BAYESUVIUS

A VISUAL DICTIONARY OF BAYESIAN NETWORKS AND CAUSAL INFERENCE

ROBERT R. TUCCI

Bayesuvius, a visual dictionary of Bayesian Networks and Causal Inference

Robert R. Tucci
www.ar-tiste.xyz

December 1, 2024

This book is constantly being expanded and improved. To download the latest version, go to `https://github.com/rrtucci/Bayesuvius`

Bayesuvius
by Robert R. Tucci
Copyright ©2020-2023, Robert R. Tucci.

This work is licensed under the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 United States License. To view a copy of this license, visit the link `https://creativecommons.org/licenses/by-nc-nd/3.0/` or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042.

Figure 1: View of Mount Vesuvius from Pompeii

Figure 2: Mount Vesuvius and Bay of Naples

Contents

Foreword **19**

Appendices **20**

A Navigating the ocean of Judea Pearl's Books **21**

B CI-2-3 track **22**

C Notational Conventions and Preliminaries **26**

C.1 Some abbreviations frequently used throughout this book 26
C.2 Drawing Bayesian Networks 26
C.3 $\mathcal{N}(!a)$ 26
C.4 Indicator function (a.k.a. Truth function) 27
C.5 One hot vector 27
C.6 L^p norm 27
C.7 Special sets 28
C.8 Kronecker delta function 28
C.9 Dirac delta function 28
C.10 Majority function 29
C.11 Underlined letters indicate random variables 29
C.12 Probability distributions 30
C.13 Independence, $\perp_P$ 30
C.14 Discretization of continuous probability distributions 31
C.15 Samples, i.i.d. variables 32
C.16 Expected Value and Variance 32
C.17 Conditional Expected Value 33
C.18 Notation for covariances 33
C.19 Conditional Covariance 34
C.20 Normal Distribution 35
C.21 Uniform Distribution 36
C.22 Softmax function (a.k.a. Boltzmann Distribution) 36
C.23 Sigmoid and log-odds functions 37
C.24 Estimand, Estimator (curve-fit), Estimate, Bias 38
C.25 Maximum Likelihood Estimate, Likelihood Ratio Test 39

C.26 Mean Square Error (MSE) . . . 40
C.27 Cramer-Rao Bound . . . 42
C.28 Bayes Rule, Bayesian Updating And Conjugate Priors . . . 46
C.29 Entropy, Kullback-Leibler divergence, Cross-Entropy . . . 47
C.30 Definition of various entropies used in Shannon Information Theory 48
C.31 Mean log likelihood asymptotic behavior . . . 50
C.32 Arc Strength (Arc Force) . . . 51
C.33 Pearson Chi-Squared Test . . . 51
C.34 Demystifying Population and Sample Variances . . . 52
C.35 Independence of $\widehat{\mu}$ and $\widehat{\sigma^2}$ . . . 54
C.36 Chi-square distribution . . . 55
C.37 Student's t-distribution . . . 56
C.38 Hypothesis testing and 3 classic test statistics (Likelihood, Score, Wald) . . . 59
C.39 Error Bars . . . 62
C.40 Confidence Interval . . . 63
C.41 Score p-value . . . 65
C.42 Convex/Concave functions, Jensen's Inequality . . . 67
C.43 Chebyshev's inequality . . . 68
C.44 Short Summary of Boolean Algebra . . . 70
C.45 Laplace transform . . . 71
C.45.1 Examples . . . 73
C.45.2 Properties . . . 74
C.46 Z-transform . . . 80
C.46.1 Examples . . . 83
C.46.2 Properties . . . 84
C.47 Legendre Transformation (dual functions) . . . 87
C.47.1 Examples . . . 88
C.47.2 Properties . . . 90
C.47.3 Connection to Fourier transform and Quantum Mechanics . . 93
C.48 Numpy tensor methods . . . 94

D Linear Regression, Ordinary Least Squares (OLS) 100
D.1 LR, assuming x_σ are non-random . . . 101
D.1.1 Derivation of LR From Minimization of Error . . . 101
D.1.2 Geometry of LR with non-random x_σ. . . . 102
D.1.3 LR Goodness of Fit, R^2 . . . 104
D.2 LR, assuming x_σ are random . . . 106
D.2.1 Transforming from non-random to random x_σ . . . 107
D.2.2 LR with random x_σ, expressed in derivative notation . . . 109
D.2.3 R^2 with random x_σ . . . 113
D.2.4 Regressing Residuals . . . 114

D.3 Logistic Regression (LoR) 115

E Classic Bayesian Network Apps **117**

F Definition of a Bayesian Network **121**

G Bayesian Networks, Causality and the Passage of Time **125**
G.1 Unifying Principle of this book 125
G.2 You say tomato, I say tomato 126
G.3 A dataset is causal model free 126
G.4 What is causality? 127
G.5 Bayesian Networks and the passage of time 128
G.6 Advice for the DAG-phobic 129

1 AdaBoost **130**
1.1 AdaBoost for general ensemble of w-classifiers 130
1.2 AdaBoost for ensemble of tree stumps 134

2 ANOVA **136**
2.1 Law of Total Variance 136
2.2 Sum of Squares Estimates 137
2.3 F-statistic and hypothesis testing 139

3 ARACNE structure learning **141**

4 Backdoor Adjustment Formula **143**
4.1 Examples 143

5 Back Propagation (Automatic Differentiation) **148**
5.1 Toy Example 148
5.2 General Theory 149
5.2.1 Jacobians 149
5.2.2 Bnets for function composition, forward propagation and back propagation 150
5.3 Application to Neural Networks 152
5.3.1 Absorbing b_i^λ into $w_{i|j}$. 152
5.3.2 Bnets for function composition, forward propagation and back propagation for NN 153
5.4 General bnets instead of Markov chains induced by layered structure of NNs 156

6 Bell and Clauser-Horne Inequalities in Quantum Mechanics **157**

7 Berkson's Paradox **158**

8 Binary Decision Diagrams **160**
8.0.1 Conversion of Binary Tree into BDD 162
8.1 Equivalent Bnet 162

9 Chow-Liu Trees and Tree Augmented Naive Bayes (TAN) **165**
9.1 Chow-Liu Trees 165
9.2 Tree Augmented Naive Bayes (TAN) 169

10 Control Theory (linear, deterministic) **171**
10.1 Basic feedback model 172
10.2 Classical model (analog) 173
10.3 Modern model (analog) 176
10.4 Classical model (digital) 180
10.5 Modern model (digital) 180
10.5.1 Discretizing derivatives 180
10.5.2 Solving Difference Equation 182
10.6 Higher than first order differential (or difference) equations 184
10.6.1 Differential Equations 184
10.6.2 Difference Equations 185
10.7 Time-Invariance, Causality, Stability 185
10.8 Controllability, Observability 187
10.9 Signal Flow Graph 187

11 Copula **192**
11.1 Examples 195

12 Counterfactual Reasoning **198**
12.1 The 3 Rungs of Causal AI 198
12.2 Do operator 199
12.3 Imagine operator 199

13 Cross-Validation **203**

14 DAG Extraction From Text (DEFT) or Time-Series **206**

15 Dataset Shift and Batch Normalization **207**
15.1 Covariate Shift 208
15.2 Concept Shift 208
15.3 Batch Normalization 209

16 Decision Trees **210**
16.1 Conversion to Bnet 211
16.2 Structure Learning for Dtrees 212
16.2.1 Information Gain, Gini 213

16.2.2 Pseudo-code 216

17 Decisions Based on Rungs 2 and 3: COMING SOON 218

18 Difference-in-Differences 219
18.1 John Snow, DID and a cholera transmission pathway 219
18.2 PO analysis 221
18.3 Linear Regression 223

19 Diffusion Models 226
19.1 Bnet for DM 226
19.2 Mean Values $M^{t-1}(x^t)$ and $M_\theta^{t-1}(x^t)$ 229
19.3 Loss function $\mathcal{L}$ 232
19.4 Algorithms for training and sampling DM 234

20 Digital Circuits 235
20.1 Mapping any dcircuit to a bnet 235
20.1.1 Option A of Fig.20.2 235
20.1.2 Option B of Fig.20.2 236

21 Dimensionality Reduction 237

22 Do Calculus 239
22.1 3 Rules of Do Calculus 242
22.2 Parent Adjustment Formula 244
22.3 Backdoor Adjustment Formula 246
22.4 Frontdoor Adjustment Formula 247
22.5 Comparison of Backdoor and Frontdoor adjustment formulae 248
22.6 Do operator for DEN bnets 248

23 Do Calculus proofs 252

24 D-Separation 269

25 D-Separation in Quantum Mechanics 272

26 Dynamical Bayesian Networks 273

27 Expectation Maximization 275
27.1 The EM algorithm: 276
27.1.1 Motivation 277
27.2 Minorize-Maximize (MM) algorithms 277
27.3 Examples 279
27.3.1 Gaussian mixture 279

27.3.2 Blood Genotypes and Phenotypes 280
27.3.3 Missing Data/Imputation 282

28 Factor Analysis **283**

29 Factor Graphs **287**

30 Finite State Machine **290**
30.1 Deterministic FSM 290
30.1.1 Example 290
30.1.2 Precise Definition 291
30.2 Non-deterministic FSM 292
30.2.1 Example 292
30.2.2 Precise Definition 293
30.3 Equivalency of deterministic and non-deterministic FSM 293

31 Frisch-Waugh-Lovell (FWL) theorem **295**
31.1 FWL, assuming x^σ are non-random 295
31.2 FWL, assuming x^σ are random 296

32 Frontdoor Adjustment Formula **298**
32.1 Examples 299

33 G-formula (Sequential Backdoor Adjustment Formula) **301**

34 Gaussian Nodes with Linear Dependence on Parents **305**

35 Generalized Linear Model (GLM) **308**
35.1 Exponential Family of Distributions 308
35.2 GLM 310

36 Generative Adversarial Networks (GANs) **315**

37 Goodness of Causal Fit **320**

38 Gradient Descent **321**

39 Granger Causality **323**

40 Hidden Markov Model **326**
40.1 Calculating $P(x_t, v^n)$ and $P(x_t, x_{t+1}, v^n)$ 328
40.2 Calculating $\mathcal{F}_t$ and $\overline{\mathcal{F}}_t$ 329
40.3 Calculating $P(x^n|v^n)$ 330
40.4 Calculating $P(v^n|A, B, \pi)$ 331
40.5 Calculating $\widehat{x}^n$ (Viterbi algorithm) 332

40.6 Calculating $\widehat{A}$, $\widehat{B}$, $\widehat{\pi}$ (Baum-Welch algorithm) 333

41 Identification of do queries via LDEN diagrams 336

42 Influence Diagrams & Utility Nodes 338
42.1 Definitions 339
42.2 Variable Elimination Algorithm (VEA) 343

43 Instrumental Inequality and beyond 346
43.1 I-inequality 346
43.1.1 I-inequality for binary z,d,y 348
43.2 Bounds on Effect of IV on treatment outcome y 349

44 Instrumental Variables 352
44.1 δ with unmeasured confounder 352
44.2 δ (with unmeasured confounder) can be inferred via IV 353
44.3 More general bnets with IVs 354
44.4 Instrumental Inequality 355

45 Jackknife Resampling 356
45.1 Case $A = \Phi(x^n; n) = \frac{1}{n}\sum_\sigma x^\sigma$ 358

46 Junction Tree Algorithm 360

47 Kalman Filter 361
47.1 Prediction Problem 362
47.2 Solution 363
47.3 Simple Example 364
47.4 Invariants 365
47.5 Derivation of Solution 365

48 Kernel Principal Component Analysis (Kernel PCA) 367

49 LATE (Local Average Treatment Effect) 371

50 LDEN with feedback loops 377

51 Linear and Logistic Regression via grad descent 384
51.1 Generalization to x with multiple components (features) 386
51.2 Alternative $V(b, m)$ for logistic regression 386

52 Linear Deterministic Bnets with External Noise (LDEN Bnets) 388
52.1 Example of LDEN bnet 389
52.2 LDEN equations and their 2 solutions 390

52.3 Fully connected LDEN bnets . . . 390
52.3.1 Fully connected LDEN bnet with $nx = 2$ . . . 391
52.3.2 Fully connected LDEN bnet with $nx = 3$ . . . 392
52.3.3 Fully connected LDEN bnet with arbitrary nx . . . 395
52.4 Not fully connected LDEN bnets . . . 396
52.5 LDEN bnet with conditioned nodes . . . 397
52.6 SCuMpy . . . 397
52.7 Non-linear DEN bnets . . . 397

53 Marginalizer Nodes 399

54 Markov Blankets 401

55 Markov Chain Monte Carlo (MCMC) 403
55.1 Inverse Cumulative Sampling . . . 403
55.2 Rejection Sampling . . . 405
55.3 Metropolis-Hastings Sampling . . . 406
55.4 Gibbs Sampling . . . 409
55.5 Importance Sampling . . . 411

56 Markov Chains 412

57 Mediation Analysis 413

58 Mendelian Randomization 419

59 Message Passing and Bethe Free Energy 421
59.1 2MRFs . . . 421
59.2 Message Passing Intuition . . . 422
59.3 $-\ln Z_\theta$= Free Energy (FE) . . . 426
59.4 $-\ln Z_{\theta^*}$= Minimum FE . . . 427
59.5 $-\ln Z_\theta^{tree}$=Tree FE (a.k.a. Bethe FE) . . . 428
59.6 $-\ln Z_{\theta^*}^{tree}$= Tree Minimum FE, and message passing . . . 429

60 Message Passing, Pearl's theory 432
60.1 Distributed Soldier Counting . . . 432
60.2 Spring Systems . . . 434
60.3 BP for Markov Chains . . . 435
60.4 BP Algorithm for Polytrees . . . 441
60.4.1 How BP algo for polytrees reduces to the BP algo for Markov chains . . . 444
60.5 Derivation of BP Algorithm for Polytrees . . . 445
60.6 Example of BP algo for a Tree . . . 448
60.7 Bipartite bnets . . . 452

60.8 BP for bipartite bnets (BP-BB) 453
60.8.1 BP-BB and general BP agree on Markov chains 455
60.8.2 BP-BB and general BP agree on tree bnets. 457
60.9 BP-BB and sum-product decomposition 458

61 Message Passing in Quantum Mechanics 460

62 Meta-learners for estimating ATE 461

63 Missing Data, Imputation 465
63.1 Imputation via EM 466
63.2 Imputation via MCMC 469
63.3 Multiple Imputations 470

64 Modified Treatment Policy 471
64.1 One time MTP 471
64.2 $\Delta_{|c}$ estimand 475
64.3 Estimates of $\Delta_{|c}$ 478
64.3.1 Empirical estimate of $\Delta_{|c}$ 478
64.3.2 OR estimate of $\Delta_{|c}$ 478
64.4 Other Estimands besides $\Delta_{|c}$ 481
64.5 Multi-time MTP 481

65 Monty Hall Problem 484

66 Multi-armed Bandits 486
66.1 Bnet for MAB 487
66.2 Reward functions 489
66.3 Regret functions 491
66.4 Strategies with random exploration 491
66.4.1 ϵ-greedy algorithm 492
66.4.2 ϵ_t-greedy algorithm 493
66.5 Strategies with nonrandom exploration 493
66.5.1 Upper Confidence Bounds (UCB) algorithms 493
Frequentist UCB (UCB1) algorithm 493
Bayesian UCB algorithm 494
66.5.2 Thompson Sampling MAB (TS-MAB) algorithm 495
Bnet for general TS-MAB algorithm 495
TS-MAB algorithm with Beta agent and Bernoulli environment 497
TS-MAB algorithm, skeletal reprise 498
66.5.3 Grad-MAB algorithm 499

67 Naive Bayes 502

68 Neural Networks **503**
68.1 Activation Functions $\mathcal{A}_i^\lambda : \mathbb{R} \to \mathbb{R}$ . . . 504
68.2 Weight optimization via supervised training and gradient descent . 505
68.3 Non-dense layers . . . 507
68.4 Autoencoder NN . . . 509

69 Noisy-OR gate **510**
69.1 3 ways to interpret the parameters π_i . . . 511

70 Non-negative Matrix Factorization **514**
70.1 Bnet interpretation . . . 514
70.2 Simplest recursive algorithm . . . 515

71 Observationally Equivalent DAGs **516**
71.1 Examples . . . 516

72 Omitted Variable Bias **519**

73 Personalized Expected Utility **525**
73.1 Goal of PEU Theory . . . 526
73.2 Bnets for PEU Theory . . . 527
73.3 Bounds on EU for unspecified bnet . . . 527
73.4 Bounds on EU for specific bnet families . . . 530

74 Personalized Treatment Effects **531**
74.1 Goal, Strategy and Rationale of PTE theory . . . 532
74.2 Bnets for PTE theory . . . 534
74.3 $ATE = PB - PH$ . . . 535
74.4 Probabilities Relevant to PTE theory . . . 536
74.5 Symmetry . . . 541
74.6 Linear Programming Problem . . . 542
74.7 Special constraints . . . 543
74.8 Matrix representation of probabilities . . . 545
74.9 Bounds on Exp. Probs. imposed by Obs. Probs. . . . 549
74.10 Bounds on $PNS3$ for unspecified bnet . . . 550
74.11 Bounds on $PNS3$ for specific bnet families . . . 556
74.12 Bounds on ATE imposed by Obs. Probs. . . . 556
74.13 Bounds on PNS in terms of ATE and Obs. Probs. . . . 557
74.14 Numerical Examples . . . 558

75 Petri Nets **560**
75.1 Original Petri Net . . . 561
75.1.1 Simple Example of Petri Net . . . 561
75.1.2 Precise Definition of Petri Net . . . 563

75.1.3 Firing of a Petri Net . . . 564
75.2 Variants . . . 565
75.2.1 Finite State Machine and Turing Machine . . . 565
75.2.2 Continuous Petri Net . . . 565
75.2.3 Colored Petri Net . . . 566
75.2.4 Stochastic Petri Net . . . 567
75.2.5 Bayes-Petri Net . . . 567

76 Plate Notation 571

77 Potential Outcomes and Beyond 574
77.1 G and G_{den} bnets, the starting point bnets . . . 575
77.2 G bnet with nodes $y^\sigma(0), y^\sigma(1)$ added to it. . . . 577
77.3 Expected Values of treatment outcome y^σ . . . 579
77.4 Translation Dictionary . . . 580
77.5 $\mathcal{Y}_{|d,x} = \mathcal{Y}_{d|d,x}$ (SUTVA) . . . 581
77.6 Conditional Independence Assumption (CIA) . . . 581
77.7 Treatment Effects . . . 582
77.8 Insights into what makes treatment effects equal and $\mathcal{Y}_{1|0} = \mathcal{Y}_1$ . . . 584
77.9 G_{do+} bnet . . . 585
77.10 $ACE = ATE$ . . . 587
77.11 Good, Bad Controls . . . 587
77.12 PO Confounder Sensitivity Analysis . . . 589
77.13 Strata-Matching . . . 591
77.13.1 Exact strata-matching . . . 591
Estimates of Treatment Effects . . . 591
Example, estimation of treatment effects . . . 594
77.13.2 Approximate strata-matching . . . 595
77.13.3 Unbiased strata-matching estimates . . . 596
77.14 (SDO, ATE) space . . . 597
77.15 Propensities . . . 600
77.15.1 Propensity based estimates of Treatment Effects . . . 602
77.15.2 Doubly Robust Estimates of Treatment Effects . . . 604
77.15.3 Positivity . . . 605
77.16 Multi-time PO bnets (Panel Data) . . . 606

78 Principal Component Analysis 609

79 Program evaluation and review technique (PERT) 616
79.1 Example . . . 618

80 Random Forest and Bagging 622
80.1 Bagging (with fully-featured bags) . . . 622

80.2 Bagging (with randomly-shortened bags) 624

81 Recurrent Neural Networks 625
81.1 Language Sequence Modeling 628
81.2 Other types of RNN 628
81.2.1 Long Short Term Memory (LSTM) unit (1997) 630
81.2.2 Gated Recurrence Unit (GRU) (2014) 632

82 Regression Discontinuity Design 634
82.1 PO analysis 634
82.2 Linear Regression 636

83 Regularization of Loss Functions 637
83.1 L^p norm ROLF 638
83.1.1 L^1 norm ROLF can lead to sparsity 638
83.1.2 L^2 norm ROLF for Least Squares 640
83.2 Proximal functions 641
83.3 Proximal ROLF 643
83.4 Unobserved Nodes of a bnet 644

84 Reinforcement Learning (RL) 646
84.1 Exact RL bnet 649
84.2 Actor-Critic RL bnet 651
84.3 Q function learning RL bnet 653

85 Reliability Box Diagrams and Fault Tree Diagrams 655
85.1 Minimal Cut Sets 661

86 Restricted Boltzmann Machines 663

87 ROC curves 665
87.1 Terminology Table Adapted from Wikipedia Ref.[177] 668

88 Scoring the Nodes of a Learned Bnet 670
88.1 Probability Distributions and Special Functions 671
88.2 Single node with no parents 673
88.3 Multiple nodes with any number of parents 675
88.4 Bayesian Scores 677
88.5 Information Theoretic scores 677

89 Selection Bias Removal 679
89.1 Pre and Post Switch Nodes 680
89.2 Removing SB from passive query $P(y|x)$ 682
89.3 Removing SB from active query $P(y|\mathcal{D}x)$ 683

90 Sentence Splitting with SentenceAx **686**
90.1 Preliminary Conventions . . . 687
90.1.1 Tensor Notation . . . 687
90.1.2 PyTorch conventions . . . 688
90.2 Bayesian Network for this model . . . 692
90.3 Loss for this model . . . 694

91 Shannon Information Theory **697**
91.1 Introduction . . . 698
91.2 Preliminaries and Notation . . . 699
91.3 Integration Over P-types . . . 702
91.3.1 Integration Over Univariate P-type . . . 702
91.3.2 Integration Over Multivariate P-types . . . 705
91.3.3 Integration Over Conditional P-types . . . 706
91.3.4 Dirac Delta Functions For P-type Integration . . . 709
91.4 Source Coding (Lossy Compression) . . . 710
91.5 Channel Coding . . . 712
91.6 Source Coding With Distortion . . . 721
91.7 Appendix: Some Integrals Over Polytopes . . . 728

92 Shapley Explainability **734**
92.0.1 Numerical examples of SHAP . . . 737

93 Simpson's Paradox **740**
93.1 Pearl Causality . . . 742
93.2 Numerical Example . . . 744

94 Stochastic Differential Equations **745**
94.1 Notation . . . 745
94.2 White Noise and Brownian Motion . . . 746
94.3 SDE bnet . . . 748
94.4 Simple Properties of SDE . . . 750
94.4.1 STD with Constant Coefficients (CC) . . . 751
94.4.2 Transition Probability Matrices . . . 751
94.4.3 Markov chain . . . 751
94.4.4 Chapman-Kolgomorov Equation . . . 752
94.4.5 Martingale . . . 752
94.5 Itô Integral . . . 752
94.6 Fokker-Planck Equation . . . 754
94.7 First and second order statistics . . . 759
94.7.1 For general SDE . . . 759
94.7.2 In case SDE has CC . . . 761
94.8 Fourier Analysis for CC case . . . 762

94.9 Lamperti Transformation . . . 765
94.10 Feynman-Kac Path Integrals . . . 766
94.11 Karhunen–Loève series . . . 767
94.12 Girsamov Theorem . . . 770
94.13 Doob's Transform . . . 771
94.14 Appendix: Some explicitly solvable examples . . . 774
94.15 Appendix: Ornstein-Uhlenbeck recurring example . . . 775

95 Structure and Parameter Learning for Bnets 776
95.1 Overview . . . 776
95.2 Score based SL algorithms . . . 778
95.3 Constraint based SL algorithms . . . 779
95.4 Pseudo-code for some bnet learning algorithms . . . 780

96 Support Vector Machines And Kernel Method 782
96.1 Learning Algorithm for SVM Classifier . . . 783
96.2 Linear (dot-product) Kernel . . . 784
96.3 Alternatives to Linear Kernel . . . 787
96.4 Random Forest and Kernel Method . . . 788

97 Survival Analysis 789
97.1 $S(t)$ estimates . . . 791
97.1.1 No-censoring estimate of $S(t)$ . . . 791
97.1.2 Kaplan-Meier estimate of $S(t)$ . . . 791
97.2 $\lambda(t)$ models . . . 796
97.2.1 $\lambda(t)$ independent of covariates Z . . . 796
97.2.2 $\lambda(t)$ dependent on covariates Z . . . 797
97.3 $S_0(t)$ estimates . . . 800

98 Synthetic Controls 802
98.1 PO analysis . . . 804

99 Table 2 Fallacy 806

100 Targeted Estimator 809
100.1 Goal, Strategy, and Rationale of TE theory . . . 809
100.2 Functional Calculus . . . 811
100.3 Linear Approximation of $\Psi[P_N]$ . . . 813
100.4 ATE estimand . . . 814
100.5 ATE estimates . . . 815
100.5.1 Ψ^E . . . 815
100.5.2 Ψ^G . . . 816
100.5.3 Ψ^{IPW} . . . 816

100.5.4 Ψ^{LIPW} . . . 816
100.5.5 Ψ^{LIPW++} (a.k.a. Ψ^{TMLE}) . . . 820
100.6 Ψ^{TMLE} in practice . . . 822

101 Thermodynamics, a Causal Perspective 824

102 Time Series Analysis: ARMA and VAR 826
102.1 White noise . . . 826
102.2 Backshift operator . . . 827
102.3 Metrics . . . 827
102.4 Definition of $ARMA(p,q)$, $AR(p)$ and $MA(q)$. . . . 829
102.5 Solving $AR(p)$ . . . 831
102.6 Solving $MA(q)$ . . . 832
102.7 Solving $ARMA(p,q)$ . . . 833
102.8 Auto-correlation and partial auto-correlation . . . 833
102.9 Generating function of auto-correlation . . . 837
102.10 Impulse Response . . . 838
102.11 $AR(p)$ and Yule-Walker equations . . . 840
102.12 Forecasting . . . 841
102.13 Model Learning . . . 847
102.14 Differencing and $ARIMA(p,d,q)$ . . . 847
102.15 Parameter Learning . . . 850
102.15.1 PL of $AR(p)$ . . . 851
102.15.2 PL of $MA(q)$ . . . 853
102.15.3 PL of $ARMA(p,q)$ . . . 855
102.16 $VAR(p)$ . . . 856

103 Transfer Learning 858

104 Transformer Networks 860
104.1 Tensor Notation . . . 861
104.2 Recurrent Neural Net with Attention . . . 862
104.2.1 Single Head Attention . . . 862
104.2.2 Multi-Head Attention . . . 865
104.3 Vanilla tranet . . . 867
104.3.1 Single Head Attention . . . 872
104.3.2 Multi-Head Attention . . . 873
104.3.3 Encoder . . . 875
104.3.4 Decoder . . . 877
104.4 BERT . . . 879
104.4.1 BERT parameter values . . . 879
104.4.2 BERT Embedding . . . 880
104.4.3 BERT training . . . 881

105 Transportability of Causal Knowledge **882**

106 Turbo Codes **886**
106.1 Decoding Algorithm 889
106.2 Message Passing Interpretation of Decoding Algorithm 891

107 Turing Machine **892**
107.1 Example 893
107.2 Precise Definition 894

108 Uplift Modelling **895**
108.1 UP types 895
108.2 Some Relevant Technical Formulas from Chapter 77 897
108.3 UP Analysis 897
108.4 UP Decision Trees 899
108.4.1 Appendix, connection between Δ_c and $\Delta_{c|j}$ 903

109 Variational Bayesian Approximation for Medical Diagnosis **904**

110 Variational Bayesian Approximation via D_{KL} **908**
110.1 Free Energy $\mathcal{F}(\vec{x})$ 910

111 XGBoost **913**
111.1 Divergences 913
111.2 Minimizing Cost function for single tree 915
111.3 Leaf Splitting 918
111.4 Pruning 918
111.5 Feature Binning 920
111.6 Final estimate of target attribute 920
111.7 Bnet for XGBoost 921

112 YAML for bnet storage **923**
112.1 Getting acquainted with YAML 924
112.2 Storing Bnets 925

113 Zero Information Transmission (Graphoid Axioms) **928**
113.1 Consequences of Eq.(113.5) 929

Bibliography **931**

Foreword

Welcome to Bayesuvius! a proto-book uploaded to github.

A different Bayesian network is discussed in each chapter. Each chapter title is the name of a Bnet. Chapter titles are in alphabetical order.

This is a volcano in its early stages. First version uploaded to a github repo called Bayesuvius on June 24, 2020. First version only covers 2 Bnets (Linear Regression and GAN). I will add more chapters periodically. Remember, this is a moonlighting effort so I can't do it all at once.

For any questions about notation, please go to Notational Conventions section.

Requests and advice are welcomed.

Thanks for reading this
Robert R. Tucci
www.ar-tiste.xyz

ADDENDA

- **August 15, 2021:** At this point in time, the book has grown to 67 Chapters and 433 pages. Today, I am self-publishing it as an ebook at Amazon and similar outlets. It will still be free.

Appendices

Appendix A

Navigating the ocean of Judea Pearl's Books

The fields of bnets and causal inference are heavily indebted to Judea Pearl and his collaborators.

Pearl has written 4 books that I have used in writing Bayesuvius. His 1988 book Ref.[60] dates back to his pre-causal period. That book I used to learn about topics such as d-separation, belief propagation, Markov-blankets, and noisy-ORs. 3 other books that he wrote later, in his causal period, are:

1. In 2000 (1st ed.), and 2013 (2nd ed.), Pearl published what is so far his most technical and exhaustive book on the subject of causality, Ref.[62].

2. In 2016, he released together with Glymour and Jewell, a less advanced "primer" on causality, Ref.[65].

3. In 2018, he released together with Mackenzie his lovely "The Book of Why", Ref.[66].

Those 3 books I used to learn about causality topics such as Do Calculus, backdoor and frontdoor adjustment formulae, linear deterministic bnets with exogenous noise, and counterfactuals.

A micro poem written by me to celebrate Judea Pearl and his work:

I, Robot

Let other robots `talk()`,
while I,
`talk()`, `do()` and `imagine()`.

Appendix B

CI-2-3 track

Figure B.1: Democritus quote, Acropolis Caryatids background

As discussed in Chapter 12, Judea Pearl has proposed 3 rungs of Causal Inference (CI). This book covers all 3 rungs.

Confusingly, it has become common to use the term CI to refer to only the highest 2 rungs of the CI hierarchy; i.e, rung 2 (do operations) and rung 3 (imagining/counterfactual thinking). Also confusingly, rung 1 uses causal diagrams and is often referred to as "inference", so it could reasonably have been defined as the whole of CI, but Pearl has defined the CI hierarchy to include two more rungs. To

Figure B.2: CI meme

patch over this linguistic confusion, I sometimes refer to rung 1 as "prediction", or as "predictive inference" instead of calling it merely "inference". Also, when I want to be precise, I use the term "CI-2-3" to refer to CI restricted to only rungs 2 and 3.

Here is a subset of chapters that I call the CI-2-3 track, that are devoted mostly to rungs 2 and 3.

1. Backdoor Adjustment Formula
2. Berkson's Paradox
3. Counterfactual Reasoning
4. Decisions Based on Rungs 2 and 3: COMING SOON
5. Difference-in-Differences
6. Do Calculus

Figure B.3: Slide from Pearl talk at IJCAI-2022. Putting the joking of Fig.B.2 aside, let me emphasize that CI advocates are not trying to vanish NNs from AI. To us, NNs and bnets are different tools, like hammer and saw. We believe AI should use both tools. For those who are trying to do CI using a NN instead of a bnet, it looks to me like you are trying to use a hammer to cut wood. Why don't you cut with a saw instead? As Pearl says in this slide, CI elevates Data Science from Deep Learning (curve-fitting) to Deep Understanding.

7. Do Calculus proofs
8. D-Separation
9. Frisch-Waugh-Lovell (FWL) theorem
10. Frontdoor Adjustment Formula
11. G-formula (Sequential Backdoor Adjustment Formula)
12. Goodness of Causal Fit
13. Granger Causality
14. Identification of do queries via LDEN diagrams
15. Instrumental Inequality and beyond
16. Instrumental Variables
17. LATE (Local Average Treatment Effect)
18. LDEN with feedback loops
19. Linear Deterministic Bnets with External Noise (LDEN Bnets)

20. Mediation Analysis
21. Mendelian Randomization
22. Meta-learners for estimating ATE
23. Modified Treatment Policy
24. Omitted Variable Bias
25. Personalized Expected Utility
26. Personalized Treatment Effects
27. Potential Outcomes and Beyond
28. Regression Discontinuity Design
29. Selection Bias Removal
30. Simpson's Paradox
31. Survival Analysis
32. Synthetic Controls
33. Targeted Estimator
34. Transportability of Causal Knowledge
35. Uplift Modelling

Appendix C

Notational Conventions and Preliminaries

C.1 Some abbreviations frequently used throughout this book

- AI/ML = Artificial Intelligence/Machine Learning
- bnet= Bnet= Bayesian Network
- CPT = Conditional Probabilities Table, same as TPM
- DAG = Directed Acyclic Graph
- i.i.d.= independent identically distributed.
- RCT= Randomized Controlled Trial, a.k.a. A/B testing.
- TPM= Transition Probability Matrix, same as CPT

C.2 Drawing Bayesian Networks

Most B nets (also Petri nets, Finite State Machines and Turing Machines) in this book were drawn using the LaTex package `xy-pic` (xypic), or the Python app `texnn` (Ref.[98]). `texnn` is a Python wrapper for `xy-pic` that I wrote specially for this book.

Simple trees in this book were also drawn using `xy-pic`, but more complicated ones were drawn using the LaTex packages `istgame` and `dirtree`.

C.3 $\mathcal{N}(!a)$

$\mathcal{N}(!a)$ will denote a normalization constant that does not depend on a. For example, $P(x) = \mathcal{N}(!x)e^{-x}$ where $\int_0^\infty dx\ P(x) = 1$.

C.4 Indicator function (a.k.a. Truth function)

$$\mathbb{1}(\mathcal{S}) = \begin{cases} 1 \text{ if } \mathcal{S} \text{ is true} \\ 0 \text{ if } \mathcal{S} \text{ is false} \end{cases} \tag{C.1}$$

For example, $\delta(x, y) = \mathbb{1}(x = y)$.

C.5 One hot vector

A **one hot vector** is a vector with all entries equal to zero with the exception of a single entry which is one. A **one cold vector** is a vector with all entries equal to one with the exception of a single entry which is zero. For example, if $x^n = (x_0, x_1, \ldots, x_{n-1})$ and $x_i = \delta(i, 0)$ then x^n is one hot.

Two types of sets that one frequently encounters are **categorical sets** (a.k.a. "nominal sets", i.e., sets with "named" elements, with elements given a "nomme") and **numerical sets** (a.k.a. "ordinal sets", i.e., sets with "ordered" elements). For example, $\{1, 2, 5\}$ is a numerical set because its elements have a natural order, and {cat, dog, bird} is a categorical set because its elements don't have a natural order.

In Machine Learning (ML), one often encodes categorical sets as one-hot vectors. For example, suppose we have 4 binary registers (i.e., nodes) x_3, x_2, x_1, x_0 and the categorical set {cat, dog, canary}. Then a possible **one-hot encoding** of the set is cat=0001, dog=0010 and canary=0100. This differs from a **binary encoding** of the set such as cat=0000, dog=0001, canary=0011. Clearly, a binary encoding requires fewer registers than a one-hot encoding to encode the same set, and the one-hot encoding of a set with n elements requires n or more registers.

C.6 L^p norm

For $p \in [0, \infty]$ and $\vec{x} \in \mathbb{R}^n$ or $\vec{x} \in \mathbb{C}^n$ (note that n and p are generally not the same), the L^p **norm** $\| \vec{x} \|_p$ of $\vec{x}$ is defined as

$$\| \vec{x} \|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \tag{C.2}$$

For example,

$$\| \vec{x} \|_0 = \sum_{i=1}^{n} \mathbb{1}(|x_i| > 0) = \text{number of non-zero } x_i \tag{C.3}$$

$$\| \vec{x} \|_1 = \sum_{i=1}^{n} |x_i| \tag{C.4}$$

$$\| \vec{x} \|_2 = \sqrt{\sum_{i=1}^{n} (x_i)^2} \tag{C.5}$$

$$\| \vec{x} \|_3 = \left(\sum_{i=1}^{n} |x_i|^3 \right)^{\frac{1}{3}} \tag{C.6}$$

$$\begin{aligned} \| \vec{x} \|_\infty &= \lim_{p \to \infty} \| \vec{x} \|_p && \text{(C.7)} \\ &= \lim_{p \to \infty} ((\max_i |x_i|)^p)^{\frac{1}{p}} \quad \text{(because one } |x_i|^p \text{ dominates the rest)} && \text{(C.8)} \\ &= \max_i |x_i| && \text{(C.9)} \end{aligned}$$

Note that as $\lim_{p \to 0} \| \vec{x} \|_p \neq \| \vec{x} \|_0$. In fact, as $p \to 0$,

$$\begin{aligned} \| \vec{x} \|_p &\to (\text{number of non-zero } x_i)^{\frac{1}{p}} \quad (\text{because } |x|^0 = 1 \text{ for } x \neq 0) && \text{(C.10)} \\ &\to \| \vec{x} \|_0^{\frac{1}{p}} \to \infty && \text{(C.11)} \end{aligned}$$

When n is large and only a few of the n components of $\vec{x} \in \mathbb{C}^n$ are non-zero, we say $\vec{x}$ is **sparse**. $\| \vec{x} \|_0$ is used to measure the **sparsity** of vectors.

Fig.C.1 shows the **unit balls** $\{\vec{x} \in \mathbb{R}^n : \| \vec{x} \|_p \leq 1\}$ for various values of p and for $n = 2$. $\{\vec{x} \in \mathbb{R}^2 : \| \vec{x} \|_0 \leq 1\}$ is not shown. It equals all the x and y axes, because, by definition, it contains all $(x, y) \in \mathbb{R}^2$ such that $x = 0$ or $y = 0$ or both (i.e., 0 or 1 non-zero components).

C.7 Special sets

Define $\mathbb{Z}, \mathbb{R}, \mathbb{C}$ to be the integers, real numbers and complex numbers, respectively.

For $a < b$, define $\mathbb{Z}_I$ to be the integers in the interval I, where $I = [a, b], [a, b), (a, b], (a, b)$ (i.e, I can be closed or open on either side).

$A_{>0} = \{k \in A : k > 0\}$ for $A = \mathbb{Z}, \mathbb{R}$.

C.8 Kronecker delta function

For x, y in discrete set S,

$$\delta(x, y) = \begin{cases} 1 \text{ if } x = y \\ 0 \text{ if } x \neq y \end{cases} \tag{C.12}$$

C.9 Dirac delta function

For $x, y \in \mathbb{R}$,

$$\int_{-\infty}^{+\infty} dx \; \delta(x - y) f(x) = f(y) \tag{C.13}$$

Figure C.1: Unit Balls $\{\vec{x} \in \mathbb{R}^n : \| \vec{x} \|_p \leq 1\}$ for various values of p and for $n = 2$.

C.10 Majority function

The **majority function** is defined as follows.

$$\texttt{majority}(L) = \begin{array}{l} \text{most common element of list } L \\ \text{(ties resolved by chance)} \end{array} \tag{C.14}$$

Note that the majority function acts on lists, not sets. By definition, all elements of a set appear only once in the set. $\texttt{majority}(L)$ is usually used when the elements of L are categorical (i.e., not real numbers). When they are real numbers, it makes more sense to use, instead of $\texttt{majority}(L)$, a simple average of the elements of L.

C.11 Underlined letters indicate random variables

Random variables will be indicated by underlined letters and their values by non-underlined letters. Each node of a bnet will be labeled by a random variable. Thus, $\underline{x} = x$ means that node $\underline{x}$ is in state x.

It is more conventional to use an upper case letter to indicate a random variable and a lower case letter for its state. Thus, $X = x$ means that random variable X is in state x. However, we have opted in this book to avoid that notation, because we often want to define certain lower case letters to be random variables or, conversely, define certain upper case letters to be non-random variables.

C.12 Probability distributions

$P_{\underline{x}}(x) = P(\underline{x} = x) = P(x)$ is the probability that random variable $\underline{x}$ equals $x \in val(\underline{x})$. $val(\underline{x})$ is the set of states (i.e., values) that $\underline{x}$ can assume and $n_{\underline{x}} = |val(\underline{x})|$ is the size (a.k.a. cardinality) of that set. Hence,

$$\sum_{x \in val(\underline{x})} P_{\underline{x}}(x) = 1 \tag{C.15}$$

$$P_{\underline{x},\underline{y}}(x, y) = P(\underline{x} = x, \underline{y} = y) = P(x, y) \tag{C.16}$$

$$P_{\underline{x}|\underline{y}}(x|y) = P(\underline{x} = x|\underline{y} = y) = P(x|y) = \frac{P(x, y)}{P(y)} \tag{C.17}$$

C.13 Independence, $\perp_P$

Two variables $\underline{a}$ and $\underline{b}$ are said to be **independent** if

$$P(a, b) = P(a)P(b) \tag{C.18}$$

or, equivalently, when $P(b) \neq 0$,

$$P(a|b) = P(a) \tag{C.19}$$

In such a case, we write $\underline{a} \perp_P \underline{b}$ or just $\underline{a} \perp \underline{b}$ if the distribution P being alluded to is clear. Note that $\underline{a} \perp \underline{b}$ iff $\underline{b} \perp \underline{a}$.

Two variables $\underline{a}$ and $\underline{b}$ are said to be **conditionally independent at fixed** $\underline{z}$ if

$$P(a, b|z) = P(a|z)P(b|z) \tag{C.20}$$

or, equivalently, if $P(b|z) \neq 0$,

$$P(a|b, z) = P(a|z) \tag{C.21}$$

In such a case, we write $\underline{a} \perp_P \underline{b}|\underline{z}$ or just $\underline{a} \perp \underline{b}|\underline{z}$ if the distribution P being alluded to is clear. Note that $\underline{a} \perp \underline{b}|\underline{z}$ iff $\underline{b} \perp \underline{a}|\underline{z}$.

In this book, we use both $(\underline{a}, \underline{b}) \perp^{sep} \underline{x}$ and $(\underline{a}, \underline{b}) \perp^{joint} \underline{x}$.

When we write $(\underline{a}, \underline{b}) \perp^{sep} \underline{x}$, we mean that $\underline{a} \perp \underline{x}$ and $\underline{b} \perp \underline{x}$ separately; that is, $P(a|x) = P(a)$ and $P(b|x) = P(b)$.

When we write $(\underline{a}, \underline{b}) \perp^{joint} \underline{x}$ or simply $(\underline{a}, \underline{b}) \perp \underline{x}$, we mean $P(a, b|x) = P(a, b)$.

Claim 1 *$(\underline{a}, \underline{b}) \perp^{joint} \underline{x}$ implies $(\underline{a}, \underline{b}) \perp^{sep} \underline{x}$*

proof: Summing $P(a, b|x) = P(a, b)$ over a gives $P(b|x) = P(b)$, and summing it over b gives $P(a|x) = P(a)$.
QED

Claim 2 *Assume* $(\underline{a}, \underline{b}) \perp^{sep} \underline{x}$ *and* $\underline{a} \perp \underline{b}$. *Then* $(\underline{a}, \underline{b}) \perp^{joint} \underline{x}$ *iff* $(\underline{a}, \underline{b}) \perp^{sep} \underline{x}$.

proof: We already know that $(\underline{a}, \underline{b}) \perp^{joint} \underline{x}$ implies $(\underline{a}, \underline{b}) \perp^{sep} \underline{x}$. To prove the converse, note that $(\underline{a}, \underline{b}) \perp^{sep} \underline{x}$ means

$$P(a|x) = P(a) \ , \quad P(b|x) = P(b) \tag{C.22}$$

Hence

$$P(a, b|x) = P(a|x)P(b|x) = P(a)P(b) = P(a, b) \tag{C.23}$$

QED

C.14 Discretization of continuous probability distributions

The TPM of a node of a bnet can be either a discrete or a continuous probability distribution. To go from continuous to discrete, one replaces integrals over states of a node by sums over new states, and Dirac delta functions by Kronecker delta functions. More precisely, consider a function $f : [a, b] \to \mathbb{R}$. Express $[a, b]$ as a union of small, disjoint (except for one point) closed sub-intervals (bins) of length Δx. Name one point in each bin to be the representative of that bin, and let $val(\underline{x})$ be the set of all the bin representatives. This is called discretization or binning. Then

$$\frac{1}{(b-a)} \int_{[a,b]} dx \ f(x) \to \frac{\Delta x}{(b-a)} \sum_{x \in val(\underline{x})} f(x) = \frac{1}{n_{\underline{x}}} \sum_{x \in val(\underline{x})} f(x) \ . \tag{C.24}$$

Both sides of last equation are 1 when $f(x) = 1$. Furthermore, if $y \in val(\underline{x})$, then

$$\int_{[a,b]} dx \ \delta(x - y) f(x) = f(y) \to \sum_{x \in val(\underline{x})} \delta(x, y) f(x) = f(y) \ . \tag{C.25}$$

As usual in this book, let $val(\underline{x})$ denote the set of values that the random variable $\underline{x}$ can take. When $val(\underline{x}) \subset \mathbb{R}$, we will assume that $val(\underline{x})$ for a probability distribution $P(x)$ can be either a discrete or a continuous subset of $\mathbb{R}$.[1] When $val(\underline{x})$ is a discrete subset of $\mathbb{R}$, $P(x)$ will denote a probability distribution for which $\sum_{x \in val(\underline{x})} P(x) = 1$, whereas when $val(\underline{x})$ is continuous, $P(x)$ will denote a probability density for which $\int_{x \in val(\underline{x})} dx \ P(x) = 1$.

[1] By a "continuous set" we mean a finite set of intervals each of which has non-zero length.

C.15 Samples, i.i.d. variables

$$\vec{x} = (x[0], x[1], x[2] \ldots, x[nsam(\vec{x}) - 1]) = x[:] \tag{C.26}$$

$nsam(\vec{x})$ is the number of samples of $\vec{x}$. $\underline{x}[\sigma] \in val(\underline{x})$ are i.i.d. (independent identically distributed) samples with

$$x[\sigma] \sim P_{\underline{x}} \;\; (\text{i.e. } P_{\underline{x[\sigma]}} = P_{\underline{x}}) \tag{C.27}$$

$$P(\underline{x} = x) = \frac{1}{nsam(\vec{x})} \sum_{\sigma} \mathbb{1}(x[\sigma] = x) \tag{C.28}$$

Hence, for any $f : val(\underline{x}) \to \mathbb{R}$,

$$\sum_{x} P(\underline{x} = x) f(x) = \frac{1}{nsam(\vec{x})} \sum_{\sigma} f(x[\sigma]) \tag{C.29}$$

If we use two sampled variables, say $\vec{x}$ and $\vec{y}$, in a given bnet, their number of samples $nsam(\vec{x})$ and $nsam(\vec{y})$ need not be equal.

$$P(\vec{x}) = \prod_{\sigma} P(x[\sigma]) \tag{C.30}$$

$$\sum_{\vec{x}} = \prod_{\sigma} \sum_{x[\sigma]} \tag{C.31}$$

$$\partial_{\vec{x}} = [\partial_{x[0]}, \partial_{x[1]}, \partial_{x[2]}, \ldots, \partial_{x[nsam(\vec{x})-1]}] \tag{C.32}$$

$$P(\vec{x}) \approx [\prod_{x} P(x)^{P(x)}]^{nsam(\vec{x})} \tag{C.33}$$

$$= e^{nsam(\vec{x}) \sum_x P(x) \ln P(x)} \tag{C.34}$$

$$= e^{-nsam(\vec{x}) H(P_{\underline{x}})} \tag{C.35}$$

C.16 Expected Value and Variance

Given a random variable $\underline{x}$ with states $val(\underline{x})$ and a function $f : val(\underline{x}) \to \mathbb{R}$, define

$$E_{\underline{x}}[f(\underline{x})] = E_{x \sim P(x)}[f(x)] = \sum_{x} P(x) f(x) \tag{C.36}$$

$$Var_{\underline{x}}[f(\underline{x})] = E_{\underline{x}}\left[(f(\underline{x}) - E_{\underline{x}}[f(\underline{x})])^2\right] \tag{C.37}$$

$$= E_{\underline{x}}[f(\underline{x})^2] - (E_{\underline{x}}[f(\underline{x})])^2 \tag{C.38}$$

$$E[\underline{x}] = E_{\underline{x}}[\underline{x}] \tag{C.39}$$

$$Var[\underline{x}] = Var_{\underline{x}}[\underline{x}] \tag{C.40}$$

C.17 Conditional Expected Value

Given a random variable $\underline{x}$ with states $val(\underline{x})$, a random variable $\underline{y}$ with states $val(\underline{y})$, and a function $f : val(\underline{x}) \times val(\underline{y}) \to \mathbb{R}$, define

$$E_{\underline{x}|\underline{y}}[f(\underline{x}, \underline{y})] = \sum_x P(x|\underline{y}) f(x, \underline{y}) \ , \tag{C.41}$$

$$E_{\underline{x}|\underline{y}=y}[f(\underline{x}, y)] = E_{\underline{x}|y}[f(\underline{x}, y)] = \sum_x P(x|y) f(x, y) \ . \tag{C.42}$$

Note that

$$\begin{aligned} E_{\underline{y}}[E_{\underline{x}|\underline{y}}[f(\underline{x}, \underline{y})]] &= \sum_{x,y} P(x|y) P(y) f(x, y) && \text{(C.43)} \\ &= \sum_{x,y} P(x, y) f(x, y) && \text{(C.44)} \\ &= E_{\underline{x},\underline{y}}[f(\underline{x}, \underline{y})] \ . && \text{(C.45)} \end{aligned}$$

C.18 Notation for covariances

Consider two random variables $\underline{x}, \underline{y}$.

- Mean value of $\underline{x}$

$$\langle \underline{x} \rangle = E_{\underline{x}}[\underline{x}] \tag{C.46}$$

- Signed distance of $\underline{x}$ to its mean value

$$\Delta \underline{x} = \underline{x} - \langle \underline{x} \rangle \tag{C.47}$$

- Covariance of $(\underline{x}, \underline{y})$

$$Cov(\underline{x}, \underline{y}) = \left\langle \underline{x}, \underline{y} \right\rangle = \left\langle \Delta \underline{x} \Delta \underline{y} \right\rangle = \left\langle \underline{x} \underline{y} \right\rangle - \langle \underline{x} \rangle \left\langle \underline{y} \right\rangle \tag{C.48}$$

  $\left\langle \underline{x}, \underline{y} \right\rangle$ is symmetric (i.e., $\left\langle \underline{x}, \underline{y} \right\rangle = \left\langle \underline{y}, \underline{x} \right\rangle$) and bilinear (i.e., $\left\langle \sum_i \alpha_i \underline{x}_i, \underline{y} \right\rangle = \sum_i \alpha_i \left\langle \underline{x}_i, \underline{y} \right\rangle$, where $\alpha_i \in \mathbb{R}$ are non-random scalars and $\underline{x}_i, \underline{y} \in \mathbb{R}$ are real-valued random variables.)

- Variance of $\underline{x}$

$$Var(\underline{x}) = \langle \underline{x}, \underline{x} \rangle \tag{C.49}$$

- Standard deviation or $\underline{x}$

$$\sigma_{\underline{x}} = \sqrt{\langle \underline{x}, \underline{x} \rangle} \tag{C.50}$$

- Correlation Coefficient of $(\underline{x}, \underline{y})$

$$\rho_{\underline{x},\underline{y}} = \frac{\left\langle \underline{x}, \underline{y} \right\rangle}{\sqrt{\left\langle \underline{x}, \underline{x} \right\rangle \left\langle \underline{y}, \underline{y} \right\rangle}} \tag{C.51}$$

- Partial derivative of $\underline{y}$ wrt (i.e., with respect to) $\underline{x}$

$$\partial_{\underline{x}} \underline{y} = \frac{\partial \underline{y}}{\partial \underline{x}} = \frac{\left\langle \underline{x}, \underline{y} \right\rangle}{\left\langle \underline{x}, \underline{x} \right\rangle} = \rho_{\underline{x},\underline{y}} \frac{\sigma_{\underline{y}}}{\sigma_{\underline{x}}} \tag{C.52}$$

C.19 Conditional Covariance

Let $\underline{x}, \underline{y}, \underline{a}$ be random variables. The covariance $Cov(\underline{x}, \underline{y}|\underline{a} = a)$ of $\underline{x}$ and $\underline{y}$ given $\underline{a} = a$, is defined the same way as $Cov(\underline{x}, \underline{y})$, except that all expected values are conditioned on $\underline{a} = a$.

$$Cov(\underline{x}, \underline{y}|\underline{a} = a) = \left\langle \underline{x}, \underline{y} \right\rangle^{|a} = \left\langle (\underline{x} - \langle \underline{x} \rangle^{|a})(\underline{y} - \left\langle \underline{y} \right\rangle^{|a}) \right\rangle^{|a} \tag{C.53}$$

where

$$\langle \underline{x} \rangle^{|a} = E_{\underline{x}|a}[\underline{x}] \ . \tag{C.54}$$

In this book, we will use the following notation for conditional averages. For any random variables $\underline{x}, \underline{y}, \underline{a}$, let

$$E_{|a}[\underline{x}] = \langle \underline{x} \rangle^{|a} \quad \text{(mean)} \tag{C.55}$$

$$\left\langle \underline{x}, \underline{y} \right\rangle^{|a} = \left\langle \underline{x}\underline{y} \right\rangle^{|a} - \langle \underline{x} \rangle^{|a} \left\langle \underline{y} \right\rangle^{|a} \quad \text{(covariance)} \tag{C.56}$$

$$\sigma_{\underline{x}}^{|a} = \sqrt{\langle \underline{x}, \underline{x} \rangle^{|a}} \quad \text{(standard deviation)} \tag{C.57}$$

$$\rho_{\underline{x},\underline{y}}^{|a} = \frac{\left\langle \underline{x}, \underline{y} \right\rangle^{|a}}{\sigma_{\underline{x}}^{|a} \sigma_{\underline{y}}^{|a}} = \left[\frac{\left\langle \underline{x}, \underline{y} \right\rangle}{\sigma_{\underline{x}} \sigma_{\underline{y}}} \right]^{|a} \quad \text{(correlation)} \tag{C.58}$$

$$\partial_{\underline{x}}^{|a} \underline{y} = \left[\frac{\partial}{\partial \underline{x}} \right]^{|a} \underline{y} = \frac{\left\langle \underline{x}, \underline{y} \right\rangle^{|a}}{\left\langle \underline{x}, \underline{x} \right\rangle^{|a}} = \rho_{\underline{x},\underline{y}}^{|a} \frac{\sigma_{\underline{y}}^{|a}}{\sigma_{\underline{x}}^{|a}} = \left[\rho_{\underline{x},\underline{y}} \frac{\sigma_{\underline{y}}}{\sigma_{\underline{x}}} \right]^{|a} \quad \text{(partial derivative)} \tag{C.59}$$

"$|a$" means that the variable $\underline{a}$ is held fixed to a when taking all averages.

C.20 Normal Distribution

For $x, \mu, \sigma \in \mathbb{R}$, $\sigma > 0$, we define the Normal Distribution (see Fig.C.2) by

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} . \tag{C.60}$$

For a **standard deviation** σ, the **precision** τ is defined as $\tau = \frac{1}{\sigma^2}$.

Claim 3 *If*

$$\underline{x}_1 \sim \mathcal{N}(\mu_1, \sigma_1^2) \tag{C.61}$$

and

$$\underline{x}_2 \sim \mathcal{N}(\mu_2, \sigma_2^2) \tag{C.62}$$

then

$$\underline{x} = \underline{x}_1 + \underline{x}_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2) . \tag{C.63}$$

proof:

$$\begin{aligned} P(\underline{x} = x) &= \mathcal{N}(!x) \int_{-\infty}^{+\infty} dx_2 \; P(\underline{x}_1 + \underline{x}_2 = x | \underline{x}_2 = x_2) P(x_2) && \text{(C.64)} \\ &= \mathcal{N}(!x) \int_{-\infty}^{+\infty} dx_2 \; \mathcal{N}(x - x_2; \mu_1, \sigma_1^2) \mathcal{N}(x_2; \mu_2, \sigma_2^2) && \text{(C.65)} \\ &= \mathcal{N}(x; \mu_1 + \mu_2; \sigma_1^2 + \sigma_2^2) && \text{(C.66)} \end{aligned}$$

QED

The **Standard Normal Distribution** $P_{SND}(x)$ and its cumulative distribution $\Phi(x)$ are defined by

$$P_{SND}(x) = \mathcal{N}(x; \mu = 0, \sigma = 1) \tag{C.67}$$

$$\Phi(x) = \int_{-\infty}^{x} dx' \; P_{SND}(x') \tag{C.68}$$

The **error function** $\mathrm{erf} : \mathbb{R} \to [-1, 1]$ is defined by

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x du \; e^{-\frac{u^2}{2}} \tag{C.69}$$

Note that

$$\Phi(x) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}(x) \tag{C.70}$$

Eq.(C.70) is interpreted geometrically in Fig.C.3.

Figure C.2: Normal Distribution $\mathcal{N}(x; \mu, \sigma^2)$.

Figure C.3: Plot of Standard Normal Distribution $P_{SND}(x)$. Values of $\mathrm{erf}(x)$ and $\Phi(x)$ equal indicated areas.

C.21 Uniform Distribution

For $a < b$, $x \in [a, b]$

$$\mathcal{U}(x; a, b) = \frac{1}{b-a} \tag{C.71}$$

C.22 Softmax function (a.k.a. Boltzmann Distribution)

The Softmax function is defined by

$$P(x_i|x.) = \frac{e^{x_i}}{\sum_i e^{x_i}} = \mathrm{softmax}(x.)(i) \tag{C.72}$$

The Boltzmann distribution is defined as

$$P(\underline{E}_a = E_a) = \frac{\exp(-\frac{E_a}{kT})}{\sum_a \exp(-\frac{E_a}{kT})} = P(\frac{-E_a}{kT}|E.) \tag{C.73}$$

for a system with energies E_a and temperature T, where k is Boltzmann's constant.

The function softmax() is called softmax because if we approximate the exponentials, both in the numerator and denominator of Eq.(C.72), by the largest one of them or zero, we get

$$\text{softmax}(x.)(i) \approx \delta(i, \operatorname*{argmax}_k x_k) \ . \tag{C.74}$$

Thus, softmax$(x.)(i)$ returns a continuous function that approximates a Kronecker delta function. The softmax function doesn't really return the soft maximum of a finite set, so its name is a bit of a misnomer. A better name for it would have been "soft Kronecker delta function".

Note that

$$\frac{\partial \ln P(x_i|x)}{\partial x_a} = \frac{\partial}{\partial x_a} \ln \left[\frac{e^{x_i}}{\sum_i e^{x_i}} \right] = \delta(a,i) - P(x_a|x) \tag{C.75}$$

For 2 variables x_0, x_1,

$$P(x_0|x.) = \frac{e^{x_0}}{e^{x_0} + e^{x_1}} \tag{C.76}$$

$$= \text{smoid}(x_0 - x_1) \ , \tag{C.77}$$

$$P(x_1|x.) = \text{smoid}(x_1 - x_0) \ . \tag{C.78}$$

C.23 Sigmoid and log-odds functions

The **sigmoid (a.k.a. exp-it, logistic) function** smoid:$\mathbb{R} \to [0,1]$ is defined by

$$\text{smoid}(x) = \frac{1}{1+e^{-x}} \tag{C.79}$$

smoid() is monotonically increasing with smoid$(-\infty) = 0$, smoid$(0) = 1/2$ and smoid$(+\infty) = 1$. Note that for $x << 0$, smoid$(x) \approx e^x$, which is why "smoid" is also called "expit".

$$\text{smoid}(x) + \text{smoid}(-x) = \frac{1}{1+e^{-x}} + \frac{1}{1+e^{x}} \tag{C.80}$$

$$= \frac{2+e^x+e^{-x}}{2+e^x+e^{-x}} \tag{C.81}$$

$$= 1 \tag{C.82}$$

The **log-odds (a.k.a. log-it) function** lodds:$[0,1] \to \mathbb{R}$ is defined by

$$\text{lodds}(p) = \ln \frac{p}{1-p} \tag{C.83}$$

Note that for $0 < p << 1$, $\text{lodds}(x) \approx \ln p$, which is why "lodds" is also called "logit".

Note that for $x << 1$, $\text{smoid}(x) \approx e^x << 1$, so $\text{lodds}(e^x) \approx \ln(e^x) = x$. More generally, it is easy to check that for any $p \in [0,1]$ and $x \in \mathbb{R}$,

$$\text{lodds}[\text{smoid}(x)] = x \tag{C.84}$$

$$\text{smoid}[\text{lodds}(p)] = p \tag{C.85}$$

Hence, lodds() is the inverse of smoid() and vice-versa.

Claim 4

$$\text{smoid}'(x) = \text{smoid}(x)[1 - \text{smoid}(x)] \tag{C.86}$$

$$\text{smoid}''(x) = \text{smoid}'(x)[1 - 2\text{smoid}(x)] \tag{C.87}$$

proof:

In this proof, we will abbreviate $\text{smoid}(x)$ by $s(x)$.

$$1 - s(x) = 1 - \frac{1}{1+e^{-x}} = \frac{e^{-x}}{1+e^{-x}} \tag{C.88}$$

$$s'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = s(x)[1 - s(x)] \tag{C.89}$$

$$\begin{aligned} s''(x) &= s'(x)[1 - s(x)] + s(x)(-1)s'(x) && \text{(C.90)} \\ &= s'(x)[1 - 2s(x)] && \text{(C.91)} \\ &= s(x)[1 - s(x)][1 - 2s(x)] && \text{(C.92)} \end{aligned}$$

QED

C.24 Estimand, Estimator (curve-fit), Estimate, Bias

For an **estimand** θ, an **estimator (a.k.a. curve-fit)** $\underline{\widehat{\theta}}$ gives **estimate** $E[\underline{\widehat{\theta}}(\theta)] = \theta + b$ with **bias** b. We say this estimate is an **unbiased estimate** if $b = 0$.

Note that, strictly speaking, an estimator is a function waiting to be averaged over and denoted by a letter with a hat, whereas an estimate is a real number denoted by a letter without a hat. Unfortunately, the words "estimator" and "estimate" are often used interchangeably, as if they were synonyms. And often the estimate $\theta + b$ is denoted by a letter with a hat too. In some sense, an estimator is an estimate of a curve, so it's understandable that the terms "estimator" and "estimate" are used synonymously. In this book, we will bow to traditional practice and also use the terms "estimator" and "estimate" synonymously, and use a letter with a hat to denote either of them. This is not ambiguous as long as we don't use the same letter with a hat to denote two different quantities, of course. When we need to distinguish semantically between the real value and the function, we will call the function a curve-fit, and the real value the estimate.

C.25 Maximum Likelihood Estimate, Likelihood Ratio Test

Given a bnet, let $P(x|\theta)$ be its full joint probability distribution, where x denotes the joint state of all the nodes and θ denotes all the parameters. $P(x|\theta)$ is often called the **likelihood function of** θ and is denoted by

$$L(\theta) = P(x|\theta) \tag{C.93}$$

It's called a likehood of θ because, even though it's a probability, it isn't the probability of θ, but rather of x.

The value of θ that we obtain by maximizing $L(\theta)$ over θ is called the **maximum likelihood estimate (MLE) of** θ. Let us denote it by $\widehat{\theta}$. Note that[2]

$$\sup_{\theta \in S} L(\theta) = L(\widehat{\theta}) \tag{C.94}$$

Let S_0, S_1 be disjoint sets such that $S = S_0 \cup S_1$. We'll say the **null hypothesis** H_0 holds if $\theta \in S_0$, and the **alternative hypothesis** H_1 holds if $\theta \in S_1$. The **likelihood ratio (LR) test statistic** is defined by

$$R = -2 \ln \left(\frac{\sup_{\theta \in S_0} L(\theta)}{\sup_{\theta \in S} L(\theta)} \right) \tag{C.95}$$

$R \geq 0$ and $R = 0$ if $S_0 = S$. For some small $c > 0$, if $R < c$, then we reject the alternative hypothesis, and if $R > c$, we accept it.

If $S_0 = \{\theta_0\}$, then

[2] "sup" stands for supremum. It's a generalization of the function max() to arbitrary sets that might not be discrete or finite. If S is a finite set, then $\sup_{\theta \in S} f(\theta) = \max_{\theta \in S} f(\theta)$ for any function $f : S \to \mathbb{R}$. Likewise, "inf" stands for infimum, and it generalizes the min() function.

$$R = -2[\ln L(\theta_0) - \ln L(\widehat{\theta})] \tag{C.96}$$

C.26 Mean Square Error (MSE)

Suppose we are given $nsam$ samples $y^\sigma \in \mathbb{R}$ labeled by an index σ, and a curve-fit $\widehat{y}^\sigma(a) \in \mathbb{R}$ that depends on a parameter $a \in \mathbb{R}$. Define the **Mean Square Error (MSE)** by

$$MSE(a) = \frac{1}{nsam} \sum_\sigma (y^\sigma - \widehat{y}^\sigma(a))^2 \ . \tag{C.97}$$

For example, in Linear Regression (LR), we have $\widehat{y}^\sigma = a_0 + a_1 x^\sigma$ where $a = (a_0, a_1)$ is a deterministic parameter. If the samples y^σ are i.i.d, then we can also write

$$MSE(a) = E_{|a}[(\underline{y} - \underline{\widehat{y}}(a))^2] \ . \tag{C.98}$$

and for LR, $\underline{\widehat{y}}(a) = a_0 + a_1 \underline{x}$.

Define the **residual** $\Delta \underline{y}$ by:

$$\Delta \underline{y}(a) = \underline{y} - \underline{\widehat{y}}(a) \quad (\text{Hence } \underline{y} = \underline{\widehat{y}} + \Delta \underline{y}) \tag{C.99}$$

In the rest of this section, we will discuss the case that $\widehat{y}^\sigma(a)$ is independent of x^σ. I call this the **deterministic MSE (D-MSE)** model. Note that this is different from the LR case where $\widehat{y}^\sigma(a)$ does depend on x^σ. In LR, we are trying to fit a line to a cigar-shaped 2-D scatter plot. Here, we are just trying to estimate the mean value (center of mass) of a scatter plot.

Claim 5 *MSE is minimized over all functions $\widehat{y}$ if*

$$\widehat{y} = E_{|a}[\underline{y}] \tag{C.100}$$

proof:

$$MSE = E_{|a}[\underline{y}^2] - 2\widehat{y} E_{|a}[\underline{y}] + \widehat{y}^2 \tag{C.101}$$

$$0 = \frac{d}{d\widehat{y}} MSE = 2(-E_{|a}[\underline{y}] + \widehat{y}) \tag{C.102}$$

Hence,

$$\widehat{y} = E_{|a}[\underline{y}] \tag{C.103}$$

QED

Sometimes, we will use the notation

$$\widehat{y}_{MSE} = E_{|a=a_{MSE}}[\underline{y}] \ . \tag{C.104}$$

Claim 6 *Suppose $f(a)$ is a function of a. If $\widehat{y} = E_{|a}[\underline{y}]$, then*

$$E_{|a}[\Delta \underline{y}] = E[\Delta \underline{y}] = 0 \tag{C.105}$$

$$E_{|a}\left[\Delta \underline{y} f(\underline{a})\right] = E\left[\Delta \underline{y} f(\underline{a})\right] = 0 \tag{C.106}$$

proof:

$$E_{|a}[\Delta \underline{y}] = E_{|a}\left[\underline{y} - E_{|a}[\underline{y}]\right] = E_{|a}[\underline{y}] - E_{|a}[\underline{y}] = 0 \tag{C.107}$$

$$E[\Delta \underline{y}] = E_{\underline{a}}[E_{|\underline{a}}[\Delta \underline{y}]] = 0 \tag{C.108}$$

$$E_{|\underline{a}}\left[\Delta \underline{y} f(\underline{a})\right] \quad = \quad f(\underline{a}) \underbrace{E_{|\underline{a}}[\Delta \underline{y}]}_{=0} \tag{C.109}$$

$$E[\Delta \underline{y} f(\underline{a})] = E_{\underline{a}}[E_{|\underline{a}}[\Delta \underline{y} f(\underline{a})]] = 0 \tag{C.110}$$

QED

Claim 7 *If $\widehat{y} = E_{|a}[\underline{y}]$, then*

$$\left\langle \Delta \underline{y}, \widehat{y} \right\rangle_{|a} = 0 \tag{C.111}$$

$$Var_{|a}[\underline{y}] = Var_{|a}[\widehat{y}] + Var_{|a}[\Delta \underline{y}] \tag{C.112}$$

The same results hold without the conditioning on a.

proof:

$$\left\langle \Delta \underline{y}, \widehat{\underline{y}} \right\rangle_{|a} \quad = \quad \underbrace{E_{|a}[\Delta \underline{y} \underbrace{\widehat{\underline{y}}}_{f(\underline{a})}]}_{=0} - \underbrace{E_{|a}[\Delta \underline{y}]}_{=0} E_{|a}[\widehat{\underline{y}}] \tag{C.113}$$

$$\begin{aligned} Var_{|a}[\underline{y}] &= \left\langle \widehat{y} + \Delta \underline{y}, \widehat{y} + \Delta \underline{y} \right\rangle_{|a} && \text{(C.114)} \\ &= \langle \widehat{y}, \widehat{y} \rangle_{|a} + \left\langle \Delta \underline{y}, \Delta \underline{y} \right\rangle_{|a} \text{ (by Eq.(C.111))} && \text{(C.115)} \\ &= Var_{|a}[\widehat{y}] + Var_{|a}[\Delta \underline{y}] && \text{(C.116)} \end{aligned}$$

The same proof holds if we remove all the $|a$ subscripts.
QED

Fig.C.4 illustrates how $\underline{y} = \widehat{\underline{y}} + \Delta \underline{y}$ and the variances of these quantities add.

Figure C.4: $\underline{y} = \hat{y} + \Delta y$ and the variances (not standard deviations) of these quantities add.

C.27 Cramer-Rao Bound

This discussion of the Cramer-Rao (CR) bound is based on Ref.[126].

Suppose $\underline{x}$ is a random variable with values $x \in val(\underline{x})$ and $\theta \in \mathbb{R}$ is a parameter. For any function $f_{\underline{x},\theta} : val(\underline{x}) \times \mathbb{R} \to \mathbb{R}$, define

$$\left\langle f_{\underline{x},\theta} \right\rangle = \sum_x P(x|\theta) f_{x,\theta} \tag{C.117}$$

$$\Delta f_{\underline{x},\theta} = f_{\underline{x},\theta} - \left\langle f_{\underline{x},\theta} \right\rangle \tag{C.118}$$

$$\left\langle f_{\underline{x},\theta}, f_{\underline{x},\theta} \right\rangle = \left\langle \Delta f_{\underline{x},\theta}\ \Delta f_{\underline{x},\theta} \right\rangle \tag{C.119}$$

Define the **log likelihood function** by

$$LL_\theta = \ln P(x|\theta) \tag{C.120}$$

Define the **Fisher information** by

$$I_\theta = \langle \partial_\theta LL_\theta, \partial_\theta LL_\theta \rangle \tag{C.121}$$

Note that $LL_\theta \leq 0$. Let θ^* be the value of θ that maximizes LL_θ.

Note that $I_\theta \geq 0$ and $I_\theta = 0$ when $\theta = \theta^*$ because $\partial_\theta LL_\theta|_{\theta=\theta^*} = 0$. This suggests that I_θ measures the distance between θ and θ^*.

Note that

$$\begin{aligned} \langle \partial_\theta LL_\theta \rangle &= \sum_x P(x|\theta) \frac{1}{P(x|\theta)} \partial_\theta P(x|\theta) && \text{(C.122)} \\ &= \partial_\theta \sum_x P(x|\theta) && \text{(C.123)} \\ &= 0 && \text{(C.124)} \end{aligned}$$

Therefore

$$I_\theta = \left\langle [\partial_\theta LL_\theta]^2 \right\rangle - \langle \partial_\theta LL_\theta \rangle^2 \tag{C.125}$$
$$= \left\langle [\partial_\theta LL_\theta]^2 \right\rangle \tag{C.126}$$

Claim 8

$$I_\theta = -\left\langle \partial_\theta^2 LL_\theta \right\rangle \tag{C.127}$$

proof:

$$I_\theta = \left\langle [\partial_\theta LL_\theta]^2 \right\rangle \tag{C.128}$$
$$= \sum_x P(x|\theta) \frac{1}{P(x|\theta)} \partial_\theta P(x|\theta) \partial_\theta \ln P(x|\theta) \tag{C.129}$$
$$= -\sum_x P(x|\theta) \partial_\theta^2 \ln P(x|\theta) + \partial_\theta \sum_x P(x|\theta) \partial_\theta \ln P(x|\theta) \tag{C.130}$$
$$= -\left\langle \partial_\theta^2 LL_\theta \right\rangle + \partial_\theta^2 \sum_x P(x|\theta) \tag{C.131}$$
$$= -\left\langle \partial_\theta^2 LL_\theta \right\rangle \tag{C.132}$$

QED

Claim 9 *If* $x = [x_i]_{i=1,2,\ldots\nu} \in \mathbb{R}^\nu$ *are i.i.d., then*

$$I_\theta = \nu \left\langle [\partial_\theta LL_{\theta,i}]^2 \right\rangle \tag{C.133}$$

where

$$LL_{\theta,i} = \ln P(x_i|\theta) \tag{C.134}$$

proof:

$$LL_\theta = \ln \prod_i P(x_i|\theta) \tag{C.135}$$
$$= \sum_i LL_{\theta,i} \tag{C.136}$$

$$I_\theta = \sum_i \sum_j \langle \partial_\theta LL_{\theta,i} \partial_\theta LL_{\theta,j} \rangle \tag{C.137}$$
$$= \sum_i \left\langle [\partial_\theta LL_{\theta,i}]^2 \right\rangle \tag{C.138}$$
$$= \nu \left\langle [\partial_\theta LL_{\theta,i}]^2 \right\rangle \tag{C.139}$$

QED

A function $t_{\underline{x}} : val(\underline{x}) \to \mathbb{R}$ is called a **test statistic** of random variable $\underline{x}$.

Claim 10 *(Cramer-Rao bound for single parameter $\theta \in \mathbb{R}$)*

$$\left\langle t_{\underline{x}}, t_{\underline{x}} \right\rangle I_\theta \geq \left[\partial_\theta \left\langle t_{\underline{x}} \right\rangle \right]^2 \tag{C.140}$$

proof:

Cauchy-Schwartz inequality

For two vectors $\vec{a}, \vec{b} \in \mathbb{R}^n$:

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos\phi \leq |\vec{a}||\vec{b}| \tag{C.141}$$

For two real valued random variables $\underline{a}, \underline{b}$:

$$\langle \underline{a}, \underline{a} \rangle \langle \underline{b}, \underline{b} \rangle \geq | \langle \underline{a}, \underline{b} \rangle |^2 \tag{C.142}$$

Replace

$$\underline{a} \to t_{\underline{x}}, \quad \underline{b} \to \partial_\theta LL_\theta \tag{C.143}$$

Then

$$\begin{aligned}
\left\langle t_{\underline{x}}, \partial_\theta LL_\theta \right\rangle &= \left\langle t_{\underline{x}} \partial_\theta LL_\theta \right\rangle - \left\langle t_{\underline{x}} \right\rangle \underbrace{\langle \partial_\theta LL_\theta \rangle}_{=0} && \text{(C.144)} \\
&= \sum_x P(x|\theta) t_{\underline{x}} \frac{1}{P(x|\theta)} \partial_\theta P(x|\theta) && \text{(C.145)} \\
&= \partial_\theta \sum_x t_{\underline{x}} P(x|\theta) && \text{(C.146)} \\
&= \partial_\theta \left\langle t_{\underline{x}} \right\rangle && \text{(C.147)}
\end{aligned}$$

QED

See Fig.C.5 for a pictorial representation of Eq.(C.140).

Now suppose the test statistic $t_{\underline{x}}$ equals an **estimator** $\widehat{\theta}$ of θ with bias $b_{\underline{x}} : val(\underline{x}) \to \mathbb{R}$.

$$t_{\underline{x}} = \widehat{\theta}(\underline{x}) = \theta + b_{\underline{x}} \tag{C.148}$$

$\widehat{\theta}$ is said to be a **biased estimator** if $b_{\underline{x}} \neq 0$ and an **unbiased estimator** if $b_{\underline{x}} = 0$.

Figure C.5: In this drawing, θ^* is the value of θ that maximizes LL_θ. According to the CR bound, the product of the variance $\left\langle t_{\underline{x}}, t_{\underline{x}} \right\rangle$ and the distance I_θ must be greater or equal to $[\partial_\theta \left\langle t_{\underline{x}} \right\rangle]^2$. At fixed $[\partial_\theta \left\langle t_{\underline{x}} \right\rangle]^2$, if the variance increases, the distance decreases, and vice versa.

Claim 11

$$\left\langle \widehat{\theta} \right\rangle = \theta + \left\langle b_{\underline{x}} \right\rangle \tag{C.149}$$

$$\left\langle \widehat{\theta}, \widehat{\theta} \right\rangle \geq \frac{[1 + \partial_\theta \left\langle b_{\underline{x}} \right\rangle]^2}{I_\theta} \tag{C.150}$$

$$\left\langle [\widehat{\theta} - \theta]^2 \right\rangle \geq \frac{[1 + \partial_\theta \left\langle b_{\underline{x}} \right\rangle]^2}{I_\theta} + \left\langle b_{\underline{x}} \right\rangle^2 \tag{C.151}$$

proof:

$$\partial_\theta \left\langle t_{\underline{x}} \right\rangle = \partial_\theta \left\langle \theta + b_{\underline{x}} \right\rangle \tag{C.152}$$

$$= \partial_\theta \left[\theta + \left\langle b_{\underline{x}} \right\rangle \right] \tag{C.153}$$

$$= 1 + \partial_\theta \left\langle b_{\underline{x}} \right\rangle \tag{C.154}$$

Eq.(C.150) follows from Eq.(C.140) once we replace $t_{\underline{x}}$ by $\widehat{\theta}$.

Let

$$\Delta\widehat{\theta} = \widehat{\theta} - \left\langle \widehat{\theta} \right\rangle = \underbrace{(\widehat{\theta} - \theta)}_{\xi} - \left\langle b_{\underline{x}} \right\rangle \tag{C.155}$$

Then

$$0 = \left\langle \Delta\widehat{\theta} \right\rangle = \langle \xi \rangle - \left\langle b_{\underline{x}} \right\rangle \tag{C.156}$$

$$\frac{[1+\partial_\theta \left\langle b_{\underline{x}} \right\rangle]^2}{I_\theta} \leq \left\langle [\Delta\widehat{\theta}]^2 \right\rangle \tag{C.157}$$

$$= \left\langle \xi^2 - 2\xi \left\langle b_{\underline{x}} \right\rangle + \left\langle b_{\underline{x}} \right\rangle^2 \right\rangle \tag{C.158}$$

$$= \left\langle \xi^2 \right\rangle - \left\langle b_{\underline{x}} \right\rangle^2 \tag{C.159}$$

QED

Multi-dimensional case: parameter $\theta = [\theta_1, \theta_2, \ldots, \theta_n]^T \in \mathbb{R}^n$ and test statistic $t_{\underline{x}} = [t_{\underline{x},1}, t_{\underline{x},2}, \ldots, t_{\underline{x},n}]^T \in \mathbb{R}^n$ are column vectors.

Define **Fisher information matrix** by

$$[I_\theta]_{i,j} = \left\langle \partial_{\theta_i} LL_\theta, \partial_{\theta_j} LL_\theta \right\rangle = \left\langle \partial_{\theta_i} LL_\theta \; \partial_{\theta_j} LL_\theta \right\rangle \tag{C.160}$$

CR bound for multi-dimensional parameter $\theta \in \mathbb{R}^n$:

$$\text{matrix} \left[\left\langle t_{\underline{x},i}, t_{\underline{x},j} \right\rangle \right] \geq \text{matrix} \left[\partial_{\theta_i} \left\langle t_{\underline{x},a} \right\rangle [I_\theta]^{-1}_{a,b} \partial_{\theta_j} \left\langle t_{\underline{x},b} \right\rangle \right] \tag{C.161}$$

where we are using the Einstein summation convention (repeated indices are summed over). For two matrices $A, B \in \mathbb{R}^n$, $A \geq B$ means $A - B$ has non-negative eigenvalues.

C.28 Bayes Rule, Bayesian Updating And Conjugate Priors

Bayes Rule says:

$$P(\theta|x)P(x) = P(x|\theta)P(\theta) \tag{C.162}$$

Expressed diagramatically[3], we have for $\underline{x} \in \mathbb{R}$:

$$\underline{\theta} \longleftarrow \underline{x} \quad = \quad \underline{\theta} \longrightarrow \underline{x} \tag{C.163}$$

and for $\underline{x} = (\underline{x}_1, \underline{x}_2) \in \mathbb{R}^2$:

$$\begin{tikzcd} & \underline{x}_1 \arrow[dl] \arrow[d] \\ \underline{\theta} & \underline{x}_2 \end{tikzcd} \quad = \quad \begin{tikzcd} & \underline{x}_1 \arrow[d] \\ \underline{\theta} \arrow[ur] \arrow[r] & \underline{x}_2 \end{tikzcd} \tag{C.164}$$

[3]Two bnets are equated if their full probability distributions (i.e., their full instantiations) are equal numerically. For example,

$$\underline{a} \to \underline{b} \to \underline{c} = P(c|b)P(b|a)P(a) = \underline{a} \leftarrow \underline{b} \leftarrow \underline{c}$$

Note how Bayes rule allows us to reverse the direction of the arrows impinging on θ. We see from Bayes Rule that even though the directions of the arrows in a bnet can have causal motivation, a bnet with arrows reversed from their causally motivated directions can still be very useful as a calculation tool.

Another way of stating Bayes Rule is

$$\underbrace{P(\theta|x)}_{\text{posterior}} = \mathcal{N}(!\theta)\, \underbrace{P(x|\theta)}_{\text{likelihood}}\, \underbrace{P(\theta)}_{\text{prior}} \ . \tag{C.165}$$

If, for a given likelihood, the prior and posterior distributions belong to the same family (for instance, they are both Beta distributions), then we say that the prior is the **conjugate prior** of that likelihood.

For example, Beta $\sim$ Bernoulli*Beta. Hence, the Beta distribution[4] is the conjugate prior of the Bernoulli distribution[5]. More explicitly, if

$$p_1 \sim \text{Beta}(p_1; \alpha, \beta) \tag{C.166}$$

and

$$x|p_1 \sim \text{Bernoulli}(x; p_1) \ , \tag{C.167}$$

where $p_1 = P(x = 1)$, then

$$p_1|x \sim \text{Beta}(p_1; \alpha', \beta') \tag{C.168}$$

where

$$\alpha' = \alpha + x \tag{C.169}$$

$$\beta' = \beta + (1 - x) \tag{C.170}$$

Ref.[124] has a table of conjugate priors.

Conjugate priors facilitate Bayesian updating of the prior to posterior in a feedback loop(see Fig.C.6).

C.29 Entropy, Kullback-Leibler divergence, Cross-Entropy

For probability distributions $p(x), q(x)$ of $x \in val(\underline{x})$

- Entropy:

$$H(p) = -\sum_x p(x) \ln p(x) \geq 0 \tag{C.171}$$

[4]See Ref.[113] for a discussion of the Beta distribution.

[5]See Ref.[111] for a discussion of the Bernoulli distribution

Figure C.6: Bayesian updating facilitated by conjugate prior. In this figure, $x_{\leq t} = (x_0, x_1, \ldots, x_{t-1}, x_t)$.

- Kullback-Leibler divergence:

$$D_{KL}(p \parallel q) = \sum_x p(x) \ln \frac{p(x)}{q(x)} \geq 0 \tag{C.172}$$

- Cross entropy:

$$\begin{aligned} CE(p \parallel q) &= -\sum_x p(x) \ln q(x) && \text{(C.173)} \\ &= H(p) + D_{KL}(p \parallel q) && \text{(C.174)} \end{aligned}$$

C.30 Definition of various entropies used in Shannon Information Theory

- **(plain) Entropy of $\underline{x}$**

$$H(\underline{x}) = -\sum_x P(x) \ln P(x) \tag{C.175}$$

This quantity measures the spread of $P_{\underline{x}}$. $H(\underline{x}) \geq 0$ and it vanishes iff $P(x) = \delta(x, x_0)$ (deterministic case)

- **Conditional Entropy of $\underline{y}$ given $\underline{x}$**

$$\begin{aligned} H(\underline{y}|\underline{x}) &= -\sum_{x,y} P(x, y) \ln P(y|x) && \text{(C.176)} \\ &= H(\underline{y}, \underline{x}) - H(\underline{x}) && \text{(C.177)} \end{aligned}$$

This quantity measures the conditional spread of $\underline{y}$ given $\underline{x}$. $H(\underline{y}|\underline{x}) \geq 0$.

- **Mutual Information (MI) of $\underline{x}$ and $\underline{y}$.**

$$H(\underline{y} : \underline{x}) = \sum_{x,y} P(x,y) \ln \frac{P(x,y)}{P(x)P(y)} \tag{C.178}$$

$$= H(\underline{x}) + H(\underline{y}) - H(\underline{y}, \underline{x}) \tag{C.179}$$

This quantity measures the correlation between $\underline{x}$ and $\underline{y}$. $H(\underline{y} : \underline{x}) \geq 0$ and it vanishes iff $P(x,y) = P(x)P(y)$.

- **Conditional Mutual Information (CMI)[6] of $\underline{x}$ and $\underline{y}$ given $\underline{\lambda}$**

$$H(\underline{y} : \underline{x}|\underline{\lambda}) = \sum_{x,y,\lambda} P(x,y,\lambda) \ln \frac{P(x,y|\lambda)}{P(x|\lambda)P(y|\lambda)} \tag{C.180}$$

$$= H(\underline{x}|\underline{\lambda}) + H(\underline{y}|\underline{\lambda}) - H(\underline{y}, \underline{x}|\underline{\lambda}) \tag{C.181}$$

This quantity measures the conditional correlation of $\underline{x}$ and $\underline{y}$ given $\underline{\lambda}$. $H(\underline{y} : \underline{x}|\underline{\lambda}) \geq 0$ and it vanishes iff $P(x,y|\lambda) = P(x|\lambda)P(y|\lambda)$.

An interesting special case occurs when $P(\lambda) = \delta(\lambda, \lambda_0)$ (the frequentist case of no λ prior.) In that case CMI reduces to

$$H(\underline{y} : \underline{x}|\lambda_0) = \sum_{x,y} P(x,y|\lambda_0) \ln \frac{P(x,y|\lambda_0)}{P(x|\lambda_0)P(y|\lambda_0)} \geq 0 \tag{C.182}$$

- **Kullback-Leibler Divergence from $P_{\underline{x}}$ to $P_{\underline{y}}$.**

Assume random variables $\underline{x}$ and $\underline{y}$ have the same set of states $val(\underline{x}) = val(\underline{y})$. Then

$$D_{KL}(P_{\underline{x}} \parallel P_{\underline{y}}) = \sum_{x} P_{\underline{x}}(x) \ln \frac{P_{\underline{x}}(x)}{P_{\underline{y}}(x)} \tag{C.183}$$

This quantity measures a non-symmetric distance between the probability distributions $P_{\underline{x}}$ and $P_{\underline{y}}$. $D_{KL}(P_{\underline{x}} \parallel P_{\underline{y}}) \geq 0$ and it equals zero iff $P_{\underline{x}} = P_{\underline{y}}$.

[6] CMI can be read as "see me".

C.31 Mean log likelihood asymptotic behavior

Define the log likelihood by

$$LL_{y|\theta} = \ln P(y|\theta) \ . \tag{C.184}$$

In this section, we will represent averages over $\underline{y}|\theta$ by angular brackets:

$$\langle f(y) \rangle = \sum_y P(y|\theta) f(y) = E_{\underline{y}|\theta}[f(y)] \ . \tag{C.185}$$

Note that the mean log likelihood equals minus the entropy:

$$H(\underline{y}|\theta) = -\left\langle LL_{y|\theta} \right\rangle \tag{C.186}$$

Claim 12

$$\left\langle \partial_\theta LL_{y|\theta} \right\rangle = 0 \tag{C.187}$$

$$\left\langle \partial_\theta^2 LL_{y|\theta} \right\rangle = -\left\langle (\partial_\theta LL_{y|\theta})^2 \right\rangle \tag{C.188}$$

proof:

$$\begin{aligned} \left\langle \partial_\theta LL_{y|\theta} \right\rangle &= \sum_y P(y|\theta) \partial_\theta \ln P(y|\theta) && \text{(C.189)} \\ &= \sum_y \partial_\theta P(y|\theta) && \text{(C.190)} \\ &= 0 && \text{(C.191)} \end{aligned}$$

$$\begin{aligned} \left\langle \partial_\theta^2 LL_{y|\theta} \right\rangle &= \sum_y P(y|\theta) \partial_\theta \left[\frac{1}{P(y|\theta)} \partial_\theta P(y|\theta) \right] && \text{(C.192)} \\ &= -\sum_y P(y|\theta) \frac{1}{P(y|\theta)^2} [\partial_\theta P(y|\theta)]^2 + \underbrace{\sum_y \partial_\theta^2 P(y|\theta)}_{=0} && \text{(C.193)} \\ &= -\sum_y P(y|\theta) [\partial_\theta \ln P(y|\theta)]^2 && \text{(C.194)} \\ &= -\left\langle (\partial_\theta LL_{y|\theta})^2 \right\rangle && \text{(C.195)} \end{aligned}$$

QED

Define

$$\Delta\theta = \theta' - \theta \tag{C.196}$$

and

$$-\Delta H(\underline{y}|\theta) = \Delta \left\langle LL_{y|\theta} \right\rangle = \left\langle LL_{y|\theta'} \right\rangle - \left\langle LL_{y|\theta} \right\rangle \tag{C.197}$$

If we expand $\left\langle LL_{y|\theta'} \right\rangle$ as a Taylor series to second order about the point $\theta' = \theta$, we get

$$\left\langle LL_{y|\theta'} \right\rangle = \left\langle LL_{y|\theta} \right\rangle + \Delta\theta \underbrace{\left\langle \partial_\theta LL_{y|\theta} \right\rangle}_{=0} + \frac{(\Delta\theta)^2}{2} \underbrace{\left\langle \partial_\theta^2 LL_{y|\theta} \right\rangle}_{-\left\langle (\partial_\theta LL_{y|\theta})^2 \right\rangle} + \mathcal{O}((\Delta\theta)^3) \tag{C.198}$$

$$-\Delta H(\underline{y}|\theta) = \Delta \left\langle LL_{y|\theta} \right\rangle = -\frac{(\Delta\theta)^2}{2} \left\langle (\partial_\theta LL_{y|\theta})^2 \right\rangle + \mathcal{O}((\Delta\theta)^3) \tag{C.199}$$

Thus, $\theta' = \theta$ maximizes the mean log likelihood $\left\langle LL_{y|\theta} \right\rangle$ (and minimizes the entropy $H(\underline{y}|\theta)$).

Note that

$$\Delta \left\langle LL_{y|\theta} \right\rangle = \left\langle \ln \frac{P(y|\theta')}{P(y|\theta)} \right\rangle \ . \tag{C.200}$$

If we approximate the ratio of these 2 probabilities by a Gaussian,

$$\frac{P(y|\theta')}{P(y|\theta)} \approx \exp\left(-\frac{(\Delta\theta)^2}{2\sigma_\theta^2} \right) \ , \tag{C.201}$$

then

$$\sigma_\theta^2 = \left\langle (\partial_\theta LL_{y|\theta})^2 \right\rangle^{-1} \ . \tag{C.202}$$

C.32 Arc Strength (Arc Force)

Given a bnet with an arc (i.e., arrow) $\underline{x} \rightarrow \underline{y}$, we define the **arc strength or arc force** of arc $\underline{x} \rightarrow \underline{y}$ to be $H(\underline{x} : \underline{y})$ (i.e., the mutual information between $\underline{x}$ and $\underline{y}$). Evaluation of $H(\underline{x} : \underline{y})$ requires knowing $P(y|x)$, $P(x)$ and $P(y)$. $P(y|x)$ is the TPM of node $\underline{y}$, so it is immediately available from the specification of the bnet. Calculating $P(x)$ and $P(y)$ is more involved, and requires marginalizing the full probability distribution of the bnet. Such marginalizations can be done using the junction tree algorithm described in Chapter 46.

C.33 Pearson Chi-Squared Test

The **Pearson divergence** (a.k.a. **Pearson Chi-squared test statistic**) for two probability distributions $PO(x)$ and $PE(x)$, where $x \in val(\underline{x})$, is defined as follows:

$$D_{\chi^2} = \sum_x \frac{[PO(x) - PE(x)]^2}{PE(x)} = \sum_x \frac{PO^2(x)}{PE(x)} - 1 \ . \tag{C.203}$$

Usually PO is the observed probability distribution and PE is the expected, theoretical one.

As the following claim shows, the Pearson divergence is closely related to the Kullback-Leibler divergence.

Claim 13 *If* $\left|\frac{PO(x)}{PE(x)} - 1\right| << 1$ *for all* $x \in val(\underline{x})$*, then*

$$D_{KL}(PO \parallel PE) \approx D_{\chi^2} \ . \tag{C.204}$$

proof:

$$\begin{aligned} D_{KL}(PO \parallel PE) &= \sum_x PO(x) \ln \frac{PO(x)}{PE(x)} && \text{(C.205)} \\ &= \sum_x PO(x) \ln \left(1 + \frac{PO(x)}{PE(x)} - 1\right) && \text{(C.206)} \\ &\approx \sum_x PO(x) \left(\frac{PO(x)}{PE(x)} - 1\right) && \text{(C.207)} \\ &= \sum_x \frac{PO^2(x)}{PE(x)} - 1 && \text{(C.208)} \\ &= D_{\chi^2} && \text{(C.209)} \end{aligned}$$

QED

Let $nx = |val(\underline{x})|$. Let $P_{\chi^2}(y)$ be the χ^2 (with $nx - 1$ degrees of freedom) probability distribution, and let $F_{\chi^2}(\alpha)$ be its cumulative distribution. Find α such that

$$95\% = \int_0^\alpha dy \ P_{\chi^2}(y) = F_{\chi^2}(\alpha) \tag{C.210}$$

If $D_{\chi^2} < \alpha$, then we say that $PO = PE$ to 95% significance level (SL), whereas if $D_{\chi^2} > \alpha$, we say that $PO \neq PE$ to 95% SL (i.e., SL= 95%). The higher SL becomes, the higher α becomes, and the bigger the divergence D_{χ^2} has to be, before we are willing to declare that $PO \neq PE$.

C.34 Demystifying Population and Sample Variances

Let $x[\sigma] = x^\sigma$. Given i.i.d.real variables $(x^\sigma)_{\sigma=0,1,\ldots,n-1}$, let[7]

$$\widehat{\mu} = \overline{x} = \frac{1}{n} \sum_\sigma x^\sigma \tag{C.211}$$

[7]Do not confuse the sample index σ and the standard deviation σ.

$$(\widehat{\sigma^2})_\infty = \frac{1}{n}\sum_\sigma (x^\sigma - \mu)^2 \tag{C.212}$$

$$\widehat{\sigma^2} = \frac{1}{n-1}\sum_\sigma (x^\sigma - \widehat{\mu})^2 \tag{C.213}$$

Statisticians[8] call $(\widehat{\sigma^2})_\infty$ the "population variance". I will call it the **population variance for fixed** μ. Note that it depends on the fixed parameter μ. Statisticians call $\widehat{\sigma^2}$ the "sample variance". Instead, I will call $\widehat{\sigma^2}$ the **population variance for random** μ.

If one treats x^σ as a random variable, then one must treat $\widehat{\mu}$ as a random variable too. Let

$$E[\underline{x}^\sigma] = \mu \tag{C.214}$$

and

$$\left\langle \underline{x}^\sigma, \underline{x}^{\sigma'} \right\rangle = \delta(\sigma,\sigma')\sigma^2 \ . \tag{C.215}$$

Then one can show that

$$E[\underline{(\widehat{\sigma^2})_\infty}] = \frac{1}{n} E\left[\sum_\sigma (\underline{x}^\sigma - \mu)^2\right] \tag{C.216}$$

$$= \sigma^2 \tag{C.217}$$

and

$$E[\underline{\widehat{\sigma^2}}] = \frac{1}{n-1} E\left[\sum_\sigma (\underline{x}^\sigma - \underline{\widehat{\mu}})^2\right] \tag{C.218}$$

$$= \sigma^2 \tag{C.219}$$

This is the reason why we use an $n-1$ instead of an n in $\widehat{\sigma^2}$. Because it makes $E[\widehat{\sigma^2}] = \sigma^2$ so $\widehat{\sigma^2}$ is an unbiased estimator of the single individual variance σ^2.

The intuitive reason for why $\widehat{\sigma^2}$ is divided by $n-1$ instead of n is that whereas μ in $(\widehat{\sigma^2})_\infty$ is kept fixed and is "quiet", the $\underline{\widehat{\mu}}$ in $\widehat{\sigma^2}$ is a random variable, noisy instead of quiet. The fluctuations in $\underline{\widehat{\mu}}$ are strongly correlated with the fluctuations of the $\underline{x}^\sigma$, so they decrease the fluctuations in $\widehat{\sigma^2}$ compared to those in $(\widehat{\sigma^2})_\infty$. By dividing by $n-1$ instead of n, we compensate for this decrease in fluctuations so that the ratio of the numerator and denominator of $\widehat{\sigma^2}$ equals σ^2, instead of something *smaller* than

[8] In the language of Statisticians, a "population" is supposed to be so large that its μ does not fluctuate, and a "sample" is supposed to be a small subset of that population for which the μ is assumed to fluctuate. In this book, I use the word "population" to mean a set of any size containing individuals, I use the word "sub-population" to refer to a subset of the population, and I use the word "sample" (a.k.a. individual, observation, unit, record) to mean a single individual of the population.

σ^2, as would happen if were to divide by n instead of $n-1$. In terms of "degrees of freedom"(DOFs), $(\widehat{\sigma^2})_\infty$ has n DOFs (namely one for each $\underline{x}^\sigma$), whereas $\widehat{\sigma^2}$ has $n-1$ DOFs. (the presence of $\underline{\widehat{\mu}}$ subtracts one DOF). In both $(\widehat{\sigma^2})_\infty$ and $\widehat{\sigma^2}$, one divides by the number of DOFs.

C.35 Independence of $\widehat{\mu}$ and $\widehat{\sigma^2}$

Let $x[\sigma] = x^\sigma$. Consider i.i.d.real variables $(x^\sigma)_{\sigma=0,1,\ldots,n-1}$ such that[9]

$$E[\underline{x}^\sigma] = \mu \tag{C.220}$$

$$\left\langle \underline{x}^\sigma, \underline{x}^{\sigma'} \right\rangle = \delta(\sigma,\sigma')\sigma^2 \ . \tag{C.221}$$

$$\widehat{\mu} = \overline{x} = \frac{1}{n}\sum_\sigma x^\sigma \tag{C.222}$$

$$(\widehat{\sigma^2})_\infty = \frac{1}{n}\sum_\sigma (x^\sigma - \mu)^2 \tag{C.223}$$

$$\widehat{\sigma^2} = \frac{1}{n-1}\sum_\sigma (x^\sigma - \widehat{\mu})^2 \tag{C.224}$$

Claim 14 *Let*

$$\underline{\Delta}^\sigma = \underline{x}^\sigma - \mu \ . \tag{C.225}$$

For any $\sigma_1, \sigma_2, \sigma_3$,

$$\langle \underline{\Delta}^{\sigma_1}\underline{\Delta}^{\sigma_2}, \underline{\Delta}^{\sigma_3}\rangle = 0 \ . \tag{C.226}$$

proof:

Suppose $\sigma_2 \neq \sigma_3$. Then

$$\langle \underline{\Delta}^{\sigma_1}\underline{\Delta}^{\sigma_2}, \underline{\Delta}^{\sigma_3}\rangle = \underbrace{\langle \underline{\Delta}^{\sigma_2}\rangle}_{0} \langle \underline{\Delta}^{\sigma_1}, \underline{\Delta}^{\sigma_3}\rangle = 0 \ . \tag{C.227}$$

So assume $\sigma_2 = \sigma_3 = \sigma$ and evaluate $\langle \underline{\Delta}^{\sigma_1}\underline{\Delta}^{\sigma}, \underline{\Delta}^{\sigma}\rangle$.

Suppose $\sigma_1 \neq \sigma$. Then

$$\langle \underline{\Delta}^{\sigma_1}\underline{\Delta}^{\sigma}, \underline{\Delta}^{\sigma}\rangle = \underbrace{\langle \underline{\Delta}^{\sigma_1}\rangle}_{0} \langle \underline{\Delta}^{\sigma}, \underline{\Delta}^{\sigma}\rangle = 0 \ . \tag{C.228}$$

So suppose $\sigma_1 = \sigma$ and evaluate $\langle (\underline{\Delta}^\sigma)^2, \underline{\Delta}^\sigma\rangle$.

[9] Do not confuse the sample index σ and the standard deviation σ.

$$\left\langle (\underline{\Delta}^\sigma)^2, \underline{\Delta}^\sigma \right\rangle = \underbrace{\left\langle (\underline{\Delta}^\sigma)^3 \right\rangle}_{0} - \left\langle (\underline{\Delta}^\sigma)^2 \right\rangle \underbrace{\left\langle \underline{\Delta}^\sigma \right\rangle}_{0} = 0 \ . \tag{C.229}$$

QED

Claim 15

$$\left\langle \widehat{\underline{\sigma^2}}, \widehat{\underline{\mu}} \right\rangle = 0 \ . \tag{C.230}$$

proof:

$$\begin{aligned}
\left\langle \widehat{\underline{\sigma^2}}, \widehat{\underline{\mu}} \right\rangle &= \frac{1}{n(n-1)} \sum_{\sigma,\sigma'} \left\langle \left(\underline{x}^\sigma - \frac{1}{n} \sum_{\sigma''} \underline{x}^{\sigma''} \right)^2, \underline{x}^{\sigma'} \right\rangle && \text{(C.231)} \\
&= \frac{1}{n(n-1)} \sum_{\sigma,\sigma'} \left\langle \left(\underline{x}^\sigma - \frac{1}{n} \sum_{\sigma''} \underline{x}^{\sigma''} \right)^2, \underline{\Delta}^{\sigma'} \right\rangle && \text{(C.232)} \\
&= \frac{1}{n(n-1)} \sum_{\sigma,\sigma'} \left\langle \left(\underline{\Delta}^\sigma - \frac{1}{n} \sum_{\sigma''} \underline{\Delta}^{\sigma''} \right)^2, \underline{\Delta}^{\sigma'} \right\rangle && \text{(C.233)} \\
&= 0 \quad \text{by Claim 14 .} && \text{(C.234)}
\end{aligned}$$

QED

C.36 Chi-square distribution

This section is based on Ref.[121].

$$\underline{q} \longleftarrow \underline{z}.$$

Figure C.7: Bnet used to define the Chi-square distribution.

Let $q \in \mathbb{R}$ and $z. = \{z_i\}_{i=0,1,\ldots,\nu-1}$ where $z_i \in \mathbb{R}$. Consider the bnet of Fig.C.7. The TPMs, printed in blue, for that bnet, are as follows:[10]

$$P(z_i) = \mathcal{N}(z_i; \mu = 0, \sigma^2 = 1) \ . \tag{C.235}$$

We want

$$\underline{q} = \sum_{i=0}^{\nu-1} (\underline{z}_i)^2 \tag{C.236}$$

[10]Don't confuse the q independent constant $\mathcal{N}(!q)$ with the normal probability distribution $\mathcal{N}(x; \mu, \sigma^2)$.

so $P(q|z.)$ is a Dirac delta function:

$$P(q|z.) = \delta(q - \sum_{i=0}^{\nu-1}(z_i)^2) \ . \tag{C.237}$$

Therefore

$$\begin{aligned} P(q) &= \prod_{i=0}^{\nu-1}\left\{\int dz_i \ P(z_i)\right\} P(q|z.) && \text{(C.238)} \\ &= \mathcal{N}(!q)q^{\frac{\nu}{2}-1}e^{-q/2} = \chi^2(q;\nu) \ , && \text{(C.239)} \end{aligned}$$

where $\mathcal{N}(!q)$ is a constant that does not depend on q and is adjusted so that $\int_0^\infty dq \ P(q) = 1$.

C.37 Student's t-distribution

This section is based on Ref.[184].

Let $x[\sigma] = x^\sigma$. Consider i.i.d.real variables $(x^\sigma)_{\sigma=0,1,\ldots,n-1}$ such that[11]

$$E[\underline{x}^\sigma] = \mu \tag{C.240}$$

$$\left\langle \underline{x}^\sigma, \underline{x}^{\sigma'} \right\rangle = \delta(\sigma,\sigma')\sigma^2 \ . \tag{C.241}$$

$$\widehat{\mu} = \overline{x} = \frac{1}{n}\sum_\sigma x^\sigma \tag{C.242}$$

$$(\widehat{\sigma^2})_\infty = \frac{1}{n}\sum_\sigma (x^\sigma - \mu)^2 \tag{C.243}$$

$$\widehat{\sigma^2} = \frac{1}{n-1}\sum_\sigma (x^\sigma - \widehat{\mu})^2 \ . \tag{C.244}$$

If we define

$$z = \frac{\widehat{\mu} - \mu}{\frac{\sigma}{\sqrt{n}}} \ , \tag{C.245}$$

then $\underline{z}$ has a Standard Normal Distribution (SND):

$$P(z) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}} = \mathcal{N}(z;\mu=0,\sigma^2=1) \tag{C.246}$$

[11] Do not confuse the sample index σ and the standard deviation σ.

But what if we allow the standard deviation σ to fluctuate in the expression Eq.(C.245) for z? Define

$$t = \frac{\widehat{\mu} - \mu}{\sqrt{\frac{\widehat{\sigma^2}}{n}}} \ . \tag{C.247}$$

Then one can show that $\underline{t}$ has the **Student's t-distribution** $\text{Stud}(t; \nu = n-1)$ given by:

$$P(t) = \mathcal{N}(!t)(1 + \frac{t^2}{\nu})^{-\frac{\nu+1}{2}} = \text{Stud}(t; \nu = n-1) \tag{C.248}$$

Note that if we use the approximation $e^x \approx 1 + x + \mathcal{O}(x^2)$, we can show that $\text{Stud}(t)$ tends to the SND when $n >> 1$:

$$\begin{aligned} P(t) &= \mathcal{N}(!t)(1 + \frac{t^2}{\nu})^{-\frac{\nu+1}{2}} && \text{(C.249)} \\ &\approx \mathcal{N}(!t)e^{-\frac{t^2}{2}\frac{\nu+1}{\nu}} && \text{(C.250)} \\ &\approx \mathcal{N}(t; \mu = 0, \sigma^2 = 1) \ . && \text{(C.251)} \end{aligned}$$

Partial derivation of the explicit form of $\text{Stud}(t)$**.**

Note that the z definition Eq.(C.245) and the t definition Eq.(C.247), imply that

$$t = z \underbrace{\sqrt{\frac{\sigma^2}{\widehat{\sigma^2}}}}_{\underline{\varrho}} \ , \tag{C.252}$$

In the expression $\underline{t} = \underline{z}\underline{\varrho}$, the random variables $\underline{z}$ and $\underline{\varrho}$ are independent because, as shown in Section C.35, $\widehat{\mu}$ and $\widehat{\sigma^2}$ are independent. Therefore, the random variable $\underline{t}$ can be defined using the bnet of Fig.C.8.

Figure C.8: Bnet used to define the Student's t-distribution.

The TPMs, printed in blue, for the bnet Fig.C.8, are as follows:

$$P(t|z, \varrho) = \delta(t - z\varrho) \quad \text{(Dirac delta function)} \tag{C.253}$$

$$P(z) = \mathcal{N}(z; \mu = 0, \sigma^2 = 1) \tag{C.254}$$

$$P(\varrho) = \text{given by Eq.(C.267) below.} \tag{C.255}$$

Note that

$$\begin{aligned} P(\underline{t} = t) &= P(\underline{z}\underline{\varrho} = t) && \text{(C.256)} \\ &= \int d\varrho \; P(\underline{z} = \frac{t}{\varrho}|\varrho) P(\varrho) && \text{(C.257)} \\ &= \int d\varrho \; \mathcal{N}(\frac{t}{\varrho}; 0, 1) P(\varrho) && \text{(C.258)} \\ &= \int d\varrho \; \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t}{\varrho})^2} P(\varrho) \; . && \text{(C.259)} \end{aligned}$$

If we define q by

$$q = \frac{n-1}{\varrho^2} \; , \tag{C.260}$$

then

$$q = \frac{(n-1)\widehat{\sigma^2}}{\sigma^2} = \frac{1}{\sigma^2} \sum_{\sigma=0}^{n-1} (x^\sigma - \widehat{\mu})^2 \; . \tag{C.261}$$

As a consequence of "Cochran's Theorem" (see Ref.[123]), $\underline{q}$ given by Eq.(C.261) must have a Chi-square probability distribution with $\nu = n - 1$ degrees of freedom:[12]

$$P(q) = \chi^2(q; \nu = n - 1) \tag{C.262}$$

Henceforth, let $\nu = n - 1$. From the definition Eq.(C.260) of q, we get

$$dq = \frac{-2\nu}{\varrho^3} d\varrho \; . \tag{C.263}$$

Therefore,

[12]Note that this q is a quadratic form $q = \vec{x}^T M \vec{x}$, where $\vec{x}$ is an n dimensional column vector with components x^σ, and M is an $n \times n$ matrix. Cochran's Theorem diagonalizes M and replaces the vectors $\vec{x}$ by equivalent ones in a new basis. Then the number of DOFs (degrees of freedom) of the chi-square distribution is the number of non-zero diagonal elements in the diagonalized M (this number is called the rank of M). In the particular case of Eq.(C.261), $DOF = n - 1$.

$$
\begin{aligned}
P(\varrho)d\varrho &= P(q)dq && \text{(C.264)}\\
&= \chi^2(\frac{\nu}{\varrho^2};\nu)\frac{(-2\nu)}{\varrho^3}d\varrho && \text{(C.265)}\\
&= \mathcal{N}(!\varrho)\left(\frac{\nu}{\varrho^2}\right)^{\frac{\nu}{2}-1} e^{-\frac{\nu}{2\varrho^2}}\frac{d\varrho}{\varrho^3} && \text{(C.266)}\\
&= \mathcal{N}(!\varrho)\frac{d\varrho}{\varrho^{\nu+1}}e^{-\frac{\nu}{2\varrho^2}} \ . && \text{(C.267)}
\end{aligned}
$$

Hence,

$$
\begin{aligned}
P(t) &= \mathcal{N}(!t)\int_0^\infty \frac{d\varrho}{\varrho^{\nu+1}}e^{-\frac{\nu}{2\varrho^2}}e^{-\frac{1}{2}(\frac{t}{\varrho})^2} && \text{(C.268)}\\
&= \mathcal{N}(!t)\int_0^\infty \frac{d\varrho}{\varrho^{\nu+1}}e^{-\frac{1}{2}\frac{t^2+\nu}{\varrho^2}} \ . && \text{(C.269)}
\end{aligned}
$$

C.38 Hypothesis testing and 3 classic test statistics (Likelihood, Score, Wald)

Suppose we have data $\vec{x} = [x^\sigma]_{\sigma=0,1,\ldots,nsam-1}$ which is distributed according to a probability distribution $P(\vec{x};\theta)$ which depends on some parameters $\theta = (\theta_i)_{i=0,1,\ldots,n-1} \in \mathbb{R}^n$. We define the

- **Likelihood of** θ by

$$L(\theta) = P(\vec{x}|\theta) \ , \tag{C.270}$$

- the **Maximum Likelihood estimate of** θ by

$$\widehat{\theta} = \operatorname*{argmax}_\theta L(\theta) \ , \tag{C.271}$$

- the **log likelihood of** θ by

$$LL(\theta) = \ln L(\theta) \ , \tag{C.272}$$

- the **Score** or **Lagrange Multiplier vector**[13] of θ by

[13] $sc_i(\theta) = \partial_{\theta_i} LL(\theta)$ is called a Lagrange multiplier because to maximize $LL(\theta)$ over θ subject to the constraint $\theta = \theta^*$, we maximize the Lagrangian $\mathcal{L} = LL(\theta) - \sum_i \alpha_i[\theta_i - \theta_i^*]$ over (θ_i, α_i) for all i. The latter gives $\alpha_i = \partial_{\theta_i} LL(\theta^*)$ and $\theta = \theta^*$. For a general constrained optimization problem, $\mathcal{L} = LL(\theta) - \sum_i \alpha_i c_i(\theta)$ for some constraint functions $c_i(\theta)$. Hence, $\alpha_i = \partial_{\theta_i} LL/\partial_{\theta_i} c_i$ and $c_i(\theta) = 0$ for all i. So, in general, a Lagrange multiplier is a ratio of two partial derivatives.

$$sc(\theta) = [sc_i(\theta)] \in \mathbb{R}^n \ , \tag{C.273}$$

where

$$sc_i(\theta) = \partial_{\theta_i} LL(\theta) \ , \tag{C.274}$$

- and the **Fisher Information Matrix of** θ by

$$FI(\theta) = [FI_{i,j}(\theta)] \in \mathbb{R}^{n \times n} \tag{C.275}$$

where

$$FI_{i,j}(\theta) = -E_{\underline{\vec{x}}|\theta}[\partial_{\theta_i}\partial_{\theta_j} \overbrace{\ln P(\underline{\vec{x}}|\theta)}^{LL(\theta)}] \ . \tag{C.276}$$

Note that if

$$LL(\theta) = \ln P(x|\theta) = \frac{-(\theta - \widehat{\theta})^2}{2\sigma^2} + \mathcal{N}(!\theta) \ , \tag{C.277}$$

then

$$sc(\theta) = \frac{-(\theta - \widehat{\theta})}{\sigma^2} \tag{C.278}$$

and

$$FI(\theta) = \frac{1}{\sigma^2} \tag{C.279}$$

In hypothesis testing, one has a **null hypothesis** H_0: $\theta = \theta^*$, and an **alternative hypothesis** H_1: $\theta \neq \theta^*$. We use a **test statistic** $\lambda(\theta^*, \widehat{\theta}) \geq 0$ which measures a kind of distance or separation between the estimate $\widehat{\theta}$ and the value θ^* for the null Hypothesis H_0. For some **confidence level** $C > 0$, if $\lambda(\theta^*, \widehat{\theta}) > C$, H_0 is **rejected**, whereas if $\lambda(\theta^*, \widehat{\theta}) \leq C$, H_0 is **accepted**.

$\alpha = 1 - C$ is called the **significance level**. Usually $\alpha << 1$ represents the area under both tails of a Normal distribution, and $C \approx 1$ represents all the area except the tails of a Normal distribution.

Henceforth in this section, we will occasionally use the Einstein summation convention; i.e., implicit sum over repeated indices.

Three classic test statistics are (See Fig.C.9):

1. **Likelihood Ratio test statistic** (Ref.[158].)

$$\lambda_{Li} = 2 \ln \left[\frac{L(\widehat{\theta})}{L(\theta^*)} \right] = 2[LL(\widehat{\theta}) - LL(\theta^*)] \tag{C.280}$$

Figure C.9: For $n = 1$ and $\theta \in \mathbb{R}$, this figure shows the geometrical significance of certain quantities that characterize the 3 classic test statistics (Likelihood, Score, Wald) for hypothesis testing.

2. **Score (a.k.a. Lagrange multiplier) test statistic** (Ref.[179].)

$$\lambda_{Sc} = \partial_{\theta_i} LL(\theta^*) \left[FI(\theta^*)^{-1} \right]_{i,j} \partial_{\theta_j} LL(\theta^*) \tag{C.281}$$

$$= \frac{[\partial_\theta LL(\theta^*)]^2}{FI(\theta^*)} \quad \text{if } n = 1 \tag{C.282}$$

Doesn't depend on $\widehat{\theta}$.

3. **Wald test statistic** (Ref.[196].)

$$\lambda_{Wa} = (\widehat{\theta} - \theta^*)_i \left[\left\langle \underline{\widehat{\theta}}, \underline{\widehat{\theta}}^T \right\rangle^{-1} \right]_{i,j} (\widehat{\theta} - \theta^*)_j \tag{C.283}$$

$$= \frac{(\theta^* - \widehat{\theta})^2}{\left\langle \widehat{\theta}, \widehat{\theta} \right\rangle} \quad \text{if } n = 1 \tag{C.284}$$

More generally, one can replace $\theta^* \to R\theta^*$ and $\widehat{\theta} \to R\widehat{\theta}$ in Eq.(C.283), where θ^* and $\widehat{\theta}$ are n dimensional column vectors, and $R \in \mathbb{R}^{\nu \times n}$. The null and alternative hypotheses become: $H_0 : R\theta = R\theta^*$ and $H_1 : R\theta \neq R\theta^*$. Note that ν is the number of constraints imposed by the null hypothesis. R is called a reparametrization of θ. The Wald test is not reparametrization invariant (i.e., R invariant), but the Likelihood Ratio test is.

Note that if $LL(\theta)$ is given by Eq.(C.277), then $\left\langle \underline{\widehat{\theta}}, \underline{\widehat{\theta}} \right\rangle = \sigma^2 = \frac{1}{FI(\theta)}$. Hence,

$$\lambda_{Li} = \lambda_{Sc} = \lambda_{Wa} = \frac{(\widehat{\theta} - \theta^*)^2}{\sigma^2} \tag{C.285}$$

Many other commonly used test statistics (or their squares) are special cases of one of the 3 classic test statistics. For example, the z-statistic used with normal distributions, the t-statistic used with the Student t-distribution, the F-statistic used in linear regression, the chi-squared statistic used to do Pearson's chi-squared test.

Asymptotic Behavior

If the data $\vec{x}$ is i.i.d.,

$$P(\vec{x}|\theta) = \prod_{\sigma=0}^{nsam-1} P(x^\sigma|\theta) \tag{C.286}$$

Hence, as $nsam \to \infty$,

$$\begin{aligned} LL(\theta) &= \ln P(\vec{x}|\theta) && \text{(C.287)} \\ &= \sum_\sigma \ln P(x^\sigma|\theta) && \text{(C.288)} \\ &\to nsam \sum_x P(x|\theta) \ln P(x|\theta) && \text{(C.289)} \\ &= -nsam\ H(\underline{x}|\theta) && \text{(C.290)} \end{aligned}$$

Thus, *maximizing* the log likehood $LL(\theta)$ and *minimizing* the entropy $H(\underline{x}|\theta)$ give the same estimate $\widehat{\theta}$.

When the data is i.i.d. and $nsam \to \infty$, it is also possible to prove that the 3 test statistics defined above all tend to the same probability distribution, namely $\mathcal{X}^2(\theta^*;\nu)$, the chi-square distribution with ν degrees of freedom, where $\theta \in \mathbb{R}^n$, $R \in \mathbb{R}^{\nu\times n}$, and $\nu = n$ if $R = 1$.

C.39 Error Bars

Never report measurements without error bars!!

Assume a distribution with mean μ and standard deviation σ for a subpopulation with n samples.

$SE = \frac{\sigma}{\sqrt{n}}$ is called the **standard error**.

Some popular types of error bars:

- **Box and Whiskers plot (a.k.a. Boxplot)**

  See Fig.C.10. IQR stands for **Intermediate Quantile Range**. Sometimes, the endpoints of the error bars are taken to be the minimum and maximum samples instead of $Q_1 - 1.5 * IQR$ and $Q_3 + 1.5 * IQR$. The points that fall in the intervals $[\min, Q_1 - 1.5 * IQR]$ and $[Q_3 + 1.5 * IQR, \max]$ are called **outliers**.

- **Standard Deviation**

Figure C.10: Boxplot plot for Normal distribution $\mathcal{N}(\mu = 0, \sigma)$. Q_1 and Q_3 are the first and third quantiles, and M is the median (i.e., half-way point). For a non-normal skewed distribution, Q_1 and Q_3 are not equidistant from the median, and the median is not exactly equal to the mean.

Error bar endpoints are located one standard deviation away from the mean.

$$\mu - \sigma < \mu < \mu + \sigma \tag{C.291}$$

- **Confidence Interval**

$$\mu - |z^*|SE < \mu < \mu + |z^*|SE \tag{C.292}$$

$|z^*| = 1.96$ for a confidence level of 95%.

The origin of Eq.(C.292) is explained in the next section entitled "Confidence Intervals". Confidence intervals are derived from the Gaussian in Fig.C.11, which should not be confused with the Gaussian of Fig.C.10. They are different!

C.40 Confidence Interval

Normal distribution with mean μ and standard deviation σ:

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \ . \tag{C.293}$$

Standard Normal Distribution (SND):

$$P(z) = \mathcal{N}(z; 0, 1) \tag{C.294}$$

Cumulative distribution for $P(z)$:

$$\Phi(z) = \int_{-\infty}^{z} dz' \; P(z') \; . \tag{C.295}$$

Figure C.11: Interpretation of confidence level C and p-value as areas under curve of the Standard Normal Distribution (SND).

Confidence Level C and corresponding $|z^*|$ **value** (see Fig.C.11):

$$C = \int_{-|z^*|}^{|z^*|} dz \; P(z) = \Phi(|z^*|) - \Phi(-|z^*|) = 2\left(\Phi(|z^*|) - \frac{1}{2}\right) \tag{C.296}$$

Equivalent definition:

$$C = P\left(\underbrace{\frac{|\underline{x} - \mu|}{\frac{\sigma}{\sqrt{n}}}}_{|\underline{z}|} < |z^*|\right) \tag{C.297}$$

For $C = 95\%$, $|z^*| = 1.960 \approx 2$. For $C = 99\%$, $|z^*| = 2.576$.

Area of each tail in Fig.C.11 is usually called α, and the area of both tails is called the **p-value**:

$$C + \underbrace{2\alpha}_{p-value} = 1 \; . \tag{C.298}$$

Estimators[14] of mean μ and standard deviation σ from measurements x^σ of a sub-population Σ_1 of size $n = |\Sigma_1|$:

$$\widehat{\mu} = \overline{x} = \frac{1}{n} \sum_{\sigma \in \Sigma_1} x^\sigma \tag{C.299}$$

$$\widehat{\sigma}^2 = \frac{1}{n-1} \sum_{\sigma \in \Sigma_1} (x^\sigma - \overline{x})^2 \tag{C.300}$$

[14]Don't confuse the sample index σ with the standard deviation σ.

We get from Eq.(C.297), the **Error bars (a.k.a. confidence intervals)** and **Error E (a.k.a. margin of error)**:

$$\text{estimate of } x \text{ with error bars} = \overline{x} \pm \underbrace{|z^*| \frac{\widehat{\sigma}}{\sqrt{n}}}_{E} \tag{C.301}$$

$$n = \left(\frac{|z^*| \widehat{\sigma}}{E} \right)^2 \tag{C.302}$$

So far, we have assumed that the sub-population (a.k.a. sample population) is normally distributed. This might be false for several reasons. Some red flags: (1) n is too small (according to a rule of thumb derived from Central Limit Theorem, n should be larger than 30 to insure a Normal Distribution). (2) Sub-population not truly random (i.i.d.) because was taken without replacement. In many cases, especially when $n < 30$, the Student's t-distribution models the sub-population statistics much better than the Normal distribution.

The Student's t-distribution $\text{Stud}(t; \nu = n - 1)$, depends on a parameter ν called the number of degrees of freedom. In the case being considered here, ν equals the sub-population size n minus one. When fitting the data with Stud(), variable t replaces variable z, and $\text{Stud}(t; \nu = n - 1)$ replaces the Standard Normal distribution (SND) $\mathcal{N}(z; \mu = 0, \sigma = 1)$. Stud() is symmetric about the origin like SND, but its tails are fatter. When fitting the data with Stud(), the $|z^*|$ value is replaced by a $|t^*|$ value. Eq.(C.296) is replaced by

$$C = \int_{-|t^*|}^{|t^*|} dt \, \text{Stud}(t) = \Phi_S(|t^*|) - \Phi_S(-|t^*|) = 2 \left(\Phi_S(|t^*|) - \frac{1}{2} \right) , \tag{C.303}$$

where $\Phi_S()$ is the cumulative distribution for Stud(). Also, Eq.(C.301) is replaced by

$$\text{estimate of } x \text{ with error bars} = \overline{x} \pm \underbrace{|t^*| \frac{\widehat{\sigma}}{\sqrt{n}}}_{E} . \tag{C.304}$$

Tables of $|t^*|(C, \nu = n - 1)$ are available. Note that $|t^*|$ depends on both C and ν, whereas $|z^*|(C)$ depends only on C.

C.41 Score p-value

When defining error bars and confidence intervals in Fig.C.11, we defined a triplet of values (z, C, p) where $C + p = 1$. In this section, we will consider a different triplet of those values. We will refer to the triplet (z_{th}, C_{th}, p_{th}) used in error bars as the **threshold triplet**, and to the triplet (z_{sc}, C_{sc}, p_{sc}) introduced in this section as the **score triplet**. When we do hypothesis testing, if $|z_{sc}| > |z_{th}|$ (i.e., if z_{sc} falls outside

the error bars), we say that the null hypothesis is violated. Equivalently, we say the null hypothesis is violated if $p_{sc} < p_{th} = 1 - C_{th} = 0.05$ typically. Most statistics books do a poor job at distinguishing between the threshold and score triplets, and seldom use distinguishing subscripts like th and sc. In this book, we will often drop the sc subscripts, but we will try not to drop the th subscripts. Often, instead of a th subscript, we will use an asterisk superscript. For instance, instead of z_{th}, we might use z^*. When statisticians use the term "p-value", they are usually referring to the score p-value, although not always.

Given a parameter θ, call $\theta = \theta_0$ (or $\theta < \theta_0$ or $\theta > \theta_0$) the **null hypothesis** h_0, and call the negation of h_0 (i.e., $\theta \neq \theta_0$ (or $\theta \geq \theta_0$ or $\theta \leq \theta_0$)) the **alternative or opposite hypothesis** h_1. Assume we are given data $\vec{x} = \{x^\sigma | \sigma \in \Sigma\}$. Assume also that we are given distributions $P(\underline{x} = x|h)$ for $h \in \{h_0, h_1\}$, and $P(\underline{x} = x)$. Now let

$$P(\vec{x}|h) = \prod_\sigma P(\underline{x} = x^\sigma|h) \tag{C.305}$$

$$P(\vec{x}) = \prod_\sigma P(\underline{x} = x^\sigma) \tag{C.306}$$

(so the x^σ are i.i.d.).

A Bayesian would assume that there is a prior $P(h)$, and use it to calculate $P(h|\vec{x}) = \frac{P(\vec{x}|h)P(h)}{P(\vec{x})}$. $P(\underline{h} = h_0|\vec{x})$ is the probability that the null hypothesis is true. A p-value is a monotonically increasing function of $P(\underline{h} = h_0|\vec{x})$, so Bayesians have no trouble saying that a p-value is a measure of $P(\underline{h} = h_0|\vec{x})$, i.e., a measure of the probability that the null-hypothesis is true.

Frequentists, on the other hand, believe that h is a "parameter" which has an priori value; therefore, it's not a random variable, so $P(\underline{h} = h_0|\vec{x})$ is undefined. To circumvent this objection, a frequentist would conduct a bunch of experiments to decide whether h equals h_0 or h_1. Then he/she would say the p-value is the fraction of those experiments that claim $h = h_0$.

Next, we explain in more detail the correct way of thinking about p-values, according to Frequentists. p-values were invented by Frequentists, so it's worth hearing what they have to say about them. The Frequentist definition is not against Bayesianism, and Bayesians, unlike Frequentists, don't accuse Frequentists of having a sinfully incorrect definition of p-values. A Bayesian would just say: our definition of p-values (shown in red above) is not incorrect, but the Frequentist definition is more precise than ours, and doesn't assume a particular form for a prior. We welcome it.

Call the random variable $\underline{t}$ the **test or score statistic** and let t^* be a user defined parameter. $\underline{t}$ and t^* are defined so that when $\underline{t} = t^*$, the h_0 hypothesis is on the threshold between being and not being satisfied. Frequentists define the **p-value** p as

$$p = \begin{cases} P(\underline{t} \geq t^* | h_0) & \text{right-sided-tail, if } h_0 \text{ is } \theta < \theta_0 \\ P(\underline{t} \leq t^* | h_0) & \text{left-sided-tail, if } h_0 \text{ is } \theta > \theta_0 \\ P(|\underline{t}| > |t^*| \; | h_0) & \text{double-sided-tail, if } h_0 \text{ is } \theta = \theta_0 \end{cases} \tag{C.307}$$

Thus, for a Frequentist, a p-value is a probabilistic weight of the region where the h_0 hypothesis is defined (by the user) to be violated. If that weight is large, then the region where the h_0 hypothesis is defined to be satisfied is small, which means the h_0 hypothesis is expected to be close to the truth. The larger the p-value, the closer h_0 is expected to be near the truth, just like the Bayesian definition says. Note that the p-value is a probability so it ranges in value from 0 to 1.

Suppose we are given a sub-population with n samples, mean $\overline{x}$ and variance $\widehat{\sigma}$. Let $\theta_0 = \mu_0$. Define

$$\underline{t} = \underline{z} = \frac{\underline{x} - \mu_0}{\frac{\widehat{\sigma}}{\sqrt{n}}} \; . \tag{C.308}$$

For $n > 30$,

$$P(\underline{z} \geq z^* | h_0) = 1 - \Phi(z^*) = \Phi(-z^*) \qquad \text{if } h_0 \text{ is } \mu < \mu_0 \tag{C.309a}$$

$$P(\underline{z} \leq z^* | h_0) = \Phi(z^*) \qquad \text{if } h_0 \text{ is } \mu > \mu_0 \tag{C.309b}$$

$$P(|\underline{z}| \geq |z^*| \; | h_0) = 2\Phi(-|z^*|) \qquad \text{if } h_0 \text{ is } \mu = \mu_0 \tag{C.309c}$$

where $\Phi(x)$ is the cumulative distribution for the Standard Normal Distribution $\mathcal{N}(x; \mu = 0, \sigma = 0)$. For $n < 30$, $\Phi()$ is replaced by $\Phi_S()$, where $\Phi_S()$ is the cumulative distribution for the Student t-distribution $\text{Stud}(x; \nu = n - 1)$. Note that Eq.(C.309c) agrees with Eq.(C.297).

The quantity

$$\hat{z}_{sc} = \frac{\overline{x} - \mu_0}{\frac{\widehat{\sigma}}{\sqrt{n}}} \tag{C.310}$$

is called the **z score estimator**. If $|\hat{z}_{sc}| > |z^*|$, then the h_0 hypothesis is defined to be violated (for the double sided case).

C.42 Convex/Concave functions, Jensen's Inequality

Suppose $f : \mathbb{R} \to \mathbb{R}$. $f(x)$ is a **concave function** if looks like a cave ($\cap$) (i.e., $f''(x) > 0$ if differentiable) and it's a **convex function** if it looks like a valley ($\cup$) (i.e., $f''(x) < 0$ if differentiable). More generally, if $f : \mathbb{R}^a \to \mathbb{R}$, $f(x)$ is said to be concave if $f(\alpha x + \beta y) \geq \alpha f(x) + \beta f(y)$ and convex if $f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y)$ for $\alpha, \beta \in \mathbb{R}$ and $x, y \in \mathbb{R}^a$.

Figure C.12: Jensen's inequality for sum of 2 terms, when $f : \mathbb{R} \to \mathbb{R}$ is convex.

Suppose $f : \mathbb{R}^a \to \mathbb{R}$ is a convex function. Let α, β be non-negative numbers that sum to 1, and $x, y \in \mathbb{R}^a$. From the definition of convexity, it follows that (see Fig.C.12 for a geometrical representation of this when $a = 1$)

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y) \tag{C.311}$$

Jensen's inequality is a simple generalization of this inequality to sums of more than 2 terms. Let $\{p_i\}_{i=0}^n$ be non-negative numbers that sum to one. Also assume that $\{x_i\}_{i=0}^n$ are elements of $\mathbb{R}^a$. Then

$$f\left(\sum_{=1}^{n} p_i x_i\right) \leq \sum_{=1}^{n} p_i f(x_i) \tag{C.312}$$

or, written in terms of expected values,

$$f(E[\underline{x}]) \leq E[f(\underline{x})] \tag{C.313}$$

The same result is true if f is concave instead of convex and we reverse the inequality signs.

C.43 Chebyshev's inequality

Chebyshev's inequality (CI) gives an upper bound for the area under the 2 tails (left and right ones) of a probability distribution.

Below, we follow the common practice of proving CI as a corollary of Markov's Inequality (MI).

Claim 16 *(Markov's Inequality) Let $\underline{x}$ be a non-negative random variable and let $a > 0$. Then*

$$P(\underline{x} \geq a) \leq \frac{E[\underline{x}]}{a} \tag{C.314}$$

proof:

$$\frac{E[\underline{x}]}{a} = \frac{\int_0^\infty dx\ xP(x)}{a} \tag{C.315}$$
$$\geq \frac{\int_a^\infty dx\ xP(x)}{a} \tag{C.316}$$
$$\geq \frac{\int_a^\infty dx\ aP(x)}{a} \tag{C.317}$$
$$= P(\underline{x} \geq a) \tag{C.318}$$

QED

Claim 17 *(Chebyshev's inequality) Let $\underline{x}$ be a random variable with mean μ and variance σ^2. Then for any real number $k > 0$,*

$$P(|\underline{x} - \mu| \geq k\sigma) \leq \frac{1}{k^2} \tag{C.319}$$

proof:

$$P(|\underline{x} - \mu| \geq k\sigma) = P(|\underline{x} - \mu|^2 \geq k^2\sigma^2) \tag{C.320}$$
$$\leq \frac{\sigma^2}{k^2\sigma^2} \quad \text{(by MI)} \tag{C.321}$$
$$= \frac{1}{k^2} \tag{C.322}$$

QED

Figure C.13: Pictorial representation of Chebyshev's Inequality (CI). Markov's Inequality (MI) has a similar representation, but k is proportional to $\sqrt{a}$, so the upper bound for MI goes as $1/a$ instead of $1/k^2$.

See Fig.C.13 for a pictorial representation of CI. Note that the CI approximation is always a bad approximation for small k, and might not become good

even for large k. For example, if the $\underline{x}$ distribution is a box centered at μ, then $P(|\underline{x}-\mu| \geq k\sigma) = 0$ for $k\sigma$ larger than the width of the box, so $1/k^2$ is a terrible approximation for large k too.

A (1 dimensional) Center of Mass (CM) for a unit mass object is an expectation value where the probability distribution is a mass distribution. Hence, it's not too surprising that MI can be interpreted in terms of CMs.

Physics Intuition for MI in terms of CMs:

$$\underbrace{\int_0^\infty dx\, xP(x)}_{E[\underline{x}]=CM} = \underbrace{\frac{\int_0^a dx\, xP(x)}{P(\underline{x}<a)}}_{E_{\underline{x}<a}[\underline{x}]=CM^-} \underbrace{P(\underline{x}<a)}_{f^-} + \underbrace{\frac{\int_a^\infty dx\, xP(x)}{P(\underline{x}>a)}}_{E_{\underline{x}>a}[\underline{x}]=CM^+} \underbrace{P(\underline{x}>a)}_{f^+} \tag{C.323}$$

$$\begin{aligned} CM\underbrace{(f^+ + f^-)}_{=1} &= CM^- f^- + CM^+ f^+ && \text{(C.324)} \\ &\geq CM^+ f^+ && \text{(C.325)} \\ &\geq a f^+ \quad (\text{because } CM^+ \geq a) && \text{(C.326)} \end{aligned}$$

You can think of $CM^+ f^+$ as a torque with moment arm= CM^+ and force=f^+. In this picture, MI is the approximation of a torque with moment arm=CM and force=1, by a smaller torque with moment arm= any real $a > 0$ and force=$f^+ \leq 1$.

C.44 Short Summary of Boolean Algebra

See Ref.[117] for more info about this topic.

Suppose $x, y, z \in \{0, 1\}$. Define

$$x \text{ or } y = x \vee y = x + y - xy \ , \tag{C.327}$$

$$x \text{ and } y = x \wedge y = xy \ , \tag{C.328}$$

and

$$\text{not } x = \overline{x} = 1 - x \ , \tag{C.329}$$

where we are using normal addition and multiplication on the right hand sides.[15]

Actually, since $x \wedge y = xy$, we can omit writing the symbol $\wedge$. The symbol $\wedge$ is useful to exhibit the symmetry of the identities, and to remark about the analogous identities for sets, where $\wedge$ becomes intersection $\cap$ and $\vee$ becomes union $\cup$. However, for practical calculations, $\wedge$ is an unnecessary nuisance.

[15] Note the difference between $\vee$ and modulus 2 addition $\oplus$. For $\oplus$ (a.k.a. XOR): $x \oplus y = x + y - 2xy$.

Associativity	$x \vee (y \vee z) = (x \vee y) \vee z$ $x \wedge (y \wedge z) = (x \wedge y) \wedge z$
Commutativity	$x \vee y = y \vee x$ $x \wedge y = y \wedge x$
Distributivity	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
Identity	$x \vee 0 = x$ $x \wedge 1 = x$
Annihilator	$x \wedge 0 = 0$ $x \vee 1 = 1$
Idempotence	$x \vee x = x$ $x \wedge x = x$
Absorption	$x \wedge (x \vee y) = x$ $x \vee (x \wedge y) = x$
Complementation	$x \wedge \overline{x} = 0$ $x \vee \overline{x} = 1$
Double negation	$\overline{(\overline{x})} = x$
De Morgan Laws	$\overline{x} \wedge \overline{y} = \overline{(x \vee y)}$ $\overline{x} \vee \overline{y} = \overline{(x \wedge y)}$

Table C.1: Boolean Algebra Identities

Since $x \in \{0, 1\}$,

$$P(\overline{x}) = 1 - P(x) \ . \tag{C.330}$$

Clearly, from analyzing the simple event space $(x, y) \in \{0, 1\}^2$,

$$P(x \vee y) = P(x) + P(y) - P(x \wedge y) \ . \tag{C.331}$$

C.45 Laplace transform

This section is a watered down version of the Wikipedia entry for Laplace Transforms (Ref.[155]), which we highly recommend.

Let $0^- = 0 - \epsilon$, $0^+ = 0 + \epsilon$ for some $\epsilon \in \mathbb{R}$ such that $0 < \epsilon << 1$.

Let $s = \sigma + i\omega$ for $\sigma, \omega \in \mathbb{R}$. σ is called the **decay constant** and ω is called the **angular frequency**.

The **Laplace Transform (LT)** of $f : [0, \infty] \to \mathbb{C}$ is defined as

$$\mathcal{L}[f](s) = \widetilde{f}(s) = \int_{0-}^{\infty} dt \ e^{-st} f(t) \tag{C.332}$$

Note that the LT is a linear functional[16] because

[16] A functional $\mathcal{F}[f]$ is a function of a function f, or, equivalently, a function of a vector with possibly infinitely many components given by $[f(x)]_{\forall x}$.

$$\mathcal{L}[af + bg](t) = a\mathcal{L}[f](t) + b\mathcal{L}[g](t) \tag{C.333}$$

for $f, g : [0, \infty] \to \mathbb{C}$ and $a, b \in \mathbb{C}$.

The **Inverse Laplace Transform** is defined so that

$$\mathcal{L}^{-1}[\underbrace{\mathcal{L}[f]}_{\widetilde{f}}](t) = f(t) \tag{C.334}$$

For LTs, we assume functions $f(t)$ that vanish for $t < 0^-$. They can jump to a finite value (with a step function) or an infinite value (with a Dirac delta function) at $t = 0$, but must vanish for $t < 0^-$. LTs are ideally suited for solving ordinary differential equations with **initial conditions** such as $x(0) = 5$, $\partial_t x(0) = 10$.

name	formula	comment
Bilateral Laplace transform (BLT)	$\mathcal{B}[f](s) = \int_{-\infty}^{\infty} dt\ e^{-st} f(t)$	same as LT but with $-\infty < t < \infty$ instead of $t > 0$
Fourier transform (FT)	$\mathcal{F}[f](\omega) = \int_{-\infty}^{\infty} dt\ e^{-i\omega t} f(t)$	Same as BLT but with $s = i\omega \in i\mathbb{R}$
Star Transform (ST)	$\mathcal{L}^*[f](s) = \sum_{n=0}^{\infty} e^{-snT} f(nT)$	Same as LT but it samples only discrete points at $t = nT$
Moment Generating Function	$E_{\underline{x}}[e^{-s\underline{x}}] = \int_0^{\infty} dx\ e^{-sx} P(x)$ $\langle \underline{x}^n \rangle = \left[(-\partial_{\underline{s}})^n E_{\underline{x}}[e^{-s\underline{x}}]\right]_{s=0}$	Same as LT but for probability distribution $P : [0, \infty] \to [0, 1]$

Table C.2: Transforms that are akin to the Laplace transform. Don't be intimidated by all these transforms. They are all just fancy dot products like $\vec{a} \cdot \vec{b}$.

Table C.2 is a table of transforms that are akin to the LT. If the function $f(t)$ does not vanish for $t < 0$, we can use the **Bilateral Laplace Transform (BLT)**. The BLT becomes the **Fourier transform (FT)** when $s = i\omega$. In this section, we will only discuss the LT.

The following intuition about LTs might be helpful to the reader. A LT is like a dot product of two vectors, e^{-st} and $f(t)$, except that in this case the index t for their components is an uncountable set $[0, \infty)$. As with all dot products, its maximum is achieved if the two vectors point in the same direction (this is what the Cauchy Schwartz inequality $\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos\theta \leq |\vec{a}||\vec{b}|$ says). In this case, if we substitute $f(t) = \widetilde{f}(s_0) e^{s_0 t}$ on the right hand side of Eq.(C.332), we get

$$\widetilde{f}(s) = \widetilde{f}(s_0) \int_{0^-}^{\infty} dt\, e^{-(s-s_0)t} \tag{C.335}$$

$$= \widetilde{f}(s_0)\delta_+(s - s_0) \tag{C.336}$$

So our intuition is this: whenever you see an equation involving LTs, replace each $f(t)$ by the special case $f(t) = e^{s_o t}\widetilde{f}$ (this is called a **phasor** when $s_0 = i\omega_0$), and convince yourself that the equation is valid in the special case of phasors.

Define the **Dirac delta function** by

$$\delta(t) = \int_{-\infty}^{\infty} d\omega\, e^{i\omega t} = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases} \tag{C.337}$$

and the **Heaviside step function** by

$$\mathbf{H}_a(t) = \mathbb{1}(t - a > 0) \tag{C.338}$$

Next, we will list some examples and properties of LTs. Henceforth, we will use the following notation:

$a, b \in \mathbb{C}$
$f, g : [0, \infty] \to \mathbb{C}$
$f(t) \xrightarrow[\mathcal{L}]{} \widetilde{f}(s)$ means $\mathcal{L}[f](s) = \widetilde{f}(s)$

C.45.1 Examples

- Dirac delta function

$$\delta(t - a) \xrightarrow[\mathcal{L}]{} e^{-sa} \tag{C.339}$$

- Heaviside step function

$$\mathbf{H}_a(t) \xrightarrow[\mathcal{L}]{} \frac{1}{s}e^{-sa} \quad (\text{for } Re(s) > 0,\ a > 0) \tag{C.340}$$

- box

$$\mathbf{H}_0(t) - \mathbf{H}_a(t) \xrightarrow[\mathcal{L}]{} \frac{1}{s}(1 - e^{-sa}) \quad (\text{for } Re(s) > 0,\ a > 0) \tag{C.341}$$

- ramp

$$t\, \mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{1}{s^2} \quad (\text{for } Re(s) > 0) \tag{C.342}$$

- curved ramp

$$\frac{t^n}{n!}\mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{1}{s^{n+1}} \quad (\text{for } Re(s) > 0 \text{ and } n \geq 0) \tag{C.343}$$

- sine, cosine

$$\sin(at)\mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{a}{s^2+a^2} \tag{C.344}$$

$$\cos(at)\mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{s}{s^2+a^2} \tag{C.345}$$

- polynomial rise, exponential drop

$$\frac{t^n}{n!}e^{-at}\,\mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{1}{(s+a)^{n+1}} \quad (\text{for } Re(s) > -a) \tag{C.346}$$

- Exponential approach to steady state

$$(1-e^{-at})\mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{a}{s(s+a)} \quad (\text{for } Re(s) > 0, Re(s) > -a) \tag{C.347}$$

C.45.2 Properties

- Taylor series of $f(t)$

$$\int_0^\infty dx\; e^{-x}\frac{x^n}{n!} = 1 \tag{C.348}$$

$$\int_0^\infty dt\; e^{-st}\frac{t^n}{n!} = \frac{1}{s^{n+1}} \tag{C.349}$$

$$\frac{t^n}{n!}\mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{1}{s^{n+1}} \tag{C.350}$$

$$f(t)\mathbf{H}_0(t) = \mathbf{H}_0(t)\sum_{n=0}^{\infty}\frac{t^n}{n!}\partial_t^n f(0) \tag{C.351}$$

$$\xrightarrow[\mathcal{L}]{} \sum_{n=0}^{\infty}\frac{1}{s^{n+1}}\partial_t^n f(0) \tag{C.352}$$

- derivatives of $\widetilde{f}(s)$

$$(-t)f(t) \xrightarrow[\mathcal{L}]{} \partial_s \widetilde{f}(s) \tag{C.353}$$

$$(-t)^k f(t) \xrightarrow[\mathcal{L}]{} (\partial_s)^k \widetilde{f}(s) \tag{C.354}$$

- derivatives of $f(t)$

  Define

$$f^{\geq 1}(t) = f(t) - f(0^+)\mathbf{H}_0(t) \tag{C.355}$$

$$f^{\geq 2}(t) = f(t) - f(0^+)\mathbf{H}_0(t) - tf'(0^+)\mathbf{H}_0(t) \tag{C.356}$$

$$\partial_t f(t) \xrightarrow[\mathcal{L}]{} s\widetilde{f^{\geq 1}}(s) \tag{C.357}$$

$$= s\widetilde{f}(s) - f(0^+) \quad \text{(a.k.a. } f(t) \textbf{ differentiator)} \tag{C.358}$$

  For example, for $f(t) = \frac{t^n}{n!}\mathbf{H}_0(t)$, we have $\widetilde{f}(s) = \frac{1}{s^{n+1}}$, so Eq.(C.358) becomes

$$\frac{t^{n-1}}{(n-1)!} \xrightarrow[\mathcal{L}]{} \frac{1}{s^n} \tag{C.359}$$

$$(\partial_t)^2 f(t) \xrightarrow[\mathcal{L}]{} s^2\widetilde{f^{\geq 2}}(s) \tag{C.360}$$

$$= s^2\widetilde{f}(s) - sf'(0^+) - f(0^+) \tag{C.361}$$

- integral of $f(t)$ (a.k.a. $f(t)$ **integrator**)

$$\int_0^t d\tau\ f(\tau) \xrightarrow[\mathcal{L}]{} \frac{1}{s}\widetilde{f}(s) \tag{C.362}$$

  For example, for $f(t) = \frac{t^n}{n!}\mathbf{H}_0(t)$, we have $\widetilde{f}(s) = \frac{1}{s^{n+1}}$, so Eq.(C.362) becomes

$$\frac{t^{n+1}}{(n+1)!}\mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{1}{s^{n+2}} \tag{C.363}$$

  Note that

$$\int_0^t d\tau\ f(\tau) = \int_{-\infty}^{\infty} d\tau\ f(\tau)\mathbf{H}_0(\tau)\mathbf{H}_0(t-\tau) \tag{C.364}$$

$$= ((f\mathbf{H}_0) \circledast \mathbf{H}_0)(t) \tag{C.365}$$

- integral of $\widetilde{f}(s)$

$$\frac{1}{t}f(t) \xrightarrow[\mathcal{L}]{} \int_s^\infty d\sigma\ \widetilde{f}(\sigma) \tag{C.366}$$

- shifting $\widetilde{f}(s)$ (frequency shifting)

$$e^{at}f(t) \xrightarrow[\mathcal{L}]{} \widetilde{f}(s-a) \quad (\text{for } a>0) \tag{C.367}$$

- shifting $f(t)$ (time shifting).

$$f(t-a)\,\mathbf{H}_a(t) \xrightarrow[\mathcal{L}]{} e^{-as}\widetilde{f}(s) \quad (\text{for } a>0) \tag{C.368}$$

- time scaling

$$f(at) \xrightarrow[\mathcal{L}]{} \frac{1}{a}\widetilde{f}\left(\frac{s}{a}\right) \quad (\text{for } a>0) \tag{C.369}$$

- multiplication

$$f(t)g(t) \xrightarrow[\mathcal{L}]{} \frac{1}{2\pi i}\lim_{T\to\infty}\int_{c-iT}^{c+iT} ds'\ \widetilde{f}(s')\widetilde{g}(s-s') \tag{C.370}$$

- convolution

  The **convolution** of $f:\mathbb{R}\to\mathbb{C}$ and $g:\mathbb{R}\to\mathbb{C}$ is defined by

$$(f\circledast g)(t) = \int_{-\infty}^{\infty} d\tau\ f(\tau)g(t-\tau) \tag{C.371}$$

  If $f(t)=g(t)=0$ for $t<0$,

$$(f\circledast g)(t) = \int_0^t d\tau\ f(\tau)g(t-\tau) \quad (\text{see Fig.C.14.}) \tag{C.372}$$

  It's not hard to show that

$$f\circledast g = g\circledast f \tag{C.373}$$

  and that

$$(f\circledast g)(t) \xrightarrow[\mathcal{L}]{} \widetilde{f}(s)\widetilde{g}(s) \tag{C.374}$$

  Eq.(C.374) is easy to check with phasors. Indeed, if we substitute $f(\tau)=e^{i\omega_0\tau}\widetilde{f}$ and $g(t-\tau)=e^{i\omega_0(t-\tau)}\widetilde{g}$, on the right hand side of Eq.(C.372), the right hand side becomes $e^{i\omega_0 t}\widetilde{f}\widetilde{g}$, and the LT of that is $\widetilde{f}\widetilde{g}$.

Figure C.14: Pictorial representation of the convolution $(f \circledast g)(t)$.

A common question is how does one evaluate convolutions in practice. If one can sample and remember the waveforms $f(\tau)$ and $g(\tau)$ for all $\tau \in [0, t]$, then it's just a matter of multiplication and addition of samples. Sometimes, even if we have no memory resources, it's possible to calculate a convolution. For example, if $g(t) = e^{st}\mathbf{H}_0(t)$

$$(f \circledast g)(t) = \int_0^t d\tau\ f(\tau)e^{s(t-\tau)} \tag{C.375}$$

$$= \underbrace{e^{st}}_{g(t)} \widetilde{f}(s) \tag{C.376}$$

so convolving this $g(\cdot)$ merely evaluates it at t and multiplies it by a constant $\widetilde{f}(s)$.

- circular convolution

  If f_T, g_T are periodic functions with period T, their **circular convolution** is defined as

$$(f_T \circledast_C g_T)(t) = \int_0^T d\tau\ f_T(\tau)g_T(t-\tau) \tag{C.377}$$

  One can show that

$$(f_T \circledast_C g_T)(t) \xrightarrow[\mathcal{L}]{} \widetilde{f_T}(s)\widetilde{g_T}(s) \tag{C.378}$$

- complex conjugation

$$f^*(t) \xrightarrow[\mathcal{L}]{} \widetilde{f}^*(s^*) \tag{C.379}$$

- cross correlation

  The **cross correlation** of functions $f, g : [0, \infty] \to \mathbb{C}$ is defined as

$$(f, g)_{CC} = \int_0^\infty d\tau \; f^*(\tau) g(t + \tau) \tag{C.380}$$

  One can show that

$$(f, g)_{CC} \xrightarrow[\mathcal{L}]{} \widetilde{f}^*(-s^*)\widetilde{g}(s) \tag{C.381}$$

- LT of periodic function

  If $f_T(t)$ is a periodic function with period T, then

$$f_T(t)\mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{1}{1 - e^{-sT}} \int_0^T dt \; e^{-st} f_T(t) \tag{C.382}$$

  To show this, define

$$\mathcal{I}_a^b = \int_a^b dt \; e^{-st} f_T(t) \tag{C.383}$$

  Then

$$\begin{aligned} \widetilde{f_T}(s) &= \mathcal{I}_0^T + \mathcal{I}_T^{2T} + \mathcal{I}_{2T}^{3T} + \cdots && \text{(C.384)} \\ &= \mathcal{I}_0^T (1 + e^{-sT} + e^{-s2T} + \cdots) && \text{(C.385)} \\ &= \frac{1}{1 - e^{-sT}} \mathcal{I}_0^T && \text{(C.386)} \end{aligned}$$

- periodic summation

$$\sum_{n=0}^{\infty} f(t - nT)\mathbf{H}_0(t - nT) \xrightarrow[\mathcal{L}]{} \frac{1}{1 - e^{-Ts}} \widetilde{f}(s) \tag{C.387}$$

$$\sum_{n=0}^{\infty} (-1)^n f(t - nT)\mathbf{H}_0(t - nT) \xrightarrow[\mathcal{L}]{} \frac{1}{1 + e^{-Ts}} \widetilde{f}(s) \tag{C.388}$$

- limits of $f(t)$

$$\begin{aligned} \lim_{s \to \infty} s\widetilde{f}(s) &= \lim_{s \to \infty} s \int_0^\infty dt \; e^{-st} f(t) && \text{(C.389)} \\ &\approx f(0^+) \lim_{s \to \infty} \underbrace{s \int_0^\infty dt \; e^{-st}}_{=1} && \text{(C.390)} \\ &= f(0^+) && \text{(C.391)} \end{aligned}$$

$$\lim_{s\to 0} s\widetilde{f}(s) = f(\infty) \quad \text{(a.k.a. steady state)} \tag{C.392}$$

- Inverse LT

  The inverse LT of a function $\widetilde{f}(s)$ can be calculated by performing the following complex contour integral:

$$\underbrace{\mathcal{L}^{-1}[\widetilde{f}(s)](t)}_{f(t)} = \frac{1}{2\pi i} \lim_{T\to\infty} \int_{\gamma-iT}^{\gamma+iT} ds\, e^{st} \widetilde{f}(s) \tag{C.393}$$

  where $\gamma, T \in \mathbb{R}$ and all singularities of $\widetilde{f}(s)$ must be located on the left side of the contour of integration. Another way of calculating the inverse LT of $\widetilde{f}(s)$, is to express $\widetilde{f}(s)$ as a linear combination of functions for which the inverse LT is known from LT tables. For instance,

$$\mathcal{L}^{-1}\left[\frac{1}{s(s-1)}\right] = \mathcal{L}^{-1}\left[\frac{1}{s} - \frac{1}{s+1}\right] \text{(partial fractions expansion)} \tag{C.394}$$

$$= \mathcal{L}^{-1}\left[\frac{1}{s}\right] - \mathcal{L}^{-1}\left[\frac{1}{s+1}\right] \tag{C.395}$$

$$= \mathbf{H}_0(t)[1 - e^{-t}] \tag{C.396}$$

- Bode, Nyquist plots

  Let $s = \sigma + i\omega$.

  Bode plot: plot of

$$(\log_{10}(\omega), \log_{10}|\widetilde{f}(i\omega)|)$$

  and, right below it, plot of

$$(\log_{10}(\omega), phase\{\widetilde{f}(i\omega)\})$$

  .

  Nyquist plot: plot of

$$(Re\widetilde{f}(i\omega), Im\widetilde{f}(i\omega))$$

  or, equivalently, plot of

$$(|\widetilde{f}(i\omega)|, phase(\widetilde{f}(i\omega)))$$

  on polar graph paper.

  Usually, $\widetilde{f}(i\omega)$ is a gain (i.e., LT of output divided by LT of input).

- uncertainty principle

  Here is some "Heisenberg uncertainty principle" type intuition about the relationship between a function $f(t)$ and its LT $\widetilde{f}(s)$ for $s = i\omega \in i\mathbb{R}$.

$f(t)$	$\widetilde{f}(i\omega)$	$\lvert\widetilde{f}(i\omega)\rvert$	$phase(\widetilde{f}(i\omega))$
$\delta(t)$	1	1	0
$\mathbf{H}_0(t)$	$\frac{1}{i\omega}$	$\frac{1}{\lvert\omega\rvert}$	$-\frac{\pi}{2}$
$t\,\mathbf{H}_0(t)$	$\frac{1}{(i\omega)^2}$	$\frac{1}{\omega^2}$	$-\pi$
$\frac{t^2}{2}\mathbf{H}_0(t)$	$\frac{1}{(i\omega)^3}$	$\frac{1}{\lvert\omega\rvert^3}$	$\frac{\pi}{2}$

(C.397)

  Hence, the narrower $f(t)$ is, the broader $|\widetilde{f}(i\omega)|$ is. Also, the more $f(t)$ is a high pass filter, the more $|\widetilde{f}(i\omega)|$ is a low pass filter.

C.46 Z-transform

This section is a watered down version of the Wikipedia entry for Z-transforms (Ref.[197]), which we highly recommend. Before reading this section, we recommend that the reader read Section C.45 on Laplace transforms, as those are the continuous in time version of Z-transforms.

Suppose $x^{[n]} \in \mathbb{C}$ for all $n \in \mathbb{Z}^{\geq 0}$ ($\mathbb{Z}^{\geq 0}$ =non-negative integers), and $z \in \mathbb{C}$. Then we define the **Z-transform (ZT)** by

$$\mathcal{Z}[x](z) = \widetilde{x}(z) = \sum_{n=0}^{\infty} x^{[n]} z^{-n} \tag{C.398}$$

Note that the ZT is a linear functional because

$$\mathcal{Z}[ax^{[n]} + by^{[n]}] = a\mathcal{Z}[x^{[n]}] + b\mathcal{Z}[y^{[n]}] \tag{C.399}$$

for $x^{[n]}, y^{[n]} \in \mathbb{C}$ for all $n \in \mathbb{Z}^{\geq 0}$, and $a, b \in \mathbb{C}$.

The **Inverse Z-transform** is defined so that

$$\mathcal{Z}^{-1}[\underbrace{\mathcal{Z}[x^{[n]}]}_{\widetilde{x}(z)}]^{[n]} = x^{[n]} \tag{C.400}$$

Table C.3 is a table of transforms that are akin to the ZT.

For models that are continuous in time (i.e., analog) we use Laplace transforms (LTs), and for models that are discrete in time (i.e., digital), we use Z-transforms (ZTs). Digital models are often obtained by sampling analog models at discrete times separated by a time interval T. Hence, it is useful to know how LTs and ZTs are related. To find out, let

$$e^{-st} = z^{-n} \tag{C.401}$$

name	formula	comment
Bilateral ZT (BZT)	$\widetilde{x}(z) = \sum_{n=-\infty}^{\infty} x^{[n]} z^{-n}$	Same as ZT but with $n \in \mathbb{Z}$ instead of $n \in \mathbb{Z}^{\geq 0}$
Discrete time Fourier transform (DTFT)	$\widetilde{x}_{2\pi}(\omega) = \sum_{n=-\infty}^{\infty} x^{[n]} e^{-i\omega n}$	same as BZT but with $z = e^{i\omega}$
Discrete Fourier transform (DFT)	$\widetilde{x}^{[k]} = \sum_{n=0}^{N-1} x^{[n]} e^{-i\frac{2\pi k n}{N}}$	Same as ZT but a finite (N) number of $x^{[n]}$ components, and with $z = e^{i\frac{2\pi k}{N}}$ for $k = 0, 1, \ldots, N-1$ (N roots of unity on unit circle)
Probability Generating Function	$\widetilde{P}(z) = \sum_{n=0}^{\infty} P^{[n]} z^n$	same as ZT but with $n \to -n$ and $P^{[n]} : \mathbb{Z}^{\geq 0} \to [0, 1]$ is a discrete prob. distribution. If $z = e^{-T}$, get moment genetating function.

Table C.3: Transforms that are akin to the Z-transform. Don't be intimidated by all these transforms. They are all just fancy dot products like $\vec{a} \cdot \vec{b}$.

and

$$t = nT \tag{C.402}$$

Hence, we arrive at the very useful formula

$$\boxed{z = e^{sT}} \tag{C.403}$$

If

$$s = \sigma + i\omega, \quad z = re^{i\theta} \tag{C.404}$$

then

$$r = e^{\sigma T}, \quad \theta = \omega T \tag{C.405}$$

The map given by Eqs.(C.405) is illustrated by Fig.C.15. As shown in Fig.C.15, the map from the s-plane (for LTs) to the z-plane (for ZTs) maps:

Figure C.15: Relationship between the s-plane (for Laplace transform) and z-plane (for Z-transform.).

- left half plane (LHP) $\rightarrow$ inside of the unit circle
- imaginary axis $\rightarrow$ unit circle. In particular, the s-plane origin $s = 0$ and points with $\sigma = 0, \omega = \frac{2\pi k}{T}$, where $k \in \mathbb{Z}$, are all mapped to $z = 1$.
- right half plane (RHP) $\rightarrow$ outside of unit circle

Define the **Kronecker delta function** by

$$\delta_j^{[n]} = \mathbb{1}(n = j) \tag{C.406}$$

and the **discrete Heavyside step function** by

$$\mathbf{H}_j^{[n]} = \mathbb{1}(n \geq j) \tag{C.407}$$

Note that

$$H_0^{[n]} = \sum_{k=0}^{\infty} \delta_k^{[n]} \tag{C.408}$$

Next, we will list some examples and properties of ZTs. Henceforth, we will use the following notation:

$x^{[n]}, x_1^{[n]}, x_2^{[n]} \in \mathbb{C}$ are defined only for $n \in \mathbb{Z}^{\geq 0}$

$x^{[n]} \xrightarrow[\mathcal{Z}]{} \widetilde{x}(z)$ means $\mathcal{Z}[x^{[n]}] = \widetilde{x}(z)$

The **Region Of Convergence (ROC)** for a ZTs is very important (Without knowing the ROC, you can't invert a ZT $\widetilde{x}(z)$). We won't list ROCs here, but they can be found in ZT tables like those in Ref.[197].

Note that ZT formulae can be "sanity checked" by replacing $z = e^{sT}$ and checking that for $0 < T << 1$,

$$\mathcal{Z}[x](e^{sT}) \approx \frac{1}{T}\mathcal{L}[f](s) \tag{C.409}$$

Note also that for $T << 1$:

$$z \approx 1, \quad z\partial_z = \partial_{\ln z} = \partial_{sT}, \quad (z-1)^n \approx (sT)^n \tag{C.410}$$

$\int_{0-}^{\infty} dt\ \delta(t) = 1 = \sum_{n=0}^{\infty} \delta_0^{[n]}$ so by dimensional analysis, $\delta_0^{[n]} \approx T\delta(t)$.
$\mathbf{H}_0^{[n]} \approx \mathbf{H}_0(t)$

C.46.1 Examples

- Kronecker delta function

$$\delta_{n_0}^{[n]} \xrightarrow[\mathcal{Z}]{} z^{-n_0} \tag{C.411}$$

Compare this with

$$\delta(t - t_0) \xrightarrow[\mathcal{L}]{} e^{-st_0} \tag{C.412}$$

with $t_0 = n_0 T$ and $z = e^{sT}$.

Note that

$$\begin{aligned} \mathcal{Z}[\mathbf{H}_0^{[n]}] &= \sum_{k=0}^{\infty} \mathcal{Z}[\delta_k^{[n]}] && \text{(C.413)} \\ &= \sum_{k=0}^{\infty} z^{-k} && \text{(C.414)} \\ &= \frac{1}{1 - 1/z} && \text{(C.415)} \end{aligned}$$

- unit step

For $a \in \mathbb{C}$,

$$a^n \mathbf{H}_0^{[n]} \xrightarrow[\mathcal{Z}]{} \frac{1}{1 - az^{-1}} \quad \text{for } |z| > |a| \tag{C.416}$$

$$-a^n \mathbf{H}_0^{[-n-1]} \xrightarrow[\mathcal{Z}]{} \frac{1}{1 - az^{-1}} \quad \text{for } |z| < |a| \tag{C.417}$$

Compare this with

$$\mathbf{H}_0(t) \xrightarrow[\mathcal{L}]{} \frac{1}{s} \quad (\text{for } Re(s) > 0) \tag{C.418}$$

for $a = 1$, $z = e^{sT} \approx 1 + sT$.

- ramp

  For $a \in \mathbb{C}$,

$$na^n \mathbf{H}_0^{[n]} \xrightarrow[\mathcal{Z}]{} \frac{az^{-1}}{(1 - az^{-1})^2} \quad \text{for } |z| > |a| \tag{C.419}$$

$$-na^n \mathbf{H}_0^{[-n-1]} \xrightarrow[\mathcal{Z}]{} \frac{az^{-1}}{(1 - az^{-1})^2} \quad \text{for } |z| < |a| \tag{C.420}$$

- sine, cosine

  For $a \in \mathbb{C}$,

$$a^n \sin(\omega_0 n) \mathbf{H}_0^{[n]} \xrightarrow[\mathcal{Z}]{} \frac{az^{-1} \sin \omega_0}{1 - 2az^{-1} \cos \omega_0 + a^2 z^{-2}} \tag{C.421}$$

$$a^n \cos(\omega_0 n) \mathbf{H}_0^{[n]} \xrightarrow[\mathcal{Z}]{} \frac{1 - az^{-1} \cos \omega_0}{1 - 2az^{-1} \cos \omega_0 + a^2 z^{-2}} \tag{C.422}$$

C.46.2 Properties

- time expansion

$$x^{[n/K]} \mathbb{1}(n/K \in \mathbb{Z}) \xrightarrow[\mathcal{Z}]{} \widetilde{x}(z^K) \tag{C.423}$$

- decimation

$$x^{[Kn]} \xrightarrow[\mathcal{Z}]{} \frac{1}{K} \sum_{p=0}^{K-1} \widetilde{x}\left(z^{\frac{1}{K}} e^{-i2\pi \frac{p}{K}}\right) \tag{C.424}$$

- time delay

$$x^{[n-k]} \xrightarrow[\mathcal{Z}]{} z^{-k} \widetilde{x}(z) \quad (\text{for } k > 0) \tag{C.425}$$

- time advance

$$x^{[n+k]} \xrightarrow[\mathcal{Z}]{} z^k \left(\widetilde{x}(z) - z^k \sum_{n=0}^{k-1} x^{[n]} z^{-n} \right) \quad (\text{for } k > 0) \tag{C.426}$$

- first difference backwards

$$x^{[n]} - x^{[n-1]} \xrightarrow[\mathcal{Z}]{} (1 - z^{-1})\widetilde{x}(z) \tag{C.427}$$

- first difference forward

$$x^{[n+1]} - x^{[n]} \xrightarrow[\mathcal{Z}]{} z\left((1 - z^{-1})\widetilde{x} - x^{[0]}\right) \tag{C.428}$$

- time reversal

$$x^{[-n]} \xrightarrow[\mathcal{Z}]{} \widetilde{x}(z^{-1}) \tag{C.429}$$

- scaling in z-domain

  For $a \in \mathbb{C}$,

$$a^n x^{[n]} \xrightarrow[\mathcal{Z}]{} \widetilde{x}(a^{-1}z) \tag{C.430}$$

- complex conjugation

$$(x^{[n]})^* \xrightarrow[\mathcal{Z}]{} \widetilde{x}^*(z^*) \tag{C.431}$$

$$Re(x^{[n]}) \xrightarrow[\mathcal{Z}]{} \frac{1}{2}(\widetilde{x}(z) + \widetilde{x}^*(z^*)) \tag{C.432}$$

$$Im(x^{[n]}) \xrightarrow[\mathcal{Z}]{} \frac{1}{2i}(\widetilde{x}(z) - \widetilde{x}^*(z^*)) \tag{C.433}$$

- $\widetilde{x}(z)$ differentiation

$$nx^{[n]} \xrightarrow[\mathcal{Z}]{} -z\partial_z\widetilde{x}(z) \tag{C.434}$$

- convolution

  Define the **(discrete) convolution** of $x_1^{[n]}$ and $x_2^{[n]}$ by

$$x_1^{[n]} \circledast x_2^{[n]} = \sum_{k=0}^{n} x_1^{[k]} x_2^{[n-k]} \tag{C.435}$$

  One can show that

$$x_1^{[n]} \circledast x_2^{[n]} \xrightarrow[\mathcal{Z}]{} \widetilde{x}_1(z)\widetilde{x}_2(z) \tag{C.436}$$

- cross-correlation

$$(x_1^{[-n]})^* \circledast x_2^{[n]} \xrightarrow[\mathcal{Z}]{} \widetilde{x_1}^* \left(\frac{1}{z^*}\right))\widetilde{x}_2(z) \tag{C.437}$$

- accumulation

$$\sum_{k=-\infty}^{\infty} x^{[k]} \xrightarrow[\mathcal{Z}]{} \frac{1}{1-z^{-1}}\widetilde{x}(z) \tag{C.438}$$

- multiplication

$$x_1^{[n]} x_2^{[n]} \xrightarrow[\mathcal{Z}]{} \frac{1}{2\pi i}\oint_C \frac{dw}{w}\, \widetilde{x}_1(w)\widetilde{x}_2\left(\frac{z}{w}\right) \tag{C.439}$$

- Parseval's theorem

$$\sum_{k=-\infty}^{\infty} x_1^{[n]}(x_2^{[n]})^* = \frac{1}{2\pi i}\oint_C \frac{dw}{w}\, \widetilde{x}_1(w)\widetilde{x}_2^*\left(\frac{1}{w^*}\right) \tag{C.440}$$

- limits of $x^{[n]}$

  initial value theorem

$$x^{[0]} = \lim_{z\to\infty} \widetilde{x}(z) \tag{C.441}$$

  final value theorem

$$x^{[\infty]} = \lim_{z\to 1}(z-1)\widetilde{x}(z) \tag{C.442}$$

- Inverse ZT

$$\underbrace{\mathcal{Z}^{-1}[\widetilde{x}(z)]}_{x^{[n]}} = \frac{1}{2\pi i}\oint_C dz\, \widetilde{x}(z) z^{n-1} \tag{C.443}$$

  where C is a counterclockwise closed path containing the origin and all singularities of $\widetilde{x}(z)$.

Figure C.16: The line $y = xp$ goes through the origin and has arbitrary slope p. p^* is the special slope at x for which the line $y = xp^*$, if displaced parallelly, can be made tangential (kissing) to the curve $y = f(x)$.

Figure C.17: The dual function $\widetilde{f}(p)$ of a concave or convex function $f(x)$ is the osculating (kissing) locus of the family of lines $y = px - \widetilde{f}(p)$, where p is the slope at x of the curve $y = f(x)$.

C.47 Legendre Transformation (dual functions)

This section is a watered down version of the Wikipedia article Ref.[157], which we highly recommend.

Let $x, p \in \mathbb{R}^n$. If $f : \mathbb{R}^n \to \mathbb{R}$ is a concave function, we define its **dual (a.k.a. conjugate) function** $\widetilde{f} : \mathbb{R}^n \to \mathbb{R}$ by

$$\widetilde{f}(p) = \min_x (p^T x - f(x)) \ . \tag{C.444}$$

This definition also applies if one replaces the words "concave" by "convex" and "min" by "max".[17] $\widetilde{f}(p)$ is also called the **Legendre transformation (or Legendre transform)** (LT) of $f(x)$.

[17]This book does not try to be mathematically rigorous beyond the level of applied math. To be

In Physics, x is the position vector and p is the momentum vector of a system. See Figs. C.16 and C.17 for a pictorial representation of LT.
Note that the right hand side of Eq.(C.444) implies that

$$\sum_i (p_i - \partial_{x_i} f(x))\delta x_i = 0 \tag{C.445}$$

for all variations δx_i. If minimization is achieved when $x = x^*$, then

$$p = \nabla_{x^*} f(x^*), \quad x^* = (\nabla_{x^*} f)^{-1}(p) \tag{C.446}$$

C.47.1 Examples

1. Find the dual function of

$$\boxed{f(x) = e^x}\,. \tag{C.447}$$

$$p = \partial_{x^*} f \tag{C.448}$$
$$= e^{x^*} \tag{C.449}$$

$$x^* = \ln p \tag{C.450}$$

$$f(x^*) = p \tag{C.451}$$

$$\widetilde{f}(p) = x^* p - f(x^*) \tag{C.452}$$
$$= p \ln p - p \tag{C.453}$$

---

2. Find the dual function of

$$\boxed{f(x) = f + f'x + \frac{1}{2} f'' x^2}\,. \tag{C.454}$$

$$p = \partial_{x^*} f \tag{C.455}$$
$$= f' + f'' x^* \tag{C.456}$$

truly rigorous and general, replace "max" by "supremum" and "min" by "infimum". Pure mathematicians use min and max only over finite sets, but physicists and engineers often discretize to obtain a max or min over a finite set with N points, and then, afterwards, take the limit $N \to \infty$. Not perfect, but good enough for most applied work.

$$x^* = \frac{p - f'}{f''} \tag{C.457}$$

$$\begin{aligned} f(x^*) &= f + f'\left[\frac{p-f'}{f''}\right] + \frac{1}{2}f''\left[\frac{p-f'}{f''}\right]^2 && \text{(C.458)} \\ &= \left[f - \frac{(f')^2}{2f''}\right] + p^2\left[\frac{1}{2f''}\right] && \text{(C.459)} \end{aligned}$$

$$\begin{aligned} \widetilde{f}(p) &= px^* - f(x^*) && \text{(C.460)} \\ &= p\left[\frac{p-f'}{f''}\right] - f(x^*) && \text{(C.461)} \\ &= \left[-f + \frac{(f')^2}{2f''}\right] + p\left[\frac{-f'}{f''}\right] + p^2\left[\frac{1}{2f''}\right] && \text{(C.462)} \end{aligned}$$

Note that when $f = f' = 0$, we get

$$f(x) = \frac{f''x^2}{2}, \quad \widetilde{f}(p) = \frac{p^2}{2f''} \tag{C.463}$$

Note that if f is convex (resp., concave), $\widetilde{f}$ is convex too (resp., concave too).

---

3. Find the dual function of

$$\boxed{f(x) = \ln(1 - e^{-x})}\,. \tag{C.464}$$

$$\begin{aligned} p &= \partial_{x^*} f(x^*) && \text{(C.465)} \\ &= \frac{e^{-x^*}}{1 - e^{-x^*}} && \text{(C.466)} \end{aligned}$$

$$p = (1+p)e^{-x^*} \tag{C.467}$$

$$x^* = \ln\frac{1+p}{p} \tag{C.468}$$

$$f(x^*) = \ln\left(1 - \frac{p}{1+p}\right) = -\ln(1+p) \tag{C.469}$$

$$\begin{aligned} \widetilde{f}(p) &= px^* - f(x^*) && \text{(C.470)} \\ &= p\ln\frac{1+p}{p} + \ln(1+p) && \text{(C.471)} \\ &= -p\ln p + (1+p)\ln(1+p) && \text{(C.472)} \end{aligned}$$

Figure C.18: $f(x) = e^x$ and its dual.

Figure C.19: $f(x) = \ln(1 - e^{-x})$ and its dual.

C.47.2 Properties

Claim 18 *If $f(x)$ is a concave function with dual $\widetilde{f}(p)$, then*

$$f(x) = \min_p (p^T x - \widetilde{f}(p)) \tag{C.473}$$

$$\nabla_p \widetilde{f}(p(x)) = (\nabla_x f)^{-1}(x) \tag{C.474}$$

Eqs.(C.444) and (C.473) are also true if we replace the words "concave" with "convex" and "min" with "max".

proof:

Let x^* be the value of x which minimizes $y = p^T x - f(x)$ with respect to x. Hence,

$$\widetilde{f}(p) = p^T x^* - f(x^*) \ . \tag{C.475}$$

Rearranging terms in Eq.(C.475), we get

$$f(x^*) = p^T x^* - \widetilde{f}(p) \tag{C.476}$$

Let p^* be the value of p which minimizes $y = p^T x - \widetilde{f}(p)$ with respect to p. Hence,

$$f(x) = (p^*)^T x - \widetilde{f}(p^*) \tag{C.477}$$

Replacing p^* by p and x by x^* in Eq.(C.477) yield Eq.(C.476).

Note that minimization with respect to x is achieved if

$$p_i = \partial_{x_i} f(x), \ p = \nabla_x f(x) \tag{C.478}$$

whereas minimization with respect to p is achieved if

$$x_i = \partial_{p_i} \widetilde{f}(p), \ x = \nabla_p \widetilde{f}(p) \tag{C.479}$$

Hence,

$$x = \nabla_p \widetilde{f}(\nabla_x f(x)) \tag{C.480}$$

QED

Claim 19 *If $f(x)$ is concave (resp., convex), then $\widetilde{f}(p)$ is also concave (resp., convex)*

proof:

$$\widetilde{f}(p) = x^* p - f(x^*) \tag{C.481}$$

$$\frac{d\widetilde{f}}{dp} = x^* + \underbrace{(p - f'(x^*))}_{=0} \frac{dx^*}{dp} = x^* \tag{C.482}$$

$$\frac{d^2\widetilde{f}}{dp^2} = \frac{dx^*}{dp} \tag{C.483}$$

$$p = f'(x^*), \quad x^* = (f')^{-1}(p) \tag{C.484}$$

$$dp = f''(x^*)dx^*, \quad \frac{dx^*}{dp} = \frac{1}{f''(x^*)} \tag{C.485}$$

$$\frac{d^2\widetilde{f}}{dp^2} = \frac{1}{f''(x^*)} \tag{C.486}$$

QED

Claim 20 *LT is its own inverse (i.e., LT is a self-inverse or involution transformation)*

$$\widetilde{\widetilde{f}}(x) = f(x) \tag{C.487}$$

proof: $x = x_1, p = x_2$

$$\widetilde{f}(x_2) = x_2 x_1^* - f(x_1^*), \quad x_2 = f'(x_1^*) \tag{C.488}$$

$$\widetilde{\widetilde{f}}(x_3) = x_3 x_2^* - \widetilde{f}(x_2^*), \quad x_3 = (\widetilde{f})'(x_2^*) \tag{C.489}$$

$$\begin{aligned} \widetilde{\widetilde{f}}(x_3) &= x_3 x_2^* - x_2^* x_1^* + f(x_1^*) && \text{(C.490)} \\ &= f(x_1^*) \quad (\text{ for } x_1^* = x_3) && \text{(C.491)} \end{aligned}$$

$$x = x_3 = (\widetilde{f})'(x_2^*) = (\widetilde{f})'(f'(x_1^*)) = x_1^* \tag{C.492}$$

QED

Additional properties

- Scaling

$$ag(bx) \xrightarrow[LT]{} a\widetilde{g}(\frac{p}{ab}) \tag{C.493}$$

  for $a, b > 0$

- Translation

$$g(x+y) + b \xrightarrow[LT]{} \widetilde{g}(p) - p^T y - b \tag{C.494}$$

- Frenchel's inequality

$$p^T x \leq f(x) + \widetilde{f}(p) \tag{C.495}$$

C.47.3 Connection to Fourier transform and Quantum Mechanics

Recall how Fourier transforms (FTs) arise in Quantum Mechanics. Suppose $x, p \in \mathbb{R}$ and

$$\psi(x) = \langle x|\psi\rangle\,, \quad \widetilde{\psi}(p) = \langle p|\psi\rangle\,, \quad \langle p|x\rangle = \frac{e^{-ipx}}{\sqrt{2\pi}} \tag{C.496}$$

Then

$$\widetilde{\psi}(p) = \int_{-\infty}^{\infty} dx\, \langle p|x\rangle\, \langle x|\psi\rangle \tag{C.497}$$

$$= \int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}}\, e^{-ipx}\psi(x) \tag{C.498}$$

and

$$\psi(x) = \int_{-\infty}^{\infty} dp\, \langle x|p\rangle\, \langle p|\psi\rangle \tag{C.499}$$

$$= \int_{-\infty}^{\infty} \frac{dp}{\sqrt{2\pi}}\, e^{ipx}\widetilde{\psi}(p) \tag{C.500}$$

Define a **convolution** of two wave functions $\psi_1, \psi_2 : \mathbb{R} \to \mathbb{C}$ by

$$(\psi_1 \circledast \psi_2)(x) = \int_{-\infty}^{\infty} dy\; \psi_1(y)\psi_2(x-y) \tag{C.501}$$

Then

$$(\psi_1 \circledast \psi_2)^{\sim}(p) = \widetilde{\psi_1}(p)\widetilde{\psi_2}(p) \tag{C.502}$$

If, in the definition of LT, we replace the minimum over x of an arbitrary function $\Gamma : \mathbb{R} \to \mathbb{R}$ by

$$\min_x \Gamma(x) \to i \ln \int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}} \exp\{-i\Gamma(x)\} \tag{C.503}$$

then we get the definition of a FT:

$$\underbrace{e^{-i\widetilde{f}(p)}}_{\widetilde{\psi}(p)} = \int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}}\, e^{-ip^T x}\, \underbrace{e^{if(x)}}_{\psi(x)} \tag{C.504}$$

Define the **infimal convolution** of two functions $f, g : \mathbb{R}^n \to \mathbb{R}$ by

$$(f \circledast_{inf} g)(x) = \min_y \{f(y) + g(x-y)\} \tag{C.505}$$

Then

$$(f \circledast_{inf} g)^{\sim}(p) = \widetilde{f}(p) + \widetilde{g}(p) \tag{C.506}$$

Eq.(C.506) requires a proof that we leave to the reader, but note that it's just what would be expected from our "mapping" of LT to FT.

C.48 Numpy tensor methods

Numpy contains excellent documentation which we highly recommend. So why this appendix? The purpose of this appendix is to compare the tensor operations used in Physics[18] to the tensor operations used in Machine Learning (ML) as exemplified by Numpy.

But first, some notation.

For integers $m > n$, let

$$[n : m] = [n, n+1, \ldots, m-2, m-1] \tag{C.507}$$

and

$$[n] = [0 : n] \tag{C.508}$$

$[n : m]$ acts the same way as the function `range(n,m)` in Python.

We will use

$$A^{[n]} = \begin{bmatrix} A^0 \\ A^1 \\ \vdots \\ A^{n-1} \end{bmatrix} = A^{[n],[1]} \tag{C.509}$$

to denote column vector,

$$A^{[n]T} = [A^0, A^1, \cdots, A^{n-1}] = A^{[1],[n]} \tag{C.510}$$

to denote a row vector and

$$A = \begin{bmatrix} A^{0,0} & \cdots & A^{0,m-1} \\ \vdots & \vdots & \vdots \\ A^{n-1,0} & \cdots & A^{n-1,m-1} \end{bmatrix} = A^{[n],[m]} \tag{C.511}$$

to denote an $n \times m$ matrix.

Let

$$pow(a, r) = \underbrace{[a, a, \ldots, a]}_{r \text{ repetitions}} \tag{C.512}$$

For increased clarity, we will use Greek letters to denote tensor indices.

For **scalars** $a, b \in \mathbb{R}$, the **linear combination of tensors** T, S is defined by

[18] Elasticity, Fluid Mechanics, General Relativity and Quantum Mechanics all use tensors. Tensors in various guises are ubiquitous in Physics.

$$[T^{\alpha,\beta,\gamma}, S^{\alpha,\beta,\gamma}] \to aT^{\alpha,\beta,\gamma} + bS^{\alpha,\beta,\gamma} \tag{C.513}$$

The **Kronecker (K) delta function** is defined by

$$\delta_\alpha^\beta = \delta_{\alpha,\beta} = \mathbb{1}(\alpha = \beta) \tag{C.514}$$

Upper indices can be lowered by means of the K delta function :

$$S_\beta = \sum_\alpha S^\alpha \delta_{\alpha,\beta} \tag{C.515}$$

In this case, since the metric $\delta_{\alpha,\beta}$ is the K delta function, $S^\alpha = S_\alpha$.

The **Einstein implicit summation convention** is the practice of omitting the summation sign and assuming that repeated indices are summed over if one index is covariant (upper) and the other is contravariant (lower).

$$S^\alpha T_\alpha = \sum_\alpha S^\alpha T_\alpha \tag{C.516}$$

Sums between a lower and an upper index are called **contractions**.

dim refers to the positions of indices in a tensor (e.g., $dim = 0$ for α, $dim = 1$ for β and $dim = 2$ for γ in $T^{\alpha,\beta,\gamma}$). $axis \in [n]$ refers to the values of an index along a particular dimension (e.g., we say $axis = \alpha$ along $dim = 0$ in $T^{\alpha,\beta,\gamma}$). For a 2-dim array: (1) swapping axes along $dim = 0$ (resp., $dim = 1$) refers to swapping rows (resp., columns). (2) swapping dimensions 0 and 1 means transposing the array.

Numpy contains a huge number of tensor methods. Among those, there are 3 broad types of methods that concern us here:

1. **tensor algebra methods**. These include element-wise (i.e., entry-wise) summation, subtraction, multiplication and division $(+, -, *, /)$ of 2 tensors of the same shape, or a tensor and a scalar.

2. **methods for permuting the entries of a tensor**. These entry permutation methods can be of 3 kinds (1) methods that permute entries by permuting the locations of indices of a tensor (2) methods that permute entries by permuting the axes of a tensor along a particular dimension (e.g., row permutation and column permutation for 2-dim arrays) (3) methods that are neither pure 1 or pure 2.

3. **methods for adding or removing tensor entries**.

Out of these three categories, ML uses all three frequently. Physics uses all three too, but it often favors (1). [19] This is probably due to the fact that Physicists always

[19] Category (1) echoes the Linear Algebra and category (2) the Group Representation Theory used in Quantum Mechanics to describe lossless, reversible physical phenomena. Category (3) echoes the Information Theory, Thermodynamics, Renormalization Group Theory, Noise Theory and Fluid Turbulence Theory used to describe lossy, irreversible phenomena.

assume linearity first, because it's simpler to solve than the non-linear case, plus it often describes the weak interaction case well.

Next I will discuss in a visual manner[20], a random assortment of Numpy methods that I find interesting, and difficult to understand to the beginner (like me). This discussion in no way pretends to be a substitute for the excellent Numpy documentation.

Besides the usual Physics notation discussed at the beginning of this appendix, I will use below my own way of visualizing Numpy tensor methods. Specifically, I will use a graphical box (what I call a "box of puzzle pieces") to indicate a box containing all the pieces of information from which a tensor is constructed. If you don't like my graphical box, just replace it by $f(X)$, where X is the contents of the box and f is some function.

Below, let $\alpha \in [a]$, $\beta \in [b]$, $\gamma \in [c]$, $\nu \in [n]$, $\mu \in [m]$, $\nu_i \in [n_i]$.

1. **broadcasting**

$$T^{\alpha,0} + S^{0,\beta} \to Y^{\alpha,\beta} = T^{\alpha,0} + S^{0,\beta} \tag{C.517}$$

2. **concatenate()** along $dim = 0$

$$[T_0^{[n_0],\beta,\gamma}, T_1^{[n_1],\beta,\gamma}, T_2^{[n_2],\beta,\gamma}] \to \boxed{\begin{matrix} T_0^{[n_0],\beta,\gamma} \\ T_1^{[n_1],\beta,\gamma} \\ T_2^{[n_2],\beta,\gamma} \end{matrix}}^{[n],\beta,\gamma} \tag{C.518}$$

where $n = n_0 + n_1 + n_2$.

3. **expand_dims()** (same as **unsqueeze()**) along $dim = 0$

$$T^{[b],[c]} \to Y^{0,[b],[c]} \tag{C.519}$$

$$T^{\beta,\gamma} \to Y^{0,\beta,\gamma} = T^{\beta,\gamma} \tag{C.520}$$

4. **flatten()**

$$T^{[n_0],[n_1],[n_2]} \to Y^{[n]} \tag{C.521}$$

where $n_0 n_1 n_2 = n$. A more fine grained description is

$$T^{\alpha,\beta,\gamma} \to Y^{\nu(\alpha,\beta,\gamma)} \tag{C.522}$$

where $n_0 n_1 n_2 = n$ and $\nu : [n_0] \times [n_1] \times [n_2] \to [n]$ is a 1-1 onto function.

[20] The Numpy methods in this list are discussed here visually and analytically only; that is, without numerical examples. For numerical examples, see the excellent Numpy docs.

5. **gather()**[21] along $dim = 0$

$$S^{[a],[n_1],[n_2]} \to Y^{[b],[n_1],[n_2]} \tag{C.523}$$

$$S^{\alpha,\nu_1,\nu_2} \to Y^{\beta,\nu_1,\nu_2} = S^{\beta(\alpha,\nu_1,\nu_2),\nu_1,\nu_2} \tag{C.524}$$

$$I^{\alpha,\nu_1,\nu_2} = \beta(\alpha,\nu_1,\nu_2) \tag{C.525}$$

`source=` $S^{[a],[n_1],[n_2]}$, `index=` $I^{[a],[n_1],[n_2]}$

6. **max()** and **argmax()** along $dim = 0$

$$\max: \ T^{[a],[b]} \to T^{\alpha_0,[b]} \tag{C.526}$$

$$\text{argmax}: \ T^{[a],[b]} \to \alpha_0 \tag{C.527}$$

where $T^{\alpha_0,\beta} = \max\{T^{\alpha,\beta} : \alpha \in [a]\}$

7. **repeat()** with $r = [r_0, r_1, \ldots, r_{n-1}]$ along $dim = 0$

$$T^{[n],\beta,\gamma} \to \boxed{\begin{array}{l} pow(T^{0,\beta,\gamma}, r_0) \\ pow(T^{1,\beta,\gamma}, r_1) \\ \vdots \\ pow(T^{n-1,\beta,\gamma}, r_{n-1}) \end{array}}^{[R],\beta,\gamma} \tag{C.528}$$

where $R = \sum_{i=0}^{n-1} r_i$. note that concatenate() along $dim = 0$ and repeat() with $r = pow(1, n)$ along $dim = 0$, are the same thing. So repeat() is a souped up version of concatenate().

8. **reshape()** from shape (n_0, n_1, n_2) to shape (n, m)

$$T^{[n_0],[n_1],[n_2]} \to Y^{[n],[m]} \tag{C.529}$$

where $n_0 n_1 n_2 = nm$. A more fine grained description is

$$T^{\alpha,\beta,\gamma} \to Y^{\mu(\alpha,\beta,\gamma),\nu(\alpha,\beta,\gamma)} \tag{C.530}$$

where $n_0 n_1 n_2 = nm$ and $\mu : [n_0] \times [n_1] \times [n_2] \to [n]$ and $\nu : [n_0] \times [n_1] \times [n_2] \to [m]$ are 1-1 onto functions. flatten() is clearly a special case of reshape().

[21] This operation is available in PyTorch. So far Numpy doesn't have it in direct form.

9. **split()** along $dim = 0$

$$\boxed{\begin{array}{l} T_0^{[n_0],\beta,\gamma} \\ T_1^{[n_1],\beta,\gamma} \\ T_2^{[n_2],\beta,\gamma} \end{array}}^{[n],\beta,\gamma} \rightarrow [T_0^{[n_0],\beta,\gamma}, T_1^{[n_1],\beta,\gamma}, T_2^{[n_2],\beta,\gamma}] \tag{C.531}$$

where $n = n_0 + n_1 + n_2$

10. **squeeze()** $dim = 0$

$$T^{0,[b],[c]} \rightarrow Y^{[b],[c]} \tag{C.532}$$

$$T^{0,\beta,\gamma} \rightarrow Y^{\beta,\gamma} = T^{0,\beta,\gamma} \tag{C.533}$$

11. **stack()** along $dim = 0$

$$[T_0^{\alpha,\beta,\gamma}, T_1^{\alpha,\beta,\gamma}, T_2^{\alpha,\beta,\gamma}] \rightarrow \boxed{\begin{array}{l} T_0^{\alpha,\beta,\gamma} \\ T_1^{\alpha,\beta,\gamma} \\ T_2^{\alpha,\beta,\gamma} \end{array}}^{[3],\alpha,\beta,\gamma} \tag{C.534}$$

Compare this to concatenate(). stack() creates a new dimension whereas concatenate() doesn't. concatenate() just increases the range of an existing dimension.

12. **sum()** along $dim = 0$

$$T^{[a],[b]} \rightarrow \sum_{\alpha} T^{\alpha,[b]} \tag{C.535}$$

13. **tensordot()** (i.e., contraction) along $dim = 0$

$$[T^{[n],\beta}, S^{[n],\beta}] \rightarrow T^{\alpha,\beta}\delta_{\alpha,\alpha'}S^{\alpha',\beta} = \sum_{\alpha} T^{\alpha,\beta}S^{\alpha,\beta} \tag{C.536}$$

14. **tile()**[22]

$$\texttt{reps} = [2, 3] \tag{C.537}$$

$$T^{[a],[b]} \rightarrow \boxed{\begin{array}{ccc} T^{[a],[b]} & T^{[a],[b]} & T^{[a],[b]} \\ T^{[a],[b]} & T^{[a],[b]} & T^{[a],[b]} \end{array}}^{[2a],[3b]} \tag{C.538}$$

[22] tile() in Numpy corresponds to repeat() in PyTorch.

$$T^{\alpha,\beta} \to \boxed{\begin{matrix} T^{[a],[b]} & T^{[a],[b]} & T^{[a],[b]} \\ T^{[a],[b]} & T^{[a],[b]} & T^{[a],[b]} \end{matrix}}^{A(\alpha,\beta),B(\alpha,\beta)} \tag{C.539}$$

15. **transpose()** by a permutation $\sigma : [3] \to [3]$. [23]

$$T^{\alpha,\beta,\gamma} \to T^{\alpha_1,\beta_1,\gamma_1}\delta^{\sigma(\alpha,\beta,\gamma)}_{\alpha_1,\beta_1,\gamma_1} \tag{C.540}$$

For example,

$$T^{\alpha,\beta,\gamma} \to T^{\alpha_1,\beta_1,\gamma_1}\delta^{\gamma,\beta,\alpha}_{\alpha_1,\beta_1,\gamma_1} \tag{C.541}$$

[23] a permutation $\sigma : [n] \to [n]$ is a bijection, i.e., a 1-1 onto map.

Appendix D

Linear Regression, Ordinary Least Squares (OLS)

Wikipedia articles

1. Linear Regression (LR)
   - linear regression, Ref.[159]
   - simple linear regression, Ref.[181]
   - errors in variable, Ref.[132]
2. Least squares (LS)
   - least squares, Ref.[156]
   - ordinary least squares (OLS), Ref.[172]

Some nomenclature: Let $\sigma = 0, 1, 2, \ldots, nsam - 1$ denote the **sample**. In LR, the data consists of **independent x-variables** $x_{\sigma,1}, x_{\sigma,2}, \ldots x_{\sigma,n}$ and a **dependent y-variable** y_σ. We find a **linear fit** $\widehat{y}_\sigma = \beta_0 + \sum_{i=1}^n \beta_i x_{\sigma i}$ to the data. Fitting y with a hyperplane in the variables $x^n = (x_i)_{i=1}^n$ (i.e., finding the best coefficients $\beta^n = (\beta_i)_{i=1}^n$) is called **regressing** y **on** x^n. $\widehat{y}_\sigma$ is the **estimate** of the **estimand** y_σ. The coefficients β_0, β_i are called **regression coefficients**. $y_\sigma - \widehat{y}_\sigma = \epsilon^\sigma$ are called the **residuals**. $\mathcal{E} = \sum_\sigma (\epsilon^\sigma)^2$ is called the **error or cost**. We choose the regression coefficients so as to minimize the error.

Below, we consider two types of LR:

1. LR in which the independent x-variables are non-random.
2. LR in which the independent x-variables are random and i.i.d.

The term OLS is often used to refer to LR of type 1.

For LR of type 2, there is randomness in y coming from the randomness in x and in the residuals. For LR of type 1, there is randomness in y too, but it comes from the residuals only.

If the problem being considered requires LR, especially if some of the variables describing the problem are best modelled as random variables, it is useful to specify a visual "model" (i.e., a bnet, with probabilities expressed as TPMs).

D.1 LR, assuming x_σ are non-random

Let

$\sigma \in \{0, 1, 2, \ldots, nsam - 1\}$: sample index
$i_0 \in \{0, 1, 2, \ldots, n\}$: index that can assume values 0 to n
$i \in \{1, 2, \ldots, n\}$: index that can assume values 1 to n. i is never equal to 0.
$y_\sigma \in \mathbb{R}$: dependent y-variables
$x_{\sigma i} \in \mathbb{R}$: independent x-variables
$\epsilon_\sigma \in \mathbb{R}$: residuals
$\beta_0, \beta_i \in \mathbb{R}$: regression coefficients

$$y_\sigma = \beta_0 + \sum_{i=1}^{n} x_{\sigma i}\beta_i + \epsilon_\sigma \tag{D.1}$$

If we define

$$x_{\sigma 0} = 1 \tag{D.2}$$

for all σ, then

$$y_\sigma = \sum_{i_0=0}^{n} x_{\sigma i_0}\beta_{i_0} + \epsilon_\sigma \ . \tag{D.3}$$

If y and ϵ are $nsam$ dimensional column vectors and β is an $n+1$ dimensional column vector, and X is an $nsam \times (n+1)$ matrix, then we can write the previous equation in matrix form as:

$$y = X\beta + \epsilon \ . \tag{D.4}$$

D.1.1 Derivation of LR From Minimization of Error

Let $W = [W_{\sigma,\sigma'}]$ be a symmetric matrix with non-negative diagonal elements $W_{\sigma,\sigma} \geq 0$ for all σ. W is called the **weight matrix**. The following claim describes the method of **Weighted LR** when $W \neq 1$ and of simple LR when $W = 1$.

Claim 21 *Assume the Einstein summation convention; i.e., implicit sum over repeated indices. The error function $\mathcal{E}$ given by*

$$\mathcal{E} = \underbrace{(y_\sigma - x_{\sigma j_0}\beta_{j_0})}_{residual\ \epsilon_\sigma} W_{\sigma,\sigma'} \underbrace{(y_{\sigma'} - x_{\sigma' k_0}\beta_{k_0})}_{\epsilon_{\sigma'}} , \tag{D.5}$$

is minimized over β_{k_0} *for all* $k_0 \in \{0, 1, \ldots, n\}$*, if* β_{k_0} *is given by:*

$$\widehat{\beta} = (X^T W X)^{-1} X^T W y . \tag{D.6}$$

When $W = 1$*,*

$$\widehat{\beta} = \underbrace{(X^T X)^{-1} X^T}_{\partial_X} y . \tag{D.7}$$

proof:

At the minimum of $\mathcal{E}$, the variation $\delta\mathcal{E}$ must vanish:

$$0 = \delta\mathcal{E} = -2x_{\sigma j_0}(\delta\beta_{j_0}) W_{\sigma,\sigma'}(y_{\sigma'} - x_{\sigma' k_0}\beta_{k_0}) . \tag{D.8}$$

Thus,

$$X^T W y - X^T W X \beta = 0 \tag{D.9}$$

which implies Eq.(D.6).
QED

D.1.2 Geometry of LR with non-random x_σ.

Recall that

$$y = X\beta + \epsilon . \tag{D.10}$$

Define the **projection matrices**

$$I_X = X \underbrace{(X^T X)^{-1} X^T}_{\partial_X} , \quad A_X = 1 - I_X \tag{D.11}$$

A square matrix M is symmetric if $M^T = M$ and is idempotent if $M^2 = M$. I_X is symmetric and idempotent and so is A_X. Note that I_X and A_X also satisfy:

$$A_X I_X = I_X A_X = 0 \tag{D.12}$$

and

$$I_X X = X , \quad A_X X = 0 . \tag{D.13}$$

I_X acts as the identity on X, and A_X annihilates X.

One has

$$X^T(y-\epsilon) = X^T X\beta \tag{D.14}$$

Hence

$$\beta = (X^T X)^{-1} X^T (y-\epsilon)\ . \tag{D.15}$$

Define

$$\boxed{\widehat{\beta} = \underbrace{(X^T X)^{-1} X^T}_{\partial_X}\ y\ ,} \tag{D.16a}$$

$$\widehat{y} = X\widehat{\beta} = I_X y\ , \tag{D.16b}$$

and

$$\widehat{\epsilon} = y - X\widehat{\beta} = y - \widehat{y} = (1 - I_X)y = A_X y\ . \tag{D.16c}$$

I_X is sometimes called the **hat matrix**, because it gives y a hat.

Given any function $f = f(y, X, \epsilon)$ and a scalar factor $\xi \in \mathbb{R}$, suppose

$$f(\xi y, \xi X, \xi\epsilon) = \xi^\omega f(y, X, \epsilon) \tag{D.17}$$

Then we will say that $f(\cdot)$ is of **order ω under scaling**. Note that $\{\widehat{y}, \widehat{\epsilon}\}$ are all of order 1 under scaling, $\{\beta, \widehat{\beta}, I_X, A_X\}$ are all of order 0 under scaling, and ∂_X is of order -1 under scaling. Thus, each curve-fit (i.e., symbol with a hat) scales the same way as its estimand (i.e., same symbol without a hat). Furthermore, β, its curve-fit $\widehat{\beta}$, and the projection matrices I_X, A_X are invariant ($\omega = 0$) under scaling.

Note that y can be expressed as a sum of 2 orthogonal estimates:

$$y = \underbrace{\widehat{y}}_{I_X y} + \underbrace{\widehat{\epsilon}}_{A_X y}\ . \tag{D.18}$$

Fig.D.1 shows triangles representing $y = X\beta + \epsilon$ and $y = \widehat{y} + \widehat{\epsilon}$.

Figure D.1: Triangles representing $y = X\beta + \epsilon$ and $y = \widehat{y} + \widehat{\epsilon}$.

D.1.3 LR Goodness of Fit, R^2

Assume X and β are not random. This makes $\underline{y} = X\beta + \underline{\epsilon}$ and $\underline{\widehat{\beta}} = \partial_X \underline{y}$ random.

1. First assume that the components of ϵ are random with zero mean:

$$E[\underline{\epsilon}] = \langle \underline{\epsilon} \rangle = 0 \tag{D.19}$$

Then

$$\begin{cases} \left\langle \underline{y} \right\rangle = X\beta \\ \left\langle \underline{\widehat{y}} \right\rangle = I_X \underbrace{\left\langle \underline{y} \right\rangle}_{X\beta} = \left\langle \underline{y} \right\rangle \end{cases} \tag{D.20a}$$

$$\begin{cases} \langle \underline{\epsilon} \rangle = 0 \\ \langle \underline{\widehat{\epsilon}} \rangle = A_X \underbrace{\left\langle \underline{y} \right\rangle}_{X\beta} = 0 = \langle \underline{\epsilon} \rangle \end{cases} \tag{D.20b}$$

$$\left\langle \underline{\widehat{\beta}} \right\rangle = \left\langle \partial_X \underline{y} \right\rangle = \left\langle \partial_X (\underline{y} - \underline{\epsilon}) \right\rangle = \langle \beta \rangle = \beta \tag{D.20c}$$

2. So far, we have assumed that ϵ has zero mean value. Next, assume **"homoscedasticity" (homo-spread)**[1], which means that

$$\left\langle \underline{\epsilon}, \underline{\epsilon}^T \right\rangle = \xi^2 I_{nsam} \tag{D.20d}$$

where $\xi \geq 0$, and I_{nsam} is the $nsam \times nsam$ identity matrix. It follows that

$$\begin{cases} \left\langle \underline{\epsilon}, \underline{\epsilon}^T \right\rangle = \xi^2 I_{nsam} \\ \left\langle \underline{\widehat{\epsilon}}, \underline{\widehat{\epsilon}}^T \right\rangle = A_X \left\langle \underline{y}, \underline{y}^T \right\rangle A_X^T = \xi^2 A_X \end{cases} \tag{D.21a}$$

$$\begin{cases} \left\langle \underline{y}, \underline{y}^T \right\rangle = \left\langle \underline{\epsilon}, \underline{\epsilon}^T \right\rangle = \xi^2 I_{nsam} \\ \left\langle \underline{\widehat{y}}, \underline{\widehat{y}}^T \right\rangle = I_X \left\langle \underline{y}, \underline{y}^T \right\rangle I_X^T = \xi^2 I_X \end{cases} \tag{D.21b}$$

and

$$\left\langle \underline{\widehat{\beta}}, \underline{\widehat{\beta}}^T \right\rangle = \partial_X \left\langle \underline{y}, \underline{y}^T \right\rangle \partial_X^T = \xi^2 (X^T X)^{-1} \ . \tag{D.21c}$$

[1] I find the word "homoscedasticity" unnecessarily long, cryptic and easy to misspell so I like to replace it by "homo-spread". The opposite of "homoscedasticity" is "heteroscedasticity", which I like to replace with "hetero-spread".

For any random column vector $\underline{a}$, let

$$\| \underline{a} \|^2 = \underline{a}^T \underline{a} = \mathrm{tr}(\underline{a}\underline{a}^T) \tag{D.22}$$

Hence

$$\begin{aligned} \left\langle \| \underline{a} - \langle \underline{a} \rangle \|^2 \right\rangle &= \mathrm{tr} \left\langle [\underline{a} - \langle \underline{a} \rangle][\underline{a} - \langle \underline{a} \rangle]^T \right\rangle & \text{(D.23)} \\ &= \mathrm{tr} \left\langle \underline{a}, \underline{a}^T \right\rangle & \text{(D.24)} \end{aligned}$$

Define the following sums of squares (SS):

$$SS_{\underline{y}} = \left\langle \| \underline{y} - \left\langle \underline{y} \right\rangle \|^2 \right\rangle = \mathrm{tr} \left\langle \underline{y}, \underline{y}^T \right\rangle \tag{D.25a}$$

$$SS_{\underline{\widehat{y}}} = \left\langle \| \underline{\widehat{y}} - \left\langle \underline{\widehat{y}} \right\rangle \|^2 \right\rangle = \mathrm{tr} \left\langle \underline{\widehat{y}}, \underline{\widehat{y}}^T \right\rangle \tag{D.25b}$$

$$SS_{res} = \left\langle \| \underline{y} - \underline{\widehat{y}} \|^2 \right\rangle = \left\langle \| \underline{\widehat{\epsilon}} \|^2 \right\rangle = \mathrm{tr} \left\langle \underline{\widehat{\epsilon}}, \underline{\widehat{\epsilon}}^T \right\rangle \tag{D.25c}$$

Claim 22 *The following is true even if homo-spread is violated:*

$$\underbrace{\mathrm{tr} \left\langle \underline{y}, \underline{y}^T \right\rangle}_{SS_{\underline{y}}} = \underbrace{\mathrm{tr} \left\langle \underline{\widehat{y}}, \underline{\widehat{y}}^T \right\rangle}_{SS_{\underline{\widehat{y}}}} + \underbrace{\mathrm{tr} \left\langle \underline{\widehat{\epsilon}}, \underline{\widehat{\epsilon}}^T \right\rangle}_{SS_{res}} \tag{D.26}$$

This is like the Pythagorean Theorem for the magenta right triangle in Fig.D.1.

proof:

From Eqs.D.21 and D.25, we see that

$$SS_{\underline{y}} = \mathrm{tr} \left\langle \underline{y}, \underline{y}^T \right\rangle \tag{D.27}$$

$$SS_{\underline{\widehat{y}}} = \mathrm{tr} \left\langle \underline{\widehat{y}}, \underline{\widehat{y}}^T \right\rangle = \mathrm{tr} \left\langle I_X \underline{y}, \underline{y}^T I_X^T \right\rangle = \mathrm{tr} \left\langle I_X \underline{y}, \underline{y}^T \right\rangle \tag{D.28}$$

$$SS_{res} = \mathrm{tr} \left\langle \underline{\widehat{\epsilon}}, \underline{\widehat{\epsilon}}^T \right\rangle = \mathrm{tr} \left\langle A_X \underline{y}, \underline{y}^T A_X^T \right\rangle = \mathrm{tr} \left\langle A_X \underline{y}, \underline{y}^T \right\rangle \tag{D.29}$$

Now use $I_X + A_X = 1$.

QED

The goodness of fit for this model is often measured using the **coefficient of determination** R^2. R^2 is defined by

$$R^2 = 1 - \frac{SS_{res}}{SS_{\underline{y}}} = \frac{SS_{\underline{\widehat{y}}}}{SS_{\underline{y}}} = \frac{\mathrm{tr} \left\langle \underline{\widehat{y}}, \underline{\widehat{y}}^T \right\rangle}{\mathrm{tr} \left\langle \underline{y}, \underline{y}^T \right\rangle} \tag{D.30}$$

If homo-spread holds, then R^2 reduces to

$$R^2 = \frac{\text{tr } I_X}{nsam} \ . \tag{D.31}$$

See Fig.D.2 for a pictorial explanation of R^2.

Figure D.2: Pictorial explanation of R^2.

D.2 LR, assuming x_σ are random

Let

$i_0 \in \{0, 1, 2, \ldots, n\}$: index that can assume values 0 to n
$i \in \{1, 2, \ldots, n\}$: index that can assume values 1 to n. i is never equal to 0.
$\underline{y} \in \mathbb{R}$: true value of dependent y-variable
$\underline{\widehat{y}} \in \mathbb{R}$: curve-fit of dependent y-variable
$\underline{\epsilon} \in \mathbb{R}$: residual
$\underline{x}_i \in \mathbb{R}$: independent x-variables for $i \in \{1, \ldots, n\}$
$\beta_0, \beta_i \in \mathbb{R}$: regression coefficients

$$\underline{\widehat{y}} = \beta_0 + \sum_{j=1}^{n} \beta_j \underline{x}_j \tag{D.32}$$

$$= \sum_{j_0=0}^{n} \beta_{j_0} \underline{x}_{j_0} \quad \text{(Assume } \underline{x}_0 = 1.\text{)} \tag{D.33}$$

$$\underline{y} = \underline{\widehat{y}} + \underline{\epsilon} \tag{D.34}$$

D.2.1 Transforming from non-random to random x_σ

Define the following population averages:

$$E_\sigma[x_\sigma] = \frac{1}{nsam} \sum_\sigma x_\sigma \ , \tag{D.35}$$

$$E_\sigma[x_\sigma y_\sigma] = \frac{1}{nsam} \sum_\sigma x_\sigma y_\sigma \ , \tag{D.36}$$

$$\langle x_\sigma, y_\sigma \rangle_\sigma = E_\sigma[x_\sigma y_\sigma] - E_\sigma[x_\sigma] E_\sigma[y_\sigma] \ . \tag{D.37}$$

Claim 23 *If the x_σ are i.i.d. random variables,*

$$E_\sigma[x_\sigma] = \langle \underline{x} \rangle \tag{D.38}$$

$$E_\sigma[x_\sigma y_\sigma] = \left\langle \underline{x}\underline{y} \right\rangle \tag{D.39}$$

$$\langle x_\sigma, y_\sigma \rangle_\sigma = \left\langle \underline{x}, \underline{y} \right\rangle \tag{D.40}$$

proof:

$$\frac{1}{nsam} \sum_\sigma x_\sigma = \frac{1}{nsam} \sum_{x \in val(\underline{x})} x \underbrace{\sum_\sigma \mathbb{1}(x_\sigma = x)}_{N(x_\sigma = x)} \tag{D.41}$$

$$= \sum_x x \ P(x) \tag{D.42}$$

$$= \langle \underline{x} \rangle \tag{D.43}$$

$$\frac{1}{nsam} \sum_\sigma x_\sigma y_\sigma = \frac{1}{nsam} \sum_{x \in val(\underline{x})} \sum_{y \in val(\underline{y})} xy \underbrace{\sum_\sigma \mathbb{1}(x_\sigma = x, y_\sigma = y)}_{N(x_\sigma = x, y_\sigma = y)} \tag{D.44}$$

$$= \sum_{x,y} xy \ P(x,y) \tag{D.45}$$

$$= \left\langle \underline{x}\underline{y} \right\rangle \tag{D.46}$$

Eq.(D.40) follows from Eq.(D.38) and Eq.(D.39).
QED

Recall that

$$y_\sigma = \beta_0 + \sum_{j=1}^{n} x_{\sigma j}\beta_j + \epsilon_\sigma \ . \tag{D.47}$$

Assume

$$E_\sigma[x_{\sigma k}\epsilon_\sigma] = E_\sigma[x_{\sigma k}]\underbrace{E_\sigma[\epsilon_\sigma]}_{=0} = 0 \ . \tag{D.48}$$

Then we have

$$E_\sigma[x_{\sigma k}y_\sigma] = E_\sigma[x_{\sigma k}]\beta_0 + \sum_{j=1}^{n} E_\sigma[x_{\sigma k}x_{\sigma j}]\beta_j + \underbrace{E_\sigma[x_{\sigma k}\epsilon_\sigma]}_{=0} \tag{D.49}$$

and

$$E_{\sigma'}[x_{\sigma' k}]E_\sigma[y_\sigma] = E_{\sigma'}[x_{\sigma' k}]\beta_0 + \sum_{j=1}^{n} E_{\sigma'}[x_{\sigma' k}]E_\sigma[x_{\sigma j}]\beta_j + \underbrace{E_{\sigma'}[x_{\sigma' k}]E_\sigma[\epsilon_\sigma]}_{=0} \ . \tag{D.50}$$

Subtracting Eq.(D.50) from Eq.(D.49), we get

$$\langle x_{\sigma k}, y_\sigma\rangle_\sigma = \sum_{j=1}^{n} \langle x_{\sigma k}, x_{\sigma j}\rangle_\sigma \beta_j \ . \tag{D.51}$$

Define the n dimensional covariance matrix C by

$$C_{k,j} = \langle x_{\sigma k}, x_{\sigma j}\rangle_\sigma \ . \tag{D.52}$$

Then Eq.(D.51) implies

$$\beta_j = \sum_{k=1}^{n} C^{-1}_{j,k} \langle x_{\sigma k}, y_\sigma\rangle_\sigma \tag{D.53}$$

for all $j = 1, 2, \ldots, n$.

If we assume that the x_σ are i.i.d., then, by virtue of Claim 23, the matrix C tends to

$$C_{k,j} \to \left\langle \underline{x}_k, \underline{x}_j \right\rangle \tag{D.54}$$

and Eq.(D.53) implies

$$\beta_j = \sum_{k=1}^{n} C^{-1}_{j,k} \left\langle \underline{x}_k, \underline{y} \right\rangle \ . \tag{D.55}$$

D.2.2 LR with random x_σ, expressed in derivative notation

Recall our notation for *conditional* averages.(See sections C.18, C.19) For any random variables $\underline{x}, \underline{y}, \underline{a}$, let

$$E_{|a}[\underline{x}] = \langle \underline{x} \rangle^{|a} \quad \text{(mean)} \tag{D.56}$$

$$\left\langle \underline{x}, \underline{y} \right\rangle^{|a} = \left\langle \underline{x}\underline{y} \right\rangle^{|a} - \langle \underline{x} \rangle^{|a} \left\langle \underline{y} \right\rangle^{|a} \quad \text{(covariance)} \tag{D.57}$$

$$\sigma_{\underline{x}}^{|a} = \sqrt{\langle \underline{x}, \underline{x} \rangle^{|a}} \quad \text{(standard deviation)} \tag{D.58}$$

$$\rho_{\underline{x},\underline{y}}^{|a} = \frac{\left\langle \underline{x}, \underline{y} \right\rangle^{|a}}{\sigma_{\underline{x}}^{|a} \sigma_{\underline{y}}^{|a}} = \left[\frac{\left\langle \underline{x}, \underline{y} \right\rangle}{\sigma_{\underline{x}} \sigma_{\underline{y}}} \right]^{|a} \quad \text{(correlation)} \tag{D.59}$$

$$\partial_{\underline{x}}^{|a} \underline{y} = \left[\frac{\partial}{\partial \underline{x}} \right]^{|a} \underline{y} = \frac{\left\langle \underline{x}, \underline{y} \right\rangle^{|a}}{\langle \underline{x}, \underline{x} \rangle^{|a}} = \rho_{\underline{x},\underline{y}}^{|a} \frac{\sigma_{\underline{y}}^{|a}}{\sigma_{\underline{x}}^{|a}} = \left[\rho_{\underline{x},\underline{y}} \frac{\sigma_{\underline{y}}}{\sigma_{\underline{x}}} \right]^{|a} \quad \text{(partial derivative)} \tag{D.60}$$

"$|a$" means that the variable $\underline{a}$ is held fixed to a when taking all averages.

Recall that

$$\underline{y} = \underbrace{\beta_0 + \sum_{j=1}^{n} \beta_j \underline{x}_j}_{\widehat{\underline{y}}} + \underline{\epsilon} \ . \tag{D.61}$$

Assume

$$\langle \underline{\epsilon} \rangle = 0 \tag{D.62}$$

and

$$\left\langle \underline{x}_j, \underline{\epsilon} \right\rangle = 0 \tag{D.63}$$

for all j.

For $k = 1, \ldots, n$,

$$\left\langle \underline{x}_k, \underline{y} \right\rangle = \sum_{j=1}^{n} \beta_j \left\langle \underline{x}_k, \underline{x}_j \right\rangle \ . \tag{D.64}$$

Define the linear operator

$$\frac{\partial \cdot}{\partial \underline{a}} = \frac{\langle \underline{a}, \cdot \rangle}{\langle \underline{a}, \underline{a} \rangle} \tag{D.65}$$

for any random variable $\underline{a}$. Then Eq.(D.64), after dividing both of its sides by $\langle \underline{x}_k, \underline{x}_k \rangle$, can be written as

$$\frac{\partial \underline{y}}{\partial \underline{x}_k} = \sum_{j=1}^{n} \beta_j \frac{\partial \underline{x}_j}{\partial \underline{x}_k} \tag{D.66}$$

Let $\underline{x}^n$ and β^n be n-dimensional column vectors. If we further define the gradient

$$\nabla_{\underline{x}^n} \underline{y} = \left[\frac{\partial \underline{y}}{\partial \underline{x}_1}, \frac{\partial \underline{y}}{\partial \underline{x}_2}, \ldots, \frac{\partial \underline{y}}{\partial \underline{x}_n} \right]^T \tag{D.67}$$

and the Jacobian matrix

$$J_{j,k} = \frac{\partial \underline{x}_j}{\partial \underline{x}_k} \tag{D.68}$$

then

$$\nabla_{\underline{x}^n} \underline{y} = J^T \beta^n \tag{D.69}$$

so

$$\boxed{\beta^n = (J^T)^{-1} \nabla_{\underline{x}^n} \underline{y}} \tag{D.70}$$

Note that $J_{k,k} = 1$ for all k. Eq.(D.55) and Eq.(D.70) are equivalent. Whereas the matrix C has the nice property that it is symmetric, the matrix J has the nice property that its diagonal entries are 1.

Next, we will write Eq.(D.70) for the special cases $n = 1, 2, 3$, where n is the number of independent x-variables $\underline{x}_j$.

1. $n = 1$ ($\underline{y}$ fitted by a line)

$$\underline{y} = \beta_0 + \beta_1 \underline{x} + \underline{\epsilon} \tag{D.71}$$

Eq.(D.70) becomes

$$\beta_1 = \frac{\partial \underline{y}}{\partial \underline{x}} = \frac{\langle \underline{x}, \underline{y} \rangle}{\langle \underline{x}, \underline{x} \rangle} = \rho_{\underline{x},\underline{y}} \frac{\sigma_{\underline{y}}}{\sigma_{\underline{x}}} \tag{D.72}$$

2. $n = 2$ ($\underline{y}$ fitted by a plane)

$$\underline{y} = \beta_0 + \beta_1 \underline{x}_1 + \beta_2 \underline{x}_2 + \underline{\epsilon} \tag{D.73}$$

Eq.(D.70) becomes[2]

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = (J^T)^{-1} \begin{bmatrix} \partial_{\underline{x}_1} \underline{y} \\ \partial_{\underline{x}_2} \underline{y} \end{bmatrix} \tag{D.74}$$

$$= \frac{1}{\det J^T} \begin{bmatrix} J_{22} & -J_{21} \\ -J_{12} & J_{11} \end{bmatrix} \begin{bmatrix} \partial_{\underline{x}_1} \underline{y} \\ \partial_{\underline{x}_2} \underline{y} \end{bmatrix} \tag{D.75}$$

[2]Recall that if $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ then $M^{-1} = \frac{1}{\det M} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

Hence,

$$\beta_1 = \frac{J_{22}\partial_{\underline{x}_1}\underline{y} - J_{21}\partial_{\underline{x}_2}\underline{y}}{J_{11}J_{22} - J_{21}J_{12}} = \frac{\partial_{\underline{x}_1}\underline{y} - J_{21}\partial_{\underline{x}_2}\underline{y}}{1 - J_{21}J_{12}} \tag{D.76}$$

We can express Eq.(D.76) in terms of variances and correlations as follows.

$$\begin{aligned}
\beta_1 &= \left[\frac{1}{\langle \underline{x}_1, \underline{x}_1\rangle}\right] \frac{\left\langle \underline{x}_1, \underline{y}\right\rangle - \langle \underline{x}_2, \underline{x}_1\rangle \left\langle \underline{x}_2, \underline{y}\right\rangle \langle \underline{x}_2, \underline{x}_2\rangle^{-1}}{1 - \rho^2_{\underline{x}_1,\underline{x}_2}} && \text{(D.77)}\\
&= \left[\frac{1}{\sigma^2_{\underline{x}_1}}\right] \frac{\rho_{\underline{x}_1,\underline{y}}\sigma_{\underline{y}}\sigma_{\underline{x}_1} - \rho_{\underline{x}_1,\underline{x}_2}\rho_{\underline{x}_2,\underline{y}}\sigma_{\underline{x}_1}\sigma_{\underline{y}}}{1 - \rho^2_{\underline{x}_1,\underline{x}_2}} && \text{(D.78)}\\
&= \left[\frac{\sigma_{\underline{y}}}{\sigma_{\underline{x}_1}}\right] \frac{\rho_{\underline{x}_1,\underline{y}} - \rho_{\underline{x}_1,\underline{x}_2}\rho_{\underline{x}_2,\underline{y}}}{1 - \rho^2_{\underline{x}_1,\underline{x}_2}} && \text{(D.79)}
\end{aligned}$$

Eq.(D.79) agrees with the value of $\beta_{YX,Z}$ in Ref.[56] by Pearl, if we replace in Pearl's formulae $X \to \underline{x}_1$, $Y \to \underline{y}$, $Z \to \underline{x}_2$.

Note that Eq.(D.76) can also be written as

$$\begin{aligned}
\beta_1 &= \frac{\partial_{\underline{x}_1}\underline{y} - J_{21}\partial_{\underline{x}_2}\underline{y}}{1 - J_{21}J_{12}} && \text{(D.80)}\\
&= \partial_{\underline{x}_1}\underline{y} + \underbrace{\frac{J_{21}J_{12}\partial_{\underline{x}_1} - J_{21}\partial_{\underline{x}_2}}{1 - J_{21}J_{12}}}_{-A_{\underline{x}_1}}\underline{y} && \text{(D.81)}
\end{aligned}$$

The linear operator $A_{\underline{x}_1}$ satisfies

$$A_{\underline{x}_1}(\underline{x}_1) = 0 \quad (A_{\underline{x}_1} \text{ annihilates } \underline{x}_1) \tag{D.82}$$

and

$$A_{\underline{x}_1}(\underline{x}_2) = J_{21} = \partial_{\underline{x}_1}\underline{x}_2 \tag{D.83}$$

Therefore

$$A_{\underline{x}_1} = \partial^{|x_1}_{\underline{x}_1} \tag{D.84}$$

and

$$\beta_1 = \partial_{\underline{x}_1}\underline{y} - \partial^{|x_1}_{\underline{x}_1}\underline{y} \tag{D.85}$$

If we define

$$I_{\underline{x}_1} = 1 - A_{\underline{x}_1} \tag{D.86}$$

then

$$\beta_1 = \partial_{\underline{x}_1}(I_{\underline{x}_1}\underline{y}) \tag{D.87}$$

3. $n = 3$ ($\underline{y}$ fitted by a volume)

Figure D.3: Bnet for Linear Regression of $\underline{y}$ with a 3 dimensional feature vector $\underline{x} = (\underline{x}_1, \underline{x}_2, \underline{x}_3)$. $\left\langle \underline{x}_j, \underline{\epsilon} \right\rangle = 0$ because the path from $\underline{x}_j$ to $\underline{\epsilon}$ is blocked by a collider node. Note that even though node $\underline{y}$ is deterministic, nodes $\underline{x}_j$ may be probabilistic. Hence, this is only a partial LDEN (LDEN are discussed in Chapter 52)

$$\underline{x}^3 = [\underline{x}_1, \underline{x}_2, \underline{x}_3]^T \tag{D.88}$$

$$\beta^3 = [\beta_1, \beta_2, \beta_3]^T \tag{D.89}$$

$$\underline{y} = [\beta^3]^T \underline{x}^3 + \underline{\epsilon} \tag{D.90}$$

$$J_{i,j} = \frac{\partial \underline{x}_i}{\partial \underline{x}_j} \tag{D.91}$$

$$\beta^3 = (J^T)^{-1} \nabla_{\underline{x}^3} \underline{y} \tag{D.92}$$

$$J^T = \begin{bmatrix} 1 & a_{12} & a_{13} \\ a_{21} & 1 & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} = A \tag{D.93}$$

$$a_{i,j} = \frac{\partial \underline{x}_j}{\partial \underline{x}_i} \tag{D.94}$$

Using Figs.D.4 and D.5, we get

$$\beta_1 = \frac{1}{\det A}\left(\det\begin{bmatrix} 1 & a_{23} \\ a_{32} & 1 \end{bmatrix}\partial_{\underline{x}_1}\underline{y} + \det\begin{bmatrix} a_{13} & a_{12} \\ 1 & a_{32} \end{bmatrix}\partial_{\underline{x}_2}\underline{y} + \det\begin{bmatrix} a_{12} & a_{13} \\ 1 & a_{23} \end{bmatrix}\partial_{\underline{x}_3}\underline{y}\right) \tag{D.95}$$

$$\mathsf{A} \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

Figure D.4: Arbitrary 3 dimensional matrix A

$$\mathsf{A}^{-1} = \frac{1}{|\mathsf{A}|}\begin{bmatrix} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{13} & a_{12} \\ a_{33} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ \\ \begin{vmatrix} a_{23} & a_{21} \\ a_{33} & a_{31} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{13} & a_{11} \\ a_{23} & a_{21} \end{vmatrix} \\ \\ \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{11} \\ a_{32} & a_{31} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}.$$

Figure D.5: Inverse of the 3 dimensional matrix A given by Fig.D.4.

D.2.3 R^2 with random x_σ

Recall that

$$\underline{y} = \underbrace{\beta_0 + \sum_{j=1}^{n} \beta_j \underline{x}_j}_{\widehat{\underline{y}}} + \underline{\epsilon}\ . \tag{D.96}$$

Assume

$$\langle \underline{\epsilon} \rangle = 0 \tag{D.97}$$

and

$$\left\langle \underline{x}_j, \underline{\epsilon} \right\rangle = 0 \tag{D.98}$$

for all j.

$$\widehat{\underline{y}} = \underline{y} - \underline{\epsilon} \tag{D.99}$$

$$\left\langle \underline{\widehat{y}}, \underline{\widehat{y}} \right\rangle = \left\langle \underline{\widehat{y}}, \underline{y} - \underline{\epsilon} \right\rangle \tag{D.100}$$

$$= \left\langle \underline{\widehat{y}}, \underline{y} \right\rangle \tag{D.101}$$

$$\left\langle \underline{y}, \underline{y} \right\rangle = \left\langle \underline{\widehat{y}} - \epsilon, \underline{\widehat{y}} - \epsilon \right\rangle \tag{D.102}$$

$$= \left\langle \underline{\widehat{y}}, \underline{\widehat{y}} \right\rangle + \langle \underline{\epsilon}, \underline{\epsilon} \rangle \tag{D.103}$$

The goodness of fit measure R^2 for this model is defined by

$$R^2_{\underline{y} \sim \underline{\widehat{y}}} = \frac{\left\langle \underline{\widehat{y}}, \underline{\widehat{y}} \right\rangle}{\left\langle \underline{y}, \underline{y} \right\rangle} = 1 - \frac{\langle \underline{\epsilon}, \underline{\epsilon} \rangle}{\left\langle \underline{y}, \underline{y} \right\rangle} \tag{D.104}$$

where we are using Eq.(D.103). By Eq.(D.101), we also have

$$R^2_{\underline{y} \sim \underline{\widehat{y}}} = \frac{\left\langle \underline{y}, \underline{\widehat{y}} \right\rangle}{\left\langle \underline{y}, \underline{y} \right\rangle} = \frac{\partial \underline{\widehat{y}}}{\partial \underline{y}} = \rho_{\underline{y}, \underline{\widehat{y}}} \frac{\sigma_{\underline{\widehat{y}}}}{\sigma_{\underline{y}}} \tag{D.105}$$

$$R^2_{\underline{y} \sim \underline{\widehat{y}}} R^2_{\underline{\widehat{y}} \sim \underline{y}} = \rho^2_{\underline{y}, \underline{\widehat{y}}} \tag{D.106}$$

D.2.4 Regressing Residuals

Recall that

$$\underline{y} = \beta_0 + \sum_{j=1}^{n} \underline{x}_j \beta_j + \underline{\epsilon} \ . \tag{D.107}$$

Therefore,

$$\left\langle \underline{x}_i, \underline{y} \right\rangle = \sum_{j=1}^{n} \left\langle \underline{x}_i, \underline{x}_j \right\rangle \beta_j \tag{D.108}$$

$$= \langle \underline{x}_i, \underline{x}_i \rangle \beta_i + \sum_{j=1}^{n} \mathbb{1}(j \neq i) \left\langle \underline{x}_i, \underline{x}_j \right\rangle \beta_j \ . \tag{D.109}$$

Hence,

$$\beta_i = \frac{\left\langle \underline{x}_i, \underline{y} \right\rangle}{\langle \underline{x}_i, \underline{x}_i \rangle} - \sum_{j=1}^{n} \mathbb{1}(j \neq i) \frac{\left\langle \underline{x}_i, \underline{x}_j \right\rangle}{\langle \underline{x}_i, \underline{x}_i \rangle} \beta_j \ . \tag{D.110}$$

Eq.(D.110) can be expressed in derivative notation as:

$$\beta_i = \frac{\partial \underline{y}}{\partial \underline{x}_i} - \sum_{j=1}^{n} \mathbb{1}(j \neq i) \frac{\partial \underline{x}_j}{\partial \underline{x}_i} \beta_j \tag{D.111}$$

Note that, because of the linearity of the derivative operator, Eq.(D.111) implies:

$$\begin{aligned} \beta_i &= \frac{\partial}{\partial \underline{x}_i} \left(\underline{y} - \underbrace{\sum_{j=1}^{n} \mathbb{1}(j \neq i) \underline{x}_j \beta_j}_{\underline{y} - \underline{x}_i \beta_i - \beta_0 - \underline{\epsilon}} \right) && \text{(D.112)} \\ &= \left[\partial_{\underline{x}_i} - \partial_{\underline{x}_i}^{|x_i} \right] \underline{y} \ . && \text{(D.113)} \end{aligned}$$

Eq.(D.113) implies that if we know $(\widehat{\beta}_j)_{j \in \{1,2,\ldots,n\} - \{i\}}$, we can regress the residual $\underline{y} - \sum_{j \neq i} \underline{x}_j \widehat{\beta}_j$ on x_i, to get an estimate of $\widehat{\beta}_i$. For example, in the case $n = 2$, if we know $\widehat{\beta}_1$, we can regress the residual $\underline{y} - \underline{x}_1 \widehat{\beta}_1$ on $\underline{x}_2$ to get an estimate of $\widehat{\beta}_2$.

D.3 Logistic Regression (LoR)

Suppose $x_\sigma \in \mathbb{R}^n$, $y_\sigma \in \mathbb{R}$, and Σ is a population of individuals σ. In general, a **regression** is when we curve-fit a dataset $\{(x_\sigma, y_\sigma) : \sigma \in \Sigma\}$ with a function $\widehat{y} = f(x)$. In **Linear Regression (LR)**, which we discussed earlier, $f(x)$ is a hyperplane in x. On the other hand, in **Logistic Regression (LoR)**, $y_\sigma \in [0, 1]$ and $f(x)$ is the sigmoid of a hyperplane in x.

More specifically, for LR we have Eq.(D.1) which reads as follows:

$$y_\sigma = \underbrace{\beta_0 + \sum_{i=1}^{n} x_{\sigma i} \beta_i}_{\widehat{y}_\sigma} + \epsilon_\sigma \quad \text{(LR)} \ . \tag{D.114}$$

For LoR, we have instead

$$p_\sigma = \text{smoid} \left(\beta_0 + \sum_{i=1}^{n} x_{\sigma i} \beta_i + \epsilon_\sigma \right) \quad \text{(LoR)} \ , \tag{D.115}$$

or, equivalently,

$$\underbrace{\text{lodds}(p_\sigma)}_{\ln \frac{p_\sigma}{1 - p_\sigma}} = \underbrace{\beta_0 + \sum_{i=1}^{n} x_{\sigma i} \beta_i}_{\widehat{y}_\sigma} + \epsilon_\sigma \quad \text{(LoR)} \ , \tag{D.116}$$

where we have used the fact that lodds() is the inverse function of smoid(). Hence, an LoR fit can be calculated by collecting a dataset $\{(x_\sigma, p_\sigma) : \sigma \in \Sigma\}$, transforming that dataset to the dataset $\{(x_\sigma, \text{lodds}(p_\sigma)) : \sigma \in \Sigma\}$, and fitting the latter dataset with a

hyperplane. Let $P(\underline{b}_\sigma = 1) = p_\sigma \in [0, 1]$. LoR can be used for binary classification if we define the binary class variable $c_\sigma \in \{0, 1\}$ by

$$c_\sigma = \mathbb{1}(P(\underline{b}_\sigma = 1) > \alpha) \tag{D.117}$$

for some $0 < \alpha < 1$.

Appendix E

Classic Bayesian Network Apps

The figures in this chapter were generated using the free, open source app "PyAgrum" (see Ref.[2]) [1]

Whenever I introduce the subject of Bayesian Networks to someone who has never met them before, I recommend that the first thing they do is to download from the internet a "classic" bnet app such as PyAgrum[2], and play with it. It's a sure way of getting hooked for life. That's how I got hooked, by an early bnet application called Ergo. As you can see from the pictures in this chapter, the visual output generated by such applications is very "explainable", intuitive and appealing.

In the late 1980's and early 1990's, partially fueled by the invention of the junction tree algorithm (JTA) (see Chapter 46), there occurred much free and commercial software writing activity related to Bnets. Bill Gates was a fan of bnets at that time, and he dedicated a lot of Microsoft manpower to do R&D of bnets. The first version of Clippy and the first XBox recommender were in fact Bayesian Networks. PyAgrum is a modern version of those "classic" 1990's apps. It implements many of the same functions (most notably, the JTA) as its predecessors but with a modern Python API and a C++ engine. Other non-classic types of causal AI apps have arisen since the 1990's[3] but PyAgrum fills a big gap in the availability of classic open source bnet apps.[4]

Here are a few of these addictive visuals generated by PyAgrum for the simple diamond shaped "Wet Grass bnet".

To use PyAgrum, one first enters the input data consisting of the structure of

[1] "agrum" means citric fruit in French.

[2] Alternative choices for bnet apps are Netica by www.norsys.com, or Hugin by www.hugin.com and several others.

[3] Examples of Causal AI apps of a non-classic type: Marco Scutari's bnlearn and my JudeasRx, DAG_Lie_Detector, Mappa_Mundi, SCuMpy, SentenceAx, CausalFitBit, texnn.

[4] PyAgrum is similar to my earlier Python app Quantum Fog (QFog). Both implement the JTA, but PyAgrum has more features and is more polished. PyAgrum's engine is in C++ and QFog's is in the slower Python. PyAgrum implements influence diagrams, dynamic bnets and do-calculus, which QFog doesn't, but QFog implements the JTA for both classical and quantum physics, whereas PyAgrum does not do quantum stuff.

the bnet and a TPM (Transition Probability Matrix)[5] for each node of the bnet.

Figure E.1: Structure (network, DAG) and a TPM for each node, of the Wet Grass bnet.

When asked to display the input data, PyAgrum shows Fig.E.1.

When asked to "infer" the probabilities of unobserved nodes $\underline{r} = Rain$ and $\underline{w} = WetGrass$, assuming that as evidence, we observe that $\underline{c} = Cloudy = 0$ and $\underline{s} = Sprinkler = 1$, PyAgrum shows Fig.E.2.

When asked to redraw the WetGrass DAG so that each arrow's thickness is proportional to the Mutual Information (a measure of correlation) between the two nodes connected by the arrow, PyAgrum shows Fig.E.3.

[5] A TPM is also called a CPT, which stands for Conditional Probability Table

Figure E.2: Probability histograms for every node of the Wet Grass bnet, assuming that as evidence, we observe that $\underline{c} = Cloudy = 0$ and $\underline{s} = Sprinkler = 1$, i.e. the sprinkler was left ON and the day was NOT cloudy.

PyAgrum can also

- consider hard or soft evidence (or no evidence) for each node. **Hard evidence** means that only one state of the node is observed whereas **soft evidence** means that each state of the node occurs with some probability.
- calculate joint probability distributions and conditional joint probability distributions for multiple nodes of a bnet. For example, it can calculate $P(a, b|c, d)$ if $\underline{a}, \underline{b}, \underline{c}, \underline{d}$ are 4 distinct nodes of the bnet.
- find the most likely value for each node, i.e., find $x^*. = \underset{x.}{\operatorname{argmax}} P(x.)$ where $\underline{x}.$ contains all the nodes of the bnet. $x.^*$ is called the **most probable explanation**.
- calculate the Markov blanket for any node (See Chapter 54)
- draw influence diagrams and do associated calculations (see Chapter 42)
- calculate conditional probabilities and adjustment formulae that involve multiple do operators in the condition part. Hence, it does do-calculus (see Chapter 22)

Figure E.3: Wet Grass bnet drawn so that each arrow's thickness is proportional to the Mutual Information (a measure of correlation) between the two nodes connected by the arrow.

- do causal DAG discovery (what used to be called structure learning) (see Chapter 95)
- calculate dynamic Bayesian networks (see Chapter 26)
- calculate information theoretic quantities related to a bnet (see Section C.30)

and much more! Numerous examples are included in the PyAgrum repo, from The Book of Why (see Ref.[66]), Kaggle's Titanic dataset, and other sources.

Appendix F

Definition of a Bayesian Network

A **directed graph** $G = (V, E)$ consists of two sets, V and E. V contains the **vertices (nodes)** and E contains the **edges (arrows)**. An arrow $\underline{a} \to \underline{b}$ is an ordered pair $(\underline{a}, \underline{b})$ where $\underline{a}, \underline{b} \in V$.

The **parents** of a node $\underline{x}$ are those nodes $\underline{a}$ such that there are arrows $\underline{a} \to \underline{x}$. The **children** of a node $\underline{x}$ are those nodes $\underline{b}$ such that there are arrows $\underline{x} \to \underline{b}$. A **root node** is a node with no parents. A **leaf node** is a node with no children. The **neighbors** of a node $\underline{x}$ is the set of parents and children of $\underline{x}$.

A **path** is a set of nodes that are connected by arrows, so that all nodes have 1 or 2 neighbors, but only two nodes (**open path**) or zero nodes (**closed path**) have only one neighbor. A **directed path** is a path in which all the arrows point in the same direction. A **loop** is a closed path;i.e., a path in which all nodes have exactly 2 neighbors. A **cycle** is a directed loop. A **Directed Acyclic Graph (DAG)** is a directed graph that has no cycles.

A **fully connected directed graph** is a directed graph in which every node has all other nodes as neighbors. Figs.F.1 and F.2 show 2 different ways of drawing the same directed graph, a fully connected graph with 4 nodes. Note that a convenient way to label the nodes of a fully connected directed graph with N nodes is to point arrows from $\underline{x}_k$ to $\underline{x}_j$ where $j = 0, 1, 2, \ldots, N-1$ and $k = j-1, j-2, \ldots, 0$.

Figure F.1: Fully connected directed graph with 4 nodes, drawn as a line.

A **connected graph** is a graph for which there is no way of separating the nodes into two sets so that there is no arrow from one set to the other. A **tree** is a directed graph in which all nodes have a single parent except for a single node called the "root" node which has no parents. A **polytree** is a DAG with no loops.

A **Bayesian network (bnet)** consists of a DAG and a **Transition Probability Matrix (TPM)** associated with each node of the graph. A TPM is often

Figure F.2: Fully connected directed graph with 4 nodes, drawn as a square.

called a **Conditional Probability Table (CPT)**. The **structure** of a bnet is its DAG alone, sans the TPMs. The **skeleton** of a bnet is the undirected graph beneath the bnet's DAG.

In this book, random variables are indicated by underlined letters and their values by non-underlined letters. We use $val(\underline{x})$ to denote the set of states (i.e., values) that a random variable $\underline{x}$ can assume. Each node of a bnet is labeled by a random variable. Thus, $\underline{x} = x$ means that node $\underline{x}$ is in state x.

Some sets of nodes associated with each node $\underline{a}$ of a bnet

- $ch(\underline{a})$ = children of $\underline{a}$.
- $pa(\underline{a})$ = parents of $\underline{a}$.
- $nb(\underline{a}) = pa(\underline{a}) \cup ch(\underline{a})$ = neighbors of $\underline{a}$.
- $de(\underline{a}) = \cup_{n=1}^{\infty} ch^n(\underline{a}) = ch(\underline{a}) \cup ch \circ ch(\underline{a}) \cup \ldots$, descendants of $\underline{a}$.
- $an(\underline{a}) = \cup_{n=1}^{\infty} pa^n(\underline{a}) = pa(\underline{a}) \cup pa \circ pa(\underline{a}) \cup \ldots$, ancestors of $\underline{a}$.

In this book, we will use $\underline{a}.$ to indicate a **multi-node (node set, node array)** $\underline{a}. = (\underline{a}_j)_{j=0,1,\ldots,na-1}$. We will often treat multinodes as if they were sets, and combine them with the usual set operators. For instance, for two multinodes $\underline{a}.$ and $\underline{b}.$, we define $\underline{a}.\cup\underline{b}.$, $\underline{a}.\cap\underline{b}.$, $\underline{a}.-\underline{b}.$ and $\underline{a}. \subset \underline{b}.$ in the obvious way. We will indicate a singleton set (single node multi-node) $\underline{a}. = \{\underline{a}\}$ simply by $\underline{a}. = \underline{a}$. For instance, $\underline{a}.-\underline{b} = \underline{a}.-\{\underline{b}\}$.

The TPM of a node $\underline{x}$ of a bnet is a matrix of probabilities $P(\underline{x} = x|pa(\underline{x}) = a.)$, where $x \in val(\underline{x})$ and $a. \in val(pa(\underline{x}))$.

A **deterministic node** is a node such that its TPM is of the form

$$P(\underline{x} = x|pa(\underline{x}) = a.) = \delta(x, f(a.)) \tag{F.1}$$

for some function $f : val(pa(\underline{x})) \to val(\underline{x})$, where $\delta(x, y)$ is the Kronecker delta function.

A bnet with nodes $\underline{x}.$ represents a probability distribution

$$P(x.) = \prod_j P(\underline{x}_j = x_j|(\underline{x}_k = x_k)_{k:\underline{x}_k \in pa(\underline{x}_j)}) \ . \tag{F.2}$$

Note that for a fully connected bnet with N nodes, Eq.(F.2) becomes

$$P(x.) = \prod_{j=0}^{N-1} P(x_j|(x_k)_{k=j-1,j-2,\ldots,0}) \ . \tag{F.3}$$

For example, if $N = 4$, Eq.(F.3) becomes

$$P(x_0, x_1, x_2, x_3) = P(x_3|x_2, x_1, x_0)P(x_2|x_1, x_0)P(x_1|x_0)P(x_0) \ . \tag{F.4}$$

We see that Eq.(F.3) is just the chain rule for conditional probabilities.

Of course, not all bnets are fully connected. So what determines whether in a bnet with nodes $\underline{x}$ and $\underline{y}$, we draw or not draw, an arrow connecting the two nodes?

Recall that two random variables $\underline{x}, \underline{y}$ are **(probabilistically) independent** (denoted by $\underline{x} \perp \underline{y}$) if $P(y|x) = P(x)$. Let us denote the correlation between $\underline{x}$ and $\underline{y}$ by $\left\langle \underline{x}, \underline{y} \right\rangle = \left\langle \underline{x}\underline{y} \right\rangle - \langle \underline{x} \rangle \left\langle \underline{y} \right\rangle$. It's easy to show that $\underline{x} \perp \underline{y}$ implies $\left\langle \underline{x}, \underline{y} \right\rangle = 0$. However, the converse is not true: it's possible for $\underline{x}$ and $\underline{y}$ to be uncorrelated but dependent. For example, if $\underline{y} = \underline{x} - \langle \underline{x}, \underline{x} \rangle$, we get $\left\langle \underline{x}, \underline{y} \right\rangle = 0$, but $P(y|x) \neq P(x)$. However, $\underline{x} \perp \underline{y}$ if and only if $\left\langle f(\underline{x}), g(\underline{y}) \right\rangle$ for all functions f, g.

Consider the bnet $\underline{x} \to \underline{m} \to \underline{y}$. In this case, $\underline{x}$ and $\underline{y}$ are dependent, but there is no arrow connecting them. Henceforth, we will say that there is a **direct dependence** between nodes $\underline{x}$ and $\underline{y}$ if there is an arrow connecting them. Note that $\underline{x}$ and $\underline{y}$ are not directly dependent iff[1] $P(y|do(x)) = P(y)$ (i.e., no $\underline{x} \to \underline{y}$) and $P(x|do(y)) = P(x)$ (i.e., no $\underline{y} \to \underline{x}$). $\underline{x}$ and $\underline{y}$ can be dependent but not directly dependent.

In this book, we use the following conventions for bnet diagrams:

Random variables are underlined and their values are not. For example, $\underline{a} = a$ means the random variable $\underline{a}$ takes the value a. A diagram with all its nodes underlined represents a Bayesian Network (bnet), whereas the same diagram with the letters not underlined represents a specific **instantiation** of that bnet. For example $\underline{a} \to \underline{b} \to \underline{c}$ represents the bnet with full probability distribution $P(c|b)P(b|a)P(a)$, whereas $a \to b \to c$ represents $P(c|b)P(b|a)$. Note that, for convenience, we define $a \to b \to c$ to exclude the priors of root nodes such as $P(a)$.

If $\underline{a}$ is a root node, then $\mathbb{E}a$ signifies a weighted sum $\sum_a P(a)$. For example,

$$\mathbb{E}a \to b \to c = \sum_a P(c|b)P(b|a)P(a) \tag{F.5}$$

If $\underline{a}$ is not a root node, then $\sum a$ signifies a simple unweighted sum $\sum_a$. For example,

$$x \to \sum a \to y = \sum_a P(y|a)P(a|x) \tag{F.6}$$

[1] As will be defined in Chapter 12, $P(y|do(x))$ equals $P(y|x)$ for the bnet in which all arrows entering node $\underline{x}$ have been amputated and node $\underline{x}$ has been set to state x.

Two bnets are said to be equal if their full probability distributions (i.e., their full instantiations) are equal numerically. For example,

$$\underline{a} \rightarrow \underline{b} \rightarrow \underline{c} = P(c|b)P(b|a)P(a) = \underline{a} \leftarrow \underline{b} \leftarrow \underline{c} \tag{F.7}$$

Unobserved (a.k.a. hidden, latent) nodes are indicated in a bnet by enclosing their label in a dashed circle. For example, $\overset{\frown}{\underset{\smile}{u}}$. Alternatively, they are indicated by using dashed arrows for all arrows emanating from the unobserved node.

In Chapter 37, we define a measure of goodness (G) of causal fit (CF) for each bnet of a finite set of bnets $\mathcal{G}$.

Given a dataset of samples for the random variables $(\underline{x}_i)_{i=0,1,\ldots,N-1}$, and a finite set of possible bnets $\mathcal{G}$, set $\mathcal{G}$ may contain several bnets (differing in the direction of some arrows) that fit the data well causally. However, the one with the highest GCF is most likely to be used by Nature. In this book, we will refer to that single one as the **best CF bnet**. We will also refer to a bnet in $\mathcal{G}$ that has a high (resp., low) value of GCF, though not necessarily the highest (resp., lowest), as a **good (resp., bad) CF bnet**.[2]

It's important to realize that bnets that are not a good CF are far from useless; they are frequently used as intermediate calculational tools. They are incorrect causally, but they aren't incorrect numerically. For instance, in Bayes rule, we switch from a good CF bnet $P(x|\theta)P(\theta) = \underline{x} \leftarrow \underline{\theta}$ to a bad CF bnet $P(\theta|x)P(x) = \underline{x} \rightarrow \underline{\theta}$, where x is the data and θ are the parameters.

[2]In this book, the term "causal bnet" will mean a good CF bnet. Pearl is fond of using the term "causal bnet", but he uses it in a different sense that does not allude to a measure of goodness of causal fit.

Appendix G

Bayesian Networks, Causality and the Passage of Time

This chapter is based on a blog post (see Ref.[87]) from my blog "Quantum Bayesian Networks".

G.1 Unifying Principle of this book

The unifying principle of this book is Bayesian Networks (bnets). The main goal of this book is to explain as much of Artificial Intelligence (AI) and Machine Learning (ML) as possible using bnets.

Bayesian Networks are a graphical representation of the chain rule for conditional probabilities. They are not a "heuristic algorithm" like XGBoost or Neural Nets. They are a very simple, intuitive, basic and general definition. I would say that the definition of a Bayesian Network is as important to Probability Theory as the definition of a Group is to Abstract Algebra. Algebraic groups are never going to go out of fashion and neither are B nets.

An Artificial Neural Net can be defined (see Chapter 68) as a Bayesian Network with a layered structure, and such that all its nodes are deterministic[1]. A decision tree can be easily converted (see the Chapter 16) into a B net that has the same tree structure. A transformer net can also be converted (see Chapter 104) into a bnet. The SCM diagrams favored by Pearl (see Chapter 52) are just bnets whose internal nodes are deterministic and external ones are probabilistic. In fact, as I show in this book, quite a few AI methods can be understood in terms of B nets—just like many theorems in Abstract Algebra can be understood in terms of groups.

[1]Neural Nets (NNs) are DAGs, but they contain a lot of spurious arrows whose direction or very existence has no causal motivation, and they could be missing other arrows which would have a causal motivation. So I like to say that NNs are acausal DAGs.

G.2 You say tomato, I say tomato

In this chapter, I will use the terms Bayesian Network (bnet), causal model and DAG as if they were synonymous. My justification for doing this is as follows.

A Bayesian Network is a DAG + probability tables. One can easily compute the probability tables from DAG + Dataset. Therefore,

You say DAG+Dataset, I say Bayesian Network.

The use of the terms "causal model" and "DAG", as an alternative to the term "Bayesian Network", has become popular in the last decade among economists, epidemiologists, AI researchers and even Judea Pearl himself. It seems some people think "causal models" and "DAGs" are revolutionary, whereas Bayesian Networks are a concept that was tried 25 years ago, and has been replaced since then by stuff that works better. But any time you have a Dataset, which is almost always true in practice in Economics, Epidemiology and AI, a DAG implies a Bayesian Network and vice versa.

G.3 A dataset is causal model free

Time and time again, Judea Pearl makes the point on Twitter to neural net advocates that they are trying to do a provably impossible task, to derive a causal model from data. I could be wrong, but this is what I think he means.

When Pearl says "data", he is referring to what is commonly called a dataset. A **dataset** is a table of data, where all the entries of each column have the same units, and measure a single feature, and each row refers to one particular sample or individual. Datasets are particularly useful for estimating probability distributions and for training neural nets. When Pearl says a "causal model", he is referring to a DAG (directed acyclic graph) or a bnet (Bayesian Network= DAG + probability table for each node of DAG).

Sure, you can try to derive a causal model from a dataset, but you'll soon find out that you can only go so far.

The process of finding a partial causal model from a dataset is called structure learning (SL). SL can be done quite nicely with Marco Scutari's open source program bnlearn. There are 2 main types of SL algorithms: score-based and constraint based. The first and still very competitive constraint-based SL algorithm was the Inductive Causation (IC) algorithm proposed by Pearl and Verma in 1991. So Pearl is quite aware of SL. The problem is that SL often cannot narrow down the causal model to a single one. It finds an undirected graph (UG), and it can determine the direction of some of the arrows in the UG, but it is often incapable, for well understood fundamental —not just technical— reasons, of finding the direction of ALL the arrows of the UG. So it often fails to fully specify a DAG model.

Let's call the ordered pair (dataset, causal model) a dataset++ . Then what I believe Pearl is saying is that a dataset is causal model-free or causal model-less

(although sometimes one can find a partial causal model hidden in there). A dataset is not a dataset++.

Caveat to this section: Define a **time-series table (TST)** to be a table of data, where all the entries of each column have the same units, and measure a single feature at different times with time increasing down the table. Hence, the rows of a TST are chronologically ordered (they specify a time series) whereas those of a dataset aren't. Whereas it is not possible to fully specify a DAG from a dataset alone, it is possible to do so from a TST. See the python app CausalFit (Ref.[89]) for a possible way extracting a causal DAG from a Fitbit TST.

G.4 What is causality?

What is Causality, really, and how do Bayesian Networks (a.k.a. Causal Models, DAGs) encode it? For me, Causality is a time-induced ordering between two events, the transmission of information (and its accompanying energy) from the earlier of the two events to the later one, and the physical response of the later event to the reception of that information.

Note that this definition of causality does not mention correlation. It is often assumed that even though correlation does not imply causation, causation implies correlation. But the latter statement is false; there are scenarios, albeit unusual, "fine tuned" ones, in which there is causation without correlation. For example, consider a bnet with arrows $\underline{x} \to \underline{y}$ and $\underline{x} \to \underline{c} \to \underline{y}$. When we amputate the arrows entering $\underline{x}$, a dependence between $\underline{x}$ and $\underline{y}$ persists, so we say $\underline{x}$ causes $\underline{y}$. Even though $\underline{x}$ causes $\underline{y}$, it's possible to tune the probabilities of the bnet so that the effect of the path $\underline{x} - \underline{c} - \underline{y}$ and the effect of the direct path $\underline{x} - \underline{y}$ cancel each other out and produce zero correlation between $\underline{x}$ and $\underline{y}$. As a trivial example, suppose $\underline{c} = 2\underline{x}$, $\underline{y} = \underline{c} - 2\underline{x} = 0$. Since $\underline{y} = 0$, it's uncorrelated with $\underline{x}$.

The nodes of a bnet represent random variables. Some of those random variables are clearly events (i.e., they occur at a definite time). For example, let D=0 if a patient is not given a drug, D=1 if he/she is given it. D occurs at a definite time. But other random variables represent qualities which do not occur at a definite time. For example, G=gender=male, female. G does not occur at a definite time. But even in the case of a quality like G, its value is first decided at birth, so one can ascribe to G a particular, albeit fuzzy time interval during which it is decided. If M=0(single), 1(married), then we can assign to M the day of the marriage. Both the time interval assigned to G and to M are somewhat ambiguous, but still, most people would say that G occurs before M (if a marriage occurs at all). Saying the opposite, that M occurs before G, seems pretty hard to understand. If two nodes A and B of a bnet have time intervals ascribed to them such that the time interval of A does not clearly occur before or after the time interval of B, and if also there is a large causal correlation between A and B, then it probably does not matter much whether one draws an arrow from A to B, or the opposite.

Now that we understand that the arrows in a bnet really do encode the direction of time, it becomes clear why a dataset does not fully specify a bnet. By a dataset (think of a dataframe in Pandas or R), I mean an array of numbers where the columns refer to features and the rows refer to individuals in a population. The column labels of the dataset become the node names of the bnet. Nowhere in a dataset is there any indication of the time ordering of the features. Hence, it's impossible to create, from a dataset alone, a bnet, because bnets do carry such time-ordering information.

Chapter 39 discusses Granger Causality (GC). The critics of GC point out that it assumes, somewhat erroneously, that if event A precedes B and the two events are correlated, A must cause B. I agree. Most roosters crow in response to the stimulus of the sunrise light. A rooster could crow before sunrise if, for example, he had an alarm clock that woke him up 30 minutes before sunrise, but such cases are uncommon, and seem to involve other intermediate events. The moral is that time ordering and correlation are not sufficient conditions for causality. To establish causality with more certainty, one also needs a pinch of prior expert knowledge, or one must gain that expert knowledge through "do" operator experimentation.

G.5 Bayesian Networks and the passage of time

Now that we understand that a bnet's arrows are encoding roughly the passage of time, it becomes possible to glean from this insight a simple method, which, although not very rigorous, is really helpful to me. I will illustrate said method with the famous "Asia" bnet in Fig.G.1. In this bnet, all nodes have two possible values, 0 and 1.

Figure G.1: Asia bnet. Dyspnea=trouble breathing

Given a dataset for this bnet, one can calculate the correlation between every 2 features of the dataset. The feature names become the node names, and links are drawn between any 2 nodes whose correlation is causal and greater in absolute value to some threshold value. This gives an undirected graph that can be obtained

from bnet Fig.G.1 by erasing the directions of the arrows. So how can we guess the directions of the arrows? Well, one uses a little bit of "expert knowledge" to conclude that

time(Visited Asia) < time(Tuberculosis) < time(Or) < time(X-Ray, Dyspnea)

Also

time(smokes) < time(LungCancer, Bronchitis) < time(Or) < time(Dyspnea)

If time(A) < time(B), then A → B. Like I said before, the times we ascribe to these events are somewhat fuzzy and open to debate, so this algorithm is far from being rigorous. But often, saying that time(A)<time(B) makes much more sense than saying that time(B)<time(A). When in doubt about the best direction to give to an arrow of an undirected graph, I recommend calculating a Goodness of Causal Fit metric (see Chapter 37) which makes use of "do" operator experimentation.

A dataset cannot fully specify a bnet because it lacks time ordering info. A dataset also cannot do the harder task of specifying a bnet that is a good causal fit to the problem, because it lacks time ordering info AND prior expert knowledge AND expert knowledge gained from posterior "do" operator experimentation.

G.6 Advice for the DAG-phobic

DAGs are your friends. DAGs should be easy and fun to dream up. After all, I am convinced that DAGs are an integral part of how humans think, so they should come naturally to us. Nevertheless, many people are scared of, or detest, DAGs. I think it's because they fail to grasp the following 3 things:

1. DAGs are not unique. Stop thinking that you have to find the unique DAG for the situation being considered. You just have to find a DAG that is a good causal fit for the situation. If a DAG is too complicated, you can always simplify it by merging several nodes into a single more abstract one, or by summing over unwanted nodes.

2. The nodes of a DAG are roughly ordered from past to present. The arrows of a DAG roughly reflect the passage of time.

3. DAGs represent scientific hypotheses that can and should be tested with do experiments. Causal Inference is an application of the scientific method, which consists of the following steps: formulate hypothesis (DAG), devise experiment to test it, test it.

Chapter 1

AdaBoost

This chapter is based on Ref.[105].

Adaptive Boosting (AdaBoost) is a method of constructing a strong classifier function as a linear combination of an ensemble of weak classifier functions.

Below, we will abbreviate "ensemble classifiers" by "e-classifiers" and "weak classifiers" by "w-classifiers".

Chapter 16 defines decision trees (dtrees) and explains how to construct them. A **tree stump** is a dtree with only one parent and 2 children nodes. Usually the w-classifier functions for AdaBoost come from tree stumps (because tree stumps are w-classifiers and simple to compute), but the core AdaBoost algorithm is oblivious to where the w-classifier functions came from.

Boosting (see this chapter on AdaBoost and Chapter 111 on XGBoost) and bagging (see Chapter 80 on Random Forest) are two methods of building a classifier function from an ensemble of classifier functions. These two methods are most commonly applied to dtrees: Boosting for an ensemble of small dtrees, and Bagging for a random forest (which is an ensemble of dtrees that are usually much more complicated than small dtrees).

1.1 AdaBoost for general ensemble of w-classifiers

In this section we discuss the core AdaBoost algorithm, valid for a generic ensemble of w-classifiers.

Let $L = [0, 1, 2, \ldots, nsam-1]$ be a list of individuals (samples) in a population. In this chapter, we will use the notation $A^\sigma = A[\sigma]$ and $\vec{A} = [A^\sigma : \sigma \in L]$ for a list (vector, 1-D array) indexed by L. We will refer to $DS = (\vec{x}, \vec{y})$ where $x^\sigma \in val(\underline{x})$, $y^\sigma \in val(\underline{y})$, as a dataset. Let $T = \{0, 1, \ldots, nt-1\}$. Let $x^\sigma = (x_0^\sigma, x_1^\sigma, \ldots, x_{nt-1}^\sigma) \in val(\underline{x}_0) \times val(\underline{x}_1) \times \ldots \times val(\underline{x}_{nt-1}) = val(\underline{x})$.

AdaBoost assumes that the classifier class set $val(\underline{y})$ and all the feature sets $val(\underline{x}_t)$ are binary: $val(\underline{y}) = val(\underline{x}_t) = \{-1, 1\}$ for all $t \in T$.

Suppose that we are given an ensemble of nt **w-classifiers** $Y_t : val(\underline{x}) \to \{-1, 1\}$, where $t \in T$. Suppose we want to find **intermediate e-classifiers** $\mathcal{Y}_t$:

$val(\underline{x}) \to \mathbb{R}$ given by

$$\mathcal{Y}_t(x^\sigma) = \sum_{t'=0}^{t} \alpha_{t'} Y_{t'}(x^\sigma) \in \mathbb{R} \tag{1.1}$$

for $t \in T$ and a **final e-classifier** given by

$$\mathcal{Y}_{fin}(x^\sigma) = \text{sign}(\mathcal{Y}_{nt-1}(x^\sigma)) \in \{-1, 1\} \ . \tag{1.2}$$

The AdaBoost algorithm yields a set of coefficients α_t for which the final e-classifier is much stronger (i.e., less error prone) than any of the w-classifiers of the ensemble.

Note that

$$\mathcal{Y}_t(x^\sigma) = \mathcal{Y}_{t-1}(x^\sigma) + \alpha_t Y_t(x^\sigma) \tag{1.3}$$

for $t \in T$ if we define $\mathcal{Y}_{-1} = 0$. Hence

$$\underbrace{e^{-y^\sigma \mathcal{Y}_t(x^\sigma)}}_{Z_t w_{t+1}^\sigma} = \underbrace{e^{-y^\sigma \mathcal{Y}_{t-1}(x^\sigma)}}_{Z_{t-1} w_t^\sigma} e^{-\alpha_t y^\sigma Y_t(x^\sigma)} \tag{1.4}$$

where the **weights** w_t^σ and the **partition function** Z_t are defined by

$$w_{t+1}^\sigma = \begin{cases} 1/nsam & \text{for } t = -1 \\ \frac{\exp(-y^\sigma \mathcal{Y}_t(x^\sigma))}{Z_t} & \text{for } t \geq 0 \end{cases} \tag{1.5}$$

and

$$Z_t = \sum_\sigma e^{-y^\sigma \mathcal{Y}_t(x^\sigma)} \ . \tag{1.6}$$

Note that the probability distribution $P(\sigma|t+1) = w_{t+1}^\sigma$ of weights at time $t+1$ emphasizes (i.e., gives higher probability to) the errors (i.e., occurrences of $y^\sigma \mathcal{Y}_t(x^\sigma) = -1$ for some population individual σ) of the previous (i.e., at time t) intermediate e-classifier $\mathcal{Y}_t$. In other words, every new intermediate e-classifier $\mathcal{Y}_{t+1}$ concentrates on those individuals σ on which the previous e-classifier $\mathcal{Y}_t$ performed poorly.

Note also that the partition function Z_t is a good measure of the **classification error** (i.e., occurrences of $y^\sigma \mathcal{Y}_t(x^\sigma) = -1$) of $\mathcal{Y}_t$. We will therefore use Z_t for that purpose, as an error measure.

For $t > 1$, we have

$$\begin{aligned} Z_t &= \sum_\sigma e^{-y^\sigma \mathcal{Y}_t(x^\sigma)} && (1.7) \\ &= \sum_\sigma \underbrace{e^{-y^\sigma \mathcal{Y}_{t-1}(x^\sigma)}}_{Z_{t-1} w_t^\sigma} e^{-\alpha_t y^\sigma Y_t(x^\sigma)} && (1.8) \\ &= Z_{t-1} E_\sigma[e^{-\alpha_t y^\sigma Y_t(x^\sigma)}] && (1.9) \end{aligned}$$

Define the **success rate** by

$$S_t = \sum_\sigma w_t^\sigma \mathbb{1}(y^\sigma Y_t(x^\sigma) = 1) \tag{1.10}$$

$$= E_\sigma[\mathbb{1}(\underbrace{y^\sigma Y_t(x^\sigma) = 1}_{\text{iff } y^\sigma = Y_t(x^\sigma)})] \tag{1.11}$$

and the **failure rate** by

$$F_t = \sum_\sigma w_t^\sigma \mathbb{1}(y^\sigma Y_t(x^\sigma) = -1) \tag{1.12}$$

$$= E_\sigma[\mathbb{1}(\underbrace{y^\sigma Y_t(x^\sigma) = -1}_{\text{iff } y^\sigma \neq Y_t(x^\sigma)})] \ . \tag{1.13}$$

Note that

$$S_t + F_t = 1 \ , \tag{1.14}$$

and

$$Z_t = Z_{t-1}(e^{-\alpha_t} S_t + e^{+\alpha_t} F_t) \ . \tag{1.15}$$

Figure 1.1: Plot of function $\alpha_t = \frac{1}{2} \ln \frac{1-F_t}{F_t}$.

We can find the α_t values that minimize the classification error Z_t, and then evaluate Z_t at those optimum α_t values:

$$\frac{dZ_t}{d\alpha_t} = Z_{t-1}(-e^{-\alpha_t} S_t + e^{\alpha_t} F_t) = 0 \tag{1.16}$$

$$e^{2\alpha_t} = \frac{S_t}{F_t} \tag{1.17}$$

$$\alpha_t = \frac{1}{2}\ln\frac{S_t}{F_t} = \frac{1}{2}\ln\frac{1-F_t}{F_t} \tag{1.18}$$

$$\frac{Z_t}{Z_{t-1}} = 2\sqrt{S_t F_t} = 2\sqrt{(1-F_t)F_t} \leq 1 \tag{1.19}$$

$f(x) = 2\sqrt{(1-x)x}$ for $x \in [0,1]$ is dome shaped and its maximum is 1, which is achieved iff $x = \frac{1}{2}$

Figure 1.2: Bnet for AdaBoost with 5 w-classifiers, $nt = 5$. All nodes labeled $(\vec{x}, \vec{y})$ are the same node.

The AdaBoost algo described in this chapter is summarized by the bnet of Fig.1.2. The TPMs, printed in blue, for that bnet, are as follows:

$$P(w_0^\sigma) = \frac{1}{nsam} \tag{1.20}$$

for all σ.

$$P(\vec{w}_{t+1}|\vec{w}_t, \alpha_t, Y_t, \vec{x}, \vec{y}) = \prod_\sigma \mathbb{1}\left(\quad w_{t+1}^\sigma = \frac{w_t^\sigma e^{-\alpha_t y^\sigma Y_t(x^\sigma)}}{\sum_\sigma numerator} \quad \right) \tag{1.21}$$

for $t \geq 0$.

$$P(\alpha_t|\vec{w}_t, \vec{x}, \vec{y}) = \mathbb{1}\left(\quad \alpha_t = \frac{1}{2}\ln\frac{1-F_t}{F_t} \text{ where } F_t = F_t(\vec{w}_t, \vec{x}, \vec{y}) \quad \right) \tag{1.22}$$

for $t \in T$.

$$P(\mathcal{Y}_t|\mathcal{Y}_{t-1}, \alpha_t, Y_t) = \mathbb{1}\left(\quad \mathcal{Y}_t = \mathcal{Y}_{t-1} + \alpha_t Y_t \quad \right) \tag{1.23}$$

for $t \in T$, where $\mathcal{Y}_{-1} = 0$.

1.2 AdaBoost for ensemble of tree stumps

Keep in mind that AdaBoost assumes $val(\underline{y}) = val(\underline{x}_t) = \{-1, 1\}$ for all $t \in T = \{0, 1, \ldots, nt-1\}$. In order to implement AdaBoost, we need to specify nt w-classifiers $Y_t : \{-1, 1\}^{nt} \to \{-1, 1\}$ for $t \in T$. One can either specify the nt w-classifiers a priori or build them on-the-fly.

- **w-classifiers specified a priori**

  Define a classifier for each feature x_t where $t \in T$ by:

  $$Y_t(x^\sigma) = x_t^\sigma \in \{-1, 1\} \tag{1.24}$$

  Hence, for this classifier, $y^\sigma Y_t(x^\sigma) = y^\sigma x_t^\sigma$.

- **w-classifiers built on-the-fly**

  Recall dataset $(\vec{x}, \vec{y}) = [(x^\sigma, y^\sigma) : \sigma \in L]$ is indexed by the list $L = [0, 1, \ldots, nsam-1]$. If L_j is a list (possibly with duplicate items) such that $set(L_j) \subset set(L)$, then define $DS_j = (\vec{x}, \vec{y})_{L_j} = ((x^\sigma)_{\sigma \in L_j}, (y^\sigma)_{\sigma \in L_j})$. We will refer to DS_j as the **restriction of $(\vec{x}, \vec{y})$ to L_j.**

  The idea behind on-the-fly w-classifiers is to choose $Y_t(x^\sigma) = x_t^\sigma$, where x_t is the feature with the lowest Gini in the current dataset $(\vec{x}, \vec{y})_{L_t}$. To build $(\vec{x}, \vec{y})_{L_t}$, we select at random from $L = [0, 1, \ldots, nsam - 1]$, a list L_t of the same length as L, using the probability distribution $\vec{w}_{t-1}$. By choosing L_t with probabilities $\vec{w}_{t-1}$, we emphasize individuals σ that are failing. Then we define $(\vec{x}, \vec{y})_{L_t}$ as the restriction of $(\vec{x}, \vec{y})$ to L_t.

  Perhaps a causal diagram will make all these new steps clearer to the reader. The bnet Fig.1.3 is a modification of the bnet Fig.1.2 to include these new steps. The TPMs, printed in blue, for new or changed nodes, are as follows:

  $$P(Y_t|(\vec{x}, \vec{y})_{L_t}) = \mathbb{1}(\quad Y_t(x^\sigma) = x_t^\sigma \text{ where } x_t \text{ is feature in } (\vec{x}, \vec{y})_{L_t} \text{ the with lowest Gini.} \quad) \tag{1.25}$$

  $$P(L_{t+1}^\sigma = \sigma' | \vec{w}_t) = w_t^{\sigma'} \tag{1.26}$$

  $$P((\vec{x}, \vec{y})_{L_t} | L_t, (\vec{x}, \vec{y})) = \mathbb{1}(\quad (\vec{x}, \vec{y})_{L_t} = \text{ restriction of } (\vec{x}, \vec{y}) \text{ to } L_t \quad) \tag{1.27}$$

Figure 1.3: Modification of the bnet of Fig.1.2 to include on-the-fly generation of the w-classifiers.

Chapter 2

ANOVA

ANOVA stands for "Analysis of Variance".

2.1 Law of Total Variance

Claim 24 *(Law of Total Variance) Suppose* $P : val(\underline{x}) \times val(\underline{y}) \to [0,1]$ *is a probability distribution. Suppose* $f : val(\underline{x}) \times val(\underline{y}) \to \mathbb{R}$ *and* $f = f(x,y)$*. Then*

$$Var_{\underline{x},\underline{y}}(f) = E_{\underline{y}}[Var_{\underline{x}|\underline{y}}(f)] + Var_{\underline{y}}(E_{\underline{x}|\underline{y}}[f]) \ . \tag{2.1}$$

In particular,

$$Var_{\underline{x}}(x) = E_{\underline{y}}[Var_{\underline{x}|\underline{y}}(x)] + Var_{\underline{y}}(E_{\underline{x}|\underline{y}}[x]) \ . \tag{2.2}$$

proof:

Let

$$A = \sum_y P(y) \left(\sum_x P(x|y) f \right)^2 \ . \tag{2.3}$$

Then

$$\begin{aligned} Var_{\underline{x},\underline{y}}(f) &= \sum_{x,y} P(x,y) f^2 - \left(\sum_{x,y} P(x,y) f \right)^2 && (2.4) \\ &= \left\{ \begin{array}{l} \sum_{x,y} P(x,y) f^2 - A \\ + \left(A - \left(\sum_{x,y} P(x,y) f \right)^2 \right) \end{array} \right. && (2.5) \end{aligned}$$

$$\begin{aligned} E_{\underline{y}}[Var_{\underline{x}|\underline{y}}(f)] &= \sum_y P(y) \left(\sum_x P(x|y) f^2 - \left(\sum_x P(x|y) f \right)^2 \right) && (2.6) \\ &= \sum_{x,y} P(x,y) f^2 - A && (2.7) \end{aligned}$$

$$
\begin{aligned}
Var_{\underline{y}}(E_{\underline{x}|\underline{y}}[f]) &= \sum_y P(y)\left(\sum_x P(x|y)f\right)^2 - \left(\sum_y P(y)\sum_x P(x|y)f\right)^2 && (2.8)\\
&= A - \left(\sum_{x,y} P(x,y)f\right)^2 && (2.9)
\end{aligned}
$$

QED

2.2 Sum of Squares Estimates

Consider a population Σ partitioned into groups Σ_g such that $\Sigma = \cup_{g=1}^{ng}\Sigma_g$, where the Σ_g are mutually disjoint.

Let
dof stand for "degrees of freedom"
SS stand for "Sum of Squares"
MS stand for "Mean Square"
$x_{\Sigma_g} = \{x_{\sigma|g}\}_{\sigma\in\Sigma_g}$
Define the total and group mean values by

$$\overline{x} = \frac{1}{|\Sigma|}\sum_{g=1}^{ng}\sum_{\sigma\in\Sigma_g} x_{\sigma|g} \tag{2.10}$$

$$\overline{x}_g = \frac{1}{|\Sigma_g|}\sum_{\sigma\in\Sigma_g} x_{\sigma|g} \tag{2.11}$$

Define the **SS Total** by

$$
\begin{aligned}
SS_T &= \sum_{g=1}^{ng}\sum_{\sigma\in\Sigma_g}(x_{\sigma|g}-\overline{x})^2 && (2.12)\\
&= |\Sigma| Var_{\underline{x}}(x) && (2.13)
\end{aligned}
$$

with dof and MS given by

$$dof_T = |\Sigma| - 1,\ MS_T = \frac{SS_T}{dof_T} \tag{2.14}$$

Define the **SS Within** by

$$
\begin{aligned}
SS_W &= \sum_{g=1}^{ng}\sum_{\sigma\in\Sigma_g}(x_{\sigma|g}-\overline{x}_g)^2 && (2.15)\\
&= |\Sigma|\sum_{g=1}^{ng}\frac{|\Sigma_g|}{|\Sigma|}\left\{\frac{1}{|\Sigma_g|}\sum_{\sigma\in\Sigma_g}(x_{\sigma|g}-\overline{x}_g)^2\right\} && (2.16)\\
&= |\Sigma| E_{\underline{g}}[Var_{\underline{x}|\underline{g}}(x)] \quad \text{(note this is a mean value)} && (2.17)
\end{aligned}
$$

with dof and MS given by

$$dof_W = |\Sigma| - ng,\ MS_W = \frac{SS_W}{dof_W} \tag{2.18}$$

Define the **SS Between** by

$$\begin{aligned} SS_B &= \sum_{g=1}^{ng} |\Sigma_g|(\overline{x}_g - \overline{x})^2 && (2.19)\\ &= |\Sigma| \sum_{g=1}^{ng} \frac{|\Sigma_g|}{|\Sigma|}(\underbrace{\overline{x}_g}_{E_{\underline{x}|\underline{g}}[x]} - \overline{x})^2 && (2.20)\\ &= |\Sigma| Var_{\underline{g}}(E_{\underline{x}|\underline{g}}[x]) \quad \text{(note this is a variance)} && (2.21) \end{aligned}$$

with dof and MS given by

$$dof_B = ng - 1,\ MS_B = \frac{SS_B}{dof_B} \tag{2.22}$$

Claim 25

$$SS_T = SS_W + SS_B \tag{2.23}$$

$$dof_T = dof_W + dof_B \tag{2.24}$$

proof: By the just proven Law of Total Variance,

$$\underbrace{Var_{\underline{x}}(x)}_{SS_T/|\Sigma|} = \underbrace{E_{\underline{g}}[Var_{\underline{x}|\underline{g}}(x)]}_{SS_W/|\Sigma|} + \underbrace{Var_{\underline{g}}(E_{\underline{x}|\underline{g}}[x])}_{SS_B/|\Sigma|}\ . \tag{2.25}$$

QED

Figure 2.1: Bnet for calculating SS_T, SS_W and SS_B.

Fig.2.1 shows a bnet for calculating SS_T, SS_W and SS_B, The TPMs, printed in blue, for the bnet Fig.2.1, are as follows.

$$P(\overline{x}|\{x_{\Sigma_g}\}_{g=1}^{ng}) = \mathbb{1}\left(\overline{x} = \frac{1}{|\Sigma|}\sum_{g=1}^{ng}\sum_{\sigma\in\Sigma_g} x_{\sigma|g}\right) \tag{2.26}$$

$$P(\overline{x}_g|x_{\Sigma_g}) = \mathbb{1}\left(\overline{x}_g = \frac{1}{|\Sigma_g|}\sum_{\sigma\in\Sigma_g} x_{\sigma|g}\right) \tag{2.27}$$

$$P(V_g|x_{\Sigma_g}, \overline{x}_g) = \mathbb{1}\left(V_g = \frac{1}{|\Sigma_g|}\sum_{\sigma\in\Sigma_g} (x_{\sigma|g} - \overline{x}_g)^2\right) \tag{2.28}$$

$$P(SS_T|\{x_{\Sigma_g}\}_{g=1}^{ng}, \overline{x}) = \mathbb{1}(SS_T = SS_T(\{x_{\Sigma_g}\}_{g=1}^{ng}, \overline{x})) \tag{2.29}$$

$$P(SS_B|\{\overline{x}_g\}_{g=1}^{ng}, \overline{x}) = \mathbb{1}(SS_B = SS_B(\{\overline{x}_g\}_{g=1}^{ng}, \overline{x})) \tag{2.30}$$

$$P(SS_W|\{V_g\}_{g=1}^{ng}) = \mathbb{1}(SS_W = SS_W(\{V_g\}_{g=1}^{ng})) \tag{2.31}$$

2.3 F-statistic and hypothesis testing

The **F-statistic** for ANOVA is defined by

$$F = \frac{MS_B}{MS_W} \quad \left(= \frac{variance}{mean\ value}\right) \tag{2.32}$$

(Note that F is the ratio of two chi-square distributions)

Consider $ng = 3$ for definiteness. Let

h_0 = hypothesis that $\mu_1 = \mu_2 = \mu_3$ (null hypothesis)

h_1 = opposite of h_0 (alternative, opposite hypothesis)

It's not hard to find on the Internet, tables and software calculators that give the score p-value[1]

$$p_{sc} = \int_F^\infty dF'\ P(F'; dof_B, dof_W) \tag{2.33}$$

[1]Some tables give $F = F(p_{sc}, dof_B, dof_W)$ instead of $p_{sc} = p_{sc}(F, dof_B, dof_W)$, but note that there is a 1-1/onto invertible map between F and p_{sc}. Some tables refer to p_{sc} as α.

Figure 2.2: Probability distribution $P(F)$ for the F-statistic, at fixed dof_B and dof_W. See Wikipedia article Ref.[134] for more info about $P(F)$.

of the F statistic, for a given dof_B and dof_W. See Fig.2.2 for a portrait of $P(F)$ and p_{sc}. In Bayesian language, p_{sc} measures the probability that h_0 is true. Assume $p_{th} = 0.05$.[2] To determine if the difference between group means is "statistically significant", we compare p_{sc} with p_{th}. If $p_{sc} < p_{th} = 0.05$, we reject the null hypothesis. Otherwise, we fail to reject (accept) the null hypothesis.

If h_0 is rejected, we can perform **post-hoc** tests to determine how much the groups differ from each other. There are several popular post-hoc tests, such as the Tukey test, Bonferroni test, and Scheffe test.

[2]The difference between threshold (th) and score (sc) p-values is discussed in Section C.41.

Chapter 3

ARACNE structure learning

This chapter is based on Ref.[45].

The ARACNE algo is an algo for learning the structure of a bnet from data. The algo considers data samples for n random variables $(x_i)_{i=0,1,\ldots,n-1}$, and estimates the mutual information $MI_{i,j} = H(\underline{x}_i : \underline{x}_j)$ between every pair of nodes. The set UG is initialized to contain all the edges of a fully connected undirected graph. Next the algo removes from UG every edge with $MI_{i,j} < \epsilon$ for some threshold $0 < \epsilon << 1$. Then the algo examines every triplet of edges in UG, and marks for removal the edge of the triplet with the smallest MI. Finally, the algo removes from UG all edges marked for removal. Each triplet is analyzed irrespective of whether its edges have been marked for removal when considering a prior triplet. Thus the network constructed by the algorithm is independent of the order in which the triplets are examined. Some of the unoriented edges in UG can be given an orientation using the same techniques used to orient edges in constraint based structure learning (see Chapter 95).

Ref.[45] incorrectly claims that removing the smaller of 3 MI's is "an application" of the Data Processing Inequality (DPI) of Shannon Information Theory. See Chapter 56 for more info about DPI. Note that DPI is only valid for a Markov chain, and not all triplets of random variables are in a Markov chain. Removing the smaller of 3 numbers does not require DPI.

Fig.3.1 gives an example of the application of the ARACNE algo.

See Chapter 9 on Chow-Liu trees (CLT). A CLT is just a maximum spanning tree where the weights are mutual informations $MI_{i,j}$ estimated from data.

Sometimes, the outcome of the ARACNE algo is a CLT. For example, Fig.3.1 (a) was considered in Chapter 9 on CLTs, and the CLT algo also gave Fig.3.1 (c) as the final structure.

According to Ref.[45], the ARACNE algo sometimes yields a polytree (i.e., a connected graph with no loops). It may even yield a structure with loops. Hence, it does not always yield a CLT.

By breaking all cliques (i.e., fully connected subgraphs) with 3 edges and 3 nodes, ARACNE breaks all cliques with 3 or more nodes. However, cliques are not uncommon in Nature, especially 3 node cliques. Cliques become less likely in Nature

Figure 3.1: An example where the ARACNE algo gives a Chow-Liu tree. (a) Fully connected undirected graph with weights $MI_{i,j}$ along the edges. (b) All 4 possible triplets of edges with nonzero weights. Edges marked for removal have their weights printed in red.(c) Final structure.

the bigger the number of nodes they have *after* 3. Therefore, a nice generalization of ARACNE would be to list all 4 node cliques, and break each of them by eliminating their edge with the smallest MI. This will have the effect of breaking all cliques with 4 or more nodes but keeping 3 node cliques. One could also break some, not all, of the 3 node cliques, by consecutively removing the clique-breaking-edge with the smallest MI of all edges of all 3 node cliques. Let β stand for banned clique number of nodes. Then the current ARACNE has $\beta = 3$. We are suggesting that a β of 4 might be more likely to occur in Nature.

Chapter 4

Backdoor Adjustment Formula

The backdoor (BD) adjustment formula is proven in Chapter 22 from the rules of Do Calculus. The goal of this chapter is to give examples of the use of that theorem. We will restate the theorem in this chapter, sans proof. There is no need to understand the theorem's proof in order to use it. However, you will need to skim Chapter 22 in order to familiarize yourself with the notation used to state the theorem. This chapter also assumes that you are comfortable with the rules for checking for d-separation. Those rules are covered in Chapter 24.

Suppose that we have access to data that allows us to estimate a probability distribution $P(x., y., z.)$. Then we say that the backdoor $\underline{z}.$ satisfies the **backdoor adjustment criterion** relative to $(\underline{x}., \underline{y}.)$ if

1. All backdoor paths from $\underline{x}.$ to $\underline{y}.$ are blocked by conditioning on $\underline{z}.$.
2. $\underline{z}. \cap de(\underline{x}.) = \emptyset$.

Claim 26 Backdoor Adjustment Formula

If $\underline{z}.$ satisfies the backdoor criterion relative to $(\underline{x}., \underline{y}.)$, then

$$P(y.|\mathcal{D}\underline{x}. = x.) = \sum_{z.} P(y.|x., z.)P(z.) \tag{4.1}$$

$$= \tag{4.2}$$

$$\begin{tikzcd} \mathbb{E}z. \arrow[dr] & \\ x. \arrow[r] & y. \end{tikzcd}$$

proof: See Chapter 22.
QED

4.1 Examples

1.

$$\begin{tikzcd} & \underline{z} \arrow[dr] & \\ \underline{x} \arrow[ur] \arrow[rr] & & \underline{y} \end{tikzcd} \tag{4.3}$$

If $\underline{x}. = \underline{x}, \underline{y}. = \underline{y}, \underline{z}. = \underline{z}$, BD criterion not satisfied because $\underline{z} \in de(\underline{x})$. However,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} & \sum z \arrow[dr] & \\ x \arrow[ur] \arrow[rr] & & y \end{tikzcd} \tag{4.4}$$

2.

$$\begin{tikzcd} & \underline{z} & \\ \underline{x} \arrow[ur] \arrow[rr] & & \underline{y} \arrow[ul] \end{tikzcd} \tag{4.5}$$

If $\underline{x}. = \underline{x}, \underline{y}. = \underline{y}, \underline{z}. = \underline{z}$, BD criterion not satisfied because $\underline{z} \in de(\underline{x})$. However,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} & \sum z & \\ x \arrow[ur] \arrow[rr] & & y \arrow[ul] \end{tikzcd} \tag{4.6}$$

3.

$$\begin{tikzcd} & \underline{z} \arrow[dl] \arrow[dr] & \\ \underline{x} \arrow[rr] & & \underline{y} \end{tikzcd} \tag{4.7}$$

BD criterion satisfied if $\underline{x}. = \underline{x}, \underline{y}. = \underline{y}, \underline{z}. = \underline{z}$. In fact,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} & \mathbb{E}z \arrow[dr] & \\ x \arrow[rr] & & y \end{tikzcd} \tag{4.8}$$

Note that here the backdoor formula adjusts the parents of $\underline{x}$.. More complicated examples might adjust for (i.e., average over) non-parents.

4.

$$\begin{tikzcd} & \underline{z} \arrow[dl] \arrow[dr] & \\ \underline{x} \arrow[r] & \underline{m} \arrow[r] & \underline{y} \end{tikzcd} \tag{4.9}$$

BD criterion satisfied if $\underline{x}. = \underline{x}, \underline{y}. = \underline{y}, \underline{z}. = \underline{z}$. In fact,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} & \mathbb{E}z \arrow[dr] & \\ x \arrow[r] & \sum m \arrow[r] & y \end{tikzcd} \tag{4.10}$$

5.

$$\begin{tikzcd} & \underline{z}_1 \arrow[d] \arrow[ddl] \arrow[ddr] & \\ & \underline{z}_2 \arrow[dl] \arrow[dr] & \\ \underline{x} \arrow[rr] & & \underline{y} \end{tikzcd} \tag{4.11}$$

BD criterion satisfied if $\underline{x}. = \underline{x}, \underline{y}. = \underline{y}, \underline{z}. = (\underline{z}_1, \underline{z}_2)$. In fact,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} & \mathbb{E}z_1 \arrow[d] \arrow[ddr] & \\ & \sum z_2 \arrow[dr] & \\ x \arrow[rr] & & y \end{tikzcd} \tag{4.12}$$

6.

$$\begin{tikzcd} \underline{z}_1 \arrow[dr] \arrow[dd] & & \underline{z}_3 \arrow[dl] \arrow[dd] \\ & \underline{z}_2 & \\ \underline{x} \arrow[rr] & & \underline{y} \end{tikzcd} \tag{4.13}$$

BD criterion satisfied if $\underline{x}. = \underline{x}, \underline{y}. = \underline{y}$ and $\underline{z}. = \underline{z}_1$. In fact,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} \mathbb{E}z_1 \arrow[dr] & & \sum z_3 \arrow[dl] \arrow[dd] \\ & \sum z_2 & \\ x \arrow[rr] & & y \end{tikzcd} \tag{4.14}$$

7.

$$\begin{tikzcd} & \underline{z} \arrow[dl] \arrow[dr] & \\ \underline{x} \arrow[r] & \underline{m} \arrow[r] & \underline{y} \end{tikzcd} \tag{4.15}$$

First sum over $\underline{m}$. BD criterion is not satisfied if $\underline{x}. = \underline{x}, \underline{y}. = \underline{y}, \underline{z}. = \underline{z}$ because $\underline{z}$ is hidden and can't be conditioned on. However, the frontdoor criterion can be satisfied. In fact,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} & \mathbb{E}x' \arrow[dr] & \\ x \arrow[r] & \sum m \arrow[r] & y \end{tikzcd} \tag{4.16}$$

8.

$$\begin{tikzcd} \underline{w} \arrow[r] \arrow[d] & \underline{z} \arrow[d] \\ \underline{x} \arrow[r] & \underline{y} \end{tikzcd} \tag{4.17}$$

First sum over $\underline{z}$ node. Now BD criterion not satisfied if[1] $\underline{x}. = \underline{x}, \underline{y}. = \underline{y}, \underline{z}. = \underline{w}$, because $\underline{w}$ is hidden and can't be averaged over. However,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} \mathbb{E}x' \arrow[r] & \sum z \arrow[d] \\ x \arrow[r] & y \end{tikzcd} \tag{4.18}$$

9.

$$\begin{tikzcd} \underline{e} \arrow[r] \arrow[d] & \underline{z} \arrow[dl] \arrow[dr] & \underline{a} \arrow[l] \arrow[d] \\ \underline{x} \arrow[rr] & & \underline{y} \end{tikzcd} \tag{4.19}$$

[1] To be totally correct, the backdoor criterion should ask that all nodes besides $\underline{x}., \underline{y}., \underline{z}.$ be summed over.

First sum over the $\underline{z}, \underline{a}$ nodes. Now BD criterion not satisfied if $\underline{x}. = \underline{x}, \underline{y}. = \underline{y}, \underline{z}. = \underline{e}$, because $\underline{e}$ is hidden and can't be averaged over. However,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} \mathbb{E}x' \arrow[r] \arrow[dr] & \sum z \arrow[l] \arrow[dl] \arrow[dr] & \mathbb{E}a \arrow[l] \arrow[d] \\ x \arrow[rr] & & y \end{tikzcd} \tag{4.20}$$

10.

$$\begin{tikzcd} \underline{z} \arrow[d] & & \underline{t} \arrow[ll] \arrow[d] \\ \underline{w} & \underline{x} \arrow[l] \arrow[r] & \underline{y} \end{tikzcd} \tag{4.21}$$

Sum over $\underline{t}$ and $\underline{z}$. BD criterion not satisfied because $\underline{w} \in de(\underline{x})$. However,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} \sum z \arrow[d] & & \mathbb{E}t \arrow[ll] \arrow[d] \\ \sum w & x \arrow[l] \arrow[r] & y \end{tikzcd} \tag{4.22}$$

Chapter 5

Back Propagation (Automatic Differentiation)

5.1 Toy Example

This example comes from Ref.[54].

Consider the following system of 4 equations:

$$\begin{cases} b = w_1 a \\ c = w_2 a \\ d = w_3 b + w_4 c \\ L = 10 - d \end{cases} \tag{5.1}$$

To calculate $\nabla_w L$ where $w = (w_1, w_2, w_3, w_4)$, we can use the chain rule for partial derivatives. This yields

$$\frac{\partial L}{\partial w_4} = \frac{\partial L}{\partial d}\frac{\partial d}{\partial w_4} = (-1)(c) \tag{5.2a}$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial d}\frac{\partial d}{\partial w_3} = (-1)(b) \tag{5.2b}$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial d}\frac{\partial d}{\partial c}\frac{\partial c}{\partial w_2} = (-1)(w_4)(a) \tag{5.2c}$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial d}\frac{\partial d}{\partial b}\frac{\partial b}{\partial w_1} = (-1)(w_3)(a) \tag{5.2d}$$

Now note that the system of equations Eq.(5.1) can be represented graphically by the bnet (with deterministic nodes) of Fig.5.1. The calculation of $\nabla_w L$ can also be represented graphically with the aid of the defining bnet Fig.5.1. This is done in Fig.5.2. As illustrated by that figure, the derivative $\partial_{w_i} L$ for $i = 1, 2, 3, 4$ is the product of the derivatives along the arrows from w_i to L.

Figure 5.1: bnet for which we want to calculate $\nabla_w L$ where $w = (w_1, w_2, w_3, w_4)$

Figure 5.2: bnet of Fig. 5.1 with derivatives along each arrow.

5.2 General Theory

5.2.1 Jacobians

Suppose $f : \mathbb{R}^{nx} \to \mathbb{R}^{nf}$ and

$$y = f(x) \ . \tag{5.3}$$

Then the Jacobian $\frac{\partial y}{\partial x}$ is defined as the matrix with entries[1]

$$\left[\frac{\partial y}{\partial x}\right]_{i,j} = \frac{\partial y_i}{\partial x_j} \ . \tag{5.4}$$

Jacobian of function composition. Suppose $f : \mathbb{R}^{nx} \to \mathbb{R}^{nf}$, $g : \mathbb{R}^{nf} \to \mathbb{R}^{ng}$. If

$$y = g \circ f(x) \ , \tag{5.5}$$

then

$$\frac{\partial y}{\partial x} = \frac{\partial g}{\partial f}\frac{\partial f}{\partial x} \ . \tag{5.6}$$

Right hand side of last equation is a product of two matrices so order of matrices is important.

5.2.2 Bnets for function composition, forward propagation and back propagation

Let

$$y = f^4 \circ f^3 \circ f^2 \circ f^1(x) \ . \tag{5.7}$$

This function composition chain can be represented by the bnet Fig.5.3(a) with TPMs

$$P(f^\mu | f^{\mu-1}) = \mathbb{1}(f^\mu = f^\mu(f^{\mu-1})) \tag{5.8}$$

for $\mu = 1, 2, 3, 4$.

Note that

$$\begin{aligned}
\frac{\partial y}{\partial x} &= \frac{\partial y}{\partial f^3}\frac{\partial f^3}{\partial f^2}\left[\frac{\partial f^2}{\partial f^1}\frac{\partial f^1}{\partial x}\right] && (5.9)\\
&= \frac{\partial y}{\partial f^3}\left[\frac{\partial f^3}{\partial f^2}\frac{\partial f^2}{\partial x}\right] && (5.10)\\
&= \left[\frac{\partial y}{\partial f^3}\frac{\partial f^3}{\partial x}\right] && (5.11)\\
&= \frac{\partial y}{\partial x} \ . && (5.12)
\end{aligned}$$

This forward propagation can be represented by the bnet Fig.5.3(b) with node TPMs

$$P(\frac{\partial f^{\mu+1}}{\partial x} \mid \frac{\partial f^\mu}{\partial x}) = \mathbb{1}(\frac{\partial f^{\mu+1}}{\partial x} = \frac{\partial f^{\mu+1}}{\partial f^\mu}\frac{\partial f^\mu}{\partial x}) \tag{5.13}$$

[1]Mnemonic for remembering order of indices: i in numerator/j in denominator becomes index i/j of Jacobian matrix.

$$\underline{f}^4 \longleftarrow \underline{f}^3 \longleftarrow \underline{f}^2 \longleftarrow \underline{f}^1 \longleftarrow \underline{f}^0$$

(a) Composition

$$\underline{\frac{\partial f^4}{\partial x}} \longleftarrow \underline{\frac{\partial f^3}{\partial x}} \longleftarrow \underline{\frac{\partial f^2}{\partial x}} \longleftarrow \underline{\frac{\partial f^1}{\partial x}} \longleftarrow \underline{1}$$

(b) Forward-p

$$\underline{1} \longrightarrow \underline{\frac{\partial y}{\partial f^3}} \longrightarrow \underline{\frac{\partial y}{\partial f^2}} \longrightarrow \underline{\frac{\partial y}{\partial f^1}} \longrightarrow \underline{\frac{\partial y}{\partial f^0}}$$

(c) Back-p

Figure 5.3: bnets for function composition, forward propagation and back propagation for $nf = 5$ nodes.

for $\mu = 1, 2, 3$.

Note that

$$\begin{aligned}
\frac{\partial y}{\partial x} &= \left[\frac{\partial y}{\partial f^3}\frac{\partial f^3}{\partial f^2}\right]\frac{\partial f^2}{\partial f^1}\frac{\partial f^1}{\partial x} && (5.14)\\
&= \left[\frac{\partial y}{\partial f^2}\frac{\partial f^2}{\partial f^1}\right]\frac{\partial f^1}{\partial x} && (5.15)\\
&= \left[\frac{\partial y}{\partial f^1}\frac{\partial f^1}{\partial x}\right] && (5.16)\\
&= \frac{\partial y}{\partial x}\,. && (5.17)
\end{aligned}$$

This back propagation can be represented by the bnet Fig.5.3(c) with node TPMs

$$P(\frac{\partial y}{\partial f^\mu} \mid \frac{\partial y}{\partial f^{\mu+1}}) = \mathbb{1}(\frac{\partial y}{\partial f^\mu} = \frac{\partial y}{\partial f^{\mu+1}}\frac{\partial f^{\mu+1}}{\partial f^\mu}) \tag{5.18}$$

for $\mu = 2, 1, 0$.

$\frac{\partial f^{\mu+1}}{\partial f^\mu}$ is a Jacobian matrix so the order of multiplication matters. In forward prop, it pre-multiplies, and in back prop it post-multiplies.

5.3 Application to Neural Networks

5.3.1 Absorbing b_i^λ into $w_{i|j}$.

Figure 5.4: Nodes $\underline{h}_0^0, \underline{h}_0^1, \underline{Y}_0$ are all set to 1. They allow us to absorb b_i^λ into the first column of $w_{i|j}^\lambda$.

The TPMs, printed in blue, for a NN bnet, as given in Chapter 68, are as follows.

For all hidden layers $\lambda = 0, 1, \ldots, \Lambda - 2$,

$$P(h_i^\lambda \mid h_\cdot^{\lambda-1}) = \delta\left(h_i^\lambda, \mathcal{A}_i^\lambda(\sum_j w_{i|j}^\lambda h_j^{\lambda-1} + b_i^\lambda)\right) \tag{5.19}$$

for $i = 0, 1, \ldots, nh(\lambda) - 1$. For the output visible layer $\lambda = \Lambda - 1$:

$$P(Y_i \mid h_\cdot^{\Lambda-2}) = \delta\left(Y_i, \mathcal{A}_i^{\Lambda-1}(\sum_j w_{i|j}^{\Lambda-1} h_j^{\Lambda-2} + b_i^{\Lambda-1})\right) \tag{5.20}$$

for $i = 0, 1, \ldots, ny - 1$.

For each λ, replace the matrix $w_{\cdot|\cdot}^\lambda$ by the augmented matrix $[b^\lambda\cdot, w_{\cdot|\cdot}^\lambda]$ so that the new $w_{\cdot|\cdot}^\lambda$ satisfies

$$w_{i|0}^\lambda = b_i^\lambda \tag{5.21}$$

Let the nodes $\underline{h}_0^\lambda$ for all λ and $\underline{Y}_0$ be root nodes (so no arrows pointing into them). For each λ, draw arrows from $\underline{h}_0^\lambda$ to all other nodes in that same layer. Draw arrows from $\underline{Y}_0$ to all other nodes in that same layer.

After performing the above steps, the TPMs, printed in blue, for the NN bnet, are as follows:

For all hidden layers $\lambda = 0, 1, \ldots, \Lambda - 2$,

$$P(h_0^\lambda) = \delta(h_0^\lambda, 1)\ , \tag{5.22}$$

and

$$P(h_i^\lambda \mid h_\cdot^{\lambda-1}, h_0^\lambda = 1) = \delta\left(h_i^\lambda, \mathcal{A}_i^\lambda(\sum_j w_{i|j}^\lambda h_j^{\lambda-1})\right) \tag{5.23}$$

for $i = 1, \ldots, nh(\lambda) - 1$. For the output visible layer $\lambda = \Lambda - 1$:

$$P(Y_0) = \delta(Y_0, 1)\ , \tag{5.24}$$

and

$$P(Y_i \mid h_\cdot^{\Lambda-2}, Y_0 = 1) = \delta\left(Y_i, \mathcal{A}_i^{\Lambda-1}(\sum_j w_{i|j}^{\Lambda-1} h_j^{\Lambda-2})\right) \tag{5.25}$$

for $i = 1, 2, \ldots, ny - 1$.

5.3.2 Bnets for function composition, forward propagation and back propagation for NN

$$\underline{\mathcal{A}^3} \longleftarrow \underline{\mathcal{B}^3} \longleftarrow \underline{\mathcal{A}^2} \longleftarrow \underline{\mathcal{B}^2} \longleftarrow \underline{\mathcal{A}^1} \longleftarrow \underline{\mathcal{B}^1} \longleftarrow \underline{\mathcal{A}^0} \longleftarrow \underline{\mathcal{B}^0} \longleftarrow \underline{x}$$

(a)

$$\underline{\frac{\partial \mathcal{A}^3}{\partial x}} \longleftarrow \underline{\frac{\partial \mathcal{B}^3}{\partial x}} \longleftarrow \underline{\frac{\partial \mathcal{A}^2}{\partial x}} \longleftarrow \underline{\frac{\partial \mathcal{B}^2}{\partial x}} \longleftarrow \underline{\frac{\partial \mathcal{A}^1}{\partial x}} \longleftarrow \underline{\frac{\partial \mathcal{B}^1}{\partial x}} \longleftarrow \underline{\frac{\partial \mathcal{A}^0}{\partial x}} \longleftarrow \underline{\frac{\partial \mathcal{B}^0}{\partial x}} \longleftarrow \underline{1}$$

(b)

$$\underline{1} \longrightarrow \underline{\frac{\partial Y}{\partial \mathcal{B}^3}} \longrightarrow \underline{\frac{\partial Y}{\partial \mathcal{A}^2}} \longrightarrow \underline{\frac{\partial Y}{\partial \mathcal{B}^2}} \longrightarrow \underline{\frac{\partial Y}{\partial \mathcal{A}^1}} \longrightarrow \underline{\frac{\partial Y}{\partial \mathcal{B}^1}} \longrightarrow \underline{\frac{\partial Y}{\partial \mathcal{A}^0}} \longrightarrow \underline{\frac{\partial Y}{\partial \mathcal{B}^0}} \longrightarrow \underline{\frac{\partial Y}{\partial x}}$$

(c)

Figure 5.5: bnets for (a) function composition, (b) forward propagation and (c) back propagation for a neural net with 4 layers (3 hidden and output visible).

.

From here on, we will rename y above by $Y = \widehat{y}$ and consider samples $y[i]$ for $i = 0, 1, \ldots, nsam - 1$. The Error (a.k.a. loss or cost function) is

$$\mathcal{E} = \frac{1}{nsam} \sum_{\sigma=0}^{nsam-1} \sum_{i=0}^{ny-1} |Y_i - y_i[\sigma]|^2 \tag{5.26}$$

To perform simple gradient descent, one uses:

$$(w^{\lambda}_{i|j})' = w^{\lambda}_{i|j} - \eta \frac{\partial \mathcal{E}}{\partial w^{\lambda}_{i|j}} \ . \tag{5.27}$$

One has

$$\frac{\partial \mathcal{E}}{\partial w^{\lambda}_{i|j}} = \frac{1}{nsam} \sum_{\sigma=0}^{nsam-1} \sum_{i=0}^{ny-1} 2(Y_i - y_i[\sigma]) \frac{\partial Y}{\partial w^{\lambda}_{i|j}} \ . \tag{5.28}$$

Define $\mathcal{B}^{\lambda}_i$ thus

$$\mathcal{B}^{\lambda}_i(h^{\lambda-1}) = \sum_j w^{\lambda}_{i|j} h^{\lambda-1}_j \ . \tag{5.29}$$

Then

$$\frac{\partial Y}{\partial w^{\lambda}_{i|j}} = \frac{\partial Y}{\partial \mathcal{B}^{\lambda}_i} \frac{\partial \mathcal{B}^{\lambda}_i}{\partial w^{\lambda}_{i|j}} \tag{5.30}$$

$$= \frac{\partial Y}{\partial \mathcal{B}^{\lambda}_i} h^{\lambda-1}_j \tag{5.31}$$

$$\frac{\partial \mathcal{E}}{\partial w^{\lambda}_{i|j}} = \frac{\partial \mathcal{E}}{\partial \mathcal{B}^{\lambda}_j} \frac{\partial \mathcal{B}^{\lambda}_j}{\partial w^{\lambda}_{i|j}} \tag{5.32}$$

$$= \frac{\partial \mathcal{E}}{\partial \mathcal{B}^{\lambda}_j} h^{\lambda-1}_j \ . \tag{5.33}$$

This suggest that we can calculate the derivatives of the error $\mathcal{E}$ with respect to the weights $w^{\lambda}_{i|j}$ in two stages, using an intermediate quantity δ^{λ}_j:

$$\begin{cases} \delta^{\lambda}_j = \frac{\partial \mathcal{E}}{\partial \mathcal{B}^{\lambda}_j} \\ \frac{\partial \mathcal{E}}{\partial w^{\lambda}_{i|j}} = \delta^{\lambda}_j h^{\lambda-1}_j \end{cases} \tag{5.34}$$

To apply what we learned in the earlier General Theory section of this chapter, consider a NN with 4 layers (3 hidden, and the output visible one). Define the functions f_i as follows:

$$f^0_i = x_i \tag{5.35}$$

$$\text{Layer 0:} \quad f_i^1 = \mathcal{B}_i^0(x_i), \quad f_i^2 = \mathcal{A}_i^0(\mathcal{B}_i^0) \tag{5.36}$$
$$\text{Layer 1:} \quad f_i^3 = \mathcal{B}_i^1(\mathcal{A}_i^0), \quad f_i^4 = \mathcal{A}_i^1(\mathcal{B}_i^1) \tag{5.37}$$
$$\text{Layer 2:} \quad f_i^5 = \mathcal{B}_i^2(\mathcal{A}_i^1), \quad f_i^6 = \mathcal{A}_i^2(\mathcal{B}_i^2) \tag{5.38}$$
$$\text{Layer 3:} \quad f_i^7 = \mathcal{B}_i^3(\mathcal{A}_i^2), \quad f_i^8 = \mathcal{A}_i^3(\mathcal{B}_i^3) \tag{5.39}$$

See Fig.5.5. The TPMs, printed in blue, for the bnet (c) for back propagation, are as follows:

$$P(\frac{\partial Y}{\partial \mathcal{B}^\lambda} \mid \frac{\partial Y}{\partial \mathcal{B}^{\lambda+1}}) = \mathbb{1}(\frac{\partial Y}{\partial \mathcal{B}^\lambda} = \frac{\partial Y}{\partial \mathcal{B}^{\lambda+1}} \frac{\partial \mathcal{B}^{\lambda+1}}{\partial \mathcal{A}^\lambda} \frac{\partial \mathcal{A}^\lambda}{\partial \mathcal{B}^\lambda}) \ . \tag{5.40}$$

One has

$$\frac{\partial \mathcal{A}_i^\lambda}{\partial \mathcal{B}_j^\lambda} = D\mathcal{A}_i{}^\lambda(\mathcal{B}_i^\lambda)\delta(i,j) \tag{5.41}$$

where $D\mathcal{A}_i^\lambda(z)$ is the derivative of $\mathcal{A}_i^\lambda(z)$.

From Eq.(5.29)

$$\mathcal{B}_i^{\lambda+1}(\mathcal{A}^\lambda) = \sum_j w_{i|j}^{\lambda+1} \mathcal{A}_j^\lambda \tag{5.42}$$

so

$$\frac{\partial \mathcal{B}_i^{\lambda+1}}{\partial \mathcal{A}_j^\lambda} = w_{i|j}^{\lambda+1} \ . \tag{5.43}$$

Therefore, Eq.(5.40) implies

$$P(\frac{\partial Y}{\partial \mathcal{B}_j^\lambda} \mid \frac{\partial Y}{\partial \mathcal{B}_j^{\lambda+1}}) = \mathbb{1}(\frac{\partial Y}{\partial \mathcal{B}_j^\lambda} = \sum_i \frac{\partial Y}{\partial \mathcal{B}_i^{\lambda+1}} D\mathcal{A}_j^\lambda(\mathcal{B}_j^\lambda) w_{i|j}^{\lambda+1}) \ , \tag{5.44}$$

$$P(\frac{\partial \mathcal{E}}{\partial \mathcal{B}_j^\lambda} \mid \frac{\partial \mathcal{E}}{\partial \mathcal{B}_j^{\lambda+1}}) = \mathbb{1}(\frac{\partial \mathcal{E}}{\partial \mathcal{B}_j^\lambda} = \sum_i \frac{\partial \mathcal{E}}{\partial \mathcal{B}_i^{\lambda+1}} D\mathcal{A}_j^\lambda(\mathcal{B}_j^\lambda) w_{i|j}^{\lambda+1}) \ , \tag{5.45}$$

$$\boxed{P(\delta_j^\lambda \mid \delta_j^{\lambda+1}) = \mathbb{1}(\delta_j^\lambda = \sum_i \delta_i^{\lambda+1} D\mathcal{A}_j^\lambda(\mathcal{B}_j^\lambda)) w_{i|j}^{\lambda+1})} \ . \tag{5.46}$$

First delta of iteration, belonging to output layer $\lambda = \Lambda - 1$:

$$\delta_j^{\Lambda-1} = \frac{\partial \mathcal{E}}{\partial \mathcal{B}_j^{\Lambda-1}} \tag{5.47}$$

$$= \frac{1}{nsam} \sum_{\sigma=0}^{nsam-1} \sum_{i=0}^{ny-1} 2(Y_i - y_i[\sigma]) D\mathcal{A}_i^{\Lambda-1}(\mathcal{B}_i^{\Lambda-1}) \delta(i,j) \tag{5.48}$$

$$= \frac{1}{nsam} \sum_{\sigma=0}^{nsam-1} 2(Y_j - y_j[\sigma]) D\mathcal{A}_j^{\Lambda-1}(\mathcal{B}_j^{\Lambda-1}) \tag{5.49}$$

Cute expression for derivative of sigmoid function:

$$D\text{smoid}(x) = \text{smoid}(x)(1 - \text{smoid}(x)) \tag{5.50}$$

5.4 General bnets instead of Markov chains induced by layered structure of NNs

$$P(\delta_{\underline{x}} \mid (\delta_{\underline{a}})_{\underline{a} \in ch(\underline{x})}) = \mathbb{1}(\delta_{\underline{x}} = \sum_{\underline{a} \in ch(\underline{x})} \delta_{\underline{a}} D\mathcal{A}_{\underline{x}}(\mathcal{B}_{\underline{x}}) w_{\underline{a}|\underline{x}}) \tag{5.51}$$

Reverse arrows of original bnet and define the TPM of nodes of "time reversed" bnet by

$$\boxed{P(\delta_{\underline{x}} \mid (\delta_{\underline{a}})_{\underline{a} \in pa(\underline{x})}) = \mathbb{1}(\delta_{\underline{x}} = \sum_{\underline{a} \in pa(\underline{x})} \delta_{\underline{a}} D\mathcal{A}_{\underline{x}}(\mathcal{B}_{\underline{x}}) w^T_{\underline{x}|\underline{a}})} \tag{5.52}$$

Chapter 6

Bell and Clauser-Horne Inequalities in Quantum Mechanics

Figure 6.1: bnet used to discuss Bell and Clauser-Horne inequalities in Quantum Mechanics.

I wrote an article about this in 2008 for my blog "Quantum Bayesian Networks". See Ref.[88].

Chapter 7

Berkson's Paradox

For more information about Berkson's Paradox (BP), see Ref.[110]

Figure 7.1: Bnet used to discuss Berkson's Paradox (BP). $\underline{a}$ and $\underline{b}$ are both causes of collider $\underline{x}$.

Consider the bnet of Fig.7.1. For that bnet, we have

$$P(a,b,x) = P(a)P(b)P(x|a,b) \ . \tag{7.1}$$

Summing Eq.(7.1) over x, we get

$$P(a,b) = P(a)P(b) \tag{7.2}$$

so $\underline{a}$ and $\underline{b}$ are independent. It follows that a can be ignored in calculating the probability of b; i.e.,

$$\boxed{P(b|a) = P(b)} \ . \tag{7.3}$$

However, a cannot be ignored in calculating the probability of b, if x is being held fixed; i.e.,

$$\boxed{P(b|a,x) \neq P(b|x)} \ . \tag{7.4}$$

Indeed,

$$P(b|a,x) = \frac{P(b)P(x|a,b)}{\sum_b P(b)P(x|a,b)} \tag{7.5}$$

whereas

$$P(b|x) = \frac{\sum_a P(a)P(b)P(x|a,b)}{\sum_{a,b} P(a)P(b)P(x|a,b)} \ . \tag{7.6}$$

The two boxed equations are what is referred to as BP.

BP is also called **collider bias** because $\underline{x}$ is a collider.

BP is also called **explaining away** in the special case that $\underline{a}, \underline{b}, \underline{x} \in \{false = 0, true = 1\}$. In that case, if $\underline{x}$ is fixed to true, and the cause $\underline{a}$ is known to be true, then the cause $\underline{b}$ is less likely to be true. For example, suppose a car engine fails ($\underline{x} = 1$) and the two most likely causes of the failure are alternator ($\underline{a}$) and battery ($\underline{b}$). Once we know that the alternator has failed ($\underline{a} = 1$), it is less likely that the battery is failing ($\underline{b} = 1$) than when the status of $\underline{a}$ was not known; i.e., $P(b = 1|x = 1, a = 1) < P(b = 1|x = 1)$.

Figure 7.2: Example of Berkson's paradox (BP).

Fig.7.2 presents an example of BP. The figure consists of a scatter plot with axes a =her attractiveness, b =her intelligence, for a female population of possible dates for you, assuming you are a male person. Let $x \in \{false = 0, true = 1\}$ = she goes out on a date with you. For the full population,

$$(a, b) \sim P(a, b) = P(a)P(b) \tag{7.7}$$

whereas for the population in the white swath,

$$(a, b) \sim P(a, b|x) = P(b|a, x)P(a|x) \neq P(b|x)P(a|x) \ . \tag{7.8}$$

As shown by Fig.7.2, BP is an example of **selection bias**. Selection bias happens when a non-representative subset of the total population is considered (i.e., selected).

Chapter 8

Binary Decision Diagrams

x_1	x_2	x_3	$f(A_{x_1,x_2,x_2})$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figure 8.1: Example of a binary dtree and its equivalent truth table. The truth table gives the values of $f(x_1, x_2, x_3) = \bar{x}_1(x_2 + \bar{x}_3) + x_1x_2$

This chapter is based mainly on Wikipedia article Ref.[115].

A **Binary Decision Diagram** (BDD) is a graph that represents visually, in a more concise manner, the information contained in a binary dtree or its equivalent truth table.[1]

Fig.8.1 shows an example of a binary dtree and its equivalent truth table. We will convert this tree into a BDD below. Each node asks a question with a binary (Boolean) answer. An answer of 0 (resp., 1) is indicated by a dashed (resp., full) arrow. The same question is asked by all nodes at the same level (i.e., depth) of the tree. In addition to being labeled by a question, each node is labeled uniquely by the random variable $\underline{A}_{x_1,x_2,\ldots,x_n}$, where n is the level of the node and x_i is the answer to the question x_i?. The leaves of the tree are square boxes that report

$$f(x_1, x_2, x_3) = \bar{x}_1(x_2 + \bar{x}_3) + x_1 x_2 \tag{8.1}$$

Figure 8.2: The same tree as in Fig.8.1, after dropping some labels that are not needed for our discussion of BDDs.

Fig.8.2 shows the same tree as in Fig.8.1, after dropping some labels that are not needed for our discussion of BDDs. Note that the **question ordering** of the tree is $x_1? < x_2? < x_3?$. Other question orderings such as $x_2? < x_1? < x_3?$ are possible. For a given truth table, some question orderings lead to a BDD that has the full exponential complexity 2^n of the tree, where n is the number of levels. Other question orderings might lead to BDDs that have lower (such as linear in n) complexity.

[1]Decision trees (dtrees) are discussed in Chapter 16

8.0.1 Conversion of Binary Tree into BDD

A BDD is obtained from a binary dtree by successive application of the following 3 rules:

1. Merge equivalent leaves (EL)
2. Merge isomorphic nodes (IN)
3. Eliminate parallel 0/1 arrows (PA) by merging source and target nodes of the parallel arrows.

Fig.8.3 gives an example of the application of rules EL and PA. Fig.8.4 gives an example of the application of the IN rule.

Figure 8.3: This is the result of applying the EL rule, followed by the PA rule, to Fig.8.2.

8.1 Equivalent Bnet

In Fig.8.5, (a) is a BDD and (b) is a LD (Linear Deterministic) bnet equivalent to (a). The structural equations of the LD bnet, printed in blue, are as follows:

$$\underline{y} = x_2\underline{c} + x_3\underline{e} + \bar{x}_3\underline{d} \tag{8.2}$$

$$\underline{d} = \bar{x}_2\underline{b} \tag{8.3}$$

$$\underline{e} = x_2\underline{b} \tag{8.4}$$

Figure 8.4: An example to illustrate the application of the IN rule

Figure 8.5: (a) is the BDD at the end of the conversion chain in Fig.8.3. (b) is a LD (Linear Deterministic) bnet that is equivalent to the BDD of (a). (b) is obtained by keeping a subset of the arrows in (a). The remaining arrows are given gain x_i (resp., $\bar{x}_i$) if they are full (resp., dashed) and originate from a node with question x_i?.

$$\underline{b} = \bar{x}_1 \underline{a} \tag{8.5}$$

$$\underline{c} = x_1 \underline{a} \tag{8.6}$$

Therefore[2]

$$\underline{y} = [\bar{x}_1(x_2 x_3 + \bar{x}_2 \bar{x}_3) + x_1 x_2]\underline{a} \tag{8.7}$$

The equivalence of (a) and (b) follows from the following transformation of Eq.(8.1)

$$\begin{aligned} f(x_1, x_2, x_3) =& \bar{x}_1(x_2 + \bar{x}_3) + x_1 x_2 && (8.8)\\ =& \bar{x}_1[x_2(x_3 + \bar{x}_3) + \bar{x}_3(x_2 + \bar{x}_2)] + x_1 x_2 \text{ (because } x + \bar{x} = 1 \text{ in base 2)} && (8.9)\\ =& \bar{x}_1[x_2(x_3) + \bar{x}_3(\bar{x}_2)] + x_1 x_2 \text{ , (because } 2x_2\bar{x}_3 = 0 \text{ in base 2)} && (8.10) \end{aligned}$$

Note that the right hand side of Eq.(8.10) gives the same 3 "Feynman histories" as the coefficient of $\underline{a}$ in Eq.(8.7).

$\underline{c} = x_2\underline{b} = x_2 x_1 \underline{a}$ (a) $\qquad$ $\underline{c} = (x_1 + x_2)\underline{a}$ (b)

Figure 8.6: Expressing the sum or product of two boolean variables x_1, x_2 as a LD bnet. In graphs (a) and (b), one can set $\underline{a} = 1$. Alternatively, in graph (b), one can omit the $\underline{a}$ node and set $\underline{b} = \underline{c} = 1$.

It's easy to express the sum or product of two boolean variables x_1, x_2 as a LD bnet. (see Fig.8.6) In general, any boolean polynomial can be expressed as a LD bnet. In particular, the sum of products and product of sums canonical forms of any boolean expression can be expressed thus.

[2]One can set $\underline{a} = 1$ in Fig.8.5 (b) and Eq.(8.7).

Chapter 9

Chow-Liu Trees and Tree Augmented Naive Bayes (TAN)

This chapter is mostly based on chapter 8 of Pearl's 1988 book Ref.[60]. See also Ref.[122] and references therein.

This chapter uses various Shannon Information Theory entropies. Our notation for these entropies is described in Chapter C.

9.1 Chow-Liu Trees

Chow-Liu trees refers to an algorithm for finding a bnet tree that fits an a priori given probability distribution as closely as possible.

Consider a bnet with n nodes $\underline{x}^n = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{n-1})$ such that $\underline{x}_i \in val(\underline{x}_i)$ for all i. Let its total probability distribution be $P_{\underline{x}^n}$. For simplicity, we will abbreviate $P_{\underline{x}^n}$ by P. Hence

$$P(x^n) = P_{\underline{x}^n}(x^n) \ . \tag{9.1}$$

Suppose we want to fit $P_{\underline{x}^n}$ by a tree bnet with nodes $\underline{t}^n = (\underline{t}_0, \underline{t}_1, \ldots, \underline{t}_{n-1})$ such that $\underline{t}_i \in val(\underline{t}_i) = val(\underline{x}_i)$ for all i. For simplicity, we will abbreviate $P_{\underline{t}^n}$ by P_T. Hence

$$P_T(x^n) = P_{\underline{t}^n}(x^n) \ . \tag{9.2}$$

Throughout this chapter, let $V = \{0, 1, \ldots, n-1\}$, the set of vertices. Suppose μ is a function $\mu : V \to V$ such that $\mu(i) < i$. Let $T_\mu = \{\underline{t}_{\mu(i)} \to \underline{t}_i : i \in V - \{0\}\}$. Then T_μ is a tree that spans (i.e., it includes all nodes) $\underline{t}^n$. Its root node is $\underline{t}_0$, because $\underline{t}_0$ has no parents. All other nodes $\underline{t}_i$ have exactly one parent, namely $\underline{t}_{\mu(i)}$. Let P_T, the total probability distribution for the tree, be parameterized by the function μ as follows:

$$P_T(x^n) = \prod_{i=0}^{n-1} P_T(x_i|x_{\mu(i)}) \ , \tag{9.3}$$

where, for the root node 0, $P_T(x_0|x_{\mu(0)}) = P_T(x_0)$.

Claim 27 *$D_{KL}(P \parallel P_T)$ is minimized over all probability distributions P_T that are expressible as Eq.(9.3) iff*

$$P_T(x_i|x_{\mu(i)}) = P(x_i|x_{\mu(i)}) \tag{9.4}$$

for all i, and

$$\sum_i H(\underline{x}_i : \underline{x}_{\mu(i)}) \tag{9.5}$$

is maximized over all μ.

proof:

$$\begin{aligned}
D_{KL}(P \parallel P_T) &= \sum_{x^n} P(x^n) \ln \frac{P(x^n)}{P_T(x^n)} && (9.6)\\
&= -\sum_{x^n} \sum_i P(x^n) \ln P_T(x_i|x_{\mu(i)}) + \sum_i P(x^n) \ln P(x^n) && (9.7)\\
&= -\sum_i \sum_{x_i, x_{\mu(i)}} P(x_i, x_{\mu(i)}) \ln P_T(x_i|x_{\mu(i)}) - H(\underline{x}^n) && (9.8)\\
&= -\sum_i \sum_{x_{\mu(i)}} P(x_{\mu(i)}) \left[\sum_{x_i} P(x_i|x_{\mu(i)}) \ln P_T(x_i|x_{\mu(i)}) \right] - H(\underline{x}^n) \ . && (9.9)
\end{aligned}$$

Now note that

$$\sum_{x_i} P(x_i|x_{\mu(i)}) \ln \frac{P(x_i|x_{\mu(i)})}{P_T(x_i|x_{\mu(i)})} \geq 0 \tag{9.10}$$

and this inequality becomes an equality iff

$$P(x_i|x_{\mu(i)}) = P_T(x_i|x_{\mu(i)}) \ . \tag{9.11}$$

Therefore

$$D_{KL}(P \parallel P_T) \geq -\sum_i \underbrace{\sum_{x_{\mu(i)}} P(x_{\mu(i)}) \left[\sum_{x_i} P(x_i|x_{\mu(i)}) \ln P(x_i|x_{\mu(i)}) \right]}_{=H(\underline{x}_i|\underline{x}_{\mu(i)}) = H(\underline{x}_i : \underline{x}_{\mu(i)}) - H(\underline{x}_i)} - H(\underline{x}^n) \ , \tag{9.12}$$

and this inequality becomes an equality iff Eq.(9.11) is satisfied.

Note from the last equation that

$$\operatorname*{argmin}_{\mu} D_{KL}(P \parallel P_T) = \operatorname*{argmax}_{\mu} \sum_i H(\underline{x}_i : \underline{x}_{\mu(i)}) \ . \tag{9.13}$$

QED

Claim 28

$$\operatorname*{argmin}_{\mu} H(\underline{x}^n) = \operatorname*{argmax}_{\mu} \sum_i H(\underline{x}_i : \underline{x}_{\mu(i)}) \tag{9.14}$$

proof:

$$\begin{aligned}
H(\underline{x}^n) &= -\sum_{x^n} P(x^n) \sum_i \ln P(x_i|x_{\mu(i)}) && (9.15)\\
&= -\sum_i \sum_{x_i, x_{\mu(i)}} P(x_i, x_{\mu(i)}) \ln P(x_i|x_{\mu(i)}) && (9.16)\\
&= -\sum_i \sum_{x_i, x_{\mu(i)}} P(x_i, x_{\mu(i)}) \left[\ln \frac{P(x_i|x_{\mu(i)})}{P(x_i)} + \ln P(x_i) \right] && (9.17)\\
&= -\sum_i \left[H(\underline{x}_i : \underline{x}_{\mu(i)}) - H(\underline{x}_i) \right] && (9.18)\\
&= \sum_i H(\underline{x}_i) - \sum_i H(\underline{x}_i : \underline{x}_{\mu(i)}) && (9.19)
\end{aligned}$$

QED

The meaning of Claims 27 and 28 is as follows. If $D_{KL}(P \parallel P_T)$ is minimized over all P_T, then

1. P_T inherits its TPMs from P, and
2. P_T gets its structure, which is being parameterized by the function μ, by maximizing the score given by

$$\text{score} = \sum_i H(\underline{x}_i : \underline{x}_{\mu(i)}) \ . \tag{9.20}$$

(mutual information $H(\underline{a} : \underline{b})$ measures correlation between $\underline{a}$ and $\underline{b}$). Maximizing the score is the same as minimizing the entropy $H(\underline{x}^n)$ over all the structures μ. (i.e., finding least complex structure).

So far, we have studied the properties of those probability distributions P_T for a tree bnet that best approximates an a priori given probability distribution P, but we haven't yet described how to build a Chow-Liu tree based on empirical data. Next we give Chow-Liu's algorithm for doing so.

1. **Find MST using Kruskal's algorithm[1]. (see Fig.9.1)**
   Calculate weights $w_{i,j} = H(\underline{x}_i : \underline{x}_j)$ for all $i, j \in V$ and store them in a dictionary D that maps edges to weights.
   Order D by weight size.
   Let T be a list of the edges in the tree. Initialize T to empty.
   Repeat this until T has $n-1$ elements:
   > Remove largest weight w from D and corresponding edge e.
   > Add e to T if $\{e\} \cup T$ has no loops. Otherwise discard e and w.

2. **Give directions to edges in T. (see Fig.9.2)**
   Let DT be a list of directed edges. Initialize DT to empty.
   Choose any node as root node.
   Point arrows along edges in T, away from root node.
   Add new arrows to DT.
   Repeat this until DT has $n-1$ elements:
   > Point arrows along edges in T, away from leaf nodes of current DT.
   > Add new arrows to DT.

Figure 9.1: Example of finding MST (maximum spanning tree)

Figure 9.2: Example of giving directions to edges of spanning tree.

Nodes in a Chow-Liu tree can be rated in terms of their relative importance. Here are 2 possible metrics for measuring the importance of a node $\underline{a}$:

$$N_{nb}(\underline{a}) = \text{ number of neighbors of } \underline{a} \tag{9.21}$$

[1]Kruskal's algorithm is one several famous algorithms (Prim's algo is another one) for finding an MST (maximum or minimum spanning tree). An MST algorithm takes an undirected graph with weights along its edges as input. It then finds a tree subgraph (i.e., subset of the edges of the graph with no loops) that (1) spans the graph (i.e., includes every vertex of the graph) and (2) maximizes (or minimizes) the sum of weights among all possible tree subgraphs. For more information, see Ref[163] and references therein, or any other of numerous explanations of MST in the Internet.

$$\text{traffic}(\underline{a}) = \sum_{\underline{n} \in nb(\underline{a})} H(\underline{a} : \underline{n}) \tag{9.22}$$

For example, to get a tree with low depth, one can choose as the root node the node which has largest N_{nb}, and if there are several with the same largest N_{nb}, choose out of those the one with the largest traffic.

9.2 Tree Augmented Naive Bayes (TAN)

Recall from Chapter 67 that a Naive Bayes bnet consists of a class node $\underline{c}$ with n children nodes $\underline{x}^n$, called the feature nodes. A Tree Augmented Naive Bayes (TAN) bnet is a Naive Bayes bnet with a tree grafted onto it like a chimera. More precisely, one starts with a Naive Bayes bnet and adds arrows between the feature nodes. The arrows are added in such a way that the TAN bnet sans node $\underline{c}$ constitutes a tree. It's not the most well motivated bnet in human history, but at least it adds a bit of correlation between the feature nodes of the Naive Bayes bnet. Those nodes are independent at fixed $\underline{c}$ in the Naive Bayes bnet, but are no longer so in the TAN bnet. See Figs.9.3 and 9.4 for an example of a TAN bnet.

Figure 9.3: bnet for Naive Bayes with 4 feature nodes and another bnet for a tree made of the same feature nodes.

Figure 9.4: TAN bnet constructed by merging Naive Bayes bnet and tree bnet of Fig.9.3.

The total probability distribution P_{TAN} for a TAN bnet can be parameterized as follows.

$$P_{TAN}(x^n, c) = P_{TAN}(c) \prod_{i=0}^{n-1} P_{TAN}(x_i | x_{\mu(i)}, c) \ . \tag{9.23}$$

As with Chow Liu trees, we can attempt to find a TAN bnet whose total probability $P_{TAN} = P_{\underline{t}^n,\underline{c}}$ best approximates an a priori given probability distribution $P = P_{\underline{x}^n,\underline{c}}$.

Note that

Claim 29

$$\operatorname*{argmin}_{\mu} H(\underline{x}^n, \underline{c}) = \operatorname*{argmax}_{\mu} \sum_i H(\underline{x}_i : \underline{x}_{\mu(i)} | \underline{c}) \tag{9.24}$$

proof:

$$\begin{aligned} H(\underline{x}^n, \underline{c}) &= -\sum_{x^n,c} P(x^n, c) \left[\ln P(c) + \sum_i \ln P(x_i | x_{\mu(i)}, c) \right] && (9.25) \\ &= -\sum_{x^n,c} P(x^n, c) \left[\ln P(c) + \sum_i \ln \left(\frac{P(x_i, x_{\mu(i)}|c)}{P(x_i|c) P(x_{\mu(x_i)}|c)} P(x_i|c) \right) \right] && (9.26) \\ &= \sum_i H(\underline{x}_i, \underline{c}) - \sum_i H(\underline{x}_i : \underline{x}_{\mu(i)} | \underline{c}) && (9.27) \end{aligned}$$

QED

Following the same line of reasoning that we followed for Chow-Liu trees, we conclude that:

If $D_{KL}(P \parallel P_{TAN})$ is minimized over all P_{TAN}, then

1. P_{TAN} inherits its TPMs from P, and
2. P_{TAN} gets its structure, which is being parameterized by the function μ, by maximizing the score defined by

$$\text{score} = \sum_i H(\underline{x}_i : \underline{x}_{\mu(i)} | \underline{c}) \tag{9.28}$$

One can build a TAN bnet from empirical data as follows:

Calculate a Chow-Liu Tree for each $c \in val(\underline{c})$. For each of those trees, create a TAN bnet, and calculate its score given by Eq.(9.28). Keep the TAN bnet with the largest score.

Chapter 10

Control Theory (linear, deterministic)

This chapter is based on Ref.[104] and [180].

We will assume that the reader has read Section C.45 on Laplace Transforms and Section C.46 on Z-transforms.

By **discrete time or discretizing time**, we mean sampling all signals at discrete times separated by a finite time interval T called the **sampling time**.

Control Theory (CT) studies the optimal control of systems with feedback. The systems studied can be

- linear or non-linear,
- deterministic or stochastic,
- continuous time (analog) or discrete time (digital).[1].

This chapter will deal with linear deterministic systems of either the analog or digital kind.

As explained in Chapter 26, dynamical bnets and feedback are two ways of viewing the same physical phenomenon. Also, there are numerous examples of dynamical bnets in this book: Kalman filters, Hidden Markov Models, Reinforcement Learning, Recursive Neural Nets, to name a few. Hence, a chapter on CT is very pertinent to this book.

Two acronyms commonly used in CT books are: **SISO** (single input single output) and **MIMO** (multiple input multiple output). We will consider both SISO and MIMO systems in this chapter.

[1] A signal $x(t)$ is a function of time t. We can discretize t (Δt, sampling), or discretize x (Δx, quantization), or both. We will use the word "digital" or "sampled" to describe a theory with discretized t, but continuous x. Sometimes, the word "digital" is used instead to describe a theory with both x and t discretized.

Another distinction commonly made in CT books is between **time-variant** and **time-invariant** systems. We will explain what those terms mean later on in this chapter. This chapter will consider both types of systems.

10.1 Basic feedback model

CT uses feedback to control a **system or process**. Fig.10.1 shows a very basic feedback model, represented with 3 equivalent diagrams.

Figure 10.1: Basic feedback model represented as: (a) a wired time-dependent boxes diagram, (b) two time-slices of a dynamical bnet (see Chapter 26), and (c) a "rolled" dynamical bnet with feedback cycles.

The diagrams of Fig.10.1 represent graphically the following system of equations. Here $t \in [0, \infty]$ is time and $r, u, y : [0, \infty] \to \mathbb{C}$.

$$\begin{cases} u(t) = F_1(y(t), r(t), t) \\ \partial_t y(t) = F_2(y(t), u(t), t) \end{cases} \tag{10.1}$$

If we approximate the time derivative of $y(t)$ by

$$\partial_t y(t) \approx \frac{y(t + \Delta t) - y(t)}{\Delta t} \tag{10.2}$$

and we set

$t_k = t$, $t_{k+1} = t + \Delta t$, $0 < \Delta t << 0$,

$f(t_k) = f^{[k]}$ for $f = r, y, u$,

then we get Fig.10.1

The TPMs, printed in blue, of the bnet in Fig.10.1(b), are as follows:

$$P(r^{[k]}) = \text{given} \tag{10.3}$$

$$P(u^{[k]}|y^{[k]}, r^{[k]}) = \delta(\quad u^{[k]} - F_1(y^{[k]}, r^{[k]}, t_k) \quad) \tag{10.4}$$

$$P(y^{[k+1]}|y^{[k]}, u^{[k]}) = \delta(\quad y^{[k+1]} - y^{[k]} - \Delta t F_2(y^{[k]}, u^{[k]}, t_k) \quad) \tag{10.5}$$

10.2 Classical model (analog)

Figure 10.2: Classical Model represented with the same 3 types of diagrams as Fig.10.1. Bnet (b) doesn't show all the arrows. In reality, for any arrow $a^{[k]} \to b^{[k+1]}$ that points from the time-slice t_k to the time-slice t_{k+1}, there should be arrows $a^{[j]} \to b^{[k+1]}$ for $j \in \{0, 1, 2, \cdots, k\}$. That's because the classical model is defined in terms of convolutions, and a convolution at time t requires memory for all times between 0 and t.

A **classical model** is a bunch of wired convolution boxes. See Fig.10.2 for 3 graphical representations of the classical model.

Let $t \in [0, \infty]$ =time and $f, e, r, u, y : [0, \infty] \to \mathbb{C}$. The diagrams of Fig.10.2 represent graphically the following system of equations.

$$\begin{cases} f(t) = (F \circledast y)(t) \\ e(t) = r(t) - f(t) \\ u(t) = (C \circledast e)(t) \\ y(t) = (\Pi \circledast u)(t) \end{cases} \tag{10.6}$$

where $(f \circledast g)(t)$ denotes a covolution, as defined in Section C.45 on Laplace transforms.

The TPMs, printed in blue, of the bnet in Fig.10.2, are as follows:

$$P(r(t)) = \text{given} \tag{10.7}$$

$$P(f(t)|y(\cdot)) = \delta(\quad f(t) - \Pi[y](t) \quad) \tag{10.8}$$

$$P(e(t)|r(t), f(t)) = \delta(\quad e(t) - [r(t) - f(t)] \quad) \tag{10.9}$$

$$P(u(t)|e(\cdot)) = \delta(\quad u(t) - C[e](t) \quad) \tag{10.10}$$

$$P(y(t)|u(\cdot)) = \delta(\quad y(t) - \Pi[u](t) \quad) \tag{10.11}$$

If we take the Laplace transform of Eqs.(10.6, we get

$$\begin{cases} \widetilde{f}(s) = \widetilde{F}(s)\widetilde{y}(s) \\ \widetilde{e}(s) = \widetilde{r}(s) - \widetilde{f}(s) \\ \widetilde{u}(s) = \widetilde{C}(s)\widetilde{e}(s) \\ \widetilde{y}(s) = \widetilde{\Pi}(s)\widetilde{u}(s) \end{cases} \tag{10.12}$$

Thus

$$\widetilde{H}_{y|e} = \frac{\widetilde{y}}{\widetilde{e}} = \widetilde{\Pi}\widetilde{C} \tag{10.13}$$

$$\widetilde{y} = \widetilde{H}_{y|e}[\widetilde{r} - \widetilde{F}\widetilde{y}] \tag{10.14}$$

$$[1 + \widetilde{H}_{y|e}\widetilde{F}]\widetilde{y} = \widetilde{H}_{y|e}\widetilde{r} \tag{10.15}$$

$$\widetilde{H}_{y|r}(s) = \frac{\widetilde{y}(s)}{\widetilde{r}(s)} \quad \text{(output/input)} \tag{10.16}$$

$$= \frac{\widetilde{H}_{y|e}}{1+\widetilde{H}_{y|e}\widetilde{F}} \tag{10.17}$$

$\widetilde{H}_{y|e}$ and $\widetilde{H}_{y|r}(s)$ are both called **gain or transfer functions**. $\widetilde{H}_{y|e}$ is called the **open loop gain** and $\widetilde{H}_{y|r}(s)$ is called the **closed loop gain**. If $|H_{y|r}|$ is less than (resp., more than) 1, we say that there is **negative feedback** (resp., **positive feedback**) because the $\widetilde{r}$ signal is reduced (resp., magnified). If the open loop gain $|\widetilde{H}_{y|e}| >> 1$, then the closed loop gain $|\widetilde{H}_{y|r}| \approx \frac{1}{|\widetilde{F}|}$. So negative (resp., positive) feedback occurs if $|\widetilde{F}| > 1$ (resp., $|\widetilde{F}| < 1$)

A common type of controller box $\widetilde{C}(s)$ called the **Proportional-Integral-Derivative (PID) Controller** is defined as

$$u(t) = K_\Pi e(t) + K_I \int_0^t d\tau\ e(\tau) + K_D \partial_t e(t) \tag{10.18}$$

The Laplace transform of the PID controller is

$$\widetilde{u}(s) = K_\Pi \widetilde{e}(s) + K_I \frac{\widetilde{e}(s)}{s} + K_D(s\widetilde{e}(s) - e(0^+)) \tag{10.19}$$

$$= \underbrace{\left(\underbrace{K_\Pi}_{\substack{\text{proportinal}\\ \text{controller}}} + \underbrace{\frac{K_I}{s}}_{\substack{\text{integrator}\\ \text{controller}}} + \underbrace{K_D s}_{\substack{\text{differentiator}\\ \text{controller}}} \right)}_{\widetilde{C}(s),\ \text{PID controller}} \widetilde{e}(s) \quad (\text{assume } e(0^+) = 0) \tag{10.20}$$

Claim 30 *A PID controller has unit gain ($\widetilde{H}_{y|r}(s) = 1$) if:*

$$K_\Pi = 2K, \quad K_D = KT, \quad K_I = \frac{K}{T} \tag{10.21}$$

$$\widetilde{\Pi}(s) = \frac{1}{K(1+sT)} \tag{10.22}$$

and

$$\widetilde{F}(s) = \frac{1}{1+sT} \tag{10.23}$$

proof:

$$\widetilde{C} = \frac{K}{sT}\left(2sT + 1 + (sT)^2\right) \tag{10.24}$$

$$= \frac{K}{sT}(1 + sT)^2 \tag{10.25}$$

so

$$1 + \widetilde{\Pi}\widetilde{C}\widetilde{F} = 1 + \frac{1}{sT} \tag{10.26}$$

$$= \frac{1}{sT}(1 + sT) \tag{10.27}$$

$$= \widetilde{\Pi}\widetilde{C} \tag{10.28}$$

Hence,

$$\widetilde{H}_{y|r}(s) = 1 \tag{10.29}$$

QED

10.3 Modern model (analog)

Figure 10.3: Modern Model represented with the same 3 types of diagrams as Fig.10.1.

A **modern model (a.k.a. state space model)** is a bunch of wired time-dependent boxes, some with first order time derivatives. See Fig.10.3 for 3 graphical representations of the modern model.

Let
$t \in [0, \infty]$ =time
$u : [0, \infty] \to \mathbb{C}^{nu}$
$x : [0, \infty] \to \mathbb{C}^{nx}$
$y : [0, \infty] \to \mathbb{C}^{ny}$
for some integers nu, nx, ny. The diagrams of Fig.10.3 represent graphically the following system of equations.

$$\begin{cases} \partial_t x(t) = F_{AB}(x(t), u(t), t) \\ y(t) = F_{CD}(x(t), u(t), t) \end{cases} \tag{10.30}$$

These equations are called the **state space equations** and $x(t)$ is called the **state** of the system. The equation for $\partial_t x(t)$ is called the **state equation** and the one for $y(t)$ is called the **output equation**.

The TPMs, printed in blue, of the bnet in Fig.10.3(b), are as follows:

$$P(u^{[k]}) = \text{given} \tag{10.31}$$

$$P(x^{[k+1]} \mid x^{[k]}, u^{[k]}) = \delta(\quad x^{[k+1]} - x^{[k]} - \Delta t F_{AB}(x^{[k]}, u^{[k]}, t_k) \quad) \tag{10.32}$$

$$P(y^{[k]} \mid x^{[k]}, u^{[k]}) = \delta(\quad y^{[k]} - F_{CD}(x^{[k]}, u^{[k]}, t_k) \quad) \tag{10.33}$$

Henceforth, assume F_{AB} and F_{CD} are as follows. This is called the **linear** case.

$$\begin{cases} F_{AB}(x(t), u(t)) = A(t)x(t) + B(t)u(t) \\ F_{CD}(x(t), u(t)) = C(t)x(t) + D(t)u(t) \end{cases} \tag{10.34}$$

for some matrices $A(t), B(t), C(t), D(t)$. In the linear case, the modern model is described by the following equations:

$$\begin{cases} \partial_t x(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{10.35}$$

If the matrices $A(t), B(t), C(t), D(t)$ depend on (resp., are independent of) time t, we say the system is **time-variant** (resp., **time-invariant**). Next, we solve the differential equation for $x(t)$, for both the time-invariant and variant cases.

- time-invariant case

  Taking the Laplace transform of the first equation of Eqs.(10.35), we get

$$s\widetilde{x}(s) - x(0) = A\widetilde{x}(s) + B\widetilde{u}(s) \tag{10.36}$$

Hence, the Laplace transform of Eqs.(10.35) is

$$\begin{cases} \widetilde{x}(s) = (sI - A)^{-1}x(0) + (sI - A)^{-1}B\widetilde{u}(s) \\ \widetilde{y}(s) = C\widetilde{x}(s) + D\widetilde{u}(s) \end{cases} \tag{10.37}$$

If we define the transfer function $\widetilde{H}_{y|u}(s)$ by

$$\underbrace{\widetilde{y}(s)}_{output} = \widetilde{H}_{y|u}(s)\underbrace{\widetilde{u}(s)}_{input} \tag{10.38}$$

then, assuming $x(0) = 0$,

$$\widetilde{H}_{y|u}(s) = C(sI - A)^{-1}B + D \tag{10.39}$$

Note from the last equation that the set of poles of $\widetilde{H}_{y|u}(s)$ is a subset of the set of eigenvalues of A.

If we set

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -b_0 & -b_1 & -b_2 & \cdots & -b_{n-1} \end{bmatrix} \tag{10.40}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{10.41}$$

$$C = \begin{bmatrix} a_0 & a_1 & \cdots & a_{m-1} \end{bmatrix} \tag{10.42}$$

$$D = 0 \tag{10.43}$$

then one can show that (the laborious part is inverting $s - A$ algebraically)

$$\widetilde{H}_{y|u}(s) = C(sI - A)^{-1}B + D \tag{10.44}$$

$$= \frac{s^m + a_{m-1}s^{m-1} + \cdots + a_0}{s^n + b_{n-1}s^{n-1} + \cdots + b_0} \tag{10.45}$$

This makes it possible to start with a desired transfer function, and build an analog modern model that achieves it.

Claim 31

$$x(t) = e^{A(t-t_0)}x(t_0) + e^{A(t-t_0)}\int_{t_0}^{t} d\tau\ e^{-A(\tau-t_0)}Bu(\tau) \tag{10.46}$$

where

$$e^{At} = \sum_{k=0}^{\infty}\frac{t^k}{k!}A^k \tag{10.47}$$

Hence, setting $t \geq t_0 = 0$,

$$e^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}] \tag{10.48}$$

$$e^{At}\int_{0}^{t} d\tau\ e^{-A\tau}Bu(\tau) = \mathcal{L}^{-1}[(sI - A)^{-1}B\widetilde{u}(s)] \tag{10.49}$$

proof: To check Eq.(10.46), just take the time derivative of both sides and use $\partial_t \int^t d\tau\ f(\tau) = f(t)$
QED

- time-variant case

Claim 32

$$x(t) = \mathcal{E}(t,t_0)x(t_0) + \int_{t_0}^{t} d\tau\ \mathcal{E}(t,\tau)B(\tau)u(\tau) \tag{10.50}$$

where the **state transition matrix (a.k.a. evolution matrix)** $\mathcal{E}(t,t_0)$ *satisfies*

$$\partial_t\mathcal{E}(t,t_0) = A(t) \tag{10.51}$$

and

$$\mathcal{E}(t,t) = 1 \tag{10.52}$$

proof: To prove Eq.(10.50), just differentiate both sides of it with respect to t, and use $\partial_t \int^t d\tau\ f(\tau) = f(t)$
QED

In the time-invariant case,

$$\mathcal{E}(t,t_0) = e^{A(t-t_0)} \tag{10.53}$$

10.4 Classical model (digital)

In this section, we will define the classical model with discrete rather than continuous time.

If we discretize time, then the equivalent of Eqs.(10.6) is

$$\begin{cases} f^{[n]} = (F \circledast y)^{[n]} \\ e^{[n]} = r^{[n]} - f^{[n]} \\ u^{[n]} = (C \circledast e)^{[n]} \\ y^{[n]} = (\Pi \circledast u)^{[n]} \end{cases} \tag{10.54}$$

where $(x \circledast y)^{[n]}$ denotes a discrete convolution, as defined in Section C.46 on Z-transforms. And if we take the Z-transform of Eqs.(10.54), we get

$$\begin{cases} \widetilde{f}(z) = \widetilde{F}(z)\widetilde{y}(z) \\ \widetilde{e}(z) = \widetilde{r}(z) - \widetilde{f}(z) \\ \widetilde{u}(z) = \widetilde{C}(z)\widetilde{e}(z) \\ \widetilde{y}(z) = \widetilde{\Pi}(z)\widetilde{u}(z) \end{cases} \tag{10.55}$$

The **digital PID controller box** $\widetilde{C}(z)$ is given by

$$\widetilde{u}(z) = \underbrace{\left(\underbrace{K_\Pi}_{\substack{\text{proportinal}\\ \text{controller}}} + \underbrace{K_I \frac{T}{2}\left(\frac{z+1}{z-1}\right)}_{\substack{\text{integrator}\\ \text{controller}}} + \underbrace{K_D \frac{1}{T}\left(\frac{z-1}{z}\right)}_{\substack{\text{differentiator}\\ \text{controller}}} \right)}_{\widetilde{C}(z)\text{, PID controller}} \widetilde{e}(z) \quad (\text{assume } e(0^+) = 0) \tag{10.56}$$

10.5 Modern model (digital)

In this section, we will define the modern model with discrete rather than continuous time. We will do this 2 different ways. First, we will approximate the time derivatives in all differential equations with a discrete approximation. This approach is interesting and instructive but not perfect, because it's an approximation (i.e., it assumes the sampling time $T = \Delta t$ is very small). Second, we will replace all differential equations by difference equations, and solve the latter exactly. The second approach is better for most purposes, because it gives exact results (i.e., correct to all orders in T), instead of approximations.

10.5.1 Discretizing derivatives

If we approximate $\partial_t x(t)$ by

$$\partial_t x(t) = \lim_{\Delta t \to 0} \frac{x(t+\Delta t) - x(t)}{\Delta t} \tag{10.57}$$

then Eqs.(10.35) reduce to

$$\begin{cases} x(t+\Delta t) = (1+\Delta t A)x(t) + \Delta t B u(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{10.58}$$

Using the new notation

$$\widehat{A} = e^{A\Delta t} \approx 1 + \Delta t A, \tag{10.59}$$

$$\widehat{B} \approx \Delta t B \tag{10.60}$$

$$\Delta t = T, \quad t = nT \tag{10.61}$$

$$X(t) = X(nT) = X^{[n]} \quad \text{for } X = x, u, y \tag{10.62}$$

we get

$$\begin{cases} x^{[n+1]} = \widehat{A} x^{[n]} + \widehat{B} u^{[n]} \\ y^{[n]} = C x^{[n]} + D u^{[n]} \end{cases} \tag{10.63}$$

Setting $t = t_0 + T$ in Eq.(10.46), we get

$$x(t_0+T) = e^{AT} x(t_0) + e^{AT} \int_{t_0}^{t_0+T} d\tau \; e^{-A(\tau - t_0)} B u(\tau) \tag{10.64}$$

$$= e^{AT} x(t_0) + e^{AT} \int_0^T d\tau \; e^{-A\tau} B u(\tau + t_0) \quad (\text{substitute } \tau \to \tau + t_0) \tag{10.65}$$

Now setting

$$t_0 = nT \tag{10.66}$$

and

$$u(\tau + t) \approx u(t) \quad \text{for } \tau \in [0, T] \tag{10.67}$$

we get

$$x^{[n+1]} = \underbrace{e^{AT}}_{\widehat{A}} x^{[n]} + \underbrace{e^{AT} \left[\int_0^T d\tau \; e^{-A\tau} \right] B}_{\widehat{B}} \; u^{[n]} \tag{10.68}$$

where

$$\widehat{A} = e^{AT} \tag{10.69}$$

and

$$\begin{aligned}
\widehat{B} &= e^{AT}\left[\int_0^T d\tau\, e^{-A\tau}\right] B && (10.70)\\
&= e^{AT}(-A)^{-1}(e^{-AT} - I)B && (10.71)\\
&= A^{-1}(\widehat{A} - 1)B && (10.72)
\end{aligned}$$

When $T << 1$, $\widehat{A} - 1 \approx AT$ so $A^{-1}(\widehat{A} - 1)B \approx BT$.

10.5.2 Solving Difference Equation

- time-invariant case

  Consider the following difference equation taken from Eqs.(10.63)

$$x^{[n+1]} = \widehat{A}x^{[n]} + \widehat{B}u^{[n]} \tag{10.73}$$

  To solve this difference equation, we notice that

$$x^{[1]} = \widehat{A}x^{[0]} + \widehat{B}u^{[0]} \tag{10.74}$$

$$\begin{aligned}
x^{[2]} &= \widehat{A}x^{[1]} + \widehat{B}u^{[1]} && (10.75)\\
&= \widehat{A}^2x^{[0]} + \widehat{A}\widehat{B}u^{[0]} + +\widehat{B}u^{[1]} && (10.76)
\end{aligned}$$

$$\begin{aligned}
x^{[3]} &= \widehat{A}x^{[2]} + \widehat{B}u^{[2]} && (10.77)\\
&= \widehat{A}^3x^{[0]} + \widehat{A}^2\widehat{B}u^{[0]} + \widehat{A}\widehat{B}u^{[1]} + \widehat{B}u^{[2]} && (10.78)
\end{aligned}$$

  The general pattern is clear. In general,

$$x^{[n]} = \widehat{A}^n x^{[0]} + \sum_{k=0}^{n-1} \widehat{A}^{n-k-1}\widehat{B}u^{[k]} \tag{10.79}$$

  It is also possible to solve Eqs.(10.63) using Z-transforms (See Section C.46). The Z-transform of the first of those two equations is

$$z(\widetilde{x}(z) - x^{[0]}) = \widehat{A}\widetilde{x}(z) + \widehat{B}\widetilde{u}(z) \tag{10.80}$$

Therefore, the Z-transform of Eqs.(10.63) is[2]

$$\begin{cases} \widetilde{x}(z) = (zI - \widehat{A})^{-1} z x^{[0]} + (zI - \widehat{A})^{-1}\widehat{B}\widetilde{u}(z) \\ \widetilde{y}(z) = C\widetilde{x}(z) + D\widetilde{u}(z) \end{cases} \tag{10.81}$$

From this we can get the transfer matrix $\widetilde{H}_{y|u}(z)$. Assuming $x^{[0]} = 0$,

$$\widetilde{y}(z) = \underbrace{\left(\widehat{C}(zI - \widehat{A})^{-1}\widehat{B} + \widehat{D}\right)}_{\widetilde{H}_{y|u}(z)} \widetilde{u}(z) \tag{10.82}$$

As for the analog modern model, it is possible to find for the digital modern model, matrices $\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}$ that produce a transfer function of a desired form.

- time-variant case

  We can also give a discrete version of Claim 32

  Claim 33

$$x^{[n]} = \mathcal{E}^{[n,n_0]} x^{[n_0]} + \sum_{k=0}^{n-1} \mathcal{E}^{[n,k+1]} B^{[k]} u^{[k]} \tag{10.83}$$

  where

$$\mathcal{E}^{[n+1,n_0]} = \widehat{A}^{[n]} \mathcal{E}^{[n,n_0]} \tag{10.84}$$

  and

$$\mathcal{E}^{[n_0,n_0]} = I \tag{10.85}$$

  Hence

$$\begin{aligned} \mathcal{E}^{[n,n_0]} &= \widehat{A}^{[n-1]}\widehat{A}^{[n-2]}\widehat{A}^{[n-3]}\cdots\widehat{A}^{[n_0]} && (10.86) \\ &= \prod_{k\in\mathbb{Z}^{[1,n-n_0]}} \widehat{A}^{[n-k]} && (10.87) \end{aligned}$$

  proof: Left to reader.
  QED

[2]Notice the z factor multipying $x^{[0]}$. There is no counterpart s factor multiplying $x(0)$ in the analog case. That's because $z = e^{sT}$.

10.6 Higher than first order differential (or difference) equations

If in the analog, time-invariant modern model, we express $x(t)$ in terms of $y(t)$ and $u(t)$

$$x(t) = C^{-1}[y(t) - Du(t)] \tag{10.88}$$

and then we plug this into the equation for $\partial_t x(t)$, we get

$$\partial_t \left(\underbrace{C^{-1}[y(t) - Du(t)]}_{x(t)} \right) = A \underbrace{C^{-1}[y(t) - Du(t)]}_{x(t)} + Bu(t) \tag{10.89}$$

Hence, it appears that this model can only accommodate a first order time derivative of the output $y(t)$. Next we give a transformation whereby a model with $y(t)$ derivatives that are higher than 1st order, can be re-expressed in a form that only has 1st order time derivatives. We will also give an analogous result for the digital (instead of analog) time-invariant modern model; that is, we will show that those digital models can accommodate higher than 1st order time differences.

10.6.1 Differential Equations

Let

$$\Omega = \partial_t^3 + a_2 \partial_t^2 + a_1 \partial_t + a_0 \tag{10.90}$$

where $a_0, a_1, a_2 \in \mathbb{C}$ are independent of time t, and consider the **linear, constant coefficients (LCC) ordinary differential equation (ODE)**:

$$\Omega y(t) = u(t) \tag{10.91}$$

Assume $f(t)$ satisfies

$$\Omega f(t) = 0 \tag{10.92}$$

Let

$$\underbrace{\partial_t^3 y(t)}_{\partial_t x_2 + \partial_t^3 f} + a_2 \underbrace{\partial_t^2 y(t)}_{x_2 + \partial_t^2 f} + a_1 \underbrace{\partial_t y(t)}_{x_1 + \partial_t f} + a_0 \underbrace{y(t)}_{x_0 + f} = u(t) \tag{10.93}$$

and

$$x(t) = \begin{bmatrix} x_0(t) \\ x_1(t) \\ x_2(t) \end{bmatrix} \tag{10.94}$$

Then

$$\partial_t x(t) = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{B} u(t) \tag{10.95}$$

$$y(t) = x_0(t) + f(t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{C} x(t) + f(t) \tag{10.96}$$

If $\Omega u = 0$, we can define $f = Du$. Otherwise, define $f = 0$.

What's going on here, from a dynamical bnet perspective, is that we are defining the slices to have enough variables so that there only needs to be memory from one slice to the previous slice, instead of, to the previous 3 slices.

10.6.2 Difference Equations

Let

$$\underbrace{y^{[n+3]}}_{x_3^{[n+1]}} + a_2 \underbrace{y^{[n+2]}}_{x_2^{[n]}} + a_1 \underbrace{y^{[n+1]}}_{x_1^{[n]}} + a_0 \underbrace{y^{[n]}}_{x_0^{[n]}} = u^{[n]} \tag{10.97}$$

and

$$x^{[n]} = \begin{bmatrix} x_0^{[n]} \\ x_1^{[n]} \\ x_2^{[n]} \end{bmatrix} \tag{10.98}$$

Then

$$x^{[n+1]} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}}_{A} x^{[n]} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{B} u^{[n]} \tag{10.99}$$

$$y^{[n]} = x_0^{[n]} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{C} x^{[n]} \tag{10.100}$$

10.7 Time-Invariance, Causality, Stability

A linear system with input $r(t)$ and output $y(t)$ has a kernel function $H(t, \tau)$ such that

$$y(t) = \int_0^{\Lambda} H(t, \tau) r(\tau) \tag{10.101}$$

In general, r and y are vectors, so $H(t,\tau)$ is a matrix.

A system is **time-invariant** if $H(t,\tau) = H(t-\tau)$. For time-invariant systems, $H(t)$ is called the **impulse response** (because $y(t) = H(t)$ when $r(\tau) = \delta(\tau)$) and its Laplace transform $\widetilde{H}(s)$ is called the **transfer function**.

A system is **causal** if $\Lambda = t$ (i.e., the output $y(t)$ depends only on the input $r(\tau)$ for $\tau < t$.).

A system is **stable** if a bounded input $r(\cdot)$ implies a bounded output $y(\cdot)$. (BIBO=Bounded Input, Bounded Output).

Consider the analog time-invariant modern model. Suppose its transfer function is[3]

$$\widetilde{H}_{y|u}(s) = K\frac{\mathcal{N}(s)}{\mathcal{D}(s)} \tag{10.102}$$

where $K \in \mathbb{C}$ is some constant independent of s,

$$\mathcal{N}(s) = \prod_{a=1}^{A}(s-\alpha_a)^{m_a} \tag{10.103}$$

and

$$\mathcal{D}(s) = \prod_{b=1}^{B}(s-\beta_b)^{n_b} \tag{10.104}$$

and where the polynomials $\mathcal{N}(s)$ and $\mathcal{D}(s)$ have no common factors. The $\{\alpha_a\}_{a=1}^{A}$ are called the **zeros** of the transfer function, and the $\{\beta_b\}_{b=1}^{B}$ are called the **poles** of the transfer function. This system is

- **stable** if the zeros of the transfer function fall on the left half of the s-plane (i.e., if $Re(\beta_b) < 0$ for all b)
- **marginally stable** if $Re(\beta_b) = 0$ for some b and $Re(\beta_b) < 0$ for the others. Marginally stable systems sustain undamped oscillations. They may become unstable if perturbed.
- **unstable** if $Re(\beta_b) > 0$ for some b.

To get some intuition as to why this is so, let's look at the inverse Laplace transform of $1/(s-\beta_b)$:

$$\lim_{t\to\infty}\int_{\gamma-iT}^{\gamma+iT} ds\ \frac{e^{st}}{s-\beta_b} = Ke^{\beta_b t}u_0(t) \tag{10.105}$$

[3]We are assuming that the transfer function is a scalar 1×1 matrix. If it has row or column dimensions larger than one, one analyzes each entry of the transfer function matrix as if it were a scalar transfer function.

where $K \in \mathbb{C}$, $\gamma, T \in \mathbb{R}$ and $Re(\beta_b) < \gamma$. At the pole, $s = \beta_b$, so

$$e^{st} = e^{\beta_b t} = e^{Re(\beta_b)t} e^{iIm(\beta_b)t} \tag{10.106}$$

As $t \to \infty$, the integral blows up if $Re(\beta_b) > 0$ and converges if $Re(\beta_b) < 0$. As $t \to \infty$, integration under poles in the left half plane doesn't blow up because it is ultimately dampened by a decaying exponential.

An analogous result holds for the digital time-invariant modern model, except that in that case, the transfer function $\widetilde{H}_{y|u}(z)$ is a Z-transform (instead of a Laplace transform), and stability occurs if the poles of the transfer function fall inside the unit circle of the z-plane (instead of the left half plane of the s-plane.)

10.8 Controllability, Observability

Suppose $x, u, y : \mathcal{T} \to \mathcal{X}$. where $\mathcal{T} = [0, \infty)$.

- $a \in \mathcal{X}$ is **controllable at time** $t_0 \in \mathcal{T}$ if there exist a time $t_1 \in \mathcal{T}$ and an input $u(\cdot)$ such that $x(t_0) = a$ and $x(t_1) = 0$.
- $x(\cdot)$ is **controllable at time** $t_0 \in \mathcal{T}$ if, for all $a \in \mathcal{X}$, a is controllable at time $t_0 \in \mathcal{T}$.
- $a \in \mathcal{X}$ is **observable at time** $t_0 \in \mathcal{T}$ if there exists a $t_1 \in \mathcal{T}$ such that $x(t_0) = a$ and a can be determined from the values of $y(t)$ for $t \in [t_0, t_1]$.
- $x(\cdot)$ is **observable at time** $t_0 \in \mathcal{T}$ if, for all $a \in \mathcal{X}$, a is observable at time $t_0 \in \mathcal{T}$.

10.9 Signal Flow Graph

This section on Signal Flow (SF) graphs is based on Ref.[180]. According Ref.[180], SF graphs were invented by Shannon in 1942 to model "differential equation machines". They were later extended and promoted by Mason circa 1955.

SF graphs are very similar to LDEN (Linear Deterministic with External Noise) bnets discussed in Chapter 52. However, there are some important differences between the two, such as: (1) SF graphs have no external random nodes (2) unlike the LDEN considered in Chapter 52, SF graphs can have feedback cycles.

In SF graphs, the multiplicative factors carried by the arrows are called "gains". In SF graphs, the gains are always Laplace transforms (for the analog case) or Z-transforms (for the digital case). In this section, we will only discuss SF graphs for the analog case, but keep in mind that the digital case is very similar.

Next we will discuss the analog classical and analog (time-invariant) modern models in terms of SF graphs.

Figure 10.4: This figure is the Laplace transform of classical model Figs.10.2 (a) and (c).

- classical model

  Fig.10.4 is the Laplace transform of Figs.10.2 (a) and (c). Fig.10.4 (c) is an SF graph.

  The SF graph of Fig.10.4 implies the following system of equations:

$$\begin{cases} \underline{\widetilde{e}} = \underline{\widetilde{r}} - \underline{\widetilde{f}} \\ \underline{\widetilde{u}} = \widetilde{C}\underline{\widetilde{e}} \\ \underline{\widetilde{y}} = \widetilde{\Pi}\underline{\widetilde{u}} \\ \underline{\widetilde{f}} = \widetilde{F}\underline{\widetilde{y}} \end{cases} \tag{10.107}$$

- time-invariant modern model

  Fig.10.5 is the Laplace transform of Figs.10.3 (a) and (c) for the time-invariant case. Fig.10.5 (c) is an SF graph.

  The SF graph of Fig.10.5 implies the following system of equations:

$$\begin{cases} \widetilde{x} = \frac{1}{s}(A\widetilde{x} + B\widetilde{u}) \\ \widetilde{y} = C\widetilde{x} + D\widetilde{u} \end{cases} \tag{10.108}$$

Next, we shall discuss some properties of the feedback cycles of SF graphs.

An approach that I like is to re-express an SF graph with feedback cycles by one without them that is easier to understand.

Figure 10.5: This figure is the Laplace transform of modern model Figs.10.3 (a) and (c), for the time-invariant case. Figure (c') is a more detailed version of figure (c).

Claim 34 *Eliminating bubbles (i.e., self-feedback cycles)*

(10.109)

proof: From the left hand diagram,

$$\underline{x} = \mu\underline{x} + \alpha\underline{a} + \beta\underline{b} \tag{10.110}$$

Hence,

$$\underline{x} = \frac{\alpha}{1-\mu}\underline{a} + \frac{\beta}{1-\mu}\underline{b} \tag{10.111}$$

QED

Claim 35 *Eliminating 2-node cycles*

(10.112)

proof: From the left hand diagram,

$$\begin{cases} \nu \underline{y} + +\alpha \underline{a} = \underline{x} \\ \beta \underline{b} + \mu \underline{x} = \underline{y} \end{cases} \tag{10.113}$$

Hence,

$$\begin{bmatrix} 1 & -\nu \\ -\mu & 1 \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} = \begin{bmatrix} \alpha \underline{a} \\ \beta \underline{b} \end{bmatrix} \tag{10.114}$$

But

$$\begin{bmatrix} 1 & -\nu \\ -\mu & 1 \end{bmatrix}^{-1} = \frac{1}{1-\mu\nu} \begin{bmatrix} 1 & \nu \\ \mu & 1 \end{bmatrix} \tag{10.115}$$

so

$$\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} = \frac{1}{1-\mu\nu} \begin{bmatrix} 1 & \nu \\ \mu & 1 \end{bmatrix} \begin{bmatrix} \alpha \underline{a} \\ \beta \underline{b} \end{bmatrix} \tag{10.116}$$

QED

We could continue by showing how to eliminate cycles with $3, 4 \ldots$ cycle nodes, but the pattern is clear. If there are N cycle nodes $\underline{l}_1, \underline{l}_2, \cdots, \underline{l}_N$, then any arrow $\underline{a} \to \underline{l}_{k_0}$ in the feedback graph is replaced by N arrows $\underline{a} \to \underline{l}_k$ for $k = 1, 2, \ldots, N$ in the non-feedback graph. Let G_{cycle} be the product of the gains in the cycle. If α_{k_0} is the gain of arrow $\underline{a} \to \underline{l}_{k_0}$ in the feedback graph, then the gain of $\underline{a} \to \underline{l}_k$ in the non-feedback graph equals the product of the gains in the path $\underline{a} \xrightarrow[\alpha_{k_0}]{} \underline{l}_{k_0} \to \cdots \to \underline{l}_k$ divided by $1 - G_{cycle}$.

SF graphs with feedback cycles can be used to represent a general system of N linear equation with N unknowns (i.e. $y = Cx$, where C is an $N \times N$ matrix). Fig.10.6 shows an SF graph that does this for $N = 3$.

In Fig.10.6,

$$\begin{aligned} \underline{x}_1 &= (c_{11}+1)\underline{x}_1 + c_{12}\underline{x}_2 + c_{13}\underline{x}_3 - \underline{y}_1 \\ \underline{x}_2 &= c_{21}\underline{x}_1 + (c_{22}+1)x_2 + c_{23}\underline{x}_3 - \underline{y}_2 \\ \underline{x}_3 &= c_{31}\underline{x}_1 + c_{32}\underline{x}_2 + (c_{33}+1)\underline{x}_3 - \underline{y}_3 \end{aligned} \tag{10.117}$$

so

$$\begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \\ \underline{y}_3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \end{bmatrix} \tag{10.118}$$

$$y = Cx \tag{10.119}$$

Note that the self-feedback cycles in Fig.10.6 can be eliminated using the technique described in Claim 34.

Figure 10.6: SF graph that represents a general system of 3 linear equations with 3 unknowns.

Chapter 11

Copula

This chapter is based on Refs.[125] and [10].

A copula in architecture is a domed roof. Here we will discuss a copula in Statistics. Copulas are probably called this in Statistics because their probability density resembles a dome when their domain is the real plane $\mathbb{R}^2$. Furthermore, the word "copula" means "connector" or "coupler" in Latin, and both, the copula in Architecture and the one in Statistics, connect the sides (or marginals in the Statistics case) of a geometrical shape.

Let $x = [x_i]_{i=1}^n \in \mathbb{R}$ be an n dimensional column vector. Given a probability distribution (actually, a density) $P(x)$, we will refer to $P(\underline{x}_i = x_i)$ for all i as its **marginals** and to $P(\underline{x}_i \leq x_i)$ for all i as its **cumulative marginals** or **c-marginals** for short. This is normally referred to as the **CDF (cumulative distribution function)** of $\underline{x}_i$.

Suppose you know the marginals $P(x_i)$ of $P(x)$, but you don't know $P(x)$ itself. There are infinitely many possible $P(x)$'s with those marginals. Informally speaking, a **copula** is one of those $P(x)$, a smooth one. The dimension n of the domain of the copula, is referred to as the **copula dimension**. See Figs. 11.1 and 11.2 for examples of 2-dimensional copulas.

Let $x = [x_i]_{i=1}^n \in \mathbb{R}^n$ and $u = [u_i]_{i=1}^n \in \mathbb{R}^n$ be n dimensional column vectors. Henceforth, will denote the c-marginals of $\underline{x}_i$ by $\Phi_{\underline{x}_i}$

$$\Phi_{\underline{x}_i}(x_i) = P(\underline{x}_i \leq x_i) \tag{11.1}$$

and the generalization of this map to vector arguments by $\Phi_{\underline{x}}$:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}, \quad \Phi_{\underline{x}}(x) = \begin{bmatrix} \Phi_{\underline{x}_1}(x_1) \\ \Phi_{\underline{x}_2}(x_2) \\ \cdots \\ \Phi_{\underline{x}_n}(x_n) \end{bmatrix} \tag{11.2}$$

More precisely, a copula in Statistics is defined as follows. A **copula density** is a probability density $P(u)$ such that (see Fig.11.3)

Figure 11.1: Contour plot of a 2-dimensional copula with uniform marginals

Figure 11.2: Contour plot of a 2-dimensional copula with skewed bell-shaped marginals.

$$
\begin{tikzcd}
u_1 & x_1 \arrow[l, "\Phi_{\underline{x}_1}"'] \arrow[d] & P(x_1) \\
u_2 & x_2 \arrow[l, "\Phi_{\underline{x}_2}"'] & \\
& = P(u,x) = & \\
P(u_1) & u_1 \arrow[r, "\Phi^{-1}_{\underline{x}_1}"'] \arrow[d] & x_1 \\
& u_2 \arrow[r, "\Phi^{-1}_{\underline{x}_2}"'] & x_2
\end{tikzcd}
$$

Figure 11.3: Graphical representation of Eq.(11.4) for $n = 2$. $\Phi_{\underline{x}_i}(x_i) = P(\underline{x}_i \leq x_i)$ is the CDF of $\underline{x}_i$, and we define $u_i = \Phi_{\underline{x}_i}(x_i)$

$$
\underbrace{P(u|x)}_{\prod_{i=1}^{n} \delta\left(u_i - \Phi_{\underline{x}_i}(x_i)\right)} \quad P(x) = P(u,x) = \underbrace{P(x|u)}_{\prod_{i=1}^{n} \delta\left(x_i - \Phi^{-1}_{\underline{x}_i}(u_i)\right)} \quad P(u) \tag{11.3}
$$

$$
\boxed{P(\underline{x} = x) = \Big[\underbrace{P(\underline{u} = u)}_{\text{copula density}}\Big]_{u = \underbrace{\Phi_{\underline{x}}(x)}_{\text{c-marginals}}}} \tag{11.4}
$$

A **copula** $C(u)$ is defined as the CDF of its copula density.

$$
\begin{aligned}
C(u) &= \prod_{i=1}^{n} \left\{ \int_{-\infty}^{u_i} du_i' \right\} P(\underline{u} = u') && (11.5)\\
&= \underbrace{P(\forall i : \underline{u}_i \leq u_i)}_{\stackrel{\text{def}}{=} P(\underline{u} \leq u)} && (11.6)
\end{aligned}
$$

$$
\partial_{u_1} \partial_{u_2} \ldots \partial_{u_n} C(u) = P(u) \tag{11.7}
$$

There are copulas that are well-defined by Eq.(11.6), but not differentiable, so, technically, without smoothing, their copula density does not exist. For this reason, if possible, it is always best and most general to state copula results in terms of CDFs, instead of probability densities. For example, the boxed Eq.(11.4) stated in terms of CDFs, is

$$
\boxed{P(\underline{x} \leq x) = \left[\underbrace{P(\underline{u} \leq u)}_{\text{copula}} \right]_{u = \underbrace{\Phi_{\underline{x}}(x)}_{\text{c-marginals}}}} \tag{11.8}
$$

Claim 36 $\underline{u}_i = \Phi_{\underline{x}_i}(\underline{x}_i)$ *implies that the marginal* $P(u_i) = 1$ *is a uniform distribution on* $[0, 1]$.

proof:

$$
\underline{u}_i = \Phi_{\underline{x}_i}(\underline{x}_i) = P(\underline{x}_i \leq \underline{x}_i) = 1 \tag{11.9}
$$

If that doesn't convince you, here is another proof.

$$
\begin{aligned}
P(\underline{u}_i \leq u_i) &= P(\Phi_{\underline{x}_i}(\underline{x}_i) \leq u_i) && (11.10)\\
&= P(\underline{x}_i \leq \Phi_{\underline{x}_i}^{-1}(u_i)) && (11.11)\\
&= \Phi_{\underline{x}_i}\left(\Phi_{\underline{x}_i}^{-1}(u_i)\right) && (11.12)\\
&= u_i && (11.13)
\end{aligned}
$$

Hence,

$$
\begin{aligned}
P(\underline{u}_i = u_i) &= \frac{\partial}{\partial u_i} \int_{-\infty}^{u_i} du_i' \, P(\underline{u}_i = u_i') && (11.14)\\
&= \frac{\partial P(\underline{u}_i \leq u_i)}{\partial u_i} && (11.15)\\
&= 1 && (11.16)
\end{aligned}
$$

QED

11.1 Examples

In the following examples, $n = 2$, and we sometimes substitute $(u_1, u_2) = (u, v)$, $(x_1, x_2) = (x, y)$.

1. **$\underline{x}_1$ and $\underline{x}_2$ are independent**

   If $\underline{x}_1$ and $\underline{x}_2$ are independent, $\underline{u}_1$ and $\underline{u}_2$ are too. Hence,

$$\begin{aligned} C(u_1, u_2) &= P(\underline{u}_1 \leq u_1, \underline{u}_2 \leq u_2) && (11.17) \\ &= P(\underline{u}_1 \leq u_1)P(\underline{u}_2 \leq u_2) && (11.18) \\ &= u_1 u_2 && (11.19) \end{aligned}$$

$$C(\Phi_{\underline{x}}(x)) = \Phi_{\underline{x}_1}(x_1)\Phi_{\underline{x}_2}(x_2) \tag{11.20}$$

$$\partial_{x_1}\partial_{x_2} C(\Phi_{\underline{x}}(x)) = \partial_{x_1}\partial_{x_2}\Phi_{\underline{x}_1}(x_1)\Phi_{\underline{x}_2}(x_2) \tag{11.21}$$

$$P(x) = P(x_1)P(x_2) \tag{11.22}$$

2. **$\underline{x}_2 = \alpha \underline{x}_2$ for some $\alpha > 0$**

   If $\underline{x}_2 = \alpha \underline{x}_1$ for some parameter $\alpha > 0$, then

$$\begin{aligned} u_1 &= \Phi_{\underline{x}_1}(x_1) && (11.23) \\ &= P(\underline{x}_1 \leq x_1) && (11.24) \\ &= P(\alpha \underline{x}_1 \leq \alpha x_1) && (11.25) \\ &= P(\underline{x}_2 \leq x_2) && (11.26) \\ &= \Phi_{\underline{x}_2}(x_2) && (11.27) \\ &= u_2 && (11.28) \end{aligned}$$

$$\begin{aligned} C(u_1, u_2) &= P(\underline{u}_1 \leq u_1, \underline{u}_2 \leq u_2) && (11.29) \\ &= P(\underline{u}_1 \leq u_1, \underline{u}_1 \leq u_2) && (11.30) \\ &= P(\underline{u}_1 \leq \min(u_1, u_2)) && (11.31) \\ &= \min(u_1, u_2) && (11.32) \end{aligned}$$

3. $\underline{x}_2 = -\alpha \underline{x}_2$ **for some** $\alpha > 0$

   If $\underline{x}_2 = -\alpha \underline{x}_1$ for some parameter $\alpha > 0$, then

$$\begin{aligned} u_1 &= \Phi_{\underline{x}_1}(x_1) && (11.33)\\ &= P(\underline{x}_1 \le x_1) && (11.34)\\ &= P(-\alpha \underline{x}_1 > -\alpha x_1) && (11.35)\\ &= P(\underline{x}_2 > x_2) && (11.36)\\ &= 1 - \Phi_{\underline{x}_2}(x_2) && (11.37)\\ &= 1 - u_2 && (11.38) \end{aligned}$$

$$\begin{aligned} C(u_1, u_2) &= P(\underline{u}_1 \le u_1, \underline{u}_2 \le u_2) && (11.39)\\ &= P(\underline{u}_1 \le u_1, 1 - \underline{u}_1 \le u_2) && (11.40)\\ &= P(1 - u_2 \le \underline{u}_1 \le u_1)) && (11.41)\\ &= \max(u_1 - (1 - u_2), 0) \quad \text{(Area of rectangle } [1 - u_2, u_1] \times [0, 1]) && (11.42) \end{aligned}$$

   The Fréchet–Hoeffding bounds are lower and upper bounds for any n-dim copula. For $n = 2$, these bounds are

$$\underbrace{\max(u_1 + u_2 - 1, 0)}_{\text{case } \underline{x}_2 = -\alpha \underline{x}_1} \le C(u_1, u_2) \le \underbrace{\min(u_1, u_2)}_{\text{case } \underline{x}_2 = \alpha \underline{x}_1} \tag{11.43}$$

4. **Gaussian copula**

   Recall that the n-dimensional multivariate Normal Distribution has a probability density

$$\mathcal{N}(x; \mu, \Sigma) = \frac{\exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right)}{\sqrt{(2\pi)^n \det(\Sigma)}} \tag{11.44}$$

   where $\mu = E[\underline{x}]$ and $\Sigma = \left\langle \underline{x}^T, \underline{x} \right\rangle$. For $n = 2$,

$$\begin{aligned} \Sigma &= \left\langle \underline{x}^T, \underline{x} \right\rangle && (11.45)\\ &= \begin{bmatrix} \sigma^2_{\underline{x}_1} & \rho \sigma_{\underline{x}_1} \sigma_{\underline{x}_2} \\ \rho \sigma_{\underline{x}_1} \sigma_{\underline{x}_2} & \sigma^2_{\underline{x}_2} \end{bmatrix} \quad \left(\sigma_{\underline{x}_i} = \sqrt{\langle \underline{x}_i, \underline{x}_i \rangle} \text{ and } \rho = \frac{\langle \underline{x}_1, \underline{x}_2 \rangle}{\sigma_{\underline{x}_1} \sigma_{\underline{x}_2}}\right) && (11.46)\\ &= \underbrace{\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}}_{\stackrel{\text{def}}{=} \Sigma_\rho} \quad (\text{Assume } \sigma_{\underline{x}_1} = \sigma_{\underline{x}_2} = 1) && (11.47) \end{aligned}$$

$$\Sigma_\rho^{-1} = \frac{1}{1-\rho^2}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix} \tag{11.48}$$

$$\mathcal{N}\left(x; \mu = 0, \Sigma = \Sigma_\rho\right) = \frac{\exp\left(-\frac{1}{2(1-\rho^2)} x^T \begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix} x\right)}{\sqrt{(2\pi)^2(1-\rho^2)}} \tag{11.49}$$

$$= \frac{\exp\left(-\frac{1}{2(1-\rho^2)}(x_1^2 + x_2^2 - 2\rho x_1 x_2)\right)}{\sqrt{(2\pi)^2(1-\rho^2)}} \tag{11.50}$$

For each i, assume the marginal of $\underline{x}_i$ is a Normal distribution with zero mean and unit variance:

$$P(x_i) = \mathcal{N}(x_i; \mu = 0, \sigma = 1) = \frac{\exp\left(-\frac{x_i^2}{2}\right)}{\sqrt{2\pi}} \tag{11.51}$$

Then

$$\Phi_{\underline{x}_i}(x_i) = P(\underline{x}_i \leq x_i) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \tag{11.52}$$

and

$$\Phi_{\underline{x}_i}^{-1}(u_i) = \sqrt{2}\operatorname{erf}^{-1}(1 - 2u_i) \tag{11.53}$$

Besides assuming Gaussian marginals, we will assume a Gaussian copula density

$$C(u) = \prod_{i=1}^{2}\left\{\int_{-\infty}^{u_i} du_i'\right\} P(u') \tag{11.54}$$

where

$$P(u) = \quad \mathcal{N}(x = \Phi_{\underline{x}}^{-1}(u); \mu = 0, \Sigma = \Sigma_\rho) \tag{11.55}$$

Hence,

$$P(x) = P(\underline{u} = \Phi_{\underline{x}}(x)) \tag{11.56}$$

$$= \mathcal{N}(x; \mu = 0, \Sigma = \Sigma_\rho) \tag{11.57}$$

Chapter 12

Counterfactual Reasoning

12.1 The 3 Rungs of Causal AI

According to Judea Pearl, there are 3 rungs in the ladder of causal AI. These are (as I see them):

1. **Observing Passively. Answering "What next?":** Collecting data and fitting curves to it, without any plan designed to investigate Nature's causal connections. Predicting the future.

2. **Doing causal experiments. Answering "Why?":** Doing experiments consciously designed to elucidate Nature's causal connections. Even cats do this!, but current AI doesn't.

3. **Imagining counterfactual situations, Analogizing. Answering "What if?":** Imagining gedanken experiments to further understand Nature's causal connections, and to decide what future courses of action are more likely to succeed, even if those courses of action are unprecedented, and have never been taken before. Making predictions about events that have never happened ("counterfactuals") is a very Bayesian concern, well out of the purview of frequentists. Nevertheless, humans do such "analogizing" all the time to great advantage. It becomes possible if there is some foreign but similar data that can be transported (transplanted, applied) to the situation of interest.

We will use the term **intervention operator (or simply "intervention")** to refer to an operator that maps a bnet to another bnet. In Chapter 22, we introduced an intervention operator called the **do operator** $\mathcal{D}_{\underline{x}=x}$ (this is our notation for what Pearl symbolizes by $do(\underline{x}) = x$). The study of counterfactuals requires that we introduce a new kind of intervention operator that we will call an **imagine operator**, and denote by $\mathcal{I}_{\underline{x}\rightarrow\underline{y}}$. These 2 types of intervention operators will be defined in subsequent sections of this chapter. Usage of the do operator characterizes rung 2, and usage of the imagine operator characterizes rung 3.

Chapter 60 on message passing is about rung 1. Chapter 22 on Do Calculus is about rung 2. This chapter is dedicated to rung 3.

Judea Pearl is fond of discussing rung 3 solely in terms of SCM.[1] In this chapter, we define rung 3 without using SCM, using solely bnets. This gives a more general version of rung 3, because SCM are a subset of bnets.

12.2 Do operator

Figure 12.1: Action of "do" operator $\mathcal{D}_{\underline{x}=5}$ on node $\underline{x}$.

The do operator $\mathcal{D}_{\underline{x}=5}$ is defined graphically in Fig.12.1. The TPM, printed in blue, for node $\tilde{\underline{x}}$ of Fig.12.1, is as follows.[2]

$$P(\tilde{x}; 5) = \delta(5, \tilde{x}) \tag{12.1}$$

The do operator $\mathcal{D}_{\underline{x}=5}$ amputates the incoming arrows of node $\underline{x}$ and sets the TPM of the new root node $\tilde{\underline{x}}$ to a delta function $\delta(\tilde{x}, 5)$ (or some state of $\underline{x}$ other than 5). Sometimes we call the new node $\mathcal{D}\underline{x}$ instead of $\tilde{\underline{x}}$.

The uses of the do operator are discussed in detail in Chapter 22.

12.3 Imagine operator

The imagine operator $\mathcal{I}_{\underline{x}\to\underline{y}}(5)$ is defined graphically in Fig.12.2. Note that Fig.12.2 actually defines two types of imagine operators, the one with an argument: $\mathcal{I}_{\underline{x}\to\underline{y}}(5)$, and the one without an argument: $\mathcal{I}_{\underline{x}\to\underline{y}}$. The TPMs, printed in blue, for various nodes in Fig.12.2, are as follows.

- For $\mathcal{I}_{\underline{x}\to\underline{y}}(\tilde{x})G$

$$P(y|\tilde{x}, a.) = P(y|\underline{x} = \tilde{x}, a.) \tag{12.2}$$

[1] SCM are what we call DEN. DEN (deterministic systems with external noise) are discussed in Chapter 52.

[2] The "; 5" in the distribution indicates that 5 is a frequentist parameter of the distribution.

Figure 12.2: Action of "imagine" operators $\mathcal{I}_{\underline{x}\to\underline{y}}(5)$ and $\mathcal{I}_{\underline{x}\to\underline{y}}$ on arrow $\underline{x} \to \underline{y}$. In this figure, $y^{nx} = [y(x)]_{\forall x \in val(\underline{x})}$, where $nx = |val(\underline{x})|$ and $val(\underline{x})$ is the set of states of node $\underline{x}$.

$$P(\tilde{x}; 5) = \delta(\tilde{x}, 5) \tag{12.3}$$

- For $\mathcal{I}_{\underline{x}\to\underline{y}}G$

$$P(y^{nx}|a.) = \prod_{\tilde{x}} P(\underline{y}(\tilde{x}) = y(\tilde{x})|a.) \tag{12.4}$$

$$P(y|y^{nx}, x) = \delta(y, y(x)) \tag{12.5}$$

The imagine operators $\mathcal{I}_{\underline{x}\to\underline{y}}(5)$ and $\mathcal{I}_{\underline{x}\to\underline{y}}$ operate on an arrow whereas the $\mathcal{D}$ operator operates on a node. $\mathcal{I}_{\underline{x}\to\underline{y}}(5)$ deletes arrow $\underline{x} \to \underline{y}$ and creates a new root node $\tilde{\underline{x}}$ and a new arrow $\tilde{\underline{x}} \to \underline{y}$. Sometimes we call the new node $\mathcal{I}_{\underline{y}}\underline{x}$ instead of $\tilde{\underline{x}}$. $\mathcal{I}_{\underline{x}\to\underline{y}}$ creates a new node $\underline{y}^{nx}$ and an arrow $\underline{y}^{nx} \to \underline{y}$.

Fig.12.3 shows how the imagine operator arises in Potential Outcomes (PO) theory. PO theory is discussed extensively in Chapter 77. As you can see, PO theory only uses a limited version of the 3 rungs of causal inference, because it doesn't

Figure 12.3: How imagine operator arises in Potential Outcomes (PO) theory.

Figure 12.4: $\mathcal{D}_{\underline{x}=5}\mathcal{I}_{\underline{x}\to\underline{y}}G$ gives a connection between do and imagine operators.

use the do operator, and it only uses one of 2 possible types of imagine operators. Furthermore, it assumes a very limited triangular DAG.

Fig.12.4 gives a connection between do and imagine operators. We see from that figure that for $\mathcal{D}_{\underline{x}=\tilde{x}}\mathcal{I}_{\underline{x}\to\underline{y}}G$, we have[3]

$$P(y|\mathcal{D}\underline{x} = \tilde{x}, a.) = P(\underline{y}(\tilde{x}) = y|a.) \tag{12.6}$$

One can define a **do-imagine-calculus** whose objective is to express probabilities such as $P(\underline{y}|\mathcal{D}\underline{r} = r, \mathcal{I}_{\underline{b}}\underline{s} = s, t)$ in terms of observed probabilities that do not contain any do or imagine operators in them. As with Do Calculus, this reduction is not always possible, and we say a probability is **$\mathcal{D}$-identifiable**, **$\mathcal{I}$-identifiable** or **$\mathcal{DI}$-identifiable** if it can be expressed without do, imagine or both operators.

In causal inference, we often consider "counterfactual" random variables $\underline{y}(0) \in \{0, 1\}$ and $\underline{y}(1) \in \{0, 1\}$. They are called counterfactual variables because one of the 2 variables refers to an event that has occurred, whereas the other variable refers to a "counterfactual event", i.e., an event that has never occurred. For some patients,

[3]In the notation favored by Pearl, Eq.(12.6) would be

$$P(y|do(X) = \tilde{x}, a.) = P(Y_{\tilde{x}} = y|a.)$$

$\underline{y}(0)$ has a value but $\underline{y}(1)$ doesn't. For other patients, the opposite is the case. There is some disagreement in the community as to which algorithms perform rung 3 operations, and which don't. This is the convention used in this book. We will say rung 3 operations are being performed if the counterfactual variables $\underline{y}(0)$ and $\underline{y}(1)$ are being used, or, equivalently, if a bnet is being used that includes nodes $\underline{y}(0)$, and $\underline{y}(1)$ that were produced by an imagine operator. Potential Outcomes (PO) theory (see Chapter 77) qualifies as rung 3 according to this convention. However, note that PO theory only does the bare minimum to reach rung 3. In PO theory, one usually evaluates $ATE = E[\underline{y}(1)] - E[\underline{y}(0)]$, which entails calculating $P(y(0))$ and $P(y(1))$. Pearl has extended the reach of rung 3 much further by calculating expected values that require knowledge of the joint distribution $P(y(0), y(1))$.

Chapter 13

Cross-Validation

This chapter is based on Ref.[127].

Cross-Validation (CV) is a method of calculating the **"out-of-training-set" (OOTS) error** for a classifier. What this means is that the classifier is trained on a training set, and its propensity to err is evaluated on a set different from the training set.

In k**-fold CV**, the most common CV method, and the only one we will discuss in this chapter, one partitions a dataset into k disjoint datasets of equal length. One uses $k-1$ of those sub-datasets to train a model, and saves the last sub-dataset to validate the model just trained. One actually rotates which of the k sub-datasets is used for validation purposes, and calculates k validation errors $\mathcal{E}_j$ for $j = 0, 1, \ldots, k-1$. Then one averages over the $\mathcal{E}_j$ to obtain a final OOTS error $\mathcal{E}$.

CV strongly resembles Jackknife Resampling (JR) (see Chapter 45), but in JR the validation sub-dataset is never used for anything, whereas in CV, it is used for validation purposes, to calculate an OOTS error.

Next, we will explain k-fold CV more explicitly, using equations and a bnet.

Let $L = [0, 1, 2, \ldots, nsam-1]$ be a list of individuals (samples) in a population. In this chapter, we will use the notation $A^\sigma = A[\sigma]$ and $\vec{A} = [A^\sigma : \sigma \in L]$ for a list (vector, 1-D array) indexed by L. We will refer to $DS = (\vec{x}, \vec{y})$ where $x^\sigma \in val(\underline{x})$, $y^\sigma \in val(\underline{y})$, as a dataset. If L_j is a list (possibly with duplicate items) such that $set(L_j) \subset set(L)$, then define $DS_j = (\vec{x}, \vec{y})_{L_j} = ((x^\sigma)_{\sigma \in L_j}, (y^\sigma)_{\sigma \in L_j})$. We will refer to DS_j as the **restriction of** $(\vec{x}, \vec{y})$ **to** L_j**.**

Let $J = \{0, 1, 2, \ldots, nj-1\}$.

Define a **training list(TL), validation list(VL) pair** (TL, VL) to be a pair of lists such that $set(TL)$ and $set(VL)$ are disjoint subsets of $set(L)$. Let (TL_j, VL_j) for $j \in J$ be nj such TL-VL pairs.

Fig.13.1 shows the TL-VL pairs that are used when doing k-fold CV. In that figure, $k = nj = 4$. As you can see, in k-fold CV, one chooses $nj = k$ list pairs (TL_j, VL_j) such that all individuals $\sigma \in L$ appear exactly once, in either TL_j or TV_j, but not in both.

We will refer to a function $Y : val(\underline{x}) \to val(\underline{c})$ as a classifier. It maps a vector

Figure 13.1: 4-fold CV with $|L| = 12$. For all j, $|VL_j| = 3$ and $|TL_j| = 9$. All individuals $\sigma \in L$ appear exactly once, in either TL_j or TV_j, but not in both.

of features x to a class c. Let Y_j for $j \in J$ denote nj classifiers.

If $Y_j : val(\underline{x}) \to val(\underline{c})$ and $val(\underline{c})$ is a discrete set ("categorical"), then define the **OOTS error for the jth classifier** as:

$$\mathcal{E}_j = \frac{1}{|VL_j|} \sum_{\sigma \in VL_j} \mathbb{1}(y^\sigma \neq Y_j(x^\sigma)) \ . \tag{13.1a}$$

On the other hand, if $val(\underline{c}) = \mathbb{R}$, it makes more sense to define $\mathcal{E}_j$ as a mean square error:

$$\mathcal{E}_j = \frac{1}{|VL_j|} \sum_{\sigma \in VL_j} (y^\sigma - Y_j(x^\sigma))^2 \ . \tag{13.1b}$$

Finally, define the **final OOTS** error as

$$\mathcal{E} = \frac{1}{|J|} \sum_{j \in J} \mathcal{E}_j \ . \tag{13.2}$$

Fig.13.2 gives a bnet that represents the CV algorithm. The TPMs, printed in blue, for the bnet Fig.13.2, are as follows:

$$P((\vec{x}, \vec{y})_{TL_j} | (\vec{x}, \vec{y})) = \mathbb{1}(\quad (\vec{x}, \vec{y})_{TL_j} = \text{restriction of } (\vec{x}, \vec{y}) \text{ to } TL_j. \quad) \tag{13.3}$$

$$P((\vec{x}, \vec{y})_{VL_j} | (\vec{x}, \vec{y})) = \mathbb{1}(\quad (\vec{x}, \vec{y})_{VL_j} = \text{restriction of } (\vec{x}, \vec{y}) \text{ to } VL_j. \quad) \tag{13.4}$$

$$P(Y_j | (\vec{x}, \vec{y})_{TL_j}) = \mathbb{1}(\quad Y_j = \text{classifier trained with } (\vec{x}, \vec{y})_{TL_j} \quad) \tag{13.5}$$

Figure 13.2: Bnet for 3-fold CV.

$$P(\mathcal{E}_j | Y_j, (\vec{x}, \vec{y})_{VL_j}) = \mathbb{1}(\quad \mathcal{E}_j = \text{ defined by Eqs.(13.1).} \quad) \tag{13.6}$$

$$P(\mathcal{E} | (\mathcal{E}_j)_{j \in J}) = \mathbb{1}(\quad \mathcal{E} = \text{ defined by Eq.(13.2).} \quad) \tag{13.7}$$

Chapter 14

DAG Extraction From Text (DEFT) or Time-Series

To see how to extract a causal DAG from chronologically ordered text, see the github repo for Mappa-Mundi Ref.[92].

To see how to extract a causal DAG from a FitBit time-series table (TST), see the github repo for CausalFitbit Ref.[89].

Chapter 15

Dataset Shift and Batch Normalization

In this chapter, we will represent Linear Regression (LR) as follows. We list a dataset; i.e., a set of tuples indexed by the individuals σ of a population Σ such that $|\Sigma| = nsam$. The independent variables of the LR (i.e., x^σ) are unboxed and the dependent variable (a.k.a. target feature) (i.e., y^σ) is shown inside a box. Then we show an arrow with the superscript "LR-fit", followed by the fit function obtained by performing the LR.

$$\{(\sigma, x^\sigma = [x_i^\sigma], \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{LR-fit}} \widehat{y}(x) = \alpha + \sum_i x_i \beta_i \tag{15.1}$$

Analogously, we represent Supervised Machine Learning (ML) as follows.

$$\{(\sigma, x^\sigma, \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{y}(x) \tag{15.2}$$

When doing ML, we partition the full population Σ_{full} into two disjoint sets, the **training set** $\Sigma_{train} = \Sigma(\underline{s} = 0) = \Sigma$ and the **testing set** $\Sigma_{test} = \Sigma(\underline{s} = 1) = \Sigma^*$. Then we do two ML fits:

$$\begin{array}{ll} \text{training:} & \{(\sigma, x^\sigma, \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{y}(x) \\ \text{testing:} & \{(\sigma, x^\sigma, \boxed{y^\sigma}) : \sigma \in \Sigma^*\} \xrightarrow{\text{ML-fit}} \widehat{y}^*(x) \end{array} \tag{15.3}$$

Ideally, $\widehat{y}(x)$ and $\widehat{y}^*(x)$, will be almost equal for all x. Dataset shift occurs when this is not the case. Equivalently, let

$$\begin{array}{lll} P_{train}(x, y) & = P(x, y|\underline{s} = 0) & = P(x, y) \\ P_{test}(x, y) & = P(x, y|\underline{s} = 1) & = P^*(x, y) \end{array} \tag{15.4}$$

We say there is a **dataset shift** if

$$P(x, y) \neq P^*(x, y) \tag{15.5}$$

Figure 15.1: For Linear Regression, 2 types of dataset shift: covariate shift and concept shift.

15.1 Covariate Shift

We say there is a **covariate shift** if (see Fig.15.1)

$$P(y|x) = P^*(y|x) \text{ but } P(x) \neq P^*(x) \tag{15.6}$$

This can be represented in terms of bnets as follows[1]

$$\begin{tikzcd}[column sep=small] \underline{s} = 0 \arrow[d] & & \underline{s} = 1 \arrow[d] & \\ \underbrace{\underline{x} \arrow[r] & \underline{y}}_{\underline{s} = 0} & \neq & \underline{x} \arrow[r] & \underline{y} \\ = & \underline{x} \arrow[r] \arrow[u] & \underline{y} & & \end{tikzcd} \tag{15.7}$$

15.2 Concept Shift

We say there is a **concept shift** if (see Fig.15.1)

$$P(y|x) \neq P^*(y|x) \text{ but } P(x) = P^*(x) \tag{15.8}$$

This can be represented in terms of bnets as follows[2]

[1]See Chapter 89.
[2]See Chapter 89.

$$
\begin{tikzcd}
\underline{s} = 0 \arrow[dr] & & \underline{s} = 1 \arrow[dr] & \\
\underbrace{\underline{x} \arrow[r] & \underline{y}}_{\substack{\underline{s} = 0 \\ = \ \underline{x} \arrow[r] \arrow[u] & \underline{y} \arrow[ul]}} & \neq & \underline{x} \arrow[r] & \underline{y}
\end{tikzcd}
\tag{15.9}
$$

15.3 Batch Normalization

Batch Normalization (BN) is a technique that is used to diminish dataset shift in Neural Nets.

Let $h_i^{\lambda,\sigma}$ be the output, for individual $\sigma \in \Sigma$, of the ith node of layer λ of a Neural Net (NN). Using the notation of Chapter 68,

$$h_i^{\lambda,\sigma} = \mathcal{A}_i^{\lambda}(z_i^{\lambda,\sigma}) \tag{15.10}$$

where

$$z_i^{\lambda,\sigma} = \sum_j w_{i|j}^{\lambda} h_j^{\lambda-1,\sigma} + b_i^{\lambda} \tag{15.11}$$

Activation functions $\mathcal{A}_i^{\lambda} : \mathbb{R} \to \mathbb{R}$ for NNs are discussed in Section 68.1. Suppose the population Σ is partitioned into disjoint batches $\Sigma^{(b)}$ for $b = 1, 2, \ldots, B$. Let the set of points $\{z_i^{\lambda,\sigma} : \sigma \in \Sigma^{(b)}\}$ have mean $\mu_i^{\lambda(b)}$ and standard deviation $\sigma_i^{\lambda(b)}$. For any $z_i^{\lambda,\sigma}$ with $\sigma \in \Sigma$, define the BN activation function $\mathcal{A}_{i,BN}^{\lambda}(\cdot)$ by

$$\mathcal{A}_{i,BN}^{\lambda}(z_i^{\lambda,\sigma}) = \gamma_i^{\lambda} \left[\frac{z_i^{\lambda,\sigma} - \mu_i^{\lambda(b)}}{\sigma_i^{\lambda(b)}} \right] + \beta_i^{\lambda} \quad \text{if } \sigma \in \Sigma^{(b)} \ , \tag{15.12}$$

where the real valued parameters γ_i^{λ} and β_i^{λ} are learned during the optimization process. If node $\underline{h}_i^{\lambda}$ of the NN has activation function $\mathcal{A}_i^{\lambda} : \mathbb{R} \to \mathbb{R}$, defined a new activation function $\mathcal{A}_{i,new}^{\lambda} : \mathbb{R} \to \mathbb{R}$ by the composition of functions

$$\mathcal{A}_{i,new}^{\lambda} = \mathcal{A}_i^{\lambda} \circ \mathcal{A}_{i,BN}^{\lambda} \tag{15.13}$$

Hence, the BN activation function is applied after the linear transformation Eq.(15.11), but before the nonlinear transformation $\mathcal{A}_i^{\lambda}$.

Intuition on why BN diminishes dataset shift: We discussed in Section 68.1 how nonlinear activation functions have a range that is smaller than their domain. Presumably, BN helps to make the range of the activation functions even more concentrated.

Chapter 16

Decision Trees

This chapter is based mainly on Ref.[80].

Figure 16.1: Example of a decision tree.

Fig.16.1 shows a typical **decision tree (dtree)**. As you can see, a dtree contains two types of nodes: the non-leaf nodes, and the leaf nodes. The **non-leaf nodes** pose **questions**. In general, the **answers**[1] to those questions can be multiple choices with two or more choices. For each of those choices, a tree branch labeled by the choice comes down from the question node. The **leaf nodes** represent

[1]The **question-answer pairs** in dtrees are also called **attribute-value pairs**. Attributes are also called **features**.

endpoints, goals, final conclusions, payoffs, etc. Dtrees can be viewed as classifiers. They take in a large amount of information about a population and compress that information to just a few classes. If $val(\underline{c})$ is the set of distinct leaf node labels, then we call each $c \in val(\underline{c})$ a **class of the classifier**. In the case of Fig.16.1, $val(\underline{c}) = \{\text{No Loan, Loan}\}$.

16.1 Conversion to Bnet

Figure 16.2: LD (Linear Deterministic) bnet corresponding to Decision Tree in Fig.16.1. The p_i are probabilities.

Every dtree can be converted in a natural and trivial way, into a LD (Linear Deterministic) bnet.[2] Fig.16.2 is a LD bnet corresponding to the dtree shown in Fig.16.1. The structural equations of the bnet in Fig.16.2 are

$$\underline{x}_j = p_j \; pa(\underline{x}_j) \tag{16.1}$$

for all non-leaf nodes $\underline{x}_j$ ($\underline{x}_0 = 1$ for the root node), and

$$\underline{y}_j = p_j \; pa(\underline{y}_j) \tag{16.2}$$

for all leaf nodes $\underline{y}_j$.

Note the following important points about Fig.16.2

- The leaf and non-leaf nodes are different in nature. The non-leaf nodes ask questions which their branches answer, whereas the leaf nodes don't ask anything; leaf nodes only carry answers to the question of their parent node. Thus

[2]LD, DEN and LDEN bnets are defined in Chapter 52.

we represent non-leaf nodes and leaf nodes differently. Non-leaf nodes are enclosed by an oval and have an $\underline{x}_\alpha$ label. Leaf nodes are enclosed by a rectangle, and have an $\underline{y}_\alpha$ label.

- The arrows are labeled by **branch probabilities** p_i. The probabilities p_i exiting each node sum to 1. If $\underline{x}_\alpha$ is a non-leaf node,

$$P(\underline{x}_\alpha = x) = p_j \tag{16.3}$$

where j labels one of the children of $\underline{x}_\alpha$. Hence, j is a function of x, $j = f(x)$, and

$$\sum_{x \in val(\underline{x}_\alpha)} p_{f(x)} = 1 \tag{16.4}$$

16.2 Structure Learning for Dtrees

Let

$J_0 = \{0, 1, \ldots, nj - 1\}$
$\Sigma = \{0, 1, 2, \ldots, nsam - 1\}$
$DS = \{(\sigma, x^\sigma, c^\sigma) : \sigma \in \Sigma\}$ be a dataset
$\sigma \in \Sigma$ be an individual (a sample) from a population,
$x^\sigma \in val(\underline{x})$ be the **feature (attributes, questions) vector**. $val(\underline{x}) = val(\underline{x}_0) \times val(\underline{x}_1) \times \ldots \times val(\underline{x}_{nj-1})$, $x = (x_0, x_1, \ldots, x_{nj-1}) \in val(\underline{x})$, $x_j \in val(\underline{x}_j)$
$c^\sigma \in val(\underline{c})$ be a **classification class**
We will assume $val(\underline{x})$ and $val(\underline{c})$ are finite sets.

Building a **classifier** Y (curve fit) for a dtree means finding a deterministic function $Y : val(\underline{x}) \to val(\underline{c})$ such that $c^\sigma \approx Y(x^\sigma)$ for all $\sigma \in \Sigma$. If we divide the population Σ into two large disjoint sets, a **training set** Σ_{train} and a **validation set** Σ_{vali}, and if $c^\sigma \approx Y(x^\sigma)$ very closely for $\sigma \in \Sigma_{train}$ but fits poorly for $\sigma \in \Sigma_{vali}$, then we say the classifier Y suffers from **overfitting**. We can learn the structure and braching probabilities of a dtree from a dataset DS, by using the dtree **Structure Learning (SL)** algorithm that we will discuss in detail below. However, that algorithm is prone to produce a classifier Y that overfits. Two techniques commonly used to reduce the effects of overfitting are **pruning** and **Random Forest (RF)** (see Chapter 80). Pruning just means somehow removing nodes that are too specific. An RF is an ensemble of dtrees that one averages over. In this chapter, we will only deal with a single dtree, not an ensemble of them.

Dtree SL was invented in 1984-1986 so it is fairly old. Many in the AI community consider dtrees old fashioned compared to neural nets. But dtrees are **interpretable** whereas neural nets aren't.[3] Bnets are interpretable too.

[3]To be precise, only plain dtrees without boosting or bagging are interpretable. Dtrees used within

Below, we give the standard algorithm for SL of a dtree, in the form of pseudo-code. But first, we define two quantities, Information Gain and Gini, that are used in that pseudo-code.

16.2.1 Information Gain, Gini

This section uses various Shannon Information Theory entropies. Our notation for those entropies is described in Chapter C.

Call a **separation ability measure** (SAM) a measure used to decide, when constructing a dtree from a dataset, in what order to ask the questions about the feature vector x. The question order is decided by searching over all so far unused questions for the question with the largest SAM.[4]

Let $N_j(c, x_j)$ be the number of individuals σ in the population that is exiting question node j, belonging to class c and having $\underline{x}_j = x_j$. From $N_j(c, x_j)$, we can define an empirical probability distribution

$$P_j(c, x_j) = \frac{N_j(c, x_j)}{N_j} \ , \quad \text{where } N_j = \sum_{c, x_j} N_j(c, x_j) \tag{16.5}$$

Once we have an empirical probability distribution $P_j(c, x_j)$ for each node $\underline{x}_j$ and class c, we can define

$$\begin{aligned} IG_j &= H_j(\underline{c}) - H_j(\underline{c}|\underline{x}_j) && (16.6) \\ &= H_j(\underline{c} : \underline{x}_j) && (16.7) \end{aligned}$$

IG_j is called the **information gain for node** $\underline{x}_j$. Maximizing this mutual information produces a node $\underline{x}_j$ that has a large correlation to a class c. If the goal is to reach a point where each leaf node is closely correlated to a different class, then maximizing the Information Gain of each new node is a greedy move towards that goal. Thus, Information Gain is a good SAM for dtree SL.

If we approximate

$$\begin{aligned} \ln P_j(c|x_j) &\approx \ln[1 + P_j(c|x_j) - 1] && (16.8) \\ &\approx P_j(c|x_j) - 1 && (16.9) \end{aligned}$$

in $H_j(\underline{c}|x_j)$, we get what is called the **Gini (or Gini Index) for node** $\underline{x}_j$:

boosting (see Chapter 1 on AdaBoost and Chapter 111 on XGBoost) or bagging (see Chapter 80 on Random Forest) gain much accuracy but lose **interpretability**.

[4]SAM is also called, somewhat confusingly, the splitting criterion and Gain.

$$H_j(\underline{c}|x_j) = -\sum_c P_j(c|x_j) \ln P_j(c|x_j) \tag{16.10}$$

$$\approx 1 - \sum_{c \in val(\underline{c})} P_j(c|x_j)^2 \stackrel{\text{def}}{=} Gini_{x_j} \tag{16.11}$$

$Gini_{x_j}$ is a reasonable polynomial approximation to $H_j(\underline{c}|x_j)$.[5] It is computationally much less expensive than $H_j(\underline{c}|x_j)$, because it does not require computing a log.

We say a probability distribution $P_{\underline{x}}$, is **pure (i.e., deterministic)** if $P_{\underline{x}}(x) = \delta(x, x_0)$. $Gini_{x_j}$ and $H_j(\underline{c}|x_j)$ are both always non-negative. They both vanish iff $P_j(c|x_j)$ is pure. Thus, $Gini_{x_j}$ and $H_j(\underline{c}|x_j)$ are both good measures of **class impurity**.

The **average Gini of node** $\underline{x}_j$ is defined as

$$AGini_j = \sum_{x_j \in val(\underline{x}_j)} P_j(x_j) Gini_{x_j} \ . \tag{16.12}$$

$AGini_j$ measures the average impurity of the children of node $\underline{x}_j$.

In practice, the SL algorithm is done recursively. Each recursion step decides which feature x_j will be the root node of the current tree. For all "candidate" features (i.e., all $\underline{x}_j$ that haven't been used yet as tree nodes), one calculates IG_j, either exactly or approximately via Gini's, using the following formula:

$$IG_j = H_j(\underline{c}) - \sum_{x_j \in val(\underline{x}_j)} P_j(x_j) \underbrace{H_j(\underline{c}|x_j)}_{\approx Gini_{x_j}} \tag{16.13}$$

One then chooses $j = \operatorname{argmax}_j IG_j$. This maximizes the $\underline{c} - \underline{x}_j$ correlation.

Alternatively, some software programs use the average Gini $AGini_j$ as their SAM. They choose as root node $j = \operatorname{argmin}_j AGini_j$. This minimizes the average impurity of the children of node j. Since IG_j differs from $AGini_j$ by $H_j(\underline{c})$, maximizing IG_j and minimizing $AGini_j$ might lead to different results.

Example of calculation of $AGini_j$ and IG_j

Suppose we deduce from a dataset the numbers in the yellow cells in Table 16.1. These numbers are repeated in Fig.16.3. Then we can calculate the white cells in Table 16.1 as follows:

- Gini(hot)= $1 - (3/4) - (1/4) = 0.375$ (with logs=0.56)
- Gini(med)= $1 - (3/5)^2 - (2/5)^2 = 0.48$ (with logs=0.67)

[5]The average of $H(\underline{c} : b)$ over b is $H(\underline{c} : \underline{b}) = \sum_b P(b) H(\underline{c} : b)$. Likewise, the average of $H(\underline{c}|b)$ over b is $H(\underline{c}|\underline{b}) = \sum_b P(b) H(\underline{c}|b)$. $H(\underline{c} : \underline{b})$ becomes $H(\underline{c} : b)$ and $H(\underline{c}|\underline{b})$ becomes $H(\underline{c}|b)$ when $P(b)$ is a delta function.

$j = temp?$	$x_j = hot$	$x_j = med$	$x_j = cold$
$c = play$	3	3	3
$c = stay$	1	2	2
$Gini$	0.37	0.48	0.48
$AGini = 0.45$			
$H_j(\underline{c}\|x_j)$	0.56	0.67	.67
$H_j(\underline{c}\|\underline{x}_j) = 0.64$			
$H_j(\underline{c}) = 0.65$			
$IG_j(\underline{c}) = 0.01$			

Table 16.1: Evaluating AGini and IG for node *temp*.

Figure 16.3: Sub-tree corresponding to Table 16.1.

- Gini(cold)$= 1 - (3/5)^2 - (2/5)^2 = 0.48$ (with logs=0.67)
- AGini $= (4/14)(0.375)+(5/14)(0.48)+(5/14)(0.48) = 0.45$ (with logs=$H_j(\underline{c}|\underline{x}_j) = 0.64$)
- $H_j(\underline{c}) = -(9/14)\ln(9/14) - (5/14)\ln(5/14) = 0.65$
- $IG_j = H_j(\underline{c}) - H_j(\underline{c}|\underline{x}_j) = 0.01$

This gives the AGini and IG for just one candidate root node. We would have to calculate AGini (or IG) for all possible candidates and choose the candidate with the lowest AGini (or highest IG).

The **Information Gain Ratio** (IGR) is an alternative SAM.

$$IGR_j = \frac{IG_j}{H_j(\underline{x}_j)} \tag{16.14}$$

$$= \frac{H_j(\underline{c} : \underline{x}_j)}{H_j(\underline{x}_j)} \tag{16.15}$$

$$= \frac{H_j(\underline{x}_j) - H_j(\underline{x}_j|\underline{c})}{H_j(\underline{x}_j)} \tag{16.16}$$

$$= 1 - \frac{H_j(\underline{x}_j|\underline{c})}{H_j(\underline{x}_j)} \tag{16.17}$$

$0 \leq IGR_j \leq 1$
$IGR_j = 0$ iff $H_j(\underline{x}_j|\underline{c}) = H_j(\underline{x}_j)$.

16.2.2 Pseudo-code

Below, we give one of the first algorithms for SL of a dtree, in the form of pseudo-code. The strategy employed by the algo is to assume an incoming population into the current root node, then determine the feature x_j that best separates that incoming population. The feature x_j is chosen so as to maximize IG_j (or minimize $AGini_j$). This process is repeated by nominating the end of each new branch to be the current root node. Features can appear as a node more than once, so the order in which nodes are split does not matter. In essence, what we are doing is performing a top-down, greedy search through the space of possible dtrees.

The pseudo-code below describes the following historically important software programs:

- CART (Classification and Regression Trees), invented by Breiman et al in 1984. Uses $AGini_j$ as SAM.
- ID3 (Iterative Dichotomiser 3) invented by Quinlan in 1986. Uses IG_j as SAM. C4.5/C5.0 are successors to ID3.

CART and ID were invented around the same time. The main difference between them is the SAM being used.

The pseudo-code below uses the majority function defined in Chapter C.

Algorithm 1: Pseudo-code for learning a dtree from a dataset

Input :

- dataset $DS = \{(\sigma, x^\sigma, c^\sigma) : \sigma \in \Sigma\}$
- set of currently available node indices J, where $J \subset J_0$

Output:

- tree T,
- population numbers $\{(r, c, x_r, N_r(c, x_r)) : r \in J_0, c \in val(\underline{c}), x_r \in val(\underline{x}_r)\}$ stored globally

From DS, calculate J_0, $val(\underline{c})$, $val(\underline{x}_i)$ for each i
c^σ is called the target feature/attribute.
$J \leftarrow J_0$
Function `learn_dtree`(DS, J)**:**
  $\Sigma \leftarrow$ set of all σ in DS
  if $\{c^\sigma : \sigma \in \Sigma\} = \{c\}$ **then**
    $T \leftarrow$ one node tree with leaf node label$= c$
  else if $J = \emptyset$ **then**
    $T \leftarrow$ one node tree with leaf node label$=$ `majority`$([c^\sigma : \sigma \in \Sigma])$
  else
    $r \leftarrow \operatorname{argmax}_{j \in J} IG_j(DS)$ `// or replace` $\operatorname{argmax}_{j \in J} IG_j$ `by` $\operatorname{argmin}_{j \in J} AGini_j$
    from DS, calculate $\{(r, c, x_r, N_r(c, x_r)) : c \in val(\underline{c}), x_r \in val(\underline{x}_r)\}$ and store it globally
    for $v \in val(\underline{x}_r)$ **do**
      `/* Notice that` J `is the same every time repeat this loop, so order in which` $v \in val(\underline{x}_r)$ `are called does not matter. Furthermore, this means that multiple tree nodes may be labeled by same feature. */`
      On current tree T, add a branch below $\underline{x}_r$ with label "$\underline{x}_r = v$"
      $DS|_{\underline{x}_r = v} \leftarrow$ subset of DS with $\underline{x}_r = v$
      if $DS|_{\underline{x}_r = v} = \emptyset$ **then**
        below the new branch add a
        leaf node labeled $=$ `majority`$([c^\sigma : \sigma \in \Sigma])$
      else
        below the new branch add
        subtree $=$`learn_dtree`$(DS|_{\underline{x}_r = v}, J - \{r\})$
  return T

Chapter 17

Decisions Based on Rungs 2 and 3: COMING SOON

Chapter 18

Difference-in-Differences

This chapter is based on Ref.[15].

The Difference-in-Differences (DID) method was first used by John Snow in an 1854 report that argued that cholera in London was being transmitted by sewage polluted water rather than, as others at the time believed, by air (in fetid vapors called miasmas). In general, one can apply DID to discover causal effects in historical data. By **historical data** (a.k.a. a **natural experiment**. See Ref.[169]) we mean data that is collected long after the treatment (rather than during it) and is thus not subject to active intervention by the experimenter.

This chapter assumes that the reader has read Chapter 77 on Potential Outcomes (PO). The DID method applies the basic single-time PO theory described in Chapter 77, to 2 well separated times in which different conditions prevail.

18.1 John Snow, DID and a cholera transmission pathway

Let

$d \in \{0, 1\}$
$t \in \{t_0, t_1\}$, $t_0 < t_1$
$y = f(d, t) \in \mathbb{R}$.
Define

$$\Delta_t f(d, t) = f(d, t_1) - f(d, t_0) \ , \tag{18.1}$$

$$\Delta_d f(d, t) = f(1, t) - f(0, t) \ , \tag{18.2}$$

$$DID = \delta = \Delta_d \Delta_t f(d, t) \ . \tag{18.3}$$

DID is illustrated in Fig.18.1.

Figure 18.1: Pictorial representation of Difference-in-differences (DID) as a difference of two differences (i.e., a difference of two slopes).

A **time series** is any function of time for which the domain is a discrete set of times.

Figure 18.2: $DID = \delta$ expressed as difference of slopes or difference of vectors.

Note that, as shown Fig.18.2, $DID = \delta$ can also be expressed as a difference of 2 slopes times $\Delta t = t_1 - t_0$. Let $\hat{y}$ be a unit in the y direction. δ can also be expressed as the dot product of a difference of 2 vectors dotted with $\hat{y}$.

A condensation of the data collected by John Snow in 1854 is given in Table 18.1. From that data, we find that

$$\delta = \Delta_d \Delta_t f(d, t) = (19 - 85) - (147 - 135) = -66 - 12 = -78 \tag{18.4}$$

	$t = t_0$ (1849)	$t = t_1$ (1854)
$d = 1$ (town 1)	85 deaths, polluted DW	19 deaths, unpolluted DW
$d = 0$ (town 0)	135 deaths, polluted DW	147 deaths, polluted DW

Table 18.1: A condensation of the data collected by John Snow in 1854, to test the hypothesis that cholera in London was being spread by polluted drinking water (DW).

18.2 PO analysis

In this section, we show how to analyze the DID method using the formalism of PO theory.

We will speak of a treatment outcome $\underline{y}^{\sigma}_{t,g^{\sigma}}(c^{\sigma}, x^{\sigma})$ for individual σ that depends, not just on the treatment dose $c^{\sigma} \in \{0, 1\}$ and the confounder state x^{σ}, but also on a group parameter (i.e., which population or town) $g^{\sigma} \in \{0, 1\}$ and on a time parameter $t \in \{t_0, t_1\}$ (note t is independent of σ). Actually, we will assume $g^{\sigma} = c^{\sigma}$, so we will just speak of $\underline{y}^{\sigma}_{t}(c^{\sigma}, x^{\sigma})$ with no explicit g^{σ} dependence. As usual for PO theory, we will consider expected values of y^{σ}_{t}:

$$E_{\sigma|d,x}[y^{\sigma}_{t}(c)] = E_{\underline{y}_t(c)|d,x}[\underline{y}_t(c)] = \mathcal{Y}_{c|d,x}(t) \tag{18.5}$$

To calculate these expected values, we need a "model" with probability distributions. In this case, the needed model and probability distributions are provided by the bnets depicted in Fig.18.3. The TPMs, printed in blue, for the bnet $G_{t,+}$ in Fig.18.3, are as follows. Note that the TPMs for the bnet $G_{t,+}$ are defined in terms of the TPMs for the bnet G_t.

Figure 18.3: $t \in \{t_0, t_1\}$. Bnet $G_{t,+}$ is obtained by adding two new nodes $\underline{y}_t(0)$ and $\underline{y}_t(1)$ to bnet G_t.

$$P(x) = P_{\underline{x}}(x) \tag{18.6}$$

$$P(d|x) = P_{\underline{d}|\underline{x}}(d|x) \tag{18.7}$$

$$P(y_t|y_t(0), y_t(1), d) = \mathbb{1}(y_t = y_t(d)) \tag{18.8}$$

$$P(y_t(c)|x) = P(y_t(c)|d, x) = \text{given} \tag{18.9}$$

Figure 18.4: Four different time-dependent expected values $\mathcal{Y}_{c|d}(t)$ of y_t^σ for bnet $G_{t,+}$ The 4 magenta stars represents the 4 DID measurements.

We define the function $c(t)$ for $t = t_0, t_1$ by

$$c(t) = \begin{cases} 0 & \text{if } t = t_0 \\ d & \text{if } t = t_1 \end{cases} \tag{18.10}$$

Now we claim that the DID δ calculated in the previous section for John Snow's data, can be expressed in PO formalism as follows:

$$\delta = \Delta_d \Delta_t \mathcal{Y}_{c(t)|d}(t) \ . \tag{18.11}$$

Fig.18.4 depicts the four functions $\mathcal{Y}_{c|d}(t)$ for t in the interval $[t_0, t_1]$ and for $c, d \in \{0, 1\}$. The $\mathcal{Y}$ coordinates of the four magenta stars in Fig.18.4 can be calculated using bnet G_t.

Define the **parallel trends** (PT) by

$$PT = \Delta_d \Delta_t \mathcal{Y}_{0|d}(t) \ . \tag{18.12}$$

We will say the **parallel trends assumption (PTA)** holds if $PT = 0$.

Next we prove that the DID δ equals the sum of an ATT[1] and PT.

$$
\begin{aligned}
\delta &= \Delta_d \Delta_t \mathcal{Y}_{c(t)|d}(t) && (18.13)\\
&= \Big[\Delta_t \mathcal{Y}_{c(t)|1}(t) - \Delta_t \mathcal{Y}_{c(t)|0}(t)\Big] && (18.14)\\
&= \mathcal{Y}_{1|1}(t_1) - \mathcal{Y}_{0|1}(t_0) - \{\mathcal{Y}_{0|0}(t_1) - \mathcal{Y}_{0|0}(t_0)\} && (18.15)\\
&= \mathcal{Y}_{1|1}(t_1) - \mathcal{Y}_{0|1}(t_0) - \{\mathcal{Y}_{0|0}(t_1) - \mathcal{Y}_{0|0}(t_0)\} + \underbrace{\{\mathcal{Y}_{0|1}(t_1) - \mathcal{Y}_{0|1}(t_1)\}}_{\text{zero}} && (18.16)\\
&= \underbrace{\mathcal{Y}_{1|1}(t_1) - \mathcal{Y}_{0|1}(t_1)}_{ATT(t_1)} - \mathcal{Y}_{0|1}(t_0) - \{\mathcal{Y}_{0|0}(t_1) - \mathcal{Y}_{0|0}(t_0)\} + \mathcal{Y}_{0|1}(t_1) && (18.17)\\
&= ATT(t_1) - \Delta_t \mathcal{Y}_{0|0}(t) + \Delta_t \mathcal{Y}_{0|1}(t) && (18.18)\\
&= ATT(t_1) + \underbrace{\Delta_d \Delta_t \mathcal{Y}_{0|d}(t)}_{\text{zero if PTA holds}} && (18.19)
\end{aligned}
$$

18.3 Linear Regression

In this section, we show how to apply linear regression (LR) to the PO analysis of DID.

As before, let $y_t^\sigma(c^\sigma)$ be the treatment outcome for individual σ, who receives a treatment dose c^σ at times $t \in \{t_0, t_1\}$. $y_t^\sigma(c^\sigma)$ can be fitted as follows. Here ϵ^σ is the residual for individual σ, and $b_0, m_0, b_1, m_1 \in \mathbb{R}$ are the fit parameters.

$$
y_t^\sigma = [b_0 + m_0(t - t_0)](1 - c^\sigma) + [b_1 + m_1(t - t_0)]c^\sigma + \epsilon^\sigma \ . \tag{18.20}
$$

Note that Eq.(18.20) yields a straight line in the $y_t^\sigma - t$ plane for $c^\sigma = 0$, and another straight line for $c^\sigma = 1$. We are using the standard symbols b to denote the y-intercept, and m to denote the slope of a straight line.

Taking the expected value of Eq.(18.20), we get

$$
\mathcal{Y}_{c|d}(t) = [b_0 + m_0(t - t_0)](1 - c) + [b_1 + m_1(t - t_0)]c \ . \tag{18.21}
$$

If $\Delta t = t_1 - t_0$, then

$$
\mathcal{Y}_{d|d}(t_1) = [b_0 + m_0 \Delta t](1 - d) + [b_1 + m_1 \Delta t]d \ , \tag{18.22}
$$

and

$$
\mathcal{Y}_{0|d}(t_0) = b_0 \ . \tag{18.23}
$$

[1] ATT stands for the average treatment effect of the treated. ATT is defined in Chapter 77

Thus,

$$\begin{aligned}
\delta &= \Delta_d \Delta_t \mathcal{Y}_{c(t)|d}(t) && (18.24)\\
&= \Delta_d [\mathcal{Y}_{d|d}(t_1) - \mathcal{Y}_{0|d}(t_0)] && (18.25)\\
&= (m_1 - m_0)\Delta t \ . && (18.26)
\end{aligned}$$

Figure 18.5: We use Linear Regression to fit a straight line between points U_0 and U_1, and between points T_0 and T_1. (U=untreated, T=treated, subscript refers to times t_0, t_1). U_0, T_0, U_1, T_1 are the measurement points. Point I is an image of point U_1.

	$t = t_0$	$t = t_1$
$d = 1$	$\mathcal{Y}(T_0) = \mathcal{Y}_{0\|1}(t_0)$	$\mathcal{Y}(T_1) = \mathcal{Y}_{1\|1}(t_1)$
$d = 0$	$\mathcal{Y}(U_0) = \mathcal{Y}_{0\|0}(t_0)$	$\mathcal{Y}(U_1) = \mathcal{Y}_{0\|0}(t_1)$

Table 18.2: $\mathcal{Y}$ coordinates of points U_0, T_0, U_1, T_1 in Figs.18.4 and 18.5.

Figs.18.4 and 18.5 define points U_0, T_0, U_1, T_1, I, A. The $\mathcal{Y}$ coordinates of points U_0, T_0, U_1, T_1 are given by Table 18.2. The $\mathcal{Y}$ coordinates of points A, I are given by Eqs.18.27.

$$\mathcal{Y}(A) = \mathcal{Y}_{0|1}(t_1) \tag{18.27a}$$

$$\mathcal{Y}(I) = \mathcal{Y}(U_1) + [\mathcal{Y}(T_0) - \mathcal{Y}(U_0)] \tag{18.27b}$$

We can express ATT and the δ for DID in terms of the $\mathcal{Y}$ of the points U_0, T_0, U_1, T_1, I, A. Indeed,

$$\begin{aligned} \delta &= \mathcal{Y}(T_1) - \mathcal{Y}(I) && (18.28) \\ &= \mathcal{Y}(T_1) - \mathcal{Y}(U_1) - [\mathcal{Y}(T_0) - \mathcal{Y}(U_0)] && (18.29) \end{aligned}$$

$$ATT = \mathcal{Y}(T_1) - \mathcal{Y}(A) \tag{18.30}$$

Hence,

$$\delta = ATT \iff \mathcal{Y}(I) = \mathcal{Y}(A) \iff \text{PTA holds} \tag{18.31}$$

Chapter 19

Diffusion Models

This chapter is based on Ref.[103]

Diffusion Models (DM) are a way of generating fake images from an original image. They are a competitor to GANs (see Chapter 36), and are used in DALL-E (OpenAI's computer program that generates images from text).

DM works by subjecting each degree of freedom of the image to a forward Markov chain of transformations labeled $t = 0, 1, 2, \ldots, T$ and a reverse Markov chain of transformations labeled $t = T-1, T-2, \ldots, 0$. Each step of the forward chain multiplies each degree of freedom of the image by a constant and adds white noise to it. The last image ($t = T$) of the forward chain is changed to the point that it looks like a normal distribution. The reverse chain tries to undo, as well as possible, the alterations to the original image done by the forward chain. Full, faithful restoration is impossible because the forward chain is an irreversible process.

Of course, a DM is only a small part of the magic of DALL-E. I haven't studied DALL-E's algorithm, but my guess is that it works roughly as follows. Given a text description of an image, such as "A hedgehog using a calculator painted in the style of Vincent van Gogh", it uses a neural net trained on a vast corpus of words and images, to match separate words in the description with an image for each word. Then it uses a second neural net to create a pastiche/superposition from the set of images created in the first stage. Then, it uses a DM to smooth the transitions of the pastiche or assign different weights to the elements of the superposition. Finally, it modifies the image at this point by passing it through a photoshop-like stylistic filter that can be specified in the initial description.

19.1 Bnet for DM

For $t \in 1, 2, \ldots, T$, let

$$0 < \alpha^t \ < 1, \ \beta^t = 1 - \alpha^t \tag{19.1}$$

$$\pi_1^t \alpha = \prod_{\tau=1}^{t} \alpha^\tau \tag{19.2}$$

Let

$\underline{x}^t, \underline{\widetilde{x}}^t \in \mathbb{R}^{df}$ for $t = 0, 1, 2, \ldots, T$ be column vectors describing an image. Here df is the number of degrees of freedom of the image. $\underline{x}^0$ will denote the initial Original image, $\underline{x}^T = \underline{\widetilde{x}}^T$ will denote an image which is very close to a normal distribution, and $\underline{\widetilde{x}}^0$ will denote the final Fake image.

$I \in \mathbb{R}^{df \times df}$ is the identity matrix

$\underline{w}^t, \underline{a}^t, \underline{n}^t_\theta \in \mathbb{R}^{df}$

Fig.19.1 shows a bnet for DM, and 19.2 shows the same bnet in more detail.[1]

Figure 19.1: Bnet for DM with $T = 3$. See Chapter 52 for an explanation of LDEN notation.

The TPMs, printed in blue, for the bnet of Fig.19.1 are as follows:

$$P(x^0) = \delta(x^0, X^0) \quad \text{(original image)} \tag{19.3}$$

$$P(x^t | x^{t-1}, w^t) = \mathbb{1}(\quad x^t = \sqrt{\alpha^t}\, x^{t-1} + \sqrt{\beta^t}\, w^{t-1} \quad) \tag{19.4}$$

$$P(w^t) = \mathcal{N}(w^t; \mu = 0, \sigma^2 = I) \quad \text{(white noise)} \tag{19.5}$$

$$\begin{aligned} P(\underline{\widetilde{x}}^{t-1} = x^{t-1} | \underline{\widetilde{x}}^t = x^t) &= \widetilde{P}_\theta(x^{t-1} | x^t) && (19.6) \\ &= \mathcal{N}(x^{t-1}; \mu = M^{t-1}_\theta(x^t), \sigma^2 = \Sigma^{t-1}_\theta(x^t)) && (19.7) \end{aligned}$$

[1] Ref.[103] uses $\underline{z}^t$ instead of $\underline{w}^t$ for white noise. Note that the time index of $\underline{w}^t$ often does not matter because $\underline{w}^t \sim \mathcal{N}(0, I)$ for all t. This does not mean that we can drop the time index of $\underline{w}^t$, because $\underline{w}^{t_1}$ and $\underline{w}^{t_2}$ are uncorrelated for $t_1 \neq t_2$, so we can get into trouble if we assume $\underline{w}^{t_1} = \underline{w}^{t_2} = \underline{w}$.

We will assume that

$$\Sigma_\theta^{t-1}(x^t) = (\sigma^{t-1})^2 I \tag{19.8}$$

where $\sigma^{t-1} \in \mathbb{R}$.

Figure 19.2: The bnet of Fig.19.1 shown in more detail.

The TPMs, printed in blue, for the bnet of Fig.19.2 are as follows:

$$P(n_\theta^t) = \mathcal{N}(n_\theta^t; \mu = M_\theta^t(x^{t+1}), \sigma = \sigma^t) \tag{19.9}$$

$$P(a^t | \underline{\widetilde{x}}^{t+1} = x^{t+1}) = \mathbb{1}(\quad a^t = M^t(x^{t+1}) \quad) \tag{19.10}$$

$$P(\underline{\widetilde{x}}^t = x^t | a_t, n_\theta^t) = \mathbb{1}(\quad x^t = a^t - n_\theta^t \quad) \tag{19.11}$$

Note that these TPMs for the bnet Fig.19.2 imply that

$$\underbrace{P(\underline{\widetilde{x}}^t = x^t | \underline{\widetilde{x}}^{t+1} = x^{t+1})}_{\text{call this } \widetilde{P}_\theta(x^t|x^{t+1})} = \quad \mathcal{N}(x^t; \mu = M^t(x^{t+1}) - M_\theta^t(x^{t+1}), \sigma = \sigma^t) \tag{19.12}$$

19.2 Mean Values $M^{t-1}(x^t)$ and $M_\theta^{t-1}(x^t)$

Claim 37

$$\boxed{\underline{x}^t = \sqrt{\pi_1^t \alpha}\ \underline{x}^0 + \sqrt{1-\pi_1^t \alpha}\ \underline{w}^t} \tag{19.13}$$

$$\begin{tikzcd} & \underline{w}^t \arrow[d, "\sqrt{1-\pi_1^t\alpha}"] \\ \underline{x}^0 \arrow[r, "\sqrt{\pi_1^t\alpha}"'] & \underline{x}^t \end{tikzcd} \tag{19.14}$$

proof:

Suppose $\underline{x}_1$ and $\underline{x}_2$ are independent random variables with variances V_1 and V_2, respectively. Then the variance V of $\underline{x} = \underline{x}_1 + \underline{x}_2$ is

$$\begin{aligned} V &= \langle \underline{x}, \underline{x} \rangle && (19.15)\\ &= \langle \underline{x}_1 + \underline{x}_2, \underline{x}_1 + \underline{x}_2 \rangle && (19.16)\\ &= \langle \underline{x}_1, \underline{x}_1 \rangle + \langle \underline{x}_2, \underline{x}_2 \rangle && (19.17)\\ &= V_1 + V_2 && (19.18) \end{aligned}$$

By similar reasoning, the mean of $\underline{x}$ equals the sum of the means of $\underline{x}_1$ and $\underline{x}_2$. It's also true that if both $\underline{x}_1$ and $\underline{x}_2$ are normally distributed, then $\underline{x}$ is too. We will refer to a sum of independent normals as a SIN.

Let $\underline{w} \sim \mathcal{N}(0, I)$.

$$\begin{aligned} \underline{x}^t &= \sqrt{\alpha^t}\underline{x}^{t-1} + \sqrt{1-\alpha^t}\ \underline{w}^{t-1} && (19.19)\\ &= \sqrt{\alpha^t}[\sqrt{\alpha^{t-1}}\underline{x}^{t-2} + \sqrt{1-\alpha^{t-1}}\ \underline{w}^{t-2}] + \sqrt{1-\alpha^t}\ \underline{w}^{t-1} && (19.20)\\ &= \sqrt{\alpha^t\alpha^{t-1}}\ \underline{x}^{t-2} + [\sqrt{\alpha^t(1-\alpha^{t-1})}\ \underline{w}^{t-2} + \sqrt{1-\alpha^t}\ \underline{w}^{t-1}] && (19.21)\\ &= \sqrt{\alpha^t\alpha^{t-1}}\ \underline{x}^{t-2} + \sqrt{1-\alpha^t\alpha^{t-1}}\ \underline{w} \quad \text{(because it's a SIN)} && (19.22)\\ &= \cdots && (19.23)\\ &= \sqrt{\pi_1^t\alpha}\ \underline{x}^0 + \sqrt{1-\pi_1^t\alpha}\ \underline{w} && (19.24) \end{aligned}$$

Now replace $\underline{w}$ by $\underline{w}^t$. This is justified because they are both $\mathcal{N}(0, I)$, and, when $t_1 \neq t_2$, we want the $\underline{w}$ for $\underline{x}^{t_1}$ to be independent from the $\underline{w}$ for $\underline{x}^{t_2}$.

QED

Solving Eq.(19.13) for $\underline{x}^0$, we get

$$\boxed{\underline{x}^0 = \frac{1}{\sqrt{\pi_1^t \alpha}} \underline{x}^t - \frac{\sqrt{1-\pi_1^t \alpha}}{\sqrt{\pi_1^t \alpha}} \underline{w}^t} \tag{19.25}$$

$$\begin{tikzcd}
 & \underline{w}^t \arrow[dl, "-\frac{\sqrt{1-\pi_1^t\alpha}}{\sqrt{\pi_1^t\alpha}}"'] \\
\underline{x}_0 & \underline{x}^t \arrow[l, "\frac{1}{\sqrt{\pi_1^t\alpha}}"]
\end{tikzcd} \tag{19.26}$$

Claim 38

$$P(x^{t-1}|x^t, x^0) = P(x^t|x^{t-1}) \frac{P(x^{t-1}|x^0)}{P(x^t|x^0)} \tag{19.27}$$

proof:

This is a simple consequence of Bayes rule. We will prove this in two ways: algebraically and graphically.

The algebraic proof goes as follows.

$$P(x^{t-1}|x^t, x^0) P(x^t|x^0) = \overbrace{P(x^t|x^{t-1}, x^0)}^{P(x^t|x^{t-1})} P(x^{t-1}|x^0) \tag{19.28}$$

The graphical proof, although longer, is more intuitive. Start by noticing that

$$\begin{tikzcd}
 & \sum w^0 \arrow[d] & \cdots & \sum w^{t-2} \arrow[d] & \sum w^{t-1} \arrow[d] \\
x^0 \arrow[r] & \sum x^1 \arrow[r] & \cdots \arrow[r] & x^{t-1} \arrow[r] & x^t
\end{tikzcd}$$

$$= \begin{tikzcd} x^0 \arrow[r] & x^{t-1} \arrow[r] & x^t \end{tikzcd} \text{ (diagram A)} \tag{19.29}$$

$$= \begin{tikzcd} x^0 \arrow[r] \arrow[rr, bend right, "0"'] & x^{t-1} \arrow[r] & x^t \end{tikzcd} \quad \text{(make fully connected)}$$

$$= \begin{tikzcd} x^0 \arrow[r] \arrow[rr, bend right] & x^{t-1} & x^t \arrow[l] \end{tikzcd} \quad \text{(reverse arrow) (diagram B)}$$

Therefore,

$$\underbrace{P(x^{t-1}|x^t, x^0) P(x^t|x^0)}_{\text{diagram B of Eq.19.29}} = \underbrace{P(x^t|x^{t-1}) P(x^{t-1}|x^0)}_{\text{diagram A of Eq.19.29}} \tag{19.30}$$

QED

Define the following mean value:

$$M^{t-1}(x^t) = \sum_{x^{t-1}} x^{t-1} P(x^{t-1}|x^t, x^0) \tag{19.31}$$

Claim 39

$$\boxed{M^{t-1}(x^t) = \frac{\sqrt{\alpha^t}(1-\pi_1^{t-1}\alpha)}{1-\pi_1^t\alpha} x^t + \frac{\sqrt{\pi_1^{t-1}\alpha}\,\beta^t}{1-\pi_1^t\alpha} x^0} \tag{19.32}$$

proof:

By Claim 38, we know that $\mathcal{Q} = P(x^{t-1}|x^t, x^0)$ can be expressed as a product of two Gaussians $P(x^t|x^{t-1})$ and $P(x^{t-1}|x^0)$, divided by a third Gaussian $P(x^t|x^0)$, and all 3 of these Gaussians have been calculated in closed form previously in this chapter. So it's just a matter of algebra to express $\mathcal{Q}$ as a Gaussian, complete the square inside the exponent of $\mathcal{Q}$, and thus obtain its mean value $M^{t-1}(x^t)$. We leave the algebra to the reader. If in doubt, the algebra can be found in Ref.[103].

QED

Claim 40

$$\boxed{M^{t-1}(x^t) = \frac{1}{\sqrt{\alpha^t}} \left(x^t - \frac{\beta^t}{\sqrt{1-\pi_1^t\alpha}} w^t \right)} \tag{19.33}$$

proof:

Use Eq.(19.25) to replace x^0 in Eq.(19.32).

QED

Let us parameterize $M_\theta^{t-1}(x^t)$ by

$$\boxed{M_\theta^{t-1}(x^t) = \frac{1}{\sqrt{\alpha^t}} \left(x^t - \frac{\beta^t}{\sqrt{1-\pi_1^t\alpha}} n_\theta^{t-1}(x^t) \right)} \tag{19.34}$$

If we define

$$C^t = \frac{-\beta^t}{\sqrt{\alpha^t(1-\pi_1^t\alpha)}}\ , \tag{19.35}$$

then

$$M^{t-1}(x^t) - M_\theta^{t-1}(x^t) = C^t[w^t - n_\theta^{t-1}(x^t)] \tag{19.36}$$

$$= C^t[w^t - n_\theta^{t-1}(\sqrt{\pi_1^t\alpha}\, x^0 + \sqrt{1-\pi_1^t\alpha}\, w^t)] \tag{19.37}$$

19.3 Loss function $\mathcal{L}$

Note that[2]

$$
\begin{aligned}
0 &\leq D_{KL}(P(x^{1:T}|x^0) \parallel \widetilde{P}_\theta(x^{1:T}|x^0)) && (19.38)\\
&= \sum_{x^{1:T}} P(x^{1:T}|x^0) \ln \frac{P(x^{1:T}|x^0)}{\widetilde{P}_\theta(x^{1:T}|x^0)} && (19.39)\\
&= \ln \widetilde{P}_\theta(x^0) + \sum_{x^{1:T}} P(x^{1:T}|x^0) \ln \frac{P(x^{1:T}|x^0)}{\widetilde{P}_\theta(x^{0:T})} && (19.40)
\end{aligned}
$$

$$
\underbrace{-\sum_{x^0} P(x^0) \ln \widetilde{P}_\theta(x^0)}_{-E_{x^0}[\ln \widetilde{P}_\theta(x^0)]} \leq \underbrace{\sum_{x^{0:T}} P(x^{0:T}) \ln \frac{P(x^{1:T}|x^0)}{\widetilde{P}_\theta(x^{0:T})}}_{E_{x^{0:T}}\left[\ln \frac{P(x^{1:T}|x^0)}{\widetilde{P}_\theta(x^{0:T})}\right] = \mathcal{L}(\theta) = \text{ loss function}} \tag{19.41}
$$

Henceforth, expected values $E_{x^{0:T}}$ are to be understood as being with respect to $P(x^{0:T})$.

The left hand side of the inequality Eq.(19.41) is expected to be a small positive constant, because we expect $\widetilde{P}_\theta(x^0) \approx P(x^0) \approx \delta(x^0, X^0)$. Thus, we are justified in defining the right hand side of Eq.(19.41) as loss function $\mathcal{L} = \mathcal{L}(\theta)$ to be minimized.

Claim 41

$$
\mathcal{L} = \sum_{t=0}^{T} \mathcal{L}^t \tag{19.42}
$$

$$
\mathcal{L}^0 = -\ln \widetilde{P}_\theta(x^0|x^1) \tag{19.43}
$$

$$
\mathcal{L}^{t-1} = E_{x^{0:T}}\left[\ln \frac{P(x^{t-1}|x^t, x^0)}{\widetilde{P}_\theta(x^{t-1}|x^t)}\right] \qquad \textit{for } t = 2, 3, \ldots T \tag{19.44}
$$

$$
\mathcal{L}^T = E_{x^{0:T}}\left[\ln \frac{P(x^T|x^0)}{\widetilde{P}_\theta(x^T)}\right] \tag{19.45}
$$

[2] D_{KL} is the Kullback-Leibler divergence. It's defined in Chapter C.

proof:

$$\begin{aligned}
\mathcal{L} &= E_{x^{0:T}}\left[\ln \frac{P(x^{1:T}|x^0)}{\widetilde{P}_\theta(x^{0:T})}\right] && (19.46)\\
&= E_{x^{0:T}}\left[\ln \frac{\prod_{t=1}^T P(x^t|x^{t-1})}{\widetilde{P}_\theta(x^T)\prod_{t=1}^T \widetilde{P}_\theta(x^{t-1}|x^t)}\right] && (19.47)\\
&= E_{x^{0:T}}\left[\ln \frac{P(x^1|x^0)\prod_{t=2}^T \overbrace{P(x^t|x^{t-1})}^{P(x^t|x^{t-1},x^0)}}{\widetilde{P}_\theta(x^T)\widetilde{P}_\theta(x^0|x^1)\prod_{t=2}^T \widetilde{P}_\theta(x^{t-1}|x^t)}\right] && (19.48)\\
&= E_{x^{0:T}}\left[\ln \frac{P(x^1|x^0)\prod_{t=2}^T P(x^{t-1}|x^t,x^0)\frac{P(x^t|x^0)}{P(x^{t-1}|x^0)}}{\widetilde{P}_\theta(x^T)\widetilde{P}_\theta(x^0|x^1)\prod_{t=2}^T \widetilde{P}_\theta(x^{t-1}|x^t)}\right] && (19.49)\\
&= E_{x^{0:T}}\left[\ln \frac{\cancel{P(x^1|x^0)}\frac{P(x^T|x^0)}{\cancel{P(x^1|x^0)}}\prod_{t=2}^T P(x^{t-1}|x^t,x^0)}{\widetilde{P}_\theta(x^T)\widetilde{P}_\theta(x^0|x^1)\prod_{t=2}^T \widetilde{P}_\theta(x^{t-1}|x^t)}\right] && (19.50)\\
&= E_{x^{0:T}}\left[\ln \frac{1}{\widetilde{P}_\theta(x^0|x^1)}\cdot\frac{P(x^T|x^0)}{\widetilde{P}_\theta(x^T)}\cdot\frac{\prod_{t=2}^T P(x^{t-1}|x^t,x^0)}{\prod_{t=2}^T \widetilde{P}_\theta(x^{t-1}|x^t)}\right] && (19.51)
\end{aligned}$$

QED

We expect $\widetilde{P}_\theta(x^0|x^1)$ in $\mathcal{L}^0$ to depend only very weakly on θ because it comes at the end of the reverse Markov chain. Likewise, we expect $\widetilde{P}_\theta(x^T)$ in $\mathcal{L}^T$ to be approximately a normal distribution independent of θ, because it comes at the beginning of the reverse Markov chain. Hence, we are justified in using $\mathcal{L}' = \sum_{t=2}^T \mathcal{L}^{t-1}$ as the new loss function to me minimized with respect to θ.

Claim 42

$$\begin{aligned}
\mathcal{L}^{t-1} &= E_{x^t}\left[\frac{1}{2(\sigma^{t-1})^2}[M^{t-1}(x^t) - M_\theta^{t-1}(x^t)]^2\right] && (19.52)\\
&= E_{x^0,w^t}\left[\frac{(C^t)^2}{2(\sigma^{t-1})^2}\left[w^t - n_\theta^{t-1}(\sqrt{\pi_1^t\alpha}\,x^0 + \sqrt{1-\pi_1^t\alpha}\,w^t)\right]^2\right] && (19.53)
\end{aligned}$$

proof:

Eq.(19.53) follows trivially from Eq.(19.37) and Eq.(19.52).

To show Eq.(19.52), recall that

$$\mathcal{L}^{t-1} = E_{x^{0:T}}\left[\ln \frac{P(x^{t-1}|x^t,x^0)}{\widetilde{P}_\theta(x^{t-1}|x^t)}\right] \qquad (19.54)$$

The denominator $\widetilde{P}_\theta(x^{t-1}|x^t)$ is one of the TPMs for the bnet Fig.19.2 , and it's a normal distribution with mean $M_\theta^{t-1}(x^t)$. The numerator $P(x^{t-1}|x^t,x^0)$ is a

normal distribution too; it was calculated in the proof of Claim 39, where its mean was called $M^{t-1}(x^t)$. Hence, after some algebra which is left to the reader, one can show Eq.(19.52).

QED

Eq.(19.53) for $\mathcal{L}^{t-1}$ is usually simplified to

$$\mathcal{L}_{simple} = E_{x^0,w^t}\left[\left[w^t - n_\theta^{t-1}(\sqrt{\pi_1^t\alpha}\; x^0 + \sqrt{1-\pi_1^t\alpha}\; w^t)\right]^2\right] \tag{19.55}$$

19.4 Algorithms for training and sampling DM

Algorithm 2: Algorithm for training DM (i.e., finding optimum θ)

Input : $\{\beta^t\}_{t=1}^T$, $0 < \epsilon < 1$, $n_\theta^t(x)$ function, θ_{in}, $\Delta\theta_{in}$, $P(x^0)$, T, $\eta > 0$
Output: θ_{next} = optimal θ
$\Delta\theta = \Delta\theta_{in}$
$\theta_{next} = \theta_{in}$
while $|\Delta\theta| > \epsilon$ **do**
  Choose $x^0 \sim P(x^0)$
  Choose $t \sim Uniform(\{1, 2, \ldots, T\})$
  Choose $w \sim \mathcal{N}(0, I)$
  $\theta = \theta_{next}$
  `// Gradient descent for simple loss function given by Eq.(19.55).`
  $\theta_{next} = \theta + \eta\partial_\theta\left[w - n_\theta^{t-1}(\sqrt{\pi_1^t\alpha}x^0 + \sqrt{1-\pi_1^t\alpha}w)\right]^2$
  $\Delta\theta = \theta_{next} - \theta$
return θ_{next}

Algorithm 3: Algorithm for sampling DM (i.e., finding fake image x^0)

Input : $n_\theta^t(x)$, where θ is optimal, T, $\{\alpha^t\}_{t=1}^T$, $\{\sigma^t\}_{t=1}^T$
Output: x^0 = fake image
Choose $x^T \sim \mathcal{N}(0, I)$
for $t = T, T-1, \ldots, 2, 1$ **do**
  Choose $w \sim \mathcal{N}(0, I)$ if $t > 1$, else $w = 0$
  `// See Eq.19.33`
  $x^{t-1} = \frac{1}{\sqrt{\alpha^t}}\left(x^t - \frac{\beta^t}{\sqrt{1-\pi_1^t\alpha}}n_\theta^{t-1}(x^t)\right) + \sigma^t w$
return x^0

Chapter 20

Digital Circuits

Figure 20.1: Typical digital circuit of NAND gates.

Digital (logic) gate: node with na input ports and nx output ports which represents a function

$$f : \{0,1\}^{na} \to \{0,1\}^{nx} \ . \tag{20.1}$$

Suppose
$a^{na} = (a_i)_{i=0,1,\ldots,na-1}$ where $a_i \in \{0,1\}$,
$x^{nx} = (x_i)_{i=0,1,\ldots,nx-1}$ where $x_i \in \{0,1\}$.
f maps a^{na} into x^{nx}.
Digital circuit (dcircuit) = circuit of digital gates.

20.1 Mapping any dcircuit to a bnet

20.1.1 Option A of Fig.20.2

1. Replace every dcircuit gate described by Eq.(20.1) by nx bnet nodes $\underline{x}_i$ for $i = 0, 1, \ldots, nx-1$ such that

Figure 20.2: 2 options for mapping dcircuit node with multiple output ports into bnet.

$$P(x_i|a^{na}) = \delta(x_i, f_i(a^{na})) \tag{20.2}$$

2. Replace all connectors of the dcircuit by arrows pointing in the direction of the bit flow.

20.1.2 Option B of Fig.20.2

1. Replace every dcircuit gate described by Eq.(20.1) with one bnet node called $\underline{x}^{nx}$ and, if $nx > 0$, nx "marginalizer nodes" $\underline{m}_i$ for $i = 0, 1, \ldots, nx - 1$, such that

$$P(x^{nx}|a^{na}) = \delta(x^{nx}, f(a^{na})) \ , \tag{20.3}$$

and

$$P(m_i|x^{nx}) = \delta(m_i, x_i) \ . \tag{20.4}$$

2. Replace all connectors of the dcircuit by arrows pointing in the direction of the bit flow.

Options A and B don't work for digital circuits with feedback loops such as flip-flops. Those could probably be modeled with dynamical bnets.

Chapter 21

Dimensionality Reduction

This chapter is based on Ref. [130].

Suppose you have a dataset with 500 feature columns $\underline{X}$ plus a treatment column $\underline{t}$ and an outcome column $\underline{y}$. Causal DAG discovery software such as bnlearn (see Chapter 95) cannot handle that much data. In order to make any headway in this situation, some kind of lossy data compression is required. That is precisely what **dimensionality reduction** (DR) is: a type of lossy data compression for datasets whereby the number of feature columns is reduced. More specifically, DR of a 500 feature dataset might entail the following steps:

1. choose from the 500 features $\underline{X}$, a bunch of disjoint feature subsets $\underline{S}_i$ such that in each subset, the variables are very strongly correlated as in a clique, so they truly act as if they are inseparable, as a single node that we shall refer to as a combined node.

2. reduce the number of values (i.e., states) of the combined nodes using DR.

3. run causal discovery on the uncombined nodes, combined nodes and $(\underline{t}, \underline{y})$ nodes.

The above steps for **bnet coarsening** can be viewed as a special case of what physicists call a **Renormalization Group** (RG) transformation. Ken Wilson received a Nobel Prize in 1982 for his RG theory.

In our steps for bnet coarsening, we did not specify how to do DR. The most common method for doing DR is Principal Component Analysis (PCA) (see Chapter 78). PCA only keeps the largest principal components so it is truly lossy. Non-negative Matrix Factorization (NN MF) (see Chapter 70) and Factor Analysis (FA) (see Chapter 28) are often mentioned as other methods, besides PCA, for doing DR. The way I've defined DR, both of these methods aren't truly DR methods because both of them are non-lossy.

The initial number of feature columns of a dataset is often reduced via PCA before doing FA.

It is often said that the NN MF method doesn't throw away the mean value $\langle X \rangle$ of the hidden variable X, whereas PCA does. However, even though the calculation of

the principal components of X does assume $\langle X \rangle = 0$, and therefore $\langle Y \rangle = \langle X \rangle W = 0$, we can certainly calculate and store the truncated versions of $\langle X \rangle$ or $\langle Y \rangle$.

Chapter 22

Do Calculus

The Do Calculus (DC) and associated ideas were invented by Judea Pearl and collaborators. This chapter is based on Judea Pearl's books (see Chapter A).

When doing Do Calculus, it is convenient to separate the nodes of a bnet into 2 types: **observed**, and **hidden (i.e., unobserved, latent, unmeasured, non-visible)**, depending on whether data describing the state of that node is available (i.e., measured) or not. In this chapter, every hidden node will be indicated in a bnet diagram by either: (1) enclosing its random variable in a dashed circle or (2) making the arrows coming out of it dashed. Accordingly, the 3 diagrams in Fig.22.1 all mean the same thing.

A **confounder node** $\underline{c}$ for nodes $\underline{x}$ and $\underline{y}$ is a root node with arrows pointing from it to both $\underline{x}$ and $\underline{y}$. Thus, $\underline{c}$ acts as a **common cause** of $\underline{x}$ and $\underline{y}$. In general, confounders can be either observed or hidden nodes. The word "confounder" itself just means that it confuses the analysis. It says nothing about whether it is hidden or not. The node $\underline{c}$ in Fig.22.1) is a **hidden confounder**.

Figure 22.1: These 3 diagrams are equivalent. They mean that node $\underline{c}$ is hidden. Node $\underline{c}$ is implicit in the middle diagram.

Let $\mathcal{D}_{\underline{x}}$ be an operator that acts on a graph G with a node $\underline{x}$ by deleting all the arrows entering $\underline{x}$, thus converting $\underline{x}$ into a new node $\mathcal{D}\underline{x}$ that is a root node. Let $\mathcal{L}_{\underline{x}}$ be an operator that acts on a graph G with a node $\underline{x}$ by deleting all the arrows leaving $\underline{x}$, thus converting $\underline{x}$ into a new node $\mathcal{L}\underline{x}$ that is a leaf node. $\mathcal{D}_{\underline{x}}$ and $\mathcal{L}_{\underline{x}}$ are depicted in Fig.22.2. [1]

[1]Pearl uses $\mathcal{D}_X G = G_{\overline{X}}$ and $\mathcal{L}_X G = G_{\underline{X}}$ for a random variable X in a graph G. The way I

Figure 22.2: The do operator $\mathcal{D}_{\underline{x}}$ converts node $\underline{x}$ into a root node $\mathcal{D}\underline{x}$. The leaf operator $\mathcal{L}_{\underline{x}}$ converts node $\underline{x}$ into a leaf node $\mathcal{L}\underline{x}$.

If you don't know yet what we mean by a multi-node $\underline{a}.$, see Chapter F.

Given a bnet G, we define as follows the operators $\mathcal{D}_{\underline{a}.}$ and $\mathcal{L}_{\underline{a}.}$ for a multi-node $\underline{a}.$.

$$\mathcal{D}_{\underline{a}.}G = \left[\prod_j \mathcal{D}_{\underline{a}_j}\right] G \ , \quad \mathcal{L}_{\underline{a}.}G = \left[\prod_j \mathcal{L}_{\underline{a}_j}\right] G \ . \tag{22.1}$$

Consider a bnet whose totality of nodes is labeled $\underline{X}.$. Recall that

$$P(X.) = \prod_j P(X_j|(X_k)_{k:\underline{X}_k \in pa(\underline{X}_j)}) \ . \tag{22.2}$$

Define an operator $\mathcal{D}$ that acts as follows[2]: Let $X. - a. = (X_k)_{k:\underline{X}_k \notin \underline{a}.}$.

$$\begin{aligned} P(X. - a.|\mathcal{D}\underline{a}. = a.) &= \mathcal{N}(!(X. - a.)) \frac{P(X.)}{\prod_{j:\underline{X}_j \in \underline{a}.} P(X_j|(X_k)_{k:\underline{X}_k \in pa(\underline{X}_j)})} && (22.3) \\ &= \mathcal{N}(!(X. - a.)) \prod_{j:\underline{X}_j \notin \underline{a}.} P(X_j|(X_k)_{k:\underline{X}_k \in pa(\underline{X}_j)}) && (22.4) \\ &\neq P(X. - a.|\underline{a}. = a.) \ . && (22.5) \end{aligned}$$

Also,

$$P(\mathcal{D}\underline{a}. = a.) = \delta(a'., a.) \ . \tag{22.6}$$

In words, we replace the TPM for multinode $\underline{a}.$ by a delta function.

For instance, for the bnet

$$\underline{r} \longrightarrow \underline{x} \longrightarrow \underline{y} \tag{22.7}$$

with

$$P(r, x, y) = P(y|x)P(x|r)P(r) \ , \tag{22.8}$$

remember Pearl's notation is top-in (as in topping), and bottom-out (as in butt-out).

[2]As usual, $\mathcal{N}(!x)$ denotes a constant that is independent of x.

one has

$$P(r, y|\mathcal{D}\underline{x} = x) = P(y|x)P(r) \tag{22.9}$$

Hence,

$$P(y|\mathcal{D}\underline{x} = x) = P(y|x) \tag{22.10}$$

For the bnet

$$\begin{tikzcd} \underline{c} \arrow[d] \arrow[dr] & \\ \underline{x} \arrow[r] & \underline{y} \end{tikzcd} \tag{22.11}$$

with

$$P(x, y, c) = P(y|x, c)P(x|c)P(c) \ , \tag{22.12}$$

one has

$$P(y, c|\mathcal{D}\underline{x} = x) = P(y|x, c)P(c) \ . \tag{22.13}$$

Hence,

$$P(y|\mathcal{D}\underline{x} = x) = \sum_c P(y|x, c)P(c) \ . \tag{22.14}$$

This is called **adjusting the parents of** $\underline{x}$.

For $\underline{b}. \subset \underline{X}. - \underline{a}.$, define

$$P(b.|\mathcal{D}\underline{a}. = a.) = \sum_{X.-a.-b.} P(X. - a.|\mathcal{D}\underline{a}. = a.) \ , \tag{22.15}$$

and for $\underline{s}. \subset \underline{X}. - \underline{a}. - \underline{b}.$, define

$$P(b.|\mathcal{D}\underline{a}. = a., s.) = \frac{P(b., s.|\mathcal{D}\underline{a}. = a.)}{P(s.|\mathcal{D}\underline{a}. = a.)} \ . \tag{22.16}$$

$P(b.|\mathcal{D}\underline{a}. = a., s.)$ is denoted by Pearl by $P(b.|do(\underline{a}. = a.), s.)$. I prefer to use $\mathcal{D}$ instead of $do()$. I will still call $\mathcal{D}$ a **do operator**.

In $P(y|\mathcal{D}\underline{x} = x)$, node $\underline{x}$ is turned into a root node. This guarantees that there is no confounding node connecting $\underline{x}$ and $\underline{y}$. Such confounding nodes are unwelcomed when calculating causal effects between the 2 variables $\underline{x}$ and $\underline{y}$ because they introduce non-causal correlations between the two. This is also what happens in a **Randomized Controlled Trial (RCT)**. In an RCT with treatment $\underline{x}$, the value of $\underline{x}$ for each patient is determined by a coin toss, effectively turning $\underline{x}$ into a root node. Hence, the do operator mimics an RCT.

$P(b.|\mathcal{D}\underline{a}. = a., s.)$ is said to be **do-identifiable** (i.e., expressible without do()) if it can be expressed in terms of probability distributions that only depend on observed variables, and that have no do operators in them.[3]

For $\underline{x}, \underline{y} \in \{0, 1\}$, the **average causal effect (ACE)** is defined as

$$ACE = P(y = 1|\mathcal{D}\underline{x} = 1) - P(y = 1|\mathcal{D}\underline{x} = 0) \tag{22.17}$$

and the **Risk Difference (RD)** is defined as

$$RD = P(y = 1|\underline{x} = 1) - P(y = 1|\underline{x} = 0) \ . \tag{22.18}$$

22.1 3 Rules of Do Calculus

Throughout this section, suppose $\underline{a}., \underline{b}., \underline{r}., \underline{s}.$ are disjoint multinodes in a bnet G.

I this section, we will state the 3 rules for Do Calculus. All 3 rules start with a hypothesis which declares a bnet that I like to call the **hypothesis graph** (denoted by G_h). Then they propose how to transform a bnet G_e to a bnet $G_{e'}$. I like to call G_e and $G_{e'}$ the **evolving graphs**. The first G_e results from transforming the **original graph** G.

Recall from Chapter 24 on d-separation, that $(\underline{b}. \perp_G \underline{a}.|\underline{r}., \underline{s}.)$ means that we have established from the d-separation rules that all paths in G from $\underline{a}.$ to $\underline{b}.$ are blocked if we condition on $\underline{r}. \cup \underline{s}.$. Recall also that:

- **Rule 0: Insertion or deletion of condition $\underline{a}. = a.$, in the absence do-operator conditions.**

  If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ (i.e., $H(\underline{b}. : \underline{a}.|\underline{r}., \underline{s}.) = 0$) in $G_h = G$, then
  $P_{G_e}(b.|a., r., s.) = P_{G_e}(b.|r., s.)$

Zeroing an arrow is the same as deleting it. $\underline{a}.[\not\rightarrow \underline{b}.]$ is the promise that there are no unblocked paths from node $\underline{a}.$ to node $\underline{b}.$.

The 3 rules of Do Calculus can be presented in the same format as Rule 0.

[3]In Statistics, one says a probability distribution $P(x; \theta)$ of x that depends on a parameter θ is **identifiable** if $P(x; \theta_1) = P(x; \theta_2)$ implies $\theta_1 = \theta_2$.

- **Rule 1: Insertion or deletion of condition $\underline{a}. = a.$, in the presence of do-operator conditions.**

  If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ (i.e., $H(\underline{b}. : \underline{a}.|\underline{r}., \underline{s}.) = 0$) in $G_h = \mathcal{D}_{\underline{r}.}G$, then
  $P_{G_e}(b.|a., \mathcal{D}\underline{r}. = r., s.) = P_{G_e}(b.|\mathcal{D}\underline{r}. = r., s.)$

- **Rule 2: Exchange of $\underline{a}. = a.$ and $\mathcal{D}\underline{a}. = a.$ conditions.**

  If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ (i.e., $H(\underline{b}. : \underline{a}.|\underline{r}., \underline{s}.) = 0$) in $G_h = \mathcal{L}_{\underline{a}.}\mathcal{D}_{\underline{r}.}G$, then
  $P_{G_e}(b.|\mathcal{D}\underline{a}. = a., \mathcal{D}\underline{r}. = r., s.) = P_{G_{e'}}(b.|a., \mathcal{D}\underline{r}. = r., s.)$

- **Rule 3: Insertion and deletion of $\mathcal{D}\underline{a}. = a.$ condition.**

  If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ (i.e., $H(\underline{b}. : \underline{a}.|\underline{r}., \underline{s}.) = 0$) in $G_h = \mathcal{D}_{\underline{a}.-an(\underline{s}.)}\mathcal{D}_{\underline{r}.}G$, then
  $P_{G_e}(b.|\mathcal{D}\underline{a}. = a., \mathcal{D}\underline{r}. = r., s.) = P_{G_e}(b.|\mathcal{D}\underline{r}. = r., s.)$

Fig.22.3 shows a finite state machine for the 3 rules of Do Calculus.[4]

Fig.22.4 shows a pictorial representation of some of the nodes (i.e., $\underline{a}., \underline{b}., \underline{r}., \underline{s}.$) of the hypothesis graphs for the 3 rules of Do Calculus.

The 3 rules of DC are 3 separate theorems that have been proven, by Pearl and collaborators, to be correct, consistent, and sufficient for removing all do operator conditions from any expression for which it is possible to do so.

[4]Finite State Machines are discussed in Chapter 30.

Figure 22.3: Finite State Machine describing the 3 rules of Do Calculus. The arrows are commands to replace the string at the source of the arrow by the string at the target of the arrow.

Next we discuss two formulae that can be proven using Do Calculus: the backdoor and the frontdoor adjustment formulae.

The backdoor formula adjusts one multinode and the frontdoor formula adjusts two.

For any two disjoint multinodes $\underline{x}.$ and $\underline{y}.$, we define a **backdoor path** (resp., **frontdoor path**) from $\underline{x}.$ to $\underline{y}.$ as a path from $\underline{x}.$ and $\underline{y}.$ that starts with an arrow pointing into (resp., pointing out of) $\underline{x}.$.

22.2 Parent Adjustment Formula

Suppose that $\underline{x}., \underline{y}., \underline{z}.$ are disjoint multinodes and their union equals the totality of all nodes of a bnet. Suppose we have data available that allows us to estimate $P(x., y., z.)$. Hence, all nodes of the bnet are observed. Furthermore, suppose $\underline{z}. = pa(\underline{x}.)$. In other words, we are considering the bnet

$$\begin{tikzcd} \underline{z}. \arrow[d] \arrow[dr] & \\ \underline{x}. \arrow[r] & \underline{y}. \end{tikzcd} \quad . \tag{22.19}$$

Then

$$P(y., z.|\mathcal{D}\underline{x}. = x.) = P(y.|x., z.)P(z.) \tag{22.20}$$

so

$$P(y.|\mathcal{D}\underline{x}. = x.) = \sum_{z.} P(y.|x., z.)P(z.) \ . \tag{22.21}$$

This is called **adjusting the parents** of $\underline{x}.$.

We say that we are **adjusting or controlling a node** $\underline{a}$ if we condition a probability on $\underline{a}$ and then we average that probability over $\underline{a}$. More generally, we can adjust a whole multinode $\underline{a}.$ jointly.

Rule 1: $G_h = \mathcal{D}_{\underline{r}.}G$:

Rule 2: $G_h = \mathcal{L}_{\underline{a}.}\mathcal{D}_{\underline{r}.}G$:

Rule 3: $G_h = \mathcal{D}_{\underline{a}.-an(\underline{s}.)}\mathcal{D}_{\underline{r}.}G$:

Figure 22.4: Pictorial representation of some of the nodes in the hypothesis graphs (G_h) for the 3 rules of Do Calculus. In this picture, yellow nodes are conditioned on. Furthermore $\underline{a}.' = \underline{a}. - an(\underline{s}.)$ and $\underline{a}.'' = \underline{a}. \cap an(\underline{s}.)$. Note that since we are conditioning on $\underline{s}.$, there are non-blocked paths, indicated in red, between $\underline{b}.$ and $\underline{a}.''$ for Rule 3 (or between $\underline{b}.$ and $\underline{a}.$ for Rules 1 and 2). All these paths pass through the conditioned collider $\underline{s}.$. The hypothesis of Rules 1,2,3 is meant to ban paths like those in red.

Next, we will introduce a generalization of this parent adjustment formula called the backdoor adjustment formula. In a backdoor adjustment formula, the adjusted multinode is not necessarily the parents of $\underline{x}.$.

22.3 Backdoor Adjustment Formula

See Chapter 4 for examples of the use of the backdoor adjustment formula. In this section, we shall mainly be concerned with explaining this theorem, not using it.

Suppose that we have access to data that allows us to estimate a probability distribution $P(x., y., z.)$. Then we say that the backdoor $\underline{z}.$ satisfies the **backdoor adjustment criterion** relative to $(\underline{x}., \underline{y}.)$ if

1. All backdoor paths from $\underline{x}.$ to $\underline{y}.$ are blocked by conditioning on $\underline{z}.$.
2. $\underline{z}. \cap de(\underline{x}.) = \emptyset$.

Claim 43 *(Backdoor Adjustment Formula)*
If $\underline{z}.$ satisfies the backdoor criterion relative to $(\underline{x}., \underline{y}.)$, then

$$P(y.|\mathcal{D}\underline{x}. = x.) = \sum_{z.} P(y.|x., z.)P(z.) \tag{22.22}$$

$$= \tag{22.23}$$

$$\begin{tikzcd} \mathbb{E}z. \arrow[dr] & \\ x. \arrow[r] & y. \end{tikzcd}$$

proof:

$$\begin{tikzcd} \bullet \arrow[r] & \underline{z}. \arrow[dl] \arrow[dr] & \bullet \arrow[l] \arrow[d] \\ \underline{x}. \arrow[u] \arrow[rr] & & \underline{y}. \\ & \bullet \arrow[ul] \arrow[ur] & \end{tikzcd}$$

Figure 22.5: The red arrows in this figure define paths that violate the 2 constraints of the backdoor adjustment criterion. The reason for constraint 2 is that if we condition on node $\underline{z}.$ in accordance with constraint 1, we open the red path from $\underline{x}.$ to $\underline{y}.$ via collider $\underline{z}.$.

Fig.22.5 gives some motivation for the backdoor adjustment criterion. See Claim 47 for a proof of this claim for the special case of the bnet obtained by removing the red arrows from Fig.22.5.

QED

Note that the backdoor adjustment formula can be written as

$$P(y.|\mathcal{D}\underline{x}. = x.) = \sum_{z.} P(y.|x., z.)P(z.) \tag{22.24}$$

$$= \sum_{z.} \frac{P(y., x., z.)}{P(x.|z.)} \tag{22.25}$$

This assumes $P(x.|z.) \neq 0$ for all $x., z.$. This assumption is referred to as **positivity**, and is violated if $P(x.|z.) = \delta(x., x.(z.))$. $P(x.|z.)$ is called the **propensity score** of $x.$ given $z.$. This equation does **inverse probability weighting**. One can approximate $P(x.|z.)$ in this equation to get an approximation to $P(y|\mathcal{D}\underline{x} = x)$.

22.4 Frontdoor Adjustment Formula

See Chapter 32 for examples of the use of the frontdoor adjustment formula. In this section, we shall mainly be concerned with explaining this theorem, not using it.

Suppose that we have access to data that allows us to estimate a probability distribution $P(x., m., y.)$. Hence, the variables $\underline{x}., \underline{m}., \underline{y}.$ are ALL observed (i.e, not hidden). Then we say that the frontdoor $\underline{m}.$ satisfies the **frontdoor adjustment criterion** relative to $(\underline{x}., \underline{y}.)$ if

1. All directed paths from $\underline{x}.$ to $\underline{y}.$ are intercepted by (i.e., have a node in) $\underline{m}.$.
2. All backdoor paths from $\underline{x}.$ to $\underline{m}.$ are blocked.
3. All backdoor paths from on $\underline{m}.$ to $\underline{y}.$ are blocked by conditioning on $\underline{x}.$.

Claim 44 *(Frontdoor Adjustment Formula)*

If $\underline{m}.$ satisfies the frontdoor criterion relative to $(\underline{x}., \underline{y}.)$, and $P(x., m.) > 0$, then

$$P(y.|\mathcal{D}\underline{x}. = x.) = \sum_{m.} \left[\sum_{x'.} P(y.|x'., m.)P(x'.) \right] P(m.|x.) \tag{22.26}$$

$$= \tag{22.27}$$

$$\begin{tikzcd} & \mathbb{E}x'. \arrow[dr] & \\ x. \arrow[r] & \sum m. \arrow[r] & y. \end{tikzcd}$$

Figure 22.6: The red arrows in this figure define paths that violate the 3 constraints of the frontdoor adjustment criterion.

proof:

Fig.22.6 gives some motivation for the frontdoor adjustment criterion. See Claim 48 for a proof of this claim for the special case of the bnet obtained from Fig.22.6 by removing the red arrows. See also Ref.[57] for a proof by Pearl of the Frontdoor Adjustment Formula without using Do Calculus.
QED

22.5 Comparison of Backdoor and Frontdoor adjustment formulae

Define a **direct effect path** for a query $P(y|\mathcal{D}\underline{x} = x, z.)$ as a directed path that starts at $\underline{x}$ and ends at $\underline{y}$. A backdoor path (i.e., one that connects $\underline{x}$ and $\underline{y}$ starting with an arrow pointing into $\underline{x}$), is not a direct effect path; it's an **indirect effect path**.

Note that in the backdoor AF (adjustment formula), we can find a possibly empty observed multinode $\underline{z}.$ such that if we condition on $\underline{z}.$, all indirect effect paths are blocked. In the frontdoor AF, we can't find a multinode $\underline{z}.$ that blocks all indirect effect paths. Despite this, in the frontdoor scenario, the do-query is identifiable and an adjustment formula exists. How is that possible? The frontdoor AF uses the backdoor AF once and then it uses the backdoor AF again, a second time, on the result of the first use. The frontdoor AF replaces a sum over an unobserved node by a sum over an observed one.

22.6 Do operator for DEN bnets

Recall that the structural equations for a linear DEN, as given by Eq.(52.50) of Chapter 52, are:

$$\underline{x} = A\underline{x} + \underline{u} \ . \tag{22.28}$$

Therefore,

$$\underline{x} = (1 - A)^{-1}\underline{u} \tag{22.29}$$

which can be represented for both linear and non-linear DEN diagrams by:

$$\underline{x}_i = x_i(\underline{u}.) \tag{22.30}$$

If now we apply the operator $\mathcal{D}_{\underline{a}=a}$ to the diagram described by the structural equations Eqs.22.28, we get the following new structural equations:

$$\underline{x}_i^* = \begin{cases} \sum_{j<i} A_{i|j}\underline{x}_j^* + \underline{u}_i & \text{if } \underline{x}_i \neq \underline{a} \\ a & \text{if } \underline{x}_i = \underline{a} \end{cases} \ , \tag{22.31}$$

where we are calling $\underline{x}_i^*$ the nodes of the DEN diagram post intervention.

Eqs.(22.31) can be expressed in matrix notation as follows. Define $\pi_{\underline{a}}$ to be the $nx \times nx$ matrix with all entries equal to zero except for the (i_0, i_0) entry, which is 1. And define $e_{\underline{a}}$ to be the column vector with all entries zero except for the i_0'th one, which is 1. Here i_0 is defined so that $\underline{x}_{i_0} = \underline{a}$. In other words, $\pi_{\underline{a}}$ and $e_{\underline{a}}$ are defined by

$$(\pi_{\underline{a}})_{i,j} = \mathbb{1}(i = j, \underline{a} = \underline{x}_i) \tag{22.32}$$

and

$$(e_{\underline{a}})_i = \mathbb{1}(\underline{a} = \underline{x}_i) \ , \tag{22.33}$$

for $i, j \in \{0, 1, \ldots, nx - 1\}$. Next define

$$\pi_{!\underline{a}} = 1 - \pi_{\underline{a}} \ , \tag{22.34}$$

$$A^* = \pi_{!\underline{a}} A \ , \tag{22.35}$$

and

$$\underline{u}_{!\underline{a}} = \pi_{!\underline{a}} \underline{u} \ . \tag{22.36}$$

The effect of pre-multiplying the matrix A and the column vector $\underline{u}$ by $\pi_{!\underline{a}}$ is to leave all rows intact except for the i_0 row, which is set to zero. Here i_0 is defined by $\underline{a} = \underline{x}_{i_0}$. Finally, using all of the variables just defined, we can express the structural equations of the linear DEN bnet, post intervention, as

$$\underline{x}^* = A^*\underline{x}^* + \underline{u}_{!\underline{a}} + ae_{\underline{a}} \ . \tag{22.37}$$

Thus,

$$\underline{x}^* = (1 - A^*)^{-1}(\underline{u}_{!\underline{a}} + a e_{\underline{a}}) \ . \tag{22.38}$$

which can be represented for both linear and non-linear DEN diagrams by:

$$\underline{x}_i^* = x_i^*(\underline{u}_{!\underline{a}}, a) \ . \tag{22.39}$$

For any bnet,

$$P(\underline{y} = y | \underline{x} = x) = P_G(\underline{y} = y | \underline{x} = x) \tag{22.40}$$

$$P(\underline{y} = y | \mathcal{D}\underline{x} = x) = P_{\mathcal{D}_{\underline{x}=x}G}(\underline{y} = y) \tag{22.41}$$

Claim 45 *For a non-linear DEN bnet,*

$$P(y | \mathcal{D}\underline{x} = x) = E\left[\delta[y, y(\underline{u}_{!\underline{x}}, x)]\right] \ . \tag{22.42}$$

proof:

$$\begin{aligned}
P(\underline{y} = y | \mathcal{D}\underline{x} = x) &= P_{\mathcal{D}_{\underline{x}=x}G}(\underline{y} = y) && (22.43)\\
&= \sum_{u_{!\underline{x}}} P(u_{!\underline{x}}) P_{\mathcal{D}_{\underline{x}=x}G}(\underline{y} = y | u_{!\underline{x}}) && (22.44)\\
&= \sum_{u_{!\underline{x}}} P(u_{!\underline{x}}) \delta[y, y(u_{!\underline{x}}, x)] && (22.45)\\
&= E_{\underline{u}_{!\underline{x}}}[\delta[y, y(u_{!\underline{x}}, x)]] && (22.46)\\
&= E[\delta[y, y(\underline{u}_{!\underline{x}}, x)]] && (22.47)
\end{aligned}$$

QED

Claim 46 *For a nonlinear DEN bnet,*

$$E[\underline{y} | \mathcal{D}\underline{x} = x] = E[y(\underline{u}_{!\underline{x}}, x)] \ . \tag{22.48}$$

proof:

$$\begin{aligned}
E[\underline{y} | \mathcal{D}\underline{x} = x] &= \sum_y y P(\underline{y} = y | \mathcal{D}\underline{x} = x) && (22.49)\\
&= \sum_y y E[\delta[y, y(u_{!\underline{x}}, x)]] && (22.50)\\
&= E[y(\underline{u}_{!\underline{x}}, x)] && (22.51)
\end{aligned}$$

QED

For any bnet

$$P(y|\mathcal{D}\underline{x}=x,z) \quad = \quad \frac{P(y,z|\mathcal{D}\underline{x}=x)}{P(z|\mathcal{D}\underline{x}=x)} = P_{\mathcal{D}_{\underline{x}=x}G}(y|x,z) \tag{22.52}$$

For a nonlinear DEN bnet,

$$P(y,z|\mathcal{D}\underline{x}=x) = \sum_{u_{!\underline{x}}} P(u_{!\underline{x}})\delta[y,y(u_{!\underline{x}},x)]\delta[z,z(u_{!\underline{x}},x)] \tag{22.53}$$

$$P(z|\mathcal{D}\underline{x}=x) = \sum_{u_{!\underline{x}}} P(u_{!\underline{x}})\delta[z,z(u_{!\underline{x}},x)] \ . \tag{22.54}$$

Chapter 23

Do Calculus proofs

In Chapter 22, we explained Do Calculus, but referred to this chapter for proofs of claims that use Do Calculus. In this chapter, we've aggregated all proofs, from throughout the book, of claims that use Do Calculus.

Note that even though the 3 rules of Do Calculus are great for proving adjustment formulae for general classes of DAGs, they are sometimes overkill for proving adjustment formulae for a single specific DAG. Indeed, since the 3 rules of Do Calculus are a consequence of the d-separation theorem, it follows that all adjustment formulae should be provable from first principles, assuming only the d-separation theorem and the standard rules of probability theory.

In this chapter, we use the following conventions for bnet diagrams.

Random variables are underlined and their values are not. For example, $\underline{a} = a$ means the random variable $\underline{a}$ takes the value a. A diagram with all its nodes underlined represents a Bayesian Network (bnet), whereas the same diagram with the letters not underlined represents a specific **instantiation** of that bnet. For example $\underline{a} \to \underline{b} \to \underline{c}$ represents the bnet with full probability distribution $P(c|b)P(b|a)P(a)$, whereas $a \to b \to c$ represents $P(c|b)P(b|a)$. Note that, for convenience, we define $a \to b \to c$ to exclude the priors of root nodes such as $P(a)$.

If $\underline{a}$ is a root node, then $\mathbb{E}a$ signifies a weighted sum $\sum_a P(a)$. For example,

$$\mathbb{E}a \to b \to c = \sum_a P(c|b)P(b|a)P(a) \tag{23.1}$$

If $\underline{a}$ is not a root node, then $\sum a$ signifies a simple unweighted sum $\sum_a$. For example,

$$x \to \sum a \to y = \sum_a P(y|a)P(a|x) \tag{23.2}$$

Two bnets are said to be equal if their full probability distributions (i.e., their full instantiations) are equal numerically. For example,

$$\underline{a} \to \underline{b} \to \underline{c} = P(c|b)P(b|a)P(a) = \underline{a} \leftarrow \underline{b} \leftarrow \underline{c} \tag{23.3}$$

Unobserved (a.k.a. hidden, latent) nodes are indicated in a bnet by enclosing their label in a dashed circle. For example, $\overset{\frown}{\underset{\smile}{u}}$. Alternatively, they are indicated by using dashed arrows for all arrows emanating from the unobserved node.

Selection diagrams with switch nodes are discussed in Chapter 105. In a selection diagram with a switch node $\underline{s} \in \{0, 1\}$, if a node $\underline{x}$ has parents $pa(\underline{x})$ where $\underline{s} \notin pa(\underline{x})$, then the TPM of $\underline{x}$ is $P(x|pa(x))$. If, on the other hand, $\underline{x}$ has parents $pa(\underline{x}) = pa^-(\underline{x}) \cup \underline{s}$, where $pa^-(\underline{x}) = pa(\underline{x}) - \underline{s}$, then the TPM of $\underline{x}$ is

$$P(x|pa^-(x), s) = \begin{cases} P(x|pa^-(x)) & \text{if } s = 0 \\ P^*(x|pa^-(x)) & \text{if } s = 1 \end{cases} \tag{23.4}$$

Some **node summation identities** that are used in this chapter:

1.

$$P(y|x_1, x_2) = \sum_a P(y|a, x_1, x_2) P(a|x_1, x_2) \ . \tag{23.5}$$

(23.6)

2.

$$\sum_a P(a|x_1, x_2) = 1 \tag{23.7}$$

(23.8)

A zeroed arrow means the same as no arrow. Eq.(23.8) can be understood as an edge case (when $\underline{y} = \emptyset$) of Eq.(23.6).

3.

$$\sum_a P(x_2|a) P(a|x_1) = P(x_2|x_1) \tag{23.9}$$

(23.10)

4.

$$P(x) = \sum_a P(x|a)P(a) \tag{23.11}$$

$$P(x) \qquad = \qquad \xrightarrow[0]{} \mathbb{E}a \longrightarrow x \tag{23.12}$$

Eq.(23.12) can be understood as an edge case of Eq.(23.10).

An important caveat concerning the above summation identities is that the instantiations that they produce depend on the original bnet whence they come from. This caveat is illustrated in Fig.23.1.

$$\begin{array}{c|c} \sum_a P_G(x_2|a)P_G(a|x_1) = P_G(x_2|x_1) & \sum_a P_{G'}(x_2|a,x_1)P_{G'}(a|x_1) = P_{G'}(x_2|x_1) \\ x_1 \longrightarrow \sum a \longrightarrow x_2 \qquad = \qquad x_1 \longrightarrow x_2 & x_1 \longrightarrow \sum a \longrightarrow x_2 \qquad = \qquad x_1 \longrightarrow x_2 \end{array}$$

Figure 23.1: Note that there are cases when $P_G(x_2|x_1) \neq P_{G'}(x_2|x_1)$. Hence, $P(x_2|x_1)$ depends on the bnet of origin.

We define a **do-adjustment formula** for **do-query** $P(y|\mathcal{D}\underline{x} = x)$ to be an algebraic expression constructed from the TPMs for *observed* (i.e., not hidden) nodes of the original bnet G. If a do-adjustment formula exists for a particular do-query, then we say the do-query is **do-identifiable (DI)**. A **do-transport formula** is a relationship between 2 do-queries. This chapter deals with both do-adjustment and do-transport formulae.

Let G be a graph before amputation of the arrows entering node $\underline{x}$, and let $G_x = \mathcal{D}_{\underline{x}=x}G$ be the same graph after amputation. Also let $P_G() = P()$ be the full probability distribution for graph G, and $P_{G_x}()$ be that for G_x. In general, the following is always true, for bnets with or bnets without hidden nodes:[1]

$$P(y|\mathcal{D}\underline{x} = x) = P_{G_x}(y|x) \tag{23.13}$$

However, the right hand side of this equation is not a valid adjustment formula for this query because it's not expressed in terms of observed TPMs of G.

Note that it's always true that $P(y|\mathcal{D}\underline{x} = x) = P(y|x)$. However, if the bnet contains more nodes than just $\underline{x}$ and $\underline{y}$, then the bnet is identifiable iff it can be stratified (i.e., expressed as $P(y|x) = \sum_{\underline{z}.} P(y|x, \underline{z}.)P(\underline{z}.)$ for some observed multinode $\underline{z}.$). Thus, an alternative name for identifiability is **stratifiability**. Note also that there mightt be more than one adjustment formula, but they must all be numerically equal to $P(y|x)$ for all x and y.

[1]Note that $P_{G_x}(y|x) \neq P_G(y|x) = P(y|x)$. In fact, $P_{G_x}(y|x) = P(y|x)$ iff there is no confounding, so $P_{G_x}(y|x) \neq P(y|x)$ indicates confounding.

The following 3 step heuristic technique appears to be equivalent to Pearl's 3 rules of Do Calculus.[2]

1. Write down a bnet instantiation that has a DAG structure identical to the DAG structure of G, except that arrows entering node $\underline{x}$ have been amputated. All nodes of that instantiation, except nodes x and y, are summed (indicated by $\sum$) or averaged (indicated by $\mathbb{E}$) over.

2. Remove all sums $\sum h$ over hidden nodes $\underline{h}$. This is justified by the node summation identities.

3. Replace every average $\mathbb{E}h$ over a hidden node $\underline{h}$ with an $\mathbb{E}a$, where $\underline{a}$ is any observed node. This step does not always lead to a valid adjustment formula. If it doesn't, we declare the query not-identifiable.

$(a) P(y|\mathcal{D}\underline{x} = x)$ is DI $\qquad$ $(b) P(y|\mathcal{D}\underline{x} = x)$ is not-DI

Figure 23.2: Examples of bnets for which the do-query $P(y|\mathcal{D}\underline{x} = x)$ is DI and not-DI.

See Fig.23.2 for some simple examples of bnets for which the do-query $P(y|\mathcal{D}\underline{x} = x)$ is DI and not-DI. Using our heuristic technique, we can easily see why the query $P(y|\mathcal{D}\underline{x} = x)$ is DI for bnet (a) and not-DI for bnet (b) of Fig.23.2.

For bnet (a), after amputating arrow $z \to x$ and averaging over node z, we get

$$P(y|\mathcal{D}\underline{x} = x) \quad = \quad \mathbb{E}z \tag{23.14}$$

The right hand side of Eq.(23.14) is a valid adjustment formula because it's expressed in terms of observed TPMs of G.

For bnet (b), if we amputate arrow $z \to x$ and average over node z, we get

[2]Three rules to match his three :) This heuristic technique was invented solely by me. I haven't proven that it is equivalent to Pearl's 3 rules of Do Calculus. I believe it is, but it might not be. It correctly produces the backdoor and frontdoor adjustment formulae.

$$P(y|\mathcal{D}\underline{x}=x) \quad = \quad \begin{tikzcd} \mathbb{E}z \arrow[dr] & \\ x \arrow[r] & y \end{tikzcd} \tag{23.15}$$

The right hand side of Eq.(23.15) is not a valid adjustment formula because if we replace $\mathbb{E}z$ by $\mathbb{E}x'$, we get $\sum_{x'} P(y|x,x')P(x')$, and $P(y|x,x')$ is undefined.

Claim 47 *(Backdoor Adjustment Formula)*

If $\begin{tikzcd} \underline{z} \arrow[d] \arrow[dr] & \\ \underline{x} \arrow[r] & \underline{y} \end{tikzcd}$ *then*

$$P(y|\mathcal{D}\underline{x}=x) \quad = \quad \sum_z P(y|x,z)P(z) \tag{23.16}$$

$$= \quad \begin{tikzcd} \mathbb{E}z \arrow[dr] & \\ x \arrow[r] & y \end{tikzcd} \tag{23.17}$$

proof:

*** proof 1:**

$$P(y|\mathcal{D}\underline{x}=x) = \begin{tikzcd} \mathbb{E}z \arrow[dr] & \\ x \arrow[r] & y \end{tikzcd} \tag{23.18}$$

*** proof 2:**

$P(y|\mathcal{D}\underline{x}=x) = \sum_z P(y|\mathcal{D}\underline{x}=x,z)P(z|\mathcal{D}\underline{x}=x)$

by Probability Axioms

$= \sum_z P(y|x,z)P(z|\mathcal{D}\underline{x}=x)$

$P(y|\mathcal{D}\underline{x}=x,z) \to P(y|x,z)$

by Rule 2: If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ in $G_h = \mathcal{L}_{\underline{a}.}\mathcal{D}_{\underline{r}.}G$, then $\mathcal{D}\underline{a}. \rightleftarrows \underline{a}.$

$\underline{y} \perp \underline{x}|\underline{z}$ in $\mathcal{L}_{\underline{x}}\mathcal{D}_{\emptyset}G$: $\begin{tikzcd} \underline{z} \arrow[d] \arrow[dr] & \\ \underline{x} & \underline{y} \end{tikzcd}$

$= \sum_z P(y|x,z)P(z)$

$P(z|\mathcal{D}\underline{x}=x) \to P(z)$

by Rule 3: If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ in $G_h = \mathcal{D}_{\underline{a}.-an(\underline{s}.)}\mathcal{D}_{\underline{r}.}G$, then $\mathcal{D}\underline{a}. \rightleftarrows 1$

$\underline{z} \perp \underline{x}$ in $\mathcal{D}_{\underline{x}}\mathcal{D}_{\emptyset}G$:

$$\begin{tikzcd} \underline{z} \arrow[dr] & \\ \underline{x} \arrow[r] & \underline{y} \end{tikzcd}$$

QED

Claim 48 *(Frontdoor Adjustment Formula)*
If

$$\begin{tikzcd} & \underline{c} \arrow[dl] \arrow[dr] & \\ \underline{x} \arrow[r] & \underline{m} \arrow[r] & \underline{y} \end{tikzcd}$$

then

$$P(y|\mathcal{D}\underline{x} = x) = \sum_m \left[\sum_{x'} P(y|x', m) P(x') \right] P(m|x) \tag{23.19}$$

$$= \begin{tikzcd} & \mathbb{E}x' \arrow[dr] & \\ x \arrow[r] & \sum m \arrow[r] & y \end{tikzcd} \tag{23.20}$$

Using the node summation identities, the $\sum m.$ may be removed from the final instantiation, leaving a result similar to the backdoor adjustment formula, but it is usually left explicit.

proof:
*** proof 1:**

$$P(y|\mathcal{D}\underline{x} = x, c) = \begin{tikzcd} & c \arrow[dr] & \\ x \arrow[r] & \sum m \arrow[r] & y \end{tikzcd} \tag{23.21}$$

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} \mathbb{E}x' \arrow[r] & \sum c \arrow[dr] & \\ x \arrow[r] & \sum m \arrow[r] & y \end{tikzcd} \tag{23.22}$$

$$= \begin{tikzcd} & \mathbb{E}x' \arrow[dr] & \\ x \arrow[r] & \sum m \arrow[r] & y \end{tikzcd} \tag{23.23}$$

*** proof 2:**

$P(y|\mathcal{D}\underline{x}=x) = \sum_m P(y|\mathcal{D}\underline{x}=x,m)P(m|\mathcal{D}\underline{x}=x)$

by Probability Axioms

$= \sum_m P(y|\mathcal{D}\underline{x}=x,\mathcal{D}\underline{m}=m)P(m|\mathcal{D}\underline{x}=x)$

$P(y|\mathcal{D}\underline{x}=x,m) \to P(y|\mathcal{D}\underline{x}=x,\mathcal{D}m=m)$

by Rule 2: If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ in $G_h = \mathcal{L}_{\underline{a}.}\mathcal{D}_{\underline{r}.}G$, then $\mathcal{D}\underline{a}. \rightleftarrows \underline{a}.$

$\underline{y} \perp \underline{m}|\underline{x}$ in $\mathcal{L}_{\underline{m}}\mathcal{D}_{\underline{x}}G$:

$= \sum_m P(y|\mathcal{D}\underline{x}=x,\mathcal{D}\underline{m}=m)P(m|x)$

$P(m|\mathcal{D}\underline{x}=x) \to P(m|x)$

by Rule 2: If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ in $G_h = \mathcal{L}_{\underline{a}.}\mathcal{D}_{\underline{r}.}G$, then $\mathcal{D}\underline{a}. \rightleftarrows \underline{a}.$

$\underline{m} \perp \underline{x}$ in $\mathcal{L}_{\underline{x}}\mathcal{D}_{\emptyset}G$:

$= \sum_m P(y|\mathcal{D}\underline{m}=m)P(m|x)$

$P(y|\mathcal{D}\underline{x}=x,\mathcal{D}\underline{m}=m) \to P(y|\mathcal{D}\underline{m}=m)$

by Rule 3: If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ in $G_h = \mathcal{D}_{\underline{a}.-an(\underline{s}.)}\mathcal{D}_{\underline{r}.}G$, then $\mathcal{D}\underline{a}. \rightleftarrows 1$

$\underline{y} \perp \underline{x}|\underline{m}$ in $\mathcal{D}_{\underline{x}}\mathcal{D}_{\underline{m}}G$:

$= \sum_{x'} \sum_m P(y|\mathcal{D}\underline{m}=m,x')P(x'|\mathcal{D}\underline{m}=m)P(m|x)$

by Probability Axioms

$= \sum_{x'} \sum_m P(y|m,x')P(x'|\mathcal{D}\underline{m}=m)P(m|x)$

$P(y|\mathcal{D}\underline{m}=m,x') \to \quad P(y|m,x')$

by Rule 2: If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ in $G_h = \mathcal{L}_{\underline{a}.}\mathcal{D}_{\underline{r}.}G$, then $\mathcal{D}\underline{a}. \rightleftarrows \underline{a}.$

$\underline{y} \perp \underline{m}|\underline{x}$ in $\mathcal{L}_{\underline{m}}\mathcal{D}_{\emptyset}G$:

$= \sum_{x'} \sum_m P(y|m,x')P(x')P(m|x)$

$P(x'|\mathcal{D}\underline{m}=m) \to P(x')$

by Rule 3: If $(\underline{b}. \perp \underline{a}.|\underline{r}., \underline{s}.)$ in $G_h = \mathcal{D}_{\underline{a}.-an(\underline{s}.)}\mathcal{D}_{\underline{r}.}G$, then $\mathcal{D}\underline{a}. \rightleftarrows 1$

$\underline{x} \perp \underline{m}$ in $\mathcal{D}_{\underline{m}}\mathcal{D}_{\emptyset}G$:

QED

Claim 49 *(Napkin problem from Ref.[66])*

If *then*

$$\begin{tikzcd}
 & \underline{u}_1 \arrow[dl] \arrow[ddr] & & \\
 & \underline{u}_2 \arrow[dl] \arrow[dr] & & \\
\underline{w} \arrow[r] & \underline{z} \arrow[r] & \underline{x} \arrow[r] & \underline{y}
\end{tikzcd}$$

$$P(y|\mathcal{D}\underline{x} = x) = P(y|x) \tag{23.24}$$

$$= \begin{tikzcd}
\mathbb{E}z \arrow[dr] & \\
x \arrow[r] & y
\end{tikzcd} \tag{23.25}$$

proof:

$$P(y,z|\mathcal{D}\underline{x}=x,u_1,u_2) = \tag{23.26}$$

$$P(y|\mathcal{D}\underline{x}=x) = \tag{23.27}$$

$$= \tag{23.28}$$

$$= \tag{23.29}$$

QED

Claim 50 *(from Ref.[66])*

If

then

$$P(y|\mathcal{D}\underline{z}=z,x) = \sum_w P(y|z,x,w)P(w) \tag{23.30}$$

$$= \tag{23.31}$$

proof:

$$P(y,w|\mathcal{D}\underline{z}=z,x) \quad = \tag{23.32}$$

$$P(y|\mathcal{D}\underline{z}=z,x) \quad = \tag{23.33}$$

$$= \tag{23.34}$$

$$= \tag{23.35}$$

QED

Claim 51 *(Trivial Memoryless Transportability, from Ref.[64])*
If *where* $\underline{s} \in \{0,1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x}=x,z) = P^*(y|x,z) \quad \textit{(replace } \mathcal{D} \textit{ by } 1\textit{, keep } P^*\textit{)} \tag{23.36}$$

$$\tag{23.37}$$

proof:

$$P(y|\mathcal{D}\underline{x} = x, z, \underline{s} = 1) = P(y|x, z, \underline{s} = 1)$$

$$\begin{tikzcd} & z \arrow[dr] & \underline{s} = 1 \arrow[d] \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} \quad = \quad \begin{tikzcd} & z \arrow[dr] & \underline{s} = 1 \arrow[d] \\ x \arrow[rr] & & y \end{tikzcd}$$

QED

Claim 52 *(Direct Transportability, a.k.a. External Validity, from Ref.[64])*
If $\begin{tikzcd} \underline{s} \arrow[r] \arrow[dd, dashed, bend right] & \underline{z} \arrow[dl] \arrow[dr] & \\ & & \\ \underline{x} \arrow[rr] & & \underline{y} \arrow[uu, dashed, bend left] \end{tikzcd}$ *where* $\underline{s} \in \{0, 1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x} = x, z) = P(y|\mathcal{D}\underline{x} = x, z) \quad \textit{(replace } P^* \textit{ by } P\textit{, keep } \mathcal{D}\textit{)} \tag{23.38}$$

$$\begin{tikzcd} \underline{s} = 1 \arrow[r] & z \arrow[dr] & \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} \quad = \quad \begin{tikzcd} & z \arrow[dr] & \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} \tag{23.39}$$

Furthermore,

$$P^*(y|\mathcal{D}\underline{x} = x) = \sum_z P(y|\mathcal{D}\underline{x} = x, z) P^*(z) \tag{23.40}$$

$$\begin{tikzcd} \underline{s} = 1 \arrow[drr] & & \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} \quad = \quad \begin{tikzcd} \underline{s} = 1 \arrow[r] & \sum z \arrow[dr] & \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} \tag{23.41}$$

proof:

$$P(y|\mathcal{D}\underline{x} = x, z, \underline{s} = 1) = P(y|\mathcal{D}\underline{x} = x, z)$$

$$\begin{tikzcd} \underline{s} = 1 \arrow[r] & z \arrow[dr] & \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} \quad = \quad \begin{tikzcd} & z \arrow[dr] & \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd}$$

Because $\underline{s} \perp \underline{y}|\underline{z}$

Furthermore,

$$P(y|\mathcal{D}\underline{x} = x, \underline{s} = 1) = \textstyle\sum_z P(y|\mathcal{D}\underline{x} = x, z) P(z|\underline{s} = 1)$$

$$\begin{tikzcd}
\underline{s} = 1 \arrow[r] & \sum z \arrow[dr] & \\
\mathcal{D}\underline{x} = x \arrow[rr] & & y
\end{tikzcd}$$

QED

Claim 53 *(S-Admissible Transportability, from Ref.[64])*

If $\begin{tikzcd}
\underline{s} \arrow[r] & \underline{z} \arrow[r] \arrow[d] & \underline{a} \arrow[d] \\
\underline{x} \arrow[rr] \arrow[ur, dashed, bend left] & & \underline{y} \arrow[ul, dashed, bend right]
\end{tikzcd}$ *where* $\underline{s} \in \{0, 1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x} = x) = \sum_a P(y|\mathcal{D}\underline{x} = x, a) P^*(a) \tag{23.42}$$

$$\begin{tikzcd}
\underline{s} = 1 \arrow[dr] & \\
\mathcal{D}\underline{x} = x \arrow[r] & y
\end{tikzcd} = \begin{tikzcd}
\underline{s} = 1 \arrow[r] & \sum a \arrow[d] \\
\mathcal{D}\underline{x} = x \arrow[r] & y
\end{tikzcd} \tag{23.43}$$

proof:

$$P(y|\mathcal{D}\underline{x} = x, \underline{s} = 1) = \textstyle\sum_a P(y|\mathcal{D}\underline{x} = x, a) P(a|\underline{s} = 1)$$

$$\begin{tikzcd}
\underline{s} = 1 \arrow[r] & \sum z \arrow[r] & \sum a \arrow[d] \\
\mathcal{D}\underline{x} = x \arrow[rr] & & y
\end{tikzcd} = \begin{tikzcd}
\underline{s} = 1 \arrow[r] & \sum a \arrow[d] \\
\mathcal{D}\underline{x} = x \arrow[r] & y
\end{tikzcd}$$

QED

Claim 54 *(Non-transportability, from Ref.[64])*

If $\begin{tikzcd}
& \underline{h} \arrow[dl] \arrow[dr] & \underline{s} \arrow[d] \\
\underline{x} \arrow[rr] & & \underline{y}
\end{tikzcd}$ *where* $\underline{s} \in \{0, 1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x} = x) = P^*(y|\mathcal{D}\underline{x} = x) \tag{23.44}$$

$$\begin{tikzcd}
& \underline{s} = 1 \arrow[d] \\
\mathcal{D}\underline{x} = x \arrow[r] & y
\end{tikzcd} = same \tag{23.45}$$

proof:

$$P^*(y|\mathcal{D}\underline{x}=x) = P^*(y|\mathcal{D}\underline{x}=x)$$

$$\begin{tikzcd}
 & \mathbb{E}h \arrow[dr] & \underline{s}=1 \arrow[d] & & & \underline{s}=1 \arrow[d] \\
\mathcal{D}\underline{x}=x \arrow[rr] & & y & = & \mathcal{D}\underline{x}=x \arrow[rr] & & y
\end{tikzcd}$$

Can't replace $\mathcal{D}\underline{x}=x$ by x because $\underline{y} \not\perp \underline{x}$ in $\mathcal{L}_{\underline{x}}G$. Hence, Rule 2 not satisfied.
QED

Claim 55 *(from Ref.[64])*

If $\begin{tikzcd}
\underline{s} \arrow[rr, bend left] \arrow[dr, bend right] & & \underline{h} \arrow[ll, dashed, bend right] \arrow[r] \arrow[dl] \arrow[dr] & \underline{z} \\
\underline{x} \arrow[rr] & & \underline{y} \arrow[ll, dashed, bend left] &
\end{tikzcd}$ *where* $\underline{s} \in \{0,1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x}=x) = P(y|\mathcal{D}\underline{x}=x) \tag{23.46}$$

$$\begin{tikzcd}
 & \underline{s}=1 \arrow[dr] & & & \\
\mathcal{D}\underline{x}=x \arrow[rr] & & y & = & \mathcal{D}\underline{x}=x \arrow[rr] & & y
\end{tikzcd} \tag{23.47}$$

proof:

$$P(y|\mathcal{D}\underline{x}=x, \underline{s}=1) = \textstyle\sum_h P(y|\mathcal{D}\underline{x}=x,h)P(h)$$

$$\begin{tikzcd}
 & \underline{s}=1 \arrow[rr, bend left] & \mathbb{E}h \arrow[r] \arrow[dr] & \sum z & & & \mathbb{E}h \arrow[dr] & \\
\mathcal{D}\underline{x}=x \arrow[rrr] & & & y & = & \mathcal{D}\underline{x}=x \arrow[rrr] & & y
\end{tikzcd}$$

$$= P(y|\mathcal{D}\underline{x}=x)$$

$$= \mathcal{D}\underline{x}=x \longrightarrow y$$

QED

Claim 56 *(from Ref.[64])*

If $\begin{tikzcd}
\underline{s} \arrow[rr] \arrow[dr] & & \underline{h} \arrow[ll, dashed, bend right] \arrow[dl] \arrow[dr] & \\
\underline{x} \arrow[rr] & & \underline{z} \arrow[r] & \underline{y} \arrow[ll, dashed, bend left]
\end{tikzcd}$ *where* $\underline{s} \in \{0,1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x}=x) = \sum_z P(y|\mathcal{D}\underline{x}=x,z)P^*(z|x) \tag{23.48}$$

$$\begin{tikzcd}
\underline{s}=1 \arrow[dr] & \mathcal{D}\underline{x}=x \arrow[d] & & \underline{s}=1 \arrow[dr] & & \mathcal{D}\underline{x}=x \arrow[d] \\
 & y & = & x \arrow[r] & \sum z \arrow[r] & y
\end{tikzcd} \tag{23.49}$$

proof:

$P(y|\mathcal{D}\underline{x}=x,\underline{s}=1) = \sum_h \sum_z P(y|h,z)P(h)P(z|\mathcal{D}\underline{x}=x,\underline{s}=1)$

$$\begin{tikzcd}
\underline{s}=1 \arrow[dr] & \mathbb{E}\underline{h} \arrow[dr] & \\
\mathcal{D}\underline{x}=x \arrow[r] & \sum z \arrow[r] & y
\end{tikzcd}$$

$= \sum_h \sum_z P(y|h,z)P(h|\mathcal{D}\underline{x}=x)P(z|x,\underline{s}=1)$

$$\begin{tikzcd}
 & \underline{s}=1 \arrow[dr] & \sum \underline{h} \arrow[dr] & \mathcal{D}\underline{x}=x \arrow[l] & \\
= & x \arrow[r] & \sum z \arrow[r] & y &
\end{tikzcd}$$

$= \sum_z P(y|\mathcal{D}\underline{x}=x,z)P(z|x,\underline{s}=1)$

$$\begin{tikzcd}
 & \underline{s}=1 \arrow[dr] & & \mathcal{D}\underline{x}=x \arrow[d] \\
= & x \arrow[r] & \sum z \arrow[r] & y
\end{tikzcd}$$

QED

Claim 57 *(Unconfounded Mediation, from Ref.[63])*
If *then*

$$\begin{tikzcd}
 & \underline{m} \arrow[dr] & \\
\underline{d} \arrow[ur] \arrow[rr] & & \underline{y}
\end{tikzcd}$$

$$P(y|\mathcal{D}\underline{d}=d,\mathcal{I}_{\underline{m}}\underline{d}=d') = \sum_m P(y|d,m)P(m|d') \tag{23.50}$$

$$\begin{tikzcd}
\mathcal{I}\underline{d}=d' \arrow[dr] & & \mathcal{I}\underline{d}=d' \arrow[r] & \sum m \arrow[dr] & \\
 & y & = & \mathcal{D}\underline{d}=d \arrow[rr] & & y
\end{tikzcd} \tag{23.51}$$

proof:

$$P(y|\mathcal{D}\underline{d} = d, \mathcal{I}\underline{d} = d') = \sum_m P(y|d,m)P(m|d')$$

$$\begin{tikzcd}
\mathcal{I}\underline{d} = d' \arrow[r] & \sum m \arrow[dr] & \\
\mathcal{D}\underline{d} = d \arrow[rr] & & y
\end{tikzcd}$$

QED

Claim 58 *(Mediation with universal prior $\underline{\xi}$ and universal confounder $\underline{u}$, from Ref.[63])*

If

$$\begin{tikzcd}
& \underline{u} \arrow[dl] \arrow[d] \arrow[dr] \arrow[ddl] & \\
\underline{\xi} \arrow[r] \arrow[d] \arrow[drr] & \underline{m} \arrow[dr] & \\
\underline{d} \arrow[rr] \arrow[ur] & & \underline{y}
\end{tikzcd}$$

then

$$P(y|\mathcal{D}\underline{d} = d, \mathcal{I}_{\underline{m}}\underline{d} = d') = \sum_{\xi} \sum_{m} P(y|d,m,\xi)P(m|d',\xi)P(\xi) \tag{23.52}$$

$$\begin{tikzcd}
\mathcal{I}\underline{d} = d' \arrow[dr] & & \\
\mathcal{D}\underline{d} = d \arrow[r] & y
\end{tikzcd}
=
\begin{tikzcd}
\mathcal{I}\underline{d} = d' \arrow[rr, bend right] & & \mathbb{E}\xi \arrow[r] \arrow[ddr] & \sum m \arrow[dr] & \\
\mathcal{D}\underline{d} = d \arrow[rrr] & & & & y
\end{tikzcd} \tag{23.53}$$

proof:

$$P(y|\mathcal{D}\underline{d} = d, \mathcal{I}\underline{d} = d') = \sum_{\xi,u} \sum_m P(y|d,m,\xi,u)P(m|d',\xi,u) \underbrace{P(\xi|u)P(u)}_{P(\xi,u)}$$

$$\begin{tikzcd}
& & \mathbb{E}u \arrow[dl] \arrow[d] \arrow[ddr] & & \\
\mathcal{I}\underline{d} = d' \arrow[rrr, bend right] & \sum \xi \arrow[r] \arrow[ddr] & \sum m \arrow[dr] & & \\
\mathcal{D}\underline{d} = d \arrow[rrr] & & & & y
\end{tikzcd}$$

$$= \sum_{\xi} \sum_m P(y|d,m,\xi)P(m|d',\xi)P(\xi)$$

$$
\begin{tikzcd}
\mathcal{I}\underline{d} = d' \arrow[rr, bend right] & & \mathbb{E}\xi \arrow[r] \arrow[dr] & \sum m \arrow[d] \\
= & \mathcal{D}\underline{d} = d \arrow[rrr] & & & y
\end{tikzcd}
$$

We switch from averaging over the prior of ξ, u to averaging over the prior of ξ.

QED

Claim 59 *(Sequential backdoor (SBD) adjustment formula, from Ref.[67])*

If

$$
\begin{tikzcd}
\underline{z}_0 \arrow[r] \arrow[rr, bend left] \arrow[d] \arrow[drrr] & \underline{z}_1 \arrow[r] \arrow[d] & \underline{z}_2 \arrow[d] \arrow[dr] & \\
\underline{x}_0 \arrow[r] \arrow[rr, bend right] \arrow[urr] \arrow[urrr] & \underline{x}_1 \arrow[r] \arrow[ur] \arrow[urr] & \underline{x}_2 \arrow[ur] \arrow[ur] & \underline{y}
\end{tikzcd}
$$

then

$$P(y|\mathcal{D}\underline{x}^3 = x^3) = \mathcal{Q}(y|x^3) \tag{23.54}$$

$$
\begin{tikzcd}
\mathbb{E}z_0 \arrow[r] \arrow[rr, bend left] \arrow[drrr] & \sum z_1 \arrow[r] \arrow[drr] & \sum z_2 \arrow[dr] & \\
\mathcal{D}\underline{x}^3 = x^3 \arrow[r] & y \arrow[r, phantom, "="] & & y \\
x_0 \arrow[uur] \arrow[uurr] \arrow[urrr] & x_1 \arrow[uur] \arrow[urr] & x_2 \arrow[ur] &
\end{tikzcd}
\tag{23.55}
$$

The result shown here for $n = 3$ is true for any integer $n \geq 1$.

proof:

$P(y|\mathcal{D}\underline{x}^3 = x^3) = \mathcal{Q}(y|x^3)$

$$
\begin{tikzcd}
\mathbb{E}z_0 \arrow[r] \arrow[rr, bend left] \arrow[drrr] & \sum z_1 \arrow[r] \arrow[drr] & \sum z_2 \arrow[dr] & \\
\mathcal{D}\underline{x}_0 = x_0 \arrow[uur] \arrow[uurr] \arrow[urrr] & \mathcal{D}\underline{x}_1 = x_1 \arrow[uur] \arrow[urr] & \mathcal{D}\underline{x}_2 = x_2 \arrow[ur] & y
\end{tikzcd}
=
\begin{tikzcd}
\mathbb{E}z_0 \arrow[r] \arrow[rr, bend left] \arrow[drrr] & \sum z_1 \arrow[r] \arrow[drr] & \sum z_2 \arrow[dr] & \\
x_0 \arrow[uur] \arrow[uurr] \arrow[urrr] & x_1 \arrow[uur] \arrow[urr] & x_2 \arrow[ur] & y
\end{tikzcd}
$$

We can replace $\mathcal{D}\underline{x}_i = x_i$ by x_i once all nodes in bnet are observed nodes.

QED

Claim 60 *(Selection Bias (SB) Backdoor Adjustment Formula, from Ref.[4])*

If $\begin{tikzcd} \underline{s} \arrow[r] \arrow[d] \arrow[rr, bend left] & \underline{z}^{<\underline{x}} \arrow[r] \arrow[d] \arrow[dl] & \underline{z}^{>\underline{x}} \\ \underline{x} \arrow[r] \arrow[ur] & \underline{y} & \end{tikzcd}$ *where* $\underline{s} \in \{0,1\}$ *is a switch node and* $\underline{z} = (\underline{z}^{<\underline{x}}, \underline{z}^{>\underline{x}})$*, then*

$$P(y|\mathcal{D}\underline{x} = x, \underline{s} = 1) = \sum_z P(y|x,z)P(z) = P(y|x) \tag{23.56}$$

$$\begin{tikzcd} \underline{s} = 1 \arrow[dr] & & \mathbb{E}z \arrow[d] & \\ \mathcal{D}\underline{x} = x \arrow[r] & y \quad = & x \arrow[r] & y \quad = \quad x \arrow[r] & y \end{tikzcd} \tag{23.57}$$

proof:

$P(y|\mathcal{D}\underline{x} = x, \underline{s} = 1) = \sum_z P(y|\mathcal{D}\underline{x} = x, z)P(z^{<\underline{x}}|\underline{s} = 1)P(z^{>\underline{x}}|x, z^{<\underline{x}}, \underline{s} = 1)$

$$\begin{tikzcd} \underline{s} = 1 \arrow[r] \arrow[rr, bend left] & \sum z^{<\underline{x}} \arrow[r] \arrow[d] & \sum z^{>\underline{x}} \\ \mathcal{D}\underline{x} = x \arrow[r] \arrow[urr] & y & \end{tikzcd}$$

$= \sum_{z^{<\underline{x}}} P(y|\mathcal{D}\underline{x} = x, z^{<\underline{x}})P(z^{<\underline{x}}|\underline{s} = 1)$

$$= \begin{tikzcd} \underline{s} = 1 \arrow[r] & \sum z^{<\underline{x}} \arrow[d] \\ \mathcal{D}\underline{x} = x \arrow[r] & y \end{tikzcd}$$

$= \sum_z P(y|x,z)P(z|\underline{s} = 1)$

$$= \begin{tikzcd} \underline{s} = 1 \arrow[r] & \sum z \arrow[d] \\ x \arrow[r] & y \end{tikzcd}$$

$\mathcal{D}$ can be removed because there are no sums over unobserved nodes.

$= \sum_z P(y|x,z)P(z)$

$$= \begin{tikzcd} & \mathbb{E}z \arrow[d] \\ x \arrow[r] & y \end{tikzcd}$$

$\underline{s} = 1$ node can be removed because this expression must equal $P(y|x, \underline{s} = 1)$. Furthermore, $\underline{y} \perp \underline{s}|(\underline{x}, \underline{z})$ in the hypothesis bnet. Hence, this expression must also equal $P(y|x)$.

QED

Chapter 24

D-Separation

Before reading this chapter, I recommend that you read Chapter F.

A path γ that isn't a loop can have 3 types of intermediate nodes $\underline{x}$ (an intermediate node of γ is a node in γ that isn't one of the two end nodes). Suppose $\underline{a}, \underline{b} \in \gamma$ are the two neighbors of $\underline{x}$. Then the 3 possible cases are:

1. $\underline{x}$ **is a mediator node:** $(\underline{a} \leftarrow \underline{x} \leftarrow \underline{b})$ or $(\underline{a} \rightarrow \underline{x} \rightarrow \underline{b})$
2. $\underline{x}$ **is a fork node:** $(\underline{a} \leftarrow \underline{x} \rightarrow \underline{b})$
3. $\underline{x}$ **is a collider node:** $(\underline{a} \rightarrow \underline{x} \leftarrow \underline{b})$

We say that a non-loop path γ from $\underline{a}$ to $\underline{b}$ (i.e., with end nodes $\underline{a}, \underline{b}$) is **blocked** by conditioning on a multinode $\underline{Z}.$ if one or more of the following statements is true:

1. There is a node $\underline{x} \in \underline{Z}.$ which is a mediator or a fork of γ.
2. γ contains a collider node $\underline{c}$ and $(\underline{c} \cup de(\underline{c})) \cap \underline{Z}. = \emptyset$ (i.e., neither $\underline{c}$ nor any of the descendants of $\underline{c}$ is contained in $\underline{Z}.$)

This definition of a blocked path[1] is easy to remember if one thinks of the following analogy with pipes carrying water. Think of path γ as if it were a pipe carrying water. Think of the nodes of γ as junctions in the pipe. If $\underline{Z}.$ intersects γ at either a mediator or a fork junction, that acts like a stone that blocks the pipe flow. A collider junction $\underline{c}$ is like a t-joint or hole in the pipe causing a jet of water to leak away from the main flow. Its presence diminishes or totally prevents the main flow of water as long as neither $\underline{c}$ nor any of the descendants of $\underline{c}$ are in $\underline{Z}.$. If, on the other hand, $\underline{c} \in \underline{Z}.$, or $\underline{c}' \in \underline{Z}.$ where $\underline{c}' \in de(\underline{c})$, then the stone produces a complete (in the case of $\underline{c} \in \underline{Z}.$) or a partial (in the case of $\underline{c}' \in \underline{Z}.$) plug of the leak, preventing egress from the main flow.

See Fig.24.1 for some examples of paths that are blocked or not blocked by conditioning on a multinode $\underline{Z}.$.

[1] Note that we speak of blocked paths or info, not of blocked nodes. Nodes are not blocked; rather they are either conditioned upon or not.

Not Blocked (24.1)

Blocked (24.2)

Not Blocked (24.3)

Blocked (24.4)

Blocked (24.5)

Not Blocked (24.6)

Not Blocked (24.7)

Figure 24.1: Examples of paths that are blocked or not blocked by conditioning on a multinode $\underline{Z}.$. Nodes belonging to $\underline{Z}.$ are colored yellow.

Given 3 disjoint multinodes $\underline{A}., \underline{B}., \underline{Z}.$ of a graph G, we write " $\underline{A}. \perp_G \underline{B}.|\underline{Z}.$" or say " $\underline{A}.$ **and** $\underline{B}.$ **are d-separated by** $\underline{Z}.$ **in** G " iff there exists no path γ from $\underline{a} \in \underline{A}.$, to $\underline{b} \in \underline{B}.$ which is not blocked by conditioning on $\underline{Z}.$.[2]

The minimal Markov blanket (see Chapter 54) of a node $\underline{a}$ is the smallest multinode $\underline{Z}.$ such that $\underline{a} \perp_G \underline{b}|\underline{Z}.$ for all $\underline{b} \notin \underline{a} \cup \underline{Z}.$.

We are finally ready to state the d-separation theorem, without proof.

A probability distribution P **is compatible with a DAG** G if P and G have the same random variables, and they can be combined to form a bnet without contradictions; i.e., one can calculate all the TPMs from P and multiply them together to obtain P again.

Claim 61 *(d-separation Theorem)*

Suppose $\underline{A}., \underline{B}., \underline{Z}.$ are disjoint multinodes of a DAG G.

If $\underline{A}. \perp_G \underline{B}.|\underline{Z}.$, then $P(B.|A., Z.) = P(B.|Z.)$ for all $B., A., Z.$, for all P compatible with G.

The full converse of the theorem can also be proven, but we won't be using it in this book.

Often, the right hand side of this theorem is stated as "$\underline{A}. \perp_P \underline{B}.|\underline{Z}.$ for all P". Then the theorem is stated: "If $\underline{A}. \perp_G \underline{B}.|\underline{Z}.$, then $\underline{A}. \perp_P \underline{B}.|\underline{Z}.$ for all P."

Note that the following are equivalent:

- $P(B.|A., Z.) = P(B.|Z.)$ for all $B., A., Z.$.
- $\underline{A}. \perp_P \underline{B}.|\underline{Z}.$

[2] $\underline{Z}.$ are the nodes we are "conditioning on". Unmeasured (i.e., hidden, unobserved) nodes cannot be conditioned on, because that would entail measuring them.

- $H(\underline{A}. : \underline{B}.|\underline{Z}.) = 0$ (see Chapter C for definition of conditional mutual information (CMI))

Extra stuff: mostly only for pure mathematicians

Below, we will use the notation $nde(\underline{a})$ to denote all non-descendants, including $\underline{a}$ itself, of a node $\underline{a}$ in a DAG G; i.e., all nodes of G that are not in $de(\underline{a}) \cup \underline{a}$, where $de(\underline{a})$ is defined in Chapter F.

Given a DAG G, define the following sets of d-separations:[3]

$$DS(G) = \{(\underline{A}. \perp_G \underline{B}. \mid \underline{Z}.) : \ \underline{A}., \underline{B}., \underline{Z}. \text{ are multinodes of } G\} \ . \tag{24.8}$$

$$DS_{min}(G) = \{(\underline{A}. \perp_G nde(\underline{A}.) \mid pa(\underline{A}.)) : \ \underline{A}. \text{ is a multinode of } G\} \ . \tag{24.9}$$

See Chapter 71 for an example where set $DS_{min}(G)$ is calculated for a particular DAG G.

Claim 62 *For all DAGs G, $DS(G) = DS_{min}(G)$.*

Given a probability distribution P, define the following set of conditional independencies:

$$CI(P) = \{(\underline{A}. \perp_P \underline{B}. \mid \underline{Z}.) : \ \underline{A}., \underline{B}., \underline{Z}. \text{ are multinodes of } P\} \ , \tag{24.10}$$

For a DAG G and a probability distribution P compatible with G, define a map ϕ by

$$\phi : DS_{min}(G) \ \rightarrow \ CI(P) \tag{24.11}$$

$$\phi : \underline{A}. \perp_G nde(\underline{A}.) \mid pa(\underline{A}.) \ \mapsto \ \underline{A}. \perp_P nde(\underline{A}.) \mid pa(\underline{A}.) \tag{24.12}$$

In general, this map is 1-1 but not onto.

Claim 63 *For a bnet with a DAG G and a total probability distribution P, the map ϕ is a bijection.*

$DS(G)$ does not fully specify a DAG. DAGs with the same $DS(G)$ are said to be **d-separation equivalent**. See Chapter 71 for more info about d-separation equivalence.

---

[3]Note that $(\underline{A}. \perp_G nde(\underline{A}.) \mid pa(\underline{A}.))$ and $(\underline{A}. \perp_G nde(\underline{A}.) - pa(\underline{A}.) \mid pa(\underline{A}.))$ are equivalent because $H(\underline{a} : \underline{b}, \underline{c}|\underline{c}) = H(\underline{a} : \underline{b}|\underline{c})$.

Chapter 25

D-Separation in Quantum Mechanics

See Ref.[93].

Chapter 26

Dynamical Bayesian Networks

Figure 26.1: Example of a DBN. Same time-slice (in black) is repeated T times. Green arrows connect adjacent slices.

A **dynamical bnet (DBN)** is simply a Markov chain $\underline{a}_0 \to \underline{a}_1 \to \ldots \underline{a}_{T-1}$ (see Chapter 56) for which each node $\underline{a}_i$ is called a **time-slice**. Each time-slice represents at finer resolution a sub-DAG which has the same structure (but not necessarily the same TPMs) in every time-slice.[1] If the TPMs are the same for all time-slices, we call it a **time-homogeneous dynamical bnet**. Fig.26.1 gives an example of a DBN. In that figure, each time-slice is represented in black, and arrows connecting adjacent time-slices are represented in green. In Fig.26.1, we've drawn the 3 nodes of each time-slice vertically, and labeled them with a superscript $.^{(t)}$, where $t \in \{0, 1 \ldots, T-1\}$ is the time of the slice. To fully specify the DBN of Fig.26.1, we would also have to specify the TPMs

$P(c^{(0)})$,
$P(b^{(0)})$,
$P(a^{(0)})$,
$P(c^{(1)}|c^{(0)})$,

[1]Sometimes, it is convenient to define time-slices that are influenced by the n-previous time-slices. instead of just $n = 1$. By defining a bigger time-slice that contains n of the original time-slices, we can get bigger time-slices that only listen to the adjacent previous bigger time-slice.

$P(b^{(1)}|b^{(0)}, a^{(1)})$
$P(a^{(1)}|b^{(0)}, c^{(0)})$, etc.

Dynamical bnets are very common in AI and Data Science. Kalman filters (Chapter 47), Hidden Markov Models (Chapter 40) and Recurrent Neural Networks (Chapter 81) are famous examples of DBNs.

Bnets are acyclic; they can't have cycles (i.e, closed directed paths). Yet feedback loops are an important concept in Science. So what is the equivalent of feedback loops in the bnet world? Dynamical bnets are. Fig.26.2 represents Fig.26.1 more compactly using feedback loops. Any bnet with feedback loops can be "unrolled" into a DBN.

Figure 26.2: Dynamical bnet Fig.26.1 represented more compactly using feedback loops. Dashed green arrows point to the future, from nodes of the t time-slice to nodes of the $t+1$ time-slice.

Chapter 27

Expectation Maximization

This chapter is based on Refs.[133] and [198].

The Expectation Maximization (EM) algorithm is commonly used in Data Science to find the maximum over an **unknown parameter** θ of a likelihood function

$$P(\vec{x}|\theta) = \sum_{\vec{h}} P(\vec{x}, \vec{h}|\theta) \ , \tag{27.1}$$

where $\vec{x}$ denotes the **observed variables**, and $\vec{h}$ denotes the **hidden variables**. Both θ and $\vec{h}$ are hidden (i.e., unobserved).[1]

Figure 27.1: bnet for EM with $nsam = 3$. $x[\sigma] = x^\sigma$ and $h[\sigma] = h^\sigma$.

The bnet for the EM algorithm is given by Fig.27.1 for $nsam = 3$. Later on in this chapter, we will give the node TPMs for this bnet for the special case in which $P(x^\sigma \mid \theta)$ is a mixture (i.e., weighted sum) of Gaussians.

[1]The term "unknown parameter" is mainly of frequentist origin. For Bayesians, θ is a random variable with a delta function prior, whereas for frequentists, it is not a random variable at all, just an unknown parameter with no randomness.

Note that if we erase the $\underline{h}^\sigma$ nodes from Fig.27.1, we get the bnet for naive Bayes, which is used for classification into the states of $\underline{\theta}$. However, there is one big difference. With naive Bayes, the leaf nodes have different TPMs. Here, we will assume they are i.i.d. Naive Bayes is used for classification: i.e., given the states of the leaf nodes, we infer the state of the root node. EM is used for clustering; i.e., given many i.i.d. samples, we fit their distribution by a weighted sum of prob distributions, usually Gaussians.

Let

L =likelihood function.

$nsam$ = number of samples.

$\vec{x} = (x[0], x[1], \ldots, x[nsam-1])$, $x^\sigma = x[\sigma] \in val(\underline{x})$ for all σ.

$\vec{h} = (h[0], h[1], \ldots, h[nsam-1])$, $h^\sigma = h[\sigma] \in val(\underline{h})$ for all σ.

We assume that the samples (x^σ, h^σ) are i.i.d. for different σ at fixed θ. What this means is that there are probability distributions $P_{\underline{x}|\underline{h},\underline{\theta}}$ and $P_{\underline{h}|\underline{\theta}}$ such that

$$P(\vec{x}, \vec{h}|\theta) = \prod_\sigma \left[P_{\underline{x}|\underline{h},\underline{\theta}}(x^\sigma \mid h^\sigma, \theta) P_{\underline{h}|\underline{\theta}}(h^\sigma \mid \theta) \right] \ . \tag{27.2}$$

Definition of likelihood functions:

$$\underbrace{P(\vec{x}|\theta)}_{L(\theta;\vec{x})} \ = \ \sum_{\vec{h}} \underbrace{P(\vec{x}, \vec{h}|\theta)}_{L(\theta;\vec{x},\vec{h})} \tag{27.3}$$

θ^* = maximum likelihood estimate of θ (no prior $P(\theta)$ assumed):

$$\theta^* = \operatorname*{argmax}_\theta L(\theta; \vec{x}) \tag{27.4}$$

27.1 The EM algorithm:

1. **Expectation step:**[2]

$$Q(\theta|\theta^{(t)}) = E_{\vec{h}|\vec{x},\theta^{(t)}} \ln P(\vec{x}, \vec{h}|\theta) \tag{27.5}$$

2. **Maximization step:**

$$\theta^{(t+1)} = \operatorname*{argmax}_\theta Q(\theta|\theta^{(t)}) \ . \tag{27.6}$$

Claim: $\lim_{t\to\infty} \theta^{(t)} = \theta^*$.

[2]Note that
the right hand side of Eq.(27.5) is expressible in the form $\sum_\sigma \sum_{h^\sigma} f(x^\sigma, h^\sigma)$.

Figure 27.2: The EM algo generates a sequence of parameter estimates $(\theta^{(t)})_{t=0,1,2,\ldots}$ that converges to the optimum (i.e., best-fit) parameter θ^*.

Fig.27.2 portrays the recursive nature of the EM algo as a dynamical, recurrent bnet. For Fig.27.2, the TPMs, printed in blue, for the $\underline{\theta}^{(t)}$ nodes for $t = 1, 2, \ldots$, are as follows:

$$P(\theta^{(t+1)}|\vec{x}, \theta^{(t)}) = \delta(\theta^{(t+1)}, \operatorname*{argmax}_{\theta} Q(\theta|\theta^{(t)})) \ . \tag{27.7}$$

27.1.1 Motivation

$$Q(\theta|\theta^{(t)}) = E_{\vec{h}|\vec{x},\theta^{(t)}} \ln P(\vec{x}, \vec{h}|\theta) \tag{27.8}$$

$$= E_{\vec{h}|\vec{x},\theta^{(t)}} [\ln P(\vec{h}|\vec{x}, \theta) + \ln P(\vec{x}|\theta)] \tag{27.9}$$

$$= -D_{KL} \left(P(\vec{h}|\vec{x}, \theta^{(t)}) \parallel P(\vec{h}|\vec{x}, \theta) \right) - H[P(\underline{\vec{h}}|\vec{x}, \theta^{(t)})] + \ln P(\vec{x}|\theta) \tag{27.10}$$

When $\theta^{(t)} = \theta$, this becomes

$$Q(\theta|\theta) = -H[P(\underline{\vec{h}}|\vec{x}, \theta)] + \ln P(\vec{x}|\theta) \ . \tag{27.11}$$

Hence,

$$\partial_\theta Q(\theta|\theta) = -\sum_{\vec{h}} \partial_\theta P(\underline{\vec{h}}|\vec{x}, \theta) + \partial_\theta \ln P(\vec{x}|\theta) \tag{27.12}$$

$$= \partial_\theta \ln P(\vec{x}|\theta) \tag{27.13}$$

So if $\theta^{(t)} \to \theta$ and $Q(\theta|\theta)$ is max at $\theta = \theta^*$, then $\ln P(\vec{x}|\theta)$ is max at $\theta = \theta^*$ too.

For a more rigorous proof that $\lim_{t\to\infty} \theta^{(t)} = \theta^*$, see Wikipedia article Ref.[133] and references therein.

27.2 Minorize-Maximize (MM) algorithms

A function $\mu(\theta|\theta^{(t)})$ **is said to minorize a target function** $LL(\theta)$ iff for all θ at fixed $\theta^{(t)}$, it satisfies the "$\mu \leq LL$ property"

Figure 27.3: Function $\mu(\theta|\theta^{(t)})$ minorizes the function $LL(\theta)$. Note that $\mu(\theta|\theta^{(t)})$ is always below $LL(\theta)$. "max" indicates $\theta^{(t+1)} = \operatorname{argmax}_\theta \mu(\theta|\theta^{(t)})$. "kiss" indicates $\mu(\theta^{(t)}|\theta^{(t)}) = LL(\theta^{(t)})$.

$$\mu(\theta|\theta^{(t)}) \leq LL(\theta) \ , \tag{27.14}$$

and the "$\mu = LL$ property"

$$\mu(\theta^{(t)}|\theta^{(t)}) = LL(\theta^{(t)}) \ . \tag{27.15}$$

We **recursively maximize a minorizing function** $\mu(\theta|\theta^{(t)})$ if we define a sequence $(\theta^{(t)})_{t=0,1,\ldots}$ as follows:

$$\theta^{(t+1)} = \operatorname{argmax}_\theta \mu(\theta|\theta^{(t)}) \ . \tag{27.16}$$

The sequence $(LL(\theta^{(t)}))_{t=0,1,2,\ldots}$ generated by recursively maximizing a minorizing function must be nondecreasing:

$$LL(\theta^{(t+1)}) \geq \mu(\theta^{(t+1)}|\theta^{(t)}) \geq \mu(\theta^{(t)}|\theta^{(t)}) = LL(\theta^{(t)}) \ . \tag{27.17}$$

A **minorize-maximize (MM) algorithm** is any algo that specifies a minorizing function $\mu(\theta|\theta^{(t)})$ for a particular target function $LL(\theta)$. One can also define a **majorize-minimize algo (also called MM)** by inverting the inequalities throughout.

The EM algo is an MM algo. Indeed, if we define

$$LL(\theta) = \ln P(\vec{x}|\theta) \tag{27.18}$$

and

$$\mu(\theta|\theta^{(t)}) = Q(\theta|\theta^{(t)}) + H(P(\vec{\underline{h}}|\vec{x},\theta^{(t)}) \ , \tag{27.19}$$

then Eq.(27.10) establishes the $\mu \leq LL$ and $\mu = LL$ properties required of a minorizing function.

How an MM algo works is portrayed in Fig.27.3.

27.3 Examples

27.3.1 Gaussian mixture

$x^\sigma \in \mathbb{R}^d = val(\underline{x})$. $val(\underline{h})$ discrete and not too large. $n_{\underline{h}} = |val(\underline{h})|$ is number of Gaussians that we are going to fit the samples with.

Let

$$\theta = [w_h, \mu_h, \Sigma_h]_{h \in val(\underline{h})} \ , \tag{27.20}$$

where $[w_h]_{h \in val(\underline{h})}$ is a probability distribution of weights, and where $\mu_h \in \mathbb{R}^d$ and $\Sigma_h \in \mathbb{R}^{d \times d}$ are the mean value vector and covariance matrix of a d-dimensional Gaussian distribution.

The TPMs, printed in blue, for the bnet Fig.27.1, for the special case of a mixture of Gaussians, are as follows:

$$P(x^\sigma \mid h^\sigma \mid \theta) = \mathcal{N}_d(x^\sigma; \mu_{h^\sigma}, \Sigma_{h^\sigma}) \tag{27.21}$$

$$P(h^\sigma \mid \theta) = w_{h^\sigma} \tag{27.22}$$

Note that

$$P(x^\sigma \mid \theta) = \sum_h P(x^\sigma \mid h^\sigma = h, \theta) P(h^\sigma = h \mid \theta) \tag{27.23}$$

$$= \sum_h w_h \mathcal{N}_d(x^\sigma; \mu_h, \Sigma_h) \tag{27.24}$$

$$P(\vec{x}, \vec{h}|\theta) = \prod_\sigma [w_{h^\sigma} \mathcal{N}_d(x^\sigma; \mu_{h^\sigma}, \Sigma_{h^\sigma})] \tag{27.25}$$

$$= \prod_\sigma \prod_h [w_h \mathcal{N}_d(x^\sigma; \mu_h, \Sigma_h)]^{\mathbb{1}(h=h^\sigma)} \tag{27.26}$$

Old Faithful: See Wikipedia Ref.[133] for an animated gif of a classic example of using EM to fit samples with a Gaussian mixture. Unfortunately, could not include

it here because pdflatex does not support animated gifs. The gif shows samples in a 2 dimensional space (eruption time, delay time) from the Old Faithful geyser. In that example, $d = 2$ and $n_{\underline{h}} = 2$. Two clusters of points in a plane are fitted by a mixture of 2 Gaussians.

K-means clustering is often presented as the main competitor to EM for doing **clustering (non-supervised learning)**. In K-means clustering, the sample points are split into K mutually disjoint sets $S_0, S_1, \ldots, S_{K-1}$. The algorithm is easy to describe:

1. Initialize by choosing at random K data points $(\mu_k)_{k=0}^{K-1}$ called means or centroids and placing μ_k in S_k for all k.

2. **STEP 1:** For each data point, add it to the S_k whose centroid μ_k is closest to it.

3. **STEP 2:** Recalculate the centroids. Set μ_k equal to the mean value of set S_k.

4. Repeat steps 1 and 2 until the centroids stop changing by much.

Step 1 is analogous to the expectation step in EM, and Step 2 to the maximization step in EM (θ estimation versus μ_k estimation). We won't say anything further about K-means clustering because it isn't related to bnets in any way, and this is a book about bnets. For more info about K-means clustering, see Ref.[150].

27.3.2 Blood Genotypes and Phenotypes

Notation: $\vec{\underline{a}} = (\underline{a}^\sigma)_{\sigma=0,1,\ldots,nsam-1}$, where $nsam$ is the number of samples.

Suppose $\vec{\underline{x}} = (\vec{\underline{x}}_0)$ (i.e., just one component)
$\vec{\underline{h}} = (\vec{\underline{h}}_0)$ (i.e., just one component)
$\underline{h}^\sigma \in val(\underline{h}) = \{AA, AO, BB, BO, OO, AB\}$ (the 6 blood genotypes)
$\underline{x}^\sigma \in val(\underline{x}) = \{A, B, O, AB\}$ (the 4 blood phenotypes)

$$\begin{tikzcd}
\underline{\theta} \arrow[d] \arrow[dr] & \\
\underline{x}^\sigma & \underline{h}^\sigma \arrow[l]
\end{tikzcd}$$

Figure 27.4: bnet for blood phenotypes x^σ and genotypes h^σ.

For the bnet Fig.27.4, the TPMs, printed in blue, are as follows:

$$P(h^\sigma|\theta) = \begin{array}{c|c} & \\ \hline AA & p_A^2 \\ AO & 2p_Ap_O \\ BB & p_B^2 \\ BO & 2p_Bp_O \\ OO & p_O^2 \\ AB & 2p_Ap_B \end{array} , \tag{27.27}$$

where $p_A + p_B + p_O = 1$.

$$P(x^\sigma \mid h^\sigma, \theta) = \begin{array}{c|cccccc} & AA & AO & BB & BO & OO & AB \\ \hline A & 1 & 1 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 1 & 1 & 0 & 0 \\ O & 0 & 0 & 0 & 0 & 1 & 0 \\ AB & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \tag{27.28}$$

$$\theta = (p_A, p_B) \tag{27.29}$$

Multiplying the TPMs in Eqs.(27.27 and (27.28), we get

$$P(x^\sigma \mid \theta) = \begin{array}{c|c} & \\ \hline A & p_A^2 + 2p_Ap_O(= \pi_A) \\ B & p_B^2 + 2p_Bp_O(= \pi_B) \\ O & p_O^2(= \pi_O) \\ AB & 2p_Ap_B(= \pi_{AB}) \end{array} \tag{27.30}$$

Note that

$$P(\vec{x}|\theta) = \prod_\sigma P(x^\sigma|\theta) \tag{27.31}$$

$$= (\pi_A)^{N_A}(\pi_B)^{N_B}(\pi_O)^{N_O}(\pi_{AB})^{N_{AB}} , \tag{27.32}$$

where N_x for $x \in val(\underline{x}) = \{A, B, O, AB\}$ are the counts from the data. We can get estimates for the parameters p_A and p_B right here without doing EM. Just note that

$$\widehat{\pi}_x = \frac{N_x}{N_+} \tag{27.33}$$

for $x \in val(\underline{x})$, where $N_+ = \sum_x N_x$. Eqs.(27.33) give 4 quadratic equations that can be solved for the parameters p_A, p_B in terms of the observed counts N_x for $x \in val(\underline{x})$.

If, instead, you want to find the optimum parameters p_A, p_B using EM, note that

$$
\begin{aligned}
Q(\theta|\theta^{(t)}) &= \sum_{\vec{h}} P(\vec{h}|\theta^{(t)}) \ln P(\vec{x}, \vec{h}|\theta) && (27.34)\\
&= \sum_{\vec{h}} \left[\prod_{\sigma} P(h^{\sigma}|\theta^{(t)})\right] \ln \left[\prod_{\sigma} P(x^{\sigma}, h^{\sigma}|\theta)\right] && (27.35)\\
&= \sum_{\sigma} \sum_{h^{\sigma}} P(h^{\sigma}|\theta^{(t)}) \ln P(x^{\sigma}, h^{\sigma}|\theta) && (27.36)\\
&= \sum_{\sigma} \sum_{h^{\sigma}} P(h^{\sigma}|\theta^{(t)}) [\ln P(x^{\sigma}|h^{\sigma}, \theta) + \ln P(h^{\sigma}|\theta)] && (27.37)\\
&= nsam \sum_{h^{\sigma}} P(h^{\sigma}|\theta^{(t)}) \ln P(h^{\sigma}|\theta) \ . && (27.38)
\end{aligned}
$$

27.3.3 Missing Data/Imputation

The previous example on blood genotypes and phenotypes assumed no missing data in compiling the counts N_x. But what if there is missing data? Can one still apply the EM algo in that case? Yes! See Chapter 63.

Chapter 28

Factor Analysis

hidden factors, X	observed, Y	weights, W
Staff performance	waiting time	.2
	cleanliness	.4
	staff behavior	.4
Food quality	taste of food	.4
	food temperature	.3
	freashness of food	,3

Table 28.1: Example of Factor Analysis (FA) expressed in tabular form.

This chapter is based on Refs.[135] and [85].

Fig.28.1 is an example of factor analysis (FA) expressed in tabular form.

Let

$\underline{Y}, \Delta\underline{Y}, M, \underline{\mathcal{E}} \in \mathbb{R}^{na\times nc}$

$W \in \mathbb{R}^{na\times nb}$

$\underline{X} \in \mathbb{R}^{nb\times nc}$

$\underline{y}_i, \Delta\underline{y}_i, \mu_i, \underline{\epsilon}_i \in \mathbb{R}^{na\times 1}$ for $i = 1, 2, \ldots nc$ are column vectors of $\underline{Y}, \Delta\underline{Y}, M, \underline{\mathcal{E}}$. The components of y_i will be denoted by $(y_i)_a$ for $a = 1, 2, \ldots, na$.

$\underline{x}_i \in \mathbb{R}^{nb\times 1}$ for $i = 1, 2, \ldots nc$ are column vectors of $\underline{X}$. The components of x_i will be denoted by $(x_i)_b$ for $b = 1, 2, \ldots, nb$.

Suppose

$$\underbrace{\underline{Y} - M}_{\Delta\underline{Y}} = W\underline{X} + \underline{\mathcal{E}} \tag{28.1a}$$

$$\underbrace{\underline{y}_i - \mu_i}_{\Delta\underline{y}_i} = W\underline{x}_i + \underline{\epsilon}_i \tag{28.1b}$$

In FA, the bnet Fig.28.1 is repeated for $i = 1, 2, \ldots, nc$. Henceforth, will make implicit (i.e. omit) the i subscript from the column vectors $\underline{y}_i, \Delta\underline{y}_i, \underline{\mu}_i, , \epsilon_i$. and $\underline{x}_i$.

We will make that index explicit in situations when it's necessary for clarity, as when the index is being summed over.

As explained in Fig.28.2, FA can be viewed as a slight generalization of Naive Bayes (see Chapter 67).

Figure 28.1: In FA, this bnet is repeated for $i = 1, 2, \ldots, nc$.

Naive Bayes Factor Analysis

Figure 28.2: Structurally speaking (disregarding the TPMs of the nodes), if we disregard the noise nodes $\underline{\epsilon}_i$, FA can be viewed as a generalization of Naive Bayes (NB). In NB we have a single root node $\underline{c}$, whereas in FA, we can have multiple root nodes such as $\underline{x}_1$ and $\underline{x}_2$.

Besides the structural equations Eqs.(28.1), FA assumes

1. $P(x, \epsilon) = P(x)P(\epsilon)$ ($\underline{x}$ and $\underline{\epsilon}$ are independent variables)
2. $E[\underline{y}] = \mu$, $\left\langle \underline{x}, \underline{x}^T \right\rangle = 1$, $\underline{\epsilon} \sim \mathcal{N}(0, \sigma I)$

Hence

$$
\begin{aligned}
C_{\underline{y}} &= \left\langle \Delta \underline{y}, \Delta \underline{y}^T \right\rangle && (28.2)\\
&= \left\langle W\underline{x} + \underline{\epsilon}\,, [W\underline{x} + \underline{\epsilon}]^T \right\rangle && (28.3)\\
&= W \left\langle \underline{x}, \underline{x}^T \right\rangle W^T + \left\langle \underline{\epsilon}\,, \underline{\epsilon}^{\,T} \right\rangle && (28.4)\\
&= WW^T + \sigma^2 I && (28.5)
\end{aligned}
$$

In FA, the bnet Fig.28.1 is given the TPMs of an LDEN[1]. Root node $\underline{x}$ can be given either a deterministic or a random TPM. We next give the TPMs, printed in blue, for these two cases:

- LDEN Model with deterministic $\underline{x}$

$$P(\epsilon) = \mathcal{N}(\epsilon; 0, \sigma = \sigma) = \frac{1}{(2\pi\sigma^2)^{na/2}} \exp\left(-\frac{\epsilon^T \epsilon}{2\sigma^2}\right) \tag{28.6}$$

$$P(x) = \delta(x, X) \tag{28.7}$$

$$P(y|x, \epsilon) = \mathbb{1}(y = Wx + \epsilon) \tag{28.8}$$

- LDEN Model with Gaussian $\underline{x}$

$$P(\epsilon) = \mathcal{N}(\epsilon; 0, \sigma = \sigma) = \frac{1}{(2\pi\sigma^2)^{na/2}} \exp\left(-\frac{\epsilon^T \epsilon}{2\sigma^2}\right) \tag{28.9}$$

$$P(x) = \mathcal{N}(x; 0, \sigma = 1) = \frac{1}{(2\pi)^{nb/2}} \exp\left(-\frac{x^T x}{2}\right) \tag{28.10}$$

$$P(y|x, \epsilon) = \mathbb{1}(y = Wx + \epsilon) \tag{28.11}$$

If we assume that $\underline{x}$ has a Gaussian TPM, we get

$$P(y|x) = \sum_{\epsilon} P(y|x, \epsilon) P(\epsilon) \tag{28.12}$$

$$= \frac{1}{(2\pi\sigma^2)^{na/2}} \exp\left(-\frac{1}{2\sigma^2} \parallel y - Wx - \mu \parallel^2\right) \tag{28.13}$$

$$P(y) = \sum_{x} P(y|x) P(x) \tag{28.14}$$

$$= \frac{1}{(2\pi)^{na/2}\sqrt{\det C_{\underline{y}}}} \exp\left(-\frac{1}{2}\Delta y^T C_{\underline{y}}^{-1} \Delta y\right) \tag{28.15}$$

[1]LDEN=Linear Deterministic with External Noise. LDEN models are defined in Chapter 52

Let

$$\mathcal{C} = W^T W + \sigma^2 I \tag{28.16}$$

$$\widehat{\mathcal{C}} = \mathcal{C}/\sigma^2 \tag{28.17}$$

$$\Delta x = x - \mathcal{C}^{-1} W^T \Delta y \tag{28.18}$$

Note that $\mathcal{C}$ is $nb \times nb$ whereas $C_{\underline{y}} = WW^T + \sigma^2 I$ is $na \times na$.

Using Bayes rule, we get

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \tag{28.19}$$

$$= \frac{1}{(2\pi)^{nb/2}} \sqrt{\det \widehat{\mathcal{C}}} \exp\left(-\frac{1}{2} \Delta x^T \widehat{\mathcal{C}} \Delta x\right) \tag{28.20}$$

Let

$$S = \frac{1}{nc} \sum_{i=1}^{nc} \Delta y_i \Delta y_i^T \tag{28.21}$$

Then the log of the posterior probability of $\underline{Y}$ is given by

$$\mathcal{L} = \sum_{i=1}^{nc} \ln P(y_i) \tag{28.22}$$

$$= -\frac{na\ \ nc}{2} \ln(2\pi) - \frac{nc}{2} \ln \det(C_{\underline{y}}) - \frac{nc}{2} \mathrm{tr}(C_{\underline{y}}^{-1} S) \tag{28.23}$$

Chapter 29

Factor Graphs

Suppose $x^{nx} = (x_0, x_1, \ldots x_{nx-1})$. Consider a product

$$g(x^{nx}) = \prod_{\alpha=0}^{nf-1} f_\alpha(x_{A_\alpha}) \tag{29.1}$$

of scalar functions $f_\alpha : \underline{x}_{A_\alpha} \to \mathbb{R}$, where $A_\alpha \subset \{0, 1, \ldots, nx-1\}$. For instance, consider $g : val(\underline{x}_0), \times val(\underline{x}_1) \times val(\underline{x}_2) \to \mathbb{R}$ defined by:

$$g(x_0, x_1, x_2) = f_0(x_0) f_1(x_0, x_1) f_2(x_0, x_1) f_3(x_1, x_2) \ . \tag{29.2}$$

The **factor graph** for this function g is given by Fig.29.1.

Figure 29.1: Factor graph for function g defined by Eq.(29.2).

Figure 29.2: Bipartite bnet corresponding to factor graph Fig.29.1.

A **Markov Random Field (MRF)** is a statistical model whose probability distribution is of the form

$$P(x^{nx}) = \mathcal{N}(!x^{nx}) \prod_{\alpha} f_\alpha(x_{A_\alpha}) \ , \tag{29.3}$$

so it can be represented graphically by a factor graph. The factor functions f_α of a factor graph are called **potentials**.

One can map any factor graph (the "source") to a special bipartite bnet (the "image"), as follows. Replace each x_i by $\underline{x}_i \in val(\underline{x}_i)$ for $i = 0, 1, \ldots, nx-1$ and each f_α by $\underline{f}_\alpha$ for $\alpha = 0, 1, \ldots, nf-1$. Then replace the connections (edges) of the factor graph by arrows from $\underline{x}_i$ to $\underline{f}_\alpha$. [1] For example, Fig.29.2 is the image bipartite bnet of the source factor graph Fig.29.1.

Let $\underline{x}^{nx} = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{nx-1})$ and $\underline{f}^{nf} = (\underline{f}_0, \underline{f}_1, \ldots, \underline{f}_{nf-1})$. Let[2] $f_\alpha \in \{0, 1\}$ for all α, and $y_\alpha = f_\alpha(x_{nb(\underline{f}_\alpha)})$. Here we are using $nb(\underline{f}_\alpha)$ to denote the neighborhood of node $\underline{f}_\alpha$ in the image bipartite bnet, and we are using x_S to denote $(x_i)_{i \in S}$. Without loss of generality, we will assume that $y_\alpha \in [0, 1]$ for all α. Then we define the TPMs, printed in blue, for the image bipartite bnet, as follows.

$$P(f_\alpha | x_{nb(\underline{f}_\alpha)}) = y_\alpha \delta(f_\alpha, 1) + [1 - y_\alpha]\delta(f_\alpha, 0) \tag{29.4}$$

for $\alpha = 0, 1, \ldots, nf-1$ and

$$P_{\underline{x}_i}(x_i) = \text{arbitrary prior} \tag{29.5}$$

for $i = 0, 1, \ldots, nx-1$. Note that

$$P(f^{nf} = 1^{nf} | x^{nx}) = \prod_{\alpha} y_\alpha \ . \tag{29.6}$$

Fig.29.3 gives an another bipartite bnet, alternative to Fig.29.2, corresponding to factor graph Fig.29.1. In this new bnet, we replaced the $\underline{f}_\alpha \in \{0, 1\}$ nodes by $\underline{y}_\alpha \in [0, 1]$ nodes. We also defined a new leaf node $\underline{y} \in \{0, 1\}$ with incoming arrows from all nodes $\underline{y}_\alpha$. The TPMs, printed in blue, for the $\underline{y}_\alpha$ and $\underline{y}$ nodes, are as follows.

$$P(\underline{y}_\alpha = y_\alpha | x_{nb(\underline{y}_\alpha)}) = \delta(y_\alpha, f_\alpha(x_{nb(\underline{y}_\alpha)})) \tag{29.7}$$

$$P(y | \{y_\alpha\}_{\alpha=0}^{nf-1}) = y \prod_{\alpha} y_\alpha + (1 - y) \left(1 - \prod_{\alpha} y_\alpha\right) \tag{29.8}$$

[1] Pointing arrows from the $\underline{x}_i$ to the $\underline{f}_\alpha$ is more causal than the opposite because the $\underline{x}_i$ preceed the f_α in time.

[2] Note that we are using f_α to denote both a function $f_\alpha(\cdot)$ and a Boolean value. Which one we mean will be clear from context. f_α could also be used to denote, besides a function and a Boolean value, the real number $y_\alpha = f_\alpha(x_{nb(\underline{f}_\alpha)})$. However, we won't be using it that third way in this chapter.

Figure 29.3: Another bipartite bnet, alternative to Fig.29.2, corresponding to factor graph Fig.29.1.

Note that

$$P(\underline{y} = 1 | \{y_\alpha\}_{\alpha=0}^{nf-1}) = \prod_\alpha y_\alpha \tag{29.9}$$

We've shown how to go from a factor graph to a bnet. Going from a bnet to a factor graph is also possible. Fig.29.4 gives a simple example. Bnet to factor graph conversion is used in the Junction Tree Algorithm (See Chapter 46).

Figure 29.4: Example of converting a bnet to a factor graph.

Chapter 30

Finite State Machine

This chapter is mostly based on Refs.[7], [129] and [170].

In this chapter, we define a **Finite State Machine** (FSM) (a.k.a. **Finite Automaton** (FA)) as a special case of a Petri net. Petri nets are discussed in Chapter 75. We will assume that the reader has read that chapter before tackling this one.

Some nomenclature:

Let $|S|$ denote the number of elements in a set $|S|$. The **power set of set** S, denoted by 2^S, is the set of all the subsets of S, including the empty set. Note that $|2^S| = 2^{|S|}$. Indeed, consider the binomial expansion

$$(1+1)^{|S|} = \sum_{i=0}^{|S|} \binom{|S|}{i} \tag{30.1}$$

Now note that $\binom{|S|}{i}$ is the number of subsets of S with i elements.

Consider a function $f : A \to B$.

The **domain** of f is A, and its **range** is B.

The **image** of f is $Im(f) = \{f(a) : a \in A\}$.

f is **1-1** if $f(a_1) = f(a_2)$ implies $a_1 = a_2$.

f is **onto** if $Im(f) = B$.

30.1 Deterministic FSM

30.1.1 Example

Fig.30.1 gives an example of a deterministic FSM. A FSM contains exactly one **pink place node**. We will imagine that this pink place node contains a single pink token. This token moves around with each step. Each step appends a string to the **string history** of the pink token.

Figure 30.1: Example of a deterministic FSM. Pink nodes carry the single pink token. The pink token's movements trace out the string "aac".

30.1.2 Precise Definition

Recall the following terms used for pnets:

$\mathcal{X} = \{\underline{x}_i\}_{i=1}^{nx}$ are the **transition nodes** of the pnet.

$\mathcal{P} = \{\underline{p}_i\}_{i=1}^{np}$ are the **place (or state) nodes** of the pnet.[1]

$\mathcal{X}(\underline{p} \rightarrow) = \{\underline{x} : \underline{p} \rightarrow \underline{x}$ is an arrow of the pnet$\}$.

New terms and notation not used for pnets:

Σ = **alphabet**, a finite set of symbols from which to create strings. For example, Σ might be $\{0, 1\}$ or $\{a, b, c\}$.

$\lambda : \mathcal{X} \rightarrow \Sigma$, is the **transition labelling function**. $\lambda(\underline{x}) = a$ means alphabet symbol a is assigned to transition $\underline{x}$.

$\lambda_{\underline{p}} : \mathcal{X}(\underline{p} \rightarrow) \rightarrow \Sigma$, this is the **restriction of** $\lambda()$ to the domain $\mathcal{X}(\underline{p} \rightarrow)$.

$\underline{p}(0) \in \mathcal{P}$, **starting place**. Indicated by arrow starting from nothing.

$\mathcal{P}_{acc} \subset \mathcal{P}$ are the **acceptor places**. The pink token must end in an acceptor place, but it may visit an acceptor place multiple times.

$\Delta : \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{P}$ is the **transition function**.[2] It maps each place node $\underline{p}(t) \in \mathcal{P}$ and transition $\underline{x}(t) \in \mathcal{X}$ to the next place node $\underline{p}(t+1) \in \mathcal{P}$ in a **place history** $\underline{p}(0), \underline{p}(1), \ldots$. We will write this as

$$|\underline{p}(t+1)\rangle = \underline{x}(t)|\underline{p}(t)\rangle \tag{30.2}$$

The place history is the places visited by the pink token. Each transition of the pink token between places appends a string to the pink tokens's **string history**.

---

[1] In this chapter, since we are treating FSM as a special case of pnets, we use the words "state" and "place" interchangeably.

[2] It's more common to call the function $\delta : \Sigma \times \mathcal{P} \rightarrow \mathcal{P}$, with $\delta(\lambda(\underline{x}), \underline{p}) = \Delta(\underline{x}, \underline{p})$, the transition function δ.

Figure 30.2: Example of a non-deterministic FSM.

Recall from Chapter 75 that a Petri net is defined as $\Phi_{pnet} = \langle \mathcal{P}, \mathcal{X}, W, |b(0)\rangle_{\mathcal{P}} \rangle$. A **Finite Automaton (FA)** is defined by adding extra stuff to Φ_{pnet}, namely by $\Phi_{FA} = \langle \Phi_{pnet}, \mathcal{P}_{acc}, \Sigma, \lambda, \Delta \rangle$[3]. $|b(0)\rangle_{\mathcal{P}}$ in the definition of Φ_{pnet} is replaced by $\underline{p}(0) \in \mathcal{P}$.

Let $h = a_1 a_2 a_3 \ldots a_n$ be a string over the alphabet Σ. The automaton Φ_{FA} **accepts (or generates) the string** h if a sequence of places $\underline{p}(0), \underline{p}(1), \ldots, \underline{p}(T)$ exists in $\mathcal{P}$ such that

1. $\underline{p}(0)$ is the starting place
2. $|\underline{p}(t+1)\rangle = \underline{x}(n)|\underline{p}(t)\rangle$
3. $\underline{p}(T) \in \mathcal{P}_{acc}$

If the automaton does not accept the string h, it is said to **reject** it. The set $\mathcal{L}(\Phi_{FA})$ of all strings accepted by the FA is called its **language**.

For a **deterministic FA**:

- For each $\underline{p} \in \mathcal{P}$, the function $\lambda_{\underline{p}}$ is 1-1.[a]
- we will often indicate a node for which $\lambda_{\underline{p}}(\underline{x}_i) = a$ either by $\underline{x}_i = a$ or simply by a.

[a] If it weren't 1-1, it would have to choose between 2 outgoing paths that appended the same string to the string history. This choice would make the FSM non-deterministic.

30.2 Non-deterministic FSM

30.2.1 Example

Fig.30.2 gives an example of a non-deterministic FSM.

[3] This definition has some redundancy. $\langle \mathcal{P}, \Sigma, p(0), \mathcal{P}_{acc}, \delta \rangle$ is sufficient.

30.2.2 Precise Definition

For a non-deterministic FSM, everything we said for a deterministic FSM, except for the shaded box, is valid. The shaded box must be replaced by the following shaded box:

For a **non-deterministic FA**:

- $\lambda_{\underline{p}} : \mathcal{X}(\underline{p} \rightarrow) \rightarrow 2^{\Sigma}$ instead of $\lambda_{\underline{p}} : \mathcal{X}(\underline{p} \rightarrow) \rightarrow \Sigma$ (i.e., the range of $\lambda_{\underline{p}}$ is the power set of Σ instead of Σ). The function $\lambda_{\underline{p}}$ is not necessarily 1-1. Note that the new range contains the empty set. If $\lambda_{\underline{p}}(\underline{x}) = \emptyset$, we say there is a **null transition** for arrow $\underline{p} \rightarrow \underline{x}$.[a] A null transition transfers the node across without adding anything to the string history.

- we will often indicate a node for which $\lambda_{\underline{p}}(\underline{x}_i) = \{a, b, \ldots\} \subset \Sigma$ either by $\underline{x}_i = a, b, \ldots$ or simply by $a, b, \ldots$.

[a] The empty set is sometimes denoted by ϵ when discussing FSM.

30.3 Equivalency of deterministic and non-deterministic FSM

At first glance, it might seem that non-deterministic FSM are more general than deterministic ones. However, they are equivalent in the sense that they generate identical languages (i.e., collections of strings). Their equivalence can be established by using the transformations described in Figs.30.3, 30.4 and 30.5 to reduce a non-deterministic FSM to a deterministic one.

Figure 30.3: Place nodes $\underline{p}_1, \underline{p}_2$ connected by a null transition can be merged into a single place node. This will eliminate the null transition.

Figure 30.4: A place node $\underline{p}_2$ that receives two alphabet symbols x, y at once, can be cloned into two place nodes $\underline{p}'_2$ and $\underline{p}''_2$ that receive x and y respectively.

Figure 30.5: Two place nodes $\underline{p}'_2$ and $\underline{p}''_2$ that receive the same symbol x, can be merged into a single place node $\underline{p}_2$.

Chapter 31

Frisch-Waugh-Lovell (FWL) theorem

The Frisch-Waugh-Lovell (FWL) theorem (see Ref.[136]) (mnemonic: FoWL Theorem) is a method used in Linear Regression (LR). It allows us to calculate, for LR with two features $\underline{x}_1$ and $\underline{x}_2$, the regression coefficient of feature $\underline{x}_2$ by conditioning on feature $\underline{x}_1$.

As in Chapter D on LR, we will consider two cases: x^σ non-random, and x^σ random i.i.d..

31.1 FWL, assuming x^σ are non-random

Suppose

$$y = X_1\beta_1 + X_2\beta_2 + \epsilon \tag{31.1}$$

where

$y, \epsilon \in \mathbb{R}^{nsam}$
$X_a \in \mathbb{R}^{nsam \times k_a}$, $\beta_a \in \mathbb{R}^{k_a}$ for $a = 1, 2$
Define the matrices U_1 and A_1 by

$$U_1 = X_1(X_1^T X_1)^{-1} X_1^T \tag{31.2}$$

and

$$A_1 = 1 - U_1 \ . \tag{31.3}$$

Note that

$$U_1 X_1 = X_1 \ , \quad A_1 X_1 = 0 \tag{31.4}$$

(mnemonic: A_1 Annihilates X_1, and U_1 acts like Unity on X_1).

Applying A_1 to Eq.(31.1) gives

$$A_1 y = A_1 X_2 \beta_2 + A_1 \epsilon \tag{31.5}$$

so we can estimate β_2 by

$$\widehat{\beta}_2 = (A_1 X_2)^{-1} A_1 y \ . \tag{31.6}$$

31.2 FWL, assuming x^σ are random

Assume for simplicity that $k_1 = k_2 = 1$ in Eq.(31.1). Let $\beta_1, \beta_2 \in \mathbb{R}$. When the x^σ are random and i.i.d., the X_1, X_2 are replaced by the random variables $\underline{x}_1, \underline{x}_2 \in \mathbb{R}$, and Eq.(31.1) becomes

$$\underline{y} = \beta_1 \underline{x}_1 + \beta_2 \underline{x}_2 + \underline{\epsilon}_{\underline{y}} \tag{31.7}$$

Fig.31.1 shows two LDEN bnets in which Eq.(31.7) can arise.

Figure 31.1: LDEN bnets for discussing the FWL theorem. (a) and (b) only differ in the direction of the arrow between $\underline{x}_1$ and $\underline{x}_2$.

The structural equations, printed in blue, for the 2 bnets of Fig.31.1, are as follows:

For (a),

$$\begin{cases} \underline{x}_1 = \underline{\epsilon}_{\underline{x}_1} \\ \underline{x}_2 = \alpha \underline{x}_1 + \underline{\epsilon}_{\underline{x}_2} \\ \boxed{\underline{y} = \beta_1 \underline{x}_1 + \beta_2 \underline{x}_2 + \underline{\epsilon}_{\underline{y}}} \end{cases} \tag{31.8}$$

For (b),

$$\begin{cases} \underline{x}_1 = \lambda \underline{x}_2 + \underline{\epsilon}_{\underline{x}_1} \\ \underline{x}_2 = \underline{\epsilon}_{\underline{x}_2} \\ \boxed{\underline{y} = \beta_1 \underline{x}_1 + \beta_2 \underline{x}_2 + \underline{\epsilon}_{\underline{y}}} \end{cases} \tag{31.9}$$

What we are going to say next depends only on the boxed equation, so it applies equally to cases (a) and (b).

Assume

$$\left\langle \underline{\epsilon}_{\underline{y}} \right\rangle = 0 \tag{31.10}$$

Note that

$$\left\langle \underline{x}_j, \underline{\epsilon}_{\underline{y}} \right\rangle = 0 \tag{31.11}$$

because the path from $\underline{x}_j$ to $\underline{\epsilon}_{\underline{y}}$ is blocked by a collider. Hence

$$\left\langle \underline{x}_2, \underline{y} \right\rangle^{|x_1} = \beta_2 \left\langle \underline{x}_2, \underline{x}_2 \right\rangle^{|x_1} \tag{31.12}$$

$$\beta_2 = \left[\frac{\left\langle \underline{x}_2, \underline{y} \right\rangle}{\left\langle \underline{x}_2, \underline{x}_2 \right\rangle} \right]^{|x_1} = \left[\frac{\partial}{\partial \underline{x}_2} \right]^{|x_1} \underline{y} \tag{31.13}$$

Chapter 32

Frontdoor Adjustment Formula

The frontdoor (FD) adjustment formula is proven in Chapter 22 from the rules of Do Calculus. The goal of this chapter is to give examples of the use of that theorem. We will restate the theorem in this chapter, sans proof. There is no need to understand the theorem's proof in order to use it. However, you will need to skim Chapter 22 in order to familiarize yourself with the notation used to state the theorem. This chapter also assumes that you are comfortable with the rules for checking for d-separation. Those rules are covered in Chapter 24.

Suppose that we have access to data that allows us to estimate a probability distribution $P(x., m., y.)$. Hence, the variables $\underline{x}., \underline{m}., \underline{y}.$ are ALL observed (i.e, not hidden). Then we say that the frontdoor $\underline{m}.$ satisfies the **frontdoor adjustment criterion** relative to $(\underline{x}., \underline{y}.)$ if

1. All directed paths from $\underline{x}.$ to $\underline{y}.$ are intercepted by (i.e., have a node in) $\underline{m}.$.
2. All backdoor paths from $\underline{x}.$ to $\underline{m}.$ are blocked.
3. All backdoor paths from on $\underline{m}.$ to $\underline{y}.$ are blocked by conditioning on $\underline{x}.$.

Claim 64 *Frontdoor Adjustment Formula*

If $\underline{m}.$ satisfies the frontdoor criterion relative to $(\underline{x}., \underline{y}.)$, and $P(x., m.) > 0$, then

$$P(y.|\mathcal{D}\underline{x}. = x.) = \sum_{m.} \left[\sum_{x'.} P(y.|x'., m.) P(x'.) \right] P(m.|x.) \tag{32.1}$$

$$= \tag{32.2}$$

$$\begin{tikzcd} & \mathbb{E}x'. \arrow[dr] & \\ x. \arrow[r] & \sum m. \arrow[r] & y. \end{tikzcd}$$

proof: See Chapter 22.
QED

32.1 Examples

1.

$$\begin{tikzcd} & \underline{c} \arrow[dl] \arrow[dr] & \\ \underline{x} \arrow[r] & \underline{m} \arrow[r] & \underline{y} \end{tikzcd} \tag{32.3}$$

Can't satisfy backdoor criterion because $\underline{c}$ unobserved so can't condition on it to block backdoor path $\underline{x} - \underline{c} - \underline{y}$.

If $\underline{x}. = \underline{x}, \underline{m}. = \underline{m}$ and $\underline{y}. = \underline{y}$, then the FD criterion is satisfied. In fact,

$$P(y|\mathcal{D}\underline{x} = x) = \begin{tikzcd} & & \mathbb{E}x' \arrow[dr] & \\ x \arrow[r] & \sum m \arrow[r] & & y \end{tikzcd} \tag{32.4}$$

2.

$$\begin{tikzcd} \underline{z}_1 \arrow[d] \arrow[dr] & & \underline{z}_2 \arrow[dl] \arrow[d] \\ \underline{w}_1 \arrow[d] & \underline{c} \arrow[dl] \arrow[dr] & \underline{w}_2 \arrow[d] \\ \underline{x} \arrow[r] & \underline{m} \arrow[r] & \underline{y} \end{tikzcd} \tag{32.5}$$

Can't satisfy backdoor criterion because to block backdoor path $\underline{x} - \underline{c} - \underline{y}$, need to condition on $\underline{c}$ but if this is true, then long path $\underline{x} - \underline{w}_1 - \underline{z}_1 - \underline{c} - \underline{z}_2 - \underline{w}_2 - \underline{y}$ becomes unblocked.

If $\underline{x}. = \underline{x}, \underline{m}. = \underline{m}$ and $\underline{y}. = \underline{y}$, then the FD criterion is satisfied. In fact,

$$P(y|\mathcal{D}\underline{x} = x) = \tag{32.6}$$

$$= \tag{32.7}$$

$$= \tag{32.8}$$

Chapter 33

G-formula (Sequential Backdoor Adjustment Formula)

Figure 33.1: Piscina Mirabilis, ancient Roman cistern in Naples

This chapter is based on Ref.[67] by Pearl and Robins.

A **g-formula**[1] is any formula that defines recursively the full probability distribution of a bnet. In other words, it's a recursive definition of a Dynamical Bayesian Network.[2]

The goal of this chapter is to generalize the backdoor adjustment formula (see Chapter 4) from a query $P(y|\mathcal{D}\underline{x} = x)$ with a single do node to a query $P(y|\mathcal{D}\underline{x}^n = x^n)$

[1] It's not clear from the literature what the "g" stands for. I assume it stands for "generating".

[2] Dynamical Bayesian Networks are discussed in Chapter 26.

with multiple do nodes. The resulting generalized adjustment formula is called a **sequential backdoor (SBD) adjustment formula** and it is associated with an **SBD g-formula**.

For $n = 1, 2, 3 \ldots$, define

$$\mathcal{Q}(y|x^n) = \sum_{z^n} P(y|x^n, z^n) \prod_{t=0}^{n-1} P(z_t|x_{<t}, z_{<t}) \tag{33.1}$$

$$= \begin{tikzcd} \sum_{z^n} & z^n \arrow[d] & & z_{<t} \arrow[r] & z_t \\ & y & \prod_{t=0}^{n-1} & & \\ & x^n \arrow[u] & & x_{<t} \arrow[uur] & \end{tikzcd} \tag{33.2}$$

For $n = 1$,

$$\mathcal{Q}(y|x_0) = \begin{tikzcd} \sum_{z_0} & z_0 \arrow[d] & P(z_0) \\ & y & \\ & x_0 \arrow[u] & \end{tikzcd} = \begin{tikzcd} \mathbb{E}z_0 \arrow[dr] & \\ & y \\ x_0 \arrow[ur] & \end{tikzcd} . \tag{33.3}$$

For $n = 2$,

$$\mathcal{Q}(y|x^2) = \begin{tikzcd} \sum_{z^2} & (z_0, z_1) \arrow[d] & P(z_0) & z_0 \arrow[r] & z_1 \\ & y & & & \\ & (x_0, x_1) \arrow[u] & & x_0 \arrow[uur] & \end{tikzcd} \tag{33.4}$$

$$= \begin{tikzcd} \mathbb{E}z_0 \arrow[r] \arrow[dr] & \sum z_1 \arrow[dr] & \\ & & y \\ x_0 \arrow[uur] \arrow[urr] & x_1 \arrow[ur] & \end{tikzcd} . \tag{33.5}$$

For $n = 3$,

$$\mathcal{Q}(y|x^3) = \sum_{z^3} (z_0, z_1, z_2) \; P(z_0) \; \begin{tikzcd} z_0 \arrow[r] & z_1 \arrow[r] & z_2 \\ & y & \\ & (x_0, x_1, x_2) \arrow[u] & \end{tikzcd} \begin{tikzcd} x_0 \arrow[ur] & (x_0, x_1) \arrow[ur] \end{tikzcd} \tag{33.6}$$

$$= \sum_{z^3} (z_0, z_1, z_2) \; P(z_0) \; \begin{tikzcd} z_0 \arrow[r] \arrow[rr, bend left] & z_1 \arrow[r] & z_2 \\ & y & \\ & (x_0, x_1, x_2) \arrow[u] & \end{tikzcd} \begin{tikzcd} x_0 \arrow[ur] \arrow[urr] & x_1 \arrow[ur] & x_2 \end{tikzcd} \tag{33.7}$$

$$= \begin{tikzcd} \mathbb{E}z_0 \arrow[r] \arrow[rr, bend left] \arrow[drrr] & \sum z_1 \arrow[r] \arrow[drr] & \sum z_2 \arrow[dr] & \\ & & & y \\ x_0 \arrow[uur] \arrow[uurr] \arrow[urrr] & x_1 \arrow[uur] \arrow[urr] & x_2 \arrow[ur] & \end{tikzcd} \tag{33.8}$$

Suppose that we have access to data that allows us to estimate a probability distribution $P(x^n, y, z^n)$. Hence, the variables $\underline{x}^n, \underline{y}, \underline{z}^n$ are ALL observed (i.e, not hidden). Then we say that the multinode of "covariates" $\underline{z}^n$ satisfies the **sequential backdoor (SBD) adjustment criterion** relative to $(\underline{x}^n, \underline{y})$ if for all $t \in \{0, 1, \ldots, n-1\}$,

1. $\underline{y} \perp \underline{x}_t | \underbrace{(\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{t-1}, \underline{z}_0, \underline{z}_1, \ldots, \underline{z}_t)}_{\text{Past of } \underline{x}_t}$ in $\mathcal{L}_{\underline{x}_t} \mathcal{D}_{\underline{x}_{t+1}, \underline{x}_{t+2}, \ldots, \underline{x}_{n-1}} G$.

2. $\underline{z}_t \cap de(\underline{x}_t) = \emptyset$.

Claim 65 *(SBD Adjustment Formula)*
If $\underline{z}^n$ satisfies the sequential backdoor criterion relative to $(\underline{x}^n, \underline{y})$, then

$$P(y|\mathcal{D}\underline{x}^n = x^n) = \mathcal{Q}(y|x^n) \; , \tag{33.9}$$

where $\mathcal{Q}(y|x^n)$ is defined by Eq.(33.2).

proof: If z^n satisfies the SBD criterion relative to $(\underline{x}^n, \underline{y})$, then $\underline{x}^n, \underline{y}, \underline{z}^n$ might have the following structure for $n = 3$.

(33.10)

One can check using the following 3 auxiliary bnets that bnet Eq.(33.10) satisfies the SBD criterion. Note that conditioned nodes are shaded yellow.

(33.11)

See Claim 59 for a proof of this claim for the special case Eq.(33.10).

QED

Chapter 34

Gaussian Nodes with Linear Dependence on Parents

Bnet nodes that have a Gaussian TPM with a linear dependence on their parent nodes (GLP) are a very popular way of modeling continuous nodes of bnets. A convenient aspect of them is that their parents can be discrete or continuous nodes, and their children can be discrete or continuous nodes too. Also, they can be learned easily from the data because their parameters can be expressed in terms of two node covariances. For these reasons, they are commonly used when doing structure learning of bnets with continuous nodes (see Chapter 95).

Figure 34.1: GLP node $\underline{y}$ with 3 parent nodes $\underline{x}^3 = (\underline{x}_1, \underline{x}_2, \underline{x}_3)$.

Recall our notation for a Gaussian distribution:

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \ , \tag{34.1}$$

where $x, \mu \in \mathbb{R}$ and $\sigma > 0$.

A GLP node $\underline{y}$ with n parents $\underline{x}^n = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n)$ has the following TPM:

$$P(y|x^n) = \mathcal{N}(y; \beta_0 + \beta^{nT} x^n, \sigma^2) \tag{34.2}$$

where $\underline{y}, \beta_0, \in \mathbb{R}$ and $\sigma^2 > 0$, and where $\underline{x}^n, \beta^n \in \mathbb{R}^n$ are **column vectors**. The T in β^{nT} stands for transpose. Any $\underline{x}_i$ can have a discrete set of states as long as

they are real valued and ordinal (ordered by size). Fig.34.1 shows a diagrammatic representation of a GLP node with 3 parents.

Note that as $\sigma \to 0$, a GLP node becomes deterministic. In fact, it becomes a neural net node with a linear activation function.

An equivalent way of defining a GLP node $\underline{y}$ is in terms of a random variable equation expressing $\underline{y}$ as a hyperplane function of the parents $\underline{x}^n$ plus a Gaussian noise variable. Define a curve-fit $\underline{\widehat{y}}$ of a "true value" $\underline{y}$ by

$$\underline{\widehat{y}} = \beta_0 + \beta^{nT} \underline{x}^n \tag{34.3a}$$

and

$$\underline{y} = \underline{\widehat{y}} + \underline{\epsilon} \tag{34.3b}$$

where the residual $\underline{\epsilon}$ satisfies

$$P(\epsilon) = \mathcal{N}(\epsilon; 0, \sigma^2) \tag{34.3c}$$

and

$$\langle \underline{x}^n, \underline{\epsilon} \rangle = 0 \ . \tag{34.3d}$$

The notation $\left\langle \underline{x}, \underline{y} \right\rangle$ for the covariance of random variables $\underline{x}$ and $\underline{y}$ is explained in Chapter C.

Claim 66 *The parameters of a GLP node can be expressed in terms of 2-node covariances. Specifically,*

$$\beta^n = \left\langle \underline{x}^n, \underline{x}^{nT} \right\rangle^{-1} \left\langle \underline{y}, \underline{x}^n \right\rangle \tag{34.4}$$

$$\beta_0 = \left\langle \underline{y} \right\rangle - \beta^{nT} \left\langle \underline{x}^n \right\rangle \tag{34.5}$$

$$\sigma^2 = \left\langle \underline{y}, \underline{y} \right\rangle - \beta^{nT} \left\langle \underline{x}^n, \underline{y} \right\rangle \tag{34.6}$$

proof:

Note that $\left\langle \underline{x}^n, \underline{x}^{nT} \right\rangle^T = \left\langle \underline{x}^n, \underline{x}^{nT} \right\rangle$ and $\left\langle \underline{y}, \underline{x}^{nT} \right\rangle^T = \left\langle \underline{y}, \underline{x}^n \right\rangle$.

$$\left\langle \underline{y}, \underline{x}^{nT} \right\rangle = \beta^{nT} \left\langle \underline{x}^n, \underline{x}^{nT} \right\rangle \tag{34.7}$$

$$\left\langle \underline{y}, \underline{x}^n \right\rangle = \left\langle \underline{x}^n, \underline{x}^{nT} \right\rangle \beta^n \tag{34.8}$$

$$\beta^n = \left\langle \underline{x}^n, \underline{x}^{nT} \right\rangle^{-1} \left\langle \underline{y}, \underline{x}^n \right\rangle \tag{34.9}$$

$$\left\langle \underline{y} \right\rangle = \beta_0 + \beta^{nT} \left\langle \underline{x}^n \right\rangle \tag{34.10}$$

$$\begin{aligned} \left\langle \underline{y}, \underline{y} \right\rangle &= \left\langle \beta_0 + \beta^{nT} \underline{x}^n + \underline{\epsilon}, \underline{y} \right\rangle & (34.11) \\ &= \beta^{nT} \left\langle \underline{x}^n, \underline{y} \right\rangle + \sigma^2 & (34.12) \end{aligned}$$

QED

Let D=Discrete, GLP=Gaussian with Linear dependence in Parents
The following arrows are possible in a bnet.

- $GLP \leftarrow GLP$
- $GLP \leftarrow D$

  Pass to GLP a separate set of regression coefficients β_0, β^n and variance σ^2 for each state of D. If D is called $\underline{d}$, let

  $$P(y|(x^n)_d, d) = \mathcal{N}(y; (\beta_0)_d + (\beta^{nT})_d (x^n)_d, \sigma_d^2) \tag{34.13}$$

  for each $d \in val(\underline{d})$.

- $D \leftarrow GLP$

  If D expects a continuous parent, no need to preprocess GLP output. If D expects a discrete parent, break the interval $[a, b]$ that contains most of the range of the GLP node into sub-intervals and assign a discrete label to each subinterval.

- $D \leftarrow D$

Chapter 35

Generalized Linear Model (GLM)

This chapter is based on chapter 4 of Ref.[1].

35.1 Exponential Family of Distributions

The **Exponential Family (EF)** of probability distributions is defined by

$$P(y|\theta,\phi) = \exp\left(\frac{\theta y - b(\theta)}{a(\phi)} + c(y,\phi)\right) \tag{35.1}$$

In this chapter, we will denote averages over $y|\theta,\phi$ by angular brackets

$$E_{\underline{y}|\theta,\phi}[f(y)] = \sum_y P(y|\theta,\phi) f(y) = \langle f(y) \rangle \tag{35.2}$$

As usual in this book, let $val(\underline{x})$ denote the set of values that the random variable $\underline{x}$ can take. We will assume that $val(\underline{y})$ for EF can be either a discrete or a continuous subset of $\mathbb{R}$,[1] but $val(\underline{\theta})$ must be continuous. When $val(\underline{y})$ is a discrete subset of $\mathbb{R}$, $P(y|\theta,\phi)$ will denote a probability distribution, whereas when $val(\underline{y})$ is continuous, it will denote a probability density.

Claim 67

$$\mu = \left\langle \underline{y} \right\rangle = b'(\theta) \tag{35.3}$$

$$\sigma^2 = \left\langle \underline{y}, \underline{y} \right\rangle = a(\phi) b''(\theta) \tag{35.4}$$

proof:

[1] By a "continuous set" we mean a finite set of intervals each of which has non-zero length.

$$
\begin{aligned}
0 &= \partial_\theta \int_{-\infty}^{\infty} dy\ P(y|\theta,\phi) && (35.5)\\
&= \int_{-\infty}^{\infty} dy\ \frac{1}{a}[y - b'(\theta)]P(y|\theta,\phi) && (35.6)\\
&= \frac{1}{a}[\langle \underline{y} \rangle - b'(\theta)] && (35.7)
\end{aligned}
$$

$$
\begin{aligned}
0 &= \partial_\theta^2 \int_{-\infty}^{\infty} dy\ P(y|\theta,\phi) && (35.8)\\
&= \int_{-\infty}^{\infty} dy\ \left\{ \frac{-b''(\theta)}{a} + \frac{1}{a^2}[y - b'(\theta)]^2 \right\} P(y|\theta,\phi) && (35.9)
\end{aligned}
$$

Hence,

$$
\left\langle [y - b'(\theta)]^2 \right\rangle = ab''(\theta) \tag{35.10}
$$

QED

Note that the Normal Distribution belongs to the EF. Indeed,

$$
\begin{aligned}
\mathcal{N}(y;\mu,\sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2} \right) && (35.11)\\
&= \exp\left(\frac{-y^2 + 2\mu y - \mu^2}{2\sigma^2} - \ln\sqrt{2\pi\sigma^2} \right) && (35.12)\\
&= \exp\left(\frac{-\frac{1}{2}y^2 + \mu y - \frac{1}{2}\mu^2}{\sigma^2} - \ln\sqrt{2\pi\sigma^2} \right) && (35.13)
\end{aligned}
$$

So, for the Normal Distribution, $\theta = \mu$, $a = \sigma^2$, $b = \mu^2/2$, $b' = \mu$, $b'' = 1$.

EF can be defined for an ensemble $\{y_\sigma : \sigma \in \Sigma\}$ of independent (but not necessarily identically distributed) random variables y_σ describing individuals σ of a population Σ.

$$
P(\vec{y}|\vec{\theta},\phi) = \prod_\sigma \exp\left(\frac{\theta_\sigma y_\sigma - b(\theta_\sigma)}{a(\phi)} + c(y_\sigma,\phi) \right) \tag{35.14}
$$

$$
\mu_\sigma = \left\langle \underline{y}_\sigma \right\rangle = \partial_{\theta_\sigma} b(\theta_\sigma) \tag{35.15}
$$

$$
\left\langle \underline{y}_\sigma, \underline{y}_{\sigma'} \right\rangle = \delta(\sigma,\sigma') a(\phi) \partial_{\theta_\sigma}^2 b(\theta_\sigma) \tag{35.16}
$$

35.2 GLM

The Generalized Linear Model (GLM) is a statistical model for an ensemble of independent (but not necessarily identically distributed) random variables $\underline{y}_\sigma$. GML consists of 3 parts:

1. Exponential Family

   Model $\underline{y}_\sigma$ by probability distribution of exponential family (EF).

2. Linear predictor

   In EF, replace θ_σ by the **linear predictor** $X_\sigma^T\beta = \sum_i X_{\sigma,i}\beta_i$. $X_\sigma^T\beta$ is commonly denoted by η_σ, but I will avoid that notation because I think the results are much clearer when expressed in the more explicit notation $X_\sigma^T\beta$ instead of η_σ.

3. Link Function

$$\mu_\sigma = \left\langle \underline{y}_\sigma \right\rangle = \partial_{X_\sigma^T\beta} b \tag{35.17}$$

$$X_\sigma^T\beta = g(\mu_\sigma) = \widehat{\theta}(\mu_\sigma) \tag{35.18}$$

$$\mu_\sigma = g^{-1}(X_\sigma^T\beta) = \widehat{\mu}(X_\sigma^T\beta) \tag{35.19}$$

$\widehat{\theta}() = g()$ is called the **link function**.

Note that in Linear Regression (LR), we consider independent (but not necessarily identically distributed) random variables $\underline{y}_\sigma \in \mathbb{R}$ that satisfy

$$\underline{y}_\sigma = X_\sigma^T\beta + \underline{\epsilon}_\sigma \tag{35.20}$$

where

$$\langle \epsilon_\sigma \rangle = 0, \quad \langle \underline{\epsilon}_\sigma, \underline{\epsilon}_\sigma \rangle = \delta(\sigma,\sigma')\sigma_\sigma^2 \tag{35.21}$$

Equivalently, one states that

$$\underline{y}_\sigma \sim \mathcal{N}(\mu_\sigma = X_\sigma^T\beta, \sigma_\sigma^2) \tag{35.22}$$

Hence, for LR, the link function and its inverse are the identity map.

For the Bernoulli distribution with $y_\sigma \in \{0,1\}$,

$$\begin{aligned} Ber(y_\sigma; p_\sigma) &= p_\sigma^{y_\sigma}(1-p_\sigma)^{1-y_\sigma} && (35.23)\\ &= \exp(y_\sigma \ln p_\sigma + (1-y_\sigma)\ln(1-p_\sigma)) && (35.24)\\ &= \exp\left(y_\sigma \underbrace{\ln\left(\frac{p_\sigma}{1-p_\sigma}\right)}_{X_\sigma^T\beta} + \underbrace{\ln(1-p_\sigma)}_{-b} \right) && (35.25) \end{aligned}$$

$\mu_\sigma = p_\sigma$, and $X_\sigma^T\beta = \text{lodds}(\mu_\sigma)$ so $\mu_\sigma = \text{smoid}(X_\sigma^T\beta)$.
$g() = \text{lodds}()$ and $g^{-1}() = \text{smoid}()$

$$a = 1 \tag{35.26}$$

$$\begin{aligned} b &= -\ln(1-\mu_\sigma) && (35.27)\\ &= -\ln\left(1 - \frac{1}{1+e^{-X_\sigma^T\beta}}\right) && (35.28)\\ &= -\ln\left(\frac{e^{-X_\sigma^T\beta}}{1+e^{-X_\sigma^T\beta}}\right) && (35.29)\\ &= \ln(1+e^{X_\sigma^T\beta}) && (35.30) \end{aligned}$$

$$\begin{aligned} \partial_{X_\sigma^T\beta} b &= \frac{e^{X_\sigma^T\beta}}{1+e^{X_\sigma^T\beta}} && (35.31)\\ &= \text{smoid}(X_\sigma^T\beta) && (35.32)\\ &= \mu_\sigma && (35.33) \end{aligned}$$

Table 35.1 gives various probability distributions and their natural link functions, for cases where the link function is simple and easy to invert.

GLM is a generalization of LR. Recall some of the main results of LR:

$$y = X\beta + \epsilon \tag{35.34}$$

$$\langle \epsilon \rangle = 0, \quad \left\langle \epsilon, \epsilon^T \right\rangle = \sigma^2 I_N \tag{35.35}$$

where I_N is the $N = |\Sigma|$ dimensional unit matrix.

The Maximum Likelihood Estimate (MLE) for β is

$$\widehat{\beta} = (X^T X)^{-1} X^T y \ . \tag{35.36}$$

prob. distribution	link function $X_\sigma^T\beta = g(\mu_\sigma)$	$\mu_\sigma = g^{-1}(X_\sigma^T\beta)$
Normal $\underline{y}_\sigma \in (-\infty, +\infty)$	$X_\sigma^T\beta = \mu_\sigma$ (g= identity map)	$\mu_\sigma = X_\sigma^T\beta$
Exponential $\underline{y}_\sigma \in (0, +\infty)$	$X_\sigma^T\beta = -\frac{1}{\mu_\sigma}$	$\mu_\sigma = -\frac{1}{X_\sigma^T\beta}$
Poisson $\underline{y}_\sigma \in \{0, 1, 2, \ldots\}$	$X_\sigma^T\beta = \ln \mu_\sigma$	$\mu_\sigma = \exp(X_\sigma^T\beta)$
Bernoulli $\underline{y}_\sigma \in \{0, 1\}$	$X_\sigma^T\beta = \text{lodds}(\mu_\sigma)$	$\mu_\sigma = \text{smoid}(X_\sigma^T\beta)$

Table 35.1: Various probability distributions and their natural link functions.

The covariance of $\widehat{\beta}$ is

$$\begin{aligned} \left\langle \widehat{\beta}, \widehat{\beta}^T \right\rangle &= \left\langle X^T X)^{-1} X^T y, y^T X (X^T X)^{-1} \right\rangle & (35.37) \\ &= \sigma^2 (X^T X)^{-1} & (35.38) \end{aligned}$$

Next we will try to find analogous results for GLM. We will give (1) a numerical method for calculating an estimate $\widehat{\beta}$ and (2) an asymptotic expression for $\left\langle \widehat{\beta}, \widehat{\beta}^T \right\rangle$.

Let

$$LL = \sum_\sigma LL_\sigma \tag{35.39}$$

where

$$\begin{aligned} LL_\sigma &= LL_{y_\sigma|\theta_\sigma} & (35.40) \\ &= \frac{y_\sigma \theta_\sigma - b(\theta_\sigma)}{a(\phi)} + c(y_\sigma, \phi) & (35.41) \end{aligned}$$

The Newton-Raphson method for calculating an estimate $\widehat{\beta}$ is as follows. Let $u^T = [\frac{\partial \langle LL \rangle}{\partial \beta_j}]_{j=0,1,2,\ldots}$

$H = [H_{j,j'}]$, $H_{j,j'} = \frac{\partial^2 \langle LL \rangle}{\partial \beta_j \partial \beta_{j'}}$. H is called the **Hessian matrix**

For $t = 0, 1, 2, \ldots$, consider the Taylor expansion to second order of $\langle LL \rangle (\beta)$ about the point $\beta = \beta^{(t)}$

$$\langle LL \rangle (\beta) \approx \langle LL \rangle (\beta^{(t)}) + u^{(t)T}(\beta - \beta^{(t)}) + \frac{1}{2}(\beta - \beta^{(t)})^T H^{(t)} (\beta - \beta^{(t)}) \tag{35.42}$$

By Section C.31, we have,

$$0 = \frac{\partial \langle LL \rangle (\beta)}{\partial \beta^T} = u^{(t)} + H^{(t)}(\beta - \beta^{(t)}) \tag{35.43}$$

This last equation suggests the recursion

$$\beta^{(t+1)} = \beta^{(t)} - (H^{(t)})^{-1} u^{(t)} \ . \tag{35.44}$$

Fig.35.1 gives a graphical representation of one cycle of this recursion.

Figure 35.1: One cycle of Newton-Raphson method for calculating an estimate $\widehat{\beta}$ for the GLM.

Claim 68

$$\frac{\partial LL_\sigma}{\partial \beta_j} = \frac{[y_\sigma - b'(X_\sigma^T \beta)]}{\langle y_\sigma, y_\sigma \rangle} \frac{\partial \widehat{\mu}_\sigma}{\partial X_\sigma^T \beta} X_{\sigma,j} \tag{35.45}$$

proof:

$$\frac{\partial LL_\sigma}{\partial \beta_j} = \frac{\partial LL_\sigma}{\partial \theta_\sigma} \frac{\partial \theta_\sigma}{\partial \mu_\sigma} \frac{\partial \widehat{\mu}_\sigma}{\partial X_\sigma^T \beta} \frac{\partial X_\sigma^T \beta}{\partial \beta_j} \tag{35.46}$$

$$\frac{\partial LL_\sigma}{\partial \theta_\sigma} = \frac{y_\sigma - b'(\theta_\sigma)}{a(\phi)} \tag{35.47}$$

$$\frac{\partial \mu_\sigma}{\partial \theta_\sigma} = b''(\theta_\sigma) = \frac{\langle y_\sigma, y_\sigma \rangle}{a(\phi)} \tag{35.48}$$

$$\frac{\partial \widehat{\mu}_\sigma}{\partial X_\sigma^T \beta} = (g^{-1})'(X_\sigma^T \beta) \tag{35.49}$$

$$\frac{\partial X_\sigma^T \beta}{\partial \beta_j} = X_{\sigma,j} \tag{35.50}$$

$$\frac{\partial LL_\sigma}{\partial \beta_j} = \frac{[y_\sigma - b'(\theta_\sigma)]}{\langle y_\sigma, y_\sigma \rangle} \frac{\partial \widehat{\mu}_\sigma}{\partial X_\sigma^T \beta} X_{\sigma,j} \tag{35.51}$$

QED

Claim 69 *(Asymptotic expression for $\left\langle \widehat{\beta}, \widehat{\beta}^T \right\rangle$ for GLM)*

$$\left\langle \widehat{\beta}, \widehat{\beta}^T \right\rangle \to \mathcal{I}^{-1} \quad \textit{(asymptotic covariance)} \tag{35.52}$$

(Hence, more information $\mathcal{I}$ yields a smaller variance.) where

$$\mathcal{I} = X^T W X \tag{35.53}$$

and

$$W_{\sigma,\sigma'} = \left[\left(\frac{\partial \widehat{\mu}}{\partial X_\sigma^T \beta} \right)^2 \frac{\delta(\sigma, \sigma')}{\langle y_\sigma, y_\sigma \rangle} \right]_{\beta = \widehat{\beta}} \tag{35.54}$$

$\mathcal{I}$ is called the **information matrix**.

proof:

$$\left\langle \frac{\partial LL_\sigma}{\partial \beta_j} \frac{\partial LL_\sigma}{\partial \beta_{j'}} \right\rangle = \left\langle \frac{[y_\sigma - b'(X_\sigma^T \beta)]}{\langle y_\sigma, y_\sigma \rangle} \frac{\partial \widehat{\mu}_\sigma}{\partial X_\sigma^T \beta} X_{\sigma,j} \frac{[y_\sigma - b'(X_\sigma^T \beta)]}{\langle y_\sigma, y_\sigma \rangle} \frac{\partial \widehat{\mu}_\sigma}{\partial X_\sigma^T \beta} X_{\sigma,j'} \right\rangle \tag{35.55}$$

$$= \left[\frac{\partial \widehat{\mu}_\sigma}{\partial X_\sigma^T \beta} \right]^2 \frac{X_{\sigma,j} X_{\sigma,j'}}{\langle y_\sigma, y_\sigma \rangle} \tag{35.56}$$

$$\sum_\sigma \left\langle \frac{\partial LL_\sigma}{\partial \beta_j} \frac{\partial LL_\sigma}{\partial \beta_{j'}} \right\rangle = (X^T W X)_{j,j'} \tag{35.57}$$

By Section C.31, we have

$$\left\langle \frac{\partial^2 LL_\sigma}{\partial \beta_j \partial \beta_{j'}} \right\rangle = - \left\langle \frac{\partial LL_\sigma}{\partial \beta_j} \frac{\partial LL_\sigma}{\partial \beta_{j'}} \right\rangle \tag{35.58}$$

Summing both sides of the last equation over σ, we find

$$\left\langle \frac{\partial^2 LL}{\partial \beta_j \partial \beta_{j'}} \right\rangle = -(X^T W X)_{j,j'} \tag{35.59}$$

According to Section C.31, we have

$$\left\langle \widehat{\beta}, \widehat{\beta}^T \right\rangle \to (X^T W X)^{-1} \tag{35.60}$$

QED

Chapter 36

Generative Adversarial Networks (GANs)

Figure 36.1: Generative Adversarial Network (GAN)

Original GAN, Ref.[23](2014).

Generator G (counterfeiter) generates samples $\vec{f}$ of fake money and submits them to Discriminator D (Treasury agent). D also gets samples $\vec{r}$ of real money. D submits verdict $V \in [0, 1]$. G depends on parameter θ_G and D on parameter θ_D. Verdict V and initial θ_G, θ_D are used to get new parameters θ'_G, θ'_D.Process is repeated

Figure 36.2: Discriminator node $\underline{V}$ in Fig.36.1 can be split into 3 nodes $\vec{\underline{c}}$, $\vec{\underline{d}}$ and $\underline{V}$.

(Dynamical Bayesian Network) until saddle point in $V(\theta_G, \theta_D)$ is reached. D makes G better and vice versa. Zero-sum game between D and G.

Let $\mathcal{D}$ be the domain of $D(\cdot, \theta_D)$. Assume that for any $x \in \mathcal{D}$,

$$0 \leq D(x, \theta_D) \leq 1 \ . \tag{36.1}$$

For any $S \subset \mathcal{D}$, define

$$\sum_{x \in S} D(x, \theta_D) = \lambda(S, \theta_D) \ . \tag{36.2}$$

In general, $G(\cdot, \theta_G)$ need not be real valued.

Assume that for every $u \in val(\underline{u})$, $G(u, \theta_G) = f \in val(\underline{f}) \subset \mathcal{D}$. Define

$$\overline{D}(f, \theta_D) = 1 - D(f, \theta_D) \ . \tag{36.3}$$

Note that

$$0 \leq \overline{D}(f, \theta_D) \leq 1 \ . \tag{36.4}$$

Define:

$$V(\theta_G, \theta_D) = \sum_r P(r) \ln D(r, \theta_D) + \sum_u P(u) \ln \overline{D}(G(u, \theta_G), \theta_D) \ . \tag{36.5}$$

We want the first variation of $V(\theta_G, \theta_D)$ to vanish.

$$\delta V(\theta_G, \theta_D) = 0 \ . \tag{36.6}$$

This implies

$$\partial_{\theta_G} V(\theta_G, \theta_D) = \partial_{\theta_D} V(\theta_G, \theta_D) = 0 \tag{36.7}$$

and

$$V_{opt} = \min_{\theta_G} \max_{\theta_D} V(\theta_G, \theta_D) \ . \tag{36.8}$$

The TPMs, printed in blue, for bnets Figs.36.1 and 36.2, are as follows:

$$P(\theta_G) = \text{ given} \tag{36.9}$$

$$P(\theta_D) = \text{ given} \tag{36.10}$$

$$P(\vec{u}) = \prod_i P(u[i]) \ \ \text{(usually uniform distribution)} \tag{36.11}$$

$$P(\vec{r}) = \prod_i P(r[i]) \tag{36.12}$$

$$P(f[i] \mid \vec{u}, \theta_G) = \delta[f[i], G(u[i], \theta_G)] \tag{36.13}$$

$$P(c[i] \mid \vec{f}, \theta_D) = \delta(c[i], \overline{D}(f[i], \theta_D)) \tag{36.14}$$

$$P(d[j] \mid \vec{r}, \theta_D) = \delta(d[j], D(r[j], \theta_D)) \tag{36.15}$$

$$P(V|\vec{d}, \vec{c}) = \delta(V, \frac{1}{N} \ln \prod_{i,j} (c[i]d[j])) \tag{36.16}$$

where $N = nsam(\vec{r})nsam(\vec{u})$.

Let $\eta_G, \eta_D > 0$. Maximize V wrt θ_D, and minimize it wrt θ_G.

$$P(\theta'_G|V, \theta_G) = \delta(\theta'_G, \theta_G - \eta_G \partial_{\theta_G} V) \tag{36.17}$$

$$P(\theta'_D|V,\theta_D) = \delta(\theta'_D, \theta_D + \eta_D \partial_{\theta_D} V) \tag{36.18}$$

Figure 36.3: GAN, Constraining Bayesian Network

Constraining Bnet given in Fig.36.3. It adds 2 new nodes, namely $\underline{\vec{U}}$ and $\underline{\vec{R}}$, to the bnet of Fig.36.1. The purpose of these 2 barren (childrenless) nodes is to constrain certain functions to be probability distributions.

The TPMs, printed in blue, for the 2 new nodes, are as follows.

$$P(U[i] \mid \theta_G) = \frac{\overline{D}(G(U[i],\theta_G),\theta_D))}{\overline{\lambda}(\theta_G,\theta_D)} \tag{36.19}$$

where $val(\underline{U[i]}) = val(\underline{u})$ and $\overline{\lambda}(\theta_G,\theta_D) = \sum_u \overline{D}(G(u,\theta_G),\theta_D))$.

$$P(R[i] \mid \theta_G,\theta_D) = \frac{D(R[i],\theta_D)}{\lambda(\theta_D)} \tag{36.20}$$

where $val(\underline{R[i]}) = val(\underline{r})$ and $\lambda(\theta_D) = \sum_r D(r,\theta_D)$.

$$P(V|\vec{u},\vec{r}) = \delta(V, \frac{1}{N} \ln \prod_{i,j} (P(\underline{R[i]} = r[i] \mid \theta_G,\theta_D) P(\underline{U[i]} = u[j] \mid \theta_G))) \tag{36.21}$$

where $N = nsam(\vec{r})nsam(\vec{u})$.

$L =$ likelihood

$$L = P(\vec{r}, \vec{u} | \theta_G, \theta_D) \tag{36.22}$$

$$= \prod_{i,j} \left[\frac{D(r[i], \theta_D)}{\lambda(\theta_D)} \frac{\overline{D}(G(u[j], \theta_G), \theta_D))}{\overline{\lambda}(\theta_G, \theta_D)} \right] \tag{36.23}$$

$$\ln L = N[V(\theta_G, \theta_D) - \ln \lambda(\theta_D) - \ln \overline{\lambda}(\theta_G, \theta_D)] \tag{36.24}$$

Chapter 37

Goodness of Causal Fit

See my paper and software Ref.[90].

Chapter 38

Gradient Descent

Gradient Descent (GD) is when we have a sequence of points $w_k \in \mathbb{R}^n$ and a convex loss function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ such that

$$\boxed{w_{k+1} = w_k - \alpha \nabla_w \mathcal{L}(w_k)} \tag{38.1}$$

for some "learning rate" $\alpha > 0$. Since the function $\mathcal{L}(w)$ is convex, it has a minimum w^* and $w_k \to w^*$ as $k \to \infty$.

In Machine Learning (ML), it is usually the case that

$$\mathcal{L}(w) = \sum_{\sigma=1}^{nsam} \widehat{\mathcal{L}}(\widehat{y}^\sigma(x^\sigma, w), y^\sigma) \tag{38.2}$$

where the sum is over $nsam$ samples. Normally, $nsam$ would be all the samples in a dataset. In ML, $nsam$ is often very large, and $\nabla_w \widehat{\mathcal{L}}$ cannot be calculated analytically so it must be calculated numerically[1], individually for each of the $nsam$ samples. In such cases, a Monte Carlo method called **Stochastic Gradient Descent** (SGD) is often used. SGD just means choosing at random a small subset of the $nsam$ samples ("mini-batch"), and approximating $\nabla_w \mathcal{L}(w)$ by an average of the gradients for the mini-batch.

This description of GD is fairly complete, so why this chapter? In this chapter, we will give a dynamical bnet (see Chapter 26) illustrating how GD is used in ML.

Samples $(x^\sigma, y^\sigma) \in val(\underline{x}) \times val(\underline{y})$ are given. $nsam(\vec{x}) = nsam(\vec{y})$.

Estimator function $\widehat{y}(x; w)$ for $x \in val(\underline{x})$ and $w \in \mathbb{R}$ is given.

Let

$$P_{\underline{x},\underline{y}}(x, y) = \frac{1}{nsam(\vec{x})} \sum_\sigma \mathbb{1}(x = x^\sigma, y = y^\sigma) \ . \tag{38.3}$$

[1] Gradients can be calculated numerically by the method of back-propagation, which is explained in Chapter 5.

$$
\begin{tikzcd}
 & \underline{\vec{x}} \arrow[r] \arrow[d] & \underline{\vec{y}} \arrow[d] & \\
w \arrow[r] \arrow[rrr, bend right] & \underline{\widehat{\vec{y}}} \arrow[r] & \mathcal{L} \arrow[r] & w'
\end{tikzcd}
$$

Figure 38.1: Basic gradient descent bnet.

Let

$$\mathcal{L}(\vec{x}, \vec{y}, w) = \frac{1}{nsam(\vec{y})} \sum_{\sigma} |y^{\sigma} - \widehat{y}(x^{\sigma}; w)|^2 \tag{38.4}$$

$\mathcal{L}$ is called the **mean square error**.

Best fit is parameters w^* such that

$$w^* = \underset{w}{\operatorname{argmin}}\, \mathcal{L}(\vec{x}, \vec{y}, w) \ . \tag{38.5}$$

The TPMs, printed in blue, for the basic curve fitting bnet Fig.38.1, are as follows.

$$P(w) = \text{given} \ . \tag{38.6}$$

The first time it is used, w is arbitrary. After the first time, it is determined by previous stage.

$$P(\vec{x}) = \prod_{\sigma} P_{\underline{x}}(x^{\sigma}) \tag{38.7}$$

$$P(\vec{y}|\vec{x}) = \prod_{\sigma} P_{\underline{y}|\underline{x}}(y^{\sigma} \mid x^{\sigma}) \tag{38.8}$$

$$P(\widehat{y}^{\sigma}|w, \vec{x}) = \delta(\widehat{y}^{\sigma}, \widehat{y}(x^{\sigma}; w)) \tag{38.9}$$

$$P(\mathcal{L}|\widehat{\vec{y}}, \vec{y}) = \delta(\mathcal{L}, \frac{1}{nsam(\vec{x})} \sum_{\sigma} |y^{\sigma} - \widehat{y}^{\sigma}|^2) \ . \tag{38.10}$$

$$P(w'|w, \mathcal{L}) = \delta(w', w - \alpha \partial_w \mathcal{L}) \tag{38.11}$$

$\alpha > 0$ is called **the descent rate or learning rate**. If $\Delta w = w' - w = -\alpha \frac{\partial \mathcal{L}}{\partial w}$, then $\Delta \mathcal{L} = \frac{-1}{\alpha} (\Delta w)^2 < 0$ so this will minimize the error $\mathcal{L}$. This is an example of GD.

Chapter 39

Granger Causality

This chapter is based on the Wikipedia article Ref.[142] and Scholarpedia article Ref.[74] on Granger causality and on the book Ref.[26] on time series analysis by Hamilton.

This chapter assumes the reader has read Chapter 102 on the stationary time series $ARMA(p, q)$ and $VAR(p)$.

Figure 39.1: When t-series $\mathcal{X}$ Granger-causes t-series $\mathcal{Y}$, the patterns in $\mathcal{X}$ are approximately repeated in $\mathcal{Y}$ after some time lag (two examples are indicated with arrows). Thus, past values of $\mathcal{X}$ can be used for the prediction of future values of $\mathcal{Y}$. (image and caption from Ref.[142])

Let $\underline{\vec{x}}_t = [\underline{x}_t^A]_{\forall A} \in \mathbb{R}^{nr_1 \times 1}$ and $\underline{\vec{y}}_t = [\underline{y}_t^B]_{\forall B} \in \mathbb{R}^{nr_2 \times 1}$. Thus, $\underline{\vec{x}}_t$ for each t is a column vector with nr_1 rows, and $\underline{\vec{y}}_t$ for each t is a column vector with nr_2 rows. Let $nr_1 + nr_2 = nr$, the total number of rows. Consider a vector t-series $\{\underline{\vec{x}}_t, \underline{\vec{y}}_t\}_{\forall t}$ of type $VAR(p)$, as defined by Eq.(102.183). To simplify the notation of Eq.(102.183), we are replacing x^1 by x, x^2 by y, n^1 by n, and n^2 by w.

$$\begin{bmatrix} \underline{x}_t^A \\ \underline{y}_t^B \end{bmatrix} = \sum_{j=1}^{p} \begin{bmatrix} \alpha_j^{\underline{x}|\underline{x};A,A'} & \alpha_j^{\underline{x}|\underline{y};A,B'} \\ \alpha_j^{\underline{y}|\underline{x};B,A'} & \alpha_j^{\underline{y}|\underline{y};B,B'} \end{bmatrix} \begin{bmatrix} \underline{x}_{t-j}^{A'} \\ \underline{y}_{t-j}^{B'} \end{bmatrix} + \begin{bmatrix} \underline{n}_t^A \\ \underline{w}_t^B \end{bmatrix} \tag{39.1}$$

Hence,

$$E_{|\vec{x}_{<t},\vec{y}_{<t}}\left[\underline{y}_t^B\right] = \sum_{j=1}^{p} \alpha_j^{\underline{y}|\underline{x};B,A'} \underline{x}_{t-j}^{A'} + \sum_{j=1}^{p} \alpha_j^{\underline{y}|\underline{y};B,B'} \underline{y}_{t-j}^{B'} \tag{39.2}$$

Let $\mathcal{X} = \{\underline{\vec{x}}_t\}_{\forall t}$ and $\mathcal{Y} = \{\underline{\vec{y}}_t\}_{\forall t}$.

We say $\mathcal{X}$ **does not G-cause (or does not G-forecast)** $\mathcal{Y}$, and symbolize this by $\mathcal{X} \underset{G}{\not\rightarrow} \mathcal{Y}$, if

$$\alpha_j^{\underline{y}|\underline{x};B,A'} = 0 \;\; \forall (j, B, A') \tag{39.3}$$

or, equivalently,

$$E_{|\vec{x}_{<t},\vec{y}_{<t}}\left[\underline{y}_t^B\right] = E_{|\vec{y}_{<t}}\left[\underline{y}_t^B\right] \;\; \forall (B, t) \tag{39.4}$$

We say $\mathcal{X}$ **G-causes (or G-forecasts)** $\mathcal{Y}$ and symbolize this by $\mathcal{X} \underset{G}{\longrightarrow} \mathcal{Y}$, if $\mathcal{X} \underset{G}{\not\rightarrow} \mathcal{Y}$ is false.

Figure 39.2: Bnet for $VAR(2)$. For clarity, we only show the arrows entering nodes $\underline{\vec{x}}_t$ and $\underline{\vec{y}}_t$. Remove red arrows if $\mathcal{X}$ does not G-cause $\mathcal{Y}$, and keep them if it does.

Eq.(39) describing a bipartite $VAR(p)$ t-series can be represented by the bnet Fig.39.2. The TPMs, printed in blue, for the two nodes $\vec{x}_t$ and $\vec{y}_t$, are as follows:

$$P(\underline{\vec{x}}_t|\underline{\vec{x}}_{[t-p,t-1]}, \underline{\vec{y}}_{[t-p,t-1]}, \vec{n}_t) = \prod_A \mathbb{1}\left(\underline{x}_t^A = \sum_{j=1}^{p} \alpha_j^{\underline{x}|\underline{x};A,A'} \underline{x}_{t-j}^{A'} + \sum_{j=1}^{p} \alpha_j^{\underline{x}|\underline{y};A,B'} \underline{y}_{t-j}^{B'} + n_t^A\right) \tag{39.5}$$

$$P(\underline{\vec{y}}_t|\underline{\vec{x}}_{[t-p,t-1]},\underline{\vec{y}}_{[t-p,t-1]},\vec{w}_t) = \prod_B \mathbb{1}\left(\underline{y}_t^B = \sum_{j=1}^{p} \underbrace{\alpha_j^{\underline{y}|\underline{x};B,A'}}_{0?} \underline{x}_{t-j}^{A'} + \sum_{j=1}^{p} \alpha_j^{\underline{y}|\underline{y};B,B'} \underline{y}_{t-j}^{B'} + w_t^B\right) \tag{39.6}$$

Testing for GC

- Consider the datasets

$$\mathcal{D}_{\underline{x}} = \{(t, \vec{x}_{[t-p,t-1]}, \vec{y}_{[t-p,t-1]}, \boxed{\vec{x}_t}) : t\} \tag{39.7}$$

$$\mathcal{D}_{\underline{y}} = \{(t, \vec{x}_{[t-p,t-1]}, \vec{y}_{[t-p,t-1]}, \boxed{\vec{y}_t}) : t\} \tag{39.8}$$

  One can do Linear Regression on $\mathcal{D}_{\underline{x}}$ with x-variables $(\vec{x}_{[t-p,t-1]}, \vec{y}_{[t-p,t-1]})$, y-variable $\vec{x}_t$, and regression coefficients $\alpha_{[1,p]}^{\underline{x}|\underline{x}}$, $\alpha_{[1,p]}^{\underline{x}|\underline{y}}$. One can also do Linear Regression on $\mathcal{D}_{\underline{y}}$ with x-variables $(\vec{x}_{[t-p,t-1]}, \vec{y}_{[t-p,t-1]})$, y-variable $\vec{y}_t$, and regression coefficients $\alpha_{[1,p]}^{\underline{y}|\underline{x}}$, $\alpha_{[1,p]}^{\underline{y}|\underline{y}}$. The LR software yields confidence intervals for the regression coefficients.

- One can do hypothesis testing using the Likelihood Ratio Test[1]

  Null hypothesis $H_0 : \mathcal{X} \underset{G}{\not\longrightarrow} \mathcal{Y}$, Alternative hypothesis $H_1 : \mathcal{X} \underset{G}{\longrightarrow} \mathcal{Y}$

- Test for both $\mathcal{X} \underset{G}{\longrightarrow} \mathcal{Y}$ and $\mathcal{Y} \underset{G}{\longrightarrow} \mathcal{X}$.

Limitations

It has been remarked that G-causality is not true causality because, even though an event A must precede an event B in order to cause it, that does not necessarily mean that A causes B. I think the problem with G-causality is that it uses a bnet that is a good fit for the dataset, but not necessarily also a good *causal* fit for the experiment. One can measure the goodness of causal fit of a bnet by doing do-intervention experiments (See Chapter 37).

[1]The Likelihood Ratio Test is discussed in Section C.25.

Chapter 40

Hidden Markov Model

A dynamical Bayesian network (DBN) (see Chapter 26) is a generalization of a Hidden Markov Model (HMM), which in turn is a generalization of a Kalman Filter (KF) (see Chapter 47).

See Wikipedia article Ref.[143] to learn about the history and many uses of HMMs. This chapter is based on Refs.[53], [143], [195], [108].

In this chapter, we use the following conventions.

Random variables are underlined and their values are not. For example, $\underline{a} = a$ means the random variable $\underline{a}$ takes the value a. A diagram with all its nodes underlined represents a Bayesian Network (bnet), whereas the same diagram with the letters not underlined represents a specific **instantiation** of that bnet. For example $\underline{a} \to \underline{b} \to \underline{c}$ represents the bnet with full probability distribution $P(c|b)P(b|a)P(a)$, whereas $a \to b \to c$ represents $P(c|b)P(b|a)$. Note that, for convenience, we define $a \to b \to c$ to exclude the priors of root nodes such as $P(a)$.

If $\underline{a}$ is a root node, then $\mathbb{E}a$ signifies a weighted sum $\sum_a P(a)$. For example,

$$\mathbb{E}a \to b \to c = \sum_a P(c|b)P(b|a)P(a) \tag{40.1}$$

If $\underline{a}$ is not a root node, then $\sum a$ signifies a simple unweighted sum $\sum_a$. For example,

$$x \to \sum a \to y = \sum_a P(y|a)P(a|x) \tag{40.2}$$

Two bnets are said to be equal if their full probability distributions (i.e., their full instantiations) are equal numerically. For example,

$$\underline{a} \to \underline{b} \to \underline{c} = P(c|b)P(b|a)P(a) = \underline{a} \leftarrow \underline{b} \leftarrow \underline{c} \tag{40.3}$$

Unobserved (a.k.a. hidden, latent) nodes are indicated in a bnet by enclosing their label in a dashed circle. For example, $\overset{\frown}{\underset{\smile}{(\ u\ )}}$. Alternatively, they are indicated by using dashed arrows for all arrows emanating from the unobserved node.

Suppose

$\underline{v}^n = (\underline{v}_0, \underline{v}_1, \ldots, \underline{v}_{n-1})$ are n visible nodes that are measured, and

Figure 40.1: HMM bnet with $n = 4$.

$\underline{x}^n = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{n-1})$ are the n hidden, unmeasurable state nodes of a system that is being monitored.

For the bnet of Fig.40.1, one has

$$P(x^n, v^n) = \prod_{t=0}^{n-1} P(x_t|x_{t-1})P(v_t|x_t) \ , \tag{40.4}$$

where $x_{-1} = \emptyset$.

We assume that this bnet is stationary. The following notation emphasizes that fact:

$\pi(x)$ = prior probability

$$\pi(x) = P(\underline{x}_0 = x) \tag{40.5}$$

$A(x|x')$ = transition matrix

$$A(x|x') = P(\underline{x}_t = x|\underline{x}_{t-1} = x') \tag{40.6}$$

$B(v|x)$ = emission probability

$$B(v|x) = P(\underline{v}_t = v|\underline{x}_t = x) \tag{40.7}$$

Let $x_{<t} = (x_0, x_1, \ldots, x_{t-1})$.
For $t = 0, 1, \ldots, n-1$, define
$\mathcal{F}_t(x_t)$=future measurements probability

$$\mathcal{F}_t(x_t) = P(v_{>i}|x_t) \tag{40.8}$$

$$= \begin{tikzcd} x_t \arrow[r] & \sum x_{>t} \arrow[d] \\ & v_{>t} \end{tikzcd} \tag{40.9}$$

$\overline{\mathcal{F}}_t(x_t)$= past and present measurements probability

$$\overline{\mathcal{F}}_t(x_t) = P(v_{<t}, v_t, x_t) \tag{40.10}$$

$$= \begin{array}{ccc} \mathbb{E}x_{<t} & \longrightarrow & x_t \\ \downarrow & & \downarrow \\ v_{<t} & & v_t \end{array} \tag{40.11}$$

$\lambda_t(x_t)$= present measurement probability, (a.k.a. emission probability $B(v_t|x_t)$). $\lambda_t(x_t)$ is the likelihood of x_t.

$$\lambda_t(x_t) = P(v_t|x_t) \tag{40.12}$$

$$= \begin{array}{c} x_t \\ \downarrow \\ v_t \end{array} \tag{40.13}$$

40.1 Calculating $P(x_t, v^n)$ and $P(x_t, x_{t+1}, v^n)$

Claim 70 *For $t \geq 0$,*

$$P(x_t, v^n) = \overline{\mathcal{F}}_t(x_t)\mathcal{F}_t(x_t) \tag{40.14}$$

$$= \begin{array}{ccccc} \mathbb{E}x_{<t} & \longrightarrow & x_t & x_t \longrightarrow & \sum x_{>t} \\ \downarrow & & \downarrow & & \downarrow \\ v_{<t} & & v_t & & v_{>t} \end{array} . \tag{40.15}$$

For $t > 0$,

$$P(x_{t-1}, x_t, v^n) = \overline{\mathcal{F}}_{t-1}(x_{t-1})P(x_t|x_{t-1})\lambda_t(x_t)\mathcal{F}_t(x_t) \tag{40.16}$$

$$= \begin{array}{ccccccc} \mathbb{E}x_{<t-1} & \longrightarrow & x_{t-1} & x_{t-1} \longrightarrow & x_t & x_t \quad x_t \longrightarrow & \sum x_{>t} \\ \downarrow & & \downarrow & & & \downarrow & \downarrow \\ v_{<t-1} & & v_{t-1} & & & v_t & v_{>t} \end{array} . \tag{40.17}$$

proof:

$$P(x_t, v^n) = \sum_{x_{<i}} \sum_{x_{>i}} P(x^n, v^n) \tag{40.18}$$

$$= \sum_{x_{<i}} \sum_{x_{>i}} P(x^n, v^n|x_t)P(x_t) \tag{40.19}$$

$$= \sum_{x_{<i}} \sum_{x_{>i}} P(x_{<i}, v_{<i}, v_t|x_t)P(x_{>t}, v_{>i}|x_t)P(x_t) \tag{40.20}$$

$$= P(v_{<i}, v_t|x_t)P(v_{>i}|x_t)P(x_t) \tag{40.21}$$

$$= \overline{\mathcal{F}}_t(x_t)\mathcal{F}_t(x_t) \tag{40.22}$$

$$
\begin{aligned}
P(x_{t-1}, x_t, v^n) &= \sum_{x_{<t-1}} \sum_{x_{>t}} P(x^n, v^n) && (40.23)\\
&= \sum_{x_{<t-1}} \sum_{x_{>t}} P(x^n, v^n | x_{t-1}, x_t) P(x_{t-1}, x_t) && (40.24)\\
&= \sum_{x_{<t-1}} \sum_{x_{>t}} P(x_{<t-1}, v_{<t-1}, v_{t-1} | x_{t-1}) P(v_t | x_t) P(x_{t-1}, x_t) P(x_{>i}, v_{>i} | x_t) && (40.25)\\
&= P(v_{<t-1}, v_{t-1} | x_{t-1}) P(v_t | x_t) P(x_{t-1}, x_t) P(v_{>i} | x_t) && (40.26)\\
&= \overline{\mathcal{F}}_{t-1}(x_{t-1}) \lambda_t(x_t) P(x_t | x_{t-1}) \mathcal{F}_t(x_t) && (40.27)
\end{aligned}
$$

QED

40.2 Calculating $\mathcal{F}_t$ and $\overline{\mathcal{F}}_t$

Claim 71 *For $t > 0$, $\mathcal{F}_t$ and $\overline{\mathcal{F}}_t$ can be calculated recursively as follows:*

$$
\begin{aligned}
\overline{\mathcal{F}}_t(x_t) &= \sum_{x_{t-1}} \overline{\mathcal{F}}_{t-1}(x_{t-1}) P(x_t | x_{t-1}) \lambda_t(x_t) && (40.28)\\
&= \sum_{x_{t-1}} \begin{tikzcd} \mathbb{E}x_{<t-1} \arrow[r] \arrow[d] & x_{t-1} \arrow[d] & x_{t-1} \arrow[r] & x_t & x_t \arrow[d] \\ v_{<t-1} & v_{t-1} & & & v_t \end{tikzcd} && (40.29)
\end{aligned}
$$

and

$$
\begin{aligned}
\mathcal{F}_{t-1}(x_{t-1}) &= \sum_{x_t} P(x_t | x_{t-1}) \lambda_t(x_t) \mathcal{F}_t(x_t) && (40.30)\\
&= \sum_{x_t} \begin{tikzcd} x_{t-1} \arrow[r] & x_t & x_t \arrow[d] & x_t \arrow[r] & \sum x_{>t} \arrow[d] \\ & & v_t & & v_{>t} \end{tikzcd} && (40.31)
\end{aligned}
$$

proof:

$$
\begin{aligned}
\overline{\mathcal{F}}_t(x_t) \mathcal{F}_t(x_t) &= P(x_t, v^n) && (40.32)\\
&= \sum_{x_{t-1}} P(x_{t-1}, x_t, v^n) && (40.33)\\
&= \sum_{x_{t-1}} \overline{\mathcal{F}}_{t-1}(x_{t-1}) \lambda_t(x_t) P(x_t | x_{t-1}) \mathcal{F}_t(x_t) && (40.34)
\end{aligned}
$$

$$
\begin{aligned}
\overline{\mathcal{F}}_{t-1}(x_{t-1})\mathcal{F}_{t-1}(x_{t-1}) &= P(x_{t-1}, v^n) && (40.35)\\
&= \sum_{x_t} P(x_{t-1}, x_t, v^n) && (40.36)\\
&= \sum_{x_t} \overline{\mathcal{F}}_{t-1}(x_{t-1})\lambda_t(x_t)P(x_t|x_{t-1})\mathcal{F}_t(x_t) && (40.37)
\end{aligned}
$$

QED

40.3 Calculating $P(x^n|v^n)$

Claim 72

$$
P(x_t|x_{t-1}, v^n) = \frac{P(x_t|x_{t-1})\lambda_t(x_t)\mathcal{F}_t(x_t)}{\mathcal{F}_{t-1}(x_{t-1})} \tag{40.38}
$$

$$
= \frac{\begin{tikzcd} x_{t-1} \arrow[r] & x_t \arrow[d] & x_t \arrow[r] & \sum x_{>t} \arrow[d] \\ & v_t & & v_{>t} \end{tikzcd}}{\begin{tikzcd} x_{t-1} \arrow[r] & \sum x_t \arrow[r] \arrow[d] & \sum x_{>t} \arrow[d] \\ & v_t & v_{>t} \end{tikzcd}} \tag{40.39}
$$

Note that actually, $P(x_t|x_{t-1}, v^n) = P(x_t|x_{t-1}, v_{\geq t})$ by d-separation, but we won't use this fact.

$$
P(x_{t-1}|x_t, v^n) = \frac{\overline{\mathcal{F}}_{t-1}(x_{t-1})P(x_t|x_{t-1})\lambda_t(x_t)}{\overline{\mathcal{F}}_t(x_t)} \tag{40.40}
$$

$$
= \frac{\begin{tikzcd} \mathbb{E}x_{<t-1} \arrow[r] \arrow[d] & x_{t-1} \arrow[d] & x_{t-1} \arrow[r] & x_t & x_t \arrow[d] \\ v_{<t-1} & v_{t-1} & & & v_t \end{tikzcd}}{\begin{tikzcd} \mathbb{E}x_{<t-1} \arrow[r] \arrow[d] & \sum x_{t-1} \arrow[r] \arrow[d] & x_t \arrow[d] \\ v_{<t-1} & v_{t-1} & v_t \end{tikzcd}} \tag{40.41}
$$

Note that actually $P(x_{t-1}|x_t, v^n) = P(x_{t-1}|x_t, v_{\leq t-1})$ by d-separation, but we won't use this fact.

proof:

$$P(x_t|x_{t-1}, v^n) = \frac{P(x_{t-1}, x_t, v^n)}{P(x_{t-1}, v^n)} \tag{40.42}$$

$$= \frac{\overline{\mathcal{F}}_{t-1}(x_{t-1})\lambda_t(x_t)P(x_t|x_{t-1})\mathcal{F}_t(x_t)}{\overline{\mathcal{F}}_{t-1}(x_{t-1})\mathcal{F}_{t-1}(x_{t-1})} \tag{40.43}$$

Analogous proof for Eq.(40.41).
QED

$$P(x^n|v^n) = \prod_{t=1,\ldots,n-1,n} P(x_t|x_{t-1}, v^n) \quad \text{(forward propagation)} \tag{40.44}$$

$$= \prod_{t=n+1,n,\ldots 3,2} P(x_{t-1}|x_t, v^n) \quad \text{(backward propagation)} \tag{40.45}$$

40.4 Calculating $P(v^n|A, B, \pi)$

$P(v^n|A, B, \pi)$ can be calculated

- from $\overline{\mathcal{F}}_{n-1}(x_{n-1})$ (past of $\underline{x}_{n-1}$)

$$P(v^n|A, B, \pi) = \sum_{x_{n-1}} \underbrace{P(v_{<n-1}, v_{n-1}, x_{n-1})}_{\overline{\mathcal{F}}_{n-1}(x_{n-1})} \tag{40.46}$$

$$= \begin{array}{ccc} \mathbb{E}x_{<n-1} & \longrightarrow & \sum x_{n-1} \\ \downarrow & & \downarrow \\ v_{<n-1} & & v_{n-1} \end{array} \tag{40.47}$$

- from $\mathcal{F}_0(x_0)$ (future of $\underline{x}_0$)

$$P(v^n|A, B, \pi) = \sum_{x_0} P(x_0)P(v_0|x_0)\underbrace{P(v_{>0}|x_0)}_{\mathcal{F}_0(x_0)} \tag{40.48}$$

$$= \begin{array}{ccc} \mathbb{E}x_0 & \longrightarrow & \sum x_{>0} \\ \downarrow & & \downarrow \\ v_0 & & v_{>0} \end{array} \tag{40.49}$$

- from $\overline{\mathcal{F}}_t(x_t)$ and $\mathcal{F}_t(x_t)$ for some t (past and future of $\underline{x}_t$)

$$
\begin{aligned}
P(v^n|A,B,\pi) &= \sum_{x_t} \underbrace{P(v_{<t}, v_t, x_t)}_{\overline{\mathcal{F}}_t(x_t)} \underbrace{P(v_{>t}|x_t)}_{\mathcal{F}_t(x_t)} && (40.50)\\
&= \begin{tikzcd}[column sep=small] \mathbb{E}x_{<t} \arrow[r] \arrow[d] & \sum x_t \arrow[r] \arrow[d] & \sum x_{>t} \arrow[d] \\ v_{<t} & v_t & v_{>t} \end{tikzcd} . && (40.51)
\end{aligned}
$$

40.5 Calculating $\widehat{x}^n$ (Viterbi algorithm)

x^n is not visible, only v^n is. Here is how to find an estimate $\widehat{x}^n$ of x^n.

Define

$$
\begin{aligned}
\overline{\mathcal{F}}_t^{max}(x_t) &= \max_{x_{<t}} P(v_{<t}, v_t, x_{<t}, x_t) && (40.52)\\
&= \begin{tikzcd}[column sep=small] \max x_{<t} \arrow[r] \arrow[d] & x_t \arrow[d] \\ v_{<t} & v_t \end{tikzcd} && (40.53)
\end{aligned}
$$

$$
\begin{aligned}
\widehat{x}_{t-1}(x_t) &= \operatorname*{argmax}_{x_{t-1}} \overline{\mathcal{F}}_{t-1}^{max}(x_{t-1}) P(x_t|x_{t-1}) \lambda_t(x_t) && (40.54)\\
&= \begin{tikzcd}[column sep=small] \max x_{<t-1} \arrow[r] \arrow[d] & \operatorname{argmax} x_{t-1} \arrow[r] \arrow[d] & x_t \arrow[d] \\ v_{<t-1} & v_{t-1} & v_t \end{tikzcd} && (40.55)
\end{aligned}
$$

For $t > 0$, $\overline{\mathcal{F}}_t^{max}$ and $\widehat{x}_{t-1}(x_t)$ can be calculated recursively as follows:

$$
\begin{aligned}
\overline{\mathcal{F}}_t^{max}(x_t) &= \max_{x_{t-1}} \overline{\mathcal{F}}_{t-1}^{max}(x_{t-1}) P(x_t|x_{t-1}) \lambda_t(x_t) && (40.56)\\
&= \max_{x_{t-1}} \quad \begin{tikzcd}[column sep=small] \max x_{<t-1} \arrow[r] \arrow[d] & x_{t-1} \arrow[d] & x_{t-1} \arrow[r] & x_t & x_t \arrow[d] \\ v_{<t-1} & v_{t-1} & & & v_t \end{tikzcd} && (40.57)
\end{aligned}
$$

$$
\begin{aligned}
\widehat{x}_{t-1}(x_t) &= \operatorname*{argmax}_{x_{t-1}} \overline{\mathcal{F}}_{t-1}^{max}(x_{t-1}) P(x_t|x_{t-1}) \lambda_t(x_t) && (40.58)\\
&= \operatorname*{argmax}_{x_{t-1}} \quad \begin{tikzcd}[column sep=small] \max x_{<t-1} \arrow[r] \arrow[d] & x_{t-1} \arrow[d] & x_{t-1} \arrow[r] & x_t & x_t \arrow[d] \\ v_{<t-1} & v_{t-1} & & & v_t \end{tikzcd} && (40.59)
\end{aligned}
$$

Claim 73 *(Viterbi Algorithm)*

If

$$\widehat{x}^n = \underset{x^n}{\operatorname{argmax}}\, P(x^n|v^n) \ , \tag{40.60}$$

then the components of $\widehat{x}^n$ can be calculated recursively from the last one $\widehat{x}_{n-1}$ to the first one $\widehat{x}_0$, as follows. Let

$$\widehat{x}_{n-1} = \underset{x_{n-1}}{\operatorname{argmax}}\, \overline{\mathcal{F}}^{max}_{n-1}(x_{n-1}) \tag{40.61}$$

$$= \begin{array}{ccc} \max x_{<n-1} & \longrightarrow & \operatorname{argmax} x_{n-1} \\ \downarrow & & \downarrow \\ v_{<n-1} & & v_{n-1} \end{array} \tag{40.62}$$

and for $t < n-1$, use

$$\widehat{x}_{t-1} = \widehat{x}_{t-1}(\widehat{x}_t) \tag{40.63}$$

$$= \begin{array}{ccccc} \max x_{<t-1} & \longrightarrow & \operatorname{argmax} x_{t-1} & \longrightarrow & \widehat{x}_t \\ \downarrow & & \downarrow & & \downarrow \\ v_{<t-1} & & v_{t-1} & & v_t \end{array} \tag{40.64}$$

proof:
QED

40.6 Calculating $\widehat{A}$, $\widehat{B}$, $\widehat{\pi}$ (Baum-Welch algorithm)

Let $\theta = (A, B, \pi)$. θ is a set of hidden parameters. Here is how to find an estimate $\widehat{\theta}$ of θ.

If x^n and v^n were visible, we could use

$$\widehat{\pi}(x) = \mathbb{1}(\underline{x}_0 = x) \tag{40.65}$$

$$\widehat{A}(x'|x) = \frac{\sum_{t=0}^{n-2} \mathbb{1}(\underline{x}_t = x, \underline{x}_{t+1} = x')}{\sum_x numerator} \tag{40.66}$$

$$\widehat{B}(v|x) = \frac{\sum_{t=0}^{n-1} \mathbb{1}(\underline{v}_t = v, \underline{x}_t = x)}{\sum_v numerator} \tag{40.67}$$

But x^n is not visible. So how can we estimate θ under those circumstances?

Define

$$
\begin{aligned}
\gamma_t(x_t) = \quad &= \quad P(x_t|v^n) && (40.68)\\
&= \quad \frac{P(x_t, v^n)}{P(v^n)} && (40.69)\\
&= \quad \frac{\overline{\mathcal{F}}_t(x_t)\mathcal{F}_t(x_t)}{\sum_{x_t} numerator} && (40.70)\\
&= \quad \frac{\begin{array}{ccccc} \mathbb{E}x_{<t} & \longrightarrow & x_t & x_t & \longrightarrow \sum x_{>t} \\ \downarrow & & \downarrow & & \downarrow \\ v_{<t} & & v_t & & v_{>t} \end{array}}{\sum_{x_t} numerator} && (40.71)
\end{aligned}
$$

$$
\begin{aligned}
\xi_t(x_t, x_{t+1}) \quad &= \quad \frac{P(x_t, x_{t+1}, v^n)}{\sum_{x_t}\sum_{x_{t+1}} numerator} && (40.72)\\
&= \quad \frac{\overline{\mathcal{F}}_{t-1}(x_{t-1})P(x_t|x_{t-1})\lambda_t(x_t)\mathcal{F}_t(x_t)}{\sum_{x_t}\sum_{x_{t+1}} numerator} && (40.73)\\
&= \quad \frac{\begin{array}{ccccccc} \mathbb{E}x_{<t-1} & \longrightarrow & x_{t-1} & x_{t-1} & \longrightarrow & x_t \;\; x_t & x_t \longrightarrow \sum x_{>t} \\ \downarrow & & \downarrow & & & \downarrow & \downarrow \\ v_{<t-1} & & v_{t-1} & & & v_t & v_{>t} \end{array}}{\sum_{x_t}\sum_{x_{t+1}} numerator} && (40.74)
\end{aligned}
$$

Claim 74 *(Baum-Welch algorithm)*
If

$$
\widehat{\theta} = \underset{\theta}{\operatorname{argmax}}\, P(v^n|\theta) \ , \tag{40.75}
$$

then we can find $\widehat{\theta}$ using the following formulae:

$$
\widehat{\pi}(x) = \overbrace{\gamma_0(x)}^{\sim P(x)} \tag{40.76}
$$

$$
\widehat{A}(x'|x) = \frac{\sum_{t=0}^{n-2} \overbrace{\xi_t(x, x')}^{\sim P(x,x')}}{\sum_{t=0}^{n-2} \underbrace{\gamma_t(x)}_{P(x)}} \tag{40.77}
$$

$$
\widehat{B}(v|x) = \frac{\sum_{t=0}^{n-1} \overbrace{\mathbb{1}(\underline{v}_t = v)}^{\sim P(v|x)} \overbrace{\gamma_t(x)}^{\sim P(x)}}{\sum_{t=0}^{n-1} \underbrace{\gamma_t(x)}_{\sim P(x)}} \tag{40.78}
$$

proof:
QED

Chapter 41

Identification of do queries via LDEN diagrams

The most general way to decide whether a do query for a particular DAG is identifiable, is by using Pearl's Do Calculus rules (see Chapter 22). However, those rules are fairly complicated and therefore difficult to automate.

We contend that by analyzing any DAG *symbolically* using SCuMpy (see Ref.[95]), one can decide rigorously whether a do query for that DAG is identifiable or not. Hence, SCuMpy allows us, *if we have a single specific DAG in mind*, to bypass and supplant, in an automated fashion, the Do Calculus rules.

SCuMpy uses LDEN bnets (a.k.a. linear SCM, see Chapter 52) whereas the Do Calculus rules are for general bnets. However, the answer to the question of whether a do query is identifiable for a particular DAG, only depends on the DAG, so it is independent of whether the DAG came from an LDEN bnet, or from the more general corresponding bnet.

In this chapter, we will briefly summarize the results presented in the Jupyter notebook entitled "unconfounded-children" in SCuMpy (Ref .[95]). See that notebook for more details.

Consider the LDEN bnet of Fig.41.1. This DAG does not satisfy either the backdoor or frontdoor criteria, but the query $P(y|do(x))$ is still known to be identifiable for this DAG.

For the LDEN bnet of Fig.41.1, SCuMpy gives the following covariance between nodes $\underline{x}$ and $\underline{y}$:

$$\left\langle \underline{x}, \underline{y} \right\rangle = \begin{cases} \alpha_{\underline{x}|\underline{h_1}} \sigma^2_{\underline{\epsilon_{h_1}}} \left(\alpha_{\underline{m_1}|\underline{x}} \alpha_{\underline{x}|\underline{h_1}} \alpha_{\underline{y}|\underline{m_1}} + \alpha_{\underline{m_2}|\underline{x}} \alpha_{\underline{x}|\underline{h_1}} \alpha_{\underline{y}|\underline{m_2}} + \alpha_{\underline{y}|\underline{h_1}} \right) \\ + \sigma^2_{\underline{\epsilon_x}} \left(\alpha_{\underline{m_1}|\underline{x}} \alpha_{\underline{y}|\underline{m_1}} + \alpha_{\underline{m_2}|\underline{x}} \alpha_{\underline{y}|\underline{m_2}} \right) \end{cases} \tag{41.1}$$

Suppose we amputate, from the LDEN bnet of Fig.41.1, all arrows entering node $\underline{x}$ (for this example, that means amputating the arrow $\underline{h}_1 \rightarrow \underline{x}$). Such an amputation is demanded by the definition of the do query $P(y|do(x))$, Then SCuMpy gives instead:

Figure 41.1: LDEN bnet for which the query $P(y|do(x))$ is known to be identifiable. External root nodes $\underline{\epsilon_a}$ pointing into each node $\underline{a}$, are left implicit. The path coefficients (a.k.a arrow gains) are not shown either. For any two nodes $\underline{a}, \underline{b}$, these gains are denoted by $\alpha_{\underline{b}|\underline{a}}$ for an arrow $\underline{a} \to \underline{b}$.

$$\left\langle \underline{x}, \underline{y} \right\rangle = \sigma^2_{\underline{\epsilon_x}} \left(\alpha_{\underline{m_1}|\underline{x}} \alpha_{\underline{y}|\underline{m_1}} + \alpha_{\underline{m_2}|\underline{x}} \alpha_{\underline{y}|\underline{m_2}} \right) \tag{41.2}$$

so

$$\frac{\partial \underline{x}}{\partial \underline{y}} = \frac{\left\langle \underline{x}, \underline{y} \right\rangle}{\left\langle \underline{x}, \underline{x} \right\rangle} = \alpha_{\underline{m_1}|\underline{x}} \alpha_{\underline{y}|\underline{m_1}} + \alpha_{\underline{m_2}|\underline{x}} \alpha_{\underline{y}|\underline{m_2}} \tag{41.3}$$

Note that upon amputating the arrows entering $\underline{x}$, the covariance $\left\langle \underline{x}, \underline{y} \right\rangle$ becomes independent of the hidden (i.e., unobserved) variables $\underline{h}_1, \underline{h}_2$. Thus, the do query $P(y|do(x))$ is identifiable, because it doesn't depend on unobserved quantities.

Chapter 42

Influence Diagrams & Utility Nodes

This chapter is based on Refs.[77], [39] and [8].

An **Influence Diagram (ID)** is a DAG that generalizes a bnet by adding 2 new types of nodes. IDs have 3 types of nodes: chance, decision and utility (a.k.a value) nodes, represented, respectively, by ovals, rectangles and diamonds.[1]

Fig.42.1 and the information printed in blue that follows it, are a simple example of an ID.

Figure 42.1: Simple example of an Influence Diagram. Chance, decision and utility nodes are indicated by a no-frame, rectangle and doubly-lined rectangle, respectively.

- Chance Nodes:

$$P(b|c,d_1=0) = \begin{bmatrix} .1 & .3 \\ .9 & .7 \end{bmatrix}_{b,c,d_1=0}, \quad P(b|c,d_1=1) = \begin{bmatrix} .2 & .4 \\ .8 & .6 \end{bmatrix}_{b,c,d_1=1}, \quad P(c) = \begin{bmatrix} .6 \\ .4 \end{bmatrix}_c \tag{42.1}$$

[1]Since our bnet drawing software `xypic` cannot draw diamond frames for nodes, in this book, we sometimes represent chance, decision and utility nodes by no-frames, rectangles and doubly-lined rectangles, respectively.

- Utility nodes:

$$u_1(d_1) = \begin{bmatrix} 10 \\ -23 \end{bmatrix}_{d_1}, \quad u_2(d_2, c) = \begin{bmatrix} 2 & -8 \\ 45 & 7 \end{bmatrix}_{d_2,c} \tag{42.2}$$

- Decision nodes:

  $P(d_1)$ and $P(d_2|b, d_1)$ are deterministic TPMs obtain by optimization. They maximize $u = u_1 + u_2$.

Table 42.1 compares these 3 types of nodes in an ID. Note that a decision node is a deterministic node[2], but its TPM is not known a priori. Instead, it is derived by an optimization algorithm which maximizes the total utility.

	Chance node (oval)	Decision node (rectangle)	Utility node (diamond)
type of states	discrete	discrete	continuous
deterministic?	Not in general	Yes	Yes
transition info[a]	TPM	deterministic TPM	utility function, not normalized
transition info known a priori	Yes	No	Yes
children type	chance, decision, value	chance, decision, value	None

Table 42.1: Comparison of the 3 types of nodes in an influence diagram.

[a]by "transition info" of a node, we mean "personal" node info other than a list of the states of the node and a list of its parent nodes

42.1 Definitions

Let

$\underline{c}.$ = set of **chance nodes**

$\underline{d}.$ = set of **decision nodes**. Assume $\underline{d}_i$ are in chronological (a.k.a., topological) order, i.e., $\underline{d}_i$ occurs before (or concurrent with) $\underline{d}_{i+1}$ for all i.

$\underline{u}.$ = set of **utility nodes**. Often, utility nodes are called **value nodes** and only their sum is called a utility node.

$\underline{X}. = \underline{c}. \cup \underline{d}. \cup \underline{u}.$ all nodes

$pa(\underline{X}_i)$ = parents of node $\underline{X}_i$

$val(\underline{X}_i)$ = set of values (a.k.a. states) that node $\underline{X}_i$ can assume.

[2]Deterministic nodes are defined by Eq.(F.1)

$\underline{e}. =$ **evidence nodes**, the subset $\underline{e}.$ of the chance nodes $\underline{c}.$ that is **apriori observed** (i.e. measured before the experiment starts). If a decision node has a chance node as a parent, we will assume it is measured immediately before the decision is made, not apriori.

$u_i(pa(\underline{u}_i)) =$ **partial utility function**

$u(\underline{c}., d.) = \sum_i u_i(pa(\underline{u}_i))$ **total utility function**

- **Chance node partition**

  Given nd decision nodes $\{\underline{d}_i\}_{i=1}^{nd}$, we partition the set of chance nodes $\underline{c}.$ into $nd+1$ disjoint subsets $\{\underline{\mathcal{C}}_i\}_{i=0}^{nd}$ in such a way that all nodes in $\underline{\mathcal{C}}_{i-1}$ occur before (or concurrent with) $\underline{d}_i$, and $\underline{d}_i$ occurs before (or concurrent with) all nodes in $\underline{\mathcal{C}}_i$. Thus,

$$\underline{\mathcal{C}}_0 \prec \underline{d}_1 \prec \underline{\mathcal{C}}_1 \prec \underline{d}_2 \ldots \prec \underline{\mathcal{C}}_{nd-1} \prec \underline{d}_{nd} \prec \underline{\mathcal{C}}_{nd} \tag{42.3}$$

  where $\prec$ is the partial chronological (a.k.a., topological) ordering.

- **Optimal Strategy (or Policy)**

  Define the conditional probabilities

$$P(d_i|pa(d_i)) = \delta(d_i, \Delta_i(pa(d_i))) \tag{42.4}$$

  and

$$P(c.|d.) = \prod_i P(c_i|pa(c_i)) \tag{42.5}$$

  If $\underline{c}_i$ is evidence, replace $P(c_i|pa(c_i))$ by $\delta(c_i, c_i^*)$.

  Let

$$\underbrace{\underline{\mathcal{C}}_0}_{\mathcal{D}_0}, \underbrace{\underline{d}_1, \underline{\mathcal{C}}_1}_{\underline{\mathcal{D}}_1}, \underbrace{\underline{d}_2, \underline{\mathcal{C}}_2}_{\underline{\mathcal{D}}_2}, \ldots, \underbrace{\underline{\mathcal{D}}_{nd}, \underline{\mathcal{C}}_{nd}}_{\mathcal{D}_{nd}} \tag{42.6}$$

  and $\mathcal{D}_{<i} = \cup_{j=1}^{i-1} \mathcal{D}_j$.

  For $i = 1, 2, \ldots, nd$, let

$$\max_{\mathcal{D}_i} = \max_{d_i} \sum_{\mathcal{C}_i} \tag{42.7}$$

$$\operatorname*{argmax}_{\mathcal{D}_i} = \operatorname*{argmax}_{d_i} \sum_{\mathcal{C}_i} \tag{42.8}$$

$$\Psi_i^*(\mathcal{D}_{<i}) = \max_{\mathcal{D}_i} \underbrace{\max_{\mathcal{D}_{i+1}} \ldots \max_{\mathcal{D}_{nd}} P(c.|d.)u(c., d.)}_{\Psi_{i+1}^*(\mathcal{D}_{<i+1})} \tag{42.9}$$

$$\Delta_i^*(\mathcal{D}_{<i}) = \underset{\mathcal{D}_i}{\text{argmax}} \underbrace{\max_{\mathcal{D}_{i+1}} \ldots \max_{\mathcal{D}_{nd}} P(c.|d.)u(c., d.)}_{\Psi_{i+1}^*(\mathcal{D}_{<i+1})} \tag{42.10}$$

The **optimum policy** is defined by

$$\Delta^*. = (\Delta_1^*, \Delta_2^*, \ldots, \Delta_{nd}^*) \tag{42.11}$$

and the **Maximum Expected Utility** (MEU) by

$$MEU = \sum_{\mathcal{C}_0} \Psi_1(\mathcal{C}_0) \tag{42.12}$$

Thus, for example, if $nd = 3$, we have 3 max-expectation problems to be performed in the following order:

$$\begin{cases} 1) & \Psi_3(\mathcal{D}_{<3}) = P(c.|d.)u(c., d.) \\ 2) & \Psi_2(\mathcal{D}_{<2}) = \max_{\mathcal{D}_2} \Psi_3(\mathcal{D}_{<3}) \\ 3) & \Psi_1(\mathcal{C}_0) = \Psi_1(\mathcal{D}_{<1}) = \max_{\mathcal{D}_1} \Psi_2(\mathcal{D}_{<2}) \\ 4) & MEU = \sum_{\mathcal{C}_0} \Psi_1(\mathcal{C}_0) \end{cases} \tag{42.13}$$

Note that $\Psi_i()$ at time i depends solely on the decisions $\mathcal{D}_{<i}$ in its past.

On first encountering this algorithm, the reader might question how to do the maximization over d_i without knowing $\mathcal{D}_{<i}$. It turns out that due to d-separation, all you have to do is pick one instatiation and stick with it throughout when needed. The result for the optimum policy and MEU will be independent of the instantiation you chose. This is clear if you think of $\underline{\mathcal{D}}_0 \rightarrow \underline{\mathcal{D}}_1 \rightarrow \cdots \rightarrow \underline{\mathcal{D}}_{nd}$ as a Markov chain. When you maximize over d_i you hold it fixed, and $P(\mathcal{C}_i|d_i, \mathcal{D}_{<i}) = P(\mathcal{C}_i|d_i)$.

- **Limited Memory ID (LIMID)**.

  LIMIDs were first introduced in Ref.[39]. According to that reference, LIMIDs are "multistage decision problems in which the traditional assumption of no forgetting is relaxed." Note that "multistage decision problems" is another term for a dynamic bnet (see Chapter 26).

  Here is an example of LIMIDs. Figs.42.2 and 42.3 come from Ref.[39] and describe pig farming at various times $T = \{1, 2, 3, 4\}$.

  Node $\underline{h}_i \in \{0, 1\}$ refers to the health of a pig at time i.

  Node $\underline{t}_i \in \{0, 1\}$ refers to a test conducted on the pig at time i.

  Node $\underline{d}_i \in \{0, 1\}$ refers to a decision made by the farmer at time i on whether to medicate the pig.

Node $\underline{u}_i \in \mathbb{R}$ refers to the utility (monetary value) of medicating the pig at time i.

Fig.42.2 shows a dynamic bnet that obeys the **no forgetting** assumption so every decision is privy to the state of all previous decision and some chance nodes. Fig.42.3 shows the same bnet as Fig.42.2 after all arrows entering a decision $\underline{d}_i$ and originating at an earlier time, have been removed.

Note. The full previous treatment and test history is available when decisions must be taken.

Figure 42.2: Pig farming ID that obeys the "no-forgetting" assumption.

Note. Only the current test result is available when decisions are taken.

Figure 42.3: Pig farming ID that obeys the LIMID assumption. This ID is the same as the ID of Fig.42.2, except that the arrows pointing to a decision and originating at a previous time, have been removed.

- **Finding optimal strategy**

Ref.[77] proposes the following 3 exact methods for finding the optimal strategy of an ID.

1. conversion to a decision tree
2. calculation directly from ID, using Variable Elimination Algorithm (VEA)
3. conversion to a rooted cluster tree.

All 3 methods require reversing some arrows. To reverse an arrow, $\underline{x} \to \underline{y}$, we add dummy arrows going into $\underline{x}$ or into $\underline{y}$ so that $pa(\underline{x}) = pa(\underline{y}) = \mathcal{C}$. After that, we use Bayes rule conditioned on $\mathcal{C}$.

$$P(x|y,\mathcal{C}) = \frac{P(y|x,\mathcal{C})P(x|\mathcal{C})}{\sum_x \text{ numerator}} \tag{42.14}$$

In the next section, we discuss method 2 (VEA).

42.2 Variable Elimination Algorithm (VEA)

VEA utilizes the following operations:

1. Summing over barren (i.e., childless) chance nodes.
2. Reversing $\underline{x} \to \underline{y}$ to $\underline{y} \to \underline{x}$ using Bayes Rule.

(42.15)

3. Summing over the states of a chance node. Suppose $\underline{x} \to \underline{u}$ where $\underline{x}$ is a chance node and $\underline{u}$ is a utility node, and $\underline{x}$ has only one child $\underline{u}$. Then

(42.16)

4. Finding state of decision node that maximizes utility. Suppose $\underline{d} \to \underline{u}$ where $\underline{d}$ is a decision node and $\underline{u}$ is a utility node, and $\underline{d}$ has only one child $\underline{u}$. Also $\underline{a}. \cup \underline{b}. = pa(\underline{d}) \supset pa(\underline{u}) - \underline{d} = \underline{b}..$[3]. So the maximization is $\max_d u(d(b.,a.),b.)$. The value $d^* = \underset{d}{\operatorname{argmax}}\, u$ should be saved if the optimum policy is desired.

[3]This means that when you condition on $pa(\underline{d})$ you also condition on $pa(\underline{u}) - \underline{d}$

(42.17)

5. Combining partial utilities.

(42.18)

VEA consists of the following steps applied in non-sequential order. VEA is valid even with non-forgetting is assumed.

1. If there is a barren chance node $\underline{c}$, sum over it.
2. If $\underline{d} \to \underline{u}$ where $\underline{d}$ is a decision node and $\underline{u}$ is a utility node, and $\underline{d}$ has exactly one child $\underline{u}$, and all $pa(\underline{u}) - \underline{d}$ have been observed already, find $\max_d$ and remove $\underline{d}$.
3. If $\underline{c} \to \underline{u}$, where $\underline{c}$ is a chance node and $\underline{u}$ is a utility node, and $\underline{c}$ has at most one child, then, after reversing arcs to its non-utility children in reverse chronological order, sum over $\underline{c}$ and remove it.
4. Merge utility nodes $\underline{u}_1$ and $\underline{u}_2$, preferably when $pa(\underline{u}_1) \subset pa(\underline{u}_2)$.

We end this section with an example of VEA.[4]

(42.19)

[4]Note that the nodes of these IDs are not underlined. This is intentional. Rather than representing a network of abstract random variables, they represent an instantiation of that network. (see Chapter F)

(42.20)

(42.21)

(42.22)

(42.23)

Chapter 43

Instrumental Inequality and beyond

This chapter is based on Refs. [17] and [58].

Instrumental Variables (IVs) are discussed in Chapter 44. This chapter will discuss the original Instrumental inequality (I-inequality) discovered by Pearl, and other related inequalities. The I-inequality arises in bnets that use an IV. The I-inequality bounds the effect that an IV $\underline{z}$ can have on the outcome $\underline{y}$ of a treatment $\underline{d} \to \underline{y}$. Since there is a path $\underline{z} \to \underline{d} \to \underline{y}$, the treatment dose $\underline{d}$ acts as a mediator between the IV $\underline{z}$ and the treatment outcome $\underline{y}$. The I-inequality is reminiscent of the data processing inequality $H(\underline{z} : \underline{y}) \leq H(\underline{d} : \underline{y})$ which is valid for a simple Markov chain bnet $\underline{z} \to \underline{d} \to \underline{y}$. The data processing inequality is saying that the endpoint $\underline{y}$ receives more information from $\underline{d}$ than from $\underline{z}$. This is reasonable, since $\underline{y}$ is "closer" to $\underline{d}$ than to $\underline{z}$.

43.1 I-inequality

Figure 43.1: In bnet G, an IV $\underline{z}$ acts on a treatment $\underline{d} \to \underline{y}$. Bnet $\widetilde{G}$ is obtained by applying an imagine operator to arrow $\underline{d} \to \underline{y}$ of bnet G.

Claim 75 *The TPMs for the bnet G in Fig.43.1 satisfy*

$$\max_d \sum_y \max_z P(d,y|z) \leq 1 \tag{43.1}$$

proof:

Below, any probability that alludes to a value $\widetilde{d}$ refers to bnet $\widetilde{G}$. Otherwise, if it doesn't allude to $\widetilde{d}$, then it refers to G (or to $\widetilde{G}$, since the TPMs of $\widetilde{G}$ are defined from those of G in a consistent manner.)

G satisfies

$$P(d,y|z) = \sum_u P(u)P(y|u,d)P(d|u,z) \ , \tag{43.2}$$

and $\widetilde{G}$ satisfies

$$P(d,y|z,\widetilde{d}) = \sum_u P(u)P(y|u,\widetilde{d})P(d|u,z) \ . \tag{43.3}$$

Note that Eqs.(43.2) and (43.3) imply that

$$P(d,y|z,d) = P(d,y|z) \tag{43.4}$$

and that

$$\boxed{P(\widetilde{d},y|z,\widetilde{d}) \leq \sum_d P(d,y|z,\widetilde{d}) = P(y|\widetilde{d})} \ . \tag{43.5}$$

Thus,

$$\begin{aligned}
\max_{\widetilde{d}} \sum_y \max_z P(\widetilde{d},y|z,\widetilde{d}) &\leq \max_{\widetilde{d}} \sum_y \max_z P(y|\widetilde{d}) && (43.6)\\
&\leq \max_{\widetilde{d}} \sum_y P(y|\widetilde{d}) && (43.7)\\
&\leq \max_{\widetilde{d}} 1 && (43.8)\\
&\leq 1 && (43.9)
\end{aligned}$$

QED

As pointed out in Ref.[17] from which I learned the above proof, the above proof is highly generalizable.

Fig.43.2 gives a graphical representation of the boxed Eq.(43.5) which is crucial to the proof.

And here is a meta-description of the steps in the proof:

1. Use imagine operator to create a non-negative matrix $M_{d,\widetilde{d}}$.
2. Use fact that row or column sum of $M_{d,\widetilde{d}}$ is larger than diagonal element in sum: $\sum_d M_{d,\widetilde{d}} \geq M_{\widetilde{d},\widetilde{d}}$.

Figure 43.2: Graphical representation of the boxed equation Eq.(43.5).

43.1.1 I-inequality for binary z,d,y

It is enlightening to write down the I-inequality for the special case that $\underline{z}, \underline{d}, \underline{y}$ are binary.

$$P(d=1, y|z) = \begin{array}{c|c|c} & z=0 & z=1 \\ \hline y=0 & A & B \\ \hline y=1 & C & D \end{array}$$

$$A + D \leq 1$$
$$B + C \leq 1$$

Figure 43.3: I-inequality for binary $\underline{z}, \underline{d}, \underline{y}$. The same picture except with $d = 0$ is also true.

In the binary case, the I-inequality implies 4 different inequalities. These are as follows. One gets two inequalities by setting $d = 1$ in the next 2 equations.

$$\sum_{y=0}^{1} \sum_{z=0}^{1} \mathbb{1}(y = z) P(d, y|z) \ , \tag{43.10a}$$

$$\sum_{y=0}^{1} \sum_{z=0}^{1} \mathbb{1}(y \neq z) P(d, y|z) \ . \tag{43.10b}$$

One gets an additional 2 inequalities by setting $d = 0$ in Eqs.(43.10). These 4 inequalities are illustrated in Fig.43.3.

What do they mean? That at fixed $\underline{d}$, the correlation between $\underline{z}$ and $\underline{y}$ is limited.

43.2 Bounds on Effect of IV on treatment outcome y

Figure 43.4: Bnet $\mathcal{G}$ is obtained from the bnet G in Fig.43.1 by adding to G an arrow from the IV $\underline{z}$ to the treatment outcome $\underline{y}$. Bnet $\mathcal{G}_{do}$ is obtained by applying a do operator to node $\underline{d}$ of $\mathcal{G}$. Bnet $\mathcal{G}_{im}$ is obtained by applying an imagine operator to arrow $\underline{d} \to \underline{y}$ of $\mathcal{G}$.

In this section, we will assume that random variables $\underline{z}, \underline{d}, \underline{y}$ are binary. Just as with the binary case of the I-inequality, we will find an inequality for each value of $\underline{d} \in \{0, 1\}$.

Below, we will use the following 3 shorthand notations:

$$P_{y|z}(d) = P(d, y|z) \ , \tag{43.11}$$

$$P_{|z}(d) = \sum_{y} P(d, y|z) \ , \tag{43.12}$$

and

$$\pi_{|z}(d) = 1 - P_{|z}(d) \ . \tag{43.13}$$

For the bnet $\mathcal{G}_{do}$ in Fig.43.4, define the IV effect at fixed $\mathcal{D}\underline{d} = \widetilde{d}$ by

$$IVE(\widetilde{d}) = P(y = 1|z = 1, \mathcal{D}\underline{d} = \widetilde{d}) - P(y = 1|z = 0, \mathcal{D}\underline{d} = \widetilde{d}) \ . \tag{43.14}$$

Claim 76 *The TPMs for the bnet $\mathcal{G}_{do}$ in Fig.43.4 satisfy*

$$\pi_{|0}(d) \leq \Big[IVE(d) - \{P_{1|1}(d) - P_{1|0}(d)\}\Big] \leq \pi_{|1}(d) \tag{43.15}$$

proof:

$$
\begin{aligned}
P(y|z, \mathcal{D}\underline{d} = \widetilde{d}) &= \sum_u P(u) P(y|u, z, \widetilde{d}) && (43.16)\\
&= \sum_u P(u) \sum_d P(d, y|u, z, \widetilde{d}) && (43.17)\\
&\geq \sum_u P(u) P(\widetilde{d}, y|u, z, \widetilde{d}) && (43.18)\\
&= \sum_u P(u) P(\widetilde{d}, y|u, z) && (43.19)\\
&= P_{y|z}(\widetilde{d}) && (43.20)
\end{aligned}
$$

Next note that $P(d, y|z, \widetilde{d}) \geq 0$, and $\sum_{d,y} P(d, y|z, \widetilde{d}) = 1$. If we write a table for $P(d, y|z, \widetilde{d})$ at fixed $z, \widetilde{d}$ with row and column indices (d, y), then a partial sum of the entries of that table must be ≤ 1:

$$
\sum_{d \neq \widetilde{d}} P(d, y|z, \widetilde{d}) + \underbrace{\sum_{y'} P(\widetilde{d}, y'|z, \widetilde{d})}_{P_{|z}(\widetilde{d})} \leq 1 \ . \tag{43.21}
$$

Using the definitions of $P_{|z}$ and $\pi_{|z}$, we can rewrite the last equation as

$$
\sum_{d \neq \widetilde{d}} P(d, y|z, \widetilde{d}) \leq \pi_{|z}(\widetilde{d}) \ . \tag{43.22}
$$

Next note that

$$
\begin{aligned}
P(y|z, \mathcal{D}\underline{d} = \widetilde{d}) &= \sum_u P(u) P(y|u, z, \widetilde{d}) && (43.23)\\
&= \sum_u P(u) \sum_d P(d, y|u, z, \widetilde{d}) && (43.24)\\
&= P(\widetilde{d}, y|z, \widetilde{d}) + \sum_{d \neq \widetilde{d}} P(d, y|z, \widetilde{d}) && (43.25)\\
&= P_{y|z}(\widetilde{d}) + \sum_{d \neq \widetilde{d}} P(d, y|z, \widetilde{d}) && (43.26)\\
&\leq P_{y|z}(\widetilde{d}) + \pi_{|z}(\widetilde{d}) \ . && (43.27)
\end{aligned}
$$

Hence,

$$
P_{y|z}(\widetilde{d}) \leq P(y|z, \mathcal{D}\underline{d} = \widetilde{d}) \leq P_{y|z}(\widetilde{d}) + \pi_{|z} \tag{43.28}
$$

$$
P_{1|1}(\widetilde{d}) \leq P(y = 1|z = 1, \mathcal{D}\underline{d} = \widetilde{d}) \leq P_{1|1}(\widetilde{d}) + \pi_{|1} \tag{43.29}
$$

$$-P_{1|0}(\widetilde{d}) - \pi_{|0} \leq -P(y = 1|z = 0, \mathcal{D}\underline{d} = \widetilde{d}) \leq -P_{1|0}(\widetilde{d}) \tag{43.30}$$

QED

Chapter 44

Instrumental Variables

This chapter is based on Refs.[15] and [146].

The theory of potential outcomes (PO) discussed in Chapter 77 assumes that confounders can be ignored by conditioning on them. However, there are cases when that is not possible, as when there are some unmeasured (i.e., unobserved, hidden) confounder nodes in the bnet, because one can only condition on observed random variables, by definition. So what if confounders can't be ignored? Are we then precluded from using PO theory? Not necessarily. It might still be possible to use PO theory if one can find a suitable instrumental variable (IV) for the problem.

IVs were actually invented by Sewall Wright and his father Philip Wright long before PO theory was invented by Rubin. The reason why IVs save PO theory is greatly clarified by using Pearl causal DAGs and his d-separation theorem (see Chapter 24).

Most of the discussion in this chapter is limited to LDEN (linear deterministic bnets with external noise). These are discussed in Chapter 52. However, as will become obvious to the reader, IVs are also applicable and useful in general bnet modeling.

44.1 δ with unmeasured confounder

In this section, we explain, using LDENs, why unmeasured confounders prejudice PO δ calculations.

Consider the LDEN bnet of Fig.44.1. For some $\delta, \mu \in \mathbb{R}$, we have

$$\underline{y} = \delta \underline{d} + \mu \underline{u} + \underline{\epsilon}_{\underline{y}} \ . \tag{44.1}$$

Note that

$$\left\langle \underline{d}, \underline{\epsilon}_{\underline{y}} \right\rangle = 0 \tag{44.2}$$

because the path from $\underline{d}$ to $\underline{\epsilon}_{\underline{y}}$ is blocked by a collider. Note also that $\langle \underline{d}, \underline{u} \rangle \neq 0$ because there is an unblocked path between $\underline{d}$ and $\underline{u}$. Hence

Figure 44.1: An LDEN bnet. The direct path $\underline{d} \to \underline{y}$ is confounded by a hidden variable $\underline{u}$. External root nodes $\underline{\epsilon_d}, \underline{\epsilon_u}, \underline{\epsilon_y}$ are left implicit.

$$\left\langle \underline{d}, \underline{y} \right\rangle = \delta \left\langle \underline{d}, \underline{d} \right\rangle + \mu \left\langle \underline{d}, \underline{u} \right\rangle \ . \tag{44.3}$$

If $\langle \underline{d}, \underline{u} \rangle$ were always zero, or if we could measure $\langle \underline{d}, \underline{u} \rangle$ (we can't because $\underline{u}$ is unobserved), we could solve for δ, but that is unfortunately not the case.

44.2 δ (with unmeasured confounder) can be inferred via IV

Figure 44.2: Two LDEN bnets. The direct path $\underline{d} \to \underline{y}$ is confounded by a hidden variable $\underline{u}$, but by using the IV $\underline{a}$, we are still able to identify (i.e., calculate) δ. External root nodes $\underline{\epsilon_a}, \underline{\epsilon_d}, \underline{\epsilon_u}, \underline{\epsilon_y}$ are left implicit.

Now consider the two LDEN bnets shown in Fig.44.2. Note that there are no arrows $\underline{a} \to \underline{y}$ or $\underline{a} \to \underline{u}$. Note that node $\underline{d}$ is a collider in the path $\underline{a} - \underline{d} - \underline{u} - \underline{y}$, Therefore, the only unblocked path from $\underline{a}$ to $\underline{y}$ in G is $\underline{a} \to \underline{d} \to \underline{y}$ and that path has been removed in G_{im+}. These observations are encapsulated in the following statements.

$$\underline{d} \perp_G \underline{y} = \text{false}, \quad \underline{a} \perp_G \underline{y} = \text{false} \ . \tag{44.4}$$

$$\underline{d} \perp_{G_{im+}} \underline{y} = \text{false}, \quad \underline{a} \perp_{G_{im+}} \underline{y} = \text{true} \ . \tag{44.5}$$

For G, the following is true.

$$\begin{cases} \underline{y} = \delta \underline{d} + \mu \underline{u} + \underline{\epsilon_y} \\ \underline{d} = \alpha \underline{a} + \nu \underline{u} + \underline{\epsilon_d} \end{cases} \tag{44.6}$$

Note that

$$\langle \underline{a}, \underline{u} \rangle = \left\langle \underline{a}, \underline{\epsilon_y} \right\rangle = \left\langle \underline{a}, \underline{\epsilon_d} \right\rangle = 0 \tag{44.7}$$

because in all cases, paths between the two nodes in the covariance are blocked by a collider. Therefore

$$\left\langle \underline{a}, \underline{y} \right\rangle = \delta \langle \underline{a}, \underline{d} \rangle \tag{44.8}$$

and

$$\langle \underline{a}, \underline{d} \rangle = \alpha \langle \underline{a}, \underline{a} \rangle \ . \tag{44.9}$$

Note that $\left\langle \underline{a}, \underline{y} \right\rangle = \delta = 0$ for G_{im+} but not for G, so we are speaking about G from here on. It follows that[1]

$$\alpha = \frac{\langle \underline{a}, \underline{d} \rangle}{\langle \underline{a}, \underline{a} \rangle} = \frac{\partial \underline{d}}{\partial \underline{a}} \tag{44.10}$$

and

$$\begin{aligned} \delta &= \frac{\left\langle \underline{a}, \underline{y} \right\rangle}{\langle \underline{a}, \underline{d} \rangle} && (44.11) \\ &= \frac{\left\langle \underline{a}, \underline{y} \right\rangle}{\langle \underline{a}, \underline{a} \rangle} \frac{\langle \underline{a}, \underline{a} \rangle}{\langle \underline{a}, \underline{d} \rangle} && (44.12) \\ &= \frac{\frac{\partial \underline{y}}{\partial \underline{a}}}{\frac{\partial \underline{d}}{\partial \underline{a}}} \quad (\neq \frac{\partial \underline{y}}{\partial \underline{d}}) && (44.13) \\ &= \frac{\partial \underline{y}}{\partial (\alpha \underline{a})} \ . && (44.14) \end{aligned}$$

Eq.(44.13) is illustrated in Fig.44.3.

44.3 More general bnets with IVs

Figs.44.4 and 44.5 are examples of other bnets for which the effect δ is identifiable thanks to the IV $\underline{a}$.

[1] As usual in this book, we define $\frac{\partial \underline{y}}{\partial \underline{x}} = \frac{\langle \underline{x}, \underline{y} \rangle}{\langle \underline{x}, \underline{x} \rangle}$ for any random variables $\underline{x}, \underline{y}$.

Figure 44.3: Effect δ as slope of line.

Figure 44.4: The 2 paths in G_{im+} from IV variable $\underline{a}$ to $\underline{y}$ are blocked by not conditioning on colliders $\underline{v}$ and $\underline{d}$. Thus, $\underline{d} \perp_{G_{im+}} \underline{y} =$ false, $\underline{a} \perp_{G_{im+}} \underline{y} =$ true.

Figure 44.5: There are 2 paths in G_{im+} from IV variable $\underline{a}$ to $\underline{y}$. One is blocked by not conditioning on the collider $\underline{d}$ and the other can be blocked by conditioning on $\underline{v}$. Thus, $\underline{d} \perp_{G_{im+}} \underline{y}|\underline{v} =$ false, $\underline{a} \perp_{G_{im+}} \underline{y}|\underline{v} =$ true.

44.4 Instrumental Inequality

Pearl's instrumental inequality and related inequalities are discussed in Chapter 43.

Chapter 45

Jackknife Resampling

This chapter is based on Ref. [148].

Before reading this chapter, we recommend that you read the section entitled "Demystifying Population and Sample Variances", in Chapter C.

Jackknife resampling (JR) is a statistical technique used to estimate the precision or bias of sample statistics, like the mean, variance, and standard error. It involves creating from a list of n samples, n new lists with $n-1$ samples each. The new lists are created by leaving out one different sample each time. One can then relate the statistics of those n new lists with the statistics of the original list. This technique is called JR because, like a jackknife, it has multiple applications, such as variance estimation, bias correction and influence analysis.

Let $\Sigma = \{0, 1, 2, \ldots, n-1\}$. Let us consider the list of samples

$$x^n = (x^\sigma)_{\sigma\in\Sigma} \ , \tag{45.1}$$

where the x^σ are assumed to be i.i.d. with

$$E[\underline{x}^\sigma] = \mu \tag{45.2}$$

and

$$\left\langle \underline{x}^\sigma, \underline{x}^{\sigma'} \right\rangle = V_1 \delta(\sigma, \sigma') \ . \tag{45.3}$$

If we define μ and V_1 estimators by

$$\widehat{\mu} = \frac{1}{n} \sum_\sigma x^\sigma \tag{45.4a}$$

$$\widehat{V}_1 = \frac{1}{n-1} \sum_\sigma (x^\sigma - \widehat{\mu})^2 \ , \tag{45.4b}$$

then one can show that:

$$E[\underline{\widehat{\mu}}] = \mu \ , \quad E[\underline{\widehat{V}}_1] = V_1 \tag{45.5}$$

so both of these estimators are unbiased.

Now define lists of samples with one of the items in x^n deleted:

$$x_\xi^{n-1} = (x^\sigma)_{\sigma\in\Sigma-\{\xi\}} \ . \tag{45.6}$$

Suppose we are given functions of x^n and x_ξ^{n-1}

$$A = \Phi(x^n; n) \ , \tag{45.7}$$

$$A_\xi = \Phi(x_\xi^{n-1}; n-1) \ . \tag{45.8}$$

Then define a list A^n by

$$A^n = (A_\xi)_{\xi\in\Sigma} \ . \tag{45.9}$$

One can also define a list B^n by:

$$B^n = (B_\xi)_{\xi\in\Sigma} \tag{45.10}$$

where

$$\begin{aligned} B_\xi &= nA - (n-1)A_\xi & (45.11)\\ &= A_\xi - n[A_\xi - A] \ . & (45.12)\end{aligned}$$

Later on, we will see why the list B^n is of interest.

Figure 45.1: Bnet for jackknife resampling (JR).

Fig.45.1 is a bnet that encapsulates JR. The TPMs, printed in blue, for that bnet, are as follows:

$$P(x_\xi^{n-1}|x^n) = \mathbb{1}(\quad x_\xi^{n-1} = \text{ defined by Eq.(45.6)} \quad) \tag{45.13}$$

$$P(A_\xi|x_\xi^{n-1}) = \mathbb{1}(\quad A_\xi = \text{ defined by Eq.(45.8)} \quad) \tag{45.14}$$

$$P(B_\xi|A_\xi, A) = \mathbb{1}(\quad B_\xi = \text{ defined by Eq.(45.12)} \quad) \tag{45.15}$$

45.1 Case $A = \Phi(x^n; n) = \frac{1}{n}\sum_\sigma x^\sigma$

Suppose

$$A = \underbrace{\frac{1}{n}\sum_\sigma x^\sigma}_{E_\sigma[x^\sigma]} \tag{45.16}$$

and

$$A_\xi = \underbrace{\frac{1}{n-1}\sum_{\sigma\in\Sigma-\{\xi\}} x^\sigma}_{E_{\sigma|\xi}[x^\sigma] \text{ where } P(\sigma|\xi)=\frac{\mathbb{1}(\sigma\neq\xi)}{n-1}} \quad . \tag{45.17}$$

Then

$$\frac{1}{n}\sum_\xi A_\xi = E_\xi E_{\sigma|\xi}[x^\sigma] = E_\sigma[x^\sigma] = A \tag{45.18}$$

Claim 77

$$E[\underline{A}_\xi] = \mu \tag{45.19}$$

$$\left\langle \underline{A}_\xi, \underline{A}_{\xi'} \right\rangle = V_1 \left[\frac{n-2-\delta(\xi,\xi')}{(n-1)^2}\right] \tag{45.20}$$

proof:

$$E[\underline{A}_\xi] = \frac{1}{n-1}\sum_\sigma \mathbb{1}(\xi\neq\sigma)E[\underline{x}^\sigma] = \mu \tag{45.21}$$

$$\begin{aligned}
\left\langle \underline{A}_\xi, \underline{A}_{\xi'} \right\rangle &= \frac{1}{(n-1)^2}\sum_\sigma\sum_{\sigma'}[1-\delta(\sigma,\xi)][1-\delta(\sigma',\xi')]\left\langle x^\sigma, x^{\sigma'}\right\rangle && (45.22)\\
&= \frac{V_1}{(n-1)^2}\sum_\sigma\sum_{\sigma'}[1-\delta_\xi^\sigma-\delta_{\xi'}^{\sigma'}+\delta_\xi^\sigma\delta_{\xi'}^{\sigma'}]\delta_{\sigma'}^\sigma && (45.23)\\
&= \frac{V_1}{(n-1)^2}[n-2+\delta_{\xi'}^\xi] && (45.24)
\end{aligned}$$

QED

Figure 45.2: For each point x^ξ, there is a corresponding point A_ξ which is $n-1$ times closer to the average A than x^ξ is.

Claim 78

$$A_\xi - A = \frac{A - x^\xi}{n-1} \tag{45.25}$$

Hence, the distance of a point x^ξ to the mean value A is $n-1$ times as large as the distance of A_ξ to A. (see Fig.45.2)

proof:

$$\begin{aligned} A_\xi - A &= \frac{1}{n-1}\left(\sum_\sigma x^\sigma - x^\xi\right) - \frac{1}{n}\sum_\sigma x^\sigma && (45.26)\\ &= x^\xi\left(\frac{-1}{n-1}\right) + \sum_\sigma x^\sigma \underbrace{\left(\frac{1}{n-1} - \frac{1}{n}\right)}_{\frac{1}{n(n-1)}} && (45.27)\\ &= \frac{A - x^\xi}{n-1} && (45.28) \end{aligned}$$

QED

Note that

$$(n-1)\sum_\xi (A_\xi - E_\xi[A_\xi])^2 = \widehat{V}_1 \tag{45.29}$$

by Eq.(45.25). Hence, we can estimate V_1 from A^n instead of x^n.

Note that

$$\begin{aligned} B_\xi &= nA - (n-1)A_\xi && (45.30)\\ &= \sum_\sigma x^\sigma - \sum_{\sigma\neq\xi} x^\sigma = x^\xi && (45.31) \end{aligned}$$

Since $B_\xi = x^\xi$, they have identical statistics. In particular, one can use for B_ξ the same μ and V_1 estimators that we defined in Eqs.(45.4) for x^σ.

Chapter 46

Junction Tree Algorithm

The Junction Tree (JT) algorithm is an algo for evaluating exact marginals of a bnet, including cases in which some nodes are fixed to a single state. (fixed nodes are called the a priori evidence.)

The JT algo starts by clustering the loops of a bnet into bigger nodes so as to transform the bnet into a polytree bnet. Then it applies Pearl Belief Propagation (see Chapter 60) to the ensuing polytree. The first breakthrough paper to achieve this agenda in full was Ref.[40] by Lauritzen, and Spiegelhalter in 1988. See the Wikipedia article Ref.[149] for more info and references on the JT algorithm.

I won't describe the JT algo any further here, because it would take too long for this brief book to give a complete treatment of it, including the mathematical proofs. If all you want to do is to code the JT algo, without delving into the mathematical theorems and proofs behind it, I strongly recommend Ref.[32]. Ref.[32] is an excellent cookbook for programmers of the JT algo. My open source program QuantumFog (see Ref.[94]) implements the JT algo in Python, following the recipe of Ref.[32].

Chapter 47

Kalman Filter

This chapter is based on Ref.[151], except we've replaced the variables F_t, w_t, v_t in that reference by A_t, ξ_t, ζ_t, respectively.

A Kalman Filter (KF) is a special case of a Hidden Markov Model. HMMs are discussed in Chapter 40.

Figure 47.1: Kalman Filter (KF) bnet.

Let

$t = 0, 1, 2, \ldots, T-1$ be the time.

$\underline{\xi}_t \in \mathbb{R}^{nx}, \underline{\zeta}_t \in \mathbb{R}^{nz}$ be random variables that represent hidden (unobserved) external Gaussian white noise.

$\underline{x}_t \in \mathbb{R}^{nx}$ be random variables that represent the hidden (unobserved) true state of the system.

$\underline{u}_t \in \mathbb{R}^{nu}, \underline{z}_t \in \mathbb{R}^{nz}$ be random variables that represent the measured (observed) state of the system.

The TPMs, printed in blue, for the KF bnet Fig.47.1, are as follows:

$$P(\xi_t) = \mathcal{N}(\xi_t; 0, Q_t) \ , \tag{47.1}$$

where Q_t is given.

$$P(x_t|x_{t-1}, u_t, \xi_t) = \mathbb{1}(x_t = A_t x_{t-1} + B_t u_t + \xi_t) \ , \tag{47.2}$$

where $A_t, B_t u_t$ are given. $P(x_t|x_{t-1}, u_t, \xi_t)$ becomes $P(x_t)$ for $t = 0$.

$$P(\zeta_t) = \mathcal{N}(\zeta_t; 0, R_t) \ , \tag{47.3}$$

where R_t is given.

$$P(z_t|x_t, \zeta_t) = \mathbb{1}(z_t = H_t x_t + \zeta_t) \ , \tag{47.4}$$

where H_t is given.

47.1 Prediction Problem

Find $\widehat{x}_t$ (the best possible estimate of x_t) and P_t (the state of the filter at time t) in terms of

1. $\widehat{x}_{t-1}$ and P_{t-1}.
2. the 5 matrices $\mathcal{M}_t = (A_t, B_t, H_t, Q_t, R_t)$
3. the observed values of z_t and u_t.

See Fig.47.2. For that figure,

$$P(\widehat{x}_t, P_t|\widehat{x}_{t-1}, P_{t-1}, \mathcal{M}_t, z_t, u_t) = \delta(\widehat{x}_t, ?)\delta(P_t, ?) \ . \tag{47.5}$$

$$\begin{tikzcd}
\mathcal{M}_0, \underline{z}_0, \underline{u}_0 \arrow[d] & \mathcal{M}_1, \underline{z}_1, \underline{u}_1 \arrow[d] & \mathcal{M}_2, \underline{z}_2, \underline{u}_2 \arrow[d] & \mathcal{M}_3, \underline{z}_3, \underline{u}_3 \arrow[d] \\
\underline{\widehat{x}}_0, \underline{P}_0 \arrow[r] & \underline{\widehat{x}}_1, \underline{P}_1 \arrow[r] & \underline{\widehat{x}}_2, \underline{P}_2 \arrow[r] & \underline{\widehat{x}}_3, \underline{P}_3
\end{tikzcd}$$

Figure 47.2: Evolution of $\widehat{x}_t, P_t$ for a KF.

Figure 47.3: Bnet representation of the algebraic solution of the prediction problem for a KF.

47.2 Solution

The algebraic solution of the prediction problem for a KF is as follows. See Fig.47.3 for a bnet representation of this algebraic solution.

Define $\eta_{t|t} = \eta_t$ for $\eta = \widehat{x}, P$.

- **Initial Conditions** $\widehat{x}_0, P_0$
- **A priori estimates**

  a priori state estimate

$$\widehat{x}_{t|t-1} = \underbrace{A_t\widehat{x}_{t-1} + B_t u_t}_{x_t - \xi_t} \tag{47.6}$$

  a priori covariance estimate

$$P_{t|t-1} = A_t P_{t-1} A_t^T + Q_t \tag{47.7}$$

- **A posteriori estimates**

  Optimal Kalman gain K_t

$$S_t = H_t P_{t|t-1} H_t^T + R_t \tag{47.8}$$

$$
\begin{aligned}
K_t &= P_{t|t-1} H_t^T S_t^{-1} && (47.9)\\
&= P_{t|t-1} H_t^T \left[H_t P_{t|t-1} H_t^T + R_t \right]^{-1} && (47.10)
\end{aligned}
$$

a posteriori state estimate

$$
\widetilde{y}_{t|t-1} = z_t - H_t \widehat{x}_{t|t-1} \tag{47.11}
$$

$$
\begin{aligned}
\widehat{x}_t &= \widehat{x}_{t|t-1} + K_t \widetilde{y}_{t|t-1} && (47.12)\\
&= (1 - K_t H_t) \widehat{x}_{t|t-1} + K_t \underbrace{z_t}_{H_t x_t + \zeta_t} \quad \text{(interpolation formula)} && (47.13)
\end{aligned}
$$

$$
\widetilde{y}_t = z_t - H_t \widehat{x}_t \tag{47.14}
$$

a posteriori covariance estimate

$$
P_t = (I - K_t H_t) \, P_{t|t-1} \tag{47.15}
$$

47.3 Simple Example

r_t position, v_t velocity, a_t acceleration of point particle.

$$
x_t = A x_{t-1} + B u_t + \xi_t \tag{47.16}
$$

$$
x_t = \begin{bmatrix} r_t \\ v_t \end{bmatrix} , \quad u_t = a_t \tag{47.17}
$$

$$
A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} , \quad B = \begin{bmatrix} \frac{1}{2}(\Delta t)^2 \\ \Delta t \end{bmatrix} \tag{47.18}
$$

$$
z_t = H x_t + \zeta_t \tag{47.19}
$$

$$
H = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{47.20}
$$

47.4 Invariants

Note that

$$x_t - \widehat{x}_{t|t-1} = \xi_t \tag{47.21}$$

$$\begin{aligned} x_t - \widehat{x}_t &= (1 - K_t H_t)(x_t - \widehat{x}_{t|t-1}) - K_t \zeta_t & (47.22) \\ &= (1 - K_t H_t)\xi_t - K_t \zeta_t & (47.23) \end{aligned}$$

$$\begin{aligned} \widetilde{y}_{t|t-1} &= \underbrace{z_t}_{H_t x_t + \zeta_t} - H_t \widehat{x}_{t|t-1} & (47.24) \\ &= H_t \xi_t + \zeta_t & (47.25) \end{aligned}$$

$$\begin{aligned} \widetilde{y}_t &= \underbrace{z_t}_{H_t x_t + \zeta_t} - H_t \widehat{x}_t & (47.26) \\ &= H_t(x_t - \widehat{x}_t) + \zeta_t & (47.27) \\ &= H_t(1 - K_t H_t)\xi_t + (1 - H_t K_t)\zeta_t & (47.28) \end{aligned}$$

$(x_t - \widehat{x}_{t|t-1})$, $(x_t - \widehat{x}_t)$, $\widetilde{y}_{t|t-1}$ and $\widetilde{y}_t$ are called **residuals**. Since ξ_t and ζ_t have zero mean value,

$$E[\underline{x}_t - \underline{\widehat{x}}_t] = E[\underline{x}_t - \underline{\widehat{x}}_{t|t-1}] = 0 \tag{47.29}$$

$$E[\underline{\widetilde{y}}_t] = E[\underline{\widetilde{y}}_{t|t-1}] = 0 \tag{47.30}$$

These zero mean value identities are called **invariants**.

47.5 Derivation of Solution

First, some notational conventions. Let

$$Cov(\underline{a})_{i,j} = \left\langle \underline{a}_i, \underline{a}_j \right\rangle \tag{47.31}$$

$$Cov(\underline{a}) = \left\langle \underline{a}, \underline{a}^T \right\rangle = \langle \underline{a}, tp. \rangle \tag{47.32}$$

$$[A, B]_+ = AB + B^T A^T \tag{47.33}$$

$$A + tp. = A + A^T \tag{47.34}$$

tp. stands for transpose.

Now define

$$P_t = Cov(\underline{x}_t - \underline{\widehat{x}}_t) \tag{47.35}$$

$$P_{t|t-1} = Cov(\underline{x}_t - \underline{\widehat{x}}_{t|t-1}) \tag{47.36}$$

$$S_t = Cov(\underline{\widetilde{y}}_{t|t-1}) \tag{47.37}$$

It follows that

$$\begin{aligned}
P_t &= \langle \underline{x}_t - \underline{\widehat{x}}_t, tp. \rangle && (47.38)\\
&= \left\langle (1 - K_t H_t)(\underline{x}_t - \underline{\widehat{x}}_{t|t-1}) - K_t \underline{\zeta}_t, tp. \right\rangle \quad \text{(by Eq.(47.23))} && (47.39)\\
&= \left\langle (1 - K_t H_t)(\underline{x}_t - \underline{\widehat{x}}_{t|t-1}), tp. \right\rangle + \left\langle K_t \underline{\zeta}_t, tp. \right\rangle \ (\underline{\zeta}_t \text{ uncorrelated with } (\underline{x}_t - \underline{\widehat{x}}_{t|t-1})) && (47.40)\\
&= (1 - K_t H_t) \underbrace{\left\langle \underline{x}_t - \underline{\widehat{x}}_{t|t-1}, tp. \right\rangle}_{P_{t|t-1}} (1 - K_t H_t)^T + K_t \underbrace{\left\langle \underline{\zeta}_t, tp. \right\rangle}_{R_t} K_t^T && (47.41)\\
&= P_{t|t-1} - [K_t H_t, P_{t|t-1}]_+ + K_t \underbrace{\left(H_t P_{t|t-1} H_t^T + R_t \right)}_{S_t} K_t^T && (47.42)
\end{aligned}$$

Next we find the optimal Kalman gain K_t by minimizing with respect to K_t the following mean squared error.

$$\begin{aligned}
\mathcal{E} &= \sum_i E[(\underline{x}_t - \underline{\widehat{x}}_t)_i^2] && (47.43)\\
&= \mathrm{tr} E\left[(\underline{x}_t - \underline{\widehat{x}}_t)(\underline{x}_t - \underline{\widehat{x}}_t)^T \right] && (47.44)\\
&= \mathrm{tr} P_t && (47.45)\\
&= \mathrm{tr}\left(P_{t|t-1} - [K_t H_t, P_{t|t-1}]_+ + K_t S_t K_t^T \right) && (47.46)
\end{aligned}$$

If we set to zero the variation of $\mathcal{E}$ when K_t varies, we get

$$0 = \delta\mathcal{E} = \mathrm{tr}\left[\left(-P_{t|t-1} H_t^T + K_t S_t \right) \delta K_t^T \right] + tp. \tag{47.47}$$

Hence

$$-P_{t|t-1} H_t^T + K_t S_t = 0 \tag{47.48}$$

$$K_t = P_{t|t-1} H_t^T S_t^{-1} \tag{47.49}$$

Chapter 48

Kernel Principal Component Analysis (Kernel PCA)

This chapter is based on Ref.[154].

Kernel Principal Component Analysis (Kernel PCA) applies to a special type of kernel that can be defined in terms of a nonlinear transformation $\Phi : \mathbb{R}^{na} \to \mathbb{R}^N$ where usually $N > na$. The idea is that a cloud of data points $\vec{x} \in \mathbb{R}^{na}$ might look very complicated in $\mathbb{R}^{na}$ but when mapped to $\mathbb{R}^N$ as $\Phi(\vec{x}) \in \mathbb{R}^N$, it might look like a thin ellipse (a line wannabe). Kernel PCA is thus a method of dimensionality increase instead of what we called in Chapter 21, dimensionality reduction.

Henceforth, we will use a vector sign as in $\vec{x}$ to denote vectors in $\mathbb{R}^{na}$. We will use kets as in $|\Phi\rangle = \Phi \in \mathbb{R}^{N\times 1}$ and bras as in $\langle\Phi| = \Phi^T \in \mathbb{R}^{1\times N}$ to denote column vectors and row vectors in $\mathbb{R}^N$.

Let

$\vec{x}_\sigma \in \mathbb{R}^{na}$ for $\sigma = 1, 2, \ldots N$

$\Phi : \mathbb{R}^{na} \to \mathbb{R}^N$. Φ can be a matrix $A \in \mathbb{R}^{N\times na}$ so that $\Phi(\vec{x}_\sigma) = A\vec{x}_\sigma \in \mathbb{R}^N$, but the most advantageous use of this kernel arises when $\Phi(\cdot)$ is a nonlinear map.

A **"Kernel trick" kernel** $K(\vec{x}_\sigma, \vec{x}_\rho)$ is expressible as

$$K(\vec{x}_\sigma, \vec{x}_\rho) \quad = \quad \langle\Phi(\vec{x}_\sigma)|\Phi(\vec{x}_\rho)\rangle \tag{48.1}$$

Define a Kernel matrix $K \in R^{N\times N}$ with entries

$$K_{\sigma,\rho} = K(\vec{x}_\sigma, \vec{x}_\rho) \tag{48.2}$$

and a correlation matrix $C \in \mathbb{R}^{N\times N}$ by

$$C = \frac{1}{N}\sum_\sigma |\Phi(\vec{x}_\sigma)\rangle\langle\Phi(\vec{x}_\sigma)| \tag{48.3}$$

The **eigenvalue problem for C** is

$$C|v_j\rangle = \lambda_j|v_j\rangle \tag{48.4}$$

for $j = 1, 2, \ldots, na$.

We have

$$\langle \Phi(\vec{x_\sigma})|C|v_j\rangle = \lambda_j \langle \Phi(\vec{x_\sigma})|v_j\rangle \tag{48.5}$$

We can express each eigenvector $|v_j\rangle$ as a linear combination of $\{|\Phi(\vec{x}_\rho)\rangle\}_{\rho=1}^N$

$$|v_j\rangle = \sum_\rho A_{j,\rho}|\Phi(\vec{x_\rho})\rangle \tag{48.6}$$

Let

$$|A_j\rangle_\sigma = A_{j,\sigma} \tag{48.7}$$

Then

$$\boxed{\langle \Phi(\vec{x_\sigma})|v_j\rangle} = \sum_\rho A_{j,\rho} K_{\sigma,\rho} \tag{48.8}$$

$$= \boxed{(K|A_j\rangle)_\sigma = \langle \sigma|K|A_j\rangle} \tag{48.9}$$

$$\sum_\rho A_{j,\rho} \underbrace{\langle \Phi(\vec{x_\sigma})|C|\Phi(\vec{x_\rho})\rangle}_{\frac{1}{N}K^2_{\sigma,\rho}} = \lambda_j \sum_\sigma A_{j,\sigma} \underbrace{\langle \Phi(\vec{x_\sigma})|\Phi(\vec{x_\rho})\rangle}_{K_{\sigma,\rho}=K_{\rho,\sigma}} \tag{48.10}$$

Eq.(48.10) can be expressed as

$$K^2|A_j\rangle = N\lambda_j K|A_j\rangle \tag{48.11}$$

The **eigenvalue problem for K** is

$$K|A_j\rangle = N\lambda_j|A_j\rangle \tag{48.12}$$

for $j = 1, 2, \ldots, na$. Note that the basis $\{|A_j\rangle\}_{j=1}^{na}$ is orthogonal, as expected, and the basis $\{\sqrt{N\lambda_j}|A_j\rangle\}_{j=1}^{na}$ is orthonormal.

$$\delta_{i,j} = \langle v_i|v_j\rangle \tag{48.13}$$

$$= \sum_\rho \sum_\sigma A_{i,\sigma} A_{j,\rho} K_{\sigma,\rho} \tag{48.14}$$

$$= \langle A_i|K|A_j\rangle \tag{48.15}$$

$$= N\lambda_j \langle A_i|A_j\rangle \tag{48.16}$$

Examples

- Radial basis function

$$K(\vec{x}, \vec{y}) = \exp\left(-\frac{\| \vec{x} - \vec{y} \|^2}{2\sigma^2}\right) \tag{48.17}$$

where $\sigma \in \mathbb{R}^{>0}$.

- Sigmoid kernel

$$K(\vec{x}, \vec{y}) = \tanh(a\vec{x} \cdot \vec{y} + b) \tag{48.18}$$

where $a, b \in \mathbb{R}$.

- Polynomial of order d

$$K(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y})^d \tag{48.19}$$

Consider $d = 2$, $na = 2$, $N = 3$.

Claim 79 *If*

$$\Phi(\vec{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \tag{48.20}$$

then

$$(\vec{x} \cdot \vec{y})^2 = \langle \Phi(\vec{x}) | \Phi(\vec{y}) \rangle \tag{48.21}$$

proof:

$$\begin{aligned}
(\vec{x} \cdot \vec{y})^2 &= (x_1y_1 + x_2y_2)^2 && (48.22)\\
&= (x_1^2, x_2^2, \sqrt{2}x_1x_2) \cdot (y_1^2, y_2^2, \sqrt{2}y_1y_2) && (48.23)\\
&= \langle \Phi(\vec{x}) | \Phi(\vec{y}) \rangle && (48.24)
\end{aligned}$$

QED

The map defined by Eq.(48.20) is illustrated in Fig.48.1.

Figure 48.1: The map $\Phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ maps a ring in $\mathbb{R}^2$ to a line in $\mathbb{R}^3$.(plots from [41])

Chapter 49

LATE (Local Average Treatment Effect)

This chapter is based on Refs.[50, 18].

The **Local Average Treatment Effect (LATE)** is the same as the ATE estimand[1], but it only counts "compliers" (i.e., individuals that comply with the treatment they've been assigned). LATE assumes the same bnet that we considered when we discussed Instrumental Variables (IV) [2]

σ	a^σ	$d^\sigma(a^\sigma = 0)$	$d^\sigma(a^\sigma = 1)$	$y^\sigma(d^\sigma = 0)$	$y^\sigma(d^\sigma = 1)$
1	0	1			0
2	0	0		1	
3	0	1			1
4	0	0		0	
5	0	0		1	
6	1		1		0
7	1		0	0	
8	1		1		0
9	1		0	1	
10	1		1		1

Table 49.1: Hypothetical dataset for LATE problem. Pink cells indicate missing data.

Table 49.1 shows a hypothetical dataset for the LATE problem.

Consider the bnets G and G_+ of Fig.49.1.[3]

[1] ATE is defined in the Potential Outcomes chapter, i.e., Chapter 77.

[2] Instrumental Variables are discussed in Chapters 44 and 43.

[3] If there were an arrow $\underline{a} \to \underline{y}$ in bnet G, then we could also apply the imagine operator $\mathcal{I}_{\underline{a}\to\underline{y}}$ to G. This would lead to nodes $[\underline{y}(a', d')]^1_{a'=0}|^1_{d'=0}$. Since there is no arrow $\underline{a} \to \underline{y}$, we would have $y(a, d) = y(d)$ for all a, d, i.e., y does not depend directly on the instrument a. This independence of $y(a, d)$ on a is called the **excludability assumption**. One could say the excludability assumption is built into our bnet G_+.

Figure 49.1: Bnet G_+ is bnet G after application of 2 imagine operators. Imagine operators are discussed in Chapter 12.

Let
$\underline{a} \in \{0, 1\}$, instrumetal variable, initially assigned dose
$\underline{d} \in \{0, 1\}$, actual treatment dose
$\underline{y} \in \{0, 1\}$, treatment outcome
The definition of the imagine operators used in G_+ stipulates that nodes $\underline{y}$ and $\underline{d}$ in G_+ must be have the following deterministic TPMs (printed in blue below).

$$P(d|a, [d(a')]^1_{a'=0}) = \mathbb{1}(\quad d = \underbrace{ad(1) + (1-a)d(0)}_{=\sum_{a'=0}^{1}\delta(a,a')d(a') \quad = \quad d(a)} \quad) \tag{49.1}$$

$$P(y|d, [y(d')]|^1_{d'=0}) = \mathbb{1}(\quad y = y(d) \quad) \tag{49.2}$$

	$d^\sigma(0)$	$d^\sigma(1)$
never-takers	0	0
compliers	0	1
defiers	1	0
always-takers	1	1

Table 49.2: Possible compliance behaviors for individual σ.

Table 49.2 gives a name to the 4 possible compliance behaviors that might be exhibited by an individual σ of a dataset. Below, we will use $\mathcal{C}$ to denote the conditions that define a **complier**:

$$\mathcal{C} = \{\underline{d}(0) = 0, \underline{d}(1) = 1\} \tag{49.3}$$

Monotonicity is said to hold if

$$d^{\sigma}(1) \geq d^{\sigma}(0) \tag{49.4}$$

Note that monotonicity rules out defiers (i.e., $d^{\sigma}(1) = 0,\ d^{\sigma}(0) = 1$), but allows the other 3 compliance behaviors.

It is convenient to define the following expected values:

$$\mathcal{D}_{|a} = \sum_{d} d\ P(d|a) = E_{|a}[\underline{d}] \tag{49.5}$$

$$\mathcal{Y}_{|d,a} = \sum_{y} y\ P(y|d,a) = E_{|d,a}[\underline{y}] \tag{49.6}$$

$$\mathcal{Y}_{|\underline{d}=d} = \sum_{y} y\ P(y|d) = E_{|d}[\underline{y}] \tag{49.7}$$

$$\mathcal{Y}_{|\underline{a}=a} = \sum_{y} y\ P(y|a) = E_{|a}[\underline{y}] \tag{49.8}$$

Assume that $\mathcal{D}_{|1} \neq \mathcal{D}_{|0}$. Then LATE is defined by

$$\boxed{LATE = \frac{\mathcal{Y}_{|\underline{a}=1} - \mathcal{Y}_{|\underline{a}=0}}{\mathcal{D}_{|1} - \mathcal{D}_{|0}}} \tag{49.9}$$

Claim 80 *If monotonicity is satisfied, then*

$$P(\mathcal{C}) = \mathcal{D}_{|1} - \mathcal{D}_{|0} \tag{49.10}$$

proof:

$$\mathcal{D}_{|1} - \mathcal{D}_{|0} = \sum_{d(0),d(1)} P(d(0),d(1))[d(1) - d(0)] \tag{49.11}$$

$$= \begin{cases} P(d(0)=0,d(1)=0)\underbrace{[d(1)-d(0)]}_{0} \\ +P(d(0)=0,d(1)=1)\underbrace{[d(1)-d(0)]}_{1} \\ +\underbrace{P(d(0)=1,d(1)=0)}_{=0 \text{ by monotonicity}}[d(1)-d(0)] \\ +P(d(0)=1,d(1)=1)\underbrace{[d(1)-d(0)]}_{0} \end{cases} \tag{49.12}$$

$$= P(\mathcal{C}) \tag{49.13}$$

QED

Claim 81 *Recall*

$$ATE = E[\underline{y}(1) - \underline{y}(0)] \tag{49.14}$$

If monotonicity is satisfied, then

$$LATE = E_{|\mathcal{C}}[\underline{y}(1) - \underline{y}(0)] \tag{49.15}$$

proof:

$$\mathcal{Y}_{|\underline{a}=1} - \mathcal{Y}_{|\underline{a}=0} = E_{|\underline{a}=1}[\underline{y}] - E_{|\underline{a}=0}[\underline{y}] \tag{49.16}$$

$$= \sum_{x} \sum_{y(0),y(1)} \sum_{d(0),d(1)} \begin{cases} [y(d(1)) - y(d(0))] \\ *P(y(0), y(1)|x) \\ *P(d(0), d(1)|x) \\ *P(x) \end{cases} \tag{49.17}$$

$$= \sum_{y(0),y(1)} \sum_{d(0),d(1)} [y(d(1)) - y(d(0))] \begin{cases} P(y(0), y(1)|d(0), d(1)) \\ *P(d(0), d(1)) \end{cases} \tag{49.18}$$

$$= \sum_{y(0),y(1)} \begin{cases} \underbrace{[y(0) - y(0)]}_{0} \begin{cases} P(y(0), y(1)|d(0) = 0, d(1) = 0) \\ *P(d(0) = 0, d(1) = 0) \end{cases} \\ +[y(1) - y(0)] \begin{cases} P(y(0), y(1)|d(0) = 0, d(1) = 1) \\ *P(d(0) = 0, d(1) = 1) \end{cases} \\ +[y(0) - y(1)] \begin{cases} P(y(0), y(1)|d(0) = 0, d(1) = 0) \\ * \underbrace{P(d(0) = 1, d(1) = 0)}_{=0 \text{ monotonicity}} \end{cases} \\ + \underbrace{[y(1) - y(1)]}_{0} \begin{cases} P(y(0), y(1)|d(0) = 1, d(1) = 1) \\ *P(d(0) = 1, d(1) = 1) \end{cases} \end{cases} \tag{49.19}$$

$$= P(\mathcal{C}) \sum_{y(0),y(1)} [y(1) - y(0)] P(y(0), y(1)|\mathcal{C}) \tag{49.20}$$

$$= P(\mathcal{C}) E_{|\mathcal{C}}[\underline{y}(1) - \underline{y}(0)] \tag{49.21}$$

QED

It is instructive to evaluate LATE for the special case in which G of Fig.49.1 is an LDEN (Linear Deterministic with External Noise) bnet.[4]

Consider Fig.49.2. From that figure

[4]LDEN bnets are discussed in Chapter 52.

Figure 49.2: LDEN bnet for LATE. External root nodes $\underline{\epsilon_a}, \underline{\epsilon_d}, \underline{\epsilon_x}, \underline{\epsilon_y}$ are left implicit.

$$\underline{d} = \alpha \underline{a} + \mu \underline{x} + \underline{\epsilon_d} \tag{49.22}$$

$$\underline{y} = \delta \underline{d} + \nu \underline{x} + \underline{\epsilon_y} \tag{49.23}$$

Claim 82 *For the LDEN bnet of Fig.49.2,*

$$LATE = \delta = \frac{\frac{\partial \underline{y}}{\partial \underline{a}}}{\frac{\partial \underline{d}}{\partial \underline{a}}} \tag{49.24}$$

proof:

Note that

$$\langle \underline{a}, \underline{x} \rangle = \left\langle \underline{a}, \underline{\epsilon_d} \right\rangle = \left\langle \underline{a}, \underline{\epsilon_y} \right\rangle = 0 \tag{49.25}$$

because in all cases, the path between the two nodes of the covariance is blocked by a collider. Thus

$$\langle \underline{a}, \underline{d} \rangle = \alpha \langle \underline{a}, \underline{a} \rangle \tag{49.26}$$

$$\left\langle \underline{a}, \underline{y} \right\rangle \quad = \quad \delta \langle \underline{a}, \underline{d} \rangle \tag{49.27}$$

Hence,

$$\alpha = \frac{\langle \underline{a}, \underline{d} \rangle}{\langle \underline{a}, \underline{a} \rangle} = \frac{\partial \underline{d}}{\partial \underline{a}} \tag{49.28}$$

and

$$\delta = \frac{\left\langle \underline{a}, \underline{y} \right\rangle}{\langle \underline{a}, \underline{d} \rangle} = \frac{\left\langle \underline{a}, \underline{y} \right\rangle}{\langle \underline{a}, \underline{a} \rangle} \frac{\langle \underline{a}, \underline{a} \rangle}{\langle \underline{a}, \underline{d} \rangle} = \frac{\frac{\partial \underline{y}}{\partial \underline{a}}}{\frac{\partial \underline{d}}{\partial \underline{a}}} \tag{49.29}$$

$$
\begin{aligned}
E_{\underline{a}=1}[\underline{y}] - E_{\underline{a}=0}[\underline{y}] &= E_{\underline{a}=1}[\delta \underline{d} + \nu \underline{x}] - E_{\underline{a}=0}[\delta \underline{d} + \nu \underline{x}] && (49.30)\\
&= \delta(E_{\underline{a}=1}[\underline{d}] - E_{\underline{a}=0}[\underline{d}]) + \nu \underbrace{(E_{\underline{a}=1}[\underline{x}] - E_{\underline{a}=0}[\underline{x}])}_{0} && (49.31)
\end{aligned}
$$

QED

Chapter 50

LDEN with feedback loops

This chapter assumes that the reader has read Chapter 52 on LDEN (linear deterministic with external noise) diagrams and Section C.46 on the Z-transform. The algorithm described in this chapter was first published here, and is implemented in my software SCuMpy (see Ref.[95]).

Figure 50.1: LDEN bnets with two $\underline{x}_j$ nodes (exogenous nodes $\underline{u}_j$ not shown). Figure (a) shows a single time-slice with feedback loops. Figure (b) is an "unrolled" version of figure (a) showing 3 time-slices. LDEN bnets with feedback loops are a special case of Dynamic Bayesian networks (DBN) (see Chapter 26)

Consider Fig.50.1 of an LDEN bnet with feedback loops. It represents the following two "structural equations":

$$\underline{x}_0^{[n+1]} = \underbrace{\sum_{j=0}^{1} \beta_{0|j} \underline{x}_j^{[n]}}_{\text{from past}} + \underline{u}_0^{[n+1]} \tag{50.1}$$

$$\underline{x}_1^{[n+1]} = \underbrace{\sum_{j=0}^{1} \beta_{1|j} \underline{x}_j^{[n]}}_{\text{from past}} + \alpha_{1|0} \underline{x}_0^{[n+1]} + \underline{u}_1^{[n+1]} \tag{50.2}$$

for $n = 0, 1, 2, \ldots$ with $\underline{x}_j^{[0]} = \underline{u}_j^{[0]} = 0$ for all j.
From the results of Section C.46, we conclude that

$$\mathcal{Z}(\underline{x}_j^{[n+1]}) = z\left(\widetilde{\underline{x}}_j(z) - z\underline{x}_j^{[0]}\right) = z\widetilde{\underline{x}}_j(z) \tag{50.3}$$

$$\mathcal{Z}(\underline{u}_j^{[n+1]}) = z\widetilde{\underline{u}}_j(z) \tag{50.4}$$

Therefore, in z-space, the two structural equations are as follows:

$$z\widetilde{\underline{x}}_0(z) = \sum_{j=0}^{1} \beta_{0|j} \widetilde{\underline{x}}_j(z) + z\widetilde{\underline{u}}_0(z) \tag{50.5}$$

$$z\widetilde{\underline{x}}_1(z) = \sum_{j=0}^{1} \beta_{1|j} \widetilde{\underline{x}}_j(z) + \alpha_{1|0} z\widetilde{\underline{x}}_0(z) + z\widetilde{\underline{u}}_1(z) \tag{50.6}$$

We can express these two z-space structural equations in matrix form. Let

$$\widetilde{x} = \begin{bmatrix} \widetilde{x}_0 \\ \widetilde{x}_1 \end{bmatrix}, \quad \widetilde{u} = \begin{bmatrix} \widetilde{u}_0 \\ \widetilde{u}_1 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 0 \\ \alpha_{1|0} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \beta_{0|0} & \beta_{0|1} \\ \beta_{1|0} & \beta_{1|1} \end{bmatrix} \tag{50.7}$$

$$\mathbb{1}_A = 1 - A \tag{50.8}$$

$$M(z) = \mathbb{1}_A - B/z \tag{50.9}$$

Then the two z-space structural equations reduce to the single matrix equation:

$$M(z)\widetilde{x}(z) = \widetilde{u}(z) \tag{50.10}$$

If the DAG for a single time-slice has $N > 2$ nodes, then this matrix equation is still valid. In that case, A and B must be $N \times N$ matrices. The graph for a single time slice is acyclic (i.e., a DAG), so we can order its nodes topologically. This just means that $\underline{x}_i$ is a child of $\underline{x}_j$ if $i > j$. If the nodes are indexed topologically, then $\alpha_{i|j} = 0$

for $i \leq j$, so matrix A is strictly lower diagonal. Henceforth, everything we say will be valid for an arbitrary number $N \geq 2$ of nodes.

For $\underline{a} = \underline{x}, \underline{u}$, let

$$[C_{\underline{a}}^{[n]}]_{i,j} = \left\langle \underline{a}_i^{[n]}, \underline{a}_j^{[n]} \right\rangle, \quad [C_{\underline{a}}^{[n,n+1]}]_{i,j} = \left\langle \underline{a}_i^{[n]}, \underline{a}_j^{[n+1]} \right\rangle \tag{50.11}$$

and

$$[C_{\widetilde{\underline{a}}}^{z_1,z_2}]_{i,j} = \left\langle \widetilde{\underline{a}}_i(z_1), \widetilde{\underline{a}}_j(z_2) \right\rangle, \tag{50.12}$$

Let

$\mathcal{C}$ = single time covariance matrices $[C_{\underline{x}}^{[n]}]_{i,j} = \langle \underline{x}_i^{[n]}, \underline{x}_j^{[n]} \rangle$ and $[C_{\underline{x}}^{[n+1]}]_{i,j} = \langle \underline{x}_i^{[n+1]}, \underline{x}_j^{[n+1]} \rangle$, and two-times covariance matrix $[C_{\underline{x}}^{[n,n+1]}]_{i,j} = \langle \underline{x}_i^{[n]}, \underline{x}_j^{[n+1]} \rangle$.

A = strictly lower triangular matrix with entries $\alpha_{i|j}$ = gain of arrow $x_j^{[n]} \to x_i^{[n]}$

B = matrix with entries $\beta_{i|j}$ = gain of arrow $x_j^{[n]} \to x_i^{[n+1]}$

The rest of this chapter will be dedicated to accomplishing the following 2 tasks:

1. Express $\mathcal{C}$ in terms of A and B
2. Express A and B in terms of $\mathcal{C}$

Due to the linearity of the model, we will find that these two tasks can be accomplished exactly, in closed form.

Claim 83 $C_{\underline{x}}^{[n]} = \left\langle \underline{x}^{[n]}, \underline{x}^{[n]T} \right\rangle$ *satisfies*

$$C_{\underline{x}}^{[n]} = G^{n-1}\, C_{\underline{x}}^{[1]}\, (G^T)^{n-1} \tag{50.13}$$

for $n = 1, 2, 3 \ldots$, *where the "growth matrix"* G *is given by*

$$G = \mathbb{1}_A^{-1} B \tag{50.14}$$

and the "initial covariance matrix" $C_{\underline{x}}^{[1]} = \left\langle \underline{x}^{[1]}, \underline{x}^{[1]T} \right\rangle$ *by*

$$C_{\underline{x}}^{[1]} = \mathbb{1}_A^{-1} diag(\sigma_{\underline{u}_i}^2)(\mathbb{1}_A^{-1})^T \tag{50.15}$$

Once we know the single time covariance matrix $\left\langle \underline{x}^{[n]}, \underline{x}^{[n]T} \right\rangle$, *the 2 times covariance matrix* $\left\langle \underline{x}^{[n]}, \underline{x}^{[n+1]T} \right\rangle$ *can be found using the equation*

$$\left\langle \underline{x}^{[n]}, \underline{x}^{[n+1]T} \right\rangle = \left\langle \underline{x}^{[n]}, \underline{x}^{[n]T} \right\rangle G^T \tag{50.16}$$

proof:

Recall from Section C.46 that

Z-transform of same-time product

$$x_1^{[n]} x_2^{[n]} = \mathcal{Z}^{-1}\left[\frac{1}{2\pi i}\oint_C \frac{dw}{w}\, \widetilde{x}_1(w)\widetilde{x}_2\left(\frac{z}{w}\right)\right] \tag{50.17}$$

It follows that

$$\underbrace{\left\langle \underline{x}^{[n]}, \underline{x}^{[n]T}\right\rangle}_{C_{\underline{x}}^{[n]}} = \mathcal{Z}^{-1}\left[\frac{1}{2\pi i}\oint_C \frac{dw}{w}\, \underbrace{\left\langle \widetilde{\underline{x}}(w), \widetilde{\underline{x}}^T\left(\frac{z}{w}\right)\right\rangle}_{C_{\widetilde{\underline{x}}}^{w,z/w}}\right] \tag{50.18}$$

where

$$C_{\widetilde{\underline{x}}}^{w,z/w} = M^{-1}(w) C_{\widetilde{\underline{u}}}^{w,z/w} (M^{-1})^T(z/w) \tag{50.19}$$

so therefore

$$C_{\underline{x}}^{[n]} = \mathcal{Z}^{-1}\left[\frac{1}{2\pi i}\oint_C \frac{dw}{w}\, M^{-1}(w) C_{\widetilde{\underline{u}}}^{w,z/w} (M^{-1})^T(z/w)\right] \tag{50.20}$$

At this juncture, we would like to find an expression for $C_{\widetilde{\underline{u}}}^{w,z/w}$ to plug into Eq.(50.20). This shaded frame is dedicated to finding such an expression.

For all i, $\underline{u}_i^{[0]} = 0$. Keep in mind that $\mathbf{H}_0^{[n]} = 1$ at $n = 0$, so we must have

$$\left\langle \underline{u}_i^{[n]}, \underline{u}_j^{[n]}\right\rangle = \delta(i,j)\sigma_{\underline{u}_i}^2 \mathbf{H}_1^{[n]} \tag{50.21}$$

Recall from Section C.46 that:

Z-transform of Heavyside unit step function:

$$\mathcal{Z}\left[a^n \mathbf{H}_0^{[n]}\right] = \frac{1}{1 - a/z} = \frac{z}{z-a} \quad \text{for } |z| > |a| \tag{50.22}$$

Z-transform of discrete Kronecker delta function:

$$\mathcal{Z}[\delta_{n_0}^{[n]}] = z^{-n_0} \tag{50.23}$$

Unit step function as a sum of delta functions:

$$H_0^{[n]} = \sum_{k=0}^{\infty} \delta_k^{[n]} \tag{50.24}$$

It follows that

$$\mathbf{H}_1^{[n]} = \mathbf{H}_0^{[n]} - \delta_0^{[n]} \tag{50.25}$$

$$\begin{aligned}
\mathcal{Z}[\mathbf{H}_1^{[n]}] &= \mathcal{Z}[\mathbf{H}_0^{[n]}] - \mathcal{Z}[\delta_0^{[n]}] && (50.26)\\
&= \frac{z}{z-1} - 1 && (50.27)\\
&= \frac{1}{z-1} && (50.28)
\end{aligned}$$

Therefore,

$$\mathcal{Z}[\left\langle \underline{u}_i^{[n]}, \underline{u}_j^{[n]} \right\rangle] = \delta(i,j)\sigma^2_{\underline{u}_i} \frac{1}{z-1} \tag{50.29}$$

We can satisfy Eq.(50.29) and the following equation

$$\mathcal{Z}[\left\langle \underline{u}_i^{[n]}, \underline{u}_j^{[n]} \right\rangle] = \frac{1}{2\pi i} \oint_C \frac{dw}{w} \left\langle \underline{\widetilde{u}}_i(w), \underline{\widetilde{u}}_j\left(\frac{z}{w}\right) \right\rangle \tag{50.30}$$

if we set

$$\left\langle \underline{\widetilde{u}}_i(w), \underline{\widetilde{u}}_j(z/w) \right\rangle = \delta(i,j)\sigma^2_{\underline{u}_i} \frac{1}{z-1} \approx \delta(i,j)\sigma^2_{\underline{u}_i} \frac{1}{z} \tag{50.31}$$

I believe that approximating $1/(z-1)$ by $1/z$ leads to a very small change in the subsequent results of this chapter. This approximation merely shifts a pole from $z = 1$ to $z = 0$, and we will only use this expression to do complex contour integrals over a circle in the complex plane with radius $|z| >> 1$.

Plugging Eq.(50.31) into Eq.(50.20) now yields

$$C_{\underline{x}}^{[n]} = \mathcal{Z}^{-1}\left[\frac{1}{2\pi i} \oint_C \frac{dw}{w}\, M^{-1}(w) \frac{diag(\sigma^2_{\underline{u}_i})}{z} (M^{-1})^T(z/w)\right] \tag{50.32}$$

where

$$\begin{aligned}
M^{-1}(w) &= (\mathbb{1}_A - B/w)^{-1} && (50.33)\\
&= (\mathbb{1}_A w - B)^{-1} w && (50.34)\\
&= (w - \mathbb{1}_A^{-1} B)^{-1} w \mathbb{1}_A^{-1} && (50.35)\\
&= \frac{w}{w - \mathbb{1}_A^{-1} B} \mathbb{1}_A^{-1} && (50.36)
\end{aligned}$$

and

$$(M^{-1})^T(z/w) = (\mathbb{1}_A^{-1})^T \frac{z/w}{z/w - B^T(\mathbb{1}_A^{-1})^T} \tag{50.37}$$

$$= (\mathbb{1}_A^{-1})^T \frac{z}{z - B^T(\mathbb{1}_A^{-1})^T w} \tag{50.38}$$

Next, we will avail ourselves of 2 more results from Section C.46:
Inverse Z-transform:

$$\underbrace{\mathcal{Z}^{-1}[\widetilde{x}(z)]}_{x^{[n]}} = \frac{1}{2\pi i}\oint_C dz\ \widetilde{x}(z) z^{n-1} \tag{50.39}$$

Time delay:

$$x^{[n-1]} = \mathcal{Z}^{-1}\left[\frac{1}{z}\widetilde{x}(z)\right] \tag{50.40}$$

Using Eq.(50.22) and Eq.(50.40), we get

$$\mathcal{Z}_z^{-1}\left[\frac{1}{z}(M^{-1})^T(z/w)\right] = (\mathbb{1}_A^{-1})^T\left[B^T(\mathbb{1}_A^{-1})^T w\right]^{n-1} \tag{50.41}$$

$$C_{\underline{x}}^{[n]} = \frac{1}{2\pi i}\oint_C \frac{dw}{w}\ M^{-1}(w) diag(\sigma^2_{\underline{u}_i})\mathcal{Z}_z^{-1}\left[\frac{1}{z}(M^{-1})^T(z/w)\right] \tag{50.42}$$

$$= \left[\frac{1}{2\pi i}\oint_C dw\ \frac{w^{n-1}}{w - \mathbb{1}_A^{-1}B}\right]\mathbb{1}_A^{-1} diag(\sigma^2_{\underline{u}_i})(\mathbb{1}_A^{-1})^T\left[B^T(\mathbb{1}_A^{-1})^T\right]^{n-1} \tag{50.43}$$

$$= \mathcal{Z}_w^{-1}\left[\frac{1}{w - \mathbb{1}_A^{-1}B}\right]\mathbb{1}_A^{-1} diag(\sigma^2_{\underline{u}_i})(\mathbb{1}_A^{-1})^T\left[B^T(\mathbb{1}_A^{-1})^T\right]^{n-1} \quad \text{(by Eq.(50.39))} \tag{50.44}$$

$$= [\mathbb{1}_A^{-1}B]^{n-1}\mathbb{1}_A^{-1} diag(\sigma^2_{\underline{u}_i})(\mathbb{1}_A^{-1})^T\left[B^T(\mathbb{1}_A^{-1})^T\right]^{n-1} \quad \text{(by Eqs.(50.22)(50.40))} \tag{50.45}$$

Eq.(50.16) is the same as Eq.(50.49), and the latter is proven below.

QED

The structural equations in n (i.e., time) space, and in matrix form, can be expressed as:

$$\underline{x}^{[n+1]} = B\underline{x}^{[n]} + A\underline{x}^{[n+1]} + \underline{u}^{[n+1]} \tag{50.46}$$

Applying $\left\langle \cdot, \underline{x}^{[n]T}\right\rangle$ from the right to Eq.(50.46), we get

$$(1 - A)\left\langle \underline{x}^{[n+1]}, \underline{x}^{[n]T}\right\rangle = B\left\langle \underline{x}^{[n]}, \underline{x}^{[n]T}\right\rangle + \left\langle \underline{u}^{[n+1]}, \underline{x}^{[n]T}\right\rangle \tag{50.47}$$

Note that

$$\left\langle \underline{u}^{[n+1]}, \underline{x}^{[n]T}\right\rangle = 0 \tag{50.48}$$

by causality, because the $\underline{x}^{[n]}$ can't be affected by the noise in the future. Therefore

$$\boxed{\mathbb{1}_A \left\langle \underline{x}^{[n]}, \underline{x}^{[n+1]T} \right\rangle^T = B \left\langle \underline{x}^{[n]}, \underline{x}^{[n]T} \right\rangle} \tag{50.49}$$

Applying $\left\langle \cdot, \underline{x}^{[n+1]T} \right\rangle$ from the right to Eq.(50.46), we get

$$\left\langle \underline{x}^{[n+1]}, \underline{x}^{[n+1]T} \right\rangle = B \left\langle \underline{x}^{[n]}, \underline{x}^{[n+1]T} \right\rangle + A \left\langle \underline{x}^{[n+1]}, \underline{x}^{[n+1]T} \right\rangle + \left\langle \underline{u}^{[n+1]}, \underline{x}^{[n+1]T} \right\rangle \tag{50.50}$$

For any matrix A, define $SL(A)$ to be the strictly lower triangular matrix obtained from matrix A by setting to zero all entries on the main diagonal of A and above it. Note that

$$SL(\left\langle \underline{u}^{[n+1]}, \underline{x}^{[n+1]T} \right\rangle) = 0 \tag{50.51}$$

by causality, because the nodes are assumed to be topologically ordered (when the feedback arrows are removed), and the future noise $\underline{u}_i^{[n+1]}$ cannot be correlated with $\underline{x}_j^{[n+1]}$ where $i > j$. Thus,

$$\boxed{SL\left(\left\langle \underline{x}^{[n+1]}, \underline{x}^{[n+1]^T} \right\rangle - \underbrace{B \left\langle \underline{x}^{[n]}, \underline{x}^{[n+1]T} \right\rangle}_{K} \right) = SL\left(A \left\langle \underline{x}^{[n+1]}, \underline{x}^{[n+1]^T} \right\rangle \right)} \tag{50.52}$$

A and B satisfy a system of 2 linear equations (the two boxed equations above, Eq.(50.49) and Eq.(50.52)) with two unknowns A, B. To solve that system of 2 equations in (A, B), we can:

1. First solve Eq.(50.52)) for A in terms of B and $\mathcal{C}$, thus obtaining $A(B, \mathcal{C})$. To do this step, we can use the same method that was used in Chapter 52, for LDEN without feedback loops, to solve for A when $K = 0$.

   Caveat: Eq.(50.52)) for A has the same number of equations as unknowns as long as, for all $i > j$, $\alpha_{i|j} \neq 0$. If some of those $\alpha_{i|j}$ are zero, then we get an overdetermined system of linear equations. To solve that problem, for every i, j such that $i > j$ and $\alpha_{i|j} = 0$, replace that unknown by $\left\langle \underline{x}_i^{[n]}, \underline{x}_j^{[n]} \right\rangle$.

2. Then we can substitute $A(B, \mathcal{C})$ into the remaining equation Eq.(50.49) to obtain $B(\mathcal{C})$.

   Caveat: Eq.(50.49)) for B has the same number of equations as unknowns as long as all $\beta_{i|j} \neq 0$ for all i, j. If some of those $\beta_{i|j}$ are zero, then we get an overdetermined system of linear equations. To solve that problem, for every i, j such that $\beta_{i|j} = 0$, replace that unknown by $\left\langle \underline{x}_j^{[n]}, \underline{x}_i^{[n+1]} \right\rangle$.

3. Finally, we can substitute $B(\mathcal{C})$ into $A(B, \mathcal{C})$ to get $A(B(\mathcal{C}), \mathcal{C})$.

Chapter 51

Linear and Logistic Regression via grad descent

Figure 51.1: Linear Regression

Figure 51.2: Bnet of Fig.51.1 with new $\underline{\vec{Y}}$ node.

Estimators $\hat{y}$ for linear and logistic regression.

- **Linear Regression:** $y \in \mathbb{R}$. Note $\widehat{y} \in \mathbb{R}$. $(x, \widehat{y}(x))$ is the graph of a straight line with y-intercept b and slope m.

$$\widehat{y}(x; b, m) = b + mx \tag{51.1}$$

- **Logistic Regression:** $y \in \{0, 1\}$. Note $\widehat{y} \in [0, 1]$. $(x, \widehat{y}(x))$ is the graph of a sigmoid. Often in literature, b, m are replaced by β_0, β_1.

$$\widehat{y}(x; b, m) = \text{smoid}(b + mx) \tag{51.2}$$

Define

$$V(b, m) = \sum_{x,y} P(x, y)|y - \widehat{y}(x; b, m)|^2 \ . \tag{51.3}$$

We want to minimize $V(b, m)$ (called a cost or loss function) wrt b and m.

The TPMs, printed in blue, for the Bnet Fig.51.1, are as follows.

$$P(b, m) = \text{given} \tag{51.4}$$

The first time it is used, (b, m) is arbitrary. After the first time, it is determined by previous stage.

Let

$$P_{\underline{x},\underline{y}}(x, y) = \frac{1}{nsam(\vec{x})} \sum_{\sigma} \mathbb{1}(x = x^\sigma, y = y^\sigma) \ . \tag{51.5}$$

$$P(\vec{x}) = \prod_{\sigma} P(x^\sigma) \tag{51.6}$$

$$P(\vec{y}|\vec{x}) = \prod_{\sigma} P(y^\sigma \mid x^\sigma) \tag{51.7}$$

$$P(\vec{\widehat{y}}|\vec{x}, b, m) = \prod_{\sigma} \delta(\widehat{y}^\sigma, \widehat{y}(x^\sigma, b, m)) \tag{51.8}$$

$$P(V|\vec{\widehat{y}}, \vec{y}) = \delta(V, \frac{1}{nsam(\vec{x})} \sum_{\sigma} |y^\sigma - \widehat{y}^\sigma|^2) \tag{51.9}$$

Let $\eta_b, \eta_m > 0$. For $x = b, m$, if $x' - x = \Delta x = -\eta \frac{\partial V}{\partial x}$, then $\Delta V \approx \frac{-1}{\eta}(\Delta x)^2 \leq 0$ for $\eta > 0$. This is called "gradient descent".

$$P(b'|V, b) = \delta(b', b - \eta_b \partial_b V) \tag{51.10}$$

$$P(m'|V, m) = \delta(m', m - \eta_m \partial_m V) \tag{51.11}$$

51.1 Generalization to x with multiple components (features)

Suppose that for each sample σ, instead of x^σ being a scalar, it has n components called features:

$$x^\sigma = (x_0^\sigma, x_1^\sigma, x_2^\sigma, \dots x_{n-1}^\sigma) \ . \tag{51.12}$$

Slope m is replaced by weights

$$w = (w_0, w_1, w_3, , \dots, w_{n-1}) \ , \tag{51.13}$$

and the product of 2 scalars mx^σ is replaced by the inner vector product $w^T x^\sigma$.

51.2 Alternative $V(b, m)$ for logistic regression

For logistic regression, since $y^\sigma \in \{0, 1\}$ and $\widehat{y}^\sigma \in [0, 1]$ are both in the interval $[0, 1]$, they can be interpreted as probabilities. Define probability distributions $p^\sigma(x)$ and $\widehat{p}^\sigma(x)$ for $x \in \{0, 1\}$ by

$$p^\sigma(1) = y^\sigma, \quad p^\sigma(0) = 1 - y^\sigma \tag{51.14}$$

$$\widehat{p}^\sigma(1) = \widehat{y}^\sigma, \quad \widehat{p}^\sigma(0) = 1 - \widehat{y}^\sigma \tag{51.15}$$

Then for logistic regression, the following 2 cost functions $V(b, m)$ can be used as alternatives to the cost function Eq.(51.3) previously given.

$$V(b, m) = \frac{1}{nsam(\vec{x})} \sum_\sigma D_{KL}(p^\sigma \parallel \widehat{p}^\sigma) \tag{51.16}$$

and

$$\begin{aligned}
V(b, m) &= \frac{1}{nsam(\vec{x})} \sum_\sigma CE(p^\sigma \parallel \widehat{p}^\sigma) && (51.17)\\
&= \frac{-1}{nsam(\vec{x})} \sum_\sigma \{y^\sigma \ln \widehat{y}^\sigma + (1 - y^\sigma) \ln(1 - \widehat{y}^\sigma)\} && (51.18)\\
&= \frac{-1}{nsam(\vec{x})} \sum_\sigma \ln \left\{ (\widehat{y}^\sigma)^{y^\sigma} (1 - \widehat{y}^\sigma)^{(1-y^\sigma)} \right\} && (51.19)\\
&= \frac{-1}{nsam(\vec{x})} \sum_\sigma \ln P(\underline{Y} = y^\sigma \mid \widehat{y} = \widehat{y}^\sigma) && (51.20)\\
&= -\sum_{x,y} P(x, y) \ln P(\underline{Y} = y | \widehat{y} = \widehat{y}(x, b, m)) && (51.21)
\end{aligned}$$

Above, we used

$$P(\underline{Y} = Y | \widehat{y}) = \widehat{y}^Y [1 - \widehat{y}]^{1-Y} \tag{51.22}$$

for $Y \in val(\underline{Y}) = \{0, 1\}$. (Bernoulli distribution).

There is no node corresponding to $\underline{Y}$ in the Bnet of Fig.51.1. Fig.51.2 shows a new Bnet that has a new node called $\vec{\underline{Y}}$ compared to the Bnet of Fig.51.1. One defines the TPMs for all nodes of Fig.51.2 except $\vec{\underline{Y}}$ and $\underline{V}$ the same as for Fig.51.1. For $\vec{\underline{Y}}$ and $\underline{V}$, one defines

$$P(Y^\sigma \mid \vec{\widehat{y}}) = P(\underline{Y} = Y^\sigma \mid \widehat{y}^\sigma) \tag{51.23}$$

$$P(V|\vec{Y}, \vec{y}) = \delta(V, \frac{-1}{nsam(\vec{x})} \ln L) \ , \tag{51.24}$$

where $L = \prod_\sigma P(\underline{Y} = y^\sigma \mid \widehat{y}^\sigma)$=likelihood.

Chapter 52

Linear Deterministic Bnets with External Noise (LDEN Bnets)

A **Linear Deterministic (LD) bnet** is a bnet with TPMs of the form

$$P(b|\{\underline{a}_j = a_j\}_{\forall j}) = \mathbb{1}\left(b = \sum_j g_j a_j\right) \tag{52.1}$$

or, equivalently, with **structural equations** of the form

$$\underline{b} = \sum_j g_j \underline{a}_j \tag{52.2}$$

for every node $\underline{b}$ of the bnet, where $\{\underline{a}_j\}_{\forall j}$ are the parent nodes of $\underline{b}$, and the **gains** g_j are real numbers.

In this chapter, we will consider bnets which were referred to, prior to the invention of bnets, as: Sewall Wright's **Path Analysis (PA)** and **linear Structural Equations Models (SEM)**. Judea Pearl in his books calls them **linear Structural Causal Models (SCM)**, because they are very convenient for doing causal analysis. We will refer to them as **Linear Deterministic with External Noise (LDEN) bnets**. This chapter is devoted to LDEN bnets, except that we will say a few words about non-linear DEN bnets at the end.

A **DEN bnet** is a special kind of bnet. To build a DEN bnet, start with a deterministic bnet G. The deterministic nodes of G are called the **endogenous (internal) variables**. Now make a bigger bnet $\overline{G}$ called a DEN bnet by adding to each node $\underline{a}$ of G a non-deterministic root node $\underline{u}_{\underline{a}}$ pointing into $\underline{a}$ only. The nodes $\underline{u}_{\underline{a}}$ are called the **exogenous (external) variables**. The exogenous variables make their children noisy. They are assumed to be unobserved and their TPMs are prior probability distributions. Since they are root nodes, they are mutually independent. When we draw a DEN bnet, we will sometimes not draw the exogenous nodes, leaving them implicit.

A **linear DEN (LDEN) bnet** is a DEN bnet whose deterministic nodes have a TPM that is a linear function of the states of the parent nodes.

This chapter uses the notation $\left\langle \underline{x}, \underline{y} \right\rangle$ for the covariance of any two random variables $\underline{x}, \underline{y}$. This $\left\langle \underline{x}, \underline{y} \right\rangle$ notation is defined in Chapter C.

52.1 Example of LDEN bnet

Figure 52.1: Example of a LDEN bnet wherein $\underline{x}$ splits into two nodes $\underline{z}$ and $\underline{w}$, then merges into node $\underline{y}$. There is also an arrow $\underline{z} \to \underline{w}$. Exogenous nodes are not shown. The Greek letters represent real numbers.

The TPMs, printed in blue, for the LDEN bnet Fig.52.1, are as follows.

$$P(y|w, z, u_{\underline{y}}) = \mathbb{1}(y = \epsilon w + \delta z + u_{\underline{y}}) \tag{52.3}$$

$$P(w|x, z, u_{\underline{w}}) = \mathbb{1}(w = \beta x + \gamma z + u_{\underline{w}}) \tag{52.4}$$

$$P(z|x, u_{\underline{z}}) = \mathbb{1}(z = \alpha x + u_{\underline{z}}) \tag{52.5}$$

$$P(x|u_{\underline{x}}) = \mathbb{1}(x = u_{\underline{x}}) \tag{52.6}$$

Hence,

$$\begin{aligned} y &= \epsilon w + \delta z + u_{\underline{y}} && (52.7)\\ &= \epsilon(\beta x + \gamma z + u_{\underline{w}}) + \delta z + u_{\underline{y}} && (52.8)\\ &= (\epsilon\gamma + \delta)z + \epsilon\beta x + \epsilon u_{\underline{w}} + u_{\underline{y}} && (52.9)\\ &= (\epsilon\gamma + \delta)z + \epsilon\beta u_{\underline{x}} + \epsilon u_{\underline{w}} + u_{\underline{y}}\ . && (52.10) \end{aligned}$$

Therefore

$$\left(\frac{\partial y}{\partial z}\right)_{u.-u_{\underline{z}}} = \epsilon\gamma + \delta \ , \tag{52.11}$$

where the partial derivative holds fixed all exogenous variables except $u_{\underline{z}}$. Note that this partial derivative is a sum of terms, and that each of those terms represents a different directed path from $\underline{z}$ to $\underline{y}(\underline{z})$. This is a general property of LDEN bnets.

52.2 LDEN equations and their 2 solutions

LDEN bnets are described by a system of linear equations (known as the **structural equations**) of the form

$$\underline{x}_i = \sum_{j=0}^{nx-1} \alpha_{i|j}\underline{x}_j + \underline{u}_i \tag{52.12}$$

for $i = 0, 1, \ldots, nx-1$, where the $\underline{x}_i$ are the internal nodes, the $\alpha_{i|j}$ are the **path coefficients** (a.k.a. **arrow gains**), and the $\underline{u}_i$ are the external nodes that inject noise into the system. Some of the $\alpha_{i|j}$ may be zero, in which case the corresponding arrow $\underline{x}_j \to \underline{x}_i$ would not be drawn. The $\underline{u}_i$ are root nodes with zero covariance with each other.

We can view this as either

1. UNK-CM (unknown covariance matrix) solution

   a linear system of equations with the unknowns $\underline{x}_i$. We can solve for these unknowns using basic Linear Algebra. Once we solve for the unknowns, we can calculate the covariances $\left\langle \underline{x}_j, \underline{x}_k \right\rangle$.

2. UNK-G (unknown gains) solution

   a linear system of equations with the unknowns $\alpha_{i|j}$. We can solve for these unknowns using basic Linear Algebra.

Next, we will find the 2 solutions UNK-CM and UNK-G explicitly for arbitrary LDEN bnets. We will present these solutions gradually, first for fully connected LDEN bnets, and then for the general case that does not require full connectivity.

52.3 Fully connected LDEN bnets

The bnets that will be considered in this section will all be fully connected. Fully connected bnets are defined in Chapter F.

In this section, we will assume that the nodes $\underline{x}_j$ are ordered topologically, with the root nodes first. This means that $\underline{x}_i$ happens after $\underline{x}_j$ if $i > j$. When the nodes are ordered topologically, $\alpha_{i|j} = 0$ if $j \geq i$.

In the fully connected case, $\alpha_{i|j} \neq 0$ for all $j < i$, and for any node $\underline{x}_i$, all previous nodes are parents of $\underline{x}_i$ and we draw arrows $\underline{x}_j \to \underline{x}_i$ for $j = 0, 1, \ldots, i-1$.

In the not fully connected case, some of the $\alpha_{i|j}$ with $j < i$ are zero, in which case the corresponding arrow $\underline{x}_j \to \underline{x}_i$ is not drawn.

52.3.1 Fully connected LDEN bnet with $nx = 2$

$$\begin{tikzcd} \underline{x}_0 \arrow[d, "\alpha 1|0"] \\ \underline{x}_1 \end{tikzcd}$$

Figure 52.2: Fully connected LDEN bnet with two $\underline{x}_j$ nodes (exogenous nodes $\underline{u}_j$ not shown).

Consider the LDEN bnet of Fig.52.2. This bnet represents the following structural equations:

$$\underline{x}_0 = \underline{u}_0 \tag{52.13a}$$

$$\underline{x}_1 = \alpha_{1|0}\underline{x}_0 + \underline{u}_1 \ . \tag{52.13b}$$

1. UNK-CM solution

   Note that

$$\underline{x}_0 = \underline{u}_0 \tag{52.14a}$$

$$\underline{x}_1 = \alpha_{1|0}\underline{u}_0 + \underline{u}_1 \tag{52.14b}$$

   Thus, the $\underline{x}_i$ can be expressed in terms of the $\underline{u}_i$. Using the fact that $\left\langle \underline{u}_i, \underline{u}_j \right\rangle = 0$ for $i \neq j$, we get

$$\langle \underline{x}_0, \underline{x}_0 \rangle = \sigma^2_{\underline{u}_0} \tag{52.15}$$

$$\langle \underline{x}_1, \underline{x}_0 \rangle = \alpha_{1|0}\sigma^2_{\underline{u}_0} \tag{52.16}$$

$$\langle \underline{x}_1, \underline{x}_1 \rangle = \alpha^2_{1|0}\sigma^2_{\underline{u}_0} + \sigma^2_{\underline{u}_1} \tag{52.17}$$

2. UNK-G solution

   Note that

$$\langle \underline{x}_0, \underline{u}_1 \rangle = 0 \tag{52.18}$$

   because the path from $\underline{x}_0$ to $\underline{u}_1$ is blocked by a collider. Therefore,

$$\langle \underline{x}_0, \underline{x}_1 \rangle = \alpha_{1|0} \langle \underline{x}_0, \underline{x}_0 \rangle \tag{52.19}$$

   so[1]

$$\alpha_{1|0} = \frac{\langle \underline{x}_0, \underline{x}_1 \rangle}{\langle \underline{x}_0, \underline{x}_0 \rangle} = \frac{\partial \underline{x}_1}{\partial \underline{x}_0} \tag{52.20}$$

   Thus, $\alpha_{1|0}$ can be estimated from the covariances $\left\langle \underline{x}_i, \underline{x}_j \right\rangle$.

52.3.2 Fully connected LDEN bnet with $nx = 3$

$$\begin{tikzcd}
\underline{x}_0 \arrow[d, "\alpha_{1|0}"'] \arrow[dr, "\alpha_{2|0}"] & \\
\underline{x}_1 \arrow[r, "\alpha_{2|1}"'] & \underline{x}_2
\end{tikzcd}$$

Figure 52.3: Fully connected LDEN bnet with three $\underline{x}_j$ nodes (exogenous nodes $\underline{u}_j$ not shown).

Consider the LDEN bnet of Fig.52.3. This bnet represents the following structural equations:

$$\underline{x}_0 = \underline{u}_0 \tag{52.21a}$$
$$\underline{x}_1 = \alpha_{1|0}\underline{x}_0 + \underline{u}_1 \tag{52.21b}$$
$$\underline{x}_2 = \alpha_{2|0}\underline{x}_0 + \alpha_{2|1}\underline{x}_1 + \underline{u}_2 \ . \tag{52.21c}$$

1. UNK-CM solution

   Let

$$\underline{x} = \begin{bmatrix} \underline{x}_0 \\ \underline{x}_1 \\ \underline{x}_2 \end{bmatrix} , \quad \underline{u} = \begin{bmatrix} \underline{u}_0 \\ \underline{u}_1 \\ \underline{u}_2 \end{bmatrix} \tag{52.22}$$

   and

[1] We are using the notation $\frac{\partial \underline{b}}{\partial \underline{a}} = \frac{\langle \underline{a}, \underline{b} \rangle}{\langle \underline{a}, \underline{a} \rangle}$, for any two random variables $\underline{a}, \underline{b}$

$$A = \begin{bmatrix} 0 & 0 & 0 \\ \alpha_{1|0} & 0 & 0 \\ \alpha_{2|0} & \alpha_{2|1} & 0 \end{bmatrix} \tag{52.23}$$

Note that

$$\underline{x} = A\underline{x} + \underline{u} \tag{52.24}$$

so

$$\underline{x} = (1 - A)^{-1}\underline{u} \tag{52.25}$$

Thus, the $\underline{x}_i$ can be expressed in terms of the $\underline{u}_i$.

If we define the covariance matrix by

$$C = \left\langle \underline{x}, \underline{x}^T \right\rangle \ , \quad C_{i,j} = \left\langle \underline{x}_i, \underline{x}_j \right\rangle \tag{52.26}$$

then[2]

$$\begin{aligned} C &= (1 - A)^{-1} \left\langle \underline{\epsilon}, \underline{\epsilon}^T \right\rangle [(1 - A)^{-1}]^T && (52.27) \\ &= (1 - A)^{-1} diag(\sigma^2_{\underline{\epsilon}_i})[(1 - A)^{-1}]^T && (52.28) \end{aligned}$$

If we define the Jacobian matrix J by

$$J_{i,j} = \frac{\partial \underline{x}_i}{\partial \underline{x}_j} \tag{52.29}$$

then

$$\begin{aligned} J &= C[diag(\langle \underline{x}_i, \underline{x}_i \rangle)]^{-1} && (52.30) \\ &= (1 - A)^{-1} diag(\sigma^2_{\underline{\epsilon}_i})[(1 - A)^{-1}]^T [diag(\langle \underline{x}_i, \underline{x}_i \rangle)]^{-1} && (52.31) \end{aligned}$$

C has the nice property that it is a symmetric matrix, whereas J has the nice property that its diagonal elements are all 1.

2. UNK-G solution

   Note that

$$\begin{aligned} \langle \underline{x}_0, \underline{x}_1 \rangle &= \alpha_{1|0} \langle \underline{x}_0, \underline{x}_0 \rangle && (52.32a) \\ \langle \underline{x}_0, \underline{x}_2 \rangle &= \alpha_{2|0} \langle \underline{x}_0, \underline{x}_0 \rangle + \alpha_{2|1} \langle \underline{x}_0, \underline{x}_1 \rangle && (52.32b) \\ \langle \underline{x}_1, \underline{x}_2 \rangle &= \alpha_{2|0} \langle \underline{x}_1, \underline{x}_0 \rangle + \alpha_{2|1} \langle \underline{x}_1, \underline{x}_1 \rangle && (52.32c) \end{aligned}$$

[2] We are using the notation $diag(a)$ = diagonal matrix with $a \in \mathbb{R}^n$ along its diagonal.

Hence

$$\alpha_{1|0} = \frac{\langle \underline{x}_0, \underline{x}_1 \rangle}{\langle \underline{x}_0, \underline{x}_0 \rangle} = \frac{\partial \underline{x}_1}{\partial \underline{x}_0} \tag{52.33}$$

$$\begin{bmatrix} \langle \underline{x}_0, \underline{x}_2 \rangle \\ \langle \underline{x}_1, \underline{x}_2 \rangle \end{bmatrix} = \begin{bmatrix} \langle \underline{x}_0, \underline{x}_0 \rangle & \langle \underline{x}_0, \underline{x}_1 \rangle \\ \langle \underline{x}_1, \underline{x}_0 \rangle & \langle \underline{x}_1, \underline{x}_1 \rangle \end{bmatrix} \begin{bmatrix} \alpha_{2|0} \\ \alpha_{2|1} \end{bmatrix} \tag{52.34}$$

Let

$$\alpha^{(2)} = \begin{bmatrix} \alpha_{2|0} \\ \alpha_{2|1} \end{bmatrix}, \quad \underline{x}^{(2)} = \begin{bmatrix} \underline{x}_0 \\ \underline{x}_1 \end{bmatrix}, \quad \left\langle \underline{x}^{(2)}, \underline{x}_2 \right\rangle = \begin{bmatrix} \langle \underline{x}_0, \underline{x}_2 \rangle \\ \langle \underline{x}_1, \underline{x}_2 \rangle \end{bmatrix} \tag{52.35}$$

Define the covariance matrix $C^{(2)}$ for the third row of A by

$$C^{(2)} = \begin{bmatrix} \langle \underline{x}_0, \underline{x}_0 \rangle & \langle \underline{x}_0, \underline{x}_1 \rangle \\ \langle \underline{x}_1, \underline{x}_0 \rangle & \langle \underline{x}_1, \underline{x}_1 \rangle \end{bmatrix} \tag{52.36}$$

Then

$$\left\langle \underline{x}^{(2)}, \underline{x}_2 \right\rangle = C^{(2)} \alpha^{(2)} \tag{52.37}$$

so

$$\alpha^{(2)} = [C^{(2)}]^{-1} \left\langle \underline{x}^{(2)}, \underline{x}_2 \right\rangle \tag{52.38}$$

Alternatively, let

$$\nabla^{(2)} \underline{x}_2 = \begin{bmatrix} \partial_{\underline{x}_0} \underline{x}_2 \\ \partial_{\underline{x}_1} \underline{x}_2 \end{bmatrix} \tag{52.39}$$

Define the Jacobian matrix for the third row of A by

$$J^{(2)}_{i,j} = \frac{\partial \underline{x}_i}{\partial \underline{x}_j} \tag{52.40}$$

for $i, j \in \{0, 1\}$. Then

$$\nabla^{(2)} \underline{x}_2 = J^{(2)T} \alpha^{(2)} \tag{52.41}$$

so

$$\alpha^{(2)} = [J^{(2)T}]^{-1} \nabla^{(2)} \underline{x}_2 \tag{52.42}$$

From the $C^{(r)}$ and the $J^{(r)}$ expressions for $r = 1, 2$, we see that the $\alpha_{i|j}$ can be expressed in terms of the covariances $\left\langle \underline{x}_i, \underline{x}_j \right\rangle$.

52.3.3 Fully connected LDEN bnet with arbitrary nx

Let $\underline{x}. = (x_i)_{i=0,1,\ldots,nx-1}$ and $\underline{x}_{<i} = (x_k)_{k=0,1,\ldots,i-1}$. Consider a fully connected LDEN bnet with deterministic internal nodes labeled $\underline{x}_i$. The $\underline{x}_i$ labels are assumed to be in topological order (i.e., the parents of node $\underline{x}_i$ are $\underline{x}_{<i}$). Let the TPMs, printed in blue, for the nodes $\underline{x}.$ of the LDEN bnet, be as follows.

$$P(x_i|x_{<i}, u_i) = \mathbb{1}(x_i = \sum_{j<i} \alpha_{i|j} x_j + u_i) \ , \tag{52.43}$$

for some parameters $\alpha_{i|j} \in \mathbb{R}$. The external nodes $\underline{u}.$ are assumed to be independent so

$$P(u.) = \prod_i P(u_i) \tag{52.44}$$

and

$$\left\langle \underline{u}_i, \underline{u}_j \right\rangle = 0 \text{ if } i \neq j \ . \tag{52.45}$$

Note that

$$\begin{aligned} P(x.) &= \sum_{u.} P(u.) \prod_i P(x_i|x_{<i}, u_i) & (52.46) \\ &= E_{\underline{u}.}[\prod_i P(x_i|x_{<i}, u_i)] \ . & (52.47) \end{aligned}$$

In terms of random variables, this system is described by the following structural equations:

$$\underline{x}_i = \sum_{j<i} \alpha_{i|j} \underline{x}_j + \underline{u}_i \ . \tag{52.48}$$

1. UNK-CM solution

   The structural equations can be written in matrix form as follows. Define a strictly lower triangular matrix A with the arrow gains $\alpha_{i|j} \in \mathbb{R}$ as entries. For example, for $nx = 4$,

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \alpha_{1|0} & 0 & 0 & 0 \\ \alpha_{2|0} & \alpha_{2|1} & 0 & 0 \\ \alpha_{3|0} & \alpha_{3|1} & \alpha_{3|2} & 0 \end{bmatrix} \ . \tag{52.49}$$

   If we now represent the multinodes $\underline{x}.$ and $\underline{u}.$ as column vectors $\underline{x}$ and $\underline{u}$, we get

$$\underline{x} = A\underline{x} + \underline{u} \ . \tag{52.50}$$

Note that

$$\underline{x} = (1 - A)^{-1} \underline{u} \tag{52.51}$$

so the $\underline{x}_i$ can be expressed in terms of the $\underline{u}_i$.

Just like we did for the case $nx = 3$, we can now use Eq.(52.51) to express, in terms of A and the $\langle \underline{u}_i, \underline{u}_i \rangle = \sigma^2_{\underline{u}_i}$, the covariance matrix $C = \left\langle \underline{x}, \underline{x}^T \right\rangle$ and the Jacobian matrix J with $J_{i,j} = \frac{\partial \underline{x}_i}{\partial \underline{x}_j}$. .

2. UNK-G solution

   Note that

$$\underline{x}_i = f_i(\underline{u}_{\leq i}) \ . \tag{52.52}$$

   Therefore, if $i > k$,

$$\langle \underline{x}_k, \underline{u}_i \rangle = \left\langle f_k(\underline{u}_{\leq k}), \underline{u}_i \right\rangle = 0 \ . \tag{52.53}$$

   Thus, if $i > k$,

$$\begin{aligned} \langle \underline{x}_k, \underline{x}_i \rangle &= \sum_{j<i} \alpha_{i|j} \left\langle \underline{x}_k, \underline{x}_j \right\rangle + \langle \underline{x}_k, \underline{u}_i \rangle & (52.54) \\ &= \sum_{j<i} \alpha_{i|j} \left\langle \underline{x}_k, \underline{x}_j \right\rangle \ . & (52.55) \end{aligned}$$

   As shown for the cases $nx = 2, 3$ above, Eqs.(52.55) can be expressed as a system of equations for each row of the matrix A, and those systems of equations can be solved to express the $\alpha_{i|j}$ in terms of the covariances $\left\langle \underline{x}_i, \underline{x}_j \right\rangle$.

   Dividing both sides of Eq.(52.55) by $\langle \underline{x}_k, \underline{x}_k \rangle$, we get, for $i > k$,

$$\frac{\partial \underline{x}_i}{\partial \underline{x}_k} = \sum_{j<i} \alpha_{i|j} \frac{\partial \underline{x}_j}{\partial \underline{x}_k} \tag{52.56}$$

52.4 Not fully connected LDEN bnets

When the LDEN bnet is not fully connected, some $\alpha_{i|j}$ might be zero for $i > j$, and the corresponding arrow $\underline{x}_j \to \underline{x}_i$ is not drawn. If that is the case, the formulae that we gave above, for the fully connected LDEN bnet, for the UNK-CM solution, still apply, but the formulae that we gave above, for the UNK-G solution, must be modified as follows.

The problem with the previously presented UNK-G solution is that, when we solve for the $\alpha_{i|j}$ in each row of A, we are sometimes solving for $\alpha_{i|j}$ which we know a priori are equal to zero. So basically, we are solving a system of equations with more equations than unknowns, what is called an "overdetermined" system of equations. A simple solution to this quandary is to add to the set of unknown variables $\alpha_{i|j}$, a few of the covariances too. That way, we can get a system of equations with the same number of equations as number of unknowns. So which of the covariances should be made into unknowns? A very natural choice is to make the covariance $\left\langle \underline{x}_i, \underline{x}_j \right\rangle$ an unknown if $\alpha_{i|j} = 0$ for some $i > j$. This is what I do in Ref.[95], and it works like a charm. A fully connected LDEN bnet with N nodes has $N^2/2 - N/2$ arrows. If M of those are missing, our final result will be M constraints among the covariances, and $N^2/2 - N/2 - M$ equations expressing the non-zero $\alpha_{i|j}$ in terms of covariances.

52.5 LDEN bnet with conditioned nodes

Conditioning on a node $\underline{x}$ of an LDEN bnet means assuming that $\underline{x}$ is fixed at a specific value x. Normally, we assume

$$\left\langle \underline{\epsilon}_i, \underline{\epsilon}_j \right\rangle = 0 \quad \text{if } i \neq j \tag{52.57}$$

However, when we condition on some nodes $\{\underline{x}_i : i \in \mathcal{C}\}$, the constraint Eq.(52.57) must be modified. Instead, we assume

$$\left\langle \underline{\epsilon}_i, \underline{\epsilon}_j \right\rangle = 0 \quad \text{if } i \in \mathcal{C} \text{ or } j \in \mathcal{C}. \tag{52.58}$$

The reason this makes sense is as follows. The $\underline{\epsilon}_i$ for $i \in \mathcal{C}$ no longer serve a function because they are futilely pumping noise into a fixed node, so we may set those $\underline{\epsilon}_i$ to a constant. On the other hand, the $\underline{\epsilon}_i$ with $i \notin \mathcal{C}$, might have become correlated among themselves as a result of the conditioning. For example. we might have conditioned on a collider, and opened an unblocked path between two of those $\underline{\epsilon}_i$ with $i \neq \mathcal{C}$. See Ref.[95] for examples where this occurs.

52.6 SCuMpy

Check out my free open source software SCuMpy [95]. SCuMpy ia a Python library for doing both symbolic and numeric calculations for linear Structural Causal Models (SCM) (i.e., LDEN bnets).

52.7 Non-linear DEN bnets

This chapter is dedicated to linear DEN bnets. This implicitly assumes that the deterministic nodes $\underline{x}$. of the DEN bnet have an interval of real values as their possible

states. A trivial but very useful generalization of linear DEN bnets is to replace Eq.(52.43) for the TPMs of the deterministic nodes of the bnet by

$$P(x_i|x_{<i}, u_i) = \mathbb{1}(x_i = f_i(x_{<i}, u_i)) \ , \tag{52.59}$$

with structural equations

$$\underline{x}_i = f_i(\underline{x}_{<i}, \underline{u}_i) \ , \tag{52.60}$$

for $i = 0, 1, \ldots, nx - 1$. Here the f_i are possibly non-linear functions that depend on the states $x_{<i}$ and u_i of nodes $\underline{x}_{<i}$ and $\underline{u}_i$. If a node $\underline{x}_i$ has no arrows entering it (i.e., is a root node), then

$$P(x_i|x_{<i}, u_i) = P(x_i) = \delta(x_i, a) \tag{52.61}$$

and

$$\underline{x}_i = a \tag{52.62}$$

for some $a \in val(\underline{x}_i)$.

Besides a linear function, the $f_i()$ might equal a continuous function such as a polynomial, or a discrete-valued Boolean function such as an OR gate.

Chapter 53

Marginalizer Nodes

Suppose we have a bnet node $\underline{x}$ that has multiple components. For instance, suppose $\underline{x} = (\underline{x}_1, \underline{x}_2, \underline{x}_3)$. Then, we can define 3 **marginalizer nodes** with TPMs, shown in blue, as follows

$$P(\underline{x}_i = \xi_i | \underline{x} = (x_1, x_2, x_3)) = \delta(\xi_i, x_i) \tag{53.1}$$

for $\xi_i \in val(\underline{x}_i)$ and $i = 1, 2, 3$.

Figs.53.1 and 53.2 show 3 different styles for representing marginalizer nodes graphically. In this book, we will use styles (b) or (c). Style (b) is the least ambiguous.

Figure 53.1: 3 styles for representing marginalizer nodes in an arbitrary bnet.

Figure 53.2: 3 styles for representing marginalizer nodes in an LDEN bnet (see Chapter 52).

Chapter 54

Markov Blankets

This chapter is based on the Wikipedia article, Ref.[161]. Markov blankets and Markov boundaries of bnets were apparently invented by Judea Pearl. His 1988 book Ref.[60], instead of a research paper, is usually given as the original reference.

Figure 54.1: In a bnet, the minimal Markov blanket, a.k.a. Markov boundary, of node $\underline{a}$.

We will treat vectors of random variables as if they were sets when using the $\in$, $\subset$ and $-$ operations. For example, if $\underline{x} = (\underline{x}_0, \underline{x}_1, \underline{x}_2, \underline{x}_3)$ and $\underline{b} = (\underline{x}_1, \underline{x}_2)$, then $\underline{x}_1 \in \underline{b} \subset \underline{x}$ and $\underline{x} - \underline{b} = (\underline{x}_0, \underline{x}_3)$.

Below, $H(\underline{a} : \underline{b}|\underline{c})$ denotes the conditional mutual information of random variables $\underline{a}$ and $\underline{b}$ conditioned on random variable $\underline{c}$. $H(\underline{a} : \underline{b}|\underline{c})$ is used in Shannon Information Theory, where it is defined by

$$H(\underline{a} : \underline{b}|\underline{c}) = \sum_{a,b,c} P(a,b,c) \ln \frac{P(a,b|c)}{P(a|c)P(b|c)} \ . \tag{54.1}$$

$H(\underline{a} : \underline{b}|\underline{c}) = 0$ iff $\underline{a}$ and $\underline{b}$ are independent (uncorrelated) when $\underline{c}$ is held fixed.

Suppose $\underline{a} \in \underline{X}$, $\underline{B} \subset \underline{X}$, but $\underline{a} \notin \underline{B}$. Then $\underline{B}$ is a **Markov blanket** of $\underline{a}$ if

$$H(\underline{a} : \underline{X} - \underline{a}|\underline{B}) = 0 \ . \tag{54.2}$$

In other words, one may assume that $\underline{a}$ depends on $\underline{B}$ only, and is independent of all random variables in $\underline{X} - (\underline{a} \cup \underline{B})$.

The **minimal Markov blanket** is called the **Markov boundary**.

In a bnet, the Markov boundary of a node $\underline{a}$, contains:

1. the parents of $\underline{a}$,
2. the children of $\underline{a}$,
3. the parents, other than $\underline{a}$, of the children of $\underline{a}$.

This is illustrated in Fig.54.1.

From the d-separation theorem (see Chapter 24), we get an intuitive motivation for the definition of Markov boundary. By conditioning on the parents and children of node $\underline{a}$, we block almost all, but not all, information from reaching node $\underline{a}$. The reason not all info is blocked is that conditioning on a child $\underline{c}$ of $\underline{a}$ creates paths from a parent of $\underline{c}$ to $\underline{c}$ to $\underline{a}$, wherein node $\underline{c}$ acts as a conditioned collider. To block such paths, we must condition on the parents of $\underline{c}$ too.

Chapter 55

Markov Chain Monte Carlo (MCMC)

Monte Carlo methods are methods for using random number generation to sample probability distributions. The subject of Monte Carlo methods has many branches, as you can see from its Wikipedia category list, Ref.[164]. MCMC (Markov Chain Monte Carlo) is just one of those branches, albeit a major one. Metropolis-Hastings (MH) sampling is a very important MCMC method. Gibbs sampling is a special case of MH sampling. This chapter covers both, MH and Gibbs sampling. It also covers a few other types of sampling.

Throughout this chapter, we use $P_{\underline{x}} : val(\underline{x}) \rightarrow [0,1]$ to denote the target probability distribution that we wish to obtain samples from.

55.1 Inverse Cumulative Sampling

For more info about this topic and some original references, see Ref.[147].

This is one of the simplest methods for obtaining samples from a probability distribution $P_{\underline{x}}$, but it requires knowledge of the inverse cumulative distribution of $P_{\underline{x}}$, which is often not available.

The **cumulative distribution** function is defined by:

$$CUM_{\underline{x}}(x) = P(\underline{x} < x) = \int_{x'<x} dx' \; P_{\underline{x}}(x') \; . \tag{55.1}$$

Note that

$$P_{\underline{x}}(x) = \frac{d}{dx} CUM_{\underline{x}}(x) \; . \tag{55.2}$$

For $t = 0, 1, \ldots, T-1$, let

$\underline{u}^{(t)} \in [0,1]$= random variable, uniformly distributed over $[0,1]$.

$\vec{\underline{x}}^{(t)} = (\underline{x}^{(t)}[\sigma])_{\sigma=0,1,\ldots,nsam(t)-1}$ where $\underline{x}^{(t)}[\sigma] \in val(\underline{x})$ for all σ. Vector of samples collected up to time t.

Figure 55.1: bnet for Inverse Cumulative Sampling

The TPMs, printed in blue, for bnet Fig.55.1, are as follows:

$$P(u^{(t)}) = 1 \tag{55.3}$$

$$P(\vec{x}^{(t)}|\vec{x}^{(t-1)}, u^{(t)}) = \delta(\quad \vec{x}^{(t)}, [\vec{x}^{(t-1)}, CUM_{\underline{x}}^{-1}(u^{(t)})] \quad) \tag{55.4}$$

Motivation

Figure 55.2: Motivation for Inverse Cumulative Sampling.

See Fig.55.2.

Note that if $\underline{u}$ is uniformly distributed over the interval $[0, 1]$ and $a \in [0, 1]$, then

$$P(\underline{u} < a) = a \ . \tag{55.5}$$

Thus

$$\begin{aligned} P(CUM_{\underline{x}}^{-1}(\underline{u}) < x) &= P(\underline{u} < CUM_{\underline{x}}(x)) && (55.6) \\ &= CUM_{\underline{x}}(x) \ . && (55.7) \end{aligned}$$

Therefore,

$$dP(CUM_{\underline{x}}^{-1}(\underline{u}) < x) = P_{\underline{x}}(x)dx \ . \tag{55.8}$$

55.2 Rejection Sampling

For more info about this topic and some original references, see Ref.[178].

This method samples from a "candidates" probability distribution $P_{\underline{c}} : val(\underline{x}) \to [0,1]$, in cases where sampling directly from the target probability distribution $P_{\underline{x}} : val(\underline{x}) \to [0,1]$ is not possible.

Figure 55.3: bnet for Rejection Sampling

For $t = 0, 1, \ldots, T-1$, let
$\underline{u}^{(t)} \in [0,1]$= random variable, uniformly distributed over $[0,1]$.
$\underline{a}^{(t)} \in \{0,1\}$= accept candidate? (no=0, yes=1)
$\underline{c}^{(t)} \in val(\underline{x})$= sample that is a candidate for being accepted
$\vec{\underline{x}}^{(t)} = (\underline{x}^{(t)}[\sigma])_{\sigma=0,1,\ldots,nsam(t)-1}$ where $\underline{x}^{(t)}[\sigma] \in val(\underline{x})$ for all σ. Vector of samples collected up to time t.

This algorithm requires a priori definition of a candidate probability distribution $P_{\underline{c}} : val(\underline{x}) \to \mathbb{R}$ such that

$$P_{\underline{x}}(x) < \beta P_{\underline{c}}(x) \tag{55.9}$$

for all $x \in val(\underline{x})$, for some $\beta \in \mathbb{R}$.

The TPMs, printed in blue, for bnet Fig.55.3, are as follows:

$$P(u^{(t)} = u) = 1 \tag{55.10}$$

$$P(\underline{c}^{(t)} = c) = P_{\underline{c}}(c) \tag{55.11}$$

$$P(\underline{a}^{(t)} = a|\underline{c}^{(t)} = c, \underline{u}^{(t)} = u) = \begin{cases} \delta(a,0) & \text{if } u\beta P_{\underline{c}}(c) \geq P_{\underline{x}}(c) \\ \delta(a,1) & \text{if } u\beta P_{\underline{c}}(c) < P_{\underline{x}}(c) \end{cases} \tag{55.12}$$

$$P(\vec{x}^{(t)}|\vec{x}^{(t-1)}, \underline{a}^{(t)} = a, \underline{c}^{(t)} = c) = \begin{cases} \delta(\vec{x}^{(t)}, \vec{x}^{(t-1)}) & \text{if } a = 0 \\ \delta(\vec{x}^{(t)}, [\vec{x}^{(t-1)}, c]) & \text{if } a = 1 \end{cases} \tag{55.13}$$

This last equation is only defined for $t > 0$. For $t = 0$, the left hand side reduces to $P(\vec{x}^{(0)})$ which must be specified a priori.

Motivation

Figure 55.4: Motivation for Rejection Sampling.

See Fig.55.4.

55.3 Metropolis-Hastings Sampling

For more info about this topic and some original references, see Refs.[5] and [162].

An advantage of this method is that it can sample unnormalized probability distributions $(constant)P_{\underline{x}}$ because it only uses ratios of $P_{\underline{x}}$ at two different points. Another advantage of this method is that it scales much better than other sampling methods as the number of dimensions of the sampled variable $\underline{x}$ increases.

This method produces samples that take a finite amount of time to reach steady state. The samples are also theoretically correlated instead of being i.i.d. as one desires. To mitigate for the steady state problem, one discards an initial set of samples (the "burn-out" period). To mitigate for the correlation problem, one calculates the autocorrelation between the samples and keeps only samples separated by a time interval after which the samples cease to be autocorrelated to a good approximation.

For $t = 0, 1, \ldots, T-1$, let
$\underline{u}^{(t)} \in [0,1]$= random variable, uniformly distributed over $[0,1]$.
$\underline{a}^{(t)} \in \{0,1\}$= accept candidate? (no=0, yes=1)
$\underline{c}^{(t)} \in val(\underline{x})$= sample that is a candidate for being accepted

Figure 55.5: bnet for Metropolis-Hastings Sampling

$\underline{m}^{(t)} \in val(\underline{x})$= memory of last accepted sample

$\vec{\underline{x}}^{(t)} = (\underline{x}^{(t)}[\sigma])_{\sigma=0,1,\ldots,nsam(t)-1}$ where $\underline{x}^{(t)}[\sigma] \in val(\underline{x})$ for all σ. Vector of samples collected up to time t.

A **proposal TPM** $P_{\underline{c}|\underline{x}} : val(\underline{x})^2 \to [0,1]$ must be specified a priori for this algorithm.

The TPMs, printed in blue, for bnet Fig.55.5, are as follows:

$$P(u^{(t)} = u) = 1 \tag{55.14}$$

$$P(\underline{c}^{(t)} = c|\underline{m}^{(t)} = m) = P_{\underline{c}|\underline{x}}(c|m) \tag{55.15}$$

$$P(\underline{a}^{(t)} = a|\underline{c}^{(t)} = c, \underline{u}^{(t)} = u, \underline{m}^{(t)} = m) = \begin{cases} \delta(a,0) & \text{if } u \geq \alpha(c|m) \\ \delta(a,1) & \text{if } u < \alpha(c|m) \end{cases} \tag{55.16}$$

where the **acceptance probability** α is defined as

$$\alpha(c|m) = \min\left(1, \frac{P_{\underline{c}|\underline{x}}(m|c)P_{\underline{x}}(c)}{P_{\underline{c}|\underline{x}}(c|m)P_{\underline{x}}(m)}\right) . \tag{55.17}$$

Note that if the proposal distribution is symmetric, then

$$\alpha(c|m) = \min\left(1, \frac{P_{\underline{x}}(c)}{P_{\underline{x}}(m)}\right) . \tag{55.18}$$

$$P(\vec{x}^{(t)}|\vec{x}^{(t-1)}, \underline{a}^{(t)} = a, \underline{c}^{(t)} = c) = \begin{cases} \delta(\vec{x}^{(t)}, \vec{x}^{(t-1)}) & \text{if } a = 0 \\ \delta(\vec{x}^{(t)}, [\vec{x}^{(t-1)}, c]) & \text{if } a = 1 \end{cases} \tag{55.19}$$

This last equation is only defined for $t > 0$. For $t = 0$, the left hand side reduces to $P(\vec{x}^{(0)})$ which must be specified a priori.

$$P(\underline{m}^{(t)} = m|\vec{x}^{(t)}) = \delta(m, \text{last component of } \vec{x}^{(t)}) \ . \tag{55.20}$$

This last equation is only defined for $t > 0$. For $t = 0$, the left hand side reduces to $P(\underline{m}^{(0)} = m)$ which must be specified a priori.

Motivation

Figure 55.6: Motivation for Metropolis-Hastings Sampling.

See Fig.55.6.

Consider a time homogeneous (its TPM is the same for all times) Markov chain with TPM $P(x'|x) = [T]_{x',x}$. Its **stationary distribution**, if it exists, is defined as

$$\pi = \lim_{n\to\infty} T^n \pi_0 \ . \tag{55.21}$$

Suppose the prob distribution $P_{\underline{x}}(x)$ that we wish to sample from satisfies

$$P_{\underline{x}}(x) = \pi(x) \ . \tag{55.22}$$

Reversibility (detailed balance): For all $x, x' \in val(\underline{x})$,

$$P(x'|x)\pi(x) = P(x|x')\pi(x') \ . \tag{55.23}$$

Detailed balance is a sufficient (although not necessary) condition for a unique stationary prob distribution π to exist.[1]

[1] As explained lucidly in Ref.[5], besides detailed balance, 2 other properties must also be satisfied by the Markov chain, irreducibility and aperiodicity. However, because of how it is constructed, the Metropolis-Hastings algorithm automatically produces a Markov chain that has those 2 properties.

Let

$$P(x'|x) = P(\underline{a} = 1|x', x)P_{\underline{c}|\underline{x}}(x'|x) + \delta(x, x')P(\underline{a} = 0|x) \ , \tag{55.24}$$

where

$$P(\underline{a} = 0|x) = \sum_{x'} P(\underline{a} = 0|x', x)P_{\underline{c}|\underline{x}}(x'|x) \ . \tag{55.25}$$

Claim 84 *If*

$$P(\underline{a} = 1|x', x) = \alpha(x'|x) \ , \tag{55.26}$$

then detailed balance is satisfied.

proof: Assume $x \neq x'$.

$$\begin{aligned} P(x'|x)P(x) &= P(\underline{a} = 1|x', x)P_{\underline{c}|\underline{x}}(x'|x)P_{\underline{x}}(x) && (55.27) \\ &= \min\left(1, \frac{P_{\underline{c}|\underline{x}}(x|x')P_{\underline{x}}(x')}{P_{\underline{c}|\underline{x}}(x'|x)P_{\underline{x}}(x)}\right) P_{\underline{c}|\underline{x}}(x'|x)P_{\underline{x}}(x) && (55.28) \\ &= \min\Big(P_{\underline{c}|\underline{x}}(x'|x)P_{\underline{x}}(x), P_{\underline{c}|\underline{x}}(x|x')P_{\underline{x}}(x')\Big) && (55.29) \\ &= P(x|x')P(x') && (55.30) \end{aligned}$$

QED

55.4 Gibbs Sampling

For more info about this topic and some original references, see Ref.[141].

Gibbs sampling is a special case of Metropolis-Hastings sampling. Gibbs sampling is ideally suited for application to a bnet, because it is stated in terms of the conditional prob distributions of N random variables, and conditional prob distributions are part of the definition of a bnet.

Consider a bnet with nodes $\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{N-1}$

Identify the random variable $\underline{x} = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{N-1})$ with the random variable $\underline{x}$ used in Metropolis-Hastings sampling. For Gibbs sampling, we use the following proposal distribution:

$$P_{\underline{c}|\underline{x}}(c|m) = \prod_{j=0}^{N-1} P(c_j \mid [m_i]_{i \neq j}) \ . \tag{55.31}$$

Eq.(55.31) can be simplified using Markov Blankets (see Chapter 54) to the following:

$$P_{\underline{c}|\underline{x}}(c|m) = \prod_{j=0}^{N-1} P(c_j \mid [m_i : \forall i \ni \underline{x}_i \in MB(\underline{x}_j)]) \ , \tag{55.32}$$

where, for any node $\underline{a}$, we denote its Markov blanket by $MB(\underline{a})$.

Figure 55.7: In Gibbs sampling, the proposal distribution $P_{\underline{c}|\underline{x}}$ can be defined by making the components $c_j^{(t)}$ of candidate sample $c^{(t)}$ depend on the previous components $(c_i^{(t)})_{i<j}$.

An alternative proposal distribution that leads to much faster convergence is as follows. The idea is to make the components $c_j^{(t)}$ of candidate sample $c^{(t)}$ depend on the previous components $(c_i^{(t)})_{i<j}$. See the bnet Fig.55.7. The TPM for the nodes of that bnet are

$$P(\underline{c}_j^{(t)} = c_j \mid (\underline{c}_i^{(t)})_{i<j} = (c_i)_{i<j}, \underline{m}^{(t-1)} = m) = P(c_j|(c_i)_{i<j}, (m_i)_{i>j}) \tag{55.33}$$

for $j = 0, 1, \ldots, N-1$. This implies

$$P_{\underline{c}|\underline{x}}(\underline{c}^{(t)} = c|\underline{m}^{(t-1)} = m) = \prod_{j=0}^{N-1} P(c_j|(c_i)_{i<j}, (m_i)_{i>j}) \ . \tag{55.34}$$

As before, we can condition only on the Markov blanket of each node $\underline{x}_j$.

$$P_{\underline{c}|\underline{x}}(\underline{c}^{(t)} = c|\underline{m}^{(t-1)} = m) = \prod_{j=0}^{N-1} P(c_j|(c_i)_{i<j}, (m_i)_{i>j}, \text{ use only } c_i \text{ and } m_i \ni \underline{x}_i \in MB(\underline{x}_j)) \ . \tag{55.35}$$

55.5 Importance Sampling

For more info about this topic and some original references, see Ref.[145].

Suppose random variables $\underline{x}[\sigma] \in val(\underline{x})$ for $\sigma = 0, 1, \ldots, nsam - 1$ are i.i.d. with probability distribution $P_{\underline{x}}$. Then

$$E_{\underline{x}}[f(x)] \approx \frac{1}{nsam} \sum_{\sigma=0}^{nsam-1} f(x[\sigma]) \tag{55.36}$$

for any $f : val(\underline{x}) \to \mathbb{R}$. Sometimes, instead of using i.i.d. samples $\underline{x}[\sigma] \in val(\underline{x})$ where $\underline{x}[\sigma] \sim P_{\underline{x}}$, we wish to use i.i.d. samples $\underline{y}[\sigma] \in val(\underline{x})$ where $\underline{y}[\sigma] \sim P_{\underline{y}}$.

$$E_{\underline{x}}[f(\underline{x})] = \sum_x P_{\underline{x}}(x) f(x) \tag{55.37}$$

$$= \sum_x P_{\underline{y}}(x)) \frac{P_{\underline{x}}(x)}{P_{\underline{y}}(x)} f(x) \tag{55.38}$$

$$= E_{\underline{y}}[\frac{P_{\underline{x}}(y)}{P_{\underline{y}}(y)} f(y)] \tag{55.39}$$

Sampling from $P_{\underline{y}}(y)$ instead of $P_{\underline{x}}(x)$ might reduce (or increase) variance for a particular $f : val(\underline{x}) \to \mathbb{R}$.

$$Var_{\underline{x}}[f(x)] = E_{\underline{x}}[(f(x))^2] - (E_{\underline{x}}[f(x)])^2 \tag{55.40}$$

$$Var_{\underline{y}}[\frac{P_{\underline{x}}(y)}{P_{\underline{y}}(y)} f(y)] = E_{\underline{y}}[(\frac{P_{\underline{x}}(y)}{P_{\underline{y}}(y)} f(y))^2] - (E_{\underline{y}}[\frac{P_{\underline{x}}(y)}{P_{\underline{y}}(y)} f(y)])^2 \tag{55.41}$$

$$= E_{\underline{x}}[\frac{P_{\underline{x}}(x)}{P_{\underline{y}}(x)} (f(x))^2] - (E_{\underline{x}}[f(x)])^2 \tag{55.42}$$

Chapter 56

Markov Chains

A Markov Chain is simply a bnet with the graph structure of a chain. For example, Fig.56.1 shows a chain with $n = 4$ nodes.

$$\underline{x}_0 \longrightarrow \underline{x}_1 \longrightarrow \underline{x}_2 \longrightarrow \underline{x}_3$$

Figure 56.1: Markov chain with $n = 4$ nodes.

Because of its graph structure, the TPM of each node only depends on the state of the previous node:

$$P(x_t|(x_a)_{a\neq t}) = P(x_t|x_{t-1}) \ , \tag{56.1}$$

where $(x_a)_{a\neq t}$ are all the nodes except x_t itself and $t = 1, 2, \ldots, n-1$.

If there exists a single TPM $P_{\underline{x}_1|\underline{x}_0}$ such that

$$P(x_t|x_{t-1}) = P_{\underline{x}_1|\underline{x}_0}(x_t|x_{t-1}) \tag{56.2}$$

for $t = 1, 2, \ldots, n-1$, then we say that the Markov chain is **time homogeneous**.

Claim 85 *(Data Processing Inequality (DPI))*

Consider a Markov chain $\underline{x}_0 \to \underline{x}_1 \cdots \to \underline{x}_{n-1}$. *Suppose* $0 \leq a < m < b \leq n-1$. *Then*

$$H(\underline{x}_a : \underline{x}_b) \leq \min[H(\underline{x}_a : \underline{x}_m), H(\underline{x}_m : \underline{x}_b)] \tag{56.3}$$

See Ref.[128] for references where the DPI is proven. This inequality confirms our intuitive expectations that the information transmitted (i.e., the mutual information(MI)) from $\underline{a}$ to $\underline{b}$ (or vice versa since MI is symmetric) is smaller or equal to the one transmitted from $\underline{a}$ to $\underline{m}$ or from $\underline{m}$ to $\underline{b}$ because $\underline{a}$ and $\underline{b}$ are "farther apart" and "some info can get lost during transmission through the mediator node $\underline{m}$".

Chapter 57

Mediation Analysis

This chapter is mostly based on Refs.[63, 61] by Pearl.

To fully understand this chapter, the reader should first read Chapter 12 where do and imagine operators are defined.

Figure 57.1: DEN bnets G_0 and G are used to discuss mediation analysis. In graph G_0, the external variables are independent, whereas in graph G they are not.

The term "mediation analysis" (MA) refers to the analysis by Pearl of the DEN bnet G in Fig.57.1. We will discuss MA in terms of DEN bnets, just as Pearl does. However, note that much of what we will say applies also to the general (i.e., not necessarily a DEN) bnet G_{gen} show in Fig. 57.2.

In the DEN bnet G, node $\underline{d}$ influences node $\underline{y}$ both directly and via the mediator node $\underline{m}$. The structural equations for G are of the form:

$$\underline{d} = \underline{u}_{\underline{d}} \tag{57.1a}$$

$$\underline{m} = f_{\underline{m}}(\underline{d}, \underline{u}_{\underline{m}}) \tag{57.1b}$$

$$\underline{y} = f_{\underline{y}}(\underline{d}, \underline{m}, \underline{u}_{\underline{y}}) \ . \tag{57.1c}$$

G_{gen}

Figure 57.2: General bnet used to discuss mediation analysis. Node $\underline{u}$ is a hidden confounder.

Thus,

$$\underline{y} = f_{\underline{y}}(u_{\underline{d}}, f_{\underline{m}}(\underline{u}_{\underline{d}}, \underline{u}_{\underline{m}}), \underline{u}_{\underline{y}}) \ . \tag{57.2}$$

Figure 57.3: Graph G of Fig.57.1 after applying do operator $\mathcal{D}_{\underline{d}=5}$ and imagine operator $\mathcal{I}_{\underline{d}\to\underline{m}}(5)$.

If we apply $\mathcal{D}_{\underline{d}=5}G$ to Eqs.(57.1), we get

$$\underline{d} = 5 \tag{57.3a}$$
$$\underline{m} = f_{\underline{m}}(\underline{d}, \underline{u}_{\underline{m}}) \tag{57.3b}$$
$$\underline{y} = f_{\underline{y}}(\underline{d}, \underline{m}, \underline{u}_{\underline{y}}) \ . \tag{57.3c}$$

Eqs.57.3 are represented graphically in Fig.57.3. We will often denote the random

variable $\underline{y}$ in Eqs.(57.3) by the more explicit symbol $\underline{y}_{\mathcal{D}_{\underline{d}=5}G}$. Pearl often refers to this $\underline{y}$ by the less explicit symbol Y_5 or $Y_5(u)$ where $u = (u_{\underline{m}}, u_{\underline{y}}) = u_{!\underline{d}}$.

If we apply $\mathcal{I}_{\underline{d}\to\underline{m}}(5)G$ to Eqs.(57.1), we get

$$\underline{d} = \underline{u}_{\underline{d}} \tag{57.4a}$$
$$\underline{m} = f_{\underline{m}}(5, \underline{u}_{\underline{m}}) \tag{57.4b}$$
$$\underline{y} = f_{\underline{y}}(\underline{d}, \underline{m}, \underline{u}_{\underline{y}}) \ . \tag{57.4c}$$

Eqs.57.4 are represented graphically in Fig.57.3. We will often denote the random variable $\underline{y}$ in Eqs.(57.4) by the more explicit symbol $\underline{y}_{\mathcal{I}_{\underline{d}\to\underline{m}}(5)G}$. Pearl often refers to this $\underline{y}$ by the less explicit symbol Y_5 or $Y_5(u)$ where $u = (u_{\underline{d}}, u_{\underline{m}}, u_{\underline{y}})$.

Figure 57.4: Graph G of Fig.57.1 after applying the do operators $\mathcal{D}_{\underline{d}=d}$ and $\mathcal{D}_{\underline{d}=d}\mathcal{D}_{\underline{m}=m}$.

Define

$$\mathcal{Y}_d = E[\underline{y}_{\mathcal{D}_{\underline{d}=d}G}] = \sum_y yP(y|\mathcal{D}\underline{d} = d) \tag{57.5}$$

and

$$\mathcal{Y}_d^m = E[\underline{y}_{\mathcal{D}_{\underline{d}=d}\mathcal{D}_{\underline{m}=m}G}] = \sum_y yP(y|\mathcal{D}\underline{d} = d, \mathcal{D}\underline{m} = m) \tag{57.6}$$

The two DEN bnets $\mathcal{D}_{\underline{d}=d}G$ and $\mathcal{D}_{\underline{d}=d}\mathcal{D}_{\underline{m}=m}G$ used in the definitions of $\mathcal{Y}_d$ and $\mathcal{Y}_d^m$ are given in Fig.57.4.

Now define the Total Effect (TE), and the Controlled Direct Effect (CDE) by

$$TE = \mathcal{Y}_1 - \mathcal{Y}_0 \tag{57.7}$$
$$CDE(m) = \mathcal{Y}_1^m - \mathcal{Y}_0^m \tag{57.8}$$

Figure 57.5: Graph G of Fig.57.1 after applying the imagine operator $\mathcal{I}$ to arrow $\underline{d} \to \underline{m}$ and the do operator to node $\underline{d}$.

Define

$$\overline{\mathcal{Y}}_d^{d'} = E[\underline{y}_{\mathcal{D}_{\underline{d}=d}\mathcal{I}_{\underline{d}\to\underline{m}}(d')G}] = \sum_y yP(y|\mathcal{D}\underline{d} = d, \mathcal{I}_{\underline{m}}\underline{d} = d') \tag{57.9}$$

Fig.57.5 shows the diagram $\mathcal{D}_{\underline{d}=d}\mathcal{I}_{\underline{d}\to\underline{y}}(d')G$ used in the definition of $\overline{\mathcal{Y}}_d^{d'}$.

Note that

$$\overline{\mathcal{Y}}_d^{d'} = \sum_m \mathcal{Y}_d^m P(\underline{m} = m|d') \tag{57.10}$$

if there is no arrow $\underline{u}_{\underline{m}} \to \underline{m}$. This expresses $P(y|\mathcal{D}\underline{d} = d, \mathcal{I}\underline{d} = d')$ in terms of $P(y|\mathcal{D}\underline{d} = d, \mathcal{D}\underline{m} = m)$.

Define

$$\overline{\mathcal{Y}}_d^{d'-} = \overline{\mathcal{Y}}_d^{d'} - \overline{\mathcal{Y}}_{d'}^{d'} \tag{57.11}$$

and

$$\overline{\mathcal{Y}}_{d-}^{d'} = \overline{\mathcal{Y}}_d^{d'} - \overline{\mathcal{Y}}_d^{d} \tag{57.12}$$

Now define the Natural Direct Effect (NDE), and the Natural Indirect Effect (NIE) by

$$NDE_d^{d'} = \overline{\mathcal{Y}}_d^{d'-} \tag{57.13}$$

$$NIE_d^{d'} = \overline{\mathcal{Y}}_{d-}^{d'} . \tag{57.14}$$

Note that

$$NDE_1^0 - NIE_1^0 = -\overline{\mathcal{Y}}_0^0 + \overline{\mathcal{Y}}_1^1 \tag{57.15}$$

$$= TE . \tag{57.16}$$

Figure 57.6: LDEN that is used to discuss mediation analysis.

Linear Case
Consider the LDEN of Fig.57.6. One has

$$\underline{d} = \underline{u_d} \tag{57.17a}$$
$$\underline{m} = \alpha \underline{d} + \underline{u_m} \tag{57.17b}$$
$$\underline{y} = \gamma \underline{d} + \beta \underline{m} + \underline{u_y} \ . \tag{57.17c}$$

$$\mathcal{Y}_d = (\gamma + \alpha\beta) d \tag{57.18}$$

$$\mathcal{Y}_d^m = \gamma d + \beta m \tag{57.19}$$

$$\overline{\mathcal{Y}}_d^{d'} = \gamma d + \alpha\beta d' \tag{57.20}$$

$$TE = \mathcal{Y}_1 - \mathcal{Y}_0 = \gamma + \alpha\beta \tag{57.21}$$

$$CDE(m) = \mathcal{Y}_1^m - \mathcal{Y}_0^m = \gamma \tag{57.22}$$

$$NDE_1^0 = \overline{\mathcal{Y}}_1^{0-} = \gamma \tag{57.23}$$

$$NIE_1^0 = \overline{\mathcal{Y}}_{1-}^0 = -\alpha\beta \tag{57.24}$$

As expected, $NDE_1^0 - NIE_1^0 = TE$.

TE and the "controlled effect" $CDE(m)$ contain do operators but no imagine operators so one can do do-intervention experiments to calculate them. On the other hand, the "natural effects" NDE_1^0 and NIE_1^0 contain imagine operators (and therefore counterfactual distributions) so it's not obvious how to calculate them, or even whether it is possible to do so. The next claim shows how to calculate $\overline{\mathcal{Y}}_d^{d'}$ in certain

cases. In technical jargon, the claim gives sufficient conditions for $\mathcal{DI}$-identifiability of $\overline{\mathcal{Y}}_d^{d'}$. NDE_1^0 and NIE_1^0 can be calculated once $\overline{\mathcal{Y}}_d^{d'}$ is known.

Claim 86 *(Unconfounded Mediation, from Ref.[63])*

If *then*

$$\begin{tikzcd} & \underline{m} \arrow[dr] & \\ \underline{d} \arrow[ur] \arrow[rr] & & \underline{y} \end{tikzcd}$$

$$P(y|\mathcal{D}\underline{d} = d, \mathcal{I}_{\underline{m}}\underline{d} = d') = \sum_m P(y|d,m)P(m|d') \tag{57.25}$$

$$\begin{tikzcd} \mathcal{I}\underline{d} = d' \arrow[dr] & & \mathcal{I}\underline{d} = d' \arrow[r] & \sum m \arrow[dr] & \\ \mathcal{D}\underline{d} = d \arrow[r] & y & = & \mathcal{D}\underline{d} = d \arrow[rr] & & y \end{tikzcd} \tag{57.26}$$

proof: See Claim 57.
QED

Claim 87 *(Mediation with universal prior $\underline{\xi}$ and universal confounder $\underline{u}$, from Ref.[63])*

If *then*

$$\begin{tikzcd} & \underline{u} \arrow[dl] \arrow[d] \arrow[ddr] & \\ \underline{\xi} \arrow[r] \arrow[d] \arrow[drr] & \underline{m} \arrow[dr] & \\ \underline{d} \arrow[ur] \arrow[rr] & & \underline{y} \end{tikzcd}$$

$$P(y|\mathcal{D}\underline{d} = d, \mathcal{I}_{\underline{m}}\underline{d} = d') = \sum_\xi \sum_m P(y|d,m,\xi)P(m|d',\xi)P(\xi) \tag{57.27}$$

$$\begin{tikzcd} \mathcal{I}\underline{d} = d' \arrow[dr] & & \mathcal{I}\underline{d} = d' \arrow[rr, bend right] & \mathbb{E}\xi \arrow[r] \arrow[drr] & \sum m \arrow[dr] & \\ \mathcal{D}\underline{d} = d \arrow[r] & y & = & \mathcal{D}\underline{d} = d \arrow[rrr] & & & y \end{tikzcd} \tag{57.28}$$

proof: See Claim 58.
QED

Actually,

$$\sum_m P(y|d,m)P(m|d') = \sum_\xi \sum_m P(y|d,m,\xi)P(m|d',\xi)P(\xi) \tag{57.29}$$

so the adjustment formulae in Claims 86 and 87 are equivalent, but if the available dataset contains info about ξ, then the adjustment formula that uses that ξ info should be used, as it will be more sensitive to deviations from the DAG model being hypothesized.

Chapter 58

Mendelian Randomization

Mendelian Randomization (MR) is an application of the method of Instrumental Variables (IVs) to genetics. It's considered an "observational study". It's used as a substitute to an RCT (i.e., experimental study), when an RCT can't be done.

IVs are discussed in Chapter 44. Here is a quick review of the essential points of that chapter.

Figure 58.1: MR assumes this bnet

Figure 58.2: Pictorial representation of $ATE = y(1) - y(0) = \delta$.

The bnet of Fig.58.1 obeys the following equations:

$$\begin{cases} \underline{y} = \delta \underline{d} + \mu \underline{u} + \underline{u}_y \\ \underline{d} = \alpha \underline{a} + \nu \underline{u} + \underline{u}_d \end{cases} \tag{58.1}$$

where $\underline{u}_y$ and $\underline{u}_d$ are external, uncorrelated noises.

If one solves for δ, one gets[1]

$$ATE = \delta = \frac{\frac{\partial \underline{y}}{\partial \underline{a}}}{\frac{\partial \underline{d}}{\partial \underline{a}}} \tag{58.2}$$

This formula for δ can be estimated using Linear Regression. See Fig.58.2 for a pictorial representation of δ.

What these random variables stand for in MR:

$\underline{a}$: genotype (inherited genetic variant trait). This is the IV.
$\underline{u}$: hidden variable, "confounders" (good controls)
$\underline{d} \in \{0, 1\}$: exposure
$\underline{y} \in \{0, 1\}$: outcome

Example

$\underline{a}$: has genotype that is strongly associated with heavy smokers?
$\underline{u}$: city dweller?
$\underline{d}$: smoker?
$\underline{y}$: died of lung cancer?

Assumptions of MR (should be tested)

- $\underline{a} \to \underline{d}$, but no $\underline{a} \to \underline{y}$
- no confounder $\underline{c}$ such that $\underline{c} \to \underline{a}, \underline{y}$
- no feedback (a.k.a. reverse causation) arrows $\underline{y} \to \underline{a}$.

Using genotype for $\underline{a}$ makes these assumptions more likely.

[1] As usual in this book, we define $\frac{\partial \underline{y}}{\partial \underline{x}} = \frac{\langle \underline{x}, \underline{y} \rangle}{\langle \underline{x}, \underline{x} \rangle}$ for any random variables $\underline{x}, \underline{y}$.

Chapter 59

Message Passing and Bethe Free Energy

This chapter is based on Refs. [101] and [200].

59.1 2MRFs

Figure 59.1: Example of factor graph for a 2MRF. In this figure, a boxed Δ between variables x_i and x_j, denotes the propagator function $\Delta(x_i, x_j)$. Also, a boxed Δ connected to a single variable x_i, denotes the function $\Delta(x_i, k)$, where k is a fixed number. If we view this factor graph as a representation of a 3 dimensional device with the leaf nodes Δ in the back part and everything else in the front part, then the back part can represent the light inputs from a scene, and the front part can represent the analysis being done with the light inputs.

Factor graphs are discussed in Chapter 29.

A pairwise Markov Random Field (2MRF) is a statistical model whose probability distribution is of the form

$$p(x) = \mathcal{N}(!x) \prod_{i-j} \Delta(x_i, x_j) \tag{59.1}$$

where $x = (x_1, x_2, \ldots, x_n)$, where $i - j$ represents the edge that connects nodes i and j in an undirected graph G, and where the product is over all edges of G. $p(x)$ can be represented graphically by a factor graph.

Fig.59.1 shows an example of a 2MRF represented as a factor graph.

The **Ising model** is another prototypical example of a 2MRF. Let T be the temperature of a system,

$$\theta = \frac{1}{T} \ , \tag{59.2}$$

and $x = (x_1, x_2, \ldots, x_{nx}) \in \{-1, 1\}^{nx}$. The Ising model is defined for $J, h(x_i) \in \mathbb{R}$ by

$$\epsilon(x) = -J \sum_{i-j} x_i x_j - \sum_i h(x_i) \tag{59.3}$$

$$P(x) = \frac{e^{\frac{-\epsilon(x)}{T}}}{Z_\theta} \tag{59.4}$$

Note that a bnet can be easily converted to an equivalent 2MRF. You just set $\Delta(x_i, k) = p(x_i)$ if x_i is a root node of the bnet, and $\Delta(x_i, x_j) = p(x_i|x_j)$ if x_i isn't a root node. One must then choose the arrow directions of the bnet. A natural choice is to choose the bnet arrows so that they all point from an x_i to a Δ.

In what follows, we shall use **messages (a.k.a. cavity fields)** $m_{a \Rightarrow i}(x_i)$, Note that in our notation for messages, the letter i appears twice, and both occurrences are next to each other. This is always the case in our notation for messages.

Henceforth, we will refer to the functions $\Delta(x_i, x_j)$ as **propagators**, to make contact with Physics.

Henceforth, we shall use the notation ∂i to denote all nodes j that are neighbors of node i. ∂i is called the **neighborhood or boundary** of i.

59.2 Message Passing Intuition

Next we present some results that will be derived later in the chapter, but which we will first present now without proof, and explain how they make sense intuitively. Assume that the functions $\Delta(x_i, x_j)$ are known for all edges $i - j$, and that the full probability distribution $p(x)$ of the model is given by Eq.(59.1).

1. Calculate message $m_{j \Rightarrow i}(x_i)$ in terms of messages leaving x_j. This is called the **message updating rule**.

   One has that

$$m_{j \Rightarrow i}(x_i) = \sum_{x_j} \Delta(x_i, x_j) \prod_{b \in \partial j \setminus i} m_{b \Rightarrow j}(x_j) \tag{59.5}$$

This can be represented graphically by

(59.6)

2. Calculate $\widetilde{p}(x_i, x_j)$ for any edge $i-j$ in terms of messages entering x_i and leaving x_j

   One has that

$$\widetilde{p}(x_i, x_j) = \mathcal{N}(!x)\Delta(x_i, x_j) \prod_{a\in\partial i\setminus j} m_{a\Rightarrow i}(x_i) \prod_{b\in\partial j\setminus i} m_{b\Rightarrow j}(x_j) \tag{59.7}$$

   This can be represented graphically by

(59.8)

3. Calculate $\widetilde{p}(x_i)$ in terms of messages entering x_i

   One has that

$$\widetilde{p}(x_i) = \sum_{x_j} \widetilde{p}(x_i, x_j) \tag{59.9}$$

   This can be represented graphically by

(59.10)

Why it works?

Suppose we approximate the propagator $\Delta(x_i, x_j)$ by

$$\Delta(x_i, x_j) \approx m_{j\Rightarrow i}(x_i) m_{i\Rightarrow j}(x_j) \tag{59.11}$$

This can be represented graphically by

$$\textcircled{x_i} - \boxed{\Delta} - \textcircled{x_j} \quad \approx \quad \textcircled{x_i} - \boxed{m_{j\Rightarrow i}(x_i)} \quad \boxed{m_{i\Rightarrow j}(x_j)} - \textcircled{x_j} \tag{59.12}$$

Then the results 1,2,3 above can be justified as follows.

1.

$$\begin{aligned} m_{j\Rightarrow i}(x_i) &\approx \sum_{x_j} \underbrace{m_{j\Rightarrow i}(x_i) m_{i\Rightarrow j}(x_j)}_{\Delta(x_i,x_j)} \prod_{b\in\partial j\setminus i} m_{b\Rightarrow j}(x_j) && (59.13)\\ &= m_{j\Rightarrow i}(x_i) \sum_{x_j} \prod_{b\in\partial j} m_{b\Rightarrow j}(x_j) && (59.14)\\ &= \mathcal{N}(!x) m_{j\Rightarrow i}(x_i) && (59.15) \end{aligned}$$

2.

$$\begin{aligned} \widetilde{p}(x_i, x_j) &= \sum_{x\setminus x_i, x_j} \widetilde{p}(x) && (59.16)\\ &\approx \mathcal{N}(!x) \sum_{x\setminus x_i, x_j} \prod_{i-j} \underbrace{m_{j\Rightarrow i}(x_i) m_{i\Rightarrow j}(x_j)}_{\Delta(x_i,x_j)} && (59.17)\\ &\approx \mathcal{N}(!x) \Delta(x_i, x_j) \prod_{a\in\partial i\setminus j} m_{a\Rightarrow i}(x_i) \prod_{b\in\partial j\setminus i} m_{b\Rightarrow j}(x_j) && (59.18) \end{aligned}$$

3.

$$\begin{aligned} \widetilde{p}(x_i) &= \sum_{x_j} \widetilde{p}(x_i, x_j) && (59.19)\\ &\approx \mathcal{N}(!x) \sum_{x_j} \underbrace{m_{j\Rightarrow i}(x_i) m_{i\Rightarrow j}(x_j)}_{\Delta(x_i,x_j)} \prod_{a\in\partial i\setminus j} m_{a\Rightarrow i}(x_i) \prod_{b\in\partial j\setminus i} m_{b\Rightarrow j}(x_j) && (59.20)\\ &\approx \mathcal{N}(!x) \prod_{a\in\partial i} m_{a\Rightarrow i}(x_i) && (59.21) \end{aligned}$$

Henceforth, we will occasionally use the following more compressed notation for the factor graph of a 2MRF. Instead of using circles for variables and squares for functions, we will use arrows pointing from x_1 to x_2 and vice versa.

$$\textcircled{x_1} - \boxed{m_{2\Rightarrow 1}(x_1)} \quad \boxed{m_{1\Rightarrow 2}(x_2)} - \textcircled{x_2} = \quad x_1 \underset{m_{2\Rightarrow 1}(x_1)}{\overset{m_{1\Rightarrow 2}(x_2)}{\rightleftarrows}} x_2 \tag{59.22}$$

Figure 59.2: Example of compressed factor graph for a 2MRF.

Next, we give an example of how to use message passing to find the marginal $\widetilde{p}(x_i)$ for any node x_i. Refer to Fig59.2 for this example.

Using the message updating rule successively, we get

$$\widetilde{p}(x_1) = \mathcal{N}(!x) m_{2\Rightarrow 1}(x_1) \tag{59.23}$$

$$= \mathcal{N}(!x) \sum_{x_2} \Delta(x_1, x_2) m_{3\Rightarrow 2}(x_2) m_{4\Rightarrow 2}(x_2) \tag{59.24}$$

$$= \mathcal{N}(!x) \sum_{x_2} \Delta(x_1, x_2) \sum_{x_3} \Delta(x_2, x_3) \sum_{x_4} \Delta(x_2, x_4) \tag{59.25}$$

$$= \tag{59.26}$$

This algorithm is guaranteed to work only for trees. In practice, one starts by calculating the messages pointing up from the leaf nodes of the tree. Then one calculates the messages pointing up from the parents of the leaf nodes of the tree. And so forth until one calculates all the messages pointing up from the leaf nodes to the root node of the tree. Then one goes in the opposite direction, first calculating messages pointing down from the root node to its children, from the children of the root node to their children. And so forth. By the end, all upward and downward pointing messages have been calculated. From this, $\widetilde{p}(x_i)$ for all i can be calculated. $\widetilde{p}(x_i)$ is the product of all messages pointing into x_i. .

59.3 $-\ln Z_\theta$= Free Energy (FE)

Suppose $\theta, \epsilon(x) \in \mathbb{R}^{ni}$ and $x = (x_1, x_2, \ldots, x_{nx}) \in \mathbb{R}^{nx}$. Define the **partition function** Z_θ by

$$Z_\theta = \sum_x e^{-\theta^T \epsilon(x)} \tag{59.27}$$

and the probability distribution $P(x|h)$ by

$$\underbrace{P(x|\theta)}_{e^{LL_\theta(x)}} = \exp(-\theta^T \epsilon(x) - \ln Z_\theta) \tag{59.28}$$

$$= \frac{e^{-\theta^T \epsilon(x)}}{Z} \tag{59.29}$$

$\epsilon(x)$ is called a **sufficient statistic** for $\underline{x}$ because $P(x|\theta)$ is a functional (i.e., a function of a function) of $\epsilon(x)$. Note that by taking first and higher order derivatives of $\ln Z_\theta$ with respect to θ_i, we can calculate the statistics of $\epsilon_i(x)$.

$$-\partial_{\theta_i} \ln Z_\theta = \frac{1}{Z_\theta} \sum_x \epsilon_i(x) e^{-\theta^T \epsilon(x)} \tag{59.30}$$

$$= E_{\underline{x}|\theta}[\epsilon_i(\underline{x})] = \langle \epsilon_i \rangle \tag{59.31}$$

$$\partial_{\theta_j} \partial_{\theta_i} \ln Z_\theta = \partial_{\theta_j} \frac{1}{Z_\theta} \sum_x -\epsilon_i(x) e^{-\theta^T \epsilon(x)} \tag{59.32}$$

$$= \begin{cases} \frac{1}{Z_\theta} \sum_x \epsilon_j(x) \epsilon_i(x) e^{-\theta^T \epsilon(x)} \\ + \frac{-1}{Z_\theta^2} \left[\sum_x -\epsilon_j(x) e^{-\theta^T \epsilon(x)} \right] \left[\sum_x -\epsilon_i(x) e^{-\theta^T \epsilon(x)} \right] \end{cases} \tag{59.33}$$

$$= \langle \epsilon_j \epsilon_i \rangle - \langle \epsilon_j \rangle \langle \epsilon_i \rangle \tag{59.34}$$

$$= \langle \epsilon_j, \epsilon_i \rangle \tag{59.35}$$

Define the log likelihood $LL_\theta(x)$ by

$$LL_\theta(x) = -\theta^T \epsilon(x) - \ln Z_\theta \tag{59.36}$$

and the **entropy** S by [1]

$$S = -\sum_x P(x|\theta) LL_\theta(x) \tag{59.37}$$

$$= -\sum_x P(x|\theta) \ln P(x|\theta) \tag{59.38}$$

[1] In Thermodynamics, the entropy is denoted by the letter S. In Shannon Information Theory, and elsewhere in this book, it is denoted by the letter H.

Define the **internal energy** U by

$$U = \sum_x P(x|\theta)\epsilon_\theta(x) \tag{59.39}$$

$$-S = -\theta^T U - \ln Z_\theta \tag{59.40}$$

$$\partial_{U_i} S = \theta_i \tag{59.41}$$

S is concave. S and $-\ln Z_\theta = F/T$ are concave dual functions.[2].

Relationship to Thermodynamics

In Thermodynamics, U is the internal energy and S is the entropy of a system at **temperature** T. Define $\theta \in \mathbb{R}$ to be the inverse temperature

$$\theta = \frac{1}{T} \tag{59.42}$$

Define the **free energy** F by[a]

$$F = -T \ln Z_\theta \tag{59.43}$$

$$= -T \ln \sum_x e^{-\frac{\epsilon(x)}{T}} \tag{59.44}$$

Then

$$U - TS = F \tag{59.45}$$

So the free energy equals the internal energy minus the energy held in disordered form.

[a] In this chapter, we call $-\ln Z$ the free energy too.

59.4 $-\ln Z_{\theta^*}$= Minimum FE

Suppose we consider $P(x|\theta)$ for two different parameter θ and $\widetilde{\theta}$. Define

$$p(x) = P(x|\theta),\ \widetilde{p}(x) = P(x|\widetilde{\theta}) \tag{59.46}$$

The Kullback-Leibler divergence is always non-negative so:

[2] Concave dual functions are discussed in Chapter 109

$$
\begin{aligned}
0 &\leq D_{KL}(\widetilde{p}(x)) \parallel p(x)) && (59.47)\\
&= \sum_x \widetilde{p}(x) \ln \frac{\widetilde{p}(x)}{p(x)} && (59.48)\\
&= -\widetilde{S} - \sum_x \widetilde{p}(x) \left[-(\theta)^T \epsilon(x) - \ln Z_\theta \right] && (59.49)\\
&= -\widetilde{S} + (\theta)^T \widetilde{U} + \ln Z_\theta \quad (\widetilde{S}, \widetilde{U} \text{ correspond to parameter } \widetilde{\theta}) && (59.50)
\end{aligned}
$$

$$
-\ln Z_\theta \leq -\widetilde{S} + (\theta)^T \widetilde{U} \tag{59.51}
$$

Let

$$
\theta^* = \underset{\theta}{\operatorname{argmin}} \left[-\widetilde{S} + (\theta)^T \widetilde{U} \right] \tag{59.52}
$$

Henceforth, we will refer to $-\ln Z_\theta$ as the **FE (Free Energy)** and to $-\ln Z_{\theta^*}$ as the **minimum FE**.

Relationship to convex/concave dual functions[3]

$$
\widetilde{S} = \min_\theta \left[(\theta)^T \widetilde{U} + \ln Z_\theta \right] \tag{59.53}
$$

$$
-\ln Z_\theta = \min_{\widetilde{U}} \left[(\theta)^T \widetilde{U} - \widetilde{S} \right] \tag{59.54}
$$

$\widetilde{S}$ and $-\ln Z_\theta$ are concave dual functions.

$$
-\ln Z_{\theta^*} = (\theta^*)^T \widetilde{U} - \widetilde{S} \tag{59.55}
$$

59.5 $-\ln Z_\theta^{tree}$=Tree FE (a.k.a. Bethe FE)

$\widetilde{p}()x)$ is said to satisfy **Mean Field Approximation (MFA) or independent variables approximation (IVA)** if its variables x_i are independently distributed:

$$
\widetilde{p}^{ind}(x) = \prod_k \widetilde{p}(x_k) \tag{59.56}
$$

In the MFA, the entropy is

$$
\begin{aligned}
\widetilde{S}^{ind} &= -\sum_x \widetilde{p}(x) \ln \prod_i \widetilde{p}(x_i) && (59.57)\\
&= -\sum_{x_i} \widetilde{p}(x_i) \ln \widetilde{p}(x_i) && (59.58)\\
&= \sum_i \widetilde{H}(\underline{x}_i) && (59.59)
\end{aligned}
$$

[3]Convex/concave dual functions are discussed in Chapter 109

A slightly more complicated case than the MFA is when $\widetilde{p}(x)$ is defined on a tree graph G^{tree} with edges $i-j$. In such a case, it will take just a few examples of trees G^{tree} to convince the reader that in general, for any tree G^{tree}, $\widetilde{p}(x)$ must have the following form:

$$\widetilde{p}^{tree}(x) = \widetilde{p}^{ind}(x) \prod_{i-j} \frac{\widetilde{p}(x_i, x_j)}{\widetilde{p}(x_i)\widetilde{p}(x_j)} \tag{59.60}$$

Hence,

$$\begin{aligned}
\widetilde{S}^{tree} &= -\sum_i \sum_{x_i} \widetilde{p}^{tree}(x_i) \ln \widetilde{p}^{tree}(x_i) && (59.61)\\
&= \sum_i \widetilde{H}(\underline{x}_i) - \sum_{i-j} \sum_{x_i, x_j} \widetilde{p}(x_i, x_j) \ln \frac{\widetilde{p}(x_i, x_j)}{\widetilde{p}(x_i)\widetilde{p}(xj)} && (59.62)\\
&= \sum_i \widetilde{H}(\underline{x}_i) - \sum_{i-j} \widetilde{H}(\underline{x}_i : \underline{x}_j) && (59.63)
\end{aligned}$$

Note that $\widetilde{S}^{tree}$ can be written in terms of the joint entropy $\widetilde{H}(\underline{x}_i, \underline{x}_j)$ instead of the mutual entropy $\widetilde{H}(\underline{x}_i : \underline{x}_j)$.

$$\widetilde{S}^{tree} = -\sum_i (d_i - 1)\widetilde{H}(\underline{x}_i) + \sum_{i-j} \widetilde{H}(\underline{x}_i, \underline{x}_j) \tag{59.64}$$

where d_i is the number of neighbors of node i.

The following approximation is often called the **Bethe approximation**

$$-\ln Z_{\theta^*} \approx -\ln Z_{\theta^*}^{tree} \tag{59.65}$$

59.6 $-\ln Z_{\theta^*}^{tree}$= Tree Minimum FE, and message passing

In this section, we will evaluate $-\ln Z_{\theta^*}^{tree}$ exactly using a message passing ansatz (an ansatz is an initial guess).

If we replace $\widetilde{S}$ by $\widetilde{S}^{tree}$ in Eq.(59.55), we get

$$-\ln Z_{\theta^*}^{tree} = \min_{\widetilde{U}} \left[(\theta)^T \widetilde{U} - \widetilde{S}^{tree} \right] \tag{59.66}$$

But note that

$$(\theta)^T \widetilde{U} = \sum_i \theta_i \sum_x \widetilde{p}(x) \epsilon_i(x) \tag{59.67}$$

$$= \sum_x \widetilde{p}(x) \underbrace{\sum_i \theta_i \epsilon_i(x)}_{\text{call } \Theta(x)} \tag{59.68}$$

so Eq.(59.66) becomes

$$-\ln Z_{\theta^*}^{tree} = \min_{\widetilde{p}} \left[\sum_x \widetilde{p}(x)\Theta(x) - \sum_i \widetilde{H}(\underline{x}_i) + \sum_{i-j} \widetilde{H}(\underline{x}_i : \underline{x}_j) \right] \tag{59.69}$$

subject to $\sum_x \widetilde{p}(x) = 1$ and $\widetilde{p}(x) \geq 0$ for all x.

Claim 88 $-\ln Z_{\theta^*}^{tree}$ *is achieved if*

$$\widetilde{p}(x) = \mathcal{N}(!x) e^{-\Theta(x)} \tag{59.70}$$

$$\Theta(x) = \sum_i \Theta(x_i) + \sum_{i-j} \Theta(x_i, x_j) \tag{59.71}$$

(This form for $\widetilde{p}(x)$ and $\Theta(x)$ agrees with Eq.(59.60))

$$m_{t \Rightarrow s}(x_s) = e^{\lambda_{t \Rightarrow s}(x_s)} \tag{59.72}$$

$$\widetilde{p}(x_i) = \mathcal{N}(!x) e^{-\Theta(x_i)} \prod_{a \in \partial i} m_{a \Rightarrow i}(x_i) \tag{59.73}$$

$$\widetilde{p}(x_i, x_j) = \mathcal{N}(!x) e^{-\Theta(x_i, x_j) - \Theta(x_i) - \Theta(x_j)} \left[\prod_{a \in \partial i \backslash j} m_{a \Rightarrow i}(x_i) \right] \left[\prod_{b \in \partial j \backslash i} m_{b \Rightarrow j}(x_j) \right] \tag{59.74}$$

proof:

We want to minimize the following Lagrangian with respect to variations $\delta \widetilde{p}(x_i)$ of $\widetilde{p}(x)$, for each i.

$$\mathcal{L} = \left\{ \begin{array}{l} \sum_x \widetilde{p}(x)\Theta(x) \\ + \sum_i (1 - d_i) \sum_{x_i} \widetilde{p}(x_i) \ln \widetilde{p}(x_i) \\ + \sum_{i-j} \sum_{x_i, x_j} \widetilde{p}(x_i, x_j) \ln \widetilde{p}(x_i, x_j) \\ + \lambda \left[\sum_x \widetilde{p}(x) - 1 \right] \end{array} \right\} \tag{59.75}$$

The term proportional to the Lagrange multiplier λ enforces the constraint $\sum_x \widetilde{p}(x) = 1$. Note that in general,

$$\delta\widetilde{p}(x) = \sum_i \delta\widetilde{p}(x_i) \tag{59.76}$$

so if we only vary $\widetilde{p}(x_i)$,

$$\begin{aligned} \delta F(\widetilde{p}(x)) &= \frac{\partial F(\widetilde{p}(x))}{\partial \widetilde{p}(x)} \delta\widetilde{p}(x) && (59.77) \\ &= \frac{\partial F(\widetilde{p}(x))}{\partial \widetilde{p}(x)} \delta\widetilde{p}(x_i) && (59.78) \end{aligned}$$

for any well behaved function $F : \mathbb{R} \to \mathbb{R}$. Hence

$$\delta\mathcal{L} = \sum_x \delta\widetilde{p}(x_i) \left\{ \begin{array}{l} \Theta(x) \\ + \sum_i (1 - d_i) \left[1 + \ln \widetilde{p}(x_i)\right] \\ + \sum_{i-j} \left[1 + \ln \widetilde{p}(x_i, x_j)\right] \\ +\lambda \end{array} \right\} \tag{59.79}$$

for any variation $\delta\widetilde{p}(x_i)$. Setting the coefficient of $\delta\widetilde{p}(x_i)$ to zero now yields

$$0 = \left\{ \begin{array}{l} \Theta(x) \\ + \sum_i \left[1 + \ln \widetilde{p}(x_i)\right] \\ + \sum_{i-j} \left[1 + \ln \frac{\widetilde{p}(x_i, x_j)}{\widetilde{p}(x_i)\widetilde{p}(x_j)}\right] \\ +\lambda \end{array} \right\} \tag{59.80}$$

for each i. If we now substitute the equations that are hypotheses to this claim, we get

$$0 = \left\{ \begin{array}{l} \Theta(x) \\ - \sum_i \Theta(x_i) \\ - \sum_{i-j} \Theta(x_i, x_j) \end{array} \right\} \tag{59.81}$$

$$\begin{aligned} 0 &= \left\{ \begin{array}{l} + \sum_i \ln \prod_{a \in \partial i} m_{a \Rightarrow i}(x_i) \\ + \sum_{i-j} \ln \frac{\left[\prod_{a \in \partial i \setminus j} m_{a \Rightarrow i}(x_i)\right] \left[\prod_{b \in \partial j \setminus i} m_{b \Rightarrow j}(x_j)\right]}{\prod_{a \in \partial i} m_{a \Rightarrow i}(x_i) \prod_{b \in \partial j} m_{b \Rightarrow j}(x_j)} \\ +\lambda \end{array} \right\} && (59.82) \\ &= \left\{ \begin{array}{l} \sum_i \sum_{a \in \partial i} \lambda_{a \Rightarrow i}(x_i) \\ - \sum_{i-j} \left[\lambda_{j \Rightarrow i}(x_i) + \lambda_{i \Rightarrow j}(x_j)\right] \\ +\lambda \end{array} \right\} && (59.83) \\ &= \lambda && (59.84) \end{aligned}$$

So the hypotheses to this claim indeed do satisfy $\delta\mathcal{L} = 0$ for all variations $\delta\widetilde{p}(x_i)$, for each i, with Lagrange multiplier $\lambda = 0$.
QED

Chapter 60

Message Passing, Pearl's theory

This chapter is mostly based on chapter 4 of Ref.[60] by Pearl. Refs.[109], and [51] were also helpful in writing this chapter.

In his book Ref.[60], Pearl explains two types of Message Passing (i.e., distributed computing in a bnet). In Chapter 4, he discusses one type of MP which he calls Belief Propagation (BP) or Belief Updating. In Chapter 5, he introduces a second type of MP which is he calls Belief Revision, but which I prefer to call Explanation Optimization (EO). This chapter will be devoted to BP only.

BP was first proposed for bnets in 1982 Ref.[59] by Judea Pearl to simplify the exact evaluation of the probability of one node conditioned on other nodes of a bnet (exact inference). It gives exact results for trees and polytrees (i.e., bnets with a single connected component and no loops). For bnets with loops, it gives approximate results (loopy belief propagation), and it has been generalized to the junction tree algorithm (see Chapter 46) which can do exact inference for general bnets with loops. The basic idea behind the junction tree algorithm is to eliminate loops by clustering them into single nodes.

60.1 Distributed Soldier Counting

Consider a group of soldiers marching single file. Fig.60.1 shows several methods by which a member of the group can obtain a count of the soldiers without breaking the line to do global operations. This can be done in a distributed fashion, with every soldier doing only local operations (i.e., each soldier can only send messages to either the soldier in front or the one in back). Such distributed soldier counting is a rudimentary type of BP. In the next section, we will generalize this BP for soldiers to BP for a Markov chain.

Figure 60.1: Distributed soldier counting (This example comes from Chapter 4 of Ref.[60]). Green dots indicate the beginning and red dots the end of a count. Only first soldier can calculate total count in (a). Only third soldier can calculate total count in (b,c). All soldiers can calculate the total count in (d,e). One starting point in (a,b,e). Two ends as starting points in (c,d).

60.2 Spring Systems

Figure 60.2: Spring system. Point masses connected by springs.

Figure 60.3: Diagram (a) portrays two events interacting via feedback. Diagram (b) represents the same thing as diagram (a) after "unrolling" it.

See Ref.[183] for an introduction to spring systems. Ideal springs between the point mass nodes would not be sufficient. One would have to add damping to the springs so as to reach an equilibrium. Time dependent forces (loads) pointing into or out of the page, applied to the point masses, would generate signals that would propagate like BP messages.

The messages in Fig.60.3 (a) appear to travel both forward and backwards in time. But in reality, messages cannot travel backward in time. Fig.60.3 (a) is just a convenient representation of Fig.60.3 (b), which "unrolls" it. Both (a) and (b) are representations of feedback. Messages that appear to travel back in time, (but in reality don't, this are just a convenient representation of feedback) is a recurring theme in Physics. For example, in Quantum Electrodynamics, it appears that positrons can

travel backwards in time. In reality, they are portraying the feedback mechanism whereby two charged particles apply a force on each other.

60.3 BP for Markov Chains

Message Arrows Notation (See Fig.60.4):

$$\begin{tikzcd}
\underline{b} \arrow[rr, leftarrow] \arrow[rr, bend right, dashed, leftarrow, "\pi_{\underline{b}\Leftarrow\underline{a}}"] \arrow[rr, bend left, dotted, "\lambda_{\underline{b}\Rightarrow\underline{a}}"'] & & \underline{a}
\end{tikzcd}$$

Figure 60.4: One arrow ($\underline{b} \leftarrow \underline{a}$) of a bnet, adorned with message passing arrows.

The π messages (probability functions) travel downstream (i.e., they carry info in the direction of the graph arrows, towards the future) and are indicated by a dashed arrow. The λ messages (likelihood functions) travel upstream (i.e., they carry info opposite to direction of the graph arrows, towards the past) and are indicated by a dotted arrow. $\underline{\epsilon}^+$ stands for future evidence and $\underline{\epsilon}^-$ for past evidence.

$\underline{a} \Rightarrow \underline{b}$ and $\underline{b} \Leftarrow \underline{a}$ will mean the same thing, namely either a (dot or dashed) message arrow directed from $\underline{a}$ to $\underline{b}$, but not a (solid) bnet arrow.

$\pi_{\underline{b}\Leftarrow\underline{a}}(a) = P(a|\epsilon^-)$

$\lambda_{\underline{b}\Rightarrow\underline{a}}(a) = P(\epsilon^+|a)$

Note that the argument of $\pi_{\underline{b}\Leftarrow\underline{a}}(a)$ is the source node of the double arrow and the argument of $\lambda_{\underline{b}\Rightarrow\underline{a}}(a)$ is the destination node of the double arrow. It's also true that in both $\pi_{\underline{b}\Leftarrow\underline{a}}(a)$ and $\pi_{\underline{b}\Leftarrow\underline{a}}(a)$, the argument a in parenthesis appears next to $\underline{a}$ in the subscript.

Note that we indicate messages that travel "downstream" (resp., "upstream"), by arrows with dashed (resp., dotted) lines as shafts. Mnemonic: think of the shaft as a velocity vector field for the message. You travel faster when you swim downstream as opposed to upstream.

$$\begin{tikzcd}
\underline{\epsilon}^+ \arrow[r, dotted, bend right, "\lambda_{\underline{\epsilon}^+\Rightarrow\underline{b}}"'] & \underline{b} \arrow[l, leftarrow, bend right, dashed, "\pi_{\underline{\epsilon}^+\Leftarrow\underline{b}}"'] \arrow[r, dotted, bend right, "\lambda_{\underline{b}\Rightarrow\underline{x}}"'] & \underline{x} \arrow[l, leftarrow, bend right, dashed, "\pi_{\underline{b}\Leftarrow\underline{x}}"'] \arrow[r, dotted, bend right, "\lambda_{\underline{x}\Rightarrow\underline{a}}"'] & \underline{a} \arrow[l, leftarrow, bend right, dashed, "\pi_{\underline{x}\Leftarrow\underline{a}}"'] \arrow[r, dotted, bend right, "\lambda_{\underline{a}\Rightarrow\underline{\epsilon}^-}"'] & \underline{\epsilon}^- \arrow[l, leftarrow, bend right, dashed, "\pi_{\underline{a}\Leftarrow\underline{\epsilon}^-}"']
\end{tikzcd}$$

Figure 60.5: 5 node Markov chain

Consider the 5 node Markov chain $\underline{\epsilon}^+ \leftarrow \underline{b} \leftarrow \underline{x} \leftarrow \underline{a} \leftarrow \underline{\epsilon}^-$ shown in Fig.60.5. Define

$$\begin{array}{c|c|c|c} \pi_{\underline{\epsilon}^+ \Leftarrow \underline{b}}(b) = P(b|\epsilon^-) & \pi_{\underline{b} \Leftarrow \underline{x}}(x) = P(x|\epsilon^-) & \pi_{\underline{x} \Leftarrow \underline{a}}(a) = P(a|\epsilon^-) & \pi_{\underline{a} \Leftarrow \underline{\epsilon}^-}(\epsilon^-) = P(\epsilon^-|\epsilon^-) \\ \lambda_{\underline{\epsilon}^+ \Rightarrow \underline{b}}(b) = P(\epsilon^+|b) & \lambda_{\underline{b} \Rightarrow \underline{x}}(x) = P(\epsilon^+|x) & \lambda_{\underline{x} \Rightarrow \underline{a}}(a) = P(\epsilon^+|a) & \lambda_{\underline{a} \Rightarrow \underline{\epsilon}^-}(\epsilon^-) = P(\epsilon^+|\epsilon^-) \end{array} \tag{60.1}$$

Furthermore, define the Belief BEL in x to be

$$BEL_{\underline{x}}(x) = P(x|\epsilon) \ , \tag{60.2}$$

where

$$\underline{\epsilon} = \underline{\epsilon}^+ \cup \underline{\epsilon}^- \ . \tag{60.3}$$

It follows that[1]

$$\begin{aligned} BEL_{\underline{x}}(x) &= P(x|\epsilon^+, \epsilon^-) = && (60.4) \\ &= \mathcal{N}(!x) P(\epsilon^+, x, \epsilon^-) && (60.5) \\ &= \mathcal{N}(!x) P(\epsilon^+|x, \epsilon^-) P(x|\epsilon^-) && (60.6) \\ &= \mathcal{N}(!x) P(\epsilon^+|x) P(x|\epsilon^-) && (60.7) \\ &= \mathcal{N}(!x) \lambda_{\underline{b} \Rightarrow \underline{x}}(x) \pi_{\underline{b} \Leftarrow \underline{x}}(x) \ . && (60.8) \end{aligned}$$

Note that the π messages and λ messages propagate independently of each other, via the TPM $P(x|a)$.

$$\underbrace{P(x|\epsilon^-)}_{\pi_{\underline{b} \Leftarrow \underline{x}}(x)} = \sum_a P(x|a) \underbrace{P(a|\epsilon^-)}_{\pi_{\underline{x} \Leftarrow \underline{a}}(a)} \tag{60.9a}$$

$$\underbrace{P(\epsilon^+|a)}_{\lambda_{\underline{x} \Rightarrow \underline{a}}(a)} = \sum_x P(x|a) \underbrace{P(\epsilon^+|x)}_{\lambda_{\underline{b} \Rightarrow \underline{x}}(x)} \tag{60.9b}$$

Eqs.(60.9) suggest that we define a message bnet for the π and λ messages. Such a message bnet, shown in Fig.60.6, is complementary to the original bnet. We will call it the **BP 2-track bnet** for the bnet Fig.60.5, because it has two "tracks", one for π messages and another for λ ones. The TPMs, printed in blue, for the red cell of bnet Fig.60.6, are as follows:

$$P(\pi_{\underline{b} \Leftarrow \underline{x}} | \pi_{\underline{x} \Leftarrow \underline{a}}) = \prod_x \mathbb{1} \left(\pi_{\underline{b} \Leftarrow \underline{x}}(x) = \sum_a P(x|a) \pi_{\underline{x} \Leftarrow \underline{a}}(a) \right) \tag{60.10}$$

$$P(\lambda_{\underline{x} \Rightarrow \underline{a}} | \lambda_{\underline{b} \Rightarrow \underline{x}}) = \prod_a \mathbb{1} \left(\lambda_{\underline{x} \Rightarrow \underline{a}}(a) = \sum_x P(x|a) \lambda_{\underline{b} \Rightarrow \underline{x}}(x) \right) \tag{60.11}$$

[1] As usual in this book, $\mathcal{N}(!x)$ means a constant that is independent of x.

Figure 60.6: BP 2-track bnet for the bnet Fig.60.5. We've marked in red the cell of the bnet pertaining to Eqs.(60.9)

$$P(B_{\underline{x}}|\pi_{\underline{b}\Leftarrow\underline{x}}, \lambda_{\underline{b}\Rightarrow\underline{x}}) = \prod_x \mathbb{1}\left(B_{\underline{x}}(x) = BEL_{\underline{x}}(x)\right) \tag{60.12}$$

Let us represent the Markov chain of Fig.60.5 by $\underline{x}_{nx-1} \leftarrow \ldots, \underline{x}_2 \leftarrow \underline{x}_1 \leftarrow \underline{x}_0$ where $nx = 5$. For any node $\underline{x}_i$ with parent $\underline{px}_i = \underline{x}_{i-1}$ and child $\underline{cx}_i = \underline{x}_{i+1}$, define the **memory matrix** $\mathcal{M}_{\underline{x}_i}$ for node $\underline{x}_i$ as

$$\mathcal{M}_{\underline{x}_i} = [\mathcal{M}^+_{\underline{x}_i}, \mathcal{M}^-_{\underline{x}_i}] \,, \tag{60.13}$$

where $+$ =future, $-$ =past, and

$$\mathcal{M}^+_{\underline{x}_i} = \begin{bmatrix} \pi_{\underline{cx}_i \Leftarrow \underline{x}_i}(\cdot) \\ \lambda_{\underline{cx}_i \Rightarrow \underline{x}_i}(\cdot) \end{bmatrix} \,, \quad \mathcal{M}^-_{\underline{x}_i} = \begin{bmatrix} \pi_{\underline{x}_i \Leftarrow \underline{px}_i}(\cdot) \\ \lambda_{\underline{x}_i \Rightarrow \underline{px}_i}(\cdot) \end{bmatrix} \,. \tag{60.14}$$

Note that

$$\mathcal{M}^-_{\underline{x}_i} = \mathcal{M}^+_{\underline{px}_i} \tag{60.15}$$

for all nodes $\underline{x}_i$. We will refer to Eqs.(60.15) as the **memory overlap conditions**.

We will also use a permuted version of the memory matrix

$$\mathcal{M}'_{\underline{x}_i} = [\mathcal{M}^{OUT}_{\underline{x}_i}, \mathcal{M}^{IN}_{\underline{x}_i}] \,, \tag{60.16}$$

where

$$\mathcal{M}^{OUT}_{\underline{x}_i} = \begin{bmatrix} \pi_{\underline{cx}_i \Leftarrow \underline{x}_i}(\cdot) \\ \lambda_{\underline{x}_i \Rightarrow \underline{px}_i}(\cdot) \end{bmatrix} \,, \quad \mathcal{M}^{IN}_{\underline{x}_i} = \begin{bmatrix} \pi_{\underline{x}_i \Leftarrow \underline{px}_i}(\cdot) \\ \lambda_{\underline{cx}_i \Rightarrow \underline{x}_i}(\cdot) \end{bmatrix} \,. \tag{60.17}$$

Unfortunately, 2-track bnets cannot be generalized in any obvious way from Markov chains to more complicated DAGs. An alternative to 2-track bnets that still carries message info in its nodes, are memory bnets. An BP **memory bnet** is a bnet which takes each node of an original bnet and adds a local memory to it. More

$$\underline{\mathcal{M}}_{\underline{\epsilon}^+} \longleftarrow \underline{\mathcal{M}}_{\underline{b}} \longleftarrow \underline{\mathcal{M}}_{\underline{x}} \longleftarrow \underline{\mathcal{M}}_{\underline{a}} \longleftarrow \mathcal{M}_{\underline{\epsilon}^-}$$

Figure 60.7: BP Memory Bnet for the bnet Fig.60.5.

specifically, it keeps that DAG but replaces each node $\underline{x}_i$ by a memory $\underline{\mathcal{M}}_{\underline{x}_i}$. Fig.60.7 shows the memory bnet for the bnet Fig.60.5. The TPM, printed in blue, for the node $\underline{\mathcal{M}}_{\underline{x}}$ of the memory bnet Fig.60.7, is as follows

$$P(\mathcal{M}_{\underline{x}_i}|\mathcal{M}_{\underline{n}\in nb(\underline{x}_i)}) = AB \ , \tag{60.18}$$

where

$$A = \mathbb{1}(\mathcal{M}^-_{\underline{x}_i} = \mathcal{M}^+_{\underline{px}_i}) \ , \tag{60.19}$$

and

$$B = \mathbb{1}(\mathcal{M}^{OUT}_{\underline{x}_i} = \mathcal{C}(\mathcal{M}^{IN}_{\underline{x}_i})) \ . \tag{60.20}$$

The function $\mathcal{C}$, which we will call the **BP local computation**, maps $\mathcal{M}^{IN}_{\underline{x}_i}$ into $\mathcal{M}^{OUT}_{\underline{x}_i}$. More explicitly, $\mathcal{C}$ is defined so that

$$B = \underbrace{P(\pi_{\underline{b}\Leftarrow\underline{x}}|\pi_{\underline{x}\Leftarrow\underline{a}})}_{B_\pi}\underbrace{P(\lambda_{\underline{x}\Rightarrow\underline{a}}|\lambda_{\underline{b}\Rightarrow\underline{x}})}_{B_\lambda} \ , \tag{60.21}$$

where B_π and B_λ are given by Eqs.(60.10) and (60.11), respectively.

The BP memory bnet Fig. 60.7 is a deterministic bnet. A deterministic bnet is basically just a coupled system of equations (CSE) for some unknowns x_i. A CSE per se does not include with it a method for solving for the x_i. Such methods are not unique. For example, for the distributed soldier counting problem, the various methods that we described for counting soldiers are just different methods for solving the same CSE. One can describe a method for solving a CSE using a dynamical bnet.[2] To solve the CSE represented by the memory bnet Fig.60.7, we will use the dynamical bnet Fig.60.8. Henceforth, we will refer to Fig.60.8 as an **BP dynamical bnet** for Fig.60.7.

Next, we will explain the meaning of Fig.60.8. Fig.60.8 is a step by step recipe (i.e., algorithm) for solving a CSE, where the unknowns are memory matrices. Each step encoded in Fig.60.8 corresponds to a specific message sending event, where the messages are sent along the edges of the Markov chain Fig.60.5. These message sending events are portrayed in chronological order in Fig.60.9. In that figure, π messages are indicated by dashed red arrows, and λ messages by dotted red arrows.

[2]The term dynamical bnet was used in Chapter 26 to mean a time homogeneous Markov chain, but here we are stretching its meaning to include Markov chains that aren't time homogeneous.

Figure 60.8: BP dynamical bnet for the bnet Fig.60.7.

Figure 60.9: Steps encoded in the bnet Fig.60.8. Note the similarity of this figure to Fig.60.1 (d) for soldier counting.

These steps, or message sending events, lead to an updating of the memory matrices that we are solving for. Each step propagates information between the memory nodes. In the usual Pearl BP algo, the evidence nodes initiate the BP chain of message passing events. These events continue until the memory matrices reach an equilibrium and the CSE is solved.

To use bnet Fig.60.8, we need to specify the **initial conditions** (i.e., the value of $\mathcal{M}^{(0)}_{\underline{x}_i}$ for all i). For that, one can use

$$\pi^{(0)}_{\underline{px}_0 \Leftarrow \underline{x}_0} = P(x_0) \ , \tag{60.22}$$

$$\lambda^{(0)}_{\underline{cx}_i \Rightarrow \underline{x}_{nx-1}}(x_{nx-1}) = \delta(x_{nx-1}, x'_{nx-1}) \ . \tag{60.23}$$

All other $\mathcal{M}^{(0)}_{\underline{x}_i}$ entries for all i can be set to 1.

The TPMs, printed in blue, for bnet Fig.60.8, are as follows.

$$P(\mathcal{M}^{(t)}_{\underline{x}_i} | \mathcal{M}^{(t-1)}_{\underline{n} \in nb(\underline{x}_i)}, \mathcal{M}^{(t-1)}_{\underline{x}_i}) = AB \ , \tag{60.24}$$

where

$$A = \begin{cases} \mathbb{1}(\mathcal{M}^{(t)-}_{\underline{x}_i} = \mathcal{M}^{(t-1)+}_{\underline{px}_i}) & \text{if input from } \underline{px}_i \\ \\ \mathbb{1}(\mathcal{M}^{(t)+}_{\underline{x}_i} = \mathcal{M}^{(t-1)-}_{\underline{cx}_i}) & \text{if input from } \underline{cx}_i \end{cases} \ , \tag{60.25}$$

and

$$B = \mathbb{1}(\mathcal{M}^{(t)OUT}_{\underline{x}_i} = \mathcal{C}(\mathcal{M}^{(t)IN}_{\underline{x}_i})) \ . \tag{60.26}$$

The function $\mathcal{C}$, which we will call the **BP local computation**, maps $\mathcal{M}^{(t)IN}_{\underline{x}_i}$ into $\mathcal{M}^{(t)OUT}_{\underline{x}_i}$. More explicitly, $\mathcal{C}$ is defined so that

$$B = B_\pi B_\lambda \tag{60.27}$$

where

$$B_\pi = \prod_x \mathbb{1} \left(\underbrace{\pi^{(t)}_{\underline{b} \Leftarrow \underline{x}}(x)}_{OUT} = \sum_a P(x|a) \underbrace{\pi^{(t)}_{\underline{x} \Leftarrow \underline{a}}(a)}_{IN} \right) \tag{60.28}$$

and

$$B_\lambda = \prod_a \mathbb{1} \left(\underbrace{\lambda^{(t)}_{\underline{x} \Rightarrow \underline{a}}(a)}_{OUT} = \sum_x P(x|a) \underbrace{\lambda^{(t)}_{\underline{b} \Rightarrow \underline{x}}(x)}_{IN} \right) \ . \tag{60.29}$$

The basic idea behind Eq.(60.26), which we will call the **memory updating equation**, is simple: the memory overlap conditions translate the information from time $t-1$ to t, and then the local computation translates IN to OUT at fixed time t.

60.4 BP Algorithm for Polytrees

Consider Fig.60.10, which illustrates a bnet node $\underline{x}$ receiving messages from and sending them to its neighbors. The π and λ messages are labeled in accordance to the notational conventions previously stated in Section 60.3. The π messages (probability functions) travel downstream (i.e., they carry info in the direction of the graph arrows, towards the future) and are indicated by a dashed arrow or by a left double arrow $\Leftarrow$. The λ messages (likelihood functions) travel upstream (i.e., they carry info opposite to direction of the graph arrows, towards the past) and are indicated by a dotted arrow or by a right double arrow $\Rightarrow$.

Let
$pa(\underline{x})$ = parents of node $\underline{x}$
$ch(\underline{x})$ = children of node $\underline{x}$
$nb(\underline{x}) = pa(\underline{x}) \cup ch(\underline{x})$ = neighbors of node $\underline{x}$

Figure 60.10: Node $\underline{x}$ receiving messages from and sending them to its neighbors. (neighbors= parents and children).

We define a **memory matrix** $\mathcal{M}_{\underline{x}}$ for node $\underline{x}$ as

$$\mathcal{M}_{\underline{x}} = [\mathcal{M}^+_{\underline{x}}, \mathcal{M}^-_{\underline{x}}] \ , \tag{60.30}$$

where $+$ =future, $-$ =past, and

$$\mathcal{M}^+_{\underline{x}} = \left[\ \pi_{\underline{b}\Leftarrow\underline{x}}(\cdot) \quad \lambda_{\underline{b}\Rightarrow\underline{x}}(\cdot) \ \right]_{\underline{b}\in ch(\underline{x})} = [\mathcal{M}^+_{\underline{b},\underline{x}}]_{\underline{b}\in ch(\underline{x})} \ , \tag{60.31}$$

$$\mathcal{M}^-_{\underline{x}} = \left[\begin{array}{c} \pi_{\underline{x}\Leftarrow\underline{a}}(\cdot) \\ \lambda_{\underline{x}\Rightarrow\underline{a}}(\cdot) \end{array} \right]_{\underline{a}\in pa(\underline{x})} = [\mathcal{M}^-_{\underline{x},\underline{a}}]_{\underline{a}\in pa(\underline{x})} \ . \tag{60.32}$$

Note that

$$\mathcal{M}^{-}_{\underline{x},\underline{a}} = \mathcal{M}^{+}_{\underline{a},\underline{x}} \tag{60.33}$$

for every arrow $\underline{x} \leftarrow \underline{a}$. We will refer to Eqs.(60.33) as the **memory overlap conditions**.

We will also use a permuted version of the memory matrix

$$\mathcal{M}'_{\underline{x}} = [\mathcal{M}^{OUT}_{\underline{x}}, \mathcal{M}^{IN}_{\underline{x}}] \ , \tag{60.34}$$

where

$$\mathcal{M}^{OUT}_{\underline{x}} = \begin{pmatrix} [\pi_{\underline{b}\Leftarrow\underline{x}}(\cdot)]_{\underline{b}\in ch(\underline{x})} \\ [\lambda_{\underline{x}\Rightarrow\underline{a}}(\cdot)]_{\underline{a}\in pa(\underline{x})} \ , \end{pmatrix} = [\mathcal{M}^{OUT}_{\underline{x},\underline{n}}]_{\underline{n}\in nb(\underline{x})} \ , \tag{60.35}$$

$$\mathcal{M}^{IN}_{\underline{x}} = \begin{pmatrix} [\pi_{\underline{x}\Leftarrow\underline{a}}(\cdot)]_{\underline{a}\in pa(\underline{x})} \\ [\lambda_{\underline{b}\Rightarrow\underline{x}}(\cdot)]_{\underline{b}\in ch(\underline{x})} \end{pmatrix} = [\mathcal{M}^{IN}_{\underline{x},\underline{n}}]_{\underline{n}\in nb(\underline{x})} \ . \tag{60.36}$$

For times $t = 0, 1, \ldots, T-1$, we calculate $\mathcal{M}^{(t)}_{\underline{x}}$ in two steps: first we calculate $\mathcal{M}^{(t)IN}_{\underline{x}}$ from earlier memories at time $t-1$, then we calculate $\mathcal{M}^{(t)OUT}_{\underline{x}}$:

Figure 60.11: Subgraph of a bnet showing two cases (RULE 1 and RULE 2) of message info flow. The yellow node is a gossip monger. It receives messages from all the green nodes, and then it relays a joint message to the red node. Union of green nodes and the red node = full neighborhood of yellow node. There are two possible cases: the red node is either a parent or a child of the yellow one. As usual, we use arrows with dashed (resp., dotted) shafts for downstream (resp., upstream) messages. Blue boxes indicate Markov chain case.

An **evidence node** is a node whose TPM is a delta function set to a particular state of the node. We will assume, without loss of generality, that all evidence nodes are leaf nodes. If that is not the case, any evidence node $\underline{e}$ that is not a leaf node,

can be given a new companion leaf node $\underline{l}$ connected to $\underline{e}$ by an arrow $\underline{l} \leftarrow \underline{e}$, and such that $\underline{l}$ has a delta function TPM.

1. **Calculating $\mathcal{M}_{\underline{x}}^{(t)IN}$ from signals received from $\underline{n} \in nb(\underline{x})$, sent at earlier time $t-1$:**

   Set

$$\mathcal{M}_{\underline{x},\underline{a}}^{(t)-}|_{\pi} = \mathcal{M}_{\underline{a},\underline{x}}^{(t-1)+}|_{\pi} \ , \tag{60.37}$$

   for all $\underline{a} \in pa(\underline{x})$, and

$$\mathcal{M}_{\underline{b},\underline{x}}^{(t)+}|_{\lambda} = \mathcal{M}_{\underline{x},\underline{b}}^{(t-1)-}|_{\lambda} \ , \tag{60.38}$$

   for all $\underline{b} \in ch(\underline{x})$. By $X|_{\lambda}$ (resp., $X|_{\pi}$) we mean the λ (resp., π) component of X.

2. **Calculating $\mathcal{M}_{\underline{x}}^{(t)OUT}$ from already calculated $\mathcal{M}_{\underline{x}}^{(t)IN}$:**

   Let $\underline{a}^{na} = (\underline{a}_i)_{i=0,1,\ldots,na-1}$ denote the parents of $\underline{x}$ and $\underline{b}^{nb} = (\underline{b}_i)_{i=0,1,\ldots,nb-1}$ its children.

   Define

$$\pi_{\underline{x}}(x) = \sum_{a^{na}} P(x|a^{na}) \prod_i \pi_{\underline{x} \Leftarrow \underline{a}_i}(a_i) \tag{60.39}$$

$$= E_{\underline{a}^{na}}[P(x|a^{na})] \tag{60.40}$$

   (boundary case: if $\underline{x}$ is a root node, use $\pi_{\underline{x}}(x) = P(x)$.) and

$$\lambda_{\underline{x}}(x) = \prod_i \lambda_{\underline{b}_i \Rightarrow \underline{x}}(x) \ . \tag{60.41}$$

   (boundary case: if $\underline{x}$ is a leaf node, use $\lambda_{\underline{x}}(x) = 1$.)

   - **RULE 1: (red parent)**
     From the $\lambda_{\underline{x} \Rightarrow \underline{a}}$ panel of Fig.60.11, we get

$$\underbrace{\lambda_{\underline{x} \Rightarrow \underline{a}_i}(a_i)}_{OUT} = \mathcal{N}(!a_i) \sum_x \left[\underbrace{\lambda_{\underline{x}}(x)}_{IN} \sum_{(a_k)_{k \neq i}} \left(P(x|a^{na}) \prod_{k \neq i} \underbrace{\pi_{\underline{x} \Leftarrow \underline{a}_k}(a_k)}_{IN} \right) \right] \tag{60.42}$$

$$= \mathcal{N}(!a_i) \sum_x \left[\lambda_{\underline{x}}(x) E_{(\underline{a}_k)_{k \neq i}}[P(x|a^{na})] \right] \tag{60.43}$$

$$= \mathcal{N}(!a_i) E_{(\underline{a}_k)_{k \neq i}} E_{\underline{x}|a^{na}} \lambda_{\underline{x}}(x) \tag{60.44}$$

     (boundary case: if $\underline{x}$ is a root node, use $\lambda_{\underline{x} \Rightarrow \underline{a}_i}(a_i) = \mathcal{N}(!a_i)$.)

- **RULE 2: (red child)**
  From the $\pi_{\underline{b}\Leftarrow\underline{x}}$ panel of Fig.60.11, we get

$$\underbrace{\pi_{\underline{b}_i\Leftarrow\underline{x}}(x)}_{OUT} = \mathcal{N}(!x)\underbrace{\pi_{\underline{x}}(x)}_{IN}\prod_{k\neq i}\underbrace{\lambda_{\underline{b}_k\Rightarrow\underline{x}}(x)}_{IN} \tag{60.45}$$

  (boundary case: if $\underline{x}$ is a leaf node, use $\pi_{\underline{b}_i\Leftarrow\underline{x}}(x) = \mathcal{N}(!x)\pi_{\underline{x}}(x)$.)

In the above equations, if the range set of a product is empty, then define the product as 1; i.e., $\prod_{k\in\emptyset} F(k) = 1$.

Claim: Define

$$BEL^{(t)}(x) = \mathcal{N}(!x)\lambda^{(t)}_{\underline{x}}(x)\pi^{(t)}_{\underline{x}}(x) \ . \tag{60.46}$$

Then

$$\lim_{t\to\infty} BEL^{(t)}(x) = P(x|\epsilon) \ . \tag{60.47}$$

This says that the belief in $\underline{x} = x$ converges to $P(x|\epsilon)$ and it equals the product of messages received from all parents and children of $\underline{x} = x$.

60.4.1 How BP algo for polytrees reduces to the BP algo for Markov chains

It is instructive to see how the BP algo for polytrees reduces to BP algo for Markov chains.

For a Markov chain, node $\underline{x}$ has a single parent (i.e., ancestor) $\underline{a}$ and a single child $\underline{b}$.

Therefore, Eqs.(60.39) and (60.41) reduce to

$$\pi_{\underline{x}}(x) = \sum_a P(x|a)\pi_{\underline{x}\Leftarrow\underline{a}}(a) \tag{60.48}$$

and

$$\lambda_{\underline{x}}(x) = \lambda_{\underline{b}\Rightarrow\underline{x}}(x) \ . \tag{60.49}$$

RULE 1 given by Eq.(60.42) reduces to

$$\lambda_{\underline{x}\Rightarrow\underline{a}}(a) = \mathcal{N}(!a)\sum_x \lambda_{\underline{x}}(x)P(x|a) \tag{60.50}$$

$$= \mathcal{N}(!a)\sum_x \lambda_{\underline{b}\Rightarrow\underline{x}}(x)P(x|a) \tag{60.51}$$

RULE 2 given by Eq.(60.45) reduces to

$$\pi_{\underline{b}\Leftarrow\underline{x}}(x) = \mathcal{N}(!x)\pi_{\underline{x}}(x) \tag{60.52}$$

$$= \sum_a P(x|a)\pi_{\underline{x}\Leftarrow\underline{a}}(a) \ . \tag{60.53}$$

60.5 Derivation of BP Algorithm for Polytrees

This derivation is taken from the 1988 book Ref.[60] by Judea Pearl, where it is presented very lucidly. We only made some minor changes in notation.

Notation

The BP algorithm yields an expansion for $P(x|\epsilon)$.

$\underline{x}$= the focus node, arbitrary node of bnet that we are focusing on to calculate its $P(x|\epsilon)$.

$(\underline{a}_i)_{i=0,1,\ldots,na-1}$. = parent nodes (mnemonic: a=ancestor) of $\underline{x}$

$(\underline{b}_i)_{i=0,1,\ldots,nb-1}$. = children nodes of $\underline{x}$.

$\underline{\epsilon}$= set of nodes for which there is evidence; that is, $\underline{\epsilon} = \epsilon$, so the state of these nodes is fixed.

$\underline{\epsilon}^-_{\underline{x}} = \underline{\epsilon} \cap an(\underline{x})$ (evidence in past of $\underline{x}$)[3]

$\epsilon^-_{\underline{x}\underline{a}_i} = \epsilon^-_{\underline{x}} \cap an(\underline{a}_i)$.

Note that $\epsilon^-_{\underline{x}} = \cup_i \epsilon^-_{\underline{x}\underline{a}_i}$

$\underline{\epsilon}^+_{\underline{x}} = \underline{\epsilon} \cap [de(\underline{x}) \cup \underline{x}]$ (evidence in future of $\underline{x}$)

$\epsilon^+_{\underline{x}\underline{b}_i} = \epsilon^+_{\underline{x}} \cap [de(\underline{b}_i) \cup \underline{b}_i]$.

Note that $\epsilon^+_{\underline{x}} = \cup_i \epsilon^+_{\underline{x}\underline{b}_i}$

Note that $\underline{\epsilon} = \underline{\epsilon}^+_{\underline{x}} \cup \epsilon^-_{\underline{x}}$

$$\pi_{\underline{x}}(x) = P(x|\epsilon^-_{\underline{x}}) \tag{60.54}$$

$$\pi_{\underline{x}\Leftarrow\underline{a}_i}(a_i) = P(a_i|\epsilon^-_{\underline{x}\underline{a}_i}) \tag{60.55}$$

$$\pi_{\underline{b}_i\Leftarrow\underline{x}}(x) = P(x|\epsilon^-_{\underline{x}\underline{b}_i}) \tag{60.56}$$

$$\lambda_{\underline{x}}(x) = P(\epsilon^+_{\underline{x}}|x) \tag{60.57}$$

$$\lambda_{\underline{b}_i\Rightarrow\underline{x}}(x) = P(\epsilon^+_{\underline{x}\underline{b}_i}|x) \tag{60.58}$$

$$\lambda_{\underline{x}\Rightarrow\underline{a}_i}(a_i) = P(\epsilon^+_{\underline{x}\underline{a}_i}|a_i) \tag{60.59}$$

Expansions of $\lambda_{\underline{x}}(x)$ and $\pi_{\underline{x}}(x)$ into products of single node messages.

[3] Careful: Chapter 4 of Ref.[60] uses $-$ indicate the future and $+$ to indicate the past. This is the opposite of our notation.

$$\underbrace{P(x|\epsilon^-_{\underline{x}})}_{\pi_{\underline{x}}(x)} = P(x|\cup_i \epsilon^-_{\underline{xa}_i}) \tag{60.60}$$

$$= \sum_{a^{na}} P(x|a^{na})P(a^{na}|\cup_i \epsilon^-_{\underline{xa}_i}) \tag{60.61}$$

$$= \sum_{a^{na}} P(x|a^{na}) \prod_i \underbrace{P(a_i|\epsilon^-_{\underline{xa}_i})}_{\pi_{\underline{x}\Leftarrow\underline{a}_i}(a_i)} \tag{60.62}$$

$$\underbrace{P(\epsilon^+_{\underline{x}}|x)}_{\lambda_{\underline{x}}(x)} = \prod_i \underbrace{P(\epsilon^+_{\underline{xb}_i}|x)}_{\lambda_{\underline{b}_i\Rightarrow\underline{x}}(x)} \tag{60.63}$$

Note that past and future evidences $\epsilon^-_{\underline{x}}$ and $\epsilon^+_{\underline{x}}$ that are causally connected to $\underline{x}$ are conditionally independent at fixed $\underline{x}$:

$$P(\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{x}}|x) = P(\epsilon^+_{\underline{x}}|x)P(\epsilon^-_{\underline{x}}|x) \ . \tag{60.64}$$

Claim 89

$$P(x|\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{x}}) = P(\epsilon^+_{\underline{x}}|x)P(x|\epsilon^-_{\underline{x}})\frac{1}{P(\epsilon^+_{\underline{x}}|\epsilon^-_{\underline{x}})} \tag{60.65}$$

$$= \mathcal{N}(!x)P(\epsilon^+_{\underline{x}}|x)P(x|\epsilon^-_{\underline{x}}) \tag{60.66}$$

$$= \mathcal{N}(!x) \quad (\epsilon^+_{\underline{x}} \leftarrow x \leftarrow \epsilon^-_{\underline{x}}) \tag{60.67}$$

$$= \mathcal{N}(!x)\lambda_{\underline{x}}(x)\pi_{\underline{x}}(x) \tag{60.68}$$

proof:

$$P(x|\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{x}}) = P(\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{x}}|x)\frac{P(x)}{P(\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{x}})} \tag{60.69}$$

$$= P(\epsilon^+_{\underline{x}}|x)P(\epsilon^-_{\underline{x}}|x)\frac{P(x)}{P(\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{x}})} \tag{60.70}$$

$$= P(\epsilon^+_{\underline{x}}|x)P(x|\epsilon^-_{\underline{x}})\frac{P(\epsilon^-_{\underline{x}})}{P(\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{x}})} \tag{60.71}$$

$$= P(\epsilon^+_{\underline{x}}|x)P(x|\epsilon^-_{\underline{x}})\frac{1}{P(\epsilon^+_{\underline{x}}|\epsilon^-_{\underline{x}})} \tag{60.72}$$

QED

Next we prove BP rules 1 and 2.

Figure 60.12: This figure is used in the derivation of the BP RULE 1.

- **RULE 1 (red parent)**

  Let $y = (a_k)_{k\neq i}$ and $\epsilon^-_{\underline{y}} = (\epsilon^-_{\underline{x}\underline{a}_k})_{k\neq i}$. Note that

$$P(\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{y}}|a_i) = P(\epsilon^+_{\underline{x}} \cup \epsilon^-_{\underline{x}}) - \epsilon^-_{\underline{x}\underline{a}_i}|a_i) \tag{60.73}$$

$$= P(\epsilon^+_{\underline{x}\underline{a}_i}|a_i) \tag{60.74}$$

  Hence,

$$\underbrace{P(\epsilon^+_{\underline{x}\underline{a}_i}|a_i)}_{\lambda_{\underline{x}\Rightarrow\underline{a}_i}(a_i)} = P(\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{y}}|a_i) \tag{60.75}$$

$$= \sum_x \sum_y P(\epsilon^+_{\underline{x}}, \epsilon^-_{\underline{y}}|x, y)P(x, y|a_i) \tag{60.76}$$

$$= \sum_x \sum_y P(\epsilon^+_{\underline{x}}|x)P(\epsilon^-_{\underline{y}}|y)P(x|y, a_i)P(y|a_i) \tag{60.77}$$

$$= P(\epsilon^-_{\underline{y}}) \sum_x \sum_y P(\epsilon^+_{\underline{x}}|x)\frac{P(y|\epsilon^-_{\underline{y}})}{P(y)}P(x|y, a_i)\underbrace{P(y|a_i)}_{=P(y)} \tag{60.78}$$

$$= \mathcal{N}(!a_i) \sum_x \sum_y P(\epsilon^+_{\underline{x}}|x)P(x|\underbrace{y, a_i}_{a^{na}})P(y|\epsilon^-_{\underline{y}}) \tag{60.79}$$

$$= \mathcal{N}(!a_i) \sum_x \underbrace{P(\epsilon^+_{\underline{x}}|x)}_{\lambda_{\underline{x}}(x)} \sum_{(a_k)_{k\neq i}} P(x|a^{na}) \prod_{k\neq i} \underbrace{P(a_k|\epsilon^-_{\underline{x}\underline{a}_k})}_{\pi_{\underline{x}\Leftarrow\underline{a}_k}(a_k)} \tag{60.80}$$

- **RULE 2 (red child)**

Figure 60.13: This figure is used in the derivation of the BP RULE 2.

Note that

$$P(x|(\epsilon^+_{\underline{x}\underline{b}_k})_{k\neq i} \cup \epsilon^-_{\underline{x}}) = P(x|(\epsilon^+_{\underline{x}} \cup \epsilon^-_{\underline{x}}) - \epsilon^+_{\underline{x}\underline{b}_i}) \tag{60.81}$$

$$= P(x|\epsilon^-_{\underline{x}\underline{b}_i}) \tag{60.82}$$

Hence,

$$\underbrace{P(x|\epsilon^-_{\underline{x}\underline{b}_i})}_{\pi_{\underline{b}_i \Leftarrow \underline{x}}(x)} = P(x|(\epsilon^+_{\underline{x}\underline{b}_k})_{k\neq i}, \epsilon^-_{\underline{x}}) \tag{60.83}$$

$$= \mathcal{N}(!x)P((\epsilon^+_{\underline{x}\underline{b}_k})_{k\neq i}|x)P(x|\epsilon^-_{\underline{x}}) \tag{60.84}$$

$$= \mathcal{N}(!x)\left(\prod_{k\neq i}\underbrace{P(\epsilon^+_{\underline{x}\underline{b}_k}|x)}_{\lambda_{\underline{b}_k \Rightarrow \underline{x}}(x)}\right)\underbrace{P(x|\epsilon^-_{\underline{x}})}_{\pi_{\underline{x}}(x)} \tag{60.85}$$

60.6 Example of BP algo for a Tree

In this section, we describe how to apply the BP algo to the tree bnet Fig.60.14. In Fig.60.14, if we replace each integer i by the random variable $\underline{A}_i$, we get an **original bnet**, and if we replace each i by $\underline{\mathcal{M}}_{\underline{A}_i}$, we get the **BP memory bnet** of the original bnet. In Fig.60.14, the magenta nodes are evidence nodes and the green ones aren't.

We want to solve for the memory matrices of the memory bnet. To do so, we use the **BP dynamical bnet** Fig.60.15. The steps encoded in the dynamical

bnet are shown in Fig.60.16. Fig.60.16 has frames in chronological order, showing the direction of travel of the $\pi\&\lambda$ information. This sequence of frames also indicates the order in which we solve for the entries of the memory matrices. The information first emanates from the evidence nodes. It propagates generally upstream, although some nodes can generate downstream flow. Some of the info reaches the root node and is reflected there. The root node is the only one that is capable of reflection (i.e., instant output along an arrow, in response to input along that arrow). Eventually, all info reaches the leaf nodes via downstream propagation and is absorbed there.

Figure 60.14: Example tree bnet used to illustrate BP.

Figure 60.15: BP dynamical bnet for the bnet Fig.60.14.

Figure 60.16: Steps encoded in the bnet Fig.60.15.

60.7 Bipartite bnets

By a **bipartite bnet** we will mean a bnet in which all nodes are either root nodes (parentless) or leaf nodes (childless). BP simplifies when dealing with bipartite bnets. Next, we will explain how it simplifies. But before doing so, let us define tree bnets and show how these can be replaced by equivalent bipartite bnets.

A **tree bnet** is a bnet for which all nodes have exactly one parent except for the apex root node which has none. A tree bnet is very much like the filing system in a computer.

One can map a tree bnet (the "source") into an equivalent bipartite bnet (the "image") as follows. Replace each arrow

$$\underline{x} \longrightarrow \underline{y} \tag{60.86}$$

of the tree bnet by

$$\underline{x} \longrightarrow \underline{P_{\underline{y}|\underline{x}}} \longleftarrow \underline{y} \ . \tag{60.87}$$

For example, the tree bnet Fig.60.17 has the image bipartite bnet given by Fig.60.18. The bnet Fig.60.19 is just a different way of drawing the bnet Fig.60.18.

Figure 60.17: Example of a tree bnet.

The TPMs, printed in blue, for the image bipartite bnet Fig.60.18, are as follows. We express the TPMs of the image bnet in terms of the TPMs of the source bnet Fig.60.17. Let

$$P(P_{\underline{y}|\underline{x}}|x,y) = P_{\underline{y}|\underline{x}}(y|x)\delta(P_{\underline{y}|\underline{x}},1) + (1 - P_{\underline{y}|\underline{x}}(y|x))\delta(P_{\underline{y}|\underline{x}},0) \tag{60.88}$$

for all the leaf nodes $\underline{P_{\underline{y}|\underline{x}}} \in \{0,1\}$ of the image bipartite bnet. Also, let

$$P_{\underline{y}}(y) = \text{arbitrary prior} \tag{60.89}$$

for all the root nodes $\underline{y}$ of the image bipartite bnet except when $\underline{y}$ corresponds to the root node $\underline{A}$ of the source tree bnet. In that exceptional case,

Figure 60.18: Bipartite bnet corresponding to tree bnet Fig.60.17.

Figure 60.19: Different way of drawing the bnet Fig.60.18.

$$P_{\underline{y}}(y) = P_{\underline{A}}(y) \ . \tag{60.90}$$

60.8 BP for bipartite bnets (BP-BB)

For a bipartite bnet as defined above, with root nodes $\underline{x}_i$ and leaf nodes $\underline{f}_\alpha$, let

$$nb(i) = \{\alpha : \underline{f}_\alpha \in nb(\underline{x}_i)\} \ , \tag{60.91}$$

$$nb(\alpha) = \{i : \underline{x}_i \in nb(\underline{f}_\alpha)\} \ , \tag{60.92}$$

$$m_{\alpha \Leftarrow i}(x_i) = \pi_{\underline{f}_\alpha \Leftarrow \underline{x}_i}(x_i) \ , \tag{60.93}$$

$$m_{\alpha \Rightarrow i}(x_i) = \lambda_{\underline{f}_\alpha \Rightarrow \underline{x}_i}(x_i) \ , \tag{60.94}$$

Figure 60.20: Fig.60.11 becomes this figure for the special case of a bipartite bnet. Union of green nodes and the red node = full neighborhood of yellow node. There are two possible cases: the red node is either a parent or a child of the yellow node.

Next we will show how to find $m^{(t)}_{\alpha \Leftarrow i}$ and $m^{(t)}_{\alpha \Rightarrow i}$ from $m^{(t-1)}_{\alpha \Leftarrow i}$ and $m^{(t-1)}_{\alpha \Rightarrow i}$.

1. **Traversing an x (i.e., root) node.**

   See the $m_{\underline{f}_2 \Leftarrow \underline{x}_2}$ panel of Fig.60.20.

   For $i = 0, 1, \ldots, nx - 1$, if $\alpha \in nb(i)$, then,

$$m^{(t)}_{\alpha \Leftarrow i}(x_i) = \prod_{\beta \in nb(i) - \alpha} m^{(t-1)}_{\beta \Rightarrow i}(x_i) \ , \tag{60.95}$$

   whereas if $\alpha \notin nb(i)$

$$m^{(t)}_{\alpha \Leftarrow i}(x_i) = m^{(t-1)}_{\alpha \Leftarrow i}(x_i) \ . \tag{60.96}$$

2. **Traversing an f (i.e., leaf) node.**

   See the $m_{\underline{f}_2 \Rightarrow \underline{x}_2}$ panel of Fig.60.20.

   For $\alpha = 0, 1, \ldots, nf - 1$, if $i \in nb(\alpha)$, then

$$m^{(t)}_{\alpha \Rightarrow i}(x_i) = \sum_{(x_k)_{k \in nb(\alpha) - i}} f_\alpha(x_{nb(\alpha)}) \prod_{k \in nb(\alpha) - i} m^{(t-1)}_{\alpha \Leftarrow k}(x_k) \tag{60.97}$$

$$= E^{(t-1)}_{(x_k)_{k \in nb(\alpha) - i}}[f_\alpha(x_{nb(\alpha)})] \ , \tag{60.98}$$

   whereas if $i \notin nb(\alpha)$

$$m^{(t)}_{\alpha \Rightarrow i}(x_i) = m^{(t-1)}_{\alpha \Rightarrow i}(x_i) \ . \tag{60.99}$$

In the above equations, if the range set of a product is empty, then define the product as 1; i.e., $\prod_{k \in \emptyset} F(k) = 1$.

Claim:

$$P(x_i|\epsilon) = \lim_{t\to\infty} \mathcal{N}(!x_i) \prod_{\alpha\in nb(i)} m^{(t)}_{\alpha\Rightarrow i}(x_i) \tag{60.100}$$

and

$$P(x_{nb(\alpha)}|\epsilon) = \lim_{t\to\infty} \mathcal{N}(!x_{nb(\alpha)}) f_\alpha(x_{nb(\alpha)}) \prod_{k\in nb(\alpha)} m^{(t)}_{\alpha\Leftarrow k}(x_k) \ . \tag{60.101}$$

60.8.1 BP-BB and general BP agree on Markov chains

It is instructive to compare the belief values (i.e., $P(x_i|\epsilon)$) obtained by applying the general (i.e., polytree) BP and BP-BB algorithms to a Markov chain. Next we show that both algorithms yield the same belief values.

Figure 60.21: Traversing a root node of a Markov chain (a)Propagation towards left (i.e., towards future). (b)Propagation towards right (i.e., towards past).

Consider the BP-BB rule for traversing a root node. When traveling towards the left as in Fig.60.21 (a), it implies that

$$m_{\alpha\Rightarrow 2}(x_2) = m_{\beta\Leftarrow 2}(x_2) \ , \tag{60.102}$$

and when traveling towards the right as in Fig.60.21 (b), it implies that

$$m_{\beta\Rightarrow 2}(x_2) = m_{\alpha\Leftarrow 2}(x_2) \ . \tag{60.103}$$

Now consider the BP-BB rule for traversing a leaf node. When traveling to the left as in Fig.60.22 (a), it implies that

$$\underbrace{m_{\alpha\Rightarrow 2}(x_2)}_{\lambda} = \sum_{x_1} P(x_2|x_1) \underbrace{m_{\alpha\Leftarrow 1}(x_1)}_{\pi} \ . \tag{60.104}$$

One can rewrite the left and right hand sides (LHS, RHS) of Eq.(60.104) as follows

$$RHS = \sum_{x_1} P(x_2|x_1) \pi_{\alpha\Leftarrow 1}(x_1) \ , \tag{60.105}$$

Figure 60.22: Traversing a leaf node of a Markov chain (a)Propagation towards left (i.e., towards future). (b)Propagation towards right (i.e., towards past).

and

$$LHS = m_{\alpha\Rightarrow 2}(x_2) = m_{\beta\Leftarrow 2}(x_2) = \pi_{\beta\Leftarrow 2}(x_2) \ , \tag{60.106}$$

Therefore

$$\pi_{\beta\Leftarrow 2}(x_2) \sum_{x_1} P(x_2|x_1)\pi_{\alpha\Leftarrow 1}(x_1) \ . \tag{60.107}$$

Once again, consider the BP-BB rule for traversing a leaf node. When traveling to the right as in Fig.60.22 (b), it implies that

$$\underbrace{m_{\alpha\Rightarrow 1}(x_1)}_{\lambda} = \sum_{x_2} P(x_2|x_1) \underbrace{m_{\alpha\Leftarrow 2}(x_2)}_{\pi} \ . \tag{60.108}$$

One can rewrite the left and right hand sides (LHS, RHS) of Eq.(60.108) as follows

$$RHS = \ = \sum_{x_2} P(x_2|x_1)\pi_{\alpha\Leftarrow 2}(x_2) \tag{60.109}$$

$$= \sum_{x_2} P(x_2|x_1)\lambda_{\beta\Rightarrow 2}(x_2) \ , \tag{60.110}$$

and

$$LHS = \lambda_{\alpha\Rightarrow 1}(x_1) \ . \tag{60.111}$$

Therefore,

$$\lambda_{\alpha\Rightarrow 1}(x_1) = \sum_{x_2} P(x_2|x_1)\lambda_{\beta\Rightarrow 2}(x_2) \ . \tag{60.112}$$

Finally, note that Eq.(60.100) becomes

$$
\begin{aligned}
P(x_2|\epsilon) &= \mathcal{N}(!x_2)m_{\beta\Rightarrow 2}(x_2)m_{\alpha\Rightarrow 2}(x_2) && (60.113)\\
&= \mathcal{N}(!x_2)m_{\alpha\Leftarrow 2}(x_2)m_{\alpha\Rightarrow 2}(x_2) && (60.114)\\
&= \mathcal{N}(!x_2)\pi_{\alpha\Leftarrow 2}(x_2)\lambda_{\alpha\Rightarrow 2}(x_2) && (60.115)\\
&= \mathcal{N}(!x_2)P(x_2|\epsilon^-)P(x_2|\epsilon^+) && (60.116)
\end{aligned}
$$

and Eq.(60.101) becomes

$$
\begin{aligned}
P(x_2, x_1) &= \mathcal{N}(!x_2, !x_1)P(x_2|x_1)m_{\alpha\Leftarrow 1}(x_1)m_{\alpha\Leftarrow 2}(x_2) && (60.117)\\
&= \mathcal{N}(!x_2, !x_1)P(x_2|x_1)\pi_{\alpha\Leftarrow 1}(x_1)\pi_{\alpha\Leftarrow 2}(x_2)\ . && (60.118)
\end{aligned}
$$

60.8.2 BP-BB and general BP agree on tree bnets.

It is instructive to compare the belief values (i.e., $P(x_i|\epsilon)$) obtained by applying the general (i.e., polytree) BP and BP-BB algorithms to a tree bnet. Next we show that both algorithms yield the same belief values.

Figure 60.23: Subgraph of a tree bnet. This is the same as Fig.60.11, except that here the yellow node has a single parent because this is a subgraph of a tree bnet, not of an arbitrary bnet like Fig.60.11. The subgraph has been converted to a subgraph of a bipartite bnet by inserting a collider leaf node, labeled by a Greek letter, at the center of each edge of the tree bnet. Red arrows indicate the direction of message info flow.

Applying to the left panel of Fig.60.23 the BP-BB rule for traversing a root node, we get

$$m_{\alpha \Leftarrow \underline{x}}(x) = \prod_i m_{\beta_i \Rightarrow \underline{x}}(x) \ . \tag{60.119}$$

Applying to the left panel of Fig.60.23 the BP-BB rule for traversing a leaf node, we get

$$m_{\alpha \Rightarrow \underline{a}}(a) = \mathcal{N}(!a) \sum_x m_{\alpha \Leftarrow \underline{x}}(x) P(x|a) \ . \tag{60.120}$$

Combining Eqs.(60.119) and (60.120), we get

$$m_{\alpha \Rightarrow \underline{a}}(a) = \mathcal{N}(!a) \sum_x P(x|a) \prod_i m_{\beta_i \Rightarrow \underline{x}}(x) \ , \tag{60.121}$$

which can be rewritten as

$$\lambda_{\underline{x} \Rightarrow \underline{a}}(a) = \mathcal{N}(!a) \sum_x P(x|a) \underbrace{\prod_i \lambda_{\underline{b}_i \Rightarrow \underline{x}}(x)}_{\lambda_{\underline{x}}(x)} \ . \tag{60.122}$$

Eq.(60.122) is just RULE 1 for general BP.

Applying to the right panel of Fig.60.23 the BP-BB rule for traversing a root node, we get

$$m_{\beta_i \Leftarrow \underline{x}}(x) = \mathcal{N}(!x) m_{\alpha \Rightarrow \underline{x}}(x) \prod_{k \neq i} m_{\beta_k \Rightarrow \underline{x}}(x) \tag{60.123}$$

Applying to the right panel of Fig.60.23 the BP-BB rule for traversing a leaf node, we get

$$\begin{aligned} m_{\alpha \Rightarrow \underline{x}}(x) &= \sum_a P(x|a) m_{\alpha \Leftarrow a}(a) && (60.124) \\ &= \sum_a P(x|a) \pi_{\underline{x} \Leftarrow a}(a) && (60.125) \\ &= \pi_{\underline{x}}(x) \ . && (60.126) \end{aligned}$$

Combining Eqs.(60.123) and (60.126), we get

$$\pi_{\underline{b}_i \Leftarrow \underline{x}}(x) = \mathcal{N}(!x) \pi_{\underline{x}}(x) \prod_{k \neq i} \lambda_{\underline{b}_k \Rightarrow \underline{x}}(x) \ . \tag{60.127}$$

Eq.(60.127) is just RULE 2 of general BP.

60.9 BP-BB and sum-product decomposition

BP-BB yields what is often referred to as a **sum-product decomposition**. I don't like that name because it is unnecessarily confusing, and it fails to convey the recursive

nature[4] of the decomposition. I prefer to call it a **recursive sum of products (RSOP) decomposition**, and will call it so henceforth in this chapter.

Expressing the marginals of a bnet as RSOPs, which is what BP does, reduces the complexity of the calculation. (i.e., the total number of additions and multiplications that need to be performed) That makes using the BP algo very advantageous. For instance, consider a Markov chain $\underline{x}_{n-1} \leftarrow \cdots \leftarrow \underline{x}_1 \leftarrow \underline{x}_0$, where $x_i \in \{0, 1, 2\}$ for all i. Note that if we calculate $P(x_{n-1})$ as follows

$$P(x_{n-1}) = \left[\sum_{x_{n-2}} P(x_{n-1}|x_{n-2}) \ldots \left[\sum_{x_1} P(x_2|x_1) \left[\sum_{x_0} P(x_1|x_0) P(x_0) \right] \right] \ldots \right] , \quad (60.128)$$

we need to perform $2(n-1)$ additions and $3(n-1)$ multiplications. On the other hand, if we calculate $P(x_{n-1})$ as follows

$$P(x_{n-1}) = \sum_{x_{n-2}} \ldots \sum_{x_1} \sum_{x_0} P(x_{n-1}|x_{n-2}) \ldots P(x_2|x_1) P(x_1|x_0) P(x_0) , \quad (60.129)$$

we need to perform $3^n - 1$ additions and $3^n(n-1)$ multiplications.

[4] By "recursive nature", we mean bootstrapped definitions that lead to nested sums. The recursive nature of BP is evident from RULES 1 and 2 that define λ's and π's in terms of other λ's and π's.

Chapter 61

Message Passing in Quantum Mechanics

See Ref.[93].

Chapter 62

Meta-learners for estimating ATE

This section is based on the final 2 chapters of Ref.[18].

The Average Treatment Effect (ATE) is defined in Chapter 77.

Economists are huge fans of Linear Regression (LR), and traditionally calculate ATE using LR. But in recent times, they have begun to calculate ATE using Machine Learning (ML) instead. This chapter describes various methods that economists and others have devised for calculating ATE with ML. These methods are called **meta-learners** because they involve multiple ML or LR steps.

Using ML to calculate ATE captures non-linear trends whose exclusion might sometimes lead to a poor result. On the other hand, ML is more expensive computationally than LR, and it introduces the danger of overfitting, a danger which is nonexistent with LR.

Below, we represent each Linear Regression (LR) step as follows. We list a dataset; i.e., a set of tuples indexed by the individuals σ of a population Σ such that $|\Sigma| = nsam$. The independent variables of the LR (i.e., x^σ) are unboxed and the dependent variable (a.k.a. target feature) (i.e., y^σ) is shown inside a box. Then we show an arrow with the superscript "LR-fit", followed by the fit function obtained by performing the LR.[1]

$$\{(\sigma, x^\sigma = [x_i^\sigma], \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{LR}-\text{fit}} \widehat{y}(x) = \alpha + x_i\beta_i \tag{62.1}$$

Analogously, below, we represent each Supervised Machine Learning (ML) step as follows.

$$\{(\sigma, x^\sigma, \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML}-\text{fit}} \widehat{y}(x) \tag{62.2}$$

Henceforth, we will use $\delta(x)(\approx \text{ATE})$ to denote the treatment effect.

- **S (Single)-learner**

[1] In this chapter, we will occasionally use the Einstein summation convention; i.e., implicit sum over repeated indices.

$$\{(\sigma, x^\sigma, d^\sigma, \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{y}(d, x) \tag{62.3}$$

$$\delta(x) = \widehat{y}(1, x) - \widehat{y}(0, x) \tag{62.4}$$

- **T (Twin)-learner**

$$\{(\sigma, x^\sigma, d^\sigma = 0, \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{y}_0(x) \tag{62.5}$$

$$\{(\sigma, x^\sigma, d^\sigma = 1, \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{y}_1(x) \tag{62.6}$$

$$\delta(x) = \widehat{y}_1(x) - \widehat{y}_0(x) \tag{62.7}$$

- **X (Cross)-learner**

  do T-learner, get $\widehat{y}_0(x)$, $\widehat{y}_1(x)$

$$\{(\sigma, x^\sigma, d^\sigma = 0, \boxed{y^\sigma - \widehat{y}_1(x^\sigma)}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \mathcal{Y}_0(x) \tag{62.8}$$

$$\{(\sigma, x^\sigma, d^\sigma = 1, \boxed{y^\sigma - \widehat{y}_0(x^\sigma)}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \mathcal{Y}_1(x) \tag{62.9}$$

$$\delta(x) = \frac{1}{2}[\mathcal{Y}_1(x) - \mathcal{Y}_0(x)] \tag{62.10}$$

- **De-biased (a.k.a. Orthogonal) ML**

  Standard supervised ML is performed with two features, the independent feature and the dependent or target feature. Supervised ML has a target feature. Unsupervised ML doesn't.

  In **De-biased LR**, we do LR with two residuals. Let's call them the **independent residual** and the **dependent or target residual**. These two residuals are calculated with the help of two previously performed LR steps.

  In **De-biased ML**, we do ML or LR (either one) with two residuals. These two residuals are calculated with the help of two previously performed ML steps (instead of two LR steps).

  The FWL theorem discussed in Chapter 31 shows how to do De-biased LR. Next, we will describe how to do De-biased ML.

  We start by doing two ML steps:

$$\{(\sigma, x^\sigma, \boxed{d^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{d}(x) \tag{62.11}$$

$$\{(\sigma, x^\sigma, \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{y}(x) \tag{62.12}$$

After these two ML steps, we do either LR or ML to get $\delta(x)$.
Let

$$\Delta d^\sigma = d^\sigma - \widehat{d}(x^\sigma) \tag{62.13}$$

$$\Delta y^\sigma = y^\sigma - \widehat{y}(x^\sigma) \tag{62.14}$$

Options for the final learning step that calculates $\delta(x)$:

1. Do a standard LR to get a $\delta(x)$ that is a constant (i.e., x independent):

$$\{(\sigma, \Delta d^\sigma, \boxed{\Delta y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{LR-fit}} \mathcal{Y}(\Delta d) = \alpha + \Delta d \delta \tag{62.15}$$

$$\delta = \text{coefficient of } \Delta d \text{ in } \mathcal{Y}(\Delta d). \tag{62.16}$$

2. Do a LR with a $x * d$ cross term to get a $\delta(x)$ that is linear in x:

$$\{(\sigma, x^\sigma, \Delta d^\sigma, \boxed{\Delta y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{LR-fit}} \mathcal{Y}(\Delta d, x) = \alpha_0 + x\beta_0 + \Delta d(\alpha_1 + x\beta_1) \tag{62.17}$$

$$\delta(x) = \mathcal{Y}(\Delta d|_{d=1}, x) - \mathcal{Y}(\Delta d|_{d=0}, x) = \alpha_1 + x\beta_1 \tag{62.18}$$

3. Do ML with a weighted cost function to get a general fit $\widehat{\delta}(x)$.
   The cost function used in standard ML (see Eq.(62.2)) is:

$$\mathcal{C} = \frac{1}{nsam} \sum_\sigma [y^\sigma - \widehat{y}(x^\sigma)]^2 \tag{62.19}$$

   Define the cost function in this case as

$$\mathcal{C} = \frac{1}{nsam} \sum_\sigma \left[\Delta y^\sigma - \widehat{\delta}(x^\sigma)\Delta d^\sigma\right]^2 \tag{62.20}$$

$$= \frac{1}{nsam} \sum_\sigma [\Delta d^\sigma]^2 \left[\frac{\Delta y^\sigma}{\Delta d^\sigma} - \widehat{\delta}(x^\sigma)\right]^2 \tag{62.21}$$

   This is a weighted cost function with weights $[\Delta d^\sigma]^2$.

$$\{(\sigma, x^\sigma, \boxed{\frac{\Delta y^\sigma}{\Delta d^\sigma}}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{\delta}(x) \tag{62.22}$$

4. Do ML with Δd^σ as an independent feature to get a general fit $\widehat{\delta}(\Delta d, x)$:

$$\{(\sigma, x^\sigma, \Delta d^\sigma, \boxed{\frac{\Delta y^\sigma}{\Delta d^\sigma}}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{\delta}(\Delta d, x) \tag{62.23}$$

$$\delta(x) = \widehat{\delta}(\Delta d|_{d=1}, x) \tag{62.24}$$

Chapter 63

Missing Data, Imputation

This chapter assumes that the reader has read some parts of Chapter 27 on the Expectation Maximization (EM) algo and Chapter 55 on Markov Chain Monte Carlo (MCMC).

	h_0	x_0	x_1	x_2
1	NA	0	1	1
2	NA	0	0	0
3	NA	1	1	0
4	NA	NA	1	NA
5	NA	0	NA	1
6	NA	0	0	1

	h_0	x_0	x_1	x_2	m
1	NA	0	1	1	(0,0,0)
2	NA	0	0	0	(0,0,0)
3	NA	1	1	0	(0,0,0)
4	NA	0 0 1 1	1	0 1 0 1	(1,0,1)
5	NA	0	0 1	1	(0,1,0)
6	NA	0	0	1	(0,0,0)

Table 63.1: **Left Table:** Dataset with $nsam = 6$ and some missing entries, for 4 binary variables h_0, x_0, x_1, x_2. NA=not available. The h_0 column is completely missing because h_0 is an unobserved variable. **Right Table:** All possibilities for $x_i = NA$ cells of left table have been enumerated. A new column labeled m has been added. $m_i = \mathbb{1}(x_i \text{ is missing})$ for $i = 0, 1, 2$.

Suppose that you have compiled a **dataset** $\vec{x} = (x[\sigma])_{\sigma=0,1,\ldots,nsam-1}$ where $x = (x_0, x_1, \ldots, x_{nx-1})$ from a study or survey. It consists of $nsam$ number of samples (sample= row), and nx columns (each column is a different feature, or observation). Suppose that some of the cells in this matrix are empty. Throwing away all the incomplete rows is okay if the number of incomplete rows is much smaller than $nsam$. If not, throwing them away would throw away a substantial amount of information contained in all the filled cells in those incomplete rows, plus it might bias your dataset. This chapter deals with how to fill those empty cells with plausible fake data. A fancy name for this process is **imputation**. There is no unique way of

fabricating fake data, but some fakes are better than others by some metrics. This chapter will consider two popular ways (EM and MCMC) of filling those empty cells with their "most likely" values based on the cells of the dataset that aren't missing, and also based on some bnet model that is expected to describe well the dataset.

Notation: $\underline{\vec{a}} = (\underline{a}[\sigma])_{\sigma=0,1,\ldots,nsam-1}$, where $nsam$ is the number of samples. Will sometimes denote $a[\sigma]$ by a^σ.

For concreteness, we will apply the concepts of this chapter to the dataset with missing data given by Table 63.1.

63.1 Imputation via EM

We begin by augmenting Fig.27.1 (the first figure in Chapter 27) by adding to it a new node $\underline{\vec{m}}$ called the **missingness variable**. Recall that node $\underline{\theta}$ represents the **unknown parameters**, node $\underline{\vec{x}}$ represents the **observed variables**, and node $\underline{\vec{h}}$ represents the **hidden variables**. Both $\underline{\theta}$ and $\underline{\vec{h}}$ are hidden (i.e., unobserved). Fig.63.1 shows 3 popular ways of connecting node $\underline{\vec{m}}$ to the other nodes in the graph Fig.27.1.

Figure 63.1: The left bnet is seldom assumed. The middle bnet is referred to as the MAR (missing at random) assumption. The right bnet is referred to as the not-MAR (NMAR) assumption.

From Fig.63.1, we have

$$P(\vec{m}|\vec{x},\vec{h},\theta) = \begin{cases} P(\vec{m}|\theta) & \text{Seldom assumed. Called missing-CAR (MCAR)} \\ P(\vec{m}|\vec{x},\theta) & \text{MAR} \\ P(\vec{m}|\vec{x},\vec{h},\theta) & \text{not-MAR (NMAR)} \end{cases} . \tag{63.1}$$

For doing imputation via EM, we connect node $\underline{\vec{m}}$ as shown in the middle bnet (called MAR) of Fig.63.1.

For the example of Table 63.1, we have variables $\underline{\vec{m}}, \underline{\vec{x}}$ and $\underline{\vec{h}}$ whose values range over the following sets:

$\underline{\vec{x}} = (\underline{\vec{x}}_0, \underline{\vec{x}}_1, \underline{\vec{x}}_2)$

$\underline{\vec{h}} = (\underline{\vec{h}}_0)$

Figure 63.2: MAR bnet with $nsam = 3$.

$\underline{h}_0[\sigma] \in \{0, 1\}$,
$\underline{x}_i[\sigma] \in \{0, 1\}$ for $i = 0, 1, 2$,
$\underline{m}_i[\sigma] \in \{0, 1\}$ for $i = 0, 1, 2$.

Figure 63.3: Our example for imputation via EM assumes this bnet between nodes $\underline{m}[\sigma], \underline{x}[\sigma], \underline{h}[\sigma]$.

For concreteness, we will assume that the Markov chain $\underline{m}[\sigma] \leftarrow \underline{x}[\sigma] \leftarrow \underline{h}[\sigma]$ has a finer grained DAG structure given by Fig.63.3. where we will omit the dashed arrows. If one doesn't want to assume that the data can be fitted well by the bnet of Fig.63.3 without the dashed arrows, one can include those arrows too, at the expense of more unknown parameters (i.e., degrees of freedom) to be lumped into θ. We will parameterize the TPMs corresponding to Fig.63.3 using a Categorical Distribution for each column of the TPMs. We will thus assume that the TPMs, printed in blue, for bnet Fig.63.3, are as follows.

$$P(h_0^\sigma|\theta) = \begin{array}{c|c} & \\ \hline & 1-\theta_0 \\ 1 & \theta_0 \end{array} \tag{63.2}$$

$$P(x_0^\sigma|\theta) = \begin{array}{c|c} & \\ \hline 0 & 1-\theta_1 \\ 1 & \theta_1 \end{array} \tag{63.3}$$

$$P(x_1^\sigma \mid x_0^\sigma, h^\sigma, \theta) = \begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ \hline 0 & 1-\theta_2 & 1-\theta_3 & 1-\theta_4 & 1-\theta_5 \\ 1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 \end{array} \tag{63.4}$$

$$P(x_2^\sigma \mid x_1^\sigma, x_0^\sigma, \theta) = \begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ \hline 0 & 1-\theta_6 & 1-\theta_7 & 1-\theta_8 & 1-\theta_9 \\ 1 & \theta_6 & \theta_7 & \theta_8 & \theta_9 \end{array} \tag{63.5}$$

$$P(m^\sigma|x^\sigma, \theta) = \frac{1}{nsam} P((x_i)_{\forall i \ni m_i=1} \mid (x_i)_{\forall i \ni m_i=0}, \theta) \tag{63.6}$$

Eq.(63.6) can be illustrated as follows. In Table 63.2, we added a $P(m)$ column to Table 63.1.

	h_0	x_0	x_1	x_2	m	$P(m)$
1	NA	0	1	1	(0,0,0)	$\frac{1}{nsam}$
2	NA	0	0	0	(0,0,0)	$\frac{1}{nsam}$
3	NA	1	1	0	(0,0,0)	$\frac{1}{nsam}$
4	NA	0 0 1 1	1	0 1 0 1	(1,0,1)	$\frac{1}{nsam}P(x_0=0, x_2=0 \mid x_1=1, \theta)$ $\frac{1}{nsam}P(x_0=0, x_2=1 \mid x_1=1, \theta)$ $\frac{1}{nsam}P(x_0=1, x_2=0 \mid x_1=1, \theta)$ $\frac{1}{nsam}P(x_0=1, x_2=1 \mid x_1=1, \theta)$
5	NA	0	0 1	1	(0,1,0)	$\frac{1}{nsam}P(x_1=0 \mid x_0=0, x_2=1, \theta)$ $\frac{1}{nsam}P(x_1=1 \mid x_0=0, x_2=1, \theta)$
6	NA	0	0	1	(0,0,0)	$\frac{1}{nsam}$

Table 63.2: $P(m)$ column added to Table 63.1. Note that $\sum_m P(m) = 1$.

$$\theta = (\theta_i)_{i=0,1,\ldots,9} \tag{63.7}$$

$$P(m^\sigma, x^\sigma, h^\sigma|\theta) = P(m^\sigma|x^\sigma, \theta)P(x^\sigma|h^\sigma, \theta)P(h^\sigma|\theta) \tag{63.8}$$

$$P(x^\sigma|h^\sigma, \theta) = P(x_2^\sigma|x_1^\sigma, x_0^\sigma, \theta)P(x_1^\sigma|x_0^\sigma, h^\sigma, \theta)P(x_0^\sigma|\theta) \tag{63.9}$$

$$P(x_1^\sigma|x_0^\sigma,\theta) = \sum_h P(x_1^\sigma|x_0^\sigma,h^\sigma,\theta)P(h^\sigma|\theta) \tag{63.10}$$

$$P(x^\sigma|\theta) = P(x_2^\sigma|x_1^\sigma,x_0^\sigma,\theta)P(x_1^\sigma|x_0^\sigma,\theta)P(x_0^\sigma|\theta) \tag{63.11}$$

$$\begin{aligned}
Q(\theta|\theta^{(t)}) &= \sum_{\vec{m},\vec{h}} P(\vec{m},\vec{h}\mid\vec{x},\theta^{(t)})\ln P(\vec{m},\vec{x},\vec{h}|\theta) && (63.12)\\
&= \sum_{\vec{m},\vec{h}}\left[\prod_\sigma P(m^\sigma,h^\sigma\mid x^\sigma,\theta^{(t)})\right]\ln\left[\prod_\sigma P(m^\sigma,x^\sigma,h^\sigma|\theta)\right] && (63.13)\\
&= \sum_\sigma\sum_{m^\sigma,h^\sigma} P(m^\sigma,h^\sigma\mid x^\sigma,\theta^{(t)})\ln P(m^\sigma,x^\sigma,h^\sigma|\theta) && (63.14)\\
&= \sum_\sigma\sum_{m^\sigma,h^\sigma}\frac{P(m^\sigma,h^\sigma,x^\sigma\mid\theta^{(t)})}{P(x^\sigma\mid\theta^{(t)})}\ln P(m^\sigma,x^\sigma,h^\sigma|\theta) && (63.15)
\end{aligned}$$

Once you find optimal parameters θ^* by recursing this $Q(\theta|\theta^{(t)})$, you can evaluate numerically the $P(m)$ column of Table 63.2. In Table 63.2, out of the 4 sub-rows for row 4, choose the one with the highest probability. Similarly, out of the 2 sub-rows for row 5, choose the one with the highest probability.

63.2 Imputation via MCMC

A simple and popular way to do imputation via MCMC is described in Ref.[81]. It goes as follows.

Let

$$\underline{H}[\sigma] = (\underline{h}[\sigma],\underline{m}[\sigma]) \tag{63.16}$$

for $\sigma = 0,1,\ldots,nsam-1$. Initialize $\theta^{(0)}$ to a random value within the allowed ranges. Do the following 2 steps, for $t = 0,1,\ldots,T-1$, where T is large enough that $\theta^{(t)}$ has reached a steady value that is independent of $\theta^{(0)}$. To do the sampling, use a standard sampling technique such as Gibbs sampling.

- **STEP 1:** For $\sigma = 0,1,\ldots,nsam-1$, find a sample

$$(H^\sigma)^{(t+1)} \sim P(H^\sigma|x^\sigma,\theta^{(t)})\ . \tag{63.17a}$$

- **STEP 2:** Find a sample

$$\theta^{(t+1)} \sim P^{(t+1)}(\theta) \tag{63.17b}$$

where

$$\begin{aligned}
P^{(t+1)}(\theta) &= \mathcal{N}(!\theta)P(\vec{x},\vec{H}^{(t+1)}|\theta) && (63.17c)\\
&= \mathcal{N}(!\theta)\prod_\sigma P(x^\sigma,(H^\sigma)^{(t+1)}|\theta)\ . && (63.17d)
\end{aligned}$$

Fig.63.4 illustrates this two step recursive process using a bnet.

Figure 63.4: bnet illustrating Eqs.(63.17) for doing imputation via MCMC. The **same** node $\underline{\vec{x}}$ appears twice to make the graph clearer.

63.3 Multiple Imputations

Multiple imputations means calculating θ^* (i.e., the optimum θ) and the concomitant dataset $\vec{x}^*, \vec{H}^*$, via any method (such as EM or MCMC), a large number of times, starting from different, randomly chosen $\theta^{(0)}$ initial parameters. Then calculating the average and the variance of $\theta^*, \vec{x}^*, \vec{H}^*$ and functions thereof.

Chapter 64

Modified Treatment Policy

This chapter is based primarily on Ref. [27]. Refs. [29] and [30] were also very helpful.

A **Modified Treatment Policy (MTP)** is a generalization of the Potential Outcomes (PO)[1] model so as to modify the original treatment. This is accomplished by adding a modified treatment dose node $\underline{\widetilde{x}}$ acting as a mediator between the treatment dose node $\underline{x}$ and the treatment effect node $\underline{y}$ (i.e., by considering $\underline{x} \to \underline{\widetilde{x}} \to \underline{y}$).

64.1 One time MTP

Consider a typical PO bnet G and its corresponding imagined bnet G_{im2} (see Fig.64.1)[2]. A MTP adds a Bayesian prior to the node $\underline{\widetilde{x}}$ in G_{im2}. The prior depends on the treatment dose variable (a.k.a. exposure variable) $\underline{x}$ and its parents $pa(\underline{x}) = \underline{c}$. This adds to G_{im2} new arrows $\underline{c} \to \underline{\widetilde{x}}$ and $\underline{x} \to \underline{\widetilde{x}}$ (see Fig.64.2).

The TPM, printed in blue, for node $\underline{\widetilde{x}}$ of the imagined bnet G_{im2} in Fig.64.2, is as follows

$$P(\underline{\widetilde{x}} = \widetilde{x}; x) = \delta(\widetilde{x}, x) \tag{64.1}$$

The TPM, printed in blue, for node $\underline{\widetilde{x}}$ of the modified imagined bnet $G_{im2,mod}$ in Fig.64.2, is as follows

$$P(\underline{\widetilde{x}} = \widetilde{x} | \underline{x} = x, c) = \text{ prior} \tag{64.2}$$

Hence, node $\underline{\widetilde{x}}$ has a totally informative, parametric (frequentist) prior in G_{im2}, and it has a more general (Bayesian) prior in $G_{im2,mod}$.

The following assumptions will be made about bnet $G_{im2,mod}$:

1. **Consistency (a.k.a. SUTVA)**

[1] PO theory is discussed in Chapter 77.

[2] Ref.[27] uses the notation $L = \underline{c}$, $A = \underline{x}$, $Y = \underline{y}$, $Q = \underline{\widetilde{x}}$, $Y_q = \underline{y}(\widetilde{x})$. Furthermore, it does not display a DAG.

(As usual in this book, $val(\underline{a})$ denotes the set of values that a random variable $\underline{a}$ can assume. Imagine operators such as $\mathcal{I}_{\underline{x}\to\underline{y}}$ and $\mathcal{I}_{\underline{x}\to\underline{y}}(\widetilde{x})$ are discussed in Chapter 12.)

Figure 64.1: G is one of the simplest possible bnets considered in PO theory. In the usual PO theory, one only considers the bnet G_{im1}. In MTP theory, one considers the bnet G_{im2}. For G_{im2}, $\widetilde{x} \in val(\underline{\widetilde{x}}) = val(\underline{x}) = \{0, 1\}$.

Figure 64.2: In this figure, G_{im2} is the generalization of the G_{im2} in Fig.64.1. Note that $pa'(\underline{y})$ are the parents in G of $\underline{y}$, excluding $\underline{x}$. Note that $val(\underline{\widetilde{x}}) \subset val(\underline{x})$.

The TPM, printed in blue, for node $\underline{y}$ of the modified imagined bnet $G_{im2,mod}$ in Fig.64.2, is as follows

$$P(y \mid \widetilde{x}, \{y(\widetilde{x}')\}_{\widetilde{x}'\in val(\underline{\widetilde{x}})}) = \mathbb{1}(\quad y = \sum_{\widetilde{x}'\in val(\underline{\widetilde{x}})} y(\widetilde{x}')\mathbb{1}(\widetilde{x}' = \widetilde{x}) \quad) \tag{64.3}$$

When $val(\underline{\widetilde{x}}) = \{0, 1\}$, this becomes the more familiar statement from standard PO theory:

$$P(y \mid \widetilde{x}, \{y(0), y(1)\}) = \mathbb{1}(\quad y = y(0)(1 - \widetilde{x}) + y(1)\widetilde{x} \quad) \tag{64.4}$$

An immediate consequence of this deterministic TPM for node $\underline{y}$ is that

$$P(\underline{y} = y|\widetilde{x}, pa'(\underline{y}) = \xi) \quad = \quad P(\underline{y}(\widetilde{x}) = y|\widetilde{x}, pa'(\underline{y}) = \xi) \tag{64.5}$$

2. **Ignorability (a.k.a. conditional independence assumption (CIA))**

$$P(\underline{y}(\widetilde{x}') = y|\widetilde{x}, pa'(\underline{y}) = \xi) \quad = \quad P(\underline{y}(\widetilde{x}') = y|pa'(\underline{y}) = \xi) \tag{64.6}$$

CIA follows from the structure of the DAG $G_{im2,mod}$, because in that DAG, fixing the value of $pa'(\underline{y})$ blocks messages from $\underline{\widetilde{x}}$ to $\underline{y}(\widetilde{x})$ (i.e., for all $\widetilde{x} \in val(\underline{\widetilde{x}})$, we have $\underline{\widetilde{x}} \perp \underline{y}(\widetilde{x})|pa'(\underline{y})$.)

3. **Identifiability**

We must have

$$val(\underline{\widetilde{x}}) \subset val(\underline{x}) \tag{64.7}$$

in $G_{im2,mod}$ or else queries of the type $P(\underline{y}(\widetilde{x}) = y|pa'(\underline{y}) = \xi)$ are not identifiable. This is clear because if $\widetilde{x} \notin val(\underline{x})$, then we have no information of the type

$$P(\underline{y} = y|\widetilde{x}, pa'(\underline{y}) = \xi) = \quad P(\underline{y}(\widetilde{x}) = y|pa'(\underline{y}) = \xi) \ . \tag{64.8}$$

4. **Positivity**

Define the following 2 propensities for $x \in val(\underline{x})$ and $\widetilde{x} \in val(\underline{\widetilde{x}})$:

$$g_{x|c} = P(\underline{x} = x|c) \tag{64.9}$$

$$\widetilde{g}_{\widetilde{x}|c} = P(\underline{\widetilde{x}} = \widetilde{x}|c) \tag{64.10}$$

Positivity for $G_{im2,mod}$ is the requirement that

$$0 < g_{\widetilde{x}|c} < 1 \text{ for all } \widetilde{x} \in val(\underline{\widetilde{x}}) \text{ and } c \in val(\underline{c}) \tag{64.11}$$

If $g_{\widetilde{x}|c}$ is deterministic, then this requirement is not satisfied, and we cannot do IPW (i.e., inverse propensity weighing).

Note that using a MTP can allow IPW to be performed in cases when the propensity $g_{x|c}$ is anomalous (i.e., is either not defined or violates positivity) for some $x \in val(\underline{x})$, but is not anomalous for all $\widetilde{x} \in val(\underline{\widetilde{x}})$, where $val(\underline{\widetilde{x}})$ is some proper subset of $val(\underline{x})$.

The probability distribution $P(\widetilde{x}|x,c)$ (i.e., the $\underline{\widetilde{x}}$ prior for $G_{im2,mod}$) is called the **MTP**. An MTP can be either deterministic or probabilistic. An MTP that depends (resp., does not depend) on x is said to be **dynamic** (resp., **static**), because $\widetilde{x}$ depends (resp., does not depend) on the previous value of x.

1. Deterministic MTP

   Suppose $val(\underline{\widetilde{x}}) \subset val(\underline{x})$ and we are given a function $\widetilde{x}_c(\cdot) : val(\underline{x}) \to val(\underline{\widetilde{x}})$. Then let the TPM, printed in blue, for node $\underline{\widetilde{x}}$, be as follows:

$$P(\underline{\widetilde{x}} = \widetilde{x}|\underline{x} = x, c) = \delta(\widetilde{x}, \widetilde{x}_c(x)) \tag{64.12}$$

Figure 64.3: Two possible maps $\widetilde{x}_c : [a,b] \to [a,b]$ for a deterministic MTP.

   Examples of $\widetilde{x}_c(x)$:

   - threshold in the value of x for small values of x. (See Fig.64.3)
     Let $val(\underline{x}) = [a,b]$ where $a < b$. For some $\theta \in [a,b]$ and $u(c) \in [a,b]$, set

$$\widetilde{x}_c(x) = \begin{cases} \theta & \text{if } x \leq u(c) \\ x & \text{if } x > u(c) \end{cases} \tag{64.13}$$

   - upshift in the value of x for small values of x. (See Fig.64.3)
     Let $val(\underline{x}) = [a,b]$ where $a < b$. For some $\Delta x > 0$, and $u(c) \in [a,b]$, set

$$\widetilde{x}_c(x) = \begin{cases} (x + \Delta x) & \text{if } x < u(c) \text{ and } (x + \Delta x) \in [a,b] \\ x & \text{if } x > u(c) \\ b & \text{if } (x + \Delta x) > b \end{cases} \tag{64.14}$$

2. Stochastic MTP

   For some convenient, user specified, probability distribution $P(\widetilde{x}|x',x,c)$, where $\widetilde{x} \in val(\underline{\widetilde{x}})$, $x, x' \in val(\underline{x})$, and $c \in val(\underline{c})$, let the TPM, printed in blue, for node $\underline{\widetilde{x}}$, be as follows:

$$P(\widetilde{x}|x,c) = \sum_{x' \in val(\underline{x})} P(\widetilde{x}|x',x,c) \quad \underbrace{P(\underline{x}=x'|c)}_{\text{unmodified propensity}} \tag{64.15}$$

Examples

- Suppose we are given a function $x_c : val(\underline{\widetilde{x}}) \to val(\underline{x})$. ($x_c(\widetilde{x})$ could be the inverse, if it exists, of the function $\widetilde{x}_c(x)$ defined in the deterministic TPM case.)

$$P(\widetilde{x}|x,c) = \frac{P(\underline{x}=x_c(\widetilde{x})|c)}{\sum_{\widetilde{x}} numerator} \tag{64.16}$$

- Suppose $val(\underline{x}) = val(\underline{\widetilde{x}}) = \{0,1\}$, $\theta \in \mathbb{R}$, and let

$$P(\widetilde{x}|x,c) = [\pi(x|c)]^{\widetilde{x}}[1-\pi(x|c)]^{1-\widetilde{x}} \tag{64.17}$$

where

$$\pi(x|c) = \frac{\theta^x P(\underline{x}=x|c)}{\sum_x numerator} \tag{64.18}$$

64.2 $\Delta_{|c}$ estimand

Let

$$\mathcal{Y}_{|x,c} = \sum_y yP(\underline{y}(x)=y \mid x,c) \tag{64.19}$$

$$= \sum_y yP(\underline{y}=y \mid x,c) \tag{64.20}$$

$$\widetilde{\mathcal{Y}}_{|\underline{\widetilde{x}}=\widetilde{x},c} = \sum_y yP(\underline{y}(\widetilde{x})=y \mid \widetilde{x},c) \tag{64.21}$$

$$= \sum_y yP(\underline{y}=y \mid \widetilde{x},c) \tag{64.22}$$

$$\widetilde{\mathcal{Y}}_{|\underline{x}=x,c} = \sum_{\widetilde{x}} \widetilde{\mathcal{Y}}_{|\underline{\widetilde{x}}=\widetilde{x},c} P(\widetilde{x}|x,c) \tag{64.23}$$

$$\mathcal{Y}_{|c} = \sum_x \mathcal{Y}_{|x,c} P(x|c), \quad \mathcal{Y} = \sum_c P(c)\mathcal{Y}_{|c} \tag{64.24}$$

$$\widetilde{\mathcal{Y}}_{|c} = \sum_x \widetilde{\mathcal{Y}}_{|\underline{x}=x,c} P(x|c), \quad \widetilde{\mathcal{Y}} = \sum_c P(c)\widetilde{\mathcal{Y}}_{|c} \tag{64.25}$$

Note that there are two $\widetilde{\mathcal{Y}}$, namely $\widetilde{\mathcal{Y}}_{|\underline{x}=x,c}$ and $\widetilde{\mathcal{Y}}_{|\underline{\widetilde{x}}=\widetilde{x},c}$.
Define the $\Delta_{|c}$ estimand by:

$$\Delta_{|c} = \mathcal{Y}_{|c} - \widetilde{\mathcal{Y}}_{|c}, \quad \Delta = \sum_c P(c)\Delta_{|c} \tag{64.26}$$

$\Delta_{|c}$ measures the difference between the real world represented by $\mathcal{Y}_{|c}$ and a modified world represented by $\widetilde{\mathcal{Y}}_{|c}$.

Claim 90 *Consider the deterministic MTP case $P(\widetilde{x}|x,c) = \delta(\widetilde{x}, \widetilde{x}_c(x))$, where the function $\widetilde{x} = \widetilde{x}_c(x)$ is invertible with inverse $x = x_c(\widetilde{x})$ for some $x \in \mathbb{X} \subset \mathbb{R}$. Then,*

$$\widetilde{\mathcal{Y}}_{|\underline{x}=x,c} = \mathcal{Y}_{|x_c(\widetilde{x}),c} \tag{64.27}$$

proof:

$$\begin{aligned}
\widetilde{\mathcal{Y}}_{|\underline{x}=x,c} &= \sum_{\widetilde{x}} \sum_y yP(\underline{y} = y \mid \underline{\widetilde{x}} = \widetilde{x}, c)\delta(\widetilde{x}, \widetilde{x}_c(x)) && (64.28)\\
&= \sum_y yP(\underline{y} = y \mid \underline{\widetilde{x}} = \widetilde{x}_c(x), c) && (64.29)\\
&= \sum_y yP(\underline{y} = y \mid \underline{x} = x_c(\widetilde{x}), c) && (64.30)\\
&= \mathcal{Y}_{|x_c(\widetilde{x}),c} && (64.31)
\end{aligned}$$

QED

Henceforth in this chapter, we will restrict our attention to the deterministic MTP case in which $P(\widetilde{x}|x,c) = \delta(\widetilde{x}, \widetilde{x}_c(x))$. We will also assume that the domain $\mathbb{X}_c$ of $\widetilde{x}_c(x)$ is a union of disjoint sets $\mathbb{X}_c^j$ for $j = 1, 2, \ldots, nj(c)$, and that on each set $\mathbb{X}_c^j$, $\widetilde{x}_c(x)$ is invertible and differentiable.

$$\begin{cases} \mathbb{X} = val(\underline{x}) \\ \mathbb{X}_c^j \cap \mathbb{X}_c^{j'} = \emptyset \text{ for } j \neq j' \\ \mathbb{X}_c = \cup_{j=1}^{nj(c)} \mathbb{X}_c^j \subset \mathbb{X} \end{cases} \tag{64.32}$$

$$\widetilde{x}_c(x) = \sum_{j=1}^{nj(c)} \mathbb{1}(x \in \mathbb{X}_c^j)\widetilde{x}_c^j(x) \tag{64.33}$$

$$P(\widetilde{x}|x,c) = \delta(\widetilde{x}, \widetilde{x}_c(x)) = \sum_j \mathbb{1}(x \in \mathbb{X}_c^j)\delta(\widetilde{x}, \widetilde{x}_c^j(x)) \tag{64.34}$$

$$(\widetilde{x}_c^j)^{-1}(\widetilde{x}) = x^j(\widetilde{x}) \tag{64.35}$$

$$\widetilde{\mathbb{X}}_c^j = \widetilde{x}_c^j(\mathbb{X}_c^j) = \{\widetilde{x}_c^j(x) : x \in \mathbb{X}_c^j\} \tag{64.36}$$

$$P_{\mathbb{X}_c} = P(\underline{x} \in \mathbb{X}_c) \tag{64.37}$$

Claim 91

$$\widetilde{\mathcal{Y}}_{|c} = \frac{1}{P_{\mathbb{X}_c}} E_{\underline{\widetilde{x}}|c}[\lambda_c(\widetilde{x})\mathcal{Y}_{|x_c(\widetilde{x}),c}] \ , \tag{64.38}$$

$$\mathcal{Y}_{|c} \ = \ \frac{1}{P_{\mathbb{X}_c}} E_{\underline{\widetilde{x}}|c}[\lambda_c(\widetilde{x})\mathcal{Y}_{|\widetilde{x},c}] \tag{64.39}$$

and

$$P_{\mathbb{X}_c} = E_{\underline{\widetilde{x}}|c}[\lambda_c(\widetilde{x})] \tag{64.40}$$

where[3]

$$\lambda_c(\widetilde{x}) = \sum_j \mathbb{1}(\widetilde{x} \in \widetilde{\mathbb{X}}_c^j) \frac{dx_c^j(\widetilde{x})}{d\widetilde{x}} \underbrace{\frac{P(\underline{x} = x_c^j(\widetilde{x})|c)}{P(\underline{\widetilde{x}} = \widetilde{x}|c)}}_{=1} \tag{64.41}$$

proof:

$$\widetilde{\mathcal{Y}}_{|c} = \frac{1}{P_{\mathbb{X}_c}} \int_{x \in \mathbb{X}_c} dx \ P(x|c) \sum_y y P(y|\underline{\widetilde{x}} = \widetilde{x}_c(x), c) \tag{64.42}$$

$$= \frac{1}{P_{\mathbb{X}_c}} \sum_j \int_{x \in \mathbb{X}_c^j} dx \ P(x|c) \sum_y y P(y|\underline{\widetilde{x}} = \widetilde{x}_c^j(x), c) \tag{64.43}$$

$$= \frac{1}{P_{\mathbb{X}_c}} \sum_j \int_{\widetilde{x} \in \widetilde{\mathbb{X}}_c^j} d\widetilde{x} \frac{dx_c^j(\widetilde{x})}{d\widetilde{x}} P(\widetilde{x}|c) \frac{P(\underline{x} = x_c^j(\widetilde{x})|c)}{P(\widetilde{x}|c)} \underbrace{\sum_y y P(y|\underline{x} = x_c^j(\widetilde{x}), c)}_{\mathcal{Y}_{|x_c^j(\widetilde{x}),c}} \tag{64.44}$$

$$= \frac{1}{P_{\mathbb{X}_c}} \int_{\widetilde{x} \in \widetilde{\mathbb{X}}_c} d\widetilde{x} \ P(\widetilde{x}|c) \lambda_c(\widetilde{x}) \mathcal{Y}_{|x_c(\widetilde{x}),c} \quad \text{(see Eq(64.41))} \tag{64.45}$$

$$= \frac{1}{P_{\mathbb{X}_c}} E_{\underline{\widetilde{x}}|c}[\lambda_c(\widetilde{x})\mathcal{Y}_{|x_c(\widetilde{x}),c}] \tag{64.46}$$

To get Eq.(64.39), replace $\mathcal{Y}_{|x_c(\widetilde{x}),c}$ by $\mathcal{Y}_{\widetilde{x},c}$ in Eq.(64.38).
To get Eq.(64.40), replace $\mathcal{Y}_{|x_c(\widetilde{x}),c}$ by 1 in Eq.(64.38).

QED

From the last claim, it follows that

$$\Delta_{|c} = \frac{E_{\underline{y},\underline{\widetilde{x}}|c}\left[\lambda_c(\widetilde{x})(y - \mathcal{Y}_{|x_c(\widetilde{x}),c})\right]}{E_{\underline{\widetilde{x}}|c}[\lambda_c(\widetilde{x})]} \tag{64.47}$$

[3] $\lambda_c(\widetilde{x})$ is just a piecewise Jacobian. In Ref.[27], the part of Eq.(64.41) which is marked as being equal to 1 is incorrectly assumed to be different from 1.

64.3 Estimates of $\Delta_{|c}$

64.3.1 Empirical estimate of $\Delta_{|c}$

Consider a population Σ of individuals $\sigma \in \Sigma$, with $|\Sigma| = N$.

Let

$$N_c = \sum_{\sigma} \delta(c_\sigma, c) \ , \tag{64.48}$$

$$P(\sigma|c) = \frac{\delta(c_\sigma, c)}{N_c} \ , \tag{64.49}$$

and

$$E_{\sigma|c}[\xi_\sigma] = \sum_{\sigma} P(\sigma|c)\xi_\sigma \ . \tag{64.50}$$

Then set

$$\widehat{\Delta}_{|c} = \frac{E_{\sigma|c}\left[\lambda_c(\widetilde{x}_\sigma)(y_\sigma - \mathcal{Y}_{|x_c(\widetilde{x}_\sigma),c})\right]}{E_{\sigma|c}[\lambda_c(\widetilde{x}_\sigma)]} \tag{64.51}$$

64.3.2 OR estimate of $\Delta_{|c}$

Outcome Regression (OR) estimate of $\Delta_{|c}$.

Calculate the following estimates in this order: $\widehat{\tau} \to \widehat{\Delta}_{|c}$.

Steps:

1. Calculate $\widehat{\tau}$

   Let

   $$X^T = [1, \widetilde{x}, c], \quad X_\sigma^T = [1, \widetilde{x}_\sigma, c_\sigma] \ . \tag{64.52}$$

   Use Generalized Linear Modeling (GLM)[4] to approximate y_σ:

   $$y_\sigma \approx \widehat{y}(X_\sigma^T \widehat{\tau}) \tag{64.53}$$

   $$\widehat{y}(X_\sigma^T \widehat{\tau}) = g^{-1}(X_\sigma^T \widehat{\tau}) \tag{64.54}$$

   where $g()$ is the link function.

2. Calculate $\widehat{\Delta}_{|c}$

   $$\widehat{\Delta}_{|c} = \frac{E_{\sigma|c}[\lambda_c(\widetilde{x}_\sigma)(y_\sigma - \widehat{y}(X_\sigma^T \widehat{\tau}))]}{E_{\sigma|c}[\lambda_c(\widetilde{x}_\sigma)]} \tag{64.55}$$

[4] GLM is discussed in Chapter 35.

Claim 92 *(Asymptotic behavior of OR estimate)*
As $N \to \infty$,

$$\sqrt{N_c}(\widehat{\Delta}_{|c} - \Delta^*_{|c}) \to \mathcal{N}(0, \mathcal{V}) \tag{64.56}$$

where

$$\mathcal{V} = E_{\sigma|c}[(IF_\sigma)^2] \tag{64.57}$$

$$IF_\sigma = \frac{R_\sigma(\widehat{\tau}, \widehat{\Delta}_{|c}) + B(\widehat{\tau})}{C} \tag{64.58}$$

$$R_\sigma(\tau, \Delta_{|c}) = \lambda_c(\widetilde{x}_\sigma)(y_\sigma - \widehat{y}(X_\sigma^T \tau) - \Delta_{|c}) \tag{64.59}$$

$$C = -E_{\sigma|c}[\lambda_c(\widetilde{x}_\sigma)] \tag{64.60}$$

$$B(\tau) = \sum_i E_{\sigma|c}\left[\lambda_c(\widetilde{x}_\sigma)\frac{\partial \widehat{y}(X_\sigma^T \tau)}{\partial \tau_i}\right] \frac{E_{\sigma|c}[S_{i,\sigma}(\tau)]}{E_{\sigma|c}\left[\frac{\partial S_{i,\sigma}(\tau)}{\partial \tau_i}\right]} \tag{64.61}$$

(Note that index i labels the components of the column vector τ)

proof:

Eq.(64.55) that defines the OR estimate can be rewritten as

$$0 = E_{\sigma|c}[\underbrace{\lambda_c(\widetilde{x}_\sigma)(y_\sigma - \widehat{y}(X_\sigma^T \widehat{\tau}) - \widehat{\Delta}_{|c})}_{R_\sigma(\widehat{\tau}, \widehat{\Delta}_{|c})}] \tag{64.62}$$

If we Taylor expand $R_\sigma(\widehat{\tau}, \widehat{\Delta}_{|c})$ to first order in each of its 2 estimator arguments, we get

$$E_{\sigma|c}[R_\sigma(\widehat{\tau}, \widehat{\Delta}_{|c})] = \begin{cases} \underbrace{E_{\sigma|c}[R_\sigma(\tau^*, \Delta^*_{|c})]}_{A} \\ + \sum_i \underbrace{E_{\sigma|c}\left[\frac{\partial R_\sigma(\tau^*, \Delta^*_{|c})}{\partial \tau_i^*}\right]}_{B_i} (\widehat{\tau}_i - \tau_i^*) \\ + \underbrace{E_{\sigma|c}\left[\frac{\partial R_\sigma(\tau^*, \Delta^*_{|c})}{\partial \Delta^*_{|c}}\right]}_{-C} (\widehat{\Delta}_{|c} - \Delta^*_{|c}) \end{cases} \tag{64.63}$$

Solving the last equation for $\widehat{\Delta}_{|c} - \Delta^*_{|c}$ yields

$$\widehat{\Delta}_{|c} - \Delta^*_{|c} = \frac{A + \overbrace{\sum_i B_i(\widehat{\tau}_i - \tau_i^*)}^{B}}{C} \tag{64.64}$$

Assuming $y_\sigma \in \{0,1\}$, define the Cross Entropy and its first derivative with respect to τ_i by

$$CE_\sigma = \sum_{y_\sigma=0,1} y_\sigma \ln \widehat{y}(X_\sigma^T \tau) \tag{64.65}$$

$$S_{i,\sigma}(\tau) = \frac{\partial CE_\sigma}{\partial \tau_i} \tag{64.66}$$

In this notation, according to the maximum likelihood principle, the best choice of parameters τ_i must satisfy

$$0 = E_{\sigma|c}[S_{i,\sigma}(\widehat{\tau})] \tag{64.67}$$

If we Taylor expand $S_{i,\sigma}(\widehat{\tau})$ to first order in its estimator argument, we get

$$E_{\sigma|c}[S_{i,\sigma}(\widehat{\tau})] = E_{\sigma|c}[S_{i,\sigma}(\tau^*)] + E_{\sigma|c}\left[\frac{\partial S_{i,\sigma}(\tau^*)}{\partial \tau_i}\right](\widehat{\tau}_i - \tau_i^*) \tag{64.68}$$

Hence,

$$\widehat{\tau}_i - \tau_i^* = -\frac{E_{\sigma|c}[S_{i,\sigma}(\tau^*)]}{E_{\sigma|c}\left[\frac{\partial S_{i,\sigma}(\tau^*)}{\partial \tau_i}\right]} \ . \tag{64.69}$$

Now putting our previous results together, we get

$$\begin{aligned} A &= E_{\sigma|c}[R_\sigma(\tau^*, \Delta_{|c}^*)] && (64.70)\\ &= E_{\sigma|c}[\lambda_c(\widetilde{x}_\sigma)(y_\sigma - \widehat{y}(X_\sigma^T \tau^*) - \Delta_{|c}^*)] && (64.71) \end{aligned}$$

$$B_i = -E_{\sigma|c}\left[\lambda_c(\widetilde{x}_\sigma)\frac{\partial \widehat{y}(X_\sigma^T \tau^*)}{\partial \tau_i^*}\right] \tag{64.72}$$

$$\begin{aligned} B(\tau^*) &= \sum_i B_i(\widehat{\tau}_i - \tau_i^*) && (64.73)\\ &= \sum_i E_{\sigma|c}\left[\lambda_c(\widetilde{x}_\sigma)\frac{\partial \widehat{y}(X_\sigma^T \tau^*)}{\partial \tau_i^*}\right]\frac{E_{\sigma|c}[S_{i,\sigma}(\tau^*)]}{E_{\sigma|c}\left[\frac{\partial S_{i,\sigma}(\tau^*)}{\partial \tau_i}\right]} && (64.74) \end{aligned}$$

$$C = -E_{\sigma|c}[\lambda_c(\widetilde{x}_\sigma)] \tag{64.75}$$

$$IF_\sigma = \frac{R_\sigma(\widehat{\tau}, \widehat{\Delta}_{|c}) + B(\widehat{\tau})}{C} \tag{64.76}$$

$$\mathcal{V} = E_{\sigma|c}[(IF_\sigma)^2] \tag{64.77}$$

QED

64.4 Other Estimands besides $\Delta_{|c}$

Suppose $\widetilde{x}_0, \widetilde{x}_1 \in val(\underline{\widetilde{x}})$. Define the **Modified ATE (MATE)** by

$$MATE_{|c} = \widetilde{\mathcal{Y}}_{|\underline{\widetilde{x}}=\widetilde{x}_1,c} - \widetilde{\mathcal{Y}}_{|\underline{\widetilde{x}}=\widetilde{x}_0,c} \tag{64.78}$$

$$MATE = \sum_c P(c) MATE_{|c} \tag{64.79}$$

64.5 Multi-time MTP

A g-formula is any formula that defines recursively the full probability distribution of a bnet. In other words, it's a recursive definition of a Dynamical Bayesian Network.[5]

Let's define a g-formula[6] for Multi-time (a.k.a. longitudinal) MTP (LMTP).

Consider $t = 1, 2, \ldots, nt$,

For $nt = 1$,

$$\begin{tikzcd}
\underline{c}_1 \arrow[d] \arrow[dr] & \\
\underline{x}_1 \arrow[r] & \underline{y}
\end{tikzcd}
\qquad
\begin{tikzcd}
\underline{c}_1 \arrow[d] \arrow[ddr, bend left] \arrow[ddrr, bend left] & \\
\underline{x}_1 \arrow[d] & \\
\underline{\widetilde{x}}_1 \arrow[r] & \underline{y}
\end{tikzcd} \tag{64.80}$$

$$P(y, x_1, c_1) = P(y|x_1, c_1)P(x_1|c_1)P(c_1) \tag{64.81}$$

$$P(y, \widetilde{x}_1, x_1, c_1) = P(y|\widetilde{x}_1, c_1)P(x_1|c_1)P(\widetilde{x}_1|x_1, c_1)P(c_1) \tag{64.82}$$

For $nt = 2$,

$$\begin{tikzcd}
\underline{c}_1 \arrow[r] \arrow[d] \arrow[dr] & \underline{c}_2 \arrow[d] \arrow[dr] & \\
\underline{x}_1 \arrow[r] \arrow[ur] & \underline{x}_2 \arrow[r] & \underline{y}
\end{tikzcd}
\qquad
\begin{tikzcd}
\underline{c}_1 \arrow[r] \arrow[d] \arrow[dr] \arrow[ddr, bend left] \arrow[ddrr, bend left] & \underline{c}_2 \arrow[d] \arrow[dr] \arrow[ddr, bend left] & \\
\underline{x}_1 \arrow[d] \arrow[ur] & \underline{x}_2 \arrow[d] & \\
\underline{\widetilde{x}}_1 \arrow[r] & \underline{\widetilde{x}}_2 \arrow[r] & \underline{y}
\end{tikzcd} \tag{64.83}$$

$$P(y, x_{\leq 2}, c_{\leq 2}) = \begin{cases} P(x_1|c_1)P(c_1) \\ *P(x_2|x_1, c_{2,1})P(c_2|x_1, c_1) \\ *P(y|x_2, c_2) \end{cases} \tag{64.84}$$

[5]Dynamical Bayesian Networks are discussed in Chapter 26.

[6]Chapter 33 defines a different g-formula that we call a sequential backdoor g-formula.

$$P(y,\widetilde{x}_{\leq 2},x_{\leq 2},c_{\leq 2})=\left\{\begin{array}{l} P(x_1|c_1)P(\widetilde{x}_1|x_1,c_1)P(c_1) \\ *P(x_2|\widetilde{x}_1,c_{2,1})P(\widetilde{x}_2|x_2,c_{2,1})P(c_2|\widetilde{x}_1,c_1) \\ *P(y|\widetilde{x}_2,c_2) \end{array}\right. \tag{64.85}$$

For $nt=3$,

$$\begin{tikzcd}
\underline{c}_1 \arrow[r] \arrow[d] \arrow[dr] & \underline{c}_2 \arrow[r] \arrow[d] \arrow[dr] & \underline{c}_3 \arrow[d] \arrow[dr] & \\
\underline{x}_1 \arrow[r] \arrow[ur] & \underline{x}_2 \arrow[r] \arrow[ur] & \underline{x}_3 \arrow[r] & \underline{y}
\end{tikzcd}
\qquad
\begin{tikzcd}
\underline{c}_1 \arrow[r] \arrow[d] \arrow[dr] \arrow[ddr, bend left] & \underline{c}_2 \arrow[r] \arrow[d] \arrow[dr] \arrow[ddr, bend left] & \underline{c}_3 \arrow[d] \arrow[dr] \arrow[ddr, bend left] & \\
\underline{x}_1 \arrow[d] \arrow[ur] \arrow[dr] & \underline{x}_2 \arrow[d] \arrow[ur] \arrow[dr] & \underline{x}_3 \arrow[d] \arrow[ur] \arrow[dr] & \\
\widetilde{\underline{x}}_1 \arrow[r] \arrow[ur] & \widetilde{\underline{x}}_2 \arrow[r] \arrow[ur] & \widetilde{\underline{x}}_3 \arrow[r] & \underline{y}
\end{tikzcd} \tag{64.86}$$

$$P(y,x_{\leq 3},c_{\leq 3})=\left\{\begin{array}{l} P(x_1|c_1)P(c_1) \\ *P(x_2|x_1,c_{2,1})P(c_2|x_1,c_1) \\ *P(x_3|x_2,c_{3,2})P(c_3|x_2,c_2) \\ *P(y|x_3,c_3) \end{array}\right. \tag{64.87}$$

$$P(y,\widetilde{x}_{\leq 3},x_{\leq 3},c_{\leq 3})=\left\{\begin{array}{l} P(x_1|c_1)P(\widetilde{x}_1|x_1,c_1)P(c_1) \\ *P(x_2|\widetilde{x}_1,c_{2,1})P(\widetilde{x}_2|\widetilde{x}_1,x_{2,1},c_{2,1})P(c_2|\widetilde{x}_1,c_1) \\ *P(x_3|\widetilde{x}_2,c_{3,2})P(\widetilde{x}_3|\widetilde{x}_2,x_{3,2},c_{3,2})P(c_3|\widetilde{x}_2,c_2) \\ *P(y|\widetilde{x}_3,c_3) \end{array}\right. \tag{64.88}$$

In general,

$$P(y,x_{\leq nt},c_{\leq nt})=\left\{\begin{array}{l} \prod_{t=1}^{nt}\{P(x_t|x_{t-1},c_{t,t-1})P(c_t|x_{t-1},c_{t-1})\} \\ *P(y|x_{nt},c_{nt}) \end{array}\right. \tag{64.89}$$

$$P(y,\widetilde{x}_{\leq nt},x_{\leq nt},c_{\leq nt})=\left\{\begin{array}{l} \prod_{t=1}^{nt}P(x_t|\widetilde{x}_{t-1},c_{t,t-1})P(\widetilde{x}_t|\widetilde{x}_{t-1},x_{t,t-1},c_{t,t-1})P(c_t|\widetilde{x}_{t-1},c_{t-1}) \\ *P(y|\widetilde{x}_{nt},x_{nt},c_{nt}) \end{array}\right. \tag{64.90}$$

Note that Ref.[29] by Hernán and Robins (HR) uses the notation $L_t=\underline{c}_t$, $A_t=\underline{x}_t$ for $t=1,2,\ldots,nt$. Hence, the $nt=3$ bnet Eq.(64.86), written in the HR notation, is as follows (before and after adding the $\widetilde{A}$ nodes):

$$\begin{tikzcd}
L_1 \arrow[r] \arrow[d] \arrow[dr] & L_2 \arrow[r] \arrow[d] \arrow[dr] & L_3 \arrow[d] \arrow[dr] & \\
A_1 \arrow[r] \arrow[ur] & A_2 \arrow[r] \arrow[ur] & A_3 \arrow[r] & \underline{y}
\end{tikzcd}
\qquad
\begin{tikzcd}
L_1 \arrow[r] \arrow[d] \arrow[dr] \arrow[ddr, bend left] & L_2 \arrow[r] \arrow[d] \arrow[dr] \arrow[ddr, bend left] & L_3 \arrow[d] \arrow[dr] \arrow[ddr, bend left] & \\
A_1 \arrow[d] \arrow[ur] \arrow[dr] & A_2 \arrow[d] \arrow[ur] \arrow[dr] & A_3 \arrow[d] \arrow[ur] \arrow[dr] & \\
\widetilde{A}_1 \arrow[r] \arrow[ur] & \widetilde{A}_2 \arrow[r] \arrow[ur] & \widetilde{A}_3 \arrow[r] & \underline{y}
\end{tikzcd} \tag{64.91}$$

HR doesn't actually display the $\widetilde{A}$ nodes in its DAGs. Furthermore, in HR, nodes A_t and L_t have parent nodes that are time steps $0, 1, 2, 3, \ldots$ in the past. I, on the other hand, do not draw or consider parents that are more than one time steps in the past, because I assume the effect of those parents is negligible to first approximation.

Chapter 65

Monty Hall Problem

Figure 65.1: Monty Hall Problem.

Mr. Monty Hall, host of the game show "Let's Make a Deal", hides a car behind one of three doors and a goat behind each of the other two. The contestant picks Door No. 1, but before opening it, Mr. Hall opens Door No. 2 to reveal a goat. Should the contestant stick with No. 1 or switch to No. 3?

The Monty Hall problem can be modeled by the bnet Fig.65.1, where

- $\underline{c}$= the door behind which the car actually is.
- $\underline{y}$= the door opened by you (the contestant), on your first selection.
- $\underline{m}$= the door opened by Monty (game host)

We label the doors 1,2,3 so $val(\underline{c}) = val(\underline{y}) = val(\underline{m}) = \{1, 2, 3\}$.
The TPMs, printed in blue, for this bnet, are as follows:

$$P(c) = \frac{1}{3} \text{ for all } c \tag{65.1}$$

$$P(y) = \frac{1}{3} \text{ for all } y \tag{65.2}$$

$$P(m|c,y) = \mathbb{1}(m \neq c)\left[\frac{1}{2}\mathbb{1}(y = c) + \mathbb{1}(y \neq c)\mathbb{1}(m \neq y)\right] \tag{65.3}$$

It's easy to show that the above node probabilities imply that

$$P(c = 1|m = 2, y = 1) = \frac{1}{3} \tag{65.4}$$

$$P(c = 3|m = 2, y = 1) = \frac{2}{3} \tag{65.5}$$

So you are twice as likely to win if you switch your final selection to be the door which is neither your first choice nor Monty's choice.

The way I justify this to myself is: Monty gives you a piece of information. If you don't switch your choice, you are wasting that info, whereas if you switch, you are using the info.

Chapter 66

Multi-armed Bandits

Figure 66.1: Multi-armed bandit (MAB).

Figure 66.2: Bernoulli MAB (Bern-MAB) with 4 arms. For a Bern-MAB, the conditional probability distribution $P(r|a)$ for reward r, given arm a, is initially totally unknown to the gambler (agent), but it is known by the environment to be a Bernoulli distribution $P(r|a) = \mu_a^r(1-\mu_a)^{1-r}$ for $r \in \{0, 1\}$, where $0 < \mu_a < 1$ for each a. The percentages shown are μ_a for each arm a.

This chapter is mostly based on Refs. [68] and [102].

Multi-armed Bandits (MABs) are a simple version of Reinforcement Learning (RL). RL is discussed in Chapter 84.

The term "one-armed-bandit" is a humorous term for what is also called a slot machine. A slot machine is a gambling device which has a slot into which you put coins or tokens for the privilege of being allowed to pull down a lever (arm) on one side of the device. This action generates a random combination of three shapes, which may or may not, depending on their combination, entitle the player to a money award.

Multi-armed bandit (MAB) is the name given to the optimization problem that considers an agent (gambler) that is playing multiple one-armed-bandits, each with a possibly different odds of winning. The optimization problem is to determine an efficient schedule whereby the gambler can converge on the device with the highest odds of winning.

MABs are often used in marketing as an alternative to A/B testing. These 2 methods yield different information but overlap in that they both can discover consumer preferences.

The MAB problem is an optimization problem (i.e., finding the maximum of a reward function or the minimum of a cost function). As with any minimization problem, an algorithm to solve it runs the danger of converging to a local minimum that isn't the global (i.e., the overall) minimum. This danger can be diminished by doing both **exploration and exploitation**. Algorithms that do no exploration, only exploitation, are said to be **greedy**, and they are at the highest risk of converging to a non-global minimum.

66.1 Bnet for MAB

Figure 66.3: Bnet for a multi-armed bandit (MAB).

Let
$t \in \{0, 1, 2, \ldots\}$ be the **time** slice (step),

$\underline{a}_t \in \{0, 1, \ldots, N_{\underline{a}} - 1\} = val(\underline{a})$ be the **arm** of bandit that is pulled at time t, out of $N_{\underline{a}}$ arms. In RL language, it's also the **action** taken.

$\underline{r}_t \in \mathbb{R}$ be the **reward** at time t,

$s_{<t} = [\underbrace{(a_\tau, r_\tau)}_{s_\tau} : \tau < t]$ be the **state** at time t.

Fig.66.3 shows a bnet for a MAB. The TPMs, printed in blue, for this bnet, are as follows.

$$P(s_{<t}|s_{<t-1}, a_{t-1}, r_{t-1})) = \mathbb{1}(s_{<t} = s_{<t-1} \cup (a_{t-1}, r_{t-1})) \tag{66.1}$$

For $t = 0$, $s_{<0} = \emptyset$, so choose a random a_0.

$$P(a_t|s_{<t}) = \mathbb{1}(a_t = a_t^*) \ \text{ (agent's response)} \ , \tag{66.2}$$

where a_t^* depends on the **strategy (a.k.a. policy)** being used by the **gambler (agent)**. We consider various strategies below. a_t^* is defined solely in terms of empirical data (i.e., a_t, r_t values) collected from previous time-slices. That is the only type of information that the gambler is privy to. a_0^* is chosen at random. The formulae for a_t^* that we give below for various strategies should only be used for $t > 0$.

$$P(r_t|a_t) = P_{\underline{r}|\underline{a}}(r_t|a_t) \ \text{ (environment's response)} \ . \tag{66.3}$$

$P_{\underline{r}|\underline{a}}$ is a probabilistic model that models the **environment** in which the agent lives. This assumes that the a_t (and the r_t) are i.i.d. $P_{\underline{r}|\underline{a}}$ depends on parameters whose values are known by the environment, but are not known a priori by the gambler. In fact, the goal of this exercise is for the gambler to find ever more accurate estimates of those parameters, using only the empirical data he/she can collect from the past, starting from total ignorance about those parameters.

For a **Bernoulli MAB (Bern-MAB)**, the conditional probability distribution $P_{\underline{r}|\underline{a}}$ is a Bernoulli distribution

$$P(r|a) = \mu_a^r (1 - \mu_a)^{1-r} \tag{66.4}$$

for $r \in \{0, 1\}$, where $0 < \mu_a < 1$ for each a. The parameters μ_a are initially unknown to the gambler. Note that $E[\underline{r}|a] = \sum_r r P(r|a) = P(\underline{r} = 1|a) = \mu_a$.

For a **Gaussian MAB**, the conditional probability distribution $P_{\underline{r}|\underline{a}}$ is a Normal (Gaussian) distribution

$$P(r|a) = \mathcal{N}(r; \mu_a, \sigma_a^2) \ . \tag{66.5}$$

for $r \in \mathbb{R}$. The parameters μ_a, σ_a^2 are initially unknown to the gambler.

66.2 Reward functions

For $a \in val(\underline{a})$, define the **Long term Average Reward** for action a by

$$\mu_a = Q(a) = E_{|a}[\underline{r}] = \sum_r rP(r|a) \ . \tag{66.6}$$

Let

$$a^* = \operatorname*{argmax}_a Q(a) \tag{66.7}$$

$$\mu^* = \max_a Q(a) = Q(a^*) \tag{66.8}$$

$$\Delta_a = \mu^* - Q(a) \tag{66.9}$$

μ_a, a^*, μ^* and Δ_a are not known to the gambler.

Define the **Instantaneous (at time t) Average Reward** for action a by

$$Q_t(a) = \frac{1}{N_t(a)} \sum_{\tau=0}^{t} r_\tau \mathbb{1}(a_\tau = a) \tag{66.10a}$$

where

$$N_t(a) = N_{in} + \sum_{\tau=0}^{t} \mathbb{1}(a_\tau = a) \tag{66.10b}$$

$N_{in} > 0$ insures that we never divide by zero. Note that Eqs.(66.10) can be stated recursively as

$$N_t(a)Q_t(a) = N_{t-1}(a)Q_{t-1}(a) + r_t \mathbb{1}(a_t = a) \tag{66.11a}$$

$$N_t(a) = N_{t-1}(a) + \mathbb{1}(a_t = a) \tag{66.11b}$$

with $Q_{-1}(a) = 0$ and $N_{-1}(a) = N_{in}$ for all a. We assume that at large t, $N_{in} << N_t(a)$ for all a. For instance, one can use $N_{in} = 1$.

$Q(a)$ is not known to the gambler but $N_t(a)$ and $Q_t(a)$ are because they are empirical.

We will write a hat over random variables that are defined by an empirical probability distribution.[1] Whereas we assume that $\underline{a}_t$ and $\underline{r}_t$ are i.i.d., we will not assume that $\widehat{a}_t$ and $\widehat{r}_t$ are i.i.d. for finite times t. What we will assume is that as $t \to \infty$,

$$\underline{\widehat{a}}_t \to \underline{a} \ , \ \underline{\widehat{r}}_t \to \underline{r} \tag{66.12}$$

[1] An alternative convention is to not distinguish between $\underline{\widehat{a}}_t$ and $\underline{a}_t$, or between $\underline{\widehat{r}}_t$ and $\underline{r}_t$, but to distinguish between $P(r_t|a_t)$ and $\widehat{P}(r_t|a_t)$, where $\widehat{P}()$ is an empirical estimate of $P()$.

Note that

$$\sum_a \frac{N_t(a)}{t+1} = 1 \tag{66.13}$$

so define

$$P(\underline{\widehat{a}}_t = a) = \frac{N_t(a)}{t+1} \ . \tag{66.14}$$

Thus

$$E[Q(\underline{\widehat{a}}_t)] = \sum_a P(\underline{\widehat{a}}_t = a) Q(a) \ . \tag{66.15}$$

Claim 93

$$Q_t(a) = E_{|\widehat{a}_t = a}[\underline{\widehat{r}}_t] \tag{66.16}$$

proof:

Note that

$$\sum_r \sum_{\tau=0}^{t} \frac{\mathbb{1}(a_\tau = a, r_\tau = r)}{N_t(a)} = 1 \ . \tag{66.17}$$

so define

$$P(\underline{\widehat{r}}_t = r | \underline{\widehat{a}}_t = a) = \sum_{\tau=0}^{t} \frac{\mathbb{1}(a_\tau = a, r_\tau = r)}{N_t(a)} \ . \tag{66.18}$$

Thus

$$\begin{aligned} Q_t(a) &= \sum_r r \sum_{\tau=0}^{t} \frac{\mathbb{1}(a_\tau = a, r_\tau = r)}{N_t(a)} && (66.19) \\ &= \sum_r r P(\underline{\widehat{r}}_t = r | \underline{\widehat{a}}_t = a) && (66.20) \\ &= E_{|\widehat{a}_t = a}[\underline{\widehat{r}}_t] && (66.21) \end{aligned}$$

QED

Claim 94 *As $t \to \infty$,*

$$Q_t(a) \to E_{\underline{a}=a}[\underline{r}] = Q(a) \tag{66.22}$$

$$E[Q(\underline{\widehat{a}}_t)] \to E[Q(\underline{a})] \tag{66.23}$$

proof: This is clear from $\widehat{a}_t \to a$ and $\widehat{r}_t \to r$ and Claim 93.
QED

66.3 Regret functions

Define the **Instantaneous (at time t) Average Regret (I-Regret)** by

$$IReg_t = \mu^* - E[Q(\underline{\widehat{a}}_t)] = E[\Delta_{\underline{\widehat{a}}_t}] \tag{66.24}$$

and the **Cumulative (for times $\leq t$) Average Regret (C-Regret)** by

$$CReg_t = \sum_{\tau=0}^{t} IReg_\tau \ . \tag{66.25}$$

Note that

$$\begin{aligned} CReg_t &= \sum_{\tau=0}^{t} E[\Delta_{\underline{\widehat{a}}_\tau}] && (66.26) \\ &= \sum_a \Delta_a \sum_{\tau=0}^{t} \frac{N_\tau(a)}{\tau+1} && (66.27) \end{aligned}$$

Let

$$\rho_t = \frac{1}{t+1} \sum_{t=0}^{t} r_t \ . \tag{66.28}$$

Define the **Cumulative Average Reward (C-Reward)** by

$$CRew_t = E_{|\underline{\widehat{a}}_t = a}[(t+1)\underline{\widehat{\rho}}_t] \ . \tag{66.29}$$

It can be shown that minimizing the C-Regret is equivalent to maximizing the C-Reward.

Claim 95 *As $t \to \infty$,*

$$IReg_t \to E[\Delta_{\underline{a}}] \tag{66.30}$$

proof: This is clear from $\widehat{a}_t \to a$ and $\widehat{r}_t \to r$.
QED

66.4 Strategies with random exploration

In this section, we consider MAB algorithms that explore all values of the action a at random. In the section following this one, we consider MAB algorithm that do a more deliberate search of the action space.

Figure 66.4: Extra structure added to MAB Bnet Fig.66.3 for ϵ-greedy algorithm.

66.4.1 ϵ-greedy algorithm

Recall that $\underline{a}_t \in \{0, 1, \ldots, N_{\underline{a}} - 1\} = val(\underline{a})$. The user of the algorithm specifies an $\epsilon \in [0, 1]$ which measures the amount of exploration to be conducted.

For each t, add extra the structure shown in Fig.66.4 to the MAB bnet Fig.66.3. $\underline{g}_t \in \{0, 1\}$ and $\underline{a}_t, \underline{A}_t \in val(\underline{a})$. ("g" stands for greedy). The TPMs, printed in blue, for the new nodes, are as follows

$$P(g_t) = \begin{cases} \epsilon \ \text{(exploration)} & \text{if } g_t = 0 \\ 1 - \epsilon \ \text{(exploitation)} & \text{if } g_t = 1 \end{cases} . \tag{66.31}$$

$$P(A_t|g_t) = \begin{cases} \frac{1}{N_{\underline{a}}} \ \text{(exploration)} & \text{if } g_t = 0 \\ \mathbb{1}(A_t = 0) \ \text{(exploitation)} & \text{if } g_t = 1 \end{cases} \tag{66.32}$$

$$P(a_t|A_t, g_t) = \begin{cases} \mathbb{1}(a_t = A_t) \ \text{(exploration)} & \text{if } g_t = 0 \\ \mathbb{1}(a_t = a_t^*) \ \text{(exploitation)} & \text{if } g_t = 1 \end{cases} \tag{66.33}$$

where a_t^* is defined as follows:

$$\boxed{a_t^* = \operatorname*{argmax}_a Q_{t-1}(a)} . \tag{66.34}$$

As $t \to \infty$,

$$\begin{aligned} \frac{N_t(a)}{t+1} &= P(\hat{\underline{a}}_t = a) && (66.35) \\ &\to P(a, g_t = 0)\epsilon + P(a, g_t = 1)(1 - \epsilon) && (66.36) \\ &\geq P(a, g_t = 0)\epsilon && (66.37) \\ &= \frac{\epsilon}{N_{\underline{a}}} && (66.38) \end{aligned}$$

Hence

$$IReg_t = \sum_a \Delta_a \sum_{\tau=0}^{t} \frac{N_\tau(a)}{\tau+1} \tag{66.39}$$

$$\geq \frac{\epsilon}{N_{\underline{a}}} \sum_a \Delta_a \tag{66.40}$$

and $CReg_t \geq \mathcal{N}(!t)(t+1)$.

66.4.2 ϵ_t-greedy algorithm

Replace time-independent constant ϵ in the ϵ-greedy algorithm by a time dependent function ϵ_t.

66.5 Strategies with nonrandom exploration

66.5.1 Upper Confidence Bounds (UCB) algorithms

A MAB algorithm that maximizes merely $Q_t(a)$ to get a_t^* is totally greedy (i.e., does no exploration, only exploitation). This doesn't work too well because once the algo finds a particular a_t^*, it sticks with it. For times t after that, the $Q_t(a)$ for all a stay more or less the same. The $N_t(a)$ for $a \neq a_t^*$ also stay the same. Only $N_t(a_t^*)$ increases. To avoid this problem, **Upper Confidence Bounds (UCB) algorithms** maximize an effective $Q_t(a)_{\text{eff}} = Q_t(a) + U_t(a)$, where $U_t(a) > 0$, instead of maximizing merely $Q_t(a)$ to get a_t^*. $U_t(a)$ is proportional to $1/\sqrt{N_t(a)}$ (that's a property of UCBs). Adding $U_t(a)$ to $Q_t(a)$ gives those a with low $N_a(a)$ a $Q_t(a)_{\text{eff}} >> Q_t(a)$. This encourages exploration of a different from $a_t^* = \operatorname{argmax} Q_t(a)$. In conclusion, for all UCB algorithms, we have

$$a_t^* = \operatorname*{argmax}_a [Q_{t-1}(a) + U_{t-1}(a)] \tag{66.41}$$

Different UCB algos differ only in the definition of $U_t(a)$.

Next, we consider two UCB algorithms, frequentist UCB (UCB1), and Bayesian UCB.

Frequentist UCB (UCB1) algorithm

Claim 96 *(Hoeffding's Inequality- HI) Let $\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{T-1}$ be i.i.d. random variables such that $0 \leq \underline{x}_t \leq 1$ for all t. Let $\underline{m}_{T-1} = \frac{1}{T}\sum_{t=0}^{T-1} \underline{x}_t$ denote the sample mean. Then for $u > 0$, we have:*

$$P\left(E[\underline{x}] > \underline{m}_{T-1} + u\right) \leq e^{-2Tu^2} . \tag{66.42}$$

proof: See Ref.[144].
QED

Let[2]

$$\underline{x}^T = \left\{ \underline{x}_t : t \in \mathbb{Z}_{[0,T-1]} \right\} \tag{66.43}$$

and

$$\underline{r}_{\leq t-1}(a) = \left\{ \underline{r}_\tau : \underline{a}_\tau = a, \tau \in \mathbb{Z}_{[0,t-1]} \right\} . \tag{66.44}$$

Note that $\underline{x}^T$ has T components and $\underline{r}_{\leq t-1}(a)$ has $N_{t-1}(a)$ components. If we apply the HI with $\underline{x}^T$ replaced by $\underline{r}_{\leq t-1}(a)$, we get

$$P\left(Q(a) > Q_{t-1}(a) + U_{t-1}(a)\right) \quad \leq \quad e^{-2N_{t-1}(a)[U_{t-1}(a)]^2} \tag{66.45}$$

If we define a threshold probability p by

$$p = e^{-2N_{t-1}(a)[U_{t-1}(a)]^2} , \tag{66.46}$$

then

$$U_{t-1}(a) = \sqrt{\frac{-\ln p}{2N_{t-1}(a)}} \tag{66.47}$$

If we choose $p = (t-1)^{-\alpha}$,

$$\boxed{a_t^* = \underset{a}{\operatorname{argmax}} \left[Q_{t-1}(a) + \sqrt{\frac{\alpha(t-1)}{2N_{t-1}(a)}} \right]} \tag{66.48}$$

Bayesian UCB algorithm

Claim 97 [3] [4] *(Bayesian updating of mean and deviation. See Fig.66.5.)*

Suppose

$$x^n = [x_i] , \quad x_i \text{ are i.i.d. with } \underline{x_i}|\mu, \tau \sim \mathcal{N}(\mu, \tau) \tag{66.49}$$

$$\underline{\mu}|\tau \sim \mathcal{N}(\mu_0, n_0\tau) \tag{66.50}$$

$$\underline{\tau} \sim \text{Gamma}(\alpha, \beta) . \tag{66.51}$$

Then the posterior is

$$\underline{\mu}|\tau, x^n \sim \mathcal{N}\left(\frac{n\overline{x} + n_0\mu_0}{n + n_0} \quad , \quad (n + n_0)\tau \right) \tag{66.52}$$

[2] As usual, we define $\mathbb{Z}_{[a,b]} = \{a, a+1, a+2, \ldots, b\}$ for $a < b$.

[3] For a **standard deviation** σ, the **precision** τ is defined as $\tau = \frac{1}{\sigma^2}$.

[4] $\underline{x}|\lambda \sim \mathcal{D}(\lambda)$ means that $P(x|\lambda) = \mathcal{D}(x;\lambda)$. $\mathcal{N}()$ stands for the Normal distribution and Gamma() for the Gamma distribution. See Ref.[138] for a discussion of the Gamma distribution.

$$\underline{\tau}|x^n \sim \text{Gamma}\left(\alpha + \frac{n}{2} \;\;,\;\; \beta + \frac{1}{2}\sum_i (x_i - \overline{x})^2 + \frac{nn_0}{2(n+n_0)}(\overline{x} - \mu_0)^2\right) \tag{66.53}$$

proof: See Ref.[35].
QED

Figure 66.5: Prior and Posterior Bnets in Claim 97.

In the Bayesian UCB algorithm, we use

$$\boxed{a_t^* = \underset{a}{\text{argmax}}\, E_{\mu_a,\sigma_a|s_{<t}}\left[\mu_a + c\frac{\sigma_a}{\sqrt{N_{t-1}(a)}}\right]} \tag{66.54}$$

for some $c > 0$.

Eq.66.54 requires that we know $P(\mu_a, \sigma_a|s_{<t})$; i.e., the posterior distribution of μ_a, σ_a assuming the prior history $s_{<t}$. This follows from Bayes theorem if we assume a Normal distribution for the likelihood $P(s_{<t}|\mu_a, \sigma_a)$ and we assume conjugate priors for $P(\mu_a, \sigma_a)$. More precisely, if we replace $\underline{x}^n$ by $\underline{s}_{<t}$ in Claim 97, then

$$P(\mu_a, \tau_a|s_{<t}) = P(\mu_a|\tau_a, s_{<t})P(\tau_a|s_{<t}) \tag{66.55}$$

where $P(\mu_a|\tau_a, s_{<t})$ and $P(\tau_a|s_{<t})$ are given by Claim 97.

66.5.2 Thompson Sampling MAB (TS-MAB) algorithm

Bnet for general TS-MAB algorithm

The Thompson Sampling MAB (TS-MAB) algorithm is described by the bnet Fig.66.6. This bnet differs from the bnet Fig.66.3 in that it includes new nodes $\underline{\lambda}_t$. The TPMs, printed in blue, for bnet Fig.66.6, are as follows.

$$P(s_{<t}|s_{<t-1}, a_{t-1}, r_{t-1}) = \text{ same as for Fig.66.3} \tag{66.56}$$

Figure 66.6: Bnet for TS-MAB algorithm.

$$P(\lambda_t|s_{<t},\lambda_{t-1}) = \mathbb{1}(\lambda_t = \lambda_t^*(s_{<t},\lambda_{t-1})) \tag{66.57}$$

where λ_t^* is a function to be defined below.

$$P(a_t|\lambda_t) = \mathbb{1}[a_t = a_t(\lambda_t)] \;\; \text{(Agent's response)} \tag{66.58}$$

where $a_t(\lambda_t)$ is a function to be described below.

$$P(r_t|a_t) = \;\text{same as for Fig.66.3. (Environment's response)} \tag{66.59}$$

Let

$$\begin{aligned} Q_t(a,\lambda_t) &= \sum_r r P(\underline{\widehat{r}}_t = r|\underline{\widehat{a}}_t = a,\lambda_t) && (66.60)\\ &= E_{|\underline{\widehat{a}}_t=a,\lambda_t}[\underline{\widehat{r}}_t]\;. && (66.61)\end{aligned}$$

Define

$$\boxed{a_t(\lambda_t) = \operatorname*{argmax}_a Q_t(a,\lambda_t)}\;. \tag{66.62}$$

Note that

$$\begin{aligned} P(a_t|s_{<t}) &= \sum_{\lambda_t} P(\lambda_t|s_{<t})\mathbb{1}[Q_t(a_t,\lambda_t) = \max_a Q_t(a,\lambda_t)] && (66.63a)\\ &= \sum_{\lambda_t} P(\lambda_t|s_{<t})\mathbb{1}[a_t = a_t(\lambda_t)] && (66.63b)\\ &= E_{\lambda_t|s_{<t}}\{\mathbb{1}[a_t = a_t(\lambda_t)]\}\;. && (66.63c)\end{aligned}$$

If we further assume that $P(\lambda_t|s_{<t})$ is a delta function, then Eqs.(66.63) reduce to

$$P(a_t|s_{<t}) = \mathbb{1}[a_t = a_t(\lambda_t^*)] \tag{66.64}$$

TS-MAB algorithm with Beta agent and Bernoulli environment

The Beta distribution $\text{Beta}(x;\alpha,\beta)$ (see Ref.[113]) is defined for $\alpha > 0, \beta > 0$ and $x \in [0,1]$. Since $x \in [0,1]$, x can be interpreted as a probability. The mean and variance of the Beta distribution are

$$E[\underline{x}] = \frac{\alpha}{\alpha+\beta} \ , \tag{66.65}$$

$$\langle \underline{x}, \underline{x} \rangle = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \ . \tag{66.66}$$

From this mean and variance, we see that if we increase α by one and leave β the same, the mean moves towards 1 and the variance decreases. Likewise, if we increase β by 1 and leave α the same, the mean moves towards 0 and the variance decreases.

Let

$$\alpha_t^a = \sum_{\tau=0}^{t} \mathbb{1}(r_\tau = 1, a_\tau = a) \tag{66.67}$$

$$\beta_t^a = \sum_{\tau=0}^{t} \mathbb{1}(r_\tau = 0, a_\tau = a) \tag{66.68}$$

$$\lambda_t^a = (\alpha_{t-1}^a, \beta_{t-1}^a) \tag{66.69}$$

$$\lambda_t = [(\alpha_{t-1}^a, \beta_{t-1}^a) : a \in val(\underline{a})] \tag{66.70}$$

$$Q_t(a, \lambda_t) = \sum_r r\text{Beta}(r; \alpha_{t-1}^a, \beta_{t-1}^a) \tag{66.71a}$$

$$= \frac{\alpha_{t-1}^a}{\alpha_{t-1}^a + \beta_{t-1}^a} \tag{66.71b}$$

The TS-MAB algorithm for a Beta agent and Bernoulli environment can be described by the bnet Fig.66.6. For this special case of bnet Fig.66.6, the TPMs, printed in blue, are as follows.

$$P(s_{<t}|s_{<t-1}, a_{t-1}, r_{t-1}) = \text{ same as for Fig.66.6} \tag{66.72}$$

$$P(\lambda_t^a|s_{<t},\lambda_{t-1}) = \mathbb{1}(\lambda_t^a = \lambda_t^{*a}) = \begin{cases} \mathbb{1}(\lambda_t^a = \lambda_{t-1}^a) & \text{if } a \neq a_{t-1} \\ \mathbb{1}(\lambda_t^a = (\alpha_{t-2}^a + 1, \beta_{t-2}^a)) & \text{if } a_{t-1} = a \text{ and } r_{t-1} = 1 \\ \mathbb{1}(\lambda_t^a = (\alpha_{t-2}^a, \beta_{t-2}^a + 1)) & \text{if } a_{t-1} = a \text{ and } r_{t-1} = 0 \end{cases} \tag{66.73}$$

$$P(\underline{a}_t|\lambda_t) = \mathbb{1}(a_t = a_t(\lambda_t)) \tag{66.74}$$

$$= \mathbb{1}(a_t = \underset{a}{\text{argmax}} \ \underbrace{Q_t(a,\lambda_t)}_{\text{see Eq.(66.71).}} \ ) \ \text{(Beta agent response)} \tag{66.75}$$

$$P(r_t|a_t) = \mu_{a_t}^{r_t}(1-\mu_{a_t})^{r_t} \ \text{(Bernoulli environment response)} \ . \tag{66.76}$$

μ_a known to environment but unknown to agent.

TS-MAB algorithm, skeletal reprise

The TS-MAB algorithm is not very complicated but explaining it with precision requires nightmarishly many indices. Here is a pedagogical reprise of what we have said so far, where we have stripped out some of the inessential indices.

$$P(\lambda) = \mathbb{1}(\lambda, \lambda^*) \tag{66.77}$$

$$\begin{aligned} P(a|\lambda) &= \mathbb{1}(a = a(\lambda)) && (66.78) \\ &= \mathbb{1}(a = \underset{a}{\text{argmax}} \sum_r r P(\widehat{r} = r|\widehat{a} = a, \lambda)) && (66.79) \\ &= \mathbb{1}(a = \underset{a}{\text{argmax}} \sum_r r \text{Beta}(r;\lambda^a)) \ \text{(Beta agent response)} && (66.80) \end{aligned}$$

Define $q()$ by

$$q(r;\lambda^a) = P(\widehat{r} = r|\widehat{a} = a, \lambda) \tag{66.81}$$

- PRIOR:

$$\begin{aligned} P(a) &= \sum_\lambda P(\lambda)P(a|\lambda) && (66.82) \\ &= P(a|\lambda^*) && (66.83) \\ &= \mathbb{1}(a = \underset{a}{\text{argmax}} \sum_r r q(r;\lambda^{*a}) && (66.84) \end{aligned}$$

1. Use fact that q =Beta

$$\sum_r r \text{Beta}(r;\lambda^{*a}) = \frac{\alpha^{*a}}{\alpha^{*a} + \beta^{*a}} \tag{66.85}$$

2. Don't use fact that q =Beta.
   For each a, get samples $r^\sigma \sim q(r;\lambda^{*a})$ for $\sigma = 0, 1, \ldots, nsam - 1$ and estimate

$$\sum_r rq(r;\lambda^{*a}) \approx \frac{1}{nsam}\sum_\sigma r^\sigma q(r^\sigma;\lambda^{*a}) \ . \tag{66.86}$$

   This sampling is why TS is called a sampling.

- LIKELIHOOD:

$$P(r|a) = \mu_a^r(1-\mu_a)^r \ \ \text{(Bernoulli environment response)} \tag{66.87}$$

66.5.3 Grad-MAB algorithm

Let

$$\lambda_{t+1}(a) = \lambda_t(a) + \eta r_t[\mathbb{1}(a_t = a) - \pi_t(a)] \tag{66.88}$$

for some $\eta > 0$, where $\pi_t(a)$ is defined by

$$\pi_t(a) = P(\underline{a}_t = a|\lambda_t) = \underbrace{\frac{e^{\lambda_t(a)}}{\sum_a e^{\lambda_t(a)}}}_{\text{softmax}(\lambda_t)(a)} \ . \tag{66.89}$$

The $\lambda_t(a)$ are called **scores** at time t. Let

$$\boxed{a_t^* = \operatorname*{argmax}_a \pi_t(a)} \tag{66.90}$$

Figure 66.7: Bnet for Grad-MAB algorithm.

The Gradient MAB (Grad-MAB) algorithm is described by the bnet Fig.66.7. This bnet differs from the bnet Fig.66.3 in that it includes new nodes $\underline{\lambda}_t$. The TPMs, printed in blue, for bnet Fig.66.7, are as follows.

$$P(\underline{\lambda}_{t+1} = \lambda | a_t, r_t, \lambda_t) = \mathbb{1}(\lambda = \lambda_{t+1} \text{ given by Eq.(66.88).} \tag{66.91}$$

$$P(a_t|\lambda_t) = \mathbb{1}(a_t = a_t^*) \quad \text{or, alternatively, } = P(\underline{a}_t = a_t|\lambda_t) = \pi_t(a_t) \quad \text{(agent's response)} \tag{66.92}$$

$$P(r_t|a_t) = P_{\underline{r}|\underline{a}}(r_t|a_t) \quad \text{(environment's response)} \; . \tag{66.93}$$

Motivation:
Define the **Instantaneous Average Reward** by

$$\mathcal{R}_t(\lambda_t) = \sum_a P(\underline{a}_t = a|\lambda_t) E_{\underline{r}_t|\underline{a}_t=a}[\underline{r}_t] = E_{\underline{r}_t,\underline{a}_t|\lambda_t}[\underline{r}_t] \; . \tag{66.94}$$

We will assume that as $t \to \infty$, $\underline{a}_t \to \underline{a}$, $\underline{r}_t \to \underline{r}$ and $\lambda_t \to \lambda$. Therefore, $\mathcal{R}_t(\lambda_t) \to E_{\underline{r},\underline{a}|\lambda}[\underline{r}] = E_{\underline{r}|\lambda}[\underline{r}]$.

Note that if $E_{\underline{r}_t|\underline{a}_t=a}[\underline{r}_t] = B$, where B is independent of a, then $\mathcal{R}_t(\lambda_t) = B$ and $\frac{\partial \mathcal{R}_t}{\partial \lambda_t(a)} = 0$.

Claim 98 *The gradient of $\mathcal{R}_t(\lambda_t)$ is*

$$\frac{\partial \mathcal{R}_t}{\partial \lambda_t(a)} = E_{\underline{r}_t,\underline{a}_t|\lambda_t}[g(\underline{r}_t, \underline{a}_t|a, \lambda_t)] \tag{66.95}$$

where

$$g(\underline{r}_t, \underline{a}_t|a, \lambda_t) = \underline{r}_t[\mathbb{1}(\underline{a}_t = a) - \pi_t(a)] \tag{66.96}$$

proof:

$$\begin{aligned}
\frac{\partial \mathcal{R}_t}{\partial \lambda_t(a)} &= \sum_{a'} \frac{\partial \pi_t(a')}{\partial \lambda_t(a)} E_{\underline{r}_t|\underline{a}_t=a'}[\underline{r}_t] && (66.97)\\
&= \sum_{a'} \pi_t(a') \frac{\partial \ln \pi_t(a')}{\partial \lambda_t(a)} E_{\underline{r}_t|\underline{a}_t=a'}[\underline{r}_t] && (66.98)\\
&= \sum_{a'} \pi_t(a') \frac{\partial}{\partial \lambda_t(a)} \left[\ln \frac{\exp[\lambda_t(a')]}{\sum_a \exp[\lambda_t(a)]} \right] E_{\underline{r}_t|\underline{a}_t=a'}[\underline{r}_t] && (66.99)\\
&= \sum_{a'} \pi_t(a')[\mathbb{1}(a' = a) - \pi_t(a)] E_{\underline{r}_t|\underline{a}_t=a'}[\underline{r}_t] && (66.100)\\
&= \sum_{a'} P(\underline{a}_t = a'|\lambda_t) E_{\underline{r}_t|\underline{a}_t=a'}[\underline{r}_t(\mathbb{1}(a' = a) - \pi_t(a))] && (66.101)\\
&= E_{\underline{r}_t,\underline{a}_t|\lambda_t}[g(\underline{r}_t, \underline{a}_t|a, \lambda_t)] && (66.102)
\end{aligned}$$

QED

Eq.(66.88) can be written as

$$\lambda_{t+1}(a) = \lambda_t(a) + \eta g(r_t, a_t|a, \lambda_t) \tag{66.103}$$

Chapter 67

Naive Bayes

Figure 67.1: bnet for Naive Bayes with 4 features

Class node $\underline{c} \in val(\underline{c})$. $|val(\underline{c})| = n_{\underline{c}}$= number of classes.
Feature nodes $\underline{x}_i \in val(\underline{x}_i)$ for $i = 0, 1, 2, \ldots, F-1$. F=number of features.
Define

$$x. = [x_0, x_1, \ldots, x_{F-1}] \ . \tag{67.1}$$

For the bnet of Fig.67.1,

$$P(c, x.) = P(c) \prod_{i=0}^{F-1} P(x_i|c) \ . \tag{67.2}$$

Given $x.$ values, find most likely class $c \in val(\underline{c})$.
Maximum a Posteriori (MAP) estimate:

$$c^* = \operatorname*{argmax}_c P(c|x.) \tag{67.3}$$

$$= \operatorname*{argmax}_c \frac{P(c, x.)}{P(x.)} \tag{67.4}$$

$$= \operatorname*{argmax}_c P(c, x.) \ . \tag{67.5}$$

Chapter 68

Neural Networks

In this chapter, we discuss Neural Networks (NNs) of the feedforward kind, which is the most popular kind. In their plain, vanilla form, NNs only have deterministic nodes. But the nodes of a bnet can be deterministic too, because the TPM of a node can reduce to a delta function. Hence, NNs should be expressible as bnets. We will confirm this in this chapter.

Henceforth in this chapter, if we replace an index of an indexed quantity by a dot, it will mean the collection of the indexed quantity for all values of that index. For example, $x.$ will mean the array of x_i for all i.

Figure 68.1: Neural Network (feed forward) with 4 layers: input layer $\underline{x}.$, 2 hidden layers $\underline{h}^0.$, $\underline{h}^1.$ and output layer $\underline{Y}.$

Consider Fig.68.1.

$\underline{x}_i \in \{0,1\}$ for $i = 0,1,2,\ldots,nx-1$ is the **input layer**.

$\underline{h}^\lambda_i \in \mathbb{R}$ for $i = 0,1,2,\ldots,nh(\lambda)-1$ is the **λ-th hidden layer**. $\lambda = 0,1,2,\ldots,\Lambda-2$. A NN is said to be **deep** if $\Lambda > 2$; i.e., if it has more than one hidden layer.

$\underline{Y}_i \in \mathbb{R}$ for $i = 0,1,2,\ldots,ny-1$ is the **output layer**. We use a upper case y here because in the training phase, we will use pairs $(x.[\sigma], y.[\sigma])$ where $y_i[\sigma] \in \{0,1\}$

for $i = 0, 1, \ldots, ny - 1$. $Y = \widehat{y}$ is an estimate of y. Note that lower case y is either 0 or 1, but upper case y may be any real. Often, the activation functions are chosen so that $Y \in [0, 1]$.

The number of nodes in each layer and the number of layers are arbitrary. Fig.68.1 is fully connected (a.k.a. dense), meaning that every node of a layer is impinged arrow coming from every node of the preceding layer. Later on in this chapter, we will discuss non-dense layers.

Let $w^{\lambda}_{i|j}, b^{\lambda}_{i} \in \mathbb{R}$ be given, for $i \in \mathbb{Z}_{[0,nh(\lambda))}$, $j \in \mathbb{Z}_{[0,nh(\lambda-1))}$, and $\lambda \in \mathbb{Z}_{[0,\Lambda)}$.

The TPMs, printed in blue, for bnet Fig.68.1, are as follows.

$$P(x_i \mid x_{i-1}, x_{i-1}, \ldots, x_0) = \text{given} \tag{68.1}$$

$$P(h^{\lambda}_{i} \mid h^{\lambda-1}_{\cdot}) = \delta\left(h^{\lambda}_{i}, \mathcal{A}^{\lambda}_{i}(\sum_{j} w^{\lambda}_{i|j} h^{\lambda-1}_{j} + b^{\lambda}_{i})\right) , \tag{68.2}$$

where $P(h^0_i|h^{-1}) = P(h^0_i|x)$.

$$P(Y_i \mid h^{\Lambda-2}_{\cdot}) = \delta\left(Y_i, \mathcal{A}^{\Lambda-1}_{i}(\sum_{j} w^{\Lambda-1}_{i|j} h^{\Lambda-2}_{j} + b^{\Lambda-1}_{i})\right) . \tag{68.3}$$

68.1 Activation Functions $\mathcal{A}^{\lambda}_{i} : \mathbb{R} \to \mathbb{R}$

Activation functions must be nonlinear. Why? Because if they were all linear, the NN mapping would be a bijection (1-1 onto map), and its domain and range would be the same. That is not what you want for a classifier. For a classifier, you want the range to be much smaller than the domain.

- **Step function (Perceptron)**

$$\mathcal{A}(x) = \mathbb{1}(x > 0) \tag{68.4}$$

  Zero for $x \leq 0$, one for $x > 0$.

- **Sigmoid function**

$$\mathcal{A}(x) = \frac{1}{1 + e^{-x}} = \text{smoid}(x) \tag{68.5}$$

  Smooth, monotonically increasing function. smoid$(-\infty) = 0$,smoid$(0) = 1/2$, smoid$(\infty) = 1$.

- **Hyperbolic tangent**

$$\mathcal{A}(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{68.6}$$

Smooth, monotonically increasing function. $\tanh(-\infty) = -1$,$\tanh(0) = 0$, $\tanh(\infty) = 1$.

Odd function:

$$\tanh(-x) = -\tanh(x) \tag{68.7}$$

Whereas $\text{smoid}(x) \in [0, 1]$, $\tanh(x) \in [-1, 1]$.

- **ReLU (Rectified Linear Unit)**

$$\mathcal{A}(x) = \underbrace{x\mathbb{1}(x > 0)}_{x_+} = \max(0, x) \ . \tag{68.8}$$

Compare this to the step function $\mathbb{1}(x > 0)$.

- **Swish**

$$\mathcal{A}(x) = x \ \text{smoid}(x) \tag{68.9}$$

- **Softmax**

$$\mathcal{A}(x_i|x.) = \frac{e^{x_i}}{\sum_i e^{x_i}} = \text{softmax}(x.)(i) \tag{68.10}$$

The softmax definition implies that the bnet nodes within a softmax layer are fully connected by arrows to form a "clique".

68.2 Weight optimization via supervised training and gradient descent

The bnet of Fig.68.1 is used for classification of a single data point x.. It assumes that the weights $w^{\lambda}_{i|j}, b^{\lambda}_i$ are given.

To find the optimum weights via supervised training and gradient descent, one uses the bnet Fig.68.2.

In Fig.68.2, the nodes in Fig.68.1 become sampling space vectors. For example, $\underline{x}.$ becomes $\underline{\vec{x}.}$, where the components of $\underline{\vec{x}.}$ in sampling space are $\underline{x}.[\sigma] \in \{0, 1\}^{nx}$ for $\sigma = 0, 1, \ldots, nsam(\vec{x}) - 1$. $nsam(\vec{x})$ is the number of samples in the whole dataset.

To train a NN bnet with a dataset, the standard procedure is to split the dataset into 3 parts (I like to call them the **ttt datasets**):

1. **training dataset**,

2. **tuning (a.k.a. validation) dataset**, for tuning of hyperparameters like $nsam(\vec{x})$, Λ, and $nh(i)$ for each i.

3. **testing dataset**

Weights only change while training on the training dataset. While the model is being trained, its performance is periodically tested on the tuning dataset. Training continues until performace on the tuning dataset no longer improves. After that happens, the model is finally applied to the testing dataset.

The training dataset is split into batches. An **epoch** is a pass through all the batches in the training dataset.

Define

$$W^{\lambda}_{i|j} = [w^{\lambda}_{i|j}, b^{\lambda}_{i}] \ . \tag{68.11}$$

The TPMs, printed in blue, for bnet Fig.68.2, are as follows.

$$\begin{tikzcd}
 & \underline{\vec{x}.} \arrow[r] \arrow[d] & \underline{\vec{y}.} \arrow[dddd] & \\
\underline{W}^{0}_{\cdot|\cdot}, \arrow[r] & \underline{\vec{h}^{0}_{\cdot}} \arrow[d] & & \\
\underline{W}^{1}_{\cdot|\cdot} \arrow[r] & \underline{\vec{h}^{1}_{\cdot}} \arrow[d] & & \\
\underline{W}^{2}_{\cdot|\cdot} \arrow[r] & \underline{\vec{h}^{2}_{\cdot}} \arrow[d] & & \\
\underline{W}^{\cdot}_{\cdot|\cdot} \arrow[uuu] \arrow[uu] \arrow[u] \arrow[rrr, bend right] & \underline{\vec{Y}}. \arrow[r] & \mathcal{E} \arrow[r] & (\underline{W'})^{\cdot}_{\cdot|\cdot}
\end{tikzcd}$$

Figure 68.2: bnet for finding optimum weights of the bnet Fig.68.1 via supervised training and gradient descent.

$$P(x.[\sigma]) = \text{given} \ . \tag{68.12}$$

$$P(y.[\sigma] \mid x.[\sigma]) = \text{given} \ . \tag{68.13}$$

$$P(h_i^\lambda[\sigma] \mid h_\cdot^{\lambda-1}[\sigma]) = \delta\left(h_i^\lambda[\sigma], \mathcal{A}_i^\lambda(\sum_j w_{i|j}^\lambda h_j^{\lambda-1}[\sigma] + b_i^\lambda)\right) \tag{68.14}$$

$$P(Y_i[\sigma] \mid h_\cdot^{\Lambda-2}[\sigma]) = \delta\left(Y_i[\sigma], \mathcal{A}_i^{\Lambda-1}(\sum_j w_{i|j}^{\Lambda-1} h_j^{\Lambda-2}[\sigma] + b_i^{\Lambda-1})\right) \tag{68.15}$$

$$P(W_{\cdot|\cdot}) = \text{given} \tag{68.16}$$

The first time it is used, $W_{\cdot|\cdot}$ is arbitrary. After the first time, it is determined by previous stage.

$$P(W_{\cdot|\cdot}^\lambda | W_{\cdot|\cdot}) = \delta(W_{\cdot|\cdot}^\lambda, (W_{\cdot|\cdot})^\lambda) \tag{68.17}$$

$$P(\mathcal{E}|\vec{y}\cdot, \vec{Y}\cdot) = \frac{1}{nsam(\vec{x})} \sum_\sigma \sum_i d(y_i[\sigma], Y_i[\sigma]) \ , \tag{68.18}$$

where

$$d(y, Y) = |y - Y|^2 \ . \tag{68.19}$$

If $y, Y \in [0, 1]$, one can use this instead

$$d(y, Y) = XE(y \to Y) = -y \ln Y - (1 - y) \ln(1 - Y) \ . \tag{68.20}$$

$$P((W')_{i|j}^\lambda | \mathcal{E}, W_{\cdot|\cdot}) = \delta((W')_{i|j}^\lambda, W_{i|j}^\lambda - \eta \partial_{W_{i|j}^\lambda} \mathcal{E}) \tag{68.21}$$

$\eta > 0$ is called the learning rate. This method of minimizing the error $\mathcal{E}$ is called gradient descent. $W' - W = \Delta W = -\eta \partial_W \mathcal{E}$ so $\Delta \mathcal{E} = \frac{-1}{\eta}(\Delta W)^2 < 0$.

68.3 Non-dense layers

The TPM for a non-dense layer is of the form:

$$P(h_i^\lambda[\sigma] \mid h_\cdot^{\lambda-1}[\sigma]) = \delta(h_i^\lambda[\sigma], H_i^\lambda[\sigma]) \ , \tag{68.22}$$

where $H_i^\lambda[\sigma]$ will be specified below for each type of non-dense layer.

- **Dropout Layer**

  The dropout layer was invented in Ref.[78]. To dropout nodes from a fixed layer λ: For all i of layer λ, define a new node $\underline{r}_i^\lambda$ with an arrow $\underline{r}_i^\lambda \to \underline{h}_i^\lambda$. For $r \in \{0,1\}$, and some $p \in (0,1)$, define

$$P(r_i^\lambda = r) = [p]^r [1-p]^{1-r} \text{ (Bernoulli dist.)} . \tag{68.23}$$

  Now one has

$$P(h_i^\lambda[\sigma] \mid h_\cdot^{\lambda-1}[\sigma], r_i^\lambda) = \delta(h_i^\lambda[\sigma], H_i^\lambda[\sigma]) , \tag{68.24}$$

  where

$$H_i^\lambda[\sigma] = \mathcal{A}_i^\lambda(r_i^\lambda \sum_j w_{i|j}^\lambda h_j^{\lambda-1}[\sigma] + b_i^\lambda) . \tag{68.25}$$

  This reduces overfitting. Overfitting might occur if the weights follow too closely several similar batches. This dropout procedure adds a random component to each batch making groups of similar batches less likely.

  The random $\underline{r}_i^\lambda$ nodes that induce dropout are only used in the training bnet Fig.68.2, not in the classification bnet Fig.68.1. We prefer to remove the $\underline{r}_i^\lambda$ stochasticity from classification and for Fig.68.1 to act as an average over sampling space of Fig.68.2. Therefore, if weights $w_{i|j}^\lambda$ are obtained for a dropout layer λ in Fig.68.2, then that layer is used in Fig.68.1 with no $\underline{r}_i^\lambda$ nodes but with weights $\left\langle r_i^\lambda \right\rangle w_{i|j}^\lambda = p w_{i|j}^\lambda$.

  Note that dropout adds non-deterministic nodes to a NN, which in their vanilla form only have deterministic nodes.

- **Convolutional Layer**

  - 1-dim
    Filter function $\mathcal{F} : \{0, 1, \ldots, nf-1\} \to \mathbb{R}$.
    σ=stride length
    For $i \in \{0, 1, \ldots, nh(\lambda)-1\}$, let

$$H_i^\lambda[\sigma] = \sum_{j=0}^{nf-1} h_{j+i\sigma}^{\lambda-1}[\sigma] \mathcal{F}(j) . \tag{68.26}$$

    For the indices not to go out of bounds in Eq.(68.26), we must have

$$nh(\lambda-1) - 1 = nf - 1 + (nh(\lambda)-1)\sigma \tag{68.27}$$

    so

$$nh(\lambda) = \frac{1}{\sigma}[nh(\lambda-1) - nf] + 1 . \tag{68.28}$$

- 2-dim
  $h_i^\lambda[\sigma]$ becomes $h_{(i,j)}^\lambda[\sigma]$. Do 1-dim convolution along both i and j axes.

- **Pooling Layers (MaxPool, AvgPool)**

  Here each node i of layer λ is impinged by arrows from a subset $Pool(i)$ of the set of all nodes of the previous layer $\lambda - 1$. Partition set $\{0, , 1, \ldots, nh(\lambda - 1) - 1\}$ into $nh(\lambda)$ mutually disjoint, nonempty sets called $Pool(i)$, where $i \in \{0, 1, \ldots, nh(\lambda) - 1\}$.

  - AvgPool
  $$H_i^\lambda[\sigma] = \frac{1}{|Pool(i)|} \sum_{j \in Pool(i)} h_j^{\lambda-1}[\sigma] \tag{68.29}$$

  - MaxPool
  $$H_i^\lambda[\sigma] = \max_{j \in Pool(i)} h_j^{\lambda-1}[\sigma] \tag{68.30}$$

68.4 Autoencoder NN

If the sequence

$$nx, nh(0), nh(1), \ldots, nh(\Lambda - 2), ny \tag{68.31}$$

first decreases monotonically up to layer λ_{min}, then increases monotonically until $ny = nx$, then the NN is called an **autoencoder NN**. Autoencoders are useful for unsupervised learning and feature reduction. In this case, Y estimates x. The layers before layer λ_{min} are called the **encoder**, and those after λ_{min} are called the **decoder**. Layer λ_{min} is called the **code**.

Chapter 69

Noisy-OR gate

The Noisy-OR gate was first proposed by Judea Pearl in his 1988 book Ref.[60].

Figure 69.1: Noisy-OR gate $\underline{y} \in \{0,1\}$ with $n = 3$, Boolean inputs $(\underline{x}_i)_{i=0,1,2}$ and parameters $\underline{\lambda}, (\underline{\pi})_{i=0,1,2}$.

Let
$\underline{\lambda} \in [0,1]$ =**gate lea.k.a.ge**.
$y \in \{0,1\}$ = gate output
$\underline{x}^n = (\underline{x}_i)_{i=0,1,\ldots,n-1}$, where $\underline{x}_i \in \{0,1\}$ are gate inputs.
$\underline{\pi}^n = (\underline{\pi}_i)_{i=0,1,\ldots,n-1}$, where $\underline{\pi}_i \in [0,1]$ are gate parameters.

The TPM, printed in blue, for the Noisy-OR gate $\underline{y}$ shown in Fig.69.1, is as follows.

$$P(y = 1|x^n, \lambda, \pi^n) = 1 - (1 - \lambda) \prod_i [1 - \pi_i x_i] \tag{69.1}$$

$$P(y = 0|x^n, \lambda, \pi^n) = 1 - P(y = 1|x^n, \lambda, \pi^n) \tag{69.2}$$

Note that if $\lambda = 0$ and $\pi_i = 1$ for all i, then this becomes a deterministic OR-gate. Indeed,

$$P(y=1|x^n,\lambda=0,\pi^n=1^n)=1-\prod_i[1-x_i]=\vee_{i=0}^{n-1}x_i\ , \tag{69.3}$$

so

$$P(y|x^n,\lambda=0,\pi^n=1^n)=\delta(y,\vee_{i=0}^{n-1}x_i)\ . \tag{69.4}$$

69.1 3 ways to interpret the parameters π_i

1. Note that if $\lambda=0$ and x^n is one hot (i.e., $x^n=e_i^n$, where e_i^n is the vector with all components zero except for the i-th component which equals 1), then

$$P(y=1|x^n=e_i^n,\lambda=0,\pi^n)=1-[1-\pi_i]=\pi_i\ . \tag{69.5}$$

This gives an interpretation to the parameters π_i.

Figure 69.2: Fig.69.1 after replacing parameters $(\underline{\pi}_i)_{i=0,1,2}$ by hidden nodes $(\underline{h}_i)_{i=0,1,2}$.

2. Another way of interpreting the parameters π_i is to associate each of them with a hidden variable $\underline{h}_i\in\{0,1\}$ whose average equals π_i. More precisely, consider Fig.69.2.

   Let $\underline{x}_i,\underline{h}_i,\underline{A}_i,\underline{y}\in\{0,1\}$.

   The TPMs, printed in blue, for the bnet Fig.69.2, are as follows:

$$P(h_i)=\pi_i\delta(h_i,1)+(1-\pi_i)\delta(h_i,0) \tag{69.6}$$

$$P(A_i|h_i,x_i)=\delta(A_i,h_i\wedge x_i)=\delta(A_i,h_ix_i) \tag{69.7}$$

$$P(y=1|A^n) = 1-(1-\lambda)\wedge_{i=0}^{n-1}\overline{A}_i \tag{69.8}$$
$$= 1-(1-\lambda)\prod_i(1-A_i) \tag{69.9}$$

$$P(y=0|A^n) = 1-P(y=1|A^n) \tag{69.10}$$

Note that

$$P(y=1|x^n,\lambda) = \sum_{h^n}\sum_{A^n}\left[1-(1-\lambda)\prod_i(1-A_i)\right][\prod_i\delta(A_i,h_ix_i)]P(h^n) \tag{69.11}$$
$$= E_{\underline{h}^n}\left[[1-(1-\lambda)\prod_i(1-h_ix_i)\right] . \tag{69.12}$$

But

$$E_{\underline{h}_i}[h_ix_i] = \sum_{h_i=0,1}P(h_i)h_ix_i = \pi_ix_i \tag{69.13}$$

so

$$P(y=1|x^n,\lambda) = 1-(1-\lambda)\prod_i(1-\pi_ix_i) . \tag{69.14}$$

Figure 69.3: Fig.69.2 after replacing the hidden nodes $(\underline{h}_i)_{i=0,1,2}$ by vectors of samples $(\vec{\underline{h}}_i)_{i=0,1,2}$.

3. Another way to interpret the parameters π_i is to associate each of them with a vector of samples $\vec{\underline{h}}_i$ whose average is π_i. More precisely, consider Fig.69.3.

   Suppose $\underline{h}_i \in \{0,1\}$ and define

$$P_{\underline{h}_i}(h_i) = \pi_i \delta(h_i, 1) + (1-\pi_i)\delta(h_i, 0) \ . \tag{69.15}$$

   Suppose $\vec{\underline{h}}_i = (\underline{h}_i[\sigma])_{s=0,1,\ldots,nsam-1}$ and the Boolean samples $\underline{h}_i[\sigma] \in \{0,1\}$ are i.i.d. with $\underline{h}_i[\sigma] \sim P_{\underline{h}_i}$ for all σ.

   Note that for each i, an estimate $\widehat{P}_{\underline{h}_i}(h_i)$ of $P_{\underline{h}_i}(h_i)$ can be obtained from the vector of samples $\vec{\underline{h}}_i$ as follows:

$$\widehat{P}_{\underline{h}_i}(h_i) = \frac{1}{nsam} \sum_{\sigma=0}^{nsam-1} \mathbb{1}(h_i[\sigma] = h_i) \ . \tag{69.16}$$

   Let $\underline{x}_i, \underline{h}_i[\sigma], \underline{A}_i, \underline{y} \in \{0,1\}$.

   The TPMs, printed in blue, for the bnet Fig.69.3, are as follows:

$$P(\vec{h}_i) = \prod_{\sigma=0}^{nsam-1} P_{\underline{h}}(h_i[\sigma]) \tag{69.17}$$

$$\begin{aligned} P(A_i \mid \vec{h}_i, x_i) &= \delta(A_i, \frac{1}{nsam} \sum_{\sigma} h_i[\sigma] \wedge x_i) && (69.18) \\ &= \delta(A_i, \pi_i x_i) && (69.19) \end{aligned}$$

$$\begin{aligned} P(y=1|A^n) &= 1 - (1-\lambda) \wedge_{i=0}^{n-1} \overline{A}_i && (69.20) \\ &= 1 - (1-\lambda) \prod_i (1 - A_i) && (69.21) \end{aligned}$$

$$P(y=0|A^n) = 1 - P(y=1|A^n) \tag{69.22}$$

   Note that

$$\begin{aligned} P(y=1|x^n, \lambda, \vec{h}^n) &= \sum_{A^n} \left[1 - (1-\lambda) \prod_i (1 - A_i) \right] \prod_i \delta(A_i, \pi_i x_i) && (69.23) \\ &= 1 - (1-\lambda) \prod_i (1 - \pi_i x_i) \ . && (69.24) \end{aligned}$$

Chapter 70

Non-negative Matrix Factorization

This chapter about Non-Negative (NN) Matrix Factorization (MF) is based on Ref.[171].

In NN MF, we are given a matrix Y with NN entries. Our goal is to factor it into a product of two matrices W and X that also have NN entries.

$$Y = WX \ , \tag{70.1}$$

where

$Y \in \mathbb{R}_{\geq 0}^{ny \times na}$: visible data
$W \in \mathbb{R}_{\geq 0}^{ny \times nx}$: weight data
$X \in \mathbb{R}_{\geq 0}^{nx \times na}$: hidden data

Usually, $ny > nx < na$ so NN MF can be viewed as a type of (non-lossy) compression of information.

70.1 Bnet interpretation

Express node $\underline{y}$ as a chain of two nodes.

$$\underline{y} \longleftarrow \underline{a} \qquad = \qquad \underline{y} \longleftarrow \underline{x} \longleftarrow \underline{a}$$

Figure 70.1: Bnet interpretation of non-negative matrix factorization.

The TPMs, printed in blue, for bnet Fig.70.1, are as follows.

$$P(\underline{y} = y|a) = \frac{Y_{y,a}}{\sum_y Y_{y,a}} \tag{70.2}$$

$$P(y|x) = \frac{W_{y,x}}{\sum_y W_{y,x}} \tag{70.3}$$

$$P(x|a) = \frac{\sum_y W_{y,x}}{\sum_y Y_{y,a}} X_{x,a} \tag{70.4}$$

70.2 Simplest recursive algorithm

- Initialize: Choose nx. Choose $W^{(0)}$ and $X^{(0)}$ that have NN entries.
- Update: For $n = 0, 1, \ldots,$ do

$$X_{i,j}^{(n+1)} \leftarrow X_{i,j}^{(n)} \frac{[(W^{(n)})^T Y]_{i,j}}{[(W^{(n)})^T \underbrace{W^{(n)} X^{(n)}}_{\approx Y}]_{i,j}} \tag{70.5}$$

  and

$$W_{i,j}^{(n+1)} \leftarrow W_{i,j}^{(n)} \frac{[Y(X^{(n+1)})^T]_{i,j}}{[\underbrace{W^{(n)} X^{(n+1)}}_{\approx Y}(X^{(n+1)})^T]_{i,j}} \ . \tag{70.6}$$

- After each step, record error defined by

$$\mathcal{E}^{(n)} = \| \ Y - W^{(n)} X^{(n)} \ \|_2 \ \ . \tag{70.7}$$

  In the last expression $\| \cdot \|_2$ is the 2-norm, a.k.a. Frobenius matrix norm. Continue until reach acceptable error.

  Can also use Kullback-Leibler divergence for error:

$$\mathcal{E} = \sum_a P(a) D_{KL}(P(\underline{y} = y|a) \ \| \ \sum_x P(y|x) P(x|a)) \ , \tag{70.8}$$

  for some arbitrary choice of prior $P(a)$. For example, can choose $P(a)$ uniform.

Chapter 71

Observationally Equivalent DAGs

This chapter is based on Chapter 1 of Ref.[62] and on a blog post by Bruno Gonçalves (Ref.[22]).

A probability distribution P **is compatible with a DAG** G if P and G have the same random variables, and they can be combined to form a bnet without contradictions; i.e., one can calculate all the TPMs from P and multiply them together to obtain P again. Let

$$\mathcal{P}(G) = \{P : P \text{ is compatible with } G\} \ . \tag{71.1}$$

Two DAGs G **and** G' **are observationally equivalent (OE)** if $\mathcal{P}(G) = \mathcal{P}(G')$. Hence, any total probability distribution that is compatible with one of them is compatible with the other. For example, $\underline{a} \rightarrow \underline{b}$ and $\underline{a} \leftarrow \underline{b}$ are OE because

$$P(a|b)P(b) = P(a,b) = P(b|a)P(a) \ . \tag{71.2}$$

We'll say two bnets are OE if their DAGs are OE.

Two DAGs G and G' are **d-separation equivalent** if $DS(G) = DS(G')$. See Chapter 24 for definition of $DS(G)$.

Claim 99 *Two DAGs are OE iff their DAGs are d-separation equivalent.*

The **skeleton** of a DAG is its underlying undirected graph.

A **v-structure** in a DAG consists of two arrows converging to a node and such that their tails are not connected by a third arrow. Fig.71.1 shows in red all the v-structures of a particular DAG.

Claim 100 *Observational Equivalence Theorem (by Verma and Pearl, 1990)*
Two DAGs are OE iff they have the same skeletons and the same v-structures.

71.1 Examples

The 3 DAGs in Fig.71.2 are OE. They form an equivalence class of OE DAGs that represent the same probability distribution. This equivalence class of DAGs can be

Figure 71.1: Example showing in red all v-structures of a particular DAG.

Figure 71.2: These 3 DAGs are observationally equivalent (OE).

represented by the partially directed graph Fig.71.3. These 3 DAGs can be proven to be OE in the following 3 ways:

1. Write the generic probability distributions represented by the 3 DAGs, and show that they are equal, as we did in Eq.(71.2). That is the low brow way of proving OE.

2. Use d-separation (see Chapter 24). Consider DAG (a) first. Rename the nodes as $\underline{\tau}_j$ with $j = 1, 2, \ldots$ so that the names are in topological order (i.e., so that the parents of $\underline{\tau}_j$ have indices that are smaller than j). The node names $\underline{x}_j$ of DAG (a) are already in topological order, so we skip this step for DAG (a). Now write down its total probability distribution and notice which parents of a fully connected DAG were omitted.

Figure 71.3: This partially directed graph represents the 3 DAGs in Fig.71.2.

$$P(x_1, x_2, x_3, x_4, x_5) = \underbrace{P(x_5|x_4)}_{x_3, x_2, x_1 \text{ omitted}} \underbrace{P(x_4|x_3, x_2)}_{x_1 \text{ omitted}} \underbrace{P(x_3|x_1)}_{x_2 \text{ omitted}} P(x_2|x_1)P(x_1) \quad (71.3)$$

The observations of which parents were omitted can be stated in d-separation lingo as the following 3 orthogonality relations:[1]

$$\underline{x}_3 \perp_P \underline{x}_2 \quad | \quad \underline{x}_1 \quad (71.4a)$$
$$\underline{x}_4 \perp_P \underline{x}_1 \quad | \quad \underline{x}_2, \underline{x}_3 \quad (71.4b)$$
$$\underline{x}_5 \perp_P^{sep} (\underline{x}_1, \underline{x}_2, \underline{x}_3) \quad | \quad \underline{x}_4 \ . \quad (71.4c)$$

Going through the same procedure for the other 2 DAGs yields, for each of them, an equivalent set of 3 orthogonality equations.[2]

This is enough to conclude that the 3 DAGs of Fig.71.2 are OE.

Note that Eqs.(71.4) encompass all that there is to say about the observability of DAG (a). These 3 equations can be checked empirically to assess how well the DAG fits the data. For example, one can do OLS (ordinary least squares) regression $\underline{x}_5 \sim \underline{x}_1 + \underline{x}_2 + \underline{x}_3 + \underline{x}_4$ on the data, i.e., try to fit $x_5 = \beta_0 + \sum_{i=1}^4 \beta_i x_i$ to the data, and find that, to a good approximation, $\beta_1 = \beta_2 = \beta_3 = 0$.

3. Use the OE Theorem. All three DAGs have the same skeleton, and the same single v-structure $\underline{x}_2 \to \underline{x}_4 \leftarrow \underline{x}_3$.

[1]Normally, if we had changed from the original node names to the $\underline{\tau}_j$ node names, these orthogonality relations would first be stated in terms of the $\underline{\tau}_j$ names, and we could translate them so that they were stated in terms of the original node names. But for DAG (a) there was no need to use the $\underline{\tau}_j$ names.

[2]The $\underline{x}_j$ node names are no longer in topological order for DAGs (b) and (c) so for them you should go through the intermediate step of renaming the nodes $\underline{\tau}_j$, and then, after obtaining the orthogonality relations in terms of the $\underline{\tau}_j$ names, translating them back to the original $\underline{x}_j$ names.

Chapter 72

Omitted Variable Bias

This paper is loosely based on Refs.[14] and [12].

This chapter assumes that the reader has read Chapter D on Linear Regression (LR) and Section 77.12 which is an introduction to sensitivity analysis for the Potential Outcomes (PO) model.

We will use the terms **"PO Sensitivity Analysis"** and **"Omitted Variable Bias (OVB)"** to mean the same thing. In this chapter, we consider 2 types of OVB. LDEN bnets for these two cases are depicted in Fig.72.1. In Fig.72.1(a), the omitted variable is a confounder, and in Fig.72.1 (b), it's a mediator.

Next we express OVB for these two cases as a product of gains along a path that goes through the unobserved node, and then we re-express the OVB in terms of correlations between the nodes.

Figure 72.1: LDEN bnets used to do PO sensitivity analysis. Node $\underline{c}$ in (a) is an unobserved confounder, and node $\underline{m}$ in (b) is an unobserved mediator.

CASE (a), confouder

Consider the LDEN bnet of Fig.72.1(a), whose structural equations, printed in blue, are as follows:

$$\begin{cases} \underline{c} = \underline{\epsilon_c} \\ \underline{x} = \epsilon_{\underline{x}} \\ \underline{d} = \alpha \underline{x} + \alpha' \underline{c} + \epsilon_{\underline{d}} \\ \underline{y} = \delta \underline{d} + \beta \underline{x} + \beta' \underline{c} + \epsilon_{\underline{y}} \end{cases} \tag{72.1}$$

In Section 77.12, we showed that for confounder case (a),

$$\boxed{OVB_{con} = ATE - ATE|_{\beta'=0} = \frac{\beta'}{\alpha'}} \tag{72.2}$$

This result is easy to understand. $ATE|_{\beta'=0} = \delta$ due to the directed path $\underline{d} \to \underline{y}$, whereas $ATE = \delta + \frac{\beta'}{\alpha'}$ due to the two directed paths $\underline{d} \to \underline{y}$ and $\underline{d} \to \underline{c} \to \underline{y}$. Note that traveling from $\underline{c}$ to $\underline{d}$ has gain α', so traveling in the opposite direction has gain $\frac{1}{\alpha'}$.

Next we express OVB_{con} in terms of correlations between the nodes of the bnet.

Claim 101

$$OVB_{con} = \frac{\beta'}{\alpha'} = \frac{[\rho_{\underline{y},\underline{c}} \sigma_{\underline{y}}]^{|d,x}}{[\rho_{\underline{d},\underline{c}} \sigma_{\underline{d}}]^{|x}} \tag{72.3}$$

If $\sigma_{\epsilon_{\underline{m}}} = \sigma_{\epsilon_{\underline{d}}} = 0$, *then* $\rho_{\underline{y},\underline{c}} = \rho_{\underline{d},\underline{c}} = 1$ *and*

$$OVB_{con} = \frac{\sigma_{\underline{y}}^{|d,x}}{\sigma_{\underline{d}}^{|x}} \tag{72.4}$$

proof: [1]

Note that

$$\left\langle \underline{c}, \underline{\epsilon_d} \right\rangle = 0 \qquad |x \tag{72.5}$$

because the paths from $\underline{c}$ to $\underline{\epsilon_d}$ are blocked by colliders. Hence,

$$\left\langle \underline{d}, \underline{\epsilon_d} \right\rangle = \left\langle \alpha' \underline{c} + \underline{\epsilon_d}, \underline{\epsilon_d} \right\rangle = \left\langle \underline{\epsilon_d}, \underline{\epsilon_d} \right\rangle \qquad |x \tag{72.6}$$

$$\langle \underline{d}, \underline{d} \rangle = \left\langle \alpha' \underline{c} + \underline{\epsilon_d}, \alpha' \underline{c} + \underline{\epsilon_d} \right\rangle = (\alpha')^2 \langle \underline{c}, \underline{c} \rangle + \left\langle \underline{\epsilon_d}, \underline{\epsilon_d} \right\rangle \qquad |x \tag{72.7}$$

$$\sigma_{\underline{d}} = |\alpha'| \sigma_{\underline{c}} \sqrt{1 + \left(\frac{\sigma_{\underline{\epsilon_d}}}{\alpha' \sigma_{\underline{c}}} \right)^2} \qquad |x \tag{72.8}$$

[1] We use "$|x$" at the end of a line to mean all averages in that line are taken at fixed $\underline{x} = x$.

$$\rho_{\underline{d},\underline{c}} = \frac{\langle \underline{d}, \underline{c} \rangle}{\sqrt{\langle \underline{d}, \underline{d} \rangle \langle \underline{c}, \underline{c} \rangle}} \qquad |x \tag{72.9}$$

$$= \frac{\left\langle \alpha' \underline{c} + \underline{\epsilon_d}, \underline{c} \right\rangle}{\sqrt{\left\langle \alpha' \underline{c} + \underline{\epsilon_d}, \alpha' \underline{c} + \underline{\epsilon_d} \right\rangle \langle \underline{c}, \underline{c} \rangle}} \qquad |x \tag{72.10}$$

$$= \frac{\alpha' \langle \underline{c}, \underline{c} \rangle}{\sqrt{((\alpha')^2 \langle \underline{c}, \underline{c} \rangle + \left\langle \underline{\epsilon_d}, \underline{\epsilon_d} \right\rangle) \langle \underline{c}, \underline{c} \rangle}} \qquad |x \tag{72.11}$$

$$= \frac{\alpha' \sigma_{\underline{c}}}{\sqrt{(\alpha' \sigma_{\underline{c}})^2 + \sigma^2_{\underline{\epsilon_d}}}} \qquad |x \tag{72.12}$$

$$= \frac{\text{sign}(\alpha')}{\sqrt{1 + \left(\frac{\sigma_{\underline{\epsilon_d}}}{\alpha' \sigma_{\underline{c}}}\right)^2}} \qquad |x \tag{72.13}$$

$$\langle \underline{c}, \underline{d} \rangle = \alpha' \langle \underline{c}, \underline{c} \rangle \qquad |x \tag{72.14}$$

$$\alpha' = \frac{\langle \underline{c}, \underline{d} \rangle}{\langle \underline{c}, \underline{c} \rangle} = \partial_{\underline{c}} \underline{d} = \rho_{\underline{c},\underline{d}} \frac{\sigma_{\underline{d}}}{\sigma_{\underline{c}}} \qquad |x \tag{72.15}$$

All equations between Eq.(72.5) and this point, remain valid if we make the following replacements:

$$\begin{array}{lcl} \underline{d} & \rightarrow & \underline{y} \\ \underline{\epsilon_d} & \rightarrow & \underline{\epsilon_y} \\ \underline{c} & \rightarrow & \underline{c} \\ \alpha' & \rightarrow & \beta' \\ |x & \rightarrow & |d, x \end{array} \tag{72.16}$$

In particular, the following are true

$$\sigma_{\underline{y}} = |\beta'| \sigma_{\underline{c}} \sqrt{1 + \left(\frac{\sigma_{\underline{\epsilon_y}}}{\beta' \sigma_{\underline{c}}}\right)} \qquad |d, x \tag{72.17}$$

$$\rho_{\underline{y},\underline{c}} = \frac{sign(\beta')}{\sqrt{1 + \left(\frac{\sigma_{\underline{\epsilon_y}}}{\beta' \sigma_{\underline{c}}}\right)^2}} \qquad |d, x \tag{72.18}$$

$$\beta' = \partial_{\underline{c}} \underline{y} = \rho_{\underline{y},\underline{c}} \frac{\sigma_{\underline{y}}}{\sigma_{\underline{c}}} \qquad |d, x \tag{72.19}$$

Combining our newly found expressions for α' and β', we get

$$\frac{\beta'}{\alpha'} = \left[\frac{\rho_{\underline{y},\underline{c}}\sigma_{\underline{y}}}{\sigma_{\underline{c}}}\right]^{|d,x} \left[\frac{\sigma_{\underline{c}}}{\rho_{\underline{d},\underline{c}}\sigma_{\underline{d}}}\right]^{|x} \tag{72.20}$$

Now note that

$$\sigma^2_{\underline{c}} = \left\langle \epsilon_{\underline{c}}, \epsilon_{\underline{c}} \right\rangle \tag{72.21}$$

which is independent of $\underline{d}$ and $\underline{x}$, so

$$\sigma^{|x}_{\underline{c}} = \sigma^{|d,x}_{\underline{c}} \tag{72.22}$$

Therefore,

$$\frac{\beta'}{\alpha'} = \frac{[\rho_{\underline{y},\underline{c}}\sigma_{\underline{y}}]^{|d,x}}{[\rho_{\underline{d},\underline{c}}\sigma_{\underline{d}}]^{|x}} \tag{72.23}$$

QED

CASE (b)**, mediator**

Consider the LDEN bnet of Fig.72.1(b), whose structural equations, printed in blue, are as follows:

$$\begin{cases} \underline{m} = \lambda \underline{d} + \epsilon_{\underline{m}} \\ \underline{x} = \epsilon_{\underline{x}} \\ \underline{d} = \alpha \underline{x} + \epsilon_{\underline{d}} \\ \underline{y} = \delta \underline{d} + \beta \underline{x} + \mu \underline{m} + \epsilon_{\underline{y}} \end{cases} \tag{72.24}$$

By an argument analogous to the one used previously in this chapter to prove that $OVB_{con} = \frac{\beta'}{\alpha'}$, we get

$$\boxed{OVB_{med} = ATE - ATE|_{\mu=0} = \lambda\mu} \tag{72.25}$$

Next we express OVB_{med} in terms of correlations between the nodes of the bnet.

Claim 102 [2]

$$OVB_{med} = \lambda\mu = \frac{[\rho_{\underline{y},\underline{m}}\sigma_{\underline{y}}]^{|d,x}}{\sigma_{\underline{d}}} \underbrace{\left[\frac{\rho_{\underline{d},\underline{m}}}{\sqrt{1-\rho^2_{\underline{d},\underline{m}}}}\right]}_{\tan\theta} \tag{72.26}$$

If $|\tan\theta| \leq \eta$*, then*[3]

$$|OVB_{med}| \leq \left|\frac{\sigma^{|d,x}_{\underline{y}}}{\sigma_{\underline{d}}}\right| \eta \tag{72.27}$$

---

[2] Eq.(72.26) appears in Ref.[14], but that paper claims it gives the bias for unobserved confounders instead of unobserved mediators. But the formula for the bias for unobserved confounders is given by Eq.(72.3).

[3] Recall that, by the Schwarz Inequality, $|\rho_{\underline{a},\underline{b}}| \leq 1$ for any correlation coefficient $\rho_{\underline{a},\underline{b}} = \frac{\langle \underline{a},\underline{b} \rangle}{\sigma_{\underline{a}}\sigma_{\underline{b}}}$.

proof:

Note that

$$\left\langle \underline{d}, \epsilon_{\underline{m}} \right\rangle = 0 \tag{72.28}$$

because paths from $\underline{d}$ to $\underline{\epsilon}_{\underline{m}}$ are blocked by a collider. Hence,

$$\langle \underline{d}, \underline{m} \rangle = \lambda \langle \underline{d}, \underline{d} \rangle \tag{72.29}$$

$$\lambda = \frac{\langle \underline{d}, \underline{m} \rangle}{\langle \underline{d}, \underline{d} \rangle} = \frac{\partial \underline{m}}{\partial \underline{d}} = \rho_{\underline{d},\underline{m}} \frac{\sigma_{\underline{m}}}{\sigma_{\underline{d}}} \tag{72.30}$$

Note that also

$$\left\langle \underline{m}, \epsilon_{\underline{y}} \right\rangle = 0 \qquad |d, x \tag{72.31}$$

because the paths from $\underline{m}$ to $\epsilon_{\underline{y}}$ are blocked by a collider. Hence,

$$\left\langle \underline{m}, \underline{y} \right\rangle = \mu \langle \underline{m}, \underline{m} \rangle \qquad |d, x \tag{72.32}$$

$$\mu = \frac{\left\langle \underline{m}, \underline{y} \right\rangle}{\langle \underline{m}, \underline{m} \rangle} = \frac{\partial \underline{y}}{\partial \underline{m}} = \rho_{\underline{y},\underline{m}} \frac{\sigma_{\underline{y}}}{\sigma_{\underline{m}}} \qquad |d, x \tag{72.33}$$

Combining our newly found expressions for λ and μ, we get

$$\lambda\mu = \rho_{\underline{d},\underline{m}} \frac{\sigma_{\underline{m}}}{\sigma_{\underline{d}}} \left[\rho_{\underline{y},\underline{m}} \frac{\sigma_{\underline{y}}}{\sigma_{\underline{m}}} \right]^{|d,x} \tag{72.34}$$

Let

$$e_\lambda = \frac{\sigma_{\underline{\epsilon}_{\underline{m}}}}{\lambda \sigma_{\underline{d}}} \tag{72.35}$$

Then

$$\begin{aligned}
\sigma^2_{\underline{m}} &= \langle \underline{m}, \underline{m} \rangle && (72.36)\\
&= \left\langle \lambda \underline{d} + \underline{\epsilon}_{\underline{m}}, \lambda \underline{d} + \underline{\epsilon}_{\underline{m}} \right\rangle && (72.37)\\
&= \lambda^2 \langle \underline{d}, \underline{d} \rangle + \left\langle \underline{\epsilon}_{\underline{m}}, \underline{\epsilon}_{\underline{m}} \right\rangle && (72.38)\\
&= \lambda^2 \sigma^2_{\underline{d}} \left(1 + e^2_\lambda \right) , && (72.39)
\end{aligned}$$

and, since $[\sigma^2_{\underline{d}}]^{|d,x} = 0$, we have

$$[\sigma^{|d,x}_{\underline{m}}]^2 = \sigma^2_{\underline{\epsilon}_{\underline{m}}} \tag{72.40}$$

Hence,

$$\frac{\sigma_{\underline{m}}}{\sigma_{\underline{m}}^{|d,x}} = \frac{1}{e_\lambda}\sqrt{1+e_\lambda^2} \tag{72.41}$$

Furthermore,

$$\begin{aligned}\langle \underline{d}, \underline{m}\rangle &= \left\langle \underline{d}, \lambda \underline{d} + \epsilon_{\underline{m}}\right\rangle && (72.42)\\ &= \lambda \sigma_{\underline{d}}^2 && (72.43)\end{aligned}$$

so

$$\begin{aligned}\rho_{\underline{d},\underline{m}} &= \frac{\langle \underline{d}, \underline{m}\rangle}{\sigma_{\underline{d}}\sigma_{\underline{m}}} && (72.44)\\ &= \frac{\lambda\sigma_{\underline{d}}}{\sigma_{\underline{m}}} && (72.45)\\ &= \frac{1}{\sqrt{1+e_\lambda^2}} && (72.46)\end{aligned}$$

Hence,

$$\sqrt{1-\rho_{\underline{d},\underline{m}}^2} = \frac{e_\lambda}{\sqrt{1+e_\lambda^2}} = \left(\frac{\sigma_{\underline{m}}}{\sigma_{\underline{m}}^{|d,x}}\right)^{-1} \tag{72.47}$$

$$\lambda\mu = \frac{[\rho_{\underline{y},\underline{m}}\sigma_{\underline{y}}]^{|d,x}}{\sigma_{\underline{d}}}\frac{\rho_{\underline{d},\underline{m}}}{\sqrt{1-\rho_{\underline{d},\underline{m}}^2}} \tag{72.48}$$

QED

Note that both OVB_{con} and OVB_{med} are equal to a product $\mathcal{F}_{\underline{d}}\mathcal{F}_{\underline{y}}$ of two factors $\mathcal{F}_{\underline{d}}$ and $\mathcal{F}_{\underline{y}}$. For OVB_{con}

$$\mathcal{F}_{\underline{y}} = [\rho_{\underline{y},\underline{c}}\sigma_{\underline{y}}]^{|d,x}, \quad \mathcal{F}_{\underline{d}} = \frac{1}{[\rho_{\underline{d},\underline{c}}\sigma_{\underline{d}}]^{|x}} \tag{72.49}$$

For OVB_{med},

$$\mathcal{F}_{\underline{y}} = [\rho_{\underline{y},\underline{m}}\sigma_{\underline{y}}]^{|d,x}, \quad \mathcal{F}_{\underline{d}} = \frac{\rho_{\underline{d},\underline{m}}}{\sigma_{\underline{d}}\sqrt{1-\rho_{\underline{d},\underline{m}}^2}} \tag{72.50}$$

$\mathcal{F}_{\underline{y}}$ is the same for the confounder and mediator cases, except that the node $\underline{c}$ in $\mathcal{F}_{\underline{y}}$ for the confounder case is changed to $\underline{m}$ in the mediator case. This is to be expected, because in both cases the arrow from the unobserved node points into $\underline{y}$. On the other hand, $\mathcal{F}_{\underline{d}}$ is different for the confounder and mediator cases, because in the confounder case, the unobserved node $\underline{c}$ points into $\underline{d}$, whereas in the mediator case, $\underline{d}$ points into the unobserved node $\underline{m}$.

Chapter 73

Personalized Expected Utility

This chapter is based on Ref.[43].

This chapter assumes that the reader has already read Chapter 74 on Personalized Treatment Effects. Whereas Chapter 74 is concerned with finding bounds for PNS_z, this chapter will find bounds for EU_z, which is called the Personalized Expected Utility (PEU).

For $y_0, y_1 \in \{0, 1\}$, let us denote the **conditional joint experimental (causal, counterfactual) distribution** by:

$$P_{y_0,y_1|z} = P(\underline{y}_0 = y_0, \underline{y}_1 = y_1|z) \ . \tag{73.1}$$

Suppose we are given a **Utility function**[1]

$$\alpha_{y_0,y_1} : \{0, 1\}^2 \to \mathbb{R} \tag{73.2}$$

Let[2]

$$\begin{array}{lll} \beta = \alpha_{0,1} & (y_0 = 0, y_1 = 1) & \text{compliers} \\ \gamma = \alpha_{1,1} & (y_0 = 1, y_1 = 1) & \text{always takers} \\ \theta = \alpha_{0,0} & (y_0 = 0, y_1 = 0) & \text{never takers} \\ \delta = \alpha_{1,0} & (y_0 = 1, y_1 = 0) & \text{defiers} \end{array} \tag{73.3}$$

[1]Ref.[43] refers to the utility function as the benefit function.

[2]The notation $\beta, \gamma, \theta, \delta$ won't be used again in this chapter. We mention it here so the reader can translate to and from our equations and the equations in Ref.[43] where this $\beta, \gamma, \theta, \delta$ notation is used.

(73.4)

Define the **personalized expected utility (PEU)** by

$$EU = E[\alpha_{\underline{y}_0,\underline{y}_1}] = \alpha_{i,j} P_{i,j} \tag{73.5}$$

and the **conditional** PEU by

$$EU_z = E_{|z}[\alpha_{\underline{y}_0,\underline{y}_1}] = \alpha_{i,j} P_{i,j|z} \tag{73.6}$$

Note that

$$EU = \sum_z P(z) EU_z \ . \tag{73.7}$$

Above, we are using the Einstein summation convention (repeated indices are to be summed over) for the indices $i, j \in \{0, 1\}$.

Compare the definition of EU_z with that of two other causal effects used in his book:

$$PNS_z = P_{0,1||z} = P(compliers|z) \tag{73.8}$$

$$Uplift = ATE_z = E_{1|1,z} - E_{1|0,z} \tag{73.9}$$

As we shall see, EU_z contains the information in PNS_z and ATE_z, plus much more.

One can find the stratum z^* such that $z^* = \operatorname{argmax} ATE_z$ (as is done A/B= RCT testing) or $z^* = \operatorname*{argmax}_z EU_z$ or $z^* = \operatorname*{argmax}_z PNS_z$. This is called the **unit selection problem**. It's called "unit selection" rather than "stratum selection" because once the stratum z^* is found, one can find a unit (i.e., individual) within that stratum.

73.1 Goal of PEU Theory

Everything that we said in Chapter 74 in the section entitled "Goal of PTE Theory" applies to PEU theory too, if we just replace PNS_z by EU_z. As in Chapter 74, we

will consider two types of bounds (for EU_z instead of PNS_z): (1) Bounds for an unspecified bnet, (2) Bounds for specific bnet families.

Here is an explanation of why EU_z varies within a bounded region, something that might not be obvious to the beginner. In Fig.73.1, we represent the utility function α_{y_0,y_1} by a unit vector $\widehat{\alpha}$, the probability distribution $P_{y_0,y_1|z}$ by a vector $\vec{P}$, and EU_z by the dot product $\vec{P} \cdot \widehat{\alpha}$. When the probability vector $\vec{P}$ varies within a bounded region (shown in green), this causes the dot product $EU_z = \vec{P} \cdot \widehat{\alpha}$ to vary within a bounded interval (also shown in green).

Figure 73.1: Bounds (shown in green) on the probability vector $\vec{P}$ induce bounds (shown in green) on the dot product $EU_z = \vec{P} \cdot \widehat{\alpha}$. Here $\widehat{\alpha}$ is a unit vector that stands for a normalized utility function. $\widehat{\alpha}$ does not vary but $\vec{P}$ does.

73.2 Bnets for PEU Theory

Everything that we said in Chapter 74 in the section entitled "Bnets for PTE Theory" applies to PEU theory too, if we just replace PNS_z by EU_z.

73.3 Bounds on EU for unspecified bnet

Define the **balanced utility** by

$$\alpha_B = \alpha_{0,0} + \alpha_{1,1} \ , \tag{73.10}$$

the **unbalanced utility** by

$$\alpha_U = \alpha_{1,0} + \alpha_{0,1} \ , \tag{73.11}$$

and their difference by:

$$\sigma = \alpha_U - \alpha_B \ . \tag{73.12}$$

We will also use the abbreviations:

$$\alpha_{1-0,j} = \alpha_{1,j} - \alpha_{0,j} \tag{73.13}$$

and

$$\alpha_{j,1-0} = \alpha_{j,1} - \alpha_{j,0} \ . \tag{73.14}$$

Claim 103 *In general,*

$$\begin{cases} \max p_{[1...4]} \leq EU_z \leq \min p_{[5...8]} & \textit{if } \sigma < 0 \\ \max p_{[5...8]} \leq EU_z \leq \min p_{[1...4]} & \textit{if } \sigma > 0 \end{cases} \tag{73.15}$$

where[3]

$$\begin{cases} p_1 = \alpha_{0,1-0} E_{1|1,z} + \alpha_{j,0} E_{j|0,z} \\ p_2 = \alpha_{0-1,1} E_{0|0,z} + \alpha_{1,j} E_{j|1,z} \\ p_3 = p_5 + \sigma O_{*,*|z} \\ p_4 = p_1 - \sigma E_{1|0,z} + \sigma(1 - O_{*,*|z}) \end{cases} \tag{73.16}$$

$$\begin{cases} p_5 = \alpha_{1,1-0} E_{1|1,z} + \alpha_{j,0} E_{j|0,z} \\ p_6 = p_1 - \sigma E_{1|0,z} \\ p_7 = p_5 - \sigma E_{1|0,z} + \sigma P(\underline{y} = 1|z) \\ p_8 = p_1 - \sigma P(\underline{y} = 1|z) \end{cases} \tag{73.17}$$

proof: See Ref[43] or use the MRP (Matrix Representation of Probabilities) defined in Chapter 74.
QED

Later on, we will see that under monotonicity, these bounds collapse to a point estimate of EU_z. But what if monotonicity doesn't hold? In such cases, one can use the midpoint of the bounds as a non-rigorous point estimate of EU_z.

Claim 104 *In general,*

$$P_{0,0|z} = E_{0|1,z} - P_{1,0|z} \tag{73.18}$$

$$P_{1,1|z} = E_{1|0,z} - P_{1,0|z} \tag{73.19}$$

$$\begin{aligned} P_{0,1|z} &= 1 - P_{0,0|z} - P_{1,1|z} - P_{1,0|z} && (73.20) \\ &= 1 - E_{0|1,z} - E_{1|0,z} + P_{1,0|z} && (73.21) \end{aligned}$$

proof:

$$P_{0,0|z} + P_{1,0|z} = P(\underline{y}_1 = 0|z) = E_{0|1,z} \tag{73.22}$$

[3] $p_{[a...b]} = (p_a, p_{a+1}, \ldots p_b)$ for integers a, b such that $a < b$.

$$P_{1,1|z} + P_{1,0|z} = P(\underline{y}_0 = 1|z) = E_{1|0,z} \tag{73.23}$$

QED

Recall the definition of monotonicity. Monotonicity holds iff

$$P(\underline{y}_0 = 1, \underline{y}_1 = 0) = 0 \ . \tag{73.24}$$

Claim 105 *In general,*

$$EU_z = \alpha_{0,0}P_{0,0|z} + \alpha_{1,1}P_{1,1|z} + \alpha_{0,1}P_{0,1|z} + \alpha_{1,0}P_{1,0|z} \tag{73.25}$$

Hence, if monotonicity holds, or $\alpha_{1,0} = 0$, then

$$EU_z = \alpha_{0,0}P_{0,0|z} + \alpha_{1,1}P_{1,1|z} + \alpha_{0,1}P_{0,1|z} \tag{73.26}$$

proof: Trivial
QED

Claim 105 is trivial, but it provides a good motivation and inspiration for the following less trivial claim.

Claim 106 *In general,*

$$EU_z = \alpha_{0,0} \underbrace{E_{0|1,z}}_{=1-E_{1|1,z}} + \alpha_{1,1}E_{1|0,z} + \alpha_{0,1} \underbrace{(1 - E_{0|1,z} - E_{1|0,z})}_{=E_{1|1,z}-E_{1|0,z}=ATE_z} + \sigma P_{1,0|z} \tag{73.27}$$

Hence, if monotonicity holds, or $\sigma = 0$, then

$$(EU_z)_{\sigma=0} = \alpha_{0,0}E_{0|1,z} + \alpha_{1,1}E_{1|0,z} + \alpha_{0,1}ATE_z \tag{73.28}$$

proof: In general,

$$EU_z = \left\{ \begin{array}{c} \alpha_{0,0}P_{0,0|z} \\ +\alpha_{1,1}P_{1,1|z} \\ +\alpha_{0,1}P_{0,1|z} \\ +\alpha_{1,0}P_{1,0|z} \end{array} \right. = \left\{ \begin{array}{c} \alpha_{0,0}(E_{0|1,z} - P_{1,0|z}) \\ +\alpha_{1,1}(E_{1|0,z} - P_{1,0|z}) \\ +\alpha_{0,1}(1 - E_{0|1,z} - E_{1|0,z} + P_{1,0|z}) \\ +\alpha_{1,0}P_{1,0|z} \end{array} \right. \tag{73.29}$$

Hence,

$$EU_z = (EU_z)_{\sigma=0} + \sigma P_{1,0|z} \tag{73.30}$$

QED

Perhaps this will make Claim 106 less mysterious to you. Note that Claims 105 and 106 imply the following:

$$\left[\frac{\partial EU_z}{\partial \alpha_{1,0}}\right]_{\alpha_{0,0},\alpha_{1,1},\alpha_{0,1}} = \left[\frac{\partial EU_z}{\partial \sigma}\right]_{\alpha_{0,0},\alpha_{1,1},\alpha_{0,1}} = P_{1,0|z} \tag{73.31}$$

73.4 Bounds on EU for specific bnet families

Chapter 74

Personalized Treatment Effects

This chapter is based on the work of Pearl et al, as reported in Refs. [84] and [49].

Recall from Chapter 77 on Potential Outcomes (PO) and Beyond, that the **Average Treatment Effect (ATE)** is defined as

$$\begin{aligned} ATE &= E[\underline{y}_1 - \underline{y}_0] && (74.1) \\ &= P(\underline{y}_1 = 1) - P(\underline{y}_0 = 1) && (74.2) \end{aligned}$$

The **Conditional ATE (CATE)** is defined as the conditional expected value

$$\begin{aligned} ATE_z &= E_{|z}[\underline{y}_1 - \underline{y}_0] && (74.3) \\ &= P(\underline{y}_1 = 1|z) - P(\underline{y}_0 = 1|z) \ . && (74.4) \end{aligned}$$

Note that

$$ATE = \sum_z P(z) ATE_z \tag{74.5}$$

Personalized Treatment Effect (PTE) theory as envisioned by Pearl is the study of bounds for "personalized" treatment effects such as

$$PNS = P(\underline{y}_1 - \underline{y}_0 = 1) \tag{74.6}$$

and

$$PNS_z = P(\underline{y}_1 - \underline{y}_0 = 1|z) \ . \tag{74.7}$$

They are said to be **personalized** because they are averages over a *single* ensemble (i.e., population) of individuals σ with probability $P(y_0^\sigma, y_1^\sigma)$. ATE, on the other hand, is not personalized, because it equals the difference of two of those averages. Personalized effects equal the probability of a single event/person, whereas ATE equals the difference of the probabilities of 2 events/persons.

If the conditioning z is fine grained enough to pick out a single individual σ (i.e., if $z = \sigma$), then we get

$$PNS_\sigma = \mathbb{1}(y_1^\sigma - y_0^\sigma = 1) \tag{74.8}$$

whereas

$$ATE_\sigma = \mathbb{1}(y_1^\sigma = 1) - \mathbb{1}(y_0^\sigma = 1) \tag{74.9}$$

ATE and PNS measure different things. PNS measures the probability that a *single person* will switch outcomes from 0 to 1 when he/she switches treatments from 0 to 1. ATE measures the difference in populations between those who survive taking the drug and those who survive without it.

One very promising field in which PTE theory can be applied is in **Personalized Causal Medicine**. For example, suppose we want to use PNS_z, where the conditioning is on the sex of the patient (i.e., $z = male, female$), to advice a female patient whether to take a cancer drug or not.

74.1 Goal, Strategy and Rationale of PTE theory

In this section, we described briefly the goal,strategy and rationale behind PTE theory.

Figure 74.1: We use patient data to calculate at each step, increasingly tighter bounds b_j, b'_j, B_j, B'_j for PNS_z and ATE_z, ending in point bounds for both quantities.

The goal of PTE theory, as described by Fig.74.1, is to find increasingly tighter bounds for PTEs such as PNS, PNS_z, ATE, ATE_z, etc..

We say bounds, because it is not always possible to give a point estimate for PNS and other PTEs. If a point estimate for a PTE named Q is achievable, we say Q is **identifiable**. The same definition of identifiability has been used before in this

book for ATE. It's possible that ATE is identifiable, but PNS isn't, or vice versa, for certain kinds of data; i.e., both quantities need not become identifiable simultaneous at the same step above.

The bounds given by PTE theory are as tight as possible, depending on the available data, and on what bnet model assumptions the user is willing to make. We will consider two types of bounds: (1) Bounds for an unspecified bnet, (2) bounds for specific bnet families. Bounds for (2) will be tighter than bounds for (1).

The bounds are calculated from two types of data: **Observational Data (OD)** and **Experimental Data (ED)**.

For OD, one allows the patient to choose whether to take a drug or not ($x = 0, 1$), and then we conduct a **survey** to record his/her value for x and whether the treatment worked or not ($y = 0, 1$).

For ED, one conducts a **RCT** (Randomized Controlled Trial) instead of a survey. In the case of ED, we record, as in OD, the (x, y) for each patient, but the value of x for each patient is selected by the experimenter, at random, and, once selected, it is compulsory for the patient.

Unlike ED, OD is likely to be confounded, but it can still shed additional information that serves to tighten the bounds on PNS_z or other PTEs.

OD is usually collected first because it is easier and cheaper to collect than ED. Sometimes ED is too expensive or difficult or even impossible to collect, so only OD is available. Sometimes several ED (resp., several OD) need to be merged, before merging the merged ED with the merged OD. Pearl's PTE theory allows us to fuse together all the available data in all these types of situations.

Some purists such as the advocates of **EBM (Evidence Based Medicine)** advocate the use of only ED (i.e., only RCTs), no OD. This is reasonable in certain administrative professions to keep partisanship and chicanery out of the decision making process, but in other professions, throwing away OD would be foolish. A case in the history of medicine where OD was essential, was the case of John Snow (see Chapter 18). John Snow didn't have an RCT, but he was able to use OD to determine the driver of the cholera epidemic in London. He saved countless lives by doing so. An EBM purist would have thrown out his OD because it wasn't an RCT.

In the usual case, OD is collected first to aid in the design of an RCT, and then an RCT is conducted to collect ED. In this case, an EBM purist would throw away the OD, and only calculate an estimate of ATE. What PTE theory suggests is to calculate *both*, an estimate of ATE and bounds for PNS. Why? Because ATE and PNS measure different things. ATE utilizes only ED whereas PNS utilizes both OD and ED. Also, PNS is more personalized than ATE.

An RCT is called that because one chooses "at random", a subset Σ of a huge population Σ_∞, and then one splits Σ, again at random, into 2 subset, $\Sigma_{control}$ and $\Sigma_{treated}$. All patients in $\Sigma_{control}$ (resp., $\Sigma_{treated}$) are not treated (resp., treated). **Randomization** (i.e. choosing Σ, $\Sigma_{control}$, $\Sigma_{treated}$), if perfect, averages out (i.e., cancels out) the effect of all confounders. The problem is that in practice, ideal randomiza-

tion is hard to achieve. Hence, ideal RCTs are hard to achieve (and costly), plus they don't shed light on the confounders and mechanism involved. By mechanism, I mean a DAG.[1] A DAG is a hypothesis, and RCTs don't test it, whereas observational studies do. So it's a good idea to do a series of well designed observational studies (much cheaper than an RCT), with a hypothesis DAG, before doing the RCT. This elucidates the DAG, and helps design an RCT that achieves a good approximation to randomization.

Some people object to CI (Causal Inference) in general on the grounds that the causal DAGs are adhoc, arbitrary, a sort of unscientific voodoo. I think it's because they fail to grasp the true meaning and purpose of DAGs. I discuss this further in Section G.6.

74.2 Bnets for PTE theory

Let $\overline{0} = 1$ and $\overline{1} = 0$.

Whenever we write $P(\underline{a} = x, b)$, we mean $P(\underline{a} = x, \underline{b} = b)$.

In this chapter, we will not use the notation $P(y)$ and $P(y')$ used by Pearl to discuss PTE theory. Instead, we will use $P(\underline{y} = 1)$ and $P(\underline{y} = 0)$ to denote his $P(y)$ and $P(y')$, respectively.

On the other hand, in this chapter we will change the names of variables $\underline{d}, \underline{y}, \underline{z}, \underline{y}(d)$ used in Chapter 77 on Potential Outcomes (PO) to the names favored by Pearl. Hence, we will replace $\underline{d}, \underline{y}, \underline{x}, \underline{y}(d)$ by $\underline{x}, \underline{y}, \underline{z}, \underline{y}_x$, respectively.

PTE theory considers bnets of the form Fig.74.2. The bnet considered in Rubin's PO theory is a very simple special case of this where the box labeled "multiple nodes" is absent and there is an arrow pointing from $\underline{z}$ to $\underline{x}$.

The TPM, printed in blue, for node $\underline{y}$ of bnet Fig.74.2 is as follows:

$$P(y|y_0, y_1, x) = \delta(y, y_x) \tag{74.10}$$

This TPM is used frequently in PTE theory. If x, y_0, y_1 are arguments of $P()$, this TPM implies that one can swap $\underline{y}_x = y_x$ and $\underline{y} = y_x$ inside $P()$. For example,

$$\begin{aligned} P(y_0, y_1, x) &= P(\underline{y}_x = y_x, y_{\overline{x}}, x) && (74.11) \\ &= P(\underline{y} = y_x, y_{\overline{x}}, x) && (74.12) \end{aligned}$$

According to Pearl, the defining property of y_x is that

$$P(\underline{y}_x = y) = P(\underline{y} = y|\mathcal{D}\underline{x} = x) \tag{74.13}$$

[1]Note that DAGs are not unique. Some are a better causal fit than others for the physical situation being considered. See Chapter 37 for a discussion of Goodness of Causal Fit.

Figure 74.2: Type of Bnet considered in PTE theory. The box labeled "multiple nodes" contains various observed and hidden nodes with arrows to or from node $\underline{x}$ and to or from node $\underline{z}$. $\underline{z}$ can be a multinode. $\underline{z}$ is shown as hidden but could be observed instead.

Pearl likes to call Eq.(74.13) the 1st Law. The 1st Law is also a consequence of bnet Fig.74.2 and the TPM Eq.(74.10). Indeed, $\mathcal{D}\underline{x} = x$ means one should amputate all arrows entering node $\underline{x}$, and one should set the TPM of $\underline{x}$ to a delta function centered at x. If that is done, then the values of $\underline{y}$ and $\underline{y}_x$ must be equal, because the TPM at node $\underline{y}$ is a delta function that enforces this equality.

74.3 $ATE = PB - PH$

Define the **probability of benefit (PB)** (a.k.a. **Probability of Necessity and Sufficiency (PNS)**) by

$$PB = PNS = P(\underline{y}_0 = 0, \underline{y}_1 = 1) \tag{74.14}$$

and the **probability of harm (PH)** by

$$PH = P(\underline{y}_0 = 1, \underline{y}_1 = 0) \tag{74.15}$$

Claim 107

$$PB = P(\underline{y}_1 - \underline{y}_0 = 1) \tag{74.16}$$

$$PH = P(\underline{y}_1 - \underline{y}_0 = -1) \tag{74.17}$$

proof:

$\underline{y}_1 - \underline{y}_0 = 1$ iff ($\underline{y}_1 = 1$ and $\underline{y}_0 = 0$).
$\underline{y}_1 - \underline{y}_0 = -1$ iff ($\underline{y}_1 = 0$ and $\underline{y}_0 = 1$).

QED

Claim 108

$$ATE = PB - PH \tag{74.18}$$

proof:

$$
\begin{aligned}
ATE &= E_\sigma[y_1^\sigma - y_0^\sigma] && (74.19)\\
&= E[\underline{y}_1 - \underline{y}_0] && (74.20)\\
&= \sum_y y[P(\underline{y}_1 = y) - P(\underline{y}_0 = y)] && (74.21)\\
&= P(\underline{y}_1 = 1) - P(\underline{y}_0 = 1) && (74.22)\\
&= \sum_{y_0} P(y_0, \underline{y}_1 = 1) - \sum_{y_1} P(\underline{y}_0 = 1, y_1) && (74.23)\\
&= \underbrace{P(\underline{y}_0 = 0, \underline{y}_1 = 1)}_{PB} - \underbrace{P(\underline{y}_0 = 1, \underline{y}_1 = 0)}_{PH} && (74.24)
\end{aligned}
$$

QED

See Fig.74.3 for an illustration of the constant ATE contours in the (PB, PH) plane.

See also Fig.74.4 for a vector representation of the identity $ATE = PB - PH$.

See also Fig.74.5 for an illustration of the region of possible points in the (ATE, PB) plane.

Figure 74.3: Shown in pink, the probability simplex $\{(x, y) : x \geq 0, y \geq 0, x + y \leq 1\}$ with $x = PB$ and $y = PH$. All points of that simplex are possible. Also shown are the constant ATE contours in the (PB, PH) plane.

74.4 Probabilities Relevant to PTE theory

Note[2]

[2]To translate this section from our notation to the notation used by Tian and Pearl in Ref.[84], replace $P(\underline{y}_1 = i, \underline{y}_0 = j, \underline{x} = k) \to p_{i,j,k}$ $O_{1,0} \to P(x, y')$, $O_{0|1} \to P(y'|x)$, $E_{0|1} \to P(y'_x)$, etc.

Figure 74.4: The identity $ATE = PB - PH$ can be visualized as a vector sum in one dimension, where $PB, PH \in [0, 1]$ and $ATE \in [-1, 1]$.

Figure 74.5: Shown in pink, the region of possible points in the (ATE, PB) plane. $\max(0, ATE) \leq PB \leq (ATE + 1)/2$.

Let $x, y \in \{0, 1\}$ and $z \in val(\underline{z})$ for some finite, not necessarily binary set $val(\underline{z})$. Define

$$\mathcal{P}_{x',y'|x,y} = P(\underline{y}_{x'} = y'|x, y) \tag{74.25}$$

observational (non-causal) probabilities

$$O_{x,y} = P(\underline{x} = x, \underline{y} = y) \tag{74.26}$$

$$O_{y|x} = P(\underline{y} = y|\underline{x} = x) \tag{74.27}$$

$$\pi_x = P(\underline{x} = x) \tag{74.28}$$

experimental (causal) probabilities

$$E_{y|x} = P(\underline{y}_x = y) \tag{74.29}$$

Average Treatment Effect (ATE)

$$\begin{aligned} ATE &= P(\underline{y}_1 = 1) - P(\underline{y}_0 = 1) && (74.30) \\ &= E_{1|1} - E_{1|0} && (74.31) \\ &= E_{1|1} + E_{0|0} - 1 && (74.32) \end{aligned}$$

Conditional ATE (CATE)

$$\begin{aligned} ATE_z &= P(\underline{y}_1 = 1|z) - P(\underline{y}_0 = 1|z) && (74.33) \\ &= E_{1|1,z} + E_{0|0,z} - 1 && (74.34) \end{aligned}$$

Average Causal Effect

$$ACE = P(\underline{y} = 1|\mathcal{D}\underline{x} = 1) - P(\underline{y} = 1|\mathcal{D}\underline{x} = 0) \tag{74.35}$$

Note that

$$ACE = \underbrace{P(\underline{y} = 1|\mathcal{D}\underline{x} = 1)}_{P(\underline{y}_1=1)} - \underbrace{P(\underline{y} = 1|\mathcal{D}\underline{x} = 0)}_{P(\underline{y}_0=1)} = ATE \tag{74.36}$$

Conditional ACE (CACE)
This is what is called ACE_z in Chapter 77. Note that $ACE_z = ATE_z$.
Effect of Treatment on the Treated (ETT) (i.e., ATE for the treated)

$$ETT = \underbrace{P(\underline{y}_1 = 1|\underline{x} = 1)}_{\mathcal{E}_1} - \underbrace{P(\underline{y}_0 = 1|\underline{x} = 1)}_{\mathcal{E}_0} \tag{74.37}$$

Note that

$$\begin{aligned} \mathcal{E}_1\pi_1 &= P(\underline{y}_1 = 1, \underline{x} = 1) && (74.38) \\ &= O_{1,1} && (74.39) \end{aligned}$$

and

$$\begin{aligned} \mathcal{E}_0\pi_1 &= P(\underline{y}_0 = 1, \underline{x} = 1) && (74.40) \\ &= P(\underline{y}_0 = 1) - \underbrace{P(\underline{x} = 0, \underline{y}_0 = 1)}_{P(\underline{x}=0,\underline{y}=1)} && (74.41) \\ &= E_{1|0} - O_{0,1} && (74.42) \end{aligned}$$

so

$$ETT\pi_1 = \sum_x O_{x,1} - E_{1|0} \tag{74.43}$$

Probability of Necessity (PN)[3]

$$PN = \mathcal{P}_{0,0|1,1} \tag{74.44}$$

[3] I like to call PN the Probability of Nullifying, because it goes from 11 to 00

Probability of Sufficiency (PS)[4]

$$PS = \mathcal{P}_{1,1|0,0} \tag{74.45}$$

Probability of Necessity and Sufficiency (PNS) (a.k.a. Probability of Benefit (PB))

$$PNS = PB = P(\underline{y}_0 = 0, \underline{y}_1 = 1) \tag{74.46}$$

Henceforth, we will use $PNS3$ to denote the trio

$$PNS3 = (PNS, PN, PS)\ . \tag{74.47}$$

Probability of Harm (PH)

$$PH = P(\underline{y}_0 = 1, \underline{y}_1 = 0) \tag{74.48}$$

Risk ratio or relative risk (RR)

$$RR = \frac{O_{1|1}}{O_{1|0}} \tag{74.49}$$

Excess Risk Ratio (ERR)

$$ERR = \frac{O_{1|1} - O_{1|0}}{O_{1|1}} = 1 - \frac{1}{RR} \tag{74.50}$$

Corrected ERR (CERR)

$$CERR = ERR + \frac{O_{1|0} - E_{1|0}}{O_{1,1}} \tag{74.51}$$

You might be wondering how $P(\underline{y}_x = y|\underline{x} = 0)$ and $P(\underline{y}_x = y|\underline{x} = 1)$ for $x, y \in \{0,1\}^2$ are related to $O_{y|x}$ and $E_{y|x}$. The following claim shows how.

Claim 109

$$P(\underline{y}_x = y|\underline{x} = x) = O_{y|x} \tag{74.52a}$$

$$P(\underline{y}_x = y|\underline{x} = \overline{x}) = \frac{E_{y|x} - O_{y|x}\pi_x}{\pi_{\overline{x}}} \tag{74.52b}$$

proof: Before we begin the proof, note that summing both sides of Eqs.(74.52) gives $1 = 1$, so these 2 equations pass that test.

$$P(\underline{y}_x = y|\underline{x} = x) = P(\underline{y} = y|\underline{x} = x) = O_{y|x} \tag{74.53}$$

[4] I like to call PS the Probability of Surging, because it goes from 00 to 11

$$
\begin{aligned}
P(\underline{y}_x = y|\underline{x} = \overline{x}) &= \frac{P(\underline{y}_x = y, \underline{x} = \overline{x})}{\pi_{\overline{x}}} && (74.54)\\
&= \frac{P(\underline{y}_x = y) - P(\underline{y}_x = y, \underline{x} = x)}{\pi_{\overline{x}}} && (74.55)\\
&= \frac{P(\underline{y}_x = y) - P(\underline{y}_x = y|\underline{x} = x)\pi_x}{\pi_{\overline{x}}} && (74.56)\\
&= \frac{E_{y|x} - O_{y|x}\pi_x}{\pi_{\overline{x}}} && (74.57)
\end{aligned}
$$

QED

Claim 110

$$
\begin{aligned}
P(y_0, y_1|x) &= P(y_{\overline{x}}|x, \underline{y} = y_x)P(\underline{y} = y_x|x) && (74.58)\\
&= \begin{tikzcd} & y_{\overline{x}} \\ x \arrow[ur] \arrow[r] & \underline{y} = y_x \arrow[u] \end{tikzcd} && (74.59)
\end{aligned}
$$

proof:

$$
\begin{aligned}
P(y_0, y_1|x) &= \frac{P(y_0, y_1, x)}{P(x)} && (74.60)\\
&= \frac{P(y_{\overline{x}}, x, \underline{y} = y_x)}{P(x)} && (74.61)\\
&= \frac{P(y_{\overline{x}}|x, \underline{y} = y_x)P(x, \underline{y} = y_x)}{P(x)} && (74.62)\\
&= P(y_{\overline{x}}|x, \underline{y} = y_x)P(\underline{y} = y_x|x) && (74.63)
\end{aligned}
$$

QED

Claim 111

$$
\underbrace{PNS}_{P(\underline{y}_0=0, \underline{y}_1=1)} = PN * O_{1,1} + PS * O_{0,0} \qquad (74.64)
$$

Hence, if we know any two of (PN, PS, PNS)*, we can calculate the third.*

proof:

$$
\begin{aligned}
P(y_0, y_1) &= \sum_x P(y_0, y_1|x)P(x) && (74.65)\\
&= \sum_x P(y_{\overline{x}}|x, \underline{y} = y_x)P(x, \underline{y} = y_x) \quad \text{(see Claim 110.)} && (74.66)\\
&= \begin{cases} P(y_1|\underline{x} = 0, \underline{y} = y_0)P(\underline{x} = 0, \underline{y} = y_0) \\ +P(y_0|\underline{x} = 1, \underline{y} = y_1)P(\underline{x} = 1, \underline{y} = y_1) \end{cases} && (74.67)
\end{aligned}
$$

Thus,

$$P(\underline{y}_0 = 0, \underline{y}_1 = 1) = \left\{ \begin{array}{c} P(\underline{y}_1 = 1|\underline{x} = 0, \underline{y} = 0)P(\underline{x} = 0, \underline{y} = 0) \\ +P(\underline{y}_0 = 0|\underline{x} = 1, \underline{y} = 1)P(\underline{x} = 1, \underline{y} = 1) \end{array} \right. \tag{74.68}$$

$$= PS * O_{0,0} + PN * O_{1,1} \tag{74.69}$$

QED

Note that PNS refers to both $\underline{y}_0$ and $\underline{y}_1$, PN refers only to $\underline{y}_0$ and PS refers only to $\underline{y}_1$. Thus, PN and PS serve to separate the pair of variables $(\underline{y}_0, \underline{y}_1)$ and to isolate them individually. Claim 111 is a quantitative expression of that separation.

Pearl likes to say that PNS belongs to Rung 3 because it's a probability that involves both $\underline{y}_0$ and $\underline{y}_1$, and one of those two must be a counterfactual (an event that never occurred). On the other hand, PN, PS, ATE, etc., are defined in terms of probabilities that involve either $\underline{y}_0$ or $\underline{y}_1$ but not both. Probabilities that involve only one of them, can be expressed with the do operator, so they belong to Rung 2. Conditional probabilities like $P(y|x)$ that involve neither $\underline{y}_0$ nor $\underline{y}_1$ belong to Rung 1.[5]

74.5 Symmetry

Define $\sim$ to be an operator that swaps zeros and ones in $P(\underline{y}_0 = y, \underline{y}_1 = y', x)$. Hence

$$[P(\underline{y}_0 = y, \underline{y}_1 = y', \underline{x} = x)]^{\sim} = P(\underline{y}_1 = \overline{y}, \underline{y}_0 = \overline{y'}, \underline{x} = \overline{x}) \tag{74.70}$$

$$(\mathcal{P}_{x',y'|x,y})^{\sim} = \mathcal{P}_{\overline{x'},\overline{y'}|\overline{x},\overline{y}} \tag{74.71}$$

$$(O_{x,y})^{\sim} = O_{\overline{x},\overline{y}} \tag{74.72}$$

$$(O_{y|x})^{\sim} = O_{\overline{y}|\overline{x}} \tag{74.73}$$

$$(\pi_x)^{\sim} = \pi_{\overline{x}} \tag{74.74}$$

$$(E_{y|x})^{\sim} = E_{\overline{y}|\overline{x}} \tag{74.75}$$

$$(PN)^{\sim} = PS, \quad (PS)^{\sim} = PN \tag{74.76}$$

[5]Probabilities such as $P(\underline{y}_0 = 1, \underline{y} = 0, \underline{x} = 1) = P(\underline{y}_0 = 1, \underline{y}_1 = 0, \underline{x} = 1)$ are considered Rung 3.

$$(PNS)^{\sim} = PNS \tag{74.77}$$

$$(RR)^{\sim} = \frac{O_{0|0}}{O_{0|1}} \tag{74.78}$$

Recall

$$ERR = \frac{O_{1|1} - O_{1|0}}{O_{1|1}} = 1 - \frac{1}{RR} \tag{74.79}$$

Therefore, define

$$(ERR)^{\sim} = \frac{O_{0|0} - O_{0|1}}{O_{0|0}} = 1 - \frac{1}{(RR)^{\sim}} \tag{74.80}$$

Note that

$$\begin{aligned} O_{1|1} - O_{1|0} &= O_{1|1} + O_{0|0} - 1 && (74.81)\\ &= O_{0|0} - O_{0|1} && (74.82)\\ &= (O_{1|1} - O_{1|0})^{\sim} && (74.83) \end{aligned}$$

74.6 Linear Programming Problem

Probability Simplex

$$\mathcal{S} = \left\{ P(y_0, y_1, x) : \begin{array}{c} P(y_0, y_1, x) \geq 0 \\ y_0, y_1, x \in \{0, 1\} \\ \sum_{y_0=0}^{1} \sum_{y_1=0}^{1} \sum_{x=0}^{1} P(y_0, y_1, x) = 1 \end{array} \right\} \tag{74.84}$$

Note that $\mathcal{S}$ has 7 degrees of freedom (dofs).

1. **observational (non-causal) constraints on $\mathcal{S}$** (3 constraints)

$$O_{x,y} = P(x, y) = \sum_{y'} P(\underline{y}_x = y, \underline{y}_{\overline{x}} = y', x) \quad \text{for } (x, y) \in \{(0, 1), (1, 0), (1, 1)\} \tag{74.85}$$

2. **experimental (causal) constraints on $\mathcal{S}$** (2 constraints)[6]

$$E_{1|x} = P(\underline{y}_x = 1) = \sum_{x'} \sum_{y'} P(\underline{y}_x = 1, \underline{y}_{\overline{x}} = y', \underline{x} = x') \quad \text{for } x \in \{0, 1\} \tag{74.86}$$

[6] $E_{0|x}$ follows from $E_{0|x} = 1 - E_{1|x}$.

$\mathcal{S}$ has 7 dofs but the 3 observational constraints reduce the number of dofs to 4. The 5 observational and experimental constraints reduce the number of dofs to 2. $\mathcal{S}$ is embedded in $\mathbb{R}^8$ and then the 6 constraints (unit probability, 2 observational and 3 experimental) reduce it to the interior of a 6 or less sided figure in $\mathbb{R}^2$.

Henceforth, we will refer to the observational and experimental constraints together as the **minimal constraints**.

This is half of a linear programming problem. Recall that a **linear programming problem** can be stated as finding the column vector ξ that minimizes a cost $\mathcal{C} = c^T\xi$ subject to $A\xi = b$ and $\xi \geq 0$. Here we have no cost function or minimization, but we have $A\xi = b$ and $\xi \geq 0$ where $\xi = [P(y_0, y_1, x)]_{\forall y_0, y_1, x}$. Also $b = [1, O_{0,1}, O_{1,0}, O_{1,1}, E_{1|0}, E_{1|1}]^T$, and $A =$ a matrix of zeros and ones.

Note that

$$PNS = P(\underline{y}_0 = 0, \underline{y}_1 = 1) \tag{74.87}$$

$$= \sum_x P(\underline{y}_0 = 0, \underline{y}_1 = 1, x) \tag{74.88}$$

$$PN = P(\underline{y}_0 = 0 | \underline{x} = 1, \underline{y} = 1) \tag{74.89}$$

$$= \frac{P(\underline{y}_0 = 0, \underline{y}_1 = 1, \underline{x} = 1)}{O_{1,1}} \tag{74.90}$$

$$PS = P(\underline{y}_1 = 1 | \underline{x} = 0, \underline{y} = 0) \tag{74.91}$$

$$= \frac{P(\underline{y}_0 = 0, \underline{y}_1 = 1, \underline{x} = 0)}{O_{0,0}} \tag{74.92}$$

74.7 Special constraints

- **Exogeneity (a.k.a. no-confounding)** holds for simplex $\mathcal{S}$ if $\underline{y}_x \perp \underline{x}$ for $x \in \{0, 1\}$. Hence

$$\underbrace{P(\underline{y}_x = 1)}_{E_{1|x}} = P(\underline{y}_x = 1 | x) = \underbrace{P(\underline{y} = 1 | x)}_{O_{1|x}} \quad \text{for } x \in \{0, 1\} \tag{74.93}$$

  Exogeneity gives 2 constraints.

  Note that exogeneity is the same thing as identifiability of $P(\underline{y}_x = y) = P(\underline{y} = y | \mathcal{D}\underline{x} = x)$. But identifiability (i.e., do-identifiability) is a more general concept. One can speak of the identifiability of $P(y_x|z)$ or of $P(y_0, y_1, a)$, etc. In general, any expression with do operators is do-identifiable if it can be expressed as an expression without do operators.

- **Strong Exogeneity** holds for simplex $\mathcal{S}$ if $(\underline{y}_0, \underline{y}_1)_{joint} \perp \underline{x}$. Hence

$$P(y_0, y_1|x) = P(y_0, y_1) \tag{74.94}$$

Strong exogeneity gives 3 constraints. Strong exogeneity implies exogeneity but not the converse.[7]

- **Monotonicity**[8] holds for simplex $\mathcal{S}$ if

$$PH = P(\underline{y}_0 = 1, \underline{y}_1 = 0) = 0 \tag{74.95}$$

. Equivalently,

$$\sum_x P(\underline{y}_0 = 1, \underline{y}_1 = 0, x) = 0 \tag{74.96}$$

which is true iff

$$P(\underline{y}_0 = 1, \underline{y}_1 = 0, x) = 0 \quad \text{for } x \in \{0, 1\} \tag{74.97}$$

Monotonicity gives 2 constraints.

Note that when $PH = 0$, $PNS = ATE$.

Claim 112 *Monotonicity and exogeneity together imply strong exogeneity.*

proof:

This proof is presented here for completeness. Later on, we will give a much simpler proof of this result. I advise the reader to skip this proof on first reading of this chapter.

Let $a \in \{0, 1\}$.

$$\begin{aligned} P(\underline{y}_a = \overline{a}, x) &= \sum_{a'} P(\underline{y}_a = \overline{a}, \underline{y}_{\overline{a}} = a', x) && (74.98) \\ &= P(\underline{y}_a = \overline{a}, \underline{y}_{\overline{a}} = \overline{a}, x) \quad \text{(by monotonicity)} && (74.99) \end{aligned}$$

Thus,

$$P(\underline{y}_a = \overline{a}|\underline{x} = a) = P(\underline{y}_a = \overline{a}, \underline{y}_{\overline{a}} = \overline{a}|\underline{x} = a) \tag{74.100}$$

and

$$P(\underline{y}_a = \overline{a}) = P(\underline{y}_a = \overline{a}, \underline{y}_{\overline{a}} = \overline{a}) \tag{74.101}$$

[7]

[8] This property is called monotonicity because it's equivalent to the statement that $y_0^\sigma \leq y_1^\sigma$ for all individuals σ in the population. Indeed, $y_0^\sigma \leq y_1^\sigma$ includes the 3 cases $(y_0^\sigma, y_1^\sigma) = (0, 0), (1, 1), (0, 1)$, but excludes the only other case, namely $(1, 0)$.

By exogeneity, the left hand side of Eq.(74.100) and the left hand side of Eq.(74.101) are equal, so the right hand sides of those equations must be equal too.

$$P(\underline{y}_a = \overline{a}, \underline{y}_{\overline{a}} = \overline{a} | \underline{x} = a) = P(\underline{y}_a = \overline{a}, \underline{y}_{\overline{a}} = \overline{a}) \tag{74.102}$$

This immediately implies[9]

$$P(\underline{y}_a = \overline{a}, \underline{y}_{\overline{a}} = \overline{a} | x) = P(\underline{y}_a = \overline{a}, \underline{y}_{\overline{a}} = \overline{a}) \tag{74.103}$$

for $x \in \{0, 1\}$. This gives for $a = 0$

$$P(\underline{y}_0 = 1, \underline{y}_1 = 1 | x) = P(\underline{y}_0 = 1, \underline{y}_1 = 1) \tag{74.104}$$

and for $a = 1$

$$\underbrace{P(\underline{y}_1 = 0, \underline{y}_0 = 0 | x)}_{P(\underline{y}_0 = 0, \underline{y}_1 = 0 | x)} = \underbrace{P(\underline{y}_1 = 0, \underline{y}_0 = 0)}_{P(\underline{y}_0 = 0, \underline{y}_1 = 0)} \tag{74.105}$$

Monotonicity itself gives

$$P(\underline{y}_0 = 1, \underline{y}_1 = 0 | x) = 0 = P(\underline{y}_0 = 1, \underline{y}_1 = 0) \tag{74.106}$$

The remaining strong exogeneity constraint, given by

$$P(\underline{y}_0 = 0, \underline{y}_1 = 1 | x) = P(\underline{y}_0 = 0, \underline{y}_1 = 1) \ , \tag{74.107}$$

follows because

$$\sum_{y_0, y_1} P(y_0, y_1 | x) = \sum_{y_0, y_1} P(y_0, y_1) = 1 \tag{74.108}$$

QED

74.8 Matrix representation of probabilities

Suppose $\epsilon_x(y_0, y_1)$ for $x, y_0, y_1 \in \{0, 1\}$ are eight vectors in a vector space V. $\epsilon_x(y_0, y_1)$ will only be used at position (y_0, y_1) of a 2×2 matrix, where the positions are defined as follows:

$$\begin{array}{lll} y_1 & & \\ \uparrow & & \\ 1 & (0,1) & (1,1) \\ | & & \\ 0 & (0,0) & (1,0) \\ | & & \\ \llcorner\!\!\!- & 0 \text{ ——— } 1 \longrightarrow y_0 & \end{array} \tag{74.109}$$

[9] If $x \in \{0, 1\}$ and $P(a | \underline{x} = 0) = P(a)$, then $P(a) - P(a, \underline{x} = 0) = P(a)[1 - P(\underline{x} = 0)]$, so $P(a, \underline{x} = 1) = P(a)P(\underline{x} = 1)$. Hence, $P(a | \underline{x} = 1) = P(a)$.

When $\epsilon_x(y_0, y_1)$ is used inside a 2×2 matrix, we will not write the argument (y_0, y_1), leaving it implicit, unless confusion may arise. Thus, for example, we will represent $\mathcal{P}\begin{bmatrix} \epsilon_0(0,1) & 0 \\ 0 & 0 \end{bmatrix}$ by $\mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ 0 & 0 \end{bmatrix}$.

For $x \in \{0, 1\}$, let

$$P_{\underline{y}_0,\underline{y}_1,\underline{x}}(0,0,x) = \mathcal{P}\begin{bmatrix} 0 & 0 \\ \epsilon_x & 0 \end{bmatrix} \tag{74.110a}$$

$$P_{\underline{y}_0,\underline{y}_1,\underline{x}}(0,1,x) = \mathcal{P}\begin{bmatrix} \epsilon_x & 0 \\ 0 & 0 \end{bmatrix} \tag{74.110b}$$

$$P_{\underline{y}_0,\underline{y}_1,\underline{x}}(1,0,x) = \mathcal{P}\begin{bmatrix} 0 & 0 \\ 0 & \epsilon_x \end{bmatrix} \tag{74.110c}$$

$$P_{\underline{y}_0,\underline{y}_1,\underline{x}}(1,1,x) = \mathcal{P}\begin{bmatrix} 0 & \epsilon_x \\ 0 & 0 \end{bmatrix} \tag{74.110d}$$

Define a map $\mathcal{P}[\quad] : V \to \mathbb{R}$ that we will call the **matrix representation of probabilities (MRP)** (mnemonic, Mr. P). We will assume that the map $\mathcal{P}[\quad]$ is linear. Hence, for instance,

$$\mathcal{P}\begin{bmatrix} 4\epsilon_0 + 3\epsilon_1 & -5\epsilon_0 \\ 0 & 0 \end{bmatrix} = 4\mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ 0 & 0 \end{bmatrix} + 3\mathcal{P}\begin{bmatrix} \epsilon_1 & 0 \\ 0 & 0 \end{bmatrix} - 5\mathcal{P}\begin{bmatrix} 0 & \epsilon_0 \\ 0 & 0 \end{bmatrix} \tag{74.111}$$

Henceforth, we will use the abbreviation

$$\epsilon_+ = \epsilon_0 + \epsilon_1 \tag{74.112}$$

Note that

$$\mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_+ \\ \epsilon_+ & \epsilon_+ \end{bmatrix} = 1 \tag{74.113}$$

$$\pi_x = \mathcal{P}\begin{bmatrix} \epsilon_x & \epsilon_x \\ \epsilon_x & \epsilon_x \end{bmatrix} \tag{74.114}$$

$$E_{0|0} = \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ \epsilon_+ & 0 \end{bmatrix}, \quad O_{0,0} = \mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ \epsilon_0 & 0 \end{bmatrix} \tag{74.115}$$

$$E_{0|1} = \mathcal{P}\begin{bmatrix} 0 & 0 \\ \epsilon_+ & \epsilon_+ \end{bmatrix}, \quad O_{1,0} = \mathcal{P}\begin{bmatrix} 0 & 0 \\ \epsilon_1 & \epsilon_1 \end{bmatrix} \tag{74.116}$$

$$E_{1|0} = \mathcal{P}\begin{bmatrix} 0 & \epsilon_+ \\ 0 & \epsilon_+ \end{bmatrix}, \quad O_{0,1} = \mathcal{P}\begin{bmatrix} 0 & \epsilon_0 \\ 0 & \epsilon_0 \end{bmatrix} \tag{74.117}$$

$$E_{1|1} = \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_+ \\ 0 & 0 \end{bmatrix}, \quad O_{1,1} = \mathcal{P}\begin{bmatrix} \epsilon_1 & \epsilon_1 \\ 0 & 0 \end{bmatrix} \tag{74.118}$$

$$P(\underline{y} = 0) = O_{0,0} + O_{1,0} = \mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ \epsilon_+ & \epsilon_1 \end{bmatrix} \tag{74.119}$$

$$P(\underline{y} = 1) = O_{0,1} + O_{1,1} = \mathcal{P}\begin{bmatrix} \epsilon_1 & \epsilon_+ \\ 0 & \epsilon_0 \end{bmatrix} \tag{74.120}$$

$$PN * O_{1,1} = \mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ 0 & 0 \end{bmatrix} \tag{74.121}$$

$$PS * O_{0,0} = \mathcal{P}\begin{bmatrix} \epsilon_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{74.122}$$

$$PNS = \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ 0 & 0 \end{bmatrix} \tag{74.123}$$

$$PH = \mathcal{P}\begin{bmatrix} 0 & 0 \\ 0 & \epsilon_+ \end{bmatrix} \tag{74.124}$$

$$ATE = PNS - PH = \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ 0 & -\epsilon_+ \end{bmatrix} \tag{74.125}$$

When $PH = 0$,

$$P(\underline{y} = 1)|_{PH=0} = \mathcal{P}\begin{bmatrix} \epsilon_1 & \epsilon_+ \\ 0 & 0 \end{bmatrix} \tag{74.126}$$

The special constraints of exogeneity, strong exogeneity and monotonicity have a very simple in the MRP.

Suppose that $\pi_0, \pi_1 \geq 1$ and $\pi_0 + \pi_1 = 1$. Note that

$$\left.\begin{array}{c} \epsilon_+ = \frac{\epsilon_0}{\pi_0} \\ \text{or} \\ \epsilon_+ = \frac{\epsilon_1}{\pi_1} \end{array}\right\} \implies \epsilon_+ = \frac{\epsilon_0}{\pi_0} = \frac{\epsilon_1}{\pi_1} \tag{74.127}$$

Below, we will abbreviate

$$\widehat{\epsilon}_x = \frac{\epsilon_x}{\pi_x} \tag{74.128}$$

for $x \in \{0, 1\}$.

Claim 113

(a) Exogeneity holds iff ("4 sides can change color")

$$\mathcal{P}\begin{bmatrix} \widehat{\epsilon}_0 & 0 \\ \widehat{\epsilon}_0 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} \widehat{\epsilon}_1 & 0 \\ \widehat{\epsilon}_1 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ \epsilon_+ & 0 \end{bmatrix} \tag{74.129a}$$

$$\mathcal{P}\begin{bmatrix} \widehat{\epsilon}_0 & \widehat{\epsilon}_0 \\ 0 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} \widehat{\epsilon}_1 & \widehat{\epsilon}_1 \\ 0 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_+ \\ 0 & 0 \end{bmatrix} \tag{74.129b}$$

$$\mathcal{P}\begin{bmatrix} 0 & \widehat{\epsilon}_0 \\ 0 & \widehat{\epsilon}_0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} 0 & \widehat{\epsilon}_1 \\ 0 & \widehat{\epsilon}_1 \end{bmatrix} = \mathcal{P}\begin{bmatrix} 0 & \epsilon_+ \\ 0 & \epsilon_+ \end{bmatrix} \tag{74.129c}$$

$$\mathcal{P}\begin{bmatrix} 0 & 0 \\ \widehat{\epsilon}_0 & \widehat{\epsilon}_0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} 0 & 0 \\ \widehat{\epsilon}_1 & \widehat{\epsilon}_1 \end{bmatrix} \mathcal{P}\begin{bmatrix} 0 & 0 \\ \epsilon_+ & \epsilon_+ \end{bmatrix} \tag{74.129d}$$

(b) Monotonicity holds iff

$$\mathcal{P}\begin{bmatrix} 0 & 0 \\ 0 & \epsilon_x \end{bmatrix} = 0 \tag{74.130}$$

for $x \in \{0, 1\}$.

(c) Strong exogeneity holds iff ("4 corners can change color")

$$\mathcal{P}\begin{bmatrix} \widehat{\epsilon}_0 & 0 \\ 0 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} \widehat{\epsilon}_1 & 0 \\ 0 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ 0 & 0 \end{bmatrix} \tag{74.131a}$$

$$\mathcal{P}\begin{bmatrix} 0 & \widehat{\epsilon}_0 \\ 0 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} 0 & \widehat{\epsilon}_1 \\ 0 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} 0 & \epsilon_+ \\ 0 & 0 \end{bmatrix} \tag{74.131b}$$

$$\mathcal{P}\begin{bmatrix} 0 & 0 \\ 0 & \widehat{\epsilon}_0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} 0 & 0 \\ 0 & \widehat{\epsilon}_1 \end{bmatrix} = \mathcal{P}\begin{bmatrix} 0 & 0 \\ 0 & \epsilon_+ \end{bmatrix} \tag{74.131c}$$

$$\mathcal{P}\begin{bmatrix} 0 & 0 \\ \widehat{\epsilon}_0 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} 0 & 0 \\ \widehat{\epsilon}_1 & 0 \end{bmatrix} = \mathcal{P}\begin{bmatrix} 0 & 0 \\ \epsilon_+ & 0 \end{bmatrix} \tag{74.131d}$$

proof:

(a) This follows from the definitions of $E_{y|x}$ and $O_{x,y}$ for $x, y \in \{0, 1\}$, as stated above in the MRP.

(b) This follows from the identity

$$P_{\underline{y}_0,\underline{y}_1,\underline{x}}(1, 0|x) = \mathcal{P}\begin{bmatrix} 0 & 0 \\ 0 & \frac{\epsilon_x}{\pi_x} \end{bmatrix} \tag{74.132}$$

for $x \in \{0, 1\}$.

(c) This follows from definitions of $P_{\underline{y}_0,\underline{y}_1,\underline{x}}(y_0, y_1, x)$ for $y_0, y_1, x \in \{0, 1\}$, as stated above in the MRP.

QED

Note that it is obvious from Claim 113 that strong exogeneity implies exogeneity.

Claim 113 also makes it easy peasy to prove that exogeneity and monotonicity imply strong exogeneity. Indeed, here is a much simpler proof than the one we presented earlier.

Claim 114 *Exogeneity and monotonicity imply strong exogeneity.*

proof: Let's call $E(a), E(b), E(c), E(d)$ the four Eqs.(74.129) that define Exogeneity, and $SE(a), SE(b), SE(c), SE(d)$ the four Eqs.(74.131) that define Strong Exogeneity.

- Monotonicity implies the lower right entry is zero, so $SE(c)$ is true.
- Setting to zero the lower right entry in $E(c)$ and $E(d)$ implies $SE(b)$ and $SE(d)$.
- Subtracting $SE(d)$ from $E(a)$ gives $SE(a)$.

QED

74.9 Bounds on Exp. Probs. imposed by Obs. Probs.

Claim 115 *In general,*

$$O_{x,1} \leq E_{1|x} \leq 1 - O_{x,0} \quad \textit{for } x \in \{0, 1\} \ . \tag{74.133}$$

In other words,

$$O_{1,1} \leq E_{1|1} \leq 1 - O_{1,0} \tag{74.134}$$

$$O_{0,1} \leq E_{1|0} \leq 1 - O_{0,0} \tag{74.135}$$

With monotonicity, we get the tighter bounds

$$\overbrace{O_{1,1} + \underbrace{O_{0,1}}_{new}}^{P(\underline{y}=1)} \leq E_{1|1} \leq 1 - O_{1,0} \tag{74.136}$$

$$O_{0,1} \leq E_{1|0} \leq \overbrace{1 - O_{0,0} - \underbrace{O_{1,0}}_{new}}^{P(\underline{y}=1)} \tag{74.137}$$

proof:

$$O_{1,1} \leq E_{1|1} \leq \underbrace{1 - O_{1,0}}_{O_{1,1}+O_{0,0}+O_{0,1}} \tag{74.138}$$

has the MRP

$$\mathcal{P}\begin{bmatrix} \epsilon_1 & \epsilon_1 \\ 0 & 0 \end{bmatrix} \leq \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_+ \\ 0 & 0 \end{bmatrix} \leq \mathcal{P}\begin{bmatrix} \epsilon_1 & \epsilon_1 \\ 0 & 0 \end{bmatrix} + \mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ \epsilon_0 & 0 \end{bmatrix} + \mathcal{P}\begin{bmatrix} 0 & \epsilon_0 \\ 0 & \epsilon_0 \end{bmatrix} \tag{74.139}$$

which is obviously true.

$$O_{0,1} \leq E_{1|0} \leq \underbrace{1 - O_{0,0}}_{O_{1,1}+O_{0,1}+O_{1,0}} \tag{74.140}$$

has the MRP

$$\mathcal{P}\begin{bmatrix} 0 & \epsilon_0 \\ 0 & \epsilon_0 \end{bmatrix} \leq \mathcal{P}\begin{bmatrix} 0 & \epsilon_+ \\ 0 & \epsilon_+ \end{bmatrix} \leq \mathcal{P}\begin{bmatrix} \epsilon_1 & \epsilon_1 \\ 0 & 0 \end{bmatrix} + \mathcal{P}\begin{bmatrix} 0 & \epsilon_0 \\ 0 & \epsilon_0 \end{bmatrix} + \mathcal{P}\begin{bmatrix} 0 & 0 \\ \epsilon_1 & \epsilon_1 \end{bmatrix} \tag{74.141}$$

which is obviously true.

$$P(\underline{y} = 1)|_{PH=0} \leq E_{1|1} \tag{74.142}$$

has the MRP

$$\mathcal{P}\begin{bmatrix} \epsilon_1 & \epsilon_+ \\ 0 & 0 \end{bmatrix} \leq \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_+ \\ 0 & 0 \end{bmatrix} \tag{74.143}$$

which is obviously true.

$$E_{1|0}|_{PH=0} \leq P(\underline{y} = 1)|_{PH=0} \tag{74.144}$$

has the MRP

$$\mathcal{P}\begin{bmatrix} 0 & \epsilon_+ \\ 0 & 0 \end{bmatrix} \leq \mathcal{P}\begin{bmatrix} \epsilon_1 & \epsilon_+ \\ 0 & 0 \end{bmatrix} \tag{74.145}$$

which is obviously true.
QED

74.10 Bounds on $PNS3$ for unspecified bnet

Claim 116 *If $P(a, b)$ is a probability distribution, then*

$$\max\{0, P(a) + P(b) - 1\} \leq P(a, b) \leq \min\{P(a), P(b)\} \tag{74.146}$$

proof: $P(a|b) \leq 1$ and $P(b|a) \leq 1$ implies $P(a,b) \leq P(b)$ and $P(a,b) \leq P(a)$. Hence, $P(a,b) \leq \min\{P(a), P(b)\}$.

$$P(a) + P(b) - 1 = \sum_{a',b'} P(a',b') \underbrace{[\delta(a,a') + \delta(b,b') - 1]}_{T} \tag{74.147}$$

$$\leq P(a,b) \quad (T \text{ biggest when } a = a' \text{ and } b = b') \tag{74.148}$$

QED

Figure 74.6: Minimal bounds for PNS. PNS is larger than maximum of the purple segments and smaller than the minimum of the green ones. The purple segment of almost zero length represents zero. The green segment interrupted by a thin green line represents $O_{0,0} + O_{1,1}$. Note that $\sum_x O_{x,0} + \sum_x O_{x,1} = 1$. When monotonicity holds, PNS equals $E_{1,1} + E_{0,0} - 1$ which is the purple segment marked with a star.

Claim 117 *Minimal bounds (see Fig.74.6)*
If the minimal constraints hold for simplex $\mathcal{S}$, then

$$\max \left\{ \begin{array}{c} 0 \\ E_{1|1} + E_{0|0} - 1 \\ E_{0|0} - \sum_x O_{x,0} \\ E_{1|1} - \sum_x O_{x,1} \end{array} \right\} \leq PNS \leq \min \left\{ \begin{array}{c} E_{1|1} \\ E_{0|0} \\ O_{1,1} + O_{0,0} \\ E_{1|1} + E_{0|0} - O_{1,1} - O_{0,0} \end{array} \right\} \tag{74.149}$$

$$\max \left\{ \begin{array}{c} 0 \\ \frac{E_{0|0} - \sum_x O_{x,0}}{O_{1,1}} \end{array} \right\} \leq PN \leq \min \left\{ \begin{array}{c} 1 \\ \frac{E_{0|0} - O_{0,0}}{O_{1,1}} \end{array} \right\} \tag{74.150}$$

$$\max \left\{ \begin{array}{c} 0 \\ \frac{E_{1|1} - \sum_x O_{x,1}}{O_{0,0}} \end{array} \right\} \leq PS \leq \min \left\{ \begin{array}{c} 1 \\ \frac{E_{1|1} - O_{1,1}}{O_{0,0}} \end{array} \right\} \tag{74.151}$$

proof: These bounds were copied directly from Ref.[84], except the notation was changed. In certain cases, we have slightly rewritten the bounds from Ref.[84] to exhibit more explicitly their symmetry under swaps of zeros and ones. For example, instead of using $(E_{1|0}, E_{1|1})$ as parameters, we use $(E_{1|1}, E_{0|0})$.

These bounds are obviously true in the MRP. Indeed, in the MRP, the following holds. For PNS, we have

$$\begin{bmatrix} 0 \\ E_{1|1} + E_{0|0} - 1 \\ E_{0|0} - \sum_x O_{x,0} \\ E_{1|1} - \sum_x O_{x,1} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ 0 & -\epsilon_+ \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_1 & 0 \\ 0 & -\epsilon_1 \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ 0 & -\epsilon_0 \end{bmatrix} \end{bmatrix} \tag{74.152}$$

$$PNS = \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ 0 & 0 \end{bmatrix} \tag{74.153}$$

$$\begin{bmatrix} E_{1|1} \\ E_{0|0} \\ O_{1,1} + O_{0,0} \\ E_{1|1} + E_{0|0} - O_{1,1} - O_{0,0} \end{bmatrix} = \begin{bmatrix} \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ \epsilon_+ & 0 \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_+ \\ 0 & 0 \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_1 \\ \epsilon_0 & 0 \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_0 \\ \epsilon_1 & 0 \end{bmatrix} \end{bmatrix} \tag{74.154}$$

For PN, we have

$$\begin{bmatrix} 0 \\ E_{1|1} - \sum_x O_{x,1} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ 0 & -\epsilon_0 \end{bmatrix} \end{bmatrix} \tag{74.155}$$

$$PN * O_{1,1} = \mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ 0 & 0 \end{bmatrix} \tag{74.156}$$

$$\begin{bmatrix} 1 \\ E_{1|1} - O_{1,1} \end{bmatrix} = \begin{bmatrix} 1 \\ \mathcal{P}\begin{bmatrix} \epsilon_0 & \epsilon_0 \\ 0 & 0 \end{bmatrix} \end{bmatrix} \tag{74.157}$$

For PS, we have

$$\begin{bmatrix} 0 \\ E_{0|0} - \sum_x O_{x,0} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{P}\begin{bmatrix} \epsilon_1 & 0 \\ 0 & -\epsilon_1 \end{bmatrix} \end{bmatrix} \tag{74.158}$$

$$PS * O_{0,0} = \mathcal{P}\begin{bmatrix} \epsilon_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{74.159}$$

$$\begin{bmatrix} 1 \\ E_{0|0} - O_{0,0} \end{bmatrix} = \begin{bmatrix} 1 \\ \mathcal{P}\begin{bmatrix} \epsilon_1 & \epsilon_1 \\ 0 & 0 \end{bmatrix} \end{bmatrix} \tag{74.160}$$

The first two lines of the bound for PNS are a simple consequence of Claim 116. Indeed, Claim 116 implies

$$\max \left\{ \begin{array}{c} 0 \\ P(y_0) + P(y_1) - 1 \end{array} \right\} \leq P(y_0, y_1) \leq \min \left\{ \begin{array}{c} P(y_0) \\ P(y_1) \end{array} \right\} \tag{74.161}$$

When $y_0 = 0, y_1 = 1$, we get

$$\max \left\{ \begin{array}{c} 0 \\ E_{0|0} + E_{1|1} - 1 \end{array} \right\} \leq PNS \leq \min \left\{ \begin{array}{c} E_{0|0} \\ E_{1|1} \end{array} \right\} \tag{74.162}$$

QED

In Claim 117, note that

- $\sum_x O_{x,y} = P(\underline{y} = y)$ for $y = 0, 1$.
- Let

$$E'_{0|0} = E_{0|0} - \sum_x O_{x,0} = \mathcal{P} \left[\begin{array}{cc} \epsilon_1 & 0 \\ 0 & -\epsilon_1 \end{array} \right] \tag{74.163}$$

$$E'_{1|1} = E_{1|1} - \sum_x O_{x,1} = \mathcal{P} \left[\begin{array}{cc} \epsilon_0 & 0 \\ 0 & -\epsilon_0 \end{array} \right] \tag{74.164}$$

$$ATE = E_{1|1} + E_{0|0} - 1 = E'_{1|1} + E'_{0|0} = \mathcal{P} \left[\begin{array}{cc} \epsilon_+ & 0 \\ 0 & -\epsilon_+ \end{array} \right] \tag{74.165}$$

- Claim 117 applies if we have both observational data (OD) and experimental data (ED). If we only have OD (resp., ED), ignore all bounds that involve an E (resp., involve an O) probability.

  Hence, with only OD[10]

$$0 \leq PNS \leq O_{1,1} + O_{0,0} \tag{74.166}$$

  and $PN, PS \in [0, 1]$, whereas with only ED,

$$\max \left\{ \begin{array}{c} 0 \\ E_{1|1} + E_{0|0} - 1 \end{array} \right\} \leq PNS \leq \min \left\{ \begin{array}{c} E_{1|1} \\ E_{0|0} \end{array} \right\} \tag{74.167}$$

  and $PN, PS \in [0, 1]$.

[10]The bounds Eq.(74.166) assume exogeneity does not hold. For the case when exogeneity does hold, stronger bounds will be given later on in the chapter.

- At first blush, there seem to be 3 possible cases to consider: (1) Only OD. (2) Both OD and ED. (3) Only ED. Actually, Case (1) plus the assumption of exogeneity is the same as Case (3). Indeed, recall that exogeneity means no confounding, and ED (i.e., an RCT) also has no confounding. So we don't have to consider Case (3) if we consider Case (1) without and with exogeneity. Exogeneity is built into case (2), so case (2) without exogeneity is meaningless.

Claim 118 *If minimal and exogeneity constraints hold for simplex $\mathcal{S}$, then*

$$\max\left\{\begin{array}{c} 0 \\ O_{1|1}+O_{0|0}-1 \\ \cancel{O_{0|0}-\sum_x O_{x,0}} \\ \cancel{O_{1|1}-\sum_x O_{x,1}} \end{array}\right\} \leq PNS \leq \min\left\{\begin{array}{c} O_{1|1} \\ O_{0|0} \\ \cancel{O_{1,1}+O_{0,0}} \\ \cancel{O_{1|1}+O_{0|0}-O_{1,1}-O_{0,0}} \end{array}\right\} \tag{74.168}$$

$$\max\left\{\begin{array}{c} 0 \\ ERR \end{array}\right\} \leq PN \leq \min\left\{\begin{array}{c} 1 \\ \frac{O_{0|0}}{O_{1|1}} \end{array}\right\} \tag{74.169}$$

$$\max\left\{\begin{array}{c} 0 \\ (ERR)^{\sim} \end{array}\right\} \leq PS \leq \min\left\{\begin{array}{c} 1 \\ \frac{O_{1|1}}{O_{0|0}} \end{array}\right\} \tag{74.170}$$

proof: Just replace $E_{y|x}$ by $O_{y|x}$ in the minimal bounds given in Claim 117.

The canceled terms do not improve the bounds and can be dropped. We show this next using MRP.

In the MRP, the minimal bounds for PNS are

$$\begin{bmatrix} 0 \\ E_{1|1}+E_{0|0}-1 \\ E_{0|0}-\sum_x O_{x,0} \\ E_{1|1}-\sum_x O_{x,1} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ 0 & -\epsilon_+ \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_1 & 0 \\ 0 & -\epsilon_1 \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_0 & 0 \\ 0 & -\epsilon_0 \end{bmatrix} \end{bmatrix} \tag{74.171}$$

$$PNS = \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ 0 & 0 \end{bmatrix} \tag{74.172}$$

$$\begin{bmatrix} E_{1|1} \\ E_{0|0} \\ O_{1,1}+O_{0,0} \\ E_{1|1}+E_{0|0}-O_{1,1}-O_{0,0} \end{bmatrix} = \begin{bmatrix} \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ \epsilon_+ & 0 \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_+ \\ 0 & 0 \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_1 \\ \epsilon_0 & 0 \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_0 \\ \epsilon_1 & 0 \end{bmatrix} \end{bmatrix} \tag{74.173}$$

If exogeneity holds, we can replace all E's by O's on the left hand sides. We can also use $\epsilon_+ = \widehat{\epsilon}_0 = \widehat{\epsilon}_1$ on the right hand sides to conclude that

$$O_{1|1} + O_{0|0} - 1 = \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_+ - \epsilon_+ \\ 0 & -\epsilon_+ \end{bmatrix} = \mathcal{P}\begin{bmatrix} \widehat{\epsilon}_x & \widehat{\epsilon}_x - \widehat{\epsilon}_x \\ 0 & -\widehat{\epsilon}_x \end{bmatrix} = \frac{1}{\pi_x}\mathcal{P}\begin{bmatrix} \epsilon_x & 0 \\ 0 & -\epsilon_x \end{bmatrix} \tag{74.174}$$

for both $x \in \{0,1\}$. Note that $\frac{1}{\pi_x}\mathcal{P}\begin{bmatrix} \epsilon_x & 0 \\ 0 & -\epsilon_x \end{bmatrix}$ will always be greater or equal to $\mathcal{P}\begin{bmatrix} \epsilon_x & 0 \\ 0 & -\epsilon_x \end{bmatrix}$ when $\mathcal{P}\begin{bmatrix} \epsilon_x & 0 \\ 0 & -\epsilon_x \end{bmatrix} \geq 0$ (if it's negative, the 0 bound takes precedence). Hence, the two canceled lower bound terms can be dropped. A similar argument shows that the two canceled upper bound terms can be dropped too.
QED

Claim 119 *If the minimal and strong exogeneity constraints hold for simplex $\mathcal{S}$, then the inequalities for exogeneity Claim 118 hold. In addition,*

$$PN = \frac{PNS}{O_{1|1}} \tag{74.175}$$

and

$$PS = \frac{PNS}{O_{0|0}} \tag{74.176}$$

proof:

$$\begin{aligned} PN * O_{1|1} &= P(\underline{y}_0 = 0, \underline{y} = 1 | \underline{x} = 1) && (74.177) \\ &= P(\underline{y}_0 = 0, \underline{y}_1 = 1) && (74.178) \\ &= PNS && (74.179) \end{aligned}$$

$$\begin{aligned} PS * O_{0|0} &= P(\underline{y}_1 = 1, \underline{y} = 0 | \underline{x} = 0) && (74.180) \\ &= P(\underline{y}_0 = 0, \underline{y}_1 = 1) && (74.181) \\ &= PNS && (74.182) \end{aligned}$$

QED

Claim 120 *If the minimal and monotonicity constraints hold for simplex $\mathcal{S}$, then*

$$PNS = E_{1|1} + E_{0|0} - 1 \tag{74.183}$$

$$PN = \frac{E_{0|0} - \sum_x O_{x,0}}{O_{1,1}} = \frac{\sum_x O_{x,1} - E_{1|0}}{O_{1,1}} = CERR \tag{74.184}$$

$$PS = \frac{E_{1|1} - \sum_x O_{x,1}}{O_{0,0}} = \frac{\sum_x O_{x,0} - E_{0|1}}{O_{0,0}} = (CERR)^{\sim} \tag{74.185}$$

proof: These results can be easily proven using the MRP.
Note that

$$\frac{\sum_x O_{x,1} - E_{1|0}}{O_{1,1}} = \frac{O_{1|1}\pi_1 + O_{1|0}(1-\pi_1) - E_{1|0}}{O_{1|1}\pi_1} \tag{74.186}$$

$$= 1 - \frac{O_{1|0}}{O_{1|1}} + \frac{O_{1|0} - E_{1|0}}{O_{1,1}} \tag{74.187}$$

$$= CERR \tag{74.188}$$

QED

Claim 121 *If the minimal, exogeneity and monotonicity constraints hold for simplex $\mathcal{S}$, then PNS, PN,PS are identifiable, and*

$$PNS = O_{1|1} + O_{0|0} - 1 \tag{74.189}$$

$$PN = \frac{O_{0|0} - \sum_x O_{x,0}}{O_{1,1}} = \frac{\sum_x O_{x,1} - O_{1|0}}{O_{1,1}} = ERR \tag{74.190}$$

$$PS = \frac{O_{1|1} - \sum_x O_{x,1}}{O_{0,0}} = \frac{\sum_x O_{x,0} - O_{0|1}}{O_{0,0}} = (ERR)^{\sim} \tag{74.191}$$

proof: Set $E_{y|x} = O_{y|x}$ in Claim 120.
QED

74.11 Bounds on $PNS3$ for specific bnet families

74.12 Bounds on ATE imposed by Obs. Probs.

Claim 122

$$O_{0,0} + O_{1,1} - 1 \leq ATE \leq O_{0,0} + O_{1,1} \tag{74.192}$$

proof:

$$O_{0,0} + O_{1,1} - 1 = \mathcal{P}\left[\begin{array}{cc} 0 & -\epsilon_0 \\ -\epsilon_1 & -\epsilon_+ \end{array}\right] \tag{74.193}$$

$$ATE = \mathcal{P}\left[\begin{array}{cc} \epsilon_+ & 0 \\ 0 & -\epsilon_+ \end{array}\right] \tag{74.194}$$

$$O_{0,0} + O_{1,1} = \mathcal{P}\left[\begin{array}{cc} \epsilon_+ & \epsilon_1 \\ \epsilon_0 & 0 \end{array}\right] \tag{74.195}$$

QED

74.13 Bounds on PNS in terms of ATE and Obs. Probs.

Claim 123

$$\max\left\{\begin{array}{c} 0 \\ ATE \end{array}\right\} \leq PNS \leq \min\left\{\begin{array}{c} O_{0,0} + O_{1,1} \\ O_{0,1} + O_{1,0} + ATE \end{array}\right\} \tag{74.196}$$

Hence,

$$\max\left\{\begin{array}{c} 0 \\ ATE \end{array}\right\} \leq PNS \leq \min\left\{\begin{array}{c} 1 \\ 1 + ATE \end{array}\right\} \tag{74.197}$$

proof:

Note that

$$ATE = \mathcal{P}\left[\begin{array}{cc} \epsilon_+ & 0 \\ 0 & -\epsilon_+ \end{array}\right] \tag{74.198}$$

and

$$PNS = \mathcal{P}\left[\begin{array}{cc} \epsilon_+ & 0 \\ 0 & 0 \end{array}\right] \tag{74.199}$$

so

$$\max\left\{\begin{array}{c} 0 \\ ATE \end{array}\right\} = \max\left\{\begin{array}{c} 0 \\ \mathcal{P}\left[\begin{array}{cc} \epsilon_+ & 0 \\ 0 & -\epsilon_+ \end{array}\right] \end{array}\right\} \leq \mathcal{P}\left[\begin{array}{cc} \epsilon_+ & 0 \\ 0 & 0 \end{array}\right] = PNS \tag{74.200}$$

Next note that

$$O_{0,0} + O_{1,1} = \mathcal{P}\left[\begin{array}{cc} \epsilon_+ & \epsilon_1 \\ \epsilon_0 & 0 \end{array}\right] \tag{74.201}$$

$$O_{1,0} + O_{0,1} = \mathcal{P}\begin{bmatrix} 0 & \epsilon_0 \\ \epsilon_1 & \epsilon_+ \end{bmatrix} \tag{74.202}$$

$$O_{1,0} + O_{0,1} + ATE = \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_0 \\ \epsilon_1 & 0 \end{bmatrix} \tag{74.203}$$

so

$$PNS = \mathcal{P}\begin{bmatrix} \epsilon_+ & 0 \\ 0 & 0 \end{bmatrix} \leq \min\left\{ \begin{array}{c} \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_1 \\ \epsilon_0 & 0 \end{bmatrix} \\ \mathcal{P}\begin{bmatrix} \epsilon_+ & \epsilon_0 \\ \epsilon_1 & 0 \end{bmatrix} \end{array} \right\} = \min\left\{ \begin{array}{c} O_{0,0} + O_{1,1} \\ O_{1,0} + O_{0,1} + ATE \end{array} \right\} \tag{74.204}$$

QED

74.14 Numerical Examples

I've written an open source Python program (See Ref[91]) that calculates the bounds given in this chapter. The program is called "JudeasRx", in honor of Judea Pearl. Fig.74.7 is an example of its interface with data entered by Boris Sobolev from a real life case. JudeasRx considers $z = g = gender \in \{m, f\}$.

Figure 74.7: Interface for the Python program JudeasRx, with data entered by Boris Sobolev.

Chapter 75

Petri Nets

This chapter is based on Refs.[173] and [86]

A **Petri net** (pnet) is basically a diagram of an idealized machine that features actions (called transitions) and buffers (called places) that contain resources (called tokens). This diagram evolves in time like a motion picture. In that motion picture, transitions are fired at various times, sometimes concurrently (i.e., in parallel) and the effect of that is shown by the motion of the tokens. The evolution of many machines can be abstracted to a pnet.

A pnet portrays the evolution and allocation of token resources of an idealized machine, whereas a bnet portrays the causal connections of events with each other. Two big differences between the two diagrams are:

- No evolution occurs in a bnet; it's as if the bnet's TPMs (transition probabilitiy matrices) had been calculated empirically, and those empirical distributions had reached a steady state long ago. On the other hand, evolution does occur in a pnet, so we can say that a pnet occupies a transient state.
- pnet diagrams portray a machine. They do not necessarily portray events as nodes (although they can, and events do occur in their motion picture). bnets, on the other hand, do portray events as nodes and their causal connections.

So if pnets and bnets are so different, why do I discuss pnets in this book about bnets? Well, it turns out that one can build a pnet on top of a bnet, using the bnet nodes as the transition nodes of the pnet. The resulting diagram, called a **Bayes-Petri net**, gives both transient and steady state information about causality.

For $x \in \mathbb{R}$, define the **positive part** of x by

$$(x)^{\geq 0} = x\mathbb{1}(x \geq 0) \tag{75.1}$$

For $x^n \in \mathbb{R}^n$, define

$$x^n \geq 0 \quad \text{iff} \quad x_i^n \geq 0 \;\; \forall i \tag{75.2}$$

75.1 Original Petri Net

75.1.1 Simple Example of Petri Net

Fig.75.1 shows a snapshot of a pnet at a specific time, and Fig.75.2 shows the same pnet snapshot, rendered in a different style. The first picture represents the number of tokens by an integer within a circle. The second picture represents the number of tokens by the number of bullet points within a rectangle.

Figure 75.1: Simple example of a pnet. Place nodes are labeled $\{\underline{p}_i\}_{i=1}^4$ and transition nodes $\{\underline{x}_i\}_{i=1}^2$. The number of tokens in each place $\underline{p}_i$ is indicated by a number (or a whitespace for 0 tokens) within a circle.

Figure 75.2: Same pnet as that in Fig.75.1, except that here the number of tokens is represented by the number of bullet points within a rectangle.

Let

$$W^{\underline{p}\rightarrow} = \begin{array}{c|cc} & \underline{x}_1 & \underline{x}_2 \\ \hline \underline{p}_1 & 1 & 0 \\ \underline{p}_2 & 0 & 1 \\ \underline{p}_3 & 0 & 1 \\ \underline{p}_4 & 0 & 0 \end{array}, \quad W^{\underline{p}\leftarrow} = \begin{array}{c|cc} & \underline{x}_1 & \underline{x}_2 \\ \hline \underline{p}_1 & 0 & 1 \\ \underline{p}_2 & 1 & 0 \\ \underline{p}_3 & 1 & 0 \\ \underline{p}_4 & 0 & 1 \end{array}, \quad W = W^{\underline{p}\leftarrow} - W^{\underline{p}\rightarrow} = \begin{array}{c|cc} & \underline{x}_1 & \underline{x}_2 \\ \hline \underline{p}_1 & -1 & 1 \\ \underline{p}_2 & 1 & -1 \\ \underline{p}_3 & 1 & -1 \\ \underline{p}_4 & 0 & 1 \end{array} \tag{75.3}$$

Matrix W is called the incidence matrix. Note that it has a 1 (resp., -1) at position (i, j) if there is an arrow $\underline{p}_i \leftarrow \underline{x}_j$ (resp., $\underline{p}_i \rightarrow \underline{x}_j$).

If

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \; x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{75.4}$$

then:

$$p = p(0) + Wx = \begin{bmatrix} p_1(0) - x_1 + x_2 \\ p_2(0) + x_1 - x_2 \\ p_3(0) + x_1 - x_2 \\ p_4(0) + x_2 \end{bmatrix} \tag{75.5}$$

3 possible firing rules, in order of increasing severity, are:

1. If we require that all places end with zero or a positive number of tokens, then the firing rule is

$$p = (p(0) + Wx)^{\geq 0} = \begin{bmatrix} (p_1(0) - x_1 + x_2)^{\geq 0} \\ (p_2(0) + x_1 - x_2)^{\geq 0} \\ (p_3(0) + x_1 - x_2)^{\geq 0} \\ (p_4(0) + x_2)^{\geq 0} \end{bmatrix} \tag{75.6}$$

   If we fire transition $\underline{x}_2$ once (i.e., if we set $x_1 = 0, x_2 = 1$), we get

$$p = \begin{bmatrix} (p_1(0) + 1)^{\geq 0} \\ (p_2(0) - 1)^{\geq 0} \\ (p_3(0) - 1)^{\geq 0} \\ (p_4(0) + 1)^{\geq 0} \end{bmatrix} = \begin{bmatrix} (3 + 1)^{\geq 0} \\ (0 - 1)^{\geq 0} \\ (2 - 1)^{\geq 0} \\ (1 + 1)^{\geq 0} \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 1 \\ 2 \end{bmatrix} \tag{75.7}$$

2. If we require that only incoming places with enough content are used, then the firing rule is

$$p = (p(0) - W^{\underline{p}\rightarrow} x)^{\geq 0} + W^{\underline{p}\leftarrow} x \tag{75.8}$$

3. If we require that all incoming places are used ("fully enabled" condition), then the firing rule is

$$p = \begin{cases} p(0) - Wx & \text{if } [p(0) - W^{\underline{p}\rightarrow} x] \geq 0 \\ p(0) & \text{otherwise} \end{cases} \tag{75.9}$$

The fully enabled condition is usually assumed because it makes the most physical sense out of the 3. The first one can "borrow on credit" (i.e., deliver tokens that it does not have yet) and the second one can "create new tokens out of nothing".

75.1.2 Precise Definition of Petri Net

Let

$\mathbb{Z}_{>0} = \{1, 2, 3, \dots\}$ natural numbers.

$\mathbb{Z}_{\geq 0} = \{0, 1, 2, 3, \dots\}$ natural numbers and zero.

$\underline{p}_i$ = ith **place (a.k.a. buffer, token holder) node**. This node holds either an integer ≥ 0 or that number of tokens (tokens are represenxed by bullet points).

$\mathcal{P} = \{\underline{p}_i\}_{i=1}^{np}$, set of all place nodes.

Let $|\underline{p}_i\rangle \in \mathbb{Z}^{np\times 1}$ be the one-hot column vector with 1 at position i. $\{\underline{p}_i\}_{i=1}^{np}$ will be our markings or places basis

---

$\underline{x}_i$ = ith **transition node**

$\mathcal{X} = \{\underline{x}_i\}_{i=1}^{nx}$, set of all transition nodes.

Let $|\underline{x}_i\rangle \in \mathbb{Z}^{nx\times 1}$ be the one-hot column vector with 1 at position i. $\{\underline{x}_i\}_{i=1}^{nx}$ will be our transitions basis.

$\underline{x} \to \underline{y}$ denotes an **arrow (a.k.a. directed arc)**. Note that there are two types of arrows: those that go from a transition to a place node and those that go from a place to a transition node. Let $\underline{x} \to \underline{y} = \underline{y} \leftarrow \underline{x} = (\underline{x}, \underline{y})$

$\mathcal{A}$ = the set of all arrows.

$$\mathcal{A} \subset \overline{\mathcal{A}} = (\mathcal{P} \times \mathcal{X}) \cup (\mathcal{X} \times \mathcal{P}) = \{\mathcal{P} \to \mathcal{X} \text{ or } \mathcal{X} \to \mathcal{P} : \underline{p} \in \mathcal{P}, \underline{x} \in \mathcal{X}\} \tag{75.10}$$

A **flow** is a directed path of arrows from $\mathcal{A}$.

$\mu : \mathcal{P} \to \mathbb{Z}_{\geq 0}$ = **marking (i.e., content, number of tokens) of each place**. Sometimes, it is convenient to consider a fractional number of tokens in a place, so $\mu : \mathcal{P} \to \mathbb{R}_{\geq 0}$.

$\kappa : \mathcal{A} \to \mathbb{Z}_{>0}$ = **capacity of each arrow**. If unstated for some $a \in \mathcal{A}$, assume $\kappa(a) = 1$. It's also possible to extend the domain of $\kappa()$ to $\overline{\mathcal{A}}$ and define $\kappa(a) = 0$ for all $a \in \overline{\mathcal{A}} - \mathcal{A}$. Sometimes, it is convenient to consider a fractional arrow capacity, so $\kappa : \mathcal{A} \to \mathbb{R}_{\geq 0}$.

$W^{\underline{p}\to} \in \{0, 1\}^{np\times nx}$. $W^{\underline{p}\to}_{i,j} = 1$ if there is an arrow $\underline{p}_i \to \underline{x}_j$ (exiting $\underline{p}_i$); else $W^{\underline{p}\to}_{i,j} = 0$. More generally, in the case a capacity function $\kappa()$ is given, let

$$W^{\underline{p}\to}_{i,j} = \kappa(\underline{p}_i \to \underline{x}_j) \tag{75.11}$$

In Dirac notation,

$$W^{\underline{p}\to} = \sum_{i,j} \kappa(\underline{p}_i \to \underline{x}_j)|\underline{p}_i\rangle\langle\underline{x}_j| \tag{75.12}$$

$W^{\underline{p}\leftarrow} \in \{0, 1\}^{np\times nx}$. $W^{\underline{p}\leftarrow}_{i,j} = 1$ if there is an arrow $\underline{p}_i \leftarrow \underline{x}_j$ (entering $\underline{p}_i$); else $W^{\underline{p}\leftarrow}_{i,j} = 0$. More generally, in the case a capacity function $\kappa()$ is given, let

$$W^{\underline{p}\leftarrow}_{i,j} = \kappa(\underline{p}_i \leftarrow \underline{x}_j) \tag{75.13}$$

In Dirac notation,

$$W^{\underline{p}\leftarrow} = \sum_{i,j} \kappa(\underline{p}_i \leftarrow \underline{x}_j)|\underline{p}_i\rangle\langle\underline{x}_j| \tag{75.14}$$

$W = W^{\underline{p}\leftarrow} - W^{\underline{p}\rightarrow} \in \{-1, 0, 1\}^{np\times nx}$ This is called the **incidence matrix**. More generally, if a capacity function is given,

$$W_{i,j} = W^{\underline{p}\leftarrow}_{i,j} - W^{\underline{p}\rightarrow}_{i,j} = \kappa(\underline{p}_i \leftarrow \underline{x}_j) - \kappa(\underline{p}_i \rightarrow \underline{x}_j) \tag{75.15}$$

In Dirac notation,

$$W = \sum_{i,j} \left[\kappa(\underline{p}_i \leftarrow \underline{x}_j) - \kappa(\underline{p}_i \rightarrow \underline{x}_j)\right] |\underline{p}_i\rangle\langle\underline{x}_j| \tag{75.16}$$

$\mathcal{P}(\underline{x} \leftarrow) = \{\underline{p} : \underline{p} \rightarrow \underline{x} \in \mathcal{A}\}$ = all $\underline{x}$ **input places (a.k.a. $\underline{x}$ pre-set)**
$\mathcal{P}(\underline{x} \rightarrow) = \{\underline{p} : \underline{x} \rightarrow \underline{p} \in \mathcal{A}\}$ = all $\underline{x}$ **output places (a.k.a. $\underline{x}$ post-set)**
$\mathcal{X}(\underline{p} \leftarrow) = \{\underline{x} : \underline{x} \rightarrow \underline{p} \in \mathcal{A}\}$ = **all $\underline{p}$ input transitions (a.k.a. $\underline{p}$ preset)**
$\mathcal{X}(\underline{p} \rightarrow) = \{\underline{x} : \underline{p} \rightarrow \underline{x} \in \mathcal{A}\}$ = **all $\underline{p}$ output transitions (a.k.a. $\underline{p}$ post-set)**

$|b(0)\rangle_{\mathcal{P}} \in \mathbb{Z}_{\geq 0}^{np\times 1}$ = **initial marking**
$\Phi = \langle \mathcal{P}, \mathcal{X}, W, |b(0)\rangle_{\mathcal{P}}\rangle$ is a **pnet**.

75.1.3 Firing of a Petri Net

Consider a pnet $\Phi = \langle \mathcal{P}, \mathcal{X}, W, |b(0)\rangle_{\mathcal{P}}\rangle$.

By a **firing step** of a transition $\underline{x} \in \mathcal{X}$, we mean the action of removing $\kappa(\underline{p} \rightarrow \underline{x})$ tokens from all $\underline{p} \in \mathcal{P}(\underline{x} \leftarrow)$, and adding $\kappa(\underline{x} \rightarrow \underline{p}')$ tokens to all $\underline{p}' \in \mathcal{P}(\underline{x} \rightarrow)$.

A transition $\underline{x} \in \mathcal{X}$ is **enabled** (i.e., it may fire) for $\underline{p} \in \mathcal{P}(\underline{x} \leftarrow)$, if there are enough tokens in $\underline{p}$ for a firing to be possible; i.e., if $\mu(\underline{p}) \geq \kappa(\underline{p} \rightarrow \underline{x})$. A transition $\underline{x} \in \mathcal{X}$ is **fully enabled** if it is enabled for all $\underline{p} \in \mathcal{P}(\underline{x} \leftarrow)$.

$t \in \mathbb{Z}_{>0}$ is the **firing time**.

$b_j(t)$ = **marking of place** $\underline{p}_j$ at firing time t.

$$|b(t)\rangle_{\mathcal{P}} = \sum_j b_j(t)|\underline{p}_j\rangle \tag{75.17}$$

$a_i(t)$ = how many times transition $\underline{x}_i$ occurs at time t (i.e., the **strength of transition** $\underline{x}_i$ at firing time t).

$$|a(t)\rangle_{\mathcal{X}} = \sum_i a_i(t)|\underline{x}_i\rangle \tag{75.18}$$

Note that we use subscripts $\mathcal{P}$ and $\mathcal{X}$ to indicate which basis, either the $\{|\underline{p}_i\rangle\}_{i=1}^{np}$ or $\{|\underline{x}_i\rangle\}_{i=1}^{nx}$ we are referring to.

(Original) Firing rule for transition vector $|a(t)\rangle_{\mathcal{X}}$.

$$b_j(t+1) = b_j(t) + \sum_i W_{j,i}(t)a_i(t) \tag{75.19}$$

or, equivalently,

$$|b(t+1)\rangle_{\mathcal{P}} = |b(t)\rangle_{\mathcal{P}} + W|a(t)\rangle_{\mathcal{X}} \tag{75.20}$$

where $W(t) = 0$ if the transition is not fully enabled.

In general, can replace $W|a(t)\rangle_{\mathcal{X}}$ by any vector $|A(t)\rangle_{\mathcal{P}}$ that can depend on the place markings $\mu()$ for all places and arrow capacities $\kappa()$ for all arrows.

$|a(0)\rangle, |a(1)\rangle, |a(2)\rangle, \ldots$ is the **sequence of firing steps**. Often we fire a single transition node in each step, instead of firing a linear combination of them. In such a case, the sequence of firing steps reduces to $a_{i(0)}, a_{i(1)}, a_{i(2)}, \ldots$

A marking $|b^*\rangle_{\mathcal{P}}$ is **reachable** from a marking $|b\rangle_{\mathcal{P}}$ in $n = 1, 2, \ldots$ steps if $|b\rangle_{\mathcal{P}}$ can be transformed to $|b^*\rangle_{\mathcal{P}}$ by executing n firings. We write $|b\rangle_{\mathcal{P}} \xrightarrow[\Phi]{} |b^*\rangle_{\mathcal{P}}$ and say a marking $|b^*\rangle_{\mathcal{P}}$ is reachable from a marking $|b\rangle_{\mathcal{P}}$ if $|b^*\rangle_{\mathcal{P}}$ is reachable from $|b\rangle_{\mathcal{P}}$ in a finite number of steps. The **reachability set** of a Petri Net Φ with initial marking $|b(0)\rangle_{\mathcal{P}}$ is defined as

$$R(\Phi) = \{|b^*\rangle_{\mathcal{P}} : \ |b(0)\rangle_{\mathcal{P}} \xrightarrow[\Phi]{} |b^*\rangle_{\mathcal{P}} \ \} \tag{75.21}$$

75.2 Variants

75.2.1 Finite State Machine and Turing Machine

A **Finite State Machine** (FSM) can be viewed as a type of pnet that has

1. exactly one token moving from a place node to another place node with each step,

2. transitions which only have a single incoming and outgoing arrow

We discuss FSMs in detail in Chapter 30.

A **Turing Machine** (TM) can also be viewed as a type of pnet that satisfies 1 and 2 above. We discuss TMs in detail in Chapter 107.

75.2.2 Continuous Petri Net

So far we have considered discrete time $t \in \mathbb{Z}_{\geq 0}$. If we let the interval $\Delta t = t_{i+1} - t_i$ between succesive events tend to zero and if we replace W by $W/\Delta t$ in Eq.(75.20), we get a **continuous Petri net** obeying:

$$\frac{db_j(t)}{dt} = \sum_i W_{j,i} a_i(t) \tag{75.22}$$

$$\frac{d|b(t)\rangle_{\mathcal{P}}}{dt} = W|a(t)\rangle_{\mathcal{X}} \tag{75.23}$$

Figure 75.3: Representation of Eq.(75.22). The place nodes (but not the transition nodes) of a continuous pnet carry a time derivative.

We extend W to

$$\overline{W} = \begin{array}{c|cc} & \underline{p}_j & \underline{x}_i \\ \hline \underline{p}_j & 0 & W_{j,i} \\ \underline{x}_i & -W_{j,i} & 0 \end{array} \tag{75.24}$$

The extended matrix $\overline{W}$ is an anti-symmetric square matrix.

Eq.(75.22) can be represented by the bnet (see Fig.75.3). Thus, a continuous pnet can be viewed as np linear bnets, one for each time derivative $db_i(t)/dt$ for $i = 1, 2, \ldots, np$. Then the leg nodes of those np different bnets are joined if they carry the same transition $a_j(t)$. One writes an equation for each place node (time derivative), but not for each transition node.

Aside, for those who know Quantum Mechanics:

Note that if we set $H = i\overline{W}$, since $\overline{W}$ is anti-symmetric, it follows that H is Hermitian ($H = H^\dagger$). Consider Schoedinger's equation (on a finite, $np + nx$ dimensional space) with the Hamiltonian H

$$\frac{d|\psi(t)\rangle}{dt} = -iH|\psi(t)\rangle \tag{75.25}$$

If we set

$$|\psi(t)\rangle = \begin{bmatrix} |b(t)\rangle_{\mathcal{P}} \\ |a(t)\rangle_{\mathcal{X}} \end{bmatrix} \tag{75.26}$$

then the first np equations are identical to Eq.(75.23)

75.2.3 Colored Petri Net

In an original pnet, the tokens are all of the same kind, so the marking (i.e., content) of every place node is given by a scalar from either $\mathbb{Z}_{\geq 0}$ or $\mathbb{R}_{\geq 0}$. In an original pnet,

the messages that flow through the arrows are scalars too. But if one considers tokens of various kinds, then one needs a vector or list to store their numbers in each place node. This leads to the definition of a **colored pnet**, as a pnet wherein place node markings and arrow messages become lists, or matrices, or tensors, or even the objects of a class.[1]

75.2.4 Stochastic Petri Net

For **stochastic pnets**, the components $a_i(t)$ of the transition strength vector $|a(t)\rangle_{\mathcal{X}}$ are random variables rather than deterministic functions. What this means is that if we measure the value of a_i at time t, we don't get a definite value, but rather a random one that obeys a probability distribution. For example, one might have

$$P[\underline{a}_i(t) = a_i(t)] = \lambda_i e^{-\lambda_i a_i(t)} \quad (a_i(t), \lambda_i > 0, \text{ exponential distribution}) \tag{75.27}$$

for $i = 1, 2, \ldots, nx$.

Sometimes, one considers a **hybrid stochastic pnet** in which some transitions are deterministic and fire immediately (these are called **intermediate transitions**), whereas other transitions fire at random times (these are called **timed transitions**).

Note that $|\underline{a}(t)\rangle_{\mathcal{X}}$ is a random variable and $|b(t)\rangle_{\mathcal{P}}$ is expressed in terms of $|\underline{a}(t)\rangle_{\mathcal{X}}$, so $|b(t)\rangle_{\mathcal{P}}$ becomes a random variable too.

When Fig.75.3 refers to a deterministic continuous bnet, all nodes are deterministic. On the other hand, when it refers to a stochastic continuous bnet, all nodes are random variables with TPMs.

A stochastic pnet can be either discrete or continuous in time. In the continuous in time case, $\underline{a}_i(t)$ is a **stochastic process**; i.e., a random variable that depends on continuous time. (See Chapter 94). Let SOSE=system of ordinary differential equations. Whereas a deterministic continuous pnet represents a <u>SODE for deterministic variables</u>, a stochastic continuous pnet represents a <u>SODE for stochastic processes</u>.

75.2.5 Bayes-Petri Net

Given a bnet, one can construct a natural pnet, call it the bnet's **Bayes Petri Net** (Bayes pnet), that uses the nodes of the bnet as the transition nodes of the Bayes pnet. We will refer to this construction as **petrifying a bnet**.

See Fig.75.4 (a). Inspired by Pearl's message passing theory (see Chapter 60), we petrify a single arrow bnet $\underline{x} \to \underline{y}$ by adding to it two pnet arcs:

- one pnet arc pointed "downstream" with respect to the bnet arrow $\underline{x} \to \underline{y}$. This arc carries probabilities $\pi(x) = P(x|\epsilon^-)$

[1]The objects or instances of a class are defined in OOP (Object Oriented Programming).

Figure 75.4: Petrifying an arrow by adding to it arcs that transmit either matrix messages (in (a)) or scalar messages (in (b)).

- and another pnet arc pointed "upstream" with respect to the bnet arrow $\underline{x} \to \underline{y}$. This arc carries probabilities $\lambda(x) = P(\epsilon^+|x)$.

(b)

Figure 75.5: Example of petrifying a bnet with five arrows, not just one arrow as in Fig.75.4

Studying **matrix** message passing (colored pnet) as in Fig.75.4 (a) can be considered the ultimate goal of the theory of Bayes pnets, but as a first step in that direction, we will discuss **scalar** message passing (uncolored pnet) as in Fig.75.4 (b). Just like the proponents of the original pnets considered the simpler case of uncolored pnets first, and later graduated to the colored case, we will do the same for Bayes pnets.

In Fig.75.5, we give an example of petrifying a bnet with 5 arrows, instead of just one arrow as in Fig.75.4.

Henceforth in this chapter, the conditioned nodes of a bnet will be colored yellow.

Figure 75.6: What happens to an incoming downstream message impinging on bnet node $\underline{x}$, with $\underline{x}$ conditioned on or not conditioned on.

Figure 75.7: What happens to an incoming downstream message impinging on bnet node $\underline{x}$, in case $\underline{x}$ isn't conditioned on but one of its descendants is.

The **firing rules** for a Bayes pnet come directly from Pearl's d-separation rules which are discussed in Chapter 24. Figs.75.6, 75.7 and 75.8 are 3 cases that must be considered in the d-separation motivated firing rules. Since the bnet nodes and pnet transition nodes of a Bayes pnet are the same, one natural way to fire the pnet nodes is to fire them one at a time, in topological order.

For more information about Bayes pnets, check out my software called "Bayes_Petri_Net" (Ref.[86]) that implements some of the concepts about Bayes pnets presented in this section.

Figure 75.8: What happens to an incoming upstream message impinging on bnet node $\underline{x}$, with $\underline{x}$ conditioned on or not conditioned on.

Chapter 76

Plate Notation

In this chapter, we will use the Numpy-like tensor notation discussed in Section C.48. In particular, note that $[n] = [0 : n] = \{0, 1, \ldots, n-1\}$ and that $T^{[n],[m]}$ is an $n \times m$ matrix.

Plate notation is often used in Machine Learning. See, for instance, Chapter 104 on Transformers Networks, for examples of plate notation.

Plate notation is used to describe, in a compact way, a family of equal, disjoint sub-bnets of a bnet that are connected in parallel or in series. Suppose you have a bnet containing as a subset, Λ disjoint node sets S_λ ("sub-bnets"), where $\lambda \in [\Lambda]$. Suppose any two S_λ have the same number of equivalent nodes, and two equivalent nodes have the same TPM.

In case the S_λ are **connected in parallel (CIP)**: Rather than drawing all Λ sets, we think of them as **layers** of a **stack** that come out of the page like a stack of pancakes with the pancakes lying flat on the page. That way we only have to draw one pancake instead of Λ. We only draw once instead of Λ times, each node and the arrows entering and exiting that node. Quite a saving in labor and bnet complexity! And a bnet can have more than one plate, and a node can belong to more than one plate!

In case the S_λ are **connected in series (CIS)**: Rather than drawing all Λ sets, we think of them as **links** in a **chain**. That way we only have to draw one link instead of Λ. We only draw once instead of Λ times, each node and the arrows entering and exiting that node.

The simplest possible use of CIS plates is for representing a Markov chain. This is illustrated in Fig.76.1.

Fig.76.2 gives an example of a bnet with 2 nested CIP plates[1]. The TPMs for this bnet are of the following form (we print them in blue).

$$P(A^{[20]}|Q^{[20],[10]}, K^{[20],[10]}, V^{[20],[10]}) = \tag{76.1a}$$

[1]Fig.76.2 and the blue TPMs were rendered with my free, open source software `texnn` (see Ref.[98]) `texnn` can keep track of the tensor shapes of each node, for bnets with one or more plates.

Figure 76.1: 3-link Markov chain represented in plate notation and without plates.

Figure 76.2: Example of a bnet with 2 nested CIP plates. In general, multiple plates need not be nested.

$$P(K^{[20],[10]}|X) = \tag{76.1b}$$

$$P(Q^{[20],[10]}|X) = \tag{76.1c}$$

$$P(V^{[20],[10]}|X) = \tag{76.1d}$$

$$P(X) = \tag{76.1e}$$

$$P(Y|X, A^{[20]}) = \tag{76.1f}$$

Chapter 77

Potential Outcomes and Beyond

This chapter is based on Ref.[15], a book by Stephen Cunningham entitled "Causal inference: the mixtape".

The theory of potential outcomes (PO) was for the most part invented in a seminal 1974 paper by Donald B. Rubin. Rubin has also made important extensions to PO theory since 1974. However, he does not use Pearl's causal DAGs to discuss PO theory. Pearl has shown that PO theory can be substantially clarified and extended by using the language of causal DAGs. The d-separation theorem and do operator that we discuss in Chapters 24 and 77 are especially useful in this regard. In this chapter, we stress the connection of PO theory to Pearl's causal DAGs and bnets.

σ	d^σ	y^σ	$y^\sigma(0)$	$y^\sigma(1)$
Edith	0	5	5	.
Frank	0	7	7	.
George	0	8	8	.
Hank	0	10	10	.
Andy	1	10	.	10
Ben	1	5	.	5
Chad	1	16	.	16
Daniel	1	3	.	3

Table 77.1: PO dataset describing whether individual σ took a treatment drug ($d^\sigma = 1$) or didn't ($d^\sigma = 0$). The treatment outcome is measured by the real number y^σ.

Suppose a **population of individuals** $\sigma = 0, 1, 2, \ldots, nsam - 1$ is given ($d^\sigma = 1$) or is not given ($d^\sigma = 0$) a **treatment dose** (i.e., **treatment decision**) d^σ, and that the **treatment outcome** (i.e., **treatment response**) is measured by a real number y^σ. Table 77.1 gives a possible **PO dataset** for this scenario. As you can see from that table, each individual either takes a drug or doesn't, but not both. PO theory can be viewed as a **missing data (MD) problem**. MD problems are discussed in Chapter 63. However, the PO MD problem is much more specialized than the generic MD problems discussed in Chapter 63. In the PO MD problem, we

can fill in the blank cells by matching each individual that took the drug with another *similar* individual that didn't. We will have much more to say about this matching strategy later in this chapter.

One can define similar individuals as individuals that have the same value for nx features $x^\sigma = (x_i^\sigma)_{i=0,1,\ldots,nx-1}$. One can add to Table 77.1 nx extra columns giving the value of the feature vector x^σ for each individual. Members of a population with the same x^σ are referred to as a **subpopulation or stratum (i.e., layer)**.

In a **randomized controlled trial (RCT)** [1], the effect of the variable x^σ on the value of d^σ is eliminated by randomizing the population and therefore making the effect of x^σ on d^σ average out to zero. However, there are many situations in which carrying out an RCT is not possible or desirable. PO theory is a way of predicting the result of an RCT in situations where doing a real RCT is not possible or desirable.

In this chapter, x^σ will be called the confounders. Implicit throughout this chapter is the assumption that there are **no unmeasured confounders**. Because if there are some unmeasured confounders, those can send secret messages that influence the value that d^σ takes. This would ruin the predictions of someone trying to predict the results of an RCT without being privy to those secret messages. When there are **some unmeasured confounders**, it might still be possible to predict the effect of an RCT. This might be possible using instrumental variables. See Chapter 44 for a discussion of **instrumental variables**.

77.1 G and G_{den} bnets, the starting point bnets

Figure 77.1: Bnets G and G_{den} are our starting point in discussing PO theory. G is for a single individual σ of the population. Bnet G_{den} is the DEN counterpart to G. DEN (Deterministic with External Noise) bnets are discussed in Chapter 52.

[1] The term **A/B test** is often used to mean an RCT where A and B are the treated and control groups. However, sometimes the term is used to refer to an experiment that conditions on confounders, which violates the definition of an RCT, and is the same as a PO test.

In this chapter, we will abbreviate $\underline{X}[\sigma] = \underline{X}^\sigma$ for $X \in \{d, x, y\}$, where $\sigma \in \{0, 1, 2, \ldots, nsam - 1\}$.

For each individual (a.k.a. unit, sample) $\sigma = 0, 1, 2, \ldots nsam - 1$, let:

$\underline{d}^\sigma \in \{0, 1\}$ be the treatment dose or drug dose. It equals 1 if treated and 0 if untreated.

$\underline{y}^\sigma \in \mathbb{R}$ be the treatment potential outcome

$\underline{x}^\sigma$ be the column vector of treatment confounders (a.k.a. covariates because they are often used as covariates (i.e., independent variables) in linear regression.)

Consider bnets G and G_{den} in Fig.77.1. G reflects the language used in Ref.[15] to discuss PO theory. And G_{den} reflects the language that Judea Pearl prefers to use to discuss PO theory. Both languages are equivalent. To go from one language to the other, one need only perform the following swaps, where $\underline{u}$ is the external noise of the DEN bnet.

$\underline{X}^\sigma \leftrightarrow \underline{X}(\underline{u})$ for $X \in \{d, x, y\}$.

$P(\sigma) = \frac{1}{nsam} \leftrightarrow P(u)$

$\sum_\sigma P(\sigma)(\cdot) \leftrightarrow \sum_u P(u)(\cdot)$

The TPMs, printed in blue, for the bnet G in Fig.77.1, are as follows:

$$P(x^\sigma) = P_{\underline{x}}(x^\sigma) \tag{77.1}$$

$$P(d^\sigma | x^\sigma) = P_{\underline{d}|\underline{x}}(d^\sigma | x^\sigma) \tag{77.2}$$

$$P(y^\sigma | d^\sigma, x^\sigma) = P_{\underline{y}|\underline{d},\underline{x}}(y^\sigma | d^\sigma, x^\sigma) \tag{77.3}$$

Now let:

$\underline{d} \in \{0, 1\}$ be the treatment dose. It equals 1 if treated and 0 if untreated

$\underline{y} \in \mathbb{R}$ be the treatment potential outcome

$\underline{x}$ be the column vector of treatment confounders (a.k.a. covariates)

$\underline{u} = (\underline{u}_{\underline{d}}, \underline{u}_{\underline{x}}, \underline{u}_{\underline{y}})$ be the external noise

The TPMs, printed in blue, for the bnet G_{den} in Fig.77.1, are as follows:

$$P(x | u_{\underline{x}}) = \mathbb{1}(\ \ x = u_{\underline{x}}\ \ ) \tag{77.4}$$

$$P(d | x, u_{\underline{d}}) = \mathbb{1}(\ \ d = f_{\underline{d}}(x, u_{\underline{d}})\ \ ) \tag{77.5}$$

$$P(y | d, x, u_{\underline{y}}) = \mathbb{1}(\ \ y = f_{\underline{y}}(d, x, u_{\underline{y}})\ \ ) \tag{77.6}$$

If we linearize $f_{\underline{y}}$ in Eq.(77.6), we get

$$\underline{y} = \delta \underline{d} + \beta \underline{x} + \underline{u}_{\underline{y}} \ , \tag{77.7}$$

where $\delta, \beta \in \mathbb{R}$. Assuming that $\underline{x}, \underline{y} \in \mathbb{R}$ and $\underline{d} \in \{0, 1\}$, Eq.(77.7) can be plotted. The resulting plot is given in Fig.77.2. This plot is a very special case of the PO problem, but it gives a crude idea of the "effects" $\delta = y(1) - y(0)$ that PO theory gives estimates for. Any individual participating in the experiment experiences either $y(1)$ or $y(0)$, but not both.

Figure 77.2: Plot of Eq.(77.7)

77.2 G bnet with nodes $y^\sigma(0), y^\sigma(1)$ added to it.

Figure 77.3: Bnet G_+ is bnet G with two new variables $\underline{y}^\sigma(0)$ and $\underline{y}^\sigma(1)$ added to it as a tuple node $[\underline{y}^\sigma(0), \underline{y}^\sigma(1)]$.

Consider Fig.77.3. Bnet G_+ was obtained by adding a tuple node $[\underline{y}^\sigma(0), \underline{y}^\sigma(1)]$ to bnet G. Fig.77.4 shows 3 equivalent ways of adding the variables $\underline{y}^\sigma(0)$, $\underline{y}^\sigma(1)$ to

Figure 77.4: $(a), (b), (c)$ are 3 equivalent ways of adding the variables $\underline{y}^\sigma(0)$, $\underline{y}^\sigma(1)$ to graph G to obtain G_+ in Fig.77.3. Note that $c^\sigma \in \{0, 1\}$.

graph G to obtain G_+. The TPMs, printed in blue, for bnet G_+, are as follows. Note that we define them in terms of the TPMs for bnet G.

$$P(x^\sigma) = P_{\underline{x}}(x^\sigma) \tag{77.8}$$

$$P(d^\sigma|x^\sigma) = P_{\underline{d}|\underline{x}}(d^\sigma|x^\sigma) \tag{77.9}$$

For $c \in \{0, 1\}$,

$$P(y^\sigma(c)|d^\sigma, x^\sigma) = \begin{cases} P_{\underline{y}(c)|\underline{d},\underline{x}}(y^\sigma(c)|d^\sigma, x^\sigma) & \text{if INCLUDE arrow with question mark} \\ P_{\underline{y}(c)|\underline{x}}(y^\sigma(c)|x^\sigma) & \text{if EXCLUDE arrow with question mark} \end{cases} \tag{77.10}$$

$$\begin{aligned} P(y^\sigma|y^\sigma(0), y^\sigma(1), d^\sigma) = &= \mathbb{1}(y^\sigma = d^\sigma y^\sigma(1) + (1 - d^\sigma)y^\sigma(0)) && \text{(77.11a)} \\ &= \mathbb{1}(y^\sigma = y^\sigma(d^\sigma)) && \text{(77.11b)} \end{aligned}$$

Eq.(77.11) is often referred to as the **SUTVA or Consistency assumption**.

If we sum over the nodes $\underline{y}(0)$ and $\underline{y}(1)$ of this bnet, we should get the bnet G. This is easy to check. Indeed,[2]

[2]To convince yourself that Eq.(77.13) is correct, note this. If $d^\sigma = 0$, you can sum over $y^\sigma(1)$ firstly, thus setting $\sum_{y^\sigma(1)} P(y^\sigma(1)|d^\sigma, x^\sigma) = 1$. Secondly, you can sum over $y^\sigma(0)$.

$$P(y^\sigma|d^\sigma,x^\sigma) = \sum_{y^\sigma(0)}\sum_{y^\sigma(1)} \mathbb{1}(y^\sigma = y^\sigma(d^\sigma))P(y^\sigma(0)|d^\sigma,x^\sigma)P(y^\sigma(1)|d^\sigma,x^\sigma) \tag{77.12}$$

$$= \begin{cases} P_{\underline{y}(0)|\underline{d},\underline{x}}(y^\sigma|d^\sigma,x^\sigma) & \text{if } d^\sigma = 0 \\ P_{\underline{y}(1)|\underline{d},\underline{x}}(y^\sigma|d^\sigma,x^\sigma) & \text{if } d^\sigma = 1 \end{cases} . \tag{77.13}$$

Henceforth, we will refer to the case where the question mark arrow is included as the general case, and to the case when it's excluded as the **weak-d limit**. Henceforth, we will first present results for the general case, and then describe how those results change for the weak-d limit. Rubinologists always assume the weak-d limit, but we find that with little effort, we can derive many results for general case, and then compare those results to their weak-d limit. I find such comparisons instructive.

Note that in the general case, $P(\underline{y}(c) = y|\underline{d} = d, x)$ for $c, d \in \{0, 1\}$ are four different probability distributions, and that $P(\underline{y} = y|d = d, x)$ is defined in terms of two of them, the so called **factual distributions** with $c = d$. By measuring $\underline{y}$, we cannot access the other 2 probability distributions, the so called **counter-factual distributions** with $c \neq d$.

In the weak-d limit, $P(\underline{y}(c) = y|\underline{d} = d, x) = P(\underline{y}(c) = y|x)$ are two probability distributions, and they both can be accessed by measuring $\underline{y}$.

77.3 Expected Values of treatment outcome y^σ

It is convenient to define the following expected values of $\underline{y}^\sigma$ in terms of the TPMs of bnet G_+:

$$\mathcal{Y}_{c|d,x} = E_{\sigma|d,x}[\underline{y}^\sigma(c)] \to E_{\underline{y}(c)|d,x}[\underline{y}(c)] = \sum_y yP(\underline{y}(c) = y|\underline{d} = d, x) \tag{77.14}$$

$$\mathcal{Y}_{c|d} = E_{\sigma|d}[\underline{y}^\sigma(c)] \to E_{\underline{y}(c)|d}[\underline{y}(c)] = \sum_x \mathcal{Y}_{c|d,x}P(x|d) \tag{77.15}$$

$$\mathcal{Y}_{c|x} = E_{\sigma|x}[\underline{y}^\sigma(c)] \to E_{\underline{y}(c)|x}[\underline{y}(c)] = \sum_d \mathcal{Y}_{c|d,x}P(d|x) \tag{77.16}$$

$$\mathcal{Y}_c = E_\sigma[\underline{y}^\sigma(c)] \to E_{\underline{y}(c)}[\underline{y}(c)] = \sum_{x,d} \mathcal{Y}_{c|d,x}P(x,d) \tag{77.17}$$

Note that in the weak-d limit,

$$\mathcal{Y}_c = \sum_x \mathcal{Y}_{c|d,x}P(x) . \tag{77.18}$$

Note also that in the weak-d limit, $\mathcal{Y}_{c|d,x}$ is independent of d, but $\mathcal{Y}_{c|d}$ can depend on d if $P(x|d)$ depends on d.

For $\mathcal{Y}_{c|d}$, the expected values $\mathcal{Y}_{0|0}, \mathcal{Y}_{1|1}$ are said to be **factual** (indicating compliant patients) whereas $\mathcal{Y}_{0|1}, \mathcal{Y}_{1|0}$ are said to be **counterfactual** (indicating non-compliant patients).

Also let

$$\mathcal{Y}_{|d,x} = E_{\sigma|d,x}[\underline{y}^\sigma] \to E_{\underline{y}|d,x}[\underline{y}] = \sum_y yP(\underline{y} = y|\underline{d} = d, x) \tag{77.19}$$

$$\mathcal{Y}_{|d} = E_{\sigma|d}[\underline{y}^\sigma] \to E_{\underline{y}|d}[\underline{y}] = \sum_x \mathcal{Y}_{|d,x}P(x|d) \tag{77.20}$$

$$\mathcal{Y}_{|x} = E_{\sigma|x}[\underline{y}^\sigma] \to E_{\underline{y}|x}[\underline{y}] = \sum_d \mathcal{Y}_{|d,x}P(d|x) \tag{77.21}$$

$$\mathcal{Y} = E_\sigma[\underline{y}^\sigma] \to E_{\underline{y}}[\underline{y}] = \sum_d \mathcal{Y}_{|d,x}P(d, x) \tag{77.22}$$

In the weak-d limit, $\mathcal{Y}_{|d,x}$ is independent of d, but $\mathcal{Y}_{|d}$ can still depend on d if $P(x|d)$ depends on d.

77.4 Translation Dictionary

In standard PO notation	In our notation
i, individual (i.e., unit, sample) index	σ
$D_i = d_i$, treatment dose	$\underline{d}^\sigma = d^\sigma$
$Y_i = y_i$, treatment outcome	$\underline{y}^\sigma = y^\sigma$
$X_i = x_i$, treatment confounders	$\underline{x}^\sigma = x^\sigma$
$E[Y_i(c)]$	$E_\sigma[\underline{y}^\sigma(c)] = \mathcal{Y}_c$
$E[Y_i(c)\|D_i = d]$	$E_{\sigma\|d}[\underline{y}^\sigma(c)] = \mathcal{Y}_{c\|d}$
$E[Y_i(c)\|D_i = d, X_i = x]$	$E_{\sigma\|d,x}[\underline{y}^\sigma(c)] = \mathcal{Y}_{c\|d,x}$
$E[Y_i]$	$E_\sigma[\underline{y}^\sigma] = \mathcal{Y}$
$E[Y_i\|D_i = d]$	$E_{\sigma\|d}[\underline{y}^\sigma] = \mathcal{Y}_{\|d}$
$E[Y_i\|D_i = d, X_i = x]$	$E_{\sigma\|d,x}[\underline{y}^\sigma] = \mathcal{Y}_{\|d,x}$

Table 77.2: Dictionary for translating from standard PO notation of Ref.[15] to our notation. $c, d \in \{0, 1\}$.

Table 77.2 gives a dictionary for translating from the standard PO notation of Ref.[15] to our notation.

77.5 $\mathcal{Y}_{|d,x} = \mathcal{Y}_{d|d,x}$ (SUTVA)

Claim 124 [3]

$$\mathcal{Y}_{|d,x} = \mathcal{Y}_{d|d,x} \tag{77.23}$$

$$\mathcal{Y}_{|d} = \mathcal{Y}_{d|d} \tag{77.24}$$

proof:

$$\begin{aligned} \mathcal{Y}_{|d,x} &= \sum_y yP(\underbrace{\underline{y}}_{=\underline{y}(d) \text{ by Eq.(77.11)}} = y|d,x) && (77.25) \\ &= \mathcal{Y}_{d|d,x} \ . && (77.26) \end{aligned}$$

Applying $\sum_x P(x|d)$ to both sides of Eq.(77.23) gives Eq.(77.24).
QED

77.6 Conditional Independence Assumption (CIA)

The **Conditional Independence Assumption** (CIA) is said to hold if[4]

$$(\underline{y}^\sigma(0), \underline{y}^\sigma(1)) \perp_P^{sep} \underline{d}^\sigma | \underline{x}^\sigma \ . \tag{77.27}$$

This is satisfied by G_+ in the weak-d limit. This follows immediately by applying the d-separation theorem (see Chapter 24) to G_+. Indeed, conditioning on $\underline{x}^\sigma$ blocks signals from entering $[\underline{y}^\sigma(0), \underline{y}^\sigma(1)]$ through the path that passes through $\underline{x}^\sigma$. Furthermore, $\underline{y}^\sigma$ acts as a collider, blocking the path to $[\underline{y}^\sigma(0), \underline{y}^\sigma(1)]$ going through $\underline{y}^\sigma$.

I think CIA only makes sense if the individuals are treatment blind (i.e., have no knowledge of whether they are in the treated or control groups.) Otherwise, that extra knowledge becomes a confounder not being included in x.

A **Randomized Controlled Trial (RCT)** is defined to satisfy Eq.(77.27) without the $\underline{x}^\sigma$ conditioning; i.e., it satisfies

$$(\underline{y}^\sigma(0), \underline{y}^\sigma(1)) \perp_P^{sep} \underline{d}^\sigma \ . \tag{77.28}$$

This means that in an RCT, the arrow from $\underline{x}^\sigma$ to $\underline{d}^\sigma$ in G_+ is omitted.

[3] In the standard PO notation, this is the frequently used identity

$$E[Y|D=d,x] = E[Y(d)|D=d,x]$$

.

[4] See Section C.13 for definition of $\perp_P$ and $\perp_P^{sep}$

Note that if we assume both CIA (i.e., weak-d limit) and SUTVA, we get

$$
\begin{aligned}
\mathcal{Y}_{|d,x} &= E_{\sigma|d,x}[\underline{y}^\sigma] && (77.29a)\\
&= E_\sigma[\underline{y}^\sigma|\underline{d}^\sigma = d, \underline{x}^\sigma = x] && (77.29b)\\
&= E_\sigma[\underline{y}^\sigma(d)|\underline{d}^\sigma = d, \underline{x}^\sigma = x] \text{ (by SUTVA)} && (77.29c)\\
&= E_\sigma[\underline{y}^\sigma(d)|\underline{x}^\sigma = x] \text{ (by CIA)} && (77.29d)\\
&= E_{\sigma|x}[\underline{y}^\sigma(d)] && (77.29e)\\
&= \mathcal{Y}_{d|x} \ . && (77.29f)
\end{aligned}
$$

In an RCT, Eq.(77.29f) is valid without the x conditioning.

77.7 Treatment Effects

Figure 77.5: Different treatment effects. A treatment effect is a difference of two $\mathcal{Y}_{c|d}$.

Figure 77.6: Alternative representation of the same information that is contained in Fig.77.5.

A **treatment effect** is a difference of two $\mathcal{Y}_{c|d}$. It is convenient to define the following treatment effects. See Figs.77.5 and 77.6.

- average treatment effect (ATE).

$$ATE = \mathcal{Y}_1 - \mathcal{Y}_0 = \delta \tag{77.30}$$

- average treatment effect of the treated (ATT)

$$ATT = \mathcal{Y}_{1|1} - \mathcal{Y}_{0|1} \tag{77.31}$$

- average treatment effect of the untreated (ATU)

$$ATU = \mathcal{Y}_{1|0} - \mathcal{Y}_{0|0} \tag{77.32}$$

- selection bias (SB)

$$SB = \mathcal{Y}_{0|1} - \mathcal{Y}_{0|0} \tag{77.33}$$

- simple difference in outcomes (SDO)

$$SDO = \mathcal{Y}_{1|1} - \mathcal{Y}_{0|0} \tag{77.34}$$

Let

$$\pi_d = P(\underline{d} = d) \tag{77.35}$$

for $d \in \{0, 1\}$.

Note that there exist some linear constraints between these treatment effects.

$$\begin{aligned} \underbrace{\mathcal{Y}_1 - \mathcal{Y}_0}_{ATE} &= \underbrace{\mathcal{Y}_{1|1}\pi_1 + \mathcal{Y}_{1|0}\pi_0}_{\mathcal{Y}_1} - \underbrace{(\mathcal{Y}_{0|1}\pi_1 + \mathcal{Y}_{0|0}\pi_0)}_{\mathcal{Y}_0} && (77.36) \\ &= \underbrace{(\mathcal{Y}_{1|1} - \mathcal{Y}_{0|1})}_{ATT}\pi_1 + \underbrace{(\mathcal{Y}_{1|0} - \mathcal{Y}_{0|0})}_{ATU}\pi_0 && (77.37) \end{aligned}$$

$$\underbrace{\mathcal{Y}_{1|1} - \mathcal{Y}_{0|0}}_{SDO} = \underbrace{(\mathcal{Y}_{1|1} - \mathcal{Y}_{0|1})}_{ATT} + \underbrace{\mathcal{Y}_{0|1} - \mathcal{Y}_{0|0}}_{SB} \tag{77.38}$$

$$\begin{aligned} \underbrace{\mathcal{Y}_{1|1} - \mathcal{Y}_{0|0}}_{SDO} &= \underbrace{(\mathcal{Y}_{1|1} - \mathcal{Y}_{0|1})\pi_1 + (\mathcal{Y}_{1|0} - \mathcal{Y}_{0|0})\pi_0}_{ATE} \\ &\quad + \underbrace{\mathcal{Y}_{0|1} - \mathcal{Y}_{0|0}}_{SB} \\ &\quad + \underbrace{(\mathcal{Y}_{1|1} - \mathcal{Y}_{0|1})}_{ATT}\pi_0 \\ &\quad - \underbrace{(\mathcal{Y}_{1|0} - \mathcal{Y}_{0|0})}_{ATU}\pi_0 && (77.39) \end{aligned}$$

By virtue of Eq.(77.37),

$$ATT = ATU \implies ATT = ATU = ATE \tag{77.40}$$

and

$$ATE = 0 \iff \frac{ATU}{ATT} = -\left(\frac{\pi_1}{\pi_0}\right) . \tag{77.41}$$

Whenever $ATT = ATU$, we will say there is **T-U symmetry**.

In general, $SDO = ATT + SB$, but if there is T-U symmetry, then $SDO = ATE + SB$.

If there is T-U symmetry and zero bias $SB = 0$, then $SDO = ATE = ATT = ATU$.

If there is a null result for an RCT (i.e., $ATE = 0$), T-U symmetry and zero bias $SB = 0$, then $SDO = ATE = ATT = ATU = 0$.

Let

$$\mathcal{Y}_{c,d|x} = \mathcal{Y}_{c|d,x} P(d|x) \tag{77.42}$$

For each $\mathcal{E} \in \{ATE, ATT, ATU, SB, SDO\}$, we can define its restriction $\mathcal{E}_x$ to a fixed stratum x by replacing each $\mathcal{Y}_{c|d}$ with $\mathcal{Y}_{c,d|x}$. For example,

$$ATT = \mathcal{Y}_{1|1} - \mathcal{Y}_{0|1} \text{ so } ATT_x = \mathcal{Y}_{1,1|x} - \mathcal{Y}_{0,1|x} . \tag{77.43}$$

We can calculate $\mathcal{E}$ from $\mathcal{E}_x$ using

$$\mathcal{E} = E_x[\mathcal{E}_x] = \sum_x P(x)\mathcal{E}_x . \tag{77.44}$$

77.8 Insights into what makes treatment effects equal and $\mathcal{Y}_{1|0} = \mathcal{Y}_1$

1. Is it possible for $SDO = 0$ but $ATE \neq 0$ or vice versa, and what is going on when this is true?

2. What is going on when two treatment effects are equal; for instance, when $ATT = ATU$?

3. When is $\mathcal{Y}_{1|0} = \mathcal{Y}_1$, and what is going on when this is true?

Fig.77.7 gives some intuition about what is going on when any of these things happen.

Recall that each expected value $\mathcal{Y}_{c|d}$ is associated with a probability distribution $P_{\underline{y}(c)|\underline{d},x}$.

Figure 77.7: Figure 77.5 with added information about probability distributions used to obtain each expected value $\mathcal{Y}_{c|d}$.

$$\mathcal{Y}_{c|d} = \sum_y y \underbrace{\sum_x P_{\underline{y}(c)|\underline{d},x}(y|d,x)P(x|d)}_{P_{\underline{y}(c)|\underline{d}}(y|d)} \tag{77.45}$$

for $c, d \in \{0,1\}$. Fig.77.7 reminds us of which P is used to generate each $\mathcal{Y}$. From this figure, we see that

1. A sufficient condition for $SDO = 0$ is that $P_{\underline{y}(1)|\underline{d}=1} = P_{\underline{y}(0)|\underline{d}=0}$. A sufficient condition for $ATE = 0$ is that $P_{\underline{y}(1)|x} = P_{\underline{y}(0)|x}$.

2. A sufficient condition for $ATT = ATU$ is that $P_{\underline{y}(1)|\underline{d}=1,x} - P_{\underline{y}(0)|\underline{d}=1,x}$ equals $P_{\underline{y}(1)|\underline{d}=0,x} - P_{\underline{y}(0)|\underline{d}=0,x}$.

3. A sufficient condition for $\mathcal{Y}_{1|0} = \mathcal{Y}_1$ is that $P_{\underline{y}(1)|\underline{d}=0} = P_{\underline{y}(1)}$. Note that the CIA implies that $P_{\underline{y}(1)|\underline{d}=0,x} = P_{\underline{y}(1)|x}$ always, but this does not imply that $P_{\underline{y}(1)|\underline{d}=0} = P_{\underline{y}(1)}$.

77.9 G_{do+} bnet

Fig.77.8 shows how bnet G_{do} is obtained by applying the do operator to bnet G, and how bnet G_{do+} is obtained by adding a prior probability distribution to one of the nodes of G_{do}. In bnet G_{do}, node $\underline{d}^\sigma$ has been stripped of all outside influences and fixed to a specific state $\widetilde{d}^\sigma$. This is what an RCT does.

The TPMs, printed in blue, for the bnets G_{do} and G_{do+}, are as follows. Note that the TPMs for bnets G_{do} and G_{do+} are defined in terms of the TPMs of bnet G.

$$P(x^\sigma) = P_{\underline{x}}(x^\sigma) \tag{77.46}$$

Figure 77.8: Bnet $G_{do} = \mathcal{D}_{\underline{d}^\sigma}(\widetilde{d}^\sigma)G$ is obtained by applying the do operator to node $\underline{d}^\sigma$ of bnet G. Bnet G_{do+} is obtained by adding a prior probability distribution $P(\widetilde{d}^\sigma)$ to node $\mathcal{D}\underline{d}^\sigma$ of bnet G_{do}.

$$Q(\widetilde{d}) = \sum_x P_{\underline{d}|\underline{x}}(\widetilde{d}|x)P_{\underline{x}}(x) \tag{77.47}$$

$$P(\widetilde{d}^\sigma) = \begin{cases} \delta(\underline{\widetilde{d}^\sigma}, \widetilde{d}^\sigma) & \text{for } G_{do} \\ Q(\widetilde{d}^\sigma) & \text{for } G_{do+} \end{cases} \tag{77.48}$$

$$P(y^\sigma|\widetilde{d}^\sigma, x^\sigma) = P_{\underline{y}|\underline{d},\underline{x}}(y^\sigma|\widetilde{d}^\sigma, x^\sigma) \tag{77.49}$$

Note that in G_{do},

$$P(\underline{y} = y|\mathcal{D}\underline{d} = d, \underline{x} = x) = P(y|\underline{d} = d, x) \tag{77.50}$$

because, by the d-separation theorem, when we condition on the confounder $\underline{x}$, we block information from being transmitted from $\underline{d}$ to $\underline{y}$ through $\underline{x}$, and this is equivalent to amputating the arrow $\underline{x} \to \underline{d}$.

Using Eq.(77.50), we get

$$P(\underline{y} = y|\mathcal{D}\underline{d} = d) = \sum_x P(\underline{y} = y|\mathcal{D}\underline{d} = d, x)P(x|\mathcal{D}\underline{d} = d) \tag{77.51}$$

$$= \sum_x P(y|d, x)P(x) \tag{77.52}$$

Eq.(77.52) is called the **backdoor adjustment formula**. It allows us to express a probability with a do operator in its definition in terms of probabilities without do operators.

77.10 $ACE = ATE$

Define the Average Causal Effect (ACE) by

$$ACE = \sum_y y[P(y|\mathcal{D}\underline{d} = 1) - P(y|\mathcal{D}\underline{d} = 0)] \tag{77.53}$$

$$= \sum_x P(x) \sum_y y[P(y|\underline{d} = 1, x) - P(y|\underline{d} = 0, x)] \text{ . (by Eq.(77.52)} \tag{77.54}$$

Claim 125 *If we assume both SUTVA and CIA (i.e., weak-d limit), then*

$$ACE = ATE \tag{77.55}$$

proof:

$$ACE = \sum_x P(x) \sum_y y[P(\underline{y} = y|\underline{d} = 1, x) - P(\underline{y} = y|\underline{d} = 0, x)] \tag{77.56}$$

$$= \sum_x P(x)[\mathcal{Y}_{1|1,x} - \mathcal{Y}_{0|0,x}] \text{ (by SUTVA)} \tag{77.57}$$

$$= \sum_x P(x)[\mathcal{Y}_{1|x} - \mathcal{Y}_{0|x}] \text{ (by CIA)} \tag{77.58}$$

$$= \mathcal{Y}_1 - \mathcal{Y}_0 \tag{77.59}$$

$$= ATE \tag{77.60}$$

QED

We will say there is a **null result in an RCT** when $ACE = 0$. By the previous claim, this is true iff $ATE = 0$ (assuming weak-d limit).

77.11 Good, Bad Controls

The bnet G_+ of Fig.77.3, —the cornerstone of Rubin's PO theory— is limited in scope and is easily misapplied, leading to incorrect results. The problem is some features that are available and could be conditioned on shouldn't because they would introduce spurious contributions to ATE. Such features are called "bad controls", as opposed to "good controls"[5] [6]

Examples:

[5] In this section, the word "controls" refers to the covariates (i.e., independent variables), other than $\underline{d}$, in a regression with $\underline{y}$ as target (i.e., independent) variable. This should not be confused with the control (i.e., untreated) individuals of a RCT.

[6] An alternative to the terms "good and bad controls" is "good and bad strata".

Figure 77.9: In this bnet, $\underline{x}$ is a bad control (i.e., should not be conditioned on). Nodes $\underline{h}_1$ and $\underline{h}_2$ are hidden and therefore cannot be conditioned on.

1. Consider the bnet Fig.77.9, which Pearl calls M-bias, because it looks like an M. In that figure, $\underline{x}$ is a "bad control" because calculating ATE by conditioning on it, and using the formula $ATE = \sum_x P(x)[\mathcal{Y}_{1|x} - \mathcal{Y}_{0|x}]$, yields a value of ATE that is different from ACE. This value for ATE is unacceptable because it does not give the result of an RCT whereas ACE defined in terms of do operators always does. The reason[7] $ATE \neq ACE$ for this figure is that in it, $\underline{x}$ is a collider node, and conditioning on it allows rather than prevents information to flow from $\underline{d}$ to $\underline{y}$ via the path $\underline{d} - \underline{h}_1 - \underline{x} - \underline{h}_2 - \underline{y}$.

Figure 77.10: In this bnet, node $\underline{c}_1$ is a good control and nodes $\underline{c}_2, \underline{c}_3, \underline{c}_4$ are bad ones.

2. In Fig.77.10, node $\underline{c}_1$ is a good control and nodes $\underline{c}_2, \underline{c}_3, \underline{c}_4$ are bad ones.

   Conditioning on $\underline{c}_1$ blocks path $\underline{d} - \underline{c}_1 - \underline{y}$, good

   Conditioning on $\underline{c}_2$ opens path $\underline{d} - \underline{c}_2 - \underline{y}$, bad

   Conditioning on $\underline{c}_3$ blocks path $\underline{d} - \underline{c}_3 - \underline{y}$, bad

   Conditioning on $\underline{c}_4$ opens path $\underline{d} - \underline{c}_4 - \underline{y}$, bad

---

[7] We are using here arguments based on the d-separation theorem which is discussed in Chapter 24.

3. (Taken from online course notes Ref.[28])

   Consider the bnet

$$\begin{tikzcd}
\underline{x}_2 \arrow[r] \arrow[d] & \underline{x}_3 & \underline{x}_4 \arrow[l] \arrow[d] & \\
\underline{d} \arrow[r] \arrow[d] & \underline{x}_6 \arrow[r] \arrow[d] & \underline{y} \arrow[d] & \underline{x}_7 \arrow[l] \\
\underline{x}_8 & \underline{x}_9 & \underline{x}_{10} &
\end{tikzcd} \tag{77.61}$$

   Find all **good strata (i.e., conditioning) sets** $\underline{x}.$ that block all non-direct paths between $\underline{d}$ and $\underline{y}$.

   Ans:

   - $\emptyset$
   - $\underline{x}_2$
   - $\underline{x}_4$
   - $\underline{x}_2, \underline{x}_4$
   - $\underline{x}_2, \underline{x}_3$
   - $\underline{x}_3, \underline{x}_4$
   - $\underline{x}_2, \underline{x}_3, \underline{x}_4$

   Add $\underline{x}_7$ to each of the previous 7 possible $\underline{x}.$. This gives a total of 14 possible conditioning sets. For each of these conditioning sets $\underline{x}.$, we can write

$$P(y|\mathcal{D}\underline{d} = d) = \sum_{x.} P(y|d, x.)P(x.) \tag{77.62}$$

Pearl et al have a paper (Ref.[13]) that I highly recommend that gives 20 examples of good, neutral and bad controls in an ATE calculation. Those 20 examples are also analyzed by my software SCuMpy (see Ref.[95]).

77.12 PO Confounder Sensitivity Analysis

There are various "sensitivity analysis" strategies that are commonly used as a sanity check for a PO analysis.

- **vary columns of dataset**
  1. **randomize $\underline{y}$ column** (random outcome) This should make $ATE = 0$.
  2. **randomize $\underline{x}$ column** (random common cause) This should make $ATE = 0$.
  3. **randomize $\underline{d}$ column** (placebo treatment dose) This should make $ATE = 0$.

4. **add new column** $\underline{u}$, where $\underline{u}$ is an "unobserved common cause". This should change ATE in a predictable manner. See below.
5. **add new column** $\underline{u}$, where $\underline{u}$ is a "randomized common cause". This should not change ATE.

- **vary rows of dataset**, either all of them or some of them. Replace them by a dataset that should obey same DAG. This should not change ATE.
- **vary good controls**. If using a DAG that is more complicated than the naive triangular PO DAG, vary from one set of good controls to another. This should not change ATE.

Figure 77.11: LDEN bnets used to do PO confounder sensitivity analysis. Node $\underline{c}$ is an unobserved common cause confounder. The operator $A_{\underline{c}}$ in bnet (b) annihilates $\underline{c}$ (i.e., $A_{\underline{c}}\underline{c} = 0$)

We end this section by deriving a formula for the change in ATE when an unobserved common cause $\underline{c}$ is added to the naive triangular PO DAG. (See Fig.77.11)

Consider the LDEN bnet of Fig.77.11 (a), whose structural equations, printed in blue, are as follows:

$$\underline{d} = \alpha\underline{x} + \alpha'\underline{c} + \epsilon_{\underline{d}} \tag{77.63}$$

$$\underline{y} = \beta\underline{x} + \beta'\underline{c} + \delta\underline{d} + \epsilon_{\underline{y}} \tag{77.64}$$

where $\epsilon_{\underline{d}}$ and $\epsilon_{\underline{y}}$ are root nodes with zero mean. Therefore,

$$\underline{c} = \frac{1}{\alpha'}(\underline{d} - \epsilon_{\underline{d}} - \alpha\underline{x}) \tag{77.65}$$

$$\underline{y} = \left(\beta - \frac{\alpha\beta'}{\alpha'}\right)\underline{x} + \left(\delta + \frac{\beta'}{\alpha'}\right)\underline{d} - \frac{\beta'}{\alpha'}\epsilon_{\underline{d}} + \epsilon_{\underline{y}} \tag{77.66}$$

$$\mathcal{Y}_{d|x,c} = E[\underline{y}(d)|x,c] = E[\underline{y}|d,x,c] = \beta x + \beta' c + \delta d \tag{77.67}$$

$$\mathcal{Y}_{d|x} = E[\underline{y}(d)|x] = E[\underline{y}|d,x] = \left(\beta - \frac{\alpha\beta'}{\alpha'}\right)x + \left(\delta + \frac{\beta'}{\alpha'}\right)d \tag{77.68}$$

$$ATE = E_x[ATE_x] = E_x[\mathcal{Y}_{1|x} - \mathcal{Y}_{0|x}] = \delta + \frac{\beta'}{\alpha'} \tag{77.69}$$

$$ATE|_{\beta'=0} = \delta \tag{77.70}$$

$$\boxed{ATE - ATE|_{\beta'=0} = \frac{\beta'}{\alpha'}} \tag{77.71}$$

Note that the right hand side of Eq.(77.71) is the product of the gains along the path $\underline{d} \to \underline{c} \to \underline{y}$. The gain $1/\alpha'$ for $\underline{d} \to \underline{c}$ equals the inverse of the gain α' for $\underline{c} \to \underline{d}$.

Neither α' nor β' are observed, but the right hand side of Eq.(77.71) can be bounded above by an observed quantity. This is done in Chapter 72.

77.13 Strata-Matching

For a situation described by the bnet G_+ in the weak-d limit, we can match *similar* individuals to fill the blank cells of Table 77.1. By "similar", we mean that they have the same or almost the same value of $\underline{x}^\sigma$.

The reason the weak-d limit is required is because it implies that $P(y(c)|d = 0, x) = P(y(c)|d = 1, x)$ for $c \in \{0, 1\}$, Hence, we can sample from a known factual ($c = d$) distribution to fill the missing data in the unknown counterfactual ($c \neq d$) distribution.

77.13.1 Exact strata-matching

Estimates of Treatment Effects

For $d \in \{0, 1\}$ and all strata x, define the sets of individuals $A_{d,x} = \{\sigma : d^\sigma = d, x^\sigma = x\}$, $A_x = A_{0,x} \cup A_{1,x}$ and $A = \cup_x A_x$. Let $N_{d,x} = |A_{d,x}|$, $N_x = |A_x|$ and $N = |A|$.

In exact strata-matching, we match each individual with $d^\sigma = d, x^\sigma = x$ with exactly one individual with $d^\sigma = 1 - d, x^\sigma = x$. Define a map $m : A \to A$ such that, for each x, and for $d \in \{0, 1\}$, if $\sigma \in A_{d,x}$, then $m(\sigma) \in A_{1-d,x}$ This assumes $A_{0,x}$ and $A_{1,x}$ are non-empty for all x. The purpose of map $m()$ is to fill in the missing data in the PO dataset. See Table 77.3 for a pictorial representation of this.

	$y^\sigma(0)$	$y^\sigma(1)$
$d^\sigma = 0$	y^σ	$y^{m(\sigma)}$
$d^\sigma = 1$	$y^{m(\sigma)}$	y^σ

Table 77.3: Illustration of the purpose of the map $m()$. Note that $y^\sigma = y^\sigma(d^\sigma)$ and $y^{m(\sigma)} = y^\sigma(1 - d^\sigma)$.

Note that

$$\sum_{\sigma \in A_x} \frac{d^\sigma}{N_{1,x}} = \sum_{\sigma \in A_{1,x}} \frac{1}{N_{1,x}} = 1 \ . \tag{77.72}$$

Thus

$$\sum_{\sigma \in A_x} \frac{d^\sigma}{N_{1,x}} y^\sigma = E_{\sigma|d=1,x}[y^\sigma] = \mathcal{Y}_{1|1,x} \tag{77.73}$$

Table 77.4 gives estimates of $\mathcal{Y}_{c|d,x}$

	$y^\sigma(0)$	$y^\sigma(1)$
$d^\sigma = 0$	$\frac{1}{N_{0,x}} \sum_{\sigma \in A_x} (1 - d^\sigma) y^\sigma = \mathcal{Y}_{0\|0,x}$	$\frac{1}{N_{0,x}} \sum_{\sigma \in A_x} (1 - d^\sigma) y^{m(\sigma)} = \mathcal{Y}_{1\|0,x}$
$d^\sigma = 1$	$\frac{1}{N_{1,x}} \sum_{\sigma \in A_x} d^\sigma y^{m(\sigma)} = \mathcal{Y}_{0\|1,x}$	$\frac{1}{N_{1,x}} \sum_{\sigma \in A_x} d^\sigma y^\sigma = \mathcal{Y}_{1\|1,x}$

Table 77.4: Estimates of $\mathcal{Y}_{c|d,x}$.

	$y^\sigma(0)$	$y^\sigma(1)$
$d^\sigma = 0$	$\frac{1}{N_x} \sum_{\sigma \in A_x} (1 - d^\sigma) y^\sigma = \mathcal{Y}_{0,0\|x}$	$\frac{1}{N_x} \sum_{\sigma \in A_x} (1 - d^\sigma) y^{m(\sigma)} = \mathcal{Y}_{1,0\|x}$
$d^\sigma = 1$	$\frac{1}{N_x} \sum_{\sigma \in A_x} d^\sigma y^{m(\sigma)} = \mathcal{Y}_{0,1\|x}$	$\frac{1}{N_x} \sum_{\sigma \in A_x} d^\sigma y^\sigma = \mathcal{Y}_{1,1\|x}$

Table 77.5: Estimates of $\mathcal{Y}_{c,d|x}$.

Recall that

$$\mathcal{Y}_{c,d|x} = \mathcal{Y}_{c|d,x} P(d|x) \tag{77.74}$$

Hence,

$$\mathcal{Y}_{c,d|x} = (N_{d,x} \mathcal{Y}_{c|d,x}) \frac{P(d|x)}{N_{d,x}} \tag{77.75}$$

$$= (N_{d,x} \mathcal{Y}_{c|d,x}) \frac{1}{N_x} \tag{77.76}$$

Table 77.5 gives estimates of $\mathcal{Y}_{c,d|x}$

The treatment effects $\mathcal{E} \in \{ATE, ATT, ATU, SB, SDO\}$ can be estimated from the data via the following estimates.

$$\begin{aligned}
\widehat{ATE}_x &= \overbrace{\mathcal{Y}_{1|1,x}P(1|x) + \mathcal{Y}_{1|0,x}P(0|x)}^{\mathcal{Y}_{1|x}} - \overbrace{(\mathcal{Y}_{0|1,x}P(1|x) + \mathcal{Y}_{0|0,x}P(0|x))}^{\mathcal{Y}_{0|x}} && (77.77)\\
&= \frac{1}{N_x}[\widehat{ATT}_x N_{1,x} + \widehat{ATU}_x N_{0,x}] && (77.78)\\
&= \frac{1}{N_x}\left[\sum_{\sigma\in A_x} d^\sigma[y^\sigma - y^{m(\sigma)}] + \sum_{\sigma\in A_x}(1-d^\sigma)[y^{m(\sigma)} - y^\sigma]\right] && (77.79)\\
&= \frac{1}{N_x}\sum_{\sigma\in A_x}(2d^\sigma - 1)[y^\sigma - y^{m(\sigma)}] && (77.80)
\end{aligned}$$

$$\begin{aligned}
\widehat{ATT}_x &= \overbrace{\frac{1}{N_x}\sum_{\sigma\in A_x} d^\sigma y^\sigma}^{\mathcal{Y}_{1,1|x}} - \overbrace{\frac{1}{N_x}\sum_{\sigma\in A_x} d^\sigma y^{m(\sigma)}}^{\mathcal{Y}_{0,1|x}} && (77.81)\\
&= \frac{1}{N_x}\sum_{\sigma\in A_x} d^\sigma[y^\sigma - y^{m(\sigma)}] && (77.82)
\end{aligned}$$

$$\begin{aligned}
\widehat{ATU}_x &= \overbrace{\frac{1}{N_x}\sum_{\sigma\in A_x}(1-d^\sigma)y^{m(\sigma)}}^{\mathcal{Y}_{1,0|x}} - \overbrace{\frac{1}{N_x}\sum_{\sigma\in A_x}(1-d^\sigma)y^\sigma}^{\mathcal{Y}_{0,0|x}} && (77.83)\\
&= \frac{1}{N_x}\sum_{\sigma\in A_x}(1-d^\sigma)[y^{m(\sigma)} - y^\sigma] && (77.84)
\end{aligned}$$

$$\widehat{SB}_x = \overbrace{\frac{1}{N_x}\sum_{\sigma\in A_x} d^\sigma y^{m(\sigma)}}^{\mathcal{Y}_{0,1|x}} - \overbrace{\frac{1}{N_x}\sum_{\sigma\in A_x}(1-d^\sigma)y^\sigma}^{\mathcal{Y}_{0,0|x}} \quad (77.85)$$

$$\widehat{SDO}_x = \overbrace{\frac{1}{N_x}\sum_{\sigma\in A_x} d^\sigma y^\sigma}^{\mathcal{Y}_{1,1|x}} - \overbrace{\frac{1}{N_x}\sum_{\sigma\in A_x}(1-d^\sigma)y^\sigma}^{\mathcal{Y}_{0,0|x}} \quad (77.86)$$

Suppose we do linear regression to fit a hyperplane $y(x)$ to the dataset set $\{(x^\sigma, y^\sigma) : \sigma\}$, and then we calculate $\frac{\partial y}{\partial \underline{d}} = \delta$. Out of all the treatment effects, this δ is probably (?) closest to $ACE = ATE$. Note also that the linear regression method of estimating δ does imputation (guesses missing values) by doing a linear fit. One can also use machine learning to do a non-linear fit. In contrast, the estimates of treatment effects presented in this section do imputation by non-linear strata-matching.

Example, estimation of treatment effects

For $\sigma \in \{1, 2, \ldots, 10\}$, define

$$m(\sigma) = \begin{cases} \sigma + 5 & \text{if } \sigma \leq 5 \\ \sigma - 5 & \text{if } \sigma > 5 \end{cases} \tag{77.87}$$

σ	d^σ	y^σ	$d^\sigma y^\sigma$	$(1-d^\sigma)y^\sigma$	$d^\sigma y^{m(\sigma)}$	$(1-d^\sigma)y^{m(\sigma)}$
1	0	0	0	0	0	0
2	0	0	0	0	0	1
3	0	1	0	1	0	1
4	0	1	0	1	0	1
5	0	1	0	1	0	1
6	1	0	0	0	0	0
7	1	1	1	0	0	0
8	1	1	1	0	1	0
9	1	1	1	0	1	0
10	1	1	1	0	1	0

Table 77.6: Estimates of treatment effects are calculated for this example.

$N(d, y)$	$y = 0$	$y = 1$
$d = 0$	2	3
$d = 1$	1	4

Table 77.7: $N(\underline{d^\sigma} = d, \underline{y^\sigma} = y)$ for the data in Table 77.6.

Let $N(\mathcal{S})$ be the number of individuals σ that satisfy condition $\mathcal{S}$. For example, $N(\underline{d^\sigma} = d)$ is the number of individuals such that $\underline{d^\sigma} = d$.

$$N_1 = N(d^\sigma = 1) = 5 \tag{77.88}$$

$$N_0 = N(d^\sigma = 0) = 5 \tag{77.89}$$

$$N = N_0 + N_1 = 10 \tag{77.90}$$

$$\mathcal{Y}_{1|1} = \frac{1}{N_1} \sum_{\sigma} d^{\sigma} y^{\sigma} = \frac{4}{5} \tag{77.91}$$

$$\mathcal{Y}_{0|0} = \frac{1}{N_0} \sum_{\sigma} (1 - d^{\sigma}) y^{\sigma} = \frac{3}{5} \tag{77.92}$$

$$\mathcal{Y}_{0|1} = \frac{1}{N_1} \sum_{\sigma} d^{\sigma} y^{m(\sigma)} = \frac{3}{5} \tag{77.93}$$

$$\mathcal{Y}_{1|0} = \frac{1}{N_0} \sum_{\sigma} (1 - d^{\sigma}) y^{m(\sigma)} = \frac{4}{5} \tag{77.94}$$

$$ATT = \mathcal{Y}_{1|1} - \mathcal{Y}_{0|1} = \frac{1}{5} \tag{77.95}$$

$$ATE = ATT = ATU = SDO = \frac{1}{5} \ , \quad SB = 0 \tag{77.96}$$

This example is unusual in that it has a single stratum x, and for that stratum, the treated and untreated populations are **balanced** (of equal size). Also, the map $m()$ is 1-1 onto. If, for instance, $m(\sigma) = 6$ for all $\sigma \in A_0$ and $m(\sigma) = 5$ for $\sigma \in A_1$, then ATE, ATT, ATU, SDO would not all be same, and SB would not be zero. In fact, whenever there is a single balanced stratum and the map $m()$ is 1-1 onto, Eq.(77.96) can be proven to be true using the methods of section 77.8.

77.13.2 Approximate strata-matching

It is very often the case that one can't find for a given individual σ another individual that has opposite d^{σ} but exactly the same value of x^{σ}. In such cases, one can discard all matchless individuals. But that would entail a loss of precious information. Instead of discarding orphans, a better way is to relax our demands and match individual σ with another individual $m(\sigma)$ such that x^{σ} and $x^{m(\sigma)}$ are very close in some metric. Alternatively, the matching individual might not be real; it might be a composite of individuals.

More precisely, for some arbitrary parameter $\epsilon > 0$, and an individual σ, define the **strata-matching set** $\mathcal{M}_{\epsilon}(\sigma)$ by[8]

$$\mathcal{M}_{\epsilon}(\sigma) = \{m : d^m = 1 - d^{\sigma}, dist(x^{\sigma}, x^m) \leq \epsilon\} \ , \tag{77.97}$$

where

$$dist(x^{\sigma}, x^m) = [x^{\sigma}]^T [\Sigma]^{-1} x^m \ , \tag{77.98}$$

where $\Sigma = \left\langle \underline{x}^{\sigma}, [\underline{x}^m]^T \right\rangle$. This metric $dist(x^{\sigma}, x^m)$ is called the **Mahalanobis distance**. We will call the case $\epsilon = 0$ an **exact strata-matching**, and the case $\epsilon \neq 0$

[8] One can use an ϵ that depends on σ. For example, let $\epsilon(\sigma, 5)$ satisfy $|\mathcal{M}_{\epsilon(\sigma,5)}(\sigma)| = 5$.

an **approximate strata-matching.**. To do an approximate strata-matching, replace $y^{m(\sigma)}$ by $\langle y \rangle^{\mathcal{M}_\epsilon(\sigma)}$ in the estimates given above for an exact strata-matching. $\langle y \rangle^{\mathcal{M}_\epsilon(\sigma)}$ is defined by

$$\langle y \rangle^{\mathcal{M}_\epsilon(\sigma)} = \frac{1}{|\mathcal{M}_\epsilon(\sigma)|} \sum_{m \in \mathcal{M}_\epsilon(\sigma)} y^m \ . \tag{77.99}$$

77.13.3 Unbiased strata-matching estimates

The estimates we obtained via strata-matching are biased because strata-matching, due to its non-uniqueness, introduces noise into the estimate. However, one can define new bias-corrected estimates. Following Ref.[15], we will next find an unbiased estimate of ATT_x using the biased estimate of ATT_x that we obtained by strata-matching.

Let $\widehat{\mathcal{Y}}_{|d,x}$ be an estimate of $\mathcal{Y}_{|d,x} = E_{|d,x}[\underline{y}]$ that is obtained, for example, via Linear Regression.

Claim 126 *The quantity*

$$\widehat{ATT}_x^{unbi} = \frac{1}{N_x} \sum_{\sigma \in A_x} d^\sigma \left[(y^\sigma - y^{m(\sigma)}) - (\widehat{\mathcal{Y}}_{|0,x^\sigma} - \widehat{\mathcal{Y}}_{|0,x^{m(\sigma)}}) \right] \tag{77.100}$$

is an unbiased estimate of ATT_x.

proof:

We begin by assuming a special case of SUTVA. Let

$$\underline{y}^\sigma = \underline{y}(d^\sigma) = \widehat{\mathcal{Y}}_{|d^\sigma,x^\sigma} + \underline{\epsilon}^\sigma \tag{77.101}$$

where

$$\left\langle \widehat{\mathcal{Y}}_{|d^\sigma,x^\sigma}, \underline{\epsilon}^\sigma \right\rangle_\sigma = 0 \tag{77.102}$$

Recall that the biased estimate of ATT_x obtained by strata-matching is

$$\widehat{ATT}_x^{bi} = \frac{1}{N_x} \sum_{\sigma \in A_x} d^\sigma (y^\sigma - y^{m(\sigma)}) \tag{77.103}$$

where σ and $m(\sigma)$ are matched (i.e., $x^\sigma \approx x^{m(\sigma)}$ and $d^{m(\sigma)} = 1 - d^\sigma$).

$$\widehat{ATT}_x^{bi} = \frac{1}{N_x} \sum_{\sigma \in A_x} d^\sigma \left(\widehat{\mathcal{Y}}_{|1,x^\sigma} - \widehat{\mathcal{Y}}_{|0,x^{m(\sigma)}} \right) \tag{77.104}$$

$$+ \frac{1}{N_x} \sum_{\sigma \in A_x} d^\sigma (\epsilon^\sigma - \epsilon^{m(\sigma)}) \tag{77.105}$$

$$\widehat{ATT}_x^{bi} = \underbrace{\frac{1}{N_x}\sum_{\sigma\in A_x} d^\sigma\left(\widehat{\mathcal{Y}}_{|1,x^\sigma} - \widehat{\mathcal{Y}}_{|0,x^\sigma}\right)}_{ATT_x^{unbi}} \tag{77.106}$$

$$+ \underbrace{\frac{1}{N_x}\sum_{\sigma\in A_x} d^\sigma\left(\widehat{\mathcal{Y}}_{|0,x^\sigma} - \widehat{\mathcal{Y}}_{|0,x^{m(\sigma)}}\right)}_{\Delta ATT_x} \tag{77.107}$$

$$+ \underbrace{\frac{1}{N_x}\sum_{\sigma\in A_x} d^\sigma(\epsilon^\sigma - \epsilon^{m(\sigma)})}_{\mathcal{E}_x} \tag{77.108}$$

$$ATT_x^{unbi} = \underbrace{\widehat{ATT}_x^{bi} - \Delta ATT_x}_{\widehat{ATT}_x^{unbi}} - \mathcal{E}_x \tag{77.109}$$

By the Central Limit Theorem, for large N_x, this sum over $\sigma \in A_x$ of i.i.d. summands is normally distributed

$$\sqrt{N_x} ATT_x^{unbi} \sim \mathcal{N}(x; 0, var) \tag{77.110}$$

The reason for the $\sqrt{N_x}$ normalization is that we want the variance to be proportional to N_x.

$$var = N_x \left\langle ATT_x^{unbi}, ATT_x^{unbi} \right\rangle \tag{77.111}$$

$$= N_x \left\langle \widehat{ATT}_x^{unbi}, \widehat{ATT}_x^{unbi} \right\rangle + N_x \left\langle \mathcal{E}_x, \mathcal{E}_x \right\rangle \tag{77.112}$$

QED

77.14 (SDO, ATE) space

If we substitute $y^\sigma \to y^\sigma(d^\sigma)$ and $y^{m(\sigma)} \to y^\sigma(1-d^\sigma)$ into the estimate Eq.(77.80) for ATE and the estimate Eq.(77.86) for SDO, we get

$$\widehat{ATE}_x = \frac{1}{N_x}\sum_{\sigma\in A_x} (2d^\sigma - 1)[y^\sigma(d^\sigma) - y^\sigma(1-d^\sigma)] \tag{77.113}$$

$$= \frac{1}{N_x}\sum_{\sigma\in A_x} [y^\sigma(1) - y^\sigma(0)] \tag{77.114}$$

and

$$\widehat{SDO}_x = \frac{1}{N_{1,x}} \sum_{\sigma \in A_x} d^\sigma y^\sigma(d^\sigma) - \frac{1}{N_{0,x}} \sum_{\sigma \in A_x} (1 - d^\sigma) y^\sigma(d^\sigma) \tag{77.115}$$

$$= \frac{1}{N_{1,x}} \sum_{\sigma \in A_{1,x}} y^\sigma(1) - \frac{1}{N_{0,x}} \sum_{\sigma \in A_{0,x}} y^\sigma(0) \ . \tag{77.116}$$

Recall that $\widehat{\mathcal{E}} = E_x[\widehat{\mathcal{E}}_x] = \sum_x \frac{N_x}{N} \widehat{\mathcal{E}}_x$ for $\mathcal{E} \in \{ATE, SDO\}$.

Recall also that $ACE = ATE = 0$ is the null result in an RCT.

Suppose that the treatment outcome y^σ has only two possible values, 0 and 1. Then, $-1 \leq ATE \leq 1$ and $-1 \leq SDO \leq 1$. But does $ATE = 0$ imply $SDO = 0$ or vice versa? Next, we answer that question and more by finding the region of accessibility in the (SDO, ATE) plane, assuming $y^\sigma \in \{0, 1\}$.

σ	d^σ	$y^\sigma(0)$	$y^\sigma(1)$
1	0	1	0
2	0	1	0
3	0	1	0
4	1	1	0
5	1	1	0
6	1	1	0

(a) $ATE = -1$ ($SDO = -1$) point A

σ	d^σ	$y^\sigma(0)$	$y^\sigma(1)$
1	0	0	1
2	0	0	1
3	0	0	1
4	1	0	0
5	1	0	0
6	1	0	0

(b) $ATE = \frac{1}{2}$ ($SDO = 0$) point B

σ	d^σ	$y^\sigma(0)$	$y^\sigma(1)$
1	0	0	1
2	0	0	1
3	0	0	1
4	1	0	1
5	1	0	1
6	1	0	1

(c) $ATE = 1$ ($SDO = 1$) point C

Figure 77.12: Examples of PO datasets. Exploring ATE extremes.

σ	d^σ	$y^\sigma(0)$	$y^\sigma(1)$
1	0	1	1
2	0	1	1
3	0	1	1
4	1	0	0
5	1	0	0
6	1	0	0

(a) $SDO = -1$ ($ATE = 0$) point D

σ	d^σ	$y^\sigma(0)$	$y^\sigma(1)$
1	0	1	0
2	0	1	0
3	0	1	0
4	1	1	1
5	1	1	1
6	1	1	1

(b) $SDO = 0$ ($ATE = -\frac{1}{2}$) point E

σ	d^σ	$y^\sigma(0)$	$y^\sigma(1)$
1	0	0	0
2	0	0	0
3	0	0	0
4	1	1	1
5	1	1	1
6	1	1	1

(c) $SDO = 1$ ($ATE = 0$) point F

Figure 77.13: Examples of PO datasets. Exploring SDO extremes.

Figure 77.14: Green parallelogram is accessible region in (SDO, ATE) plane, assuming $y^\sigma \in \{0, 1\}$. Each of the six points A, B, ... F corresponds to one of the six tables in Figs. 77.12 and 77.13. Segment DF corresponds to the null result in an RCT.

77.15 Propensities

It is often the case that the discrete vector $\underline{x}^\sigma$ has too many possible values to make matching possible. In such cases, it is convenient to map the space of vectors $\underline{x}^\sigma$ to the real line. One very convenient choice for that map is the **propensity score**, which is defined as

$$g(x^\sigma) = P(\underline{d}^\sigma = 1|x^\sigma) \ . \tag{77.117}$$

$P(\underline{d}^\sigma = 1|x^\sigma)$ is easy to calculate from the dataset, using $g(x) = N_{1,x}/N_x$.

Figure 77.15: Bnets G_+, G_{g1}, G_{g2} used when doing propensity scoring.

To use the propensity score, one replaces the bnet G_+ by the bnet G_{g1} as shown in Fig.77.15. The TPMs, printed in blue, for the 2 nodes of G_{g1} that differ from the nodes of G_+, are as follows:[9]

$$P(g^\sigma|x^\sigma) = \delta(g^\sigma - g(x^\sigma)) \tag{77.118}$$

$$P(d^\sigma|g^\sigma) = g^\sigma d^\sigma + (1 - g^\sigma)(1 - d^\sigma) \tag{77.119}$$

Note that these TPMs are self-consistent because

$$\begin{aligned} P(d|x) &= \sum_g P(d|g)P(g|x) && (77.120)\\ &= g(x)d + [1 - g(x)](1 - d) && (77.121)\\ &= \begin{cases} g(x) & \text{if } d = 1 \\ 1 - g(x) & \text{if } d = 0 \end{cases} && (77.122)\\ &= P(d|x) && (77.123) \end{aligned}$$

[9] g is a continuous variable in the interval $[0, 1]$ so its distribution $P(g|x)$ is a Dirac delta function.

We would like to do **propensity score strata-matching** by matching g-strata instead of x-strata. To do g-strata-matching, we use the graph G_{g2} in Fig.77.15 which differs from graph G_+ in that node $\underline{x}^\sigma$ is replaced by node $\underline{g}^\sigma$. Next we get the TPM of graph G_{g2} in terms of the TPM of graph G_+. We do this by expressing $P(d|g)$, $P(g)$ and $P(y|d,g)$ in terms of $P(d|x)$, $P(x)$ and $P(y|d,x)$.

From the TPMs for G_{g1}, one has

$$\boxed{P(d|g) = gd + (1-g)(1-d)} \tag{77.124}$$

and

$$\boxed{P(g) = \sum_x \overbrace{\delta(g - g(x))}^{P(g|x)} P(x)} \, . \tag{77.125}$$

Note that $P(g)$ is a linear combination of Dirac delta functions. Next, note that

$$P(y|d,g) = \sum_x P(y|d,x)P(x|g) \tag{77.126}$$

so we need to find $P(x|g)$. Since

$$P(x|g) = \frac{P(g|x)P(x)}{P(g)} \tag{77.127}$$

$$= \frac{\delta(g - g(x))P(x)}{P(g)} \tag{77.128}$$

we finally get

$$\boxed{P(y|d,g) = \sum_x P(y|d,x)\frac{\delta(g - g(x))P(x)}{P(g)}} \, . \tag{77.129}$$

Eq.(77.129) looks complicated, but all it is saying is that

$$P(y|d,g)P(g) = N_g \sum_{x \in g^{-1}(g)} P(y|d,x)P(x) \, , \tag{77.130}$$

where $g^{-1}(g) = \{x : g(x) = g\}$ and N_g is the number of g when we discretize the interval $[0,1]$. In general,

$$\sum_x \delta(g - g(x)) = N_g \sum_{x \in g^{-1}(g)} \tag{77.131}$$

Recall that for any treatment effect $\mathcal{E} \in \{ATE, ATT, ATU, SB, SDO\}$, we can estimate $\mathcal{E}_x$, and then calculate $\mathcal{E}$ from it, using

$$\mathcal{E} = \sum_x P(x)\mathcal{E}_x \tag{77.132}$$

In terms of g, this becomes

$$\mathcal{E} = \int_0^1 dg\ P(g)\mathcal{E}_{x\to g} \tag{77.133}$$

If we smooth $P(g)$, then the integral over g can be done by complex contour integration and reduces to a finite sum over a few residues. The large sum over x has been swept under the rug, into the calculation of $P(g)$.

77.15.1 Propensity based estimates of Treatment Effects

In the strata-matching section 77.13, we gave an estimate of ACE_x. In strata-matching, one fills in the missing counterfactual values via the map $m()$. This is justified because, by CIA (weak-d limit), the counterfactual distributions are assumed to equal the factual distributions (see Fig. 77.7). In this section, we give estimates of treatment effects that are based on the formulae derived in Section 77.15. The estimates given in this section do no require the map $m()$, because the identification of counterfactual distributions with factual ones was used to derive the formulae of Section 77.15.

Let

$$g_{d|x} = g_d(x) = P(d|x) \tag{77.134}$$

for $d \in \{0,1\}$. Note that $g_{0|x} + g_{1|x} = 1$. We will refer to $g_d(x)$ for $d \in \{0,1\}$ as **dual propensities** and to $g_1(x)$ as the **propensity score**.

Define

$$\delta_{y|x} = P(y|\underline{d} = 1, x) - P(y|\underline{d} = 0, x) \tag{77.135}$$

Note that

$$\begin{aligned}
\delta_{y|x} &= \frac{P(y, \underline{d} = 1|x)}{g_{1|x}} - \frac{P(y, \underline{d} = 0|x)}{g_{0|x}} && (77.136)\\
&= \frac{P(y, \underline{d} = 1|x)g_{0|x} - P(y, \underline{d} = 0|x)g_{1|x}}{g_{0|x}g_{1|x}} && (77.137)\\
&= \frac{P(y, \underline{d} = 1|x) - P(y|x)g(x)}{g(x)(1-g(x))} && (77.138)
\end{aligned}$$

Dividing each term in a sum over $d \in \{0,1\}$ by $g_d(x)$ is often called **inverse probability (or propensity) weighting** (IPW). $g_d(x) = P(\underline{d} = d|x)$ is the likelihood of strata x, so dividing each term by this likelihood increases the contribution to the sum by less likely strata and decreases the contribution by more likely strata.

Note that the backdoor adjustment formula can expressed as an IPW:

$$\begin{aligned}
P(y|\mathcal{D}\underline{d} = d) &= \sum_x P(y|d,x)P(x) && (77.139)\\
&= \sum_x \frac{P(y,d,x)}{P(d|x)} && (77.140)
\end{aligned}$$

Note that $ATE = ACE$ can be expressed in terms of propensities:

$$
\begin{aligned}
ACE &= \sum_y y[P(y|\mathcal{D}\underline{d}=1) - P(y|\mathcal{D}\underline{d}=0)] && (77.141)\\
&= \sum_x P(x) \sum_y y\,[P(y|d=1,x) - P(y|d=0,x)] && (77.142)\\
&= \sum_x P(x) \underbrace{\sum_y y\delta_{y|x}}_{ACE_x} && (77.143)
\end{aligned}
$$

Note that

$$
\begin{aligned}
P(y(c)|d) &= \sum_x P(y(c)|d,x)P(x|d) && (77.144)\\
&= \sum_x P(y(c)|c,x)P(x|d) \text{ (by CIA)} && (77.145)\\
&= \sum_x P(y|c,x)P(x|d) \text{ (by SUTVA)} && (77.146)
\end{aligned}
$$

It's instructive to express all the other treatment effects besides ATE in terms of propensities:

$$
\begin{aligned}
ATT &= \sum_y y[P(\underline{y}(1)=y|d=1) - P(\underline{y}(0)=y|d=1)] && (77.147)\\
&= \sum_x P(x|d=1) \sum_y y\,[P(y|\underline{d}=1,x) - P(y|\underline{d}=0,x)] \text{ (by Eq.(77.146))} && (77.148)\\
&= \sum_x P(x|d=1) \sum_y y\ \delta_{y|x} && (77.149)\\
&= \sum_x P(x) \underbrace{\frac{1}{P(\underline{d}=1)} \sum_y y\ g_{1|x}\delta_{y|x}}_{ATT_x} && (77.150)
\end{aligned}
$$

$$
ATU = \sum_x P(x) \underbrace{\frac{1}{P(\underline{d}=0)} \sum_y y\ g_{0|x}\delta_{y|x}}_{ATU_x} \qquad (77.151)
$$

$$
\begin{aligned}
SB &= \sum_y y[P(\underline{y}(0)=y|d=1) - P(\underline{y}(0)=y|d=0)] && (77.152)\\
&= \sum_y y \sum_x P(y|\underline{d}=0,x)[P(x|d=1) - P(x|d=0)] \text{ (by Eq.(77.146))} && (77.153)\\
&= \sum_x P(x) \underbrace{\sum_y y \frac{P(y,\underline{d}=0|x)}{g_{0|x}} \left[\frac{g_{1|x}}{P(\underline{d}=1)} - \frac{g_{0|x}}{P(\underline{d}=0)}\right]}_{SB_x} && (77.154)
\end{aligned}
$$

$$SDO = \sum_y y[P(\underline{y}(1) = y|d = 1) - P(\underline{y}(0) = y|d = 0)] \tag{77.155}$$

$$= \sum_x P(x) \underbrace{\sum_y y \left[\frac{P(y, \underline{d} = 1|x)}{P(\underline{d} = 1)} - \frac{P(y, \underline{d} = 0|x)}{P(\underline{d} = 0)} \right]}_{SDO_x} \text{ (by SUTVA)} \tag{77.156}$$

77.15.2 Doubly Robust Estimates of Treatment Effects

Eq.(77.143) suggests the estimate

$$\widehat{ACE}_x = \frac{1}{N_x} \sum_{\sigma \in A_x} y^\sigma \left[\frac{d^\sigma - g(x)}{g(x)(1 - g(x))} \right] \tag{77.157}$$

$$= \frac{1}{N_x} \sum_{\sigma \in A_x} \left[\frac{d^\sigma y^\sigma}{g_{1|x}} - \frac{(1 - d^\sigma) y^\sigma}{g_{0|x}} \right] \tag{77.158}$$

This estimate is unbiased (because it doesn't have a source of noise like the strata-matching estimates do), but it is still possible to improve it. We next define another ACE estimate called **Doubly Robust Estimate** (DRE) that is also unbiased and has smaller variance.

Let $\widehat{\mathcal{Y}}_{|d,x}$ be an estimate of $\mathcal{Y}_{|d,x} = E_{\underline{y}|d,x}[\underline{y}]$ that is obtained, for example, via Linear Regression. Define the DRE

$$\widehat{ACE}_x^{DRE} = \widehat{\mathcal{Y}}_{1|x}^{DRE} - \widehat{\mathcal{Y}}_{0|x}^{DRE} \tag{77.159}$$

where

$$\widehat{\mathcal{Y}}_{1|x}^{DRE} = \frac{1}{N_x} \sum_{\sigma \in A_x} \left[\frac{d^\sigma (y^\sigma - \widehat{\mathcal{Y}}_{|1,x})}{g_{1|x}} + \widehat{\mathcal{Y}}_{|1,x} \right] \tag{77.160}$$

and

$$\widehat{\mathcal{Y}}_{0|x}^{DRE} = \frac{1}{N_x} \sum_{\sigma \in A_x} \left[\frac{(1 - d^\sigma)(y^\sigma - \widehat{\mathcal{Y}}_{|0,x})}{g_{0|x}} + \widehat{\mathcal{Y}}_{|0,x} \right] . \tag{77.161}$$

The DRE $\widehat{\mathcal{Y}}_{1|x}^{DRE}$ requires first estimating 2 preparatory quantities, $\widehat{\mathcal{Y}}_{|1,x}$ and $g_{1|x}$. It's called doubly robust because it remains unbiased even if one of the estimates of those 2 preparatory quantities is wrong, but not if both are wrong. Let's check this.

- Suppose the propensity $g_{1|x}$ is slightly wrong. So what because

$$E_{\sigma|1,x} \left[\frac{d^\sigma (y^\sigma - \widehat{\mathcal{Y}}_{|1,x})}{g_{1|x}} \right] = 0 \tag{77.162}$$

- Suppose $\widehat{\mathcal{Y}}_{|1,x}$ is slightly wrong. So what because

$$E_{\sigma|1,x}\left[\frac{-d^{\sigma}\widehat{\mathcal{Y}}_{|1,x}+g_{1|x}\widehat{\mathcal{Y}}_{|1,x}}{g_{1|x}}\right]=0 \tag{77.163}$$

A similar argument can be used to show that $\widehat{\mathcal{Y}}^{DRE}_{0|x}$ is doubly robust too.

77.15.3 Positivity

Figure 77.16: Pictorial representation of positivity. $g_{0|x}+g_{1|x}=1$. $g_{0|x}=0$ and $g_{1|x}=0$ are forbidden.

Positivity or **non-zero overlap** is defined as the requirement that for all layers x,

$$0<\underbrace{P(\underline{d}^{\sigma}=1|\underline{x}^{\sigma}=x)}_{g_{1|x}}<1 \tag{77.164}$$

or, equivalently,

$$\underbrace{P(\underline{d}^{\sigma}=1|\underline{x}^{\sigma}=x)}_{g_{1|x}}>0 \quad \text{and} \quad \underbrace{P(\underline{d}^{\sigma}=0|\underline{x}^{\sigma}=x)}_{g_{0|x}}>0\ . \tag{77.165}$$

In other words, for each layer x, there is a non-zero probability of being both treated and untreated. See Fig.77.16 for a pictorial representation of positivity.

If positivity is violated for some layer x, then

- the propensity based estimate Eq.(77.158) for ACE_x (which equals ATE_x) is undefined.
- all strata-matching estimates of $\mathcal{E}_x$ that use the matching function $m()$ are undefined because that function is undefined if $A_{0,x}=\emptyset$ or $A_{1,x}=\emptyset$.

If a quantity (estimand) can be estimated, it is said to be **do-identifiable** (i.e., expressible without do() operators). If positivity is violated, then $ACE=ATE$ is not identifiable.

When $P(d|x)$ becomes 0 or 1 for some x, the arrow $\underline{x} \to \underline{d}$ becomes deterministic for that x. This situation is the very antithesis of RCTs, wherein the influence exerted by $\underline{x}^\sigma$ on $\underline{d}^\sigma$ is uniformly random and therefore ignorable. Hence, it is perhaps not too surprising that a violation of positivity makes $ACE = ATE$ not identifiable.

77.16 Multi-time PO bnets (Panel Data)

In this section, we will discuss Multi-time PO bnets (MT-PO).

A **time-series** is a function $f : D \to \mathbb{R}$ whose domain D is a discrete set of times. A time-series usually describes a single unit σ (i.e., an individual) in a population.

An **observational study (or analysis or model)** can be cross-sectional or longitudinal. A **cross-sectional study** collects and analyzes a **cross-sectional dataset**; i.e., a dataset for a population at a single time. A **longitudinal study or panel study** collects and analyzes a **longitudinal dataset**; i.e., a dataset for a population at multiple times. Thus, a longitudinal study consists of one or more time-series.

Let $\mathcal{T} = \{t_0, t_1, \ldots, t_{nt-1}\}$. For any time-series $a_t : \mathcal{T} \to \mathbb{R}$, define

$$E_t a_t = \frac{1}{nt} \sum_{t \in \mathcal{T}} a_t \tag{77.166}$$

$$\Delta_t a_t = a_t - E_t a_t \tag{77.167}$$

$$\langle a_t, b_t \rangle_t = E_t \Delta_t a_t \Delta_t b_t \tag{77.168}$$

Consider a quantity a_t^σ that is a function of the time t and of the particular unit σ in a population. a_t^σ is said to be a **t-constant effect** if it is t-independent. a_t^σ is said to be a **homogeneous effect** (antonym: **heterogeneous effect**) if it is σ-independent. Henceforth, we will avoid using the word "effect" for these because that word has already been used for "treatment effect" in PO theory. Instead, we will use the word "quantity".

Fig.77.17 gives an example of a multi-time PO bnet (MT-PO). Note that in this example, $\underline{x}^\sigma$ and $\underline{u}^\sigma$ are t-constant quantities. $\underline{u}^\sigma$ is an unobserved confounder and $\underline{x}^\sigma$ is an observed confounder. For convenience and simplicity, we will assume linear deterministic TPMs. The TPMs, printed in blue, for the bnet Fig.77.17, are as follows:

$$P(x^\sigma) = P_{\underline{x}}(x^\sigma) \tag{77.169}$$

$$P(u^\sigma) = P_{\underline{u}}(u^\sigma) \tag{77.170}$$

Figure 77.17: Example of multi-time PO bnet with t-constant quantities $\underline{x}^\sigma, \underline{u}^\sigma$. The 3 nodes $\underline{x}^\sigma$ should be identified as a single node. Likewise, the 3 nodes $\underline{u}^\sigma$ should be identified as a single node.

$$P(y_t^\sigma | d_t^\sigma, x^\sigma, u^\sigma) = \mathbb{1}(\ \ y_t^\sigma = \delta d_t^\sigma + \beta x^\sigma + u^\sigma\ \ ) \tag{77.171}$$

$$P(d_{t+1}^\sigma | d_t^\sigma, x^\sigma, u^\sigma) = \mathbb{1}(\ \ d_{t+1}^\sigma = \alpha d_t^\sigma + \gamma x^\sigma + u^\sigma\ \ ) \tag{77.172}$$

Taking time averages of the treatment dose and treatment outcome, we get

$$E_t \underline{y}_t^\sigma = \delta E_t \underline{d}_t^\sigma + \beta \underline{x}^\sigma + \underline{u}^\sigma\ , \tag{77.173}$$

$$E_t \underline{d}_{t+1}^\sigma = \alpha E_t \underline{d}_t^\sigma + \gamma \underline{x}^\sigma + \underline{u}^\sigma\ . \tag{77.174}$$

Subtracting the time averages from the quantities being averaged, we get

$$\Delta_t \underline{y}_t^\sigma = \delta \Delta_t \underline{d}_t^\sigma\ , \tag{77.175}$$

$$\Delta_t \underline{d}_{t+1}^\sigma = \alpha \Delta_t \underline{d}_t^\sigma\ . \tag{77.176}$$

This allows us to find estimates for δ and α:

$$E_\sigma \left\langle \underline{y}_t^\sigma, \underline{y}_t^\sigma \right\rangle_t = \delta E_\sigma \left\langle \underline{y}_t^\sigma, \underline{d}_t^\sigma \right\rangle_t \tag{77.177}$$

$$\delta = \frac{E_\sigma \left\langle \underline{y}_t^\sigma, \underline{y}_t^\sigma \right\rangle_t}{E_\sigma \left\langle \underline{y}_t^\sigma, \underline{d}_t^\sigma \right\rangle_t} \tag{77.178}$$

$$E_\sigma \left\langle \underline{d}^\sigma_{t+1}, \underline{d}^\sigma_{t+1} \right\rangle_t = \alpha E_\sigma \left\langle \underline{d}^\sigma_{t+1}, \underline{d}^\sigma_t \right\rangle_t \tag{77.179}$$

$$\alpha = \frac{E_\sigma \left\langle \underline{d}^\sigma_{t+1}, \underline{d}^\sigma_{t+1} \right\rangle_t}{E_\sigma \left\langle \underline{d}^\sigma_{t+1}, \underline{d}^\sigma_t \right\rangle_t} \tag{77.180}$$

As shown in Fig.77.18, subtraction of time averages from each node removes the confounder nodes from the bnet of Fig.77.17 (However, this assumes that the confounders are t-constant and that the TPMs are linear deterministic, two very strong assumptions).

$$\begin{tikzcd}
\Delta_t \underline{d}^\sigma_t \arrow[r, "\alpha"'] \arrow[d, "\delta"'] & \Delta_t \underline{d}^\sigma_{t+1} \arrow[d] \\
\Delta_t \underline{y}^\sigma_t & \Delta_t \underline{y}^\sigma_{t+1}
\end{tikzcd}$$

Figure 77.18: time-average-subtracted (TAS) bnet for the bnet of Fig.77.17.

Chapter 78

Principal Component Analysis

This chapter is based on Ref.[174].

Principal Component Analysis (PCA) is the most common way of doing **Dimensionality Reduction** (See Chapter 21)

- **Notation**

  $X = [x_1, x_2, \ldots, x_{\mathcal{F}}] \in \mathbb{R}^{N\times\mathcal{F}}$ is the **initial dataset** to be dimensionality reduced (i.e., the number of its feature columns is to be reduced). Henceforth, we will assume that X has *zero mean*. If X doesn't have zero mean initially, replace $X_{\sigma,i}$ by $X_{\sigma,i} - \frac{1}{N}\sum_\sigma X_{\sigma,i}$

  $Y = [y_1, y_2, \ldots, y_{\mathcal{F}}] \in \mathbb{R}^{N\times\mathcal{F}}$. The y_i for $i = 1, 2, \ldots, \mathcal{F}$ are called the **principal components** (PCs).

  $U = [u_1, u_2, \ldots u_N] \in \mathbb{R}^{N\times N}$, $UU^T = U^TU = 1$ (U orthogonal). Components of u_σ will be denoted by $(u_\sigma)_{\sigma'}$ for $\sigma' = 1, 2, \ldots, N$.

  $\Lambda = \begin{bmatrix} \Lambda_D \\ 0^{(N-\mathcal{F})\times\mathcal{F}} \end{bmatrix} \in \mathbb{R}^{N\times\mathcal{F}}$, $\Lambda_D = diag(\lambda_1, \lambda_2, \ldots, \lambda_{\mathcal{F}})$

  $W = [w_1, w_2, \ldots w_{\mathcal{F}}] \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$, $WW^T = W^TW = 1$ (W orthogonal) Components of w_i will be denoted by $(w_i)_f$ for $f = 1, 2, \ldots, \mathcal{F}$.

  Singular Value Decomposition (SVD)

$$X = U\Lambda W^T \tag{78.1}$$

$$= \sum_{f=1}^{\mathcal{F}} \lambda_f u_f w_f^T \tag{78.2}$$

- **Covariance Matrix**

$$\begin{aligned} X^T X &= W\Lambda^T U^T U \Lambda W^T && (78.3)\\ &= W\Lambda^T \Lambda W^T && (78.4)\\ &= W\Lambda_D^2 W^T && (78.5)\\ &= \sum_{f=1}^{\mathcal{F}} \lambda_f^2 w_f w_f^T && (78.6) \end{aligned}$$

Reorder diagonal elements of Λ_D and columns of W so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{\mathcal{F}}$. We will refer to λ_i^2 as the ith **principal value** (PV) of X and to $w_i = [W_{r,i}]_{r=1}^{\mathcal{F}} \in \mathbb{R}^{\mathcal{F}\times 1}$ as the ith **principal axis** (PA) of X.

Note that

$$\begin{aligned} Y &= XW && (78.7)\\ &= U\Lambda W^T W && (78.8)\\ &= U\Lambda && (78.9) \end{aligned}$$

$Y = U\Lambda$ is called the **Polar Decomposition** of Y.

Note that

$$Y_{\sigma,i} = \sum_j X_{\sigma,j} W_{j,i} = \sum_j U_{\sigma,j} \lambda_j \delta_{j,i} \tag{78.10}$$

so

$$y_i = X w_i = u_i \lambda_i \tag{78.11}$$

Note that

$$\begin{aligned} [Y^T Y]_{i,j} &= Y_{\sigma,i} Y_{\sigma,j} && (78.12)\\ &= y_i^T y_j && (78.13)\\ &= \lambda_i \lambda_j u_i^T u_j && (78.14)\\ &= \lambda_i^2 \delta_{i,j} && (78.15) \end{aligned}$$

- **Truncation**

Define $X_{\leq\Phi}$ by

$$X_{\leq\Phi} = U\Lambda_{\leq\Phi} W^T \tag{78.16}$$

where $\Lambda_{\leq\Phi} = diag(\lambda_1, \lambda_2, \ldots, \lambda_\Phi, 0^{\mathcal{F}-\Phi})$.

Define $Y_{\leq\Phi}$ by

$$Y_{\sigma,i\leq\Phi} = \sum_j X_{\sigma,j} W_{j,i\leq\Phi} = U_{\sigma,i}\lambda_{i\leq\Phi} \tag{78.17}$$

$$\begin{aligned} Y_{\leq\Phi} &= XW_{\leq\Phi} = U\Lambda_{\leq\Phi} && (78.18)\\ &= X_{\leq\Phi}W \ \ \text{(from Eq.(78.16))} && (78.19)\end{aligned}$$

Then

$$X - X_{\leq\Phi} = U(\Lambda - \Lambda_{\leq\Phi})W^T \tag{78.20}$$

$$Y - Y_{\leq\Phi} = U(\Lambda - \Lambda_{\leq\Phi}) \tag{78.21}$$

$$\begin{aligned} Error &= \text{tr}\left[(X - X_{\leq\Phi})^T(X - X_{\leq\Phi})\right] && (78.22)\\ &= \text{tr}\left[(Y - Y_{\leq\Phi})^T(Y - Y_{\leq\Phi})\right] && (78.23)\\ &= \text{tr}\left[(\Lambda - \Lambda_{\leq\Phi})^2\right] && (78.24)\\ &= \sum_{i=\Phi+1}^{\mathcal{F}} \lambda_i^2 && (78.25)\end{aligned}$$

- **PCA as solution to maximization problem**

  Define the first PA w_1 by

$$\begin{aligned} w_1 &= \underset{w_1:\, w_1^T w_1 = 1}{\text{argmax}}\ y_1^T y_1 && (78.26)\\ &= \underset{w_1:\, w_1^T w_1 = 1}{\text{argmax}}\ w_1^T X^T X w_1 && (78.27)\\ &= \underset{w_1}{\text{argmax}} \sum_\sigma \frac{w_1^T X^T X w_1}{w_1^T w_1} && (78.28)\end{aligned}$$

  Consider the following diagonalization of $I_{\mathcal{F}}$ (This is one of many possible diagonalizations of $I_{\mathcal{F}}$) [1]

$$\sum_k w_k w_k^T = 1 \tag{78.29}$$

[1] In Quantum Mechanics,

$$\sum_i |\phi_i\rangle\langle\phi_i| = 1$$

is called a **completeness condition** for an orthonormal basis $\{|\phi_i\rangle : i\}$. **Orthonormality** means

$$\langle\phi_i|\phi_j\rangle = \delta_{i,j}$$

.

Multiplying both sides of this by X on the left

$$X = \sum_{k=1}^{\mathcal{F}} \underbrace{\underbrace{Xw_k}_{Y_k} w_k^T}_{X_k} \tag{78.30}$$

Define $X^{\geq\Phi}$ by

$$\begin{aligned} X^{\geq\Phi} &= X - \sum_{k=1}^{\Phi-1} \underbrace{Xw_k w_k^T}_{X_k} && (78.31) \\ &= \sum_{k=\Phi}^{\mathcal{F}} X_k && (78.32) \end{aligned}$$

Multiplying $X^{\geq\Phi}$ on the right by w_a gives

$$\begin{aligned} \underbrace{X^{\geq\Phi} w_a}_{Y_a^{\geq\Phi}} &= \underbrace{Xw_a}_{Y_a}[1 - \mathbb{1}(a < \Phi)] && (78.33) \\ &= Y_a \mathbb{1}(a \geq \Phi) && (78.34) \end{aligned}$$

See Fig.78.1 for a pictorial representation of Eqs.78.32 and 78.34.

Now we can define the Φ'th PA in terms of $X^{\geq\Phi}$ by

$$\begin{aligned} w_\Phi &= \underset{w_\Phi:\, w_\Phi^T w_\Phi = 1}{\operatorname{argmax}} \; w_\Phi^T X^{\geq\Phi} X^{\geq\Phi} w_\Phi && (78.35) \\ &= \underset{w_\Phi}{\operatorname{argmax}} \frac{w_\Phi^T X^{\geq\Phi} X^{\geq\Phi} w_\Phi}{w_\Phi^T w_\Phi} && (78.36) \end{aligned}$$

- **PCA calculation**

  Note that $Y = XW$, where X is the initial dataset and W is the matrix that diagonalizes the covariance matrix $X^T X$. Hence, calculating Y does not require calculating the SVD of X (in particular, it does not require calculating U).

  PCA calculation via Covariance Matrix

  1. Subtract mean from initial dataset X
  2. Diagonalize covariance matrix

$$X^T X = \sum_j \lambda_j^2 w_j w_j^T \tag{78.37}$$

Figure 78.1: Pictorial representation of Eqs.78.32and 78.34. This figure is not accurate in the sense that the vectors X_1, X_2, X_3 are in reality not coplanar but rather mutually orthogonal. Also, $X^{\geq \Phi}$ tends to zero as $\Phi \to \mathcal{F}$.

3. Order eigenvalues in diagonal of Λ^2 and corresponding columns of W so that eigenvalues λ_i^2 are in decreasing order.
4. Calculate $Y = XW$

Iterative calculation of PCA (avoids calculating covariance matrix)

Let

$r_j^{(0)} \in \mathbb{R}^{\mathcal{F}\times 1}$ random column vector.

$s_j^{(0)} = 0^{\mathcal{F}\times 1}$ zero column vector .

$$x_j = [X_{\sigma,j}]_{\sigma=1}^{N} \text{ (column vector)} \tag{78.38}$$

$$s_j^{(a+1)} = s_j^{(a)} + x_j[x_j^T r_j^{(a)}] \tag{78.39}$$

$$\lambda_j^{(a)} = r_j^{(a)T} s_j^{(a)} \tag{78.40}$$

$$r_j^{(a+1)} = \frac{s_j^{(a+1)}}{\| s_j^{(a+1)} \|} \tag{78.41}$$

If we calculate $s_j^{(a)}$ for large a, and for $j = 1, 2, \ldots \mathcal{F}$, we get an approximation of $X^T X r^{(a)} \to w_1$ and $r^{(a)T} X^T X r^{(a)} \to \lambda_1$ where (λ_1, w_1) are the first PV and PA of X.

Higher order (PV, PA) pairs are obtained by **deflation**. For example, the second (PV, PA) pair is obtained by replacing X by $X - \lambda_1 w_1 w_1^T$ and following the same algorithm that we used to calculate the 1st (PV, PA) pair of X.

Figure 78.2: Scree plot. (from Wikipedia entry "Scree plot") For a geologist, a scree is the edge of a cliff. It looks just like the blue curve in this figure, like a decaying exponential.

- **Plots**

  This section is based on Ref.[116].

  Suppose

$$X = U\Lambda W^T = \sum_{f=0}^{\mathcal{F}} \lambda_f u_f w_f^T \tag{78.42}$$

  where $\lambda_f \in \mathbb{R}$, $u_f \in \mathbb{R}^{N\times 1}$ and $w_f \in \mathbb{R}^{\mathcal{F}\times 1}$.

  Suppose the unit column vectors $w_1, w_2 \in \mathbb{R}^{\mathcal{F}\times 1}$ are the first two PA, and λ_1^2, λ_2^2 are the corresponding PVs.

  PC scores are defined as

$$\left(\frac{(Xw_1)_\sigma}{\sqrt{\lambda_1}}, \frac{(Xw_2)_\sigma}{\sqrt{\lambda_2}}\right) = (\sqrt{\lambda_1}(u_1)_\sigma, \sqrt{\lambda_2}(u_2)_\sigma) \tag{78.43}$$

  for all data points σ.

  Let $e_f \in \mathbb{R}^{\mathcal{F}\times 1}$ be a one hot vector with 1 at position $f = 1, 2 \ldots, \mathcal{F}$.

  The **loadings** (i.e., weights) are defined as

$$(\sqrt{\lambda_1} e_f^T w_1, \sqrt{\lambda_2} e_f^T w_2) = (\sqrt{\lambda_1}(w_1)_f, \sqrt{\lambda_2}(w_2)_f) \tag{78.44}$$

  for all features f.

Fig.78.2 is a plot of the PVs λ_i^2 versus i. Such a plot is called a **scree plot**.

Fig.78.3 shows three types of plots: a **PC scores plot**, a **loadings plot**, and a biplot. A **biplot** is a superposition of the PC scores plot and the loadings plot.

Figure 78.3: PC scores plot, loadings plot, and biplot. (from the documentation for the Prism software made by graphpad.com)

Chapter 79

Program evaluation and review technique (PERT)

This chapter is based on Refs.[82] and [175].

PERT diagrams are used for scheduling a project consisting of a series of interdependent activities and estimating how long it will take to finish the project. PERT diagrams were invented by the NAVY in 1958 to manage a submarine project. Nowadays they are taught in many business and management courses.

A **PERT diagram** is a Directed Acyclic Graph (DAG) with the following properties. (See Fig.79.2 for an example of a PERT diagram). The nodes $\underline{E}_i$ for $i = 1, 2, \ldots, ne$ of a PERT diagram are called **events**. The edges $i \to j$ of a PERT diagram are called **activities**. An event represents the starting (kickoff) date of one or more activities. A PERT diagram has a single root node ($i = 1$, start event) and a single leaf node ($i = ne$, end event).

The PERT diagram user must initially provide a **Duration Times (DT) table** which gives $(DO_{i\to j}, DP_{i\to j}, DM_{i\to j})$ for each activity $i \to j$, where

$DO_{i\to j}$= optimistic duration time of activity $i \to j$

$DP_{i\to j}$= pessimistic duration time of activity $i \to j$

$DM_{i\to j}$= median duration time of activity $i \to j$

From the DT table, one calculates:

Duration time of activity $i \to j$

$$D_{i\to j} = \frac{1}{6}(DO_{i\to j} + DP_{i\to j} + 4DM_{i\to j}) \tag{79.1}$$

Duration Variance of activity $i \to j$

$$V_{i\to j} = \left(\frac{DO_{i\to j} - DP_{i\to j}}{DM_{i\to j}}\right)^2 \tag{79.2}$$

Often, it is convenient to define "dummy" edges with $D_{i\to j} = 0$. That is perfectly fine.

Define:

TES_i = Earliest start time for event i
TLS_i = Latest start time for event i
$slack_i = TLS_i - TES_i$ = slack for event i
$TEF_{i\to j} = TES_i + D_{i\to j}$ = Earliest finish time for activity $i \to j$.
$TLF_{i\to j} = TLS_j - D_{i\to j}$ = Latest finish time for activity $i \to j$. See footnote below. [1]

A **critical path** is a directed path (i.e., a chain of connected arrows, all pointing in the same direction) going from the start to the end node, such that slack equals zero at every node visited. In a DAG, the neighbors of a node is the union of its parent and children nodes. A critical path must also have all other nodes as neighbors; i.e, the union of the neighbors of every node in the path plus the nodes in the path itself, equals all nodes in the graph.

GOAL of PERT analysis: The main goal of PERT analysis is to find, based on the data of the DT table, the interval $[TES_i, TLS_i]$ giving a lower and an upper bound to the starting time of each node i. Another goal is to find a critical path for the PERT diagram (which represents an entire project). By adding the $D_{i\to j}$ of each edge of the critical path, one can get the mean value of the total duration of the entire project, and by adding the variances of each edge along the critical path, one can get an estimate of the total variance of the total duration. Knowing the mean and variance of the total duration and assuming a Normal distribution, one can predict the probability that the actual duration will deviate by a certain amount from its mean.

To calculate the interval $[TES_i, TLS_i]$, one follows the following two steps.

1. Assume $TES_1 = 0$ and solve

$$TES_i = \max_{a\in pa(i)} (\underbrace{TES_a + D_{a\to i}}_{TEF_{a\to i}}) \tag{79.3}$$

   for $i \in [2, ne]$. This recursive equation is solved by what is called "forward propagation", wherein one moves up the list of nodes i in order of increasing i starting at $i = 1$ with $TES_1 = 0$.

2. Assume $TLS_{ne} = TES_{ne}$ and solve

$$TLS_i = \min_{b\in ch(i)} (\underbrace{TLS_b - D_{i\to b}}_{TLF_{i\to b}}) \tag{79.4}$$

[1] In the popular educational literature, the edge variables $TEF_{i\to j}$ and $TLF_{i\to j}$ are sometimes associated with the nodes, but they are clearly edge variables. This makes things confusing. The reason this is done is that some software draws PERT diagrams as trees whereas other software draws them as DAGs. For trees, storing $TEF_{i\to j}$ and $TLF_{i\to j}$ in a node makes some sense but not for DAGs. You will notice that giving specific names to the variables $TEF_{i\to j}$ and $TLF_{i\to j}$ is unnecessary. It is possible to delete all mention of their names from this chapter without losing any details. I only declare their names in this chapter so as tell the reader what they are in case he/she hears them mentioned and wonders what they are equal to in our notation.

for $i \in [1, ne-1]$. This recursive equation is solved by what is called "backward propagation", wherein one moves down the list of nodes i in order of decreasing i starting at $i = ne$ with $TLS_{ne} = TES_{ne}$. TES_{ne} is known from step 1.

Eqs.(79.3) and (79.4) are illustrated in Fig.79.1.

Figure 79.1: TES_i defined from info received from parents of i and TLS_i defined from info received from children of i.

79.1 Example

To illustrate PERT analysis, we end with an example. We present the example in the form of an exercise question and then provide the answer. This example comes from Ref.[82], except for part (e) about bnets, which is our own.

Question: For the PERT diagram of Fig.79.2, calculate the following:

(a) Interval $[TES_i, TLS_i]$ for all i.

(b) A critical path for this PERT diagram.

(c) The mean and variance of the total duration of the critical path.

(d) The probability that the total duration will be 225 days or less.

(e) A bnet interpretation of this problem.

Answer to (a) $[TES_i, TLS_i]$ are given by Fig.79.3.

Answer to (b) The critical path is given in red in Fig.79.3. Note that this path does indeed have zero slack at each node it visits and the union of its neighborhood and the path itself encompasses all nodes.

Answer to (c) The mean and variance of the total duration are calculated in Table 79.1.

Answer to (d)

$$P(\underline{x} < 225) = P\left[\frac{\underline{x}-\mu}{\sigma} \leq \frac{225-220}{\sqrt{7.73}}\right] \tag{79.5}$$

$$= P[\underline{z} \leq 1.80] \tag{79.6}$$

$$= 0.9641 \tag{79.7}$$

Figure 79.2: Example of a PERT diagram. The numbers attached to the arrows are the duration times $D_{i\to j}$ in days followed by, enclosed in parentheses, the variance $V_{i\to j}$ of that duration. The info given in this PERT diagram was derived from a DT table in Ref.[82]. The info in this PERT diagram is sufficient for calculating TES_i and TLS_i for each node i. The results of that calculation are given in Fig.79.3.

TES_i (given after the node name) for node $\underline{E}_i$ for all i

TLS_i (given after the node name) for node $\underline{E}_i$ for all i

Figure 79.3: Results of calculating TES_i for all i via a forward pass, followed by calculating TLS_i for all i via a backward pass. Critical path indicated in red.

Answer to (e) Define 2 bnets.

1. The first PERT bnet is for calculating TES_i for all i and is given by Fig.79.4.

edge $i \to j$	duration $D_{i\to j}$	variance $V_{i\to j}$
A $(1 \to 2)$	50	2.56
D $(2 \to 5)$	60	0.44
G $(5 \to 7)$	30	1.78
J $(7 \to 8)$	40	1.31
K $(8 \to 9)$	10	0.64
L $(9 \to 10)$	30	1.00
Total	220	7.73

Table 79.1: Calculation of mean and variance of total duration along critical path.

Figure 79.4: bnet for TES_i calculation.

The TPMs, printed in blue, for the bnet Fig.79.4, are as follows (this equation is to be evaluated recursively by a forward pass through the bnet):

$$P(TES_i|(TES_a)_{a\in pa(i)}) = \delta(TES_i, \max_{a\in pa(i)}(TES_a + D_{a\to i})) \tag{79.8}$$

2. The second PERT bnet is for calculating TLS_i for all i and is given by Fig.79.5. Note that the directions of all the arrows in the PERT diagram Fig.79.2 have been reversed so Fig.79.5 is a time reversed graph.

   The TPMs, printed in blue, for the bnet Fig.79.5, are as follows (this equation is to be evaluate recursively by a backward pass through the bnet):

$$P(TLS_i|(TLS_b)_{b\in pa(i)}) = \delta(TLS_i, \min_{b\in pa(i)}(TLS_b - D^T_{b\to i})) \ , \tag{79.9}$$

   where $D^T_{i\to j} = D_{j\to i}$.

Figure 79.5: bnet for TLS_i calculation.

Chapter 80

Random Forest and Bagging

This chapter is based on Refs.[118] and [176].

Chapter 16 defines decision trees (dtrees) and explains how to construct them. A Random Forest (RF) is an ensemble of dtrees. The RF algorithm is a method of, given a dataset, constructing a RF and averaging over the classifier functions of the RF to produce an ensemble classifier.

The RF algo uses the method of Bootstrap Aggregating (a.k.a. Bagging), which is discussed in detail below. Bagging is a method of constructing an ensemble of datasets (called **bootstrap datasets or bags**) that are fairly uncorrelated. Each of these bags is used to train a **bag-classifier**. The bag-classifiers are averaged over to produce an **ensemble classifier, or e-classifier** for short. Bagging can be used to train any type of bag-classifier, but it was invented with dtrees in mind, and is still most commonly used to train dtrees.

Boosting (see Chapter 1 on AdaBoost and Chapter 111 on XGBoost) is another method, besides Bagging, of constructing a classifier function from an ensemble of classifier functions. These two methods are most commonly applied to dtrees: Boosting for an ensemble of small dtrees, and Bagging for a random forest (which is an ensemble of dtrees that are usually much more complicated than small dtrees).

80.1 Bagging (with fully-featured bags)

In this section we discuss the bagging algorithm. As already mentioned, bagging is usually used to train dtrees. In this section, we explain bagging in general, not just for dtrees.

Let $L = [0, 1, 2, \ldots, nsam-1]$ be a list of individuals (samples) in a population. In this chapter, we will use the notation $A^\sigma = A[\sigma]$ and $\vec{A} = [A^\sigma : \sigma \in L]$ for a list (vector, 1-D array) indexed by L. We will refer to $DS = (\vec{x}, \vec{y})$ where $x^\sigma \in val(\underline{x})$, $y^\sigma \in val(\underline{y})$, as a dataset. If L_j is a list (possibly with duplicate items) such that $set(L_j) \subset set(L)$, then define $DS_j = (\vec{x}, \vec{y})_{L_j} = ((x^\sigma)_{\sigma \in L_j}, (y^\sigma)_{\sigma \in L_j})$. We will refer to DS_j as the **restriction of** $(\vec{x}, \vec{y})$ **to** L_j**.**

Figure 80.1: Bnet for Random Forest (RF) with 5 bags.

We will refer to a function $Y : val(\underline{x}) \to val(\underline{c})$ as a classifier. It maps vector of vector of features $x \in val(\underline{x})$ to a class $c \in val(\underline{c})$. Below, Y_b for all b and Y_{ens} are classifiers.

Fig.80.1 is a bnet that encapsulates the RF algo. The TPMs, printed in blue, for the bnet Fig.80.1, are as follows.

Let $b \in \{0, 1, 2, \ldots, nbags - 1\} = B$ and $\sigma \in L$. Let $L_b^\sigma \in L$ and

$$P(L_b^\sigma) = 1/nsam \tag{80.1}$$

In other words, each item in list L_b is chosen from the items of list L, uniformly at random with replacements. $|L_b| = |L|$ (same size as original). L_b can have duplicate items and be missing items from L.

$$P((\vec{x}, \vec{y})_{L_b} | (\vec{x}, \vec{y}), L_b) = \mathbb{1}(\quad (\vec{x}, \vec{y})_{L_b} = \text{restriction of } (\vec{x}, \vec{y}) \text{ to } L_b \quad) \tag{80.2}$$

We will refer to $(\vec{x}, \vec{y})$ as the **original dataset** and to the $(\vec{x}, \vec{y})_{L_b}$ for $b \in B$ as the **bootstrap datasets** or **bags**.

$$P(Y_b | (\vec{x}, \vec{y})_{L_b}) = \mathbb{1}(\quad Y_b(\cdot) = \text{classifier trained on dataset } (\vec{x}, \vec{y})_{L_b}. \quad) \tag{80.3}$$

$$P(Y_{ens} | (Y_b)_{b \in B}) = \prod_\sigma \mathbb{1}(\quad Y_{ens}(x^\sigma) = \texttt{majority}(\{Y_b(x^\sigma) : b \in B\}) \quad) \tag{80.4}$$

The `majority()` function can be replaced by an average $\frac{1}{nbags} \sum_b$ in case the set of classes $val(\underline{c})$ equals $\mathbb{R}$ rather than a finite set. We will refer to Y_{ens} as the **ensemble classifier (e-classifier)** and to the Y_b as the **bag-classifiers**.

Figure 80.2: $\Sigma(b)$ and $\Sigma^c(b)$ are disjoint sets whose union is Σ. $B(\sigma)$ and $B^c(\sigma)$ are disjoint sets whose union is B.

Define (these definitions are illustrated in Fig.80.2)

$$\Sigma = set(L)\ ,\ \ \Sigma(b) = set(L_b)\ ,\ \ \Sigma^c(b) = \Sigma - \Sigma(b) \tag{80.5}$$

and

$$B(\sigma) = \{b \in B : \sigma \in \Sigma(b)\}\ ,\ \ B^c(\sigma) = B - B(\sigma) \tag{80.6}$$

$\Sigma(b)$ is the set of **in-the-b-bag individuals** and $\Sigma^c(b)$ is the set of **out-of-the-b-bag (OOB) individuals**. $B(\sigma)$ is the set of bags that contain individual σ, and $B^c(\sigma)$ is the set of bags that don't.

The **OOB error** is defined as

$$err = \sum_{\sigma \in L} \mathbb{1}(B^c(\sigma) \neq \emptyset)\mathbb{1}(\quad y^\sigma \neq \texttt{majority}([Y_b(x^\sigma) : b \in B^c(\sigma)])\quad)\ . \tag{80.7}$$

Empirical results supposedly show that OOB error is comparable in accuracy to the error calculated by doing cross-validation (CV) (see Chapter 13), although CV error is considered more dependable.

80.2 Bagging (with randomly-shortened bags)

Suppose the feature vector x^σ in the dataset $DS = (\vec{x}, \vec{y})$ has nf components; i.e., $x^\sigma = (x_0^\sigma, x_1^\sigma, \ldots, x_{nf-1}^\sigma) \in val(\underline{x}_0) \times val(\underline{x}_1) \times \ldots \times val(\underline{x}_{nf-1}) = val(\underline{x})$.

For each bag DS_b, one chooses at random $nf' = \sqrt{nf}$ out of the nf features, and discards the remaining features from DS_b, thus producing a new, **randomly-shortened-bag (rs-bag)** DS'_b. Each rs-bag is then used to train a bag-classifier, usually a dtree, using the methods for dtree SL described in Chapter 16. Using rs-bags is called the **random subspace method**. The reason for using rs-bags is that they further decorrelate the set of bags used to train bag-classifiers.

Chapter 81

Recurrent Neural Networks

This chapter is mostly based on Ref.[52].

This chapter assumes you are familiar with the material and notation of Chapter 68 on plain Neural Nets.

$$
\begin{tikzcd}
\underline{x}(\cdot) \arrow[d] \arrow[dr] \arrow[drr] & & \\
\underline{x}(0) \arrow[d] & \underline{x}(1) \arrow[d] & \underline{x}(2) \arrow[d] \\
\underline{h}(0) \arrow[r] \arrow[d] & \underline{h}(1) \arrow[r] \arrow[d] & \underline{h}(2) \arrow[d] \\
\underline{Y}(0) & \underline{Y}(1) & \underline{Y}(2)
\end{tikzcd}
$$

Figure 81.1: Simple example of RNN with $T = 3$

Suppose
T is a positive integer.
$t = 0, 1, \ldots, T-1$,
$\underline{x}_i(t) \in \mathbb{R}$ for $i = 0, 1, \ldots, nx-1$,
$\underline{h}_i(t) \in \mathbb{R}$ for $i = 0, 1, \ldots, nh-1$,
$\underline{Y}_i(t) \in \mathbb{R}$ for $i = 0, 1, \ldots, ny-1$,
$W^{h|x} \in \mathbb{R}^{nh \times nx}$,
$W^{h|h} \in \mathbb{R}^{nh \times nh}$,
$W^{y|h} \in \mathbb{R}^{ny \times nh}$,
$b^y \in \mathbb{R}^{ny}$,
$b^h \in \mathbb{R}^{nh}$.
Henceforth, $x(\cdot)$ will mean the array of $x(t)$ for all t.

The simplest kind of recurrent neural network (RNN) has the bnet Fig.81.1 with arbitrary T. The node TPMs, printed in blue, for this bnet, are as follows.

$$P(x(\cdot)) = \text{given} \tag{81.1}$$

$$P(x(t)) = \delta(x(t), [x(\cdot)]_t) \tag{81.2}$$

$$P(h(t) \mid h(t-1), x(t)) = \delta(h(t), \mathcal{A}(W^{h|x}x(t) + W^{h|h}h(t-1) + b^h)) \ , \tag{81.3}$$

where $h(-1) = 0$.

$$P(Y(t) \mid h(t)) = \delta(Y(t), \mathcal{A}(W^{y|h}h(t) + b^y)) \tag{81.4}$$

Define

$$W^h = [W^{h|x}, W^{h|h}, b^h] \ , \tag{81.5}$$

and

$$W^y = [W^{y|h}, b^y] \ . \tag{81.6}$$

The bnet of Fig.81.1 can be used for classification once its parameters W^h and W^y have been optimized. To optimize those parameters via gradient descent, one can use the bnet of Fig.81.2.

Let $\sigma = 0, 1, \ldots, nsam(\vec{x}) - 1$ be the labels for a batch of samples. Below, we will write $A^\sigma = A[\sigma]$ for the σ component of any vector $\vec{A}$. The TPMs, printed in blue, for bnet Fig.81.2, are as follows.

$$P(x(\cdot)^\sigma) = \text{given} \tag{81.7}$$

$$P(x(t)^\sigma) = \delta(x(t)^\sigma, [x(\cdot)]_t^\sigma) \tag{81.8}$$

$$P(h(t)^\sigma \mid h(t-1)^\sigma, x(t)^\sigma) = \delta(h(t)^\sigma, \mathcal{A}(W^{h|x}x(t)^\sigma + W^{h|h}h(t-1)^\sigma + b^h)) \tag{81.9}$$

$$P(Y(t)^\sigma \mid h(t-1)^\sigma) = \delta(Y(t)^\sigma, \mathcal{A}(W^{y|h}h(t-1)^\sigma + b^y)) \tag{81.10}$$

Figure 81.2: RNN bnet used to optimize parameters W^h and W^y of RNN bnet Fig.81.1.

$$P(y(\cdot)^\sigma \mid x(\cdot)^\sigma) = \text{given} \tag{81.11}$$

$$P(\mathcal{E}(t) \mid \vec{y}(t), \vec{Y}(t)) = \frac{1}{nsam(\vec{x})} \sum_\sigma d(y(t)^\sigma, Y(t)^\sigma) \ , \tag{81.12}$$

where

$$d(y, Y) = |y - Y|^2 \ . \tag{81.13}$$

If $y, Y \in [0,1]$, one can use this instead

$$d(y,Y) = XE(y \to Y) = -y \ln Y - (1-y)\ln(1-Y) \ . \tag{81.14}$$

$$P(\mathcal{E} \mid \mathcal{E}(\cdot)) = \delta(\mathcal{E}, \sum_t \mathcal{E}(t)) \tag{81.15}$$

For $a = h, y$,

$$P(W^a) = \text{given} \ . \tag{81.16}$$

The first time it is used, W^a is arbitrary. Afterwards, it is determined by previous horizontal stage.

$$P((W^a)'|\mathcal{E}, W^a) = \delta((W^a)', W^a - \eta^a \partial_{W^a}\mathcal{E}) \ . \tag{81.17}$$

$\eta^a > 0$ is the learning rate for W^a.

81.1 Language Sequence Modeling

Estimate $P(x(\cdot))$ empirically. We can use this to:

- predict the probability of a sentence,

  example: Get $P(x(0), x(1), x(2))$.

- predict the most likely next word in a sentence,

  example: Get $P(x(2)|x(0), x(1))$.

- generate fake sentences.

  example:

  Get $x(0) \sim P(x(0))$.

  Next get $x(1) \sim P(x(1)|x(0))$.

  Next get $x(2) \sim P(x(2)|x(0), x(1))$.

81.2 Other types of RNN

Let $\mathcal{T} = \{0, 1, \ldots, T-1\}$, and $\mathcal{T}^x, \mathcal{T}^y \subset \mathcal{T}$. Above, we assumed that $\underline{x}(t)$ and $\underline{Y}(t)$ were both defined for all $t \in \mathcal{T}$. More generally, they might be defined only for subsets of $\mathcal{T}$: $\underline{x}(t)$ for $t \in \mathcal{T}^x$ and $\underline{Y}(t)$ for $t \in \mathcal{T}^y$. If $|\mathcal{T}^x| = 1$ and $|\mathcal{T}^y| > 1$, we say the RNN bnet is of the 1 to many kind. In general, can have **1 to 1, 1 to many, many to 1, many to many** RNN bnets.

Plain RNNs can suffer from the **vanishing or exploding gradients problem**. There are various ways to mitigate this (e.g., good choice of initial W^h and

Figure 81.3: RNN bnet of the many to many kind. This one can be used for translation. $x(0)$ and $x(1)$ might denote two words of an English sentence, and $Y(2)$ and $Y(3)$ might be their Italian translation.

W^y, good choice of activation functions, regularization). Or by using GRU or LSTM (discussed below). **GRU and LSTM** were designed to mitigate the vanishing or exploding gradients problem. They are very popular in NLP (Natural Language Processing).

81.2.1 Long Short Term Memory (LSTM) unit (1997)

This section is based on Wikipedia article Ref.[160]. In this section, $\odot$ will denote the Hadamard matrix product (elementwise product).

Figure 81.4: bnet for a Long Short Term Memory (LSTM) unit.

Let
$\underline{x}(t) \in \mathbb{R}^{nx}$: **input state** vector to the LSTM unit
$\underline{f}(t) \in \mathbb{R}^{nh}$: **forget activation** vector
$\underline{i}(t) \in \mathbb{R}^{nh}$: **input activation** vector
$\underline{o}(t) \in \mathbb{R}^{nh}$: **output activation** vector
$\underline{h}(t) \in \mathbb{R}^{nh}$: **hidden state** vector
$\underline{\tilde{c}}(t) \in \mathbb{R}^{nh}$: **cell activation** vector
$\underline{c}(t) \in \mathbb{R}^{nh}$: **cell state** vector
$\underline{Y}(t) \in \mathbb{R}^{ny}$: **classification** of $x(t)$.
$W \in \mathbb{R}^{nh \times nx}$, $U \in \mathbb{R}^{nh \times nh}$ and $b \in \mathbb{R}^{nh}$: weight matrices and bias vectors, parameters learned by training.
$\mathcal{W}^{y|h} \in \mathbb{R}^{ny \times nh}$: weight matrix

Fig.81.4 is a bnet for a LSTM unit. The TPMs, printed in blue, for this bnet, are as follows.

$$P(f(t)|x(t), h(t-1)) = \mathbb{1}(\quad f(t) = \text{smoid}(W^{f|x}x(t) + U^{f|h}h(t-1) + b^f) \quad) \,, \tag{81.18}$$

where $h(-1) = 0$.

$$P(i(t)|x(t), h(t-1)) = \mathbb{1}(\quad i(t) = \text{smoid}(W^{i|x}x(t) + U^{i|h}h(t-1) + b^i) \quad) \tag{81.19}$$

$$P(o(t)|x(t), h(t-1)) = \mathbb{1}(\quad o(t) = \text{smoid}(W^{o|x}x(t) + U^{o|h}h(t-1) + b^o) \quad) \tag{81.20}$$

$$P(\widetilde{c}(t)|x(t), h(t-1)) = \mathbb{1}(\quad \widetilde{c}(t) = \tanh(W^{c|x}x(t) + U^{c|h}h(t-1) + b^c) \quad) \tag{81.21}$$

$$P(c(t)|f(t), c(t-1), i(t), \widetilde{c}(t)) = \mathbb{1}(\quad c(t) = f(t) \odot c(t-1) + i(t) \odot \widetilde{c}(t) \quad) \tag{81.22}$$

$$P(h(t)|o(t), c(t)) = \mathbb{1}(\quad h(t) = o(t) \odot \tanh(c(t)) \quad) \tag{81.23}$$

$$P(Y(t)|h(t)) = \mathbb{1}(\quad Y(t) = \mathcal{A}(\mathcal{W}^{y|h}h(t) + b^y) \quad) \tag{81.24}$$

81.2.2 Gated Recurrence Unit (GRU) (2014)

This section is based on Wikipedia article Ref.[140]. In this section, $\odot$ will denote the Hadamard matrix product (elementwise product).

GRU is a more recent (17 years later) attempt at simplifying LSTM.

Figure 81.5: bnet for a Gated Recurrent Unit (GRU).

Let
$\underline{x}(t) \in \mathbb{R}^{nx}$: **input state** vector
$\underline{h}(t) \in \mathbb{R}^{nh}$: **hidden state** vector
$\widehat{\underline{h}}(t) \in \mathbb{R}^{nh}$: **hidden activation** vector
$\underline{z}(t) \in \mathbb{R}^{nh}$: **update activation** vector
$\underline{r}(t) \in \mathbb{R}^{nh}$: **reset activation** vector
$\underline{Y}(t) \in \mathbb{R}^{ny}$: **classification** of $x(t)$.
$W \in \mathbb{R}^{nh \times nx}$, $U \in \mathbb{R}^{nh \times nh}$ and $b \in \mathbb{R}^{nh}$: weight matrices and bias vectors, parameters learned by training.
$\mathcal{W}^{y|h} \in \mathbb{R}^{ny \times nh}$: weight matrix

Fig.81.5 is a bnet for a GRU. The TPMs, printed in blue, for this bnet, are as follows.

$$P(z(t)|x(t), h(t-1)) = \mathbb{1}(\quad z(t) = \text{smoid}(W^{z|x}x(t) + U^{z|h}h(t-1) + b^z) \quad) \ , \tag{81.25}$$

where $h(-1) = 0$.

$$P(r(t)|x(t), h(t-1)) = \mathbb{1}(\quad r(t) = \text{smoid}(W^{r|x}x(t) + U^{r|h}h(t-1) + b^r) \quad) \tag{81.26}$$

$$P(\widehat{h}(t)|x(t), r(t), h(t-1)) = \mathbb{1}(\quad \widehat{h}(t) = \tanh(W^{h|x}x(t) + U^{h|h}(r(t) \odot h(t-1)) + b^h) \quad) \tag{81.27}$$

$$P(h(t)|z(t), h(t-1), \widehat{h}(t)) = \mathbb{1}(\quad h(t) = (1 - z(t)) \odot h(t-1) + z(t) \odot \widehat{h}(t) \quad) \tag{81.28}$$

$$P(Y(t)|h(t)) = \mathbb{1}(\quad Y(t) = \mathcal{A}(\mathcal{W}^{y|h} h(t) + b^y) \quad) \tag{81.29}$$

Chapter 82

Regression Discontinuity Design

This chapter is based on Ref.[15].

This chapter assumes that the reader has read Chapter 77 on Potential Outcomes (PO).

In Regression Discontinuity Design (RDD), one switches the treatment dose $\underline{d}$ from 0 when $\underline{x} < \xi$ to 1 where $\underline{x} > \xi$, where $\underline{x}$ is an observed confounder (call it the **switch confounder**) and ξ is a threshold value for $\underline{x}$. One measures the jump δ in the treatment outcome $\underline{y}$ as $\underline{x}$ passes through $\underline{x} = \xi$. Then one makes the very reasonable assumption that δ equals[1] $\mathcal{Y}_{1|x=\xi} - \mathcal{Y}_{0|x=\xi} = ATE_{|x=\xi}$ for an imaginary experiment in which the confounder $\underline{x}$ acts as a normal confounder that doesn't switch the treatment dose $\underline{d}$.

For example, d^σ might be whether an individual is admitted to Harvard Univ., y^σ might be how much money the individual earns for the first 20 years after graduating from Harvard, and x^σ might be his SAT scores. We assume Harvard only admits students with an SAT score higher than ξ.

82.1 PO analysis

The TPMs, printed in blue, for the bnet G_{disc} shown in Fig.82.1, are as follows. Note that the TPMs for the bnet G_{disc} are defined in terms of the TPMs for the bnet G.

$$P(x^\sigma) = \delta(x^\sigma, x) \tag{82.1}$$

$$P(d^\sigma | x^\sigma = x) = \begin{cases} \delta(d^\sigma, 1) & \text{for } x > \xi \\ \delta(d^\sigma, 0) & \text{for } x < \xi \end{cases} \tag{82.2}$$

$$P(y^\sigma | y^\sigma(0), y^\sigma(1), d^\sigma) = \mathbb{1}(y^\sigma = y^\sigma(d^\sigma)) \tag{82.3}$$

[1] ATE, which stands for "average treatment effect", is defined in Chapter 77.

Figure 82.1: 2 bnets used in the PO analysis of RDD. The TPMs for G_{disc} are defined in terms of the TPMs for G. The TPM $P(d^\sigma|x^\sigma)$ for G_{disc} is discontinuous in x^σ.

$$P(y^\sigma(c)|x) = P(y^\sigma(c)|d^\sigma, x) = \text{given} \tag{82.4}$$

Define

$$E_{\sigma|x}[y^\sigma(c)] = E_{\underline{y}(c)|x}[\underline{y}(c)] = \mathcal{Y}_{c|x} \tag{82.5}$$

and

$$\xi\pm = \xi \pm \epsilon \tag{82.6}$$

for some infinitesimal $\epsilon > 0$.

See Fig.82.2. In RDD, we assume that if we define the following 2 δ's, one for bnet G and the other for bnet G_{disc}, then the two δ's are equal, and they equal a conditional ATE.

$$\delta_{G_{disc}} = \mathcal{Y}_{1|x=\xi+} - \mathcal{Y}_{0|x=\xi-} \tag{82.7}$$

$$\delta_G = \mathcal{Y}_{1|x=\xi} - \mathcal{Y}_{0|x=\xi} \tag{82.8}$$

$$\delta_G = \delta_{G_{disc}} = \delta \tag{82.9}$$

$$\delta = ATE_{|x=\xi} \tag{82.10}$$

Figure 82.2: The jump δ between $\mathcal{Y}_{1|x}$ and $\mathcal{Y}_{0|x}$ is the same for G and G_{disc}.

82.2 Linear Regression

In this section, we show how to apply linear regression (LR) to the PO analysis of RDD.

y^σ can be fitted as a function of $x \in \mathbb{R}$, for $c^\sigma \in \{0, 1\}$, as follows. Here ϵ^σ is the residual for individual σ and $b_0, m_0, b_1, m_1 \in \mathbb{R}$ are the fit parameters.

$$y^\sigma = [b_0 + m_0(x - \xi)](1 - c^\sigma) + [b_1 + m_1(x - \xi)]c^\sigma + \epsilon^\sigma \ . \tag{82.11}$$

Note that Eq.(82.11) yields a straight line in the $y^\sigma - x$ plane for $c^\sigma = 0$, and another straight line for $c^\sigma = 1$. These 2 lines are colored magenta in Fig.82.2. We are using the standard symbols b to denote the y-intercept, and m to denote the slope of a straight line.

Taking the expected value of Eq.(82.11), we get

$$\mathcal{Y}_{c|x} = [b_0 + m_0(x - \xi)](1 - c) + [b_1 + m_1(x - \xi)]c \ . \tag{82.12}$$

Hence,

$$\mathcal{Y}_{1|x=\xi+} = b_1 \ , \ \ \mathcal{Y}_{0|x=\xi-} = b_0 \tag{82.13}$$

$$\delta = b_1 - b_0 \tag{82.14}$$

Chapter 83

Regularization of Loss Functions

The topic of this chapter is Regularization of Loss functions (ROLF).

ROLF is the practice of adding a convex function $\mathcal{R} : \mathbb{R}^n \to \mathbb{R}$ (called the **regulator function**[1]) to a convex function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ (called the **loss function**[2]), when one wishes to minimize the loss function. In Machine Learning, the variable being minimized over is often the weights of a Neural Net, so we will denote it by $w \in \mathbb{R}^n$, and use $w^* \in \mathbb{R}^n$ to represent the minimum of $\mathcal{L}^+(w) = \mathcal{L}(w) + \mathcal{R}(w)$.

In some cases (like in L^p norm ROLF, which we discuss below), the minimum of $\mathcal{L}^+(w)$ and that of $\mathcal{L}(w)$ are not the same but very close. In such cases, displacing the minimum of $\mathcal{L}(w)$ might be done in order to avoid overfitting, or to spread out degenerate solutions, or to produce sparse solutions.

In other cases (like in proximal ROLF, which we discuss below), the minimum of $\mathcal{L}^+(w)$ and that of $\mathcal{L}(w)$ are the same. In such cases, ROLF might be done to improve the convergence properties of a sequence of points $\{w_k\}_{k=0}^{\infty}$ such that $w_k \to w^*$ as $k \to \infty$.

There are many methods of biasing the minimum of a convex function that don't involve adding a regulator function. For example, Early Stopping of training and Cross Validation for Neural Nets, or adding Latent Variables to a Bayesian Network. Or Constraint Optimization where hard equality and/or inequality constraints are imposed (as in Linear Programming, Lagrange multipliers, Khun-Tucker conditions, method of simple substitution of constraints). We won't discuss those types of ROLFs in this chapter[3], except for a brief section on latent nodes at the end.

Loss functions commonly used in Statistics and Machine Learning (ML) are of the form

[1]This function is also commonly called a **penalty function**. It can also be thought of, from a Bayesian perspective, as the log of a prior probability, and the loss function can be thought of as a log likelihood function.

[2]This function is also commonly called the **cost or error function**.

[3]ROLF that adds a regulator function (resp., doesn't add) is sometimes called **Explicit (resp., Implicit) regularization**.

$$\mathcal{L}(w) = \sum_{\sigma=1}^{nsam} \widehat{\mathcal{L}}(\widehat{y}_\sigma(x_\sigma, w), y_\sigma) \tag{83.1}$$

where the sum is over $nsam$ samples. A common, more specific type of $\mathcal{L}(w)$ is the **mean square error** which is given by

$$\mathcal{L}(w) = \frac{1}{nsam} \sum_{\sigma=1}^{nsam} \parallel \widehat{y}_\sigma(x_\sigma, w) - y_\sigma) \parallel_2 \tag{83.2}$$

This loss function is convex, so it has a minimum:

$$\begin{cases} w^* = \underset{w}{\operatorname{argmin}}\, \mathcal{L}(w) \\ Loss = \min_w \mathcal{L}(w) = \mathcal{L}(w^*) \end{cases} \tag{83.3}$$

In ROLF, we add a regulator $\mathcal{R}(w)$ to $\mathcal{L}(w)$:

$$\mathcal{L}^+(w) = \mathcal{L}(w) + \mathcal{R}(w) \tag{83.4}$$

83.1 L^p norm ROLF

See C.6 for the definition of L^p norms. Let

$$\mathcal{L}^+(w) = \mathcal{L}(w) + \mathcal{R}(w) \tag{83.5}$$

Then, for $\lambda, \lambda_1, \lambda_2 > 0$,

- L^1 norm ROLF (called LASSO or Basis Pursuit)

$$\mathcal{R}(w) = \lambda \parallel w \parallel_1 \tag{83.6}$$

- L^2 norm ROLF (called Ridge or Tikhonov Regression) (note that the L^2 norm is squared)

$$\mathcal{R}(w) = \lambda \parallel w \parallel_2^2 \tag{83.7}$$

- $L^1 + L_2$ norm ROLF (called Elastic Net)

$$\mathcal{R}(w) = \lambda_1 \parallel w \parallel_1 + \lambda_2 \parallel w \parallel_2^2 \tag{83.8}$$

83.1.1 L^1 norm ROLF can lead to sparsity

The obvious way to induce sparsity in the minimum w^* is to use the L^0 norm of w as regulator. However, calculating $\parallel w \parallel_0$ is hard (NP-hard, in fact). In this section, we will show that using the L^1 norm of w as regulator can also induce sparsity (not as well as the L^0 norm, but still significant.)

Figure 83.1: Pictorial explanation of why L^1 norm ROLF can lead to sparsity and L^2 norm ROLF can help avert it. Sometimes you want sparsity and sometimes you don't. c in this figure is some fixed constant.

Assume $\mathcal{L}(w)$ is linear in w.[4] For example,

$$\mathcal{L}(w) = Xw - y \tag{83.9}$$

where $w \in \mathbb{R}^n$, $X \in \mathbb{R}^{nsam \times n}$, and $y \in \mathbb{R}^{nsam}$. $nsam$ is often the number of samples. That's why we name it that way. The set $A = \{w : Xw - y = c\}$ for some fixed constant c might be empty, or contain a single point or a line or a plane, or a hyperplane. Let $dof = n - nsam$ be the degrees of freedom in w. Barring some exceptional cases, if $dof \leq 0$, we expect A to be empty. If $dof = 1$, we expect A to trace out a line, if $dof = 2$, we expect A to trace out a plane, and so forth.

In Fig.83.1, we imagine what would happen if $w \in \mathbb{R}^2$, $X \in \mathbb{R}^{1\times 2}$ and $y \in \mathbb{R}$ so A traces a line, represented in green. In Fig.83.1 (a), we imagine that $\mathcal{R}(w) = \| w \|_1$ and in Fig.83.1 (b), that $\mathcal{R}(w) = \| w \|_2^2$. For points w such that $\mathcal{R}(w)$ has a well defined gradient, minimization of

$$\mathcal{L}^+(w) = \mathcal{L}(w) + \mathcal{R}(w) \tag{83.10}$$

requires that the w-gradients of $\mathcal{L}$ and $\mathcal{R}$ be equal in magnitude but opposite in direction.

$$\nabla_w \mathcal{L} = -\nabla_w \mathcal{R} \tag{83.11}$$

But when $\mathcal{R}(w)$ is the L^1 norm, $\nabla_w \mathcal{R}$ is not defined along the w_1 and w_2 axes. To avoid this, we can approximate the diamond contour $\| w \|_1 = c$ by rounding out its corners by an infinitesimal amount. If the green line had slope of -1, there would be a diamond contour $\| w \|_1 = c$ that would coincide with the green line along the

[4] What we say here about sparsity also applies in some cases when $\mathcal{L}(w)$ is not linear in w. For example, it applies sometimes when $\mathcal{L}(w)$ is quadratic in w as occurs in Least Squares.

segment connecting points $(0, c)$ and $(c, 0)$. However, this is an exceptional case. Usually, the slope of the green line is not ± 1. In that case, the only way for $\mathcal{L}$ and $\mathcal{R}$ to have opposite gradients is if the diamond contour and the green line kiss at one of the (rounded) corners of the diamond contour. Call that kissing point w^*. Note from Fig.83.1 (a) that kissing point $w^* = (0, c)$ would be sparse. In going from the Fig.83.1 (a) to Fig.83.1 (b), we have replaced the diamond contour $\| w \|_1 = c$ by a circular contour $\| w \|_2 = c$, but we have kept the same green line. Note that with the circular contour, the kissing point w^* is no longer sparse; both of its components are non-zero.

The moral of Fig.83.1 is that L^1 norm ROLF can lead to sparsity and L^2 norm ROLF can help avert it. Sometimes we want the vector of weights w to be sparse so as to give a succinct description. At other times, we want solutions in set A to be spread out over many dimensions instead of being sparse and clumped together in a small number of dimensions.

83.1.2 L^2 norm ROLF for Least Squares

As in Chapter D on Linear Regression, suppose $w \in \mathbb{R}^n$, $X \in \mathbb{R}^{nsam \times n}$, and $y \in \mathbb{R}^{nsam}$. Let

$$\mathcal{L}^+(w) = \mathcal{L}(w) + \mathcal{R}(w) \tag{83.12}$$

where

$$\mathcal{L}(w) = \frac{1}{n}(Xw - y)^T(Xw - y) \tag{83.13}$$

and

$$\mathcal{R}(w) = \lambda \| w \|_2^2 \ . \tag{83.14}$$

If we vary the vector w by an infinitesimal amount δw^T, we get

$$0 = \delta \mathcal{L}^+(w) = \delta w^T \left[\frac{2X^T}{n}(Xw - y) + 2\lambda w \right] \tag{83.15}$$

Hence

$$X^T(Xw - y) + \lambda n w = 0 \tag{83.16}$$

$$(X^T X + \lambda n) w = X^T y \tag{83.17}$$

$$\begin{aligned} w &= (X^T X + \lambda n I)^{-1} X^T y && (83.18) \\ &= \frac{1}{\lambda n}(I + \frac{X^T X}{\lambda n})^{-1} X^T y && (83.19) \end{aligned}$$

Note that for $\lambda = 0$, the minimum w^* of $\mathcal{L}^+(w)$ is

$$w^* = (X^T X)^{-1} X^T y \;\; \text{(valid for } \lambda = 0) \tag{83.20}$$

When $\lambda >> 1$, we can express w^* as a Taylor expansion in w. Recall that if $|\epsilon| < 1$,

$$\frac{1}{1-\epsilon} = 1 + \epsilon + \epsilon^2 + \ldots \tag{83.21}$$

Define $n_0 = 1/\lambda$, and assume that $\frac{n_0}{n}|X^T X| < 1$. Then

$$\begin{aligned} w^* &= \frac{n_0}{n} \sum_{i=0}^{\infty} (-\frac{n_0 X^T X}{n})^i X^T y & (83.22) \\ &= -\sum_{i=0}^{\infty} \left(-\frac{n_0 X^T X}{n} \right)^{i+1} (X^T X)^{-1} X^T y \;\; \text{(valid for } \lambda = \frac{1}{n_0} >> 1) & (83.23) \end{aligned}$$

Truncating this series is itself a kind of regularization.

83.2 Proximal functions

For $v \in \mathbb{R}^n$, $w \in \mathbb{W} \subset \mathbb{R}^n$ and $\alpha > 0$, we define the **proximal function** $w^{prox} : \mathbb{R}^n \to \mathbb{W}$ by

$$\boxed{w^{prox}(v; \alpha\mathcal{L}, \mathbb{W}) = \underset{w \in \mathbb{W}}{\text{argmin}} \underbrace{\left(\mathcal{L}(w) + \underbrace{\frac{1}{2\alpha} \parallel w - v \parallel^2}_{\mathcal{R}(w,v)} \right)}_{\mathcal{L}^+(w,v)}} \tag{83.24}$$

$w^{prox}(v)$ can be viewed as a projection of $v \in \mathbb{R}^n$ onto w in the subspace $\mathbb{W} \subset \mathbb{R}^n$. Henceforth we assume $\mathbb{W} = \mathbb{R}^n$, so the projected vector v and its projection w are in the same vector space $\mathbb{R}^n$.

See Fig.83.2 for a numerical example of a proximal function $w^{prox} : \mathbb{R} \to \mathbb{R}$.

Next, we will discuss an analytical rather a numerical example of proximal functions. We begin by defining the shrinking function[5] $sh_0 : \mathbb{R} \to \mathbb{R}$ and its inverse $sh_0^{-1} : \mathbb{R} \to \mathbb{R}$ for any $\alpha > 0$:

$$sh_0(v; \alpha) = (v - \alpha)\mathbb{1}(v > \alpha) + (v + \alpha)\mathbb{1}(v < -\alpha) \tag{83.25}$$

$$sh_0^{-1}(w; \alpha) = (w + \alpha)\mathbb{1}(w > 0) + (w - \alpha)\mathbb{1}(w < 0) \tag{83.26}$$

Figure 83.2: Example of a proximal function. x-axis is w and $v = 5$. Green curve: $\mathcal{L}(w) = |w|$. Red curve: $\mathcal{R}(w, 5) = \frac{1}{2}(w-5)^2$, Blue curve: $\mathcal{L}^+(w, 5) = \mathcal{L}(w)+\mathcal{R}(w, 5)$. Note that we are adding 2 convex functions so the minimum of the sum is somewhere in between the minima of the two summands. $w^{prox}(5) = \underset{w}{\text{argmin}}\, \mathcal{L}^+(w, 5) = 4$.

Figure 83.3: Plot of the functions $sh_0(w;\alpha)$ and $sh_0^{-1}(w;\alpha)$.

Fig.83.3 shows a plot of sh_0 and sh_0^{-1}.

Claim 127 *For* $\mathbb{W} = \mathbb{R}$ *and* $w, v \in \mathbb{R}$, *if*

$$\mathcal{L}(w) = |w| \ , \tag{83.27}$$

then

[5]We call it a **shrinking function** because it shrinks a neighborhood-of-zero to zero. Another common name for this function is a **soft-threshold function**, because it makes the transition from negative to positive y axis values occur over the interval of $x \in [-\alpha, \alpha]$ instead of $x \in [0, 0]$.

$$w^{prox}(v) = sh_0(v;\alpha) \tag{83.28}$$

and

$$\mathcal{L}^+(w^{prox}(v), v) = |sh_0(v;\alpha)| + \frac{\alpha}{2} \tag{83.29}$$

proof:

$$0 = \frac{d\mathcal{L}^+}{dw} = \alpha[\mathbb{1}(w > 0) - \mathbb{1}(w < 0)] + (w - v) \tag{83.30}$$

So

$$\begin{aligned} v &= w + \alpha[\mathbb{1}(w > 0) - \mathbb{1}(w < 0)] && (83.31)\\ &= (w + \alpha)\mathbb{1}(w > 0) + (w - \alpha)\mathbb{1}(w < 0) && (83.32)\\ &= sh_0^{-1}(w;\alpha) && (83.33) \end{aligned}$$

Hence

$$w^{prox}(v;\alpha) = sh_0(v;\alpha) \tag{83.34}$$

Note that

$$|w - v| = |\alpha| \tag{83.35}$$

so

$$\begin{aligned} \mathcal{L}^+(w_{prox}(v), v) &= |w^{prox}(v;\alpha)| + \frac{\alpha}{2} && (83.36)\\ &= |sh_0(v;\alpha)| + \frac{\alpha}{2} && (83.37) \end{aligned}$$

QED

83.3 Proximal ROLF

Proximal functions can be used to do ROLF as follows. For $w, v \in \mathbb{R}^n$ and $\alpha > 0$, let

$$\mathcal{R}(w, v) = \frac{1}{2\alpha} \parallel w - v \parallel_2^2 \tag{83.38}$$

and[6]

$$\mathcal{L}^+(w) = \mathcal{L}(w) + \mathcal{R}(w, w^*) \ , \tag{83.39}$$

[6]Note that $\mathcal{L}$ here could be the mean square error plus an L^p norm. A loss function plus a regulator function gives a new loss function to which a new regulator may be added.

where w^* is the w-minimum of both $\mathcal{L}(w)$ and $\mathcal{R}(w, w^*)$:

$$w^* = \underset{w}{\operatorname{argmin}}\, \mathcal{L}(w) = \underset{w}{\operatorname{argmin}}\, \mathcal{R}(w, w^*) \tag{83.40}$$

Hence,

$$w^* = \underset{w}{\operatorname{argmin}}\, [\mathcal{L}(w) + \mathcal{R}(w, w^*)] \ . \tag{83.41}$$

Now assume the sequence of points $\{w_k\}_{k=0}^{\infty}$ satisfies $w_k \to w^*$ as $k \to \infty$. Then

$$w_{k+1} = \underbrace{\underset{w}{\operatorname{argmin}} \left(\mathcal{L}(w) + \frac{1}{2\alpha} \parallel w - w_k \parallel_2^2 \right)}_{w^{prox}(w_k)} \tag{83.42}$$

If we differentiate the argument of argmin() to find its minimum, we find

$$0 = \alpha \underbrace{\nabla_w \mathcal{L}(w_{k+1})}_{\approx \nabla_w \mathcal{L}(w_k)} + w_{k+1} - w_k \tag{83.43}$$

Thus, the following 3 recursion relations[7] can be used to calculate w_k:

$$\boxed{w_{k+1} = w_k - \alpha \nabla_w \mathcal{L}(w_k)} \tag{83.44a}$$

$$\boxed{w_{k+1} = w^{prox}(w_k)} \tag{83.44b}$$

$$\boxed{w_{k+1} = w^{prox}(w_k - \alpha \nabla_w \mathcal{L}(w_k))} \tag{83.44c}$$

Eq.(83.44a) is the familiar recursion relation for gradient descent (See Chapter 38). Eq.(83.44c) combines gradient descent and a proximal projection, so it is expected to converge faster than simple gradient descent.

83.4 Unobserved Nodes of a bnet

Nodes of a bnet for which the CPT is unknown are called unobseved nodes. In this book, Unobserved (a.k.a. hidden, latent) nodes are indicated in a bnet by enclosing their label in a dashed circle. For example, $\overset{\frown}{\underset{\smile}{(u)}}$. Alternatively, they are indicated by using dashed arrows for all arrows emanating from the unobserved node.

Unobserved nodes (UN) represent what are called **latent random variables** in Statistics.

[7]Some people use a sequence $\alpha_k \in \mathbb{R}_+$ instead of the constant $\alpha > 0$. This is called an **adaptive step size** and can yield faster convergence rates.

UN make an appearance in many places throughout this book. For example, they are essential to the methods of Kalman Filtering (see Chapter 47) and Hidden Markov Model (see Chapter 40).

This being a book about bnets and a chapter about ROLF, we would like to stress that UN can be viewed as a very natural and powerful way of doing (implicit) ROLF when using bnets.

Chapter 84

Reinforcement Learning (RL)

Figure 84.1: Axes for episode time and episode number.

I based this chapter on the following references. Refs.[21][42]

In RL, we consider an "agent" or robot that is learning.

Let $T \in \mathbb{Z}_{>0}$ be the duration time of an **episode** of learning. If $T = \infty$, we say that the episode has an infinite time horizon. A learning episode will evolve towards the right, for times $t = 0, 1, \ldots, T-1$. We will consider multiple learning episodes. The episode number will evolve from top to bottom. This is illustrated in Fig.84.1.

Let $\underline{s}_t \in val(\underline{s})$ for $t \in \mathbb{Z}_{[0,T-1]}$ be random variables that record the **state** of the agent at various times t.

Let $\underline{a}_t \in val(\underline{a})$ for $t \in \mathbb{Z}_{[0,T-1]}$ be random variables that record the **action** of the agent at various times t.

Figure 84.2: State-Action-Reward dynamical bnet

Let $\underline{\theta}_t \in val(\underline{\theta})$ for $t \in \mathbb{Z}_{[0,T-1]}$ be random variables that record the **policy parameters** at various times t.

For $\underline{X} \in \{\underline{s}, \underline{a}, \underline{\theta}\}$, define $\underline{X}$ followed by a dot to be the vector

$$\underline{X}. = [\underline{X}_0, \underline{X}_1, \ldots, \underline{X}_{T-1}] \ . \tag{84.1}$$

Also let

$$\underline{X}_{\geq t} = [\underline{X}_t, \underline{X}_{t+1}, \ldots, \underline{X}_{T-1}] \ . \tag{84.2}$$

Fig.84.2 shows the basic State-Action-Reward bnet for an agent that is learning. The TPMs, printed in blue, for the bnet Fig.84.2, are as follows.

$$P(a_t|s_t, \theta_t) = \text{given.} \tag{84.3}$$

$P(a_t|s_t, \theta_t)$ is called a **policy with parameter θ_t.**

$$P(s_t|s_{t-1}, a_{t-1}) = \text{given.} \tag{84.4}$$

$P(s_t|s_{t-1}, a_{t-1})$ is called the **TPM of the model**. $P(s_t|s_{t-1}, a_{t-1})$ reduces to $P(s_0)$ when $t = 0$.

$$P(r_t|s_t, a_t) = \delta(r_t, r(s_t, a_t))) \ . \tag{84.5}$$

$r : val(\underline{s}) \times val(\underline{a}) \to \mathbb{R}$ is a given **one-time reward function**.

Note that

$$P(s., a.|\theta.) = \prod_{t=0}^{T-1} \{P(s_t|s_{t-1}, a_{t-1})P(a_t|s_t, \theta_t)\} \ . \tag{84.6}$$

Define the **all times reward** Σ by

$$\Sigma(s., a.) = \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t) \ . \tag{84.7}$$

Here $0 < \gamma < 1$. γ, called the **discount rate**, is included to assure convergence of Σ when $T \to \infty$. If $r(s_t, a_t) < K$ for all t, then $\Sigma < K\frac{1}{1-\gamma}$.

Define the **objective (i.e., goal) function** $E\Sigma(\theta.)$ by

$$E\Sigma(\theta.) = E_{\underline{s}.,\underline{a}.|\theta.}\Sigma(\underline{s}., \underline{a}.) = \sum_{s.,a.} P(s., a.|\theta.)\Sigma(s., a.) \tag{84.8}$$

The goal of RL is to maximize the objective function over its parameters $\theta.$. The parameters $\theta^*.$ that maximize the objective function are the optimum strategy:

$$\theta.^* = \underset{\theta.}{\operatorname{argmax}}\, E\Sigma(\theta.) \tag{84.9}$$

Define a **future reward** for times $\geq t$ as:

$$\Sigma_{\geq t}((s_{t'}, a_{t'})_{t' \geq t}) = \sum_{t'=t}^{T-1} \gamma^{t'-t} r(s_{t'}, a_{t'}) \tag{84.10}$$

Define the following **expected conditional future rewards** (rewards for times $\geq t$, conditioned on certain quantities having given values):

$$v_t = v(s_t, a_t; \theta.) = E_{\underline{s}.,\underline{a}.|s_t,a_t,\theta.}[\Sigma_{\geq t}] \tag{84.11}$$

$$V_t = V(s_t; \theta.) = E_{\underline{s}.,\underline{a}.|s_t,\theta.}[\Sigma_{\geq t}] = E_{\underline{a}_t|s_t,\theta.}[v(s_t, \underline{a}_t; \theta.)] \tag{84.12}$$

v is usually called Q in the literature. We will refer to Q as v in order to follow a convention wherein an $\underline{a}_t$-average changes a lower case letter to an upper case one.

We will sometimes write $v(s_t, a_t)$ instead of $v(s_t, a_t; \theta.)$.

Since $E\Sigma_{\geq t}$ only depends on $\theta_{\geq t}$, $v(s_t, a_t; \theta.) = v(s_t, a_t; \theta_{\geq t})$, and $V(s_t; \theta.) = V(s_t; \theta_{\geq t})$.

Note that the objective function $E\Sigma$ can be expressed in terms of v_0 by averaging over its unaveraged parameters:

$$E\Sigma(\theta.) = E_{\underline{s}_0,\underline{a}_0|\theta_0} v(\underline{s}_0, \underline{a}_0; \theta.) \tag{84.13}$$

Define a **one-time reward** and an **expected conditional one-time reward** as:

$$r_t = r(s_t, a_t) \tag{84.14}$$

$$R_t = R(s_t; \theta_t) = E_{\underline{a}_t|s_t,\theta_t}[r(s_t, \underline{a}_t)] \quad . \tag{84.15}$$

Note that

$$\Sigma_{\geq t} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots + \gamma^{T-1-t} r_{t+(T-1-t)} \tag{84.16}$$

$$= r_t + \gamma \Sigma_{\geq t+1}; . \tag{84.17}$$

If we take $E_{\underline{s}_\cdot,\underline{a}_\cdot|s_t,a_t,\theta_\cdot}[\cdot]$ of both sides of Eq.(84.17), we get

$$v_t = r_t + \gamma E_{\underline{s}_{t+1},\underline{a}_{t+1}|\theta_\cdot}[v_{t+1}] \ . \tag{84.18}$$

If we take $E_{\underline{s}_\cdot,\underline{a}_\cdot|s_t,\theta_\cdot}[\cdot]$ of both sides of Eq.(84.17), we get

$$V_t = R_t + \gamma E_{\underline{s}_{t+1}|\theta_\cdot}[V_{t+1}] \ . \tag{84.19}$$

Note that

$$\begin{aligned} \Delta r_t &= r_t - R_t & (84.20)\\ &= r_t - (V_t - \gamma E_{\underline{s}_{t+1}|\theta_\cdot}[V_{t+1}]) & (84.21)\\ &= r_t + \gamma E_{\underline{s}_{t+1}|\theta_\cdot}[V_{t+1}] - V_t \ . & (84.22) \end{aligned}$$

Define

$$\Delta v_t = v_t - V_t \ . \tag{84.23}$$

Note that

$$\Delta v_t = \Delta r_t \ . \tag{84.24}$$

Next, we will discuss 3 RL bnets

- exact RL bnet (exact, assumes policy is known)
- Actor-Critic RL bnet (approximate, assumes policy is known)
- Q function learning RL bnet (approximate, assumes policy is NOT known)

84.1 Exact RL bnet

An exact RL bnet is given by Fig.84.3.

Fig.84.3 is the same as Fig.84.2 but with more nodes added in order to optimize the policy parameters. The TPMs, printed in blue, for the nodes not already discussed in connection to bnet Fig.84.2, are as follows.

$$P(\theta_t|\theta_\cdot) = \delta(\theta_t, (\theta_\cdot)_t) \tag{84.25}$$

$$\forall(s_t,a_t): \quad P(v_t(s_t,a_t)|r_t, v_{t+1}(\cdot), \theta_\cdot) = \delta(v_t(s_t,a_t), r_t + \gamma E_{\underline{s}_{t+1},\underline{a}_{t+1}|\theta_\cdot}[v_{t+1}]) \tag{84.26}$$

$$P(\theta_\cdot'|\theta_\cdot, v_0(\cdot)) = \delta(\theta'_\cdot, \theta_\cdot + \alpha\partial_{\theta_\cdot} \underbrace{E_{\underline{s}_0,\underline{a}_0|\theta_0} v(\underline{s}_0, \underline{a}_0; \theta_\cdot)}_{E\Sigma(\theta_\cdot)}) \tag{84.27}$$

Figure 84.3: Exact RL bnet. $v_t(\cdot)$ means the array $[v_t(s_t, a_t)]_{\forall s_t, a_t}$ The following arrows are implicit: for all t, arrow from $\underline{\theta}. \to \underline{v}_t(\cdot)$. We did not draw those arrows so as not to clutter the diagram.

$\alpha > 0$ is called the **learning rate**. This method of improving $\theta.$ is called gradient ascent.

Concerning the gradient of the objective function, note that

$$\begin{aligned}
\partial_{\theta_t} E\Sigma(\theta.) &= \sum_{s.,a.} \partial_{\theta_t} P(s.,a.|\theta.)\Sigma(s.,a.) && (84.28)\\
&= \sum_{s.,a.} P(s.,a.|\theta.)\partial_{\theta_t} \ln P(s.,a.|\theta.)\Sigma(s.,a.) && (84.29)\\
&= E_{\underline{s}.,\underline{a}.|\theta.}\left\{\partial_{\theta_t} \ln P(a_t|s_t,\theta_t)\Sigma(s.,a.)\right\} \ . && (84.30)
\end{aligned}$$

If we run the agent $nsam(\vec{s}_t)$ times and obtain samples $s_t[i], a_t[i]$ for all t and for $i = 0, 1, \ldots, nsam(\vec{s}_t) - 1$, we can express this gradient as follows:

$$\partial_{\theta_t} E\Sigma(\theta.) \approx \frac{1}{nsam(\vec{s}_t)} \sum_i \sum_{t=0}^{T-1} \partial_{\theta_t} \ln P(a_t[i] \mid s_t[i], \theta_t) r(s_t[i], a_t[i]) \ . \tag{84.31}$$

The exact RL bnet Fig.84.3 is difficult to use to calculate the optimum parameters θ^*.. The problem is that $\underline{s}_t$ propagates towards the future and the $\underline{v}_t(\cdot)$ propagates towards the past, so we don't have a Markov Chain with a chain link for each t (i.e., a dynamical bnet) in the episode time direction. Hence, people have come up with approximate RL bnets that are doubly dynamical (i.e., dynamical along the episode time and episode number axes.) We discuss some of those approximate RL bnets next.

84.2 Actor-Critic RL bnet

For the actor-critic RL bnet, we approximate Eq.(84.31) by

$$\partial_{\theta_t} E\Sigma(\theta.) \approx \frac{1}{nsam(\vec{s})} \sum_i \sum_{t=0}^{T-1} \underbrace{\partial_{\theta_t} \ln P(a_t[i] \mid s_t[i], \theta_t)}_{Actor} \underbrace{\Delta r_t(s_t[i], a_t[i])}_{Critic} \tag{84.32}$$

The actor-critic RL bnet is given by Fig.84.4. This bnet is approximate and assumes that the policy is known. The TPMs, printed in blue, for this bnet, are as follows.

$$P(\theta_t) = \text{given} \tag{84.33}$$

$$P(s_t[i] \mid s_{t-1}[i], a_{t-1}[i]) = \text{ given} \tag{84.34}$$

Figure 84.4: Actor-Critic RL bnet.

$$P(a_t[i] \mid s_t[i], \theta_t) = \text{ given} \tag{84.35}$$

$$P(r_t[i] \mid s_t[i], a_t[i]) = \delta(r_t[i], r(s_t[i], a_t[i])) \tag{84.36}$$

$r : val(\underline{s}) \times val(\underline{a}) \to \mathbb{R}$ is given.

$$P(\Delta v_t[i] \mid s_t[i], a_t[i], s_{t+1}[i]) = \delta(\Delta v_t[i], r(s_t[i], a_t[i]) + \gamma \widehat{V}(s_{t+1}[i]; \phi') - \widehat{V}(s_t[i]); \phi) \ . \tag{84.37}$$

$$P(\theta'.) = \delta(\theta'., \theta_t + \alpha \partial_{\theta_t} \sum_i \ln P(a_t[i] \mid s_t[i], \theta_t) \Delta v_t[i]) \tag{84.38}$$

$\widehat{V}(s_t[i]); \phi)$ is obtained by curve fitting (see Chapter 38) using samples $(s_t[i], a_t[i])$ $\forall t, i$ with

$$y[i] = \sum_{t'=t}^{T} r(s_{t'}[i], a_{t'}[i]) \tag{84.39}$$

and

$$\widehat{y}[i] = \widehat{V}(s_t[i]; \phi) \ . \tag{84.40}$$

Figure 84.5: Q function learning RL bnet.

Eq.(84.39) is an approximation because $(s_{t'}, a_{t'})_{t'>t}$ are averaged over in the exact expression for $V(s_t)$. $\widehat{V}(s_{t+1}[i]); \phi')$ is obtained in the same way as $\widehat{V}(s_t[i]); \phi)$ but with t replaced by $t+1$ and ϕ by ϕ'.

84.3 Q function learning RL bnet

The Q-function learning RL bnet is given by Fig.84.5. This bnet is approximate and assumes that the policy is NOT known. The TPMs, printed in blue, for this bnet, are as follows. (Remember that $Q = v$).

$$P(s_t|s_{t-1}, a_{t-1}) = \text{ given} \tag{84.41}$$

$$P(a_t|s_t, v_t(\cdot)) = \delta(a_t, \operatorname*{argmax}_a v_t(s_t, a)) \tag{84.42}$$

$$P(r_t|s_t, a_t) = \delta(r_t, r(s_t, a_t)) \tag{84.43}$$

$r : val(\underline{s}) \times val(\underline{a}) \to \mathbb{R}$ is given.

$$\begin{aligned} \forall (s_t, a_t): \quad & P(v_t(s_t, a_t)|v_{t-1}(\cdot)) = \\ & = \quad \delta(v_t(s_t, a_t), r(s_t, a_t) + \gamma \max_a E_{\underline{s}_{t+1}|s_t, a_t} v_{t-1}(\underline{s}_{t+1}, a)) \end{aligned} \tag{84.44}$$

This value for $v_t(s_t, a_t)$ approximates $v_t = r_t + \gamma E_{\underline{s}_{t+1}, \underline{a}_{t+1}} v_{t+1}$.

Some people use the bnet of Fig.84.6) instead of Fig.84.5 and replace Eq.(84.44) by

Figure 84.6: Q function learning RL bnet. Same as Fig.84.5 but with new arrow passing s_t to Q_{t-1}.

$$\begin{aligned} \forall(s_t, a_t): \quad & P(v_t(s_t, a_t)|s_{t+1}, v_{t-1}(\cdot)) = \\ & = \quad \delta(v_t(s_t, a_t), r(s_t, a_t) + \gamma \max_a v_{t-1}(s_{t+1}, a)) \ . \end{aligned} \tag{84.45}$$

Chapter 85

Reliability Box Diagrams and Fault Tree Diagrams

This chapter is based on Refs.[69] and [100].

In this chapter, we assume that reader is familiar with Boolean Algebra. See Chapter C for a quick review of what we recommend that you know about Boolean Algebra to fully appreciate this chapter.

Figure 85.1: Example of rbox diagram.

Figure 85.2: An ftree diagram equivalent to Fig.85.1. It represents $e = (\phi_1 \wedge \phi_3) \vee (\phi_2 \wedge \phi_3)$.

Complicated devices with a large number of components such as airplanes or rockets can fail in many ways. If their performance depends on some components

Figure 85.3: How to map an rbox diagram to a bnet.

Figure 85.4: bnet corresponding the rbox diagram Fig.85.1.

working in series and one of the components in the series fails, this may lead to catastrophic failure. To avert such disasters, engineers use equivalent components connected in parallel instead of in series, thus providing multiple backup systems. They analyze the device to find its weak points and add backup capabilities there. They also estimate the average time to failure for the device.

The two most popular diagrams for finding the failure modes and their rates for large complicated devices are

- rbox diagrams = Reliability Box diagrams. See Fig.85.1 for an example.
- ftree diagrams = Fault Tree Diagrams. See Fig.85.2 for an example.

In an ftree diagram, several nodes might stand for the same component of a physical

device. In an rbox diagram, on the other hand, each node represents a distinct component in a device. Hence, rbox diagrams resemble the device they are addressing whereas ftree diagrams don't. Henceforth, we will refer to this desirable property as **physical resemblance**.

As we will show below with an example, it is pretty straightforward to translate an rbox to an ftree diagram. Going the other way, translating an ftree to an rbox diagram is much more difficult.

Next we will define a new kind of bnet that we will call a failure bnet that has physical resemblance. Then we will describe a simple method of translating (i.e., mapping) any rbox diagram to a failure bnet. Then we will show how a failure bnet can be used to do all the calculations that are normally done with an rbox or an ftree diagram. In that sense, failure bnets seem to afford all the benefits of both ftree and rbox diagrams.

A **failure bnet** contains nodes of 5 types, labeled $\underline{b}$, $\underline{e}$, $\underline{x}_i$, $\underline{\phi}_i$, and $\underline{A}_i$. All nodes have only two possible states $S = Success = 0$, $F = Failure = 1$.

1. The bnet has a beginning node labeled $\underline{b}$ which is always set to success. The $\underline{b}$ node and the $\underline{\phi}_i$ nodes are the only root nodes of the bnet.

2. The bnet has a single leaf node, the end node, labeled $\underline{e}$. $\underline{e}$ is fixed. In rbox diagrams, $\underline{e} = S$ whereas in ftree diagrams, $\underline{e} = F$.

3. $\underline{x}^{nx} = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{nx-1})$. $\underline{x}_i \in \{S, F\}$ for all i.

   Suppose $\underline{x}_i$ has parents $\underline{\phi}_i$ and $\underline{a}^{na} = (\underline{a}_0, \underline{a}_1, \ldots \underline{a}_{na-1})$. Then the TPM of node $\underline{x}_i$ is defined to be

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee \vee_{i=0}^{na-1} a_i) \tag{85.1}$$

4. For each node $\underline{x}_i$, the bnet has a "performance" root node $\underline{\phi}_i \in \{0, 1\}$ with an arrow pointing from it to $\underline{x}_i$ (i.e, $\underline{\phi}_i \rightarrow \underline{x}_i$). For all i,

$$P(\phi_i) = \epsilon_i \delta(\phi_i, F) + \overline{\epsilon}_i \delta(\phi_i, S) \ . \tag{85.2}$$

   ϵ_i is the failure probability and $\overline{\epsilon}_i = 1 - \epsilon_i$ is the success probability. We name the failure probability ϵ_i because it is normally very small. It is usually set to $1 - e^{-\lambda_i t} \approx \lambda_i t$ when $\lambda_i t << 1$, where λ_i is the failure rate for node $\underline{x}_i$ and t stands for time. The rblock literature usually calls $\overline{\epsilon_i} = R_i$ the **reliability** of node $\underline{x}_i$, and $\epsilon_i = (1 - R_i) = F_i$ its **unreliability**.

5. The nodes $\underline{A}_i \in \{0, 1\}$ are simply AND gates. If $\underline{A}_i$ has inputs $\underline{y}^{ny} = (\underline{y}_0, \underline{y}_1, \ldots, \underline{y}_{ny-1})$, then the TPM of $\underline{A}_i$ is

$$P(A_i|y^{ny}) = \delta(A_i, \wedge_{i=0}^{ny-1} y_i) \ . \tag{85.3}$$

An instance (instantiation) of a bnet is the bnet with all nodes set to a specific state. A **realizable instance (r-instance)** of a bnet is one which has non-zero probability.

Fig.85.3 shows how to translate any rbox diagram to a failure bnet. To illustrate this procedure, we translated the rbox diagram Fig.85.1 into the failure bnet Fig.85.4.

For the failure bnet Fig.85.4, one has:

$$\begin{array}{l} P(b) = \mathbb{1}(b = 0) \\ P(x_1|\phi_1, b) = \mathbb{1}(x_1 = \phi_1 \vee b) \\ P(x_2|\phi_2, x_1) = \mathbb{1}(x_2 = \phi_2 \vee x_1) \\ P(x_3|\phi_3, b) = \mathbb{1}(x_3 = \phi_3 \vee b) \\ P(A|x_2, x_3)e = \mathbb{1}(x_2 \wedge x_3) \\ P(e|A) = \mathbb{1}(e = A) \end{array} \ . \tag{85.4}$$

Therefore, all r-instances of this bnet must satisfy

$$e = (\phi_1 \vee \phi_2) \wedge \phi_3 \tag{85.5}$$

$$= (\phi_1 \wedge \phi_3) \vee (\phi_2 \wedge \phi_3) \ . \tag{85.6}$$

Eq.(85.6) proves that Fig.85.2 is indeed a representation of Fig.85.1.

Next, we consider r-instances of this bnet for two cases: $e = S$ and $e = F$.

- **rblock analysis:** $e = S = 0$**.**
  Table 85.1 shows the probability of all possible r-instances that end in success for the failure bnet Fig.85.4. (These r-instances are the main focus of rblock analysis). The first 4 of those probabilities (those with $\phi_3 = 0$) sum to $\overline{\epsilon}_3$ so the sum $P(e = S)$ of all 5 is

  $$P(e = S) = \overline{\epsilon}_3 + \overline{\epsilon}_1\overline{\epsilon}_2\epsilon_3 \ , \tag{85.7}$$

  or, expressing it in reliability language in which $\overline{\epsilon} = R$,

  $$P(e = S) = R_3 + R_1R_2\overline{R}_3 \ . \tag{85.8}$$

- **ftree analysis:** $e = F = 1$**.**
  Table 85.2 shows the probability of all possible r-instances that end in failure for the failure bnet Fig.85.4. (These r-instances are the main focus of ftree analysis). If we set $\epsilon_i = \epsilon$ and $\overline{\epsilon}_i \approx 1$ for $i = 1, 2, 3$, then the first two of

instance	probability
	$\overline{\epsilon}_1\epsilon_2\overline{\epsilon}_3$
	$\epsilon_1\overline{\epsilon}_2\overline{\epsilon}_3$
	$\epsilon_1\epsilon_2\overline{\epsilon}_3$
	$\overline{\epsilon}_1\overline{\epsilon}_2\overline{\epsilon}_3$
	$\overline{\epsilon}_1\overline{\epsilon}_2\epsilon_3$

Table 85.1: Probabilities of all possible r-instances with $e = S = 0$ for failure bnet Fig.85.4.

instance	probability
	$\overline{\epsilon}_1\epsilon_2\epsilon_3$
	$\epsilon_1\overline{\epsilon}_2\epsilon_3$
	$\epsilon_1\epsilon_2\epsilon_3$

Table 85.2: Probabilities of all possible r-instances with $e = F = 1$ for the failure bnet Fig.85.4.

those r-instances have probabilities of $order(\epsilon^2)$ and the third has probability of $order(\epsilon^3)$. The two lowest order ($order(\epsilon^2)$) r-instances are called the "minimal cut sets" of the ftree. We will have more to say about minimal cut sets later on. For now, just note from Eq.(85.6) that the ftree Fig.85.2 is just the result of joining together with ORs two expressions, one for each of the two minimal cut sets.

More general $\underline{x}_i$.

Failure bnets can actually accommodate $\underline{x}_i$ nodes of a more general kind than what we first stipulated. Here are some possibilities:

For any $a^n \in \{0,1\}^n$, let

$$len(a^n) = \sum_i a_i \tag{85.9}$$

- **OR gate**

$$
\begin{aligned}
P(x_i|\phi_i, a^{na}) &= \delta(x_i, \phi_i \vee \vee_j a_j) && (85.10)\\
&= \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) > 0)) && (85.11)
\end{aligned}
$$

- **AND gate**

$$
\begin{aligned}
P(x_i|\phi_i, a^{na}) &= \delta(x_i, \phi_i \vee \wedge_j a_j) && (85.12)\\
&= \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) = na)) && (85.13)
\end{aligned}
$$

- **Fail if least K failures (less than K successes)**

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) \geq K)) \tag{85.14}$$

- **Fail if less than K failures (at least K successes)**

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) < K)) \tag{85.15}$$

- **Fail if exactly one failure**

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee \mathbb{1}(len(a^{na}) = 1)) \tag{85.16}$$

  This equals an XOR (exclusive OR) gate when $na = 2$.

- **General gate**
  $f : \{0,1\}^{na} \to \{0,1\}$

$$P(x_i|\phi_i, a^{na}) = \delta(x_i, \phi_i \vee f(a^{na})) \tag{85.17}$$

85.1 Minimal Cut Sets

Suppose $x \in \{0,1\}$ and $f : \{0,1\} \to \{0,1\}$. Then by direct evaluation, we see that

$$f(x) = [\overline{x} f(0)] \vee [x f(1)] \ . \tag{85.18}$$

Let

$$
\begin{aligned}
&!x = 1 - x, \\
&!^0 x = x, \\
&!^1 x = !x
\end{aligned}
\tag{85.19}
$$

Then Eq.(85.18) can be rewritten as

$$
f(x) = \vee_{a \in \{0,1\}} [(!^{\overline{a}} x) f(a)] \ . \tag{85.20}
$$

Now suppose $x^n \in \{0,1\}^n$ and $f : \{0,1\}^n \to \{0,1\}$. Eq.(85.20) generalizes to

$$
f(x^n) = \vee_{a^n \in \{0,1\}^n} [\prod_i (!^{\overline{a}_i} x_i) f(a^n)] \ . \tag{85.21}
$$

Eq.(85.21) is called an **ors-of-ands** normal form expansion. There is also an **ands-of-ors** normal form expansion obtained by swapping multiplication and $\vee$ in Eq.(85.21), but we won't need it here.

A **cut set** is a set of ϕ_i's such that if they are all equal to F, then $e = F$ for all the r-instances. A **minimal cut set** is a cut set such that there are no larger cut sets that contain it. From the failure bnet, we can always find a function $f : \{0,1\}^{nx} \to \{0,1\}$ such that $e = f(\phi^{nx})$ for all the r-instances. We did that for our example failure bnet and obtained Eq.(85.6). We can then express $f(\phi^{nx})$ as an ors-of-ands expansion to find all the minimal cut sets. The ands terms in that ors-of-ands expansion each gives a different minimal cut set, after some simplification. The ors-of-ands expression is not unique and it may be necessary to simplify (using the Boolean Algebra identities given in Chapter C) to remove those redundancies.

Chapter 86

Restricted Boltzmann Machines

In what follows, we will abbreviate "restricted Boltzmann machine" by rebo.

Let
$v \in \{0,1\}^{nv}$
$h \in \{0,1\}^{nh}$
$b \in \mathbb{R}^{nv}$ (mnemonic, v and b sound the same)
$a \in \mathbb{R}^{nh}$
$W^{v|h} \in \mathbb{R}^{nv \times nh}$
Energy:

$$E(v,h) = -(b^T v + a^T h + v^T W^{v|h} h) \tag{86.1}$$

Boltzmann distribution:

$$P(v,h) = \frac{e^{-E(v,h)}}{Z} \tag{86.2}$$

Partition function:

$$Z = \sum_{v,h} e^{-E(v,h)} = Z(a,b,W^{v|h}) \tag{86.3}$$

$$\begin{aligned} P(v|h) &= \frac{e^{b^T v + a^T h + v^T W^{v|h} h}}{\sum_v e^{b^T v + a^T h + v^T W^{v|h} h}} && (86.4) \\ &= \frac{e^{b^T v + v^T W^{v|h} h}}{\sum_v e^{b^T v + v^T W^{v|h} h}} && (86.5) \\ &= \prod_i \frac{e^{v_i (b_i + \sum_j W_{i,j}^{v|h} h_j)}}{\sum_{v_i=0,1} e^{v_i (b_i + \sum_j W_{i,j}^{v|h} h_j)}} && (86.6) \\ &= \prod_i P(v_i|h) && (86.7) \end{aligned}$$

$$P(v_i|h) = \frac{e^{v_i (b_i + \sum_j W_{i,j}^{v|h} h_j)}}{Z_i(h)} \tag{86.8}$$

Figure 86.1: bnet for a Restricted Boltzmann Machine (rebo) with $nv = 3$

Eq.(86.8) implies that a rebo can be represented by the bnet Fig.86.1.
Let

$$x_i = b_i + \sum_j W_{ij}^{v|h} h_j \ . \tag{86.9}$$

Then

$$\begin{aligned} P(v_i = 1|h) &= \frac{e^{x_i}}{1 + e^{x_i}} && (86.10) \\ &= \frac{1}{1 + e^{-x_i}} && (86.11) \\ &= \text{smoid}(x_i) \ . && (86.12) \end{aligned}$$

One could also expand the node $\underline{h}$ in Fig.86.1 into nh nodes. But note that $P(h) \neq \prod_j P(h_j)$ so there would be arrows among the h_j nodes.

Note that the rebo bnet is a special case of Naive Bayes (See Chapter 67) with $v_i, h_i \in \{0, 1\}$ and specific $P(h)$ and $P(v_i|h)$ node matrices.

Chapter 87

ROC curves

This chapter is based on Ref.[177].

ROC stands for **Receiver Operating Characteristic**. ROC curves are used in binary classification (BC).

To do BC, we are given the value $x \in \mathbb{R}$ for an individual. From this, we want to decide whether that individual has $a = 0$ or $a = 1$. The decision will depend on the value of a **threshold parameter** $\tau \in \mathbb{R}$.

$$\underline{x} \longleftarrow \underline{a}$$

Figure 87.1: bnet for BC.

Fig.87.1 shows the bnet used for BC.

Figure 87.2: x-distribution for two hypotheses $a = 0, 1$.

Fig.87.2 is a plot of $P(x|a)$, i.e., the TPM for node $\underline{x}$ of the bnet in Fig.87.1.

Whereas a is binary, x is continuous. But we can replace x by a binary variable

$$b = \mathbb{1}(x > \tau) \ . \tag{87.1}$$

$P(b|a)$ for $b, a \in \{0, 1\}$ is called the **confusion matrix** or **contingency table** for BC. The confusion matrix can be calculated from the TPM $P(x|a)$. Fig.87.3 illustrates

		Actual Value (a)	
		0	1
Predicted Value (x)	0	TNR	FNR
	1	FPR	TPR

Figure 87.3: The confusion matrix $P(b|a)$ for BC.

the confusion matrix $P(b|a)$ for BC. In that figure, the rates R are defined as follows.[1]

- **True Negative Rate (TNR)**

$$R_{0|0}(\tau) = P(x < \tau | a = 0) = \int_{x<\tau} dx\ P(x|a = 0) \tag{87.2}$$

- **False Positive Rate (FPR)**

$$R_{1|0}(\tau) = 1 - R_{0|0}(\tau) \tag{87.3}$$

$$= P(x > \tau | a = 0) = \int_{x>\tau} dx\ P(x|a = 0) \tag{87.4}$$

In Hypothesis Testing, $R_{1|0}$ is called the **p-value that $\underline{x} > \tau$ assuming curve** 0 **is the null hypothesis**.

- **False Negative Rate (FNR)**

$$R_{0|1}(\tau) = P(x < \tau | a = 1) = \int_{x<\tau} dx\ P(x|a = 1) \tag{87.5}$$

In Hypothesis Testing, $R_{0|1}$ is called the **p-value that $\underline{x} < \tau$ assuming curve** 1 **is the null hypothesis**.

- **True Positive Rate (TPR)**

$$R_{1|1}(\tau) = 1 - R_{0|1}(\tau) \tag{87.6}$$

$$= P(x > \tau | a = 1) = \int_{x>\tau} dx\ P(x|a = 1) \tag{87.7}$$

The **Receiver Operating Characteristic (ROC)** is a parametric plot with $X = R_{1|0}(\tau)$ and $Y = R_{1|1}(\tau)$, where $\tau \in \mathbb{R}$. The **Area Under the Curve (AUC)** is the area under the ROC. Fig.87.4 shows an example of a ROC and its AUC.

Fig.87.5 shows situations that give AUC=.5 (random classifier), AUC=.85, and AUC=1 (perfect classifier). It's also possible to get an $AUC \in [0, 0.5]$, but we will ignore those models because they are useless for BC.

[1] I find the notation $x|a$ where $x, a \in \{0, 1\}$ much clearer than $\alpha\beta$ where $\alpha = T, F$ and $\beta = N, P$. Note that $\alpha = \mathbb{1}(x = a)$ and $\beta = x$, if we identify $0 = F = N$ and $1 = T = P$.

Figure 87.4: Example of ROC. Green shaded area is the AUC of the ROC.

Figure 87.5: ROC curves for 3 different separations between the 0 and 1 x-distributions.

Note that

$$
\begin{aligned}
AUC &= \int_{x=0}^{1} d\tau \;\; R_{1|1}(\tau)\frac{dR_{1|0}(\tau)}{d\tau} && (87.8)\\
&= \int_{\infty}^{-\infty} d\tau \;\; \left\{\int_{-\infty}^{\infty} dx \;\; \mathbb{1}(x>\tau)P(x|a=1)\right\}(-1)P(x=\tau|a=0) && (87.9)\\
&= \int_{-\infty}^{\infty} dx' \;\; \int_{-\infty}^{\infty} dx \;\; \mathbb{1}(x>x')P(x|a=1)P(x'|a=0)\;. && (87.10)
\end{aligned}
$$

See Fig.87.6 for an example of False Positive and False Negative predictions.

Figure 87.6: Example of False Positive and False Negative predictions.

87.1 Terminology Table Adapted from Wikipedia Ref.[177]

Let $N_{x|a}$ be numbers (counts) so that

$$P(x|a) = \frac{N_{x|a}}{\sum_{x'} N_{x'|a}} \tag{87.11}$$

for all $x, a \in \{0, 1\}$.

condition positive (P): $N_{|1} = \sum_x N_{x|1}$, the number of real positive cases in the data

condition negative (N): $N_{|0} = \sum_x N_{x|0}$, the number of real negative cases in the data

true positive (TP): $N_{1|1}$, hit

true negative (TN): $N_{0|0}$, correct rejection

false positive (FP): $N_{1|0}$, false alarm, type I error or overestimation

false negative (FN): $N_{0|1}$, miss, type II error or underestimation

sensitivity, recall, hit rate, or true positive rate (TPR):

$$TPR = R_{1|1} = \frac{N_{1|1}}{N_{|1}} = \frac{N_{1|1}}{N_{1|1} + N_{0|1}} = 1 - R_{0|1} \tag{87.12}$$

specificity, selectivity or true negative rate (TNR):

$$TNR = R_{0|0} = \frac{N_{0|0}}{N_{|0}} = \frac{N_{0|0}}{N_{0|0} + N_{1|0}} = 1 - R_{1|0} \tag{87.13}$$

precision or positive predictive value (PPV):

$$PPV = \frac{N_{1|1}}{N_{1|1} + N_{1|0}} = 1 - FDR \tag{87.14}$$

negative predictive value (NPV):

$$NPV = \frac{N_{0|0}}{N_{0|0} + N_{0|1}} = 1 - FOR \tag{87.15}$$

miss rate or false negative rate (FNR):

$$FNR = R_{0|1} = \frac{N_{0|1}}{N_{|1}} = \frac{N_{0|1}}{N_{0|1} + N_{1|1}} = 1 - R_{1|1} \tag{87.16}$$

fall-out or false positive rate (FPR):

$$FPR = R_{1|0} = \frac{N_{1|0}}{N_{|0}} = \frac{N_{1|0}}{N_{1|0} + N_{0|0}} = 1 - R_{0|0} \tag{87.17}$$

false discovery rate (FDR):

$$FDR = \frac{N_{1|0}}{N_{1|0} + N_{1|1}} = 1 - PPV \tag{87.18}$$

false omission rate (FOR):

$$FOR = \frac{N_{0|1}}{N_{0|1} + N_{0|0}} = 1 - NPV \tag{87.19}$$

accuracy (ACC):

$$ACC = \frac{N_{1|1} + N_{0|0}}{N_{|1} + N_{|0}} = \frac{N_{1|1} + N_{0|0}}{N_{1|1} + N_{0|0} + N_{1|0} + N_{0|1}} \tag{87.20}$$

balanced accuracy (BA):

$$BA = \frac{R_{1|1} + R_{0|0}}{2} \tag{87.21}$$

F1 score is the harmonic mean of precision and sensitivity:

$$F_1 = 2 \times \frac{PPV \times R_{1|1}}{PPV + R_{1|1}} = \frac{2N_{1|1}}{2N_{1|1} + N_{1|0} + N_{0|1}} \tag{87.22}$$

Chapter 88

Scoring the Nodes of a Learned Bnet

Chapter 95 discusses how to learn a bnet from data. Many algorithms for doing this require scoring how well a particular bnet fits the data. This chapter is an introduction to such scoring.

Normally, each node of a bnet is scored separately, and then those node scores are summed to get the bnet score.

In this chapter, scores are defined so that a higher score means a better fit. By taking the negative of such a score, one can always get a score such that a lower score means a better fit.

There are 2 main types of bnet scores: Maximum Likelihood (ML) scores, and Shannon Information Theory (SIT) scores. ML scores consist of the log of a maximum likelihood function $P(\vec{x}|\theta)$ for i.i.d. samples $\vec{x} = (x^\sigma)_{\sigma=0,1,\ldots,nsam-1}$, where $x^\sigma \sim P_{\underline{x}|\theta}(x|\theta)$:

$$\begin{aligned}
\text{ML-score} &= \ln(P(\vec{x}|\theta)) && (88.1)\\
&= \ln \prod_\sigma P(x^\sigma \mid \theta) && (88.2)\\
&= \sum_\sigma \ln P(x^\sigma \mid \theta) && (88.3)\\
&\approx nsam \sum_x P(x|\theta) \ln P(x|\theta) && (88.4)\\
&= -nsamH(P_{\underline{x}|\theta}) \ , && (88.5)
\end{aligned}$$

and SIT scores consist of a negative entropy:

$$\text{Info-score} = -H(P_{\underline{x}|\theta}) \ . \qquad (88.6)$$

Thus, up to a factor of $nsam$, they are the same thing. Maximizing a log likelihood function for i.i.d. samples or minimizing the corresponding entropy, are the same thing, and they both yield a good estimate of the hidden parameters θ.

88.1 Probability Distributions and Special Functions

While writing this chapter, I briefly consulted the following Wikipedia articles about the definitions and properties of certain probability distributions and special functions.

- Categorical Distribution, Ref.[120]
- Multinomial Distribution, Ref.[166]
- Dirichlet Distribution, Ref.[131]
- Multivariate Normal Distribution, Ref.[168]
- Beta function, Ref.[114]
- Multinomial Coefficients, Ref.[167]
- Gamma Function Ref.[139]

Here are a few results from those Wikipedia articles that we will use later on in this chapter.

Below, we will abbreviate $q_+ = \sum_i q_i$, and $q. = (q_0, q_1, \ldots, q_{nq-1})$ for various quantities q

Gamma function. If $n > 0$ is an integer,

$$\Gamma(n+1) = n! \tag{88.7}$$

The **multivariate Beta function** is defined by

$$B(\alpha.) = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\alpha_+)} \tag{88.8}$$

where $\alpha_k > 0$ for all k.

The **multinomial coefficient** is defined by

$$C(N.) = \frac{N_+!}{\prod_k N_k!} \tag{88.9}$$

where N_k are non-negative integers.

The **inverse of the multinomial coefficient** will be denoted by

$$CI(N.) = \frac{1}{C(N.)} = \frac{\prod_k N_k!}{N_+!} \tag{88.10}$$

The **Categorical Distribution** is defined by

$$Cat(x;\pi.) = \pi_x = \prod_k \pi_k^{\mathbb{1}(k=x)} \tag{88.11}$$

for $k, x \in val(\underline{x})$, where $\pi.$ is a probability dist.(i.e., $\pi_k \geq 0$ for all k, and $\pi_+ = 1$).

The **Multinomial Distribution** is defined by

$$Mul(N.;\pi., N) = C(N.) \ , \tag{88.12}$$

where N_k is a non-negative integer for all k, $N_+ = N$, and $\pi.$ is a probability dist. Mul() satisfies:

$$E[\underline{N_k}] = N\pi_k \ . \tag{88.13}$$

The **Dirichlet Distribution** is defined by

$$Dir(\pi.;\alpha.) = \frac{1}{B(\alpha.)} \prod_k \pi_k^{\alpha_k - 1} \tag{88.14}$$

where $\alpha_k > 0$ for all k, and $\pi.$ is a probability dist. The $\alpha.$ are called **concentration parameters** or **hyperparameters**. Dir() satisfies:

$$E[\underline{\pi_k}] = \frac{N_k}{N_+} \ . \tag{88.15}$$

Dir() is **conjugate prior** of Mul()

Note that

$$Mul(N.;\pi., N)Dir(\pi.;\alpha.) = \mathcal{K}(N., \alpha.)Dir(\pi.; N. + \alpha.) \ , \tag{88.16}$$

where

$$\mathcal{K}(N., \alpha.) = \frac{B(N. + \alpha.)}{CI(N.)B(\alpha.)} \ . \tag{88.17}$$

Dir() is replaceable by a Mul() for large concentration parameters

Note that if N_k is a positive integer and $\alpha_k = N_k + 1$ for all k, then

$$\begin{aligned} Dir(\pi.; \alpha_k = N_k + 1) &= C(N.) \prod_k \pi_k^{N_k} & (88.18) \\ &= Mul(N.;\pi., N_+) \ . & (88.19) \end{aligned}$$

88.2 Single node with no parents

In this section, we consider a learned bnet consisting of a single node with no parents. We will consider arbitrary learned bnets in the next section. But we start with this simplified case so as to reduce the number of indices in most quantities from 3 to 1. All the results that we derive in this section will be used in the next section after adding the extra indices. This way, we will avoid carrying the extra indices throughout the intermediate steps of many derivations.

For state $k \in \{0, 1, \ldots, nk-1\}$ of a single node $\underline{x}$, let
$\underline{N}_k$= current count number (an integer, data)
$\underline{\pi}$.= a probability dist, the TPM for the node
$\underline{\alpha}_k$= prior count number

$$\underline{N}. \longleftarrow \underline{\pi.} \longleftarrow \underline{\alpha.}$$

Figure 88.1: For a bnet consisting of a single node with no parents, this is a Markov chain of current counts ($\underline{N}$.), TPM ($\underline{\pi.}$), and prior counts ($\underline{\alpha.}$) .

Consider the Markov chain bnet of Fig.88.1 with the following TPMs, printed in blue.

$$P(N.|\pi.) = Mul(N.; \pi., N_+) \tag{88.20}$$

$$P(\pi.|\alpha.) = Dir(\pi.; \alpha.) \tag{88.21}$$

It follows that

$$\begin{aligned} P(N., \pi.|\alpha.) &= P(N.|\pi.)P(\pi.|\alpha.) && (88.22)\\ &= Mul(N.; \pi., N_+)Dir(\pi.; \alpha.) && (88.23)\\ &= \mathcal{K}(N., \alpha.)Dir(\pi.; N. + \alpha.) \ . && (88.24) \end{aligned}$$

From Eq.(88.15) for the expected value of Dir(), we get

$$\widehat{\pi}. = E[\underline{\pi}.] = \frac{N. + \alpha.}{N_+ + \alpha_+} \ . \tag{88.25}$$

Integrating both sides of Eq.(88.24) over π., we find that

$$P(N.|\alpha.) = \mathcal{K}(N., \alpha.) \ . \tag{88.26}$$

If $N_k >> 1$ for all k, then the Dir() in Eq.(88.24) can be replaced by a Mul()

$$P(N., \pi.|\alpha.) \approx \mathcal{K}(N., \alpha.) Mul(N. + \alpha.; \pi., N_+ + \alpha_+) \ . \tag{88.27}$$

Therefore,

$$\begin{aligned} P(N.|\pi., \alpha.) &= \frac{P(N., \pi.|\alpha.)}{P(N.|\alpha.)} & (88.28) \\ &= Mul(N. + \alpha.; \pi., N_+ + \alpha_+) \ . & (88.29) \end{aligned}$$

Claim 128

$$\begin{aligned} \ln P(N.|\widehat{\pi}., \alpha.) &= -(N_+ + \alpha_+) H\left(\frac{N. + \alpha.}{N_+ + \alpha_+}\right) + \ln C(N. + \alpha.) & (88.30) \\ &> -(N_+ + \alpha_+) H\left(\frac{N. + \alpha.}{N_+ + \alpha_+}\right) - \frac{1}{2}(nk - 1) \ln N_+ & (88.31) \end{aligned}$$

proof:

$$\begin{aligned} \ln P(N.|\widehat{\pi}., \alpha.) &= \sum_k (N_k + \alpha_k) \ln \widehat{\pi}_k + \ln C(N. + \alpha.) & (88.32) \\ &= \sum_k (N_k + \alpha_k) \ln \frac{N_k + \alpha_k}{N_+ + \alpha_+} + \ln C(N. + \alpha.) & (88.33) \\ &= -(N_+ + \alpha_+) H\left(\frac{N. + \alpha.}{N_+ + \alpha_+}\right) + \ln C(N. + \alpha.) & (88.34) \end{aligned}$$

Recall Stirling's approximation of a factorial, valid for large integers n:

$$\ln n! \approx (n + \frac{1}{2}) \ln n - n \ . \tag{88.35}$$

Assume $N_k >> 1$ for all k. Applying Stirling's approximation to all factorials in $C(N)$, we get

$$\begin{aligned} \ln C(N.) &\approx (N_+ + \frac{1}{2}) \ln N_+ - N_+ - \sum_k \left[(N_k + \frac{1}{2}) \ln N_k - N_k\right] & (88.36) \\ &= (N_+ + \frac{1}{2}) \ln N_+ - \sum_k (N_k + \frac{1}{2}) \ln N_k \ . & (88.37) \end{aligned}$$

Next assume that

$$N_k \approx \frac{N_+}{nk} \ . \tag{88.38}$$

Then

$$\ln C(N.) = (N_+ + \frac{1}{2}) \ln N_+ - nk(\frac{N_+}{nk} + \frac{1}{2})[\ln N_+ - \ln nk] \tag{88.39}$$

$$= -\frac{1}{2}(nk-1)\ln N_+ + (N_+ + \frac{nk}{2}) \ln nk \tag{88.40}$$

$$> -\frac{1}{2}(nk-1)\ln N_+ \ . \tag{88.41}$$

QED

88.3 Multiple nodes with any number of parents

In the previous section, we considered a bnet consisting of a single node with no parents, so we only needed a single index k for the states of the single node. In this section, we consider an arbitrary bnet with multiple nodes each of which may have multiple parents. Most of the results in the previous section are valid for the general case if we make the following replacements: $\pi. \to \pi^i_{\cdot|\mu}$ $N. \to N^i_{\cdot,\mu}$ $\alpha. \to \alpha^i_{\cdot,\mu}$. Upon this replacement, Fig.88.1 becomes Fig.88.2. The TPMs, printed in blue, of the new Markov chain, are as follows:

$$\underline{N^i_{\cdot,\mu}} \longleftarrow \underline{\pi^i_{\cdot|\mu}} \longleftarrow \underline{\alpha^i_{\cdot,\mu}}$$

Figure 88.2: Generalization of Fig.88.1. For a bnet with multiple nodes each of which may have multiple parents, this is a Markov chain of current counts ($\underline{N^i_{\cdot,\mu}}$), TPM ($\underline{\pi^i_{\cdot|\mu}}$), and prior counts ($\underline{\alpha^i_{\cdot,\mu}}$) .

$$P(N^i_{\cdot,\mu}|\pi^i_{\cdot|\mu}) = Mul(N^i_{\cdot,\mu}; \pi^i_{\cdot|\mu}, N^i_{+,\mu}) \tag{88.42}$$

$$P(\pi^i_{\cdot|\mu}|\alpha^i_{\cdot,\mu}) = Dir(\pi^i_{\cdot|\mu}; \alpha^i_{\cdot,\mu}) \tag{88.43}$$

In these TPMs,
$i \in val(\underline{i}) = \{0, 1, \ldots, ni-1\}$= node index
$\underline{x}. = (\underline{x}_i)_{i \in val(\underline{i})}$= the nodes of the learned bnet.
$k \in val(\underline{k}^i) = \{0, 1, \ldots, nk^i - 1\}$= states of node $\underline{x}_i$
$\mu \in val(\underline{\mu}^i) = \{0, 1, \ldots, n\mu^i - 1\}$= states of parents of node $\underline{x}_i$.

In the previous section, we assumed a single node ($ni = 1$) with no parents ($n\mu^0 = 1$) so that we could drop the i, μ indices. In this section, we eliminate that restriction.

It is convenient to define the magnitude of a bnet B to equal the sum over nodes of the number of free parameters in each TPM:

$$|B| = \sum_i (nk^i - 1)n\mu^i \ . \tag{88.44}$$

Suppose that we are given $nsam$ samples $\vec{\underline{x}}_i = (\underline{x}_i^\sigma)_{\sigma=0,1,\ldots,nsam-1}$ of our learned bnet. The count numbers $N^i_{k,\mu}$ are defined in terms of those samples as follows:

$$N^i_{k,\mu} = \sum_\sigma \mathbb{1}(\underline{x}_i^\sigma = k, pa(\underline{x}_i^\sigma) = \mu) \ . \tag{88.45}$$

It is also convenient to defined count number ratios

$$N^i_{k|\mu} = \frac{N^i_{k,\mu}}{N^i_{+,\mu}} \ . \tag{88.46}$$

Note that $N^i_{k,\mu}$ is a positive integer whereas $N^i_{k|\mu} \in [0,1]$.

Let's denote the components of the TPMs by $\pi^i_{k|\mu}$:

$$\pi^i_{k|\mu} = P(\underline{x}_i = k \mid pa(\underline{x}_i) = \mu) \approx N^i_{k|\mu} \ . \tag{88.47}$$

The rest of this section lists equations that we obtained from the previous section, by adding the new indices i, μ:

$$\mathcal{K}(N^i_{\cdot,\mu}, \alpha^i_{\cdot,\mu}) = \frac{B(N^i_{\cdot,\mu} + \alpha^i_{\cdot,\mu})}{CI(N^i_{\cdot,\mu})B(\alpha^i_{\cdot,\mu})} \tag{88.48}$$

$$\widehat{\pi}^i_{k|\mu} = \frac{N^i_{k,\mu} + \alpha^i_{k,\mu}}{N^i_{+,\mu} + \alpha^i_{+,\mu}} \tag{88.49}$$

$$P(N^i_{\cdot,\mu}|\alpha^i_{\cdot,\mu}) = \mathcal{K}(N^i_{\cdot,\mu}, \alpha^i_{\cdot,\mu}) \tag{88.50}$$

$$P(N^i_{\cdot,\mu}|\pi^i_{\cdot|\mu}, \alpha^i_{\cdot,\mu}) \approx Mul(N^i_{\cdot,\mu} + \alpha^i_{\cdot,\mu}; \pi^i_{\cdot|\mu}, N^i_{+,\mu} + \alpha^i_{+,\mu}) \tag{88.51}$$

Claim 129

$$\ln P(N^i_{\cdot,\mu}|\widehat{\pi}^i_{\cdot|\mu}, \alpha^i_{\cdot,\mu}) = \sum_k (N^i_{k,\mu} + \alpha^i_{k,\mu}) \ln \left(\frac{N^i_{k,\mu} + \alpha^i_{k,\mu}}{N^i_{+,\mu} + \alpha^i_{+,\mu}} \right) + \ln C(N^i_{\cdot,\mu} + \alpha^i_{\cdot,\mu}) \tag{88.52}$$

$$> \sum_k (N^i_{k,\mu} + \alpha^i_{k,\mu}) \ln \left(\frac{N^i_{k,\mu} + \alpha^i_{k,\mu}}{N^i_{+,\mu} + \alpha^i_{+,\mu}} \right) - \frac{1}{2}(nk^i - 1) \ln N^i_{+,\mu} \tag{88.53}$$

88.4 Bayesian Scores

- Bayesian Information Criterion (BIC)

$$\text{BIC-score} = -\sum_i \sum_{k,\mu} N^i_{k,\mu} \ln\left(\frac{N^i_{k,\mu}}{N^i_{+,\mu}}\right) + \underbrace{\left[-\frac{|B|}{2}\ln N^+_{+,+}\right]}_{\sum_i \sum_\mu \ln C(N^i_{\cdot,\mu}) \text{ would be more accurate}} \tag{88.54}$$

$$\approx \sum_i \sum_\mu \ln P(N^i_{\cdot,\mu} | \widehat{\pi}^i_{\cdot|\mu}, \alpha^i_{\cdot,\mu} = 0) \tag{88.55}$$

- Bayesian Dirichlet (BD)

$$\text{BD-score} = \sum_i \sum_\mu \ln \frac{B(N^i_{\cdot,\mu} + \alpha^i_{\cdot,\mu})}{B(N^i_{\cdot,\mu})} \tag{88.56}$$

$$= \sum_i \sum_\mu \ln \left[CI(N^i_{\cdot,\mu}) P(N^i_{\cdot,\mu} | \alpha^i_{\cdot,\mu})\right] \tag{88.57}$$

- BD equivalent (BDe)

$$\text{BDe-score} = \text{BD-score}\left(\alpha^i_{k,\mu} = \alpha' N^i_{k,\mu}\right) , \tag{88.58}$$

where α' is a free parameter.

- BD equivalent unified (BDeu)

$$\text{BDeu-score} = \text{BD-score}\left(\alpha^i_{k,\mu} = \frac{\alpha'}{nk^i n\mu^i}\right) , \tag{88.59}$$

where α' is a free parameter. The BDeu score satisfies **score equivalence**; i.e., it is the same for all DAGs in an equivalence class of observational equivalent DAGs. See Chapter 71 for more information about observational equivalence.

88.5 Information Theoretic scores

- Maximum likelihood

$$\text{ML-score} = \sum_i \sum_{k,\mu} N^i_{k,\mu} \ln N^i_{k|\mu} \tag{88.60}$$

$$= -\sum_i H(\underline{k}^i | \underline{\mu}^i) , \tag{88.61}$$

where $P_{\underline{k}^i|\underline{\mu}^i}(k|\mu) = N^i_{k|\mu}$ and $P_{\underline{k}^i,\underline{\mu}^i}(k,\mu) = N^i_{k,\mu}$.

- Bayesian Information Criterion (BIC), a.k.a. Minimum Description Length (MDL)

$$\text{BIC-score} = \text{ML-score} - \frac{|B|}{2} \ln N^{+}_{+,+} \tag{88.62}$$

$$\approx \sum_{i} \sum_{k,\mu} N^{i}_{k,\mu} \ln \frac{N^{i}_{k|\mu}}{\sqrt{N^{+}_{+,+}}} \tag{88.63}$$

- Akaike Information Criterion (AIC)

$$\text{AIC-score} = \text{ML-score} - |B| \tag{88.64}$$

$$\approx \sum_{i} \sum_{k,\mu} N^{i}_{k,\mu} \left[\ln N^{i}_{k|\mu} - 1 \right] \tag{88.65}$$

Chapter 89

Selection Bias Removal

This chapter is based on Ref.[4].

Selection bias (SB) occurs when one samples from an atypical subset of a population, producing a biased dataset. Are such biased datasets useless? Not necessarily. It is possible to add auxiliary features to the biased dataset, and to sample those new features in an unbiased way, from the whole population. Then one can merge the original biased dataset with the auxiliary, unbiased one, to obtain an enhanced dataset. It is sometimes possible to do this so that the enhanced dataset is provably unbiased. It's like making horizontal the surface of a table that was not initially horizontal. The theory of Bayesian Networks and Causal Inference tells us WHEN this is possible, and HOW to do it when it is possible.

Consider the bnet Fig.89.1.

Figure 89.1: Bnet considered for selection bias (SB) removal. Arrows with question marks may or may not be present.

Let

$\underline{s} \in \{0, 1\}$ be the **switch node**. $\underline{s} = 1$ means there is SB in the parent nodes.

$\underline{x}$ = **class features**.[1]

$\underline{y}$ = **target feature**.

$\underline{A}$ = **auxiliary features**. This is a set of nodes that may contain arrows entering or exiting it, as indicated by the double arrows.

$\underline{E} = \{\underline{y}, \underline{x}\} \cup \underline{A}$ = **Enhanced feature set**.

[1] A feature is the same as a node in a bnet.

Σ = unbiased population of individuals σ

Σ_o = biased sub-population of individuals, $\Sigma_o \subset \Sigma$.

$OD = \{(\sigma_o, \underline{x}^{\sigma_o}, \underline{y}^{\sigma_o}, \underline{s}^{\sigma_o} = 1) : \sigma_o \in \Sigma_o\}$ = **Original Dataset**, dataset for $(\underline{x}, \underline{y})$ features with $\underline{s} = 1$. Gives empirical distribution $P(y|x, \underline{s} = 1)$. (Ref.[4] calls this dataset the **biased study**.)

$AD = \{(\sigma, \underline{x}^{\sigma}, \underline{A}^{\sigma}) : \sigma \in \Sigma\}$ = **Auxiliary Dataset**, dataset for $(\underline{x}, \underline{A})$ features. Gives empirical distribution $P(A|x)$. (Ref.[4] calls this dataset the **population level study**.)

$ED = \{(\sigma_o, \underline{x}^{\sigma_o}, \underline{A}^{\sigma_o}, \underline{y}^{\sigma_o}, \underline{s}^{\sigma_o} = 1) : \sigma_o \in \Sigma_o\}$ = **Enhanced Dataset**, dataset for $(\underline{x}, \underline{y}, \underline{A})$ features for $\underline{s} = 1$. Obtained by merging OD and AD. Gives empirical distribution $P(y|x, A, \underline{s} = 1)$.

89.1 Pre and Post Switch Nodes

This book uses 2 types of switch nodes: pre-switch nodes and post-switch nodes.

Pre-switch nodes are used in Chapter 105 entitled "Transportability of Causal Knowledge". Pre-switch nodes are root nodes. They are usually binary, and indicate whether the nodes pointed to belong to a set or not.

Most of the DAG literature, including Ref.[4], on which this chapter is based, define SB using a **post-switch node**. A post-switch node is a leaf node of the graph. Like pre-switch nodes, post-switch nodes are usually binary.

Figure 89.2: Common Cause and Effect for nodes $\underline{d}, \underline{y}$.

Note that in Potential Outcomes (PO) theory (see Chapter 77), pre-switch nodes such as $\underline{a}$ in Fig.89.2 are called **common cause (confounder, fork) nodes for nodes** $\underline{d}, \underline{y}$. Furthermore, post-switch nodes such as $\underline{b}$ in Fig.89.2 are called **common effect (selection bias (SB), collider) nodes for nodes** $\underline{d}, \underline{y}$. Hence, in PO theory, SB is indicated by a leaf node, just as we do in this chapter.

Note that **Simpson's paradox** (see Chapter 93) arises from an indirect effect caused by not conditioning on a confounder, whereas **Berkson's paradox** (see Chapter 7) arises from an indirect effect caused by conditioning on a collider.

It's possible to replace a pre-switch node by a post switch node, or vice versa, as follows. Suppose that we start with a bnet G_0 that is fully connected, and we add to it a switch node $\underline{s}$ that is a root node that points to all nodes of G_0. Call the

resulting bnet $\underline{s} \to G_0$. We can use Bayes rule to reverse the direction of the arrows emanating from $\underline{s}$ so that they enter node $\underline{s}$ rather than exit it. Call the resulting bnet $\underline{s} \leftarrow G_0$. In general, Bayes rule allows us to translate from $\underline{s} \to G_0$ to $\underline{s} \leftarrow G_0$, or in the opposite direction, without having to change the directions of any of the arrows in G_0. If G_0 is not fully connected, then going from $\underline{s} \to G_0$ to $\underline{s} \leftarrow G_0'$ will often require that G_0' have the same arrows in the same directions as G_0 plus some extra arrows new to G_0. Likewise, going from $\underline{s} \leftarrow G_0$ to $\underline{s} \to G_0''$ may require that G_0'' have the same arrows as G_0 plus some new arrows.

So far, we have been intentionally vague in specifying the graphs G_0' and G_0''. In Fig.89.3 we give a trick for determining possible candidates for graphs G_0' and G_0''. In Fig.89.3, we consider 3 panels going from left to right, depicting the cases where $\underline{s}$ has either 1,2 or 3 neighbors. The top graph $G_{post-sel}$, which has a post-switch node $\underline{s}$, is converted to a graph which is numerically equal to it, namely the bottom graph $G_{pre-sel}$, which has a pre-switch node $\underline{s}$. The magenta arrows represent any number of arrows exiting (but none entering) a node. If we start with a graph $s \leftarrow G_0$, we find the biggest subset X of the nodes of G_0 such that $\underline{s} \leftarrow X$ only has nodes exiting it (i.e., only magenta nodes). Then we add enough arrows to $\underline{s} \leftarrow X$ to make it a fully connected graph $\underline{s} \leftarrow X'$. Now we can reverse the incoming arrows to $\underline{s}$ and make them all outgoing and call the resulting graph $\underline{s} \to X'$.

Figure 89.3: Switching from a post-switch node to a pre-switch node.

Recall that in Chapter F, we made a distinction between a good CF bnet a bad CF bnet, and we pointed out that bad CF bnets are often useful as a numerical tool. Recall also from Chapter 71 that two bnets can be "observationally equivalent". That is what is happening here. We are faced with the choice of making switch nodes either leaf nodes or root nodes. Both choices lead to observationally equivalent bnets.

One of the two choices leads to a good CF bnet, and the other to a bad CF bnet. Both choices are numerically correct.

In this chapter, we will consider first a DAG $G_{post-sel}$ with a post-switch node, then we will transform that DAG to a numerically equal DAG $G_{pre-sel}$ with a pre-switch node. The latter DAG is more convenient for our needs. Why? Because in the SB theory that we present in this chapter, the $\underline{s} = 1$ appears as a condition in a conditional probability, as in $P(\cdots|\cdots, \underline{s} = 1)$. Such probabilities are represented in a more straightforward manner if arrows exit rather than enter node $\underline{s}$.

89.2 Removing SB from passive query $P(y|x)$

The **passive query** $Q = P(y|x, \underline{s} = 1)$ is **SB-recoverable** if it is independent of $\underline{s}$.

1. **Query** $P(y|x)$ **is SB-recoverable** with $\underline{A} = \emptyset$; SB can be removed by conditioning on $\underline{x}$.

   If $\underline{y} \perp \underline{s}|\underline{x}$, then
   $$P(y|x, \underline{s} = 1) = P(y|x) \ . \tag{89.1}$$

   For example,

(89.2a)

(89.2b)

(89.2c)

   The bnets on the left hand sides of Eqs.(89.2) satisfy $\underline{y} \perp \underline{s}|\underline{x}$.

2. **Query** $P(y|x)$ **is SB-recoverable** via $\underline{a}$; SB can be removed by conditioning on $\underline{x}$ and $\underline{a}$. Here $\underline{a}$ is a nonempty subset of $\underline{A}$

   Claim 130 *There exists* $\underline{a} \subset \underline{A}$ *such that* $\underline{y} \perp \underline{s}|(\underline{x}, \underline{a})$ *and* $\underline{a} \perp \underline{s}|\underline{x}$ *iff*
   $$P(y|x, \underline{s} = 1) = \sum_a \underbrace{P(y|x, a, \underline{s} = 1)}_{P(y|x,a)} \underbrace{P(a|x, \underline{s} = 1)}_{P(a|x)} = P(y|x) \tag{89.3}$$

$$\begin{tikzcd}
\underline{s}=1 \arrow[dr] & & \sum a \arrow[d] & \\
x \arrow[r] & y & x \arrow[r] \arrow[ur] & y
\end{tikzcd} = \begin{tikzcd} x \arrow[r] & y \end{tikzcd} \tag{89.4}$$

proof:

The $\Rightarrow$ part of this claim is obvious. For a proof of the $\Leftarrow$ part, see Ref.[4].
QED

For example,

$$\begin{tikzcd}
\underline{s} & \underline{a} \arrow[d] \\
\underline{x} \arrow[u] \arrow[r] \arrow[ur] & \underline{y}
\end{tikzcd} = \begin{tikzcd}
\underline{s} \arrow[d] & \underline{a} \arrow[d] \\
\underline{x} \arrow[r] \arrow[ur] & \underline{y}
\end{tikzcd} \tag{89.5a}$$

The bnets on the left hand sides of Eqs.(89.5) satisfy $\underline{y} \perp \underline{s}|(\underline{x}, \underline{a})$ and $\underline{a} \perp \underline{s}|\underline{x}$.

3. **Query $P(y|x)$ is not SB-recoverable**; SB cannot be removed.

   For example,

$$\begin{tikzcd}
\underline{s} & \\
\underline{x} \arrow[r] & \underline{y} \arrow[ul]
\end{tikzcd} = \begin{tikzcd}
\underline{s} \arrow[d] \arrow[dr] & \\
\underline{x} \arrow[r] & \underline{y}
\end{tikzcd} \tag{89.6}$$

$$\begin{tikzcd}
\underline{s} & \\
\underline{x} \arrow[u] \arrow[r] & \underline{y} \arrow[ul]
\end{tikzcd} = \begin{tikzcd}
\underline{s} \arrow[d] \arrow[dr] & \\
\underline{x} \arrow[r] & \underline{y}
\end{tikzcd} \tag{89.7}$$

$$\begin{tikzcd}
\underline{s} & \underline{a} \arrow[l] \arrow[dl] \\
\underline{x} \arrow[u] \arrow[r] & \underline{y} \\
\underline{w} \arrow[u] \arrow[ur] &
\end{tikzcd} = \begin{tikzcd}
\underline{s} \arrow[r] \arrow[d] & \underline{a} \arrow[dl] \\
\underline{x} \arrow[r] & \underline{y} \\
\underline{w} \arrow[u] \arrow[ur] &
\end{tikzcd} \tag{89.8}$$

89.3 Removing SB from active query $P(y|\mathcal{D}x)$

The **active query (i.e., do query)** $Q = P(y|\mathcal{D}\underline{x} = x, \underline{s} = 1)$ is

(a) **SB-recoverable** if it equals $P(y|\mathcal{D}\underline{x} = x)$,

(b) **do-identifiable** if it equals $P(y|x, \underline{s} = 1)$.

(c) both **SB-recoverable and do-identifiable** if it equals $P(y|x)$.

Examples

- $$\begin{tikzcd} \underline{s} & \underline{a} \arrow[l] \arrow[d] \arrow[dl] \\ \underline{x} \arrow[r] & \underline{y} \end{tikzcd} = \begin{tikzcd} \underline{s} \arrow[r] & \underline{a} \arrow[d] \arrow[dl] \\ \underline{x} \arrow[r] & \underline{y} \end{tikzcd} \qquad \begin{array}{l} \text{SB-recoverable: NO} \\ \text{do-identifiable: YES} \end{array} \tag{89.9}$$

  $Q = P(y|\mathcal{D}\underline{x} = x, \underline{s} = 1)$ is do-identifiable because the bnet contains no hidden variables. It's s-recoverable because the bnet on the left hand side of Eq.(89.9) satisfies $\underline{y} \perp \underline{s}|(\underline{x}, \underline{a})$ but Not $\underline{a} \perp \underline{s}|\underline{x}$.

- $$\begin{tikzcd} \underline{s} & \underline{a} \arrow[d] \\ \underline{x} \arrow[u] \arrow[ur] \arrow[r] & \underline{y} \end{tikzcd} = \begin{tikzcd} \underline{s} \arrow[d] & \underline{a} \arrow[d] \\ \underline{x} \arrow[ur] \arrow[r] & \underline{y} \end{tikzcd} \qquad \begin{array}{l} \text{SB-recoverable: YES} \\ \text{do-identifiable: NO} \end{array} \tag{89.10}$$

  Let
  V = set of nodes in graph
  $V^{<\underline{x}}$ = non-descendants of $\underline{x}$ (excluding $\underline{x}$)
  $V^{>\underline{x}}$ = descendants of $\underline{x}$ (excluding $\underline{x}$)
  $\underline{z}^{<\underline{x}} = \underline{z} \cap V^{<\underline{x}}$
  $\underline{z}^{>\underline{x}} = \underline{z} \cap V^{>\underline{x}}$
  Suppose $\underline{z} \cup \{\underline{x}, \underline{y}\} \subset E$ and $\underline{z} \subset \underline{A}$. We say $\underline{z}$ satisfies the **selection bias (SB) backdoor criterion** with respect to $(\underline{x}, \underline{y})$ if

1. all backdoor paths from $\underline{x}$ to $\underline{y}$ are blocked by conditioning on $\underline{z}^{<\underline{x}}$
2. $\underline{z}^{>\underline{x}} \perp \underline{y}|(\underline{x}, \underline{z}^{<\underline{x}})$
3. $\underline{y} \perp \underline{s}|(\underline{x}, \underline{z})$

Claim 131 *(SB Backdoor Adjustment Formula)*
If $\underline{z}$ satisfies the SB backdoor criterion relative to $(\underline{x}, \underline{y})$, then

$$P(y|\mathcal{D}\underline{x} = x, \underline{s} = 1) = \sum_z P(y|x, z)P(z) = P(y|x) \tag{89.11}$$

$$\begin{tikzcd} \underline{s} = 1 \arrow[dr] & & \sum z \arrow[d] & \\ \mathcal{D}\underline{x} = x \arrow[r] & y \arrow[r, phantom, "="] & x \arrow[r] & y \arrow[r, phantom, "="] & x \arrow[r] & y \end{tikzcd} \tag{89.12}$$

proof:

If z satisfies the SB backdoor criterion relative to $(\underline{x}, \underline{y})$, then $\underline{x}, \underline{y}, \underline{z}$ might have the following structure.

$$\begin{tikzcd}
\underline{s} & \underline{z}^{<\underline{x}} \arrow[l] \arrow[r] \arrow[d] \arrow[dl] & \underline{z}^{>\underline{x}} \arrow[ll, bend right] \arrow[dl] \\
\underline{x} \arrow[u] \arrow[r] & \underline{y} &
\end{tikzcd}
=
\begin{tikzcd}
\underline{s} \arrow[r] \arrow[d] & \underline{z}^{<\underline{x}} \arrow[r] \arrow[d] \arrow[dl] & \underline{z}^{>\underline{x}} \arrow[ll, bend left] \arrow[dl] \\
\underline{x} \arrow[r] & \underline{y} &
\end{tikzcd} \tag{89.13}$$

See Claim 60 for a proof of this claim for the special case Eq.(89.13).

QED

Chapter 90

Sentence Splitting with SentenceAx

The Openie6 (O6) software (at github repo Ref.[3]) splits complex or compound sentences into simple ones.[1] Sentence splitting is a necessary step when doing DAG extraction from text (DEFT) (See Chapter 14).

The O6 software is described by its creators in the paper Ref.[37], which we will henceforth refer to as the O6 paper.

My SentenceAx (Sax) software (at github repo Ref.[96]) is a complete re-write of the O6 software. Sax is 95% identical algorithmically to O6, but I have rewritten it in what I hope is a friendlier form.

Sax is a fine-tuning of the BERT model. What this means in the language of Bayesian Networks is that we use BERT as a prior probability. The BERT model is the encoder part of the vanilla Transformer network which we discuss in Chapter 104.

This chapter describes the technical aspects of Sax. Although this chapter can be read without reading the O6 paper, we highly recommend to our readers that they read the O6 paper also. Some parts of this chapter are taken almost verbatim from the O6 paper. Other parts try to fill-in gaps in the explanations provided by the O6 paper or to improve those explanations. Yet others parts describe small changes that we made to the O6 software, in an effort to improve its clarity.

In this chapter, we will use the Numpy-like tensor notation discussed in Section C.48. In particular, note that $[n] = [0 : n] = \{0, 1, \ldots, n-1\}$ and that $T^{[n],[m]}$ is an $n \times m$ matrix.

[1] Simple sentences are essentially the same as the triples (subject, relationship, object) which, when visualized as a directed or undirected graph, is called a "knowledge graph".

90.1 Preliminary Conventions

90.1.1 Tensor Notation

Our tensor notation is discussed in Section C.48. Here is a quick review of some of the more salient facts in that section on tensors. Below, we will often accompany an equation in tensor component notation with the equivalent matrix equation, in parenthesis.

We use Greek letters for tensor indices.

Let $\alpha \in [a]$, $\beta \in [b]$, $\gamma \in [c]$, $\delta \in [d]$, $\nu \in [n]$, $\Delta \in [D]$.

- **reshaping**

$$T^{\nu,\delta} \to T^{\Delta} \quad \left(T^{[n_{\underline{h}}],[d]} \to T^{[D]}\right) \tag{90.1}$$

$$T^{\Delta} \to T^{\nu,\delta} \quad \left(T^{[D]} \to T^{[n_{\underline{h}}],[d]}\right) \tag{90.2}$$

- **concatenation**

$$T^{[n]} = (T^0, T^1, \ldots, T^{n-1}) = (T^{\nu})_{\nu \in [n]} \tag{90.3}$$

- **Hadamard product (element-wise, entry-wise multiplication)**

$$T^{[n]} * S^{[n]} = (T^{\nu} S^{\nu})_{\nu \in [n]} \tag{90.4}$$

- **Matrix multiplication**

  $T^{[n]} = T^{[n],[1]}$ is a column vector.

$$(T^{[n]})^T S^{[n]} = \text{scalar} \tag{90.5}$$

$$T^{[a],[b]} S^{[b],[c]} = \left[\sum_{\beta \in [b]} T^{\alpha,\beta} S^{\beta,\gamma}\right]_{\alpha \in [a], \gamma \in [c]} \tag{90.6}$$

Most treatments of Transformer Networks (tranets), including the O6 paper and PyTorch, order the operations chronologically from left to right (L2R). So if A occurs before B, they write AB. This is contrary to what is done in Linear Algebra, where one orders the operations chronologically from right to left (R2L), and one writes BA. In Chapter 104 on tranets, we followed the Linear Algebra convention. In this chapter, we will follow the PyTorch convention, because Sax is written with PyTorch so it uses the PyTorch convention.

90.1.2 PyTorch conventions

- **Linear**

  Some pseudo-code

  ```
  lin = nn.Linear(b, a)
  y^{[n],[b]} = lin(x^{[n],[a]})
  ```

  In the L2R (left to right) convention followed by PyTorch

  $$x^{\nu,[a]} \to y^{\nu,[b]} = x^{\nu,[a]} W^{[a],[b]} \tag{90.7}$$

  for all $\nu \in [n]$. Alternatively, in the R2L convention followed in Linear Algebra,

  $$x^{[a],\nu} \to y^{[b],\nu} = W^{[b],[a]} x^{[a],\nu} \tag{90.8}$$

  Note that in PyTorch, the rightmost index of the input is the one that is summed over.

  The weights matrix $W^{[b],[a]}$ is learned by training.

- **Dropout**

  Some pseudo-code

  ```
  dropo = nn.Dropout(p_drop)
  y^{[n],[a]} = dropo(x^{[n],[a]})
  ```

  $$x^{\nu,[a]} \to y^{\nu,[a]} = x^{\nu,[a]} \widehat{W}_R^{[a],[a]} \quad \text{(in L2R)} \tag{90.9}$$

  $$x^{[a],\nu} \to y^{[a],\nu} = \widehat{W}_L^{[a],[a]} x^{[a],\nu} \quad \text{(in R2L)} \tag{90.10}$$

  for all $\nu \in [n]$.

  `Dropout` learns a weight matrix W just like `Linear`. But at the end of the training, `Dropout` flips a coin for each row of $W_L^{[a],[a]}$ (resp., column of $W_R^{[a],[a]}$), with

  $$P(Heads) = p_{drop}, \text{ and } P(Tails) = 1 - p_{drop} = p_{keep}.$$

  If the coin lands on T, it keeps that row of $W_L^{[a],[a]}$ (resp., column of $W_R^{[a],[a]}$), whereas if lands on H, it sets that row (resp., column) to zero. Then the matrix is divided by p_{keep}. The final matrix after all these operations is denoted by $\widehat{W}_L$ (resp., $\widehat{W}_R$).

- **Embedding**

  Some pseudo-code

```
emb = nn.Embedding(num_embeddings=L, embedding_dim=d)
Y^{[n1],[n2],[d]} = emb(λ^{[n1],[n2]})
```

In Sax, we use $L = 100$ and $d = 768$ (for BERT base). The d is the "hidden dimension" of BERT. The L could be as large as the vocab size of BERT, but since we consider only sentences with 100 words at most, we may set $L = 100$. L doesn't appear in the final answer because we sum over $\lambda \in [L]$.

Next, we explain in more detail the meaning of the tensors λ and Y.

Let

$L =$ number of embeddings

$d =$ embedding dimension

$\lambda \in [L]$, $\alpha \in [\ell]$, $\nu_1 \in [n_1]$, $\nu_2 \in [n_2]$

$\ell = \nu_1 \nu_2$.

Consider matrices Y, E, X such that

$$Y^{\delta,\alpha} = \sum_{\lambda} E^{\delta,\lambda} X^{\lambda,\alpha} \quad \left(Y^{[d],[\ell]} = E^{[d],[L]} X^{[L],[\ell]}\right) \tag{90.11}$$

Assume that matrix X has 1-hot columns

$$X^{\lambda,\alpha} = \delta(\lambda, \lambda(\alpha)) \tag{90.12}$$

where $\lambda() : [\ell] \to [L]$.

Hence,

$$Y^{\delta,\alpha} = E^{\delta,\lambda(\alpha)} \tag{90.13}$$

If we define

$$\Lambda^{\alpha} = \lambda(\alpha) \tag{90.14}$$

then `emb()` maps

$$\Lambda^{\alpha} \to Y^{\delta,\alpha} = E^{\delta,\lambda(\alpha)} \ \ (\Lambda^{[\ell]} \to Y^{[d],[\ell]}) \tag{90.15}$$

Now replace α by (ν_1, ν_2). `emb()` also maps

$$\Lambda^{\nu_1,\nu_2} \to Y^{\delta,\nu_1,\nu_2} = E^{\delta,\lambda(\nu_1,\nu_2)} \ \ (\Lambda^{[n_1],[n_2]} \to Y^{[d],[n_1],[n_2]}) \tag{90.16}$$

Actually, `emb()` orders the tensor indices of the output so that the δ index is on the right side rather than the left side of the input indices. Thus,

$$Y^{[n_1],[n_2],[d]} = \texttt{emb}(\Lambda^{[n_1],[n_2]}) \tag{90.17}$$

- **Cross Entropy Loss**

  Some pseudo-code

```
loss = nn.CrossEntropyLoss()
output = loss(input=x^{[n_c],[n_s]}, target=t^{[n_s]})
```

  Cross Entropy in Information Theory:

$$
\begin{aligned}
H(P^{\sigma}_{tar}, P^{\sigma}_{in}) &= -\sum_{\gamma\in[n_{\underline{c}}]} P_{tar}(\gamma|\sigma)\ln P_{in}(\gamma|\sigma) && (90.18)\\
&= -\sum_{\gamma\in[n_{\underline{c}}]} P_{tar}(\gamma|\sigma)\ln\left[\frac{P_{in}(\gamma|\sigma)}{P_{tar}(\gamma|\sigma)}P_{tar}(\gamma|\sigma)\right] && (90.19)\\
&= H(P^{\sigma}_{tar}) + D_{KL}(P^{\sigma}_{tar} \parallel P^{\sigma}_{in}) && (90.20)
\end{aligned}
$$

  Cross Entropy Loss in PyTorch:

  Let

  $n_{\underline{s}}$ = total number of samples being considered (usually batch size). $\sigma \in [n_{\underline{s}}]$

  $n_{\underline{c}}$ = number of classes in classification. $\gamma \in [n_{\underline{c}}]$

  $x^{[n_{\underline{c}}],[n_{\underline{s}}]}$ = input samples. Roughly speaking, if x, y is the data in supervised training, then this is the prediction $pred = forward(x)$.

  $t^{[n_{\underline{s}}]}$ = target samples. Roughly speaking, if x, y is the data in supervised training, then this is y.

  Define

$$
\begin{aligned}
P_{in}(\gamma|\sigma) &= \frac{\exp(x^{\gamma,\sigma})}{\sum_{\gamma'\in[n_{\underline{c}}]}\exp(x^{\gamma',\sigma})} && (90.21)\\
&= \mathrm{softmax}(x^{[n_{\underline{c}}],\sigma})(\gamma|\sigma) && (90.22)
\end{aligned}
$$

  Suppose $W^{\gamma} : values(\underline{t}) \to [0,1]$ for all $\gamma \in [n_{\underline{c}}]$.

  Define

$$
P_{tar}(\gamma|\sigma) = \frac{W^{\gamma}(t^{\sigma})\mathbb{1}(t^{\sigma} \neq -100)}{\sum_{\gamma\in[n_{\underline{c}}]} numerator} \tag{90.23}
$$

  The -100 on the right side of the last equation can be replaced by any other integer in $values(\underline{t})$ for which we want the loss to be zero (for example, it could be an integer used for padding)

  Now define the cross entropy loss $\mathcal{L}_{CE}$ by

$$\mathcal{L}_{CE} = \frac{1}{n_{\underline{s}}} \sum_{\sigma \in [n_{\underline{s}}]} H(P_{tar}(\cdot|\sigma), P_{in}(\cdot|\sigma)) \tag{90.24}$$

For example, if $W^{\gamma} = 1$, and $n_{\underline{c}} = 2$,

$$\mathcal{L}_{CE} \quad = \quad \frac{-1}{n_{\underline{s}}} \sum_{\sigma \in [n_{\underline{s}}]} [P_{tar}(0|\sigma) \ln P_{in}(0|\sigma) + P_{tar}(1|\sigma) \ln P_{in}(1|\sigma)] \tag{90.25}$$

- **unsqueeze-repeat-gather**

  Some pseudo-code

```
lll_loc = ll_loc0.unsqueeze(2).\
    repeat(1, 1, lll_state.shape[2])
lll_out = torch.gather(
    input=lll_state, dim=1, index=lll_loc)
```

  Sax uses the trio of operations unsqueeze-repeat-gather in the manner of the above pseudo-code. We have already discussed in Section C.48 how each of these 3 operations acts individually. Here we will discuss how they act jointly, when used as in the above pseudo-code.

  Let

  `lll_state=` $S^{[s_{ba}],[\Lambda],[d]}$

  `ll_loc0=` $L_0^{[s_{ba}],[a]}$

  `lll_loc=` $L^{[s_{ba}],[a],[d]}$

  `lll_out=` $O^{[s_{ba}],[a],[d]}$

  $\sigma \in s_{ba}, \lambda \in [\Lambda], \alpha \in [a], \delta \in [d]$

  `unsqueeze(2)` takes

$$L_0^{[s_{ba}],[a]} \to L_0^{[s_{ba}],[a],0} \tag{90.26}$$

  `lll_state.shape[2]` equals d, and `repeat(1, 1, d)` takes

$$L_0^{[s_{ba}],[a],0} \to L^{[s_{ba}],[a],[d]} = (\underbrace{L_0^{[s_{ba}],[a],0}, \ldots, L_0^{[s_{ba}],[a],0}}_{d \text{ times}}) \tag{90.27}$$

  Now define

$$\lambda(\sigma, \alpha) = L^{\sigma,\alpha,\delta} = L_0^{\sigma,\alpha} \tag{90.28}$$

  Then the `gather()` with `dim=1` outputs:

$$O^{\sigma,\alpha,\delta} = S^{\sigma,\lambda(\sigma,\alpha),\delta} \tag{90.29}$$

90.2 Bayesian Network for this model

Let

$\ell_{pad} = 86$, padding length, for this batch

$\ell_{enc} = 121$, encoded length, for this batch, $\ell_{enc} \geq \ell_{pad}$

$n_{dep} = 5$, number of copies of plain box connected in series, number of depths

$n_{att} = 2$, number of copies of dashed box connected in series, number of iterative (attention) layers.

$d = 768$, hidden dimension per head

$n_{\underline{h}} = 12$, number of heads (BERT base)

$D = dn_{\underline{h}}$, hidden dimension for all heads

$s_{ba} = 24$, batch size

$n_{il} = 6$, number of ilabels

$d_{me} = 300$, merge dimension

Fig.90.1 shows the bnet for Sax.[2]. The structural equations, printed in blue, for that bnet, are as follows.

$\underline{a}^{[86]}$: `ll_greedy_ilabel`
$\underline{B}^{[121],[768]}$: `lll_hidstate`
$\underline{d}^{[121],[768]}$: `lll_hidstate`
$\underline{E}^{[86],[768]}$: `lll_pred_code`
$\underline{G}^{[86],[768]}$: `lll_word_hidstate`
$\underline{I}^{[121],[768]}$: `lll_hidstate`
$\underline{L}^{[86],[6]}$: `lll_word_score`
$\underline{M}^{[86],[300]}$: `lll_word_hidstate`
$\underline{S}^{[86],[768]}$: `lll_word_hidstate`
$\underline{X}^{[86],[6]}$: `lll_word_score`

$$a^{[86]} = \text{argmax}(X^{[86],[6]}; dim = -1) \quad : \texttt{ll_greedy_ilabel} \tag{90.30a}$$

$$B^{[121],[768]} = \text{BERT}() \quad : \texttt{lll_hidstate} \tag{90.30b}$$

$$d^{[121],[768]} = \text{dropout}(I^{[121],[768]}) \quad : \texttt{lll_hidstate} \tag{90.30c}$$

$$E^{[86],[768]} = \text{embedding}(a^{[86]}) \quad : \texttt{lll_pred_code} \tag{90.30d}$$

[2]The bnet of Fig.90.1 and its structural equations printed in blue, were produced via the texnn software (Ref.[98])

Figure 90.1: Sax bnet. 2 copies of dashed box are connected in series. 5 copies (5 depths) of plain box are connected in series. However, in the first of those 5 plain box copies, the dotted box is omitted and node $\underline{G}$ feeds directly into node $\underline{M}$ (indicated by red arrow). We display the tensor shape superscripts in the PyTorch L2R order. All tensor shape superscripts have been simplified by omitting a $[s_{ba}]$ from their left side, where $s_{ba} = 24$ is the batch size.

$$G^{[86],[768]} = \text{gather}(d^{[121],[768]}; dim = -2) \\ : \texttt{lll_word_hidstate} \tag{90.30e}$$

$$I^{[121],[768]} = \left[B^{[121],[768]} \mathbb{1}(depth = 0) + M^{[86],[300]} \mathbb{1}(depth > 0)\right] \\ : \texttt{lll_hidstate} \tag{90.30f}$$

$$L^{[86],[6]} = M^{[86],[300]} W_{il}^{[300],[6]} \\ : \texttt{lll_word_score} \tag{90.30g}$$

$$M^{[86],[300]} = \left[G^{[86],[768]}\mathbb{1}(depth = 0) + S^{[86],[768]}\mathbb{1}(depth > 0)\right] W_{me}^{[768],[300]} \tag{90.30h}$$
$$: \texttt{lll_word_hidstate}$$

$$S^{[86],[768]} = E^{[86],[768]} + G^{[86],[768]} \tag{90.30i}$$
$$: \texttt{lll_word_hidstate}$$

$$X^{[86],[6]} = L^{[86],[6]}\mathbb{1}(depth > 0) \tag{90.30j}$$
$$: \texttt{lll_word_score}$$

90.3 Loss for this model

The Loss $\mathcal{L}$ is the sum of the Cross Entropy Loss $\mathcal{L}_{CE}$ and 4 penalty losses $\mathcal{L}_i$ for $i \in PL$ where $PL = \{POSC, HVC, HVE, EC\}$.

$$\mathcal{L} = \mathcal{L}_{CE} + \sum_{i \in PL} \lambda_i \mathcal{L}_i \tag{90.31}$$

where the λ_i are hyper-parameters to be determined heuristically.

In an earlier section, we discussed the Cross Entropy Loss at length. In this section, we will discuss the 4 penalty losses.

Below, we will use the standard notation for the **positive-part function** (a.k.a. the **reLU** function)

$$(x)_+ = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{90.32}$$
$$= \max(0, x) \tag{90.33}$$

Since loss is supposed to be bounded below (usually it is defined to be greater or equal to zero), the positive-part function comes in handy when defining a loss.

Let

ℓ = number of words, length of sentence. $\alpha \in [\ell]$

M = number of depths. $\mu \in [M]$

w^α = word at position α

$T_{pos} = \{N, V, JJ, RB\}$, POS tags, POS=Part Of Speech, N=Noun, V=Verb, JJ=Adjective, RB=Adverb

$T_{ex} = \{S, R, O, N\}$[3]. extraction tags (extags), S=Subject, R=Relation, O=Object, N=None

$T_{ex} \backslash N = T_{ex} - \{N\}$

$POS^{\alpha} \in T_{pos}$, Part Of Speech of w^{α}.

Importance indicator function.

$$IMP^{\alpha} = \mathbb{1}(POS^{\alpha} \in T_{pos}) \tag{90.34}$$

Head verb indicator function. A **head verb** is any verb that isn't a **light verb** (do, be, is, has, etc.).

$$HV^{\alpha} = \mathbb{1}(w^{\alpha} \text{ is a head verb}) \tag{90.35}$$

Let $P(\underline{t}^{\mu,\alpha} = t)$ denote an empirical probability for a table element $\underline{t}^{\mu,\alpha} \in T_{ex}$, for all $\mu \in [M]$ and $\alpha \in [\ell]$.

The O6 paper uses the following sentence to exemplify the 4 types of penalty losses.

Obama gained popularity after Oprah endorsed him for the presidency.

Henceforth, we will refer to this sentence as the osent (original sentence).

For the osent, the head verbs are gained, endorsed

Two valid extractions from osent are: (Obama; gained; popularity) and (Oprah; endorsed him for; the presidency).

1. **Part of Speech Coverage (POSC)**

   Penalize if some important words do not belong to at least one extraction.

   In osent: all the words Obama, gained, popularity, Oprah, endorsed, presidency must be covered in the set of extractions.

$$\mathcal{L}_{POSC} = \sum_{\alpha \in [\ell]} IMP^{\alpha} P_{POSC}(\alpha) \tag{90.36}$$

$$P_{POSC}(\alpha) = 1 - \max_{\mu \in [M]} \max_{t \in T_{ex} \backslash N} P(\underline{t}^{\mu,\alpha} = t) \tag{90.37}$$

[3]The Sax software uses a different set for T_{ex} than $T_{ex} = \{S, R, O, N\}$. In Sax, we use for T_{ex} the list `BASE_EXTAGS` (defined globally in the file `sax_globals`.) In `BASE_EXTAGS`, N becomes NONE (or 0) and R becomes REL (or 3). Also note that 2 tranets are trained by Sax, one for extraction (task=ex), and one for splitting (task=cc). For task=cc, T_{ex} is replaced by T_{cc}. In Sax, we use for T_{cc} the list `BASE_CCTAGS` (defined globally in the file `sax_globals`.) In `BASE_CCTAGS`, N becomes NONE (or 0) and R becomes CC (or 3).

2. **Head Verb Coverage (HVC)**

   Penalize if a head verb is not present in the relation (R) of any extraction.

   In osent: (Obama; gained; popularity), (Obama; gained; presidency) is not a comprehensive set of extractions.

$$\mathcal{L}_{HVC} = \sum_{\alpha \in [\ell]} HV^{\alpha} P_{HVC}(\alpha) \tag{90.38}$$

$$P_{HVC}(\alpha) = \left| 1 - \sum_{\mu \in [M]} P(\underline{t}^{\mu,\alpha} = R) \right| \tag{90.39}$$

3. **Head Verb Exclusivity (HVE)**

   Penalize extractions that contain more than one head verb in their relation (R).

   In osent: gained popularity after Oprah endorsed is not a good relation as it contains two head verbs

$$\mathcal{L}_{HVE} = \sum_{\mu \in [M]} \left(\sum_{\alpha \in [\ell]} HV^{\alpha} P(\underline{t}^{\mu,\alpha} = R) - 1 \right)_{+} \tag{90.40}$$

4. **Extraction Count (EC)**

   Penalize if the total number of extractions with head verbs in the relation (R) is too small; i.e., it is smaller than the number of head verbs in the osent.

$$\mathcal{L}_{EC} = \left(\sum_{\alpha \in [\ell]} HV^{\alpha} - \sum_{\mu \in [M]} EC^{\mu} \right)_{+} \tag{90.41}$$

$$EC^{\mu} = \max_{\alpha \in [\ell]} HV^{\alpha} P(\underline{t}^{\mu,\alpha} = R) \tag{90.42}$$

Chapter 91

Shannon Information Theory

Throughout this book, we often use the definitions of entropy, mutual information, conditional mutual information, cross entropy, etc. All these definitions originated from the seminal work of Claude Shannon. Note that the connection between bnets and Shannon Information Theory (SIT) goes much deeper than that. Most of SIT can be expressed graphically using bnets, as I explain in my paper Ref.[97].

Below I present Ref.[97] in its entirety.

Title: *Shannon Information Theory Without Shedding Tears Over Delta & Epsilon Proofs or Typical Sequences*

Abstract: This paper begins with a discussion of integration over probability types (p-types). After doing that, the paper re-visits 3 mainstay problems of classical (non-quantum) Shannon Information Theory (SIT): source coding without distortion, channel coding, and source coding with distortion. The paper proves well-known, conventional results for each of these 3 problems. However, the proofs given for these results are not conventional. They are based on complex integration techniques (approximations obtained by applying the method of steepest descent to p-type integrals) instead of the usual delta & epsilon and typical sequences arguments. Another unconventional feature of this paper is that we make ample use of classical Bayesian networks (CB nets). This paper showcases some of the benefits of using CB nets to do classical SIT.

91.1 Introduction

For a good textbook on classical (non-quantum) Shannon Information Theory (SIT), see, for example, Ref.[83] by Cover and Thomas. Henceforth we will refer to it as C&T. For a good textbook on classical (non-quantum) Bayesian Networks, see, for example, Ref.[16] by Koller and Friedman.

This paper begins with a discussion of integration over probability types (p-types). After doing that, the paper re-visits 3 mainstay problems of classical SIT:

- source coding (lossy compression) without distortion
- channel coding
- source coding with distortion

The paper proves well-known, conventional results for each of these 3 problems. However, the proofs given for these results are not conventional. They are based on complex integration techniques (approximations obtained by applying the method of steepest descent to p-type integrals) instead of the usual delta & epsilon and typical sequences arguments.

Another unconventional feature of this paper is that we make ample use of classical Bayesian networks (CB nets). This paper showcases some of the benefits of using CB nets to do classical SIT.

P-types were introduce into SIT by Csiszár and Körner (see Ref.[33]). P-type integration is a natural, almost obvious consequence of the theory of p-types, although it is not spelled out explicitly in the book by Csiszár and Körner. In fact, all workers whose work I am familiar with, including Csiszár and Körner, use p-types frequently, but they do not use p-type integration. Instead, they use delta & epsilon and typical sequences arguments to bound some finite sums which are discrete approximations of p-type integrals.

The conventional delta & epsilon arguments are more rigorous than the p-type integration arguments presented here. Although less rigorous than traditional arguments, p-type integration arguments have the virtue that they are easier to understand and follow, especially by people who are not well versed in rigorous analysis. Such is the case with many physicists and engineers. A similar problem occurs when teaching Calculus. One can teach Calculus with the full panoply of delta & epsilon arguments from a textbook such as the legendary one by W. Rudin (Ref.[70]). Or one can teach Calculus at the level and scope of a college freshman course for engineers. Each approach appeals to a different audience and fulfils different needs.

Most of our results are not exact. They are leading order terms in asymptotic expansions for large n, where n is the number of letters in a codeword. These approximations become increasingly more accurate as $n \to \infty$.

This paper is almost self contained, although a few times we assume certain inequalities and send the reader to C&T for a proof of them.

91.2 Preliminaries and Notation

In this section, we will describe some basic notation used throughout this paper.

As usual, $\mathbb{Z}, \mathbb{R}, \mathbb{C}$ will denote the integers, real numbers, and complex numbers, respectively. We will sometimes add superscripts to these symbols to indicate subsets of these sets. For instance, we'll use $\mathbb{R}^{\geq 0}$ to denote the set of non-negative reals. For $a, b \in \mathbb{Z}$ such that $a \leq b$, let $Z_{a,b} = \{a, a+1, a+2, \ldots, b\}$.

Let $\delta_y^x = \delta(x, y)$ denote the Kronecker delta function: it equals 1 if $x = y$ and 0 if $x \neq y$. Let $\theta(S)$ denote the truth function: it equals 1 if statement S is true and 0 otherwise. For example, $\delta_x^y = \theta(x = y)$. Another example is the step function $\theta(x > 0)$: it equals 1 if $x > 0$ and is zero otherwise.

For any matrix $M \in \mathbb{C}^{p \times q}$, M^* will denote its complex conjugate, M^T its transpose, and $M^\dagger = M^{*T}$ its Hermitian conjugate.

Random variables will be denoted by underlined letters; e.g., $\underline{a}$. The (finite) set of values (states) that $\underline{a}$ can assume will be denoted by $val(\underline{a})$. Let $N_{\underline{a}} = |val(\underline{a})|$. The probability that $\underline{a} = a$ will be denoted by $P(\underline{a} = a)$ or $P_{\underline{a}}(a)$, or simply by $P(a)$ if the latter will not lead to confusion in the context it is being used. We will use $pd(val(\underline{a}))$ to denote the set of all probability distributions with domain $val(\underline{a})$. For joint random variables $(\underline{a}, \underline{b})$, let $val(\underline{a}, \underline{b}) = val(\underline{a}) \times val(\underline{b}) = \{(a, b) : a \in val(\underline{a}), b \in val(\underline{b})\}$.

Sometimes, when two random variables $\underline{a}\langle 1\rangle$ and $\underline{a}\langle 2\rangle$ satisfy $val(\underline{a}\langle 1\rangle) = val(\underline{a}\langle 2\rangle)$, we will omit the indices $\langle 1\rangle$ and $\langle 2\rangle$ and refer to both random variables as $\underline{a}$. We shall do this sometimes even if the random variables $\underline{a}\langle 1\rangle$ and $\underline{a}\langle 2\rangle$ are not identically distributed! This notation, *if used with caution*, does not lead to confusion and does avoid a lot of index clutter.

Suppose $\{P_{\underline{x},\underline{y}}(x, y)\}_{\forall x,y} \in pd(val(\underline{x}, \underline{y}))$. We will often use the expectation operators $E_x = \sum_x P(x)$, $E_{x,y} = \sum_{x,y} P(x, y)$, and $E_{y|x} = \sum_y P(y|x)$. Note that $E_{x,y} = E_x E_{y|x}$. Let

$$P(x : y) = \frac{P(x, y)}{P(x)P(y)} \ . \tag{91.1}$$

Note that $E_x P(x : y) = E_y P(x : y) = 1$.

Suppose n is any positive integer. Let $\underline{x}^n = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n)$ be the random variable that takes one values $x^n = (x_1, x_2, \ldots, x_n) \in val(\underline{x})^n$.

The **rate of** $\underline{x}$ is defined as $R_{\underline{x}} = \frac{\ln N_{\underline{x}}}{n}$.

$\underline{x}^n$ is said to be i.i.d. (independent, identically distributed) if $val(\underline{x}_j) = val(\underline{x})$ for all $j \in Z_{1,n}$ and there is a $P_{\underline{x}} \in pd(val(\underline{x}))$ such that $P_{\underline{x}^n}(x^n) = \prod_{j=1}^n \{P_{\underline{x}}(x_j)\}$. When $\underline{x}^n$ is i.i.d., we will sometimes use $P_{\underline{x}}(x^n)$ to denote the more correct expression $P_{\underline{x}^n}(x^n)$ and say that $P_{\underline{x}}(x^n)$ is an i.i.d. source.

Suppose $\{P(y^n|x^n)\}_{\forall y^n} \in pd(val(\underline{y})^n)$ for all $x^n \in val(\underline{x})^n$. $P(y^n|x^n)$ is said to be a discrete memoryless channel (DMC) if $P(y^n|x^n) = \prod_{j=1}^n P(y_j|x_j)$.

We will use the following measures of various types of information (entropy):

- The (plain) entropy of the random variable $\underline{x}$ is defined in the classical case by

$$H(\underline{x}) = E_x \ln \frac{1}{P(x)} \ , \tag{91.2}$$

which we also call $H_{P_{\underline{x}}}(\underline{x})$, $H\{P(x)\}_{\forall x}$, and $H(P_{\underline{x}})$. This quantity measures the spread of $P_{\underline{x}}$.

One can also consider plain entropy for a joint random variable $\underline{x} = (\underline{x}_1, \underline{x}_2)$. For $P_{\underline{x}_1,\underline{x}_2} \in pd(val(\underline{x}_1, \underline{x}_2))$ with marginal probability distributions $P_{\underline{x}_1}$ and $P_{\underline{x}_2}$, one defines a joint entropy $H(\underline{x}_1, \underline{x}_2) = H(\underline{x})$ and partial entropies $H(\underline{x}_1)$ and $H(\underline{x}_2)$.

- The conditional entropy of $\underline{y}$ given $\underline{x}$ is defined in the classical case by

$$\begin{aligned} H(\underline{y}|\underline{x}) &= E_{x,y} \ln \frac{1}{P(y|x)} && (91.3) \\ &= H(\underline{y}, \underline{x}) - H(\underline{x}) \ , && (91.4) \end{aligned}$$

which we also call $H_{P_{\underline{x},\underline{y}}}(\underline{y}|\underline{x})$. This quantity measures the conditional spread of $\underline{y}$ given $\underline{x}$.

- The Mutual Information (MI) of $\underline{x}$ and $\underline{y}$ is defined in the classical case by

$$\begin{aligned} H(\underline{y} : \underline{x}) &= E_{x,y} \ln P(x : y) = E_x E_y P(x : y) \ln P(x : y) && (91.5) \\ &= H(\underline{x}) + H(\underline{y}) - H(\underline{y}, \underline{x}) \ , && (91.6) \end{aligned}$$

which we also call $H_{P_{\underline{x},\underline{y}}}(\underline{y} : \underline{x})$. This quantity measures the correlation between $\underline{x}$ and $\underline{y}$.

- The Conditional Mutual Information (CMI, which can be read as "see me") of $\underline{x}$ and $\underline{y}$ given $\underline{\lambda}$ is defined in the classical case by:

$$\begin{aligned} H(\underline{y} : \underline{x}|\, \underline{\lambda}\,) &= E_{x,y,\lambda} \ln \frac{P(x, y|\lambda)}{P(x|\lambda)P(y|\lambda)} && (91.7) \\ &= E_{x,y,\lambda} \ln \frac{P(x, y, \lambda)P(\lambda)}{P(x, \lambda)P(y, \lambda)} && (91.8) \\ &= H(\underline{x}|\, \underline{\lambda}\,) + H(\underline{y}|\, \underline{\lambda}\,) - H(\underline{y}, \underline{x}|\, \underline{\lambda}\,) \ , && (91.9) \end{aligned}$$

which we also call $H_{P_{\underline{x},\underline{y},\underline{\lambda}}}(\underline{y} : \underline{x}|\, \underline{\lambda}\,)$. This quantity measures the conditional correlation of $\underline{x}$ and $\underline{y}$ given $\underline{\lambda}$.

- The relative information of $P \in pd(val(\underline{x}))$ divided by $Q \in pd(val(\underline{x}))$ is defined by

$$D\{P(x)//Q(x)\}_{\forall x} = \sum_x P(x) \ln \frac{P(x)}{Q(x)} \ , \tag{91.10}$$

which we also call $D(P_{\underline{x}}//Q_{\underline{x}})$.

Note that we define entropies using natural logs. Our strategy is to use natural log entropies for all intermediate analytical calculations, and to convert to base-2 logs at the end of those calculations if a base-2 log numerical answer is desired. Such a conversion is of course trivial using $\log_2 x = \frac{\ln x}{\ln 2}$ and $\ln 2 = 0.6931$

We will use the following well-known integral representation of the Dirac delta function:

$$\delta(x) = \int_{-\infty}^{+\infty} \frac{dk}{2\pi} \ e^{ikx} \ . \tag{91.11}$$

We will also use the following integral representation of the step function:

$$\theta(x > 0) = \int_{-\infty}^{+\infty} \frac{dk}{2\pi i} \ \frac{e^{ikx}}{(k - i\epsilon)} \ , \tag{91.12}$$

for some $\epsilon > 0$. Eq.(91.12) follows because the integrand has a simple pole at $k = i\epsilon$. Let $k = k_r + ik_i$. If $x > 0$, the integrand goes to zero in the upper half of the (k_r, k_i) plane and it goes to infinity in the lower half plane, so we are forced to close the contour of integration in the upper half plane, which means the pole lies inside the contour. When $x < 0$, we are forced to close the contour in the lower half plane and thus the pole lies outside the contour.

Suppose $\mathcal{L}(v)$ is a real valued function that depends in a continuous manner on N real variables $v = \{v_j\}_{j=1}^N$. The following variational operator can be applied to $\mathcal{L}(v)$:

$$\delta = \sum_j \delta v_j \frac{\partial}{\partial v_j} \ . \tag{91.13}$$

The N-dimensional Taylor expansion of $\mathcal{L}(v)$ about the point $v = 0$ can be expressed as

$$f(v) = f(0) + [\delta f(v)]_{v=0} + \frac{1}{2!}[\delta^2 f(v)]_{v=0} + + \frac{1}{3!}[\delta^3 f(v)]_{v=0} + \dots \ . \tag{91.14}$$

We will often use the following Taylor expansions:

$$x^\epsilon = e^{\epsilon \ln x} = 1 + \epsilon \ln x + \frac{1}{2}(\epsilon \ln x)^2 + \dots \ , \tag{91.15}$$

and

$$\ln(1+x) = x - \frac{x^2}{2} + \ldots \text{ (converges if } |x| < 1) \ . \tag{91.16}$$

91.3 Integration Over P-types

In this section, we will define integration over probability types (p-types). The set of p-types for a given n fills all of $pd(val(\underline{x}))$ in an increasingly finer way as $n \to \infty$. Thus, once the density of p-types at each point of $pd(val(\underline{x}))$ is known, we can integrate that density over a particular region $R \subset pd(val(\underline{x}))$ to get the number of p-types within R. We will define integration over p-types that depend on a single variable (univariate p-types), or multiple variables (multivariate p-types). We will also define integration over conditional p-types. Finally, we will define Dirac delta functions for integration over p-types.

91.3.1 Integration Over Univariate P-type

For any $x^n \in S^n_{\underline{x}}$, denote the number of occurrences of $x \in val(\underline{x})$ within x^n by $N(x|x^n)$. Hence

$$N(x|x^n) = \sum_{j=1}^{n} \theta(x_j = x) \ . \tag{91.17}$$

One can now say that two elements x^n and x'^n of $val(\underline{x})^n$ are equivalent if, for all $x \in val(\underline{x})$, x^n and x'^n both have the same number of occurrences of x. This equivalence relation partitions $S^n_{\underline{x}}$ into equivalence classes given by, for any $x^n \in val(\underline{x})^n$,

$$[x^n] = \{x'^n \in val(\underline{x})^n : N(x|x^n) = N(x|x'^n) \forall x \in val(\underline{x})\} \ . \tag{91.18}$$

For each class $[x^n]$ and $x \in val(\underline{x})$, we can define

$$P_{[x^n]}(x) = \frac{N(x|x^n)}{n} \ . \tag{91.19}$$

Clearly, $\{P_{[x^n]}(x)\}_{\forall x} \in pd(val(\underline{x}))$. We will refer to this probability distribution as a p-type.

Note that if $Q(x^n)$ is an i.i.d. source,

$$Q(x^n) = \prod_{j=1}^{n} Q(x_j) \ , \tag{91.20}$$

so

$$Q(x^n) = \prod_{x \in val(\underline{x})} \left\{ Q(x)^{N(x|x^n)} \right\} = e^{n \sum_x P_{[x^n]}(x) \ln Q(x)} \ . \tag{91.21}$$

Define the following integration operator:

$$\int \mathcal{D}P_{[x^n]} = \prod_x \left\{ \int_0^1 dP_{[x^n]}(x) \right\} \delta \left(\sum_x P_{[x^n]}(x) - 1 \right) . \tag{91.22}$$

We will denote the number of elements in a class $[x^n]$ by

$$d_{[x^n]} = |[x^n]| . \tag{91.23}$$

Claim 132

$$\sum_{x^n} = \sum_{[x^n]} d_{[x^n]} . \tag{91.24}$$

proof: The classes $[x^n]$ are non-overlapping and they cover all of $S^n_{\underline{x}}$.
QED

Claim 133 *For any* $x^n \in S^n_{\underline{x}}$,

$$d_{[x^n]} = (d_{[x^n]})_{H=0} \ e^{nH(P_{[x^n]})} , \tag{91.25}$$

where

$$(d_{[x^n]})_{H=0} = \frac{1}{(2\pi n)^{\frac{N_{\underline{x}}-1}{2}} \sqrt{\prod_x P_{[x^n]}(x)}} . \tag{91.26}$$

proof: Let

$$val(\underline{x}) = \{x(j) : j \in Z_{1,N_{\underline{x}}}\} \tag{91.27}$$

and

$$r_j = N(x(j)|x^n) \tag{91.28}$$

for all $j \in Z_{1,N_{\underline{x}}}$. Note that $\sum_{j=1}^{N_{\underline{x}}} r_j = n$. Recall Stirling's formula:

$$n! \approx \sqrt{2\pi n} \ \ n^n e^{-n} \tag{91.29}$$

for $n >> 1$. Combinatorics gives a value for $|[x^n]|$ in terms of factorials. If we approximate those factorials using Stirling's formula, we get

$$\begin{aligned} |[x^n]| &= \frac{n!}{\prod_{j=1}^{N_{\underline{x}}} \{r_j!\}} && (91.30) \\ &= \frac{1}{(2\pi)^{\frac{N_{\underline{x}}-1}{2}}} \left(\frac{n}{r_1 r_2 \ldots r_{N_{\underline{x}}}} \right)^{\frac{1}{2}} e^{-n+n\ln n - \sum_j \{-r_j + r_j \ln r_j\}} && (91.31) \\ &= \frac{\exp(-n \sum_j \frac{r_j}{n} \ln \frac{r_j}{n})}{(2\pi n)^{\frac{N_{\underline{x}}-1}{2}} \sqrt{\prod_j \left\{ \frac{r_j}{n} \right\}}} . && (91.32) \end{aligned}$$

QED

Claim 134

$$\sum_{[x^n]} = \int \mathcal{D}P_{[x^n]} n^{N_{\underline{x}}-1} \ . \tag{91.33}$$

proof: For any i.i.d. source $Q(x^n)$, we have that

$$\begin{aligned} 1 &= \sum_{x^n} Q(x^n) && (91.34) \\ &= \sum_{[x^n]} d_{[x^n]} e^{n \sum_x P_{[x^n]}(x) \ln Q(x)} && (91.35) \\ &= \int \frac{\mathcal{D}P_{[x^n]}}{\Delta V} \frac{e^{\mathcal{L}_0}}{(2\pi n)^{\frac{N_{\underline{x}}-1}{2}} \sqrt{\prod_x P_{[x^n]}(x)}}, && (91.36) \end{aligned}$$

where ΔV is yet to be determined and

$$\mathcal{L}_0 = n \sum_x P_{[x^n]}(x) \ln \frac{Q(x)}{P_{[x^n]}(x)} \ . \tag{91.37}$$

We add to $\mathcal{L}_0$ a Lagrange multiplier term that constrains the components of the vector $\{P_{[x^n]}(x)\}_{\forall x}$ so that they sum to one:

$$\mathcal{L} = \mathcal{L}_\lambda = \mathcal{L}_0 + n\lambda \left(\sum_x P_{[x^n]}(x) - 1 \right) \tag{91.38}$$

for any $\lambda \in \mathbb{R}$. Our goal is to approximate the integral Eq.(91.36) using the method of steepest descent. We just want to get the leading order term in an asymptotic expansion of the integral for large n. To get this leading order term, it is sufficient to approximate $\mathcal{L}$ to second order in $\delta P_{[x^n]}(x)$, about the point (or points) that have a vanishing first variation $\delta\mathcal{L}$. Thus, approximate

$$\mathcal{L} \approx \widetilde{\mathcal{L}} + \delta\widetilde{\mathcal{L}} + \frac{1}{2}\delta^2\widetilde{\mathcal{L}} \ , \tag{91.39}$$

where quantities with a tilde over them are evaluated at a tilde (saddle) point that satisfies

$$\delta\widetilde{\mathcal{L}} = 0 \ . \tag{91.40}$$

It's easy to check that

$$\delta\mathcal{L} = n \sum_x \delta P_{[x^n]}(x) \ln \left(\frac{Q(x) e^{-1+\lambda}}{P_{[x^n]}(x)} \right) \ , \tag{91.41}$$

and

$$\delta^2\mathcal{L} = -n \sum_x \frac{[\delta P_{[x^n]}(x)]^2}{P_{[x^n]}(x)} \ . \tag{91.42}$$

Next, for each x, we set to zero the coefficient of $\delta P_{[x^n]}(x)$ in $\delta\mathcal{L}$. After doing that, we enforce the constraint that $\sum_x P_{[x^n]}(x) = 1$. This leads us to conclude that

$$\widetilde{P}_{[x^n]}(x) = Q(x) \ . \tag{91.43}$$

Using this value of $\widetilde{P}_{[x^n]}(x)$, we get

$$\widetilde{\mathcal{L}} = 0 \tag{91.44}$$

and

$$\delta^2\widetilde{\mathcal{L}} = -n\sum_x \frac{[\delta P_{[x^n]}(x)]^2}{Q(x)} \ . \tag{91.45}$$

From Eq.(91.36), we get

$$1 = \frac{1}{\Delta V(2\pi n)^{\frac{N_{\underline{x}}-1}{2}}\sqrt{\prod_x Q(x)}}\Gamma \ , \tag{91.46}$$

where

$$\Gamma = \int \mathcal{D}P_{[x^n]} e^{-n\sum_x \frac{\left[\delta P_{[x^n]}(x)\right]^2}{2Q(x)}} = \sqrt{\frac{\pi^{N_{\underline{x}}-1}}{\prod_x \left\{\frac{n}{2Q(x)}\right\}\frac{2}{n}}} \ . \tag{91.47}$$

The final integral was performed using Eq.(91.256). This implies $1/\Delta V = n^{N_{\underline{x}}-1}$. **QED**

Note that Eqs.(91.33) and (91.251) imply that

$$\sum_{[x^n]} 1 = \frac{n^{N_{\underline{x}}-1}}{(N_{\underline{x}}-1)!} \tag{91.48}$$

so the number of p-types with a given n in $pd(val(\underline{x}))$ varies polynomial with n.

91.3.2 Integration Over Multivariate P-types

There exists a very natural 1-1 onto map from $val(\underline{x})^n \times val(\underline{y})^n$ to $(val(\underline{x}) \times val(\underline{y}))^n$, namely the one that identifies $(x_j)_{\forall j}(y_j)_{\forall j}$ with $\begin{bmatrix} x_j \\ y_j \end{bmatrix}_{\forall j}$. Thus, the definitions and claims given in the previous section for $N(x|x^n)$, $[x^n]$, $P_{[x^n]}(x)$ and $\int \mathcal{D}P_{[x^n]}$ generalize very naturally to give analogous definitions and claims for $N(x,y|x^n,y^n)$, $[x^n,y^n]$, $P_{[x^n,y^n]}(x,y)$ and $\int \mathcal{D}P_{[x^n,y^n]}$. For example,

$$N(x,y|x^n,y^n) = N(\begin{pmatrix} x \\ y \end{pmatrix} | \begin{pmatrix} x^n \\ y^n \end{pmatrix}) = \sum_j \theta(\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_j \\ y_j \end{pmatrix}) \ . \tag{91.49}$$

We will sometimes use [ ] as an abbreviation for a class. For example, we might abbreviate $P_{[a^n,b^n,c^n]}(a,b,c)$ by $P_{[\,]}(a,b,c)$.

Note that when $y^n = x^n$ in $P_{[x^n,y^n]}$,

$$P_{[x^n,x^n]}(x,y) = \delta_y^x \; P_{[x^n]}(x) \; . \tag{91.50}$$

Note also that we can express $\delta_{x^n}^{y^n}$ as follows

$$\begin{aligned} e^{n\sum_{x,y} P_{[x^n,y^n]}(x,y)\ln\delta_y^x} &= \begin{cases} 0, \text{ if } \exists(x,y) \text{ such that } \delta_x^y = 0 \text{ and } P_{[x^n,y^n]}(x,y) \neq 0 \\ 1, \text{ otherwise} \end{cases} && (91.51) \\ &= \theta(\forall(x,y) : y \neq x \Rightarrow P_{[x^n,y^n]}(x,y) = 0) && (91.52) \\ &= \delta_{x^n}^{y^n} \; . && (91.53) \end{aligned}$$

91.3.3 Integration Over Conditional P-types

For any $x^n \in S_{\underline{x}}^n$ and $y^n \in S_{\underline{y}}^n$, define conditional classes by

$$[y^n|x^n] = \left\{ (x'^n, y'^n) \in val(\underline{x})^n \times val(\underline{y})^n : \frac{N(x,y|x^n,y^n)}{\sum_y N(x,y|x^n,y^n)} = \frac{N(x,y|x'^n,y'^n)}{\sum_y N(x,y|x'^n,y'^n)} \forall (x,y) \in val(\underline{x}) \times val(\underline{y}) \right\} \tag{91.54}$$

and conditional probability types by

$$P_{[y^n|x^n]}(y|x) = \frac{N(x,y|x^n,y^n)}{\sum_y N(x,y|x^n,y^n)} = \frac{P_{[x^n,y^n]}(x,y)}{P_{[x^n,y^n]}(x)} \tag{91.55}$$

for all $x \in val(\underline{x})$ and $y \in val(\underline{y})$.

We will sometimes use [ ] as an abbreviation for a conditional class. For example, we might abbreviate $P_{[a^n,b^n|c^n,d^n]}(a,b|c,d)$ by $P_{[\,]}(a,b|c,d)$.

Define the following integration operator:

$$\int \mathcal{D}P_{[y^n|x^n]} = \prod_{x,y} \left\{ \int_0^1 dP_{[y^n|x^n]}(y|x) \right\} \prod_x \left\{ \delta\left(\sum_y P_{[y^n|x^n]}(y|x) - 1 \right) \right\} \; . \tag{91.56}$$

We will denote the number of elements in conditional class $[y^n|x^n]$ by

$$d_{[y^n|x^n]} = |[y^n|x^n]| \; . \tag{91.57}$$

Claim 135

$$\sum_{x^n,y^n} = \sum_{[x^n]} d_{[x^n]} \sum_{[y^n|x^n]} d_{[y^n|x^n]} \; . \tag{91.58}$$

proof: For any DMC $Q(y^n|x^n)$, we must have

$$1 = \sum_{[y^n|x^n]} d_{[y^n|x^n]} Q(y^n|x^n) \ . \tag{91.59}$$

If $Q(x^n)$ is an i.i.d source and $Q(x^n, y^n) = Q(y^n|x^n)Q(x^n)$, then the last equation implies

$$\begin{aligned} 1 &= \sum_{[x^n]} d_{[x^n]} Q(x^n) \sum_{[y^n|x^n]} d_{[y^n|x^n]} Q(y^n|x^n) & (91.60) \\ &= \sum_{[x^n]} d_{[x^n]} \sum_{[y^n|x^n]} d_{[y^n|x^n]} Q(x^n, y^n) \ . & (91.61) \end{aligned}$$

But also

$$1 = \sum_{x^n, y^n} Q(x^n, y^n) \ . \tag{91.62}$$

Since $Q(x^n, y^n)$ is an arbitrary i.i.d. source, the claim follows.
QED

Claim 136

$$d_{[y^n|x^n]} = \frac{d_{[x^n, y^n]}}{d_{[x^n]}} \ . \tag{91.63}$$

proof: Combinatorics?
QED

Claim 137

$$\sum_{[x^n]} \sum_{[y^n|x^n]} = \sum_{[x^n, y^n]} \ . \tag{91.64}$$

proof: This follows from Claims 135 and 136 and the fact that $\sum_{x^n, y^n} = \sum_{[x^n, y^n]} d_{[x^n, y^n]}$.

QED

Alternatively, one could prove Claim 137 by combinatorics and then prove Claim 136 from Claims 135 and 137.

Claim 138

$$\int \mathcal{D}P_{[x^n]} \int \mathcal{D}P_{[y^n|x^n]} \left[\prod_x \left\{ P_{[x^n]}(x) \right\} \right]^{N_{\underline{y}}-1} = \int \mathcal{D}P_{[x^n, y^n]} \tag{91.65}$$

proof: Let LHS and RHS denote the left hand side and right hand side of Eq.(91.65).

Recall that Dirac delta functions obey $\delta(ax) = \frac{1}{|a|}\delta(x)$. This proof hinges on that simple identity.

Define

$$\Omega_1 = \prod_x \left\{ \int_0^1 dP_{[x^n]}(x) \right\} \delta\left(\sum_x P_{[x^n]}(x) - 1 \right) \tag{91.66}$$

and

$$\Omega_2 = \prod_{x,y} \left\{ \int_0^1 dP_{[y^n|x^n]}(y|x) \right\} \prod_x \left\{ \delta\left(\sum_y P_{[y^n|x^n]}(y|x) - 1 \right) \right\} \left[\prod_x \left\{ P_{[x^n]}(x) \right\} \right]^{N_{\underline{y}}-1} . \tag{91.67}$$

Then

$$\begin{aligned} LHS &= \Omega_1 \Omega_2 && (91.68) \\ &= \Omega_1 \prod_{x,y} \left\{ \int_0^1 dP_{[x^n,y^n]}(x,y) \right\} \prod_x \left\{ \delta\left(\sum_y P_{[x^n,y^n]}(x,y) - P_{[x^n]}(x) \right) \right\} && (91.69) \\ &= \prod_{x,y} \left\{ \int_0^1 dP_{[x^n,y^n]}(x,y) \right\} \delta\left(\sum_{x,y} P_{[x^n,y^n]}(x,y) - 1 \right) && (91.70) \\ &= RHS && (91.71) \end{aligned}$$

This works because LHS has $n_i = N_{\underline{x}} + N_{\underline{x}} N_{\underline{y}}$ integrals and $n_\delta = N_{\underline{x}} + 1$ delta functions, for a total of $n_i - n_\delta = N_{\underline{x}} N_{\underline{y}} - 1$ degrees of freedom. RHS has $N_{\underline{x}} N_{\underline{y}}$ integrals and one delta function for the *same total* of $N_{\underline{x}} N_{\underline{y}} - 1$ degrees of freedom. **QED**

Claim 139

$$\begin{aligned} \sum_{[y^n|x^n]} &= \int \mathcal{D}P_{[y^n|x^n]} \frac{n^{N_{\underline{y}} N_{\underline{x}}}}{n^{N_{\underline{x}}}} \left[\prod_x \left\{ P_{[x^n]}(x) \right\} \right]^{N_{\underline{y}}-1} && (91.72) \\ &= \int \mathcal{D}P_{[y^n|x^n]} (n P_{[x^n]}{}^{g.m.})^{N_{\underline{x}} N_{\underline{y}} - N_{\underline{x}}} , && (91.73) \end{aligned}$$

where

$$P_{[x^n]}{}^{g.m.} = \left[\prod_x \left\{ P_{[x^n]}(x) \right\} \right]^{\frac{1}{N_{\underline{x}}}} \tag{91.74}$$

is the geometric mean of $P_{[x^n]}$.

proof: Substitute

$$\int \mathcal{D}P_{[x^n]} = \frac{1}{n^{N_{\underline{x}}-1}} \sum_{[x^n]} , \tag{91.75}$$

and

$$\int \mathcal{D}P_{[x^n,y^n]} = \frac{1}{n^{N_{\underline{x}}N_{\underline{y}}-1}} \sum_{[x^n,y^n]} \tag{91.76}$$

into Eq.(91.65) and then compare the result with Eq.(91.64).
QED

91.3.4 Dirac Delta Functions For P-type Integration

One occasionally finds it useful to use Dirac delta functions for p-type integration. Suppose $x^n, y^n \in S^n_{\underline{x}}$ and ϵ is a real number satisfying $0 < \epsilon << 1$. Let $\mathcal{X} = [x^n]$ and $\mathcal{Y} = [y^n]$. Define

$$V_a = \frac{a^{N_{\underline{x}}-1} \pi^{\frac{N_{\underline{x}}-1}{2}}}{\sqrt{N_{\underline{x}}}} \tag{91.77}$$

for any positive real number a. We will refer to the following functions as Dirac delta functions for setting $\mathcal{X}$ and $\mathcal{Y}$ equal

$$\delta(\mathcal{X}, \mathcal{Y}) = \theta(\mathcal{X} = \mathcal{Y}) , \tag{91.78}$$

$$\delta_\epsilon(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{\epsilon^2} \sum_x \{P_{\mathcal{X}}(x) - P_{\mathcal{Y}}(x)\}^2\right) , \tag{91.79}$$

$$\delta_\epsilon(x^n, y^n) = \frac{\delta_\epsilon(\mathcal{X}, \mathcal{Y})}{\sqrt{d_{\mathcal{X}} d_{\mathcal{Y}}}\, V_{n\epsilon}} , \tag{91.80}$$

and

$$\delta_\epsilon(P_{\mathcal{X}} - P_{\mathcal{Y}}) = \frac{\delta_\epsilon(\mathcal{X}, \mathcal{Y})}{V_\epsilon} . \tag{91.81}$$

Claim 140

$$\sum_{x^n} \delta_\epsilon(x^n, y^n) = 1 , \tag{91.82}$$

and

$$\int \mathcal{D}P_{\mathcal{X}}\ \delta_\epsilon(P_{\mathcal{X}} - P_{\mathcal{Y}}) = 1 . \tag{91.83}$$

proof: This follows from integration formula Eq.(91.256).
QED

91.4 Source Coding (Lossy Compression)

We consider all source coding protocols that can be described by the following CB net

$$\underline{\widehat{x}}^n \longleftarrow \underline{m} \longleftarrow \underline{x}^n \ , \tag{91.84}$$

with $val(\underline{x}) = val(\underline{\widehat{x}})$ and

$$P(x^n) = \prod_{j=1}^{n} P_{\underline{x}}(x_j) \ , \tag{91.85}$$

$$P(m|x^n) = \delta(m, m(x^n)) \tag{91.86}$$

and

$$P(\widehat{x}^n|m) = \delta(\widehat{x}^n, \widehat{x}^n(m)) \ . \tag{91.87}$$

Assume that we are given a source $P_{\underline{x}} \in pd(val(\underline{x}))$. The encoding function $m(\cdot)$ and the decoding function $\widehat{x}^n(\cdot)$ are yet to be specified.[1]

The probability of error is defined by

$$P_{err} = P(\underline{\widehat{x}}^n \neq \underline{x}^n) \ . \tag{91.88}$$

We find it more convenient to work with the probability of success, which is defined by $P_{suc} = 1 - P_{err}$. One has

$$\begin{aligned} P_{suc} &= 1 - P_{err} && (91.89)\\ &= P(\underline{\widehat{x}}^n = \underline{x}^n) && (91.90)\\ &= \sum_{\widehat{x}^n, m, x^n} \theta(\widehat{x}^n = x^n) P(\widehat{x}^n|m) P(m|x^n) P_{\underline{x}}(x^n) && (91.91)\\ &= \sum_{x^n} P_{\underline{x}}(x^n) \delta[x^n, \widehat{x}^n \circ m(x^n)] \ . && (91.92) \end{aligned}$$

Now it's time to decide what encoding and decoding functions we want to consider. Suppose A is a proper subset of $S_{\underline{x}}^n$. One can give each element of A an individual number (its index) from 1 to $|A|$. Assume, without loss of generality, that $0^n \notin A$. As we shall see, the following encoding and decoding functions are good enough:

$$m(x^n) = \begin{cases} \text{index of } x^n \text{ in } A & \text{, if } x^n \in A \\ 0 & \text{, if } x^n \notin A \end{cases} \ , \tag{91.93}$$

[1] Many authors (for instance, C&T) denote the encoding function $m(\cdot)$ by $f(\cdot)$ and the decoding function $\widehat{x}^n(\cdot)$ by $g(\cdot)$.

and

$$\widehat{x}^n(m) = \begin{cases} m^{-1}(m) & \text{, if } m \in Z_{1,|A|} \\ 0^n & \text{, if } m = 0 \end{cases} , \tag{91.94}$$

where the set A is given by either

$$A_{P_{\underline{x}}} = \left\{ x^n : R \geq \frac{1}{n} \ln \frac{1}{P_{\underline{x}}(x^n)} \right\} = \left\{ x^n : R \geq \sum_x P_{[x^n]}(x) \ln \frac{1}{P_{\underline{x}}(x)} \right\} , \tag{91.95}$$

or

$$A_{univ} = \left\{ x^n : R \geq H(P_{[x^n]}) \right\} = \left\{ x^n : R \geq \sum_x P_{[x^n]}(x) \ln \frac{1}{P_{[x^n]}(x)} \right\} \tag{91.96}$$

for some positive number R yet to be specified. These two interesting options for the set A can be considered simultaneously by defining

$$A = \left\{ x^n : R \geq \sum_x P_{[x^n]}(x) \ln \frac{1}{Q(x)} \right\} , \tag{91.97}$$

where

$$Q(x) = \begin{cases} P_{\underline{x}}(x) & \text{, source dependent coding} \\ P_{[x^n]}(x) & \text{, universal coding} \end{cases} . \tag{91.98}$$

In the case of source dependent coding, Q (and therefore the functions $m(\cdot)$ and $\widehat{x}^n(\cdot)$) depend on the source distribution $P_{\underline{x}}$. In the case of universal coding, Q is independent of the source.

Note that for this encoding and decoding functions,

$$\delta[x^n, \widehat{x}^n \circ m(x^n)] = \theta(x^n \in A) = \theta\left(R \geq \sum_x P_{[x^n]}(x) \ln \frac{1}{Q(x)} \right) \tag{91.99}$$

for all $x^n \in val(\underline{x})^n - \{0^n\}$ so

$$\begin{aligned} P_{suc} &= \sum_{x^n} P_{\underline{x}}(x^n) \theta\left(R \geq \sum_x P_{[x^n]}(x) \ln \frac{1}{Q(x)} \right) && (91.100) \\ &\sim \int \mathcal{D}P_{[x^n]} e^{n \sum_x P_{[x^n]}(x) \ln \frac{P_{\underline{x}}(x)}{P_{[x^n]}(x)}} \theta\left(R \geq \sum_x P_{[x^n]}(x) \ln \frac{1}{Q(x)} \right) && (91.101) \\ &\approx \theta(R \geq H(P_{\underline{x}})) . && (91.102) \end{aligned}$$

Eq.(91.102) follows because, as is easily proven, applying the method of steepest descent to the p-type integral yields a tilde point:

$$\widetilde{P}_{[x^n]}(x) = P_{\underline{x}}(x) \ . \tag{91.103}$$

As mentioned in the notation section, we define $R_{\underline{m}}$ by

$$R_{\underline{m}} = \frac{\ln N_{\underline{m}}}{n} \ . \tag{91.104}$$

So far, it's not clear what value to use for the constant R that appears in the definition of set A. In the next Claim, we will show that it must equal $R_{\underline{m}}$ for our arguments to be valid.

Claim 141

$$R = R_{\underline{m}} \tag{91.105}$$

for consistency of our arguments.

proof: We must have

$$\begin{aligned}
N_{\underline{m}} &= \sum_{x^n} \theta(x^n \in A) && (91.106)\\
&\sim \int \mathcal{D}P_{[x^n]} e^{n\sum_x P_{[x^n]}(x) \ln \frac{1}{P_{[x^n]}(x)}} \theta\left(R > \sum_x P_{[x^n]}(x) \ln \frac{1}{Q(x)}\right) && (91.107)\\
&\sim e^{nR} \int \mathcal{D}P_{[x^n]} e^{n\sum_x P_{[x^n]}(x) \ln \frac{Q(x)}{P_{[x^n]}(x)}} \theta\left(R > \sum_x P_{[x^n]}(x) \ln \frac{1}{Q(x)}\right) && (91.108)\\
&\sim e^{nR} \theta(R > H(P_{\underline{x}})) \ . && (91.109)
\end{aligned}$$

As long as $R > H(\underline{x})$, our approximations are valid and $N_{\underline{m}} = e^{nR}$.
QED

91.5 Channel Coding

We define a codebook $\mathcal{C}$ as an $N_{\underline{m}} \times n$ matrix given by $\mathcal{C} = \{x^n(m)\}_{\forall m} = x^n(\cdot)$ where $x^n(m) \in S^n_{\underline{x}}$ for all $m \in val(\underline{m})$.

We consider all channel coding protocols that can be described by the following CB net

, (91.110)

with

$$P(m) = \frac{1}{N_{\underline{m}}} \ , \tag{91.111}$$

$$P(x^n|m, \mathcal{C}) = \delta(x^n, x^n(m)) \ , \tag{91.112}$$

$$P(y^n|x^n) = \prod_j P(y_j|x_j) = e^{n \sum_{x,y} P_{[x^n,y^n]}(x,y) \ln P(y|x)} \ , \tag{91.113}$$

$$P(\mathcal{C}) = \text{to be specified} \ , \tag{91.114}$$

and

$$P(\widehat{m}|y^n, \mathcal{C}) = \text{to be specified} \ . \tag{91.115}$$

Assume that we are given a channel $\{P_{\underline{y}|\underline{x}}(y|x)\}_{\forall y} \in pd(val(\underline{y}))$ for all $x \in val(\underline{x})$. The encoding $P(\mathcal{C})$ and decoding $P(\widehat{m}|y^n, \mathcal{C})$ probability distributions are yet to be specified.

It's convenient to define the coding rate $R_{\underline{m}}$ by

$$R_{\underline{m}} = \frac{\ln N_{\underline{m}}}{n} \tag{91.116}$$

and the channel capacity C by

$$C = \max_{P_{\underline{x}}} H(\underline{y} : \underline{x}) \ . \tag{91.117}$$

Claim 142 *(Independence upper bound for mutual information of DMC) If $P(y^n|x^n) = \prod_{j=1}^n P(y_j|x_j)$ (this is what is called a discrete memoryless channel, DMC), then*

$$H(\underline{y}^n : \underline{x}^n) \leq \sum_{j=0}^n H(\underline{y}_j : \underline{x}_j) \ . \tag{91.118}$$

Furthermore, equality holds iff the $\underline{x}_j$ are mutually independent.

proof: Assume $n = 3$ for illustrative purposes. If the $\underline{x}_j$ are not independent, we must consider the following CB net

. (91.119)

If the $\underline{x}_j$ are independent, then this becomes

(91.120)

In the case of Eq.(91.119),

$$H(\underline{y}^n : \underline{x}^n) = H(\underline{y}^n) - H(\underline{y}^n|\underline{x}^n) = H(\underline{y}^n) - \sum_j H(\underline{y}_j|\underline{x}_j) \tag{91.121}$$

$$\leq \sum_j H(\underline{y}_j) - \sum_j H(\underline{y}_j|\underline{x}_j) \tag{91.122}$$

$$= \sum_j H(\underline{y}_j : \underline{x}_j) \tag{91.123}$$

Eq.(91.122) follows from the "subadditivity" or "independence upper bound" of the joint entropy, which says that $H(\underline{a}, \underline{b}) \leq H(\underline{a}) + H(\underline{b})$ for any random variables $\underline{a}$ and $\underline{b}$. (See C&T for a proof of subadditivity). If the $\underline{x}_j$ are mutually independent, then the $\underline{y}_j$ must be mutually independent too, in which case Eq.(91.122) becomes an equality. Conversely, if Eq.(91.122) is an equality, then the $\underline{y}_j$ must be mutually independent so the $\underline{x}_j$ must be too.
QED

Claim 143 *Optimality: $\forall R_{\underline{m}}$, if $\exists$ an encoding and a decoding that satisfy $\lim_{n\to\infty} P_{err} = 0$ for the CB net of Eq.(91.110), then $R_{\underline{m}} \leq C$.*

proof:

$$nR_{\underline{m}} = \ln N_{\underline{m}} = H(\underline{m}) = H(\underline{y}^n : \underline{m}) + H(\underline{m}|\underline{y}^n) \tag{91.124}$$

$$\leq H(\underline{y}^n : \underline{m}) + n\delta \tag{91.125}$$

$$\leq H(\underline{y}^n : \underline{x}^n) + n\delta \tag{91.126}$$

$$\leq \sum_{j=1}^{n} H(\underline{y}_j : \underline{x}_j) + n\delta \tag{91.127}$$

$$\leq n(C + \delta) \tag{91.128}$$

(91.125): This follows from Fano's inequality. (See C&T for a proof of Fano's inequality.) δ is some positive number that tends to zero as $n \to \infty$

(91.126): This follows from the data processing inequalities. (See C&T for a proof of the data processing inequalities.)

(91.127): This follows from Claim 142.

(91.128): This follows from the definition of channel capacity C.

QED

Claim 144 *Achievability: $\forall R_{\underline{m}}$, if $R_{\underline{m}} \leq C$, then $\exists$ an encoding and a decoding that satisfy $\lim_{n\to\infty} P_{err} = 0$ for the CB net of Eq.(91.110).*

proof: So far, the encoding and decoding probability distributions are unspecified. In this proof, we will use one possible choice for these distributions. This choice, although not very practical, turns out to yield optimal results. For $P(\mathcal{C})$ we choose what is called random coding:

$$P(\mathcal{C}) = P_{\underline{x}}(x^n(\cdot)) = \prod_m P_{\underline{x}}(x^n(m)) = \prod_{m,j} P_{\underline{x}}(x_j(m)) \tag{91.129}$$

for some source $P_{\underline{x}} \in pd(val(\underline{x}))$. For $P(\widehat{m}|y^n, \mathcal{C})$ we choose a maximum likelihood decoder:[2]

$$P(\widehat{m}|y^n, \mathcal{C}) = \prod_{m \neq \widehat{m}} \theta\left(R < \frac{1}{n} \ln \frac{P(y^n|x^n(\widehat{m}))}{P(y^n|x^n(m))}\right) \tag{91.130}$$

$$= \prod_{m \neq \widehat{m}} \theta\left(R < \frac{1}{n} \ln \frac{P(y^n : x^n(\widehat{m}))}{P(y^n : x^n(m))}\right) \tag{91.131}$$

for some $R > 0$. Note that there is no guarantee that this definition of $P(\widehat{m}|y^n, \mathcal{C})$ is a well defined probability distribution satisfying $\sum_{\widehat{m}} P(\widehat{m}|y^n, \mathcal{C}) = 1$. In the next Claim, we will prove that if $R = R_{\underline{m}}$, then $P(\widehat{m}|y^n, \mathcal{C})$ is well defined.

The probability of error is defined by

$$P_{err} = P(\underline{\widehat{m}} \neq \underline{m}) \ . \tag{91.132}$$

We find it more convenient to work with the probability of success, which is defined by $P_{suc} = 1 - P_{err}$. One has

[2] By $\prod_{m \neq \widehat{m}}$ we mean $\prod_{m \in val(\underline{m}) - \{\widehat{m}\}}$.

$$
\begin{aligned}
P_{suc} &= 1 - P_{err} && (91.133)\\
&= P(\underline{\widehat{m}} = \underline{m}) && (91.134)\\
&= \sum_{\widehat{m},m} \theta(\widehat{m} = m) P(\widehat{m}, m) && (91.135)\\
&= \sum_{\widehat{m},m,y^n,x^n,\mathcal{C}} \theta(\widehat{m} = m) P(\widehat{m}|y^n, \mathcal{C}) P(y^n|x^n) \delta(x^n, x^n(m)) P(m) P(\mathcal{C}) && (91.136)\\
&= \frac{1}{N_{\underline{m}}} \sum_{\widehat{m}} \sum_{\mathcal{C}} P(\mathcal{C}) \sum_{y^n} P(\widehat{m}|y^n, \mathcal{C}) P(y^n|x^n(\widehat{m})) \ . && (91.137)
\end{aligned}
$$

The choice of $\widehat{m} \in val(\underline{m})$ does not matter. Any choice would give the same answer for P_{suc}

$$
\frac{1}{N_{\underline{m}}} \sum_{\widehat{m}} \sum_{\mathcal{C}} P(\mathcal{C}) = \sum_{\mathcal{C}} P(\mathcal{C}) = E_{\mathcal{C}} \ . \tag{91.138}
$$

Thus

$$
P_{suc} = E_{\mathcal{C}} \sum_{y^n} P(y^n|x^n(\widehat{m})) \prod_{m \neq \widehat{m}} \theta \left(R < \frac{1}{n} \ln \frac{P(y^n : x^n(\widehat{m}))}{P(y^n : x^n(m))} \right) \ . \tag{91.139}
$$

Let

$$
\oint_{k(\cdot)} = \prod_{m \neq \widehat{m}} \left\{ \int_{-\infty}^{+\infty} \frac{dk(m)}{2\pi i} \ \frac{1}{(k(m) - i\epsilon)} \right\} \ , \tag{91.140}
$$

and

$$
K = \sum_{m \neq \widehat{m}} k(m) \ . \tag{91.141}
$$

Expressing the θ functions in Eq.(91.139) as integrals (see Eq.(91.12)), we get

$$
P_{suc} = \oint_{k(\cdot)} e^{-iKR} \sum_{y^n, x^n(\cdot)} \exp \left(n \sum_{y \in val(\underline{y}) \ , \ x(\cdot) \in val(\underline{x})^{N_{\underline{m}}}} P_{[\,]}(y, x(\cdot)) \ln Z(y, x(\cdot)) \right) \ , \tag{91.142}
$$

where

$$
Z(y, x(\cdot)) = P(y|x(\widehat{m})) \prod_{m} \left\{ P_{\underline{x}}(x(m)) \right\} \prod_{m \neq \widehat{m}} \left\{ \frac{P^{i\frac{k(m)}{n}}(y : x(\widehat{m}))}{P^{i\frac{k(m)}{n}}(y : x(m))} \right\} \ . \tag{91.143}
$$

Next we express the sum over $y^n, x^n(\cdot)$ as a p-type integral to get

$$P_{suc} = \oint_{k(\cdot)} e^{-iKR} \int \mathcal{D}P_{[\,]} n^{N_{\underline{y}}+N_{\underline{x}}N_{\underline{m}}-1} (d_{[y^n,x^n(\cdot)]})_{H=0} e^{\mathcal{L}_0} \ , \tag{91.144}$$

where

$$\mathcal{L}_0 = n \sum_{y,x(\cdot)} P_{[\,]}(y,x(\cdot)) \ln \frac{Z(y,x(\cdot))}{P_{[\,]}(y,x(\cdot))} \ . \tag{91.145}$$

We add to $\mathcal{L}_0$ a Lagrange multiplier term that constrains the components of the vector $\{P_{[\,]}(y,x(\cdot))\}_{\forall y,x(\cdot)}$ so that they sum to one:

$$\mathcal{L} = \mathcal{L}_\lambda = \mathcal{L}_0 + n\lambda \left(\sum_{y,x(\cdot)} P_{[\,]}(y,x(\cdot)) - 1 \right) \tag{91.146}$$

for any $\lambda \in \mathbb{R}$. It's easy to check that $\mathcal{L}$ is maximized when

$$\widetilde{P}_{[\,]}(y,x(\cdot)) = \frac{Z(y,x(\cdot))}{\sum_{y,x(\cdot)} Z(y,x(\cdot))} \ . \tag{91.147}$$

Evaluating the integrand of the p-type integral in Eq.(91.144) at this tilde point yields

$$P_{suc} = \oint_{k(\cdot)} e^{-iKR} e^{n \ln Z} \ , \tag{91.148}$$

where

$$Z = \sum_{y,x(\cdot)} Z(y,x(\cdot)) \ . \tag{91.149}$$

Using the shorthand notations

$$E_y = \sum_y P(y), \ E_{x(m)} = \sum_{x(m)} P_{\underline{x}}(x(m)) \ , \tag{91.150}$$

Z can be expressed as

$$Z = E_y \left[E_{x(\widehat{m})}[P^{1+i\frac{K}{n}}(y:x(\widehat{m}))] \prod_{m \neq \widehat{m}} \left\{ E_{x(m)}[P^{-i\frac{k(m)}{n}}(y:x(m))] \right\} \right] \ . \tag{91.151}$$

Define

$$Z_0 = [Z]_{k(m)=0\ \forall m} = E_y E_{x(\widehat{m})}[P^{1+i\frac{K}{n}}(y:x(\widehat{m}))] \ . \tag{91.152}$$

Note that 1 equals

$$1 = \int_{-\infty}^{+\infty} dK \ \delta(\sum_{m \neq \widehat{m}} \{k(m)\} - K) \tag{91.153}$$

$$= \int_{-\infty}^{+\infty} dK \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \ e^{ih\left(\sum_{m \neq \widehat{m}} \{k(m)\} - K\right)} \ . \tag{91.154}$$

Multiplying P_{suc} by 1 certainly doesn't change it. Thus the right hand sides of Eqs.(91.148) and (91.154) can be multiplied to get

$$P_{suc} = \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \int_{-\infty}^{+\infty} dK \; e^{iK(-h-R)} \oint_{k(\cdot)} e^{ih\sum_{m\neq\widehat{m}} k(m)} e^{n\ln Z} \; . \tag{91.155}$$

Next we will assume that, for all m, when doing the contour integration over $k(m)$ in Eq.(91.155) with Z given by Eq.(91.151), the $e^{n\ln Z}$ can be evaluated at the value $k(m) = i\epsilon \to 0$ of the pole.[3] Symbolically, this means we assume

$$\begin{aligned} \oint_{k(\cdot)} e^{ih\sum_{m\neq\widehat{m}} k(m)} e^{n\ln Z} &= e^{n\ln Z_0} \oint_{k(\cdot)} e^{ih\sum_{m\neq\widehat{m}} k(m)} && (91.156) \\ &= e^{n\ln Z_0}\theta(h>0) \; . && (91.157) \end{aligned}$$

Applying Eq.(91.157) to Eq.(91.155) gives

$$P_{suc} = \int_{-\infty}^{+\infty} \frac{dh}{2\pi}\theta(h>0) \int_{-\infty}^{+\infty} dK \; e^{iK(-h-R)} e^{n\ln Z_0} \; . \tag{91.158}$$

Next we use Eqs.(91.15) and (91.16) to expand $\ln Z_0$ to second order in K. This yields

$$\ln Z_0 \approx i\frac{K}{n}a - \frac{K^2}{2n^2}b \; , \tag{91.159}$$

where

$$a = H(\underline{y} : \underline{x}) \; , \tag{91.160}$$

and

$$\begin{aligned} b &= E_y E_x P(y:x) \ln^2 P(y:x) - H^2(\underline{y} : \underline{x}) && (91.161) \\ &= E_{y,x} \ln^2 P(y:x) - [E_{y,x} \ln P(y:x)]^2 && (91.162) \\ &\geq 0 && (91.163) \end{aligned}$$

(The inequality follows from the identity $\langle \underline{x}^2 \rangle - \langle \underline{x} \rangle^2 = \langle (\underline{x} - \langle \underline{x} \rangle)^2 \rangle$ where $\langle \cdot \rangle$ denotes an average and $\underline{x}$ is any random variable.)

With the $\ln Z_0$ expanded to second order in K, Eq.(91.158) becomes

$$P_{suc} = \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \; \theta(h>0) \int_{-\infty}^{+\infty} dK \; e^{iK(a-h-R) - \frac{K^2}{2n}b} \; . \tag{91.164}$$

[3] I don't know how to prove this assumption rigorously. The assumption is plausible, and it does lead to the correct result for the channel capacity. It may just be an approximation that becomes increasingly good as $n \to \infty$

If we keep only the term linear in K in the argument of the exponential, we immediately get

$$P_{suc} = \theta(R < H(\underline{y} : \underline{x})) \ . \tag{91.165}$$

If we also keep the term quadratic in K, we get

$$P_{suc} = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{n}{2b}}[R - H(\underline{y} : \underline{x})]\right) \ . \tag{91.166}$$

Maximizing both sides of Eq.(91.165) with respect to the source $P_{\underline{x}}$, and using the definition of channel capacity C, we get that there is an encoding and a decoding for which

$$P_{suc} = \theta(R < C) \ . \tag{91.167}$$

QED

Claim 145

$$R = R_{\underline{m}} \tag{91.168}$$

for consistency of our arguments.

proof: Rather than checking that $\sum_{\widehat{m}} P(\widehat{m}|y^n, \mathcal{C}) = 1$, we will check that the total probability distribution for the whole CB net Eq.(91.110) sums to one. We want

$$1 = \sum_{\widehat{m}, m, y^n, x^n, \mathcal{C}} P(\widehat{m}|y^n, \mathcal{C}) P(y^n|x^n) \delta(x^n, x^n(m)) P(m) P(\mathcal{C}) \ . \tag{91.169}$$

Using

$$\sum_{\widehat{m}, m} = \sum_{\widehat{m}, m} \theta(\widehat{m} = m) + \sum_{\widehat{m}, m} \theta(\widehat{m} \neq m) \ , \tag{91.170}$$

and

$$\sum_{\widehat{m}, m} \theta(\widehat{m} \neq m) P(m) \sum_{\mathcal{C}} P(\mathcal{C}) = \frac{(N_{\underline{m}}^2 - N_{\underline{m}})}{N_{\underline{m}}} \sum_{\mathcal{C}} P(\mathcal{C}) \approx N_{\underline{m}} E_{\mathcal{C}} \ , \tag{91.171}$$

we get for any pair $m_0, \widehat{m} \in val(\underline{m})$ such that $m_0 \neq \widehat{m}$,

$$1 = P_{suc} + N_{\underline{m}} E_{\mathcal{C}} \sum_{y^n} P(\widehat{m}|y^n, \mathcal{C}) P(y^n|x^n(m_0)) \ . \tag{91.172}$$

Substituting into Eq.(91.172) the specific values of the probability distributions $P(\widehat{m}|y^n, \mathcal{C})$ and $P(y^n|x^n(m_0))$, we get

$$P_{err} = N_{\underline{m}} \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \int_{-\infty}^{+\infty} dK \; e^{iK(-h-R)} \oint_{k(\cdot)} e^{ih \sum_{m \neq \widehat{m}} k(m)} e^{n \ln W} \; , \tag{91.173}$$

where $\oint_{k(\cdot)}$ is defined as before (see Eq.(91.140)) and where

$$W = E_y \left[\begin{array}{l} E_{x(\widehat{m})}[P^{i\frac{K}{n}}(y : x(\widehat{m}))] \\ E_{x(m_0)}[P^{1-i\frac{k(m_0)}{n}}(y : x(m_0))] \\ \prod_{m \neq \widehat{m}, m_0} \left\{ E_{x(m)}[P^{-i\frac{k(m)}{n}}(y : x(m))] \right\} \end{array} \right] . \tag{91.174}$$

Let

$$W_0 = [W]_{k(m)=0 \; \forall m} = E_y E_{x(\widehat{m})}[P^{i\frac{K}{n}}(y : x(\widehat{m}))] \; . \tag{91.175}$$

Next assume that

$$\oint_{k(\cdot)} e^{ih \sum_{m \neq \widehat{m}} k(m)} e^{n \ln W} = e^{n \ln W_0} \oint_{k(\cdot)} e^{ih \sum_{m \neq \widehat{m}} k(m)} \tag{91.176}$$

$$= e^{n \ln W_0} \theta(h > 0) \; . \tag{91.177}$$

Applying Eq.(91.177) to Eq.(91.173) yields

$$P_{err} = N_{\underline{m}} \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \theta(h > 0) \int_{-\infty}^{+\infty} dK \; e^{iK(-h-R)} e^{n \ln W_0} \; . \tag{91.178}$$

Now we can make the following change of variables

$$K \to K - in \; . \tag{91.179}$$

Note that this change of variables changes W_0 defined by Eq.(91.175) to Z_0 defined by Eq.(91.152). Under this change of variables, Eq.(91.178) becomes

$$P_{err} = N_{\underline{m}} \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \theta(h > 0) e^{n(-h-R)} \int_{-\infty}^{+\infty} dK \; e^{iK(-h-R)} e^{n \ln Z_0} \tag{91.180}$$

$$\approx N_{\underline{m}} e^{-nR} P_{suc} \; , \tag{91.181}$$

or, equivalently,

$$\theta(R > H(\underline{y} : \underline{x})) \approx N_{\underline{m}} e^{-nR} \theta(R < H(\underline{y} : \underline{x})) \; . \tag{91.182}$$

Thus, when R equals (or is very close to) $H(\underline{y} : \underline{x})$, we must have $N_{\underline{m}} = e^{nR}$.
QED

91.6 Source Coding With Distortion

Assume that we are given a function $d(x,y)$ that measures the distance between two letters of $x,y \in val(\underline{x})$. Assume $d(x,x)=0$ and $d(x,y) \geq 0$ for all $x,y \in val(\underline{x})$.

Assume that random variables $\underline{x}$ and $\widehat{\underline{x}}$ both have the same set of possible values $val(\underline{x})$. We define codebook $\mathcal{C}$ as an $N_{\underline{m}} \times n$ matrix given by $\mathcal{C} = \{x^n(m)\}_{\forall m} = x^n(\cdot)$ where $x^n(m) \in S^n_{\underline{x}}$ for all $m \in val(\underline{m})$. We define another codebook $\widehat{\mathcal{C}}$ as an $N_{\underline{m}} \times n$ matrix given by $\widehat{\mathcal{C}} = \{\widehat{x}^n(m)\}_{\forall m} = \widehat{x}^n(\cdot)$ where $\widehat{x}^n(m) \in S^n_{\underline{x}}$ for all $m \in val(\underline{m})$.

We consider all source coding protocols that can be described by the following CB net:

(91.183)

with $val(\underline{x}) = val(\widehat{\underline{x}})$ and

$$P(x^n) = \prod_{j=1}^{n} P_{\underline{x}}(x_j) \ , \tag{91.184}$$

$$P(m|x^n, \widehat{\mathcal{C}}) = \text{to be specified} \ , \tag{91.185}$$

$$P(\mathcal{C}) = \text{to be specified} \ , \tag{91.186}$$

$$P(\widehat{\mathcal{C}}|\mathcal{C}) = \prod_{m} P_{\widehat{\underline{x}}|\underline{x}}(\widehat{x}^n(m)|x^n(m)) = \prod_{m,j} P_{\widehat{\underline{x}}|\underline{x}}(\widehat{x}_j(m)|x_j(m)) \ , \tag{91.187}$$

and

$$P(\widehat{x}^n|m, \widehat{\mathcal{C}}) = \delta(\widehat{x}^n, \widehat{x}^n(m)) \ . \tag{91.188}$$

Assume that we are given a source $\{P_{\underline{x}}(x)\}_{\forall x} \in pd(val(\underline{x}))$ and a channel $\{P_{\widehat{\underline{x}}|\underline{x}}(\widehat{x}|x)\}_{\forall \widehat{x} \in val(\underline{x})} \in pd(val(\underline{x}))$ for all $x \in val(\underline{x})$. The encoding $P(m|x^n, \widehat{\mathcal{C}})$ and decoding $P(\mathcal{C})$ probability distributions are yet to be specified.

Henceforth, we will use the following shorthand notations

$$E_j = \frac{1}{n} \sum_{j=1}^{n} \ , \quad E_{\widehat{x},x} = \sum_{\widehat{x},x} P_{\widehat{\underline{x}}|\underline{x}}(\widehat{x}|x) P_{\underline{x}}(x) \ . \tag{91.189}$$

As usual, we define the **rate of** $\underline{m}$ by $R_{\underline{m}} = \ln(N_{\underline{m}})/n$. We define the probability of success by

$$P_{suc} = P[E_j d(\widehat{\underline{x}}_j, \underline{x}_j) \leq D] \tag{91.190}$$

where $D \in \mathbb{R}^{>0}$ is called the **distortion**. Note that when $D = 0$, $P_{suc} = P(\underline{\widehat{x}}^n = \underline{x}^n)$, which is what we used previously when we considered source coding without distortion.

For any source $P_{\underline{x}}$ and distortion D, it is useful to define a **rate distortion function** $H_{\underline{x}}(D)$ by

$$H_{\underline{x}}(D) = \min_{P_{\underline{\widehat{x}}|\underline{x}}: E_{\widehat{x},x} d(\widehat{x},x) < D} H_{P_{\underline{\widehat{x}}|\underline{x}} P_{\underline{x}}}(\underline{\widehat{x}} : \underline{x}) \ . \tag{91.191}$$

Claim 146 *(Properties of $H_{\underline{x}}(D)$)*

(a) $H_{\underline{x}}(D)$ is a monotonically non-increasing, convex function of D.

(b) $H_{\underline{x}}(0) = H(\underline{x})$

(c) $H_{\underline{x}}(E^Q_{\widehat{x},x} d(\widehat{x},x)) \leq H_Q(\underline{\widehat{x}} : \underline{x})$, where $E^Q_{\widehat{x},x} = \sum_{\widehat{x},x} Q(\widehat{x},x)$, where $\{Q(\widehat{x},x)\}_{\forall \widehat{x},x} \in pd(val(\underline{\widehat{x}}, \underline{x}))$ such that $\sum_{\widehat{x}} Q(\widehat{x},x) = P_{\underline{x}}(x)$ for all x.

proof:

proof of (a): Monotonicity is obvious. To prove convexity, recall (see C&T for a proof) that the mutual information is a convex function of its joint probability. This means that for any $\lambda \in [0,1]$ and $P_1, P_0 \in pd(val(\underline{\widehat{x}}, \underline{x}))$, if

$$P_\lambda(\widehat{x}, x) = \lambda P_1(\widehat{x}, x) + (1-\lambda) P_0(\widehat{x}, x) \tag{91.192}$$

for all $\widehat{x}, x$, then

$$H_{P_\lambda}(\underline{\widehat{x}} : \underline{x}) \leq \lambda H_{P_1}(\underline{\widehat{x}} : \underline{x}) + (1-\lambda) H_{P_0}(\underline{\widehat{x}} : \underline{x}) \ . \tag{91.193}$$

For any $\lambda \in [0,1]$, let $D_0, D_1 \in \mathbb{R}^{\geq 0}$ and

$$D_\lambda = \lambda D_1 + (1-\lambda) D_0 \ . \tag{91.194}$$

Suppose $P_0, P_1 \in pd(val(\underline{\widehat{x}}, \underline{x}))$ such that $\sum_{\widehat{x}} P_j(\widehat{x}, x) = P_{\underline{x}}(x)$ for all x and

$$H_{\underline{x}}(D_j) = H_{P_j}(\underline{\widehat{x}} : \underline{x}) \tag{91.195}$$

for $j = 0, 1$. Define P_λ by Eq.(91.192). Then

$$\begin{aligned} H_{\underline{x}}(D_\lambda) &\leq H_{P_\lambda}(\underline{\widehat{x}} : \underline{x}) && (91.196) \\ &\leq \lambda H_{P_1}(\underline{\widehat{x}} : \underline{x}) + (1-\lambda) H_{P_0}(\underline{\widehat{x}} : \underline{x}) && (91.197) \\ &= \lambda H_{\underline{x}}(D_1) + (1-\lambda) H_{\underline{x}}(D_0) \ . && (91.198) \end{aligned}$$

proof of (b): If $D = 0$, then $P(\widehat{x}|x) = \delta^{\widehat{x}}_x$ so $H(\underline{\widehat{x}} : \underline{x}) = H(\underline{x})$.

proof of (c): This follows from definition of $H_{\underline{x}}(D)$.

QED

Claim 147 *Optimality: $\forall(D, R_{\underline{m}})$, if $\exists$ an encoding and a decoding that satisfy $\lim_{n\to\infty} P_{err} = 0$ for the CB net of Eq.(91.183), then $R_{\underline{m}} \geq H_{\underline{x}}(D)$.*

proof:

$$
\begin{aligned}
nR_{\underline{m}} &= \ln N_{\underline{m}} = H(\underline{m}) = H(\underline{\widehat{x}}^n : \underline{m}) + H(\underline{m}|\underline{\widehat{x}}^n) && (91.199)\\
&\geq H(\underline{\widehat{x}}^n : \underline{m}) && (91.200)\\
&\geq H(\underline{\widehat{x}}^n : \underline{x}^n) && (91.201)\\
&= \sum_j H(\underline{\widehat{x}}_j : \underline{x}_j) && (91.202)\\
&\geq \sum_j H_{\underline{x}}\left(E_{\widehat{x}_j,x_j} d(\widehat{x}_j, x_j)\right) && (91.203)\\
&\geq nH_{\underline{x}}\left(\frac{1}{n}\sum_j E_{\widehat{x}_j,x_j} d(\widehat{x}_j, x_j)\right) && (91.204)\\
&= nH_{\underline{x}}\left(E_{\widehat{x},x} d(\widehat{x}, x)\right) && (91.205)\\
&\geq nH_{\underline{x}}(D) && (91.206)
\end{aligned}
$$

(91.201): This follows from the data processing inequalities. (See C&T for a proof of the data processing inequalities.)

(91.202): This follows from Claim 142 in the case of equality. We are assuming that $P(\widehat{\mathcal{C}}|\mathcal{C})$ is a DMC, and that $P(\mathcal{C})$ is an i.i.d. source. This forces $(\widehat{x}_j(m), x_j(m))$ and $(\widehat{x}_{j'}(m), x_{j'}(m))$ with $j \neq j'$ to be independent.

(91.203): This follows from Claim 146, part (c).

(91.204): This follows because $H_{\underline{x}}(D)$ is a convex function of D.

(91.205): This follows from using $P_{[\,]}(\widehat{x}, x) \to P(\widehat{x}, x)$.

(91.206): Eq.(91.190) is the definition of D. Expressing Eq.(91.190) in terms of p-types and using $P_{[\,]}(\widehat{x}, x) \to P(\widehat{x}, x)$, we find that $E_{\widehat{x},x} d(\widehat{x}, x) < D$ is necessary for success. Then use the fact that $H_{\underline{x}}(D)$ is non-increasing.

QED

Claim 148 *Achievability: $\forall(D, R_{\underline{m}})$, if $R_{\underline{m}} \geq H_{\underline{x}}(D)$, then $\exists$ an encoding and a decoding that satisfy $\lim_{n\to\infty} P_{err} = 0$ for the CB net of Eq.(91.183).*

proof: So far, the encoding and decoding probability distributions are unspecified. In this proof, we will use one possible choice for these distributions. For decoder $P(\mathcal{C})$ we choose:

$$P(\mathcal{C}) = P_{\underline{x}}(x^n(\cdot)) = \prod_m \left\{P_{\underline{x}}(x^n(m))\right\} = \prod_{m,j} \left\{P_{\underline{x}}(x_j(m))\right\} \ , \tag{91.207}$$

and for encoder $P(m|x^n, \widehat{\mathcal{C}})$ we choose:

$$\begin{aligned} P(m|x^n, \widehat{\mathcal{C}}) &= \prod_{m' \neq m} \theta\left(R > \frac{1}{n} \ln \frac{P(x^n|\widehat{x}^n(m))}{P(x^n|\widehat{x}^n(m'))}\right) & (91.208) \\ &= \prod_{m' \neq m} \theta\left(R > \frac{1}{n} \ln \frac{P(x^n : \widehat{x}^n(m))}{P(x^n : \widehat{x}^n(m'))}\right) & (91.209) \end{aligned}$$

for some $R > 0$. Note that there is no guarantee that this definition of $P(m|x^n, \widehat{\mathcal{C}})$ is a well defined probability distribution satisfying $\sum_m P(m|x^n, \widehat{\mathcal{C}}) = 1$. In the next Claim, we will prove that if $R = R_{\underline{m}}$, then $P(m|x^n, \widehat{\mathcal{C}})$ is well defined.

Let

$$P(\widehat{\mathcal{C}}) = \sum_{\mathcal{C}} P(\widehat{\mathcal{C}}|\mathcal{C}) P(\mathcal{C}) \ . \tag{91.210}$$

One has

$$\begin{aligned} P_{suc} &= P[E_j d(\underline{\widehat{x}}_j, \underline{x}_j) < D] & (91.211) \\ &= \sum_{\widehat{x}^n, x^n} P(\widehat{x}^n, x^n) \theta(E_j d(\widehat{x}_j, x_j) < D) & (91.212) \\ &= \sum_{\widehat{x}^n, x^n, m, \widehat{\mathcal{C}}} P(\widehat{x}^n|m, \widehat{\mathcal{C}}) P(m|x^n, \widehat{\mathcal{C}}) P(x^n) P(\widehat{\mathcal{C}}) \theta(E_j d(\widehat{x}_j, x_j) < D) & (91.213) \\ &= \sum_m E_{\widehat{\mathcal{C}}} E_{x^n} P(m|x^n, \widehat{\mathcal{C}}) \theta(E_j d(\widehat{x}_j(m), x_j) < D) \ . & (91.214) \end{aligned}$$

Consider what happens to $P(m|x^n, \widehat{\mathcal{C}})$ in Eq.(91.214) as $D \to 0$. When $D \to 0$, $\widehat{x}^n(m) \to x^n$ by virtue of Eq.(91.214). Hence $P(x^n|\widehat{x}^n(m)) \to 1$. Furthermore, $P(x^n|\widehat{x}^n(m')) \to P(x^n(m)|\widehat{x}^n(m')) = P(x^n(m))\delta_m^{m'} = P(x^n)\delta_m^{m'}$. Thus

$$P(m|x^n, \widehat{\mathcal{C}}) \to \theta\left(R > \frac{1}{n} \ln \frac{1}{P(x^n)}\right) = \theta(x^n \in A_{P_{\underline{x}}}) \ . \tag{91.215}$$

Hence, when $D = 0$, the encoder $P(m|x^n, \widehat{\mathcal{C}})$ in Eq.(91.214) is the same as the one we used when we considered source coding without distortion.

For any $Q \in pd(val(\underline{\widehat{x}}, \underline{x}))$ such that $\sum_{\widehat{x}} Q(\widehat{x}, x) = P_{\underline{x}}(x)$ for all x, define

$$\theta_{Q(\widehat{x},x)} = \theta_{Q_{\underline{\widehat{x}},\underline{x}}} = \theta(\sum_{\widehat{x},x} Q(\widehat{x}, x) d(\widehat{x}, x) < D) \ . \tag{91.216}$$

Note that

$$\theta(E_j d(\widehat{x}_j(1), x_j) < D) = \theta_{P_{[\,]}(\widehat{x}(1),x)} \ . \tag{91.217}$$

Note that

$$\sum_m E_{\widehat{\mathcal{C}}} = N_{\underline{m}} E_{\widehat{\mathcal{C}}} \ . \tag{91.218}$$

Hence, the choice of $m \in val(\underline{m})$ in Eq.(91.214) does not matter. Any choice would give the same answer for P_{suc}. Thus, Eq.(91.214) can be replaced by the following. Assume $1 \in val(\underline{m})$ and replace m by 1 and m' by m. Also use Eq.(91.217). Then

$$P_{suc} \quad = \quad N_{\underline{m}} E_{\widehat{\mathcal{C}}} E_{x^n} \prod_{m \neq 1} \left\{ \theta \left(R > \frac{1}{n} \ln \frac{P(x^n : \widehat{x}^n(1))}{P(x^n : \widehat{x}^n(m))} \right) \right\} \theta_{P_{[\,]}(\widehat{x}(1),x)} \ . \tag{91.219}$$

If we assume that our formalism will eventually justify the physically plausible assumption that $P_{[\,]}(\widehat{x}(1), x) \to P_{\widehat{\underline{x}},\underline{x}}(\widehat{x}(1), x)$, then we may replace $\theta_{P_{[\,]}(\widehat{x}(1),x)}$ by $\theta_{P_{\widehat{\underline{x}},\underline{x}}}$ at this point. This would simplify the analysis below. Instead, we will continue with $\theta_{P_{[\,]}(\widehat{x}(1),x)}$ and show that our formalism does indeed lead to the same result as if we had replaced $\theta_{P_{[\,]}(\widehat{x}(1),x)}$ by $\theta_{P_{\widehat{\underline{x}},\underline{x}}}$ at this point.

Let

$$\oint_{k(\cdot)} = \prod_{m \neq 1} \left\{ \int_{-\infty}^{+\infty} \frac{dk(m)}{2\pi i} \ \frac{1}{(k(m) - i\epsilon)} \right\} \ , \tag{91.220}$$

and

$$K = \sum_{m \neq 1} k(m) \ . \tag{91.221}$$

Expressing the θ functions in Eq.(91.219) as integrals (see Eq.(91.12)), we get

$$P_{suc} = N_{\underline{m}} \oint_{k(\cdot)} e^{iKR} \sum_{\widehat{x}^n(\cdot), x^n} \exp \left(n \sum_{\widehat{x}(\cdot) \in val(\underline{x})^{N_{\underline{m}}} \ , \ x \in val(\underline{x})} P_{[\,]}(\widehat{x}(\cdot), x) \ln Z(\widehat{x}(\cdot), x) \right) \theta_{P_{[\,]}(\widehat{x}(1),x)} \ , \tag{91.222}$$

where

$$Z(\widehat{x}(\cdot), x) = P(x) \prod_m \{P(\widehat{x}(m))\} \prod_{m \neq 1} \left\{ \frac{P^{-i\frac{k(m)}{n}}(x : \widehat{x}(1))}{P^{-i\frac{k(m)}{n}}(x : \widehat{x}(m))} \right\} \ . \tag{91.223}$$

Next we express the sum over $\widehat{x}^n(\cdot), x^n$ as a p-type integral to get

$$P_{suc} = N_{\underline{m}} \oint_{k(\cdot)} e^{iKR} \int \mathcal{D}P_{[\,]} n^{N_{\underline{x}}(N_{\underline{m}}+1)-1} (d_{[\widehat{x}^n(\cdot), x^n]})_{H=0} e^{\mathcal{L}_0} \theta_{P_{[\,]}(\widehat{x}(1),x)} \ , \tag{91.224}$$

where

$$\mathcal{L}_0 = n \sum_{\widehat{x}(\cdot),x} P_{[\,]}(\widehat{x}(\cdot),x) \ln \frac{Z(\widehat{x}(\cdot),x)}{P_{[\,]}(\widehat{x}(\cdot),x)} \ . \tag{91.225}$$

We add to $\mathcal{L}_0$ a Lagrange multiplier term that constrains the components of the vector $\{P_{[\,]}(\widehat{x}(\cdot),x)\}_{\forall \widehat{x}(\cdot),x}$ so that they sum to one:

$$\mathcal{L} = \mathcal{L}_\lambda = \mathcal{L}_0 + n\lambda \left(\sum_{\widehat{x}(\cdot),x} P_{[\,]}(\widehat{x}(\cdot),x) - 1 \right) \tag{91.226}$$

for any $\lambda \in \mathbb{R}$. It's easy to check that $\mathcal{L}$ is maximized when

$$\widetilde{P}_{[\,]}(\widehat{x}(\cdot),x) = \frac{Z(\widehat{x}(\cdot),x)}{\sum_{\widehat{x}(\cdot),x} Z(\widehat{x}(\cdot),x)} \ . \tag{91.227}$$

Evaluating the integrand of the p-type integral in Eq.(91.224) at this tilde point yields

$$P_{suc} = N_{\underline{m}} \oint_{k(\cdot)} e^{iKR} e^{n \ln Z} \theta_{\widetilde{P}_{[\,]}(\widehat{x}(1),x)} \tag{91.228}$$

where

$$Z = \sum_{\widehat{x}(\cdot),x} Z(\widehat{x}(\cdot),x) \ . \tag{91.229}$$

Z can be expressed as

$$Z = E_x \left[E_{\widehat{x}(1)}[P^{-i\frac{K}{n}}(\widehat{x}(1):x)] \prod_{m\neq 1} \left\{ E_{\widehat{x}(m)}[P^{i\frac{k(m)}{n}}(\widehat{x}(m):x)] \right\} \right] \ . \tag{91.230}$$

Define

$$Z_0 = [Z]_{k(m)=0\,\forall m} = E_x E_{\widehat{x}(1)}[P^{-i\frac{K}{n}}(\widehat{x}(1):x)] \ . \tag{91.231}$$

Note that 1 equals

$$1 = \int_{-\infty}^{+\infty} dK \ \delta(\sum_{m\neq 1} \{k(m)\} - K) \tag{91.232}$$

$$= \int_{-\infty}^{+\infty} dK \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \ e^{ih\left(\sum_{m\neq 1}\{k(m)\} - K\right)} \ . \tag{91.233}$$

Multiplying P_{suc} by 1 certainly doesn't change it. Thus the right hand sides of Eqs.(91.228) and (91.233) can be multiplied to get

$$P_{suc} = N_{\underline{m}} \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \int_{-\infty}^{+\infty} dK \; e^{iK(-h+R)} \oint_{k(\cdot)} e^{ih\sum_{m\neq 1} k(m)} e^{n \ln Z} \theta_{\widetilde{P}_{[\,]}(\widehat{x}(1),x)} \; . \quad (91.234)$$

Next we will assume that, for all m, when doing the contour integration over $k(m)$ in Eq.(91.234) with Z given by Eq.(91.230), the $e^{n \ln Z} \theta_{\widetilde{P}_{[\,]}(\widehat{x}(1),x)}$ can be evaluated at the value $k(m) = i\epsilon \to 0$ of the pole.[4] Symbolically, this means we assume

$$\begin{aligned} \oint_{k(\cdot)} e^{ih\sum_{m\neq 1} k(m)} e^{n \ln Z} \theta_{\widetilde{P}_{[\,]}(\widehat{x}(1),x)} &= e^{n \ln Z_0} \theta_{P^{-i\frac{K}{n}}(\widehat{x}(1),x)} \oint_{k(\cdot)} e^{ih\sum_{m\neq 1} k(m)} \quad (91.235) \\ &= e^{n \ln Z_0} \theta_{P^{-i\frac{K}{n}}(\widehat{x}(1),x)} \theta(h > 0) \; . \quad (91.236) \end{aligned}$$

Applying Eq.(91.236) to Eq.(91.234) gives

$$P_{suc} = N_{\underline{m}} \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \theta(h > 0) \int_{-\infty}^{+\infty} dK \; e^{iK(-h+R)} e^{n \ln Z_0} \theta_{P^{-i\frac{K}{n}}(\widehat{x}(1),x)} \; . \quad (91.237)$$

Next we make the following change of variables:

$$K \to K + in \; . \quad (91.238)$$

Let

$$W_0 = [Z_0]_{K \to K+in} = E_x E_{\widehat{x}(1)} [P^{1-i\frac{K}{n}}(\widehat{x}(1) : x)] \; . \quad (91.239)$$

Under this change of variables, Eq.(91.237) becomes

$$P_{suc} = N_{\underline{m}} \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \theta(h > 0) e^{-n(-h+R)} \int_{-\infty}^{+\infty} dK \; e^{iK(-h+R)} e^{n \ln W_0} \theta_{P^{1-i\frac{K}{n}}(\widehat{x}(1),x)} \; . \quad (91.240)$$

Next we use Eqs.(91.15) and (91.16) to expand $\ln W_0$ to second order in K. This yields

$$\ln W_0 \approx -i\frac{K}{n} a - \frac{K^2}{2n^2} b \; , \quad (91.241)$$

where

$$a = H(\underline{\widehat{x}} : \underline{x}) \; , \quad (91.242)$$

and

[4] I don't know how to prove this assumption rigorously. The assumption is plausible, and it does lead to the correct result for the channel capacity. It may just be an approximation that becomes increasingly good as $n \to \infty$

$$
\begin{aligned}
b &= E_{\widehat{x}} E_x P(\widehat{x}:x) \ln^2 P(\widehat{x}:x) - H^2(\underline{\widehat{x}}:\underline{x}) && (91.243)\\
&= E_{\widehat{x},x} \ln^2 P(\widehat{x}:x) - [E_{\widehat{x},x} \ln P(\widehat{x}:x)]^2 && (91.244)\\
&\geq 0 \; . && (91.245)
\end{aligned}
$$

With the $\ln W_0$ expanded to second order in K, and $\theta_{P^{1-i\frac{K}{n}}(\widehat{x}(1),x)}$ to zeroth order in K, Eq.(91.240) becomes

$$
P_{suc} = \theta_{P_{\underline{\widehat{x}},\underline{x}}} N_{\underline{m}} \int_{-\infty}^{+\infty} \frac{dh}{2\pi} \, \theta(h>0) e^{n(h-R)} \int_{-\infty}^{+\infty} dK \; e^{iK(-a-h+R)-\frac{K^2}{2n}b} \; . \qquad (91.246)
$$

If we keep only the term linear in K in the argument of the exponential, we immediately get

$$
P_{suc} \approx \theta_{P_{\underline{\widehat{x}},\underline{x}}} N_{\underline{m}} e^{-na} \theta(R>a) \approx N_{\underline{m}} e^{-nR} \theta(R > H(\underline{\widehat{x}}:\underline{x})) \; . \qquad (91.247)
$$

Minimizing both sides of Eq.(91.247) with respect to the channel $P_{\widehat{x}|x}$ and using the definition of the rate distortion function $H_{\underline{x}}(D)$, we get that there is an encoding and a decoding for which

$$
P_{suc} = N_{\underline{m}} e^{-nR} \theta(R > H_{\underline{x}}(D)) \; . \qquad (91.248)
$$

QED

Claim 149

$$
R = R_{\underline{m}} \qquad (91.249)
$$

for consistency of our arguments.

proof: For consistency, must have $N_{\underline{m}} e^{-nR} = 1$ in Eq.(91.248).
QED

91.7 Appendix: Some Integrals Over Polytopes

This appendix is a collection of integration formulas for doing integrals over polytope shaped regions. These formulas are useful for doing p-type integrations.

The standard polytope is defined as the set $\Delta^n = \{(t_0, t_1, \ldots, t_n) : t_0 + t_1 + \ldots + t_n = 1, t_j \geq 0 \text{ for all } j\}$.

For $\{P_x\}_{\forall x} \in pd(val(\underline{x}))$, we define the following integration operator:

$$
\int \mathcal{D}P_{\underline{x}} = \prod_x \left\{ \int_0^1 dP_x \right\} \delta \left(\sum_x P_x - 1 \right) \; . \qquad (91.250)
$$

This is the same definition as Eq.(91.22), except for an arbitrary vector $\{P_x\}_{\forall x}$ instead of just for a p-type $\{P_{[x^n]}(x)\}_{\forall x}$.

It is well known and easy to show by induction that

$$\int \mathcal{D}P_{\underline{x}}\, 1 = \frac{1}{(N_{\underline{x}}-1)!}\ . \tag{91.251}$$

More generally, the so called Dirichlet integral, defined by

$$\begin{aligned} I_n &= \prod_{j=1}^{n}\left\{\int_0^1 dx_j\ \ x_j^{a_j-1}\right\}\int_0^1 dx_0\delta\left(\sum_{j=0}^{n} x_j - 1\right) && (91.252)\\ &= \prod_{j=1}^{n}\left\{\int_0^1 dx_j\ \ x_j^{a_j-1}\right\}\theta(\sum_{j=1}^{n} x_j \leq 1) && (91.253)\end{aligned}$$

can be shown[5] to be equal to

$$I_n = \frac{\prod_{j=1}^{n}\Gamma(a_j)}{\Gamma(\sum_{j=1}^{n} a_j)}\ , \tag{91.254}$$

where $\Gamma(\cdot)$ stands for the Gamma function. $\Gamma(n) = (n-1)!$ for any positive integer n.

In SIT, when doing p-type integrals for large n, one often encounters integrals of sharply peaked Gaussian functions integrated over polytope regions. Since the Gaussians are sharply peaked, as long as their peak is not near the boundary of the polytope region, the integrals can be easily evaluated approximately in a Gaussian approximation which becomes increasingly accurate as n increases.

Recall that

$$\int_{-\infty}^{+\infty} dx\ e^{-\lambda x^2} = \sqrt{\frac{\pi}{\lambda}} \tag{91.255}$$

for $\lambda > 0$.

Claim 150 *Suppose $\{Q_x\}_{\forall x} \in pd(val(\underline{x}))$, $\Delta P_x = P_x - Q_x$, and $\lambda_x >> 1$ for all $x \in val(\underline{x})$. Then*

$$\int \mathcal{D}P_{\underline{x}}\ \ \exp\left(-\sum_x \lambda_x(\Delta P_x)^2\right) \approx \sqrt{\frac{\pi^{N_{\underline{x}}-1}}{\prod_x\{\lambda_x\}\left(\frac{1}{\lambda_\parallel}\right)}}\ , \tag{91.256}$$

where $\lambda_\parallel = \left(\sum_x \frac{1}{\lambda_x}\right)^{-1}$. (If the λ_x are thought of as electrical resistances connected in parallel, then $\lambda_\parallel$ is the equivalent resistance.)

[5]See, for example, Ref.[25] for a proof.

proof: Let LHS and RHS denote the left hand side and right hand side of Eq.(91.256). One has

$$LHS \approx \prod_x \left\{ \int_{-\infty}^{+\infty} d\Delta P_x \right\} \delta(\sum_x \Delta P_x) \exp\left(-\sum_x \lambda_x (\Delta P_x)^2 \right) \tag{91.257}$$

$$= \int_{-\infty}^{+\infty} \frac{dk}{2\pi} \Gamma \tag{91.258}$$

where

$$\Gamma = \prod_x \left\{ \int_{-\infty}^{+\infty} d\Delta P_x \ \exp\left(-\lambda_x (\Delta P_x)^2 + ik\Delta P_x \right) \right\} \tag{91.259}$$

$$= \prod_x \left\{ e^{-\frac{k^2}{4\lambda_x}} \int_{-\infty}^{+\infty} d\Delta P_x \exp\left(-\lambda_x (\Delta P_x - \frac{ik}{2\lambda_x})^2 \right) \right\} \tag{91.260}$$

$$= e^{-\frac{k^2}{4\lambda_\parallel}} \prod_x \left\{ \sqrt{\frac{\pi}{\lambda_x}} \right\} . \tag{91.261}$$

Thus

$$LHS = \prod_x \left\{ \sqrt{\frac{\pi}{\lambda_x}} \right\} \int_{-\infty}^{+\infty} \frac{dk}{2\pi} \ e^{-\frac{k^2}{4\lambda_\parallel}} \tag{91.262}$$

$$= \prod_x \left\{ \sqrt{\frac{\pi}{\lambda_x}} \right\} \frac{1}{2\pi} \sqrt{\frac{\pi}{\frac{1}{4\lambda_\parallel}}} \tag{91.263}$$

$$= RHS . \tag{91.264}$$

QED

Claim 151 *Suppose matrix* $(A_{x,x'})_{\forall x,x'}$ *has eigenvalues* $\{\lambda_x\}_{\forall x}$*. Suppose* $\{Q_x\}_{\forall x} \in pd(val(\underline{x}))$*,* $\Delta P_x = P_x - Q_x$*, and* $\lambda_x >> 1$ *for all* $x \in val(\underline{x})$*. Then*

$$\int \mathcal{D}P_{\underline{x}} \ \exp\left(-\sum_{x,x'} \Delta P_x A_{x,x'} \Delta P_{x'} \right) \approx \sqrt{\frac{\pi^{N_{\underline{x}}-1}}{\det(A)\mathrm{tr}(A^{-1})}} , \tag{91.265}$$

proof: Just diagonalize the matrix $A_{x,x'}$ and use the previous claim, where now the λ_x are the eigenvalues of A.
QED

For $\{P_{y|x}\}_{\forall y} \in pd(val(\underline{y}))$ for all $x \in val(\underline{x})$, we define the following integration operator:

$$\int \mathcal{D}P_{\underline{y}|\underline{x}} = \prod_{x,y} \left\{ \int_0^1 dP_{y|x} \right\} \prod_x \left\{ \delta \left(\sum_y P_{y|x} - 1 \right) \right\} \ . \tag{91.266}$$

This is the same definition as Eq.(91.56), except for an arbitrary vector $\{P_{y|x}(y|x)\}_{\forall y}$ instead of just for a p-type $\{P_{[y^n|x^n]}(y|x)\}_{\forall y}$.

Note that Eq.(91.251) implies that

$$\int \mathcal{D}P_{\underline{y}|\underline{x}} \ 1 = \left[\frac{1}{(N_{\underline{y}} - 1)!} \right]^{N_{\underline{x}}} \ . \tag{91.267}$$

Claim 152 *Suppose matrix $A_{y|x\,,\,y'|x'}$ has eigenvalues $\{\lambda_{y|x}\}_{\forall x,y}$. Suppose $\{Q_{y|x}\}_{\forall y} \in pd(val(\underline{y}))$, $\Delta P_{y|x} = P_{y|x} - Q_{y|x}$, and $\lambda_{y|x} >> 1$ for all $x \in val(\underline{x})$ and $y \in val(\underline{y})$. Then (using Einstein's repeated index summation convention)*

$$\int \mathcal{D}P_{\underline{y}|\underline{x}} \ \exp\left(-\Delta P_{y|x} A_{y|x\,,\,y'|x'} \Delta P_{y'|x'} \right) \approx \sqrt{\frac{\pi^{N_{\underline{y}} N_{\underline{x}} - N_{\underline{x}}}}{\det(A) \det\left[\left(\sum_{y_1,y_2} A^{-1}_{y_1|x_1\,,\,y_2|x_2} \right)_{\forall x_1,x_2} \right]}} \ , \tag{91.268}$$

proof: Let LHS and RHS denote the left hand side and right hand side of Eq.(91.268). Let $(\omega_y)_{y \in val(\underline{y})}$ be a vector with all components equal to one. Then

$$\begin{aligned} LHS &\approx \prod_{x,y} \left\{ \int_{-\infty}^{+\infty} d\Delta P_{y|x} \right\} \prod_x \left\{ \delta(\omega_y \Delta P_{y|x}) \right\} e^{-\Delta P_{y|x} A_{y|x\,,\,y'|x'} \Delta P_{y'|x'}} && (91.269) \\ &= \prod_x \left\{ \int_{-\infty}^{+\infty} \frac{dk_x}{2\pi} \right\} \Gamma \ , && (91.270) \end{aligned}$$

where

$$\begin{aligned} \Gamma &= \prod_{x,y} \left\{ \int_{-\infty}^{+\infty} d\Delta P_{y|x} \right\} e^{-\Delta P_{y|x} A_{y|x\,,\,y'|x'} \Delta P_{y'|x'} + i\omega_y \Delta P_{y|x} k_x} && (91.271) \\ &= e^{-\frac{1}{4} k_{x_1} \omega_{y_1} A^{-1}_{y_1|x_1\,,\,y_2|x_2} \omega_{y_2} k_{x_2}} \prod_{x,y} \left\{ \int_{-\infty}^{+\infty} d\Delta P_{y|x} \right\} e^{-\widetilde{\Delta} P_{y|x} A_{y|x\,,\,y'|x'} \widetilde{\Delta} P_{y'|x'}} && (91.272) \end{aligned}$$

where

$$\widetilde{\Delta} P_{y|x} = \Delta P_{y|x} - \frac{i}{2} k_{x_1} \omega_{y_1} A^{-1}_{y_1|x_1\,,\,y|x} \ . \tag{91.273}$$

Thus

$$\Gamma = e^{-\frac{1}{4} k_{x_1} \omega_{y_1} A^{-1}_{y_1|x_1\,,\,y_2|x_2} \omega_{y_2} k_{x_2}} \sqrt{\frac{\pi^{N_{\underline{x}} N_{\underline{y}}}}{\det A}} \ . \tag{91.274}$$

Thus

$$
\begin{aligned}
LHS &= \sqrt{\frac{\pi^{N_{\underline{x}} N_{\underline{y}}}}{\det A}} \prod_x \left\{ \int_{-\infty}^{+\infty} \frac{dk_x}{2\pi} \right\} e^{-\frac{1}{4} k_{x_1} \omega_{y_1} A^{-1}_{y_1|x_1\,,\,y_2|x_2} \omega_{y_2} k_{x_2}} && (91.275) \\
&= \sqrt{\frac{\pi^{N_{\underline{x}} N_{\underline{y}}}}{\det A}} \frac{\pi^{\frac{N_{\underline{x}}}{2}}}{(2\pi)^{N_{\underline{x}}}} \frac{1}{\sqrt{\det\left[\left(\frac{\omega_{y_1} A^{-1}_{y_1|x_1\,,\,y_2|x_2} \omega_{y_2}}{4}\right)_{\forall x_1, x_2}\right]}} && (91.276) \\
&= RHS\ . && (91.277)
\end{aligned}
$$

QED

When using many of the integration formulas presented in this appendix, it is necessary to calculate the inverse and determinant of a large matrix. I found the following formulas can often be helpful in doing this.

Claim 153 *Suppose E is an $n \times n$ matrix. Suppose p and q are n component column vectors. Suppose*

$$A = E + pq^T\ . \tag{91.278}$$

Then

$$A^{-1} = E^{-1} - \frac{E^{-1} p q^T E^{-1}}{1 + q^T E^{-1} p}\ , \tag{91.279a}$$

$$\det(A) = \det(E)(1 + q^T E^{-1} p)\ . \tag{91.279b}$$

proof: To prove Eq.(91.279a), just show that the right hand sides of Eqs.(91.278) and (91.279a) multiply to one.

To prove Eq.(91.279b), one may proceed as follows. We will assume $A \in \mathbb{C}^{3\times 3}$ for concreteness. The proof we will give generalizes easily to A's of dimension different from 3. Let $\epsilon_{j_1 j_2, j_3}$ be the totally antisymmetric tensor with 3 indices. We will use Einstein summation convention. Let

$$Q_j = q_k (E^{-1})_{k,j}\ . \tag{91.280}$$

Then

$$
\begin{aligned}
\det(A) &= \det(E) \det(\delta_{i,j} + p_i Q_j) && (91.281) \\
&= \det(E) \epsilon_{j_1, j_2, j_3} (\delta_{1,j_1} + p_1 Q_{j_1})(\delta_{2,j_2} + p_2 Q_{j_2})(\delta_{3,j_3} + p_3 Q_{j_3}) && (91.282) \\
&= \det(E)(1 + p_j Q_j)\ . && (91.283)
\end{aligned}
$$

QED

Claim 154 *Suppose A is an $n \times n$ matrix, and $0 < \epsilon << 1$. Then*

$$\det(1 + \epsilon A) = 1 + \epsilon \mathrm{tr}(A) + O(\epsilon^2) \ . \tag{91.284}$$

proof: Just diagonalize A.
QED

Chapter 92

Shapley Explainability

This chapter is based on Refs.[46] and [47], which I highly recommended.

"AI" is an ill-defined term. So is the term "explainability". So the term "Explainable AI (XAI)" is doubly ill-defined. In 2018, the European Union codified the need for XAI — because of an individual's "right to explanation" — into a law called the General Data Protection Right (GDPR). This EU law was a strong motivation for Neural Net and boosted decision tree practitioners to come up with a way to enhance their machine learning algorithms so that these comply with that law. Shapley explainability (SX) is one of the most popular methods for doing XAI.

So what does SX do? It ranks, for each individual of a population, the features (for example, race) of a dataset, in the order of how influential those features were in arriving at the decision the classifier made for that individual.

In my opinion, the goal of XAI is accomplished better with bnets than with SX enhanced NNs. SX "explains", a posteriori, the outcome of a model, whereas bnets reveal the a priori process whereby that outcome was reached. Thus, SX can tell you that a model is racist, but it can't suggest how to fix it. On the other hand, if a bnet is acting racist, you don't have to throw it away. It can be fixed. Another weakness of SX is that it is quite expensive computationally. Bnets have explainability built into them. For bnets, explainability is not an additional, posterior and quite onerous, calculation. That is why I like to call bnets the gold standard of XAI.

Let

F be the feature set. For example, $F = \{age, gender, job\}$.

$\mathcal{P}(A) = \{S : S \subset A\}$ be the power set of the set A (i.e., the set of all subsets of A, including the empty set $\emptyset$).

$|\mathcal{P}(A)| = \sum_{k=0}^{|A|} \binom{|A|}{k} = \sum_{k=0}^{|A|} \binom{|A|}{k} 1^k 1^{|A|-k} = (1+1)^{|A|} = 2^{|A|}$

$\mathcal{P}_f(F) = \{S \in \mathcal{P}(F) : f \in S\}$ for $f \in F$, be all sets in $\mathcal{P}(F)$ containing feature f. Note that $\mathcal{P}_f(F) = \mathcal{P}(F) - \mathcal{P}(F - \{f\})$

$\mathcal{P}_{!f}(F) = \{S \in \mathcal{P}(F) : f \notin S\}$ for $f \in F$, be all sets in $\mathcal{P}(F)$ not containing feature f. Note that $\mathcal{P}_{!f}(F) = \mathcal{P}(F - \{f\})$

Fig.92.1 shows a graph of $\mathcal{P}(F)$ for $F = \{age, gender, job\}$. Henceforth, we will refer to the generalization of Fig.92.1 to an arbitrary finite set F, as the **power**

Figure 92.1: Graph of $\mathcal{P}(F)$ for $F = \{age, gender, job\}$. An arrow $H \leftarrow T$, where $H \in \mathcal{P}(F)$ is the head of the arrow and $T \in \mathcal{P}(F)$ is the tail of the arrow, means $H \supset T$ and $|H| = |T| + 1$.

set graph of F.[1]

For any finite set F, consider its power set graph. Let $S \in \mathcal{P}(F)$ be a node of that graph. Let

$N_{arr/nd}(S) = |S|$ = number of arrows entering node S.

$N_{nds}(|S|) = \binom{|F|}{|S|}$ = number of nodes in level $|S|$ (i.e., row $|S|$).

$N_{arr}(S) = N_{arr/nd}(S) N_{nds}(|S|) = |S| \binom{|F|}{|S|}$ = number of arrows going from row $|S| - 1$ to row $|S|$.

$$P(S) = \frac{1}{N_{arr}(S)} = \frac{1}{|S| \binom{|F|}{|S|}} = \frac{1}{|F| \binom{|F|-1}{|S|-1}} \ . \tag{92.1}$$

Claim 155

$$\sum_{S \in \mathcal{P}_f(F)} P(S) = 1 \tag{92.2}$$

proof:

[1]Note that a power set graph is a DAG. We won't define TPMs for its nodes in this chapter, so it's a DAG but not a bnet, in this chapter at least.

$$
\begin{aligned}
\sum_{S \in \mathcal{P}_f(F)} P(S) &= \sum_{S \in \mathcal{P}_f(F)} \frac{1}{|F| \binom{|F|-1}{|S|-1}} && (92.3) \\
&= \sum_{k=1}^{|F|} \sum_{S \in \mathcal{P}_f(F)} \mathbb{1}(|S| = k) \frac{1}{|F| \binom{|F|-1}{k-1}} && (92.4) \\
&= \sum_{k=1}^{|F|} \frac{1}{|F| \binom{|F|-1}{k-1}} \underbrace{\sum_{S \in \mathcal{P}_f(F)} \mathbb{1}(|S| = k)}_{\binom{|F|-1}{k-1}} && (92.5) \\
&= 1 && (92.6)
\end{aligned}
$$

QED

Consider any $S \in \mathcal{P}(F)$. Henceforth, we will represent a Machine Learning (ML) model ML_S as follows. ML_S can be a Linear Regression (LR_S) model, a Neural Net model (NN_S), or any other type of ML model. We will list a dataset; i.e., a set of tuples indexed by the individuals σ of a population Σ such that $|\Sigma| = nsam$. The independent variables of ML_S (i.e., $x_S^\sigma = [x_f^\sigma : f \in S]$) will be shown unboxed and the dependent variable (a.k.a. target feature) (i.e., y^σ) will be shown inside a box. Then we will show an arrow with the superscript "ML-fit", followed by the fit function obtained by performing ML_S.

$$
ML_S : \quad \{(\sigma, x_S^\sigma, \boxed{y^\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{y}(x_S) \tag{92.7}
$$

For any feature $f \in F$ and individual $\sigma \in \Sigma$, define the **Shapley Value** (SHAP) by

$$
\begin{aligned}
SHAP_f^\sigma &= \sum_{S \in \mathcal{P}_f(F)} P(S) \left[\widehat{y}(x_S^\sigma) - \widehat{y}(x_{S-\{f\}}^\sigma) \right] && (92.8) \\
&= E_S[\widehat{y}(x_S^\sigma) - \widehat{y}(x_{S-\{f\}}^\sigma)] && (92.9)
\end{aligned}
$$

Hence,

- $SHAP_f^\sigma$ is an average over the ensemble $\{ML_S : S \in \mathcal{P}_f(F)\}$ for each individual $\sigma \in \Sigma$.
- $SHAP_f^\sigma$ averages the change in output $\widehat{y}$ when we change the model from one without feature f to one with feature f.
- $SHAP_f^\sigma$ can be negative or positive. Zero $SHAP_f^\sigma$ for individual σ means feature f does not influence how the decision $\widehat{y}$ was arrived at for individual σ.

- An exact calculation of $SHAP_f^\sigma$, for all f and for a single σ, requires training a different ML model for each node of the power set graph of F, so it requires training $2^{|F|}$ models. Yikes! As $|F|$ grows, this quickly becomes unfeasible. For large $|F|$, one must resort to using sampling and approximations to get an approximation of the SHAP.

92.0.1 Numerical examples of SHAP

Next we present 2 numerical examples of SHAP. The figures and numerical values in this section were taken directly from Refs. [46] and [47].

1. **Predicting Income from $F = \{age, gender, job\}$.**

Figure 92.2: Same as Fig.92.1, but with added information for a specific individual σ. This figure contains enough information to evaluate $SHAP_{age}^\sigma$.

Consider the problem of predicting the income of a person based on the feature set $F = \{age, gender, job\}$. Suppose we are given a dataset for this problem. We can train models ML_S for each $S \in \mathcal{P}_{age}(F)$ where $\widehat{y}$ is the income. Then we can calculate the matrix $SHAP_{age}^\sigma$. Fig.92.2 gives all the information necessary to calculate $SHAP_{age}^\sigma$ for a single individual σ.

$$P(S) = \frac{1}{3\binom{2}{|S|-1}} = \begin{cases} \frac{1}{3\binom{2}{0}} = \frac{1}{3} & \text{if } |S| = 1 \\ \frac{1}{3\binom{2}{1}} = \frac{1}{6} & \text{if } |S| = 2 \\ \frac{1}{3\binom{2}{2}} = \frac{1}{3} & \text{if } |S| = 3 \end{cases} \tag{92.10}$$

$$\begin{aligned} SHAP^{\sigma}_{age} &= \underbrace{P(age)}_{1/3} \underbrace{[\widehat{y}(x^{\sigma}_{age}) - \widehat{y}(x^{\sigma}_{\emptyset})]}_{40K-50K} && (92.11) \\ &+ \underbrace{P(age, job)}_{1/6} \underbrace{[\widehat{y}(x^{\sigma}_{age,job}) - \widehat{y}(x^{\sigma}_{job})]}_{85K-100K} && (92.12) \\ &+ \underbrace{P(age, gender)}_{1/6} \underbrace{[\widehat{y}(x^{\sigma}_{age,gender}) - \widehat{y}(x^{\sigma}_{gender})]}_{39K-48K} && (92.13) \\ &+ \underbrace{P(age, gender, job)}_{1/3} \underbrace{[\widehat{y}(x^{\sigma}_{age,gender,job}) - \widehat{y}(x^{\sigma}_{gender,job})]}_{83K-95K} && (92.14) \\ &= -\$11.33K && (92.15) \end{aligned}$$

2. **Predicting passenger survival in the Titanic disaster.**

PassengerId	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
1	3	male	22.0	1	0	7.2500	S
2	1	female	38.0	1	0	71.2833	C
3	3	female	26.0	0	0	7.9250	S
4	1	female	35.0	1	0	53.1000	S
5	3	male	35.0	0	0	8.0500	S

Figure 92.3: First five rows (passengers) of an abridged version of the Titanic Dataset available at kaggle.com. This figure shows (σ, x^{σ}_{F}) for $\sigma = 1, 2, \ldots, 5$. It doesn't show the column $y^{\sigma} \in \{died, survived\}$.

Consider the problem of predicting whether an individual will survive or not based on a Titanic Dataset. We can train models ML_S for each $S \in \mathcal{P}(F)$ where $\widehat{y} \in \{died, survived\}$. Then we can calculate the matrix $SHAP^{\sigma}_{f}$ for all individuals σ and features f. Fig.92.0.1 shows the first 5 rows of an abridged[2] version of the Titanic Dataset available at kaggle.com. Fig.92.4 displays $SHAP^{\sigma}_{f}$ in tabular form and Fig.92.5 displays $SHAP^{\sigma}_{f}$ in graphical form (in what is called a beeswarm plot).

[2]The Titanic Dataset available at kaggle.com has 891 rows and 15 columns, including columns for passenger ID and for $y^{\sigma} = survived? \in \{0, 1\}$. This abridged version has 8 columns.

PassengerId	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
1	-0.36	-0.76	0.11	0.05	-0.03	-0.3	-0.08
2	1.22	2.18	0.1	0.06	0.1	0.78	0.42
3	-0.75	1.62	0.07	0.1	-0.02	-0.01	-0.13
4	1.15	2.07	0.11	0.05	0.08	0.89	-0.13
5	-0.41	-0.73	-0.06	0.08	-0.03	-0.15	-0.08

Figure 92.4: For the Titanic dataset, this is a table of $SHAP_f^\sigma$, where $\sigma \in \{1, 2, \ldots 5\}$ and $f \in F$. Cells with positive SHAP are colored green, and those with negative SHAP are colored red. The colors are not an indication of whether the passenger died or survived. Note that the table of a dataset and the matrix $SHAP_f^\sigma$ have the same shape $(|\Sigma|, |F|)$.

Figure 92.5: For the Titanic Dataset, this is a so called "beeswarm" plot of $SHAP_f^\sigma$, where $\sigma \in \{1, 2, \ldots, 891\}$ and $f \in F$. In a beeswarm plot, the thickness of each row is proportional to how many individuals of the population have that value of the x coordinate. This plot comes from Ref.[46], where it was generated using the Titanic Dataset from kaggle.com and the wonderful Python library "SHAP". The SHAP library can plot Shapley Values in many other styles besides this one.

Chapter 93

Simpson's Paradox

This chapter is based on Chapter 6 of "The Book of Why", Ref.[66]. See also Ref.[182] and references therein.

Simpson's paradox is a recurring nightmare for all statisticians overseeing a clinical trial for a medicine. It is possible that if they leave out a certain "confounding" variable from a study, the study's conclusion on whether a medicine is effective or not, might be, without measuring that confounding variable, the opposite of what it would have been had that variable been measured.

Simpson's Paradox is greatly clarified by Judea Pearl's theory of causality. At the end of this chapter, we explain how.

Here is a simple example of Simpson's Paradox.

An equal number of patients of male and female genders are given a heart medicine or a placebo in a double blind study. Some subsequently have a heart attack. Let

$\underline{a}$ = heart attack? No=0, Yes=1
$\underline{t}$ = took medicine? No=0, Yes=1
$\underline{g}$ = gender? Female=F, Male=M

Figure 93.1: bnet for a simple example of Simpson's paradox. Here node $\underline{g}$ is a chain junction and a mediator.

This situation can be modeled by either bnet Fig.93.1. or bnet Fig.93.2. The two bnets are probabilistically equivalent (i.e., they both represent the same probability distribution $P(a, t, g)$) because

Figure 93.2: bnet that is probabilistically but not physically equivalent to bnet Fig.93.1. Here node $\underline{g}$ is a fork junction and a confounder.

$$P(g|t)P(t) = P(g,t) = P(t|g)P(g) \ . \tag{93.1}$$

For the bnet Fig.93.1, one has

$$P(a,g,t) = P(a|g,t)P(g|t)P(t) \ . \tag{93.2}$$

Therefore,

$$P(a=1|t) = \sum_g P(a=1|t,g)P(g|t) = E_{\underline{g}|t}P(a=1|t,\underline{g}) \ , \tag{93.3}$$

where $E_{\underline{g}|t}$ is a conditional expected value (a kind of weighted average).

Suppose q_0, q_1 are non-negative real numbers. For the vector $\vec{q} = (q_0, q_1)$:
Define a negative outcome (or failure or q_t increasing with t) if $q_0 \leq q_1$.
Define a positive outcome (or success or q_t decreasing with t) if $q_0 \geq q_1$.
Let

$$\vec{q}^{\,g} = [P(\underline{a}=1|t,g)]_{t=0,1} \tag{93.4}$$

for $g = M, F$, and

$$\vec{q}^{\,*} = [P(\underline{a}=1|t)]_{t=0,1} \ . \tag{93.5}$$

It is possible (see Fig.93.3 for a graphical explanation of how) to find perverse cases in which $P(a=1|t,g=M)$ and $P(a=1|t,g=F)$ increase with t but $P(a=1|t)$ decreases with t. So it is possible to conclude that the medicine is a failure for each of the two g populations considered separately, yet the medicine is a success when both populations are "amalgamated". The lesson is that a "trend reversal" is possible upon amalgamation. Trends are not necessarily preserved when we do a weighted average of type $E_{\underline{g}|t}$. $E_{\underline{g}|t}$ is an expected value on the random variable $\underline{g}$ conditioned on the root random variable $\underline{t}$.

So far we have assumed $a \in \{0,1\}$. Suppose that instead we assume a is a continuous variable taking values in the interval $[0,1]$. This could reflect a continuum of possible attacks from none to a deadly one. Likewise, suppose the treatment variable t takes on values in the interval $[0,1]$. This might reflect a continuum of

Figure 93.3: $\vec{q}^{\,M}$, $\vec{q}^{\,F}$ vectors and bounding box for vector $\vec{q}^{\,*}$.

Figure 93.4: Illustrative example of Simpson's paradox, assuming t and a are continuous. The pink (for $g = F$) and blue (for $g = M$) elliptical regions are clouds of sample points. The lines are the result of doing linear regression. For $g = M, F$ separately, increasing treatment increases the probability of attack, but the opposite trend occurs if we amalgamate genders.

possible doses of a medicine. Fig. 93.4 gives an illustrative example of Simpson's paradox for this case of continuous a and t.

So far, we have proven that probabilistically, the drug can be a failure for the populations of both sexes considered separately, but a success for the aggregate population.

93.1 Pearl Causality

Pearl Causality would add the following two important insights to this problem:

1. bnets Fig.93.1 and Fig.93.2, although they are probabilistically equivalent, do not represent the same physical situation. In fact, only Fig.93.2 occurs in this case, because gender is determined at birth, long before treatment $\underline{t}$ and effect $\underline{y}$ are. (See Chapter G)

2. To decide whether the medicine is effective, we must apply a $do()$ operator to the $\underline{t}$ variable in Fig.93.2. The effect of that $do()$ operator is to erase the arrow going from $\underline{g}$ to $\underline{t}$. This in turn means that the average $E_{\underline{g}|t}$ in our equation for $P(a = 1|t)$ becomes a simpler average $E_{\underline{g}}$ which is independent of $\underline{t}$. But for such an average, the bounding box in Fig.93.3 degenerates to its diagonal line that connects the tips of the two vectors $\vec{q}^{\,M}$ and $\vec{q}^{\,F}$. The vector $\vec{q}^{\,*}$ must now fall on that diagonal line and must therefore also fall in the success region.

In conclusion, as Judea Pearl would say, if we ask the right question to Nature, i.e., what is $P[a = 1|do(\underline{t} = t)]$ for $t = 0, 1$, we get as an answer that the aggregate population preserves rather than reverses the unanimous trend of the two gendered populations.

93.2 Numerical Example

(a,t,g)	number of patients segregated by gender	number of patients of either gender
0,0,M	19	47
0,0,F	28	
0,1,M	37	49
0,1,F	12	
1,0,M	1	13
1,0,F	12	
1,1,M	3	11
1,1,F	8	

Table 93.1: Data for numerical example of Simpson's Paradox. This fictitious data was taken directly from Table 6.4, page 210 of "The Book of Why", Ref.[66].

$$P(a|t,g) = \begin{array}{c|cccc} & 0,M & 0,F & 1,M & 1,F \\ \hline 0 & 19/20 & 28/40 & 37/40 & 12/20 \\ 1 & 1/20 & 12/40 & 3/40 & 8/20 \end{array} \tag{93.6}$$

$$P(a|t) = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 47/60 & 49/60 \\ 1 & 13/60 & 11/60 \end{array} \tag{93.7}$$

$$\begin{aligned} \frac{P(a=1,t=1,g=M)}{\sum_a P(a,t=1,g=M)} &= P(a=1|t=1,g=M) &= \frac{3}{40} \\ \frac{P(a=1,t=0,g=M)}{\sum_a P(a,t=0,g=M)} &= P(a=1|t=0,g=M) &= \frac{1}{20} = \frac{2}{40} \end{aligned} \tag{93.8}$$

$$\begin{aligned} \frac{P(a=1,t=1,g=F)}{\sum_a P(a,t=1,g=F)} &= P(a=1|t=1,g=F) &= \frac{8}{20} = \frac{16}{40} \\ \frac{P(a=1,t=0,g=F)}{\sum_a P(a,t=0,g=F)} &= P(a=1|t=0,g=F) &= \frac{12}{40} \end{aligned} \tag{93.9}$$

$$\begin{aligned} \frac{\sum_g P(a=1,t=1,g)}{\sum_g \sum_a P(a,t=1,g)} &= P(a=1|t=1) &= \frac{11}{60} \\ \frac{\sum_g P(a=1,t=0,g)}{\sum_g \sum_a P(a,t=0,g)} &= P(a=1|t=0) &= \frac{13}{60} \end{aligned} \tag{93.10}$$

Note that the right hand side of Eq.(93.8) is higher for $t=1$ than for $t=0$. Same trend occurs in Eqs.93.9 but is reversed in Eqs.93.10.

Chapter 94

Stochastic Differential Equations

This chapter is based mostly on Ref.[72].

Stochastic Differential Equations (SDE) are deterministic first order differential equations with additive external white noise. When discretized, they can be modelled as dynamic bnets of type DEN (Deterministic bnet with External Noise) (see Chapter 52).

This chapter deals with **Classical Stochastic Calculus**. In that calculus, the SDE have real valued solutions. A theory of **Quantum Stochastic Calculus** has also been developed (see, for example, Ref.[55]) that is very similar to the classical theory. In the quantum theory, the SDEs have complex valued solutions. The quantum theory describes quantum mechanical systems (such as lasers) whereas the classical theory describes classical macroscopic systems such as a pollen particle undergoing Brownian motion while submerged in a liquid.

94.1 Notation

Random variables, mean, covariance

$$\langle \underline{a} \rangle = E[\underline{a}] \tag{94.1}$$

$$\Delta \underline{a} = \underline{a} - \langle \underline{a} \rangle \tag{94.2}$$

$$Cov(\underline{a}, \underline{b}) = \langle \underline{a}, \underline{b} \rangle = \langle \Delta \underline{a} \Delta \underline{b} \rangle \tag{94.3}$$

$$\Delta_{t_0}^{t_1} a = a(t_1) - a(t_0) \tag{94.4}$$

Intervals of real numbers and of integers

$[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}$

Suppose $i, j \in \{0, 1, 2, \ldots\}$

$[i : j] = \{i, i+1, \ldots, j-1\}$ (like Python)

$[i-j] = \{i, i+1, \ldots, j\}$ (To distinguish from Python, we use dash instead of colon to indicate that last int is included.)

$[n] = [0:n] = \{0, 1, 2, \ldots, n-1\}$

Consider times $t_0 = 0 < t_1 < t_2 < \ldots < t_{N-1}$

$t_{[i-j]} = [t_i, t_{i+1}, \ldots, t_j]$

$$\lim_{N\to\infty} t_{[i-j]} = [t_i, t_j] \tag{94.5}$$

Stochastic Process

An **outcome space** Ω is a set of **events** ω.

A **stochastic process** $\underline{x}(t,\omega) \in \mathbb{R}^n$ with $t \geq 0$ and $\omega \in \Omega$ is a map $\underline{x} : (\mathbb{R}^+, \Omega) \to \mathbb{R}^n$. Normally, we don't write the ω dependence: we use $\underline{x}(t)$ instead of $\underline{x}(t,\omega)$.

For compactness, we will sometimes denote $x(t)$ by x_t and an event (x,t) by $\binom{x}{t}$.

Will often use $x_i = x(t_i)$.

We will use lower case Latin indices like $i, j, k \in [N]$ for time indices and lower case Greek letters like $\alpha, \beta, \mu, \nu \in [n]$ for $x \in \mathbb{R}^n$ components. Hence $x_{\mu,i} = x_\mu(t_i)$

We will use the **Einstein implicit summation convention** for lower case Greek indices. Hence

$$A_\mu B_\mu = \sum_{\mu\in[n]} A_\mu B_\mu \tag{94.6}$$

$dx^n = dx_0 dx_1 \ldots dx_{n-1}$

Path

A **path** is defined as one of the following sets, depending on whether we are considering continuous or discretized time:

$\underline{x}([t,s]) = \{\underline{x}(\tau) : \tau \in [t,s]\}$

$\underline{x}(t_{[j-k]}) = \{x(\tau) : \tau \in \{t_j, t_{j+1}, \ldots, t_k\}\}$

$\underline{x}_{[j-k]} = \{\underline{x}_j, \underline{x}_{j+1}, \ldots, \underline{x}_k\}$

Measure theorists speak of a set of sigma algebras parametrized by t with the t algebra containing all algebras with smaller t. They call this a **filtration**. A path $\underline{x}([t,s])$ is equivalent to a filtration, so we won't speak of filtrations here.

For any matrix A, we will use $A^\dagger$ to denote the Hermitian conjugate of A, A^T to denote its transpose, and A^* to denote its complex conjugate. $A^\dagger = A^{*T}$.

94.2 White Noise and Brownian Motion

White noise $\underline{W}(t) \in \mathbb{R}^n$ for $t \geq 0$ is a random process with the following properties:

- ==========

$$\underline{W}(t) = \mathcal{N}(W(t); \mu = 0, Cov = Q) \tag{94.7}$$

Figure 94.1: One dimensional white noise $\underline{W}(t)$

Figure 94.2: One dimensional Brownian motion $\underline{B}(t)$

- ==========

  $\underline{W}(t)$ and $\underline{W}(s)$ are independent for $t \neq s$

- ==========

$$E[\underline{W}(t)] = 0 \tag{94.8}$$

- ==========

$$C_{\underline{W}}(t,s) = \left\langle \underline{W}(t), \underline{W}^T(s) \right\rangle = Q\delta(t-s) \tag{94.9}$$

Brownian motion (a.k.a. **Wiener process**) $\underline{B}(t) \in \mathbb{R}^n$ for $t \geq 0$ is a random process with the following properties:

- ==========

$$\underline{B}(0) = 0^1 \tag{94.10}$$

[1] If you wish to consider $\underline{B}(0) = \beta_0 \neq 0$, replace $\underline{B}$ by $\underline{B} - \beta_0$

- ==========

$$\frac{\Delta_{t_k}^{t_{k+1}} \underline{B}}{\Delta_{t_k}^{t_{k+1}} t} \sim \mathcal{N}(\mu = 0, Cov = Q) \tag{94.11}$$

$$\underline{W}(t_k) \sim \mathcal{N}(\mu = 0, Cov = Q) \tag{94.12}$$

$$\frac{d\underline{B}}{dt} = \underline{W} \tag{94.13}$$

- ==========

  If $\underline{B}(t) \in \mathbb{R}^n$ and $[r, s] \cap [r', s'] = \emptyset$, then

$$E[(\Delta_r^s \underline{B})(\Delta_{r'}^{s'} \underline{B})] = 0 \tag{94.14}$$

  and

$$E[|\Delta_s^t \underline{B}|^2] = n|t - s| \tag{94.15}$$

  For example,

$$\begin{aligned} E[\Delta_1^4 \underline{B} \Delta_3^6 \underline{B}] &= E[(\Delta_1^3 \underline{B} + \Delta_3^4 \underline{B})(\Delta_3^4 \underline{B} + \Delta_4^6 \underline{B})] && (94.16) \\ &= E[(\Delta_3^4 \underline{B})^2] && (94.17) \\ &= n|4 - 3| && (94.18) \end{aligned}$$

  Eqs.(94.14) and (94.15) can be combined as follows

$$E[(\Delta_r^s \underline{B})(\Delta_{r'}^{s'} \underline{B})] = n \text{ len}([r, s] \cap [r', s']) \tag{94.19}$$

94.3 SDE bnet

In this section, we propose a bnet for a time discretized SDE. This bnet will be constructed by combining some nodes of various special types that occur frequently in bnet-tology. One such special type of node that we have discussed already, in Chapter 53), is a **marginalizer node** . Here are a few others. The TPM or structural equation associated with the node are printed in blue.

- **diff and diff0 nodes**

  The diff node is defined by

$$\begin{tikzcd}
\underline{a} \arrow[dr] & & \underline{b} \arrow[dl] \\
& \underline{x} &
\end{tikzcd}$$

$$P(x|a,b) = \mathbb{1}(x = a - b) \tag{94.20}$$

$$x = a - b \tag{94.21}$$

The diff0 node is the diff node with $x = 0$.

- **accumulator nodes**

$$\begin{tikzcd}
\underline{x}_3 \arrow[d] & \underline{x}_2 \arrow[d] & \underline{x}_1 \arrow[d] & \underline{x}_0 \arrow[d] \\
\underline{s}_3 & \underline{s}_2 \arrow[l] & \underline{s}_1 \arrow[l] & \underline{s}_0 \arrow[l]
\end{tikzcd}$$

$$\begin{aligned}
\underline{s}_3 &= \underline{x}_3 + \underline{s}_2 \\
\underline{s}_2 &= \underline{x}_2 + \underline{s}_1 \\
\underline{s}_1 &= \underline{x}_1 + \underline{s}_0 \\
\underline{s}_0 &= x_0
\end{aligned} \tag{94.22}$$

- **increment nodes**

$$\begin{tikzcd}
\underline{B}_3 \arrow[dr] & & \underline{B}_2 \arrow[dl] \arrow[dr] & & \underline{B}_1 \arrow[dl] \arrow[dr] & & \underline{B}_0 \arrow[dl] \\
& \Delta^3_2\underline{B} & & \Delta^2_1\underline{B} & & \Delta^1_0\underline{B} &
\end{tikzcd}$$

$$\begin{aligned}
\Delta^3_2\underline{B} &= \underline{B}_3 - \underline{B}_2 \\
\Delta^2_1\underline{B} &= \underline{B}_2 - \underline{B}_1 \\
\Delta^1_0\underline{B} &= \underline{B}_1 - \underline{B}_0
\end{aligned} \tag{94.23}$$

- **un-increment nodes**

$$\begin{tikzcd}
& \Delta^3_2\underline{x}_2 \arrow[dl] & & \Delta^2_1\underline{x} \arrow[dl] \arrow[dll] & & \Delta\underline{x}^1_0 \arrow[dl] \arrow[dll] \arrow[dll, bend right] & \\
\underline{x}_3 & & \underline{x}_2 & & \underline{x}_1 & & \underline{x}_0 \arrow[ll] \arrow[ll, bend left] \arrow[ll, bend left]
\end{tikzcd}$$

Figure 94.3: Bnet for general SDE with $N = 4$ number of times. Note that this bnet contains within it first an increment bnet (in red) for the $\underline{B}_i$, and then a un-increment bnet (in green) for the $\underline{x}_i$.

$$\begin{aligned} \underline{x}_3 &= \Delta^3_2\underline{x} + \Delta^2_1\underline{x} + \Delta^1_0\underline{x} + x_0 \\ \underline{x}_2 &= \Delta^2_1\underline{x} + \Delta^1_0\underline{x} + x_0 \\ \underline{x}_1 &= \Delta^1_0\underline{x} + x_0 \\ \underline{x}_0 &= x_0 \end{aligned} \tag{94.24}$$

SDE bnet

Fig.94.3 gives a bnet for a general one-dimensional ($n = 1$) SDE defined by

$$d\underline{x} = f(\underline{x}, t)dt + L(\underline{x}, t)d\underline{B}(t) \tag{94.25}$$

with $\underline{x}, \underline{B} \in \mathbb{R}$. Some of the structural equations, printed in blue, for the bnet of Fig.94.3, are as follows.

$$\Delta^3_2\underline{B} = \underline{B}_3 - \underline{B}_2 \tag{94.26}$$

$$\Delta^3_2\underline{x} = f(x_2, t)\Delta^3_2 t + L(x_2, t)\Delta^3_2\underline{B} \tag{94.27}$$

$$\underline{x}_3 = \Delta^3_2\underline{x} + \Delta^2_1\underline{x} + \Delta^1_0\underline{x} + x_0 \tag{94.28}$$

94.4 Simple Properties of SDE

In this section, we discuss several simple properties of SDEs.

94.4.1 STD with Constant Coefficients (CC)

The most general system discussed in this chapter obeys the following SDE

$$dx_\mu = f_\mu(x,t)dt + L_{\mu,\nu}(x,t)d\underline{B}_\nu(t) \tag{94.29}$$

where $\mu, \nu \in [n]$. A system for which f and L are both constant (i.e., independent of the event (x,t)) is said to have CC **Constant Coefficients**[2]

94.4.2 Transition Probability Matrices

Suppose $t, s \geq 0$ and $x, y \in \mathbb{R}^n$. We define the **event transition probability matrix (TPM)** as

$$P\left(\begin{smallmatrix} y|x \\ t|s \end{smallmatrix}\right) \tag{94.30}$$

If you are familiar with the Dirac bra-ket notation used in Quantum Mechanics, note that a TPM can be expressed in such notation as

$$\langle y|P_{t,s}|x\rangle \tag{94.31}$$

where $\{|x\rangle : x \in \mathbb{R}^n\}$ is a complete orthonormal basis:

$$\int dx\ |x\rangle\langle x| = 1, \quad \langle y|x\rangle = \delta(y-x) \tag{94.32}$$

94.4.3 Markov chain

Let $\underline{x}(t_k) = x_k$ for $k \in [N]$. The bnet

$$x_0 \to \underline{x}_1 \to \underline{x}_2 \to \cdots \to \underline{x}_{N-1}$$

is called a **Markov chain**. It satisfies

$$P(x_{i+1}|x_i, x_{i-1}, \ldots, x_0) = P(x_{i+1}|x_i) \tag{94.33}$$

For continuous instead of discrete time, the Markov chain definition is generalized as follows. For $s < t$,

$$P(x(t)|x([0,s])) = P(x(t)|x(s)) \tag{94.34}$$

[2]Ref.[72], on which most of this chapter is based, calls systems with CC, LTI (linear, time-invariant) systems.

94.4.4 Chapman-Kolgomorov Equation

Claim 156 *(Chapman-Kolgomorov equation)*
Let $x(t_k) = x_k$

$$P(x_3|x_1) = \int dx_2^n \; P(x_3|x_2)P(x_2|x_1) \tag{94.35}$$

proof:

$$\begin{aligned} P(x_3, x_2|x_1) &= P(x_3|x_2, x_1)P(x_2|x_1) && (94.36) \\ &= P(x_3|x_2)P(x_2|x_1) && (94.37) \end{aligned}$$

Now integrate both sides over x_2.
QED

94.4.5 Martingale

Let $0 \leq t, t_i \leq T$ and $x(t_i) = x_i$ for $i \in [N]$. A **martingale** for discrete (resp., continuous) time is a process $\underline{y}_i$ (resp., $\underline{y}(t)$) which satisfies

$$E[\ |\underline{y}(t_i)|\ ] < \infty \quad \forall i \quad (\text{resp., } E[\ |\underline{y}(t)|\ ] < \infty \quad \forall t \geq 0) \tag{94.38}$$

$$E\left[\underline{y}(t)|x(t_{[i-j]})\right] = y(t_j) \quad (\text{resp., } E\left[\underline{y}(t)|x([0,s])\right] = y(s)) \tag{94.39}$$

Brownian motion is a martingale.

$$E[\underline{B}(t)|B(t_{[i-j]}] = B(t_j) \tag{94.40}$$

Itô integrals $\int L_t^x d\underline{B}_t$ (discussed later) are martingales too, but Stratonovich integrals $\int L_t^x \circ d\underline{B}_t$ (discussed later) aren't.

94.5 Itô Integral

Consider the 1 dimensional case $\underline{x}, \underline{W} \in \mathbb{R}$.

$$\frac{d\underline{x}}{dt} = f(\underline{x}, t) + L(\underline{x}, t)\underline{W}(t) \tag{94.41}$$

$$\underline{x}(t) - \underline{x}(0) = \int_0^t dt \; f(x,t) + J \tag{94.42}$$

where

$$J = \int_0^t dt\ L(x,t)\underline{W}(t) \tag{94.43}$$

For $t_0 = 0 < t_1 < \ldots < t_{N-1}$, $t_k^* \in [t_k, t_{k+1})$, define

$$J_N = \sum_k L(\underline{x}, t_k^*)\Delta_{t_k}^{t_{k+1}}\underline{B} \tag{94.44}$$

and

$$J = \lim_{N\to\infty} J_N \tag{94.45}$$

Consider the integrand $L(x,t)\underline{W}(t)$. In non-rigorous calculus, we normally consider integrands that are smooth, so that as $N \to \infty$ and the separation between successive t_i goes to zero, the value of J is independent of where t_k^* is located inside the interval $[t_k, t_{k+1})$. In the SDE case, the integrand $L\underline{W}$ is continuous but very jagged, so the value of J does depend on the choice of t_k^*.

Consider the special case that $L = \underline{x} = \underline{B}$, so

$$J_N = \sum_k \underline{B}(t_k^*)\Delta_{t_k}^{t_{k+1}}\underline{B} \tag{94.46}$$

Let us see how in this simple case, the value of J depends on the choice of t_k^*.

1. $t_k^* = t_{k+1}$

$$\begin{aligned} E[J_N] &= \sum_k E[\underline{B}(t_{k+1})\Delta_{t_k}^{t_{k+1}}\underline{B}] && (94.47)\\ &= \sum_k E[(\Delta_0^{t_{k+1}}\underline{B})\Delta_{t_k}^{t_{k+1}}\underline{B}] && (94.48)\\ &= \sum_k E[(\Delta_{t_k}^{t_{k+1}}\underline{B})^2] && (94.49)\\ &= t && (94.50) \end{aligned}$$

2. $t_k^* = \frac{t_k + t_{k+1}}{2}$ Stratonovich integral

3. $t_k^* = t_k$, Itô (Ito) integral

$$\begin{aligned} E[J_N] &= \sum_k E[\underline{B}(t_k)\Delta_{t_k}^{t_{k+1}}\underline{B}] && (94.51)\\ &= \sum_k E[(\Delta_0^{t_k}\underline{B})\Delta_{t_k}^{t_{k+1}}\underline{B}] && (94.52)\\ &= 0 && (94.53) \end{aligned}$$

$$
\begin{aligned}
J_N &= \sum_k \underline{B}(t_k)\Delta_{t_k}^{t_{k+1}}\underline{B} && (94.54)\\
&= \sum_k B_k(B_{k+1}-B_k) && (94.55)\\
& && (94.56)\\
&= \frac{1}{2}\sum_k \left[-[B_{k+1}-B_k]^2 + (B_{k+1}^2 - B_k^2)\right] && (94.57)\\
&= \frac{1}{2}\sum_k \left[-[\Delta_{t_k}^{t_{k+1}}\underline{B}]^2 + \Delta_{t_k}^{t_{k+1}}(\underline{B}^2)\right] && (94.58)
\end{aligned}
$$

$$
\begin{aligned}
J &= \lim_{N\to\infty} J_N && (94.59)\\
&= \frac{1}{2}(-t + \underline{B}^2(t)) && (94.60)
\end{aligned}
$$

$E[J] = \frac{1}{2}(-t+t) = 0$

$$d[\underline{B}^2(t)] = 2\underline{B}(t)d\underline{B}(t) + t \tag{94.61}$$

$$[d\underline{B}(t)]^2 = dt \tag{94.62}$$

For $\underline{B} \in \mathbb{R}^n$,

$$\boxed{d[\underline{B}_\alpha(t)\underline{B}_\beta(t)] = \delta(\alpha,\beta)\left[2\underline{B}_\alpha(t)d\underline{B}_\alpha(t) + nt\right]} \tag{94.63}$$

$$\boxed{d[\underline{B}_\alpha(t)]d[\underline{B}_\beta(t)] = Q_{\alpha,\beta}dt} \tag{94.64}$$

94.6 Fokker-Planck Equation

Consider the n dimensional case $\underline{x}, \underline{W} \in \mathbb{R}^n$. Let $\mu, \nu, \alpha, \beta \in [n]$. Suppose

$$d\underline{x}_\mu = f_\mu(\underline{x}, t)dt + L_{\mu,\nu}(\underline{x}, t)d\underline{B}_\nu(t) \tag{94.65}$$

where

$$R_{\mu,\nu} = \frac{1}{2}L_{\mu,\alpha}Q_{\alpha,\beta}L^T_{\beta,\nu} \tag{94.66}$$

Claim 157

$$d\phi = \left[\frac{\partial\phi}{\partial t} + f_\mu\frac{\partial\phi}{\partial x_\mu} + R_{\mu,\nu}\frac{\partial^2\phi}{\partial x_\mu\partial x_\nu}\right]dt + \frac{\partial\phi}{\partial x_\mu}L_{\mu,\nu}d\underline{B}_\nu \tag{94.67}$$

proof:

The Taylor expansion of $\phi(x,t)$, up to second order derivatives, is

$$d\phi = \frac{\partial \phi}{\partial t}dt + \sum_{\mu} \frac{\partial \phi}{\partial x_\mu} d\underline{x}_\mu + \frac{1}{2}\sum_{\mu}\sum_{\nu} \frac{\partial^2 \phi}{\partial x_\mu \partial x_\nu} d\underline{x}_\mu d\underline{x}_\nu \tag{94.68}$$

$$\frac{\partial \phi}{\partial x_\mu} d\underline{x}_\mu = \frac{\partial \phi}{\partial x_\mu}\left[f_\mu dt + L_{\mu,\nu} d\underline{B}_\nu\right] \tag{94.69}$$

$$\begin{aligned}
\frac{\partial^2 \phi}{\partial x_\mu \partial x_\nu} d\underline{x}_\mu d\underline{x}_\nu &= \frac{\partial^2 \phi}{\partial x_\mu \partial x_\nu}\left[f_\mu dt + L_{\mu,\alpha} d\underline{B}_\alpha\right]\left[f_\nu dt + L_{\nu,\beta} d\underline{B}_\beta\right] && (94.70)\\
&= \frac{\partial^2 \phi}{\partial x_\mu \partial x_\nu} L_{\mu,\alpha} L_{\nu,\beta} d\underline{B}_\alpha d\underline{B}_\beta && (94.71)\\
&= \frac{\partial^2 \phi}{\partial x_\mu \partial x_\nu} L_{\mu,\alpha} L_{\nu,\alpha} Q_{\alpha,\alpha} dt && (94.72)
\end{aligned}$$

$$\begin{aligned}
d\phi &= \frac{\partial \phi}{\partial t}dt + \frac{\partial \phi}{\partial x_\mu}[f_\mu dt + L_{\mu,\nu} d\underline{B}_\nu] + R_{\mu,\nu}\frac{\partial^2 \phi}{\partial x_\mu \partial x_\nu} dt && (94.73)\\
&= \left[\frac{\partial \phi}{\partial t} + f_\mu \frac{\partial \phi}{\partial x_\mu} + R_{\mu,\nu}\frac{\partial^2 \phi}{\partial x_\mu \partial x_\nu}\right] dt + \frac{\partial \phi}{\partial x_\mu} L_{\mu,\nu} d\underline{B}_\nu && (94.74)
\end{aligned}$$

QED

For example, if $n = 1$, $\underline{x} = \underline{B}$, $\phi = \underline{B}^K$ and $L = Q = 1$, $f = 0$, we get

$$d(\underline{B}^m) = \left[m\underline{B}^{m-1} + m(m-1)\underline{B}^{m-2}\right] dt + m\underline{B}^{m-1} d\underline{B} \tag{94.75}$$

The next claim defines the **Fokker-Planck equation** (FP equation) (a.k.a. **Fokker-Planck-Kolgomorov equation**) for the probability $P(x,t)$ of single event (x,t)

Claim 158 *(Forward FP equation)*

If

$$dx = f(x,t)dt + L(x,t)d\underline{B} \tag{94.76}$$

Then

$$\frac{\partial P(x,t)}{\partial t} = \mathcal{F}_{\underline{x}} P(x,t) \tag{94.77}$$

with

$$\mathcal{F}_{\underline{x}} \bullet = -\frac{\partial}{\partial x_\mu}(\bullet f_\mu) + \frac{\partial^2}{\partial x_\mu \partial x_\nu}(\bullet R_{\mu,\nu}) \tag{94.78}$$

proof:

$$\int dx^n \; P_t^x \left[\frac{\partial \phi}{\partial x_\mu} L_{\mu,\nu} d\underline{B}_\nu \right] = 0 \tag{94.79}$$

Integration by parts

$$udv = d(uv) - (du)v \tag{94.80}$$

$$\int_{-\infty}^{+\infty} udv = \underbrace{uv|_{-\infty}^{+\infty}}_{0} \quad - \int_{-\infty}^{+\infty} (du)v \tag{94.81}$$

$$\int dtdx^n \; P_t^x \frac{d\phi}{dt} = \int dtdx^n \; P_t^x \left[\frac{\partial \phi}{\partial t} + f_\mu \frac{\partial \phi}{\partial x_\mu} + R_{\mu,\nu} \frac{\partial^2 \phi}{\partial x_\mu \partial x_\nu} \right] \tag{94.82}$$

$$-\int dtdx^n \; \phi \frac{dP}{dt} = \int dtdx^n \; \phi \left[-\frac{\partial P}{\partial t} - \frac{\partial (P f_\mu)}{\partial x_\mu} + \frac{\partial^2}{\partial x_\mu \partial x_\nu} (P R_{\mu,\nu}) \right] \tag{94.83}$$

QED

Claim 159

$$\frac{\partial P}{\partial t} = \mathcal{F}_{\underline{x}} P \tag{94.84}$$

is solved formally by

$$P(x,t) = e^{(t-t_0)\mathcal{F}_{\underline{x}}} P(x,t_0) \tag{94.85}$$

proof: Just use the Taylor expansion of an exponential function and analogize with the case when $\mathcal{F}_{\underline{x}}$ is a constant.
QED

Note that

$$\frac{\partial P}{\partial t} = -\frac{\partial J_\mu}{\partial x_\mu} \tag{94.86}$$

where

$$J_\mu = P f_\mu - \frac{\partial (P R_{\mu,\nu})}{\partial x_\nu} \tag{94.87}$$

Eq.(94.86) is the **equation for conservation of probability**[3]. J_μ is called the **probability flux**, f_μ is called the **drift**, and $R_{\mu,\nu}$ is called the **diffusion matrix** (or **diffusion coefficient** if it's a scalar)

[3]It implies conservation of probability because

$$0 = \frac{\partial}{\partial t} \int_V dV \; P = \int_V dV \; \nabla \cdot \vec{J} = \int \vec{J} \cdot d\vec{S}$$

The forward FP equation when $n = 1$, $f = 0$, $L = 1$, $R = D > 0$, is called the **Diffusion Equation**.

$$dx = d\underline{B} \tag{94.88}$$

$$\frac{\partial P}{\partial t} = D\frac{\partial^2 P}{\partial^2 x} \tag{94.89}$$

As another example of the forward FP, consider the **Overdamped Langevin Equation**[4]

$$dx_\mu = -\frac{1}{2}\frac{\partial U}{\partial x_\mu}dt + d\underline{B} \tag{94.90}$$

Claim 160 *For the overdamped Langevin equation, if $Q = \frac{1}{\lambda} > 0$, then the steady state solution is*

$$P(x) = \frac{e^{-\lambda U(x)}}{Z} \tag{94.91}$$

where

$$Z = \int dx^n\ e^{-\lambda U(x)} \tag{94.92}$$

proof:

The forward FP equation with $L = 1$, $\frac{\partial P}{\partial t} = 0$ is

$$0 = -\frac{\partial(Pf_\mu)}{\partial x_\mu} + \frac{1}{2\lambda}\frac{\partial^2 P}{\partial x_\mu^2} \tag{94.93}$$

If we substitute $f_\mu = -\frac{1}{2}\frac{\partial U}{\partial x_\mu}$ into this, we get

$$0 = \frac{1}{2}\frac{\partial}{\partial x_\mu}\left(P\frac{\partial U}{\partial x_\mu} + \frac{1}{\lambda}\frac{\partial P}{\partial x_\mu}\right) \tag{94.94}$$

If we now substitute the proposed value of P, we get $0 = 0$

$$0 = \frac{\partial}{\partial x_\mu}\left(\lambda e^{-\lambda U}\frac{\partial U}{\partial x_\mu} + \frac{\partial e^{-\lambda U}}{\partial x_\mu}\right) \tag{94.95}$$

QED

Recall that

[4]This equation, also known as Brownian dynamics (see Ref.[119]), arises from Newton's equation $m\ddot{x} = -\lambda\dot{x} - U'(x)$ when the acceleration $\ddot{x}$ is negligible, so the drag force and potential force cancel each other.

$$\mathcal{F}_{\underline{x}}\bullet = -\frac{\partial}{\partial x_\mu}(\bullet f_\mu) + \frac{\partial^2}{\partial x_\nu \partial x_\mu}(\bullet R_{\mu,\nu}) \tag{94.96}$$

Define $\mathcal{B}_{\underline{x}}$ to be the same as $\mathcal{F}_{\underline{x}}$ but with every derivative $A\frac{\partial B}{\partial x_\mu}$ replaced by $-B\frac{\partial A}{\partial x_\mu}$. Hence[5]

$$\mathcal{B}_{\underline{x}}\bullet = f_\mu \frac{\partial \bullet}{\partial x_\mu} + R_{\mu,\nu}\frac{\partial^2 \bullet}{\partial x_\mu \partial x_\nu} \tag{94.97}$$

Claim 161 *(Forward FP equation for transition matrix $P\left(\begin{smallmatrix} x \\ t \end{smallmatrix}\middle|\begin{smallmatrix} w \\ s \end{smallmatrix}\right)$)*
$P\left(\begin{smallmatrix} x \\ t \end{smallmatrix}\middle|\begin{smallmatrix} w \\ s \end{smallmatrix}\right)$ satisfies

$$\frac{\partial P\left(\begin{smallmatrix} x \\ t \end{smallmatrix}\middle|\begin{smallmatrix} w \\ s \end{smallmatrix}\right)}{\partial t} = \mathcal{F}_{\underline{x}} P\left(\begin{smallmatrix} x \\ t \end{smallmatrix}\middle|\begin{smallmatrix} w \\ s \end{smallmatrix}\right) \tag{94.98}$$

with

$$P\left(\begin{smallmatrix} x \\ s \end{smallmatrix}\middle|\begin{smallmatrix} w \\ s \end{smallmatrix}\right) = \delta(x-w) \tag{94.99}$$

In case you know Dirac bra-ket notation, note that the differential equation Eq.(94.98) can be expressed in Dirac notation as

$$\frac{\partial \langle x|P_{t,s}|w\rangle}{\partial t} = \int \langle x|\mathcal{F}_t|x'\rangle\, dx'\, \langle x'|P_{t,s}|w\rangle \tag{94.100}$$

with

$$\langle x|P_{s,s}|w\rangle = \delta(x-w) \quad \textit{(Hence, } P_{s,s}=1\textit{)} \tag{94.101}$$

Claim 162 *(Backward FP equation for transition matrix $P\left(\begin{smallmatrix} y \\ u \end{smallmatrix}\middle|\begin{smallmatrix} x \\ t \end{smallmatrix}\right)$)*
$P\left(\begin{smallmatrix} y \\ u \end{smallmatrix}\middle|\begin{smallmatrix} x \\ t \end{smallmatrix}\right)$ satisfies

$$-\frac{\partial P\left(\begin{smallmatrix} y \\ u \end{smallmatrix}\middle|\begin{smallmatrix} x \\ t \end{smallmatrix}\right)}{\partial t} = \mathcal{B}_{\underline{x}} P\left(\begin{smallmatrix} y \\ u \end{smallmatrix}\middle|\begin{smallmatrix} x \\ t \end{smallmatrix}\right) \tag{94.102}$$

with

$$P\left(\begin{smallmatrix} y \\ t \end{smallmatrix}\middle|\begin{smallmatrix} x \\ t \end{smallmatrix}\right) = \delta(y-x) \tag{94.103}$$

In case you know Dirac bra-ket notation, note that the differential equation Eq.(94.102) can be expressed in Dirac notation as

$$-\frac{\partial \langle y|P_{u,t}|x\rangle}{\partial t} = \int \langle y|\mathcal{B}_t|y'\rangle\, dy'\, \langle y'|P_{u,t}|x\rangle \tag{94.104}$$

with

$$\langle y|P_{t,t}|x\rangle = \delta(y-x) \quad \textit{(Hence, } P_{t,t}=1\textit{)} \tag{94.105}$$

[5]For those who know Quantum Mechanics, our $\mathcal{F}$ equals a Hamiltonian H times i, $\mathcal{F} = Hi$.

The forward FP equation is the time reversed version of the backward FP equation. Thus, they describe the same stochastic process, in opposite time directions.

Note that if we view a transition matrix $P\binom{y}{u}\binom{x}{t}$ as a matrix whose rows and columns are labeled by all possible events $\binom{x}{t}$, then the forward FP equation (resp., backward FP equation) is a differential equation constraining all the rows $\binom{y}{u}$ for a fixed column $\binom{x}{t}$ (resp., all the columns for a fixed row). They both constrain the same matrix $P\binom{y}{u}\binom{x}{t}$, but in 2 different ways.

Figure 94.4 shows the TPM arrows for the forward FP in red and for the backward FP in blue.

Figure 94.4: Red arrows refer to $P((x,t)|(w,s))$ and $P((x,t+dt)|(w,s))$ for the forward FP equation. Blue arrows refer to $P((y,u)|(x,t))$ and $P((y,u)|(x,t+dt))$ for the backwards FP equation.

94.7 First and second order statistics

In this section, we derive differential equations for the first and second order statistics of an SDE without and with CC.

94.7.1 For general SDE

Suppose the SDE coefficients f and L can depend on the event (x,t).

Let

$$m_\mu(t) = E[\underline{x}_\mu(t)] = \left\langle \underline{x}_\mu(t) \right\rangle \tag{94.106}$$

$$C_{\mu,\nu}(t,s) = \left\langle \underline{x}_\mu(t), \underline{x}_\nu(s) \right\rangle \tag{94.107}$$

$$V_{\mu,\nu}(t) = C_{\mu,\nu}(t,t) \tag{94.108}$$

$$dx_\mu = [a_\mu(t) + F_{\mu,\nu}(t)x_\nu]\,dt + L_{\mu,\nu}(t)d\underline{B}_\nu \tag{94.109}$$

$$R_{\mu,\nu} = \frac{1}{2} L_{\mu,\alpha} Q_{\alpha,\beta} L^T_{\beta,\nu} = \frac{1}{2}(LQL^T)_{\mu,\nu} \tag{94.110}$$

Claim 163

$$\frac{dm}{dt} = a + Fm \tag{94.111}$$

$$\frac{dV}{dt} = VF^T + FV^T + 2R \tag{94.112}$$

proof:

$$\begin{aligned} \frac{dm}{dt} &= \langle a + F\underline{x} \rangle & (94.113) \\ &= a + Fm & (94.114) \end{aligned}$$

Let $\partial_\alpha = \frac{\partial}{\partial x_\alpha}$

$$\begin{aligned} d\left\langle \underline{x}_\mu, \underline{x}_\nu \right\rangle &= dx_\alpha \partial_\alpha \left\langle \underline{x}_\mu, \underline{x}_\nu \right\rangle + dx_\alpha dx_\beta \frac{1}{2} \partial_\alpha \partial_\beta \left\langle \underline{x}_\mu, \underline{x}_\nu \right\rangle & (94.115) \\ &= \left\langle \underline{x}_\mu, d\underline{x}_\nu \right\rangle + \left\langle d\underline{x}_\mu, \underline{x}_\nu \right\rangle + \left\langle d\underline{x}_\mu, d\underline{x}_\nu \right\rangle & (94.116) \\ &= \left\langle \underline{x}_\mu, F_{\nu,\alpha} \underline{x}_\alpha dt \right\rangle + \langle F_{\mu,\alpha} \underline{x}_\alpha dt, \underline{x}_\nu \rangle + 2R_{\mu,\nu} dt & (94.117) \\ &= (VF^T + FV^T + 2R)dt & (94.118) \end{aligned}$$

QED

We call a **propagator**, a function $\Psi : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}$ that satisfies

$$\frac{\partial \Psi(t,s)}{\partial t} = F(t)\Psi(t,s) \tag{94.119}$$

$$\begin{aligned} &\Psi(a,c) = \Psi(a,b)\Psi(b,c) \\ &\Psi^{-1}(a,b) = \Psi(b,a) \\ &\Psi(a,a) = 1 \end{aligned} \tag{94.120}$$

Claim 164

$$m(t) = \Psi(t,t_0)m(t_0) + \int_{t_0}^{t} d\tau \; \Psi(t,\tau)a(t) \tag{94.121}$$

$$V(t) = \left\langle \underline{x}(t), \underline{x}^T(t) \right\rangle = \Psi(t,t_0)V(t_0)\Psi^T(t,t_0) + \int_{t_0}^{t} d\tau \; \Psi(t,\tau)R(\tau)\Psi^T(t,\tau) \tag{94.122}$$

$$\left\langle \underline{x}(t), \underline{x}^T(s) \right\rangle = \begin{cases} V(t)\Psi^T(s,t) & \textit{if } t < s \\ \Psi(t,s)V(s) & \textit{if } t \geq s \end{cases} \tag{94.123}$$

proof: Eqs.(94.121 and (94.122)) can be proven simply by taking the time derivative of both sides. This yields the differential equations for $m(t)$ and $V(t)$ that were established in Claim 163. We won't prove Eq.(94.123) here. For a proof, see Ref.[73]. **QED**

If $P(x, t = 0)$ is a Gaussian, $P(x, t)$ must be Gaussian too, because the transformation is linear. Therefore,

$$P(x,t) = P(x(t)) = \mathcal{N}(x(t); \mu = m(t), \Sigma^2 = V(t)) \tag{94.124}$$

$$P(x(t)|x(s)) = \mathcal{N}(x(t); \mu = m(t|s), \Sigma^2 = V(t|s)) \tag{94.125}$$

where

$$m(t|s) = \Psi(t,s)x(s) + \int_{t_0}^{t} d\tau \Psi(t,\tau)a(\tau) \tag{94.126}$$

$$V(t|s) = \int_{s}^{t} d\tau \ \Psi(t,\tau)R(\tau)\Psi^T(t,\tau) \tag{94.127}$$

This transition matrix $P(x(t)|x(s))$, when we discretize time and set $x(t_k) = x_k$, defines the dynamic bnet of Fig.94.5. Define

$$\Psi_k = \Psi(t_{k+1}, t_k) \tag{94.128}$$

$$a_k = \int_{t_k}^{t_{k+1}} d\tau \ \Psi(t_{k+1},\tau)a(\tau) \tag{94.129}$$

$$\Sigma_k = \int_{t_k}^{t_{k+1}} d\tau \ \Psi(t_{k+1},\tau)R(\tau)\Psi^T(t_{k+1},\tau) \tag{94.130}$$

The TPMs, printed in blue, for the bnet of Fig.94.5, are as follows:

$$P(x_{k+1}|x_k) = \mathbb{1}(x_{k+1} = \Psi_k x_k + a_k + \epsilon_k) \tag{94.131}$$

$$P(\epsilon_k) = \mathcal{N}(\epsilon_k; \mu = 0, \Sigma = \Sigma_k) \tag{94.132}$$

94.7.2 In case SDE has CC

When the SDE has CC,

$$\Psi(t,s) = e^{(t-s)F} \tag{94.133}$$

Figure 94.5: Bnet that satisfies the general SDE, assuming $P(x, t = 0)$ is a Gaussian.

At time $t = \infty$, a system with CC reaches a **steady state** (i.e., zero time derivatives of expected values) with mean value m_∞ and variance V_∞ given by

$$0 = a + Fm_\infty \tag{94.134}$$

$$0 = V_\infty F^T + Fv_\infty^T + 2R \tag{94.135}$$

94.8 Fourier Analysis for CC case

In this section, we will apply Fourier analysis to the SDE with CC.

We begin by recalling a few definitions from Fourier analysis.

The **Dirac delta function** satisfies:

$$\delta(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega\ e^{i\omega t} \tag{94.136}$$

The **Fourier transform** of $x(t)$ is

$$\widetilde{x}(\omega) = \mathcal{F}[x(t)](\omega) = \int_{-\infty}^{\infty} dt\ x(t)e^{-i\omega t} \tag{94.137}$$

The **inverse Fourier transform** of $\widetilde{x}(\omega)$ is

$$x(t) = \mathcal{F}^{-1}[\widetilde{x}(\omega)](t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} d\omega\ \widetilde{x}(\omega)e^{i\omega t} \tag{94.138}$$

Define the **bounded-time Fourier transform** $\widetilde{x}_T(\omega)$ by

$$\widetilde{x}_T(\omega) = \int_{-\frac{T}{2}}^{\frac{T}{2}} dt\ x(t)e^{-i\omega t} \tag{94.139}$$

Define the **power spectral density** of process $x(t)$ by

$$val(\underline{x})(\omega) = \lim_{T\to 0}\frac{1}{T}E[\widetilde{x}_T(\omega)\widetilde{x}_T^\dagger(\omega)] \tag{94.140}$$

In case $\underline{x} = \underline{W}$= white noise,

$$S_{\underline{W}}(\omega) = Q \tag{94.141}$$

Define the **autocorrelation function for stationary process** $\underline{x}(t)$ by

$$AC_{\underline{x}}(\tau) = E[x(t)x^T(t+\tau)] \tag{94.142}$$

τ is called the **time lag**.

Claim 165

$$\int_{-\frac{T}{2}}^{\frac{T}{2}} dt \int_{-\frac{T}{2}}^{\frac{T}{2}} ds\ g(t-s) = \int_{-T}^{T} d\tau\ g(\tau)(T-|\tau|) \tag{94.143}$$

proof:

Let

$$\tau = \frac{t-s}{\sqrt{2}}, \quad \sigma = \frac{t+s}{\sqrt{2}} \tag{94.144}$$

The absolute value of the Jacobian $|\frac{\partial(s,t)}{\partial(\tau,\sigma)}|$ equals 1. Hence,

$$\int_{-\frac{T}{2}}^{\frac{T}{2}} dt \int_{-\frac{T}{2}}^{\frac{T}{2}} ds\ g(t-s) = \int_{-\frac{T}{\sqrt{2}}}^{\frac{T}{\sqrt{2}}} d\tau \int_{-\frac{T}{\sqrt{2}}-|\tau|}^{\frac{T}{\sqrt{2}}+|\tau|} d\sigma\ g(\sqrt{2}\tau) \quad \text{(See Fig. 94.6)} \tag{94.145}$$

$$= \int_{-\frac{T}{\sqrt{2}}}^{\frac{T}{\sqrt{2}}} d\tau\ g(\sqrt{2}\tau)(\sqrt{2}T - 2|\tau|) \tag{94.146}$$

$$= \int_{-T}^{T} d\tau'\ g(\tau')(T-|\tau'|) \quad (\tau' = \sqrt{2}\tau) \tag{94.147}$$

Figure 94.6: Integration region for integral given by Eq.(94.143)

QED

Claim 166 *(Wiener Khinchin theorem (WK))*

$$val(\underline{x})(\omega) = \int_{-\infty}^{\infty} d\tau\ AC_{\underline{x}}(\tau)e^{i\omega\tau} \tag{94.148}$$

If $AC(\tau) = AC(-\tau)$,

$$val(\underline{x})(\omega) = 2\int_{0}^{\infty} d\tau\ AC_{\underline{x}}(\tau)\cos(\omega\tau) \tag{94.149}$$

proof:

$$\int_{-\infty}^{\infty} d\tau\ AC_{\underline{x}}(\tau)e^{i\omega\tau} = \int_{-\infty}^{\infty} d\tau\ \left\langle x(t)x^{\dagger}(t+\tau)\right\rangle e^{i\omega\tau} \tag{94.150}$$

$$= \int_{-\infty}^{\infty} d\tau \int_{-\infty}^{\infty} \frac{d\omega_1}{2\pi} \int_{-\infty}^{\infty} \frac{d\omega_2}{2\pi} \left\langle \widetilde{x}(\omega_1)\widetilde{x}^{\dagger}(\omega_2)\right\rangle e^{i\omega\tau + i\omega_1 t - i\omega_2(t+\tau)} \tag{94.151}$$

To get rid of the t dependence on the right hand side of the last equation, apply $\frac{1}{T}\int_{-T/2}^{T/2} dt$ to both sides to get

$$\int_{-\infty}^{\infty} d\tau\ AC_{\underline{x}}(\tau)e^{i\omega\tau} = \int_{-\infty}^{\infty} \frac{d\tau}{T} \int_{-\infty}^{\infty} \frac{d\omega_1}{2\pi} \int_{-\infty}^{\infty} d\omega_2 \left\langle \widetilde{x}(\omega_1)\widetilde{x}^{\dagger}(\omega_1)\right\rangle e^{i(\omega-\omega_2)\tau}\delta(\omega_1-\omega_2) \tag{94.152}$$

$$= \int_{-\infty}^{\infty} \frac{d\tau}{T} \int_{-\infty}^{\infty} \frac{d\omega_1}{2\pi} \left\langle \widetilde{x}(\omega_1)\widetilde{x}^{\dagger}(\omega_1)\right\rangle e^{i(\omega-\omega_1)\tau} \tag{94.153}$$

$$= \frac{1}{T}\int_{-\infty}^{\infty} d\omega_1 \left\langle \widetilde{x}(\omega_1)\widetilde{x}^{\dagger}(\omega_1)\right\rangle \delta(\omega-\omega_1) \tag{94.154}$$

$$= \frac{1}{T}\left\langle \widetilde{x}(\omega)\widetilde{x}^{\dagger}(\omega)\right\rangle = \frac{1}{T}\left\langle \widetilde{x}(\omega)\widetilde{x}^{T}(\omega)\right\rangle \tag{94.155}$$

Alternative proof:

$$\frac{1}{T}E[|\widetilde{x}_T(\omega)|^2] = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} ds \int_{-\frac{T}{2}}^{\frac{T}{2}} dt\ E[x(t)x^T(s)]e^{-iw(t-s)} \tag{94.156}$$

$$= \frac{1}{T}\int_{-T}^{T} d\tau\ AC(\tau)e^{i\omega\tau}(T-|\tau|) \tag{94.157}$$

$$\rightarrow \int_{-\infty}^{\infty} d\tau\ AC(\tau)e^{i\omega\tau} \quad \text{(Use Eq.(94.143))} \tag{94.158}$$

QED

As a trivial example of the WK theorem, note that

$$AC_{\underline{W}}(\tau) = Q\delta(\tau) \tag{94.159}$$

Hence, by the WK theorem,

$$val(\underline{x})(\omega) = Q \tag{94.160}$$

Next, let us solve the SDE with CC using Fourier transforms. Substituting Fourier transforms for $x(t)$ into this equation

$$\frac{dx}{dt} = Fx + L\underline{W} \tag{94.161}$$

we get

$$-i\omega\widetilde{x} = F\widetilde{x} + L\widetilde{\underline{W}} \tag{94.162}$$

Hence,

$$\widetilde{x} = -(F + i\omega)^{-1} L\widetilde{\underline{W}} \tag{94.163}$$

$$\begin{aligned} S_{\underline{W}}(\omega) &= \widetilde{x}\widetilde{x}^{\dagger} && (94.164) \\ &= (F + i\omega)^{-1} \underbrace{LQL^{\dagger}}_{2R} (F^T - i\omega)^{-1} && (94.165) \end{aligned}$$

$$AC_{\underline{x}}(\tau) = \int_{-\infty}^{\infty} d\omega \; e^{i\omega\tau} (F + i\omega)^{-1} (2R)(F^T - i\omega)^{-1} \tag{94.166}$$

When steady state is reached, the expected values (averages) of functions of $\underline{x}(t)$ cease to vary with time. Thus, we only need to compare $\underline{x}(t)$ to itself, not to $\underline{x}(t+\tau)$ with $\tau > 0$. Hence, we only need to know $AC(\tau)$ at $\tau = 0$.

For steady state, by the Wiener Khinchin theorem,

$$val(\underline{x})(\omega) = AC_{\underline{x}}(0) 2\pi\delta(w) \tag{94.167}$$

where

$$AC_{\underline{x}}(0) = \int_{-\infty}^{\infty} dw \; (F + i\omega)^{-1} (2R)(F^T - i\omega)^{-1} \tag{94.168}$$

94.9 Lamperti Transformation

The **Lamperti Transformation** answers the following question: If the next two SDE are satisfied, express g_μ in terms of f_μ and $L_{\mu,\nu}$

$$dx_\mu = f_\mu(x,t)dt + L_{\mu,\nu}(x,t)d\underline{B}_\nu \tag{94.169}$$

$$dy_\mu = g_\mu(y,t)dt + d\underline{B}_\mu \tag{94.170}$$

Claim 167 *For $n = 1$, the function $g(y,t)$ in Eq.(94.170) is given by*

$$g(y,t) = \left(\frac{\partial}{\partial t} \int_{\xi}^{x} \frac{du}{L_t^u} + \frac{f_t^x}{L_t^x} - \frac{1}{2} \frac{\partial L_t^x}{\partial x} \right) \Bigg|_{x \to h^{-1}(y,t)} \tag{94.171}$$

proof: Suppose

$$y = h \stackrel{\text{def}}{=} \int_{\xi}^{x} \frac{du}{L(u,t)} \tag{94.172}$$

Then

$$\begin{aligned} dy &= \frac{\partial h}{\partial t} dt + \frac{dx}{L} - \frac{1}{2L^2}\frac{\partial L}{\partial x}(dx)^2 && (94.173) \\ &= \frac{\partial h}{\partial t} dt + \frac{f dt + L d\underline{B}}{L} - \frac{1}{2L^2}\frac{\partial L}{\partial x}\underbrace{(L d\underline{B})^2}_{L^2 dt} && (94.174) \\ &= \left(\frac{\partial h}{\partial t} + \frac{f}{L} - \frac{1}{2}\frac{\partial L}{\partial x}\right) dt + d\underline{B} && (94.175) \end{aligned}$$

QED

94.10 Feynman-Kac Path Integrals

Kac path integrals (Kac PI) are weighted sums over paths. Kac FPs are solutions of a classical SDE for specific boundary conditions. They were first proposed by Kac.

Feynman path integrals (Feynman PI) are similar to Kac PI, but the weights of the sum are complex valued instead of real valued. Feynman FPs are solutions of a quantum SDE (i.e., a Schroedinger equation) for specific boundary conditions. They were first proposed by Feynman, who wrote this book about them: Ref.[20])

Despite its name, this section will deal only with Kac PI. Furthermore, we will only consider the one dimensional case $\underline{x} \in \mathbb{R}$. The higher dimensional case is similar.

Figure 94.7: Bnet for defining 1-dim Kac PI.

Let $x(t_k) = x_k$ and consider the bnet of Fig.94.7. The TPMs, printed in blue, for that bnet, are as follows

$$P(x_k|x_{k-1}) = \mathbb{1}(x_{k-1} + f_k \Delta t + \Delta \underline{B}_k) \tag{94.176}$$

$$\Delta \underline{B}_k \sim \mathcal{N}(\mu = 0, \sigma^2 = q\Delta t) \tag{94.177}$$

Then

$$P(x_{[1-N]}|x_0) = \prod_{k=1}^{N} P(x_k|x_{k-1}) \tag{94.178}$$

$$= \prod_{k=1}^{N} \left[\frac{1}{\sqrt{2\pi q \Delta t}} \exp\left(- \frac{(x_k - x_{k-1})^2}{2q\Delta t} \right) \right] \tag{94.179}$$

$$= \prod_{k=1}^{N} \left[\frac{1}{\sqrt{2\pi q \Delta t}} \exp\left(- \frac{(f_k \Delta t + \Delta B_k)^2}{2q\Delta t} \right) \right] \tag{94.180}$$

$$= \underbrace{\prod_{k=1}^{N} \left[\frac{1}{\sqrt{2\pi q \Delta t}} \right]}_{\gamma^N} \exp\left(- \frac{1}{2q} \int_0^{t_N} dt \left[(\ddot{B})^2 + 2f\dot{B} + f^2 \right] \right) \tag{94.181}$$

$$P(B_{[0-N]}) = \gamma^N \exp\left(- \frac{1}{2q} \int_0^{t_N} dt (\ddot{B})^2 \right) \tag{94.182}$$

$$\frac{P(x_{[1-N]})}{P(B_{[1-N]})} = \exp\left(- \frac{1}{2q} \int_0^{t_N} dt \left[2f\dot{B} + f^2 \right] \right) \tag{94.183}$$

$$\mathcal{D}B = \gamma^N \prod_{k=1}^{N} dB_k \tag{94.184}$$

Note that $(B_k, t_k) \in \mathbb{R}^n \times [0, T]$ for $k \in [0 - N]$. Let $\mathcal{E} \subset \mathbb{R}^n \times [0, T]$. Fig.94.8 shows a possible set $\mathcal{E}$ when $n = 1$. The path integration is over all paths (B_k, t_k) for $k \in [0 - N]$ that live inside $\mathcal{E}$.

It follows that

$$P(x_{N+1}|x_0 = 0) = \int_{\mathcal{E}} \mathcal{D}B \ \exp\left(\frac{1}{2q} \int_0^{t_N} dt \left[(\ddot{B})^2 + 2f\dot{B} - f^2 \right] \right) \tag{94.185}$$

94.11 Karhunen–Loève series

In this section we explain the **Karhunen–Loève (KL) series**. The most intuitive way of explaining KL series is using Dirac bra-ket notation.

Let $\{|t\rangle : t \in [0, T]\}$ be a complete orthonormal basis

$$\int_0^T dt \ |t\rangle\langle t| = 1, \ \langle t|t'\rangle = \delta(t - t') \tag{94.186}$$

Express $\underline{x}(t)$ in bra-ket notation

$$\underline{x}(t) = \langle t|\underline{x}\rangle \,, \quad \underline{x}^*(t) = \langle \underline{x}|t\rangle \,, \tag{94.187}$$

Figure 94.8: The pink area is a possible set $\mathcal{E} \subset \mathbb{R} \times [0, T]$ such that $(B_k, t_k) \in \mathcal{E}$

Consider the operator

$$C = E[|\underline{x}\rangle\langle\underline{x}|] - E[|\underline{x}\rangle]E[\langle\underline{x}|] \tag{94.188}$$

The matrix elements of C are:

$$C(t, t') = \langle t|C|t'\rangle = E[\underline{x}(t)\underline{x}^*(t')] - E[\underline{x}(t)]E[\underline{x}^*(t')] \tag{94.189}$$

Hence, $C(t, t')$ is the correlation matrix in the $\{|t\rangle : t \in [0, T]\}$ basis.

Suppose C has eigenvalue, eigenvector pairs $(\lambda_n, |\phi_n\rangle)$ for $n = 1, 2, 3, \ldots$. Hence

$$C|\phi_n\rangle = \lambda_n|\phi_n\rangle \tag{94.190}$$

where the states $|\phi_n\rangle$ for all n, are a complete orthonormal basis. Then

$$\sum_{n=1}^{\infty} |\phi_n\rangle\langle\phi_n| = 1, \ \langle\phi_n|\phi_{n'}\rangle = \delta(n, n') \tag{94.191}$$

$$C = \sum_{n=1}^{\infty} |\phi_n\rangle\lambda_n\langle\phi_n| \tag{94.192}$$

If we define

$$\phi_n(t) = \langle t|\phi_n\rangle \, , \ \phi_n^*(t) = \langle\phi_n|t\rangle \tag{94.193}$$

$$\underline{x}_n^* = \langle\underline{x}|\phi_n\rangle \, , \ \underline{x}_n = \langle\phi_n|\underline{x}\rangle \tag{94.194}$$

then

$$|\underline{x}\rangle = \sum_n \underline{x}_n|\phi_n\rangle \tag{94.195}$$

Note also that both

$$\langle \phi_n | C | \phi_m \rangle = E[\underline{x}_n \underline{x}_m^*] - E[\underline{x}_n] E[\underline{x}_m^*] \tag{94.196}$$

and

$$\langle \phi_n | C | \phi_m \rangle = \lambda_n \delta(n, m) \tag{94.197}$$

are satisfied. Therefore,

$$\boxed{E[\underline{x}_n \underline{x}_m^*] - E[\underline{x}_n] E[\underline{x}_m^*] = \lambda_n \delta(n, m)} \tag{94.198}$$

Claim 168 *The Karhunen–Loève expansion for Brownian motion $\underline{B}(t)$ is*

$$C = \sum_n |\phi_n\rangle \lambda_n \langle \phi_n| \tag{94.199}$$

for $n = 0, 1, 2 \ldots$ where

$$\underline{B}(t) = \sum_n z_n \langle t | \phi_n \rangle \tag{94.200}$$

$$\langle t | \phi_n \rangle = \sqrt{\frac{2}{T}} \sin \omega_n t \tag{94.201}$$

$$\omega_n = \frac{\pi}{T}(n + \frac{1}{2}) \tag{94.202}$$

$$\lambda_n = \frac{1}{\omega_n^2} \tag{94.203}$$

$$z_n \sim \mathcal{N}(\mu = 0, \sigma^2 = \lambda_n) \tag{94.204}$$

proof:

$$E[\underline{B}_t \underline{B}_s] - \underbrace{E[\underline{B}_t] E[\underline{B}_s]}_{0} = E[\underline{B}_t \underline{B}_s] = \min(t, s) \tag{94.205}$$

$$\int_0^T dt \ \min(s, t) \phi_n(t) = \lambda_n \phi_n(s), \ 0 \le s \le T \tag{94.206}$$

$$\int_0^T dt \ \min(s, t) \sin \omega_n t = \int_0^s dt \ t \sin \omega_n t + \int_s^T dt \ s \sin \omega_n t \tag{94.207}$$

$$\int_0^s dt \ t \sin \omega_n t = \left. \frac{\sin \omega_n t}{\omega_n^2} \right|_{t=0}^s - \left. \frac{t \cos \omega_n t}{\omega_n} \right|_{t=0}^s \tag{94.208}$$

$$= \frac{\sin \omega_n s}{\omega_n^2} - \frac{s \cos \omega_n s}{\omega_n} \tag{94.209}$$

$$\int_s^T dt\ s \sin \omega_n t = -\ \frac{s \cos \omega_n t}{\omega_n}\bigg|_{t=s}^T = \frac{s \cos \omega_n s}{\omega_n} \tag{94.210}$$

Hence

$$\int_0^T dt\ \min(s,t) \sin \omega_n t = \frac{\sin \omega_n s}{\omega_n^2} \tag{94.211}$$

The $\sqrt{\frac{2}{T}}$ factor in the definition of $\phi_n(t)$ is necessary to insure that $\langle \phi_n | \phi_n \rangle = 1$.

QED

94.12 Girsamov Theorem

In this section, we explain the **Grisamov theorem**. We divide the explanation into 3 parts.

Suppose

$$dx = f(x,t)dt + d\underline{B} \tag{94.212}$$

$$dy = g(y,t)dt + d\underline{\beta} \tag{94.213}$$

and

$$dx = dy \tag{94.214}$$

Claim 169 *(Girsanov theorem, part 1)*

$$d\underline{\beta} = (f-g)dt + d\underline{B} \tag{94.215}$$

proof: Just subtract Eq.(94.212) from Eq.(94.213) .
QED

Claim 170 *(Girsanov theorem, part 2)*

$$\frac{P(x_{[1-N]})}{P(y_{[1-N]})} = \exp\left(-\ \frac{1}{2q} \int_0^{t_N} dt\ (f-g)^2 - \frac{1}{2q} \int 2(f-g) d\underline{B} \right) \tag{94.216}$$

proof:

From Eq.(94.183)

$$\frac{P(x_{[1-N]})}{P(y_{[1-N]})} = \exp\left(-\ \frac{1}{2q} \int_0^{t_N} dt\ \underbrace{\left[2f\dot{B} + f^2 - 2g\dot{\beta} - g^2 \right]}_{\mathcal{A}} \right) \tag{94.217}$$

$$\begin{aligned}
\mathcal{A} &= 2f\underline{\dot{B}} - 2g[\underline{\dot{B}} + f - g] + f^2 - g^2 && (94.218)\\
&= (f-g)2\underline{\dot{B}} + (f-g)(-2g) + (f-g)(f+g) && (94.219)\\
&= (f-g)[2\underline{\dot{B}} - 2g + f + g] && (94.220)\\
&= (f-g)^2 + 2\underline{\dot{B}}(f-g) && (94.221)
\end{aligned}$$

QED

Claim 171 *(Girsanov theorem, part 3)*

If

$$Z = \frac{P(x_{[1-N]})}{P(y_{[1-N]})} \tag{94.222}$$

then

$$E[h(x_{[1-N]})] = E[Zh(y_{[1-N]})] \tag{94.223}$$

proof:

$$\begin{aligned}
E[h(\underline{x}_{[1-N]})] &= \int \underbrace{dx_{[1-N]}}_{dy_{[1-N]}} P(x_{[1-N]}) \underbrace{h(x_{[1-N]})}_{h(y_{[1-N]})} && (94.224)\\
&= \int dy_{[1-N]} ZP(y_{[1-N]})h(y_{[1-N]}) && (94.225)\\
&= E[Zh(y_{[1-N]})] && (94.226)
\end{aligned}$$

94.13 Doob's Transform

In this section, we explain **Doob's transform**.

We begin by assuming that a function $D(.|_t^x)$ [6] [7] satifies the equation

$$D(.|_t^x) = \int dy\ D(.|_{t+s}^y) P({}_{t+s}^y|_t^x) \tag{94.227}$$

Next we define a function $P^D({}_{t+s}^y|_t^x)$ by

$$P^D({}_{t+s}^y|_t^x) = \frac{D(.|_{t+s}^y) P({}_{t+s}^y|_t^x)}{D(.|_t^x)} \tag{94.228}$$

Note that $P^D({}_{t+s}^y|_t^x)$ is a conditional probability as its symbol suggests.

$$\int dy\ P^D({}_{t+s}^y|_t^x) = 1 \tag{94.229}$$

[6] Sometimes, $D(.|_t^x)$ is denoted by the letter h, and this transform is called Doob's h-transform.

[7] Later on, we will see that the conditional probability $P({}_T^x|_t^x) = D(.|_t^x)$ satisfies Eq.(94.227).

Recall that

$$\mathcal{F}_{\underline{x}}\bullet = -\frac{\partial}{\partial x_\mu}(\bullet f_\mu) + \frac{\partial^2}{\partial x_\mu \partial x_\nu}(\bullet R_{\mu,\nu}) \tag{94.230}$$

and

$$\mathcal{B}_{\underline{x}}\bullet = f_\mu \frac{\partial \bullet}{\partial x_\mu} + R_{\mu,\nu}\frac{\partial^2 \bullet}{\partial x_\mu \partial x_\nu} \tag{94.231}$$

If

$$\left[\frac{\partial}{\partial s} + \mathcal{B}_{\underline{y}}\right]\phi_s^y = 0, \tag{94.232}$$

then that implies we can define a process $y(s)$ such that

$$dy_\mu = f_\mu(^y_s)ds + L_{\mu,\nu}(^y_s)d\underline{B}_\nu \tag{94.233}$$

with

$$P(^z_t|^y_s) = \phi_s^y \tag{94.234}$$

Claim 172 *(Doob's Transform)*

Let $\mathcal{F}_{\underline{y}}^D$ (resp., $\mathcal{B}_{\underline{y}}^D$) be the same as $\mathcal{F}_{\underline{y}}$ (resp., $\mathcal{B}_{\underline{y}}$), but with $f_\mu(^y_t)$ replaced by $f_\mu^D(^y_t)$ given by

$$f_\mu^D(^y_t) = f_\mu(^y_t) + 2R_{\mu,\nu}\frac{\partial \ln D(.|^y_t)}{\partial y_\nu} \tag{94.235}$$

Suppose

$$\left[\frac{\partial}{\partial s} + \mathcal{B}_{\underline{y}}\right] D(.|^y_s) = 0, \tag{94.236}$$

and

$$\left[\frac{\partial}{\partial s} + \mathcal{B}_{\underline{y}}^D\right] P(^y_{t+s}|^x_t) = 0 \tag{94.237}$$

Then

$$\left[\frac{\partial}{\partial s} + \mathcal{B}_{\underline{y}}\right] P^D(^y_{t+s}|^x_t) = 0 \tag{94.238}$$

proof: For conciseness, define $P_s = P(^y_{t+s}|^x_t)$, $D_s = D(.|^y_{t+s})$, $D_0 = D(.|y_t)$, and $\partial_\mu = \frac{\partial}{\partial y_\mu}$, $\partial_\nu = \frac{\partial}{\partial y_\nu}$, $\partial_s = \frac{\partial}{\partial s}$

$$(\partial_s + \mathcal{B}_{\underline{y}})P_s^D = \frac{1}{D_0}\left[\partial_s(D_sP_s) + f_\mu\partial_\mu(D_sP_s) + R_{\mu,\nu}\partial_\mu\partial_\nu(D_sP_s)\right] \tag{94.239}$$

$$\partial_\mu \partial_\nu (D_s P_s) = \begin{cases} \partial_\mu \partial_\nu (D_s) P_s \\ +\partial_\nu (D_s) \partial_\mu (P_s) \\ +\partial_\mu (D_s) \partial_\nu (P_s) \\ +D_s \partial_\mu \partial_\nu (P_s) \end{cases} \tag{94.240}$$

$$(\partial_s + \mathcal{B}_{\underline{y}}) P_s^D = \begin{cases} \frac{P_s}{D_s} \cancel{\left[\partial_s + \mathcal{B}_{\underline{y}}\right] D_s} \\ +\frac{1}{D_0} \left[D_s \partial_s (P_s) + f_\mu D_s \partial_\mu P_s\right] \\ +R_{\mu,\nu} \begin{cases} \partial_\nu (D_s) \partial_\mu (P_s) \\ +\partial_\mu (D_s) \partial_\nu (P_s) \\ +D_s \partial_\mu \partial_\nu (P_s) \end{cases} \end{cases} \tag{94.241}$$

$$(\partial_s + \mathcal{B}_{\underline{y}}) P_s^D = \begin{cases} \frac{D_s}{D_0} \left[\partial_s P_s + f_\mu \partial_\mu P_s + R_{\mu,\nu} \partial_\mu \partial_\nu P_s\right] \\ +R_{\mu,\nu} \partial_\nu (D_s) \partial_\mu (P_s) \end{cases} \tag{94.242}$$

$$= \frac{D_s}{D_0} \left[\partial_s P_s + \left[f_\mu + 2R_{\mu,\nu} \partial_\nu (\ln D_s)\right] \partial_\mu P_s + R_{\mu,\nu} \partial_\mu \partial_\nu P_s\right] \tag{94.243}$$

$$= \frac{D_s}{D_0} \left[\partial_s P_s + f_\mu^D \partial_\mu P_s + R_{\mu,\nu} \partial_\mu \partial_\nu P_s\right] \tag{94.244}$$

$$= \frac{D_s}{D_0} (\partial_s + \mathcal{B}_{\underline{y}}^D) P_s \tag{94.245}$$

QED

Let $A = \binom{x}{t+s}$, $B = \binom{x}{T}$. Bayes Rule says

$$P(A|B, \tbinom{x}{t}) = \frac{P(B|A, \tbinom{x}{t}) P(A|\tbinom{x}{t})}{P(B|\tbinom{x}{t})} \tag{94.246}$$

Hence,

$$P(\tbinom{x}{t+s}|\tbinom{x}{T}, \tbinom{x}{t}) = \frac{P(\tbinom{x}{T}|\tbinom{x}{t+s}, \tbinom{x}{t}) P(\tbinom{x}{t+s}|\tbinom{x}{t})}{P(\tbinom{x}{T}|\tbinom{x}{t})} \tag{94.247}$$

$$= \frac{P(\tbinom{x}{T}|\tbinom{x}{t+s}) P(\tbinom{x}{t+s}|\tbinom{x}{t})}{P(\tbinom{x}{T}|\tbinom{x}{t})} \tag{94.248}$$

If we set

$$D(.|\tbinom{x}{t+s}) = P(\tbinom{x}{T}|\tbinom{x}{t+s}) \tag{94.249}$$

then Claim 172 applies.

94.14 Appendix: Some explicitly solvable examples

Let $\mu, \nu, \alpha, \beta \in [n]$

$$d\underline{x}_\mu = f_\mu(\underline{x}, t)dt + L_{\mu,\nu}(\underline{x}, t)d\underline{B}_\nu(t) \tag{94.250}$$

$$\frac{\partial P}{\partial t} = -\frac{\partial P f_\mu}{\partial x_\mu} + \frac{\partial^2}{\partial x_\mu \partial x_\nu}(PR_{\mu,\nu}) \tag{94.251}$$

- **Brownian motion** ($f_\mu = 0$, $R_{\mu,\nu} = D\delta(\mu,\nu)$)

$$dx_\mu = d\underline{B}_\mu \tag{94.252}$$

- **Overdamped Langevin Equation** ($f_\mu = -\frac{1}{2}\frac{\partial U}{\partial x_\mu}$, $R_{\mu,\nu} = D\delta(\mu,\nu)$)

$$dx_\mu = -\frac{1}{2}\frac{\partial U}{\partial x_\mu}dt + d\underline{B}_\mu \tag{94.253}$$

- **Ornstein–Uhlenbeck process (a.k.a. Langevin equation)** ($f_\mu = -\lambda x_\mu$, $R_{\mu,\nu} = D\delta(\mu,\nu)$)

  This is the same as the Langevin equation, if identify $\underline{x}$ with the velocity of the particle and λ with the drag coefficient.

$$d\underline{x} = -\lambda \underline{x}dt + d\underline{B} \tag{94.254}$$

- **1-dim ($n = 1$) Black-Sholes** ($f = ax$, $R = (b\underline{x})^2 q/2$)

$$d\underline{x} = a\underline{x}dt + b\underline{x}d\underline{B} \tag{94.255}$$

- **1-dim ($n = 1$) SDE with ($f = ax + c$, $R = (b\underline{x} + d)^2 q/2$)**

$$d\underline{x} = [a(t)\underline{x} + c(t)]dt + [b(t)\underline{x} + d(t)]d\underline{B} \tag{94.256}$$

$$x(t) = \Psi(t, t_0)\left(x(t_0) + \int_{t_0}^t ds\ \Psi^{-1}(s, t_0)[c(s) - b(s)] + \int_{t_0}^t \Psi^{-1}(s, t_0)d(s)d\underline{W}(s)\right) \tag{94.257}$$

$$\Psi(t, t_0) = \exp\left(\int_{t_0}^t ds\ \left[a(s) - \frac{1}{2}b^2(s)\right] + \int_{t_0}^t b(s)d\underline{W}(s)\right) \tag{94.258}$$

94.15 Appendix: Ornstein-Uhlenbeck recurring example

SDE

$$dx = -\lambda x dt + d\underline{B} \tag{94.259}$$

$P(x_t)$, Mean and Variance

$$P(x_t) = \mathcal{N}(x_t; \mu = \langle \underline{x}_t \rangle, \sigma^2 = \langle \underline{x}_t, \underline{x}_t \rangle) \tag{94.260}$$

$$m(t) = \langle \underline{x}_t \rangle = x_0 e^{-\lambda t} \tag{94.261}$$

$$V(t) = \langle \underline{x}_t, \underline{x}_t \rangle = \frac{q}{2\lambda}(1 - e^{-\lambda t}) \tag{94.262}$$

transition matrix, $P(^{x}_{t}|^{x}_{s})$, conditional mean and variance

$$P(^{x}_{t}|^{x}_{s}) = \mathcal{N}(x_t; \mu = m(t|s), \sigma^2 = V(t|s)) \tag{94.263}$$

$$m(t|s) = x_s e^{-\lambda(t-s)} \tag{94.264}$$

$$V(t|s) = \frac{q}{2\lambda}(1 - e^{-2\lambda(t-s)}) \tag{94.265}$$

Power Spectrum, Autocorrelation

$$S(\omega) = \frac{q}{\omega^2 + \lambda^2} \tag{94.266}$$

$$AC(\tau) = \frac{q}{2\lambda} e^{-\lambda|\tau|} \tag{94.267}$$

steady state covariance

$$V_\infty = \langle x_\infty, x_\infty \rangle = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \; S(\omega) = \frac{q}{2\lambda} \tag{94.268}$$

Doob's transform

$$D(^{x}_{T}|^{x}_{t}) = \mathcal{N}(x_T; \mu = a(t) x_t, \sigma^2 = \sigma^2(t)) \tag{94.269}$$

$$a(t) = e^{-\lambda(T-t)} \tag{94.270}$$

$$\sigma^2(t) = \frac{q}{2\lambda}[1 - e^{-2\lambda(T-t)}] \tag{94.271}$$

$a(0) = e^{-\lambda T}$, $a(T) = 1$. $\sigma^2(T) = 0$. $D(^{x}_{T}|^{x}_{T}) = \mathcal{N}(x_T; \mu = x_T, \sigma^2 = 0)$

$$dx = \left[-\lambda x + \frac{qa}{\sigma^2}[x_T - ax] \right] dt + d\underline{B} \tag{94.272}$$

Chapter 95

Structure and Parameter Learning for Bnets

Learning a bnet from data is a computationally intensive NP-complete problem. Therefore, the best one can hope for is for heuristic algorithms that solve this problem approximately. A huge number of such algorithms have been tried and continue to be tried. Luckily, there exists a free open source software library called `bnlearn` that covers many of them. The goal of this chapter is to give a brief overview of the subject of bnet learning, after which we recommend to those readers who want to pursue this subject further, to learn `bnlearn` .

This chapter is based on the `bnlearn` website Ref.[75], and on a 2019 survey paper [76] by Scutari et al. I highly recommend looking at both. Refs. [9] and [44] were also helpful to me in understanding this subject.

`bnlearn` (Ref.[75]) (free, open source) is very comprehensive and well maintained. It is written mostly in C with an R front-end. It was developed by Marco Scutari and collaborators over a time period of more than 10 years, and is still under active development. How things stand in the field of bnet learning software reminds me of how things stand in the field of linear algebra (LA) software. Perfecting and optimizing LA software takes many years so I would not advise you to write your own LA software library starting from scratch. There is no need to do so. Instead, you can use LAPACK (free, open source), which has been perfected and expanded for decades by world experts. I view `bnlearn` as the LAPACK of bnet learning.

95.1 Overview

To give the reader an overview of the subject and of `bnlearn` itself, here is a highly simplified tree, compiled from the `bnlearn` website and documentation, of some of the subjects covered by `bnlearn` .

```
│ Parameter Learning
│  └─ missing data
└─ Structure Learning
```

```
│   ├── tree-like structures given a priori
│   │   ├── Naive Bayes
│   │   ├── Chow-Liu tree
│   │   ├── Tree Augmented Naive Bayes (TAN)
│   │   └── ARACNE
│   ├── score based
│   │   ├── algorithms
│   │   │   ├── hill climbing (HC)
│   │   │   ├── HC with random restarts
│   │   │   ├── HC with Tabu list (Tabu)
│   │   │   ├── simulated annealing
│   │   │   └── genetic algorithms
│   │   ├── scoring functions
│   │   │   ├── Information Theoretic scores
│   │   │   ├── Bayesian Information Criterion (BIC)
│   │   │   └── Bayesian Dirichlet (BD) family
│   ├── constraint based
│   │   ├── algorithms
│   │   │   ├── PC family
│   │   │   ├── Grow-Shrink (GS)
│   │   │   └── Incremental Association Markov Blanket (IAMB) family
│   │   ├── conditional independence tests
│   │   │   ├── mutual information (parametric, semiparametric and permutation
│   │   │   │   tests)
│   │   │   └── shrinkage-estimate for the mutual information
│   ├── hybrid
│   │   ├── Max-Min Hill Climbing (MMHC)
│   │   ├── Hybrid HPC (H2PC)
│   │   └── General 2-Phase Restricted Maximization (RSMAX2)
├── parallel mode structure learning
├── node types
│   ├── all-discrete
│   ├── all-continuous
│   └── mixed
└── utility functions
    ├── model comparison and manipulation
    ├── random data generation
    ├── arc orientation testing
    ├── simple and advanced plots
    ├── parameter estimation (maximum likelihood and Bayesian)
    ├── inference, conditional probability queries
    └── cross-validation
```

```
bootstrap
model averaging
```

Let

- PL=parameters learning (i.e, learning the TPMs)
- SL= structure learning (i.e., learning the DAG)
- ML= model (or bnet) learning, SL+PL

PL is easy, once the structure is known. PL assuming no missing data goes as follows. Using the notation of Chapter 88, define

$$\pi^i_{k|\mu} = P(\underline{x}_i = k \mid pa(\underline{x}_i) = \mu) \ . \tag{95.1}$$

Then $\pi^i_{k|\mu}$ can be estimated from the data $N^i_{k,\mu}$ using:

$$\pi^i_{k|\mu} \approx N^i_{k|\mu} = \frac{N^i_{k,\mu}}{N^i_{+,\mu}} \ . \tag{95.2}$$

PL described by Eq.(95.2) is only for discrete nodes with no missing data. `bnlearn` can also do PL with missing data and continuous (Gaussian linear only) nodes. See Chapter 63 on missing data and Chapter 34 on Gaussian linear nodes. SL actually does PL and SL at the same time.

There are 3 main types of SL: score based, constraint based, and hybrid. `bnlearn` can perform many algorithms of each of these 3 types of SL. It can perform most of them with either all-discrete, or all-continuous or mixed nodes. It can perform many of them in parallel mode. The 2019 survey paper Ref.[76] by Scutari et al compares the performance of many different bnet learning algorithms.

95.2 Score based SL algorithms

Score based SL algorithms require scoring bnets (with either all-discrete, all-continuous or mixed nodes). See Chapter 88 for an introduction to scoring bnets. The BIC score explained in that chapter is very popular and works for all-discrete, all-continuous or mixed nodes.

Score-based SL algorithms apply standard optimisation techniques. In the Hill Climbing algorithm, the current best bnet is changed slightly and then given a score that measures how well it fits the data. The bnet with the highest (=best) score so far, as well as that highest score, are stored. (Hence, this is called a greedy search). The process continues until the latest highest score stops changing. The problem with being greedy all the time is that the answer might converge to a local maximum. To mitigate this problem and allow some probability of visiting more than one local maximum, one uses a Tabu Table, random restarts, simulated annealing, genetic algorithms, etc.

95.3 Constraint based SL algorithms

To fully understand constraint based SL algorithms, the reader is advised to read Chapters 24 and 71 first.

Constraint based SL algorithms require estimating from the data the conditional independence $\underline{x}. \perp_P \underline{y}.|\underline{a}.$ for any 3 disjoint multinodes $\underline{x}., \underline{y}., \underline{a}.$. This can be done by estimating the conditional mutual information (CMI) $H(\underline{x}. : \underline{y}.|\underline{a}.)$. `bnlearn` can calculate CMI and other metrics of $\underline{x}. \perp_P \underline{y}.|\underline{a}.$. All these metrics are very similar; they all measure how close $P(x.|y., a.)$ and $P(x.|a.)$ are.

The first constraint-based SL algorithm was the Inductive Causation (IC) algorithm proposed by Pearl and Verma in 1991. Incremental improvements have been proposed since then, such as the PC family of algorithms, Grow-Shrink and the Incremental Association Markov Blanket (IAMB) family of algorithms.

95.4 Pseudo-code for some bnet learning algorithms

Algorithm 4: Pseudo-code for Hill Climbing algorithm

```
Input  : Data D, Vertices V
Output: a bnet B = (G, T), where G = (V, E) is a DAG, where V are its
        vertices (nodes) and E are its edges (arrows). T are all its
        Transition Probability Matrices (TPMs) T = TPMs(G, D).
E ← ∅
T ← ∅
B ← (V, E, T)
maxscore ← −∞
// DE= all possible directed edges
DE = {x → y ∈ V × V : x ≠ y}
again ← True
while again do
    for all x → y ∈ DE do
        // add arrow
        E+ ← E ∪ {x → y}
        // delete arrow
        E− ← E − {x → y}
        // reverse arrow
        ER ← E− ∪ {y → x}
        for E' = E+, E−, ER do
            if E' ≠ E and G' = (V, E') is a legal DAG then
                T' ← TPMs(G', D)
                B' ← (G', T')
                newscore = BIC-score(B')
                if newscore > maxscore then
                    B ← B'
                    maxscore ← newscore
                else
                    again ← False
return B
```

Algorithm 5: Pseudo-code for PC-Stable algorithm

Input : Data D, Vertices (nodes) V, tolerance in CMI $\epsilon > 0$

Output: partially oriented acyclic graph $G = (V, E, UE)$, where V are the vertices (nodes), E are the oriented edges (arrows) and UE are the unoriented edges.

```
E ← ∅
// initialize UE to fully-connected undirected graph
UE ← {x − y ∈ V × V : x − y = y − x, x ≠ y}
// Shrink phase.  Deletes edges from E.
for λ = 0, 1, 2, ..., |V| − 2 do
    for all x − y ∈ UE do
        for all S = {a ∈ V : x − a ∈ UE, a ≠ x, y} ∋ |S| = λ do
            if H(x : y|S) < ε then
                /* If there were an arrow between x and y, then
                   conditioning on S would not be enough to interrupt
                   info transmission H(x : y|S) between x and y        */
                UE ← UE − {x − y}
                S(x − y) ← S
// Growth phase.  Adds v structures to E.
for all x, y, a such that x − a ∈ UE, a − y ∈ UE, x − y ∉ UE, a ∉ S(x − y) do
    /* If there were no collider at a, then there would be info
       transmission between x and y                                    */
    UE ← UE − {x − a, a − y}
    E ← E ∪ {x → a, y → a}
// Orienting edges.
again ← True
size ← |UE|
while again do
    for all x − y ∈ UE do
        if x → y ∈ E, y − z ∈ UE, x − z ∉ UE, ∄w ∋ w → y ∈ E then
            // to avoid introducing new v structure
            UE ← UE − {y − z}
            E ← E ∪ {y → z}
        if x → y ∈ E and there is directed path from x to y in E then
            // to avoid introducing cycles
            UE ← UE − {x − y}
            E ← E ∪ {x → y}
    newsize ← |UE|
    if size == newsize then
        again ← False
    else
        size ← newsize
return G = (V, E, UE)
```

Chapter 96

Support Vector Machines And Kernel Method

This chapter is based on Refs.[152]. [185] and [153].

The Support Vector Machines (SVM) method was first invented with a linear kernel, but was later generalized to arbitrary kernels. We will use the terms SVM method and Kernel Method indistinguishably.

The SVM method is a fairly general method for calculating, via supervised learning, a *binary* classifier. The SVM method finds a continuous surface that separates a space into two disjoint parts.

Let $\Sigma = [0, 1, 2, \ldots, nsam-1]$ be a list of individuals (samples) in a population. In this chapter, we will use the notation $A^\sigma = A[\sigma]$ and $\vec{A} = [A^\sigma : \sigma \in \Sigma]$ for a list (vector, 1-D array) indexed by Σ. We will refer to $DS = (\vec{x}, \vec{y})$ where $x^\sigma \in val(\underline{x})$, $y^\sigma \in \{-1, 1\}$, as a dataset. Let $x^\sigma = (x_0^\sigma, x_1^\sigma, \ldots, x_{nf-1}^\sigma) \in val(\underline{x}_0) \times val(\underline{x}_1) \times \ldots \times val(\underline{x}_{nf-1}) = val(\underline{x})$. When $x_j^\sigma \in \mathbb{R}$ for all j, we will take $x^\sigma \in \mathbb{R}^{nf}$ to be a column vector. x^σ is the feature vector for individual σ, and its components x_i^σ for $i = 0, 1, \ldots, nf-1$ are the features. $y^\sigma \in \{-1, 1\}$ is the binary class to which x^σ belongs.

Let $\widehat{y}(x^{\sigma_0}) \in \{-1, 1\}$ be an estimate of $y^{\sigma_0} \in \{-1, 1\}$. The **SVM classifier** is defined as

$$\widehat{y}(x^{\sigma_0}) = \text{sign}(Y(x^{\sigma_0})) \tag{96.1}$$

where[1]

$$Y(x^{\sigma_0}) = \sum_\sigma \alpha^\sigma y^\sigma K(x^\sigma, x^{\sigma_0}) \ . \tag{96.2}$$

The **binary weight coefficients** $\alpha^\sigma \in \{0, 1\}$ for all $\sigma \in \Sigma$ are found by training, via an algorithm to be described below.

The function $K : val(\underline{x}) \times val(\underline{x}) \to \mathbb{R}$ is called the **Kernel or Similarity function**. We assume that $K(x^\sigma, x^{\sigma_0})$ grows bigger when its two arguments x^σ and

[1] Define $\text{sign}(0) = 1$.

x^{σ_0} become more "similar". We also assume that $K(x^\sigma, x^{\sigma_0})$ is symmetric in its two arguments.

96.1 Learning Algorithm for SVM Classifier

Figure 96.1: Time-slice σ_0 of dynamical bnet for learning binary weights $\vec{\alpha}$ of SVM classifier.

Given a kernel function K and a dataset $(\vec{x}, \vec{y})$, the SVM classifier is fully specified except for its binary weights $\vec{\alpha}$. Those weights can be learned via the algorithm represented as a causal diagram in Fig.96.1. That figure shows two time-slices of a dynamical bnet. The TPMs, printed in blue, of bnet Fig.96.1, are as follows:

$$P(\widehat{y}|\vec{\alpha}, (\vec{x}, \vec{y}), x^{\sigma_0}) = \mathbb{1}(\quad \widehat{y} = \text{ given by Eq.(96.1).} \quad) \tag{96.3}$$

$$P(\mathcal{E}|\widehat{y}, y^{\sigma_0}) = \mathbb{1}(\quad \mathcal{E} = \mathbb{1}(\widehat{y} \neq y^{\sigma_0}) \quad) \tag{96.4}$$

The first (but not the second, third , etc.) $\vec{\alpha}$ node of Fig.96.1 is a root node. The TPM for that root node should set all components of $\vec{\alpha}$ to zero:

$$P(\vec{\alpha}) = \prod_\sigma \mathbb{1}(\quad \alpha^\sigma = 0 \quad) \,. \tag{96.5}$$

After that initialization,

$$P(\vec{\alpha'}|\vec{\alpha}, \mathcal{E}) = \mathbb{1}(\quad (\alpha')^{\sigma_0} = \alpha^{\sigma_0} + \mathcal{E} \quad) \prod_{\sigma \neq \sigma_0} \mathbb{1}(\quad (\alpha')^\sigma = \alpha^\sigma \quad) \tag{96.6}$$

Why this learning algorithm works.

Figure 96.2: Define the neighborhood of x^{σ_0} by $\mathcal{N}(x^{\sigma_0}) = \{x^\sigma : |K(x^\sigma, x^{\sigma_0})| < \epsilon\}$ for some $\epsilon > 0$.

$K(x^\sigma, x^{\sigma_0})$ sets to zero any contribution to $Y(x^{\sigma_0})$ from points x^σ outside the neighborhood $\mathcal{N}(x^{\sigma_0})$ of x^{σ_0}. (See Fig.96.2). If $\widehat{y}(x^{\sigma_0}) = y^{\sigma_0}$, keep $\alpha^{\sigma_0} = 0$ because the neighbors of x^{σ_0} are giving the correct $\widehat{y}(x^{\sigma_0})$ when they are polled and the majority wins. If, on the other hand, $\widehat{y}(x^{\sigma_0}) \neq y^{\sigma_0}$, then switch α^{σ_0} from 0 to 1, which means x^{σ_0} gets to vote by adding y^{σ_0} to $Y(x^{\sigma_0})$. So we start off with all $\alpha^\sigma = 0$ and we end with most of them still zero except for a select few. If we were to set all α^σ equal to one, we would get overfitting and a very jagged separation between the two classes. The fact that we end with only a select few α^σ equal to 1, and the rest equal to 0, helps make the demarcation between the two classes less jagged.

96.2 Linear (dot-product) Kernel

$$Y(x^\sigma) = w^T x^\sigma + b$$

Figure 96.3: Graph of line $Y(x^\sigma) = 0$ splits plane into regions with $Y < 0$, $Y = 0$ and $Y > 0$.

So far, we have discussed the SVM method for an arbitrary kernel. This section is devoted to the **Linear (a.k.a. dot-product) Kernel**. Said kernel is defined as

$$K(x^\sigma, x^{\sigma_0}) = (x^\sigma)^T x^{\sigma_0} \ . \tag{96.7}$$

For this kernel, Eq.(96.2) specializes to

$$Y(x^{\sigma_0}) = \sum_\sigma \alpha^\sigma y^\sigma K(x^\sigma, x^{\sigma_0}) + b \tag{96.8}$$

$$= w^T x^{\sigma_0} + b \tag{96.9}$$

where

$$w = \sum_\sigma \alpha^\sigma y^\sigma x^\sigma \ . \tag{96.10}$$

Figure 96.4: We refer to the gray shaded region with $-1 < Y < 1$, where $Y = w^T x + b$, as the DMZ.

We started this chapter by pulling the SVM classifier out of a hat. We did give reasons why it works, but we did not derive it from a more general minimization principle. Such a derivation is possible, at least in the linear kernel case, and we give it next.

Consider the following 3 straight lines:

$$w^T x^\sigma + b = +A \tag{96.11}$$

$$w^T x^\sigma + b = 0 \tag{96.12}$$

$$w^T x^\sigma + b = -A \tag{96.13}$$

where $w, x^\sigma \in \mathbb{R}^{nf}$, and $b, A \in \mathbb{R}$. We can re-scale the vector w and scalar b so as to get rid of the A. (i.e., replace $w \to wA$ and $b \to bA$ and divide common factor A out of equations). This rescaling does not affect the graphs (i.e., x loci) of these 3 lines. Now we have:

$$w^T x^\sigma + b = +1 \tag{96.14}$$
$$w^T x^\sigma + b = 0 \tag{96.15}$$
$$w^T x^\sigma + b = -1 \tag{96.16}$$

If Y stands for

$$Y = w^T x^\sigma + b \ , \tag{96.17}$$

then we define the **DMZ (demilitarized zone)** to be the region

$$DMZ = \{x^\sigma : |Y(x^\sigma)| < 1\} \ . \tag{96.18}$$

The lines $Y = \pm 1$ will be called the **borders (a.k.a. margins)** of the DMZ, and line $Y = 0$ will be called the **line of demarcation** of the DMZ. The DMZ is illustrated in Fig.96.4.

Let D_{DMZ} be the **DMZ width** (i.e., the distance from one border of the DMZ to the other.) Position vectors pointing from the origin to either of the two DMZ borders are called **support vectors**. Suppose $X^+, X^- \in \mathbb{R}^{nf}$ are two support vectors on opposite DMZ borders with $|X^+ - X^-| = D_{DMZ}$. Then

$$w^T X^+ + b = 1 \tag{96.19}$$
$$w^T X^- + b = -1 \tag{96.20}$$

so

$$D_{DMZ} = \frac{2}{|w|} \ . \tag{96.21}$$

For any $a \in \mathbb{R}$, let the **positive** a_+ **and negative** a_- **parts of** a be given by

$$a = \underbrace{a_+}_{a\mathbb{1}(a>0)} + \underbrace{a_-}_{a\mathbb{1}(a\leq 0)} \ . \tag{96.22}$$

When $Y = \pm 1$, an error in $Y(x^\sigma)$ occurs iff $y^\sigma Y(x^\sigma) = -1$. But how should we define errors when Y is a real number? Define the **Cost of erring for sample** σ to be

$$CE^\sigma(x^\sigma, y^\sigma) \ = \ (1 - y^\sigma Y(x^\sigma))_+ \ . \tag{96.23}$$

CE^σ is shown in Fig.96.5. As you can see, there is a penalty for living on the incorrect side, and even a penalty for living on the correct side but too close to the DMZ.

Figure 96.5: Plot of CE^σ versus Y.

Note that the line of demarcation should have the lowest CE^σ for all possible w, b. So to find that line, we want to minimize CE^σ with respect to w, b. But note that $CE^\sigma \geq 0$, and it can be zero for an appropriately chosen w. So we need to add another cost in order to get a non-zero total cost. Define the **DMZ cost** as

$$CZ = \frac{1}{2}|w|^2 = \frac{2}{D_{DMZ}^2} \tag{96.24}$$

Note that $CZ \to \infty$ as $D_{DMZ} \to 0$, so CZ penalizes DMZ's that are too narrow.

Now define a Lagrangian $\mathcal{L}$ to be the sum of these 2 contributions.

$$\mathcal{L} = CZ + \sum_\sigma CE^\sigma \tag{96.25}$$

$$= \frac{1}{2}|w|^2 + \sum_\sigma (1 - y^\sigma Y(x^\sigma))_+ \tag{96.26}$$

This particular choice of CZ is not unique, but it isn't totally arbitrary either. We want it to be independent of the sample σ, and to depend on a geometrical aspect of the DMZ, like its width $D_{DMZ} = 2/|w|$. Note that CE^σ behaves, when $|w| \to \infty$, linearly in $|w|$. We are going to differentiate $\mathcal{L}$ with respect to $|w|$ to find an optimum. But straight lines have no optima, so we need CZ to behave, when $|w| \to \infty$, as $|w|^p$ for some integer $p > 1$.

Setting the variation $\delta\mathcal{L}$ to zero, we get

$$0 = \delta\mathcal{L} = \delta w_i \left\{ w_i - \sum_\sigma \mathbb{1}(y^\sigma Y(x^\sigma) < 1) y^\sigma x_i^\sigma \right\} \tag{96.27}$$

so

$$w = \sum_\sigma \underbrace{\mathbb{1}(y^\sigma Y(x^\sigma) < 1)}_{\alpha^\sigma} y^\sigma x^\sigma \ . \tag{96.28}$$

96.3 Alternatives to Linear Kernel

Sometimes it is convenient to replace the dot-product kernel given above by a different kernel. Other popular kernels are:

- **Radial Basis Function (RBF) Kernel.** In this case, K is a radial function; i.e., a function that depends only on the magnitude (radius, Euclidean distance, L^2 norm) of a vector. An example of an RBF kernel is the **Gaussian Kernel**

$$K(x^\sigma, x^{\sigma_0}) = e^{-\gamma|x^\sigma - x^{\sigma_0}|^2} \tag{96.29}$$

  for some free parameter $\gamma > 0$.

- **Inhomogeneous Polynomial Kernel**

$$K(x^\sigma, x^{\sigma_0}) = [(x^\sigma)^T x^{\sigma_0} + 1]^d \tag{96.30}$$

  for some positive integer d.

- **Homogeneous Polynomial Kernel**

$$K(x^\sigma, x^{\sigma_0}) = [(x^\sigma)^T x^{\sigma_0}]^d \tag{96.31}$$

  for some positive integer d.

- **"Kernel trick" Kernel.** Consider a map $\Phi : \mathbb{R}^{nf} \to \mathbb{R}^N$. Usually $N > nf$. Φ can be a rectangular matrix $A \in \mathbb{R}^{N \times nf}$ so that $\Phi(x^\sigma) = Ax^\sigma \in \mathbb{R}^N$, but the most advantageous use of this kernel arises when $\Phi(\cdot)$ is a non-linear map. Let

$$K(x^\sigma, x^{\sigma_0}) = [\Phi(x^\sigma)]^T \Phi(x^{\sigma_0}) \tag{96.32}$$

  Although the constant surfaces of this kernel are hyperplanes in $\mathbb{R}^N$, its constant surfaces in $\mathbb{R}^{nf}$ can be curved and even closed compact surfaces (e.g. spheres).

96.4 Random Forest and Kernel Method

Chapter 97

Survival Analysis

This chapter is based on Refs.[199] and [186].

Survival Analysis (SA) is used for curve-fitting, to fit a curve $S(t)$ to data indicating the number of patients surviving after receiving a treatment for time t. Alternatively, it can be used with data indicating the number of devices that haven't failed after running for time t. SA can also be used to compare two such $S(t)$ curves—for example, one for treated patients, and another for untreated patients. Hence, SA can be used to analyze the data of an RCT.

Let

$\sigma \in \Sigma$, individual (e.g., patient) in population Σ.

$N = |\Sigma|$, size of Σ, nsam, number of samples

Note: A subset of Σ, (a.k.a, sub-population or stratum) is often called a **cohort** in epidemiology.

$\tau^\sigma \geq 0$, time to **final event** such as death of an organism, or failure of a device, duration of stay, follow-up time period, time period, lifetime.

$\tau^\sigma = b^\sigma - a^\sigma$ for some absolute times a^σ, b^σ satisfying $a^\sigma < b^\sigma$

$U^\sigma =$ **censoring upper bound**

$L^\sigma =$ **censoring lower bound**

$B^\sigma = \min(b^\sigma, U^\sigma)$, **right censoring**

$A^\sigma = \max(a^\sigma, L^\sigma)$, **left censoring**

$T^\sigma = B^\sigma - A^\sigma$, where $A^\sigma < B^\sigma$

$d^\sigma = \mathbb{1}(b^\sigma < U^\sigma)$, equals 1 if death (i.e., no censoring), equals 0 if no death (i.e., censoring)

$e^\sigma = \mathbb{1}(a^\sigma > L^\sigma)$, equals 1 if death (i.e., no censoring), equals 0 if no death (i.e., censoring)

Will only use right censoring in this chapter.

For $t \geq 0$, define

- **Survival function** $S(t)$ and **Cumulative hazard function** $\Lambda(t)$

$$S(t) = P(\underline{\tau} \geq t) = e^{-\Lambda(t)} \tag{97.1}$$

Note that we define in this chapter $S(t) = P(\underline{\tau} \geq t)$ (those present among

survivors (PAS)) instead of $P(\underline{\tau} > t)$ (not PAS), like other authors do. For continuous functions, these 2 definitions of $S(t)$ are the same.

Note that, since $S(t)$ is a probability and $t, \tau \in [0, \infty]$, $\Lambda(t)$ must be a monotonically increasing function with $\Lambda(0) = 0$ and $\Lambda(\infty) = \infty$.

- **hazard function (a.k.a. instantaneous failure rate)** $\lambda(t)$

$$\Lambda(t) = \int_0^t du\ \lambda(u) \tag{97.2}$$

$$\lambda(t) = \Lambda'(t) \tag{97.3}$$

Note that $\lambda(t) \geq 0$ and its integral over $[0, \infty]$ must be ∞. $\lambda(t)$ is in fact a conditional probability as we show next.

$$\begin{aligned}
\lambda(t) &= \frac{P(\underline{\tau} = t)}{S(t)} \quad \text{(shown below)} && (97.4)\\
&= \frac{P(\underline{\tau} = t)}{P(\underline{\tau} \geq t)} && (97.5)\\
&= \frac{P(\underline{\tau} = t, \underline{\tau} \geq t)}{P(\underline{\tau} \geq t)} \quad \text{(because } P(A \wedge B) = P(A) \text{ if } A \text{ implies } B\text{)} && (97.6)\\
&= P(\underline{\tau} = t \mid \underline{\tau} \geq t) && (97.7)
\end{aligned}$$

- $\underline{\tau}$ **density function** $P_{\underline{\tau}}(t)$

$$P_{\underline{\tau}}(t) = -S'(t) = \lambda(t)e^{-\Lambda(t)} = \lambda(t)S(t) \tag{97.8}$$

- $\underline{\tau}$ **cumulative distribution function** $\Phi_{\underline{\tau}}(t)$

$$\Phi_{\underline{\tau}}(t) = P(\underline{\tau} < t) = 1 - S(t) \tag{97.9}$$

$$P_{\underline{\tau}}(t) = \Phi'_{\underline{\tau}}(t) \tag{97.10}$$

- **mean survival time** μ

$$\mu = \int_0^\infty dt\ tP_{\underline{\tau}}(t) \tag{97.11}$$

- **median survival time** τ_{med}

$$S(\tau_{med}) = 0.5 \tag{97.12}$$

97.1 $S(t)$ estimates

97.1.1 No-censoring estimate of $S(t)$

Let

$$r^\sigma(t) = \mathbb{1}(\tau^\sigma \geq t) \tag{97.13}$$

$r^\sigma(t)$ equals 1 iff individual σ at risk (i.e., still alive and not censored, not out) at time t

$$\widehat{S}(t) = \frac{1}{N}\sum_\sigma r^\sigma(t) \tag{97.14}$$

$\{\underline{r}^\sigma(t) : \sigma \in \Sigma\}$ are i.i.d.

$\underline{r}^\sigma(t) = \underline{x}$ is a Bernoulli random variable $Bern(p = S(t))$(i.e., simple coin flip with $P(\underline{x} = 1) = p$, $P(\underline{x} = 0) = q$, mean p and variance pq, where $p = S(t)$)

$$\widehat{S}(t) \to \mathcal{N}\left(\mu = S(t), \sigma^2 = \frac{S(t)[1 - S(t)]}{N}\right) \text{ as } N \to \infty \tag{97.15}$$

(convergence in probability) A sanity check for Eq.(97.15) is to note that

$$\begin{aligned} \left\langle \widehat{S}(t) \right\rangle &= \frac{1}{N}\sum_\sigma \underbrace{\langle \underline{r}^\sigma(t) \rangle}_{S(t)} && (97.16) \\ &= S(t) && (97.17) \end{aligned}$$

$$\begin{aligned} \left\langle \widehat{S}(t), \widehat{S}(t') \right\rangle &= \frac{1}{N^2}\sum_\sigma\sum_{\sigma'} \left\langle \underline{r}^\sigma(t), \underline{r}^{\sigma'}(t') \right\rangle && (97.18) \\ &= \frac{1}{N^2}\sum_\sigma \langle \underline{r}^\sigma(t), \underline{r}^\sigma(t) \rangle && (97.19) \\ &= \frac{S(t)[1 - S(t)]}{N} && (97.20) \end{aligned}$$

97.1.2 Kaplan-Meier estimate of $S(t)$

Let

$[\tau^j[_{j=1,2,\ldots,J}$, times at which there is a final event, all measured from the same time origin, and ordered so that $\tau^j < \tau^{j+1}$

n_D^j = number of individuals that die at time τ^j

n_C^j = number of individuals that are censored at time τ^j

$n_O^j = n_D^j + n_C^j$ number of individuals that drop-Out at time τ^j, either because they die or are censored

$n_R^j = \sum_{k \geq j} n_O^k$, number of individuals that are at risk at or after time τ^j(i.e., still alive and not censored, not out, surviving before time τ^j)

$d^\sigma \in \{0, 1\}$, it equals 1 iff individual σ dies at any time.
$d^\sigma(t) \in \{0, 1\}$, it equals 1 iff individual σ dies at time t.
$d^\sigma(\tau^j) \in \{0, 1\}$, it equals 1 iff individual σ dies at time τ^j.
$d^\sigma(\tau^j) \to \underline{d}(\tau^j)$ since i.i.d.

$c^\sigma = 1 - d^\sigma$, it equals 1 iff individual σ is censored at any time.
$c^\sigma(t) = 1 - d^\sigma(t)$, it equals 1 iff individual σ is censored at time t.
$c^\sigma(\tau^j) = 1 - d^\sigma(\tau^j)$, it equals 1 iff individual σ is censored at time τ^j.
$c^\sigma(\tau^j) \to \underline{c}(\tau^j)$ since i.i.d.

$o^\sigma(t) = \mathbb{1}(\tau^\sigma = t)$, it equals 1 iff individual σ drops out at time t.
$o^\sigma(\tau^j) = \mathbb{1}(\tau^\sigma = \tau^j)$, it equals 1 iff individual σ is drops out at time τ^j.
$o^\sigma(\tau^j) \to \underline{o}(\tau^j)$ since i.i.d.

Note that[1]

$$\underbrace{\bigwedge_{k \leq j} \{o^\sigma(\tau^k) = 0\}}_{\sigma \text{ did not drop out in past or present } \tau^j} = \underbrace{\bigoplus_{k > j} \{o^\sigma(\tau^k) = 1\}}_{\sigma \text{ drops out in future}} \tag{97.21}$$

or, replacing o^σ by $\underline{o}$,

$$\bigwedge_{k \leq j} \{\underline{o}(\tau^k) = 0\} = \bigoplus_{k > j} \{\underline{o}(\tau^k) = 1\} \tag{97.22}$$

[1]notation: $\wedge_{i=1,2} a_i = (a_1 \wedge a_2) = (a_1 \text{ And } a_2)$,
$\vee_{i=1,2}\, a_i = (a_1 \vee a_2) = (a_1 \text{ Or } a_2)$,
$\oplus_{i=1,2}\, a_i = (a_1 \oplus a_2)$ =exclusive or= modulus 2 addition= binary addition with $(1 \oplus 1) = 0$.

Kaplan-Meier (KM) estimate of $S(t)$ (see[2])

$$\begin{aligned}
\widehat{S}(\tau^j) &= P\left(\bigoplus_{k>j}\{\underline{o}(\tau^k)=1\}\right) && (97.23)\\
&= P\left(\bigwedge_{k\le j}\{\underline{o}(\tau^k)=0\}\right) && (97.24)\\
&= \prod_{k\le j} P\left(\underline{o}(\tau^k)=0 \mid \bigwedge_{k'<k}\{\underline{o}(\tau^{k'})=0\}\right) \quad \text{(chain rule)} && (97.25)\\
&= \left\{\begin{array}{l} \prod_{k\le j} P\left(\underline{d}(\tau^k)=0 \mid \wedge_{k'<k}\{\underline{o}(\tau^{k'})=0\}\right) \\ + \underbrace{\prod_{k\le j} P\left(\underline{c}(\tau^k)=0 \mid \bigwedge_{k'<k}\{\underline{o}(\tau^{k'})=0\}\right)}_{=0\text{(see footnote)}} \end{array}\right. && (97.26)\\
&= \prod_{k\le j}\left[1-P\left(\underline{d}(\tau^k)=1 \mid \bigwedge_{k'<k}\{\underline{o}(\tau^{k'})=0\}\right)\right] && (97.27)\\
&= \prod_{k\le j}\left[1-\frac{n_D^k}{n_R^k}\right] && (97.28)
\end{aligned}$$

Figure 97.1: Bnet for KM estimate of $S(t)$ for $N=4$.

Fig.97.1 gives a bnet for the KM estimate of $S(t)$. The TPMs, printed in blue, for that bnet, must be as follows:

$$P(\underline{o}(\tau^k)=1 \mid \bigwedge_{k'<k}\{\underline{o}(\tau^{k'})=0\}) = \frac{n_O^k}{n_R^k} \tag{97.29}$$

[2]Even though $S(t)$ has been defined as $P(\underline{\tau}\ge t)$, a more precise definition in the presence of censoring is $P(\underline{\tau}\ge t \mid \text{patient dies rather than being censored})$

$$P(\underline{d}(\tau^k) = 1 \mid \underline{o}(\tau^k) = 1) = \frac{n_D^k}{n_O^k} \tag{97.30}$$

$$P(\underline{c}(\tau^k) = 1 \mid \underline{o}(\tau^k) = 1) = \frac{n_C^k}{n_O^k} \tag{97.31}$$

Intuition: Let

$$\widehat{\lambda}^k = \frac{n_D^k}{n_R^k} \tag{97.32}$$

If $\widehat{\lambda}^k << 1$, then, since $e^x \approx 1 + x$ for $|x| << 1$,

$$\begin{aligned} \widehat{S}(\tau^j) &\approx \prod_{k \le j} e^{-\widehat{\lambda}^k} && (97.33) \\ &= \exp[-\sum_{k \le j} \widehat{\lambda}^k] && (97.34) \\ &\approx \exp[-\int_0^{\tau_j} dt\ \widehat{\lambda}(t)] && (97.35) \end{aligned}$$

Note that:

- $\widehat{S}(t)$ only changes when there is a death.
- $\widehat{S}(t) = 1$ before the first death time
- $\widehat{S}(t)$ only goes to 0 if the last observation is a death, so $n_D^j / n_R^j = 1$.
- When there is no censoring, the KM estimate equals the no censoring estimate given earlier.

Greenwood's formula for variance of KM estimate of $S(t)$

$$\left\langle \widehat{S}(t), \widehat{S}(t) \right\rangle = [S(t)]^2 \sum_{k:\tau^k \le t} \frac{n_D^k}{n_C^k n_R^k} \tag{97.36}$$

Fig.97.2 and Table 97.1 give a numerical example of the KM estimate.

Fig.97.3 shows relevant parameters for steps τ^{j-1} and τ^j in a plot of a KM estimate.

6	6	6	6C	7	9C	10
10C	11C	13	16	17C	19C	20C
22	23	25C	32C	32C	34C	35C

Figure 97.2: Plot of KM estimate for $N = 21$ patients with τ^{σ} given in table below the plot. $\tau's$ with C are censored.

j	τ^j	n_R^j	n_D^j	n_C^i
1	6	21	3	1
2	7	17	1	0
3	9	16	0	1
4	10	15	1	1
5	11	13	0	1

Table 97.1: Parameters $j, \tau^j, n_R^j, n_D^j, n_C^j$ for first five lifetimes for Fig.97.2.

Figure 97.3: Steps τ^{j-1} and τ^j in a plot of a KM estimate. An **X** indicates a dead individual and ■ a censored one. Note that
$n_O^j = n_D^j + n_C^j$,
$n_R^j = n_R^{j-1} - n_O^{j-1}$ and
$n_R^j = \sum_{k \geq j} n_O^k$.

97.2 $\lambda(t)$ models

97.2.1 $\lambda(t)$ independent of covariates Z

Recall $t \geq 0$

- $\lambda(t)$ is independent of time

$$\lambda(t) = \lambda \tag{97.37}$$

where $\lambda \geq 0$.

$$\Lambda(t) = \lambda t \tag{97.38}$$

$$P_{\underline{\tau}}(t) = \lambda e^{-\lambda t} \quad \text{(Exponential distribution)} \tag{97.39}$$

- $\lambda(t)$ is proportional to power of t

$$\lambda(t) = \kappa \lambda t^{\kappa - 1} \tag{97.40}$$

where $\lambda, \kappa \geq 0$.

$$\Lambda(t) = \lambda t^{\kappa} \tag{97.41}$$

$$P_{\underline{\tau}}(t) = \kappa\lambda t^{\kappa-1} e^{-\lambda t^{\kappa}} \quad \text{(Weibull distribution)} \tag{97.42}$$

- $\lambda(t) = a + bt$ for $a, b \geq 0$
- $\lambda(t)$ is piecewise constant for $t \in [0, \infty]$
- etc.

Maximum Likelihood Estimate (MLE) of λ for $\lambda(t) = \lambda$, allowing censoring: Likelihood function is

$$\begin{aligned} L(\lambda) &= \prod_j [P_{\underline{\tau}}(\tau^j)]^{d(\tau^j)} [S(\tau^j)]^{c(\tau^j)} && (97.43) \\ &= \prod_j [\lambda e^{-\lambda\tau^j}]^{d(\tau^j)} [e^{-\lambda\tau^j}]^{1-d(\tau^j)} && (97.44) \\ &= \prod_j \lambda^{d(\tau^j)} e^{-\lambda\tau^j} && (97.45) \end{aligned}$$

Hence

$$\ln L(\lambda) = (\ln \lambda) \underbrace{\sum_j d(\tau^j)}_{D} - \lambda \underbrace{\sum_j \tau^j}_{T} \tag{97.46}$$

Setting $\partial_\lambda \ln L = 0$, we get

$$\widehat{\lambda} = \frac{D}{T} \tag{97.47}$$

97.2.2 $\lambda(t)$ dependent on covariates Z

Suppose $\beta, Z \in \mathbb{R}^{nind}$ are column vectors, where *nind* is number of independent variables (covariates) in a regression.

Cox Proportional Hazards (PH) model for $\lambda(t)$

$$\lambda(t) = \lambda(t|Z) = \lambda_0(t) e^{\beta^T Z} \tag{97.48}$$

where $\lambda_0(t) \geq 0$. $\lambda_0(t)$ is called the **baseline hazard function**. This $\lambda(t)$ model is called a PH model because $\lambda(t|Z_1)/\lambda(t|Z_2) = \exp[\beta^T(Z_1 - Z_2)]$ is independent of time. In a **Cox hazards model with time-dependent covariates (TC)**, one assumes that the covariates $Z(t)$ depend on time.

Note that since exponentials are always positive, the components of β and Z can range over all real values. If we had chosen $\lambda(t) = \lambda_0(t)\beta^T Z$ instead, then we would have to constrain $\beta^T Z \geq 0$.

If we define

$$\Lambda_0(t) = \int_0^t du\ \lambda_0(u) \tag{97.49}$$

then

$$P_{\underline{\tau}}(t) = \lambda_0(t)e^{\beta^T Z} \exp[-\Lambda_0(t)e^{\beta^T Z}] \tag{97.50}$$

$P_{\underline{\tau}}(t)$ is called the **Cox PH distribution**. It's a **semi-parametric distribution** because it depends on both, a prior parameter β and a prior function $\lambda_0(t)$ (a function is like an infinite dimensional parameter). A **parametric distribution** depends only on a prior parameter and a **non-parametric distribution** depends only on a prior function.

Recall that we defined earlier

n_D^j = number of individuals that die at time τ^j

n_C^j = number of individuals that are censored at time τ^j

To define the Cox partial likelihood function, we will assume that $n_D^j + n_C^j = 1$, i.e., each time τ^j has either a single death, or a single censorship, but not both. Since every individual eventually dies or is censored (but, we will assume, not both), there is a 1-1 onto map $j(\sigma)$ mapping Σ to the set of indices j of τ^j. So we can label the population individuals σ by their index j, or vice versa.

Let

$$L^j(\beta) = \frac{e^{\beta^T Z^j}}{\sum_{k \geq j} e^{\beta^T Z^k}} \tag{97.51}$$

Then define the **Cox partial likelihood function** by

$$L(\beta) = \prod_j [L^j(\beta)]^{d(\tau^j)} \tag{97.52}$$

Cox's approximate method for finding the best fit for β is to set $\frac{\partial \ln L(\beta)}{\partial \beta_a} = 0$ for all a. This does not determine the baseline hazard function, however.

Recall that

$$\lambda(\tau^j | Z^k) = \lambda_0(\tau^j)e^{\beta^T Z^k} = P(\underline{\tau}^k = \tau^j \mid \underline{\tau}^k \geq \tau^j) \tag{97.53}$$

Therefore

$$L^j(\beta) = \frac{\lambda_0(\tau^j)e^{\beta^T Z^j}}{\sum_{k\geq j}\lambda_0(\tau^j)e^{\beta^T Z^k}} \tag{97.54}$$

$$= \frac{\lambda(\tau^j|Z^j)}{\sum_{k\geq j}\lambda(\tau^j|Z^k)} \tag{97.55}$$

Next, we try to justify Cox's partial likelihood function. We will give two arguments.

1. Bayesian argument

   Assume that Z^k is a random variable with a non-informative prior $P(Z^k) = \mathcal{N}(!k)$. Then

   $$P(Z^k|\tau^j) = f(\tau^j)\underbrace{P(\tau^j|Z^k)}_{\lambda(\tau^j|Z^k)} \tag{97.56}$$

   for some function $f : \mathbb{R} \to \mathbb{R}$. Hence

   $$L^j(\beta) = \frac{P(Z^j \mid \tau^j)}{\sum_{k\geq j} P(Z^k \mid \tau^j)} \tag{97.57}$$

   $$= \frac{P(Z^j \mid \tau^j)}{P(\bigvee_{k\geq j}\{\underline{Z}^k = Z^k\} \mid \tau^j)} \quad \text{(because the } Z^k \text{ are independent)} \tag{97.58}$$

   $$= \frac{P(Z^j, \bigvee_{k\geq j}\{\underline{Z}^k = Z^k\}, \tau^j)}{P(\bigvee_{k\geq j}\{\underline{Z}^k = Z^k\}, \tau^j)} \quad \text{(because } P(A\wedge B) = P(A) \text{ if } A \implies B\text{)} \tag{97.59}$$

   $$= P(Z^j \mid \bigvee_{k\geq j}\{\underline{Z}^k = Z^k\}, \tau^j) \tag{97.60}$$

   Note that $L^j(\beta)$ depends on $\{Z^k : k \geq j\}$ because at time j, the past Z^k are fixed already, so the only ones we are allowed to optimize (remember, we are acting as Bayesians here, so we can optimize parameters) are the present and future ones.

2. Maximum Likelihood argument

$$L^j(\beta) = \begin{cases} P(\underline{\tau} \geq \tau^j) = S(\tau^j) & \text{if } d(\tau^j) = 0 \text{ (i.e., } c(\tau^j) = 1) \\ P(\underline{\tau} = \tau^j) = \lambda(\tau^j \mid Z^j) S(\tau^j) & \text{if } d(\tau^j) = 1 \end{cases}$$

$$= \lambda(\tau^j \mid Z^j)^{d(\tau^j)} S(\tau^j) \tag{97.62}$$

$$= \underbrace{\left[\frac{\lambda(\tau^j \mid Z^j)}{\sum_{k \geq j} \lambda(\tau^k \mid Z^j)}\right]^{d(\tau^j)}}_{L_1} \underbrace{\left[\sum_{k \geq j} \lambda(\tau^k \mid Z^j)\right]^{d(\tau^j)} S(\tau^j)}_{L_2} \tag{97.63}$$

Cox argued that L_2 varies very slowly with β.

97.3 $S_0(t)$ estimates

$$S_Z(t) = e^{-\Lambda(t)} \tag{97.64}$$

$$= e^{-\Lambda_0(t) \exp(\beta^T Z)} \tag{97.65}$$

$$= S_{Z=0}(t)^{\exp(\beta^T Z)} \tag{97.66}$$

Claim 173 *(Breslow $S_0(t)$ estimate)*

$$\widehat{S}_0(t) = e^{-\widehat{\Lambda}_0(t)} \tag{97.67}$$

where

$$\widehat{\Lambda}_0(t) = \sum_{j:\tau^j < t} \left[\frac{n_D^j}{\sum_{k \geq j} e^{\beta^T Z^k}}\right] \tag{97.68}$$

proof:

$$\frac{n_D^j}{\Delta t} \approx \sum_{k \geq j} P(\underline{\tau}^k = \tau^j \mid \underline{\tau}^k \geq \tau^j) \tag{97.69}$$

$$= \sum_{k \geq j} \lambda(\tau^j \mid Z^k) \tag{97.70}$$

$$= \lambda_0(\tau^j) \sum_{k \geq j} e^{\beta^T Z^k} \tag{97.71}$$

Hence

$$\lambda_0(\tau^j) \Delta t = \frac{n_D^j}{\sum_{k \geq j} e^{\beta^T Z^k}} \tag{97.72}$$

If we now apply $\sum_{j:\tau^j<t}$ to both sides of the last equation, we get Eq.(97.68).
QED

Note that

$$\widehat{S}_0(t) = \prod_{j:\tau^j<t} \exp\left[\frac{-n_D^j}{\sum_{k\geq j} e^{\beta^T Z^k}}\right] \tag{97.73}$$

$$\approx \prod_{j:\tau^j<t} \left[1 - \underbrace{\frac{n_D^j}{\sum_{k\geq j} e^{\beta^T Z^k}}}_{\widehat{\lambda}^j}\right] \text{(because } e^x \approx 1 + x \text{ for } |x| << 1) \tag{97.74}$$

Chapter 98

Synthetic Controls

This chapter is based on Refs.[18] and [15].

This chapter assumes that the reader has read Chapter 18 on the Difference-in-Differences (DID) method.

The Synthetic Controls (SC) method is a simple enhancement of the DID method. SC enhances DID in two simple yet powerful ways:

1. **Better time resolution.** DID considers just 2 time-snapshots (i.e., a time-series with only 2 times) whereas SC considers arbitrarily many time-snapshots (i.e., a time-series with more than 2 times).

2. **Weighted average of controls.** DID divides the population of individuals into just 2 kinds: the treated and the untreated (a.k.a. controls). SC divides the total population into treated and controls just like DID does, but it goes further and divides the control population into multiple subpopulations, and calculates a weighted average, called a "synthetic control", of those subpopulations. The weights of the synthetic control are chosen so that it mimics as closely as possible the behavior of the treated population for all times measured before the treatment was applied.

Let us describe these two enhancements more precisely.

- **timing**: Let t_k for $k = 0, 1, \ldots, npre - 1$ be the pre-treatment times at which a measurement occurs. Let t_k for $k = npre, npre + 1, \ldots, nt - 1$ be the post-treatment times at which a measurement occurs. Note that $npre + npost = nt$. Note that $t_* = t_{npre+1}$ is the first measurement time after the treatment is applied, t_0 is the first measurement time, and $t_{fin} = t_{nt-1}$ is the last one.

- **subpopulations**: Let $S_1 = \{\sigma_1\}$ be the set of treated units (just one). Let $S_0 = \{\sigma : \sigma \neq \sigma_1\}$ be the set of untreated units (i.e., controls). Let $nsam =$ number of all units σ, $n_1 = |S_1| = 1$, and $n_0 = |S_0| = nsam - 1$.

- **weights**:

Figure 98.1: Pictorial representation of the Synthetic Controls (SC) method. The outcome y of the synthetic control unit is colored red and that of the treated unit is colored blue. They roughly agree for $t < t_*$.

We want to define a time-independent weight w^σ for each unit σ in such a way that the output y_t^σ for the synthetic control unit behaves like the output for the treated unit σ_1 for $t < t_*$.

Let

$$w^{\sigma_1} = 0 \tag{98.1}$$

and

$$w^{n_0} = \{w^\sigma\}_{\sigma \neq \sigma_1} \ . \tag{98.2}$$

Define a cost function $\mathcal{C}$:

$$\mathcal{C}(w^{n_0}) = \sum_{t<t_*} \left(y_t^{\sigma_1} - \sum_{\sigma \neq \sigma_1} w^\sigma y_t^\sigma \right)^2 \tag{98.3}$$

Then calculate w^{n_0} by minimizing the cost function, subject to the constraint that w^{n_0} be a probability distribution:

$$w^{n_0} = \underset{W^{n_0}}{\operatorname{argmin}} \left\{ \mathcal{C}(W^{n_0}) : W^\sigma \geq 0, \sum_{\sigma \neq \sigma_1} W^\sigma = 1 \right\} \ . \tag{98.4}$$

Now that we have defined a weight w^σ for every unit σ, we can define for $c \in \{0, 1\}$,

$$y_t^{\sigma_1}(c) = \begin{cases} y_t^{\sigma_1} & \text{if } c = 1 \\ \sum_{\sigma \neq \sigma_1} w^\sigma y_t^\sigma & \text{if } c = 0 \end{cases} \tag{98.5}$$

$$\mathcal{Y}_c(t) = E_\sigma[y_t^\sigma(c)] \tag{98.6}$$

and

$$\delta_t = \mathcal{Y}_1(t) - \mathcal{Y}_0(t) \tag{98.7}$$

δ_t is illustrated in Fig.98.1. It approximates $ATE(t)$.

98.1 PO analysis

In this section, we show how to analyze the SC method using the formalism of PO theory.

Figure 98.2: $t \in \{t_0, t_1, \ldots, t_{fin}\}$. Bnet $G_{t,+}$ is obtained by adding two new nodes $\underline{y}_t(0)$ and $\underline{y}_t(1)$ to bnet G_t.

As usual for PO theory, we will consider expected values of y_t^σ:

$$E_{\sigma|d,x}[y_t^\sigma(c)] = E_{\underline{y}_t(c)|d,x}[\underline{y}_t(c)] = \mathcal{Y}_{c|d,x}(t) \tag{98.8}$$

To calculate these expected values, we need a "model" with probability distributions. In this case, the needed model and probability distributions are provided by the bnets depicted in Fig.98.2. The TPMs, printed in blue, for the bnet $G_{t,+}$ in Fig.98.2, are as follows. Note that the TPMs for the bnet $G_{t,+}$ are defined in terms of the TPMs for the bnet G_t.

$$P(x) = P_{\underline{x}}(x) \tag{98.9}$$

$$P(d|x) = P_{\underline{d}|\underline{x}}(d|x) \tag{98.10}$$

$$P(y_t|y_t(0), y_t(1), d) = \mathbb{1}(y_t = y_t(d)) \tag{98.11}$$

$$P(y_t(c)|x) = P(y_t(c)|d, x) = \text{given} \tag{98.12}$$

Figure 98.3: Four different time-dependent expected values $\mathcal{Y}_{c|d}(t)$ of y_t^σ for bnet $G_{t,+}$ The $2 * nt$ magenta stars represents the $2 * nt$ SC measurements.

Fig.98.3 depicts the four functions $\mathcal{Y}_{c|d}(t)$ for t in the interval $[t_0, t_{fin}]$ and for $c, d \in \{0, 1\}$. The $\mathcal{Y}$ coordinates of the $2 * nt$ magenta stars in Fig.98.3 can be calculated using bnet G_t. Note that in Fig.98.3, we display a large gap between the curves $\mathcal{Y}_{0|d}(t)$ for $d \in \{0, 1\}$. In reality, $P(y_t(0)|d)$ has been constructed so as to make that gap as small as possible. Thus, to a good(?) approximation,

$$\delta_t \approx ATE_t \tag{98.13}$$

Unlike in the DID method, in the SC method, to a good(?) approximation, we don't have to worry about parallel trends.

Chapter 99

Table 2 Fallacy

The **Table 2 Fallacy** (T2F) is so named because it is common in epidemiology papers to present a dataset in Table 1, and a Linear Regression (LR) analysis of that dataset in Table 2. Thus, a T2F is an error in the interpretation of LR results.

In LR, we define 2 types of variables: The dependent variables x_i and the independent one y. So in LR, the set of dependent variables is not divided into finer classes. However, in Causal Inference, dependent variables can be of various kinds, such as confounders, mediators, etc.

The covariance matrices in Eqs.(99.1, 99.2, 99.3) were obtained using the software SCuMpy (see Ref.[95])

Suppose we do a LR of the form $\underline{Y} \sim \underline{X} + \underline{Z}$ for the DAG in Fig.99.1 wherein $\underline{Z}$ is a confounder. Conditioning on $\underline{Z}$ (i.e., holding $\underline{Z}$ fixed) corresponds to setting $\sigma^2_{\epsilon_{\underline{Z}}} = \langle \underline{Z}, \underline{Z} \rangle = 0$ in Eq.(99.1). Likewise, conditioning on $\underline{X}$ corresponds to setting $\sigma^2_{\epsilon_{\underline{X}}} = \langle \underline{X}, \underline{X} \rangle = 0$. Note that

- The coefficient $\alpha_{\underline{Y}|\underline{X}}$ of $\underline{X}$ when we condition on $\underline{Z}$, equals the **full effect** (a.k.a. total effect) of $\underline{X}$ on $\underline{Y}$.
- The coefficient $\alpha_{\underline{Y}|\underline{Z}}$ of $\underline{Z}$ when we condition on $\underline{X}$, is NOT equal to the full effect $\left(\alpha_{\underline{X}|\underline{Z}}\alpha_{\underline{Y}|\underline{X}} + \alpha_{\underline{Y}|\underline{Z}}\right)$ of $\underline{Z}$ on $\underline{Y}$; rather its a **partial effect** (a.k.a. direct effect) of $\underline{Z}$ on $\underline{Y}$.

T2F is the false assumption that both coefficients in the LR given by $\underline{Y} \sim \underline{X} + \underline{Z}$ are full effects.

Suppose we do a LR of the form $\underline{Y} \sim \underline{X} + \underline{M}$ for the DAG in Fig.99.2 wherein $\underline{M}$ is a mediator. Note that in this case $\alpha_{\underline{Y}|\underline{X}}$ is a partial effect and $\alpha_{\underline{Y}|\underline{M}}$ is a full effect.

Finally, suppose we do a LR of the form $\underline{Y} \sim \underline{X} + \underline{M} + \underline{Z}$ for the DAG in Fig.99.3 wherein $\underline{M}$ is a mediator and $\underline{Z}$ is a confounder. Note that in this case $\alpha_{\underline{Y}|\underline{Z}}$ and $\alpha_{\underline{Y}|\underline{X}}$ are partial effects whereas $\alpha_{\underline{Y}|\underline{M}}$ is a full effect (if we condition on both $\underline{X}$ and $\underline{Z}$).

$$\begin{tikzcd}
 & \underline{Z} \arrow[dl] \arrow[dr] & \\
\underline{X} \arrow[rr] & & \underline{Y}
\end{tikzcd}$$

Figure 99.1: $\underline{X} \to \underline{Y}$ with confounder $\underline{Z}$.

$$\begin{aligned}
\langle \underline{Z}, \underline{Y} \rangle &= \sigma^2_{\underline{\epsilon}_{\underline{Z}}} \left(\alpha_{\underline{X}|\underline{Z}} \alpha_{\underline{Y}|\underline{X}} + \alpha_{\underline{Y}|\underline{Z}} \right) \\
\langle \underline{X}, \underline{Y} \rangle &= \alpha_{\underline{X}|\underline{Z}} \sigma^2_{\underline{\epsilon}_{\underline{Z}}} \left(\alpha_{\underline{X}|\underline{Z}} \alpha_{\underline{Y}|\underline{X}} + \alpha_{\underline{Y}|\underline{Z}} \right) + \alpha_{\underline{Y}|\underline{X}} \sigma^2_{\underline{\epsilon}_{\underline{X}}}
\end{aligned} \tag{99.1}$$

Eq.(99.1) gives some covariance matrices for Fig.99.1.

$$\begin{tikzcd}
\underline{X} \arrow[rr] \arrow[dr] & & \underline{Y} \\
 & \underline{M} \arrow[ur] &
\end{tikzcd}$$

Figure 99.2: $\underline{X} \to \underline{Y}$ with mediator $\underline{M}$.

$$\begin{aligned}
\langle \underline{X}, \underline{Y} \rangle &= \sigma^2_{\underline{\epsilon}_{\underline{X}}} \left(\alpha_{\underline{M}|\underline{X}} \alpha_{\underline{Y}|\underline{M}} + \alpha_{\underline{Y}|\underline{X}} \right) \\
\langle \underline{M}, \underline{Y} \rangle &= \alpha_{\underline{M}|\underline{X}} \sigma^2_{\underline{\epsilon}_{\underline{X}}} \left(\alpha_{\underline{M}|\underline{X}} \alpha_{\underline{Y}|\underline{M}} + \alpha_{\underline{Y}|\underline{X}} \right) + \alpha_{\underline{Y}|\underline{M}} \sigma^2_{\underline{\epsilon}_{\underline{M}}}
\end{aligned} \tag{99.2}$$

Eq.(99.2) gives some covariance matrices for Fig.99.2.

$$\begin{tikzcd}
 & \underline{Z} \arrow[dl] \arrow[dr] & \\
\underline{X} \arrow[rr] \arrow[dr] & & \underline{Y} \\
 & \underline{M} \arrow[ur] &
\end{tikzcd}$$

Figure 99.3: $\underline{X} \to \underline{Y}$ with confounder $\underline{Z}$ and mediator $\underline{M}$.

$$
\begin{aligned}
\langle \underline{Z}, \underline{Y} \rangle &= \sigma^2_{\underline{\epsilon}_{\underline{Z}}} \left(\alpha_{\underline{M}|\underline{X}} \alpha_{\underline{X}|\underline{Z}} \alpha_{\underline{Y}|\underline{M}} + \alpha_{\underline{X}|\underline{Z}} \alpha_{\underline{Y}|\underline{X}} + \alpha_{\underline{Y}|\underline{Z}} \right) \\
\langle \underline{X}, \underline{Y} \rangle &= \alpha_{\underline{X}|\underline{Z}} \sigma^2_{\underline{\epsilon}_{\underline{Z}}} \left(\alpha_{\underline{M}|\underline{X}} \alpha_{\underline{X}|\underline{Z}} \alpha_{\underline{Y}|\underline{M}} + \alpha_{\underline{X}|\underline{Z}} \alpha_{\underline{Y}|\underline{X}} + \alpha_{\underline{Y}|\underline{Z}} \right) + \sigma^2_{\underline{\epsilon}_{\underline{X}}} \left(\alpha_{\underline{M}|\underline{X}} \alpha_{\underline{Y}|\underline{M}} + \alpha_{\underline{Y}|\underline{X}} \right) \\
\langle \underline{M}, \underline{Y} \rangle &= \left\{ \begin{array}{l} \alpha_{\underline{M}|\underline{X}} \alpha_{\underline{X}|\underline{Z}} \sigma^2_{\underline{\epsilon}_{\underline{Z}}} \left(\alpha_{\underline{M}|\underline{X}} \alpha_{\underline{X}|\underline{Z}} \alpha_{\underline{Y}|\underline{M}} + \alpha_{\underline{X}|\underline{Z}} \alpha_{\underline{Y}|\underline{X}} + \alpha_{\underline{Y}|\underline{Z}} \right) \\ + \alpha_{\underline{M}|\underline{X}} \sigma^2_{\underline{\epsilon}_{\underline{X}}} \left(\alpha_{\underline{M}|\underline{X}} \alpha_{\underline{Y}|\underline{M}} + \alpha_{\underline{Y}|\underline{X}} \right) + \alpha_{\underline{Y}|\underline{M}} \sigma^2_{\underline{\epsilon}_{\underline{M}}} \end{array} \right.
\end{aligned}
\tag{99.3}
$$

Eq.(99.3) gives some covariance matrices for Fig.99.3.

Chapter 100

Targeted Estimator

This chapter is based on Refs.[6] and [31].

Targeted Estimator (TE) theory addresses the following concerns. Suppose $\Psi[P]$ is an estimator that depends on the full probability distribution P of a fixed bayesian network. $\Psi[P]$ is a functional (i.e., a function of a function) of P. If P is perturbed by a small amount δP, we get $\Psi[P+\delta P]$. $\Psi[P+\delta P]$ can be expanded in powers of δP. The term linear in δP defines the "influence function"; it also defines the "functional derivative" of Ψ with respect to P . *Why are influence functions useful?* The influence function measures, to first order in δP, how the estimator Ψ responds to a perturbation δP in P. In general, Ψ does not have to be a counterfactual estimator, but it might be one, like an estimator of ATE, or PNS or whatever. *So what is this good for?* It is a way of generating linear "targeted" estimators that are less noisy and converge more quickly. It measures the sensitivity of an estimator to perturbations in P. It does not, however, measure sensitivity to changes in the DAG (the DAG is fixed throughout). And it does not generate new estimands.

The goal of TE and the strategy one uses to achieve it, is explained more precisely in the next section.

100.1 Goal, Strategy, and Rationale of TE theory

Let $\underline{b} = (\underline{b}_1, \underline{b}_2, \ldots, \underline{b}_n)$ denote the n nodes of a Bayesian Network, and let $P_{\underline{b}}(b)$ for $b \in val(\underline{b})$ denote the full probability distribution of the bnet.

Consider a population Σ of individuals $\sigma \in \Sigma$ with $N = |\Sigma|$. The **empirical probability distribution** $P_N : val(\underline{b}) \to [0,1]$ for this bnet is defined by

$$P_N(b) = \frac{1}{N} \sum_{\sigma} \delta(b, b_\sigma) \tag{100.1}$$

$$\sum_b P_N(b) f(b) = \frac{1}{N} \sum_{\sigma} f(b_\sigma) \tag{100.2}$$

As $N \to \infty$, $P_N(b)$ tends to the probability distribution $P_{\underline{b}}(b)$ of the bnet.

Figure 100.1: Example of bnet considered in TE theory.

Let $\underline{b} = (\underline{X}, \underline{\xi})$ and $\underline{X} = (\underline{d}, \underline{x}, \underline{y})$, where node $\underline{d} \in \{0, 1\}$ denotes a decision to treat a patient, node $\underline{y} \in \{0, 1\}$ denotes the treatment outcome, multi-node $\underline{x}$ denotes the covariates that are good controls, and multi-node $\underline{\xi}$ denotes the covariates that we don't want to control. See Fig.100.1 for an example of a bnet that fits this description.

The curve-fit $\widehat{y}$ of y is a function $\widehat{y} : val(\underline{d}) \times val(\underline{x}) \to \mathbb{R}$ that minimizes the **loss (a.k.a. loss functional)** $\mathcal{L}$ given by

$$\mathcal{L}[P, \widehat{y}] = \sum_X P(X) \widehat{\mathcal{L}}[y, \widehat{y}(d, x)]^2 \ , \tag{100.3}$$

where $\widehat{\mathcal{L}}(a, b)$, the **loss curve-fit**, is a non-negative function that vanishes when $a = b$. The function $\widehat{\mathcal{L}}$ is designed to minimize a particular kind of error. In this chapter, the $\widehat{\mathcal{L}}$ is designed to reduce ATE error.

The estimate $\Psi[P, \widehat{y}]$ for the curve-fit $\widehat{y}$ is defined by

$$\Psi[P, \widehat{y}] = \sum_X P(X) \widehat{y}(d, x) \tag{100.4}$$

Unfortunately, the words "estimator" and "estimate" are often used interchangeably. See Section C.24. In this chapter, we use $(\widehat{y}, \Psi)$ for our (estimator, estimate) pair, and refer to estimators as curve-fits.

Let

$$\delta P(X) = P(X) - P_{in}(X) \tag{100.5}$$

where $P, P_{in} : val(\underline{X}) \to [0, 1]$ are probability distributions.

Define $\delta \widehat{y}(X)$ by

$$\delta \widehat{y}(X) = \frac{\widehat{y}(d, x) \delta P(X)}{P(X)} \tag{100.6}$$

Hence

$$P(X) \delta \widehat{y}(X) = \widehat{y}(d, x) \delta P(X) \tag{100.7}$$

Since $\Psi[P, \widehat{y}]$ is linear in P and $\widehat{y}$, it follows that

$$\Psi[P, \widehat{y} + \delta\widehat{y}] = \Psi[P + \delta P, \widehat{y}] \tag{100.8}$$

Suppose P_{in++} satisfies

$$P_{in++} = \underset{P_{in}}{\operatorname{argmin}}\, \mathcal{L}[P_{in}, \widehat{y}] \tag{100.9}$$

and

$$\lim_{N\to\infty} \Psi[P_N, \widehat{y} + \delta\widehat{y}] = \lim_{N\to\infty} \Psi[\underbrace{P_N + \delta P}_{P_{in++}}, \widehat{y}] = \lim_{N\to\infty} \Psi[P_N, \widehat{y}]\ . \tag{100.10}$$

Eq.(100.10) is illustrated in Fig.100.2.

The goal of TE theory is to find, given a curve-fit $\widehat{y}$, a new curve-fit $\widehat{y} + \delta\widehat{y}$ so that the estimate $\Psi[P_N, \widehat{y} + \delta\widehat{y}]$ has better behavior as $N \to \infty$ than $\Psi[P_N, \widehat{y}]$ (i.e., converges faster, has smaller bias and variance).

Figure 100.2: Plot of $\Psi[P, \widehat{y}] \in \mathbb{R}$ versus P at fixed $\widehat{y}$. In reality, the P are not real numbers but functions.

100.2 Functional Calculus

Define the Hilbert space of square integrable functions over $\underline{X} \in val(\underline{X})$ by

$$\mathcal{H}_{\underline{X}} = \{h : (h : val(\underline{X}) \to \mathbb{R}) \text{ and } \sum_{X \in S_{\underline{X}}} [h(X)]^2 < \infty\} \tag{100.11}$$

For any $f, g \in \mathcal{H}_{\underline{X}}$, define the dot product (a.k.a. inner product) of f and g by

$$f \cdot g = \sum_X f(X)g(X) \tag{100.12}$$

and the norm

$$\| f \|_P = \sqrt{\sum_X [f(X)]^2} \ . \tag{100.13}$$

Suppose $P : val(\underline{X}) \to [0,1]$ is a probability distribution. Note that $P \in \mathcal{H}_{\underline{X}}$. For any $f, g \in \mathcal{H}_{\underline{X}}$, define the P expected value by

$$\langle f \rangle_P = P \cdot f \tag{100.14}$$

and the P covariance by

$$\langle f, g \rangle_P = \langle fg \rangle_P - \langle f \rangle_P \langle g \rangle_P \tag{100.15}$$

Suppose $\Psi[\eta] \in \mathbb{R}$ is a real valued function that depends on a function $\eta \in \mathcal{H}_{\underline{X}}$. $\Psi[\eta]$ is said to be a **functional** of η. Define the **functional derivative or gradient** [1] of $\Psi[\eta]$ with respect to η, as follows

$$\frac{\delta \Psi[\eta]}{\delta \eta(a)} = \lim_{\epsilon \to 0} \frac{\Psi[\eta(x) + \epsilon \frac{\delta(x,a)}{\Delta x}] - \Psi[\eta(x)]}{\epsilon} \tag{100.16}$$

where $\delta(x,a)$ is the Kronecker delta function. For example,

$$\frac{\delta}{\delta \eta(a)} \sum_x \Delta x \ \eta(x) h(x) = \sum_x \Delta x \frac{\delta(x,a)}{\Delta x} h(x) = h(a) \tag{100.17}$$

Let $\delta(x-a)$ denote the Dirac delta function. If we replace $\frac{\delta(x,a)}{\Delta x} \to \delta(x-a)$ and $\sum_x \Delta x \to \int dx$, we go from the discrete to the continuous version of the functional derivative.

$$\sum_x \Delta x \frac{\delta(x,a)}{\Delta x} \to \int dx \, \delta(x-a) \tag{100.18}$$

In this chapter, we will use only the discrete version. The Δx in the numerator and the one in the denominator, always cancel each other when we go from discrete to continuous, so we can set $\Delta x = 1$ with impunity.

We will also use the notation

$$\nabla \Psi[\eta](a) = \frac{\delta \Psi[\eta]}{\delta \eta(a)} \ . \tag{100.19}$$

Suppose $\eta, \eta_0 \in \mathcal{H}_{\underline{X}}$. Define the **functional Taylor expansion** of $\Psi[\eta]$ at η_0, as follows:

$$\Psi[\eta] = \Psi[\eta_0] + \sum_x \left[\frac{\delta \Psi[\eta]}{\delta \eta(x)} \right]_{\eta = \eta_0} \delta \eta(x) + \frac{1}{2!} \sum_{x,x'} \left[\frac{\delta^2 \Psi[\eta]}{\delta \eta(x) \delta \eta(x')} \right]_{\eta = \eta_0} \delta \eta(x) \delta \eta(x') + \cdots \tag{100.20}$$

[1] Functional derivatives are commonly used in Physics especially in Quantum Field Theory. See Ref.[137] for more information about them.

where we abbreviate $\eta = \eta(x)$, $\eta_0 = \eta_0(x)$ and $\delta\eta(x) = \eta(x) - \eta_0(x)$.

Define the **functional directional derivative of $\Psi[\eta]$ in the $h \in \mathcal{H}_{\underline{X}}$ direction** by $h \cdot \frac{\delta\Psi[\eta]}{\delta\eta}$. If one compares functional calculus with vector calculus, we see that $\frac{\delta\Psi[\eta]}{\delta\eta}$ corresponds to a gradient $\nabla f(\vec{x})$ and $h \cdot \frac{\delta\Psi[\eta]}{\delta\eta}$ corresponds to a directional derivative $\vec{d} \cdot \nabla f(\vec{x})$. For $|\vec{d}| << 1$, $\vec{d} \cdot \nabla f(\vec{x})$ approximates the change in $f(\vec{x})$ when $\vec{x}$ moves from $\vec{x}$ to $\vec{x} + \vec{d}$

100.3 Linear Approximation of $\Psi[P_N]$

Consider probability distributions $P, P_{in} : val(\underline{X}) \to [0, 1]$. The **linear approximation (a.k.a. one-step-approximation)** to the Taylor expansion of $\Psi[P]$ at P_{in} is given by

$$\underbrace{\Psi[P] - \Psi[P_{in}]]}_{\delta\Psi[P,P_{in}]} = \sum_X \underbrace{\left[\frac{\delta\Psi[P]}{\delta P(X)}\right]_{P=P_{in}}}_{\nabla\Psi[P_{in}](X)} \underbrace{\delta P(X)}_{P(X)-P_{in}(X)} + \mathcal{R}[P, P_{in}] \tag{100.21}$$

If we set

$$\nabla\Psi_{in} = \langle \nabla\Psi[P_{in}] \rangle_{P_{in}} \ , \tag{100.22}$$

then Eq.(100.21) becomes

$$\delta\Psi[P, P_{in}] \ = \ \sum_X P(X) \underbrace{\{\nabla\Psi[P_{in}](X) - \nabla\Psi_{in}\}}_{\lambda(X)} + \mathcal{R}[P, P_{in}] \tag{100.23}$$

$\lambda(X)$ is called the **efficient influence curve (EIF)**.

Recall the Cauchy-Schwartz (CS) inequality for $\vec{a}, \vec{b} \in \mathbb{R}^n$:

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos\theta \leq |\vec{a}||\vec{b}| \tag{100.24}$$

Note that

$$\begin{aligned}
\langle \lambda \rangle_P &= \sum_X \lambda(X) \sqrt{P(X)} \sqrt{P(X)} && (100.25)\\
&= (\sqrt{P}\lambda) \cdot \sqrt{P} && (100.26)\\
&\leq \sqrt{\sqrt{P}\lambda \cdot \sqrt{P}\lambda} \underbrace{\sqrt{\sqrt{P} \cdot \sqrt{P}}}_{=1} \quad \text{(by CS inequality)} && (100.27)\\
&\leq \sqrt{\langle \lambda^2 \rangle_P} && (100.28)
\end{aligned}$$

When $P = P_N$,

$$\Psi[P_N] \approx \Psi[P_{in}] + \frac{1}{N}\sum_{\sigma} \lambda(X_\sigma) \tag{100.29}$$

If the random variables $\{\underline{X}_\sigma\}_{\sigma\in\Sigma}$ are i.i.d. with probability distribution $P_{\underline{X}}(X)$, then, as $N \to \infty$, $\Psi[P_N]$ tends to a normally distributed random variable with mean

$$\begin{aligned} \langle \Psi[P_N] \rangle_{P_{\underline{X}}} &= \Psi[P_{in}] + \frac{1}{N}\sum_{\sigma} \langle \lambda(X_\sigma) \rangle_{P_{\underline{X}}} && (100.30) \\ &= \Psi[P_{in}] + \langle \lambda \rangle_{P_{\underline{X}}} && (100.31) \end{aligned}$$

and variance

$$\begin{aligned} \langle \Psi[P_N], \Psi[P_N] \rangle_{P_{\underline{X}}} &= \frac{1}{N^2}\sum_{\sigma}\sum_{\sigma'} \left\langle \lambda(X_\sigma), \lambda(X^{\sigma'}) \right\rangle_{P_{\underline{X}}} && (100.32) \\ &= \frac{1}{N} \underbrace{\langle \lambda, \lambda \rangle_{P_{\underline{X}}}}_{\langle \lambda^2 \rangle_{P_{\underline{X}}} - \langle \lambda \rangle^2_{P_{\underline{X}}}} && (100.33) \end{aligned}$$

Later on, we will discuss the so called TMLE estimate, for which P_{in} takes a special value P_{in++} that makes $\langle \lambda \rangle_{P_{\underline{X}}} = 0$, so $\Psi[P_N]$ tends to a normally distributed random variable with mean $\Psi[P_{in++}]$ and variance $\frac{1}{N}\langle \lambda^2 \rangle_{P_{\underline{X}}}$.

100.4 ATE estimand

If we set

$$\mathcal{Y}_{|d,x}[P] = \sum_{y} yP(y|d,x) = P(\underline{y} = 1|d,x) \tag{100.34}$$

and

$$\mathcal{Y}_{|d}[P] = \sum_{x} \mathcal{Y}_{|d,x}[P]P(x) \ , \tag{100.35}$$

then the **Average Treatment Effect (ATE)** is defined as follows

$$ATE = \mathcal{Y}_{|1}[P_N] - \mathcal{Y}_{|0}[P_N] \tag{100.36}$$

The rest of this chapter is devoted to discussing ATE estimates.

In discussing ATE, it is convenient to define the **propensity** $g[P]$:

$$g[P](y) = P(\underline{d} = 1|y) \tag{100.37}$$

and the Kronecker difference function $\Delta(d)$ for $d \in \{0,1\}$:

$$\Delta(d) = \delta(d,1) - \delta(d,0) \tag{100.38}$$
$$= (2d-1)\mathbb{1}(d \in \{0,1\}) \ . \tag{100.39}$$

Claim 174

$$\left\langle y \frac{\delta(d',d)}{P(d|x)} \right\rangle_P = \mathcal{Y}_{|d'}[P] \tag{100.40}$$

proof:

$$\left\langle y \frac{\delta(d',d)}{P(d|x)} \right\rangle_P = \sum_d \sum_y \sum_x P(y|d,x) \cancel{P(d|x)} P(x) y \frac{\delta(d',d)}{\cancel{P(d|x)}} \tag{100.41}$$
$$= \sum_y \sum_x P(y|\underline{d} = d', x) P(x) y \tag{100.42}$$
$$= \mathcal{Y}_{|\underline{d}=d'}[P] \tag{100.43}$$

QED

100.5 ATE estimates

100.5.1 Ψ^E

Empirical (E) estimate Ψ^E.

$$P_N(y,d,x) = \frac{1}{N} \sum_\sigma \delta(y,y_\sigma)\delta(d,d_\sigma)\delta(x,x_\sigma) \tag{100.44}$$

$$P_N(d,x) = \frac{1}{N} \sum_\sigma \delta(d,d_\sigma)\delta(x,x_\sigma) \tag{100.45}$$

$$P_N(x) = \frac{1}{N} \sum_\sigma \delta(x,x_\sigma) \tag{100.46}$$

$$P_N(y|d,x) = \frac{P_N(y,d,x)}{P_N(d,x)} \tag{100.47}$$

$$\Psi^E = \sum_x P_N(x) \sum_y y \sum_d P_N(y|d,x)\Delta(d) \tag{100.48}$$

100.5.2 Ψ^G

G estimate Ψ^G (a.k.a. g-computing or g-formula estimate).

Use Generalized Linear Modeling (GLM)[2] to approximate y_σ:

$$y_\sigma \approx E_{\underline{y}|d_\sigma,x_\sigma;\widehat{\beta}}[\underline{y}] \ , \tag{100.49}$$

where $\widehat{\beta}$ are the best curve fit parameters.

$$\Psi^G = \sum_x P_N(x)\{E_{\underline{y}|d=1,x;\widehat{\beta}}[\underline{y}] - E_{\underline{y}|d=0,x;\widehat{\beta}}[\underline{y}]\} \tag{100.50}$$

100.5.3 Ψ^{IPW}

Inverse Propensity Weighted (IPW) estimate Ψ^{IPW} (a.k.a. Inverse Probability of Treatment Weighted (IPTW) estimate).

Assume propensity $P(\underline{d} = 1|x)$ is known.

Define

$$\Psi^{IPW}[P] \quad = \quad \left\langle y\frac{\Delta(d)}{P(d|x)}\right\rangle_P \tag{100.51}$$

$$\Psi^{IPW} = \Psi^{IPW}[P_N] = \left\langle y\frac{\Delta(d)}{P(d|x)}\right\rangle_{P_N} = \frac{1}{N}\sum_\sigma y_\sigma \frac{\Delta(d_\sigma)}{P(d_\sigma|x_\sigma)} \tag{100.52}$$

100.5.4 Ψ^{LIPW}

Linearized IPW (LIPW) estimate Ψ^{LIPW}.

Ψ^{LIPW} is the linear approximation of $\Psi^{IPW}[P_N]$ at point P_{in}:

$$\Psi^{LIPW} \quad = \quad \Psi^{IPW}[P_{in}] + \left\langle \nabla\Psi^{IPW}[P_{in}]\right\rangle_{P_N} - \nabla\Psi^{IPW}_{in} \tag{100.53}$$

Claim 175

$$\nabla\Psi^{IPW}[P](X) = \mathcal{Y}_{|1,x}[P] - \mathcal{Y}_{|0,x}[P] + \frac{\Delta(d)}{P(d|x)}(y - \mathcal{Y}_{|d,x}[P]) \tag{100.54}$$

proof:

$$\frac{\delta\Psi^{IPW}[P]}{\delta P(X)} \quad = \quad \sum_{X'} y'\Delta(d')\frac{\delta}{\delta P(X)}\frac{P(X')}{P(d'|x')} \tag{100.55}$$

[2]GLM is discussed in Chapter 35.

$$\frac{\delta}{\delta P(X)}\frac{P(X')}{P(d'|x')} = \frac{\delta(X,X')}{P(d'|x')} - \frac{P(X')}{[P(d'|x')]^2}\frac{\delta P(d'|x')}{\delta P(X)} \tag{100.56}$$

$$= \frac{\delta(X,X')}{P(d'|x')} - \frac{P(X')}{P(d'|x')}\frac{\delta \ln P(d'|x')}{\delta P(X)} \tag{100.57}$$

$$= \underbrace{\frac{\delta(X,X')}{P(d'|x')}}_{\delta^3/P(d'|x')} - P(y'|d',x')P(x')\frac{\delta \ln P(d'|x')}{\delta P(X)} \tag{100.58}$$

$$\frac{\delta \ln P(d',x')}{\delta P(X)} = \frac{1}{P(d',x')}\sum_{y'}\frac{\delta P(X')}{\delta P(X)} \tag{100.59}$$

$$= \underbrace{\frac{\delta(d,d')\delta(x,x')}{P(d',x')}}_{\delta^2/P(d',x')} \tag{100.60}$$

$$\frac{\delta \ln P(x')}{\delta P(X)} = \underbrace{\frac{\delta(x,x')}{P(x')}}_{\delta^1/P(x')} \tag{100.61}$$

$$\frac{\delta \ln P(d'|x')}{\delta P(X)} = \frac{\delta^2}{P(d',x')} - \frac{\delta^1}{P(x')} \tag{100.62}$$

$$\frac{\delta}{\delta P(X)}\frac{P(X')}{P(d'|x')} = \frac{\delta^3}{P(d'|x')} - P(y'|d',x')P(x')\left[\frac{\delta^2}{P(d',x')} - \frac{\delta^1}{P(x')}\right] \tag{100.63}$$

$$\sum_{X'} y'\Delta(d')\left[\frac{\delta^3}{P(d'|x')}\right] = \boxed{\frac{\Delta(d)}{P(d|x)}y} \tag{100.64}$$

$$\sum_{X'} y'\Delta(d')\left[\frac{-P(y'|d',x')\delta^2}{P(d'|x')}\right] = -\sum_{y'} y'\Delta(d)\frac{P(y'|d,x)}{P(d|x)} \tag{100.65}$$

$$= \boxed{\frac{\Delta(d)}{P(d|x)}(-\mathcal{Y}_{|d,x}[P])} \tag{100.66}$$

$$\sum_{X'} y'\Delta(d')\left[P(y'|d',x')\delta^1\right] = \sum_{y'}\sum_{d'} y'\Delta(d')P(y'|d',x) \tag{100.67}$$

$$= \boxed{\mathcal{Y}_{|1,x}[P] - \mathcal{Y}_{|0,x}[P]} \tag{100.68}$$

QED

Claim 176

$$\left\langle \nabla\Psi^{IPW}[P] \right\rangle_P = \Psi^{IPW}[P] \tag{100.69}$$

Hence,

$$\nabla\Psi^{IPW}_{in} = \left\langle \nabla\Psi^{IPW}[P_{in}] \right\rangle_{P_{in}} = \Psi^{IPW}[P_{in}] \tag{100.70}$$

proof:

$$\nabla\Psi^{IPW}[P](X) = \mathcal{Y}_{|1,x}[P] - \mathcal{Y}_{|0,x}[P] + \frac{\Delta(d)}{P(d|x)}(y - \mathcal{Y}_{|d,x}[P]) \tag{100.71}$$

$$\begin{aligned} \left\langle \mathcal{Y}_{|1,x}[P] - \mathcal{Y}_{|0,x}[P] \right\rangle_P &= \sum_x P(x)(\mathcal{Y}_{|1,x}[P] - \mathcal{Y}_{|0,x}[P]) && (100.72) \\ &= \Psi^{IPW}[P] && (100.73) \end{aligned}$$

$$\left\langle \frac{\Delta(d)}{P(d|x)} y \right\rangle_P = \Psi^{IPW}[P] \tag{100.74}$$

$$\begin{aligned} \left\langle \frac{\Delta(d)}{P(d|x)} \mathcal{Y}_{|d,x}[P] \right\rangle_P &= \sum_y \sum_x \sum_d P(y|d,x)P(x)\Delta(d)\mathcal{Y}_{|d,x}[P] && (100.75) \\ &= \sum_x \sum_d P(x)\Delta(d)\mathcal{Y}_{|d,x}[P] && (100.76) \\ &= \Psi^{IPW}[P] && (100.77) \end{aligned}$$

QED

Note the following cancellation:

$$\begin{aligned} \Psi^{IPW}[P] &= \Psi^{IPW}[P_{in}] + \left\langle \nabla\Psi^{IPW}[P_{in}] \right\rangle_P - \nabla\Psi^{IPW}_{in} + \mathcal{R}^{IPW}[P, P_{in}] && (100.78) \\ &= \cancel{\Psi^{IPW}[P_{in}]} + \left\langle \nabla\Psi^{IPW}[P_{in}] \right\rangle_P - \cancel{\Psi^{IPW}[P_{in}]} + \mathcal{R}^{IPW}[P, P_{in}] && (100.79) \\ &= \left\langle \nabla\Psi^{IPW}[P_{in}] \right\rangle_P + \mathcal{R}^{IPW}[P, P_{in}] && (100.80) \end{aligned}$$

Claim 177

$$\mathcal{R}^{IPW}[P, P_{in}] = -\sum_x P(x) \sum_d \Delta(d) \left(P(d|x) - P_{in}(d|x)\right) \left(\frac{\mathcal{Y}_{|d,x}[P] - \mathcal{Y}_{|d,x}[P_{in}]}{P_{in}(d|x)} \right) \tag{100.81}$$

proof:

$$\mathcal{R}^{IPW}[P, P_{in}] = \Psi^{IPW}[P] - \left\langle \nabla\Psi^{IPW}[P_{in}] \right\rangle_P \tag{100.82}$$

$$= \left\langle y\frac{\Delta(d)}{P(d|x)} - \left(\mathcal{Y}_{|1,x}[P_{in}] - \mathcal{Y}_{|0,x}[P_{in}] + \frac{\Delta(d)}{P_{in}(d|x)}(y - \mathcal{Y}_{|d,x}[P_{in}])\right)\right\rangle_P \tag{100.83}$$

$$= \begin{cases} \sum_x P(x)\left(-\mathcal{Y}_{|1,x}[P_{in}] + \mathcal{Y}_{|0,x}[P_{in}]\right) \\ + \sum_{d,x} P(d,x)\left(\frac{\Delta(d)}{P_{in}(d|x)}\mathcal{Y}_{|d,x}[P_{in}]\right) \\ + \sum_{y,d,x} P(y,d,x)\left(\frac{1}{P(d|x)} - \frac{1}{P_{in}(d|x)}\right) y\Delta(d) \end{cases} \tag{100.84}$$

$$= \sum_x P(x) \sum_d \Delta(d) \begin{cases} \left(\frac{P(d|x)}{P_{in}(d|x)} - 1\right)\mathcal{Y}_{|d,x}[P_{in}] \\ + \sum_y P(y|d,x)\left(\frac{P_{in}(d|x) - P(d|x)}{P_{in}(d|x)}\right) y \end{cases} \tag{100.85}$$

$$= \sum_x P(x) \sum_d \Delta(d) \left(\frac{P(d|x) - P_{in}(d|x)}{P_{in}(d|x)}\right)(\mathcal{Y}_{|d,x}[P_{in}] - \mathcal{Y}_{|d,x}[P]) \tag{100.86}$$

QED

Claim 177 allows us to put a bound on the absolute value of the remainder $\mathcal{R}^{IPW}$:

$$|\mathcal{R}^{IPW}[P, P_{in}]| \leq \sum_d \sum_x \underbrace{\sqrt{P(x)}\,|P(d|x) - P_{in}(d|x)|}_{A(d,x)} \underbrace{\sqrt{P(x)}\left|\frac{\mathcal{Y}_{|d,x}[P] - \mathcal{Y}_{|d,x}[P_{in}]}{P_{in}(d|x)}\right|}_{B(d,x)}$$

$$\text{(because } |\Delta(d)| = 1\text{, and } \left|\sum_i a_i\right| \leq \sum_i |a_i| \text{)} \tag{100.87}$$

$$\leq \sum_d \underbrace{\sqrt{\sum_x A^2(d,x)}}_{A_P(d)} \underbrace{\sqrt{\sum_x B^2(d,x)}}_{B_P(d)} \quad \text{(by CS inequality) .} \tag{100.88}$$

Define

$$A_P(d') = \sqrt{\sum_x P(x)\left(P(d'|x) - P_{in}(d'|x)\right)^2} \tag{100.89}$$

$$= \sqrt{\left\langle \left(P(d'|x) - P_{in}(d'|x)\right)^2 \right\rangle_P} \tag{100.90}$$

and

$$B_P(d') = \sqrt{\sum_x P(x) \left(\frac{\mathcal{Y}_{|d',x}[P] - \mathcal{Y}_{|d',x}[P_{in}]}{P_{in}(d'|x)} \right)^2} \tag{100.91}$$

$$= \sqrt{\left\langle \left(\frac{\mathcal{Y}_{|d',x}[P] - \mathcal{Y}_{|d',x}[P_{in}]}{P_{in}(d'|x)} \right)^2 \right\rangle_P} . \tag{100.92}$$

Then

$$|\mathcal{R}^{LIPW}[P_N, P_{in}]| \leq \sum_{d'=0}^{1} A_{P_N}(d') B_{P_N}(d') \tag{100.93}$$

If either $A_{P_N}(1) = A_{P_N}(0) = 0$ (i.e., zero error in the propensities) or $B_{P_N}(0) = B_{P_N}(1) = 0$ (i.e., zero bias), then $\mathcal{R}^{LIPW}[P_N, P_{in}] = 0$. This property of Ψ^{LIPW} is referred to as **double robustness**.

100.5.5 Ψ^{LIPW++} (a.k.a. Ψ^{TMLE})

LIPW++ estimate Ψ^{LIPW++} (a.k.a, targeted minimum loss estimate (TMLE)) .

Ψ^{LIPW++} is the linear approximation of $\Psi^{IPW}[P_N]$ at the point P_{in++}, where the linear term of its Taylor expansion vanishes:

$$\Psi^{LIPW++} = \Psi^{TMLE} = \Psi^{IPW}[P_{in++}] + \underbrace{\left\langle \nabla \Psi^{IPW}[P_{in++}] \right\rangle_{P_N} - \nabla \Psi^{IPW}_{in++}}_{=0} \tag{100.94}$$

This property of Ψ^{TMLE} that the linear term in its Taylor expansion at P_{in++} vanishes is referred to as **substitution invariance**, and Ψ^{TMLE} is said to be a **substitution estimate**. A substitution estimate is very desirable because its absolute value is bounded, unlike the value of Ψ^{LIPW}.

Ψ^{TMLE} is both a doubly robust estimate and a substitution estimate.

Claim 175 allows us express more explicitly the constraint that defines P_{in++}:

$$0 = P_N \cdot \nabla \Psi^{IPW}[P_{in++}] - \nabla \Psi^{IPW}_{in++} \tag{100.95}$$

$$= -\nabla \Psi^{IPW}_{in++} + \left\{ \begin{array}{l} \overbrace{\frac{1}{N} \sum_\sigma \left(\mathcal{Y}_{|1,x_\sigma}[P_{in++}] - \mathcal{Y}_{|0,x_\sigma}[P_{in++}] \right)}^{\approx \nabla \Psi^{IPW}[P_{in++}]} \\ + \frac{1}{N} \sum_\sigma \frac{\Delta(d_\sigma)}{P_{in++}(d_\sigma|x_\sigma)} (y_\sigma - \mathcal{Y}_{|d_\sigma,x_\sigma}[P_{in++}]) \end{array} \right. \tag{100.96}$$

$$= \frac{1}{N} \sum_\sigma \underbrace{\frac{\Delta(d_\sigma)}{P_{in++}(d_\sigma|x_\sigma)} (y_\sigma - \mathcal{Y}_{|d_\sigma,x_\sigma}[P_{in++}])}_{\lambda(X_\sigma)} \tag{100.97}$$

Figure 100.3: This figure portrays the space of functions $\mathcal{H}_{\underline{X}}$ as if it were the real plane $\mathbb{R}^2$, and the functional $\mathcal{L}[P, \widehat{y} = fixed] : \mathcal{H}_{\underline{X}} \to \mathbb{R}$ as if it were a real valued function on $\mathbb{R}^2$. It shows the constant contours of the loss functional $\mathcal{L}[P, \widehat{y}]$ at fixed $\widehat{y}$ in green. P_N for $N = 10, 100, 1000$ represent empirical distributions. The loss is non-negative and it equals zero when $\epsilon = 0$ and $P = P_{in++}$.

Note that function $\lambda(X_\sigma)$ defined in Eq.(100.97) can be positive or negative. Hence, it can't be defined as the loss curve-fit, because a loss curve-fit must be non-negative. However, it can be defined as the derivative of a loss curve-fit. Suppose we define an $\epsilon \in \mathbb{R}$ parameterized family of probability distributions $P_\epsilon : val(\underline{X}) \to [0, 1]$, and we expand the loss $\mathcal{L}[P_\epsilon, \widehat{y}]$ in powers of ϵ:

$$\mathcal{L}[P_\epsilon, \widehat{y}] = \mathcal{L}[P_0, \widehat{y}] + \epsilon\{\partial_\epsilon \mathcal{L}[P_\epsilon, \widehat{y}]\}_{\epsilon=0} + \mathcal{O}(\epsilon^2) \tag{100.98}$$

The parameter ϵ is called the **fluctuation parameter**. The function $\mathcal{L}[P_\epsilon, \widehat{y}]$ is obviously not unique because all we know about it is the value of its ϵ derivative in the vicinity of $\epsilon = 0$. Next we will pick a convenient $\mathcal{L}[P_\epsilon, \widehat{y}]$ that satisfies

$$\mathcal{L}[P_\epsilon, \widehat{y}] \geq 0, \quad \mathcal{L}[P_0, \widehat{y}] = 0, \quad \partial_\epsilon\{\mathcal{L}[P_\epsilon, \widehat{y}]\}_{\epsilon=0} = 0 \tag{100.99}$$

Use as loss curve-fit $\widehat{\mathcal{L}}$ the Cross Entropy $CE(p \parallel q)$ for $p, q \in [0, 1]$

$$\widehat{\mathcal{L}} = CE(p \parallel q) = -[p \ln q + (1 - p) \ln(1 - q)] \tag{100.100}$$

$\widehat{\mathcal{L}} \geq 0$ and attains its minimum when $p = q$. When $p = q$, it equals the entropy of p, i.e., when $p = q$, $\widehat{\mathcal{L}} = -\sum_{x \in \{0,1\}} P(x) \ln P(x) = H(P)$, where $P(0) = p, P(1) = 1 - p$.

For some $y \in \{0, 1\}$ and $\epsilon \in \mathbb{R}$, make the following substitutions in the loss curve-fit $\widehat{\mathcal{L}}$ and call it $\widehat{\mathcal{L}} = \widehat{\mathcal{L}}(\beta, y, \widehat{y}, \epsilon)$.[3]

$$p = y, \quad q = \text{expit}[\text{logit}(\widehat{y}) + \epsilon\beta] \tag{100.101}$$

Note that since $p = y$ is binary, the minimum of this loss curve-fit is zero.

[3]To agree with the TE literature, we are using expit(x) (resp., logit(p)) to denote the sigmoid function (resp., log-odds function) which we normally denote in this book by smoid(x) (resp., lodds(p)).

Recall that in Section C.23, we proved that the derivative of $\text{expit}(x)$ satisfies

$$\text{expit}'(x) = \text{expit}(x)[1 - \text{expit}(x)] \ . \tag{100.102}$$

Hence,

$$\begin{aligned}
\lim_{\epsilon \to 0} \partial_\epsilon \widehat{\mathcal{L}}(\beta, y, \widehat{y}, \epsilon) &= \lim_{\epsilon \to 0} \left[-\frac{p}{q} + \frac{1-p}{1-q} \right] \partial_\epsilon q && (100.103) \\
&= \left[-\frac{y}{\widehat{y}} + \frac{1-y}{1-\widehat{y}} \right] \lim_{\epsilon \to 0} \left\{ \begin{array}{l} \text{expit}[\text{logit}(\widehat{y}) + \epsilon\beta] \\ * \{1 - \text{expit}[\text{logit}(\widehat{y}) + \epsilon\beta]\} \, \beta \end{array} \right. && (100.104) \\
&= \left[-\frac{y}{\widehat{y}} + \frac{1-y}{1-\widehat{y}} \right] \widehat{y}(1-\widehat{y})\beta && (100.105) \\
&= \beta[\widehat{y} - y] && (100.106)
\end{aligned}$$

If we define $\widehat{\mathcal{L}}(X)$ by

$$\widehat{\mathcal{L}}(X) = \widehat{\mathcal{L}} \left(\begin{array}{l} \beta = \frac{\Delta(d)}{P_{in++}(d|x)}, \\ y = y, \\ \widehat{y} = \mathcal{Y}_{|d,x}[P_{in++}], \\ \epsilon = \epsilon \end{array} \right) \ , \tag{100.107}$$

and $\mathcal{L}$ by

$$\mathcal{L} = \frac{1}{N} \sum_\sigma \widehat{\mathcal{L}}(X_\sigma) = P_N \cdot \widehat{\mathcal{L}} \ , \tag{100.108}$$

then

$$\left\{ \begin{array}{l} \mathcal{L} = P_N \cdot \widehat{\mathcal{L}} \geq 0 \\ \mathcal{L}_{\epsilon=0} = P_N \cdot \widehat{\mathcal{L}}_{\epsilon=0} = 0 \\ \{\partial_\epsilon \mathcal{L}\}_{\epsilon=0} = P_N \cdot \{\partial_\epsilon \widehat{\mathcal{L}}\}_{\epsilon=0} = 0 \end{array} \right. \tag{100.109}$$

100.6 Ψ^{TMLE} in practice

This section is based on Ref.[31].

In practice, one can calculate Ψ^{TMLE} by performing the following steps.

Below, "ML-fit" denotes a curve fitting obtained using any valid machine learning method, such as linear regression, a Neural Net, a decision tree, etc. The TE software often uses a "Super-Learner", a program that merges the results of multiple fits obtained via various ML methods and also does cross validation.

Below , $\{(\sigma, d_\sigma, x_\sigma, \boxed{y_\sigma}) : \sigma \in \Sigma\}$ represents a dataset. The dependent variable y_σ is boxed, the independent ones d_σ, x_σ (a.k.a. covariates) aren't.

1. Find curve-fit $\widehat{y}(d,x) = \mathcal{Y}_{|d,x}$ of outcome y

$$\{(\sigma, d_\sigma, x_\sigma, \boxed{y_\sigma}) : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{y}(d,x) \tag{100.110}$$

2. Estimate propensity $g(x)$

$$g(x) = P(\underline{d} = 1|x) \approx \frac{\sum_\sigma \delta(d_\sigma, 1)\delta(x_\sigma, x)}{\sum_\sigma \delta(x_\sigma, x)} \tag{100.111}$$

$$P(d|x) = dg(x) + (1-d)[1-g(x)] \tag{100.112}$$

$$\beta(d,x) = \frac{\Delta(d)}{P(d|x)} \tag{100.113}$$

3. Estimate fluctuation parameter ϵ

$$\eta(d,x) = \underbrace{\text{logit}[\widehat{y}(d,x)]}_{\lambda(d,x)} + \epsilon\beta(d,x) \tag{100.114}$$

$$\{(\sigma, \lambda(d_\sigma, x_\sigma), \beta(d_\sigma, x_\sigma), \boxed{\eta(d_\sigma, x_\sigma)} : \sigma \in \Sigma\} \xrightarrow{\text{ML-fit}} \widehat{\eta}(d,x) \tag{100.115}$$

4. Estimate ATE

$$ATE = \frac{1}{N}\sum_\sigma \{\text{expit}[\widehat{\eta}(d=1, x_\sigma)] - \text{expit}[\widehat{\eta}(d=0, x_\sigma)]\} \tag{100.116}$$

Chapter 101

Thermodynamics, a Causal Perspective

For a summary of Thermodynamics, see [187].

Modern day books on Thermodynamics derive its 3 laws from either classical or quantum statistical mechanics, or using classical or quantum stochastic equations (see Chapter 94). However, the 3 laws were originally derived from causal type arguments and experimentation, in much the same way that one derives a bnet as a hypothesis which is then tested. Fig.101.1 is a bnet for thermo that captures some of those causal arguments. The structural equations for the bnet are printed in blue.

Figure 101.1: Thermodynamics, a causal perspective. Extrinsic variables in green, Intrinsic ones in pink, and Legendre transforms of U in blue.

$$\Phi = U - TS - \sum_i \mu_i N_i \quad \text{(Grand Potential)} \tag{101.1a}$$

$$\{\mu_i\} = \frac{\partial U}{\partial \{N_i\}} \ \text{(chemical potential for species } i\text{)} \tag{101.1b}$$

$$\{N_i\} = \text{prior} \ \text{(number of particles of species } i\text{)} \tag{101.1c}$$

$$F = U - TS \ \text{(Helmholtz free energy)} \tag{101.1d}$$

$$G = U + pV - TS \ \text{(Gibbs free energy)} \tag{101.1e}$$

$$H = U + pV \ \text{(enthalpy)} \tag{101.1f}$$

$$p = -\frac{\partial U}{\partial V} \ \text{(pressure)} \tag{101.1g}$$

$$S = \text{prior} \ \text{(entropy)} \tag{101.1h}$$

$$T = \frac{\partial U}{\partial S} \ \text{(temperature)} \tag{101.1i}$$

$$U = U(S, V, \{N_i\}) \ \text{(internal energy)} \tag{101.1j}$$

$$V = \text{prior} \ \text{(volume)} \tag{101.1k}$$

Chapter 102

Time Series Analysis: ARMA and VAR

This chapter is based mostly on the book Ref.[26] on time series analysis by Hamilton, and on the lectures Ref. [38] by Chung-Ming Kuan. In writing this chapter, we also profited greatly from numerous Wikipedia entries on time series analysis, such as the entries on time series (Ref.[188]), ARMA time series (Ref.[107]), AR time series (Ref.[106]), MA time series (Ref.[165]), and VAR time series (Ref.[194]).

We cover only a small fraction of the treasures covered in those sources, and only cover stationary time-series. Non-stationary time series we don't even touch. But we hope to have covered enough to pique our readers's interest in time series analysis, and make him/her appreciate how bnets make time series much more intuitive and fun. The time-series considered in this chapter can be represented by one of the simplest types of bnets, namely, the LDEN bnets introduced in Chapter 52.

As usual, for $t, t_a, t_b \in \mathbb{Z}$, let $\mathbb{Z}_{<t} = \{t-1, t-2, t-3, \ldots\}$, $\mathbb{Z}_{[t_a,t_b]} = \{t_a, t_a + 1, \ldots, t_b\}$, etc.

Let $\underline{x}_t \in \mathbb{R}$. A **time series (t-series)**, denoted variously by $\{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_{nt}\} = \{\underline{x}_t\}_{t=1}^{nt} = \{\underline{x}_t\}_{\forall t}$, is a set of real numbers index by a discrete set of times $\mathbb{Z}_{[0,nt]}$.

For $t_a < t_b$, let $\underline{x}_{[t_a,t_b]} = (\underline{x}_{t_a}, \underline{x}_{t_a+1}, \ldots, \underline{x}_{t_b})$. Let $x_{<t} = (\ldots, x_{t-2}, \underline{x}_{t-1})$.

102.1 White noise

By **white noise** $\{\underline{n}_t\}_{\forall t} \sim WN(0, \sigma^2)$ we mean a t-series $\{\underline{n}_t\}_{\forall t}$ that satisfies

$$E[\underline{n}_t] = 0 \tag{102.1}$$

and

$$\langle \underline{n}_t, \underline{n}_{t'} \rangle = \sigma^2 \delta(t, t') \ . \tag{102.2}$$

Gaussian white noise $\{\underline{n}_t\}_{\forall t} \sim WN(0, \sigma^2)_{\mathcal{N}}$ is white noise such that also $\underline{n}_t \sim \mathcal{N}(0, \sigma^2)$.

102.2 Backshift operator

$\mathcal{B}$ is the backshift (a.k.a. lag) operator. For any t-series $\{x_t\}_{\forall t}$, $\{y_t\}_{\forall t}$ and scalars $a, b \in \mathbb{R}$,

$$\mathcal{B}\underline{x}_t = \underline{x}_{t-1} \tag{102.3}$$

$$\mathcal{B}(a\underline{x}_t + b\underline{y}_t) = a\mathcal{B}(\underline{x}_t) + b\mathcal{B}(\underline{y}_t) \tag{102.4}$$

A $\mathcal{B}$ Polynomial with coefficients $\alpha_{[0,p]}$:

$$\alpha(\mathcal{B}) = \alpha_0 + \alpha_1\mathcal{B} + \alpha_2\mathcal{B}^2 + \ldots \alpha_p\mathcal{B}^p \tag{102.5}$$

$\mathcal{B}^{-1}$ is the inverse of the backshift operator (a.k.a. frontshift operator)

$$\mathcal{B}^{-1}\underline{x}_t = \underline{x}_{t+1} \tag{102.6}$$

The following two Taylor expansions prove useful in finding the inverse of backshift operator polynomials:

- $$\frac{1}{1-z} = 1 + z + z^2 + \ldots \tag{102.7}$$

  converges for $z \in \mathbb{C}$ with $|z| < 1$. We will use this expansion with z replaced by $\alpha\mathcal{B}$, where $\alpha \in \mathbb{R}$.

- $$\frac{1}{1-z} = (-z^{-1})\left[\frac{1}{1-z^{-1}}\right] = (-z^{-1})\left[1 + z^{-1} + +z^{-2} + \ldots\right] \tag{102.8}$$

  converges for $z \in \mathbb{C}$ with $|z| > 1$. We will use this expansion with z^{-1} replaced by $(\alpha\mathcal{B})^{-1}$, where $\alpha \in \mathbb{R}$.

102.3 Metrics

Consider a t-series $\{\underline{x}_t\}_{\forall t}$.

In general, if we have a metric like Auto-covariance (ACov) that is defined for $\tau = 1, 2, 3, \ldots$, it is conventional in time series analysis to refer to the plot of that metric for all values of τ as the Auto-covariance *Function* (ACovF).

- **Expected value and Variance**

$$E[\underline{x}_t] \tag{102.9}$$

$$Var[\underline{x}_t] = \langle \underline{x}_t, \underline{x}_t \rangle \tag{102.10}$$

- **Auto-covariance (ACov)**

$$\gamma_{t,t+\tau} = \langle \underline{x}_t, \underline{x}_{t+\tau} \rangle \tag{102.11}$$

- **Auto-correlation (ACorr)** (assumes w-stationarity)

$$\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)} \tag{102.12}$$

- **Generating function of auto-covariance** (assumes w-stationarity)

$$\widetilde{\gamma}(z) = \sum_{\tau=-\infty}^{\infty} \gamma(\tau) z^{\tau} \tag{102.13}$$

  Note that this transform is double sided. Fourier Transform if $z = e^{-i\omega\tau}$.

- **Expected value and variance conditioned on all past information**

  For $\tau = 1, 2, 3, \ldots,$

$$E_{|x_{\leq t}}[\underline{x}_{t+\tau}] \tag{102.14}$$

$$Var_{|x_{\leq t}}[\underline{x}_{t+\tau}] = \langle \underline{x}_{t+\tau}, \underline{x}_{t+\tau} \rangle_{|x_{\leq t}} \tag{102.15}$$

- **Partial auto-covariance (PACov)**

  Assume w-stationarity. For $\tau = 1, 2, 3, \ldots$

$$\gamma^{part}(\tau) = \langle \underline{x}_t, \underline{x}_{t+\tau} \rangle_{|x_{\leq t}, \underline{x}_{t+\tau}} \tag{102.16}$$

  The idea is that we set to zero the nodes $\underline{x}_{(t,t+\tau)} = \{\underline{x}_{t+1}, \underline{x}_{t+2}, \ldots, \underline{x}_{t+\tau-1}\}$ that lie between (but not including) $\underline{x}_t$ and $\underline{x}_{t+\tau}$.

- **Partial auto-correlation (PACorr)**

$$\rho^{part}(\tau) = \frac{\gamma^{part}(\tau)}{\gamma^{part}(0)} \tag{102.17}$$

---

weak stationarity (w-stationarity) means that $E[\underline{x}_t] = \mu$ and $\gamma_{t,t+\tau} = \gamma(\tau)$ are both independent of t. If we have w-stationarity, then

$$\begin{aligned} \gamma(-\tau) &= \langle \underline{x}_t, \underline{x}_{t-\tau} \rangle && (102.18) \\ &= \langle \underline{x}_{t-\tau}, \underline{x}_t \rangle && (102.19) \\ &= \gamma(\tau) && (102.20) \end{aligned}$$

We will often abbreviate $\gamma(\tau)$ by γ_τ.

Example of various metrics. If $\{\underline{n}_t\}_{\forall t} \sim WN(0, \sigma^2)$ then

$$E[\underline{n}_t] = 0 \tag{102.21a}$$

$$\gamma(\tau) = \sigma^2 \delta(\tau, 0) \tag{102.21b}$$

$$\gamma(0) = \sigma^2 \tag{102.21c}$$

$$\widetilde{\gamma}(z) = \gamma(0) \tag{102.21d}$$

For $\tau > 0$,

$$E_{|n_{\leq t}}[\underline{n}_{t+\tau}] = E[\underline{n}_{t+\tau}] = 0 \tag{102.21e}$$

$$\langle \underline{n}_{t+\tau}, \underline{n}_{t+\tau} \rangle_{|n_{\leq t}} = E[\underline{n}_{t+\tau}^2] = \sigma^2 \tag{102.21f}$$

102.4 Definition of $ARMA(p, q)$, $AR(p)$ and $MA(q)$.

Suppose $\{\underline{y}_t\}_{\forall t}$ is a zero mean t-series. Hence $\underline{y}_t = \underline{Y}_t - \mu$, $E[\underline{Y}_t] = \mu$, $E[\underline{y}_t] = 0$. $\underline{y}_t$ is said to be the **demeaned** version of Y_t.

Suppose also that $\{\underline{n}_t\}_{\forall t} \sim WN(0, \sigma^2)$.

Then we define the **Auto-Regressive Moving-Average t-series** $ARMA(p, q)$ by

$$\underline{y}_t = \underbrace{\sum_{j=1}^{p} \alpha_j \underline{y}_{t-j}}_{\mathcal{Y}_t^{AR(p)}} + \underline{n}_t + \underbrace{\sum_{j=1}^{q} \nu_j \underline{n}_{t-j}}_{\mathcal{Y}_t^{MA(q)}} \tag{102.22}$$

(α stands for the first letter of "auto-regressive". ν stands for first letter of "noise".)

Special cases

1. **Auto-Regressive t-series** $AR(p)$

$$\underline{y}_t = \mathcal{Y}_t^{AR(p)} + \underline{n}_t \tag{102.23}$$

2. **Moving-Average t-series** $MA(q)$

$$\underline{y}_t = \underline{n}_t + \mathcal{Y}_t^{MA(q)} \tag{102.24}$$

Figure 102.1: $ARMA(2,3)$, $AR(2)$ and $MA(3)$ bnets. For clarity, we show only the arrows entering node $\underline{y}_t$. The full bnet has the same structural pattern of incoming arrows (including the weights α_j, ν_j) for each node $\underline{y}_{t'}$ for all t'.

Fig.102.1 shows the bnets for $ARMA(2,3)$, $AR(2)$ and $MA(3)$. The TPM, printed in blue, for node $\underline{y}_t$ in those bnets, is as follows:

For $ARMA(p,q)$,

$$P(y_t|y_{[t-p,t-1]}, n_{[t-q,t]}) = \mathbb{1}(y_t = \text{see Eq.102.22}) \tag{102.25}$$

For $AR(p)$,

$$P(y_t|y_{[t-p,t-1]}, n_t) = \mathbb{1}(y_t = \text{see Eq.102.23}) \tag{102.26}$$

For $MA(q)$,

$$P(y_t|n_{[t-q,t]}) = \mathbb{1}(y_t = \text{see Eq.102.24}) \tag{102.27}$$

The $\underline{n}_t$ variable is variously referred to as the **external noise, impulse, shock, innovation** at time t.

102.5 Solving $AR(p)$

Suppose $\{\underline{y}_t\}_{\forall t}$ is an $AR(p)$ t-series. Hence

$$\underline{y}_t = \sum_{j=1}^{p} \alpha_j y_{t-j} + \underline{n}_t \tag{102.28}$$

$AR(0)$ satisfies:

$$\underline{y}_t = \underline{n}_t \tag{102.29}$$

See Fig.102.2. This is just white noise.

$AR(1)$ satisfies:

$$\underline{y}_t = \alpha_1 \underline{y}_{t-1} + \underline{n}_t \tag{102.30}$$

See Fig102.2. This is a Markov chain with external i.i.d. noise injected to each node. $AR(1)$ is the discrete form of the so called **Ornstein-Uhlenbeck t-series** (a.k.a. as the **Langevin Equation**). When $\alpha_1 = 1$, it is called a **random walk**.

$$\begin{tikzcd}
AR(0) & \cdots \underline{n}_{t-2} \arrow[d, "1"'] & \underline{n}_{t-1} \arrow[d, "1"'] & \underline{n}_t \arrow[d, "1"'] & \underline{n}_{t+1} \cdots \arrow[d, "1"'] \\
 & \cdots \underline{y}_{t-2} & \underline{y}_{t-1} & \underline{y}_t & \underline{y}_{t+1} \cdots \\
AR(1) & \cdots \underline{n}_{t-2} \arrow[d, "1"'] & \underline{n}_{t-1} \arrow[d, "1"'] & \underline{n}_t \arrow[d, "1"'] & \underline{n}_{t+1} \cdots \arrow[d, "1"'] \\
 & \cdots \underline{y}_{t-2} \arrow[r, "\alpha_1"'] & \underline{y}_{t-1} \arrow[r, "\alpha_1"'] & \underline{y}_t \arrow[r, "\alpha_1"'] & \underline{y}_{t+1} \cdots
\end{tikzcd}$$

Figure 102.2: Bnets for $AR(0)$ and $AR(1)$.

Note that

$$\alpha^-(\mathcal{B})\underline{y}_t = \underline{n}_t \tag{102.31}$$

where

$$\alpha^-(\mathcal{B}) = 1 - \sum_{j=1}^{p} \alpha_j \mathcal{B}^j \tag{102.32}$$

Note that we can get $AR(p)$ from $AR(\infty)$ by setting $\alpha_{>p} = 0$.

If $\alpha^-(\beta)$ is invertible, then, using the Taylor expansion Eq.(102.7), we get

$$\underline{y}_t = \alpha'(\mathcal{B})\underline{n}_t \tag{102.33}$$

where

$$\alpha'(\mathcal{B}) = \frac{1}{\alpha^-(\mathcal{B})} = 1 + \sum_{k=1}^{\infty}\left[\sum_{j=1}^{p}\alpha_j\mathcal{B}^j\right]^k = \sum_{j=0}^{\infty}\alpha'_j\mathcal{B}^j \tag{102.34}$$

where $\alpha'_0 = 1$.

102.6 Solving $MA(q)$

Suppose $\{\underline{y}_t\}_{\forall t}$ is an $MA(q)$ t-series. Hence,

$$\underline{y}_t = \underline{n}_t + \sum_{j=1}^{q}\nu_j\underline{n}_{t-j} = \sum_{j=0}^{q}\nu_j\underline{n}_{t-j} \tag{102.35}$$

where $\nu_0 = 1$. Thus,

$$\underline{y}_t = \nu(\mathcal{B})\underline{n}_t \tag{102.36}$$

where

$$\nu(\mathcal{B}) = 1 + \sum_{j=1}^{q}\nu_j\mathcal{B}^j = \sum_{j=0}^{q}\nu_j\mathcal{B}^j\ . \tag{102.37}$$

Note that we can get $MA(q)$ from $MA(\infty)$ by setting $\nu_{>q} = 0$.

Claim 178 *If $\{\underline{y}_t\}_{\forall t}$ is an $MA(q)$ t-series, then*

$$E[\underline{y}_t] = 0 \tag{102.38a}$$

For $\tau \geq 0$,

$$\gamma(\tau) = \mathbb{1}(\tau \leq q)\sigma^2\sum_{j=0}^{q-\tau}\nu_j\nu_{\tau+j} \tag{102.38b}$$

$$\gamma(0) = \sigma^2\sum_{j=0}^{q}\nu_j^2 \tag{102.38c}$$

proof:

$$\gamma(\tau) = \left\langle \underline{y}_t, \underline{y}_{t+\tau} \right\rangle \tag{102.39}$$

$$= \left\langle \sum_{j=0}^{q} \nu_j \underline{n}_{t-j}, \sum_{k=0}^{q} \nu_k \underline{n}_{t+\tau-k} \right\rangle \tag{102.40}$$

$$= \sigma^2 \sum_{j=0}^{q} \sum_{k=0}^{q} \nu_j \nu_k \delta(t-j, t+\tau-k) \tag{102.41}$$

$$= \sigma^2 \sum_{j=0}^{q} \sum_{k=0}^{q} \nu_j \nu_k \delta(k, \tau+j) \tag{102.42}$$

$$= \mathbb{1}(\tau \leq q) \sigma^2 \sum_{j=0}^{q-\tau} \nu_j \nu_{\tau+j} \tag{102.43}$$

QED

102.7 Solving $ARMA(p,q)$

Suppose $\{\underline{y}_t\}_{\forall t}$ is an $ARMA(p,q)$ t-series. Hence, using Eqs.(102.31) and (102.36), $\underline{y}_t$ satisfies

$$\alpha^-(\mathcal{B})\underline{y}_t = \nu(\mathcal{B})\underline{n}_t \tag{102.44}$$

If $\alpha^-(\beta)$ is invertible, then, using the Taylor expansion Eq.(102.7), we get

$$\underline{y}_t = \frac{\nu(\mathcal{B})}{\alpha^-(\mathcal{B})}\underline{n}_t = \nu(\mathcal{B})\alpha'(\mathcal{B})\underline{n}_t \tag{102.45}$$

We see that, if the $\alpha^-(\mathcal{B})$ operator is invertible, an $AR(p)$ or an $ARMA(p,q)$ t-series can be represented as an $MA(\infty)$ t-series. $MA(\infty)$ is often called **Wold's Decomposition**. Furthermore, if the $\nu(\mathcal{B})$ operator is invertible, an $MA(q)$ or an $ARMA(p,q)$ t-series can be represented as an $AR(\infty)$ t-series.

The polynomials $\alpha^-(z)$ and $\nu(z)$ can be expressed in factored form $\alpha^-(z) = \prod_{j=1}^{p}(z - z_j^{\alpha})$ and $\nu(z) = \prod_{j=1}^{q}(z - z_j^{\nu})$. If these two polynomials have a root z_0 in common, both polynomials should be divided by $(z - z_0)$. This reduces an $ARMA(p,q)$ t-series to an $ARMA(p-1,q-1)$ t-series. The bnet for $ARMA(p-1,q-1)$ has one less α_j arrow and one less ν_j arrow than the bnet for $AR(p,q)$.

102.8 Auto-correlation and partial auto-correlation

Note from Eq.(102.38b) that if $\{\underline{y}_t\}_{\forall t}$ is a $MA(q)$ t-series, then $\gamma(\tau) = 0$ for all $\tau > q$. Fig.102.3 gives a graphical proof, using bnets and the d-separation theorem, that for a $MA(2)$ t-series, $\gamma(\tau) = 0$ for $\tau > 2$. As a consequence of this, a plot of $\gamma(\tau)$ versus τ for a typical $MA(2)$ t-series looks like Fig.102.4.

Figure 102.3: $MA(2)$ bnet. For clarity, we show only arrows entering nodes $\underline{y}_t$ and $\underline{y}_{t+\tau}$ for $\tau = 1, 2, 3$. For $\tau = 1, 2$, there is a path through which information can flow from node $\underline{y}_t$ to node $\underline{y}_{t+\tau}$. For $\tau = 2$, there is no such path.

Figure 102.4: Plot of auto-correlation function (ACorrF) $\rho(\tau)$ and partial auto-correlation function (PACorrF) $\rho^{part}(\tau)$ for an instance of $MA(2)$. Note that $\rho(\tau)$ vanishes for $\tau > 2$.

Claim 179 *If $\{\underline{y}_t\}_{\forall t}$ is an $AR(p)$ t-series, then $\gamma^{part}(\tau) = 0$ for all $\tau > p$.*

proof:

Let

$$\underline{\xi} = \underline{y}_{\leq t}, \underline{y}_{t+\tau} \tag{102.46}$$

Recall that

$$\gamma^{part}(\tau) = \left\langle \underline{y}_t, \underline{y}_{t+\tau} \right\rangle_{|\xi} \tag{102.47}$$

$$= \left\langle \underline{y}_t \; \underline{y}_{t+\tau} \right\rangle_{|\xi} - \left\langle \underline{y}_t \right\rangle_{|\xi} \left\langle \underline{y}_{t+\tau} \right\rangle_{|\xi} \tag{102.48}$$

$$= \left\langle \underline{y}_t \; \underline{y}_{t+\tau} \right\rangle_{|\xi} \tag{102.49}$$

Define

$$Z_{(t,t+\tau)} = \mathbb{1}(\underline{y}_{(t,t+\tau)} = 0) \tag{102.50}$$

Hence, the operator $Z_{(t,t+\tau)}$ sets all $\underline{y}_{t'}$ with $t < t' < t + \tau$ equal to zero. Note that we can express the PACov as

$$\gamma^{part}(\tau) = \left\langle Z_{(t,t+\tau)} \; (\underline{y}_t \; \underline{y}_{t+\tau}) \right\rangle \; . \tag{102.51}$$

Since

$$\underline{y}_{t+\tau} = \sum_{j=1}^{p} \alpha_j \underline{y}_{t+\tau-j} + \underline{n}_{t+\tau} \tag{102.52}$$

$$= \alpha_p \underline{y}_{t+\tau-p} + \ldots + \alpha_2 \underline{y}_{t+\tau-2} + \alpha_1 \underline{y}_{t+\tau-1} + \underline{n}_{t+\tau} \; , \tag{102.53}$$

we get

$$Z_{(t,t+\tau)} \underline{y}_{t+\tau} = \underline{n}_{t+\tau} \quad \text{if } \tau > p \; . \tag{102.54}$$

Hence, for $\tau > p$,

$$\gamma^{part}(\tau) = \left\langle \underline{y}_t \; \underline{n}_{t+\tau} \right\rangle = 0 \tag{102.55}$$

QED

Fig.102.5 gives a graphical proof, using bnets and the d-separation theorem, that for an $AR(2)$ t-series, $\gamma^{part}(\tau) = 0$ for $\tau > 2$. As a consequence of this, a plot of $\gamma^{part}(\tau)$ versus τ for a typical $AR(2)$ t-series looks like Fig.102.6.

	ACorr	PACorr
$AR(p)$	tapers off	jumps to zero for $\tau > p$
$MA(q)$	jumps to zero for $\tau > q$	tapers off

Table 102.1: Detecting $AR(p)$ and $MA(q)$ using auto-correlation and partial auto-correlation.

Figure 102.5: $AR(2)$ bnet. For clarity, we show only arrows entering nodes $\underline{y}_t$ and $\underline{y}_{t+\tau}$ for $\tau = 1, 2, 3$. Yellow nodes are in set $\underline{y}_{(t,t+\tau)}$. They are conditioned on, so information can't flow through them to node $\underline{y}_{t+\tau}$ by the d-separation theorem.

Figure 102.6: Plot of auto-correlation function (ACorrF) $\rho(\tau)$ and partial auto-correlation function (PACorrF) $\rho^{part}(\tau)$ for an instance of $AR(2)$. Note that $\rho^{part}(\tau)$ vanishes for $\tau > 2$.

102.9 Generating function of auto-correlation

Claim 180 *If*

$$\alpha(z) = \sum_{\tau=-\infty}^{\infty} \alpha_\tau z^\tau \tag{102.56}$$

then

$$\alpha(z)\alpha(z^{-1}) = \sum_{\tau=-\infty}^{\infty} z^\tau \sum_{j=-\infty}^{\infty} \alpha_j \alpha_{j+\tau} \tag{102.57}$$

proof:

$$\begin{aligned}
\alpha(z)\alpha(z^{-1}) &= \sum_{j'=-\infty}^{\infty} z^{j'} \alpha_{j'} \sum_{j=-\infty}^{\infty} z^{-j} \alpha_j && (102.58)\\
&= \sum_{\tau=-\infty}^{\infty} \sum_{j'=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} z^\tau \alpha_{j'} \alpha_j \delta(j'-j,\tau) && (102.59)\\
&= \sum_{\tau=-\infty}^{\infty} z^\tau \sum_{j=-\infty}^{\infty} \alpha_j \alpha_{j+\tau} && (102.60)
\end{aligned}$$

QED

As an example of this claim, note that

$$(\alpha_0 + \alpha_1 z)(\alpha_0 + \alpha_1 z^{-1}) = \alpha_0\alpha_1 z^{-1} + (\alpha_0^2 + \alpha_1^2) + \alpha_0\alpha_1 z \tag{102.61}$$

Claim 181 *If* $\{\underline{y}_t\}_{\forall t}$ *is an* $AR(p)$ *t-series, then*

$$\tilde{\gamma}(z) = \alpha'(z)\sigma^2\alpha'(z^{-1}) \tag{102.62}$$

proof: Left to reader. See Claim 180.
QED

Claim 182 *If* $\{\underline{y}_t\}_{\forall t}$ *is an* $MA(q)$ *t-series, then*

$$\tilde{\gamma}(z) = \nu(z)\sigma^2\nu(z^{-1}) \tag{102.63}$$

proof:

$$\underline{y}_t = \sum_{j=-\infty}^{\infty} \nu_j \underline{n}_{t-j} \tag{102.64}$$

$$\begin{aligned}
\tilde{\gamma}(z) &= \sum_{\tau=-\infty}^{\infty} \left\langle \underline{y}_0, \underline{y}_\tau \right\rangle z^\tau && (102.65)\\
&= \sum_{\tau=-\infty}^{\infty} z^\tau \sum_{j=-\infty}^{\infty} \sum_{j'=-\infty}^{\infty} \nu_j \nu_{j'} \underbrace{\left\langle \underline{n}_{-j}, \underline{n}_{\tau-j'} \right\rangle}_{\sigma^2 \delta(j', \tau+j)} && (102.66)\\
&= \sigma^2 \sum_{\tau=-\infty}^{\infty} z^\tau \sum_{j=-\infty}^{\infty} \nu_j \nu_{j+\tau} && (102.67)
\end{aligned}$$

Now use Claim 180.
QED

Claim 183 *If $\{\underline{y}_t\}_{\forall t}$ is an $ARMA(p,q)$ t-series, then*

$$\tilde{\gamma}(z) = \alpha'(z)\nu(z)\sigma^2\nu(z^{-1})\alpha'(z^{-1}) \tag{102.68}$$

proof: Left to reader. See Claim 180.
QED

102.10 Impulse Response

The derivatives

$$IR_\tau = \frac{\partial y_{t+\tau}}{\partial n_t} \tag{102.69}$$

are called **impulse responses** or **dynamical multipliers**. Note that this derivative depends on τ but not t by w-stationarity. A plot of IR_τ versus τ is called the **Impulse Response Function (IRF)**. Examples:

- **Claim 184** *For $MA(q)$,*

  $$\frac{\partial y_{t+\tau}}{\partial n_t} = \nu_\tau \mathbb{1}(0 \le \tau \le q) \tag{102.70}$$

  where $\nu_0 = 1$. (See Fig.102.7)

  proof:

  $$\underline{y}_{t+\tau} = \sum_{j=0}^{q} \nu_j \underline{n}_{t+\tau-j} \tag{102.71}$$

  so Eq.(102.70) follows.
  QED

- **Claim 185** *For $AR(1)$,*

$$\frac{\partial y_{t+\tau}}{\partial n_t} = (\alpha_1)^\tau \mathbb{1}(\tau \geq 0). \tag{102.72}$$

(See Fig.102.8)

proof:

$$(1 - \alpha_1 \mathcal{B})\underline{y}_{t+\tau} = \underline{n}_{t+\tau} \tag{102.73}$$

Therefore, using the Taylor expansion Eq.(102.7),

$$\underline{y}_{t+\tau} = 1 + \sum_{j=1}^{\infty} \alpha_1^j \mathcal{B}^j \underline{n}_{t+\tau} \tag{102.74}$$

$$= 1 + \sum_{j=1}^{\infty} \alpha_1^j \underline{n}_{t+\tau-j} \tag{102.75}$$

so Eq.(102.72) follows.
QED

In general, IR_τ equals a sum over paths from $\underline{n}_t$ to $\underline{y}_{t+\tau}$. Each path contributes the product of the weights of all the arrows in the path.

$$\begin{tikzcd}
\cdots \underline{n}_t \arrow[drr, "\nu_2"'] & \underline{n}_{t+1} & \underline{n}_{t+2} \cdots \\
\cdots \underline{y}_t & \underline{y}_{t+1} & \underline{y}_{t+2} \cdots
\end{tikzcd}$$

Figure 102.7: Pictorial representation of impulse response $IR_2 = \nu_2$ for a $MA(q)$ t-series with $q \geq 2$.

$$\begin{tikzcd}
\cdots \underline{n}_t \arrow[d, "1"'] & \underline{n}_{t+1} & \underline{n}_{t+2} \cdots \\
\cdots \underline{y}_t \arrow[r, "\alpha_1"'] & \underline{y}_{t+1} \arrow[r, "\alpha_1"'] & \underline{y}_{t+2} \cdots
\end{tikzcd}$$

Figure 102.8: Pictorial representation of impulse response $IR_2 = \alpha_1^2$ for an $AR(1)$ t-series.

102.11 $AR(p)$ and Yule-Walker equations

Claim 186 *(Yule-Walker equations) If $\{\underline{y}_t\}_{\forall t}$ is an $AR(p)$ t-series,*

$$\gamma_\tau = \sum_{j=1}^{p} \gamma_{\tau-j}\alpha_j + \sigma^2\delta(\tau,0) \tag{102.76}$$

for all $\tau \in \mathbb{Z}$, where we are abbreviating $\gamma(j) = \gamma_j$ for all j.

proof:

Recall that

$$\underline{y}_\tau = \sum_{j=1}^{p} \underline{y}_{\tau-j}\alpha_j + \underline{n}_\tau \tag{102.77}$$

Therefore

$$\underbrace{\left\langle \underline{y}_0, \underline{y}_\tau \right\rangle}_{\gamma_\tau} = \sum_{j=1}^{p} \underbrace{\left\langle \underline{y}_0, \underline{y}_{\tau-j} \right\rangle}_{\gamma_{\tau-j}} \alpha_j + \underbrace{\left\langle \underline{y}_0, \underline{n}_\tau \right\rangle}_{\sigma^2\delta(\tau,0)} \tag{102.78}$$

QED

Figure 102.9: Bnet representing Yule-Walker (Christmas Walker) Eqs.(102.76) with $p = 3$. For clarity, we show only arrows entering node $\underline{\gamma}_\tau$. The full bnet has the same structural pattern of incoming arrows (including the α_j) for each node $\gamma_{\tau'}$ for all τ'.

Fig.102.9 shows a bnet for the Yule-Walker Eqs.(102.76. The TPMs, printed in blue, for the τ time-slice of that bnet, are as follows:

$$P(\gamma_\tau) = \mathbb{1}\left(\gamma_\tau = \text{ See Eq.(102.76)}\right) \tag{102.79}$$

$$P(c_\tau) = \mathbb{1}(c_\tau = \sigma^2\delta(\tau,0)) \tag{102.80}$$

The Yule-Walker Eqs.(102.76) imply that

$$\underbrace{\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_p \end{bmatrix}}_{\gamma^p} = \underbrace{\begin{bmatrix} \gamma_0 & \gamma_{-1} & \gamma_{-2} & \cdots & \gamma_{1-p} \\ \gamma_1 & \gamma_0 & \gamma_{-1} & \cdots & \gamma_{2-p} \\ \vdots & & & & \\ \gamma_{p-1} & \gamma_{p-2} & \gamma_{p-3} & \cdots & \gamma_0 \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix}}_{\alpha^p} \tag{102.81}$$

Hence

$$\gamma^p = \Gamma \alpha^p \ . \tag{102.82}$$

If Γ is invertible,

$$\alpha^p = \Gamma^{-1} \gamma^p \ . \tag{102.83}$$

Once we know α^p, we can solve for σ^2

$$\gamma_0 = \sum_{j=1}^{p} \gamma_{-j} \alpha_j + \sigma^2 \ , \tag{102.84}$$

i.e., Eq.(102.76) for $\tau = 0$.

102.12 Forecasting

Before submerging the reader in the messy details of t-series forecasting, and running the risk of quickly losing him/her, let me explain to the reader that what we are about to do in this section, is, in the final analysis, quite trivial, and boils down to solving simple systems of linear equations. That is all there is to it!! How hard can that be? In reading this section, don't lose sight of that fact and you'll be okay.

Suppose $\{\underline{y}_t\}_{\forall t}$ is an $AR(\infty)$ t-series. For $\tau > 0$, the "orthogonal projection" $\underline{y}_{t+\tau}|\underline{y}_{\leq t}$ is defined as the linear combination of $\underline{y}_{\leq t}$ obtained by doing Linear Regression with x-variables $\underline{y}_{\leq t}$ and y-variable $\underline{y}_{t+\tau}$. Thus

$$E\left[\underline{y}_{t+\tau}|\underline{y}_{\leq t}\right] = E_{|\underline{y}_{\leq t}}[\underline{y}_{t+\tau}] \tag{102.85}$$

and

$$\begin{aligned} Var\left[\underline{y}_{t+\tau}|\underline{y}_{\leq t}\right] &= E\left[\left(\underline{y}_{t+\tau}|\underline{y}_{\leq t} - E[\underline{y}_{t+\tau}|\underline{y}_{\leq t}]\right)^2\right] & (102.86) \\ &= Var_{|\underline{y}_{\leq t}}[\underline{y}_{t+\tau}] \ . & (102.87) \end{aligned}$$

For example, for $\tau = 1$,

$$\underline{y}_{t+1} = \sum_{j=1}^{p} \alpha_j \underline{y}_{t+1-j} + \underline{n}_{t+1} \tag{102.88}$$

$$\underline{y}_{t+1}|\underline{y}_{\leq t} = \sum_{j=1}^{p} \alpha_j \underline{y}_{t+1-j} \tag{102.89}$$

$$\underline{y}_{t+1} - \underline{y}_{t+1|\underline{y}\leq t} = \underline{n}_{t+1} \tag{102.90}$$

$$\begin{aligned} E\left[\underline{y}_{t+1}|\underline{y}_{\leq t}\right] &= E_{|\underline{y}_{\leq t}}[\underline{y}_{t+1}] && (102.91) \\ &= 0 && (102.92) \end{aligned}$$

$$\begin{aligned} Var\left[\underline{y}_{t+1}|\underline{y}_{\leq t}\right] &= Var_{|\underline{y}_{\leq t}}[\underline{y}_{t+1}] && (102.93) \\ &= \sigma^2 && (102.94) \end{aligned}$$

If $\{\underline{y}_t\}_{\forall t}$ is an $AR(\infty)$ t-series and $\tau > 0$, by **forecasting** $\underline{y}_{t+\tau}|\underline{y}_{\leq t}$, we will mean finding $\underline{y}_{t+\tau}|\underline{y}_{\leq t}$, given $y_{\leq t}$ as input data or some other input data from which $y_{\leq t}$ can be derived. Fig.102.10 illustrates 3 types of input data that we will consider next.

Note that if $\{\underline{y}_t\}_{\forall t}$ is an $AR(\infty)$ t-series, then $\alpha^{-}(\mathcal{B})\underline{y}_t = n_t$. Thus, $\underline{n}_{\leq t}$ can be expressed in terms of $\underline{y}_{\leq t}$, yielding a function $\underline{n}_{\leq t}(\underline{y}_{\leq t})$. If the operator $\alpha^{-}(\mathcal{B})$ is invertible, the opposite is also true: $\underline{y}_{\leq t}$ can be expressed in terms of $\underline{n}_{\leq t}$, yielding a function $\underline{y}_{\leq t}(\underline{n}_{\leq t})$. Thus,

$$\underline{y}_{t+\tau}|\underline{y}_{\leq t} = \underline{y}_{t+\tau}|\underline{y}_{\leq t}(\underline{n}_{\leq t}) = \underline{y}_{t+\tau}|\underline{n}_{\leq t} \tag{102.95}$$

We've seen that if we assume invertibility of $\alpha^{-}(\mathcal{B})$, then, in order to forecast $\underline{y}_{t+\tau}|\underline{y}_{\leq t}$, we have the option of considering $\{\underline{y}_t\}_{\forall t}$ to be either an $AR(\infty)$ or an $MA(\infty)$ t-series. We will assume the latter, because this simplifies the algebra necessary to calculate both $\underline{y}_{t+\tau}|\underline{y}_{\leq t}$ and its variance.

(a) **find** $\underline{y}_{t+\tau}|y_{\leq t}$ **given** $\underline{y}_{\leq t}$

Suppose $\{\underline{y}_t\}_{\forall t}$ is an $MA(\infty)$ t-series. Then

$$\begin{aligned} \underline{y}_{t+\tau} &= \sum_{j=0}^{\infty} \nu_j \underline{n}_{t+\tau-j} && (102.96) \\ &= \underbrace{\sum_{j=0}^{\tau-1} \nu_j \underline{n}_{t+\tau-j}}_{\underline{y}_{t+\tau}-\underline{y}_{t+\tau}|y_{\leq t}} + \underbrace{\sum_{j=\tau}^{\infty} \nu_j \underline{n}_{t+\tau-j}}_{\underline{y}_{t+\tau}|y_{\leq t}} && (102.97) \end{aligned}$$

Figure 102.10: Bnet for $MA(\infty)$ for forecasting $\underline{y}_{t+\tau}|\underline{y}_{\leq t}$. Input data in red. For clarity, we show only the arrows entering node $\underline{y}_t$.

$$\underline{y}_{t+\tau}|y_{\leq t} = \sum_{j=\tau}^{\infty} \nu_j \underline{n}_{t+\tau-j} \tag{102.98}$$

$$E_{|y_{\leq t}}[\underline{y}_{t+\tau}] = E[\underline{y}_{t+\tau}|y_{\leq t}] = 0^1 \tag{102.99}$$

$$\begin{aligned} \left\langle \underline{y}_{t+\tau}\underline{y}_{t+\tau} \right\rangle_{|y_{\leq t}} &= \left\langle \underline{y}_{t+\tau} - \underline{y}_{t+\tau}|y_{\leq t},\ \underline{y}_{t+\tau} - \underline{y}_{t+\tau}|y_{\leq t} \right\rangle & (102.100) \\ &= \sum_{j=0}^{\tau-1} \nu_j^2 & (102.101) \end{aligned}$$

[1]Remember that $\underline{y}_t$ has zero mean by definition. $Y_t = y_t + \mu$ for some time independent constant μ.

Figure 102.11: $MA(2)$ bnet. For clarity, we show only arrows entering node $\underline{y}_{t+\tau}$ for $\tau = 1, 2, 3$. Green arrows originate at a time after time t.

For $MA(q)$, because $\nu_j = 0$ for $j > q$, this reduces to

$$\underline{y}_{t+\tau}|y_{\leq t} = \mathbb{1}(\tau \leq q) \sum_{j=\tau}^{q} \nu_j \underline{n}_{t+\tau-j} \tag{102.102}$$

$$E_{|y_{\leq t}}[\underline{y}_{t+\tau}] = E[\underline{y}_{t+\tau}|y_{\leq t}] = 0 \tag{102.103}$$

$$\left\langle \underline{y}_{t+\tau}, \underline{y}_{t+\tau} \right\rangle_{|y_{\leq t}} = \sum_{j=0}^{\min(\tau-1,q)} \nu_j^2 \tag{102.104}$$

Eq.(102.104) is motivated by Fig.102.11). Only the green arrows entering $\underline{y}_{t+\tau}$ and originating in a node $\underline{n}_{t'}$ with $t' > t$ contribute to the right hand side of Eq.(102.104).

Thus, the mean of $\underline{y}_{t+\tau}$ can be predicted to remain zero for all $\tau > 0$ (forever). The error in that prediction increases with τ until τ reaches q. After that, the error remains constant.

Note that

$$\underline{y}_{t+\tau}|y_{\leq t} = \sum_{j=\tau}^{\infty} \nu_j \underline{n}_{t+\tau-j} \tag{102.105}$$

$$= \sum_{j=\tau}^{\infty} \nu_j \mathcal{B}^{j-\tau} \underline{n}_t \tag{102.106}$$

$$= \left[\sum_{j=0}^{\infty} \nu_j \mathcal{B}^{j-\tau}\right]_{\mathcal{B}^{\geq 0}} \underline{n}_t \tag{102.107}$$

$$= \left[\mathcal{B}^{-\tau}\nu(\mathcal{B})\right]_{\mathcal{B}^{\geq 0}} \underline{n}_t \tag{102.108}$$

$\mathcal{B}^{\geq 0}$ means only the non-negative powers of $\mathcal{B}$ are kept. Eq.(102.108) is just Eq.(102.98) in fancy notation.

In $MA(\infty)$, if $\nu(\mathcal{B})$ is invertible, then

$$\underline{n}_t = \nu(\mathcal{B})^{-1}\underline{y}_t \tag{102.109}$$

so

$$\boxed{\underline{y}_{t+\tau}|y_{\leq t} = \mathcal{L}_{WK}\underline{y}_t} \tag{102.110}$$

where

$$\mathcal{L}_{WK} = \left[\mathcal{B}^{-\tau}\nu(\mathcal{B})\right]_{\mathcal{B}^{\geq 0}} \nu(\mathcal{B})^{-1} \tag{102.111}$$

Eq.(102.110) is called the **Wiener-Kolgomorov prediction formula (WKPF)**. Next, we apply WKPF to $AR(1)$ and $MA(1)$ as examples of its usage.

- $AR(1)$

$$\underbrace{(1-\alpha_1\mathcal{B})}_{\nu(\mathcal{B})^{-1}}\underline{y}_t = \underline{n}_t \tag{102.112}$$

$$\underline{y}_t = \underbrace{\left[\sum_{j=0}^{\infty}(\alpha_1\mathcal{B})^j\right]}_{\nu(\mathcal{B})}\underline{n}_t \tag{102.113}$$

$$\nu_j = (\alpha_1)^j \tag{102.114}$$

$$\left[\mathcal{B}^{-\tau}\nu(\mathcal{B})\right]_{\mathcal{B}^{\geq 0}} = (\alpha_1)^\tau \nu(\mathcal{B}) \tag{102.115}$$

$$\underline{y}_{t+\tau}|y_{\leq t} = (\alpha_1)^\tau \underline{y}_t \tag{102.116}$$

Hence, $\underline{y}_{t+\tau}|y_{\leq t}$ decreases geometrically as τ grows.

- $MA(1)$

$$\underline{y}_t = \underbrace{(1+\nu_1\mathcal{B})}_{\nu(\mathcal{B})}\underline{n}_t \tag{102.117}$$

$$\left[\mathcal{B}^{-\tau}\nu(\mathcal{B})\right]_{\mathcal{B}^{\geq 0}} = \delta(\tau,1)\nu_1 \tag{102.118}$$

$$\begin{aligned} \underline{y}_{t+\tau}|y_{\leq t} &= \delta(\tau,1)\nu_1 \underbrace{\left[\sum_{j=0}^{\infty}(-\nu_1\mathcal{B})^j\right]}_{\nu(\mathcal{B})^{-1}} \underline{y}_t && (102.119) \\ &= \delta(\tau,1)\nu_1\underline{n}_t && (102.120) \end{aligned}$$

(b) **find** $\underline{y}_{t+\tau}|y_{\leq t}$ **given** $\underline{y}_{\leq t}$**, but, for some** $m>0$**,** $\underline{y}_{\leq t-m}\approx 0$ **(approximation)**

$$\underline{y}_{t+\tau}|\underline{y}_{\leq\tau} \approx [\mathcal{L}_{WK}]_{\mathcal{B}^{<m}}\,\underline{y}_t \tag{102.121}$$

(c) **find** $\underline{y}_{t+\tau}|y_{\leq t}$ **given** $\gamma_{[0,m-1]}$ **and** $\gamma^{part}_{[\tau,\tau+m-1]}$

$$\underline{y}_{t+\tau}|\underline{y}_{\leq t} = \sum_{j=0}^{m-1}\underline{y}_{t-j}\beta_j \tag{102.122}$$

Suppose that $k\in\mathbb{Z}_{[0,m-1]}$.

Note that

$$\underline{y}_{t-k}|\underline{y}_{\leq t} = \underline{y}_{t-k}\,. \tag{102.123}$$

$$\underbrace{\left\langle \underline{y}_{t-k}|\underline{y}_{\leq t}\ ,\ \underline{y}_{t+\tau}|\underline{y}_{\leq t}\right\rangle}_{\gamma^{part}_{\tau+k}} = \sum_{j=0}^{m-1}\underbrace{\left\langle \underline{y}_{t-k}\ ,\ \underline{y}_{t-j}\right\rangle}_{\gamma_{k-j}}\beta_j \tag{102.124}$$

$$\begin{bmatrix}\gamma^{part}_{\tau}\\ \gamma^{part}_{\tau+1}\\ \vdots\\ \gamma^{part}_{\tau+m-1}\end{bmatrix} = \underbrace{\begin{bmatrix}\gamma_0 & \gamma_{-1} & \gamma_{-2} & \cdots & \gamma_{1-m}\\ \gamma_1 & \gamma_0 & \gamma_{-1} & \cdots & \gamma_{2-m}\\ \vdots & & & & \\ \gamma_{m-1} & \gamma_{m-2} & \gamma_{m-3} & \cdots & \gamma_0\end{bmatrix}}_{\Gamma}\underbrace{\begin{bmatrix}\beta_0\\ \beta_1\\ \vdots\\ \beta_{m-1}\end{bmatrix}}_{\beta^m} \tag{102.125}$$

Solve Eq.(102.125) for β^m and plug β^m into Eq.(102.122).

102.13 Model Learning

Box-Jenkins Method

1. **Filtering Data (FD)**

   Removing trends and periodicity (seasonality) via differencing, so as to achieve a w-stationary t-series. We deal with FD in Section 102.14.

2. **Model Selection**

   Finding best possible p, q for $ARMA(p, q)$ using various statistical tests and metrics.

3. **Parameter Learning (PL)**

   Finding best possible coefficients α_j and ν_j. We deal with PL in Section 102.15.

4. **Testing**

   Measuring the goodness of fit.

102.14 Differencing and $ARIMA(p, d, q)$

Let $\alpha \in (0, 1)$. We say that $\{\underline{s}_t\}_{\forall t}$ is a **Simple Exponential Smoothing (SES) t-series** if it satisfies

$$s_t = (1 - \alpha)s_{t-1} + \alpha x_t \tag{102.126}$$

$$\begin{tikzcd}
\cdots \underline{x}_{t-2} \arrow[d, "\alpha"'] & \underline{x}_{t-1} \arrow[d, "\alpha"'] & \underline{x}_{t} \arrow[d, "\alpha"'] & \underline{x}_{t+1} \arrow[d, "\alpha"'] \cdots \\
\cdots \underline{s}_{t-2} \arrow[r, "1-\alpha"] & \underline{s}_{t-1} \arrow[r, "1-\alpha"] & \underline{s}_{t} \arrow[r, "1-\alpha"] & \underline{s}_{t+1} \cdots
\end{tikzcd}$$

Figure 102.12: Bnet for Simple Exponential Smoothing t-series. $\alpha \in (0, 1)$

A SES t-series can be represented by the bnet of Fig.102.12. The TPM, printed in blue, for node $\underline{s}_t$, is as follows

$$P(s_t) = \mathbb{1}(s_t = \text{ see Eq.(102.126)}) \tag{102.127}$$

One has

$$(1 - (1 - \alpha)\mathcal{B})\underline{s}_t = \alpha \underline{x}_t \tag{102.128}$$

so

$$\underline{s}_t = \frac{\alpha}{1-(1-\alpha)\mathcal{B}}\underline{x}_t \tag{102.129}$$

$$= \alpha\sum_{j=0}^{\infty}(1-\alpha)^j\mathcal{B}^j\underline{x}_t \quad \text{(by Eq.(102.7))} \tag{102.130}$$

Note from Fig.102.12 and Eq.(102.130) that each shock $\underline{x}_{t-j}$ with $j > 0$ contributes to $\underline{s}_t$ proportionally to the product $\alpha(1-\alpha)^j$ of arrow weights in the path from $\underline{x}_{t-j}$ to $\underline{s}_t$. For example, $\underline{x}_{t-2}$ contributes to $\underline{s}_t$ in the proportion $\alpha(1-\alpha)^2$. The contributions of $\underline{x}_{t-j}$ to $\underline{s}_t$ decrease geometrically as the lag j increases.

Henceforth, we will abbreviate "**differencing**" by " **diff**".

- **First order diff operator**

$$\Delta x_t = x_t - x_{t-1} = (1-\mathcal{B})x_t \tag{102.131}$$

Seasonal first order diff:

$$\Delta_{sea}x_t = x_t - x_{t-ts} \tag{102.132}$$

where ts is length of season.

- **Second order diff operator**

$$\Delta^2 x_t = \Delta x_t - \Delta x_{t-1} \tag{102.133}$$

$$= x_t - 2x_{t-1} + x_{t-2} \tag{102.134}$$

$$= (1-2\mathcal{B}+\mathcal{B}^2)x_t \tag{102.135}$$

$$= (1-\mathcal{B})^2 x_t \tag{102.136}$$

- **d-th order diff operator**

$$\Delta^d\underline{x}_t = (1-\mathcal{B})^d\underline{x}_t \tag{102.137}$$

Recall that if $\{\underline{y}_t\}_{\forall t}$ is an $ARMA(p+d,q)$ t-series,

$$\left(1-\sum_{j=1}^{p+d}\alpha_j\mathcal{B}^j\right)\underline{y}_t = \left(1+\sum_{k=1}^{q}\nu_k\mathcal{B}^k\right)\underline{n}_t \tag{102.138}$$

We will say that $\{\underline{y}_t\}_{\forall t}$ is an **Autoregressive Integrated Moving Average** $ARIMA(p,d,q)$ **t-series** if

$$\left(1-\sum_{j=1}^{p}\alpha_j\mathcal{B}^j\right)\underbrace{\Delta^d\underline{y}_t}_{(1-\mathcal{B})^d\underline{y}_t} = \left(1+\sum_{k=1}^{q}\nu_k\mathcal{B}^k\right)\underline{n}_t \tag{102.139}$$

So an $ARIMA(p,d,q)$ t-series is a special case of an $ARMA(p+d,q)$ t-series.

Define $\mathcal{O}_{BEF}$ and $\mathcal{O}_{AFT}$ by

$$\mathcal{O}_{BEF} = 1 - \alpha_1\mathcal{B} - \alpha_2\mathcal{B}^2 - \alpha_3\mathcal{B}^3 \tag{102.140}$$

$$\begin{aligned}
\mathcal{O}_{AFT} &= \mathcal{O}_{BEF}(1-\mathcal{B}) && (102.141)\\
&= \left\{ \begin{array}{lllll} 1 & -\alpha_1\mathcal{B} & -\alpha_2\mathcal{B}^2 & -\alpha_3\mathcal{B}^3 & \\ & -\mathcal{B} & +\alpha_1\mathcal{B}^2 & +\alpha_2\mathcal{B}^3 & +\alpha_3\mathcal{B}^4 \end{array} \right. && (102.142)\\
&= 1 - \underbrace{(\alpha_1+1)}_{\alpha'_1}\mathcal{B} - \underbrace{(\alpha_2-\alpha_1)}_{\alpha'_2}\mathcal{B}^2 - \underbrace{(\alpha_3-\alpha_2)}_{\alpha'_3}\mathcal{B}^3 - \underbrace{(0-\alpha_3)}_{\alpha'_4}\mathcal{B}^4 && (102.143)
\end{aligned}$$

Figure 102.13: Effect on $\underline{y}_t$ of going from before (BEF) with a t-series $ARIMA(3,0,0)$ to after (AFT) with a t-series $ARIMA(3,1,0)$.

$\mathcal{O}_{BEF}\underline{y}_t$ and $\mathcal{O}_{AFT}\underline{y}_t$ are illustrated in Fig.102.13. Note that in going from BEF to AFT, a new arrow is introduced with weight $\alpha'_4 = -\alpha_3$. The intermediate length arrows $\underline{y}_{t-3} \to \underline{y}_t$ and $\underline{y}_{t-2} \to \underline{y}_t$ have weights $\alpha'_3 = \alpha_3 - \alpha_2$ and $\alpha'_2 = \alpha_2 - \alpha_1$ so if $\alpha_j \approx \alpha_{j-1}$, then those 2 arrows have negligible weights. Only the longest arrow $\underline{y}_{t-4} \to \underline{y}_t$ and the shortest arrow $\underline{y}_{t-1} \to \underline{y}_t$ have non-negligible weights.

Suppose

$$\underline{y}_t \approx a + bt + ct^2 + dt^3 \tag{102.144}$$

This smooth, low-order polynomial-fit to the t-series is called its **trend**.

$$\Delta\underline{y}_t \approx \Delta t \frac{d}{dt}\ \underline{y}_t = b + 2ct + 3dt^2 \tag{102.145}$$

$$\Delta^2\underline{y}_t \approx (\Delta t)^2 \frac{d^2}{dt^2}\ \underline{y}_t = 2c + 6dt \tag{102.146}$$

$$\Delta^3 \underline{y}_t \approx (\Delta t)^3 \frac{d^2}{dt^3} \ \underline{y}_t = 6d \tag{102.147}$$

Note that the mean $6d$ in $\Delta^3 \underline{y}_t$ can be adjusted to zero (demeaning). From this example, we see that every time we apply a diff operator to a time series, we reduce the order of its polynomial trend by one.

Note that if $\{\underline{y}_t\}_{\forall t}$ is an $ARIMA(0,1,1)$ t-series,

$$\Delta \underline{y}_t = (1 + \nu_1 \mathcal{B}) \underline{n}_t \tag{102.148}$$

$$\underline{y}_t = \underline{y}_{t-1} + \underline{n}_t + \nu_1 \underline{n}_{t-1} \tag{102.149}$$

Claim 187 *$ARIMA(0,1,1)$ is equivalent to a simple exponential smoothing t-series.*

proof: Setting

$$\underline{n}_t = \underline{y}_t - \widehat{\underline{y}}_t \tag{102.150}$$

in Eq.(102.148), we get

$$(1 - \mathcal{B}) \underline{y}_t = (1 - \nu_1 \mathcal{B})(\underline{y}_t - \widehat{\underline{y}}_t) \tag{102.151}$$

$$(1 - \nu_1 \mathcal{B}) \widehat{\underline{y}}_t = (1 - \nu_1) \mathcal{B} \underline{y}_t \tag{102.152}$$

$$\widehat{\underline{y}}_t = \nu_1 \widehat{\underline{y}}_{t-1} + (1 - \nu_1) \underline{y}_{t-1} \tag{102.153}$$

Now replace $\widehat{\underline{y}}_t \to \underline{s}_t$, $\underline{y}_{t-1} \to \underline{n}_t$ and $\nu_1 \to 1 - \alpha$.

Note that this proof has established the equivalence of two different bnets. Those two bnets are pictured in Fig.102.14.

QED

102.15 Parameter Learning

In this section, we will show how to find an estimate $\widehat{\theta}$ for the parameters $\theta = (\{\alpha_j\}_{\forall j}, \{\nu_j\}_{\forall j}, \sigma^2)$ used in $ARMA(p,q)$. We do this **parameter learning (PL)** by postulating a "quasi" log likelihood function $LL(\theta)$, and maximizing that over θ. The estimate $\widehat{\theta}$ that we obtain is called the **Quasi-Maximum Likelihood Estimate (QMLE)**. The reason for using the prefix "quasi" is that the likelihood probability involved doesn't really satisfy the i.i.d assumption.

Figure 102.14: Bnets for two t-series that were proven equivalent in Claim 187. Note that the 2 bnets have NO arrows in common!

Figure 102.15: Bnet for PL of $AR(3)$. For clarity, we show only the arrows entering nodes $\underline{y}_t$ and $\underline{n}_t$.

102.15.1 PL of $AR(p)$

Recall that if $\{\underline{y}_t\}_{\forall t}$ is an $AR(p)$ t-series,

$$y_t = \sum_{j=1}^{p} \alpha_j y_{t-j} + n_t \tag{102.154}$$

Let $\theta = (\sigma^2, \alpha_{[1,p]})$. We want to learn the parameters θ of the $AR(p)$ bnet Fig.102.15, assuming the TPMs, printed in blue, are as follows.

For $t \geq p+1$

$$P(y_t | y_{<t}, n_t, \theta) = \mathbb{1}\left(y_t = \text{see Eq.(102.154)}\right) \tag{102.155}$$

$$P(n_t|\theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(n_t)^2}{2\sigma^2}\right] \tag{102.156}$$

Note that

$$P(y_{\leq t}|\theta) = \sum_{n_{[p+1,t]}} \left\{ \prod_{\tau\in[p+1,t]} P(y_\tau|y_{[\tau-p,\tau-1]}, n_\tau) P(n_\tau|\theta) \right\} P(y_{\leq p}|\theta) \tag{102.157}$$

$$= \prod_{\tau\in[p+1,t]} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-\left(y_\tau - \sum_{j=1}^p \alpha_j y_{\tau-j}\right)^2}{2\sigma^2}\right] \right\} P(y_{\leq p}|\theta) \tag{102.158}$$

The log likelihood function $LL_t(\theta)$ for this bnet is defined by

$$LL_t(\theta) = \frac{1}{t} \ln P(y_{\leq t}|\theta) \ . \tag{102.159}$$

Hence,

$$LL_t(\theta) \quad = \quad \underbrace{\frac{1}{t} \ln P(y_{\leq p}|\theta)}_{LL_t^{prior}(\theta)} + LL'_t(\theta) \tag{102.160a}$$

where

$$LL'_t(\theta) = \frac{-(t-p)}{t} \ln\sqrt{2\pi\sigma^2} - \frac{1}{2t\sigma^2} \sum_{\tau=p+1}^{t} \left(y_\tau - \sum_{j=1}^{p} \alpha_j y_{\tau-j} \right)^2 \ . \tag{102.160b}$$

Henceforth, we wll assume that $y_{[1,t]}$ is known.

We start by assuming that the prior log likelihood $LL_t^{prior}(\theta)$ is zero. This means that we have uniform (uninformative) priors. Later on, we will consider instead a Gaussian prior that uses the info that $y_{\leq p}$ is known a priori.

If we assume that $y_{[1,t]}$ is known, then we can plug this info into $LL'_t(\theta)$ and use a non-linear optimization method to maximize $LL'_t(\theta)$ and find the optimum θ.

Alternatively, one can use LR. Define the strange looking dataset

$$\mathcal{D} = \{(t, y_{[t-p,t-1]}, \boxed{y_t}) : t\} \tag{102.161}$$

This dataset is strange because it replaces sampling from a population with sampling in time. (i.e., a "stochastic" average by an "ergodic" average.) The datasets that we use to do LR usually satisfy the i.i.d. assumption, but that assumption does not necessarily hold for this one. Luckily, that assumption is not necessary for doing

LR with x-variables $y_{[t-p,t-1]}$, y-variable y_t, and regression coefficients $\alpha_{[1,p]}$. The LR software gives estimates $\widehat{\alpha}_{[1,p]}$ that we can use to calculate residuals

$$\epsilon_\tau = y_\tau - \sum_{j=1}^{p} \widehat{\alpha}_j y_{\tau-j} \tag{102.162}$$

One can estimate σ^2 from those residuals using

$$\widehat{\sigma^2} = \frac{1}{t-p} \sum_{\tau=p+1}^{t} \epsilon_\tau^2 \,. \tag{102.163}$$

So far, we have assumed an uninformative prior. Alternatively, one can assume the Gaussian prior

$$P(y_{\leq p}|\theta) = \frac{1}{(2\pi\sigma^2)^{\frac{p}{2}}} \exp\left[-y_{\leq p}^T \frac{\Gamma^{-1}}{2} y_{\leq p}\right] \tag{102.164}$$

where

$$\Gamma_{i,j} = \left\langle \underline{y}_i, \underline{y}_j \right\rangle = \gamma_{j-i} \tag{102.165}$$

for $i, j \in \mathbb{Z}_{[1,p]}$. We are assuming that all $\underline{y}_\tau$ have been demeaned; i.e., the mean

$$\widehat{\mu} = \frac{1}{t} \sum_{\tau=1}^{t} y_\tau \tag{102.166}$$

has been subtracted from each y_τ for $\tau \in \mathbb{Z}_{[1,t]}$.

102.15.2 PL of $MA(q)$

Figure 102.16: Bnet for PL of $MA(3)$. For clarity, we show only the arrows entering nodes $\underline{y}_t$ and n_t.

Recall that if $\{\underline{y}_t\}_{\forall t}$ is an $MA(q)$ t-series,

$$y_t = \sum_{j=1}^{q} \nu_j n_{t-j} + n_t \tag{102.167}$$

Let $\theta = (\sigma^2, \nu_{[1,q]})$. We want to learn the parameters θ of the $MA(q)$ bnet Fig.102.16, assuming the TPMs, printed in blue, are as follows.

For $t \geq q+1$, let

$$P(y_t|n_{\leq t}, \theta) = \mathbb{1}\left(y_t = \text{see Eq.(102.167)}\right) \tag{102.168}$$

$$P(n_t|\theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(n_t)^2}{2\sigma^2}\right] \tag{102.169}$$

Note that

$$P(y_{\leq t}|\theta) = \sum_{n_{[q+1,t]}} \left\{ \prod_{\tau\in[q+1,t]} P(y_\tau|n_{[\tau-q,\tau-1]}, n_\tau) P(n_\tau|\theta) \right\} P(n_{\leq q}|\theta) \tag{102.170}$$

$$= \prod_{\tau\in[q+1,t]} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-\left(y_\tau - \sum_{j=1}^{q} \nu_j n_{\tau-j}\right)^2}{2\sigma^2}\right] \right\} P(n_{\leq q}|\theta) \tag{102.171}$$

The log likelihood function $LL_t(\theta)$ for this bnet is defined by

$$LL_t(\theta) = \frac{1}{t} \ln P(y_{\leq t}|\theta) \tag{102.172}$$

$$LL_t(\theta) = \underbrace{\frac{1}{t} \ln P(y_{\leq q}|\theta)}_{LL_t^{prior}(\theta)} + LL'_t(\theta) \tag{102.173a}$$

where

$$LL'_t(\theta) = \frac{-(t-q)}{t} \ln\sqrt{2\pi\sigma^2} - \frac{1}{2t\sigma^2} \sum_{\tau=q+1}^{t} \left(y_\tau - \sum_{j=1}^{q} \nu_j n_{\tau-j}\right)^2 \tag{102.173b}$$

Henceforth, we will assume that $n_{\leq q} = 0$. By Eq.(102.167), this implies that $y_{\leq q} = 0$, so the prior log likelihood $LL_t^{prior}(\theta)$ is independent of θ.

Eq.(102.173) expresses $LL'_t(\theta)$ in terms of y_τ's and n_τ's, but the input data consist of only y_τ's. Luckily, Eq.(102.167) can be used to express all the $n_{\leq t}$ in terms of the $y_{\leq t}$, assuming the input $n_{\leq q} = 0$. Next, we will explain how to do this. For

Figure 102.17: Bnet for PL of $MA(2)$. We assume $\underline{n}_1 = \underline{n}_2 = 0$. For clarity, we show only the arrows entering node $\underline{y}_{2+\tau}$ for $\tau = 1, 2, 3$.

definiteness and simplicity, we will assume $q = 2$ and $\nu_1 = \nu_2 = 1$, and we will use Fig.102.17 as a pedagogical aid.

For $q = 2$, Eq.(102.167) gives $\underline{y}_\tau$ as a linear combination of 3 $\underline{n}$ nodes, $\underline{n}_\tau, \underline{n}_{\tau-1}$ and $\underline{n}_{\tau-2}$ But since it's a linear equation, it can be used to express the latest $\underline{n}$ node, i.e., $\underline{n}_\tau$, in terms of $\underline{y}_\tau$ and the 2 previous $\underline{n}$ nodes.

Hence, we see from Fig.102.17 that:

From $\underline{n}_1 = \underline{n}_2 = 0$, we get $\underline{y}_1 = \underline{y}_2 = 0$.

From the $\tau = 1$ bnet, we get

$$\underline{n}_3 = \underline{y}_3 \tag{102.174}$$

From the $\tau = 2$ bnet, we get

$$\underline{n}_4 = \underline{y}_4 - \underline{n}_3 \tag{102.175}$$

From the $\tau = 3$ bnet, we get

$$\underline{n}_5 = \underline{y}_5 - \underline{n}_3 - \underline{n}_4 \tag{102.176}$$

102.15.3 PL of $ARMA(p, q)$

Recall that if $\{\underline{y}_t\}_{\forall t}$ is an $AR(p, q)$ t-series, then

Figure 102.18: Bnet for PL of $AR(3,2)$. For clarity, we show only the arrows entering nodes $\underline{y}_t$ and $\underline{n}_t$.

$$y_t = \sum_{j=1}^{p} \alpha_j y_{t-j} + n_t + \sum_{j=1}^{q} \nu_j y_{t-j} \tag{102.177}$$

Let $\theta = (\sigma^2, \alpha_{[1,p]}, \nu_{[1,q]})$. We want to learn the parameters θ of the $ARMA(p,q)$ bnet Fig.102.18, in which the TPMs, printed in blue, are as follows.

For $t \geq \max(p+1, q+1)$, let

$$P(y_t|y_{<t}, n_{\leq t}, \theta) = \mathbb{1}\left(y_t = \text{see Eq.(102.177)}\right) \tag{102.178}$$

$$P(n_t|\theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(n_t)^2}{2\sigma^2}\right] \tag{102.179}$$

As we did for $AR(p)$ and $MA(q)$, we can now calculate a log likelihood function and maximize it to obtain the parameters θ. But first we will have to assume that $n_{\leq q} = 0$ and $y_{[1,t]}$ is known, and we will have to express the unknown $n_{[q+1,t]}$ in terms of the known $y_{[1,t]}$ using the t-series definition Eq.(102.177).

102.16 $VAR(p)$

Let $\vec{\underline{x}}_t = (\underline{x}_t^1, \underline{x}_t^2, \ldots, \underline{x}_t^{nr})^T \in \mathbb{R}^{nr\times 1}$. Thus, $\vec{\underline{x}}_t$ for each t is a column vector with nr rows. A **vector time series**, denoted variously by $\{\vec{\underline{x}}_1, \vec{\underline{x}}_2, \ldots, \vec{\underline{x}}_{nt}\} = \{\vec{\underline{x}}_t\}_{t=1}^{nt} = \{\vec{\underline{x}}_t\}_{\forall t}$, is a set of $\mathbb{R}^{nr\times 1}$ vectors index by a discrete set of times $\mathbb{Z}_{[0,nt]}$.

Henceforth, we will use the Einstein summation convention for uppercase indices like $A, B, C \in \mathbb{Z}_{[1,nr]}$; i.e., they should be summed over from 1 to nr if they are repeated in a product. For example, $x^A y^A = \sum_{A=1}^{nr} x^A y^A$.

By **vector white noise** $\{\underline{\vec{n}}_t\}_{\forall t} \sim WN(0, \Sigma)$, where $\Sigma \in \mathbb{R}^{nr \times nr}$, we mean a vector t-series $\{\underline{\vec{n}}_t\}_{\forall t}$ that satisfies, for $A, B \in \mathbb{Z}_{[1,nr]}$ and $t, t' \in \mathbb{Z}_{[1,nt]}$,

$$E[\underline{n}_t^A] = 0 \tag{102.180}$$

$$\left\langle \underline{n}_t^A, \underline{n}_{t'}^B \right\rangle = \Sigma^{A,B} \delta(t, t') \tag{102.181}$$

Suppose $\{\underline{\vec{y}}_t\}_{\forall t}$ is a zero mean vector t-series. Hence $\underline{y}_t^A = \underline{Y}_t^A - \mu^A$, $E[\underline{Y}_t^A] = \mu^A$, $E[\underline{y}_t^A] = 0$. Suppose also that $\{\underline{\vec{n}}_t\}_{\forall t} \sim WN(0, \Sigma)$. Then we define a **Vector Auto-Regressive t-series** $VAR(p)$ by

$$\underline{y}_t^A = \sum_{j=1}^{p} \alpha_j^{A,B} \underline{y}_{t-j}^B + \underline{n}_t^A \tag{102.182}$$

The bnet for $VAR(p)$ is the same as the bnet for $AR(p)$ except that now, for all t, the nodes $\underline{y}_t$ and $\underline{n}_t$ are replaced by $\underline{\vec{y}}_t$ and $\underline{\vec{n}}_t$ and now the node weights α_t are $nr \times nr$ matrices with entries $\alpha_t^{A,B}$. For example, for $nr = 2$, we have

$$\begin{bmatrix} \underline{y}_t^1 \\ \underline{y}_t^2 \end{bmatrix} = \sum_{j=1}^{p} \begin{bmatrix} \alpha_j^{1,1} & \alpha_j^{1,2} \\ \alpha_j^{2,1} & \alpha_j^{2,2} \end{bmatrix} \begin{bmatrix} \underline{y}_{t-j}^1 \\ \underline{y}_{t-j}^2 \end{bmatrix} + \begin{bmatrix} \underline{n}_t^1 \\ \underline{n}_t^2 \end{bmatrix} \tag{102.183}$$

Chapter 103

Transfer Learning

Historically, the term Transfer Learning (TL) has been used in AI mostly by Artificial Neural Net (ANN) proponents (see the Wikipedia article on TL, Ref.[189]). Recently, however, a theory of causal TL has begun to emerge (see Chapter 105).

TL in AI is a fairly wide topic that is somewhat ambiguously defined. Some subjects that can be lumped under the heading of TL in AI are:

- data fusion/combining models
- model generalization
- transportability of causal knowledge, external validity (see Chapter 105)

Most AI researchers will agree that it is highly desirable to have TL in AI, because the human brain obviously does plenty of TL to great advantage. Although reams of papers have been written about the subject of TL in AI, a systematic theory of TL in AI that is universally accepted and popular remains elusive. The current theory of TL for ANN looks to me like a grab bag of heuristic approaches that are fragile, meaning they can be easily spoofed. The theory of TL for bnets (see Chapter 105) seems to me to be in much better shape: it's more elaborate, systematic, and it yields more robust results.

In this brief chapter, I will limit myself to describing a possible way of classifying, from a bnet perspective, the various approaches to TL in AI. Note that this method of classification even works for TL for ANNs, because ANNs can be viewed as bnets with deterministic nodes and a layered structure. One can describe TL in AI as a systematic way of defining a bnet $\mathcal{B}^*$ using a bnet $\mathcal{B}$ and other information. Bnet $\mathcal{B}$ is associated with a dataset $\mathcal{D}$, and bnet $\mathcal{B}^*$ is associated with a dataset $\mathcal{D}^*$. A bnet $\mathcal{B} = (\mathcal{S}, \theta)$ comprises a DAG structure $\mathcal{S}$ and a TPM for each node of the DAG. We'll denote the TPMs (a.k.a. parameters) of $\mathcal{B}$ by θ. So let's classify the various approaches to TL in AI by specifying what parts of the structure $\mathcal{S}$ and parameters θ of $\mathcal{B}$ are transferred to the structure $\mathcal{S}^*$ and parameters θ^* of $\mathcal{B}^*$, and what parts of $\mathcal{B}^* = (\mathcal{S}^*, \theta^*)$ are new.

1. **Fine tune parameters**. $\mathcal{S}^* = \mathcal{S}$, $\theta^* \approx \theta$.

   In this approach, we use the dataset $\mathcal{D}^*$ associated with bnet $\mathcal{B}^*$ to adjust slightly the parameters from θ to θ^*.

2. **Replace final layers of $\mathcal{S}$ and of θ by new ones**. $\mathcal{S}^* = \mathcal{S}$ and $\theta^* = \theta$ except for final layers,

   For example, in an ANN, replace the final layers by new ones, and use $\mathcal{D}^*$ to find the parameters of those new final layers.

3. **Transfer only the TPM of a single node $\underline{y}$ of $\mathcal{B}$**. $\mathcal{S}^*$ and θ^* are new except $P^*(y|pa(y)) = P(y|pa(y))$.

Chapter 104

Transformer Networks

The primary reference for this chapter is Ref.[99]. Ref.[99] is the highly influential 2017 paper entitled "Attention is all you need" that introduced **Transformer Networks** (tranets) and Attention into the AI vernacular. Besides Ref.[99], I also read blog posts such as Ref.[36] and the Wikipedia article on tranet (Ref. [190]). For a complete list of the large number of excellent blog post that I read to learn about this subject, see my open source software texnn (Ref.[98]).[1]

Transformer Networks (tranets) have been taking the fields of Natural Language Processing (NLP) and Large Language Models (LLM) by storm in recent years. They were introduced in 2017 and already are the basis of numerous LLMs. Two famous examples are, BERT (Bidirectional Encoder Representations from Transformers) and ChatGPT (Generative Pre-trained Transformer). Both of these have been trained with huge databases, of which all of the English Wikipedia ($\sim 10^9$ words) is but a small part.

How well ChatGPT works was a huge surprise to most people, including experts in AI/ML. My conjecture is that this surprising LLM performance is due to causality. Let me explain. I believe tranets and the LLM that use them, are just curve-fitters (so are Least Squares, vanilla NNs, Convolutional NNs, etc.). But, we lucked out, because tranets are very good at fitting causal data, and the space of all human generated text, including math equations and computer code, is causally connected (i.e., has a **causally connected topology**.).

Normally, tranets are drawn as box diagrams that are somewhat cryptic and ambiguous, at least to me. In this chapter, instead of drawing them as box diagrams, I represent them as causal DAGs (bnets). This makes their causal nature more explicit than the box diagrams, and, in my opinion, also makes them less ambiguous and more understandable than the box diagrams.

Recurrent Neural Nets (RNNs) are discussed in Chapter 81. tranets are quickly displacing RNNs, an older method, in NLP. tranets are better than RNNs for doing

[1] texnn is Python software that I wrote specifically for drawing the bnets of this chapter, but later I generalized it to a stand-alone app that can draw any bnet (including SCMs, NNs and tranets), not just a tranet bnet.

NLP in several important ways. Whereas RNNs analyze the tokens (words) of a sentence sequentially (like a Kalman Filter), tranets analyze them in parallel, and thus are more amenable to parallel computing. Also, because RNNs analyze the words of a sentence sequentially, they tend to give more importance to the end of a sentence than to its beginning. That's because RNNs start forgetting the beginning of a sentence by the time they reach its end, like a patient with Alzheimer's. tranets do not suffer from this malady.

Dynamical bnets are discussed in Chapter 26. In Chapter 81, we showed that RNNs are dynamical bnets. In this chapter we will show that tranets are dynamical bnets too.

In this chapter, we will use the Numpy-like tensor notation discussed in Section C.48. In particular, note that $[n] = [0 : n] = \{0, 1, \ldots, n-1\}$ and that $T^{[n],[m]}$ is an $n \times m$ matrix.

104.1 Tensor Notation

Our tensor notation is discussed in Section C.48. Here is a quick review of some of the more salient facts in that section on tensors. Below, we will often accompany an equation in tensor component notation with the equivalent matrix equation, in parenthesis.

We use Greek letters for tensor indices.

Let $\alpha \in [a]$, $\beta \in [b]$, $\gamma \in [c]$, $\delta \in [d]$, $\nu \in [n]$, $\Delta \in [D]$.

- **reshaping**

$$T^{\nu,\delta} \to T^{\Delta} \quad \left(T^{[n_{\underline{h}}],[d]} \to T^{[D]}\right) \tag{104.1}$$

$$T^{\Delta} \to T^{\nu,\delta} \quad \left(T^{[D]} \to T^{[n_{\underline{h}}],[d]}\right) \tag{104.2}$$

- **concatenation**

$$T^{[n]} = (T^0, T^1, \ldots, T^{n-1}) = (T^{\nu})_{\nu \in [n]} \tag{104.3}$$

- **Hadamard product (element-wise, entry-wise multiplication)**

$$T^{[n]} * S^{[n]} = (T^{\nu} S^{\nu})_{\nu \in [n]} \tag{104.4}$$

- **Matrix multiplication**

  $T^{[n]} = T^{[n],[1]}$ is a column vector.

$$(T^{[n]})^T S^{[n]} = \text{scalar} \tag{104.5}$$

$$T^{[a],[b]}S^{[b],[c]} = \left[\sum_{\beta\in[b]} T^{\alpha,\beta}S^{\beta,\gamma}\right]_{\alpha\in[a],\gamma\in[c]} \tag{104.6}$$

Most treatments of tranets, including the "Attention is all you need" paper, order the operations chronologically from left to right (L2R). So if A occurs before B, they write AB. This is contrary to what is done in Linear Algebra, where one orders the operations chronologically from right to left (R2L), and one writes BA. In this chapter, will adhere to the Linear Algebra convention, since it is so prevalent and is the overwhelming precedent.

104.2 Recurrent Neural Net with Attention

104.2.1 Single Head Attention

Let

ℓ be the maximum number of words allowed in a sentence. Some words might be blanks (padding).

d be the so called **hidden or embedding dimension**.

$e^t_\alpha \in \mathbb{R}^d$ be a d-dimensional column vector for word $\alpha \in [\ell]$ at time t.

$W^t_{\underline{q}}, W^t_{\underline{k}}, W^t_{\underline{v}} \in \mathbb{R}^{d\times d}$ be the weight matrices for time slice t. The letters Q, K, V stand for Query, Key and Value, respectively. These matrices are learned by training the net. They transform e^t_α as follows

$$v^t_\alpha = W^t_{\underline{v}} e^t_\alpha \tag{104.7}$$

$$q^t_\alpha = W^t_{\underline{q}} e^t_\alpha \tag{104.8}$$

$$k^t_\alpha = W^t_{\underline{k}} e^t_\alpha \tag{104.9}$$

Fig.104.1 represents a tranet of a 3-word sentence as a dynamical bnet. The TPMs (Transition Probability Matrices), printed in blue, for bnet Fig.104.1, are as follows:

$$P(v^t_\alpha|e^t_\alpha) = \mathbb{1}(\quad v^t_\alpha = W^t_{\underline{v}} e^t_\alpha \quad) \tag{104.10}$$

$$P(q^t_\alpha|e^t_\alpha) = \mathbb{1}(\quad q^t_\alpha = W^t_{\underline{q}} e^t_\alpha \quad) \tag{104.11}$$

$$P(k^t_\alpha|e^t_\alpha) = \mathbb{1}(\quad k^t_\alpha = W^t_{\underline{k}} e^t_\alpha \quad) \tag{104.12}$$

Figure 104.1: Dynamical bnet with single-head Attention for 3 words. Time-slice t. Note that k^t_α for all α points to $\underline{a}^t_{\alpha'}$ for all α'. Likewise, $\underline{v}^t_\alpha$ for all α points to $\underline{a}^t_{\alpha'}$ for all α'. However, $\underline{q}^t_\alpha$ points only to $\underline{a}^t_\alpha$.

$$P(e^{t+1}_\alpha | a^t_\alpha) = \mathbb{1}(\quad e^{t+1}_\alpha = a^t_\alpha \quad) \tag{104.13}$$

$$P(a^{t+1}_\alpha | v^t_\cdot, q^t_\alpha, k^t_\cdot) = \mathbb{1}(\quad a^{t+1}_\alpha = \sum_{\alpha' \in [\ell]} v^t_{\alpha'} P(\alpha'|\alpha) \quad) \tag{104.14}$$

where the conditional probability $P(\alpha'|\alpha)$ is defined as[2]

[2]The reason sums over $\delta \in [d]$ are divided by $\sqrt{d}$ is to prevent the argument of the exponential

$$P(\alpha'|\alpha) = \text{softmax}\left[\frac{1}{\sqrt{d}}\sum_{\delta\in[d]}(k^t)^{\delta,[\ell]}(q^t)^{\delta,\alpha}\right](\alpha'|\alpha) \tag{104.15}$$

$$= \frac{\exp\left(\frac{1}{\sqrt{d}}(k^t_{\alpha'})^T q^t_\alpha\right)}{\sum_{\alpha''\in[\ell]}\exp\left(\frac{1}{\sqrt{d}}(k^t_{\alpha''})^T q^t_\alpha\right)} \tag{104.16}$$

The right hand side of Eq.(104.14) constitutes an average over all the word vectors $\{\underline{v}^t_\alpha : \alpha \in [\ell]\}$ in a sentence. This average is called the **Attention** (for a single head).[3]

$$\boxed{\text{Attention}^{\delta,\alpha}\left((v^t)^{[d],[\ell]}, (k^t)^{[d],[\ell]}, (q^t)^{[d],[\ell]}\right) = \sum_{\alpha'\in[\ell]}(v^t)^{\delta,\alpha'}P(\alpha'|\alpha)} \tag{104.17}$$

On first encounter, the structure of an Attention bnet seems a bit mysterious. Then one realizes that this is an old friend. If the dashed boxes in Fig.104.1 are each "shrunk" to single nodes, then it becomes a TAN Bayes Net. Each of the 3 subgraphs $\underline{e}^t, (\underline{v}^t, \underline{q}^t, \underline{k}^t), \underline{a}^t$ also constitutes a TAN Bayes net. [4].[5] In broad terms, Fig.104.1 can be described by saying that each word undergoes a special kind of 3-class (q,k,v) Naive Bayes classification, and the results of that classification are sent to the new version of every word (except the q class which only sends info to one word, not all of them).

It's also useful to think of Attention as a filter with input signal $(e^t)^{[d],[\ell]}$ and output signal $(e^{t+1})^{[d],[\ell]}$.

Fig.104.1 can be "folded" (i.e., the 3 words can be represented by as single node). When folded, Fig.104.1 becomes Fig.104.2. Note that in Fig.104.2, we have started indicating the shapes of tensors by a superscript, using the tensor notation explained in Section C.48. We will continue doing this henceforth in this chapter.

The structural equations for Fig.104.2, printed in blue, are as follows.

$$(a^t)^{[d],[\ell]} = \text{Attention}((v^t)^{[d],[\ell]}, (k^t)^{[d],[\ell]}, (q^t)^{[d],[\ell]}) \tag{104.18a}$$

from getting too large.

[3]Variations of this definition of Attention have been proposed. This particular one is the original one from the "Attention is all you need paper". Some people call it the "scaled dot product Attention".

[4]Tree Augmented Naive (TAN) Bayes nets were introduced in Chapter 9.

[5]A **reverse or upside down tree** is obtained by reversing the directions of all the arrows of a tree directed graph. A TAN Bayes net is normally defined as in Chapter9, as a Naive Bayes net augmented with a tree. In an Attention bnet, the Naive Bayes Net is augmented with a reverse tree (RT) instead of a tree (T), so technically Attention bnets contain RTAN Bayes nets, not TAN Bayes nets.

Figure 104.2: Folded version of Fig.104.1 when $\ell = 3$. Note that all orange nodes have the same tensor shape.

$$(e^t)^{[d],[\ell]} = \text{ prior} \tag{104.18b}$$

$$(e^{t+1})^{[d],[\ell]} = (a^t)^{[d],[\ell]} \tag{104.18c}$$

$$(k^t)^{[d],[\ell]} = W_{\underline{k}}^{[d],[d]}(e^t)^{[d],[\ell]} \tag{104.18d}$$

$$(q^t)^{[d],[\ell]} = W_{\underline{q}}^{[d],[d]}(e^t)^{[d],[\ell]} \tag{104.18e}$$

$$(v^t)^{[d],[\ell]} = W_{\underline{v}}^{[d],[d]}(e^t)^{[d],[\ell]} \tag{104.18f}$$

104.2.2 Multi-Head Attention

In this section, we will generalize the single head Attention, as defined in the previous section, to multi-head Attention.

Let

$n_{\underline{h}}$ = number of heads. $\nu \in [n_{\underline{h}}]$.

d = same as before, the hidden, embedding dimension. $\delta \in [d]$

$D = n_{\underline{h}} d$. $\Delta \in [D]$. We will do some tensor reshaping: $T^{[n_{\underline{h}}],[d]} \to T^{[D]}$, or, in component form, $T^{\nu,\delta} \to T^{\Delta}$.

Consider weight matrices $W_{\underline{k}}^{[D],[d]}, W_{\underline{q}}^{[D],[d]}$, and $W_{\underline{v}}^{[D],[d]}$ such that

$$(k^t)^{\nu,\delta,\alpha} = \sum_{\delta' \in [d]} W_{\underline{k}}^{\nu,\delta,\delta'} (e^t)^{\delta',\alpha} \tag{104.19}$$

$$(q^t)^{\nu,\delta,\alpha} = \sum_{\delta' \in [d]} W_{\underline{q}}^{\nu,\delta,\delta'} (e^t)^{\delta',\alpha} \tag{104.20}$$

$$(v^t)^{\nu,\delta,\alpha} = \sum_{\delta' \in [d]} W_{\underline{v}}^{\nu,\delta,\delta'} (e^t)^{\delta',\alpha} \tag{104.21}$$

We define the **Multi-head Attention** by

$$\boxed{\text{Attention}^{\nu,\delta,\alpha}\left((v^t)^{[D],[\ell]}, (k^t)^{[D],[\ell]}, (q^t)^{[D],[\ell]}\right) = \sum_{\alpha' \in [\ell]} (v^t)^{\nu,\delta,\alpha'} P(\alpha'|\alpha,\nu)} \tag{104.22}$$

where

$$P(\alpha'|\alpha,\nu) = \text{softmax}\left[\frac{1}{\sqrt{d}} \sum_{\delta \in [d]} (k^t)^{\nu,\delta,[\ell]} (q^t)^{\nu,\delta,\alpha}\right](\alpha'|\alpha,\nu) \tag{104.23}$$

$$= \frac{\exp\left[\frac{1}{\sqrt{d}} \sum_{\delta \in [d]} (k^t)^{\nu,\delta,\alpha'} (q^t)^{\nu,\delta,\alpha}\right]}{\sum_{\alpha'' \in [\ell]} \exp\left[\frac{1}{\sqrt{d}} \sum_{\delta \in [d]} (k^t)^{\nu,\delta,\alpha''} (q^t)^{\nu,\delta,\alpha}\right]} \tag{104.24}$$

The structural equations, printed in blue, for the bnet Fig.104.3, are as follows. Note that Attention() always has the same tensor shape as its 3 arguments. Note also that the 3 weight matrices $W_{\underline{k}}^{[D],[d]}$, $W_{\underline{q}}^{[D],[d]}$, and $W_{\underline{v}}^{[D],[d]}$ raise the hidden dimension, whereas the weight matrix $W_{\underline{a}}^{[d],[D]}$ lowers it. $W_{\underline{a}}^{[d],[D]} = 1$ in the single head case.

$$(a^t)^{[D],[\ell]} = \text{Attention}((v^t)^{[D],[\ell]}, (k^t)^{[D],[\ell]}, (q^t)^{[D],[\ell]}) \tag{104.25a}$$

$$(e^t)^{[d],[\ell]} = \text{prior} \tag{104.25b}$$

$$(e^{t+1})^{[d],[\ell]} = W_{\underline{a}}^{[d],[D]} (a^t)^{[D],[\ell]} \tag{104.25c}$$

$$(k^t)^{[D],[\ell]} = W_{\underline{k}}^{[D],[d]} (e^t)^{[d],[\ell]} \tag{104.25d}$$

Figure 104.3: Dynamical bnet with single-head Attention for ℓ words. Time-slice t. This is a generalization of the single head Attention of Fig.104.2. Note that all orange nodes have the same tensor shape.

$$(q^t)^{[D],[\ell]} = W_{\underline{q}}^{[D],[d]}(e^t)^{[d],[\ell]} \tag{104.25e}$$

$$(v^t)^{[D],[\ell]} = W_{\underline{v}}^{[D],[d]}(e^t)^{[d],[\ell]} \tag{104.25f}$$

104.3 Vanilla tranet

In this section, we will discuss the tranet of the "Attention is all you need" paper, Ref .[99]. As is common in the literature, we will refer to that tranet as the "Vanilla" tranet. Ref.[99] describes its tranet graphically with Fig.104.4. Our goal is to find a causal DAG (bnet) version of that figure.

Let

$\ell =$, context window, maximum number of words in a sentence segment. $\alpha \in [\ell]$, $\ell \sim 100$

$L =$ number of words in vocabulary, $\beta \in [L]$, $L >> \ell$

$d = d_{\underline{q}} = d_{\underline{k}} = d_{\underline{v}} = 64$, hidden dimension per head, $\delta \in [d]$.

$n_{\underline{h}} = 8$, number of heads, $\nu \in [n_{\underline{h}}]$

$D = n_{\underline{h}} d = 8(64) = 512$, hidden dimension for all heads, $\Delta \in [D]$

$\Lambda = 6$, number copies, connected in series, of boxed bnet, $\lambda \in [\Lambda]$

Before we present the bnet version of Fig.104.4, we discuss some of the definitions needed to understand and motivate Fig.104.4.

Figure 104.4: Vanilla tranet

- **Encoder Input** $x^{\beta,\alpha}$

$$x^{\beta,\alpha} = \delta(\beta, \beta(\alpha)) \left(x^{[L],[\ell]} \text{ has one hot columns.} \right) \tag{104.26}$$

- **Embedding (a.k.a. encoding) Matrix** $\mathcal{E}^{\delta,\beta}$

$$e^{\delta,\alpha} = \sum_{\beta} \mathcal{E}^{\delta,\beta} x^{\beta,\alpha} \quad \left(e^{[d],[\ell]} = \mathcal{E}^{[d],[L]} x^{[L],[\ell]} \right) \tag{104.27}$$

- **Weight matrices** $W_{\underline{q}}, W_{\underline{k}}, W_{\underline{v}}$

$$Q^{\nu,\delta,\alpha} = \sum_{\delta'} W_{\underline{q}}^{\nu,\delta,\delta'} e^{\delta',\alpha} \quad \left(Q^{[D],[\ell]} = W_{\underline{q}}^{[D],[d]} e^{[d],[\ell]} \right) \tag{104.28}$$

$$K^{\nu,\delta,\alpha} = \sum_{\delta'} W_{\underline{k}}^{\nu,\delta,\delta'} e^{\delta',\alpha} \quad \left(K^{[D],[\ell]} = W_{\underline{k}}^{[D],[d]} e^{[d],[\ell]} \right) \tag{104.29}$$

$$V^{\nu,\delta,\alpha} = \sum_{\delta'} W_{\underline{v}}^{\nu,\delta,\delta'} e^{\delta',\alpha} \quad \left(V^{[D],[\ell]} = W_{\underline{v}}^{[D],[d]} e^{[d],[\ell]} \right) \tag{104.30}$$

- **Multi-head Attention**

$$B^{\nu,\alpha',\alpha} = \frac{1}{\sqrt{d}} \sum_{\delta} K^{\nu,\delta,\alpha'} Q^{\nu,\delta,\alpha} \quad \left(B^{\nu,\alpha',\alpha} = \frac{1}{\sqrt{d}} (K^{\nu,[d],\alpha'})^T Q^{\nu,[d],\alpha} \right) \tag{104.31}$$

$$A^{\nu,\delta,\alpha} = \sum_{\alpha'} V^{\nu,\delta,\alpha'} \underbrace{\text{softmax}(B^{\nu,[\ell],\alpha})(\alpha'|\alpha,\nu)}_{P(\alpha'|\alpha,\nu)} \tag{104.32}$$

$$\sum_{\alpha' \in [\ell]} P(\alpha'|\alpha,\nu) = 1 \tag{104.33}$$

$$A^{\nu,\delta,\alpha} \to A^{\Delta,\alpha} \left(A^{[n_{\underline{h}}],[d],[\ell]} \to A^{[D],[\ell]} \right) \tag{104.34}$$

Important: Note that the softmax() makes the α' component a probability, not the α one!

For example, suppose $\nu = 1$ (one head), $\ell = 2$ (a 2 word segment), and $d = 3$ (hidden dimension is 3). The $Q^{[3],[2]}, K^{[3],[2]}, V^{[3],[2]}$ are 3×2 matrices (i.e., two 3-dim column vectors). One uses the $Q^{[3],[2]}$ and $K^{[3],[2]}$ to arrive at a 2×2 matrix $P(\alpha'|\alpha)$ of probabilities. Then one uses that matrix of probabilities to replace

$$\left[V^{[3],0}, V^{[3],1} \right] \to \left[V^{[3],0} P(0|0) + V^{[3],1} P(1|0), V^{[3],0} P(0|1) + V^{[3],1} P(1|1) \right] \tag{104.35}$$

- **Positional Embedding Matrix $\mathcal{E}^{\delta,\beta}_{pos}$**

$$\mathcal{E}^{\delta,\beta}_{pos} = \begin{cases} \sin\left(2\pi \frac{\beta}{(2\pi)10^{4\delta/d}}\right) = \sin(2\pi \frac{\beta}{\lambda(\delta)}) & \text{if } \delta \text{ is even} \\ \cos\left(2\pi \frac{\beta}{(2\pi)10^{4(\delta-1)/d}}\right) = \cos(2\pi \frac{\beta}{\lambda(\delta)}) & \text{if } \delta \text{ is odd} \end{cases} \tag{104.36}$$

$\mathcal{E}^{\delta,\beta}_{pos}$ changes in phase by $\pi/2$ every time δ changes by 1. Its wavelength λ is independent of β, but increases rapidly with δ, from $\lambda(\delta = 0) = 2\pi * 1$ to $\lambda(\delta = d) = 2\pi * 10^4$.

Total Embedding equals initial embedding plus positional embedding:

$$\mathcal{E}^{\delta,\beta} = \mathcal{E}^{\delta,\beta}_0 + \mathcal{E}^{\delta,\beta}_{pos} \tag{104.37}$$

The purpose of positional embedding is to take $e^{\beta,\alpha}$ to $e^{\delta,\alpha} = \sum_\beta \mathcal{E}^{\delta,\beta}_{pos} e^{\beta,\alpha}$ where $e^{\delta,\alpha}$ changes quickly as δ (i.e., position) changes.

- **ReLU**

  For a tensor T of arbitrary shape,

$$ReLU(T) = (T)_+ = max(0, T) \tag{104.38}$$

  max element-wise.

- **Feed Forward Neural Net**

$$F(e^{\delta,\alpha}) = \sum_{\Delta \in [n_{ff}]} W_2^{\delta,\Delta} ReLU \left(\sum_{\delta' \in [d]} W_1^{\Delta,\delta'} e^{\delta',\alpha} + b_1^{\Delta,\alpha} \right) + b_2^{\delta,\alpha} \tag{104.39}$$

  n_{ff} is called the `intermediate_size` in BERT.

- **Softmax**

  softmax() takes a vector and returns a vector of probabilities of the same length

$$e^{[n]} \to P^{[n]} \tag{104.40}$$

  where

$$P^\alpha = \frac{\exp(e^\alpha)}{\sum_{\alpha \in [n]} \exp(e^\alpha)} \quad \left(P^{[n]} = \frac{\exp(e^{[n]})}{\| \exp(e^{[n]}) \|_0} \right) \tag{104.41}$$

  For example,

$$(1, 0, 0) \to (e, 1, 1)/norm \tag{104.42}$$

$$(10, 0, 0) \to (e^{10}, 1, 1)/norm \approx (1, 0, 0) \tag{104.43}$$

  For any $a \in \mathbb{R}$,

$$(a, a, a) \to \frac{1}{3}(1, 1, 1) \tag{104.44}$$

- **Skip Connection (Add & Normalize)**

  A **skip connection** is when you split the input to a **filter** into two streams, one stream goes through the filter, the other doesn't. The one that doesn't is then merged with the output of the filter via a **add & normalize** node. The reason for making skip connections is that the signal exiting a filter is usually full of jumps and kinks. By merging that filter output with some of the filter input, one smooths out the filter output to some degree. This makes back-propagation differentiation better behaved.

  The filter might be a Multi-Head Attention or a Feed Forward NN.

Add & Normalize just means $(A+B)/norm$ where A and B are the two input signals and "norm" is some norm of $A+B$ (for instance, $\| A+B \|_2$).

Normalization keeps the signal from growing too big and saturating the signal that will enter components upstream. Normalization can also involve subtracting the mean $\langle X \rangle$ of the signal X so as to get a signal $X - \langle X \rangle$ that has zero mean.

- **Redundancy**

  For better results, the Encoder and Decoder both contain Λ copies, connected in series, of the boxed bnet.

  Redundancy (see Chapter 85) has been used to avoid catastrophic failure at least as early as the dawn of the age of rocketry, when it was used to avoid the all too common occurrence of exploding rockets. There are 2 basic types of redundancy: in series connection (as in the repeated identical layers in a feedforward NN or a recurrent NN), and in parallel connection (as in tranet heads, and the plates in a bnet (see Chapter 76)).

- **Right Shifted Outputs**

  "Outputs (Shifted Right)" in Fig.104.4 refers to what is called **forced teaching** in the RNN (recurrent neural net) literature. We explain forced teaching in Fig.104.5.

Figure 104.5: Training and Inference for vanilla transformer. "enc" and "dec" denote the encoder and decoder, respectively. A hash character represents the SOS (start of sentence) token, and a period represents the EOS (end of sentence) token. Capital letters represent ground truth tokens, and lower case ones represent predictions.

- **Masked Attention**

$$P(\alpha'|\alpha,\nu) = 0 \quad \text{if } \alpha' < \alpha \tag{104.45}$$

α, and α' are word positions in a sentence, and α' is in the future (downstream) compared to α. So as to not violate causality, this condition enforces the constraint that no attention is paid to word positions in the future of α.

104.3.1 Single Head Attention

Fig.104.6 gives a bnet representation of the "Single Head Attention" portion of Fig.104.4. The structural equations for that bnet, printed in blue, are as follows.

Figure 104.6: Single Head Attention. (Scaled Dot Product)

$$A^{[d],[\ell]} = V^{[d],[\ell]} P^{[\ell],[\ell]} \quad \left(\text{Note that } \sum_{\alpha \in [\ell]} P^{\alpha,[\ell]} = 1\right) \tag{104.46a}$$

$$B^{[\ell],[\ell]} = (K^{[d],[\ell]})^T Q^{[d],[\ell]} \tag{104.46b}$$

$$K^{[d],[\ell]} = \text{ prior} \tag{104.46c}$$

$$M^{[\ell],[\ell]} = \text{mask}(S^{[\ell],[\ell]}) \tag{104.46d}$$

$$P^{[\ell],[\ell]} = \text{softmax}(M^{[\ell],[\ell]}) \quad \left(\text{Note that } \sum_{\alpha\in[\ell]} P^{\alpha,[\ell]} = 1\right) \tag{104.46e}$$

$$Q^{[d],[\ell]} = \text{ prior} \tag{104.46f}$$

$$S^{[\ell],[\ell]} = \frac{B^{[\ell],[\ell]}}{\sqrt{d}} \tag{104.46g}$$

$$V^{[d],[\ell]} = \text{ prior} \tag{104.46h}$$

104.3.2 Multi-Head Attention

Fig.104.7 gives a bnet representation of the "Multi-Head Attention" portion of Fig.104.4. The structural equations for that bnet, printed in blue, are as follows.

$$A^{[D],[\ell]} = [A_0^{[d],[\ell]}|A_1^{[d],[\ell]}] \tag{104.47a}$$

$$A_0^{[d],[\ell]} = \text{Attention}(V_0^{[d],[\ell]}, K_0^{[d],[\ell]}, Q_0^{[d],[\ell]}) \tag{104.47b}$$

$$A_1^{[d],[\ell]} = \text{Attention}(V_1^{[d],[\ell]}, K_1^{[d],[\ell]}, Q_1^{[d],[\ell]}) \tag{104.47c}$$

$$K^{[D],[\ell]} = W_{\underline{k}}^{[D],[d]} e^{[d],[\ell]} \tag{104.47d}$$

$$K_0^{[d],[\ell]} = \text{linear}(K^{[D],[\ell]}) \ \text{ (split, then project a component)} \tag{104.47e}$$

$$K_1^{[d],[\ell]} = \text{linear}(K^{[D],[\ell]}) \ \text{ (split, then project a component)} \tag{104.47f}$$

Figure 104.7: Multi-head Attention with 2 heads. Note that the orange nodes all have the same tensor shape.

$$O^{[d],[\ell]} = W_{\underline{a}}^{[d],[D]} A^{[D],[\ell]} \tag{104.47g}$$

$$Q^{[D],[\ell]} = W_{\underline{q}}^{[D],[d]} e^{[d],[\ell]} \tag{104.47h}$$

$$Q_0^{[d],[\ell]} = \text{linear}(Q^{[D],[\ell]}) \ \ \text{(split, then project a component)} \tag{104.47i}$$

$$Q_1^{[d],[\ell]} = \text{linear}(Q^{[D],[\ell]}) \;\; \text{(split, then project a component)} \tag{104.47j}$$

$$V^{[D],[\ell]} = W_{\underline{v}}^{[D],[d]} e^{[d],[\ell]} \tag{104.47k}$$

$$V_0^{[d],[\ell]} = \text{linear}(V^{[D],[\ell]}) \;\; \text{(split, then project a component)} \tag{104.47l}$$

$$V_1^{[d],[\ell]} = \text{linear}(V^{[D],[\ell]}) \;\; \text{(split, then project a component)} \tag{104.47m}$$

$$e^{[d],[\ell]} = \;\; \text{prior} \tag{104.47n}$$

104.3.3 Encoder

Fig.104.8 gives a bnet representation of the "Encoder" portion of Fig.104.4. The structural equations for that bnet, printed in blue, are as follows.

$$A^{[D],[\ell]} = \text{Attention}(Q^{[D],[\ell]}, K^{[D],[\ell]}, V^{[D],[\ell]}) \tag{104.48a}$$

$$e^{[d],[\ell]} = \mathcal{E}^{[d],[L]} x^{[L],[\ell]} \tag{104.48b}$$

$$F^{[d],[\ell]} = \text{feed_forward_nn}(N^{[d],[\ell]}) \tag{104.48c}$$

$$K^{[D],[\ell]} = W_{\underline{k}}^{[D],[d]} e^{[d],[\ell]} \tag{104.48d}$$

$$n^{[d],[\ell]} = \text{normalize}(N^{[d],[\ell]} + F^{[d],[\ell]}) \tag{104.48e}$$

Figure 104.8: Encoder of Vanilla Transformer Net. Λ copies of the boxed part are connected in series.

$$N^{[d],[\ell]} = \text{normalize}(e^{[d],[\ell]} + W_{\underline{a}}^{[d],[D]} A^{[D],[\ell]}) \tag{104.48f}$$

$$Q^{[D],[\ell]} = W_{\underline{q}}^{[D],[d]} e^{[d],[\ell]} \tag{104.48g}$$

$$V^{[D],[\ell]} = W_{\underline{v}}^{[D],[d]} e^{[d],[\ell]} \tag{104.48h}$$

$$x^{[L],[\ell]} = \text{ prior} \tag{104.48i}$$

104.3.4 Decoder

Fig.104.9 gives a bnet representation of the "Decoder" portion of Fig.104.4. The structural equations for that bnet, printed in blue, are as follows.

Figure 104.9: Decoder of Vanilla Transformer Net. Λ copies of the boxed part are connected in series.

$$a^{[D],[\ell]} = \text{Attention}(v^{[D],[\ell]}, k^{[D],[\ell]}, q^{[D],[\ell]}) \tag{104.49a}$$

$$A^{[D],[\ell]} = \text{Attention}(Q^{[D],[\ell]}, K^{[D],[\ell]}, V^{[D],[\ell]}) \tag{104.49b}$$

$$e^{[d],[\ell]} = \mathcal{E}^{[d],[L]} x^{[L],[\ell]} \tag{104.49c}$$

$$F^{[d],[\ell]} = \text{feed_forward_nn}(j^{[d],[\ell]}) \tag{104.49d}$$

$$I^{[L],[\ell]} = W_{fin}^{[L],[d]} Y^{[d],[\ell]} \tag{104.49e}$$

$$j^{[d],[\ell]} = \text{normalize}(U_{\underline{a}}^{[d],[D]} a^{[D],[\ell]} + J^{[d],[\ell]}) \tag{104.49f}$$

$$J^{[d],[\ell]} = \text{normalize}(W_{\underline{a}}^{[d],[D]} A^{[D],[\ell]} + e^{[d],[\ell]}) \tag{104.49g}$$

$$K^{[D],[\ell]} = W_{\underline{k}}^{[D],[d]} e^{[d],[\ell]} \tag{104.49h}$$

$$k^{[D],[\ell]} = U_{\underline{k}}^{[D],[d]} n^{[d],[\ell]} \tag{104.49i}$$

$$n^{[d],[\ell]} = \text{ Prior coming from Encoder.} \tag{104.49j}$$

$$P^{[L],[\ell]} = \text{softmax}(I^{[L],[\ell]}) \ \ (\sum_{\alpha \in [\ell]} P^{[L],\alpha} = 1) \tag{104.49k}$$

$$q^{[D],[\ell]} = U_{\underline{q}}^{[D],[d]} J^{[d],[\ell]} \tag{104.49l}$$

$$Q^{[D],[\ell]} = W_{\underline{q}}^{[D],[d]} e^{[d],[\ell]} \tag{104.49m}$$

$$V^{[D],[\ell]} = W_{\underline{v}}^{[D],[d]} e^{[d],[\ell]} \tag{104.49n}$$

$$v^{[D],[\ell]} = U_{\underline{v}}^{[D],[d]} n^{[d],[\ell]} \tag{104.49o}$$

$$x^{[L],[\ell]} = \text{ prior, right shifted output} \tag{104.49p}$$

$$Y^{[d],[\ell]} = \text{normalize}(F^{[d],[\ell]} + J^{[d],[\ell]}) \tag{104.49q}$$

104.4 BERT

I used the Wikipedia article on BERT, Ref[112] to write this section.

BERT (Bidirectional Encoder Representations from Transformer) is a realization of the Encoder half of the Vanilla tranet. One can either add a smaller NN to the output of BERT (this is called **fine-tuning**), or one can add a de-embedding layer to its output so that the total device takes word lists to word lists.

In the language of Bayesian Networks, fine-tuning is the same as using BERT as a prior probability. See Chapter 90 on sentence splitting for an example of BERT fine-tuning.

104.4.1 BERT parameter values

BERT comes in two sizes, base and large. See Table 104.1 for a listing of some BERT parameter values.

	BERT base	BERT large
ℓ, context window	512	512
L, `vocab_size`	30,522	30,522
d, `hidden_size`	768	1024
$n_{\underline{h}}$, `num_attention_heads`	12	16
Λ, `num_hidden_layers`	12	24
D', `intermediate_size`	3,072	3,072
number of parameters	110M	340M

Table 104.1: Some hyper-parameter values for BERT base and BERT large

104.4.2 BERT Embedding

So far, we have described the embedding step as a single step from tokenization into words, to 1 hot vectors, to embedding vectors. There are other additional steps in the embedding process that we haven't described so far (namely, tokenization into subwords, adding special tokens, and padding). We would like to describe those additional steps now, in the context of the BERT model. Here is an example.

Let's consider a short sentence: "The cat is on the mat."

1. **Tokenization into words:** Tokenize the sentence into individual words:

   "The", "cat", "is", "on", "the", "mat", "."

2. **Tokenization into subwords:** Further tokenize words into subword units using WordPiece tokenization or a similar method. For example:

$$
\begin{aligned}
&The \to The \\
&cat \to ca, t \\
&is \to is \\
&on \to on \\
&the \to the \\
&mat \to mat \\
&. \to .
\end{aligned}
$$

   Any subword not appearing in BERTs vocabulary is replaced by [UNK] for "unknown".

3. **Adding Special Tokens:** Add special tokens, such as [CLS] (classification) at the beginning and [SEP] (separator) at the end:

   "[CLS]", "The", "ca", "t", "is", "on", "the", "mat", ".", "[SEP]"

4. **Padding:** If necessary, pad or truncate the sequence to a fixed length. Add padding tokens "[PAD]" to reach a specified sequence length.

5. **Embedding Matrix:** Create a tensor with 1-hot columns

$$x^{\beta,\alpha} = \delta(\beta, \beta(\alpha)) \tag{104.50}$$

   where $\alpha \in [\ell]$, $\beta \in [L]$ and where $\beta(\alpha)$ is the location in the BERT vocab corresponding to token α in the padded string. Now multiply x times the previously discussed embedding matrix $\mathcal{E}$ to get

$$e^{[d],[\ell]} = \mathcal{E}^{[d],[L]} x^{[L],[\ell]} \in \mathbb{R}^{d \times \ell} \tag{104.51}$$

   This gives a vector in $\mathbb{R}^d$ for each token α in the padded string. The matrix $\mathcal{E}$ is pre-trained and captures contextual information and word similarities. It can also include positional embedding, as discussed before.

104.4.3 BERT training

BERT was trained[6] simultaneously on two tasks.[7]

1. **language modeling**: 15% of tokens were selected for prediction. Those tokens selected for prediction were replaced by the [MASK] token 80% of the time, by a random word 10% of the time, and not replaced at all 10% of the time. The training objective was to predict the selected token given its context.

2. **next sentence prediction:** Given two spans of text, the model predicts if these two spans appeared sequentially in the training corpus, outputting either [IsNext] or [NotNext]. For example,

   - Given "[CLS] my dog is cute [SEP] he likes playing", the model should output token [IsNext].
   - Given "[CLS] my dog is cute [SEP] how do magnets work", the model should output token [NotNext]

[6]Sometimes this is called "pre-training" to distinguish it from the "training" of the smaller NN that is attached to the output of BERT when doing fine-tuning.

[7]This section on BERT training quotes Wikipedia Ref.[112] heavily.

Chapter 105

Transportability of Causal Knowledge

This chapter is mostly based on Refs.[64] and [50].

Figure 105.1: Example of selection bnet $G_{\underline{s}}$ created from bnet G.

Suppose one wants to transfer causal knowledge from a **source population** Σ to a **target population** Σ^*.

Given a bnet G, define a **selection diagram** $G_{\underline{s}}$ as a bnet formed by adding to G a new root node $\underline{s}$ and new arrows pointing from **switch node (a.k.a, selection node** $\underline{s}$ to one or more **target nodes** of G. We'll call the set of target nodes of $\underline{s}$ the **target set** $T_{\underline{s}}$. $\underline{s} = 0$ corresponds to population Σ and $\underline{s} = 1$ to population Σ^*. For bnet G, the TPM for a node $\underline{x}$ with parents $pa(\underline{x})$, is given by:

$$P_G(x|pa(x)) = P(x|pa(x)) \tag{105.1}$$

For bnet $G_{\underline{s}}$, nodes $\underline{x}$ with parents $pa(\underline{x})$, where $\underline{s} \notin pa(\underline{x})$, have TPMs:

$$P_{G_{\underline{s}}}(x|pa(x)) = P(x|pa(x)) \ . \tag{105.2}$$

Nodes $\underline{x}$ with parents $pa(\underline{x}) \cup \underline{s}$, have TPMs:

$$P_{G_{\underline{s}}}(x|pa(x), s) = \begin{cases} P(x|pa(x), \underline{s} = 0) = P(x|pa(x)) & \text{if } \underline{s} = 0 \\ P(x|pa(x), \underline{s} = 1) = P^*(x|pa(x)) & \text{if } \underline{s} = 1 \end{cases} \tag{105.3}$$

Fig.105.1 shows an example of a selection diagram $G_{\underline{s}}$. In that figure, the target set of $\underline{s}$ is $\{\underline{x}, \underline{z}\}$.

All this can be generalized so as to have more than one switch node, with the target sets of the switch nodes being disjoint, and such that a switch node can have more than 2 states.

Do-transport formulae

Claim 188 *(Trivial Memoryless Transportability, from Ref.[64])*
If $\begin{tikzcd} & \underline{z} \arrow[dl] \arrow[dr] \arrow[dd, bend right=60, dashed] & \underline{s} \arrow[d] \\ \underline{x} \arrow[rr] & & \underline{y} \arrow[ll, bend left=60, dashed] \end{tikzcd}$ *where* $\underline{s} \in \{0, 1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x} = x, z) = P^*(y|x, z) \quad \textit{(replace } \mathcal{D} \textit{ by } 1\textit{, keep } P^*\textit{)} \tag{105.4}$$

$$\begin{tikzcd} & z \arrow[dr] & \underline{s} = 1 \arrow[d] \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} = \begin{tikzcd} & z \arrow[dr] & \underline{s} = 1 \arrow[d] \\ x \arrow[rr] & & y \end{tikzcd} \tag{105.5}$$

proof: See Claim 51.
QED

Claim 189 *(Direct Transportability, a.k.a. External Validity, from Ref.[64])*
If $\begin{tikzcd} \underline{s} \arrow[r] & \underline{z} \arrow[dl] \arrow[dr] \arrow[dd, bend right=60, dashed] & \\ \underline{x} \arrow[rr] & & \underline{y} \arrow[ll, bend left=60, dashed] \end{tikzcd}$ *where* $\underline{s} \in \{0, 1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x} = x, z) = P(y|\mathcal{D}\underline{x} = x, z) \quad \textit{(replace } P^* \textit{ by } P\textit{, keep } \mathcal{D}\textit{)} \tag{105.6}$$

$$\begin{tikzcd} \underline{s} = 1 \arrow[r] & z \arrow[dr] & \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} = \begin{tikzcd} & z \arrow[dr] & \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} \tag{105.7}$$

Furthermore,

$$P^*(y|\mathcal{D}\underline{x} = x) = \sum_z P(y|\mathcal{D}\underline{x} = x, z) P^*(z) \tag{105.8}$$

$$\begin{tikzcd} \underline{s} = 1 \arrow[dr] & \\ \mathcal{D}\underline{x} = x \arrow[r] & y \end{tikzcd} = \begin{tikzcd} \underline{s} = 1 \arrow[r] & \sum z \arrow[dr] & \\ \mathcal{D}\underline{x} = x \arrow[rr] & & y \end{tikzcd} \tag{105.9}$$

proof: See Claim 52.
QED

Claim 190 *(S-Admissible Transportability, from Ref.[64])*

If $\underline{s} \to \underline{z} \to \underline{a}$ *where* $\underline{s} \in \{0,1\}$ *is a switch node, then*

$$\begin{tikzcd} \underline{s} \arrow[r] & \underline{z} \arrow[r] \arrow[d] \arrow[dl] & \underline{a} \arrow[d] \\ \underline{x} \arrow[rr] & & \underline{y} \arrow[ll, dashed, bend left] \end{tikzcd}$$

$$P^*(y|\mathcal{D}\underline{x} = x) = \sum_a P(y|\mathcal{D}\underline{x} = x, a) P^*(a) \tag{105.10}$$

$$\begin{tikzcd} \underline{s} = 1 \arrow[dr] & & \\ \mathcal{D}\underline{x} = x \arrow[r] & y \end{tikzcd} = \begin{tikzcd} \underline{s} = 1 \arrow[r] & \sum a \arrow[d] \\ \mathcal{D}\underline{x} = x \arrow[r] & y \end{tikzcd} \tag{105.11}$$

proof: See Claim 53.
QED

Claim 191 *(Non-transportability, from Ref.[64])*

If $\underline{h} \to \underline{x}, \underline{h} \to \underline{y}, \underline{s} \to \underline{y}, \underline{x} \to \underline{y}$ *where* $\underline{s} \in \{0,1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x} = x) = P^*(y|\mathcal{D}\underline{x} = x) \tag{105.12}$$

$$\begin{tikzcd} & \underline{s} = 1 \arrow[d] \\ \mathcal{D}\underline{x} = x \arrow[r] & y \end{tikzcd} = same \tag{105.13}$$

proof: See Claim 54.
QED

Claim 192 *(from Ref.[64])*

If $\underline{s} \to \underline{z}, \underline{s} \to \underline{h} \to \underline{z}, \underline{h} \to \underline{x}, \underline{h} \to \underline{y}, \underline{x} \to \underline{y}, \underline{y} \dashrightarrow \underline{x}$ *where* $\underline{s} \in \{0,1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x} = x) = P(y|\mathcal{D}\underline{x} = x) \tag{105.14}$$

$$\begin{tikzcd}
\underline{s} = 1 \arrow[dr] & & & \\
\mathcal{D}\underline{x} = x \arrow[r] & y & = \mathcal{D}\underline{x} = x \arrow[r] & y
\end{tikzcd} \tag{105.15}$$

proof: See Claim 55.
QED

Claim 193 *(from Ref.[64])*

If $\begin{tikzcd}
\underline{s} \arrow[r] \arrow[dr] & \underline{h} \arrow[dl] \arrow[dr] \arrow[dll, dashed, bend right] & \\
\underline{x} \arrow[r] & \underline{z} \arrow[r] & \underline{y} \arrow[ll, dashed, bend left]
\end{tikzcd}$ *where* $\underline{s} \in \{0, 1\}$ *is a switch node, then*

$$P^*(y|\mathcal{D}\underline{x} = x) = \sum_z P(y|\mathcal{D}\underline{x} = x, z) P^*(z|x) \tag{105.16}$$

$$\begin{tikzcd}
\underline{s} = 1 \arrow[dr] & \mathcal{D}\underline{x} = x \arrow[d] & & \underline{s} = 1 \arrow[dr] & & \mathcal{D}\underline{x} = x \arrow[d] \\
& y & = & x \arrow[r] & \sum z \arrow[r] & y
\end{tikzcd} \tag{105.17}$$

proof: See Claim 56.
QED

Chapter 106

Turbo Codes

This chapter is based on Ref.[48].

In this chapter, vectors with n components will be indicated by an n superscript. For example, $a^n = (a_0, a_1, \ldots, a_{n-1})$.

Consider an n-letter message $u^n = (u_0, u_1, \ldots, u_{n-1})$, where for all i, $u_i \in \mathcal{A}$ is an element of an alphabet $\mathcal{A}$, and where for all i, the $\underline{u}_i$ are i.i.d.. Suppose u^n is encoded deterministically in two different ways, $e_1(u^n)$ and $e_2(u^n)$. After passing through the same memoryless channel, the variables u^n, e_1, e_2 become $\widetilde{u}^n, \widetilde{e}_1, \widetilde{e}_2$, respectively. The letter u stands for unencoded, and e for encoded. Quantities with a tilde $\widetilde{u}^n, \widetilde{e}_1, \widetilde{e}_2$ occur after channel passage and are visible (measurable). Quantities without a tilde u^n, e_1, e_2 are hidden (unmeasurable).

The situation just described can be represented by the bnet Fig.106.1, or by its abridged version Fig.106.2. But note that the abridged version does not show explicitly that the u_i are i.i.d. or that the channel is memoryless (i.e., that the u_i for all i pass independently through the channel).

Define

$$x = (u^n, e_1, e_2) \tag{106.1}$$

and

$$\widetilde{x} = (\widetilde{u}^n, \widetilde{e}_1, \widetilde{e}_2) \ . \tag{106.2}$$

Fig.106.1 implies that

$$P(x, \widetilde{x}) = P(\widetilde{u}^n | u^n) \left[\prod_{r=1,2} P(\widetilde{e}_r | e_r) P(e_r | u^n) \right] P(u^n) \ . \tag{106.3}$$

Because the u^n are i.i.d.,

$$P(u^n) = \prod_i P(u_i) \ . \tag{106.4}$$

Because the channel is memoryless,

Figure 106.1: Turbo coding Bnet representing a message being encoded two different ways and then the original message and the 2 encodings pass through a memoryless channel.

Figure 106.2: Abridged version of Fig.106.1.

$$P(\widetilde{u}^n|u^n) = \prod_i P(\widetilde{u}_i|u_i) \ . \tag{106.5}$$

Because the encoding is deterministic, we must have for $r = 1, 2$

$$P(e_r|u^n) = \delta(e_r, e_r(u^n)) \ . \tag{106.6}$$

Define the belief functions

$$BEL_i = BEL_i(\underline{u}_i = a) = P(\underline{u}_i = a|\widetilde{x}) \ . \tag{106.7}$$

The best estimate of u_j given all visible evidence $\widetilde{x}$ is

$$\widehat{u}_i = \operatorname*{argmax}_{u_i} BEL_i(u_i) \ . \tag{106.8}$$

Define the probability functions

$$\pi_i = \pi_i(u_i) = P(u_i) \ , \tag{106.9}$$

and the likelihood functions

$$\lambda_i = \lambda_i(u_i) = P(\widetilde{u}_i|u_i) \ . \tag{106.10}$$

For $r = 1, 2$, define the Kernel functions

$$K_r = K_r(u^n) = P(\widetilde{e}_r|e_r = e_r(u^n)) \ . \tag{106.11}$$

In this book, $\mathcal{N}(!a)$ denotes a normalization constant that does not depend on a. Define

$$\mathcal{N}_i = \mathcal{N}(!u_i) \ . \tag{106.12}$$

Claim 194

$$BEL_i = \mathcal{N}_i \lambda_i \pi_i \mathcal{T}_i^{K_1 K_2} [\prod_{j \neq i} \lambda_j \pi_j] \ , \tag{106.13}$$

where $\mathcal{T}_i^K(\cdot)$ with $K = K_1 K_2$ is an operator (transform) that acts on functions of u^n:

$$\mathcal{T}_i^K(\cdot) = \sum_{u^n} \delta(u_i, a) K(u^n)(\cdot) \ . \tag{106.14}$$

proof:

$$\begin{aligned}
P(\underline{u}_i = a|\widetilde{x}) = & \\
= & \sum_x \delta(u_i, a) P(x|\widetilde{x}) && (106.15)\\
= & \sum_x \delta(u_i, a) \frac{P(\widetilde{x}|x)P(x)}{P(\widetilde{x})} && (106.16)\\
= & \mathcal{N}(!a) \sum_x \delta(u_i, a) P(\widetilde{x}|x) P(x) && (106.17)\\
= & \mathcal{N}(!a) \sum_x \delta(u_i, a) P(u^n) \left[\prod_{r=1,2} P(\widetilde{e}_r|e_r) \delta(e_r, e_r(u^n)) \right] \prod_j P(\widetilde{u}_j|u_j) && (106.18)\\
= & \mathcal{N}(!a) \lambda_i(a) \pi_i(a) R \ , && (106.19)
\end{aligned}$$

where

$$
\begin{aligned}
R &= \sum_{u^n} \delta(u_i, a) \left[\prod_{r=1,2} P(\widetilde{e}_r | e_r(u^n)) \right] \prod_{j \neq i} P(\widetilde{u}_j | u_j) P(u_j) && (106.20) \\
&= \sum_{u^n} \delta(u_i, a) \left[\prod_{r=1,2} K_r(u^n) \right] \prod_{j \neq i} \lambda_j(u_j) \pi_j(u_j) && (106.21) \\
&= \mathcal{T}_i^{K_1 K_2} [\prod_{j \neq i} \lambda_j(u_j) \pi_j(u_j)] \ . && (106.22)
\end{aligned}
$$

Hence

$$
BEL_i(a) = \mathcal{N}(!a) \lambda_i(a) \pi_i(a) \mathcal{T}_i^{K_1 K_2} [\prod_{j \neq i} \lambda_j(u_j) \pi_j(u_j)] \ . \tag{106.23}
$$

QED

106.1 Decoding Algorithm

The Turbo algorithm for decoding the encode message is as follows. For $m = 0$, let

$$
\pi_j^{(0)}(u_j) = \frac{1}{n_{\underline{u}_j}} \ . \tag{106.24}
$$

Then for $m = 1, 2, \ldots,$ let

$$
\pi_i^{(m)} = \mathcal{N}_i \mathcal{T}_i^{K_{m\%2}} [\prod_{j \neq i} \lambda_j \pi_j^{(m-1)}] \ , \tag{106.25}
$$

where $m\%2 = 1$ if m is odd and $m\%2 = 2$ if m is even. Furthermore, for $m > 0$, let

$$
\begin{aligned}
BEL_i^{(m)} &= \mathcal{N}_i \lambda_i \pi_i^{(m-1)} \pi_i^{(m)} && (106.26) \\
&= \mathcal{N}_i \lambda_i \pi_i^{(m-1)} \mathcal{T}_i^{K_{m\%2}} [\prod_{j \neq i} \lambda_j \pi_j^{(m-1)}] \ . && (106.27)
\end{aligned}
$$

As $m \to \infty$, $BEL_i^{(m)}$ given by Eq.(106.27) is expected to converge to the exact BEL_i given by Eq.(106.13).

Turbo decoding can be represented by the bnets Figs.106.3 and 106.4.

The TPMs, printed in blue, for bnet Fig.106.3, are as follows.

$$
P(d_i^{(m)} = a \mid \widetilde{u}^n, \widetilde{e}_{m\%2}) = BEL_i^{(m)}(a) \ . \tag{106.28}
$$

The TPMs, printed in blue, for bnet Fig.106.4, are as follows.

Figure 106.3: Bnet describing Turbo code generation of $BEL_i^{(m)}(a)$ for $m = 1, 2, \ldots$.

Figure 106.4: Bnet describing Turbo code generation of $BEL^{n(m)}(\cdot)$ and $\pi^{n(m)}(\cdot)$ for $m = 0, 1, 2 \ldots$. The following arrows were not drawn for clarity: Arrows pointing from node $\underline{\lambda}^n(\cdot)$ to nodes $\underline{\pi}^{n(m)}(\cdot)$ and $\underline{BEL}^{n(m)}(\cdot)$ for $m = 0, 1, 2, \ldots$.

$$P((\lambda^n)'(\cdot)|\widetilde{u}^n) = \delta((\lambda^n)'(\cdot), \lambda^n(\cdot)) \tag{106.29}$$

$$P(\pi^{n(m)}(\cdot)|\lambda^n(\cdot), \pi^{n(m-1)}(\cdot), \widetilde{e}_{m\%2}) = \prod_i \prod_{u_i} \delta(\pi_i^{(m)}(u_i), \mathcal{N}_i \mathcal{T}_i^{K_{m\%2}}[\prod_{j \neq i} \lambda_j \pi_j^{(m-1)}]) \tag{106.30}$$

$$P(BEl^{n(m)}(\cdot)|\lambda^n(\cdot), \pi^{n(m)}(\cdot), \pi^{n(m-1)}(\cdot)) = \prod_i \prod_{u_i} \delta(BEL_i(u_i), \mathcal{N}_i \lambda_i \pi_i^{(m-1)} \pi_i^{(m)}) \tag{106.31}$$

106.2 Message Passing Interpretation of Decoding Algorithm

Ref.[48] shows that the Turbo code decoding algo can be interpreted as an application of Message Passing. We leave all talk of Message Passing to a separate chapter, Chapter 60.

Chapter 107

Turing Machine

This chapter is based on Ref.[191].

In this chapter, we define a **Turing Machine** (TM) as a generalization of a special case of a Petri net called a Finite State Machine (FSM). Petri nets are discussed in Chapter 75. FSM are discussed in Chapter 30. We will assume that the reader has read those two chapters before tackling this one.

Fig.107.1 presents an idealized physical model of a TM. It consists of a

- a **tape**. The tape is infinite in both directions, and separated into tape cells, but only a finite number of tape cells may be non-blank. Each non-blank cell contains one symbol from a pre-defined set of symbols called the tape alphabet.
- a **head** that is one tape cell wide. It moves, relative to the tape, from one tape cell, to the cell immediately to its right or to its left. The head can read the value of the symbol under it and replace it (write) by another symbol.
- a **control/memory** unit that remembers the current place/state ($\underline{p}_2$ in Fig.107.1)

Figure 107.1: Idealized physical model of Turing Machine.

107.1 Example

$\underline{p} \rightarrow \underline{x}$	$\lvert\underline{p}'\rangle = \underline{x}\lvert\underline{p}\rangle$	$\lambda = (\lambda_r, \lambda_w, \lambda_m)$
$\underline{a} \rightarrow \underline{x}_1$	$\underline{b}$	$(0, 1, R)$
$\underline{b} \rightarrow \underline{x}_2$	$\underline{a}$	$(0, 1, L)$
$\underline{a} \rightarrow \underline{x}_3$	$\underline{c}$	$(1, 1, L)$
$\underline{c} \rightarrow \underline{x}_4$	$\underline{b}$	$(0, 1, L)$
$\underline{c} \rightarrow \underline{x}_5$	$\underline{h}$	$(1, 1, R)$
$\underline{b} \rightarrow \underline{x}_6$	$\underline{b}$	$(1, 1, R)$

Figure 107.2: The "3-state Busy Beaver" Turing machine represented as a FSM Petri net.

1					$\underline{a}$					
0	0	0	0	0	$\underline{0}$	0	0	0	0	0
2						$\underline{b}$				
0	0	0	0	0	1	$\underline{0}$	0	0	0	0
3					$\underline{a}$					
0	0	0	0	0	$\underline{1}$	1	0	0	0	0
4				$\underline{c}$						
0	0	0	0	$\underline{0}$	1	1	0	0	0	0
5			$\underline{b}$							
0	0	0	$\underline{0}$	1	1	1	0	0	0	0

Table 107.1: First five steps of Busy Beaver TM defined by Fig.107.2.

Fig.107.2 shows an example of a TM represented as a FSM Petri net. Everything in this diagram should be familiar to the reader after reading the chapters on pnets and FSM, except for the transition labelling function $\lambda()$.

Whereas λ is a scalar (i.e., an element of the alphabet Σ) for a FSM, for a TM it's a triple $(\lambda_r, \lambda_w, \lambda_m)$. λ_r is the tape symbol read, λ_w is the tape symbol written (after erasing the previous one). $\lambda_m \in \{R, L\}$ commands the head to move one cell to the right (R) or to the left (L).

Table 107.1 gives the first five steps of the Busy Beaver.

For the Busy Beaver example:

$\underline{h} = HALT$

$\mathcal{P} = \{\underline{a}, \underline{b}, \underline{c}, \underline{h}\}$

$0 =$ blank symbol

$\Sigma^+ = \{0, 1\}$,
$\Sigma = \{1\}$
$\underline{p}(0) = \underline{a}$
$\overline{\mathcal{P}}_{acc} = \{\underline{h}\}$

107.2 Precise Definition

Same as in FSM:

$\mathcal{X} = \{\underline{x}_i\}_{i=1}^{nx}$ are the **transition nodes** of the pnet.

$\mathcal{P} = \{\underline{p}_i\}_{i=1}^{np}$ are the **place (or state) nodes** of the pnet.

$\underline{p}(0) \in \mathcal{P}$, **starting/initial place**

$\overline{\mathcal{P}}_{acc} \subset \mathcal{P}$ are the **acceptor/final places**.

$\Delta : \mathcal{X} \times \mathcal{P} \to \mathcal{P}$ is the **transition function**

New for TM

$HALT \in \mathcal{P}_{acc}$

$\Sigma^+ =$ **tape alphabet**

$B \in \Sigma^+$ is the symbol for a **blank space**.

$\Sigma =$ **input alphabet**. $\Sigma \subset \Sigma^+ - \{B\}$. This is the set of symbols other than B that occur in the initial tape.

$\lambda_r : \mathcal{X} \to \Sigma^+$ is the **read labelling function**

$\lambda_w : \mathcal{X} \to \Sigma^+$ is the **write labelling function**

$\lambda_m : \mathcal{X} \to \{L, R\}$ is the **move labelling function** L =move Left, R =move Right. Some machines use $\{L, R, N\}$ as the range of λ_m. N is the command "No movement".

$\lambda = (\lambda_r, \lambda_w, \lambda_m)$ is the **transition labelling function**

$\lambda_{\underline{p}} : \mathcal{X}(\underline{p} \to) \to \Sigma^+$ is the **restriction of** $\lambda()$ **to** $\mathcal{X}(\underline{p} \to)$

$\Phi_{TM} = \langle \Phi_{pnet}, \mathcal{P}_{acc}, \Sigma, \Sigma^+, \lambda, \Delta \rangle$ is a **TM**

Chapter 108

Uplift Modelling

This chapter is based on many references, including Ref.[24, 19, 192, 71].

Uphill Modelling (UP) deals with the application of Rubin's Theory of Potential Outcomes (PO) to advertisement and marketing.

PO, which is discussed in Chapter 77, is a subset of Pearl's Causal Inference. Besides UP, other applications of PO theory that are discussed in this book are: Regression Discontinuity (Chapter 82), Difference-in-Differences (Chapter 18) and Synthetic Controls (Chapter 98).

In UP, each **participant person** (i.e., **sample** σ, where

$$\sigma \in \Sigma = \{0, 1, 2, \ldots, nsam - 1\}$$

) is interrogated at two well anticipated, fairly closely spaced times t_0 and t_1 (as opposed to Difference-in-Differences (DID), where t_0 and t_1 might be years apart, and long before the DID analysis is attempted.). In between those two times, a treatment which we will refer to as the **UP diagnostic test** is applied. For example, at times t_0 and t_1, every participant might be asked how important he/she rates climate change on a scale of 1 to 10. In between times t_0 and t_1, every participant might be sent a brochure on climate change.

In UP, as in all other PO applications, each participant σ is in the treated or control groups, but not both. The participants are aware of which of those groups they are in, so they are not "treatment blind".

108.1 UP types

Let $y_t^\sigma \in \mathbb{R}$ for $t = t_0, t_1$ be the treatment response at time t for participant (i.e., sample) σ. (We are using here the same notation as in Chapter 77). Call

$$\delta^\sigma = y_{t_1}^\sigma - y_{t_0}^\sigma \tag{108.1}$$

the **participant uplift** for participant σ. As shown in Fig.108.1, UP classifies participants into 4 **UP-types**: Persuadables, SureThings, LostCauses, and Sleepy-

Figure 108.1: UP diagnostic test can be used to classify all participants of the population into 4 UP-types. This figure assumes $y \in \{0, 1\}$. More generally, $y \in \mathbb{R}$. t represents time. $t = t_0$ corresponds to $d = 0 = untreated$, and $t = t_1$ corresponds to $d = 1 = treated$.

Dogs. The UP-type of a participant depends on the changes that are induced on that participant by an **UP-diagnostic-test**.

- For a **Persuadable** participant, $\delta^\sigma > 0$.
- For a **SleepyDogs** participant, $\delta^\sigma < 0$.
- For a **SureThings** participant, $\delta^\sigma \approx 0$ and $y^\sigma_{t_0}$ is high.
- For a **LostCauses** participant, $\delta^\sigma \approx 0$ and $y^\sigma_{t_0}$ is low.

In general, $x = (x_0, x_1, \ldots, x_{n-1})$ is an n dimensional vector of features x_i. If any of the x_i is a priori continuous, we will assume it has been binned into a finite number of bins. Let $val(\underline{x})$ be the finite set of all feature vectors.

Let $A_x = \{\sigma : x^\sigma \approx x\}$ for $x \in val(\underline{x})$. The set A_x of all samples with a feature vector x^σ close to x is called the x **stratum**.

The **stratum-uplift** is $\delta_x = ACE_x$. Strata can also be classified into the 4 UP-types, depending on the sign and size of their δ_x. A participant may not be typical for his stratum and may have different **participant and stratum UP-types**. For example, he may have positive participant uplift δ^σ and therefore have a Persuadable participant UP-type, but his stratum-uplift δ_x might be negative, so he has the SleepyDogs stratum UP-type.

Advertisers are very interested in finding the Persuadable strata in a population so as to focus their resources on them. For example, UP was used very successfully during the Obama presidential campaigns. Team Obama conducted UP-diagnostic tests much like the climate change one described earlier. This allowed them to identify voters who might be sitting on the fence on whether to vote for Obama or not. Then Team Obama spent the lion share of resources on those fence-sitters.

108.2 Some Relevant Technical Formulas from Chapter 77

Recall the following technical formulae that were proven in Chapter 77:

- Recall Eq.(77.143):

$$ACE = \sum_x P(x) \underbrace{\sum_y y \left[P(y|d=1,x) - P(y|d=0,x)\right]}_{ACE_x} \tag{108.2}$$

  If $\underline{y} \in \{0,1\}$, then

$$\underbrace{ACE_x}_{\delta_x} = \underbrace{P_{\underline{y}|\underline{d},\underline{x}}(1|1,x)}_{Y_x^1} - \underbrace{P_{\underline{y}|\underline{d},\underline{x}}(1|0,x)}_{Y_x^0} \ . \tag{108.3}$$

- Recall Eq.(77.158):

$$\underbrace{\widehat{ACE_x}}_{\delta_x} = \underbrace{\frac{1}{N_x} \sum_{\sigma \in A_x} \frac{d^\sigma y^\sigma}{g_{1|x^\sigma}}}_{Y_x^1} - \underbrace{\frac{1}{N_x} \sum_{\sigma \in A_x} \frac{(1-d^\sigma) y^\sigma}{g_{0|x^\sigma}}}_{Y_x^0} \tag{108.4}$$

108.3 UP Analysis

The input to UP is a PO dataset $DS = \{(\sigma, d^\sigma, x^\sigma, y^\sigma) : \sigma \in \Sigma\}$. where $d^\sigma \in \{0,1\}$, $x^\sigma \in val(\underline{x})$, $y^\sigma \in \mathbb{R}$.

Starting with DS, UP performs the following steps. Fig.108.2 is a pictorial representation of the quantities that are calculated during these steps.

1. Find

$$A_x = \{\sigma : x^\sigma \approx x\} \tag{108.5}$$

   for each observed $x \in val(\underline{x})$. Set $A_x = \emptyset$ for unobserved $x \in val(\underline{x})$.

2. Use Eq.(108.4) to calculate δ_x for each $x \in val(\underline{x})$. Set $\delta_x = 0$ if $A_x = \emptyset$.

3. Partition the set $\{\delta_x : x \in val(\underline{x})\}$ into disjoint bins. Call Δ_c the average δ_x within bin c for $c = 0, 1, \ldots, nc-1$. The class labels c should be assigned so that the sequence of Δ_c is monotonic and non-increasing; i.e.,

$$\Delta_0 \geq \Delta_1 \geq \cdots \geq \Delta_{nc-1} \ . \tag{108.6}$$

Figure 108.2: Pictorial representation of the sequence $\{(\mathcal{X}_c, \Delta_c)\}_{c=0,1,\ldots,nc-1}$.

Now calculate

$$\mathcal{X}_c = \{x : \delta_x \approx \Delta_c\} \tag{108.7}$$

for each c. By the end of this step, we will have calculated $\{(\mathcal{X}_c, \Delta_c)\}_{c=0,1,\ldots,nc-1}$ from $\{(x, \delta_x)\}_{x \in val(\underline{x})}$. We will refer to the $\mathcal{X}_c$ as **strata-bins**. Note that

$$\Delta_c = \frac{1}{|\mathcal{X}_c|} \sum_{x \in \mathcal{X}_c} \delta_x \tag{108.8}$$

$$= \underbrace{\frac{1}{|\mathcal{X}_c|} \sum_{x \in \mathcal{X}_c} Y_x^1}_{Y_c^1} - \underbrace{\frac{1}{|\mathcal{X}_c|} \sum_{x \in \mathcal{X}_c} Y_x^0}_{Y_c^0} \ . \tag{108.9}$$

4. For each c, calculate

$$A_{d,x} = \{\sigma \in A_x : d^\sigma = d\} \tag{108.10}$$

$$\Sigma_{d,c} = \cup_{x \in \mathcal{X}_c} A_{d,x} \tag{108.11}$$

for $d \in \{0, 1\}$ and

$$\Sigma_c = \Sigma_{0,c} \cup \Sigma_{1,j} \ . \tag{108.12}$$

Figure 108.3: Plot of UP results.

Fig.108.3 is a way of plotting the results of UP in an intuitive way.

Another common plot of UP results is called a **Qini curve**. A Qini curve is a plot of (X_c, Y_c) for all c, where

$$X_c = \sum_{c'=0}^{c} \sum_{x \in \mathcal{X}_{c'}} |A_x| \tag{108.13}$$

is the cumulative population (counting the samples in order of decreasing uplift) and

$$Y_c = \sum_{c'=0}^{c} \Delta_{c'} \tag{108.14}$$

is the cumulative uplift. As c increases, a Qini curve rises fast at first and then its slope decreases until at some value c_0, the slope is defined as too small to care. No marketing resources are directed towards participants σ for whom $c > c_0$.[1]

108.4 UP Decision Trees

In this section, we will describe how to build UP decision trees (UP dtrees), and explain why they are needed for UP. This section is based mainly on Ref.[71] (highly recommended).

[1]Let $S_\Delta = \{\sigma : \delta^\sigma \geq \Delta\}$. A Qini curve can also be defined, without stratification, as a plot of (X_Δ, Y_Δ) for all $\Delta > 0$, where $X_\Delta = |S_\Delta|$ and $Y_\Delta = \sum_{\sigma \in S_\Delta} \delta^\sigma$. As Δ decreases, a Qini curve rises fast at first and then its slope decreases.

Generic dtrees are described in Chapter 16. In the discussion below, we will assume that the reader has read that chapter already.

For our previous discussion, we assumed a dataset $DS = \{(\sigma, d^\sigma, x^\sigma, y^\sigma) : \sigma \in \Sigma\}$ and assumed we had a stratifying method to obtain the strata c^σ of each sample σ. In this section, we will assume from the onset an extended dataset $DS^+ = \{(\sigma, c^\sigma, d^\sigma, x^\sigma, y^\sigma) : \sigma \in \Sigma\}$ that contains the strata c^σ for each sample σ. This extended dataset will be used to train a dtree to find the strata of a sample. Thus, the dtree will do the job of the prior stratifying method.

Next, let us describe how to modify the results of Chapter 16 on generic dtrees to the case of UP dtrees. The main difference, as we will explain in detail next, is that we use two dtrees ($d = 0, 1$) instead of one. Each dtree reduces to a LD bnet. The two bnets have the same structure but different branch probabilities. Furthermore, instead of the Information Gain metric used for generic dtrees, we use a different metric, one that connects the branch probabilities of the twin bnets.

For UP, we consider two dtrees, labeled by the treatment dose $d = 0, 1$ (no dose, dose). For each d, let $N_j^d(c, x_j)$ be the number of individuals σ in the population that is exiting question node j, belonging to class c and having $\underline{x}_j = x_j$. From $N_j^d(c, x_j)$, we can define an empirical probability distribution

$$P(c, d, x_j|j) = \frac{N_j^d(c, x_j)}{N_j} \ , \quad \text{where } N_j = \sum_{c,d,x_j} N_j^d(c, x_j) \tag{108.15}$$

Once we have an empirical probability distribution $P(c, d, x_j|j)$ for each node $\underline{x}_j$ and class c, we can define various conditional probabilities, and from those conditional probabilities, we can define various metrics.

In Chapter 16, we used Information Gain (a mutual information) as the SAM (Separation Ability Measure) in SL (Structure Learning) of dtrees (Decision Trees). Information Gain is a bad SAM for SL of UP dtrees, because it knows nothing about $d = 0, 1$. For UP dtrees, we need a SAM specifically designed to separate $d = 0, 1$, and generate classes that are uplift bins (i.e., uplift intervals).

Ref.[71] proposes and studies the following 3 SAMs for doing SL of UP dtrees.

1. **SAM_DD** (DD=Delta Delta)

   For $d \in \{0, 1\}$ and $c, c' \in val(\underline{c})$, define the increments

   $$\partial_d f(d) = f(1) - f(0) \tag{108.16}$$

   and

   $$\partial_{c',c} f(c) = f(c') - f(c) \ . \tag{108.17}$$

   Let

$$\Delta_{c|j} = P(c|j,1) - P(c|j,0) \tag{108.18}$$
$$= \partial_d P(c|j,d) \tag{108.19}$$

$$SAM_DD_j = \max_{c,c'} |\partial_{c',c}\partial_d P(c|j,d)| \tag{108.20}$$
$$= \max_{c,c'} |\partial_{c',c}\Delta_{c|j}| \tag{108.21}$$

2. **SAM_KL** (KL=Kullback Leibler)

$$SAM_KL_j = \left[\sum_{x_j \in val(\underline{x}_j)} P(x_j|j) D_{KL}(P_{\underline{c}|x_j,j,1} \parallel P_{\underline{c}|x_j,j,0})\right] - D_{KL}(P_{\underline{c}|j,1} \parallel P_{\underline{c}|j,0}) \tag{108.22}$$

$$= \begin{cases} \left[\sum_{x_j \in val(\underline{x}_j)} P(x_j|j) \sum_{c \in val(\underline{c})} P(c|x_j,j,1) \ln \frac{P(c|x_j,j,1)}{P(c|x_j,j,0)}\right] \\ -\sum_{c \in val(\underline{c})} P(c|j,1) \ln \frac{P(c|j,1)}{P(c|j,0)} \end{cases} \tag{108.23}$$

SAM_KL_j **can be negative.**

3. **SAM_E** (E=Euclidean)

SAM_E_j is defined the same way as SAM_KL_j except with the KL divergence $D_{KL}(P \parallel Q)$ in SAM_KL replaced by the Euclidean distance squared.

$$D(P,Q) = \sum_x (P(x) - Q(x))^2 \tag{108.24}$$

The intuitive reason for using these quantities as SAMs is that they maximize the change in uplift between successive tree levels, so that the uplift increases as quickly as possible as we descend down the UP tree. In the case of generic dtrees for which we use Information Gain as SAM, we are maximizing the correlation between classes and nodes as we descend down the tree. These two goals are related. In fact, in the limit where the number of control individuals becomes zero, SAM_KL_j and IG_j become the same, as will be shown later.

Next we show that SAM_KL_j satisfies the following 3 axioms[2]

Claim 195 .

[2]We won't show it here, but according to Ref.[71], SAM_E_j also satisfies these 3 axioms, but SAM_DD_j satisfies only the first two.

1. *SAM_KL_j is minimum iff $P(c|x_j, j, 0) = P(c|x_j, j, 1)$ for all c and x_j.*
2. *If $P(c|j, d) = P(c|d)$ for all c, d, then $SAM_KL_j = 0$.*
3. *Suppose $N(d = 0) = 0$ for all nodes $r \in J_0$ (i.e., no control population) and we use the Laplace Correction when warranted. Then*

$$SAM_KL_j = H(\underline{c} : \underline{x}_j|j, 1) \tag{108.25}$$
$$= IG_j \quad \textit{for treated population} . \tag{108.26}$$

proof:

The proof of items 1 and 2 follow by inspection of Eq.(108.23). Item 3 is proven in Claim 196 below.

QED

Let $N_{\underline{c}} = |val(\underline{c})|$. Define the uniform probability distribution

$$U_{\underline{c}}(c) = \frac{1}{N_{\underline{c}}} \tag{108.27}$$

for all $c \in val(\underline{c})$.

Empirical probabilities that are undefined when $N_j^d = 0$ can be "Laplace Corrected" as follows:

$$P(c|j, d) = \begin{cases} \frac{N_j^d(c)}{N_j^d} & \text{if } N_j^d > 0 \\ U_{\underline{c}}(c) & \text{if } N_j^d = 0 \text{ (Laplace Correction)} \end{cases} \tag{108.28}$$

Claim 196 *Suppose $N_r(0) = 0$ for all dtree nodes $r \in J_0$ and we use the Laplace Correction when warranted. Then*

$$SAM_KL_j = H(\underline{c} : \underline{x}_j|j, 1) . \tag{108.29}$$

proof:

For all nodes $r \in J_0$, we must have

$$P_{\underline{c}|r,0} = U_{\underline{c}} \tag{108.30}$$

so

$$D_{KL}(P_{\underline{c}|r,1} \parallel P_{\underline{c}|r,0}) = D_{KL}(P_{\underline{c}|r,1} \parallel U_{\underline{c}}) \tag{108.31}$$
$$= \ln(N_{\underline{c}}) - H(\underline{c}|r, 1) . \tag{108.32}$$

For all $x_j \in val(\underline{x}_j)$, we must also have

$$N_j = N_j^1, \quad N_j(x_j) = N_j^1(x_j) \tag{108.33}$$

so

$$P(x_j|j) = P(x_j|j,1) \ . \tag{108.34}$$

Now using Eqs.(108.32) and (108.34), we get

$$\begin{aligned}
SAM_KL_j &= -\left[\sum_{x_j \in val(\underline{x}_j)} P(x_j|j) H(\underline{c}|x_j,j,1)\right] + H(\underline{c}|j,1) && (108.35)\\
&= -\left[\sum_{x_j \in val(\underline{x}_j)} P(x_j|j,1) H(\underline{c}|x_j,j,1)\right] + H(\underline{c}|j,1) && (108.36)\\
&= -H(\underline{c}|\underline{x}_j,j,1) + H(\underline{c}|j,1) && (108.37)\\
&= H(\underline{c} : \underline{x}_j|j,1) && (108.38)
\end{aligned}$$

QED

108.4.1 Appendix, connection between Δ_c and $\Delta_{c|j}$

Recall Eq.(108.9):

$$\begin{aligned}
\Delta_c &= \underbrace{\frac{1}{|\mathcal{X}_c|}\sum_{x\in\mathcal{X}_c} Y_x^1}_{Y_c^1} - \underbrace{\frac{1}{|\mathcal{X}_c|}\sum_{x\in\mathcal{X}_c} Y_x^0}_{Y_c^0} && (108.39)\\
&= \partial_d Y_c^d \ . && (108.40)
\end{aligned}$$

Compare that to Eq.(108.19):

$$\begin{aligned}
\Delta_{c|j} &= P(c|j,1) - P(c|j,0) && (108.41)\\
&= \partial_d P(c|j,d) && (108.42)
\end{aligned}$$

What is the connection between these 2 deltas, Δ_c and $\Delta_{c|j}$? Are they equal?

First off, notice that $\Delta_{c|j}$ is defined for all nodes j of the dtree. Let $j(c)$ be the leaf node for which $\Delta_c \approx \Delta_{c|j(c)}$. Assume $y^\sigma \in \{0,1\}$. Then

$$P(c|j=j(c),d) = \frac{N_{j(c)}^d(c)}{N_{j(c)}^d} \approx Y_c^d \tag{108.43}$$

So the two deltas are indeed approximately equal when $y^\sigma \in \{0,1\}$ and $j = j(c)$.

Chapter 109

Variational Bayesian Approximation for Medical Diagnosis

This chapter is based on Ref.[34].

A Variational Bayesian Approximation (VBA) is when we approximate a probability distribution by another probability distribution that depends on a continuous "variational parameter". This parameter is adjusted within its range of possible values, to make the approximation as good as possible. There are many VBA methods. VBA methods are inspired by ancient methods used in Calculus of Variations applied to Physics and Engineering problems.

In this chapter, we do VBA via Jensen's inequality and convex/concave dual functions.

Ref.[34], on which this chapter is based, applies VBA methods to the problem of diagnostic inference using the Quick Medical Reference (QMR) bipartite Bayesian Network. According to Ref.[34] the maximal clique size of the QMR bnet is 150 nodes, which rules out exact methods of inference like the Junction Tree Algorithm (see Chapter 46). For such high complexity cases, one is forced to use either a VBA or a Monte Carlo method.

Figure 109.1: Typical bnet (bipartite, 2 level graph) for medical diagnosis to which we will apply VBA methods. In this case, $nd = 3$ and $ns = 5$. According to Ref.[34], for QMR, $nd \approx 600$ and $ns \approx 4000$.

Fig.109.1 gives a typical bnet for medical diagnosis to which we will apply VBA methods. $\underline{d}_i \in \{0,1\}$ for $i = 1, 2, \ldots, nd$ are the possible diseases, $\underline{s}_\sigma \in \{0,1\}$ for $\sigma = 1, 2, \ldots, ns$ are the possible symptoms, and $\underline{\ell} \in \{0,1\}$ is the leakage due to possible error in the parents of the symptoms. Note that the arrows point from diseases to symptoms because diseases precede in time the symptoms.

Let

$$\mathbb{Z}_{[1,n]} = \{1, 2, \ldots, n\} \tag{109.1}$$

$$pa_\sigma = \{i \in \mathbb{Z}_{[1,nd]} : \underline{d}_i \in pa(\underline{s}_\sigma)\} = \text{ parents of } \underline{s}_\sigma \tag{109.2}$$

Note that pa_σ does not include $\underline{\ell}$, which is also a parent of $\underline{s}_\sigma$.

$$ch_i = \{\sigma \in \mathbb{Z}_{[1,ns]} : \underline{s}_\sigma \in ch(\underline{d}_i)\} = \text{ children of } \underline{d}_i \tag{109.3}$$

$$\underline{d}_A = \{\underline{d}_k : k \in A\} \tag{109.4}$$

$$\underline{d}^{nd} = \{\underline{d}_k : k \in \mathbb{Z}_{[1,nd]}\} \tag{109.5}$$

$$\underline{d}_{!j} = \{\underline{d}_k : k \in \mathbb{Z}_{[1,nd]} - \{j\}\} \tag{109.6}$$

The TPMs, printed in blue, for the bnet Fig.109.1, are as follows:

$$P(d_j) = \text{ given} \tag{109.7}$$

$$P(\ell) = \text{ given} \tag{109.8}$$

$$P(\underline{s}_\sigma = 0 | d_{pa_\sigma}, \ell) = \underbrace{P(\underline{s}_\sigma = 0|\ell)}_{e^{-\theta_{\sigma|0}}} \prod_{j \in pa_\sigma} \underbrace{P(\underline{s}_\sigma = 0|d_j)}_{e^{-\theta_{\sigma|j} d_j}} \tag{109.9}$$

$$= e^{-\theta_{\sigma|0} - \sum_{j \in pa_\sigma} \theta_{\sigma|j} d_j} \tag{109.10}$$

where $\theta_{\sigma|0}, \theta_{\sigma|j} > 0$. This $P(\underline{s}_\sigma = 0|d_{pa_\sigma}, \ell)$ corresponds to the noisy-or model (See Chapter 69.).

$$P(\underline{s}_\sigma = 1|d_{pa_\sigma}) = 1 - e^{-\theta_{\sigma|0} - \sum_{j \in pa_\sigma} \theta_{\sigma|j} d_j} \tag{109.11}$$

Define

$$x_\sigma = \theta_{\sigma|0} + \sum_{j \in pa_\sigma} \theta_{\sigma|j} d_j \tag{109.12}$$

Suppose $A \subset \mathbb{Z}_{[1,ns]}$, $A^c = \mathbb{Z}_{[1,ns]} - A$. Then

$$P(d_j|\underline{s}_A = 0, \underline{s}_{A^c} = 1) = \frac{P(\underline{s}_A = 0, \underline{s}_{A^c} = 1|d_j)P(d_j)}{P(\underline{s}_A = 0, \underline{s}_{A^c} = 1)} \tag{109.13}$$

$$= \mathcal{N}(!d_j) \sum_{d_{!j}} P(\underline{s}_A = 0, \underline{s}_{A^c} = 1|d^{nd})P(d^{nd}) \tag{109.14}$$

$$= \mathcal{N}(!d_j) \sum_{d_{!j}} P(\underline{s}_{A^c} = 1|d^{nd})P(\underline{s}_A = 0|d^{nd})P(d^{nd}) \tag{109.15}$$

$$= \mathcal{N}(!d_j) \sum_{d_{!j}} \begin{cases} \prod_{\sigma \in A^c} \overbrace{(1 - e^{-x_\sigma})}^{\text{call } \mathcal{P}_\sigma} \\ \prod_{\sigma \in A} e^{-\theta_{\sigma|0}} \prod_{j \in pa_\sigma} [e^{-\theta_{\sigma|j}}]^{d_j} \\ \prod_{j \in \mathbb{Z}_{[1,nd]}} P(d_j) \end{cases} \tag{109.16}$$

Summing over $d_{!j}$ seems crazy, because $nd >> 1$, but we will approximate the summand so that the sum can be done in closed form.

Define

$$f(x_\sigma) = \ln(1 - e^{-x_\sigma}) \tag{109.17}$$

and

$$\mathcal{P}_\sigma = 1 - e^{-x_\sigma} = e^{f(x_\sigma)} \tag{109.18}$$

$f(x_\sigma)$ is a concave function. See Fig.C.19 for a plot of it.

Next, we shall find a lower and upper bound for $\mathcal{P}_\sigma$.

To find the upper bound, we will use dual functions, which are discussed in Section C.47.

Let $\widetilde{f}(p_\sigma)$ be the dual function of $f(x_\sigma) = \ln(1 - e^{-x_\sigma})$. In Section C.47, we show that

$$\widetilde{f}(p_\sigma) = \min_{x_\sigma}(x_\sigma p_\sigma - f(x_\sigma)) \tag{109.19}$$

$$= -p_\sigma \ln p_\sigma + (1 + p_\sigma) \ln(1 + p_\sigma) \tag{109.20}$$

and

$$f(x_\sigma) \leq x_\sigma p_\sigma - \widetilde{f}(p_\sigma) \ . \tag{109.21}$$

Therefore

$$\mathcal{P}_\sigma = e^{f(x_\sigma)} \tag{109.22}$$

$$\leq e^{x_\sigma p_\sigma - \widetilde{f}(p_\sigma)} \tag{109.23}$$

$$= \underbrace{e^{-\widetilde{f}(p_\sigma)} e^{-\theta_{\sigma|0} p_\sigma} \prod_{j \in pa_\sigma} [e^{-\theta_{\sigma|j} p_\sigma}]^{d_j}}_{\text{call } \mathcal{B}(p_\sigma)} \tag{109.24}$$

To find a lower bound for P_σ, we will use Jensen's inequality, which is discussed in Section C.42. Let $q_{j|\sigma} \in [0,1]$ satisfy $\sum_j q_{j|\sigma} = 1$. Then

$$\begin{aligned}
\mathcal{P}_\sigma &= e^{f(x_\sigma)} && (109.25)\\
&= e^{f\left(\theta_{\sigma|0}+\sum_j \theta_{\sigma|j} d_j\right)} && (109.26)\\
&= e^{f\left(\theta_{\sigma|0}+\sum_j q_{j|\sigma} \frac{\theta_{\sigma|j} d_j}{q_{j|\sigma}}\right)} && (109.27)\\
&\geq e^{\sum_j q_{j|\sigma} f\left(\theta_{\sigma|0}+\frac{\theta_{\sigma|j} d_j}{q_{j|\sigma}}\right)} && (109.28)\\
&= e^{\sum_j q_{j|\sigma}\left[d_j f\left(\theta_{\sigma|0}+\frac{\theta_{\sigma|j} d_j}{q_{j|\sigma}}\right)+(1-d_j) f(\theta_{\sigma|0})\right]} && (109.29)\\
&= \underbrace{e^{f(\theta_{\sigma|0})+\sum_j q_{j|\sigma} d_j\left[f\left(\theta_{\sigma|0}+\frac{\theta_{\sigma|j} d_j}{q_{j|\sigma}}\right)-f(\theta_{\sigma|0})\right]}}_{\text{call } \mathcal{A}(q_{\cdot|\sigma})} && (109.30)
\end{aligned}$$

In conclusion,

$$\boxed{\mathcal{A}(q_{\cdot|\sigma}) \leq \mathcal{P}_\sigma \leq \mathcal{B}(p_\sigma)} \tag{109.31}$$

with variational parameters $q_{\cdot|\sigma}$ and p_σ.

Chapter 110

Variational Bayesian Approximation via D_{KL}

For more info and references about this topic, see Ref.[193].

A Variational Bayesian Approximation (VBA) is when we approximate a probability distribution by another probability distribution that depends on a continuous "variational parameter". This parameter is adjusted within its range of possible values, to make the approximation as good as possible. There are many VBA methods. VBA methods are inspired by ancient methods used in Calculus of Variations applied to Physics and Engineering problems.

In this chapter, we do VBA via the Kullback-Leibler divergence D_{KL}. We approximate the probability distribution $P(h|\vec{x})$, where h are the hidden variables and $\vec{x}$ is the data. More precisely, suppose $\underline{h} \in val(\underline{h})$ and $\underline{q} \in val(\underline{h})$. Suppose $\underline{\vec{x}} \in val(\underline{x})^{nsam}$ is a vector of $nsam$ samples and the samples $\underline{x}[\sigma] \in val(\underline{x})$ are i.i.d.. The VBA is simply an approximation $P_{\underline{q}|\underline{\vec{x}}}$ to $P_{\underline{h}|\underline{\vec{x}}}$:

$$P_{\underline{h}|\underline{\vec{x}}}(h|\vec{x}) \approx P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x}) \tag{110.1}$$

obtained by minimizing the Kullback-Leibler divergence $D_{KL}(P_{\underline{q}|\underline{\vec{x}}} \parallel P_{\underline{h}|\underline{\vec{x}}})$ over all $P_{\underline{q}|\underline{\vec{x}}}$. The minimization is usually subject to some constraints on the admissible forms of $P_{\underline{q}|\underline{\vec{x}}}$.

$D_{KL}(Q \parallel P) \neq D_{KL}(P \parallel Q)$; i.e., D_{KL} is not symmetric. So why do we use $D_{KL}(P_{\underline{q}|\underline{\vec{x}}} \parallel P_{\underline{h}|\underline{\vec{x}}})$ instead of $D_{KL}(P_{\underline{h}|\underline{\vec{x}}} \parallel P_{\underline{q}|\underline{\vec{x}}})$? Because $D_{KL}(P_{\underline{h}|\underline{\vec{x}}} \parallel P_{\underline{q}|\underline{\vec{x}}})$ requires knowledge of $P_{\underline{h}|\underline{\vec{x}}}$, but calculating $P_{\underline{h}|\underline{\vec{x}}}$ is what we are trying to do in the first place.

See Fig.110.1 for some intuition on what minimizing $D_{KL}(P_{\underline{q}|\underline{\vec{x}}} \parallel P_{\underline{h}|\underline{\vec{x}}})$ means.

Suppose $\underline{h} = (\underline{h}_0, \underline{h}_1, \ldots, \underline{h}_{nh-1})$ and $\underline{q} = (\underline{q}_0, \underline{q}_1, \ldots, \underline{q}_{nh-1})$ where $\underline{h}_i \in val(\underline{h}_i)$ and $\underline{q}_i \in val(\underline{h}_i)$ for all i. We say $\underline{q}$ and $\underline{h}$ have nh mirroring components and those of $\underline{q}$ are independent at fixed $\underline{\vec{x}}$ if

$$P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x}) = \prod_i P_{\underline{q}_i|\underline{\vec{x}}}(h_i|\vec{x}) \ . \tag{110.2}$$

The bnet Fig.110.2 describes the scenario that we have in mind: The samples $\underline{x}[\sigma]$ are

Figure 110.1: If $P_{\underline{q}}(h)$ is Gaussian shaped and $P_{\underline{h}}(h)$ has multiple bumps (modes) then $D_{KL}(P_{\underline{q}} \parallel P_{\underline{h}})$ is minimized when $P_{\underline{q}}$ fits one of the modes of $P_{\underline{h}}$. That is because $D_{KL}(P_{\underline{q}} \parallel P_{\underline{h}}) = \sum_h P_{\underline{q}}(h) \ln \frac{P_{\underline{q}}(h)}{P_{\underline{h}}(h)}$ is a weighted average with weights $P_{\underline{q}}$, so nothing going on outside the support of $P_{\underline{q}}$ influences much the final average.

Figure 110.2: $\underline{q}$ and $\underline{h}$ have $nh = 2$ mirroring components and those of $\underline{q}$ are independent at fixed $\underline{\vec{x}}$.

i.i.d.. Each component $\underline{h}_i$ of $\underline{h}$ has a mirroring component $\underline{q}_i$ in $\underline{q}$. The components of $\underline{h}$ are correlated whereas those of $\underline{q}$ are independent at fixed $\underline{\vec{x}}$.

Claim 197 *If $\underline{q}$ and $\underline{h}$ have nh mirroring components and those of $\underline{q}$ are independent at fixed $\underline{\vec{x}}$ and $D_{KL}(P_{\underline{q}|\underline{\vec{x}}} \parallel P_{\underline{h}|\underline{\vec{x}}})$ is minimum over all $P_{\underline{q}|\underline{\vec{x}}}$, then*

$$P_{\underline{q}_i|\underline{\vec{x}}}(q_i|\vec{x}) = \mathcal{N}(!q_i) e^{E_{(\underline{q}_j)_{j\neq i}}[\ln P_{\underline{h}|\underline{\vec{x}}}(\underline{h}=q|\vec{x})]} \tag{110.3}$$

$$= \mathcal{N}(!q_i) e^{E_{(\underline{q}_j)_{j\neq i}}[\ln P_{\underline{h},\underline{\vec{x}}}(\underline{h}=q,\vec{x})]} \tag{110.4}$$

for all i.

proof:

Since all quantities in Eq.(110.3) are conditioned on $\vec{x}$, let us omit all mention of $\vec{x}$ in this proof.

Let

$$\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_1 \tag{110.5}$$

where

$$\begin{aligned}
\mathcal{L}_0 &= D_{KL}(P_{\underline{q}} \parallel P_{\underline{h}}) && (110.6)\\
&= \sum_h P_{\underline{q}}(h) \ln \frac{P_{\underline{q}}(h)}{P_{\underline{h}}(h)} && (110.7)\\
&= \sum_h P_{\underline{q}}(h) \ln P_{\underline{q}}(h) - \sum_h P_{\underline{q}}(h) \ln P_{\underline{h}}(h) && (110.8)\\
&= \sum_i \sum_{h_i} P_{\underline{q}_i}(h_i) \ln P_{\underline{q}_i}(h_i) - \sum_h P_{\underline{q}}(h) \ln P_{\underline{h}}(h) && (110.9)
\end{aligned}$$

and

$$\mathcal{L}_1 = \sum_i \lambda_i \left[\sum_{h_i} P_{\underline{q}_i}(h_i) - 1 \right] . \tag{110.10}$$

Then

$$\delta\mathcal{L} = \sum_i \sum_{h_i} \delta P_{\underline{q}_i}(h_i) \left[\ln P_{\underline{q}_i}(h_i) + 1 + \lambda_i - \frac{1}{nh} \sum_{(h_j)_{j\neq i}} \prod_{(h_j)_{j\neq i}} \{P_{\underline{q}_j}(h_j)\} \ln P_{\underline{h}}(h) \right] . \tag{110.11}$$

Hence,

$$P_{\underline{q}_i}(h_i) = \mathcal{N}(!h_i) e^{\sum_{(h_j)_{j\neq i}} \left\{ \prod_{(h_j)_{j\neq i}} P_{\underline{q}_j}(h_j) \right\} \ln P_{\underline{h}}(h)} . \tag{110.12}$$

QED

Note that Eq.(110.3) yields a system of nh nonlinear equations in nh unknowns $(P_{\underline{q}_i|\underline{\vec{x}}})_{i=0,1,\ldots,nh-1}$. This system is usually solved recursively.

110.1 Free Energy $\mathcal{F}(\vec{x})$

To simplify the notation below, let us introduce the following abbreviations:

$$P(h|\vec{x}) = P_{\underline{h}|\underline{\vec{x}}}(h|\vec{x}) \tag{110.13}$$

$$P(h, \vec{x}) = P_{\underline{h},\underline{\vec{x}}}(h, \vec{x}) \tag{110.14}$$

$$P(\vec{x}) = P_{\underline{\vec{x}}}(\vec{x}) \tag{110.15}$$

Note that

$$\begin{aligned} D_{KL}(P_{\underline{q}|\underline{\vec{x}}} \parallel P_{\underline{h}|\underline{\vec{x}}}) &= \sum_h P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x}) \ln \frac{P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x})}{P(h|\vec{x})} && (110.16) \\ &= \sum_h P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x}) \ln \frac{P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x})}{P(h, \vec{x})} + \ln P(\vec{x}) && (110.17) \\ &= \mathcal{F}(\vec{x}) + \ln P(\vec{x}) && (110.18) \end{aligned}$$

Hence, the **Free energy** $\mathcal{F}(\vec{x})$ is defined as

$$\begin{aligned} \mathcal{F}(\vec{x}) &= \sum_h P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x}) \ln \frac{P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x})}{P(h, \vec{x})} && (110.19) \\ &= E_{\underline{q}|\underline{\vec{x}}} \left[\ln \frac{P_{\underline{q}|\underline{\vec{x}}}(q|\vec{x})}{P_{\underline{h},\underline{\vec{x}}}(q, \vec{x})} \right] . && (110.20) \end{aligned}$$

The name free energy is justified because

$$\mathcal{F}(\vec{x}) = \underbrace{-\sum_h P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x}) \ln P_{\underline{h},\underline{\vec{x}}}(h, \vec{x})}_{U\text{, Internal Energy}} + \underbrace{\sum_h P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x}) \ln P_{\underline{q}|\underline{\vec{x}}}(h|\vec{x})}_{-S\text{, minus Entropy}} . \tag{110.21}$$

It is also common to define a quantity called "ELBO" to be the negative of the free energy.

$$ELBO(\vec{x}) = -\mathcal{F}(\vec{x}) \tag{110.22}$$

ELBO stands for "Evidence Lower BOund". That name is justified because

$$\underbrace{\ln P_{\underline{\vec{x}}}(\vec{x})}_{\text{evidence}\leq 0} = \underbrace{D_{KL}(P_{\underline{q}|\underline{\vec{x}}} \parallel P_{\underline{h}|\underline{\vec{x}}})}_{\geq 0} - |ELBO(\vec{x})| . \tag{110.23}$$

Some properties of $\mathcal{F}$ are:

- **$\mathcal{F}$ is non-negative.**

$$\underbrace{D_{KL}(P_{\underline{q}|\underline{\vec{x}}} \parallel P_{\underline{h}|\underline{\vec{x}}})}_{\geq 0} + \underbrace{\ln \frac{1}{P_{\underline{\vec{x}}}(\vec{x})]}}_{\geq 0} = \mathcal{F}(\vec{x}) \tag{110.24}$$

Figure 110.3: $D_{KL} + \ln \frac{1}{P(\vec{x})} = \mathcal{F}$.

- **KL divergence is min iff $\mathcal{F}$ is min at fixed $P(\vec{x})$.**

  During a variation δ that holds $P(\vec{x})$ fixed, the KL divergence and $\mathcal{F}$ change by the same amount:

$$\delta D_{KL}(P_{\underline{q}|\underline{\vec{x}}} \parallel P_{\underline{h}|\underline{\vec{x}}}) = \delta \mathcal{F}(\vec{x}) \tag{110.25}$$

Chapter 111

XGBoost

This chapter is based on the original XGBoost paper Ref.[11] and the excellent StatQuest videos [79].

Extreme Gradient Boosting (XGBoost) (Chen, Guestrin, 2016) improves gradient boosting (Friedman, 1999) in a number of ways, such as by using a quadratic rather than linear approximation for the variational function.

In XGBoost, one calculates a sequence of functions where each function tries to correct the errors of the previous function. Then a series (i.e., linear combination) of those functions yields an estimate $\widehat{y}^\sigma$ of the target attribute y^σ. As will be shown in this chapter, XGBoost can be used in two cases: Regression (continuous target attribute) and Binary Classification (binary target attribute)[1]. An implementation of the XGBoost algorithm is available as open source. It's written in C++, with Python and R interfaces.

Boosting (see this chapter on XGBoost and Chapter 1 on AdaBoost) and bagging (see Chapter 80 on Random Forest) are two methods of building a classifier function from an ensemble of classifier functions. These two methods are most commonly applied to dtrees (see Chapter 16 on Decision Trees): Boosting for an ensemble of small dtrees, and Bagging for a random forest (which is an ensemble of dtrees that are usually much more complicated than small dtrees).

111.1 Divergences

To set up a cost function for XGBoost, we begin by defining 2 types of "divergences" and calculating the first and second derivatives of those divergences:

- **Divergence for regression** (i.e., continuous classification, continuous target attribute). For $x, y \in \mathbb{R}$

$$D_{reg}(x, y) = \frac{1}{2}(x - y)^2 \tag{111.1}$$

[1]XGBoost can also be used for classification into more than two classes, as long as a suitable Divergence function can be defined.

- **Divergence for binary classification** (i.e., binary target attribute). For $p, q \in [0, 1]$.

$$\begin{aligned} D_{bc}(p, q) &= -\{p \ln q + (1-p)\ln(1-q)\} && (111.2) \\ &= CE(\{p, 1-p\} \parallel \{q, 1-q\}) && (111.3) \end{aligned}$$

The cross-entropy $CE()$ is defined in Chapter C.

Claim 198

$$\partial_{\widehat{y}} D_{reg}(y, \widehat{y}) = \widehat{y} - y \tag{111.4}$$

$$\partial^2_{\widehat{y}} D_{reg}(y, \widehat{y}) = 1 \tag{111.5}$$

$$D(y, \widehat{y} + h) \approx D(y, \widehat{y}) + (\widehat{y} - y)h + \frac{1}{2}h^2 \tag{111.6}$$

proof: Obvious.
QED

Recall from Section C.23 in Chapter C that

$$\widehat{y} = \text{smoid}(\text{lodds}(\widehat{y})) \tag{111.7}$$

Let us abbreviate $s() = \text{smoid}()$ and $l() = \text{lodds}()$ so

$$\widehat{y} = s(l(\widehat{y}))\ . \tag{111.8}$$

Claim 199 [2]

$$\partial_l D(y, \widehat{y}(l)) = \widehat{y} - y \tag{111.9}$$

$$\partial^2_l D(y, \widehat{y}(l)) = \widehat{y}(1 - \widehat{y}) \tag{111.10}$$

$$D(y, \widehat{y}(l + \Delta l)) \approx D(y, \widehat{y}(l)) + (\widehat{y} - y)\Delta l + \frac{1}{2}\widehat{y}(1 - \widehat{y})(\Delta l)^2 \tag{111.11}$$

proof:

[2] Note that $D_{bc}(y, \widehat{y}) \neq D_{bc}(\widehat{y}, y)$; i.e., D_{bc} is not symmetric in its two arguments. Normally $0 < \widehat{y} < 1$ and $y \in \{0, 1\}$. Since the log is applied only to the second argument, to avoid logs of zero, it is better to have $\widehat{y}$ as the second argument.

$$\frac{\partial D(y,s)}{\partial s} = -\partial_s\{y\ln s + (1-y)\ln(1-s)\} \tag{111.12}$$

$$= -\frac{y}{s} + \frac{1-y}{1-s} \tag{111.13}$$

$$= \frac{-y(1-s)+(1-y)s}{s(1-s)} \tag{111.14}$$

$$= \frac{-y+s}{s(1-s)} \tag{111.15}$$

Recall that $\text{smoid}'(l) = \text{smoid}(l)[1-\text{smoid}(l)]$ so

$$\frac{\partial s}{\partial l} = s(1-s)\ . \tag{111.16}$$

$$\frac{\partial D(y,s)}{\partial l} = \frac{\partial s}{\partial l}\ \frac{\partial D(y,s)}{\partial s} = -y+s = -y+\widehat{y} \tag{111.17}$$

$$\frac{\partial^2 D(y,s)}{\partial l^2} = \frac{\partial s}{\partial l} = s(1-s) \tag{111.18}$$

QED

111.2 Minimizing Cost function for single tree

Let

$\sigma \in \Sigma$ be an individual in a population Σ
$x^\sigma \in val(\underline{x})$ be a feature vector $x^\sigma = (x_i^\sigma)_{i=0,1,\ldots,nf-1}$
$t \in \{0,1,\ldots,nt-1\}$ be the tree index
$\mathcal{L}_t$ be the set of leafs of tree t
$\ell \in \mathcal{L}_t$ be a leaf in tree t
$w_t^\ell \in \mathbb{R}$
$f_t : val(\underline{x}) \to \mathbb{R}$
$\ell_t : \Sigma \to \mathcal{L}_t$, $\ell_t(\Sigma) = \mathcal{L}_t$.
$\ell_t(\sigma)$ be the leaf of individual σ in tree t
$\Sigma_t^\ell = \{\sigma \in \Sigma : \ell_t(\sigma) = \ell\}$
Define the function f_t by

$$f_t(x^\sigma) = \sum_{\ell\in\mathcal{L}_t} w_t^\ell \mathbb{1}(\sigma \in \Sigma_t^\ell) = w_t^{\ell_t(\sigma)}\ . \tag{111.19}$$

$f_t(x^\sigma)$ gives the output value for tree t and feature vector x^σ.

Define the **cost function** for tree t as follows

$$\mathcal{C}_t = \sum_\sigma D(\widehat{y}_t^\sigma, y^\sigma) + \underbrace{\sum_{\ell \in \mathcal{L}_t} \left[\gamma + \frac{\lambda}{2} (w_t^\ell)^2 \right]}_{\text{regulator}} , \tag{111.20}$$

where $\gamma > 0, \lambda > 0$ are regulator parameters.

The estimate $\widehat{y}_t^\sigma$, using trees from 0 to t, of the target attribute y^σ, is defined by

$$\begin{aligned}
\widehat{y}_0^\sigma &= f_0(x^\sigma) = \text{arbitrary constant, XGBoost default for this is 0.5} && (111.21)\\
\widehat{y}_1^\sigma &= f_0(x^\sigma) + f_1(x^\sigma) && (111.22)\\
\widehat{y}_2^\sigma &= f_0(x^\sigma) + f_1(x^\sigma) + f_2(x^\sigma) && (111.23)\\
&\vdots && (111.24)\\
\widehat{y}_t^\sigma &= \sum_{t' \le t} f_{t'}(x^\sigma) = \widehat{y}_{t-1}^\sigma + f_t(x^\sigma) && (111.25)
\end{aligned}$$

In the cost function $\mathcal{C}_t$, we approximate the divergence $D()$ by a second order Taylor approximation:

$$\begin{aligned}
D(y^\sigma, \widehat{y}_t^\sigma) &= D(y^\sigma, \widehat{y}_{t-1}^\sigma + \underbrace{f_t(x^\sigma)}_{\delta}) && (111.26)\\
&\approx D(y^\sigma, \widehat{y}_{t-1}^\sigma) + a_t^\sigma \delta + \frac{1}{2} b_t^\sigma \delta^2 && (111.27)\\
&= D(y^\sigma, \widehat{y}_{t-1}^\sigma) + a_t^\sigma w_t^{\ell_t(\sigma)} + \frac{1}{2} b_t^\sigma (w_t^{\ell_t(\sigma)})^2 , && (111.28)
\end{aligned}$$

where

$$a_t^\sigma = [\partial_{\widehat{y}} D(y^\sigma, \widehat{y})]_{\widehat{y} = \widehat{y}_{t-1}^\sigma} , \tag{111.29}$$

$$b_t^\sigma = [\partial_{\widehat{y}}^2 D(y^\sigma, \widehat{y})]_{\widehat{y} = \widehat{y}_{t-1}^\sigma} . \tag{111.30}$$

a_t^σ is called g for gradient and b_t^σ is called h for Hessian in Ref.[11]. Table 111.1gives the values for a_t^σ and b_t^σ for Regression (reg) and Binary classification (bc).

Define the **residual** for tree t and individual σ by

$$r_t^\sigma = y^\sigma - \widehat{y}_{t-1}^\sigma . \tag{111.31}$$

Note that

$$\sum_\sigma = \sum_{\ell \in \mathcal{L}_t} \sum_{\sigma \in \Sigma_t^\ell} . \tag{111.32}$$

Define

	Regression $(y^\sigma \in \mathbb{R}, \widehat{y}_t^\sigma \in \mathbb{R},\ D = D_{reg})$	Binary Classification $(y^\sigma \in \{0,1\},\ \widehat{y}_t^\sigma \in [0,1],\ D = D_{bc})$
a_t^σ	$\widehat{y}_{t-1}^\sigma - y^\sigma =$ neg.residual	$\widehat{y}_{t-1}^\sigma - y^\sigma =$ neg.residual
b_t^σ	1	$\widehat{y}_{t-1}^\sigma(1-\widehat{y}_{t-1}^\sigma)$

Table 111.1: The first (a_t^σ) and second (b_t^σ) derivatives for Regression (reg) and Binary classification (bc).

$$A_t^\ell = \sum_{\sigma \in \Sigma_t^\ell} a_t^\sigma \tag{111.33}$$

and

$$B_t^\ell = \sum_{\sigma \in \Sigma_t^\ell} b_t^\sigma \ . \tag{111.34}$$

Now we can rewrite the cost function as

$$\begin{aligned} \mathcal{C}_t &= \underbrace{\sum_\sigma D(\widehat{y}_{t-1}^\sigma, y^\sigma)}_{\mathcal{K}} + \sum_{\ell \in \mathcal{L}_t} \left[A_t^\ell w_t^\ell + \frac{1}{2} B_t^\ell (w_t^\ell)^2 \right] + \sum_{\ell \in \mathcal{L}_t} \left[\gamma + \frac{\lambda}{2} (w_t^\ell)^2 \right] && (111.35) \\ &= \sum_{\ell \in \mathcal{L}_t} \left[\gamma + A_t^\ell w_t^\ell + \frac{1}{2}(B_t^\ell + \lambda)(w_t^\ell)^2 \right] \quad \text{(absorbed } \mathcal{K} \text{ into } \gamma) && (111.36) \end{aligned}$$

The cost function $\mathcal{C}_t$ can be minimized over w_t^ℓ:

$$0 = \delta \mathcal{C}_t = \sum_{\ell \in \mathcal{L}_t} \delta w_t^\ell \left[A_t^\ell + (B_t^\ell + \lambda) w_t^\ell \right] \tag{111.37}$$

Therefore, the cost is minimized when

$$w_t^\ell = \frac{-A_t^\ell}{B_t^\ell + \lambda} \ . \tag{111.38}$$

The optimum cost is

$$\mathcal{C}_t = \sum_{\ell \in \mathcal{L}_t} \underbrace{\left[\gamma - \underbrace{\frac{(A_t^\ell)^2}{2(B_t^\ell + \lambda)}}_{SS_t^\ell} \right]}_{c_t^\ell} \tag{111.39}$$

SS_t^ℓ is called the **similarity score** for leaf ℓ and tree t. Note that an increase in similarity decreases the cost.

111.3 Leaf Splitting

Suppose leaf ℓ_0 splits into leafs ℓ_L and ℓ_R. Then

$$\Sigma_t^{\ell_0} = \Sigma_t^{\ell_L} \cup \Sigma_t^{\ell_R}, \quad \Sigma_t^{\ell_L} \cap \Sigma_t^{\ell_R} = \emptyset \tag{111.40}$$

so

$$A_t^{\ell_0} = A_t^{\ell_L} + A_t^{\ell_R} \tag{111.41}$$

and

$$B_t^{\ell_0} = B_t^{\ell_L} + B_t^{\ell_R} \ . \tag{111.42}$$

Hence,

$$\mathcal{C}_t^{\ell_j} = \gamma - \frac{(A_t^{\ell_j})^2}{2(B_t^{\ell_j} + \lambda)} \quad \text{for } j = L, R \tag{111.43}$$

$$\mathcal{C}_t^{\ell_0} = \gamma - \frac{(A_t^{\ell_L} + A_t^{\ell_R})^2}{2(B_t^{\ell_L} + B_t^{\ell_R} + \lambda)} \tag{111.44}$$

Define the **XGBoost branch Gain** for a binary tree branch by

$$GainXGB = \mathcal{C}_t^{\ell_0} - \mathcal{C}_t^{\ell_L} - \mathcal{C}_t^{\ell_R} \tag{111.45}$$

$$= \underbrace{\left\{SS_t^{\ell_L} + SS_t^{\ell_R} - SS_t^{\ell_0}\right\}}_{\Delta SS_t} - \gamma \tag{111.46}$$

By splitting a leaf, we create a new binary branch with 2 new leafs. We successively add new branches to the tree until some stopping criterion is satisfied. We continue adding branches to the tree as long as they have a positive gain, and as long as the number of levels is smaller than an upper bound input parameter (XGBoost's default maximum number of tree levels is six). XGBoost also rules out leafs that have a denominator quantity B_t^ℓ (called the **cover**) smaller than a lower bound input parameter. This branching process is illustrated in Figs.111.1 and 111.2.

111.4 Pruning

If we raise γ or raise λ, branches that previously had positive gain may acquire a negative gain. Get rid of branches from highest level that now have negative gain. This will generate new branches. Get rid of new branches from the highest level that have negative gain. Continue this process until all highest level branches have positive gain. This may reduce a tree to a single node or even rule out the entire tree.

λ measures **level of insensitivity** to observations, and γ measures **level of tree simplicity**.

Figure 111.1: Plot of target attribute y^σ versus a feature vector x^σ with a single component. More generally, the feature vector $x^\sigma = (x^\sigma_j)_{j=0,1,\ldots,nf-1}$ can have multiple components called features or attributes. The left plot refers to a population of 4 individuals and regression so $y^\sigma, \widehat{y}^\sigma \in \mathbb{R}$. The right plot refers to a population of 4 individuals and binary classification so $y^\sigma \in \{0,1\}, \widehat{y}^\sigma \in [0,1]$. $X_{j,j+1}$ for $j = 0,1,2$ is the average between 2 adjacent values of x^σ.

Figure 111.2: This figure refers to the situation of Fig.111.1. Out of all allowed tree branches, we choose the one with the highest gain.

111.5 Feature Binning

For really large population sizes $|\Sigma|$, it is convenient to bin the set $\{x_i^\sigma : \sigma \in \Sigma\}$ for each i.

A common bin type is quantiles. Quantiles are bins $[X_{j-1}, X_j]$ for $j = 0, 1, \ldots, nbins-1$ that all contain approximately the same number of points. For example, in Fig.111.2, with 1 point quantile bins, use $[X_{j-1}, X_j]$ for $j = 0, 1, 2, 3$. With 2 point quantile bins, use $[X_{j-1}, X_j]$ for $j = 0, 1$.

Use the right edge of each bin as the X_j in the question $x^\sigma < X_j$?, and choose the question which yields the highest gain.

111.6 Final estimate of target attribute

In this section, we will give a formula for the final estimate of the target attribute. Previously, we set the learning rate η to one. Here we restore η to an arbitrary value between 0 and 1.

Instead of using

$$f(x^\sigma) = \sum_{t=0}^{nt-1} f_t(x^\sigma) \ , \tag{111.47}$$

we will use

$$f(x^\sigma) = \sum_{t=0}^{nt-1} (\eta)^t f_t(x^\sigma) \ , \tag{111.48}$$

where $\eta \in [0, 1]$ is called the **learning rate**. This has the effect of compensating for the f_t's with $t > nt - 1$ that would have been included had we used an infinite series. Also, think of Eq.(111.47) as an approximation (a truncated Taylor expansion in powers of Δx) of a function $f(x + \Delta x)$, and think of Eq.(111.48) as an analogous approximation of the function $f(x + \eta\Delta x)$.

For the case of Regression, we get:

$$f(x^\sigma) \quad = \quad \sum_t (\eta)^t \left(\frac{-A_t^{\ell_t(\sigma)}}{B_t^{\ell_t(\sigma)} + \lambda} \right) \ . \tag{111.49}$$

For the case of Binary Classification, we get

$$f(x^\sigma) = \sum_{t=0}^{nt-1} (\eta)^t \underbrace{\text{smoid} \underbrace{\left(\frac{-A_t^{\ell_t(\sigma)}}{B_t^{\ell_t(\sigma)} + \lambda} \right)}_{\text{lodds(probability)}}}_{\text{probability}} \ . \tag{111.50}$$

111.7 Bnet for XGBoost

Figure 111.3: Bnet for XGBoost assuming $nt = 3$. Nodes that appear multiple times (namely $\underline{\vec{x}}$ and $\underline{\vec{y}}$) should be considered the same node, drawn multiple times for clarity. The parameters λ, γ, η are taken to be global. Recall that $\ell_t : \Sigma \to \mathcal{L}_t$. From the function $\ell_t()$, we can derive its range $\mathcal{L}_t = \{\ell_t(\sigma) : \sigma \in \Sigma\})$, and $\Sigma_t^\ell = \{\sigma \in \Sigma : \ell_t(\sigma) = \ell\}$ for all $\ell \in \mathcal{L}_t$.

Fig.111.3 gives our bnet for the XGBoost algo assuming $nt = 3$. The TPMs, printed in blue, for this bnet, are as follows:

For $t = 0$, ℓ_0 describes a single node "tree" such that $\ell_0(x^\sigma) = f_0(x^\sigma) = 0.5$ for all $\sigma \in \Sigma$. For $t \geq 1$,

$$P(\ell_t|\vec{y}, [\widehat{y}_{t-1}^\sigma]_\sigma) = \begin{array}{l}\text{build tree } \ell_t \text{ using } y^\sigma \text{ and } \widehat{y}_{t-1}^\sigma \text{ for all } \sigma \in \Sigma. \\ \lambda \text{ and } \gamma \text{ are used here.}\end{array} \tag{111.51}$$

$$P([\widehat{y}_t^\sigma]_\sigma \mid f_t, \vec{x}) = \prod_{\sigma\in\Sigma} \mathbb{1}(\quad \widehat{y}_t^\sigma = f_t(x^\sigma) \quad) \tag{111.52}$$

$$P([A_t^\ell]_{\ell\in\mathcal{L}_t} \mid \vec{y}, [\widehat{y}_{t-1}^\sigma]_\sigma, \ell_t) = \prod_{\ell\in\mathcal{L}_t} \mathbb{1}(\quad A_t^\ell = -\sum_{\sigma\in\Sigma_t^\ell} \underbrace{(y^\sigma - \widehat{y}_{t-1}^\sigma)}_{r_t^\sigma} \quad) \tag{111.53}$$

$$P([B_t^\ell]_{\ell\in\mathcal{L}_t} \mid [\widehat{y}_{t-1}^\sigma]_\sigma, \ell_t) = \prod_{\ell\in\mathcal{L}_t} \mathbb{1}\left(\quad B_t^\ell = \sum_{\sigma\in\Sigma_t^\ell} \left\{ \begin{array}{ll} 1 & \text{if reg.} \\ \widehat{y}_{t-1}^\sigma(1-\widehat{y}_{t-1}^\sigma) & \text{if b.c.} \end{array} \right\} \quad \right) \tag{111.54}$$

$$P(f_t \mid [A_t^\ell]_{\ell\in\mathcal{L}_t}, [B_t^\ell]_{\ell\in\mathcal{L}_t}, \ell_t, \vec{x}) = \prod_{\sigma\in\Sigma} \mathbb{1}\left(f_t(x^\sigma) = \left\{ \begin{array}{ll} 1(& \text{if reg.} \\ \text{smoid}(& \text{if b.c.} \end{array} \right\} \frac{-A^{\ell_t(\sigma)}}{2(B^{\ell_t(\sigma)}+\lambda)}) \right) \tag{111.55}$$

$$P(f \mid [f_t]_t, \vec{x}) = \prod_{\sigma\in\Sigma} \mathbb{1}\left(\quad f(x^\sigma) = \sum_t (\eta)^t f_t(x^\sigma) \quad \right) \tag{111.56}$$

Chapter 112

YAML for bnet storage

When dealing with bnets, it is often necessary to store them for future reuse. For instance, my Mappa Mundi software (Ref.[92]) stores bnets for future reuse. I does so continuously, as they are learned by the AI. The bnets are stored in a directory that I call a **DAG atlas**.

In this chapter, we will discuss a language called YAML, and how to store bnets using YAML.

There are infinitely many ways of storing a bnet. The reasons why we propose using the YAML language is that it is a popular, standardized, human readable, and fairly succinct language.

The configuration information of a software app, and the data exchanged between apps, is often stored in a YAML data structure.

XML

```
<Servers>
  <Server>
    <name>Server1</name>
    <owner>Prajwal</owner>
    <status>active</status>
  </Server>
  <Server>
    <name>Server2</name>
    <owner>John</owner>
    <status>inactive</status>
  </Server>
</Servers>
```

JSON

```
{
  "Servers": [
    {
      "name": "Server1",
      "owner": "Prajwal",
      "status": "active"
    },
    {
      "name": "Server2",
      "owner": "John",
      "status": "inactive"
    }
  ]
}
```

YAML

```
Servers:
  - name: Server1
    owner: Prajwal
    status: active
  - name: Server2
    owner: John
    status: inactive
```

Figure 112.1: For simple data structures, one can translate between JSON, XML and YAML.

YAML is a human-readable data serialization language. XML and JSON are too. As illustrated by Fig.112.1, for simple data structures, one can translate a data

structure from one of those languages to the other 2. But note that YAML is the most succinct language of the 3. So in this chapter, we will speak only about YAML.

112.1 Getting acquainted with YAML

In Chapter 16, we demonstrated that a decision tree can be converted to a bnet with the same structure as the tree.

The purpose of this section is not to teach YAML to the reader. In this section, we will assume that the reader has learned YAML already, from one of many excellent introductions to YAML available on the internet.

The purpose of this section is to demonstrate that a YAML data structure can be converted to a tree. Then using the results of Chapter 16, that tree can be converted to a decision tree which can be converted to a bnet.

Like Python, YAML code can contain 2 data structures: dictionaries such as

$$\boxed{\begin{array}{l}\texttt{a : 1}\\ \texttt{b : 5}\\ \texttt{c : 1}\end{array}} \leftrightarrow \{\texttt{a : 1},\quad \texttt{b : 5},\quad \texttt{c : 1}\}$$

and lists such as

$$\boxed{\begin{array}{ll}-&\texttt{x}\\ -&\texttt{y}\\ -&\texttt{z}\end{array}} \leftrightarrow [\texttt{x},\quad \texttt{y},\quad \texttt{z}]$$

In YAML, the dictionaries have all their key-value pairs start with the same level of indentation. The list items also start with the same level of indentation, but all start with a hyphen and a space. Lists can always be replaced by dictionaries:

$$[\texttt{x},\quad \texttt{y},\quad \texttt{z}] \rightarrow \{\texttt{0 : x},\quad \texttt{1 : y},\quad \texttt{2 : z}\}$$

$$\boxed{\begin{array}{ll}-&\texttt{x}\\ -&\texttt{y}\\ -&\texttt{z}\end{array}} \rightarrow \boxed{\begin{array}{ll}\texttt{0 :}&\texttt{x}\\ \texttt{1 :}&\texttt{y}\\ \texttt{2 :}&\texttt{z}\end{array}}$$

Once you replace in a YAML data structure, all lists by dictionaries, then you have dictionaries with key-value pairs such that some of the values in the pairs can lead to new dictionaries. Thus, we get a tree. See Figs.112.2 and 112.3 for examples of conversions of YAML data structures to trees.

Figure 112.2: The nodes of a YAML list can contain various cargoes.

Figure 112.3: The nodes of a YAML dictionary can contain various cargoes.

112.2 Storing Bnets

As an example, below is a possible way of fully specifying the LDEN bnet[1]:

[1]LDEN bnets are defined in Chapter 52

using YAML. Of course, there are many other ways of doing this.

```
graph0:
  nodes:
    - id: A
      label: Node A
      values:
        - 0
        - 1
      parents: None
      probabilities: [0.3, 0.7]
    - id: B
      label: Node B
      values:
        - 0
        - 1
      parents:
        -A
      probabilities: [[0.8, 0.2], [0.6, 0.4]]
    - id: C
      label: Node C
      values:
        - 0
        - 1
      parents:
        - A
      probabilities: [[0.8, 0.2], [0.6, 0.4]]
    - id: D
      label: Node D
      values:
        - 0
        - 1
      parents:
        - B
        - C
```

```
      probabilities: [[0.9, 0.1], [0.3, 0.7], [0.5, 0.5], [0.4, 0.6]]
  edge_gains:
    (A, B): 3 # arrow from A to B has gain 3
    (A, C): 5
    (B, D): -6
    (C, D): 3
```

Note that if the `probabilities` are given (for a bnet with probabilitic nodes), then the `edge_gains` (for an LDEN) should excluded, and vice versa.

Another thing to notice is that the `probabilities` is a tensor representing a transition probability matrix (TPM) $T^{[n_1],[n_2],[a]}$ with components

$$T^{\nu_1,\nu_2,\alpha} = P(\alpha|\nu_1,\nu_2) \tag{112.1}$$

where[2]

$\nu_1 \in [n_1]$= values (a.k.a. states) of parent 1,

$\nu_2 \in [n_2]$= values of parent 2,

$\alpha \in [a]$= values of focus node,

where the focus node is the node being considered, and we are assuming the focus node has 2 parents.

[2]As usual in this book, we use the notation $[n] = [0:n] = \{0, 1, \ldots, n-1\}$

Chapter 113

Zero Information Transmission (Graphoid Axioms)

This chapter assumes that you have read Chapter 24 on d-separation.

The following quantities play a very prominent role in the d-separation Theorem that we enunciated in Chapter 24.

- the mutual information (MI)
  (a.k.a. information transmission) $H(\underline{a} : \underline{b})$
- the conditional mutual information (CMI)
  (a.k.a. conditional information transmission) $H(\underline{a} : \underline{b}|\underline{c})$

MI can be viewed as the special case of CMI, when the set of variables being conditioned on is empty. Particularly prominent in d-separation discussions are probability distributions for which CMI vanishes. The goal of this chapter is to study such probability distributions.

Recall that CMI is non-negative and symmetric in its first two variables (i.e., $H(\underline{a} : \underline{b}|\underline{c}) = H(\underline{b} : \underline{a}|\underline{c})$). Another very useful property of CMI is its chain rule

Claim 200 *(Chain Rule for CMI)*

$$H(\underline{y} : \underline{x}^n) = \sum_i H(\underline{y} : \underline{x}_i|\underline{x}_{<i}) \ , \tag{113.1}$$

where $\underline{x}^n = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{n-1})$ *and* $\underline{x}_{<i} = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{i-1})$.

proof:

$$\frac{P(y|x_{<i+1})}{P(y)} = \frac{P(y|x_i, x_{<i})}{P(y|x_{<i})} \quad \frac{P(y|x_{<i})}{P(y)} \tag{113.2}$$

Therefore,

$$\ln \frac{P(y|x_{<i+1})}{P(y)} = \ln \frac{P(y|x_i, x_{<i})}{P(y|x_{<i})} + \ln \frac{P(y|x_{<i})}{P(y)} \tag{113.3}$$

If we now apply $\sum_{y,x^n} P(y,x^n)$ to both sides, we get

$$H(\underline{y}:\underline{x}_{<i+1}) = H(\underline{y}:\underline{x}_i|\underline{x}_{x<i}) + H(\underline{y}:x_{<i}) \tag{113.4}$$

QED

A trivial but very useful consequence of the chain rule for CMI is:

$$\boxed{H(\underline{y}:\underline{x}^n) = 0 \iff H(\underline{y}:\underline{x}_i|\underline{x}_{<i}) = 0 \text{ for all } i}\ . \tag{113.5}$$

113.1 Consequences of Eq.(113.5)

Table 113.1 gives a set of statements about CMI referred to as the Graphoid Axioms in chapter 1 of Ref.[62]. See Ref.[62] to learn the history of these axioms. The purpose of this section is to prove that the graphoid axioms are all a simple consequence of Eq.(113.5).

Symmetry	$\underline{a} \perp_P \underline{b} \implies \underline{b} \perp_P \underline{a}$ $H(\underline{a}:\underline{b}) = 0 \implies H(\underline{b}:\underline{a}) = 0$
Decomposition	$\underline{a} \perp_P \underline{b},\underline{c} \implies \underline{a} \perp_P \underline{b}$ and $\underline{a} \perp_P \underline{c}$ $H(\underline{a}:\underline{b},\underline{c}) = 0 \implies H(\underline{a}:\underline{b}) = 0$ and $H(\underline{a}:\underline{c}) = 0$
Weak Union	$\underline{a} \perp_P \underline{b},\underline{c} \implies \underline{a} \perp_P \underline{b}\|\underline{c}$ and $\underline{a} \perp_P \underline{c}\|\underline{b}$ $H(\underline{a}:\underline{b},\underline{c}) = 0 \implies H(\underline{a}:\underline{b}\|\underline{c}) = 0$ and $H(\underline{a}:\underline{c}\|\underline{b}) = 0$
Contraction	$\underline{a} \perp_P \underline{b}\|\underline{c}$ and $\underline{a} \perp_P \underline{c} \implies \underline{a} \perp_P \underline{b},\underline{c}$ $H(\underline{a}:\underline{b}\|\underline{c}) = 0$ and $H(\underline{a}:\underline{c}) = 0 \implies H(\underline{a}:\underline{b},\underline{c}) = 0$
Intersection	$\underline{a} \perp_P \underline{b}\|\underline{c},\underline{d}$ and $\underline{a} \perp_P \underline{d}\|\underline{c},\underline{b} \implies \underline{a} \perp_P \underline{b},\underline{d}\|\underline{c}$ $H(\underline{a}:\underline{b}\|\underline{c},\underline{d}) = 0$ and $H(\underline{a}:\underline{d}\|\underline{c},\underline{b}) = 0 \implies H(\underline{a}:\underline{b},\underline{d}\|\underline{c}) = 0$

Table 113.1: Graphoid Axioms

Claim 201 *Table 113.1 is true.*

proof:

- **Symmetry**

  Follows trivially from $H(\underline{a}:\underline{b}) = H(\underline{b}:\underline{a})$.

- **Decomposition**

  From the chain rule for CMI, we have

$$H(\underline{a}:\underline{b},\underline{c}) = H(\underline{a}:\underline{b}|\underline{c}) + H(\underline{a}:\underline{c})\ , \tag{113.6}$$

and

$$H(\underline{a} : \underline{b}, \underline{c}) = H(\underline{a} : \underline{c}|\underline{b}) + H(\underline{a} : \underline{b}) \ . \tag{113.7}$$

Hence,

$$H(\underline{a} : \underline{b}, \underline{c}) = 0 \tag{113.8}$$

implies

$$H(\underline{a} : \underline{b}|\underline{c}) = H(\underline{a} : \underline{c}) = 0 \ , \tag{113.9}$$

and

$$H(\underline{a} : \underline{c}|\underline{b}) = H(\underline{a} : \underline{b}) = 0 \ . \tag{113.10}$$

- **Weak Union**

  Already proven in proof of Decomposition.

- **Contraction**

  From chain rule for CMI, we have

$$H(\underline{a} : \underline{b}, \underline{c}) = H(\underline{a} : \underline{b}|\underline{c}) + H(\underline{a} : \underline{c}) \ . \tag{113.11}$$

- **Intersection**

  From the chain rule for CMI, we have

$$H(\underline{a} : \underline{b}, \underline{d}|\underline{c}) = H(\underline{a} : \underline{b}|\underline{d}, \underline{c}) + H(\underline{a} : \underline{d}|\underline{c}) \ , \tag{113.12}$$

  and

$$H(\underline{a} : \underline{b}, \underline{d}|\underline{c}) = H(\underline{a} : \underline{d}|\underline{b}, \underline{c}) + H(\underline{a} : \underline{b}|\underline{c}) \ . \tag{113.13}$$

  Thus,

$$H(\underline{a} : \underline{b}, \underline{d}|\underline{c}) = 0 \tag{113.14}$$

  implies

$$H(\underline{a} : \underline{b}|\underline{d}, \underline{c}) = H(\underline{a} : \underline{d}|\underline{c}) = 0 \ , \tag{113.15}$$

  and

$$H(\underline{a} : \underline{d}|\underline{b}, \underline{c}) = H(\underline{a} : \underline{b}|\underline{c}) = 0 \ . \tag{113.16}$$

.

QED

Bibliography

[1] Alan Agresti. *An introduction to categorical data analysis*. John Wiley & Sons, 2018.

[2] Agrumery. PyArum, free open source app at github. https://github.com/agrumery/aGrUM.

[3] Data Analytics and IIT Delhi Intelligence Research (DAIR) Group. Openie6. https://github.com/dair-iitd/openie6.

[4] Elias Bareinboim, Jin Tian, and Judea Pearl. Recovering from selection bias in causal and statistical inference. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. https://ftp.cs.ucla.edu/pub/stat_ser/r425.pdf.

[5] Dan Bendel. Metropolis-Hastings: A comprehensive overview and proof. https://similarweb.engineering/mcmc/.

[6] David Benkeser and Antoine Chambaz. A ride in targeted learning territory. https://achambaz.github.io/tlride/tlride-book.pdf.

[7] Brilliant.org. Finite state machines. https://brilliant.org/wiki/finite-state-machines/.

[8] Rafael Cabañas de Paz, Alessandro Antonucci, Andrés Cano Utrera, Manuel Gómez-Olmedo, et al. Evaluating interval-valued influence diagrams. 2017. https://digibug.ugr.es/bitstream/handle/10481/67901/antonucci2016b.pdf?sequence=1.

[9] Alexandra M Carvalho. Scoring functions for learning Bayesian networks. http://www.lx.it.pt/~asmc/pub/talks/09-TA/ta_pres.pdf.

[10] Bo Chang. Copula: a very short introduction, article in Bo's Blog. https://bochang.me/blog/posts/copula/.

[11] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. https://arxiv.org/abs/1603.02754.

[12] Victor Chernozhukov, Carlos Cinelli, Whitney Newey, Amit Sharma, and Vasilis Syrgkanis. Long story short: Omitted variable bias in causal machine learning. https://www.nber.org/system/files/working_papers/w30302/w30302.pdf.

[13] Carlos Cinelli, Andrew Forney, and Judea Pearl. A crash course in good and bad controls. https://ftp.cs.ucla.edu/pub/stat_ser/r493.pdf.

[14] Carlos Cinelli and Chad Hazlett. Making sense of sensitivity: Extending omitted variable bias. https://carloscinelli.com/files/Cinelli%20and%20Hazlett%20(2020)%20-%20Making%20Sense%20of%20Sensitivity.pdf.

[15] Scott Cunningham. *Causal inference: The mixtape*. Yale University Press, 2021. https://mixtape.scunning.com/index.html.

[16] Nir Friedman Daphne Koller. *Probabilistic Graphical Models, Principles and Techniques*. MIT Press, 2009.

[17] Robin J. Evans. Graphical methods for inequality constraints in marginalized DAGs. https://arxiv.org/abs/1209.2978.

[18] Matheus Facure Alves. *Causal Inference for The Brave and True*. 2021. https://matheusfacure.github.io/python-causality-handbook/landing-page.html.

[19] George Fei. Modeling uplift directly: Uplift decision tree with kl divergence and euclidean distance as splitting criteria. https://www.aboutwayfair.com/tech-innovation/modeling-uplift-directly-uplift-decision-tree-with-kl-divergence-and-euclidean-distance-as-splitting-criteria.

[20] Richard P Feynman, Albert R Hibbs, and Daniel F Styer. *Quantum mechanics and path integrals*. Courier Corporation, 2010.

[21] Charles Fox, Neil Girdhar, and Kevin Gurney. A causal Bayesian network view of reinforcement learning. https://www.aaai.org/Papers/FLAIRS/2008/FLAIRS08-030.pdf".

[22] Bruno Gonçalves. Model testing and causal search. blog post https://medium.com/data-for-science/causal-inference-part-vii-model-testing-and-causal-search-536b796f0384.

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, David Warde-Farley Bing Xu, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. https://arxiv.org/abs/1406.2661.

[24] Pierre Gutierrez and Jean-Yves Gérardy. Causal inference and uplift modelling: A review of the literature. In *Proceedings of The 3rd International Conference on Predictive Applications and APIs*, pages 1–13, 2017. http://proceedings.mlr.press/v67/gutierrez17a.html.

[25] B.S. Jeffreys H. Jeffreys. *Methods of Mathematical Physics, 3rd ed.* Cambridge University Press, 1988.

[26] James Douglas Hamilton. *Time series analysis.* Princeton University Press, 2020.

[27] Sebastian Haneuse and Andrea Rotnitzky. Estimation of the effect of interventions that modify the received treatment. *Statistics in medicine*, 32(30):5260–5277, 2013. Main:https://sci-hub.se/10.1002/sim.5907, Supplement: https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1002%2Fsim.5907&file=sim5907-sup-0001-Appendix.pdf.

[28] Christina Heinze-Deml. Causality, spring semester 2019 at ETH Zurich. https://stat.ethz.ch/lectures/ss19/causality.php#course_materials.

[29] MA Hernán and J Robins. Causal inference: What if. Boca Raton: Chapman & Hill/CRC, https://www.hsph.harvard.edu/miguel-hernan/causal-inference-book/.

[30] Katherine Hoffman. An illustrated guide to modified treatment policies, part 1: Introduction and motivation, article in KHstats blog. https://www.khstats.com/blog/lmtp/lmtp/.

[31] Katherine Hoffman. An illustrated guide to TMLE, article in KHstats blog. https://www.khstats.com/blog/tmle/tutorial/.

[32] Cecil Huang and Adnan Darwiche. Inference in belief networks: A procedural guide. *International journal of approximate reasoning*, 15(3):225–263, 1996. http://www.ar-tiste.com/Huang-Darwiche1996.pdf.

[33] Janos Körner Imre Csiszár. *Information Theory- Coding Theorems for Discrete Memoryless Systems.* Academic Press, 1981.

[34] Tommi S. Jaakkola and Michael I. Jordan. Variational probabilistic inference and the QMR-DT network. http://arxiv.org/abs/1105.5462.

[35] Michael I. Jordan. course: Stat260-Bayesian modeling and inference, lecture date: February 8-2010, title: The conjugate prior for the Normal distribution. https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/lectures/lecture5.pdf.

[36] Chaitanya K. Joshi. Transformer (machine learning model). https://graphdeeplearning.github.io/post/transformers-are-gnns/.

[37] Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. Openie6: Iterative grid labeling and coordination analysis for open information extraction. https://arxiv.org/abs/2010.03147.

[38] Chung-Ming Kuan. Introduction to time series analysis, Fall 2014 lectures given at the Department of Finance, National Taiwan University. https://homepage.ntu.edu.tw/~ckuan/pdf/2014fall/Lec-TimeSeries_slide-Fall2014.pdf.

[39] Steffen L Lauritzen and Dennis Nilsson. Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–1251, 2001. https://people.duke.edu/~rnau/DAS_pub_award_2003/lauritzen_nilsson.pdf.

[40] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988. http://www.eecis.udel.edu/~shatkay/Course/papers/Lauritzen1988.pdf.

[41] Boram Lee. Lecture notes by Boram Lee, September 28, 2015.

[42] Sergey Levine. Course CS 285 at UC Berkeley, Deep reinforcement learning. http://rail.eecs.berkeley.edu/deeprlcourse/.

[43] Ang Li. Ph.D. Thesis, UCLA 2021. https://ftp.cs.ucla.edu/pub/stat_ser/r507.pdf.

[44] Dimitris Margaritis. Learning Bayesian network model structure from data (thesis, 2003, Carnegie Mellon Univ). https://apps.dtic.mil/sti/citations/ADA461103.

[45] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In *BMC bioinformatics*, volume 7, page S7. Springer, 2006. https://link.springer.com/article/10.1186/1471-2105-7-S1-S7.

[46] Samuele Mazzanti. Black-box models are actually more explainable than a logistic regression. https://towardsdatascience.com/black-box-models-are-actually-more-explainable-than-a-logistic-regression-f263c22795d.

[47] Samuele Mazzanti. SHAP values explained exactly how you wished someone explained to you. https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30.

[48] Robert J. McEliece, David J. C. MacKay, and Jung-Fu Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. http://authors.library.caltech.edu/6938/1/MCEieeejstc98.pdf.

[49] Scott Mueller, Ang Li, and Judea Pearl. Causes of effects: Learning individual responses from population data. *arXiv preprint arXiv:2104.13730*, 2021. https://arxiv.org/abs/2104.13730.

[50] Brady Neal. Introduction to causal inference, Fall 2020, lectures and book. https://www.bradyneal.com/causal-inference-course.

[51] Richard E Neapolitan. *Learning Bayesian networks*. Pearson Prentice Hall, 2004.

[52] Andrew Ng. Lecture at deeplearning.ai on recurrent neural networks. http://www.ar-tiste.com/ng-lec-rnn.pdf.

[53] Gregory Nuel. Tutorial on exact belief propagation in Bayesian networks: from messages to algorithms. https://arxiv.org/abs/1201.4724.

[54] paperspace.com. PyTorch 101, Part 1: Understanding graphs, automatic differentiation and autograd. https://blog.paperspace.com/pytorch-101-understanding-graphs-and-automatic-differentiation/.

[55] Kalyanapuram R Parthasarathy. *An introduction to quantum stochastic calculus*, volume 85. Birkhäuser, 2012.

[56] Judea Pearl. Linear models: A useful microscope for causal analysis. https://ftp.cs.ucla.edu/pub/stat_ser/r409-corrected-reprint.pdf.

[57] Judea Pearl. Mediating instrumental variables. https://ftp.cs.ucla.edu/pub/stat_ser/r210.pdf.

[58] Judea Pearl. On the testability of causal models with latent and instrumental variables. https://arxiv.org/abs/1302.4976.

[59] Judea Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. https://www.aaai.org/Papers/AAAI/1982/AAAI82-032.pdf, 1982.

[60] Judea Pearl. *Probabilistic Inference in Intelligent Systems*. Morgan Kaufmann, 1988.

[61] Judea Pearl. The causal mediation formula—a guide to the assessment of pathways and mechanisms. *Prevention science*, 13(4):426–436, 2012. https://apps.dtic.mil/sti/pdfs/ADA557663.pdf.

[62] Judea Pearl. *Causality: Models, Reasoning, and Inference, Second Edition*. Cambridge University Press, 2013.

[63] Judea Pearl. Causal and counterfactual inference. *The Handbook of Rationality*, pages 1–41, 2019. https://ftp.cs.ucla.edu/pub/stat_ser/r485.pdf.

[64] Judea Pearl and Elias Bareinboim. Transportability of causal and statistical relations: A formal approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, 2011. https://ojs.aaai.org/index.php/AAAI/article/view/7861.

[65] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[66] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018.

[67] Judea Pearl and James M Robins. Probabilistic evaluation of sequential plans from causal models with hidden variables. *arXiv preprint arXiv:1302.4977*, 2013. https://arxiv.org/abs/1302.4977.

[68] Ashwin Rao and Tikhon Jelvis. *Foundations of Reinforcement Learning with Applications in Finance*. https://stanford.edu/~ashlearn/RLForFinanceBook/book.pdf.

[69] ReliaSoft. System analysis reference. http://reliawiki.org/index.php/System_Analysis_Reference.

[70] Walter Rudin. *Principles of Mathematical Analysis, 3rd ed.* McGraw-Hill, 1976.

[71] Piotr Rzepakowski and Szymon Jaroszewicz. Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, 32(2):303–327, 2012. https://link.springer.com/content/pdf/10.1007/s10115-011-0434-0.pdf.

[72] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019. https://users.aalto.fi/~ssarkka/pub/sde_book.pdf.

[73] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*. Cambridge University Press, 2019. https://users.aalto.fi/~ssarkka/pub/sde_book.pdf.

[74] Scholarpedia. Granger causality. http://www.scholarpedia.org/article/Granger_causality.

[75] Marco Scutari. bnlearn. https://www.bnlearn.com/.

[76] Marco Scutari, Catharina Elisabeth Graafland, and José Manuel Gutiérrez. Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms. *International Journal of Approximate Reasoning*, 115:235–253, 2019. https://arxiv.org/abs/1805.11908.

[77] Ross D Shachter and Debarun Bhattacharjya. Solving influence diagrams: Exact algorithms. *Wiley encyclopedia of operations research and management science.*, 2010. http://www.ar-tiste.com/Shachter2010.pdf.

[78] Nitish Srivastava, G E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf.

[79] StatQuest. XGBoost Parts 1-4 (videos). https://statquest.org/video-index/.

[80] Russell Stuart and Norvig Peter. Artificial intelligence-a modern approach 3rd ed, 2016.

[81] Masayoshi Takahashi. Statistical inference in missing data by MCMC and non-MCMC multiple imputation algorithms: Assessing the effects of between-imputation iterations. *Data Science Journal*, 16, 2017. https://datascience.codata.org/articles/10.5334/dsj-2017-037/.

[82] theinvestorsbook.com. Pert analysis. https://theinvestorsbook.com/pert-analysis.html.

[83] Joy A. Thomas Thomas M. Cover. *Elements of Information Theory*. Wiley-Interscience, 1991.

[84] Jin Tian and Judea Pearl. Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence*, 28(1):287–313, 2000. https://ftp.cs.ucla.edu/pub/stat_ser/r271-A.pdf.

[85] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999. https://publications.aston.ac.uk/id/eprint/38367/1/NCRG_97_010.pdf.

[86] Robert R. Tucci. Bayes-Petri net. https://github.com/rrtucci/Bayes_Petri_Net.

[87] Robert R. Tucci. Bayesian networks (a.k.a. causal models, DAGs) and the passage of time. blog post in blog Quantum Bayesian Networks, https://qbnets.wordpress.com/2021/07/16/bayesian-networks-aka-causal-models-dags-and-the-passage-of-time/.

[88] Robert R. Tucci. Bell's inequalities for Bayesian statisticians. blog post in blog Quantum Bayesian Networks, https://qbnets.wordpress.com/2008/09/19/bells-inequaties-for-bayesian-statistician/.

[89] Robert R. Tucci. CausalFitbit. https://github.com/rrtucci/CausalFitbit.

[90] Robert R. Tucci. Goodness of causal fit. https://github.com/rrtucci/DAG_Lie_Detector.

[91] Robert R. Tucci. JudeasRx. https://github.com/rrtucci/JudeasRx.

[92] Robert R. Tucci. Mappa Mundi. https://github.com/rrtucci/mappa_mundi.

[93] Robert R. Tucci. Quantum d-separation and quantum belief propagation. https://arxiv.org/abs/2012.09635.

[94] Robert R. Tucci. Quantum Fog. https://github.com/artiste-qb-net/quantum-fog.

[95] Robert R. Tucci. SCuMpy. https://github.com/rrtucci/scumpy.

[96] Robert R. Tucci. SentenceAx. https://github.com/rrtucci/SentenceAx.

[97] Robert R. Tucci. Shannon information theory without shedding tears over delta & epsilon proofs or typical sequences. https://arxiv.org/abs/1208.2737.

[98] Robert R. Tucci. texnn. https://github.com/rrtucci/texnn.

[99] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. https://arxiv.org/abs/1706.03762.

[100] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl. Fault tree handbook nureg-0492. https://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0492/.

[101] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference (book). https://people.eecs.berkeley.edu/~wainwrig/Papers/WaiJor08_FTML.pdf.

[102] Lilian Weng. The multi-armed bandit problem and its solutions. lilianweng.github.io/lil-log, http://lilianweng.github.io/lil-log/2018/01/23/the-multi-armed-bandit-problem-and-its-solutions.html, 2018.

[103] Lilian Weng. What are diffusion models? lilianweng.github.io/lil-log, https://lilianweng.github.io/posts/2021-07-11-diffusion-models/, 2021.

[104] Wikibooks. Control systems. https://en.wikibooks.org/wiki/Control_Systems.

[105] Wikipedia. AdaBoost. https://en.wikipedia.org/wiki/AdaBoost.

[106] Wikipedia. Autoregressive model. https://en.wikipedia.org/wiki/Autoregressive_model.

[107] Wikipedia. Autoregressive moving-average model. https://en.wikipedia.org/wiki/Autoregressive%E2%80%93moving-average_model.

[108] Wikipedia. Baum-Welsh algorithm. https://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm.

[109] Wikipedia. Belief propagation. https://en.wikipedia.org/wiki/Belief_propagation.

[110] Wikipedia. Berkson's paradox. https://en.wikipedia.org/wiki/Berkson%27s_paradox.

[111] Wikipedia. Bernoulli distribution. https://en.wikipedia.org/wiki/Bernoulli_distribution.

[112] Wikipedia. BERT language model. https://en.wikipedia.org/wiki/BERT_(language_model).

[113] Wikipedia. Beta distribution. https://en.wikipedia.org/wiki/Beta_distribution.

[114] Wikipedia. Beta function. https://en.wikipedia.org/wiki/Beta_function.

[115] Wikipedia. Binary decision diagram. https://en.wikipedia.org/wiki/Binary_decision_diagram.

[116] Wikipedia. Biplot. https://en.wikipedia.org/wiki/Biplot.

[117] Wikipedia. Boolean algebra. https://en.wikipedia.org/wiki/Boolean_algebra.

[118] Wikipedia. Bootstrap aggregating. https://en.wikipedia.org/wiki/Bootstrap_aggregating.

[119] Wikipedia. Brownian dynamics. https://en.wikipedia.org/wiki/Brownian_dynamics.

[120] Wikipedia. Categorical distribution. https://en.wikipedia.org/wiki/Categorical_distribution.

[121] Wikipedia. Chi-square distribution. https://en.wikipedia.org/wiki/Chi-square_distribution.

[122] Wikipedia. Chow-Liu tree. https://en.wikipedia.org/wiki/Chow%E2%80%93Liu_tree.

[123] Wikipedia. Cochran's theorem. https://en.wikipedia.org/wiki/Cochran%27s_theorem.

[124] Wikipedia. Conjugate prior. https://en.wikipedia.org/wiki/Conjugate_prior.

[125] Wikipedia. Copula. https://en.wikipedia.org/wiki/Copula_(probability_theory).

[126] Wikipedia. Cramer-Rao bound. https://en.wikipedia.org/wiki/Cram%C3%A9r%E2%80%93Rao_bound.

[127] Wikipedia. Cross-validation. https://en.wikipedia.org/wiki/Cross-validation_(statistics).

[128] Wikipedia. Data processing inequality. https://en.wikipedia.org/wiki/Data_processing_inequality.

[129] Wikipedia. Deterministic finite automaton. https://en.wikipedia.org/wiki/Deterministic_finite_automaton.

[130] Wikipedia. Dimensionality reduction. https://en.wikipedia.org/wiki/Dimensionality_reduction.

[131] Wikipedia. Dirichlet distribution. https://en.wikipedia.org/wiki/Dirichlet_distribution.

[132] Wikipedia. Errors in variables models. https://en.wikipedia.org/wiki/Errors-in-variables_models.

[133] Wikipedia. Expectation maximization. https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm.

[134] Wikipedia. F-distribution. https://en.wikipedia.org/wiki/F-distribution.

[135] Wikipedia. Factor analysis. https://en.wikipedia.org/wiki/Factor_analysis.

[136] Wikipedia. Frisch-Waugh-Lovell theorem. https://en.wikipedia.org/wiki/Frisch%E2%80%93Waugh%E2%80%93Lovell_theorem.

[137] Wikipedia. Functional derivative. https://en.wikipedia.org/wiki/Functional_derivative.

[138] Wikipedia. Gamma distribution. https://en.wikipedia.org/wiki/Gamma_distribution.

[139] Wikipedia. Gamma function. https://en.wikipedia.org/wiki/Gamma_function.

[140] Wikipedia. Gated recurrent unit. https://en.wikipedia.org/wiki/Gated_recurrent_unit.

[141] Wikipedia. Gibbs sampling. https://en.wikipedia.org/wiki/Gibbs_sampling.

[142] Wikipedia. Granger causality. https://en.wikipedia.org/wiki/Granger_causality.

[143] Wikipedia. Hidden Markov model. https://en.wikipedia.org/wiki/Hidden_Markov_model.

[144] Wikipedia. Hoeffding's inequality. https://en.wikipedia.org/wiki/Hoeffding%27s_inequality.

[145] Wikipedia. Importance sampling. https://en.wikipedia.org/wiki/Importance_sampling.

[146] Wikipedia. Instrumental variables estimation. https://en.wikipedia.org/wiki/Instrumental_variables_estimation.

[147] Wikipedia. Inverse transform sampling. https://en.wikipedia.org/wiki/Inverse_transform_sampling.

[148] Wikipedia. Jackknife resampling. https://en.wikipedia.org/wiki/Cross-validation_(statistics).

[149] Wikipedia. Junction tree algorithm. https://en.wikipedia.org/wiki/Junction_tree_algorithm.

[150] Wikipedia. k-means clustering. https://en.wikipedia.org/wiki/K-means_clustering.

[151] Wikipedia. Kalman filter. https://en.wikipedia.org/wiki/Kalman_filter.

[152] Wikipedia. Kernel method. https://en.wikipedia.org/wiki/Kernel_method.

[153] Wikipedia. Kernel perceptron. https://en.wikipedia.org/wiki/Kernel_perceptron.

[154] Wikipedia. Kernel principal component analysis. https://en.wikipedia.org/wiki/Kernel_principal_component_analysis.

[155] Wikipedia. Laplace transform. https://en.wikipedia.org/wiki/Laplace_transform.

[156] Wikipedia. Least squares. https://en.wikipedia.org/wiki/Least_squares.

[157] Wikipedia. Legendre transformation. https://en.wikipedia.org/wiki/Legendre_transformation.

[158] Wikipedia. Likelihood-ratio test. https://en.wikipedia.org/wiki/Likelihood-ratio_test.

[159] Wikipedia. Linear regression. https://en.wikipedia.org/wiki/Linear_regression.

[160] Wikipedia. Long short term memory. https://en.wikipedia.org/wiki/Long_short-term_memory.

[161] Wikipedia. Markov blanket. https://en.wikipedia.org/wiki/Markov_blanket.

[162] Wikipedia. Metropolis-Hastings method. https://en.wikipedia.org/wiki/Metropolis%E2%80%93Hastings_algorithm.

[163] Wikipedia. Minimum spanning tree. https://en.wikipedia.org/wiki/Minimum_spanning_tree.

[164] Wikipedia. Monte Carlo methods. https://en.wikipedia.org/wiki/Category:Monte_Carlo_methods.

[165] Wikipedia. Moving-average model. https://en.wikipedia.org/wiki/Moving-average_model.

[166] Wikipedia. Multinomial distribution. https://en.wikipedia.org/wiki/Multinomial_distribution.

[167] Wikipedia. Multinomial theorem. https://en.wikipedia.org/wiki/Multinomial_theorem.

[168] Wikipedia. Multivariate normal distribution. https://en.wikipedia.org/wiki/Multivariate_normal_distribution.

[169] Wikipedia. Natural experiment. https://en.wikipedia.org/wiki/Natural_experiment.

[170] Wikipedia. Non-deterministic finite automaton. https://en.wikipedia.org/wiki/Nondeterministic_finite_automaton.

[171] Wikipedia. Non-negative matrix factorization. https://en.wikipedia.org/wiki/Non-negative_matrix_factorization.

[172] Wikipedia. Ordinary least squares. https://en.wikipedia.org/wiki/Ordinary_least_squares.

[173] Wikipedia. Petri net. https://en.wikipedia.org/wiki/Petri_net.

[174] Wikipedia. Principal component analysis. https://en.wikipedia.org/wiki/Principal_component_analysis.

[175] Wikipedia. Program evaluation and review technique. https://en.wikipedia.org/wiki/Program_evaluation_and_review_technique.

[176] Wikipedia. Random forest. https://en.wikipedia.org/wiki/Random_forest.

[177] Wikipedia. Receiver operating characteristic. https://en.wikipedia.org/wiki/Receiver_operating_characteristic.

[178] Wikipedia. Rejection sampling. https://en.wikipedia.org/wiki/Rejection_sampling.

[179] Wikipedia. Score test. https://en.wikipedia.org/wiki/Score_test.

[180] Wikipedia. Signal flow graph. https://en.wikipedia.org/wiki/Signal-flow_graph.

[181] Wikipedia. Simple linear regression. https://en.wikipedia.org/wiki/Simple_linear_regression.

[182] Wikipedia. Simpson's paradox. https://en.wikipedia.org/wiki/Simpson's_paradox.

[183] Wikipedia. Spring system. https://en.wikipedia.org/wiki/Spring_system.

[184] Wikipedia. Student's t-distribution. https://en.wikipedia.org/wiki/Student%27s_t-distribution.

[185] Wikipedia. Support vector machine. https://en.wikipedia.org/wiki/Support-vector_machine.

[186] Wikipedia. Survival analysis. https://en.wikipedia.org/wiki/Survival_analysis.

[187] Wikipedia. Thermodynamics. https://en.wikipedia.org/wiki/Thermodynamics.

[188] Wikipedia. Time series. https://en.wikipedia.org/wiki/Time_series.

[189] Wikipedia. Transfer learning. https://en.wikipedia.org/wiki/Transfer_learning.

[190] Wikipedia. Transformer (machine learning model). https://en.wikipedia.org/wiki/Transformer_(machine_learning_model).

[191] Wikipedia. Turing machine. https://en.wikipedia.org/wiki/Turing_machine.

[192] Wikipedia. Uplift modelling. https://en.wikipedia.org/wiki/Uplift_modelling.

[193] Wikipedia. Variational Bayesian methods. https://en.wikipedia.org/wiki/Variational_Bayesian_methods.

[194] Wikipedia. Vector autoregression. https://en.wikipedia.org/wiki/Vector_autoregression.

[195] Wikipedia. Viterbi algorithm. https://en.wikipedia.org/wiki/Viterbi_algorithm.

[196] Wikipedia. Wald test. https://en.wikipedia.org/wiki/Wald_test.

[197] Wikipedia. Z-transform. https://en.wikipedia.org/wiki/Z-transform.

[198] Hao Wu and Zhaohui Steve Qin. course notes, BIOS731: Advanced statistical computing, 2016 Emory Univ. http://web1.sph.emory.edu/users/hwu30/teaching/statcomp/statcomp.html.

[199] Ronghui (Lily) Xu. Lecture notes, MATH 284, Spring 2020, Survival analysis. https://mathweb.ucsd.edu/~rxu/math284/.

[200] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Understanding belief propagation and its generalizations, Mitsubishi Technical Report tr-2001-22. https://merl.com/publications/docs/TR2001-22.pdf.