

赌书消得泼茶香 当时只道是寻常

RESEARCH NOTES

彭畅 (CHANG PENG)

SOUTHEAST UNIVERSITY



2019 年 6 月 29 日 To 2024 年 12 月 16 日

目录

封面	1
目录	1
第一部分 二次事故 (Secondary Crash)	1
第一章 Real-time Estimation of Secondary Crash Likelihood on Freeways Using High-resolution Loop Detector Data	2
1.1 Introduction	2
1.2 Data sources	3
1.3 Methodology	3
1.3.1 贝叶斯随机效应 Logit 模型 (Bayesian random effect logit model)	3
1.3.2 受试者工作特性曲线 (Receiver Operating Characteristic curve, ROC curve)	4
1.4 Data analysis and result	4
1.4.1 Identification of secondary crashes	4
1.4.2 Characteristics of primary and normal crashes	4
1.4.3 Secondary crash risk prediction model	4
1.5 附录	5
1*I 英汉互译	5
1*II 随机效应模型 (random effect model)	6
1*III 二次事故风险评估模型候选变量及解释	6
1*IV 弹性分析 (Elasticity Analysis)	6
1*V 受试者工作特性曲线 (Receiver Operating Characteristic curve, ROC curve)	7
1*VI 似然与似然比检验 (Likelihood & Likelihood Ratio Test, LRT)	7
1*VII 二次事故风险评估模型	8
第二章 Investigating the factors affecting secondary crash frequency caused by one primary crash using zero-inflated ordered probit regression	9
2.1 Introduction	9
2.2 Data source	10
2.3 Methodology	10
2.3.1 零膨胀有序 probit 模型 (Zero-inflated ordered probit model)	10
2.3.2 Vuong 检验 (Vuong test)	10
2.4 Data analysis and results	10
2.4.1 Results of the ZIOP models	10
2.5 附录	11

2*I	英汉互译	11
2*II	零膨胀模型 (zero-inflated model, ZIM)	11
2*III	有序 probit 模型 (ordered porbit model)	12
2*IV	候选变量及描述	12
2*V	vuong 检验	13
2*VI	由单次事故引起的二次事故次数预测模型 (ZIOP)	14
第三章	Use of ubiquitous probe vehicle data for identifying secondary crashes	16
3.1	Introduction	16
3.2	Literature review	16
3.3	Methodology	17
3.3.1	Detecting the impact area	17
3.3.2	Estimation boundary of the impact data	18
3.3.3	Automated identification of secondary crashes & Ray casting algorithm	18
3.4	Simulation test & Data analysis and result	19
3.5	附录	19
3*I	英汉互译	19
3*II	模糊 C 均值 (Fuzzy C-means, FCM) 聚类	20
3*III	参考图片	20
第四章	Variable speed limit control at fixed freeway bottlenecks using connected vehicles	22
4.1	Introduction	22
4.2	VSL strategies using connected vehicle(s)	23
4.2.1	Baseline case	23
4.2.2	Strategy 1: control using single connected vehicle	24
4.2.3	Strategy 2: control using single connected vehicle and VMSs	25
4.2.4	Strategy 3: control using multiple CVs	26
4.3	Adaptive control to remedy control failure	27
4.4	Probabilistic control failure	27
4.4.1	Probability of control failure	27
4.4.2	Optimal design to maximize delay saving	28
4.4.3	Delay saving and optimal control speed for the adaptive control	28
4.5	Discuss on traffic instability & Conclusions and discussions	28
4.6	英汉互译	29
第二部分 交通网络		30
第五章	网络需求-状态解析	31
5.1	Empirics of a Generalized Macroscopic Fundamental Diagram for Urban Freeways (<i>TRR, 2013</i>)	31
5.1.1	Introduction	31
5.1.2	Data processing	31
5.1.3	Predicting production & Comparison of MFD and GMFD	32
5.1.4	英汉互译	33
5.2	Urban Traffic Pattern Analysis and Applications Based on Spatio-Temporal Non-Negative Matrix Factorization (<i>TITS, 2022</i>)	33
5.2.1	Introduction & Related work	33

5.2.2	Methodology: spatial-temporal non-negative matrix factorization	34
5.2.3	Traffic pattern analysis and extended applications based on ST-NMF	36
5.2.4	Experiment results and analysis	36
5.2.5	英汉互译	37
5.3	Predicting electric vehicle charging demand using a heterogeneous spatio-temporal graph convolutional network (<i>TRC</i> , 2023)	37
5.3.1	Introduction & Literature review	37
5.3.2	Preliminaries & Methodology	38
5.3.3	Case study of Beijing & Evaluating model performance with a public dataset	39
5.3.4	英汉互译	40
5.4	KST-GCN: A Knowledge-Driven Spatial-Temporal Graph Convolutional Network for Traffic Forecasting (<i>TITS</i> , 2022)	40
5.4.1	Introduction & Related works	40
5.4.2	Methods	41
5.4.3	Experiments	41
5.4.4	英汉互译	41
5.5	An Ensemble of Deep Clustering Models With Autoencoders to Mine Travel Patterns From Smart Card Data (<i>TITS</i> , 2024)	42
5.5.1	Introduction & Related works	42
5.5.2	Deep clustering algorithms & Travel pattern mining system & Overfitting in deep clustering	42
5.5.3	Results and discussion	43
5.5.4	英汉互译	43
5.6	Motif discovery based traffic pattern mining in attributed road networks (<i>Knowledge-Based Systems</i> , 2022)	44
5.6.1	Introduction & Related works	44
5.6.2	Methodology	45
5.6.3	Experiments	45
5.6.4	英汉互译	46
第六章 网络资源配置		47
6.1	A model for planning locations of temporary distribution facilities for emergency response (<i>Socio-Economic Planning Sciences</i> , 2015)	47
6.1.1	Introduction	47
6.1.2	Emergency logistics resource distribution model	47
6.1.3	Numerical analysis	48
6.1.4	英汉互译	49
6.2	A Stochastic Emergency Response Location Model Considering Secondary Incidents on Freeways (<i>TITS</i> , 2016)	49
6.2.1	Introduction	49
6.2.2	Literature review	50
6.2.3	Stochastic process of incident occurrence	51
6.2.4	Stochastic ERU deployment model	52
6.2.5	Numerical examples	54
6.2.6	英汉互译	55
6.3	Demand-driven timetable design for metro services (<i>TRC</i> , 2014)	55

6.3.1	Introduction & Background	55
6.3.2	Timetable design problem	56
6.3.3	Case study & Results and analysis	59
6.3.4	英汉互译	59
6.4	Planning of static and dynamic charging facilities for electric vehicles in electrified transportation networks (<i>Energy</i> , 2023)	59
6.4.1	Introduction	60
6.4.2	System model	60
6.4.3	英汉互译	60
6.5	附录	60
6*I	模型参数说明 (A model for planning locations of temporary distribution facilities)	60
6*II	模型参数说明 (A Stochastic Emergency Response Location Model Considering Secondary Incidents on Freeways)	61
6*III	混合 0-1 非线性规划线性化的一些方法	62

第七章 网络控制 63

7.1	Mitigating freeway off-ramp congestion: A surface streets coordinated approach (<i>TRC</i> , 2012)	63
7.1.1	Introduction	63
7.1.2	Determining the stationary fraction to detour	64
7.1.3	Advisability of the strategy during a rush hour	64
7.1.4	Field implementation	65
7.1.5	英汉互译	66
7.2	Design and Implementation of Integrated Network Management Methodology in a Regional Network (<i>TRR</i> , 2015)	66
7.2.1	Introduction	66
7.2.2	System design and functional architecture	67
7.2.3	Coordination of controlled intersections for ramp-metering support	67
7.2.4	Control approach analysis and tuning	68
7.2.5	英汉互译	69
7.3	Coordinated Ramp Metering Based on On-Ramp Saturation Time Synchronization (<i>TRR</i> , 2015)	69
7.3.1	Background	69
7.3.2	Control approach	70
7.3.3	Test case	71
7.3.4	Results	72
7.3.5	英汉互译	72
7.4	A varying parameter multi-class second-order macroscopic traffic flow model for coordinated ramp metering with global and local environmental objectives (<i>TRC</i> , 2021)	73
7.4.1	Introduction and background	73
7.4.2	Traffic flow model development	74
7.4.3	Emissions and environmental policies	77
7.4.4	Optimal control problem formulation	78
7.4.5	Differential evolution and solution evaluation	79
7.4.6	Test network case study & Results and discussion	80
7.4.7	英汉互译	80

7.5	Real-time Traffic Network State Estimation And Prediction with Decision Support Capabilities: Application to Integrated Corridor Management (<i>TRC, 2016</i>)	81
7.5.1	Background	81
7.5.2	Problem definition and formulation	82
7.5.3	Real-time traffic network management system: overall framework & Decision support capabilities	83
7.5.4	Testbed description & Experiments, results and analysis	84
7.5.5	英汉互译	85
7.6	A Simulation-Based Optimization Framework for Urban Transportation Problems (<i>OR, 2013</i>)	86
7.6.1	Introduction & Literature review	86
7.6.2	Metamodel	87
7.6.3	Optimization algorithm	87
7.6.4	Traffic signal control	88
7.6.5	Empirical analysis	89
7.6.6	英汉互译	89
第八章	网络需求调控	90
8.1	Active learning for multi-objective optimal road congestion pricing considering negative land use effect (<i>TRC, 2021</i>)	90
8.1.1	Introduction	90
8.1.2	Model formulation	91
8.1.3	An active learning algorithm for the multi-objective bi-level programming model	92
8.1.4	Case study	93
8.1.5	英汉互译	94
8.2	A novel mobility consumption theory for road user charging (<i>TRB, 2024</i>)	94
8.2.1	Introduction	94
8.2.2	An analogy between transport and electricity systems	95
8.2.3	A novel mobility consumption theory & Application to road user charging	95
8.2.4	Impact of mobility-based charging on travel behaviour and congestion	96
8.2.5	Technology innovation's impact on mobility consumption	98
8.2.6	英汉互译	99
第三部分	自动驾驶 (Autonomous Driving)	100
第九章	轨迹规划	101
9.1	Motion planning in complex environments using closed-loop prediction (2008)	101
9.1.1	Introduction	101
9.1.2	Overview of the approach	101
9.1.3	Controller	102
9.1.4	英汉互译	106
9.2	Generation of Reference Trajectories for Safe Trajectory Planning (<i>ICANN, 2018</i>)	106
9.2.1	Introduction	106
9.2.2	Variational autoencoder & HARRT+ Algorithm	107
9.2.3	英汉互译	107

第四部分 交通仿真	108
第十章 静态需求估计与交通分配	109
10.1 Modeling capacity flexibility of transportation networks (<i>TRA, 2011</i>)	109
10.1.1 Introduction	109
10.1.2 Definitions and basic approaches to measuring flexibility	110
10.1.3 Capacity flexibility assessment of a passenger transportation system	111
10.1.4 Numerical Results	113
10.1.5 英汉互译	114
10.2 Metamodel-based calibration of large-scale multi-modal microscopic traffic simulation (<i>TRC, 2021</i>)	114
10.2.1 Introduction	114
10.2.2 Problem statement & Measurement formation	115
10.2.3 Metamodal SBO (simulation-based optimization) structure	115
10.2.4 Multimodal traffic model formulation	116
10.2.5 Traffic model improvement & Metamodel formulation and fitting	118
10.2.6 Trust region SBO algorithm	119
10.2.7 Numerical examples & Application to Hong Kong network	119
10.2.8 英汉互译	120
10.3 Calibration of Microscopic Traffic Simulation Models: Methods and Application (<i>TRR, 2007</i>)	120
10.3.1 Introduction	120
10.3.2 Methodology & Solution approaches	121
10.3.3 Case study & Calibration results	121
10.3.4 英汉互译	122
10.4 Reducing the Dimension of Online Calibration in Dynamic Traffic Assignment Systems (<i>TRR, 2017</i>)	123
10.4.1 Introduction & Literature review	123
10.4.2 Online calibration: problem formulation	123
10.4.3 PC-based calibration	124
10.4.4 Case study on Singapore expressway network	125
10.4.5 英汉互译	125
10.5 Data driven origin-destination matrix estimation on large networks: A joint origin-destination-path-choice formulation (<i>TRC, 2024</i>)	126
10.5.1 Introduction	126
10.5.2 Methodology	126
10.5.3 Case setup & Case study results	128
10.5.4 英汉互译	130
10.6 Discrete choice theory, information theory and the multinomial logit and gravity models (<i>TRB, 1983</i>)	130
10.6.1 Stochastic utility maximization and the multinomial logit model	130
10.6.2 Information minimization and the multinomial logit model	131
10.6.3 The doubly constrained gravity model is a logit model of joint origin-destination choice	132
10.6.4 英汉互译	133
10.7 交通分配模型求解	133
10.7.1 Frank-Wolfe 算法	133
10.7.2 partial linearization 算法	134
10.7.3 MSA 算法与 MSWA 算法	135
10.8 附录	136

10*I 基于路径尺度的 Logit 模型 (path-size Logit, PSL)	136
10*II 最大熵原理、双重约束重力模型与交通分布预测	137
第十一章 动态交通分配与交通流模型	139
11.1 A demand model with departure time choice for within-day dynamic traffic assignment (<i>EJOR</i> , 2006)	139
11.1.1 The choice and demand models	139
11.1.2 Supply and equilibrium models	143
11.1.3 Algorithm	148
11.1.4 英汉互译	150
11.2 A novel metamodel-based framework for large-scale dynamic origin-destination demand calibration (<i>TRC</i> , 2022)	151
11.2.1 Introduction	151
11.2.2 Methodology	152
11.2.3 Demonstrating in the Sioux-falls network	154
11.2.4 Case study of a large-scale network problem for the Melbourne CBD	154
11.2.5 英汉互译	155
11.3 SUMO 换道模型	155
11.3.1 DK2008	155
11.3.2 LC2013	156
11.3.3 英汉互译	159
11.4 附录	159
11*I 有限容量排队论 (finite capacity queueing theory) 与路网建模	159
第十二章 车载自组网仿真	161
12.1 Simulation environment for VANET	161
12.1.1 Introduction	161
12.1.2 Simulation environment	161
12.1.3 英汉互译	162
12.2 Effect of information availability on stability of traffic flow: Percolation theory approach (<i>TRB</i> , 2018)	162
12.2.1 Introduction	163
12.2.2 Background	163
12.2.3 Definitions	164
12.2.4 Percolation of vehicular ad-hoc networks	164
12.2.5 Analytical investigation of string stability	165
12.2.6 Results and discussion	168
12.2.7 英汉互译	168
第十三章 基础设施仿真	169
13.1 Electric vehicle charging station diffusion: An agent-based evolutionary game model in complex networks (<i>Energy</i> , 2022)	169
13.1.1 Introduction & Literature review	169
13.1.2 Method	170
13.1.3 Case study	173
13.1.4 英汉互译	173

第五部分 Statistical and Econometric Methods for Transportation Data Analysis, SEMTDA	174
第十四章 Fundamentals	175
14.1 Descriptive Statistics, 描述统计学	175
14.2 Interval Estimation, Hypothesis Testing and Population Comparison, 区间估计、假设检验与总体比较	177
14.2.1 Confidence Interval, 置信区间	178
14.2.2 Hypothesis Testing, 假设检验	179
14.2.3 Comparing Two Populations, 两个总体的比较	179
14.2.4 Nonparametric Methods, 非参数方法	181
14.3 英汉互译	184
第十五章 Continuous Dependent Variable Models	186
15.1 Linear Regression	186
15.1.1 递推最小二乘 (Recursive least square, RLS)	188
15.2 Latent Variable Models, 隐变量模型	188
15.2.1 Principal component analysis	188
15.2.2 Factor analysis	189
15.2.3 Structural equation modelling	190
15.3 英汉互译	192
第十六章 Count and Discrete Dependent Variable Models	193
16.1 Count Data Models	193
16.1.1 Poisson regression model	193
16.1.2 Negative binomial model	194
16.1.3 Zero-inflated Poisson and negative binomial regression models	194
16.1.4 Random-effects count models	195
16.2 Logistic Regression	195
16.3 英汉互译	195
第六部分 数学工具	197
第十七章 运筹学	198
17.1 线性规划	198
17.1.1 线性规划的基本定理	199
17.1.2 单纯形法 (simplex method)	201
17.1.3 单纯形表	204
17.1.4 大 M 法与两阶段法 (big M method & two-phase method)	206
17.2 线性规划的对偶理论与灵敏度分析	208
17.2.1 线性规划的对偶问题	208
17.2.2 对偶理论	210
17.2.3 影子价格 (shadow price)	213
17.2.4 对偶单纯形法	214
17.2.5 灵敏度分析	215
17.3 非线性规划基本概念	215
17.3.1 最优性条件——从拉格朗日乘数法到 KKT 条件 (Karush-Kuhn-Tucker conditions)	215
17.3.2 一般化对偶原理推导——拉格朗日对偶	217

17.3.3 对偶问题求解算法——优化算法的对偶形式	219
17.3.4 线搜索 (Line search) 优化与信赖域 (Trust region) 优化	220
17.4 次梯度 (Sub-gradient) 与近端梯度下降 (Proximal gradient descent)	221
17.4.1 次梯度最优化条件与次梯度下降优化	221
17.4.2 近端点法 (proximal point method, PPM)	223
17.4.3 近端梯度下降优化	224
17.4.4 改进近端梯度下降优化	225
17.5 对偶分解 (Dual decomposition) 与原始分解 (Primal decomposition)	226
17.6 Douglas-Rachford 分裂算法与交替方向乘子法	227
17.6.1 Douglas-Rachford 分裂算法 (Douglas-Rachford splitting, DRS)	227
17.6.2 交替方向乘子法 (Alternating direction method of multipliers, ADMM)	228
17.7 坐标下降 (Coordinate descent) 与块坐标下降 (Block coordinate descent)	229
17.8 多目标优化	230
17.8.1 帕累托最优 (Pareto optimality) 基本概念	230
第十八章 图论与复杂网络	232
18.1 图的基本概念	232
18.1.1 树	234
18.1.2 割点与桥	235
18.2 图的连通度	235
18.3 最短路问题	235
18.3.1 狄克斯托 (Dijkstra) 算法	236
18.4 最大流问题	236
18.4.1 基本概念	236
18.4.2 Ford-Fulkerson 算法	237
18.5 拉普拉斯矩阵与图傅里叶变换	238
18.5.1 从拉普拉斯算子到拉普拉斯矩阵 (graph Laplacian)	238
18.5.2 标准化拉普拉斯矩阵 (Laplacian matrix normalization)	239
18.5.3 图的微分运算	241
18.5.4 图傅里叶变换 (graph Fourier transform, GFT)	242
18.6 复杂网络基本概念	242
18.7 ER 网络	245
18.8 小世界网络	247
18.8.1 从 W-S 模型到 N-W 模型——网络的小世界特性	247
18.8.2 从 N-W 模型到 Kleinberg 模型——网络的可导航性	248
18.9 无标度网络	249
18.9.1 BA 无标度网络模型	250
第十九章 启发式 (Heuristic) 算法与元启发式 (Meta-heuristic) 算法	252
19.1 场景分析 (Scenario analysis) 与场景缩减 (Scenario reduction)	252
19.1.1 基本定义	252
19.1.2 同步回代缩减法 (simultaneous backward reduction, SBR)	253
19.1.3 快速前向选择法 (fast forward selection, FFS)	253
19.1.4 场景树 (scenario tree)	253
19.2 同步扰动随机逼近算法 (Simultaneous perturbation stochastic approximation, SPSA)	254
19.2.1 Weighted-SPSA (W-SPSA)	254

19.3 遗传算法 (Genetic algorithm, GA)	255
19.3.1 算法思想	255
19.3.2 常用编码方法 (Coding)	255
19.3.3 常用选择算法 (Selection)	256
19.3.4 常用重组算法 (Crossover)	256
19.3.5 常用变异算法 (Mutation)	256
19.4 差分进化 (Differential evolution, DE)	257
19.5 布谷鸟搜索 (Cuckoo search, CS)	258
19.5.1 算法思想	258
19.5.2 随机数更新算法	259
19.6 蚁群算法 (Ant colony optimization, ACO)	260
19.7 快速扩展随机树算法 (Rapidly-exploring random tree, RRT)	261
19.7.1 基础 RRT 算法 (basic RRT)	262
19.7.2 面向高效采样的改进 RRT 算法	262
19.7.3 面向路径优化的改进 RRT 算法	263
第二十章 矩阵分解与经典降维	264
20.1 主成分分析 (Primary component analysis, PCA) 与因子分析 (Factor analysis)	264
20.2 非负矩阵分解 (Non-negative matrix factorization, NMF)	264
20.2.1 问题建模	264
20.2.2 基于拉格朗日乘数法优化——元素形式	266
20.2.3 基于拉格朗日乘数法优化——矩阵形式	267
20.2.4 基于梯度下降优化	267
20.2.5 基于辅助函数优化与收敛性证明	268
20.3 基于零膨胀 Tweedie 分布假设的 NMF 模型	269
20.4 张量分解 (Tensor decomposition/factorization)	272
20.4.1 张量基础知识	272
20.4.2 CP 分解 (canonical polyadic decomposition, CPD)	273
20.4.3 Tucker 分解	274
20.5 概率矩阵分解 (Probabilistic matrix factorization, PMF)	277
20.6 贝叶斯概率矩阵分解 (Bayesian probabilistic matrix factorization, BPMF)	278
20.7 贝叶斯非负矩阵分解 (Bayesian non-negative matrix factorization, BNMF)	280
20.8 贝叶斯张量分解 (Bayesian tensor factorization, BTF)	283
20.8.1 贝叶斯 CP 张量分解	283
20.9 T 分布随机近邻嵌入算法 (T-Distribution Stochastic Neighbor Embedding, t-SNE)	285
20.9.1 随机近邻嵌入算法 (SNE)	285
20.9.2 对称随机近邻嵌入算法 (Symmetric SNE)	287
20.9.3 T 分布随机近邻嵌入算法 (t-SNE)	287
第二十一章 信号处理	289
21.1 信号与系统概述	289
21.2 连续时间系统的时域分析	290
21.2.1 冲激响应的求解	291
21.3 傅里叶级数	291
21.4 傅里叶变换	293
21.4.1 非周期信号的傅里叶变换	293

21.4.2 周期信号的傅里叶变换	295
21.4.3 傅里叶变换的性质	296
21.5 连续时间系统的频域分析方法	296
21.5.1 因果性、Paley-Wiener 准则与物理可实现滤波器	298
21.5.2 调制与解调	298
21.6 拉普拉斯变换	299
21.6.1 拉普拉斯逆变换的求解	302
21.7 连续时间系统的拉普拉斯变换分析法	303
21.8 离散时间系统概述	304
21.9 离散时间傅里叶变换 (DTFT) 与 Z 变换	305
21.9.1 Z 变换	305
21.9.2 Z 变换的性质	306
21.9.3 离散时间傅里叶变换 (DTFT) 与离散时间序列傅里叶级数 (DFS)	307
21.9.4 离散时间傅里叶变换的性质	307
21.10 离散时间系统的变换域分析法	308
21.10.1 Z 域分析法	308
21.10.2 频域分析法	309
21.11 数字信号处理概述	309
21.12 离散傅里叶变换 (DFT)	310
21.13 快速傅里叶变换 (FFT)	312
21.14 傅里叶变换的扩展——信号的多分辨分析	314
21.14.1 短时傅里叶变换 (STFT)	314
21.14.2 连续小波变换 (CWT)	315
21.14.3 离散小波变换 (DWT)	315
21.14.4 小波包变换 (WPT)	316
21.15 数据平滑去噪	316
21.15.1 Savitzky-Golay 滤波器 (S-G 滤波器)	316
第二十二章 控制论	318
22.1 自动控制原理基本概念	318
22.2 经典控制理论的系统输入—输出模型	319
22.2.1 传递函数 (transfer function)	319
22.2.2 频率特性函数	321
22.2.3 极零图与全通系统、最小相移系统	323
22.3 系统的 BIBO 稳定与劳斯-霍尔维茨判据	324
22.4 奈奎斯特 (Nyquist) 判据与稳定裕度	326
22.4.1 闭环系统稳定的奈奎斯特判据	326
22.4.2 稳定裕度——相位裕度与幅值裕度	327
22.5 现代控制理论及控制系统的状态空间表达式	327
22.5.1 状态空间表达式的实现——从传递函数到状态空间表达式	329
22.5.2 状态向量的线性变换	330
22.5.3 从状态空间表达式到传递函数矩阵	332
22.5.4 组合系统的状态空间表达式与传递函数矩阵	332
22.6 控制系统状态空间表达式的解	333
22.6.1 线性定常系统的零输入响应	333

22.6.2 线性定常系统的零状态响应	335
22.6.3 线性时变系统状态方程的解	336
22.7 线性控制系统的能控性和能观性	337
22.7.1 线性定常系统的能控性及判据	337
22.7.2 线性定常系统的能观性及判据	338
22.7.3 线性时变系统的能控性、能观性及判据	339
22.7.4 能控与能观性的对偶关系	340
22.7.5 状态空间表达式的能控性标准型与能观性标准型	341
22.8 稳定性与李雅普诺夫 (Lyapunov) 方法	341
22.8.1 系统平衡状态与李雅普诺夫稳定性定义	341
22.8.2 李雅普诺夫第一法 (间接法)	342
22.9 鲁棒控制与 H 无穷控制基础	343
22.9.1 单输入单输出 (SISO) 系统的 H 无穷优化问题建模	343
22.9.2 多输入多输出 (MIMO) 系统的 H 无穷优化问题建模	345
第二十三章 其它知识	346
23.1 正交实验 (Orthogonal Test)	346
23.1.1 正交实验实例	347
23.2 卷积计算及其性质	347
23.3 特殊矩阵运算介绍	348
23.4 矩阵求导	348
23.4.1 标量对矩阵求导	349
23.4.2 矩阵对矩阵求导	350
23.5 离散概率分布采样算法	350
23.6 概率分布的特征函数 (Characteristic function)	351
23.7 经典概率分布	352
23.7.1 理解 Poisson 分布	352
23.7.2 多元高斯分布	352
23.7.3 威沙特分布 (wishart distribution) 与高斯-威沙特分布 (normal-wishart distribution)	355
23.7.4 指数族分布 (exponential family distribution)	356
23.7.5 幂律分布 (Power law distribution)	358
23.8 混合高斯模型与 EM 算法	361
23.8.1 混合高斯模型 (gaussian Mixture Model, GMM)	361
23.8.2 EM 算法 (expectation maximum)	362
23.9 距离 (相似性) 度量	363
23.9.1 点间距离——闵可夫斯基距离 (Minkowski distance)	363
23.9.2 点间距离——马氏距离 (Mahalanobis Distance)	363
23.9.3 序列间距离——Hausdorff 距离	363
23.9.4 序列间距离——EDR (Edit Distance on Real sequence) 距离	364
23.9.5 序列间距离——动态时间规整 (dynamic time warping, DTW)	364
23.9.6 概率分布间的距离——从信息量到 KL 散度与交叉熵	365
23.9.7 概率分布间的距离——Wasserstein 距离	367
23.10 超球体积与超球坐标系	368
23.11 图像特征提取	369
23.11.1 梯度方向直方图 (histogram of oriented gradient, HOG) 特征	369

23.11.2 Gabor 特征	370
23.11.3 尺度不变特征 (scale-invariant feature transform, SIFT)	371
23.12 反函数的数值计算	373
23.13 瑞利商 (Rayleigh quotient) 与广义瑞利商 (generalized Rayleigh quotient)	374
23.14 优劣解距离法 (Technique of order preference by similarity to an ideal solution, TOPSIS)	374
23.15 模糊数学 (Fuzzy mathematics) 基础——模糊集与模糊数	376
23.15.1 模糊集 (fuzzy set) 及其运算性质	376
23.15.2 模糊数 (fuzzy number) 及其运算性质	378
23.16 复变函数理论	380
23.16.1 约当引理 (Jordan lemma)	380
23.16.2 柯西幅角原理 (principle of argument)	380
第七部分 人工智能	382
第二十四章 人工神经网络 (Artificial Neural Network, ANN)	384
24.1 人工神经网络初入	384
24.1.1 BP (backward propagation) 算法与 BP 神经网络	385
24.2 循环神经网络 (RNN) 与长短期记忆 (LSTM)、GRU 结构	386
24.3 面向序列数据的卷积模型	388
24.3.1 门控线性单元 (gated linear unit, GLU)	388
24.3.2 因果卷积模型 (causal convolutional network)	389
24.4 注意力机制 (Attention mechanism)	390
24.4.1 自注意力 (self-attention) 与多头自注意力 (multi-head self-attention)	390
24.4.2 通道注意力 (channel attention) 与空间注意力 (spatial attention)	391
24.5 Word2Vec 工具包	392
24.5.1 文本向量化 (text vectorization) 与词嵌入 (word embedding) 基础	392
24.5.2 Skip-Gram 模型	393
24.5.3 CBOW (continuous bag of words) 模型	394
24.6 Seq2Seq 模型	395
24.7 Transformer 系列	396
24.7.1 标准 Transformer 模型	396
24.7.2 BERT (bidirectional encoder representations from transformer) 模型	397
24.7.3 sentence-BERT 模型	398
24.7.4 GPT-1 (generative pre-training) 模型	399
24.8 通用大语言模型	400
24.8.1 GPT-2/GPT-3	400
24.8.2 GPT-3.5 (InstructGPT) 与基于人类反馈的强化学习 (reinforcement learning from human feedback, RLHF)	402
24.9 自编码器框架	403
24.9.1 经典自编码器 (autoencoder, AE) 与降噪自编码器 (denoising autoencoder, DAE)	403
24.9.2 变分自编码器 (variational autoencoder, VAE)	404
24.9.3 VaDE 模型 (variational deep encoding, IJCAI-17)	406
24.9.4 DCC 模型 (deep continuous clustering)	408

第二十五章 图机器学习 (Graph Machine Learning)	410
25.1 初代图神经网络 (Vanilla graph neural network, Vanilla GNN)	410
25.2 谱域图卷积神经网络 (Spectral graph convolutional neural network)	413
25.2.1 Spectral CNN	414
25.2.2 Chebyshev CNN (ChebNet)	414
25.2.3 一阶 ChebNet (1stChebNet, GCN)	416
25.2.4 扩散图卷积 (<i>ICLR-18</i>)	417
25.3 图递归神经网络 (Graph recurrent neural network, RecGNN)	418
25.3.1 门控图序列神经网络 (gated graph sequence neural network, GGNN)	418
25.4 经典空域图卷积模型 (Spatial graph convolutional neural network)	419
25.4.1 GraphSAGE 模型 (graph sample and aggregate)	419
25.4.2 图注意力神经网络模型 (graph attention network, GAT)	420
25.4.3 ECC (edge-conditioned graph convolution) 模型	421
25.5 时空图模型 (Spatial-temporal graph model)	422
25.5.1 STGCN (spatial-temporal graph convolutional network, <i>IJCAI-18</i>)	422
25.5.2 ASTGCN (attention based spatial-temporal graph convolutional network, <i>AAAI-19</i>)	423
25.5.3 GMAN (graph multi-attention network, <i>AAAI-20</i>)	423
25.5.4 AGCRN (adaptive graph convolutional recurrent network, <i>NeurIPS-20</i>)	425
25.6 基于矩阵分解的图嵌入模型	426
25.6.1 图因子分解 (graph factorization)	426
25.6.2 GraRep 算法	427
25.6.3 LINE (large-scale information network embedding) 模型	428
25.7 基于随机游走 (RandomWalk) 的图嵌入模型	429
25.7.1 DeepWalk 模型	430
25.7.2 Node2Vec 模型	430
25.7.3 Struc2Vec 模型	431
25.8 深度图生成模型 (Deep graph generative model)	432
25.8.1 图自编码器 (graph autoencoder, GAE) 与变分图自编码器 (variational graph autoencoder, VGAE)	432
25.8.2 GraphVAE 模型 (<i>ICANN 2018</i>)	433
25.9 知识图谱 (Knowledge graph) 与知识表示学习 (Knowledge representation learning) 基础	435
25.10 基于翻译距离 (Translational distance) 的知识三元组表示学习 (Trans 系列模型)	436
25.10.1 KR-EAR (knowledge representation learning with entities, attributes and relations, <i>IJCAI-16</i>)	440
25.11 融合大语言模型的图深度学习	441
25.11.1 GLEM 框架 (graph and language learning by expectation-maximization, <i>ICLR 2023</i>)	441
25.11.2 Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning (<i>ICLR 2024</i>)	443
25.11.3 Exploring the potential of large language models (LLMs) in learning on graphs (<i>ACM SIGKDD Explorations Newsletter, 2024</i>)	444
25.12 物理/知识驱动的图深度学习	448
第二十六章 集成学习 (Ensemble Learning)	449
26.1 决策树模型 (Decision tree)	449
26.1.1 决策树生成算法——ID3、C4.5、CART	449
26.1.2 决策树剪枝 (pruning)	451

26.2 AdaBoost (Adaptive boosting, 自适应增强)	451
26.2.1 AdaBoost 分类	452
26.2.2 AdaBoost 回归	454
26.3 梯度提升树 (Gradient boosting decision tree, GBDT)	454
26.4 XGBoost (Extreme gradient boosting, 极端梯度提升)	456
26.4.1 误差函数二阶泰勒展开的意义——与 AdaBoost、GBDT 算法的理论联系	457
26.4.2 XGBoost 决策树生成算法	457
26.5 LightGBM	458
26.5.1 LightGBM 的决策树模型优化	458
26.5.2 LightGBM 的样本规模压缩	459
26.6 CatBoost (Gradient boosting with categorical features support)	460
26.7 随机森林 (Random forest)	460
第二十七章 贝叶斯机器学习	461
27.1 马尔可夫链蒙特卡罗方法 (Markov Chain Monte Carlo, MCMC)	461
27.1.1 蒙特卡罗方法 (Monte Carlo method)	461
27.1.2 马尔可夫链采样	462
27.1.3 MCMC 采样与估计	463
27.2 贝叶斯变分推断 (Variational Bayesian Inference)	465
27.2.1 问题建模	465
27.2.2 坐标上升变分推断 (coordinate ascent variational inference, CAVI)	466
27.2.3 黑盒变分推断 (black box variational inference, BBVI)	467
27.3 朴素贝叶斯模型 (Naive Bayesian Model, NBM)	468
27.4 贝叶斯线性回归	468
27.5 高斯过程回归 (Gaussian Process Regression, GPR)	469
27.5.1 从高斯过程到高斯过程回归——函数空间视角 (function-space)	469
27.5.2 从 KNN 到高斯过程回归——无参数机器学习	471
27.5.3 从贝叶斯线性回归到高斯过程回归——权重空间视角 (weight-space)	471
27.5.4 超参数优化	472
27.6 贝叶斯优化 (Bayesian Optimization)	473
27.7 基于贝叶斯推断的矩阵与张量分解	474
第二十八章 强化学习 (Reinforcement Learning)	475
28.1 强化学习概述	475
28.2 动态规划 (Dynamic programming, DP) 与马尔科夫决策过程 (Markov decision process, MDP)	477
28.2.1 策略迭代法 (policy iteration)	478
28.2.2 值迭代法 (value iteration)	478
28.3 蒙特卡洛法 (Monte Carlo method)	478
28.4 时间差分 (Time difference, TD)	479
28.4.1 SARSA 算法	479
28.4.2 Q-Learning 算法	480
28.5 深度强化学习简介	480
28.6 Deep Q Network (DQN) 算法	480
28.6.1 NIPS DQN 与 Nature DQN	480
28.6.2 Double DQN	481
28.6.3 Dueling DQN	482

28.6.4 优先回放 DQN (Priority Replay DQN)	482
28.7 深度确定性策略梯度法 (Deep deterministic policy gradient, DDPG)	483
28.7.1 随机策略梯度 (stochastic policy gradient, SPG)	483
28.7.2 确定性策略梯度 (deterministic policy gradient, DPG)	484
28.7.3 深度确定性策略梯度 (deep deterministic policy gradient, DDPG)	485
28.8 异步优势 AC 算法 (Asynchronous advantage actor-critic, A3C)	486
第八部分 计算机科学	488
第二十九章 计算机网络	489
29.1 概述	489

赌书消得泼茶香 当时只道是寻常

第一部分

二次事故 (*Secondary Crash*)

第1章

Real-time Estimation of Secondary Crash Likelihood on Freeways Using High-resolution Loop Detector Data

使用高分辨率环形探测器数据实时估计高速公路上的二次碰撞可能性

ABSTRACT

1. 交通数据与事故数据来源：加州 I-880 高速公路（五年内）
2. 二次事故识别方法：等速度线图（speed contour plot）
3. 采用随机效应 Logit 模型（random effect logit model）分析碰撞参数、环境参数和几何参数对二次事故可能性的影
响
4. 对二次事故概率有显著影响的交通变量：交通量、平均速度、探测器占用率的标准偏差、相邻车道交通量之差
5. 对交通流数据和随机效应的采用使得预测准确度分别增加了 16.6% 和 7.7%

1.1 Introduction

1. 优化事故管理体系可降低二次事故的风险，而事故管理体系的优化离不开对二次事故影响因素的研究
2. 几种识别二次事故的方法：
 - (a) 静态阈值法（Static Threshold Method）：基于固定的时间和空间阈值识别二次事故。假定二次事故必然发生在主事故所产生的的最大时空影响范围内。例如取静态时间阈值为 15min、空间阈值为 1km，则认为在主事故发生后 15min 内发生的位于主事故位置上游 1km 范围内的事故为二次事故。缺点在于静态阈值的正确取值较难，导致精度往往偏低。
 - (b) 动态法（Dynamic Method）：包括事故发展曲线法（Incident Progression Curve）、队列长度估计法（Queue Length Estimation）、累计到达离开图（Cumulative Arrival and Departure Plot）、基于仿真的考虑冲击波的方法（Simulation-based Method Considering Shockwave）。
事故发展曲线法与队列长度估计法的目的是为了得到动态的时空阈值，累计到达离开图用于预测因车道堵塞而引起的车队的最大长度和消散时间。
动态法在一定程度上解决了静态阈值法的问题，但不少动态法本身存在着其它的缺点：例如基于队列长度预测的方法需要详细的队列长度数据，而这往往无法获得；基于事故发展曲线的方法采用相同的发展曲线识别所有二次事故，降低了识别的准确率。
 - (c) 通过识别主事故时空影响区域的方法：等速度线图、自动追踪移动堵塞法（Automatic Tracking on Moving Jams Method）、贝叶斯结构方程模型（Bayesian Structure Equation Model）、冲击波原理。
3. 相比于主事故，二次事故特征：
 - (a) 更可能仅有财产损失（property-damage-only, PDO）；

- (b) 更可能是追尾 (rear-end);
 - (c) 更易发生与高速和雨天;
 - (d) 商用车更容易引发二次事故;
 - (e) 引发更长的延迟 (与两次事故的间隔有关);
4. 影响二次事故的相关特征:
- (a) 持续时间长的追尾事故更容易引发二次事故;
 - (b) 发生于非高峰期或周末的主事故更不可能引发二次事故;
 - (c) 主事故涉及的车辆越多越可能引发二次事故;
 - (d) 不利的气候条件 (雨雪) 容易引发二次事故;
 - (e) 日平均交通量 (AADT) 越大越容易引发二次事故;
 - (f) 发生于主干道的事故较发生于高速的更容易引发二次事故;
 - (g) 曲线段更容易引发二次事故;
 - (h) 由高速路闭路电视 (Closed Circuit Television, CCTV) 检测的事故更容易引发二次事故;
5. Logit 模型常用于二次事故预测, 其优势包括:
- (a) 不需要对变量的方差和分布进行假设;
 - (b) 避免过拟合 (Over-fitting);
 - (c) 可处理不平衡数据;
 - (d) 可提供切实的数学模型并得到自变量和因变量的相互关系, 更有利于解决实际的工程问题;
6. 基于动态交通流数据实时预测二次事故风险的研究较少;
7. 动态交通管理系统 (Dynamic Traffic Management System, DTMS), 如变量控制系统 (Variables Limit System)、匝道仪控系统 (Ramp Metering System) 可降低高速路事故概率, 而二次事故实时预测模型有助于发展更加积极的动态交通管理系统, 即通过流量控制和限速等方式消除可能引发二次事故的交通情况;
8. 本次研究的根本目的在于建立一个考虑了实时交通流状况影响的二次事故预测模型, 从而填补之前研究未考虑实时交通流的空白, 以期达到更高的预测准确率。

1.2 Data sources

1. 数据来源: 06-10 年加州 I-880 高速公路 35 英里区段内的交通和事故数据, 事故特征包括日期、时间、事故严重程度、碰撞类型、路面情况、天气情况和光照情况, 模型的变量详见 [1*III](#);
2. 共识别出 9188 次事故 (113 次二次事故、97 次主事故、8978 次普通事故), 其中 4846 次事故为南向, 4342 次事故为北向;
3. 交通数据取自事故发生前 5-10 分钟, 离事故发生地最近的探测线圈记录的数据;

1.3 Methodology

1.3.1 贝叶斯随机效应 Logit 模型 (**Bayesian random effect logit model**)

1. 基于实时交通变量、主事故特征、天气状况及几何特征预测高速上二次事故发生的概率;
2. y_n 表示样本 n 的二次碰撞指标, $y_n = 1$ 表示发生二次事故, $y_n = 0$ 表示未发生; x_{in} 表示样本 n 的第 i 个变量; θ_r 表示高速路 r 区段上的不均匀效应, 并假设其服从正态分布;

$$y_n \sim \text{Bernouli}(p_n)$$

$$\text{logit}(p_n) = \ln \frac{p_n}{1 - p_n} = -(\beta_0 + \theta_r + \beta_1 x_{1n} + \beta_2 x_{2n} + \cdots + \beta_i x_{in})$$

$$\theta_r \sim N(0, \Sigma_\theta)$$

3. 采用基于马尔可夫链蒙特卡洛 (Markov chain Monte Carlo, MCMC) 算法仿真的贝叶斯方法估计随机效应 Logit 模型, 采用无信息先验分布 (non-informative prior distribution);

4. 采用弹性分析 (Elasticity Analysis) 研究交通流变量对二次事故发生概率的影响;

$$E_i = \frac{\partial p_n}{\partial x_i} \times \frac{x_i}{p_n} = (1 - p_n)\beta_i x_i$$

5. 上述点弹性公式无法用于指标变量的弹性计算, 采用伪弹性 (pseudo-elasticity) 公式, 即弧弹性:

$$E_i = \left\{ \frac{e^{\Delta(x/\beta)} \times (1 + e^{x_i/\beta_i})}{e^{\Delta(x/\beta)} \times e^{x_i/\beta_i} + 1} - 1 \right\} \times 100$$

1.3.2 受试者工作特性曲线 (Receiver Operating Characteristic curve, ROC curve)

详见1*V。

1.4 Data analysis and result

1.4.1 Identification of secondary crashes

1. 为克服静态阈值法和动态法的缺点, 采用等速度线图识别二次事故;
2. 中心思想: 在考虑反复拥堵影响的前提下使用实时交通流数据确定主事故引发的时空影响范围;
3. 方法细节如下:
 - (a) 提取环形探测器记录的 5min 速度信息用于生成主事故的速度等高线图, 纵坐标为空间位置, 横坐标为时间, 其中时间跨度为事故发生前后各 6 小时。尽管可以清晰地识别队列, 但无法确定队列是由事故引发或是因为重复性的拥挤导致的;
 - (b) 为确定反复拥堵的影响, 提取该年中该道路无事故发生的所有日期中, 在事故发生时事故发生处探测器记录的所有数据, 并取均值。在初始速度等高线图中扣除该值, 得到修改的速度等高线图, 用于确定事故产生的时空影响范围并识别二次事故;

1.4.2 Characteristics of primary and normal crashes

1. 追尾是主事故和二次事故最显著的类型, 高于一般事故 15%。比例测试 (proportionality test) 的结果显示追尾于主事故与一般事故两类事故中的不同比例具有显著性 ($p\text{-value}=0.001$);
2. 同样的, 刷蹭事故于一般事故中的占比显著高于主事故, 上述结果均与现有成果相符;
3. 0:00-6:00am 发生的事故不大可能引发二次事故, 因为这一事件交通量较低, 事故对交通流的影响不大, 而其它时间段内发生一般事故和主事故的比例无显著性, 同样与现有成果相符。

1.4.3 Secondary crash risk prediction model

1. 采用贝叶斯 Logit 模型基于实时交通流数据实现二次事故预测, 以引发二次事故的主事故作为正类, 而一般事故作为负类;
2. 之前的研究大多未考虑实时交通流数据, 为进行比较, 构造一个未考虑实时交通流数据的简化模型;
3. 模型采用的候选特征见章节1*III, 为保证各特征之间的独立性, 计算变量两两间的皮尔逊相关系数 (Pearson correlation parameter)¹ 并生成由相关性最低的几个变量组合的集合, 皮尔逊相关系数公式如下:

$$R_{X,Y} = \frac{N \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{N \sum x_i^2 - (\sum x_i)^2} \sqrt{N \sum y_i^2 - (\sum y_i)^2}}$$

4. 使用逐步回归法 (Stepwise regression) 进行 Logit 分析确定最终的特征, 比较每一模型的对数似然值, 使得对数似然值最大的模型即为最优模型;
5. 采用似然比检验 (likelihood ratio test) 判断是否考虑实时交通流数据会否显著提高二次事故风险预测模型的拟合优度 (goodness-of-fit), 结果显示实时交通流数据的引入可显著提高预测模型的表现;

¹皮尔逊相关系数用于计算一对变量之间的线性相关性, 要求变量线性、连续、服从正态分布, 且成对观测并且没对观测值之间相互独立, 一般 $|R| \geq 0.6$ 即认为两变量为强相关

6. 对交通变量考虑其随机效应以考虑由某些潜在因素导致的不均匀性。同样地，采用似然比检验比较随机效应引入的效果，检验的结果同样支持这一做法。;
7. 建模结果分析（具体特征及对应参数详见1*VII）：
- (a) 事故发生时交通量越大，越可能引发二次事故。此时可通过匝道仪控系统（ramp metering system）减少主要交通流上游的流量降低二次事故发生的风险；
 - (b) 交通流流速越慢，引发二次事故的风险越大。为此需要加快拥挤队列消散，例如增加下游流速并控制上游流速；
 - (c) 探测器占用率标准差（DevOcc）与二次事故风险呈正相关。探测器占用率标准差反映了交通量的震荡情况：标准差越大，说明车辆加速减速越频繁。为此降低上游流速以形成更平稳的交通流有助于减小二次事故风险；
 - (d) 相邻车道交通量之差同样与二次事故风险呈正相关。因为交通量分布的不均衡会驱使驾驶员更多地变道从而增加二次事故的风险。为此建议在事故发生后提醒上游驾驶员并禁止在事故上游影响区内变道以降低二次事故风险；
 - (e) 造成人员伤亡的事故引发二次事故的可能性更小。一方面可能是当事故有人员伤亡时管理部门会更快地到达现场，另一方面造成人员伤亡的事故大多发生在流量较小的情况下；
 - (f) 刚蹭较追尾不容易引发二次事故。
 - (g) 周末较周中不容易引发二次事故。
 - (h) 车道数越多越不容易引发二次事故。
 - (i) 潮湿天气更容易引发二次事故。

1.5 附录

1*I 英汉互译

English	Chinese	English	Chinese	English	Chinese
high-resolution	高分辨率	loop detector	线圈探测器	contour	等高线
traffic volume	交通量	occupancy	占用率	adjacent lane	相邻车道
evaluation	评估	deem	认为	proactive	积极主动的
spatial	空间的	temporal	时间的	queue	队列
static	静态的	criteria	标准（复数）	fixed	固定的
spatiotemporal	时空的	proportional	比例的	rear-end	追尾
severity	严重	assumption	假设	tangible	有形的
explanatory	解释的	explanatory/independent variable	自变量	dependent variable	因变量
ramp	斜坡、匝道	ramp metering system	匝道仪控系统	prone	有倾向... 的
incorporate	包含	southbound	向南的	interval	(时间的) 间隔
sidewipe crash	刚蹭	heterogeneity	不均匀性	denote	表示
prior distribution	先验分布	inference	推理	customary	习惯的
deviation	偏离	standard deviation	标准差	indicator variable	指标变量
pseudo-elasticity	伪弹性	complementary	互补的	compensate	补偿
identical	相同的	recurrent	反复出现的	predominant	显著的
reduced model	简化模型	stepwise	逐步的	convergence	收敛
goodness-of-fit	拟合优度	test statistic	检验统计量	degrees of freedom	自由度
time headway	车头时距	maneuver	调动	disturbance	紊乱
propagate	传播	dissipation	消散	oscillate	摇摆
thereby	因此	scenario	设想	dimensionless	无量纲的
intuitive	凭直觉的	decelerate	减速	turbulence	紊乱/湍流
transferability	可转移性				

1*II 随机效应模型 (random effect model)

固定效应模型 (Fixed Effect Model): 仅分析选择的变量, 其结论仅限于选择的变量而丌做推广。例如比较三所大学毕业生的薪资水平。

随机效应模型 (Random Effect Model): 分析选择的变量所属的类, 其结论将推广至更高的维度, 一般设计随机抽样。例如比较不同类型大学毕业生的薪资水平, 则从每一类大学中随机抽取部分样本, 对样本进行分析, 此时可采用随机效应模型。模型中将原本固定的回归系数视为随机变量, 一般均假定服从正态分布。

混合效应模型 (Mixed Effect Model): 模型中一部分系数是固定的, 另一部分系数是随机的。^{2 3}

1*III 二次事故风险评估模型候选变量及解释

Variable category	Variable	Description
Real-time traffic conditions	AvgCnt	Average vehicle count during 5-min penod(veh/30s)
	AvgSpd	Average veNcle speed during 5-min period (mile/h)
	AvgOcc	Average detector occupancy during 5-min period (%)
	DevCnt	Std. dev. of vehicle count during 5-min period (veh/30s)
	DevSpd	Std. dev. of vehicle speed during 5-min period (mile/h)
	DevOcc	Std. dev. of detector occupancy during 5-min period (%)
	CovCnt	Coefficient of variation of count during 5-min period (veh/30s)
	CovSpd	Coefficient of variation of speed during 5-min period (mile/h)
	Covdce	Coefficient of variation of occupancy during 5-min period (mile/h)
	DifCnt	Vehicle count difference between adjacent lanes (veh/30s)
	DifSpd	Vehicle speed difference between adjacent lanes (veh/30s)
	DifOcc	Occupancy difference between adjacent lanes (veh/30s)
Primary crash characteristics	Severity	1 - Injury crashes; 0 - PDO
	Sideswipe	1 - Sideswipe crash; 0 - otherwise
	Rear-end	1 - Rear end crash; 0 - otherwise
	Peak	1 - Peak period; 0 - otherwise
	Dayweek	1 - Weekend; 0 - weekday
Environmental conditions	father	1 - Adverse weather conditions; 0 - clear
	Roadsurf	1 - Road surface is wet; 0 - otherwise
	Lighting	1 - No street lights or street lights not functioning; 0 - otherwise
Geometric characteristics	Une	Number of lanes
	Widths	Road surface width(ft)
	Widthm	Inner median width(ft)
	Curve	1 - Curve section; 0 - otherwise

1*IV 弹性分析 (Elasticity Analysis)

- 经济学上研究自变量和应变量间定量变动关系有两种基本手段:
 - 弹性分析-Elasticity Analysis: 分析变量间相对变动的关系;
 - 边际分析-Margin Analysis: 分析变量间绝对变动的关系;
- 弹性定义为自变量变化 1% 而引起的因变量变化的百分比, 度量一个变量的相对变化关于另一个变量相对变化的敏感程度。与边际效应相比, 弹性效应为无量纲量, 因此更有利于比较不同变量的影响;
 - 点弹性-elasticity at a point: 特定点 (Y, x_1, \dots, x_i) 上自变量微小相对变化对因变量相对变化的影响

$$E_i = \frac{\partial Y/Y}{\partial x_i/x_i} = \frac{\partial Y}{\partial x_i} \cdot \frac{x_i}{Y}$$

- 弧弹性-elasticity over an interval: 曲线两点间自变量相对变化对因变量相对变化的影响

$$E_i = \frac{\Delta Y/\bar{Y}}{\Delta x_i/\bar{x}_i} = \frac{Y_2 - Y_1}{x_{i2} - x_{i1}} \cdot \frac{x_{i2} + x_{i1}}{Y_2 + Y_1}$$

² 《随机效应与固定效应 & 面板数据回归》: <https://www.jianshu.com/p/3ed6575a21c8>

³ 《统计学中「固定效应 vs. 随机效应」》: <http://xueshu.blogchina.com/721185424.html>

3. 在经济学中，需求价格大多呈负相关，故需求价格弹性往往取绝对值， $E \in [0, +\infty)$ 。价格弹性越低，则需求价格曲线越陡，可替代产品越少，需求对价格的敏感程度越低。特别地，价格弹性等于 1 表示无论产品价格如何变化，用于产品的总支出不变，此时需求价格曲线为反比例函数。
4. 需要注意的是，点弹性计算公式无法用于指标变量（Indicator variable）的分析，此时可采取一些其它方法计算，如伪弹性（pseudo-elasticity）公式，即弧弹性。

1*V 受试者工作特性曲线 (Receiver Operating Characteristic curve, ROC curve)

1. ⁴ROC 曲线是一种用于评价分类器性能的曲线；
2. 以二分类问题为例，分类混淆矩阵如下：

真实情况	预测结果	
	正例	反例
正例	True Positive, TP	False Negative, FN
反例	False Positive, FP	True Negative, TN

定义如下指标：

- (a) TPR (Sensitivity): 实际为正例的所有样本中，被预测为正例的比例， $TPR = TP \div (TP + FN)$
- (b) TNR (Specificity): 实际为负例的所有样本中，被预测为负例的比例， $TNR = TN \div (FP + TN)$
- (c) FPR (1-Specificity): 实际为负例的所有样本中，被预测为正例的比例， $FPR = FP \div (FP + TN)$
- 显然，使得 TPR 越接近 1，FPR 越接近 0 的分类器效果越好；
3. 对于某些分类器，其预测的结果为概率指标而非具体分类，需采取阈值实现分类，而阈值的选取则会影响具体的分类。一般来说，采用更大的阈值，能保证更高的 TPR，但 FPR 也会相应增加。此时需要一个与阈值选择无关的评价方式评价分类器的效果，即 ROC 曲线；
4. 显然，对于不同的阈值，对应不同的 TPR 和 FPR。以 TPR(Sensitivity) 为纵坐标、FPR(1-Specificity) 为横坐标，连接每一阈值所对应的 (TPR, FPR) 点对即 ROC 曲线，曲线距离左上角越近，证明分类器效果越好。

1*VI 似然与似然比检验 (Likelihood & Likelihood Ratio Test, LRT)

1. ⁵似然表示对确定的一组观测值，对应的参数取值的可信度。概率针对事件，而似然针对参数：
 - (a) 条件概率 $P(A|B = b)$: 当参数 $B=b$ 时，事件 A 发生的概率；
 - (b) 似然 $L(b|A)$: 当事件 A 发生时，参数 $B=b$ 的可信度；
2. 设总体 X 的分布为 $f(x|\theta_1, \dots, \theta_k)$ ，其中 $\theta_1, \dots, \theta_k$ 为未知参数，则称

$$L(\theta_1, \dots, \theta_k) = \prod_{i=1}^n f(x_i|\theta_1, \dots, \theta_k)$$

为参数 $\theta_1, \dots, \theta_k$ 的似然函数。其中当 X 为连续随机变量时 $f(x|\theta_1, \dots, \theta_k)$ 为 X 的概率密度函数，当 X 为离散随机变量时 $f(x|\theta_1, \dots, \theta_k)$ 为 X 的分布律 ($P(X = x_i)$)；

3. 以下通过抛硬币的例子进行说明似然的作用：
 - (a) 记抛一次硬币正面向上的概率为 p_H ，则连续抛两次硬币均为正面向上的概率为 $P(HH|p_H) = p_H^2$ ，对应的似然函数 $L(p_H|HH) = P(HH|p_H) = p_H^2$ ；
 - (b) 此时，连续抛两次硬币，观测到两次均为正面向上，在充分信任观测值的情况下即有 $P(HH) = 1$ ：
 - i. 假设 $p_H = 0.5$ ，则对应的似然值 $L(p_H = 0.5|HH) = 0.25$
 - ii. 假设 $p_H = 0.6$ ，则对应的似然值 $L(p_H = 0.6|HH) = 0.36$
- 即在连续抛两次硬币均为正面向上的情况下，相比于单次正面向上的概率为 0.5，认为单次正面向上的概率更可能为 0.6。

⁴《模型评估与选择（中篇）-ROC 曲线与 AUC 曲线》https://blog.csdn.net/qq_37059483/article/details/78595695

⁵似然函数的详细分析—似然函数的本质意义<https://blog.csdn.net/lwq1026/article/details/70161857>

(c) 同理，可以推断单次正面向上的最大概率为 1，此时似然值等于 1 达到最大，为最大似然值。

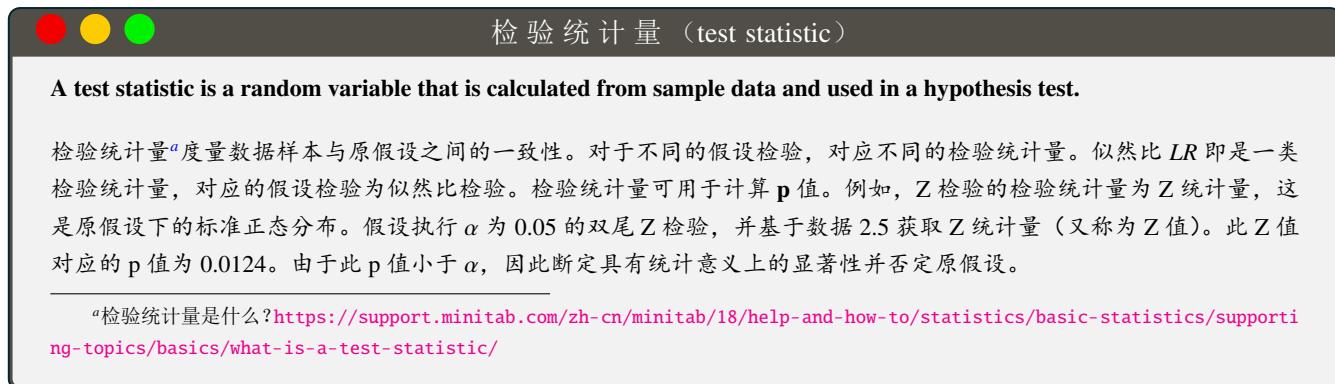
4. 一般来说，对训练的模型，其似然值越大，认为模型越有效。但需要说明的是，在确定具体模型的特征时，选择的特征越多，对模型的约束越强，似然比往往增大，所以如果仅依据似然值的大小选择特征将趋向于构建一个包含冗余特征的复杂模型，而我们的目的是使模型仅包含关键特征。所以需要进行似然比检验，确定每一个增加的特征是否对模型似然值的提升有显著影响。

5. ⁶似然比检验的实质是比较有约束条件下的似然函数最大值与无约束条件下似然函数最大值。似然比 LR :

$$LR = 2 \times \ln \frac{L_1}{L_2}$$

其中 L_1 、 L_2 分别为复杂模型和简单模型的最大似然值。 LR 近似服从卡方分布，卡方分布的自由度为两模型特征数的差，可根据卡方分布临界值表判断模型差异是否显著。

6. 不少书中都认为，在样本较小的情况下似然比检验有明显的优势，比如，要优于 logistic 回归、poisson 回归或 cox 回归的 wald 检验。大样本情况下，似然比检验和 wald 检验、score 检验等都是一致的。
7. 在对两个模型做似然比检验时需要注意以下两点：
 - (a) 简单模型是由复杂模型中简化而来，满足嵌套关系（嵌套模型，nested model）；
 - (b) 两个模型所用的观察对象完全相同。



1*VII 二次事故风险评估模型

Variables	Mean	st.d	2.50%	Median	97.50%	Elasticity(st.d)
Constant	-4.94	0.90	-6.39	-5.01	-3.35	-
Severity	-0.50	0.26	-0.94	-0.49	-0.07	-25.58 (1.91)
Sideswipe	-0.69	0.33	-1.28	-0.68	-0.18	-34.83 (2.75)
Dayweek	-1.32	0.64	-2.45	-1.28	-0.37	-59.93 (4.14)
Roadsurf	0.65	0.28	0.18	0.65	1.10	29.86 (3.84)
Lane	-0.31	0.17	-0.64	-0.29	-0.05	-1.22 (0.22)
AvgCnt	0.23	0.04	0.17	0.23	0.31	2.29 (0.78)
AvgSpd	-0.03	0.01	-0.04	-0.03	-0.02	-1.84 (0.46)
DevOcc	2.56	1.30	0.32	2.62	4.53	0.13 (0.18)
DifCnt	0.13	0.06	0.03	0.13	0.24	0.55 (0.24)
Random effect variance	0.40	0.21	0.05	0.41	0.74	-
Log-likelihood	-478.30	-	-	-	-	-

⁶关于似然比检验http://blog.sina.com.cn/s/blog_6e60da090102uxzo.html

第2章

Investigating the factors affecting secondary crash frequency caused by one primary crash using zero-inflated ordered probit regression

基于零膨胀有序 probit 回归的单次主事故引发的二次事故次数的影响因素研究

ABSTRACT

— □ ×

近年来已有多项研究讨论了二次事故发生的影响因素，然而上述研究大多仅关注二次事故的发生而未关注一次主事故可能引发的二次事故次数。本文采用零膨胀有序 probit (zero-inflated ordered probit, ZIOP) 回归模型研究实时交通环境下单次主事故可能引发的二次事故次数。

这一研究分成两个阶段。首先研究单次事故下二次事故发生的概率，而后讨论在单次主事故下二次事故可能发生的次数。研究表示上述两个阶段的主要影响因素存在较大差异。

2.1 Introduction

1. 一般事故的发生往往是随机的，与之不同二次事故的发生则大多是不可抗拒的。当前对二次事故的研究一般集中在二次事故的识别、预测和预防三个方面：
 - (a) 等速度图法是目前最常用的二次事故识别方法之一。利用等速度图可以确定每次事故的时空影响范围，而落在这一范围内的事故即被识别为二次事故；
 - (b) 对二次事故的预测主要由一些统计学模型实现，主要为离散选择模型，除了常见的 logit 和 probit 回归，这一类模型还包括 cloglog、Cox 等模型；
 - (c) 为预防二次事故的发生，现有的策略包括变速限制系统 (variable speed limit system, VSL) 和车联网技术 (connected vehicle technology)。
2. 二次事故预测模型的准确性对后续预防措施的作用至关重要。一次主事故可能引发多次事故，而现有的研究大多集中于预测二次事故是否发生，较少关注其发生的次数。少数讨论二次事故发生次数的研究中采用的是 AADT 等静态交通数据，而实时数据的效果尚未被研究；
3. 本项目旨在研究实时交通环境对单次主事故引发的二次事故次数的影响，具体来说是回答一下两个问题：
 - (a) 实时交通环境和其它变量对由单次主事故引发的二次事故次数的影响；
 - (b) 建立一个单次主事故引发的二次事故次数的预测模型 (零膨胀有序 probit (zero-inflated ordered probit, ZIOP) 回归模型)。

2.2 Data source

1. 项目采用的事故数据采集自 2010-2015 年间加州 5 号州际公路北向道，包括交通、几何、时间、气候、路面情况和事故特征等几类信息（全体候选变量见附录2*IV）；
2. 共识别出 9828 次事故（214 次主事故、304 次二次事故和 9310 次普通事故）；
3. 原始交通流数据采集自事故发生后 5-10min 内上下游探测线圈的记录值，并从中提取出 15 项交通流变量。所述 15 项交通流变量可分为均值、标准差和空间差异三类，均值和标准差用于描述交通流的总体情况和时间变异性，空间差异描述交通流的空间变异性；

2.3 Methodology

2.3.1 零膨胀有序 probit 模型 (Zero-inflated ordered probit model)

1. 在数据集中有 90% 的事故不会引发二次事故，即数据集中存在“过多的零”，这些零数据来源于两个状态：
 - (a) **secondary-crash-free**: 此时事故的发生不会引起交通流过多的紊乱；
 - (b) **secondary-crash-prone**: 此时事故的发生引起交通流紊乱，此时可能引发二次事故但也可能没有引发。
2. 采用零膨胀有序 probit (zero-inflated ordered probit, ZIOP) 回归模型以区分上述两个完全不同的状态：
 - (a) 由二元 probit 模型确定观测值所处的状态 ($k = 0$: secondary-crash-free; $k = 1$: secondary-crash-prone, k^* 表示观测值处于 secondary-crash-prone 的倾向)

$$k^* = \beta x + \varepsilon \implies P(k = 1|x) = P(k^* > 0|x) = \Phi(\beta x)$$

- (b) 当 $k = 1$ 时，建立有序 probit 模型预测二次事故次数（次数 $y = 0, 1, \dots, J$ ，对应的概率值由 y^* 表征），有序 probit 模型的具体展开见附录2*III；

$$y^* = wz + \delta$$

- (c) 将上述两个 probit 模型组合起来即可得到二次事故次数 $y = j$ 的概率

$$\begin{cases} P(y = 0|x, z) = [1 - \Phi(\beta x)] + \Phi(\beta x)\Phi(-wz) \\ P(y = j|x, z) = \Phi(\beta x)[\Phi(\mu_j - wz) - \Phi(\mu_{j-1} - wz)], \quad j = (1, \dots, J-1) \\ P(y = J|x, z) = \Phi(\beta x)[1 - \Phi(\mu_{J-1} - wz)] \end{cases}$$

3. 模型的效果由最大似然值评估，其对数似然函数为

$$l(\theta) = \sum_{i=1}^N \sum_{j=0}^J h_{ij} \ln[P(y_i = j|x_i, z_i, \theta)]$$

式中 N 为样本量；当 $i = j$ 时 h_{ij} 取 1，反之取 0。

2.3.2 Vuong 检验 (Vuong test)

详见附录2*V。

2.4 Data analysis and results

2.4.1 Results of the ZIOP models

1. 为避免多重共线性引起的偏差，计算候选特征两两间的皮尔逊相关系数；

2. 分别建立零膨胀有序 probit 模型和未考虑零膨胀的有序 probit 模型，无论是定性地比较对数似然值或是定量地 vuong 检验，均表明考虑零膨胀现象使得模型具有更高的拟合能力。具体结果见附录2*VI；
3. 对二元 probit 模型部分有显著影响的变量主要为下游的平均速度(AvgSpd_d)、上游的平均车流量(AvgCnt_u)以及上下游汇入匝道与流出匝道数量之差(Dif - ramp)。具体而言：
 - (a) 主事故发生时 AvgSpd_d 越低（往往意味着交通密度大），越容易引发二次事故；
 - (b) 主事故发生时 AvgCnt_u 越高，越容易引发二次事故；
 - (c) 主事故发生时 Dif - ramp 越高（往往意味着交通流越紊乱），越容易引发二次事故。
4. 有序 probit 模型阶段将由单次主事故引起的二次事故数目划分为 0 次、1 次、2 次、3 次或以上，共四个类别。对该部分模型有显著影响的变量主要包括雨天情况(Rain)、上游探测器的平均占用率(AvgOcc_u)、主事故是否存在伤亡(Noinjury)和主事故是否为逃逸事故(Hitrun)。具体而言：
 - (a) 主事故发生时 AvgOcc_u 越高（往往意味着交通环境拥挤），越容易引发多次二次事故；
 - (b) 主事故发生时若出于雨天，越容易引发多次二次事故；
 - (c) 若主事故为逃逸事故，越容易引发多次二次事故；
 - (d) 若主事故未引起伤亡（往往意味着对交通流稳定性的影响较低），越不容易引发多次二次事故。

因为有序 probit 回归为非线性回归，其回归系数无法用于推断特征对标签的具体影响，故计算特征的边际效应(Marginal effect)。

2.5 附录

2*I 英汉互译

English	Chinese	English	Chinese	English	Chinese
inflate	膨胀(v)	prone	有倾向的	countermeasure	对策
stochastic	随机的	deterministic	不可抗拒的	discrete	分离的
connected vehicle	车联网	excessive	过度的	evaluation	评估
milepost	里程标	meteorology	气象学	simultaneously	同时地
error term	误差项	independent and identically distributed	独立同分布	jointly	连带的
propensity	倾向	multicollinearity	多重共线性	timely	及时地
volatility	不稳定性	interpret	诠释	intermidiate	中间的
denote	表示	insight	了解		

2*II 零膨胀模型(zero-inflated model, ZIM)

1. ¹首先讨论一个问题：假设一项研究，需要调查当前交通参与者驾龄的分布情况，研究者采用问卷的形式，随机进行抽样调查。那么很可能存在以下情况：在调查者中除了拥有驾驶资格的人群之外，尚未拥有驾驶资格的人群也参与了调查，对于这类人群，毫无疑问其驾龄基本为 0。
那么如果不加区分的基于所有调查数据绘制概率分布曲线，这部分人群的贡献将主要集中在 0 值附近，那么概率曲线于 0 值处必然存在一定程度的凸起，即“膨胀”，这类人群越多，“膨胀”现象就会越显著。
2. 上述的问题被称为计数资料的零膨胀(zero-inflated)。因为一些常规的计数模型如泊松分布、负二项分布等的概率密度曲线在 0 处的值往往偏低，零膨胀现象越显著对这些分布模型的影响越大。零膨胀模型的提出即是为了在计数资料存在零膨胀的情况下依然能保证上述分布模型的预测效果。
3. 零膨胀模型是一种针对零值较多且符合泊松或负二项分布的等离散或过离散数据进行的复合计数模型。一般来说，零膨胀模型包括 Hurdle 模型、零膨胀泊松模型(zero-inflated Poisson, ZIP) 和零膨胀负二项模型(zero-inflated negative binomial, ZINB)。

¹零膨胀模型(zero inflated model) http://blog.sina.com.cn/s/blog_b5c8908c0101cuzx.html

4. 对于上述例子，显然只需要在调查的时候区分受调查者是否拥有驾驶资格即可解决，这也是零膨胀模型的思路。零膨胀模型的基本思想是把事件数的发生视为两种过程：第一种过程对应零事件的发生，假设服从伯努利分布；第二种过程对应事件数的发生，假设服从泊松分布或负二项分布。
5. 此时统计数据中的 0 被分为两个部分：“过多的 0”（extra zero）和“真实的 0”（true zero），对“0”和“非 0”分别建立二项选择模型（logit, probit）和一般计数模型（泊松、负二项），前者主要回答协变量影响事件发生与否的问题，后者主要回答协变量影响事件发生次数的问题。
6. 零膨胀模型中，随机变量 Y 的混合概率分布为

$$Y \sim \begin{cases} 0, & p_i \\ g(y_i), & 1 - p_i \end{cases}$$

p_i 表示个体来源于第一个过程的概率； $g(y_i)$ 表示个体来源于第二个过程，如果服从泊松分布即为零膨胀泊松模型，服从二项分布则为零膨胀二项模型。 $Y = y_i$ 的概率密度为

$$P(Y = y_i|0) = \begin{cases} P(Y = 0|x) = p_i + (1 - p_i)g(0) \\ P(Y = y_i|x) = (1 - p_i)g(y_i), \quad y > 0 \end{cases}$$

7. 当 p_i 的取值受个体自身协变量影响时，即 $p_i = F(\alpha x')$ ， $F(\alpha x')$ 称为零膨胀连接函数（zero-inflated link function），可选择 logit 或 probit

$$p_i(\text{logit}) = \frac{e^{\alpha x'}}{1 + e^{\alpha x'}} \quad p_i(\text{probit}) = \int_0^{\alpha x'} \frac{1}{\sqrt{2\pi}} e^{-\frac{\mu^2}{2}} d\mu$$

x' 为模型自变量，可以与 x 相同也可以不相同； α 为对应的 x' 的系数。

2*III 有序 probit 模型 (ordered porbit model)

1. probit 模型是一类常见的分类选择模型。常规的 probit 模型用于二分类问题，而在实际研究中，有很多问题属于离散多分类问题，为此提出了一系列改进模型，有序 probit 模型即是其中之一。
2. 考虑两类多分类问题。对于第一类问题，选择集包含多个类别且多个类别之间为并列关系，例如职业倾向，其选择可包括工程师、科学家、服务人员等多个并列的职业；而对于第二类问题，选择集包含的多个类别存在明显的次序关系，例如用户满意程度，其选择可能是满意（1）、一般（2）、不满意（3）等。对于第一类多分类问题，可采用多项 probit (logit) 模型；而对于后者则适用于有序 probit (logit) 模型。
3. 有序 probit 模型的具体展开如下：

$$\begin{aligned} y^* &= \beta X + \varepsilon, \quad \varepsilon \sim N(0, 1) \\ y &= \begin{cases} 0, & y^* \leq \alpha_1 \\ i, & \alpha_i < y^* \leq \alpha_{i+1} \\ N, & y^* > \alpha_N \end{cases} \\ \text{Probit}(y) &= \begin{cases} \text{Probit}(y = 0|X) = \Phi(-\beta X) \\ \text{Probit}(y = i|X) = \Phi(\alpha_i - \beta X) - \Phi(\alpha_{i-1} - \beta X) \\ \text{Probit}(y = N|X) = 1 - \Phi(\alpha_N - \beta X) \end{cases} \end{aligned}$$

2*IV 候选变量及描述

Symbol	Description
$AvgCnt_u$	Average vehicle count at the upstream station during 5-10 min period (veh/30s)
$AvgOcc_u$	Average detector occupancy at the upstream station during 5-10 min period (%)
$AvgSpd_u$	Average vehicle speed at the upstream station during 5-10 min period (mile/h)
$DevCnt_u$	Std. dev. of vehicle count during at the upstream during 5-10 min period (veh/30s)
$DevOcc_u$	Std. dev. of detector occupancy during at the upstream during 5-10 min period (%)

Symbol	Description
$DevSpd_u$	Std. dev. of vehicle speed during at the upstream during 5-10 min period (mile/h)
$AvgCnt_d$	Average vehicle count at the downstream station during 5-10 min period (veh/30s)
$AvgOcc_d$	Average detector occupancy at the downstream station during 5-10 min period (%)
$AvgSpd_d$	Average vehicle speed at the downstream station during 5-10 min period (mile/h)
$DevCnt_d$	Std. dev. of count during at the downstream during 5-10 min period (veh/30s)
$DevOcc_d$	Std. dev. of occupancy during at the downstream during 5-10 min period (%)
$DevSod_d$	Std. dev. of speed during at the downstream during 5-10 min period (mile/h)
$DifCnt$	Count difference between upstream and downstream in 5-10 min interval (veh/30s)
$DifOcc$	Occupancy difference between upstream and downstream in 5-10 min interval (%)
$DifSpd$	Speed difference between upstream and downstream in 5-10 min period (mile/h)
$Width_w$	1 = Inner shoulder width >12 ft; 0 = otherwise
$Width_n$	1 = outer shoulder width >12 ft; 0 = otherwise
$Width_r$	Road width (ft)
$Width_l$	Lane width (ft)
$Width_i$	Inner median width (ft)
$Bottlenecks$	1 = bottleneck section; 0 = otherwise
$On-ramp$	Number of on-ramps between upstream and downstream stations
$Off-ramp$	Number of off-ramps between upstream and downstream stations
$Dif-ramp$	The difference between the numbers of on-ramp and off-ramp
$Rain$	1 = adverse weather conditions (rain); 0 = otherwise
Fog	1 = adverse weather conditions (fog); 0 = otherwise
$Visibility$	Hour visibility
$Temp$	Hour temperature
$Humid$	Hour relative humidity (%)
$WndSpd$	Hourly wind speed (m/s)
$Peak$	1 = 07:00-09:00, 16:30-20:00; 0 = other time
$NoInjury$	1 = no injury; 0 = otherwise
$Fatality$	1 = call an ambulance; 0 = otherwise
$HitRun$	1 = hit and run; 0 = otherwise

2*V vuong 检验

1. ²Vuong 检验是一种通过最大似然值比较两模型之间拟合效果的一种检验方法，与似然比检验相反的是，其要求两个模型为非嵌套模型（non-nested model）；
2. vuong 检验的零假设 H_0 为两个模型 $g_1(x)$ 、 $g_2(x)$ 对数据的拟合效果相同，其中模型对数据的拟合情况由 KL 散度（Kullback–Leibler divergence, KLD）³ 表征

$$\begin{aligned}
 H_0 : D_{KL}(g_t||g_1) &= D_{KL}(g_t||g_2) \\
 D_{KL}(g_t||g) &= \sum_{y=0}^{\infty} g_t(y) \ln \frac{g_t(y)}{g(y)} = \sum_{y=0}^{\infty} g_t(y) \ln g_t(y) - \sum_{y=0}^{\infty} g_t(y) \ln g(y) = E_{g_t}[\ln g_t(y)] - E_{g_t}[\ln g(y)] \\
 \therefore H_0 : \sum_{y=0}^{\infty} g_t(y) \ln g_1(y) - \sum_{y=0}^{\infty} g_t(y) \ln g_2(y) &= E_{g_t}[\ln g_1(y)] - E_{g_t}[\ln g_2(y)] = 0
 \end{aligned}$$

式中 g_t 表示事件的真实分布， $E_{g_t}[\ln g(x)]$ 表示基于 g_t 的预测模型 g 的对数似然值的期望⁴；

²Testing for zero inflation in count models: Bias correction for the Vuong test. <https://www.stata-journal.com/article.html?article=st0319>

³KL 散度，又称相对熵（relative entropy）、信息散度（information divergence），是描述两个概率分布 P、Q 差异的一种非对称度量 ($D_{KL}(P||Q) \neq D_{KL}(Q||P)$)， $D_{KL}(P||Q)$ 表示用概率分布 Q 拟合真实分布 P 时产生的信息损耗。 $D_{KL}(P||Q) \geq 0$ ，当且仅当分布 $P = Q$ 时有 $D_{KL}(P||Q) = 0$ 。

⁴期望： $E(X) = \sum_{i=0}^{\infty} x_i p_i$

3. 构造 $m_i = \ln g_1(y_i|x_i) - \ln g_2(y_i|x_i)$, 即两模型对数似然只差, 由此有 vuong 检验的检验统计量 V

$$V = \frac{\sqrt{N} \left(\frac{1}{N} \sum_1^N m_i \right)}{\sqrt{\frac{1}{N} \sum_1^N (m_i - \bar{m})^2}} = \frac{\sqrt{N} \cdot \bar{m}}{S_m}$$

式中 \bar{m} 、 S_m 分别为 m_i 的均值和标准差;

4. 首先说明统计量 \bar{m} 为 $E_{g_t}[\ln g_1(y)] - E_{g_t}[\ln g_2(y)]$ 的一致（相合、相容）估计量（consistent estimator）:

$$\lim_{n \rightarrow \infty} \bar{m} = \lim_{n \rightarrow \infty} \sum_1^n \left[\frac{1}{n} \ln g_1(y_i|x_i) - \frac{1}{n} \ln g_2(y_i|x_i) \right] \longrightarrow \lim_{n \rightarrow \infty} \sum_1^n [g_t(y_i) \ln g_1(y_i|x_i) - g_t(y_i) \ln g_2(y_i|x_i)]$$

所以, 在满足 H_0 的条件下, 根据中心极限定理, \bar{m} 近似服从均值为 0、标准差为 S_m / \sqrt{N} 的正态分布, 则统计量 V 近似服从标准正态分布 (V 即 \bar{m} 的标准化), 且这个统计量为双向的。



中心极限定理 (central limit theorem)

对任意分布的总体 (定义良好, 均值、标准差存在), 抽取多组样本, 任意一个总体的样本平均值都会围绕在总体的整体平均值周围, 并且呈正态分布。记抽样分布的均值、方差分别为 \bar{X}, S_n^2 , 原分布的均值、方差分别为 μ, σ^2 , 抽样次数为 n , 此时有:

$$E\bar{X} = \mu, \quad D\bar{X} = \frac{\sigma^2}{n}, \quad ES_n^2 = \sigma^2$$

5. 在双侧显著水平为 0.05 的条件下, 当 $V \geq 1.96$, 选择模型 1; 当 $V \leq -1.96$, 选择模型 2; 当 $-1.96 < V < 1.96$, 说明 vuong 检验不支持任何一个模型。

6. 需要说明的是, 检验统计量 V 是 $D_{KL}(g_t||g_1) - D_{KL}(g_t||g_2)$ 的相合估计, 但同时也是有偏估计 (biased estimator), 有偏是因为在模型参数确定和检验统计量计算的过程中采用了相同的数据, 而且当两个模型采用的参数个数不相同时 (p_1, p_2), 这一偏差会增强。为此常常通过特定的准则对偏差进行修正。例如可以为 m_i 增加修正项

$$m_i^c = m_i + \frac{p_2 - p_1}{n} \quad \text{or} \quad m_i^c = m_i + (p_2 - p_1) \frac{\ln n}{2n}$$

另外也可以采用留一交叉验证⁵的方法消除偏差。

2*VI 由单次事故引起的二次事故次数预测模型 (ZIOP)

Type	Variable	Estimate	P-value
Binary probit process	AvgSpd _d	-0.962	< 0.001
	AvgCnt _u	0.383	< 0.001
	Dif-ramp	0.905	< 0.001
	Constant	-0.854	< 0.001
Ordered probit process	AvgOcc _u	0.017	< 0.001
	Rain	0.284	0.018
	NoInjury	-0.252	< 0.001
	Hitrun	0.341	< 0.001
Cutting point	Cut point 1	1.983	-
	Cut point 2	2.524	-
	Cut point 3	2.913	-
LL comparison		-1039.7(ZIOP) vs. 1065.4(OP)	
Vuong's test		ZIOP model vs. OP model: z = 9.48	

⁵留一交叉验证 (leave-one-out cross-validation, LOOCV): 一种极端的数据集分割的方式。如果数据集 D 的大小为 N , 那么用 $N-1$ 条数据进行训练, 用剩下的一条数据作为验证, 并循环 N 次。

Marginal effects Variables	Possibility	P(y = 0)	P(y = 1)	P(y = 2)	P(y ≥ 3)
AvgOcc _u		-0.0016	0.0004	0.0002	0.0001
Rain		-0.0273	0.0075	0.0026	0.0019
Noinjury		0.0243	-0.0066	-0.0024	-0.0017
Hitrun		-0.0328	0.009	0.0032	0.0023

第3章

Use of ubiquitous probe vehicle data for identifying secondary crashes

使用广泛的车载探测器探测数据识别二次事故

ABSTRACT

二次事故的出现次数是评估事故管理系统的一个重要指标，因此对二次事故的识别就显得尤为重要。目前已有一系列的识别二次事故的方法，如静态法、速度等值线图法、震荡波法 (shockwave-based) 等等，但这些现有方法受制于一些缺陷，如需要额外的事故信息及简化猜想等等。

本文提出一种新的基于数据的二次事故识别方法，旨在基于广泛的**车载探测器探测数据**（无需额外信息）识别二次事故。二次事故识别的过程分为以下三个部分：

1. 探测主事故的影响区域（基于聚类方法标记主事故引起的拥挤范围）；
2. 基于元启发式优化算法 (metaheuristic optimization algorithms) 识别影响区域的边界；
3. 位于边界内的事故即被识别为二次事故。

将方法应用于仿真情景中进行测试，测试结果表明，基于蚁群优化算法 (ant colony optimization algorithm) 的识别方法具有最好的识别性能 (95%)。

3.1 Introduction

1. 交通事故造成了大量的经济损失，同时也会增大其它事故发生的风脸，并且主事故持续时间越长，发生二次事故的概率也越大，且二次事故的发生又可能对其他交通参与者和事故处理人员造成伤害。二次事故占全美事故约 20%，对二次事故的预防尤为重要；
2. 二次事故发生于主事故上游，可能与主事故同向，也可能对向（由于对向驾驶员回头看主事故）；
3. 识别二次事故是研究其预防方法的必要条件，已有的一些后事件 (post-event) 的识别方法如队列模型、速度等值线图、事故影响区边界确定等方法大多需要各类简化假设。同时，交通环境的变异性也增大了识别的难度；
4. 本文旨在基于广泛的**车载探测器探测数据**建立更精确的二次事故识别方法，方法包括主事故影响区确定、影响区边界定量计算和二次事故识别三部分。本方法可更精确地捕捉事故的影响，并更快地识别二次事故。

3.2 Literature review

1. *Static spatiotemporal threshold-based method*

静态时空阈值法通过划定静态的事故时间-空间影响范围进行二次事故识别，这一方法无法适应多种复杂的交通环境，易存在过估计和欠估计的问题；

2. *Queuing-based method*

队列理论基于排队论，根据一系列自变量计算冲突发生后的队列长度，从而确定主事故的影响范围。纳入考虑的自变量可以包括事故持续时间、事故发生时间、事故严重程度、封锁车道数、上游车辆到达率、下游消散率、最大容量等等。队列理论改善了对二次事故的识别效果，但依然存在如下缺点：

- (a) 不同的路段因其几何特征和交通特征的差异排队模型不同，不能用相同的排队模型进行描述，对每一路段建立各自的排队模型又不切实际；
- (b) 精确的排队模型需要大量参数，精确采集这些参数难以实现；
- (c) 排队模型中存在一些不合理的简化假设，例如假设主事故引起的最长队列发生于主事故消散时期等等。

3. Speed contour plot-based method

基于探测器数据和事故数据绘制二元速度等值线图识别二次事故是一类数据驱动的方法，可有效预测主事故的时空影响范围。这一方法的关键在于基准速度的定义。基准速度可以简单的由速度分位值定义，也可以借助一些复杂的模型如高斯混合模型 (Gaussian Mixture Model, GMM) 等。

这一方法的缺陷在于其需要大量的传感器数据——大量的历史数据用于定义基准速度，大量的事故发生前后的数据用于确定事故的时空影响范围。然而存在很多的路段因为种种原因未配备足够的探测器，无法提供足够的数据。为解决这一问题，可以通过一些第三方网络地图平台补充数据，但又会增加额外的工作；

4. Shockwave-based method

这一方法基于交通流理论中的车流波识别事故的时空影响范围。一般来说，交通流状态的变化将产生车流波，车流波的边界即为对应状态的边界（图 3.1）。事故将会产生队列波和消散波，具体的分析过程如下：

- (a) 假设事故 A 发生的时刻和位置分别为 t_A, s_A ，此时因为车道封闭将会产生向后传播的队列波，波的速度为 ω_{A1} ；
- (b) 记事故被移除车道重新恢复开放的时间为 t_0 ，此时拥堵开始消散，形成同样向后传播的消散波，速度记为 ω_{01} ，且有 $\omega_{01} > \omega_{A1}$ ，因此消散波将会追上队列波，意味着拥堵完全消散，记此时时间为 t_1 ，假设以上过程中 ω 均为常数¹，则易得

$$t_1 = \frac{\omega_{A1} t_A - \omega_{01} t_0}{\omega_{A1} - \omega_{01}}, \quad \omega = \frac{\Delta q}{\Delta k}$$

$\Delta q, \Delta k$ 分别表示流量和密度的变化量；

- (c) 假设另一事故 B 发生的时刻和位置分别为 t_B, s_B ，且有 $t_B > t_A, s_B < s_A$ ，则当下式满足时，事故 B 即为二次事故

$$\frac{s_B - s_A}{\omega_{A1}} + t_A \leq t_B \leq \frac{s_B - s_A}{\omega_{01}} + t_0$$

然而这一方法也不可避免地存在以下缺陷：

- (a) 方法需假设车流波的传递速度和事故发生前、中、后的交通流状况及其交通流参数；
- (b) 方法同样需要大量的探测器数据和事故信息以实现车流波的预测；
- (c) 现有的研究仅讨论了引起排队的主事故的情况，而未考虑无排队现象的主事故；
- (d) 方法难以优化。在实际的交通环境中，复杂的交通现象将产生速度方向各不相同的复杂车流波，难以通过简单模型进行拟合。

3.3 Methodology

3.3.1 Detecting the impact area

1. 事故的发生将改变原有的交通状态，事故消散后交通状态又将得到恢复，因此识别事故的发生及其影响范围的大小应属于无监督学习的过程。在已知事故信息的前提下，基于探测车 j 于第 i 个探测步长所探测的时间、速度、位置 (t_{ij}, v_{ij}, s_{ij}) ，通过聚类方法可以实现事故影响范围的识别；

¹也有学者提出救援人员的行动将使得原有车流波发生改变，为一段函数

2. 聚类的簇可以固定为 2, 即仅区分是否受事故影响两类情况, 不考虑离群数据, 因此可以选择 K 均值 (K-means) 聚类。K-means 聚类假定所有观测值各自属于特定的簇, 然而实际事故的影响范围将不会有清晰的边界, 因此选择模糊 C(Fuzzy C-means, FCM) 均值聚类更为合适。

3.3.2 Estimation boundary of the impact data

1. 由聚类方法得到事故影响范围后, 还需对影响范围边界进行高效识别: 移除边界内部点并由均值滤波器去除边界噪点, 从而定量描述影响范围的时空特性;
2. 考虑到两个连续的测点之间或许间隔了很长的一段时间, 通过上述方法直接获得的边界未必能反映实际的影响范围, 因此将对边界进行拟合。首先将从构成边界的全部点集中抽取一部分作为特征点, 特征点定义了边界的基本形状。两个连续的特征点定义一段直线段, 拟合的边界多段直线段闭合围成。选择三种方法定义特征点并对边界进行拟合, 分别是多段线性拟合、基于遗传算法的方法和基于蚁群优化的方法;

3. Multi-stage linear approximation, 多段线性拟合

多段线性拟合算法中特征点的确定方法如下:

- (a) 定义 $\theta_{i-1,i}, \theta_{i,i+1}$ 分别表示边界点 i 与其前后两个点的连线的转角;
- (b) 定义 $\Delta\theta_i = |\theta_{i-1,i} - \theta_{i,i+1}|$ 表示边界点 i 与其前后两个点的连线的相对转角, 从而组成集合 $\Delta\theta = \{\Delta\theta_1, \dots, \Delta\theta_n\}$;
- (c) 当满足 $\Delta\theta_i > \max \Delta\theta \cdot \tau$ 时, 点 i 即为特征点, $\tau \in (0, 1)$ 用于调节特征点的数量, τ 越大选取的特征点越少, 反之亦然。

对相邻两个特征点及其之间的所有非特征点进行线性拟合, 拟合的多段直线形成的闭合边界即认为是事故的影响边界。当相邻两拟合直线未交于特征点时, 以一通过该特征点的竖线作为补充边界连接相邻两端直线;

4. Genetic algorithm based estimation

多段线性拟合往往面临补充边界的问题, 另外由于每段线性回归相互独立, 整体的残差未必是最优的。为了保证整体残差的最优化, 提出一种基于遗传算法的最优边界识别方法。

遗传算法中一条染色体代表一个闭合边界, 染色体中的每个基因为候选点集中的一个点, 染色体的长度(即包含的基因数)代表闭合边界的边数, 为自定义。将单个染色体中的所有基因所代表的点首尾相接即形成闭合边界。计算构成闭合边界的每一子边界的残差, 得到整个闭合边界的残差和, 残差和的倒数即是该染色体所代表个体的适应度。经过多代选择、重组与变异, 即可得到使得适应度最高的染色体;

5. Ant colony optimization based estimation

蚁群算法是另一类优化算法, 同样可以优化闭合边界的整体残差。蚁群算法的原理与遗传算法不同, 但整体结构类似, 与以上两种算法的区别在于无需预设构成闭合边界的特征点个数。蚂蚁自一点出发不回头地移动直至回到起点, 期间可以自由决定每一步跳过多少个点。蚂蚁的位置更新基于备选路径的信息素水平和启发(路径长度), 而信息素的更新与路径的残差和有关, 残差和越小更新的信息素越多。

3.3.3 Automated identification of secondary crashes & Ray casting algorithm

1. 理论上, 只需绘出影响边界及事故位置, 即可判断事故是否位于边界内, 从而判断事故是否为二次事故。然而人工识别需要耗费大量的精力, 自动识别二次事故具有积极的意义;
2. 上述过程本质上是一个判断点是否在多边形内部 (Point-in-Polygon, PIP) 的问题。有多种算法可解决 PIP 问题, 光线投影算法 (Ray casting algorithm) 即是一种简单且高效的方法。算法基于奇偶规则 (even-odd rule), 假设从某一点引一射线, 若射线与边界有偶数个交点, 则点位于边界外; 反之位于边界内;
3. 已知围成闭合边界的所有特征点坐标及待判断点坐标, 为方便分析假设射线为直线, 则由解析几何知识可轻松判断射线与边界的交点。

3.4 Simulation test & Data analysis and result

- 通过仿真检验提出的方法。选择 Paramics 微观交通仿真平台，具有内建的事故模块模拟事故。路段为 10mile 长的直线段，两车道，限速 65mile/h，交通需求 2400veh/h。一小部分仿真车视为探测车，每 0.1s 提取其轨迹 (t, s, v)。仿真模拟不同事故位置、不同事故发生、持续时间、不同探测车市场渗透率²等多种场景，每一场景仿真时间为 75min；
- 以每一轨迹点的速度为指标，由 K-means 和 Fuzzy C-means 进行聚类，簇的个数定义为 2，即可得到事故的影响范围。Fuzzy C-means 聚类的结果为每一坐标点的隶属度，隶属度介于 0 至 1，设定一阈值 (0.5) 即可将坐标点分为 2 类；
- 得到事故影响范围后提取其边界，并进行平滑降噪，在由多段线性拟合、遗传算法和蚁群算法共三种方法分别进行边界拟合。其中多段线性拟合和遗传算法都需要指定构成闭合边界的边数，而蚁群算法则是一种完全的无监督算法，且在本次仿真中蚁群算法的运算速度较遗传算法更快；
- 算法效果检验的实质是比较拟合的影响范围与实际的影响范围的相似性。**具体验证方法可以由蒙特卡洛实现：从原始的经 Fuzzy C-means 聚类后的点集中随机抽取部分点，根据得到的隶属度标签将其分为拥堵与非拥堵两类，即判断其是否处于事故影响范围内，再根据拟合的边界判断这些点是否位于影响范围内，由以上两结果得到混淆矩阵，从而判断算法优劣；
- 经检验，遗传算法与蚁群算法所拟合的边界较多段线性拟合更接近原影响范围，且在探测车渗透率大于等于 15% 的情况下蚁群算法的结果略优于遗传算法。另外，在探测车渗透率大于等于 15% 的情况下即可保证较理想 (> 90%) 的相似度。

3.5 附录

3*I 英汉互译

English	Chinese	English	Chinese	English	Chinese
ubiquitous	十分普遍的	performance measure	性能指标	shock-wave	震荡波
untapped	未开发的	metaheuristic	元启发式算法	novel	新颖的
colony	殖民地、聚居群	penetration	渗透 (n)	notable	显著的
fund	基金、拨款 (v)	allocate	分配 (v)	complementary	互补的
post-event	事件后	heterogeneous	由多类组成的	automatic	使自动化
sake	缘故	leverage	影响力	state-of-the-art	最先进的
deterministic	不可抗拒的	polynomial	多项式	high-order	高阶
construe	理解 (v)	prevail	流行、战胜 (v)	clearance time	清空时间
premise	假定 (n)	complicate	使复杂化	sparse	稀少的
scalable	可攀登的、可称量的	saturate	使饱和	piece-wise	分段的
crude	粗略的	consecutive	连续的	induce	诱使、引发
bottleneck	瓶颈	rubberneck	(驾驶时) 扭头 回看 (v)	milepost	里程标
instance	例子、举例 (v)	onset	开端 (n)	intuitively	直觉地
heavy-duty	结实的	centroid	质心	transient	转瞬即逝的
fuzzy	模糊的	membership	成员资格	degree of membership	隶属度
evolve	逐渐形成 (v)	vice versa	反之亦然 (adv)	vice	罪行
residual	剩余的	stochastic	随机的	mutation	变异 (n)
a population of	一群	chromosome	染色体	roulette	轮盘赌 (n)
with respect to	关于	fixation	固恋 (n)	forage	觅食 (v)
swarm	一大群 (n)	even-odd	奇偶的	odd	奇数的
even	偶数的	vertex	(三角形) 角点	trajectory	轨迹

²市场渗透率 (market penetrate rate): 预计某一产品所能占据的最大市场份额

English	Chinese	English	Chinese	English	Chinese
ground truth	正确的带标签数据	fraction	小部分		

3*II 模糊 C 均值 (Fuzzy C-means, FCM) 聚类

1. ³1965, L. A. Zadeh 发表模糊集合 “Fuzzy Sets” 的论文, 首次引入隶属度函数的概念, 打破了经典数学“非 0 即 1”的局限性, 用 [0,1] 之间的实数来描述中间状态, 即隶属度 (degree of membership);
2. 经典的 K 均值聚类属于硬聚类, 即每个观测值只能被归为一个簇。模糊 C 均值聚类在 K 均值聚类的基础上引入隶属度的概念, 属于模糊聚类, 由观测值的隶属度表示其属于某个簇的程度。与 K 均值聚类类似, 模糊 C 均值聚类的优化目标同样是使得簇中心至观测值的距离最小, 但因为隶属度的存在, 需要同时优化隶属度和簇中心两个变量, 目标函数 J 表示全体簇中心至全体观测值的距离加权和

$$J_m = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|x_j - c_i\|^2, \quad \sum_{i=1}^c u_{ij} = 1$$

式中 c, n 分别表示簇及观测值的个数; $\|x_j - c_i\|^2$ 分别表示第 j 个观测值至第 i 个簇中心的距离 (不一定是欧氏距离); u_{ij} 表示第 j 个观测值对第 i 个簇的隶属度; m 为隶属度的因子, 位于指数项, 推荐取 2;

3. 因为带约束的优化问题, 采用拉格朗日乘数法得

$$\begin{aligned} u_{ij}, c_i &= \arg \min_{u_{ij}, c_i} J_{m,\lambda} = \arg \min_{u_{ij}, c_i} \left\{ \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|x_j - c_i\|^2 + \lambda_1 \left[\sum_{i=1}^c u_{i1} - 1 \right] + \cdots + \lambda_c \left[\sum_{i=1}^c u_{ij} - 1 \right] + \cdots + \lambda_n \left[\sum_{i=1}^c u_{in} - 1 \right] \right\} \\ \frac{\partial J_{m,\lambda}}{\partial u_{ij}} = 0 &\implies m u_{ij}^{m-1} \|x_j - c_i\|^2 + \lambda_j u_{ij} = 0 \implies u_{ij} = \left(-\frac{\lambda_j}{m \|x_j - c_i\|^2} \right)^{\frac{1}{m-1}} \\ \frac{\partial J_{m,\lambda}}{\partial c_i} = 0 &\implies -2 \sum_{j=1}^n u_{ij}^m (x_j - c_i) = 0 \implies c_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m} \end{aligned}$$

4. 注意到 u_{ij}, c_j 的表达式互相关联, 因此可以首先假设 u_{ij} (或 c_j), 求得 c_j (或 u_{ij}), 再反代得到 u_{ij} (或 c_j) 循环直至收敛。以上即为模糊 C 均值聚类算法的优化过程;
5. 上式 u_{ij} 的计算式中包含未知参数 λ_j , 需由约束条件反解, 避免为迭代带来不便

$$\begin{aligned} \therefore 1 &= \sum_{i=1}^c u_{ij} = \sum_{i=1}^c \left(-\frac{\lambda_j}{m \|x_j - c_i\|^2} \right)^{\frac{1}{m-1}} = \left(-\frac{\lambda_j}{m} \right)^{\frac{1}{m-1}} \sum_{i=1}^c \left(\frac{1}{\|x_j - c_i\|^2} \right)^{\frac{1}{m-1}} \\ \therefore \left(-\frac{\lambda_j}{m} \right)^{\frac{1}{m-1}} &= \frac{1}{\sum_{k=1}^c \frac{1}{\|x_j - c_k\|^{m-1}}} \\ \therefore u_{ij} &= \left(-\frac{\lambda_j}{m \|x_j - c_i\|^2} \right)^{\frac{1}{m-1}} = \frac{1}{\sum_{k=1}^c \frac{1}{\|x_j - c_k\|^{m-1}}} \frac{1}{\|x_j - c_i\|^{\frac{2}{m-1}}} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_j - c_k\|}{\|x_j - c_i\|} \right)^{\frac{2}{m-1}}} \end{aligned}$$

6. 观察 c_j, u_{ij} 的定义式, K 均值聚类中簇中心为簇中所有观测值的平均值, 而模糊 C 均值聚类中簇中心为所有观测值按对于隶属度加权平均, 而某一观测值对某一簇的隶属度为该观测值至全体簇中心总距离与该观测值至该簇中心的距离之比。

3*III 参考图片

³聚类之详解 FCM 算法原理及应用: <https://blog.csdn.net/on2way/article/details/47087201>

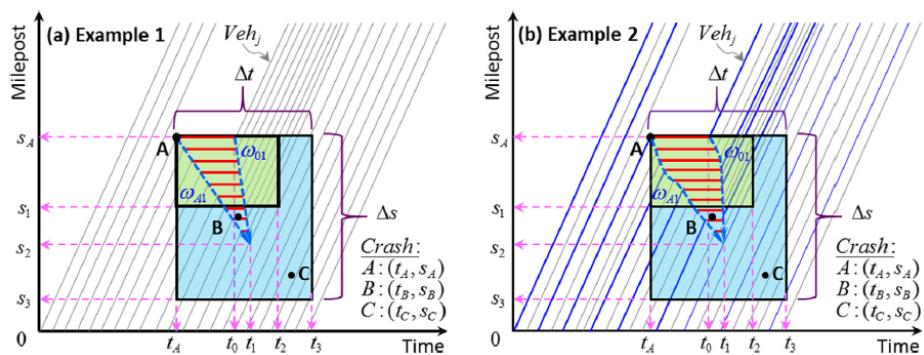


图 3.1 基于车流波理论的二次事故识别图示

第4章

Variable speed limit control at fixed free-way bottlenecks using connected vehicles

ABSTRACT

研究基于网联车（connected vehicle, CV）技术以建立旨在提升瓶颈流量和缓解系统拥堵的动态限速策略。根据复杂度和保持交通稳定性能力的差异，策略可分为以下三类：

- 策略一：每车道一辆网联车；
- 策略二：每车道一辆网联车，辅之以少量可变信号标识（variable message sign, VMS）；
- 策略三：多辆网联车。

进一步地为了缓解每一策略的潜在缺陷对三者进行动态组合得到动态控制方案，以应对不同的车队探测方案（由网联车探测或传感器探测）和网联车渗透率。最后，基于与交通稳定性有关的随机特征评估每一策略的失效概率以评估延误节省时间、控制策略稳定性并决定最优的控制速度。研究显示基于网联车的动态限速策略具有以下优点：

- 更快的消散速度；
- 更高的控制速度与更温和的控制强度；
- 仅通过一辆或少量网联车即可实现。

4.1 Introduction

1. 公路在瓶颈处存在明显的容量下降问题，即消散率在达到临界密度后即明显降低 10% ~ 30%。这一现象很可能是由于换道、驾驶时扭头观看和交通流时间参数的变化特性（例如瓶颈处车头时距的明显增加）造成的；
2. 动态车速限制原本用于保障行车安全，后被发现可有效缓解瓶颈处流量下降问题。早期研究指出动态限速可达到更高的通行能力和临界密度、更小的速度差与更一致的车道利用。已有的可变限速控制往往是由可变信号标识（variable message sign, VMS）实现的，从而具有以下缺点：
 - 可变信号标识部署于道路上，往往是离散且静态的，难以感知大空间内的动态交通环境并及时做出指令；
 - 作为道路设施，这些信号标识往往需要大量经费，部分设备同样需要足够的空间以发挥效果；
 - 另外信号标识需要具探测器足够近，还需要处理驾驶员的不满和法律的强制性规定。
3. 与传统方法相比，网联车技术提供了一种新型的交通探测和可变限速控制的方法。网联车可以实现车 - 车、车 - 设施之间的通信，为更高效、恰当的控制提供了可能。通过网联车进行交通控制时需考虑以下几点：
 - 网联车渗透率。需要用尽可能少的网联车最大限度地改变其它车的行为；
 - 瓶颈容量的随机性。实验表明，公路的最大流量是可变的，瓶颈的容量分布与流量或速度有关，而目前大多数研究假设这一容量为定值；

- 控制速度的大小。控制速度的大小及受其影响的交通环境将影响控制策略失效的概率，最优的控制策略应该使得瓶颈消散率的期望值达到最大。

4. 本文的目标是建立一套基于网联车的旨在缓解固定瓶颈拥堵的可变限速策略。假设车辆中有一部分网联车，策略的主要目标是当瓶颈发生容量下降时，调整上游网联车速度于稳定交通流（无通行能力下降）所对应的最大速度。此时网联车前部形成的空隙将有利于已有车队的消散，而网联车后的来车将被约束以维持稳定交通流，从而避免容量下降并提升消散率。为应对不同的车队探测方案（由网联车探测或传感器探测）和网联车渗透率，策略可分为以下三类：

- 策略一：每车道一辆网联车；
- 策略二：每车道一辆网联车，辅之以少量可变信号标识；
- 策略三：多辆网联车。

考虑到交通稳定性的随机性，计算以上每一策略的失效概率以确定使得瓶颈消散率期望值最大的最优限速。

4.2 VSL strategies using connected vehicle(s)

本文的研究场景为高速路上因需求高于固定交通瓶颈¹通行能力而形成的拥堵路段，并为提出理论进一步作出如下假设：

- 车流由网联车和常规车组成；
- 网联车为受控制时同常规车具有相同的行为；
- 基于交通波理论；
- 仿真车完全服从速度控制；
- 瓶颈路段可以短暂地支持高于平均通行能力的车流；
- 单车道交通环境，无换道现象。

4.2.1 Baseline case

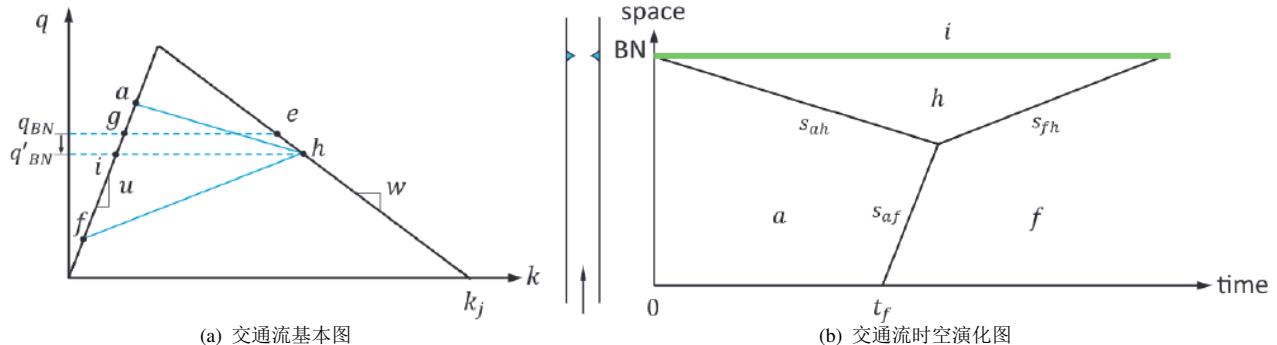


图 4.1 无控制状态下的交通流演化图

1. 以上为无控制状态下的交通流演化图。图 4.1(a) 称为交通流基本图 (fundamental diagram, FD)²。折线上升段表示自由流阶段，斜率 u 为自由流速度，对应于瓶颈下游的交通状况，下降段表示拥堵阶段，斜率 w 为拥堵波速度，对应于瓶颈上游的交通状况。 q_{BN} 表示瓶颈的最大通行能力， q'_{BN} 为发生通行能力下降后的瓶颈最大通行能力；
2. 假设初始时刻汇入的交通流状态为 a ，因为该状态的流量高于瓶颈的最大通行能力，所以上游的交通流状态将受限表示为 h ，而瓶颈下游为自由流，但流量与 h 状态一致，为状态 i 。交通流状态的变化产生向

¹ 交通瓶颈是产生交通拥挤的路段不断向下游传播，并进一步引发交通拥挤的一种交通现象，根据其产生原因是否具有稳定性和可预见性分为固定交通瓶颈和动态交通瓶颈。其中固定交通瓶颈主要指产生交通瓶颈的源头，其特点是具有可预见性。

² 交通流基本图：反应交通环境中流量和密度关系的曲线，曲线的斜率即为交通流速度，因此图中能同时反应交通流三参数——速度、密度、流量，故被称为交通流基本图。交通流基本图近似为抛物线形，为了表示方便又往往表示成三角形或梯形。

- 后扩散的集结波 s_{ah} (图 4.1(b)), 为汇入的自由流 a 与拥堵状态 h 的分界线, 集结波的速度也用 s_{ah} 表示;
3. 假设从时间 t_f 开始上游汇出自由流状态变为流量更低的 f , 则状态 a, f 之间同样会产生交通波, 为向前扩散的消散波 s_{af} , 速度也记为 s_{af} 。向前扩散的消散波 s_{af} 与向后扩散的集结波 s_{ah} 相遇后会形成新的交通波 s_{fh} ;
 4. 因为状态 f 的流量低于瓶颈的最大通行能力, 因此 s_{fh} 依然是向前扩散的消散波, 但因为 s_{ah} 的影响速度会变慢。

4.2.2 Strategy 1: control using single connected vehicle

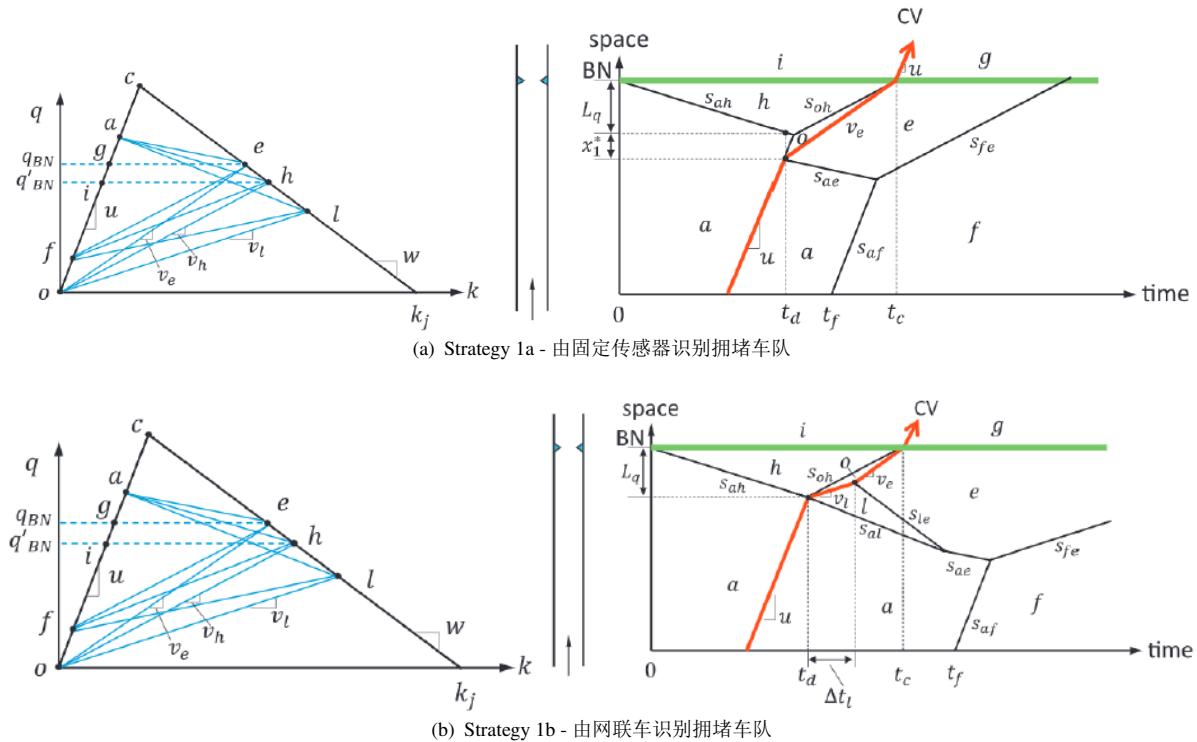


图 4.2 仅由一辆网联车控制的交通流演化图

1. 这一方法的基本思想是当探测到拥堵车队形成后, 控制网联车速度将上游来车的流量限制为瓶颈的最大通行能力 q_{BN} 。当网联车渗透率较低时, 单靠网联车无法实现准确的拥堵车队识别, 需要辅之以固定传感器 (图 4.2(a)), 当网联车渗透率较高时, 网联车本身即可实现较精确的拥堵车队识别 (图 4.2(b))。图中红线为网联车轨迹线;
2. 由固定传感器实现车队识别 (图 4.2(a) 所示情况):
 - (a) 假设初始时上游交通流状态为 a 、速度为 u , 因其流量超过了瓶颈的最大通行能力 q_{BN} , 故瓶颈处发生通行能力减小, 交通流状态变为 h , 同时形成向后传递的集结波 s_{ah} , 而瓶颈下游交通流状态变为 i ;
 - (b) 假设探测器布置在距瓶颈上游 L_q 处, 则当 s_{ah} 推进至该处时拥堵车队将被探测器识别, 记此时为 t_d , 此时网联车由 u 减速为 v_e 使得车后交通流状态变为 e , 而车前将形成一段空隙 (状态 o), 从而形成三股交通波: 向后的集结波 s_{ae} 、向前的集结波 s_{oe} 和向前的消散波 s_{ao} 。因为 o 为空状态, 所以消散波 s_{ao} 的速度为 u , 集结波 s_{oe} 的速度为 v_e ;
 - (c) 消散波 s_{ao} 与集结波 s_{ah} 相遇后将形成新的消散波 s_{ho} , 使得出现通行能力折减的状态 h 逐渐消散, 记状态 h 完全消散时所对应的时间为 t_c 。于此同时, 向前的集结波 s_{oe} 也会使状态 o 逐渐消散;
 - (d) 假设集结波 s_{oe} 过早追上消散波 s_{ho} , 即空隙 o 过早消失时, 因为状态 e 的流量大于 h , 则依然会形成新的向后的集结波, 故当且仅当状态 h, o 同时消散时控制效果达到最优, 这意味着在 t_d 时刻网联

车需要正好处于集结波上游 x_1^* 处，由解析几何计算得

$$x_1^* = \frac{(s_{ah} + u)(v_e - v_h)}{v_h(u - v_e) - s_{ah}(v_e - v_h)} L_q = \alpha L_q$$

- (e) 显然在实际应用时，无法保证 t_d 时刻正好有一网联车位于集结波 s_{ah} 上游 x_1^* 处，因此需要对上述控制策略进行微调。记 t_d 时刻距 x_1^* 处最近的网联车的位置为 x_{CV} ，当 $x_{CV} > x_1^*$ 时，只需延迟合适的时间 Δt_w 再减速至 v_l 即可保证状态 h, o 同时消散。同样由几何关系

$$\Delta t_w = \frac{x_{CV} - x_1^*}{(1 + \alpha)s_{ah} + u}$$

当 $x_{CV} < x_1^*$ 时，则需要网联车进行更大幅度的减速至 $v_l (< v_e)$ ，维持时间 Δt_l 以形成足够的空隙 o 再后恢复至 v_e ，此时存在 Δt_l 和 v_l 两变量，且根据网联车的实际位置 v_l 存在一上限

$$\Delta t_l = (v_e - v_l)^{-1} \left[x_{CV} \left(\frac{v_e(v_h + s_{ah})}{v_h(s_{ah} + u)} - 1 \right) + L_q \left(\frac{v_e}{v_h} - 1 \right) \right] \quad 0 \leq v_l < \frac{v_h(L_q + x_{CV})(s_{ah} + u)}{x_{CV}(v_h + s_{ah}) + L_q(s_{ah} + u)}$$

3. 由网联车实现车队识别（图 4.2(b) 所示情况）。当网联车开始排队时，即意味着车队被识别，进而又有以下两种策略可供选择：

- 在某网联车开始排队的时将速度减至 $v_l (< v_e)$ ，形成足够大的空隙后再恢复速度至 v_e 。这一策略与上文 $x_{CV} < x_1^*$ 的情况完全一致，控制速度 v_l 与持续时间 Δt_l 的计算公式同上，只需代入 $x_{CV} = 0$ 即可。这一策略在识别车队的同时立即进行控制，理论上可减少更多的延误，但会形成一段更拥堵的状态 l ，不利于自 l 向 e 的状态转化，且增大事故风险；
- 网联车识别车队后发出信息，指挥上游的另一辆位置合适的网联车进行控制，这一情况同样与上文完全相同。因为非即刻控制，方法对延误的减少效果较低，但期间状态的变化更为平缓，事故风险更低。两种方法需进行权衡。

4. 策略一仅需对一辆仿真车进行控制，最为简单，其核心为将瓶颈上游交通流状态自会引发通行能力折减的自由流状态 a 控制为不会引发通行能力折减的拥堵状态 e 。然而因为拥堵交通流状态的非稳定性，这一策略并不完全理想。后续的策略将尝试控制交通流状态为不会引发通行能力折减的自由流状态 g ，提高控制效果的稳定性，并假设车队的识别完全由网联车实现。

4.2.3 Strategy 2: control using single connected vehicle and VMSs

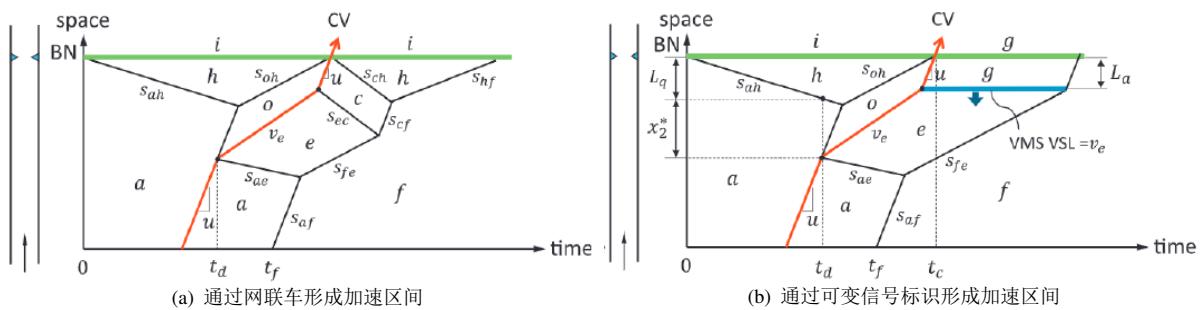


图 4.3 在瓶颈上游设置加速区间的交通流演化图

1. 方法首先通过网联车限速，将瓶颈上游交通流状态限制为不会引发通行能力折减的拥堵状态 e ，随后将网联车速度重新提升为自由流速度 u ，使得跟驰车流经过加速区间恢复自由流状态，但在没有其它辅助控制措施情况下，后续车流在自由流状态下将达到最大流量 q_c ，从而依然会造成通行能力折减（图 4.3(a)）；
2. 为了避免上述情况，在恢复自由流速度的同时借助可变信号标识（variable message signs, VMSs）将拥堵状态 e 转变为流量相同的自由流状态 g ，并以该状态通过瓶颈（图 4.3(b)）。方法不会降低消散率，还可带来更高的稳定性；

3. 策略中新引入了加速区间长度 L_a , 记策略二中的最优控制位置为 x_2^* , 则 x_2^* 除了与排队长度 L_q 有关外还需考虑 L_a 的要求;

$$x_2^* = \frac{(s_{ah} + u)(v_e - v_h)}{v_h(u - v_e) - s_{ah}(v_e - v_h)} L_q + \frac{(s_{ah} + u)(u - v_e)}{v_h(u - v_e) - s_{ah}(v_e - v_h)} \frac{v_h}{u} L_a = \alpha L_q + \beta L_a$$

4. 将网联车与可变信号标识结合的方法不仅优于单纯依靠网联车的方法, 较单纯依靠信号标识的方法也具有以下优点:

- 严格执行降速命令的网联车可在车前形成绝对空隙, 而通过信号标识指导普通车时受到跟驰模型的影响无法形成绝对空隙, 而会是一流量过渡区间。因此前者更有利于已有拥堵的消散, 限制条件也越少;
- 需要的可变信号标识越少 (2个), 有助于节省资金;
- 网联车的应用有助于该策略与其它同样基于网联车的技术的结合, 进一步降低成本;

4.2.4 Strategy 3: control using multiple CVs

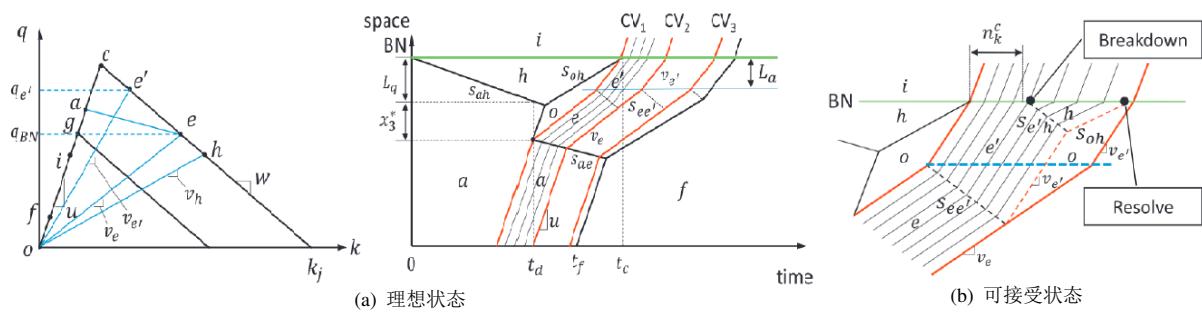


图 4.4 多网联车控制

1. 在网联车渗透率较高、时间空间分布较均衡时, 完全可摆脱对路侧设施的依赖单纯依靠网联车实现控制, 策略三 (图 4.4) 即是基于这一思想, 另外策略也保留了加速区间的设定;
2. 同样地, 在探测队列存在后, 首先控制网联车减速至 v_e , 将车后交通流状态调整为不会发生通行能力衰减的拥挤状态 e , 并在车前形成空隙。随后命令每一网联车先后于加速区间处加速至 $v_{e'}$ 并通过瓶颈。因为不同位置的网联车需在同一位置加速, 因此不同网联车加速的时刻不同, 从而形成一段段由网联车引导的车队, 车队之间由空隙分隔, 每一车队的交通流状态为 e' 。尽管 $q_{e'} > q_{BN}$, 但因为瓶颈处 $q_{e'}$ 与 q_o 交替出现, 只要两者时间控制得当, 使得长期的平均流量为 $q_e = q_{BN}$, 同样可以保证不发生通行能力衰减, 这即是基于 “瓶颈路段可以短暂地支持高于平均通行能力的车流”的假设;
3. 另外, 若因为某一车队偏长, 通行能力衰减不可避免时, 只要其后间隙足够长, 形成的拥挤状态 h 即可在下一段车队来临之前消散。因此在策略三下, 不仅局部超过最大通行能力是可接受的, 局部的通行能力衰减同样可接受;
4. 策略的核心在于 $v_{e'}$ 的设置, 其大小影响状态 h 的发生概率与程度。由直觉可得, 若网联车间距离越长, 形成的单列车队长度也越长, 车队间的空隙则越短, 则 $v_{e'}$ 应该越小, 而网联车间的距离又与网联车的渗透率和分布情况有关。定义第 k 段车队包含 n_k^c 辆车, 引入比例 $\frac{n_k^c}{n_k}$

$$\frac{n_k^c}{n_k} = \left(\frac{1}{q_{BN}} - \frac{1}{q_e} \right) \left(\frac{1}{q_{BN}} - \frac{1}{q_{e'}} \right)^{-1}$$

n_k^c 是使得状态 h 刚好能被消散的临界位置, 只要 h 发生在其之后则能不对下一车队造成影响;

5. 策略三与策略二在控制形式上高度相似, 不同之处仅在于将目标加速速度 u 换成 $v_{e'}$, 因此策略三的最优控制位置 x_3^* 的公式与策略二具有完全相同的结构, 仅将最后一部分的 u 换成 $v_{e'}$;

4.3 Adaptive control to remedy control failure

1. 对以上策略进行适应性调整，以消散排队并在控制策略失效时再次恢复至最大稳定消散率；
2. 若发生控制失效，形成新的拥挤状态 h 时，再次命令上游尚处于自由流状态 a 的网联车减速至 v_e ，形成新的空隙重新消散状态 h ，从而重新恢复消散率为最大稳定消散率；
3. 另外，为了更快地恢复至状态 e ，可控制更靠近瓶颈的网联车进行更大幅度地减速 (v_l)，形成足够的空隙后再恢复至 v_e 。方法可减少更多的时间浪费，但因为需大幅度减速，不利于交通状态的平稳过渡；
4. 当采用策略三时，如发生控制失效，另一种有效的适应性调整方法是减少加速区间长度，即命令网联车更晚地加速至 $v_{e'}$ 。这一方法可避免引入低速度 v_l ，但如果控制失效发生得过于频繁，加速区间的效果将消失，策略三退化为策略一。

4.4 Probabilistic control failure

本文假设瓶颈最大通行能力是随机的，控制策略失效的概率即是源于瓶颈最大通行能力的随机性，这一随机性与交通流状态及其持续时间有关。为尽可能避免控制失效，本文将给出使得控制失效概率最低的 v_e 和 $v_{e'}$ 。

4.4.1 Probability of control failure

1. 首先对策略一和策略二作出如下假设：
 - 处于状态 i 中的每一辆车，具有相同的引起失稳的概率 p_i ；
 - 驾驶员引起失稳的概率由其前车决定。

根据上述假设，每一辆车引起瓶颈交通流状态失稳的概率服从伯努利二项分布，则第一辆引起失稳的车所在的位置服从几何分布。从而给出每个策略 s 失效的概率质量函数 f_s (probability mass function, PMF)³、累计概率函数 F_s (cumulative probability function, CPF) 和期望；

2. 对策略一，瓶颈处交通流状态为 e ， f_1 即是处于状态 e 对第 n 辆车引起失稳的概率， f_1 的积分即是 F_1

$$f_1(n) = p_e(1 - p_e)^{n-1}, \quad F_1(n) = \sum_{j=1}^n f_1(j) = 1 - (1 - p_e)^n$$

对策略二，瓶颈处交通流状态为 g ，同样有

$$f_2(n) = p_g(1 - p_g)^{n-1}, \quad F_2(n) = \sum_{j=1}^n f_2(j) = 1 - (1 - p_g)^n$$

因为状态 g 处于自由流状态，理论上应该更稳定，有 $p_g < p_e$ ；

3. 对策略三，在已有的假设上新增如下假设：

- 因为每一网联车均领导一车队，网联车前为空隙，故网联车应具有更小的引发失稳的概率 $p_{e'}^{CV} < p_{e'}$ ；
- 只要后续车队未受到前车队的影响，则认为两车队情况相互独立；
- 网联车之间的距离服从一定的概率分布，故车队的长度不同。

4. 在策略三中，策略失效的情况等价于第 k 列车队出现拥挤状态 h 且影响到第 $k+1$ 列车队，而只有在状态 h 发生于车队的第 n_k^c 辆车之前时才会影响到下一车队，记这一概率为 P_k ，有

$$P_k = p_{e'}^{CV} + \sum_{n=2}^{n_k^c} (1 - p_{e'}^{CV}) p_{e'} (1 - p_{e'})^{n-2} = p_{e'}^{CV} + (1 - p_{e'}^{CV}) [1 - (1 - p_{e'})^{n_k^c-1}]$$

从而有

$$f_3(k) = \begin{cases} P_1 & k = 1 \\ P_k \prod_{j=2}^k (1 - P_{j-1}) & k > 1 \end{cases}, \quad F_3(k) = 1 - \prod_{j=1}^k (1 - P_j)$$

³ 概率质量函数的概念与概率分布函数类似，差别之处在于分布函数针对连续随机变量，质量函数针对离散随机变量。

4.4.2 Optimal design to maximize delay saving

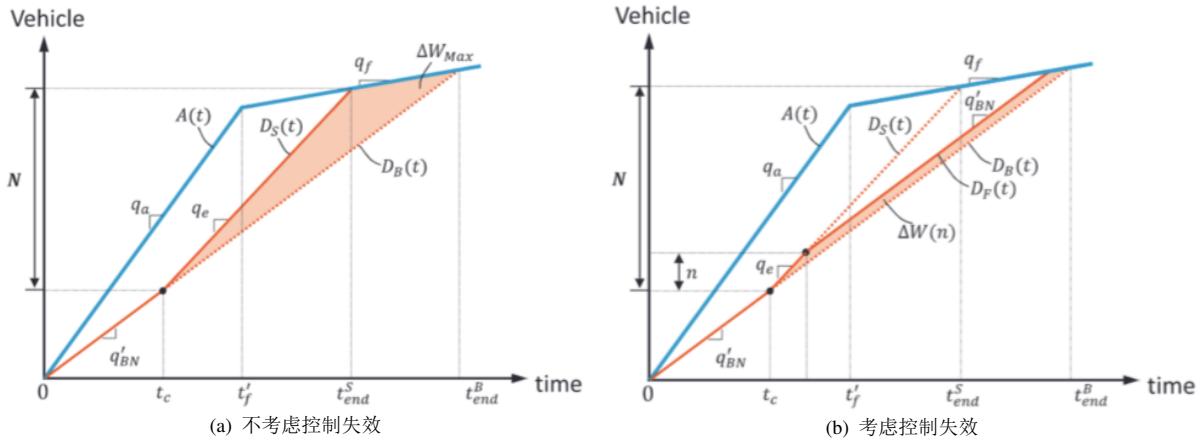


图 4.5 瓶颈断面离去-到达曲线

- 瓶颈断面的离去-到达曲线如图 4.5 所示。图中 N 表示累计通过瓶颈的车辆数， $A(t)$, $D(t)$ 分别表示到达和离去曲线，曲线的距离即为对应时刻延误的车辆数，其积分表示总的延误时间。 t'_f 表示上游交通需求变小所对应的时间。 $D_s(t)$, $D_F(t)$, $D_B(t)$ 分别表示无控制失效、有控制失效和无控制三种情况下的离去曲线；
- 当无控制失效时，策略对延误的减少效果达到最优，最大的延误减少量 ΔW_{Max}

$$\Delta W_{Max} = \frac{1}{2}(q_e - q'_{BN})(t_{end}^S - t_c)(t_{end}^B - t_c), \quad t_{end}^S = \frac{(q_a - q_f)t_{f'} + (q_e - q'_{BN})t_c}{q_e - q_f}, \quad t_{end}^B = \frac{(q_a - q_f)t_{f'}}{q'_{BN} - q_f}$$

当控制策略于第 n 辆车时失效，则起到的延误较少效果 $\Delta W(n)$

$$\Delta W(n) = \left[1 - \left(1 - \frac{n}{N} \right)^2 \right] \Delta W_{Max}, \quad N = q_e(t_{end}^S - t_c)$$

综上所述，可以得到策略 s 的延误较少效果的期望 $E_s(\Delta W)$

$$E_s(\Delta W) = [1 - F_s(N)]\Delta W_{Max} + \sum_{n=1}^N f_s(n)\Delta W(n) = \Delta W_{Max} \left[1 - \sum_{n=1}^N f_s(n) \left(1 - \frac{n}{N} \right)^2 \right] = \Delta W_{Max} \cdot P$$

- 上式描述了控制策略中控制速度 v_e 与延误较少期望 $E_s(\Delta W)$ 之间的关系： v_e 越大，对应有 q_e 越大，则 ΔW_{Max} 越大，同时失效概率 p_e 越大， $f_s(n)$ 也越大， P 越小。即 v_e 与 $E_s(\Delta W)$ 存在凹形曲线关系，很可能存在最优 v_e 使得 $E_s(\Delta W)$ 达到最大；
- 以上分析思路适用于策略一和策略二，策略三则更加复杂，因其需同时优化两个参数 v_e , $v_{e'}$ 。 v_e , $v_{e'}$ 和 $E_s(\Delta W)$ 的关系可能存在多个局部最优解，且和其它参数（如 $p_{e'}$, $p_{e'}^{CV}$ ）对假设息息相关。

4.4.3 Delay saving and optimal control speed for the adaptive control

以上分析进考虑策略是否失效，未考虑失效后可采取适应性控制措施恢复控制效果。借鉴上述四路，适应性控制措施可在 $\Delta W(n)$ 的基础上进一步提升延误减少效果 $\Delta W^R(n)$ 。 $\Delta W^R(n)$ 也存在最大值 $\Delta W_{Max}^R(n)$ ，即控制效果恢复后不再失效的情况下，而考虑到其依然可能再次失效，也可得到其期望值 $E_s[\Delta W^R(n)]$ 。可以定性分析得，在考虑了适应性调整措施后，策略即可得到更高的最优控制速度 v_e 。

4.5 Discuss on traffic instability & Conclusions and discussions

- 在以上分析过程中，不同交通流状态的失稳概率与最优控制速度密切相关。已有的研究指出：
 - 当自由流状态中的流量达到或超过其上限时，不稳定概率迅速增加；
 - 对于给定流量，密度越高不稳定概率也越高。

2. 由以上结论易知，策略二的效果理论上应优于策略一，至于策略三，受到 v_e , $v_{e'}$ 和网联车渗透率三参数的影响，其效果难以确定。一般认为，当网联车渗透率低时，策略三将退化为无可变标识的策略二，其效果很可能劣于前两项策略，而渗透率较高时，随着车队数的增加、单车队规模的减小，控制效果将得到明显提升；
3. 研究提出了三种基于网联车的可变限速策略以提升固定瓶颈的通行能力，并进一步给出了在控制失效时的调整方案。策略相较于传统方法具有以下优点：
- 通过在网联车前创造空隙消除拥堵，方法更简单、效果更好；
 - 因为方法不依赖或仅依赖少量交通设施，更经济；
 - 网联车可同时完成交通流状态感知和控制执行两项关键任务，更方便应用。
4. 为实际应用，本研究还存在以下问题：
- 本文为方便未考虑车辆的换道行为，在多车道环境中，换道行为可能会降低策略的控制效果，因此需要对多车道的网联车进行协同控制；
 - 本文采用一次加速或减速的方法控制网联车，在现实环境中速度骤变容易增加交通流失稳概率，宜采用逐步变速，也就需要对文章结论进行微调；
 - 无论何种策略都离不开首先识别拥堵，而拥堵识别本身和多种主观参数（如速度阈值等）和算法有关；
 - 本文所采用的交通流基本图和其它参数（如通行能力下滑的幅度、失稳概率）需要经真实实验数据校准。

4.6 英汉互译

English	Chinese	English	Chinese	English	Chinese
accommodate	容纳、适应 (v)	scheme	计划、方案	hinder	妨碍 (v)
onset	开端	presumably	很可能	mitigate	缓和 (v)
rubberneck	扭头观看 (v)	proactive	积极主动的	severe	极其严重的
gantry	龙门架	compulsory	强制的	intentional	故意的
vast	辽阔的	empirical	基于实验的	kinematic	运动学的
instigate	使发生	dissipate	消散 (v)	sophistication	高水平 (n)
prompt	快速的	trade-off	协调、权衡	albeit	尽管
tune	调整 (v)、曲子	thereafter	之后	facilitate	促进 (v)
enable	使可行	platoon	(军队) 排	offset	抵消 (n,v)
in the event of	万一发生...	incorporate	包含 (v)	premise	假定 (n)
parenthesis	插入语	concave	凹的	locus	核心
diagonal	对角线、对角线的	regime	政体、组织方法	exogenous	外部的

赌书消得泼茶香 当时只道是寻常

第二部分

交通网络

第5章

网络需求-状态解析

5.1 Empirics of a Generalized Macroscopic Fundamental Diagram for Urban Freeways (TRR, 2013)

ABSTRACT

交通控制规模的扩大带来了新的挑战：1) 处理大规模数据；2) 更好的评估交通状态；3) 求解巨大控制范围内的多个控制器的最优解。本文提出了一种新的描述大范围交通状态的方法，方法只需较少的数据即可准确的描述区域内的交通状态。具体地，本文将传统的宏观基本图（MFD）推广至城市高速场景，得到广义宏观基本图（generalized MFD, GMFD），GMFD 描述了交通生成量 (production)、总车辆数 (accumulation) 和密度空间分布 (spatial spread of density) 的关系。类似 MFD, GMFD 为连续函数，交通生成量随交通总量的增加先增后减，随密度空间分布的增加而减小。相比于 MFD, GMFD 可更好地预测交通生成量，因此基于其预测的交通状态可作为交通控制的基础。

5.1.1 Introduction

1. 随着交通控制控制尺度和控制变量的增加，多控制联合优化的难度不断增大，一方面是因为数据集的增大，另一方面是因为解空间的增大；
2. 近期学者重新关注 MFD 的概念。MFD 描述了一个区域内的平均流量和车辆数的关系，可以充分地表征交通状态。需要注意的是 MFD 要求研究区域内交通环境相似，而在不均衡的交通环境下 MFD 不成立；
3. 本文描述城市高速网络交通的不均匀性（由密度空间分布表示）对交通生成量的影响，并旨在建立一种简单的交通生成量定量描述方法——广义 MFD 模型（GMFD），生成量为总车辆数 (accumulation) 和密度空间分布 (spatial spread of density) 的函数。

5.1.2 Data processing

1. 本文所使用的数据来源于阿姆斯特丹 A10 高速公路 22km 长的一个路段，包括 4 个交叉口和 13 处进出口匝道。数据类型为断面检测器数据，包括统计间隔内的平均流量和时间平均速度；
2. 检测器的输出为每车道 l 的平均流量 q_l 和时间平均速度 v_l ，而计算密度需要空间平均速度 u_l ，因此需要首先计算空间平均速度；
3. 空间平均速度与时间平均速度在数值上的差异源于车辆速度的变异性。在荷兰，因为不允许右侧超车，所以内侧车道的车速一般会高于外侧，可以认为道路车辆速度的变异性主要体现为各车道速度的变异性，而认为车道内速度是均匀的，此时有 $u_l \approx v_l$ ，并可近似计算得整个道路的空间平均速度 u 和密度 $k = \frac{q}{u}$

$$u = \left(\frac{\sum_l q_l \frac{1}{v_l}}{\sum_l q_l} \right)^{-1}$$

基于断面探测器得到的交通状态可表示上下游两个探测断面中间的区域的交通状况；

图 5.1 网络交通生成量与总车辆数关系图: (a) 二维直方图, 在临界密度附近交通生成量的分布相对分散; (b) 网络 MFD 图, 散点的颜色表示密度空间分布, 显然与总车辆数正相关, 且在总车辆数相同的情况下密度空间分布与生成量负相关。

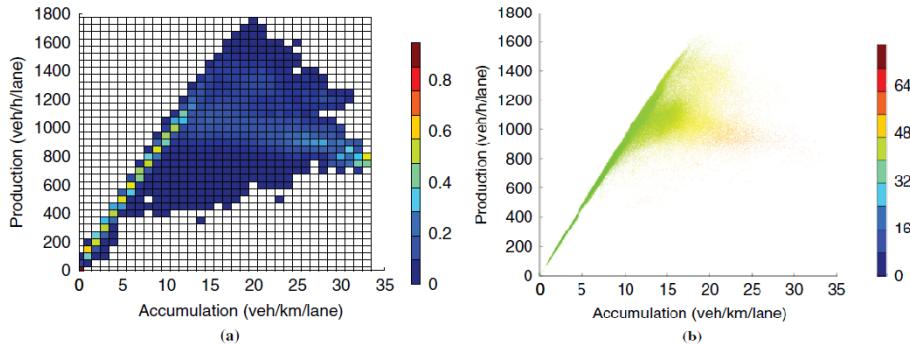
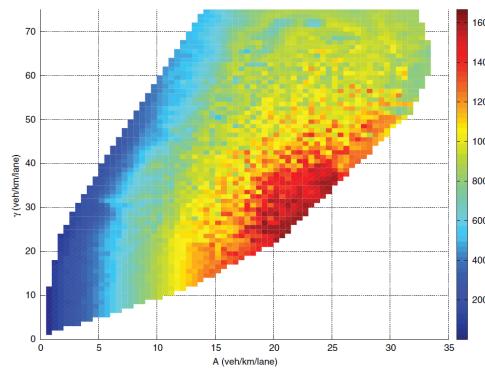


图 5.2 城市高速 GMFD 图, 颜色表示每小时每车道的交通生成量。交通生成量为因变量, 为总车辆数和密度空间分布的函数。



4. 记路段 s 内的流量、密度、车道长度分别为 q_s, k_s, L_s , 其中 L_s 为路段 s 的长度乘路段的车道数。此时可计算路网的交通生成量 P 、总车辆数 A 和密度空间分布 γ (表示为空间密度的标准差)

$$P = \frac{\sum_s L_s q_s}{\sum_s L_s} \quad A = \frac{\sum_s L_s k_s}{\sum_s L_s} \quad \gamma = \sqrt{\frac{\sum_s (L_s(k_s - \bar{k})^2)}{\sum_s L_s}}$$

5. 路网采集的交通生成量与总车辆数的关系如图5.1所示。其中图5.1(b) 即为网络 MFD 图 (在经典的 MFD 图中交通生成量仅表示为总车辆数的函数)。可以看到, 仅基于总车辆数预测交通生成量存在较大的不确定性, 而加入密度空间分布作为解释变量可更好地提升预测效果, 此时得到广义 MFD 图 (图5.2)。

5.1.3 Predicting production & Comparison of MFD and GMFD

1. MFD 模型基于路网的总车辆数预测路网交通生成量, GMFD 模型则基于总车辆数和密度空间分布共同预测, 预测的交通生成量是后续交通控制的基础。具体地有两种预测方法——非参数法 (插值法) 和函数拟合法:
 • MFD 模型仅考虑总车辆数 A 一个自变量, 仅针对其进行聚合。设置沿横坐标 A 均匀设置若干个格点, 对于每个格点, 选择距其最近的 1000 个样本点并基于二项式模型拟合, 拟合的二项式曲线对应于该格点的取值即为预测的该格点的交通生成量 P ;
2. 相比于函数拟合法, 插值法不需要预设函数形式, 属于无先验预测方法。对于待预测的样本点, 可基于已采集的数据通过插值直接预测新样本点对应的交通生成量。当训练样本规模过大时, 可提前聚合训练样本, 压缩需储存的数据集规模:
 - MFD 模型仅考虑总车辆数 A 一个自变量, 仅针对其进行聚合。设置沿横坐标 A 均匀设置若干个格点, 对于每个格点, 选择距其最近的 1000 个样本点并基于二项式模型拟合, 拟合的二项式曲线对应于该格点的取值即为预测的该格点的交通生成量 P ;
 - GMFD 模型考虑总车辆数 A 和密度空间分布 γ 两个自变量, 需在两个维度上进行聚合。同样地均匀设置二维网格点划分训练样本并取均值作为相应网格点的预测交通生成量 P 。
3. 相比于插值法, 函数拟合法只需要保存极少数拟合参数即可保存原始样本的全部信息, 且可直接得到临界车辆数、堵塞车辆数、自由流速度、最大交通生成量等关键参数。考虑 P 和 A 之间的关系, 经典理论常以二项曲线表示, 意味着临界车辆数为堵塞车辆数的一半, 显然不符合实际, 因此本文假设 P 和 A 之间成三次曲线关系。因为缺少关于 P, γ 关系的研究, 本文假设两种为线性关系。最终得到 MFD 模型和 GMFD 模型的拟合曲线

$$P_{MFD}(A) = m_1 A + m_2 A^2 + m_3 A^3 \quad P_{GMFD}(A, \gamma) = m_1 A + m_2 A^2 + m_3 A^3 + r\gamma$$

因为假设 P, A 之间为三次曲线关系，拟合的结果会使得自由流速度被高估（因为在自由流阶段 P, A 之间基本为线性关系）；

4. 得到拟合的参数后还可以考虑针对参数拟合结果和用于拟合的样本量作敏感性分析，判断选择多少样本量可实现对参数的充分估计——选择一系列样本比例取值，对于每一取值多次随机抽取相应数量的样本并估计参数，比较不同样本量下参数估计值的波动情况和函数对预测值的拟合情况；
5. 非参数插值法和函数拟合法各有优缺点，比较两种方法所建的 MFD 和 GMFD 模型可得到相似的结论： A 较高时 P 散布在一个二维区域内，此时基于 MFD 模型预测的 P 实际上是二维区域散点的均值，无法反映 P 在二维区域内的散布情况，而引入 γ 的 GMFD 模型的预测效果则得到显著提升。

5.1.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
lean	斜靠 (v)、精干的	idealized	理想化的	stretch	一段、拉长 (v)
hysteresis	磁滞	hysteresis loop	磁滞回线	speeding	超速行驶
harmonic average	调和平均	crisp	脆的、洁净的	lengthy	很长的
interpolation	插值法	a priori	先验的	polynomial	多项式，多项式的
quadratic	二次方的	prediction power	预测能力	parsimony	吝啬 (n)
gridlocked	(系统) 瘫痪的	curvature	曲率	in line with	与…一致

5.2 Urban Traffic Pattern Analysis and Applications Based on Spatio-Temporal Non-Negative Matrix Factorization (TITS, 2022)

ABSTRACT

- □ ×

为分析城市级路网的交通状态，提出了一种基于时空非负矩阵分解 (spatial-temporal non-negative matrix factorization, ST-NMF) 的路网交通模式分析方法，将网络状态表征为若干基本模式的线性组合。相关方法进一步应用于交通数据缺失重建及预测研究。因此所提方法不仅可用于获取高质量的交通数据基础，也可挖掘典型时空特征，并服务于交通状态预测。基于真实数据验证所提方法有效性。□

5.2.1 Introduction & Related work

1. 研究城市交通模式大致包括研究交通状态特征、拥堵区域与疏散区域分布、以及交通状态时间演化规律。相关特征具有较高的时空维度，随机性、复杂性较高；
2. 基于城市级交通数据以理解交通状态至关重要。但受制于交通数据规模大、范围广、噪声多以及不完全等问题，大规模城市路网的交通模式研究具有较大挑战，信号灯、交叉口、交通标志等因素也会进一步增大交通分析的难度；
3. 早期的研究主要基于固定检测器数据挖掘单一或少量路段的交通时变特征，方法主要包括模型驱动和数据驱动：
 - 模型驱动方法依托各类交通流仿真模型，受交叉口、信号灯、行人、事故等一系列因素的干扰，基于物理规律建模大规模城市路网状态并不便利；
 - 数据驱动方法依托机器学习算法，因不存在结构性约束在拟合交通状态时偏差较小，然而应用于大规模路网时模型将面临维度灾难。
4. 随着移动 GPS 数据的大量累积，考虑时空信息提取城市交通模式的新研究相继出现，方法包括宏观基本图模型、自相关模型、核密度估计、聚类、降维、复杂网络等等。然而，相关方法或未能充分利用路段间的时空相关性，或无法直接应用于大规模路网，且效果受限于数据质量；
5. 为克服维度挑战，部分研究尝试基于数据驱动方法对交通数据进行降维，同时挖掘交通状态的时空特征。但研究在降维时更多考虑交通状态的表征与时间演化，较少依据时空结构特征的组合优化降维过程；

6. 本文旨在基于时空交通数据从整体交通流中推断出低维的潜在交通模式，具体是将网络流分解为若干基本交通模式；
7. 为此，将城市网络所有连边状态视为整体，提出了时空非负矩阵分解 (**spatial-temporal non-negative matrix factorization, ST-NMF**) 算法以进行模式分析和挖掘，识别路网典型时空交通模式，探索交通状态整体表示方法；
8. 本文的主要贡献概括为：
 - **ST-NMF 算法**将交通状态分解为两个具有实际物理意义的非负矩阵，其中时间模式矩阵描述交通状态的时间演化，空间模式矩阵描述路网连边的内在属性；
 - **ST-NMF 算法**引入稀疏约束 (**sparse constraint**)、时间变异约束和空间相似度约束，前者用于克服异常数据的污染，后两者则用于保留交通数据的局部时空特征；
 - **ST-NMF 算法**可为数据重建和预测提供启发。

5.2.2 Methodology: spatial-temporal non-negative matrix factorization

1. 本节主要讨论如何提取网络交通的基本模式以挖掘网络交通状态的时空特征；
2. 将网络时空状态表示为 $n \times m$ 维矩阵 $S = [s_1, s_2, \dots, s_m]$, n, m 分别为网络连边和采样时段数目，其中列向量 s_i 即表征特定时刻的网络整体状态。假设任意时刻的网络交通状态 s_i 均可表示为 r 种基本模式 (b_1, b_2, \dots, b_r) 的线性组合

$$s_i = B p_i + a_i + n_i, \quad B = [b_1, b_2, \dots, b_r], \quad \forall i = 1, 2, \dots, m$$

式中矩阵 B 为 $n \times r$ 维矩阵； p_i 为 $r \times 1$ 维列向量，表征相应时刻各基本模式的线性组合系数；列向量 a_i, n_i 均表示误差，前者为事故等突发事件造成的异常值，后者表示弱噪声。将上式改写为矩阵形式，则有

$$S = BP + A + N, \quad P = [p_1, p_2, \dots, p_m]$$

式中 BP 为网络时空状态矩阵 S 的确定性部分，公式的核心即是将 S 分解为 B 和 P 两部分：

- 矩阵 B 的行向量为相应连边状态的低维表示，与矩阵 S 的行向量一一对应，故可理解为矩阵 B 反映了网络状态 S 的空间相关性；另一方面也可将矩阵 B 称为隐式空间模式矩阵 (**latent spatial pattern matrix**)，因其列向量对应网络交通基本空间模式；
- 矩阵 P 的列向量为各基本模式的线性组合系数，与矩阵 S 的列向量一一对应，故可理解为矩阵 P 反映了网络状态 S 的时间相关性；另一方面也可将矩阵 P 称为隐式时间模式矩阵 (**latent temporal pattern matrix**)，因其行向量表示基本模式线性组合系数的时间变化，对应网络交通基本时间模式。

3. 进一步提取时间模式矩阵 P 和空间模式矩阵 B 的特征。定义 r_t 量化网络交通的时间变异性，有

$$r_t = \|PT\|_F^2, \quad T = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 1 & -1 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & -1 \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{m \times (m-1)}$$

式中 T 为 $m \times (m-1)$ 维托普利茨 (Toeplitz) 矩阵， PT 表示对矩阵 P 按列作差分，从而得到网络空间状态基本模式线性组合系数的时间变化；而 $\|\cdot\|_F$ 表示矩阵的 F 范数，为各元素平方和的开方，因此 $\|PT\|_F^2$ 即可综合量化网络时空状态矩阵 S 的时间变异性。同时定义 r_s 量化网络交通的空间变异性，为考虑路网拓扑特征的影响，定义为图拉普拉斯二次型指标（具体物理意义及推导见第 18.5 节），有

$$r_s = \text{tr}(B^T LB) = \frac{1}{2} \sum_{j,l} \|\tilde{b}_j - \tilde{b}_l\|^2 W_{jl}$$

上式中 \tilde{b} 表示空间模式矩阵 B 的行向量， L, W 分别表示图结构的拉普拉斯矩阵和邻接矩阵（权重矩阵）。注意到实际路网中交通状态的空间相关性不仅受路网拓扑影响，也与管控策略等其它因素有关，故不直

接基于地面路网结构计算 r_s , 而是首先基于观测数据 S 估计各路段交通状态间的相关性并以图结构表示, 在提取图结构的 W, L 矩阵以计算 r_s 。具体的计算流程为:

- 以矩阵 S 的各行 (即道路路段) 为节点构建带权图, 计算各路段交通状态间的余弦相似度 (cosine similarity)

$$S(j, l) = \frac{C(j, l)}{\sqrt{C(j, j)C(l, l)}}, \quad C(j, l) = \tilde{b}_j \odot \tilde{b}_l$$

式中 $S(j, l) \in [0, 1]$ 即为路段 j, l 交通状态的余弦相似度, \odot 表示各元素相乘;

- 对于各节点, 与其相似度最大的前 K 个节点相连, 连边权重 $W_{jl} = S(j, l)$, 另有 $W_{jj} = 1$;
- 最后保证矩阵 W 为对称阵, 并计算度矩阵 D 和拉普拉斯矩阵 L

$$W = \frac{1}{2}(W + W^T), \quad L = D - W$$

4. 为计算路网交通时空基本模式矩阵 P, B , 注意到根据物理定义恒有 $P, B \geq 0$, 并考虑到 $S \geq 0$, 天然满足非负矩阵分解 (non-negative matrix factorization, NMF) 算法 (详见第 20.2 节) 的基本假设, 故基于其求解。考虑到表征异常事件的矩阵 A , 则应用 NMF 算法可建模为

$$B, P, A = \arg \min_{B, P, A} \|S - BP - A\|_F^2 + \lambda_1 \|A\|_1, \quad \text{s.t. } B, P \geq 0$$

式中 $\|A\|_1$ 为使得矩阵 A 尽可能稀疏的正则项, 其中 $\|\cdot\|_1$ 为 1 范数, 定义为矩阵列向量各元素绝对值之和的最大值 $\|A\|_1 = \max\{\sum_i |a_{ij}|\}$;

5. 进一步注意到, 在分解矩阵 S 得到 B, P 时, 为避免引入噪声以保证 B, P 的物理解释性, 应要求 B, P 在充分还原矩阵 S 的同时使用尽量少的时空信息。故将交通状态时空变异性指标引入上述 NMF 模型, 得到考虑时空特征的 ST-NMF 模型

$$B, P, A = \arg \min_{B, P, A} \|S - BP - A\|_F^2 + \lambda_1 \|A\|_1 + \lambda_2 \text{tr}(B^T LB) + \lambda_3 \|PT\|_F^2, \quad \text{s.t. } B, P \geq 0$$

按第 20.2 节所介绍的 NMF 求解思路, 基于拉格朗日乘数法求解 B, P, A , 首先写出拉格朗日函数

$$\mathcal{L} = \text{tr}((S - A)(S - A)^T) - 2\text{tr}((S - A)P^T B^T) + \text{tr}(BPP^T B^T) + \lambda_1 \|A\|_1 + \lambda_2 \text{tr}(B^T LB) + \lambda_3 \text{tr}(PTT^T P^T) - \text{tr}(\Phi B^T) - \text{tr}(\Psi P^T)$$

式中矩阵 Φ, Ψ 为拉格朗日乘子矩阵。为降低计算复杂度, 并不同时求解矩阵 B, P, A , 而是基于交替方向乘子法 (alternating direction method of multipliers, ADMM) 解耦求解 (详见第 17.6.2 节), 每一轮迭代时:

- 首先固定 A , 按经典 NMF 算法基于偏导信息构造 B, P 的乘法更新公式 (详见第 20.2 节)

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial B} = -2(S - A)P^T + 2BPP^T + 2\lambda_2 LP - \Phi \\ \frac{\partial \mathcal{L}}{\partial P} = -2B^T(S - A) + 2B^T BP + 2\lambda_3 PTT^T - \Psi \end{cases} \Rightarrow \begin{cases} B_{ij} \leftarrow \frac{[(S - A)P^T + \lambda_2 WB]_{ij}}{[BPP^T + \lambda_2 DB]_{ij}} B_{ij} \\ P_{ij} \leftarrow \frac{[B^T(S - A)]_{ij}}{[B^T BP + \lambda_3 PTT^T]_{ij}} P_{ij} \end{cases}$$

- 固定 B, P 优化矩阵 A , 即求解以下优化问题

$$A = \arg \min_A \|S - BP - A\|_F^2 + \lambda_1 \|A\|_1 = \text{sign}(S - BP) \max \left\{ |S - BP| - \frac{\lambda_1}{2}, 0 \right\}$$

上式中 $\|A\|_1$ 不可微, 故无法直接基于传统梯度下降算法求解, 而应采用如近端梯度下降等非平滑凸优化算法 (详见第 17.4 节); 求解后得到的数学形式被称为软阈值算子 (soft-threshold operator), 其中 $\text{sign}(\cdot)$ 表示符号函数, 当自变量大于 0 时输出 1, 小于 0 时输出 -1, 等于 0 时输出 0, 而 $|S - BP|$ 表示对矩阵 $S - BP$ 元素取绝对值;

- 按下式判断 ST-NMF 算法是否收敛

$$\max \left\{ \frac{\|B^{(t+1)} - B^{(t)}\|_\infty}{\|B^{(t)}\|_\infty}, \frac{\|P^{(t+1)} - P^{(t)}\|_\infty}{\|P^{(t)}\|_\infty}, \frac{\|A^{(t+1)} - A^{(t)}\|_\infty}{\|A^{(t)}\|_\infty} \right\} \leq \varepsilon$$

式中 $\|\cdot\|_\infty$ 为矩阵的无穷范数, 定义为矩阵行向量各元素绝对值之和的最大值 $\|A\|_\infty = \max\{\sum_j |a_{ij}|\}$ 。

5.2.3 Traffic pattern analysis and extended applications based on ST-NMF

1. 本小节进一步介绍如何分析 ST-NMF 算法分解得到的网络交通时空基本模式和 ST-NMF 算法的扩展；
2. 分析矩阵 B, P 可以得到路网交通的确定性特征，通过分析 B, P 的行、列各向量并提取相关特征即可得到网络交通状态的时空演化特征，也可对降维后的 B, P 矩阵聚类以进一步归纳网络交通的时空演化规律。相关时空演化信息也可用于预测路网状态。根据 $S \approx BP$ ，则 $t+1$ 时刻的网络状态 \hat{s}_{t+1} 有

$$\hat{s}_{t+1} = B\hat{p}_{t+1}$$

因此只需基于任意时间序列模型预测 $t+1$ 时刻网络各空间基本交通模式的线性组合系数 \hat{p}_{t+1} ，即可预测相应时刻的路网状态；

3. 分析矩阵 A 则可得到网络交通的随机性特征，特别是异常交通事件，从而有助于异常事件预测；
4. 进一步考虑实际环境中普遍存在的数据缺失情况。定义缺失位置映射算子 $\mathcal{P}(\cdot)$ ，当元素 X_{ij} 数据缺失时有 $\mathcal{P}_{ij}(X) = 0$ ，此时考虑数据缺失的 ST-NMF 问题建模为

$$\mathcal{P}(S) = \mathcal{P}(BP + A + N)$$

则相应的优化问题改写为

$$B, P, A = \arg \min_{B, P, A} \|\mathcal{P}(S - BP - A)\|_F^2 + \lambda_1 \|A\|_1 + \lambda_2 \text{tr}(B^T LB) + \lambda_3 \|PT\|_F^2 + \alpha (\|B\|_F^2 + \|P\|_F^2), \quad \text{s.t. } B, P \geq 0$$

式中目标函数额外引入了 $\|B\|_F^2 + \|P\|_F^2$ 作为正则项。

5.2.4 Experiment results and analysis

1. 采集的数据为北京市出租车浮动车数据，覆盖了 9044 条路段，以 10 分钟为间隔将一天划分为 144 个时段，则路网时空状态矩阵维度为 9044×144 ，每一元素为相应路段于相应时段的平均速度与限速之比；
2. 基于 ST-NMF 提取网络基本交通模式，关键超参是降维数目 r ，即预提取的基本模式数目。确定依据为：
 - 从统计角度， r 的选取应使得降维后 BP 尽可能保留原矩阵 S 的信息，可基于矩阵 S 的奇异值确定。奇异值由奇异值分解提取，所有奇异值之和可表征矩阵 S 的总信息量，故可以占比较大的奇异值数目确定 r 的大致取值范围；
 - 从物理角度，ST-NMF 算法要求分解后的矩阵 B, P 具有相对清晰的物理意义，基于此原则可以确定 r 的最终取值。

本例中前 10 个奇异值之和即可覆盖矩阵 S 超过 50% 的信息量，故确定 r 的取值范围为 3 至 10，发现 $r = 5$ 时分解后物理意义最为清晰；

3. 首先分析矩阵 P ，其行向量对应路网交通相应基本模式的时间演化规律。因此对矩阵按列作归一化后即可得到各基本模式于不同时刻的占比，从而归纳各基本模式占优的典型时段。进一步地以 P 的列向量为样本，对各样本进行聚类可将研究时段划分为若干基本时段，每一基本时段内的网络交通特征相似。令聚类数为 5，则得到的基本时段为早高峰时段 (7:20-9:50)、后半夜时段 (0:10-6:10)、前半夜与清晨时段 (6:20-7:10, 19:00-24:00)、晚高峰时段 (17:00-18:50) 及日间时段 (10:00-16:50)；
4. 而后分析矩阵 B ，其列向量对应不同的路网交通空间分布基本模式。结合矩阵 P 行向量表征的时间演化信息即可得到不同时刻对应的典型交通空间分布特征。进一步地以 B 的行向量为样本，对各样本进行聚类可将研究路网划分为若干路段集合，每一集合内的路段具有相似的空间特征，如交通水平、几何特征、可达性、土地利用等等；
5. 以上讨论的为日内路网交通时空基本模式的提取。若采集的数据覆盖多日（如一周），则基于相似的方法可挖掘路网交通的日间时空演化规律；
6. 另外分析矩阵 A ，其元素反应用对路段于对应时刻的异常误差的大小，负值表示复现的路网速度 BP 高于实际速度 S ，可能意味着路网运行状态受异常事件影响而恶化；

7. 进一步考虑所构建的 ST-NMF 模型在数据随机缺失时的网络状态重建效果。与其它 5 种算法对比：考虑缺失数据的一般 NMF 算法、稀疏正则奇异值分解、稀疏正则矩阵分解、低秩矩阵拟合以及自适应时空低秩算法。基于标准化平均绝对误差 $NMAE$ 量化缺失数据重建准确率，有

$$NMAE = \frac{\sum_{ij|\mathcal{P}_{ij}=0} |M_{ij} - \hat{M}_{ij}|}{\sum_{ij|\mathcal{P}_{ij}=0} |M_{ij}|}$$

式中矩阵 M, \hat{M} 分别表示真实的（无数据缺失）和模型还原后的网络交通状态。在三类数据集上验证，假设数据随机缺失，自 0% 至 100% 设置多项缺失率。对比发现所提方法的缺失数据修补效果显著优于其它对比算法，特别是在高数据缺失率的情况下，这一结果也进一步证明了 ST-NMF 模型可科学可行地描述网络交通状态；

8. 最后评价 ST-NMF 模型结果用于网络状态预测的效果。与其它 6 种算法对比：不考虑时空特征的一般 NMF 算法、人工神经网络、ARIMA、向量自回归、支持向量回归以及不区分异常矩阵 A 的 ST-NMF 模型。分别以 10 分钟、20 分钟、30 分钟为预测步长预测早高峰网络状态。对比发现所提方法结果用于网络状态预测时在整体上优于其它对比算法，特别是对比不区分异常信息的算法具有显著的预测精度提升，这一结果也表明了建模时区分异常信息的必要性。

5.2.5 英汉互译

English	Chinese	English	Chinese	English	Chinese
inhibitor	抑制剂	stimulate	刺激	plentiful	丰富的
extrapolate	推断	underuse	利用不足	prohibitive	禁止的
be termed as	被称为	canonical	典型的	conform	符合 (v)
manifold	流形 (n)	in relative terms	相对而言	anomaly	异常 (n)
backbone	骨干	salient	显著的	granularity	粒度
coil	线圈				

5.3 Predicting electric vehicle charging demand using a heterogeneous spatio-temporal graph convolutional network (TRC, 2023)

ABSTRACT

— □ ×

预测电动车的充电需求时空分布可服务于充电站运营及电力网络相应。本文的目标是提出一种面向多时空精度下电动车充电需求预测的异质时空图卷积网络 (*heterogeneous spatial-temporal graph convolutional network*) 模型。首先构建包含地理信息图 (*geographic graph*) 和需求图 (*demand graph*) 的异质图结构以捕捉不同充电区域的空间相关性，并基于图卷积和门控循环单元 (*gated recurrent unit, GRU*) 挖掘数据的时空特征。最后设计基于图嵌入和 POI 数据的面向特定区域的预测模型。基于北京市电动车 GPS 数据进行模型训练。实验结果表明所提模型在不同区域尺度下具有更高的预测精度和稳健性。消融实验 (*ablation study*) 和超参数敏感性测试指出考虑的地理信息图和需求图均可提升模型性能。

5.3.1 Introduction & Literature review

- 过去十年间在技术进步和政策扶持下电动汽车市场取得了长足的发展，但电动车充电的供需失衡问题依然存在：一方面是低密度的充电设施并不能满足日益增长的充电需求；另一方面一些充电站因位置不合理导致利用率低；另外电动车充电需求的周期性和不稳定性也会加剧能源网络的供电压力；
- 因此有必要精准预测短期内电动车充电需求的时空分布。这一信息一方面可引导电动车驾驶员错峰充电；另一方面也可指导能源企业合理配置充电设施以保证设备利用率；还可服务供电系统智能化设计从而保障经济效益和安全性；
- 与长期需求预测相比，因缺少高质量的电动车运营数据短时充电需求预测的研究较少。部分研究基于滑动平均法 (*moving average, MA*)、自回归 (*autoregression, AR*)、ARIMA 等经典时间序列统计模型预测短时充电需求。此类普遍假设历史数据与预测数据存在线性关系。另一部分研究则考虑机器学习或

深度学习方法建模的复杂的时序非线性特征并取得了良好的效果。然而，现有充电需求预测研究普遍聚焦时序特征的挖掘，而忽略了不同区域间充电需求的空间相关性；

4. 常用的挖掘数据空间相关性的方法是将数据整理为栅格表的形式后基于 CNN 模型及其变体提取空间特征，但针对欧式空间数据的 CNN 模型无法捕捉各区域电动车充电需求的异质空间相关性。图数据结构和图卷积模型（详见第二十五章）则可应对这一需求。目前时空图模型已应用于叫车服务（ride-hailing）、公共交通、个体出行等交通相关需求预测任务并取得喜人的效果，但在电动车充电需求预测领域的相关应用仍较少；
5. 作者构建异质时空图卷积模型以预测不同区域的电动车短时充电需求。模型的创新点包括：
 - 构建包含地理信息图（geographic graph）和需求图（demand graph）的异质图结构以捕捉不同充电区域的空间相关性；
 - 将充电区域分组并设计针对特定区域充电需求的预测模块。
6. 基于北京市 2018 年 1 月的电动车轨迹数据进行模型训练和测试，数据覆盖 76000 名电动车驾驶员。

5.3.2 Preliminaries & Methodology

1. 研究考虑的电动车充电需求是以一小时集计的分区域需求，预测目标是未来若干小时；
2. 记 t 时刻各区的电动车充电需求为向量 $X_t = (x_{1t}, \dots, x_{Nt})^\top$ ，其中 N 表示区域划分的数目；记需求预测时输入的历史时间片数目为 m ，预测的时间步长为 p ；又令无向图 G 表示电动车充电网络。则研究的电动车充电时空需求预测任务的目标即是学习映射 $f(\cdot)$ 满足

$$[X_{t-m+1}, \dots, X_t, G] \xrightarrow{f(\cdot)} [X_{t+1}, \dots, X_{t+p}]$$

上式中序列 $\{X_{t-m+1}, \dots, X_t\}$ 提供了各区域充电需求演化的时间信息，而 G 则旨在捕捉各区域间充电需求的空间相关性；

3. 模型由两个模块组成——时空学习模块和特定区域的需求预测模块。前者基于时空图神经网络学习充电需求的时空相关性的向量表征，后者基于向量表征结果和 POI 数据对充电区域进行 k-means 聚类，针对每个类别由全连接层预测充电需求；
4. 首先介绍时空学习模块的输入。为充分捕捉充电需求的空间相关性构建了两种不同的图结构——地理关系图和需求图。地理关系图表征了各充电区域在地理空间上的距离关系，用于捕捉空间中相近区域间充电需求的交互作用。记图的邻接矩阵为 $A^g \in \mathbb{R}^{N \times N}$ ，则其元素有

$$A_{ij}^g = \begin{cases} 1 & d_{geo}(v_i, v_j) < \varepsilon_g \\ 0 & \text{else} \end{cases}$$

式中 $d_{geo}(v_i, v_j)$ 表示区域 i 和区域 j 之间的空间距离（定义为欧氏距离）； ε_g 为预设的判断两个区域在空间上是否相近的阈值。需求图建模了各区域充电需求时变模式的相关性。记图的邻接矩阵为 $A^d \in \mathbb{R}^{N \times N}$ ，则其元素有

$$A_{ij}^d = \begin{cases} 1 & d_{dem}(v_i, v_j) < \varepsilon_d \\ 0 & \text{else} \end{cases}$$

式中 ε_d 同样为预设的阈值； $d_{dem}(v_i, v_j)$ 表示区域 i 和区域 j 之间充电需求时间序列的相似度，研究基于动态时间规整（dynamic time warping, DTW）算法（见第 23.9.5 节）度量。与欧式距离相比，动态时间规整法会首先对两个待比较的序列作非线性规整后再计算相似度；

5. 地理关系图和需求图分别输入两个独立的时空学习模块提取向量表示信息。模块首先由两层的 GCN 模型（见第 25.2 节）提取图结构中的空间特征，再输入 GRU 模块（见第 24.2 节）提取时间特征，分别得到地理关系图和需求图的向量表示 $H^g, H^d \in \mathbb{R}^{N \times K}$ 。其中 K 表示预设的图嵌入空间的维数（默认为 128），也是 GRU 模块中隐状态的维数。融合地理关系图和需求图的向量表示得到最后的异质图嵌入结果 $H \in \mathbb{R}^{N \times K}$

$$H = \alpha H^g + (1 - \alpha) H^d, \quad H^g = ST([X_{t-m+1}, \dots, X_t], A^g), \quad H^d = ST([X_{t-m+1}, \dots, X_t], A^d)$$

式中 $ST(\cdot)$ 即表示由两层 GCN 和 GRU 组成的时空图学习单元；线性融合参数 α 为可学习参数，随反向传播自动更新。实验结果表明，随着训练进行 α 由初值 0.5 逐渐降低至 0.35 以下，表明充电需求相关信息较地理相关信息对充电时空需求预测的贡献度更大；

6. 进一步介绍充电需求预测模块。常规的方法是直接基于单一的预测模型对时空图学习模块学习的向量表征结果作预测。考虑到不同区域充电需求模式的潜在差异，研究对各区域作聚类后针对每一类别构建独立的全连接预测模型；
7. 聚类的主要依据是各充电区域间的空间相关性。考虑两种相关形式——地理上的相关性和区域间用地类型的相关性。前者由前述构建的地理关系图表征，后者由各区域的 POI 数据表征。为融合两种空间相关形式，基于 DeepWalk 算法（见第 25.7 节）将地理关系图中的节点（代表充电区域）嵌入到低维向量空间中，再与 POI 数据拼接即可。设 DeepWalk 模型中的窗口尺寸超参为 10、嵌入维数为 16。每一区域的 POI 数据向量长度为 12，是对区域内 12 种功能设施的集计结果。基于 k-means 算法对拼接后的数据作聚类，从而将所有的充电区域聚类为预设的 C 个类别（默认为 16）。最后通过 C 个单层全连接模型预测各类别区域的充电需求

$$[\hat{X}_{t+1}, \dots, \hat{X}_{t+p}] = \sigma(W_c H_c + b_c), \quad c = 1, \dots, C$$

上式中 $\sigma(\cdot)$ 表示 sigmoid 激励函数； H_c 表示时空图学习模块提取的图嵌入数据中对应于类别 c 充电区域的部分； W_c, b_c 为针对类别 c 充电区域充电需求预测模型的可学习参数；

8. 以上即为所构建的电动车充电需求预测模型。后续测试时将其与多种成熟的预测模型效果作对比，包括传统统计模型、经典机器学习模型（支持向量回归、随机森林）、传统深度学习模型（GRU 模型）和各类图深度学习模型。

5.3.3 Case study of Beijing & Evaluating model performance with a public dataset

1. 首先以北京市 2018 年 1 月包含 76774 条真实电动车轨迹数据的数据集作为模型的训练、测试依据。每条数据记录了电动车的时间、位置、充电状态和瞬时速度。场景共包含 1128 个公共电动车充电站，配备快充和慢充设施。发现每天的充电高峰为上午 11 时至下午 5 时，而一周内各天的充电量并无明显差别；
2. POI 数据共包含交通设施、教育机构、金融机构、风景区、传媒设施、卫生设施、体育馆、政府机构、企业、服务设施、商业机构和居民楼盘共 12 中功能设施类型，共计 117901 条 POI 数据；
3. 分别针对 $1\text{km} \times 1\text{km}$ 、 $2\text{km} \times 2\text{km}$ 、 $3\text{km} \times 3\text{km}$ 共三种尺度划分的充电区域训练充电时空需求预测模型。空间区域划分越细，则预测结果对电动车驾驶员和充电站运营商决策的参考价值则越大，而对于能源系统管理而言则只需粗粒度空间下的预测结果；
4. 模型以过去 12 小时的充电需求数据作为数据，分别预测未来 3 小时和 6 小时内的充电时空需求。训练集、测试集和验证集的划分比例为 3:1:1。批训练时设每一批数据的样本量为 32，总训练次数为 200；
5. 选择对称平均绝对百分比误差 (symmetric mean absolute percentage error, SMAPE)、根均方误差 (root mean square error, RMSE) 和平均绝对误差 (mean absolute error, MAE) 评价预测精度

$$SMAPE(y, \hat{y}) = \frac{1}{N} \sum_i^N \left| \frac{y_i - \hat{y}_i}{(y_i + \hat{y}_i)/2} \right| \times 100\%, \quad RMSE(y, \hat{y}) = \sqrt{\frac{1}{N} \sum_i^N (y_i - \hat{y}_i)^2}, \quad MAE(y, \hat{y}) = \frac{1}{N} \sum_i^N |y_i - \hat{y}_i|$$

6. 测试结果指出当预测未来 3 小时的充电需求时，所搭建的预测模型在所有场景的所有指标中均取得最优；当预测窗口扩大至 6 小时时，所提模型与最优模型的表现非常接近，而因为所提模型的参数量更低，训练效率远高于后者；
7. 进一步通过消融实验 (ablation studies) 评价所搭建模型各模块的贡献。具体考虑三种对照模型：不对充电区域进行聚类而直接预测的模型、不考虑充电需求关系图作为输入的模型和不考虑地理关系图作为输入的模型。实验结果指出，所提模型包含的上述三个环节均有助于提升预测性能，其中充电需求关系图的重要性最高，而对充电区域进行聚类后分类预测的重要性最低；

8. 并对模型超参（充电区域聚类数 C 和时空图学习的嵌入维数 K ）分别设计敏感性实验。显然上述超参数值均与模型复杂度正相关。发现随着超参的增加模型的预测性能呈先增后减趋势，表明只有在合适的复杂度下才能获得最佳的预测效果。另外实验也指出时空图学习嵌入维数 K 对模型性能的影响大于充电区域聚类数 C ；
9. 另外为评估模型的可扩展性再训练其对预测长期充电需求。先后设预测窗口为 12 小时和 24 小时。实验结果表明所提模型仅稍弱于最优模型，表明模型不仅适用于短期充电需求预测，在长期预测任务上也有较好的表现；
10. 最后又以美国博尔德城 (Boulder City) 开源的 15 个充电站于 2020 年一年的充电数据训练并测试所提模型。每日的充电高峰同样为上午 11 时至下午 5 时。因为仅包含 15 个充电站，故此轮实验中训练的模型预测前不再对充电区域作聚类。实验结果显示所提模型在所有对照模型中取得了较好的效果，在 3 小时预测任务中取得了最优的 RMSE 和 MAE 指标，而在 6 小时预测任务中取得了最优的 RMSE 指标。

5.3.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
ablation	消融、磨蚀	volatile	不稳定的	dispatch	调度 (v)
warp	弯曲 (v)	instantaneous	瞬时的	scenic	风景优美的

5.4 KST-GCN: A Knowledge-Driven Spatial-Temporal Graph Convolutional Network for Traffic Forecasting (TITS, 2022)

ABSTRACT

已有的关于时空交通预测的研究较少考虑天气、POI 等外部因素对交通流的影响，而知识图谱可自然地建模此类关系。因此本文旨在提出一种面向交通预测的知识表示驱动时空图卷积模型。首先构造用于交通预测的交通知识图谱并进行表示学习，而后构造知识融合单元 (knowledge fusion cell, KF-Cell) 融合知识表示与交通特征作为时空图卷积模型的输入。真实数据实验指出所提方法可提升多种基准时空图模型的多尺度预测精度。消融实验和扰动分析进一步验证了所提方法的有效性和鲁棒性。目

5.4.1 Introduction & Related works

1. 基于历史交通信息预测未来交通状态，其核心在于捕捉交通流间的时间和空间相关性：
 - 经典的预测模型多基于时间序列模型、卡尔曼滤波等统计模型捕捉交通流的时间相关性。而后支持向量回归等非线性机器学习模型也得到广泛应用。随着深度学习技术的发展，后续研究多采用 RNN、LSTM、GRU 等循环神经结构（见第 24.2 节）；
 - 为捕捉交通状态的空间关系，早期研究采用贝叶斯网络、CNN 等方法。随着图学习模型的发展（见第二十五章），其非欧特征学习能力使其被广泛应用于捕捉不同路段交通状态间的相关性；
 - 近年来大量研究将图模型与时序神经网络结合，得到能同时捕捉交通流时空关系的时空图神经网络（见第 25.5 节）用于交通预测。
2. 除时空相关性外，交通流还受天气、基础设施、突发事件、节假日和 POI 等外部因素的影响，但较少交通预测研究考虑此类因素。如何融合多源数据的语义相关性是提升交通流预测精度的关键；
3. 多源数据的相关性主要通过网络结构进行建模。捕捉此类相关性的主流方法是生成反映网络结构和关系特征的表示向量。近年来知识图谱 (knowledge graph) 技术被广泛应用于此类场景（见第 25.9 节）；
4. 因此，研究基于交通状态与多源外部特征构建异质语义网络——知识图谱。进而基于知识表示技术捕捉交通状态与外部特征间的语义关系并输入时空图卷积模型。研究的创新点主要包括：
 - 构建了面向交通预测的知识图谱以捕捉外部特征对交通状态的影响，并设计知识融合单元 (knowledge fusion cell, KF-Cell) 将知识输入时空图卷积模型；
 - 基于真实数据评估所提方法。在多尺度预测条件下方法均可提升多个基准时空图模型的预测精度；
 - 进一步设计了消融实验研究语义相关性和静、动态外部特征对交通预测的影响。

5.4.2 Methods

- 首先构建知识图谱描述交通状态与外部特征之间的相关性，基于知识表示模型生成表示此类相关性的知识向量，并将其与交通状态融合作为时空图卷积模型的输入，则考虑的交通预测问题可建模如下

$$y = f(A, X, CKG)$$

其中 A 为路网 0-1 邻接矩阵； X 为交通状态矩阵； CKG 表示知识图谱；

- 首先介绍构建的知识图谱，涉及的知识三元组大体包括两类：

- **路段邻接三元组**：表示路段间的拓扑关系，如 $(\text{road } i, \text{connect}, \text{road } j)$ 表示路段 i 与路段 j 相连；
- **路段属性三元组**：描述各路段的属性，从而捕捉外部特征对交通的影响。研究主要考虑天气和 POI 两类属性。天气属于动态属性，通过如 $(\text{road } i, \text{weather condition}, \text{time } t)$ 和 $(\text{time } t, \text{weather}, \text{rain})$ 的两组三元组描述每个路段于每个时段的天气状态。POI 属于静态属性，通过如 $(\text{road } i, \text{school}, \text{number } n)$ 的三元组表示每个路段周围的 POI 分布。

研究构建的知识图谱不包含具体的交通状态，而是聚焦于路段间和路段与外部特征间的相互关系。因此从时空图学习的角度而言，引入知识图谱的目的可解释为为每个节点生成更针对性的表示向量以弥补邻接图结构的信息缺失。该思路已应用于 GMAN、AGCRN 等时空图模型（第 25.5 节）并表现出极佳的效果。但此类时空图模型仅依托误差传播自适应生成节点向量，而引入知识图谱可提供更解释性的结果；

- 进一步介绍采用的知识表示算法。因为搭建的知识图谱中包含两类三元组，故采用区分关系三元组和属性三元组的 KR-EAR 算法（见第 25.10 节），生成知识嵌入矩阵 X_E ；
4. 基于单层全连接模型融合知识嵌入矩阵 X_E 与 t 时刻的交通状态矩阵 X_t 得到融合特征 X'_t

$$X'_t = [X_s, X_d], \quad X_s = \text{ReLU}(e_s x_t w_s + b_s), \quad X_d = \text{ReLU}(e_d x_t w_d + b_d)$$

式中 e_s, e_d 分别指代静态外部特征与动态外部特征的表示向量； w_s, w_d, b_s, b_d 为知识融合单元的可学习参数； $[\cdot]$ 为拼接算子。生成的融合特征 X'_t 将替代原始特征 X_t 作为后续任意时空图学习模型的输入。

5.4.3 Experiments

- 实验采用深圳罗湖区的数据集，包含出租车轨迹数据、路网拓扑数据、气象数据和路段 POI 数据，时间范围为 2015 年 1 月份。网络涉及 156 个路段，并分有 9 种 POI 类型，分别为餐饮、企业、商场、交通站点、教育、生活、医药、住宅和其它。气象数据以 15 分钟为单位集计，分为温度、天气、风速、湿度、大气压和能见度共 6 项，天气状态又被分为晴天、多云、雾天、小雨和大雨 5 项。预测的交通特征为速度；
- 选择 DCRNN 和 T-GCN 两种代表性的时空图学习模型作为基准模型进行测试，并考虑 SVR、ARIMA、GCN 和 GRU 等代表性的经典统计、机器学习和深度学习模型作为对照。知识表示学习的嵌入维度设为 20，时空模型的隐藏层维度分别为 128 (DCRNN 模型) 和 64 (T-GCN 模型)；
- 首先仅测试模型的 15 分钟预测 (单步预测) 精度，发现时空图模型均优于经典统计、机器学习和深度学习模型，而引入知识图谱后的时空图模型又优于基准模型。又考虑多步预测，将预测时长增加为 30、45、60 分钟，得到一致的结论；
- 为评价通过知识图谱建模外部特征的效果，将模型与 AST-GCN (attribute-augmented spatiotemporal graph convolutional network) 模型对比。后者无需知识表示，直接拼接外部特征与交通状态，基于数据驱动的方法进行交通预测。与 AST-GCN 相比，所提方法具有更高的预测精度，且随着预测时长的增加精度差距逐渐拉大；若以预测误差作为指标，则两种模型的表现相似；
- 设计消融实验 (ablation experiments) 区分知识图谱中的 POI 和天气两类外部特征对交通预测的影响。发现两类特征均有助于提升预测精度，而同时考虑可进一步提升预测效果；
- 最后设计噪声实验评估模型鲁棒性。考虑高斯噪声 (Gaussian noise) 和泊松噪声 (Poisson noise) 两种噪声类型扰动交通特征，发现所提模型预测结果无显著变化，表明模型可克服潜在数据噪声的影响。

5.4.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
barometric	气压的	contingency	偶发事件		

5.5 An Ensemble of Deep Clustering Models With Autoencoders to Mine Travel Patterns From Smart Card Data (TITS, 2024)

ABSTRACT

- □ ×

已有基于聚类算法分析刷卡出行模式的研究往往需要预设聚类数目并处理高维刷卡数据。为此，研究基于自编码器模型挖掘高维刷卡数据的低维表示，并通过集成多个深度聚类模型识别出行模式。其主要优势在于自动确定聚类数目，并缓解聚类算法过拟合。应用于一万多条伦敦地铁刷卡样本，算法识别出7种出行模式，每一种出行模式对应一周的一天，而基准算法只能识别出3种。对于每一种模式，算法进一步识别出3种子模式，对应早高峰、平峰和晚高峰时段，从而得到总共21种模式，精准解析典型一周内的出行规律。

5.5.1 Introduction & Related works

- 挖掘刷卡数据时空模式可服务于智能交通设施规划设计。时空模式指在特定时间段周期性发生于特定站点的出行行为。聚类算法因其可从大量无标签数据中挖掘潜在模式而适用于刷卡数据出行挖掘模式；
- 应用此类算法面临三类挑战——预设聚类数目、难以准确聚类高维数据、过拟合：
 - 已有聚类研究常采用如肘部法则 (elbow method) 或 Silhouette 系数、Calinski-Harabasz 指数和 Daview-Bouldin 指数等方法确定聚类数目。但此类方法具有一定的模糊性，常对应不同的结果；
 - 另外刷卡数据常包含大量特征，而经典聚类算法依托的欧氏距离等相似度度量方法在高维空间中将无法提取有效信息。深度神经网络可用于提取图像、文本等高维数据的低维表示；
 - 聚类模型训练时可能面临过拟合问题导致聚类过多或过少，降低结果解释性，但较少研究深度聚类的研究考虑这一问题。
- 研究通过集成多个深度连续聚类模型 (deep continuous clustering, DCC, 第 24.9.4 节) 挖掘刷卡出行模式。DCC 模型基于自编码器框架同时生成高维数据的低维表示和聚类结果，且无需确定聚类数目。每个模型的结果都作为候选以确定最终的出行模式。另外基于 Rademacher 复杂度 (Rademacher complexity) 以检测 DCC 模型的过拟合情况，用于选择表现最佳的聚类结果；
- 研究创新点在于基于深度聚类挖掘刷卡数据且无需预设聚类数目，并可检测聚类算法的过拟合程度。

5.5.2 Deep clustering algorithms & Travel pattern mining system & Overfitting in deep clustering

- 一般地可将深度聚类算法的损失函数表示为

$$L = \lambda L_n + (1 - \lambda)L_c, \quad \lambda \in [0, 1]$$

其中 L_n 表示低维映射损失； L_c 表示聚类损失。DCC 模型基于自编码器架构， L_n 具体表示为重建损失；

- 为挖掘刷卡出行模式数据，首先进行数据预处理。对于每次刷卡出行样本，关注以下7类特征：出发时间、到达时间、起点经纬度、终点经纬度和星期几 (day of week)。进一步补充总行程时间和总行程距离指标。计算站点距离时不采用地理距离，而是考虑地铁网络内的实际行程距离。另外考虑到星期几指标具有周期性，将其替代为正弦和余弦编码。最终构造得到样本集 $X \in \mathbb{R}^{N \times 10}$ ；
- 而后基于 DCC 模型聚类出行模式。因为 DCC 模型以刷卡数据 X 建模的 k 近邻图作为输入，为确定超参 k 取值，研究设计了一套复杂的超参选取和最终结果生成机制，包括模型构建、过拟合检测、模型选择和模式生成四个步骤：

- 首先建立 K 个 DCC 模型用于超参遍历，令第 k 个模型以 k 近邻图作为输入 ($k = \{1, \dots, 10\}$)。每个模型首先进行 $M = 10$ 轮训练；
- 训练完成后检测模型的过拟合水平。删除过拟合模型后剩余模型继续进行 $4M$ 轮训练，保存每个模型每 M 轮训练后得到的聚类结果；

- 对于每个模型得到的 4 个聚类结果，基于多数投票确定该模型的聚类数，再基于所有剩余模型的聚类数由多数投票确定最终的聚类数，并删除聚类数不一致的其它模型；
- 继续训练剩余模型，每 M 轮训练后计算戴维斯-博尔丁指数 (Davies-Bouldin index, DBI)。DBI 为评价聚类模型效果的常见指标，其统计意义为每个聚类与其最相似聚类间的平均相似度，定义为

$$DBI = \frac{1}{G} \sum_{i=1}^G \max_{i \neq j} \left\{ \frac{S_i + S_j}{d(c_i, c_j)} \right\} \geq 0, \quad S_i = \frac{1}{|C_i|} \sum_{x \in C_i} d(x, c_i)$$

式中 G 表示聚类数目； $d(\cdot, \cdot)$ 为距离函数； c_i 表示第 i 个簇的聚类中心； C_i 表示第 i 个簇的样本集合； S_i 表示第 i 个簇所有样本点到聚类中心的平均距离。显然 S_i 越大则表示第 i 个簇的半径越大，而 $d(c_i, c_j)$ 越小则表示两个簇之间的距离越小，则分式 $\frac{S_i + S_j}{d(c_i, c_j)}$ 越大则表示簇 i, j 之间重合的程度越大，因此 DBI 越小表明簇越紧凑且分离越好。研究设定若观察到模型连续 $3M$ 次训练的 DBI 指标增大，则结束训练并保留 DBI 最小对应的聚类结果。对比剩余模型的所有 DBI 指标，取最小 DBI 对应的聚类结果作为最终输出。

- 进一步介绍深度聚类模型过拟合的检测方法。认为过多或过少的类别均属于过拟合聚类，并基于 Rademacher 复杂度 (Rademacher complexity) 构建过拟合判别指标。对于向量集合 $A \subset \mathbb{R}^d$ ，其 Rademacher 复杂度 $Rad(A)$ 定义为

$$Rad(A) = \frac{1}{d} \mathbb{E}_\sigma \left(\sup_{a \in A} \sigma^\top a \right), \quad \sigma \in \{-1, 1\}^d$$

式中 σ 被称为 Rademacher 变量，其每个元素等概率地取 -1 或 1。将其与集合 A 中的向量 a 点积可量化 σ 与 a 的相似度，则 $\sup_{a \in A} \sigma^\top a$ 即表示给定随机向量 σ 的前提下集合 A 中与 σ 最相似的元素对应的相似度。考虑 σ 的所有可能取值并取期望 \mathbb{E}_σ ，则 $Rad(A)$ 越大表示集合 A 中包含的向量越多样。对于聚类结果 C_i ，显然 $Rad(C_i)$ 不宜过大，则构造如下过拟合检测指标

$$Sharon = N \cdot \left(\sum_{i=1}^G Rad(C_i) \right)$$

作者认为上式于 [1, 2] 之间时表示模型拟合良好，反之均为过拟合，可能是聚类数目过多或过少导致的。

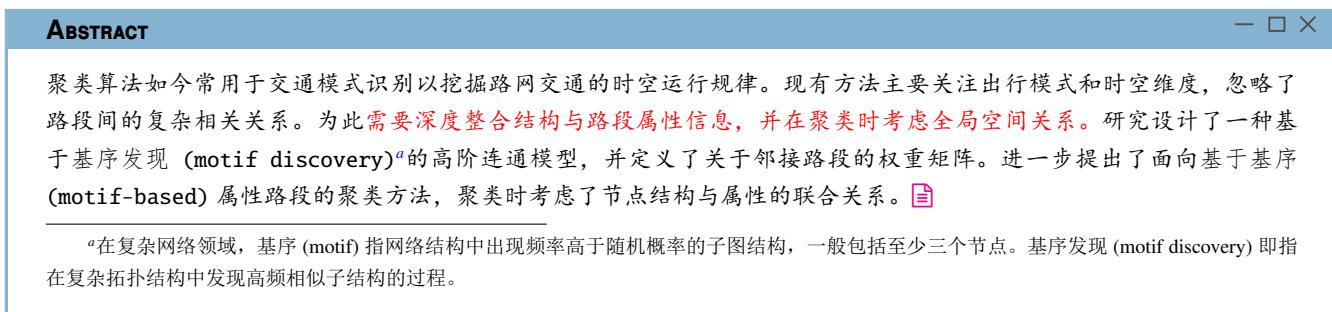
5.5.3 Results and discussion

- 研究选择 2009 年 11 月内一周的伦敦地铁卡交易数据，共包含 1,048,576 条样本，去除缺失值后得到 764,691 条数据；
- 选择 K-means、DBSCAN、Ward 聚类、谱聚类和仿射传播 (affinity propagation) 等五种经典聚类方法作为基线对照模型，并考虑有、无降维两种情况。前者基于 PCA 算法首先降维出行数据再进行聚类，降维特征数取 6，可捕捉 98% 的方差。所搭建的 DCC 模型的降维数目则设为 2；
- 选择 Silhouette 系数、Calinski-Harabasz 指数和 Davies-Bouldin 指数评价聚类效果。其中 Silhouette 系数取值介于 [-1, 1] 之间，越大表示聚类效果越好；Davies-Bouldin 指数是越接近 0 表示聚类效果越好；Calinski-Harabasz 指数则是越大表示聚类效果越好；
- 对于基线模型，发现引入 PCA 降维普遍可提升聚类效果，但聚类表现仍显著低于所提方法。另外大多数基线模型仅能识别 3 个类别，而对于 DCC 模型，发现当 $k = 3$ 时聚类效果最优，可识别出 7 个类别；
- 识别的每一种类别对应一种出行模式。为进一步分析所识别模式的具体特征，研究每个类别中频率最高的样本作为代表性样本，发现识别的 7 个类别对应一周的七天，且每个类别内部存在三种子模式，分别对应早高峰、晚高峰和平峰时段。

5.5.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
geodetic	大地测量的	rigorously	严密地	persist	坚持 (v)

5.6 Motif discovery based traffic pattern mining in attributed road networks (Knowledge-Based Systems, 2022)



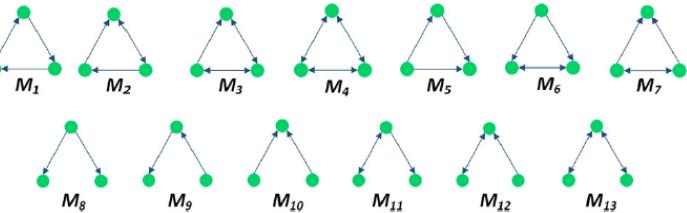
5.6.1 Introduction & Related works

1. 交通模式随路网拓扑、车辆轨迹、个体出行等交通属性而变化，已有研究基于聚类方法挖掘出行模式。部分研究关注以路段为单位的交通特征或以个体为单位的出行数据，对于路段之间的复杂关系则考虑较少；另一部分研究基于图聚类模型捕捉路段间的结构特征，但忽略了每个路段的属性特征；
2. 属性网络 (attributed networks) 是指存在节点属性特征的网络结构。交通网络即是典型的属性网络，交通模式既与交通属性有关，也应考虑潜在的物理拓扑结构。部分面向属性网络的图聚类模型可同时考虑邻接关系和节点特征，但因为结构特征和属性特征间的异质性，相关方法无法有效融合上述两类特征；
3. 矩阵分解 (matrix factorization) 可将原始数据矩阵近似为两个或多个矩阵的乘积。非负矩阵分解 (nonnegative matrix factorization, 见第 20.2 节) 是一种典型的矩阵分解算法，广泛应用于挖掘节点属性和拓扑关系间的共有模式，但已有研究普遍仅考虑图的邻接关系，未考虑路段间的高阶连接；
4. 具体地，已有网络场景下的交通聚类研究存在以下两类问题：
 - 异质的结构特征和属性特征导致聚类准确率低。路段特征包含结构特征和属性特征两个维度。已有研究普遍未考虑路段间的复杂非线性关系，且未能高效融合此类结构特征与路段自身的属性特征；
 - 仅考虑路段间的低阶连通性。路网中远距离的路段也存在相关性，然而已有研究在考虑路段结构特征时仅考虑其邻近的有限路段。
5. 基序 (motif) 指属性网络中频繁出现的至少包含三个节点的非同构 (non-isomorphic) 连通子图。基序考虑了结构间的高阶关系，可以基序为路网图结构的基本单位，从而捕捉全局结构信息以提升结构特征集计的效果；
6. 因此，研究提出了一种基于基序和非负矩阵分解的属性路网的交通模式聚类算法以考虑路段的高阶相关性并提升交通特征识别的准确性。非负矩阵分解算法用于分解节点特征相似度矩阵和拓扑邻接矩阵以聚类交通模式。与已有研究相比，所提方法具有以下两项优势：
 - 基于基序挖掘路段的高阶空间相关性；
 - 基于一致性图聚类 (consensus graph clustering) 算法缓解路段结构特征与属性特征的异质性并挖掘基本交通模式。
7. 研究的创新点主要为：
 - 所提方法联合路段结构和属性特征进行聚类，且该联合特征基于非负矩阵分解和 KKT 条件（见第 17.3.1 节）确定；
 - 提出了一种基于基序的路网高阶连通性模型以准确表示相邻路段的权重矩阵并捕捉路网全局信息；
 - 基于两类真实数据集进行大量实验证了所提方法的性能。

5.6.2 Methodology

- 交通数据可通过路网邻接矩阵 $W \in \mathbb{R}^{N \times N}$ 和节点特征矩阵 $F \in \mathbb{R}^{N \times d}$ 联合建模。其中 N, d 分别表示路段数和路段特征维数；
- 基于基序搜索识别路网的基序结构，在邻接矩阵 W 的基础上生成基于基序的加权邻接矩阵 $W_M \in \mathbb{R}^{N \times N}$ 捕捉路段的高阶关系。又基于特征矩阵 F 构造特征相似度矩阵 $S \in \mathbb{R}^{N \times N}$ ，其元素 $S_{ij} = \langle f_i, f_j \rangle$ 。基于非负矩阵分解将 W_M, S 分解为 BF_1, BF_2 ，其中 $B \in \mathbb{R}^{N \times m}$ 称为公共基矩阵 (**common basis matrix**)，其行向量可综合表征对应路段的结构和属性特征。再对 B 作 K-means 聚类即可提取交通模式；
- 进一步介绍基于基序的邻接矩阵 W_M ，其元素 $(W_M)_{ij}$ 定义为有向连边 (i, j) 所在的基序数目与网络中所有基序数目的比值。仅考虑由三个节点构成的基序，共包含 13 种基本类型。在有向路网的实际背景下，研究考虑 $M_1, M_5, M_8, M_9, M_{10}$ 共五种类型，以表示环形、绕行、分流、串联和合流等五种交通基本场景。需要说明的是，构建 W_M 时并不区分不同的基序类型；

图 5.3 包含三节点的图基序基本类型。其中 $M_1, M_5, M_8, M_9, M_{10}$ 可分别表示环形、绕行、分流、串联和合流等五种交通基本场景。



- 路网所包含的基序由遍历确定。对于包含 N 个节点的网络，按无序抽样生成 $\frac{N(N-1)(N-2)}{6}$ 组三元组，若三元组中的三个节点构成连通子图，则该三元组即为基序；
- 基于非负矩阵分解分解矩阵 S, W_M ，并注意到对任意矩阵 $\|A\|_F^2 = \text{tr}(A^\top A)$

$$\begin{aligned} (\min) \quad \mathcal{L}(B, F_1, F_2) &= \frac{1}{2} (\|W_M - BF_1\|_F^2 + \alpha \|S - BF_2\|_F^2), \quad B, F_1, F_2 \geq 0 \\ &= \frac{1}{2} [\text{tr}((W_M - BF_1)^\top (W_M - BF_1)) + \alpha \cdot \text{tr}((S - BF_2)^\top (S - BF_2))] \\ &= \frac{1}{2} [\text{tr}(W_M^\top W_M - 2F_1^\top B^\top W_M + F_1^\top B^\top BF_1) + \alpha \cdot \text{tr}(S^\top S - 2F_2^\top B^\top S + F_2^\top B^\top BF_2)] \end{aligned}$$

按第 20.2 节所介绍的非负矩阵分解求解思路，有

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{L}}{\partial B} = -W_M F_1^\top + BF_1 F_1^\top - \alpha S F_2^\top + \alpha BF_2 F_2^\top \\ \frac{\partial \mathcal{L}}{\partial F_1} = -B^\top W_M + B^\top BF_1 \\ \frac{\partial \mathcal{L}}{\partial F_2} = -\alpha B^\top S + \alpha B^\top BF_2 \end{array} \right. \implies \left\{ \begin{array}{l} B = B \odot \frac{W_M F_1^\top + \alpha S F_2^\top}{BF_1 F_1^\top + \alpha BF_2 F_2^\top} \\ F_1 = F_1 \odot \frac{B^\top W_M}{B^\top BF_1} \\ F_2 = F_2 \odot \frac{B^\top W_M}{B^\top BF_2} \end{array} \right.$$

5.6.3 Experiments

- 选择成都和深圳的滴滴数据集进行验证。其中成都数据集包含 2018 年九月至十月 74 个路段的速度数据，深圳数据集包含 2018 年四月至五月 79 个路段的速度数据。仅考虑高峰时段（上午 7-9 点，下午 5-7 点）以 10 分钟为单位集计，每路段每天包括 24 组属性。构建图模型，以交通指数 C_i^t 为路段属性

$$C_i^t = \frac{v_i^{\max} - v_i^t}{v_i^{\max} - v_i^{\min}}$$

其中 v_i^t 为路段 i 于 t 时刻的速度； v_i^{\max}, v_i^{\min} 分别表示路段 i 的观测最高速度与最低速度。 C_i^t 越大表示路段 i 于该时刻越拥堵。基于经验将路段状态按交通指数聚为 4 类（自由流、轻度拥堵、中度拥堵、重度拥堵）作为聚类真值；

- 选择多种基于图自编码器和图变分自编码器的模型与所提方法对比。采用聚类准确度 (clustering accuracy)、归一化互信息 (normalized mutual information, NMI) 和宏观 F1 评分 (macro F1-score) 指标评价聚类效果；

3. 令聚类数 $K = 2, \dots, 7$, 对比所提方法与对照方法的聚类效果并取平均, 发现所提方法的表现优于对照方法, 且当 $K = 4, 5$ 时优势最为明显。另外考虑不采用基序的退化版本, 此时算法分解的矩阵为邻接矩阵 W 而非 W_M , 发现聚类效果显著降低;
4. 进一步将数据划分为四个时段——周一至周五、其余工作日、周末和假期, 并基于 Silhouette 指标确定最优聚类数目 K , 发现周一至周五具有 5 类交通模式, 而其余时段则识别出 4 类交通模式。

5.6.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
motif	主题、基序	disjoint	使分离、分离的	repository	仓库
isomorphic	同构的	diversification	多元化	breadth	宽度
fine-grained	细粒度的	tier	层级		

第6章

网络资源配置

6.1 A model for planning locations of temporary distribution facilities for emergency response (Socio-Economic Planning Sciences, 2015)

ABSTRACT

- □ ×

提出了一种旨在动态选择临时设施并分配应急响应资源的网络流模型 (network flow model)。模型对不同临时设施的过剩资源进行调配以缓解应急资源短缺。定量分析指出临时设施的位置由需求点与供应点位置共同决定。图

6.1.1 Introduction

1. 自然灾害后的响应规划 (response planning) 应该将足够的物资及时地运输至恰当的地点。响应规划应该与对资源的需求同步，然而这一需求往往随时间的改变而（数量上的）消涨或（空间上的）转移；
2. 应急响应的评价标准大体包含响应时间、运输损耗和分配公平性等几大类指标。救助分配设施 (relief distribution facility) 的位置与分配对应急响应的效果至关重要；
3. 一般情况下，应急物资会被集中存放于预设的设施内，通过设施间的物资调配即可满足受灾地区的需求。然而这些设施往往数量较少，距受灾地区也较远，从而带来较大的运输成本，因此有必要在需求中心附近设置一系列临时设施 (temporary facility) 并将物资存放于其中，从而减少反应时间避免物资短缺；
4. 本文提出了一种动态应急响应规划模型。模型针对需求的实时变化，提供了一种实现快速配送的选址、配送规划，并对过剩资源进行跨区域调配。

6.1.2 Emergency logistics resource distribution model

1. 模型假设物资分配开始于中心供应点 (central supply point, CSP)，并假设其的总供应量超过对物资的总需求量（不代表时时刻刻都能满足），即所有的物资短缺问题理论上都可以通过跨时空调配满足；
2. 将时间离散化，针对每一时间段作出规划并分配物资。在每一时间段的开始，模型启用合适的临时分配中心 (temporal distribution center, TDC)，某个 TDC 在某个时间段内是否被启用取决于以下因素：TDC 容量、运行该 TDC 所需消耗、CSP 至 TDC 的行程时间、TDC 至需求点 (aggregated demand point, ADP) 的行程时间（具体参数表示见附录 6*I 节）。同时模型输出从 CSP 分配至每一启用 TDC 的物资数量，这些物质再由 TDC 运输至 ADP；
3. 在某一时间段内，部分 ADP 的需求可能无法被完全满足，这部分需求应该在下一时间段得到满足；而如果该时间段内的供给大于需求，过剩的资源将分配至 TDC 储存等待下一时间段的运输，若之后该 TDC 不存在，则将被转移至近邻的其它 TDC；
4. 模型将资源分为非消耗性资源 (non-consumable resources) 与消耗性资源 (consumable resources)。人们对前

者的需求往往是随机的，对这类需求可以有轻微的延迟；而对后者的需求往往是周期性的，应尽量避免延迟；

5. 模型的目标是最小化代价函数，是物流成本与物资缺乏所产生代价的总和：

- 物流成本包括启用 TDC 的固定成本和运输成本，并假设运输成本与运输距离、运输资源数量呈线性。同时单位运输成本 (TC_j^{k0} , TC_{mj}^{kI} , TC_{ji}^{kE}) 由运输方式和运输类型共同决定，并假设在整个规划期间内为常数；
- 物资缺乏所产生的代价由延迟送达的资源量 w_{jilt}^k 与延迟惩罚 P_{jilt}^k 共同决定，延迟越久延迟惩罚 P_{jilt}^k 越大。

6. 以下属于混合整数线性规划 (mixed integer linear programming, MILP)，具体地目标代价函数如下

$$\begin{aligned} \sum_{j \in \mathcal{U}} \sum_{t \in \mathcal{T}} F_j y_{jt} + \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{U}} \sum_{t \in \mathcal{T}} TC_j^{k0} r_{jt}^k + \sum_{k \in \mathcal{K}} \sum_{m \in \mathcal{U}} \sum_{j \in \mathcal{U}} \sum_{t \in \mathcal{T}} TC_{mj}^{kI} q_{mj}^k \\ + \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{U}} \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} TC_{ji}^{kE} \left[z_{jilt}^k + \sum_{l \in \mathcal{T}} w_{jilt}^k \right] + \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{U}} \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}} \sum_{l \in \mathcal{T}} P_{jilt}^k w_{jilt}^k \end{aligned}$$

同时，约束条件如下，其中：

- 约束条件 1、5、6 旨在避免出现 ADP 收到来自未启用 TDC 的资源的情况；
- 约束条件 2 旨在避免出现 TDC 收到超过 CSP 最大供给能力的资源的情况；
- 约束条件 3 旨在避免出现 TDC 所收物资超过其自身容量的情况；
- 约束条件 4 为网络流平衡方程，说明 t 时刻 TDC j 所具有的资源应等于该时刻该 TDC 向 ADP 分发的资源与下一时刻向其它 TDC 转移的资源的和；
- 约束条件 7 旨在避免离散化的规划周期长度过小，以至于无法在一个规划周期内完成分配；
- 约束条件 8 为供需平衡方程，即在规划完成最终时，每一个 ADP 对资源的需求应该恰好得到满足。

$$\sum_{i \in \mathcal{N}} x_{ijt} - B y_{jt} \leq 0 \quad \forall j \in \mathcal{U}, t \in \mathcal{T} \quad (6.1)$$

$$\sum_{j \in \mathcal{U}} r_{jt}^k \leq C_t^k \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \quad (6.2)$$

$$\sum_{k \in \mathcal{K}} S^k a_{jt}^k \leq V_j y_{jt} \quad \forall j \in \mathcal{U}, t \in \mathcal{T} \quad (6.3)$$

$$b_{jt}^k + \sum_{m \in \mathcal{U}} q_{jm(t+1)}^k = a_{jt}^k \quad \forall j \in \mathcal{U}, k \in \mathcal{K}, t \in \mathcal{T} \quad (6.4)$$

$$z_{jilt}^k \leq B x_{ijt} \quad \forall i \in \mathcal{N}, j \in \mathcal{U}, k \in \mathcal{K}, t \in \mathcal{T} \quad (6.5)$$

$$w_{jilt}^k \leq B x_{ijt} \quad l < t \text{ and } \forall i \in \mathcal{N}, j \in \mathcal{U}, k \in \mathcal{K}, t, l \in \mathcal{T} \quad (6.6)$$

$$\tau_{ji} x_{ijt} \leq L_t \quad \forall i \in \mathcal{N}, j \in \mathcal{U}, t \in \mathcal{T} \quad (6.7)$$

$$\sum_{j \in \mathcal{U}} \left(z_{jilt}^k + \sum_{t \in \mathcal{T}, t > l} w_{jilt}^k \right) = D_{il}^k \quad \forall i \in \mathcal{N}, k \in \mathcal{K}, l \in \mathcal{T} \quad (6.8)$$

$$q_{mj}^k = 0 \quad t = 1, N + 1 \text{ and } \forall m, j \in \mathcal{U}, k \in \mathcal{K} \quad (6.9)$$

$$y_{jt} \in \{0, 1\} \quad \forall j \in \mathcal{U}, t \in \mathcal{T} \quad (6.10)$$

$$x_{ijt} \in \{0, 1\} \quad \forall i \in \mathcal{N}, j \in \mathcal{U}, t \in \mathcal{T} \quad (6.11)$$

$$r_{jt}^k, q_{mj}^k, z_{jilt}^k \geq 0 \quad \forall i \in \mathcal{N}, j, m \in \mathcal{U}, k \in \mathcal{K}, t \in \mathcal{T} \quad (6.12)$$

$$w_{jilt}^k \geq 0 \quad l < t \text{ and } \forall i \in \mathcal{N}, j \in \mathcal{U}, k \in \mathcal{K}, t, l \in \mathcal{T} \quad (6.13)$$

6.1.3 Numerical analysis

1. 设置实验网络与基本参数：

- 以美国南卡莱罗纳州的 15 个城市组成的运输网络作为实验网络，包含 1 个 CSP、4 个 TDC 和 10 个 ADP。总规划时长为 2 天，均分为 4 个子时段；
 - 设置资源类别数及每一类单位资源所占据的空间，同时设置 CSP 和每一个 ADP 于每一时间段内对每一类资源的供应能力和需求量，另外设置启用每一 TDC 所需的固定费用和相应容量；
 - 因为子时段长度仅为 12 小时，且旨在运输紧急物资，设定运输方式为空运；
 - 针对每一类资源设置延误惩罚因子，延误时间越长，惩罚因子越大（本研究设置其与延误时间呈线性关系，实际上更接近指数关系）。
- 相比已有研究，实验模型允许物资于不同 TDC 之间调配。对比得知，在允许物资转运的情况下，尽管物流成本可能会略微上升，但充分避免了运输延迟的情况，实现总成本的显著降低；
 - 注意到本研究中规定物资从 CSP 配送至 ADP 必须经过 TDC，因此进一步分析当允许 CSP 直接向 ADP 配送物资时的效果。对比得知，在允许 CSP 直接向 ADP 运输时，尽管物流成本可能会略微上升，但进一步避免了运输延迟，从而进一步降低总成本。需要说明的是，在实际场合中，往往只有在运输能力非常宽裕时才考虑从 CSP 到 ADP 的直接运输；
 - 另外，注意到模型的很多关键参数取值单纯由假设给出，因此需要分析优化结果对部分模型参数的敏感性，包括延迟惩罚因子与规划周期长度两项主要参数。

6.1.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
transfer	转移、调动 (n,v)	deprivation	缺乏、剥夺 (n)	aftermath	后果、创伤
necessitate	是成为必要 (v)	depot	仓库、车库	centralize	集中 (v)
backorder	延期交货	discretized	离散化的	concentration	集中、专注 (n)
non-consumable	非消耗性的	commodity	商品	conservation	保护 (n)
cater	承办酒席 (v)	cater to	迎合……需求	in line with	与……一致
deficit	赤字、亏损 (n)				

6.2 A Stochastic Emergency Response Location Model Considering Secondary Incidents on Freeways (TITS, 2016)

考虑高速公路二次事故的随机应急响应选址模型

ABSTRACT

- □ ×

事故管理系统中，应急响应单元的选址 (location) 与配给 (allocation) 应该是一个整体，即如果响应单元选址时未同时考虑到相应资源配给的合理性，则往往无法达到最有效果。综上，对应急响应单元的规划可由选址-配给模型 (location-allocation model)⁴解释。本文提出了一种面向高速公路二次事故预防的应急车辆选址-配给模型。模型包括两个阶段：选址阶段与配给阶段，在主事故发生后进入选址阶段管控事故发展，配给阶段则旨在应对随机发生的二次事故。本模型克服了以往模型的诸多缺陷，如不考虑等待时间、返回站点规则、以及站点固定等情况。模型采用启发式算法求解，以实时求解大规模问题。

⁴选址配给模型，又称 LA 模型，是一种为一定量设施寻找最优空间位置以达到最大经济效益的模型。模型考虑服务的供给者与需求者，同时配以约束条件，往往通过混合整数规划求解。所谓“选址”，即为设施进行选址，而“配给”则是将服务分配给最合理的需求者。

6.2.1 Introduction

- 在事故发生后及时合理地调动应急响应单元有助于减少伤亡降低影响。且在一部分响应单元被安排管控交通事故后，剩余的响应单元也应该重分配，以更好地应对未来可能的交通事故；
- 在重分配相应单元时，管理中心不知道未来事故信息。传统模型往往假设事故的发生是独立同分布的（泊松分布），进而预测最有可能发生事故的位置，并将其分配至相关位置；
- 但上述传统思路忽略了三点关键信息：

事故的发生并不完全独立同分布 一般事故的发生可以近似为独立同分布，但主事故与二次事故则明显具有不同的分布特征。考虑路网中相近的两个位置：位置 1 与位置 2，前者位于后者上游。如果认为各自发生事故的概率是独立的，则“事故先发生于 2 再发生于 1”与“事故先发生于 1 再发生于 2”具有相同的概率，然而考虑到二次事故的特殊性，显然以上两事件是不对称的。特别需要说明的是，因为事故的发生会造成通行能力折减，因此在向二次事故位置部署应急响应时，不仅要克服二次事故的影响，还有克服已有主事故的影响，代价更大，因此将二次事故与一般事故等同的假设实际上低估了二次事故的影响，也降低了系统对二次事故的应对能力；

忽略了事故发生的顺序 同样考虑位置 1 与位置 2，假设模型推算得前者发生事故的概率为 10%，后者为 90%，则很可能将响应单元部署至位置 2。但如果位置 1 先于位置 2 发生事故，此时要么调动更远处的响应单元，要么等待位置 2 处的相应单元可以重新被调动，显然无论如何将延长对事故的响应时间；

事故的发生概率由多个时间步共同决定 已有研究在计算事故发生概率时往往仅考虑单个时间步，但未来二次事故发生的条件概率显然既取决于当前事故信息，也取决于过去的事故信息。

4. 通过明确考虑未来事故的随机信息，本研究展望了应急响应单元的选址配给模型。因为引入了随机信息，规划过程也变为随机规划 (**stochastic programming**)，求解该问题有助于减少延迟、油耗和排放。本研究设定应急响应单元的候选分配位置为有限的几个区域，故模型为离散选址配给模型；
5. 选址配给模型包括两阶段——在事故发生前，模型决定应急响应单元的位置（即选址阶段）；在事故发生后，模型对应急响应单元进行分配（即配给阶段）；
6. 考虑到主事故的发生会降低通行能力，从而降低对二次事故的响应效率，模型基于主事故信息对可能的响应延迟进行预测；
7. 另外，以往的研究常常要求应急响应单元在完成事故处理后返回永久或临时站点（即 **returning rule**），这一假设将带来不必要的出行安排，本研究不采用这一设定，而是假设响应单元在工作完成后留在原事故地点，等待下一时间步的重分配。

6.2.2 Literature review

1. 确定性选址问题 (**deterministic location problem**) 是最基础的选址问题，常见的确定性选址问题可分为：
中值问题（中位问题） 中值问题的目标是最小化所有需求点到最近服务者的平均权重距离（时间、成本）。P- 中值问题 (**P-Median problem**) 是典型的中值问题，要求在备选工厂集中指定 p 个工厂，使得需求点到其最近工厂的加权距离总和最小；
中心问题 中值问题的目标是最小化总成本，但对于部分场合中值问题并不合适。例如应急救援时要求应急救援团队尽可能快的到达任意现场，此时关注的是供给者到网络中需求点的最大距离，即中心问题。中心问题的目标是最小化需求点到最近服务者的距离的最大值。**P- 中心问题 (P-Center problem)** 是典型的中心问题，要求选择 p 个工厂，使得所有需求点得到服务，且要求需求点到其最近设施的距离的最大值最小；
覆盖问题 覆盖问题与中心问题的使用场景类似。模型预设供给者的服务范围，进而优化供给者的位置或数量。覆盖问题分为集合覆盖问题 (**location set covering problem**) 和最大覆盖问题，前者要求用最少的设施覆盖所有的需求点；后者则在给定的设施数量下寻求覆盖尽可能多的需求点。覆盖问题是中心问题的变形，要求距离小于一给定阈值，适用于诸如要求救援团队在规定时间内到达现场的场景。
2. 在确定性问题的基础上考虑部分或全部的不确定性因素，即为概率选址问题 (**probabilistic location problem**)。常见的不确定性因素包括服务者数量、需求点位置、需求量以及运输时间等等。在应急响应场景下，因需求量过多和需求点位置变化导致的服务设施繁忙是最主要的考虑因素。为描述这一现象，一般引入服务设施被占用的概率 q ，整体的建模思路又可分为基于概率论与基于排队论两类：
基于概率论的概率选址问题 此类模型的基本假设是各服务设施是否被占用相互独立，服从二项分布，此时 q 由一次呼叫的平均持续时间（小时） \bar{t} 、需求点 i 每天的呼叫频数 w_i 和服务设施的总数 p 共同

决定

$$q = \frac{\bar{t} \sum w_i}{24p}$$

在最大覆盖问题的基础上引入 q , 即为最大期望覆盖选址问题 (**maximum expected covering location problem**), 其最大的特点是以数学期望的形式将不确定信息体现在目标函数上。1989年, 在最大期望覆盖选址问题的基础上, 提出了最大可达性选址问题 (**maximum availability location problem**) 模型, 模型进一步引入了可靠性 α , α 为预设, 要求

$$\Pr(\text{邻域内至少有一个服务设施可提供服务}) \geq \alpha$$

通过 α 的引入模型将不确定信息体现在约束条件上。最大期望覆盖选址问题与最大可达性选址问题均为最大覆盖问题的发展, 类似地, 在集合覆盖问题的基础上引入 q, α , 即为概率集合覆盖问题 (**probabilistic location set covering problem**) ;

基于排队论的概率选址问题 尽管以上概率选址问题模型考虑了随机变量, 但其关于服务设施繁忙独立服从二项分布的假定并不完全合理, 在实际应用中, 所有服务设施均处于控制中心的调度之下。基于排队论的概率选址问题模型进一步放宽了这一限制, 模型基于排队论计算某需求点邻域内所有服务设施均处于饱和状态的概率。将排队论与最大可达性选址问题结合, 即为排队最大可达性选址问题 (**queuing maximum availability location problem**)。

6.2.3 Stochastic process of incident occurrence

1. 本节主要介绍未来事故发生的随机过程;
2. 首先考虑事故的发生概率。记在时间阶段 r 、位置 i 处发生事故的概率为 $\tau(i, r)$, 则显然有

$$\mathbb{E}[\tau(i, r)] = \Pr_{i,r}^P + \Pr_{i,r}^S$$

式中 $\Pr_{i,r}^P, \Pr_{i,r}^S$ 分别表示在时间阶段 r 、位置 i 处发生主事故 (或一般事故) 与二次事故的概率;

3. 因为主事故与一般事故的发生独立同分布, 则可假设其服从泊松分布, 此时相邻事故的时间长度服从指数分布。记 λ_i^r 为时间阶段 r 内、位置 i 处主事故 (一般事故) 的平均发生次数。当该时间段内发生 1 起事故时

$$\Pr_{i,r}^P = \frac{\lambda_i^r e^{-\lambda_i^r}}{\sum_i \lambda_i^r e^{-\lambda_i^r}}$$

式中分子为随机变量取 1 时的泊松分布概率, 分母则是标准化参数保证 $\sum_i \Pr_{i,r}^P = 1$, λ_i^r 可由实测数据估算;

4. $\Pr_{i,r}^S$ 是主事故概率 $\Pr_{k,u}^P$ 的函数, 同时与主事故严重性 (如封锁车道数, 伤亡和财产损失情况等, 记为 Ω) 和上游交通流状态 (事故发生前后的速度差, 记为 Δ) 有关。进一步地定义函数 $\delta(\Omega, \Delta)_{(k,u)(i,r)}$ 量化发生于位置 k 、时段 u 的事故与发生于位置 i 、时段 r 处的事故之间的关系, 当两事故完全无关时则为 0。 $\delta(\Omega, \Delta)_{(k,u)(i,r)}$ 的建立了主事故与二次事故间的关系, 越大表示事故越容易引发二次事故;
5. 假设自上游到下游的依次三个位置: 位置 1、位置 2、位置 3, 并认为二次事故既可能发生于主事故持续期间, 也可能发生于主事故消除后的恢复期, 因此将二次事故的条件概率的预测分为两个时间段。考虑位置 2 的二次事故条件概率, 在第一个阶段, 条件概率既与当前下游位置 3 的主事故概率有关 (持续期), 又与之前上游位置 1 的主事故概率有关 (恢复期); 在第二个阶段, 条件概率既与当前下游位置 3 的主事故概率有关 (持续期), 也与当前上游位置 1 的主事故概率有关 (持续期)。综上, 时段 r 的二次事故概率不仅与上一时段 $r-1$ 的主事故概率有关, 也与上上时段 $r-2$ 的主事故概率有关

$$\Pr_{i,r}^S = \sum_k \delta(\Omega, \Delta)_{(k,r-1)(i,r)} \Pr_{k,r-1}^P + \sum_k \delta(\Omega, \Delta)_{(k,r-2)(i,r)} \Pr_{k,r-2}^P$$

代入 $\Pr_{i,r}^S$, 得

$$\mathbb{E}[\tau(i, r)] = \Pr_{i,r}^P + \Pr_{i,r}^S = \Pr_{i,r}^P + \sum_k \delta(\Omega, \Delta)_{(k,r-1)(i,r)} \Pr_{k,r-1}^P + \sum_k \delta(\Omega, \Delta)_{(k,r-2)(i,r)} \Pr_{k,r-2}^P$$

6. 记 $p(i, j)$ 表示首先于 $r + 1$ 时段位置 i 处发生事故，接下来于 $r + 2$ 时段位置 j 处发生事故的概率，显然有

$$p(i, j) = \mathbb{E}[\tau(i, r + 1)] \times \mathbb{E}[\tau(j, r + 2)]$$

注意到 $p(i, j)$ 包含了 $i = j$ 与 $i \neq j$ 的两种情况，两种情况具有不同的概率 $p(\forall i = j)$, $p(\forall i \neq j)$ 。另外注意到，在引入主事故与二次事故关系后， $p(i, j) \neq p(j, i)$ 。

7. 进一步地考虑事故清空时间期望。事故清空时间期望对应急响应单元的调度至关重要，如果期望的事故清空时间早于下一次事故的期望发生时间，则相关的响应单元即可参与下一次调度任务；
8. 清空时间在很大程度上影响了总延误。记位置 i 发生事故的总延误时间为 D_i ，总延误时间定义为所有车辆的延误时间之和，单位为 $h \cdot \text{veh}$, D_i 由交通流量 q_i 、折减后的通行能力 s'_i 和正常通行能力 s_i 共同决定。其中折减的通行能力即是指在响应时间 (**response time**) R_i 与标准事故清空时间 (**normal clearance time**) NC_i 内的通行能力，显然 R_i , NC_i 越大， D_i 即越大。根据交通流理论，在 $R_i + NC_i$ 时段内，流入车辆数为 $(R_i + NC_i)q_i$ 、流出车辆数为 $(R_i + NC_i)s'_i$ ，则在 $R_i + NC_i$ 时刻，排队的车辆数为 $(R_i + NC_i)(q_i - s'_i)$ ，假设初始时刻无车辆排队，且 $R_i + NC_i$ 时段内排队车辆数线性增长，则

$$D_i = \frac{q_i - s'_i}{2}(R_i + NC_i)^2$$

定义平均延误时间 d_i 为总延误时间除以所有参与过排队的车辆数，又因为 $R_i + NC_i$ 时段内流入的全部 $(R_i + NC_i)q_i$ 辆车均参与排队，则

$$d_i = \frac{q_i - s'_i}{2q_i}(R_i + NC_i)$$

9. 上式中，响应时间 R_i 一般指从指挥中心得知事故发生到第一批响应单元到达事故地点的时间，标准事故清空时间 NC_i 即是指在第一批响应单元到达至事故被清除的时间，一般采用统计数据的平均值。然而，应急响应单元可能包括不止一批，后续响应单元的到达也会对事故的清除产生显著影响，特别是当后续响应单元遭遇延误（**响应延误 (response delay)**）时将不利于对事故的管理，而响应延误又与具体的响应类型有关。定义系数 β_i^η 以区分不同响应延误类型，上标 η 表示不同的响应延误类型，则在位置 i 处发生事故，且响应延误类型为 η 时的事故清空时间为

$$NC_i\beta_i^\eta$$

上式中 $\eta = 1$ 表示不存在响应延误； $\eta = 2$ 表示响应延误由协调高速公路行动响应小组 (coordinate highway action response team, CHART) 造成； $\eta = 3$ 表示响应延误由其它响应方式造成； $\eta = 4$ 表示响应延误由 CHART 与其它响应方式共同造成； $\eta = 5$ 表示 CHART 没有响应；

10. 特别地，定义纯清空时间 (**pure clearance time**) C_i 如下

$$C_i = NC_i\beta_i^1 < NC_i$$

纯清空时间除去响应延误的影响，有助于避免高估事故的清空时间。在本优化模型中，纯清空时间作为模型的输入值，目标函数为最小化总延误时间。

6.2.4 Stochastic ERU deployment model

- 在标准的两阶段随机规划中，第一阶段决策属于预防性决策，发生在实际的随机事件发生前；第二阶段决策属于修正性决策，在随机事件发生后对第一阶段决策进行调整；
- 研究仅考虑两类随机变量：事故的发生与位置。在未发现事故时，应急响应单元部署于预定位置，在事故清空后相关响应单元将留在原地直至下一次事故发生；
- 模型相关变量见 6*II 节。模型的目标函数是最小化所有场景下的总延误时间，即

$$\min z = \sum_w \sum_o P_w d_{ow}$$

约束条件如下。具体地，包括以下几部分：

响应单元配给 约束条件 1 确保在任何场景 w 下, 事故 o 被列为车辆 (即响应单元) n 的第 j 项任务的前提条件是在此之前已有事故 $p < o$ 被列为车辆 n 的第 $j - 1$ 项任务; 约束条件 2 确保只能有一起事故被列为车辆 n 的第 j 项任务; 约束条件 3 确保每一次事故都得到分配; 约束条件 4 为初始条件; 约束条件 5 确保每一辆车只有一个起点;

事故开始处理时间 约束条件 6 确保若事故 o 被分配为车辆 n 的第一项任务, 则车辆 n 赶到的时间不能早于其从起点出发到事故地点的时间; 约束条件 7 确保若事故 o 被分配为车辆 n 的第 $j(\neq 1)$ 项任务, 则车辆 n 赶到的时间不能早于其清空上一事故后从上一事故位置赶到该事故位置的时间, 式中 $M_{o,w}^7$ 的引入旨在使得 $a_{onjw} = 0$ 时仍满足上式, 上标 7 指其应用于约束条件 7; 约束条件 8 确保若事故 o 被分配为车辆 n 的第 $j(\neq 1)$ 项任务, 则车辆 n 赶到的时间不能早于事故的发生时间加上车辆从上一事故位置赶到该事故位置的时间; 约束条件 9 确保若事故 o 被分配为车辆 n 的第 $j(\neq 1)$ 项任务, 则车辆 n 赶到的时间不能早于其清空上一事故的时间; 约束条件 10 定义事故 o 开始得到处理的时间 (即车辆赶到的时间);

事故完成处理时间 约束条件 12 定义事故 o 完成处理的时间; 约束条件 13 定义事故的最早完成处理时间;

变量约束 约束条件 14 定义了事故的延误时间。

$$a_{onjw} \leq \sum_{p < o} a_{pn(j-1)w} \quad \forall w, n, o, j \neq 1 \quad (6.14)$$

$$\sum_o a_{onjw} \leq 1 \quad \forall w, n, j \quad (6.15)$$

$$\sum_n \sum_j a_{onjw} = 1 \quad \forall w, o \quad (6.16)$$

$$a_{111w} = 1 \quad \forall w \quad (6.17)$$

$$\sum_i x_{in} = 1 \quad \forall n \quad (6.18)$$

$$sv_{onw} \cdot a_{on1w} \geq \sum_i TT_{iL_{ow}} \cdot x_{in} \cdot a_{on1w} + H_{ow} \cdot a_{on1w} \quad \forall w, o, n \quad (6.19)$$

$$\begin{aligned} sv_{onw} \cdot a_{onjw} \geq & \sum_{p < o} TT_{L_p L_o} \cdot a_{pn(j-1)w} \\ & + \sum_{p < o} cv_{pn(j-1)w} \cdot a_{pn(j-1)w} - M_{o,w}^7 (1 - a_{onjw}) \end{aligned} \quad \forall w, n, o, j \neq 1 \quad (6.20)$$

$$\begin{aligned} sv_{onw} \cdot a_{onjw} \geq & \sum_{p < o} TT_{L_p L_o} \cdot a_{pn(j-1)w} \\ & + H_{0w} \cdot a_{pnjw} - M_{o,w}^8 (1 - a_{onjw}) \end{aligned} \quad \forall w, n, o, j \neq 1 \quad (6.21)$$

$$sv_{onw} \cdot a_{onjw} \leq M_{o,w}^9 \sum_{p < o} f_{pn(j-1)w} \quad \forall w, o \neq 1, n, j \neq 1 \quad (6.22)$$

$$s_{ow} = \sum_n \sum_j sv_{onw} \cdot a_{onjw} \quad \forall w, o \quad (6.23)$$

$$cv_{onjw} \cdot a_{onjw} - sv_{onw} \cdot a_{onjw} - CD_{L_{ow}} \cdot a_{onjw} + \epsilon \cdot a_{onjw} \leq M_{o,w}^{11} \cdot f_{onjw} \quad \forall w, o, n, j \quad (6.24)$$

$$c_{ow} = \sum_n \sum_j cv_{onjw} \cdot a_{onjw} \quad \forall w, o \quad (6.25)$$

$$c_{ow} \geq s_{ow} + CD_{L_{ow}} \quad \forall w, o \quad (6.26)$$

$$c_{ow} - H_{ow} = d_{ow} \quad \forall w, o \quad (6.27)$$

$$f_{onjw}, a_{onjw} \in \{0, 1\} \quad \forall w, o, n, j \quad (6.28)$$

$$x_{in} \in \{0, 1\} \quad \forall i, n \quad (6.29)$$

4. 注意到以上部分约束条件含有非线性项 $sv_{onw} \cdot a_{onjw}$, $x_{in} \cdot a_{onjw}$, $cv_{onjw} \cdot a_{onjw}$, 因此以上规划属于非线性规

划。为提高计算速度, 将非线性规划线性化:

- 首先线性化 $sv_{onw} \cdot a_{onjw}$ 项, 令 $d1_{onjw} = sv_{onw} \cdot a_{onjw}$, 要求当 $a_{onjw} = 1$ 时有 $d1_{onjw} = sv_{onw}$, 当 $a_{onjw} = 0$ 时有 $d1_{onjw} = 0$, 因此增加以下两项约束条件:

$$d1_{onjw} \leq sv_{onw} \quad \forall w, o, n, j \quad (6.30)$$

$$d1_{onjw} \leq M \cdot a_{onjw} \quad \forall w, o, n, j \quad (6.31)$$

$$d1_{onjw} \geq 0 \quad \forall w, o, n, j \quad (6.32)$$

- 另外线性化 $x_{in} \cdot a_{onjw}$ 项, 令 $d2_{onjw} = x_{in} \cdot a_{onjw}$, 显然 $d2_{onjw}$ 为二值变量, 设置等价约束条件如下。注意到以下约束条件并不要求 $d2_{onjw} \in \{0, 1\}$, 是因为在 x_{in}, a_{onjw} 均为二值变量的前提下以下四项约束条件即可保证 $d2_{onjw}$ 为二值变量。主流的混合整数线性规划求解方法是将其放宽为一般线性规划, 因此要求 $d2_{onjw} \in \{0, 1\}$ 反而会增加计算复杂度;

$$d2_{onjw} \geq x_{in} + a_{onjw} - 1 \quad \forall w, o, n, j \quad (6.33)$$

$$d2_{onjw} - x_{in} \leq 0 \quad \forall w, o, n, j, i \quad (6.34)$$

$$d2_{onjw} - a_{onjw} \leq 0 \quad \forall w, o, n, j \quad (6.35)$$

$$d2_{onjw} \geq 0 \quad \forall w, o, n, j \quad (6.36)$$

- 最后线性化 $cv_{onjw} \cdot a_{onjw}$, 同样地令 $d3_{onjw} = cv_{onjw} \cdot a_{onjw}$, 并增加以下约束条件:

$$d3_{onjw} \leq cv_{onjw} \quad \forall w, o, n, j \quad (6.37)$$

$$d3_{onjw} \leq M \cdot a_{onjw} \quad \forall w, o, n, j \quad (6.38)$$

$$d3_{onjw} \geq 0 \quad \forall w, o, n, j \quad (6.39)$$

5. 本研究采用场景缩减 (*scenario reduction*) 提升大规模随机规划问题的求解速度, 具体的算法为快速前向选择法 (*fast forward selection*)。

6.2.5 Numerical examples

Illustrative case study

- 研究场景为美国马里兰州巴尔的摩环城高速 (I-695), 路段长 51 英里, 含 40 个出口以及与其它主要道路的交叉口。在 2014 年, 工作日早高峰时段高速管理中心有 4 组应急响应单元可供调配。历史数据包括历史事故数据和探测车的相关数据;
- 在事故发生后, 应急响应单元将被部署至事故位置附近的出口处。每一出口被视为一个节点。为减少节点规模, 对接距离对历史事故进行聚类——将相距不到 1.9 英里的事故归为一组, 可以将节点数减少至 17 个; 将相距不到 1.3 英里的事故归为一组, 可以将节点数减少至 34 个;
- 研究场景为环形高速, 因此任意两点间存在两条备选路径。本研究所提模型可应用于更复杂的网络, 当存在多条备选路径时可由最短路算法获取最短路, 从而得到相应的出行时间;
- 根据采集的数据, 平均事故持续时间为 19.8 分钟, 意味着在得知事故发生后应急响应单元需要 19.8 分钟清空事故。而平均每 18.5 分钟就会发生一起事故, 意味着事故发生时很可能没有空闲的应急响应单元。为此研究者将早高峰时段按指数分布 (均值为 18.5 分钟) 划分为多个子区间, 每当发生事故时将预测接下来两个时段的事故概率;
- 如果某次事故发生时之前指派的应急响应单元尚未到达目的地, 则可以重新求解模型得到新的分配指令;
- 模型基于当前阶段和上一阶段的事故信息 (Ω, Δ) 实时推测接下来两个阶段的事故场景 $(\mathbb{E}[\tau(i, r+1)], \mathbb{E}[\tau(j, r+2)])$, 假设考虑路网中的 n 个节点, 则将生成 n^2 种场景。模型中 $\lambda_i^r, \delta(\Omega, \Delta)_{(k,u)(i,r)}, NC_i, \beta_i^\eta, C_i$ 等参数的估计值均可由历史数据给出, 其中部分参数为常数, 另一部分则需实时更新。

Results

- 求解时首先仅考虑 1 组应急响应单元的情况，随后不断增加响应单元的数量以进行敏感性分析，同时判断最佳的响应单元数量（即继续增加响应单元基本没有边际效益）；
- 在事故发生前应急响应单元会被提前部署于最优位置（最有可能发生事故的位置），事故发生后更新模型，重新部署响应单元；
- 事故的期望清空时间 \hat{C}_i 为模型的输入值。如果清空时间早于下一次事故的发生时间则对所有 n 组响应单元重新部署，反之则仅重部署 $n - 1$ 组单元；
- 应急响应单元的行程时间为模型的输入值。一般情况下行程时间受交通环境影响，事故造成拥堵后应急车辆也会被迫排队，但当高速公路有足够宽的路肩时，排队车辆即可以为应急车辆让路，行程时间大大缩短。因此研究考虑行程时间最短（自由流）和最长（拥挤）两种情况，分别作为输入值；
- 当只有 1 至 2 组应急响应单元时，有无考虑二次事故概率并不影响模型的求解质量，但当应急响应单元更多时，区分二次事故概率可以显著降低总延误时间（响应单元行程时间、响应延误与清空时间之和），主要表现为行程时间与延误时间的降低。

6.2.6 英汉互译

English	Chinese	English	Chinese	English	Chinese
interdependent	互相依存的	explicitly	清楚地	dispatcher	调度员
dice	骰子	overlook	忽视 (v)	hedge	限制 (v)
prompt	迅速的、促使 (v)	accrue	(逐渐) 增长 (v)	discount	折扣 n、低估 v
in response to	对…有反应	corridor	走廊、通道	interrelation	相互关系
convex	凸的	recourse	依靠 n、求助 n	branch and bound	分支界定
equate	使等同	reclusive	独居的	tractable	易处理的
redistribute	重分配 v	pend	悬而未决 v	stipulate	明确要求 v

6.3 Demand-driven timetable design for metro services (TRC, 2014)

ABSTRACT

- □ ×

地铁时刻表设计对地铁系统至关重要。最主流的策略是区分高峰、平峰设计静态方案。然而相应策略仍可能无法匹配动态客流需求，造成站点饱和与车内拥挤。本文构建了三种基于数据的优化模型以优化需求敏感的地铁时刻表。模型一旨在增强时刻表的动态性；模型二在模型一的基础上考虑了容量约束；模型三旨在生成考虑容量约束且需求敏感的高峰-平峰地铁时刻表。以新加坡地铁网为场景进行模型验证，结果显示模型二的整体效果最优；模型一则可得到最优的地铁调度时间配置；模型三生成的方案表现不及前两者，但便于应用与用户理解。

6.3.1 Introduction & Background

- 随着城市交通规模的发展，很多城市以提升公共交通便捷性作为主要任务。受益于大运力、高速度、准时等优点，轨道公交系统对都市交通系统至关重要；
- 为提升服务质量并减少乘客等待，近期研究聚焦于设计高效的地铁运营策略以提升服务可靠性，包括车速动态调整、优化停站时间、优化时刻表等。其中时刻表优化被认为是最直接且最有效的手段；
- 为提供以乘客为中心的地铁服务，其时刻表设计的核心是满足客流需求，最小化等待时间并避免过饱和；
- 在缺少高精客流需求时间分布信息的情况下，传统的方法是为平峰和高峰时段分别指定静态地铁班次。然而因为客流需求并不确定，仍可能出现供需失衡。另外忽视客流需求时间分布也可能引发扰动并降低服务可靠性。因此掌握客流需求动态变化并实时调整服务频次至关重要；
- 得益于地铁收费系统，如今可以提取数天至数周时段内的个体精度的时空行程信息；
- 本文的主要贡献包括以下两方面：1) 提出了三种需求驱动的地铁时刻表优化模型，模型一不考虑地铁运力约束，后两者则考虑；2) 基于地铁刷卡数据深度分析客流需求日内变异特征，提取高精时空客流信息；
- 已有的关于时刻表设计的研究大体可分网络时刻表设计与单线时刻表设计两类，本研究属于后者：

- 网络时刻表设计问题一般针对整体公交线网，旨在通过多线协同与同步最小化换乘时耗；
- 单线时刻表设计问题一般针对单一公交线路，旨在通过确定其服务频次以最大化乘坐体验并最小化等待时间，或将目标函数设为最大化公交到达频率分布与客流到达需求的相关性。相应约束包括津贴、车队规模与载客量。

8. 相比于常规公交系统，轨道公交时间可靠性较高，时刻表优化较为简单。

6.3.2 Timetable design problem

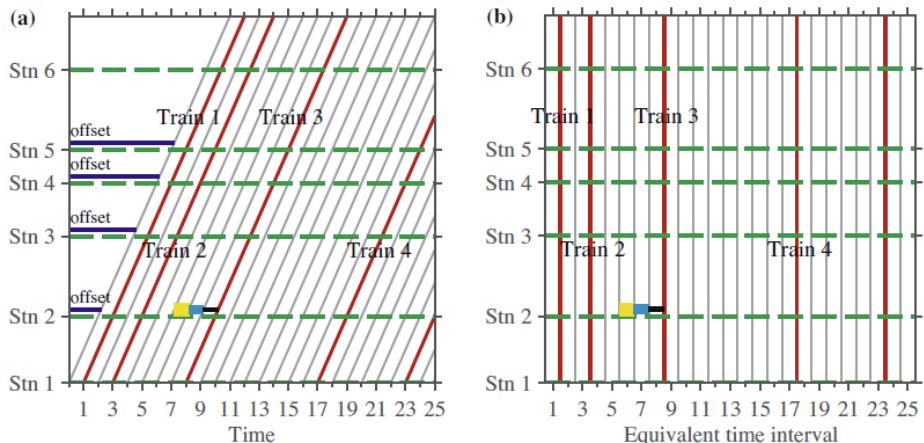


图 6.1 地铁时空轨迹表示方法。
横坐标为时间，纵坐标为站点，红线为同一线路地铁各班次轨迹：(a) 以全局统一时间为横坐标；(b) 不同站点的时间坐标系不一致，以首班车依次到达各站点的时刻作为各站点时间坐标系的基准时刻，此时每一班车的时空轨迹图为垂线。

1. 为描述各站点的地铁到站情况，可首先基于回归模型拟合线路地铁时空轨迹。假设地铁运行高度可靠，则不同班次地铁轨迹互相平行，并假设线路全段运行速度均等，则地铁时空轨迹可由直线描述（见图 6.1 (a)，线路的时刻表优化问题即可简化为起点站的发车时刻优化问题。优化时进一步离散化时间域；
2. 因为不同站点的地铁到站时间不同，一致的客流需求时间分布作用于不同站点影响不同。为便于数学描述，针对不同站点设计不同的时间坐标系（见图 6.1 (b)）。**基于等效时间 (equivalent time) 人为同步不同站点的时间信息。**对于站点 s ，其等效时间定义为真实时间与首班车到站时间之差；
3. 结合以上建模描述，研究基于假设 **A1-A2**，并考虑若干约束 **(C1-C5)**：
 - A1 服务可靠：**假设地铁运行速度与等待时间一致，时空轨迹完全相同，显然为强假设；
 - A2 到达均匀：**对于任意站点 s ，等效时段 t 内的乘客进站需求分布均匀；
 - C1 发车时间离散：**规定地铁的发车时间为等效时段 t 的末时刻；
 - C2 运行成本：**每日的地铁班次数固定；
 - C3 运行安全：**车头时距不小于最小要求；
 - C4 服务水平：**进站乘客需在给定时间内上车；
 - C5 最后服务：**末班车的发车时间固定。
4. 进一步给出问题的数学形式。首先定义等待时段 p 。假设乘客于等效时段 u 进站，于等效时段 t ($t \geq u$) 上车，则定义 $p = t - u + 1$ ，而同一时段内进站乘客的平均等待时间则为 $p - 0.5$ 。**设置优化目标为最小化所有乘客的等待时间；**
5. 定义所有数学符号如下：
 - **集合：**
 - T : 等效时段集合, $T = \{1, 2, \dots, T_n\}$;
 - S : 不含终点站的站点集合, $S = \{1, 2, \dots, S_n\}$;
 - P : 等待时段集合, $P = \{1, 2, \dots, P_n\}$ 。
 - **上下标：**
 - t, u : 分别关于地铁服务与用户到达的等效时段标签, $t, u \in T$;
 - s : 站点标签, $s \in S$;
 - p : 等待时段标签, $p \in P$ 。

- 参数:

- CAP : 地铁最大客容量;
- T_n, S_n : 等效时段与不含终点站的站点数量;
- K_n : 每日地铁班次;
- P_n : 等待时段数量;
- $B_s^u, B_{s,d}^u$: 站点 s 于等效时间 u 的进站客流总需求和以站点 d 为目的地的需求, 有 $B_s^u = \sum_{d \in S} B_{s,d}^u$;
- N_{\max}, N_{\min} : 最大与最小车头时距 (以时段数为单位)。

6. 模型一假设地铁容量无限, 因此同一时段的进站乘客具有相同的等待时段 p 。模型的决策变量为 $x_t \in \{0, 1\}$, $y_{u,p}$, 前者为 t 时段末时刻有无地铁发车的指示变量, 后者表示等效时段 u 内进站的乘客等待时段为 p 的比例。则在乘客均匀到达假设下等效时段 u 内的进站乘客的平均等待时间 w^u 可以表示为

$$w^u = \sum_{p \in P} y_{u,p} (p - 0.5)$$

则模型一假设下的时刻表优化问题可建模为如下混合整数线性规划问题

$$\begin{aligned} \min \quad & \sum_{u \in T} w^u \cdot \left(\sum_{s \in S} B_s^u \right) \\ \text{s.t.} \quad & x_t \in \{0, 1\}, \quad \forall t \in T \\ & \sum_{t \in T} x_t = K_n \\ & \sum_{t \in [t_1, t_2]} x_t \leq 1, \quad t_2 = t_1 + N_{\min} - 1, \quad \forall t_1, t_2 \in T \\ & \sum_{t \in [t_1, t_2]} x_t \geq 1, \quad t_2 = t_1 + N_{\max} - 1, \quad \forall t_1, t_2 \in T \\ & x_{T_n} = 1 \\ & 0 \leq y_{u,p} \leq 1, \quad \forall u \in T, \forall p \in P \\ & \sum_{p \in [1, \min(T_n - u + 1, P_n)]} y_{u,p} = 1, \quad \forall u \in T \\ & x_t \geq y_{u,p}, \quad t = u + p - 1, \quad \forall t \in T, \forall p \in P \end{aligned}$$

因为假设地铁容量无限, $y_{u,p}$ 实际上也为 0-1 变量, 但因为离散变量的优化难度远大于连续变量, 故在建模时通过约束进行隐式表示。重点关注最后两约束。其中集合 $[1, \min(T_n - u + 1, P_n)]$ 表示 u 时段进站的乘客所可能的等待时长的范围, 因此倒数第二条约束使得 u 时段进站的乘客最晚于 $\min(T_n - u + 1, P_n)$ 时段全部上车。而最后一条约束中 $t = u + p - 1$ 为等待时长 p 已知时 u 时段进站乘客的上车时间, 该式使得当 $x_t = 0$ 时, 必然有 $y_{t-p+1,p} = 0$, 而若 $y_{u,p} > 0$, 则必然有 $x_{u+p-1} = 1$, 结合其它约束, 则最优解的 $y_{u,p}$ 只能取 0 或 1。模型一适用于地铁运力相对充分的情况, 而且其结果有助于推断现有客流需求及地铁总班次下地铁所承载的最大需求, 以进一步优化车厢内空间 (如减少座位增加站立空间) 满足运力需求;

7. 模型二在模型一的基础上进一步考虑了地铁容量限制, 其决策变量包括 $x_t \in \{0, 1\}$, $y_{u,p}^s$, q_t^s , 其中 $y_{u,p}^s$ 表示等效时段 u 内进入地铁站 s 的乘客等待时段为 p 的比例; q_t^s 表示等效时段 t 末时刻到达地铁站 s 并完成上下客后的地铁的占用率。此时等效时段 u 内的地铁站 s 的进站乘客平均等待时间 w_s^u 可以表示为

$$w_s^u = \sum_{p \in P} y_{u,p}^s (p - 0.5)$$

同样将模型二的时刻表优化问题建模为混合整数线性规划问题

$$\min \quad \sum_{s \in S} \sum_{u \in T} w_s^u \cdot \left(\sum_{d \in S} B_{s,d}^u \right)$$

$$\begin{aligned}
& \text{s.t. } x_t \in \{0, 1\}, \quad \forall t \in T \\
& \sum_{t \in T} x_t = K_n \\
& \sum_{t \in [t_1, t_2]} x_t \leq 1, \quad t_2 = t_1 + N_{\min} - 1, \quad \forall t_1, t_2 \in T \\
& \sum_{t \in [t_1, t_2]} x_t \geq 1, \quad t_2 = t_1 + N_{\max} - 1, \quad \forall t_1, t_2 \in T \\
& x_{T_n} = 1 \\
& 0 \leq y_{u,p}^s \leq 1, \quad \forall s \in S, \forall u \in T, \forall p \in P \\
& \sum_{p \in [1, \min(T_n - u + 1, P_n)]} y_{u,p}^s = 1, \quad \forall s \in S, \forall u \in T \\
& x_t \geq y_{u,p}^s, \quad t = u + p - 1, \quad \forall s \in S, \forall t \in T, \forall p \in P \\
& q_t^s = \begin{cases} \sum_{u \in [\max(1, t - P_n + 1), t]} \sum_{d \in [2, S_n]} B_{1,d}^u y_{u,t-u+1}^1 & s = 1, \forall t \in T \\ q_t^{s-1} - \sum_{u \in [\max(1, t - P_n + 1), t]} \sum_{o \in [1, s-1]} B_{o,s}^u y_{u,t-u+1}^o + \sum_{u \in [\max(1, t - P_n + 1), t]} \sum_{d \in [s+1, S_n]} B_{s,d}^u y_{u,t-u+1}^s & \forall s - 1, s \in S, \forall t \in T \end{cases} \\
& q_t^s \leq CAP, \quad \forall s \in S, \forall t \in T
\end{aligned}$$

与模型一相比，模型二主要增加倒数第二条约束以限制地铁客流量。其中 $\max(1, t - P_n + 1)$ 表示 t 时段上车的乘客的最早进站时段。显然模型二的决策变量维数更大，约束也更多，因此求解难度高于模型一；

8. 模型三在模型二的基础上进行简化，设计地铁时刻表时仅考虑高峰发车间隔和平峰发车间隔两种间隔。

模型三的决策变量和目标函数同模型二

$$\begin{aligned}
& \min \sum_{s \in S} \sum_{u \in T} w_s^u \cdot \left(\sum_{d \in S} B_{s,d}^u \right) \\
& \text{s.t. } x_t \in \{0, 1\}, \quad \forall t \in T \\
& \sum_{t \in T} x_t = K_n \\
& x_{T_n} = 1 \\
& 0 \leq y_{u,p}^s \leq 1, \quad \forall s \in S, \forall u \in T, \forall p \in P \\
& \sum_{p \in [1, \min(T_n - u + 1, P_n)]} y_{u,p}^s = 1, \quad \forall s \in S, \forall u \in T \\
& x_t \geq y_{u,p}^s, \quad t = u + p - 1, \quad \forall s \in S, \forall t \in T, \forall p \in P \\
& q_t^s = \begin{cases} \sum_{u \in [\max(1, t - P_n + 1), t]} \sum_{d \in [2, S_n]} B_{1,d}^u y_{u,t-u+1}^1 & s = 1, \forall t \in T \\ q_t^{s-1} - \sum_{u \in [\max(1, t - P_n + 1), t]} \sum_{o \in [1, s-1]} B_{o,s}^u y_{u,t-u+1}^o + \sum_{u \in [\max(1, t - P_n + 1), t]} \sum_{d \in [s+1, S_n]} B_{s,d}^u y_{u,t-u+1}^s & \forall s - 1, s \in S, \forall t \in T \end{cases} \\
& q_t^s \leq CAP, \quad \forall s \in S, \forall t \in T \\
& x_{t+N_{Peak}} + x_{t+N_{Off}} \geq x_t, \quad \forall t, t + N_{Peak}, t + N_{Off} \in T \\
& x_{t+N_{Peak}} + x_{t+N_{Off}} \leq 1, \quad \forall t, t + N_{Peak}, t + N_{Off} \in T, \text{ if } N_{Off} \bmod N_{Peak} \neq 0 \\
& \sum_{t \in [t_1, t_2]} x_t \leq 1, \quad t_2 = t_1 + N_{Peak} - 1, \quad \forall t_1, t_2 \in T
\end{aligned}$$

式中 N_{Peak} , N_{Off} 为预设的高峰发车间隔和平峰发车间隔（为避免过度失衡，研究令 $N_{Off} < 3N_{Peak}$ ）。倒数第三道约束限制发车间隔必须选 N_{Peak} 或 N_{Off} ，且在 N_{Peak} , N_{Off} 不满足整数倍数关系时 ($N_{Off} \bmod N_{Peak} \neq 0$)，进一步引入倒数第二条约束避免同时选择 N_{Peak} 和 N_{Off} 。观察上述约束可知，模型三仅限制发车间隔，而不限制高峰间隔与平峰间隔的切换次数；

9. 最后分析三类模型的计算复杂度。**模型一等价于一维设施选址问题 (one-dimensional Facility Location Problem)**, 其中地铁出发时刻选择对应于设施选址, 乘客等待时间对应于需求点至设施位置的距离, 乘客数量对应于需求的权重。因此**模型一属于 P 问题**, 可以在多项式时间内求解。当不考虑最大和做小发车间距约束时, 考虑容量约束的**模型二和三等价于产量约束的 p-设施选址问题 (capacitated p-Facility Location Problem)**, 属于**NP 问题**。本研究基于基准求解器 (CPLEX 等) 求解上述问题。

6.3.3 Case study & Results and analysis

1. 以新加坡的一条地铁线路为场景测试所提的三类模型;
2. 基于一周的刷卡数据提取个体级的地铁出行时空需求。需要注意的是, 为集计研究线路的出行时空需求, 需考虑两类出行: 起终点均位于研究线路上的出行和仅途经该线路的多线换乘出行。对于后者, 需推测其途经该线路的具体路段。为此, 基于 MATSim 多智能体仿真平台, 以行程时间为出行效用, 在全市地铁线上仿真全部地铁需求, 得到均衡状态下的个体出行路径, 确定研究线路上的所有出行时空需求;
3. 基于余弦相似度 (cosine similarity) 量化地铁时空客流需求的变异性。定义第 i 天和第 j 天的进站客流相似度 $Sim_B(i, j)$ 为

$$Sim_B(i, j) = \frac{\sum_{s \in S} \sum_{u \in T} b_{s,u}^i b_{s,u}^j}{\sqrt{\sum_{s \in S} \sum_{u \in T} (b_{s,u}^i)^2} \sqrt{\sum_{s \in S} \sum_{u \in T} (b_{s,u}^j)^2}}, \quad b_{s,u}^i = \frac{B_s^u(i)}{\sum_s \sum_u B_s^u(i)}$$

上式 $Sim_B(i, j) \in [0, 1]$, 越接近 1 表示两天的时空需求越接近。分析一周内的地铁出行时空需求发现, 工作日内的地铁时空需求高度一致, 故以五天工作的平均时空需求作为后续实验的时空需求;

4. 测试时考虑两类场景。场景一考虑地铁运行全时段, 设计全线时刻表; 场景二仅考虑早高峰时段和线路中的部分站点;
5. 比较模型求解时间发现, 模型一的求解效率极高, 对于两类场景均在秒级时间内完成求解; 而考虑容量约束的模型二、三的求解效率则较低, 求解场景一的用时超过一小时, 求解场景二也需数分钟;
6. 对比模型得到的时刻表下的乘客等待时间和溢流情况, 模型二的结果最优, 模型一次之, 模型三最次;
7. 进一步评价模型所得时刻表对乘客需求变化的鲁棒性。在 85% 至 115% 范围内等比例放大或缩小当前客流需求, 评价模型所得时刻表在不同需求水平下的乘客等待时间和溢流情况。结果表明模型二所得结果的鲁棒性显著优于其它模型, 在客流需求小幅上升时模型一所得时刻表效果略优于模型三, 但在较大幅度增长后模型一与模型三结果的效果基本一致;
8. 注意到模型一的结果可作为问题的基准解。当地铁运力充分时, 模型一的结果即是最优解, 但当运力低于需求, 则其结果即存在偏差。因为线路运力由总班次和容量共同决定, 因此有必要评价地铁总班次 K_n 与地铁容量 CAP 对模型一计算结果的影响。

6.3.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
capacitated	带容量约束的	user-centric	用户中心的	subsidy	津贴
fleet	舰队、车队	irregularity	无规律行为	confine	限制(v)、范围
substantial	重大的				

6.4 Planning of static and dynamic charging facilities for electric vehicles in electrified transportation networks (Energy, 2023)

ABSTRACT



随着电动汽车和充电设施的发展, 电气化交通系统逐渐成型。动态无线充电技术有助于推动电动汽车的进一步普及。为提升电气化交通系统的综合质量, 研究固定和动态充电设施混合规划策略。考虑交通出行者对充电设施规划的反馈设计双层优化框架。下层模型以多类型车辆用户均衡为目标求解交通分配问题; 上层模型最优化电气化交通系统综合评价指

标。基于代理模型 (surrogate model) 方法求解该双层规划问题。基于案例验证了混合充电设施规划的优势和求解算法的可行性、有效性。[图](#)

6.4.1 Introduction

1. 低噪声、零排放的电动汽车是碳中和的关键。随着电动汽车的快速普及，充电设施也随之发展。与燃油车相比，电动汽车的缺点在于更长的充电时间和较少的充电站点，是限制电动汽车发展的主要因素；
2. 动态无线充电技术可无接触地为运动的电动汽车充电，不仅可节省驾驶员充电时间，也可更高效地利用道路资源。同时无线充电也可降低电动车电池包容量需求，从而降低电动车成本、促进电动车普及；
3. 电动汽车的普及将形成电网和交通网的耦合——路网中运动的电动汽车将影响充电需求于电网中的分布，而电价等因素也会影响电动车的路径选择。无线充电技术的发展将进一步促进这一耦合；
4. 已有研究关注电动车充电设施与电气化交通系统领域主要可分为系统运行和系统规划两个领域：
 - 电气化交通系统运行研究主要聚焦于现有电网-交通网基础上的管控策略研究。以收费为主要手段，可分为电价和道路收费两类。其中电价又可分为节点边际电价 (locational marginal price) 和零售价 (retail price) 两类。在充电设施上从固定设施到动态无线充电已有研究都有探讨；
 - 电气化交通系统规划研究则需要考虑基础设施的投资建设成本以研究最优布局。现有研究主要关注固定充电设施。部分关注动态无线充电的研究仅考虑交通网的运行状态。另一部分研究考虑了充电行为对电网的影响，考虑了电网-交通网耦合下的动态无线充电设施规划。但**动态无线充电往往只能提供较低的充电电压，而当具有迫切充电需求时则需要具有高电压的固定充电设施。因此需要同时考虑固定设施和动态无线充电设施的协同规划。**另外，建模充电需求时还需要考虑路网中不同类型的车辆（如燃油车不存在充电需求）。
5. 研究贡献主要为：
 - 构建了更现实的充电设施规划模型。考虑了电网-交通网的耦合、固定充电和动态无线充电、以及燃油车、低充电需求电动车和急切充电需求电动车三种类型车辆；
 - 基于双层优化框架求解混合充电设施规划策略以最大化耦合电气化交通系统综合性能。为高效求解大规模路网下的多类型交通分配，提出了改进自适应路径生成算法 (*modified adaptive path generation algorithm*) 以避免枚举所有有效路径。同时提出可变更新因子方法 (*variable updating factor method*) 以处理多类型车辆用户均衡时的互补约束。

6.4.2 System model

1.

6.4.3 英汉互译

English	Chinese	English	Chinese	English	Chinese
interdisciplinary	跨学科的	cone	圆锥	complementary	互补的

6.5 附录

6*I 模型参数说明 (A model for planning locations of temporary distribution facilities)

Symbol	Description
Acronyms	
TDC	Temporary Distribution Center
ADP	Aggregated Demand Point
CSP	Central Supply Point
Parameters	
\mathcal{U}	Set of potential TDCs indexed by j and by m
\mathcal{N}	Set of ADPs indexed by i

Symbol	Description
\mathcal{T}	Set of time periods indexed by t ($\mathcal{T} = \{1, 2, \dots, n\}$)
\mathcal{K}	Set of resource types indexed by k
S^k	Space required for storing each unit of resource type k
L_t	Length of time period t within the planned distribution period t
N	Number of time periods
D_u^k	Demand of resource type k at ADP i in time period t
C_t^k	Capacity of resource type k at CSP in time period t
V_j	Capacity of TDC j
F_j	Fixed cost of operation of a TDC j
τ_j^0	Travel time from CSP to TDC j
τ_{ji}	Travel time between TDC j to ADP i
TC_j^{k0}	Unit cost of transportation of resource type k from CSP to TDC j
TC_{mj}^{kl}	Unit cost of transportation of resource type k from TDC m to TDC j ($m, j \in \mathcal{U}$)
TC_{ji}^{kE}	Unit cost of transportation of resource type k from TDC j to ADP i
P_{itl}^k	Unit cost of delay penalty of satisfying demand of ADP i of resource type k of time period l in time period t
B	Big number
Decision variable	
y_{jt}	Binary variable that equals 1 if facility j is selected as TDC in time period t and 0 otherwise
x_{ijt}	Binary variable that equals 1 if ADP i receives resources from TDC j in time period t
r_{jt}^k	Quantity of resource type k allocated from the CSP to selected TDC j in time period t
q_{mjt}^k	Quantity of resource type k transferred from TDC m of time period $t - 1$ to TDC j of time period t
$a_{jt}^k = r_{jt}^k + \sum_m q_{mjt}^k$	Quantity of resource type k available at TDC j in time period t
z_{jlt}^k	Quantity of resource type k distributed from TDC j at time period t to satisfy the demand at ADP i generated at the same period
w_{jilt}^k	Total of resource k satisfied by TDC j at time period t to meet backordered demand of period l at ADP i ($l < t \in \mathcal{T}$)
$b_{ji}^k = \sum_i (z_{jlt}^k + \sum_l w_{jilt}^k)$	Total quantity of resource type k distributed to all ADP from TDC j in time period t

6*II 模型参数说明 (A Stochastic Emergency Response Location Model Considering Secondary Incidents on Freeways)

Indexes	
n	index n , set for incident response-units (vehicles)
i	index i , set of candidate locations of origins for response units (vehicles)
j	index j , set of jobs for each incident-response unit, n
o	index o , set for defining requested incidents
w	index w , set of scenarios
Input parameters	
TT_{ij}	Travel time of response-unit going from location i to location j
CD_i	Service time required for incident at node i , also called as clearance duration (CD)
L_{ow}	Location of incident o under scenario w
P_w	probability of scenario w
H_{ow}	Time that incident o happens under scenario w
M	Big-M used for modeling
ϵ	A very small number used for modeling
Decision variables	
x_{in}	Binary decision variable which equals to one if candidate location i is selected as the starting point for vehicle n and 0 otherwise.
a_{onjw}	Binary decision variable equals one if incident o is assigned as the j^{th} job in scenario w that vehicle n covers and 0 otherwise.
sv_{onw}	A continuous variable indicating the service start time for incident o if which vehicle n is going to serve under scenario w
cv_{onjw}	A continuous variable indicating the time of clearance of incident o if done as the j^{th} job by vehicle n under scenario w

d_{ow}	Delay of incident o under scenario w
s_{ow}	Time at which incident o starts getting served under scenario w and the vehicle is at the location of the incident
c_{ow}	Time at which incident o is cleared under scenario w
$d1_{onjw}$	Dummy variable used for linearization
$d2_{onjw}$	Dummy variable used for linearization
$d3_{onjw}$	Dummy variable used for linearization
f_{onjw}	Binary variable indicating whether incident o is served as the j^{th} job of vehicle n under scenario w ($= 1$) or not ($= 0$). The serving vehicle, n , has to be at the location of the incident for at least CD

6*III 混合 0-1 非线性规划线性化的一些方法

1. 对于非线性项

$$x \cdot y, \quad x \in \{0, 1\} \quad y \in [\underline{y}, \bar{y}] \quad \underline{y} \geq 0 \quad (6.40)$$

令 $z = x \cdot y$, 并增加等价约束条件如下, 使上式线性化。第一道公式使得 $x = 0$ 时有 $z = 0$, 且 $x = 1$ 时 z 与 y 具有相同的定义域; 第二道公式使得 $x = 1$ 时有 $z = y$, 且 $x = 0$ 时使得 $z = 0$ 位于区间内。

$$\begin{aligned} \underline{y}x &\leq z \leq \bar{y}x \\ -\bar{y}(1-x) + y &\leq z \leq y \end{aligned}$$

2. 对于非线性项

$$x \cdot y, \quad x \in \{0, 1\} \quad y \geq 0 \quad (6.41)$$

显然为上述情况的特殊情况 ($\underline{y} = 0, \bar{y} = \infty$), 同样地令 $z = x \cdot y$, 并代入 $\underline{y} = 0, \bar{y} = \infty$, 得到约束条件如下

$$0 \leq z$$

$$0 \leq z \leq y$$

可以看到因为 ∞ 的引入使得以上约束条件中不存在 x , 导致约束条件无法起到应有的效果, 因此引入一非常大的常数 M , 令 $\bar{y} = M$, 同样得到等价约束条件

$$\begin{aligned} 0 &\leq z \leq M \cdot x \\ -M(1-x) + y &\leq z \leq y \end{aligned}$$

3. 对于非线性项

$$\prod_i^N x_i, \quad x_i \in \{0, 1\} \quad (6.42)$$

令 $z = \prod_i^N x_i$, 显然当且仅当 $\forall x_i = 1$ 时才有 $z = 1$, 否则 $z = 0$, 则构造等价约束条件如下:

$$-z + \sum_i^N x_i \leq N - 1$$

$$z - x_i \leq 0$$

$$z \geq 0$$

第7章

网络控制

7.1 *Mitigating freeway off-ramp congestion: A surface streets coordinated approach (TRC, 2012)*

ABSTRACT

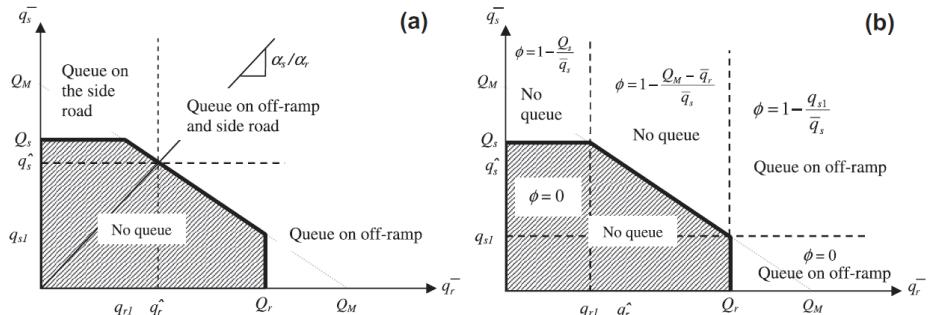
— □ ×

1. 出口匝道下游的堵塞可能沿匝道传递至高速公路上，进而形成高速公路上的瓶颈；
2. 多数研究关注队列在高速路上的形成与传播，却较少有研究关注队列形成的原因；
3. 如果将出口匝道上游的部分地面交通转移至其它街道即可增加出口匝道的通行能力，这也是本文的研究内容。具体地，提出了确定绕行流量的方法以提升系统通行能力并降低总延误，并研究在高峰时段下可使得该方法有效的条件。方法的目标是避免出口匝道拥堵、消散高速排列、并预留一部分通行能力满足冲突方向的地面交通。

7.1.1 Introduction

1. 出口匝道与地面道路相接存在两种形式：
 - 汇入形式：此时节点的通行能力受限于车辆的换道行为；
 - 交叉口形式：此时节点的通行能力受限于信号配时或主路方向上的流量。
无论何种形式，若出口或地面流量过大将在结点处产生拥堵，并同时沿匝道和地面道路向上游传播；
2. 已有的研究中缓解这一现象的控制策略大体可分为两类：
 - 第一类策略针对高速上的车辆，包括车道分配 (**lane assignment**) 和动态关闭拥挤出口匝道等；
 - 第二类策略针对地面交通，主要通过邻近交叉口的信号控制和几何改善实现。最原始的做法是当检测到出口匝道发生拥堵时，通过调整信号配时为出口匝道交通分配更多的优先权以消散匝道拥堵，但该方法也会打乱原有的信号配时。进一步地可基于高速、出口匝道和地面路网的交通状态协同邻近的信号灯以最小化总延误。
在具体应用控制策略时，研究者往往需要基于交通需求管控 (**traffic demand management, TDM**)、先进出行者信号系统 (**advanced traveler information system, ATIS**)、先进交通管控系统 (**advanced traffic management system**) 等技术；
3. 本文所提策略通过引导出口匝道上游部分地面交通绕行至邻近、平行且未被完全利用的通道为出口匝道驶出交通预留充分通行能力，并在满足该目标的前提下要求绕行流量尽可能低。策略并不假设出口匝道末端存在信号灯，因此更适合一般的出口匝道接入形式；
4. 本文所提策略适用于以下两种情况：
 - 出口匝道未拥堵，但地面交通存在部分绕行的条件。此时策略有助于平衡汇入节点附近的交通压力；
 - 汇入节点发生通行能力折减，地面路网和匝道均出现拥堵。此时策略有助于缓解拥堵。
5. 本策略基于以下两个重要假设，若实际条件不满足以下假设本策略的效果可能受限：
 - 汇入节点周围的路网存在剩余的通行能力；

图 7.1 出口匝道汇入节点需求量与排队情况关系图。 (a) 为无控制情况, (b) 为加入绕行引导后的情况, 其中 ϕ 表示绕行流量的比例。



- 高速公路下游不存在瓶颈。

7.1.2 Determining the stationary fraction to detour

1. 本节基于 Daganzo 等提出的汇入节点模型¹分析出口匝道和地面支线的排队模式并确定绕行流量的比例 (图7.1);
 2. 以下分析假设出口匝道是以汇入的形式接入地面路网, 但也适用于无信号或有信号交叉的情况;
 3. 记地面支线和出口匝道的需求量分别为 \bar{q}_s, \bar{q}_r ; 地面支线和出口匝道的流量分别为 q_s, q_r ; 地面支线、出口匝道和汇入节点的通行能力分别为 Q_s, Q_r, Q_M , 并不失一般性地假设 $Q_s < Q_M, Q_r < Q_M$; 若地面支线与出口匝道均发生拥堵, 则记支线与匝道的流量比为 α_s/α_r ,
 4. 在无控制的情况下 (图7.1(a)), 当匝道需求量超过其通行能力 ($\bar{q}_r > Q_r$) 或地面支线需求量超过其通行能力 ($\bar{q}_s > Q_s$) 或总需求量超过节点通行能力 ($\bar{q}_r + \bar{q}_s > Q_M$) 时均会发生排队, 但只有在 $\bar{q}_r > \hat{q}_r = \alpha_r Q_r, \bar{q}_s > \hat{q}_s = \alpha_s Q_s$ 时排队现象才会同时发生在匝道和地面支线上;
 5. 记 ϕ 表示绕行流量的比例, 控制策略的目标是尽可能避免排队现象的产生 (图7.1(b)):
- 当 $\bar{q}_r > Q_r$ 且 $\bar{q}_s \leq q_{s1} = Q_M - Q_r$: 此时匝道需求量超过其通行能力因此发生排队, 但总的需求量并未超过汇入节点通行能力, 因而匝道处的拥堵完全是由自身需求量过多造成的, 引导地面交通绕行并不能改善其消散率, 故此时进行绕行引导没有意义, 令 $\phi = 0$;
 - 当 $\bar{q}_r > Q_r$ 且 $\bar{q}_s > q_{s1}$: 此时匝道需求量超过其通行能力因此发生排队, 同时总需求量也超过汇入节点通行能力, 因而匝道处的拥堵一方面是由自身需求量过多造成, 另一方面也包括汇入节点处产生的溢流, 此时可引导部分地面交通绕行将地面支线需求量降至 q_{s1} , 显然有 $(1 - \phi)\bar{q}_s = q_{s1} \Rightarrow \phi = 1 - \frac{q_{s1}}{\bar{q}_s}$ 。绕行引导消除了汇入节点溢流对出口匝道的影响, 但出口匝道处的需求量仍过高, 仍会产生拥堵;
 - 当 $Q_M - Q_s = q_{rl} < \bar{q}_r < Q_r$ 且 $\bar{q}_s > Q_M - \bar{q}_r$: 此时匝道需求量并未超过其通行能力, 但总需求量超过汇入节点通行能力, 因而至少会有一个方向产生拥堵, 可引导部分地面交通绕行将总需求量降至 Q_M , 显然有 $\bar{q}_r + (1 - \phi)\bar{q}_s = Q_M \Rightarrow \phi = 1 - \frac{Q_M - \bar{q}_r}{\bar{q}_s}$ 。此时无论出口匝道或地面支线均不存在排队现象;
 - 当 $\bar{q}_r < q_{rl}$ 且 $\bar{q}_s > Q_s$: 此时地面支线需求量超过其通行能力, 但总需求量并未超过汇入节点通行能力, 因而排队只会发生在地面支线, 此时可引导部分地面交通绕行将支线需求量降至 Q_s , 显然有 $(1 - \phi)\bar{q}_s = Q_s \Rightarrow \phi = 1 - \frac{Q_s}{\bar{q}_s}$ 。此时无论出口匝道或地面支线均不存在排队现象。

7.1.3 Advisability of the strategy during a rush hour

1. 本节将以上基于静态模型得到的绕行引导策略应用于整个高峰时刻并确定该策略可发挥效果的条件;
2. 引导支线交通绕行至周围道路不可避免地增加了绕行车辆和周围道路原本车辆的成本, 因此本节进一步提出一个简单但有效的基于到达离去曲线图的模型判断绕行策略是否值得推荐;
3. 考虑出口匝道末端、相接地面支线、汇入节点、高速公路出口匝道下游路段和周围路网的平行道路共五种场景。定义 $A_r(t), A_s(t), A_M(t), A_f(t), A_p(t)$ 分别表示上述五种场景的累计期望到达曲线 ($A_M(t) = A_r(t) + A_s(t)$), $D_r(t), D_s(t), D_M(t), D_f(t), D_p(t)$ 分别表示累计离去曲线。结合有无绕行引导两种情况 (有引导情况由上标 c 表示)。到达曲线与离去曲线所包围的区域的面积 W 即为相应位置的总延误, 定义 d_r, d_s, d_f, d_p 为相应位置单车平均延误, n_r, n_s, n_f, n_p 为相应位置延误车辆数;

¹Daganzo C F, Newell G F. Methods of Analysis for Transportation Operations. Institute of transportation studies.

4. 假设匝道排队最先开始于汇入节点。记匝道排队的开始和结束时间为 t_i, t_e 。为研究绕行策略的合理性，只需计算在时间区间 $[t_i, t_e]$ 内所有车辆的总延误；
5. 进一步作如下假设：
 - 匝道需求量不会超过其通行能力；
 - 平行的道路原本不存在拥堵现象；
 - 当汇入节点总需求量超过其通行能力时开始产生排队，排队现象最先出现与出口匝道，随后再出行于地面支线上。
6. 当无控制时：
 - 汇入节点首先在 t_i 时刻因匝道需求量过高造成的总需求量过高而发生拥堵，排队现象最早出现于匝道上，匝道消散率固定为 Q_r ，节点消散率固定位 Q_M ；
 - t_i^f 时刻匝道的拥堵现象传递至高速公路主线上，造成主线拥堵；
 - t_i^s 时刻因为地面支线需求量的变化导致地面支线也发生排队，此时因为支线与匝道均出现排队，出口匝道消散率降为 $\alpha_r Q_M$ ，支线消散率为 $\alpha_s Q_M$ ；
 - t_e^s 时刻因为地面支线需求量的变化地面支线排队最先消失，此时只有匝道发生排队，出口匝道消散率重新恢复为 Q_r ；
 - t_e^f 时刻高速主线拥堵现象结束，并于 t_e 时刻匝道拥堵现象最终结束。

因为假设邻近平行道路不发生拥堵，此时系统中的总延误 W_{base} 仅包括出口匝道、地面支线和高速主线

$$W_{base} = n_r d_r + n_f d_f + n_s d_s$$

7. 当加入绕行控制时，理论上出口匝道和地面支线均不会产生排队，此时系统延误主要体现在以下两方面：
 - 原本地面支路上的车辆绕行至邻近平行道路上花费的时间 t_{ff} ；
 - 邻近平行道路因为绕行流量所造成的拥堵。

记 $[t_i, t_e]$ 时段内平行道路的平均延误为 d_p^c ，绕行流量为 n_s^c ，则系统总延误 $W_{control}$

$$W_{control} = n_s^c(t_{ff} + d_p^c) + n_p d_p^c$$

8. 当 $W_{control} < W_{base}$ 时控制策略即可效益，即当绕行车辆数 n_s^c 小于下式规定的上界时控制策略下的系统效益优于无控制的情况

$$n_s^c < (n_r d_r + n_s d_s + n_f d_f - n_p d_p^c) / (t_{ff} + d_p^c)$$

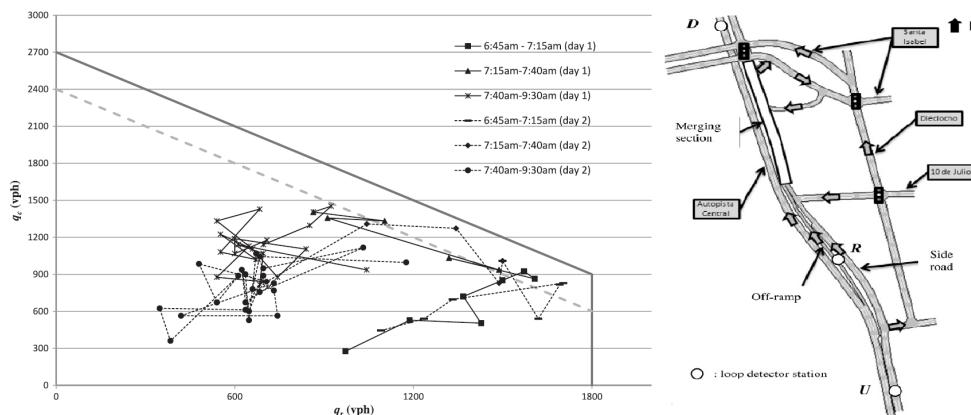
9. 考虑更一般的情况：假设出口匝道需求量本身可能超过通行能力，绕行引导也无法完全消除匝道和高速主线的延误；同时考虑平行街道本身也可能存在拥挤。则以上三式改写为

$$\begin{cases} W_{base} = n_r d_r + n_f d_f + n_s d_s + n_p d_p \\ W_{control} = n_r d_r^c + n_f d_f^c + n_s^c(t_{ff} + d_p^c) + n_p d_p^c \end{cases} \implies n_s^c < \frac{n_f \Delta d_f + n_r \Delta d_r + n_s d_s + n_p \Delta d_p}{t_{ff} + d_p^c}$$

7.1.4 Field implementation

1. 研究的出口匝道接入场景如图7.2（右）所示。其中 R 为分别位于出口匝道上的探测器，探测器 U、D 分别位于出口匝道上游和下游。出口匝道以汇入的形式与地面支线相接，汇入点下游为有信号交叉口；
2. 出口匝道、地面支线、汇入路段的车道数分别为 1、2、3 满足车道数平衡。匝道和地面支线的理论通行能力分别为 1800 pcv/h、3600 pcv/h。因为下游有信号交叉口的绿信比为 0.5，汇入点的理论通行能力为 $0.5 \times (1800 + 3600) = 2700$ pcv/h。因为 $Q_M < Q_s$ ，该场景的 $\bar{q}_r - \bar{q}_s$ 关系图（图7.2（左））与图7.1略有不同；
3. 图7.2（左）为观测到的两天的早高峰时段内的 $q_r - q_s$ 散点图。早高峰时段被分成三段，分别为 6:45-7:15、7:15-7:40 以及 7:40-9:30：
 - 在 6:45-7:15 时段，匝道和地面支线流量总体处于较低水平，两者同步上升直至接近汇入点通行能力；

图 7.2 出口匝道接入场景 (右) 与实测匝道-地面支线流量数据 (左)。



- 在 7:15-7:40 时段，因为汇入点达到通行能力，匝道处最先出现拥挤 (q_r 降低)，但地面支线流量依然保持上升趋势。这段时间内 $q_r + q_s$ 大体为定值，但普遍低于理论的节点通行能力 (2700 pcv/h)，此时可通过回归得到节点的实际通行能力为 2400 pcv/h 左右；
- 在 7:40-9:30 时段，因为地面支线需求量持续增长，地面支线发生拥挤， q_r, q_s 同步下降，而且大体成正比，与汇入模型基本一致，反映为节点处出行交替路权现象。需要注意的是此时汇入节点处发生了通行能力折减（降至 1800 pcv/h 甚至以下）。考虑到节点下游为信号控制交叉口，地面支线与出口匝道的汇入流需要在到达交叉口前根据转向计划进入相应的车道，因此产生强烈的交织现象，这也是造成通行能力折减的主要原因。

4. 因为缺少合适的设备使得支线车辆按计算的比例被引导绕行，实地实验中由交警人员代为管理。具体做法是在有限时段内（从上午 6:00 开始）完全关闭地面支路上游车道（相当于 $\phi = 1$ ），而后再重新开放。

7.1.5 英汉互译

English	Chinese	English	Chinese	English	Chinese
scant advisability	缺乏的合理 (n)	rush hour	高峰时段	side road	旁路、支线

7.2 Design and Implementation of Integrated Network Management Methodology in a Regional Network (TRR, 2015)

ABSTRACT

— □ ×

Praktijkproef Amsterdam (Field Operational Test Integrated Network Management Amsterdam) 是一项研究阿姆斯特丹区域交通网络协同控制的项目。本文主要关注基于主从结构 (master-slave structure) 和启发式匝道协同控制算法 (heuristic ramp-metering coordination, HERO) 的控制器设计。控制的目标是将城市干道作为高速的缓冲区，一方面改善高速交通状况，另一方面也解决城市干道本身的问题。最后本文针对不同的控制器设计，分析控制器的动态特性与控制参数的关系。

7.2.1 Introduction

- 整个研究项目分为两个阶段 (本文属于项目的第一阶段):
 - 阶段一聚焦于一条城市高速 (A10 West, 包括其进出口匝道) 和一条与之相连的城市干道 (s102)。高速的所有进口匝道均配有匝道控制设施，城市干道的所有交叉口均为有控制交叉口；
 - 阶段二将研究范围扩大至阿姆斯特丹路网的其他区域，并引入其它控制手段，包括车车之间的控制。
- 项目的控制措施设计遵循如下原则:
 - 尽可能避免通行能力折减；
 - 网络中的交通流除非必要不应该受到阻碍，避免溢流和死锁的发生 (可通过设置交通缓冲区实现)；

- 实现网络通行能力利用效率的最大化，因此缓冲区的设置取决于网络的服务水平，服务水平越差需要的缓冲空间越大；
 - 瓶颈需要在其自身表现的层次上加以解决，即局部问题应该尽可能在局部范围内得到解决。如果缓冲空间耗尽，需要在网络的其它地区寻找新的缓冲空间。
3. 网络协调控制并非新领域，但实际应用的例子较少。大多数已有研究基于最优控制 (*optimal control*) 和模型预测控制 (*model-predictive control, MPC*)，尽管这些方法在灵活性和鲁棒性上具有优势，但在实际应用时则存在计算复杂性和解释性低的问题，因此有必要考虑新的具有通用性的方法；
 4. 启发式匝道协同算法 (*heuristic ramp-metering coordination, HERO*) 是一种非常成功的多匝道协同控制算法，已得到广泛的实际应用。在实际应用中，多匝道协同控制的有效性取决于进口匝道长度和匝道处瓶颈处交通流的关系；
 5. 在项目的研究场景中，高速路的匝道长度普遍较短，意味着匝道控制只能持续一小段时间（时间过长会导致匝道排队影响地面主线交通），此外通过匝道进入瓶颈的流量相对较低，仅依靠匝道控制所能起到的效果较为有限；
 6. 针对上述问题，项目将进口匝道和对瓶颈交通贡献较大的地面干道共同作为缓冲空间，同时避免过多影响其它方向的交通。

7.2.2 System design and functional architecture

1. 项目所涉及的模块大体分为监测诊断模块 (*logical monitoring units, LMU*) 和控制模块 (*logical control units, LCU*) 两类，控制任务分为单点控制、干道控制、子网控制²和网络控制；
2. **LMU 模块实现的功能包括交通状态评估预测、瓶颈位置识别、剩余缓冲空间计算和网络服务水平确定，作为 LCU 模块的输入：**
 - 高速公路状态基于探测线圈提供的流量、速度、密度信息确定，为此基于冲击波和运动波理论 (*shock wave and kinematic wave theory*) 修正探测线圈提供的异质交通数据；
 - 基于当前高速公路状态和历史数据预测高速公路瓶颈位置，具体地，算法预测三分钟后有较大概率发生通行能力折减的位置，并基于改进卡尔曼滤波法预测瓶颈处的临界密度和通行能力；
 - 地面干道的交通状态选择排队长度作为指标。排队长度检测器预测进口匝道、出口匝道和交叉口处的排队长度。基于排队长度识别瓶颈位置。地面干道的瓶颈指排队过长导致回流的位置或某个交叉口严重过饱和的方向。剩余的缓冲空间同样基于排队长度确定，当前排队长度与允许排队长度的差值即为剩余缓冲空间；
 - 网络服务水平由网络车辆数及其空间分布情况共同确定。
3. **LCU 模块为单点控制器，包括匝道控制器和交叉口控制器，可独立自适应控制也可直接接收上层控制器 (*supervisor*) 的指令。** 上层控制器包括子网控制器 (*subnetwork supervisor*)、城市干道控制器 (*urban arterial supervisor*)、连接控制器 (*connection supervisor*) 和高速干道控制器 (*freeway arterial supervisor*)：
 - 子网控制器是协同控制的中枢。基于服务水平和当前交通状况计算目前缓冲区可承载的交通量；
 - 城市干道控制器负责城市干道各信号控制交叉口的协调控制。控制器基于子网控制器提供的干道缓冲区配置和瓶颈检测器提供的干道瓶颈位置均匀分布排队；
 - 连接控制器实现匝道控制器和其上游的信号交叉口控制器的协同。控制器基于子网控制器计算的上游交叉口缓冲区配置调整交叉口信号配时缓解匝道流入压力；
 - 高速干道控制器实现高速沿线各匝道控制器的协同。

7.2.3 Coordination of controlled intersections for ramp-metering support

1. 主从控制结构 (*master-slave concept*) 是一种经典的协同控制结构。以多匝道协同控制为例，主控制器往往为距瓶颈最近的上游进口匝道控制器，通过匝道控制直接缓解瓶颈处的排队，从控制器则是其它进口匝道控制器，其目标是通过匝道控制缓解主控制器的控制压力；

²子网定义为由一条高速干道和与之相连的地面干道组成的网络

2. 项目同样基于主从控制结构实现网络协同控制，进一步地将从控制器的范围扩展至信号交叉口控制器；
3. 首先基于与瓶颈的相关程度和控制效益确定主控制器。主控制器可以为进口匝道控制器或进口匝道-交叉口联合控制器，通过减少流入高速的流量缓解高速瓶颈；也可以为出口匝道出口处信号交叉口控制器，通过增加出口匝道流出率避免出口匝道溢流；还可以是一般信号交叉口控制器以缓解信号交叉口处的瓶颈；
4. 从控制器可以是上游进口匝道控制器，也可以是其它信号交叉口控制器；
5. 考虑一个由匝道控制器和上游交叉口控制器组成的主从控制结构。假设匝道控制基于经典的 ALINEA 反馈控制算法，算法通过调节匝道调节率 $q_r(t)$ 控制瓶颈处占用率位于目标值附近。此时进口匝道即为缓冲区，记匝道所能承载的车辆数为 L_r ，当前的车辆数为 $w_r(t)$ ，则剩余的缓冲空间 $\theta_r(t)$

$$\theta_r(t) = \frac{L_r - w_r(t)}{L_r}$$

记 J_r 为与匝道强相关的其它缓冲区的集合，记集合中每一缓冲区 j 其可承载的车辆数为 L_j 、剩余的缓冲空间为 $\theta_j(t)$ 。此时从控制器的目标为调整交叉口调节率使得 $\theta_j(t) = \theta_r(t)$ 。为此可基于简单的反馈控制算法得到缓冲区调节率 $u_j(t)$ 的更新公式，其中 $K_1, K_2 \geq 0$ 为控制增益 (control gain)：

$$u_j(t+1) = u_j(t) + K_1 e_j(t) + K_2 \Delta e_j(t) \quad e_j(t) = \theta_r(t) - \theta_j(t) \quad \Delta e_j(t) = e_j(t) - e_j(t-1)$$

对于每一缓冲区基于上式独立计算其调节率；

6. 上述匝道-交叉口主从控制与著名的多匝道协同控制 HERO 算法非常接近，不同之处在于：
 - 缓冲区中的流量只有一部分会经过匝道，因此需要严格选取缓冲区，避免过多影响其它方向的交通；
 - 尽管每一缓冲区的渗透率是独立更新的，但不代表缓冲区直接是独立的，上游缓冲区的输出即为下游缓冲区的输入。

7.2.4 Control approach analysis and tuning

1. 本节提出了控制器理论分析方法并选择最优控制增益。研究一个进口匝道与上游缓冲区组成的系统，考虑不同的缓冲区流量输出对匝道排队的影响（直接影响或延迟影响，较大影响或有限影响），并设置不同复杂度的三种情况；
2. 首先考虑一个进口匝道与上游一个缓冲区组成的系统，则进口匝道缓冲空间 $s_r(t)$ 的更新公式如下，其中 $q_r(t)$ 为匝道控制率 (veh/h)、 $d_r(t)$ 为未受控制的匝道进口流量、 Δt 为时间不长、 $u(t)$ 为上游缓冲区的输出流、 T 为从缓冲区到匝道行程时间和排队延迟导致的总延迟、 α 用于反映缓冲区输出流对进口匝道的需求的贡献（并非所有通过缓冲区的流量都要上匝道）

$$s_r(t+1) = s_r(t) + [q_r(t) - d_r(t) - \alpha \cdot u(t-T)] \Delta t$$

同样地可以计算上游缓冲区的缓冲空间 $s_b(t)$

$$s_b(t+1) = s_b(t) + [u(t) - d_b(t)] \Delta t$$

至此可以构造系统的状态向量 (state vector) $x(t)$ ，定义为为预测 $t+1$ 时刻状态信息所需要的全部系统信息，考虑到系统存在延迟 T ，因此 $x(t) = \{s_r(t), s_b(t), u(t), s_r(t-1), s_b(t-1), u(t-1), \dots, u(t-T)\}$ ，从而整理得系统的离散时间状态方程 (discrete time state-space formation)

$$x(t+1) = Ax(t) + b(t) \quad A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & -\alpha \Delta t \\ 0 & 1 & \Delta t & 0 & 0 & 0 & \cdots & 0 \\ \frac{K_1+K_2}{L_r} & -\frac{K_1+K_2}{L_b} & 1 & -\frac{K_2}{L_r} & \frac{K_2}{L_b} & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \quad b(t) = \begin{bmatrix} q_r(t) - d_r(t) \\ -d_b(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Delta t$$

记矩阵 V 为系统矩阵 A 的特征列向量组成的矩阵，则可对系统状态方程做线性变化将其转化为对角规范型（或约当规范型）。对于特征向量 $z(t)$ 中的任意元素 z_i ，若其对应于系统矩阵 $V^{-1}AV$ 的特征值（也是系统矩阵 A 的特征值，线性变化不改变特征值） $|\lambda_i| < 1$ ，则状态 z_i 是稳定的。

$$z(t+1) = V^{-1}AVz(t) + V^{-1}b(t) \quad z(t) = V^{-1}x(t)$$

因为控制系统的目地为使得误差 $e(t) = x_1(t)/L_r - x_2(t)/L_b$ 趋于零，因此在分析系统稳定性时只需讨论状态向量中与 $e(t)$ 相关的元素 z_i 对应的特征值 λ_i 。令向量 $\kappa(t) = \{1/L_r, -1/L_b, 0, \dots, 0\}$ ，则有 $e(t) = \kappa(t)x(t) = \kappa(t)Vz(t)$ ，则对应于 $\{\kappa(t)V\}_i \neq 0$ 的 i 即为需要关注的元素。综上，系统的收敛特性 γ 可量化为

$$\gamma = \max \left\{ |\lambda_i| \mid \{\kappa(t)V\}_i \neq 0 \right\}$$

- 给定超参 $\alpha, T, \Delta t, L_r, L_b$ 即可得到系统收敛性 γ 与系统增益 K_1, K_2 的关系，并可求解得最优 K_1, K_2 取值；
3. 考虑一个进口匝道与上游两个串联缓冲区组成的系统，记距进口匝道最近的缓冲区为一级缓冲区、再上游的缓冲区为二级缓冲区，一级缓冲区的需求量即为二级缓冲区的控制输出量，得到两级缓冲区的缓冲空间 $s_b^1(t), s_b^2(t)$ 更新公式分别如下，控制器的目标为使得误差 $e_1(t), e_2(t)$ 趋近于零

$$s_b^1(t+1) = s_b^1(t) + [u_1(t) - u_2(t-T_1)]\Delta t \quad s_b^2(t+1) = s_b^2(t) + [u_2(t) - d_b^2(t-T_2)]\Delta t$$

4. 考虑一个进口匝道与上游两个平行缓冲区组成的系统，得到进口匝道的缓冲空间 $s_r(t)$ 更新公式分别如下，控制器的目标为使得误差 $e_1(t), e_2(t)$ 趋近于零

$$s_r(t+1) = s_r(t) + [q_r(t) - d_r(t) - \alpha_1 u_1(t-T_1) - \alpha_2 u_2(t-T_2)]\Delta t$$

7.2.5 英汉互译

English	Chinese	English	Chinese	English	Chinese
hamper	阻碍 (v)	blockade	封锁 (n,v)	diagnose	诊断 (v)
to this end	为此	municipality	自治区	for instance	比如
instance	例子、举例 (v)	at hand	在附近	mimic	模仿 (v)
eigenvector	特征向量	eigenvalue	特征值	mockup	实体模型

7.3 Coordinated Ramp Metering Based on On-Ramp Saturation Time Synchronization (TRR, 2015)

ABSTRACT

— □ ×

本文基于协同多匝道空间耗尽所需的时间提出多匝道协同控制方法。协同瓶颈上游的多个进口匝道控制器可降低下游瓶颈的交通压力，而匝道控制所能持续的时间取决于该匝道被完全排满的时间和多个匝道先后被排满的顺序。针对多匝道协同控制存在多种协同方式，本文提出的协同策略旨在确保更上游的进口匝道总比相对下游的进口匝道更早地耗尽其存储空间（即最大化下游匝道的控制时间）。得到的协同控制策略将与另外三种情况比较：无控制情况、单点控制情况、和旨在均衡各匝道占用率的多匝道协同控制的情况。比较结果显示两种协同控制策略更好地提升网络运行水平，且本文提出的协同控制策略可以更好地避免交通崩溃。图

7.3.1 Background

- 匝道控制可分为单点控制和多匝道协同控制。单点匝道控制又可分为固定时间控制 (fixed-time / pretimed control) 和自适应控制 (adaptive / traffic responsive control)。前者基于历史数据确定调节率 (metering rate)，后者则基于当前匝道附近的交通状况确定。单点自适应匝道控制器主要基于前馈策略

(feed-forward strategy) 或反馈策略 (feedback strategy)³, 前者如需求-通行能力控制 (demand-capacity control)、占用率控制 (occupancy control), 后者如著名的 ALINEA 控制;

2. 因为要求匝道不发生溢流, 匝道控制的效果受限于匝道自身的容量, 多匝道协同控制可充分利用路网容量, 起到更好的效果;
3. 已有的启发式多匝道协同控制算法包括 bottleneck、helper、SWARM、zone 和 HERO 等等:
 - helper 算法中每个单点匝道控制器预设有 6 个等级的调节率, 若某个匝道的排队长度增加, 则将其调节率增大一个等级 (即放行更多的车辆以消散排队), 同时将其上游匝道的调节率下调一个等级, 如果该匝道排队仍然严重, 则寻找再上游匝道继续下调调节率直至所有上游匝道均参与协助。helper 算法与本研究所提算法类似, 均是通过上游匝道协助控制避免下游匝道过早饱和, 然而匝道控制位置上移不可避免增大下游瓶颈崩溃的风险, 而且通过上游匝道的车也有更大概率不通过下游瓶颈 (可能通过中间的出口匝道离开);
 - bottleneck 和 zone 算法则是预先对沿线各进口匝道分配权重, 当下游出现瓶颈时上游各匝道基于预设的权重分配匝道控制任务, 但在计算权重时并未精确考虑进口匝道的储存空间;
 - SWARM 算法确定多个探测器得到的密度较期望密度的超出程度, 基于需求和进口匝道储存空间确定各匝道权重系数, 进而按权重分配匝道控制任务。与 bottleneck 和 zone 算法相比 SWARM 算法中权重系数的确定考虑了匝道的储存空间, 但并不能证明权重系数的取值使得匝道储存空间得到最大程度的利用;
 - HERO 算法同样精确地考虑了匝道的储存空间, 算法的协同思路是使得上下游进口匝道保持相同的饱和程度, 即多匝道最终将同时达到饱和, 注意到最下游瓶颈消散最有利的为距瓶颈最近的匝道, 该算法仍然不能最大化匝道控制的时间。
4. 除了以上所述的启发式算法, 也有一些研究基于最优控制和模型预测控制从理论上给出多匝道协同控制算法, 但算法复杂性限制了其大规模应用;
5. 在多匝道协同控制策略中, 下游匝道距瓶颈最近, 其流量控制对瓶颈缓解的效果最好, 而上游匝道控制的目的在于分担下游匝道的控制压力, 尽量避免下游匝道因为饱和而停止匝道控制。因此本文提出的启发式协同控制策略旨在保证上游匝道先于下游匝道饱和, 即最大化下游匝道的控制时间。

7.3.2 Control approach

1. 类似于 HERO 算法, 本研究提出的协同策略基于主从控制结构, 称最接近下有瓶颈的进口匝道控制器为主控制器, 上游的其它匝道控制器为从控制器;
2. 从控制器激活后因为进口流量降低, 会在主线上形成一段密度相对较低的区域, 称为“间隙”(gap)。上游从控制器激活形成的间隙随交通流向下游传播, 且速度为自由流速度 (假设从控制器所在位置不受瓶颈影响, 交通流为自由流)。当间隙到达主控制器时, 主控制器的控制压力可暂时降低, 可提高调节率消散匝道队列, 因此当间隙到达时主控制器达到饱和即可完全利用间隙, 所以控制的目标即为使得从控制器的期望饱和时间与主控制器的期望饱和时间减去两者间的自由行程时间保持同步;
3. 从控制器的目标饱和时间可基于其储存空间和需求量确定, 并满足最大最小调节率和单点匝道控制任务的约束。另外当主控制器饱和而从控制器仍有储存空间时从控制器应该以最小调节率继续协助主控制器;
4. 如果路段存在多处进口匝道则可能形成多处瓶颈, 此时需要设置多个主控制器并为其合理分配从控制器。本文仅考虑只有一处瓶颈的情况;
5. 记瓶颈上游一共有 r_{\max} 个可协同控制的进口匝道, 对其中最下游匝道编号为 1、最上游匝道编号为 r_{\max} 。对于控制周期 k_c , 若匝道 r 的车辆数为 $w_r(k_c)$, 则其剩余储存空间 $s_r(k_c)$ 有绝对值或相对值两种表示方法

$$s_r(k_c) = s_r^{\max} - w_r(k_c) \quad s_r^{rel}(k_c) = \frac{s_r(k_c)}{s_r^{\max}}$$

³前馈控制系统属于开环系统而反馈控制系统属于闭环系统。前馈控制系统的控制依据是扰动, 系统测量可能影响被控变量的那些过程变量, 当过程变量出现扰动时 (此时被控变量可能尚未发生变化) 发出调节信号, 调节过程更快, 但因为系统不测量被控变量, 因此无法保证调节的效果, 迭代学习控制 (iterative learning control, ILC) 即属于前馈控制。反馈控制系统的控制依据是偏差, 系统直接测量被控变量, 当被控变量与目标值发生偏差时才进行调节, 因此调节过程较慢, 但调节效果更稳定, 比例积分微分控制 (proportional integral derivative, PID) 即属于反馈控制。以打球为例, 根据对方在出球前的动作进行预判即属于前馈控制, 而在对方出球后再根据球的轨迹运动既属于反馈控制。

6. 集合 $A(k_c)$ 为包含 k_c 周期所有候选主控制器的集合。当集合中至少一个进口匝道的相对储存空间低于预定阈值时激活协同控制，且其中最靠近下游的进口匝道为该时刻的主控制器（记编号为 $m(k_c)$ ），在其上游的其它匝道则落入从控制器集合 $S(k_c)$ ，实验时设 $s_{threshold}^{rel} = 0.95$

$$m(k_c) = \min\{r | s_r^{rel}(k_c) < s_{threshold}^{rel}, r \in A(k_c)\} \quad S(k_c) = \{m(k_c) + 1, \dots, r_{max}\}$$

7. 主控制器基于任意定点匝道控制算法（如 ALINEA）计算调节率 $q_r^{loc}(k_c)$ ，其期望饱和时间 $T_m(k_c)$ 由当前剩余储存空间 $s_m(k_c)$ 与剩余储存空间的变化率 $\Delta s_m(k_c)$ 确定。 $T_m(k_c)$ 为正值意味着主控器需要从控制器减少调节率缓解主控制器压力，为负值意味着从控制器被允许调大调节率

$$T_m(k_c) = \frac{s_m(k_c)}{\Delta s_m(k_c)} \quad \Delta s_m(k_c) = s_m(k_c) - s_m(k_c - 1)$$

上游匝道协同控制形成的间隙有助于缓解主控制器压力，延长其控制时间，但按上式计算的 $T_m(k_c)$ 并未考虑这一影响，造成 $T_m(k_c)$ 被低估，并增大上游从控制器的控制压力。考虑多个从控制器协同控制形成的多个空隙的影响，修正上式为

$$T_m(k_c) = \frac{s_m(k_c) + s_m^{gaps}(k_c)}{\Delta s_m(k_c)} \quad s_m^{gaps}(k_c) = \sum_{s \in S(k_c - 1)} s_s^{gap}(k_c) = \sum_{s \in S(k_c - 1)} T_c \cdot [\min\{d_s(k_c - 1), q_s^{loc}(k_c - 1)\} - q_s^{crd}(k_c - 1)]$$

上式中 T_c 为一个控制周期的长度， $d_s(k_c)$, $q_s^{loc}(k_c)$, $q_s^{crd}(k_c)$ 的定义见下；

8. 从控制器的期望饱和时间 $T_s(k_c)$ 基于主控制器期望饱和时间 $T_m(k_c)$ 与从控制器至主控制器的自由行程时间 τ_{sm} 确定

$$T_s(k_c) = \begin{cases} \max\{c, T_m(k_c) - \tau_{sm}\} & T_m(k_c) \geq 0 \\ T_m(k_c) & T_m(k_c) < 0 \end{cases}$$

当主控制器接近饱和而从控制器与主控制器距离较远时，为避免从控制器期望饱和时间为负值，设置最小期望饱和时间约束 $c > 0$ （如 $c = 0.01$ ），意味着加强从控制器的控制效果缓解下游主控制器的压力；

9. 基于从控制器需求量 $d_s(k_c)$ 、剩余储存空间 $s_s(k_c)$ 和期望饱和时间 $T_s(k_c)$ 得到其协同控制调节率 $q_s^{crd}(k_c)$

$$q_s^{crd}(k_c) = \begin{cases} d_s(k_c) - \frac{s_s(k_c)}{T_s(k_c)} & s_m(k_c) \geq 0, s_m(k_c - 1) > 0 \\ q_s^{\min} & s_m(k_c) = s_m(k_c - 1) = 0 \end{cases}$$

上式中 $s_m(k_c) = s_m(k_c - 1) = 0$ 意味着主控制器在多个控制区间内均处于饱和状态，此时上游从控制器以最小调节率放行匝道车辆，最大限度避免下游瓶颈处交通崩溃；

10. 对于任意匝道 r ，其可能同时有单点控制任务和协同控制任务，则选择其中最小的调节率 $q_r(k_c)$ ，并考虑最大调节率和最小调节率的约束得到实际调节率 $\hat{q}_r(k_c)$

$$\hat{q}_r(k_c) = \min\{q_r^{\max}, \max\{q_r^{\min}, q_r(k_c)\}\} \quad q_r(k_c) = \min\{q_r^{loc}(k_c), q_r^{crd}(k_c)\}$$

7.3.3 Test case

1. 基于仿真实验研究多匝道协同控制策略的效果。其中设置无控制和单点控制作为对照以确定协同控制的优势，又设置简化 HERO 算法作为对照以确定上游匝道饱和时间优化的优势。最后讨论当对主控制器饱和时间和从控制器需求量检测出现误差时对控制效果的影响；
2. 仿真路网包括两个进口匝道，不考虑出口匝道分流的情况以保证所有载入流量最终通向下游瓶颈；
3. 系统评价指标由网络下游输出流量确定，输出流量越高说明更好地抑制通行能力崩溃现象（注意到上游匝道控制本身也可能降低下游输出流量）。具体地，以网络中所有车辆行程时间之和 (total time spend, TTS) J_{TTS} 作为系统评价指标，其中 k 为时间步序号、 T 为时间步长度、 M 为路网中各连边的集合、 C_m 为连边 m 中元胞的集合、 $\rho_{m,c}, \lambda_{m,c}$ 分别为元胞内的密度和长度

$$J_{TTS} = T \sum_{k=1}^K \sum_{m \in M} \sum_{c \in C_m} \rho_{m,c}(k) \lambda_{m,c}$$

总行程时间减去自由流情况下的行程时间即可得到总延误，另外还可以比较拥堵的形成时间 t_0^{cong} 、结束时间 t_1^{cong} 、持续时间 $T^{cong} = t_1^{cong} - t_0^{cong}$ 和最大排队长度 W_{max}^{cong} ；

4. 仿真基于 Fastlane 模型实现。Fastlane 模型为一类基于多类别元胞传输模型的宏观交通流模型，可较好地模拟网络层面的一些宏观交通现象，包括拥挤的生成和消散、排队的阻截效应 (blocking back effects) 和通行能力崩溃。其中通行能力崩溃现象基于 Godunov 规则 (Godunov scheme) 实现，当一个元胞拥挤时，其向输出下游元胞输出的流量将减少特定的比例 (如 15%)；
5. 进一步地介绍对照实验采取的单点控制算法和另一种协同控制算法：

- 单点控制算法采用需求量-通行能力算法 (demand-capacity algorithm)，基于匝道上游的主线流率 $q_r^{up}(k_c)$ 与接入点下游通行能力估算值 C_r 确定调节率 $q_r^{loc}(k_c)$ 。下式中 $v_r^{dn}(k_c)$ 为接入点下游速度、 v_r^{cong} 为以最小调节率进行调节的速度阈值、 $w_r(k_c)$ 为匝道排队车辆数、 w_r^{\max} 为匝道最大排队车辆数

$$q_r^{loc}(k_c) = \begin{cases} C_r - q_r^{up}(k_c) & w_r(k_c) < w_r^{\max}, v_r^{dn}(k_c) \geq v_r^{cong} \\ q_r^{\min} & w_r(k_c) < w_r^{\max}, v_r^{dn}(k_c) < v_r^{cong} \\ q_r^{\max} & w_r(k_c) = w_r^{\max} \end{cases}$$

需要说明的是，上述单点控制算法不仅用于对照实验，也用于协同控制中的主控制器设计；

- 替代协同控制算法采用简化 HERO 算法，算法的目标是使得多匝道同时达到饱和。算法基于反馈控制使得所有从控制器的相对剩余储存空间 $s_s^{rel}(k_c)$ 与主控制器的相对储存空间 $s_m^{rel}(k_c)$ 保持一致。从控制器 s 的调节率 $q_s^{crd}(k_c)$ 的更新公式如下，其中 $\alpha_1, \alpha_2 > 0$ 为反馈增益 (或控制增益)，在实验中取 $\alpha_1 = 200, \alpha_2 = 1250$

$$q_s^{crd}(k_c + 1) = q_s^{crd}(k_c) + \alpha_1 e_s(k_c) + \alpha_2 \Delta e_s(k_c) \quad e_s(k_c) = s_m^{rel}(k_c) - s_s^{rel}(k_c) \quad \Delta e_s(k_c) = e_s(k_c) - e_s(k_c - 1)$$

7.3.4 Results

1. 首先考虑主线情况：
 - 与无控制情况相比，加入单点匝道控制后交通崩溃的开始时间变晚、持续时间和最大排队长度缩短；
 - 进一步引入协同控制后，原有的严重交通瓶颈被分解为位于两匝道下游的两处瓶颈，主控制器下游的瓶颈较严重，从控制器下游的瓶颈较轻，且交通崩溃开始时间、持续时间和最大排队长度等指标进一步得到改善。对比两种协同控制策略，本文所提策略优于对比策略，但因为本文所提策略要求从控制器先于主控制器达到饱和，从控制器的调节压力更大，因此从控制器下游瓶颈早于除控制器下游瓶颈出现。
2. 考虑匝道利用率：
 - 在仅进行单点控制的情况下，大部分调节任务均由距瓶颈最近的匝道承担，只有在较短的时间内上游更远处的匝道才激活匝道控制；
 - 基于剩余储存空间一致的反馈式协同控制使得两匝道的剩余储存空间基本保持一致，且主控制器所在匝道更早达到饱和；
 - 在确保从控制器先于主控制器饱和的协同控制下，在达到饱和前从控制器的剩余储存空间低于主控制器的剩余储存空间，也更早达到饱和。
3. 考虑网络的总体性能。协同控制情况下的系统总延误显著优于无控制和单点控制的情况，且本文所提策略稍稍优于基于剩余储存空间一致的协同策略；
4. 最后考虑主控制器饱和时间和从控制器需求量检测误差的影响——估值低于实际值导致主控制器过早饱和，而估值高于实际值导致从控制器过早饱和。

7.3.5 英汉互译

English	Chinese	English	Chinese	English	Chinese
terminology	术语	as long as	只要	distinction	差别、优秀 (n)
incentive	激励 (n)	intuitive	直观的	relevance	重要性、意义

7.4 A varying parameter multi-class second-order macroscopic traffic flow model for coordinated ramp metering with global and local environmental objectives (TRC, 2021)

ABSTRACT

- □ ×

本文基于一耦合了污染物排放模型的多类别二阶宏观交通流模型提出多匝道协同控制策略。本文提出的多类别交通模型是 METANET 模型^a的改进版本，而污染物排放模型则采用 COPERT 模型^b。控制策略要求具有合理性、高效性和环境可持续性，并基于最优控制实现，且为求解大规模静态优化问题引入差分进化算法 (differential evolution algorithm)。控制策略除考虑网络层面的整体环境效益外还考虑局部的环境需求，为此引入两类环境约束。

^aTraffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET

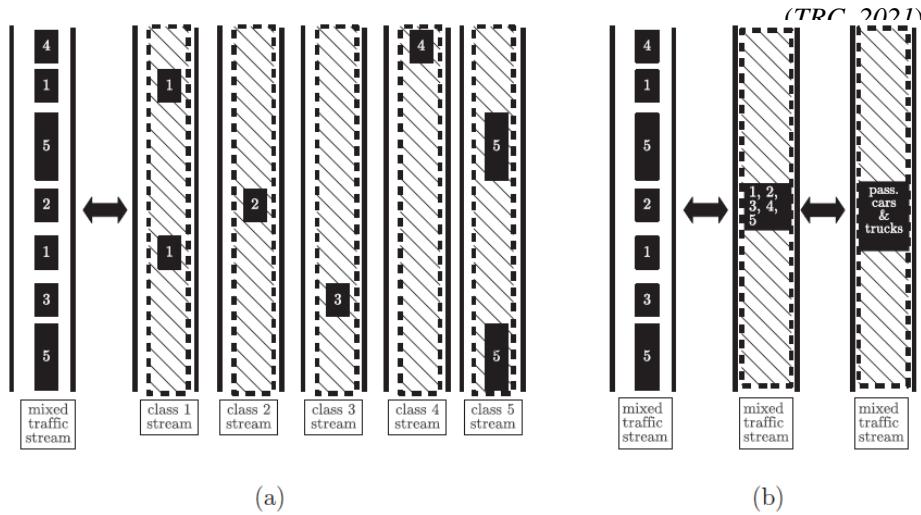
^bRoad transport emission chapter of the EMEP/CORINAIR Emission Inventory Guidebook

7.4.1 Introduction and background

1. 考虑一个由多种车辆类型组成的交通流，单类别模型 (**single-class macroscopic models**) 将其视为宏观连续介质，而多类别模型 (**multi-class macroscopic models**) 则将其理解为由多股单一类别车流组成的混合流，对每股单一流分别建立动力学方程，并考虑不同单一流之间的交互；
2. 多股单一流之间的交互机制取决于交通流基本假设和交通流模型的阶数：
 - (a) 一阶连续交通流模型又称 LWR 模型，模型的守恒方程为关于流率和密度的偏微分方程，并进一步地引入基本图模型，假设平均速度和密度存在平衡关系。若基于一阶模型，则为多类别模型中的每一单类别模型单独建立守恒方程，而所有单类别模型共用同一基本图方程。基本图模型反映混合流处于平衡态的情况，表现不同类型车辆间的影响，不同类型车辆通过标准小车换算系数 (**passenger car equivalence, PCE**) 同质化。此时无法很好地描述非平衡状态下的情况，包括时走时停波、磁滞现象等；
 - (b) 高阶连续模型则舍去了平衡态速度-密度基本图关系，取而代之为包含速度微分项的新的微分方程，可更好地描述非平衡状态下的交通现象。若基于高阶模型，则为多类别模型中的每一单类别模型单独建立守恒方程和速度微分方程。
3. 多类别模型在保持对交通流宏观描述的基础上可反映出诸如拥挤形成、激波传播、通行能力折减、交通流同步化 (flow synchronization)、磁滞 (**hysteresis**)、排队消散、基本图右支散布 (fundamental diagram scatter) 等重要交通流现象；
4. 基于车辆类别的交通模型可与排放模型耦合以评价环境效益。常用于与宏观交通模型耦合的排放模型包括 COPERT、VERSIT+、VT-micro/macro 等等：
 - COPERT 模型是一系列污染物的平均速度排放模型，基于不同发动机标准和燃料类型车辆的现场排放数据得到；
 - VERSIT+ 为一微观模型，基于车辆的加速情况确定污染物排放；
 - VT-micro 模型为一微观油耗模型，与宏观交通流模型 METANET 耦合后得到 VT-macro 模型。

相比于单类别模型，多类别模型因其研究对象为多股单一类别车流组成的混合流，因此更适合与排放模型耦合研究交通流的环境效益；
5. 基于耦合环境模型的多类别交通模型是多目标交通管控研究的重要形式。本文基于最优控制设计多匝道协同控制系统，引入模型预测控制 (model predictive control, MPC) 框架求解最优控制问题，这也是最高效的路网控制设计方法之一；
6. 若对混合交通流中每一类型的车辆单独建立单类别模型，可能存在为多种动力特性相同的车辆重复建模的情况。有必要按动力特性将车辆分组，同组的车流共享相同的动力学模型，并对不同类型车辆设置不

图 7.3 两种多类别宏观连续交通模型构建方法。图中混合交通流由五类车辆组成，前四种为小汽车，第五种为卡车。**(a)** 为经典的多类别模型建模方法——对每一类车流单独建立单类别模型；**(b)** 为本文的建模方法——五类车辆按动力性能分成小汽车和卡车两组，对两组汽车单独建立单类别模型，并考虑同组车流内不同类型车辆的其它参数差异。



同参数以体现同组车流中不同车辆类型的环境效益差异：

- 分析环境效益时，一方面应考虑全局环境效益；另一方面注意到部分地区（如学校、医院等，Special Interest Areas (SIA)）部分时段可能对环境有更高的要求，也要考虑局部环境效益。

7.4.2 Traffic flow model development

single-class motorway network model

- 将研究路网视为包括 N 个节点 \mathcal{L} 个连边的有向图 $(\mathcal{N}, \mathcal{L})$ 。将连边集合 \mathcal{L} 分解为三个子集 $\mathcal{L}_M, \mathcal{L}_O, \mathcal{L}_D$ ，分别表示研究路网内的道路集合、接入研究路网的道路集合（起点属于外部路网，终点属于研究路网）和离开研究路网的道路集合（起点属于研究路网，终点属于外部路网）。对于一般道路 $\mu \in \mathcal{L}_M$ ，记其车道数为 l_μ ，将其等分为 S_μ 段，每段长度均等为 Δ_μ ；
- 路段 (μ, i) 的交通状况由车辆密度 $\rho_{\mu,i}$ 、空间平均速度 $v_{\mu,i}$ 和流量 $q_{\mu,i}$ 表示。记仿真周期长度为 T_m ，仿真周期为 k_m ，交通流的动力学特性为

$$\begin{aligned} \rho_{\mu,i}(k_m + 1) &= \rho_{\mu,i}(k_m) + \frac{T_m}{\Delta_\mu l_\mu} [q_{\mu,i-1}(k_m) - q_{\mu,i}(k_m)] \\ v_{\mu,i}(k_m + 1) &= v_{\mu,i}(k_m) + T_m \text{sat} \left\{ \sum_{\chi \in \mathcal{X}} \text{sat} [g_{\mu,i}^\chi(k_m)] \right\} \\ V[\rho_{\mu,i}(k_m)] &= v_{\mu,i,\max} \exp \left\{ -\frac{1}{a_{\mu,i}} \left[\frac{\rho_{\mu,i}(k_m)}{\rho_{\mu,i,cr}} \right]^{a_{\mu,i}} \right\} \\ q_{\mu,i}(k_m) &= \rho_{\mu,i}(k_m) v_{\mu,i}(k_m) l_\mu \end{aligned}$$

其中 $\rho_{\mu,i,cr}, v_{\mu,i,\max}$ 分别表示临界密度和自由流速度； V 为速度-密度基本图方程而 $a_{\mu,i}$ 为基本图方程中的无量纲超参； $\text{sat}(g) = \min\{g_{\max}, \max\{-g_{\min}, g\}\}$ 为饱和算子 (saturation operator)，其中 $g_{\min}, g_{\max} > 0$ ； $\mathcal{X} = \{(relax), (conv), (antic), (merge), (drop)\}$ 为涉及速度更新的几种状态（松弛、对流、预期、上匝道汇入和车道数折减）⁴， $g_{\mu,i}^\chi$ 为相应状态下的加速度 (km/h^2)

$$\begin{aligned} \text{relaxation} \quad g_{\mu,i}^{(relax)}(k_m) &= \frac{1}{\tau_{\mu,i}} \{V[\rho_{\nu,i}(k_m)] - v_{\mu,i}(k_m)\} \\ \text{convection} \quad g_{\mu,i}^{(conv)}(k_m) &= \frac{1}{\Delta_\mu} v_{\mu,i}(k_m) [v_{\mu,i-1}(k_m) - v_{\mu,i}(k_m)] \\ \text{anticipation} \quad g_{\mu,i}^{(antic)}(k_m) &= \frac{v_{\mu,i}(k_m) \rho_{\mu,i}(k_m) - \rho_{\mu,i+1}(k_m)}{\Delta_\mu \tau_{\mu,i} \rho_{\mu,i}(k_m)} \\ \text{on-ramp merge} \quad g_{\mu,i}^{(merge)}(k_m) &= -\delta_{\mu,i} \frac{q_{o,out}(k_m) \cdot v_{\mu,1}(k_m)}{\Delta_\mu l_\mu \rho_{\mu,1}(k_m)} \end{aligned}$$

⁴ 松弛项 (relaxation term) 考虑了驾驶人的行驶速度趋向于稳态速度的行为特性；对流项 (convection term) 考虑了上游流入车流量对速度的影响；预期项 (anticipation term) 考虑了下游车流密度变化对速度的影响。

赌书消得泼茶香，当时只道是寻常

7.4. A VARYING PARAMETER MULTI-CLASS SECOND-ORDER MACROSCOPIC TRAFFIC FLOW MODEL FOR COORDINATED RAMP METERING WITH GLOBAL AND LOCAL ENVIRONMENTAL OBJECTIVES
 (TRC, 2021)

$$\text{lane drop } g_{\mu,i}^{(drop)}(k_m) = -\phi_{\mu,i} \frac{(l_{\mu_2} - l_{\mu_1}) \cdot \rho_{\mu_1,S_{\mu_1}}(k_m) \cdot v_{\mu_1,S_{\mu_1}}^2(k_m)}{\Delta_{\mu_1} l_{\mu_1} \rho_{\mu_1,cr}}$$

其中 $\tau_{\mu,i}(h)$ 、 $v_{\mu,i}(\text{km}^2/\text{h})$ 、 $\delta_{\mu,i}$ 和 $\phi_{\mu,i}$ 均为超参（后两者无量纲）； μ_1, μ_2 分别表示车道数折减处上下游的路段； $q_{o,out}(k_m)$ 为从起始路段 $o \in \mathcal{L}_O$ 出发流向下游第一个路段的流量，在上式中即为进口匝道的输入流量

$$q_{o,out}(k_m) = \min\{q_o^{(1)}(k_m), q_o^{(2)}(k_m)\} \quad q_o^{(1)}(k_m) = \min\left\{d_o + \frac{w_o}{T_m}, Q_o^{\max}\right\} \quad q_o^{(2)}(k_m) = Q_o^{\max} \min\left\{1, \frac{\rho_{\mu,1,\max} - \rho_{\mu,1}}{\rho_{\mu,1,\max} - \rho_{\mu,1,cr}}\right\}$$

其中 d_o 为进口匝道的需求量； Q_o^{\max} 为进口匝道通行能力（即可能的最大的进口匝道流量）； $\rho_{\mu,1,\max}$ 为汇入点下游路段临界密度； w_o 为仿真周期 k_m 内的排队车辆数 (veh)，由需求量、下游路段交通状况和有无进行匝道控制决定，且进口匝道的交通状况由 w_o 确定

$$w_o(k_m + 1) = w_o(k_m) + T_m[d_o(k_m) - q_{o,out}(k_m)]$$

对于接入外部路网的路段 $b \in \mathcal{L}_D$ ，记其上游的节点为 $n \in \mathcal{N}$ ，认为路段 b 的输出流量 $q_{b,exit}(k_m)$ 等于节点 n 的输出流量 $q_{n,b,0}(k_m)$ ；

3. 以上构建了交通网络中连边上的交通模型。进一步构建静态节点模型 (static node model)。记节点 n 的速度和密度分别为 $v_n(k_m), \rho_n(k_m)$ ，流入节点的流量为 $q_{n,in}(k_m)$ ，流至下一连边 μ 的流量为 $q_{n,\mu,0}$ 。考虑三种节点形式：一进一出 ($\mu_1 \rightarrow n \rightarrow \mu_2$)、两进一出 ($\mu_1, \mu_2 \rightarrow n \rightarrow \mu_3$) 和一进两出 ($\mu_1 \rightarrow n \rightarrow \mu_2, \mu_2$)：

- 对于一进一出的情况，一般有

$$v_n(k_m) = v_{\mu_1,S_{\mu_1}}(k_m) \quad \rho_n(k_m) = \rho_{\mu_2,1}(k_m) \quad q_{n,in}(k_m) = q_{\mu_1,S_{\mu_1}}(k_m) \quad q_{n,\mu_2,0}(k_m) = q_{n,in}(k_m)$$

特别地，当 $\mu_1 \in \mathcal{L}_O, \mu_2 \in \mathcal{L}_M$ 时， μ_1 无速度定义，因此 v_n 无定义且无需计算 $g_{\mu_2,1}^{(conv)}(k_m)$ ，另外有 $q_{\mu_1,S_{\mu_1}}(k_m) = q_{\mu_1,out}(k_m)$ ；当 $\mu_1 \in \mathcal{L}_M, \mu_2 \in \mathcal{L}_D$ 时， v_n 无定义，且令 $\rho_n = 0$ ，另外有 $q_{n,\mu_2,0}(k_m) = q_{\mu_2,exit}(k_m)$ ；

- 对于两进一出的情况，一般有

$$v_n(k_m) = \frac{\sum_{\mu \neq \mu_3} q_{\mu,S_\mu}(k_m) v_{\mu,S_\mu}(k_m)}{\sum_{\mu \neq \mu_3} v_{\mu,S_\mu}(k_m)} \quad \rho_n(k_m) = \rho_{\mu_3,1}(k_m)$$

$$q_{n,in}(k_m) = \sum_{\mu \neq \mu_3} q_{\mu,S_\mu}(k_m) \quad q_{n,\mu_3,0}(k_m) = q_{n,in}(k_m)$$

特别地，当 $\mu_2 \in \mathcal{L}_O, \mu_1, \mu_3 \in \mathcal{L}_M$ 时， μ_2 无速度定义，因此 $v_n(k_m) = v_{\mu_1,S_{\mu_1}}(k_m)$ ，另外有 $q_{\mu_2,S_{\mu_2}}(k_m) = q_{\mu_2,out}(k_m)$ ；简化起见未考虑 $\mu_1, \mu_2 \in \mathcal{L}_O$ 的情况，但该情况是有意义的，可表示同时存在于路网某点的两条车队的情况；

- 对于一进两出的情况，此时需要考虑转向系数 $\beta_{n,\mu_2}(k_m), \beta_{n,\mu_3}(k_m)$

$$\beta_{n,\mu}(k_m) = \frac{q_{n,\mu,0}(k_m)}{q_{n,in}(k_m)} \quad \beta_{n,\mu_2}(k_m) + \beta_{n,\mu_3}(k_m) = 1$$

计算转向系数时不考虑不同类型车辆间的差异，假设所有车辆转向系数相同，此时在一般情况下有

$$v_n(k_m) = v_{\mu_1,S_{\mu_1}}(k_m) \quad \rho_n(k_m) = \frac{\sum_{\mu \neq \mu_1} \rho_{\mu,1}^2(k_m)}{\sum_{\mu \neq \mu_1} \rho_{\mu,1}(k_m)} \quad q_{n,in}(k_m) = q_{\mu_1,S_{\mu_1}}(k_m) \quad q_{n,\mu,0}(k_m) = \beta_{n,\mu}(k_m) q_{n,in}(k_m)$$

特别地，当 $\mu_3 \in \mathcal{L}_D, \mu_1, \mu_2 \in \mathcal{L}_M$ 时，因为 $\rho_{\mu_3,1}(k_m) = 0$ 所以 $\rho_n(k_m) = \rho_{\mu_2,1}(k_m)$ ，另外有 $q_{n,\mu_3,0}(k_m) = q_{\mu_3,exit}(k_m)$ ；当 $\mu_1 \in \mathcal{L}_O, \mu_2, \mu_3 \in \mathcal{L}_M$ 时， μ_1 无速度定义，因此 v_n 无定义且无需计算 $g_{\mu_2,1}^{(conv)}(k_m), g_{\mu_3,1}^{(conv)}(k_m)$ ，另外有 $q_{\mu_1,S_{\mu_1}}(k_m) = q_{\mu_1,out}(k_m)$ 。

from single-class to multi-class

1. 记 \mathcal{J} 为车辆类型集合，每类车具有完全不同的特性。记第 j 类车的最大加减速速度（绝对值）分别为 $g_{j,\max}, g_{j,\min}$ ，则令 $g_{\max} = \max\{g_{j,\max}\}$, $g_{\min} = \max\{g_{j,\min}\}$ ，同时给出了时间步长 T_m 的上界（这一约束也适用于单一类别模型的情况）

$$T_m \leq \min \left\{ \min_{\mu \in \mathcal{L}_M} \left\{ \frac{\Delta_\mu}{\min_{j \in \mathcal{J}} v_{j,\max}} \right\}, \min_{j \in \mathcal{J}} \left\{ \frac{v_{j,\max}}{g_{j,\max}} \right\}, \min_{j \in \mathcal{J}} \left\{ \frac{v_{j,\max}}{g_{j,\min}} \right\} \right\}$$

上式中的后两项是为了避免出行不合理加减速行为,而第一项则是 CFL 条件 (Courant-Friedrichs-Lowy condition)⁵;

- 在单类别模型中,因为假设网络中各路段交通流组成一致,各路段具有完全相同的基本图,因此建模时与基本图有关的参数 $\rho_{\mu,i,cr}, a_{\mu,i}, \tau_{\mu,i}, v_{\mu,i}, \phi_{\mu,i}, \delta_{\mu,i}, Q_{o,\max}$ 为定值。而多类别模型下网络中各连边可能具有不同的交通组成,基本图也不一致,因此建模时不同连边的基本图相关参数取值也不相同,定义参数向量 \mathbf{p}_m 表示网络中各路段的基本图相关参数如下

$$\mathbf{p}_m = \left[\rho_{cr}^T \quad v_{\max}^T \quad \mathbf{a}^T \quad \boldsymbol{\tau}^T \quad \mathbf{v}^T \quad \boldsymbol{\kappa}^T \quad \boldsymbol{\delta}^T \quad \boldsymbol{\phi}^T \quad \mathbf{Q}_{\max}^T \right]^T$$

- 考虑建立 \mathbf{p}_m 随交通流组成的变化模型。记 $\gamma_{\mu,i,j}(k_m)$ 为路段 (μ, i) 中车辆类型 j 所占的比例。特别地对于 $o \in \mathcal{L}_O$, $\gamma_{o,j}(k_m)$ 为路段 o 排队中车辆类型 j 所占的比例,而 $\theta_{o,j}(k_m)$ 表示路段 o 需求量中车辆类型 j 所占的比例。考虑小汽车和卡车两类车,记其类别集合分别为 $\mathcal{J}_{pc}, \mathcal{J}_{tr}$,且小汽车和卡车的比例分别为 $\Gamma_{\zeta,1}(k_m), \Gamma_{\zeta,2}(k_m)$,有

$$\Gamma_{\zeta,1}(k_m) = \sum_{j \in \mathcal{J}_{pc}} \gamma_{\zeta,j}(k_m) \quad \Gamma_{\zeta,2}(k_m) = \sum_{j \in \mathcal{J}_{tr}} \gamma_{\zeta,j}(k_m) \quad \Gamma_{\zeta,1}(k_m) + \Gamma_{\zeta,2}(k_m) = 1$$

上式中 ζ 为位置参数,用于替代 (μ, i) 或 o 。记向量 $\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2$ 分别为网络中各路段小汽车和卡车的比例向量。因为参数向量 \mathbf{p}_m 与网络交通组成有关,因此有 $\mathbf{p}_m = \mathbf{p}_m[\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2]$,并可进一步简化为 $\mathbf{p}_m = \mathbf{p}_m[\boldsymbol{\Gamma}_1]$ 。考虑交通流完全由小汽车组成和完全由卡车组成两种极端情况,两种情况对应的 \mathbf{p}_m 取值即为 \mathbf{p}_m 的上下界,且混合场景下 \mathbf{p}_m 取值随交通组成连续单调变化:

- 对于临界密度 $\rho_{\mu,cr}$ 和最大速度 $v_{\mu,\max}$,其最大、最下值分别对应于全为小汽车和全为卡车两种情况;
- 对于指数项 $a_{\mu,i}$,其最大、最下值分别对应于全为卡车和全为小汽车两种情况 ($a_{\mu,i}$ 越大则速度随密度增长而衰减的速度越慢);

- 进一步地建模参数 $p_{m,\zeta}$ 与路段交通组成 $\Gamma_{\zeta,1}$ 的具体关系,即给出 $p_{m,\zeta}[\Gamma_{\zeta,1}]$ 的解析形式。因为路段交通组成对交通特性的影响存在强非线性——交通流中少量的重型车即可显著影响整体交通状况,且总交通量越大少量重车的影响越明显,因此基于 logistics 曲线构建非线性关系

$$p_{m,\zeta}[\Gamma_{\zeta,1}(k_m)] = \sigma[\Gamma_{\zeta,1}(k_m)] \cdot p_{m,\zeta}[\Gamma_{\zeta,1}(k_m)] + (1 - \sigma[\Gamma_{\zeta,1}(k_m)]) p_{m,\zeta}[1 - \Gamma_{\zeta,1}(k_m)]$$

$$\sigma[\Gamma_{\zeta,1}(k_m)] = \left[\left\{ 1 + \exp\{-\psi_1[\Gamma_{\zeta,1}(k_m) - \psi_2]\} \right\}^{-1/\psi_3} \right]^{-1}$$

上式中 $\sigma[\Gamma_{\zeta,1}(k_m)] \in [0, 1]$ 为权重系数, ψ_1, ψ_2, ψ_3 量化交通组成对模型参数的影响,对于不同参数 ψ_1, ψ_2, ψ_3 的取值不同。为了反映交通总量对交通特性的影响,定义 ψ_2 为路段密度 $\rho_{\mu,i}(k_m)$ 的函数,相当于将交通量的影响转化为重车比例的影响

$$\psi_2[\rho_{\mu,i}(k_m)] = \psi_2^{(0)} + (1 - \psi_2^{(0)}) \frac{\rho_{\mu,i}(k_m)}{\rho_{\max}^{(0)} l_\mu}$$

上式考虑了路段车道数 l_μ 对密度 $\rho_{\mu,i}(k_m)$ 影响的影响——在大车比例一定的情况下车道数越多则路段密度的影响显然越小。

vehicle oriented dynamic traffic flow model

- 假设同一元胞中不同类型的车辆均匀分布,且同一元胞中的所有车辆速度相同,则易知 j 类车的密度 $y_{\mu,i,j}(k_m) = \gamma_{\mu,i,j}(k_m) \rho_{\mu,i}(k_m)$,并且有

$$y_{\mu,i,j}(k_m + 1) = y_{\mu,i,j}(k_m) + \frac{T_m}{\Delta_\mu l_\mu} [q_{\mu,i-1,j}(k_m) - q_{\mu,i,j}(k_m)]$$

⁵CFL 条件是有限差分和有限体积方法中的稳定性和收敛性分析的重要概念,最早于 1928 年提出。微分方程可通过离散化时间转为差分方程求解,显然时间步长越小得到的解越收敛于原微分方程,时间步长大甚至可能得到错误的解。对 CFL 条件的一般理解为基于离散化时间步长的求解速度必须小于物理扰动传播的速度,只有这样才能捕获所有物理扰动。考虑一维的情况,记 $\Delta t, \Delta x$ 分别为时间步长和区间长度, u 为速度,则定义 $C = u \Delta t / \Delta x$ 为 CFL 数,CFL 数即决定了差分方程解的稳定性,要求 $C \leq C_{\max}$ 。对于多维情况,则 CFL 条件变为 $C = \Delta t \sum_i u_i / \Delta x_i \leq C_{\max}$ 。对于精确解法一般令 $C_{\max} = 1$ 。

时 空 相 互 作 用 及 其 对 路 网 流 量 分 布 的 影 响

7.4. A VARYING PARAMETER MULTI-CLASS SECOND-ORDER MACROSCOPIC TRAFFIC FLOW MODEL FOR COORDINATED RAMP METERING WITH GLOBAL AND LOCAL ENVIRONMENTAL OBJECTIVES (TRC, 2021)

代入 $q_{\mu,i,j}(k_m) = \gamma_{\mu,i,j}(k_m)q_{\mu,i}(k_m) = \gamma_{\mu,i,j}(k_m)\rho_{\mu,i}(k_m)v_{\mu,i}(k_m)l_m = y_{\mu,i,j}(k_m)v_{\mu,i}(k_m)l_m$, 得

$$y_{\mu,i,j}(k_m + 1) = y_{\mu,i,j}(k_m) + \frac{T_m}{\Delta_\mu l_\mu} [y_{\mu,i-1,j}(k_m)v_{\mu,i-1}(k_m)l_m - y_{\mu,i,j}(k_m)v_{\mu,i}(k_m)l_m]$$

当 $i = 1$ 时 (即路段 μ 的第一个元胞), $i - 1 = 0$ 表示其上游的节点, 因此上式描述了相邻元胞或元胞与相邻节点间的 $y_{\mu,i,j}(k_m + 1)$ 递推式;

2. 注意到当 $\mu = o \in \mathcal{L}_O$ 时, 路段 μ 无速度定义, 且此时路段状态由排队长度表示。记 $y_{o,j}(k_m)$ 为 j 类车的排队长度 (veh), 则有

$$y_{o,j}(k_m + 1) = y_{o,j}(k_m) + T_m[\theta_{o,j}(k_m)d_o(k_m) - q_{o,out,j}(k_m)], \quad y_{o,j}(k_m) = \gamma_{o,j}(k_m)w_o(k_m)$$

上式中 $q_{o,out,j}$ 为路段 o 中 j 类车的输出流量。在长为 T_m 的时段内, $q_{o,out,j}(k_m)T_m$ 由两部分组成—— k_m 时的排队车辆数 $w_o(k_m)$, 其中车辆 j 占比为 $\gamma_{o,j}(k_m)$; T_m 时段内的总需求量 $d_o(k_m)T_m$, 其中车辆 j 占比为 $\theta_{o,j}(k_m)$ 。因此有

$$q_{o,out,j}(k_m) = \begin{cases} \frac{y_{o,j}(k_m) + \theta_{o,j}(k_m)d_o(k_m)T_m}{w_o(k_m) + d_o(k_m)T_m} q_{o,out}(k_m) & w_o(k_m) + d_o(k_m)T_m > 0 \\ 0 & w_o(k_m) + d_o(k_m)T_m = 0 \end{cases}$$

将 $q_{o,out,j}(k_m)$ 的表达式代入 $y_{o,j}(k_m)$ 递推式, 有

$$y_{o,j}(k_m + 1) = \begin{cases} \left[1 - \frac{q_{o,out}(k_m)}{\frac{w_o(k_m)}{T_m} + d_o(k_m)} \right] [y_{o,j}(k_m) + \theta_{o,j}(k_m)d_o(k_m)T_m] & w_o(k_m) + d_o(k_m)T_m > 0 \\ 0 & w_o(k_m) + d_o(k_m)T_m = 0 \end{cases}$$

ramp metering operation

记控制周期长度为 $T_c \geq T_m$ (为方便假设 T_c 为 T_m 的整数倍), 每隔一个控制周期匝道控制器计算调节率 $r_o(k_m) \in [r_{\min}, 1]$, 其中 $r_{\min} = 0$, 从而得到实际的匝道流率 $q_o(k_m) = r_o(k_m)q_{o,out}(k_m)$ 。

7.4.3 Emissions and environmental policies

COPERT emissions models

- 本节基于著名的 COPERT 模型计算环境效益指标;
- 记 \mathcal{Z} 为污染物集合; $z \in \mathcal{Z}$ 为相应的污染物。则对于 j 类车, 单车的单位排放 (grams/km/veh) 可表示为平均速度 v 的函数 $\Xi_{j,z}(v)$ 。对于小汽车和大卡车, 分别有

$$\Xi_{1,z}(v) = \frac{\xi_{1,z}^{(1)} + \xi_{1,z}^{(2)}v + \xi_{1,z}^{(3)}v^2}{1 + \xi_{1,z}^{(4)}v + \xi_{1,z}^{(5)}v^2} \quad \Xi_{2,z}(v) = \xi_{2,z}^{(1)} + \frac{\xi_{2,z}^{(2)}}{1 + \exp\{-\xi_{2,z}^{(3)} + \xi_{2,z}^{(4)}\ln v + \xi_{2,z}^{(5)}v\}}$$

上式中 $\xi_{j,z}^{(1)}, \dots, \xi_{j,z}^{(5)}$ 均为 COPERT 模型参数。则对于路段 μ 的元胞 i , j 类车在仿真周期 k_m 内的总排放量 $E_{\mu,i,j,z}(k_m)$ (grams) 计算为

$$E_{\mu,i,j,z}(k_m) = \Delta_\mu \cdot l_\mu \cdot \Xi(v_{\mu,i}(k_m)) \cdot q_{\mu,i,j}(k_m) \cdot T_m = \Delta_\mu \cdot l_\mu \cdot \Xi(v_{\mu,i}(k_m)) \cdot y_{\mu,i,j}(k_m) \cdot v_{\mu,i}(k_m) \cdot T_m$$

- 注意到计算排放需要速度, 而对于 $\mu = o \in \mathcal{L}_O$ 则无速度概念, 因此将 $E_{o,j,z}(k_m)$ 重新定义为

$$E_{o,j,z}(k_m) = w_o(k_m) \cdot \Xi(\hat{v}_o(k_m)) \cdot \hat{v}_o(k_m) \cdot T_m$$

上式中 $\hat{v}_o(k_m) = 10\text{km/h}$ 为 COPERT 模型中的速度下界。

global and local environmental policies

- 全局环境政策基于仿真周期 k_m 内的路网总排放 $E_{z,total}(k_m)$ (grams)

$$E_{z,total}(k_m) = \sum_{\mu \in \mathcal{L}_M} \sum_{i=1}^{S_\mu} \sum_{j \in \mathcal{J}} E_{\mu,i,j,z}(k_m) + \sum_{o \in \mathcal{L}_O} \sum_{j \in \mathcal{J}} E_{o,j,z}(k_m)$$

- 除了全路网的总排放 $\sum_{k_m} E_{z,total}(k_m)$, 还需注意某些需要特别关心的区域 (Special Interest Areas, SIA) 周围路段的排放。记 SIA 的集合为 \mathcal{A} , 对 $\eta \in \mathcal{A}$, 存在时间段 $(k_\eta^{(1)}, k_\eta^{(2)})$, 在此时间段内需额外限制其周围路段的排放, 具体地包括两类要求:

- 要求附近路网于该时间段内的总排放 $E_z^{\mathcal{A}_\eta}(k_\eta^{(1)}, k_\eta^{(2)})$ (grams) 低于某一限值, 即

$$E_z^{\mathcal{A}_\eta}(k_\eta^{(1)}, k_\eta^{(2)}) = T_m \sum_{j \in \mathcal{J}} \sum_{k_m=k_\eta^{(1)}}^{k_\eta^{(2)}} \left\{ \sum_{(\mu,i) \in \mathcal{A}_\eta} \Delta_\mu l_\mu \Xi(v_{\mu,i}(k_m)) y_{\mu,i,j}(k_m) v_{\mu,i}(k_m) + \sum_{o \in \mathcal{A}_\eta} w_o(k_m) \Xi(\hat{v}_o(k_m)) \hat{v}_o(k_m) \right\} \leq E_{z,max}^{\mathcal{A}_\eta}(k_\eta^{(1)}, k_\eta^{(2)})$$

- 要求附近路网于该时间段内的排放率 (任意时刻的排放) $G_z^{\mathcal{A}_\eta}(k_m)$ (grams/h) 低于相应限值, 即

$$G_z^{\mathcal{A}_\eta}(k_m) = \sum_{j \in \mathcal{J}} \left\{ \sum_{(\mu,i) \in \mathcal{A}_\eta} \Delta_\mu l_\mu \Xi(v_{\mu,i}(k_m)) y_{\mu,i,j}(k_m) v_{\mu,i}(k_m) + \sum_{o \in \mathcal{A}_\eta} w_o(k_m) \Xi(\hat{v}_o(k_m)) \hat{v}_o(k_m) \right\} \leq G_{z,max}^{\mathcal{A}_\eta}(k_m), \quad k_\eta^{(1)} \leq k_m \leq k_\eta^{(2)}$$

7.4.4 Optimal control problem formulation

- 记 $\mathbf{x}, \mathbf{r}, \mathbf{d}, \mathbf{p}$ 分别表示状态向量、控制参数向量、扰动向量和模型参数向量; Λ 表示总损失; λ, φ 分别表示状态转移损失 (state transition cost) 和最终状态损失 (final state cost)⁶; $\mathbf{f}, \mathbf{\Pi}$ 分别表示系统模型和状态、控制变量约束条件; \mathbf{r}_{min} 表示控制变量下限。离散时间的最优控制问题的一般形式可以写为

$$\min_r \Lambda = \varphi[\mathbf{x}(K_m)] + \sum_{k_m} \lambda [\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)]$$

$$\text{s.t. } \mathbf{x}(k_m + 1) = \mathbf{f}[\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)]$$

$$\mathbf{\Pi}[\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)] \geq \mathbf{0}$$

$$\mathbf{r}_{min} \leq \mathbf{r}(k_m) \leq \mathbf{1}$$

- 具体地, 在本最优控制模型中, 状态向量 \mathbf{x} 包括路段的密度、速度和排队等交通状态, 也包括 SIA 附近路段各污染物的排放信息; 控制向量 \mathbf{r} 则为各匝道的调节率; 扰动向量 $\mathbf{d} = [\dots, d_o, \theta_{o,1}, \dots, \theta_{o,|\mathcal{J}|}, \dots, \beta_{n,\mu}, \dots]^T$ 则包括各进口的总需求量、需求组成和转向系数;

- 另外, $\mathbf{\Pi}$ 表示局部环境约束, 包括上节所列的两道不等式:

- 对于第一道不等式——要求 SIA 附近特定时间段内的总排放低于特定阈值, 定义状态变量 $x_{\mathcal{A}_\eta,z}(k_m)$ 表示 \mathcal{A}_η 处仿至 k_m 时的污染物 z 的总排放 (grams), 则可将这一约束表现于目标函数——状态损失函数 $\varphi[\mathbf{x}(k_m)]$ 中 ($\omega_{z,term}$ 表示不同污染物的权重)

$$\varphi[\mathbf{x}(k_m)] = \sum_{\eta \in \mathcal{A}} \sum_{z \in \mathcal{Z}} \omega_{z,term} \max \left\{ 0, x_{\mathcal{A}_\eta,z}(k_m) - E_{z,max}^{\mathcal{A}_\eta} \right\}^2$$

⁶注意区分 λ, φ 的不同。顾名思义, 状态转移损失意味着 $\lambda(k_m)$ 的值仅与 k_m 时刻的状态 (包括控制) 有关; 而最终状态损失 $\varphi(k_m)$ 的值不仅与 k_m 有关, 也与之前的状态有关。

赌书消得泼茶香，当时只道是寻常

7.4. A VARYING PARAMETER MULTI-CLASS SECOND-ORDER MACROSCOPIC TRAFFIC FLOW MODEL FOR COORDINATED RAMP METERING WITH GLOBAL AND LOCAL ENVIRONMENTAL OBJECTIVES

- 采用类似地方法，将第二道不等式——要求 SIA 附近特定时间段内的排放率低于特定阈值——表现于目标函数的状态转移损失函数 $\lambda[\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)]$ 中，具体地定义 λ_{maxe} 表示状态转移过程中的 SIA 附近路段排放率造成的损失

$$\lambda_{maxe} [\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)] = \sum_{\eta \in \mathcal{A}} \sum_{z \in \mathcal{Z}} \max \left\{ 0, G_z^{\mathcal{A}_\eta}(k_m) - G_{z,\max}^{\mathcal{A}_\eta} \right\}^2$$

通过罚函数法将局部环境约束转化至目标函数中，可以舍去相关约束；

4. 除了局部环境效益外，状态转移损失函数 λ 还考虑其它因素：

全局环境效益 同样地引入不同污染物 z 的权重 $\omega_{z,tot}$ ，则定义 λ_{toem} 为

$$\lambda_{toem} [\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)] = \sum_{z \in \mathcal{Z}} \omega_{z,tot} E_{z,total}(k_m)$$

效率 效率指标由总行程时间 (total time spend, TTS, veh·h) λ_{TTS} 表示，考虑了网络中所有通行的车辆和排队车辆

$$\lambda_{TTS} [\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)] = T_m \left[\sum_{\mu \in \mathcal{L}_M} \sum_{i=1}^{S_\mu} \Delta_\mu l_\mu \rho_{\mu,i}(k_m) + \sum_{o \in \mathcal{L}_O} w_o(k_m) \right]$$

均衡性 (equity) 均衡性具有直接和间接两种量化方法。其一方面可以显式地表示为各进口匝道车辆完成排队并在路网上行驶一定距离的平均时间的标准差 $\lambda_{equi}(h^2)$ 。记 Δ_{eq} 表示预设的“代表性距离”，集合 \mathcal{L}_o^{eq} 为满足 $\sum_{\mu \in \mathcal{L}_o^{eq}} \Delta_\mu \approx \Delta_{eq}$ 的直接位于进口匝道 o 下游的路段集合，则进口匝道 o 上车辆完成排队并行驶相应距离所需的平均时间 $t_o^{eq}(k_m)$ 为

$$t_o^{eq}(k_m) = \frac{w_o(k_m)}{q_o(k_m)} + \sum_{\mu \in \mathcal{L}_o^{eq}} \sum_{i=1}^{S_\mu} \frac{\Delta_\mu}{v_{\mu,i}(k_m)}$$

进而可以得到 λ_{equi}

$$\lambda_{equi} [\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)] = \frac{1}{|\mathcal{L}_O|} \sum_{o \in \mathcal{L}_O} [\bar{t}^{eq}(k_m) - t_o^{eq}(k_m)]^2 \quad \bar{t}^{eq}(k_m) = \frac{1}{|\mathcal{L}_O|} \sum_{o \in \mathcal{L}_O} t_o^{eq}(k_m)$$

另一方面，均衡性也可以间接地由各进口匝道 o 的排队长度表示。引入匝道 o 的最大排队长度 $w_{o,max}$ ，若 $w_o(k_m) > w_{o,max}$ ，则该匝道发生溢出。因此定义 λ_{maxq} 如下

$$\lambda_{maxq} [\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)] = \sum_{o \in \mathcal{L}_O} \max \left\{ 0, w_o(k_m) - w_{o,max} \right\}^2$$

控制稳定性 定义 λ_{rvar} 以避免控制措施过大幅度地变化

$$\lambda_{rvar} [\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)] = \sum_{o \in \mathcal{L}_O} [r_o(k_m - 1) - r_o(k_m)]^2$$

综上所述，可以得到最终的交通转移损失函数 λ

$$\lambda = \omega_0 \lambda_{TTS} + \omega_1 \lambda_{equi} + \omega_2 \lambda_{maxq} + \omega_3 \lambda_{rvar} + \omega_4 \lambda_{toem} + \omega_5 \lambda_{maxe}$$

7.4.5 Differential evolution and solution evaluation

- 求解上述最优控制问题的主流解法可大体分为两大类。其中最精确的是基于梯度的全局优化算法 (gradient based coupled with a globalization method)，方法要求解析地计算 $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ 以得到梯度信息。在本例中因为速度更新较为复杂，不太适合该方法，因此选择更为简单的随机搜索算法，具体地选择差分进化 (differential evolution, DE) 算法 (详见19.4节)。在进行变异操作时，对前一半变异向量变异算子选择 **DE/rand/1**，而对后一半则选择 **DE/rand-to-best/1**，因数值研究表明相关设置具有更好的收敛性；

- 7.4. A VARYING PARAMETER MULTI-CLASS SECOND-ORDER MACROSCOPIC TRAFFIC FLOW MODEL FOR COORDINATED RAMP METERING WITH GLOBAL AND LOCAL ENVIRONMENTAL OBJECTIVES
2. 在给定的初始状态和全时空需求信息的基础上, 即可基于某一时刻的控制参数推出下一时刻的全局状态, 因此可直接优化全时空的控制参数, 将动态优化控制问题转化为大规模的静态优化问题;
 3. 对应于损失函数考虑的各类指标, 设置如下全局指标评价控制策略于不同方面的效果:

效率 考虑总行程时间 (TTS) 和总延误时间 (TWT) 两类指标, 分别记为 $\Phi_{TTS,tot}$, $\Phi_{TWT,tot}$:

$$\Phi_{TTS,tot} = \sum_{k_m=0}^{K_m} \lambda_{TTS} [\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)] \quad \Phi_{TWT,tot} = T_m \sum_{k_m=0}^{K_m} \sum_{o \in \mathcal{L}_o} w_o(k_m)$$

均衡性 由 λ_{equi} 的时间均值表示: $\Phi_{equi,tot} = \frac{1}{K_m} \sum_{k_m=0}^{K_m-1} \lambda_{equi} [\mathbf{x}(k_m), \mathbf{r}(k_m), \mathbf{d}(k_m) | \mathbf{p}(k_m)]$

全局环境效益 对于污染物 $z \in \mathcal{Z}$, 以其网络总排放量为评价指标: $\Phi_{z,tot} = \sum_{k_m=0}^{K_m} E_{z,total}(k_m)$

局部环境效益 对于任意一处 SIA $\mathcal{A}_\eta \in \mathcal{A}$ 和污染物 $z \in \mathcal{Z}$, 考虑 $\Phi_{term}^{\mathcal{A}_\eta,z}$, $\Phi_{maxe}^{\mathcal{A}_\eta,z}$ 两项指标, 分别表示其在特定时段内的总排放量和排放率:

$$\Phi_{term}^{\mathcal{A}_\eta,z} = x_{\mathcal{A}_\eta,z}(k_m) \quad \Phi_{maxe}^{\mathcal{A}_\eta,z} = \sqrt{\sum_{k_m=0}^{K_m-1} \max \left\{ 0, G_z^{\mathcal{A}_\eta}(k_m) - G_{z,max}^{\mathcal{A}_\eta} \right\}^2}$$

注意到 $x_{\mathcal{A}_\eta,z}(k_m)$ 原本即定义为 \mathcal{A}_η 处 k_m 时刻污染物 z 的总排放, 因此 $\Phi_{term}^{\mathcal{A}_\eta,z}$ 的定义式中无连加符号。

7.4.6 Test network case study & Results and discussion

1. 考虑五种类型的车辆 $\mathcal{J} = \{1, 2, 3, 4, 5\}$, 其中 $\mathcal{J}_{pc} = \{1, 2, 3, 4\}$ 为汽油小汽车、 $\mathcal{J}_{tr} = \{5\}$ 为柴油卡车, 小汽车和卡车各占 50%;
2. 其中四类小汽车具有不同的排放模型, 分别为 Euro1-4, 环保等级依次增加。柴油卡车则属于 Euro-3 型。Euro-3 型柴油卡车与 Euro-3 型汽油小汽车具有不同的排放模型;
3. 除了无控制场景外, 对于多目标协同控制策略, 考虑不同的目标权重设计共 8 种策略:
 - 策略 1 仅考虑效率指标, 以最小化总行程时间 (TTS) 为目标;
 - 策略 2 仅考虑全局环境效益, 以最小化系统总排放为目标;
 - 策略 3 仅考虑关于 SIA 附近排放指标的约束;
 - 策略 4 结合策略 2 和策略 3, 在最小化系统总排放的同时考虑 SIA 附近的排放约束;
 - 策略 5 考虑了所有优化目标和约束, 包括效率、均衡性、最大排队约束、全局环境效益和局部排放约束;
 - 策略 6 相比于策略 5 少考虑了局部排放约束;
 - 策略 7 相比于策略 6 采用了更为严格的最大排队约束;
 - 策略 8 相比于策略 7 少考虑了局部排放约束。

需要说明的是以上所有 8 种策略均考虑了控制稳定性约束;

4. 对于每种策略, 在基于差分进化算法求解时, 先以较小的种群规模和迭代次数多次求解, 计算多次求解的均值和方差。又以较大的种群规模和迭代次数进行一次求解。比较两种求解策略的效果差异;
5. 对比策略 1 和 2 可以看出, 效率优化与环境改善之间存在一定的矛盾。策略 1 在取得最高效率的同时增加了 NOx 的排放, 而策略 2 则可在减少排放的同时一定地提升效率。另外策略 1 的稳定性弱于策略 2。其原因在于策略 1 为了最优化效率, 倾向于最大限度地限制直接影响瓶颈的交通流, 从而依然会形成较为集中的排队。控制策略过于严格的同时也不利于降低网络排放。相比之下, 网络总排放更适合作为独立的目标函数, 因其有助于得到更温和的控制策略以分散排队、减少排放, 从而实现效率、环境和控制稳定性的均衡;
6. 仅考虑局部排放约束并通过罚函数法优化的策略在多数指标上的表现均是最差的。相比于以网络总排放为目标函数, 仅考虑局部排放使得策略过于短视。

7.4.7 英汉互译

English	Chinese	English	Chinese	English	Chinese
equitable	合理的、公平的	hysteresis	磁滞现象	medium	介质 (n)
resilient	可迅速恢复的	dedicated	专用的	invariant	不变的
convection	对流	exogenous	外生的	characterization	塑造 (n)
monotonically	单调地	agile	灵活的	depreciate	贬值 (v)
bifurcation	分叉 (n)	the bulk of	大部分	detrimental	有害的
viable	可行的	compromise	妥协 (n,v)	myopic	短视的
pronouncedly	明确地	at the expense of	以…为代价		

7.5 Real-time Traffic Network State Estimation And Prediction with Decision Support Capabilities: Application to Integrated Corridor Management (TRC, 2016)

面向综合通道控制决策支持的实时交通网络状态评估与预测

ABSTRACT

- □ ×

- 提出了一种面向交通网络管理的带有决策支持功能的实时网络状态评估与预测系统；
- 系统允许交通网络管理者评估实时网络状态、预测拥挤演化、并针对常发性和偶发性拥堵生成控制策略；
- 系统中网络状态评估预测模块与网络控制模块相集成，闭环滚动更新；
- 网络控制模块采用元启发搜索机制以集成多种控制策略；
- 以德克萨斯达拉斯 (Dallas) 的 US-75 通道为场景进行事故仿真实验，结果表明系统可提升网络全局性能。

7.5.1 Background

- 面对日益增长的交通需求，无法无限制的提升路网的物理通行能力，因此需要综合一系列交通控制策略提升现有交通网络的利用率，而前提在于精确建模交通网络的时空供需交互特征及其相关的拥挤现象；
- 多数综合交通控制方案中所涉及的控制策略包括行程前 (pre-trip) 和途中 (en-route) 信息提供、动态配时、匝道控制、动态路肩 (dynamic shoulder lanes) 与拥堵收费 (congestion pricing) 等；
- 已有研究主要关注实时交通管理系统的研究并在离线 (offline) 仿真环境下检验不同交通控制策略于不同交通状况下的有效性。然而以下问题较少被研究：
 - 提出可以集成多种控制策略的实时交通管理系统；
 - 在合理的网络空间尺度和预测时间窗口内评估控制系统的效果。
- 本研究将带有决策支持功能的实时交通控制系统集成于动态交通分配仿真模型 DIRECT (Dynamic Intermodal Routing Environment for Control and Telematics):
 - 系统可评估网络当前状况、预测拥堵发展、并针对常发和偶发性拥堵生成控制策略；
 - 系统采用滚动窗口 (rolling horizon) 框架，并集成交通状态评估和预测模块；
 - 交通状态评估模块由实时动态交通仿真模型实现，与真实时间保持同步，接收实时数据更新仿真环境，确保估计的结果与实际观测状况一致；
 - 交通状态预测模块周期性地激活另一仿真器，以更快速度仿真以预测一定时间窗口后的交通状况；
 - 给定预测交通状况后激活决策支持模块以提供交通控制方案。
- 综合通道控制旨在整合多模式交通网络与交通设施，并基于协同的多种控制方案提升通道的总体服务水平。将系统应用于综合通道控制，具体地：
 - 假设各交通管理机构已经提出了各自的具体交通控制方案，如信号配时、匝道控制等等；
 - 综合通道控制系统基于多种单一控制方式的具体效果组合得到综合控制方案；
 - 为优化最优综合控制方案，基于元启发搜索算法构造综合控制方案候选集合，并基于仿真预测模块判断各综合控制方案的效果，将最优方案用于实际控制。
- 本研究的贡献主要如下：

- 基于仿真预测模块，系统可以得到与期望网络状况和驾驶员路径选择相匹配的主动交通控制策略；
- 交通管理过程由准闭环 (quasi closed-loop) 系统实现；
- 生成的控制策略涉及多个交通控制机构相互配合，更适合区域交通控制；
- 可以综合一系列单一交通控制策略。

7.5.2 Problem definition and formulation

变量	说明	变量	说明
A, N	网络连边与节点集合	a, n	网络连边与节点 $a \in A, n \in N$
Z	网络需求子区集合	I, J	需求发生与结束子区集合 $I, J \subset Z$
i, j	需求发生与结束子区 $i \in I, j \in J$	K	所有 OD 路径集合
k	网络路径 $k \in K_{ij} \quad i \in I, j \in J$	T	用于研究、分析的窗口长度
T', T''	窗口内的仿真间隔数和发车间隔数	t, τ	当前时间与发车时间 $t = 1, \dots, T', \tau = 1, \dots, T''$
H	用于评价生成的交通控制方案的预测窗口长度	D	网络交通需求水平 $D = \sum_{\tau} \sum_{i \in I} \sum_{j \in J} r_{ijk}^{\tau}$
Δ, h	仿真间隔与预测窗口仿真间隔数 $\Delta = \frac{T}{T'}, h = \frac{H}{\Delta}$	$r_{ijk}^{\tau}, r_{ijk}^{*}$	在发车时间 τ 以 $i - j$ 为 OD 的车辆数及其中选择路径 k 的车辆数
c	事故编号	t_s^c, t_e^c, a^c	事故 c 的开始时间、结束时间和发生道路
TT_{ijk}^{τ}	在发车时间 τ 以 $i - j$ 为 OD 并选择路径 k 的车辆的期望行程时间	R	被交通管理方案覆盖的区域 (受事故影响的区域)，为事故位置和需求水平的函数 $R = f(a^c, D)$
Y^R	位于区域 R 内的交通控制设施集合	y	交通控制设施编号 $y \in Y^R$
$P(y)$	交通控制设施 y 的可行设置集合	π_t^y	t 时刻区域 R 内控制设施 y 的一种可行的设置方案
Π_t^R	t 时刻区域 R 内的综合控制方案 $\Pi_t^R = \{\pi_t^y : y \in Y^R\}$	$g_H(\Pi_t^R)$	在预设的预测窗口 H 内应用控制方案 Π_t^R 的网络总期望行程时间
$\Phi(\cdot), \Gamma(\cdot)$	综合控制方案下驾驶员的期望行程时间和路径选择		

1. 假设需求路径流 r_{ijk}^{τ} 已知，车辆沿着其预设的需求路径直至终点，途中会受到当前交通状况的影响；
2. 在整个分析窗口内网络总行程时间表示为

$$\sum_{\tau=1}^{T''} \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} r_{ijk}^{\tau} \times TT_{ijk}^{\tau}$$

3. 事故 c 发生后在其位置和当时需求水平作用下，会对路网的一部分区域 R 造成影响，需要交通管控；
4. 交通管控方案 Π_t^R 应综合区域 R 内的多种控制方式。如通过调整信号配时、调整匝道调节率、和通过可变信号板发布更多参考性建议。用数学语言表示为： $\Pi_t^R = \{\pi_t^y : y \in Y^R\}$ ；
5. 认为综合交通控制方案 Π_t^R 的应用将影响网络的需求模式。在方案执行前通过网络状态预测模块预测方案的有效性和对需求模式的影响，预测窗口长度即为 H ；
6. 使得预测的总期望行程时间 $g_H(\Pi_t^R)$ 最小的综合交通控制方案 Π_t^{*R} 即为最优控制方案。除了行程时间外，也可采用其它如排放、流量输出等指标评价并排序策略；
7. 交通网络具有较高的动态性和随机性，滚动窗口法 (**rolling horizon, RH**) 是实际中最常用的动态系统求解方法：记 l 表示单次滚动的间隔数 (在本文中即为预测模块激活的周期大小)、 α 为目前滚动预测的次数 (即已完成多少次滚动)、 h 为预测窗口的间隔数。则当第 $\alpha + 1$ 次滚动预测开始时，预测窗口 H 时段内网络的车辆 D_H 包括预测窗口 H 内发生的需求量 r_{ijk}^{τ} 和从仿真开始至今进入网络但尚未离开的车辆数 s'_{ijk}^{τ}

$$D_H = \sum_{\tau=\alpha l+1}^{\alpha l+h} \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} r_{ijk}^{\tau} + \sum_{\tau=1}^{\alpha l} \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} s'_{ijk}^{\tau}$$

注意在上式中 r_{ijk}^{τ} 与 s'_{ijk}^{τ} 的上标 τ 对应不同的取值范围；

8. 定义函数 $\Gamma(\Pi_t^R)$ 表示在预测窗口 H 内策略 Π_t^R 的实施对驾驶员路径选择的影响 (进而影响交通分布)，其输出为向量 $(r_{ijk}^{\tau}, s'_{ijk}^{\tau})$ 。又定义 $\Phi(r_{ijk}^{\tau}, s'_{ijk}^{\tau}, \Pi_t^R)$ 表示在预测窗口 H 内策略 Π_t^R 所导致的路径期望通行时间

TT_{ijk}^τ 。在得知所以 τ 时刻的路径行程时间 TT_{ijk}^τ 后即可得到策略 Π_t^R 所导致网络总行程时间 $g_H(\Pi_t^R)$ 。则优化问题可用数学语言表示为

$$\begin{aligned} \min \quad & g_H(\Pi_t^R) = \sum_{\tau=\alpha l+1}^{\alpha l+h} \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} (r_{ijk}^\tau \times TT_{ijk}^\tau) + \sum_{\tau=1}^{\alpha l} \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} (s'_{ijk}^\tau \times TT_{ijk}^\tau + \tau \Delta - \alpha l \Delta) \\ \text{s.t.} \quad & R = f(a^c, D_H) \\ & \pi_t^y \in P(y) \quad \forall y \in Y^R \\ & \Pi_t^R = \{\pi_t^y : y \in Y^R\} \\ & r_{ij}^\tau = \sum_{k \in K} r_{ijk}^\tau \quad \forall i, j, \quad \tau = \alpha l + 1, \dots, \alpha l + h \\ & (r_{ijk}^\tau, s'_{ijk}^\tau) = \Gamma(\Pi_t^R) \quad \forall i, j, k, \quad \tau = \alpha l + 1, \dots, \alpha l + h \\ & TT_{ijk}^\tau = \Phi(r_{ijk}^\tau, s'_{ijk}^\tau, \Pi_t^R) \quad \forall i, j, k, \quad \tau = \alpha l + 1, \dots, \alpha l + h \\ & \text{All variables} \geq 0 \end{aligned}$$

7.5.3 Real-time traffic network management system: overall framework & Decision support capabilities

1. 求解上述优化问题存在以下难点:
 - 目标函数非线性, 且无法保证为凸优化;
 - 难以给出 $\Phi(\cdot), \Gamma(\cdot)$ 的封闭解析形式;
 - 随着网络规模的增大, 网络中控制设施和控制策略的组合呈指数增长, 带来较大的计算压力。
2. 针对上述问题, 基于仿真的动态交通分配模型捕捉网络交通状态演化特征, 又基于元启发搜索算法搜索最优的控制策略组合;
3. 动态交通分配仿真模型采用 DIRECT 软件实现, 是交通状态评估与预测模块的基础。DIRECT 的输入文件包括驾驶员的数量、OD 和发车时间, 并基于网络交通状态在发车时为驾驶员分配路径。每隔 3-5 分钟更新网络交通路径。分配路径时考虑网络各连边的综合费用, 包括行程时间、高速收费和私家车运行费用。驾驶员默认将遵从分配的路径, 除非在行程中收到额外的控制信息;
4. 设预测模块的激活间隔 $l = 5 - 10 \text{ min}$, 预测窗口 $H = 0.5 - 1 \text{ h}$;
5. 除了交通状态评估和预测模块, 进一步地引入在线 (online) 校正模块。为使得交通状态评估与预测模块仿真的交通状态与实际状态具有一致的演化特征, 定期激活校正模块:
 - 校正模块主要校正网络各连边的速度与需求水平;
 - 定义参数 δ_V, δ_D 分别表示连边速度与需求的不一致阈值, 当状态评估模块仿真状态与实际观测状态之差超过相应阈值时认为需要调整;
 - 若速度不一致, 则校正模块通过反馈控制调整交通流传播模型的相关参数。进一步地介绍参数调整方法。假设对于网络连边 a , 宏观交通流模型如下, 其中 v_{\max}^a, k_{\max}^a 均为存在清晰物理意义的超参, v_t^a, k_t^a 为模型估计的交通状态, 而 β_t^a 则为待调整参数, 可反解出 β_t^a 的表达式如下

$$v_t^a = v_{\max}^a \left[1 - \left(\frac{k_t^a}{k_{\max}^a} \right)^{\beta_t^a} \right] \Rightarrow \beta_t^a = \frac{\ln \left(1 - \frac{v_t^a}{v_{\max}^a} \right)}{\ln k_t^a - \ln k_{\max}^a}$$

定义 $v_{t,o}^a, e_{t,V}^a$ 分别为观测的实际速度和观测值与估计值的误差, 若 $|e_{t,V}^a| \geq \delta_V$ 则基于误差修正 β_t^a 。但为避免修正幅度过大造成仿真系统不稳定, 定义修正误差上限 Δ , 若 $|e_{t,V}^a| \leq \Delta$ 时则修正所有误差, 若 $|e_{t,V}^a| > \Delta$ 则仅修正部分误差。具体的修正方法如下, 其中 $\beta_{t,c}^a, v_{t,c}^a$ 分别表示 β_t^a, v_t^a 的修正值

$$\beta_{t,c}^a = \frac{\ln \left(1 - \frac{v_{t,c}^a}{v_{\max}^a} \right)}{\ln k_t^a - \ln k_{\max}^a} \quad v_{t,c}^a = \begin{cases} v_t^a - e_t^a = v_{t,o}^a & \delta_V \leq |e_{t,V}^a| \leq \Delta \\ v_t^a \pm \Delta & |e_{t,V}^a| > \Delta \end{cases}$$

- 若需求不一致，则校正模块令在线仿真（状态评估模块）倒退（rollback）若干个步长后在新的需求模式下重新仿真，其中新的需求模式基于优化求解得到。进一步地介绍需求模式的求解方法。假设倒退的时间长度为 R ，且可进一步细分为 R^s 个观测间隔和 R^d 个发车间隔。记 $V_{t,o}^a, V_t^a$ 分别表示 t 时刻连边 a 观测和预测的流量，则构造如下二次规划问题⁷。其中 \hat{d}_{ij}^τ 为决策变量，表示 τ 时刻发车的 OD 对 $i - j$ 间的需求量。 $p_{ij\tau}^{at}$ 表示模型估计的 t 时刻连边 a 上于 τ 时刻发车的以 $i - j$ 为 OD 的车辆的比例。

$$\begin{aligned} \min \quad & \sum_a \sum_{t \in R^s} (V_{t,o}^a - V_t^a)^2 \\ \text{s.t.} \quad & V_t^a = \sum_i \sum_j \sum_{\tau \in R^d} p_{ij\tau}^{at} \cdot \hat{d}_{ij}^\tau \\ & \hat{d}_{ij}^\tau \geq 0 \quad \forall i, j, \tau \end{aligned}$$

二次规划问题是非线性规划中求解方法较为成熟的一类。注意到 V_t^a 是有界的（通行能力）。将其取值范围离散化为 N 等份，对于第 n 个子区间，以子区间中值 $u_t^{a,n}$ 作为整个子区间的取值。若 V_t^a 的实际取值落入第 n 个子区间，则有 $u_t^{a,n} \simeq V_t^a$ 。此时即可将以上二次规划问题近似转化为如下线性规划问题，其中 $\lambda_t^{a,n}$ 为构造的指示变量，为新规划问题的决策变量。求解 $\lambda_t^{a,n}$ 记得得到需求模式 \hat{d}_{ij}^τ

$$\begin{aligned} \min \quad & \sum_a \sum_{t \in R^s} \sum_n e_t^{a,n} \cdot \lambda_t^{a,n} \\ \text{s.t.} \quad & e_t^{a,n} = (V_{t,o}^a - u_t^{a,n})^2 \quad \forall a, t, n \\ & \sum_n u_t^{a,n} \cdot \lambda_t^{a,n} = \sum_i \sum_j \sum_{\tau \in R^d} p_{ij\tau}^{at} \cdot \hat{d}_{ij}^\tau \quad \forall a, t \\ & \sum_n \lambda_t^{a,n} = 1 \quad \forall a, t \\ & \lambda_t^{a,n} \geq 0 \quad \forall a, t, n \end{aligned}$$

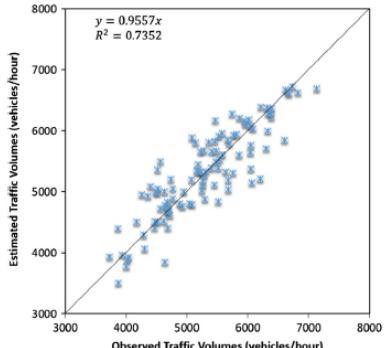
6. 基于遗传算法（详见第19.3节）搜索最优策略组合方案，策略组合方案编码为染色体 (chromosome) 的形式，代表一个个体，染色体上的基因对应不同的控制方式，如信号配时、匝道控制、路径引导等；
7. 以预测窗口内的总行程时间为适应度，计算每一个体的适应度，并通过淘汰、重组、编译、遗传等方式生成新的适应性更强的种群。多次迭代直至算法收敛。

7.5.4 Testbed description & Experiments, results and analysis

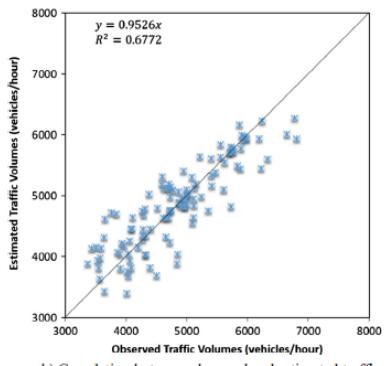
1. 为验证所述系统的效果，将系统应用于处理德克萨斯达拉斯 (Dallas)US-75 通道晚高峰时段发生的一次偶发性拥堵事件；
2. 基于探测器采集实时交通数据，主要是以五分钟为精度的车辆计数和速度数据，同时包括事故特征——位置、持续时间和封闭车道数。速度和车辆统计数据用于校准仿真模块参数；
3. 在后续实验中，系统将在动态配时、路径诱导、动态匝道控制和动态路肩等四种控制策略中集成一种或多种进行综合通道控制：
 - 对于动态配时，研究者预设了 11 种确认可行的配时方案，设计综合控制方案时只需从 11 种预设策略中选择最合适的一种即可，无需自行优化信号配时；
 - 对于路径诱导，基于可变信号板发布沿途交通信息引导部分驾驶员分流绕行。假设每一可变信号板均可分流一定比例的驾驶员，具体的比例由综合控制方案确定；
 - 对于匝道控制，假设每一匝道控制器支持其饱和流率的 100%、75%、50%、25% 作为匝道调节率，综合控制方案采用其中一种用于匝道控制；
 - 动态路肩控制可在事故期间开放路肩以增加通行能力，综合控制方案只需确定是否开启路肩。
4. 在控制实验中，控制模块随事故的发生而被激活，并持续至最后一起事故结束 30 分钟后；

⁷二次规划：目标函数为二次非线性函数，而约束为线性的规划问题。

7.5. REAL-TIME TRAFFIC NETWORK STATE ESTIMATION AND PREDICTION WITH DECISION SUPPORT CAPABILITIES: APPLICATION TO INTEGRATED CORRIDOR MANAGEMENT (TRC, 2016)



a) Correlation between observed and estimated traffic volumes for the US-75 freeway - northbound

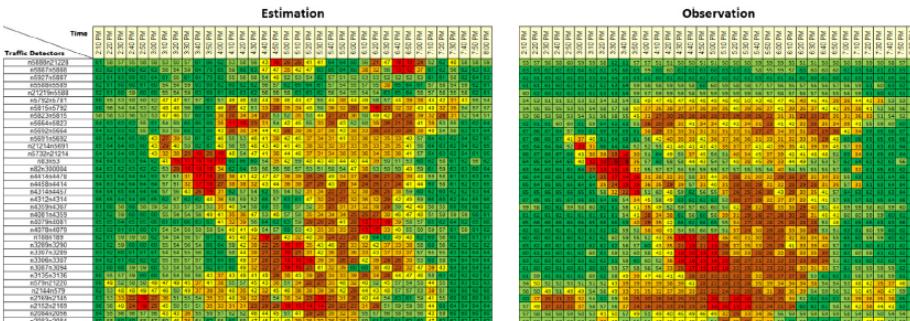


b) Correlation between observed and estimated traffic volumes for the US-75 freeway - southbound

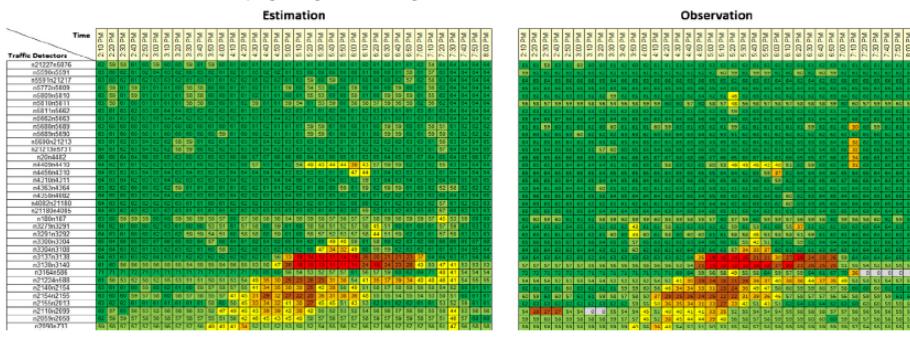
Northbound RMSE = 377 vehicles per hour
Southbound RMSE = 395 vehicles per hour

Fig. 5. Observed vs. estimated traffic hourly volumes.

图 7.4 流量（左）、速度（右）校正效果。可以看到校正后仿真器可较好的复现真实交通状态。



a) Speed profile comparison for northbound direction



b) Speed profile comparison for southbound direction



Northbound RMSE = 8.97 miles per hour
Southbound RMSE = 5.68 miles per hour

Fig. 6. Estimated and observed speed profile for the US-75 freeway.

- 理论上控制方案的覆盖区域应等于事故的影响区域。而在控制实验中，覆盖区域的大小由研究者预设。实验表明，随着覆盖范围的增大，综合控制方案的灵活度更高，因此具有更好的效果；
- 控制实验进一步指出，当控制模块以较短的间隔更新综合控制方案时可起到更好的控制效果。注意到以**10分钟为更新间隔的控制效果优于以5分钟为更新间隔**，一个合理的解释是更新间隔过小会使得对控制策略的过渡修正，即可能控制策略还未发生效果即被更换；
- 控制实验最后指出，当所有四种策略均可被用于集成综合控制方案时可起到更好的控制效果。

7.5.5 英汉互译

English	Chinese	English	Chinese	English	Chinese
envision	展望 (v)	hypothetical	假想的	fidelity	忠诚、精确性
normative	规范的	cumbersome	冗长的	quadratic	二次的
toll facility	收通行费的道路设施	telematics	远程通信技术	intermodal	多式联运的
paradigm	范式	priori	先验的	disparity	差距
jurisdiction	管辖权	quasi	类似的	cope	(成功) 应付 (v)
habitual	习惯的	intractable	棘手的	recursively	递归地
rollback	倒退 (v)	plausible	合理的	synergy	协同作用 (n)

7.6 A Simulation-Based Optimization Framework for Urban Transportation Problems (OR, 2013)

ABSTRACT

- □ ×

本文提出了一种基于仿真的优化方法 (simulation-based optimization, SBO) 以高效利用复杂随机的微观交通仿真器处理各种交通问题。本文基于元模型 (metamodel) 实现对仿真器的近似。元模型由通用项 (general-purpose component) 和物理项 (physical component) 组成。其中通用项为一二次多项式 (quadratic polynomial), 用于局部近似。物理项为一解析交通模型 (本文采用网络排队模型), 提供简易的全局解析信息。包含通用项和物理项的元模型确保了 SBO 问题可以高效实现, 基于较少的计算资源处理复杂问题。元模型与无导数信赖域算法 (derivative-free trust region algorithm) 结合, 并应用于处理瑞士洛桑的信号控制问题, 考虑多种需求及较少的计算资源。结果显示方法可得到较好的配时方案以减少行程时间。目

7.6.1 Introduction & Literature review

- 因为微观交通仿真器可以捕捉微观交通演化特征, 因此常被用于评价交通控制策略。理论上也可反过来基于微观交通仿真研究合适的策略, 然而该问题具有较大的难度;
- 记网络特征参数为 p (如拓扑结构、网络需求、交通参数等), 控制策略为 x , 显然网络交通状态 f 为 p, x 的函数 $f = f(x|p)$ 。以 p, x 为参数进行微观交通仿真, 可以得到 $f(x|p)$ 的估计值 $F(x|p)$ 。记 $F_i(x|p)$ 表示第 i 次仿真的结果, 则基于仿真的优化问题 (simulation-based optimization, SBO) 可以建模为以下优化问题

$$\min_{x \in \Omega} f(x|p) = \mathbb{E}[F(x|p)] = \frac{1}{r} \sum_{i=1}^r F_i(x|p)$$

上式中 Ω 为策略空间, r 为重复仿真的次数;

- 尽管微观仿真器可以提供复杂信息, 但也使得目标函数具有极强的非线性, 且无闭合形式, 需要多次仿真得到近似值。因此 SBO 问题难以求解;
- SBO 问题的传统求解方法包括三种: 直接搜索、随机梯度和元模型 (替代模型) 方法。本文主要关注元模型方法。基于元模型的 SBO 问题求解的核心在于以确定性的元模型替代微观交通仿真的随机部分, 从而将随机优化问题转为确定性优化问题。元模型优化过程包括迭代以下两步:
 - 首先基于仿真观测值拟合元模型;
 - 基于拟合的元模型求解优化问题, 得到试验点 (trial point)。试验点即为本文中的 x , 代表一种控制方案。

为拟合得到准确的元模型, 仍需要进行数次交通仿真, 这也是该方法的缺点;

- 传统上元模型被分为物理元模型 (physical metamodel) 和功能元模型 (functional metamodel):
 - 物理元模型常常针对特定问题设计, 其结构与参数往往具有较明确的物理意义;
 - 功能元模型大多适用于一般问题, 其结构并不反映物理规律, 设计时更多考虑分析的易用性。低阶多项式是最常用的功能元模型形式, 例如置信域方法中大多采用二阶多项式作为原问题的替代。除此之外也有采用样条函数 (spline method)、径向基函数 (radial basis function)、克里金方法 (Kriging surrogate) 等。
- 因为功能元模型强调对一般问题的拟合能力, 因此 SBO 问题中元模型常采用功能元模型以估计目标函数。然而因为功能元模型不反映问题的物理规律, 因此只能用于近似经过微观仿真的参数。当参数未经过仿真时, 模型往往无法有效近似目标函数;
- 本文提出了一种基于元模型的快速求解 SBO 问题的方法, 可以以较小的计算消耗得到可行的控制策略。元模型包含物理项 (实现低精度的全局估计) 和功能项 (实现高精度的局部估计), 实现了详细而复杂的交通仿真器与粗略而简易的交通模型的结合;
- 在本文中, 记控制策略维度为 d , 则所需的仿真次数只需介于 $d \sim 5d$ 之间, 即只需极少的计算资源。

7.6.2 Metamodel

1. 采用 AIMSUN 作为微观仿真器。仿真器基于车头时距模型和 OD 矩阵生成出行 (trip)，基于跟驰模型、换道模型、间距接受模型和路径选择模型模拟驾驶员行为。考虑多种控制方案 $\{x^1, x^2, \dots\}$ ，多次仿真后可得到相应控制方案 x 下控制效果 $f(x|p)$ 的估计值 $\{\hat{f}(x^1|p), \hat{f}(x^2|p), \dots\}$ ；
2. 采用可解析的网络排队模型作为简易交通模型。网络排队模型结合了传统交通模型、排队论和部分交通法规。将道路视为排队系统，基于有限容量排队论 (**finite capacity queueing theory**) 建模受通行能力限制的道路的运行水平 (详见第 11*I 节)。记参数 q 表示网络拓扑结构、总需求、转向概率等信息，参数 y 表示交通流特征 (如溢出概率，溢出速度等)，则在控制方案 x 下交通模型可得到 $f(x|p)$ 的确定性近似 $T(x, y|q)$ ；
3. 通过元模型 m 建立交通模型结果 T 与微观仿真结果 \hat{f} 之间的联系

$$m(x, y|\alpha, \beta, q) = \alpha \cdot T(x, y|q) + \varphi(x|\beta)$$

上式中 φ 即为元模型的功能项， α, β 为元模型参数；

4. 元模型与无导数⁸信赖域 (**derivative-free trust region**) 算法 (第 17.3.4 节) 结合。**信赖域算法**的核心思想是在每一次迭代时在一信赖域内对目标函数进行近似，常以二次多项式近似。定义 φ 如下

$$\varphi(x|\beta) = \beta_1 + \sum_{j=1}^d \beta_{j+1} \cdot x_j + \sum_{j=1}^d \beta_{d+j+1} \cdot x_j^2$$

注意到所定义的 φ 并非完全二次多项式 (不含 $x_i x_j$ 项)，因为已有的无导数信赖域数值仿真实验指出当近似项的二阶导矩阵为对角阵时具有更高的优化效率；

5. 信赖域算法需要多次迭代。每轮迭代时，基于当前和以往所有迭代的仿真观测值拟合元模型参数 α, β 。拟合时为多次仿真的观测值分配权重。记第 k 轮迭代时已仿真过 n_k 个不同的方案 $\{x^1, x^2, \dots, x^{n_k}\}$ ，基于最小二乘拟合 α, β

$$\alpha, \beta = \arg \min_{\alpha, \beta} \sum_{i=1}^{n_k} \left\{ w_{ki} \cdot [\hat{f}(x^i|p) - m(x^i, y^i|\alpha, \beta, q)] \right\}^2 + [w_0 \cdot (\alpha - 1)]^2 + \sum_{i=1}^{2d+1} (w_0 \cdot \beta_i)^2 \quad w_{ik} = \frac{1}{1 + \|x^{n_k} - x^i\|_2}$$

7.6.3 Optimization algorithm

1. 本文集成元模型与信赖域优化算法，该过程称为多模型融合算法 (**multi-model algorithm**)。之所以选择信赖域方法而非线搜索，是因为信赖域方法可以更自然地扩展至具有正定 **Hessian** 矩阵的非二次型模型，也更接近本文的研究对象；
2. 考虑到 SBO 问题中难以求目标函数导数，因此采用无导数方法进行优化；
3. 记 n_{\max} 为最大仿真次数； r 为一个控制方案的重复仿真次数； $0 < \bar{d} < \Delta_{\max}$ 分别为信赖域半径的下界和上界。对于第 k 轮仿真，记 $m_k(x, y|\nu_k, q)$ 为元模型； x_k 为控制方案； Δ_k 为信赖域半径； $\nu_k = (\alpha_k, \beta_k)$ 为元模型参数； n_k 为总仿真次数； u_k 为被拒绝的连续试验点数量 (the number of successive trial points rejected)； g_k 为元模型于控制方案 x_k 处的一阶导数。基于信赖域算法优化 SBO 问题的算法流程如下：
 - 首先判断元模型于控制方案 x_k 处的平稳性。定义下界 $\varepsilon_c > 0$ 。若 $\|g_k\| > \varepsilon_c$ ，说明控制方案 x_k 并非位于元模型 m_k 一阶驻点 (**first-order stationary point**, 也称 **first-order critical point**) 附近，可以优化 x_k 使得元模型 m_k 取值降低。然而若 $\|g_k\| \leq \varepsilon_c$ ，即 x_k 位于元模型 m_k 一阶驻点附近，则难以进行进一步优化，此时需要构造新的元模型 m_k 及相应的信赖域 Δ_k 。对于本文所涉及的复杂交通问题，SBO 优化过程中并未出现 $\|g_k\| \leq \varepsilon_c$ 的情况；
 - 在信赖域 Δ_k 内优化元模型，得到 x_k 的增量 s_k 。 $x_k + s_k$ 称为试验点 (**trial point**)；
 - 模型拟合：定义阈值 $0 < \eta < 1$ 并计算指标 ρ_k 以判断是否接受得到的试验点

$$\rho_k = \frac{\hat{f}(x_k) - \hat{f}(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$$

⁸ 所谓无导数优化 (**derivative-free optimization**) 又称无梯度优化，是指在优化过程中不需要计算目标函数的梯度，只利用目标函数的值优化目标函数。

若 $\rho_k \geq \eta_1$, 则接受试验点, 令 $x_{k+1} = x_k + s_k$, $u_k = 0$; 反之则拒绝该点, 令 $x_{k+1} = x_k$, $u_k = u_k + 1$ 。若试验点被接受增加仿真总数量 $n_k = n_{k-1} + r$ (本文中 $r = 1$, 即重复一次仿真), 并更新权重 w 以拟合新的元模型 m_{k+1} ;

- 模型提升: 拟合新模型 m_{k+1} 后定义阈值 $0 < \bar{\tau} < 1$, 计算元模型参数更新量 τ_{k+1}

$$\tau_{k+1} = \frac{\|\nu_{k+1} - \nu_k\|}{\|\nu_k\|}$$

若 $\tau_{k+1} < \bar{\tau}$, 则进一步提升其对微观交通状态的捕捉能力。基于特定分布 (本文采用均匀分布) 采样得到新的控制方案 x , 进行微观交通仿真和数值仿真得到 x 下的 \hat{f} 和 T 。将其纳入仿真总量中 $n_k = n_{k-1} + r$ (本文中 $r = 1$, 即重复一次仿真), 并更新权重 w 以提升元模型 m_{k+1} ;

- 定义阈值 $0 < \gamma < 1 < \gamma_{inc}$, $\bar{u} \in \mathbb{N}^*$, 则信赖域半径 Δ_k 的更新公式如下

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_{inc}\Delta_k, \Delta_{\max}\} & \rho_k > \eta_1 \\ \max\{\gamma\Delta_k, \bar{d}\} & \rho_k \leq \eta_1, u_k \geq \bar{u} \\ \Delta_k & \text{otherwise} \end{cases}$$

本文中相关超参取值为 $\Delta_{\max} = 10^{10}$, $\Delta_0 = 10^3$, $\eta_1 = 10^{-3}$, $\gamma = 0.9$, $\gamma_{inc} = 1.2$, $\varepsilon_c = 10^{-6}$, $\bar{\tau} = 0.1$, $\bar{d} = 10^{-2}$, $\bar{u} = 10$, $w_0 = 0.1$ 。

7.6.4 Traffic signal control

1. 设置偏置变量 (offset variable) 协同相邻交叉口的信号配时;
2. 考虑固定配时的信号控制策略, 并通过离线仿真优化。其中偏置长度、周期长度和全红相位长度均为定值, 周期内各阶段的组成和顺序同样已知。优化变量为不同交叉口的绿信比 (green split);
3. 该问题属于传统的信号控制问题, 对于单交叉口可以轻易求解, 但多交叉口需考虑相邻调查口排队的交互, 求解难度较大;
4. 定义 b_i 表示交叉口 i 的可用周期比 (available cycle ratio)⁹; $x(j)$ 为 j 相位的绿信比; x_L 为最小绿信比; \mathcal{J} 为信号交叉口集合; $\mathcal{P}_i(i)$ 为信号交叉口 i 的相位集合。又定义目标函数 $f(x|p)$ 表示行程时间, 则信号控制问题可以表示为

$$\begin{aligned} \min_x \quad & f(x|p) = \mathbb{E}[F(x|p)] \\ \text{s.t.} \quad & \sum_{j \in \mathcal{P}_i(i)} x(j) = b_i, \quad \forall i \in \mathcal{J} \\ & x \geq x_L \end{aligned}$$

5. 基于信赖域方法求解上述问题。则在算法的第 k 轮迭代时, 方法求解的子问题较原问题新增三个约束。上式中约束 $h_2(x, y|q) = 0$ 表示组成元模型物理项 (网络排队模型, 详见第 11*I 节) 的方程组。网络排队模型以网络中每一车道视为存在容量限制的排队系统, 车道交通状态由排队长度表示;

$$\begin{aligned} \min_x \quad & m_k = \alpha_k T(x, y|q) + \varphi(x|\beta_k) \\ \text{s.t.} \quad & \sum_{j \in \mathcal{P}_i(i)} x(j) = b_i, \quad \forall i \in \mathcal{J} \\ & h_2(x, y|q) = 0 \\ & \|x - x_k\|_2 \leq \Delta_k \\ & x \geq x_L, \quad y \geq 0 \end{aligned}$$

6. 出于模型可辨识性的角度, 对于一共 i 个交叉口的 p 个相位, 则模型中共有 $p - i$ 个独立的决策变量。因此在元模型的二次多项式 φ 中, 仅包含 $p - i$ 个自变量。

⁹ 定义周期长度减去所有全红时长的时间为可用周期长度 (available cycle time), 又定义可用周期长度与周期长度的比值为可用周期比

7.6.5 Empirical analysis

1. 采用经过校正的微观仿真器 (AIMSUN) 仿真瑞士洛桑路网交通状态。基于元模型拟合微观交通仿真器。考虑三种元模型的效果：仅包含物理项（网络排队模型）的元模型 T ; 仅包含通用项（二次多项式）的元模型 φ ; 和本文所提的包括物理项和通用项的元模型 M 。三种模型优化时采用相同的信赖域算法；
2. 首先考虑一个包含 2 个相邻交叉口、由 12 条道路组成的小型路网。路网共有 21 条车道，其中 13 条车道受信号控制。路网共 13 个相位，因此元模型中的决策变量维度为 13。令最大仿真次数为 150；
3. 为比较不同元模型的优化效果，随机抽样 10 种信号配时方案作为初始解，对于每种初始解进行 n 轮仿真优化得到 10 组最终解，以平均行程时间为指标评价信号配时方案的效果，且对于每一组最终解重复进行 50 次仿真，最终可以得到 500 个行程时间观测值组成行程时间分布。以行程时间的累积分布曲线评价模型的优化效果；
4. 实验结果指出，元模型 M 和 φ 相较于 T 均可以得到更优的控制策略，因为模型 M 和 φ 均可以基于观测结果更新参数以更好地拟合实际交通状态。**基于所提元模型 M 优化的配时方案对应的 500 组行程时间观测值具有最小的均值和方差，而且仅需要较小的仿真轮数 ($n = 10$) 即可得到很好的优化策略。结果说明优化策略具有最优的优化效果和稳定性，且计算损耗极小；**
5. 进一步地考虑一个更大的路网，并以晚高峰 (17-18h) 的需求作为输入。路网包含 15 个相邻交叉口，由 48 条道路组成，共有 102 条车道，其中 60 条受信号控制。路网共 51 个相位；
6. 对于大规模优化问题，仅考虑 1 组初始解，每次优化进行 150 轮仿真，共进行 10 次优化，得到 10 组最终解。实验结果指出，**元模型 φ 得到的 10 组最终解并非均优于初始解；而元模型 M 得到的 10 组解基本均优于初始解；**
7. 进一步增加计算资源。**优化仿真上限增加至 3000 次时，三种模型求解的控制策略的控制效果非常相似，其中元模型 M 的优化结果略优于其它模型。**

7.6.6 英汉互译

English	Chinese	English	Chinese	English	Chinese
quadratic	平方的	tractable	简易的	budget	预算
derivative-free	无导数	intricate	复杂的	disaggregate	无组织的，分解 (v)
asymptotically	渐进地	interpolate	内插 (v)	radial	辐射状的
asymptotic	渐进的	compatible	相容的	centroid	质心
delimit	划界 (v)	precede	先于 (v)		

第8章

网络需求调控

8.1 Active learning for multi-objective optimal road congestion pricing considering negative land use effect (TRC, 2021)

ABSTRACT

道路拥挤收费策略旨在改善交通系统于高峰期的拥堵水平，然而也需要考虑收费策略对土地利用效率的负面影响。本文提出了基于多目标双层规划的拥堵收费策略，旨在优化区域可达性、区域土地利用多样性、和区域总行程时间。因为所提的问题为 NP 难，本文创造性地提出了基于多目标贝叶斯优化的主动学习优化 (active learning optimization) 算法。算法基于概率信息寻找新的可行解，从而改善双层规划问题的求解效率。以江阴市为例进行验证优化收费策略的效果。所提算法也可应用于其它交通相关的复杂黑盒问题。

8.1.1 Introduction

- 经济学家认为道路拥堵收费是一种有效且可持续的应对交通拥堵的经济手段；
- 案例指出收费政策在改善交通拥堵的同时也对区域土地利用产生了负面影响，降低了区域可达性和土地利用多样性；
- 需要优化拥堵收费策略，在保证其对交通系统改善的同时降低其对土地利用的影响。为此存在两个问题：
 - 量化拥堵收费策略对土地利用的影响
 - 已有研究的做法大体可分为调查法（在应用收费策略前）、应用前后对比分析法（ex-post monitoring and analysis, 在应用收费策略后）和建模法；
 - 通过大规模调查，研究者主要关注拥堵收费策略对居住、人口、商业区和就业的影响。例如拥堵收费策略将影响个体的职住选择；
 - 通过观察部分拥堵收费实例，研究者主要关注策略对商业、零售、房地产和工作的影响¹。例如拥堵收费将打击收费区内的零售行业；
 - 通过模型分析，研究者认为拥堵收费策略可能使得就业分布分散，降低土地利用多样性；策略也可能形成边界效应 (boundary effect)，降低某些区域的可达性。独立模型 (independent model) 和复合模型 (integrated model) 是两种常用的建模拥堵收费对土地利用的影响的方法。前者仅考虑交通建模，而忽视了交通系统和土地利用的交互反馈，因而适用于短期效用评价。后者协同了交通模型和土地利用模型，通过考虑交通与土地利用的交互现象而适用于长期效用评价，且可规避外部不确定性因素的影响。
 - 改善收费策略对土地利用的负效应。尽管大量研究已研究拥堵收费策略对土地利用的负效应，较少研究优化该负效应。主要原因在于交通-土地利用协同模型的计算复杂度极高，难以得到解析解。

¹需要指出的是相关数据未必准确，因其可能受到其它外部因素如经济形势等的影响。同时因为拥堵收费对土地利用的影响是一个长期过程，因此需要长期观测。

4. 本文采用交通-土地利用复合模型的方法建模拥堵收费对土地利用的影响，并基于多目标双层规划算法进行优化。其中上层模型优化收费策略，下层模型为考虑拥堵收费的交通-土地利用交互模型；
5. 所提双层规划问题的求解难点如下：
 - 本研究要求实现多目标优化，已有研究多采用帕累托最优解 (Pareto optimal solution)；
 - 即使上下两层模型均为线性模型，求解一个双层规划问题仍是 NP 难的，而本研究中的下层模型为非线性，进一步增大求解难度。为此本文采用高斯过程模型 (Gaussian process model) (见27.5节) 作为概率代理模型 (probabilistic surrogate model) 以替代下层模型，直接建模决策变量（下层模型输入，即收费方案）和目标函数（下层模型输出，即土地利用和交通状态指标）的关系。为了进一步降低计算消耗，本研究基于贝叶斯多目标优化提出了主动学习优化 (active learning optimization) 算法，可减少调用下层复合模型的时间并更快确定上层模型的收费方案。相比结合仿真信息和解析模型的元模型，本文所提替代模型更容易搭建；而相比元启发算法，所提算法更为高效、求解质量也更高。因为借助概率替代模型可以将下层的黑箱模型近似为解析模型，提升求解效率。
6. 贝叶斯优化 (见27.6节) 属于无梯度全局优化算法。方法一般假设目标函数采样自一概率替代模型，通过基于训练数据搭建高斯过程模型，贝叶斯优化可以搜索未探索区域的潜在最优解，并通过采集函数 (acquisition function) 获取下一采样点，重复迭代搜索全局最优解。

8.1.2 Model formulation

1. 将研究区域划分为交通分析小区，记为集合 $Z = \{z^1, \dots, z^M\}$ 。问题为多目标优化，记目标类型个数为 W ，则目标集合为 $\{c^1, \dots, c^W\}$ ；
2. 双层优化模型中下层模型为交通-土地利用复合模型，基于给定收费方案确定平衡态下的出行成本、交通流模式和土地利用模式；上层模型为多目标优化模型，基于平衡态下的交通-土地利用模式优化收费方案；
3. 上层优化模型考虑土地利用水平和交通状态两类指标：
 - 土地利用水平指标包括平均区域可达性和平均区域土地利用多样性。平均区域可达性 \bar{A} 定义为居民实现特定活动的容易程度。定义 A_i 为区域 i 的可达性； E_j 为区域 j 中一种活动的数量； t_{ij} 为区域 j 至 i 的最小行程时间； $I(t_{ij})$ 为关于 t_{ij} 的函数； ω, γ, λ 为函数 $I(t_{ij})$ 的参数，则 \bar{A} 由下式计算

$$\bar{A} = \frac{1}{M} \sum_{i \in Z} A_i = \frac{1}{M} \sum_{i \in Z} \left(\sum_{j \in Z} E_j I(t_{ij}) \right) = \frac{1}{M} \sum_{i \in Z} \left(\sum_{j \in Z} E_i \cdot \omega \cdot \exp\{\gamma + \lambda \cdot t_{ij}\} \right)$$

类似地，平均区域土地利用多样性 \bar{D} 则定义为一个区域内活动的丰富程度。**如果某区域的土地利用多样性高，则意味着从该区域可能存在较大规模的内部出行。**以香农信息熵量化土地利用多样性 \bar{D} 。定义 D_i 为区域 i 的土地利用多样性； p_i^s 为区域 i 中土地利用类型 s 的面积占比； S_i 为区域 i 中土地利用类型的数量，则 \bar{D} 由下式计算

$$\bar{D} = \frac{1}{M} \sum_{i \in Z} D_i = \frac{1}{M} \sum_{i \in Z} \left(-\frac{\sum_{s=1}^{S_i} p_i^s \ln p_i^s}{\ln S_i} \right)$$

- 交通状态指标采用区域总行程时间。定义 q_{ij} 为区域 j 与 i 间的流量，则区域总行程时间 T 为

$$T = \sum_{i \in Z} \sum_{j \in Z, j \neq i} q_{ij} t_{ij}$$

4. 下层模型采用经典的 TRANUS 模型，为基于 logit 模型的土地利用-交通复合模型。模型可以实现土地利用和交通系统的复杂反馈交互，基于最大化期望效用实现空间位置选择、出行生成、出行分布、模式划分、交通分配等。模型广泛应用于研究交通政策（如拥堵收费）或多种土地利用影响下的交通系统-土地利用演化关系；
5. 将上下层模型结合起来，得到多目标双层规划问题。令 $\tau, v(\tau), y(\tau), L(\tau), f^w$ 分别表示收费费用、均衡交通流量、均衡土地利用模式、交通-土地利用复合模型和第 w 个目标函数，则本文所研究的考虑拥堵收费的

土地利用负效益的多目标双层规划问题可以写为

$$\begin{aligned} \max_{\tau} \quad & \mathbf{f}(\tau, \mathbf{v}(\tau), \mathbf{y}(\tau)) = \{f^w(\tau, \mathbf{v}(\tau), \mathbf{y}(\tau)), \forall w \in C\} \\ \text{s.t.} \quad & \tau \in \Omega \\ & [\mathbf{v}(\tau), \mathbf{y}(\tau)] = L(\tau) \end{aligned}$$

后文中 $\mathbf{f}(\tau, \mathbf{v}(\tau), \mathbf{y}(\tau)), f^w(\tau, \mathbf{v}(\tau), \mathbf{y}(\tau))$ 将分别简记为 $\mathbf{f}(\tau), f^w(\tau)$ 。

8.1.3 An active learning algorithm for the multi-objective bi-level programming model

- 本研究的目标是确定最优拥堵收费方案使得平均区域可达性、平均区域土地利用多样性和区域总行程时间可以达到帕累托最优（见第17.8.1节）；
- 设计基于多目标贝叶斯优化的主动学习优化算法以减少调用下层交通-土地利用复合模型的次数并提升计算效率；
- 本研究基于贝叶斯优化求解所提的多目标双层规划拥堵收费问题。贝叶斯优化是一种全局优化算法，借助算法中的两个核心组件——概率代理模型（常为高斯过程回归模型）和采集函数，算法可以选择对当前目标函数具有最大提升效果的采样点从而加速迭代。借助这一特点有研究指出贝叶斯优化具有优于元启发算法的计算效率。具体介绍可参考第27.6节；
- 因为所提的多目标双层规划拥堵收费问题是一个 NP 难问题，难以得到解析解，因此本文基于代理模型直接拟合决策变量和目标函数之间的关系。以零噪声、零均值的高斯过程回归模型作为代理模型。对于多目标优化问题，每一个目标函数对应一个高斯过程回归模型。令高斯过程回归中的核函数为 $\nu = \frac{5}{2}$ 的 Matern 核函数，具体形式如下

$$k(\tau_i, \tau_j) = \left(1 + \frac{\sqrt{5}\|\tau_i - \tau_j\|}{l} + \frac{5\|\tau_i - \tau_j\|^2}{3l^2} \right) \exp \left\{ -\frac{\sqrt{5}\|\tau_i - \tau_j\|}{l} \right\}$$

上式中 l 为高斯过程回归模型超参，由最大化边际似然确定。给定下一次迭代决策变量采样点 τ_{n+1} ，根据高斯过程基本原理，目标函数的每一个分量取值均服从高斯分布 $f^w(\tau_{n+1}) \sim N(\mu^w(\tau_{n+1}), \sigma^w(\tau_{n+1}))$ 。关于高斯过程回归模型技术细节和超参数优化可参考第27.5节；

- 每一轮迭代的采样点由采集函数确定。第27.6节初步介绍了采集函数的原理和几种采集函数，但主要针对单目标优化问题进行讨论。而对于多目标优化问题，采集函数可分为基于标量 (scalar-based) 和基于帕累托最优 (Pareto optimal-based) 两类。本研究选用后者。对于基于帕累托最优的采集函数，其输出为一个向量，每一个分量对应一个目标函数的高斯过程回归，因此每一次采样将与之前的所有采样组合，去除被支配解后得到一个可能的 Pareto 前沿（见第17.8.1节）。该前沿将未知的目标函数空间划分为两部分，即被支配区 (dominated region) 和非被支配区 (non-dominated region)。问题的 Pareto 最优前沿位于非被支配区内。基于 Pareto 最优的采集函数在每轮迭代中的目标即是尽可能充分地探索并压缩非被支配区，当目标函数空间内不再存在非被支配区时则得到问题的 Pareto 最优前沿（相关概念可参考下图）；
- 为了在每轮迭代时最大限度地探索非被支配区，基于 Pareto 最优的采集函数包括三种主流的构造方法：基于超体积的概率提升 (hypervolume-based probability of improvement)、期望最大提升 (expected maximum improvement) 和基于欧式距离的期望提升 (Euclidean distance-based expected improvement)。研究采用基于超体积的概率提升采集函数，方法不要求数据标准化并显著降低计算复杂性，具体流程为：
 - 首先计算概率提升 $P_M(\tau)$ ，定义为新的采样点 τ 位于非支配区内的概率，也可以理解为新的采样点可提升当前的 Pareto 解集 \mathcal{P} 的概率。记 Q 表示目标函数空间的非支配区； $\phi^w[f^w(\tau)]$ 为目标函数 $f^w(\tau)$ 取值的概率密度函数。则 $P_M(\tau)$ 的定义式为一个 W 维空间内的重积分：

$$P_M(\tau) = \int_{\mathbf{f}(\tau) \in Q} \prod_{w=1}^W \phi^w[f^w(\tau)] d f^w(\tau)$$

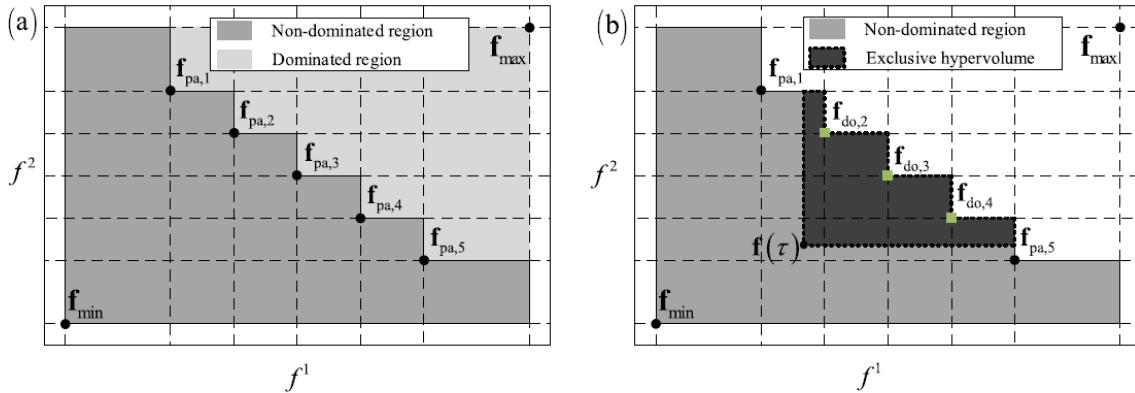


Fig. 3. Illustration of the Pareto front of two objective functions. Note: $\mathbf{f}_{pa,i}$ and $\mathbf{f}_{do,i}$ are the points on the Pareto front and the dominated points, where $i = 1, \dots, 5$; \mathbf{f}_{min} and \mathbf{f}_{max} denote the lower and upper bound of the hypervolume region. The dark and light shaded areas in figure (a) represent the non-dominated area and the dominated area, respectively. The non-dominated area is the integration area Q based on multi-dimensional space $P_M(\tau)$, and the hypervolume of the dominated area bounded by \mathbf{f}_{min} and \mathbf{f}_{max} is the improvement of the Pareto set \mathcal{P} . The entire shaded area is decomposed into several independent rectangular areas by a binary partitioning procedure based on the existing points on the Pareto front. Figure (b) illustrates that the exclusive hypervolume of the objective function value point $\mathbf{f}(\tau)$ relative to the Pareto set \mathcal{P} (the black shaded region) can be computed from the existing cells.

图 8.1 基于 Pareto 最优的多目标贝叶斯优化采集函数相关概念参考

为简化计算, 将区域 Q 离散化为若干超立方体。记超立方体的个数为 b ; 立方体 a 内第 w 个目标函数的上下界分别为 u_a^w, l_a^w ; Φ^w 为目标函数 $f^w(\tau)$ 取值的概率累计函数, 则 $P_M(\tau)$ 的计算式可简化为

$$P_M(\tau) = \sum_{a=1}^b \prod_{w=1}^W (\Phi^w[u_a^w] - \Phi^w[l_a^w])$$

- 对于当前的 Pareto 解集 \mathcal{P} , 计算每一采样点 τ 所带来的超体积提升 (hypervolume improvement), 即加入采样的目标函数 $\mathbf{f}(\tau)$ 后可缩小的非支配区的超体积, 记为 $I(\mathbf{f}(\tau), \mathcal{P})$:

$$I(\mathbf{f}(\tau), \mathcal{P}) = \begin{cases} H_{imp}(\mathbf{f}(\tau), \mathcal{P}) & \mathbf{f}(\tau) \text{ is not dominated by } \mathcal{P} \\ 0 & \text{else} \end{cases} \quad H_{imp}(\mathbf{f}(\tau), \mathcal{P}) = H(\mathcal{P} \cup \{\mathbf{f}(\tau)\}) - H(\mathcal{P})$$

式中 $H(\cdot)$ 表示当前 Pareto 前沿划分的支配区的超体积。本研究中 $I(\mathbf{f}(\tau), \mathcal{P})$ 一方面用于构建采集函数, 一方面也作为贝叶斯优化的收敛判据;

- 同时考虑采样点 τ 的提升概率和提升幅度 (超体积提升程度), 构造基于超体积的概率提升采集函数 $P_{HV}(\tau)$

$$P_{HV}(\tau) = I(\mu(\tau), \mathcal{P}) \cdot P_M(\tau)$$

式中 $\mu = [\mu^1(\tau), \dots, \mu^W(\tau)]^T$ 为高斯过程回归的均值函数。同样地, 可通过将 Q 离散化为若干超立方体简化计算。记超立方体的个数为 b ; 立方体 a 内各个目标函数的上下界向量分别为 $\mathbf{u}_a^w, \mathbf{l}_a^w$, 则上式简化为

$$P_{HV}(\tau) = \left(\sum_{a=1}^b \mathcal{V}(\mu(\tau), \mathbf{u}_a^w, \mathbf{l}_a^w) \right) \cdot P_M(\tau)$$

$$\mathcal{V}(\mu(\tau), \mathbf{u}_a^w, \mathbf{l}_a^w) = \begin{cases} \prod_{w=1}^W (u_a^w - \max\{l_a^w, \mu^w(\tau)\}) & u_a^w > \mu^w(\tau), \forall w \in C \\ 0 & \text{else} \end{cases}$$

8.1.4 Case study

- 以江阴市为研究案例。以交通分析小区为土地利用和交通状态的基本分析单元。将江阴市划分为 265 个交通小区, 其中 56 个小区位于拥堵收费区内, 费用区间介于 10-30 元之间 (即决策变量定义域);

2. 分析的时间跨度从 2010 年开始，以 5 年为步长直至 2030 年，其中拥堵收费政策执行于 2020 年；
3. 首先从收敛速度、收敛状态、和求解质量三方面评估主动学习算法的优化效率，并为了使得结果更直观暂时仅考虑总行程时间和平均区域可达性 2 类目标：
 - 随机选择 5 种收费水平作为初始采样点开始贝叶斯优化，经过 10 轮迭代后 Pareto 前沿基本收敛，收敛速度非常理想；
 - 收敛状态由超体积的提升进行量化，自第 7 轮迭代后超体积的提升就已经非常微小；
 - 超体积提升量也可用于评价求解质量，因其反映了新的 Pareto 前沿相较于已有的 Pareto 前沿的提升。
4. 当考虑全部 3 类目标时，同样选择 5 个初始采样点，则经过 20 轮迭代后实现收敛。

8.1.5 英汉互译

English	Chinese	English	Chinese	English	Chinese
entrant	新生、新职员	retail	零售	real estate	房地产
premise	假定 (n,v)	equilibrate	(使)平衡	whilst	(conj) 同时
bypass	旁路、绕过 (v)	scalar	标量	cordon	警戒线
cruciform	十字形的	parenthesis	圆括号		

8.2 A novel mobility consumption theory for road user charging (TRB, 2024)

ABSTRACT

通过将交通出行类比为电能，基于预留车头间距的概念提出了一种新型的出行消耗理论 (mobility consumption theory)。理论认为出行是道路基础设施的“产品”，并被驾驶员“消耗”，与电能由发电站生产并被电子设备消耗类似。所提模型只需要行程时间和距离两个参数作为输入，并包含车长和反应时间两个物理参数。因为从时间和空间两个维度捕捉了驾驶员对道路资源的占用，该理论优于仅基于行程距离或时间的方法。研究基于该理论设计了一种新的交通收费策略。目

8.2.1 Introduction

1. 道路收费的初衷是补贴基础设施建设、养护和公交系统，但近年来因其有助于引导交通需求、改善拥堵、减少排放而引起更多关注；
2. 交通经济学家主要关注最优收费 (first-best pricing)，通过收费补贴私家车出行对系统产生的边际成本。实际应用只能采取次优收费 (second-best pricing)，主要以区域或距离为收费标准。区域收费主要为拥堵收费，研究则关注后者；
3. 基于行程距离的收费因其考虑了交通使用者对交通资源占用的差异而愈发被关注。传统上各国政府对出行者的收费主要通过车辆年费和燃油税实现。前者不能反映出行者对交通资源的占用，而后者因电动车占用率的上升在经济上不可持续；
4. 然而基于行程距离的收费策略忽略了道路使用者在时间层面的占用。其可能引导驾驶员抄近路从而加剧中心城区拥堵，且无法引导驾驶员错峰出行。因此，研究提出了基于出行者道路占用时间和距离的综合收费方法，并基于“消耗”的概念反映单位时间内对道路空间的占用。已有部分研究考虑了基于行程距离和时间的收费策略，但从未从有限道路资源被占用的角度提出合理的量化指标和收费标准；
5. 所提的基于出行消耗理论的收费策略具有如下优点：
 - 同时考虑了出行对道路时空资源的占用；
 - 仅基于行程时间和里程收费，易于核查；
 - 可增加拥堵区域的行驶成本，从而实现更短路径与更快路径的平衡；
 - 便于为不同车辆类型设置不同的收费费率。
6. 研究的创新点在于建立了交通和能源系统的类比关系，并参考能源消耗 (千瓦时) 的概念采用“千米时”量化出行对道路资源的占用。已有研究曾将交通类比为物理系统，但从能源系统角度出发尚属首次。

8.2.2 An analogy between transport and electricity systems

1. 本节主要讨论交通系统和电力系统的相似性，但并不表明两系统在物理上具有一致性或表现机制一致；
2. 首先讨论“能源”与“出行”在概念上的相似性：
 - 能源定义为工作的能力。端用户消耗电能为设备运作提供能源。功率的度量为瓦特，表示电子设备运行时消耗能源的速率。能源消耗的度量为千瓦时。能源供应商基于高峰和非高峰时段的能源消耗对用户收费；
 - 类似地出行可定义为运动的能力。端用户消耗出行以服务交通需求。道路空间的度量为米，可表示为道路使用者消耗“出行”的速率。例如自由流下安全车头间距约 30 米，表示车辆单位时间内消耗了 30 米的道路资源。因此可以以“千米时”度量出行消耗，并基于此由道路运营商对出行者收费。
3. 进一步从供给侧讨论电力系统和交通系统的相似性：
 - 电能由发电站提供。设计供电能力定义为发电站能产生的最大电能，单位为千瓦。发电量定义为发电站于一年内的总能源生产量，单位为千瓦时。发电站往往存在基准荷载发电机 (**baseload power generator**) 和高峰发电机 (**peaking power generator**) 两部分，前者提供稳定的能源供应且不能随时关停，后者则根据需求灵活启动；
 - 出行资源由道路基础设施提供，可定义设计出行能力 (**installed mobility capacity**) 为道路所具有的最大出行空间，单位为千米。出行资源生产量即可定义为一天内道路设施所提供的总出行空间，单位为千米时。道路设施同样存在由一般车道表示的基准出行资源供应和由动态车道或潮汐车道表示的临时出行资源供应。
4. 再从需求侧讨论电力系统和交通系统的相似性：
 - 能源消耗一般逐日变化。典型工作日的能源消耗高峰一般为傍晚初夜时段。能源消耗超过供给将导致停电，而停电可通过多种需求管控方式避免；
 - 出行资源消耗同样逐日变化。典型工作日的出行需求消耗高峰为早晚通勤时段。出行资源消耗超过供给将导致拥堵溢流，而拥堵溢流同样可通过多种需求管控方式避免。
5. 最后从损耗的角度讨论电力系统和交通系统的相似性：
 - 在没有储能设备的前提下，未被使用的能源便被损耗。电子于导线中运动传输电能会产生功率传输损耗 (**power transmission loss**)。电器工作时也会产生功率低效损耗 (**power inefficiency loss**)，即只有部分被消耗的电能能用于电器运作，其余将转化为热能损失（热力学第二定律）；
 - 未被使用的出行资源同样会被损耗，且无法被储存。车辆于路网中运动实现出行同样会因延误产生传输损耗。另外交通系统中也存在低效损耗，即存在部分被消耗的道路出行资源无法用于满足交通需求，如空载的出租车等。

8.2.3 A novel mobility consumption theory & Application to road user charging

1. 基于安全车头间距提出出行消耗理论以描述驾驶员对道路资源的使用。记 $x_n(t)$ 表示车辆 n 的时空轨迹曲线， $h_n(t)$ 表示车辆 n 于 t 时刻的安全车头间距。安全车头间距与车速 $v_n(t)$ 正相关，采用简单的线性公式

$$h_n(t) = \lambda_n + \tau_n v_n(t), \quad \lambda_n, \tau_n > 0$$

式中 λ_n 表示车辆 n 的最小车头间距（即排队停车时的车头间距）； τ_n 表示车辆 n 的最小反应时间。上述线性关系被应用于 Newell 简化跟驰模型（对应三角基本图），也可采用 IDM 或 Gibbs 等更复杂的跟驰模型的安全车头间距公式，但研究发现简单的线性关系式即可得到合理的出行资源消耗表征；

2. 安全车头间距表示车辆于某一时刻所需预留的道路空间资源，沿行程时间积分即得到整个行程中车辆 n 的出行资源消耗 M_n

$$M_n = \int_{t_n^0}^{t_n^0 + T_n} h_n(t) dt = \int_{t_n^0}^{t_n^0 + T_n} \lambda_n + \tau_n v_n(t) dt = \lambda_n T_n + \tau_n D_n$$

式中 T_n, D_n 表示行程的总时间和路程。注意到 $\lambda_n T_n$ 和 $\tau_n D_n$ 的量纲均为千米时，则出行资源消耗 M_n 的量纲也为千米时；

3. 出行资源消耗的概念也可应用于描述一个时空区域内的整体交通状态。如叠加路网所有车辆的总行程里程和总行程时间可得到交通系统的道路空间资源总消耗 $\sum_n M_n$:

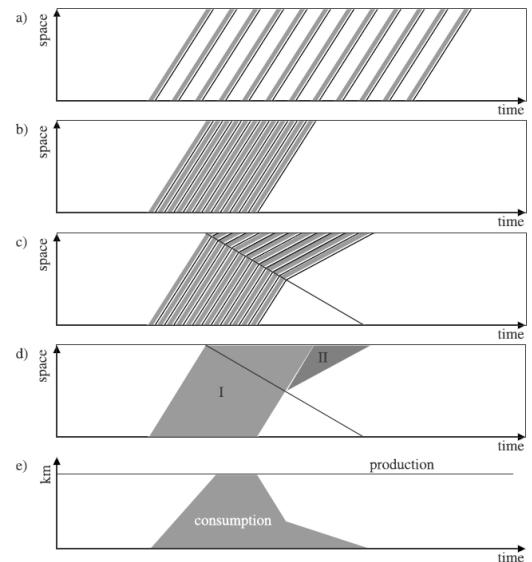


图 8.2 基于时空轨迹图表示的出行资源消耗，灰色阴影区域表示出行资源的时空占用。当车辆进入拥堵排队状态后，虽然每辆车的安全车头间距因车速降低和缩短，但因为行程时间的延长总的出行资源消耗增加（图(c)、(d)），并在一段时间内占据了路段的所有出行资源（图(e)）。

4. 进一步介绍模型参数 λ_n, τ_n 取值。根据参数的实际物理意义， λ_n 一般取 0.010-0.012km， τ_n 一般取 1/3600-2/3600h。对于具体场景可结合实际数据标定；
 5. 参照电价基于用电量（千瓦时）收费，可基于出行资源消耗（千米时）进行出行收费。记费率为 c_n ，则车辆 n 收费 $c_n M_n$

$$c_n M_n = \mu_n T_n + \delta_n D_n, \quad \mu_n = c_n \lambda_n, \quad \delta_n = c_n \tau_n$$

式中 μ_n, δ_n 分别表示针对行程时间和里程的费率。

8.2.4 Impact of mobility-based charging on travel behaviour and congestion

- 分析所提收费策略对于行为选择和交通状态的影响。考虑仅基于距离的收费 ($\mu_n = 0, \delta_n > 0$) 和无收费 ($\mu_n = \delta_n = 0$) 作为对照。假设存在 N 辆同质车，并假设收费不影响需求总量；
- 首先考虑基于出行的收费策略对出行时间选择的影响。聚焦一个简单的单路径场景，令路径长度为 L ，自由流速度为 V ，路径末端存在固定瓶颈，瓶颈通行能力为 s ，所有出行者的预期到达时间为 t^* 。记 t 时刻出发的行程时间为 $T(t)$ ，则对应广义行程成本 $C(t)$ 表示为

$$C(t) = \alpha T(t) + \beta \max\{0, t^* - t - T(t)\} + \gamma \max\{0, t + T(t) - t^*\} + cM(t)$$

式中前三项分别为出行者关于行程时间、提前到达、和延迟到达的感知成本，最后一项为外部收费。根据实测数据观测，一般有 $\gamma > \alpha > \beta$ 。根据瓶颈模型，行程时间 $T(t)$ 满足

$$T(t) = \frac{L}{V} + \frac{Q(t)}{s}$$

其中 $Q(t)$ 表示 t 时刻出发所需经历的排队长度。将上式和收费 $cM(t) = \mu T(t) + \delta L$ 代入 $C(t)$ 定义式有

$$C(t) = (\alpha + \mu) \frac{Q(t)}{s} + \beta \max \left\{ 0, t^* - t - \frac{L}{V} - \frac{Q(t)}{s} \right\} + \gamma \max \left\{ 0, t + \frac{L}{V} + \frac{Q(t)}{s} - t^* \right\} + C^0, \quad C^0 = (\alpha + \mu) \frac{L}{V} + \delta L$$

式中 C^0 表示最小广义出行成本。进一步消去上式的 max。令 t', t'' 分别表示首位和末位出行者的出发时间， \tilde{t} 表示可以及时抵达的临界出发时间（晚于 \tilde{t} 出发会迟到，早于 \tilde{t} 出发则会提前到达），则 $C(t)$ 可改写为如下分段函数的形式

$$C(t) = C^0 + (\alpha + \mu) \frac{Q(t)}{s} + \begin{cases} \beta \left(t^* - t - \frac{L}{V} - \frac{Q(t)}{s} \right), & t \in [t', \tilde{t}] \\ \gamma \left(t + \frac{L}{V} + \frac{Q(t)}{s} - t^* \right), & t \in [\tilde{t}, t''] \end{cases}$$

当在出发时间选择下系统达到用户均衡时出行者无法再通过改变出发时间降低出行成本，即 $C(t)$ 为常数，有 $\frac{dC(t)}{dt} = 0$ 。代入上式有

$$\frac{dQ(t)}{dt} = \frac{\beta s}{\alpha + \mu - \beta}, \quad t \in [t', \tilde{t}], \quad \frac{dQ(t)}{dt} = -\frac{\gamma s}{\alpha + \mu + \gamma}, \quad t \in [\tilde{t}, t'']$$

记出行者出发率为 $r(t)$ （即出发曲线的斜率，而离去曲线的斜率最大为 s ），则排队曲线 $Q(t)$ 与 $r(t)$ 恒有

$$Q(t) = \int_{\hat{t}}^t r(u)du - s(t - \hat{t}) \implies \frac{dQ(t)}{dt} = r(t) - s$$

式中 \hat{t} 表示无需排队的临界出发时间（晚于 \hat{t} 即需要排队）。代入上式得到用户均衡状态下的均衡出发率，并基于 $\int r(t)dt = N$, $C(t') = C(t'')$ 和 $t'' - t' = \frac{N}{s}$ 可反解 t', \tilde{t}, t''

$$r(t) = \begin{cases} \frac{(\alpha + \mu)s}{\alpha + \mu - \beta}, & t \in [t', \tilde{t}] \\ \frac{(\alpha + \mu)s}{\alpha + \mu + \gamma}, & t \in [\tilde{t}, t''] \end{cases}, \quad \begin{bmatrix} t' \\ \tilde{t} \\ t'' \end{bmatrix} = t^* - \frac{L}{V} + \begin{bmatrix} -\frac{\gamma}{\beta + \gamma} \cdot \frac{N}{s} \\ -\frac{\beta}{\alpha + \mu} \cdot \frac{\gamma}{\beta + \gamma} \cdot \frac{N}{s} \\ \frac{\gamma}{\beta + \gamma} \cdot \frac{N}{s} \end{bmatrix}$$

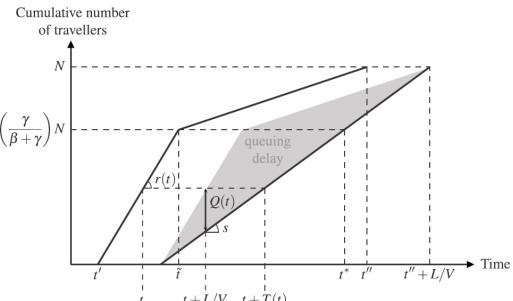
基于出发-离去曲线可计算系统的总延误 H （单位为车·小时），进一步按定义得到系统的对道路时空资源的总消耗 $\int M(t)r(t)dt$

$$H = \frac{1}{2} \left(\frac{\beta}{\alpha + \mu} \right) \left(\frac{\gamma}{\beta + \gamma} \right) \left(\frac{N^2}{s} \right), \quad \int M(t)r(t)dt = \lambda \left(\frac{NL}{V} + H \right) + \tau NL$$

注意到所提道路时空资源总消耗指标与系统总延误成正比，表明以此为收费标准可反映出行对交通系统的影响；

图 8.3 瓶颈模型用户均衡下的考虑出行收费的累计出发-到达曲线。

阴影部分总面积为系统总延误 H ，可根据几何关系计算。可以发现，因为所提收费标准考虑了对出行时间的收费费率 μ ，使得实际出发率 $r(t)$ 对行程时间更敏感，从而降低了系统的总延误和对道路时空资源的总消耗。且当 $\mu \rightarrow \infty$ 时有 $H \rightarrow 0$ 。另外关于距离的费率 δ 对出发率和总延误无影响，表明仅基于里程的收费无法解决此类拥堵。



3. 进一步考虑基于出行的收费策略对路径选择的影响。聚焦一个简单的双路径场景，令路径长度 $L_1 < L_2$ ，自由流速度 $V_1 < V_2$ ，通行能力 $s_1 < s_2$ 。不考虑出发时间选择，假设 N 位出行者于 $[t', t'']$ 时段内均匀出发，即 $r^* = \frac{N}{t'' - t'}$ ，且假设 $s_1 < r^* < s_2$ ，即拥堵仅可能发生于路径 1。则得到两条路径的广义出行成本为

$$C_i(t) = \alpha T_i(t) + cM_i(t) = (\alpha + \mu)T_i(t) + \delta L_i, \quad T_1(t) = \frac{L_1}{V_1} + \frac{Q_1(t)}{s_1}, \quad T_2(t) = \frac{L_2}{V_2} \\ \implies C_1(t) = (\alpha + \mu) \frac{Q_1(t)}{s_1} + C_1^0, \quad C_2(t) = C_2^0, \quad C_i^0 = (\alpha + \mu) \frac{L_i}{V_i} + \delta L_i$$

令路径 i 的出发率为 $r_i(t)$ ，显然有 $r_1(t) + r_2(t) = r^*$ ，因此只需重点关注 $r_1(t)$ 的特征即可反推 $r_2(t)$ 。取决于具体的路径参数取值，用户均衡状态下的需求分布具有三种形式：

- 所有需求分布于路径 2，即 $r_1(t) = 0$ 。意味着即使在自由流状态下路径 1 的综合出行成本仍不低于路径 2。为此定义 \tilde{T} 表示路径 1 相对于路径 2 的初始行程成本优势

$$\tilde{T} = \frac{C_2^0 - C_1^0}{\alpha + \mu} = \frac{L_2}{V_2} - \frac{L_1}{V_1} + \left(\frac{\delta}{\alpha + \mu} \right) (L_2 - L_1)$$

\tilde{T} 的量纲为时间。显然该形式等价于 $\tilde{T} \leq 0$ ；

- 所有需求分布于路径 1，即 $r_1(t) = r^*$ 。意味着即使在最拥堵状态下路径 1 的综合出行成本仍不高于路径 2，等价于

$$(\alpha + \mu) \frac{Q_1^{\max}}{s_1} + C_1^0 \leq C_2^0 \implies \tilde{T} \geq \frac{Q_1^{\max}}{s_1} = \frac{\int_{t'}^{t''} r^* du - s_1(t'' - t')}{s_1} = \left(\frac{r^* - s_1}{s_1} \right) \frac{N}{r^*}$$

- 需求分布于路径 1 和 2，等价于 $0 < \tilde{T} < \left(\frac{r^* - s_1}{s_1} \right) \frac{N}{r^*}$ 。此时 $r_1(t)$ 有

$$r_1(t) = \begin{cases} r^*, & t \in [t', \tilde{t}] \\ s_1, & t \in [\tilde{t}, t''] \end{cases}$$

$\tilde{t} \in [t', t'']$ 表示路径 2 首次被使用的时间，等价于路径 1 的广义路阻等于路径 2 的时间，即

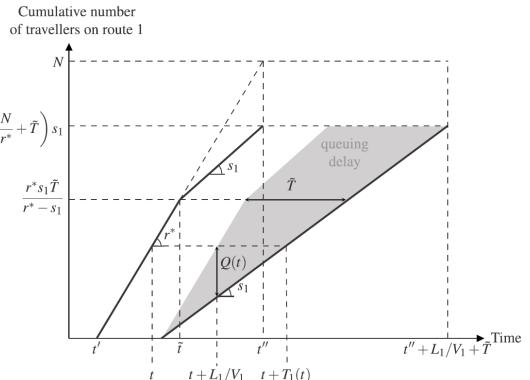
$$\tilde{T} = \frac{Q_1(\tilde{t})}{s_1} = \frac{\int_{t'}^{\tilde{t}} r^* du - s_1(\tilde{t} - t')}{s_1} = \frac{(r^* - s_1)(\tilde{t} - t')}{s_1} \implies \tilde{t} = t' + \frac{s_1 \tilde{T}}{r^* - s_1}$$

因为路径 2 不存在拥堵，只需基于路径 1 的累积出发-到达曲线即可得到系统的总延误 H ，综合考虑以上三种情况有

$$H = \begin{cases} 0, & \tilde{T} \leq 0 \\ \left(\frac{N}{r^*} + \frac{(r^* - 2s_1)\tilde{T}}{2(r^* - s_1)} \right) \tilde{T}s_1, & 0 < \tilde{T} < \left(\frac{r^* - s_1}{s_1} \right) \frac{N}{r^*} \\ \left(\frac{r^* - s_1}{2r^* s_1} \right) N^2, & \tilde{T} \geq \left(\frac{r^* - s_1}{s_1} \right) \frac{N}{r^*} \end{cases}$$

图 8.4 路径选择用户均衡下的考虑出行收费的累计出发-到达曲线。

阴影部分总面积为系统总延误 H ，可根据几何关系计算。可以发现，与无收费相比，考虑距离的收费策略加剧了此场景的系统延误。若仅基于行程距离收费会放大短路线的吸引力，从而加剧其拥堵。而所提的基于出行的收费策略因额外考虑了时间维度，在一定程度上缓解了系统延误加剧的问题。当 $\mu \rightarrow \infty$ 时系统总延误将收敛至无收费的程度。



8.2.5 Technology innovation's impact on mobility consumption

- 进一步讨论所提出行消耗理论和对应的收费策略在交通智能网联化和电气化发展前景下的潜在应用；
- 理论上，智能网联车具有更小的安全车头间距和更短的反应时间，因此在纯智能网联环境中可更高效地利用道路时空资源。而在现阶段的混合交通环境中，最新研究发现智能网联车倾向于更保守的驾驶风格，以更长的安全车头间距和反应时间追求安全性，因此现阶段的智能网联车反而会增大道路时空资源消耗。所提的出行资源消耗指标 M_n 的参数 λ_n, τ_n 具有明确的物理意义（安全车头间距和反应时间），只需基于实际环境为智能网联车确定合适的 λ_n, τ_n 取值即可精确捕捉其的交通资源占用；
- 在电气化交通发展的趋势下，基于出行消耗理论的收费策略也能基于电动汽车对道路时空资源的消耗而为政府收取资金，补充燃油费的缺口。同时为引导绿色交通系统的发展，还可扩展出行消耗理论以考虑如污染等除时空占用以外的额外交通成本。令 $e_n(t)$ 表示车辆 n 的单位时间二氧化碳排放量，研究发现其与速度呈正比

$$e_n(t) = \eta_n^0 + \eta_n^1 v_n(t)$$

式中 η_n^0, η_n^1 由车辆类型决定, 对于电动汽车有 $\eta_n^0 = \eta_n^1 = 0$ 。对 $e_n(t)$ 沿时间积分即可得到车辆行程中的总二氧化碳排放

$$E_n(t) = \int_{t_n^0}^{t_n^0 + T_n} e_n(t) dt = \eta_n^0 T_n + \eta_n^1 D_n$$

与出行资源消耗指标 M_n 的形式完全一致。记 ω_n 为关于碳排放的费率, 则可基于 M_n 和 E_n 实现出行-碳排放联合收费

$$c_n M_n + \omega_n E_n = (c_n \lambda_n + \omega_n \eta_n^0) T_n + (c_n \tau_n + \omega_n \eta_n^1) D_n = \mu_n^* T_n + \delta_n^* D_n$$

8.2.6 英汉互译

English	Chinese	English	Chinese	English	Chinese
power plant	发电厂	odometer	里程表	baseload	基本负荷
diesel	柴油、柴油机	blackout	断电 (n)	load shedding	甩负荷 (n)
pumped hydro	抽水蓄能	incandescent	白炽的、耀眼的	parsimonious	小气的
bumper	保险杠	excise	消费税、切除 (v)	internalize	使内在化
inelastic	无弹性的	whereby	由此	tailpipe	车辆排气管

赌书消得泼茶香 当时只道是寻常

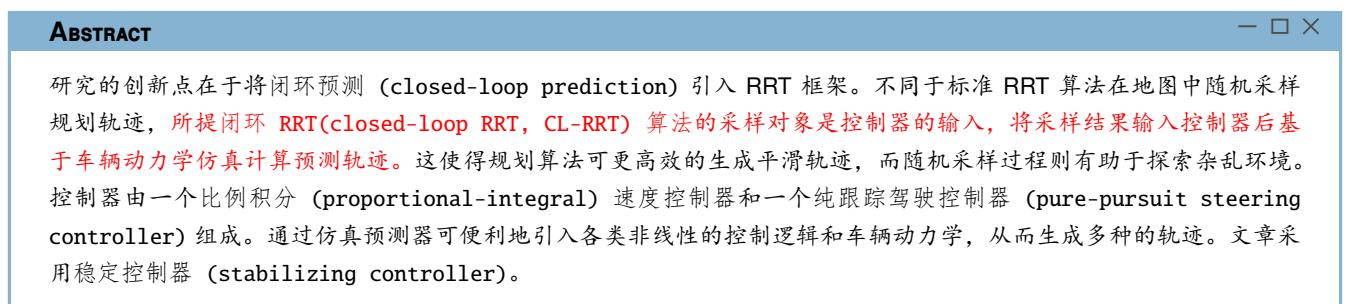
第三部分

自动驾驶 (*Autonomous Driving*)

第9章

轨迹规划

9.1 Motion planning in complex environments using closed-loop prediction (2008)



9.1.1 Introduction

- 实现城市环境中的自动驾驶需要更精细、复杂的规划系统。一方面要求适应动态且不可预测的驾驶环境；另一方面还需应对如连续弯路 (winding road)、部分封闭道路、停车区等等的复杂场景；此外交通规则和交通状态也使得车辆轨迹规划应同时考虑车辆实时和历史状态的影响；
- 因为自动驾驶的需求，轨迹规划需要实时进行。而在复杂动态和约束下的车辆实时运动规划仍是一个挑战。例如基于网格 (grid-based) 的搜索算法如 A*、D*、E* 等可高效搜索全局方案，但难以考虑车辆的非线性动力特征和严格的行为约束；
- 本文提出了一种基于仿真的闭环框架，规划器基于车辆模型和底层控制器生成车辆轨迹。与可能失稳的开环系统相比，闭环系统中规划器可聚焦更长时段内的行为并高效生成轨迹。具体地，规划器生成的轨迹作为底层控制器的输入，而控制器采用闭环设计从而保证车辆运动的稳定性，仿真则用于预测车辆的实际轨迹并判断其是否满足障碍物避让等额外约束。通过仿真可使得规划的轨迹符合车辆动力学，且便于引入非线性动力特征、非线性控制器、或输入饱和 (input saturation)。将所构建的闭环仿真嵌入 RRT 框架（见第 19.7 节），与标准 RRT 算法相比可更高效地生成平滑轨迹。

9.1.2 Overview of the approach

- 记车辆的非线性动力模型为 $\dot{x} = f(x, u)$ ，其中 x 表示状态（即实际轨迹）而 u 为车辆模型的输入。将控制饱和度、制动延迟、速度上限等约束建模为 $x \in X, u \in U$ 。而将对于轨迹的约束（如车道线约束、避障等）建模为 $x(t) \in X_{free}(t)$ 。则本研究关注的运动规划问题建模为：给定当前状态 $x(t_0) \in X \cap X_{free}(t_0)$ ，生成车辆轨迹和相应车辆模型输入 $x(t), u(t), t \in [t_0, t_f]$ ，使得车辆在满足约束 $x(t) \in X \cap X_{free}(t), u(t) \in U$ 的情况下到达目标点 $X_{goal} \subseteq X$ ；
- 研究仅考虑小汽车，但可迁移至其它任意载具；

3. 进一步介绍控制系统细节。系统自上而下由规划器、控制器、车辆模型（用于仿真预测）组成：

- 规划器的输入为短期内的目标 X_{goal} , 以 10Hz 的频率运行, 即每 0.1s 规划一次轨迹以应对不确定性交通环境。规划器的输出称为参考轨迹 (reference trajectory), 为若干随机采样点连成的折线段 (piece-wise linear), 记为 r ;
- 将 r 和相应的速度指令输入控制器, 则后者输出油门、刹车、转向等控制指令 u , 以 25Hz 的频率输入车辆模型 (即硬件);
- 车辆模型基于 u 得到实际轨迹 x , 要求 x 应较好的贴合参考轨迹 r 。所构建的控制系统为闭环式反馈系统, 即车辆模型的输出 x 会反馈回控制器。

4. 对比开环式控制系统 (即直接采样 u 输入车辆模型), 所搭建的闭环系统具有如下优势：

- 通过引入稳定控制器, 使得系统可应用于具有不稳定运动特征的载具如小汽车、直升机等;
- 稳定控制器可减小因车辆仿真误差导致的轨迹预测误差;
- 通过仿真可简单地考虑各种非线性车辆模型或控制器, 并得到相应的轨迹;
- 只需输入简单的 r 系统即可得到较长时段内的稳定、平滑的轨迹, 显著提升了以 RRT 算法为代表的随机方法的效率。因为此类方法难以仅凭随机采样得到高质量的车辆连续输入。

9.1.3 Controller

1. 控制器的两个核心模块为纯跟踪驾驶控制器 (pure-pursuit steering controller) 和比例积分速度控制器 (proportional-integral (PI) speed controller)。前者使得车辆尽可能沿规划的轨迹行驶, 而速度控制器则使车辆按规划的速度运行;
2. 控制器设计的有效考量为对延误、噪声、误差等的鲁棒性, 因此可在一定程度上牺牲其对参考轨迹 r 的服从度。但因为规划算法的效果是通过实际轨迹 x 评估的, 若 x 与 r 偏差过大则难以通过误差反馈优化规划器的输出 r ;
3. 首先介绍纯跟踪驾驶控制器。纯跟踪控制旨在控制机器人沿特定轨迹运动, 因其简单、几何直观等特点广泛应用于地面机器人运动设计, 近期也开始被应用于无人机领域;
4. 不考虑侧滑等动力学特征, 研究基于运动学 (kinematic) 模型设计纯跟踪驾驶控制器, 从而将问题简单转化为几何问题。因为忽略受力作用, 运动学模型仅适用于低速情况下车轮无滑动摩擦的场景。故仅在控制器设计时采用运动学模型, 而仿真预测中的车辆模型则会考虑动力学特征;
5. 记 x, y 分别表示车辆后轴 (rear axle) 的位置; θ 表示车辆航向 (即后轮方向) 与 x 轴的夹角 (逆时针为正); v 表示前向速度; δ 表示前轮转向角度 (与车辆航向的夹角, 逆时针为正); L 表示前后轴轴距 (wheelbase)。则车辆的运动学自行车模型 (kinematic bicycle model) 建模为

$$\dot{x} = v \cos \theta, \quad \dot{y} = v \sin \theta, \quad \dot{\theta} = \frac{v}{L} \tan \delta \quad (\text{运动学自行车模型})$$

“自行车模型”是一种经典的车辆简化建模方法。模型假设：1) 车辆前轮驱动；2) 左右轮转向一致；3) 车身为刚体；4) 车辆在平面内运动。基于上述假设即可将四轮车辆简化成前后两轮的形如自行车的形式, 从而得名“自行车模型”。上式的前两个等式关系易证, 简单证明车辆航向 θ 与前轮转向 δ 的关系。易知前轮方向与 x 轴的夹角为 $\theta + \delta$, 则对前轮有

$$\frac{\dot{y}_f}{\dot{x}_f} = \frac{d(y + L \sin \theta)/dt}{d(x + L \cos \theta)/dt} = \frac{\dot{y} + L \cos \theta \cdot \dot{\theta}}{\dot{x} - L \sin \theta \cdot \dot{\theta}} = \tan(\theta + \delta) \implies \frac{v \sin \theta + L \cos \theta \cdot \dot{\theta}}{v \cos \theta - L \sin \theta \cdot \dot{\theta}} = \frac{\tan \theta + \tan \delta}{1 - \tan \theta \tan \delta} \implies \dot{\theta} = \frac{v}{L} \tan \delta$$

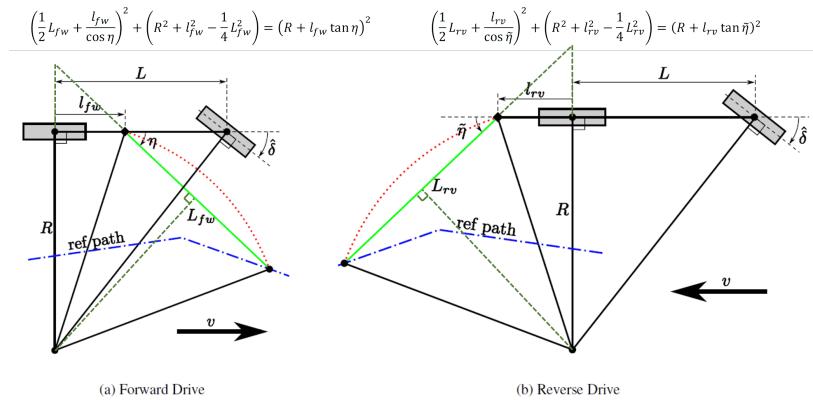
式中 x_f, y_f 分别表示车辆前轴的位置。

6. 进一步介绍基于运动自行车模型的改进纯跟踪控制算法, 算法的目标是确定合适的前轮转角 δ 使得车辆的某处固定参考点跟踪参考轨迹。考虑前行和倒车两种场景。记 R 为后轮转弯半径; $l_{fw}, l_{rv} \geq 0$ 分别为前行和倒车时车辆后轴距固定参考点的距离 (以运动方向为正方向); L_{fw}, L_{rv} 分别为前行和倒车时车辆固定参考点距参考路径目标点的直线距离; $\eta, \tilde{\eta}$ 分别为前行和倒车时车辆固定参考点与参考路径目标点连线与车辆运动方向的夹角。为使得车辆参考点到达目标点, 易知 $-\tan \delta = \frac{L}{R}$, 再基于勾股关系有

$$\delta = -\tan^{-1} \left(\frac{L \sin \eta}{L_{fw}/2 + l_{fw} \cos \eta} \right), \quad \delta = -\tan^{-1} \left(\frac{L \sin \tilde{\eta}}{L_{rv}/2 + l_{rv} \cos \tilde{\eta}} \right) \quad (\text{改进纯跟踪控制})$$

上式中当 $l_{fw} = l_{rv} = 0$ 时即为标准的纯跟踪控制算法，表示以后轴为车辆参考点跟踪参考路径。此处令 $l_{fw}, l_{rv} > 0$ 可使得控制器具有更高的稳定性；

图 9.1 基于运动自行车模型的改进纯跟踪控制算法。左、右图分别表示前行和倒车的情况。每图中的两个矩形分别表示自行车模型简化的车辆前轮和后轮。蓝色虚线为输入的参考轨迹；红色虚线为跟踪参考轨迹目标点而形成的车辆参考点的轨迹；绿色虚线为辅助线。基于勾股定理可得到后轮转弯半径 R 与 η, l_{fw}, L_{fw} （或 $\tilde{\eta}, l_{rv}, L_{rv}$ ）的关系。车辆前轮转角 $\hat{\delta} = -\delta$ 。



7. 另外再考虑关于前轮转向速度 $\dot{\delta}$ 的约束。为便于稳定性分析将最大转速约束 $\dot{\delta} \leq \dot{\delta}_{\max}$ 近似为

$$\dot{\delta} = \frac{1}{\tau}(-\delta + \delta_c)$$

式中 δ 表示当前的前轮转向； δ_c 为目標的前轮转向； τ 为执行时间常数，取前轮从零转向到最大转向所需时间的 90%；

8. 进一步基于李雅普诺夫第一法和劳斯-霍尔维茨稳定性判据 (Routh-Hurwitz criterion) 分析上述控制器的稳定性条件。首先考虑前进的情况，则目標转向 δ_c 有

$$\delta_c = -\tan^{-1} \left(\frac{L \sin \eta}{L_{fw}/2 + l_{fw} \cos \eta} \right), \quad \eta = \theta + \sin^{-1} \left(\frac{y + l_{fw} \sin \theta}{L_{fw}} \right)$$

上式中 η 的关系式为参考路径为与 x 轴重合时的结果。参考路径的具体方向和位置并不会影响控制器的稳定性分析结果，但作此假设可便于稳定性分析。令系统状态向量 $z = [y, \theta, \delta]^T$ ，则由 $\dot{z} = 0$ 可知系统的平衡状态为 $y = \theta = \delta = \delta_c = \eta = 0$ 。注意到关于状态向量 z 的状态方程为非线性，应用李雅普诺夫第一法（见第 22.8 节）分析系统于平衡状态处的稳定性需首先对其作线性化 ($\sin x \approx x, \tan x \approx x, \cos x \approx 1, \tan^{-1} x \approx x, \sin^{-1} x \approx x$)

$$\begin{aligned} \dot{y} &= v\theta, \quad \dot{\theta} = \frac{v\delta}{L}, \quad \dot{\delta} = -\frac{1}{\tau} \left(\delta + \frac{L}{L_{fw}/2 + l_{fw}} \left(\theta + \frac{y + l_{fw}\theta}{L_{fw}} \right) \right) = -\frac{\delta}{\tau} - \frac{(L_{fw} + l_{fw})L\theta}{\tau(L_{fw}/2 + l_{fw})L_{fw}} - \frac{Ly}{\tau(L_{fw}/2 + l_{fw})L_{fw}} \\ \Rightarrow \begin{bmatrix} \dot{y} \\ \dot{\theta} \\ \dot{\delta} \end{bmatrix} &= \begin{bmatrix} 0 & v & 0 \\ 0 & 0 & \frac{v}{L} \\ -\frac{L}{\tau(L_{fw}/2 + l_{fw})L_{fw}} & -\frac{(L_{fw} + l_{fw})L}{\tau(L_{fw}/2 + l_{fw})L_{fw}} & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} y \\ \theta \\ \delta \end{bmatrix} \end{aligned}$$

从而得到形如 $\dot{z} = Az$ 的线性状态方程。按李雅普诺夫第一法，原非线性系统于平衡状态邻域内渐进稳定的充要条件是线性化后的系统矩阵 A 的特征值均具有负实部。按 $\det(sI - A) = 0$ 得到 A 的特征方程有

$$s^3 + \frac{1}{\tau}s^2 + \frac{v(L_{fw} + l_{fw})}{\tau L_{fw}(L_{fw}/2 + l_{fw})}s + \frac{v^2}{\tau L_{fw}(L_{fw}/2 + l_{fw})} = 0$$

劳斯-霍尔维茨判据（见第 22.3 节）提供了一种直接由多项式的系数判断根的位置的方法。从而得到原系统的稳定性条件为

$$\frac{1}{\tau} > 0, \quad \frac{v(L_{fw} + l_{fw} - v\tau)}{\tau L_{fw}(L_{fw}/2 + l_{fw})} > 0, \quad \frac{v^2}{\tau L_{fw}(L_{fw}/2 + l_{fw})} > 0$$

由物理关系天然有 $\tau > 0, L_{fw} > 0, l_{fw} > 0$ ，且在前进的情况下有 $v > 0$ ，则得到最终的稳定性条件为

$$L_{fw} > v\tau - l_{fw} \quad (\text{前进稳定性条件})$$

上述稳定性条件要求为保证车辆稳定性，参考路径距车辆参考点的距离 L_{fw} 应随车辆速度 v 的增加而增加，而且对于响应越慢的制动系统 L_{fw} 也应该越大。另外可以看出与经典的纯跟踪模型 ($L_{fw} = 0$) 相比，所提改进纯跟踪模型设 $L_{fw} > 0$ 有助于提升系统稳定性。在倒车场景下有 $v < 0$ ，则类似地得到模型的倒车稳定性条件为

$$L_{rv} > -v\tau - l_{rv} \quad (\text{倒车稳定性条件})$$

9. 最后介绍为保证纯跟踪控制器稳定性的 L_{fw}, L_{rv} 的取值。研究将其设为车辆控制速度 v_{cmd} 而非实际速度 v 的函数。合理的 $L_{fw}(v_{cmd}), L_{rv}(v_{cmd})$ 应在留有稳定性富余的前提下尽可能小，过大的 L_{fw}, L_{rv} 会降低控制器跟踪参考轨迹的能力。令 $\dot{\delta}_{\max} = 0.406\text{rad/s}$ ，则有 $\tau = 0.717\text{s}$ ，则设

$$L_{fw}(v_{cmd}) = L_{rv}(v_{cmd}) = \begin{cases} 3 & v_{cmd} < 1.34 \text{ m/s} \\ 2.24 \cdot v_{cmd} & 1.34 \text{ m/s} \leq v_{cmd} < 5.36 \text{ m/s} \\ 12 \cdot v_{cmd} & 5.36 \text{ m/s} \leq v_{cmd} \end{cases}$$

10. 进一步介绍自动驾驶车的速度控制器。记车辆规划的指令速度为 v_{cmd} 、实际速度为 v ，则速度控制器基于反馈结构最小化 v 与 v_{cmd} 之间的误差。最常用的范围控制结构为比例-积分-微分 (PID) 结构，但测试发现测试车辆存在内生的速度阻尼 (speed damping)，使微分单元产生较大的噪声信号 (比例、积分、微分各控制单元的作用见第 22.2.1 节)。因此为加、减速设计相同的比例-积分 (PI) 反馈速度控制器

$$\frac{V(s)}{U(s)} = \frac{K_n(v)}{\tau_v s + 1}, \quad K_n(v) = 0.1013v^2 + 0.5788v + 49.1208, \quad (\text{PI 速度控制器})$$

$$u(t) = K_p(v_{cmd}(t) - v(t)) + K_i \int_0^t v_{cmd}(\tau) - v(\tau) d\tau$$

式中 K_p, K_i 分别表示比例增益和积分增益， $V(s), U(s)$ 分别表示 $v(t), u(t)$ 的拉氏变换。速度控制器的输入为实际速度 $v(t)$ 与 $v_{cmd}(t)$ 之间的误差信号 $v_{cmd}(t) - v(t)$ ，误差信号经比例单元和积分单元后得到无量纲的速度控制信号 $u(t)$ ，再经一个惯性环节 $\frac{K_n}{\tau_v s + 1}$ 输出实际速度 $v(t)$ 。 $\tau_v = 12\text{s}$ 为惯性系统的时间常数。惯性环节的惯性增益 K_n 为关于速度 v 的多项式，被称为线性变参数控制 (linear parameter-varying control) 或线性时变控制，在满足线性特性的同时系统特性随参数变化而变化。 $K_n(v)$ 的多项式系数由系统辨识 (system identification) 技术确定；

11. 基于比例、积分和惯性环节的传递函数即可得到 PI 速度控制器的闭环传递函数 $H(s)$ 和闭环特征方程 $p(s)$

$$H(s) = \frac{\left(K_p + \frac{K_i}{s}\right) \frac{K_n}{\tau_v s + 1}}{1 + \left(K_p + \frac{K_i}{s}\right) \frac{K_n}{\tau_v s + 1}} = \frac{K_n(K_p s + K_i)}{s(\tau_v s + 1) + (K_p s + K_i)K_n} \implies p(s) = \tau_v s^2 + (K_n K_p + 1)s + K_n K_i$$

反馈系统的开环传递函数和闭环传递函数之间的关系同样见第 22.2.1 节。确定比例增益和积分增益 K_p, K_i 的取值即可完成 PI 速度控制器的设计。研究采用参数空间法 (parameter space approach)。参数空间法是针对系统参数大范围变化时鲁棒控制系统设计的一种新方法。其基本思路是在复平面内根据各项要求预设一个区域，被称为 D 稳定区域 (D-stable region)，并要求系统的所有闭环极点均应落在该区域中，从而反解出为满足该约束系统参数 K_p, K_i 的取值范围，该范围即为参数空间。在实际应用中只需从该参数空间内任取一个点作为 K_p, K_i 的取值；

12. 考虑一个形如梯形的 D 稳定区域，并称其左边界、右边界、和上下边界分别为 $\partial_3, \partial_1, \partial_2$ ：

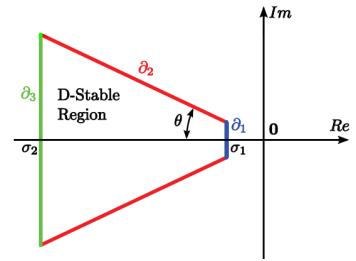
- 右边界 ∂_1 要求系统的所有闭环极点均具有不大于 $-\sigma_1 < 0$ 的负实部。一般地闭环极点实部小于 0 即可保证系统稳定，在此基础上负实部离虚轴越远则鲁棒性越强。为此只需将 $s' = s + \sigma_1$ 代入系统的闭环特征方程 $p(s)$ 得到

$$p(s') = \tau_v(s')^2 + (K_n K_p + 1 - 2\tau_v \sigma_1)s' + [\tau_v \sigma_1^2 - (K_n K_p + 1)\sigma_1 + K_n K_i]$$

要求 $p(s)$ 的全部零点实部不大于 $-\sigma_1$ 即等价于要求 $p(s')$ 的全部零点实部不大于 0，因此由劳斯-霍尔维茨判据 (见第 22.3 节) 易得 K_p, K_i 关于边界 ∂_1 的取值范围 $R_p(\partial_1)$

$$R_p(\partial_1) = \left\{ (K_p, K_i) \mid K_p \geq \frac{2\sigma_1 \tau_v - 1}{K_n}, \quad K_i \geq \frac{(1 + K_n K_p)\sigma_1 - \tau_v \sigma_1^2}{K_n} \right\}$$

图 9.2 用于 PI 速度控制器设计的 D 稳定区域。系统的所有闭环极点均应落入该区域内，从而反解出控制器参数 K_p, K_i 的取值范围。区域边界包含 $\partial_1, \partial_2, \partial_3$ 三部分。 $\sigma_1, \sigma_2, \theta > 0$ 均为预设参数。边界 ∂_1 是系统相对稳定性的最小边界；边界 ∂_2 是系统阻尼的最小边界（超出该边界会使得系统阻尼过小，产生的振荡过大）；边界 ∂_3 避免系统具有过大的闭环带宽。



- 左边界 ∂_3 要求系统的所有闭环极点均具有不小于 $-\sigma_2 < 0$ 的负实部。因为系统极点实部越小意味着性能越强，但设计难度也越大。与 $R_p(\partial_1)$ 同理得到 K_p, K_i 关于边界 ∂_3 的取值范围 $R_p(\partial_3)$ 有

$$R_p(\partial_3) = \left\{ (K_p, K_i) \mid K_p \leq \frac{2\sigma_2\tau_v - 1}{K_n}, \quad K_i \geq \frac{(1 + K_n K_p)\sigma_2 - \tau_v \sigma_2^2}{K_n} \right\}$$

- 上下边界 ∂_2 要求系统的所有闭环极点与复平面原点的连线与实轴的夹角不超过 θ 。因为极点距实轴越远，产生的响应信号的角频率（振荡）便越大。若极点距虚轴较近，则响应信号幅度衰减较慢，因此角频率不宜过大，极点需距离实轴较近；而当极点距虚轴较远，其响应信号幅度将快速衰减，故允许具有较大的角频率，极点可距实轴较远。为计算 $R_p(\partial_2)$ 需考虑两类条件。首先要求 $p(s)$ 的根位于虚轴左侧，按劳斯判据有

$$\begin{cases} K_n K_p + 1 \geq 0 \\ K_n K_i \geq 0 \end{cases} \implies K_p \geq -\frac{1}{K_n}, \quad K_i \geq 0$$

当 $p(s) = 0$ 有实根时该条件自然满足，故再考虑 $p(s) = 0$ 有虚根的情况，由二次方程求根公式有

$$\frac{4\tau_v K_n K_i - (1 + K_n K_p)^2}{(1 + K_n K_p)^2} \leq \tan^2 \theta \implies K_i \leq \frac{(1 + K_n K_p)^2}{4\tau_v K_n \cos^2 \theta}$$

综上得到 $R_p(\partial_2)$ 有

$$R_p(\partial_2) = \left\{ (K_p, K_i) \mid K_p \geq -\frac{1}{K_n}, \quad 0 \leq K_i \leq \frac{(1 + K_n K_p)^2}{4\tau_v K_n \cos^2 \theta} \right\}$$

上述关于极点位置对系统稳定性和响应特性的影响详见第 21.6 节拉普拉斯变换和第 21.7 节系统拉氏变换分析法的相关内容；

13. 另外再在惯性环节后考虑时延的情况。时延本身并不影响系统稳定性，但若反馈系统中存在时延则可能对其稳定性产生较大影响。假设系统延迟 T 秒，即

$$\frac{V(s)}{U(s)} = \frac{K_n(v)}{\tau_v s + 1} e^{-Ts}$$

相应的 PI 速度控制器开环传递函数 $G(s)$ 有

$$G(s) = \left(K_p + \frac{K_i}{s} \right) \frac{K_n}{\tau_v s + 1} e^{-Ts} = \frac{K_p K_n s + K_i K_n}{s} \frac{e^{-Ts}}{\tau_v s + 1}$$

为避免系统因时延而失稳，引入稳定裕度的概念（见第 22.4.2 节）。具体要求系统应具有 m_ϕ 的相位裕度 (phase margin)。按相位裕度的定义等价于令

$$G(i\omega_A) = e^{i(m_\phi - \pi)} = -\cos m_\phi - i \sin m_\phi$$

式中 $\omega_A > 0$ 为截止频率或增益交叉频率 (gain crossover frequency)，是使得系统开环频率特性函数模长 $|G(i\omega_A)| = 1$ 的频率。对上式等号两边同乘 $i\omega_A - \tau_v \omega_A^2$ (即消去 $G(i\omega_A)$ 的分母) 有

$$(K_p K_n i\omega_A + K_i K_n) (\cos T\omega_A - i \sin T\omega_A) = (i\omega_A - \tau_v \omega_A^2) (-\cos m_\phi - i \sin m_\phi)$$

$$\Rightarrow \begin{cases} K_i K_n \cos T\omega_A + K_p K_n \omega_A \sin T\omega_A = \tau_v \omega_A^2 \cos m_\phi + \omega_A \sin m_\phi \\ K_p K_n \omega_A \cos T\omega_A - K_i K_n \sin T\omega_A = \tau_v \omega_A^2 \sin m_\phi - \omega_A \cos m_\phi \end{cases}$$

将其视为关于 K_i, K_p 的线性方程组，则可得到在给定时延 T 和相位裕度 m_ϕ 需求下的 K_i, K_p 取值范围 $\partial_{m_\phi, T}$

$$\partial_{m_\phi, T} = \{(K_p, K_i) | K_p = f(m_\phi, T, \omega_A), K_i = g(m_\phi, T, \omega_A), \omega_A \in (0, \infty)\},$$

$$f(m_\phi, T, \omega_A) = \frac{\tau_v \omega_A \sin(m_\phi + T\omega_A) - \cos(m_\phi + T\omega_A)}{K_n}, \quad g(m_\phi, T, \omega_A) = \frac{\tau_v \omega_A^2 \cos(m_\phi + T\omega_A) + \omega_A \sin(m_\phi + T\omega_A)}{K_n}$$

14. 以上先后基于系统的闭环传递函数和开环频率特性函数确定系统参数 K_i, K_p 的参数空间。但上述过程无法直接反映系统的扰动拒绝、跟踪或噪声衰减特性。为此考虑参数空间法外的另一种鲁棒控制方法—— H_∞ 鲁棒控制确定新的 K_i, K_p 参数空间。

9.1.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
cluttered	杂乱的	steer	驾驶、操纵	pursuit	追求
impose	把…强加于	maneuver	演习、调动	winding	蜿蜒的
actuator	致动器	drivability	驾驶性	bandwidth	带宽
sideslip	侧滑 (n)	axle	车轴	wheelbase	轴距
nonholonomic	不完整的	slew	突然转向 (v)	sluggish	迟缓的
analogous	类似的	damping	阻尼	nondimensional	无量纲的
attenuation	衰减、稀释	complementary	互补的、补充的		

9.2 Generation of Reference Trajectories for Safe Trajectory Planning (ICANN, 2018)

ABSTRACT

- □ ×

大量面向运动规划的 RRT 变体算法采用有偏采样 (biased-sampling) 以加速收敛。混合增广 CL-RRT+ (hybrid-augmented CL-RRT+) 算法是近期提出的一种 RRT 变体，基于机器学习算法预测用于有偏采样的模板轨迹 (template trajectory)。然而因为预测的模板轨迹数量有限，只有当实际轨迹与预测的模板轨迹接近时才可取得较少的收敛时间。为此，本文基于变分自编码器生成大量参考轨迹 (reference trajectory)，并训练三维卷积网络模型预测关键交通环境下的参考轨迹。基于这一框架提出两种安全轨迹规划算法，并基于仿真验证多种关键交通场景下的算法有效性。

9.2.1 Introduction

- 为实现完全自动驾驶，车辆应具备同时规划横纵向运动以实现动态交通环境下的驾驶安全性和舒适性；
- 得益于计算效率和可在连续空间和动态约束下规划路径的优势，RRT 算法（见第 19.7 节）是最主流的一类运动规划算法，但已有的 RRT 变体普遍存在一些问题：
 - 一方面是只有少数变体具有在动态约束下实时运算的能力，但也需要再很多安全场景下进行预计算或需要高性能计算机；
 - 多数算法定义了基于规则的启发式有偏采样以提升收敛速度，但有偏采样需要预输入理想的初始解。
- 机器学习算法具有在短时间内求解复杂问题的能力，但因为其黑箱特性而不适用于如车辆轨迹规划等强调安全的场景。已有的尝试也仅是将机器学习算法用于有偏采样；
- 结合了机器学习算法和基于模型的搜索算法的混合机器学习算法 (hybrid machine learning algorithm) 为机器学习算法在安全关切领域的应用提供了新的思路。混合增广 CL-RRT (hybrid augmented CL-RRT, HARRT) 和混合增广 CL-RRT+ (hybrid augmented CL-RRT+, HARRT+) 是复杂、关键交通环境下规划安全轨迹的代表算法。此类算法结合了 RRT 变体和三维卷积模型；
- 本研究结合了变分自编码器（见第 24.9 节）和三维卷积模型两种机器学习算法以生成更好的参考轨迹。其中变分自编码器用于轨迹生成，而卷积模型则用于预测关键交通场景下的轨迹生成结果。在将此框架与两种运动规划算法结合以生成最终的规划路径。

9.2.2 Variational autoencoder & HARRT+ Algorithm

1.

9.2.3 英汉互译

English	Chinese	English	Chinese	English	Chinese
immense	巨大的				

赌书消得泼茶香 当时只道是寻常

第四部分

交通仿真

第 10 章

静态需求估计与交通分配

10.1 Modeling capacity flexibility of transportation networks (TRA, 2011)

ABSTRACT

交通系统的弹性 (flexibility) 是研究需求变化时的一个重要指标。本文基于双层网络通行能力模型提出了两种定量评价交通网络通行能力弹性的方法：

- 方法 1 基于备用通行能力 (reserve capacity) 的概念，仅反映了通行能力对需求量变化的弹性；
- 方法 2 除了考虑需求量的变化，还考虑了需求模式 (demand pattern) 的变化。

本文基于方法 2 提出两种通行能力弹性模型，分别考虑两类弹性概念：

- 总体弹性 (total flexibility)：允许用户同时进行路径选择和目的地选择，对应极限通行能力 (ultimate capacity) 的概念；
- 有限弹性 (limited flexibility)：在当前需求模式基础上所能新增的需求量，允许新增需求服从新的需求模式，对应实际通行能力 (practical capacity) 的概念。

10.1.1 Introduction

1. 弹性描述了系统应对不确定性的能力。在交通领域，弹性是描述交通系统应对由多种原因导致的需求变化能力的重要指标。需求变化的主要原因为：

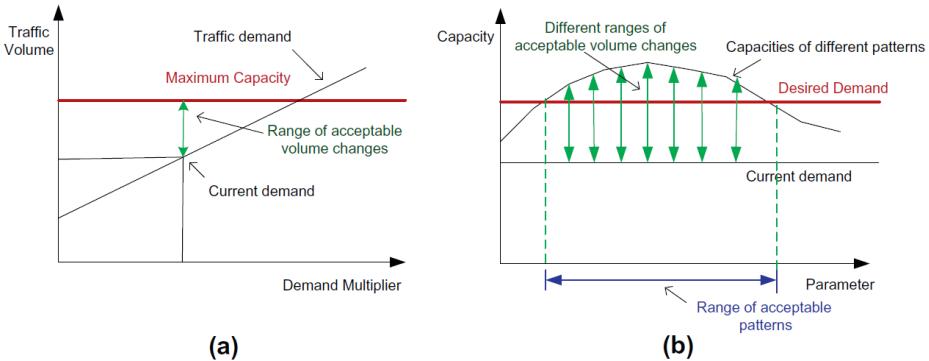
- 随着经济发展而基础设施服务水平相对不变带来的需求量的持续上升；
- 由土地利用政策或偶发事件等外部干涉导致的出行模式转变。

因此，交通系统需要确保充分的通行能力以适应交通需求的变化；

2. 具体地对网络弹性具有多种定义方法：

- Feitelson and Salomon (2000) 提出了一种主观定性评价方法，将网络弹性定义为网络适应不断变化的环境和需求的容易程度，包括设施和运行层面的变化。网络弹性包括节点弹性 (node flexibility)、连边弹性 (link flexibility) 和时间弹性 (temporal flexibility) 的概念：
 - 节点弹性指网络节点位置选择的容易程度 (ease)；
 - 连边弹性由在现有网络基础上新增连边的容易程度和代价 (cost) 决定；
 - 时间弹性指对基础设施投资进行排序的能力和使用基础设施时需要用户之间协调的程度。
- Cho (2002) 指出交通弹性可视为包含运行状况、损失和网络设计等参数的概念，可由网络通行能力、OD 连通性和行程时间等评价；
- Morlok and Chang (2004) 将通行能力弹性定义为交通系统适应交通需求变化并保持可接受的表现水平的能力，可由最大通行能力或系统最大承载交通量量化。此时弹性有两种定义方式：
 - 需求模式固定情况下系统可处理的需求量的范围；
 - 运行突破基本需求模式情况下系统可处理的需求量的范围。

图 10.1 两种 capacity flexibility 定义: (a) 仅考虑需求量变化的备用通行能力 (reserve capacity); (b) 同时考虑需求量变化和需求模式变化的实际通行能力 (practical capacity)



- Sun et al. (2006) 进一步对上述工作进行以下拓展,有助于评价考虑需求变化的退化交通系统 (degradable transportation system)¹:
 - 考虑未来交通模式的不确定性;
 - 引入流量-延迟函数和服务限制水平分别表征拥挤效应和服务质量;
 - 采用基于 probit 模型的随机交通分配模型以增加路径选项。

3. 在研究网络通行能力弹性时,客运 (passenger transportation) 与货运 (freight transportation) 存在以下区别:

- 前者的流量为人组成而后者为商品;
- 前者的交通延误受拥挤影响与流量正相关,而后的延误为定值;
- 评估网络通行能力时前者需考虑用户的路径选择行为,而后者仅由最短路确定;
- 客运系统中存在多个 OD 对,且不同 OD 对间的交通流不可交换或替代。

以上特性使得客运系统较货运系统更为复杂,目前尚无针对客运网络通行能力弹性的直接定量评价方法。本文将货运网络通行能力弹性的定义应用于客运网络并定量研究其通行能力弹性,且同时考虑需求量和需求模式的变化。

10.1.2 Definitions and basic approaches to measuring flexibility

1. 可靠性 (capacity reliability) 是和弹性类似的概念,相比之下前者的研究更多,一般定义为在当前运行环境下设施在预期时间内充分发挥其作用的概率。针对可靠性的研究往往涉及概率,常用的可靠性类型包括:
 - 连通可靠性 (connectivity reliability): 网络节点保持连通的概率;
 - 行程时间可靠性 (travel time reliability): 在给定时间和服务水平下完成特定 OD 对出行的概率;
 - 通行能力可靠性 (capacity reliability): 网络在要求的服务水平下承载特定需求量的概率。
2. 通行能力弹性可描述为网络通行能力与需求变化之间的交互特性。需求变化包括总需求量的变化和需求模式的变化 (不同 OD 对之间的需求量的转移)。本文将客运网络通行能力弹性定义为在要求的表现水平下客运网络对需求变化的适应能力;
3. 参考工程经济领域的盈亏平衡分析 (breakeven analysis) 方法给出了网络通行能力弹性的评价指标。交通网络的盈亏平衡点可定义为使得网络效益和损失相等的服务水平,该服务水平即为最小服务水平,其与最大通行能力的范围即为可接受的需求的变化范围,即为通行能力弹性:
 - 当不考虑需求模式的变化时,通行能力弹性定义为当前需求量与最大通行能力之间的范围,即为备用通行能力 (reserve capacity) 的概念;
 - 当考虑需求模式的变化时,因为网络对于不同的需求模式具有不同的适应能力,此时通行能力弹性包括可接受的需求模式变化范围和相应需求模式下可接受的需求量变化范围两部分。

¹退化交通系统是指组成部分存在退化现象的交通系统。退化主要指交通设施通行能力的退化,往往受不利天气、自然灾害和拥挤事故等扰动的影响。

10.1.3 Capacity flexibility assessment of a passenger transportation system

1. 首先介绍基于备用通行能力的通行能力弹性表征计算方法。此时假设 OD 矩阵中各 OD 对的需求量 q_{ij} 按同一梯度线性增长，定义乘子 μ ，则使得任意边 a 的需求量 $v_a(\mu\mathbf{q})$ 不超过其通行能力 C_a 或不超过特定服务水平限值的最大的 μ 即为所求的弹性，因此得到上层规划模型

$$\begin{aligned} \max \quad & \mu \\ \text{s.t.} \quad & v_a(\mu\mathbf{q}) \leq \beta C_a, \quad \forall a \in A, \beta \in [0, 1] \end{aligned} \quad (10.1)$$

上式约束条件中系数 β 表示服务水平约束，当 $\beta = 1$ 时即表示 E 级服务水平的情况。下层模型为一标准的用户均衡 (user equilibrium, UE) 模型²以求解线性增长后的各路段需求量 $v(\mu\mathbf{q})$

$$\begin{aligned} \min \quad & \sum_{a \in A} \int_0^{v_a} t_a(v) dv \\ \text{s.t.} \quad & \sum_{r \in R_{ij}} f_r^{ij} = \mu q_{ij}, \quad \forall i \in I, j \in J \\ & v_a = \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} f_r^{ij} \delta_{ar}^{ij}, \quad \forall a \in A, \\ & f_r^{ij} \geq 0, \quad \forall i \in I, j \in J, r \in R_{ij} \end{aligned} \quad (10.2)$$

用户均衡模型的目标函数中 $t_a(v)$ 为路段 a 需求量为 v 时的路阻函数 (link performance function)，其物理意义为行程时间。需要说明的是，用户均衡模型的目标函数并不存在直观的经济学或行为学解释。因为标准用户均衡模型得到的最优解为用户效益最优解而不存在随机扰动项，因此这一模型又被称为确定性用户均衡模型 (deterministic user equilibrium, DUE)；

2. 进一步地考虑需求模式和需求量同时变化的情况。基于总体通行能力弹性 (total capacity flexibility) 和有限通行能力弹性 (limited capacity flexibility) 两种概念分别提出两类模型：

- 总体弹性对应极限通行能力 (ultimate capacity) 的概念，极限通行能力定义为不违背路段和区域通行能力限制的最大系统输入，网络中的所有用户都可进行目的地和路径选择以降低行程损失，达到极限通行能力的 OD 需求矩阵与当前需求矩阵往往不存在比例关系。以最大输入量为目标，则上层模型为

$$\begin{aligned} \max \quad & \sum_{i \in I} o_i \\ \text{s.t.} \quad & v_a(\mathbf{o}) \leq C_a, \quad \forall a \in A \\ & o_i = \sum_{j \in J} q_{ij}(\mathbf{o}) \leq o_i^{\max}, \quad \forall i \in I \\ & d_j = \sum_{i \in I} q_{ij}(\mathbf{o}) \leq d_j^{\max}, \quad \forall j \in J \\ & o_i \geq 0, \quad \forall i \in I \end{aligned} \quad (10.3)$$

² 用户均衡为 Wardrop 于 1952 年提出的网络流量分配原则，模型假设 1) 每个出行者都力图选择阻抗最小的路径；2) 出行者能随时掌握整个网络的状态；3) 所有出行者的计算能力和计算水平相同。当达到用户均衡时，不存在出行者能通过单方面改变其出行路径降低自身出行阻抗。当前主流标准用户均衡模型由 Beckmann 于 1956 年给出。

上式中 $v_a(\mathbf{o}), q_{ij}(\mathbf{o})$ 分别表示在系统载入模式 \mathbf{o} 下路段 a 和 OD 对 i, j 间的需求量, 由下层模型求解

$$\begin{aligned} \min \quad & \sum_{a \in A} \int_0^{v_a} t_a(v) dv + \frac{1}{\theta} \sum_{i \in I} \sum_{j \in J} q_{ij} (\ln q_{ij} - 1) \\ \text{s.t.} \quad & \sum_{j \in J} q_{ij} = o_i, \quad \forall i \in I \\ & \sum_{r \in R_{ij}} f_r^{ij} = q_{ij}, \quad \forall i \in I, j \in J \\ & v_a = \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} f_r^{ij} \delta_{ar}^{ij}, \quad \forall a \in A \\ & q_{ij} \geq 0, \quad \forall i \in I, j \in J \\ & f_r^{ij} \geq 0, \quad \forall i \in I, j \in J, r \in R_{ij} \end{aligned} \quad (10.4)$$

下层模型的目标函数同时优化路段需求量 $v_a(\mathbf{o})$ (表示用户路径选择) 和 OD 需求分布 $q_{ij}(\mathbf{o})$ (表示用户目的地选择)。目标函数包含两项, 第一项为标准用户均衡模型, 表示路径选择过程; 第二项为无先验信息的交通分布预测最大熵模型, 表示目的地选择, 因为“熵”的引入, 该模型又称随机用户均衡模型 (stochastic user equilibrium, SUE); 阻抗系数 θ 表示用户进行路径选择时对出行时间 (阻抗) 的敏感程度;

- 有限弹性对应实际通行能力 (practical capacity) 的概念, 实际通行能力定义为当前 OD 需求量与网络所能承载的额外需求量的和, 额外载入的交通生成量可基于目的地吸引和行程损失进行目的地和路径选择。因此上层模型的优化目标为最大化额外的载入量 $\tilde{\mathbf{o}}$

$$\begin{aligned} \max \quad & \sum_{i \in I} \tilde{o}_i \\ \text{s.t.} \quad & v_a(\mathbf{o}) \leq C_a, \quad \forall a \in A \\ & \tilde{o}_i = \sum_{j \in J} \tilde{q}_{ij}(\mathbf{o}) \leq o_i^{\max} - \bar{o}_i, \quad \forall i \in I \\ & \tilde{d}_j = \sum_{i \in I} \tilde{q}_{ij}(\mathbf{o}) \leq d_j^{\max} - \bar{d}_j, \quad \forall j \in J \\ & \tilde{o}_i \geq 0, \quad \forall i \in I \end{aligned} \quad (10.5)$$

同样地, 下层模型优化 $v_a(\mathbf{o}), \tilde{q}_{ij}(\mathbf{o})$, 不同之处在于因为网络中已经存在了一部分的需求 OD 分布 (即存在先验信息), 因此载入新的分布时需考虑目的地选择的损失。定义函数 $c_j(q_j)$ 为节点 j 的选择代价函数 (destination cost function) (对应于路段的阻抗函数), 下层模型如下

$$\begin{aligned} \min \quad & \sum_{a \in A} \int_0^{v_a} t_a(v) dv + \frac{1}{\theta} \sum_{i \in I} \sum_{j \in J} \tilde{q}_{ij} (\ln \tilde{q}_{ij} - 1) + \sum_{j \in J} \int_0^{\sum_{i \in I} \tilde{q}_{ij} + \bar{q}_{ij}} c_j(q) dq \\ \text{s.t.} \quad & \sum_{j \in J} \tilde{q}_{ij} = \tilde{o}_i, \quad \forall i \in I \\ & \sum_{r \in R_{ij}} h_r^{ij} = \bar{q}_{ij}, \quad \forall i \in I, j \in J \\ & \sum_{r \in R_{ij}} f_r^{ij} = \tilde{q}_{ij}, \quad \forall i \in I, j \in J \\ & v_a = \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} (h_r^{ij} + f_r^{ij}) \delta_{ar}^{ij}, \quad \forall a \in A \\ & \tilde{q}_{ij} \geq 0, \quad \forall i \in I, j \in J \\ & f_r^{ij}, h_r^{ij} \geq 0, \quad \forall i \in I, j \in J, r \in R_{ij} \end{aligned} \quad (10.6)$$

进一步地基于拉格朗日乘数法从解析角度理解下层优化目标函数的意义。令原目标函数为 Z , 松弛

前三项等式约束得拉格朗日函数 $L(\tilde{q}_{ij}, h_r^{ij}, f_r^{ij}, \lambda'_i, \lambda''_{ij}, \lambda'''_{ij})$

$$L(\tilde{q}_{ij}, h_r^{ij}, f_r^{ij}, \lambda'_i, \lambda''_{ij}, \lambda'''_{ij}) = Z + \sum_{i \in I} \lambda'_i \left(\sum_{j \in J} \tilde{q}_{ij} - \tilde{o}_i \right) + \sum_{i \in I} \sum_{j \in J} \lambda''_{ij} \left(\sum_{r \in R_{ij}} h_r^{ij} - \bar{q}_{ij} \right) + \sum_{i \in I} \sum_{j \in J} \lambda'''_{ij} \left(\sum_{r \in R_{ij}} f_r^{ij} - \tilde{q}_{ij} \right)$$

基于 KKT 条件 (第17.3.1节), 有

$$\begin{cases} \frac{\partial L}{\partial \tilde{q}_{ij}} \geq 0 \\ \tilde{q}_{ij} \frac{\partial L}{\partial \tilde{q}_{ij}} = 0 \end{cases} \quad \begin{cases} \frac{\partial L}{\partial h_r^{ij}} \geq 0 \\ h_r^{ij} \frac{\partial L}{\partial h_r^{ij}} = 0 \end{cases} \quad \begin{cases} \frac{\partial L}{\partial f_r^{ij}} \geq 0 \\ f_r^{ij} \frac{\partial L}{\partial f_r^{ij}} = 0 \end{cases}$$

显然上式与仅包含等式约束的 KKT 条件不完全一致 (一般只需 $\frac{\partial L}{\partial \tilde{q}_{ij}} = \frac{\partial L}{\partial h_r^{ij}} = \frac{\partial L}{\partial f_r^{ij}} = 0$)。因为以上拉格朗日函数并非松弛所有约束, 仍保留了边界约束 $\tilde{q}_{ij}, h_r^{ij}, f_r^{ij} \geq 0$ 。当最优解位于边界内部时 ($\tilde{q}_{ij}, h_r^{ij}, f_r^{ij} > 0$, 说明路径 r 被分配需求), 要求相关偏导数为零; 当最优解位于边界上时 ($\tilde{q}_{ij}, h_r^{ij}, f_r^{ij} = 0$, 说明路径 r 未被分配需求), 要求相关偏导数大于零。仅考虑被分配需求的路径, 首先关注对 h_r^{ij}, f_r^{ij} 的偏导

$$\begin{cases} \frac{\partial L}{\partial h_r^{ij}} = \frac{\partial Z}{\partial h_r^{ij}} + \lambda''_{ij} = \sum_{a \in A} t_a(v_a) \delta_{ar}^{ij} + \lambda''_{ij} = 0 \\ \frac{\partial L}{\partial f_r^{ij}} = \frac{\partial Z}{\partial f_r^{ij}} + \lambda'''_{ij} = \sum_{a \in A} t_a(v_a) \delta_{ar}^{ij} + \lambda'''_{ij} = 0 \end{cases} \implies \lambda''_{ij} = \lambda'''_{ij} = - \sum_{a \in A} t_a(v_a) \delta_{ar}^{ij}$$

考虑到 $t_a(v_a)$ 表示路段 a 的需求量为 v_a 时的路阻, 上式说明 $\lambda''_{ij}, \lambda'''_{ij}$ 的物理含义为连接 ij 的路径 r 的阻抗总和 (时间成本) 的相反数, 记为 $-T_{ij}$ 。又因为对于连接 ij 的任意路径 r , $\lambda''_{ij}, \lambda'''_{ij}$ 取值均相同, 故有 $\lambda''_{ij} = \lambda'''_{ij} = -T_{ij}$ 。对 \tilde{q}_{ij} 求偏导有

$$\frac{\partial L}{\partial \tilde{q}_{ij}} = \frac{\partial Z}{\partial \tilde{q}_{ij}} + \lambda'_i - \lambda'''_{ij} = \frac{1}{\theta} \ln \tilde{q}_{ij} + c_j(q_j) + \lambda'_i - \lambda'''_{ij} = 0 \implies \tilde{q}_{ij} = \exp\{-\theta(c_j(q_j) + \lambda'_i - \lambda'''_{ij})\}$$

记 $c_j(q_j) = C_j$ 表示节点 j 的阻抗, 并将 $-\lambda'''_{ij} = T_{ij}$ 代入上式, 得

$$\tilde{q}_{ij} = \exp\{-\theta(C_j + T_{ij} + \lambda'_i)\} = \exp\{-\theta\lambda'_i\} \exp\{-\theta(C_j + T_{ij})\}$$

又因为 $\sum_{j \in J} \tilde{q}_{ij} = \tilde{o}_i$

$$\begin{aligned} \exp\{-\theta\lambda'_i\} \sum_{j \in J} \exp\{-\theta(C_j + T_{ij})\} = \tilde{o}_i &\implies \exp\{-\theta\lambda'_i\} = \frac{\tilde{o}_i}{\sum_{j \in J} \exp\{-\theta(C_j + T_{ij})\}} \\ &\implies \tilde{q}_{ij} = \frac{\tilde{o}_i \exp\{-\theta(C_j + T_{ij})\}}{\sum_{j \in J} \exp\{-\theta(C_j + T_{ij})\}} \end{aligned}$$

可以看到, 下层模型实际上是基于多项选择模型 (MNL 模型) 分配额外载入的 OD 分布。此类基于 Logit 模型进行路径选择的随机用户均衡模型统称 Logit 型随机用户均衡模型 (Logit-SUE), 除此之外还有 MNW-SUE 和 MDM-SUE 等模型。

10.1.4 Numerical Results

- 对于第一类模型——基于备用通行能力的模型, 论文采用 Frank-Wolfe 交通分配算法求解。对于考虑需求模式变化的后两种模型, 论文基于遗传算法求解整个双层规划问题, 而底层的交通分配问题则基于部分线性化算法 (partial linearization algorithm) 求解 (详见10.7.2节);
- 路阻函数 t_a 为标准 BPR 函数 (standard Bureau of Public Road function), 式中 t_a, t_a^f, v_a, C_a 分别表示行程时间、自由流行程时间、路段流量和路段通行能力;

$$t_a = t_a^f \left[1 + 0.15 \left(\frac{v_a}{C_a} \right)^4 \right]$$

3. 目的地 j 的选择代价函数 $c_j(q_j)$ 表达式如下, 其中 q_j 为目的地 j 的总交通吸引量, α_j, β_j, m_j 为预设超参数

$$c_j(q_j) = \alpha_j q_j^{\beta_j} - m_j$$

4. 当基于备用通行能力的概念量化通行能力弹性时, 仅考虑需求量的变化而不考虑需求模式的变化, 但此时通行能力弹性仍受初始需求模式的影响。初始需求模式与路网通行能力匹配程度越高通行能力弹性也越高。数值分析时对同一初始需求量设置多种初始需求模式, 研究不同需求模式对通行能力弹性的影响;
5. 进一步考虑需求量和需求模式同时变化的情况。需求模式由下层交通分配模型的阻抗参数 θ 体现。 θ 越大说明出行者对行程时间越敏感, 对应需求模式中短途出行的比例越高。预设多组 θ 反映多种需求模式。

10.1.5 英汉互译

English	Chinese	English	Chinese	English	Chinese
borrow	借 (v)	breakeven	盈亏平衡 (n,adj)	impedance	阻抗 (n)
summation	总和 (n)	congruous	一致的	nil	零 (n)

10.2 Metamodel-based calibration of large-scale multi-modal microscopic traffic simulation (TRC, 2021)

ABSTRACT

- □ ×

微观交通仿真模型在建模家庭成员互动、车辆共享和车辆交互等方面具有优越性。但因其较高的复杂性（非线性、无封闭形式等），校正微观交通仿真模型存在较大的难度，且随着出行模式的增加和网络规模的增大问题愈加严重。本文提出了一种基于元模型 (metamodel) 的仿真优化框架以校正多模式微观交通仿真平台。元模型用于替代仿真模型，以实现更快速便捷的仿真校准。具体地，提出一种简化多模式交通仿真模型并将其整合入元模型中，并比较了 3 种基于梯度的元模型方案。所提的方法用于香港路网离线仿真校正，结果证实了方法对大规模交通仿真校正的适用性。目

10.2.1 Introduction

1. 为应用微观交通仿真框架于真实路网, 仿真模型校正至关重要;
2. 校正的对象包括 **OD 需求模式校正、路径流校正和交通流参数校正**等。校正的过程又可分为在线校正和离线校正。具体校正时, 往往基于计数数据等集计数据或车牌、轨迹数据等个体数据分别或同时地校正行为参数 (如出行选择模型) 和交通流参数;
3. 多模式多智能体大规模微观交通仿真校正的难度主要在于:
 - 仿真中的多智能体互相独立, 具有独立的行为。因而参数与实际仿真效果之间的关系存在极强的随机性和非线性, 且无封闭的映射形式;
 - 大规模网络微观交通仿真存在较大的计算压力;
 - 引入多模式交通后将存在多个交通网络层, 并引入更多的出行选择。
4. 为解决微观交通仿真校正的问题, 部分研究提出了基于仿真的校正优化方法, 并采用如遗传算法等元启发算法。元启发算法要求多次仿真迭代, 因此常应用于单模式交通走廊、高速等小型网络。神经网络也曾被用于微观仿真校正, 但神经网络也需要大规模训练。同步扰动随机逼近 (simultaneous perturbation stochastic approximation, SPSA) 算法 (详见第 19.2 节) 是另一种适用于高维问题的随机优化算法, 也曾被成功的应用于微观仿真校正, 但其准确度随着问题规模的增加而下降。扩展卡尔曼滤波 (extended Kalman filter) 也曾被用于微观交通仿真校正;
5. 另一类基于仿真的校正优化方法选择构造目标函数的近似模型以降低计算量, 此类方法称为元模型 (meta-model) 法。然而相关技术尚未被用于大规模多模式交通仿真校正;
6. 本文采用主要基于集计数据离线地校正微观交通仿真器的行为参数:
 - 基于 MATSim (Multiple-Agent Transportation Simulation) 进行微观交通仿真校正平台;
 - 采用元模型对原始复杂目标函数的整体近似, 目标是以较低的计算成本求近似解;

- 元模型中包含一简化交通模型和一般多项式 (generic polynomial)。其中交通模型考虑多模式交通仿真的复杂性进行简化以确保快速评价；一般多项式捕捉简化交通模型与实际仿真的差异；
- 在校正过程中，元模型持续更新以拟合仿真观测结果。元模型中交通模型的参数也同时调整；
- 元模型校正包括两个过程：模型提升 (metamodel improvement) 与模型优化 (metamodel optimization)。
模型提升时主要校正交通模型参数，而后在优化过程中校正行为参数，参数返回模型后即可进一步提升。迭代执行以上两步实现模型校正；
- 为提升模型的最优性 (optimality)，本文进一步研究了梯度信息（例如元模型拟合过程中道路流量关于校正参数的导数）的作用。

7. 本文的主要贡献如下：

- 提出了基于仿真的大规模多模式交通仿真校正方法，采用路段流量和公交刷卡数据用于多模式交通仿真校正；
- 采用元模型进行高效校正；
- 设计交通模型提升过程以利于模型参数充分优化近似实际仿真；
- 为探索梯度信息对于求解质量的作用设计了三种基于梯度的元模型方案。

10.2.2 Problem statement & Measurement formation

1. 本文采用多种类型的集计指标（路段计数、公交刷卡）校正微观交通仿真。对于特定的指标 u ，记 $y_{\xi,t}$ 表示为 t 时刻于位置 ξ 的计数值。且 $y_{\xi,t}^{real}, y_{\xi,t}^{sim}$ 分别表示实际值和仿真值。将多类指标通过权重 χ_u 聚合为单目标函数，则仿真校正问题可以转化为如下带权重的最小二乘问题

$$\min_{\theta} f(\theta) = \sum_u \chi_u \left[\sum_{\xi \in E_u} \sum_t (y_{\xi,t}^{sim}(\theta) - y_{\xi,t}^{real})^2 \right]$$

其中 θ 为校正参数； E_u 为指标 u 所相关的位置集合。注意到 $y_{\xi,t}^{sim}$ 的取值取决于交通状态（供给）和人的出行选择（需求）的交互关系，这是一个典型的固定点问题 (fix-point problem)，可以通过迭代求解实现最终的均衡状态（如用户均衡）。因此欲最小化目标函数 $f(\theta)$ 即要求通过迭代优化求解 $y_{\xi,t}^{sim}$ ；

2. 本文考虑的指标类型包括路段计数数据和公交刷卡数据：

- 路段计数数据为网络连边上的标准小车流量数据。定义二元变量 $\zeta_{l,t,n} \in 0, 1$ 表示驾驶员 n 于 t 时刻是否选择连边 l ，又定义二元变量 $\psi_{l,t,v} \in 0, 1$ 表示标准小车系数为 P_v 的公交车 v 于 t 时刻是否选择连边 l 。则连边 l 上的标准小车流量仿真值 $x_{l,t}^{sim}$ 表示为

$$x_{l,t}^{sim} = \sum_n \zeta_{l,t,n} + \sum_v \psi_{l,t,v} \cdot P_v$$

- 考虑基于进出站的刷卡数据（如地铁刷卡）和基于线路的刷卡数据（如常规公交刷卡）。定义费用路径 fl 表示一次公交出行及支付信息。对于基于进出站的刷卡数据， fl 记录了公交模式、上车站点和下车站点；对于基于线路的刷卡数据， fl 记录了公交线路、上车站点和下车站点。定义二元变量 $\zeta_{fl,t,n} \in 0, 1$ 表示乘客 n 于 t 时刻是否选择费用路径 fl 。则费用路径 fl 上的公交刷卡数据仿真值 $\varepsilon_{l,t}^{sim} = \sum_n \zeta_{fl,t,n}$ 。

3. 本文考虑的指标类型 $x_{l,t}^{sim}, \varepsilon_{l,t}^{sim}$ 均为基于路径 (trip-based) 的指标，也可替换为基于活动 (activity-based) 的指标，如平均工作或购物时长、活动地点等等。

10.2.3 Metamodal SBO (simulation-based optimization) structure

- 上节所提目标函数包含仿真结果变量 $y_{\eta,t}^{sim}$ ，因此该优化问题属于基于仿真的优化问题 (simulation-based optimization, SBO)。元模型用于近似替代基于仿真的目标函数，同时结合信赖域优化算法 (trust region optimization algorithm)，以高效优化该问题；
- 在 SBO 框架下，仿真校正过程通过迭代执行元模型提升（或拟合）和元模型优化两部分实现：
 - 在提升步中，基于仿真稳定时得到的仿真结果生成元模型，元模型局具有相应的信赖域；

- 在优化步中，于信赖域内生成新的校正参数取值，以最小化实际观测与元模型结果的误差。

3. 记第 i 次迭代中 t 时刻 ξ 位置处的元模型为 $y_{\xi,t}^{m,i}$ ，且有

$$y_{\xi,t}^{m,i}(\theta|\alpha, \beta, \kappa) = \alpha_{\xi,t}^i \lambda_{\xi,t}^i(\theta|\kappa) + g(\theta|\beta_{\xi,t}^i)$$

θ 为待校正参数（行为参数）； $\lambda_{\xi,t}^i(\theta|\kappa)$ 表示简化交通模型得到的相应位置、时刻的计算指标 $(x_{l,t}^{sim}, \varepsilon_{l,t}^{sim})$ ， κ 为简化模型参数； $g(\theta|\beta_{\xi,t}^i)$ 为一般多项式，与行为参数 θ 有关； $\alpha_{\xi,t}^i, \beta_{\xi,t}^i$ 为用于拟合元模型与仿真结果的参数，与迭代轮次、时间、位置有关； $\alpha_{\xi,t}^i, g(\theta|\beta_{\xi,t}^i)$ 可共同视为对简化交通模型 $\lambda_{\xi,t}^i(\theta|\kappa)$ 的校正；

4. $\alpha_{\xi,t}^i, \beta_{\xi,t}^i, \kappa$ 的取值于元模型提升步中确定，而在元模型优化步中则基于下式求解 θ

$$\min_{\theta} f_M(\theta) = \sum_u \chi_u \left[\sum_{\xi \in E_u} \sum_t (y_{\xi,t}^{m,i}(\theta|\alpha, \beta, \kappa) - y_{\xi,t}^{real})^2 \right] \quad \theta \in \Theta_{\Delta}$$

下标 M 表示元模型 $y_{\xi,t}^{m,i}$ 的集合； $f_M(\theta)$ 即为元模型与真实结果的误差； Θ_{Δ} 为元模型集合 M 的信赖域。

10.2.4 Multimodal traffic model formulation

1. 本文校正基于活动的 (activity-based) 大规模交通仿真器的大部分行为参数。参数主要包括金钱和进行活动的边际效用，此外还包括时间、距离和与出行模式相关（等待时间、步行时间、换乘等）的边际效用；
2. 记简化多模式交通模型为 M_{λ} ，设计其为包含上述参数的基于出行的 (trip-based) 静态交通分配模型³：
 - 路段交通状态由 BPR 函数描述。基于 BPR 函数得到的路段行程时间对小车和公交车均一致；
 - 假设路段公交车流量由固定的公交时刻表决定，与乘客流量无关。因此若公交需求量增加则乘客面临的延误也会加剧；
 - 采用包含模式和路径选择的随机用户均衡模型（参考 P112 式 10.4）；
 - 采用双层巢式 logit 模型 (two-level nested logit model)，上层模型实现模式选择，下层模型实现路径选择。
3. 出行模式和路径选择模型：定义 $q_{s,t}$ 为 t 时刻 OD 对 s 的需求量，分为小车和公交两种类型 $(q_{s,t}^a, q_{s,t}^{\omega})$ 。基于 logit 模型实现出行模式选择，考虑两种模式的期望最大效用 $v_{s,t}^a, v_{s,t}^{\omega}$ 。记小车和公交的路径分别为 r, ω_r ，其中小车路径为恒定值，而公交路径则会随着公交运力的饱和而随时间变化。基于 logit 路径选择模型分配 $q_{s,t}^a, q_{s,t}^{\omega}$ 至相应路径得到两种出行模式路径流 $f_{s,t,r}, f_{s,t,\omega_r}$ 。出行模式和路径选择模型如下， κ 为模型参数

$$\begin{aligned} q_{s,t} &= q_{s,t}^a + q_{s,t}^{\omega} \\ q_{s,t}^a &= \frac{q_{s,t} \cdot \exp\{\kappa_5 v_{s,t}^a\}}{\exp\{\kappa_5 v_{s,t}^a\} + \exp\{\kappa_5 v_{s,t}^{\omega}\}} \quad q_{s,t}^{\omega} = \frac{q_{s,t} \cdot \exp\{\kappa_5 v_{s,t}^{\omega}\}}{\exp\{\kappa_5 v_{s,t}^a\} + \exp\{\kappa_5 v_{s,t}^{\omega}\}} \\ f_{s,t,r} &= \frac{q_{s,t}^a \cdot \exp\{\kappa_6 v_{r,t}\}}{\sum_c \exp\{\kappa_6 v_{c,t}\}} \quad f_{s,t,\omega_r} = \frac{q_{s,t}^{\omega} \cdot \exp\{\kappa_6 v_{\omega_r}\}}{\sum_{\omega_r} \exp\{\kappa_6 v_{\omega_r}\}} \end{aligned}$$

采用 MNL 模型形式的出行模式和路径选择模型，于宏观层面对应 Logit 型随机用户均衡问题 (Logit-SUE, Logit-SUE 问题与 MNL 模型的关系可参考第 10.7.2 节推导)。基于连续加权平均法 (method of successive weighted average, MSWA) 求解 SUE 问题 (参考第 10.7.3 节)；

4. 出行模式与路径效用函数：上式中由效用函数 $v_{s,t}^a, v_{s,t}^{\omega}, v_{r,t}, v_{\omega_r}$ 定义了模式和相应路径的效用。其中 $v_{r,t}$ 考虑行程时间 $T_{r,t}$ 和行程距离 d_r 。 v_{ω_r} 则考虑了行程时间 T_{ω_r} 、步行时间 $T_{\omega_r}^{walk}$ 、等待时间 $T_{\omega_r}^{wait}$ 、步行距离 $d_{\omega_r}^{walk}$ 、路径费用 $f_{r,t}$ 和换乘次数 $|\Gamma_{\omega_r}|$ 。因为传统的多项 logit 模型要求多个选项完全平等独立，而路径选择模型中不可避免地存在路线部分重合的情况，因此在定义路径效用时考虑了路径尺度系数 (path size factor) ps_r, ps_{ω_r} ，用于描述选择集中的路径与其它路径重合的情况 (参考第 10*1 节)

$$v_{r,t} = \frac{\theta_1}{3600} T_{r,t} + \theta_2 d_r + \ln ps_r$$

³静态交通分配是对固定的 OD 需求进行分配，而动态交通分配则是在对时变的需求进行分配，分配过程中往往假设时变需求 $q(t)$ 已知，因为需求时变，故分配过程中路阻也是时变的。

$$\begin{aligned}\nu_{\omega_t} &= \theta_3 + \frac{\theta_4}{3600} T_{\omega_t} + \theta_5 f r_{\omega_t} + \theta_6 T_{\omega_t}^{walk} + \theta_7 T_{\omega_t}^{wait} + \theta_8 (|\Gamma_{\omega_t}| - 1) + \theta_5 \theta_9 d_{\omega_t}^{walk} + \ln p s_{\omega_t} \\ \nu_{s,t}^a &= \frac{1}{\kappa_6} \ln \sum_r \kappa_6 \nu_{r,t} \quad \nu_{s,t}^\omega = \frac{1}{\kappa_6} \ln \sum_{\omega_t} \kappa_6 \nu_{\omega_t}\end{aligned}$$

5. 流量计算：基于出行模式与路径选择模型和相应效用函数即可将交通需求分配至网络中，进而以路段为单位对流量进行集计。包括路段流量和公交客流。记 t 时刻路段 l 的小车公交总流量为 $\lambda_{l,t}$ ，其中小车和公交通流量分别为 $\lambda_{l,t}^a, \lambda_{l,t}^\tau$ ，则

$$\lambda_{l,t} = \lambda_{l,t}^a + \lambda_{l,t}^\tau \quad \lambda_{l,t}^a = \sum_s \sum_t \sum_{r \in R_s} f_{s,t,r} \cdot \delta_{lr} \quad \lambda_{l,t}^\tau = \sum_{b \in B} \sum_t \sum_{\tau \in TVR_b} \Upsilon_{\tau,t,b} \cdot p_b \cdot \delta_{l\tau}$$

上式中 s 为网络中的一对 OD 对； R_s 为连接 OD 对 s 的小车路径集合； B 为公交车类型集合； TVR_b 为公交车类型 b 的所有公交线路集合； p_b 为公交车类型 b 的小车当量系数； $f_{s,t,r}, \Upsilon_{\tau,t,b}$ 为相应的小车、公交路径流； $\delta_{lr}, \delta_{l\tau} \in \{0, 1\}$ 为指示变量。进一步地集计公交客流。将公交网络中的连边划分为公交直达边 (**transit direct link**) π_t 、公交换乘边 (**transit transfer link**) γ_t 与公交上车边 (**transit boarding link**) l_γ ，则相应连边的客流（乘车客流、换乘客流、上车客流）分别为 $\lambda_{\pi_t}, \lambda_{\gamma_t}, \lambda_{l_\gamma, t, \tau}$

$$\lambda_{l_\gamma, t, \tau} = \sum_{\pi_t \in \Pi_{\tau,t}} \lambda_{\pi_t} \delta_{l_\gamma, \pi_t} \quad \lambda_{\pi_t} = \sum_s \sum_{\omega_t \in \Omega_t} f_{s,\omega_t} \delta_{\pi_t, \omega_t} \quad \lambda_{\gamma_t} = \sum_s \sum_{\omega_t \in \Omega_t} f_{s,\omega_t} \delta_{\gamma_t, \omega_t}$$

上式中 $\Pi_{\tau,t}$ 为所有可由换乘边上车的直达边集合⁴；

6. 距离与时间计算：基于网络各路段流量即可由交通流模型得到行程时间和距离，用于更新效用函数。首先考虑小车和公交于实际路网上的行程时间。记 $T_{r,t}, T_{\omega_t}$ 分别表示小车路径 r 和公交路径 ω_t 的行程时间，由网络各路段的行程时间 $T_{l,t}$ 决定。 $T_{l,t}$ 由 BPR 函数描述，其中 $T_{l,o}, \lambda_{l,c}$ 分布为连边 l 的自由流时间和通行能力，而 $g_{l,t}$ 表示 t 时段内路段 l 的绿灯时长比例

$$T_{r,t} = \sum_{l \in L_r} T_{l,t} \quad T_{\omega_t} = \sum_{\pi_t \in \Pi_{\omega_t}} \sum_{l \in L_{\omega_t}} T_{l,t} \quad T_{l,t} = T_{l,o} \left[1 + \kappa_1 \left(\frac{\lambda_{l,t}}{\lambda_{l,c} \cdot g_{l,t}} \right)^{\kappa_2} \right]$$

进一步地定义换乘步行时间 $T_{\omega_t}^{walk}$ 和等待时间 $T_{\omega_t}^{wait}$ ，由换乘边 γ_t 上的步行时间 $T_{\gamma_t}^{walk}$ 和等待时间 $T_{\gamma_t}^{wait}$ 决定。其中 $T_{\gamma_t}^{wait}$ 由经验公式⁵确定，其中分子部分包括 t 时刻由上车边 l_γ 选择公交线路 τ 上车的客流 $\lambda_{l_\gamma, t, \tau}$ 和尚在步行换乘准备上车的客流 $\sum_{\gamma_t \in \Gamma_{l_\gamma}} \lambda_{\gamma_t}$ ，而分母部分则表示线路 τ 上所有类型公交车的总客运能力，其中 C_b 为 b 类公交车的载客人数。除此之外经验公式中 H_{τ_γ} 表示线路 τ 的频率

$$T_{\omega_t}^{walk} = \sum_{\gamma_t \in \Gamma_{\omega_t}} T_{\gamma_t}^{walk} \quad T_{\omega_t}^{wait} = \sum_{\gamma_t \in \Gamma_{\omega_t}} T_{\gamma_t}^{wait} \quad T_{\gamma_t}^{wait} = \kappa_3 H_{\tau_\gamma} + H_{\tau_\gamma} \left(\frac{\lambda_{l_\gamma, t, \tau} + \sum_{\gamma_t \in \Gamma_{l_\gamma}} \lambda_{\gamma_t}}{\sum_{b \in B} \Upsilon_{\tau,t,b} \cdot C_b} \right)^{\kappa_4}$$

除了行程时间、步行时间和等待时间外，定义公交出行路径 ω_t 所需的步行距离 $d_{\omega_t}^{walk}$ ，由路径中所有换乘边的步行距离 $d_{\gamma_t}^{walk}$ 决定

$$d_{\omega_t}^{walk} = \sum_{\gamma_t \in \Gamma_{\omega_t}} d_{\gamma_t}^{walk}$$

7. 指标提取：以上即为本文所提简化交通模型。本文考虑的指标包括路段标准小车流量和费用路径 fl 上的公交刷卡数据。其中路段标准小车流量已于上文“流量计算”时由 $\lambda_{l,t}$ 得到，而刷卡数据 $\lambda_{fl,t}$ 则定义如下，其中 f_{s,ω_t}^* 为“流量计算”是得到的连接 OD 对 s 的公交出行路径 ω_t 的客流量 f_{s,ω_t} 的收敛解（均衡解）

$$\lambda_{fl,t} = \sum_s \sum_{\omega_t \in \Omega_{s,t}} f_{s,\omega_t}^* \cdot \delta_{\omega_t, fl}$$

⁴注意区分 ω_t 和 τ 。前者指公交出行路径，可能通过多次换乘实现；后者指公交车线路，不存在换乘现象。

⁵Cea J D , E Fernández. Transit Assignment for Congested Public Transport Systems: An Equilibrium Model[J]. Transportation Science, 1993, 27.

10.2.5 Traffic model improvement & Metamodel formulation and fitting

1. 注意到在上节所提简化交通模型中包含两类参数 κ, θ 。其中 κ 为交通模型特有参数，包括 BPR 函数参数、等待时间参数、模式路径选择模型尺度参数等，用于描述连边的交通状态。 θ 则为行为参数（也是微观交通仿真的参数），用于构建效用函数，反映出行者的偏好。为尽可能提升简化模型对复杂交通状态（微观交通仿真）的拟合效果，基于简化模型与微观交通仿真观测值的偏差校正交通参数 κ

$$\kappa^{k+1} = \arg \min_{\kappa} \sum_{i=1}^{n_k} \sum_{\xi} \sum_t [\omega^{ki} (y_{\xi,t}^{sim}(\theta^i) - \lambda_{\xi,t}^i(\theta^i|\kappa))]^2 \quad \omega^{ki} = \frac{1}{1 + \|\theta^i - \theta^k\|}$$

假设进行至第 k 轮优化时一共进行了 n_k 次仿真（即每轮优化可能不止执行 1 次仿真，但每轮优化的行为参数 θ^i 一致），得到 n_k 组仿真结果，则第 $k+1$ 轮仿真时的交通流参数 κ^{k+1} 则基于之前 n_k 次仿真的观测值确定。仿真观测值对应不同的权重 ω^{ki} 。第 i 轮优化的参数 θ^i 与当前参数 θ^k 越接近则相应的权重越大；

2. 进一步地，设计不同具体形式的元模型：

- **Metamodel with problem-specific traffic model**

首先考虑的元模型由上节所提简化交通模型和线性多项式组成，后续称该模型为 M_λ 。相应地可以得到元模型提升时参数 α, β 的优化过程。该模型优化过程并不要求梯度信息。

$$y^m(\theta|\alpha, \beta, \kappa) = \alpha \lambda(\theta|\kappa) + \beta_0 + \sum_{j=1}^d \theta_{j-1} \beta_j$$

$$\alpha, \beta = \arg \min_{\alpha, \beta} \sum_{i=1}^{n_k} [\omega^{ki} (y^{sim}(\theta^i) - y^m(\theta^i|\alpha, \beta, \kappa))]^2 + (\alpha - 1)^2 + \sum_{j=0}^d \beta_j^2$$

- **Gradient-based metamodel formulation**

将梯度信息引入优化过程往往有助于提升优化的收敛速度和解的质量。在基于梯度的元模型中，简化交通模型 λ 用于对微观交通仿真的全局近似，而梯度信息则提供更为精确的局部信息。因此本小节提出包括梯度信息的元模型，并与上文所提的无需梯度信息的元模型的优化效果比较。基于有限差分 (**finite difference, FD**) 计算梯度信息。下式中 $y(\theta)$ 表示路段集计流量，可以是仿真模型 $y^{sim}(\theta)$ 或交通模型 λ_θ 的结果

$$\frac{\partial y(\theta)}{\partial \theta_j} = \frac{y(\theta_1, \theta_2, \dots, \theta_j + h, \dots, \theta_n) - y(\theta_1, \theta_2, \dots, \theta_j, \dots, \theta_n)}{h}$$

注意到元模型的通式中 $\Lambda = g(\theta|\beta)$ 用于拟合元模型 y^m 与交通模型 λ 的误差。在第 k 轮优化时，可对其于 θ^k 处作一阶近似

$$\begin{aligned} \Lambda &= g(\theta|\beta) = y^m(\theta|\alpha, \beta, \kappa) - \alpha \lambda(\theta|\kappa) \\ &\approx y^m(\theta^k|\alpha, \beta, \kappa) - \alpha \lambda(\theta^k|\kappa) + (\nabla y^m(\theta^k|\alpha, \beta, \kappa) - \alpha \nabla \lambda(\theta^k|\kappa))^T (\theta - \theta^k) \end{aligned}$$

上式即引入梯度信息。然而上式中 Λ 要求计算元模型 y^m 的梯度 ∇y^m ，而原函数 y^m 又由 Λ 决定。为避免循环定义，以实际仿真结果 y^{sim} 替代元模型 y^m ，则

$$\Lambda^k = y^{sim}(\theta^k) - \alpha \lambda(\theta^k|\kappa) + (\nabla y^{sim}(\theta^k) - \alpha \nabla \lambda(\theta^k|\kappa))^T (\theta - \theta^k)$$

考虑三种不同具体形式的基于梯度的元模型：

GD-I (仅考虑当前点 θ^k 附近的信息更新元模型) 直接以 Λ^k 替代 $g(\theta|\beta)$ ，出于简化考虑令 $\alpha = 1$ 。

此时式中不存在元模型参数 (κ 属于交通模型参数)，因此不需要执行元模型优化步以优化元模型参数，只需计算梯度 $\nabla y^{sim}, \nabla \lambda$ 即可更新元模型

$$y^{m, GD-I}(\theta|\kappa) = \lambda(\theta|\kappa) + \Lambda^k = \lambda(\theta|\kappa) + y^{sim}(\theta^k) - \lambda(\theta^k|\kappa) + (\nabla y^{sim}(\theta^k) - \nabla \lambda(\theta^k|\kappa))^T (\theta - \theta^k)$$

GD-II (同时基于全局和局部信息更新元模型) 同样地令 $\alpha = 1$, 定义元模型如下。模型中存在参数 β , 需基于元模型优化步求解最优取值。优化时引入局部信息 β_o^e, β_o^j

$$y^{m, GD-II}(\theta|\beta, \kappa) = \lambda(\theta|\kappa) + \beta_0 + \sum_{j=1}^d \beta_j (\theta_{j-1} - \theta_{j-1}^k)$$

$$\beta = \arg \min_{\beta} \sum_{i=1}^{n_k} [\omega^{ki} (y^{sim}(\theta^i) - y^{m, GD-II}(\theta^i|\beta, \kappa))]^2 + (\beta_0 - \beta_o^e) + \sum_{j=1}^d (\beta_j - \beta_o^j)^2$$

$$\beta_o^e = y^{sim}(\theta^k) - \lambda(\theta^k) \quad \beta_o^j = \nabla_j y^{sim}(\theta^k) - \nabla_j \lambda(\theta^k)$$

GD-III (同时基于全局和局部信息更新元模型) 元模型更新 α 以调整简化交通模型的全局近似效果, 同时引入 Λ 以确保元模型于局部点处的准确性。令 α 为待优化参数而不考虑 β , 则引入 Λ^k 后元模型如下

$$y^{m, GD-III}(\theta|\alpha, \kappa) = \alpha \lambda(\theta|\kappa) + y^{sim}(\theta^k) - \alpha \lambda(\theta^k|\kappa) + (\nabla y^{sim}(\theta^k) - \alpha \nabla \lambda(\theta^k|\kappa))^T (\theta - \theta^k)$$

$$\alpha = \arg \min_{\alpha} \sum_{i=1}^{n_k} [\omega^{ki} (y^{sim}(\theta^i) - y^{m, GD-III}(\theta^i|\alpha, \kappa))]^2 + (\alpha - 1)^2 \quad \alpha \neq 0$$

- **generic polynomial metamodel**

最后考虑不包括简化交通模型, 仅由多项式组成的元模型, 并考虑线性 (Linear) 和二次型 (Quadratic) 两种多项式形式。因为不存在简化交通模型, 因此元模型中只有 β 一个参数

$$y^{m, lin}(\theta|\beta) = \beta_0 + \sum_{j=1}^d \theta_{j-1} \beta_j$$

$$y^{m, qua}(\theta|\beta) = \beta_0 + \theta^T G + \theta^T H \theta \quad G_i = \beta_{i+1} \quad H_{ij} = H_{ji} = \beta_{1+d+i} \quad i, j = 0, \dots, d-1$$

$$\beta = \arg \min_{\beta} \sum_{i=1}^{n_k} [\omega^{ki} (y^{sim}(\theta^i) - y^{m}(\theta^i|\beta))]^2 + \beta^T \beta$$

10.2.6 Trust region SBO algorithm

信赖域方法相关内容可参考第 17.3.4 节。

10.2.7 Numerical examples & Application to Hong Kong network

- 首先将所提基于元模型的校正方法用于一多模式交通小型路网, 校正 MATSim 行为参数。**MATSim 校正参数即为元模型中的参数 θ , 元模型经校正后参数 θ 即可用于 MATSim 仿真。**网络中包括 20 条道路和 4 条公交线路, 包括 2 条常规公交线路和 2 条地铁线路。公交的线路、班次间隔、费用等均为定值。共生成 18000 个智能体;
- 在本例中仅校正 2 个参数——小汽车行程时间的边际效应 $\theta_1 \in [-60, -10]$ 和费用的边际效应 $\theta_5 \in [0.5, 1.5]$ 。分成 5 个时段进行网络校正: 12-7 am、7-10 am、10 am - 4 pm、4-8 pm、8-12 pm。校正的指标包括 **20 条道路的流量、22 处公交刷卡进站数据和 27 处公交刷卡进站-出站数据**;
- 仿真校正时, 采用另一微观交通仿真器替代真实环境, 令上述两参数的真实值分别为 $(-30, 1)$ 。并设待优化元模型的两参数初始值分别为 $(-45, 1.2), (-50, 0.6), (-40, 0.8)$ 。最终考虑以下 7 种元模型:
 - 线性元模型 (Linear);
 - 二次型元模型 (Quadratic);
 - 包含简化交通模型的元模型 M_λ (但在优化过程中不更新简化交通模型参数 κ);
 - 包含简化交通模型的元模型 M_λ (在优化过程中更新简化交通模型参数 κ);
 - 基于梯度的元模型 GD-I;
 - 基于梯度的元模型 GD-II;
 - 基于梯度的元模型 GD-III。

4. 采用均方根差 (root mean square error, RMSE) 和 GEH 统计量评价校正元模型的拟合效果。GEH 统计量在交通工程领域常用于评价仿真与实际的路网连边计数数据的拟合程度。不同于均方根差, GEH 统计量针对每一连边 i 单独计算。若至少 85% 连边的 GEH_i 小于 5 则认为仿真场景与实际较接近 (FHWA 建议);

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i^{sim} - y_i^{real})^2} \quad GEH_i = \sqrt{\frac{2(y_i^{sim} - y_i^{real})^2}{y_i^{sim} + y_i^{real}}}$$

5. 首先仅比较前 4 中元模型。校正后的多项式元模型仅在均方根差上取得微弱的提升。相比于简单的多项式元模型, 包含简化交通模型的元模型 M_λ 在优化过程中体现出具有更快的收敛速度与收敛效果。优化简化交通模型参数 κ 有助于进一步收敛;
6. 进一步地测试基于梯度的元模型的优化效果, 并以无梯度信息的元模型 M_λ 作为对比。四种模型中 GD-I 最差, 但其仍可得到优于初始值的解。GD-III 与 M_λ 模型的优化效果最好且非常接近, 以均方根差为指标前者甚至在三个初始解下均优于后者。就收敛速度而言, GD-III 模型最快, GD-II 与 M_λ 模型接近;
7. 将所提元模型方法应用于大规模多模式多智能体微观交通仿真校正 (仿真香港岛路网)。共校正 21 个参数, 其它参数取默认值。同样地以另一微观仿真平台替代真实交通环境。定义最大仿真次数为 20 次;
8. 校正结果指出, 对于大规模多模式交通网络, 所提元模型 M_λ 可以实现快速收敛 (仅需 3 秒), 而且采用校正后仿真参数的微观交通仿真结果与真实情况非常接近。

10.2.8 英汉互译

English	Chinese	English	Chinese	English	Chinese
superiority	优越性	intrinsic	本征的	deteriorate	恶化
general-purpose	通用的	generic	一般的	discrepancy	差异 (n)
optimality	最优性	elaboration	精致	alight	降落 (v)
disutility	负效用	disregard	忽视 (n,v)	elaborate	复杂的、详述 (v)
carry-over	遗留 (n)	coevolutionary	共同进化的	tertiary	第三级、第三级的

10.3 Calibration of Microscopic Traffic Simulation Models: Methods and Application (TRR, 2007)

ABSTRACT

微观交通仿真可以精确地估计时变需求模式和驾驶员个体行为参数下的交通状况。而在实际微观仿真之前需要精确估计部分仿真参数, 包括模型输入 (OD 需求模式) 和跟驰换道模型参数。本文提出了一种基于一般交通观测值同时估计所有微观模型参数的方法。并通过大规模仿真论证所提方法的可行性和有效性。

10.3.1 Introduction

1. 微观交通模型的仿真结果由输入需求和一系列微观参数共同决定, 要求上述参数尽可能与实际交通状态一致, 因此仿真校正至关重要;
2. 非集计数据是微观仿真校正的最理想数据, 但难以获得。需要基于更常见的集计数据校正微观交通仿真模型, 最小化仿真与观测结果的偏差。然而该过程又受到微观模型校正参数多、计算消耗大的限制;
3. 微观交通仿真模型中的参数包括行为参数、路径选择参数和 OD 矩阵等。仿真校正时应考虑各参数之间的交互效应;
4. 已有研究常采用不同类型数据校正不同微观仿真参数: 如以非集计轨迹数据校正跟驰换道模型, 又以集计数据调整 OD 需求;
5. 本文的主要贡献即提出一种在可接受时间范围内基于集计数据同时校正各种类型的参数的优化方法, 并通过大规模微观交通仿真验证其可行性和有效性。

10.3.2 Methodology & Solution approaches

- 将观测时段划分为 H 个区间，对于第 h 个区间，记待校正参数为 β_h ，仿真生成的 OD 需求（非路径需求）为 x_h ，仿真观测的交通状态为 \tilde{M}_h ，实际观测的交通状态为 M_h ，则仿真校正问题可以表示为

$$\min z(x_1, \dots, x_H, \beta_1, \dots, \beta_H) = \sum_{h=1}^H \left[z_1(M_h, \tilde{M}_h) + z_2(x_h, x_h^a) + z_3(\beta_h, \beta_h^a) \right] \quad x_h \in [l_h^x, u_h^x], \quad \beta_h \in [l_h^\beta, u_h^\beta]$$

上式中 z_1, z_2, z_3 为误差函数， x_h^a, β_h^a 分别为 x_h, β_h 的先验值（经验值）；

- 实际的 OD 需求与路段流量间存在复杂的非线性关系，传统研究多基于线性模型拟合未知的 OD 需求 x_h 与可观测的路段流量 y_h ：

$$y_h = \sum_{p=h-p'}^h \alpha_h^p x_p + v_h$$

上式中 α_h^p 为 p 时刻发生的需求至 h 时刻的分配矩阵， p' 为考虑的最大行程时间， v_h 为随机误差；

- 因为微观仿真校正问题为复杂非线性问题，且无解析形式，因此传统的基于梯度的优化方法并不适用，而应采用直接基于目标函数取值的优化方法。具体地，理想的优化方法应满足以下要求：

- 具有良好的全局搜索能力；
- 可实现无梯度优化；
- 收敛速度较快，且对问题规模不敏感。

- 一些基于群体的 (population-based) 仿真优化算法如复合形法 (complex method)⁶ 和 SNOBFIT 算法 (stable noise optimization by branch and fit) 均无需梯度信息进行优化，因此理论上可用于仿真校正问题，但相关算法尚未被用于复杂大规模交通问题；

- 因为仿真校正的目标在于迭代优化大规模随机仿真平台的输出值，因此随机近似算法同样适用。有限差分随机逼近算法 (finite difference stochastic approximation, FDSA) 单独随机扰动每一变量后比较目标函数取值从而确定搜索方向，而且效果较好，但的算法消耗较大：假设需校正 n 个参数，则每计算一次梯度至少需要计算 $n+1$ 次目标函数。相比于 FDSA 算法，同步扰动随机近似算法 (simultaneous perturbation stochastic approximation, SPSA) 同时对所有变量进行扰动，因而极大地降低计算消耗：无论 n 的取值，算法每一轮优化仅需计算 2 次目标函数（详见第 19.2 节）。SPSA 算法对大规模问题的应用效果同样已被证实。

10.3.3 Case study & Calibration results

- 以纽约州下韦斯特切斯特县 (Lower Westchester County) 的一处路网作为示例，对 MITSimLab 微观交通仿真器进行校正；
- 首先对实测数据进行预处理：
 - 实测数据的集计程度并不一致。大多数观测数据以小时为单位进行集计，而少数数据以 15 分钟为单位。将所有数据的集计单位统一为 15 分钟：
 - 首先将以小时为单位的集计数据瓶颈平均分配至四个 15 分钟，进一步地进行数据平滑；
 - 平滑时将控制位置相近的探测器分为一组，统一对每一组内的探测器集计数据进行平滑。假设组内有 n 个探测器，考虑一个 m 小时的观测区间，对观测区间数据重新以 15 分钟为单位采样，则每个探测器有 $4m$ 项数据，组内所有探测器数据共组成 $n \times 4m$ 矩阵；
 - 计算矩阵每一列向量的 L2 范数，得到 $1 \times 4m$ 的 L2 范数行向量。基于 S-G 滤波器（见第 21.15.1 节）对 L2 范数行向量进行平滑，得到平滑向量；
 - 将平滑向量除以原始 L2 范数行向量，得到平滑系数行向量。在原始 $n \times 4m$ 数据矩阵的基础上乘上平滑系数，即得到一组各探测器的平滑结果。

⁶复合形法是一种适用于求解有约束问题的优化方法。定义复合形为 n 维空间中以 $k \in [n+1, 2n]$ 个点围成的超多面体。算法的基本优化思路是：在可行域内构造一初始复合形，计算并比较各点目标函数值，将其中较差的定点替换为可行域中新的能使目标函数改善的点，从而完成复合形的更新。通过多次迭代后复合形将不断变形缩小并收敛至最优解。算法不需梯度信息，仅通过比较目标函数取值确定搜索方向，算法比较简单，对目标函数要求不苛刻。

- 区分工作日与周末（假期）的探测数据。各工作日的需求模式基本一致。只校正典型工作日下的微观仿真；
 - 研究路网轻重混合交通较严重，不同车辆类型具有不同的路径选择模型（大型车限行等），区分小汽车和大卡车构建各自的OD矩阵进行多类别校正。因为探测器数据只能得到断面交通流组成，无法得到各OD对的交通流组成，因此以每一时间间隔（15分钟）内的平均大车比例作为大车系数，等比例得到小车和大车的OD矩阵。
3. 记观测样本量为 N , n 时段的观测值和仿真值分别为 Y_n^o , Y_n^s 。考虑多种类型的误差指标以捕捉仿真结果不同方面的误差特征：
- 以标准化均方根误差 (normalized root-mean-square error, RMSN) 和均方根百分误差 (root-mean-square percent error, RMSPE) 量化仿真的全局误差。指标更关注 Y_n^s 相对于 Y_n^o 的偏离程度，而不考虑 Y_n^s 相对于 Y_n^o 的偏离方向

$$RMSN = \sqrt{\frac{N \sum_{n=1}^N (Y_n^s - Y_n^o)^2}{\sum_{n=1}^N Y_n^o}} \quad RMSPE = \sqrt{\frac{1}{N} \sum_{n=1}^N \left[\frac{Y_n^s - Y_n^o}{Y_n^o} \right]^2}$$

- 以平均百分误差 (mean percent error, MPE) 和泰尔不平等系数 (Theil's inequality coefficient) U 量化相对误差，同时考虑 Y_n^s 相对于 Y_n^o 的偏离程度和偏离方向

$$MPE = \frac{1}{N} \sum_{n=1}^N \frac{Y_n^s - Y_n^o}{Y_n^o} \quad U = \frac{\sqrt{N \sum_{n=1}^N (Y_n^s - Y_n^o)^2}}{\sqrt{\frac{1}{N} \sum_{n=1}^N (Y_n^s)^2} + \sqrt{\frac{1}{N} \sum_{n=1}^N (Y_n^o)^2}}$$

$U \in [0, 1]$ ，当 $U = 0$ 时表示仿真结果与实际观测完全一致。泰尔不平等系数还可进一步拆分为三种指标以量化不同类型的误差：偏差项 U^M 、方差项 U^S 和协方差项 U^C

$$U^M = \frac{(\bar{Y}^s - \bar{Y}^o)^2}{\frac{1}{N} \sum_n (Y_n^s - Y_n^o)^2} \quad U^S = \frac{(s^s - s^o)^2}{\frac{1}{N} \sum_n (Y_n^s - Y_n^o)^2} \quad U^C = \frac{2(1-\rho)s^s s^o}{\frac{1}{N} \sum_n (Y_n^s - Y_n^o)^2} \quad U^M + U^S + U^C = 1$$

上式中 \bar{Y}, s 分别表示观测样本或仿真样本的均值与方差， ρ 为观测样本与仿真样本的协方差。根据定义， U^M, U^S, U^C 三项之和为1。其中偏差项 U^M 量化系统误差，方差项 U^S 反映仿真结果可在多大程度反映实际结果的方差， U^M, U^S 应尽可能接近0。协方差项 U^C 表示 U^M, U^S 外的其它误差，因此应尽可能接近1；

- 当网络中包含多种道路类型时，上述所列指标的有效性可能降低，GEH统计量则可有效避免该问题。统计量计算误差时不仅考虑 Y_n^s 与 Y_n^o 的偏差和真值 Y_n^o ，还考虑了估计值 \hat{Y}_n^s

$$GEH = \sqrt{\frac{2(Y_n^s - \hat{Y}_n^s)^2}{Y_n^s + \hat{Y}_n^s}}$$

GEH统计量多用于评价路段流量的误差，一般认为取值小于5表示仿真流量与实际流量较接近。FHWA指南要求良好的仿真应使得至少85%的路段的GEH小于5。

4. 校正对象为时变OD矩阵（以15分钟为单位），矩阵中包含482个OD对。校正时区分小车和大卡车。
 基于SPSA算法（详见第19.2节）进行优化；
 5. 校正结果证明所提校正方法可得到与实测交通状态匹配的时变OD矩阵；

10.3.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
surveillance	监控 (n)	prevalence	流行的	plausible	可信的
penalize	惩罚 (v)	steer	操控 (v)	foregoing	上述的

10.4 Reducing the Dimension of Online Calibration in Dynamic Traffic Assignment Systems (TRR, 2017)

ABSTRACT

- □ ×

因为在线标定的计算复杂度，相关研究多局限于小型网络。本文基于主成分分析对在线标定问题进行降维，从而克服上述问题。具体地，将OD需求实时估计问题由需求的主成分表示，主成分可以表示需求的空间结构特征。基于新加坡高速路网实测数据进行验证。发现所提算法仅以2%的估计误差为代价使问题复杂度降低了50倍，并使得计算时间降低了100个数量级。目

10.4.1 Introduction & Literature review

1. 一般地，可将仿真标定划分为离线标定和在线标定两类：
 - 离线标定旨在使得仿真器符合典型时段内的交通状态，以服务中短期管控策略制定；
 - 在线标定实时地对离线标定的结果进行进一步标定使得仿真器符合实时交通状态，以服务实时管控策略制定。
2. 在线仿真标定的对象主要包括OD需求与供给模型参数，已有研究的算法主要包括最小二乘、广义最小二乘、卡尔曼滤波、动态规划、反馈控制等等；而近期部分研究则尝试将多源数据引入在线标定，包括车牌识别数据、蓝夜数据、GPS数据等等；
3. 为降低问题的复杂度，一部分研究对网络进行分解（network decomposition），另一部分研究则基于主成分分析对标定算法进行降维；
4. 总而言之，在线标定相关研究的关注点主要集中于算法效率提升与多源数据融合两个方面。但现有在线标定算法应用于大规模网络仍需要较高的计算资源，而且在OD在线标定时往往只考虑了需求的时间相关性，忽略了其空间结构特征；
5. 本文考虑到大规模高效在线标定的迫切需求，研究OD需求的在线标定，但所提算法也可用于在线标定其它参数。本文目标具体包括：
 - 基于主成分分析降低OD需求的维度；
 - 将在线标定问题由主成分进行表示；
 - 基于实测数据测试所提算法，并与传统算法对比。

10.4.2 Online calibration: problem formulation

1. 将研究范围 T 划分为 $h = 1, 2, \dots, n$ 个时段。记交通网络为 $G(N, L, S)$ ，其中 N 为节点集合（节点数 n_N ）， L 为连边集合（连边数 n_L ）， S 为子区集合（子区数 n_S ），假设子区中的所有路段具有几何一致性。记OD对集合为 K ，包含 n_K 对OD对；
2. 首先给出仿真在线标定问题的基本数学形式。令 π_h 表示仿真时 h 时刻的待标定参数， π_h^a 表示相应参数的先验估计结果。注意到仿真在线标定往往是对离线标定结果的进一步优化，故先验值 π_h^a 即可取离线标定的结果。此时 π_h^a 即为 π_h 的确定提供了一定依据，有

$$\pi_h^a = \pi_h + \eta_h$$

上式被称为直接量测方程 (direct measurement equation)，因其直接提供了待标定参数的初步估计。式中 η_h 为随机误差，其大小决定了离线标定 π_h^a 结果对 π_h 在线标定的贡献度。与此同时， π_h 的在线标定结果还取决于另一类指标，即基于 π_h 的仿真交通状态 $M_h^s = M(\pi_h)$ 与真实观测状态 M_h^o 间的误差 ζ_h

$$M_h^o = M(\pi_h) + \zeta_h$$

上式被称为间接量测方程 (indirect measurement equation)，因其提供了基于仿真结果间接估计待标定参数的方法。显然， π_h 的在线标定结果 $\hat{\pi}_h$ 应使得误差 η_h, ζ_h 均取得最小

$$\hat{\pi}_h = \arg \min_{\pi_h} N(\eta_h) + N(\zeta_h) = \arg \min_{\pi_h} N(\pi_h^a - \pi_h) + N(M_h^o - M(\pi_h))$$

上式中 $N(\cdot)$ 为误差量化函数。若简单地令 $N(\cdot)$ 表示均方误差，则上式实际上即是求解最小二乘问题

$$\hat{\pi}_h = \arg \min_{\pi_h} (\pi_h^a - \pi_h)^T (\pi_h^a - \pi_h) + (M_h^o - S(\pi_h))^T (M_h^o - M(\pi_h))$$

然而，普通最小二乘不止假设误差 η_h, ζ_h 服从正态分布，还要求同方差假设⁷。但因为在线标定时处理的数据多为时序数据，往往不能完全满足同方差假设，故不妨分放松假设允许异方差，则上式转化为广义最小二乘问题 (generalized least squares, GLS)

$$\hat{\pi}_h = \arg \min_{\pi_h} (\pi_h^a - \pi_h)^T \Omega_{\eta,h}^{-1} (\pi_h^a - \pi_h) + (M_h^o - M(\pi_h))^T \Omega_{\zeta,h}^{-1} (M_h^o - M(\pi_h))$$

上式中 $\Omega_{\eta,h}, \Omega_{\zeta,h}$ 分别为误差 η_h, ζ_h 的协方差矩阵；

3. 考虑在线 OD 标定的具体场景，令 x_h, x_h^a 分别表示 h 时刻仿真输入的 OD 需求和先验， $y_h^s = y(x_h), y_h^o$ 分别表示 h 时刻仿真和观测得到的网络流量，则在线 OD 标定问题则可建模为如下广义最小二乘形式

$$\hat{x}_h = \arg \min_{x_h} (x_h^a - x_h)^T \Omega_{\eta,h}^{-1} (x_h^a - x_h) + (y_h^o - y(\pi_h))^T \Omega_{\zeta,h}^{-1} (y_h^o - y(\pi_h)), \quad x_h \geq 0$$

上式以实时仿真 OD 需求 x_h 为决策变量，求解上式只需离线标定的先验 OD 需求和用于拟合的观测网络流量两类数据。为充分利用积累的其它数据，记 x_h^H 为 h 时段的历史观测真实需求，令 $\Delta x_h = x_h - x_h^H$, $\Delta x_h^a = x_h^a - x_h^H$, $\Delta y_h^o = y_h^o - y(x_h^H)$ ，并以 Δx_h 为决策变量。进一步为避免求解非线性问题，不妨令

$$y_h^o = y(x_h) + \zeta_h = y_h^H + \sum_{i=h-p}^h A_i^h (x_i - x_i^H) + v_h \implies \Delta y_h^o = \sum_{i=h-p}^h A_i^h \Delta x_i + v_h$$

式中系数矩阵 A_i^h 基于历史数据预标定，则有线性化后以 Δx_h 为决策变量的在线 OD 标定问题的最小二乘形式表示为

$$\hat{x}_h = \arg \min_{\Delta x_h} (\Delta x_h^a - \Delta x_h)^T \Omega_{\eta,h}^{-1} (\Delta x_h^a - \Delta x_h) + \left(\Delta y_h^o - \sum_{i=h-p}^h A_i^h \Delta x_i \right)^T \Omega_{v,h}^{-1} \left(\Delta y_h^o - \sum_{i=h-p}^h A_i^h \Delta x_i \right), \quad x_h \geq 0$$

引入 Δx_h 作为决策变量而非直接 x_h 除了可引入历史数据中的先验信息外，也有助于改善变量分布的有偏性（交通流变量往往是有偏分布，影响大多数统计方法的效果）；

4. 按上式在线标定仿真 OD 需求，需要预先确定 $\Omega_{\eta,h}, \Omega_{v,h}, A_i^h$ 等大量参数。其中矩阵 $\Omega_{\eta,h}$ 便被假设为对角阵以简化计算，这意味着假设 OD 需求间不相关，从而影响在线标定结果，本文所提方法即有助于解决这一问题。

10.4.3 PC-based calibration

1. 基于主成分分析，将原广义最小二乘问题的决策变量由 OD 需求变为 OD 需求的主成分，有效地减少了 OD 在线标定问题的维度；
2. 所谓“主成分”是可解释 OD 需求主要变异性的一组向量线性组合。各主成分间相互独立，因此以其为决策变量求解广义最小二乘问题时可假设协方差矩阵为对角阵；
3. 为构建 OD 需求的主成分，需预先采集或离线标定大量 OD 数据集 X ，记 OD 矩阵 X 维度为 $n_p \times n_K$ ，其行向量为某一时间切片的 OD 需求， n_K 为 OD 对数目。将 OD 矩阵 X 中心化得到矩阵 \tilde{X} ，并对其进行奇异值分解 (singular-valued decomposition)

$$\tilde{X} = U \Sigma V^T$$

其中矩阵 U, V 分别称为左奇异矩阵和右奇异矩阵，维度分别为 $n_p \times n_p, n_K \times n_K$ ，均为单位正交阵（即 $UU^T = I, VV^T = I$ ）；矩阵 Σ 为 $n_p \times n_K$ 的对角阵，其对角线元素非负且依次递减，称为奇异值。奇异值分解是计算 PCA 的快速方法，分解后的右奇异矩阵 V 的列向量 V_i 即对应矩阵 \tilde{X} 的主成分的线性组合系

⁷即各样本的误差互相独立，协方差矩阵为对角元素相等的对角阵。关于普通最小二乘的基本假设详见第 15.1 节。

数。假设仅取前 n_d 个主成分，则构建矩阵 $\tilde{V} = [V_1, V_2, \dots, V_{n_d}]$ ，则对于 OD 需求向量样本 x ，其前 n_d 个主成分 $z = \tilde{V}^T x$ ，对应地可以由 n_d 维主成分向量 z 反推 n_K 维 OD 向量 x

$$x \simeq \tilde{V}z$$

4. 以 OD 需求主成分向量 z 替代 OD 需求向量 x 改写上述 OD 在线标定问题数学模型，则首先直接量测方程和间接量测方程变为

$$\begin{cases} \Delta x_h^a = \Delta z_h + \eta_h \\ \Delta y_h^o = \sum_{i=h-p}^h A_i^h \Delta x_i + v_h \end{cases} \Rightarrow \begin{cases} \Delta z_h^a = \Delta z_h + w_h \\ \Delta y_h^o = \sum_{i=h-p}^h A_i^h \tilde{V} \Delta z_i + v_h \end{cases}$$

式中 $\Delta z_h = z_h - z_h^H$, $\Delta z_h^a = z_h^a - z_h^H$, $w_h = \tilde{V}^T \eta_h$ ，令 Δz_h 为决策变量，则原广义最小二乘问题变为

$$\hat{z}_h = \arg \min_{\Delta z_h} (\Delta z_h^a - \Delta z_h)^T \Omega_{w,h}^{-1} (\Delta z_h^a - \Delta z_h) + \left(\Delta y_h^o - \sum_{i=h-p}^h A_i^h \tilde{V} \Delta z_i \right)^T \Omega_{v,h}^{-1} \left(\Delta y_h^o - \sum_{i=h-p}^h A_i^h \tilde{V} \Delta z_i \right), \quad \tilde{V} z_h \geq 0$$

式中 $\Omega_{w,h}$ 为误差 w_h 的协方差矩阵，因为 $w_h = \tilde{V}^T \eta_h$ ，则有 $\Omega_{w,h} = \tilde{V}^T \Omega_{\eta,h} \tilde{V}$ 。在实际标定中，可自然地假设 $\Omega_{w,h}$ 为对角阵，因为各主成分间线性无关；

5. 对 OD 需求的主成分而非需求本身进行标定，在减少了标定问题维度的同时，也有助于捕捉各 OD 需求间的空间相关性。

10.4.4 Case study on Singapore expressway network

1. 测试时的实时 DTA 仿真器选择 DynaMIT-R，网络包含 4121 对 OD 对。仿真时段为 6:00-12:00，以 5 分钟为区间进行实时 OD 标定，则总的待标定变量数达 $72 \times 4121 = 102312$ ；
2. 基于 2015 年八月至九月间的 30 个工作日的探测器数据，其中前 25 天的数据作为训练集，后 5 天的数据用于测试；
3. 首先估计协方差矩阵 $\Omega_{v,h}, \Omega_{\eta,h}$ 。其中 $\Omega_{v,h}$ 表示探测器的误差，由探测器的置信度决定。 $\Omega_{\eta,h}$ 由基于前 10 天的数据计算的偏差 η_h 确定 $\Omega_{\eta,h} = \frac{1}{n-1} \eta_h \eta_h^T$ ；
4. 进一步地以第 11 至 25 天的探测器数据计算 OD 需求矩阵主成分。发现仅需前 75 个主成分即可解释 OD 矩阵 95% 的方差，因此令决策变量维度 $n_d = 75$ ，与原始的 4121 维比，问题维度降低 54 倍；
5. 以最后 5 天的数据作为测试场景评价在线标定效果。选择标准化根均方误差 (normalized root mean squared error, NRMSE) 和平均绝对百分误差 (mean absolute percentage error, MAPE) 作为评价指标

$$NRMSE = \frac{\sqrt{n \sum_{i=1}^n (y_i - \hat{y}_i)^2}}{\sum_{i=1}^n y_i}, \quad MAPE = \frac{100}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i}$$

6. 测试结果指出，引入 PCA 降维后，在线标定算法在不显著影响标定结果的情况下带来额外的计算效益。

10.4.5 英汉互译

English	Chinese	English	Chinese	English	Chinese
efficacy	功效	discrepancy	不一致	unscented	无味的
drastically	极大地	surveillance	监视	medium-term	中期
precede	先于 (v)				

10.5 Data driven origin-destination matrix estimation on large networks: A joint origin-destination-path-choice formulation (TRC, 2024)

ABSTRACT

提出了一种基于 OD-路径联合选择建模的动态 OD 估计方法。方法的核心是将传统的估计 OD 对需求的问题转化为估计路径需求的问题，同时完成小区和路段层面的交通分配。方法的理论基础为经典宏观双重约束重力模型 (doubly constraint gravity model) 和微观层面的多项 Logit 的 OD 选择模型 (multinomial logit model) 的等价性。直接分配路段需求将导致问题维度激增，但也增加了可用的数据类型——传统的 OD 估计研究需要结合仿真器才能得到路段状态，而将问题扩展后即可直接基于实测的路段速度和流量反推路径需求。若最终构建的方程组仍不满秩，则基于 PCA 算法进行降维。

10.5.1 Introduction

- 提出了面向拥堵路网时变 OD 矩阵估计的方法。大规模路网 OD 估计的难点在于问题的维数与交通小区的数目（正比于网络的节点数）二次正相关，而独立约束的数目则仅与网络规模（连边数、节点数、小区数）线性正相关，导致问题常常不可解。同时在拥堵的影响下，只有部分路段的流量能反映真实的路段需求，进一步减少了有效约束数目；
- OD 矩阵估计问题是一个数据融合问题——基于描述交通需求和状态关系的交通模型和与需求相关的实测数据推断未知 OD 矩阵，包括未来需求预测和历史需求重建两类子问题；
- 未来需求主要基于调查数据和描述活动、土地利用和 OD 需求关系的模型进行预测；
- 历史需求重建是一个典型的逆向工程问题，旨在寻找符合已有流量、速度观测结果的最大似然 OD 矩阵，一般被建模为如下优化问题

$$\hat{X} = \arg \min_X f_1(X, \tilde{X}) + f_2(y(X), \tilde{y})$$

式中 \tilde{X}, \tilde{y} 分别表示先验需求和观测状态； $y(X)$ 表示基于需求生成状态的交通模型；距离函数 f_1, f_2 分别量化重建需求与先验需求和重建状态与观测状态的相似度。通过正则项 f_1 要求重建需求与先验需求接近符合直觉，也有助于减小搜索空间，但如何选择适当的相似度函数并确定正则项的权重则缺少共识；

- 另一个替代方案是通过引入额外的观测数据作为硬约束以缩小 OD 估计解空间并替代正则项 f_1 。注意到 OD 需求重建的根本难点在于路径选择的未知性——正是未知的路径选择行为建立了 OD 需求和路段状态的因果关系，因此可以将 OD 重建问题扩展为路段需求重建问题，借此可引入路段需求非负约束和路径-路段需求一致性约束等一系列基于各种路段级观测数据的约束。相关约束可替换均衡交通分配等假设生成更符合真实数据的结果。因为传统 OD 重建需要小区级别的行程时间等状态数据。为基于观测路段状态集计小区状态，便需要引入额外的假设；
- 研究创新点包括：1) 提出了面向拥堵路网的 OD-路径选择联合时变需求估计方法；2) 所提方法融合了多种异质数据（出行生成量、路径行程时间、路段流量等）；3) 设计了考虑多个路网的敏感性实验。

10.5.2 Methodology

- 首先回顾宏观的双重约束重力模型 (doubly constraint gravity model) 和微观的多项 Logit 的 OD 选择模型 (multinomial logit model)。双重约束重力模型是经典的宏观出行分布 (trip distribution) 方法，旨在基于各小区的出行发生和吸引量反推 OD 矩阵 (见第 10*II 节)，其更新公式为

$$x_{ij} = a_i b_j P_i A_j \exp\{U_{ij}\}, \quad a_i = \frac{1}{\sum_j b_j A_j \exp\{U_{ij}\}}, \quad b_j = \frac{1}{\sum_i a_i P_i \exp\{U_{ij}\}}, \quad U_{ij} = - \sum_b \alpha_b X_b^{ij}$$

式中 x_{ij} 表示自小区 i 至 j 的交通需求； P_i, A_j 分别表示小区 i, j 的出行生成量和吸引量； U_{ij} 为小区 i 至 j 出行的效用，由行程时间、费用等属性 X_b^{ij} 和效用参数 α_b 决定。反复迭代上式直至 x_{ij} 收敛即可得到估计

的 OD 矩阵。可以看到，上式认为交通需求与发生、吸引量成正比，与出行阻抗成反比，分别对应力学中的引力和斥力，故得名“重力”模型。所谓“双重约束”则是因为系数 a_i, b_j 的引入，使得 x_{ij} 的更新恒满足约束 $\sum_j x_{ij} = P_i, \sum_i x_{ij} = A_j$ 。可以严格证明，宏观层面的双重约束重力模型等价于微观层面基于多项 Logit 的 O-D 联合选择（见第 10.6 节），即

$$x_{ij} = \beta_{ij} \sum_i P_i = \beta_{ij} \sum_j A_j, \quad \beta_{ij} = \frac{\exp\{U_{ij}\}}{\sum_i \sum_j \exp\{U_{ij}\}}$$

式中 β_{ij} 表示自小区 i 至 j 的交通需求占比；

2. 基于上述等价关系，便可自然地将选择集从 OD 对扩展至各 OD 路径集合，从微观层面构建一个基于多项 Logit 的 OD-路径联合选择模型进行交通分配，同样等价于宏观层面的双重约束重力模型。记 P_{ij}^k 表示 k 时刻自小区 i 至 j 的路径集合； N_{ij} 为小区 i 至 j 的路径总数； $p_{ijn}^k \in P_{ij}^k$ 表示第 n 条路径；决策变量 x_{ijn}^k 表示对应路径的需求； p_{ijn}^k, x_{ijn}^k 可进一步简写为 p_r^k, x_r^k 。需要说明的是，所提 Logit 模型并不区分 OD 选择和路径选择的层次关系，且出发时间并不扩展至选择集中，但最短路径集的计算将区分不同时间的影响；
3. 进一步介绍 x_{ijn}^k 的求解方法。首先确定路径效用 U_r^k 的形式。认为路径效用应该具有路段可加性，即各路段的效用组成共同构成路径效用。研究定义为

$$U_r^k = -[a_0^{ij} + \alpha_\omega \omega_r^k + \alpha_\tau \tau_r^k] + \alpha_{ps} \ln PS_r, \quad \omega_r^k = \sum_{a=1}^{N_a} \delta_{p_r^k}^a \omega_a^k, \quad \tau_r^k = \sum_{a=1}^{N_a} \delta_{p_r^k}^a \tau_a^k, \quad \tau_a^k = \frac{l_a}{u_a^k} + \frac{c_a^k}{\eta_\tau}$$

其中 a_0^{ij} 表示常效用，只与起讫点有关而与具体路径无关； ω_r^k 表示由路径功能属性（道路类型、车道数、通行能力等）决定的效用； τ_r^k 表示路径的广义行程时间，由各路段的广义行程时间 τ_a^k 决定，考虑路段长度 l_a 、行程速度 u_a^k 、行程收费 c_a^k 和费用的时间价值 η_τ ； PS_r 表示路径尺度系数 (path-size factor)，因为随着选择集包含所有 OD 的所有有效路径，路径间的重叠现象将更为明显，引入路径尺度系数可修正选项重叠对离散选择模型基本假设的影响（见第 10*I 节）；

4. 进一步介绍有效路径集的构建。研究假设各路段速度和收费已知，故路径集可预先计算。关键超参为 OD 对间的路径数 N_{ij}^* ，可设为定值 $\forall N_{ij}^* = N^*$ 后直接基于最短路算法构建有效路径集。但该假设并不合理，且由此得到的有效路径集可能存在大量重复路段，增大后续路径尺度系数的计算成本。故研究采用两种替代的最短路算法——路段惩罚算法 (link penalty algorithm, LP) 和 ESX 算法，在预设 N^* 的前提下两种算法均可自适应地确定每个 OD 对间的实际路径数，并避免路径重叠：

- 对于路段惩罚算法，给定预设的最短路数目 N^* ，算法首先基于 Dijkstra 算法（见第 18.3 节）计算每个 OD 对间的第一条最短路。每一条被选择的路段的阻抗将被乘上超参 λ 后进行第二轮最短路搜索。若第二条最短路与第一条一致，则不更新有效路径集合，否则将其加入集合。重复迭代 N^* 次后得到有效路径集合即为最终集合。显然算法可以避免某些高效用路段被频繁选择降低路径重复，也能避免生成过多的低效用路径；
- ESX 算法的思路与路段惩罚算法接近，但更为激进。在每次基于 Dijkstra 算法计算每个 OD 对最短路之后，算法直接去除网络中效用最高的连边进行新一轮的最短路计算。

理论上基于有效路径集合可直接估计先验 OD 需求 \check{x}_{ij}^k

$$\check{x}_{ij}^k = \sum_n \left(\beta_{ijn}^k \sum_{i'} P_{i'}^k \right) = \sum_n \left(\beta_{ijn}^k \sum_{j'} A_{j'}^k \right), \quad \beta_{ijn}^k = \frac{\exp\{U_{ijn}^k\}}{\sum_{i'} \sum_{j'} \sum_{n'} \exp\{U_{i'j'n'}^k\}}$$

然而该式得到的需求只是先验，仅基于观测速度生成，无法保证与观测的路段流量和行程时间匹配；

5. 进一步介绍在流量和行程时间约束下后验 OD 需求的估计方法。记 $m \geq k$ 时刻经过路段 a 的路径集合为 $P_a^{k \leq m}$ ，路段 a 流量为 y_a^m 。则当且仅当以下条件满足时 y_a^m 才可反映 $[k, m]$ 时段内邻近路段交通需求的大小：
 - m 时刻路段 a 无拥堵；
 - 在 $[k, m]$ 时段内所有路径 $p_r^k \in P_a^{k \leq m}$ 中于路段 a 上游的路段均无拥堵；
 - 记 $TT_{r|a}^k$ 表示从 k 时刻出发沿路径 r 至 a 的行程时间，则要求相关需求应于 m 时刻左右到达，即 $TT_{r|a}^k / \Delta t \approx m - k$ 。 Δt 为单位时间长度。

前两条件使得观测流量 y_a^m 不受自身或上游拥堵的影响，最后的条件则用于满足 **FIFO** 规则和观测行程时间约束。由此即可构造如下约束

$$y_a^m = \sum_{k=m-\frac{TT_{\max}}{\Delta t}}^m \sum_{r \in P_a^k} \delta_{r|a}^{mk} x_r^k, \quad \delta_{r|a}^{mk} = \begin{cases} 0 & \left| \frac{TT_{r|a}^k}{\Delta t} - (m-k) \right| \geq \epsilon_{TT} \\ 1 & \text{else} \end{cases}$$

式中 TT_{\max} 表示从任意起点到达路段 a 的最大行程时间； $\delta_{r|a}^{mk}$ 用于区分于 k 时刻沿路径 r 出发的需求能否于 m 时刻到达路段 a ； ϵ_{TT} 为一个取值较小的超参。进一步将上述约束自变量从路径需求改为 OD 需求 x_{ij}^k ，并加入关于交通小区出行发生量和吸引量的约束，可以构建估计路网 OD 需求的方程组

$$\begin{cases} \sum_i \sum_j \sum_{k=m-\frac{TT_{\max}}{\Delta t}}^m \sum_{n=1}^{N_{ij}} \delta_{r|a}^{mk} \beta_{ijn}^k x_{ij}^k = y_a^m \\ \sum_j x_{ij}^k = P_i^k \\ \sum_i x_{ij}^k = A_j^k \end{cases}, \quad \text{s.t. } \forall x_{ij}^k \geq 0$$

即求解一个带非负约束的线性方程组 $CX = b, X \geq 0$ ；

6. 可以看到，上述方程组的约束数目与交通小区数和不受拥堵的路段数线性正相关，而未知量的数目与交通小区数二次正相关。因此对于大规模拥堵路网该方程组可能存在无数个最优解。**为此基于 PCA 算法（第 15.2.1 引入额外的先验约束（相关思路可参考第 10.4 节）。**具体地，对于上文直接基于 β_{ijn}^k 计算的先验 OD 矩阵 $\check{X} \in \mathbb{R}^{N_k \times N_x \times N_x}$ ，进行 PCA 分解近似为

$$\check{X} \simeq ZV^\top + \mu_X \implies \check{x}_{ij}^k \simeq \sum_{p=1}^{N_p} z_p^k v_{ij}^p + \mu_{ij}$$

上式本质上是将 k 时刻的任意先验 OD 需求 \check{x}_{ij}^k 近似为 N_p 个元素 z_p^k 的线性组合，其中 μ_X 表示 \check{X} 的均值。不妨固定 μ_X 和主成分矩阵 V 增加先验约束 $x_{ij}^k = \sum_{p=1}^{N_p} z_p^k v_{ij}^p + \mu_{ij}$ 并假设 z_p^k 未知，则解空间缩小至 $\mathbb{R}^{N_k \times N_p}$

$$\begin{cases} \sum_i \sum_j \sum_{k=m-\frac{TT_{\max}}{\Delta t}}^m \sum_{n=1}^{N_{ij}} \sum_{p=1}^{N_p} \delta_{r|a}^{mk} \beta_{ijn}^k v_{ij}^p z_p^k = y_a^m - \sum_i \sum_j \sum_{k=m-\frac{TT_{\max}}{\Delta t}}^m \sum_{n=1}^{N_{ij}} \delta_{r|a}^{mk} \beta_{ijn}^k \mu_{ij} \\ \sum_j \sum_{p=1}^{N_p} v_{ij}^p z_p^k = P_i^k - \sum_j \mu_{ij} \\ \sum_i \sum_{p=1}^{N_p} v_{ij}^p z_p^k = A_j^k - \sum_i \mu_{ij} \end{cases}$$

基于普通最小二乘算法 (ordinary least square, OLS) 解得 z_p^k 后即可反推 x_{ij}^k 。需要说明的是因为 z_p^k 不要求非负，故上述线性方程组无非负约束。在反解 x_{ij}^k 后启发性地令下限为 0 即可。

10.5.3 Case setup & Case study results

- 首先比较经典 K 最短路算法和研究考虑的路段惩罚算法、ESX 算法的计算性能和解的质量。令每个 OD 对的最短路数目 $K = 10$ 。以五种经典的静态交通分配网络——Sioux falls、Anaheim、Winnipeg、Barcelona、Chicago-Sketch——作为测试路网，路网的拓扑结构、静态 OD 需求和静态分配结果均已知。但因为缺乏动态需求，相关路网不用于后续实验：
 - 首先比较计算复杂度。发现路段惩罚算法具有显著最高的计算效益。对于小规模网络，K 最短路算法的效率优于 ESX 算法，但随着网络和 OD 规模的增大，其时间开销逐渐超过后者。表明路段惩罚算法和 ESX 算法均具有更强的可扩展性；

- 另外比较路段惩罚算法和 ESX 算法生成路径的平均行程成本，发现前者可生成平均行程成本更低的路径集合，即路径的平均质量更高；
- 最后基于平均绕行比例比较生成路径的空间分布。定义某一路径的绕行比例为其里程与相应 OD 最短路里程的比值。发现路段惩罚算法较 ESX 算法生成了平均绕行比例更低的路径集，且对于不同路网解的质量基本稳定。

综上，发现路段惩罚算法较 ESX 算法具有全方面的优势；

2. 进一步验证在 OD-路径联合建模场景下双重约束重力算法与 NML 算法的等价性。考虑 Four-pairs 和 Santander 两个网络作为测试路网：

- Four-pairs 网络是一个仅包含两个起点和两个终点的小型路网，共有八条路径连接四个 OD 对，涉及主干路和次干路两种道路类型，仅一条路段设有流量检测器。对四个 OD 对，基于扰动正弦函数生成十分钟粒度的两小时动态需求，并假设时变 OD 矩阵估计的时间精度为五分钟。考虑以下三种效用函数类型——仅考虑行程时间；考虑行程时间和道路属性；考虑行程时间、道路属性和路径常数

$$U_{ijn}^k = \begin{cases} -\alpha_\tau \tau_{ijn}^k + \alpha_{ps} \ln PS_{ijn} \\ -\left(\alpha_\tau \tau_{ijn}^k + \sum_{u=1}^2 \alpha_w^u w_{ijn}^u \right) + \alpha_{ps} \ln PS_{ijn} \\ -\left(\alpha_{ijn}^0 + \alpha_\tau \tau_{ijn}^k + \sum_{u=1}^2 \alpha_w^u w_{ijn}^u \right) + \alpha_{ps} \ln PS_{ijn} \end{cases}$$

式中 w_{ijn}^u 表示几种类型路段里程占路径总里程的比例；

- Santander 路网是一个大型城市路网，具有四种道路类型共 4205 条路段，其中 295 条设有流量检测器。考虑高峰时段内 4 小时的真实流量，时间精度为五分钟。考虑如下效用函数

$$U_{ijn}^k = -\left(\alpha_{ijn}^0 + \alpha_\tau \tau_{ijn}^k + \sum_{u=1}^4 \alpha_w^u w_{ijn}^u \right) + \alpha_{ps} \ln PS_{ijn}$$

以两个网络为场景分别基于双重约束重力模型和 Logit 算法计算各路径的需求流量，发现两算法的结果完全一致，经验地表明其等价性在将选择集扩展至路径选择后仍成立；

3. 同样考虑 Four-pairs 和 Santander 网络，验证所提 OD 估计算法。在已知路网拓扑和动态 OD 的前提下，基于 Aimsun Next 软件仿真结果作为真实状态。为增加真实性，对真实数据加入 20% 噪声以标定 Logit 效用函数参数。对于 Santander 网络，每个 OD 对仅考虑一条最短路径，并只取 PCA 分解的第一个主成分，本例中仅第一个主成分便表述了近 100% 的先验 OD 需求方差。所提算法在两个案例中均取得了较好的 OD 估计精度。与无降维的参考算法对比，发现 PCA 降维对大规模 OD 估计至关重要；

4. 最后进行敏感性分析：

- 对于 Four-pairs 网络研究不同效用函数类型和需求强度对 OD 估计精度的影响。在基准需求的基础上加减 25% 生成新的需求。发现效用函数类型的影响较小，而需求强度的增加导致误差增大，可能是因为更大的需求加剧了网络拥堵，从而降低了约束的质量；
- 对于 Santander 网络首先评估最短路数目 K 的影响。考虑两种最短路生成算法。发现随着 OD 对最短路数目的增加，路段惩罚算法生成的路径集可使得需求绝对误差稍微下降，但相对误差显著提升；ESX 算法则基本不受影响。可能是因为路径集尺寸的增加导致解空间增大，且每条路径的绝对需求降低，从而在绝对误差基本不变的前提下显著提升相对误差；而 ESX 算法因为采取切除关键路段的方式生成路径集，多次迭代后网络连通性显著下降，使得算法生成的路径集规模随 K 增大收敛，因此过大的 K 值并不会影响路径集，故而对误差影响较小；
- 另外对于 Santander 案例同时考虑最短路数目和效用函数参数误差的影响。发现 OD 估计绝对误差和相对误差均随效用函数参数误差的增大而增大；
- 最后对于 Santander 案例评估 PCA 降维数目影响。发现仅保留 1 个主成分时效果反而最好，随着主成分的增加 OD 估计绝对误差和相对误差均增加。这是因为本例中仅第一个主成分即可捕捉先验

OD 矩阵的绝大多数方差（信息），意味着增加其它主成分不仅不能提供更多有效信息，反而会增大搜索空间降低解的质量。

10.5.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
assimilation	消化、吸收	quadratically	平方地	exacerbate	使恶化
codify	编撰	schematically	概要地	tomography	体层摄影
ex ante	事前	ex post	事后	elaborate	复杂的、详细说明
deterrence	威慑 (n)	consolidate	统一、巩固 (v)	streamline	流线 (n)、使流线型
imperative	迫切的	far-fetched	牵强的	reconcile	调和 (v)
disutility	负效用	discretize	使离散	inadvertently	无意识地
granularity	粒度	hereafter	以后 (adv)、未来 (n)	monotonic	单调的
scrutinize	仔细检查 (v)	exacerbate	恶化		

10.6 Discrete choice theory, information theory and the multinomial logit and gravity models (TRB, 1983)

ABSTRACT

- □ ×

一段时间以来，研究者已发现最大熵估计（最小信息估计）和最大效用估计之间具有明显的相似性，但并未对其深入研究。本研究证实对于多项 Logit 模型最大熵估计和最大效用估计完全一致，且双重约束重力模型等价于基于多项 Logit 的 OD 联合选择模型，意味着双重约束重力模型的结果可以通过微观层面的行为选择建模实现。图

10.6.1 Stochastic utility maximization and the multinomial logit model

1. 多项 Logit(multinomial logit, MNL) 模型是最常采用的离散选择模型。假设总量为 H 的决策者具有同质偏好以及可加的随机误差项；又假设每个出行者面临 J 个独立离散选项，每个选项具有 K 维属性。则每个选项 j 对决策者 h 的效用定义为如下线性形式

$$\hat{U}_j^h = \alpha_{0j} + \sum_{k=1}^K \alpha_k X_{jk}^h + \epsilon_j^h$$

式中 α 表示效用系数，所有决策者具有一致的效用系数； X_{jk}^h 为选项 j 的第 k 个属性对于决策者 h 的取值； ϵ 为随机效用，与选项和决策者有关；

2. 假设所有决策者均选择效用最高的选项，则在随机效用的干扰下，决策者 h 选择选项 i 的概率 P_i^h 定义为

$$P_i^h = \text{Prob.} [\hat{U}_i^h > \hat{U}_j^h; \forall j \neq i]$$

又假设所有随机项 ϵ 独立同服从于众数为 0、方差为 σ^2 的 Gumbel 分布，有如下累积概率函数

$$\text{Prob.} (\epsilon_i^h \leq \epsilon) = \exp \left\{ -\exp \left[-\frac{\pi}{\sqrt{6}\sigma} \epsilon \right] \right\}$$

代入上式有

$$P_i^h = \frac{\exp \left\{ \beta_{0i} + \sum_{k=1}^K \beta_k X_{ik}^h \right\}}{\sum_{j=1}^J \exp \left\{ \beta_{0j} + \sum_{k=1}^K \beta_k X_{jk}^h \right\}}, \quad \beta_{0i} = \frac{\pi \alpha_{0i}}{\sqrt{6}\sigma}, \quad \beta_k = \frac{\pi \alpha_k}{\sqrt{6}\sigma}$$

3. 可基于极大似然法估计参数 β ，即

$$\max_{\beta} \ln \mathcal{L} = \sum_h \sum_j \delta_j^h \ln P_j^h(\beta)$$

式中指示变量 $\delta_j^h = 1$ 表示决策者 h 选择选项 j , 否则取 0。对于上述无约束最优化问题, 只需令一阶偏导为 0, 即

$$\frac{\partial \ln \mathcal{L}}{\partial \beta} = \sum_h \sum_j \delta_j^h \frac{\partial P_j^h(\beta)/\partial \beta}{P_j^h(\beta)} = 0$$

又因为

$$\begin{cases} \frac{\partial P_j^h(\beta)}{\partial \beta_k} = P_j^h(\beta) X_{jk}^h - P_j^h(\beta) \sum_i P_i^h(\beta) X_{ik}^h \\ \frac{\partial P_j^h(\beta)}{\partial \beta_{0j}} = P_j^h(\beta) [1 - P_j^h(\beta)] \end{cases} \implies \begin{cases} \frac{\partial \ln \mathcal{L}}{\partial \beta_k} = \sum_h \sum_j P_j^h(\beta) X_{jk}^h = \sum_h \sum_j \delta_j^h X_{jk}^h \\ \frac{\partial \ln \mathcal{L}}{\partial \beta_{0j}} = \sum_h P_j^h(\beta) = \sum_h \delta_j^h \end{cases}$$

解上述方程组即可得到 MNL 模型系数 β 。方程组等式右边的项具有清晰的物理意义。其中 $\sum_h \sum_j \delta_j^h X_{jk}^h$ 表示观测选择样本数据集的第 k 个属性的取值之和; $\sum_h \delta_j^h$ 表示观测样本中选项 j 被选择的次数;

4. 最后总结 NML 模型的基本假设如下:

- 所有决策者具有相同的线性效用函数, 且效用函数参数不随个体变化;
- 随机效用具有可加性, 并假设独立同服从众数为 0、方差为 σ^2 的 Gumbel 分布;
- 所有出行者基于最大效用原则选择选项。

10.6.2 Information minimization and the multinomial logit model

1. 进一步基于最小信息 (information-minimization) 或最大熵 (entropy-maximization) 原理推导 MNL 模型。此时不再是从微观行为的角度出发, 而是从宏观层面寻找符合观测样本的最大概率事件, 建模为

$$\min_{P_j^h} -\mathcal{E} = \sum_h \sum_j P_j^h \ln P_j^h, \quad \text{s.t.} \quad \begin{cases} \sum_j P_j^h = 1 & \forall h = 1, \dots, H \\ \sum_h P_j^h = \sum_h \delta_j^h & \forall j = 1, \dots, J \\ \sum_h \sum_j P_j^h X_{jk}^h = \sum_h \sum_j \delta_j^h X_{jk}^h & \forall k = 1, \dots, K \end{cases}$$

上式的决策变量为概率 P_j^h , 目标函数为信息熵 (见第 23.9.6 节) 的相反数。发现后两组约束与上一小节基于最大效用预计 MNL 模型参数的方程组一致, 其物理意义分别表示每个选项期望被选择的次数应等于实际被选择的次数、和期望被选选项的每个属性取值之和应等于实际样本集中对应属性的取值之和。因为目标函数为凸函数且约束为线性, 故该问题为凸优化问题, 有唯一最优解;

2. 基于拉格朗日乘子法求解上式。构造拉格朗日函数 L

$$L = \sum_h \sum_j P_j^h \ln P_j^h - \sum_h \theta_h \left[\sum_j P_j^h - 1 \right] - \sum_j \lambda_{0j} \left[\sum_h P_j^h - \sum_h \delta_j^h \right] - \sum_k \lambda_k \left[\sum_h \sum_j P_j^h X_{jk}^h - \sum_h \sum_j \delta_j^h X_{jk}^h \right]$$

按最优性条件 $\frac{\partial L}{\partial P_j^h} = 0$ 有

$$\frac{\partial L}{\partial P_j^h} = 1 + \ln P_j^h - \theta_h - \lambda_{0j} - \sum_k \lambda_k X_{jk}^h = 0 \implies P_j^h = \exp \{-1 + \theta_h\} \exp \left\{ \lambda_{0j} + \sum_k \lambda_k X_{jk}^h \right\}$$

为消去上式的拉格朗日乘子 θ_h , 代入 $\frac{\partial L}{\partial \theta_h} = \sum_j P_j^h - 1 = 0$, 有

$$\exp \{-1 + \theta_h\} = \frac{1}{\sum_j \exp \{ \lambda_{0j} + \sum_k \lambda_k X_{jk}^h \}} \implies P_j^h = \frac{\exp \{ \lambda_{0j} + \sum_k \lambda_k X_{jk}^h \}}{\sum_j \exp \{ \lambda_{0j} + \sum_k \lambda_k X_{jk}^h \}}$$

可以看到，上述形式与上小结基于最大效用和最大似然估计的结果完全一致，仅将选择效用系数 β 改为拉格朗日乘子 λ 。为求解 λ 考虑另外两条一阶最优性条件构建方程组

$$\begin{cases} \frac{\partial L}{\partial \lambda_k} = \sum_h \sum_j P_j^h(\lambda) X_{jk}^h - \sum_h \sum_j \delta_j^h X_{jk}^h = 0 & \forall k = 1, \dots, K \\ \frac{\partial L}{\partial \lambda_{0j}} = \sum_h P_j^h(\lambda) - \sum_h \delta_j^h = 0 & \forall j = 1, \dots, J \end{cases}$$

以上即为基于最大熵估计的 MNL 模型。与微观层面的最大效用建模相比，前者仅基于较简单的假设：

- 假设最可能的选择概率为满足给定约束的前提下使得熵最大的概率；
- 给定约束包括 1) 每个选项期望被选择的次数应等于实际被选择的次数；2) 期望被选选项的每个属性取值之和应等于实际样本集中对应属性的取值之和。

3. 观察从宏观集计层面基于最大熵估计的 MNL 模型和从微观出行选择层面基于最大似然估计的 MNL 模型发现，两者得到的选择概率 $P_i^h(\cdot)$ 具有完全一致的数学形式，并基于完全一致的方程组估计 $P_i^h(\cdot)$ 参数，故两者完全等价：

- 记 β^*, λ^* 分别表示估计的行为选择参数和拉格朗日乘子，则有 $\beta^* = \lambda^*$ ；
- 注意到 $\frac{\partial(-\mathcal{E})}{\partial(\sum_h \sum_j \delta_j^h X_{jk}^h)} = \lambda_k^*$ ，表明选项第 k 维属性的边际效用 β_k^* 等价于 $\sum_h \sum_j \delta_j^h X_{jk}^h$ 对信息熵的边际效应；
- 注意到 $\frac{\partial(-\mathcal{E})}{\partial(\sum_h \delta_j^h)} = \lambda_{0j}^*$ ，表明选项第 j 个选项的基准效用 β_{0j}^* 等价于 $\sum_h \delta_j^h$ 对信息熵的边际效应；
- 选项的任意第 k 维属性对 MNL 模型的贡献既可以用信息熵的变化 $\mathcal{E}^* - \mathcal{E}_k^*$ 描述，也可以用对数似然值的变化（对数似然比）描述 $\ln \frac{\mathcal{L}^*}{\mathcal{L}_k^*}$ ，且两者完全等价；
- 宏微观层面的等价性进一步表明为建模个体的决策行为，仅需宏观集计的观测数据 $\sum_h \sum_j \delta_j^h X_{jk}^h, \sum_h \delta_j^h$ ，而无需每个个体的决策行为 δ_j^h 。这意味着对于出行行为选择建模，若得到了宏观层面的行程时间、成本等集计数据便无需知道每个出行者的行为选择。该结论为减少繁重且高成本的个体出行调查提供了理论依据。

10.6.3 The doubly constrained gravity model is a logit model of joint origin-destination choice

1. 早在 1967 年 Wilson 即证明双重约束重力模型可由最大熵理论导出（见 10*II 节）。而通过本节介绍的宏观维度最大熵理论和微观层面最大效用行为选择的等价性可进一步得出——双重约束重力模型在数学上等价于基于 MNL 的 OD 联合选择。建模如下

$$\min_{P_{ij}} -\mathcal{E} = \sum_i \sum_j P_{ij} \ln P_{ij}, \quad \text{s.t. } \begin{cases} \sum_j P_{ij} = O_i/H & \forall i = 1, \dots, I \\ \sum_i P_{ij} = D_j/H & \forall j = 1, \dots, J \\ \sum_i \sum_j P_{ij} X_{ijk} = \bar{X}_k/H & \forall k = 1, \dots, K \end{cases}$$

式中 P_{ij} 表示交通小区 i, j 间出行占总出行的比例； O_i, D_j 分别表示交通小区 i, j 的出行生成和吸引量； H 表示总出行量； X_{ijk} 表示交通小区 i, j 间出行属性的第 k 维特征。注意到上式不含 $\sum_i \sum_j P_{ij} = 1$ 约束，这是因为在交通分布的特殊场景下恒有 $\sum_i O_i = \sum_j D_j = H$ ，因此前两条约束可使得 $\sum_i \sum_j P_{ij} = 1$ 天然成立，因此求解上式同样为求解 MNL 模型；

2. 构建拉格朗日函数如下

$$L = \sum_i \sum_j P_{ij} \ln P_{ij} - \sum_i \lambda_{oi} \left[\sum_j P_{ij} - O_i/H \right] - \sum_j \lambda_{dj} \left[\sum_i P_{ij} - D_j/H \right] - \sum_k \lambda_k \left[\sum_i \sum_j P_{ij} X_{ijk} - \bar{X}_k/H \right]$$

同样基于最优性条件 $\frac{\partial L}{\partial P_{ij}} = 0$ 有

$$\frac{\partial L}{\partial P_{ij}} = 1 + \ln P_j^h - \lambda_{oi} - \lambda_{dj} - \sum_k \lambda_k X_{ijk} = 0 \implies P_{ij} = \exp \left\{ \lambda_{oi} + \lambda_{dj} + \sum_k \lambda_k X_{ijk} - e \right\}$$

在此基础上参考见 10*II 节介绍的推导过程即可导出双重约束重力模型。

10.6.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
corollary	推论 (n)	culmination	高潮、终点		

10.7 交通分配模型求解

观察上文列举的几种模型发现，交通分配模型主要为非线性模型，具体地目标函数为非线性函数，而约束条件则常常为线性约束。因此经典的交通分配模型求解算法也大多为求解线性约束的非线性问题的算法。

10.7.1 Frank-Wolfe 算法

1. 算法由 Frank 和 Wolfe 于 1956 年提出，是一种求解线性约束的非线性问题的算法，其本质上为梯度下降方法。算法的核心思路是将非线性的目标函数在某个可行解邻域内局部线性化，结合线性约束将非线性规划问题转化为线性规划问题。对于每一轮迭代，基于线性规划问题的最优解得到可行的下降方向并沿下降方向做一维搜索，得到新一轮迭代的可行解；
2. 考虑非线性规划问题

$$\min f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{Ax} = \mathbf{b}, \quad \mathbf{x} \geq 0$$

约束条件为线性，目标函数 f 为可微凸函数，可行域 \mathbf{X} 为凸集。则对于任意可行解 \mathbf{x}^k ，可对目标函数线性化

$$f(\mathbf{x}) \approx f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T (\mathbf{x} - \mathbf{x}^k) \implies \arg \min f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T (\mathbf{x} - \mathbf{x}^k) = \arg \min \nabla f(\mathbf{x}^k)^T \mathbf{x} = \arg \min f^k(\mathbf{x})$$

此时可以将原非线性规划问题转化为线性规划问题，并解得最优解 \mathbf{y}^k

$$\begin{aligned} \min & \quad f^k(\mathbf{x}) \\ \text{s.t.} & \quad \mathbf{Ax} = \mathbf{b}, \quad \mathbf{x} \geq 0 \end{aligned}$$

3. 算法的精髓即在于：

- 若 \mathbf{x}^k 是线性规划问题的最优解 ($\nabla f(\mathbf{x}^k)^T (\mathbf{y}^k - \mathbf{x}^k) = \mathbf{0}$)，则 \mathbf{x}^k 即为原非线性规划问题的最优解；
- 若 \mathbf{x}^k 不是线性规划问题的最优解 ($\nabla f(\mathbf{x}^k)^T (\mathbf{y}^k - \mathbf{x}^k) \neq \mathbf{0}$)，则方向 $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$ 即为使得原目标函数 $f(\mathbf{x})$ 下降的可行方向。

以下证明上述结论：若 \mathbf{x}^k 不是线性规划问题的最优解，则由定义必有

$$f^k(\mathbf{y}^k) < f^k(\mathbf{x}^k) \implies \nabla f(\mathbf{x}^k)^T \mathbf{y}^k < \nabla f(\mathbf{x}^k)^T \mathbf{x}^k \implies \nabla f(\mathbf{x}^k)^T (\mathbf{y}^k - \mathbf{x}^k) < 0 \implies f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T (\mathbf{y}^k - \mathbf{x}^k) < f(\mathbf{x}^k)$$

上式指出了 Frank-Wolfe 算法与梯度下降算法的一致性。 $\mathbf{y}^k - \mathbf{x}^k$ 可理解为梯度下降的步长。又因为可行域为凸集， $\mathbf{y}^k, \mathbf{x}^k$ 均满足线性约束条件，因此 $\mathbf{y}^k - \mathbf{x}^k$ 方向上的任意一点均为问题的可行解。结论得证；

4. 最后总结 Frank-Wolfe 算法的流程：

- 初始化可行解 \mathbf{x}^0 和允许误差 $\varepsilon > 0$ ，进入第一轮迭代；
- 对于第 k 轮迭代，基于原问题可行解 \mathbf{x}^k ，构造近似线性规划问题并求解最优解 \mathbf{y}^k ；
- 构造可行下降方向 $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$ ，判断收敛性：

- 若 $\|\nabla f(\mathbf{x}^k)^T \mathbf{d}^k\| \leq \varepsilon$, 则认为收敛, 输出近似最优解 \mathbf{x}^k ;
- 若 $\|\nabla f(\mathbf{x}^k)^T \mathbf{d}^k\| > \varepsilon$, 则认为尚未收敛, 按下式进行一维搜索, 求解最优 λ^*

$$\lambda^* = \arg \min_{0 \leq \lambda \leq 1} f(\mathbf{x}^k + \lambda \mathbf{d}^k)$$

- 更新原问题可行解 $\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda^* \mathbf{d}^k$, 回到第二步。

- 回顾 Frank-Wolfe 算法发现实际上算法并不完全要求约束为线性, 理论上对于非线性约束也能求解。作为求解非线性规划问题的一类方法, 以上即是 Frank-Wolfe 算法的全部内容。但注意到交通分配问题本质上属于图问题, 而一般优化算法求解图问题的效率普遍低于针对性的图算法, 因此基于 Frank-Wolfe 算法进行交通分配, 需进行进一步变换;
- 以经典的标准静态 Logit 用户均衡问题 (式10.2) 为例, 以 f_r^{ij} 为自变量 (即 $\mathbf{x} = \{f_r^{ij} | \forall i \in I, \forall j \in J, \forall r \in R_{ij}\}$), 对上式求一阶微分线性化, 则得到第 k 次迭代时的目标函数

$$\begin{aligned} \min \sum_{a \in A} \int_0^{v_a} t_a(v) dv &\simeq \min \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} \left(\frac{\partial}{\partial f_r^{ij,k}} \sum_{a \in A} \int_0^{v_a} t_a(v) dv \right) f_r^{ij} \\ &= \min \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} \left(\sum_{a \in A} t_a(v_a^k) \delta_{ar}^{ij} \right) f_r^{ij} = \min \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} T_r^{ij,k} f_r^{ij} \end{aligned}$$

式中 $v_a^k, t_a(v_a^k)$ 分别表示第 k 次迭代时连边 a 的流量和行程时间; δ_{ar}^{ij} 为判别连边 a 是否位于连接 OD 对 $i-j$ 的路径 r 上的指示变量; 因此 $\sum_a t_a(v_a^k) \delta_{ar}^{ij} = T_r^{ij,k}$ 即为路径 r 的行程时间;

- 基于 Frank-Wolfe 算法进行线性化后, 该交通分配问题的目标函数仍是分配各路径流 f_r^{ij} , 但目标函数中决策变量的系数 $T_r^{ij,k}$ 为常数, 相当于路径阻抗与交通需求无关。因此每一轮迭代中对应的交通分配即是一个典型的全有全无分配, 只需基于阻抗 $T_r^{ij,k}$ 更新各 OD 最短路后将需求全部加载即可, 而无需真正求解一个线性规划问题。

10.7.2 partial linearization 算法

- partial linearization 算法最早由 Larsson 等于 1990 年提出⁸。顾名思义, 算法仅对非线性的目标函数的一部分进行线性化 (Frank-Wolfe 算法则是完全线性化)。算法实际上是包括 Frank-Wolfe 算法等在内的多种算法的一般形式。因为算法仅对目标函数部分线性化, 构造得到的近似问题仍属于非线性规划问题, 因此算法并未要求约束条件为线性;
- 考虑连续可微非线性目标函数 $f(\mathbf{x})$ (凸函数), 构造任意形式较为简单的对 \mathbf{x} 连续可微凸函数 $\varphi(\mathbf{x}, \mathbf{y})$ 作为 $f(\mathbf{x})$ 的近似, 则可作如下变换

$$\min f(\mathbf{x}) = \min \varphi(\mathbf{x}, \mathbf{x}^k) + (f(\mathbf{x}) - \varphi(\mathbf{x}, \mathbf{x}^k))$$

\mathbf{x}^k 为原问题可行解。变换后的目标函数由两项组成, 第一项为对原目标函数的近似, 第二项为近似误差;

- 仅对目标函数中的误差项在点 \mathbf{x}^k 处线性化, 得到部分线性化的目标函数 $f_\varphi^k(\mathbf{x})$ 和新问题的最优解 \mathbf{y}^k

$$\begin{aligned} \min \varphi(\mathbf{x}, \mathbf{x}^k) + (f(\mathbf{x}) - \varphi(\mathbf{x}, \mathbf{x}^k)) &\simeq \min \varphi(\mathbf{x}, \mathbf{x}^k) + (f(\mathbf{x}^k) - \varphi(\mathbf{x}^k, \mathbf{x}^k)) + (\nabla f(\mathbf{x}^k) - \nabla \varphi(\mathbf{x}^k, \mathbf{x}^k))^T (\mathbf{x} - \mathbf{x}^k) \\ &= \min \varphi(\mathbf{x}, \mathbf{x}^k) + (\nabla f(\mathbf{x}^k) - \nabla \varphi(\mathbf{x}^k, \mathbf{x}^k))^T \mathbf{x} = \min f_\varphi^k(\mathbf{x}) \end{aligned}$$

\mathbf{y}^k 可通过求解 KKT 条件或基于对偶原理得到。同样地可以证明:

- 若 \mathbf{x}^k 是部分线性规划问题的最优解 ($\nabla f(\mathbf{x}^k)^T (\mathbf{y}^k - \mathbf{x}^k) = \mathbf{0}$), 则 \mathbf{x}^k 即为原非线性规划问题的最优解;
- 若 \mathbf{x}^k 不是部分线性规划问题的最优解 ($\nabla f(\mathbf{x}^k)^T (\mathbf{y}^k - \mathbf{x}^k) \neq \mathbf{0}$), 则方向 $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$ 即为使得原目标函数 $f(\mathbf{x})$ 下降的可行方向。

因此 partial linearization 算法的步骤与 frank-wolfe 算法完全一致;

⁸Larsson T , Rnnqvist M . A method for structural optimization which combines secondorder approximations and dual techniques[J]. Structural optimization, 1993, 5(4):225-232.

4. 算法通过构造较简单的连续可微凸函数 $\varphi(\mathbf{x}, \mathbf{y})$ 将求解较为复杂的非线性凸规划问题转化为多个较简单非线性凸规划问题的近似。实际上可以发现当 $\varphi(\mathbf{x}, \mathbf{y}) = 0$ 时算法即为 **Frank-Wolfe 算法**；当 $\varphi(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\mathbf{x}^T \nabla^2 f(\mathbf{y})\mathbf{x}$ 时算法即为**带约束牛顿算法**；
5. 因为 \mathbf{y}^k 为非线性规划问题的最优解，在一般情况下求解难度仍较高，针对不同的问题可以采取不同的部分线性化方法以降低求解难度。以经典的随机用户均衡问题（式10.4）为例，以 f_r^{ij} 为自变量（即 $\mathbf{x} = \{f_r^{ij} | \forall i \in I, \forall j \in J, \forall r \in R_{ij}\}$ ），将目标函数改写为

$$\min \sum_{a \in A} \int_0^{\nu_a} t_a(\nu) d\nu + \frac{1}{\theta} \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} f_r^{ij} (\ln f_r^{ij} - 1)$$

仅针对目标函数的第一项线性化（相当于令 $\varphi(\mathbf{f}, \mathbf{f}^k) = \frac{1}{\theta} \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} f_r^{ij} (\ln f_r^{ij} - 1)$ ），可以得到第 k 次迭代时的目标函数

$$\begin{aligned} & \min \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} \left(\frac{\partial}{\partial f_r^{ij,k}} \sum_{a \in A} \int_0^{\nu_a} t_a(\nu) d\nu \right) f_r^{ij} + \frac{1}{\theta} \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} f_r^{ij} (\ln f_r^{ij} - 1) \\ &= \min \sum_{i \in I} \sum_{j \in J} \sum_{r \in R_{ij}} \left(T_r^{ij,k} f_r^{ij} + \frac{1}{\theta} f_r^{ij} (\ln f_r^{ij} - 1) \right) \quad T_r^{ij,k} = \sum_a t_a(\nu_a^k) \delta_{ar}^{ij} \end{aligned}$$

上式为严格的凸函数，可由拉格朗日乘数法直接求解最优解 $\bar{f}_r^{ij,k}$ ，且最优解服从 logit 形式

$$\bar{f}_r^{ij,k} = o_i \frac{\exp\{-\theta T_r^{ij,k}\}}{\sum_j \sum_r \exp\{-\theta T_r^{ij,k}\}}$$

同样地，此时每一轮迭代时对应的交通分配即是一个典型的多路径分配，只需基于阻抗 $T_r^{ij,k}$ 更新各 OD 的有效路径后将需求按 MNL 选择概率加载即可，而无需真正求解一个非线性规划问题。

10.7.3 MSA 算法与 MSWA 算法

1. 本节 partial linearization 算法部分证明，宏观层面上的经典随机用户均衡问题对应于微观层面的 MNL 路径选择问题。因此可以通过迭代以下过程实现交通分配，从而避免直接求解待约束的非线性规划问题：

- (a) 基于（初始）路径需求 f_r^i 更新路径效用 $c_r^{i+1} = C(f_r^i)$ ；
- (b) 基于路径选择模型和路径效用分配路径需求 $f_r^{i+1} = F(c_r^{i+1})$ 。

当以上过程收敛时相应的需求分布模式即为平衡模式。可以看到，该过程将求解待约束的非线性规划问题转化为求解不动点问题，即寻找不动点（平衡态路径流） f_r^i ，使得

$$f_r^{i+1} = F(c_r^{i+1}) = F(C(f_r^i)) = \tilde{F}(f_r^i) = f_r^i$$

2. 1951 年，Robbins 和 Monro 提出了一种经典的求解不动点问题的方法——迭代平均法 (method of successive average, MSA)。MSA 算法计算简单，在保证精度的前提下具有良好的收敛性，因此一经提出便得到广泛应用。假设函数 $y = F(x)$ 存在不动点 $x^* = F(x^*)$ ，则对于第 k 次迭代，MSA 算法的计算过程如下：

- 记第 k 次迭代得到的自变量为 x^k ，相应的因变量 $y^{k+1} = F(x^k)$ ，误差 $\varepsilon^{k+1} = y^{k+1} - x^k$ ；
- 以误差 ε^{k+1} 作为更新方向更新自变量 x^{k+1}

$$x^{k+1} = x^k + \chi^k \varepsilon^{k+1} = x^{k+1} = x^k + \chi^k (y^{k+1} - x^k)$$

上式中 χ^k 为第 k 次迭代的更新步长。Robbins 和 Monro 证明，若 χ^k 满足 $\sum_{k \rightarrow \infty} \chi^k = \infty$ 且 $\sum_{k \rightarrow \infty} (\chi^k)^2 = 0$ （或 $\lim_{k \rightarrow \infty} \chi^k = 0$ ），则以上迭代过程基本处处收敛于不动点 $x^* = F(x^*)$ 。Robbins 和 Monro 进一步给出了经典 MSA 算法的 χ^k 计算式

$$\chi^k = \frac{1}{k}$$

3. 将 χ^k 计算式代入 x^{k+1} 迭代式，有

$$x^{k+1} = \frac{1}{k} \cdot y^{k+1} + \frac{k-1}{k} \cdot x^k = \frac{1}{k} (y^{k+1} + y^k) + \frac{k-2}{k} \cdot x^{k-1} = \dots = \frac{1}{k} (y^{k+1} + \dots + y^1)$$

可以看到，经典 MSA 算法中自变量更新值 x^{k+1} 实际上是已有多轮因变量的均值，这也是名字中“平均”的由来。尽管 MSA 算法方便易用，但其收敛速度较慢。这是因为计算均值更新 x^{k+1} 时所有因变量 y^i 具有相同的权重，然而越新的 y^i 越接近不动点，应该具有更大的权重；

4. 2007 年，Liu 等在经典 MSA 算法的基础上提出 MSWA 算法 (method of successive weighted averages, 连续权重平均)，可以看到其核心在于增加了“权重”的概念。MSWA 算法中 x^{k+1} 定义为

$$x^{k+1} = x^k + \frac{k}{\sum_{i=1}^k i} (y^{k+1} - x^k) = \frac{1}{\sum_{i=1}^k i} (k \cdot y^{k+1} + \dots + 1 \cdot y^1)$$

进一步推广 MSWA 算法中关于 χ^k 的计算式，可以得到更一般的 MSWA 算法，显然当 $n = 0$ 时广义 MSWA 算法即退化为经典 MSA 算法

$$x^{k+1} = x^k + \chi^k (y^{k+1} - x^k) \quad \chi^k = \frac{k^n}{\sum_{i=1}^k i^n}$$

已有研究大多证明，MSWA 算法较 MSA 算法于随机用户均衡问题中具有更强的收敛性。

10.8 附录

10*I 基于路径尺度的 Logit 模型 (path-size Logit, PSL)

1. 传统的多项 Logit 模型 (multi-nominal Logit, MNL) 要求选择集满足无关方案独立性 (independence of irrelevant alternative, IIA) 假设。即要求选择集中的方案必须完全无关且互相独立。在此基础上，假设选择集为 C_n ，选项 $i \in C_n$ 的效用值为 V_{in} ，则 MNL 模型认为选择选项 i 的概率 $P(i|C_n)$ 为

$$P(i|C_n) = \frac{\exp\{\mu V_{in}\}}{\sum_{j \in C_n} \exp\{\mu V_{jn}\}}$$

2. 然而 IIA 假设对于部分问题而言过于苛刻。以路径选择问题为例，IIA 假设要求一个 OD 对间的所有候选路径必须完全不重合，限制了 MNL 模型于此类问题的应用。为此，部分研究在 MNL 模型的效用函数增加一项路径尺度系数 (path-size factor)，从而得到基于路径尺度的 Logit 模型 (path-size Logit, PSL)，以拓展传统 MNL 模型的应用范围。在 PSL 模型下，选择选项 i 的概率 $P(i|C_n)$ 为

$$P(i|C_n) = \frac{\exp\{\mu(V_{in} + \ln PS_{in})\}}{\sum_{j \in C_n} \exp\{\mu(V_{jn} + \ln PS_{jn})\}}$$

3. 上式中 PS_{in} 即为路径 i 的路径长度系数，表示路径 i 与路径集 C_n 中其它路径的重合程度。 $0 < PS_{in} \leq 1$ ， $PS_{in} = 1$ 表示路径 i 与其它路径完全不重合。 PS_{in} 计算式如下。其中 Γ_i 为路径 i 所包含的连边集合， l_a, L_i 分别为连边 a 和路径 i 的长度， $L_{C_n}^*$ 为集合 C_n 中的最短路长度，二元变量 $\delta_{aj} \in \{0, 1\}$ 用于表示连边 a 是否位于路径 j 中

$$PS_{in} = \sum_{a \in \Gamma_i} \frac{l_a}{L_i} \frac{1}{\sum_{j \in C_n} \delta_{aj} (L_{C_n}^*/L_j)}$$

4. 考虑网络中连接相同两点的三条长度均为 c 的路径。路径 1 与路径 2、3 完全不重合。路径 2、3 各有长为 δ 的路段不重合，另有长为 $c - \delta$ 的共线段。则三条路线的路径尺度系数分别为

$$PS_1 = \frac{c}{c} = 1 \quad PS_2 = PS_3 = \frac{c - \delta}{c} \times \frac{1}{2} + \frac{\delta}{c} \times \frac{1}{1} = \frac{1}{2} + \frac{\delta}{2c}$$

可以看到，当 $\delta \rightarrow c$ 时，意味着路径 2、3 趋向于完全不重合的两条道路，有 $PS_2, PS_3 \rightarrow PS_1$ ，此时三条道路拥有相同的路径系数，与认知一致。当 $\delta \rightarrow 0$ 时，意味着路径 2、3 趋向于完全重合，网络中仅有两条道路，有 $PS_2 + PS_3 \rightarrow PS_1$ ，此时两条道路拥有相同的路径尺度系数，也与认知一致。因此路径尺度系数的引入和设计是合理的。

10*II 最大熵原理、双重约束重力模型与交通分布预测

1. Wilson 最早将最大熵原理应用于 OD 估计问题中⁹。记 t_{ij} 为从 i 到 j 的交通量，总交通量 $T = \sum_i \sum_j t_{ij}$ 为常数。若假设出行者的目的地选择完全随机，则对于特定的出行 OD 向量 $\{t_k | k = 1, \dots, m \times n\}$ ，基于排列组合公式共有 $W(t)$ 种生成方式

$$W(t) = C_T^{t_1} \cdot C_{T-t_1}^{t_2} \cdots C_{T-\sum_k^{m \times n-1} t_k}^{t_{m \times n-2}} = \frac{T!}{t_1!(T-t_1)!} \cdots \frac{(T-\sum_k^{m \times n-2} t_k)!}{t_{m \times n-1}!(T-t_{m \times n-1})!} = \frac{T!}{\prod_k^{m \times n} t_k!} = \frac{T!}{\prod_{i \in I} \prod_{j \in J} t_{ij}!}$$

2. 显然具有最多生成方式的出行 OD 向量 $\{t_k\}$ 即具有最大的熵，基于最大熵原理，此时的出行 OD 向量即为最可能出行的出行分布情况，因此优化目标为 $\max W(t)$ 。取对数形式将连乘形式转化为连加形式

$$\arg \max \ln W(t) = \arg \max \left\{ \ln T! - \sum_{i \in I} \sum_{j \in J} \ln t_{ij}! \right\} = \arg \min \sum_{i \in I} \sum_{j \in J} \ln t_{ij}!$$

由 Stirling 近似公式 $\ln(x!) \approx x \ln x - x$ ，有

$$\arg \max \ln W(t) = \arg \min \sum_{i \in I} \sum_{j \in J} \ln t_{ij}! = \arg \min \sum_{i \in I} \sum_{j \in J} t_{ij}(\ln t_{ij} - 1)$$

3. 以上即为最经典的交通分布预测最大熵模型。模型的优点在于其无需先验信息，但这同时也是其最大的缺陷——因缺少先验信息故只能假设各目的地的选择概率相同而未考虑阻抗。模型首次将最大熵原理应用于交通分配问题中，具有极大启发性；
4. 在此基础上进一步引入先验的观测信息作为约束，即可导出著名的双重约束重力模型 (**doubly constrained gravity model**)。引入如下先验约束

$$\min \sum_{i \in I} \sum_{j \in J} t_{ij}(\ln t_{ij} - 1), \quad \text{s.t.} \quad \begin{cases} \sum_j t_{ij} = O_i, & \forall i = 1, \dots, I \\ \sum_i t_{ij} = D_j, & \forall j = 1, \dots, J \\ \sum_i \sum_j t_{ij} c_{ij} = C \end{cases}$$

式中 O_i, D_j 分别表示观测的小区 i 的需求发生量和小区 j 的需求吸引量； C 表示总行程成本（如行程时间）； c_{ij} 为小区 i, j 间出行的行程成本。构建拉格朗日函数如下

$$\min \mathcal{L} = \sum_{i \in I} \sum_{j \in J} t_{ij}(\ln t_{ij} - 1) + \sum_i \lambda_{oi} \left(\sum_j t_{ij} - O_i \right) + \sum_j \lambda_{dj} \left(\sum_i t_{ij} - D_j \right) + \lambda_c \left(\sum_i \sum_j t_{ij} c_{ij} - C \right)$$

式中 $\lambda_{oi}, \lambda_{dj}, \lambda_c$ 均为拉格朗日乘子。基于一阶最优性原理代入 $\frac{\partial \mathcal{L}}{\partial t_{ij}} = 0$ 有

$$\frac{\partial \mathcal{L}}{\partial t_{ij}} = \ln t_{ij} + \lambda_{oi} + \lambda_{dj} + \lambda_c c_{ij} = 0 \implies t_{ij} = \exp \{-\lambda_{oi} - \lambda_{dj} - \lambda_c c_{ij}\}$$

再代入 $\frac{\partial \mathcal{L}}{\partial \lambda_{oi}} = \frac{\partial \mathcal{L}}{\partial \lambda_{dj}} = 0$ 消去拉格朗日乘子 $\lambda_{oi}, \lambda_{dj}$ 有

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \lambda_{oi}} = \sum_j t_{ij} - O_i = \exp \{-\lambda_{oi}\} \sum_j \exp \{-\lambda_{dj} - \lambda_c c_{ij}\} - O_i = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda_{dj}} = \sum_i t_{ij} - D_j = \exp \{-\lambda_{dj}\} \sum_i \exp \{-\lambda_{oi} - \lambda_c c_{ij}\} - D_j = 0 \end{cases} \implies \begin{cases} \exp \{-\lambda_{oi}\} = \frac{O_i}{\sum_j \exp \{-\lambda_{dj} - \lambda_c c_{ij}\}} \\ \exp \{-\lambda_{dj}\} = \frac{D_j}{\sum_i \exp \{-\lambda_{oi} - \lambda_c c_{ij}\}} \end{cases}$$

⁹Wilson, Alan. "A statistical theory of spatial distribution models." *Transportation Research* 1 (1967): 253-269. <https://www.sciencedirect.com/science/article/pii/0041164767900354>

最后将 $\exp\{-\lambda_{oi}\}$, $\exp\{-\lambda_{dj}\}$ 重新代入 t_{ij} 表达式有

$$t_{ij} = \exp\{-\lambda_{oi} - \lambda_{dj} - \lambda_c c_{ij}\} = \frac{O_i D_j \exp\{-\lambda_c c_{ij}\}}{(\sum_j \exp\{-\lambda_{dj} - \lambda_c c_{ij}\})(\sum_i \exp\{-\lambda_{oi} - \lambda_c c_{ij}\})}$$

为简化上式再令 $\alpha_i = \frac{\exp\{-\lambda_{oi}\}}{O_i}$, $\beta_j = \frac{\exp\{-\lambda_{dj}\}}{D_j}$, 则

$$\begin{cases} \alpha_i = \frac{1}{\sum_j \exp\{-\lambda_{dj} - \lambda_c c_{ij}\}} = \frac{1}{\sum_j \beta_j D_j \exp\{-\lambda_c c_{ij}\}} \\ \beta_j = \frac{1}{\sum_i \exp\{-\lambda_{oi} - \lambda_c c_{ij}\}} = \frac{1}{\sum_i \alpha_i O_i \exp\{-\lambda_c c_{ij}\}} \end{cases} \Rightarrow t_{ij} = \alpha_i \beta_j O_i D_j \exp\{-\lambda_c c_{ij}\}$$

以上即为双重约束重力模型的经典形式。需要说明的是在实际应用中一般不假设行程成本 c_{ij}, C 已知, 而是令 c_{ij} 为 t_{ij} 的函数。计算时首先令 $\alpha_i = 1$ 计算 β_j , 更新 t_{ij}, c_{ij} 后再由 β_j 更新 α_i 。如此迭代直至 t_{ij} 收敛。

第 11 章

动态交通分配与交通流模型

11.1 A demand model with departure time choice for within-day dynamic traffic assignment (EJOR, 2006)

ABSTRACT

构建了一个全天候的动态需求模型，除了经典的需求生成、分配和方式划分外，还包括考虑出发时间选择的实际需求模型。本文重点讨论出发时间建模，通过扩展离散选择模型至连续选择集实现。同时给出了基于隐式路径枚举 (implicit path enumeration) 的动态多模式供给和均衡模型，从而将全天候动态弹性需求随机多模式均衡问题转化为一个关于用户流量和公交频次的固定点问题。具体地，文章基于 Logit 路径选择模型和 MSA 算法（见 10.7.3 节），在真实维度网络上进行需求均衡，并给出了出发时间选择和实际需求模型的具体算法。

11.1.1 The choice and demand models

- 在建模交通需求时采用基于随机效用理论的行为建模方法。假设每一出行者均为理性决策者 (rational decision maker)，在做出行决策时遵循以下流程：
 - 考虑一个包含有限且完全互斥 (mutually exclusive) 选项的选择集 J ；
 - 每个选项 $j \in J$ 均有一个感知的效用，在不确定性地影响下感知效用记为随机变量 $U_j = V_j + \varepsilon_j$ ，其中 V_j, ε_j 分别为确定性系统效用和随机偏差；
 - 最终选择的出行选项应使得效用最大化。

在上述假设下选择选项 j 的概率被记为

$$P_j = \text{Prob} \left[\bigcap_{k \in J} \varepsilon_k \leq V_j - V_k + \varepsilon_j \right]$$

定义最大感知效用的期望为满意度 (satisfaction)，记为 $W = \mathbb{E}[\max_j\{V_j + \varepsilon_j\}]$ ；

- 假设可以将选择的过程分阶段序列化，每一阶段的选择集由上一阶段的选择确定。给定阶段下每一选项的效用为特定项和考虑后续阶段所有选项的满意度之和；
- 记多模式网络为 $G(N, A)$ ，其中 N, A 分别为点集和边集。一般节点记为 x ，路径的起点和终点分别记为 o, d ，并记起讫点集合为 $C \subseteq N$ 。以 $TL(a), HD(a)$ 分别表示连边 $a \in A$ 的起点和终点。对于每一模式 $m \in M$ ，相应的网络记为 $G_m(N_m, A_m)$ ，并记模式 m 下的 $o-d$ 对间的有效路径集 (efficient path set) 为 K_m^{od} 。记出发时间范围为 $[0, \Theta]$ ；
- 所提的需求模型为一个包含 5 层的巢式 Logit 模型 (Nested Logit)，具体包括需求生成、需求分布、方式划分、出发时间选择和路径选择。动态均衡框架如下：
 - 需求模型以 $N^o(\sigma') \geq 0, \sigma' \in [0, \Theta]$ 作为输入，定义为起点 o 处期望出发时间为 σ' 的出行者的数量；
 - 首先确定目的地选择和方式选择，对于相同的期望出发时间，模型的形式和参数相同；

- 针对期望出发时间为 σ' 的出行者，模型基于时间窗 $[\max\{\sigma' - ADV, 0\}, \min\{\sigma' + DEL, \Theta\}]$ 内的出发时间选择概率密度函数确定实际出发时间 τ' ，其中 ADV, DEL 为出行者所接受的最大提前和延后实际。对于 o, d, m, σ' 相同的出行者，出发时间选择时模型的形式和参数相同；
- 最后基于方式 m 进行路径选择。对于 o, d, τ' 相同的出行者，路径选择时模型的形式和参数相同。

需要说明的是出发时间选择定义在连续时间选择集上，但总体框架与离散选择一致。

Definitions and notations

1. 对于模式 m ，点 x 到 d 之间的有效路径集 K_m^{xd} 基于点的拓扑顺序 (node topological order) 确定。给定终点 d 和模式 m ，记 $TO_m^d(x)$ 表示图 G_m 中任意节点 x 的拓扑顺序， $TO_m^d(x)$ 关于 x 到 d 的距离单调非减¹，且与拥堵或时间无关；
2. 对于任意连边 a ，若其终点的拓扑顺序相对于起点更靠前（即 $TO_m^d(HD(a)) < TO_m^d(TL(a))$ ），则该连边为有效连边，而完全由有效连边组成的路径即为有效路径；
3. 将连接 x 的所有有效边分为两类：以 x 为起点的所有有效边集合称为有效前向星 (efficient forward star, FSE)；以 x 为终点的所有有效边集合称为有效后向星 (efficient backward star, BSE)²

$$FSE(x)_m^d = \{a \in A_m | TL(a) = x, TO_m^d(x) > TO_m^d(HD(a))\} \quad BSE(x)_m^d = \{a \in A_m | HD(a) = x, TO_m^d(x) < TO_m^d(TL(a))\}$$

4. 另外定义 τ' 时刻每一方式 m 连边的状态特征：

- $f_a^m(\tau')$: τ' 时刻以方式 m 进入连边 a 的流量；
- $t_a^m(\tau')$: τ' 时刻以方式 m 进入连边 a 的用户的离去时间（单调递增）；
- $t_a^{m-1}(\tau')$: τ' 时刻以方式 m 离开连边 a 的用户的进入时间（为 $t_a^m(\tau')$ 的反函数，同样单调递增）；
- $mc_a^m(\tau')$: τ' 时刻以方式 m 进入连边 a 的用户的金钱成本；
- $c_a^m(\tau')$: τ' 时刻以方式 m 进入连边 a 的用户的广义成本。记 η 为时间成本效用函数，则 $c_a^m(\tau')$ 可以简单记为：

$$c_a^m(\tau') = \eta \cdot (t_a^m(\tau') - \tau') + mc_a^m(\tau')$$

5. 进一步地定义路径的状态特征：

- A_k^{xdm} : 有效路径 $k \in K_m^{xd}$ 的所有连边集合；
- A_{ka}^{xdm} : 有效路径 $k \in K_m^{xd}$ 内从 x 到连边 $a \in A_k^{xdm}$ 的子路径的连边集合；
- $T_k^{xdm}(\tau')$: 出行者在 τ' 时刻从 x 出发选择有效路径 $k \in K_m^{xd}$ 至 d 的到达时间；
- $T_{ka}^{xdm}(\tau')$: 出行者在 τ' 时刻从 x 出发选择有效路径 $k \in K_m^{xd}$ 至进入连边 $a \in A_k^{xdm}$ 的时间，显然有

$$T_{ka}^{xdm}(\tau') = \tau' + \sum_{b \in A_{ka}^{xdm}} [t_b^m(T_{kb}^{xdm}(\tau')) - T_{kb}^{xdm}(\tau')]$$

- $C_k^{xdm}(\tau')$: 出行者在 τ' 时刻从 x 出发选择有效路径 $k \in K_m^{xd}$ 至至 d 的广义成本，显然有

$$C_k^{xdm}(\tau') = \sum_{a \in A_k^{xdm}} c_a^m(T_{ka}^{xdm}(\tau'))$$

Implicit path choice model

1. 隐式路径选择 (implicit path choice) 和显式路径选择 (explicit path choice) 是两种路径选择建模方式。传统的路径选择模型显式地以每一独立的路径为单位，即显式路径选择，而隐式路径选择则旨在将路径选择的过程分解为对路径的每一连边的选择。所采用的隐式路径选择模型基于两个重要概念：
 - 连边条件概率 (arc conditional probability) $p_a^{dm}(\tau')$: τ' 时刻位于节点 $TL(a)$ 处的出行者经连边 a 以方式 m 到达目的地 d 的概率；

¹若点 x 相比于 x' 距 d 的距离更小，则 x 的拓扑顺序更靠前，记为 $TO_m^d(x) < TO_m^d(x')$ 。

²前向星（后向星）和邻接矩阵、邻接表均为图建模方法，尽管相比之下后两种更常用。

- 节点满意度 (node satisfaction) $w_x^{dm}(\tau')$: τ' 时刻以方式 m 从 x 至 d 的所有有效路径的最大感知效用的期望。
2. 基于连边条件概率 $p_a^{dm}(\tau')$, 可以得到 τ' 时刻以方式 m 从 o 至 d 的出行者选择路径 $k \in K_m^{od}$ 的概率 $P_k^{odm}(\tau')$

$$P_k^{odm}(\tau') = \prod_{a \in A_k^{odm}} p_a^{dm}(T_{ka}^{odm}(\tau'))$$

3. 在显式路径建模的视角下, 假设每一选项的效用服从 Gumbel 分布³ (即 Logit 建模的基本假设), 则基于满意度的定义, 有

$$w_x^{dm}(\tau') = \theta_R \cdot \ln \left(\sum_{k \in K_m^{xd}} \exp \left\{ -\frac{C_k^{xdm}(\tau')}{\theta_R} \right\} \right)$$

上式中 $\theta_R \geq 0$ 为 Gumbel 分布尺度参数。而在隐式路径视角下, 则可将 $w_x^{dm}(\tau')$ 改写为如下递归形式

$$\begin{aligned} w_d^{dm}(\tau') &= 0 \\ w_x^{dm}(\tau') &= \theta_R \cdot \ln \left(\sum_{a \in FSE(x)_m^d} \exp \left\{ \frac{-c_a^m(\tau') + w_{HD(a)}^{md}(t_a^m(\tau'))}{\theta_R} \right\} \right) \end{aligned}$$

以上两种视角下导出的 $w_x^{dm}(\tau')$ 两种形式完全等价 (可由数学归纳法证明);

4. 并且基于 $w_x^{dm}(\tau')$ 的递归形式可由 $w_x^{dm}(\tau')$ 计算 $p_a^{dm}(\tau')$

$$\begin{aligned} p_a^{dm}(\tau') &= \frac{\exp \left\{ \theta_R^{-1} \cdot (-c_a^m(\tau') + w_{HD(a)}^{md}(t_a^m(\tau'))) \right\}}{\sum_{b \in FSE(TL(a))_m^d} \exp \left\{ \theta_R^{-1} \cdot (-c_b^m(\tau') + w_{HD(b)}^{md}(t_b^m(\tau'))) \right\}} \\ &= \frac{\exp \left\{ \theta_R^{-1} \cdot (-c_a^m(\tau') + w_{HD(a)}^{md}(t_a^m(\tau'))) \right\}}{\exp \left\{ \theta_R^{-1} \cdot w_{TL(a)}^{dm}(\tau') \right\}} = \exp \left\{ \theta_R^{-1} \cdot (-c_a^m(\tau') + w_{HD(a)}^{md}(t_a^m(\tau')) - w_{TL(a)}^{dm}(\tau')) \right\} \end{aligned}$$

Departure time choice model

1. 考虑理想出发时间 σ' , 基于扩展到连续选择集 $B(\sigma')$ 的 Logit 模型确定实际出发时间 τ' , 即以区间 $B(\sigma')$ 内的每一微量作为一个选项, 并假设残差独立同服从 Gumbel 分布。本研究中不考虑出发时间选项间的相关性, 但已有实验证据指出 Logit 模型可作为出发时间预测的合理模型;
2. 出于可读性的角度考虑, 以下定义变量将省去上下标 o, d, m , 需要说明的是以下变量所涉及的对象均是以方式 m 从 o 到 d 的出行者:
 - $w(\tau')$: 路径选择满意度, 即为上一节中的 $w_o^{dm}(\tau')$;
 - $V(\tau'|\sigma')$: 理想出发时间 σ' 确定时选择实际出发时间 τ' 的确定性效用。直接假设其正比于 τ' 相对于 σ' 的提前或滞后量

$$V(\tau'|\sigma') = -\max\{b_{ADV} \cdot (\sigma' - \tau'), b_{DEL} \cdot (\tau' - \sigma')\}$$

其中 $b_{ADV}, b_{DEL} \geq 0$ 分别为提前或延迟出发的边际成本;

- $p(\tau'|\sigma')$: 理想出发时间 σ' 确定时实际出发时间 τ' 的条件概率密度。基于扩展至连续选择集的 Logit 模型, 可以直接给出出行者的实际出发时间落入 $[\tau' - d\tau/2, \tau' + d\tau/2]$ 的概率

$$p(\tau'|\sigma')d\tau = \frac{\exp \left\{ \frac{V(\tau'|\sigma') + w(\tau')}{\theta_{DT}} \right\}}{\int_{\sigma'-ADV}^{\sigma'+DEL} \exp \left\{ \frac{V(t|\sigma') + w(t)}{\theta_{DT}} \right\} dt} dt$$

3. 注意到 $p(\tau'|\sigma')$ 的计算式中的分母可视为 σ' 的函数, 类比上一子节中离散选择的场景, 显然该分母表达式的物理意义与理想出发时间 σ' 的以方式 m 从 o 到 d 的出行者的满意度 $W(\sigma')$ 有关

$$W(\sigma') = \sigma_{DT} \cdot \ln \left[\int_{\sigma'-ADV}^{\sigma'+DEL} \exp \left\{ \frac{V(t|\sigma') + w(t)}{\theta_{DT}} \right\} dt \right]$$

³Gumbel 分布又称第一类极值分布 (type 1 extreme value distribution)。随机变量 X 服从 Gumbel 分布写作 $X \sim EV(\mu, \sigma)$, 概率密度函数 $f(x) = \sigma \cdot e^{-\sigma(x-\mu)} \cdot \exp\{-e^{-\sigma(x-\mu)}\}$, 其中 μ, σ 分别称为位置参数和尺度参数。Gumbel 分布具有如下性质: 若 $X_i \sim EV(\mu_i, \sigma)$, 则 $X = \max\{X_i\} \sim EV(\sigma \ln \sum_i \exp\{\mu_i/\sigma\}, \sigma)$ 。

Mode, destination and traveling choice models

1. 首先是方式选择建模。考虑理想出发时间为 σ' 并以 o, d 为起讫点的出行者，定义：

- $W_m^{od}(\sigma')$: 选择方式 m 的满意度，即为上一子节中的 $W(\sigma')$ ；
- $V_m^{od}(\sigma')$: 选择方式 m 的确定性效用，假设

$$V_m^{od}(\sigma') = \beta_M^T X_m^{od}(\sigma')$$

式中 $X_m^{od}(\sigma')$ 为出行方式 m 的属性向量，为时变函数； β_M 为参数向量；

- $P_m^{od}(\sigma')$: 选择方式 m 的概率，基于 Logit 模型，可以直接写为

$$P_m^{od}(\sigma') = \frac{\exp \left\{ \frac{V_m^{od}(\sigma') + W_m^{od}(\sigma')}{\theta_M} \right\}}{\sum_{m' \in M} \exp \left\{ \frac{V_{m'}^{od}(\sigma') + W_{m'}^{od}(\sigma')}{\theta_M} \right\}}$$

2. 而后是目的地建模。考虑理想出发时间为 σ' 并以 o 为起点的出行者，定义：

- $W^{od}(\sigma')$: 选择目的地 d 的满意度。同样地， $P_m^{od}(\sigma')$ 表达式中的分母的物理意义与理想出发时间 σ' 并以 o, d 为起讫点的所有出行者的满意度 $W^{od}(\sigma')$ 有关

$$W^{od}(\sigma') = \theta_D \cdot \ln \left[\sum_{d \in C} \exp \left\{ \frac{V_d^{od}(\sigma') + W_d^{od}(\sigma')}{\theta_D} \right\} \right]$$

- $V^{od}(\sigma')$: 选择目的地 d 的确定性效用，假设

$$V^{od}(\sigma') = \beta_D^T X^{od}(\sigma')$$

式中 $X^{od}(\sigma')$ 为目的地 d 的属性向量，为时变函数； β_D 为参数向量；

- $P_d^{od}(\sigma')$: 选择目的地 d 的概率，基于 Logit 模型，可以直接写为

$$P_d^{od}(\sigma') = \frac{\exp \left\{ \frac{V_d^{od}(\sigma') + W_d^{od}(\sigma')}{\theta_D} \right\}}{\sum_{d' \in C} \exp \left\{ \frac{V_{d'}^{od}(\sigma') + W_{d'}^{od}(\sigma')}{\theta_D} \right\}}$$

3. 最后是否出行建模。考虑理想出发时间为 σ' 并以 o 为起点的潜在出行者，定义：

- $W^o(\sigma')$: 选择出行的满意度。同样地， $P_d^o(\sigma')$ 表达式中的分母的物理意义与理想出发时间 σ' 并以 o 为起点的所有出行者的满意度 $W^o(\sigma')$ 有关

$$W^o(\sigma') = \theta_E \cdot \ln \left[\sum_{d \in C} \exp \left\{ \frac{V_d^o(\sigma') + W_d^o(\sigma')}{\theta_E} \right\} \right]$$

- $V^o(\sigma')$: 选择出行的确定性效用，假设

$$V^o(\sigma') = \beta_E^T X^o(\sigma')$$

式中 $X^o(\sigma')$ 为出发点 o 的属性向量，为时变函数； β_E 为参数向量；

- $P^o(\sigma')$: 选择出行的概率，设不出行的效用为 0，则基于 Logit 模型，可以直接写为

$$P^o(\sigma') = \frac{\exp \left\{ \frac{V^o(\sigma') + W^o(\sigma')}{\theta_E} \right\}}{1 + \exp \left\{ \frac{V^o(\sigma') + W^o(\sigma')}{\theta_E} \right\}}$$

4. 同样地， $P^o(\sigma')$ 表达式中的分母的物理意义与可能于理想出发时间 σ' 从 o 出发的所有潜在出行者的满意度 $S^o(\sigma')$ 有关

$$S^o(\sigma') = \theta_E \cdot \ln \left[1 + \exp \left\{ \frac{V^o(\sigma') + W^o(\sigma')}{\theta_E} \right\} \right]$$

Desired demand models & Actual demand model

1. 定义了 $P_m^{od}(\sigma'), P_d^o(\sigma'), P^o(\sigma')$ 后, 首先考虑理想出发时间 σ' 完成需求生成、需求分布和方式划分, 生成初步的网络需求分布;
2. 假设起始点 o 处以 σ' 为理想出发时间的潜在出行者的数量 $N^o(\sigma')$ 已知, 则显然有:
 - 选择于起始点 o 处以 σ' 为理想出发时间的出行者的数量 $q^o(\sigma') = N^o(\sigma') \cdot P^o(\sigma');$
 - 选择于起始点 o 处以 σ' 为理想出发时间并以 d 为终点的出行者的数量 $q^{od}(\sigma') = q^o(\sigma') \cdot P_d^o(\sigma');$
 - 选择以 σ' 为理想出发时间、以 o, d 为起讫点并选择方式 m 的出行者的数量 $q_m^{od}(\sigma') = q^{od}(\sigma') \cdot P_m^{od}(\sigma');$
3. 而后基于初步的网络需求分布构建实际出发时间概率分布, 从而考虑实际出发实际 τ' 完成出发时间选择和路径选择, 得到实际且精确到路径的网络需求分布;
4. 同样地为简化考虑省去上下标 o, d, m , 即将理想需求时间分布 $q_m^{od}(\sigma')$ 简化为 $q(\sigma')$, 而记实际需求时间分布为 $d(\tau')$ 。考虑实际出发时间选择条件概率 $p(\tau'|\sigma')$, 可以得到 $q(\sigma')$ 至 $d(\tau')$ 的转化式 (类似于卷积)

$$d(\tau') = \int_{\tau'-DEL}^{\tau'+ADV} q(\sigma)p(\tau'|\sigma)d\sigma = \int_{\tau'-DEL}^{\tau'+ADV} q(\sigma) \cdot \frac{\exp\left\{\frac{w(\tau') - \max\{b_{ADV}(\sigma-\tau'), b_{DEL}(\tau'-\sigma)\}}{\theta_{DT}}\right\}}{\int_{\sigma'-ADV}^{\sigma'+DEL} \exp\left\{\frac{w(t) - \max\{b_{ADV}(\sigma-t), b_{DEL}(t-\sigma)\}}{\theta_{DT}}\right\} dt} d\sigma$$

显然上式无法得到闭合形式的解析表达式。为此在后文中将介绍 $d(\tau')$ 的专门数值求解方法, 并为了简化计算假设 $q(\sigma)$ 为分段常函数, 而 $w(\tau)$ 为分段线性函数。

Network flow propagation model

1. 基于网络的实际需求分布 $d(\tau')$, 即可将动态需求载入网络各连边中 (dynamic network loading, DNL)。不同于静态交通分配假设需求瞬间载入整条路径, 动态交通分配要求需求载入满足时间一致性, 即先产生的需求应该先载入, 且需考虑路径需求沿路径各路段的行程时间;
2. 若路径选择时采用显式路径建模, 则需求动态载入时还需专门给出额外的约束保证动态需求载入的时间一致性, 而隐式路径选择以连边选择为基本单位, 则不存在这一问题;
3. 直接给出 $f_a^m(\tau')$ 的表达式如下

$$f_a^m(\tau') = \sum_{d \in C} f_a^{md}(\tau') = \sum_{d \in C} \left\{ p_a^{md}(\tau') \cdot \left[d_m^{TL(a)d}(\tau') + \sum_{b \in BS E(TL(a))^d} \left(f_b^{md}(t_b^{m-1}(\tau')) \cdot \frac{\partial t_b^{m-1}(\tau)}{\partial \tau} \Big|_{\tau=\tau'} \right) \right] \right\}$$

$f_a^m(\tau')$ 即为实际需求分布 $d(\tau')$ 下的路段驶入流量时变函数。进一步解释上式, 核心在于 $f_a^{md}(\tau')$ 的计算。可以看到 $f_a^{md}(\tau')$ (即 τ' 时刻以方式 m 驶入连边 a 的流量中以 d 为目的地的流量) 由两部分组成:

- 一部分为从 $TL(a)$ 出发至 d 的流量, 即 $p_a^{md}(\tau') \cdot d_m^{TL(a)d}(\tau');$
- 另一部分为来自上游路段驶向 d 的流量, 写为加权和的形式:

$$p_a^{md}(\tau') \cdot \sum_b f_b^{md}(t_b^{m-1}(\tau')) \cdot \frac{\partial t_b^{m-1}(\tau)}{\partial \tau} \Big|_{\tau=\tau'}$$

$t_b^{m-1}(\tau')$ 为 τ' 时刻自上游路段 b 到达节点 $TL(a)$ 的需求进入上游路段 b 的时间, 因而理论上 $f_b^{md}(t_b^{m-1}(\tau'))$ 即为 τ' 时刻自上游路段 b 到达节点 $TL(a)$ 的需求, 然而该表达式只有当路段 b 车辆匀速时才成立, 当路段速度变化时, 则需导数项 $\partial t_b^{m-1}(\tau')/\partial \tau$ 进行加权。进一步解释加权的原理。考虑 $[\tau', \tau' + \Delta]$ 时段自连边 b 到达节点 $TL(a)$ 的交通量, 其进入连边 b 的时间范围为 $[t_b^{m-1}(\tau'), t_b^{m-1}(\tau' + \Delta)]$ 。当 $\Delta \rightarrow 0$ 时, 可近似假设 $f_b^{md}(t_b^{m-1}(\tau'))$ 在时段 $[t_b^{m-1}(\tau'), t_b^{m-1}(\tau' + \Delta)]$ 内为常数, 因此 $[\tau', \tau' + \Delta]$ 时段自连边 b 到达节点 $TL(a)$ 的交通量即为 $f_b^{md}(t_b^{m-1}(\tau')) \cdot [t_b^{m-1}(\tau' + \Delta) - t_b^{m-1}(\tau')]$ 。同样地假设 $[\tau', \tau' + \Delta]$ 时段自连边 b 到达节点 $TL(a)$ 的流率为常数, 即为 $f_b^{md}(t_b^{m-1}(\tau')) \cdot \frac{t_b^{m-1}(\tau' + \Delta) - t_b^{m-1}(\tau')}{\Delta} = f_b^{md}(t_b^{m-1}(\tau')) \cdot \partial t_b^{m-1}(\tau')/\partial \tau$ 。

11.1.2 Supply and equilibrium models

类比静态分配模型, 全天候动态交通分配模型对应为动态均衡问题, 可写为一个关于连边流量和公交频次的不动点问题——即连边流量和公交频次共同影响出行选择, 而出行选择的结果又反过来影响连边流量和公

交频次。首先介绍均衡流 (equivalent flow) $v_a(\tau)$ 的概念。对于公交网络, 均衡流即为用户流; 而对于地面路网, 均衡流则是各模式车辆流的线性组合。具体介绍连边状态模型, 旨在建模连边在给定流量下的行程时间 (考虑拥堵延误)。

对于地面路网: 将地面交通流视为一维部分可压缩流体 (monodimensional partially compressible fluid), 基于简化运动波 (kinematic wave) 理论和三角形基本图建模地面连边状态。记连边长度 L_a 、限速 V_a , 考虑信号灯折减后的通行能力为 Q_a 。将总时间区间离散为若干时段, 每一子区间记为 $(\tau^{i-1}, \tau^i]$ 。假设每一子区间内的连边输入需求为常量 $f_a^i = f_a(\tau^i)$ 。考虑地面连边交通流两种状态下的延误并给出简化计算方法:

- 运动状态: 交通流正常流动通过末端节点, 这一状态下产生的延误是因为需求过大产生拥堵造成的。定义 t_{Ra}^i 为 τ^i 时刻进入连边 a 的车流结束运动状态的时间, 则在区间 $(t_{Ra}^{i-1}, t_{Ra}^i]$ 内显然有两种可能 (如右图所示): 1) 不存在拥堵; 2) 拥堵向上游传播。对于第一种情况显然有

$$t_{Ra}^i = \tau^i + \frac{L_a}{V_a}$$

而对于第二种情况, 考虑交通流守恒定律和先进先出 (FIFO) 规则, 有

$$Q_a \cdot (t_{Ra}^i - t_{Ra}^{i-1}) = f_a^i \cdot (\tau^i - \tau^{i-1}) \implies t_{Ra}^i = t_{Ra}^{i-1} + \frac{f_a^i}{Q_a} (\tau^i - \tau^{i-1})$$

综上即可得到考虑运动状态下过饱和延误的运动状态行程时间

$$t_{Ra}^i = \max \left\{ t_{Ra}^{i-1} + \frac{f_a^i}{Q_a} (\tau^i - \tau^{i-1}), \tau^i + \frac{L_a}{V_a} \right\}$$

- 排队状态: 因为信号灯控制下节点关闭, 交通流在末端节点上游排队等待, 这一状态下产生的延误是因为信号控制造成的。为简化计算, 以信号灯一个周期内的平均延误作为排队延误。假设末端节点信号灯周期为 $CT_{HD(a)} \geq 0$, 有效红灯时长占比为 $0 \leq r_a < 1$, 整个周期内的平均通行能力为 Q_a , 则有效绿灯相位内的通行能力为 $Q_a/(1-r_a)$ 。注意到 τ^{i-1} 时刻进入连边 a 的车流的排队延误必然发生于区间 $(t_{Ra}^{i-1}, t_{Ra}^i]$ 内。记 e_{Ra}^i 为区间 $(t_{Ra}^{i-1}, t_{Ra}^i]$ 内结束运动状态进入排队状态的平均流率, 由守恒定律和先进先出 (FIFO) 规则, 有

$$e_{Ra}^i \cdot (t_{Ra}^i - t_{Ra}^{i-1}) = f_a^i \cdot (\tau^i - \tau^{i-1}) \implies e_{Ra}^i = f_a^i \cdot \frac{\tau^i - \tau^{i-1}}{t_{Ra}^i - t_{Ra}^{i-1}}$$

而对于输入需求为 e_{Ra}^i 、绿灯通行能力为 $Q_a/(1-r_a)$ 的交叉口, 容易基于流入-流出曲线计算一个周期 $CT_{HD(a)} \geq 0$ 内的平均停车延误

$$\delta_a^i = \frac{0.5 \cdot r_a^2 \cdot CT_{HD(a)}}{1 - (1 - r_a) \frac{e_{Ra}^i}{Q_a}} = \frac{0.5 \cdot r_a^2 \cdot CT_{HD(a)}}{1 - (1 - r_a) \frac{f_a^i \cdot (\tau^i - \tau^{i-1})}{Q_a \cdot (t_{Ra}^i - t_{Ra}^{i-1})}}$$

理论上上式得到的 δ_a^i 即为 τ^i 时刻进入 a 的交通流的排队延误。尽管上式在区间 $(t_{Ra}^{i-1}, t_{Ra}^i]$ 内满足守恒定律和先进先出 (FIFO) 规则, 当 $e_{Ra}^{i-1} \neq e_{Ra}^i$ 时在区间边界 t_{Ra}^i 处不满足守恒规则, 且当 $e_{Ra}^{i-1} > e_{Ra}^i$ 时在区间边界 t_{Ra}^i 处也不满足先进先出 (FIFO) 规则。为此, 假设排队延误为分段线性函数, 则最终的交通流的排队延误 $\tilde{\delta}_a^i$ 为

$$\tilde{\delta}_a^i = \frac{1}{2} (\delta_a^i + \delta_a^{i+1})$$

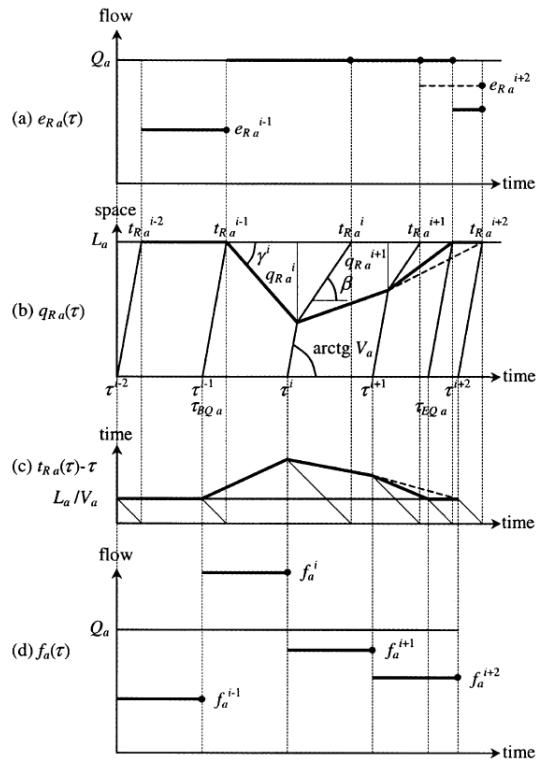


Fig. 5. Temporal profiles for the running phase.

- 行驶状态 + 排队状态：综合考虑行驶状态和排队状态下的自由流行程时间、拥堵延误和排队延误后即可得到连边的总行程时间 $t_a^i = t_a(\tau^i) = t_{Ra}^i + \tilde{\delta}_a^i$ 。给出伪代码如下

Algorithm 11.1 给定地路面网连边输入需求 f_a^i 下的考虑延误的连边行程时间 t_a^i 计算

输入: $f_a^i, L_a, V_a, r_a, Q_a, CT_{HD(a)}, \tau^i, \forall a \in A, i = 0, \dots, I$, 初始化 $i = 0$, 此时有 $f_a^0 = 0, \tau^0 = 0$

```

1: for  $\forall a \in A$  do
2:    $t_a^0 = L_a/V_a$ 
3:   for  $i = 1, \dots, I$  do
4:      $t_a^i = \max \left\{ t_a^{i-1} + \frac{f_a^i}{Q_a} (\tau^i - \tau^{i-1}), \tau^i + \frac{L_a}{V_a} \right\}$ 
5:   end for
6:    $t_a^0 = t_a^0 + 0.25r_a^2 \cdot CT_{HD(a)} \left( 1 + \frac{1}{1 - (1 - r_a) \frac{f_a^1 \cdot \tau^1}{Q_a \cdot (t_a^1 - t_a^0)}} \right)$  (即  $f_a^0 = 0, \tau^0 = 0$ )
7:   for  $i = 1, \dots, I - 1$  do
8:      $t_a^i = t_a^i + 0.25r_a^2 \cdot CT_{HD(a)} \left( \frac{1}{1 - (1 - r_a) \frac{f_a^i \cdot (\tau^i - \tau^{i-1})}{Q_a \cdot (t_a^i - t_a^{i-1})}} + \frac{1}{1 - (1 - r_a) \frac{f_a^{i+1} \cdot (\tau^{i+1} - \tau^i)}{Q_a \cdot (t_a^{i+1} - t_a^i)}} \right)$ 
9:   end for
10:   $t_a^I = t_a^I + 0.25r_a^2 \cdot CT_{HD(a)} \left( \frac{1}{1 - (1 - r_a) \frac{f_a^I \cdot (\tau^I - \tau^{I-1})}{Q_a \cdot (t_a^I - t_a^{I-1})}} + 1 \right)$  (即  $f_a^{I+1} = 0$ )
11: end for

```

对于公交路网：考虑乘客的等待时间和乘车时间确定总行程时间，认为其为网络均衡流和发车频率的函数。具体地建模公交站点如下图所示。记公交线为 l ，路线经过的每一地面连边 a 为路线连边 (line arc)， $\phi_a(\tau)$ 为 τ 时刻连边 a 的公交到达频率。来自上游连边的公交车到达站点连边 (stop arc) 停泊，完成上下客之后进入下游连边。具体地，公交车上的乘客在站点连边的开始节点完成下客，到站乘客通过下车连边 (alighting arc) 离开公交车进入地面路网，而地面路网的乘客则依次通过若干连边由上车连边 (boarding arc) 进入公交车。进一步详细介绍公交站点模型中每一元素的作用和相应时间计算方法。

- 停车站到达连边 (stop access arc)、上车连边 (boarding arc)、站点连边 (stop arc)：均不计行程时间和成本。其中停车站到达连边用于区分从地面路网同一节点前往不同公交站的上车人流。

- 步行连边 (walking arc)、下车连边 (alighting arc)：不考虑拥堵，行程时间为路段长度与步行速度之比。

- 等待连边 (waiting arc)：建模每一时刻上车乘客的

平均等待时间。记 $t_{WA(a)}(\tau)$ 为 τ 时刻进入等待连边的乘客的上车时间，则其反函数 $t_{WA(a)}^{-1}(\tau)$ 即为 τ 时刻上车的乘客进入等待连边的时间， $\tau - t_{WA(a)}^{-1}(\tau)$ 即为平均等待时间。显然该等待时间与公交到达频率 $\phi_a(\tau)$ 成反比

$$t_{WA(a)}^{-1}(\tau) = \tau - \frac{\alpha_l}{\phi_a(\tau)}$$

式中 α_l 为考虑公交车到达不均匀性的调节系数， $\alpha_l = 0.5$ 对应公交到达车头时距服从均匀分布， $\alpha_l = 1$ 对应公交到达车头时距服从负二项分布。通过解上述反函数 $t_{WA(a)}^{-1}(\tau)$ 即可计算 $t_{WA(a)}(\tau)$ 。因为受公交排班和地面拥堵的影响 $\phi_a(\tau)$ 具有时变性，难以得到 $t_{WA(a)}^{-1}(\tau)$ 的解析表达式。故假设 $t_{WA(a)}^{-1}(\tau)$ 为分段线性函数，基于第23.12节所提方法给出数值计算 $t_{WA(a)}(\tau)$ 的伪代码

Algorithm 11.2 公交站点等待连边 $t_{WA(a)}(\tau)$ 计算

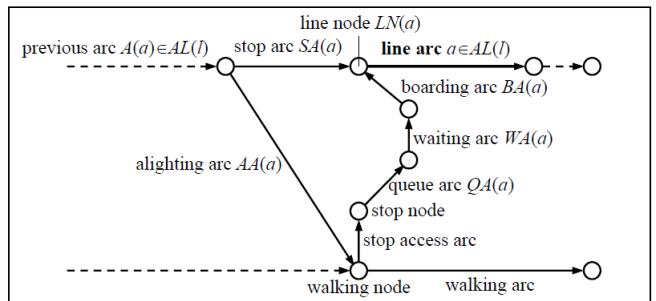


Figure 2 – Transit stop model

输入: $\alpha_l, \phi_a(\tau^i), \forall l \in L, a \in A_l, i = 0, \dots, I$, 初始化 $i = 0$, 此时有 $\tau^0 = 0$

```

1: for  $\forall l \in L$  do
2:   for  $\forall a \in A_l$  do
3:     for  $i = I, \dots, 0$  do
4:        $t_{WA(a)}^{-1}(\tau^i) = \min \left\{ \tau^i - \frac{\alpha_l}{\phi_a(\tau^i)}, t_{WA(a)}^{-1}(\tau^{i+1}) \right\}$ 
5:     end for
6:      $j = 0$ 
7:     for  $i = 0, \dots, I$  do
8:       while  $t_{WA(a)}^{-1}(\tau^j) < \tau^i$  do
9:          $j = j + 1$ 
10:         $t_{WA(a)}(\tau^i) = \tau^{j-1} + (\tau^i - t_{WA(a)}^{-1}(\tau^{j-1})) \cdot \frac{\tau^j - \tau^{j-1}}{t_{WA(a)}^{-1}(\tau^j) - t_{WA(a)}^{-1}(\tau^{j-1})}$  (按第23.12节所提方法于每个点  $\tau = \tau^{j-1}$  处进行泰勒展开, 因为假设  $t_{WA(a)}^{-1}(\tau)$  为分段线性函数, 故数值计算时只进行一阶展开, 不考虑高阶差分。)
11:      end while
12:    end for
13:  end for
14: end for

```

- **排队连边 (queue arc):** 当上车需求过大时, 存在一部分乘客无法顺利上车, 此时将在排队连边上等待下一班车。同样地计算 $t_{QA(a)}(\tau)$, 即 τ 时刻进入排队连边 $QA(a)$ 的乘客离开排队连边的时间。显然, 若排队连边的累计流入和流出曲线 $F_{QA(a)}(\tau), E_{QA(a)}(\tau)$ 已知, 则 $t_{QA(a)}(\tau)$ 可按下式计算

$$t_{QA(a)}(\tau) = \max \left\{ 0, \arg \min_{\sigma} \{E_{QA(a)}(\sigma) = F_{QA(a)}(\tau)\} \right\} = \max \left\{ 0, \arg \min_{\sigma} \{E_{QA(a)}(\sigma) - F_{QA(a)}(\tau) = 0\} \right\}$$

因为公交需求 $v_{QA(a)}(\tau)$ 已知, 则 $F_{QA(a)}(\tau)$ 已知, 故只需计算 $E_{QA(a)}(\tau)$ 。显然 $E_{QA(a)}(\tau)$ 与公交车的剩余通行能力 (available capacity) 有关。记公交连边 a 和排队连边 $QA(a)$ 末端节点的剩余通行能力分别为 $AK_a(\tau), AK_{QA(a)}(\tau)$, 则首先有

$$AK_a(\tau) = VK_l \cdot \phi_a(\tau) - v_{SA(a)}(\tau)$$

式中 VK_l 为线路 l 的公交车的容量, $v_{SA(a)}(\tau)$ 为公交车到达站点连边时的公交均衡流 (已去除下车乘客)。注意到当排队连边存在乘客排队时, 必然意味着等待连边上的乘客以最大通行能力上车, 因此公交连边 a 的剩余通行能力 $AK_a(\tau)$ 同时也是等待连边 $WA(a)$ 末端节点的流率, 而基于上小节给出的网络传播模型, 可以得到排队连边 $QA(a)$ 末端节点的流率 $AK_{QA(a)}(\tau)$

$$AK_{QA(a)}(\tau') \cdot \frac{\partial t_{WA(a)}^{-1}(\tau')}{\partial \tau} = AK_a(\tau') \implies AK_{QA(a)}(\tau') = \frac{AK_a(\tau')}{\partial t_{WA(a)}^{-1}(\tau')/\partial \tau}$$

基于 $AK_{QA(a)}(\tau)$ 计算 $E_{QA(a)}(\tau)$, 直接给出下式

$$\begin{aligned} E_{QA(a)}(\tau) &= \min_{\sigma \leq \tau} \left\{ F_{QA(a)}(\sigma) + \int_0^\tau AK_{QA(a)}(\tau') d\tau' - \int_0^\sigma AK_{QA(a)}(\tau') d\tau' \right\} \\ &= \min_{\sigma \leq \tau} \left\{ \int_0^\sigma v_{QA(a)}(\tau') d\tau' + \int_\sigma^\tau AK_{QA(a)}(\tau') d\tau' \right\} = F_{QA(a)}(\tau) + \min_{\sigma \leq \tau} \left\{ \int_\sigma^\tau AK_{QA(a)}(\tau') - v_{QA(a)}(\tau') d\tau' \right\} \end{aligned}$$

解释上式。考虑以下两种情况:

- 当 τ 时刻排队连边上不存在排队时, 则对 $\forall \sigma \leq \tau$ 必然有 $\int_\sigma^\tau AK_{QA(a)}(\tau') - v_{QA(a)}(\tau') d\tau' \geq 0$ (表示从任意时刻开始至 τ 时刻的所有输入需求都能完成输出), 此时有 $E_{QA(a)}(\tau) = F_{QA(a)}(\tau)$, 符合物理规律;
- 当 τ 时刻排队连边上存在排队时, 假设该排队于 $\tau^* < \tau$ 时刻开始生成, 则显然可以计算得 $E_{QA(a)}(\tau) = F_{QA(a)}(\tau) + \int_{\tau^*}^\tau AK_{QA(a)}(\tau') - v_{QA(a)}(\tau') d\tau' = E_{QA(a)}(\tau^*) + \int_{\tau^*}^\tau AK_{QA(a)}(\tau') d\tau'$, 符合物理规律。

至此通过解 $y(\sigma) = E_{QA(a)}(\sigma) - F_{QA(a)}(\sigma)$ 的反函数 y^{-1} 即可计算 $t_{QA(a)}(\tau) = y^{-1}(0)$ 。同样地假设 $y(\sigma)$ 为分段线性函数, 基于第23.12节所提方法给出数值计算 $t_{QA(a)}(\tau)$ 的伪代码

Algorithm 11.3 公交站点排队连边 $t_{QA(a)}(\tau)$ 计算

输入: $VK_l, \phi_a(\tau^i), v_{SA(a)}(\tau^i), t_{WA(a)}^{-1}(\tau^i), \forall l \in L, a \in A_l, i = 0, \dots, I$, 初始化 $i = 0$, 此时有 $\tau^0 = 0$

```

1: for  $\forall l \in L$  do
2:   for  $\forall a \in A_l$  do
3:     for  $i = 0, \dots, I$  do
4:        $AK_a(\tau^i) = VK_l \cdot \phi_a(\tau^i) - v_{SA(a)}(\tau^i)$ 
5:        $AK_{QA(a)}(\tau^i) = AK_a(\tau^i) \cdot \frac{\tau^i - \tau^{i-1}}{t_{WA(a)}^{-1}(\tau^i) - t_{WA(a)}^{-1}(\tau^{i-1})}$ 
6:     end for
7:      $F_{QA(a)}(\tau^0) = E_{QA(a)}(\tau^0) = 0$ 
8:     for  $i = 1, \dots, I$  do
9:        $F_{QA(a)}(\tau^i) = F_{QA(a)}(\tau^{i-1}) + v_{SA(a)}(\tau^i) \cdot (\tau^i - \tau^{i-1})$ 
10:       $E_{QA(a)}(\tau^i) = \min \{F_{QA(a)}(\tau^i), E_{QA(a)}(\tau^{i-1}) + AK_{QA(a)}(\tau^i) \cdot (\tau^i - \tau^{i-1})\}$ 
11:    end for
12:     $t_{QA(a)}(\tau^0) = 0, j = 0$ 
13:    for  $i = 1, \dots, I$  do
14:      while  $E_{QA(a)}(\tau^j) < F_{QA(a)}(\tau^i)$  do
15:         $j = j + 1$ 
16:         $t_{QA(a)}(\tau^i) = \max \left\{ \tau^i, \tau^{j-1} + (F_{QA(a)}(\tau^i) - E_{QA(a)}(\tau^{j-1})) \frac{\tau^j - \tau^{j-1}}{E_{QA(a)}(\tau^j) - E_{QA(a)}(\tau^{j-1})} \right\}$ 
17:      end while
18:    end for
19:  end for
20: end for

```

• 路线连边 (line arc): 组成公交车线路的地面连边。一条公交车连边连接相邻的两个公交站，包含若干地面连边。公交车于路线连边的总行程时间由停泊时间 (dwelling time) 和行驶时间决定。行驶时间由地面路网的拥挤程度决定，上文已介绍相应计算方法。停泊时间考虑公交站点处上下客流量和车上乘客数的影响，具体定义为上下客所需总时间与车门开闭所需时间之和。记车门开闭时间均为 δ_l ，下客和上客的流率分别为 $v_{AA(a)}(\tau)$, $v_{BA(a)}(\tau)$ ，车门打开时最大上车和下车流量分别为 AQ_l, BQ_l 。则当公交车只有一个车门，或所有车门均可上下客时，其停泊时间为

$$\tau + \frac{v_{AA(a)}(\tau) \cdot \frac{1}{\phi_a(\tau)}}{AQ_l \left(1 - \alpha \cdot \left(\frac{v_a}{VK_l \cdot \phi_a(\tau)} \right)^\beta \right)} + \frac{v_{BA(a)}(\tau) \cdot \frac{1}{\phi_a(\tau)}}{BQ_l \left(1 - \alpha \cdot \left(\frac{v_a}{VK_l \cdot \phi_a(\tau)} \right)^\beta \right)} + 2\delta_l, \quad \alpha > 0$$

当公交车的车门区分上下客时，其停泊时间为

$$\tau + \max \left\{ \frac{v_{AA(a)}(\tau) \cdot \frac{1}{\phi_a(\tau)}}{AQ_l \left(1 - \alpha \cdot \left(\frac{v_a}{VK_l \cdot \phi_a(\tau)} \right)^\beta \right)}, \frac{v_{BA(a)}(\tau) \cdot \frac{1}{\phi_a(\tau)}}{BQ_l \left(1 - \alpha \cdot \left(\frac{v_a}{VK_l \cdot \phi_a(\tau)} \right)^\beta \right)} \right\} + 2\delta_l, \quad \alpha > 0$$

式中分子部分 $v_{AA(a)}(\tau) \cdot \frac{1}{\phi_a(\tau)}$, $v_{BA(a)}(\tau) \cdot \frac{1}{\phi_a(\tau)}$ 分别表示一班公交到达时需要上下客的总人数，而分子部分则考虑公交车内的乘客密度对其上下客速度进行折减。

• 公交到达频率：受地面拥堵的影响，公交到达每一个站点的频率往往与其从初始站点发车的频率不一致。因此类比拥堵传播模型，给出地面公交到达频率的传播模型。记 $T_l^n(\tau)$ 为 τ 时刻从线路 l 的起点站出发的公交到达线路的第 n 个站点的时间，则显然有

$$T_l^1(\tau) = \tau, \quad T_l^n(\tau) = t_a(T_l^{n-1}(\tau)), \quad n \geq 2$$

记 τ 时刻线路 l 的第 n 个站点的公交到达频率为 $\phi_l^n(\tau)$, 注意到 $\phi_l^n(\tau)$ 同时可表示公交流率⁴。又记 τ 时刻线路 l 的公交发车频率为 $\psi_l(\tau)$, 则基于流量传播公式, 有

$$\phi_l^n(T_l^n(\tau')) = \frac{\psi_l(\tau')}{\partial T_l^n(\tau')/\partial \tau}$$

11.1.3 Algorithm

1. 为简化运算对时间进行离散化, 并假设相关时序变量 $x(\tau^0)$ 为分段常函数 (流率、需求) 或分段线性函数 (连边行程时间、连边费用、满意度、选择概率), 有

$$x(\tau^0) = \tau^0, \quad x(\tau) = x^i, \quad \tau \in (\tau^{i-1}, \tau^i], \quad i = 1, \dots, I$$

$$x(\tau^0) = \tau^0, \quad x(\tau) = x^{i-1} + (\tau - \tau^{i-1}) \cdot \frac{x^i - x^{i-1}}{\tau^i - \tau^{i-1}}, \quad \tau \in (\tau^{i-1}, \tau^i], \quad i = 1, \dots, I$$

2. 将考虑出发时间选择的动态需求多模式动态交通分配问题建模为不动点问题后, 基于 MSA 算法求解 (见第 10.7.3 节), 抽象的总体流程如下所示:

- 初始化: $k = 0, f^{k+1} = \{0|f_{inz}\}$;
- 开始一轮迭代: $k = k + 1$;
- 计算均衡流: $v^k = v(f^k, \phi^k)$;
- 计算连边状态 (行程时间和成本): $t^k = t(v^k, \phi^k), c^k = c(v^k, \phi^k)$;
- 路径选择: $w^k = w(c^k, t^k), p^k = p(w^k, c^k, t^k)$;
- 出发时间选择: $P_{DT}^k = P(V_{DT}, w^k), W_{DT}^k = W(V_{DT}, w^k)$;
- 方式选择: $P_M^k = P(V_M, W_{DT}^k), W_M^k = W(V_M, W_{DT}^k)$;
- 目的地选择: $P_D^k = P(V_D, W_M^k), W_D^k = W(V_D, W_M^k)$;
- 是否出行选择: $P_E^k = P(V_E, W_D^k), S_E^k = S(V_E, W_D^k)$;
- 理想需求分布计算: $q^k = q(N, P_G^k, P_D^k, P_M^k)$;
- 实际需求分布计算: $d^k = d(q^k, W_{DT}^k, w^k)$;
- 地面网络流传播: $f_{NLM}^k = \omega(p^k, t^k, d^k)$;
- 公交到达频率传播: $\phi_{NLM}^k = \phi(t^k)$;
- 地面网络流更新 (基于 MSA 算法): $f^{k+1} = f^k + \frac{1}{k} \cdot (f_{NLM}^k - f^k)$;
- 公交到达频率更新 (基于 MSA 算法): $\phi^{k+1} = \phi^k + \frac{1}{k} \cdot (\phi_{NLM}^k - \phi^k)$;
- 若 $\max_{a \in A, i \in I} \|y_a^{ik} - f_a^{ik}\| > \varepsilon$, $k < k_{\max}$, 继续进行新一轮迭代。

在上述计算过程中, 因为出发时间选择和实际需求分布计算需要进行积分运算, 复杂度较大, 故于后文中重点介绍相关内容;

3. 在前文中, 将出发时间选择建模为定义在无穷解集上的 Logit 模型, 定义 $p(\tau'|\sigma')$ 为理想出发时间 σ' 确定时实际出发时间 τ' 的条件概率密度。而通过将时间离散化为若干子区间, 则可将 $p(\tau'|\sigma')$ 简化为 $P_i^j = P((\tau^{i-1}, \tau^i]|\sigma^j)$, 表示理想出发时间 σ^j 确定时实际出发时间 $\tau' \in (\tau^{i-1}, \tau^i]$ 的概率。显然 P_i^j 为 $p(\tau'|\sigma')$ 在区间 $(\tau^{i-1}, \tau^i]$ 内的积分

$$P_i^j = \int_{\tau^{i-1}}^{\tau^i} \frac{\exp \left\{ \frac{w(\tau) - \max\{b_{ADV} \cdot (\sigma^j - \tau), b_{DEL} \cdot (\tau - \sigma^j)\}}{\theta_{DT}} \right\}}{\int_{\sigma^j-ADV}^{\sigma^j+DEL} \exp \left\{ \frac{w(t) - \max\{b_{ADV} \cdot (\sigma^j - t), b_{DEL} \cdot (t - \sigma^j)\}}{\theta_{DT}} \right\} dt} d\tau$$

注意到在时间离散化后上式中 $w(\tau)$ 简化为分段线性函数, 因此上式积分可以解析计算。因为相似性, 给出被积函数的分母定积分的解析计算过程作为示例

$$\int_{\sigma^j-ADV}^{\sigma^j+DEL} \exp \left\{ \frac{w(t) - \max\{b_{ADV} \cdot (\sigma^j - t), b_{DEL} \cdot (t - \sigma^j)\}}{\theta_{DT}} \right\} dt$$

⁴公交到达频率为 $\phi_l^n(\tau)$ 意味着在长达 $1/\phi_l^n(\tau)$ 时段内的公交到达频数为 1, 因此该时段内的公交流率即为 $\phi_l^n(\tau)$ 。

$$= \int_{\sigma^j - ADV}^{\sigma^j} \exp \left\{ \frac{w(t) - b_{ADV} \cdot (\sigma^j - t)}{\theta_{DT}} \right\} dt + \int_{\sigma^j}^{\sigma^j + DEL} \exp \left\{ \frac{w(t) - b_{DEL} \cdot (t - \sigma^j)}{\theta_{DT}} \right\} dt$$

以第一项定积分为例。代入 $w(t) = w^{i-1} + (t - t^{i-1}) \cdot \frac{w^i - w^{i-1}}{t^i - t^{i-1}}$, 有

$$\begin{aligned} & \int_{\sigma^j - ADV}^{\sigma^j} \exp \left\{ \frac{w(t) - b_{ADV} \cdot (\sigma^j - t)}{\theta_{DT}} \right\} dt \\ &= \int_{\sigma^j - ADV}^{\sigma^j} \exp \left\{ \frac{w^{i-1} + (t - t^{i-1}) \cdot \frac{w^i - w^{i-1}}{t^i - t^{i-1}} - b_{ADV} \cdot (\sigma^j - t)}{\theta_{DT}} \right\} dt \\ &= \int_{\sigma^j - ADV}^{\sigma^j} \exp \left\{ \frac{w^{i-1} - t^{i-1} \cdot \frac{w^i - w^{i-1}}{t^i - t^{i-1}} - b_{ADV} \cdot \sigma^j}{\theta_{DT}} \right\} \cdot \exp \left\{ \frac{t \cdot \frac{w^i - w^{i-1}}{t^i - t^{i-1}} + b_{ADV} \cdot t}{\theta_{DT}} \right\} dt \\ &= \int_{\sigma^j - ADV}^{\sigma^j} \exp \{ \alpha_A + \beta_A t \} dt = \frac{\exp\{\alpha_A\}}{\beta_A} (\exp\{\beta_A \sigma^j\} - \exp\{\beta_A (\sigma^j - ADV)\}) \end{aligned}$$

因此有

$$\begin{aligned} & \int_{\sigma^j - ADV}^{\sigma^j + DEL} \exp \left\{ \frac{w(t) - \max\{b_{ADV} \cdot (\sigma^j - t), b_{DEL} \cdot (t - \sigma^j)\}}{\theta_{DT}} \right\} dt \\ &= \frac{\exp\{\alpha_A\}}{\beta_A} (\exp\{\beta_A \sigma^j\} - \exp\{\beta_A (\sigma^j - ADV)\}) + \frac{\exp\{\alpha_D\}}{\beta_D} (\exp\{\beta_D (\sigma^j + DEL)\} - \exp\{\beta_D \sigma^j\}) \end{aligned}$$

上式中 $\alpha_A, \beta_A, \alpha_D, \beta_D$ 均为常数,

$$\begin{aligned} \alpha_A &= \frac{w^{i-1} - t^{i-1} \cdot \frac{w^i - w^{i-1}}{t^i - t^{i-1}} - b_{ADV} \cdot \sigma^j}{\theta_{DT}}, \quad \beta_A = \frac{w^i - w^{i-1}}{t^i - t^{i-1} + b_{ADV}} \\ \alpha_D &= \frac{w^{i-1} - t^{i-1} \cdot \frac{w^i - w^{i-1}}{t^i - t^{i-1}} + b_{DEL} \cdot \sigma^j}{\theta_{DT}}, \quad \beta_D = \frac{w^i - w^{i-1}}{t^i - t^{i-1} - b_{DEL}} \end{aligned}$$

4. 基于以上推导, 即可给出出发时间选择概率 P_i^j 和满意度 W^j 的数值计算流程

Algorithm 11.4 出发时间选择概率 P_i^j 和满意度 W^j 计算

输入: $ADV, DEL, b_{ADV}, b_{DEL}, \theta_{DT}, w^i, \forall i, j = 0, \dots, I$

```

1: for  $\forall j = 0, \dots, I$  do
2:    $A = D = 0, W^j = 0$ 
3:   for  $\forall i = 0, \dots, I$  do                                ▶ 初始化
4:      $P_i^j = 0$ 
5:   end for
6:   while  $\tau^A < \sigma^j - ADV, A < I$  do                ▶ 找到  $\sigma^j - ADV$  对应的时间索引
7:      $A = A + 1$ 
8:   end while
9:   while  $\tau^D < \sigma^j + DEL, D < I$  do                ▶ 找到  $\sigma^j + DEL$  对应的时间索引
10:     $D = D + 1$ 
11:  end while
12:   $i = A$                                               ▶ 计算  $(\sigma^j - ADV, \tau^A]$  范围内的积分 (处理  $\tau^A > \sigma^j - ADV$  的情况)
13:   $\alpha = \left( w^{i-1} - \tau^{i-1} \cdot \frac{w^i - w^{i-1}}{t^i - t^{i-1}} - b_{ADV} \cdot \sigma^j \right) / \theta_{DT}$ 
14:   $\beta = \left( \frac{w^i - w^{i-1}}{t^i - t^{i-1}} + b_{ADV} \right) / \theta_{DT}$ 
15:   $P_A^j = \frac{\exp\{\alpha\}}{\beta} (\exp\{\beta \tau^A\} - \exp\{\beta(\sigma^j - ADV)\})$ 
16:   $W^j = P_A^j$ 

```

```

17:   for  $\forall i = A + 1, \dots, j$  do                                ▷ 计算  $(\tau^A, \sigma^j]$  范围内的积分
18:      $\alpha = \left( w^{i-1} - \tau^{i-1} \cdot \frac{w^i - w^{i-1}}{\tau^i - \tau^{i-1}} - b_{ADV} \cdot \sigma^j \right) / \theta_{DT}$ 
19:      $\beta = \left( \frac{w^i - w^{i-1}}{\tau^i - \tau^{i-1}} + b_{ADV} \right) / \theta_{DT}$ 
20:      $P_i^j = \frac{\exp\{\alpha\}}{\beta} (\exp\{\beta\tau^i\} - \exp\{\beta\tau^{i-1}\})$ 
21:      $W^j = W^j + P_i^j$ 
22:   end for
23:   for  $\forall i = j + 1, \dots, D - 1$  do                                ▷ 计算  $(\sigma^j, \tau^{D-1}]$  范围内的积分
24:      $\alpha = \left( w^{i-1} - \tau^{i-1} \cdot \frac{w^i - w^{i-1}}{\tau^i - \tau^{i-1}} + b_{DEL} \cdot \sigma^j \right) / \theta_{DT}$ 
25:      $\beta = \left( \frac{w^i - w^{i-1}}{\tau^i - \tau^{i-1}} - b_{DEL} \right) / \theta_{DT}$ 
26:      $P_i^j = \frac{\exp\{\alpha\}}{\beta} (\exp\{\beta\tau^i\} - \exp\{\beta\tau^{i-1}\})$ 
27:      $W^j = W^j + P_i^j$ 
28:   end for
29:    $i = D$                                               ▷ 计算  $(\tau^{D-1}, \sigma^j + DEL]$  范围内的积分 (处理  $\tau^D > \sigma^j + DEL$  的情况)
30:    $\alpha = \left( w^{i-1} - \tau^{i-1} \cdot \frac{w^i - w^{i-1}}{\tau^i - \tau^{i-1}} + b_{DEL} \cdot \sigma^j \right) / \theta_{DT}$ 
31:    $\beta = \left( \frac{w^i - w^{i-1}}{\tau^i - \tau^{i-1}} - b_{DEL} \right) / \theta_{DT}$ 
32:    $P_D^j = \frac{\exp\{\alpha\}}{\beta} (\exp\{\beta(\sigma^j + DEL)\} - \exp\{\beta\tau^{D-1}\})$ 
33:    $W^j = W^j + P_D^j$ 
34:   for  $\forall i = A, \dots, D$  do                                ▷ 完成  $P_i^j$  计算
35:      $P_i^j = P_i^j / W^j$ 
36:   end for
37:    $W^j = \theta_{DT} \cdot \ln(W^j)$                                 ▷ 完成  $W^j$  计算
38: end for

```

5. 已知理想出发时间 σ 确定时实际出发时间 $\tau' \in (\tau^{i-1}, \tau^i]$ 的概率 $P((\tau^{i-1}, \tau^i]|\sigma)$, 则可以积分得到 $(\tau^{i-1}, \tau^i]$ 时段内的实际出发需求量 D^i

$$D^i = \int_{\tau^{i-1}-DEL}^{\tau^i+ADV} q(\sigma) \cdot P((\tau^{i-1}, \tau^i]|\sigma) d\sigma$$

又因为假设需求量为分段常函数, 因此任意时刻 τ 的需求发生率 $d(\tau) = d^i = \frac{D^i}{(\tau^i - \tau^{i-1})}$ 。又因为 $P((\tau^{i-1}, \tau^i]|\sigma)$ 简化为分段线性函数, $q(\sigma)$ 简化为分段常函数, 则实际需求分布 d^i 的数值计算流程如下

Algorithm 11.5 实际需求分布 d^i 计算

```

输入:  $ADV, DEL, q^i, P_i^j, \forall i, j = 0, \dots, I$ 
1:  $A = D = 0$ 
2: for  $\forall i = 0, \dots, I$  do
3:   while  $\sigma^D < \tau^i - DEL, D < I$  do
4:      $D = D + 1$ 
5:   end while
6:   while  $\sigma^A < \tau^i + ADV, A < I$  do
7:      $A = A + 1$ 
8:   end while
9:    $d^i = q^D \cdot 0.5 \cdot P_i^D \cdot (\tau^D - (\tau^i - DEL)) / (\tau^i - \tau^{i-1})$ 
10:  for  $\forall j = D + 1, \dots, A - 1$  do
11:     $d^i = d^i + q^j \cdot 0.5 \cdot (P_i^j + P_i^{j-1}) \cdot (\sigma^j - \sigma^{j-1}) / (\tau^i - \tau^{i-1})$ 
12:  end for
13:   $d^i = d^i + q^A \cdot 0.5 \cdot P_i^{A-1} \cdot ((\tau^i + ADV) - \tau^{A-1}) / (\tau^i - \tau^{i-1})$ 
14: end for

```

11.1.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
embody	体现 (v)	enumeration	枚举	in the case of	就…来说
procedure	流程	mutually	互相地	analyst	分析师 (n)
generic	通用的	biunivocal	一对一的	acyclic	无环的
apart from	除了…	apart	分开 (adv), 不同的	inverse	相反的、反转 (v)
monotone	单调、单调的	recursive	递归的	infinitesimal	无穷小 (n,adj)
analogy	类比 (n)	piece-wise	分段的	ad hoc	特别地、特别的
monodimensional	一维的	dwelling	住所	alight	从 (公交车、火车) 下车、点燃

11.2 A novel metamodel-based framework for large-scale dynamic origin-destination demand calibration (TRC, 2022)

ABSTRACT

由于计算限制, 标定随机仿真器的动态交通需求难度较大。本文借助元模型优化方法, 提出了可高效标定大规模路网动态OD矩阵的新型框架。元模型基于最新提出的双模式宏观基本图 (**bi-modal macroscopic fundamental diagram**) 理论捕捉宏观交通演化。以广泛采用的SPSA算法作为对照验证所提算法的有效性。基于所提算法标定得到的OD需求可高效且真实复现路网各路段交通状态, 与最新研究相比所提方法在效率层面也具有优势。方法应用时不受路网拓扑结构影响, 其结果受模型参数影响也较小。

11.2.1 Introduction

- 交通需求可以说是一系列交通研究问题（规划、设计、组织）的核心要素。近年来大量多模态数据的涌现为揭示大规模路网运行状态提供了空前的支撑, 但尚无合适的方法合理利用上述数据, 特别是在交通需求标定领域;
- 交通需求标定问题复杂且棘手, 其主要难点包括以下三点: 1) 如何保证解的鲁棒性; 2) 如何捕捉OD矩阵的随机性和动态性; 3) 如何平衡精度与计算复杂度;
- 本文提出了多模式网络下基于双模式宏观基本图 (**bi-modal macroscopic fundamental diagram**) 的车辆OD需求快速标定方法以应对上述三类挑战;
- 在上世纪, 绝大多数研究仅基于流量数据估计OD矩阵, 典型方法包括重力模型、极大似然模型、广义最小二乘模型、最大熵模型和贝叶斯方法等, 且研究普遍不考虑OD需求的动态性;
- 如今部分研究尝试挖掘多模态数据提升OD估计准确性, 包括行程时间数据、手机信令与GPS数据、车牌识别数据、公交刷卡数据等等;
- 丰富的交通数据也提升了OD标定⁵的热度。通用方法包括SPSA算法、遗传算法、卡尔曼滤波算法等等, 但相关算法在计算效率上不具备优势;
- 目前关于OD标定的一个关键瓶颈在于缺乏评估大规模网络标定结果的定量模型, 具体表现为难以在宏观集计层面联系OD矩阵与复杂交通演化;
- 近期宏观交通流理论的进展有助于解决上述瓶颈。宏观基本图 (**macroscopic fundamental diagram, MFD**) 理论建立了网络流量与网络密度的关系, 反映了网络基本性质, 且受需求模式影响较小⁶。而在传统的单一模式宏观基本图 (**uni-modal MFD**) 基础上, 双模式宏观基本图 (**bi-modal MFD**) 针对小汽车与地面公交混行路网, 可表征多模式路网层面的交通状态和拥堵水平, 已被用于道路空间分配、边界信号控制等领域;
- 基于双模式MFD, 提出了面向OD标定的新型元模型仿真优化框架。元模型优化方法最大的难点在于克服元模型与仿真模型的偏差, 本文基于MFD提供的网络级交通状态信息对该问题, 同时显著减少未

⁵本文区分OD估计(OD estimation)与OD标定(OD calibration)的概念。

⁶MFD相关概念可参考第5.1节

知参数数量：

10. 研究的贡献主要包括：

- 提出了面向双模式网络的 OD 需求标定方法；
- 首次将 MFD 理论应用于 OD 估计；
- 提出了应对解析模型与仿真模型偏差的元模型优化方法；
- 利用多源数据构建双模式 MFD；
- 通过多个城市路网验证了所提标定方法的可扩展性、准确性和有效性。

11.2.2 Methodology

1. 将大规模网络划分为 N 各包含小汽车和常规公交的子网。理想的子区划分应使得子区内部的交通特征相似而突出不同子区间交通特征的差异性，且应使得各子区的拓扑特征相似。将子区内部的交通生成和吸引量集中于质心，则研究目标为标定以子区为单位的 OD 需求矩阵。令 I, T 分别为子区和研究时段集合，OD 矩阵为 D ， $D_{i,j}(t)$ 为 t 时刻自子区 i 至 j 的需求， $D_{i,j}^0(t)$ 为初始输入需求， $N_i(t)$ 表示子区 i 时刻 t 实测的网络总密度， $n_i^c(t; D), n_i^b(t; D)$ 分别表示子区 i 时刻 t 仿真得到的小汽车和公交密度，则 OD 矩阵标定可建模如下

$$\min_D \sum_{i \in I} \sum_{t \in T} (N_i(t) - \mathbb{E}[n_i^c(t; D) + n_i^b(t; D)])^2 + \delta_1 \sum_{i \in I} \sum_{j \in I} (D_{i,j}^0(t) - D_{i,j}(t))^2$$

式中第一项为仿真网络子区密度与实测密度的误差；第二项为正则化误差，避免标定后的 OD 矩阵与输入结果相比差距过大。参数 δ_1 为权重系数，其大小反映了初始输入 OD 矩阵的可信度。敏感性分析结果显示其大小对优化结果影响不大；

2. 为计算仿真结果 $n_i^c(t; D), n_i^b(t; D)$ ，将以子区为单位集计的 OD 矩阵 D 按规则分布至子区各节点中进行微观仿真。分布方式可为均匀分布（符合 MFD 的同质性假设）、按人口密度分布（基于实测数据或重力模型）等等；
3. 问题的维度为 $I \times I \times T$ ，显然为高维优化问题。同时优化多个决策变量具有较大难度，且仿真优化问题对计算时长具有较高要求。为此研究构建针对性的元模型优化算法以期只需较少的迭代完成标定；
4. 此类研究一般以连边交通状态作为仿真优化依据，本研究以子区状态为依据具有如下优势：
 - 因为不可能同时获得路网所有连边的状态，通用的做法为仅选取部分代表性连边，但也会导致参考信息较少。本研究则只需子网级的真实数据，并通过仿真推断各连边状态；
 - 若考虑每一连边的基本图构造元模型参数量较大，而以子区为单位则可改善该问题。
5. 在元模型优化前，首先基于多源数据提取真实子区状态指标并构建双模式 MFD。具体地，基于蓝牙行程时间数据和流量数据估计小汽车密度相关指标，基于公交刷卡数据估计公交密度相关指标；
6. 元模型仿真优化的基本框架详见第 7.6 节，本文大体包括以下步骤：
 - 基于双模式 MFD 构建元模型捕捉宏观网络状态，进而对初始 OD 需求完成一次优化；
 - 基于优化后的 OD 需求进行微观仿真，集计宏观网络状态，计算与实测状态误差；
 - 若误差小于预设阈值则结束标定；反之则基于仿真宏观网络状态更新元模型参数，进行新一轮优化。
7. 进一步介绍基于双模式 MFD 的元模型仿真优化数学模型。令 $n_i^c(t), n_i^b(t)$ 分别表示双模式 MFD 模型中的区域小汽车和地面公交密度，则可构建元模型 $f_1(D; \beta)$ 以近似仿真网络子区密度与实测密度的误差，进而将 OD 矩阵标定问题的目标函数简化为

$$\min_D f_1(D; \beta) + \delta_1 \sum_{i \in I} \sum_{j \in I} (D_{i,j}^0(t) - D_{i,j}(t))^2, \quad f_1(D; \beta) = \sum_{i \in I} \sum_{t \in T} (N_i(t) - (n_i^c(t) + n_i^b(t)))^2$$

式中 β 为元模型参数，旨在拟合 MFD 模型理论结果与高精仿真结果的误差。MFD 理论旨在建模网络层面基本交通状态间的关系。在小汽车与地面公交混行环境下，双模式 MFD 模型假设

$$v_i(t) = a_i + a_i^c n_i^c(t) + a_i^b n_i^b(t), \quad TT_i(t) = \frac{L_i}{v_i(t)}$$

式中 $v_i(t), TT_i(t)$ 分别表示子区 i 网络 t 时刻车辆的平均速度和平均行程时间, L_i 表示子区 i 中各车辆路径的平均长度 (经典的 MFD 模型普遍假设区域内各车辆路径的平均长度随时间不变), a_i, a_i^c, a_i^b 即为双模式 MFD 模型参数, 由多源观测数据预先标定。显然子区的宏观交通状态由密度 $n_i^c(t), n_i^b(t)$ 的时间演化决定。因为公交的运营模式较为简单, 故区域公交密度 $n_i^b(t)$ 的时间演化直接基于时刻表确定, 而小汽车密度 $n_i^c(t)$ 的演化则需考虑网络需求于各子区间的流出-流出特征。直接基于 MNL 模型建模各子区间的需求转移

$$P_r(t) = \frac{\exp\{\theta \cdot TT_r(t)\}}{\sum_{l \in L} \exp\{\theta \cdot TT_l(t)\}}$$

式中 $P_r(t)$ 为选择 “路径” r 的概率, 所谓 “路径” 非具体的出行路径, 而是由一系列相邻子区串联的序列, 相应的路径行程时间 $TT_r(t)$ 为路径所经各子区行程时间之和。基于需求转移概率即可建模相邻子区间的需求转移

$$M_{i,j}^k(t) = \sum_{r \in R} P_r(t) O_{i,j}(t), \quad O_{i,j}(t) = \frac{n_{i,j}^c(t) v_i(t)}{L_i}$$

式中 $M_{i,j}^k(t)$ 表示自子区 i 转移至相邻子区 k 并以子区 j 为最终目的地的需求, 集合 R 为自子区 i 途经相邻子区 k 最终至子区 j 的子区序列集合, $O_{i,j}(t)$ 表示子区 i 中以子区 j 为最终目的地的总流量, 由区域平均速度 $v_i(t)$ 、平均路径长度 L_i 和区域 i 内以区域 j 为目的地的车辆密度 $n_{i,j}^c(t)$ 共同决定。在转移需求 $M_{i,j}^k(t)$ 的基础上进一步考虑相邻子区 k 的通行能力约束即可得到实际的转移流量 $\hat{M}_{i,j}^k(t)$

$$\hat{M}_{i,j}^k(t) = \min \left\{ M_{i,j}^k(t), C_{i,k} (n_k^c(t), n_k^b(t)) \right\}$$

显然子区 i, k 间的最大通行能力 $C_{i,k} (n_k^c(t), n_k^b(t))$ 一方面与两者边界处的拓扑特征有关, 另一方面则与子区 k 内的密度 $n_k^c(t), n_k^b(t)$ 有关。基于转移流量, 即可构建子区密度的动力学平衡方程, 确定 $n_{i,j}^c(t)$ 的时间演化特征

$$n_{i,j}^c(t+1) = \begin{cases} n_{i,j}^c(t) + D_{i,j}(t) + \sum_{k \in V_i} \hat{M}_{k,j}^i(t) - O_{i,j}(t) & i = j \\ n_{i,j}^c(t) + D_{i,j}(t) + \sum_{k \in V_i} \hat{M}_{k,j}^i(t) - \sum_{k \in V_k} \hat{M}_{i,j}^k(t) & i \neq j \end{cases}$$

式中 $D_{i,j}(t)$ 表示 t 时刻子区 i 内生成的以子区 j 为最终目的地的需求。显然基于 $n_{i,j}^c(t)$ 即可确定子区内的小汽车总密度 $n_i^c(t)$

$$n_i^c(t) = \beta_{i,t} \sum_{j \in I} n_{i,j}^c(t)$$

理论上子区 i 的小汽车密度 $n_i^c(t)$ 即为区域内各目的地小汽车密度 $n_{i,j}^c(t)$ 之和。但因为构建的 MFD 模型计算 $n_{i,j}^c(t)$ 时的子区内交通状态同质性假设 (如假设区域内各目的地出行的平均行程距离 L_i 一致), 其结果可能存在误差, 故简单地引入元模型参数 $\beta_{i,t}$ 进行误差修正。以上即为基于双模式 MFD 元模型的 OD 标定数学模型;

8. 在元模型迭代优化的过程中, 需动态更新元模型参数 $\beta_{i,t}$ 以修正 MFD 理论结果与高精仿真结果的误差。对于第 h 轮迭代, 元模型参数 $\beta_{i,t}^h$ 基于下式加权最小二乘确定

$$\begin{aligned} \beta_{i,t}^h &= \eta_{i,t}^h \beta_{i,t}^{h-1} \\ \eta_{i,t}^h &= \arg \min_{\eta^h} \sum_{s \in S} w_h(D_s) \sum_{i \in I} \sum_{t \in T} (\eta_{i,t}^h \beta_{i,t}^{h-1} n_{i,s}^c(t) - \mathbb{E}[n_i^c(t; D_s)])^2 + w_0 \sum_{i \in I} \sum_{t \in T} (\eta_{i,t}^h - 1)^2, \quad \text{s.t. } \beta_{i,t}^l \leq \eta_{i,t}^h \beta_{i,t}^{h-1} \leq \beta_{i,t}^u \end{aligned}$$

式中 $\eta_{i,t}^h$ 为对第 $h-1$ 轮优化时元模型参数 $\beta_{i,t}^{h-1}$ 的缩放系数, 用于确定新一轮优化的元模型参数 $\beta_{i,t}^h$; 元模型参数 $\beta_{i,t}^h$ 应不超过上下界 $\beta_{i,t}^l, \beta_{i,t}^u$; S 为之前 $h-1$ 轮优化中的仿真采样集合, D_s 为样本 s 对应的 OD 需求。上述优化目标函数第一项为最小化 MFD 理论结果与仿真的偏差, 而第二项为正则项, 避免迭代过程中元模型参数 $\beta_{i,t}^h$ 变化过大。正则误差权重 w_0 自定义, 而样本权重 $w_h(D_s)$ 定义为

$$w_h(D_s) = \frac{1}{1 + c \|D_s - D_h\|}$$

式中参数 c 同样需自定义，研究取 1×10^{-8} 。

11.2.3 Demonstrating in the Sioux-falls network

1. 首先在小规模混合交通网络（SiouxFall 路网）中验证所提 OD 标定算法的效果。将路网划分为三个子区，每一子区均具有唯一的 MFD；
2. 仿真时段为上午 7 点至 10 点，集计时间间隔为 15 分钟；
3. 进行 10 轮仿真优化，算法于第 6 次迭代后目标函数基本收敛，标定后的 OD 需求可较好地复现瓶颈路段的交通状态时间演化。选择以下算法作为对照：
 - SPSA 算法：目标函数与所提算法一致，均以子网集计状态指标作为标定依据；
 - Aimsun 仿真软件自带 OD 标定算法：基于卡尔曼滤波，以连边状态数据作为标定依据；
 - 经典元模型算法：组合 MFD 理论模型与线性统计模型构建元模型（见第 7.6 节），以连边状态数据作为标定依据。
4. 对比所提算法，发现：
 - 标定完成后无论从子区尺度集计或连边尺度，所提元模型方法的 OD 标定均显著优于 SPSA 算法，而 Aimsun 自带算法的效果甚至弱于 SPSA 算法；
 - 针对所考虑的小型路网，所提算法、SPSA 与 Aimsun 三种标定方法的用时基本接近。其中 SPSA 算法与 Aimsun 的 OD 标定算法的计算时间将随仿真时间的增加而增加，而所提元模型标定算法的计算时间则与网络子区数正相关；
 - 进一步讨论网络子区划分数目对标定效果与用时的影响。发现随着子区数增加标定效果改善而用时增加，且只需较少的子区划分数目（满足子区内同质性假设保证 MFD 存在性）即可保证较好的标定效果；
 - 对比经典元模型算法，当网络子区划分数目较少时，所提元模型算法在用时和标定结果上均优于经典元模型算法；而如果精细划分网络使得每个子区仅包含少数连边，则两者的标定结果可能接近，且经典元模型算法于用时上可能具有优势；
 - 最后讨论目标函数中的正则权重 δ_1 与初始需求分布对标定结果的影响。选择三种 δ_1 取值（0.1、0.001、0.0001）与三种初始需求强度（低需求、中等需求、重需求），发现其取值对所提算法最终优化结果影响并不显著。

11.2.4 Case study of a large-scale network problem for the Melbourne CBD

1. 进一步将所提 OD 标定算法应用于墨尔本 CBD 大规模网络，将网络划分为 6 个子区；
2. 为得到真实地网络子区状态集计数据，采用以下三类数据集：

- 行程时间蓝牙数据集：基于相应路段的行程速度估计区域平均速度 $v_i(t)$

$$v_i(t) = \frac{\sum_{b \in B_i} v_b(t) l_b}{\sum_{b \in B_i} l_b}$$

式中 $v_i(t)$ 为区域 i 的集计速度； $v_b(t)$ 为蓝牙检测器于路段 b 实测的数据； l_b 为路段 b 的长度； B_i 为区域 i 中部署蓝牙检测器的路段的集合；

- SCATS 流量数据集：首先基于相应路段的流量估计区域流量 $Q_i(t)$

$$Q_i(t) = \frac{\sum_{g \in G_i} \sum_{l \in L_g} q_{g,l}(t)}{N_{G_i}} TL_i$$

式中 $Q_i(t)$ 为区域 i 的集计流量； $q_{g,l}(t)$ 为 SCATS 数据集提供的交叉口 g 的进口道 l 的流量； G_i, N_{G_i} 分别表示 SCATS 数据集包含的区域 i 交叉口集合和交叉口数目； L_g 为交叉口 g 的进口道集合； TL_i 为区域 i 内的路网总长。基于区域流量与速度，即可估计区域密度 $N_i(t) = \frac{Q_i(t)}{v_i(t)}$ ；

- 公交刷卡数据集：用于估计区域公交状态。

3. 仿真验证时长为 4.25 小时（上午 5:45-10:00），以 15 分钟为单位进行时间集计；

4. 首先基于 Aimsun 自带的 OD 估计算法生成初始 OD 需求，发现仿真结果与实际状态存在较大偏差；
5. 应用所提标定算法对初始 OD 需求进行标定，于第 9 轮优化时得到最优解。基于标定后的 OD 需求进行仿真，与实际状态高度接近。

11.2.5 英汉互译

English	Chinese	English	Chinese	English	Chinese
underpinned	加强、加固 (v)	arguably	按理说	unprecedented	空前的
tedious	冗长的	intractable	棘手的	tripod	三脚架
parsimonious	极简的	drastically	彻底地		

11.3 SUMO 换道模型

Fundamentals of Traffic Simulation

Sec. 7.3.2 Lane-Changing Model

SUMO 换道模型源码: https://sumo.dlr.de/doxygen/db/d77/_m_s_l_c_m__l_c2013_8cpp.html

11.3.1 DK2008

1. DK2008 换道模型是 SUMO 早期版本（至 0.18.0）的默认换道模型，由 Daniel Krajzewicz 提出；
2. 截至 SUMO 发展时期，纵向动力模型（跟驰模型）已经基本成熟，具有充分的健壮性，但横向动力模型（换道模型）仍有较大发展空间。由 Krauß 提出的跟驰模型⁷仅基于战术决策（驾驶员对提速的需求）描述驾驶行为，而未考虑战略层面上对车道的选择需求（选择特定的车道以保证正确的路径）；
3. 在 Krajzewicz 提出的换道模型中解决了上述问题。模型首先定义有效车道 (valid lane)，车辆在这些车道上可无需换道而保证正确的路径；
4. 模型中进一步定义车辆 i 在 t 时刻进行换道所需的距离 $d_{lc,i}(t)$ 。当车辆所处的位置距必须换道点（若此时不在有效车道上则将走向错误路径）小于 $d_{lc,i}(t)$ 时，车辆必须换至有效车道上；反之车辆根据充分利用道路占用率的原则可于任意车道行驶；

$$d_{lc,i}(t) = \begin{cases} v_i(t) \cdot \alpha_1 + 2l_i, & v_i(t) \leq v_{thresh} \\ v_i(t) \cdot \alpha_2 + 2l_i & v_i(t) > v_{thresh} \end{cases}$$

- $v_i(t)$: 车辆 i 于时刻 t 的速度 (m/s)；
- v_{thresh} : 区分车辆与公路和城市道路不同换道行为的速度阈值 (m/s , 默认为 $14m/s$)；
- α_1, α_2 : 比例因子（分别取 $5s, 15s$ ）；
- l_i : 车辆 i 的长度 (m)。

需要说明的是，在计算车辆所处位置距必须换道点的距离时应扣除前车长度，从而避免车辆在有其它选项的情况下跟驰于拥挤车道，同时保证其它车道的车辆在必要时刻汇入拥挤车道避免路径错误；

5. 以上过程为战略决策中判断车辆是否换至有效车道的基本原则；逆过程（判断车辆是否从有效车道换至其它车道）的判断方法同理；
6. SUMO 的换道模型 DK2008 中同样包括战术决策——通过换道获得更高的速度——的过程，基于 Ehmanns 于 2001 年提出的模型⁸。模型根据换道的效益进行决策。车辆于时刻 t 换至相邻车道 l_n 的效益 $b(t, l_n)$ 由车辆于当前车道 (l_c) 和目标车道 (l_n) 所能达到的最大安全速度表征

$$b(t, l_n) = \frac{v_{pos}(t, l_n) - v_{pos}(t, l_c)}{v_{max}(l_c)}$$

- $v_{pos}(t, l)$: 时刻 t 时车道 l 上的车辆的安全速度 (m/s)，由跟驰模型计算；
- $v_{max}(l)$: 自由流状态下车道 l 允许的最大速度。

⁷SUMO 早期版本默认跟驰模型，新版本默认跟驰模型为其改良版

⁸Ehmanns D (2001). *Simulationsmodell des menschlichen Spurwechselverhaltens; Simulation model of human lane change behavior*.

7. 进一步地介绍战术换道的决策逻辑。模型认为驾驶员的换道意愿可以基于某一次强烈的激励 ($b(t, l)$ 为大值, 也可以是多次轻微正向激励累积的结果 (连续多个较小 $b(t, l)$ 叠加), 由此模型需要相关变量 (*speedGainProbability*) 表征激励的累积。*speedGainProbability* 的大小表示换至对应车道的意愿, 基于每个仿真步长的结果进行更新, 当意愿超过预设的阈值时, 车辆将执行换道, 同时对 *speedGainProbability* 重新初始化;
8. 基于每一仿真步长的 $b(t, l)$ 更新换道意愿的具体过程如下:
 - (a) *speedGainProbability* 为二维向量, 分别储存向左侧和右侧相邻车道换道的意愿, 初始值均为 0;
 - (b) 以 l_l, l_r 分别表示左侧车道和右侧车道。当 $b(t, l_l) > 0$ 时, 在原有左向换道的意愿上叠加 $b(t, l_l)$, 认为该换道意愿可以鼓励; 当 $b(t, l_l) < 0$ 时, 对原有左向换道的意愿除以 2, 表示抑制该意愿。对右侧换道的意愿更新过程同上;
 - (c) 当向某一侧换道的意愿超过设定的阈值且持续三个仿真步长时, 执行换道, 同时重新初始化 *speedGainProbability* 为 0。
9. 以上即为战术换道的基本决策流程。战术换道决策的过程本质上是车道相互比较的过程, 得到的是相互关系, 而非绝对值, 由此带来的附加问题是相对更优的车道依然可能没有充分的空间保证其他车辆汇入。此时, 换道车辆会与目标车道上下游车辆沟通, 上下游车辆对应作出速度调整, 尽可能提供换道空间。
10. 以上即是 SUMO DK2008 换道模型的全部介绍。模型同时保证了尽可能高的车道利用率和尽可能及时的换道决策。但模型未考虑其它车辆的运动状态。当有效车道于必须换道点附近被占用时, 模型将趋于抑制换道意愿, 由此带来问题。

SUMO's Lane-Changing Model

11.3.2 LC2013

Introduction

1. LC2013 换道模型为 SUMO 目前的默认换道模型, 由 Jakob Erdmann 基于 DK2008 提出。早期的换道模型在仿真过程中存在以下问题:
 - (a) 当大多数车需要于高速公路分叉口换道时将出现严重拥堵, 与现实情况下不同;
 - (b) 当入口匝道有车辆汇入时, 高速公路上相关方向车辆会停驶, 造成严重拥堵;
 - (c) 车辆未及时换入有效车道时也会形成严重拥堵;
 - (d) 对双车道环岛, 车辆仅占用外侧车道, 从而形成拥堵。
2. 新的换道模型在仿真过程中主要完成以下两项工作:
 - (a) 在每一仿真步长中基于行车路径和当前及之前车辆周围的交通环境进行换道决策;
 - (b) 进一步地计算因为换道导致的当前车辆和被阻碍车辆的速度变化, 保证换道的成功。
3. 在当前换道模型中, 将换道的原因由重要性自大到小分为以下四类:
 - (a) **Strategic change:** 战略性换道, 车辆为了保证路径正确而进行的换道;
 - (b) **Cooperative change:** 配合性换道, 车辆为了方便其它车辆成功换道而进行的换道;
 - (c) **Tactical change:** 战术性换道, 车辆为了加速而进行的换道;
 - (d) **Regulatory change:** 规范性换道, 车辆为适应法律要求让出超车道所进行的换道。

Architecture

1. 车辆换道成功的必要条件: 目标车道上存在足够的空间, 即换道后车辆距前车和后车的距离均不至于过小 (临界距离由跟驰模型定义);
2. ***ego vehicle:*** 当前考虑是否换道的目标车, 与前车和后车区分;
3. 默认情况下车辆的换道是瞬时完成的。可通过修改仿真选项 **-lanechange.duration** 实现连续过程, 但该功能没默认设置成熟。

Strategic lane changing

- **evaluating subsequent lanes:**

1. **bestLanes**: 即有效车道，车辆在这些车道上可无需换道直至下一路段；
2. **occupation**: bestLanes 上的交通密度；
3. **bestLaneOffset**: 路段上每一车道的标签 (index) 与最近有效车道标签之间的偏差 ($index_{best} - index$)。

- **determining urgency:**

1. 当仿真车接近必须换道点时，车辆有一定的概率选择尽早换至有效车道或晚一些换道而形成冲突，换道的急迫性取决于以下因素：
 - (a) 距必须换道点的距离（负相关）；
 - (b) 预估的到达必须换道点的速度 (**lookAheadSpeed**, 基于当前速度和之前速度推算)；
 - (c) **bestLaneOffset** 的绝对值的大小；
 - (d) 最终目标车道（有效车道）的占用率；
 - (e) 瞬时目标车道（转至有效车道过程中需要经过的其它车道）的占用率。
2. 当以下关系式成立时，认为换道需求为紧急

$$d - o < lookAheadSpeed \cdot |bestLaneOffset| \cdot f$$

- d : 仿真车距必须换道点的距离；
- o : 基于车道占用率的折减（目前仅考虑 **bestLanes** 的占用率，折减系数疑似取 1）；
- f : 代表成功换道所需的特征时间，左向换道取 10、右向取 20.

3. **remainingSeconds**: 可用于换道的时间，由到达必须换道点的期望时间除以 $|bestLaneOffset|$ 。

- **speed adjustment to support lane changing:**

1. 当准备换道车辆的目标车道存在其它车辆影响换道时，车辆会通过调整速度实现换道。一般来说车辆会以最大安全速度行驶，故速度调整一般是通过减速实现。然而，跟驰模型中车辆可能有一随机组件 **dawdling**，该组件可避免车辆以最大加速度加速，从而禁用该随机组件可实现一定程度上的加速调整；
2. 为方便进行合适的速度调整，根据换道车 (EGO) 的计划速度 (**plannedSpeed**)、障碍车辆的速度、间距和剩余时间 (**remainingSeconds**) 划分一下几种情况，并提出相应策略：
 - (a) **前车形成障碍**:
 - i. **可超车**: 要求前车避免加速、换道车加速超车（法律可能禁止向右超车 **-lanechange.overtake-right**）；
 - ii. **无法超车**: 减速或保持原速等待前车加速。
 - (b) **前车未形成障碍**: 采用在保证安全车头时距和间距的前提下最大速度；
 - (c) **无前车**: 采用最大安全速度；
 - (d) **后车形成障碍**:
 - i. **可在后车之前插入**:
 - A. **换道车的当前速度足够快**: 要求后车避免加速、关闭换道车 **dawdling** 组件；
 - B. **后车只需一次减速即可形成充足空间**: 要求后车减速、关闭换道车 **dawdling** 组件。
 - ii. **需要让后车超车**:
 - A. **后车可稍微减速以方便换道车充分减速**: 要求后车稍微减速、换道车充分减速；
 - B. **后车需要快速超车**: 关闭后车 **dawdling** 组件、换道后车充分减速。
 - iii. **后车无法向右超车**: 要求后车减速。
 - (e) **后车未形成障碍**: 要求后车保持原速；
 - (f) **无后车**: 采用最大安全速度。
3. **当前车形成障碍时**, 只有在满足以下几个条件时才会认为可以超车:
 - (a) 目标车的计划车速高于障碍车车速 ($dv = plannedSpeed - blocker speed > 0$);

- (b) 障碍车位于目标车左侧（法律允许时也可位于右侧）；
- (c) 距必须换道点的距离满足成功超车的要求；
- (d) 剩余时间 (*remainingSpeed*) 满足以当前 *dv* 实现超车的要求。

• **preventing deadlock:**

1. 锁死 (*deadlock*) 现象是指相邻车道的两辆车同时需要换至对方车道 (*counterLaneChange*)，且两辆车均到达必须换道点而无法错开空间⁹，则两辆车均会停止，从而将两个车道都锁死。发生锁死后目前唯一的解决方法是设置“传送装置”，在到达预设的时间阈值后将锁死车辆移走；
2. 为避免锁死现象的发生，当两车互为 *counterLaneChange* 关系时，会要求后车尽量减速，为前车让出空间；另外在车辆需要进行多车道换道时，车辆会在必须换道点前预留额外的空间，目前右向换道时预留空间为 20m，左向换道为 40m，通过这种不平衡设置避免多车道换道时的锁死现象。

Cooperative lane changing

1. 在当前模型中，当车辆作为后车¹⁰影响到其它车道车辆换入本车道时，在条件允许时就会执行配合性换道，当配合性换道不成功时则会微调速度以方便后续仿真；
2. 配合性换道行为在多车道环形交叉口会增多。环形交叉口只有最外侧环道可供车辆汇入流出，因此出于战略考虑 (*strategic*)，车辆往往选择留在最外侧环道绕行，而这降低了多车道环形交叉口的通行能力。SUMO 会强制那些尚未到达出口的仿真车换入内侧环道绕行，这一决策有时会改善环形交叉口的通行能力。

Tactical lane changing

1. 战术性换道是换道效益与换道代价平衡的结果，其中换道效益以车速的增长衡量，而换道代价则是个主观量。新的换道模型 (LC2013) 中该部分的算法与旧版本 (DK2008) 完全相同；
2. 有时从其它车辆的右侧超车是不被允许的，此时后方快车即会进行战术换道，具体可以有以下两种选项：
 - (a) 后方快车换入左侧车道（即前方慢车车道），待前方慢车换入右侧车道后（即交换车道）从左侧超车；
 - (b) 当有多车道时，后方快车也可以选择连续左向换道至前方慢车左侧，从而自左侧超车。
 作为新换道模型的一部分，以上行为将在以下条件全部满足后执行：
 - (a) 选项 *--lanechange.overtake-right* 未设置；
 - (b) EGO 时速高于 60km/h¹¹；
 - (c) 有前车位于左侧相邻车道且速度慢于 EGO；
 - (d) 若 EGO 打算跟随前车，则必须减速。

Regulatory lane changing

1. 在靠右行驶的规定中，左侧车道往往被用于超车道，当车辆无超车需求时不允许占用超车道，从而产生换至右侧车道的需求。这一需求由变量 *keepRightProbability* 表征，该变量随时间减小，当低于预设阈值时触发换道。记当前与下一个仿真步长的 *keepRightProbability* 分别为 *q* 和 *q'*，则

$$q' = q - T \frac{t \cdot m}{d \cdot v}$$

- *t*: EGO 能够以全速行驶在右车道上的期望时间；
- *m*: 目标车道的最高速度；
- *d, v*: EGO 的目标速度与当前速度；
- *T*: 表征换道紧迫性的常量。

2. 上述迭代公式包含以下三个部分：

- *t* 越大 *q'* 越小，表示在右侧车道越能持续高速行驶的车辆越倾向离开超车道；
- *m/v* 越大 *q'* 越小，表示车速相对越慢的车辆越倾向于离开超车道；

⁹SUMO 仿真中无法实现倒车

¹⁰在实际中，车辆的转向提示灯一般也只会被后车接收。

¹¹60km/h 的阈值是一套复杂得多的规则的粗略近似值，这些规则取决于要建模国家的法律。在德国 60km/h 的取值并非来自法律条文，而是一个判断是否慢速的法律先例。

- $1/d$ 越大 q' 越小，表示越希望保持高速的车辆越倾向于留在超车道。
- 由此，上述公式在保证车辆尽快让出超车道的同时避免重复驶出驶入超车道现象的发生。

11.3.3 英汉互译

English	Chinese	English	Chinese	English	Chinese
derivation	起源 (n)	discriminate	区分、歧视	swell	膨胀
subtract	减去	scaling factor	比例因子	normalize	使标准化
look-ahead	前瞻、前瞻的	hierarchy	层次体系	suite	套房、套件
a host of	很多 (可数)	interplay	互相影响	blockage	阻塞 (n)、阻塞物
determinant	决定因素	underscore	强调 (v)、下划线	implausible	不合情理的
exhibit	展示 (v)	on-ramp	入口匝道	roundabout	环岛
undertake	承担	interweave	交织	impractical	不现实的
herein	在此处	maneuver	运动 (v)	regulatory	监管的
detailed account	详细说明	in terms of	从…方面	whereas	尽管、鉴于
customization	定制 (n)	swap	替换	some degree of	某种程度的
unambiguous	意思清楚的	pursue	追求	presume	推定 (v)
deem	认为 (v)	discount	折减	stochastic	随机的
dawdle	拖延 (v)	somewhat	稍微 (adv)	refrain (from)	克制 (v)
warrant	许可令、使必要	impediment	障碍 (n)	elapse	(时间) 消逝 (v)
reserve	预定、储存 (v)	rigorous	严格的	calibration	标定
be subject to	受…影响	detrimental	有害的	compel	强迫 (v)
arguable	无把握的、有证	oscillation	摆动 (n)、犹豫	increment	增加 (v)
precedent	据的	concretize	(使) 具体化	jurisdiction	司法权
designate	先例	mandate	强制执行 (v)	allusion	暗指 (n)
tune	指定 (v)	encapsulate	压缩、概括 (v)	schema	纲要

11.4 附录

11*I 有限容量排队论 (finite capacity queueing theory) 与路网建模

- 将交通网络中的每一车道视为一 M/M/1 排队系统，即认为车道的输入服从泊松分布，而服务服从负指数分布，则可基于排队论对路网交通状态进行建模。注意到车道存在通行能力限制，相应的排队模型则为有限容量排队模型 (finite capacity queueing model)；
- 记 i 表示车道 (排队) 编号， k_i 表示排队系统容量， N_i 表示排队系统车辆数，则当 $N_i = k_i$ 时排队系统达到饱和，系统关闭 (block)，对应于车道堵塞与溢流。定义 $P(N_i = k_i)$ 表示系统关闭的概率。当车道堵塞后，网络中已存在的需求将被迫于上游车道等待车道开启 (定义 P_i^f 表示服务系统 i 车辆因下游饱和而被迫等待的概率)，而网络中尚未存在的准备载入的车辆数则会消失 (相当于放弃出行)；
- 定义 λ_i 表示总到达率，包括来自网络上游的需求和尚未载入网络的需求； λ_i^{eff} 表示有效到达率 (effective arrival rate)，即去除放弃出行的实际载入服务系统的需求。则显然有

$$\lambda_i = \frac{\lambda_i^{eff}}{1 - P(N_i = k_i)} \iff \lambda_i^{eff} = \lambda_i \cdot (1 - P(N_i = k_i))$$

又定义 γ_i 为外部到达率 (external arrival rate)，即准备从网络外部载入服务系统的需求；定义 p_{ji} 表示由排队系统 j 至排队系统 i 的概率，反映网络内部其它服务系统载入当前服务系统的需求，则

$$\lambda_i^{eff} = \gamma_i \cdot (1 - P(N_i = k_i)) + \sum_j p_{ij} \lambda_j^{eff} \implies \lambda_i = \gamma_i + \frac{\sum_j p_{ij} \lambda_j^{eff}}{1 - P(N_i = k_i)} = \gamma_i + \frac{\sum_j p_{ji} \lambda_j \cdot (1 - P(N_j = k_j))}{1 - P(N_i = k_i)}$$

- 基于 p_{ij} ，可以建模 P_i^f 与 $P(N_i = k_i)$ 的关系

$$P_i^f = \sum_j p_{ij} \cdot P(N_j = k_j)$$

5. 在服务系统 i 关闭后, 定义 $\tilde{\mu}_i$ 表示疏通率 (unblocking rate), 即被堵在服务系统 i 的需求通过服务系统至下游服务系统的速率, 其倒数 $\frac{1}{\tilde{\mu}_i}$ 即为平均堵塞时间。考虑排队系统的服务率 (service rate) μ_i 和疏通率 $\tilde{\mu}_i$, 则系统的有效服务率 (effective service rate) μ_i^{eff} 为

$$\frac{1}{\mu_i^{eff}} = \frac{1}{\mu_i} + P_i^f \cdot \frac{1}{\tilde{\mu}_i} \quad \frac{1}{\tilde{\mu}_i} = \sum_{j \in \mathcal{D}_i} \tilde{p}_{ij} \frac{1}{r_{ij} \mu_j^{eff}} = \sum_{j \in \mathcal{D}_i} \frac{\lambda_j^{eff}}{\lambda_i^{eff}} \cdot \frac{1}{\mu_j^{eff}} \quad r_{ij} = \tilde{p}_{ij} \frac{\lambda_i^{eff}}{\lambda_j^{eff}} \quad \tilde{p}_{ij} = \frac{p_{ij} P(N_j = k_j)}{P_i^f}$$

式中 \tilde{p}_{ij} 表示排队系统 i 中被堵塞车辆的转移概率。而 r_{ij} 表示服务系统 j 的输入需求中来自服务系统 i 的被堵塞车辆的比例, 因此 $r_{ij} \mu_j^{eff}$ 即为服务系统 j 对服务系统 i 的被堵塞车辆的实际服务速率。则堵塞时间由下游所有服务系统 $j \in \mathcal{D}_i$ 对服务系统 i 的被堵塞车辆的实际服务时间加权平均决定, 而堵塞时间和通行时间共同构成了车道的平均通行时间;

6. 考虑服务系统的总到达率 λ_i (需求) 和有效服务率 μ_i^{eff} , 则可以得到服务系统的服务强度 (service intensity, 在路网模型中也为交通强度) ρ_i

$$\rho_i = \frac{\lambda_i}{\mu_i^{eff}}$$

7. 在一般 M/M/1 服务系统假设下, 服务强度为 ρ 的服务系统中有 n 个任务的概率 $P(n) = \rho^n(1 - \rho)$, 而因为本节中考虑带容量约束服务系统, 要求 $P(n > k) = 0$, 则在 $\sum_n P(n) = 1$ 的约束下, 有

$$P(N_i = k_i) = \frac{\rho_i^{k_i}(1 - \rho_i)}{\sum_{n=0}^{k_i} \rho_i^n(1 - \rho_i)} = \frac{\rho_i^{k_i}(1 - \rho_i)}{1 - \rho_i^{k_i+1}}$$

8. 最后考虑服务率 μ_i 的约束。对于连接有信号交叉口的车道, 其服务率应由饱和流率 s_i 和下游交叉口信号配时共同决定。记 S 为信号控制车道 (服务系统) 集合, \mathcal{P}_i 为相应信号交叉口的相位集合, x_p 为相位 p 的绿信比 (相位内绿灯长度占总周期长度的比例), 则

$$\mu_i = \sum_{p \in \mathcal{P}_i} x_p s_i \quad \forall i \in S$$

9. 以上即为基于有限容量排队论理论的路网建模示例。模型众多变量中 $\gamma_i, \mu_i, p_{ij}, k_i, s_i$ 属于超参, 其它变量均为内生变量。

第 12 章

车载自组网仿真

12.1 Simulation environment for VANET

ABSTRACT

- □ ×

车载自组织网络 (**Vehicular Ad-hoc Network, VANET**) 是移动自组织网络 (**Mobile Ad-hoc Network, MANET**) 的应用，有助于保障交通安全及智能运输系统的构建。与移动自组织网络相比，车载自组织网络存在一些特别的特性，如对道路拓扑结构的约束、多路径衰减、路侧障碍、交通流模型、出行模式、车辆速度和机动性的变异性、交通灯、交通拥堵以及驾驶员行为等等。为实现车载自组织网络的仿真，需要交通仿真平台和网络仿真平台的组合。本文主要介绍不同的仿真平台。

12.1.1 Introduction

1. 车载自组织网络 (VANET) 是目前最前沿的智能运输系统技术，可实现车辆（车载单元，on board unit - OBU）与路侧单元 (road side unit, RSU) 的无线通讯，保障通信范围内人的安全。VANET 具有以下特点：
 - (a) 基于轨迹的运动模式，具有可预测的位置和时变拓扑结构；
 - (b) 具有独立或相关速度的数量可变的车辆；
 - (c) 快速时变的信道（例如可能由于建筑物的阻挡频繁地影响信号的传输）；
 - (d) 车道约束的运动模式；
 - (e) 能源消耗降低的需求。
2. VANET 与 MANET 存在多处相似，但也有一些特性。例如车辆的快速移动使得网络的拓扑结构具有时变性，但同时这种拓扑结构又受到路面结构、周围建筑和交通设施等的限制。因此传统的协议无法提供可靠、高效率、低延迟的 VANET 仿真；
3. VANET 仿真包括交通仿真和网络仿真两部分，具体包括以下两步：
 - (a) 首先是实现车辆间的通信（网络仿真）；
 - (b) 其次是实现网络节点的机动性（交通仿真）。

12.1.2 Simulation environment

表 12.1 traffic simulator

Simulation in urban mobility, SUMO	SUMO 是一款基于 C++ 的开源离散时间微观交通仿真模块，用于城市交通和运输的仿真。SUMO 中车辆的属性包括出发时间、到达时间、行驶路径、车道使用、位置、速度、车辆类型等几点；
Mobility model generator for vehicular networks, MOVE	MOVE 是一种基于 JAVA 的开源交通仿真器，位于 SUMO 的顶层。利用 MOVE 仿真需要运动模型 (mobility model, 类似于 SUMO 的.net.xml) 和交通模型 (traffic model, 类似于 SUMO 的.rou.xml) 两部分。与 SUMO 相比，MOVE 具有更好的封装性，可避免脚本编写和内部细节学习的困扰；

表 12.1 traffic simulator (续)

VanetMobiSim	VanetMobiSim 是 CanuMobiSim 仿真器的扩展，基于 JAVA 且支持离散事件的宏微观仿真。宏观仿真主要依靠 TRIGER (topologically integrated geographic encoding and referencing, 拓扑集成的地理编码和引用) 实现地理特征的定义，如公路、铁路、河流、湖泊等等；微观仿真主要支持含交叉口管理的智能驾驶模型 (Intelligence Driving Model with Intersection Management, IDM/IM)、含车道变化的智能驾驶模型 (Intelligence Driving Model with Lane Changing, IDM/LC) 和超车模型 (Overtaking Model) 等；
FreeSim	FreeSim 是一个开源的宏微观交通仿真器。在仿真时，每辆独立车辆会将各自当前的速度和位置发送至中央服务器，而中央服务器基于最短路径和最快速度算法向车辆更新路径。FreeSim 内嵌有六种常用的最短路径算法；
Street Random Way Points, STRAW	STRAW 是基于 JAVA 的开源离散事件仿真器。STRAW 中运动模型定义了如下四部分：段间 (inter segment)、段内 (intra segment)、路线管理 (route management)、执行管理 (execution management)。其中段内运动模型定义了车辆的跟驰和换道情况，段间运动模型则定义了不同路段间（如交叉口）的车辆行为，路线管理和执行管理则由出行 OD 决定。

表 12.2 network simulator

Network simulator, NS-2	NS-2 是一个基于 C++ 的开源的事件驱动网络模拟器，支持在所有网络上实现 TCP、路由、组播路由等的模拟；
Network simulator version 3, NS-3	NS-3 是 NS-2 的改进版本，为解决 NS-2 可伸缩性 (Scalability) 较差的缺点，这一改进主要是通过数据结构的调整实现的。经过改进后，一个简单的数列结构被扩展为四叉树 (quad tree) 结构，随着距离的增加，信号的强度将减弱，但四叉树结构可以保持一定范围的信号的强度，从而降低丢包率 (Packet Loss Rate) 和拥挤效应 (Congestion Effect) 并提升可扩展性；
Objective modular network testbed, OMNet++	OMNet++ 是基于 C++ 环境的开源离散事件网络模拟器。其网络拓扑结构由 NED 语言定义，具体地是实现简单模块 (simple module)、复合模块 (compound module) 和网络的定义。复合模块由简单模块组成，而简单模块的功能由 C++ 实现；
JAVA in simulation time, JiST	JiST 是基于 JAVA 的开源离散事件网络模拟器。JiST 结构有四个基本部分：编译器、虚拟机、重写器 (rewriter, 一个动态类加载器) 和仿真内核 (simulation run time kernel)；
Optimized network engineering tool, OPNet	OPNet 是基于 C 和 C++ 的商用离散事件网络模拟器。OPNet 最初是用于构造有线网络，如今也能用于无线网络的构造。在建模的过程中主要包括三个层次：网络域 (network domain)、节点域 (node domain) 和进程域 (processor domain)。

12.1.3 英汉互译

English	Chinese	English	Chinese	English	Chinese
ad-hoc	点对点、自组织	impose	提出	constrain	约束 (v)
trajectory	轨迹、弹道	partition	分隔 (n, v)	indispensable	不可或缺的
protocol	协议	throughput	吞吐量	latency	延迟
prominent	重要的	integration	整合	microscopic	微小的、微观的
hassle	困难	entity	实体	scalability	可拓展性
instantiate	例示 (v)	concrete	具体的、混凝土	satellite	卫星

12.2 Effect of information availability on stability of traffic flow: Percolation theory approach (TRB, 2018)

信息可达性对交通流稳定性的影响：渗流理论方法

ABSTRACT

网联化和自动化有助于提升交通运输系统的效率与安全性。从操作的层面考虑，确保交通流的稳定性是重中之重，因为不稳定的交通流会形成交通波并可能引发事故。目前已有部分研究讨论了信息可达性对交通流稳定性的影响，但这类研

究多只考虑自动驾驶的情况，而未考虑由普通车、网联车和自动驾驶车组成的混合交通环境。在这一环境下，通信范围与网联车密度将影响互联的效果，进一步影响交通流稳定性。为此，本文基于连续渗流理论（continuum percolation theory）研究上述影响。**连续渗流理论指出，存在一临界密度，当网联车密度超过此临界密度时，即可认为系统是互联的。**而这一密度又与通信范围有关。进一步地，本文还研究了不同通信范围和不同网联车、自动驾驶车密度情况下的交通流稳定性。**结果显示随着通信范围的增加，系统稳定性随之增加，并逐渐接近完全互联的系统。另外还可看出存在一最优通信范围可使得稳定性达到最大并保证信息的有效传输。**

12.2.1 Introduction

1. 在车联网 (V2V, V2I) 环境下，车辆可接收来自周围车辆和交通指挥中心 (traffic management center, TMC) 的信息，有利于驾驶员和自动驾驶车执行更安全高效的动作，而这又会改变原有的交通动力学，形成更利于通信的交通模式。目前对交通动力学的这一改变尚不了解，而这又是最大化车联网效果的前提；
2. 已有的研究多假设交通流为完全连接，而现实中因为信号干扰和物理障碍的影响，完全连接不一定能实现。除了不完全连接，密度和通信范围同样会影响车联网的效果。在现实环境下，单跳通信（即两车间直接通信）不一定能实现，而且也不同保证传播所有的相关信息。为确保所有相关车辆均能受到信息，往往需要多跳通信，这就要求每一辆车的通信范围内存在足够的车辆。如果无法实现有效通信，车辆的行为将会退化为普通的交通行为；
3. 信息于车联网间的传播与流体于多孔介质中的渗流、疾病于人群中的传播、信息于互联网中的传播相似。而后的运动机理已得到大量研究，**渗流理论¹**是其中最可靠、最准确的模型；
4. 本文给出了临界车辆密度的理论推导，高于此密度则认为交通环境为连通的，并基于渗流理论研究车辆密度和通信范围对交通流稳定性的影响。本文包含以下几点贡献：
 - 给出了临界车辆密度的理论推导；
 - 基于渗流理论研究车队稳定性，其中车队稳定性表示为通信范围和交通情况的函数；
 - 研究混合交通车队的稳定性时内生地考虑了普通车、网联车和自动驾驶车的占用率。

12.2.2 Background

1. 传统的渗流理论为离散渗流理论，研究临界概率。讨论一由无限多节点形成的点集，当点集内任意相邻两节点连通的概率大于临界概率时，则认为这一点集内存在一无限大的互连网络；
2. 连续渗流理论与前者类似，同样是研究临界密度。连续渗流理论分为布尔模型 (boolean model, 或 Poisson blob model) 和随机连接模型 (random connection model) 两类。布尔模型认为无限大点集中的每个点均为一半径为 R 的圆的圆心，这些圆将整个空间分为内外两部分，当任意两点同处于内部或外部时，则认为两点相连。而随机连接模型中任意两点间的连通性由一值域为 $[0, 1]$ 的函数描述，这一函数往往被设为一随两点间距离增大而减小的函数。在本文中，采用的连续渗透模型为布尔模型；
3. 连续渗流理论被广泛应用于无线通信网络连通性的研究中。而与一般的无线通信场景不同，车载自组网中的节点——车辆——具有很强的运动性，从而不断改变网络的拓扑结构和连接情况。因此理解车载自组网的连接效果对研究互联对交通流动力学的影响至关重要；
4. 目前已有研究讨论了互联化和自动化对交通流稳定性的影响，这些研究大多在自适应巡航控制 (adaptive cruise control, ACC) 或协同自适应巡航控制 (cooperative adaptive cruise control, CACC) 系统下进行。例如研究分散 CACC 控制逻辑下实现交通流稳定性的充要条件；也有研究指出自动化对提升交通流稳定性效果优于互联化。而较少研究关注信息可达性对交通流稳定性的影响，也较少关注通信对于混合交通编队稳定性的影响；
5. 本研究基于渗流理论对信息可达性进行建模，并研究通信范围和网联车密度对不同组成的混合交通流稳定性的影响。

¹ 渗流理论属于随机图理论。这一理论的出发点是：假设将一多孔的石头浸入水中，水必然由石头中的孔隙渗透，石头中的孔隙可能是连通的，也可能为封闭的，很显然水只能沿其中连通的孔隙渗透，问题在于石头中的孔隙网络的某一点是否会被水湿润？由直觉可知，如果石头中孔隙的连通性越强，则该网络中某一点被水湿润的概率也就越大。用概率 p 表示网络某点的连通性，即认为网络中任意点有 p 的概率是通的，有 $1 - p$ 的概率是闭的，则显然 p 越大网络的连通性越大，且渗流理论认为：存在一临界概率 p_c ，当 $p > p_c$ 时，即可认为网络是连通的，反之则认为网络是由一系列孤立的节点组成。

12.2.3 Definitions

本文假设网联车于公路上的分布服从二维齐次泊松点过程(two-dimensional homogeneous Poisson point process), 并记密度为 λ 。

1. 交通流稳定性: 跟驰模型的一般表达式可以表示为下式

$$\dot{v}_n = f(s_n, \Delta v_n, v_n)$$

式中 $s_n, \Delta v_n$ 分别表示前车 $n - 1$ 与后车 n 间的距离和速度差。实验观测结果指出, 速度与间距之间存在一平衡关系, 即假设速度和间距间存在一表达式 $v = V(s)$, 则存在一平衡间距 s^* 与对应的平衡速度 $v^* = V(s^*)$, 满足公式 $f(s^*, 0, v^*) = 0$ 。如果车队中某一车辆偏移平衡间距, 则引起的扰动会向上游传播。当交通流处于稳定条件下, 扰动在传播的过程中会逐渐衰减;

2. 齐次泊松点过程: 记一长为 L 的公路路段上的车辆数为 $X_\lambda(L)$ 。假设公路宽度恒为 S , 并假设 $X_\lambda(L)$ 服从泊松分布, 则该路段上存在 k 辆车的概率可以表示为

$$P(X_\lambda(L) = k) = \frac{(\lambda S L)^k}{k!} e^{-\lambda S L}$$

3. 传输范围: 记车辆 i 的位置为 x_i , 则其传输范围为一半径为 R_i 的圆形区域, 且本研究中假设所有网联车的传输距离相等;
4. 车间通信: 定义两车辆 i, j 间的通信函数 $h(x_i, x_j)$

$$h(x_i, x_j) = \begin{cases} 1 & |x_i - x_j| < \min\{R_i, R_j\} \\ 0 & |x_i - x_j| \geq \min\{R_i, R_j\} \end{cases}$$

5. 通信路径: 一系列互相通信的车辆前后相接组成通信路径;
6. k 连通域(**connected k -component**): 如果存在一通信路径, 且该路径不是其它路径的子集, 则称该路径为连通域, 记为 CC_k , 下标 k 表示该连通域由 k 辆车组成;
7. 连通域的距离: 本研究认为两个连通域间的最小距离为连通域的距离;
8. 车载自组网连续渗流理论(布尔模型): 车载自组网的连续渗流理论采用布尔模型, 且与一般的布尔模型类似, 区别仅在于模型中的距离由 $|x_i - x_j|$ 表示。根据连续渗流理论, 存在一临界密度 λ_c , 使得超过这一密度时, 即存在一包含无限节点的连通域。

12.2.4 Percolation of vehicular ad-hoc networks

本节将介绍特征密度 λ_c 的推导, 当超过这一密度时, 场景内的所有网联车即可顺畅地实现信息交互并辅助驾驶员驾驶。

connected components: length estimation

假设(已建立通信的)网联车服从参数为 λ 的泊松分布, 因默认此处的网联车已建立通信, 则只需 k 辆这样的网联车即可形成一 k 连通域。因假设道路宽度相等则每一连通域仅存在长度和方向两个特征。又因为模型中的距离只考虑纵向, 则每一连通域可由一矩形表示。则在长为 L 的路段上有且只有一个 k 连通域的概率 $P(N_L = k)$ 为

$$P(N_L = k) = \frac{\frac{[\lambda(L+2R)]^k}{k!} e^{-\lambda(L+2R)}}{e^{-\lambda R} \cdot e^{-\lambda R}} = \frac{[\lambda(L + 2R)]^k}{k!} e^{-\lambda L}$$

要求长为 L 的路段上有且只有一个 k 连通域意味着不仅在该路段上存在 k 辆网联车, 还要求路段上下游各长为 R 的区间内无网联车。

connected component: critical density

1. 假设网联车的分布服从参数为 λ 的泊松分布, 则 k 连通域也会相应服从参数为 $\lambda(k)$ 的泊松分布, 则对应有临界密度 $\lambda_c, \lambda_c(k)$ 。以下两节将给出两者的关系式;

2. 已知 k 连通域会服从参数为 $\lambda(k) = \lambda_c(k)$ 的泊松分布，则易知，两 k 连通域之间的期望距离 d_{avg} 为

$$d_{avg} = \frac{1}{\lambda_c(k)}$$

3. 又因为当 $\lambda = \lambda_c$ 时意味着存在一无限大的连通域，即两个 k 连通域之间的距离 $Y \leq R$ ，对应的概率

$$P(Y \leq R) = 1 - e^{-\lambda_c R}$$

对应的有概率密度函数 $f(Y|Y \leq R) = \lambda_c e^{-\lambda_c R}$ ，从而同样可计算两者的期望距离 d_{avg}

$$d_{avg} = E(Y|Y \leq R) = \int_0^R t \times f(Y|Y \leq t) dt = \int_0^R t \lambda_c e^{-\lambda_c R} dt = \frac{1}{\lambda_c} - \left(R + \frac{1}{\lambda_c} \right) e^{-\lambda_c R}$$

4. 联立以上两式，即可得到

$$\frac{1}{\lambda_c(k)} = \frac{1}{\lambda_c} - \left(R + \frac{1}{\lambda_c} \right) e^{-\lambda_c R} \iff \lambda_c(k) = \frac{\lambda_c}{1 - (\lambda_c R + 1)e^{-\lambda_c R}}$$

connected component: critical length

1. 渗流理论指出，当达到临界密度时，任意非空的长为 R 的区域内必然存在一 k 连通域；
2. 记路段内网联车数量为 N ，有 $N = \lambda L$ ，对应的也存在 $\bar{N}_k = \lambda(k)L$ ， \bar{N}_k 为路段内 k 连通域的平均个数，故存在

$$\lambda(k) = \lambda \frac{\bar{N}_k}{N}$$

3. 注意到 $\frac{\bar{N}_k}{N}$ 可理解为在长为 R 的区域内存在一 k 连通域的概率，即

$$\frac{\bar{N}_k}{N} = \frac{[\lambda(3R)]^k}{k!} e^{-\lambda L} \implies \lambda_c(k) = \lambda_c \frac{[\lambda_c(3R)]^k}{k!} e^{-\lambda_c R}$$

4. 将上式与上一节所提结论结合，有

$$f(\lambda_c, R, k) = \left[1 - (\lambda_c R + 1)e^{-\lambda_c R} \right] \times \left[\frac{[\lambda_c(3R)]^k}{k!} e^{-\lambda_c R} \right] - 1 = 0$$

上式为稳定性分析计算中考虑连通性水平奠定了基础，即当上式 $f(\lambda_c, R, k) = 0$ 成立时，意味着渗流现象恰好发生。令 $A_c(R) = 1 - e^{-\lambda_c R}$ ，再令 $\mu = -\log[1 - A_c(R)]$ ，可将上式进一步简化得

$$f(\mu, k) = \left[1 - (\mu + 1)e^{-\mu} \right] \times \left[\frac{[3\mu]^k}{k!} e^{-\mu} \right] - 1 = 0$$

5. 经过计算可知，对任意 $A(R) \geq 0.785$ 即可满足 $f(A(R), k) \geq 0$ ， $k \geq 3$ ，则认为网络完全连通，此时 λ 与 R 的具体取值对交通流稳定性并无影响，而当 $A(R) < 0.785$ 时，则必然存在一部分网联车无法收到信息，其行为将表现为普通车；
6. 另外同样由上式可知， R 越大，所对应的 λ_c 也就越小，意味着理论上通信范围越大，使得网络完全连通所需的网联车密度也就越小。然而已有研究指出，通信范围越大，信息传输时的丢包率也越大，时延也越久，特别是在传输距离大于 120-140m 时，上述影响较为显著。因此本研究设定通信范围为 130m，尽可能避免信息丢失和时延的影响，并就通信范围对交通流稳定性的影响进行敏感性分析。

12.2.5 Analytical investigation of string stability

1. 网联车和自动驾驶车的存在将对原有的交通动力学造成影响，进而影响由常规车、网联车和自动驾驶车组成的混合交通流的稳定性。而且信息的可达性同样会对上述稳定性造成影响，因为当网联车和自动驾驶车无法接收到信息时，其行为均会发生响应的退化；
2. 本文的主要目标即是将渗流理论与交通流稳定性分析结合，分析不同连接水平对交通流稳定性的影响。本节将对不同组成的混合交通流稳定性进行分析；

3. 假设一无限长车队中某车的车头间距发生微小扰动 \bar{s}_l 得 $s_l = s^* + \bar{s}_l$, 对应地其速度也发生扰动 $v_l = V(s^*) + \bar{v}_l$, 则研究指出在满足下式时该扰动即会引起失稳

$$\sum_n \left[\frac{f_v^{n2}}{2} - f_{\Delta v}^n - f_s^n \right] \left[\prod_{m \neq n} f_s^m \right]^2 < 0 \quad (12.1)$$

上式中 n 表示不同类型的车辆, 系数分别为

$$f_s^n = \frac{\partial f(s_n, \Delta v_n, v_n)}{\partial s_n} \Big|_{(s^*, 0, V(s^*))} \quad f_{\Delta v}^n = \frac{\partial f(s_n, \Delta v_n, v_n)}{\partial \Delta v_n} \Big|_{(s^*, 0, V(s^*))} \quad f_v^n = \frac{\partial f(s_n, \Delta v_n, v_n)}{\partial \Delta s_v} \Big|_{(s^*, 0, V(s^*))}$$

acceleration model formulation

1. 本文的主要目的在于分析网联车和自动驾驶车对交通流稳定性的影响而非提出新的动力学模型, 因此本文采用目前最前沿的动力学模型对车辆的行为进行建模:

- **普通车:** 普通车的行为具有随机性, 可能因为驾驶员的误操作而引发事故, 因为当驾驶员不确定前车的行为时, 可能采取冒险性的行为, 故动力学模型中没有必要进行人为限制以避免事故发生。Kahneman 与 Tversky 提出了评价驾驶行为的效益和风险的预期理论即可描述上述现象;
- **网联车:** 车联网技术为网联车提供周围交通情况的确切信息, 从而使得车辆的行为具有更高的确定性。智能驾驶员模型 (**intelligent driver model**, IDM) 是一确定的、无事故的模型, 本文用其描述车联网环境下的车辆行为;
- **自动驾驶车:** 自动驾驶车的行为同样具有很强的确定性, 其行为基于车载传感器所采集的信息。本研究模拟了这种信息采集行为, 并采用另一类确定性跟驰模型——最优速度模型 (**optimal-velocity model**, OV 模型) 描述其跟驰行为。

以下将对上述三类动力学模型进行详细介绍;

2. 普通车的行为基于驾驶员对周围交通环境的感知, 从而具有较强的随机性, 驾驶员会尽量避免事故的发生, 但当感知有误时事故依然可能发生。Kahneman 与 Tversky 提出了评价驾驶行为的效益和风险的预期理论², Hamdar 等基于此提出了一可避免大部分事故的具有随机性的跟驰模型³。本研究将基于此对普通车行为进行建模, 则对于第 n 辆车的加速度 a_n 的值函数 $U_{PT}(a_n)$ 为

$$U_{PT}(a_n) = \frac{w_m + (1 - w_m) \left(\tanh \frac{a_n}{a_0} + 1 \right)}{2} \left[\frac{\frac{a_n}{a_0}}{(1 + \frac{a_n}{a_0})^2} \right]^\lambda$$

式中 a_0 仅为一用于标准化的参数, 可令其为 $1m/s^2$; w_m 和 $\lambda > 0$ 为超参数, 前者为负效益的非对称系数 (可设为 2), 后者为标准化效益的敏感性指数 (可设为 0.5)。基于加速度值函数, 即可进一步得到加速度的效用函数 $U(a_n)$

$$U(a_n) = (1 - p_{n,i})U_{PT}(a_n) + p_{n,i}w_c k(v, \Delta v) \quad a_n > 0$$

式中 $p_{n,i}$ 为第 $i - 1$ 次加速无碰撞的前提下第 n 辆车与前车追尾的概率; w_c 为事故的权重因子 (可设为 40); $k(v, \Delta v)$ 用于衡量事故的严重程度。最后, 为了描述驾驶员行为的随机性, 认为 a_n 服从概率分布函数为 $f(a_n)$ 的分布 (logistic 函数), 具体的取值由该分布得到

$$f(a_n) = \begin{cases} \frac{e^{\beta_{PT} U(a_n)}}{\int_{a_{\min}}^{a_{\max}} e^{\beta_{PT} U(a')} da'} & a_{\min} < a_n < a_{\max} \\ 0 & \text{else} \end{cases}$$

式中 β_{PT} 表示对于效用 $U(a_n)$ 的敏感性 (可设为 3), 并记 $a_{\min} = -8m/s^2$, $a_{\max} = 4m/s^2$ 。在这一模型下,

²Daniel Kahneman and Amos Tversky. Prospect Theory: An Analysis of Decision under Risk[J]. *Econometrica*, 47(2):263-292.

³Hamdar S , Treiber M , Mahmassani H , et al. Modeling Driver Behavior as Sequential Risk-Taking Task[J]. *Transportation Research Record: Journal of the Transportation Research Board*, 2008, 2088:208-217.

式 12.1 中的三项偏导数计算为

$$\begin{cases} f_s = \frac{2}{\tau_{\max}^2} \\ f_{\Delta v} = \frac{-2}{\tau_{\max}} \\ f_v = \frac{2\alpha z_e^*}{\tau_{\max}} + \frac{2\alpha v_e}{\tau_{\max}} \frac{1}{\sqrt{2}v_e} \sqrt{\ln \frac{w_c \tau_{\max}}{2\sqrt{2\pi}\alpha v_e}} \quad z_e^* = \sqrt{2 \ln \frac{w_c \tau_{\max}}{2\sqrt{2\pi}\alpha v_e}} \end{cases}$$

上式 τ_{\max} 为最大预期时间范围 (可设为 5s); α 为速度变异系数 (可设为 0.1); v_e 为均衡速度;

3. 当网联车处于联网状态下时, 车辆可得到更充分准确的交通信息, 驾驶行为也就具有更高的确定性。本文以 IDM 模型⁴描述联网状态下的网联车的行为, 这一模型在描述拥堵动态时较其它确定性模型具有更高的准确性。IDM 模型认为后者的加速度为一与当前速度、前后车速度差、当前间距与理想间距比值有关的连续函数, 并考虑其它感知参数如理想加速度、理想间距、合适减速度等

$$a_{IDM}^n(s_n, v_n, \Delta v_n) = \bar{a}_n \left[1 - \left(\frac{v_n}{v_0^n} \right)^{\delta_n} - \left(\frac{s^*(v_n, \Delta v_n)}{s_n} \right)^2 \right] \quad s^*(v_n, \Delta v_n) = s_0^n + T_n v_n + \frac{v_n \Delta v_n}{2 \sqrt{\bar{a}_n \bar{b}_n}}$$

式中 $\delta_n, T_n, \bar{a}_n, \bar{b}_n, s_0^n, v_0^n$ 分别表示与自由加速有关的幂 (4.0)、理想时距 (4.5s)、最大加速度 ($1.4m/s^2$)、理想减速度 ($-2.0m/s^2$)、拥堵长度和理想速度。在这一模型下, 式 12.1 中的三项偏导数计算为

$$\begin{cases} f_s^n = \frac{2\bar{a}_n}{s_e} \left(\frac{s_0 + T v_e}{s_e} \right)^2 \\ f_{\Delta v}^n = \frac{-v_e}{s_e} \sqrt{\frac{\bar{a}}{\bar{b}}} \left(\frac{s_0 + T v_e}{s_e} \right) \\ f_v^n = \frac{-\bar{a}\delta}{v_0} \left(\frac{v}{v_0} \right)^{\delta-1} - \sqrt{\frac{\bar{a}}{\bar{b}}} \left(\frac{s_0 + T v_e}{s_e} \right) \frac{2T \sqrt{\bar{a}\bar{b}}}{s_e} \end{cases}$$

式中 v_e 和 s_e 分别表示平衡速度和平衡间距, 且有 $\Delta v_e = 0$ 。需要注意的是, 上述模型仅针对处于车联网模式下的网联车, 当网联车无法实现有效车车通信时, 其行为将退化为普通车的行为;

4. 自动驾驶车由摄像头获取周围的交通信息, 从而实现确定性、安全性更高的驾驶行为。自动驾驶车的反应时间仅受限于传感过程中的延时。本文对 Van Arem 等⁵和 Reece 等⁶所提模型进行改进以描述上述驾驶行为。另外本文也对摄像头进行仿真, 探测距离为 $90m \pm 2.5\%$ 、水平视野 $\pm 35^\circ$, 布置多个以实现 360° 全覆盖, 摄像头每 $35ms$ 进行一次更新, 一次可同时捕捉 64 个目标。需要注意的是, 自动驾驶车无法感知摄像头探测范围以外的环境, 也就无法针对其做出反应。下式给出了自动驾驶车最大安全速度的计算公式

$$v_{\max} = \sqrt{-2a_i^{decc} \Delta x} \quad \Delta x = \min \left\{ (x_{i-1} - x_i - l_{i-1}) + v_i \tau + \frac{v_{i-1}^2}{2a_{i-1}^{decc}}, \text{ Sensor Detection Range } \right\}$$

式中 i 和 $i-1$ 分别为自动驾驶车及其前车; $x_i, l_i, v_i, a_i^{decc}$ 分别指车辆 i 的位置、长度、速度和最大减速度; τ 为车辆 i 的反应时间。进而可由下式描述其跟驰模型

$$a_i^d(t) = k_a a_{i-1}(t - \tau) + k_v(v_{i-1}(t - \tau) - v_i(t - \tau)) + k_d(s_i(t - \tau) - s_{ref})$$

式中 a_i^d 为车辆 i 的加速度; 模型参数 k_a, k_v, k_d 分别推荐为 1.0、0.58 和 0.1; s_{ref} 为最小间距 $s_{\min} (= 2m)$ 、考虑反应时间的跟驰距离 s_{system} 和安全跟驰距离 s_{safe} 的最小值, 后两个参数的计算式如下

$$s_{system} = \frac{v_{i-1}^2}{2} \left(\frac{1}{a_i^{decc}} - \frac{1}{a_{i-1}^{decc}} \right) \quad s_{safe} = v_i \tau$$

⁴Kesting A , Treiber M , Helbing D . Enhanced Intelligent Driver Model to Access the Impact of Driving Strategies on Traffic Capacity[J]. Philos Trans A Math Phys Eng Sci, 2009, 368(1928):4585-4605.

⁵Arem B V , Driels C J G V , Visser R . The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics[J]. IEEE Transactions on Intelligent Transportation Systems, 2007, 7(4):429-436.

⁶Reece D A , Shafer S A . A computational model of driving for autonomous vehicles[J]. Transportation Research. Part A: Policy and Practice, 1993, 27(1):23-50.

而后可得到最终的车辆 i 加速度 a_i

$$a_i(t) = \min\{a_i^d(t), k(v_{\max} - v_i(t))\} \quad k = 1.0$$

在这一模型下，假设在计算时前车的加速度为 0，则式 12.1 中的三项偏导数可简化为

$$f_s = k_d \quad f_{\Delta v} = k_v \quad f_v = -k_d \tau$$

12.2.6 Results and discussion

- 本节研究不同组成的混合车队的稳定性，并重点关注通信对车队稳定性的影响，为此对普通车跟驰模型参数进行刻意调整使其形成高度不稳定的交通流。本节对稳定性研究基于以上的理论推导而非仿真；
- 仿真比较了在不同网联车和自动驾驶车占比以及不同通信范围的情况下车流稳定性，稳定性以临界速度表征，即使得交通流保持稳定（式 12.1 非负）的最大平衡速度 v_e ，研究的主要结果如下：
 - 在自动驾驶车比例较低的情况下，网联车比例的增加可有效提升交通流稳定性；
 - 在网联车比例较低的情况下，自动驾驶车比例的增加对交通流稳定性的提升效果较低；
 - 随着传输距离的增加，交通流稳定性逐渐提升（但总体而言变化不明显），且在传输距离达到 130m 时其效果近似于网联车完全联网的效果。

12.2.7 英汉互译

English	Chinese	English	Chinese	English	Chinese
percolation	渗流、渗透 (n)	continuum	连续体	interference	干扰 (n)
derivation	起源、推导 (n)	endogenous	内生的	vertices	顶点 (复)
euclidean	欧几里得的	recede	逐渐远去 (v)	decentralize	分散 (v)
equilibrium	均衡 (n)	perturbation	扰动 (n)	decay	衰退 (n,v)
deviate	偏离 (v)	impose	推行、迫使 (v)	lemma	引理 (n)
linearize	线性化 (v)	exponent	幂 (n)	anticipation	预期 (n)
perceptive	感知的	remainder	剩余物		

第 13 章

基础设施仿真

13.1 Electric vehicle charging station diffusion: An agent-based evolutionary game model in complex networks (Energy, 2022)

ABSTRACT

充电设施与电动车之间的双向因果关系导致复杂网络环境下两者存在复杂且尚不清晰的演化交互机理，从而影响了对于充电站设施的投资。为此，本文提出了一种新型的基于智能体的演化博弈模型 (evolutionary game model)，模型在建模充电站扩散机理时考虑了消费者的微观行为的影响。在真实场景下仿真电动车与电动车充电站的扩散演化，并研究复杂网络拓扑、碳税、电价等因素的影响。

13.1.1 Introduction & Literature review

1. 电动车充电站的布设可产生多项长期效益，包括引导交通布局、创造经济效益、诱发电动车购车需求等；
2. 尽管电动车充电站至关重要，世界范围内的公共充电站的发展普遍落后于电动车的发展。以中国为例，2020 年电动车与充电桩的比例 (vehicle-to-pile ratio) 仅为 3.3:1，远低于国际要求的 1.5:1 和我国目标的 1.2:1。公共充电站发展的滞后导致了充电难题：
 - 早期的研究多聚焦于对充电站投资的决定因素的经验调研，较少关注充电站的扩散演化过程。为此，后续研究主要基于微分方程、**演化博弈论 (evolutionary game theory)** 等方法建模充电站规模的动态发展；
 - 在研究充电站市场发展的过程中，学者们广泛调研政府参与的推动作用，但未考虑到充电站推广扩散是多因素共同交互的结果。**复杂网络**是一种可建模多因素相互关系的数学模型；
 - 近年来结合复杂网络理论的演化博弈模型得到学界的关注，模型可建模多因素交互下的博弈行为。但在研究充电站扩散机制时往往只考虑小世界网络 (**small-world network**) 或无标度网络 (**scale-free network**)，未考虑不同拓扑特征网络的影响。研究不同网络拓扑结构下充电站的扩散机制有助于指导现实布局以推动相关行业发展；
 - 与此同时，大量研究者在研究充电站扩散时未考虑需求侧的变化，使得模型无法捕捉宏观市场发展与微观个体消费之间的关系。忽略需求侧动态变化背后的假设是消费行为频繁且稳定地发生，该假设一般适用于快速消耗品 (**fast-consuming goods**)，而购车行为显然不属于此类。**智能体模型 (agent-based model, ABM)**是自下而上建立微观与宏观环境间关系的强大工具；
 - 已有学者从需求侧分析消费者选购电动车的偏好因素，但较少考虑相关偏好对充电站扩散的影响。
3. 针对已有研究缺陷，**本研究首次提出了一种基于智能体的演化博弈模型以推动充电站的部署**。所提模型考虑了社交网络的影响，包含供给侧和需求侧演化的动力模型：
 - 在需求侧基于智能体模型建模了消费者在给定汽车使用寿命下的周期性购车行为。其购车选择（燃

油车或电动车)既受供给侧的影响,也与其自身偏好和亲友等重要社会关系的选择有关;

- 在供给侧基于演化博弈模型建模了能源企业投资充电站的规则。

建模供需两端的动态特征不仅有助于研究碳税、补贴、需求偏好等因素对公共充电站投资建设的影响,也有助于理解充电站对供需两端的微观影响机制;

5. 因为 WS 小世界网络(见第 18.8 节)最接近现实的大部分网络,故主要以其为场景构建演化博弈模型,并考虑 BA 无标度网络(见第 18.9 节)、最近邻耦合网络(nearest-neighbor coupled network)¹、ER 随机网络(见第 18.7 节)等三种网络结构的影响;

6. 综上,本研究主要聚焦以下问题:

- 考虑动态需求演化对充电站扩散模型结果的影响;
- 网络拓扑结构对充电站及电动车扩张的影响以及最合适的网络结构;
- 如何促使电动车市场从插入式混合电动车(plug-in hybrid electric vehicle, PHEV)转向纯电动车(battery electric vehicle, BEV)。

在此基础上,研究的创新点主要如下:

- 提出了一种基于智能体的演化博弈模型,同时考虑了供给侧对充电站的投资与需求侧电动车的扩张;
- 建立了消费者微观消费行为与充电站投资发展间的联系;
- 挖掘了复杂网络拓扑结构对充电站扩散的影响。

13.1.2 Method

1. 考虑两类智能体——消费者与能源站。在需求侧,消费者根据自身偏好及宏观经济因素(油价、电价、充电设施配套等)选择购买常规燃油车、油电混动车和纯电动车等;在供给侧,能源企业基于消费者需求、友商决策及政策约束等因素投资电动车充电设施。供需两侧智能体交互决策,构成非线性复杂动态系统,故构建基于智能体的演化博弈模型仿真电动车充电站的动态扩张;

2. 模型假设消费者为异质、有限理性(bounded rational),并介绍消费者的决策逻辑如下:

- 当纯电动车花费低于传统燃油车且周围存在电动车充电站,或者消费者为环保主义者、技术主义者时,将购买纯电动车;
- 当纯电动车花费低于传统燃油车但周围不存在电动车充电站时,将基于整体效益选择购买纯电动车或油电混动车;
- 当传统燃油车花费更低,且消费者不是环保主义者或技术主义者时,将基于整体效益选择购买油电混动车或传统燃油车。

参考已有研究的调查,假设消费者中技术主义者的比例为 2.5%,环保主义者的比例为 16%。为实现上述决策流程,需解决两个关键问题——纯电动车经济效益的量化与基于整体效益的决策流程;

3. 首先介绍纯电动车经济效益 ce 的量化方法

$$ce = pre_{BEV} - sav_{BEV} \cdot yer_{BEV}, \quad pre_{BEV} = p_{BEV} - p_{CV}, \quad sav_{BEV} = (p_{oil} \cdot m_{CV} - p_e \cdot m_{BEV}) \cdot VKT$$

上式中 pre_{BEV} 表示纯电动车售价 p_{BEV} 与燃油车售价 p_{CV} 的差额; yer_{BEV} 表示纯电动车的使用寿命; sav_{BEV} 表示使用纯电动车相对于燃油车每年可节省的成本,由单位油价 p_{oil} 、单位电价 p_e 、燃油车公里油耗 m_{CV} 、电动车公里电耗 m_{BEV} 、和全年车公里数 VKT 决定。考虑到随着技术的成熟短期内纯电动车的成本具有下降的趋势,基于成本加成定价法(cost-plus pricing approach)确定电动车售价 p_{BEV}

$$p_{BEV} = (1 + \mu)c_{BEV}, \quad c_{BEV} = c_{BEV}^0 Q^{-\beta}$$

上式中 μ 为加成率; c_{BEV} 表示电动车的单位生产成本,由电动车的初始单位生产成本 c_{BEV}^0 、电动车的累计销量 Q 、以及参数 $\beta > 0$ 决定。确定 c_{BEV} 的表达式被称为技术学习曲线(technique learning curve),反映了产品累计产量提升与技术成熟对单位生产成本的影响(设 $\beta = 0.18$);

¹一种规则的复杂网络模型,网络中每一节点与距其最近的若干节点相连。

4. 进一步介绍基于整体效益的不同车辆类型决策算法。作者提出基于模糊数学理论（详见第 23.15 节）与 TOPSIS 算法（详见第 23.14 节）的消费者购车选择算法：

- 设计调查问卷搜集数据。问卷主要调查两类问题——一方面是车辆各维度性能（费用、维护成本、安全性、技术先进性、动力性能、噪声和碳排放）对购车者购车决策的影响权重；另一方面是亲友购车决策对自身的影响。问卷采用李克特 7 级量表 (7-point Likert scale)，每一等级对应一三角模糊数 (triangular fuzzy number)²，从而得到考虑了模糊效应的消费者偏好数据；
- 考虑燃油车、油电混动车、和纯电动车三类车辆，通过专家评分得到三类车辆各维度性能（费用、维护成本、安全性、技术先进性、动力性能、噪声和碳排放）的评价。评价时同样基于李克特 7 级量表，评价结果 $e_{ij} = (e_{ij}^1, e_{ij}^2, e_{ij}^3)$ 同样为三角模糊数，其中下标 i 对应车辆类型， j 对应评价维度；
- 融合专家对三类车辆各维度性能的评价 e_{ij} 和每个消费者 k 对车辆各维度性能的偏好 $w_{jk} = (w_{jk}^1, w_{jk}^2, w_{jk}^3)$ ，得到每个消费者对三类车辆各维度性能的评分矩阵 $S_k = [x_{ij}^k]$

$$x_{ij}^k = P(e_{ij}) \times P(w_{jk}) = \frac{e_{ij}^1 + 4e_{ij}^2 + e_{ij}^3}{6} \times \frac{w_{jk}^1 + 4w_{jk}^2 + w_{jk}^3}{6}$$

上式中 $P(\cdot)$ 表示梯级平均积分表示法 (graded mean integration representation, GMIR)，是一种经典的模糊数去模糊化得到确切数值的方法（详见第 23.15 节）；

- 得到评价矩阵 S_k 后即可按 TOPSIS 算法综合各维度性能进行方案比选。首先是数据标准化预处理 $\bar{x}_{ij}^k = \frac{x_{ij}^k}{\sqrt{\sum_i (x_{ij}^k)^2}}$ 。但作者考虑了亲友购车偏好对购车者的影响对 \bar{x}_{ij}^k 作进一步标准化

$$\bar{x}_{ij}^k = (1 - \alpha_k) \bar{x}_{ij}^k + \alpha_k \sum_{l \in L_k} \frac{\bar{x}_{ij}^l}{|L_k|}$$

上式中 $\alpha_k \in [0, 1]$ 表示问卷得到的亲友购车偏好对购车者 k 的影响， L_k 表示在小世界网络模型上购车者 k 的邻居集合；

- 数据标准化后后续的方案比选流程与经典 TOPSIS 算法（详见第 23.14 节）完全一致。

5. 在介绍了消费者智能体的决策流程后再介绍能源站智能体在复杂网络中的演化博弈流程。模型假设：

- 博弈模型中包含 N 个能源站智能体，每个智能体决策其是否成为充电站或加油站；
- 相连能源站智能体可交换利润信息，记每个能源站周围有 n_{CS} 个充电站和 n_{GS} 个加油站；
- 三种类型的车辆均匀分布于能源站周边，记每个能源站周边有 ω 辆各型车辆。需要注意的是在加油或充电时油电混动车等价于 0.83 辆燃油车或 0.17 辆纯电动车；
- 能源站的收支按整个寿命周期计算，记 L_{CS}, L_{GS} 分别表示充电站和加油站的使用周期（设为 10 年）。记 C_{cs}, C_{gs} 分别表示投资一个充电站和加油站的总成本， r 表示利率，则考虑资金的时间价值得到投资一个充电站和加油站后在其使用周期内的均摊年度成本 C_{csc}, C_{gsc} 为

$$C_{csc} = C_{cs} \frac{r(1+r)^{L_{cs}}}{(1+r)^{L_{cs}} - 1}, \quad C_{gsc} = C_{gs} \frac{r(1+r)^{L_{gs}}}{(1+r)^{L_{gs}} - 1}$$

同理记 RV_{cs}, RV_{gs} 分别表示投资一个充电站和加油站的残值 (residual value)，则其使用周期内的均摊年度残值 RV_{csc}, RV_{gsc} 为

$$RV_{csc} = RV_{cs} \frac{r(1+r)^{L_{cs}}}{(1+r)^{L_{cs}} - 1}, \quad RV_{gsc} = RV_{gs} \frac{r(1+r)^{L_{gs}}}{(1+r)^{L_{gs}} - 1}$$

6. 在此基础上介绍能源站收益的计算方法。能源站收益包括经营利润和政策补贴两部分，其中经营利润由油价（电价）利润和车辆加油（充电）需求决定。区分以下 12 种基本情况：

²李克特 7 级量表中将受试者对问题的态度分为极差、差、偏差、中立、偏好、好、极好共 7 个等级，分别对应 (0, 0, 0.1), (0, 0.1, 0.3), (0.1, 0.3, 0.5), (0.3, 0.5, 0.7), (0.5, 0.7, 0.9), (0.7, 0.9, 1), (0.9, 1, 1) 共 7 个三角模糊数

- 站点为充电站，相邻均为充电站，消费者全购买纯电动车。此时充电站服务的 ω 辆车每年的总充电需求为 $\omega \cdot m_{BEV} \cdot VKT$ ，则该情况下站点的年收益为

$$case1 = (p_e - p_{ec})\omega \cdot m_{BEV} \cdot VKT + s_{cs} - C_{csc} - C_{cs0} + RV_{csc}$$

式中 p_{ec} 为充电站的单位购电成本； s_{cs} 为建设充电站的年财政补贴； C_{csc} 为运营充电站的年成本；

- 站点为充电站，相邻均为充电站，消费者全购买燃油车，则 $case2 = s_{cs} - C_{csc} - C_{cs0} + RV_{csc}$ ；
- 站点为充电站，相邻均为充电站，消费者全购买油电混动车。此时充电站服务的 ω 辆车每年的总充电需求为 $0.17\omega \cdot m_{BEV} \cdot VKT$ ，则 $case3 = 0.17(p_e - p_{ec})\omega \cdot m_{BEV} \cdot VKT + s_{cs} - C_{csc} - C_{cs0} + RV_{csc}$ ；
- 站点为充电站，相邻均为加油站，消费者全购买纯电动车。此时充电站不仅需要服务周边的 ω 辆电动车，还需服务相邻 n_{GS} 个加油站周边的 $n_{GS}\omega$ 辆电动车，则

$$case4 = (p_e - p_{ec})\omega \cdot m_{BEV} \cdot VKT \cdot (n_{GS} + 1) + s_{cs} - C_{csc} - C_{cs0} + RV_{csc}$$

- 站点为充电站，相邻均为加油站，消费者全购买燃油车，则 $case5 = s_{cs} - C_{csc} - C_{cs0} + RV_{csc}$ ；
- 站点为充电站，相邻均为加油站，消费者全购买油电混动车，则 $case6 = 0.17(p_e - p_{ec})\omega \cdot m_{BEV} \cdot VKT \cdot (n_{GS} + 1) + s_{cs} - C_{csc} - C_{cs0} + RV_{csc}$ ；
- 站点为加油站，相邻均为充电站，消费者全购买纯电动车，则 $case7 = -t_{gs} - C_{gsc} - C_{gs0} + RV_{gsc}$ ，式中 t_{gs} 为建设加油站的年税收； C_{gs0} 为运营加油站的年成本；
- 站点为加油站，相邻均为充电站，消费者全购买燃油车，则

$$case8 = (p_{oil} - p_{oc})\omega \cdot m_{CV} \cdot VKT \cdot (n_{CS} + 1) - t_{gs} - C_{gsc} - C_{gs0} + RV_{gsc}$$

式中 p_{oc} 为加油站的单位购油成本；

- 站点为加油站，相邻均为充电站，消费者全购买油电混动车，则 $case9 = 0.83(p_{oil} - p_{oc})\omega \cdot m_{CV} \cdot VKT \cdot (n_{CS} + 1) - t_{gs} - C_{gsc} - C_{gs0} + RV_{gsc}$ ；
- 站点为加油站，相邻均为加油站，消费者全购买纯电动车，则 $case10 = -t_{gs} - C_{gsc} - C_{gs0} + RV_{gsc}$ ；
- 站点为加油站，相邻均为加油站，消费者全购买燃油车，则 $case11 = (p_{oil} - p_{oc})\omega \cdot m_{CV} \cdot VKT - t_{gs} - C_{gsc} - C_{gs0} + RV_{gsc}$ ；
- 站点为加油站，相邻均为加油站，消费者全购买油电混动车，则 $case12 = 0.17(p_{oil} - p_{oc})\omega \cdot m_{CV} \cdot VKT - t_{gs} - C_{gsc} - C_{gs0} + RV_{gsc}$ 。

所有实际情况均可视为以上 12 种基本情况的线性组合。记 $y_{BEV}, y_{PHEV}, y_{CV}$ 分别表示纯电动车、油电混动车和燃油车的市场占有率，则充电站和加油站的期望收益 U_i^{CS}, U_i^{GS} 可按下式线性插值

$$U_i^{CS} = \begin{bmatrix} \frac{n_{CS}}{n_{CS} + n_{GS}} & \frac{n_{GS}}{n_{CS} + n_{GS}} \end{bmatrix} \begin{bmatrix} case1 & case2 & case3 \\ case4 & case5 & case6 \end{bmatrix} \begin{bmatrix} y_{BEV} \\ y_{PHEV} \\ y_{CV} \end{bmatrix}$$

$$U_i^{GS} = \begin{bmatrix} \frac{n_{CS}}{n_{CS} + n_{GS}} & \frac{n_{GS}}{n_{CS} + n_{GS}} \end{bmatrix} \begin{bmatrix} case7 & case8 & case9 \\ case10 & case11 & case12 \end{bmatrix} \begin{bmatrix} y_{BEV} \\ y_{PHEV} \\ y_{CV} \end{bmatrix}$$

- 在复杂网络环境下，每个能源站基于其收益和相邻站点收益进行演化博弈，按概率更新其自身的站点类型，更新规则如下

$$\tau(i \leftarrow j) = \frac{1}{1 + \exp\{(U_i - U_j)/k\}}$$

上式表明，对于任意站点 j ，从其邻居中随机抽取站点 i 比较两者收益。若 $U_i > U_j$ ，则按概率 τ 更新为站点 i 所属的类型。超参 k 设为 0.1；

- 研究搭建的智能体演化博弈模型以重庆市为场景，以 1:10 的比例设置 20000 辆仿真车智能体，并假设单个充电站服务约 200 辆电动车设置 100 个能源站智能体，模型的其它参数也根据重庆市现状、龙头企业（比亚迪）数据及各类型调研和规范确定。

13.1.3 Case study

1. 以 ANYLOGIC 8.5.2 Professional 作为仿真软件仿真智能体演化博弈模型，每个实验重复 300 次取均值；
2. 在当前的政策环境下，仿真结果表明，**50 年内**充电站与纯电动车的市场占有率将呈现 S 型曲线增长。均衡情况下充电站的占有率达到 62%，燃油车将彻底被纯电动车 (75%) 和油电混动车 (25%) 取代；
3. 再分析补贴、碳税、电价、消费者社交关系影响、和电动车技术学习率等模型参数对充电站、油电混动车和纯电动车市场占用率扩张的影响：
 - 充电站建设补贴和碳税均与充电站和纯电动车的市场占有率正相关，而与油电混动车的市场占有率负相关，表明提升充电站建设补贴或征收碳排放税有助于引导需求由油电混动车向纯电动车转移；
 - 传统观点认为提升电价有助于充电站扩散，本文指出电价对充电站扩散的影响呈现反 U 型曲线，适当提升电价有可增加充电站收益从而促进其扩散，但过高的电价也会削弱其发展。与此同时在电动车行业发展的早期阶段适当提升电价也有助于纯电动车市场的发展，因为可增加充电站的市场占有率缓解纯电动车充电难问题。另外发现无论什么阶段提升电价均不利于油电混动车的发展，只会利于纯电动车或燃油车；
 - 亲友购车偏好的影响与充电站和纯电动车的市场占有率负相关，而与油电混动车的市场占有率正相关。这可能是因为现阶段纯电动车的技术和配套尚不成熟，社会面上对其的认可度仍较低，增强亲友购车偏好的影响会加深此类社会共识对自身购车选择的影响；
 - 电动车技术学习曲线可降低电动车的生产成本，进而促进充电站、油电混动车和纯电动车市场的发展，并在一定程度上引导油电混动车需求向纯电动车转移。对比起对充电站市场的影响，其对油电混动车和纯电动车的促进效果更大。这可能是因为受益于较高的建设补贴和在电动车市场达到一定规模后，充电站市场的规模即可满足电动车需求的后续发展而无需进一步扩大。
4. 在上述实验中均假设能源站是在小世界网络下进行演化博弈，进一步讨论不同的复杂网络模型（最近邻网络、随机网络、小世界网络、无标度网络）对演化博弈结果的影响：
 - 最近邻网络中每个节点与最近的有限个节点连接，表明每个能源站与最近的若干能源站交换信息；
 - 随机网络中每个节点随机与其他有限个节点连接，表明每个能源站随机选择其他能源站交换信息；
 - 小世界网络与现实大多数网络最为接近，具有平均距离小、聚类系数大的特点（见第 18.8 节），表明存在能源站企业联盟或信息共享平台实现信息共享；
 - 无标度网络的特点是网络中的大部分节点只和很小部分节点连接，而极少数节点与非常多的节点连接（见第 18.9 节），表明能源站市场由少数垄断型企业控制，同属一个企业的能源站信息共享。

从最近邻网络到小世界网络、无标度网络再到随机网络，整体上网络的聚类系数依次减小、平均距离逐渐增大，而充电站和纯电动车的市场占有率也按此顺序减小，油电混动车的市场占有率则按此顺序增大。表明现实中政府应优先组织建立行业协会或信息共享平台并补贴鼓励行业扩张以扶持电动车和充电站市场的发展，在此过程中也促进油电混动车需求向纯电动车转移。同时小世界网络下充电站和纯电动车的市场占有率优于无标度网络也表明建立行业协会或信息共享平台的优先度应高于扶持龙头企业的发展。

13.1.4 英汉互译

English	Chinese	English	Chinese	English	Chinese
proliferate	增殖、扩散 (v)	pile	桩	sector	行业、领域
dilemma	两难困境	outlet	电源插座	adequacy	充足 (n)
advent	降临 (n)	punitive	惩罚的	determinant	决定因素
lifespan	有效期	unilateral	单方面的	trait	特质
up-bottom	自上而下的	premium	附加费	contingent	依情况而定的
cascade	级联、串联 (n,v)	designate	指派	payoff	收益
depreciation	折旧 (n)	compatible	兼容的	hinder	阻碍 (v)
offset	抵消 (v)	stifle	扼杀 (v)	conducive	有利于的 (adj)
oligopoly	寡头垄断 (n)	agglomeration	集聚 (n)		

赌书消得泼茶香 当时只道是寻常

第五部分

***Statistical and Econometric Methods for
Transportation Data Analysis, SEMTDA***

第 14 章

Fundamentals

14.1 Descriptive Statistics, 描述统计学

概述

- □ ×

本章主要介绍两类用于描述与理解数据的方法，采用合理的分析方法将有助于实现对数据的精确、客观分析：

1. 数值描述法 (numerical descriptive measures)，最著名的即是点估计 (point estimator)：使用单个值对未知总体参数进行估计以对总体进行推断；
2. 图示法。

1. Measures of Relative Standing, 相对位置量数

对一组数据集自小到大排序，即可以分析数据的边界及比较不同观测值的相对位置关系。常见的量数为百分位数 (percentile)。 P^{th} 分位数表示有 $P\%$ 的观测值低于该值。对于一组总量为 n 的观测值，其 P^{th} 分位数是指第 $(n + 1)P/100$ 个观测值。

四分位数 (quartile) 是一类特殊的百分位数，包含 25%、50%、75% 三个百分位数，分别称为第 1、2、3 四分位数。四分位距 (interquartile range, IQR) 定义为第 3 四分位数与第 1 四分位数的差。

2. Measures of Central Tendency, 集中趋势量数

中位数 (median) 和众数 (mode) 均是常用的集中趋势量数，其中众数一般用于离散变量。但更常用的量数为算数平均值 (arithmetic mean)。根据研究对象的不同，均值可分为样本均值 (sample mean, \bar{X}) 和总体均值 (population mean, μ) 两类，其中样本均值是一个变量，而总体均值为一个常量。

当数据对称 (symmetric) 且单峰 (unimodal) 时，中位数、众数和均值大致相等；当数据为定性数据时，其均值或中位数是无意义的，只能采用众数。

3. Measures of Variability, 差异量数 (变异量数、离散趋势量数)

四分位距 (IQR) 和极差 (range) 即是两个常用的差异量数，两者相比，IQR 的优势在于其能更有效地避免离群值的影响。

标准差 (standard deviation) 和方差 (variance) 是最常用的两个差异量数，因为与极差或 IQR 相比，他们考虑了样本的所有信息。与均值分为样本均值和总体均值类似，标准差和方差同样分为样本标准差 (sample standard deviation, s)、样本方差 (sample variance, s^2) 和总体标准差 (population standard deviation, σ)、总体方差 (population variance, σ^2)。

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n-1}, \quad \sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$$

由上式可以看出，对于一样本量为 n 的样本，计算其样本方差 (或标准差) 时分母为 $n - 1$ ，而非 n ，对此存在以下两种解释：

- (a) s^2 的计算公式中采用了 \bar{X} ，而在计算 \bar{X} 的过程中已经计集了全部的样本，所以此时自由度减一，即在算 \bar{X} 确定的情况下只需要再确定 $n - 1$ 个样本值，即可推出全部的样本值；

(b) 当样本量较小时, s^2 的计算结果有被低估的趋势, 故放大分母为 $n - 1$ 进行修正。

切比雪夫不等式指出, 样本或总体中至少有 $1 - 1/k^2$ 的观测值落在区间 $(\bar{X} - ks, \bar{X} + ks)$ 中, 其中 k 可以是任意正实数。

变异系数 (coefficient of variation, CV) 是一个反映数据相对离散程度的量值, 其消除了数据观测尺度和量纲的影响, 便于比较两组尺度或量纲不同的数据。特别地, 由 CV 的计算公式可以看出, 当两组数据标准差相同但均值不同时, 均值越高的数据被认为具有更低的变异程度。

$$CV = \frac{s}{\bar{X}}$$

4. Skewness and Kurtosis, 偏度与峰度

偏度 (skewness) 是数据分布不对称性程度的度量, 被称为均值附近的三阶中心矩 (central moment), 而方差是二阶中心距。当某概率分布的尾部 (tail) 的右侧大于左侧时被称为右偏 (right skewed) 或正偏 (positively skewed), 反之称为左偏 (left skewed) 或负偏 (negatively skewed), 正偏或负偏的分布统称为偏态分布 (skewed distribution), 与之对应的为正态分布。当分布右偏时, 均值在中位数的右边, 中位数在众数的右边, 左偏分布与之相反。样本偏度 g_1 为总体偏度 γ_1 的估计量, 为三阶中心距的标准化, 式中 m_k 表示 k 阶中心距。

$$g_1 = \frac{m_3}{m_2 \sqrt{m_2}}, \quad m_k = \frac{\sum_{i=1}^n (x_i - \bar{X})^k}{n}$$

$g_1 = 0$ 时表示样本服从正态分布、 $g_1 < 0$ 时表示左偏、 $g_1 > 0$ 时表示右偏。当总体服从正态分布时, g_1 服从均值为 0、标准差为 $\sqrt{n/6}$ 的正态分布。

峰度 (kurtosis) 是数据分布平坦程度的度量, 被称为均值附近的四阶中心矩, 有总体峰度 γ_2 与其估计量样本峰度 g_2 之分。当总体服从正态分布时, 有 $\gamma_2 = 3$, 当 $\gamma_2 > 3$ 时说明数据分布较正态分布更加集中, 称为尖峰态分布 (leptokurtic distribution); 反之则称为低峰态分布 (platykurtic distribution)。在实际应用中, 常常令

$$g_2 = \gamma_2 - 3 = \frac{m_4}{m_2^2} - 3$$

此时对低峰态分布, 有 $g_2 < 0$; 对尖峰态分布, 有 $g_2 > 0$ 。

5. Measures of Association, 相关量数

前述的所有量数仅用于描述单个变量的分布情况, 而相关量数则用于描述不同变量间的潜在关系。协方差 (covariance) 是常用的相关量数。对于两个各自服从正态分布的随机变量 X, Y , 其协方差是他们与各自均值之差的乘积的期望。协方差为正表示两个随机变量为正相关, 绝对值越大表示相关性越强。有总体协方差 COV_p 和样本协方差 COV_s ,

$$COV_p(X, Y) = \frac{\sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)}{N} \quad COV_s(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{n-1}$$

类比变异系数 CV 与标准差 s 的关系, 协方差最大的缺陷在于其结果会受到数据观测尺度和量纲的影响, 使其无法有效量化变量间的具体关系。相关系数即是对协方差进行标准化。皮尔逊积矩相关系数 (Pearson product-moment correlation parameter), 又称皮尔逊相关系数, 用于度量两个随机变量之间的线性关系。有总体皮尔逊相关系数 ρ 和样本皮尔逊相关系数 r 。 $\rho \in [-1, 1]$, $|\rho|$ 越大表示两个变量间的线性关系越强; 而 $\rho > 0$ 表示两个变量间的相关性为正。

$$\rho = \frac{COV_p(X, Y)}{\sigma_X \sigma_Y} \quad r = \frac{COV_s(X, Y)}{s_X s_Y}$$

综上所述, 皮尔逊相关系数的应用要求两个变量均服从正态分布, 而在很多时候存在至少一个变量不服从或者不确定是否服从正态分布, 此时皮尔逊相关系数即不适用, 可采用斯皮尔曼等级相关系数 (Spearman rank correlation parameter, r_s) 作为替代。斯皮尔曼相关系数是一种非参数方法 (nonparametric method)

¹, 其思想是: 对随机变量 X 进行排序, x_i 的排名 (rank) 为 $R(x_i)$, 随后计算 X, Y 排名的皮尔逊相关系数。 r_s 同样介于 $[-1, 1]$, 判断法则与皮尔逊相关系数相同。斯皮尔曼相关系数较皮尔逊相关系数的适用范围更广, 但当变量服从正态分布时, 斯皮尔曼相关系数的精度较低, 故此时推荐皮尔逊相关系数。

$$r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}, \quad d_i = R(x_i) - R(y_i)$$

需要强调的是, 相关性 (correlation) 无法推出因果性 (causality)。

6. Properties of Estimators, 估计量的特性

前文介绍了多个样本统计量, 在实际应用中它们将作为总体参数的估计量。估计量为一个变量, 受样本影响而一个好的估计量应该满足以下几项特性:

- (a) 无偏性 (unbiasedness): 当估计量的期望等于估计量所对应的真值时 (如 $E(\bar{X}) = \mu_X$), 该估计量为无偏估计量;
- (b) 有效性 (efficiency): 方差越小的估计量被认为具有更高的有效性。对于参数 θ 的无偏估计量 $\hat{\theta}$, 克拉美-罗下界 (Cramer-Rao lower bound, CRLB) 给出了判断 $\hat{\theta}$ 是否有效的充分条件, 若满足下式, 则认为 $\hat{\theta}$ 为 θ 的有效估计量;

$$VAR(\hat{\theta}) \geq \frac{1}{I(\theta)}, \quad I(\theta) = -E \left[\frac{\partial^2 \ln f(x|\theta)}{\partial \theta^2} \right]$$

- (c) 相合性 (consistency): 若满足 $\lim_{n \rightarrow \infty} P[|\hat{\theta} - \theta| > \varepsilon] = 0$, 则称 $\hat{\theta}$ 为 θ 的相合估计量;
- (d) 充分性 (sufficiency): 如果一个估计量包含样本中涉及参数的全部信息, 则被称为充分估计量。

7. Methods of Displaying Data

- (a) 直方图 (histogram): 适用于数据可被自然分组 (如性别) 或当定义小的子区间有助于挖掘数据的更多信息时。直方图有助于揭示数据的对称性、偏度和峰度;
- (b) 累计频率曲线 (ogive): ogive 相当于直方图的积分, 为单调递增曲线, 收敛于 1。ogive 图可直观地显示样本中高于或低于某观测值的数据所占的比例;
- (c) 箱须图 (box & whisker plot): 箱须图由五个特征点组成, 分别是数据的两个最值和三个四分位数。第 1 四分位数和第 3 四分位数之间的范围组成箱, 第 1 四分位数至最小值和第 3 四分位数至最大值之间的范围组成须², 由此绘制成箱须图。箱须图有助于判断数据的中心趋势 (中位数)、数据的分布情况 (箱和须的长度)、偏度 (中位数于箱的位置) 以及识别可能的异常值 (箱两边 1.5IQR 以外的点);
- (d) 点图 (scatter diagram): 用于表示两个连续变量之间的关系;
- (e) 饼图、条形图和折线图 (pie, bar and line chart): 饼图和条形图均适用于展示名义变量的分布情况, 两者可互相替代。折线图常用于时间序列数据。

14.2 Interval Estimation, Hypothesis Testing and Population Comparison, 区间估计、假设检验与总体比较

概述



置信区间 (confidence interval, CI)、假设检验 (hypothesis test) 和总体比较 (population comparison) 在交通领域有着广泛的应用。但在实际工程中, 置信区间往往被忽视, 而假设检验和总体比较也常被误解和误用。本章介绍的这些技术基于一系列假设用于解释、检验以及决策, 有助于回答当今交通工程中遇到的问题, 例如: 在特定的交叉口发生事故是否意味着该路口存在问题? 解除空中运输市场的管控是否有助于商务出行比重的增加? 改变运输系统的运营补贴水平是否会改变它们的运营业绩?

本章所述的一些方法是基于对估计量或样本总体分布的猜想 (大多是近似正态分布) 下推导的, 所以在运用这类方法时需

¹但在许多实际问题中, 我们对总体分布的形式往往所知甚少 (如只能作出诸如连续型分布、关于均值对称等微弱的假定)。这时就需要使用不必 (或很少) 依赖于总体分布形式的统计推断方法, 此类推断方法通常称为非参数方法

²须的长度最长为 1.5IQR, 之外的点被认为异常值

要首先验证数据是否满足相关的假设或猜想，如果不满足，则必须采用非参数统计方法 (nonparametric statistical method)。

14.2.1 Confidence Interval, 置信区间

1. 之前已介绍了多个估计量以及判断估计量好坏的几项指标，但无论估计量满足什么特性，它们均由样本决定，与真值之间不可避免地存在误差。为此本节讨论区间估计 (interval estimation) 的概念。区间估计即是在估计总体时给出一个区间而非一个值，位于区间内的值将满足预先确定的置信度，这一区间被称为置信区间 (confidence interval, CI)。CI 的下限为下置信限 (lower confidence limit, LCL)，上限被称为上置信限 (upper confidence limit, UCL)。CI 越宽，则认为其总体置信度越高。

2. Confidence Interval for μ with Known σ

中心极限定理 (central limit theorem, CLT) 指出，对于服从任意分布的总体，假设其均值为 μ 、标准差为 σ ，则其样本均值 \bar{X} 服从均值为 μ 、标准差为 σ/\sqrt{n} 的正态分布，则构造随机变量 $Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$ 服从标准正态分布，有 $P(-1.96 < Z < 1.96) = 0.95$ ，变换得

$$P\left(\bar{X} - 1.96 \frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + 1.96 \frac{\sigma}{\sqrt{n}}\right) = 0.95$$

因此，当 σ 已知时， μ 的一系列置信区间如下（实际应用中，置信度一般取 90%-99%）：

$$\text{CI}_{95\%} : \bar{X} \pm 1.96 \frac{\sigma}{\sqrt{n}}, \quad \text{CI}_{90\%} : \bar{X} \pm 1.645 \frac{\sigma}{\sqrt{n}}, \quad \text{CI}_{1-\alpha} : \bar{X} \pm Z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

3. Confidence Interval for μ with Unknown σ

当总体标准差 σ 未知、且总体服从正态分布时，自然地构造统计量 $t^* = \frac{\bar{X} - \mu}{s/\sqrt{n}}$ ， t^* 近似服从自由度为 $n-1$ 的 t 分布³，则 μ 的置信区间如下

$$\text{CI}_{1-\alpha} : \bar{X} \pm t_{\alpha/2} \frac{s}{\sqrt{n}}$$

4. Confidence Interval for a Population Proportion, 总体比例的置信区间

在很多时候需要涉及名义变量而非连续变量，例如调查选择公交出行或地铁出行的比例，此时便有必要计算名义变量各分类所占的比例及对应的置信区间。无论这一名义变量为几分类，针对任一分类都是一个二分类问题，显然有 $\mu = EX = p$, $\sigma = DX = p(1-p)$

$$\begin{cases} P(X = 1) = p, & (\text{Classification} = i) \\ P(X = 0) = 1 - p, & (\text{Classification} \neq i) \end{cases}$$

对于样本比例 \hat{p} ，当样本量足够大时 ($np \geq 5$ 且 $n(1-p) \geq 5$)，由中心极限定理， \hat{p} 服从均值为 p 、标准差为 $\sqrt{p(1-p)/n}$ 的正态分布，此时 p 的置信区间为

$$\text{CI}_{1-\alpha} : \hat{p} \pm Z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

5. Confidence Interval for the Population Variance

在很多时候总体的变异情况具有很高的分析价值，例如车道上行车速度的变异程度（即交通流的紊乱程度）与事故的概率密切相关，此时在计算样本方差或样本标准差后，还需要计算其置信区间。构造 $X^2 = \frac{(n-1)s^2}{\sigma^2} = \frac{\sum_{i=1}^n (x_i - \bar{X})^2}{\sigma^2}$ ，假设总体服从正态分布，则 $\frac{x_i - \bar{X}}{\sigma}$ 近似服从标准正态分布，则 X^2 近似服从自由度为 $n-1$ 的 χ^2 分布⁴，则 σ^2 的置信区间

$$\text{CI}_{1-\alpha} : \left[\frac{(n-1)s^2}{\chi_{\alpha/2}^2}, \frac{(n-1)s^2}{\chi_{1-\alpha/2}^2} \right]$$

³ t 分布：设随机变量 $X \sim N(0, 1)$, $Y \sim \chi_n^2$ 且 X, Y 相互独立，则称 $T = \frac{X}{\sqrt{Y/n}}$ 服从自由度为 n 的 t 分布。 t 分布与标准正态分布的曲线类似， t 分布曲线被称为丘形曲线 (mound shaped curve)，而正态分布曲线被称为钟形曲线 (bell shaped curve)，当 $n \rightarrow \infty$ 时， t 分布将收敛于标准正态分布。

⁴ χ^2 分布：假设 Z_i 服从标准正态分布，则 $X^2 = \sum_{i=1}^k Z_i^2 \sim \chi^2(k)$ ，即服从自由度为 k 的 χ^2 分布。 χ^2 分布为右偏，当 $k \rightarrow \infty$ 时， χ^2 分布收敛于均值为 k 、方差为 $2k$ 的正态分布。

14.2.2 Hypothesis Testing, 假设检验

- 假设检验 (**hypothesis test**) 用于评估两个或以上群体之间的总体参数 (期望、方差、比例等) 的差异是偶然产生或能反映不同群体间的结构性差异。假设检验首先假设总体服从的分布，并得到在假设成立的情况下样本可能的取值，当实际观测值在原有的假设中被认为难以实现时，则认为原假设不成立。需要补充的是，在进行假设检验时，需要尽可能地控制单一变量条件，否则将降低结果的可靠性，尽管在很多实际情况下很难做到。以下介绍假设检验的具体过程：
- 首先，需要做出两个互为对立的假设：零假设 (**null hypothesis, H_0**) 和备择假设 (**alternative hypothesis, H_a**)。其中 H_0 对一个或多个参数做出具体的假设，当存在足够多的证据时将拒绝 H_0 ，而 H_a 则是包含了所有与 H_0 对立的情况，由此包含了参数的所有情况。假设检验的目的即是决定是否拒绝 H_0 。即在 H_0 成立的条件下，预先确定置信度 (90%-99%)，则可推算待观测统计量的置信区间，当观测得到的统计量落在置信区间之外时，即可得出结论：在给定的置信度下，有充分理由拒绝 H_0 。
- 在假设检验中，无论是否拒绝 H_0 ，均是基于一定的置信度做出的决定，也就是说，假设检验永远无法证明 H_0 是切实的正确或错误。这也就意味着当基于假设检验做出决定时，依然有可能犯错
 - 第一类错误：** H_0 为真却拒绝 H_0 ，记犯第一类错误的概率为 α ， α 被称为显著性水平 (**significance level**)，是预先确定的，而 $1 - \alpha$ 则为 H_0 为真且接受的概率；
 - 第二类错误：** H_0 为假却接受 H_0 ，记犯第二类错误的概率为 β ， $1 - \beta$ 则为 H_0 为假且拒绝的概率。一般来说，我们希望 α 和 β 都尽可能小，但两者的变化呈负相关。合理地设计实验和扩充样本数是同时控制 α 和 β 最有效的方法。在具体应用中， α 往往取 1%-10%，而 β 很多时候会被忽略。 α 的具体取值和两类错误优先关注哪个取决于研究的问题和犯错的严重性。
- 很多时候，在对总体参数 θ 做假设检验时，我们会做出如下的 H_0 和 H_a ，此时可直接根据前述关于置信区间的方法根据预定的显著水平做出置信区间进行检验；

$$H_0 : \theta = x \iff H_a : \theta \neq x$$

但有时候，我们不只关注 θ 是否会显著变化，更关注是否会显著增长（或减少），此时会做出如下 H_0 和 H_a ，此时方法与上类似，但此时为单边检验 (**one-tailed test**)，做出的置信区间为单边置信区间。

$$H_0 : \theta \leq x \iff H_a : \theta > x$$

- 由上述分析得到，在计算检验统计量后，需要基于显著水平 α 以确定假设检验的结果（拒绝或不拒绝），每一个 α 对应一个结果，且 α 越小越可能接受 H_0 。定义 **p** 值 (**p-value, probability-value**) 为所有使得结果为拒绝的 α 中最小的那个。**p** 值越小说明 H_0 越有可能被拒绝，故 **p** 值可量化样本数据对 H_a 的支持程度。假设 $Z \sim N(0, 1)$ 且观测到 $Z=3.27$ ，则对应的 **p** 值为

$$p-value(Z = 3.27) = P(Z \leq -3.27) + P(Z \geq 3.27) = 0.001$$

这意味着当预设 $\alpha \geq 0.001$ 时（如 $\alpha = 0.05$ ），将拒绝 H_0 ；反之将接受。

- 在给出了 **p** 值的概念后，就可以得到假设检验的另一种更广泛的做法：
 - 首先确定 H_0 和 H_a ；
 - 基于 H_0 及研究的统计量构造并计算对应的检验统计量（见章节 14.2.1）；
 - 最后计算该统计量的 **p** 值⁵。

14.2.3 Comparing Two Populations, 两个总体的比较

- 本章将采用假设检验的方法比较两类不同的总体的统计量。需要说明的是，本章介绍的方法适用于数字型变量，如定距变量 (**interval variable**)⁶ 和定比变量 (**ratio variable**)⁷，并要求大致服从正态分

⁵在所列的检验统计量公式中，样本量 n 均位于分母，意味着样本量越大检验统计量也越大，**p** 值则越小

⁶定距变量 interval variable：数字型变量，可排序、加减，不存在基准。

⁷定比变量 ratio variable：数字型变量，可排序、加减，存在基准。

布。对于定序变量 (**ordinal variable**)⁸、定类变量 (**nominal variable**)⁹或明显不服从正态分布的变量，需采用非参数方法。本节假设检验的方法为：计算假设统计量及对应 **p** 值，与预定 α 比较得到结果；

2. 在介绍比较方法之前，首先介绍待比较的两类样本间可能的关系：可能是独立样本 (**independent sample**)，也可能是配对样本 (**paired sample**)。以下通过一个例子说明：假设要比较一批新型轮胎和现用轮胎的质量差别，可设计如下两种实验

- (a) 随机抽取 20 辆车换上新型轮胎作为一组，另一组为使用现用轮胎的另外 20 辆同类型的车，分别对两组车进行试验并比较轮胎的寿命；
- (b) 随机抽取 20 辆车，每辆车四个轮胎中两个换为新型轮胎，另外两个采用现用轮胎，对这 20 辆车进行试验并比较轮胎的寿命。

对于第一种试验方法，两组样本的数据互相独立，即独立样本 (**independent sample**)，而第二种试验方法的两组样本数据一一配对，即配对样本 (**paired sample**)，这一实验设计方法被称为配对设计 (**matched pairs design**)。配对设计能充分实现单一变量，降低样本间变异性。

3. Testing Differences between Two Means: Independent Samples

既然是比较均值，由中心极限定理，只要样本数足够大（要求两个样本数均不小于 25），均值即近似服从正态分布。根据实际的目的，可作出以下两类假设

$$H_0 : \mu_1 - \mu_2 = c \iff H_a : \mu_1 - \mu_2 \neq c \quad \text{or} \quad H_0 : \mu_1 - \mu_2 \leq c \iff H_a : \mu_1 - \mu_2 > c$$

构造检验统计量 Z^* 如下，当样本足够大 ($n_1 \geq 25, n_2 \geq 25$) 时， Z^* 近似服从正态分布；当样本较小 ($n_1 < 25, n_2 < 25$) 但两个样本均近似服从正态分布时， Z^* 近似服从自由度 n 的 t 分布（均假设 $\sigma_1 \neq \sigma_2$ ）

$$Z^* = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}, \quad \text{CI}_{1-\alpha} : (\bar{X}_1 - \bar{X}_2) \pm Z_{\alpha/2} \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}, \quad n = (s_1^2/n_1 + s_2^2/n_2)^2 \cdot \left[\frac{(s_1^2/n_1)^2}{n_1 - 1} + \frac{(s_2^2/n_2)^2}{n_2 - 1} \right]^{-1}$$

当假设 $\sigma_1 = \sigma_2$ 时，可将两个样本的方差合并以简化表达式，此时优点在于当样本量较小时， t^* 的自由度直接等于 $n_1 + n_2 - 2$ ，而不需专门计算

$$t^* = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{s_p^2(1/n_1 + 1/n_2)}}, \quad \text{CI}_{1-\alpha} : (\bar{X}_1 - \bar{X}_2) \pm t_{\alpha/2} \sqrt{s_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}, \quad s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

以上的 CI 均是 $\mu_1 - \mu_2$ 的置信区间。

4. Testing Differences between Two Means: Paired Observations (配对样本)

本节所作的 H_0, H_a 与独立样本比较的方法完全相同，因为数据两两配对，记 $\mu_d = \mu_1 - \mu_2$, $\bar{X}_d = \bar{X}_1 - \bar{X}_2$, \bar{X}_d 的标准差为 s_d ，样本大小为 n ，则统计量 t^* 近似服从自由度为 $n - 1$ 的 t 分布，对应的有 t^* 的置信区间

$$t^* = \frac{\bar{X}_d - \mu_d}{s_d / \sqrt{n}}, \quad \text{CI}_{1-\alpha} : \bar{X}_d \pm t_{\alpha/2} \frac{s_d}{\sqrt{n}}$$

5. Testing Differences between Two Population Proportions

本段讨论的内容适用于名义变量，但分析方法与前文类似。无论名义变量可取几类，对其中任意一类，都是一个二分类问题，有 $\mu = p$, $\bar{X} = \hat{p}$, $\sigma^2 = p(1 - p)$, $s^2 = \hat{p}(1 - \hat{p})$ ，以此代入，得到统计量和 CI

$$Z^* = \frac{(\hat{p}_1 - \hat{p}_2) - (p_1 - p_2)}{\sqrt{\hat{p}_1(1 - \hat{p}_1)/n_1 + \hat{p}_2(1 - \hat{p}_2)/n_2}}, \quad \text{CI}_{1-\alpha} : (\hat{p}_1 - \hat{p}_2) \pm Z_{\alpha/2} \sqrt{\frac{\hat{p}_1(1 - \hat{p}_1)}{n_1} + \frac{\hat{p}_2(1 - \hat{p}_2)}{n_2}}$$

对于二分类问题，假设 $E(X) = p$ 确定， $D(X) = p(1 - p)$ 也随之确定。如果 $H_0 : p_1 - p_2 = 0$ ，即假设两个二分类问题期望相同时，也就意味着假设两个问题服从相同的二项分布。此时不只有 $p_1 - p_2 = 0$, \hat{p}_1, \hat{p}_2 也可以用 \hat{p} 联合表示

$$Z^* = \frac{(\hat{p}_1 - \hat{p}_2) - 0}{\sqrt{\hat{p}(1 - \hat{p})(1/n_1 + 1/n_2)}}, \quad \text{CI}_{1-\alpha} : (\hat{p}_1 - \hat{p}_2) \pm Z_{\alpha/2} \sqrt{\hat{p}(1 - \hat{p}) \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}, \quad \hat{p} = \frac{x_1 + x_2}{n_1 + n_2}$$

⁸定序变量 ordinal variable：离散变量，可排序、不可加减，如年级变量。

⁹定类变量 nominal variable：离散变量，不可排序、不可加减，如性别变量。

6. Testing Differences between Two Population Variances

当定义 $H_0 : \sigma_1^2 = \sigma_2^2$ 时, 构造检验统计量 $F_{(n_1-1, n_2-1)}^*$ 服从分子自由度 $n_1 - 1$ 、分母自由度 $n_2 - 1$ 的 F 分布¹⁰, 式中 s_1 与 s_2 的位置可互相调换。

$$F_{(n_1-1, n_2-1)}^* = \frac{s_1^2}{s_2^2}$$

当定义 $H_0 : \sigma_1^2 \leq \sigma_2^2$ 时需要特别注意。此时为了拒绝 H_0 , 要求 s_1 显著大于 s_2 。这类检验往往为单侧检验, 此时在构造检验统计量时, 要求 s_1 必须置于分子, 使 $F_{(n_1-1, n_2-1)}^* > 1$, 这是因为对 F 分布做单侧检验时, 拒绝域一般都在右侧, 也就是说统计量必须充分大才有可能落入拒绝域。当双侧检验时, 尽管拒绝域位于两侧, 但因为 F 分布的不对称性, 右侧拒绝域显著大于左侧, 故此时会选择将 s_1, s_2 中较大的一项置于分子。

14.2.4 Nonparametric Methods, 非参数方法

1. 本章之前所述的统计方法大多基于对总体参数或统计数据的特定假设, 例如近似服从正态分布、必须为定距变量或定比变量 (均值和方差检验时) 以及样本充分大 (T 检验可在一定程度上解决小样本问题) 等等, 这些方法被称为参数方法。而当变量不符合上述假设时, 则需要采用非参数方法;
2. 非参数方法对样本及总体的假设更少, 且往往不涉及参数 (例如判断一个样本是否为随机抽样, 此时显然与总体参数无关), 为此非参数方法较参数方法具有更大的适用范围, 但同时, 当对适用于参数方法的数据应用非参数方法时, 检验的结果将使第二类错误的概率增大。以下介绍非参数方法的适用范围:
 - (a) 样本数据是频率计数且参数检验方法不适用;
 - (b) 样本数据是定序变量;
 - (c) 所做的检验不关心总体参数 (如期望、方差);
 - (d) 样本严重违反参数检验方法的假设: 近似服从正态分布、样本足够大、定距或定比变量等等;
 - (e) 样本在一定程度上违反参数检验方法的假设, 且检验的结果具有统计学上的边际效应。
3. **Sign Test, 符号检验**

符号检验是多种非参数检验方法的基础。 符号检验的本质目的是检验某一选项同其它选项相比是否显著地更受欢迎。假设一共有 A B 两个选项, 进行符号检验时, 用 “+” 表示选择 A 选项, “-” 表示选择 B 选项, 符号检验因此得名。记 p 为调查者更倾向 A 的概率, 则作假设如下

$$H_0 : p = 0.5 \iff H_a : p \neq 0.5$$

当 H_0 成立时, 记 X 为 n 个样本中选择 A 选项的次数, X 服从 $p = 0.5$ 的二项分布, 且当样本量足够大时 ($n \geq 20$), 二项分布将趋近于正态分布, 此时构造统计检验量 Z^*

$$Z^* = \frac{X - E(X)}{\sqrt{D(X)}}, \quad E(X) = pn = 0.5n, \quad D(X) = \sqrt{p(1-p)n} = \sqrt{0.25n}$$

4. Median Test, 中位数检验

中位数检验的目的是基于采样的样本判断假想的中位数是否可信, 其假设如下

$$H_0 : median = \lambda \iff H_a : median \neq \lambda$$

中位数检验可以由符号检验实现: 只需记样本中大于 λ 的观测值为 “+”, 样本中小于 λ 的观测值为 “-” (舍去等于 λ 的观测值)。因为中位数将总体分布平分为大小相等的两部分, 则当 H_0 成立时, 同样有 $p = 0.5$, 其余部分同上。

5. Mann-Whitney U Test, 曼 - 惠特尼 U 检验

曼 - 惠特尼 U 检验, 又称 Mann-Whitney-Wilcoxon (MWW) 检验, 目的是比较两组独立样本 (**independent sample**) 的均值 (或中位数) 是否显著不同。MWW 检验允许两个样本大小不同、也适用于小样本。假设

¹⁰ F 分布: 设随机变量 $\chi_1^2 \sim \chi^2(n_1)$, $\chi_2^2 \sim \chi^2(n_2)$ 且相互独立, 则称 $F_{(n_1, n_2)}^* = \frac{\chi_1^2/n_1}{\chi_2^2/n_2}$ 服从分子自由度 n_1 、分母自由度 n_2 的 F 分布。 F 分布为非对称的。

H_0 : 两个样本分布来源于相同的总体; $\iff H_a$: 两个样本分布来源于不同的总体。

首先将两个样本合并为一个大样本并排序, 对应地计算每个观测值的排名 (rank: 1-n, 当出现并列值时, 并列值的排名为并列的均值, 如 2、3 名并列, 则它们的排名均为 2.5), 记 n_1, n_2 分别为样本 1、2 的尺寸, R_1, R_2 为样本 1、2 的观测值的排名的和, 则构造 U 统计量如下

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \quad U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2$$

令 $n = n_1 + n_2$, 此时有 $R_1 + R_2 = 0.5 \times n(1 + n)$, 则

$$U_1 + U_2 = 2n_1 n_2 + \frac{n_1(n_1 + 1) + n_2(n_2 + 1)}{2} - \frac{n(1 + n)}{2} = n_1 n_2$$

令 $U = \min\{U_1, U_2\}$, 此时 $U_{max} = 0.5n_1 n_2$ 。当 H_0 成立时, U 的期望及方差如下, 当样本较小时, 需要借助 U 统计量的分布表; 当样本足够大时 ($n_1 \geq 10, n_2 \geq 10$), U 快速收敛为正态分布, 则可构造检验统计量 Z^*

$$Z^* = \frac{U - E(U)}{\sigma_U}, \quad E(U) = \frac{n_1 n_2}{2} \quad \sigma_U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$$



推导 σ_U

根据以上的思路, 可以非常简单的推导出 $E(U) = 0.5n_1 n_2$, 但推导 σ_U 的表达式就存在困难, 以下从定义的角度重新推导 σ_U 的表达式, $E(U)$ 的表达式也可同理推导。根据 U 的表达式, 显然存在以下关系

$$E(U) = n_1 n_2 + \frac{n_1(n_1 + 1) + n_2(n_2 + 1)}{4} - \frac{1}{2} [E(R_1) + E(R_2)], \quad \sigma_U = \sigma_{R_1} = \sigma_{R_2}$$

显然, 只需要求 σ_{R_1} 的表达式。注意到, 当 H_0 成立, 即两个样本来源于相同的总体时, 样本 1 中的 n_1 个观测值相当于从大小为 n 的混合样本中随机抽取 n_1 个而来的, 此时一共有 $C_n^{n_1}$ 中抽样方法组成样本 1, 且每一个组合的概率相等, 那么在样本 1 中, 出现 x_i 的概率为 $\frac{C_{n-1}^{n_1-1}}{C_n^{n_1}} = \frac{n_1}{n}$, 同时出现 x_i, x_j 的概率为 $\frac{C_{n-2}^{n_1-2}}{C_n^{n_1}} = \frac{n_1(n_1-1)}{n(n-1)}$ 。
猜想: 与样本 1 有关的统计量 R_1 的方差 σ_{R_1} 和混合样本的方差 s^2 有关。

$$\sigma_{R_1}^2 = \frac{1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} (R_{1i} - \bar{R}_1)^2, \quad s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

首先讨论 s^2

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i^2 - 2x_i \bar{x} + \bar{x}^2) = \frac{1}{n-1} \sum_{i=1}^n x_i^2 - \frac{2\bar{x}}{n-1} \cdot \frac{n}{n-1} \sum_{i=1}^n x_i + \frac{1}{n-1} \sum_{i=1}^n \bar{x}^2 = \frac{1}{n-1} \sum_{i=1}^n x_i^2 - \frac{n}{n-1} \bar{x}^2$$

同样地讨论 σ_{R_1} 。在一共 $C_n^{n_1}$ 个可能的 R_{1i} 中, x_i 出现的总次数为 $\frac{n_1}{n} C_n^{n_1}$, x_i, x_j 同时出现的总次数为 $\frac{n_1(n_1-1)}{n(n-1)} C_n^{n_1}$

$$\begin{aligned} \sigma_{R_1}^2 &= \frac{1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} R_{1i}^2 - \frac{2\bar{R}_1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} R_{1i} + \frac{1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} \bar{R}_1^2 = \frac{1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} R_{1i}^2 - \bar{R}_1^2 \\ \therefore \bar{R}_1 &= \frac{1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} R_{1i} = \frac{1}{C_n^{n_1}} \frac{C_n^{n_1} n_1}{n} \sum_{i=1}^n x_i = n_1 \bar{x} \\ \therefore \sigma_{R_1}^2 &= \frac{1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} R_{1i}^2 - n_1^2 \bar{x}^2 \end{aligned}$$

难点在于 $\frac{1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} R_{1i}^2$ 。在一共 $C_n^{n_1}$ 个可能的 R_{1i}^2 中, 拆开完全平方式, x_i^2 出现的总次数为 $\frac{n_1}{n} C_n^{n_1}$, $x_i x_j$ 出现的总次数为 $2 \frac{n_1(n_1-1)}{n(n-1)} C_n^{n_1}$ 。注意到

$$\frac{n_1}{n} C_n^{n_1} = \frac{n_1(n_1-1)}{n(n-1)} C_n^{n_1} + \frac{n_1 n_2}{n(n-1)} C_n^{n_1}$$

因此, $\frac{n_1}{n} C_n^{n_1} x_i^2$ 可分成两部分: $\frac{n_1 n_2}{n(n-1)} C_n^{n_1} x_i^2$ 单独一组; $\frac{n_1(n_1-1)}{n(n-1)} C_n^{n_1} x_i^2$ 与 $2 \frac{n_1(n_1-1)}{n(n-1)} C_n^{n_1} x_i x_j$ 配成完全平方式

$$\begin{aligned}
 & (\sum_{i=1}^n x_i)^2 \\
 \therefore \quad & \frac{1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} R_{1i}^2 = \frac{1}{C_n^{n_1}} \cdot \frac{n_1 n_2}{n(n-1)} C_n^{n_1} \sum_{i=1}^n x_i^2 + \frac{1}{C_n^{n_1}} \cdot \frac{n_1(n_1-1)}{n(n-1)} C_n^{n_1} \left(\sum_{i=1}^n x_i \right)^2 = \frac{n_1 n_2}{n(n-1)} \sum_{i=1}^n x_i^2 + \frac{n n_1(n_1-1)}{n-1} \bar{x}^2 \\
 \implies \quad & \sigma_{R_1}^2 = \frac{1}{C_n^{n_1}} \sum_{i=1}^{C_n^{n_1}} R_{1i}^2 - n_1^2 \bar{x}^2 = \frac{n_1 n_2}{n(n-1)} \sum_{i=1}^n x_i^2 + \left[\frac{n n_1(n_1-1)}{n-1} - n_1^2 \right] \bar{x}^2 = \frac{n_1 n_2}{n(n-1)} \sum_{i=1}^n x_i^2 - \frac{n_1 n_2}{n-1} \bar{x}^2 \\
 \implies \quad & \sigma_U^2 = \sigma_{R_1}^2 = \frac{n_1 n_2}{n} s^2
 \end{aligned}$$

以上即是 σ_U 与 s^2 的关系，无论样本 1、2 中是否存在重复观测值，上式均适用。

6. Wilcoxon Signed-Rank Test for Matched Pairs, 威尔科克森符号秩检验

威尔科克森符号秩检验与前述曼 - 惠特尼 U 检验的目的一致，但后者适用于两组独立样本的比较，而威尔科克森符号秩检验则适用于两组配对样本 (paired sample) 的比较。即两配对样本分别为 x, y ，则假设如下

$$H_0 : \text{median}_x - \text{median}_y = 0 \iff H_a : \text{median}_x - \text{median}_y \neq 0$$

首先计算 $d_i = x_i - y_i$ ，并对 $|d_i|$ 自小到大排序得到对应的每个 d_i 的排名 (rank: 1-n)。根据 d_i 的正负将其分为两个集合 d_i^+, d_i^- ，则构造威尔科克森 T 统计量如下

$$T = \min\{R^+, R^-\}, \quad R^+ = \sum d_i^+ \quad R^- = \sum d_i^-$$

参考曼 - 惠特尼 U 检验公式推导，可以得到 $R^+ + R^- = 0.5n(n+1)$ ，当 H_0 成立时同样可以得到 T 的期望及方差。当样本足够大时 ($n \geq 25$)，统计量 T 收敛至正态分布，则可以构造检验统计量 Z^*

$$Z^* = \frac{T - E(T)}{\sigma_T}, \quad E(T) = \frac{n(n+1)}{4} \quad \sigma_T = \sqrt{\frac{n(n+1)(2n+1)}{24}}$$

7. Kruskal-Wallis Test, 克鲁斯卡尔 - 沃利斯检验

克鲁斯卡尔 - 沃利斯 (Kruskal-Wallis) 检验，又称 K-W 检验，是曼 - 惠特尼 (MWW) 检验的推广。K-W 检验用于判断 k 组独立样本是否来自相同的概率分布。假设如下

$$\begin{aligned}
 H_0: \quad & k \text{ 组样本的概率分布具有相同的中位数 (均值);} \\
 \iff H_a: \quad & \text{至少有一组样本的概率分布的中位数 (均值) 与其它样本不同。}
 \end{aligned}$$

同 MWW 检验方法类似，首先将 k 组样本的全部观测值混合排序，当无重复观测值时，构造统计量 H

$$H = \frac{12}{n(n+1)} \sum_{i=1}^k \frac{R_i^2}{n_i} - 3(n+1)$$

当样本充分大时 ($n_i \geq 5$)，H 近似服从自由度为 k-1 的 χ^2 分布。在给定的显著水平 α 下，当 $H > \chi_{k-1;\alpha}^2$ 则拒绝 H_0 。当 K-W 检验拒绝 H_0 时，说明至少存在一组样本的概率分布的中位数 (均值) 不同于其它样本。进一步地，给出判断具体哪组样本存在差异的方法：比较样本 i, j ，首先计算 $\bar{R}_i = R_i/n_i$, $\bar{R}_j = R_j/n_j$ ，由此得到统计量 $D_{ij} = |\bar{R}_i - \bar{R}_j|$ ，若满足 $D_{ij} \geq C_{KW}$ ，则说明样本 i, j 的概率分布的中位数存在差异

$$C_{KW} = \sqrt{\chi_{k-1;\alpha}^2 \left[\frac{n(n+1)}{12} \right] \left(\frac{1}{n_i} + \frac{1}{n_j} \right)}$$

8. Chi-Square Goodness-of-Fit Test

卡方检验具有极大的应用性。应用于卡方检验的数据可以是定类、定序、定距、定比等任意类型，也可以是计数数据或频率数据。需要注意的是，当样本量过小或计算期望概率偏小时，卡方检验将不适用。卡方检验的步骤可概括如下：

- (a) 针对总体分布作出假设 ($H_0 \iff H_a$);

- (b) 基于 H_0 和某一特定的统计模型, 得到事件的期望概率 $E_i, i = 1, \dots, k (\sum_i^k E_i = 1)$ 。对于连续数据或定义域长度无穷的数据, 应进行分箱, 分为 k 个箱;
- (c) 实验并记录实际观测频率 $O_i, i = 1, \dots, k (\sum_i^k O_i = 1)$;
- (d) 计算预测值与实际观测值的差值, 由此计算卡方检验的统计量 X^2 (服从自由度为 $k-1$ 的 χ^2 分布);

$$X^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

(e) 基于计算的统计量和预定的显著水平 α 得到假设检验结果。

除了检验模型的匹配程度, 卡方检验还可用于其它诸多场合, 例如检验两个分类变量是否独立 (H_0 : 两个分类变量互相独立)。假设某总体中的个体具有 A, B 两种属性, 且每种属性分别具有 r, c 种分类, 则可以得到一个二阶列联表。对应的, 有 X^2 服从自由度为 $(r-1)(c-1)$ 的 χ^2 分布

属性 A	属性 B			Total
	1	...	c	
1	C_{11}	...	C_{ic}	R_1
...	...	C_{ij}
r	C_{1r}	...	C_{rc}	R_r
Total	C_1	...	C_c	n

$$X^2 = \sum_{j=1}^r \sum_{i=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \quad E_{ij} = \frac{R_i C_j}{n}$$

9. Kolmogorov-Smirnov Goodness-of-Fit Test, KS 检验

与卡方检验一样, KS 检验同样是一种拟合优度检验方法。KS 检验基于累积概率函数, 因此不适用于定类数据, 当数据定量或连续, 且不服从正态分布时, 可优先选择 KS 检验。KS 检验分为单样本检验与双样本检验, 前者检验一组观测数据是否符合已知的概率分布, 后者比较两组样本是否源于同一分布。因为双样本 KS 检验时对样本分布的位置和形状都敏感, 所以成为两样本比较的最有用且最常用的非参数方法之一。单样本 KS 检验的步骤可概括如下:

- 定义累积概率函数 $F_n(x), F(x)$, 前者为观测序列, 后者为从某一概率分布抽样得到的序列, 作出零假设 $H_0 : F_n(x) = F(x)$;
- 计算 KS 检验统计量 $D_n = \max_{1 \leq i \leq n} \|F_n(i) - F(i)\|$;
- 查表得到 D_n 的临界值 $D_n(\alpha)$, 其中 n 为样本容量, α 为显著水平;
- 若 $D_n < D_n(\alpha)$, 则接受零假设, 即认为概率分布拟合结果满意, 反之拒绝零假设。

两样本检验的步骤基本同上, 差别仅在于此时 $F(x)$ 为另一观测序列的累积概率函数。需要说明的是, 虽然累积分布曲线的形状会随着对数据做转换处理而改变 (如 log 转换), 但是统计量 D_n 的大小是不变的。

14.3 英汉互译

English	Chinese	English	Chinese	English	Chinese
descriptive statistics	描述统计学	population parameters	总体参数	representation	表现
subset	子集	countless	数不尽的	rigorous	严谨的
interquartile range	四分位距	arithmetic mean	算术平均值	succinctly	简洁地
mode	众数	ordinal	序数的	nominal	名义的
probabilistically	从概率上讲的	herein	在此处	preclude	排除
unimodal	单峰的	symmetric	对称的	asymmetric	非对称的
qualitative	定性的	dispersion	分散	denominator	分母
bell-shaped	钟形的	empirical	基于经验的	magnitude	巨大

English	Chinese	English	Chinese	English	Chinese
skewness	偏度	kurtosis	峰度	leptokurtic	distribu- tion
platykurtic distribution	低峰态分布	whereas	然而 (conj)	stem	茎 (n)、来源 (v)
covariance	协方差	drawback	缺点	remedy	修正
aviation	航空	revenue	税收收入	enplanement	航空客流量
deflate	使泄气	monetary	货币的	causality	因果关系
ridership	客运量	sole	仅有的	nonparametric	非参数的
asymptotic	渐近的	depict	描述	pertain	使用 (v)
whisker	(动物的)须	curvilinear	曲线的	account for	占比、对... 负责
deregulation	解除管制	motorist	驾驶员	subsidy	补贴
prespecified	预先确定的	vehicular	车辆的	vehicular speed	行车速度
mound	山丘	commuter	(远距离)通勤者	overtake	超过
assess	评估	repeal	废除 (n, v)	be instrumental in	对... 起重要作用
attribute ... to	把... 归因于	to the extend possible	尽可能	nullify	使无效
null hypothesis	零假设	alternative hypothesis	备择假设	refute	驳斥 (v)
trade off	此消彼长	proper	正确的	at stake	处于危险
stringent	严格的	tire	轮胎	belt	皮带、系 (v)
rear	尾部的、尾部	numerator	分子	essence	本质
absolute value	绝对值	versatility	用途广泛	instance	实例
contingency	可能发生的事	contingency table	列联表	cross-classification	交叉分类
attribute	属性 (n)				

第 15 章

Continuous Dependent Variable Models

15.1 Linear Regression

1. 线性回归模型的基本假设如下：

- (a) 因变量 Y 为连续变量，其中计数变量 (count variables)、定类变量 (nominal variables) 和定序变量 (ordered variables) 都可视为连续变量；
- (b) 自变量 X 与因变量 Y 直接的参数成线性关系；
- (c) 观测值互相独立且随机采样；
- (d) 变量之间存在不确定性关系，包含因变量获取过程中的观测误差和自变量获取中的随机误差，以随机项 ε_i 表示对于样本 i 的偏差；
- (e) 各样本的偏差 ε_i 互相独立（即同方差性 (homoscedasticity)），期望为 0 ($E[\varepsilon_i] = 0$, $D[\varepsilon_i] = \sigma^2$)；
- (f) 各样本的偏差之间不存在自相关 (autocorrelation) 性，即 $Cov[\varepsilon_i, \varepsilon_j] = 0$, $i \neq j$ 。其物理含义为样本取值在个体、时间、空间等维度上相互独立；
- (g) 自变量和偏差之间不存在相关性，即 $Cov[X_i, \varepsilon_j] = 0$, $\forall i, j$ 。具体地解释外生性 (exogeneity) 和内生性 (endogeneity) 的概念：
 - 外生变量指变量的取值不由模型内参数决定。从数学上 $Cov[X_i, \varepsilon_j] = 0$, $\forall i, j$ 即要求模型自变量为外生变量，既不受其它自变量影响，也不受自变量 Y 的影响；
 - 内生变量指变量的取值由模型内参数决定。因变量即属于内生变量。若模型中的关键自变量为内生变量，则需进行数学处理或选择其它模型。
- (h) 偏差近似服从正态分布，即 $\varepsilon_i \sim N(0, \sigma^2)$ 。

2. 最小平方估计 (ordinary least squares estimation, OLS) 是最常用的线性回归模型估计方法，方法寻找一条使得平方误差和 (sum of squared errors, SSE) 最小的直线

$$SSE = \sum_i \left[y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i) \right]^2$$

为最小化上式只需对 β_0, β_1 求偏导，得到 $\beta_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$, $\beta_0 = \bar{y} - \beta_1 \bar{x}$ ；

3. 最大似然近似 (maximum likelihood estimation, MLE) 是另一种著名的模型预测方法，方法最大化估计的似然。基于基本假设，可以得到样本 i 因变量取值 y_i 服从正态分布 $y_i \sim N(X_i \beta, \sigma^2)$ ，则似然 $L(\theta|X)$ 计算为

$$L(\theta|Y) = \prod_i f(y_i|\theta) = (2\pi\sigma^2)^{-\frac{n}{2}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_i (y_i - X_i \beta)^2 \right\} = (2\pi\sigma^2)^{-\frac{n}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (Y - X\beta)^T (Y - X\beta) \right\}$$

上式中 $f(y_i)$ 为概率密度函数；

4. OLS 和 MLE 算法均可得到 β 的无偏、充分估计量 $\hat{\beta}$ 。 $\hat{\beta}$ 服从正态分布 $\hat{\beta}_1 \sim N(\beta_1, \frac{\sigma^2}{\sum(X_i - \bar{X})^2})$ ，其中参数 β_1 的总体方差 σ^2 不可知；
5. 考虑模型的均方误差 MSE，可以构造如下 t 分布量，满足自由度 $n-p$ 、置信度 α 的 t 分布 ($t^* \sim t(\alpha; n-p)$)，其中 n, p 分别为样本量和模型待估计参数量

$$t^* = \frac{\hat{\beta}_i - \beta_i}{\sqrt{\frac{MSE}{\sum_i(X_i - \bar{X})^2}}} \quad MSE = \frac{\sum_i(y_i - \hat{y}_i)^2}{n-p}$$

进一步地可得到 β_i 的置信区间 $\beta_i \pm t(1 - \frac{\alpha}{2}; n-p)s\{\beta_i\}$ ，其中 $s\{\beta_i\}$ 为 β_i 的标准差；

6. 标准回归模型允许通过比较自变量系数比较不同自变量的相对重要性，为此在建模前应该对自变量进行 Z 标准化，使所有自变量期望为 0、方差为 1；
7. 当自变量包含定类变量或定序变量时，需将其转化为指示变量 (indicator variable) 再建模：
- 对于 m 类的定类变量，需将其表示为 $m-1$ 个指示变量，另外一类作为基准；
 - 定序变量是存在相对顺序的，直接建模会使得模型复杂化，同样需要进行处理。因为线性模型假设自变量每增长一个单位具有相同的边际效应 (marginal effect)，然而定序变量不同类别之间的边际效益可能不满足线性关系，直接建模会引入非线性，因此同样需要将其拆解成多个指示变量进行建模。
8. 交互效应 (interaction) 是指模型中一个自变量的取值会对另一个自变量的重要性造成影响的情况。若考虑自变量 X_1, X_2 存在交互效应，则为模型引入交互项 $\beta X_1 X_2$ 。需要说明的是，考虑模型的交互效应不会提高模型的预测性能，但能提升模型的解释能力；
9. 为检验模型变量是否满足线性回归模型的诸多假设，常基于可视化方法定性评价：
- 检验自变量与因变量间是否符合线性假设 以因变量（真实值或拟合值）为横轴、偏差为纵轴，绘制散点图（又称残差图 (residual plot)）。若散点均匀分布于 $y = 0$ 两侧则说明模型自变量与因变量间满足线性假设；
- 确定与自变量存在非线性关系的自变量 以每一自变量取值为横轴、偏差为纵轴，绘制散点图。若散点均匀分布于 $y = 0$ 两侧则说明该自变量与因变量间满足线性假设；
- 检验偏差之间是否符合同方差性 以因变量拟合值为横轴、偏差为纵轴，绘制散点图。若观察到偏差不随拟合值的变化而系统地变化则说明偏差之间满足同方差性。若偏差分布于抛物线、椭圆、三角形等区域内，则可以通过适当地在数学上变化 Y 值或选择其它模型改善这一问题；
- 检验偏差之间是否存在自相关性 以样本获得的时间（空间）为横轴、偏差为纵轴，绘制散点图。若散点均匀分布于 $y = 0$ 两侧则说明偏差之间不存在自相关性；
- 检验偏差是否呈正态分布 可通过计算偏差最小值、四分位数、中位数和最大值等描述性统计量判断；也可通过绘制偏差直方图判断；也可通过绘制 Q-Q 图 (quantile-quantile plot) 判断偏差分布于正态分布间的关系；或通过卡方拟合优度检验或 Kolmogorov-Smirnov 拟合优度检验等非参数方法。
10. 残差图还可用于离群值检验。在假设偏差呈正态分布的前提下，可基于 3σ 准则识别离群值。离群值可能源于设定错误（如模型本身无法捕捉某些信息）、编码错误、数据采集错误、计算错误等原因。检测得离群值后需研究其产生原因。若确定其是由错误导致的则需对其进行舍去或修正，若无法判断离群的原因则需要保留样本不能随意修改；
11. 拟合优度 (goodness-of-fit, GOF) 是模型拟合程度的度量。 R^2 和 R^2_{adj} (修正 R^2 (R² adjusted measures)) 是回归模型常用的拟合优度指标，其物理含义为模型在多大程度上捕捉了因变量的变异性，越大则模型拟合效果越好，具体数学定义为

$$R^2 = \frac{SST - SSE}{SST} = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \quad R^2_{adj} = 1 - \frac{\frac{SSE}{n-p}}{\frac{SST}{n-1}} = 1 - \frac{n-1}{n-p} \frac{SSE}{SST}$$

上式中 $SSE = \sum_i(y_i - \hat{y}_i)^2$ 表示误差平方和 (sum of square errors)； $SSR = \sum_i(\hat{y}_i - \bar{y})^2$ 表示回归平方和 (regression sum of square)； $SST = \sum_i(y_i - \bar{y})^2$ 表示总体平方和 (total sum of square)。 R^2

和 R_{adj}^2 仅适用于同相同场景下的已有模型的结果比较，仅靠其也不足以判断模型的质量，因为较高的拟合度可能是样本的人工修正造成的；

12. 当自变量间存在多重共线性 (multicollinearity) 时（即自变量间存在精确相关性或高度相关性）一方面可能影响建模结果，另一方面也使得变量标准化不适用，因为对一个变量的变换会改变另一个变量的值。为此在建模前需要进行特征筛选去除存在强线性相关的变量：
 - (a) 首先对每一候选自变量单独建模（模型仅包含一个自变量）并判断其显著性，得到显著变量的集合；
 - (b) 基于相关系数判断变量之间的线性相关性，从显著变量集中剔除强相关变量。剔除依据为单独建模时的 R^2 大小，对应较小 R^2 的变量将被剔除。

15.1.1 递推最小二乘 (Recursive least square, RLS)

1. 对于样本特征集 $X_k = [x_1, \dots, x_k]^\top$ (x_i 为列向量) 与样本标签集 $Y_k = [y_1, \dots, y_k]^\top$ (y_i 为标量)，基于最小二乘法估计线性回归系数 β_k ，显然有

$$\beta_k = \arg \min_{\beta} f = (Y_k - X_k \beta)^\top (Y_k - X_k \beta) \implies \frac{\partial f}{\partial \beta_k} = -2X_k^\top (Y_k - X_k \beta_k) = 0 \implies \beta_k = (X_k^\top X_k)^{-1} X_k^\top Y_k$$

然而当数据不断在线输入时，重复基于上式估计 β 意味着需要存储全部历史数据，且随着数据量的增加对空间和计算资源的消耗也愈加严重¹；

2. 递推最小二乘即由此出发，将最小二乘估计过程改写为递推形式，从而对于新数据 x_{k+1}, y_{k+1} ，仅需结合 β_k 即可更新 β_{k+1} 。进一步推导最小二乘估计的递推形式，注意到 $X_{k+1} = [X_k^\top, x_{k+1}]^\top$, $Y_{k+1} = [Y_k^\top, y_{k+1}]^\top$ ，则

$$\begin{aligned} X_{k+1}^\top X_{k+1} &= X_k^\top X_k + x_{k+1} x_{k+1}^\top, & X_{k+1}^\top Y_{k+1} &= X_k^\top Y_k + x_{k+1} y_{k+1} \\ \implies \beta_{k+1} &= (X_{k+1}^\top X_{k+1})^{-1} X_{k+1}^\top Y_{k+1} = (X_{k+1}^\top X_{k+1})^{-1} (X_k^\top Y_k + x_{k+1} y_{k+1}) = (X_{k+1}^\top X_{k+1})^{-1} (X_k^\top X_k \beta_k + x_{k+1} y_{k+1}) \end{aligned}$$

可以看到，上式实际上即为最小二乘估计的基本递推形式

$$\beta_{k+1} = P_{k+1} (P_k^{-1} \beta_k + x_{k+1} y_{k+1}), \quad P_{k+1} = (P_k^{-1} + x_{k+1} x_{k+1}^\top)^{-1}, \quad P_k^{-1} = X_k^\top X_k$$

基于上式递推时每一轮迭代除了 $x_{k+1}, y_{k+1}, \beta_k$ 外还需保存 P_k ，但因为 P_k 的尺寸并未随样本量增加而增加，故递推更新 β_{k+1} 可避免内存过度消耗问题，但每一轮迭代时均需要矩阵求逆，计算效率偏低；

3. 进一步地尝试消去上述递推公式中的逆运算。代入 $P_k^{-1} = P_{k+1}^{-1} - x_{k+1} x_{k+1}^\top$ 有

$$\beta_{k+1} = P_{k+1} [(P_{k+1}^{-1} - x_{k+1} x_{k+1}^\top) \beta_k + x_{k+1} y_{k+1}] = \beta_k + P_{k+1} x_{k+1} (y_{k+1} - x_{k+1}^\top \beta_k), \quad P_{k+1} = (P_k^{-1} + x_{k+1} x_{k+1}^\top)^{-1}$$

并基于矩阵求逆引理 (matrix inversion lemma) $(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$ ，且注意到 $x_{k+1}^\top P_k x_{k+1}$ 为标量

$$P_{k+1} = (P_k^{-1} + x_{k+1} x_{k+1}^\top)^{-1} = P_k - P_k x_{k+1} (I + x_{k+1}^\top P_k x_{k+1})^{-1} x_{k+1}^\top P_k = P_k - \frac{P_k x_{k+1} x_{k+1}^\top P_k}{1 + x_{k+1}^\top P_k x_{k+1}}$$

从而得到最终的递推公式

$$\beta_{k+1} = \beta_k + P_{k+1} x_{k+1} (y_{k+1} - x_{k+1}^\top \beta_k), \quad P_{k+1} = P_k - \frac{P_k x_{k+1} x_{k+1}^\top P_k}{1 + x_{k+1}^\top P_k x_{k+1}}, \quad (\text{RLS})$$

按上式迭代更新 β_{k+1} ，只需在计算初始 $P_{k_0} = (X_{k_0}^\top X_{k_0})^{-1}$ 时进行求逆运算，后续运算仅需基本的四则运算，计算效率大大提升。

15.2 Latent Variable Models, 隐变量模型

15.2.1 Principal component analysis

1. 主成分分析适用于定距和定比数据，其主要目的为：1) 压缩数据集；2) 更好地理解数据。方法对原始变量进行线性组合构造新的变量，以较少的新变量部分或完全替代原始变量，同时保持对原始数据方差的解释能力；

¹递推最小二乘法推导 (RLS) ——全网最简单易懂的推导过程: <https://zhuanlan.zhihu.com/p/111758532>

2. 注意到只有当原样本存在线性相关的特征时主成分分析才能起到压缩特征的效果;
3. 主成分分析不属于统计学模型, 方法中不存在自变量和因变量的区分²;
4. 进一步地介绍主成分的提取方法。考虑一个包含 n 个样本 p 个特征的矩阵 X (为避免度量单位的影响对每一特征样本进行标准化), 假设欲从中提取 $K < p$ 个主成分, 则:
 - (a) 对于第一个主成分 Z_1 , 记其为所有 p 的特征的线性组合。注意到主成分分析的思路是令主成分最大限度地表现原样本的方差, 可以得到如下优化目标求解 Z_1

$$\max \text{VAR}[Z_1] = \max \text{VAR}[a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_p] \quad \text{s.t. } a_{11}^2 + a_{12}^2 + \cdots + a_{1p}^2 = 1$$

约束条件限制了系数的取值范围, 从而使得目标函数可最优化;

- (b) 对于第二个主成分 Z_2 , 采用基本相同的优化思路, 不同之处为使得 Z_2 与 Z_1 线性无关 (即确保 Z_2 所能解释的方差不来源于 Z_1), 新增约束条件 $\text{COV}[Z_1, Z_2] = 0$;
- (c) 对于后续主成分同理, 要求 $\text{COV}[Z_1, Z_2, \dots, Z_K] = 0$ 。
5. 以向量形式重新表示以上过程。对于第一个主成分 Z_1 , 有

$$Z_1 = Xa_1 \quad a_1^T a_1 = 1$$

则基于方差定义, 且考虑到 X 经过标准化有 $\mathbb{E}[Xa_1] = 0$, 则

$$\text{VAR}[Z_1] = \text{VAR}[Xa_1] = \mathbb{E}[(Xa_1)^T(Xa_1)] - (\mathbb{E}[Xa_1])^2 = \mathbb{E}[a_1^T X^T X a_1] = \frac{1}{n-1} a_1^T X^T X a_1$$

注意到 X 的协方差矩阵 $s^2[X]$ 定义为 (同样考虑到 X 经过标准化有 $\mathbb{E}[X] = 0$)

$$s^2[X] = \text{COV}[X, X] = \mathbb{E}[(X - \mathbb{E}[X])^T(X - \mathbb{E}[X])] = \mathbb{E}[X^T X] = \frac{1}{n-1} X^T X$$

则有

$$\text{VAR}[Z_1] = \frac{1}{n-1} a_1^T X^T X a_1 = a_1^T s^2[X] a_1$$

基于拉格朗日乘数法, 得到无约束的最优化函数如下

$$\max L = \max \{a_1^T s^2[X] a_1 - \lambda(a_1^T a_1 - 1)\}$$

上式对 a_1 求偏导解得极值点

$$\frac{\partial L}{\partial a_1} = 0 \implies 2s^2[X]a_1 - 2\lambda a_1 = 0 \implies s^2[X]a_1 = \lambda a_1$$

显然优化得到的 a_1 为 X 的协方差矩阵 $s^2[X]$ 的一个特征向量, 相应的主成分 Z_1 与协方差矩阵 $s^2[X]$ 的一个特征值对应。进一步地可以推广得, X 的所有 p 个主成分与其协方差矩阵的 p 个特征值一一对应, 特征值的大小反映了相应主成分所能捕捉的原样本方差的大小。对于第 j 个主成分 Z_j , 即其对应的特征值为 λ_j , 则其所能捕捉的样本方差占原样本总方差的比值为 $\frac{\lambda_j}{\sum_j \lambda_j}$;

6. 主成分分析可以得到主成分矩阵 A , 元素 a_{ij} 可以表示主成分 Z_i 所包含的特征 X_j 的信息 (方差) 的多少。但因为提取主成分时仅考虑其所反映的原样本信息量, 所以得到的每个主成分普遍包含了大部分原始特征的信息, 使得无法准确地判断出每一主成分所能代表的原始特征。

15.2.2 Factor analysis

1. 因子分析与主成分分析具有很高的相关性。因子分析可视为主成分分析的扩展, 其目的是提取几个不可观察的因子以描述多个变量之间的协方差。在提取公因子时, 不仅注意变量之间是否相关, 而且考虑相关关系的强弱, 使得提取出来的公因子不仅起到降维的作用, 而且能够被很好地解释;
2. 不同于主成分分析, 因子分析属于统计模型, 存在自变量和因变量之分。但因子分析和主成分一样均基于样本的协方差矩阵, 且均适用于定距数据和定比数据;

²在机器学习的领域中属于无监督学习, 样本无标签

3. 因子分析模型如下

$$X_1 - \mu_1 = l_{11}F_1 + l_{12}F_2 + \cdots + l_{1m}F_m + \varepsilon_1$$

$$X_2 - \mu_2 = l_{21}F_1 + l_{22}F_2 + \cdots + l_{2m}F_m + \varepsilon_2$$

$$\vdots$$

$$X_p - \mu_p = l_{p1}F_1 + l_{p2}F_2 + \cdots + l_{pm}F_m + \varepsilon_p$$

上式中 F 为因子; $l_{ij} < 1$ 为因子载荷 (factor loading), 因子载荷越接近 1 说明 X_i 受因子 F_j 的影响越大, 理想情况下因子载荷应该尽可能分散, 以尽可能体现与每一因子关联最密切的变量; 误差项 ε_i 仅与随机变量 X_i 有关;

4. 注意到因子分析模型中包含 p 个方程, 而有 $m + p$ 个未知量, 因此无法直接求解 F, l , 须通过因子旋转法 (factor rotation method) 增加约束:

- 正交旋转 (orthogonal rotation) 可以得到互相正交的因子, 模型新增以下约束:

- F, ε 相互独立;
- $\mathbb{E}[F] = 0, \mathbb{E}[\varepsilon] = 0$;
- $COV[F] = I, COV[\varepsilon] = \mathbf{v}$, 其中 I, \mathbf{v} 分别为单位阵和对角阵。

为得到尽可能分散的因子载荷, 以最大化因子载荷的总方差为目标, 称为方差最大旋转法 (varimax rotation);

- 非正交因子分析 (oblique factor analysis) 放宽了对因子载荷不相关的约束, 从而得到非正交的因子。因为放宽约束, 非正交因子分析可能可以得到解释性更强的结构, 但也因为交叉载荷 (cross loading) 的存在有时会得到意义不明显的因子。

5. 因子分析可以得到载荷矩阵 L , 相比于主成分矩阵 A , 前者更加稀疏, 从而反映出每一因子能用哪些载荷大的原始变量来解释。

15.2.3 Structural equation modelling

1. 结构方程模型是测量模型的自然延伸, 是一种成熟的统计建模技术。模型可应对很多复杂的建模难题, 包括隐变量与外生变量、变量间的内生性等等。图结构是主要的表示结构方程模型的方法:

- 模型显式地将测量误差纳入建模框架以解决潜在的问题;
- 不同于传统回归分析, 模型可以以潜变量作为因变量;
- 不同于普通回归模型隐式地建模相关性, 模型显式地建模直接、间接和相关关系;
- 模型需要协方差矩阵所含信息。

2. 结构方程模型由测量模型 (measurement model) 和结构模型 (structural model) 两部分组成。前者常由经典的因子分析模型实现, 解析如何基于各种被测外生变量观测潜变量并估计外生变量测量误差; 后者为一联立方程组, 以建模多个自变量与多个因变量之间的相互联系;

3. 结构方程模型中涉及两种变量: 显变量 (可观测变量, 由矩形框表示) 和潜变量 (不可观测变量, 由椭圆形框表示), 后者类似于因子分析中的因子。模型中的变量又可分为外生变量和内生变量³;

4. 模型中需区分两种参数 (路径系数): 固定参数与自由参数 (free parameter)。前者由研究者预设 (一般只取 0 或 1), 后者基于数据集估值;

5. 记 $\boldsymbol{\eta}_{n \times p}$ 表示研究的因变量, 其中 n, p 分别表示样本数和因变量特征数 (结构方程中测量模型的部分可实现对多因变量的回归)。记观测的因变量的协方差矩阵为 $\mathbf{S}_{p \times p}$ 。现假设 $\boldsymbol{\eta}$ 与自变量 $\boldsymbol{\xi}$ 服从如下关系, 其中 β, γ 为路径系数, 分别表示因变量间的影响和自变量对因变量的影响; $\boldsymbol{\varepsilon}$ 为拟合残差

$$\boldsymbol{\eta} = \boldsymbol{\beta}\boldsymbol{\eta} + \boldsymbol{\gamma}\boldsymbol{\xi} + \boldsymbol{\varepsilon} \iff \boldsymbol{\eta} = (\mathbf{I} - \boldsymbol{\beta})^{-1}(\boldsymbol{\gamma}\boldsymbol{\xi} + \boldsymbol{\varepsilon})$$

显然对于任意给定的路径系数都可以基于自变量 $\boldsymbol{\xi}$ 得到相应的因变量估值 $\hat{\boldsymbol{\eta}}$, 同时可以求得估计的因变

³ 判断显变量与潜变量属于内生或外生变量的方法存在差异。对于潜变量, 若没有箭头指向它, 则它属于内生潜变量, 否则则属于外生潜变量。显变量的内生性则由与它连接的潜变量决定。若其中存在内生潜变量, 则相应变量属于内生显变量, 否则属于外生显变量。

量的协方差矩阵 Σ 。为使得上式的拟合效果最佳，结构方程模型的思路是使得估计的协方差矩阵 Σ 与实际观测的协方差矩阵 S 最接近；

6. 推导 Σ 的解析形式。可以证明

$$COV(Au, Bv) = A COV(u, v) B^T$$

因此有

$$\Sigma = \text{VAR}[\eta] = COV((I - \beta)^{-1}(\gamma\xi + \epsilon), (I - \beta)^{-1}(\gamma\xi + \epsilon)) = (I - \beta)^{-1} COV(\gamma\xi + \epsilon, \gamma\xi + \epsilon) (I - \beta)^{-1T}$$

对自变量标准化后有 $E[\xi] = 0$ ，并假设误差服从 $E[\epsilon] = 0$ ，则上式中

$$COV(\gamma\xi + \epsilon, \gamma\xi + \epsilon) = COV(\gamma\xi, \gamma\xi) + \text{VAR}[\epsilon] = \gamma\Phi\gamma^T + \Psi$$

上式中 Φ, Ψ 分别为自变量和误差项的协方差矩阵。则 Σ 的解析式为

$$\Sigma = (I - \beta)^{-1} (\gamma\Phi\gamma^T + \Psi) (I - \beta)^{-1T}$$

7. 为估计参数 β, γ ，令 $\Sigma = S$ ，可以得到一组联立方程。尽管矩阵维度为 $p \times p$ ，但因为协方差矩阵为对称阵，因此只能得到 $p^* = p + C_p^2 = \frac{p(p+1)}{2}$ 组独立的方程，当且仅当模型未知参数（自由参数）数量 $Q \leq p^*$ 时模型才可解，以上也是结构方程模型可被识别 (identification) 的充要条件；
8. 对于上述联立方程组求解问题，在图结构较简单时可基于最小二乘法估计参数，此时称为路径分析，而在更一般的模式下常基于最大似然估计参数，此时称为结构方程模型。定义函数 $F(S, \Sigma)$ 量化估计方差矩阵与实际方差矩阵的距离。在结构方程模型中， F 定义为对数似然 $LL(\Sigma|S)$ 与对数似然 $LL(\Sigma = S|S)$ 的差。首先推导 $LL(\Sigma|S)$ 。假设对于任意样本 l 有 $\eta_l \sim N(0, \Sigma)$ ，则有

$$\begin{aligned} \Pr(\eta_l = \eta_i | \Sigma) &= \sqrt{\frac{|\Sigma^{-1}|}{(2\pi)^p}} \exp \left\{ -\frac{1}{2} \eta_i^T \Sigma^{-1} \eta_i \right\} \\ \implies LL(\Sigma|\eta) &= \sum_i^n \ln \Pr(\eta_l = \eta_i | \Sigma) = \frac{n}{2} \ln |\Sigma^{-1}| - \frac{np}{2} \ln 2\pi - \frac{1}{2} \sum_i^n \eta_i^T \Sigma^{-1} \eta_i \end{aligned}$$

注意到上式中 $\eta_i^T \Sigma^{-1} \eta_i$ 的计算结果为一个数，自然地有 $\eta_i^T \Sigma^{-1} \eta_i = \text{trace}(\eta_i^T \Sigma^{-1} \eta_i)$ ，则

$$\begin{aligned} LL(\Sigma|\eta) &= \frac{n}{2} \ln |\Sigma^{-1}| - \frac{np}{2} \ln 2\pi - \frac{1}{2} \sum_i^n \text{trace}(\eta_i^T \Sigma^{-1} \eta_i) \\ &= \frac{n}{2} \ln |\Sigma^{-1}| - \frac{np}{2} \ln 2\pi - \frac{1}{2} \text{trace} \left(\sum_i^n \eta_i^T \Sigma^{-1} \eta_i \right) \\ &= \frac{n}{2} \ln |\Sigma^{-1}| - \frac{np}{2} \ln 2\pi - \frac{1}{2} \text{trace} \left(\left(\sum_i^n \eta_i \eta_i^T \right) \Sigma^{-1} \right) \\ &= \frac{n}{2} \ln |\Sigma^{-1}| - \frac{np}{2} \ln 2\pi - \frac{n}{2} \text{trace}(S\Sigma^{-1}) = LL(\Sigma|S) \end{aligned}$$

对上式令 $\Sigma = S$ ，显然有

$$LL(\Sigma = S|S) = \frac{n}{2} \ln |S^{-1}| - \frac{np}{2} \ln 2\pi - \frac{n}{2} \text{trace}(S\Sigma^{-1}) = S|S) = \frac{n}{2} \ln |S^{-1}| - \frac{np}{2} \ln 2\pi - \frac{n}{2} p$$

因此可以计算 F_{MLE} 如下

$$F_{MLE} = LL(\Sigma|S) - LL(\Sigma = S|S) = \frac{n}{2} \ln |\Sigma^{-1}| - \frac{n}{2} \ln |S^{-1}| - \frac{n}{2} \text{trace}(S\Sigma^{-1}) + \frac{np}{2}$$

舍去常系数简化得

$$F_{MLE} = \ln |\Sigma^{-1}| - \ln |S^{-1}| - \text{trace}(S\Sigma^{-1}) + p$$

9. 在 F_{MLE} 的基础上可定义卡方统计量 $X^2 = (n - 1) \times F_{MLE} \sim \chi^2(\alpha, p^* - Q)$, 该统计量可用于结构方程模型适配度 (goodness-of-fit, GOF) 评价:

- X^2 : 该卡方统计量可用于判断零假设 $\Sigma = S$, 若在样本数介于 100 至 200 之间时有 $p > 0.05$ (即无法拒绝原假设), 则模型可接受;
- 简洁率 (parsimony rate, PR): 参数越少 (自由度越大) 则越简洁。定义模型自由度 $d = p^* - Q$, 则有

$$PR = \frac{d}{d_i}$$

d_i 为独立模型 (independence model), 即不存在任何参数模型中所有变量互相独立的模型的自由度;

- X^2/d : 经验表明该统计量应小于 5, 且最好接近 1;
- RMSEA (root mean square error of hypothesis test): < 0.05 意味着适配良好, < 0.08 意味着适配合理;
- 标准拟合指数 (normed fit index, NFI): 将模型与基准模型比较, $NFI < 0.9$ 意味着模型可以继续改进

$$NFI = 1 - \frac{X^2}{X_b^2}$$

X_b^2 表示基准模型的卡方值。常用的基准模型为独立模型或饱和模型 (saturated model), 即考虑所有变量互相影响的模型。

15.3 英汉互译

English	Chinese	English	Chinese	English	Chinese
homoscedasticity	同方差性	autocorrelation	自相关	exogeneity	外生性
endogeneity	内生性	heteroscedasticity	异方差性	multiplicative	乘法的
misspecification	设定错误	multicollinearity	多重共线性	counterintuitive	违反直觉的
parsimonious	过分节俭的	orthogonal	正交的	oblique	非正交的
parsimony	简洁性				

第 16 章

Count and Discrete Dependent Variable Models

线性模型的重要假设之一即为偏差服从正态分布。然而对于计数数据（包括二元计数和多元计数），其分布恒正且存在强烈的偏度，此时不能假设偏差服从正态分布。广义线性模型扩展了线性模型的应用范围，适用于拟合服从更一般分布（指数族分布）的变量，包括正态分布、二项分布、负二项分布、伽马分布等等。

16.1 Count Data Models

16.1.1 Poisson regression model

1. 给点时间内的事故计数为典型的泊松分布数据。记交叉口 i 的事故数为 y_i ，则在泊松回归模型中，有

$$P(y_i) = \frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!}$$

其中 λ_i 为交叉口 i 的泊松系数，其物理意义为该交叉口的期望事故数，有 $\mathbb{E}[y_i] = \text{VAR}[y_i] = \lambda_i$ 。显然泊松回归的目的即是为了求解参数 λ_i 。假设 λ_i 与解释变量 X_i 间服从线性关系，将问题转化为求解以下线性回归模型（加上指数相是为了使得 $\lambda_i \geq 0$ ），并基于极大似然估计求解

$$\lambda_i = \exp\{\beta X_i\} \implies LL(\beta) = \sum_i [-\exp\{\beta X_i\} + y_i \beta X_i - \ln(y_i!)]$$

2. 定义弹性 (elasticity) 的概念以深入了解参数估计结果的含义。弹性定义为自变量变化 1% 而引起的因变量变化的百分比，度量一个变量的相对变化关于另一个变量相对变化的敏感程度，在数学上定义为 $E_i = \frac{\partial Y/Y}{\partial x_i/x_i}$ ，代入 λ_i 的回归公式，可以得到样本 i 的相关变量 x_{ik} 对 λ_i 的弹性效益 $E_{x_{ik}}^{\lambda_i}$ 为

$$E_{x_{ik}}^{\lambda_i} = \frac{\partial \lambda_i / \lambda_i}{\partial x_{ik} / x_{ik}} = \frac{\partial \lambda_i}{\partial x_{ik}} \frac{x_{ik}}{\lambda_i} = \beta_k x_{ik}$$

若变量 X_k 为指示变量，则以差分替代微分，得到伪弹性计算公式 $E_{x_{ik}}^{\lambda_i} = \frac{\exp\{\beta_k\} - 1}{\exp\{\beta_k\}}$ 。变量 X_k 对所有观测值的弹性均值即为变量 X_k 的弹性；

3. 泊松回归模型同样有相应的拟合优度统计量：

- 似然比检验是常用的评价两嵌套模型的方法。似然比 (likelihood ratio) 为服从卡方分布的统计量，定义为无约束模型 U （复杂模型）与有约束模型 R （简单模型）的对数似然之差，即 $-2[LL(\beta_R) - LL(\beta_U)] \sim \chi^2$ 。卡方分布的自由度为两个模型的参数数量差值；
- G^2 定义为模型总偏差 (sum of model deviances)，当模型完美预测时 $G^2 = 0$

$$G^2 = 2 \sum_i y_i \ln \frac{y_i}{\hat{\lambda}_i}$$

- 因为泊松回归的残差不满足同方差性且条件均值 (conditional mean) $\mathbb{E}[y|X]$ 不满足线性, 无法直接计算 R^2 , 但可定义类似的统计量 R_p^2

$$R_p^2 = 1 - \frac{\sum_i \left(\frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i}} \right)^2}{\sum_i \left(\frac{y_i - \bar{y}}{\sqrt{\bar{y}}} \right)^2}$$

上式中分子类似于平方误差和 (SSE), 分母类似于总平方和 (SST);

- ρ^2 是另一个度量模型拟合度的统计量, 取值范围介于 0-1 之间, $\rho^2 = 1$ 表示完美预测。下式中 $LL(0)$ 表示初始模型的对数似然值, 此时模型中所有参数均为 0。

$$\rho^2 = 1 - \frac{LL(\beta)}{LL(0)}$$

16.1.2 Negative binomial model

1. 泊松分布的重要假设即为 $\mathbb{E}[y_i] = VAR[y_i]$, 然而实际场景下采集的数据不一定满足上述关系。称 $\mathbb{E}[y_i] > VAR[y_i]$ 和 $\mathbb{E}[y_i] < VAR[y_i]$ 分别为欠分散 (under dispersed) 和过分散 (over dispersed)。对于后者, 应采用负二项分布模型 (negative binomial model);
2. 基于泊松模型的拟合函数, 若将 λ_i 与自变量 X_i 的关系拓展为如下形式, 则泊松模型即可转化为负二项分布模型

$$\lambda_i = \exp\{\beta X_i + \varepsilon_i\}$$

上式中误差项 ε_i 服从均值为 1、方差为 α^2 的伽马分布。显然当 $\alpha = 0$ 时模型退化为泊松模型。负二项模型的概率分布 $P(y_i)$ 和方差 $VAR[y_i]$ 分别为

$$P(y_i) = \frac{\Gamma(y_i + \alpha^{-1})}{y_i! \Gamma(\alpha^{-1})} \left(\frac{\alpha \lambda_i}{1 + \alpha \lambda_i} \right)^{y_i} \left(\frac{1}{1 + \alpha \lambda_i} \right)^{\alpha^{-1}} \quad VAR[y_i] = \mathbb{E}[y_i] [1 + \alpha \mathbb{E}[y_i]] = \mathbb{E}[y_i] + \alpha \mathbb{E}[y_i]^2$$

进一步地可以得到负二项分布的似然函数 $L(\lambda_i)$ 为

$$L(\lambda_i) = \prod_i \frac{\Gamma(y_i + \alpha^{-1})}{y_i! \Gamma(\alpha^{-1})} \left(\frac{\alpha \lambda_i}{1 + \alpha \lambda_i} \right)^{y_i} \left(\frac{1}{1 + \alpha \lambda_i} \right)^{\alpha^{-1}}$$

3. 负二项模型中参数 α 表示了样本的过分散程度, α 越大样本越分散, 模型参数估值的标准差也越大。

16.1.3 Zero-inflated Poisson and negative binomial regression models

1. 在数据采集时可能存在零膨胀 (zero-inflated) 现象 (详见第2*II节), 即数据中的零值来源于两种情况, 一种是在观测期间内未观测到样本, 另一种是根本不存在观测到样本的可能性;
2. 称根本不存在观测到样本的可能性的状态为零状态, 并对于样本 i 记零状态的概率为 p_i , 则可构建相应的零膨胀泊松模型 (zero-inflated Poisson, ZIP) 和零膨胀负二项分布模型 (zero-inflated negative binomial, ZINB)

$$ZIP : P(y_i) = \begin{cases} P(y_i = 0) = p_i + (1 - p_i) \exp\{-\lambda_i\} \\ P(y_i = y) = \frac{(1 - p_i) \exp\{-\lambda_i\} \lambda_i^y}{y!} \end{cases} \quad ZINB : P(y_i) = \begin{cases} P(y_i = 0) = p_i + (1 - p_i) \left[\frac{1/\alpha}{1/\alpha + \lambda_i} \right]^{1/\alpha} \\ P(y_i = y) = (1 - p_i) \frac{\Gamma(1/\alpha + y) u_i^{1/\alpha} (1 - u_i)^y}{\Gamma(1/\alpha) y!} \end{cases}$$

上式中 $u_i = 1/\alpha(1/\alpha + \lambda_i)$ 。可基于最大似然近似求解上述模型参数估值;

3. 为判断是否需要采用零膨胀模型, Vuong 提出 Vuong 检验方法 (详见第2*V节)。方法可以比较两种非嵌套模型 (nonnested model) 的拟合程度。构造统计量 V 如下

$$V = \frac{\sqrt{N} \left(\frac{1}{N} \sum_1^N m_i \right)}{\sqrt{\frac{1}{N} \sum_1^N (m_i - \bar{m})^2}} = \frac{\sqrt{N} \cdot \bar{m}}{S_m} \quad m_i = \ln \frac{f_1(y_i|X_i)}{f_2(y_i|X_i)}$$

上式中 f_1, f_2 分别表示两模型的概率密度函数, \bar{m} 为 m_i 的均值。 V 近似服从标准正态分布, 在双侧显著水平为 0.05 的条件下, 当 $V \geq 1.96$, 选择模型 1; 当 $V \leq -1.96$, 选择模型 2; 当 $-1.96 < V < 1.96$, 说明 vuong 检验不支持任何一个模型。

16.1.4 Random-effects count models

1. 收到空间相关性或时间相关性的影响, 在某些情况下观测样本间可能存在相关性 (correlation), 即处于同一空间或时间区间内的样本受到相似的影响, 随机效用模型 (random effects model) 有助于描述这一相关性;
2. 对样本分组, 认为相同组内的样本受到相似的影响, 则以 Poisson 模型为例引入随机效应如下

$$\lambda_{ij} = \exp\{\beta X_{ij}\} \exp\{\eta_j\} \iff \ln \lambda_{ij} = \beta X_{ij} + \eta_j$$

上式中 λ_{ij} 表示第 j 组的样本 i 的期望观测值。 η_j 表示第 j 组的随机效用。相应的随机效用 Poisson 模型如下

$$P(y_{ij}|X_{ij}, \eta_j) = \frac{\exp\{-\exp\{\beta X_{ij}\} \exp\{\eta_j\}\} [\exp\{\beta X_{ij}\} \exp\{\eta_j\}]^{y_i}}{y_{ij}!}$$

一般认为各组的 η_j 随机分布且 $\exp\{\eta_j\}$ 服从均值为 1 方差为 α 的伽马分布。

16.2 Logistic Regression

1. Logit 为因变量取 1 的似然比 (likelihood ratio) 或几率 (odds) 的对数值, 数学形式为

$$Y = \text{logit}(P) = \ln \frac{P_i}{1 - P_i} = \beta X_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} \implies P(y_i = 1) = \frac{e^{\beta X_i}}{1 + e^{\beta X_i}} = \frac{1}{1 + e^{-\beta X_i}}$$

同样地, 可基于最大似然近似求解模型参数估值, 有

$$L(\beta) = \prod_i \left[\left(\frac{1}{1 + e^{-\beta X_i}} \right)^{y_i} \left(1 - \frac{1}{1 + e^{-\beta X_i}} \right)^{1-y_i} \right]$$

2. 观察系数 β_k 的数学含义, 表示当相应自变量 x_{ik} 增长一个单位时几率 $P_i/(1 - P_i)$ 的对数值 $\ln[P_i/(1 - P_i)]$ 的改变量。定义 e^{β_k} 为几率比 (odds ratio, OR), 几率比大于 1 表示因变量 y 与自变量 X_k 正相关, 小于 1 表示负相关;
3. 同样地可以基于弹性的定义计算样本 i 的相关变量 x_{ik} 对 P_i 的弹性效益 $E_{x_{ik}}^{P_i}$ 为

$$E_{x_{ik}}^{P_i} = \frac{\partial P_i / P_i}{\partial x_{ik} / x_{ik}} = \frac{\partial P_i}{\partial x_{ik}} \frac{x_{ik}}{P_i} = (1 - P_i) \beta_k x_{ik}$$

变量 X_k 对所有观测值的弹性均值即为变量 X_k 的弹性;

4. logistics 回归模型同样有相应的拟合优度统计量:

- 似然比检验是常用的评价两嵌套模型的方法。似然比 (likelihood ratio) 为服从卡方分布的统计量, 定义为无约束模型 U (复杂模型) 与有约束模型 R (简单模型) 的对数似然之差, 即 $-2[LL(\beta_R) - LL(\beta_U)] \sim \chi^2$ 。卡方分布的自由度为两个模型的参数数量差值;
- ρ^2 是另一个度量模型拟合度的统计量, 取值范围介于 0-1 之间, $\rho^2 = 1$ 表示完美预测。下式中 $LL(0)$ 表示初始模型的对数似然值, 此时模型中所有参数均为 0。

$$\rho^2 = 1 - \frac{LL(\beta)}{LL(0)}$$

- 另外因为 logistics 回归模型为二值变量回归模型, 因此也可基于 ROC 曲线和 AUC 评价其拟合效果。

16.3 英汉互译

English	Chinese	English	Chinese	English	Chinese
plausibility	合理性				

赌书消得泼茶香 当时只道是寻常

第六部分

数学工具

第 17 章

运筹学

17.1 线性规划

1. 在本节中，定义线性规划的标准形式如下：

$$\begin{aligned} \min f &= \sum_{j=1}^n c_j x_j \\ \text{s.t. } &\sum_{j=1}^n a_{ij} x_j = b_i \quad (i = 1, \dots, m, b_i \geq 0) \\ &x_j \geq 0 \quad (j = 1, \dots, n) \end{aligned} \tag{17.1}$$

即要求目标函数求最小，约束条件为等值约束，决策变量非负，约束右端项常数非负。进一步地，可将上式写成矩阵式：记矩阵 A 是 $m \times n$ 矩阵，向量 c, x, b 均为列向量，其中 c, x 为 $n \times 1$ ， b 为 $m \times 1$

$$\begin{aligned} \min f &= c^T x \\ \text{s.t. } &Ax = b \quad (i = 1, \dots, m, b_i \geq 0) \\ &x \geq 0 \quad (j = 1, \dots, n) \end{aligned} \tag{17.2}$$

另外还可写成向量式，仅需将系数矩阵拆成 n 个列向量的组合 $A = (p_1, \dots, p_n)$ ，并将约束条件改写为

$$x_1 p_1 + \dots + x_n p_n = b \quad p_j = (a_{1j}, \dots, a_{mj})^T$$

2. 接下来给出线性规划问题解的若干定义：

可行解 称满足线性规划问题全部约束条件（包括等值约束和决策变量非负）的解为可行解 (**feasible solution**)，称所有可行解构成的集合为可行域 (**feasible region**)；

最优解 使得目标函数取得最小的解为最优解；

基解 基解 (**basic solution**) 的概念与基 (**basis**) 的概念有关，首先介绍基的概念。考虑等值约束方程组 $Ax = b$ ，假设系数矩阵 $A = (p_1, \dots, p_n)$ 为 $m \times n$ 的行满秩阵，则从 n 项列向量中取出任意线性无关的 m 项组成 $m \times m$ 方阵 B ，不失一般性地假设 $B = (p_1, \dots, p_m)$ ，则称 B 是该线性规划问题的一个基，对应的列向量 p_j ($j = 1, \dots, m$) 和变量 x_j ($j = 1, \dots, m$) 分别称为基向量和基变量 (**basic variable**)，全体基变量可组成列向量 x_B ($j = 1, \dots, m$)。将 A, x 中剩下的部分分别记为 N, x_N ，则

$$A = \begin{pmatrix} B & N \end{pmatrix} \quad x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$$

因此可以将等值约束改写为如下形式

$$Ax = b \implies Bx_B + Nx_N = b \implies Bx_B = b - Nx_N$$

又因为 B 中列向量线性无关，则 B 为非奇异矩阵，即可逆

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \quad (17.3)$$

令 $\mathbf{x}_N = \mathbf{0}$ ，则得到线性规划问题的基解 $\mathbf{x} = (\mathbf{B}^{-1}\mathbf{b}, \mathbf{0})^T$ 。一个线性规划问题最多有 C_n^m 个基解；



如何快速判断一个给定解 \mathbf{x} 是不是线性规划问题的基解

首先判断给定向量 \mathbf{x} 中有没有特定数量的 0 元素 ($\geq n - m$)，并判断其中基变量对应的基向量是否线性无关。

基可行解 注意到基解已经满足所有约束条件中的等值约束，若再满足决策变量非负的约束条件（即 $\mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$ ），则称其为基可行解 (basic feasible solution)；

退化基解 当基解中的非零分量大于 $n - m$ 时，则称该基解为退化基解。如果一个线性规划问题中至少有一个基可行解为退化解，则称该线性规划问题为退化问题 (degeneracy)，只有所有基可行解均为非退化解时才是非退化问题。退化是因为约束条件中存在多余约束导致的。采用单纯形法求解退化问题时可能会出现以下两类不利情况：

- 多次迭代后仍未跳出某一目标值。因为问题出现多余约束，因此会出现三个或以上超平面交于同一条线的情况，此时有多个基可行解对应不同的基向量，但解的形式数值上完全相同，因此也就对应相同的目标值。对于这类情况，单纯形法需要有限次迭代后才能跳出这一目标值，造成计算资源的浪费，但不会影响最终解的质量；
- 多次迭代后又回到某一基可行解。这类情况非常少见，此时程序将陷入死循环，无法得到最优解。遵循 Bland 规则可以避免这一情况。

最优基可行解 使得目标函数达到最小的基可行解称为最优基可行解 (optimal basic feasible solution)，对应的基称为最优基 (optimal basis)。

3. 对于任意一个线性规划问题，其解可能出现以下四种情况：

- 有一个最优解；
- 有无穷多个最优解；
- 无界解；
- 无可行解。

17.1.1 线性规划的基本定理

1. (定理) 若线性规划问题存在可行解，则可行域为凸集¹。

证：记 $\mathbf{x}_1, \mathbf{x}_2 \in S$ 为线性规划问题的两个可行解，只需证明对于任意 $\alpha \in [0, 1]$ ，有 $\bar{\mathbf{x}} = \alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2 \in S$ ，即证 $\bar{\mathbf{x}}$ 满足两类约束条件。首先证 $\bar{\mathbf{x}}$ 满足等值约束：

$$\begin{aligned} & \because \mathbf{A}\mathbf{x}_1 = \mathbf{A}\mathbf{x}_2 = \mathbf{b} \\ & \therefore \mathbf{A}\bar{\mathbf{x}} = \alpha\mathbf{A}\mathbf{x}_1 + (1 - \alpha)\mathbf{A}\mathbf{x}_2 = \alpha\mathbf{b} + (1 - \alpha)\mathbf{b} = \mathbf{b} \end{aligned}$$

进一步地，易证在 $\mathbf{x}_1, \mathbf{x}_2 \geq 0$ 且 $\alpha \in [0, 1]$ 的条件下有 $\bar{\mathbf{x}} \geq 0$ 满足决策变量非负约束条件。综上命题得证。

2. (引理) 线性规划问题的可行解 \mathbf{x} 为基可行解的充要条件是 \mathbf{x} 的正分量所对应的系数列向量线性独立。

证：首先证明必要性，即证“如果线性规划问题的可行解 \mathbf{x} 为基可行解，则 \mathbf{x} 的正分量所对应的系数列向量线性独立”，显然，根据定义该命题成立。

进一步地证明充分性，即证“如果线性规划问题的可行解 \mathbf{x} 的正分量所对应的系数列向量线性独立，则 \mathbf{x} 为基可行解”。假设 \mathbf{x} 共有 k 个正分量，因为所对应的系数列向量线性独立，必有 $k \leq m$ 。

- 当 $k = m$ 时，则 k 个正分量所对应的系数列向量正好对应一组基，则 \mathbf{x} 为基可行解；
- 当 $k < m$ 时，因为矩阵 A 为行满秩矩阵（此时行秩列秩均为 m ），必能从剩下 $n - k$ 项列向量中取出 $m - k$ 项凑成一组基，对应的基可行解正好也是 \mathbf{x} 。

¹ 凸集 (convex set): 对 $\forall \mathbf{x}_1, \mathbf{x}_2 \in S$ ，若对 $\forall \mathbf{x} = \alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2 \in S, 0 \leq \alpha \leq 1$ ，则集合 S 为凸集。

综上命题得证。

3. (定理) 线性规划问题的基可行解与可行域的顶点²一一对应。

证: 首先证明必要性, 即证“线性规划问题的基可行解是可行域的顶点”。采用反证法, 即证假设“存在一个线性规划问题的基可行解 \mathbf{x} 不处于可行域的顶点”不成立。假设基可行解 \mathbf{x} 不是可行域的顶点, 则必然存在另外两个不同的可行解 $\mathbf{x}_1, \mathbf{x}_2$ ($A\mathbf{x}_1 = A\mathbf{x}_2 = \mathbf{b}, \mathbf{x}_1 \geq \mathbf{0}, \mathbf{x}_2 \geq \mathbf{0}, \mathbf{x}_1 \neq \mathbf{x}_2$), 有

$$\mathbf{x} = \alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2 \quad \alpha \in (0, 1)$$

因为 \mathbf{x} 为基可行解, 则其中至少有 $n - m$ 个分量为 0, 又 $\alpha > 0, 1 - \alpha > 0, \mathbf{x}_1 \geq \mathbf{0}, \mathbf{x}_2 \geq \mathbf{0}$, 则 $\mathbf{x}_1, \mathbf{x}_2$ 中对应的至少 $n - m$ 个分量也必须同样为 0。不失一般性地, 假设 $\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2$ 的前 $k (\leq m)$ 个分量大于 0, 此时有

$$\sum_{j=1}^k \mathbf{p}_j x_{1j} = \sum_{j=1}^k \mathbf{p}_j x_{2j} = \mathbf{b} \implies \sum_{j=1}^k \mathbf{p}_j (x_{1j} - x_{2j}) = \mathbf{0}$$

因为 $\mathbf{x}_1 \neq \mathbf{x}_2$, 因此系数 $x_{1j} - x_{2j}$ 不全为 0, 因此 $(\mathbf{p}_1, \dots, \mathbf{p}_k)$ 线性相关, 故 \mathbf{x} 不是基可行解, 显然与假设相矛盾, 即不存在一个线性规划问题的基可行解位于可行域中其它两不相同可行解的连线上, 故线性规划问题的基可行解是可行域的顶点。

进一步地证明充分性, 即证“线性规划问题可行域的顶点是基可行解”。采用反证法, 即证假设“线性规划问题可行域中存在一个顶点 \mathbf{x} 不是基可行解”不成立。假设 \mathbf{x} 具有 k 个正分量, 并不失一般性地令 $\mathbf{x} = (x_1, \dots, x_k, 0 \dots, 0)^T$ 。记 $x_{\min}^+ = \min\{x_1, \dots, x_k\}$ 。因为 \mathbf{x} 不是基可行解, 则对应的 k 个列向量线性相关, 必然存在不全为零的 λ_j , 使得

$$\sum_{j=1}^k x_{\min}^+ \frac{\lambda_j}{|\lambda|_{\max}} \mathbf{p}_j = \sum_{j=1}^k \delta_j \mathbf{p}_j = \mathbf{0} \quad \delta_j = x_{\min}^+ \frac{\lambda_j}{|\lambda|_{\max}} \quad |\lambda|_{\max} = \max\{|\lambda_1|, \dots, |\lambda_k|\}$$

记向量 $\boldsymbol{\delta} = (\delta_1, \dots, \delta_k, 0 \dots, 0)^T$, 显然有

$$A\mathbf{x} = A\mathbf{x}_1 = A\mathbf{x}_2 = \mathbf{b} \quad \mathbf{x}_1 = \mathbf{x} + \boldsymbol{\delta} \geq \mathbf{0} \quad \mathbf{x}_2 = \mathbf{x} - \boldsymbol{\delta} \geq \mathbf{0} \quad \mathbf{x} = 0.5\mathbf{x}_1 + 0.5\mathbf{x}_2$$

此时 $\mathbf{x}_1, \mathbf{x}_2$ 均为问题的可行解, 且 \mathbf{x} 在 $\mathbf{x}_1, \mathbf{x}_2$ 连线的中点上, 因此 \mathbf{x} 不是可行域的顶点, 故假设不成立, 可行域的顶点一定是基可行解。

综上命题得证。

4. (定理) 若线性规划问题有最优解, 一定存在一个基可行解是最优解。³

证: 记 \mathbf{x} 为线性规划的一个最优解, 假设其正分量个数为 k , 并记 $\mathbf{x} = (x_1, \dots, x_k, 0, \dots, 0)^T$ 。

- 若 \mathbf{x} 对应的 k 个列向量线性无关, 则为最优基可行解, 命题得证;
- 若 \mathbf{x} 对应的 k 个列向量线性相关, 则非最优基可行解, 则不位于可行域的顶点处。此时必然有

$$\mathbf{c}^T \mathbf{x} = \alpha \mathbf{c}^T \mathbf{x}_1 + (1 - \alpha) \mathbf{c}^T \mathbf{x}_2 = f_{\min} \quad \mathbf{x}_1 \neq \mathbf{x}_2 \in S \quad \alpha \in (0, 1)$$

为使上式成立, 只有 $\mathbf{c}^T \mathbf{x}_1 = \mathbf{c}^T \mathbf{x}_2 = f_{\min}$, 即 $\mathbf{x}_1, \mathbf{x}_2$ 同样为问题的最优解。因此“若最优解 \mathbf{x} 不是最优基可行解, 则必然可以构造新的最优解 $\mathbf{x}_1, \mathbf{x}_2$ ”。新最优解的构造参考上一定理的充分性证明: 因为 \mathbf{x} 不是基可行解, 则对应的 k 个列向量线性相关, 必然存在不全为零的 δ_j , 使得

$$\sum_{j=1}^k \delta_j \mathbf{p}_j = \mathbf{0}$$

记向量 $\boldsymbol{\delta} = (\delta_1, \dots, \delta_k, 0 \dots, 0)^T$, 显然可以构造新的最优解 $\mathbf{x}_1 = \mathbf{x} + \boldsymbol{\delta} \geq \mathbf{0}$ 或 $\mathbf{x}_2 = \mathbf{x} - \boldsymbol{\delta} \geq \mathbf{0}$ 。注意到系数向量 $\boldsymbol{\delta}$ 可以等比缩放, 因此总能通过构造 \mathbf{x}_1 或 \mathbf{x}_2 使得新的最优解的正分量数量至少减一。

- 若新的最优解正分量对应的列向量线性无关, 则得到最优基可行解, 命题得证;

²顶点 (vertex): 设 x 为凸集 S 中的一个点, 若凸集中不存在两个不同于 x 的点 x_1, x_2 , 使得 $x = \alpha x_1 + (1 - \alpha)x_2, \alpha \in (0, 1)$, 则称 x 为凸集的一个顶点。

³命题“线性规划的最优解一定是最优基可行解”是错误的, 因为当问题有无穷多个最优解时, 并非每个解都是最优基可行解。

- 若新的最优解只有一个正分量，单一的列向量自然线性无关，同样得到最优基可行解，命题得证；
- 若新的最优解正分量对应的列向量线性相关，则再构造新的最优解，直至归于以上两种可能。综上命题得证。

17.1.2 单纯形法 (simplex method)

1. 由线性规划的基本定理可以看出，只需搜寻基可行解即可得到最优解，而基可行解的个数是有限的，因此大大缩小了最优解的搜索范围；
2. 单纯形法就是一种对基可行解的搜索算法。其基本思想是从线性规划问题的某一个基可行解出发，经过基变换另一个使得目标函数改善的基可行解，循环执行至得到最优基可行解（或判定无可行解、无界解）。单纯形法包括初始基可行解的确定；解的最优性判定；基的转换共三个关键问题；
3. 初始基可行解的确定

注意到基可行解 \mathbf{x} 中基变量分量 \mathbf{x}_B 与非基变量分量 \mathbf{x}_N 满足以下关系

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N$$

显然为方便计算，可选择对应于 $\mathbf{B} = \mathbf{I}$ 为单位阵的变量作为基变量，此时 $\mathbf{B}^{-1} = \mathbf{B} = \mathbf{I}$ 。对于约束条件均为“≤”的线性规划问题，在将其转化为标准形式时，引入的松弛变量的系数矩阵即为单位阵；当存在“≤”或“=”的约束条件时，则可通过引入人工变量 (artificial variables) 构造出单位阵（详见第 17.1.4 节）；

4. 解的最优性判定

将 $\mathbf{x} = (\mathbf{x}_B^T \quad \mathbf{x}_N^T)^T$ 及 $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N$ 代入线性规划问题的标准式

$$\begin{aligned} \min f &= \mathbf{c}^T \mathbf{x} = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N = \mathbf{c}_B^T (\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N) + \mathbf{c}_N^T \mathbf{x}_N \\ &= \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b} - (\mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N} - \mathbf{c}_N^T) \mathbf{x}_N \end{aligned} \quad (17.4)$$

$$\text{s.t. } \mathbf{x} \geq \mathbf{0}$$

因为 $\mathbf{x}_B, \mathbf{x}_N$ 的关系式是基于线性规划问题的等值约束推导而成，因此代入标准式后约束仅剩决策变量非负约束，而目标函数可以仅由非基变量 \mathbf{x}_N 表示。记 \mathbf{x}_B 为 \mathbf{x} 的前 m 个元素，并令 $\mathbf{B}^{-1} = \mathbf{B} = \mathbf{I}$ ，则上式改写为

$$\begin{aligned} \min f &= f_0 - \sum_{j=m+1}^n \sigma_j x_j \quad f_0 = \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b} = \mathbf{c}_B^T \mathbf{b} \quad \sigma_j = (\mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N} - \mathbf{c}_N^T)_j = (\mathbf{c}_B^T \mathbf{N} - \mathbf{c}_N^T)_j \quad (17.5) \\ \text{s.t. } x_i &= b_i - \sum_{j=m+1}^n a_{ij} x_j \geq 0 \quad (i = 1, \dots, m) \\ x_j &\geq 0 \quad (j = m+1, \dots, n) \end{aligned}$$

一般地，称 σ_j 为检验数或判别数；

●
●
●
计算 检验数 σ_j

计算检验数分为两步：1) 首先根据等值约束求解基变量（将基变量用非基变量表示）；2) 而后代入目标函数，并将目标函数化简为常数减非基变量的形式。

例：考虑以下线性规划问题，若以 x_1, x_2 为基变量，计算检验数 σ_j

$$\min f = 4x_1 + x_2 + x_3$$

$$\text{s.t. } 2x_1 + x_2 + 2x_3 = 4$$

$$3x_1 + 3x_2 + x_3 = 3$$

$$x_j \geq 0 \quad (j = 1, 2, 3)$$

解：求解以上线性规划问题等值约束

$$\begin{cases} x_1 = 3 - \frac{5}{3}x_3 \\ x_2 = -2 + \frac{4}{3}x_3 \end{cases}$$

并将上式代入目标函数后化简

$$\min f = 4 \times \left(3 - \frac{5}{3}x_3\right) + \left(-2 + \frac{4}{3}x_3\right) + x_3 = 10 - \frac{13}{3}x_3$$

对比系数得出检验数， $\sigma_1 = \sigma_2 = 0, \sigma_3 = \frac{13}{3}$ 。观察以上求解过程发现即使 $B \neq I$ 也能求解出检验数，建议 $B = I$ 仅是出于方便计算考虑。另外，若以 x_1, x_2 为基变量，对应的基解 $(3, -2, 0)^T$ 不满足决策变量非负约束条件，不是基可行解。

- (最优解判别定理) 对于基可行解 $\mathbf{x} = (x_1, \dots, x_m, 0, \dots, 0)^T$, 若 $\forall \sigma_j \leq 0 (j = m+1, \dots, n)$, 则 \mathbf{x} 为最优解。

证：观察目标函数 $f = f_0 - \sum_{j=m+1}^n \sigma_j x_j$, 若存在 $\sigma_k > 0$, 则只需构造一个新的可行解 \mathbf{x}' , 使得 $x'_k = \theta > 0$, 且 $x'_j = 0, (j = m+1, \dots, k-1, k+1, \dots, n)$, 即可使得目标函数减小 ($f = f_0 - \sigma_k \theta < f_0$), 即原先的基可行解 \mathbf{x} 不是最优解。因此只需证明 $\exists \sigma_k > 0$ 时新的可行解 \mathbf{x}' 存在即可证明本定理，在 $B = I$ 的前提条件下，即只需证明下式成立

$$x'_i = b_i - \sum_{j=m+1}^n a_{ij} x'_j = b_i - a_{ik} \theta \geq 0 \quad (i = 1, \dots, m)$$

分类讨论如下

- 当 $\forall b_i > 0$ 时, 只需令 θ 满足下式即可保证 $\forall x_i \geq 0$ 。

$$0 < \theta \leq \min \left\{ \frac{b_i}{a_{ik}} \mid a_{ik} > 0 \right\}$$

- 当 $\exists b_i = 0, a_{ik} > 0$ 时, 显然不可能有 $\theta > 0$ 使得 $x_i > 0$ 。然而当存在 $b_i = 0$ 时, 将原基可行解 $\mathbf{x} = (x_1, \dots, x_m, 0, \dots, 0)^T$ 代入 $x_i = b_i - \sum_{j=m+1}^n a_{ij} x_j$, 有 $x_i = 0$, 说明 $\exists b_i = 0$ 时对应退化情况。退化现象可以通过消去多余约束避免。如果不消去多余约束, 则可通过遵守 Bland 法则避免死循环情况, 因此令 $\theta = 0$ 并进入下一次迭代即可, 此时至少可以保证不会使得目标函数增大。

综上所述, 在不考虑退化情况的前提下, 在 $\exists \sigma_k > 0$ 时总存在新的可行解 \mathbf{x}' 使得目标函数继续减小, 因此只有 $\forall \sigma_j \leq 0$ 时对应的基可行解才是最优解。

- (无穷多最优解判别定理) 对于基可行解 $\mathbf{x} = (x_1, \dots, x_m, 0, \dots, 0)^T$, 若 $\forall \sigma_j \leq 0 (j = m+1, \dots, n)$, 且 $\exists \sigma_k = 0 (m+1 \leq k \leq n)$, 则 \mathbf{x} 为最优解且问题有无穷多个最优解。

证：由上一定理已知, 在 $\forall \sigma_j \leq 0 (j = m+1, \dots, n)$ 时对应的基可行解 $\mathbf{x} = (x_1, \dots, x_m, 0, \dots, 0)^T$ 为最优解, 因此若能证明 “ $\exists \sigma_k = 0$ 时存在另一可行解 \mathbf{x}' 使得 $f_{\min} = \mathbf{c}^T \mathbf{x} = \mathbf{c}^T \mathbf{x}'$ ”, 则在 \mathbf{x} 与 \mathbf{x}' 连线上的任意一点均为最优解。采用同样的方法, 构造满足约束条件的新可行解 $\mathbf{x}' = (x'_1, \dots, x'_m, 0, \dots, 0, \theta, 0, \dots, 0)^T$, 其中 $x'_k = \delta > 0$, 在上一定理的证明过程中已证明这一构造方法总是可行。将新可行解代入目标函数

$$f = \mathbf{c}^T \mathbf{x}' = f_0 - \sum_{j=m+1}^n \sigma_j x'_j = f_0 - \sigma_k \theta = f_0 = f_{\min} (= \mathbf{c}^T \mathbf{x})$$

此时新的可行解也是最优解。综上命题得证。

- (无界解判别定理) 对于基可行解 $\mathbf{x} = (x_1, \dots, x_m, 0, \dots, 0)^T$, 若 $\exists \sigma_k > 0 (m+1 \leq k \leq n)$, 且 $\forall a_{ik} \leq 0 (i = 1, \dots, m)$, 则问题有无界解。

证：同样地采用类似于最优解判别定理的证明方法。已知对于基可行解 $\mathbf{x} = (x_1, \dots, x_m, 0, \dots, 0)^T$, 若 $\exists \sigma_k > 0 (m+1 \leq k \leq n)$, 则总能构造新的可行解 $\mathbf{x}' = (x_1, \dots, x_m, 0, \dots, 0)^T$ (其中 $\mathbf{x}' = \theta > 0$) 使得目标函数继续下降, 且 θ 取值越大, 目标函数越小。 θ 的取值范围为

$$0 < \theta \leq \min \left\{ \frac{b_i}{a_{ik}} \mid a_{ik} > 0 \right\}$$

若 $\forall a_{ik} \leq 0$, 意味着 θ 没有上界。因此可以令 $\mathbf{x}' = \theta \rightarrow \infty$, 此时有 $f = \mathbf{c}^T \mathbf{x}' \rightarrow -\infty$, 即问题有无界解。

5. 相邻基的转换

在单纯形法中, 基的转换是在相邻基 (adjacent basis) 之间发生的, 所谓“相邻”是指从一个基可行解 $\mathbf{x}^{(u)}$ 到新的基可行解 $\mathbf{x}^{(u+1)}$ 的转换过程中有且仅有一个基变量发生改变。因此需要在原本的非基变量中确定一个进基变量 (entering variable) 使其变为正数, 再从原本的基变量中确定一个出基变量 (leaving variable) 使其变为 0 保证基变量个数为 m 。

进基变量的确定 当某个非基变量的检验数 $\sigma_j^{(u)} > 0$ 时, 如前所述, 只要适当增加相应变量 $x_j^{(u)}$ 的值, 即可使得目标函数下降。当只有一个检验数 $\sigma_k^{(u)} > 0$ 时, 则相应的 $x_k^{(u)}$ 则为进基变量; 当存在两个或以上的检验数 $\sigma_j^{(u)} > 0$ 时, 一种选择准则是选择其中最大的 $\sigma_k^{(u)}$ 所对应的非基变量 $x_k^{(u)}$ 作为进基变量 ($\sigma_j^{(u)}$ 越大意味着相应 $x_j^{(u)}$ 增大一个单位导致的目标函数下降程度越大)。令 $x_k^{(u+1)} = \theta > 0$ 。

$$\sigma_k^{(u)} = \max\{\sigma_j^{(u)} \mid \sigma_j^{(u)} > 0, j = m+1, \dots, n\} \quad (17.6)$$

出基变量的确定 在原基变量中选择一个变量 $x_l^{(u)}$ 另其为 0, 变为非基变量。出基变量的确定原则是保持解的可行性——当令进基变量 $x_k^{(u+1)} = \theta > 0$ 时, 代入约束条件可以得到原基变量 $x_l^{(u+1)} = b_l^{(u)} - a_{lk}^{(u)} \theta$, 为保持解的可行性, 同时使得原基变量中的一个分量变为 0 称为非基变量, 应有

$$\min_i \{x_i^{(u+1)} \mid i = 1, \dots, m\} = 0 \implies \theta = \min_i \left\{ \frac{b_i^{(u)}}{a_{ik}^{(u)}} \mid a_{ik}^{(u)} > 0 \right\}, x_l^{(u)} = \arg \min_i \left\{ \frac{b_i^{(u)}}{a_{ik}^{(u)}} \mid a_{ik}^{(u)} > 0 \right\} \quad (17.7)$$

当为退化问题时, 则可能出现 $\theta = \min_i \{b_i/a_{ik} \mid a_{ik} > 0\} = 0$ 的情况, 此时新的基可行解在形式上完全不变, 但实际上基变量、基向量都已经发生改变, 可以进入下一次迭代;

Bland 法则 Bland 法则是另一种确认进基变量与出基变量的法则, 只需遵守非常简洁的规则即可避免死循环现象的出现。法则包括以下两部分:

进基变量确定 选择所有满足 $\sigma_j > 0$ 的非基变量 x_j 中下标最小的作为进基变量 x_k ;

出基变量确定 当多个基变量 x_i 均使得 $\{b_i/a_{ik} \mid a_{ik} > 0\}$ 取最小时, 选择下标最小的作为出基变量 x_l 。

旋转运算 在确定出基变量与进基变量后矩阵分块 $\mathbf{A} = [\mathbf{B} \quad \mathbf{N}]$ 也会随之调整。因为新的基向量阵 $\mathbf{B}^{(u+1)}$ 很可能不是单位阵, 需要通过初等行变换将其变为单位阵, 此时非基向量 $\mathbf{N}^{(u+1)}$ 和右端项常数 $\mathbf{b}^{(u+1)}$ 也会随之改变。这一过程被称为旋转运算, 运算公式如下

$$\begin{cases} a_{lj}^{(u+1)} = \frac{a_{lj}^{(u)}}{a_{lk}^{(u)}} & a_{ij}^{(u+1)} = a_{ij}^{(u)} - a_{ik}^{(u)} \frac{a_{lj}^{(u)}}{a_{lk}^{(u)}} \quad (i \neq l) \\ b_l^{(u+1)} = \frac{b_l^{(u)}}{a_{lk}^{(u)}} & b_l^{(u+1)} = b_l^{(u)} - a_{ik}^{(u)} \frac{b_l^{(u)}}{a_{lk}^{(u)}} \quad (i \neq l) \end{cases} \quad (17.8)$$

6. 不妨进一步推导基变换后目标函数的增益和约束条件的改变。对于初始基可行解, 已知目标函数和约束如下

$$\begin{aligned} f &= f_0 - \sum_{j=m+1}^n \sigma_j x_j \\ \text{s.t. } x_i &= b_i - \sum_{j=m+1}^n a_{ij} x_j \geq 0 \quad (i = 1, \dots, m) \\ x_j &\geq 0 \quad (j = m+1, \dots, n) \end{aligned}$$

从原非基变量中选择进基变量 x_k 再从原基变量中选择出基变量 x_l 后，首先需要修改约束条件。观察 x_k, x_l 关系，有

$$x_l = b_l - a_{lk}x_k - \sum_{\substack{j=m+1 \\ j \neq k}}^n a_{lj}x_j$$

因为 x_k 成了新的基变量，应将其移到等式右边，有

$$x_k = \frac{b_l}{a_{lk}} - \frac{x_l}{a_{lk}} - \sum_{\substack{j=m+1 \\ j \neq k}}^n \frac{a_{lj}}{a_{lk}}x_j$$

对于原先的其它基变量 $x_i (i \neq l)$ ，则应修改其中与 x_k 有关的分量

$$\begin{aligned} x_i &= b_i - a_{ik}x_k - \sum_{\substack{j=m+1 \\ j \neq k}}^n a_{ij}x_j = b_i - a_{ik} \left(\frac{b_l}{a_{lk}} - \frac{x_l}{a_{lk}} - \sum_{\substack{j=m+1 \\ j \neq k}}^n \frac{a_{lj}}{a_{lk}}x_j \right) - \sum_{\substack{j=m+1 \\ j \neq k}}^n a_{ij}x_j \\ &= \left(b_i - \frac{a_{ik}}{a_{lk}}b_l \right) - \left(-\frac{a_{ik}}{a_{lk}} \right)x_l - \sum_{\substack{j=m+1 \\ j \neq k}}^n \left(a_{ij} - \frac{a_{ik}}{a_{lk}}a_{lj} \right)x_j \quad (i \neq l) \end{aligned}$$

最后修改目标函数中与 x_k 有关的分量

$$\begin{aligned} f &= f_0 - \sigma_k x_k - \sum_{\substack{j=m+1 \\ j \neq k}}^n \sigma_j x_j = f_0 - \sigma_k \left(\frac{b_l}{a_{lk}} - \frac{x_l}{a_{lk}} - \sum_{\substack{j=m+1 \\ j \neq k}}^n \frac{a_{lj}}{a_{lk}}x_j \right) - \sum_{\substack{j=m+1 \\ j \neq k}}^n \sigma_j x_j \\ &= \left(f_0 - \frac{\sigma_k}{a_{lk}}b_l \right) - \left(-\frac{\sigma_k}{a_{lk}} \right)x_l - \sum_{\substack{j=m+1 \\ j \neq k}}^n \left(\sigma_j - \frac{a_{lj}}{a_{lk}}\sigma_k \right)x_j \end{aligned}$$

综上，迭代一次之后新的目标函数和约束条件变为

$$\begin{aligned} f &= \left(f_0 - \frac{\sigma_k}{a_{lk}}b_l \right) - \left(-\frac{\sigma_k}{a_{lk}} \right)x_l - \sum_{\substack{j=m+1 \\ j \neq k}}^n \left(\sigma_j - \frac{a_{lj}}{a_{lk}}\sigma_k \right)x_j \tag{17.9} \\ \text{s.t. } x_k &= \frac{b_l}{a_{lk}} - \frac{x_l}{a_{lk}} - \sum_{\substack{j=m+1 \\ j \neq k}}^n \frac{a_{lj}}{a_{lk}}x_j \geq 0 \\ x_i &= \left(b_i - \frac{a_{ik}}{a_{lk}}b_l \right) - \left(-\frac{a_{ik}}{a_{lk}} \right)x_l - \sum_{\substack{j=m+1 \\ j \neq k}}^n \left(a_{ij} - \frac{a_{ik}}{a_{lk}}a_{lj} \right)x_j \geq 0 \quad (i = 1, \dots, m, i \neq l) \\ x_j &\geq 0 \quad (j = l, m+1, \dots, n, j \neq k) \end{aligned}$$

可以看到，新的目标函数值 $f_0 - \frac{\sigma_k}{a_{lk}}b_l$ 确实小于原目标函数值 f_0 ，且对比迭代前后的目标函数和约束条件表达式，同样能推导得旋转公式：

7. 以上即为单纯形法的基本原理。单纯形法的计算步骤大体如下：

- (a) 首先选取一组基变量，并计算对应的基可行解、目标函数值、等值约束系数矩阵及检验数；
- (b) 再基于检验数和等值约束系数进行解的最优化判定，若当前即为最优解或问题有无界解则结束运算，若当前可进一步优化则进入下一步；
- (c) 选取合适的进基变量和出基变量，得到一组新的基变量，并返回第一步。

17.1.3 单纯形表

单纯形表时单纯形法运算时的辅助工具。在基于单纯形法求解线性规划问题时，将相关参数 A, b, c, x, σ 以及目标函数值 f 以表格的形式表示，即为单纯形表。单纯形表的基本格式及计算流程如下：

		c_1	c_2	\cdots	c_m	c_{m+1}	\cdots	c_n			
c_B	x_B	x_1	x_2	\cdots	x_m	x_{m+1}	\cdots	x_n		b	
c_1	x_1	1	0	\cdots	0	$a_{1,m+1}$	\cdots	$a_{1,n}$		b_1	
c_2	x_2	0	1	\cdots	0	$a_{2,m+1}$	\cdots	$a_{2,n}$		b_2	
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots		\vdots	
c_m	x_m	0	0	\cdots	1	$a_{m,m+1}$	\cdots	$a_{m,n}$		b_m	
σ		0	0	\cdots	0	σ_{m+1}	\cdots	σ_n		f	

1. 首先确定基变量 x_B 与非基变量 x_N , 填入相应系数 c, B, N, b ;

2. 再对矩阵 $[B \ N \ b]$ 进行初等行变换使得 $B = I$, 此时右端项常数 b 的取值即为相应基变量 x_B 的值, 得到基可行解 $(b_1, \dots, b_m, 0, \dots, 0)^T$, 并可计算目标函数值 f

$$f = c_B^T B^{-1} x_B = c_B^T b = \sum_{i=1}^m c_i b_i$$

3. 最后计算判别数 σ

$$\sigma_j = (c_B^T B^{-1} N - c_B^T)_j = (c_B^T N - c_B^T)_j = -c_j + \sum_{i=1}^m c_i a_{ij}$$



单纯形法算例

例: 求解以下线性规划问题

$$\begin{aligned} \min f &= -3x_1 - 4x_2 \\ \text{s.t. } 2x_1 + x_2 + x_3 &= 6 \\ x_1 + 3x_2 + x_4 &= 6 \\ -x_1 + 2x_2 + x_5 &= 2 \\ x_j &\geq 0 \quad (j = 1, \dots, 5) \end{aligned}$$

解: 注意到以上问题本可以化为二维线性规划问题由图解法求解, 为熟悉单纯形法计算本例采用单纯形法。观察到等式约束中存在单位阵, 因此令初始基向量 $x_B = (x_3, x_4, x_5)^T$, 绘制单纯形表如下

		-3	-4	0	0	0	
c_B	x_B	x_1	x_2	x_3	x_4	x_5	b
0	x_3	2	1	1	0	0	6
0	x_4	1	3	0	1	0	6
0	x_5	-1	2	0	0	1	2
σ		3	4	0	0	0	0

此时基可行解为 $(0, 0, 6, 6, 2)^T$, 目标函数 $f = 0$ 。观察检验数, 显然不全 ≤ 0 , 且不存在某一列 a_{ik} 全部 ≤ 0 , 因此可继续优化。因为 $\sigma_2 > \sigma_1 > 0$, 故选择 x_2 为进基变量; 又因为 $\min\{b_i/a_{ik} | a_{ik} > 0\} = \min\{6, 2, 1\} = 1$, 故选取 x_5 为出基变量。新的单纯形表如下

		-3	-4	0	0	0	
c_B	x_B	x_1	x_2	x_3	x_4	x_5	b
0	x_3	$5/2$	0	1	0	$-1/2$	5
0	x_4	$5/2$	0	0	1	$-3/2$	3
-4	x_2	$-1/2$	1	0	0	$1/2$	1
σ		5	0	0	0	-2	-4

此时基可行解为 $(0, 1, 5, 3, 0)^T$, 目标函数 $f = -4$ 。观察检验数, 显然不全 ≤ 0 , 且不存在某一列 a_{ik} 全部 ≤ 0 , 因此可继续优化。因为 $\sigma_1 = 5 > 0$, 故选择 x_1 为进基变量; 又因为 $\min\{b_i/a_{ik} | a_{ik} > 0\} = \min\{2, 6/5\} = 6/5$, 故选取 x_4 为出基变量。新的单纯形表如下

		-3	-4	0	0	0	
c_B	x_B	x_1	x_2	x_3	x_4	x_5	b
0	x_3	0	0	1	-1	1	2
-3	x_1	1	0	0	$2/5$	$-3/5$	$6/5$
-4	x_2	0	1	0	$1/5$	$1/5$	$8/5$
σ		0	0	0	-2	1	-10

此时基可行解为 $(6/5, 8/5, 2, 0, 0)^T$, 目标函数 $f = -10$ 。观察检验数, 显然不全 ≤ 0 , 且不存在某一列 a_{ik} 全部 ≤ 0 , 因此可继续优化。因为 $\sigma_5 = 1 > 0$, 故选择 x_5 为进基变量; 又因为 $\min\{b_i/a_{ik} | a_{ik} > 0\} = \min\{2, 8\} = 2$, 故选取 x_3 为出基变量。新的单纯形表如下

\mathbf{c}_B	\mathbf{x}_B	-3	-4	0	0	0	\mathbf{b}
0	x_5	0	0	1	-1	1	2
-3	x_1	1	0	3/5	-1/5	0	12/5
-4	x_2	0	1	-1/5	2/5	0	6/5
σ		0	0	-1	-1	0	-12

此时基可行解为 $(12/5, 6/5, 0, 0, 2)^T$, 目标函数 $f = -12$ 。观察检验数, 全部 ≤ 0 且不存在 $\sigma_j = 0$, 因此 $\mathbf{x} = (12/5, 6/5, 0, 0, 2)^T$ 即为问题的唯一最优解。

17.1.4 大 M 法与两阶段法 (big M method & two-phase method)

在用单纯形法求解线性规划问题时首先需要确定基变量, 为了方便矩阵求逆常常选择使得基向量组合为单位阵的变量作为基向量。然而当等值约束方程组系数矩阵 \mathbf{A} 中不存在单位阵时, 就需要通过引入人工变量 (artificial variable) 构造单位阵。具体地包括大 M 法 (big M method) 与两阶段法 (two-phase method) 两类常用方法。

首先介绍大 M 法。观察如下线性规划问题

$$\begin{aligned} \min f &= \mathbf{c}^T \mathbf{x} \\ \text{s.t. } \mathbf{A}\mathbf{x} &= \mathbf{b} \quad (i = 1, \dots, m, b_i \geq 0) \\ \mathbf{x} &\geq \mathbf{0} \quad (j = 1, \dots, n) \end{aligned}$$

若矩阵 \mathbf{A} 中不包含单位阵, 则可引入人工变量 $\mathbf{x}_A = (x_{n+1}, \dots, x_{n+m})^T$, 将约束条件和目标函数改写为

$$\begin{aligned} \min f &= \mathbf{c}^T \mathbf{x} + \mathbf{c}_M^T \mathbf{x}_A \tag{17.10} \\ \text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{I}\mathbf{x}_A &= \mathbf{b} \quad (i = 1, \dots, m, b_i \geq 0) \\ \mathbf{x}, \mathbf{x}_A &\geq \mathbf{0} \quad (j = 1, \dots, n+m) \end{aligned}$$

上式中 \mathbf{c}_M 为 $m \times 1$ 的常向量, 其元素为一充分大的正数, 这也是“大 M 法”的由来。观察新的线性规划问题, 人工变量引入后系数矩阵即存在单位阵, 若新的线性规划问题的目标函数具有最优值 f^* , 且对应的最优基可行解中人工变量全部出基, 则原线性规划问题有最优解, 目标函数最优值同为 f^* ; 反之则原线性规划问题无解。

●
●
●
大 M 法 算例

例: 用大 M 法求解以下线性规划问题

$$\begin{aligned} \min f &= 4x_1 + x_2 + x_3 \\ \text{s.t. } 2x_1 + x_2 + 2x_3 &= 4 \\ 3x_1 + 3x_2 + x_3 &= 3 \\ x_j &\geq 0 \quad (j = 1, \dots, 3) \end{aligned}$$

解: 观察到等式约束中不存在单位阵, 采用大 M 法构造新的线性规划问题如下

$$\begin{aligned} \min f &= 4x_1 + x_2 + x_3 + Mx_4 + Mx_5 \\ \text{s.t. } 2x_1 + x_2 + 2x_3 + x_4 &= 4 \\ 3x_1 + 3x_2 + x_3 + x_5 &= 3 \\ x_j &\geq 0 \quad (j = 1, \dots, 5) \end{aligned}$$

因此令初始基向量 $\mathbf{x}_B = (x_4, x_5)^T$, 单纯形表迭代过程如下

c_B	x_B	4	1	1	M	M	b
	x_1		x_2	x_3	x_4	x_5	
M	x_4	2	1	2	1	0	4
M	x_5	3	3	1	0	1	3
σ		5M-4	4M-1	3M-1	0	0	7M

$\mathbf{x} = (0, 0, 0, 4, 3)^T \quad f = 7M$
 σ_j 不全 ≤ 0 , 且不存在某一列 a_{ik} 全部 ≤ 0 , 可继续优化。
 $\sigma_1 > \sigma_2 > \sigma_3 > 0 \Rightarrow x_k = x_1$
 $\min\{b_i/a_{ik}|a_{ik} > 0\} = \min\{2, 1\} = 1 \Rightarrow x_l = x_5$

c_B	x_B	4	1	1	M	M	b
	x_1		x_2	x_3	x_4	x_5	
M	x_4	0	-1	4/3	1	-2/3	2
4	x_1	1	1	1/3	0	1/3	1
σ		0	3-M	4/3M+1/3	0	4/3-5/3M	2M+4

$\mathbf{x} = (1, 0, 0, 2, 0)^T \quad f = 2M + 4$
 σ_j 不全 ≤ 0 , 且不存在某一列 a_{ik} 全部 ≤ 0 , 可继续优化。
 $\sigma_3 > 0 \Rightarrow x_k = x_3$
 $\min\{b_i/a_{ik}|a_{ik} > 0\} = \min\{3/2, 3\} = 3/2 \Rightarrow x_l = x_4$

c_B	x_B	4	1	1	M	M	b
	x_1		x_2	x_3	x_4	x_5	
1	x_3	0	-3/4	1	3/4	-1/2	3/2
4	x_1	1	5/4	0	-1/4	1/2	1/2
σ		0	13/4	0	-1/4-M	3/2-M	7/2

$\mathbf{x} = (1/2, 0, 3/2, 0, 0)^T \quad f = 7/2$
 σ_j 不全 ≤ 0 , 且不存在某一列 a_{ik} 全部 ≤ 0 , 可继续优化。
 $\sigma_2 > 0 \Rightarrow x_k = x_2$
 $\min\{b_i/a_{ik}|a_{ik} > 0\} = \min\{2/5\} = 2/5 \Rightarrow x_l = x_1$

c_B	x_B	4	1	1	M	M	b
	x_1		x_2	x_3	x_4	x_5	
1	x_3	3/5	0	1	3/5	-1/5	9/5
1	x_2	4/5	1	0	-1/5	2/5	2/5
σ		-13/5	0	0	2/5-M	1/5-M	11/5

$\mathbf{x} = (0, 2/5, 9/5, 0, 0)^T \quad f = 11/5$
 σ_j 全部 ≤ 0 且不存在 $\sigma_j = 0$, 则 \mathbf{x} 为新问题的唯一最优解。
 又因为所有人工变量全部出基, 所以原问题存在唯一最优解且最优解为 $\mathbf{x} = (0, 2/5, 9/5)^T$ 。

综上所述, 大 M 法的优点为只需一次求解即可得到原问题的解, 但方法假设一个充分大的“M”, 在进行计算机求解时需要给出“M”的具体大小, 此时如果太大会造成计算资源消耗, 太小则很可能得到错误解, 因此方法不适合计算机求解。

进一步地介绍两阶段法, 对于一个线性规划问题, 方法需要求解两次才可得到最优解, 但实际上其计算量并不比大 M 法大。同样地对于一个标准形式的线性规划问题, 若矩阵 A 中不包含单位阵, 则可引入人工变量 $\mathbf{x}_A = (x_{n+1}, \dots, x_{n+m})^T$, 构造新的线性规划问题

$$\begin{aligned} \min f &= \sum \mathbf{x}_A \\ \text{s.t. } A\mathbf{x} + I\mathbf{x}_A &= \mathbf{b} \quad (i = 1, \dots, m, b_i \geq 0) \\ \mathbf{x}, \mathbf{x}_A &\geq \mathbf{0} \quad (j = 1, \dots, n+m) \end{aligned} \tag{17.11}$$

对比大 M 法, 可以看到两类方法构造的新线性规划问题具有完全相同的约束形式(引入单位阵的方法相同), 但目标函数不同。若以上新的线性规划问题有最优解且使得目标函数值 $f^* = 0$, 即人工变量全部出基, 意味着找到了原问题的一个基可行解, 此时可代入原问题继续优化。

●
●
●
两阶段法算例

例: 用两阶段法求解以下线性规划问题

$$\begin{aligned} \min f &= 4x_1 + x_2 + x_3 \\ \text{s.t. } 2x_1 + x_2 + 2x_3 &= 4 \\ 3x_1 + 3x_2 + x_3 &= 3 \\ x_j &\geq 0 \quad (j = 1, \dots, 3) \end{aligned}$$

解：观察到等式约束中不存在单位阵，采用两阶段法构造新的线性规划问题如下

$$\begin{aligned} \min f &= x_4 + x_5 \\ \text{s.t. } &2x_1 + x_2 + 2x_3 + x_4 = 4 \\ &3x_1 + 3x_2 + x_3 + x_5 = 3 \\ &x_j \geq 0 \quad (j = 1, \dots, 5) \end{aligned}$$

因此令初始基向量 $\mathbf{x}_B = (x_4, x_5)^T$ ，单纯形表迭代过程如下

c_B	x_B	x_1	x_2	x_3	x_4	x_5	b
1	x_4	2	1	2	1	0	4
1	x_5	3	3	1	0	1	3
σ		5	4	3	0	0	7

$$\begin{aligned} \mathbf{x} &= (0, 0, 0, 4, 3)^T \quad f = 7 \\ \sigma_j &\text{ 不全} \leq 0, \text{ 且不存在某一列 } a_{ik} \text{ 全部} \leq 0, \text{ 可继续优化。} \\ \sigma_1 > \sigma_2 > \sigma_3 > 0 \implies x_k &= x_1 \\ \min\{b_i/a_{ik}|a_{ik} > 0\} = \min\{2, 1\} &= 1 \implies x_l = x_5 \end{aligned}$$

c_B	x_B	x_1	x_2	x_3	x_4	x_5	b
1	x_4	0	-1	4/3	1	-2/3	2
0	x_1	1	1	1/3	0	1/3	1
σ		0	-1	4/3	0	-2/3	2

$$\begin{aligned} \mathbf{x} &= (1, 0, 0, 2, 0)^T \quad f = 2 \\ \sigma_j &\text{ 不全} \leq 0, \text{ 且不存在某一列 } a_{ik} \text{ 全部} \leq 0, \text{ 可继续优化。} \\ \sigma_3 > 0 \implies x_k &= x_3 \\ \min\{b_i/a_{ik}|a_{ik} > 0\} = \min\{3/2, 3\} &= 3/2 \implies x_l = x_4 \end{aligned}$$

c_B	x_B	x_1	x_2	x_3	x_4	x_5	b
0	x_3	0	-3/4	1	3/4	-1/2	3/2
0	x_1	1	5/4	0	-1/4	1/2	1/2
σ		0	0	0	-1	-1	0

$$\begin{aligned} \mathbf{x} &= (1/2, 0, 3/2, 0, 0)^T \quad f = 0 \\ \sigma_j &\text{ 全部} \leq 0 \text{ 且不存在 } \sigma_j = 0, \mathbf{x} \text{ 为新问题的最优解。} \\ \text{又因为所有人工变量全部出基, 因此得到了原问题的一个基可行解,} \\ \text{代入原问题继续求解 (此时需要修改单纯形表的检验数和目标函数)} \\ \cdot \end{aligned}$$

c_B	x_B	x_1	x_2	x_3	b
1	x_3	0	-3/4	1	3/2
4	x_1	1	5/4	0	1/2

$$\begin{aligned} \mathbf{x} &= (1/2, 0, 3/2)^T \quad f = 7/2 \\ \sigma_j &\text{ 不全} \leq 0, \text{ 且不存在某一列 } a_{ik} \text{ 全部} \leq 0, \text{ 可继续优化。} \\ \sigma_2 > 0 \implies x_k &= x_2 \\ \min\{b_i/a_{ik}|a_{ik} > 0\} = \min\{2/5\} &= 2/5 \implies x_l = x_1 \end{aligned}$$

c_B	x_B	x_1	x_2	x_3	b
1	x_3	3/5	0	1	9/5
1	x_2	4/5	1	0	2/5

$$\begin{aligned} \mathbf{x} &= (0, 2/5, 9/5)^T \quad f = 11/5 \\ \sigma_j &\text{ 全部} \leq 0 \text{ 且不存在 } \sigma_j = 0, \text{ 因此 } \mathbf{x} \text{ 为原问题的唯一最优解。} \end{aligned}$$

对比以上两算例可以看到两阶段法无法直接得到原问题的最优解，但方法无需引入一个充分大的“M”，因此适用于计算机求解。比较两种方法的计算量发现采用两阶段法计算时需要比大M法多做一个单纯形表，在两阶段法的第一个阶段中目标函数系数仅为0或1，计算检验数时较简单，第二个阶段则在初等行变换时省去了无意义的人工变量，同样简化计算，因此两阶段法的计算量未必高于大M法。

17.2 线性规划的对偶理论与灵敏度分析

17.2.1 线性规划的对偶问题

- 在数学中，对偶(duality)一般是指对同一事物(问题)从不同角度(立场)提出的两种不同表述；
- 考虑以下线性规划问题：某工厂拥有A、B两种原材料，总量分别为 b_1, b_2 ，可用于生产C、D两种产品。

其中生成一单位产品 C 需要消耗 a_{11} 单位的原材料 A 和 a_{21} 单位的原材料 B, 生成一单位产品 D 需要消耗 a_{12} 单位的原材料 A 和 a_{22} 单位的原材料 B。而一单位的产品 C、D 的利润分别为 c_1, c_2 。若该厂欲生产 C、D 两种产品, 为使得利润最高, 记产品 C、D 的产量分别为 x_1, x_2 , 则即为如下线性规划问题:

$$\begin{aligned} \max f &= c_1x_1 + c_2x_2 \\ \text{s.t. } &a_{11}x_1 + a_{12}x_2 \leq b_1 \\ &a_{21}x_1 + a_{22}x_2 \leq b_2 \\ &x_1, x_2 \geq 0 \end{aligned}$$

不妨换个角度, 除了自行生产外, 该厂也可通过出售全部原材料实现盈利。假设该厂对 A、B 两种原材料的单位定价分别为 ω_1, ω_2 。注意到为了售出原材料, 单位定价不宜过高, 但工厂显然也希望出售原材料所获得的利润能不低于自行生产产品的利润, 故得到新的线性规划问题:

$$\begin{aligned} \min z &= b_1\omega_1 + b_2\omega_2 \\ \text{s.t. } &a_{11}\omega_1 + a_{21}\omega_2 \geq c_1 \\ &a_{12}\omega_1 + a_{22}\omega_2 \geq c_2 \\ &\omega_1, \omega_2 \geq 0 \end{aligned}$$

以上两个问题是两个互相伴随、密切相关的问题, 若称其中一个问题是原问题 (**primal problem**), 另一个则为对偶问题 (**dual problem**)。另外注意到, 原问题与对偶问题之间存在博弈关系, 即工厂是选择自行生产产品亦或是出售原材料本身就存在博弈关系。实际上 Neumann 即是从二人零和博弈得到启发而正式提出对偶理论;

3. 以上原问题和对偶问题的表达式具有明显的对称性, 称其为对称形式 (**symmetric form**) 下的原问题和对偶问题, 一般表达式为

原问题	对偶问题
$\min f = \mathbf{c}^T \mathbf{x}$	$\max z = \mathbf{b}^T \boldsymbol{\omega}$
s.t. $\mathbf{A}\mathbf{x} \geq \mathbf{b}$	s.t. $\mathbf{A}^T \boldsymbol{\omega} \leq \mathbf{c}$
$\mathbf{x} \geq \mathbf{0}$	$\boldsymbol{\omega} \geq \mathbf{0}$

(17.12)

所谓对称形式是指线性规划问题满足 1) 决策变量非负; 2) 约束条件当目标函数求最大时取 “ \leq ”, 目标函数求最小时取 “ \geq ”;

4. 进一步地, 给出标准形式线性规划问题的对偶问题。需将标准形式转化为对称形式。首先将标准形式的等值约束转化为对称形式中的 “ \geq ” 形式

$$\mathbf{A}\mathbf{x} = \mathbf{b} \implies \begin{cases} \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{cases} \implies \begin{cases} \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ -\mathbf{A}\mathbf{x} \geq -\mathbf{b} \end{cases}$$

从而可以得到标准形式线性规划问题的对偶问题

原问题	对偶问题
$\min f = \mathbf{c}^T \mathbf{x}$	$\max z = \mathbf{b}^T \boldsymbol{\omega}$
s.t. $\mathbf{A}\mathbf{x} = \mathbf{b}$	s.t. $\mathbf{A}^T \boldsymbol{\omega} \leq \mathbf{c}$
$\mathbf{x} \geq \mathbf{0}$	

(17.13)

5. 给出从任意的线性规划原问题求其对偶问题的一般对应关系:

- 若原问题求 \min , 则对偶问题求 \max ; 若原问题求 \max , 则对偶问题求 \min ;
- 若原问题求 \min , 则

- 若原问题有 m_1 个 “ \geq ” 的约束，则对偶问题有 m_1 个 “ ≤ 0 ” 的变量；
 - 若原问题有 m_2 个 “ \leq ” 的约束，则对偶问题有 m_2 个 “ ≥ 0 ” 的变量；
 - 若原问题有 m_3 个 “ $=$ ” 的约束，则对偶问题有 m_3 个无约束的变量；
 - 若原问题有 n_1 个 “ \geq ” 的变量，则对偶问题有 n_1 个 “ ≤ 0 ” 的约束；
 - 若原问题有 n_2 个 “ \leq ” 的变量，则对偶问题有 n_2 个 “ ≥ 0 ” 的约束；
 - 若原问题有 n_3 个无约束的变量，则对偶问题有 n_3 个 “ $=$ ” 的约束。
- 若原问题求 \max , 则
 - 若原问题有 m_1 个 “ \geq ” 的约束，则对偶问题有 m_1 个 “ ≤ 0 ” 的变量；
 - 若原问题有 m_2 个 “ \leq ” 的约束，则对偶问题有 m_2 个 “ ≥ 0 ” 的变量；
 - 若原问题有 m_3 个 “ $=$ ” 的约束，则对偶问题有 m_3 个无约束的变量；
 - 若原问题有 n_1 个 “ \geq ” 的变量，则对偶问题有 n_1 个 “ ≥ 0 ” 的约束；
 - 若原问题有 n_2 个 “ \leq ” 的变量，则对偶问题有 n_2 个 “ ≤ 0 ” 的约束；
 - 若原问题有 n_3 个无约束的变量，则对偶问题有 n_3 个 “ $=$ ” 的约束。

6. 对偶问题的对偶即为原问题。

对偶问题求解算例

例：求解以下线性规划原问题的对偶问题

$$\begin{aligned} \min f &= 3x_1 + 2x_2 - 8x_3 + 5x_4 \\ \text{s.t. } &x_1 + 8x_2 + x_3 - x_4 = -2 \\ &3x_1 - 6x_2 + 5x_3 - 2x_4 \geq 3 \\ &7x_1 - 3x_2 - x_3 + 3x_4 \leq -1 \\ &x_1, x_3 \leq 0 \end{aligned}$$

解：根据原问题与对偶问题的对应关系，得到其对偶问题如下

$$\begin{aligned} \max z &= -2\omega_1 + 3\omega_2 - \omega_3 \\ \text{s.t. } &\omega_1 + 3\omega_2 + 7\omega_3 \geq 3 \\ &8\omega_1 - 6\omega_2 - 3\omega_3 = 2 \\ &\omega_1 + 5\omega_2 - \omega_3 \geq -8 \\ &-\omega_1 - 2\omega_2 + 3\omega_3 = 5 \\ &\omega_2 \geq 0 \quad \omega_3 \leq 0 \end{aligned}$$

17.2.2 对偶理论

1. (弱对偶定理, **weak duality theorem**) 如果 x 是原问题 (求最小) 的可行解, ω 是对偶问题的可行解, 则

$$c^T x \geq b^T \omega$$

证：不妨假设原问题为标准形式。则因为 x, ω 分别为原问题和对偶问题的可行解，有

$$\begin{cases} Ax = b \\ A^T \omega \leq c \end{cases} \Rightarrow c^T \geq (A^T \omega)^T = \omega^T A \Rightarrow c^T x \geq \omega^T A x = \omega^T (Ax) \Rightarrow c^T x \geq \omega^T b = b^T \omega$$

2. (推论) 如果原问题有可行解且目标函数只界, 则其对偶问题无可行解; 同理, 如果对偶问题有可行解且目标函数值无界, 则其原问题无可行解。

3. (最优性定理) 如果 x, ω 分别是原问题和对偶问题的可行解, 并且满足下式, 则 x, ω 分别为原问题和对偶问题的最优解。

$$c^T x = b^T \omega$$

证：假设 x^*, ω^* 分别为原问题和对偶问题的最优解，则显然有

$$c^T x \geq c^T x^* \quad b^T \omega^* \geq b^T \omega$$

又因为 x^*, ω^* 同时分别也是原问题和对偶问题的可行解，有 $c^T x^* \geq b^T \omega^*$ ，即

$$c^T x \geq c^T x^* \geq b^T \omega^* \geq b^T \omega$$

又因为 $c^T x = b^T \omega$ ，则

$$c^T x = c^T x^* = b^T \omega^* = b^T \omega \Rightarrow x = x^* \quad \omega = \omega^*$$

4. (强对偶定理) 在互为对偶的两个线性规划问题中，如果其中一个有最优解，那么另一个也有最优解，并且两者的最优目标函数值相等。

证：不妨假设原问题为标准型，且有最优解，则由单纯形法，最优解 $f_{\min} = c_B^T B^{-1} b$ ，且此时基变量与非基变量的检验数均小于等于 0，即

$$\sigma = c_B^T B^{-1} A - c^T \leq 0 \Rightarrow A^T (c_B^T B^{-1})^T \leq c$$

显然只需令 $\omega = (c_B^T B^{-1})^T$ 即有 $A^T \omega \leq c$ ，即 ω 为对偶问题的可行解，且目标函数值 z 为

$$z = b^T \omega = \omega^T b = c_B^T B^{-1} b = f_{\min}$$

由最优性定理，显然 ω 为对偶问题的最优解，且两者最优目标函数值相等。

5. (推论) 如果原问题有可行解而其对偶问题无可行解，则原问题目标函数值无界；同理，如果对偶问题有可行解而其原问题无可行解，则对偶问题目标函数值无界。

6. (松紧定理) 如果 x, ω 分别是原问题和对偶问题的可行解

- 若原问题为标准型，则 x, ω 分别为原问题和对偶问题最优解的充要条件是

$$(c^T - \omega^T A)x = 0$$

- 若原问题为对称型，则 x, ω 分别为原问题和对偶问题最优解的充要条件是

$$\begin{cases} (c^T - \omega^T A)x = 0 \\ (Ax - b)^T \omega = 0 \end{cases}$$

证：首先证原问题为标准型的情况，易知

$$(c^T - \omega^T A)x = 0 \iff c^T x = \omega^T A x = \omega^T b = b^T \omega$$

显然命题得证。进一步地证原问题为对称型的情况。首先观察新增的约束，因为 $Ax \geq b$ ，则

$$(Ax - b)^T \omega = 0 \iff Ax = b$$

代入第一道约束同样可以得到 $c^T x = b^T \omega$ ，命题得证。

7. (互补松弛条件, complementary slackness) 如果 x, ω 分别是原问题和对偶问题的可行解，则 x, ω 分别为原问题和对偶问题最优解的充要条件是，对于所有 i, j

- 若原问题为标准型

- 若 $x_j > 0$, 必有 $p_j^T \omega = c_j$;
- 若 $p_j^T \omega < c_j$, 必有 $x_j = 0$ 。

- 若原问题为对称型

- 若 $x_j > 0$, 必有 $p_j^T \omega = c_j$;
- 若 $p_j^T \omega < c_j$, 必有 $x_j = 0$;

- 若 $\omega_i > 0$, 必有 $a_i x = b_i$ ⁴;
- 若 $a_i x > b_i$, 必有 $\omega_i = 0$ 。

证: 首先证原问题为标准型的情况, 由松紧定理

$$(c^T - \omega^T A)x = 0 \iff \sum_{x_j > 0} (c^T - \omega^T A)j x_j = \sum_{x_j > 0} (c_j - \omega^T p_j) x_j = \sum_{x_j > 0} (c_j - p_j^T \omega) x_j = 0 \iff c_j = p_j^T \omega \quad \forall x_j > 0$$

即证明“若 $x_j > 0$, 必有 $p_j^T \omega = c_j$ ”的情况。进一步地因为标准型原问题的对偶问题约束条件有

$$A^T \omega \leq c \Rightarrow (A^T \omega)_j = p_j^T \omega \leq c_j$$

即只可能存在 $p_j^T \omega < c_j$ 和 $p_j^T \omega = c_j$ 两种情况, 从而可以由同样地方法证明“若 $p_j^T \omega < c_j$, 必有 $x_j = 0$ ”。再证明原问题为对称型的情况。前两子命题证明方法同上一致, 仅证后两子命题。同样地由松紧定理

$$(Ax - b)^T \omega = 0 \iff \sum_{\omega_i > 0} [(Ax - b)^T]_i \omega_i = \sum_{\omega_i > 0} (a_i x - b_i) \omega_i = 0 \iff b_i = a_i x \quad \forall \omega_i > 0$$

即证明“若 $\omega_i > 0$, 必有 $a_i x = b_i$ ”的情况。进一步地因为对称型原问题的约束条件有

$$Ax \geq b \Rightarrow (Ax)_i = a_i x \geq b_i$$

即只可能存在 $a_i x > b_i$ 和 $a_i x = b_i$ 两种情况, 从而可以由同样地方法证明“若 $a_i x > b_i$, 必有 $\omega_i = 0$ ”。

● ● ●

互补松弛条件应用示例

互补松弛条件的一个重要应用是在已知原问题(对偶问题)最优解的情况下不通过单纯形法直接求解对偶问题(原问题)的最优解。

例: 已知如下线性规划问题的对偶问题最优解为 $\omega_1 = \frac{4}{5}, \omega_2 = \frac{3}{5}$, 试求解该问题的最优解

$$\min f = 2x_1 + 3x_2 + 5x_3 + 2x_4 + 3x_5$$

$$\text{s.t. } x_1 + x_2 + 2x_3 + x_4 + 3x_5 \geq 4$$

$$2x_1 - x_2 + 3x_3 + x_4 + x_5 \geq 3$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

解: 首先写出上述问题的对偶问题

$$\max z = 4\omega_1 + 3\omega_2$$

$$\text{s.t. } \omega_1 + 2\omega_2 \leq 2$$

$$\omega_1 - \omega_2 \leq 3$$

$$2\omega_1 + 3\omega_2 \leq 5$$

$$\omega_1 + \omega_2 \leq 2$$

$$3\omega_1 + \omega_2 \leq 3$$

$$\omega_1, \omega_2 \geq 0$$

代入对偶问题最优解, 有

$$p_1^T \omega - c_1 = \omega_1 + 2\omega_2 - 2 = 0 \Rightarrow x_1 > 0$$

$$p_2^T \omega - c_2 = \omega_1 - \omega_2 - 3 < 0 \Rightarrow x_2 = 0$$

$$p_3^T \omega - c_3 = 2\omega_1 + 3\omega_2 - 5 < 0 \Rightarrow x_3 = 0$$

$$p_4^T \omega - c_4 = \omega_1 + \omega_2 - 2 < 0 \Rightarrow x_4 = 0$$

$$p_5^T \omega - c_5 = 3\omega_1 + \omega_2 - 3 = 0 \Rightarrow x_5 > 0$$

⁴其中 p_j, a_i 分别表示矩阵 A 的列向量和行向量。

212

又因为对偶问题最优解 $\omega_1, \omega_2 > 0$, 有

$$\begin{cases} a_1x = b_1 \\ a_2x = b_2 \end{cases} \Rightarrow \begin{cases} x_1 + x_2 + 2x_3 + x_4 + 3x_5 = 4 \\ 2x_1 - x_2 + 3x_3 + x_4 + x_5 = 3 \end{cases} \Rightarrow \begin{cases} x_1 + 3x_5 = 4 \\ 2x_1 + x_5 = 3 \end{cases} \Rightarrow x_1 = x_5 = 1$$

所以原问题的最优解为 $(1 \ 0 \ 0 \ 0 \ 0 \ 1)^T$ 。

8. (定理) 原问题单纯形表的检验数行对应其对偶问题的一个基解。

证: 为方便讨论仅考虑原问题和对偶问题为对称形式的情况, 并补充松弛变量使得约束条件为等值约束, 即

原问题	对偶问题
$\min f = c^T x$	$\max z = b^T \omega$
s.t. $Ax - x_s = b$	s.t. $A^T \omega + \omega_s = c$
$x, x_s \geq 0$	$\omega, \omega_s \geq 0$

则原问题未知量 x, x_s 对应的检验数 (假设为列向量) 分别为

$$\sigma_x = A^T(B^{-1})^T c_B - c \quad \sigma_{x_s} = -(B^{-1})^T c_B$$

为证明原问题的检验数行对应对偶问题的基解, 只需将检验数作为 ω 代入对偶问题的等值约束并证明其成立即可。注意到检验数公式中包含 B, c_B , 故将对偶问题的等值约束拆为

$$A^T \omega + \omega_s = c \Rightarrow \begin{cases} B^T \omega + \omega_{s1} = c_B \\ N^T \omega + \omega_{s2} = c_N \end{cases} \quad \omega_s = \begin{bmatrix} \omega_{s1} \\ \omega_{s2} \end{bmatrix}$$

假设 A 为 $m \times n$ 维, 则 ω 为 $m \times 1$ 维, ω_s 为 $n \times 1$ 维, ω_{s1} 为 $m \times 1$ 维, ω_{s2} 为 $(n-m) \times 1$ 维。欲求对偶问题的一组基解, 必要求至少 m 个决策变量取 0 值, 则不妨令 $\omega_{s1} = 0$, 代入等值约束, 有

$$\omega = (B^T)^{-1} c_B = (B^{-1})^T c_B \quad \omega_{s2} = c_N - N^T (B^T)^{-1} c_B = c_N - N^T (B^{-1})^T c_B$$

可以看到, 对偶问题的基解是原问题检验数行的相反数。

17.2.3 影子价格 (shadow price)

1. 有强对偶定理可知, 当原问题和对偶问题同时存在可行解时, 两者具有相同的最优值, 即

$$f_{\min} = z_{\max} = \sum_{i=1}^m b_i \omega_i^* \Rightarrow \frac{\partial f_{\min}}{\partial b_i} = \omega_i^*$$

可以看到对偶变量 ω_i^* 表示第 i 个约束条件的右端项 b_i 增加一个单位时最优目标函数值 f_{\min} 的增量;

2. 在实际管理中, ω_i^* 是资源约束 b_i 价值的一种度量, 通常解释为相应资源对目标总利润的边际贡献, 被称为影子价格 (shadow price);
3. 和资源的市场价格不同, 影子价格依赖于资源的利用情况, 是未知数, 并且随数量、价格以及企业生产情况等因素的变化而变化。影子价格可以看成是一种动态价格, 是在给定生产条件下, 对系统内部资源的一种客观估价;
4. 影子价格的主要应用包括:

- 影子价格的大小反应资源在系统内的稀缺程度。考虑原问题为对称型的情况, 由互补松弛条件可知, 若影子价格 $\omega_i^* = 0$, 则原问题的第 i 项约束并未取等, 意味着资源 b_i 在系统内过剩, 在当前条件下可以不考虑这一约束; 若影子价格 $\omega_i^* > 0$, 则原问题的第 i 项约束取等, 意味着资源 b_i 属稀缺资源, 且 $\omega_i^* > 0$ 越大意味着资源 b_i 每增加一个单位对目标函数的影响越大, 即资源越稀缺;
- 影子价格实际上是一种机会成本。在完全市场经济条件下, 当某种资源的市场价格低于影子价格时, 企业应该买进该资源用于扩大再生产。因为增加该资源用于生产可获得的收益高于购买该资源所付出的费用。相反, 当市场价格高于影子价格时, 企业应卖出已有资源。随着资源的买进卖出, 其影子价格也随之变化。影子价格与市场价格保持同等水平时意味着处于平衡状态。

17.2.4 对偶单纯形法

1. 在第 17.2.2 节中已证明，在单纯形法求解线性规划问题时，单纯形表的 b 列对应原问题的基可行解， σ 行则对应对偶问题的基解。而当 $\sigma \leq 0$ 时原问题取得最优解，意味着对偶问题取得基可行解时也同时取得最优解；
2. 对称地，若保持单纯形表的 $\sigma \leq 0$ ，即保证对偶问题恒取得基可行解，则通过基变换使得 $b \geq 0$ 时（即原问题取得基可行解），也可同时求得原问题和对偶问题的最优解。以上即为对偶单纯形法的思路；
3. 对偶单纯形法的计算步骤如下：
 - 对线性规划问题进行变换，使得初始单纯形表中检验数 $\forall \sigma_j \leq 0$ ，即找到对偶问题的一个基可行解；
 - 检查 b_i
 - 若 $\forall b_i \geq 0$ ，则已求得最优解；
 - 若 $\exists b_i < 0$ ，且 $\exists a_{lj} < 0$ ，则进行基变换。出基变量 x_l 与进基变量 x_k 的确定规则如下（Bland 法则依然适用）

$$l = \arg \min_i \{b_i | b_i < 0\} \quad k = \arg \min_j \left\{ \frac{\sigma_j}{a_{lj}} \mid a_{lj} < 0 \right\}$$
 - 若 $\exists b_i < 0$ ，且 $\forall a_{lj} \geq 0$ ，则问题无可行解。

对偶单纯形法算例

例：用对偶单纯形法求解以下线性规划问题

$$\begin{aligned} \min f &= 2x_1 + x_2 \\ \text{s.t. } &3x_1 + x_2 - x_3 = 3 \\ &4x_1 + 3x_2 - x_4 = 6 \\ &x_1 + 2x_2 - x_5 = 2 \\ &x_j \geq 0 \quad (j = 1, \dots, 5) \end{aligned}$$

解：观察到等式约束中不存在单位阵，如果采用单纯形法计算需采用大 M 法或两阶段法，但注意到只需对所有等式约束均取相反数即可得到单位阵，且令初始基向量 $\mathbf{x}_B = (x_3, x_4, x_5)^T$ 可满足 $\forall \sigma_j \leq 0$

c_B	\mathbf{x}_B	2	1	0	0	0	b
		x_1	x_2	x_3	x_4	x_5	
0	x_3	-3	-1	1	0	0	-3
0	x_4	-4	-3	0	1	0	-6
0	x_5	-1	-2	0	0	1	-2
σ		-2	-1	0	0	0	0

$$\begin{aligned} \mathbf{x} &= (0, 0, -3, -6, -2)^T \quad f = 0 \\ b_4 < b_3 < b_5 < 0 &\implies x_l = x_4 \\ \min\{\sigma_j/a_{4j} | a_{4j} < 0\} &= \min\{1/2, 1/3\} = 1/2 \implies x_k = x_2 \end{aligned}$$

c_B	\mathbf{x}_B	2	1	0	0	0	b
		x_1	x_2	x_3	x_4	x_5	
0	x_3	-5/3	0	1	-1/3	0	-1
1	x_2	4/3	1	0	-1/3	0	2
0	x_5	5/3	0	0	-2/3	1	2
σ		-2/3	0	0	-1/3	0	2

$$\begin{aligned} \mathbf{x} &= (0, 2, -1, 0, 2)^T \quad f = 2 \\ b_3 < 0 &\implies x_l = x_3 \\ \min\{\sigma_j/a_{3j} | a_{3j} < 0\} &= \min\{2/5, 1\} = 2/5 \implies x_k = x_1 \end{aligned}$$

c_B	\mathbf{x}_B	2	1	0	0	0	b
		x_1	x_2	x_3	x_4	x_5	
2	x_1	1	0	-3/5	1/5	0	3/5
1	x_2	0	1	4/5	-3/5	0	6/5
0	x_5	0	0	1	-1	1	1
σ		0	0	-2/5	-1/5	0	12/5

$$\begin{aligned} \mathbf{x} &= (3/5, 6/5, 0, 0, 1)^T \quad f = 12/5 \\ \forall b_i \geq 0, \text{ 则 } \mathbf{x} &\text{ 为原问题的最优解。} \end{aligned}$$

17.2.5 敏感度分析

在求解线性规划问题时往往假定 a_{ij}, b_i, c_i 为常数，然而实际中这些参数往往是估计值，存在误差，或可能随时间而变化。因此得到最优解后将产生以下两个问题：1) 当系数有一个或几个发生变化时，已求得的线性规划问题的最优解会有什么变化；2) 这些系数在什么范围内变化时，线性规划问题的最优解或最优基不变。敏感度分析即为了回答以上问题。进一步地就各种情况进行讨论：

- 资源系数 b 变化的分析。若初始单纯形表中的 b 列变为 $b' = b + \Delta b$ ，则代入单纯形法公式，最终单纯形表中的 b 列变为 $B^{-1}(b + \Delta b)$ ，而最终单纯形表中 σ 行不受影响。因此，只需保证 $B^{-1}(b + \Delta b) \geq 0$ ，则线性规划问题的最优基不变，最优解变为 $B^{-1}(b + \Delta b)$ ；若不满足 $B^{-1}(b + \Delta b) \geq 0$ ，则转为对偶单纯形法继续优化；
- 价值系数 c 变化的分析。显然任意系数 c_k 发生变化均不会影响最终单纯形表的 b 列，但其对 σ 行的影响需要分类讨论：

- 当 c_k 对应的变量 x_k 为非基变量时，显然其只会影响检验数 σ_k ，代入单纯形法公式得到新的检验数

$$\sigma'_k = \sum_{i=1}^m c_i a_{ik} - (c_k + \Delta c_k) = \sigma_k - \Delta c_k$$

显然若 $\sigma'_k \leq 0$ ，则线性规划问题的最优解和最优基均不变；若不满足，则继续采用单纯形法优化；

- 当 c_k 对应的变量 x_k 为基变量时，显然其会影响所有非基变量对应的检验数 σ_j ，代入单纯形法公式得到新的检验数

$$\sigma'_j = \sum_{i=1, i \neq k}^m c_i a_{ij} + (c_k + \Delta c_k) a_{kj} - c_j = \sigma_j + \Delta c_k a_{kj} \quad j = m+1, \dots, n$$

显然若 $\forall \sigma'_j \leq 0$ ，则线性规划问题的最优解和最优基均不变；若不满足，则继续采用单纯形法优化。

- 技术系数 a 变化的分析。系数 a 的变化分为两种情况：

- 在原有问题的基础上引入新的变量 x_{n+1} 及对应的系数 p_{n+1} ，此时其既不会影响最终单纯形表的 b 列也不会影响 σ 行，只会产生新的系数列 p'_{n+1} 和对应的检验数 σ'_{n+1}

$$p'_{n+1} = B^{-1} p_{n+1} \quad \sigma'_{n+1} = C_B B^{-1} p_{n+1} - c_{n+1}$$

显然若 $\sigma'_{n+1} \leq 0$ ，则线性规划问题的最优解和最优基均不变；若不满足，则继续采用单纯形法优化；

- 原有问题的系数列变为 $p'_k = p_k + \Delta p_k$ ，此时既会影响最终单纯形表的 b 列也会影响 σ 行。为方便不妨首先化为上一种情况：即保持单纯形表中 p_k 不变，引入新变量 x_{n+1} ，令 $p_{n+1} = p'_k$ ，从而求得 p'_{n+1} 和对应的检验数 σ'_{n+1} 。再令新变量 x_{n+1} 入基，原变量 x_k 出基，得到新的单纯形表。若新的单纯形表已是最优解、或原问题仍为可行解、或对偶问题仍为可行解，则按正常流程得解；若原问题和对偶问题都是非可行解，需引入人工变量。

17.3 非线性规划基本概念

17.3.1 最优性条件——从拉格朗日乘数法到 KKT 条件 (Karush-Kuhn-Tucker conditions)

- 对于具有等式约束和非等式约束的一般归化问题，若目标函数或约束条件中存在非线性项，则称该问题为非线性规划问题。不同于线性规划问题，目前不存在针对非线性规划问题全局最优解的通用解法。定义非线性规划问题的标准形式如下：

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0 \\ & h_k(\mathbf{x}) = 0 \end{aligned}$$

2. 当规划问题中只含等式约束时，常基于拉格朗日法 (lagrange multiplier method) 消除等式约束，将问题转化为求解无约束非线性函数极值点的问题。引入拉格朗日乘子 λ_k ，构造拉格朗日函数 $L(\mathbf{x}, \lambda)$ 如下

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_k \lambda_k h_k(\mathbf{x})$$

求解上式疑似极值点，即得到原规划问题的极值必要条件，至于相应疑似极值点是否为真极值点需进一步检验。可以看到，以下必要条件中并不包括对 λ_k 符号的约束，因为等值约束本身即为强约束，欲使得 $L(\mathbf{x}, \lambda)$ 与 $f(\mathbf{x})$ 具有相同的最小值，必然要求 $h_k(\mathbf{x}) = 0$ ，故无需约束 λ_k 的取值（只要非零保证等式约束生效即可）

$$\frac{\partial L}{\partial x_i} = 0 \quad \frac{\partial L}{\partial \lambda_k} = 0 \quad \lambda_k \neq 0$$

上式为仅存在等式约束时的 KKT 条件，也可扩展至存在不等式约束的情况：

3. 在线性规划部分指出，对于不等式约束，可以通过增加松弛变量使之变为等式约束。因此，对于存在不等式约束的规划问题，将不等式约束转化为等式约束后应用拉格朗日法即得到 KKT 条件，KKT 条件可视为拉格朗日法的推广。对于不等式约束 $g_j(\mathbf{x}) \leq 0$ ，增加松弛变量 y_j^2 得到等式约束 $g_j(\mathbf{x}) + y_j^2 = 0$ （采用 y_j^2 而非 y_j 的优势在于无需额外增加约束 $y_j \geq 0$ ），得到拉格朗日函数 $L(\mathbf{x}, \mathbf{y}, \lambda, \mu)$ 和相应必要条件如下

$$L(\mathbf{x}, \mathbf{y}, \lambda, \mu) = f(\mathbf{x}) + \sum_j \mu_j (g_j(\mathbf{x}) + y_j^2) + \sum_k \lambda_k h_k(\mathbf{x})$$

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial x_i} = \frac{\partial f}{\partial x_i} + \sum_j \mu_j \frac{\partial g_j}{\partial x_i} + \sum_k \lambda_k \frac{\partial h_k}{\partial x_i} = 0 \\ \frac{\partial L}{\partial y_j} = 2\mu_j y_j = 0 \implies \mu_j y_j = 0 \\ \frac{\partial L}{\partial \mu_j} = g_j(\mathbf{x}) + y_j^2 = 0 \implies g_j(\mathbf{x}) \leq 0 \\ \frac{\partial L}{\partial \lambda_k} = h_k(\mathbf{x}) = 0 \\ \lambda_k \neq 0 \end{array} \right.$$

观察上述条件中的第二项，欲使 $\mu_j y_j = 0$ ，则存在以下两种情况：

- $\mu_j = 0$: 此时约束条件 $g_j(\mathbf{x})$ 不发挥作用，松弛变量 y_j 可有可无；
- $y_j = 0$: 此时松弛变量不存在，约束条件 $g_j(\mathbf{x}) = 0$ 为强约束。

综合以上两种情况，可将约束条件 $\mu_j y_j = 0$ 转化为 $\mu_j g_j(\mathbf{x}) = 0$ ，同样可对应 $\mu_j = 0$ 和 $g_j(\mathbf{x}) = 0$ 两种情况，该式也被称为互补松弛条件。因此得到新的必要条件如下

$$\frac{\partial f}{\partial x_i} + \sum_j \mu_j \frac{\partial g_j}{\partial x_i} + \sum_k \lambda_k \frac{\partial h_k}{\partial x_i} = 0 \quad \mu_j g_j(\mathbf{x}) = 0 \quad g_j(\mathbf{x}) \leq 0 \quad h_k(\mathbf{x}) = 0 \quad \lambda_k \neq 0$$

4. 需要说明的是，以上约束本身就组成了一般规划问题疑似极值点的必要条件，但是还可以进一步增强上述条件，引入对系数 μ_j 的约束，得到最终的 KKT 条件

$$\frac{\partial f}{\partial x_i} + \sum_j \mu_j \frac{\partial g_j}{\partial x_i} + \sum_k \lambda_k \frac{\partial h_k}{\partial x_i} = 0 \quad \mu_j g_j(\mathbf{x}) = 0 \quad g_j(\mathbf{x}) \leq 0 \quad h_k(\mathbf{x}) = 0 \quad \mu_j \geq 0 \quad \lambda_k \neq 0$$

可以看到，KKT 条件与拉格朗日乘数法必要条件表面上最大差别即存在对乘子 μ 的约束，然而这一约束也可以由已有的约束条件推出。由以上分析可知，若 $g_j(\mathbf{x})$ 为松约束，则必有 $\mu_j = 0$ ，因此假设所有的 $g_j(\mathbf{x})$ 均为有效约束，并证明要求 $\mu_j > 0$ 。将第一道约束写成梯度的形式

$$\nabla f + \sum_j \mu_j \nabla g_j + \sum_k \lambda_k \nabla h_k = 0 \implies -\nabla f = \sum_j \mu_j \nabla g_j + \sum_k \lambda_k \nabla h_k$$

若以上所有约束均为有效约束，则疑似极值点 \mathbf{x}^* 必然处在所有曲面 $g_j(\mathbf{x}) = 0, h_k(\mathbf{x}) = 0$ 的公共点上。结合梯度的几何定义，上述约束可以解释为 \mathbf{x}^* 在场 f 下的负梯度（最速下降方向）是其在其它场 g, h 下的

梯度的线性组合。若 $\exists \mu_j < 0$, 意味着 \mathbf{x}^* 在场 f 下的负梯度与场 g_j 下的负梯度同向, 此时给 \mathbf{x}^* 一个微小的扰动, 即可在满足 $g_j(\mathbf{x}^*) < 0$ 的情况下使得 $f(\mathbf{x}^*)$ 进一步减小。

17.3.2 一般化对偶原理推导——拉格朗日对偶

- 在本章第 17.2 节中已在线性规划的视角下简单介绍了“对偶”的概念以及相应性质。然而上文并未给出线性规划对偶问题形式的推导思路, 所介绍的对偶理论也局限于线性规划的范畴中。注意到“对偶”是运筹学中的重要概念, 本节将从更一般的视角(非线性规划)推导任意优化问题的对偶形式, 并介绍相应性质^{5,6};
- 对于任意优化问题 $\min f(x), \quad \text{s.t. } g(x) \leq 0, h(x) = 0$, 可构造拉格朗日函数 $L(x, \lambda, \mu)$ 如下

$$L(x, \lambda, \mu) = f(x) + \lambda^\top h(x) + \mu^\top g(x), \quad \mu \geq 0$$

拉格朗日函数 $L(x, \lambda, \mu)$ 对求解优化问题的一个直观作用是可以将任意带约束的优化问题松弛为无约束的优化问题, 将 λ, μ 视为超参即可由任意无约束优化算法(如梯度下降等)优化 $\min L(x|\lambda, \mu)$, 只要 λ, μ 取值合适即可得到原问题的次优解;

- 进一步理解上述拉格朗日松弛过程所依托的数学原理, 发现对于 $\forall x \in C, C$ 为原问题约束 $g(x) \leq 0, h(x) = 0$ 构成的可行域, 拉格朗日函数 $L(x, \lambda, \mu)$ 具有如下性质

$$f(x) \geq L(x, \lambda, \mu), \quad x \in C \implies f(x^*) \geq L(x^*, \lambda, \mu) \geq \min_{x \in C} L(x, \lambda, \mu) \geq \min_x L(x, \lambda, \mu) = g(\lambda, \mu)$$

式中 x^* 表示原问题的最优解, 又记 $\hat{x}^* = \arg \min_x L(x, \lambda, \mu)$ 。可以看到, 无论 λ, μ 取何值, 均有 $f(x^*) \geq g(\lambda, \mu)$, 即 $g(\lambda, \mu)$ 提供了 $f(x^*)$ 的下限估计。进而不妨猜想: 若能找到合适的 λ, μ 取值使得 $g(\lambda, \mu)$ 尽可能逼近 $f(x^*)$, 则相应次优解 \hat{x}^* 的质量也会越高, 从而将原先的求解带约束最小化问题转化为求解无约束最大化问题

$$\max_{\lambda, \mu} g(\lambda, \mu) = \max_{\lambda, \mu} \min_x L(x, \lambda, \mu), \quad \mu \geq 0$$

根据第 17.2 节介绍的“对偶”的概念可以发现, $\max_{\lambda, \mu} g(\lambda, \mu)$ 即可视为原问题 $\min f(x), \quad \text{s.t. } g(x) \leq 0, h(x) = 0$ 的对偶问题, 而函数 $g(\lambda, \mu)$ 也被称为拉格朗日对偶函数;

- 令 $\lambda^*, \mu^* = \arg \max g(\lambda, \mu)$, 进一步介绍拉格朗日对偶函数 $g(\lambda, \mu)$ 的主要性质:

- 弱对偶性 (weak duality):** 因为恒有 $f(x^*) \geq g(\lambda, \mu)$, 仍有 $f(x^*) \geq g(\lambda^*, \mu^*)$, 并称 $f(x^*) - g(\lambda^*, \mu^*)$ 为对偶间隙 (duality gap);
- 凹凸性:** 无论原问题是否为凸优化问题, 拉格朗日对偶函数 $g(\lambda, \mu)$ 恒为凹函数, 则对偶问题恒为凸优化问题。这一性质并不难证明。观察拉格朗日函数 $L(x, \lambda, \mu)$, 若以 x 为参数, 则 $L(x, \lambda, \mu)$ 可视为关于 λ, μ 的仿射函数(即线性函数)簇, 各函数的斜率和截距由参数 x 确定。而 $\min_x L(x, \lambda, \mu)$ 则表示为逐 λ, μ 对一系列线性函数取下界, 得到的 $g(\lambda, \mu)$ 显然为凹函数。该性质保证了对偶问题具有较低的求解难度, 特别是原问题非常复杂时改求其对偶问题得到次优解即具有较高的吸引力;
- 强对偶性 (strong duality):** 强对偶性要求 $f(x^*) = g(\lambda^*, \mu^*)$, 若该性质成立则可通过求解对偶问题得到原问题的最优解, 然而强对偶性并非对所有优化问题均成立。**Slater 条件 (Slater's condition)** 提供了强对偶性成立的充分必要条件, 其指出若原问题是凸优化问题, 且 $\exists x \in \text{relint}(C)$ 使得约束条件满足 $g(x) < 0, h(x) = 0$, 则强对偶性成立。其中 $\text{relint}(C)$ 表示原问题可行域 C 的“相对内部”, 即要求 x 不在 C 的边界上;
- 微分特性:** 了解拉格朗日对偶函数 $g(\lambda, \mu)$ 的微分特性对求解对偶问题 $\max_{\lambda, \mu \geq 0} g(\lambda, \mu)$ 至关重要⁷。假设拉格朗日函数 $L(x, \lambda, \mu)$ 中 x 的定义域为非空紧致集合 (nonempty compact set)⁸, 记 $\hat{x}^* = \arg \min_x L(x, \lambda, \mu)$, 则有 $g(\lambda, \mu) = L(\hat{x}^*, \lambda, \mu)$, 当 $f(x), h(x), g(x)$ 均连续时, 有:

⁵【凸优化笔记 6】-拉格朗日对偶 (Lagrange duality)、KKT 条件: <https://zhuanlan.zhihu.com/p/103961917>

⁶如何理解拉格朗日对偶函数: <https://blog.csdn.net/sikong00/article/details/107319707/>

⁷拉格朗日对偶函数 $g(\lambda, \mu)$ 的微分特性详见 Nonlinear Programming: Theory and Algorithms (Bazaraa et al., 2006) 的引理 6.3.2、定理 6.3.3 和定理 6.3.4。

⁸紧致集是拓扑分析领域的重要基础概念, 在欧式几何空间中, 一个集合是紧致集意味着其封闭且有界。

- 若 \hat{x}^* 是 $\min_x L(x, \lambda, \mu)$ 的唯一最优解，则 $g(\lambda, \mu)$ 可微，且有

$$\nabla_{\lambda} g(\lambda, \mu) = h(\hat{x}^*) \quad \nabla_{\mu} g(\lambda, \mu) = g(\hat{x}^*), \quad \hat{x}^* = \arg \min_x L(x, \lambda, \mu)$$

- 若 \hat{x}^* 不是 $\min_x L(x, \lambda, \mu)$ 的唯一最优解，则 $g(\lambda, \mu)$ 不可微，但可计算次微分（详见第 17.4 节）

$$h(\hat{x}^*) \in \partial_{\lambda} g(\lambda, \mu) \quad g(\hat{x}^*) \in \partial_{\mu} g(\lambda, \mu), \quad \hat{x}^* \in \arg \min_x L(x, \lambda, \mu)$$

上式中 $\partial_{\lambda} g(\lambda, \mu), \partial_{\mu} g(\lambda, \mu)$ 表示 $g(\lambda, \mu)$ 的次微分。

5. 拉格朗日函数不仅可用于导出原问题的对偶形式，也可表示原问题本身

$$(\text{原问题}) \quad \min_x \max_{\lambda, \mu \geq 0} L(x, \lambda, \mu) \iff (\text{对偶问题}) \quad \max_{\lambda, \mu \geq 0} \min_x L(x, \lambda, \mu)$$

6. 第 17.3.1 节中从最优化的角度导出了 KKT 条件——即任意优化问题最优解所满足的性质，其中 λ, μ 的意义仅为拉格朗日乘子。本小节从对偶的角度出发，可进一步理解 KKT 条件——当强对偶性成立，且 $L(x, \lambda, \mu)$ 对 x 可微时，则对于原问题最优解 x^* 和对偶问题最优解 λ^*, μ^* ，恒有 x^*, λ^*, μ^* 满足 KKT 条件：

$$\begin{cases} \frac{\partial L(x, \lambda^*, \mu^*)}{\partial x} \Big|_{x=x^*} = 0 & (\text{stationarity, 稳定性条件}) \\ \mu_j^* g_j(x^*) = 0 & (\text{complementary slackness, 互补松弛条件}) \\ g_j(x^*) \leq 0, h_k(x^*) = 0 & (\text{primal feasibility, 原问题可行性}) \\ \mu_j^* \geq 0 & (\text{dual feasibility, 对偶问题可行性}) \end{cases}$$

上式中稳定性条件、原问题可行性条件和对偶问题可行性均显而易见，以下简单证明互补松弛条件成立。当强对偶性成立时，显然有

$$f(x^*) = g(\lambda^*, \mu^*) = \min_x L(x, \lambda^*, \mu^*) \leq L(x^*, \lambda^*, \mu^*) = f(x^*) + \sum_j \mu_j^* g_j(x^*) + \sum_k \lambda_k^* h_k(x^*) \leq f(x^*)$$

观察上式，并考虑到 $\mu_j \geq 0, g_j(x^*) \leq 0, h_k(x^*) = 0$ ，显然有

$$\sum_j \mu_j^* g_j(x^*) + \sum_k \lambda_k^* h_k(x^*) = 0 \implies \mu_j^* g_j(x^*) = 0$$

在强对偶性成立——即原问题与对偶问题等价的情况下，KKT 条件建立了原问题与对偶问题最优解之间的关系，进而提供了基于对偶问题最优解快速确定原问题最优解的方式。需要说明的是，对于一般原问题，KKT 条件仅为 x^*, λ^*, μ^* 是最优解的必要条件，而当原问题为凸优化问题时，KKT 条件为充要条件。

算例：基于拉格朗日对偶函数推导线性规划对偶问题

线性规划问题是一般非线性规划问题的特例，故理论上可以由拉格朗日对偶函数推导各类型线性规划问题的对偶形式，且因为线性规划为凸优化问题，满足 Slater 条件，故原问题与其对偶问题等价。考虑标准形式的线性规划问题

$$\min_x f = c^T x, \quad \text{s.t. } Ax = b, \quad x \geq 0$$

构造拉格朗日函数 $L(x, \lambda, \mu)$

$$L(x, \lambda, \mu) = c^T x + \lambda^T (Ax - b) - \mu^T x = -\lambda^T b + (c + A^T \lambda - \mu)^T x, \quad \mu \geq 0$$

进而求解拉格朗日对偶函数 $g(\lambda, \mu)$

$$g(\lambda, \mu) = \min_x L(x, \lambda, \mu) = \begin{cases} -\lambda^T b, & c + A^T \lambda - \mu = 0 \\ -\infty, & c + A^T \lambda - \mu \neq 0 \end{cases}$$

由此可知其对偶问题表示为

$$\begin{aligned} \max_{\lambda, \mu} g(\lambda, \mu) &= -\lambda^T b \\ \text{s.t. } c + A^T \lambda - \mu &= 0, \quad \mu \geq 0 \end{aligned} \implies \begin{aligned} \max_{\lambda} -\lambda^T b \\ \text{s.t. } c + A^T \lambda &\geq 0 \end{aligned} \implies \begin{aligned} \max_{\lambda} \lambda^T b \\ \text{s.t. } A^T \lambda &\leq c \end{aligned}$$

上式即为标准型线性规划问题的对偶形式，同理可推导第 17.2 节中介绍的其它形式线性规划问题的对偶形式。

17.3.3 对偶问题求解算法——优化算法的对偶形式

- 第 17.3.2 节中已指出，当强对偶性成立时，对于任意优化问题均可通过求解其对偶问题得到原问题的最优解，且对偶问题为无约束凸优化问题，在某些情况下更易求解。本小节更进一步，介绍如何将常见的优化算法应用于对偶问题求解。第 17.2 节介绍了线性规划问题单纯形法的对偶形式——对偶单纯形法，实际上几乎所有的优化算法均有相应的对偶形式；
- 以常见的梯度下降法为例，介绍其对偶形式——对偶梯度下降 (dual gradient descent)，即基于梯度下降法求解如下对偶问题

$$\max_{\lambda, \mu \geq 0} g(\lambda, \mu) = \max_{\lambda, \mu \geq 0} \min_x L(x, \lambda, \mu) = \max_{\lambda, \mu \geq 0} \min_x f(x) + \lambda^\top h(x) + \mu^\top g(x)$$

基于第 17.3.2 节介绍的 $g(\lambda, \mu)$ 微分特性，考虑到 $g(\lambda, \mu)$ 可能不可微的情况，代入更一般的次梯度下降算法 $x_k \in x_{k-1} - t_k \partial f(x_{k-1})$ (详见第 17.4 节)，则 λ, μ 的更新规则有

$$\begin{cases} x_{k+1} \in \arg \min_x L(x, \lambda_k, \mu_k) \\ \lambda_{k+1} = \lambda_k + t_{k+1} h(x_{k+1}), \quad \mu_{k+1} = \max\{\mu_k + t_{k+1} g(x_{k+1}), 0\} \end{cases} \quad (\text{对偶梯度下降})$$

上式即为一般形式的对偶梯度下降算法，因为梯度下降法在更新解 x_k 时基于上一轮解 x_{k-1} 的梯度信息 $\partial f(x_{k-1})$ ，故对偶梯度下降法在更新 x_{k+1} 也是基于上一轮的结果 λ_k, μ_k 。算法易于实现，便于将无约束优化算法应用于带约束问题求解；

- 当 $g(\lambda, \mu)$ 不可微时，对偶梯度下降算法优化效率较慢，此时可尝试基于近端点法（见第 17.4 节）求解，得到近端点法的对偶形式——对偶近端点法 (dual proximal point method)，代入近端点法的解更新公式 $x_k \in x_{k-1} - t_k \partial f(x_k)$ ，则对偶近端点法的 λ, μ 更新规则有

$$\begin{cases} x_{k+1} \in \arg \min_x L(x, \lambda_{k+1}, \mu_{k+1}) \\ \lambda_{k+1} = \lambda_k + t_{k+1} h(x_{k+1}), \quad \mu_{k+1} = \max\{\mu_k + t_{k+1} g(x_{k+1}), 0\} \end{cases} \quad (\text{对偶近端点法})$$

可以看到，因为近端点法与梯度下降法的数学形式非常相似，故两者的对偶形式也高度相似。因为近端点法在更新解 x_k 时依赖同属于解 x_k 的梯度信息 $\partial f(x_k)$ ，故对偶近端点法在更新 x_{k+1} 时也是基于本轮的结果 λ_{k+1}, μ_{k+1} ，从而造成上式互相嵌套，不易直接求解。为此需要尝试解耦，即在求解 x_{k+1} 时避免使用 λ_{k+1}, μ_{k+1} 。暂不考虑不等式约束 $g(x) \leq 0$ ，并注意到 $\lambda_{k+1} = \lambda_k + t_{k+1} h(x_{k+1})$

$$\min_x L(x, \lambda_{k+1}) = \min_x f(x) + \lambda_{k+1}^\top h(x) = \min_x f(x) + [\lambda_k + t_{k+1} h(x_{k+1})]^\top h(x)$$

则 $x_{k+1} \in \arg \min_x L(x, \lambda_{k+1})$ 等价于

$$\begin{aligned} x_{k+1} \in \arg \min_x f(x) + [\lambda_k + t_{k+1} h(x_{k+1})]^\top h(x) &\implies 0 \in \nabla f(x_{k+1}) + \lambda_k \nabla h(x_{k+1}) + t_{k+1} h^\top(x_{k+1}) \nabla h(x_{k+1}) \\ &\implies x_{k+1} \in \arg \min_x f(x) + \lambda_k h(x) + \frac{t_{k+1}}{2} \|h(x)\|_2^2 \\ &\implies x_{k+1} \in \arg \min_x L_{t_{k+1}}(x, \lambda_k) \end{aligned}$$

此时无需 λ_{k+1} 即可求解 x_{k+1} ，再由 x_{k+1} 求解 λ_{k+1} ，从而实现对偶近端点法的解耦。定义 $L_{t_{k+1}}(x, \lambda_k)$ 为增广拉格朗日函数 (augmented Lagrangian function)。与拉格朗日函数相比，增广拉格朗日函数在形式上引入了关于约束的惩罚项 $\|h(x)\|_2^2$ ，一方面使得每一列迭代的解 x_{k+1} 更接近约束 $h(x) = 0$ ，另一方面也增强了目标函数的凹凸性。基于增广拉格朗日函数，则对偶近端点法的更新公式改写为

$$x_{k+1} \in \arg \min_x L_{t_{k+1}}(x, \lambda_k), \quad \lambda_{k+1} = \lambda_k + t_{k+1} h(x_{k+1})$$

进一步考虑不等式约束的情况，因为此时要求相应的拉格朗日乘子 $\mu \geq 0$ ，则相对简单的处理方法是将不等式约束转为等式约束 $g(x) \leq 0 \Rightarrow g(x) + s^2 = 0$ ，此时增广拉格朗日函数 $L_{t_{k+1}}(x, s, \lambda_k, \mu_k)$ 写为

$$L_{t_{k+1}}(x, s, \lambda_k, \mu_k) = f(x) + \lambda_k h(x) + \mu_k g(x) + \frac{t_{k+1}}{2} (\|h(x)\|_2^2 + \|g(x) + s^2\|_2^2)$$

则得到完整形式的更新公式

$$\begin{cases} x_{k+1}, s_{k+1} \in \arg \min_{x, s} L_{t_{k+1}}(x, s, \lambda_k, \mu_k) \\ \lambda_{k+1} = \lambda_k + t_{k+1} h(x_{k+1}), \quad \mu_{k+1} = \mu_k + t_{k+1} [g(x_{k+1}) + s_{k+1}^2] \end{cases} \quad (\text{对偶近端点法 \& 增广拉格朗日法})$$

上式又称为增广拉格朗日法 (augmented Lagrangian method, ALM)，也称为乘子法 (method of multipliers)。即对偶问题的近端点法等价于原问题的对偶拉格朗日法。因此对偶近端点法相较于对偶梯度下降的效率优势不仅可以解释为近端点法相较于次梯度下降法的效率优势，也可以解释为增广拉格朗日函数相较于拉格朗日函数更凸所带来的收敛性优势。

17.3.4 线搜索 (Line search) 优化与信赖域 (Trust region) 优化

- 因为大多数非线性规划模型均不存在解析地最优解法，因此常通过迭代搜索求解非线性规划问题的数值近似解。算法的基本迭代过程均是：1) 确定解 x_i ；2) 更新解 x_{i+1} 使得目标函数 $f(x_{i+1})$ 下降（或上升）。经典的优化策略可分为线搜索 (line search) 和信赖域 (trust region) 两大类，两者的差异在于解 x_{i+1} 的更新思路；
- 首先介绍线搜索算法。算法更新解的思路在于首先确定解 x_i 的更新方向 d_{i+1} ，再结合更新步长 $\alpha_i > 0$ 得到新解 x_{i+1}

$$x_{i+1} = x_i + \alpha_i \cdot d_{i+1} \quad \alpha_i > 0$$

显然对于线搜索算法，其核心在于更新方向 d_{i+1} 的确定：

- 经典的 MSA 算法及其改进版本 MSWA 算法（详见第 10.7.3 节）即是典型的线搜索算法。算法直接基于原解 x_i 和目标函数 $f(x_i)$ 得到更新方向 d_{i+1} 而无需目标函数 $f(x)$ 的梯度信息，属于无梯度优化 (derivative-free optimization) 算法。除此之外大多数元启发优化算法如遗传算法（第 19.3 节）、差分进化（第 19.4 节）等等均属于无需梯度信息的线搜索算法；
- 更新方向 d_{i+1} 可以更直观地由目标函数 $f(x)$ 的一阶梯度信息 $\nabla f(x_i)$ 确定。 $\nabla f(x_i)$ 的反方向即为 $f(x)$ 于 x_i 处的最速下降方向。机器学习方法中最常用的最速下降法 (steepest descent method) 即是典型的基于目标函数一阶梯度的线搜索方法。集成学习方法论中的经典 GBDT 模型本质上也是基于最速下降更新决策树模型（第 26.3 节）。当考虑约束时，则最速下降法转化为 Frank-Wolfe 算法（第 10.7.1 节）；
- 目标函数 $f(x)$ 的一阶梯度 $\nabla f(x_i)$ 的反方向指向 $f(x)$ 于 x_i 处的最速下降方向，然而该方向往往不是最优解 x^* 所在的方向。为在确定更新方向 d_{i+1} 时更充分地捕捉目标函数 $f(x)$ 的全局信息，常对其做二阶近似。因为二阶近似后得到的函数为二次凸函数，可以方便地同时求解最优更新方向与更新步长 $\delta_{i+1} = \alpha_i \cdot d_{i+1}$

$$f(x_i + \delta) \approx f(x_i) + \delta^T g_i + \frac{1}{2} \delta^T h_i \delta \implies \delta_{i+1} = \alpha_i \cdot d_{i+1} = -\frac{g_i}{h_i} \quad g_i = \nabla f(x_i) \quad h_i = \nabla^2 f(x_i)$$

该方法称为牛顿法 (Newton method)。集成学习方法论中的 XGBoost 模型本质上也是基于二阶梯度信息更新决策树模型（第 26.4 节）。

- 不同于线搜索方法，信赖域方法以直接求解 x_i 的更新量 x_{i+1} 为目标，策略可以表示为：首先基于一简单模型 $m_i(x)$ 近似替代 $f(x)$ ，而后在给定信赖域 Δ_i 下确定最优更新量 δ_{i+1} 。其数学形式可以表示为

$$\delta_{i+1} = \arg \min_{\delta} m_i(\delta) \quad m_i(x) = f(x_i) + \delta^T \nabla f(x_i) + \frac{1}{2} \delta^T B_i \delta \quad \|\delta\| \leq \Delta_i$$

为便于求解, $m_i(x)$ 常表示为二阶多项式的形式。注意到若 $B_i = \nabla^2 f(x_i)$, 在不考虑信赖域 Δ_i 的情况下即为牛顿法, 而相应的信赖域方法则为信赖域牛顿法 (**trust-region Newton method**)。信赖域方法的核心在于信赖域 Δ_i 的确定。增大或缩小 Δ_i 可以得到不同 x_{i+1} , 与此同时 $m_i(x)$ 与 $f(x)$ 于信赖域内的近似程度也会有变化。定义 r_i 表示 $m_i(x)$ 与 $f(x)$ 于信赖域内的近似程度, r_i 越接近 1 表示 $m_i(x)$ 对 $f(x)$ 的近似效果越好

$$r_i(x) = \frac{\Delta f_i}{\Delta m_i} = \frac{f(x_i) - f(x_{i+1})}{m_i(0) - m_i(x_{i+1} - x_i)}$$

- 若 r_i 小于 0: 因为一般情况下分母不会小于 0 (假设求最小值), 则意味着分子小于 0, 即近似效果极差。此时应不更新 x_i , 同时缩小 Δ_i 提升近似效果重新计算 x_{i+1} ;
- 若 r_i 大于 0 但接近 0: 说明更新值 x_{i+1} 使得目标函数下降, 但近似效果依然较差, 因此仍需缩小 Δ_i , 但可以更新 x_i ;
- 若 r_i 大于 0 但小于 1: 说明近似效果可以接受, 此时可保持 Δ_i , 并更新 x_i ;
- 若 r_i 大于 0 但接近 1: 说明近似效果极好, 意味着信赖域可能偏小, 此时可增大 Δ_i , 并更新 x_i 。

4. 相比于线搜索方法, 信赖域方法可以更自然地扩展至具有正定 Hessian 矩阵的非二次型模型;

5. 关于信赖域方法的具体应用可参考第 7.6 节。

17.4 次梯度 (**Sub-gradient**) 与近端梯度下降 (**Proximal gradient descent**)

17.4.1 次梯度最优性条件与次梯度下降优化

1. “梯度”是经典凸优化算法中的重要概念, 其不仅提供了优化迭代的更新方向, 也给出了解的最优性判据——即对于凸优化问题 $\min_x f(x)$, 若 $f(x)$ 可微, 则 x^* 是最优解的充要条件有 $\nabla f(x^*) = 0$ 。然而在实际建模过程中, 可能存在目标函数 $f(x)$ 是凸函数但不可微的情况, 此时即需要扩展梯度的概念, 引入“次梯度 (**sub-gradient**)”⁹;
2. 对于凸函数 $f(x)$, 向量 g 是其在点 x 处的次梯度当且仅当对于定义域内的任意 y 满足

$$f(y) \geq f(x) + g^\top (y - x)$$

易知若 $f(x)$ 在 x 处可微, 则该点处的次梯度与梯度一致 $g = \nabla f(x)$; 若 $f(x)$ 在 x 处可微, 则该点处的次梯度为一个集合。称 $f(x)$ 于 x 处的所有次梯度集合为该点处的次微分 (**sub-differential**), 记为 $\partial f(x)$, 显然 $\partial f(x)$ 为凸集。需要注意的是, 梯度必然是凸函数于相应邻域内的增大方向, 而次梯度不一定;

3. 通过引入次梯度即可将传统的无约束凸优化问题最优性条件扩展至目标函数不可微的情况——对于凸优化问题 $\min_x f(x)$, 则 x^* 是最优解的充要条件有 $0 \in \partial f(x^*)$;

直接基于次梯度最优性条件的不可微凸优化算例: 软阈值问题与 Lasso 回归

首先考虑如下优化问题

$$\min_x f(x) = \frac{1}{2} \|y - x\|_2^2 + \lambda \|x\|_1, \quad (\text{软阈值问题})$$

可以看到因为 L1 范数 $\|x\|_1$ 的存在上式目标函数不可微, 但可计算其各点处的次微分 $\partial f(x)$ 。首先考虑 $\|x\|_1$ 的次微分 $\partial\|x\|_1$ 。逐元素考虑, 显然有

$$g_i \in \begin{cases} \{\text{sign}(x_i)\} & x_i \neq 0 \\ [-1, 1] & x_i = 0 \end{cases}$$

式中 g_i 为 $\|x\|_1$ 对元素 x_i 的次梯度。由此即可计算 $\partial f(x)$

$$\partial f(x) = \frac{1}{2} \partial\|y - x\|_2^2 + \lambda \partial\|x\|_1 = \frac{1}{2} \nabla\|y - x\|_2^2 + \lambda \partial\|x\|_1 = x - y + \lambda \partial\|x\|_1$$

⁹机器学习中的数学理论 2: 近端梯度下降 <https://zhuanlan.zhihu.com/p/277041051>

代入最优化条件 $0 \in \partial f(x^*)$, 则有

$$y - x^* \in \lambda \partial \|x^*\|_1 \implies \begin{cases} y_i - x_i^* = \lambda \cdot \text{sign}(x_i^*) & x_i^* \neq 0 \\ |y_i - x_i^*| \leq \lambda & x_i^* = 0 \end{cases} \implies x_i^* = S_\lambda(y_i) = \text{sign}(y_i) \cdot \max\{|y_i| - \lambda, 0\} = \begin{cases} y_i - \lambda & y_i > \lambda \\ 0 & |y_i| \leq \lambda \\ y_i + \lambda & y_i < -\lambda \end{cases}$$

上式中函数 $S_\lambda(\cdot)$ 称为软阈值函数 (soft-threshold function), λ 为函数参数, 也是软阈值大小, 相应的优化问题称为软阈值问题。观察函数 $S_\lambda(\cdot)$ 的图像可知, 所谓软阈值问题, 即是对任意输入 x , 若其绝对值不超过阈值 λ 则直接输出 0, 若超过 λ 则令其向 0 偏移 λ 作为输出。可以看到, 上述软阈值问题与经典的 Lasso 回归模型具有相似的数学形式

$$\min_w g(w) = \frac{1}{2} \|y - Xw\|_2^2 + \lambda \|w\|_1, \quad (\text{Lasso 回归问题})$$

则同样地有次微分 $\partial g(w)$

$$\partial g(w) = \frac{1}{2} \partial \|y - Xw\|_2^2 + \lambda \partial \|w\|_1 = X^\top (Xw - y) + \lambda \partial \|w\|_1$$

令 X_i 表示样本矩阵 X 的列向量, 则同样基于最优化条件 $0 \in \partial g(w^*)$, 有

$$\begin{cases} X_i^\top (y - Xw^*) = \lambda \cdot \text{sign}(w_i^*) & w_i^* \neq 0 \\ |X_i^\top (y - Xw^*)| \leq \lambda & w_i^* = 0 \end{cases}$$

上式无法实现向量 w 各元素的解耦, 故对于 Lasso 回归问题无法直接基于最优化条件得到 w^* 的解析表达式。

4. 由上述算例可知, 存在大量不可微凸优化问题无法直接基于最优化条件得到最优解析解, 此时需可以基于次梯度概念拓展传统的梯度下降算法, 从而计算数值最优解。迭代公式表示为

$$x_{best,k} = \arg \min_{0 \leq i \leq k} f(x_i), \quad x_k = x_{k-1} - t_k g_{k-1}, \quad g_{k-1} \in \partial f(x_{k-1})$$

上式中 $x_k, x_{best,k}$ 分别表示第 k 轮迭代时的解和经过 k 轮迭代后的最优解; g_k 为 k 轮迭代时的次梯度; t_k 为 k 轮迭代时的更新步长, 可设为固定步长, 也可设为动态递减步长, 但为避免步长递减过快或过慢, 一般要求满足 $\sum_{k=0}^{\infty} t_k^2 < \infty$, $\sum_{k=0}^{\infty} t_k = \infty$;

5. 基于上述更新公式理论上即可求解目标函数不可微的无约束凸优化问题, 然而因为上式收敛速度较慢, 仅具有理论指导意义, 以下分析上述次梯度下降算法的收敛效果。记 x^*, x_k 分别表示问题的最优解和第 k 轮迭代时的数值解, 并代入 $x_k - x_{k-1} = -t_k g_{k-1}$, 则误差有

$$\|x_k - x^*\|_2^2 = \|x_{k-1} - x^*\|_2^2 + \|x_k - x_{k-1}\|_2^2 + 2(x_k - x_{k-1})^\top (x_{k-1} - x^*) = \|x_{k-1} - x^*\|_2^2 + t_k^2 \|g_{k-1}\|_2^2 - 2t_k g_{k-1}^\top (x_{k-1} - x^*)$$

为评价所述迭代公式的收敛性, 需要估计上式误差的上限。根据次梯度定义有 $f(x^*) - f(x_{k-1}) \geq g_{k-1}^\top (x^* - x_{k-1})$, 则

$$\|x_k - x^*\|_2^2 = \|x_{k-1} - x^*\|_2^2 + t_k^2 \|g_{k-1}\|_2^2 - 2t_k g_{k-1}^\top (x_{k-1} - x^*) \leq \|x_{k-1} - x^*\|_2^2 + t_k^2 \|g_{k-1}\|_2^2 - 2t_k (f(x_{k-1}) - f(x^*))$$

上式实际上给出了关于 $\|x_k - x^*\|_2^2$ 的递推不等式, 则有

$$\|x_k - x^*\|_2^2 \leq \|x_{k-1} - x^*\|_2^2 + t_k^2 \|g_{k-1}\|_2^2 - 2t_k (f(x_{k-1}) - f(x^*)) \leq \|x_0 - x^*\|_2^2 + \sum_{i=1}^k t_i^2 \|g_{i-1}\|_2^2 - 2 \sum_{i=1}^k t_i (f(x_{i-1}) - f(x^*))$$

注意到算法的最优数值解为 $x_{best,k}$ 而非 x_k , 故需将 $x_{best,k}$ 引入上述不等式。由定义 $f(x_{best,k}) \leq f(x_i)$, 则

$$\|x_k - x^*\|_2^2 \leq \|x_0 - x^*\|_2^2 + \sum_{i=1}^k t_i^2 \|g_{i-1}\|_2^2 - 2 (f(x_{best,k}) - f(x^*)) \sum_{i=1}^k t_i$$

上式中 x_0, x^*, t_i 均为问题参数, $x_{best,k}$ 为算法输出, 其它中间变量需进一步消去以方便讨论。注意到 $\|x_k - x^*\|_2^2 \geq 0$, 并假设存在正常数 $G \geq \{\|g_{i-1}\|_2, \forall i\}$, 则上式可进一步改写为

$$2 (f(x_{best,k}) - f(x^*)) \sum_{i=1}^k t_i \leq \|x_0 - x^*\|_2^2 + G^2 \sum_{i=1}^k t_i^2 \implies f(x_{best,k}) - f(x^*) \leq \frac{R^2 + G^2 \sum_{i=1}^k t_i^2}{2 \sum_{i=1}^k t_i}$$

式中 $R = \|x_0 - x^*\|_2$ 。基于上式即可首先判断算法收敛性，首先考虑步长 t_k 为常数的情况，则有

$$\lim_{k \rightarrow \infty} f(x_{best,k}) \leq \lim_{k \rightarrow \infty} f(x^*) + \frac{R^2}{2kt} + \frac{G^2t}{2} \implies \lim_{k \rightarrow \infty} f(x_{best,k}) \leq f(x^*) + \frac{G^2t}{2}$$

此时若欲使得误差不超过 ε ，即要求 $t \leq \frac{2\varepsilon}{G^2}$ 。而当 t_k 为满足 $\sum_{k=0}^{\infty} t_k^2 < \infty$, $\sum_{k=0}^{\infty} t_k = \infty$ 的递减步长时，有

$$\lim_{k \rightarrow \infty} f(x_{best,k}) \leq \lim_{k \rightarrow \infty} f(x^*) + \frac{R^2 + G^2 \sum_{i=1}^k t_i^2}{2 \sum_{i=1}^k t_i} \implies \lim_{k \rightarrow \infty} f(x_{best,k}) = f(x^*)$$

上述推导表明，若采用固定步长 t ，则经大量迭代后误差上限将收敛至 $\frac{G^2t}{2}$ ；若采用动态递减的步长，则可保证 $x_{best,k}$ 收敛至 x^* 。进一步分析算法的收敛速度，为便于分析仅考虑固定步长的情况，欲使得经 k 轮迭代后误差上限低于 ε ，则

$$\frac{R^2}{2kt} + \frac{G^2t}{2} \leq \varepsilon \implies k \geq \frac{R^2}{2\varepsilon t - G^2t^2}$$

并考虑到 $t \leq \frac{2\varepsilon}{G^2}$ ，则可知若采用固定步长 t 进行次梯度下降，欲使得算法精度为 ε ，需要的迭代次数量级应为 $O(1/\varepsilon^2)$ ，收敛速度较慢（梯度下降法需要的迭代次数数量级为 $O(1/\varepsilon)$ ）。

17.4.2 近端点法 (proximal point method, PPM)

- 由上小节可知，尽管梯度下降法是无约束优化时最常用的算法，但当目标函数不可微时梯度下降将退化为次梯度下降，收敛速度大大降低。**近端点法 (proximal point method)** 是另一类常用的无约束优化算法，无论目标函数是否可微，算法均可以收敛速度 $O(1/\varepsilon)$ 快速优化（要求目标函数为凸函数）；
- 近端点法与梯度下降法在结构上高度相似。回顾梯度下降法，发现其解的更新规则实际上等价于如下数学优化形式

$$x_{k+1} = x_k - t_{k+1} \nabla f(x_k) \iff x_{k+1} = \arg \min_x f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2t_{k+1}} \|x - x_k\|_2^2 \quad (\text{梯度下降})$$

上式目标函数中 $f(x_k) + \nabla f(x_k)^T (x - x_k)$ 为 $f(x)$ 于 x_k 处的一阶泰勒近似，即相应位置处的切线函数，对其求最小显然是无界的，为此加入正则项 $\frac{1}{2t_{k+1}} \|x - x_k\|_2^2$ 以限制最优解 x_{k+1} 与 x_k 偏移过远，从而得到 $x_{k+1} = x_k - t_{k+1} \nabla f(x_k)$ 。上述目标函数也可整体理解为对 $f(x)$ 于 x_k 处二阶泰勒近似的近似，若令 $\frac{1}{t_{k+1}} I = \nabla^2 f(x_k)$ 即为二阶泰勒近似；

- 从梯度下降法的等价数学优化形式出发，注意到若目标函数不引入 $f(x)$ 于 x_k 处的一阶泰勒近似而是直接以 $f(x)$ 替代，即可不要求 $f(x)$ 可微，此时即为近端点法

$$x_{k+1} = \arg \min_x f(x) + \frac{1}{2t_{k+1}} \|x - x_k\|_2^2 \quad (\text{近端点法})$$

引入正则项 $\frac{1}{2t_{k+1}} \|x - x_k\|_2^2$ 后可使得目标函数为强凸，确保问题的收敛性。定义近端映射 (proximal mapping) 函数 $\text{prox}_{f,t}(\cdot)$ ，则近端点法迭代公式可简化表示为

$$x_{k+1} = \text{prox}_{f,t_k}(x_k), \quad \text{prox}_{f,t}(x) = \arg \min_z f(z) + \frac{1}{2t} \|z - x\|_2^2$$

基于次梯度最优性条件求解近端映射函数，可得到近端点法解的另一种形式

$$0 \in \partial f(x_{k+1}) + \frac{1}{2t_{k+1}} \partial \|x_{k+1} - x_k\|_2^2 \implies x_{k+1} \in x_k - t_{k+1} \partial f(x_{k+1})$$

显然，若将上式中的 $\partial f(x_{k+1})$ 改为 $\partial f(x_k)$ 近端点法即转为梯度下降法；

- 进一步研究算法的收敛速度。注意到对于近端点法有 $-\frac{1}{t_{k+1}}(x_{k+1} - x_k) \in \partial f(x_{k+1})$ ，显然 $-\frac{1}{t_{k+1}}(x_{k+1} - x_k)$ 为 $f(x)$ 于点 x_{k+1} 的次微分，又假设 $f(x)$ 为凸函数，则根据次微分定义有

$$f(x^*) \geq f(x_{k+1}) - \frac{1}{t_{k+1}}(x_{k+1} - x_k)^T (x^* - x_{k+1})$$

$$\implies t_{k+1} (f(x_{k+1}) - f(x^*)) \leq (x_{k+1} - x_k)^\top (x^* - x_{k+1}) \leq (x_{k+1} - x_k)^\top (x^* - x_{k+1}) + \frac{1}{2} \|x_k - x_{k+1}\|_2^2$$

注意到上式不等号右侧部分可作如下变换

$$\begin{aligned} (x_{k+1} - x_k)^\top (x^* - x_{k+1}) + \frac{1}{2} \|x_k - x_{k+1}\|_2^2 &= -((x_k - x^*) + (x^* - x_{k+1}))^\top (x^* - x_{k+1}) + \frac{1}{2} \|(x_k - x^*) + (x^* - x_{k+1})\|_2^2 \\ &= \frac{1}{2} \|x_k - x^*\|_2^2 - \frac{1}{2} \|x^* - x_{k+1}\|_2^2 = \frac{1}{2} \|x_k - x^*\|_2^2 - \frac{1}{2} \|x_{k+1} - x^*\|_2^2 \end{aligned}$$

则有

$$\begin{aligned} (f(x_{best,k}) - f(x^*)) \sum_{i=0}^{k-1} t_i &\leq \sum_{i=0}^{k-1} t_i (f(x_{i+1}) - f(x^*)) \\ &\leq \sum_{i=0}^{k-1} \left(\frac{1}{2} \|x_i - x^*\|_2^2 - \frac{1}{2} \|x_{i+1} - x^*\|_2^2 \right) = \frac{1}{2} \|x_0 - x^*\|_2^2 - \frac{1}{2} \|x_k - x^*\|_2^2 \leq \frac{1}{2} \|x_0 - x^*\|_2^2 \end{aligned}$$

最终得到 $f(x_{best,k}) - f(x^*)$ 的上界为

$$f(x_{best,k}) - f(x^*) \leq \frac{\|x_0 - x^*\|_2^2}{2 \sum_{i=0}^{k-1} t_i} = \frac{R^2}{2 \sum_{i=0}^{k-1} t_i}$$

因此基于近端点法优化时，只需使得迭代步长满足 $\sum_i^\infty t_i = \infty$ ，即可保证经过充分迭代后数值最优解 $f(x_{best,k})$ 可收敛至理论最优解 $f(x^*)$

$$\lim_{k \rightarrow \infty} f(x_{best,k}) \leq \lim_{k \rightarrow \infty} f(x^*) + \frac{R^2}{2 \sum_{i=0}^{k-1} t_i} \implies \lim_{k \rightarrow \infty} f(x_{best,k}) = f(x^*)$$

进一步考虑迭代步长 t_k 为定值的情况。此时

$$f(x_{best,k}) - f(x^*) \leq \frac{R^2}{2kt}$$

可知若采用固定步长 t 应用近端点法，因为不限制 t 的取值，故欲使得算法精度为 ε ，需要的迭代次数量级仅为 $O(1/\varepsilon)$ 。

17.4.3 近端梯度下降优化

1. 上文指出，当目标函数为不可微凸函数时应用近端点法较次梯度下降具有更高地求解效率。近端点法的主要难点在于近端映射函数求解。当目标函数过于复杂时，求解近端映射函数难度较大，从而限制了近端点法的应用。此时一个可行的思路是将目标函数分为可微和不可微两部分，其中可微部分可基于梯度下降求解，而不可微部分的形式也更为简单，可基于近端点法求解。近端梯度下降算法即采取上述思路，可以收敛速度 $O(1/\varepsilon)$ 快速求解满足特定数学形式的不可微凸优化问题。具体地，近端梯度下降算法适用于求解如下优化问题

$$\min_x f(x) = g(x) + h(x)$$

其中 $g(x)$ 为可微凸函数； $h(x)$ 为不可微凸函数；

2. 注意到若目标函数仅包含 $g(x)$ ，则可基于梯度下降算法优化，而梯度下降算法中解的更新规则实际上等价于如下数学优化形式

$$x_{k+1} = x_k - t \nabla g(x_k) \iff x_{k+1} = \arg \min_x g(x_k) + \nabla g(x_k)^\top (x - x_k) + \frac{1}{2t} \|x - x_k\|_2^2$$

近端梯度下降算法同样是从一般梯度下降算法的数学优化形式出发，考虑不可微凸函数 $h(x)$ 的影响，定义解的更新规则如下

$$x_{k+1} = \arg \min_x g(x_k) + \nabla g(x_k)^\top (x - x_k) + \frac{1}{2t} \|x - x_k\|_2^2 + h(x) = \arg \min_x \frac{1}{2t} \|x - (x_k - t \nabla g(x_k))\|_2^2 + h(x)$$

需要强调的是，上式中构建了两个数学规划问题，两者的目标函数并不相等，但可解得完全一致的 x_{k+1} 。进一步通过近端映射函数 $\text{prox}_{h,t}(\cdot)$ 将近端梯度下降的解更新规则简化表示为

$$x_{k+1} = \text{prox}_{h,t}(x_k - t\nabla g(x_k)), \quad \text{prox}_{h,t}(x) = \arg \min_z \frac{1}{2t} \|z - x\|_2^2 + h(z) \quad (\text{近端梯度下降})$$

由此即可理解近端梯度下降算法的优化步骤：

- 对于可行解 x_k ，首先不考虑目标函数中的不可微部分 $h(x)$ ，仅考虑 $g(x)$ 进行梯度下降，得到 \hat{x}_{k+1} ；
- 进一步考虑不可微部分 $h(x)$ ，在 \hat{x}_{k+1} 邻域内寻找使得 $h(x)$ 最小的 x_{k+1} （即求解近端映射函数）为下一轮的可行解。此时主要基于次微分最优性条件求解。



近端梯度下降优化算例：Lasso 回归问题

Lasso 回归问题目标函数即可分解为可微凸函数与不可微凸函数两部分，故可基于近端梯度下降算法求解数值解

$$\min_w f(w) = g(w) + h(w), \quad g(w) = \frac{1}{2} \|y - Xw\|_2^2, \quad h(w) = \lambda \|w\|_1$$

则近端映射函数表示为

$$\text{prox}_t(w) = \arg \min_z \frac{1}{2t} \|z - w\|_2^2 + h(w) = \arg \min_z \frac{1}{2t} \|z - w\|_2^2 + \lambda \|w\|_1 = \arg \min_z \frac{1}{2} \|z - w\|_2^2 + \lambda t \|w\|_1$$

显然上式为上小结所介绍的以 λt 为阈值的软阈值问题，其封闭解析形式表示为

$$\text{prox}_t(w) = \arg \min_z \frac{1}{2} \|z - w\|_2^2 + \lambda t \|w\|_1 = S_{\lambda t}(w) = \begin{cases} w - \lambda t & w > \lambda t \\ 0 & |w| \leq \lambda t \\ w + \lambda t & w < -\lambda t \end{cases}$$

进而可得到 Lasso 问题解的迭代更新公式：

- 对于可行解 w_k ，首先仅考虑 $g(w)$ 进行梯度下降，得到

$$\hat{w}_{k+1} = w_k - t\nabla g(w) = w_k - tX^\top(Xw_k - y)$$

- 进而代入 \hat{w}_{k+1} 求解近端映射函数，即求解软阈值问题，有

$$w_{k+1} = \text{prox}_t(\hat{w}_{k+1}) = S_{\lambda t}(\hat{w}_{k+1}) = \begin{cases} \hat{w}_{k+1} - \lambda t & \hat{w}_{k+1} > \lambda t \\ 0 & |\hat{w}_{k+1}| \leq \lambda t \\ \hat{w}_{k+1} + \lambda t & \hat{w}_{k+1} < -\lambda t \end{cases}$$

17.4.4 改进近端梯度下降优化

1. 与次梯度下降算法相比，近端梯度下降显著地提升了优化速率¹⁰。但实际上只需进行简单地改进，即可使得算法的收敛速度进一步提升至 $O(1/\sqrt{\epsilon})$ ；
2. 仅简单介绍数种加速算法中的一种：

- 首先暂考虑 $g(x)$ 进行梯度下降，而是根据之前连续两轮迭代的可行解 x_k, x_{k-1} ，引入“动量”概念有

$$v = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$$

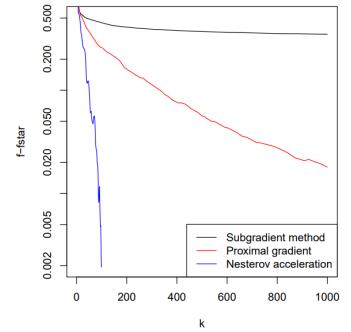
为计算上式，令初值有 $x_0 = x_{-1}$ ；

- 基于 v 对 $g(x)$ 进行梯度下降得到 $\hat{x}_{k+1} = v - t\nabla g(v)$ ；
- 最后与近端下降算法一致，基于 \hat{x}_{k+1} 求解近端映射函数，得到 $x_{k+1} = \text{prox}_{h,t}(\hat{x}_{k+1})$ 。

3. 需要说明的是，在引入“动量”概念后，改进近端梯度下降算法不再是一种下降算法。

¹⁰Proximal Gradient Descent (and Acceleration): <https://www.stat.cmu.edu/~ryantibs/convexopt/lectures/prox-grad.pdf>

图 17.1 以 Lasso 回归问题为算例对比次梯度下降算法（黑线）、常规近端梯度下降算法（红线）和改进近端梯度下降算法（蓝线）的优化效率。图中纵轴为解得的最优数值解与最优解析解之差，横轴为迭代轮数。可以看到三种算法的收敛速度存在显著的差异，自慢而快分别为 $O(1/\varepsilon^2)$, $O(1/\varepsilon)$, $O(1/\sqrt{\varepsilon})$ 。



17.5 对偶分解 (*Dual decomposition*) 与原始分解 (*Primal decomposition*)

- “分解”是运筹优化领域的重要思想，其目标是将难以优化的问题分解为若干个易于优化的子问题。最基本的应用场景即是当优化目标函数可分解为相互独立的多个部分时，有

$$x^*, z^* = \arg \min_{x,z} f(x) + g(z) \iff x^* = \arg \min_x f(x), \quad z^* = \arg \min_z g(z)$$

然而当决策变量间存在约束，或目标函数无法拆成相互独立的子目标时，则无法直接分解原问题。

- 首先考虑第一类基本情况，即目标函数可拆成多个子目标，但决策变量间存在约束，此时一个自然的思路是构造原问题的对偶问题以消去约束（详见第 17.3.2 节），并尝试对无约束的对偶问题进行分解。主要考虑原问题约束为线性等值约束的情况，则与原问题等价的对偶问题构造为

$$\max_{\lambda} \min_{x,z} L(x,z,\lambda) = f(x) + g(z) + \lambda^T (Ax + Bz - c) = \left[f(x) + \lambda^T Ax - \frac{1}{2} \lambda^T c \right] + \left[g(z) + \lambda^T Bz - \frac{1}{2} \lambda^T c \right]$$

基于对偶梯度下降算法求解（详见第 17.3.3 节），则 x, z, λ 的更新公式有

$$\begin{cases} x_{k+1}, z_{k+1} \in \arg \min_{x,z} L(x,z,\lambda_k) \\ \lambda_{k+1} = \lambda_k + t_{k+1}(Ax_{k+1} + Bz_{k+1} - c) \end{cases}$$

特别注意到求解 $\min_{x,z} L(x,z,\lambda_k)$ 时 $L(x,z,\lambda_k)$ 即可拆成 $L_1(x,\lambda_k)$ 与 $L_2(z,\lambda_k)$ 两部分关于 x, z 独立的形式，因此可分别求解 x_{k+1}, z_{k+1} ，将上式改写为

$$\begin{cases} x_{k+1} \in \arg \min_x f(x) + \lambda_k^T Ax, \quad z_{k+1} \in \arg \min_z g(z) + \lambda_k^T Bz \\ \lambda_{k+1} = \lambda_k + t_{k+1}(Ax_{k+1} + Bz_{k+1} - c) \end{cases} \quad (\text{对偶分解})$$

以上通过将原问题改写为对偶问题并对对偶问题进行分解的思路即称为对偶分解 (*dual decomposition*)。

对偶分解本质上即是可分解情况下的对偶梯度下降算法；

- 进一步考虑第二类基本情况，即决策变量间无约束，但目标函数无法完全拆成多个独立子目标的形式，即

$$\min_{x,y,z} f(x,y) + g(z,y)$$

上式中因为 y 的存在使得无法直接分解目标函数，同理若固定 y 则上式的求解难度大大降低，故称此类变量为复杂变量 (*complicating variable*)，即使得求解复杂度增大的决策变量。对此注意到

$$\min_{x,y,z} f(x,y) + g(z,y) = \min_y \left(\min_{x,z} f(x,y) + g(z,y) \right) = \min_y f(x^*(y),y) + g(z^*(y),y) = \min_y \phi_f(y) + \phi_g(y)$$

上式中 $x^*(y), z^*(y)$ 是 $\min_{x,z} f(x,y) + g(z,y)$ 的最优解，显然可分解求解 $x^*(y) = \arg \min_x f(x,y)$, $z^*(y) = \arg \min_z g(z,y)$ 。实际上，观察上述第一类基本情况所构造的等价对偶问题，发现也可写为类似的形式

$$\max_{\lambda} \min_{x,z} L(x,z,\lambda) = \max_{\lambda} G_f(\lambda) + G_g(\lambda)$$

其中 $G_f(\lambda) + G_g(\lambda)$ 为可写成两部分加和形式的拉格朗日对偶函数。因此对于目前讨论的第二类基本情况 $\min_{x,y,z} f(x,y) + g(z,y)$, 可类比代入对偶分解算法迭代公式, 得

$$\begin{cases} x_{k+1} \in \arg \min_x f(x, y_k), & z_{k+1} \in \arg \min_z g(z, y_k) \\ y_{k+1} = y_k - t_{k+1} \left(\frac{\partial f(x_{k+1}, y)}{\partial y_k} + \frac{\partial g(z_{k+1}, y)}{\partial y_k} \right) \end{cases} \quad (\text{原始分解})$$

上式直接对原问题进行分解, 故称为原始分解 (**primal decomposition**)。实际上对于所述问题, 也可构造为第一类基本情况的形式并基于对偶分解算法求解, 只需令

$$\min_{x,y,z} f(x,y) + g(z,y) \iff \min_{x,y_1,y_2,z} f(x,y_1) + g(z,y_2), \quad \text{s.t. } y_1 = y_2$$

17.6 Douglas-Rachford 分裂算法与交替方向乘子法

17.6.1 Douglas-Rachford 分裂算法 (Douglas-Rachford splitting, DRS)

- 与近端梯度下降算法 (详见第 17.4 节) 类似, DRS 算法适用于求解如下无约束凸优化问题, 其中 $f(\cdot), g(\cdot)$ 均为闭凸函数

$$\min_x f(x) + g(x)$$

- 回顾近端梯度下降算法的求解流程: 当目标函数可分解为 $f(x) + g(x)$ 两部分时, 近端梯度下降算法的每次迭代由对 $f(x)$ 的梯度下降和对 $g(x)$ 的近端优化两步组成, 从而在优化过程中实现对目标函数的解耦。不妨猜想, 是否存在一种类似的算法, 每次迭代时的两步均为近端优化, 分别针对 $f(x)$ 与 $g(x)$, 此时不仅实现了优化过程中对目标函数的解耦, 也可处理 $f(x), g(x)$ 均不可微的情况。DRS 算法即属于此类, 具体的迭代公式写为

$$x_{k+1} = \text{prox}_{g,\gamma}(z_k), \quad y_{k+1} = \text{prox}_{f,\gamma}(2x_{k+1} - z_k), \quad z_{k+1} = z_k + t_{k+1}(y_{k+1} - x_{k+1}) \quad (\text{DRS})$$

上式中 y_k, z_k 为中间变量; 近端优化时参数 $\gamma > 0$; 迭代步长 $t_k \in [0, 2]$ 并要求 $\sum_k^\infty t_k(2 - t_k) = +\infty$, 常令 $t_k = 1$; $f(x), g(x)$ 的顺序是可交换的, 但由此产生的迭代序列会不一致, 从而可能会影响收敛速度;

- 上式从形式上并不规律, 以下将尝试介绍其收敛机制。将 x_{k+1}, y_{k+1} 的定义式代入 z_{k+1} 的更新公式, 有

$$z_{k+1} = z_k + t_{k+1} [\text{prox}_{f,\gamma}(2\text{prox}_{g,\gamma}(z_k) - z_k) - \text{prox}_{g,\gamma}(z_k)]$$

定义函数 $F(z)$, 则 z_k 的更新公式可以改写为

$$z_{k+1} = z_k + t_{k+1}(F(z_k) - z_k), \quad F(z) = z + \text{prox}_{f,\gamma}(2\text{prox}_{g,\gamma}(z) - z) - \text{prox}_{g,\gamma}(z)$$

显然上式即为不动点迭代 (**fix-point iteration**) 的基本公式, 经典的 MSA 及 MSWA 算法均是其特殊形式 (见第 10.7.3 节)。若 $F(z)$ 连续且存在不动点, 则经过若干轮迭代后 z_k 将收敛至不动点

$$\lim_{k \rightarrow \infty} z_{k+1} = \lim_{k \rightarrow \infty} z_k = \lim_{k \rightarrow \infty} F(z_k) \implies \lim_{k \rightarrow \infty} z_{k+1} = \lim_{k \rightarrow \infty} z_k + t_{k+1}(y_{k+1} - x_{k+1}) \implies \lim_{k \rightarrow \infty} y_{k+1} - x_{k+1} = 0$$

即随着 z_k 收敛至 $F(z)$ 的不动点, x_k, y_k 也将收敛至同一个解。在此基础上分别考虑 x_{k+1}, y_{k+1} 的更新公式

$$\begin{cases} x_{k+1} = \text{prox}_{g,\gamma}(z_k) \\ y_{k+1} = \text{prox}_{f,\gamma}(2x_{k+1} - z_k) \end{cases} \implies \begin{cases} z_k - x_{k+1} \in \gamma \partial g(x_{k+1}) \\ 2x_{k+1} - z_k - y_{k+1} \in \gamma \partial f(y_{k+1}) \end{cases} \implies \frac{1}{\gamma}(x_{k+1} - y_{k+1}) \in \partial(f(y_{k+1}) + g(x_{k+1}))$$

当 $k \rightarrow \infty$ 时, 代入 $\lim_{k \rightarrow \infty} y_{k+1} - x_{k+1} = 0$, 则上式改写为

$$0 \in \partial(f(x_{k+1}) + g(x_{k+1})) \quad \text{or} \quad 0 \in \partial(f(y_{k+1}) + g(y_{k+1})), \quad k \rightarrow \infty$$

即随着 z_k 收敛至 $F(z)$ 的不动点, x_k, y_k 均将收敛至 $f(x) + g(x)$ 的最优解, 从而证明了算法的收敛性;

- 需要强调的是, DRS 算法的更新公式并不唯一。例如上式中每一轮迭代按 x_k, y_k, z_k 顺序依次更新。实际上也可从 y_k 开始, 按 y_k, z_k, x_k 顺序更新, 此时迭代公式写为

$$y_{k+1} = \text{prox}_{f,\gamma}(2x_k - z_k), \quad z_{k+1} = z_k + t_{k+1}(y_{k+1} - x_k), \quad x_{k+1} = \text{prox}_{g,\gamma}(z_{k+1}) \quad (\text{DRS})$$

17.6.2 交替方向乘子法 (Alternating direction method of multipliers, ADMM)

1. 交替方向乘子法 (alternating direction method of multipliers, ADMM) 提供了一个高效求解线性等式约束凸优化问题的框架。算法最早由 Glowinski & Marrocco 及 Gabay & Mercier 于 1975 和 1976 年提出, 但因当时大规模分布式计算系统及大规模优化问题尚未出现, 算法并未受到重视。至 2011 年由 Boyd 等人重新综述并证明其适用于大规模分布式优化问题, 算法才因其处理速度快、收敛性能好的优点广泛应用于统计学习、机器学习等领域。其核心思想是 “分解协调 (decomposition-coordination)”, 其中 “分解” 是指将难以求解的凸优化问题分解为若干易求解的子问题分别求解, 而 “协调” 是指将多个子问题的最优解整合得到整体问题最优解的过程。具体地, 算法适用于求解如下形式的优化问题:

$$\min_{x,z} f(x) + g(z), \quad \text{s.t. } Ax + Bz = c$$

其中 $x \in \mathbb{R}^p, z \in \mathbb{R}^q, c \in \mathbb{R}^m, A \in \mathbb{R}^{m \times p}, B \in \mathbb{R}^{m \times q}$, 且 $f(\cdot), g(\cdot)$ 均为标量凸函数。为高效求解上述问题, **ADMM** 算法结合了增广拉格朗日算法 (详见第 17.3.3 节) 和分解的基本思想;

2. 首先直接给出 ADMM 算法的迭代公式。对于上述问题, 构造增广拉格朗日函数 $L_\gamma(x, z, \lambda)$

$$L_\gamma(x, z, \lambda) = f(x) + g(z) + \lambda^\top(Ax + Bz - c) + \frac{\gamma}{2}\|Ax + Bz - c\|_2^2$$

则 ADMM 算法将 x, z 同时求解分解为以下过程

$$x_{k+1} = \arg \min_x L_\gamma(x, z_k, \lambda_k), \quad z_{k+1} = \arg \min_z L_\gamma(x_{k+1}, z, \lambda_k), \quad \lambda_{k+1} = \lambda_k + \gamma(Ax_{k+1} + Bz_{k+1} - c) \quad (\text{ADMM})$$

注意到 ADMM 算法在形式上与增广拉格朗日法高度相似。可以说, **ADMM 算法即是可分解情况下的增广拉格朗日算法**。又因为增广拉格朗日算法也称为 “乘子法”, 故交替方向乘子法 (ADMM) 由此得名;

3. 进一步介绍 ADMM 公式的推导。针对约束 $Ax + Bz = c$, 构造拉格朗日对偶问题消除

$$\begin{aligned} \max_{\lambda} \min_{x,z} L(x, z, \lambda) &= \max_{\lambda} \min_{x,z} f(x) + g(z) + \lambda^\top(Ax + Bz - c) \\ &= \min_{\lambda} - \left[\min_x f(x) + \lambda^\top Ax \right] - \left[\min_z g(z) + \lambda^\top Bz - \lambda^\top c \right] \\ &= \min_{\lambda} - \left[\min_x L_f(x, \lambda) \right] - \left[\min_z L_g(z, \lambda) \right] = \min_{\lambda} d_f(\lambda) + d_g(\lambda) \end{aligned}$$

对于目标函数可写为上述两部分加和形式的无约束优化问题, 本章已介绍多种针对性的求解算法, 如对偶分解 (见第 17.5 节)、近端梯度下降 (见第 17.4 节)、Douglas-Rachford splitting (DRS) 算法等。本问题基于 DRS 算法同时求解上述对偶问题与原问题的最优解;

4. 构造中间变量 $\hat{\lambda}_k, \bar{\lambda}_k$, 并令 DRS 算法中的迭代步长 $t_k = 1$, 则对偶问题拉格朗日乘子 λ_k 的更新规则写为

$$\hat{\lambda}_{k+1} = \text{prox}_{d_f, \gamma}(2\lambda_k - \bar{\lambda}_k), \quad \bar{\lambda}_{k+1} = \bar{\lambda}_k + \hat{\lambda}_{k+1} - \lambda_k, \quad \lambda_{k+1} = \text{prox}_{d_g, \gamma}(\bar{\lambda}_{k+1})$$

上式给出了对偶变量的迭代规则。为同时得到原问题的解, 首先观察 $\hat{\lambda}_k$ 的迭代式, 基于对偶近端点法 (增广拉格朗日法)

$$\hat{\lambda}_{k+1} = \text{prox}_{d_f, \gamma}(2\lambda_k - \bar{\lambda}_k) \Rightarrow \begin{cases} \hat{\lambda}_{k+1} = 2\lambda_k - \bar{\lambda}_k - \gamma \partial d_f(\hat{\lambda}_{k+1}) = 2\lambda_k - \bar{\lambda}_k + \gamma Ax_{k+1} \\ x_{k+1} = \arg \min_x L_{f, \gamma}(x, 2\lambda_k - \bar{\lambda}_k) = \arg \min_x f(x) + (2\lambda_k - \bar{\lambda}_k)^\top Ax + \frac{\gamma}{2}\|Ax\|_2^2 \end{cases}$$

上式中 $L_{f, \gamma}$ 为对应于拉格朗日函数 L_f 的增广拉格朗日函数。进一步观察 λ_k 的迭代式

$$\lambda_{k+1} = \text{prox}_{d_g, \gamma}(\bar{\lambda}_{k+1}) \Rightarrow \begin{cases} \lambda_{k+1} = \bar{\lambda}_{k+1} - \gamma \partial d_g(\lambda_{k+1}) = \bar{\lambda}_{k+1} + \gamma(Bz_{k+1} - c) \\ z_{k+1} = \arg \min_z L_{g, \gamma}(z, \bar{\lambda}_{k+1}) = \arg \min_z g(z) + \bar{\lambda}_{k+1}^\top (Bz - c) + \frac{\gamma}{2}\|Bz - c\|_2^2 \end{cases}$$

综上即可得到基于 DRS 算法的对偶问题和原问题的解的更新公式

$$x_{k+1} = \arg \min_x f(x) + (2\lambda_k - \bar{\lambda}_k)^\top Ax + \frac{\gamma}{2}\|Ax\|_2^2$$

$$\widehat{\lambda}_{k+1} = 2\lambda_k - \bar{\lambda}_k + \gamma A x_{k+1}$$

$$\bar{\lambda}_{k+1} = \bar{\lambda}_k + \widehat{\lambda}_{k+1} - \lambda_k$$

$$z_{k+1} = \arg \min_z g(z) + \bar{\lambda}_{k+1}^\top (Bz - c) + \frac{\gamma}{2} \|Bz - c\|_2^2$$

$$\lambda_{k+1} = \bar{\lambda}_{k+1} + \gamma (Bz_{k+1} - c)$$

5. 上述表达式过于复杂，尝试消去中间变量 $\widehat{\lambda}_k, \bar{\lambda}_k$ 。首先将 $\widehat{\lambda}_{k+1} = 2\lambda_k - \bar{\lambda}_k + \gamma A x_{k+1}$ 代入 $\bar{\lambda}_{k+1}$ 计算式

$$\bar{\lambda}_{k+1} = \bar{\lambda}_k + (2\lambda_k - \bar{\lambda}_k + \gamma A x_{k+1}) - \lambda_k = \lambda_k + \gamma A x_{k+1}$$

进而将上式分别代入 z_{k+1}, λ_{k+1} 的计算式

$$\begin{aligned} z_{k+1} &= \arg \min_z g(z) + (\lambda_k + \gamma A x_{k+1})^\top (Bz - c) + \frac{\gamma}{2} \|Bz - c\|_2^2 \\ &= \arg \min_z g(z) + \lambda_k^\top (Bz - c) + \gamma (A x_{k+1})^\top (Bz - c) + \frac{\gamma}{2} \|Bz - c\|_2^2 \\ &= \arg \min_z g(z) + \lambda_k^\top (A x_{k+1} + Bz - c) + \frac{\gamma}{2} \|A x_{k+1} + Bz - c\|_2^2 \end{aligned}$$

$$\lambda_{k+1} = \lambda_k + \gamma A x_{k+1} + \gamma (Bz_{k+1} - c) = \lambda_k + \gamma (A x_{k+1} + Bz_{k+1} - c)$$

同时将 $\lambda_{k+1} - \bar{\lambda}_{k+1} = \gamma (Bz_{k+1} - c)$ 代入 x_{k+1} 计算式

$$\begin{aligned} x_{k+1} &= \arg \min_x f(x) + (\lambda_k + \gamma (Bz_k - c))^\top Ax + \frac{\gamma}{2} \|Ax\|_2^2 \\ &= \arg \min_x f(x) + \lambda_k^\top Ax + \gamma (Bz_k - c)^\top Ax + \frac{\gamma}{2} \|Ax\|_2^2 = \arg \min_x f(x) + \lambda_k^\top (Ax + Bz_k - c) + \frac{\gamma}{2} \|Ax + Bz_k - c\|_2^2 \end{aligned}$$

以上即为 ADMM 算法的迭代公式。由此可知，原问题的 ADMM 算法等价于对偶问题的 Douglas-Rachford splitting 算法。

17.7 坐标下降 (Coordinate descent) 与块坐标下降 (Block coordinate descent)

1. 坐标下降 (coordinate descent) 是凸优化算法中除梯度下降和近端下降（第 17.4 节）外的另一类优化框架。方法源自一个非常简单的猜想：对于凸函数 f ，若得到一个解 $x^* = (x_1^*, \dots, x_n^*)$ 使得对于 $\forall i = 1, \dots, n$, $\forall \delta \in \mathbb{R}$ 均存在 $f(x^*) \leq f(x^* + \delta e_i)$ (e_i 为第 i 维元素为 1, 其它元素为 0 的 n 维向量)，则 x^* 为 $\min_x f$ 的全局最优解。若上述猜想成立，则求解多维优化问题 $\min_x f(x)$ 可分解为多次求解一维优化问题 $\min_{x_i} f(x_i|x)$ ，而每一次一维优化均是沿坐标轴方向，“坐标下降”由此得名；

2. 进一步讨论上述猜想的正确性，分两类情况讨论：

- 当凸目标函数 $f(x)$ 可微时猜想成立， x^* 为 $\min f(x)$ 的全局最优解，因为 x^* 显然满足下式

$$\nabla f(x^*) = \left(\frac{\partial f}{\partial x_1^*}, \dots, \frac{\partial f}{\partial x_n^*} \right)^\top = 0$$

- 当凸目标函数 $f(x)$ 不可微时猜想不一定成立，当且仅当 $f(x)$ 可以写为形如 $g(x) + \sum_i h_i(x_i)$ 的表达式（其中 $g(\cdot)$ 可微， $h_i(\cdot)$ 不可微）时猜想成立。证明如下

$$\begin{aligned} f(x) - f(x^*) &= g(x) - g(x^*) + \sum_i (h_i(x_i) - h_i(x_i^*)) \geq \nabla g(x^*)^\top (x - x^*) + \sum_i (h_i(x_i) - h_i(x_i^*)) \\ &= \sum_i \left(\frac{\partial g(x^*)}{\partial x_i^*} (x_i - x_i^*) + h_i(x_i) - h_i(x_i^*) \right) \end{aligned}$$

根据 x^* 的定义，对 $\forall i$ 显然均有 $\frac{\partial g(x^*)}{\partial x_i^*} (x_i - x_i^*) + h_i(x_i) - h_i(x_i^*) \geq 0$ 成立，故对 $\forall x$ 恒有 $f(x) - f(x^*) \geq 0$ 成立，意味着 $x^* = \arg \min_x f(x)$ 。

综上，当凸目标函数 $f(x)$ 可微或可写为形如 $g(x) + \sum_i h_i(x_i)$ 的表达式（其中 $g(\cdot)$ 可微， $h_i(\cdot)$ 不可微）时坐标下降法可收敛至目标函数的全局最优解，而在其它情况下算法的收敛性现阶段仍在研究中；

3. 最后给出坐标下降法的基本形式：给定初解 $x^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})$ ，每一轮迭代时按下式更新解 $x^{(k)}$

$$x_i^{(k)} \in \arg \min_{x_i} f(x_1^{(k)}, \dots, x_{i-1}^{(k)}, x_i, x_{i+1}^{(k-1)}, \dots, x_n^{(k-1)}) \quad (\text{坐标下降算法})$$

因为坐标下降算法在每一轮迭代时不止用了一阶梯度信息，故对于某些问题（如线性回归）算法的收敛速度可能显著优于梯度下降法。上式计算每一个分量 $x_i^{(k)}$ 时也可将最优化更新 ($\arg \min$) 改为梯度下降更新或近端下降更新；

4. 受制于坐标下降算法沿坐标轴方向依次更新的基本原理，算法除收敛性无法得到保证外的另一个主要缺点是难以并行化设计，从而限制了算法于大规模问题中的应用；
5. 在坐标下降的基础上进一步扩展即得到块坐标下降算法 (block coordinate descent)。算法依次优化变量的若干子集，其中每个子集可能包含多个决策变量元素。与坐标下降算法相比，在分块合适前提下块坐标下降可能具有更高的优化速度与收敛性。观察 ADMM 算法（详见第 17.6.2 节）的迭代公式，若不考虑约束及其对应的拉格朗日算子 λ ，则写为

$$x_{k+1} = \arg \min_x L_\gamma(x, z_k, \lambda_k), \quad z_{k+1} = \arg \min_z L_\gamma(x_{k+1}, z, \lambda_k)$$

若决策变量 x, z 为标量，则显然 (x, z) 按坐标下降更新；若 x, z 为向量，则形如 (x, z) 的更新过程即为块坐标下降。

17.8 多目标优化

17.8.1 帕累托最优 (Pareto optimality) 基本概念

1. 多目标优化是一类重要且复杂的优化问题。与单目标优化相比，多目标优化问题的目标为一个至少包括两个元素的向量，每一个元素即为一个优化目标。多目标优化问题的数学模型往往可以写成如下形式

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_n(\mathbf{x})]^T \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0 \\ & h_k(\mathbf{x}) = 0 \\ & \mathbf{x} \in X \end{aligned}$$

上式中 $f_1(\mathbf{x}), \dots, f_n(\mathbf{x})$ 即是问题所包括的 n 个目标，且往往不存在一个最优解 \mathbf{x}^* ，使得问题的所有目标 $\forall f_i(\mathbf{x})$ 均取得最小，即单目标优化中最优解的定义对于多目标优化问题不适用；

2. 为此，不同于单目标优化以求解最优解为目标，**多目标优化问题的目标在于求解一组解，这一组解中的每一个解相对于解空间的所有其他解具有某一种独特的“优势”**。这种优势可以通过某些理论进行描述，其中最常用理论即为帕累托最优 (Pareto optimality)；
3. 帕累托最优是指资源分配的一种理想状态。从福利学的角度解释，在资源总量和待分配对象固定的约束下，若不存在一种新的分配方式，使得在不损害一部分人利益的前提下提升另一部分人所享受的资源，则相应的状态即为帕累托最优，对应的分配方案即为该多目标优化问题最优解集中的一一个解，也称为帕累托有效解。反之，则该分配方案还可进一步改进，称为帕累托改进；
4. 在给出帕累托最优的数学描述前，首先给出多目标问题中相等、严格小于、小于、小于且不相等（支配）的数学定义。定义 n 维向量 $\mathbf{y} = [y_1, \dots, y_n]^T$ 和 $\mathbf{z} = [z_1, \dots, z_n]^T$ ，若：
- 对于 $\forall i$ ，均有 $y_i = z_i$ ，则称 \mathbf{y}, \mathbf{z} 相等，记为 $\mathbf{y} = \mathbf{z}$ ；
 - 对于 $\forall i$ ，均有 $y_i < z_i$ ，则称 \mathbf{y} 严格小于 \mathbf{z} ，记为 $\mathbf{y} < \mathbf{z}$ ；
 - 对于 $\forall i$ ，均有 $y_i \leq z_i$ ，则称 \mathbf{y} 小于 \mathbf{z} ，记为 $\mathbf{y} \leq \mathbf{z}$ ；
 - 对于 $\forall i$ ，均有 $y_i \leq z_i$ 且 $\mathbf{y} \neq \mathbf{z}$ ，则称 \mathbf{y} 小于且不等于（支配） \mathbf{z} ，记为 $\mathbf{y} \leqslant \mathbf{z}$ 。

5. 基于上述数学语言，给出帕累托最优相关基本概念的数学定义：

Pareto 支配 (Pareto dominance) 对于解空间内的两个解 $\mathbf{x}_1, \mathbf{x}_2 \in X$, 若有 $\mathbf{f}(\mathbf{x}_1) \leq \mathbf{f}(\mathbf{x}_2)$, 则称 \mathbf{x}_1 支配 \mathbf{x}_2 (\mathbf{x}_1 dominates \mathbf{x}_2 , or \mathbf{x}_2 is dominated by \mathbf{x}_1);

Pareto 有效解 对于解空间内的解 $\mathbf{x}^* \in X$, 若不存在另外一个解 $\mathbf{x} \in X$ 使得 $\mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}^*)$, 则称 \mathbf{x}^* 为 Pareto 有效解;

Pareto 弱有效解 对于解空间内的解 $\mathbf{x}^* \in X$, 若不存在另外一个解 $\mathbf{x} \in X$ 使得 $\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}^*)$, 则称 \mathbf{x}^* 为 Pareto 弱有效解;

Pareto 最优解集 (Pareto-optimal set) 给定多目标优化问题的所有 Pareto 有效解 $\mathbf{x}^* \in X$ 构成的解集称为 Pareto 最优解集, 最优解集中的解互相不支配;

Pareto 最优前沿 (Pareto-optimal front) 给定多目标优化问题的所有 Pareto 有效解 $\mathbf{x}^* \in X$ 对应的目标函数 $\mathbf{f}(\mathbf{x}^*)$ 构成的集合称为 Pareto 最优前沿。对于一个包含 n 个互斥目标的多目标优化问题, 其 Pareto 最优前沿为一个 $n - 1$ 个自由度的超曲面。当 $n = 2$ 时, Pareto 最优前沿为一段二维空间内的曲线, 曲线的两个端点分别对应单独考虑某一个目标时的最优解。多目标优化的目标即是求解问题的 Pareto 最优前沿。

第 18 章

图论与复杂网络

18.1 图的基本概念

1. 设集合 V 为有穷集, 集合 $\mathcal{R}(V) = \{\{u, v\} | u, v \in V\}$ 为集合 V 的二元子集, 则对于集合 $E \subseteq \mathcal{R}(V)$, 称二元组 $G = (V, E)$ 为无向图, 其中 V 为顶点集、 E 为边集, 任意元素 $\{u, v\} \in E$ 表示图 G 的边 uv , 称 u 与 v 邻接, uv 与 u, v 关联。记 $|V| = p$, $|E| = q$, 则称 G 为 (p, q) 图¹。称 $(1, 0)$ 图为平凡图, $(p, 0)$ 图为零图;
2. 同理设集合 V 为有穷集, 集合 $A \subseteq V \times V - \{\{v, v\} | v, v \in V\}$ ², 则称 $D = (V, A)$ 为有向图;
3. 在图模型中, 两顶点之间至多只能有一条边直接相连, 若模型中存在两顶点有两条直接相连的边, 则称其为多重图, 若模型中存在一条边, 边的两端点重合, 则称其为带环图, 多重图与带环图合称伪图;
4. 设 $G = (V, E)$, 记集合 $V_1 \subseteq V$, $E_1 \subseteq E \cap \mathcal{R}(V_1)$, 则称 $G_1 = (V_1, E_1)$ 为 G 的子图:
 - $G' = G - v$ 表示由图 G 去掉一顶点得到的子图;
 - $G' = G - uv$ 表示由图 G 去掉一边得到的子图;
 - $G' = (V, E')$ 表示与 G 具有相同顶点的子图, 称为 G 的生成子图。
5. 设 (p, q) 图 $G = (V, E)$, $\forall v \in V$, 则称与 v 关联的边数为 v 的度, 即为 $\deg(v)$, 满足

$$\sum_{v \in V} \deg(v) = 2q$$

记 $\delta(G) = \min\{\deg(v)\}$ 为图 G 的最小度数, 对应的有最大度数 $\Delta(G) = \max\{\deg(v)\}$, 当 $\delta(G) = \Delta(G) = r$ 时称 G 为 r - 正则图, 当 $r = p - 1$ 时称图 G 为完全图, 记为 K_p ;

6. 对两个图 $G = (U, E)$, $H = (V, F)$, $G \neq H$, $|E| = |F|$, 若存在一映射关系 $\varphi : U \rightarrow V$, φ 为一一对应, 则当满足下式时称 G, H 同构, 记为 $G \cong H$ 。

$$u_1u_2 \in E \iff \varphi(u_1)\varphi(u_2) \in F$$

直观理解, 两图同构意味着两图具有完全相同的拓扑结构, 仅节点命名不同。(而两图相同要求两图不仅拓扑结构完全相同, 节点名称也相同。)

7. 对图 $G = (V, E)$, 称一组由其顶点与边组成的交错序列 $v_1e_1v_2e_2 \dots v_{n-1}e_{n-1}v_n$ 为一条通道, 记为 v_1v_n 通道, 通道的长度为其中边的个数, 当 $v_0 = v_n$ 时称通道为闭通道:
 - 若通道中无重复的边, 则称为一条迹; 若闭通道中无重复的边, 则称为一条闭迹;
 - 若通道中无重复的顶点, 则称为一条路; 若闭通道中无重复的顶点, 则称为一条闭路, 又称为圈。若满足 $\delta(G) \geq m$, $m \geq 2$, 则图中有长至少为 $m + 1$ 的圈;

¹ p, q 分别为集合 V, E 的基数 (cardinal number), 集合的基数表示集合的大小, 空集的基数为 0, 单元素集的基数为 1, 则 p, q 分别指图 G 的顶点数和边数。

² 对集合 A, B , 称二元集合 $C = A \times B = \{(a, b) | a \in A, b \in B\}$ 为 A, B 的笛卡尔积, C 中元素为有序二元序列; 又称集合 $C = A - B$ 为 A, B 的差集, 也可写作 $C = A \setminus B$

- 若图 G 中任意两顶点 v_1, v_2 之间均存在至少一条路，则称图 G 为连通图。图 G 连通的充分条件为 $\delta(G) \geq [0.5p]$ 。当图不连通时，则称呼图中一极大连通子图为一个支。对于连通图 G ，若去掉任意一条边即会使得 G 不连通，则称为极小连通图。

8. 对图 $G = (V, E)$ ，进一步定义以下几何概念：

- 两顶点的距离：定义任意两顶点 u, v 间最短路径的长度为 u, v 的距离，记为 $d(u, v)$ ；
- 顶点的偏心：记任意顶点 v 的偏心为 $e(v) = \max_u\{d(v, u)\}$ ；
- 图的半径：记图 G 的半径为 $r(G) = \min_v\{e(v)\}$ ；
- 图的中心：图 G 的中心为一系列特定顶点组成的集合，记为 $H(G) = \{v|v \in V, e(v) = r(G)\}$ 。

9. 对图 $G = (V, E)$ ，称另一图 $G^c = (V, E^c)$ ， $E^c = \mathcal{R}(V) - E$ 为 G 的补图；

10. 对图 $G = (V, E)$ ，假设存在一个 V 的二划分 $\{V_1, V_2\}$ ³，使得任意 $uv \in E$ 都满足 $u \in V_1, v \in V_2$ 或 $u \in V_2, v \in V_1$ ，则称 G 为双图。图 G 为双图的充要条件为 G 中圈的长度为偶数（可为 0）。记 $|V_1| = m, |V_2| = n$ ，若双图 G 中的边数 $|E| = mn$ ，则称 G 为完全双图，记为 $K_{m,n}$ ；

11. 称图 $G = (V, E, f, g)$ 为带权图，其中 f, g 分别为图的顶点和边的权重，顶点 $v_i \in V$ 的权重为 $f(v_i)$ ，边 e_i 的权重为 $g(e_i)$ ；

12. 除了直观的几何结构外，图模型还存在以下几种表示方法：

- 邻接矩阵 (Adjacency Matrix): 邻接矩阵 A 用于表示图模型中节点间的连接关系，若图中含 p 个节点，则 A 为 $p \times p$ ，对任意 $v_i, v_j \in V$ 且 $v_i v_j \in E$ ，有 $A_{ij} = A_{ji} = 1$ ，反之 $A_{ij} = A_{ji} = 0$ 。邻接矩阵存在以下特点：

- $A_{ij} = A_{ji}, A_{ii} = 0, 0 \leq i, j \leq p - 1$ ；
- $\sum_j A_{ij} = \deg(v_i)$ ；
- $\sum_{i,j} A_{ij} = 2|E|$ ；
- $(A^l)_{ij}$ 表示 v_i 与 v_j 之间长为 l 的通道数；

由数学归纳法证明以上结论：即假设 v_i 与 v_j 之间长为 l 的通道数为 $(A^l)_{ij}$ 成立，只需证 v_i 与 v_j 之间长为 $l + 1$ 的通道数为 $(A^{l+1})_{ij}$ 。由矩阵乘法法则定义式

$$(A^{l+1})_{ij} = \sum_{h=0}^{p-1} (A^l)_{ih} A_{hj}$$

结合图模型，求 v_i 与 v_j 之间长为 $l + 1$ 的通道，即等价于求 v_i 与所有同 v_j 邻接的节点之间的长为 l 的通道，而式中 $(A^l)_{ih}$ 表示 v_i 与 v_h 之间长度为 l 的通道数， A_{hj} 表示 v_h 与 v_j 之间长度为 1 的通道数（即有无通道），显然满足图模型，则命题得证。

- 邻接表 (Adjacency List): 邻接矩阵以二维数组形式表示图模型，考虑了任意两节点间的邻接关系，当图中节点过多而边数稀疏时将造成巨大的空间消耗。邻接表是一种实现稀疏图的更空间高效的方法，以字典形式表示，字典的键为图的节点，字典的值为与该节点邻接的其它节点。对于带权图，其邻接表由二维字典表示；
- 关联矩阵 (Incidence Matrix): 关联矩阵 B 描述顶点与边的关系，对于 (p, q) 图，则 B 为 $p \times q$ ， $v_i, v_j \in V$ 且 $v_i v_j = e_h \in E$ ，有 $B_{ih} = B_{jh} = 1, 0 \leq i, j \leq p - 1, 0 \leq h \leq q - 1$ ，反之 $B_{ih} = B_{jh} = 0$ 。关联矩阵每一列只有两个 1。



欧拉图、欧拉定理及其推广

对无向图 $G = (V, E)$ ，包含 G 的所有顶点和所有边的（闭）迹称为欧拉（闭）迹，包含欧拉闭迹的图则称为欧拉图（可一笔画成、边不重复且回到原点）。

- 定理（欧拉定理）： G 为欧拉图的充要条件是 G 为连通图，且每个顶点的度为偶数；
- 定理：图 G 中有一条欧拉开迹（可一笔画成但回不到原点）的充要条件是 G 中恰有两个奇度顶点；
- 定理：图 G 中有 $2n$ 个奇度顶点，则 G 中至少有 n 条迹（最少需要 n 笔才能画完）；

³划分为集合论中的概念，对于集合 S ，若存在一二划分 $\{S_1, S_2\}$ ，则满足 $S_1 \cup S_2 = S, S_1 \cap S_2 = \emptyset$

以上定理同样适用于伪图。

哈密顿图

对无向图 $G = (V, E)$, 如果 G 中存在一包含的所有顶点的圈(生成圈), 则称其为哈密顿图(可一笔且不重复地遍历各个顶点并回到原点)。目前无充要条件判断任意图是否为哈密顿图, 仅存在一系列必要条件与充分条件:

1. 必要条件: 对任意点集 $S \subseteq V$, G 为哈密顿图的必要条件为 $w(G - S) \leq |S|$ 。式中 $w(G)$ 表示图 G 中支的个数;
2. 充分条件(Dirac 定理): 记 $|V| = p > 3$, 若对任意顶点 $v \in V$ 有 $\deg(v) \geq \frac{p}{2}$, 则 G 为哈密顿图;
3. 充分条件(Ore 定理): 对任意顶点对 $u, v \in V$ 且 $uv \notin E$, 若有 $\deg(u) + \deg(v) \geq p$, 则 G 为哈密顿图(若 $\deg(u) + \deg(v) \geq p - 1$, 则 G 中有哈密顿路);
4. 充分条件: 对任意顶点对 $u, v \in V$ 且 $uv \notin E$, 若有 $\deg(u) + \deg(v) \geq p$, 则 G 为哈密顿图;

18.1.1 树

1. 一个连通的无圈的图即称为树, 树属于双图, 而一个无圈的图则称为森林。只有一个顶点的树称为平凡树。树中度为 1 的节点称为叶, 非平凡树至少有两片叶子。记 G 为 (p, q) 图, 则以下条件完全等价, 互为充要(由任一条件即可推出其它全部条件):

- G 是树(G 连通且无圈);
- G 中任两个顶点间只有唯一一条路;
- G 连通, 且 $p = q + 1$;
- G 无圈, 且 $p = q + 1$;
- G 无圈, 且 G 中任两个不邻接的顶点加一条边即可得到一仅含唯一圈的图;
- G 为极小连通图。

欲严格证明上述条件完全等价, 需循环论证——由条件 1 推条件 2、由条件 2 推条件 3、……直至由最后一个条件推回条件 1, 以下仅对部分命题展开讨论

树的部分性质证明

命题 若 G 连通且无圈, 则必有 $p = q + 1$ 。

因为 G 连通, 则必有 $\delta(G) \geq 1$, 又因为 G 无圈, 则 $\delta(G) = 1$, 即 G 中必然存在叶子, 记叶子集为 V_1 , 则剩余节点组成集合 $V' = V - V_1$ 。去掉 G 中所有叶子及对应的边, 得新图 $G' = G - V_1$ 为 (p', q') 图, 记 $|V_1| = p_1$, 则显然有:

$$p' = p - p_1, \quad q' = q - p_1$$

又因为 G' 同样连通无圈, 则产生新的叶子集 V_2 , 剩余节点组成集合 $V'' = V' - V_2$, 同理继续去掉 G' 中所有叶子及对应的边, 得 (p'', q'') 图 $G'' = G' - V_2$, 同样有 $p'' = p - p_1 - p_2, \quad q'' = q - p_1 - p_2$ 。

循环上述过程得 $(p^{(k)}, q^{(k)})$ 图 $G^{(k)} = G^{(k-1)} - V_k$, 至 $0 < p^{(k)} \leq 2$ 时终止上述过程, 此时同样有:

$$p^{(k)} = p - \sum p_i, \quad q^{(k)} = q - \sum p_i$$

且显然有:

$$p^{(k)} = q^{(k)} + 1$$

综上, 即有 $p = q + 1$, 命题得证。

命题 若 G 连通且 $p = q + 1$, 则必无圈。

反证命题“存在连通且有圈图 G , 有 $p = q + 1$ ”不成立。

考虑一由 n 个顶点 n 条边组成的圈 C_n , 其本身即为一连通图, 且显然不满足 $p = q + 1$ 。在 C_n 的基础上依次增加一个顶点得到新的图, 为保证新的图仍为连通图, 每增加一个顶点需至少增加一条边, 显然不可能满足 $p = q + 1$, 命题得证。

命题 若 G 无圈且 $p = q + 1$, 则必连通。

反证命题“存在无圈且不连通图 G , 有 $p = q + 1$ ”不成立。

假设无圈且不连通图 G 可分为 k 个支, 则每个支都连通且无圈, 则对于任意支 G_i 有 $p_i = q_i + 1$, 此时

$$\sum p_i = \sum (q_i + 1) \iff p = q + k, \quad k > 1$$

显然与假设矛盾，则命题得证。

2. 树作为一种特殊的图同样有中心的概念，而且树只有一个或两个中心，只需要不断去掉树的叶子即可最终得到其中心；
3. 设图 G ，若 G 的一个生成子图为树，则称其为 G 的生成树。 G 有生成树的充要条件是 G 连通。对于带权图 $G = (V, E, w)$ (w 为边的权重)，则使得树的各边权值之和最小的生成树称为最小生成树，**Prim 算法**与**Kruskal 算法**是两类求最小生成树的算法，两者的复杂度分别为 $O(p^2)$, $O(q \log q)$ 。

18.1.2 割点与桥

1. 设图 $G = (V, E)$, $v \in V$, 若 $w(G - v) > w(G)$ 则称 v 为 G 的割点 ($w(G)$ 表示 G 的支数)，每个非平凡图至少有两个顶点不是割点，哈密顿图无割点。关于割点，以下命题完全等价、互为充要：
 - v 为割点；
 - 存在 $u, w \in V$, $u \neq w$, 使得 u, w 间所有的路均通过 v ；
 - 存在 $V - v$ 的二划分 $\{U, W\}$, 使得 $\forall u \in U, \forall w \in W$, 使得 u, w 间所有的路均通过 v 。
2. 设图 $G = (V, E)$, $e \in E$, 若 $w(G - e) > w(G)$ 则称 e 为 G 的桥 ($w(G)$ 表示 G 的支数)。关于桥，以下命题完全等价、互为充要：
 - e 为桥；
 - 存在 $u, w \in V$, $u \neq w$, 使得 u, w 间所有的路均通过 e ；
 - 存在 $V - e$ 的二划分 $\{U, W\}$, 使得 $\forall u \in U, \forall w \in W$, 使得 u, w 间所有的路均通过 e ；
 - e 不在任何圈上。

18.2 图的连通度

1. 设 $G = (V, E)$, 定义使得 G 不连通或变为平凡图所需要的最少顶点（边）数称为图的顶点（边）连通度，记为 $\kappa(G)$ ($\lambda(G)$)：
 - 不连通图或平凡图的顶点连通度和边连通度均为 0；
 - 树的顶点连通度和边连通度均为 1；
 - 若 G 带割点，则 $\kappa(G) = 1$ ；
 - 若 G 带桥，则 $\lambda(G) = 1$ ；
 - K_p 图的顶点连通度和边连通度均为 $p - 1$ ；
 - 图 G 连通当且仅当 $\kappa(G) \geq 1$ 。
2. 对于任意图 G , $\kappa(G) \leq \lambda(G) \leq \delta(G)$, 且对任意满足 $a \leq b \leq c$ 的正整数 a, b, c , 存在图 G 使得 $\kappa(G) = a$, $\lambda(G) = b$, $\delta(G) = c$ 。若满足 $\delta(G) \geq [0.5p]$, 则有 $\lambda(G) = \kappa(G)$ ；
3. 设图 $G = (V, E)$, 若 $\kappa(G) \geq n \geq 0$, 则称 G 为 n - 顶点连通图（简称 n - 连通图）；若 $\lambda(G) \geq n \geq 0$, 则称 G 为 n - 边连通图；

18.3 最短路问题

1. 最短路问题是图论中应用最广泛的问题之一。许多优化问题可以使用这个模型，如设备更新、管道铺设、线路安排、厂区布局等；
2. 设 $G = (V, E)$ 为连通图，图中各边 (v_i, v_j) 有权 w_{ij} ($w_{ij} = \infty$ 表示 v_i, v_j 间无连边)。求一条道路 μ , 使它是从 v_s 到 v_t 的所有路中总权最小的路，即

$$L(\mu) = \min \sum_{(v_i, v_j) \in \mu} w_{ij}$$

3. 最短路问题完全可以用线性规划问题描述，因而它的求解也可以用线性规划的方法，但图论的方法更为简单。

18.3.1 狄克斯托 (Dijkstra) 算法

1. 该算法由 Dijkstra 于 1959 年提出，可用于求解指定两点 v_i, v_j 间的最短路，或从指定点 v_s 到其余各点的最短路，目前被认为是求解无负权图（可以是有向图也可以是无向图）最短路的最好方法；
2. 算法的核心思想是：若序列 $\{v_s, v_1, \dots, v_{n-1}, v_n\}$ 是从 v_s 到 v_n 的最短路，则序列 $\{v_s, v_1, \dots, v_{n-1}\}$ 一定是从 v_s 到 v_{n-1} 的最短路。算法从 v_s 出发，逐步向外探寻 v_s 至各点的最短路，并对探索的路径进行标记，标记内容包括当前路径的总权值和当前路径终点的前一个点。算法定义两种标号：**T** 标号（试探性标号，tentative label）与 **P** 标号（永久性标号，permanent label）；
3. 算法每一步总是从上一个被打上 P 标号的点出发，给多个点打上 T 标号，表示从 v_s 至 v_i 的最短路权的上界，同时会在所有被打上 T 标号的点中确定一个点将其 T 标号转为 P 标号，表示从 v_s 至 v_i 的最短路权。得到 P 标号后 v_i 的标号将不再改变。当终点 v_n （或所有点）得到 P 标号时，算法结束。对于有 n 个点的图，最多经 $n - 1$ 步就可以得到从起始点到终点的最短路；
4. 定义集合 S_i 表示当前得到 P 标号的点集， $P(v_i)$ 表示从 v_s 到 v_i 的最短路权（P 标号）， $T(v_i)$ 表示探索的从 v_s 到 v_i 的最短路权上界（T 标号）， $\lambda(v_i)$ 表示当前从 v_s 到 v_i 的路径中 v_i 的前一个点。算法的具体步骤为：
 - 开始 ($i = 0$) 令 $S_0 = \{v_s\}$, $P(v_s) = 0$, $\lambda(v_s) = 0$, 对于每一个 $v \neq v_s$, 令 $T(v) = +\infty$, $\lambda(v) = M$;
 - 若 $S_i = V$ 或 $v_n \in S_i$, 则算法终止, 否则转入下一步;
 - 进行 **T** 标号：记上一个得到 P 标号的点为 v_k , 考察所有与 v_k 相连且无 P 标号的点 v_j (即 $(v_k, v_j) \in E$ 且 $v_j \notin S_i$), 若 $T(v_j) > P(v_k) + w_{ij}$, 则令 $T(v_j) = P(v_k) + w_{ij}$, $\lambda(v_j) = v_k$, 否则转入下一步;
 - 进行 **P** 标号：令 $v'_k = \arg \min_{v_j \notin S_i} \{T(v_j)\}$, 若 $T(v'_k) < +\infty$, 则令 $P(v'_k) = T(v'_k)$, $S_{i+1} = S_i \cup \{v'_k\}$ 。

18.4 最大流问题

18.4.1 基本概念

1. 对于一个有向图 $D = (V, A)$, 在 V 中指定一个点称为发点 (v_s), 而另一点称为收点 (v_t), 其余点称为中间点。对于每一个有向弧 $(v_i, v_j) \in A$, 对应有一个 $c(v_i, v_j) \geq 0$ 称为弧 (v_i, v_j) 的容量 (简写为 c_{ij})。通常我们就把这样的有向图 D 称为一个网络, 记做 $D = (V, A, C)$;
2. 定义在弧集 A 上的一个函数 $f(v_i, v_j)$ 为弧 (v_i, v_j) 的流量 (简写为 f_{ij})。若所有弧的流量 $f_{ij} = 0$, 则称 f 为零流。定义满足下述条件的流 f 为可行流, 并称 $v(f)$ 为可行流 f 的流量:
 - 容量限制条件：对于每一弧 $(v_i, v_j) \in A$, 有 $0 \leq f_{ij} \leq c_{ij}$;
 - 平衡条件：

$$\begin{cases} \sum_{j, (v_i, v_j) \in A} f_{ij} - \sum_{j, (v_j, v_i) \in A} f_{ji} = 0 & \forall i \neq s, t \\ \sum_{j, (v_s, v_j) \in A} f_{sj} - \sum_{j, (v_j, v_s) \in A} f_{js} = v(f) \\ \sum_{j, (v_t, v_j) \in A} f_{tj} - \sum_{j, (v_j, v_t) \in A} f_{jt} = -v(f) \end{cases}$$

对于一个可行流 f , 称满足 $f_{ij} = c_{ij}$ 的弧为饱和弧, 满足 $f_{ij} < c_{ij}$ 的弧为非饱和弧, 满足 $f_{ij} = 0$ 的弧为零流弧, $f_{ij} > 0$ 的弧为非零流弧;

3. 最大流问题就是求一个可行流 f , 使其流量 $v(f)$ 达到最大, 即

$$\max v(f)$$

$$\text{s.t.} \begin{cases} 0 \leq f_{ij} \leq c_{ij} & \forall i, j (v_i, v_j) \in A \\ \sum_{j, (v_i, v_j) \in A} f_{ij} - \sum_{j, (v_j, v_i) \in A} f_{ji} = 0 & \forall i \neq s, t \\ \sum_{j, (v_s, v_j) \in A} f_{sj} - \sum_{j, (v_j, v_s) \in A} f_{js} = v(f) \\ \sum_{j, (v_t, v_j) \in A} f_{tj} - \sum_{j, (v_j, v_t) \in A} f_{jt} = -v(f) \end{cases}$$

显然，最大流问题也是一个特殊的线性规划问题；

4. 对于连接发点 v_s 到收点 v_t 的链 μ ，定义其方向是从 v_s 到 v_t ，则链中的弧被分为两类：一类是弧的方向与链的方向一致，叫做前向弧，前向弧的全体记为 μ^+ ；另一类是弧的方向与链的方向相反，叫做后向弧，后向弧的全体记为 μ^- 。若 f 是一可行流， μ 是从 v_s 到 v_t 的一条链，若 μ 满足如下条件，则称其为（关于可行流 f 的）增广链：

- 在弧 $(v_i, v_j) \in \mu^+$ 上， $0 \leq f_{ij} < c_{ij}$ ，即 μ^+ 中的每一弧是非饱和弧；
- 在弧 $(v_i, v_j) \in \mu^-$ 上， $0 < f_{ij} \leq c_{ij}$ ，即 μ^- 中的每一弧是非零流弧。

若网络中存在关于 f 的增广链，则网络的流量 $v(f)$ 可进一步增大。换言之，可行流 f^* 是最大流，当且仅当网络中不存在关于 f^* 的增广链；

5. 给定网络 $D = (V, A, C)$ ，若点集 V 被划分为两个非空集合 V_1, \bar{V}_1 ，使 $v_s \in V_1, v_t \in \bar{V}_1$ ，则把弧集 (V_1, \bar{V}_1) 称为（分离 v_s, v_t ）的截集，把截集 (V_1, \bar{V}_1) 中所有弧的容量和称为这个截集的容量（简称截量），记为 $c(V_1, \bar{V}_1)$ 。显然，任何一个可行流的流量 $v(f)$ 都不会超过任意截集的容量，而从 v_s 到 v_t 的最大流的流量等于分离 v_s, v_t 的最小截集的容量（最大流量最小截量定理），即

$$v(f) \leq c(V_1, \bar{V}_1) \quad v(f^*) = c(V_1^*, \bar{V}_1^*)$$

18.4.2 Ford-Fulkerson 算法

Ford-Fulkerson 算法是经典的计算网络最大流问题的算法。算法将计算网络最大流问题转化为寻找网络增广链的问题。包括标号过程和调整过程共两个过程。在标号过程中，算法找到当前网络在当前流量下的一条增广链，并在调整过程中调整增广链各弧的流量使其不再为增广链。直至网络中找不到增广链即可得到最大流。

1. **标号过程：**一般地一步对多个未标号的点 v_i 进行标号，标号内容包括两部分——从 v_s 到 v_i 的链的倒数第二个点（即 v_i 在链中的前一个点）和调整量 θ （用于调整找到的增广链流量）。具体步骤为：
 - 开始时对 v_s 标号 $(0, +\infty)$ ；
 - 任取一个被标号的点为 v_i ，对于一切未被标号的点 v_j ：
 - 若 $(v_i, v_j) \in \mu^+, f_{ij} < c_{ij}$ ，则对 v_j 标号 $(v_i, l(v_j))$ ，其中 $l(v_j) = \min\{l(v_i), c_{ij} - f_{ij}\}$ ；
 - 若 $(v_j, v_i) \in \mu^-, f_{ji} > 0$ ，则对 v_j 标号 $(-v_i, l(v_j))$ ，其中 $l(v_j) = \min\{l(v_i), f_{ji}\}$ ；
 - 若以上条件均不满足，则 v_j 不满足标号条件。
 - 若所有的标号均检查过而标号过程无法继续进行，则说明此时的可行流就是最大流，算法结束。反之若 v_t 被标上号，则得到一条从 v_s 到 v_t 的增广链，进入调整过程。
2. **调整过程：**对于找到的增广链 μ ，基于调整量 θ 对网络流量进行调整

$$f'_{ij} = \begin{cases} f_{ij} + \theta & (v_i, v_j) \in \mu^+ \\ f_{ij} - \theta & (v_i, v_j) \in \mu^- \\ f_{ij} & (v_i, v_j) \notin \mu \end{cases} \quad \theta = l(v_t)$$

得到新的可行流 f' 后去掉所有标号，重新回到标号过程，寻找下一条增广链。

18.5 拉普拉斯矩阵与图傅里叶变换

18.5.1 从拉普拉斯算子到拉普拉斯矩阵 (graph Laplacian)

- 除了邻接矩阵外，拉普拉斯矩阵 L 是另一种重要的表示方法，是数学分析中拉普拉斯算子（Laplace Operator） Δ 在离散非欧几里得空间中的扩展，定义为

$$L = D - A \quad (\text{拉普拉斯矩阵})$$

式中 D, A 分别为复杂网络的度矩阵与邻接矩阵。进一步地简要回顾拉普拉斯算子 Δ 并推导拉普拉斯矩阵 L 的定义式，最后介绍拉普拉斯矩阵的物理意义；

- 拉普拉斯算子为 n 维欧几里得空间中的二阶微分算子，定义为函数梯度的散度，对于二维连续可微函数 $f(x_1, x_2)$ ，有

$$\Delta f = \nabla^2 f = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2}$$

对于欧几里得空间下的离散函数 f ，也有

$$\Delta f = f(x_1 + 1, x_2) + f(x_1, x_2 + 1) + f(x_1 - 1, x_2) + f(x_1, x_2 - 1) - 4f(x_1, x_2)$$

从物理的角度说， $\Delta f < 0$ 意味着 $f(x_1, x_2)$ 处的能量倾向于向外部辐射；

- 在图论或复杂网络中也可以引入节点的拉普拉斯算子。假设图 G 包含 N 个节点，则对应函数 f 为 N 维向量函数，其中元素 f_i 表示图 G 中节点 i 的权重，有

$$\Delta f_i = \sum_j a_{ij}(f_i - f_j)$$

式中 a_{ij} 指边 e_{ij} 的权重，也是图 G 邻接矩阵 A 的对应元素，当节点 i, j 不邻接或 $i = j$ 时有 $a_{ij} = 0$ 。可以看到图空间下拉普拉斯算子的数学及物理定义与欧几里得空间下的拉普拉斯算子恰好相反——无论是以活跃度、吸引力、污染指数等定义网络顶点的权重， $\Delta f_i > 0$ 都意味着信息、吸引、污染等更容易由节点 i 向相邻接点传播。进一步地变换上述定义式

$$\Delta f_i = \sum_j a_{ij}(f_i - f_j) = f_i \sum_j a_{ij} - \sum_j a_{ij} f_j = d_i f_i - \vec{a}_i \cdot \vec{f}$$

式中 d_i 表示节点 i 的度， \vec{a}_i 表示邻接矩阵 A 中的第 i 组行向量， \vec{f} 则为包含图中各节点权重的列向量。将对节点的拉普拉斯算子推广至全图，即可得到

$$\Delta \vec{f} = (D - A) \vec{f} = L \vec{f}$$

故称 $L = D - A$ 为图的拉普拉斯矩阵。可以看到，将拉普拉斯矩阵 L 视为算子，可由 $L \vec{f}$ 提取图中各节点邻域的局部结构特征；

- 除此之外，拉普拉斯矩阵也可用于提取图结构的全局特征，注意到

$$\begin{aligned} \vec{f}^\top L \vec{f} &= \vec{f}^\top D \vec{f} - \vec{f}^\top A \vec{f} = \sum_i d_i f_i^2 - \sum_i \sum_j a_{ij} f_i f_j \\ &= \frac{1}{2} \left(\sum_i d_i f_i^2 - 2 \sum_i \sum_j a_{ij} f_i f_j + \sum_j d_j f_j^2 \right) \\ &= \frac{1}{2} \left(\sum_i \sum_j a_{ij} f_i^2 - 2 \sum_i \sum_j a_{ij} f_i f_j + \sum_i \sum_j a_{ij} f_j^2 \right) = \frac{1}{2} \sum_i \sum_j a_{ij} (f_i - f_j)^2 \end{aligned}$$

上式被称为图拉普拉斯二次型 (**graph Laplacian quadratic form**), 其结果也被称为图的总变差 (**total variation**), 反映了向量 \vec{f} 于图结构 G 上的平滑程度, 数值 $\vec{f}^\top L \vec{f}$ 越小, 表示图中各节点与相邻节点的属性差异越小。上式也可进一步扩展为图节点属性 f_i 为向量的场景, 则

$$\begin{aligned}\frac{1}{2} \sum_i \sum_j a_{ij} \|\vec{f}_i - \vec{f}_j\|^2 &= \frac{1}{2} \sum_i \sum_j a_{ij} \left(\vec{f}_i^\top \vec{f}_i - 2\vec{f}_i^\top \vec{f}_j + \vec{f}_j^\top \vec{f}_j \right) \\ &= \sum_i d_i \vec{f}_i^\top \vec{f}_i - \sum_i \sum_j a_{ij} \vec{f}_i^\top \vec{f}_j = \text{tr}(F^\top DF) - \text{tr}(F^\top AF) = \text{tr}(F^\top LF)\end{aligned}$$

式中 $\text{tr}(\cdot)$ 表示矩阵的迹 (即对角线元素之和), $F = [\vec{f}_1, \vec{f}_2, \dots, \vec{f}_N]^\top$ 为以 \vec{f}_i 为行向量的矩阵;

5. 需要强调的是虽然拉普拉斯矩阵按定义适用于任意图结构, 但其很多性质及进一步应用仅考虑边权非负的无向图。这是因为拉普拉斯矩阵的很多特殊性质仅在其对称 (**symmetric**) 和半正定 (**positive semidefinite**) 时成立, 而相关性质又是谱聚类等大量基于拉普拉斯矩阵的算法的基础。拉普拉斯矩阵的对称性要求图为无向图。对于有向图, 如需应用相关要求对称拉普拉斯矩阵的算法时, 可构造对称邻接矩阵 $\hat{A} = A + A^\top$ 从而计算对称的拉普拉斯矩阵。拉普拉斯矩阵的半正定性要求图为非负权图。对于负权图, 已有研究表明只有连边的负权绝对值小于一定程度时拉普拉斯矩阵的半正定性才成立;
6. 拉普拉斯矩阵能成为一系列图相关理论和算法 (如谱聚类、图深度学习) 的基础, 在很大程度上得益于其 (无向非负权图) 特征值和特征向量的特殊性质:
 - 因为拉普拉斯矩阵为对称阵, 故其必然存在 n 个相互正交的特征向量 $U = \{v_i\}_{i=1}^n$;
 - 因为拉普拉斯矩阵为实对称阵, 故其特征值均为实数, 特征向量也均为实向量;
 - 因为拉普拉斯矩阵为半正定阵 (由拉普拉斯二次型可知), 故其特征值均为非负;
 - **拉普拉斯矩阵的最小特征值为 0**, 对应特征向量 1_n (表示全为 1 的 n 维列向量), 因为 $L1_n = 0 = 0 \times 1_n$ 。
7. 记拉普拉斯矩阵 L 的 n 个特征值自小到大排列分别为 $\lambda_1, \dots, \lambda_n$, 进一步分析其每一个特征值的物理意义。因为 L 为实对称阵, 故其可作特征值分解得

$$L = U \Lambda U^\top = \sum_{i=1}^n \lambda_i U U^\top$$

不妨令 $L \simeq \hat{L}_i = \lambda_i U U^\top$ 。因为拉普拉斯矩阵可作为图结构的代数表示方法, 故 \hat{L}_i 也可表示为某种图结构, 则上式可理解为 L 所表示的图结构可视为若干个 \hat{L}_i 所表示的图的组合, 而每一个 \hat{L}_i 所表示的图各自表征了原图的一部分信息, 且表征的信息由相应的特征值 λ_i 决定。进一步分析 λ_i 的取值对 \hat{L}_i 所表征的信息的影响。考虑图的总变差有

$$f^\top L f = \sum_{i=1}^n \lambda_i f^\top U U^\top f = \sum_{i=1}^n f^\top \hat{L}_i f$$

上式指出每一个 \hat{L}_i 所表示的图各自表征了原图的一部分总变差, 且 λ_i 越大表征的总变差也越大。又因为总变差刻画了图数据的不均匀性, 故可得出**拉普拉斯矩阵的特征值捕捉了图结构数据的分布特征, 其中较大的特征值捕捉了图数据分布的不均匀结构, 而较小的特征值反映了图数据分布的均匀结构**。

18.5.2 标准化拉普拉斯矩阵 (**Laplacian matrix normalization**)

1. 尽管拉普拉斯矩阵可表征图的大量信息, 但其在某些场合仍具有一定问题。上小结类比拉普拉斯算子推导拉普拉斯矩阵时指出拉普拉斯矩阵 L 可捕捉图中各节点邻域的局部结构特征

$$[Lf]_i = \sum_j a_{ij} (f_i - f_j)$$

可以看到若以 L 作为算子, 则其挖掘图局部结构特征时不仅考虑了 f 于节点 i 及其相邻节点 $\forall j$ 的分布特征, 也对相应连边的权重 a_{ij} 敏感, 且因为上式为加权求和的形式, 故其计算结果还会对节点 i 的度敏感。总之, 拉普拉斯矩阵在捕捉图局部特征时保留了过多信息, 反而降低了指标统计意义的清晰度——即使 $[Lf]_i \simeq [Lf]_j$ 也无法确定 i, j 节点具有相似的局部结构特征, 因为可能的干扰因素过多;

2. 为减少指标的干扰信息，一个自然的思路即是对其标准化，具体到本例，不妨将加权和改为加权平均

$$[L^{rw} f]_i = \frac{\sum_j a_{ij}(f_i - f_j)}{\sum_j a_{ij}} \implies L^{rw} = D^{-1}L = I - D^{-1}A \quad (\text{随机游走标准化拉普拉斯矩阵})$$

上式中的 L^{rw} 即为一类典型的标准化拉普拉斯矩阵，其主对角线元素均为 1，而非对角线元素的绝对值均不超过 1，每行元素之和为 0。经标准化后 $[L^{rw} f]_i$ 的统计意义也更为清晰——其量化了节点 i 的属性 f_i 与其相邻节点属性 f_j , $\forall j$ 的相似性。进一步关注上式中的矩阵 $D^{-1}A$ 。注意到 $[D^{-1}A]_{ij} = \tilde{a}_{ij} = \frac{a_{ij}}{\sum_j a_{ij}}$ ，故 $D^{-1}A$ 实际上是将图中所有节点的度均标准化为 1，则标准化后的边权 \tilde{a}_{ij} 可视为从节点 i 转移至节点 j 的概率，此时矩阵 $D^{-1}A$ 即可理解为随机游走 (**random walk**) 的概率转移矩阵，故而 L^{rw} 得名为随机游走标准化拉普拉斯矩阵 (**random-walk normalized Laplacian**)。尽管标准化后 L^{rw} 具有诸多优秀性质，但其并非对称阵，从而失去了诸多计算和分析上的优势；

3. 为构造对称的标准化拉普拉斯矩阵，同样从 $[Lf]_i$ 的定义入手，只需构造 L^{sym} 满足

$$[L^{sym} f]_i = \frac{\sum_j a_{ij}(f_i - f_j)}{\sqrt{(\sum_i a_{ij})(\sum_j a_{ij})}} \implies L^{sym} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \quad (\text{对称标准化拉普拉斯矩阵})$$

上式中 L^{sym} 被称为对称标准化拉普拉斯矩阵 (**symmetrically normalized Laplacian**)，是被应用最广的拉普拉斯矩阵标准化形式。与 L^{rw} 相似， L^{sym} 的主对角线元素均为 1，非对角线元素的绝对值均不超过 1，但每行元素之和不为 0。 $[L^{sym} f]_i$ 的统计意义也与 $[L^{rw} f]_i$ 相似。另外与 L 一致， L^{sym} 也为半正定阵

$$f^\top L^{sym} f = f^\top D^{-\frac{1}{2}}LD^{-\frac{1}{2}}f = g^\top Lg \geq 0, \quad g = D^{-\frac{1}{2}}f$$

4. 因为同样满足半正定性和对称性， L^{sym} 的特征值和特征向量也继承了 L 的诸多特殊性质：

- 因 L^{sym} 为对称阵，故其必然存在 n 个相互正交的特征向量 $U = \{v_i\}_{i=1}^n$ ；
- 因 L^{sym} 为实对称阵，故其特征值均为实数，特征向量也均为实向量；
- 因 L^{sym} 为半正定阵，故其特征值均为非负；
- L^{sym} 的最小特征值为 0，对应的特征向量为 $D^{\frac{1}{2}}1_n$ ；
- L^{sym} 的最大特征值不超过 2。该性质是对称标准化拉普拉斯矩阵最重要的性质，也是其被广泛应用的原因——特征值上限固定可确保一系列计算结果更为稳定。进一步证明该性质。注意到

$$R(L^{sym}, f) = \frac{f^\top L^{sym} f}{f^\top f} = \frac{f^\top (I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}) f}{f^\top f} = 1 - \frac{f^\top D^{-\frac{1}{2}}AD^{-\frac{1}{2}}f}{f^\top f} = 1 - R(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, f)$$

上式中 $R(L^{sym}, f)$ 表示瑞利商 (**Rayleigh quotient**)，由第 23.13 节可知，计算 L^{sym} 的最大特征值等价于计算瑞利商 $R(L^{sym}, f)$ 的极大值，又等价于计算 $R(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, f)$ 的极小值。首先证明矩阵 $I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ 为半正定阵，按定义有

$$\begin{aligned} f^\top (I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}) f &= f^\top f + f^\top D^{-\frac{1}{2}}AD^{-\frac{1}{2}}f = \sum_i f_i^2 + \sum_{ij} \frac{a_{ij}f_i f_j}{\sqrt{d_i d_j}} \\ &= \frac{1}{2} \left(\sum_i d_i g_i^2 + \sum_j d_j g_j^2 + 2 \sum_{ij} a_{ij} f_i f_j \right) \\ &= \frac{1}{2} \left(\sum_{ij} a_{ij} g_i^2 + \sum_{ij} a_{ij} g_j^2 + 2 \sum_{ij} a_{ij} f_i f_j \right) \\ &= \frac{1}{2} \sum_{ij} a_{ij} (g_i + g_j)^2 \geq 0, \quad g_i = \frac{f_i}{d_i} \end{aligned}$$

当且仅当 $\exists g \neq 0$ 使得对 $\forall a_{ij} > 0$ 均有 $g_i = -g_j$ 时上式等号成立。由此可知 $R(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, f)$ 的下限

$$R(I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, f) = 1 + R(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, f) \geq 0 \implies R(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, f) \geq -1 \implies R(L^{sym}, f) \leq 2$$

当上式等号成立时 L^{sym} 的最大特征值为 2，当等号不成立时则最大特征值小于 2。

18.5.3 图的微分运算

1. 本节开头基于欧式空间中的拉普拉斯算子直观类比得出非欧图结构中的拉普拉斯矩阵，但没给出推导的数学基础。拉普拉斯算子建立在传统微分运算的基础上，本小节同样补充针对非欧图结构的微分运算相关定义⁴；
2. 首先记函数 $f(v_i)$ 为定义在图节点 v_i 上的函数，又记函数 $F(v_i, v_j)$ 为定义在图连边 e_{ij} 上的函数。 f 对应欧式空间中的标量场函数，每处取值具有强度信息而无方向信息； F 对应欧式空间中的向量场函数，每处取值既有强度信息也有方向信息；
3. 首先定义非欧图空间下的一阶微分算子：

一阶偏导 传统欧式空间中偏导运算定义在标量场函数，故图的偏导运算同样针对 $f(v_i)$ 定义。在节点 v_i 处， f 对于某条连边的一阶偏导有

$$\frac{\partial f(v_i)}{\partial v_j} = \sqrt{a_{ij}}(f(v_j) - f(v_i))$$

梯度 传统欧式空间中梯度运算定义在标量场函数，且标量场函数的梯度为向量场函数。故图的梯度运算同样针对 $f(v_i)$ 定义，且梯度运算的结果为定义在图连边 e_{ij} 上的函数。在节点 v_i 处， f 的梯度为对各连边一阶偏导组成的向量

$$\nabla_i f = \left\{ \frac{\partial f(v_i)}{\partial v_j} \mid v_j \in N(v_i) \right\} \Rightarrow (\nabla f)(v_i, v_j) = \frac{\partial f(v_i)}{\partial v_j}$$

局部变异性 (local variation) 基于节点 v_i 的梯度信息可定义该点处的局部变异性为其梯度的 l_p 范数

$$\|\nabla_i f\|_p = \begin{cases} \left(\sum_j a_{ij}^{p/2} |f(v_j) - f(v_i)|^p \right)^{\frac{1}{p}}, & 1 \leq p < \infty \\ \max_j \sqrt{a_{ij}} |f(v_j) - f(v_i)|, & p = \infty \end{cases}$$

散度 传统欧式空间中散度运算定义在向量场函数，且向量场函数的散度为标量场函数。故图的散度运算同样针对 $F(v_i, v_j)$ 定义，且散度运算的结果为定义在图节点 v_i 上的函数。在节点 v_i 处， F 的散度

$$(\operatorname{div} F)(v_i) = \frac{1}{2} \sum_j \sqrt{a_{ij}}(F(v_i, v_j) - F(v_j, v_i))$$

若将 $F(v_i, v_j)$ 理解为连边 e_{ij} 上的流量，则图节点散度 $(\operatorname{div} F)(v_i)$ 的物理意义即为节点 v_i 处的“净流出”。 $(\operatorname{div} F)(v_i) > 0$ 表示该点有散发通量的正源（发散源）， $(\operatorname{div} F)(v_i) < 0$ 表示该点有吸收通量的负源（洞），与传统散度的定义及性质一致。

4. 在上述一阶微分算子的基础上，进一步定义图结构的主要二阶微分算子：

图拉普拉斯算子 传统欧式空间中的拉普拉斯算子定义为梯度的散度，而图结构的拉普拉斯算子则需要对该结果取相反数

$$\begin{aligned} (\Delta f)(v_i) &= -(\operatorname{div}(\nabla f))(v_i) = \frac{1}{2} \sum_j \sqrt{a_{ij}}((\nabla f)(v_j, v_i) - (\nabla f)(v_i, v_j)) \\ &= \frac{1}{2} \sum_j a_{ij}(f(v_i) - f(v_j) - f(v_j) + f(v_i)) = \sum_j a_{ij}(f(v_i) - f(v_j)) \end{aligned}$$

上式即为上小节给出的拉普拉斯矩阵的推导；

图 p - 拉普拉斯算子 p - 拉普拉斯算子是一般拉普拉斯算子的推广，对于图结构其定义为

$$(\Delta_p f)(v_i) = \sum_j a_{ij}^{p/2} |f(v_i) - f(v_j)|^{p-2} (f(v_i) - f(v_j)), \quad 1 \leq p < \infty$$

⁴Calculus on finite weighted graphs: https://en.wikipedia.org/wiki/Calculus_on_finite_weighted_graphs

18.5.4 图傅里叶变换 (graph Fourier transform, GFT)

- 对于一般的定义在欧式空间的函数 $f(x)$, 既可以直接在 x 的定义域内分析其特征, 也可以对其做傅里叶变换 (或拉普拉斯变换) 后从频域 (或复频域) 进行分析。大量关于信号处理的研究指出, 针对某些问题从频域角度分析函数具有显著优势, 例如简化计算、特征提取、信号降噪等等;
- 在此基础上不妨猜想, 对于定义在非欧图空间的函数 $f(v)$, 是否存在类似于傅里叶变换的方法, 允许从另一个角度分析 $f(v)$ 于图空间中的分布特征。相关方法称为图傅里叶变换 (graph Fourier transform, GFT), 是图信号处理 (graph signal process, GSP) 的基础, 可以将 $f(v)$ 变换至谱空间 (spectral space)。此时 $f(v)$ 被称为图信号, 为定义在图节点 v 上的函数;
- 图作为一种非欧离散结构, 故直观类比离散傅里叶变换 (详见第 21.12 节) 给出图傅里叶变换的定义。对于离散时间序列 $x(n)$, 其离散傅里叶变换 (DFT) 与逆离散傅里叶变换 (IDFT) 公式如下

$$\text{DFT} : X(k) = \sum_{n=0}^{N-1} x(n)e^{-i\frac{2\pi}{N}kn} \iff \text{IDFT} : x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{i\frac{2\pi}{N}kn}$$

由上式可知, 离散傅里叶变换实际上是将序列 $x(n)$ 分解至有限个正交基 $\{e^{i\frac{2\pi}{N}kn}\}_{k=0}^{N-1}$, 其本质是向量的正交分解 (详见第 21.3 节)。因此可以推断——图傅里叶变化也应该是将图信号 $f(v)$ 分解至有限个特定的正交基;

- 进一步讨论离散傅里叶变换的基 $e^{i\frac{2\pi}{N}kn}$ 的特点。将 $e^{i\frac{2\pi}{N}kn}$ 视为关于 n 的连续函数, 作用拉普拉斯算子 Δ 有

$$\Delta e^{i\frac{2\pi}{N}kn} = \frac{\partial^2}{\partial n^2} e^{i\frac{2\pi}{N}kn} = -\frac{4\pi^2 k^2}{N^2} e^{i\frac{2\pi}{N}kn}$$

类比特征值与特征向量的定义 $Ax = \lambda x$, 则由上式可认为离散傅里叶变换选择的基 $e^{i\frac{2\pi}{N}kn}$ 是拉普拉斯算子的特征向量, 相应地图拉普拉斯变换的基也应为拉普拉斯矩阵的特征向量, 而图的拉普拉斯矩阵恰好具有有限个单位正交的特征向量;

- 经上述类比后即可给出图傅里叶变换 (GFT) 与逆图傅里叶变换 (IGFT) 公式如下

$$\text{GFT} : \tilde{f}(\lambda_i) = \sum_{j=1}^n f(v_j)U_{ji} \iff \text{IGFT} : f(v_j) = \sum_{i=1}^n \tilde{f}(\lambda_i)U_{ji}$$

上式中 λ_i, v_j 分别表示拉普拉斯矩阵的第 i 个特征值与图的第 j 个节点, U 为 n 个拉普拉斯矩阵的单位正交特征向量组成的矩阵。上式也可进一步写为矩阵形式

$$\text{GFT} : \tilde{f} = U^\top f \iff \text{IGFT} : f = U\tilde{f}$$

- 根据图傅里叶变换与离散傅里叶变换的对应关系, 显然图傅里叶变换将拉普拉斯矩阵的特征值 λ 与传统时间序列的频率对应, 与本节上文对拉普拉斯矩阵特征值物理意义的解读一致——较大的特征值捕捉了图数据分布的不均匀结构 (高频特征), 而较小的特征值反映了图数据分布的均匀结构 (低频特征)。

18.6 复杂网络基本概念

- 复杂网络理论发展于图论, 其很多基本概念来源于图论, 但特指一种呈现高度复杂性的网络, 常用于对复杂系统进行建模。钱学森对于复杂网络给出了一种严格的定义: 具有自组织、自相似、吸引子、小世界、无标度中部分或全部性质的网络称之为复杂网络;
- 复杂网络可以由图语言、集合论语言与线性代数语言表示, 其中邻接矩阵与拉普拉斯矩阵是最常用的线性代数表示方法;
- 对 (p, q) 图 $G = (V, E)$, 进一步地介绍复杂网络结构的基本静态几何量, 同样源于图论定义:
度、平均度、度分布 度是针对网络节点, 而平均度与度分布则是针对网络本身。顾名思义, 平均度指网络中所有节点度的均值, 而度分布指网络中节点度的概率分布, 有

$$P(k) = \frac{n_k}{p} \quad \int_0^\infty kP(k)dk = K = \frac{2q}{p}$$

式中 K , $P(k)$ 分别指网络的平均度与度分布, n_k 指网络中度值为 k 的节点的数目。在已知网络中节点数的情况下, 平均度给出了网络中的边数, 而度分布则不仅给出边数, 还给出了网络中不同度的节点所占的比例, 因此称平均度为网络的**0**阶度分布特性, 度分布为网络的**1**阶度分布特性;

聚类系数 (clustering coefficient) 聚类系数分为局部聚类系数与全局聚类系数两类, 前者针对节点, 后者针对网络。首先介绍局部聚类系数。如果说节点的度反映了某一节点与其它节点的连接情况, 节点的聚类系数则反映了某一节点的所有相邻节点间的连接情况。具体地, 节点 i 的聚类系数 C_i 定义为节点 i 的 k_i 个相邻节点之间实际存在的边数 e_i 与总的可能的边数之比, 定义式如下

$$C_i = \frac{2e_i}{k_i(k_i - 1)} = \frac{1}{k_i(k_i - 1)} \sum_{j,k \in V} a_{ij}a_{ik}a_{kj} = \frac{\sum_{j,k} a_{ij}a_{ik}a_{kj}}{\sum_{j,k} a_{ij}a_{ik}}$$

观察以上分数可知, 当分母 $a_{ij}a_{ik} \neq 0$ 时说明 i, k, j 三节点间至少存在 ik, ij 两条边, 图论中称这一结构为以 i 为中心的三元组 (**triplet**), 记为 $\{i, (j, k)\}$, 而当分子 $a_{ij}a_{ik}a_{kj} \neq 0$ 时说明 i, k, j 三节点构成三角形, 称为闭三元组 (**closed triplet**), 而无法构成三角形的三元组则称为开三元组 (**open triplet**)⁵。引入三元组的定义, 又可将节点 i 的聚类系数重新定义为包含 i 的闭三元组个数与所有以 i 为中心的三元组个数的比值。

网络的全局聚类系数正是基于三元组, 定义为网络中所有闭三元组的个数与所有三元组个数的比值 (注意一个三角形包括三个闭三元组), 计算公式如下

$$C = \frac{\sum_{i,j,k} a_{ij}a_{ik}a_{kj}}{\sum_{i,j,k} a_{ij}a_{ik}}$$

路径、距离、介数 (Betweenness) 复杂网络中路径与距离的定义已于图论中介绍, 需要注意的是, 在带权图中距离的定义与边权的类型有关。边权可分为相似权 (similarity weight) 与相异权 (dissimilarity weight) 两类, 前者中边权越大表示节点越接近, 无连接时边权取 0, 后者中边权越小表示节点越接近, 无连接时边权取无穷。在相似权或相异权下的距离计算公式分别为

$$d_{ik} = \begin{cases} w_{ik} = w_{ij} + w_{jk}, & \text{similarity weight} \\ \frac{1}{w_{ik}} = \frac{1}{w_{ij}} + \frac{1}{w_{jk}}, & \text{dissimilarity weight} \end{cases}$$

介数分为点介数与边介数, 定义为某一节点或边在任意一对节点间最短路径所经过的次数, 反映了相应节点或边在整个网络中的作用和影响力, 是一个全局几何量, 在交通网络中, 介数值较高的点往往为交通枢纽点, 介数值较高的边往往为关键线路 (如桥、隧、主干路等);

网络直径与平均距离 网络直径 d^G 指网络中任意两点最小距离的最大值, 而网络中所有节点间最小距离的均值则是网络的平均距离 $\overline{d^G}$, 网络的度分布、聚类系数与平均距离是网络性质的三个基本方面;

$$d^G = \max\{d_{ij}\}, i, j \in V \quad \overline{d^G} = \frac{1}{2q} \sum_{i,j \in V} d_{ij}$$

网络稀疏性 类似于节点聚类系数的定义, 网络稀疏程度定义为网络中实际存在的边数与最大可能边数的比值, 即

$$\frac{2q}{p(p-1)}$$

大多数网络都属于稀疏网络, 即 $q \ll 0.5p(p-1)$;

度度相关性 度度相关性 (degree correlation) 针对网络整体, 指代网络中互相连接的任意两节点的度的相关性。在统计学中相关性按强度有强相关与弱相关之分, 按方向又可分为正相关与负相关。类似地, 将网络的度度相关性定性地分为以下三类:

- **同配 (Assortative):** 指度值大的节点倾向于连接度值大的节点;

⁵注意无论是闭三元组或开三元组, $\{i, (j, k)\}$ 与 $\{i, (k, j)\}$ 均表示同一三元组, 但 $\{i, (j, k)\}$ 与 $\{j, (i, k)\}$ 均指不同的三元组。

- 异配 (**Disassortative**)：指度值大的节点倾向于连接度值小的节点；
- 中性 (**Neutral**)：指节点的互相连接与其度值无关。

研究结果表明，众多社会关系网络（如论文合作网络）呈现出同配特性，而信息网络与生物神经网络则表现为异配特性，常用的 ER 模型等则没有任何倾向性。进一步地介绍四种评价（量化）无向图网络度度相关性的指标（方法）：

- **皮尔逊相关系数。**借用统计学中的皮尔逊相关系数量化网络的度度相关性是最简单直观的一类方法。方法 1) 首先初始化两组空序列以保存节点度值；2) 进一步地取出网络中的所有连边并计算每一连边的两节点的度值，将两度值中较大的至于一组序列而较小的至于另一组，最终得到两组长度一致的序列；3) 最后计算两组序列的皮尔逊相关系数，为正值表示网络同配、负值表示异配、接近零表示中性；
- **联合概率分布。**联合概率分布是最常用的一类描述网络度度相关性的指标，联合概率 e_{jk} 定义为网络中随机取一条边，两端点度数分别为 j, k 的概率（或近似为端点度数分别为 j, k 的边数占总边数的比例），定义为

$$e_{jk} = \frac{m_{jk}\mu_{jk}}{2q} \quad \mu_{jk} = \begin{cases} 2, & j = k \\ 1, & j \neq k \end{cases}$$

式中 m_{jk} 表示端点度分别为 j, k 的边数，定义系数 μ_{jk} 旨在解决连边的两端点度值相同的情况。 e_{jk} 具有如下性质

$$e_{jk} = e_{kj} \quad \sum_{j,k} e_{jk} = 1 \quad \sum_j e_{jk} = q_k = \frac{\sum_j m_{jk}\mu_{jk}}{2q} = \frac{kn_k}{2q}$$

式中 q_k 表示网络中随机取一条边的随机一个端点的度值为 k 的概率，称为余度分布。一般地，通过绘制 e_{jk} 的二维分布热力图定性地评价网络度度相关性，因为 e_{jk} 的对称性，二维热力图必然沿对角线对称。当网络同配时，意味着连边两节点度值接近，此时 e_{jk} 主要沿对角线分布；当网络异配时，意味着连边界点度值差距较大，此时 e_{jk} 主要沿横轴和纵轴分布；当网络中性时， e_{jk} 大致呈均匀分布。最后补充联合概率分布与度分布的关系，注意到

$$\sum_j \frac{K}{k} e_{jk} = \frac{1}{k} \frac{2q}{p} \frac{kn_k}{2q} = \frac{n_k}{p} = P(k)$$

可以看到联合概率分布进一步描述了度分布的微观特征，因此称联合概率分布为网络的 2 阶度分布特性；

- **同配系数 (Assortativity coefficient)。**同样地关注联合概率分布 e_{jk} ，注意到如果网络中节点的连接关系与各自的度值无关，即意味着 $q_k, q_j, (k \neq j)$ 相互独立，此时必然有

$$q_j q_k = e_{jk}$$

因此除了定性评价外也可以根据 e_{jk} 与 $q_j q_k$ 的关系定量评价网络度度相关性，同配系数即由此提出，顾名思义即评价网络的同配程度。定义度相关函数如下

$$\sum_{j,k} jk(e_{jk} - q_j q_k) = \sum_{j,k} jk e_{jk} - \sum_j j q_j \sum_k k q_k = \mathbb{E}(jk) - \mathbb{E}^2(k) = \mathbb{E}(jk) - K^2$$

即为网络中连边两节点的度乘积的期望与网络的平均度的平方的差值。由基本不等式可知，当网络同配时上式取正值，反之取负值，故上述函数即可评价网络度相关性，但具体大小受网络规模影响。为方便不同网络度度相关性的比较，对上式归一化得到同配系数 r 。注意到当网络完全同配时有 $e_{jk} = q_j \delta_{jk}$ ，此时度相关函数取得最大值，且该值正好为余度分布 q_k 的标准差 σ_q

$$r = \frac{1}{\sigma_q} \left(\sum_{j,k} jk(e_{jk} - q_j q_k) \right) \quad \sigma_q^2 = \sum_k k^2 q_k - \left(\sum_k k q_k \right)^2$$

$r \in [-1, 1]$, 需要说明的是同配系数 r 的本质即是上文所述皮尔逊相关系数, 两类指标计算结果与具体数值大小完全一致;

- **余平均度 (Excess average degree)**。余平均度是另一类更简洁的指标。对任意节点 i , 其度值为 k_i , 则其余平均度 $\langle k_{nn} \rangle_i$ 定义为另外与其相连的 k_i 个节点的度值的平均值, 计算式如下

$$\langle k_{nn} \rangle_i = \frac{1}{k_i} \sum_j a_{ij} k_j$$

除了针对特定节点 i 的余平均度 $\langle k_{nn} \rangle_i$, 也可以定义针对全体度值为 k 的节点的余平均度 $\langle k_{nn} \rangle(k)$, 即为全体度值为 k 的节点的邻居节点的平均度。统计学上认为 $\langle k_{nn} \rangle(k)$ 与 k 之间存在如下函数关系

$$\langle k_{nn} \rangle(k) = ak^\mu$$

式中 a, μ 为待定系数, 通过拟合确定, 而 μ 值的正负即反映了网络的度相关性——正值表示同配、负值表示异配、接近零表示中性。

富人俱乐部系数 (Rich club connectivity) 富人俱乐部系数与复杂网络中的“富人俱乐部现象”有关, 该现象指即使是一些存在异配特性的网络 (如互联网), 高度节点之间依然存在很高的连接概率的现象。系数即旨在量化网络中的这一拓扑结构, 最早被定义为 $\varphi(k)$

$$\varphi(k) = \frac{2E_{>k}}{N_{>k}(N_{>k} - 1)}$$

式中 $N_{>k}$ 指度数大于 k 的节点的数量, $E_{>k}$ 指这些度数大于 k 的节点间的连边数量, 如果 $\varphi(k)$ 随 k 的增加而增加则网络中存在富人俱乐部现象。但后续研究进一步发现按上述规则即使是某些随机网络也可能存在富人俱乐部现象, 为此进一步提出归一化富人俱乐部系数 $\rho(k)$

$$\rho(k) = \frac{\varphi(k)}{\varphi_{ran}(k)}$$

式中 $\varphi_{ran}(k)$ 指对应的随机化网络的富人俱乐部系数。当 $\rho(k) > 1$ 时认为网络中存在富人俱乐部现象。 $\rho(k)$ 消除了相同度分布下由结构所引起的差异, 更能体现富人俱乐部效应的重要性。

18.7 ER 网络

1. ER 网络是最早的一类复杂网络模型, 于上世纪 50 年代末两位匈牙利数学家 Erdos 与 Renyi 所提出并因此得名。ER 网络包含两种生成方式:

- $G(N, L)$ 模型: 一个随机图由 N 个节点组成, 并且有 L 条边随机连接 L 对节点, 且不形成重边和自环;
- $G(N, p)$ 模型: 一个随机图由 N 个节点组成, 任意两不同节点间存在连边的概率为 p 。

2. 考虑 $G(N, p)$ 模型, 网络中的边数 L 为一随机变量, 服从二项分布

$$P(L) = C_N^L p^L (1-p)^{C_N^2 - L}$$

同样地, 网络的度分布 $P(k)$ 同样服从二项分布

$$P(k) = C_{N-1}^k p^k (1-p)^{N-1-k} \xrightarrow[p \rightarrow 0]{N \rightarrow \infty} e^{-K} \frac{K^k}{k!}$$

套入二项分布相关公式, 可计算以上随机变量的各项统计量

$$\mathbb{E}(L) = \frac{1}{2}pN(N-1) \quad \mathbb{E}(k) = K = p(N-1) \quad \sigma_L^2 = \frac{1}{2}p(1-p)N(N-1)$$

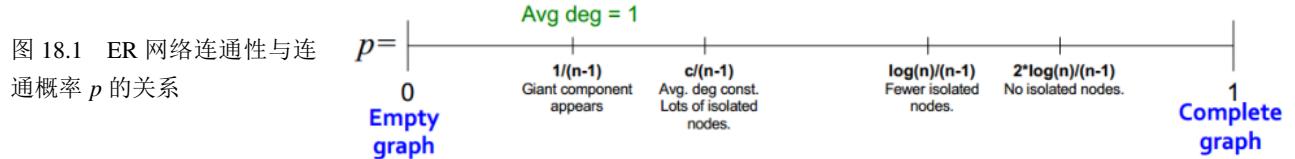
3. 根据网络的节点数 N 及连通概率 p , 即可判别网络的连通性, 具体由网络的平均度 K 判别, 包括以下状态:

亚临界 ($K < 1$) 此时网络中不存在最大连通集团⁶，最大群为树结构，其规模 $\sim \ln N$ ；

临界 ($K = 1$) 此时网络中存在唯一的最大连通集团，其中包含环，其规模 $\sim \ln N^{2/3}$ ，而其它规模小的子图均为树结构；

超临界 ($K > 1$) 此时网络中存在唯一的最大连通集团，其中包含环，其规模 $\sim (p - \frac{\ln N}{N})N$ ；

连通 ($K > \ln N$) 此时网络为连通图。



4. 现实网络的平均度 K 往往远小于节点数 N ，此时即可近似网络度分布服从泊松分布。对于大规模网络，遍历节点将造成较大的资源开销，因此可基于泊松分布近似此计算网络的某些特征，如最大度和最小度。首先介绍网络最大度 k_{\max} 的近似计算方法：已知网络度分布 $P(k) = e^{-K} \frac{K^k}{k!}$ ，此时有

$$P(k > k_{\max}) = e^{-K} \sum_{k_{\max}+1}^{\infty} \frac{K^k}{k!} = 1 - P(k \leq k_{\max})$$

根据泰勒展开余项，有

$$P(k > k_{\max}) = 1 - P(k \leq k_{\max}) \cong e^{-K} \frac{K^{k_{\max}+1}}{(k_{\max} + 1)!}$$

$P(k > k_{\max})$ 即为网络中任一节点的度大于 k_{\max} 的概率，显然大于 0，理论上只要网络规模够大，就一定会有一节点的度值超过 k_{\max} 。而既然假设网络的最大度为 k_{\max} ，即说明 N 相对于 $P(k > k_{\max})$ 不够大，无法保证网络中一定有至少 1 个节点的度超过 k_{\max} ，不妨取临界状态，有

$$N \cdot P(k > k_{\max}) = Ne^{-K} \frac{K^{k_{\max}+1}}{(k_{\max} + 1)!} \cong 1$$

代入 N, K 即可计算得 k_{\max} 。网络最小度 k_{\min} 的近似计算思路同上，差别之处仅在于 $N \cdot P(k < k_{\min}) \cong 1$ ；

5. 同样地，也可近似计算大规模 ER 网络的网络直径 d_{\max} 与平均距离 \bar{d} 。因为 ER 网络中各节点间的连接概率完全独立且相等，故对于任一节点，其一阶近邻数 $\cong K$ ，二阶近邻数 $\cong K^2$ ， n 阶近邻数 $\cong K^n$ 。设网络直径为 d_{\max} ，意味着自任一节点出发最少 d_{\max} 步之后即可遍历完所有节点，有

$$N = 1 + K + K^2 + \dots + K^{d_{\max}} = \frac{K^{d_{\max}+1} - 1}{K - 1} \cong K^{d_{\max}} \implies d_{\max} \cong \frac{\ln N}{\ln K}$$

进一步地也可估算网络的平均距离 \bar{d} ，根据定义得

$$\bar{d} \cong \frac{\sum_{d=1}^{d_{\max}} d K^d}{\sum_{d=1}^{d_{\max}} K^d}$$

分母的计算过程同上

$$\sum_{d=1}^{d_{\max}} K^d = K \frac{K^{d_{\max}} - 1}{K - 1} \cong K^{d_{\max}}$$

而后考虑分子，假设 $f(x) = x + 2x^2 + \dots + nx^n$ ，将其转换为等比数列求和形式

$$\begin{aligned} \frac{1}{x} f(x) = 1 + 2x + \dots + nx^{n-1} &\implies \int \frac{1}{x} f(x) dx = C + x + x^2 + \dots + x^n = C + \frac{x^{n+1} - x}{x - 1} \\ &\implies \frac{1}{x} f(x) = \frac{nx^{n+1} - (n+1)x^n + 1}{(x - 1)^2} \end{aligned}$$

⁶在网络中连通集团 (component) 指所有互通点集组成的网络，而只有在网络中存在一个包含大部分节点的连通集团时才称网络中存在最大连通集团 (giant component)，因此最大连通集团不一定是最大的连通集团

$$\Rightarrow f(x) = \frac{nx^{n+2} - (n+1)x^{n+1} + x}{(x-1)^2} \cong nx^n - (n+1)x^{n-1}$$

因此有 $\sum_{d=1}^{d_{\max}} dK^d \cong d_{\max} K^{d_{\max}} - (d_{\max} + 1)K^{d_{\max}-1}$, 即

$$\bar{d} \cong \frac{d_{\max} K^{d_{\max}} - (d_{\max} + 1)K^{d_{\max}-1}}{K^{d_{\max}}} = d_{\max} - \frac{d_{\max} + 1}{K}$$

可以看到 \bar{d} 与 d_{\max} 具有相同的量级, 一般直接近似为 $\bar{d} \cong d_{\max} \cong \frac{\ln N}{\ln K}$;

6. 同样地, 可以近似计算网络的聚类系数 C , 根据数学期望的运算性质:

$$C = \frac{\sum_{i,j,k} a_{ij}a_{ik}a_{kj}}{\sum_{i,j,k} a_{ij}a_{ik}} \cong \frac{\sum_{i,j,k} p^3}{\sum_{i,j,k} p^2} = p \cong \frac{K}{N}$$

7. 实际网络及其对应的随机网络的度分布、距离、聚类系数具有如下大致关系:

- 实际网络的度分布普遍不满足二项分布或泊松分布, 往往更加重尾;
- 实际网络的平均距离与对应随机网络的平均距离具有相近的量级, 而直径远大于随机网络的直径;
- 实际网络的聚类系数普遍高于对应随机网络的聚类系数。

实际网络往往具有小世界特性, 即规模较大的网络同时有着较小的平均距离和较大的聚类系数, ER 随机网络能较好地反映前者, 但不能反映后者。

18.8 小世界网络

小世界网络中的“小”包括两个层面: 平均距离小 (平均距离大致与其节点数的对数成正比) 同时聚类系数大, 前者反映为网络中两个无直接联系的节点往往只需少数中间结点即可建立联系, 而后者反映为网络中两个直接联系的节点间往往存在其它的联系渠道 (例如你的朋友很可能也是你的另一个朋友的朋友)。很多实际网络都具有较强的小世界特性。

Milgram 小世界实验与六度分离理论

关于人际网络的小世界特性, 不得不提 Milgram 小世界实验与六度分离理论。

Milgram 小世界实验是 1967 年哈佛大学社会心理学家 Milgram 所做的一项试验。Milgram 将一套连锁信件随机发送给居住在内布拉斯加州奥马哈的 160 个人, 信中放了个波士顿股票经纪人的名字, 要求每个收信人将这套信件寄给自己认为比较接近那个股票经纪人朋友。最终, 44 份信件最终送达那位股票经纪人, 而信函到达的平均中间节点为 5 人。根据实验结果 Milgram 推断: 世界上任意两个人的平均距离是 6。“小世界现象”(small world phenomenon) 假设与“六度分离”(six degrees of separation) 理论由此而来。

18.8.1 从 W-S 模型到 N-W 模型——网络的小世界特性

1. ER 网络不是真正的小世界网络, 因此自上世纪末, 一系列小世界网络模型得以提出, 其中 WS 模型与 NW 模型是最经典的两个小世界网络生成模型。1998 年, Watts 和 Strogatz 开创性地提出 WS 模型, 算法首先初始化一个最近邻网络, 其中每个节点都与它相邻的 $2m$ 个节点相连。然后以概率 p 进行断边重连, 即以概率 p 随机选择网络的任意节点作为该边的另一端点断边重连, 而以概率 $1-p$ 保持另一端点不变。在此过程中避免出现重边和自环。当 $p = 0$ 时得到的即是最近邻网络, 当 $p = 1$ 时即为 ER 随机网络, 当 p 取 0-1 之间的较小值时得到的网络即可表现出小世界特性;
2. WS 网络存在两类连边: 与最近邻节点的短程连边以及与远处其它节点的长程连边。两类连边分别描述了社会网络的两个主要性质: 同质性 (homophily) 与弱连接 (weak tie), 例如相近的两个人更可能认识, 同时人们也会与一些不大熟的人建立联系。因为只有在合适的 p 值下 WS 网络才表现出小世界特性, 说明短程连边与长程连边必须达到合适的比例;
3. 由于 WS 网络可能存在孤立点, 不利于网络性质分析 (例如会出现距离无穷的情况), 因此 Newman 与 Watts 进一步提出了 NW 模型, 从根本上避免了孤立点的产生。算法同样初始化一个最近邻网络, 然后以概率 p 新增连边, 即以概率 p 随机选择网络的任意节点作为新的连边的端点, 而以概率 $1-p$ 不新增连

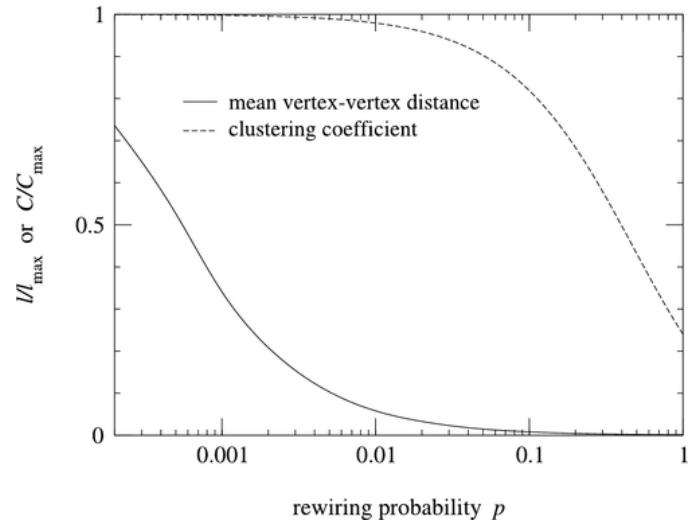


图 18.2 不同重连概率 p 下的网络小世界特性。下方曲线为 p 与网络平均距离的关系，上方曲线为 p 与网络聚类系数的关系，可以看出只要 p 取值合适即可保证较小的平均距离与较大的聚类系数。

边。在此过程中同样避免出现重边和自环。当 $p = 0$ 时得到的即是最近邻网络，当 $p = 1$ 时即为 ER 随机网络与最近邻网络的叠加，当 p 取 0-1 之间的较小值时得到的网络即可表现出小世界特性。

18.8.2 从 N-W 模型到 Kleinberg 模型——网络的可导航性

- 包括 WS 模型、NW 模型在内的一系列小世界网络模型均能很好地反映实际网络的小世界特性。然而回顾 Milgram 实验，除了说明现实人际网络存在相对较小的平均距离外，还说明网络中的个体能（仅凭局部网络信息）有效找到并且利用这些短路径——人们仅凭自己认识的人与收信人即能有效地找到可能认识收信人的中间人，这一特性被解释为网络的可导航性；
- WS 模型说明只要合适比例的短程连边与长程连边即可反映出网络的小世界特性，同样地网络的可导航性也必然与这一结构有关。2000 年 Kleinberg 在 Nature 上发表一篇文章，基于 NW 模型提出 Kleinberg 模型，通过引入一个聚集指数 (clustering exponent) 解释这一结构；
- Kleinberg 提出了一个添加了长程连边的 $n \times n$ 二维规则网络，网络中都与离它网格距离 (lattice distance)⁷ 为 m 的所有节点存在短程连边，同时每个节点还拥有 n 条长程连接，任意两节点 u, v 间存在长程连接的概率与它们网格距离的幂函数 $[d(u, v)]^{-q}$ 呈正比。显然，当 $q = 0$ 时网络退化为普通的 NW 网络，随着 q 的增加长程连接越来越向节点附近靠拢；
- Kleinberg 模拟 Milgram 实验中志愿者所采用的分散式搜索 (decentralized search)——假设信息从节点 s 发出，目标为节点 t ， s 知道自己与 t 在网络中的位置，也知道其所有邻居（短程与长程连接）的位置，但不知道其邻居的邻居的位置，因此其只能将信息传递给离 t 最近的邻居。显然，仅依靠一阶近邻信息的分散式搜索并不能保证得到最短的路径，因此只有当节点的长程连接尽可能均匀时才能保证较好的搜索效果，体现出小世界网络平均距离小的特点（即在认识人数相同的情况下，交际圈越广、远近都认识的人越有可能发挥小世界网络的优势）；
- Kleinberg 证明，当网络规模趋于无穷时，当且仅当 $q = 2$ 时得到的网络可达到最佳的搜索效率，同时表现出小世界特性与可导航性。其进一步推广指出，对于 k 维规则网络，对应的最优 q 值即为 k 。以下定性分析：
- 构造单调递增非负数列 $\{d(j)\}$ 。对于给定的任意节点 u , $d(j)$ 将网络的其它节点划分为多个区域 A_0, A_1, \dots ，其中 A_j 包含所有距 u 的距离 $d \in [d(j), d(j+1)]$ 的节点。建议令 $\{d(j)\}$ 为等比数列（即对数分箱）

$$d(j) = a \cdot b^j \quad b > 1$$

此时有 $\frac{d(j+1)}{d(j)} = b$ 为常数，这一特性保证了 $\{A_j\}$ 的划分精度随距 u 的距离动态调整——距 u 越近，划分精度越大，反之精度越小，而且任意相邻区域间的差别仅与常数 b 有关，与区域距 u 的位置（即 j ）无

⁷ 在这一规则网络中，所谓网格距离类似于曼哈顿距离，两节点的网格距离仅与两节点在网络的空间位置有关，而与具体的连边无关。两节点的网格距离最小为 1。

关。考虑到与节点 u 相距 r 的节点数正比于 πr , u 与这些节点存在长程连接的概率正比于 r^{-q} , 因此 u 在区域 $A(j)$ 内的长程连接数大致为

$$C \int_{A(j)} r \cdot r^{-q} dr = C \int_{a \cdot b^j}^{a \cdot b^{j+1}} r^{1-q} dr = \begin{cases} C \cdot \ln b & q = 2 \\ C \cdot \frac{(a \cdot b^j)^{2-q}}{2-q} (b^{2-q} - 1) & q \neq 2 \end{cases}$$

显然, 当且仅当 $q = 2$ 时 u 在区域 $A(j)$ 内的长程连接数与 $A(j)$ 无关, 为常量, 意味着网络中的节点具有相对均匀的长程连接, 因此网络既能体现小世界特性也能体现可导航性。

18.9 无标度网络

- 首先介绍标度不变性 (或尺度不变性 **scale invariance**): 对函数 $f(x)$, 若对于任意常数 a , 存在相应的常数 b , 使得

$$f(ax) = bf(x)$$

则称 $f(x)$ 具有标度不变性, 即对函数的横坐标缩放均可通过对函数纵坐标缩放实现。进一步地推导 $f(x)$ 解析式, 首先令 $x = 1$, 有 $b = \frac{f(a)}{f(1)}$, 代入得

$$f(ax) = \frac{f(a)}{f(1)} f(x)$$

对 a 求偏导, 此时有

$$\frac{\partial f(ax)}{\partial a} = \frac{f(x)}{f(1)} \frac{df(a)}{da} \Rightarrow x \frac{df(ax)}{dax} = \frac{f(x)}{f(1)} f'(a) \Rightarrow xf'(ax) = \frac{f'(a)}{f(1)} f(x) \Rightarrow f(x) = (cx)^{f'(a)/f(1)}$$

为构造微分方程, 令 $a = 1$, 则

$$xf'(x) = \frac{f'(1)}{f(1)} f(x) \Rightarrow f(x) = (cx)^{f'(1)/f(1)}$$

显然, 满足标度不变性的函数为幂函数;

- 进一步地, 对随机变量 X , 若其概率密度函数满足标度不变性, 则称其服从幂律分布 (详见第 23.7.5 节), 其概率密度函数为

$$f(x) = cx^{-\lambda} \quad c, \lambda > 0$$

标度不变性等价于无标度性 (**scale free**), 因此称网络的度分布服从幂律分布 (**Power-law distribution**) 时定义其为无标度网络 (**Scale Free Network**)。具体地, 网络的度分布函数

$$P(k) = ck^{-\lambda} = \begin{cases} \frac{\lambda-1}{k_{\min}} \left(\frac{k}{k_{\min}}\right)^{-\lambda} & \text{Continuous} \\ \frac{k^{-\lambda}}{\sum_{n=1}^{\infty} n^{-\lambda}} & \text{Discrete } (k_{\min} = 1) \end{cases}$$

因此有 $\ln P(k) \propto -\lambda \ln k$, 即网络度分布的双对数坐标图大致呈线性。因为度分布服从幂律分布, 无标度网络的典型特征即是网络中的大部分节点只和很小部分节点连接, 而极少数节点与非常多的节点连接, 因此无法仅由平均度、度方差等度分布指标反映网络连接的离散或聚合情况, 无标度网络因此得名, 最先由匈牙利物理学家巴拉巴西 (Albert-László Barabási) 于 1999 年提出。现实世界的网络, 从经过数十亿年演化的蛋白质互动网, 到人类文明产生的万维网、因特网、城市网, 大多为无标度网络;

- 几乎所有自然竞争演化而成的网络都具有无标度特性, 这是因为自然选择下网络的生长机制是优先连接 (所谓优胜劣汰、强者恒强)。在网络扩张过程中, 度值更高的节点更可能获得新的连接, 长此以往一个随机初始化的网络将发展为无标度网。在无标度网络中极少数节点维系着绝大多数连边, 大多数其它普通节点无关紧要, 因此这类网络对意外攻击有很强的承受能力, 但面对协同攻击时则显得脆弱;

4. 同样地，基于其度分布函数，同样可以估算无标度网络的最大度 k_{\max} ，注意到离散度分布和连续度分布时具有不同的度分布函数。当网络度分布连续时，有

$$P(k > k_{\max}) = \int_{k_{\max}}^{\infty} \frac{\lambda - 1}{k_{\min}} \left(\frac{k}{k_{\min}}\right)^{\lambda} dk = \left(\frac{k}{k_{\min}}\right)^{1-\lambda} \Big|_{\infty}^{k_{\max}} = \left(\frac{k_{\max}}{k_{\min}}\right)^{1-\lambda}$$

另外考虑到 $N \cdot P(k > k_{\max}) \cong 1$ ，因此有

$$\left(\frac{k_{\max}}{k_{\min}}\right)^{1-\lambda} \cong \frac{1}{N} \iff k_{\max} \cong k_{\min} N^{\frac{1}{\lambda-1}}$$

当网络度分布离散且为正整数时，有

$$P(k > k_{\max}) = \frac{\sum_{k=k_{\max}+1}^{\infty} k^{-\lambda}}{\sum_{k=k_{\min}}^{\infty} k^{-\lambda}} \cong \frac{\int_{k_{\max}+1}^{\infty} k^{-\lambda} dk}{\int_{k_{\min}}^{\infty} k^{-\lambda} dk} = \left(\frac{k_{\max}+1}{k_{\min}}\right)^{1-\lambda} \cong \frac{1}{N} \implies k_{\max} \cong k_{\min} N^{\frac{1}{\lambda-1}} - 1$$

可以看到， k_{\max} 总体上随网络规模的增加而增加；

5. 无标度网络的特性与其幂指数 λ 的取值密切相关。具体地：

- $1 < \lambda < 2$ 时， k_{\max} 的增长快于 N ，又因为 $k_{\max} \leq N - 1$ ，显然网络规模不可能无限扩大。此时网络的平均距离 \bar{d} 与网络规模无关，近似为常数；
- $\lambda = 2$ 时， k_{\max} 的增长正比于 N ；
- $2 < \lambda < 3$ 时， k_{\max} 的增长慢于 N ，且有 $\bar{d} \cong \ln \ln N$ ，网络具有超小世界特征⁸，也是一般认为的无标度网络。需要说明的是此时网络的平均度 K 随网络规模增加而收敛，但度分布的二阶矩 γ 随网络规模增加而发散；
- $\lambda = 3$ 时， k_{\max} 的增长慢于 N ，且有 $\bar{d} \cong \frac{\ln N}{\ln \ln N}$ ；
- $\lambda > 3$ 时， k_{\max} 的增长慢于 N ，且有 $\bar{d} \cong \frac{\ln N}{\ln K}$ ，网络具有小世界特性。此时度分布的一阶矩（平均度）与二阶矩均随网络规模增加而收敛。

18.9.1 BA 无标度网络模型

1. 巴拉巴西通过观察现实中具有无标度特性的网络，指出这些网络的形成离不开以下两种机制的共同作用：

- 增长机制 (Growth)：随着时间的延长，不断有新的节点加入网络并形成新的连接；
- 偏好链接机制 (Preferential attachment)：新节点建立新连接时更可能选择度大节点。

基于以上发现，巴拉巴西提出 BA 模型，这也是最早的无标度网络生成模型⁹。网络的生成流程如下：

- 初始一个由 m_0 个孤立节点组成的网络；
- 每隔一个时间步（记为单位 1），往网络中加入一个新的节点，新节点将选择 $m (m \leq m_0)$ 个旧节点与之建立连接。显然，至 t 时刻，网络中共包含 $m_0 + t$ 个节点、 mt 条边，网络的总度数为 $2mt$ ；
- 对于旧节点 i ，记其能与新节点建立连接的概率为 $\Pi(k_i)$ ，有

$$\Pi(k_i) = \frac{k_i}{\sum_{j=1}^{m_0+t-1} k_j} = \frac{k_i}{2mt}$$

2. 在同一篇文献中，巴拉巴西基于平均场理论 (mean-field theory)¹⁰ 最早地分析了 BA 网络模型的度分布，以下进行简单介绍；

3. 首先分析网络中任意节点 i 的度 k_i 随时间的变化关系，有

$$\frac{\partial k_i}{\partial t} = \frac{\Delta k_i}{\Delta t} = m\Pi(k_i) = \frac{k_i}{2t}$$

⁸满足 $\bar{d} \cong \ln \ln N$ 的网络即可认为具有超小世界特征，满足 $\bar{d} \cong \ln N$ 的网络即可认为具有小世界特征。

⁹Barabasi A, Albert R, Jeong H, et al. Mean-field theory for scale-free random networks[J]. Physica A-statistical Mechanics and Its Applications, 1999, 272(1): 173-187.

¹⁰平均场理论又称平均场近似，最早提出于统计物理学领域，如今在多个领域都有应用，其本质是一种将多体问题分解为单体问题的近似方法。在多体问题，不同个体会相互影响，从而极大地增加了分析的复杂性。此时当个体数量非常多时，可首先研究多个个体相互作用所形成的场，即为平均场，再对处于平均场中的单个物体进行分析，从而降低问题复杂性。举例分析，在鱼群中，每条鱼的行为决策都会受周围鱼行为的影响，同时也会影周围鱼的行为决策，此时在一个大规模鱼群中分析单条鱼的行为将非常困难，因此可以首先分析鱼群整体的行为，再根据单条鱼在鱼群中的位置估算其行为。

上式 $m\Pi(k_i)$ 中 m 表示网络在一个时间步内增加的边数，而 $\Pi(k_i)$ 表示节点 i 获得连边的概率，两者乘积即表示一个时间步内节点 i 新增度的期望，注意到所谓“期望”即是平均场近似，实际上任意节点度值的变化应该与其它节点度值的变化互为相关，而此时从网络整体的角度分析，跳出了节点间的相互影响，即是所谓的平均场近似。记节点 i 加入网络的时间为 t_i ，则考虑初始条件 $k_i(t_i) = m$ 解以上微分方程，有

$$k_i(t) = m \left(\frac{t}{t_i} \right)^{0.5}$$

4. 进一步地推导 $k_i(t)$ 的累积概率分布 $P(k_i(t) \leq k)$ ，有

$$P(k_i(t) \leq k) = P \left(m \left(\frac{t}{t_i} \right)^{0.5} \leq k \right) = P \left(t_i \geq \frac{m^2 t}{k^2} \right) = 1 - P \left(t_i < \frac{m^2 t}{k^2} \right)$$

在总共 $m_0 + t$ 个节点中，只有 1 个节点在 $t = t_i (t_i > 0)$ 时刻加入，假设网络中节点加入的时间步是均匀的，则不妨假设 $P(t_i)$ 近似服从均匀分布

$$P(t_i) = \frac{1}{m_0 + t} \implies P \left(t_i < \frac{m^2 t}{k^2} \right) = \frac{1}{m_0 + t} \cdot \frac{m^2 t}{k^2} \implies P(k_i(t) \leq k) = 1 - \frac{1}{m_0 + t} \cdot \frac{m^2 t}{k^2}$$

5. 可以看到， $P(k_i(t) \leq k)$ 的表达式与具体的 i 无关，说明对任意节点均成立，因此得到网络的度分布

$$P(k) = \frac{\partial P(k_i(t) \leq k)}{\partial j} = \frac{2m^2 t}{m_0 + t} k^{-3} \simeq 2m^2 k^{-3}$$

综上所述，当 $t \rightarrow \infty$ 时，BA 网络度分布近似服从 $\lambda = 3$ 的幂律分布，与 m_0, m 的具体取值无关。除平均场理论外，其他学者也曾基于主方程、速率方程等统计物理学方法证实以上结论（BA 网络度分布近似服从 $\lambda = 3$ 的幂律分布）；

6. 另外，巴拉巴西进一步证明，BA 网络中的增长机制与偏好连接机制缺一不可：

- 当不考虑偏好连接仅考虑增长机制时，此时新节点随机选择任意旧节点建立连接，即 $\Pi(k_i) = \frac{1}{m_0 + t - 1}$ 。显然以上过程不具有记忆性，即新选择与旧选择无关，因此可推断网络的度分布收敛于指数分布。按以上流程推导，有

$$\begin{aligned} \frac{\partial k_i}{\partial t} &= \frac{m}{m_0 + t - 1} \implies k_i = m (\ln(m_0 + t - 1) - \ln(m_0 + t_i - 1) + 1) \\ P(k) &= \frac{m_0 + t - 1}{m(m_0 + t)} \exp \left(1 - \frac{k}{m} \right) \simeq \frac{1}{m} \exp \left(1 - \frac{k}{m} \right) \end{aligned}$$

- 当不考虑增长机制仅考虑偏好连接时，此时初始化一个包含 N 个节点 0 条边的网络，每一时间步随机选择一节点并以概率 $\Pi(k_i) = \frac{k_i}{\sum_j k_j}$ 同节点 i 建立连接，每次仅生成一条连边，直至网络变为完全图。同样地基于平均场理论，有

$$\frac{\partial k_i}{\partial t} = \frac{1}{N} + \frac{N}{N-1} \frac{k_i}{2t} \implies k_i = \frac{2(N-1)}{N(N-2)} t + C t^{\frac{N}{2(N-1)}} \simeq \frac{2}{N} t$$

随着时间增长，网络的度分布将从初始的幂律分布转化为正态分布。

第 19 章

启发式 (Heuristic) 算法与元启发式 (Meta-heuristic) 算法

启发式算法是一类在求解某个具体问题时，在可以接受的时间和空间内能给出可接受的可行解，但又不保证求得最优解（以及可行解与最优解的偏离）的算法的总称。通用性、稳定性以及较快的收敛性是衡量启发式算法性能的主要标准。启发式算法可分为传统启发式算法和元启发式算法。传统启发式算法包括构造型方法、局部搜索算法、松弛方法、解空间缩减算法等。元启发式算法是启发式算法的改进，是随机算法与局部搜索算法相结合的产物，包括禁忌搜索算法、模拟退火算法、遗传算法、蚁群优化算法、粒子群优化算法、人工鱼群算法、人工蜂群算法、人工神经网络算法等。一般地，因为传统启发式算法是通过对原问题适当简化而达到较快的求解速度，因而应用场合较小，一类算法往往只能针对特定问题，而元启发式算法的应用范围更大。

19.1 场景分析 (*Scenario analysis*) 与场景缩减 (*Scenario reduction*)

- □ ×

简介

场景分析 (*scenario analysis*) 是求解不确定性问题的一种方法。与动态规划或强化学习等求解不确定性问题的算法不同，**场景分析将所有不确定性信息分解，从而将不确定性问题转化为多个确定性问题并求解**。一种“场景”对应一种可能情况。具体地，称一个可能的时间序列为一个场景 s ，所有场景组成场景集 S 。**场景集可由历史数据生成，当缺乏历史数据时也可采用蒙特卡罗模拟得到。**

场景缩减 (*scenario reduction*) 是场景分析领域的重要技术，属于启发式算法，最早提出于 2003 年^a。当不确定性因素过多维度过大时，场景分析法可能得到规模过大的场景集，从而产生较大的计算负担。场景缩减可以理解为一种聚类或降维算法，**缩减原场景集得到一个近似子集以此替代原场景集，并保证缩减后的场景集可最好地代表原场景集，在减小计算量的同时保证计算效果**。同步回代缩减法 (*simultaneous backward reduction*) 与快速前向选择法 (*fast forward selection*) 是两种经典的场景缩减算法。

^aHolger, HeitschWerner, Römisch. Scenario Reduction Algorithms in Stochastic Programming[J]. Computational Optimization & Applications, 2003.

19.1.1 基本定义

- 记 T 时段下的一个场景为 $s(T)$ ，可简写为 s ，则场景 s 下的状态时间序列为 $l(s) = \{l^1(s), l^2(s), \dots, l^T(s)\}$ ， $l^\tau(s)$ 表示场景 s 在 $t = \tau$ 时刻的数值；
- 定义场景 s 发生的概率为 P_s ，则易知

$$P_s = \Pr[l(s)] = \Pr[l^1(s)] \cdot \prod_{\tau=2}^T \Pr\left\{\Pr[l^\tau(s)] \mid \Pr[l^1(s)], \dots, \Pr[l^{\tau-1}(s)]\right\}$$

- 进一步地定义场景 i, j 之间的距离 $d(l(i), l(j))$ ，距离越小则两场景的相似度越高，以 L2 范数表示

$$d(l(i), l(j)) = \|l(i) - l(j)\|_2 = \sqrt{\sum_{\tau=1}^T |l^\tau(i) - l^\tau(j)|^2}$$

4. 定义原场景集为 S , 减去场景集为 J , 则缩减后的场景集 $I = S - J$ 。为使得 I 具有尽可能大的代表性, 应该使得 J 与 I 尽可能相似, 从而保证 J 的删去不会造成过多的信息量损失。定义 I, J 之间的距离 $D(I, J)$

$$D(I, J) = \sum_{j \in J} P_j \cdot \min_{i \in I} d(l(i), l(j))$$

距离越小则两集合的相似度越高, 应使得 $D(I, J)$ 尽可能小。

19.1.2 同步回代缩减法 (simultaneous backward reduction, SBR)

简要介绍同步回代缩减法的流程。定义预设的最终缩减场景集 J 的大小 K , 并初始化迭代次数 $k = 0$ 与缩减场景集 $J = J^0$ 。每次迭代时从集合 I^{k-1} 中选择一个场景 j^k 剔除, 从而有 $I^k = I^{k-1} - j^k$, $J^k = J^{k-1} + j^k$, 具体地:

- 对于第 1 次迭代, 选择使得下式最小的场景 j^1 将其剔除

$$P_j \cdot \min_{i \neq j} d(l(i), l(j))$$

- 对于第 k 次迭代, 选择使得下式最小的场景 j^k 将其剔除

$$D(I^{k-1} - \{j^k\}, J^{k-1} + \{j^k\}) = \sum_{j \in J^{k-1} + \{j^k\}} P_j \cdot \min_{i \in I^{k-1} - \{j^k\}} d(l(i), l(j)), \quad j^k \in I^{k-1}$$

- 剔除场景 j^k 后, 应该使得 I^k 中所有场景的概率和仍为 1, 故将 j^k 在 I^{k-1} 时所对应的概率 P_j^{k-1} 并入 I^{k-1} 中与 j^k 最接近的场景 i^k , 即

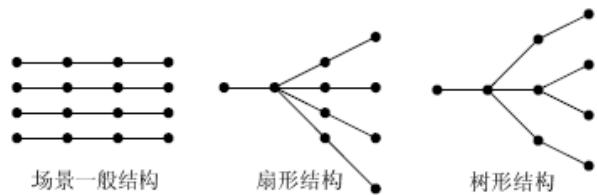
$$P_i^k = P_i^{k-1} + P_j^{k-1}, \quad i^k = \arg \min_{i \in I^{k-1} - \{j^k\}} d(l(i), l(j^k))$$

19.1.3 快速前向选择法 (fast forward selection, FFS)

19.1.4 场景树 (scenario tree)

1. 场景树是一类特殊的场景集形式。一般地场景集由抽样得到, 此时场景集中的每一场景 (时间序列) 互相独立, 将场景集中的所有时间序列以时序图的形式画在一张图上, 将得到一系列杂乱交错的线条。然而这类场景结构并不能很好地反映这一随机过程在多个连续时段的随机变化, 因此有必要将场景集转化为树结构, 即称场景树。假设一场景树 S 中包含 n 个场景, 则一般 n 个场景共享同一个根节点, 并随时间逐渐分叉, 最终形成包含 n 片叶子的树结构;

图 19.1 场景集结构类型



2. 当原场景集为一般结构时, 缩减后的场景集也是一般结构, 因此可以对缩减后的场景集进一步处理成场景树的形式。场景树的生成算法本质上即是场景缩减算法, 相当于从 T 时段向初始时段逐次进行场景缩减;
3. 记从初始时段到 t 时段的场景为 $s(t)$, 所有场景 $s(t)$ 构成场景集 $S(t)$, 假设场景集 $S(T)$ 中包含 n 个场景。则首先令 $t = T - 1$, 对场景集 $S(t)$ 进行一次场景缩减, 此时 $S(T)$ 中仍包括 n 个场景, 但 $S(T-1)$ 中仅包括 $n-1$ 个场景; 再令 $t = t-1$ 进行一次场景缩减, 并重复迭代至 $t=1$, 使得 $S(1)$ 中仅包含 1 个场景, 从而 $S(T)$ 将表现出树状结构。

19.2 同步扰动随机逼近算法 (*Simultaneous perturbation stochastic approximation, SPSA*)

随机逼近简介

- □ ×

随机逼近法是一种在考虑随机误差干扰下的无导数优化方法，适用于不知道函数表达式且函数值估计存在误差的情况。随机逼近法基于变量 x_1, x_2, \dots 及其对应的随机变量 y_1, y_2, \dots ，通过迭代计算逐步优化逼近最优解。常用的迭代算法包括 Robbins-Monro (RM) 算法和 Keifer-Wolfowitz (KW) 算法。基于 KW 算法进行变形，可以得到有限微分随机逼近算法 (FDSA)、随机方向的随机逼近算法 (RDSA) 和同步扰动随机逼近算法 (SPSA)。

- 顾名思义，同步扰动随机逼近算法对目标函数 $y(x)$ 于可行点 x 附近进行随机微小扰动 Δ ，从而估计目标函数 $y(x)$ 于可行点 x 处的梯度信息 $g(x)$ ，进而由梯度下降进行优化；
- 考虑一个最小化问题。记决策变量 x 的维度为 p ，算法超参数包括 a, c, A, α, γ ；
- 对于第 k 次迭代，记当前可行解为 x_k 。随机生成扰动向量 Δ_k ，向量中的元素随机取-1 或 1。同时计算扰动尺度 c_k ，即可得到随机扰动 $c_k \Delta_k$

$$c_k = \frac{c}{k^\gamma}$$

- 对可行解 x_k 进行随机扰动，得到可行解 $x_k - c_k \Delta_k$ 与 $x_k + c_k \Delta_k$ 对应的目标函数估计值 $y(x_k - c_k \Delta_k)$ 和 $y(x_k + c_k \Delta_k)$ ，从而得到相应位置的梯度估计值 $g_k(x_k)$

$$g_k(x_k) = \frac{y(x_k + c_k \Delta_k) - y(x_k - c_k \Delta_k)}{2c_k \Delta_k} = \frac{y(x_k + c_k \Delta_k) - y(x_k - c_k \Delta_k)}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix}$$

- 基于更新步长 α_k ，即可对当前的可行解 x_k 进行更新

$$x_{k+1} = x_k - \alpha_k g_k(x_k) = x_k - \frac{\alpha}{(A + k)^\alpha} \cdot g_k(x_k)$$

- 以上即为最基本的 SPSA 算法。对于一个多决策变量优化问题，当目标函数不可微时，基于 SPSA 算法优化时每次迭代仅需两次采样即可估计梯度，计算负担较小，因而算法在多变量黑箱优化时得到广泛应用；
- 在经典 SPSA 算法的基础上又发展出诸多改进算法，主要是为改进经典 SPSA 算法的如下部分缺陷：
 - 不适用于离散变量优化；
 - 多变量优化时假设各决策变量的扰动对目标函数的贡献相等，造成一阶梯度估计结果误差较大，影响优化速度和最终收敛；
 - 未能挖掘二阶梯度信息，影响后期收敛效果。

19.2.1 Weighted-SPSA (W-SPSA)

- W-SPSA 算法由 Lu 等于 2015 年提出¹，旨在解决经典 SPSA 算法一阶梯度估计时噪声过大的问题。自提出后迅速跻身交通仿真 OD 标定的主流方法之一；
- 算法认为，对于包含 m 个决策变量 $x = [x_1, x_2, \dots, x_m]$ 和 n 个目标分量 $y(x) = \sum_{j=1}^n y_j(x)$ 的黑箱优化问题，经典 SPSA 算法梯度估计的噪声主要有两个来源：
 - 一方面是假设不同决策变量 x_i 对 y 的贡献相等；
 - 另一方面是假设任意决策变量 x_i 对 y 的各分量 y_j 的贡献相等。

上述两方面问题本质上源于同一个问题——经典 SPSA 算法为减少计算对梯度的估计过于简单，估计梯度 $\frac{y(x + \Delta) - y(x - \Delta)}{2\Delta}$ 时既未区分 x_i 的影响，也无法体现 y_j 的区别；

¹Lu, L., Xu, Y., Antoniou, C., & Ben-Akiva, M. (2015). An enhanced SPSA algorithm for the calibration of Dynamic Traffic Assignment models. *Transportation research part C: emerging technologies*, 51, 149-166.

3. 为改进上述缺陷，直观的做法即是引入权重矩阵 $W \in \mathbb{R}^{m \times n}$ ，其元素 w_{ij} 表征决策变量 x_i 对目标分量 y_j 的影响（相关性），W-SPSA 算法由此得名；
4. 基于权重矩阵 W ，W-SPSA 算法估计梯度时即可区分任意决策变量 x_i 的差异。具体地，在第 k 次迭代时，可行解 x^k 的第 i 个分量 x_i^k 的梯度 $g_i^k(x^k)$ 按下式计算

$$g_i^k(x^k) = \frac{\sum_{j=1}^n w_{ij}^k (y_j(x^k + c^k \Delta^k) - y_j(x^k - c^k \Delta^k))}{2c_i^k \Delta_i^k}$$

算法的其它流程与经典 SPSA 算法保持一致；

5. 综上，W-SPSA 算法的关键在于权重矩阵 W 的估计。需要说明的是，并没有完全通用的估计 W 的方法，不过一般情况下可以令 $w_{ij}^k = \frac{\partial y_j}{\partial x_i^k}$ ，由线性回归求解。

19.3 遗传算法 (Genetic algorithm, GA)

简介

- □ ×

遗传算法 (Genetic Algorithm) 是模拟达尔文进化论自然选择和基因遗传、重组、变异过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，不需要确定的规则就能自动获取和指导优化的搜索空间，自适应地调整搜索方向。

遗传算法的核心内容包括参数编码、初始种群设定、适应度函数设计、遗传操作设计、控制参数设定共五个部分，其中遗传操作设计由选择 (selection)、重组 (crossover)、变异 (variation) 三部分组成。^a

^a 【算法】超详细的遗传算法 (Genetic Algorithm) 解析：<https://www.jianshu.com/p/ae5157c26af9>

19.3.1 算法思想

1. 由基因学说与达尔文进化论描述的生物进化过程如下：
 - (a) 生物的基因型 (genotype) 表现为某种编码 (coding)，基因型决定表现型 (phenotype)，在特定的环境下不同的表现型具有不同的适应度 (fitness)，适应度越高的个体越有可能被选择，被选择的个体的基因型将通过婚配过程传递给后代；
 - (b) 子代的基因型很大程度由父母基因型决定：父母各自的基因型通过基因重组形成新的基因型，这些基因型由有一定概率发生突变，新的基因型代表新的个体，具有新的表现型和适应度；
 - (c) 经过以上多代遗传变异，得到的个体将对环境具有普遍高的适应度，这些个体即可以被认为是该环境下的“最优解”。
2. 将以上生物进化过程抽象得到的遗传算法的大致流程：
 - (a) 首先初始化一组编码，编码代表个体的基因型，与表现型形成某种映射（例如二进制编码与十进制实数之间的映射），得到不同编码的适应度（与目标函数值有关），并根据适应度筛选掉一定数量的编码；
 - (b) 对选择的编码进行重组得到新的一组编码，并随机地对得到的编码进行变异形成新的个体，计算新编码的适应度；
 - (c) 重复以上循环直至收敛或达到最大迭代次数，即得到问题的最优解。
3. 遗传算法中包括重组概率 P_c 、变异概率 P_m 和种群规模三个重要参数。其中一般建议 $P_c \in [0.4, 0.99]$, $P_m \in [0.0001, 0.1]$ ，而种群规模根据需求可在 10 至 200 之间选择。

19.3.2 常用编码方法 (Coding)

编码是应用遗传算法时要解决的首要问题，也是设计遗传算法的关键步骤。编码方法影响到交叉算子、变异算子等遗传算子的运算方法，很大程度上决定了遗传进化的效率。De Jong 提出编码应该满足以下规则：

1. 有意义积木块编码规则：编码应易于生成与所求问题相关的短距和低阶的积木块；
2. 最小字符集编码规则：编码应采用最小字符集，以使问题能够得到自然、简单的表示和描述。

编码方法	描述	特点
二进制编码	与四种碱基编码的基因序列模型最为相似，由 0、1 进行编码，只要序列足够长即可以表示足够多的特征。	易于编码、解码、重组突变等操作，但对于高精度问题局部搜索能力较差，需要较长的序列。
浮点数编码	将个体的每个基因用某一范围内的一个浮点数来表示，此时变异和重组往往由四则运算实现。	精度和计算效率较高，便于和经典方法混用，且便于处理复杂的约束条件，但需控制浮点数的范围。
符号编码	取用无数值含义、而只有代码含义的符号集表示基因序列，如 {A, B …}。	符合有意义积木块编码原则，但需要认真设计交叉、变异等遗传操作以满足问题的各种约束要求。

19.3.3 常用选择算法 (Selection)

选择算法	描述
轮盘赌选择 (Roulette Wheel Selection)	每个个体进入下一代的概率等于它的适应度值与整个种群中个体适应度值和的比例。选择误差较大。
随机竞争选择 (Stochastic Tournament)	每次按轮盘赌选择一对个体，让这两个个体进行竞争，适应度高的被选中，如此反复，直到选满为止。
均匀排序	对群体中的所有个体按期适应度大小进行排序，基于这个排序来分配各个个体被选中的概率。
最佳保存策略	当前群体中适应度最高的个体不参与交叉运算和变异运算，而是用它来代替掉本代群体中经过交叉、变异等操作后所产生的适应度最低的个体。

19.3.4 常用重组算法 (Crossover)

重组是指由两组编码生成新编码的过程，对不同的编码方式，可采用的重组算法不尽相同，以下仅介绍二进制编码与浮点数编码时的常用重组算法。

重组算法	描述
单点交叉 (One-point Crossover)	在个体编码串中只随机设置一个交叉点，在该点相互交换两个配对个体的部分染色体。
多点交叉 (Multi-point Crossover)	在个体编码串中随机设置了两个或多个交叉点，然后再进行部分基因交换。
均匀交叉 (Uniform Crossover)	两个配对个体的编码串上的编码以相同的交叉概率进行交换，从而形成两个新个体。
算术交叉 (Arithmetic Crossover)	由两个个体编码串的线性组合而产生出两个新的个体。该操作对象一般是由浮点数编码表示的个体。

19.3.5 常用变异算法 (Mutation)

变异是指将一组编码中的一个值随机替换为另一个值形成新编码的过程。

变异算法	描述
基本位变异 (Simple Mutation)	依变异概率，随机替换编码中某一位或某几位的值。
均匀变异 (Uniform Mutation)	用符合某一范围内均匀分布的随机数，依某一较小的突变概率替换编码中的各个值。特别适用于在算法的初级运行阶段。
边界变异 (Boundary Mutation)	随机取编码上的两个对应边界基因值之一去替代原有基因值。特别适用于最优点位于或接近于可行解的边界时的一类问题。
非均匀变异	对原有的基因值做一随机扰动，以扰动后的结果作为变异后的新基因值。
高斯近似变异	进行变异操作时用符号均值为 P 的平均值，方差为 P^{**2} 的正态分布的一个随机数来替换原有的基因值。

19.4 差分进化 (*Differential evolution, DE*)

简介

- □ ×

差分进化 (differential evolution, DE) 算法是由 Rainer Storn 和 Kenneth Price 于 1997 年在遗传算法等进化算法 (evolution algorithm) 思想的基础上提出的，是一种连续变量优化算法。差分进化算法可视为经典遗传算法的改进。与遗传算法相似，差分进化算法同样包括变异、交叉和选择三个步骤，但每个步骤的算子不同、作用不同，而且步骤间的顺序也不同。

与其它进化算法相比，差分进化算法具有以下优点：

- 算法在求解非凸、多峰、非线性函数优化问题上表现出极强的鲁棒性；
- 在同样的精度要求下，算法的收敛速度更快；
- 算法尤其擅长求解高维的函数优化问题；
- 算法非常简单，方便实现。

同时算法仍然存在缺点——算法的关键操作“差分”基于群体内部的差异信息修正个体的值，随着进化次数的增加群体内部个体逐渐趋同，差异化信息减少，从而导致后期收敛速度变慢。^a

^a差分进化算法及其应用 - 道客巴巴.<https://www.doc88.com/p-4149739781802.html?r=1>

1. 类似于遗传算法，记种群规模为 N 、解的维度为 D ，首先进行随机编码生成初始种群 $\{X_1(0), \dots, X_N(0)\}$ 。一般要求 $N \geq 4$ ，且介于 $5D$ 与 $10D$ 之间。对于大多数工程问题，种群规模介于 30-50 之间即可基本满足要求。因为差分进化算法一般用于优化连续变量，因此常采用实数编码进行初始化，即解向量中的元素 $x_{ij}(0)$ 为实数；
2. 不同于遗传算法，差分进化算法基于变异生成新种群。其基本思路为从当前种群中随机抽取多个个体计算差分向量，并将差分向量加权后叠加至其它个体上得到新个体。记变异向量为 $V_i(g)$ ，常用的变异算子包括以下几种：

$$\text{DE/rand/1} \quad V_i(g) = X_{r1}(g) + F \cdot (X_{r2}(g) - X_{r3}(g));$$

$$\text{DE/best/1} \quad V_i(g) = X_{best}(g) + F \cdot (X_{r1}(g) - X_{r2}(g));$$

$$\text{DE/rand/2} \quad V_i(g) = X_{r1}(g) + F \cdot (X_{r2}(g) - X_{r3}(g)) + F \cdot (X_{r4}(g) - X_{r5}(g));$$

$$\text{DE/best/2} \quad V_i(g) = X_{best}(g) + F \cdot (X_{r1}(g) - X_{r2}(g)) + F \cdot (X_{r3}(g) - X_{r4}(g));$$

$$\text{DE/rand-to-best/1} \quad V_i(g) = X_{r1}(G) + F \cdot (X_{best}(g) - X_{r1}(g)) + F \cdot (X_{r2}(g) - X_{r3}(g));$$

$$\text{DE/current-to-best/1} \quad V_i(g) = X_i(G) + F \cdot (X_{best}(g) - X_i(g)) + F \cdot (X_{r1}(g) - X_{r2}(g));$$

$$\text{DE/current-to-rand/1} \quad V_i(g) = X_i(G) + F \cdot (X_{r1}(g) - X_i(g)) + F \cdot (X_{r2}(g) - X_{r3}(g)).$$

上式 $r1, r2, r3, r4, r5$ 为随机整数，且 $i \neq r1 \neq r2 \neq r3 \neq r4 \neq r5$ ； $X_{best}(G)$ 为当前种群适应度最高的个体； $F \in [0, 2]$ 为缩放因子 (scaling factor)，控制种群的多样性与收敛性。 F 越小收敛越慢，且小于一定值后将导致算法不收敛；增大 F 有助于跳出局部最优点；但 $F > 1$ 时收敛速度将变慢，一般令 $F = 0.6$ ；

3. 进一步地通过交叉增强变异结果的多样性。在遗传算法中，交叉是指两个父代间部分编码互换形成新的两个个体的过程。而在差分进化算法中，交叉是指按一定交叉概率 (crossover probability) $CR \in [0, 1]$ 将生成的变异向量 $V_i(G)$ 部分替换为原向量 $X_i(g)$ 的过程。记交叉后的个体为 $U_i(G)$ ，有

$$u_{ij}(g) = \begin{cases} v_{ij}(g) & \text{rand}(0, 1) \leq CR \\ x_{ij}(g) & \text{rand}(0, 1) > CR \end{cases}$$

4. 基于原向量 $X_i(g)$ 与交叉向量 $U_i(g)$ 的适应度选择新个体 $X_i(g+1)$ 。选择时采用贪心选择，若 $f(X_i(g)) \leq f(U_i(g))$ 则令 $X_i(g+1) = U_i(g)$ ，否则则仍令 $X_i(g+1) = X_i(g)$ 。

差分进化 vs 遗传算法		
	遗传算法	差分进化
常用编码	二进制编码	实数编码
种群迭代	父代产生新子代	父代自身进化
算法核心	交叉（重组）	变异
变异算子	子代自身变异（增强交叉结果的多样性）	基于多组父代编码进行变异（生成新子代）
交叉算子	基于多组父代编码生成新编码（生成新子代）	随机替换子代编码为父代编码（增强变异结果的多样性）
选择算子	基于适应度进行随机性选择	基于适应度进行确定性选择
鲁棒性	一般	强
收敛速度	一般	快
全局搜索	强	较强

19.5 布谷鸟搜索 (Cuckoo search, CS)

简介

布谷鸟搜索 (Cuckoo Search, CS) 是由剑桥大学 Xin-She Yang 和 Suash Deb 于 2009 年开发的自然启发式算法^a。其思想主要基于两个策略：布谷鸟的巢寄生性和莱维飞行 (Levy flight) 机制。通过随机游走的方式搜索得到一个最优的鸟窝来孵化自己的鸟蛋，这种方法可以达到一种高效的寻优模式。

CS 算法具有参数少、速度快、操作简单、易实现、随机搜索路径优和寻优能力强等优点。CS 算法具有较强的全局搜索能力，还可以同其它算法无缝对接。差分进化算法 (Differential Evolution, DE)、粒子群优化算法 (Particle Swarm Optimization, PSO) 和模拟退火算法 (Simulated Annealing) 等元启发算法均可视为 CS 算法的特殊情况。^{b,c}

^aCuckoo Search via Levy Flights. https://www.cs.tufts.edu/comp/150GA/homeworks/hw3/_reading%20Cuckoo%20search.pdf

^b布谷鸟搜索算法研究综述. <https://wenku.baidu.com/view/7d25285b69dc5022abea00cf.html>

^c“一个例子”入坑“布谷鸟算法(附完整 py 代码). <https://blog.csdn.net/sj2050/article/details/98496868>

19.5.1 算法思想

1. 布谷鸟算法思想源于现实生活中布谷鸟（杜鹃）的巢寄生性，具体表现如下：
 - (a) 布谷鸟不会自己筑巢育儿，而是会寻找多个其它种类鸟类的巢穴，将卵产至其它鸟类的巢穴中。布谷鸟雏鸟破壳后会将其它未孵化的卵推落巢，从而独自享受寄主鸟的照顾；
 - (b) 布谷鸟的策略并非总会成功：所下的卵有一定概率被识别，而那些与寄主鸟卵相似的卵则更可能不被发现。若卵被发现，布谷鸟需要寻找新的替代巢穴；
 - (c) 每一轮产的卵中，最优的卵将会成功孵化长成成鸟，从而开始新一轮寻巢，经过多代寻巢，布谷鸟所产的卵将会与宿主鸟高度匹配，而每一轮寻找的巢穴越多，卵与寄主鸟卵相似的概率也越大。
2. 根据以上现象，抽象得到布谷鸟算法基本思想；
3. 首先，初始化一定大小的随机数组，数组元素的个数 (n) 对应布谷鸟每一轮寻找的巢穴个数，数组元素的值对应所产的卵的特征，这些“卵”为待优化对象²。随后执行如下循环：
 - (a) 计算每一随机数对应的目标函数值，对应卵的匹配程度，保留使得目标值最优的随机数和该目标值，即认为最优的卵将能成功发育为成鸟；
 - (b) 对全体随机数进行更新，对应雏鸟长成后开始新一轮寻巢；
 - (c) 预设一概率 P_a ，新生成的随机数将以该概率被再次更新，对应部分卵被宿主鸟发现；
 - (d) 再一次计算每一随机数对应的目标函数值，循环进行以上步骤。
4. 布谷鸟算法中最重要的参数为概率 P_a ，根据建议取 $P_a = 25\%$ 即可满足大多数情况的需求。

²若待优化对象有 m 个，则随机数组大小为 $n \times m$

19.5.2 随机数更新算法

莱维飞行 (Levy flight)

- 在计算得到每一轮的最优目标值和对应的随机数后，将对全体随机数进行更新，此时更新算法即为莱维飞行；
- 莱维飞行是随机行走 (random walk)³的一种，其每一次更新的步长方向服从均匀分布，而步长大小则服从莱维分布。莱维分布是一类在正区间连续且稳定⁴的概率分布，其概率密度函数 $f(x)$ 与特征函数 $\varphi(t; \mu, c)$ 为：

$$\begin{cases} f(x) = \sqrt{\frac{c}{2\pi}} \frac{e^{-\frac{1}{2}c(x-\mu)}}{(x-\mu)^{\frac{3}{2}}}, & c > 0, \quad x \geq \mu \\ \varphi(t; \mu, c) = e^{i\mu t - \sqrt{-2ct}} \end{cases}$$

式中 c 为缩放系数， μ 为平移参数。莱维分布属于重尾分布 (heavy-tailed distribution) 的一种；



重尾分布 (heavy-tailed distribution) 与轻尾分布 (thin-tailed distribution)

重尾分布与轻尾分布根据概率密度曲线的下降速度进行区分，当下降速度快于负指数分布时，称该分布为轻尾分布，反之称为重尾分布。从数学角度上，随机变量 X 服从重尾分布的定义式如下：

$$\lim_{x \rightarrow \infty} e^{\lambda x} \Pr[X > x] = \infty, \quad \text{for all } \lambda > 0$$

式中 $\Pr[X > x]$ 称为 X 的尾分布函数。重尾分布的特点为：尽管小值集中了大部分概率，但出现大值的概率依然不低，不可忽略。

- 在布谷鸟算法中，莱维飞行用于模拟布谷鸟寻找新一轮巢穴的过程。观察发现，自然中绝大多数动物的觅食方式都与莱维飞行类似——即找到一块区域后细致查找猎物，如果没找到，就换一片区域，此时步长即服从重尾分布；
- Mantegna 方法⁵可以近似实现莱维分布：

$$s = \frac{u}{|v|^{\frac{1}{\beta}}}, \quad u \sim N(0, \sigma^2), \quad v \sim N(0, 1), \quad \sigma = \left[\frac{\Gamma(1+\beta)}{\beta \cdot \Gamma(\frac{1+\beta}{2})} \cdot \frac{\sin(\frac{\pi\beta}{2})}{2^{\frac{\beta-1}{2}}} \right]^{\frac{1}{\beta}}, \quad \beta = 1.5$$

局部随机行走

- 更新的随机数有 P_a 的概率被舍弃，此时需要对其进行重新更新，此时即采用局部随机行走算法，具体更新随机数 x_i 的算法如下，式中 α 为缩放因子、 $\epsilon(p)$ 为阶跃函数、 x_j, x_k 为两个随机抽取自本组随机数中的随机数

$$x_{i,new} = x_i + \alpha \cdot \epsilon(p - P_a)(x_j - x_k)$$

- 相对于莱维飞行的随机性，局部随机行走时已有一定的方向性了，其最大程度地利用已有点的位置信息，新产生的点便是这些已有点的“折中”。

³随机行走：大意是在任意维度的空间里，一个点随机地向任意方向前进随机长度的距离，然后重复这一步骤的过程。

⁴设随机变量 X, Y 服从某概率分布，若随机变量 $X + Y$ 依然服从该概率分布，则称该概率分布稳定。正态分布、柯西分布与莱维分布是仅有的具有明确概率密度函数的稳定概率分布。

⁵Towards the Improvement of Cuckoo Search Algorithm. http://www.mirlabs.org/ijcisim/regular_papers_2014/IJ CISIM_8.pdf

19.6 蚁群算法 (Ant colony optimization, ACO)

简介

- □ ×

蚁群算法于 1992 年意大利学者 Marco Dorigo 等受蚂蚁觅食行为启发而提出。蚂蚁在觅食的过程中会沿路径释放信息素，而后方蚂蚁会倾向于沿着信息素浓度高的路径移动，从而形成一种正反馈机制，使得蚁群整体表现出一些智能行为。算法具有分布计算、信息正反馈和启发式搜索的特征，本质上是进化算法中的一种启发式全局优化算法。算法可应用于组合优化问题的求解，在解决离散组合优化方面具有良好的性能。

最初提出的蚁群算法称为基本蚁群算法，大量研究针对离散域蚁群算法的改进。自适应蚁群算法是一种对蚁群算法的状态转移概率 $P_{ij}^k(t)$ 、信息素挥发因子 ρ 、信息量 Q 等因素采用自适应调节策略的蚁群算法，最经典的两个自适应一群算法为蚁群系统 (Ant colony system, ACS) 和最大-最小蚁群系统 (MAX-MIN ant system, MMAS)。

1. 蚁群算法的基本思想源于蚁群活动过程中的信息素遗留和信息素跟踪两个过程：

- **信息素遗留**：蚂蚁会在走过的路上释放信息素，一定范围内的蚂蚁会感受到信息素的存在并基于感知的信息素进行决策；
- **信息素跟踪**：蚂蚁按照一定的概率沿着信息素浓度较高的路径觅食。

2. 基于上述蚁群活动的机理，蚁群算法的假设如下：

- **觅食规则**：蚂蚁会在一定范围内寻找是否存在食物或信息素，每只蚂蚁都会以小概率犯错；
- **移动规则**：蚂蚁首选向信息素多的方向移动，若无信息素则朝原方向移动，运动过程中会发生随机扰动，蚂蚁也具有一定的记忆性；
- **避让规则**：若存在障碍物，蚂蚁会随机选择任意方向避让；
- **信息素规则**：越靠近食物散布的信息素越多，反之越少。

3. 蚁群算法的基本参数如下：

- m : 表示蚂蚁种群中蚂蚁的数量；
- x, y : 分别表示待优化问题中处理节点和任务的个数⁶；
- $b_i(t)$: 表示 t 时刻位于节点 i 的蚂蚁数量；
- $d_{ij}(t)$: 表示 t 时刻第 j 个任务由第 i 个节点处理所需的代价（距离、时间、费用等等）；
- $\eta_{ij}(t)$: 表示能见度，称为启发信息函数，等于代价的倒数 $\eta_{ij}(t) = \frac{1}{d_{ij}(t)}$ ；
- $\tau_{ij}(t)$: 表示 t 时刻“第 j 个任务由第 i 个节点处理”这一路径上的信息素水平，初始时各路径信息素水平相等。

4. 蚁群算法的基本流程如下：

- (a) 随机初始化所有蚂蚁的初始节点；
- (b) 按一定的规则更新每一只蚂蚁的下一个节点， t 时刻第 k 只蚂蚁于任务 j 选择节点 i 的概率记为 $P_{ij}^k(t)$ ，一般来说 $P_{ij}^k(t)$ 同时与先验条件（即代价）和似然（即已有信息素水平）有关

$$P_{ij}^k(t) = \begin{cases} \frac{|\tau_{ij}(t)|^\alpha |\eta_{ij}(t)|^\beta}{\sum_i |\tau_{ij}(t)|^\alpha |\eta_{ij}(t)|}, & i \in \text{allowed}_k(t) \\ 0, & \text{else} \end{cases}$$

- $\text{allowed}_k(t)$: t 时刻蚂蚁 k 所能选择的下一个节点的集合；
- α : 信息启发式因子，反映蚁群在路径搜索中随机性因素作用的强度；
- β : 期望启发式因子，反映蚁群在路径搜索中先验性、确定性因素作用的强度。

为避免过早陷入局部最优解，除了调整上述参数外，还可以仅设置蚁群中的部分蚂蚁按以上规则移动，另一部分蚂蚁完全随机行走；

⁶在旅行商问题中，因为要求遍历所有城市且每个城市只去一次，此时节点数和任务数相等，为城市总数；而在流水车间调度问题中，节点（设备）与任务（工序）的个数往往不相等。

- (c) 蚂蚁位置更新后，会在已有路径上留下新的信息素 $\Delta\tau_{ij}(t)$ ，而原有的信息素也会随时间挥发，记信息素挥发因子为 ρ ，则残留因子为 $1 - \rho$ ，则信息素更新规则如下

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad \Delta\tau_{ij}(t) = \sum_k \Delta\tau_{ij}^k(t)$$

Marco Dorigo 给出 $\Delta\tau_{ij}^k(t)$ 的三种模型：

- 蚂蚁圈系统 (**Ant-cycle system**)：认为蚂蚁在完成一次遍历的过程中信息素的总量相等，记为 Q ，留下的信息素水平按该遍历的总长度 L_k 分配。该方法利用全局信息，蚂蚁完成一个循环后进行信息素更新，保证信息素不至于无限累积，效果较好；
- 蚂蚁数量系统 (**Ant-quantity system**)：认为蚂蚁在从一个节点至另一个节点的过程中信息素的总量相等，记为 Q ，留下的信息素水平按该次转移的长度 d_k 分配。该方法利用局部信息，蚂蚁每完成一次移动进行信息素更新；
- 蚂蚁密度系统 (**Ant-density system**)：认为蚂蚁在移动过程中信息素的密度相等，记为 Q 。该方法利用局部信息，蚂蚁每完成一次移动进行信息素更新。

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{Ant - cycle system} \\ \frac{Q}{d_k}, & \text{Ant - quantity system} \\ Q, & \text{Ant - density system} \end{cases}$$

在每一只蚂蚁完成遍历后都更新信息素的更新规则称为局部更新规则，蚁群系统采用全局更新规则，即等所有蚂蚁完成遍历后对效果最好的蚂蚁进行更新；

- (d) 全部蚂蚁遍历完所有任务后即完成一次循环。重复以上步骤直至达到最大循环次数或目标值收敛。

参数	取值范围	影响分析
m		m 越大得到的最优解就越精确，但会产生更多的重复解，从而随着收敛正反馈作用减弱，增加计算量。
α	[0, 5]	α 越大蚁群越倾向于选择之前走过的路径，搜索路径的随机性会降低；反之则蚂蚁越倾向于局部短路径，即贪婪搜索，降低搜索范围，容易陷入局部最优。
β	[0, 5]	β 越大蚁群越容易选择局部最短路径，可提高收敛速度但随机性不高易陷入局部最优。
ρ	[0.1, 0.99]	ρ 过小时无效路径依然会被重复搜索，降低收敛速度；反之则会使得无法判断是否有效的路径过早被放弃搜索，陷入局部最优。
Q	[10, 10000]	

19.7 快速扩展随机树算法 (*Rapidly-exploring random tree, RRT*)

图 19.2 快速扩展随机树算法示意。从给定点出发，算法通过随机采样构造随机数快速探索未知空间，适用于高维非凸空间和复杂障碍物约束下从起始点到目标点的路径规划问题。

- 【学界】快速扩展随机树路径规划算法：https://zhuanlan.zhihu.com/p/47571333?utm_id=0
- 基于采样的运动规划算法-RRT(Rapidly-exploring Random Trees)：https://zhuanlan.zhihu.com/p/133224593?utm_id=0
- 自动驾驶路径规划——基于概率采样的路径规划算法(RRT、RRT*)：https://blog.csdn.net/sinat_52032317/article/details/127197120

19.7.1 基础 RRT 算法 (basic RRT)

1. 快速扩展随机树算法 (rapidly-exploring random tree, RRT) 是 Steven M. LaValle 于上世纪末提出的一种简单且高效的快速探索给定地图空间的算法。算法从给定点出发，通过状态空间的随机采样点，把搜索导向空白区域，构建在空间内快速搜索的树结构，从而寻找到一条从起始点到目标点的规划路径，特别适用于高维非凸空间和复杂障碍物约束下的运动路径规划问题，被广泛应用于各种机器人运动规划；
2. 首先介绍基础 RRT 算法 (basic RRT)。记 x_{init}, x_{goal} 分别表示起点和终点， M 表示带障碍约束的空间， T 为从 x_{init} 出发随机探索构建树结构。算法的思想非常简单。每一步迭代时在 M 空间内随机取一个点 x_{rand} ，再从 T 中选择与 x_{rand} 最近的点 x_{near} ，沿 (x_{near}, x_{rand}) 方向以预设的 Δ 为步长取得下一个点 x_{new} ，对应连边 $e_{new} = (x_{near}, x_{new})$ 。若连边 e_{new} 不与障碍物冲突，则将 x_{new}, e_{new} 加入树结构 T ，反之则进入下一轮迭代重新采样 x_{rand} 。算法的收敛条件为 x_{new} 达到或者接近目标点 x_{goal} ；
3. 进一步分析 RRT 算法的设计原理，重点在于随机采样 x_{rand} 的效果：
 - 若 x_{rand} 位于 x_{goal} 附近，则生成的新连边 (x_{near}, x_{new}) 可使得路径接近 x_{goal} ；
 - 若 x_{rand} 距离 x_{goal} 较远，则生成的新连边 (x_{near}, x_{new}) 也可增加从其它方向接近 x_{goal} 的选择。

可见简单的随机采样即可使得随机采样点“拉着”树向外生长，引导树结构快速高效地探索未知空间。理论推导结果进一步指出，只要到 x_{goal} 的路径切实存在，则存在一个与障碍空间 M 有关的超参 a, n_0 ，使得

$$P(\text{a path is found}) \geq 1 - \exp\{-a \cdot n\}, \quad n \geq n_0$$

式中 n 表示采样数。上式说明只要采样数够多，则 RRT 算法成功搜索得路径的概率趋近于 1，这一特性被称为概率完备性 (probabilistic completeness)。RRT 算法因其简单高效的特点吸引了大量关注。然而 RRT 算法的缺点也无法避免：

- 随机采样过程对环境不敏感，当空间中包含大量障碍物或狭窄通道约束时算法效率将大幅下降；
- 在探索后期大量随机生成的样本无助于接近目标，反而增加了算法的时间和空间损耗；
- 算法无法得到最优路径，也不会收敛至近似最优解。

针对基础 RRT 算法的缺陷衍生出了众多变体。优化的目标大体分为两类——加速算法的探索效率与提升规划路线的质量。

19.7.2 面向高效采样的改进 RRT 算法

1. 为提升 RRT 的采样效率以更高效地探索空间，最直观的思路便是提升采样的方向性。基于目标概率采样的 RRT 算法（也称为 Goal-bias RRT 算法）因其简单、直观的优势成为 RRT 的一种重要变体。算法针对基础 RRT 在探索后期计算效率过低的问题将完全随机采样调整为有偏随机采样使得采样过程具有更强的方向性（类似图随机游走模型（第 25.7 节）中 Node2Vec 对 DeepWalk 的优化）。具体地模型令 x_{rand} 生成时以 $p \in [0, 1]$ 的概率直接取 x_{goal} 。超参 p 的取值决定了算法的效果—— p 越大则算法的方向性越强但探索性也越弱。特别是当 x_{goal} 周围存在障碍时，较大的 p 易使算法陷入局部收敛而无法跳出，造成路径规划失败。为保证算法的探索能力 p 一般取值较小（常设为 0.05-0.3）；
2. APF-RRT 算法是指将人工势场法 (artificial potential field, APF) 与 RRT 算法结合的一系列算法，可认为是对 Goal-bias RRT 算法的进一步改进。人工势场法是一种基于人工构建的场函数的路径规划方法。人工场由引力场和斥力场组成。引力场提供目标位置信息，引导路径向目标方向发展；而斥力场提供障碍物信息，引导路径绕开障碍物。例如可以仅引入引力场，使随机树的生长方向由目标节点 x_{goal} 和随机采样结果 x_{rand} 共同决定

$$x_{new} = x_{near} + \Delta \left(\frac{x_{rand} - x_{near}}{\|x_{rand} - x_{near}\|} + k_p \frac{x_{goal} - x_{near}}{\|x_{goal} - x_{near}\|} \right)$$

k_p 为算法超参； $k_p \frac{x_{goal} - x_{near}}{\|x_{goal} - x_{near}\|}$ 即为引力场函数；

3. 除了优化随机采样过程，另一种常见的提升探索方向性的改进方法是从起点 x_{init} 和终点 x_{goal} 同时构建随机树对空间作双向搜索，此类算法被称为双向 RRT(Bi-RRT) 或连接 RRT(RRT-connect)。需要注意的是，Bi-RRT 算法中两棵随机树并非均随机生长，而是分为主随机树 T_1 和副随机树 T_2 ：

- 一般以从 x_{init} 生长的随机数为主随机树 T_1 , 从 x_{goal} 生长的随机数为副随机树 T_2 ;
- T_1 以探索空间为目标, 生长过程与基础 RRT 算法一致, 也可参考其它变体的改进; T_2 的生长则更具方向性, 以向 T_1 生长为目标。具体地, 在 T_1 探索得到新节点 x_{new}^1 后, T_2 不作采样而直接以其为随机采样节点 $x_{rand}^2 \leftarrow x_{new}^1$ 从而构建 x_{new}^2 ;
- 因为障碍的约束无法保证每次迭代时 T_1, T_2 均可顺利生长, 经若干次迭代后两棵随机树间可能存在严重的不平衡。为平衡两棵树的生长需要在每次迭代后比较两棵树的规模 (如节点数、路径总长等等), 交替选择规模更小的树作为主随机树 T_1 ;
- 当 T_1, T_2 间的最小距离小于预设阈值后即可认为两棵树找到彼此, 直接连接 T_1, T_2 得到规划路径。

19.7.3 面向路径优化的改进 RRT 算法

1. 此类算法的代表是 2010 年提出的 RRT* 算法。与基础 RRT 算法相比, RRT* 算法可显著提升轨迹规划的质量且随迭代进行将收敛至最优解, 但算法的计算复杂度也显著更高。算法与基础 RRT 采用相同的样本点采样过程以保证探索性, 而为提升轨迹质量重点关注快速扩展随机树 T 的构建过程:

- 父节点确定: 在 x_{near} 确定并生成 x_{new} 后, 基础 RRT 算法直接以 x_{near} 为 x_{new} 的父节点将连边 (x_{near}, x_{new}) 加入 T 中, 而 RRT* 算法则需重新确定 x_{new} 的父节点。具体地, 算法以 x_{new} 为中心, 超参 r 为半径寻找 T 中包括 x_{near} 在内的全部邻域节点作为 x_{new} 的候选父节点集合 $\{x_p^i | i = 1, \dots, n\}$ 。则 x_{new} 的最优父节点 x_p^* 按下式确定

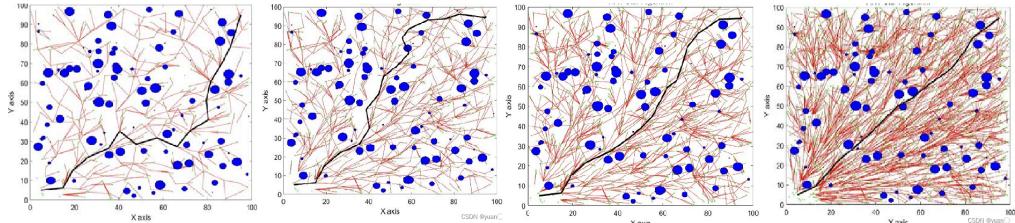
$$x_p^* = \arg \min_i \{f(x_{init} \rightarrow x_p^i \rightarrow x_{new})\}$$

式中 $x_{init} \rightarrow x_p^i \rightarrow x_{new}$ 表示从起点 x_{init} 经 x_p^i 至 x_{new} 的最短路 (假设 x_p^i 与 x_{new} 连接); $f(\cdot)$ 表示路径的总成本函数, 由每一路段的成本加和得到。因此上式实际上即是寻找从 x_{init} 至 x_{new} 的最短路从而确定 x_{new} 于随机树结构 T 中的父节点;

- 随机树重布线: 插入 x_{new} 后算法并不直接进入下一步迭代, 而是对 T 重布线以减少冗余通路。具体地仍关注 x_{new} 的邻域节点集合 $\{x_p^i | i = 1, \dots, n\}$ 。对 $\forall x_p^i$ 在 x_{new} 插入后再考虑 $x_{init} \rightarrow x_p^i \rightarrow x_{new}$ 和 $x_{init} \rightarrow x_{new} \rightarrow x_p^i$ 两条通路。若 $f(x_{init} \rightarrow x_p^i \rightarrow x_{new}) > f(x_{init} \rightarrow x_{new} \rightarrow x_p^i)$, 则断掉 x_p^i 与原父节点的连接而以 x_{new} 为 x_p^i 的父节点; 反之则不作调整。

综上所述, RRT* 算法实际上即是在随机树 T 生长的过程中不断基于最短路算法优化 T 的内部结构, 从而以牺牲计算效率为代价获得持续优化轨迹质量的能力;

图 19.3 与基础 RRT 相比, RRT* 可随迭代收敛至最优解, 从而得到高质量的规划路径。



第 20 章

矩阵分解与经典降维

在机器学习中经典的降维算法一般服务于以下三类目的：冗余特征去除、有效特征提取与可视化。前两个目的均属特征工程，一般为建模分析的前期准备阶段。注意到特征工程完成后的预测阶段有大量可捕捉非线性特征的非线性模型可供选择，故大多数应用于特征工程的经典降维算法属于线性算法，仅消除数据中的线性相关特征以避免后续建模时发生维数灾难。因此，此类经典线性降维算法大多可建模为矩阵分解问题。矩阵分解是一类经典的线性代数问题，除降维之外，矩阵分解还有另外两类主要用途——稀疏数据填补与隐形关系挖掘。

另有一些经典降维算法为非线性模型，在降维时即可捕捉数据的非线性特征，因此降维效果往往更好，故多应用于可视化（要求降维至 3 维或以下）。但此类模型一般不用于特征工程。因为非线性算法鲁棒性普遍较差，且过于稠密的非线性数据也会加大后续的非线性预测模型的学习难度。

为实现降维相关的各项目的，降维模型一般从以下两类思路出发设计：**1)** 以数据特征重构为目标，代表性模型为主成分分析（第 15.2.1 节）、非负矩阵分解（第 20.2 节）和自编码器（第 24.9 节）等等；**2)** 以最小化降维数据分布与原数据分布误差为目标，代表性模型为 T-SNE 算法（第 20.9 节）。后者一般可更好地提取数据的分布结构信息，实现更理想的降维效果，但因为技术路线往往仅考虑了从原始数据至降维数据的单向映射，因此相关模型只能用于降维。而基于第一类思路设计的降维模型为重构数据特征往往包含原始数据与降维数据间的双向映射，故模型不仅可用于降维，理论上还可用于降噪和数据生成。

20.1 主成分分析 (*Primary component analysis, PCA*) 与因子分析 (*Factor analysis*)

主成分分析与因子分析的相关概念及理论分别详见第 15.2.1 节与第 15.2.2 节。

20.2 非负矩阵分解 (*Non-negative matrix factorization, NMF*)

20.2.1 问题建模

- 由主成分分析算法和因子分析算法的基本概念可知，两者本质上均是将 $n \times m$ 维数据矩阵 V 分解为 $n \times d$ 维压缩矩阵 W 和 $d \times m$ 维系数矩阵 H 实现降维

$$V_{n \times m} \approx W_{n \times d} H_{d \times m}, \quad d < m$$

其中 W 即为降维后的矩阵，其各列表示基于原数据矩阵 V 各列特征提取的基本特征，而 H 的列向量即为将 W 中的基本特征还原为原数据矩阵 V 的线性组合系数；

- 求解 W 和 H 的不同思路即对应不同的降维算法。主成分分析在正交约束上以最大化表征原数据矩阵 V 的方差为目标，故降维后的矩阵 W 可较好地避免信息损失，缺点则是解释性不足。因子分析则以最大化线性组合系数方差为目标，以使得系数矩阵 H 尽可能稀疏，从而令矩阵 W 表征的基本特征具有更清晰的统计解释性；

3. 非负矩阵分解 (Non-negative Matrix Factorization) 于 1999 年由 Lee 和 Seung 发表于《Nature》上¹, 其目标同样是求解 W, H , 但不同于主成分分析与因子分析, 算法引入非负约束后可使得降维后的矩阵 W 在特定场景下具有更低的噪声与更高的物理意义。以图像处理为例, 原始的像素矩阵 V 显然为非负, 降维后矩阵 W 同样也应为非负, 但主成分分析等算法并未考虑相关情况, 可能使得 W, H 为负的情况, 既引入噪声, 也使得结果无实际意义。而 NMF 算法即针对 V 非负的情况, 要求分解后矩阵 W, H 同样非负, 于此类场景下更为适用;

4. NMF 算法的基本思路是在 W, H 非负的前提下最小化 V 与 WH 间的误差, 故直观地可建模为

$$W, H = \arg \min_{W, H} \|V - WH\|_F^2, \quad \text{s.t. } W, H \geq 0$$

式中 $\|\cdot\|_F$ 表示矩阵的 Frobenius 范数 (简称 F 范数), 类比向量的 L2 范数, 定义为矩阵各元素平方和的开方。构建所示目标函数意味着以欧式距离量化误差, 其本质是假设样本 V_{ij} 服从期望为 $(WH)_{ij}$ 的高斯分布, 则易得似然函数 $L(W, H)$ 有

$$L(W, H) = \prod_{ij} \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(V_{ij} - (WH)_{ij})^2}{2\sigma^2} \right\} \implies \ln L(W, H) = \sum_{ij} \ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{ij} (V_{ij} - (WH)_{ij})^2$$

显然, 欲使得对数似然函数 $\ln L(W, H)$ 最大, 则只需令 $\sum_{ij} (V_{ij} - (WH)_{ij})^2 = \|V - WH\|_F^2$ 取最小, 即为上述 NMF 算法的目标函数。注意到上式暗含了同方差假设, 即基于 WH 估计 V 时每一元素的误差服从相同的方差 σ 。而对于异方差的情况, 则一般构建权重矩阵 $D = [1/\sigma_{ij}^2]$, 带权的 NMF 问题建模为

$$W, H = \arg \min_{W, H} \|D^{\frac{1}{2}} \odot (V - WH)\|_F^2, \quad \text{s.t. } W, H \geq 0$$

式中 \odot 表示矩阵的 Hadamard 积, 为矩阵逐元素相乘。以欧式距离量化误差虽然直观, 相应的 NMF 算法也最为常用, 但其背后的高斯分布假设却并非完全合理。因为在高斯分布中负值同样具有正概率密度, 与 NMF 算法要求的非负场景不符;

5. 这意味着, NMF 算法建模时也可以其它距离度量方法量化误差, 意味着假设观测样本 V_{ij} 服从其它特定分布, 从而使算法适应不同的实际场景。例如当 V 各元素恒为整数时 (如图像处理等), 则常假设 V_{ij} 服从期望为 $(WH)_{ij}$ 的泊松分布, 对应的距离度量为 KL 散度

$$P(\tilde{V}_{ij} = V_{ij}) = \frac{(WH)_{ij}^{V_{ij}}}{(V_{ij})!} \exp\{-(WH)_{ij}\}$$

式中 \tilde{V}_{ij} 表示 V_{ij} 对应的随机变量。构造似然函数 $L(W, H)$ 有

$$L(W, H) = \prod_{ij} \frac{(WH)_{ij}^{V_{ij}}}{(V_{ij})!} \exp\{-(WH)_{ij}\} \implies \ln L(W, H) = \sum_{ij} \{V_{ij} \ln(WH)_{ij} - (WH)_{ij} - \ln(V_{ij}!)\}$$

由 Stirling 近似公式 $\ln(x!) \approx x \ln x - x$, 有

$$\ln L(W, H) = \sum_{ij} \{V_{ij} \ln(WH)_{ij} - (WH)_{ij} - V_{ij} \ln V_{ij} + V_{ij}\} = \sum_{ij} \left\{ V_{ij} \ln \frac{(WH)_{ij}}{V_{ij}} - (WH)_{ij} + V_{ij} \right\}$$

显然, 欲使得对数似然函数 $\ln L(W, H)$ 最大, 则只需将 NMF 问题构造为

$$W, H = \arg \min_{W, H} \sum_{ij} \left\{ V_{ij} \ln \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right\}, \quad \text{s.t. } W, H \geq 0$$

当 $\sum_{ij} V_{ij} = \sum_{ij} (WH)_{ij} = 1$ 时 V, WH 即可视为概率分布, 目标函数即为经典的 KL 散度 (见第 23.9.6 节)。令 E 为与 V 同尺寸且各元素均为 1 的矩阵, 则上式也可写为矩阵形式, 并去掉与 W, H 无关项, 有

$$W, H = \arg \min_{W, H} \text{tr} \{E^\top WH - V^\top \ln(WH)\}, \quad \text{s.t. } W, H \geq 0$$

¹Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788–791. doi:10.1038/44565

20.2.2 基于拉格朗日乘数法优化——元素形式

1. 进一步地, 以高斯分布假设为例, 介绍 NMF 算法中矩阵 W, H 的求解。显然对于带约束最优化问题, 可基于拉格朗日乘数法求解。为提升可读性, 先以矩阵元素形式建模松弛约束后的问题为²

$$W, H = \arg \min_{W, H} \mathcal{L} = \sum_{ij} \left(V_{ij} - \sum_k W_{ik} H_{kj} \right)^2 - \sum_{ik} \lambda_{ik} W_{ik} - \sum_{kj} \nu_{kj} H_{kj}$$

式中 λ_{ik}, ν_{kj} 分别为决策变量 W_{ik}, H_{kj} 的拉格朗日乘子。基于 KKT 条件 (见第 17.3.1 节), 有

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial W_{ik}} = 0 \\ \frac{\partial \mathcal{L}}{\partial H_{kj}} = 0 \end{cases} \Rightarrow \begin{cases} -\lambda_{ik} - 2 \sum_j \left(V_{ij} - \sum_k W_{ik} H_{kj} \right) H_{kj} = 0 \\ -\nu_{kj} - 2 \sum_i \left(V_{ij} - \sum_k W_{ik} H_{kj} \right) W_{ik} = 0 \end{cases} \Rightarrow \begin{cases} -\lambda_{ik} W_{ik} - 2 \sum_j \left(V_{ij} - \sum_k W_{ik} H_{kj} \right) H_{kj} W_{ik} = 0 \\ -\nu_{kj} H_{kj} - 2 \sum_i \left(V_{ij} - \sum_k W_{ik} H_{kj} \right) W_{ik} H_{kj} = 0 \end{cases}$$

进一步代入 KKT 条件中的 $\lambda_{ik} W_{ik} = \nu_{kj} H_{kj} = 0$, 则上两式中可消去拉格朗日乘子 λ_{ik}, ν_{kj}

$$\sum_j \left(V_{ij} - \sum_k W_{ik} H_{kj} \right) H_{kj} W_{ik} = 0, \quad \sum_i \left(V_{ij} - \sum_k W_{ik} H_{kj} \right) H_{kj} = 0$$

2. 简化上述矩阵元素形式。首先注意到 $V_{ij} - \sum_k W_{ik} H_{kj}$ 为矩阵 $V - WH$ 的第 i 行第 j 列元素, 则上式变为

$$\sum_j (V - WH)_{ij} (H^T)_{jk} W_{ik} = 0, \quad \sum_i (W^T)_{ki} (V - WH)_{ij} H_{kj} = 0$$

进一步注意到 $\sum_j (V - WH)_{ij} (H^T)_{jk}$ 为矩阵 $(V - WH)H^T$ 的第 i 行第 k 列元素, $\sum_i (W^T)_{ki} (V - WH)_{ij}$ 为矩阵 $W^T(V - WH)$ 的第 k 行第 j 列元素, 则上式进一步简化为

$$\begin{cases} [(V - WH)H^T]_{ik} W_{ik} = 0 \\ [W^T(V - WH)]_{kj} H_{kj} = 0 \end{cases} \Rightarrow \begin{cases} (VH^T)_{ik} W_{ik} = (WHH^T)_{ik} W_{ik} \\ (W^TV)_{kj} H_{kj} = (W^TWH)_{kj} H_{kj} \end{cases}$$

3. 求解上述方程组中的 W_{ik}, H_{kj} 即可得到 NMF 的最优解。但注意到上述方程组中 W_{ik}, H_{kj} 相互嵌套, 为求解 W_{ik} 需预知 H_{kj} , 反之亦然, 为此 NMF 算法一般基于迭代更新方法计算收敛解, 即基于第 t 轮迭代时的 W, H 更新第 $t + 1$ 轮的结果

$$W_{ik} \leftarrow \frac{(VH^T)_{ik} W_{ik}}{(WHH^T)_{ik}}, \quad H_{kj} \leftarrow \frac{(W^TV)_{kj} H_{kj}}{(W^TWH)_{kj}} \Rightarrow W \leftarrow W \odot \frac{VH^T}{WHH^T}, \quad H \leftarrow H \odot \frac{W^TV}{W^TWH}$$

若考虑权重矩阵 D 的情况, 则上式可进一步改写为

$$W \leftarrow W \odot \frac{(D \odot V)H^T}{[D \odot (WH)]H^T}, \quad H \leftarrow H \odot \frac{W^T(D \odot V)}{W^T[D \odot (WH)]}$$

4. 上式即为高斯分布假设时经典 NMF 算法的乘法迭代公式, 而当采用泊松分布假设时, 算法的迭代更新公式如下, 其中 E 为与矩阵 V 尺寸相同, 元素均为 1 的矩阵

$$\begin{cases} W_{ik} \leftarrow W_{ik} \frac{\sum_j H_{kj} V_{ij} / (WH)_{ij}}{\sum_j H_{kj}} \\ H_{kj} \leftarrow H_{kj} \frac{\sum_i W_{ik} V_{ij} / (WH)_{ij}}{\sum_i W_{ik}} \end{cases} \Rightarrow \begin{cases} W \leftarrow W \odot \frac{[V/(WH)]H^T}{EH^T} \\ H \leftarrow H \odot \frac{W^T[V/(WH)]}{W^TE} \end{cases}$$

²非负矩阵分解 (NMF) 迭代公式推导证明: <https://zhuanlan.zhihu.com/p/340774022>

20.2.3 基于拉格朗日乘数法优化——矩阵形式

1. 同样以高斯分布假设为例，以矩阵形式改写上述拉格朗日函数 \mathcal{L} 使数学形式更为简洁。首先改写 $\|V - WH\|_F^2$ 。注意到 $\|A\|_F^2 = \sum_{ij} A_{ij}^2$ ，且 $\text{tr}(A^T B) = \sum_{ij} A_{ij} B_{ij}$ ，则 $\|A\|_F^2 = \text{tr}(A^T A)$ ，因此 \mathcal{L} 改写为

$$W, H = \arg \min_{W, H} \mathcal{L} = \text{tr}((V - WH)^T(V - WH)) - \text{tr}(\Lambda^T W) - \text{tr}(\Phi^T H)$$

上式中矩阵 Λ, Φ 的各元素分别对应 W, H 各元素的拉格朗日乘子。同样地需要基于 KKT 条件计算 $\frac{\partial \mathcal{L}}{\partial W} = 0, \frac{\partial \mathcal{L}}{\partial H} = 0$ 以求解 W, H ，其中 \mathcal{L} 为标量， W, H 为矩阵，即需要计算标量对矩阵的导数；

2. 按第 23.4 节所介绍的矩阵求导方法推导 $\frac{\partial \mathcal{L}}{\partial W}, \frac{\partial \mathcal{L}}{\partial H}$ ，首先计算全微分

$$\begin{aligned} d\mathcal{L} &= \text{tr}(-(dWH + WdH)^T(V - WH) - (V - WH)^T(dWH + WdH)) - \text{tr}(\Lambda^T dW) - \text{tr}(\Phi^T dH) \\ &= \text{tr}(-2(V - WH)^T(dWH + WdH)) - \text{tr}(\Lambda^T dW) - \text{tr}(\Phi^T dH) \\ &= \text{tr}((-2H(V - WH)^T - \Lambda^T) dW) + \text{tr}((-2(V - WH)^T W - \Phi^T) dH) \end{aligned}$$

进而可得

$$\begin{aligned} \begin{cases} \frac{\partial \mathcal{L}}{\partial W} = (-2H(V - WH)^T - \Lambda^T)^T = -2(V - WH)H^T - \Lambda \\ \frac{\partial \mathcal{L}}{\partial H} = (-2(V - WH)^T W - \Phi^T)^T = -2W^T(V - WH) - \Phi \end{cases} \\ \Rightarrow \begin{cases} \frac{\partial \mathcal{L}}{\partial W} = 0 \\ \frac{\partial \mathcal{L}}{\partial H} = 0 \end{cases} \rightarrow \begin{cases} 2(V - WH)H^T + \Lambda = 0 \\ 2W^T(V - WH) + \Phi = 0 \end{cases} \rightarrow \begin{cases} 2(V - WH)H^T \odot W + \Lambda \odot W = 0 \\ 2W^T(V - WH) \odot H + \Phi \odot H = 0 \end{cases} \end{aligned}$$

为消去拉格朗日乘子矩阵 Λ, Φ ，同样地基于 KKT 条件中的 $\lambda_{ik} W_{ik} = \nu_{kj} H_{kj} = 0$ ，写为矩阵形式即为 $\Lambda \odot W = 0, \Phi \odot H = 0$ ，则同样可由上式导出 NMF 算法中 W, H 的更新公式

$$\begin{cases} (V - WH)H^T \odot W = 0 \\ W^T(V - WH) \odot H = 0 \end{cases} \Rightarrow \begin{cases} [(V - WH)H^T]_{ik} W_{ik} = 0 \\ [W^T(V - WH)]_{kj} H_{kj} = 0 \end{cases}$$

20.2.4 基于梯度下降优化

1. 因为 NMF 算法建模为带约束的最优化问题，故拉格朗日乘数法是最经典且合理的求解算法。然而上文基于拉格朗日乘数法推导时并未能回答为何构造相应形式的迭代更新公式可使得目标函数下降；
2. 为此本文基于梯度下降法重新推导 NMF 问题更新公式。梯度下降法不仅可巧妙地得到 NMF 算法的 W, H 迭代更新公式，也可在一定程度上理解更新公式的机制。同样以高斯分布假设下的 NMF 优化问题为例，暂不考虑约束 $W, H \geq 0$ ，直接对目标函数 $z = \|V - WH\|_F^2$ 求偏导，由上文有

$$\begin{cases} \frac{\partial z}{\partial W} = -2(V - WH)H^T \\ \frac{\partial z}{\partial H} = -2W^T(V - WH) \end{cases} \Rightarrow \begin{cases} \frac{\partial z}{\partial W_{ik}} = -2[(V - WH)H^T]_{ik} \\ \frac{\partial z}{\partial H_{kj}} = -2[W^T(V - WH)]_{kj} \end{cases}$$

3. 若直接按梯度下降优化，则只需将 $\frac{\partial z}{\partial W_{ik}}, \frac{\partial z}{\partial H_{kj}}$ 代入下式更新

$$W_{ik} \leftarrow W_{ik} - \mu_{ik} \frac{\partial z}{\partial W_{ik}}, \quad H_{kj} \leftarrow H_{kj} - \eta_{kj} \frac{\partial z}{\partial H_{kj}}$$

然而通用的梯度下降更新算法并未考虑到 W, H 的非负约束，其根本原因在于加法更新时无法保证参数不变号，为此将加法更新改为乘法更新，只需令 $\mu_{ik} = \frac{W_{ik}}{2[WHH^T]_{ik}}, \eta_{kj} = \frac{H_{kj}}{2[W^TWH]_{kj}}$ ，则更新公式变为

$$W_{ik} \leftarrow \frac{(VH^T)_{ik} W_{ik}}{(WHH^T)_{ik}}, \quad H_{kj} \leftarrow \frac{(W^TV)_{kj} H_{kj}}{(W^TWH)_{kj}}$$

上式即为 **NMF** 算法更新公式，其本质上即是考虑了非负约束的梯度下降更新，基于乘法更新即可保证更新过程中 W, H 不变号，从而满足非负约束。

20.2.5 基于辅助函数优化与收敛性证明

1. 上文分别基于拉格朗日乘数法和最小二乘法构造得到了 NMF 算法的迭代更新公式。然而，在推导过程中回避了一个关键问题，即如何保证构造的更新公式可使得 W, H 收敛于局部最优解。本节将主要以高斯分布假设下的 NMF 优化问题为例进行讨论³；
2. 首先定义辅助函数 (**auxiliary function**)。对于任意函数 $F(h)$ ，若存在函数 $G(h, h')$ 满足 $G(h, h') \geq F(h)$, $G(h, h) = F(h)$ ，则称 $G(h, h')$ 为 $F(h)$ 的辅助函数，其实际上是为 $F(h)$ 提供了上界。基于 $G(h, h')$ 实际上给出了一种确保 $F(h)$ 非增更新的方法

$$F(h^{t+1}) \leq F(h^t), \quad h^{t+1} = \arg \min_h G(h, h^t)$$

式中 h^t, h^{t+1} 分别表示用于第 t 轮和第 $t+1$ 轮迭代的参数取值。上式可由定义易证： $F(h^{t+1}) \leq G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h^t)$ 。当且仅当 $h^{t+1} = h^t$ 时上述更新过程收敛，此时 h^{t+1} 为 $G(h, h^t)$ 的局部极小点，也意味着 h^{t+1} 为 $F(h)$ 的局部极小点。故基于辅助函数的概念，只需交替地构造 $G(h, h^t)$ 并更新 $h^t, F(h^t)$ ，即可保证收敛至 $F(h)$ 的局部极小点；

3. 进一步地，即是要针对 **NMF** 的目标函数构造其辅助函数。对于目标函数 $z = \|V - WH\|_F^2$ ，暂令 W 为参数，以 H 为自变量将其改写为列向量的形式

$$z = \sum_j F(H_j) = \sum_j \|V_j - WH_j\|_2^2 = \sum_j (V_j - WH_j)^T (V_j - WH_j)$$

式中 V_j, H_j 分别表示矩阵 V, H 的第 j 个列向量。可以看到目标函数 z 可分解为若干 $F(H_j)$ 之和，且 $F(H_j)$ 之间互相独立，故最小化 z 实际上即是对每一 $F(H_j)$ 求最小。故定义

$$F(h) = (v - Wh)^T (v - Wh)$$

计算 $F(h)$ 的导数。向量求导是矩阵求导的特殊情况，故同样参考第 23.4 节，有

$$\nabla F(h) = \frac{\partial F(h)}{\partial h} = -2W^T(v - Wh) = -2W^Tv + 2W^TWh, \quad \frac{\partial^2 F(h)}{\partial h \partial h} = 2W^TW$$

进而将 $F(h)$ 按二阶泰勒展开

$$F(h) = F(h^t) + (h - h^t)^T \nabla F(h^t) + \frac{1}{2}(h - h^t)^T (2W^T W)(h - h^t)$$

针对泰勒展开形式的新目标函数 $F(h)$ ，猜测存在辅助函数 $G(h, h^t)$ 形如

$$G(h, h^t) = F(h^t) + (h - h^t)^T \nabla F(h^t) + \frac{1}{2}(h - h^t)^T M(h - h^t)$$

4. 构造的 $G(h, h^t)$ 显然满足 $G(h^t, h^t) = F(h^t)$ ，故欲使得 $G(h, h^t)$ 为 $F(h)$ 的辅助函数，只需使得 $G(h, h^t) \geq F(h)$ ，即构造合适的矩阵 M 使得

$$(h - h^t)^T (M - 2W^T W)(h - h^t) \geq 0 \implies M - 2W^T W \text{ 为半正定阵}$$

从而又将辅助函数构造问题转化为形如 $M - 2W^T W$ 的半正定阵的构造问题。注意到如下引理

● ● ●

引理

对于非负对称阵 A 和正向量 x ，则可证如下矩阵为半正定阵

$$\hat{A} = \text{Diag} \left(\frac{(Ax)_i}{x_i} \right) - A$$

³非负矩阵分解 (NMF) 论文笔记 (1): https://blog.csdn.net/Bear_Kai/article/details/73498586

式中 $\text{Diag}(y_i)$ 表示对角线元素取 y_i 的对角阵，则 \widehat{A} 的对角线元素为 $\sum_{j \neq i} \frac{x_j a_{ij}}{x_i}$ ，非对角线元素为 $-a_{ij}$ ，其中 a_{ij} 为矩阵 A 的元素。进一步给出引理证明。对于任意向量 v ，则

$$\begin{aligned} v^T \widehat{A} v &= \sum_{ij} v_i \widehat{A}_{ij} v_j = \sum_i \left[v_i^2 \widehat{A}_{ii} + \sum_{j \neq i} v_i \widehat{A}_{ij} v_j \right] = \sum_i \sum_{j \neq i} v_i^2 \frac{x_j a_{ij}}{x_i} - v_i v_j a_{ij} \\ &= \sum_i \sum_{j \neq i} \frac{a_{ij}}{x_i x_j} (x_j^2 v_i^2 - x_i x_j v_i v_j) \stackrel{a_{ij}=a_{ji}}{=} \sum_i \sum_{j \neq i} \frac{a_{ji}}{x_i x_j} (x_j^2 v_i^2 - x_i x_j v_i v_j) \end{aligned}$$

上式中下标 i 表示行，下标 j 表示列。但实际上 i, j 的意义完全可以对调，即 $\widehat{A} v = \sum_{ij} v_i \widehat{A}_{ij} v_j = \sum_{ji} v_j \widehat{A}_{ji} v_i$ ，此时有

$$v^T \widehat{A} v = \sum_i \sum_{j \neq i} \frac{a_{ji}}{x_i x_j} (x_j^2 v_i^2 - x_i x_j v_i v_j) = \sum_j \sum_{i \neq j} \frac{a_{ji}}{x_i x_j} (x_i^2 v_j^2 - x_i x_j v_i v_j)$$

又因为 $\sum_j \sum_{i \neq j}$ 与 $\sum_i \sum_{j \neq i}$ 实际意义相同，因此有

$$v^T \widehat{A} v = \frac{1}{2} \sum_i \sum_{j \neq i} \frac{a_{ij}}{x_i x_j} (x_j^2 v_i^2 - x_i x_j v_i v_j) + \frac{1}{2} \sum_i \sum_{j \neq i} \frac{a_{ij}}{x_i x_j} (x_i^2 v_j^2 - x_i x_j v_i v_j) = \frac{1}{2} \sum_i \sum_{j \neq i} \frac{a_{ij}}{x_i x_j} (x_i v_j - x_j v_i)^2 \geq 0$$

即证 \widehat{A} 为半正定阵。

基于上述引理，注意到 $W^T W$ 为非负对称阵，并强假设向量 h^t 为正向量（NMF 仅要求其非负），则可构造对角阵 M 如下使得 $M - 2W^T W$ 为半正定阵，从而使得 $G(h, h^t)$ 为 $F(h)$ 的辅助函数

$$M_{kj} = \frac{2\delta_{kj}(W^T W h^t)_k}{h_k^t}$$

δ_{kj} 为指示函数，当且仅当 $k = j$ 时有 $\delta_{kj} = 1$ ，其余情况均为 0；

5. 至此，即可通过优化辅助函数 $G(h, h^t)$ 以降低 $F(h)$ 的取值，易知

$$\frac{\partial G(h, h^t)}{\partial h} = 0 \implies \nabla F(h) + M(h - h^t) = 0 \implies h = h^t - M^{-1} \nabla F(h) \implies h_k = \frac{h_k^t (W^T v)_k}{(W^T W h^t)_k}$$

上式即为矩阵 H 的乘法更新公式，同理也可导出 W 的迭代更新公式，并证明了相关更新公式的收敛性；

6. 对于泊松分布假设的 NMF 问题，同样可通过构造辅助函数 $G(h, h^t)$ 导出相应的更新公式并保证收敛性

$$\begin{aligned} F(h) &= \sum_i v_i \ln \frac{v_i}{\sum_j W_{ij} h_j} - v_i + \sum_j W_{ij} h_j \\ G(h, h^t) &= \sum_i (v_i \ln v_i - v_i) + \sum_{ij} W_{ij} h_j - \sum_{ij} v_i \frac{W_{ij} h_j^t}{\sum_k W_{ik} h_k^t} \left(\ln W_{ij} h_j \ln \frac{W_{ij} h_j^t}{\sum_k W_{ik} h_k^t} \right) \end{aligned}$$

20.3 基于零膨胀 Tweedie 分布假设的 NMF 模型

- 第 20.2 节已详细介绍了 NMF 算法的基本建模思路和算法推导。NMF 算法之所以在经典降维算法中占据重要地位，主要是因其高度灵活的建模特性，使得研究者可根据各自研究问题的特性建立针对性的目标函数从而得到更好地降维效果与更高的结果解释性。例如在经典的 NMF 模型中即可针对一般情况选用基于高斯分布误差假设的 NMF 模型，而针对计数数据可切换为基于泊松分布误差假设的 NMF 模型；
- 基于估计误差分布的不同假设即可导出不同的 NMF 模型，因此若假设估计误差服从一个更为一般的分布（不同的参数取值对应多种分布），则可得到更为一般的 NMF 模型；
- Tweedie 分布是正态分布、泊松分布、伽马分布等一系列分布的一般形式，分布具有三个参数 μ, ϕ, β ，对于随机变量 y ，若其服从 Tweedie 分布，则其概率密度函数 $f(y|\mu, \phi, \beta)$ 有

$$f(y|\mu, \phi, \beta) = \alpha_\beta(y, \phi) \exp \left\{ \frac{y\theta_\beta(\mu) - \kappa_\beta(\mu)}{\phi} \right\}, \quad \theta_\beta(\mu) = \begin{cases} \frac{\mu^{\beta-1} - 1}{\beta - 1} & \beta \neq 1 \\ \ln \mu & \beta = 1 \end{cases}, \quad \kappa_\beta(\mu) = \begin{cases} \frac{\mu^\beta - 1}{\beta} & \beta \neq 0 \\ \ln \mu & \beta = 0 \end{cases}$$

其中 μ 为分布期望; ϕ 与分布离散性正相关, 具体方差为 $\phi\mu^{2-\beta}$; 参数 $\beta \in (-\infty, 1] \cup [2, +\infty)$ 决定了分布的具体形状, $\beta = 2$ 表示正态分布, $\beta = 1$ 表示泊松分布, $\beta = 0$ 表示伽马分布 (指数分布), $\beta \in (0, 1)$ 表示复合泊松-伽马分布 (**Compound Poisson/Gamma distribution**), $\beta = -1$ 表示 **Wald** 分布; 函数 $\alpha_\beta(y, \phi)$ 与 β 有关, 但仅在少数几种特定情况具有封闭形式;

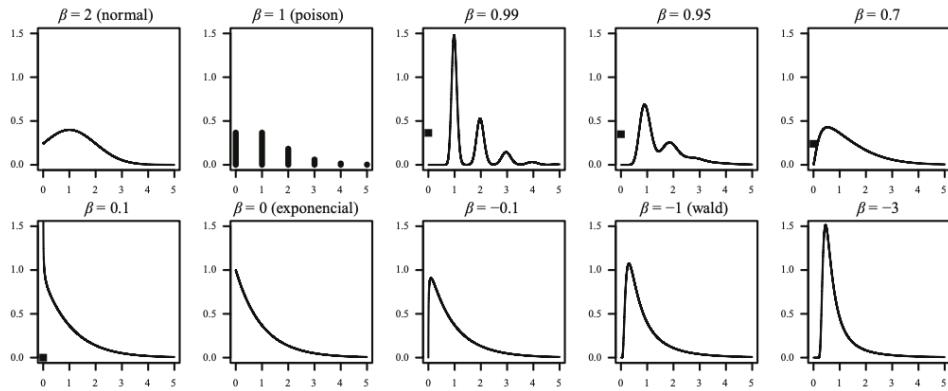


图 20.1 参数 β 不同取值下的 Tweedie 分布概率密度函数形状。

4. 因为多种分布均可表示为 Tweedie 分布的特殊形式, 故基于 Tweedie 分布假设构建 NMF 模型可使得模型适应更一般的数据分布特征。但是当数据存在明显的“混合分布”特征时, 算法仍将失效, 此时需要引入相应的混合分布模型。混合高斯模型 (见第 23.8 节) 与零膨胀模型 (见第 2*II 节) 均是典型的混合分布模型, 本节重点考虑零膨胀的情况——即数据中存在大量的零数据, 介绍基于零膨胀 Tweedie 分布假设的 NMF 模型⁴;
5. 对于原数据矩阵 V , 基于 NMF 分解后得到 $V \approx WH$, 则对于任意元素 V_{ij} , 假设其服从零膨胀 Tweedie 分布, 其中 Tweedie 分布的期望为 $(WH)_{ij}$, 而零膨胀的比例为 ω , 则 V_{ij} 的概率密度函数有

$$f(V_{ij}|(WH)_{ij}, \omega, \phi, \beta) = \omega I(V_{ij} = 0) + (1 - \omega)\alpha_\beta(V_{ij}, \phi) \exp\left\{\frac{V_{ij}\theta_\beta((WH)_{ij}) - \kappa_\beta((WH)_{ij})}{\phi}\right\}$$

相应的得到对数似然函数 $\ln L(W, H, \omega)$ 有

$$\ln L(W, H, \omega) = \sum_{i,j} \ln \left[\omega I(V_{ij} = 0) + (1 - \omega)\alpha_\beta(V_{ij}, \phi) \exp\left\{\frac{V_{ij}\theta_\beta((WH)_{ij}) - \kappa_\beta((WH)_{ij})}{\phi}\right\} \right]$$

上式对数似然函数中 \ln 内部存在加和的形式, 使得极值点计算不便, 故一般不直接基于极大似然法求解。对于此类混合概率分布模型, 一般基于 EM 算法进行参数估计 (详见第 23.8 节)。具体地引入 0-1 二值隐变量 z_{ij} 表示样本 V_{ij} 所属的分布类型。令 $z_{ij} = 1$ 表示 V_{ij} 以 100% 的概率取 0, $z_{ij} = 0$ 表示 V_{ij} 服从 Tweedie 分布, 则在引入隐变量 z_{ij} 后对数似然函数 $\ln L(W, H, Z, \omega)$ 可改写为

$$\begin{aligned} \ln L(W, H, Z, \omega) &= \sum_{i,j} \left\{ z_{ij} \ln \omega + (1 - z_{ij}) \ln \left[(1 - \omega)\alpha_\beta(V_{ij}, \phi) \exp\left\{\frac{V_{ij}\theta_\beta((WH)_{ij}) - \kappa_\beta((WH)_{ij})}{\phi}\right\} \right] \right\} \\ &= \sum_{i,j} \left\{ z_{ij} \ln \omega + (1 - z_{ij}) \left[\ln(1 - \omega) + \ln \alpha_\beta(V_{ij}, \phi) + \frac{V_{ij}\theta_\beta((WH)_{ij}) - \kappa_\beta((WH)_{ij})}{\phi} \right] \right\} \end{aligned}$$

此时对数似然函数内的 \ln 内部加和结构即因为隐变量 z_{ij} 的引入而消除;

6. 基于 EM 算法估计参数 W, H, Z, ω , 需要迭代执行以下两步:

- 期望步: 计算隐变量 z_{ij} 的后验估计。基于贝叶斯公式, 有

$$z_{ij} = \begin{cases} \frac{\omega P(z_{ij} = 1)}{\omega P(z_{ij} = 1) + (1 - \omega)P(z_{ij} = 0)} = \frac{\omega}{\omega + (1 - \omega)f(0|(WH)_{ij}, \phi, \beta)} & V_{ij} = 0 \\ 0 & V_{ij} \neq 0 \end{cases}$$

⁴A non-negative matrix factorization model based on the zero-inflated Tweedie distribution (Hiroyasu Abe1 & Hiroshi Yadohisa, 2017)

- 最大化步：固定 z_{ij} 基于极大似然法估计 W, H, ω 。首先估计 ω ，有

$$\frac{\partial \ln L(W, H, Z, \omega)}{\partial \omega} = \frac{1}{\omega} \sum_{i,j} z_{ij} - \frac{1}{1-\omega} \sum_{i,j} (1-z_{ij}) = 0 \implies \omega = \frac{1}{mn} \sum_{i,j} z_{ij}$$

进一步估计 W, H ，只保留 $L(W, H, Z, \omega)$ 中与 W, H 有关的项，则

$$\begin{aligned} W, H &= \arg \max_{W, H \geq 0} \ln L(W, H, Z, \omega) = \arg \max_{W, H \geq 0} \sum_{i,j} (1-z_{ij}) [V_{ij} \theta_\beta((WH)_{ij}) - \kappa_\beta((WH)_{ij})] \\ &= \arg \max_{W, H \geq 0} \sum_{i,j} (1-z_{ij}) \left[V_{ij} \frac{(WH)_{ij}^{\beta-1} - 1}{\beta-1} - \frac{(WH)_{ij}^\beta - 1}{\beta} \right] \\ &= \arg \min_{W, H \geq 0} \frac{1}{\beta} \sum_{i,j} (1-z_{ij})(WH)_{ij}^\beta - \frac{1}{\beta-1} \sum_{i,j} (1-z_{ij})V_{ij}(WH)_{ij}^{\beta-1} \end{aligned}$$

注意到上述目标函数在数学上其实要求 $\beta \notin \{0, 1\}$ ，但因为

$$\lim_{\beta \rightarrow 0} \frac{1}{\beta} \sum_{i,j} (1-z_{ij}) \mu^\beta = \sum_{i,j} (1-z_{ij}) \ln \mu, \quad \lim_{\beta \rightarrow 1} \frac{1}{\beta-1} \sum_{i,j} (1-z_{ij}) V_{ij} \mu^{\beta-1} = \sum_{i,j} (1-z_{ij}) V_{ij} \ln \mu$$

因此在补充了 $\theta_{\beta=1}(\mu) = \ln \mu$, $\kappa_{\beta=0}(\mu) = \ln \mu$ 后上述目标函数即对 β 连续且可导，因此基于上式推导得到的 W, H 更新公式也将适用于 $\beta \in \{0, 1\}$ 的情况。尝试基于梯度下降公式（详见第 20.2 节）构造 W, H 的乘法更新公式。为简化考虑先假设 $z_{ij} = 0$ （即不考虑零膨胀情况），并令 $X = WH$ ，则基于 Tweedie 分布假设的 NMF 目标函数实际上即是最小化 V 与 X 之间的 β 散度 (β -divergence)

$$\begin{aligned} f(X) &= \frac{1}{\beta} \sum_{i,j} X_{ij}^\beta - \frac{1}{\beta-1} \sum_{i,j} V_{ij} X_{ij}^{\beta-1} = \frac{1}{\beta} \text{tr} \{ E^\top X^\beta \} - \frac{1}{\beta-1} \text{tr} \{ V^\top X^{\beta-1} \} \\ \implies df &= \text{tr} \{ E^\top (X^{\beta-1} \odot dX) \} - \text{tr} \{ V^\top (X^{\beta-2} \odot dX) \} \end{aligned}$$

上式中 E 为与 X 尺寸相同，元素全为 1 的矩阵。因为迹运算下矩阵乘法与 Hadamard 乘法具有交换公式 $\text{tr}(X^\top(Y \odot Z)) = \text{tr}((X \odot Y)^\top Z)$ ，则

$$\begin{aligned} df &= \text{tr} \{ (E \odot (X^{\beta-1})^\top) dX \} - \text{tr} \{ (V \odot (X^{\beta-2})^\top) dX \} \\ &= \text{tr} \{ (X^{\beta-1} - V \odot X^{\beta-2})^\top dX \} = \text{tr} \{ H (X^{\beta-1} - V \odot X^{\beta-2})^\top dW \} + \text{tr} \{ (X^{\beta-1} - V \odot X^{\beta-2})^\top W dH \} \\ \implies \frac{\partial f}{\partial W} &= (X^{\beta-1} - V \odot X^{\beta-2}) H^\top, \quad \frac{\partial f}{\partial H} = W^\top (X^{\beta-1} - V \odot X^{\beta-2}) \end{aligned}$$

代入梯度下降更新公式 $W_{ik} = W_{ik} - \mu_{ik} \left[\frac{\partial f}{\partial W} \right]_{ik}$, $H_{kj} = H_{kj} - \eta_{kj} \left[\frac{\partial f}{\partial H} \right]_{kj}$ ，则只需选择合适的步长 μ_{ik}, η_{kj} 即可构造 W, H 的乘法更新公式

$$\begin{cases} W = W \odot \frac{(V \odot (WH)^{\beta-2}) H^\top}{(WH)^{\beta-1} H^\top} & \mu_{ik} = \frac{W_{ik}}{[(WH)^{\beta-1} H^\top]_{ik}} \\ H = H \odot \frac{W^\top (V \odot (WH)^{\beta-2})}{W^\top (WH)^{\beta-1}} & \eta_{ij} = \frac{H_{kj}}{[W^\top (WH)^{\beta-1}]_{kj}} \end{cases}$$

最后考虑 $1 - z_{ij}$ ，得到零膨胀情况下的 W, H 乘法更新公式

$$W = W \odot \frac{[(E-Z) \odot V \odot (WH)^{\beta-2}] H^\top}{[(E-Z) \odot (WH)^{\beta-1}] H^\top}, \quad H = H \odot \frac{W^\top [(E-Z) \odot V \odot (WH)^{\beta-2}]}{W^\top [(E-Z) \odot (WH)^{\beta-1}]}$$

上式即为基于梯度下降法构造的零膨胀情况下 W, H 的乘法更新公式。但第 20.2 节的相关内容已指出，基于梯度下降法构造的更新公式并不一定能保证算法收敛，实际上上式仅在 $\beta \in [1, 2]$ 时收敛。为保证 NMF 算法的收敛性，构造辅助函数严格推导，得到本问题 W, H 的乘法更新规则⁵

$$W = W \odot \left[\frac{[(E-Z) \odot V \odot (WH)^{\beta-2}] H^\top}{[(E-Z) \odot (WH)^{\beta-1}] H^\top} \right]^{\rho(\beta)} \quad H = H \odot \left[\frac{W^\top [(E-Z) \odot V \odot (WH)^{\beta-2}]}{W^\top [(E-Z) \odot (WH)^{\beta-1}]} \right]^{\rho(\beta)}$$

⁵Convergence-guaranteed multiplicative algorithms for non-negative matrix factorization with divergence (Nakano et al., 2010)

$$\rho(\beta) = \begin{cases} 1/(2-\beta) & \beta < 1 \\ 1 & 1 \leq \beta \leq 2 \\ 1/(\beta-1) & \beta > 2 \end{cases}$$

基于 EM 算法估计参数 W, H, Z, ω , 迭代收敛的判据为零膨胀 Tweedie 分布的似然函数, 即

$$LL^{(k)} - LL^{(k-1)} < \varepsilon, \quad LL^{(k)} = \sum_{i,j} \ln f(V_{ij}|(WH)_{ij}^{(k)}, \omega^{(k)}, \phi, \beta)$$

7. 以上即为基于零膨胀 Tweedie 分布的 NMF 模型求解算法。尽管推导过程对 $\beta \in (-\infty, 1] \cup [2, +\infty)$ 均成立, 但最终结果仅在 $\beta \in \{1, 2\} \cup (0, 1)$ 时有意义, 因为:

- 当 $\beta \in (1, 2) \cup (2, +\infty)$ 时 Tweedie 分布的概率密度函数无法被计算;
- 当 $\beta \in (-\infty, 0)$ 时 Tweedie 分布于 0 处的概率密度恒为 0, 使得 $V_{ij} = 0$ 时恒有 $z_{ij} = 1$, 此时相当于仅对非零数据进行 NMF 分解, “零膨胀”假设将毫无意义。

20.4 张量分解 (*Tensor decomposition/factorization*)

20.4.1 张量基础知识

符号 为区分一般的高阶张量与低维的向量、矩阵, 以手写风格的大写字母表示张量, 如 X , 而以加粗大写字母、加粗小写字母和小写字母分别表示矩阵、向量和标量, 如 $\mathbf{X}, \mathbf{x}, x$;

维度 张量的维度 (**dimension**) 也就是张量的阶数 (**order**), 也可称为张量的模 (**mode**);

Fiber 意为“纤维”, 张量中类似于矩阵行向量与列向量的结构。严格定义为固定一个张量的大部分维度仅保留其中的一个维度所形成的向量。一个三阶张量 X 具有列、行和通道三种 **fiber**, 分别记为 $\mathbf{x}_{:jk}, \mathbf{x}_{ik}, \mathbf{x}_{ij}$, 称为 mode-1 (column) fiber, mode-2 (row) fiber 和 mode-3 (tube) fiber;

Slice 意为“切片”, 张量中类似于矩阵的结构。严格定义为固定一个张量的大部分维度仅保留其中的两个维度所形成的矩阵。同样地一个三阶张量 X 具有水平、垂直和正面三种 **slice**, 分别记为 $\mathbf{X}_{i::}, \mathbf{X}_{::j}, \mathbf{X}_{::k}$, 称为 mode-1 (horizontal) slice, mode-2 (lateral) slice 和 mode-3 (frontal) slice;

内积 (inner product) 两个形状相同的张量对应分量相乘并求和, 写为 $\langle X, Y \rangle$;

外积 (outer product) 两个任意维度/形状的张量任意分量两两相乘形成的更高维张量。记张量 X, Y 的尺寸分别为 $(k_1, \dots, k_m), (l_1, \dots, l_n)$, 则两者外积后的张量的尺寸为 $(k_1, \dots, k_m, l_1, \dots, l_n)$, 写作 $X \circ Y$, 对其中任意元素有

$$(X \circ Y)_{i_1, \dots, i_m, j_1, \dots, j_n} = x_{i_1, \dots, i_m} \times y_{j_1, \dots, j_n}$$

单秩张量 (rank-one tensor) 如果一个 N 阶张量 $X \in \mathbb{R}_{I_1 \times \dots \times I_N}$ 能写成 N 个向量的外积, 则称该张量为单秩张量 (**rank-one tensor**)。从微观角度上看即该张量的每一个元素均为 N 个向量相应元素的乘积

$$X = a^{(1)} \circ \dots \circ a^{(N)} \iff x_{i_1, \dots, i_N} = a_{i_1}^{(1)} \times \dots \times a_{i_N}^{(N)}$$

n 模矩阵化 (n-mode matricization) 张量矩阵化是指将高阶张量元素重新排列为矩阵的过程。**n** 模矩阵化是张量矩阵化的一种方法。回顾 fiber 的概念, 对于一个 N 阶张量 $X \in \mathbb{R}_{I_1 \times \dots \times I_N}$, 显然其具有 $J = I_1 \times \dots \times I_{n-1} \times I_{n+1} \dots \times I_N$ 个 mode-n fiber, 而 **n** 模矩阵化就是将其重新排列为 $I_n \times J$ 的矩阵, 记为 $\mathbf{X}_{(n)}$ 。原张量中的元素 (i_1, \dots, i_N) 被映射到新矩阵元素 (i_n, j) , 其中

$$j = 1 + \sum_{k=1, k \neq n}^N (i_k - 1)J_k, \quad J_k = \prod_{m=1, m \neq n}^{k-1} I_m$$

Tucker 秩 N 阶张量 X 的 Tucker 秩为一个 N 维向量, 其第 n 个元素为张量 X 的 **n** 模矩阵化 $\mathbf{X}_{(n)}$ 的秩, 又被称为张量的 **n** 秩 (**n-rank**);

n 模积 (n-mode product) 是张量与矩阵 (或向量) 相乘的一种方式。对于一个 N 阶张量 $X \in \mathbb{R}_{I_1 \times \dots \times I_N}$ 和一个矩阵 $\mathbf{U} \in \mathbb{R}_{K \times I_n}$, 两者的 n 模积的结果是一个 $I_1 \times \dots \times I_{n-1} \times K \times I_{n+1} \times \dots \times I_N$ 的张量, 即将维度 I_n 替换为 K , 记为 $X \times_n \mathbf{U}$, 其任意元素按下式计算

$$(X \times_n \mathbf{U})_{i_1 \dots i_{n-1} k i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 \dots i_N} u_{k i_n}$$

进一步介绍计算原理。首先将 X 模矩阵化得到 $\mathbf{X}_{(n)} \in \mathbb{R}_{I_n \times J}$, $J = I_1 \times \dots \times I_{n-1} \times I_{n+1} \dots \times I_N$, 进而计算矩阵乘法 $\mathbf{U}\mathbf{X}_{(n)} \in \mathbb{R}_{K \times J}$, 最后将 $\mathbf{U}\mathbf{X}_{(n)}$ 重新堆叠为 $I_1 \times \dots \times I_{n-1} \times K \times I_{n+1} \times \dots \times I_N$ 的张量。由计算流程不难得知一个张量与多个矩阵连续相乘的结果与顺序无关

$$X \times_n \mathbf{U} \times_m \mathbf{V} = X \times_m \mathbf{V} \times_n \mathbf{U}$$

另外, 若矩阵 $\mathbf{U} \in \mathbb{R}_{K \times I_n}$, $\mathbf{V} \in \mathbb{R}_{L \times K}$, 则

$$X \times_n \mathbf{U} \times_n \mathbf{V} = X \times_n (\mathbf{V}\mathbf{U})$$

20.4.2 CP 分解 (canonical polyadic decomposition, CPD)

1. 张量分解是机器学习中十分重要的一种方法。作为矩阵分解的自然扩展, 张量分解旨在解析高维数据 (或参数) 的低维模式 (或结构)。CP 分解即是一类经典的张量分解算法;
2. CP 分解是矩阵因子分解的直观推广, 其基本思想是将一个张量分解为多个单秩张量的和。首先以三阶张量为例。假设任意张量 $X \in \mathbb{R}_{I \times J \times K}$, 则 CP 分解认为存在因子矩阵 (factor matrix) $\mathbf{A} \in \mathbb{R}_{I \times R}$, $\mathbf{B} \in \mathbb{R}_{J \times R}$, $\mathbf{C} \in \mathbb{R}_{K \times R}$, 使得

$$x_{ijk} \simeq \sum_{r=1}^R a_{ir} b_{jr} c_{kr} \iff X \simeq \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

上式中 R 为 CP 分解的秩, 为算法超参; $\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r$ 分别为因子矩阵 $\mathbf{A}, \mathbf{B}, \mathbf{C}$ 的列向量。通常假设因子矩阵为列向量经过标准化后的矩阵, 将提取的权重并入向量 $\lambda \in \mathbb{R}^R$, 则上式又可改写为

$$x_{ijk} \simeq \sum_{r=1}^R \lambda_r a_{ir} b_{jr} c_{kr} = \sum_{r=1}^R \hat{a}_{ir} b_{jr} c_{kr} \iff X \simeq \sum_{r=1}^R \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = \sum_{r=1}^R \hat{\mathbf{a}}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \quad \hat{\mathbf{A}} = \text{Adiag}(\lambda)$$

CP 分解的目标即是求解列向量标准后的因子矩阵 $\mathbf{A}, \mathbf{B}, \mathbf{C}$ 及相应的权重向量 λ ;

3. 为求解因子矩阵, 显然可将优化目标建模为如下

$$\lambda, \mathbf{A}, \mathbf{B}, \mathbf{C} = \arg \min_{\lambda, \mathbf{A}, \mathbf{B}, \mathbf{C}} \mathbf{f} = \arg \min_{\lambda, \mathbf{A}, \mathbf{B}, \mathbf{C}} \frac{1}{2} \left\| X - \sum_{r=1}^R \hat{\mathbf{a}}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \right\|_F^2$$

基于交替迭代优化方法优化上述目标——即每轮迭代中首先固定 \mathbf{B}, \mathbf{C} 优化 $\hat{\mathbf{A}}$, 再固定 $\hat{\mathbf{A}}, \mathbf{C}$ 优化 \mathbf{B} , 最后固定 $\hat{\mathbf{A}}, \mathbf{B}$ 优化 \mathbf{C} 。以 $\hat{\mathbf{A}}$ 优化为例, 需要计算目标函数 f 对矩阵 $\hat{\mathbf{A}}$ 的导数, 为此需要将目标函数改写为矩阵形式。注意到张量 X 可通过矩阵化展开为矩阵, 且有

$$\mathbf{X}_{(1)} \simeq \hat{\mathbf{A}}(\mathbf{C} \odot \mathbf{B})^\top, \quad \mathbf{X}_{(2)} \simeq \mathbf{B}(\mathbf{C} \odot \hat{\mathbf{A}})^\top, \quad \mathbf{X}_{(3)} \simeq \mathbf{C}(\mathbf{B} \odot \hat{\mathbf{A}})^\top$$

则可计算目标函数 f 对矩阵 $\hat{\mathbf{A}}$ 的导数

$$\begin{aligned} \frac{\partial}{\partial \hat{\mathbf{A}}} \left[\frac{1}{2} \left\| \mathbf{X}_{(1)} - \hat{\mathbf{A}}(\mathbf{C} \odot \mathbf{B})^\top \right\|_F^2 \right] &= -(\mathbf{X}_{(1)} - \hat{\mathbf{A}}(\mathbf{C} \odot \mathbf{B})^\top)(\mathbf{C} \odot \mathbf{B}) = 0 \\ \Rightarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B}) &= \hat{\mathbf{A}}(\mathbf{C} \odot \mathbf{B})^\top(\mathbf{C} \odot \mathbf{B}) \Rightarrow \hat{\mathbf{A}} = \mathbf{X}_{(1)}[(\mathbf{C} \odot \mathbf{B})^\top]^\dagger \end{aligned}$$

上式中 \odot 表示矩阵的 Khatri-Rao 积, $[(\mathbf{C} \odot \mathbf{B})^\top]^\dagger$ 表示矩阵 $[(\mathbf{C} \odot \mathbf{B})^\top]$ 的 M-P 广义逆。关于 Khatri-Rao 积与 M-P 广义逆的相关介绍详见第 23.3 节。同理可按上式求解另外的因子矩阵 \mathbf{B}, \mathbf{C} 。在固定了 \mathbf{B}, \mathbf{C} 后 $\hat{\mathbf{A}}$ 的求解即是典型的最小二乘问题, 而上述迭代中交替固定部分参数进行最小二乘的算法则被称为交替最小二乘 (alternating least square, ALS) 算法, 是矩阵分解的标志性算法;

4. 按上式基于交替最小二乘求解 CP 分解问题需要计算形如 $[(\mathbf{C} \odot \mathbf{B})^\top]$ 的矩阵的 M-P 广义逆。根据 Khatri-Rao 积的定义, $[(\mathbf{C} \odot \mathbf{B})^\top]$ 的维度为 $R \times (JK)$, 而 M-P 广义逆针对的大规模矩阵的计算效率极低 ($O(n^3)$, n 为矩阵元素数目), 因此如果能减少计算广义逆的矩阵的规模则可显著提升 CP 分解的效率。注意到 M-P 广义逆与 Khatri-Rao 积具有如下性质 (详见第 23.3 节)

$$[(\mathbf{C} \odot \mathbf{B})^\top]^\dagger = [(\mathbf{C} \odot \mathbf{B})^\dagger]^\top = \left((\mathbf{C}^\top \mathbf{C} * \mathbf{B}^\top \mathbf{B})^\dagger (\mathbf{C} \odot \mathbf{B})^\top \right)^\top = (\mathbf{C} \odot \mathbf{B}) \left((\mathbf{C}^\top \mathbf{C} * \mathbf{B}^\top \mathbf{B})^\dagger \right)^\top$$

又因为矩阵 $(\mathbf{C}^\top \mathbf{C} * \mathbf{B}^\top \mathbf{B})$ 为对称阵, 故 $(\mathbf{C}^\top \mathbf{C} * \mathbf{B}^\top \mathbf{B})^\dagger$ 也为对称阵, 则

$$\widehat{\mathbf{A}} = \mathbf{X}_{(1)} [(C \odot B)^\top]^\dagger = \mathbf{X}_{(1)} (C \odot B) \left((\mathbf{C}^\top \mathbf{C} * \mathbf{B}^\top \mathbf{B})^\dagger \right)^\top = \mathbf{X}_{(1)} (C \odot B) (\mathbf{C}^\top \mathbf{C} * \mathbf{B}^\top \mathbf{B})^\dagger$$

按上式进行 CP 分解, 则只需计算 $R \times R$ 的矩阵 $\mathbf{C}^\top \mathbf{C} * \mathbf{B}^\top \mathbf{B}$ 的 M-P 广义逆。只要 R 远小于 JK 即可显著提升算法效率;

5. 最后将上述算法扩展至任意 N 阶张量 $X \in \mathbb{R}_{I_1 \times \dots \times I_N}$, 则 CP 分解算法旨在求解最后的列归一化后的因子矩阵 $\{\mathbf{A}^{(n)} \in \mathbb{R}_{I_n \times R} | n = 1, \dots, N\}$ 和权重向量 $\lambda \in \mathcal{R}_R$ 使得

$$X \simeq \sum_{r=1}^R \lambda_r \mathbf{a}_r^{(1)} \circ \dots \circ \mathbf{a}_r^{(N)}$$

而对张量 X 进行模 n 矩阵化后上式可改写为矩阵形式

$$\mathbf{X}_{(n)} \simeq \mathbf{A}^{(n)} \mathbf{\Lambda} \left(\mathbf{A}^{(N)} \odot \dots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \dots \odot \mathbf{A}^{(1)} \right), \quad \mathbf{\Lambda} = \text{diag}(\lambda)$$

从而得到最终的算法如下

Algorithm 20.1 基于交替最小二乘 (ALS) 算法的 CP 张量分解

输入: N 阶张量 $X \in \mathbb{R}_{I_1 \times \dots \times I_N}$, CP 分解的秩 R

- 1: 初始化 $\{\mathbf{A}^{(n)} \in \mathbb{R}_{I_n \times R} | n = 1, \dots, N\}$
- 2: **repeat**
- 3: **for** $n = 1, \dots, N$ **do**
- 4: $\mathbf{V} \leftarrow \mathbf{A}^{(1)\top} \mathbf{A}^{(1)} * \dots * \mathbf{A}^{(n-1)\top} \mathbf{A}^{(n-1)} * \mathbf{A}^{(n+1)\top} \mathbf{A}^{(n+1)} * \dots * \mathbf{A}^{(N)\top} \mathbf{A}^{(N)}$
- 5: $\mathbf{A}^{(n)} \leftarrow \mathbf{X}_{(n)} (\mathbf{A}^{(N)} \odot \dots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \dots \odot \mathbf{A}^{(1)}) \mathbf{V}^\dagger$
- 6: 对矩阵 $\mathbf{A}^{(n)}$ 进行列标准化并计算权重向量 λ : $\lambda_r \leftarrow \|\mathbf{a}_r^{(n)}\|$, $\mathbf{a}_r^{(n)} \leftarrow \mathbf{a}_r^{(n)} / \lambda_r$
- 7: **end for**
- 8: **until** 到达最大迭代次数或误差收敛

输出: $\lambda, \{\mathbf{A}^{(n)} \in \mathbb{R}_{I_n \times R} | n = 1, \dots, N\}$

6. 需要说明的是, ALS 算法具有如下缺点:

- 需要多次迭代才能收敛;
- 无法保证收敛到全局最小点, 也无法保证是驻点, 只能保证收敛到目标函数不再下降为止;
- 依赖初始确定的因子矩阵以及 R 。

因此许多学者对 ALS 算法做了很多种改进, 比如线搜索法、Tikhonov 正则化等等。但在时间不是紧缺的情况下, 研究表明并没有一个比 ALS 全面优秀的算法存在。

20.4.3 Tucker 分解

1. Tucker 分解是除 CP 分解之外的另一种经典张量分解算法, 可视为高维场景下的 SVD 分解 (或 PCA 分解) 算法。与 CP 分解相比, Tucker 分解在建模时提供了更高的自由度。同样以三阶张量为例, 假设任意张量 $X \in \mathbb{R}_{I_1 \times J \times K}$, 则 Tucker 分解将 X 分解为一个核心张量 (core tensor) $\mathcal{G} \in \mathbb{R}_{R_1 \times R_2 \times R_3}$ 和三个因子矩阵 (factor matrix) $\mathbf{A} \in \mathbb{R}_{I_1 \times R_1}, \mathbf{B} \in \mathbb{R}_{J \times R_2}, \mathbf{C} \in \mathbb{R}_{K \times R_3}$

$$x_{ijk} \simeq \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1 r_2 r_3} \cdot a_{ir_1} \cdot b_{jr_2} \cdot c_{kr_3} \iff X \simeq \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1 r_2 r_3} \mathbf{a}_{r_1} \circ \mathbf{b}_{r_2} \circ \mathbf{c}_{r_3} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

上式中 R_1, R_2, R_3 为 Tucker 算法超参。上式也可写为矩阵形式

$$\mathbf{X}_{(1)} \simeq \mathbf{A}\mathbf{G}_{(1)}(\mathbf{C} \otimes \mathbf{B})^\top, \quad \mathbf{X}_{(2)} \simeq \mathbf{B}\mathbf{G}_{(2)}(\mathbf{C} \otimes \mathbf{A})^\top, \quad \mathbf{X}_{(3)} \simeq \mathbf{C}\mathbf{G}_{(3)}(\mathbf{B} \otimes \mathbf{A})^\top$$

扩展至一般的 N 阶张量，则 Tucker 分解的一般形式写为

$$X \simeq G \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_N A^{(N)} \iff X_{(n)} \simeq A^{(n)} G_{(n)} (A^{(N)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)})^\top$$

2. 因为 Tucker 分解较 CP 分解具有更高的自由度，故前者的分解结果对原张量也具有更强的拟合性。实际上，记原张量的 Tucker 秩为 (R_1^*, \dots, R_N^*) ，若 Tucker 分解的超参满足 $R_n = R_n^*, \forall n$ ，则原张量 X 存在多个精确的 Tucker 分解（即上式中 “ \simeq ” 可改为 “ $=$ ”）；反之若 $R_n < R_n^*, \exists n$ ，则原张量只存在近似 Tucker 分解，此时称为截断 Tucker 分解 (truncated Tucker decomposition)；
3. 首先证明当 $R_n = R_n^*, \forall n$ 时对任意张量 X 均存在精确 Tucker 分解。首先对 X 的 n 模矩阵化 $\mathbf{X}_{(n)} \in \mathbb{R}_{I_n \times J}$ 作紧致奇异值分解 (compact singular value decomposition, compact SVD)

$$\mathbf{X}_{(n)} = \mathbf{U}^{(n)} \Sigma^{(n)} \mathbf{V}^{(n)\top}, \quad \mathbf{U}^{(n)\top} \mathbf{U}^{(n)} = \mathbf{I}, \quad \mathbf{V}^{(n)\top} \mathbf{V}^{(n)} = \mathbf{I}$$

上式中 $\Sigma \in \mathbb{R}_{R_n^* \times R_n^*}$ 为对角阵，对角线元素为 $\mathbf{X}_{(n)}$ 的所有非零奇异值；左奇异矩阵 $\mathbf{U}^{(n)} \in \mathbb{R}_{I_n \times R_n^*}$ 和右奇异矩阵 $\mathbf{V}^{(n)} \in \mathbb{R}_{R_n^* \times J}$ 均为正交阵， $\mathbf{U}^{(n)}$ 的列向量与 $\mathbf{V}^{(n)}$ 的行向量均为单位正交向量。仅考虑左奇异矩阵 $\mathbf{U}^{(n)}$ ，注意到

$$\begin{aligned} \mathbf{U}^{(n)} \mathbf{U}^{(n)\top} \mathbf{X}_{(n)} &= \mathbf{U}^{(n)} \mathbf{U}^{(n)\top} (\mathbf{U}^{(n)} \Sigma^{(n)} \mathbf{V}^{(n)\top}) = \mathbf{U}^{(n)} (\mathbf{U}^{(n)\top} \mathbf{U}^{(n)}) \Sigma^{(n)} \mathbf{V}^{(n)\top} = \mathbf{U}^{(n)} \Sigma^{(n)} \mathbf{V}^{(n)\top} = \mathbf{X}_{(n)} \\ \implies \mathbf{X}_{(n)} &= \mathbf{U}^{(n)} \mathbf{U}^{(n)\top} \mathbf{X}_{(n)} = \mathbf{U}^{(n)} (\mathbf{U}^{(n)\top} \mathbf{X}_{(n)}) \implies X = (X \times_n \mathbf{U}^{(n)\top}) \times_n \mathbf{U}^{(n)} = X \times_n \mathbf{U}^{(n)\top} \times_n \mathbf{U}^{(n)} \end{aligned}$$

则基于上式易知

$$\begin{aligned} X &= X \times_1 \mathbf{U}^{(1)\top} \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)\top} \times_N \mathbf{U}^{(N)} \\ &= (X \times_1 \mathbf{U}^{(1)\top} \cdots \times_N \mathbf{U}^{(N)\top}) \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)} = G \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_N A^{(N)}, \\ G &= X \times_1 \mathbf{U}^{(1)\top} \times_2 \mathbf{U}^{(2)\top} \cdots \times_N \mathbf{U}^{(N)\top}, \quad A^{(n)} = \mathbf{U}^{(n)} \end{aligned}$$

显然基于紧致 SVD 分解即可得到任意张量 X 的一组精确 Tucker 分解，这也是 Tucker 分解可视为 SVD 分解的高维推广的原因；

4. 进一步证明当 $R_n = R_n^*, \forall n$ 时张量 X 的精确 Tucker 分解不唯一。记 X 的一组精确 Tucker 分解结果为 $\{G, A^{(n)}\}$ ，则易证明 $\{G \times_1 B^{(1)} \cdots \times_N B^{(N)}, A^{(n)} B^{(n)-1}\}$ 同样为 X 的一组精确 Tucker 分解结果，其中 $\forall B^{(n)}$ 为非奇异方阵

$$\begin{aligned} &(G \times_1 B^{(1)} \times_2 B^{(2)} \cdots \times_N B^{(N)}) \times_1 (A^{(1)} B^{(1)-1}) \times_2 (A^{(2)} B^{(2)-1}) \cdots \times_N (A^{(N)} B^{(N)-1}) \\ &= G \times_1 B^{(1)} \times_1 (A^{(1)} B^{(1)-1}) \times_2 B^{(2)} \times_2 (A^{(2)} B^{(2)-1}) \cdots \times_N B^{(N)} \times_N (A^{(N)} B^{(N)-1}) \\ &= G \times_1 (A^{(1)} B^{(1)-1} B^{(1)}) \times_2 (A^{(2)} B^{(2)-1} B^{(2)}) \cdots \times_N (A^{(N)} B^{(N)-1} B^{(N)}) \\ &= G \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_N A^{(N)} = X \end{aligned}$$

从而证明了当 $R_n = R_n^*, \forall n$ 时 Tucker 分解不具有唯一性，这也是 Tucker 分解与 CP 分解的一个主要不同；

5. 考虑到矩阵 SVD 分解具有唯一性，故按照上述过程将其推断至张量分解可保证结果的唯一性，相应的算法称为高阶奇异值分解 (high-order singular value decomposition, HOSVD) 算法

Algorithm 20.2 HOSVD 算法

输入： N 阶张量 $X \in \mathbb{R}_{I_1 \times \cdots \times I_N}$ ，Tucker 分解超参 R_1, \dots, R_N

- 1: **for** $n = 1, \dots, N$ **do**
- 2: 计算 $\mathbf{X}_{(n)}$ 的完全 SVD 分解 $\mathbf{X}_{(n)} = \mathbf{U}^{(n)} \Sigma^{(n)} \mathbf{V}^{(n)\top}$ ► 完全 SVD 分解时 $\mathbf{U}^{(n)}, \mathbf{V}^{(n)}$ 为列正交方阵， $\Sigma^{(n)}$ 含 0 奇异值
- 3: 截取左奇异矩阵 $\mathbf{U}^{(n)}$ 的前 R_n 列得到 $\mathbf{A}^{(n)}$

4: **end for**

$$5: \mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\top} \times_2 \mathbf{A}^{(2)\top} \cdots \times_N \mathbf{A}^{(N)\top}$$

输出: $\mathcal{G}, \{\mathbf{A}^{(n)} \in \mathbb{R}_{I_n \times R_n} | n = 1, \dots, N\}$

显然 HOSVD 算法对 $\forall R_n \in (0, I_n]$ 均适用。但由上述 HOSVD 算法的推导过程易知, 当 $R_n < R_n^*$, $\exists n$ 时 HOSVD 算法得到的结果不是最优的 Tucker 分解。因为从 $\mathcal{X} \simeq \mathcal{X} \times_n \mathbf{U}^{(n)\top} \times_n \mathbf{U}^{(n)}$ 到 $\mathcal{X} \simeq \mathcal{X} \times_1 \mathbf{U}^{(1)\top} \times_1 \mathbf{U}^{(1)} \cdots \times_N \mathbf{U}^{(N)\top} \times_N \mathbf{U}^{(N)}$ 的过程中存在误差累积放大;

6. 进一步推导当 $R_n < R_n^*$, $\exists n$ 时的最优 Tucker 分解。同样地一般情况下最优 Tucker 分解不唯一, 故补充约束 $\mathbf{A}^{(n)\top} \mathbf{A}^{(n)} = \mathbf{I}$, 则 Tucker 分解建模为

$$\min_{\mathcal{G}, \mathbf{A}} \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}\|_F^2, \quad \text{s.t. } \mathbf{A}^{(n)\top} \mathbf{A}^{(n)} = \mathbf{I}$$

对于上述最小二乘问题, 仍按交替最小二乘 (ALS) 算法求解。首先关注核心张量 \mathcal{G} 。假设 $\mathcal{G}, \{\mathbf{A}^{(n)} \in \mathbb{R}_{I_n \times R_n} | n = 1, \dots, N\}$ 为 \mathcal{X} 的最优逼近, 则按定义有

$$\mathbf{X}_{(n)} \simeq \mathbf{A}^{(n)} \mathbf{G}_{(n)} (\mathbf{A}^N \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)})^\top$$

又因为 Kronecker 积的性质 (详见第 23.3 节) 有

$$\begin{aligned} & (\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)})^\top (\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)}) \\ &= (\mathbf{A}^{(N)\top} \otimes \cdots \otimes \mathbf{A}^{(n+1)\top} \otimes \mathbf{A}^{(n-1)\top} \otimes \cdots \otimes \mathbf{A}^{(1)\top}) (\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)}) \\ &= (\mathbf{A}^{(N)\top} \mathbf{A}^{(N)}) \otimes \cdots \otimes (\mathbf{A}^{(n+1)\top} \mathbf{A}^{(n+1)}) \otimes (\mathbf{A}^{(n-1)\top} \mathbf{A}^{(n-1)}) \otimes \cdots \otimes (\mathbf{A}^{(1)\top} \mathbf{A}^{(1)}) = \mathbf{I} \end{aligned}$$

故矩阵 $\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)}$ 同样为列单位正交阵, 则 \mathcal{G} 最优解必然满足

$$\begin{aligned} \mathbf{X}_{(n)} &\simeq \mathbf{A}^{(n)} \mathbf{G}_{(n)} (\mathbf{A}^N \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)})^\top \\ \Rightarrow \mathbf{G}_{(n)} &\simeq \mathbf{A}^{(n)\top} \mathbf{X}_{(n)} (\mathbf{A}^N \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)}) \Rightarrow \mathcal{G} \simeq \mathcal{X} \times_1 \mathbf{A}^{(1)\top} \times_2 \mathbf{A}^{(2)\top} \cdots \times_N \mathbf{A}^{(N)\top} \end{aligned}$$

再讨论 $\mathbf{A}^{(n)}$ 最优解 $\mathbf{A}^{(n)*}$ 的形式。注意到目标函数可作如下变换

$$\begin{aligned} & \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}\|_F^2 \\ &= \|\mathcal{X}\|_F^2 - 2 \langle \mathcal{X}, \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \rangle + \|\mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}\|_F^2 \end{aligned}$$

上式的第二项表示张量内积, 按定义写为元素形式有

$$\begin{aligned} \langle \mathcal{X}, \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \rangle &= \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} x_{i_1 \cdots i_N} \times \left(\sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1 \cdots r_N} \times a_{i_1 r_1} \cdots \times a_{i_N r_N} \right) \\ &= \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1 \cdots r_N} \times \left(\sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} x_{i_1 \cdots i_N} \times a_{r_1 i_1} \cdots \times a_{r_N i_N} \right) \\ &= \langle \mathcal{G}, \mathcal{X} \times_1 \mathbf{A}^{(1)\top} \times_2 \mathbf{A}^{(2)\top} \cdots \times_N \mathbf{A}^{(N)\top} \rangle = \|\mathcal{G}\|_F^2 \end{aligned}$$

再讨论第三项, 同理直接代入上式推导的结论有

$$\begin{aligned} \|\mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}\|_F^2 &= \langle \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}, \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \rangle \\ &= \langle \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_1 \mathbf{A}^{(1)\top} \times_2 \mathbf{A}^{(2)} \times_2 \mathbf{A}^{(2)\top} \cdots \times_N \mathbf{A}^{(N)} \times_N \mathbf{A}^{(N)\top}, \mathcal{G} \rangle \\ &= \langle \mathcal{G} \times_1 (\mathbf{A}^{(1)\top} \mathbf{A}^{(1)}) \times_2 (\mathbf{A}^{(2)\top} \mathbf{A}^{(2)}) \cdots \times_N (\mathbf{A}^{(N)\top} \mathbf{A}^{(N)}), \mathcal{G} \rangle = \|\mathcal{G}\|_F^2 \end{aligned}$$

用几何语言解释上述结论即是——单位正交变换仅是旋转变换, 不会改变长度 (F-范数)。经过上述推导目标函数简化为

$$\arg \min_{\mathbf{A}} \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}\|_F^2 = \arg \min_{\mathbf{A}} \|\mathcal{X}\|_F^2 - \|\mathcal{G}\|_F^2 = \arg \max_{\mathbf{A}} \|\mathcal{G}\|_F^2$$

$$\Rightarrow \mathbf{A}^{(n)*} = \arg \max_{\mathbf{A}^{(n)}} \|\mathbf{A}^{(n)\top} \mathbf{W}\|_F^2, \quad \mathbf{W} = \mathbf{X}_{(n)} (\mathbf{A}^N \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)})$$

写出拉格朗日函数并代入 KKT 条件有

$$L = \|\mathbf{A}^{(n)\top} \mathbf{W}\|_F^2 - \text{tr}(\Lambda (\mathbf{A}^{(n)\top} \mathbf{A}^{(n)} - \mathbf{I})) \Rightarrow \frac{\partial L}{\partial \mathbf{A}^{(n)*}} = 2\mathbf{W}\mathbf{W}^\top \mathbf{A}^{(n)*} - 2\Lambda \mathbf{A}^{(n)*} = 0 \Rightarrow \mathbf{W}\mathbf{W}^\top \mathbf{A}^{(n)*} = \Lambda \mathbf{A}^{(n)*}$$

上式中 Λ 为对角方阵，其对角线元素为拉格朗日乘子。显然 $\mathbf{W}\mathbf{W}^\top \mathbf{A}^{(n)*} = \Lambda \mathbf{A}^{(n)*}$ 意味着 $\mathbf{A}^{(n)*}$ 的每一列向量均为矩阵 $\mathbf{W}\mathbf{W}^\top$ 的特征向量，即 \mathbf{W} 的左奇异矩阵的列向量。又因为目标函数是取 max，则 $\mathbf{A}^{(n)*}$ 是由 \mathbf{W} 的左奇异矩阵的前 R_n 列（对应最大的 R_n 个奇异值）构成的矩阵。上述对 $\mathbf{A}^{(n)*}$ 的推导过程与 PCA 分解（第 15.2.1 节）的推导基本一致，故 Tucker 分解又可视为高维 PCA 分解；

7. 上述最优 Tucker 分解算法被称为高阶正交迭代 (high-order orthogonal iteration, HOOI) 算法。其中正交迭代 (orthogonal iteration) 算法是一种通过迭代计算矩阵前 n 个特征值的算法。

Algorithm 20.3 HOOI 算法

输入: N 阶张量 $X \in \mathbb{R}_{I_1 \times \cdots \times I_N}$, Tucker 分解超参 R_1, \dots, R_N

- 1: 按 HOSVD 算法初始化 $\mathcal{G}, \{\mathbf{A}^{(n)} \in \mathbb{R}_{I_n \times R_n} | n = 1, \dots, N\}$
- 2: **repeat**
- 3: **for** $n = 1, \dots, N$ **do**
- 4: $\mathcal{W} \leftarrow X \times_1 \mathbf{A}^{(1)\top} \cdots \times_{n-1} \mathbf{A}^{(n-1)\top} \times_{n+1} \mathbf{A}^{(n+1)\top} \cdots \times_N \mathbf{A}^{(N)\top}$
- 5: 截取 $\mathbf{W}_{(n)}$ 的左奇异矩阵的前 R_n 列得到 $\mathbf{A}^{(n)}$
- 6: **end for**
- 7: **until** 误差收敛或达到最大迭代次数
- 8: $\mathcal{G} \leftarrow X \times_1 \mathbf{A}^{(1)\top} \times_2 \mathbf{A}^{(2)\top} \cdots \times_N \mathbf{A}^{(N)\top}$

输出: $\mathcal{G}, \{\mathbf{A}^{(n)} \in \mathbb{R}_{I_n \times R_n} | n = 1, \dots, N\}$

与上小节的 CP 分解一致，因为 HOOI 算法本质上也是交替最小二乘 (ALS) 法，故算法同样无法保证收敛到全局最小点，也无法保证是驻点，只能保证收敛到目标函数不再下降为止。

20.5 概率矩阵分解 (Probabilistic matrix factorization, PMF)

1. 概率矩阵分解 (Probabilistic Matrix Factorization, PMF) 算法由 Andriy Mnih 等人于 2007 年提出，适用于大型、稀疏且不平衡的数据集，是目前推荐领域最经典与基础的算法^{6,7,8}；
2. 算法可以视为基于贝叶斯理论优化的因子分解算法（详见第 15.2.2 节）。记数据矩阵 $R_{n \times m}$ ，其中元素 R_{ij} 表示用户 i 对商品 j 的评分。与因子分解、非负矩阵分解等降维算法类似，假设潜在特征数为 d ，则数据矩阵 $R_{n \times m}$ 可近似为两个潜在特征矩阵的乘积

$$R_{n \times m} \approx U_{d \times n}^\top \times V_{d \times m}$$

其中矩阵 $U_{d \times n}, V_{d \times m}$ 分别表示用户潜在特征矩阵和商品潜在特征矩阵，列向量 U_i, V_j 分别表示用户 i 和商品 j 的潜在特征矩阵；

3. PMF 算法主要基于以下两项假设求解潜在特征矩阵 U, V :

- 观测误差 $R_{ij} - U_i^\top V_j$ 服从高斯分布，从而得到似然

$$p(R|U, V, \sigma) = \prod_i \prod_j [N(R_{ij}|U_i^\top V_j, \sigma^2)]^{\mathbb{I}_{ij}}$$

上式中 \mathbb{I}_{ij} 为指示函数，若用户 i 有对商品 j 评分则为 1，反之为 0； $N(R_{ij}|U_i^\top V_j, \sigma^2)$ 表示以 $U_i^\top V_j$ 为期望、 σ 为标准差的正态分布概率密度函数于 R_{ij} 处的概率密度；

⁶概率矩阵分解 (PMF) 及 MovieLens 上的 Python 代码: <https://zhuanlan.zhihu.com/p/34422451>

⁷推荐基础算法之矩阵分解 PMF: https://zhuanlan.zhihu.com/p/268274823?utm_id=0

⁸Probabilistic Matrix Factorization (Mnih, Andriy, and Russ R. Salakhutdinov, 2007): https://proceedings.neurips.cc/paper_files/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf

- 潜在特征矩阵 U, V 服从均值为 0 的高斯先验，则

$$p(U|\sigma_U) = \prod_i N(U_i|0, \sigma_U^2 I), \quad p(V|\sigma_V) = \prod_j N(V_j|0, \sigma_V^2 I)$$

上式中 I 表示单位阵； σ_U, σ_V 分别表示矩阵 U, V 服从的高斯先验的标准差。

- 因为 PMF 算法依托贝叶斯推断框架，故基于极大后验估计方法确定 U, V 。由贝叶斯公式

$$p(U, V|R, \sigma) \propto p(R|U, V, \sigma) \cdot p(U, V|\sigma_U, \sigma_V) = p(R|U, V, \sigma) \cdot p(U|\sigma_U) \cdot p(V|\sigma_V)$$

对上式求极大值，显然有

$$\begin{aligned} \arg \max_{U, V} p(U, V|R, \sigma) &\implies \arg \max_{U, V} \ln p(U, V|R, \sigma) \\ &\implies \arg \min_{U, V} E = \frac{1}{2} \sum_i \sum_j \mathbb{I}_{ij} (R_{ij} - U_i^\top V_j)^2 + \frac{\lambda_U}{2} \sum_i \|U_i\|_2^2 + \frac{\lambda_V}{2} \sum_j \|V_j\|_2^2 \end{aligned}$$

上式中 $\lambda_U = \frac{\sigma^2}{\sigma_U^2}$, $\lambda_V = \frac{\sigma^2}{\sigma_V^2}$ 为算法超参。PMF 算法与 NMF 算法在求解特征矩阵时的一个很大不同是后者假设特征矩阵各元素独立以元素为单位进行优化，而前者则是以列向量（即特征向量）为单位进行优化。基于随机梯度下降算法 (stochastic gradient descent, SGD) 计算上述目标函数 E 关于 U_i, V_j 的梯度进行优化，有

$$\begin{cases} \frac{\partial E}{\partial U_i} = - \sum_j \mathbb{I}_{ij} (R_{ij} - U_i^\top V_j) V_j + \lambda_U U_i \\ \frac{\partial E}{\partial V_j} = - \sum_i \mathbb{I}_{ij} (R_{ij} - U_i^\top V_j) U_i + \lambda_V V_j \end{cases} \implies \begin{cases} U_i \leftarrow U_i + \eta [(R_{ij} - U_i^\top V_j) V_j - \lambda_U U_i] \\ V_j \leftarrow V_j + \eta [(R_{ij} - U_i^\top V_j) U_i - \lambda_V V_j] \end{cases}$$

注意到上式计算目标函数梯度时仅随机使用了一个观测样本 R_{ij} 而非考虑所有样本，这正是随机梯度下降算法的特点；

- 上述经典的基于极大后验估计的 PMF 算法的复杂度取决于正则化超参 λ_U, λ_V 的取值。一般地需要基于待分析数据集特点人工确定超参数取值，在此过程中需要训练大量模型多次调优，在计算上非常昂贵。

20.6 贝叶斯概率矩阵分解 (*Bayesian probabilistic matrix factorization, BPMF*)

- 考虑到经典概率矩阵分解算法效果需要人工确定超参数的缺点，论文作者进一步于 2008 年提出了贝叶斯概率矩阵分解算法 (bayesian probabilistic matrix factorization, BPMF)⁹；
- 算法的解决思路是不止引入模型优化参数 U, V 的先验，同时引入模型超参数的先验，进而最大化模型参数和超参数的后验，从而允许基于训练数据自动控制模型的复杂度。算法之所以得名“贝叶斯概率矩阵分解”，正是因为其在矩阵分解时完全基于贝叶斯方法确定模型参数与超参数。与经典概率矩阵分解相比，算法预测精度显著提高，且同样适用于大规模数据集；
- 与经典 PMF 算法类似，BPMF 算法同样假设潜在特征矩阵 U, V 服从高斯先验，则

$$p(U|\mu_U, \Lambda_U) = \prod_i N(U_i|\mu_U, \Lambda_U^{-1}), \quad p(V|\mu_V, \Lambda_V) = \prod_j N(V_j|\mu_V, \Lambda_V^{-1})$$

将上述高斯分布超参数记为 $\Theta_U = \{\mu_U, \Lambda_U\}$, $\Theta_V = \{\mu_V, \Lambda_V\}$ 。为避免人工确定超参数，假设其服从高斯-威沙特分布（详见第 23.7.3 节）先验

$$p(\Theta_U|\Theta_0) = N(\mu_U|\mu_0, (\beta_0 \Lambda_U)^{-1}) W(\Lambda_U|\nu_0, W_0), \quad p(\Theta_V|\Theta_0) = N(\mu_V|\mu_0, (\beta_0 \Lambda_V)^{-1}) W(\Lambda_V|\nu_0, W_0)$$

⁹论文笔记 Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo (ICML 2008): https://blog.csdn.net/qq_40206371/article/details/120863684

上式中 $W(\cdot)$ 为自由度 ν_0 的威沙特分布概率密度函数, ν_0, μ_0, W_0 可分别取 $d, 0, I$, 其中 d 为潜在特征的数目 (降维维度);

4. 基于上述先验信息, 严格的贝叶斯推断方法应计算参数与超参数的后验分布 $p(U, V, \Theta_U, \Theta_V | R, \Theta_0)$ 并通过极大后验估计确定 U, V, Θ_U, Θ_V 的取值, 从而完成概率矩阵分解。但在引入 Θ_U, Θ_V 的先验后后验分布 $p(U, V, \Theta_U, \Theta_V | R, \Theta_0)$ 的计算与优化复杂度显著增加, 为此 **BPMF** 算法采用 **MCMC** 方法进行数值近似。具体地, 即是生成大量服从后验分布 $p(U, V, \Theta_U, \Theta_V | R, \Theta_0)$ 的样本 $(U^{(t)}, V^{(t)})$, 根据贝叶斯学派的观点, 这些样本均可作为 U, V 的估值, 且基于充分样本还可计算 U, V 的置信区间;
5. Gibbs 采样是现阶段最简单最主流的 MCMC 采样算法 (详见第 27.1 节)。算法无需目标分布的联合概率分布, 仅需基于条件概率分布即可采样得到服从目标分布的样本。显然本节关注的后验联合分布 $p(U, V, \Theta_U, \Theta_V | R, \Theta_0)$ 难以计算, 但各参数、超参的后验条件分布则易于分析, 故基于 Gibbs 采样算法进行采样, 具体流程为:

- 首先确定模型参数初值 $(U^{(1)}, V^{(1)})$, 进而开始 T 轮迭代采样;
- 在第 $t = 1, \dots, T$ 轮迭代中, 基于模型超参后验条件分布 $p(\Theta_U | U^{(t)}, \Theta_0), p(\Theta_V | V^{(t)}, \Theta_0)$ 采样模型超参

$$\Theta_U^{(t)} \sim p(\Theta_U | U^{(t)}, \Theta_0), \quad \Theta_V^{(t)} \sim p(\Theta_V | V^{(t)}, \Theta_0)$$

- 进而基于模型参数后验条件分布采样模型参数

$$U_i^{(t+1)} \sim p(U_i | R, V_i^{(t)}, \Theta_U^{(t)}), \quad V_i^{(t+1)} \sim p(V_i | R, U_i^{(t+1)}, \Theta_V^{(t)})$$

假设马尔可夫链状态转移概率于 t^* 轮迭代后收敛至平稳分布, 则抽样样本 $\{(U^{(t^*)}, V^{(t^*)}), \dots, (U^{(T+1)}, V^{(T+1)})\}$ 均服从后验分布 $p(U, V, \Theta_U, \Theta_V | R, \Theta_0)$;

6. 进一步介绍上述 Gibbs 采样时模型超参与参数的采样方法。首先介绍超参 $(\Theta_U^{(t)}, \Theta_V^{(t)})$ 的采样。因为假设超参服从高斯-威沙特分布先验, 根据高斯-威沙特分布的共轭先验 (conjugate prior) 性质, 超参的后验同样服从高斯-威沙特分布 (详见第 23.7.3 节)。以 Θ_U 为例, 有

$$p(\Theta_U | U^{(t)}, \Theta_0) = N\left(\mu_U | \mu_0^{(t)}, (\beta_0^{(t)} \Lambda_U)^{-1}\right) W\left(\Lambda_U | \nu_0^{(t)}, W_0^{(t)}\right), \quad \beta_0^{(t)} = \beta_0 + n, \quad \nu_0^{(t)} = \nu_0 + n \\ \mu_0^{(t)} = \frac{\beta_0 \mu_0 + n \bar{U}}{\beta_0 + n}, \quad [W_0^{(t)}]^{-1} = W_0^{-1} + \sum_i^n (U_i - \bar{U})(U_i - \bar{U})^\top + \frac{n \beta_0}{n + \beta_0} (\mu_0 - \bar{U})(\mu_0 - \bar{U})^\top$$

上式中 n 为矩阵 U 的列数 (即用户数)。因为后验高斯-威沙特分布的超参已知, 则可采样得到算法超参 Θ_U , 采样方法同样见第 23.7.3 节。对超参 Θ_V 的采样同理;

7. 而后介绍算法参数 U, V 的采样。因为假设参数服从高斯分布先验, 且高斯分布同样具有共轭先验性质, 故参数的后验分布仍服从高斯分布 (详见第 23.7.2 节)。以 U 为例, 有

$$p(U_i | R, V, \Theta_U) = N\left(U_i | \mu_{U,i}^{(t)}, [\Lambda_{U,i}^{(t)}]^{-1}\right) \propto \prod_{j=1}^m [N(R_{ij} | U_i^\top V_j, \alpha^{-1})]^{\mathbb{I}_{ij}} N(U_i | \mu_U, \Lambda_U^{-1})$$

参考第 23.7.2 节推导过程写出 $\mu_{U,i}^{(t)}, \Lambda_{U,i}^{(t)}$ 的表达式。首先关注 $N(U_i | \mu_{U,i}^{(t)}, [\Lambda_{U,i}^{(t)}]^{-1})$ 的指数项

$$N(U_i | \mu_{U,i}^{(t)}, [\Lambda_{U,i}^{(t)}]^{-1}) \propto \exp\left\{-\frac{1}{2} (U_i - \mu_{U,i}^{(t)})^\top \Lambda_{U,i}^{(t)} (U_i - \mu_{U,i}^{(t)})\right\} \propto \exp\left\{-\frac{1}{2} [U_i^\top \Lambda_{U,i}^{(t)} U_i - 2U_i^\top \Lambda_{U,i}^{(t)} \mu_{U,i}^{(t)}]\right\}$$

再关注 $\prod_{j=1}^m [N(R_{ij} | U_i^\top V_j, \alpha^{-1})]^{\mathbb{I}_{ij}} N(U_i | \mu_U, \Lambda_U^{-1})$ 的指数项

$$\prod_{j=1}^m [N(R_{ij} | U_i^\top V_j, \alpha^{-1})]^{\mathbb{I}_{ij}} N(U_i | \mu_U, \Lambda_U^{-1}) \propto \exp\left\{-\frac{1}{2} \left[\sum_j [(R_{ij} - U_i^\top V_j)^\top \alpha (R_{ij} - U_i^\top V_j)]^{\mathbb{I}_{ij}} + (U_i - \mu_U)^\top \Lambda_U (U_i - \mu_U) \right]\right\} \\ \propto \exp\left\{-\frac{1}{2} \left[\sum_j [-2\alpha R_{ij} U_i^\top V_j + \alpha U_i^\top V_j V_j^\top U_i]^{\mathbb{I}_{ij}} + U_i^\top \Lambda_U U_i - 2U_i^\top \Lambda_U \mu_U \right]\right\}$$

$$= \exp \left\{ -\frac{1}{2} \left[U_i^\top \left(\Lambda_U + \alpha \sum_j [V_j V_j^\top]^{I_{ij}} \right) U_i - 2 U_i^\top \left(\Lambda_U \mu_U + \alpha \sum_j [R_{ij} V_j]^{I_{ij}} \right) \right] \right\}$$

对比系数即可得到

$$\Lambda_{U,i}^{(t)} = \Lambda_U + \alpha \sum_j [V_j V_j^\top]^{I_{ij}}, \quad \mu_{U,i}^{(t)} = [\Lambda_{U,i}^{(t)}]^{-1} \left(\Lambda_U \mu_U + \alpha \sum_j [R_{ij} V_j]^{I_{ij}} \right)$$

- 论文中令上式 $\alpha = 2$ 。因为 $\mu_{U,i}^{(t)}$, $\Lambda_{U,i}^{(t)}$ 已知, 即可通过采样得到高斯分布样本 $U_i^{(t+1)}$, 同理也可得到 $V_i^{(t+1)}$;
8. 基于 Gibbs 采样得到服从后验分布 $p(U, V, \Theta_U, \Theta_V | R, \Theta_0)$ 的样本 $\{(U^{(t)}, V^{(t)}) | t = t^*, t^* + 1, \dots, T + 1\}$ 后即完成了概率矩阵分解的过程。其结果不仅可得到潜在特征矩阵 U, V 的置信区间, 也可进一步预测原始数据矩阵 R 的后验分布以计算 R 的置信区间

$$p(R_{ij}^* | R, \Theta_0) \simeq \frac{1}{T - t^* + 2} \sum_{t=t^*}^{T+1} p(R_{ij}^* | U_i^{(t)}, V_j^{(t)})$$

上式实际上也是蒙特卡罗方法近似方法 (详见第 27.1 节)。

20.7 贝叶斯非负矩阵分解 (**Bayesian non-negative matrix factorization, BNMF**)

- 贝叶斯非负矩阵分解 (bayesian non-negative matrix factorization, BNMF) 算法由 Schmidt 等人提出于 2009 年¹⁰。算法本质上是将第 20.6 节介绍的基于完全贝叶斯框架的矩阵分解流程应用于传统的非负矩阵分解问题 (见第 20.2 节)。由此可见贝叶斯推理框架的一个重要优势——传统视角将矩阵分解问题视为运筹优化问题, 意味着对矩阵分解的任意额外限制都可能使得原有优化算法失效, 而在贝叶斯推理框架下则可通过修改先验估计解决不同需求的矩阵分解;
- 对于非负矩阵 $V \in \mathbb{R}_{n \times m}^+$, 非负矩阵分解问题旨在寻找两个同样非负的矩阵 $W \in \mathbb{R}_{n \times d}^+, H \in \mathbb{R}_{d \times m}^+$, 使得 $V \simeq WH$ 。与一般的矩阵分解问题相比增加了对矩阵 W, H 的非负约束。类比一般的贝叶斯概率矩阵分解 (见第 20.6 节), 基于完全贝叶斯推理框架实现非负矩阵分解需要依托以下三项假设:
 - 假设残差 $V - WH$ 独立服从高斯分布, 从而得到似然 $p(V|W, H, \sigma)$

$$p(V|W, H, \sigma) = \prod_{ij} N(V_{ij} | (WH)_{ij}, \sigma^2)$$

此项假设与一般的 BPMF 算法假设一致, 也对应经典的基于高斯误差假设的 NMF 问题;

- 假设参数 W, H 先验服从指数分布

$$p(W) = \prod_{ik} \varepsilon(W_{ik} | \omega_{ik}), \quad p(H) = \prod_{kj} \varepsilon(H_{kj} | \eta_{kj}), \quad \varepsilon(x|\lambda) = \lambda \exp\{-\lambda x\} u(x)$$

上式中 $\varepsilon(\cdot)$ 为指数分布概率密度函数, 其中 $u(\cdot)$ 为单位阶跃函数 (unit step function), 当自变量为正时取 1, 反之为 0。一般的 BPMF 算法假设参数先验服从高斯分布, 故矩阵分解时可能出现负值, 而 BNMF 算法通过假设参数先验服从仅定义于正区间的指数分布以实现 W, H 的非负约束;

- 假设超参 σ^2 先验服从倒伽马分布 (inverse gamma distribution), 即 $1/\sigma^2$ 先验服从伽马分布

$$p(\sigma^2) = G^{-1}(\sigma^2 | k, \theta) = \frac{\theta^k}{\Gamma(k)} (\sigma^2)^{-k-1} \exp\left\{-\frac{\theta}{\sigma^2}\right\}$$

此项假设与一般的 BPMF 算法假设一致 (BPMF 算法中假设超参服从威沙特分布, 即高维场景下的伽马分布, 详见第 23.7.3 节)。

¹⁰Schmidt, M.N., Winther, O., Hansen, L.K. (2009). Bayesian Non-negative Matrix Factorization. In: Adali, T., Jutten, C., Romano, J.M.T., Barros, A.K. (eds) Independent Component Analysis and Signal Separation. ICA 2009. Lecture Notes in Computer Science, vol 5441. Springer, Berlin, Heidelberg. https://link.springer.com/chapter/10.1007/978-3-642-00599-2_68#citeas

3. 融合上述先验信息与似然信息即可得到参数 W, H 的后验分布。与 BPMF 算法一致, BNMF 算法并不直接基于 W, H, σ 的后验联合分布估计 W, H 取值, 而是基于后验条件分布由 Gibbs 采样算法近似估计。首先讨论算法参数 W, H 的后验条件分布, 显然其服从指数修正高斯分布 (exponentially modified gaussian distribution, EMG), 以 W 为例

$$p(W_{ik}|V, W, H, \sigma^2) \propto \prod_j N(V_{ij}|(WH)_{ij}, \sigma^2) \varepsilon(W_{ik}|\omega_{ik}) = N(W_{ik}|\mu_{W_{ik}}, \sigma_{W_{ik}}^2) \varepsilon(W_{ik}|\omega_{ik})$$

指数修正高斯分布是常用的概率分布模型, 大量统计程序内嵌了针对其的采样方法, 故只需确定超参数 $\mu_{W_{ik}}, \sigma_{W_{ik}}^2$ 。首先展开 $N(W_{ik}|\mu_{W_{ik}}, \sigma_{W_{ik}}^2)$ 的指数部分并只保留关于 W_{ik} 的二次项和一次项

$$N(W_{ik}|\mu_{W_{ik}}, \sigma_{W_{ik}}^2) \propto \exp\left\{-\frac{1}{2\sigma_{W_{ik}}^2}(W_{ik} - \mu_{W_{ik}})^2\right\} \propto \exp\left\{-\frac{1}{2\sigma_{W_{ik}}^2}(W_{ik}^2 - 2W_{ik}\mu_{W_{ik}})\right\}$$

进一步展开 $\prod_j N(V_{ij}|(WH)_{ij}, \sigma^2)$ 的指数部分并同样只保留关于 W_{ik} 的二次项和一次项

$$\begin{aligned} \prod_j N(V_{ij}|(WH)_{ij}, \sigma^2) &\propto \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_j \left(V_{ij} - \sum_{k'} W_{ik'} H_{k'j}\right)^2\right)\right\} \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_j \left(\sum_{k'} W_{ik'} H_{k'j}\right)^2 - 2\sum_j V_{ij} \sum_{k'} W_{ik'} H_{k'j}\right)\right\} \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_j (W_{ik} H_{kj})^2 + 2\sum_j W_{ik} H_{kj} \sum_{k' \neq k} W_{ik'} H_{k'j} - 2\sum_j V_{ij} W_{ik} H_{kj}\right)\right\} \\ &= \exp\left\{-\frac{1}{2\sigma^2}\left(W_{ik}^2 \sum_j H_{kj}^2 - 2W_{ik} \left(\sum_j H_{kj} \left(V_{ij} - \sum_{k' \neq k} W_{ik'} H_{k'j}\right)\right)\right)\right\} \end{aligned}$$

由对比系数可知

$$\sigma_{W_{ik}}^2 = \frac{\sigma^2}{\sum_j H_{kj}^2}, \quad \mu_{W_{ik}} = \frac{\sum_j H_{kj} (V_{ij} - \sum_{k' \neq k} W_{ik'} H_{k'j})}{\sum_j H_{kj}^2}$$

最后将上式由元素形式改写为矩阵形式, 由 $\sum_j H_{kj}^2 = (HH^\top)_{kk}$, $\sum_j H_{kj} V_{ij} = (VH^\top)_{ik}$, $\sum_j H_{kj} \sum_{k' \neq k} W_{ik'} H_{k'j} = \sum_{k' \neq k} W_{ik'} (HH^\top)_{k'k} = (WHH^\top)_{ik} - W_{ik} (HH^\top)_{kk}$, 则

$$\sigma_{W_{ik}}^2 = \frac{\sigma^2}{(HH^\top)_{kk}}, \quad \mu_{W_{ik}} = \frac{(VH^\top)_{ik} - (WHH^\top)_{ik} + W_{ik} (HH^\top)_{kk}}{(HH^\top)_{kk}}$$

4. 进一步讨论算法超参 σ^2 的后验条件分布。其中 σ^2 的先验 $p(\sigma^2)$ 服从倒伽马分布, 而似然 $p(V|W, H, \sigma)$ 服从高斯分布, 则后验条件分布 $p(\sigma^2|V, W, H)$ 仍服从倒伽马分布 (高维场景下的威沙特分布具有共轭先验 (conjugate prior) 特性, 故倒伽马分布同样满足该性质)

$$p(\sigma^2|V, W, H) = G^{-1}(\sigma^2|k_{\sigma^2}, \theta_{\sigma^2}) \propto \prod_{ij} N(V_{ij}|(WH)_{ij}, \sigma^2) G^{-1}(\sigma^2|k, \theta)$$

基于对比系数法估计 $k_{\sigma^2}, \theta_{\sigma^2}$ 。展开 $\prod_{ij} N(V_{ij}|(WH)_{ij}, \sigma^2) G^{-1}(\sigma^2|k, \theta)$ 中关于自然系数与 σ 的指数项

$$\prod_{ij} N(V_{ij}|(WH)_{ij}, \sigma^2) G^{-1}(\sigma^2|k, \theta) \propto \left((\sigma^2)^{-k-1} \cdot \prod_{ij} \frac{1}{\sigma}\right) \exp\left\{-\frac{1}{2\sigma^2} \sum_{ij} (V_{ij} - (WH)_{ij})^2 - \frac{\theta}{\sigma^2}\right\}$$

则对比 $G^{-1}(\sigma^2|k_{\sigma^2}, \theta_{\sigma^2})$ 中的相应项有

$$\theta_{\sigma^2} = \frac{1}{2} \sum_{ij} (V_{ij} - (WH)_{ij})^2 + \theta = \theta + \frac{1}{2} \text{tr}\{(V - WH)(V - WH)^\top\}, \quad k_{\sigma^2} = \frac{mn}{2} + k$$

5. 基于上述推导的关于算法参数 W, H 和超参 σ 的后验条件概率分布, 即可基于 Gibbs 采样实现贝叶斯非负矩阵分解。并参考贝叶斯概率矩阵分解中的 Gibbs 采样流程, 在采样 W, H 时也可以潜在特征 (W 的列和 H 的行) 为单位进行采样以提升采样效率。最终得到基于 Gibbs 采样的 BNMF 算法如下

Algorithm 20.4 基于 Gibbs 采样的 BNMF 算法 (不确定性估计)

```

1: for  $t = 1, \dots, T$  do
2:   for  $k = 1, \dots, d$  do
3:      $\mu \leftarrow \left[ (VH^\top)_{:,k} - (WHH^\top)_{:,k} + W_{:,k}(HH^\top)_{kk} \right] / (HH^\top)_{kk}$ 
4:     采样  $W_{:,k}^{(t)} \sim EMG\left(\mu, \frac{\sigma^2}{(HH^\top)_{kk}}, \omega_{:,k}\right)$ 
5:   end for
6:   采样  $\sigma^2 \sim G^{-1}\left(\frac{mn}{2} + k, \theta + \frac{1}{2}\text{tr}\{(V - WH)(V - WH)^\top\}\right)$ 
7:   for  $k = 1, \dots, d$  do
8:      $\mu \leftarrow \left[ (W^\top V)_{k,:} - (W^\top WH)_{k,:} + H_{k,:}(W^\top W)_{kk} \right] / (W^\top W)_{kk}$ 
9:     采样  $H_{k,:}^{(t)} \sim EMG\left(\mu, \frac{\sigma^2}{(W^\top W)_{kk}}, \eta_{k,:}\right)$ 
10:    end for
11:  end for
输出:  $\{W^{(t)}, H^{(t)}|, t = 1, \dots, T\}$ 

```

6. 基于 Gibbs 采样进行贝叶斯非负矩阵分解实际上是采样若干服从 W, H, σ 后验分布的样本 $\{W^{(t)}, H^{(t)}|, t = 1, \dots, T\}$ 。相比传统的确定性非负矩阵分解算法, Gibbs 采样的结果可得到关于 W, H 分布的更充分信息以评估不确定性。然而在某些场景下研究者只需得到 W, H 的确定性估计结果, 在贝叶斯框架中对应极大后验估计。此时 Gibbs 算法即效率偏低, 可替换为迭代条件峰值 (**iterated conditional modes, ICM**) 算法 (详见第 27.1 节);
7. ICM 算法与 Gibbs 采样非常相似, 不同之处在于后者每次迭代时是对参数的条件后验分布进行采样, 而前者则是直接进行极大条件后验估计, 故多次迭代后的估计结果 $\{W, H\}$ 将收敛于 $\{W, H, \sigma\}$ 的联合后验分布的极大值点。故主要难点在于计算 W, H, σ 的极大条件后验估计结果。在本问题中 W, H 的条件后验分布为指数修正高斯分布, σ^2 的条件后验分布为倒伽马分布, 两者的极大后验估计结果均存在封闭形式, 保证了 ICM 算法的计算效率。首先讨论 W_{ik} 的极大条件后验估计

$$\begin{aligned}
 \frac{\partial p(W_{ik}|V, W, H, \sigma^2)}{\partial W_{ik}} = 0 &\implies \frac{\partial}{\partial W_{ik}} [N(W_{ik}|\mu_{W_{ik}}, \sigma_{W_{ik}}^2) \varepsilon(W_{ik}|\omega_{ik})] = 0 \\
 &\implies \frac{\partial}{\partial W_{ik}} \left[\frac{\omega_{ik}}{\sigma_{W_{ik}}} \exp \left\{ -\frac{1}{2\sigma_{W_{ik}}^2} (W_{ik} - \mu_{W_{ik}})^2 - \omega_{ik}W_{ik} \right\} \right] = 0 \\
 &\implies \exp \left\{ -\frac{1}{2\sigma_{W_{ik}}^2} (W_{ik} - \mu_{W_{ik}})^2 - \omega_{ik}W_{ik} \right\} \cdot \left(-\frac{1}{\sigma_{W_{ik}}^2} (W_{ik} - \mu_{W_{ik}}) - \omega_{ik} \right) = 0 \\
 &\implies -\frac{1}{\sigma_{W_{ik}}^2} (W_{ik} - \mu_{W_{ik}}) - \omega_{ik} = 0 \implies W_{ik} = \mu_{W_{ik}} - \omega_{ik}\sigma_{W_{ik}}^2
 \end{aligned}$$

进一步讨论 σ^2 的极大条件后验估计

$$\begin{aligned}
 \frac{\partial p(\sigma^2|V, W, H)}{\partial \sigma^2} = 0 &\implies \frac{\partial}{\partial \sigma^2} [G^{-1}(\sigma^2|k_{\sigma^2}, \theta_{\sigma^2})] = 0 \\
 &\implies \frac{\partial}{\partial \sigma^2} \left[(\sigma^2)^{-k_{\sigma^2}-1} \exp \left\{ -\frac{\theta_{\sigma^2}}{\sigma^2} \right\} \right] = 0 \\
 &\implies (-k_{\sigma^2} - 1) (\sigma^2)^{-k_{\sigma^2}-2} \exp \left\{ -\frac{\theta_{\sigma^2}}{\sigma^2} \right\} + (\sigma^2)^{-k_{\sigma^2}-1} \exp \left\{ -\frac{\theta_{\sigma^2}}{\sigma^2} \right\} \frac{\theta_{\sigma^2}}{\sigma^4} = 0 \\
 &\implies (-k_{\sigma^2} - 1) + \frac{\theta_{\sigma^2}}{\sigma^2} = 0 \implies \sigma^2 = \frac{\theta_{\sigma^2}}{k_{\sigma^2} + 1}
 \end{aligned}$$

最终得到基于 ICM 算法的 BNMF 算法如下, 其中 P_+ 运算符是令小于 0 元素取 0

Algorithm 20.5 基于 ICM 算法的 BNMF 算法 (确定性估计)

1: **repeat**

```

2:   for  $k = 1, \dots, d$  do
3:      $W_{:,k} \leftarrow P_+ \left[ \left[ (VH^\top)_{:,k} - (WHH^\top)_{:,k} + W_{:,k}(HH^\top)_{kk} - \omega_{:,k}\sigma^2 \right] / (HH^\top)_{kk} \right]$ 
4:   end for
5:    $\sigma^2 \leftarrow \frac{\theta + \text{tr}\{(V-WH)(V-WH)^\top\}/2}{mn/2 + k + 1}$ 
6:   for  $k = 1, \dots, d$  do
7:      $H_{k,:} \leftarrow P_+ \left[ \left[ (W^\top V)_{k,:} - (W^\top WH)_{k,:} + H_{k,:}(W^\top W)_{kk} - \eta_{k,:}\sigma^2 \right] / (W^\top W)_{kk} \right]$ 
8:   end for
9: until  $W, H$  收敛
输出:  $\{W, H\}$ 

```

20.8 贝叶斯张量分解 (*Bayesian tensor factorization, BTF*)

20.8.1 贝叶斯 CP 张量分解

- 受到概率矩阵分解算法（第 20.5 节）与贝叶斯概率矩阵分解算法（第 20.6 节）的启发，自 2009 年开始大量研究尝试将贝叶斯推断框架应用于各类矩阵分解及更高维的张量分解问题。其中得到最多关注的张量分解问题即是经典也是与矩阵分解最接近的 CP 张量分解问题（第 20.4 节）；
- 本小节主要讨论最简单的三阶张量的贝叶斯 CP 分解算法¹¹。对于任意三阶张量 $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$ ，CP 分解算法将其分解为三个因子矩阵 (factor matrix) $\mathbf{U} \in \mathbb{R}^{M \times R}, \mathbf{V} \in \mathbb{R}^{N \times R}, \mathbf{X} \in \mathbb{R}^{T \times R}$ ，使得

$$y_{ijt} \simeq \sum_{r=1}^R u_{ir} v_{jr} x_{tr}$$

- 与其它基于完全贝叶斯推理框架实现各类矩阵分解的过程一致，贝叶斯 CP 张量分解依托于对算法误差、参数和超参的各项先验假设：

- 假设残差 $y_{ijt} - \sum_{r=1}^R u_{ir} v_{jr} x_{tr}$ 独立服从高斯分布（与一般的 BPMF 算法假设一致），从而得到似然 $p(\mathcal{Y}|\mathbf{U}, \mathbf{V}, \mathbf{X}, \tau)$

$$p(\mathcal{Y}|\mathbf{U}, \mathbf{V}, \mathbf{X}, \tau) = \prod_{ijt} N\left(y_{ijt} \mid \sum_{r=1}^R u_{ir} v_{jr} x_{tr}, \tau^{-1}\right)$$

- 假设参数 $\mathbf{U}, \mathbf{V}, \mathbf{X}$ 先验服从高斯分布（与一般的 BPMF 算法假设一致）

$$\begin{aligned} p(\mathbf{U}|\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u) &= \prod_i N(\mathbf{u}_i|\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u^{-1}), & p(\mathbf{V}|\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v) &= \prod_j N(\mathbf{v}_j|\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v^{-1}), \\ p(\mathbf{X}|\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x) &= \prod_t N(\mathbf{x}_t|\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x^{-1}) \end{aligned}$$

上式中 $\mathbf{u}_i, \mathbf{v}_j, \mathbf{x}_t$ 分别表示 $\mathbf{U}, \mathbf{V}, \mathbf{X}$ 的列向量。上式实际上是假设同一个因子矩阵的各列先验服从相同的多元高斯分布。但在实际研究中可能出现某一因子矩阵（假设为 \mathbf{X} ）表征时序变化的情况。此时为体现时序演化趋势可设置如下先验

$$p(\mathbf{X}|\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x) = \prod_t N(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\Lambda}_x^{-1}), \quad \mathbf{x}_0 = \boldsymbol{\mu}_x$$

- 假设因子矩阵高斯先验超参 $\boldsymbol{\Theta}_u = \{\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u\}, \boldsymbol{\Theta}_v = \{\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v\}, \boldsymbol{\Theta}_x = \{\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x\}$ 服从高斯-威沙特分布（第 23.7.3 节）先验（与一般的 BPMF 算法假设一致）

$$\begin{aligned} p(\boldsymbol{\Theta}_u|\boldsymbol{\Theta}_0) &= N(\boldsymbol{\mu}_u|\boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_u)^{-1}) W(\boldsymbol{\Lambda}_u|\boldsymbol{\nu}_0, \mathbf{W}_0), & p(\boldsymbol{\Theta}_v|\boldsymbol{\Theta}_0) &= N(\boldsymbol{\mu}_v|\boldsymbol{\nu}_0, (\beta_0 \boldsymbol{\Lambda}_v)^{-1}) W(\boldsymbol{\Lambda}_v|\boldsymbol{\nu}_0, \mathbf{W}_0) \\ p(\boldsymbol{\Theta}_x|\boldsymbol{\Theta}_0) &= N(\boldsymbol{\mu}_x|\boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_x)^{-1}) W(\boldsymbol{\Lambda}_x|\boldsymbol{\nu}_0, \mathbf{W}_0) \end{aligned}$$

¹¹ 教程 | 贝叶斯张量分解: <https://zhuanlan.zhihu.com/p/578395380>

- 假设残差先验超参 τ 服从伽马分布 (**gamma distribution**) 先验 (与一般的 BNMF 算法假设一致)

$$p(\tau) = G(\tau|k, \theta) = \frac{\theta^k}{\Gamma(k)} \tau^{k-1} \exp\{-\theta\tau\}$$

4. 与其它基于完全贝叶斯推理框架实现各类矩阵分解的过程一致, 基于 **Gibbs** 采样算法求解贝叶斯 CP 张量分解问题, 只需再推导算法各参数与超参的条件后验分布。推导细节也基本见前述各节内容:

- 以 \mathbf{U} 为例给出参数 $\mathbf{U}, \mathbf{V}, \mathbf{X}$ 的后验条件分布 $p(\mathbf{u}_i|\mathcal{Y}, \mathbf{V}, \mathbf{X}, \tau, \Theta_u)$ 。易知 \mathbf{U} 的后验条件分布仍为高斯分布, 且有

$$p(\mathbf{u}_i|\mathcal{Y}, \mathbf{V}, \mathbf{X}, \tau, \Theta_u) = N\left(\mathbf{u}_i \middle| \boldsymbol{\mu}_{u,i}, [\Lambda_{u,i}]^{-1}\right), \quad \begin{cases} \boldsymbol{\mu}_{u,i} = [\Lambda_{u,i}]^{-1} \left(\Lambda_u \boldsymbol{\mu}_u + \tau \sum_{jt} y_{ijt} (\mathbf{v}_j * \mathbf{x}_t) \right) \\ \Lambda_{u,i} = \Lambda_u + \tau \sum_{jt} (\mathbf{v}_j * \mathbf{x}_t)(\mathbf{v}_j * \mathbf{x}_t)^\top \end{cases}$$

上式的推导细节可参考第 20.6 节中 BPMF 算法参数后验的推导, 其中 $*$ 为 Hadamard 积, 表示对应元素相乘;

- 以 $\Theta_u = \{\boldsymbol{\mu}_u, \Lambda_u\}$ 为例给出超参 $\Theta_u = \{\boldsymbol{\mu}_u, \Lambda_u\}, \Theta_v = \{\boldsymbol{\mu}_v, \Lambda_v\}, \Theta_x = \{\boldsymbol{\mu}_x, \Lambda_x\}$ 的后验条件分布 $p(\Theta_u|\mathbf{U}, \Theta_0)$ 。易知 Θ_u 的后验条件分布仍为高斯-威沙特分布, 且有

$$p(\Theta_u|\mathbf{U}, \Theta_0) = N\left(\boldsymbol{\mu}_u \middle| \boldsymbol{\mu}_0^{(l)}, (\beta_0^{(l)} \Lambda_u)^{-1}\right) W\left(\Lambda_u \middle| \nu_0^{(l)}, \mathbf{W}_0^{(l)}\right), \quad \beta_0^{(l)} = \beta_0 + M, \quad \nu_0^{(l)} = \nu_0 + M$$

$$\boldsymbol{\mu}_0^{(l)} = \frac{\beta_0 \boldsymbol{\mu}_0 + M \bar{\mathbf{u}}}{\beta_0 + M}, \quad [\mathbf{W}_0^{(l)}]^{-1} = \mathbf{W}_0^{-1} + \sum_i^M (\mathbf{u}_i - \bar{\mathbf{u}})(\mathbf{u}_i - \bar{\mathbf{u}})^\top + \frac{M\beta_0}{M + \beta_0} (\boldsymbol{\mu}_0 - \bar{\mathbf{u}})(\boldsymbol{\mu}_0 - \bar{\mathbf{u}})^\top$$

上式的推导细节可参考第 20.6 节中 BPMF 算法超参后验的推导;

- 最后给出超参 τ 的后验条件分布 $p(\tau|\mathcal{Y}, \mathbf{U}, \mathbf{V}, \mathbf{X}, k, \theta)$ 。易知 τ 的后验条件分布仍为伽马分布, 且有

$$p(\tau|\mathcal{Y}, \mathbf{U}, \mathbf{V}, \mathbf{X}, k, \theta) = G(\tau|k_\tau, \theta_\tau), \quad \theta_\tau = \theta + \frac{1}{2} \sum_{ijt} \left(y_{ijt} - \sum_{r=1}^R u_{ir} v_{jr} x_{tr} \right)^2, \quad k_\tau = k + \frac{M \cdot N \cdot T}{2}$$

上式的推导细节可参考第 20.7 节中 BNMF 算法超参后验的推导。

5. 综上即可得到最基本的三阶张量贝叶斯 CP 分解算法, 整体整理如下

Algorithm 20.6 基于 Gibbs 采样的三阶张量贝叶斯 CP 分解算法

输入: 三阶张量 $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$, CP 分解的秩 R , $\boldsymbol{\mu}_0 = 0, \beta_0 = 1, \mathbf{W}_0 = \mathbf{I}, \nu_0 = R, \theta = k = 10^{-6}$

```

1: 初始化因子矩阵  $\mathbf{U} = 0, \mathbf{V} = 0, \mathbf{X} = 0$ 
2: for  $l = 1, \dots, L_1 + L_2$  do
3:   采样  $\Theta_u^{(l)} \sim p(\Theta_u|\mathbf{U}, \Theta_0), \Theta_v^{(l)} \sim p(\Theta_v|\mathbf{V}, \Theta_0), \Theta_x^{(l)} \sim p(\Theta_x|\mathbf{X}, \Theta_0)$ 
4:   for  $i = 1, \dots, M$  do
5:     采样  $\mathbf{u}_i^{(l)} \sim p(\mathbf{u}_i|\mathcal{Y}, \mathbf{V}^{(l-1)}, \mathbf{X}^{(l-1)}, \tau^{(l-1)}, \Theta_u^{(l)})$ 
6:   end for
7:   for  $j = 1, \dots, N$  do
8:     采样  $\mathbf{v}_j^{(l)} \sim p(\mathbf{v}_j|\mathcal{Y}, \mathbf{U}^{(l)}, \mathbf{X}^{(l-1)}, \tau^{(l-1)}, \Theta_u^{(l)})$ 
9:   end for
10:  for  $t = 1, \dots, T$  do
11:    采样  $\mathbf{x}_t^{(l)} \sim p(\mathbf{x}_t|\mathcal{Y}, \mathbf{U}^{(l)}, \mathbf{V}^{(l)}, \tau^{(l-1)}, \Theta_u^{(l)})$ 
12:  end for
13:  采样  $\tau^{(l)} \sim p(\tau|\mathcal{Y}, \mathbf{U}^{(l)}, \mathbf{V}^{(l)}, \mathbf{X}^{(l)}, k, \theta)$ 
14:  if  $l > L_1$  then
15:     $\hat{\mathbf{U}}_+ = \mathbf{U}^{(l)}, \hat{\mathbf{V}}_+ = \mathbf{V}^{(l)}, \hat{\mathbf{X}}_+ = \mathbf{X}^{(l)}$ 
16:  end if
17: end for
18:  $\mathbf{U} \leftarrow \hat{\mathbf{U}}_+/L_2, \mathbf{V} \leftarrow \hat{\mathbf{V}}_+/L_2, \mathbf{X} \leftarrow \hat{\mathbf{X}}_+/L_2$ 

```

输出: $\mathbf{U}, \mathbf{V}, \mathbf{X}$

20.9 T 分布随机近邻嵌入算法 (**T-Distribution Stochastic Neighbor Embedding**, **t-SNE**)

1. t-SNE 与 PCA 为目前数据科学领域最常用的两类降维算法。不同于已提出近百年的 PCA 算法 (1933)，t-SNE 由深度学习巨擘 Geoffrey Hinton 提出于 2008 年，为近年来最优秀的降维算法^{12,13}。PCA 属于线性算法，难以解释特征之间的复杂多项式关系，而 t-SNE 属于非参数、非线性算法，更便于寻找数据中的结构关系，且除 t-SNE 之外的大多数非线性技术都不能同时保留数据的局部和全局结构；
2. 在可视化时，t-SNE 算法的效果往往比 PCA 更好——相似数据距离更近、不相似数据距离更远，但算法复杂度也远高于 PCA；
3. 实际应用中，t-SNE 很少用于特征工程，而是更多地用于可视化展示，主要原因包括以下几点：
 - 实际特征工程去除的冗余变量往往是高度线性相关的特征，此时由 PCA 等线性降维算法即可实现；
 - 对于存在非线性相关的特征，也可由现有非线性模型直接解决，一般不需要在数据预处理阶段进行降维；
 - 一般特征工程中的降维工作仍要求保留较高的维度特征（如从几十维降到十几维），而 t-SNE 算法采用自由度为 1 的 T 分布，难以达到较好的效果。反之数据可视化要求将高维数据降至三维以下，此时 t-SNE 算法即可更好地保留数据间的结构信息；
 - t-SNE 算法的计算复杂度很高，且目标函数非凸，可能会得到局部最优解。

20.9.1 随机近邻嵌入算法 (SNE)

1. t-SNE 算法针对 SNE 算法的几项缺点做出改进，故首先简要介绍 SNE 算法的基本原理。对于给定 n 个高维数据 x_1, x_2, \dots, x_n ，所有降维算法均旨在将其映射至低维空间下的另一组数据 y_1, y_2, \dots, y_n ，同时保持两组数据之间的一致性——即假设高维数据中 x_i, x_j 点对接近（疏远），则要求低维数据中对应的点对 y_i, y_j 同样接近（疏远）；
2. SNE 算法中使用条件概率 $p_{j|i}$ 表示点对 x_i, x_j 之间的相似性（接近或疏远），定义式如下

$$p_{j|i} = \frac{\exp(-|x_i - x_j|^2)/(2\sigma_i^2)}{\sum_{k \neq i} \exp(-|x_i - x_k|^2)/(2\sigma_i^2)}$$

表示为：假设高维空间中 x_i 的临近点服从以 x_i 为均值、 σ_i 为标准差的正态分布时， x_j 是 x_i 的临近点的概率，显然 x_j 距 x_i 越近则该概率越大，越远则概率越小。计算高维空间中 x_i 与其它所有点的条件概率 $p_{j|i}$ ，($j \neq i$) 后即可得到条件概率分布 P_i ， σ_i 即为 P_i 的标准差；

3. 同样地，对于映射得到的低维空间数据集，也可以采用条件概率 $q_{j|i}$ 表示点对 y_i, y_j 间的相似性

$$q_{j|i} = \frac{\exp(-|y_i - y_j|^2)}{\sum_{k \neq i} \exp(-|y_i - y_k|^2)}$$

为简化计算，直接假设低维空间中标准差 $\sigma_i = \sqrt{2}$ 。同样地可以得到低维空间下的条件概率分布 Q_i ；

4. 进一步地设置目标函数，旨在使得低维空间下的条件概率分布 Q_i 尽可能接近高维空间下的条件概率分布 P_i （完全一致时即是“完美降维”），因而以 KL 散度作为目标函数，表示以概率 Q_i 表示概率 P_i 时造成的信息量增益（详见 23.9.6 节）

$$C = \sum_i KL(P_i || Q_i) = \sum_i \sum_{j \neq i} p_{j|i} \ln \frac{p_{j|i}}{q_{j|i}}$$

5. 对以上目标函数求偏导得下式

$$\frac{\partial C}{\partial y_i} = -\frac{\partial}{\partial y_i} \sum_i \sum_{j \neq i} p_{j|i} \ln q_{j|i}$$

¹²t-SNE 算法 https://blog.csdn.net/sinat_20177327/article/details/80298645

¹³Laurens V D M , Hinton G . Visualizing Data using t-SNE[J]. Journal of Machine Learning Research, 2008, 9(2605):2579-2605. <http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf>

$$= -\frac{\partial}{\partial y_i} \sum_{j,k \neq i} p_{j|k} \ln q_{j|k} - \frac{\partial}{\partial y_i} \sum_{j \neq i} p_{j|i} \ln q_{j|i} - \frac{\partial}{\partial y_i} \sum_{j \neq i} p_{i|j} \ln q_{i|j}$$

又因为

$$\begin{aligned} \frac{\partial}{\partial y_i} \ln q_{j|i} &= \frac{\partial}{\partial y_i} \left(-(y_i - y_j)^2 - \ln \sum_{k \neq i} \exp(-(y_i - y_k)^2) \right) \\ &= -2(y_i - y_j) - \frac{-2 \sum_{k \neq i} (y_i - y_k) \exp(-(y_i - y_k)^2)}{\sum_{k \neq i} \exp(-(y_i - y_k)^2)} = -2 \left[(y_i - y_j) - \sum_{k \neq i} (y_i - y_k) q_{k|i} \right] \\ \frac{\partial}{\partial y_i} \ln q_{i|j} &= \frac{\partial}{\partial y_i} \left(-(y_j - y_i)^2 - \ln \sum_{k \neq j} \exp(-(y_j - y_k)^2) \right) \\ &= -2(y_i - y_j) - \frac{-2(y_i - y_j) \exp(-(y_j - y_i)^2)}{\sum_{k \neq j} \exp(-(y_j - y_k)^2)} = -2(y_i - y_j)(1 - q_{i|j}) \\ \frac{\partial}{\partial y_i} \ln q_{j|k} &= \frac{\partial}{\partial y_i} \left(-(y_k - y_j)^2 - \ln \sum_{l \neq k} \exp(-(y_k - y_l)^2) \right) \\ &= -\frac{-2(y_i - y_k) \exp(-(y_k - y_i)^2)}{\sum_{l \neq k} \exp(-(y_k - y_l)^2)} = 2(y_i - y_k) q_{i|k} \end{aligned}$$

所以有

$$\begin{aligned} \frac{\partial}{\partial y_i} \sum_{j,k \neq i} p_{j|k} \ln q_{j|k} &= 2 \sum_{j,k \neq i} p_{j|k} (y_i - y_k) q_{i|k} = 2 \sum_{k \neq i} (y_i - y_k) q_{i|k} \left(\sum_{j \neq i} p_{j|k} \right) = 2 \sum_{k \neq i} (y_i - y_k) q_{i|k} (1 - p_{i|k}) \\ \frac{\partial}{\partial y_i} \sum_{j \neq i} p_{j|i} \ln q_{j|i} &= -2 \sum_{j \neq i} p_{j|i} \left[(y_i - y_j) - \sum_{k \neq i} (y_i - y_k) q_{k|i} \right] \\ &= -2 \sum_{j \neq i} (y_i - y_j) p_{j|i} + 2 \sum_{k \neq i} (y_i - y_k) q_{k|i} \left(\sum_{j \neq i} p_{j|i} \right) = -2 \sum_{j \neq i} (y_i - y_j) p_{j|i} + 2 \sum_{k \neq i} (y_i - y_k) q_{k|i} \\ \frac{\partial}{\partial y_i} \sum_{j \neq i} p_{i|j} \ln q_{i|j} &= -2 \sum_{j \neq i} (y_i - y_j) p_{i|j} (1 - q_{i|j}) \end{aligned}$$

综上可得目标函数偏导如下

$$\begin{aligned} \frac{\partial C}{\partial y_i} &= -2 \sum_{k \neq i} (y_i - y_k) q_{i|k} (1 - p_{i|k}) + 2 \sum_{j \neq i} (y_i - y_j) p_{j|i} - 2 \sum_{k \neq i} (y_i - y_k) q_{k|i} + 2 \sum_{j \neq i} (y_i - y_j) p_{i|j} (1 - q_{i|j}) \\ &= 2 \sum_{j \neq i} (y_i - y_j) (p_{j|i} + p_{i|j} - p_{i|j} q_{i|j}) - 2 \sum_{k \neq i} (y_i - y_k) (q_{i|k} + q_{k|i} - q_{i|k} p_{i|k}) \\ &= 2 \sum_{j \neq i} (y_i - y_j) (p_{j|i} + p_{i|j} - p_{i|j} q_{i|j}) - 2 \sum_{j \neq i} (y_i - y_j) (q_{i|j} + q_{j|i} - q_{i|j} p_{i|j}) \\ &= 2 \sum_{j \neq i} (y_i - y_j) (p_{j|i} + p_{i|j} - q_{i|j} - q_{j|i}) \end{aligned}$$

6. 以上即为 SNE 算法的理论基础。简而言之该算法存在如下缺点：

- 条件概率不对称造成算法计算复杂度高。观察目标函数导数可知，算法需计算条件概率 $p_{j|i}, p_{i|j}, q_{i|j}, q_{j|i}$ ，而因为 $p_{j|i} \neq p_{i|j}, q_{i|j} \neq q_{j|i}$ ，当数据量偏大时将带来较大的计算资源消耗；
- KL 散度不对称造成算法更关注数据的局部结构。观察目标函数，当点对 x_i, x_j 非常接近，即 $p_{j|i}$ 较大时，目标函数对 $q_{j|i}$ 敏感，但当 x_i, x_j 相距较远， $p_{j|i}$ 趋于 0 时，此时 $q_{j|i}$ 的大小对目标函数的影响将非常小，即算法对数据的局部结构敏感而对全局结构不敏感。该特点也可从目标函数导数总结得出，该导数由两项组成：一项比较低维数据点对间的距离，另一项比较高维数据分布与低维数据分布的一致性，当高维数据点对 x_i, x_j 较为接近时，要求 y_i, y_j 同样接近且高低维数据分布一致，此时组成导函数的两项均有助于梯度收敛，但当高维数据点对 x_i, x_j 较远时，此时 y_i, y_j 不可能过分接近，仅能通过优化高低维数据分布一致性实现梯度收敛，因而无法保证充分的优化效果。

20.9.2 对称随机近邻嵌入算法 (Symmetric SNE)

- 对称 SNE 算法同样以 KL 散度作为误差函数，旨在解决 SNE 算法的第一个缺点，在一定程度上降低了算法计算复杂度；
- 对称 SNE 以联合概率替代条件概率，此时有 $p_{ij} = p_{ji}$, $q_{ij} = q_{ji}$ 。简单地，定义低维空间下 q_{ij} 如下

$$q_{ij} = \frac{\exp(-|y_i - y_j|^2)}{\sum_{k \neq l} \exp(-|y_k - y_l|^2)}$$

- 同样地也可以定义高维空间下的 p_{ij} 如下

$$p_{ij} = \frac{\exp(-|x_i - x_j|^2 / \sigma^2)}{\sum_{k \neq l} \exp(-|x_k - x_l|^2 / \sigma^2)}$$

但该定义难以处理存在离群值的情况：假设高维数据集中存在离群值 x_i ，则对于任意 j , p_{ij} 均接近 0，此时因为 KL 散度的不对称性使得误差函数同样接近 0，造成降维时难以保留对应数据点特征（类似 SNE 算法易忽视数据全局结构）。故定义 p_{ij} 如下

$$p_{ij} = \frac{p_{ilj} + p_{jli}}{2}$$

- 此时算法的误差梯度变为

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (y_i - y_j)(p_{ij} - q_{ij})$$

- 可以看到，相比于原始 SNE，对称 SNE 梯度更简单，计算复杂度更低，但因为对称 SNE 未解决原始 SNE 易忽略数据全局结构的问题，其效果仅略微优于原始 SNE。

20.9.3 T 分布随机近邻嵌入算法 (t-SNE)

- t-SNE 在对称 SNE 的基础上进一步引入 t 分布，从而解决了原始 SNE 难以兼顾数据全局特征的缺陷。为说明 t 分布引入的必要性，先介绍降维算法中的“拥挤问题 (crowding problem)”；
- 假设在 n 维空间中，数据点均匀地分布在半径为 R 的超球内部，则数据点距离超球球心的平均距离 d_n

$$d_n = \frac{\underbrace{\int \cdots \int}_{x_1^2 + \cdots + x_n^2 \leq R^2} \sqrt{x_1^2 + \cdots + x_n^2} dx_1 \cdots dx_n}{\underbrace{\int \cdots \int}_{x_1^2 + \cdots + x_n^2 \leq R^2} dx_1 \cdots dx_n}$$

基于第 23.10 节所介绍的超球坐标系积分变换可变换 d_n 表达式如下

$$d_n = \frac{\int_0^R r^n dr \int_0^\pi \sin^{n-2} \varphi_1 d\varphi_1 \int_0^\pi \sin^{n-3} \varphi_2 d\varphi_2 \cdots \int_0^\pi \sin \varphi_{n-2} d\varphi_{n-2} \int_0^{2\pi} d\varphi_{n-1}}{\int_0^R r^{n-1} dr \int_0^\pi \sin^{n-2} \varphi_1 d\varphi_1 \int_0^\pi \sin^{n-3} \varphi_2 d\varphi_2 \cdots \int_0^\pi \sin \varphi_{n-2} d\varphi_{n-2} \int_0^{2\pi} d\varphi_{n-1}} = \frac{n}{n+1} R$$

可以看到在高维空间中均匀分布的数据，其与分布中心的距离并不呈均匀分布，且随着维度增加数据集距分布中心的平均距离也随之增加，这一现象即称为“拥挤问题”；

- 回顾 SNE 算法中关于 p_{ij}, q_{ij} 的定义：假设 $x_i, (y_i)$ 周边的数据服从以 $x_i, (y_i)$ 为中心的正态分布时 $x_j, (y_j)$ 落入 $x_i, (y_i)$ 邻域的概率。可以看到算法对高维数据与低维数据均以正态分布构建数据点间的相似性，而拥挤问题又指出高维状态下将有更多的数据向尾部集中，因此 SNE 算法无法解决高维拥挤问题，这也是算法难以保留数据全局结构的根本原因；
- 与正态分布不同，t 分布数据重尾分布，允许更多地数据分布于外围。自由度为 n 的 t 分布概率密度函数如下

$$f(x) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})} \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}$$

t-SNE 算法在定义低维数据 q_{ij} 时引入自由度为 1 的 t 分布，具体定义如下

$$q_{ij} = \frac{(1 + |x_i - x_j|^2)^{-1}}{\sum_{k \neq l} (1 + |x_k - x_l|^2)^{-1}}$$

同时算法保留高维数据 p_{ij} 定义时关于正态分布的假设，此时当高维数据向低维空间映射时低维的重尾分布可更好地拟合高维数据的拥挤情况；

5. 综上所述，t-SNE 算法的误差梯度计算如下

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (y_i - y_j) (1 + |y_i - y_j|^2)^{-1} (p_{ij} - q_{ij})$$

可以看到此时误差梯度中不仅有 $y_i - y_j$ 项，也有 $y_i - y_j$ 的倒数项，故无论 x_i, x_j 相距远近均有一项有助于梯度收敛，说明经过两项优化后的 t-SNE 算法可很好地反映数据的局部特征和宏观特征。

第21章

信号处理

21.1 信号与系统概述

1. 信息传输的任务：将带有信息的信号，通过某种系统由发送者传送给接受者。而系统在其间的作用即是对带有信息的信号进行适当的处理和变换。

2. 以通信系统为例，其组成如下：



以上各个组成部分均可视为一个系统

3. 信号是一个随时间变化的物理量，可通过时域法描述 $f(t)$ ，也可通过频域法表示，即通过正交变换（如傅里叶变换 $F(\omega) = \mathcal{F}[f(t)]$ ）将信号表示成其它变量的函数；

4. 信号 $f(t)$ 的总能量 $U = \lim_{T \rightarrow \infty} \int_{-T}^T f(t)^2 dt$ 、平均功率 $\bar{P} = \lim_{T \rightarrow \infty} 1/(2T) \cdot \int_{-T}^T f(t)^2 dt$ ，由此可将信号分为能量信号和功率信号：

- (a) 能量信号：总能量有限的信号；
- (b) 功率信号：平均功率有限且非零的信号。

5. 信号的简单变换：

- (a) 时移： $f(t) \rightarrow f(t - t_0)$ （左加右减）；
- (b) 反褶： $f(t) \rightarrow f(-t)$ ；
- (c) 尺度变换： $f(t) \rightarrow f(\omega t)$ ($|\omega| > 1$ 尺度缩小)；
- (d) 标量乘法： $f(t) \rightarrow af(t)$

6. 系统是由若干互相联系的单元组成的、具有某种功能、用以达到某种目的的有机整体；

7. 系统的描述方法（一般均针对其输入与输出之间的关系而建立）：

- (a) 输入输出方程： $r(t) = f[e(t)]$ ，其中 $e(t), r(t)$ 分别为输入和输出信号；
- (b) 框图模型；

8. 线性系统：同时符合叠加性和齐次性的系统

- (a) 齐次性：假设系统 $e(t) \rightarrow r(t)$ ，若系统满足 $k \cdot e(t) \rightarrow k \cdot r(t)$ ，则称系统满足齐次性；
- (b) 叠加性：假设系统 $e_1(t) \rightarrow r_1(t)$ 且 $e_2(t) \rightarrow r_2(t)$ ，若系统满足 $e_1(t) + e_2(t) \rightarrow r_1(t) + r_2(t)$ ，则称系统满足叠加性。

9. 增量线性系统：对系统 $r(t) = e(t) + C$ ，显然其不属于线性系统，但其输入信号和输出信号的增量 $\Delta e(t)$ 和 $\Delta r(t)$ 满足齐次性和叠加性，故称为增量线性系统；

10. 非时变系统：如果系统有 $e(t) \rightarrow r(t)$ ，同时满足 $e(t - t_0) \rightarrow r(t - t_0)$ ，则为非时变系统；

11. 零输入响应和零状态响应：

- (a) 零输入响应：系统在输入信号为 0 的情况下仅由其初始储能导致的响应；

(b) 零状态响应: 系统在初始储能为 0 的情况下仅由输入信号导致的响应。

12. 乘法器: $r(t) = e_1(t) \cdot e_2(t)$;

13. 用微分算子简化微分符号的表示: $p \rightarrow \frac{d}{dt}$, $p^2 \rightarrow \frac{d^2}{dt^2}$, $\frac{1}{p} \rightarrow \int_{-\infty}^t dt$, 此时可将系统的输入输出信号 $e(t), r(t)$ 的关系表示为 $r(t) = H(p)e(t)$, 其中 $H(p)$ 称为转移算子, 为有理分式:

$$H(p) = \frac{N(p)}{D(p)} = \frac{b_m p^m + b_{m-1} p^{m-1} + \cdots + b_1 p + b_0}{p^n + a_{n-1} p^{n-1} + \cdots + a_1 p + a_0}$$

21.2 连续时间系统的时域分析

1. 线性连续时间系统的时域分析就是一个建立和求解线性微分方程的过程。可采用经典法直接求解线性系统对信号的具体响应 (通解/自然响应 + 特解/受迫响应)。但为降低求解难度, 更多地将响应 (微分方程) 分成零输入响应和零状态响应两部分分别求解, 此时总响应 $r(t) = r_1(t) + r_2(t)$:

(a) 零状态响应 $r_1(t)$: 微分方程不变, 初始条件 $r_1^{(n-1)}(0) = r_1^{(n-2)}(0) = \cdots = r_1(0) = 0$;

(b) 零输入响应 $r_2(t)$: 初始条件不变, 微分方程 $a_{n-1}r_2^{(n-1)}(t) + a_{n-2}r_2^{(n-2)}(t) + \cdots + a_0r_2(t) = 0$;

2. 零输入响应 $r_2(t)$ 可由经典法直接求解, 此时为求解一个带初始条件的线性常系数其次微分方程, 可列特征方程快速求解; 零状态响应 $r_1(t)$ 采用卷积法进行求解, 其基本思想为:

(a) 将输入信号 $e(t)$ 分解为一系列“简单”子信号 $e_i(t)$ 的叠加 (和、级数、积分);

(b) 计算线性系统对子信号 $e_i(t)$ 的响应 $r_{1i}(t)$ 并叠加。

3. 在时域分析法中, 以奇异函数¹作为子信号, 常用的奇异函数有阶跃函数 $\varepsilon(t)$ ² 和冲激函数 $\delta(t)$:

$$\delta(t) = \frac{d}{dt}\varepsilon(t); \quad \varepsilon(t) = \int_{-\infty}^t \delta(\tau)d\tau; \quad \varepsilon(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}; \quad \int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt = f(t_0)$$

4. 由阶跃函数 $\varepsilon(t)$ 分解任意有始函数³ $f(t)$ 的方法为:

易知 $f(n\Delta t) = f(0) + [f(\Delta t) - f(0)] + \cdots + [f(n\Delta t) - f(n\Delta t - \Delta t)]$, 令

$$\begin{aligned} f_0(t) &= f(0)\varepsilon(t) \\ f_1(t) &= [f(\Delta t) - f(0)]\varepsilon(t - \Delta t) = \frac{f(\Delta t) - f(0)}{\Delta t}\varepsilon(t - \Delta t)\Delta t \\ &\dots \\ f_n(t) &= [f(n\Delta t) - f(n\Delta t - \Delta t)]\varepsilon(t - n\Delta t) = \frac{f(n\Delta t) - f(n\Delta t - \Delta t)}{\Delta t}\varepsilon(t - n\Delta t)\Delta t \end{aligned}$$

则有 $f(n\Delta t) = \sum_{i=0}^n f_i(t)$, 令 $\Delta t \rightarrow 0$, 此时 $n\Delta t = t$, 有

$$f(t) = f(0)\varepsilon(t) + \int_0^t f'(\tau)\varepsilon(t - \tau)d\tau \tag{21.1}$$

如果 $f(t)$ 在 $t = 0$ 处连续可导, 即有 $f(t) = \int_0^t f'(\tau)\varepsilon(t - \tau)d\tau$

5. 由冲激函数 $\delta(t)$ 分解任意有始函数 $f(t)$ 的方法为:

由冲激函数性质, 对 $\forall t_0 \in [0, t]$, 有 $f(t_0) = \int_{-\infty}^{\infty} f(t)\delta(t - t_0)dt = \int_0^{t_0} f(t)\delta(t - t_0)dt$, 变换有

$$f(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t - \tau)d\tau = \int_0^t f(\tau)\delta(t - \tau)d\tau \tag{21.2}$$

式21.2不要求 $f(t)$ 连续可导, 故较式21.1有更大的适用范围。

¹ 奇异函数: 一种理想化函数, 具有一个或多个间断点, 在这些点上无法确定其具体的函数值或导数值。

² 阶跃函数也长写作 $u(t)$ (unit function)

³ 有始函数: $t < 0$ 时 $f(t) = 0$; $t > 0$ 时 $f(t) \neq 0$

6. 假设线性时不变系统对单位冲击信号 $\delta(t)$ 的零状态响应为 $h(t)$, $h(t)$ 称为冲激响应, 则根据系统的时不变性和齐次性, 有

$$e(\tau)\delta(t-\tau) \xrightarrow{\text{零状态}} e(\tau)h(t-\tau)$$

再根据系统的叠加性有

$$e(t) \xrightarrow{\text{零状态}} r(t) \Rightarrow \int_{-\infty}^{\infty} e(\tau)\delta(t-\tau)d\tau \xrightarrow{\text{零状态}} \int_{-\infty}^{\infty} e(\tau)h(t-\tau)d\tau \Rightarrow r(t) = \int_{-\infty}^{\infty} e(\tau)h(t-\tau)d\tau$$

称系统对 $e(t)$ 的零状态响应 $r(t)$ 为 $e(t)$ 和 $h(t)$ 的卷积积分, 为零状态响应的一般形式, 记作 $r(t) = e(t) \otimes h(t)$ 。卷积运算具体法则及性质见第 23.2 节。在一般情况下输入信号 $e(t)$ 均为有始信号, 有 $e(t) = 0, t < 0$; 且系统多为因果系统, 有 $h(t) = 0, t < 0$ (表示系统响应在冲激之前应为 0)。则 $r(t)$ 的表达式也可写为

$$r(t) = \int_0^t e(\tau)h(t-\tau)d\tau$$

上式的应用条件是“有始信号作用于因果系统”。综上, 只需求解冲激响应 $h(t)$, 即可得到系统的零状态响应, 再结合系统的零输入响应, 即能得到系统的总体响应。

21.2.1 冲激响应的求解

1. 首先考虑一阶线性时不变系统 $r'(t) - \lambda r(t) = k e(t), \lambda$ 为分母多项式的特征根, 则 $\delta(t)$ 与 $h(t)$ 满足 $h'(t) - \lambda h(t) = k \delta(t)$ 或由微分算子表示为 $h(t) = k \delta(t)/(p - \lambda)$, 微分方程两边同时配凑 $e^{-\lambda t}$ 并结合求导的乘法法则有:

$$\begin{aligned} e^{-\lambda t} h'(t) - \lambda e^{-\lambda t} h(t) &= k \delta(t) e^{-\lambda t} \Rightarrow [e^{-\lambda t} h(t)]' = k \delta(t) e^{-\lambda t} \\ &\Rightarrow \int_{0^-}^t [e^{-\lambda \tau} h(\tau)]' d\tau = \int_{0^-}^t k \delta(\tau) e^{-\lambda \tau} d\tau \\ &\Rightarrow e^{-\lambda t} h(t) - h(0) = k e^{-\lambda t} \end{aligned}$$

由零状态响应的初始条件, $h(0) = 0$, 有

$$h(t) = \frac{k}{p - \lambda} \delta(t) = k e^{\lambda t} \varepsilon(t)$$

2. 考虑高阶线性微分方程, 可将有理分式 $H(p)$ 因式分解为若干个简单分式的和, 不同类型的简单分式和冲激函数 $\delta(t)$ 的乘积如下

$$\frac{k}{p - \lambda} \delta(t) = k e^{\lambda t} \varepsilon(t) \quad \frac{k}{(p - \lambda)^n} \delta(t) = k \frac{t^{n-1}}{(n-1)!} e^{\lambda t} \varepsilon(t) \quad k p^n \delta(t) = k \delta(t)^{(n)}$$

21.3 傅里叶级数

1. 在时域分析法中, 信号 $f(t)$ 被分解为多个冲激或阶跃子信号并求响应, 实际上还可以选择其它子信号进行分解。在频域分析法中, 便将任意信号分解为多个正弦或余弦子信号, 通过求系统对简谐信号的响应求对其它信号的响应。

2. 以下以矢量的正交分解为例, 介绍函数的正交分解:

- (a) 对任意矢量 \mathbf{A} 及某一固定矢量 \mathbf{A}_1 , 必然存在系数 c_1 及误差项 $\boldsymbol{\varepsilon}$ 使得 \mathbf{A} 被 \mathbf{A}_1 拟合: $\mathbf{A} = c_1 \mathbf{A}_1 + \boldsymbol{\varepsilon}$ 。当 $|\boldsymbol{\varepsilon}|^2$ 最小时, 认为 \mathbf{A}_1 的拟合效果最好, 此时

$$c_1 = \frac{\mathbf{A}_1 \cdot \mathbf{A}}{\mathbf{A}_1 \cdot \mathbf{A}_1}$$

c_1 称为矢量 \mathbf{A} 和 \mathbf{A}_1 的相似系数, 当 $c_1 = 0$ 时称 \mathbf{A} 和 \mathbf{A}_1 正交。

对任意矢量 \mathbf{A} 及一组矢量标准正交基 $\{\mathbf{A}_1, \dots, \mathbf{A}_n\}$, 则 \mathbf{A} 与对应的相似系数 c_i 可被唯一地表示:

$$\mathbf{A} = c_1 \mathbf{A}_1 + \dots + c_n \mathbf{A}_n \quad c_i = \frac{\mathbf{A}_i \cdot \mathbf{A}}{\mathbf{A}_i \cdot \mathbf{A}_i}$$

矢量标准正交基要求满足以下几个条件 (第 1 项和第 3 项不要求完全满足):

- i. 归一化：标准矢量的模为 1；
 - ii. 正交化：标准矢量两两正交；
 - iii. 完备性：可以不失真地组合出任意矢量。
- (b) 同样地，对任意函数 $f(t)$ 及某一固定函数 $f_1(t)$ ，必然存在系数 c_1 及误差项 ε 使得 $f(t)$ 被 $f_1(t)$ 拟合：
 $f(t) = c_1 f_1(t) + \varepsilon$ 。当方均误差 $\frac{1}{t_1-t_2} \int_{t_2}^{t_1} \varepsilon^2 dt$ 最小时，认为 $f_1(t)$ 在区间 $[t_2, t_1]$ 内的拟合效果最好，此时

$$c_1 = \frac{\int_{t_2}^{t_1} f_1(t) f(t) dt}{\int_{t_2}^{t_1} f_1(t) f_1(t) dt}$$

c_1 称为函数 $f(t)$ 和 $f_1(t)$ 的相似系数，当 $c_1 = 0$ 时 $f(t)$ 和 $f_1(t)$ 正交。当 $f(t)$ 和 $f_1(t)$ 为复函数，则上式如下，其中 $\overline{f_1(t)}$ 表示 $f_1(t)$ 的共轭

$$c_1 = \frac{\int_{t_2}^{t_1} \overline{f_1(t)} f(t) dt}{\int_{t_2}^{t_1} \overline{f_1(t)} f_1(t) dt}$$

对任意函数 $f(t)$ 及一组函数基 $\{f_1(t), \dots, f_n(t)\}$ ，可以证明，当函数基为标准正交基，即满足（第 1 项和第 3 项不要求完全满足）：

- i. 归一化： $\int_{t_2}^{t_1} \overline{f_i(t)} f_i(t) dt = 1$ ；
- ii. 正交化： $\int_{t_2}^{t_1} \overline{f_i(t)} f_j(t) dt = 0, i \neq j$ ；
- iii. 完备性：可以不失真地由线性组合表示任意函数。

则 $f(t)$ 可分解为 $\{f_1(t), \dots, f_n(t)\}$ ：

$$f(t) = c_1 f_1(t) + \dots + c_n f_n(t) = \sum_1^n c_i f_i(t) \quad c_i = \frac{\int_{t_2}^{t_1} \overline{f_i(t)} f(t) dt}{\int_{t_2}^{t_1} \overline{f_i(t)} f_i(t) dt}$$

- (c) 可以证明，前述的冲激函数即是一组正交函数，而三角函数 $1, \cos t, \sin t, \dots, \cos kt, \sin kt$ 同样相互正交。

3. 对周期为 T_1 、频率 $\omega_1 = 2\pi/T_1$ 的周期函数 $f(t)$ ，若在一个周期内 $f(t)$ 满足狄利赫里 (Dirichlet) 条件：

- (a) 间断点个数有限
- (b) 极大值和极小值个数有限
- (c) 绝对可积

则 $f(t)$ 可展开为傅里叶级数形式，即分解为无数个正弦函数和余弦函数之和，式中 a_0, a_n, b_n 分别称为 $f(t)$ 的直流分量、n 次谐波的余弦分量幅度和正弦分量幅度。 a_n 是关于 n 的偶函数，而 b_n 是关于 n 的奇函数。

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_1 t + b_n \sin n\omega_1 t)$$

$$a_0 = \frac{1}{T_1} \int_{-\frac{T_1}{2}}^{\frac{T_1}{2}} f(t) dt, \quad a_n = \frac{2}{T_1} \int_{-\frac{T_1}{2}}^{\frac{T_1}{2}} f(t) \cos n\omega_1 t dt, \quad b_n = \frac{2}{T_1} \int_{-\frac{T_1}{2}}^{\frac{T_1}{2}} f(t) \sin n\omega_1 t dt$$

4. 对上式，由和差化积公式可转化为仅含余弦函数的表达式。 c_n 是关于 n 的偶函数， ϕ_n 是关于 n 的奇函数。显然，只需要确定 c_n 与 ϕ_n 即可确定整个周期信号。

- (a) 幅度 c_n 与频率 $n\omega_1$ 的关系称为信号的幅度频谱，简称幅度谱，即以频率 $n\omega_1$ 为横坐标，幅度 c_n 为纵坐标所做的图。
- (b) 相位 ϕ_n 与频率 $n\omega_1$ 的关系称为信号的相位频谱，简称相位谱，以 2π 为周期
- (c) 幅度谱和相位谱构成周期信号的频谱函数，简称频谱。频谱具有离散性、谐波性、收敛性。

$$f(t) = c_0 + \sum_{n=1}^{\infty} c_n \cos(n\omega_1 t + \phi_n)$$

$$c_n = \sqrt{a_n^2 + b_n^2}, \quad \phi_n = \arctan \left(-\frac{b_n}{a_n} \right)$$

5. 对傅里叶级数，还可将其表示为复指数形式。 $F(n\omega_1)$ 称为周期信号的复数频谱，简称复数谱。

$$\begin{aligned} f(t) &= a_0 + \sum_{n=1}^{\infty} \left(\frac{a_n - ib_n}{2} e^{in\omega_1 t} + \frac{a_n + ib_n}{2} e^{-in\omega_1 t} \right) \\ \because a_{-n} &= a_n, \quad b_{-n} = -bn \\ \therefore f(t) &= a_0 + \sum_{n=1}^{\infty} \frac{a_n - ib_n}{2} e^{in\omega_1 t} + \sum_{n=-\infty}^{-1} \frac{a_n - ib_n}{2} e^{in\omega_1 t} = \sum_{n=-\infty}^{\infty} F(n\omega_1) e^{in\omega_1 t} \end{aligned}$$

由直角坐标与极坐标的转换关系，复数可用模与辐角表示，其中模称为复数谱的幅度谱，辐角称为复数谱的相位谱：

$$F(n\omega_1) = \frac{a_n - ib_n}{2} = \frac{1}{T_1} \int_{-\frac{T_1}{2}}^{\frac{T_1}{2}} f(t) e^{-in\omega_1 t} dt = |F(n\omega_1)| e^{i\phi_n}$$

$$|F(n\omega_1)| = \frac{1}{2} \sqrt{a_n^2 + b_n^2} = \frac{1}{2} c_n, \quad \phi_n = \arctan \left(-\frac{b_n}{a_n} \right)$$

6. 讨论信号的功率（或能量）。假设信号 $f(t)$ 傅里叶展开为 $f(t) = \sum c_i f_i(t)$ ，易知周期信号的功率等于其一个周期内的功率

$$P = \frac{1}{t_1 - t_2} \int_{t_2}^{t_1} f^2(t) dt = \frac{1}{t_1 - t_2} \int_{t_2}^{t_1} \left[\sum c_i f_i(t) \right]^2 dt$$

因为 $\{f_i(t) | i = 0, 1, \dots, n\}$ 为正交函数集，有

$$P = \frac{1}{t_1 - t_2} \int_{t_2}^{t_1} \left[\sum c_i f_i(t) \right]^2 dt = \frac{1}{t_1 - t_2} \int_{t_2}^{t_1} \sum [c_i f_i(t)]^2 dt = \sum \frac{1}{t_1 - t_2} \int_{t_2}^{t_1} [c_i f_i(t)]^2 dt = \sum P_i$$

上式即 **Parseval 定理**：信号的功率等于信号在完备正交函数集中分解后各子信号功率的和。同理，信号与正交分解下子信号的能量也具有相似的关系。

Gibbs 现象

前文说明，对任意满足 Dirichlet 条件的信号都可按上述公式进行傅里叶级数展开。然而对于存在跳变点但满足 Dirichlet 条件的信号，进行傅里叶级数展开时将存在 Gibbs 现象：随展开项趋于无穷，在函数的间断点附近至少存在一点，其函数的分解误差收敛于函数在这点上的跳变值的 **8.948987%**。

需要说明的是，Gibbs 现象与 Dirichlet 条件并不矛盾，这其间涉及到级数逼近概念中的“逐点收敛”和“均方收敛”两个概念之间的差异。傅里叶级数展开的公式是基于均方误差收敛的目标得到的，不一定满足逐点收敛的要求。

信号的频带

由于信号的收敛性，一般可以在一个信号分量主要集中的频率区间内研究信号的特性，而忽略信号的其它分量，响应的频率区间就是信号的频带。信号的频带有多种定义方法：

1. 以子信号最大幅度的 1/10 为限，其它部分忽略；
2. 以信号振幅频谱中的第一个过零点为限，其它部分忽略；
3. 以包含信号总能量的 90% 为限，其它部分忽略。

信号的频带受信号时间宽度和信号边沿变化速度影响：时间宽度越窄、边沿变化速度越快，信号频带越宽，表示能量向高频扩散。

21.4 傅里叶变换

21.4.1 非周期信号的傅里叶变换

1. 对于非周期信号，当假设其为 $T_1 = \infty$ 的周期信号时，采用傅里叶级数方法分析，有：

$$\because T_1 \rightarrow \infty, \quad F(n\omega_1) = \frac{1}{T_1} \int_{-\frac{T_1}{2}}^{\frac{T_1}{2}} f(t) e^{-in\omega_1 t} dt, \quad \omega_1 T_1 = 2\pi, \quad \text{且要求 } f(t) \text{ 在周期内绝对可积}$$

$$\therefore \omega_1 = 0, \quad F(n\omega_1) = 0$$

$$\therefore |F(n\omega_1)| = 0, \quad \Delta(n\omega_1) = \omega_1 = 0$$

频谱幅度趋于 0，由离散趋于连续，但对振幅的收敛性没有影响。

2. 显然，对非周期信号展开为傅里叶级数在数学上是可行的。但对于两个完全不同的信号，其幅度谱均为无穷小，此时无法从频域进行区分，为此需要引入傅里叶变换，即将趋于无穷小的频谱放大为有限值。

具体概念可参考概率论中概率与概率密度的关系，对于一连续变量，其等于任意定值的概率均为无穷小，但其在不同点上的概率密度则可被区分。

3. 将傅里叶级数两边同乘 T_1 ：

$$F(n\omega_1)T_1 = \int_{-\frac{T_1}{2}}^{\frac{T_1}{2}} f(t)e^{-in\omega_1 t} dt$$

$$\because T_1 \rightarrow \infty$$

$$\therefore \Delta(n\omega_1) = 0 \implies n\omega_1 = \omega$$

令 $F(\omega) = F(n\omega_1)T_1 = \lim_{\omega_1 \rightarrow 0} 2\pi F(n\omega_1)/\omega_1 = \lim_{\omega_1 \rightarrow 0} F(n\omega_1)/f_1$ ，此时 $F(\omega)$ 显然为有限值，其表示单位频带宽度的频谱值，即频谱密度函数，简称频谱函数。

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (21.3)$$

4. 此时观察 $f(t)$ ，有：

$$f(t) = \sum_{n=-\infty}^{\infty} F(n\omega_1)e^{in\omega_1 t} = \sum_{n=-\infty}^{\infty} F(n\omega_1)e^{in\omega_1 t} \cdot \frac{\Delta(n\omega_1)}{\omega_1} = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} T_1 F(n\omega_1)e^{in\omega_1 t} \cdot \Delta(n\omega_1)$$

显然 $T_1 \rightarrow \infty, \omega_1 \rightarrow 0$ 时级数转化成积分式：

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{i\omega t} d\omega \quad (21.4)$$

5. 式21.3,21.4分别称为傅里叶正变换(FT)和反变换(IFT)公式， $F(\omega)$ 为 $f(t)$ 像函数， $f(t)$ 为 $F(\omega)$ 的像原函数，傅里叶变换可表示为 $F(\omega) = \mathcal{F}[f(t)]$ ， $f(t) = \mathcal{F}^{-1}[F(\omega)]$ 。

6. 除了式21.3,21.4的形式外，傅里叶变换还有其它形式，例如可以用自然频率 $f = 1/T = \omega/(2\pi)$ 代替 ω

$$\begin{cases} F(f) = \int_{-\infty}^{\infty} f(t)e^{-i2\pi ft} dt, & FT \\ f(t) = \int_{-\infty}^{\infty} F(f)e^{i2\pi ft} df, & IFT \end{cases}$$

7. 非周期信号可进行傅里叶变换的充分条件：

(a) $f(t)$ 绝对可积，即 $\int_{-\infty}^{\infty} f(t) dt < M$ ， $M > 0$

(b) 当 $\int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt = +\infty$ ，但能唯一地表示这个无穷大，一般引入狄拉克函数 δ 进行表示。

$$\int_{-\infty}^{\infty} \delta(\omega) d\omega = 1, \quad \delta(\omega) = 0, \omega \neq 0$$

8. 常见非周期信号的傅里叶变换：

(a) 冲激信号： $\mathcal{F}[\delta(t)] = 1$

(b) 单边指数信号： $\mathcal{F}[e^{-at}\varepsilon(t)] = \frac{1}{\alpha + i\omega}$ ， $\alpha > 0$

(c) 双边指数信号： $\mathcal{F}[e^{-\alpha|t|}] = \frac{2\alpha}{\alpha^2 + \omega^2}$ ， $\alpha > 0$

(d) 门函数： $\mathcal{F}[G_\tau(t)] = \tau \cdot \text{Sa}(\frac{\omega\tau}{2})$ ， $G_\tau(t) = \varepsilon(t + \tau/2) - \varepsilon(t - \tau/2)$ ， $\text{Sa}(x) = \frac{\sin x}{x}$

(e) 阶跃信号： $\mathcal{F}[\varepsilon(t)] = \pi \cdot \delta(\omega) + \frac{1}{i\omega}$

(f) 直流信号： $\mathcal{F}[1] = 2\pi \cdot \delta(\omega)$

Rayleigh 定理

以下讨论信号 $f(t)$ 的能量与其对应傅里叶变换后 $F(\omega)$ 的能量的关系：易知，信号 $f(t)$ 的总能量

$$U = \int_{-\infty}^{\infty} \|f(t)\|^2 dt = \int_{-\infty}^{\infty} \overline{f(t)} \cdot f(t) dt = \int_{-\infty}^{\infty} \overline{f(t)} \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \int_{-\infty}^{\infty} \overline{f(t)} e^{i\omega t} dt d\omega$$

注意到 $\int_{-\infty}^{\infty} \overline{f(t)} e^{i\omega t} dt$ 类似傅里叶变换公式，有 $\int_{-\infty}^{\infty} \overline{f(t)} e^{i\omega t} dt = \overline{\left[\int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \right]} = \overline{F(\omega)}$ ，即

$$U = \int_{-\infty}^{\infty} \|f(t)\|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \overline{F(\omega)} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} \|F(\omega)\|^2 d\omega = \frac{1}{\pi} \int_0^{\infty} \|F(\omega)\|^2 d\omega$$

以上即 Rayleigh 定理：信号在时域和频域的能量相等。

$$\int_{-\infty}^{\infty} \|f(t)\|^2 dt = \frac{1}{\pi} \int_0^{\infty} \|F(\omega)\|^2 d\omega$$

21.4.2 周期信号的傅里叶变换

- 对比非周期信号的傅里叶变换有，非周期信号的幅度谱为无穷小，傅里叶变换后对应的频谱密度函数为有限值，而周期信号的幅度谱为离散的有限值，进行傅里叶变换后得到的频谱密度函数必然为一系列位于角频率谐波处的冲激函数，其冲击强度与幅度呈正比，用狄拉克函数表示。
- 由定义易证，傅里叶变换具有如下基本性质：
 - 对称性： $\mathcal{F}[f(t)] = F(\omega) \Rightarrow \mathcal{F}[F(t)] = 2\pi f(-\omega)$
 - 时移性： $\mathcal{F}[f(t)] = F(\omega) \Rightarrow \mathcal{F}[f(t - t_0)] = F(\omega)e^{-i\omega t_0}$
 - 频移性： $\mathcal{F}[f(t)] = F(\omega) \Rightarrow \mathcal{F}[f(t)e^{i\omega_0 t}] = F(\omega - \omega_0)$ (可由定义证，也可由时移性和两次对称性证)
- 以下计算周期信号的傅里叶变换。首先计算狄拉克函数的傅里叶变换，因为只有在 $t = 0$ 时 $\delta \neq 0$ ，则有

$$\mathcal{F}[\delta(t)] = \int_{-\infty}^{\infty} \delta(t) e^{-i\omega t} dt = \int_{-\infty}^{\infty} \delta(t) dt = 1$$

由傅里叶变换的对称性，有

$$\mathcal{F}[1] = 2\pi\delta(-\omega) = 2\pi\delta(\omega)$$

对周期信号 $f(t)$ ，进行傅里叶级数展开有 $f(t) = \sum_{n=-\infty}^{\infty} F(n\omega_1) e^{in\omega_1 t}$ ，对 $f(t)$ 进行傅里叶变换有

$$\mathcal{F}[f(t)] = \mathcal{F} \left[\sum_{n=-\infty}^{\infty} F(n\omega_1) e^{in\omega_1 t} \right] = \sum_{n=-\infty}^{\infty} F(n\omega_1) \mathcal{F}[e^{in\omega_1 t}]$$

由频移性 $\mathcal{F}[e^{in\omega_1 t}] = \mathcal{F}[1 \times e^{in\omega_1 t}] = 2\pi\delta(\omega - n\omega_1)$ ，即

$$\mathcal{F}[f(t)] = \mathcal{F} \left[\sum_{n=-\infty}^{\infty} F(n\omega_1) e^{in\omega_1 t} \right] = 2\pi \sum_{n=-\infty}^{\infty} F(n\omega_1) \delta(\omega - n\omega_1)$$

对于周期函数 $f(t)$ ，仅分析其一个周期，即构造函数 $f_1(t)$ ，此时 $f_1(t)$ 为非周期函数：

$$f_1(t) = \begin{cases} f(t), & t \in [-T_1/2, T_1/2] \\ 0, & t \notin [-T_1/2, T_1/2] \end{cases}$$

显然 $f(t)$ 傅里叶级数系数 $F(n\omega_1) = 1/T_1 \cdot \int_{-T_1/2}^{T_1/2} f(t) e^{-in\omega_1 t} dt$ ，而 $f_1(t)$ 傅里叶变换 $F_1(\omega) = \mathcal{F}[f_1(t)] = \int_{-\infty}^{\infty} f_1(t) e^{-i\omega t} dt$ ，显然

$$F(n\omega_1) = \frac{1}{T_1} F_1(n\omega_1)$$

将上述结论带入，有

$$\mathcal{F}[f(t)] = 2\pi \sum_{n=-\infty}^{\infty} F(n\omega_1) \delta(\omega - n\omega_1) = \frac{2\pi}{T_1} \sum_{n=-\infty}^{\infty} F_1(n\omega_1) \delta(\omega - n\omega_1) = \omega_1 \sum_{n=-\infty}^{\infty} F_1(n\omega_1) \delta(\omega - n\omega_1) \quad (21.5)$$

显然，对于非周期信号，其周期化后信号的傅里叶变换，为原频谱与周期性冲激序列相乘，其包络保持原频谱的形状。

4. 常见周期信号的傅里叶变换：

$$(a) \text{ 正弦信号: } \mathcal{F}[\sin \omega_1 t] = -i\pi[\delta(\omega - \omega_1) - \delta(\omega + \omega_1)]$$

$$\text{余弦信号: } \mathcal{F}[\cos \omega_1 t] = \pi[\delta(\omega - \omega_1) + \delta(\omega + \omega_1)]$$

$$\text{复正弦信号: } \mathcal{F}[e^{i\omega_1 t}] = 2\pi\delta(\omega - \omega_1)$$

$$(b) \text{ 周期冲击信号: } \mathcal{F}\left[\sum_{n=-\infty}^{\infty} \delta(t - nT_1)\right] = \frac{2\pi}{T_1} \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_1) = \omega_1 \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_1)$$

21.4.3 傅里叶变换的性质

除了前述对称性、时移性、频移性等特性，傅里叶变换还具有其它特性：

$$\text{线性特性} \quad \mathcal{F}[f_1(t)] = F_1(\omega), \mathcal{F}[f_2(t)] = F_2(\omega) \implies \mathcal{F}[af_1(t) + bf_2(t)] = aF_1(\omega) + bF_2(\omega)$$

$$\text{尺度变换} \quad \mathcal{F}[f(t)] = F(\omega) \implies \mathcal{F}[f(at)] = \frac{1}{|a|} F\left(\frac{\omega}{a}\right)$$

$$\text{奇偶虚实性} \quad \mathcal{F}[f(t)] = R(\omega) - iX(\omega) = F(\omega)$$

如果 $f(t)$ 是实偶函数，则 $F(\omega)$ 也为实偶函数；如果 $f(t)$ 是实奇函数，则 $F(\omega)$ 为虚奇函数。

$$\text{微分特性} \quad \mathcal{F}[f(t)] = F(\omega) \implies \mathcal{F}\left[\frac{d^n}{dt^n} f(t)\right] = (i\omega)^n F(\omega)$$

$$\text{积分特性} \quad \mathcal{F}[f(t)] = F(\omega) \implies \mathcal{F}\left[\int_{-\infty}^t f(\tau) d\tau\right] = \pi F(0)\delta(\omega) + \frac{1}{i\omega} F(\omega) = \left[\pi\delta(\omega) + \frac{1}{i\omega}\right] F(\omega)$$

$$\text{频域微积分} \quad \mathcal{F}[f(t)] = F(\omega) \implies \mathcal{F}[t^n f(t)] = i^n \frac{d^n}{d\omega^n} F(\omega), \quad \mathcal{F}\left\{\left[\pi\delta(t) + i\frac{1}{t}\right] f(t)\right\} = \int_{-\infty}^{\omega} F(\Omega) d\Omega$$

$$\text{卷积特性} \quad \mathcal{F}[f_1(t)] = F_1(\omega), \mathcal{F}[f_2(t)] = F_2(\omega) \implies \mathcal{F}[f_1(t) \otimes f_2(t)] = F_1(\omega)F_2(\omega), \mathcal{F}[f_1(t)f_2(t)] = \frac{1}{2\pi} F_1(\omega) \otimes F_2(\omega)$$



信号的功率谱与能量谱

功率谱和能量谱分别从功率和能量的角度出发研究信号。参考信号的幅度谱和相位谱的定义，若以频率 ω 为横坐标，以单位频带宽度上的子信号的功率或能量为纵坐标，则可得到信号的功率谱或能量谱。信号的功率谱和能量谱均只有大小，没有相位。

1. 功率谱主要用于分析周期信号。对于单边功率谱，非直流子信号功率 $P = \frac{1}{2}c_n^2$ ，直流子信号 $P = c_0^2$ ；对于双边功率谱，子信号功率 $P = \|F(n\omega_1)\|^2$ ；

2. 能量谱主要用于分析非周期信号。对于单边能量谱， $U = \frac{1}{\pi} \|F(\omega)\|^2$ ；对于双边能量谱， $U = \frac{1}{2\pi} \|F(\omega)\|^2$ 。

需要说明的是，对于确定性信号，由幅度谱和相位谱已经可以实现对信号的分析，故功率谱和能量谱在此意义不大。功率谱和能量谱主要用于对随机信号的分析。

21.5 连续时间系统的频域分析方法

- 系统的频域分析法是将信号通过傅里叶变换分解成一系列正弦（复正弦）信号的和，得到信号的频谱；然后计算系统对每个正弦（复正弦）分量的响应，得到响应的频谱；最后通过反变换得到响应信号的过程。与时域分析法相比，频域法不需要求解微分方程及卷积计算，但需要进行两次傅里叶变换。但很多情况下，会直接给定激励信号的频谱，且只需得到响应信号的频谱，此时频域法就具有巨大优势。⁴
- 因为由傅里叶变换即可以将一般信号分解为正弦或复正弦信号，因此系统的频率响应是指系统对正弦信号 $\sin \omega t$ 或复正弦信号 $e^{i\omega t}$ 的响应。以下说明求解系统对复正弦信号响应的方法。对于微分方程描述的一般系统，假设微分方程

$$\frac{d^n}{dt^n} r(t) + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} r(t) + \cdots + a_1 \frac{d}{dt} r(t) + a_0 r(t) = b_m \frac{d^m}{dt^m} e(t) + b_{m-1} \frac{d^{m-1}}{dt^{m-1}} e(t) + \cdots + b_1 \frac{d}{dt} e(t) + b_0 e(t)$$

⁴需要说明的是，频域法只能求解系统的稳态响应或零状态响应，因为周期信号是一个无始无终的信号，可以认为在很远的过去即加在系统上，此时系统的响应已经进入一个稳定状态，即稳态响应。

激励信号为复正弦信号 $E(i\omega)e^{i\omega t}$, 假设系统对复正弦信号的响应仍是同频率的复正弦信号 $R(i\omega)e^{i\omega t}$, 代入微分方程有

$$\begin{aligned} & [(i\omega)^n + a_{n-1}(i\omega)^{n-1} + \cdots + a_1(i\omega) + a_0] R(i\omega)e^{i\omega t} = [b_m(i\omega)^m + b_{m-1}(i\omega)^{m-1} + \cdots + b_1(i\omega) + b_0] E(i\omega)e^{i\omega t} \\ \implies & R(i\omega) = \frac{b_m(i\omega)^m + b_{m-1}(i\omega)^{m-1} + \cdots + b_1(i\omega) + b_0}{(i\omega)^n + a_{n-1}(i\omega)^{n-1} + \cdots + a_1(i\omega) + a_0} E(i\omega) \end{aligned}$$

显然猜想正确, 记系统传输函数 $H(i\omega) = \frac{b_m(i\omega)^m + b_{m-1}(i\omega)^{m-1} + \cdots + b_1(i\omega) + b_0}{(i\omega)^n + a_{n-1}(i\omega)^{n-1} + \cdots + a_1(i\omega) + a_0}$, 则

$$R(i\omega) = H(i\omega)E(i\omega)$$

本式说明, 响应信号的复振幅等于激励信号的复振幅和系统传输函数复振幅之积, 而相位为两者相位之和。需要说明的是, $H(i\omega)$ 同时也是系统冲激响应 $h(t)$ 的傅里叶变换, 即 $H(i\omega) = \mathcal{F}[h(t)]$;

3. 进一步地, 讨论系统对余弦信号 $A \cos \omega t$ 的响应, 由欧拉公式, $A \cos \omega t = A \cdot \frac{e^{i\omega t} + e^{-i\omega t}}{2}$
此时系统对 $e^{i\omega t}$ 和 $e^{-i\omega t}$ 的传输函数分别为 $H(i\omega)$ 和 $H(-i\omega)$, 则系统对余弦信号的响应为

$$r(t) = \frac{A}{2} [H(i\omega)e^{i\omega t} + H(-i\omega)e^{-i\omega t}]$$

记 $H(i\omega) = \|H(i\omega)\|e^{i\varphi(\omega)}$, 当微分方程的系数均为实数时, 易证存在 $H(-i\omega) = \|H(i\omega)\|e^{-i\varphi(\omega)}$, 此时

$$r(t) = A\|H(i\omega)\| \cdot \frac{e^{i\varphi(\omega)}e^{i\omega t} + e^{-i\varphi(\omega)}e^{-i\omega t}}{2} = A\|H(i\omega)\| \cdot \cos[\omega t + \varphi(\omega)]$$

由此即可得到对周期性信号的系统响应求解方法, 而对非周期信号的系统响应求解方法与此类似。

4. 由此可见, 系统的传输函数 $H(i\omega)$ 对系统的响应具有重要影响, 具体体现在 $\|H(i\omega)\|$ 对幅度的影响和 $\varphi(\omega)$ 对相位的影响两个方面, 由此可具体分为系统的幅频特性和相频特性, 系统的幅频特性和相频特性统称为系统的频域特性:
- (a) 幅频特性曲线为 ω 和 $\|H(i\omega)\|$ 的关系, 描述系统对各个频率的(复)正弦信号的振幅的影响;
 - (b) 相频特性曲线为 ω 和 $\varphi(\omega)$ 的关系, 描述系统对各个频率的(复)正弦信号的相位的影响。
5. 对于任意输入信号, 可通过傅里叶级数展开或傅里叶变换得到其频谱, 而对于任意系统, 可根据其传输函数做出系统的频域特性图, 由上述四张图两两对应即可得到输出信号的频谱而无需进行卷积计算和微分方程求解, 这一方法即是系统对信号的频域分析法。



理想低通滤波器及其冲激响应、阶跃响应

1. 对于一类系统, 如果其系统特性“允许某些信号分量通过、同时阻止其它信号分量通过”, 则这类系统被称为滤波器(filter)。根据系统传输函数幅频特性的不同, 滤波器可分为以下几种基本类型:

- (a) 低通滤波器 (low pass filter, LPF): 系统仅允许低于截止频率的信号分量通过;
- (b) 高通滤波器 (high pass filter, HPF): 系统仅允许高于截止频率的信号分量通过;
- (c) 带通滤波器 (band pass filter, BPF): 系统仅允许两个特定频率之间的信号分量通过;
- (d) 带阻滤波器 (band stop filter, BSF): 系统仅允许两个特定频率之外的信号分量通过。

2. 显然上述的分类仅与系统的幅频特性有关, 而不涉及相频特性。理想滤波器 (ideal filter) 的定义则与系统的幅频和相频特性均有关, 是指能使通带内信号的幅值和相位都不失真, 阻带内的频率成分都衰减为零的滤波器, 其中:

- (a) 幅值失真: 系统对各个子信号幅度放大或衰减的程度不一样;
- (b) 相位失真: 系统对各个子信号延时不一样。

为此要求系统的幅频特性 $\|H(i\omega)\| = K$ 、相频特性 $\varphi(\omega) = -t_0 \cdot \omega$, 具体推导如下: 记输入信号与输出信号分别为 $e(t), r(t)$, 假设系统为不失真系统, 则只允许 $r(t)$ 较 $e(t)$ 出现延时和整体幅度的变化, 即 $r(t) = Ke(t - t_0)$, 对等式两边同做傅里叶变换

$$R(i\omega) = KE(i\omega)e^{-i\omega t_0} \implies H(i\omega) = \frac{R(i\omega)}{E(i\omega)} = Ke^{-i\omega t_0}$$

上式即是不失真系统应满足的条件。

3. 综上, 理想低通滤波器 (ideal low pass filter, ILPF) 即是理想滤波器与低通滤波器的结合, 系统 $H(i\omega)$

$$H_{ILPF}(i\omega) = \begin{cases} Ke^{-it_0\omega} & |\omega| < \omega_0 \\ 0 & |\omega| \geq \omega_0 \end{cases}$$

以下讨论理想低通滤波器的冲激响应 $h_{ILPF}(t)$ 和阶跃响应 $r_{ILPF}(t)$ 。根据系统的冲激响应 $h(t)$ 与传输函数 $H(i\omega)$ 的关系可以得到

$$h_{ILPF}(t) = \mathcal{F}^{-1}[H_{ILPF}(i\omega)] = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} Ke^{i(t-t_0)\omega} d\omega = \frac{K\omega_0}{\pi} \text{Sa}[(t-t_0)\omega_0]$$

再根据系统响应的频域解法, 理想低通滤波器阶跃响应 $r_{ILPF}(t)$

$$r_{ILPF}(t) = \mathcal{F}^{-1}\left\{ H_{ILPF}(i\omega) \mathcal{F}[\varepsilon(t)] \right\} = \frac{K}{2} + \frac{K}{\pi} \text{Si}[(t-t_0)\omega_0], \quad \text{Si}(x) = \int_0^x \text{Sa}(y) dy$$

可以看出, 系统相位特性的系数 t_0 即是系统响应的时间延迟。

21.5.1 因果性、Paley-Wiener 准则与物理可实现滤波器

- 由微分方程或传输函数即可在数学上定义一个系统, 然而并不是每一个由微分方程或传输函数表征的系统都能被实际设计出来。实际可实现的系统被称为因果系统或物理可实现系统, 这类系统必须在任何情况下均满足因果关系: 系统对激励的响应不应早于激励本身。
- 根据因果性的定义, 系统满足因果性的充分必要条件为: 对冲击信号 $\delta(t)$ 的响应 $h(t)$, 在冲击出现之前 ($t < 0$) 响应也为零 ($h(t) = 0$), 或表示为 $h(t)\varepsilon(t) = h(t)$ 。
- 除了从时域角度, 也可以从频域角度判断因果性, 具体地说, 是从系统的幅频特性 $\|H(i\omega)\|$ 进行判断。具体表述为: 在 $\int_{-\infty}^{\infty} \|H(i\omega)\|^2 d\omega$ 存在且有限的前提下, 系统物理可实现的充要条件是

$$\int_{-\infty}^{\infty} \frac{|\ln \|H(i\omega)\||}{1 + \omega^2} d\omega < \infty$$

以上即 Paley-Wiener 准则。根据 Paley-Wiener 准则, 物理可实现滤波器要求系统的幅频特性可以在某些频率点上为零, 但不能在一个区间内都是零, 显然上述四种理想滤波器都不是物理可实现滤波器。

- 为了得到物理可实现的滤波器, 必须对要求进行弱化:
 - 通带 (允许通过的信号频率范围) 内的各个频率上的增益允许有一定的差异;
 - 阻带 (不允许通过的信号频率范围) 内的幅频特性允许不为零, 只要求足够小;
 - 通带和阻带之间允许有缓冲。
- 以下介绍两种常见的物理可实现低通滤波器:
 - 最大平坦 LPF (Butterworth LPF): 式中 ω_0 为截止频率, n 为超参, n 越大越接近理想低通滤波器。

$$\|H(i\omega)\|^2 = \frac{1}{1 + (\omega/\omega_0)^{2n}}$$

- 通带起伏型 LPF (Chebyshev LPF, Cauer LPF): 允许在通带内的幅频特性有一定的起伏, 以此减小缓冲带宽度。

21.5.2 调制与解调

- 信号的调制是用待传输的信号 (调制信号), 控制另一个便于传输 (一般为高频) 的信号 (载波) 的某一个参数的变化, 以便达到传输信号的目的。通过调制可以使信号便于发射、具有更强的传输能力以及充分利用通信资源。而解调则是调制的逆过程。
- 在信号调制时, 一般基于正弦波或周期性脉冲信号作为载波进行调制, 载波不同控制的参数也不同。
 - 基于正弦波调制时, 相应的调制方法有幅度调制 (调幅, AM)、频率调制 (调频, FM) 和相位调制 (调相, PM), 即用调制信号控制载波的幅度、频率或相位;
 - 基于周期性脉冲信号调制时, 可相应控制脉冲的幅度 (脉冲幅度调制)、宽度 (脉冲宽度调制) 或间隔 (脉冲间隔调制) 等。

3. 首先介绍调幅信号。由定义得，调幅信号即是将低频的调制信号 $e(t)$ 加在高频的载波信号 $\cos \omega_c t$ 上。因为 $e(t)$ 的频率远低于载波，在 A_0 足够大时 $a(t)$ 的包络的形状将与 $e(t)$ 一致。由下式易知，幅度调制属于线性调制，满足齐次性和叠加性，而 FM 和 PM 均为非线性调制。

$$a(t) = [A_0 + ke(t)] \cos \omega_c t$$

另外可以证明，对于 AM 波，不使 AM 波携带的信号产生失真的第二个条件可以减弱为 $\varphi(\omega) = -t_0 \omega + b$ 。

4. 定义调制系数 m 为调幅波幅度变化的最大值与载波调制前的幅度的比值， m 又可分为上调制系数 m_a 和下调制系数 m_b 。若 $m = m_a = m_b$ 则称调制为对称调制，若 $m_b > 1$ 则称为过调制。

$$m_a = \frac{A_{max} - A_0}{A_0}, \quad m_b = \frac{A_0 - A_{min}}{A_0}, \quad m = \max\{m_a, m_b\} = \frac{\max\{|ke(t)|\}}{A_0}$$

5. 定义 AM 波的一些功率参数如下：

(a) 载波功率 $P_c = \frac{1}{2}A_0^2$;

(b) 瞬时功率 $P_T(t) = \frac{1}{2}[A_0 + ke(t)]^2$ (一个载波周期内的平均功率);

(c) 最大功率 $P_{max} = \frac{1}{2}(A_0 + \Delta A_{max})^2 = (1 + m_a)^2 P_c$;

(d) 平均功率 $\bar{P} = \left(1 + \sum_{i=0}^n \frac{m_i^2}{2}\right) P_c$ (一个调制波周期内的平均功率， m_i 为 $e(t)$ 傅里叶变换后第 i 次谐波的调制系数)。

6. 同步解调法是目前常用的解调方法，借助傅里叶变换实现解调。假设 $\mathcal{F}[e(t)]$ 已知，则 $\mathcal{F}[a(t)] = \mathcal{F}\left\{[A_0 + ke(t)] \frac{e^{i\omega_c t} + e^{-i\omega_c t}}{2}\right\}$ 可由傅里叶变换的频移特性得到，即将 $E(\omega)$ 向两侧各自平移 ω_c ，同时相位减少一半。

此时对调制信号再乘上同频率同相位的载波信号 $\cos \omega_c t$ ，则 $\mathcal{F}[a(t) \cos \omega_c t]$ 的变化规律同上。此时其频谱占用三段频带，且中间部分的频谱与原调制信号 $e(t)$ 形状相同，幅度为其一半。经过一个 $K = 2$ 的低通滤波器后，即可得到与原调制信号相同的频谱。

7. 除了基于正弦波调制外，还可以基于周期性脉冲信号调制，其中只有脉冲幅度调制的方法是线性的。记调制波为 $e(t)$ 、周期性脉冲信号为 $s_T(t)$ ，则脉冲幅度调制信号

$$a(t) = e(t)s_T(t)$$

只需脉冲信号的周期 T 足够大，使得频谱 $\mathcal{F}[a(t)]$ 不会互相重叠，则经过一个低通滤波器后即可解调。

●
●
●
信道的复用

1. 需要说明的是，脉冲幅度调制的主要意义不在于信号传输，而在于信道的复用。即对于两个调制信号 $e(t) = e_1(t) + e_2(t)$ ，只需要选择两个同周期但相位不同且不重叠的脉冲信号 $s_{1T}(t), s_{2T}(t)$ ，即可实现两脉冲幅度调制信号 $a(t) = a_1(t) + a_2(t)$ 在时间上不重叠，可以在一个信道中传输，这一思想即是时分复用 (Time Division Multiplexing, TDM)；
2. 除了时分复用外，频分复用 (Frequency Division Multiplexing, FDM) 也是一种基本的信道复用方法。对于两个调制信号 $e(t) = e_1(t) + e_2(t)$ ，只需要选择两个不同频率的余弦信号 $\cos \omega_1 t, \cos \omega_2 t$ ，同样可得到调幅信号 $a(t) = a_1(t) + a_2(t)$ ，有 $\mathcal{F}[a(t)] = \mathcal{F}[a_1(t)] + \mathcal{F}[a_2(t)]$ ，故只需要选取合适的 ω_1, ω_2 即可实现 $\mathcal{F}[a_1(t)]$ 和 $\mathcal{F}[a_2(t)]$ 在频域上不重叠，由此实现频分复用。

21.6 拉普拉斯变换

1. 与时域分析法相比，基于傅里叶变换的频域分析法将解微分方程的问题转化为了解代数方程的问题，并避免了卷积运算，极大简化了系统响应的求解过程。同时因为傅里叶变换具有清晰的物理意义也被广泛应用于各类信号分析任务中。但傅里叶变换方法仍具有以下缺点：
- 只能处理满足收敛条件的信号，大量信号的傅里叶变换不存在；

- 只能求系统的零状态响应而无法得到零输入响应。因为傅里叶变换的积分域是 $(-\infty, \infty)$, 使得变换时只涉及外加激励而没有初始状态, 所以求解出来的响应只跟外加激励有关。

拉普拉斯变换可视为傅里叶变换的推广。拉普拉斯变换在保持后者优点（无需卷积、变微积分为乘除）的同时对信号具有更强的适应性，而且可自动引入初始条件求解系统全响应；

- 对于信号 $f(t)$, 假设其傅里叶变换不存在, 则不妨引入衰减因子 $e^{-\sigma t}$ 从而构造新的信号 $f_1(t) = e^{-\sigma t}f(t)$ 。显然通过调整 σ 的取值可以加速 $f(t)$ 于 $t \rightarrow \infty$ (或 $t \rightarrow -\infty$) 方向上的收敛, 同时削弱另一个方向的收敛性。因此与原信号 $f(t)$ 相比, $f_1(t)$ 的自由度更高, 理论上其傅里叶变化存在的概率越大

$$F_1(i\omega) = \mathcal{F}[f_1(t)] = \int_{-\infty}^{\infty} f(t)e^{-\sigma t}e^{-i\omega t}dt \xlongequal{s=\sigma+i\omega} \int_{-\infty}^{\infty} f(t)e^{-st}dt$$

上式 $f_1(t)$ 的傅里叶变换结果可写为关于复数 s 的函数, 定义其为 $f(t)$ 的双边拉普拉斯变换

$$F(s) = \mathcal{L}_d[f(t)] = \int_{-\infty}^{\infty} f(t)e^{-st}dt \quad (\text{双边拉普拉斯变换})$$

式中 $\mathcal{L}_d[\cdot]$ 的下标 d 表示“双边”, 即积分域同傅里叶变换为 $(-\infty, \infty)$ 。将复数 s 的实部 σ 取 0 时双边拉普拉斯变换即退化为傅里叶变换。同理推导双边拉普拉斯逆变换。注意到 $f_1(t)$ 为 $F(s)$ 的傅里叶逆变换

$$\begin{aligned} f_1(t) &= f(t)e^{-\sigma t} = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(s)e^{i\omega t}d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\sigma + i\omega)e^{i\omega t}d\omega \\ \Rightarrow f(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\sigma + i\omega)e^{(\sigma+i\omega)t}d\omega = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} F(s)e^{st}ds = \mathcal{L}_d^{-1}[F(s)] \end{aligned} \quad (\text{双边拉普拉斯逆变换})$$

双边拉普拉斯逆变换是对复变函数 $F(s)$ 沿平行于虚轴的直线 $Re(s) = \sigma$ 作积分, 其几何意义是将 $f(t)$ 正交分解至直线 $Re(s) = \sigma$ 上的幅度变化的复正弦函数簇。双边拉普拉斯变换对周期信号、平稳随机过程、非因果系统分析具有很大的应用价值;

- 实际研究中的信号大多为有始信号, 即存在一个零时刻, 信号于零时刻后才施加于系统, 相应的拉普拉斯变换称为单边拉普拉斯变换。一般的拉普拉斯变换均指单边拉普拉斯变换

$$F(s) = \mathcal{L}[f(t)] = \int_0^{\infty} f(t)e^{-st}dt \quad (\text{单边拉普拉斯变换})$$

$$f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} F(s)e^{st}ds \cdot \varepsilon(t) \quad (\text{单边拉普拉斯逆变换})$$

进一步根据积分起点的不同又可将单边拉普拉斯变换分为 0^- 变换和 0^+ 变换, 相应的系统称为 0^- 系统和 0^+ 系统。前者自动考虑了系统于零时刻的初始条件, 故一般均使用前者;

- 由拉普拉斯变换的推导过程可知, 并非对所有的 σ 取值信号 $f(t)$ 的拉普拉斯变换均存在, 也并非所有信号 $f(t)$ 均存在适当的 σ 使得拉普拉斯变换存在。为此进一步分析拉普拉斯变换的收敛域。 $f(t)$ 的拉普拉斯变换收敛域定义为使 $e^{-\sigma t}f(t)$ 满足绝对可积条件的 σ 的区间。具体地:

- 单边拉普拉斯变换只针对右边信号 (有始信号)。对于右边信号 $f(t)$, 若存在 σ_0 使得 $e^{-\sigma_0 t}f(t)$ 收敛, 则收敛域为 $\sigma \in (\sigma_0, +\infty)$ 。 σ_0 称为收敛坐标; 复平面垂线 $Re(s) = \sigma_0$ 称为收敛边界 (或收敛轴);
- 双边拉普拉斯变换可针对右边信号 (有始信号)、左边信号 (有终信号) 和双边信号。其中右边信号的双边拉普拉斯变换收敛域与单边拉普拉斯变换的情况相同; 左边信号的双边拉普拉斯变换收敛域与右边信号的情况相反, 为 $\sigma \in (-\infty, \sigma_0)$; 而对于双边信号则需要将其拆为左边信号与右边信号的和 $f(t) = f_L(t)\varepsilon(-t) + f_R(t)\varepsilon(t)$, 最终的收敛域为左边信号 $f_L(t)\varepsilon(-t)$ 和右边信号 $f_R(t)\varepsilon(t)$ 收敛域的交集。如交集不存在则相应的双边拉普拉斯变换不存在 (如 1 、 $\sin(t)$ 、 $\cos(t)$ 等等)。

若 $f(t)$ 的拉普拉斯变换收敛域不包含 $\sigma = 0$, 则其傅里叶变换不存在;

- 进一步介绍几种常见的基本信号的单边拉普拉斯变换:

- 单边复正弦信号 $\mathcal{L}[e^{\alpha t}\varepsilon(t)] = \frac{1}{s-\alpha}$, $Re(s) > Re(\alpha)$ 。其中 α 可为任意复数;
- 阶跃信号 $\mathcal{L}[\varepsilon(t)] = \frac{1}{s}$, $\sigma > 0$ 。相当于令单边复正弦信号中的 $\alpha = 0$;

- 单边正弦、余弦信号 $\mathcal{L}[\sin(\omega_c t)\epsilon(t)] = \frac{\omega_c}{s^2 + \omega_c^2}$, $\mathcal{L}[\cos(\omega_c t)\epsilon(t)] = \frac{s}{s^2 + \omega_c^2}$, $\sigma > 0$;
- 单边正幂信号 $\mathcal{L}[t^n\epsilon(t)] = \frac{n!}{s^{n+1}}$, $\sigma > 0$ 。由分部积分按定义推导;
- 单边复正弦-正幂复合信号 $\mathcal{L}[e^{\alpha t}t^n\epsilon(t)] = \frac{n!}{(s - \alpha)^{n+1}}$, $\sigma > \alpha$;
- 冲击信号 $\mathcal{L}[\delta(t)] = 1$, $\mathcal{L}[\delta^{(n)}(t)] = s^n$, $\sigma > -\infty$ 。 $\delta^{(n)}(t)$ 表示 $\delta(t)$ 的 n 阶导。

可以看到, 拉普拉斯变换的结果往往非常简洁, 很多信号的拉普拉斯变换可以写为有理函数的形式。需要说明的是, 只有在单边拉普拉斯变换的情况下 $f(t)$ 与 $F(s)$ 才具有一一对应的关系, 而在双边拉普拉斯变换时则可能出现一个 $F(s)$ 对应多组 $f(t)$ 的情况, 此时即需要通过 $F(s)$ 的收敛域确定具体对应的 $f(t)$;

6. 最后介绍拉普拉斯变换的性质。记 $\mathcal{L}[f(t)] = F(s)$, $Re(s) \in (\sigma_1, \sigma_2)$, 则

线性特性	$\mathcal{L}[af_1(t) + bf_2(t)] = aF_1(s) + bF_2(s)$, 收敛域至少为 $F_1(s), F_2(s)$ 的交集, 适用于双边和单边拉氏变换
尺度变换	$\mathcal{L}[f(at)] = \frac{1}{ a }F\left(\frac{s}{a}\right)$, $Re(s) \in (\min\{a\sigma_1, a\sigma_2\}, \max\{a\sigma_1, a\sigma_2\})$, 适用于双边和 $a > 0$ 时的单边拉氏变换
时延特性	$\mathcal{L}[f(t - t_0)] = F(s)e^{-st_0}$, $Re(s) \in (\sigma_1, \sigma_2)$, 适用于双边和 $t_0 \geq 0$ 时的单边拉氏变换
单边周期化	$\mathcal{L}[f_T(t)] = \frac{F(s)}{1 - e^{-sT}}$, $f_T(t) = \sum_{n=0}^{\infty} f(t - nT)$ 表示以 T 为周期对 $f(t)$ 单边周期化的结果, 是时延特性的推广
复移频特性	$\mathcal{L}[f(t)e^{s_0 t}] = F(s - s_0)$, $Re(s) \in (\sigma_1 + Re(s_0), \sigma_2 + Re(s_0))$, 适用于双边和单边拉氏变换
时域微分	$\mathcal{L}\left[\frac{d^n}{dt^n}f(t)\right] = s^n F(s) - s^{n-1} \sum_{k=0}^{n-1} s^{-k} \frac{d^k}{dt^k}f(t) \Big _{t \rightarrow 0^-}$, 收敛域可能增大, 适用于单边和右边信号的双边拉氏变换
时域积分	$\mathcal{L}\left[\int_{0^-}^t f(\tau)d\tau\right] = \frac{F(s)}{s}$, 收敛域可能减小, 适用于单边和右边信号的双边拉氏变换
复频域微分	$\mathcal{L}[tf(t)] = -\frac{d}{ds}F(s)$, 收敛域可能增大
复频域积分	$\mathcal{L}\left[\frac{f(t)}{t}\right] = \int_s^{+\infty} F(p)dp$, 收敛域可能减小
参量微积分	$\mathcal{L}\left[\frac{\partial}{\partial a}f(t, a)\right] = \frac{\partial}{\partial a}F(s, a)$, $\mathcal{L}\left[\int_{a_1}^{a_2} f(t, a)da\right] = \int_{a_1}^{a_2} F(s, a)$, 收敛域不变
初值定理	$f(0^+) = \lim_{s \rightarrow \infty} sF_p(s)$, $f'(0^+) = \lim_{s \rightarrow \infty} s^2 F_p(s)$, 要求 $f(t), f'(t)$ 存在, 其中 $F_p(s)$ 表示 $F(s)$ 的真分式部分, 若 $\lim_{s \rightarrow \infty} F(s)$ 存在则 $F_p(s) = F(s)$
终值定理	$f(+\infty) = \lim_{s \rightarrow 0} sF(s)$, 要求 $f(t), f'(t)$ 存在, 且 $F(s)$ 的极点位于复平面的左半区, 在 $s = 0$ 上至多存在单极点
卷积特性	$\mathcal{L}[f_1(t) \otimes f_2(t)] = F_1(s)F_2(s)$, $\mathcal{L}[f_1(t)f_2(t)] = \frac{1}{2\pi i}F_1(s) \otimes F_2(s)$
对偶特性	$\mathcal{L}[F(t)] = 2\pi i f(-s)$

基于上述性质可以不按定义快速计算大部分一般信号的单边和双边拉普拉斯变换。对于单边拉普拉斯变换, 根据线性特性只需将信号分解为若干基本信号, 再查表得到各子信号拉普拉斯变换之和, 收敛域为相应子信号拉普拉斯变换收敛域的并集。对于双边拉普拉斯变换:

- 若信号为右边信号, 则按单边拉普拉斯变换计算;
- 若信号为左边信号 $f_L(t)$, 注意到其双边拉普拉斯变换按定义有

$$\mathcal{L}_d[f_L(t)] = \int_{-\infty}^0 f_L(t)e^{-st}dt \stackrel{t=-t}{=} \int_0^{\infty} f_L(-t)e^{st}dt = F_{-L}(-s)$$

式中 $f_L(-t)$ 即为右边信号。上式表明对于任意左边信号, 可将其翻折为右边信号后计算其右边信号的拉普拉斯变换结果 $F_{-L}(p)$, 再令 $p = -s$ 并翻转收敛域得到原左边信号 $f_L(t)$ 的双边拉氏变换结果;

- 若信号为双边信号, 则将其拆为左边和右边信号分别计算, 最终的收敛域为左边和右边信号拉普拉斯变换收敛域的交集, 此时位于收敛域左侧的极点源于右边信号, 而收敛域右侧的极点源于左边信号。若交集为空则该信号的双边拉普拉斯变换不存在。

21.6.1 拉普拉斯逆变换的求解

- 拉普拉斯的逆变换被定义为二维复平面上的广义线积分。因为复变函数积分较为复杂，工程中一般不直接按定义计算拉普拉斯逆变换，更多采用若干数学处理绕过积分运算；
- 首先讨论单边拉普拉斯逆变换的求解。常用的求解方法包括部分分式展开法和留数法。基于拉普拉斯变换的线性特性，部分分式展开法的基本思想是将复杂的 $F(s)$ 展开为若干个基本信号的拉普拉斯变换的和，则对应的信号即为相应基本信号的和。又因为大多常见信号的拉普拉斯变换结果可以写为有理函数的形式，不妨记

$$F(s) = \frac{N(s)}{D(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}$$

式中 $N(s), D(s)$ 分别表示分子多项式和分母多项式。代数知识指出基于部分分式展开可将 $F(s)$ 表示为多个简单分式的和。记 $D(s) = 0$ 的根分别为 s_1, \dots, s_n ，则

- 当 $m < n$ ，且 $D(s) = 0$ 无重根时，则 $F(s)$ 可以表示为

$$F(s) = \sum_{j=1}^n \frac{K_j}{s - s_j} \implies \mathcal{L}^{-1}[F(s)] = \sum_{j=1}^n K_j \mathcal{L}^{-1}\left[\frac{1}{s - s_j}\right] = \sum_{j=1}^n K_j e^{s_j t} \varepsilon(t)$$

显然只需确定系数 K_j 的取值即可得到 $F(s)$ 的拉普拉斯逆变换的结果。最直观的计算方法是通过对比系数解方程组得到。但也存在更巧妙的计算 K_j 通项公式的方法

$$\begin{aligned} K_j + \sum_{j' \neq j} \frac{K_{j'}(s - s_j)}{s - s_{j'}} &= (s - s_j)F(s) \\ \implies K_j &= (s - s_j)F(s) - \sum_{j' \neq j} \frac{K_{j'}(s - s_j)}{s - s_{j'}} = \lim_{s \rightarrow s_j} (s - s_j)F(s) = \lim_{s \rightarrow s_j} \frac{N(s)}{(D(s) - D(s_j))/(s - s_j)} = \frac{N(s_j)}{D'(s_j)} \end{aligned}$$

- 当 $m < n$ ，且 $D(s) = 0$ 有重根时，记 s_j 为 p 重根，则分解的因式中会出现

$$\frac{K_{j,p}}{(s - s_j)^p} + \frac{K_{j,p-1}}{(s - s_j)^{p-1}} + \cdots + \frac{K_{j,2}}{(s - s_j)^2} + \frac{K_{j,1}}{s - s_j}, \quad \mathcal{L}^{-1}\left[\frac{1}{(s - s_j)^n}\right] = \frac{t^{n-1}}{(n-1)!} e^{s_j t} \varepsilon(t)$$

- 当 $m \geq n$ 时，先通过长除将 $F(s)$ 变为一个关于 s 的多项式 $M(s)$ 和真有理分式 $\frac{N_1(s)}{D(s)}$ 的和。对于有理分式部分直接按上述方法处理，而对于 $M(s)$ 则注意到 $\mathcal{L}^{-1}[s^n] = \delta^{(n)}(t)$ 。

以上即为基于分式分解法求解拉普拉斯逆变换的思路。方法非常简单，其理论层面只涉及基础的代数知识，但只适用于 $F(s)$ 为有理函数的形式。在介绍分式分解法的求解过程时发现， $F(s)$ 的极点（即 $D(s) = 0$ 的根）具有清晰的物理意义，其决定了原函数中各个子信号的基本模式；

- 进一步介绍求解单边拉普拉斯逆变换的留数法。留数法具有完备的数学基础，其基本思想是将复平面上的线积分（即拉氏逆变换的原始定义）转化为复平面上的闭合曲线积分，再基于留数定理转化为对闭合曲线内各极点留数的计算。当 $F(s)$ 满足 $\lim_{s \rightarrow \infty} |F(s)| = 0$ 时按单边拉普拉斯逆变换的定义有

$$\mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi i} \int_{\sigma_0 - i\infty}^{\sigma_0 + i\infty} F(s) e^{st} ds \cdot \varepsilon(t) = \frac{1}{2\pi i} \lim_{r \rightarrow +\infty} \int_{\sigma_0 - ir}^{\sigma_0 + ir} F(s) e^{st} ds \cdot \varepsilon(t) = \sum_{s_j \in S_L} \text{Res}[F(s)e^{st}, s_j] \cdot \varepsilon(t)$$

上式只需将坐标轴原点平移至 $(\sigma_0, 0)$ 点，再顺时针旋转坐标轴使得原虚轴正方向为实轴正方向、原实轴负方向为虚轴正方向后基于复变函数中的留数定理和约当引理（见第 23.16.1 节）即可得证。式中 S_L 表示收敛域左侧的极点集合， $\text{Res}[F(s)e^{st}, s_j]$ 表示极点 s_j 关于函数 $F(s)e^{st}$ 的留数，有

$$\text{Res}[F(s)e^{st}, s_j] = \begin{cases} \lim_{s \rightarrow s_j} (s - s_j)F(s)e^{st} = \frac{N(s_j)e^{s_j t}}{D'(s_j)} & s_j \text{ 为单极点} \\ \lim_{s \rightarrow s_j} \frac{1}{(m-1)!} \frac{d^{m-1}}{ds^{m-1}} [(s - s_j)^m F(s)e^{st}] & s_j \text{ 为 } m \text{ 阶极点} \end{cases}$$

与部分分式展开法相比，留数法数学基础更为完备，且不要求 $F(s)$ 为有理函数，但留数法要求 $F(s)$ 满足 $\lim_{s \rightarrow \infty} |F(s)| = 0$ ，即不能解决有理函数中 $m \geq n$ 的情况；

4. 双边拉普拉斯逆变换的计算同样可采用部分分式展开法和留数法两种方法，但此时需考虑收敛域的影响：

- 对于双边拉普拉斯逆变换的部分分式展开法，展开 $F(s)$ 时需区分极点于收敛域左右侧的情况。仅考虑 $m < n$ 且 $D(s) = 0$ 无重根的情况，则

$$F(s) = \sum_{s_j \in S_L} \frac{K_j}{s - s_j} + \sum_{s_j \in S_R} \frac{K_j}{s - s_j} = F_R(s) + F_L(s), \quad \mathcal{L}_d^{-1} \left[\frac{1}{s - s_j} \right] = \begin{cases} e^{s_j t} \varepsilon(t) & s_j \in S_L \\ -e^{s_j t} \varepsilon(-t) & s_j \in S_R \end{cases}$$

S_L, S_R 分别表示收敛域左侧和右侧的极点集合； $F_R(s)$ 只包含位于收敛域左侧的极点，属于右边信号的拉普拉斯变换； $F_L(s)$ 只包含位于收敛域右侧的极点，属于左边信号的拉普拉斯变换。计算 $\mathcal{L}_d^{-1}[F(s)]$ 时对 $F_R(s), F_L(s)$ 分别计算。其中 $\mathcal{L}_d^{-1}[F_R(s)]$ 按单边拉普拉斯逆变换计算； $\mathcal{L}_d^{-1}[F_L(s)]$ 则首先翻折 $F_L(s)$ 后对 $F_L(-s)$ 作单边拉普拉斯逆变换得到右边信号 $f_L(\tau)$ ，再令 $\tau = -t$ 得到左边信号；

- 当采用留数法时注意到围线积分的正方向为逆时针，而在计算收敛域右侧极点的留数时围线积分是按顺时针方向，因此同样需要区分极点于收敛域左右侧的情况

$$\mathcal{L}_d^{-1}[F(s)] = \sum_{s_j \in S_L} \text{Res} [F(s)e^{st}, s_j] \cdot \varepsilon(t) - \sum_{s_j \in S_R} \text{Res} [F(s)e^{st}, s_j] \cdot \varepsilon(-t)$$

5. 综上所述，若 $F(s)$ 的收敛域右侧无极点，则其单边拉普拉斯逆变换和双边拉普拉斯逆变换的结果一致；若 $F(s)$ 的收敛域右侧存在极点，则其单边拉普拉斯逆变换的结果只包含了原信号 $t > 0$ 时的结果。

21.7 连续时间系统的拉普拉斯变换分析法

1. 与系统的频域分析法类似，线性系统的拉普拉斯变换分析法就是将前者采用的傅里叶变换替换为拉普拉斯变换。但因为单边拉普拉斯变换的微分特性，后者可直接得到系统考虑了零输入和零状态响应的全响应。以如下二阶系统为例，其中 $r(t), e(t)$ 分别表示响应信号和输入信号

$$\frac{d^2}{dt^2}r(t) + a_1 \frac{d}{dt}r(t) + a_0 r(t) = b_1 \frac{d}{dt}e(t) + b_0 e(t)$$

记两者的单边拉普拉斯变换分别为 $R(s), E(s)$ ，对等式两边同时作拉普拉斯变换，且注意到一般情况下输入信号必然满足 $\frac{d^{(n)}}{dt^{(n)}}e(0) = 0$ ，有

$$\begin{aligned} & (s^2 R(s) - s \cdot r(0^-) - r'(0^-)) + a_1 (sR(s) - r(0^-)) + a_0 R(s) = b_1 s E(s) + b_0 E(s) \\ \Rightarrow R(s) &= \frac{b_1 s E(s) + b_0 E(s) + (s + a_1) \cdot r(0^-) + r'(0^-)}{s^2 + a_1 s + a_0} = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} E(s) + \frac{(s + a_1) \cdot r(0^-) + r'(0^-)}{s^2 + a_1 s + a_0} \end{aligned}$$

再对 $R(s)$ 求拉普拉斯逆变换即可得到响应信号 $r(t)$ 。注意到上式 $R(s)$ 可分为两部分——只与输入 $e(t)$ 有关的部分（即零状态响应）和只与初始状态有关的部分（即零输入响应），因此拉普拉斯变换分析法可以得到系统的全响应。而傅里叶变换无法得到与初始状态有关的部分，故只能分析系统的零状态响应；

2. 如果只关注系统的零状态响应，则可得到 $R(s) = H(s)E(s)$ ，其中 $H(s)$ 称为系统的传递函数或系统函数。所谓“传递函数”是指 $H(s)$ 可实现从输入 $E(s)$ 到响应 $R(s)$ 的转换，而“系统函数”是指 $H(s)$ 只与系统的固有特征有关，而与输入或初始状态无关。另外 $H(s)$ 可视为系统对冲激信号响应 $h(t)$ 的拉氏变换，通过系统的 $H(s)$ 反求冲激响应 $h(t)$ 也是最常用的求解系统 $h(t)$ 的方法；

3. 进一步系统全响应的组成。上文介绍拉普拉斯逆变换的计算时已经指出信号的基本模式是由拉普拉斯变换的各极点决定，而系统全响应信号的极点显然有 $H(s)$ 和 $E(s)$ 两个来源：

- 零状态响应的极点既源于系统性质 $H(s)$ ，也源于输入信号 $E(s)$ ，故零状态响应信号中同时包含系统的自然响应和受迫响应；
- 零输入响应的极点只源于系统性质 $H(s)$ ，故零输入响应信号属于系统的自然响应。

而且极点的位置决定了响应的性质：

- 如果极点位于虚轴左侧，则对应的响应信号随时间而衰减，属于暂态响应；
- 如果极点位于虚轴上，则对应的响应信号幅度随时间不变，属于稳态响应；

- 如果极点位于虚轴右侧，则对应的响应信号随时间而放大，将导致系统崩溃，而如果该极点源于系统性质 $H(s)$ ，则系统不稳定。
4. 一般情况下施加的信号均为右边信号，因此拉普拉斯变换分析法一般也是考虑单边拉普拉斯变换。但也可按 $R(s) = H(s)E(s)$ 分析线性系统对双边信号的响应（仅零状态响应），此时对应双边拉普拉斯变换。需要说明的是，只要 $H(s), E(s)$ 存在，则拉普拉斯变换分析法对右边信号的响应一定适用，但因收敛域的影响对双边信号的响应则不一定：

- 首先分析 $H(s)$ 的收敛域。因为 $H(s)$ 对应系统的冲激响应 $h(t)$ ，而对于因果系统而言 $h(t)$ 一定是一个右边信号（不可能在冲激开始前就已经自发相应），故 $H(s)$ 的收敛域一定位于复平面的右侧区间；
- 再分析 $E(s)$ 的收敛域。当输入信号 $e(t)$ 为右边信号时， $E(s)$ 的收敛域也应为复平面的右侧区间，故 $H(s), E(s)$ 的收敛域一定存在交集。而当 $e(t)$ 为双边信号时， $E(s)$ 的收敛域存在左右边界，因此可能存在 $H(s), E(s)$ 的收敛域交集为空的情况，此时系统的响应 $r(t)$ 存在，但无法使用拉普拉斯变换分析法求解。

21.8 离散时间系统概述

- 离散时间信号是只在某些离散的时间点上有定义（有定义而非有非零值，例如周期冲击信号，只在某些点上有非零值，其它都为零，属于连续时间信号）的信号，这些离散的时间点可以是等间隔或不等间隔的，其后讨论的离散时间信号基本都是等间隔的。处理离散时间信号的系统被称为离散时间系统。
- 首先介绍几个特殊的离散时间信号：单位取样函数 $\delta(k)$ 、单位阶跃函数 $\varepsilon(k)$ 、单边指数序列 $a^k \varepsilon(k)$ （ a 可以为任意复数）、单边正弦序列 $A \cos(\omega_0 k + \varphi) \varepsilon(k)$ 等等。

$$\delta(k) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad \varepsilon(k) = \begin{cases} 1, & k \geq 0 \\ 0, & k < 0 \end{cases}$$

- 在实际应用中，绝大多数信号都属于模拟信号，发展离散时间系统的目的是因为其较连续时间系统具有更大的优越性，为此就需要首先将连续时间信号 $f(t)$ 转化为离散时间信号，这一过程称为抽样。抽样的原理与前述“脉冲幅度调制”相同。记周期冲击信号 $s(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_1)$ ，则 $f_s(t) = f(t)s(t)$ 为理想抽样信号；
- 进一步地将说明抽样的原理。假设 $F(\omega) = \mathcal{F}[f(t)]$ ，则

$$\mathcal{F}[f_s(t)] = \frac{1}{2\pi} \mathcal{F}[f(t)] \otimes \mathcal{F}\left[\sum_{k=-\infty}^{\infty} \delta(t - kT_1)\right] = \frac{1}{2\pi} F(\omega) \otimes \left[\omega_1 \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_1)\right] = \frac{\omega_1}{2\pi} \sum_{n=-\infty}^{\infty} [F(\omega) \otimes \delta(\omega - n\omega_1)]$$

因为 $f(t) \otimes \delta(t - t_0) = f(t - t_0)$ ，有

$$\mathcal{F}[f_s(t)] = \frac{\omega_1}{2\pi} \sum_{n=-\infty}^{\infty} F(\omega - n\omega_1) = \frac{1}{T_1} \sum_{n=-\infty}^{\infty} F(\omega - n\omega_1)$$

即理想抽样信号的频谱为模拟信号的频谱以 ω_1 为间隔多次平移，削减 T_1 后叠加， ω_1 被称为抽样频率，对应的 T_1 被称为抽样周期。此时，只需 ω_1 足够大，使频谱平移后不互相重叠，则经过一个增益 $K = T_1$ 的低通滤波器后即可还原出原模拟信号。这就是耐斯特 (Nyquist) 抽样定理或香农 (Shannon) 抽样定理：模拟信号可以有条件的由无数个离散点上的数值恢复出。

- 连续时间系统可以用微分方程进行描述，对应地离散时间系统则用差分方程进行描述，其一般形式如下

$$r(k+n) + a_{n-1}r(k+n-1) + \cdots + a_1r(k+1) + a_0r(k) = b_m e(k+m) + b_{m-1}e(k+m-1) + \cdots + b_1e(k+1) + b_0e(k)$$

从形式上，将微分方程的微分计算换成移序计算即为差分方程。与解微分方程一样，解差分方程同样需要初始条件，初始条件的个数等于差分方程的阶数，差分方程的阶数定义为方程中最大移序与最小移序的差：

6. 为方便表示, 引入移序算子 S : $S \cdot y(k) = y(k+1)$, 则差分方程一般式可表示为

$$\begin{aligned} S^n \cdot r(k) + a_{n-1}S^{n-1} \cdot r(k) + \cdots + a_1S \cdot r(k) + a_0r(k) &= b_mS^m \cdot e(k) + b_{m-1}S^{m-1} \cdot e(k) + \cdots + b_1S \cdot e(k) + b_0e(k) \\ \Rightarrow r(k) &= \frac{b_mS^m + b_{m-1}S^{m-1} + \cdots + b_1S + b_0}{S^n + a_{n-1}S^{n-1} + \cdots + a_1S + a_0} e(k) \\ \Rightarrow r(k) &= H(S) \cdot e(k) \end{aligned}$$

式中 $H(S) = \frac{N(S)}{D(S)}$, 其中 $D(S) = 0$ 为差分方程的特征方程;

7. 差分方程的解同样可以分为零状态响应 $r_1(k)$ 和零输入响应 $r_2(k)$ 两部分分别求解。其中 $r_2(k)$ 为求解一个齐次差分方程。以下说明线性常系数齐次差分方程的解法。首先讨论一阶差分方程及其解

$$Sr(k) + vr(k) = 0 \Rightarrow r(k+1) + vr(k) = 0 \Rightarrow r(k) = (-v)^k r(0)$$

注意到 $-v$ 同时也是特征方程 $S + v = 0$ 的解, 由此推广至高阶线性常系数齐次差分方程。对任意线性常系数差分方程 $D(S) \cdot r(k) = 0$, 记 $D(S) = 0$ 的特征根为 $\{v_1, v_2, \dots, v_n\}$, v_i 可取任意复数, 则 $r(k)$

$$r(k) = \begin{cases} [C_1v_1^k + C_2v_2^k + \cdots + C_nv_n^k] \varepsilon(k) & \text{特征根无重根时} \\ [(C_1 + C_2k + \cdots + C_m k^{m-1})v_1^k + C_{m+1}v_{m+1}^k + \cdots + C_nv_n^k] \varepsilon(k) & v_1 \text{ 为 } m \text{ 阶重根时} \end{cases}$$

8. 很多时候要求系统是稳定的, 即要求 $\lim_{k \rightarrow \infty} r(k) < \infty$, 根据上式线性常系数齐次差分方程的解的公式, 系统的稳定性与特征根有关

- (a) 当 $\|v\| < 1$ 时, 有 $\lim_{k \rightarrow \infty} v^k = 0$, $\lim_{k \rightarrow \infty} k^m v^k = 0$, 则无论有无重根, 系统均稳定;
- (b) 当 $\|v\| > 1$ 时, 有 $\lim_{k \rightarrow \infty} v^k = \infty$, $\lim_{k \rightarrow \infty} k^m v^k = \infty$, 则无论有无重根, 系统均不稳定;
- (c) 当 $\|v\| = 1$ 时, 有 $\lim_{k \rightarrow \infty} v^k = 1$, $\lim_{k \rightarrow \infty} k^m v^k = \infty$, 则当无重根, 系统临界稳定 (取决于具体的激励信号); 当有重根, 系统不稳定。

由此, 当 v_i 均落在复平面单位圆内时, 系统稳定; 当存在 v_i 均落在复平面单位圆外时, 系统不稳定; 单位圆上最多只能有单根。

21.9 离散时间傅里叶变换(DTFT)与Z变换

21.9.1 Z变换

1. 离散时间系统和离散时间序列也可以通过正交分解的方法在频域进行分析, 这就是离散时间傅里叶变换 (Discrete-time Fourier Transform, DTFT)。DTFT 可以视为 Z 变换的一个特例。以下基于连续信号的傅里叶变换推导离散信号的 Z 变换;

2. 对于连续信号 $f(t)$, 可以得到对应的离散时间信号 $f(k)$, 定义周期冲击信号 $\delta_T(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT)$, 得到理想抽样信号 $f_{\delta}(t) = f(t)\delta_T(t) = f(k)\delta_T(t)$ 。显然, 理想抽样信号为连续信号, 可进行傅里叶变换

$$F(i\omega) = \mathcal{F}[f(k)\delta_T(t)] = \int_{-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} f(k)\delta(t - kT) \right] e^{-i\omega t} dt = \sum_{k=-\infty}^{\infty} f(k) \left[\int_{-\infty}^{\infty} \delta(t - kT) e^{-i\omega t} dt \right] = \sum_{k=-\infty}^{\infty} f(k) e^{-i\omega kT}$$

由 Dirichlet 条件, 只有信号绝对可积, 即 $\sum_{k=-\infty}^{\infty} |f(k)| < \infty$ 时, 上述傅里叶变换存在, 当不满足绝对可积条件时, 对 $f_{\delta}(t)$ 首先乘上衰减因子 e^{-rkT} , 再计算 $\mathcal{F}[f_{\delta}(t)e^{-rkT}]$

$$F(r + i\omega) = \int_{-\infty}^{\infty} f_{\delta}(t) e^{-rkT} \cdot e^{-i\omega t} dt = \sum_{k=-\infty}^{\infty} f(k) e^{-(r+i\omega)kT} \xrightarrow[r=1]{z=e^{r+i\omega}} F(z) = \sum_{k=-\infty}^{\infty} f(k) z^{-k} \quad (21.6)$$

式 21.6 即是序列 $f(k)$ 的 Z 变换公式, 记做 $F(z) = \mathcal{Z}[f(k)]$, $F(z)$ 被称为序列 $f(k)$ 的生成函数。上述定义的 Z 变换中的求和区间为 $(-\infty, +\infty)$, 称为双边 Z 变换; 在很多情况下, 激励信号为有始信号, 而系统为因果系统, 则响应信号也是一个有始信号, 此时只需要考虑 $[0, +\infty)$ (右边序列), 相应的 Z 变换称为单边 Z 变换: $F(z) = \sum_{k=0}^{\infty} f(k) z^{-k}$;

3. 进一步地, 假设右边序列 $\mathcal{Z}[f(k)\varepsilon(k)] = F(z)$ 已知, 则左边序列 $f(-k)\varepsilon(-k-1)$ 的 Z 变换

$$\mathcal{Z}[f(-k)\varepsilon(-k-1)] = \sum_{k=-\infty}^{-1} f(-k)z^{-k} = \sum_{k=1}^{\infty} f(k)z^k = F(z^{-1}) - f(0)$$

同理, 也可以求出双边序列 $f(k)$ 的 Z 变换。

4. 由以上推导可以看出, Z 变换可以理解为对 $f(k)\delta_T(t)e^{-rkT}$ 的傅里叶变换, 而傅里叶变换成立要求满足绝对可积条件, 所以为使 Z 变换存在, r 应处于一定区间使得 $f(k)\delta_T(t)e^{-rkT}$ 绝对可积。根据 $z = e^{r+i\omega}$, 存在一个针对 z 的收敛域

(a) $f(k)$ 为有限长序列, 即 $F(z) = \sum_{k=k_1}^{k_2} f(k)z^{-k}$:

- i. 当 $k_1 < 0, k_2 < 0$, 收敛域为 $0 \leq |z| < +\infty$;
- ii. 当 $k_1 > 0, k_2 > 0$, 收敛域为 $0 < |z| \leq +\infty$;
- iii. 当 $k_1 < 0, k_2 > 0$, 收敛域为 $0 < |z| < +\infty$ 。

(b) $f(k)$ 为右边序列, 即 $F(z) = \sum_{k=0}^{\infty} f(k)z^{-k}$, 由根值法, 收敛域为圆心在原点半径为 R 的圆以外的区域

$$\lim_{k \rightarrow \infty} \sqrt[k]{|f(k)z^{-k}|} < 1 \implies |z| > \lim_{k \rightarrow \infty} \sqrt[k]{|f(k)|} = R$$

(c) $f(k)$ 为左边序列, 即 $F(z) = \sum_{k=-\infty}^{-1} f(k)z^{-k}$, 由根值法, 收敛域为圆心在原点半径为 R 的圆以内的区域

$$\lim_{k \rightarrow \infty} \sqrt[k]{|f(-k)z^k|} < 1 \implies |z| < \lim_{k \rightarrow \infty} \frac{1}{\sqrt[k]{|f(-k)|}} = R$$

(d) $f(k)$ 为双边序列, 则收敛域为左边序列和右边序列收敛域的交集。

考虑右边序列 $\mathcal{Z}[a^k\varepsilon(k)] = \frac{z}{z-a}$ 和左边序列 $\mathcal{Z}[-a^k\varepsilon(-k-1)] = \frac{z}{z-a}$, 可以发现两者 Z 变换的形式完全相同, 但收敛域不同。所以对于 Z 变换, 收敛域的意义不仅在于判断 Z 变换是否存在, 还能够区分不同的离散时间信号。

5. 常见信号的 Z 变换:

(a) 单位脉冲序列: $\mathcal{Z}[\delta(k)] = 1$ (收敛域: 全平面);

(b) 单边指数序列: $\mathcal{Z}[a^k\varepsilon(k)] = \frac{z}{z-a}$ (收敛域: $|z| > |a|$);

当 $a = 1$ 时为单位阶跃序列。另外 a 可以为任意复数, 故还可以推出单边正弦、余弦序列的变换。

6. 在此之前, 尚未介绍 Z 变换的逆变换。观察 Z 变换 $F(z) = \mathcal{Z}[f(k)] = \sum_{k=-\infty}^{\infty} f(k)z^{-k}$, 为洛伦级数形式。所以, Z 变换的逆变换的一种思路即是将 $F(z)$ 展开为洛伦级数 $\sum_{k=-\infty}^{\infty} f(k)z^{-k}$, 此时每一项的系数即是 $f(k)$

(a) 当 $F(z)$ 为多项式形式, 则 $f(k)$ 即是 $F(z)$ 每一项的系数的集合;

(b) 当 $F(z)$ 为有理分式形式, 可将 $F(z)$ 分解为多个简单分式之和, 并将每个简单分式展开为等比级数。

对一般形式的 $F(z)$, 可借助复变函数闭合积分与留数定理实现逆变换。考虑如下闭合积分, 其中积分路径 C 为以原点为圆心, 半径为 r 的正向圆曲线

$$\oint_C z^{-k} dz \xrightarrow{z=re^{i\theta}} ir^{-k+1} \int_0^{2\pi} e^{-i(k-1)\theta} d\theta = ir^{-k+1} \int_0^{2\pi} \cos[(k-1)\theta] - i \sin[(k-1)\theta] d\theta = \begin{cases} 2\pi i & k = 1 \\ 0 & k \neq 1 \end{cases}$$

应用上式, 对洛伦级数做闭合积分

$$\oint_C \sum_{n=-\infty}^{\infty} f(n)z^{-n} dz = 2\pi i \cdot f(1) \implies \oint_C \sum_{n=-\infty}^{\infty} f(n)z^{-n} \cdot z^{k-1} dz = 2\pi i \cdot f(k) \implies f(k) = \frac{1}{2\pi i} \oint_C F(z)z^{k-1} dz$$

再由留数定理, 即可得到 Z 变换的逆变换 $f(k) = \mathcal{Z}^{-1}[F(z)]$ (式 21.7), 即 $f(k)$ 为 $F(z)z^{k-1}$ 在曲线 C 内所有奇点的留数的和, 曲线 C 位于 $F(z)$ 的收敛域内

$$f(k) = \frac{1}{2\pi i} \oint_C F(z)z^{k-1} dz = \sum_n \text{Res}_n [F(z)z^{k-1}]_C \quad (21.7)$$

21.9.2 Z 变换的性质

线性特性
移序特性	$\mathcal{Z}[f(k)] = F(z) \Rightarrow \mathcal{Z}[f(k+n)] = z^n F(z)$ (双边序列移序)
	$\mathcal{Z}[f(k)] = F(z) \Rightarrow \mathcal{Z}[f(k+n)] = z^n [F(z) - f(0) - f(1)z^{-1} - \dots - f(n-1)z^{-(n-1)}]$ (右边序列增序)
	$\mathcal{Z}[f(k)] = F(z) \Rightarrow \mathcal{Z}[f(k-n)] = z^{-n} F(z)$ (右边序列减序, 当序列在 $k \leq -1$ 有定义时需加上 $f(k)z^{-k}$ 项)
尺度变换	$\mathcal{Z}[f(k)] = F(z) \Rightarrow \mathcal{Z}[a^k f(k)] = F\left(\frac{z}{a}\right)$
微分特性	$\mathcal{Z}[f(k)] = F(z) \Rightarrow \mathcal{Z}[kf(k)] = -z \frac{d}{dz} F(z)$
卷积特性	$\mathcal{Z}[f_1(k)] = F_1(z), \mathcal{Z}[f_2(k)] = F_2(z) \Rightarrow \mathcal{Z}[f_1(k) \otimes f_2(k)] = F_1(z)F_2(z)$
初值、终值定理	$\mathcal{Z}[f(k)] = F(z) \Rightarrow f(0) = \lim_{k \rightarrow \infty} F(z), f(+\infty) = \lim_{k \rightarrow 1} (z-1)F(z)$ ($f(0), f(+\infty)$ 存在且有限)

21.9.3 离散时间傅里叶变换 (DTFT) 与离散时间序列傅里叶级数 (DFS)

1. 离散时间傅里叶变换是 Z 变换的特例: 考虑 Z 变换 $F(z) = \mathcal{Z}[f(k)]$, 式中 $z = e^{r+i\omega}$, 当 **定义域包含单位圆时**, 令 $r = 0$, Z 变换即退化为离散时间傅里叶变换, 此时 $C: z = e^{i\omega}$ 为单位圆。式 21.8 即是离散时间傅里叶变换的正变换及反变换公式, 可以看出频谱函数 $F(e^{i\omega})$ 以 2π 为周期

$$\begin{cases} F(z) = \sum_{k=-\infty}^{\infty} f(k)z^{-k} \\ f(k) = \frac{1}{2\pi i} \oint_C F(z)z^{k-1} dz \end{cases} \xrightarrow{z=e^{i\omega}} \begin{cases} F(e^{i\omega}) = \sum_{k=-\infty}^{\infty} f(k)e^{-ik\omega} \\ f(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(e^{i\omega})e^{ik\omega} d\omega \end{cases} \quad (21.8)$$

2. 对于离散时间序列同样能够进行傅里叶级数展开 (式 21.9)。离散时间序列傅里叶级数和连续信号的傅里叶级数一样, 是将周期为 N 的离散时间序列 $f(k)$ 正交分解至同样周期为 N 的正交函数集 $\{e^{im\frac{2\pi}{N}k} | m = 0, 1, \dots, N-1\}$, 其推导也可以根据正交分解推导 (P291)

$$F(m) = \frac{1}{N} \sum_{k=0}^{N-1} f(k) e^{-i\frac{2\pi}{N}mk} \iff f(k) = \sum_{m=0}^{N-1} F(m) e^{i\frac{2\pi}{N}mk} \quad (21.9)$$

观察上述公式可以看出, 对于离散时间序列, 只需要有限组正交函数基即可以充分表示, 而连续时间信号需要无穷组正交基, 这是两者一个巨大的不同。

21.9.4 离散时间傅里叶变换的性质

线性特性

时移特性 $\mathcal{F}[f(k)] = F(e^{i\omega}) \Rightarrow \mathcal{F}[f(k-k_0)] = e^{-ik_0\omega} F(e^{i\omega})$

频移特性 $\mathcal{F}[f(k)] = F(e^{i\omega}) \Rightarrow \mathcal{F}[f(k)e^{i\omega_0 k}] = F(e^{i(\omega-\omega_0)})$

反褶特性 $\mathcal{F}[f(k)] = F(e^{i\omega}) \Rightarrow \mathcal{F}[f(-k)] = F(e^{-i\omega})$

奇偶虚实性 $\mathcal{F}[f(k)] = R(e^{i\omega}) - iX(e^{i\omega}) = F(e^{i\omega})$

- 当 $f(k)$ 为实数数列, 则 $F(e^{i\omega})$ 的实部或幅度偶对称, 虚部或相角奇对称;
- 当 $f(k)$ 为实偶数列, 则 $F(e^{i\omega})$ 只有实部, 为实偶函数;
- 当 $f(k)$ 为实奇数列, 则 $F(e^{i\omega})$ 只有虚部, 为虚奇函数。

频域微分特性 $\mathcal{F}[f(k)] = F(e^{i\omega}) \Rightarrow \mathcal{F}[kf(k)] = i \frac{d}{d\omega} F(e^{i\omega})$

卷积特性 $\mathcal{F}[f_1(k)] = F_1(e^{i\omega}), \mathcal{F}[f_2(k)] = F_2(e^{i\omega}) \Rightarrow \mathcal{F}[f_1(k) \otimes f_2(k)] = F_1(e^{i\omega})F_2(e^{i\omega})$ (时域卷积)

$\mathcal{F}[f_1(k)] = F_1(e^{i\omega}), \mathcal{F}[f_2(k)] = F_2(e^{i\omega}) \Rightarrow \mathcal{F}[f_1(k) \cdot f_2(k)] = \frac{1}{2\pi} F_1(e^{i\omega}) \odot F_2(e^{i\omega})$ (频域卷积)

Rayleigh 定理 $\mathcal{F}[f(k)] = F(e^{i\omega}) \Rightarrow \sum_{k=-\infty}^{\infty} \|f(k)\|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} \|F(e^{i\omega})\|^2 d\omega$

21.10 离散时间系统的变换域分析法

21.10.1 Z 域分析法

1. 本章将主要介绍如何从变换域的角度分析系统及其响应，而首先需要给出系统的响应信号的变换域求解方法。以下先介绍 Z 变换下求解系统响应的方法，此时可以分别求解零状态响应 $r_1(t)$ 和零输入响应 $r_2(t)$ ，也可以直接求解全响应 $r(t)$ （注意连续时间系统频域分析法时只能求解零状态响应）。讨论单边序列差分方程

$$\sum_{i=0}^n a_i r(k+i) = \sum_{i=0}^m b_i e(k+i)$$

两边同求 Z 变换，根据 Z 变换的移续特性 $\mathcal{Z}[f(k+n)] = z^n [F(z) - \sum_{i=0}^{n-1} f(i)z^{-i}]$ ，有

$$\begin{aligned} \sum_{i=0}^n a_i z^i \left[R(z) - \sum_{j=0}^{n-1} r(j)z^{-j} \right] &= \sum_{i=0}^m b_i z^i \left[E(z) - \sum_{j=0}^{m-1} e(j)z^{-j} \right] \\ \Rightarrow R(z) &= \frac{\sum_{i=0}^m b_i z^i E(z) - \sum_{i=0}^m \sum_{j=0}^{m-1} b_i e(j)z^{i-j}}{\sum_{i=0}^n a_i z^i} + \frac{\sum_{i=0}^n \sum_{j=0}^{n-1} a_i r(j)z^{i-j}}{\sum_{i=0}^n a_i z^i} \end{aligned}$$

式中 $r(j), e(j)$ 由差分方程的初始条件给出。可以看到，在全响应公式中，只有第一项与初始条件无关，说明为零状态响应。单独考虑零状态响应时，有

$$R_1(z) = \frac{\sum_{i=0}^m b_i z^i}{\sum_{i=0}^n a_i z^i} E(z) = \frac{N(z)}{D(z)} E(z) = H(z) E(z)$$

其中 $H(z)$ 同样称为系统的转移函数。

2. 前文 (P305) 介绍了系统稳定性与特征根的关系：若系统特征根都在单位圆内，则系统稳定；若单位圆上存在单根，则系统临界稳定；若单位圆上存在重根或单位圆外存在特征根，则系统不稳定。然而，当特征方程 $D(z) = 0$ 阶数过高时，不存在求根公式，此时利用特征根判断系统稳定将变得困难。由此讨论系统稳定性与特征方程系数的关系：

● ● ●

系统稳定性与特征方程系数——罗斯判据

(a) 首先，引入双线性变换： $z = \frac{\lambda + 1}{\lambda - 1} \iff \lambda = \frac{z + 1}{z - 1}$ ，得到新的特征方程 $D'(\lambda) = D\left(\frac{\lambda + 1}{\lambda - 1}\right)$ ，观察这一映射，有

$$D'(\lambda) = (a_n \lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_1 \lambda + a_0) / (\lambda - 1)^n$$

- i. λ 与 z 一一对应，即这一映射为单射；
- ii. 若 $D(z)$ 为有理函数，则 $D'(\lambda)$ 同样为有理函数；
- iii. z 平面单位圆以外的点映射到 λ 平面的右半平面； z 平面单位圆以内的点映射到 λ 平面的左半平面； z 平面单位圆上的点映射到 λ 平面的虚轴。

此时即将讨论“特征根是否在单位圆内”的问题转化为讨论“特征根是否在平面左半部分”的问题

(b) 接下来介绍判断系统稳定性的一个必要条件：即要求 $D'(\lambda)$ 分子多项式所有系数 a_i 全部同号。（若不同号则系统必然不稳定，但同号不代表系统必然稳定）

(c) 进一步地，介绍判断系统稳定性的一个充要条件——罗斯 - 霍维斯法则：首先构造罗斯 - 霍维斯数列

A_n	B_n	C_n	D_n	\dots
A_{n-1}	B_{n-1}	C_{n-1}	D_{n-1}	\dots
A_{n-2}	B_{n-2}	C_{n-2}	\dots	
A_{n-3}	B_{n-3}	C_{n-3}		
\vdots	\vdots	\vdots		
A_2	B_2	0		
A_1	0	0		
A_0	0	0		

308

上述列表前两行的元素为 $D'(\lambda)$ 分子多项式系数 a_i 的组合

$$\{A_n, B_n, C_n, D_n, \dots\} = \{a_n, a_{n-2}, a_{n-4}, \dots\}, \quad \{A_{n-1}, B_{n-1}, C_{n-1}, D_{n-1}, \dots\} = \{a_{n-1}, a_{n-3}, a_{n-5}, \dots\}$$

从第三行起，每一个元素的计算公式如下

$$A_{i-1} = \frac{1}{A_i}(A_i B_{i+1} - A_{i+1} B_i), \quad B_{i-1} = \frac{1}{A_i}(A_i C_{i+1} - A_{i+1} C_i), \quad C_{i-1} = \frac{1}{A_i}(A_i D_{i+1} - A_{i+1} D_i)$$

由此构成的数列 $\{A_n, A_{n-1}, \dots, A_0\}$ 称为罗斯 - 霍维斯数列。则系统稳定的充要条件是 **罗斯 - 霍维斯数列中所有元素均同号；且罗斯 - 霍维斯数列中符号变化的次数即是位于右半平面的特征根的个数。**

数字滤波器的分类

1. 按系统传输函数 $H(z)$ 分类：(记 $H(z) = \frac{\sum_{i=0}^m b_i z^i}{\sum_{i=0}^{n-1} a_i z^i + z^n} = \frac{b_m z^m + b_{m-1} z^{m-1} + \dots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0}$)
 - (a) 递归滤波器，或自回归 (AR) 滤波器： $H_{AR}(z) = H(z) \Big|_{b_i=0, i \geq 1} = \frac{b_0}{z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0}$
 - (b) 非递归滤波器，或滑动平均 (MA) 滤波器： $H_{MA}(z) = H(z) \Big|_{a_i=0, i < n} = z^{-n} [b_m z^m + b_{m-1} z^{m-1} + \dots + b_1 z + b_0]$
 - (c) 自回归滑动平均 (ARMA) 滤波器： $H_{ARMA}(z) = H(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \dots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0}$
2. 按单位函数响应 $h(k)$ 分类：($h(k) = \mathcal{Z}^{-1}[F(z)]$)
 - (a) 有限单位响应滤波器 (FIR): $h(k)$ 为有限长序列，差分方程描述为 $y(n) = \sum_{i=0}^{N-1} a_i x(n-i)$;
 - (b) 无限单位响应滤波器 (IIR): $h(k)$ 为无限长序列。

21.10.2 频域分析法

1. 同连续时间系统的频率响应推导方法 (P296) 类似，也可以求得离散时间系统的频率响应，只不过此时的响应是指系统对离散正弦信号 $\sin \omega k$ 或离散复正弦信号 $e^{i\omega k}$ 的响应。例如，对 $e^{i\omega k}$ 的响应为

$$r(k) = H(e^{i\omega})e^{i\omega k}$$

2. 可以看出，对于 (复) 正弦信号，系统的响应同样是频率不变、相位和幅度变化的 (复) 正弦信号。对幅度和相位的影响分别由 $\|H(e^{i\omega})\|$ 和 $\arctan H(e^{i\omega})$ 决定，由此同样可以得到离散时间系统的相频特性曲线和幅频特性曲线。

21.11 数字信号处理概述

1. 首先需要区分模拟信号和数字信号两个概念：
 - (a) 模拟信号：一类特殊的连续信号，是时间和值域均连续的信号；
 - (b) 离散信号：时间上不连续，但值域连续的信号；
 - (c) 数字信号：时间和值域均不连续的信号，其中值域经过量化处理。为了节省存储空间，数字信号的值域不是连续的，例如精确到小数点后几位。那么对于任意一个定义在实数域上的量值，其被储存为数字信号时必然存在约减，这一过程称为**量化**。

与模拟信号处理相比，数字信号处理具有巨大的优势，例如精度高、灵活性强、可实现多维信号处理，而且可以实现模拟系统很难达到的其它指标和特性等等。因此尽管数字信号处理存在着其它诸如复杂性大、功耗大等缺点，但依然得到广泛的应用。

2. 定义一些常用的离散时间序列，包括单位脉冲序列 $\delta(n)$ 、单位阶跃序列 $u(n)$ 、矩形序列 $R_N(n)$

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad R_N(n) = \begin{cases} 1, & 0 < n \leq N-1 \\ 0, & n < 0, n \geq N \end{cases}$$

此外还有正弦序列 $x(n) = \sin \omega_0 n$ 、实指数序列 $x(n) = a^n u(n)$ 、复指数序列 $x(n) = (re^{i\omega_0})^n = r^n (\cos \omega_0 n + i \sin \omega_0 n)$ 等等；

3. 离散时间序列同样可以进行相加、相乘和移位等基本运算（但要求两个序列长度相同，即 n 相同）。特别地，定义序列的能量 $S = \sum_{-\infty}^{\infty} |x(n)|^2$ 。对于任意实序列 $x(n)$ ，可进行奇偶化，分成奇部 $x_o(n)$ 和偶部 $x_e(n)$

$$x_o(n) = \frac{1}{2} [x(n) - x(-n)], \quad x_e(n) = \frac{1}{2} [x(n) + x(-n)]$$

另外，根据连续信号 $f(t)$ 基于 $\delta(t)$ 的分解（式21.2），同样可以得到序列 $x(n)$ 的分解 $x(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n-m)$ 。

21.12 离散傅里叶变换 (DFT)

1. 在介绍离散傅里叶变换之前，将重新回顾离散傅里叶级数 (DFS)。记 $\tilde{x}(n)$ 为周期为 N 的序列，即满足 $\tilde{x}(n) = \tilde{x}(n + kN)$, $k \in \mathbb{N}$ ，则 $\tilde{x}(n)$ 的 DFS 和对应的 IDFS 如下

$$\begin{aligned} \text{DFS : } \tilde{X}(k) &= \sum_{n=0}^{N-1} \tilde{x}(n) e^{-i \frac{2\pi}{N} kn} \iff \text{IDFS : } \tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) e^{i \frac{2\pi}{N} kn} \\ \xrightarrow{W_N = e^{-i \frac{2\pi}{N}}} \quad \text{DFS : } \tilde{X}(k) &= \sum_{n=0}^{N-1} \tilde{x}(n) W_N^{kn} \iff \text{IDFS : } \tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) W_N^{-kn} \end{aligned}$$

可以证明， $\tilde{X}(k)$ 和 W_N^{kn} 均是以 N 为周期的序列。周期序列仅需要有限个正交函数基即可展开为离散傅里叶级数，一方面是因为正交函数基中只有有限个独立的正交函数基，另一方面也是因为周期序列只有有限个独立的函数值；



离散傅里叶级数展开的主要性质

线性特性：DFS $[a\tilde{x}(n) + b\tilde{y}(n)] = a\tilde{X}(k) + b\tilde{Y}(k)$

移序特性：DFS $[\tilde{x}(n+m)] = w_N^{-km}\tilde{X}(k) \iff \text{IDFS} [\tilde{X}(k+l)] = w_N^{ml}\tilde{x}(n)$

证明 DFS $[\tilde{x}(n+m)] = w_N^{-km}\tilde{X}(k)$:

$$\text{DFS} [\tilde{x}(n+m)] = \sum_{n=0}^{N-1} \tilde{x}(n+m) w_N^{kn} \xlongequal{j=n+m} \sum_{j=m}^{N+m-1} \tilde{x}(j) w_N^{kj} w_N^{-km} = w_N^{-km} \sum_{j=m}^{N+m-1} \tilde{x}(j) w_N^{kj}$$

因为 w_N^{kj} 和 $\tilde{x}(j)$ 均以 N 为周期，有 $\tilde{x}(j)w_N^{kj}$ 也是以 N 为周期，而区间 $[m, N+m-1]$ 恰好是一个周期，此时

$$\text{DFS} [\tilde{x}(n+m)] = w_N^{-km} \sum_{j=m}^{N+m-1} \tilde{x}(j) w_N^{kj} = w_N^{-km} \sum_{j=0}^{N-1} \tilde{x}(j) w_N^{kj} = w_N^{-km} \tilde{X}(k)$$

同理也可证 IDFS $[\tilde{X}(k+l)] = w_N^{ml}\tilde{x}(n)$

共轭对称性：DFS $[\overline{\tilde{x}(n)}] = \overline{\tilde{X}(-k)} = \overline{\tilde{X}(N-k)}$

证明：

$$\because f(x)e^{iy} = [R(x) + iX(x)][\cos y + i \sin y] = [R(x)\cos y - X(x)\sin y] + i[R(x)\sin y + X(x)\cos y]$$

$$\overline{f(x)e^{iy}} = [R(x) - iX(x)][\cos y - i \sin y] = [R(x)\cos y - X(x)\sin y] - i[R(x)\sin y + X(x)\cos y]$$

$$\therefore \overline{f(x)e^{iy}} = \overline{f(x)e^{-iy}}$$

$$\therefore \text{DFS} [\overline{\tilde{x}(n)}] = \sum_{n=0}^{N-1} \overline{\tilde{x}(n)} e^{-i \frac{2\pi}{N} kn} = \sum_{n=0}^{N-1} \overline{\tilde{x}(n)} e^{i \frac{2\pi}{N} kn} = \overline{\tilde{X}(-k)}$$

进一步地，可以得到 $\tilde{X}(k)$ 的共轭偶对称分量 $\tilde{X}_e(k)$ 和共轭奇对称分量 $\tilde{X}_o(k)$

$$\begin{cases} \tilde{X}_e(k) = \text{DFS} \left\{ \text{Re} [\overline{\tilde{x}(n)}] \right\} = \frac{1}{2} \text{DFS} [\tilde{x} + \overline{\tilde{x}(n)}] = \frac{1}{2} [\tilde{X}(k) + \overline{\tilde{X}(N-k)}] \\ \tilde{X}_o(k) = \text{DFS} \left\{ i\text{Im} [\overline{\tilde{x}(n)}] \right\} = \frac{1}{2} \text{DFS} [\tilde{x} - \overline{\tilde{x}(n)}] = \frac{1}{2} [\tilde{X}(k) - \overline{\tilde{X}(N-k)}] \end{cases} \xrightarrow{k=N-k} \begin{cases} \tilde{X}_e(k) = \overline{\tilde{X}_e(-k)} = \overline{\tilde{X}_e(N-k)} \\ \tilde{X}_o(k) = -\overline{\tilde{X}_o(-k)} = -\overline{\tilde{X}_o(N-k)} \end{cases}$$

周期卷积：($\tilde{x}(n)$, $\tilde{y}(n)$ 的周期均为 N)

$$\text{DFS} [\tilde{x}(n)\tilde{y}(n)] = \frac{1}{N} \sum_{l=0}^{N-1} \tilde{X}(l)\tilde{Y}(k-l) = \frac{1}{N} \sum_{l=0}^{N-1} \tilde{X}(k-l)\tilde{Y}(l) \iff \text{IDFS} [\tilde{X}(k)\tilde{Y}(k)] = \sum_{m=0}^{N-1} \tilde{x}(m)\tilde{y}(n-m) = \sum_{m=0}^{N-1} \tilde{x}(n-m)\tilde{y}(m)$$

证明 $\text{IDFS}[\tilde{X}(k)\tilde{Y}(k)] = \sum_{m=0}^{N-1} \tilde{x}(m)\tilde{y}(n-m) = \sum_{m=0}^{N-1} \tilde{x}(n-m)\tilde{y}(m)$:

$$\begin{aligned}\text{IDFS}[\tilde{X}(k)\tilde{Y}(k)] &= \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k)\tilde{Y}(k)w_N^{-kn} = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} \tilde{x}(m)w_N^{km}\tilde{Y}(k)w_N^{-kn} \\ &= \sum_{m=0}^{N-1} \tilde{x}(m) \left[\frac{1}{N} \sum_{k=0}^{N-1} \tilde{Y}(k)w_N^{-k(n-m)} \right] = \sum_{m=0}^{N-1} \tilde{x}(m)\tilde{y}(n-m)\end{aligned}$$

2. 离散傅里叶变换 (DFT) 的概念可由离散傅里叶级数得到。注意到离散傅里叶级数的公式：尽管 $\tilde{x}(n), \tilde{X}(k)$ 均为长度无限的周期数列，但 $\text{DFS}[\tilde{x}(n)], \text{IDFS}[\tilde{X}(k)]$ 的表达式仅与其一个周期内的取值有关。定义区间 $[0, N-1]$ 为 $\tilde{x}(n)$ 的主值区间，记有限长序列 $x(n) = \tilde{x}(n)R_N(n)$ 为 $\tilde{x}(n)$ 的主值序列⁵。令 $x(n)$ 替代 $\tilde{x}(n)$ ，即可得到离散傅里叶变换的公式（式 21.10）

$$\text{DFT} : X(k) = \sum_{n=0}^{N-1} x(n)w_N^{kn} \iff \text{IDFT} : x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)w_N^{-kn} \quad (21.10)$$

3. 式 21.10 也可以写成矩阵形式（式 21.11）。显然，对长度为 N 的序列 \mathbf{x} ，需要进行 N 次乘法才可以得到 \mathbf{X} 中的一个元素，需要进行 N^2 次乘法才可得到序列 \mathbf{X} ，离散傅里叶反变换亦然。综上所述，离散傅里叶变换的计算复杂度为 N^2 。离散傅里叶变换的性质与离散傅里叶级数展开的性质完全相同。

$$\begin{aligned}\text{DFT} : \mathbf{X} = \mathbf{W}_N \cdot \mathbf{x} \iff \text{IDFT} : \mathbf{x} = \mathbf{W}_N^{-1} \cdot \mathbf{X} \\ \mathbf{x} = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}, \quad \mathbf{W}_N = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & w_N^{1 \times 1} & \cdots & w_N^{(N-1) \times 1} \\ \vdots & \ddots & & \\ 1 & w_N^{1 \times (N-1)} & \cdots & w_N^{(N-1) \times (N-1)} \end{bmatrix} \quad (21.11)\end{aligned}$$

4. 进一步地，讨论负指数组列 $x(n) = e^{iq\omega_0 n}$ 的离散傅里叶变换，其中 $q \in \mathbb{N}$, $\omega_0 = 2\pi/N$

$$\text{DFT}[e^{iq\omega_0 n}] = \sum_{n=0}^{N-1} e^{iq\omega_0 n} \cdot e^{-ik\omega_0 n} = \sum_{n=0}^{N-1} e^{i(q-k)\omega_0 n} = \frac{1 - e^{i2\pi(q-k)}}{1 - e^{i2\pi(q-k)/N}} = \begin{cases} N, & k = q \\ 0, & k \neq q \end{cases}$$

可以看到，由任意频率负指数组合形成的时间序列，其离散傅里叶变换的结果将出现在各自信号分量的基频处。因此，离散傅里叶变换算法实质上对频率具有选择性，这被称为离散傅里叶变换的选频性。利用选频性可以实现时间序列的压缩。

5. 利用离散傅里叶变换对连续时间信号 $f(t)$ 进行分析的完整流程如下

$$f(t) \xrightarrow{\text{采样}} f(n) \xrightarrow[x(n)=f(n)R_N(n)]{\text{截短}} x(n) \xrightarrow{\text{DFT}} X(k)$$

可以看到，从原始信号 $f(t)$ 至 $x(n)$ 的过程中经历了两次近似，而这两个过程都有可能产生失真

- (a) 在采样过程中可能产生信号的混迭，这就要求采样的频率尽可能大以满足耐斯特抽样定理。关于采样的具体描述见 P 304；
- (b) 在截短过程中可能产生信号的泄露。因为 $x(n) = f(n)R_N(n)$ ，易知 $x(n)$ 的频谱为 $f(n)$ 与 $R_N(n)$ 频谱的卷积。讨论与矩形窗口序列 $R_N(n)$ 的频谱

$$X(e^{i\omega}) = \text{DTFT}[R_N(n)] = \sum_{n=-\infty}^{\infty} R_N(n)e^{-i\omega n} = \sum_{n=0}^{N-1} e^{-i\omega n} = \frac{1 - e^{-i\omega N}}{1 - e^{-i\omega}} = \frac{\sin(\omega \frac{N}{2})}{\sin(\frac{\omega}{2})} e^{-i\omega \frac{N-1}{2}} \implies \|X(e^{i\omega})\| = \frac{\sin(\omega \frac{N}{2})}{\sin(\frac{\omega}{2})}$$

当矩形窗口不断加宽，窗口序列逐渐趋于直流序列，有

$$\|X(e^{i\omega})\| = \frac{\sin(\omega \frac{N}{2})}{\sin(\frac{\omega}{2})} \implies \lim_{N \rightarrow \infty} \|X(e^{i\omega})\| = \delta(\omega)$$

⁵ $R_N(n)$ 为宽度为 N 的门序列， $x(n)$ 在主值区间外的取值均为 0

所以 $f(n)$ 与 $R_N(n)$ 频谱的卷积时, 对非周期信号当且仅当矩形窗口趋于无穷宽使其频谱函数为 $\delta(\omega)$, 才能使 $x(n)$ 的频谱与 $f(n)$ 的频谱完全相等, 否则将不可避免地产生畸变, 这就是信号的泄露; 需要说明的是, 若 $f(n)$ 具有周期性, 周期为 T , 则不要求矩形窗口宽度 N 趋于无穷, 仅要求 $N = kT$, $k \in \mathbb{N}^*$, 若不满足同样会产生泄露。

21.13 快速傅里叶变换 (FFT)

- 由式 21.11, 长度为 N 的序列的离散傅里叶变换 (DFT) 需要 N^2 次乘法运算与 $N(N - 1)$ 次加法运算, 需要消耗较大的计算资源。快速傅里叶变换 (FFT) 是 DFT 的多种快速算法的统称。1965 年提出的基 2FFT 算法是第一种 FFT 算法, 至今仍有广泛应用;
- FFT 的基本思想是将长度为 N 的序列分解为多个子序列, 对每个子序列进行 DFT 运算从而减少总计算量。这一过程需要利用系数 $W_N^{kn} = e^{-j\frac{2\pi}{N}kn}$ 的周期性和对称性:
 - 周期性: $W_N^{k(n+N)} = W_N^{(k+N)n} = W_N^{kn}$
 - 对称性: $W_N^{k+N/2} = -W_N^k$, $W_N^{N/2} = -W_N^0 = -1$
- 基 2FFT 算法得名于其假设序列长度 $N = 2^M$, 当实际序列长度不满足时可在其后补零, 将序列 $x(n)$ 按奇偶顺序拆分成两项, 则其 DFT 如下:

$$\begin{aligned} X(k) = \text{DFT}[x(n)] &= \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_N^{(2r+1)k} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_N^{2rk}, \quad k \in \{0, \dots, N-1\} \end{aligned}$$

其中序列 $x(2r)$, $x(2r+1)$ 的长度仅为 $\frac{N}{2}$, 又因为 $W_N^{2rk} = W_{N/2}^{rk}$, 则

$$\begin{aligned} \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk} &= \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_{N/2}^{rk} = \text{DFT}[x(2r)] = G(k) \\ \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_N^{2rk} &= \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_{N/2}^{rk} = \text{DFT}[x(2r+1)] = H(k), \quad k \in \left\{0, \dots, \frac{N}{2}-1\right\} \\ \therefore X(k) &= G(k) + W_N^k H(k) \\ X(k + \frac{N}{2}) &= G(k) - W_N^k H(k), \quad k \in \left\{0, \dots, \frac{N}{2}-1\right\} \end{aligned}$$

此时将长度为 N 的序列的 DFT 转化为两个长度为 $\frac{N}{2}$ 的序列的 DFT, 只需 $\frac{N^2}{4} + \frac{N}{2} \left(\frac{N}{2} + 1 \right)$ 次乘法运算与 $N \left(\frac{N}{2} - 1 \right) + N$ 次加法运算;

- 因为 $N = 2^M$, 则 $\frac{N}{2}$ 依然是 2 的倍数, 即可共进行 $M = \log_2 N$ 次细分至 $\frac{N}{2}$ 组子序列, 每组序列长度为 2, 需 1 次乘法运算与 2 次加法运算, 则基 2FFT 共需要 $\frac{N}{2} \log_2 N$ 次乘法运算与 $N \log_2 N$ 次加法运算。

$$X(0) = x(0) + W_2^0 x(1) = x(0) + W_N^0 x(1) \quad X(1) = x(0) + W_2^1 x(1) = x(0) - W_N^0 x(1)$$

- 当 $N = P \cdot Q \neq 2^M$, $P, Q \in \mathbb{N}^*$ 时, 也可以不补零而直接采取其它的 FFT 算法, 算法思路同样为拆分长序列的 DFT 运算为多组短序列的 DFT 运算⁶;
- 首先需要对序列进行变换, 以长度 $N = 12 = 3 \times 4$ 的序列 ($P = 3$, $Q = 4$) 为例介绍序列变换方法:

$$\begin{bmatrix} x(0) & x(1) & x(2) & x(3) \\ x(4) & x(5) & x(6) & x(7) \\ x(8) & x(9) & x(10) & x(11) \end{bmatrix} \iff \begin{bmatrix} x(0,0) & x(0,1) & x(0,2) & x(0,3) \\ x(1,0) & x(1,1) & x(1,2) & x(1,3) \\ x(2,0) & x(2,1) & x(2,2) & x(2,3) \end{bmatrix}$$

⁶ 此时计算复杂度将大于基 2FFT。

$$\begin{bmatrix} X(0) & X(3) & X(6) & X(9) \\ X(1) & X(4) & X(7) & X(10) \\ X(2) & X(5) & X(8) & X(11) \end{bmatrix} \iff \begin{bmatrix} X(0,0) & X(0,1) & X(0,2) & X(0,3) \\ X(1,0) & X(1,1) & X(1,2) & X(1,3) \\ X(2,0) & X(2,1) & X(2,2) & X(2,3) \end{bmatrix}$$

上式中, 将一维时域序列 $x(n)$ 与一维频域序列 $X(k)$ 依不同的法则变换为二维时域序列 $x(n_0, n_1)$ 和二维频域序列 $X(k_0, k_1)$

$$n = n_0 \cdot Q + n_1, \quad k = k_0 + k_1 \cdot P \quad n_0, k_0 \in \{0, \dots, P-1\}, \quad n_1, k_1 \in \{0, \dots, Q-1\}$$

7. 此时序列 $x(n)$ 的 DFT 运算如下

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} \iff X(k_0 + k_1 \cdot P) &= \sum_{n_0=0}^{P-1} \sum_{n_1=0}^{Q-1} x(n_0 \cdot Q + n_1) W_N^{(n_0 \cdot Q + n_1)(k_0 + k_1 \cdot P)} \\ &\iff X(k_0, k_1) = \sum_{n_0=0}^{P-1} \sum_{n_1=0}^{Q-1} x(n_0, n_1) W_N^{n_0 k_1 P Q} W_N^{n_0 k_0 Q} W_N^{n_1 k_1 P} W_N^{n_1 k_0} \end{aligned}$$

因为 $W_N^{n_0 k_1 P Q} = W_N^{n_0 k_1 N} = 1$, $W_N^{n_0 k_0 Q} = W_P^{n_0 k_0}$, $W_N^{n_1 k_1 P} = W_Q^{n_1 k_1}$, 则上式

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \iff X(k_0, k_1) = \sum_{n_1=0}^{Q-1} \left\{ \left[\sum_{n_0=0}^{P-1} x(n_0, n_1) W_P^{n_0 k_0} \right] W_N^{n_1 k_0} \right\} W_Q^{n_1 k_1}$$

8. 可以看到, 上述算法将长度为 N 的 DFT 分解为两次长度分别为 P, Q 的 DFT, 此外还需乘上系数 $W_N^{n_1 k_0}$, 总的乘法次数为 $P^2 Q + N + Q^2 P = N(P + Q + 1)$, 加法次数为 $P(P - 1)Q + Q(Q - 1)P = N(P + Q - 2)$;
9. 根据算数基本定理, 当 N 不为素数时, 即可拆成多个素数因子的乘积, 并采用以上算法计算其 DFT, 因子数越多, 计算量越小。
10. 以上均是介绍离散傅里叶变换的快速算法, 实际上利用这一快速算法同样可以进行离散傅里叶反变换, 即 IFFT。由式 21.10, 对 IDFT 计算式两边同时取共轭, 有

$$\begin{aligned} \overline{x(n)} &= \frac{1}{N} \sum_{k=0}^{N-1} \overline{X(k) W_N^{-kn}} = \frac{1}{N} \sum_{k=0}^{N-1} \overline{X(k)} W_N^{kn} = \frac{1}{N} \text{DFT}[\overline{X(k)}] \\ \therefore \quad x(n) &= \frac{1}{N} \overline{\text{DFT}[\overline{X(k)}]} \end{aligned}$$

可以看到, 对 $X(k)$ 的离散傅里叶反变换可以转换为对 $\overline{X(k)}$ 的离散傅里叶正变换的共轭。



FFT 的应用——利用 FFT 计算多项式乘法

1. FFT 提供了高效实现 DFT 的算法, 从而使得很多运算都可以首先转换为 DFT 运算, 再由 FFT 实现。本处以多项式乘法为例介绍 FFT 的应用;
2. 记 $p_n(x), q_n(x)$ 为最高项均为 n 的多项式^a, 则两者乘积 $r_{2n}(x) = p_n(x)q_n(x)$ 的最高项为 $2n$, 将两多项式最高项补至 x^{2n} 项, 记两者的系数序列分别为 a_k, b_k , 则 $r_{2n}(x)$ 的系数 c_k 可视为 a_k 与 b_k 的卷积

$$\begin{cases} p_n(x) = \sum_{k=0}^{2n} a_k x^k \\ q_n(x) = \sum_{k=0}^{2n} b_k x^k \end{cases} \implies r_{2n}(x) = p_n(x)q_n(x) = \sum_{k=0}^{2n} c_k x^k, \quad c_k = \sum_{j=0}^{2n} a_j b_{k-j} = \sum_{j=0}^{2n} a_{k-j} b_j$$

3. 由 DFT 的卷积运算的性质, 两序列时域卷积相当于频域乘积再做 IDFT, 由两次 DFT 运算和一次 IDFT 运算即可得到 c_n , 以上三次运算均可以由 FFT 实现

$$c_k = \sum_{j=0}^{2n} a_j b_{k-j} = \sum_{j=0}^{2n} a_{k-j} b_j = \text{IDFT}[A_k B_k]$$

4. 当 a_k, b_k 均为实数序列时, 上述算法还可进一步优化——由一次 DFT 运算同时计算 A_k, B_k 。构造复数序列 $g_k =$

$$a_k + b_k i$$

$$G_k = \text{DFT}[a_k + b_k i] = A_k + B_k i = [Re\{A_k\} - Im\{B_k\}] + [Im\{A_k\} + Re\{B_k\}]i = Re\{G_k\} + Im\{G_k\}i$$

进一步地由 DFT 的共轭对称性从上式分离出 A_k 与 B_k

$$\begin{cases} A_k = \text{DFT}[Re\{g_k\}] = \frac{1}{2}(G_k + \overline{G_{2n-k}}) \\ B_k = -i\text{DFT}[Im\{g_k\}] = -\frac{i}{2}(G_k - \overline{G_{2n-k}}) \end{cases}$$

^a项数不等时补零

21.14 傅里叶变换的扩展——信号的多分辨分析

21.14.1 短时傅里叶变换 (STFT)

- 标准傅里叶变换将时域信号分解为多组正交的复正弦函数，从而将复杂的时域信息变化至频率域中，傅里叶变换具有以下缺陷：
 - 只适用于分析平稳信号，对于非平稳信号无能为力⁷；
 - 为了得到时域信号的频域特征，必须使用信号在时域中的全部信息，甚至未来信息；
 - 如果信号在某一时刻的小邻域内发生变化，则频谱无法标定变化的时间位置和发生变化的强度；
 - 对于包含高频信息和低频信息的信号而言，其时域精度和频域精度不容易调整。
- 为了改善傅里叶变换对非平稳信号的处理效果，可以对序列进行局部平稳化，即将长的非平稳随机过程看成是一系列短时随机平稳信号的叠加，此时对短时信号进行傅里叶变换的结果即同时具有了时域和频域的特征，局部平稳化的过程通过加窗实现，以上即是短时傅里叶变换 (short-time Fourier transform, STFT) 的基本思路；
- 定义窗口函数 $w(t-b)$ ， b 表示位置分量，则对信号 $x(t)$ 的短时傅里叶变换及反其变换如式 21.12：

$$\text{STFT} : G(\omega, b) = \int_{-\infty}^{\infty} x(t)w(t-b)e^{-i\omega t}dt \iff \text{ISTFT} : x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(\omega, b)w(t-b)e^{i\omega t}d\omega db \quad (21.12)$$

- STFT 中一般以 Gaussian 函数 $g_a(t)$ 作为窗口函数，参数 a 越大，窗口越宽：

$$g_a(t) = \frac{1}{2\sqrt{\pi a}} e^{-\frac{t^2}{4a}}, \quad a > 0$$

记 $g_a(t)$ 的傅里叶变换为 $G_a(\omega)$ ，参考力学中重心和惯性半径的概念，可以得到窗口函数的时窗中心 t^* 、时窗半径 Δ_t 、频窗中心 ω^* 和频窗半径 Δ_ω

$$\begin{cases} t^* = \frac{\int_R t \cdot g_a^2(t)dt}{\int_R g_a^2(t)dt} \\ \Delta_t = \sqrt{\frac{\int_R (t-t^*)^2 \cdot g_a^2(t)dt}{\int_R g_a^2(t)dt}} \end{cases} \quad \begin{cases} \omega^* = \frac{\int_R \omega \cdot G_a^2(\omega)d\omega}{\int_R G_a^2(\omega)d\omega} \\ \Delta_\omega = \sqrt{\frac{\int_R (\omega-\omega^*)^2 \cdot G_a^2(\omega)d\omega}{\int_R G_a^2(\omega)d\omega}} \end{cases}$$

- 时窗半径 Δ_t ：窗口函数在信号的时域范围内能分析的区间， $\Delta_t = \sqrt{a}$ ；
- 频窗半径 Δ_ω ：窗口函数在信号的频域范围内能分析的区间， $\Delta_\omega = \frac{1}{2\sqrt{a}}$ 。

注意到对任意固定窗口函数，时窗半径与频窗半径的乘积 $\Delta_t \cdot \Delta_\omega = 0.5$ 为定值，说明 STFT 的时间和频率分辨率不可能同时达到最高，这是其最大的局限性：

- a 越小，窗口函数的时窗半径越小，频窗半径越大，时域分辨率增加而频域分辨率降低；
- a 越大，窗口函数的时窗半径越大，频窗半径越小，时域分辨率减小而频域分辨率增加。

⁷ 平稳信号：如果信号的频率成分不随时间的变换而发生改变，那么该信号就被称为平稳信号。

21.14.2 连续小波变换 (CWT)

- 如上文所述，当缩放系数 a 固定时，STFT 无法同时提高时间和频率分辨率。一个自然的解决方法即是设定多组缩放系数 a ，对目标信号进行多组变换，从而同时到达最佳时域和频域分辨率，连续小波变换 (Continuous Wavelet Transform, CWT) 即采取这一思路；
- 连续小波变换的基函数不再是正弦或复正弦函数，而是定义小波函数 $\psi_{a,b}(t)$ ，其中 $\psi(t)$ 称为基本小波或母小波， b, a 分别为平移量（时间参数）和伸缩量（频率参数）。与 STFT 的窗口函数类似，小波函数同样具有时窗中心 t^* 、时窗半径 Δ_t 、频窗中心 ω^* 和频窗半径 Δ_ω 等四个参数

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

小波函数具有以下特点：

- (a) 快速衰减： $\int_R \psi_{a,b}^2(t)dt = \int_R \psi^2(t)dt = C_t < \infty$
- (b) 波动性： $\int_R \psi_{a,b}(t)dt = 0$

- 将信号 $f(t)$ 分解至小波函数，得到小波变换及其逆变换的表达式（式 21.13）

$$\text{CWT : } W(a, b) = \int_{-\infty}^{\infty} f(t)\overline{\psi_{a,b}(t)}dt \iff \text{ICWT : } f(t) = \frac{1}{C_\omega} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{a^2} W(a, b) \psi_{a,b}(t) db da \quad (21.13)$$

式中 $W(a, b)$ 称为小波系数， $C_\omega = \int_R \frac{1}{\omega} \mathcal{F}^2[\psi_{a,b}(t)]d\omega$ 。小波系数越大，表示信号在该时间区间内与对应的小波函数越接近。小波系数是时间参数和频率参数的双窗函数，所以说小波变换是一种信号的时间-尺度分析，可以自动调整时频分辨率；

- 对同一信号，选取不同的小波基，其 CWT 的结果不同，因此选取合适的小波函数至关重要。常用的小波函数包括 Daubechies 小波、Coiflets 小波、Symlets 小波等等。CWT 的步骤如下：
 - 确定合适的小波函数 $\psi_{a,b}(t)$ 及其初始频率参数 $a = a_0$ ；
 - 沿时间轴扫描信号（即代入式 21.13），得到一组小波系数 $\{W(a_0, b_0), W(a_0, b_1), \dots\}$ ；
 - 改变频率参数一个单位，即 $a = a_1$ ，再次沿时间轴扫描信号；
 - 循环执行上一步骤，直至完成沿频域轴的扫描，最终得到一组二维的小波系数矩阵 $W(a, b)$ 。

21.14.3 离散小波变换 (DWT)

- CWT 最大的缺陷在于结果的冗余性：一维序列经过 CWT 后将变为二维小波系数矩阵，离散小波变换 (Discrete Wavelet Transform) 有助于解决上述问题。DWT 是对信号的时间-频率相平面进行采样，即针对尺度参数 a 与平移参数 b ，与 DFT 不同，DWT 的采样频率随尺度改变而改变：当尺度较大时，可以采取较小的采样频率以节省储存及运算资源；
- 在连续小波函数 $\psi_{a,b}(t)$ 中，对尺度参数 a 与平移参数 b 进行离散化，并没有对时间参数进行离散化：
 - 尺度参数离散化： $a = a_0^j, a_0 > 1, j \in \mathbb{Z}$ (a_0 越接近 1 离散程度越低，逆变换难度越低)；
 - 平移参数离散化： $b = k a_0^j b_0, b_0 > 0, k \in \mathbb{Z}$ (平移参数的离散化依赖于尺度参数的离散化)。
 由此得到离散的小波函数 $\psi_{j,k}(t)$ ，且一般令 $a_0 = 2, b_0 = 1$ ，即二进小波 (dyadic wavelet)，式中 j 决定缩放、 k 决定平移

$$\psi_{j,k}(t) = a_0^{-\frac{j}{2}} \psi(a_0^{-j}t - kb_0) \xrightarrow{a_0=2, b_0=1} 2^{-\frac{j}{2}} \psi(2^{-j}t - k), \quad k, j \in \mathbb{Z}$$

- 二进小波依然存在增维现象，若能将参数 j, k 整合为一个参数，则可避免增维现象，即紧支二进小波 (compact dyadic wavelet)。式 21.14 中 $n = 2^j + k$ ， j 是满足 $2^j \leq n$ 的最大整数。另外应限制 $\psi_n(t)$ 和 $f(t)$ 的非零区间为 $[0, 1]$ ；

$$f(t) = \sum_{n=0}^{\infty} d_n \psi_n(t) \quad \psi_n(t) = 2^{-\frac{j}{2}} \psi(2^{-j}t - k) \quad (21.14)$$

- 式 21.14 解决了小波变换的增维问题，然而式中依然会得到无穷个的小波系数 d_n ，占据大量存储空间，由此在小波函数 $\psi(t)$ 的基础上提出尺度函数 $\varphi(t)$ 的概念，其中尺度函数可视为低通滤波器，用于拟合信号

的低频部分，而小波函数可视为高通滤波器，拟合信号的高频部分。式 21.15 中 L 为信号分解的层数；

$$f(t) = a_{L-1}\varphi_L(t) + \sum_{n=0}^{L-1} d_n\psi_n(t) \quad \varphi_n(t) = 2^{-\frac{L}{2}}\varphi(2^{-j}t - k), L \leq N \quad (21.15)$$

5. 由紧支二进小波对信号进行分解时，每一次仅对信号的低频部分进行分解，进行 L 层分解将得到 $L+1$ 个分量，这一现象有可能导致部分频率接近的高频信号分量无法被分解。

21.14.4 小波包变换 (WPT)

1. 小波包变换 (wavelet packet transform, WPT) 与上文所述 DWT 基本相同，但可以从根本上避免相似高频信号分量无法被分解的问题。用小波包变换对信号进行多层分解时，将同时对每一层的高频分量和低频分量进行分解， L 层分解将得到 2^L 个信号分量。

21.15 数据平滑去噪

数据去噪算法多可以视为一种低通滤波器，即将数据中频率高于特定值的部分视为噪声并将其滤除。

21.15.1 Savitzky-Golay 滤波器 (S-G 滤波器)

1. S-G 滤波器最初由 Savitzky A 和 Golay M 于 1964 年提出⁸，是一种特殊的低通滤波器被广泛应用于数据流平滑降噪。不同于传统滤波器，S-G 滤波器直接对时域数据进行平滑，而无需进行频域与时域的转换。算法将噪声视为误差，在时域内基于多项式和最小二乘法通过移动窗口对数据进行拟合。算法的特点在于能保留相对极大值、极小值和宽度等分布特征；
2. 考虑一个由 $2m+1$ 个数据点组成的窗口，令窗口中心点的坐标为 0，则观测点集可以表示为 $\{y_{-m}, \dots, y_0, \dots, y_m\}$ 。算法对窗口内的数据点进行平滑降噪。构造 $k-1$ 次多项式拟合各点，则对于第 $x \in \{-m, \dots, 0, \dots, m\}$ 个点，有

$$y_x = \hat{y}_x + e_x = (a_0 + a_1x + \dots + a_{k-1}x^{k-1}) + e_x$$

上式中 \hat{y}_x 为 y_x 的平滑值， e_x 为误差。对于窗口内 $2m+1$ 个数据点，可联立 $2m+1$ 组方程，并写成矩阵的形式

$$Y = \hat{Y} + E \quad \hat{Y} = XA \quad X = \begin{bmatrix} 1 & \dots & (-m)^{k-1} \\ 1 & \dots & (-m+1)^{k-1} \\ \vdots & \ddots & \vdots \\ 1 & \dots & m^{k-1} \end{bmatrix} \quad A = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{bmatrix}$$

上式中 X 为 $(2m+1) \times k$ 阶系数矩阵， A 为 $k \times 1$ 阶未知向量。欲使方程组有解，应使得 $2m+1 \geq k$ ；

3. 基于最小二乘法拟合多项式求解未知向量 A ，有

$$\begin{aligned} \hat{Y} &= \arg \min_A \|E\|_2^2 = \arg \min_A \|Y - XA\|_2^2 = \arg \min_A (Y - XA)^T(Y - XA) \\ &= \arg \min_A (Y^T - A^T X^T)(Y - XA) \\ &= \arg \min_A Y^T Y - 2A^T X^T Y + A^T X^T X A \end{aligned}$$

注意到对于矩阵乘法求导，有

$$\frac{\partial(u^T v)}{\partial x} = \frac{\partial u^T}{\partial x} v + \frac{\partial v^T}{\partial x} u$$

⁸Savitzky, A., & Golay, M. J. E. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. Analytical Chemistry, 36(8), 1627–1639.
doi:10.1021/ac60214a047

则上式对 A 求导并令导函数为 0 得到极小值

$$\frac{\partial}{\partial A} (Y^T Y - 2A^T X^T Y + A^T X^T X A) = 2X^T X A - 2X^T Y = 0 \implies A = (X^T X)^{-1} X^T Y, \quad \hat{Y} = X (X^T X)^{-1} X^T Y$$

以上即为对数据集 Y 的平滑降噪公式。注意到当 $2m + 1 = k$ 时，多项式恰好可以完全拟合 $2m + 1$ 个点，此时恒有 $(X^T X)^{-1} X^T = I$, I 为单位阵，即不对数据集作任何平滑；当 $2m + 1 > k$ 时则进行平滑。

第 22 章

控制论

22.1 自动控制原理基本概念

1. 控制是指为了克服干扰的影响达到期望的目标而对被控对象中的某一个（某一些）物理量（被控变量）进行的操作；
2. 自动控制系统是指由被控对象和控制器按一定方式连接起来，以完成某种自动控制任务的有机整体；
3. 控制系统可分为开环控制系统与闭环控制系统两类。前者是指被控变量（输出量）对系统的控制作用没有产生任何影响的系统，即输出量不作为系统的输入值。开环控制系统存在的最大问题是抗干扰能力差。闭环控制系统则是指控制器与被控对象之间不仅存在正向作用而且存在反馈作用的控制系统，即输出量对控制量有直接影响。闭环控制系统可以减小或消除偏差。闭环控制系统又可分为反馈控制系统与偏差控制系统；
4. 在闭环控制系统中存在以下几类信号：
 - 输入信号 $u(t)$: 又称为输入量、给定值；
 - 扰动：分为内部扰动与外部扰动；
 - 输出信号 $y(t)$: 被控变量，可简称输出量；
 - 主反馈信号 $b(t)$: 由系统输出端取出并反向送回系统输入端的信号，分为正反馈与负反馈；
 - 偏差信号 $e(t)$: 输入信号与主反馈信号的偏差，简称偏差，是输入端对误差的衡量；
 - 误差信号：系统被控变量期望值与实际值的差，简称误差。
5. 按不同分类标准，自动控制系统可分为以下几类：
 - 按系统的输入信号特征：
 - 恒值控制系统：输入信号恒定不变的系统。此时控制的目标是希望系统的被控变量尽可能保持在期望值附近，因此控制器的任务是增强系统的抗干扰能力，使干扰作用于系统时，被控变量尽快恢复至期望值；
 - 随动控制系统：输入信号随时间任意变化的系统。此时控制的目标是希望系统的输出信号紧跟输入信号的变化，因此控制器的任务是克服系统的惯性，提高系统的跟踪能力。
 - 按系统中传递的信号变化特征：
 - 连续控制系统：系统中各环节信号都是时间的连续函数；
 - 离散控制系统：系统中某处或几处信号是时间的离散函数。
 - 按系统的固有特性：
 - 线性控制系统：同时满足叠加性和均匀性（齐次性）的控制系统。线性系统总可以用线性微分方程表示：

$$\frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \cdots + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_m \frac{d^m u(t)}{dt^m} + b_{m-1} \frac{d^{m-1} u(t)}{dt^{m-1}} + \cdots + b_1 \frac{du(t)}{dt} + b_0 u(t)$$

当线性微分方程的系数为常数时则称系统为定常系统（时不变系统），反之则称为时变系统；

- 非线性控制系统：不同时满足叠加性和均匀性的控制系统，典型的非线性特性有饱和特性、死区特性、间隙特性、继电特性和磁滞特性等等。
6. 对控制系统的要求包括：稳定、尽量快的瞬态响应、尽量小的稳态误差，因此系统的稳定性、瞬态性能和稳态性能也就是系统分析的三个维度。

22.2 经典控制理论的系统输入一输出模型

22.2.1 传递函数 (transfer function)

1. 传递函数 $H(s)$ 是由系统的拉普拉斯变换分析法引入的、在经典控制领域中最重要的描述线性系统输入一输出关系的数学模型之一（第 21.7 节）。一般地，**定义系统的传递函数 $H(s)$ 为零初始条件下系统的输出信号的拉普拉斯变换 $Y(s)$ 与输入信号的拉普拉斯变换 $U(s)$ 之比**

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \quad (\text{传递函数})$$

如果传递函数的分母中 s 的最高次数为 n ，则称该系统为 n 阶系统。为保证系统的因果性，传递函数应是有理函数，即 $m \leq n$ 。根据多项式理论，一个 n 次多项式可唯一地分解为 n 个一次因式的乘积，则又有

$$H(s) = K \cdot \frac{(s + z_1)(s + z_2) \cdots (s + z_m)}{(s + p_1)(s + p_2) \cdots (s + p_n)}$$

式中 K 为传递函数的传递系数，也是根轨迹增益； $\{-z_i | i = 1, \dots, m\}$ 为传递函数的零点； $\{-p_j | j = 1, \dots, n\}$ 为传递函数的极点。传递函数零、极点在复平面中的分布和位置直接决定了系统的品质；

2. 传递函数具有如下性质：

- 传递函数只适用于描述线性定常系统；
- 传递函数与微分方程同属于系统的数学模型，描述输入变量和输出变量间的动态关系。不同的物理系统可以具有相同的传递函数；
- 传递函数表征的是系统本身的一种属性，与输入信号的大小和性质无关；
- 传递函数是系统单位脉冲响应的拉普拉斯变换。

因为传递函数描述了系统的固有属性，因此可以**以其替代微分方程，以代数方程式的形式表示系统，从而通过代数运算建模系统的作用，极大地便利系统分析；**

3. 基于传递函数建模几类基本的系统结构，主要考虑串联、并联、和反馈结构：

串联结构 记两个传递函数分别为 $H_1(s), H_2(s)$ 的子系统串联。输入信号 $U(s)$ 经子系统 $H_1(s)$ 得到输出 $Y_1(s)$ 后再输入子系统 $H_2(s)$ ，最终得到串联系统的最终输出 $Y(s)$ 。则串联系统的传递函数 $H(s)$ 有

$$H(s) = \frac{Y(s)}{U(s)} = \frac{H_2(s)Y_1(s)}{U(s)} = \frac{H_2(s)H_1(s)U(s)}{U(s)} = H_2(s)H_1(s) \quad (\text{串联系统传递函数})$$

将其推广至 n 个子系统串联，则串联系统的传递函数 $H(s) = \prod_{i=0}^{n-1} H_{n-i}(s)$ ；

并联结构 记两个传递函数分别为 $H_1(s), H_2(s)$ 的子系统并联。输入信号 $U(s)$ 经分支（不折减）后分别输入子系统 $H_1(s), H_2(s)$ 得到输出 $Y_1(s), Y_2(s)$ ，再相加（代数和）得到并联系统的最终输出 $Y(s)$ 。则并联系统的传递函数 $H(s)$ 有

$$H(s) = \frac{Y(s)}{U(s)} = \frac{Y_1(s) + Y_2(s)}{U(s)} = \frac{H_1(s)U(s) + H_2(s)U(s)}{U(s)} = H_1(s) + H_2(s) \quad (\text{并联系统传递函数})$$

将其推广至 n 个子系统并联，则并联系统的传递函数 $H(s) = \sum_{i=1}^n H_i(s)$ ；

反馈结构 反馈是指将一个对象的输入信号经某个环节后反向传输到其输入端的过程。记两个传递函数分别为 $G(s), F(s)$ 的子系统反馈连接。其中子系统 $G(s)$ 输出 $Y(s)$ 。 $Y(s)$ 经分支后一方面作为总系统的输出，另一方面反馈输入子系统 $F(s)$ 得到反馈信号 $B(s)$ 。而子系统 $G(s)$ 的输入即为系统总输入 $U(s)$ 与反馈信号 $B(s)$ 的偏差 $E(s)$ 。从而得到反馈系统的整体传递函数，也称为**闭环传递函数**

$$H(s) = \frac{Y(s)}{U(s)} = \frac{G(s)E(s)}{U(s)} = \frac{G(s)(U(s) - F(s)Y(s))}{U(s)} \implies H(s) = \frac{G(s)}{1 + G(s)F(s)} \quad (\text{闭环传递函数})$$

称从偏差信号 $E(s)$ 到输出信号 $Y(s)$ 的通道为前向通道；而从输出信号 $Y(s)$ 到反馈信号 $B(s)$ 的通道为反馈通道，若 $F(s) = 1$ 则称反馈为单位反馈。反馈系统由前向通道和反馈通道组成的闭合回路组成，闭环传递函数由此得名。而如果不考虑最后的闭环，只考虑从偏差信号 $E(s)$ 到反馈信号 $B(s)$ 的过程则子系统 $G(s), F(s)$ 之间呈串联关系，称相应的传递函数为开环传递函数

$$H(s) = G(s)F(s) \quad (\text{反馈系统的开环传递函数})$$

反馈系统对系统内部的参数扰动和系统外部的信号噪声均具有较好的抑制作用。以系统内部参数扰动为例，假设 $G(s)$ 经扰动后变为 $G(s) + \Delta G(s)$ ，记 $\Delta Y_{cl}(s), \Delta Y_{ol}(s)$ 分别表示有、无反馈下系统输出信号的变化，当 $|\Delta G(s)| \ll |G(s)|$ 时有

$$\begin{aligned} \Delta Y_{ol}(s) &= (G(s) + \Delta G(s))U(s) - G(s)U(s) = \Delta G(s)U(s), \\ \Delta Y_{cl}(s) &= \frac{(G(s) + \Delta G(s))U(s)}{1 + (G(s) + \Delta G(s))F(s)} - \frac{G(s)U(s)}{1 + G(s)F(s)} \approx \frac{\Delta G(s)U(s)}{1 + G(s)F(s)} \Rightarrow \frac{\Delta Y_{cl}(s)}{\Delta Y_{ol}(s)} = \frac{1}{1 + G(s)F(s)} \end{aligned}$$

上式表明，增加反馈后系统内部扰动引发的输出扰动较无反馈情况小 $|1 + G(s)F(s)|$ 倍，通常情况下 $|1 + G(s)F(s)|$ 远大于 1。

4. 在控制系统分析中，除了将系统分解为若干个子系统的连接关系外，还需进一步区分各子系统的功能。**实现特定基本功能的子系统被称为典型环节，是系统的基本组成要素。**在系统设计时也常通过在某处增加若干个典型环节而改变系统性能。进一步介绍主要典型环节的传递函数，主要包括**放大环节、惯性环节、积分环节、微分环节、时延环节和振荡环节**：

放大环节 (amplifying element) 放大环节又称比例环节 (proportional element)，是最简单的控制元件。其输出量是以一定比例复现输入信号。在系统中引入放大环节有助于提升系统的灵敏度，但放大信号的同时也会放大扰动和误差，从而影响系统稳定性

$$y(t) = Ku(t) \iff H(s) = K \quad (\text{放大环节})$$

惯性环节 (inertial element) 在放大环节中引入储能元件，使系统不能立刻复现突变的输入信号，而是在输入稳定后输出才逐渐稳定为输入的固定比值，从而提升系统稳定性，但也降低系统响应速度

$$T \frac{dy(t)}{dt} + y(t) = Ku(t) \iff H(s) = \frac{K}{Ts + 1} \quad (\text{惯性环节})$$

其中 K 称为惯性环节的增益； T 称为时间常数，当 $T = 0$ 时惯性环节退化为放大环节；

积分环节 (integral element) 其输出量与输入信号的时间积分成正比。显然在输入结束后积分环节的输出将维持在原值上保持不变，因此积分环节具有记忆性，利用这一特性可用其量化并消除系统的静态误差，从而提升控制精度

$$y(t) = \frac{1}{T} \int_0^t u(\tau) d\tau \iff H(s) = \frac{1}{Ts} \quad (\text{积分环节})$$

其中 T 称为积分时间常数； $K = \frac{1}{T}$ 称为积分环节的增益；

微分环节 (derivative element) 其输出量是输入信号的时间导数成正比，可使系统在输入信号尚未变化时即捕捉其未来的变化趋势，从而实现超前调节、消除震荡，但超前调节也会影响系统的稳定性

$$y(t) = \tau \frac{du(t)}{dt} \iff H(s) = \tau s \quad (\text{微分环节})$$

其中 τ 称为微分时间常数。因为理想的微分环节是非因果的（只有在上帝视角下才可能超前预知输入的变化趋势），不具有物理可实现性，因此实际的微分环节均为近似微分环节，如设计为

$$H(s) = \frac{\tau s}{\tau s + 1} \Rightarrow \lim_{\tau \rightarrow 0} \frac{dH(s)}{ds} = \lim_{\tau \rightarrow 0} \frac{\tau}{(\tau s + 1)^2} = \lim_{\tau \rightarrow 0} \tau$$

在 τ 设计得较小的情况下上述环节即可近似理想微分环节的效果；

延时环节 也称为时滞环节，其输出在经过一段时间的延时后才复现输入信号。最简单的延时环节物理模型是一个输水管道，流入量的变化只有在一定时间后才反应为流出量的变化。延时环节具有一定的黑箱属性，其本身不会是稳定的，但在控制系统中延时的存在可能导致系统不稳定

$$y(t) = u(t - \tau) \iff H(s) = e^{-\tau s} \quad (\text{延时环节})$$

其中 τ 称为延迟时间。其传递函数是一个超越函数，具有无穷多个零点和极点，这也是其对系统稳定性具有较大影响的数学解释；

振荡环节 (oscillation element) 是一个二阶的系统，其输出信号随输入信号周期性上下震荡

$$T^2 \frac{d^2 y(t)}{dt^2} + 2T\zeta \frac{dy(t)}{dt} + y(t) = u(t) \iff H(s) = \frac{1}{T^2 s^2 + 2T\zeta s + 1} \quad (\text{振荡环节})$$

其中 T 为振荡时间常数； ζ 为阻尼比； $\omega_n = \frac{1}{T}$ 为自然角频率。

22.2.2 频率特性函数

- 频率特性函数是经典控制理论中除微分方程和传递函数外的另一种系统数学模型。模型由系统的傅里叶变换分析法引入（见第 21.5 节），聚焦于刻画系统输入与稳态输出（零状态输出）在频域中的相互关系。定义系统的频率特性函数 $H(i\omega)$ 为零初始条件下系统的输出信号的傅里叶变换 $Y(i\omega)$ 与输入信号的傅里叶变换 $U(i\omega)$ 之比

$$H(i\omega) = \frac{Y(i\omega)}{U(i\omega)} = \frac{b_m(i\omega)^m + b_{m-1}(i\omega)^{m-1} + \dots + b_1(i\omega) + b_0}{a_n(i\omega)^n + a_{n-1}(i\omega)^{n-1} + \dots + a_1(i\omega) + a_0} \quad (\text{频率特性函数})$$

可以看到频率特性函数 $H(i\omega)$ 与系统的传递函数 $H(s)$ 有直接的关系，只需将 $H(s)$ 的复变量 s 替换为虚数变量 $i\omega$ 即得到 $H(i\omega)$ ；

- 尽管在数学上 $H(i\omega)$ 仅为 $H(s)$ 的一个特例，但与后者相比 $H(i\omega)$ 具有更清晰的物理意义从而在某些领域中较 $H(s)$ 更有优势。将 $H(i\omega)$ 写为极坐标的形式有

$$H(i\omega) = A(\omega)e^{i\varphi(\omega)}, \quad A(\omega) = \|H(i\omega)\| = \frac{\|Y(i\omega)\|}{\|U(i\omega)\|}, \quad \varphi(\omega) = \text{Arg}(H(i\omega)) = \text{Arg}(Y(i\omega)) - \text{Arg}(U(i\omega))$$

其中 $A(\omega), \varphi(\omega)$ 分别为系统的幅频特性函数和相频特性函数。系统的幅频特性反映了输出幅值与输入幅值之比和角频率 ω 之间的关系，而相频特性反映了输出相角与输入相角之差和角频率 ω 之间的关系。由此特性可知，只需知道系统的幅频特性和相频特性即可得到系统的频率特性函数 $H(i\omega)$ ，从而可以在不知道系统内部机理的情况下通过频率响应实验得到系统的数学模型，这也是引入 $H(i\omega)$ 的主要原因之一；

- 系统幅频特性 $A(\omega)$ 的物理意义可由其定义易得，重点讨论相频特性 $\varphi(\omega)$ 的物理意义。对于因果系统，其响应相对于输入必然存在某种程度的滞后，在时域中表现为时间的滞后而在频域中表现为相角的滞后。具体地，对于频率分量为 ω 的输入信号 $\sin(\omega t + \varphi_u)$ ，记相应的同频率输出信号为 $\sin(\omega t + \varphi_y)$ ，则：

- 当角频率 $\omega > 0$ 时 $\omega t + \varphi$ 随时间增大而增大，因此必然有 $\omega t + \varphi_u \geq \omega t + \varphi_y \implies \varphi_u \geq \varphi_y$ ，即相频特性曲线 $\varphi(\omega)$ 在 $\omega > 0$ 时一般小于 0；
- 当角频率 $\omega < 0$ 时 $\omega t + \varphi$ 随时间增大而减小，因此必然有 $\omega t + \varphi_u \leq \omega t + \varphi_y \implies \varphi_u \leq \varphi_y$ ，即相频特性曲线 $\varphi(\omega)$ 在 $\omega < 0$ 时一般大于 0。

- 进一步关注 $H(i\omega), A(\omega), \varphi(\omega)$ 与 $H(-i\omega), A(-\omega), \varphi(-\omega)$ 的关系。由系统的傅里叶变换分析法可知， $H(i\omega)$ 的另一个定义为系统冲激响应 $h(t)$ 的傅里叶变换，又基于傅里叶变换的尺度变换特性， $H(-i\omega)$ 对应于 $h(-t)$ 的傅里叶变换，则按傅里叶变换定义有

$$\begin{aligned} H(i\omega) &= A(\omega)e^{i\varphi(\omega)} = \int_{-\infty}^{\infty} h(t)e^{-i\omega t} dt = \int_{-\infty}^{\infty} h(t) \cos \omega t dt - i \int_{-\infty}^{\infty} h(t) \sin \omega t dt \\ H(-i\omega) &= A(-\omega)e^{i\varphi(-\omega)} = \int_{-\infty}^{\infty} h(-t)e^{-i\omega t} dt = \int_{-\infty}^{\infty} h(t)e^{i\omega t} dt = \int_{-\infty}^{\infty} h(t) \cos \omega t dt + i \int_{-\infty}^{\infty} h(t) \sin \omega t dt \end{aligned}$$

显然 $H(i\omega), H(-i\omega)$ 关于实轴对称，意味着 $A(\omega)$ 为 ω 的偶函数， $\varphi(\omega)$ 为 ω 的奇函数；

5. $H(i\omega)$ 相较 $H(s)$ 的另一个特别的优势在于前者易于可视化理解。 $H(s)$ 定义在二维复平面上，故其为三维函数。而 $H(i\omega)$ 仅定义在虚轴上：

- 一方面可基于 $A(\omega), \varphi(\omega)$ 直接表示为二维复平面上的复曲线，被称为极坐标频率特性图或奈奎斯特图。又因为 $H(i\omega)$ 与 $H(-i\omega)$ 关于实轴对称，故一般仅绘制 $\omega \in [0, +\infty)$ 的部分即可；
- 另一方面可用两个一维的幅频特性函数 $A(\omega)$ 和相频特性函数 $\varphi(\omega)$ 共同表征 $H(i\omega)$ 。并注意到横坐标 $\omega \in [0, +\infty)$ 范围极宽。如果采用普通坐标均匀分度的话难以展示较宽的频率特性，因此一般由对数坐标绘制对数幅频特性图和对数相频特性图，得到对数坐标频率特征图，也称为伯德 (Bode) 图。需要说明的是对数幅频特性图的纵坐标定义如下，单位为分贝，记作 dB

$$L(\omega) = 20 \lg A(\omega)$$

6. 进一步介绍主要典型环节的频率特性函数，主要包括放大环节、惯性环节、积分环节、微分环节、一阶微分环节、振荡环节、二阶微分环节和时延环节。其中大部分环节的特点均已在第 22.2.1 节中介绍，因此本小节重点从幅频、相频特性的角度介绍：

放大环节 放大环节的幅频和相频特性均为常数，有 $H(i\omega) = Ke^{i0}$, $A(\omega) = K$, $\varphi(\omega) = 0^\circ$ ；

惯性环节 将惯性环节的传递函数 $H(s)$ 的复变量 s 替换为 $i\omega$ ，有

$$H(i\omega) = \frac{1}{i\omega T + 1}, \quad A(\omega) = \frac{1}{\sqrt{1 + \omega^2 T^2}}, \quad \varphi(\omega) = -\arctan(\omega T)$$

积分环节 随着 ω 从 0 趋于 $+\infty$, $H(i\omega)$ 的幅值 $A(\omega)$ 从 $-\infty$ 趋于 0，而相角始终为 -90° 。故积分环节的频率特性是虚轴的下半轴，由无穷远点指向原点

$$H(i\omega) = \frac{1}{i\omega}, \quad A(\omega) = \frac{1}{\omega}, \quad \varphi(\omega) = -\frac{\pi}{2} = -90^\circ$$

微分环节 与积分环节相反，随着 ω 从 0 趋于 $+\infty$, $H(i\omega)$ 的幅值 $A(\omega)$ 从 0 趋于 $+\infty$ ，而相角始终为 90° 。故微分环节的频率特性是虚轴的上半轴，由原点指向无穷远点

$$H(i\omega) = i\omega, \quad A(\omega) = \omega, \quad \varphi(\omega) = \frac{\pi}{2} = 90^\circ$$

一阶微分环节 一阶微分环节的传递函数为 $H(s) = Ts + 1$ ，则相应的频率特性有

$$H(i\omega) = i\omega T + 1, \quad A(\omega) = \sqrt{\omega^2 T^2 + 1}, \quad \varphi(\omega) = \arctan(\omega T)$$

显然其频率特性曲线为一条平行于虚轴的沿虚轴正方向的射线，射线的顶点在 $(1, i0)$ 。因为一阶微分环节的传递函数为惯性环节的倒数，则其对数幅频特性函数与相频特性函数与惯性环节的对应函数关于实轴对称；

振荡环节 记 ω_n 为自然振荡频率， ζ 为阻尼比，并将振荡环节传递函数 $H(s)$ 的复变量 s 替换为 $i\omega$ ，有

$$H(i\omega) = \frac{1}{1 - (\omega/\omega_n)^2 + i2\zeta\omega/\omega_n}, \quad A(\omega) = \frac{1}{\sqrt{[1 - (\omega/\omega_n)^2]^2 + (2\zeta\omega/\omega_n)^2}}, \quad \varphi(\omega) = -\arctan \frac{1 - (\omega/\omega_n)^2}{2\zeta\omega/\omega_n}$$

二阶微分环节 二阶微分环节的传递函数为 $H(s) = \frac{s^2}{\omega_n^2} + 2\zeta\frac{s}{\omega_n} + 1$, $\zeta \in (0, 1)$ ，则相应的频率特性有

$$H(i\omega) = 1 - \left(\frac{\omega}{\omega_n}\right)^2 + i2\zeta\frac{\omega}{\omega_n}, \quad A(\omega) = \sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left(2\zeta\frac{\omega}{\omega_n}\right)^2}, \quad \varphi(\omega) = \arctan \frac{1 - (\omega/\omega_n)^2}{2\zeta\omega/\omega_n}$$

因为二阶微分环节的传递函数为振荡环节的倒数，则其对数幅频特性函数与相频特性函数与振荡环节的对应函数关于实轴对称；

时延环节 易知时延环节的频率特性有 $H(i\omega) = e^{-i\tau\omega}$, $A(\omega) = 1$, $\varphi(\omega) = -\omega\tau$ 。显然 $H(i\omega)$ 在奈奎斯特图中是以复平面原点为圆心，1 为半径的单位圆。

7. 基于典型环节的频率特性即可绘制由各典型环节串联而成的一般系统的开环频率特性。记系统的开环传递函数 $H_0(s)$ 为

$$H_0(s) = H_1(s)H_2(s) \cdots H_n(s)$$

其中各 $H_i(s)$ 为基本环节的传递函数。则系统的开环频率特性函数 $H_0(i\omega)$ 有

$$\begin{aligned} H_0(s) &= H_1(i\omega)H_2(i\omega) \cdots H_n(i\omega) = \left(\prod_i A_i(\omega) \right) e^{\sum_i \varphi_i(\omega)} \\ \Rightarrow A_0(\omega) &= A_1(\omega)A_2(\omega) \cdots A_n(\omega), \quad \varphi_0(\omega) = \varphi_1(\omega) + \varphi_2(\omega) + \cdots + \varphi_n(\omega) \end{aligned}$$

22.2.3 极零图与全通系统、最小相移系统

1. 对一般系统的传递函数的分子和分母多项式作因数分解，可将传递函数写为如下形式

$$H(s) = K \cdot \frac{(s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)}$$

式中 K 为传递函数的传递系数，也是根轨迹增益； $\{z_j | j = 1, \dots, m\}$ 为传递函数的零点； $\{p_j | j = 1, \dots, n\}$ 为传递函数的极点。若忽略传递系数 K ，仅在二维复平面中标记出系统所有零点 $\{z_j | j = 1, \dots, m\}$ 和极点 $\{p_j | j = 1, \dots, n\}$ 的位置，则可得到系统的极零图。显然系统的极零图可在相当程度上替代其传递函数或频率特性函数，从几何而非代数角度直观地反映系统的诸多特性；

2. 首先介绍系统极零图的特点：

- 在一般实际应用中， $H(s)$ 是一个实系数的有理分式，则其极零点要么是实数，要么是一对共轭复数，故极零点一般关于实轴对称；
- 如果将 $H(s)$ 于 $s = \infty$ 处的极、零点考虑在内，则系统的极、零点个数相等。

3. 从时域分析的角度看，极零图中的极点对应系统的特征根，反映了系统自然响应或零输入响应信号的基本模式（自然响应对应受迫响应，零输入响应是自然响应的一部分）。记系统的极点为 $\{p_j | j = 1, \dots, n\}$ ，其零输入响应信号 $y_{zi}(t)$ 具有如下通式

$$y_{zi}(t) = C_1 e^{p_1 t} + C_2 e^{p_2 t} + \cdots + C_n e^{p_n t}$$

其中每一个 $C_j e^{p_j t}$ 被称为系统的一个模态。系统极点与其时域响应之间的关系详见第 21.7 节分析；

4. 从频域分析的角度看，极零图中的极、零点共同反映了系统的幅频和相频特性。为此，不妨将复变量 $s - z_j, s - p_j$ 整体分别记为 $A_j e^{i\zeta_j}, P_j e^{i\pi_j}$ ，其中 Z_j, P_j 分别表示 $s - z_j, s - p_j$ 的幅度， ζ_j, π_j 分别表示 $s - z_j, s - p_j$ 的相位，则 $H(s)$ 可重新表示为

$$H(s) = K \cdot \frac{(s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)} = K \cdot \frac{\prod_{j=1}^m Z_j}{\prod_{j=1}^n P_j} \cdot e^{i(\sum_{j=1}^m \zeta_j - \sum_{j=1}^n \pi_j)}$$

令 $s = i\omega$ ，则可以得到系统的幅频曲线 $A(\omega)$ 和相频曲线 $\varphi(\omega)$ 为

$$A(\omega) = K \cdot \frac{\prod_{j=1}^m Z_j(\omega)}{\prod_{j=1}^n P_j(\omega)}, \quad \varphi(\omega) = \sum_{j=1}^m \zeta_j(\omega) - \sum_{j=1}^n \pi_j(\omega)$$

其中 $Z_j(\omega), \zeta_j(\omega)$ 分别表示以 z_j 为起点以 $(0, i\omega)$ 为终点的向量的模长和关于实轴的偏角；而 $P_j(\omega), \pi_j(\omega)$ 分别表示以 p_j 为起点以 $(0, i\omega)$ 为终点的向量的模长和关于实轴的偏角。由此便可由系统的极零图快速估计系统的幅频曲线和相频曲线。其中幅频特性只与极、零点的实轴坐标绝对值和虚轴坐标取值有关，而相频特性则同时与极、零点的实轴和虚轴坐标取值有关。特别地对于系统的幅频特性 $A(\omega)$ 有：

- 当 $(0, i\omega^*)$ 靠近极点时， $A(\omega)$ 于 $\omega \approx \omega^*$ 处将存在一个峰；
- 当 $(0, i\omega^*)$ 靠近零点时， $A(\omega)$ 于 $\omega \approx \omega^*$ 处将存在一个谷；
- 极、零点越靠近虚轴，相应的峰或谷越尖锐。当极、零点正好位于虚轴上时， $A(\omega)$ 于相应位置处将产生一个无穷大或零的点。

5. 基于极零图与系统频率特性的关系，从极零图角度引出两类特殊的系统——全通系统与最小相移系统：

- 当系统的零点和极点关于虚轴对称分布时称其为全通系统 (**all-pass system**)。当 z_j, p_j 关于虚轴对称时，显然有 $Z(\omega) = P(\omega), \forall \omega$ 。因此当所有零、极点两两关于虚轴对称分布时，系统的幅频特性曲线 $A(\omega) = K$ 为常数，表明系统对输入信号的所有频率分量的幅度均产生相同的增益而不针对性地压缩或放大某一部分，故称“全通”。全通系统可在不改变幅频特性的前提下调整系统相位；
- 当系统的零点和极点全部位于虚轴以左平面时称其为最小相移系统 (**minimum-phase system**)。因为系统的幅频特性与极、零点的实轴坐标绝对值而非取值有关，因此具有相同幅频特性的系统可能具有不同的极零图，从而具有不同的相频特性。又因为在系统稳定的约束下极点仅能分布于虚轴左侧（详见第 22.3 节），故此类系统的极零图差异仅源于零点的位置。首先考虑极点形成的 $\sum_{j=1}^n \pi_j(\omega)$ 。对于所有 n 个极点，因其关于实轴对称且均位于左半平面，故有 $\sum_{j=1}^n \pi_j(0) = 0, \sum_{j=1}^n \pi_j(+\infty) = \frac{2\pi}{n}$ 。如果系统的全部零点同样位于左半平面，则 $\sum_{j=1}^m \zeta_j(\omega)$ 随 ω 由 0 增加至 $+\infty$ 的变化规律与 $\sum_{j=1}^n \pi_j(\omega)$ 高度相似，同样有 $\sum_{j=1}^m \zeta_j(0) = 0, \sum_{j=1}^m \zeta_j(+\infty) = \frac{2\pi}{m}$ 。综上，当系统的零点与极点同处于左半平面时，则由 $\varphi(\omega) = \sum_{j=1}^m \zeta_j(\omega) - \sum_{j=1}^n \pi_j(\omega)$ ，其零点可最大限度地抵消极点对系统相频特性的影响，从而使相频曲线尽可能接近实轴，即在具有相同幅频特性的系统中对输入信号产生的相位偏移最小，故称“最小相移”。

22.3 系统的 BIBO 稳定与劳斯-霍尔维茨判据

1. 系统的稳定性是控制理论的研究重点。根据系统数学模型的特点，稳定系统也具有多种定义方法。经典的控制理论以传递函数 $H(s)$ 或关于输入信号 $u(t)$ 与输出信号 $y(t)$ 的微分方程作为系统的数学模型，**重在关注系统对输入的响应，由此出发定义的稳定性被称为系统的外部稳定性**。具体地，系统的外部稳定性定义为对任意一个有界的输入 $u(t), |u(t)| < \infty$ ，对应的系统输出 $y(t)$ 也均有界 $|y(t)| < \infty$ ，此类稳定也被称为有界输入-有界输出 (**bounded input-bounded output, BIBO**) 稳定；

2. 对于线性定常系统，其 BIBO 稳定的充要条件是系统的冲激响应 $h(t)$ 绝对可积，即满足 $\int_{-\infty}^{\infty} |h(t)| dt < +\infty$ 。首先证明其充分性。假设 $\int_{-\infty}^{\infty} |h(t)| dt < +\infty$ 成立，由第 21.2 节介绍的系统时域分析法可知，线性定常系统对任意输入 $u(t)$ 的零状态响应 $y(t)$ 有

$$|y(t)| = \left| \int_{-\infty}^{\infty} u(\tau) h(t-\tau) d\tau \right| \leq \int_{-\infty}^{\infty} |u(\tau) h(t-\tau)| d\tau = \int_{-\infty}^{\infty} |u(\tau)| \cdot |h(t-\tau)| d\tau \leq |u(t)|_{\max} \cdot \int_{-\infty}^{\infty} |h(t-\tau)| d\tau$$

只考虑零状态响应而不考虑零输入响应是因为第 21.7 节介绍的系统拉氏变换分析法中已指出零状态响应即可同时体现系统属性和输入的影响，而零输入响应只与系统属性有关，只要零状态响应恒有界则零输入响应必然恒有界。关注上式，因为 $u(t)$ 有界，且冲激响应 $h(t)$ 绝对可积，则系统输出 $y(t)$ 也有界。充分性得证。进一步证明必要性，即证只要系统 BIBO 稳定，则冲激响应 $h(t)$ 绝对可积。采用反证法，证明 $\int_{-\infty}^{\infty} |h(t)| dt = \infty$ 时存在有界输入得到无界输出的情况。构造如下有界输入信号 $u(t)$ 即可使得相应的响应 $y(t)$ 无界，必要性得证

$$y(0) = \int_{-\infty}^{\infty} u(\tau) h(-\tau) d\tau = \int_{-\infty}^{\infty} |h(-\tau)| d\tau = \infty, \quad u(t) = \operatorname{sgn}[h(-t)] = \begin{cases} 1 & h(-t) \geq 0 \\ -1 & h(-t) < 0 \end{cases}$$

3. 按上述充要条件评估系统的 BIBO 稳定需要得到系统的冲激响应 $h(t)$ 再求绝对积分，计算难度较大。而第 21.7 节又指出系统的传递函数 $H(s)$ 即为冲激响应 $h(t)$ 的拉氏变换，因此可通过 $H(s)$ 的性质判断 $h(t)$ 的性质，进而判断系统的 BIBO 稳定性。由拉氏变换性质（见第 21.6、21.7 节）可知， $H(s)$ 的极点位置决定了信号 $h(t)$ 的子信号模式，所有子信号的线性叠加即为 $h(t)$ 。因此，为使得 $h(t)$ 的绝对积分有限，则要求 $h(t)$ 的所有子信号绝对积分均有限，则：

- 若 $H(s)$ 的所有极点均位于虚轴左侧，则 $h(t)$ 的所有子信号振幅均随时间指数衰减，系统 BIBO 稳定；
- 若 $H(s)$ 存在虚轴右侧的极点，则 $h(t)$ 存在振幅随时间指数放大的子信号，不满足 BIBO 稳定；
- 若 $H(s)$ 存在虚轴处的高阶极点，则 $h(t)$ 存在振幅随时间按幂函数放大的子信号，不满足 BIBO 稳定；

- 若 $H(s)$ 存在虚轴处的单阶极点，则 $h(t)$ 存在振幅随时间不变的子信号，不满足绝对可积条件。但系统对于大部分的有界输入仍可保证有界输出，仅对于极点恰好与 $H(s)$ 虚轴上的单极点重合的有界输入无法得到有界输出，而是随时间振荡放大，称为临界稳定。除理论分析场景外，临界稳定系统在大多数情况下均被归为不稳定，因为实际工作中只要系统参数略有变化就可能转为不稳定系统。
4. 上述分析将系统 BIBO 稳定性的判断问题转化为判断 $H(s)$ 极点（即分母函数 $D(s)$ 的零点）位置的问题。一般情况下 $D(s)$ 为多项式。注意到高阶多项式不存在求根公式，只能通过数值计算方法估计零点，计算量较大，且只能判断系统稳定性而无法指导含参系统的稳定性设计。**一个理想的思路是——寻找一种解析方法，无需求根仅根据多项式 $D(s)$ 的各项系数即可判断其零点的位置；**
5. 记 $D(s)$ 的数学形式如下。易证若 $D(s)$ 的所有系数 $\{a_i | i = 0, \dots, n\}$ 不同号，则 $D(s)$ 必然存在虚轴右侧的零点，系统必然不稳定

$$D(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0$$

若 $D(s)$ 的所有系数 $\{a_i | i = 0, \dots, n\}$ 同号，则需进一步基于劳斯-霍尔维茨稳定性判据 (**Routh-Hurwitz criterion**) 判断 $D(s)$ 零点的位置。劳斯-霍尔维茨判据，又称为代数稳定性判据，是劳斯于 1877 年提出的稳定性判据，可无需解方程而判定一个多项式方程中是否存在位于复平面右半部的正根，由此劳斯获得了亚当奖。劳斯-霍尔维茨判据的核心是计算如下劳斯表：

s^n	$A_{n,0}$	$A_{n,1}$	$A_{n,2}$	$A_{n,3}$	\dots
s^{n-1}	$A_{n-1,0}$	$A_{n-1,1}$	$A_{n-1,2}$	$A_{n-1,3}$	\dots
s^{n-2}	$A_{n-2,0}$	$A_{n-2,1}$	$A_{n-2,2}$		
s^{n-3}	$A_{n-3,0}$	$A_{n-3,1}$	$A_{n-3,2}$	$A_{n,k} = a_{n-2k}$, $A_{n-1,k} = a_{n-(2k+1)}$, $A_{n-j,k} = -\frac{1}{A_{n-j+1,0}} \begin{vmatrix} A_{n-j+2,0} & A_{n-j+2,k+1} \\ A_{n-j+1,0} & A_{n-j+1,k+1} \end{vmatrix}$	
\vdots	\vdots	\vdots	\vdots		
s^2	$A_{2,0}$	$A_{2,1}$	0		
s^1	$A_{1,0}$	0	0		
s^0	$A_{0,0}$	0	0		

对于 n 阶多项式，劳斯表包含 $n+1$ 行。其中第 n 行和第 $n-1$ 行元素由 $D(s)$ 多项式的系数 $\{a_i | i = 0, \dots, n\}$ 决定，而从 $n-2$ 行开始每行元素由上两行元素决定。如果劳斯表计算正确，则必然有 $A_{0,0} = a_0$ 。劳斯表的第一列元素称为劳斯-霍尔维茨数列。**劳斯-霍尔维茨判据指出——劳斯-霍尔维茨数列系数同号等价于 $D(s)$ 只存在位于虚轴左侧的零点，若不同号，则数列的符号变化次数等于 $D(s)$ 存在于虚轴右侧的零点的数目。**因为劳斯表各元素都有理论定义式，故劳斯-霍尔维茨判据不仅可用于判定系统稳定性，还有助于稳定系统的设计；

6. 最后介绍劳斯表计算时的两类特殊情况。由定义可知，劳斯表计算时要求第一列元素非 0，因此当：
- 当某一行元素的首元素 $A_{n-j,0} = 0$ ，而该行其它元素不全为 0 时，令 $A_{n-j,0} = \varepsilon$ 后按定义计算其它元素，其中 ε 为一个正的无穷小量；
 - 当某一行的所有元素均为 0 时，意味着系统在虚轴上存在极点，此时系统至多为临界稳定。若需进一步区分系统为临界稳定或不稳定，假设第 $n-j$ 行元素为零，则基于上一行元素构建辅助多项式

$$A(s|n-j+1) = \sum_{k \geq 0} A_{n-j+1,k} \cdot s^{n-j-2k+1}$$

辅助多项式的最高次幂为 $n-j+1$ ，其它项的幂按 2 为公差递减，各项系数为劳斯表中第 $n-j+1$ 行的相应元素。将辅助多项式导函数的系数设为第 $n-j$ 行元素，即 $A_{n-j,k} = A_{n-j+1,k} \cdot (n-j-2k+1)$ 。而后按定义计算劳斯表其它元素。若最终劳斯-霍尔维茨数列系数存在变号，则系统存在虚轴右侧的极点，不稳定；若劳斯-霍尔维茨数列系数无变号，则系统至多存在虚轴上的极点，且相关极点与辅助多项式 $A(s|n-j+1)$ 的根对应——若 $A(s|n-j+1)$ 在虚轴上的根均为单根，则系统在虚轴上的根也为单根，系统临界稳定；若 $A(s|n-j+1)$ 存在虚轴上的重根，则系统也存在虚轴上的重根，不稳定。

22.4 奈奎斯特 (Nyquist) 判据与稳定裕度

22.4.1 闭环系统稳定的奈奎斯特判据

- 经典控制理论的稳定性判别方法包括劳斯判据（第 22.3 节）、根轨迹法和奈奎斯特判据三大方法。劳斯判据和根轨迹法均基于系统的传递函数 $H(s)$ ，但实际中有些系统的传递函数是写不出来的。1933 年奈奎斯特从频域角度提出了奈奎斯特判据，基于系统的开环频率特性函数分析闭环系统的稳定性。与 $H(s)$ 相比，系统的频率特性 $H(i\omega)$ 除解析推导外还可简单地通过实验获得。因此对于不能列出微分方程的系统可由奈奎斯特判据分析其稳定性。另外其不仅可判别闭环系统的稳定性，还能指出稳定的程度；
- 为严格给出奈奎斯特判据，仍从系统的传递函数出发。记闭环系统的闭环传递函数和开环传递函数分别为 $H(s), H_0(s)$ ，则有

$$H(s) = \frac{H_0(s)}{1 + H_0(s)} = \frac{B(s)}{A(s) + B(s)}, \quad H_0(s) = \frac{B(s)}{A(s)}$$

其中 $B(s), A(s)$ 分别表示系统开环传递函数的分子和分母多项式，则 $A(s), A(s) + B(s)$ 分别构成开环系统和闭环系统的特征多项式。基于 $A(s), A(s) + B(s)$ 构建如下辅助函数

$$F(s) = \frac{A(s) + B(s)}{A(s)} = 1 + H_0(s)$$

显然 $F(s)$ 的零点即为系统的闭环极点，而 $F(s)$ 的极点为系统的开环极点。为使得系统稳定，等价于要求 $F(s)$ 的零点全部位于复平面的左半平面；

- 奈奎斯特判据的理论基础为复变函数理论中的幅角原理（见第 23.16.2 节）。幅角原理指出，如果复平面内的封闭正向围线 γ 能包围整个右半平面，并记 γ 在 $F(s)$ 的映射围线为 Γ' ，则有

$$N = Z - P \quad (\text{奈奎斯特判据})$$

式中 N 表示映射围线 Γ' 绕原点的圈数； Z 表示 $F(s)$ 在右半平面上的零点数（即闭环系统于右半平面上的极点数）； P 表示 $F(s)$ 在右半平面上的极点数（即开环系统于右半平面上的极点数）。若 $H_0(s), \gamma$ 已知，则 Γ', N, P 可知，则由上式可推出 Z 。**当且仅当 $Z = 0$ 时系统稳定，这便是奈奎斯特判据；**

- 至此，为应用奈奎斯特判据，问题转嫁为如何构建包围复平面右半平面的围线 γ 以及如何确定开环传递函数 $H_0(s)$ ：

- 为直观起见假设围线 γ 的正方向为顺时针（对应地 N 表示映射围线 Γ' 顺时针绕原点的圈数）。假设 $F(s)$ 不存在虚轴上的零、极点。则令 γ 为以原点为圆心、 R 为半径的位于虚轴右侧的半圆。具体地 γ 包含两条路径

$$\gamma = \gamma_1 + \gamma_2$$

γ_1 表示沿虚轴正方向从 $0 - iR$ 到 $0 + iR$ 的直线； γ_2 表示沿顺时针方向以 R 为半径从 $0 + iR$ 到 $0 - iR$ 的半圆弧。当 $R \rightarrow +\infty$ 时 γ 即可包围整个右半平面，此时称其为奈奎斯特围线；

- 进一步讨论奈奎斯特围线 γ 经 $F(s)$ 的映射围线 Γ' 。记 γ 经 $H_0(s)$ 的映射围线为 Γ ，显然 Γ' 为 Γ 右偏一个单位，因此只需讨论 Γ 即可。首先关注 γ_1 经 $H_0(s)$ 的映射 Γ_1 。当 s 沿虚轴正方向从 $0 - i\infty$ 到 $0 + i\infty$ 时 Γ_1 即为开环系统的频率特性曲线 $H_0(i\omega)$ ，且起终点均收敛于坐标原点；当 s 沿顺时针方向以 $R \rightarrow +\infty$ 为半径从 $0 + i\infty$ 到 $0 - i\infty$ 时，恒有 $H_0(s) \rightarrow 0$ ，即 Γ_2 收敛于原点。因此 γ 经 $H_0(s)$ 的映射围线 Γ 实际上即是开环系统的频率特性曲线 $H_0(i\omega)$ ，由此可绕过开环传递函数 $H_0(s)$ 。并以 Γ 替代 Γ' ，则奈奎斯特判据中 N 的物理意义变为 Γ 绕点 $-1 + i0$ 的圈数（顺时针为正，逆时针为负）；
- 因为开环系统为串联系统，故可通过实验测得每一单元的幅频特性和相频特性得到整个开环系统的频率特性 $H_0(i\omega)$ 。同样因为开环系统结构更为简单，其右端极点数目 P 也通常可以确定。

以上即为奈奎斯特判据的推导及应用过程——对于给定闭环系统，首先分析其开环右极点数 P ，并判断其开环频率特性函数 $H_0(i\omega)$ 顺时针绕点 $-1 + i0$ 的圈数 N ，从而确定系统的闭环右极点数 Z 以判断闭环系统的稳定性。

22.4.2 稳定裕度——相位裕度与幅值裕度

- 奈奎斯特判据不仅能判别闭环系统的稳定性，而且还能指出稳定的程度，后者是奈奎斯特判据的重要优点，有着重要的实际意义。因为在系统数学建模时不可避免地存在误差，再加上硬件老化等影响实际系统的运行状态往往与数学模型存在偏差。工程中一般采用安全系数的方法处理这一问题，在控制系统设计时留有一定的稳定裕度，保证系统实际工作中的鲁棒性。另外系统的稳定裕度还与系统的瞬态性有密切关系。稳定裕度包括相位裕度和幅值裕度；
- 考虑一类特殊的系统——最小相移系统（见第 22.2.3 节）。最小相移系统一般定义为零、极点均在虚轴左侧的系统，而具体在控制论中，闭环控制系统的最小相移系统指开环零、极点均在虚轴左侧的系统。显然此类系统有 $P = 0$ 。为使得闭环系统稳定（即 $Z = 0$ ），则按奈奎斯特判据必然要求其开环频率特性函数 $H_0(i\omega)$ 顺时针绕点 $-1 + i0$ 的圈数 $N = 0$ ；
- 假设闭环系统稳定。首先考虑对最小相移系统引入额外的相移 γ_0 使得开环频率特性函数变为 $H'_0(i\omega) = H_0(i\omega)e^{-i\gamma_0}$ ，表示将原本的频率特性曲线 $H_0(i\omega)$ 顺时针旋转 γ_0 角度。引入相移并不会增加开环系统的右半区极点（因为引入相移并不会使开环系统本身不稳定），即 P 不变。但原本不包围点 $-1 + i0$ 的频率特性曲线 $H_0(i\omega)$ 可能在旋转后包围点 $-1 + i0$ ，导致 $N > 0$ ，意味着闭环系统变为不稳定系统；
- 在此基础上即可引入相位裕度 γ 的概念——如果一个最小相移系统再引入 γ 的相位延迟将变为临界稳定状态，则称其相位裕度为 γ 。所谓临界稳定是指 $H_0(i\omega)$ 顺时针旋转 γ 后恰好经过点 $-1 + i0$ ，则可得到 γ 的定义式为

$$\gamma = \pi + \text{Arg}\{H_0(i\omega_c)\}, \quad \|H_0(i\omega_c)\| = 1, \quad \omega_c \geq 0 \quad (\text{相位裕度})$$

式中 ω_c 是使得 $H_0(i\omega_c)$ 幅度为 1 的角频率，被称为系统的截止频率； $\text{Arg}\{\cdot\}$ 表示复向量的幅角。显然对于最小相位系统而言：

- 其相位裕度为正，则闭环系统稳定；反之则系统不稳定；
- 其相位裕度越大，则系统越稳健。由此即可分析系统的稳定程度。

- 相位裕度表述了开环系统相位延迟对闭环系统稳定性的影响程度，对应地也可分析幅度增益的影响。考虑将最小相移系统的开环频率特性幅值放大 k 倍变为 $H'_0(i\omega) = kH_0(i\omega)$ 。幅值放大并不会增加开环系统的右半区极点，即 P 不变。但原本不包围点 $-1 + i0$ 的频率特性曲线 $H_0(i\omega)$ 可能在幅度放大后包围点 $-1 + i0$ ，导致 $N > 0$ ，意味着闭环系统变为不稳定系统。在此基础上即可引入幅值裕度 k_g 的概念——如果一个最小相移系统的幅值放大 k_g 倍将变为临界稳定状态，则称其幅值裕度为 k_g 。进而得到 k_g 的定义式为

$$k_g = \frac{1}{\|H_0(i\omega_g)\|}, \quad \text{Arg}\{H_0(i\omega_g)\} = -\pi, \quad \omega_g \geq 0 \quad (\text{幅值裕度})$$

显然对于最小相位系统而言：

- 其幅值裕度大于 1，则闭环系统稳定；反之则系统不稳定；
- 其幅值裕度越大，则系统越稳健。由此即可分析系统的稳定程度。

- 需要说明的是，因为最小相移系统是指相同幅频特性下相移最小的系统，故其幅频特性曲线和相频特性曲线存在一一对应关系。因此只要最小相位系统满足相位裕度判据，则其一般也将满足幅值裕度判据，反之亦然（但存在某些特例）；
- 稳定裕度判据可作为最小相移系统稳定性评价的充要条件，但对于非最小相移系统则只能作为必要不充分条件。非最小相移系统依然存在相位裕度和幅值裕度的概念，但此时仅特指系统对特定频率 ω_c, ω_g 的输入信号的输出稳定特性。

22.5 现代控制理论及控制系统的状态空间表达式

- 控制理论可分为经典控制理论与现代控制理论：
 - 经典控制理论以单变量线性定常系统为主要研究对象，以频域法为研究控制系统动态特性的主要方法，以各种图表（Bode 图、Nyquist 曲线、根轨迹等）为系统分析和综合的主要工具；

- 现代控制理论以多变量线性、非线性系统为主要研究对象，以时域法特别是以状态空间法为主要研究方法，研究内容包括线性系统理论、非线性系统理论、最优控制、鲁棒控制、自适应控制、大系统理论、深度学习、智能控制等。
2. 经典控制理论中常以传递函数 $G(s)$ 作为系统的数学模型，线性定常系统的传递函数是指在初始状态为 0 的条件下，系统输出变量 $y(t)$ 的拉氏变换 $Y(s)$ 与输入变量 $u(t)$ 的拉氏变换 $U(s)$ 的比值，即

$$G(s) = \frac{Y(s)}{U(s)}$$

传递函数仅描述系统输入与输出之间的关系，而不考虑系统的内部结构；

3. 现代控制理论中以状态空间表达式作为系统的数学模型，状态空间表达式不仅关心系统输入与输出之间的关系，也能描述系统的内部结构。首先介绍相关定义：

- 状态变量：足以完全表征系统运动状态的最少个数的一组变量称为状态变量。所谓“完全表征”即只要给定状态变量的初值 $x(t_0)$ 以及 $t \geq t_0$ 时间段内的输入 $u(t)$ ，就能够完全确定系统在 $t \geq t_0$ 任意时间内的动态行为。所谓“最小”体现在减小变量个数就不能完全表征系统的动态行为，而增加变量则是完全表征系统动态行为所不需要的。进一步地对状态变量作几点说明：
 - 状态变量相互独立；
 - 对于实际的物理系统，状态变量的个数应大于等于系统中独立储能元件的个数，常常直接等于独立储能元件的个数，在此基础上增加合适的状态变量可更精细地描述系统的运动行为。对于某些系统，若不考虑具体的物理意义，理论上可以选取无穷多个状态变量；
 - 对于同一个动态系统，在内涵精度描述相同的情况下，状态变量的选取不是唯一的，但状态变量的个数是唯一的。
- 状态向量：由系统状态变量构成的向量即为系统的状态向量，常常写成列向量。因为状态变量的选取是非唯一的，故同一系统的状态向量也是非唯一的，又因为不同的状态向量均能完全地表征同一系统，故同一系统不同状态向量之间必然存在一种线性变换的关系；
- 状态空间：以状态向量的每一个分量 $x_1(t), x_2(t), \dots, x_n(t)$ 未坐标轴所构成的空间即为状态空间；
- 状态轨迹：系统状态随时间变化的过程，在状态空间中描绘出一条轨迹，即为状态轨迹；
- 状态方程：由系统状态变量构成的描述系统动态过程的一阶微分方程组称为系统的状态方程，用于描述系统输入 u 引起系统状态 x 变化的动态过程，一般形式为

$$\dot{x} = Ax + Bu$$

对于 n 个状态变量、 r 个输入的动态系统，有 $A \in \mathbf{R}^{n \times n}$ 为系统矩阵，表征系统的内在联系； $B \in \mathbf{R}^{n \times r}$ 为控制矩阵，表征输入对状态的作用；

- 输出方程：描述系统输入 u 和系统状态 x 对系统输出 y 的代数方程，一般形式为

$$y = Cx + Du$$

对于 n 个状态变量、 r 个输入、 m 个输出的动态系统，有 $C \in \mathbf{R}^{m \times n}$ 为输出矩阵，表征状态变量对输出的影响； $D \in \mathbf{R}^{m \times r}$ 为直接传输矩阵，表征输入对输出的作用，常常令 $D = 0$ ；

- 状态空间表达式：将状态方程与输出方程组合起来即为系统的状态空间表达式，即能表示输入对系统状态的影响，又能表示输入和系统状态对输出的影响，一般形式为

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

4. 对于离散系统，其状态空间表达式可用差分方程组表示

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

5. 若系数矩阵 $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ 中至少有一个元素是时间 t 的函数，则系统为线性时变系统，其状态空间表达式为

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} + \mathbf{B}(t)\mathbf{u} \\ \mathbf{y} = \mathbf{C}(t)\mathbf{x} + \mathbf{D}(t)\mathbf{u} \end{cases}$$

6. 非线性系统又可分为非线性时变系统和非线性定常系统，当状态空间表达式显含时间 t 时则称其为非线性时变系统，反之为非线性定常系统。记 f, g 为向量函数，有

$$\text{非线性时变系统: } \begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) \\ \mathbf{y} = g(\mathbf{x}, \mathbf{u}, t) \end{cases} \quad \text{非线性定常系统: } \begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = g(\mathbf{x}, \mathbf{u}) \end{cases}$$

当非线性系统在某个平衡状态周围小范围变化时，可将通过一阶泰勒展开将系统线性化。当系统处于平衡状态时，有 $\dot{\mathbf{x}}_0 = f(\mathbf{x}_0, \mathbf{u}_0) = 0$ ，此时对非线性函数 f, g 在 $\mathbf{x}_0, \mathbf{u}_0$ 附近做一阶泰勒展开并舍去余项，有

$$f(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}_0, \mathbf{u}_0) + \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \delta \mathbf{x} + \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \delta \mathbf{u} \quad g(\mathbf{x}, \mathbf{u}) = g(\mathbf{x}_0, \mathbf{u}_0) + \left. \frac{\partial g}{\partial \mathbf{x}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \delta \mathbf{x} + \left. \frac{\partial g}{\partial \mathbf{u}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \delta \mathbf{u}$$

则非线性系统的一次线性化方程可表示为

$$\begin{cases} \delta \dot{\mathbf{x}} = \dot{\mathbf{x}} - \dot{\mathbf{x}}_0 = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \delta \mathbf{x} + \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \delta \mathbf{u} \\ \delta \mathbf{y} = \mathbf{y} - \mathbf{y}_0 = \left. \frac{\partial g}{\partial \mathbf{x}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \delta \mathbf{x} + \left. \frac{\partial g}{\partial \mathbf{u}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \delta \mathbf{u} \end{cases} \xrightarrow[\tilde{\mathbf{y}} = \delta \mathbf{x}, \tilde{\mathbf{u}} = \delta \mathbf{u}]{\tilde{\mathbf{y}} = \mathbf{y}} \begin{cases} \dot{\tilde{\mathbf{x}}} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \tilde{\mathbf{x}} + \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \tilde{\mathbf{u}} = A \tilde{\mathbf{x}} + B \tilde{\mathbf{u}} \\ \tilde{\mathbf{y}} = \left. \frac{\partial g}{\partial \mathbf{x}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \tilde{\mathbf{x}} + \left. \frac{\partial g}{\partial \mathbf{u}} \right|_{\mathbf{x}_0, \mathbf{u}_0} \tilde{\mathbf{u}} = C \tilde{\mathbf{x}} + D \tilde{\mathbf{u}} \end{cases}$$

22.5.1 状态空间表达式的实现——从传递函数到状态空间表达式

系统的状态空间表达式可由系统方框图得到，也可基于物理定律直接推导，还可基于系统的微分方程和传递函数得到。以下介绍由传递函数 $G(s)$ 得到系统状态空间表达式的方法，若系统由微分方程或方框图表示可先将其转为传递函数。需要注意的是，因为系统的传递函数是唯一的，但状态变量的选取可不唯一，故存在多种转换方式，以下介绍的仅为较主流的方法。考虑传递函数的一般形式

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}$$

1. 当 $m < n$ 时，引入中间变量 $\tilde{Y}(s)$

$$\tilde{Y}(s) = \frac{1}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} U(s) \implies \begin{cases} U(s) = \tilde{Y}(s) \cdot (s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0) \\ Y(s) = \tilde{Y}(s) \cdot (b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0) \end{cases}$$

对以上两等式分别取拉氏反变换¹，有

$$u = \tilde{y}^{(n)} + a_{n-1} \tilde{y}^{(n-1)} + \cdots + a_1 \tilde{y}^{(1)} + a_0 \tilde{y} \quad y = b_m \tilde{y}^{(m)} + b_{m-1} \tilde{y}^{(m-1)} + \cdots + b_1 \tilde{y}^{(1)} + b_0 \tilde{y}$$

因为微分方程最高阶为 n ，故选取 n 个状态变量 $x_1 = \tilde{y}, x_2 = \tilde{y}^{(1)}, \dots, x_n = \tilde{y}^{(n-1)}$ ，从而可得状态方程

$$\begin{cases} \dot{x}_1 = x_2 \\ \vdots \\ \dot{x}_{n-1} = x_n \\ \dot{x}_n = u - a_{n-1} x_n - \cdots - a_1 x_2 - a_0 x_1 \end{cases} \implies \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

同样地可得到输出方程

$$y = b_m x_{m+1} + b_{m-1} x_m + \cdots + b_1 x_2 + b_0 x_1 \implies y = [b_0 \quad \cdots \quad b_m \quad 0 \quad \cdots \quad 0] \mathbf{x}$$

以上即为系统的状态空间表达式。注意到：

¹ 记 $\mathcal{L}^{-1}[\tilde{Y}(s)] = \tilde{y}(t)$ ，且注意到拉氏反变换的微分特性 $\mathcal{L}^{-1}[s^n \tilde{Y}(s)] = \tilde{y}^{(n)}(t)$

- 系统矩阵 A 的前 $n - 1$ 行为次对角单位阵，最后一行由系统特征多项式系数的相反数组成；
- 输入矩阵 B 的最后一个元素为 1，其余元素全为 0；
- 输出函数中直接传输矩阵 $D = \mathbf{0}$ ；
- 具有如上系统矩阵 A 和输入矩阵 B 的状态空间表达式称为系统的能控标准 I 型。

2. 当 $m = n$ 时，对传递函数作分数除法

$$G(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} = b_n + \frac{(b_{n-1} - b_n a_{n-1}) s^{n-1} + \cdots + (b_1 - b_n a_1) s + (b_0 - b_n a_0)}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}$$

同样地引入中间变量 $\tilde{Y}(s)$

$$\begin{aligned} \tilde{Y}(s) &= \frac{1}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} U(s) \\ \Rightarrow &\begin{cases} U(s) = \tilde{Y}(s) \cdot (s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0) \\ Y(s) = \tilde{Y}(s) \cdot [(b_{n-1} - b_n a_{n-1}) s^{n-1} + \cdots + (b_1 - b_n a_1) s + (b_0 - b_n a_0)] + b_n U(s) \end{cases} \end{aligned}$$

同样地进行拉氏反变换，并选取同样的状态变量，得到状态表达式如下

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [(b_0 - b_n a_0) \quad (b_1 - b_n a_1) \quad \cdots \quad (b_{n-1} - b_n a_{n-1})]x + b_n u$$

注意到此时输出函数中直接传输矩阵 $D \neq \mathbf{0}$ ；

3. 当 $m > n$ 时，得到的输出函数将不再是代数方程，而是一个包含输入信号微分项的微分方程，这在实际系统中是不允许的，因为这意味着系统的输出将因输入信号的变化产生较大的噪声。

22.5.2 状态向量的线性变换

1. 因为同一系统的不同状态向量之间必然存在一种线性变换关系，则对于一个 n 阶系统²，记状态向量为 x ，可得到系统的另一个状态向量 $z = T^{-1}x$ ， $T \in \mathbb{R}^{n \times n}$ 为任意非奇异矩阵；

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \xrightarrow{z=T^{-1}x} \begin{cases} \dot{z} = T^{-1}ATz + T^{-1}Bu \\ y = CTz + Du \\ z_0 = T^{-1}x_0 \end{cases}$$

2. 在经典控制论中，系统的性能主要由传递函数的极点决定，也就是由系统特征方程的系数决定，而状态空间表达式中系统矩阵 A 的特征根与传递函数的极点相对应，因此在现代控制理论中，决定系统性能的主要是系统矩阵 A 的特征值，称其为系统的特征值，系统特征值也就是特征方程 $|\lambda I - A| = 0$ 的根，显然 n 阶系统有 n 个特征值；

3. 对于系统的特征值 λ_i 则有系统的特征向量 p_i 。一个特征值对应无穷多个特征向量。根据定义 λ_i, p_i 满足 $(A - \lambda_i I)p_i = \mathbf{0}$ ；

4. 在对系统状态向量进行非奇异线性变换时，系统矩阵也会随之改变，但系统的特征值不变，称为系统特征值的不变性，因此对系统状态向量进行非奇异线性变换不改变系统性能。欲证明特征值不变，只需证明特征方程本质不变，由定义得到变换后的特征方程为

$$|\lambda I - T^{-1}AT| = 0 \implies |\lambda T^{-1}T - T^{-1}AT| = 0 \implies |T^{-1}\lambda T - T^{-1}AT| = 0 \implies |T^{-1}(\lambda I - A)T| = 0$$

² 所谓 n 阶系统即状态向量为 $n \times 1$ ，系统矩阵 $A \in \mathbb{R}^{n \times n}$ 。

由线性代数规则：矩阵乘积的行列式等于矩阵行列式的乘积，且非奇异矩阵的行列式不为 0，则

$$|\mathbf{T}^{-1}| \cdot |\lambda\mathbf{I} - \mathbf{A}| \cdot |\mathbf{T}| = 0 \xrightarrow[|\mathbf{T}^{-1}| \neq 0]{|\mathbf{T}| \neq 0} |\lambda\mathbf{I} - \mathbf{A}| = 0$$

显然非奇异线性变换后的特征方程的本质不变，故线性变换不改变特征根；

5. 对于给定系统 $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ ，设其特征值 $\lambda_1, \dots, \lambda_n$ 两两互异，由它们的特征向量组成变换阵 $\mathbf{T} = [\mathbf{p}_1, \dots, \mathbf{p}_n]$ ，则系统的状态方程在变换 $\mathbf{z} = \mathbf{T}^{-1}\mathbf{x}$ 下将化为如下形式，称为对角规范型

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \xrightarrow[T=(\mathbf{p}_1, \dots, \mathbf{p}_n)]{z=\mathbf{T}^{-1}\mathbf{x}} \dot{\mathbf{z}} = \mathbf{T}^{-1}\mathbf{ATz} + \mathbf{T}^{-1}\mathbf{Bu} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \mathbf{z} + \mathbf{T}^{-1}\mathbf{Bu}$$

显然只需证明 $\mathbf{T}^{-1}\mathbf{AT}$ 等于以系统特征值为对角元素的对角阵。根据定义

$$\begin{aligned} \mathbf{T}^{-1}\mathbf{AT} &= [\mathbf{p}_1 \ \cdots \ \mathbf{p}_n]^{-1} \mathbf{A} [\mathbf{p}_1 \ \cdots \ \mathbf{p}_n] = [\mathbf{p}_1 \ \cdots \ \mathbf{p}_n]^{-1} [A\mathbf{p}_1 \ \cdots \ A\mathbf{p}_n] \\ &= [\mathbf{p}_1 \ \cdots \ \mathbf{p}_n]^{-1} [\lambda_1\mathbf{p}_1 \ \cdots \ \lambda_n\mathbf{p}_n] \\ &= [\mathbf{p}_1 \ \cdots \ \mathbf{p}_n]^{-1} [\mathbf{p}_1 \ \cdots \ \mathbf{p}_n] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \end{aligned}$$

在对角规范型下，系统各状态变量实现完全解耦，状态方程可表示为 n 个独立的状态变量方程。另外，当原状态方程为能控标准型时，则称将其变换为对角标准型的变换阵 \mathbf{T} 为范德蒙德阵

$$\mathbf{T} = \begin{bmatrix} 1 & \cdots & 1 \\ \lambda_1 & \cdots & \lambda_n \\ \vdots & \vdots & \vdots \\ \lambda_1^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}$$

需要注意的是，为将状态方程转变为对角规范型要求系统矩阵无重复的特征根，因为当存在重复的特征根时变换矩阵 \mathbf{T} 将是奇异矩阵，无法求逆；

6. 当系统的特征值存在重根时，只能通过非奇异线性变换将系统矩阵变换为约当规范型。首先引入特征根的代数重数、几何重数和广义特征向量的概念：



特征根的代数重数、几何重数和广义特征向量

设 λ_i 为 n 维矩阵 \mathbf{A} 的特征值，则定义：

- λ_i 的重根数为代数重数，记为 σ_i ，满足 $\sum_i \sigma_i = n$ ；
- 齐次线性方程组 $(\lambda_i\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{0}$ 的维数^a为几何重数，记为 α_i ，根据定义有

$$\alpha_i = n - \text{rank}(\lambda_i\mathbf{I} - \mathbf{A})$$

- 可以证明任意特征值的代数重数 σ_i 和几何重数 α_i 恒有 $\sigma_i \geq \alpha_i$ 。

对于矩阵 \mathbf{A} ，若存在一个不为零的向量 \mathbf{p}_i 和一个标量 λ_i 使得下式成立，则称 \mathbf{p}_i 为 \mathbf{A} 的 k 级广义特征向量，其中 k 为正整数

$$\begin{cases} (\mathbf{A} - \lambda_i\mathbf{I})^k \mathbf{p}_i = \mathbf{0} \\ (\mathbf{A} - \lambda_i\mathbf{I})^{k-1} \mathbf{p}_i \neq \mathbf{0} \end{cases}$$

特别地，当特征值 λ_i 对应的代数重数 $\sigma_i > 1$ ，而几何重数 $\alpha_i = 1$ 时，其另外的 $\sigma_i - 1$ 个线性无关的广义特征向量 \mathbf{p}_{ij} 可由如下公式求得，其中 \mathbf{p}_{i1} 为 λ_i 对应的特征向量。

$$\mathbf{A}\mathbf{p}_{i,j+1} - \lambda_i\mathbf{p}_{i,j+1} = \mathbf{p}_{ij} \quad j = 1, \dots, \sigma_i - 1$$

^a齐次线性方程组 $\mathbf{Ax} = \mathbf{0}$ 的维数即为其解空间的维数，系数矩阵 \mathbf{A} 的秩越大（独立约束越多），则其解空间的维数越小，两者相加等于独立未知量的个数。当 $\text{rank}(\mathbf{A}) = n$ 时方程组只有一个零解，此时解空间的维数是 0。

对于给定的 n 维定常系统 $\dot{x} = Ax + Bu$, 设系统的特征值为 $\lambda_1, \dots, \lambda_l$, 每一特征值的重数分别为 σ_i, α_i , 则存在由各特征值特征向量和广义特征向量组成的变换阵 Q , 可将系统的状态方程化为如下的约当规范型

$$\dot{x} = Ax + Bu \xrightarrow{z=Q^{-1}x} \dot{z} = Q^{-1}AQz + Q^{-1}Bu = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_l \end{bmatrix} z + Q^{-1}Bu$$

变换后的系统矩阵矩阵 $Q^{-1}AQ$ 为 l 个约当块组成的对角块矩阵, l 个约当块对应 l 组不同的特征值, 矩阵中的任意约当块 J_i 的维数为 λ_i 的代数重数 σ_i . J_i 又是由 α_i 个约当子块 J_{ik} 组成的对角块矩阵, J_{ik} 的对角元为 λ_i 、次对角元为 1

$$J_i = \begin{bmatrix} J_{i1} & & \\ & \ddots & \\ & & J_{i\alpha_i} \end{bmatrix}_{\sigma_i \times \sigma_i} \quad J_{ik} = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}$$

当系统矩阵特征值存在重根时, 约当规范型是系统各状态变量所能达到的最简耦合形式, 同一特征根、同一约当子块所对应的状态变量之间才存在耦合关系。

22.5.3 从状态空间表达式到传递函数矩阵

1. 传递函数是一个有理多项式分式函数, 只能描述单输入单输出系统的输入输出关系, 而多输入多输出系统的输入输出关系则以传递函数矩阵表示。对于多输入多输出定常系统, 记输入向量、输出向量分别为 $u = [u_1 \ \dots \ u_r]^T$, $y = [y_1 \ \dots \ y_m]^T$, 且假设初始状态为零, $\hat{u}_i(s), \hat{y}_i(s)$ 分别表示 u_i, y_i 的拉氏变换, $w_{ij}(s)$ 表示第 j 个输入端到第 i 个输出端的传递函数, 系统的输入输出关系可描述为下式, 其中 $W(s)$ 即为传递函数矩阵

$$\begin{cases} \hat{y}_1(s) = w_{11}(s)\hat{u}_1(s) + \dots + w_{1r}(s)\hat{u}_r(s) \\ \dots \\ \hat{y}_m(s) = w_{m1}(s)\hat{u}_1(s) + \dots + w_{mr}(s)\hat{u}_r(s) \end{cases} \implies \hat{y}(s) = \begin{bmatrix} \hat{y}_1(s) \\ \vdots \\ \hat{y}_m(s) \end{bmatrix} = \begin{bmatrix} w_{11}(s) & \dots & w_{1r}(s) \\ \vdots & \ddots & \vdots \\ w_{m1}(s) & \dots & w_{mr}(s) \end{bmatrix} \begin{bmatrix} \hat{u}_1(s) \\ \vdots \\ \hat{u}_r(s) \end{bmatrix} = W(s)\hat{u}(s)$$

2. 显然对于任意系统状态空间表达式, 只需消去状态变量 x 即可得到输入向量 u 与输出向量 y 之间的直接联系, 从而得到传递函数矩阵 $W(s)$;
3. 注意到状态空间表达式中存在状态变量的微分项 \dot{x} , 为了消去状态变量可对其取拉氏变换

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \implies \begin{cases} sX(s) = AX(s) + BU(s) \\ Y(s) = CX(s) + DU(s) \end{cases} \xrightarrow{X(s)=(sI-A)^{-1}BU(s)} Y(s) = [C(sI - A)^{-1}B + D]U(s) = W(s)U(s)$$

4. 显然可以得到 $W(s) = C(sI - A)^{-1}B + D$, 并且当 $D \neq 0$ 时 $W(s)$ 为真有理分式矩阵, 当 $D = 0$ 时 $W(s)$ 为严格真有理分式矩阵;
5. 一个系统的状态空间表达式是非唯一的, 但传递函数矩阵是唯一的。

22.5.4 组合系统的状态空间表达式与传递函数矩阵

由两个或以上的子系统按照一定方式联接构成的系统成为组合系统。组合系统的基本组合方式可分为串联、并联和反馈三种类型。以下分别介绍三类基本组合系统的状态空间表达式和传递函数。考虑两个输入输出分别为 u_1, u_2, y_1, y_2 的子系统, 假设两子系统以某种形式联接构成组合系统, 组合系统的输入输出分别为 u, y , 则

- 子系统并联: 此时有 $\mathbf{u} = \mathbf{u}_1 = \mathbf{u}_2, \mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2$, 从而导出组合系统的状态空间表达式

$$\begin{cases} \dot{x}_1 = A_1 x_1 + B_1 u \\ \dot{x}_2 = A_2 x_2 + B_2 u \\ y = y_1 + y_2 = C_1 x_1 + C_2 x_2 + (D_1 + D_2) u \end{cases} \Rightarrow \begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_1 & D_2 \end{bmatrix} u \end{cases}$$

同样地可以得到系统的传递函数矩阵

$$\mathbf{y}(s) = \mathbf{y}_1(s) + \mathbf{y}_2(s) = \mathbf{w}_1(s)\mathbf{u}(s) + \mathbf{w}_2(s)\mathbf{u}(s) = [\mathbf{w}_1(s) + \mathbf{w}_2(s)]\mathbf{u}(s) \Rightarrow \mathbf{w}(s) = \mathbf{w}_1(s) + \mathbf{w}_2(s)$$

- 子系统串联: 此时有 $\mathbf{u} = \mathbf{u}_1, \mathbf{y}_1 = \mathbf{u}_2, \mathbf{y} = \mathbf{y}_2$, 从而导出组合系统的状态空间表达式和传递函数矩阵

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & \mathbf{0} \\ B_2 C_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 D_1 \end{bmatrix} u \\ y = \begin{bmatrix} D_2 C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_2 D_1 \end{bmatrix} u \end{cases} \quad \mathbf{w}(s) = \mathbf{w}_1(s) \cdot \mathbf{w}_2(s)$$

- 子系统反馈连接: 此时有 $\mathbf{u} = \mathbf{u}_1 + \mathbf{y}_2, \mathbf{y} = \mathbf{y}_1 = \mathbf{u}_2$, 从而导出组合系统的状态空间表达式和传递函数矩阵

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & -B_1 C_2 \\ B_2 C_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ \mathbf{0} \end{bmatrix} u \\ y = \begin{bmatrix} C_1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{cases} \quad \mathbf{w}(s) = [I + \mathbf{w}_1(s)\mathbf{w}_2(s)]^{-1} \mathbf{w}_1(s)$$

22.6 控制系统状态空间表达式的解

- 控制系统状态空间表达式的解, 就是基于系统的状态空间表达式, 求解其在给定初始条件 $\mathbf{x}(0) = \mathbf{x}_0$ 和控制输入 $\mathbf{u}(t)$ 共同作用下状态向量和输出向量 $\mathbf{x}(t), \mathbf{y}(t)$ 的运动规律;
- 线性系统满足叠加性原理, 因而系统的全响应 $\mathbf{x}(t)$ 可以分解为由初始状态 $\mathbf{x}(0) = \mathbf{x}_0$ 和控制输入 $\mathbf{u}(t)$ 分别单独作用产生的运动状态 $\mathbf{x}_{0u}(t), \mathbf{x}_{0x}(t)$ 的叠加, 分别称为零输入响应和零状态响应。

22.6.1 线性定常系统的零输入响应

- 对于任意 t , 零输入响应 $\mathbf{x}_{0u}(t)$ 就是系统由初始状态 \mathbf{x}_0 经线性变换矩阵得到的新状态, 对应于一系列 t , $\mathbf{x}_{0u}(t)$ 即表示系统自由运动的轨迹。求解 $\mathbf{x}_{0u}(t)$, 即求解以下线性定常齐次状态方程的解

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad t \geq t_0 \implies \mathbf{x}_{0u}(t) = \left(\sum_{k=0}^{\infty} \frac{1}{k!} \mathbf{A}^k (t-t_0)^k \right) \mathbf{x}_0 = e^{A(t-t_0)} \mathbf{x}_0 = e^{At} \mathbf{x}_0, \quad t_0 = 0$$

上式中 e^{At} 称为系统矩阵 \mathbf{A} 的矩阵指数函数, 零输入响应轨迹的形态由矩阵指数函数唯一确定, 系统渐进稳定的充要条件是 $\lim_{t \rightarrow \infty} \mathbf{A} = \mathbf{0}$ 。矩阵指数函数具有如下性质:

- 分解性: 令 t, τ 为两个时间变量, 则有 $e^{A(t+\tau)} = e^{At}e^{A\tau} = e^{A\tau}e^{At}$;
- 可逆性: 基于分解性可以得到矩阵指数函数的逆 $(e^{At})^{-1} = e^{-At}$;
- 倍时性: 同样地基于分解性可以得到矩阵指数函数的积 $(e^{At})^m = e^{A(mt)}, m = 0, 1, 2, \dots$;
- 对矩阵 \mathbf{A}, \mathbf{B} , 若可交换 $\mathbf{AB} = \mathbf{BA}$, 则有 $e^{At}e^{Bt} = e^{Bt}e^{At}$;
- 微分性和交换性: 矩阵指数函数对 t 求导 $\frac{d}{dt} e^{At} = Ae^{At} = e^{At}A$ 。

- 可以看到, 求解线性定常系统的零输入响应的关键在于计算矩阵指数函数 e^{At} , 以下介绍几种常用的方法:

- 直接求解法

注意到 e^{At} 定义为一个无穷级数和, 仅考虑级数的前几项可得到 e^{At} 的近似值;

- 将 \mathbf{A} 化为对角规范型或约当规范型

- 当 A 的特征根不存在重根时, 则 A 可化为对角规范型, 式中 T 为 A 特征向量组成的矩阵

$$T^{-1}AT = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \implies A = T \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} T^{-1}$$

代入 e^{At} 的定义

$$\begin{aligned} e^{At} &= \sum_{k=0}^{\infty} \frac{1}{k!} A^k t^k = \sum_{k=0}^{\infty} \frac{1}{k!} \left(T \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} T^{-1} \right)^k t^k \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} T \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}^k T^{-1} t^k \\ &= T \begin{bmatrix} \sum_{k=0}^{\infty} \frac{1}{k!} \lambda_1^k t^k & & \\ & \ddots & \\ & & \sum_{k=0}^{\infty} \frac{1}{k!} \lambda_n^k t^k \end{bmatrix} T^{-1} = T \begin{bmatrix} e^{\lambda_1 t} & & \\ & \ddots & \\ & & e^{\lambda_n t} \end{bmatrix} T^{-1} \end{aligned}$$

- 当 A 的特征根存在重根时, 则 A 可化为约当规范型, 特别地当所有特征值的几何重数均为 1 时

$$e^{At} = Q \begin{bmatrix} e^{J_1 t} & & \\ & \ddots & \\ & & e^{J_n t} \end{bmatrix} Q^{-1} \quad e^{J_i t} = e^{\lambda_i t} \begin{bmatrix} 1 & t & \frac{1}{2!} t^2 & \cdots & \frac{1}{(\sigma_i-1)!} t^{\sigma_i-1} \\ 0 & 1 & t & \cdots & \frac{1}{(\sigma_i-2)!} t^{\sigma_i-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & t \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

• 拉氏变换法

对 e^{At} 定义式两端同求拉氏变换, 并注意到 $\mathcal{L}[t^k/k!] = \frac{1}{s^{k+1}}$, 有

$$\mathcal{L}[e^{At}] = \frac{1}{s} I + \frac{1}{s^2} A + \cdots = \sum_{k=0}^{\infty} \frac{1}{s^{k+1}} A^k = s^{-1} \sum_{k=0}^{\infty} (s^{-1} A)^k$$

代入等比级数求和公式, 有

$$\mathcal{L}[e^{At}] = s^{-1} \sum_{k=0}^{\infty} (s^{-1} A)^k = s^{-1} (I - s^{-1} A)^{-1} = (sI - A)^{-1} \implies e^{At} = \mathcal{L}^{-1} [(sI - A)^{-1}]$$

- 进一步地, 给出系统状态转移矩阵的定义。对于给定的线性定常系统 $\dot{x} = Ax + Bu$, 称满足如下矩阵微分方程的 $n \times n$ 阶矩阵 $\Phi(t - t_0)$ 为系统的状态转移矩阵

$$\dot{\Phi}(t - t_0) = A\Phi(t - t_0) \quad \Phi(t_0) = I$$

显然可知对于线性定常系统, 系统的状态转移矩阵即为系统矩阵 A 的矩阵指数函数 $\Phi(t - t_0) = e^{A(t-t_0)}$, 从而可将系统的零输入响应改写为

$$x_{0u}(t) = \Phi(t - t_0)x(t_0), \quad t \geq t_0 \quad \text{or} \quad x_{0u}(t) = \Phi(t)x(0), \quad t \geq 0$$

可以看到, $\Phi(t - t_0)$ 的物理意义即是在零输入条件下将时刻 t_0 的状态 $x(t_0)$ 转移到 t 时刻的状态 $x(t)$ 的一个线性变换;

- 同样地 $\Phi(t)$ 也有与 e^{At} 相似的性质, 但不同于 e^{At} 本质上是一个无穷级数和 (具有数学定义), $\Phi(t)$ 是基于物理意义引出的一个概念, 因而仅从物理意义入手给出其各项性质的证明:

- 分解性: $\Phi(t + \tau) = \Phi(t)\Phi(\tau)$

证: 根据状态转移矩阵的物理意义, 易知

$$\begin{aligned} \mathbf{x}(t) &= \Phi(t - (-\tau))\mathbf{x}(-\tau) = \Phi(t + \tau)\mathbf{x}(-\tau) \\ \mathbf{x}(t) &= \Phi(t)\mathbf{x}(0) = \Phi(t)\Phi(0 - (-\tau))\mathbf{x}(-\tau) = \Phi(t)\Phi(\tau)\mathbf{x}(-\tau) \end{aligned}$$

显然命题得证。说明从 $-\tau$ 到 t 的转移可以看成先从 $-\tau$ 转移到 0 再从 0 转移到 t ;

- 可逆性: $[\Phi(t)]^{-1} = \Phi(-t)$

证: 根据定义 $\Phi(0) = \mathbf{I}$, 有

$$\Phi(t - t) = \Phi(t)\Phi(-t) = \mathbf{I} \implies [\Phi(t)]^{-1} = [\Phi(t)]^{-1}\mathbf{I} = [\Phi(t)]^{-1}\Phi(t)\Phi(-t) = \Phi(-t)$$

显然命题得证。说明从 0 状态到 t 状态转移的逆相当于从 t 状态到 0 状态的转移;

- 传递性: $\Phi(t_2 - t_1)\Phi(t_1 - t_0) = \Phi(t_2 - t_0)$, 证明过程同“分解性”;

- 倍时性: $[\Phi(t)]^k = \Phi(kt)$, 相当于“传递性”的扩展;

- 微分性和交换性: $\dot{\Phi}(t) = A\Phi(t) = \Phi(t)A$;

- 唯一性: 系统的状态转移矩阵唯一地由系统的系统矩阵决定。因此已知系统矩阵 A 可求解 $\Phi(t)$, 方法与 e^{At} 的计算方法完全一致, 反过来也可已知 $\Phi(t)$ 求解 A :

- 解法一: 由 $\dot{\Phi}(t) = A\Phi(t)$ 有 $A = \dot{\Phi}(t)[\Phi(t)]^{-1} = \dot{\Phi}(t)\Phi(-t)$;

- 解法二: 由 $\dot{\Phi}(t) = A\Phi(t)$ 有 $A = \dot{\Phi}(t)[\Phi(t)]^{-1} = \dot{\Phi}(t)\Phi(-t)$, 令 $t = 0$, 又因为 $\Phi(0) = \mathbf{I}$, 有 $A = \dot{\Phi}(0)$ 。

22.6.2 线性定常系统的零状态响应

1. 求解线性定常系统的零状态响应 $\mathbf{x}_{0x}(t)$, 即求解以下线性定常齐次状态方程的解

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}, \quad \mathbf{x}(t_0) = \mathbf{0} \quad t \geq t_0$$

以上为一个一阶线性常系数非齐次向量微分方程, 其求解方法与解的形式与一阶线性常系数非齐次实数微分方程类似, 即

$$\begin{aligned} \dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} &\iff e^{-At}(\dot{\mathbf{x}} - A\mathbf{x}) = e^{-At}\mathbf{Bu} \iff \int_{t_0}^t e^{-A\tau}(\dot{\mathbf{x}} - A\mathbf{x})d\tau = \int_{t_0}^t e^{-A\tau}\mathbf{Bu}d\tau \\ &\iff e^{-At}\mathbf{x}(t) - e^{-At_0}\mathbf{x}(t_0) = \int_{t_0}^t e^{-A\tau}\mathbf{Bu}(\tau)d\tau \\ &\iff e^{-At}\mathbf{x}(t) = \int_{t_0}^t e^{-A\tau}\mathbf{Bu}(\tau)d\tau \\ &\iff \mathbf{x}_{0x}(t) = \int_{t_0}^t e^{A(t-\tau)}\mathbf{Bu}(\tau)d\tau \end{aligned}$$

上式中的矩阵指数函数也可由状态转移矩阵表示, 即 $\mathbf{x}_{0x}(t) = \int_{t_0}^t \Phi(t - \tau)\mathbf{Bu}(\tau)d\tau$;

2. 将零输入响应与零状态响应的表达式相加, 即可得到线性定常系统非齐次状态方程的解, 即

$$\mathbf{x} = \mathbf{x}_{0u}(t) + \mathbf{x}_{0x}(t) = \Phi(t - t_0)\mathbf{x}(t_0) + \int_{t_0}^t \Phi(t - \tau)\mathbf{Bu}(\tau)d\tau \quad t > t_0 \quad \Phi(t) = e^{At}$$

3. 考虑 $t_0 = 0$ 的情况, 以下给出几个典型输入信号下的线性定常系统响应 (K 为常向量):

- 脉冲信号, 即 $\mathbf{u}(t) = K\delta(t)$, 有 $\mathbf{x}(t) = e^{At}\mathbf{x}_0 + e^{At}BK$;
- 阶跃信号, 即 $\mathbf{u}(t) = K\varepsilon(t)$, 有 $\mathbf{x}(t) = e^{At}\mathbf{x}_0 + A^{-1}(e^{At} - \mathbf{I})BK$;
- 斜坡信号³, 即 $\mathbf{u}(t) = KR(t)$, 有 $\mathbf{x}(t) = e^{At}\mathbf{x}_0 + [A^{-2}(e^{At} - \mathbf{I}) - A^{-1}t]BK$ 。

³斜坡信号在负半轴函数值为 0, 正半轴为正比例函数, 可定义为 $R(t) = \max\{0, t\}$

22.6.3 线性时变系统状态方程的解

1. 首先计算线性时变系统状态方程的零输入响应，即求解方程 $\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}$ 。当且仅当系统矩阵 $\mathbf{A}(t)$ 满足 $\mathbf{A}(t) \int_{t_0}^t \mathbf{A}(\tau)d\tau = (\int_{t_0}^t \mathbf{A}(\tau)d\tau)\mathbf{A}(t)$ 时，有

$$\mathbf{x}_{0u}(t) = e^{\int_{t_0}^t \mathbf{A}(\tau)d\tau} \mathbf{x}(t_0)$$

但在大多数情况下上述条件往往不成立，因此时变系统的零输入响应通常不能写成封闭形式；

2. 尽管无解析通解，线性时变系统的零输入响应仍可通过状态转移矩阵 $\Phi(t, t_0)$ 表示为

$$\mathbf{x}_{0u}(t) = \Phi(t, t_0)\mathbf{x}(t_0) \quad \dot{\Phi}(t, t_0) = \mathbf{A}(t)\Phi(t, t_0), \quad \Phi(t_0, t_0) = \mathbf{I}$$

不同于线性定常系统的状态转移矩阵 $\Phi(t - t_0)$ ，线性时变系统的状态转移矩阵 $\Phi(t, t_0)$ 表示为关于 t, t_0 的二元函数的形式。线性时变系统的状态转移矩阵 $\Phi(t, t_0)$ 与定常系统的状态转移矩阵 $\Phi(t - t_0)$ 具有相似的性质：

- 传递性： $\Phi(t_2, t_1)\Phi(t_1, t_0) = \Phi(t_2, t_0)$ ；
- 可逆性： $\Phi^{-1}(t, t_0) = \Phi(t_0, t)$ ；
- 唯一性；
- 微分性： $\dot{\Phi}(t, t_0) = \mathbf{A}(t)\Phi(t, t_0)$ ，注意 $\Phi(t, t_0)$ 一般不满足微分交换性，即一般地有 $\dot{\Phi}(t, t_0) \neq \Phi(t, t_0)\mathbf{A}(t)$ 。

3. 进一步地求解线性定常系统的零状态响应 $\mathbf{x}_{0x}(t)$ ，即求解以下线性定常齐次状态方程的解

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} + \mathbf{B}(t)\mathbf{u}, \quad \mathbf{x}(t_0) = \mathbf{0} \quad t \geq t_0$$

同样地一般也不存在解析解。参考零输入响应的解 $\mathbf{x}_{0u}(t)$ 的形式，不妨猜想

$$\mathbf{x}_{0x}(t) = \Phi(t, t_0)[\mathbf{x}(t_0) + \mathbf{x}_u(t)] = \Phi(t, t_0)\mathbf{x}_u(t)$$

表示为控制激励状态的转移。显然只需求解 $\mathbf{x}_u(t)$ 。将上式代入系统状态方程

$$\begin{aligned} \dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} + \mathbf{B}(t)\mathbf{u} &\iff \dot{\Phi}(t, t_0)\mathbf{x}_u(t) + \Phi(t, t_0)\dot{\mathbf{x}}_u(t) = \mathbf{A}(t)\Phi(t, t_0)\mathbf{x}_u(t) + \mathbf{B}(t)\mathbf{u}(t) \\ &\iff \mathbf{A}(t)\Phi(t, t_0)\mathbf{x}_u(t) + \Phi(t, t_0)\dot{\mathbf{x}}_u(t) = \mathbf{A}(t)\Phi(t, t_0)\mathbf{x}_u(t) + \mathbf{B}(t)\mathbf{u}(t) \\ &\iff \dot{\mathbf{x}}_u(t) = \Phi^{-1}(t, t_0)\mathbf{B}(t)\mathbf{u}(t) \\ &\iff \mathbf{x}_u(t) = \int_{t_0}^t \Phi(t_0, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau + \mathbf{x}_u(t_0) \end{aligned}$$

于是有

$$\mathbf{x}_{0x}(t) = \Phi(t, t_0)\mathbf{x}_u(t) = \Phi(t, t_0) \left[\int_{t_0}^t \Phi(t_0, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau + \mathbf{x}_u(t_0) \right]$$

因为 $\mathbf{x}_{0x}(t_0) = \mathbf{0}$ ，代入 $t = t_0$ 有 $\mathbf{x}_u(t_0) = \mathbf{0}$ ，则

$$\mathbf{x}_{0x}(t) = \Phi(t, t_0) \int_{t_0}^t \Phi(t_0, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau = \int_{t_0}^t \Phi(t, t_0)\Phi(t_0, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau = \int_{t_0}^t \Phi(t, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau$$

4. 将零输入响应与零状态响应叠加，可以得到线性时变系统非齐次状态方程的解 \mathbf{x}

$$\mathbf{x} = \mathbf{x}_{0u}(t) + \mathbf{x}_{0x}(t) = \Phi(t, t_0)\mathbf{x}(t_0) + \int_{t_0}^t \Phi(t, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau \quad t > t_0$$

对比其与线性定常系统非齐次状态方程的解可发现两者在形式上具有一致性。

22.7 线性控制系统的能控性和能观性

能控性和能观性是现代控制理论中的两个重要概念，是反馈控制与最优控制的基础。所谓系统的能控、能观是指系统的状态能控、能观：

- **能控：**若存在一个无约束的容许控制 $u(t)$ ，能在有限时间区间内 $t \in [t_0, t_1]$ ，使系统由某一非零（非平衡）的初始状态 $x(t_0) = \mathbf{0}$ ，转移到指定的终端状态（平衡状态） $x(t_1)$ ，则称系统的状态 $x(t_0)$ 是能控的（若为线性时变系统，则称系统在 t_0 时刻的状态 $x(t_0)$ 是能控的）；
- **完全能控：**若系统在状态空间中的所有非零状态 $x(t_0)$ 均能控，则称系统是完全能控的（若为线性时变系统，则称系统在 t_0 时刻完全能控）；
- **不完全能控：**若系统在状态空间中至少存在一个非零状态 $x(t_0)$ 是不能控的，则称系统是不完全能控的（若为线性时变系统，则称系统在 t_0 时刻不完全能控）；
- **能观：**对于线性定常系统，若存在有限的观测区间 $[t_0, t_1]$ ，使根据 $[t_0, t_1]$ 期间的输出 $y(t)$ ，能唯一地确定系统的状态 $x(t_0)$ ，则称状态 $x(t_0)$ 是能观的（若为线性时变系统，则称系统在 t_0 时刻的状态 $x(t_0)$ 是能观的）；
- **不能观：**对于线性定常系统，若取 t_0 时刻的一个非零状态 $x(t_0)$ ，存在有限的观测区间 $[t_0, t_1]$ ，使对所有的 $t \in [t_0, t_1]$ 有 $y(t) = 0$ ，则称状态 $x(t_0)$ 是不能观的（若为线性时变系统，则称系统在 t_0 时刻的状态 $x(t_0)$ 是不能观的）；
- **完全能观：**若系统的每一个非零状态 $x(t_0)$ 都是能观测的，则称系统是完全能观的（若为线性时变系统，则称系统在 t_0 时刻完全能观）。

22.7.1 线性定常系统的能控性及判据

结合线性定常系统状态方程的解，可以得到线性定常系统为完全能控系统的数学表述：对于任意给定的非零初始状态 $x(0) \neq \mathbf{0}$ ，寻找一个控制变量 $u(t)$ ，使系统在有限时间 t_1 内转移到平衡时刻，即

$$x(t_1) = e^{At_1}x(0) + \int_{t_0}^{t_1} e^{A(t_1-t)}Bu(t)dt = \mathbf{0}$$

进一步地，介绍几种主要的判断线性定常系统是否完全能控的判据：

1. 格拉姆 (Gram) 矩阵判据是最经典的线性定常系统完全能控的判据：对于任意线性定常系统，令 $t_0 = 0$ ，在 $t_1 > 0$ 时刻定义如下 Gram 矩阵 $W_c[0, t_1]$

$$W_c[0, t_1] = \int_0^{t_1} (e^{-At} B) (e^{-At} B)^T dt$$

则系统为完全能控的充要条件为矩阵 $W_c[0, t_1]$ 非奇异。格拉姆矩阵判据直接来源于系统完全能控的定义式，但格拉姆矩阵计算较为复杂，不适合工程应用；

2. 秩判据是工程中最常用的简单实用判据：系统为完全能控的充要条件是矩阵 $M = [B \ AB \ \cdots \ A^{n-1}B]$ 的秩 $\text{rank } M = n$ ，其中 n 为系统矩阵 A 的秩。秩判据可由格拉姆矩阵判据推导而来，且计算量显著减小。秩判据简单实用，但只能给出整个系统是否完全能控的信息，而无法得到不完全能控系统中不可控的特征根的信息；
3. PBH 秩判据：线性定常系统完全能控的充要条件是对系统矩阵的所有特征根 $\lambda_i (i = 1, \dots, n)$ （允许重根），下式均成立

$$\text{rank}[\lambda_i I - A \quad B] = n \quad i = 1, \dots, n$$

若 $\exists \lambda_j$ 使得上式不成立，则系统为不完全能控，且特征根 λ_k 是不可控的（即无法通过状态反馈控制移动 λ_j 所对应的传递函数的极点）。PBH 秩判据可由秩判据推导而来，尽管计算量稍大，但能具体地得到系统在每一特征根是否可控制的信息；

4. 规范型判据：若系统的状态空间表达式已经变换为对角规范型或约当规范型，则可简单地得到系统与每一特征根相关的能控性信息，即为规范型判据：

- 对角规范型判据（可由秩判据证明）：当系统状态方程经线性变换导出为对角规范型：

$$\dot{\bar{x}} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \bar{x} + \begin{bmatrix} \bar{b}_1 \\ \vdots \\ \bar{b}_n \end{bmatrix} u$$

则线性定常系统完全能控的充要条件为 $\bar{b}_i \neq 0$, 即 \bar{B} 中不含全为零的行；

- 约当规范型判据（可由 PBH 秩判据证明）：当系统状态方程经线性变换导出为约当规范型：

$$\dot{\bar{x}} = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_l \end{bmatrix} \bar{x} + \begin{bmatrix} \bar{B}_1 \\ \vdots \\ \bar{B}_l \end{bmatrix} u$$

设特征值 λ_i 的代数重数为 σ_i 、几何重数为 α_i , λ_i 对应的第 k ($k = 1, \dots, \alpha_i$) 个约当子块的维数记为 r_{ik} , 则线性定常系统完全能控的充要条件为：

- 若 $\sigma_i = 1$, 则 λ_i 对应的输入矩阵 \bar{B}_i (此时 \bar{B}_i 只有一行) 不全为 0;
- 若 $\sigma_i > 1, \alpha_i = 1$, 则 λ_i 对应的输入矩阵 \bar{B}_i 的最后一行不全为 0;
- 若 $\alpha_i > 1$, 则 λ_i 对应的每一个约当子块最后一行对应的输入矩阵行组成的矩阵为行线性无关。

22.7.2 线性定常系统的能观性及判据

因为能观所表示的是输出反应状态变量的能力, 与控制作用无直接关系, 因此在分析能观性时可令 $u = 0$, 且除了考虑状态方程外还需考虑输出方程, 从而得到系统的状态空间表达式

$$\dot{x} = Ax \quad y = Cx, \quad x(0) = x_0 \quad t \geq 0$$

此时状态方程为齐次方程, 其响应为零输入响应 $x(t) = e^{At}x_0$ 。线性定常系统能观性的判据与能控性判据完全对应：

1. 格拉姆 (Gram) 矩阵判据：对于任意线性定常系统, 定义如下 Gram 矩阵 $W_o[0, t_1]$

$$W_o[0, t_1] = \int_0^{t_1} (Ce^{At})^T (Ce^{At}) dt$$

则系统为完全能观的充要条件为矩阵 $W_o[0, t_1]$ 非奇异。格拉姆矩阵判据直接来源于系统完全能观的定义, 简单地给出证明：

- 证明充分性：若 $W_o[0, t_1]$ 非奇异, 则系统完全能观。此时只需构造一表达式使得可由 $y(t)$ 求解 $x(0)$

$$\begin{aligned} x(0) &= W_o^{-1}[0, t_1] \cdot W_o[0, t_1] \cdot x(0) = W_o^{-1}[0, t_1] \int_0^{t_1} (Ce^{At})^T (Ce^{At}) dt \cdot x_0 \\ &= W_o^{-1}[0, t_1] \int_0^{t_1} (Ce^{At})^T (Ce^{At} x_0) dt \\ &= W_o^{-1}[0, t_1] \int_0^{t_1} (Ce^{At})^T (Cx) dt = W_o^{-1}[0, t_1] \int_0^{t_1} (Ce^{At})^T y dt \end{aligned}$$

显然充分性得证；

- 证明必要性：若系统完全能观, 则 $W_o[0, t_1]$ 非奇异。采用反证法, 设 $W_o[0, t_1]$ 非奇异且系统能观。因为 $W_o[0, t_1]$ 非奇异, 则根据线性代数相关理论, 必存在非零向量 x_0 使得 $x_0^T W_o[0, t_1] x_0 = 0$, 即

$$\begin{aligned} x_0^T W_o[0, t_1] x_0 &= x_0^T \cdot \int_0^{t_1} (Ce^{At})^T (Ce^{At}) dt \cdot x_0 = \int_0^{t_1} (Ce^{At} x_0)^T (Ce^{At} x_0) dt \\ &= \int_0^{t_1} y^T y dt = \int_0^{t_1} \|y\|^2 dt = 0 \implies y(t) = 0, t \in [0, t_1] \end{aligned}$$

显然若 $W_o[0, t_1]$ 非奇异则系统不能观, 必要性得证。

2. 秩判据：系统为完全能观的充要条件是矩阵 $\mathbf{N} = [\mathbf{C} \quad \mathbf{CA} \quad \cdots \quad \mathbf{CA}^{n-1}]^T$ 的秩 $\text{rank } \mathbf{N} = n$ ，其中 n 为系统矩阵 \mathbf{A} 的秩。秩判据可由格拉姆矩阵判据推导而来，且计算量显著减小。秩判据简单实用，但只能给出整个系统是否完全能观的信息，而无法得到不完全能控系统中不能观的特征根的信息；
3. PBH 秩判据：线性定常系统完全能观的充要条件是对系统矩阵的所有特征根 $\lambda_i (i = 1, \dots, n)$ （允许重根），下式均成立

$$\text{rank}[\mathbf{C} \quad \lambda_i \mathbf{I} - \mathbf{A}]^T = n \quad i = 1, \dots, n$$

若 $\exists \lambda_j$ 使得上式不成立，则系统为不完全能观，且特征根 λ_k 是不能观的。PBH 秩判据可由秩判据推导而来，尽管计算量稍大，但能具体地得到系统在每一特征根是否能观的信息；

4. 规范型判据：若系统的状态空间表达式已经变换为对角规范型或约当规范型，则可简单地得到系统与每一特征根相关的能观性信息，即为规范型判据：

- 对角规范型判据（可由秩判据证明）：当系统状态空间表达式经线性变换导出为对角规范型：

$$\dot{\bar{x}} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \bar{x} \quad \bar{y} = [\bar{c}_1 \quad \cdots \quad \bar{c}_n] \bar{x}$$

则线性定常系统完全能观的充要条件为 $\bar{c}_i \neq 0$ ，即 \bar{C} 中不含全为零的列；

- 约当规范型判据（可由 PBH 秩判据证明）：当系统状态空间表达式经线性变换导出为约当规范型：

$$\dot{\bar{x}} = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_l \end{bmatrix} \bar{x} \quad \bar{y} = [\bar{C}_1 \quad \cdots \quad \bar{C}_l] \bar{x}$$

设特征值 λ_i 的代数重数为 σ_i 、几何重数为 α_i ， λ_i 对应的第 k ($k = 1, \dots, \alpha_i$) 个约当子块的维数记为 r_{ik} ，则线性定常系统完全能观的充要条件为：

- 若 $\sigma_i = 1$ ，则 λ_i 对应的输出矩阵 \bar{C}_i （此时 \bar{C}_i 只有一列）不全为 0；
- 若 $\sigma_i > 1, \alpha_i = 1$ ，则 λ_i 对应的输出矩阵 \bar{C}_i 的第一列不全为 0；
- 若 $\alpha_i > 1$ ，则 λ_i 对应的每一个约当子块第一列对应的输出矩阵列组成的矩阵为列线性无关。

22.7.3 线性时变系统的能控性、能观性及判据

线性时变系统的能控性、能观性定义与线性定常系统的相关定义基本一致，唯一的区别在线性时变系统强调其在具体的某一时刻 t_0 是能控或能观的，因为状态空间表达式的系数会随时间而变化。线性时变系统也存在相应的能控性、能观性判据：

1. 能控性格拉姆 (Gram) 矩阵判据：对于任意线性时变系统，定义如下 Gram 矩阵 $\mathbf{W}_c[t_0, t_1]$

$$\mathbf{W}_c[t_0, t_1] = \int_{t_0}^{t_1} [\Phi(t_0, t) \mathbf{B}(t)] [\Phi(t_0, t) \mathbf{B}(t)]^T dt$$

则系统在 t_0 时刻完全能控的充要条件为矩阵 $\mathbf{W}_c[t_0, t_1]$ 非奇异。线性时变系统的格拉姆矩阵与线性定常系统完全一致，仅是由状态转移矩阵替代矩阵指数函数；

2. 能控性秩判据：线性时变系统在 t_0 时刻完全能控的充分条件是矩阵 $\mathbf{M}(t_1)$ 的秩 $\text{rank } \mathbf{M}(t_1) = n$ ，其中 n 为系统矩阵 $\mathbf{A}(t)$ 的秩

$$\mathbf{M}(t_1) = [\mathbf{M}_0(t_1) \quad \cdots \quad \mathbf{M}_{n-1}(t_1)], \quad \mathbf{M}_0(t) = \mathbf{B}(t) \quad \mathbf{M}_{k+1}(t) = -\mathbf{A}(t)\mathbf{M}_k(t) + \frac{d\mathbf{M}_k(t)}{dt}$$

显然上式要求矩阵 $\mathbf{A}(t), \mathbf{B}(t)$ 是 $n-1$ 阶可微的，因此该判据仅为充分条件而非充要条件；

3. 能观性格拉姆 (Gram) 矩阵判据：对于任意线性时变系统，定义如下 Gram 矩阵 $\mathbf{W}_o[t_0, t_1]$

$$\mathbf{W}_o[t_0, t_1] = \int_{t_0}^{t_1} [\mathbf{C}(t) \Phi(t, t_0)]^T [\mathbf{C}(t) \Phi(t, t_0)] dt$$

则系统在 t_0 时刻完全能观的充要条件为矩阵 $\mathbf{W}_o[t_0, t_1]$ 非奇异；

4. 能观性秩判据：线性时变系统在 t_0 时刻完全能观的充分条件是矩阵 $N(t_1)$ 的秩 $\text{rank } N(t_1) = n$, 其中 n 为系统矩阵 $A(t)$ 的秩

$$N(t_1) = \begin{bmatrix} N_0(t_1) \\ \vdots \\ N_{n-1}(t_1) \end{bmatrix}, \quad N_0(t) = C(t) \quad N_{k+1}(t) = N_k(t)A(t) + \frac{dN_k(t)}{dt}$$

显然上式要求矩阵 $A(t), B(t)$ 是 $n - 1$ 阶可微的，因此该判据仅为充分条件而非充要条件。

22.7.4 能控与能观性的对偶关系

1. 首先给出对偶系统的定义。线性时变系统与线性定常系统关于对偶系统的定义存在差异

线性时变系统 称满足如下关系的两系统 Σ_1, Σ_2 互为对偶系统

$$\begin{aligned} \Sigma_1 : \dot{x}_1 &= A(t)x_1 + B(t)u_1 & \Sigma_2 : \dot{x}_2 &= -A^T(t)x_2 + C^T(t)u_2 \\ y_1 &= C(t)x_1 & y_2 &= B^T(t)x_2 \end{aligned}$$

其中 x_1, x_2 为 n 维状态向量； u_1, u_2 分别为 r, m 维控制向量； y_1, y_2 分别为 m, r 维输出向量；

线性定常系统 称满足如下关系的两系统 Σ_1, Σ_2 互为对偶系统

$$\begin{aligned} \Sigma_1 : \dot{x}_1 &= Ax_1 + Bu_1 & \Sigma_2 : \dot{x}_2 &= -A^T x_2 + C^T u_2 & \left(\text{or} \quad \Sigma_2 : \dot{x}_2 = A^T x_2 + C^T u_2 \right) \\ y_1 &= Cx_1 & y_2 &= B^T x_2 \end{aligned}$$

2. (定理) 互为对偶的两线性时变系统的状态转移矩阵互为转置逆，即 $\Phi_2(t, t_0) = \Phi_1^T(t_0, t)$ 。⁴

证：根据状态转移矩阵的定义，有

$$\dot{\Phi}_1(t, t_0) = A(t)\Phi_1(t, t_0) \quad \dot{\Phi}_2(t, t_0) = A(t)\Phi_2(t, t_0) \quad \Phi_1(t_0, t_0) = \Phi_2(t_0, t_0) = I$$

因为恒等式 $\Phi_1(t, t_0)\Phi_1^{-1}(t, t_0) = I$ 成立，对两边同时求导有

$$\begin{aligned} \frac{d}{dt} (\Phi_1(t, t_0)\Phi_1^{-1}(t, t_0)) &= \mathbf{0} \implies \dot{\Phi}_1(t, t_0)\Phi_1^{-1}(t, t_0) + \Phi_1(t, t_0)\dot{\Phi}_1^{-1}(t, t_0) = \mathbf{0} \\ &\implies A(t)\Phi_1(t, t_0)\Phi_1^{-1}(t, t_0) + \Phi_1(t, t_0)\dot{\Phi}_1^{-1}(t, t_0) = \mathbf{0} \\ &\implies A(t) + \Phi_1(t, t_0)\dot{\Phi}_1(t_0, t) = \mathbf{0} \\ &\implies \dot{\Phi}_1(t_0, t) = -\Phi_1^{-1}(t, t_0)A(t) = -\Phi_1(t_0, t)A(t) \\ &\implies \dot{\Phi}_1^T(t_0, t) = -A^T(t)\Phi_1^T(t_0, t) \end{aligned}$$

因为矩阵 $\Phi_1^T(t_0, t)$ 同样满足 $\Phi_1^T(t_0, t_0)$ 且 $-A^T(t)$ 为对偶系统 Σ_2 的系统矩阵，由状态转移矩阵的唯一性有 $\Phi_2(t, t_0) = \Phi_1^T(t_0, t)$ ；

3. (对偶原理) 互为对偶的两系统 Σ_1, Σ_2 ，则 Σ_1 的能控性等价于 Σ_2 的能观性， Σ_1 的能观性等价于 Σ_2 的能控性。

证明 Σ_1, Σ_2 为线性时变系统的情况：对于 Σ_1 ，其能控性 Gram 矩阵 $W_{\Sigma_1 c}[t_0, t_1]$ 定义为

$$W_{\Sigma_1 c}[t_0, t_1] = \int_{t_0}^{t_1} [\Phi_1(t_0, t)B(t)] [\Phi_1(t_0, t)B(t)]^T dt$$

令 $\Phi_2(t, t_0) = \Phi_1^T(t_0, t)$ ，则

$$W_{\Sigma_1 c}[t_0, t_1] = \int_{t_0}^{t_1} [\Phi_2^T(t, t_0)(B^T(t))^T] [\Phi_2^T(t, t_0)B(t)]^T dt = \int_{t_0}^{t_1} [B^T(t)\Phi_2(t, t_0)]^T [B^T(t)\Phi_2(t, t_0)] dt = W_{\Sigma_2 o}[t_0, t_1]$$

显然 Σ_1 的能控性等价于 Σ_2 的能观性，同理也可证明 Σ_1 的能观性等价于 Σ_2 的能控性。对于线性定常系统，两种形式的对偶系统均满足对偶原理，其中第一种形式的证明与线性时变系统的证明完全类似，而第二种形式可通过秩判据证明。

⁴对于互为对偶的两线性定常系统，若对偶系统 Σ_2 的系统矩阵为 $-A^T$ ，则上述定理依然满足，即 $\Phi_2(t - t_0) = \Phi_1^T(t_0 - t)$ ；若对偶系统 Σ_2 的系统矩阵为 A^T ，则不满足，两状态转移矩阵的关系为 $\Phi_2(t - t_0) = \Phi_1^T(t - t_0)$ 。

22.7.5 状态空间表达式的能控性标准型与能观性标准型

1.

22.8 稳定性与李雅普诺夫 (Lyapunov) 方法

22.8.1 系统平衡状态与李雅普诺夫稳定性定义

1. 稳定性是所有系统的最重要特性。经典控制理论仅关注线性时不变系统，提出了劳斯判据（见第 22.3 节）、奈奎斯特判据等稳定性判据。这些判据普遍认为系统的稳定性只与系统的结构和参数有关，而与初始条件和扰动无关。但对于更一般的非线性系统而言，显然其稳定性还会受到初始条件和扰动大小的影响（以卫星绕地公转为例，当扰动过大（超过第一宇宙速度），则卫星将摆脱地球引力束缚而无法重新回到绕地轨道）。历史上曾有大批学者尝试将经典控制理论应用于非线性系统稳定性分析，但均以失败告终；
2. 1892 年，俄国数学、力学家李雅普诺夫在其博士论文中给出了系统稳定性的严格的一般性的数学定义和一般性方法，从而奠定了稳定性理论的基础。李雅普诺夫方法由第一法和第二法组成，又分别称为间接法 (*indirect method*) 和直接法 (*direct method*)。李雅普诺夫第一法与经典控制理论的稳定性方法一脉相承，而第二法则是适用于线性系统、非线性系统、时变系统稳定性分析的一般性方法，但第二法需要构建李雅普诺夫函数，而李雅普诺夫函数不存在针对任意系统的一般性设计方法；
3. 李雅普诺夫首次提出系统的稳定性是相对于其平衡状态而言的。在经典控制理论中并不强调平衡状态，因为经典控制理论只适用于线性系统，而线性系统往往只存在一个平衡状态。但对非线性系统而言则可能存在多个平衡状态，因此需要区分系统对某个平衡点的稳定性；
4. 进一步介绍系统平衡状态的定义。对于一个不受外力的系统（自治系统） $\dot{x} = f(x, t)$, $x(t_0) = x_0$, $t \geq t_0$, 如果存在状态 x_e 使得

$$\dot{x}_e = f(x_e, t) = 0, \quad \forall t \geq t_0 \quad (\text{系统平衡状态定义})$$

则称 x_e 为系统的一个平衡状态。其物理意义是出于平衡状态下的系统其状态不随时间而变化。对于线性定常系统 $\dot{x} = Ax$, 按定义求解其稳定性等价于解方程组 $Ax = 0$ 。显然若 A 为非奇异矩阵则系统只存在 $x_e = 0$ 一个平衡状态；若 A 奇异则系统存在无穷多个连续的平衡状态。实际工程中 A 一般为非奇异阵。若系统的某个平衡状态的任意小邻域内不存在其它平衡状态，则称其为孤立的平衡状态。非线性系统往往存在多个孤立的平衡状态；

5. 进一步介绍基于孤立平衡状态 x_e 的李雅普诺夫稳定性定义。仍关注自治系统 $\dot{x} = f(x, t)$, $x(t_0) = x_0$, $t \geq t_0$
- 李雅普诺夫稳定：对于 $\forall \varepsilon > 0$, $\exists \delta(\varepsilon, t) > 0$, 使得在 x_e 的 $\delta(\varepsilon, t)$ 邻域范围内的任意初始状态 x_0 导致的受扰运动状态轨迹与平衡状态的距离不超过 ε ，则称孤立平衡状态 x_e 为李雅普诺夫意义上的稳定状态。用数学语言写为下式

$$\|\Phi(t; x_0, t_0) - x_e\| < \varepsilon, \quad \forall t \geq t_0, \quad \forall \|x_0 - x_e\| \leq \delta(\varepsilon, t_0) \quad (\text{李雅普诺夫稳定})$$

式中 $\Phi(t; x_0, t_0)$ 表示 t_0 时刻初始状态 x_0 造成的系统于 t 时刻的扰动状态。若 δ 不存在，则称孤立平衡状态 x_e 为李雅普诺夫意义上的不稳定状态。注意到李雅普诺夫稳定并不要求系统状态扰动后重新回到原平衡状态，与工程中认为的稳定状态不同；

- 一致稳定：若 δ 的取值与 t_0 无关则称孤立平衡状态 x_e 是一致稳定的；
- 渐进稳定：在李雅普诺夫稳定的基础上，若孤立平衡状态 x_e 邻域范围内的任意初始状态 x_0 导致的受扰运动状态会随时间收敛回 x_e ，则称孤立平衡状态 x_e 是渐进稳定的。用数学语言描述如下

$$\lim_{t \rightarrow \infty} \|\Phi(t; x_0, t_0) - x_e\| = 0, \quad \forall t \geq t_0, \quad \forall \|x_0 - x_e\| \leq \delta(\varepsilon, t_0) \quad (\text{李雅普诺夫渐进稳定})$$

显然李雅普诺夫意义上的渐进稳定即为工程上的稳定；

- **大范围渐进稳定:** 若孤立平衡状态 x_e 是李雅普诺夫意义上的稳定状态, 且对系统的任意非平衡初始状态均渐进稳定, 则称孤立平衡状态 x_e 是大范围渐进稳定的。用数学语言描述如下

$$\lim_{t \rightarrow \infty} \|\Phi(t; x_0, t_0) - x_e\| = 0, \quad \forall t \geq t_0, \quad \forall f(x_0, t_0) \neq 0 \quad (\text{李雅普诺夫大范围渐进稳定})$$

6. 由上述定义发现, 李雅普诺夫意义下的稳定性评价只考虑了系统状态 x , 而不考虑系统的输入-输出关系, 与经典控制理论中稳定性评估方法存在显著差异。这是因为现代控制理论中将系统的输入-输出关系分解为状态方程 $\dot{x} = f(x, u, t)$ 和输出方程 $y = g(x, u, t)$ 两部分, 由此也衍生出了稳定性评价的两个维度:

- **内部稳定 (状态稳定):** 从状态方程出发, 针对系统的内部状态。要求当扰动信号取消后系统的内部状态会在一定时间内恢复到原来的平衡状态, 则称系统状态稳定;
- **外部稳定 (输出稳定):** 从输出方程出发, 针对系统的响应特性。要求当扰动信号取消后系统的输出会在一定时间内恢复到原来的稳态输出, 则称系统输出稳定。经典控制理论只研究系统的外部稳定, 也称为 BIBO 稳定。

在一般情况下输出方程会假设输入 u 只能通过影响状态 x 影响输出 y 。此时由状态空间表达式的组成可知, 若系统内部稳定则必然外部稳定, 但外部稳定则不能反推内部稳定。根据能控性和能观性的定义, 只有在系统能控、能观的前提下, 系统的内部稳定性和外部稳定性等价。

22.8.2 李雅普诺夫第一法 (间接法)

1. 李雅普诺夫第一法可视为经典的线性系统外部稳定性分析方法于内部稳定性的延伸, 应用于非线性系统时需要对其线性化, 且只可分析部分非线性系统的稳定性;
2. 首先关注线性系统的内部稳定性。对于定常系统 $\dot{x} = Ax + Bu$, $y = Cx$, 其平衡状态 $x_e = 0$, 则系统于平衡状态 x_e 处渐进稳定的充要条件是系统特征矩阵 A 的所有特征值均有负实部。可按定义证明。因为内部稳定性只关注系统扰动后的状态轨迹, 对应于系统的零输入响应, 按第 22.6 节介绍的线性定常系统零输入响应的解有

$$x_{0u}(t) = e^{At}x_0, \quad e^{At} = Q \begin{bmatrix} e^{J_1 t} & & \\ & \ddots & \\ & & e^{J_n t} \end{bmatrix} Q^{-1}, \quad e^{J_i t} = \sum_{k=0}^{\infty} \frac{1}{k!} J_i^k t^k$$

式中矩阵 Q 为 A 的特征向量和广义特征向量构成的矩阵, 而对角块矩阵中的矩阵块 $\{e^{J_i t} | i = 1, \dots, n\}$ 与 A 的特征值 $\{\lambda_i | i = 1, \dots, n\}$ 一一对应。若 λ_i 为单根则 $J_i = \lambda_i$ 为标量; 若 λ_i 为重根则 J_i 为约当矩阵块, 对角线元素为 λ_i , 副对角线元素为 0 和 1, 具体形式由 λ_i 的代数重数和几何重数决定 (详见第 22.5 节中的状态向量线性变换方法)。显然零输入响应 $x_{0u}(t)$ 可表现为 A 的矩阵指数函数 e^{At} 各元素所代表的信号分量的线性组合。而根据第 22.6 节的内容, 在 A 既包含单根特征值和重根特征值的一般情况下, e^{At} 的各元素按形式可分为 $e^{\lambda_i t}, e^{\lambda_i t} t^n$ 两种形式。为使得 $x_{0u}(t)$ 随时间收敛至平衡状态 $x_e = 0$, 则应使得 e^{At} 中的所有信号分量均收敛至平衡状态 $x_e = 0$, 则应使得 A 的所有特征值 $\{\lambda_i | i = 1, \dots, n\}$ 均有负实部, 问题得证。若 A 的特征值难以求解, 也可列其特征多项式 $|sI - A|$ 再由劳斯判据 (见第 22.3 节) 判别;

3. 再关注线性系统的外部稳定性。由经典控制理论易知, 线性定常系统输出稳定的充要条件是其传递函数 $W(s)$ 的极点均有负实部, 与内部稳定性充要条件非常相似。进一步分析外部稳定性充要条件与内部稳定性充要条件的关系。对状态空间表达式作拉氏变换求传递函数, 有

$$W(s) = C(sI - A)^{-1}B = \frac{C(sI - A)^*B}{|sI - A|}$$

式中 $(sI - A)^*$ 为矩阵 $sI - A$ 的伴随矩阵; 分母 $|sI - A|$ 为 A 的特征多项式 (或矩阵 $sI - A$ 的行列式):

- 若 $C(sI - A)^*B$ 与 $|sI - A|$ 不存在公因子, 则 $W(s)$ 的极点即为 A 的特征值, 此时系统的外部稳定性和内部稳定性等价, 对应于系统同时能控、能观的情况;
- 若 $C(sI - A)^*B$ 与 $|sI - A|$ 存在公因子, 则 $W(s)$ 的极点少于 A 的特征值, 此时系统可能外部稳定而内部不稳定, 对应于系统不能同时能控、能观的情况。

4. 另外关注非线性系统的稳定性。设非线性系统的状态方程为 $\dot{x} = f(x, t)$, x_e 为平衡状态, 则为应用李雅普诺夫第一法需对系统于平衡状态 x_e 处作一阶线性化。假设 $f(x, t)$ 对 x 有连续的一阶偏导, 则有

$$\dot{x} = f(x_e, t) + \frac{\partial f}{\partial x^\top} \Big|_{x=x_e} (x - x_e) + R(x) \implies \Delta \dot{x} = \frac{\partial f}{\partial x^\top} \Big|_{x=x_e} \Delta x + R(x), \quad \Delta x = x - x_e, \quad \frac{\partial f}{\partial x^\top} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

式中 $R(x)$ 为关于 x 的高阶小量。略去 $R(x)$ 即为线性化的状态方程。则李雅普诺夫第一法指出:

- 若线性化状态方程的系数矩阵的所有特征值均有负实部, 则原非线性系统在平衡状态 x_e 处是渐进稳定的, 且稳定性与 $R(x)$ 无关;
- 若线性化状态方程的系数矩阵的特征值存在正实部, 则原非线性系统在平衡状态 x_e 处是不稳定的;
- 若线性化状态方程的系数矩阵存在虚轴上的特征值, 则原非线性系统在平衡状态 x_e 处的稳定性取决于高阶小量 $R(x)$, 不能由线性化后系统的稳定性判断原系统的稳定性。

22.9 鲁棒控制与 H_∞ 无穷控制基础

1. 控制论发展的过程可大体分为三个阶段——基于输入-输出模型的经典控制理论(30's-40's)、基于状态空间法的现代控制理论(60's-70's)和以鲁棒控制、模糊控制和智能控制等为代表的后现代控制理论。经典控制理论和现代控制理论均是基于确定性数学模型研究控制系统。但由于工作状况变动、外部干扰以及建模误差的缘故, 实际工业过程的精确模型很难得到。在考虑不确定的情况下设计有效的控制系统的需要便催生出了后现代控制理论;
2. 鲁棒控制理论依托于鲁棒性(robustness)概念, 聚焦于一个基于确定性数学模型设计的控制器在不确定扰动下保持良好状态的能力。具体地, 鲁棒性描述为: 假定对象的数学模型属于一集合 \mathcal{P} 。考察反馈系统的某些特性, 如内部稳定性。给定一控制器 K , 如何集合 \mathcal{P} 中的每一个对象都能保持这种特性成立, 则称该控制器对此特性是鲁棒的。因此讨论鲁棒性必有一个控制器、一个对象集合和某些系统特性;
3. 鲁棒性是一个自然提出的概念, 在控制理论发展的不同时期不同方法都有对应的数学解释:
 - 微分方程建模时方程的解对初值和参数的连续依赖性即可表征鲁棒性;
 - 单输入-单输出系统频率特性的稳定裕度概念(第 22.4.2 节)也可表征鲁棒性;
 - 李雅普诺夫稳定性分析和系统灵敏度分析也体现了鲁棒性概念。
- 自然地, 鲁棒控制并不特指某种方法, 而是包含了多种理论流派;
4. 兴起于 20 世纪 80 年代的 H_∞ 控制理论是现阶段鲁棒控制各理论流派中最为成熟的一支。 H_∞ 优化控制问题始于 1981 年。加拿大学者 Zames 首次用明确的数学语言描述了单输入-单输出线性反馈系统的灵敏度函数的无穷范数极小化问题。这一工作处理了古典控制理论中的一些基本问题, 因此立即引起了人们的注意。尤其是当人们意识到这种方法在处理鲁棒性时比其它方法更直接以后, 该方法就很快被应用到更一般的问题中。

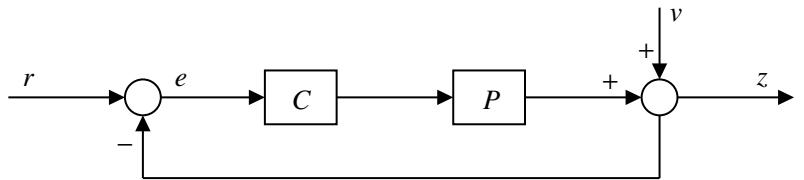
22.9.1 单输入单输出(SISO)系统的 H_∞ 无穷优化问题建模

1. H_∞ 优化控制最早针对单输入-单输出(SISO)线性反馈系统的鲁棒性设计而提出。此时系统的鲁棒性可以由经典控制理论从开环传递函数的频率特性中直观且可靠地估计;
2. 考虑一个带扰动信号的 SISO 反馈系统。显然扰动信号 v 会影响系统输出 z , 从而可能影响系统的输入-输出稳定性。为量化 v 对 z 的影响, 定义反馈系统的灵敏度(sensitivity)函数 S 为 v 到 z 的闭环传递函数, 则

$$\begin{aligned} E &= R - (CP \cdot E + V) \implies E = \frac{R - V}{1 + CP} \\ \implies S &= \frac{Z}{V} = \frac{CP \cdot E + V}{V} = \frac{CP}{1 + CP} \left(\frac{R}{V} - 1 \right) + 1 = -\frac{CP}{1 + CP} + 1 = \frac{1}{1 + CP} \end{aligned} \quad (\text{SISO 系统灵敏度函数})$$

另外易证 S 同样为系统参考输入 r 到偏差 e 的闭环传递函数，则灵敏度函数既反映了系统对外部干扰的抑制能力，也反映了系统对参考信号的跟踪能力，因此可表征系统性能。理想情况下 S 应为 0；

图 22.1 SISO 反馈回路。其中 r, e, v, z 分别表示输入参考信号、偏差信号、干扰信号和输出信号； P 为广义被控对象的传递函数； C 为补偿器的传递函数。



3. 为保证系统的鲁棒性，自然地即要求最小化扰动信号 v 对系统输出 z 的影响。**Zames** 最初考虑的问题即是寻找一个补偿器 C ，使得系统闭环稳定且极小化灵敏度函数 S 的无穷范数 $\|S\|_\infty$ （即 S 的峰值）。直观地将 $\|S\|_\infty$ 定义为相应相频特性曲线 $|S(i\omega)|$ 的上确界

$$\|S\|_\infty = \sup_{\omega \in \mathbb{R}} |S(i\omega)| \quad (\text{灵敏度函数的无穷范数})$$

上述定义的物理意义是如果 $\|S\|_\infty$ 尽可能小，意味着所有频率上 S 的幅值都较小，从而保证所有频率的干扰均被一致衰减。 H_∞ 优化问题即得名于对 $\|S\|_\infty$ 的优化

$$\min_C \|W_S S\|_\infty = \min_C \sup_{\omega \in \mathbb{R}} \|W_S(i\omega) S(i\omega)\| \quad (H_\infty \text{ 优化问题建模})$$

上式中 W_S 为自定义的频率加权函数，一般设置低频处具有较大的增益。因为实际工程中绝大多数信号都集中于低频处，而实际的被控对象 P 和补偿器 C 的频率响应函数在 $\omega \rightarrow \infty$ 处均会衰减至 0，使得 S 在 $\omega \rightarrow \infty$ 处趋于 1。引入 W_S 的目的即是使得优化时聚焦 S 在低频处的大小；

4. 另外再将 S 视为从 v 到 z 的系统，则定义系统 S 的范数为

$$\|S\| = \sup_{\|v\|_2 < \infty} \frac{\|z\|_2}{\|v\|_2}, \quad \|v\|_2 = \sqrt{\int_{-\infty}^{\infty} v^2(t) dt} \quad (\text{系统范数})$$

式中 $\|\cdot\|_2$ 表示信号的能量的开根。 $\|S\|$ 在数学上定义为 2 范数的诱导范数。Rayleigh 定理指出信号在时域和频域具有相同的能量（第 21.4 节），则上式有

$$\frac{\|z\|_2}{\|v\|_2} = \sqrt{\frac{\int_{-\infty}^{\infty} \|Z(i\omega)\|^2 d\omega}{\int_{-\infty}^{\infty} \|V(i\omega)\|^2 d\omega}} = \sqrt{\int_{-\infty}^{\infty} \|S(i\omega)\|^2 \frac{\|V(i\omega)\|^2}{\int_{-\infty}^{\infty} \|V(i\omega)\|^2 d\omega} d\omega} \leq \sup_{\omega \in \mathbb{R}} \|S(i\omega)\| \implies \|S\| = \sup_{\omega \in \mathbb{R}} \|S(i\omega)\| = \|S\|_\infty$$

上式表明 H_∞ 优化问题也是对系统范数 $\|S\|$ 的极小化问题；

5. 进一步描述极小化 $\|S\|_\infty$ 与鲁棒性之间的关系。记反馈系统的开环传递函数 $L = CP$ ，并称未受扰动的标称开环传递函数为 L_0 ，扰动下的开环传递函数为 L 。根据控制系统鲁棒性的定义，要求扰动并不改变系统特性。故不妨要求标称开环系统和受摄动开环系统具有相同数目的右半平面极点，且系统保持闭环稳定。根据奈奎斯特判据（第 22.4.1 节）等价于要求 $L_0(i\omega)$ 与 $L(i\omega)$ 绕点 $-1 + i0$ 的圈数不变。一个充分条件是对于 $\forall \omega \in \mathbb{R}$ ，扰动前后 $L_0(i\omega)$ 与 $L(i\omega)$ 的距离应小于 $L_0(i\omega)$ 与点 $-1 + i0$ 的距离，即

$$\frac{\|L(i\omega) - L_0(i\omega)\|}{\|L_0(i\omega) + 1\|} < 1 \implies \frac{\|L(i\omega) - L_0(i\omega)\|}{\|L_0(i\omega)\|} \cdot \frac{\|L_0(i\omega)\|}{\|L_0(i\omega) + 1\|} < 1 \implies \frac{\|L(i\omega) - L_0(i\omega)\|}{\|L_0(i\omega)\|} \cdot \|T_0(i\omega)\| < 1, \quad \forall \omega \in \mathbb{R}$$

上式中 T 为反馈系统的补灵敏度（complementary sensitivity）函数，定义为系统的闭环传递函数

$$T = 1 - S = \frac{CP}{1 + CP} \quad (\text{SISO 系统补灵敏度函数})$$

再引入自定义的频率加权函数 W_T ，并将相频特性写为无穷范数的形式，有

$$\|W_T(i\omega) T(i\omega)\| < 1, \quad \forall \omega \in \mathbb{R} \implies \|W_T T\|_\infty < 1, \quad \frac{\|L(i\omega) - L_0(i\omega)\|}{\|L_0(i\omega)\|} \leq \|W_T(i\omega)\| \quad (\text{鲁棒稳定条件})$$

上式被称为 SISO 系统的鲁棒稳定条件。因为高频段系统标称模型 $L_0(i\omega)$ 与实际模型 $L(i\omega)$ 之间的差异一般较大，因此通常设计 W_T 在高频段具有较大的增益。上式表明系统鲁棒性（极小化 $\|T\|_\infty$ ）与系统性能（极小化 $\|S\|_\infty$ ）之间存在矛盾。

22.9.2 多输入多输出 (MIMO) 系统的 H 无穷优化问题建模

1.

第 23 章

其它知识

23.1 正交实验 (Orthogonal Test)

1. ¹假设某个实验需考虑 3 个自变量，每个自变量又有 4 种状态，则全面实验的次数为 $4^3 = 64$ 次，显然随着自变量个数及每一自变量变化状态的增加，将需要进行大量的实验，消耗大量的资源。正交实验是研究多因素多水平的一种实验方法，借助正交表，根据正交性从全面试验中挑选出部分有代表性的点进行试验，这些有代表性的点具备了“均匀分散，齐整可比”的特点，在保证实验结果准确性的同时减少实验次数；

2. 相关概念：

- 因素（因子）：影响实验结果的实验条件，即自变量；
- 水平（位级）：实验因素变化的各种状态，即自变量取值范围；
- 实验指标：用于衡量实验结果的量。

3. 正交表是正交实验中用于安排实验的一种辅助分析表格，其定义如下

$$L_{row}(n_1^{col_1} \times n_2^{col_2} \times \cdots \times n_m^{col_m}) \xrightarrow[n=n_1=n_2=\cdots=n_m]{col=col_1+col_2+\cdots+col_m} L_{row}(n^{col})$$

定义式中 L 为正交表的符号；下标 row 为正交表的行数，表示准备实验的次数，自行定义；上标 col 表示列数，即因素个数； x 为水平数，当 $n = n_1 = n_2 = \cdots = n_m$ 时表示所有因素的水平数均为 n ；

表 23.1 正交表选择参考表

因素个数	有重点因素		无重点因素	
	要求少做实验	允许多做实验	要求少做实验	允许多做实验
3	$L_8(4^1 \times 2^4)$	$L_{18}(6^1 \times 3^6)$	$L_4(2^3)$	$L_9(3^4), L_{16}(4^5)$
4	$L_8(4^1 \times 2^4)$	$L_{18}(6^1 \times 3^6)$	$L_9(3^4)$	$L_9(3^4), L_{16}(4^5)$
5	$L_8(4^1 \times 2^4)$	$L_{18}(6^1 \times 3^6), L_{16}(4^4 \times 2^3)$	$L_8(2^7)$	$L_{16}(4^5)$
6	$L_{18}(6^1 \times 3^6)$	$L_{18}(6^1 \times 3^6), L_{16}(4^2 \times 2^3)$	$L_8(2^7)$	$L_{18}(6^1 \times 3^6), L_{16}(4^4 \times 2^3)$
7	$L_{18}(6^1 \times 3^6)$	$L_{18}(6^1 \times 3^6), L_{16}(4^4 \times 2^3)$	$L_8(2^7)$	$L_{18}(6^1 \times 3^6), L_{16}(4^4 \times 2^3)$

4. 正交表具有如下特点：

- (a) 表中任意一列，不同的数字出现的次数相同；
- (b) 表中任意两列，每一行两个数字组成的有序实数对出现的次数相同。

5. 正交实验的流程大致如下：

- (a) 明确实验因素 col 、水平数 n ，确定正交实验次数 row 得到正交表并实验，得到各指标 y_1, y_2, \dots, y_{row} ；
- (b) 计算 $\bar{K}_{1,i}, \bar{K}_{2,i}, \dots, \bar{K}_{n,i}$ ，分别表示每个因素 i 的每个水平数相同的各次实验结果的平均值；
- (c) 计算每个因素 i 的极差 R_1, R_2, \dots, R_{col} ，极差越大说明该因素对指标的显著效果；

¹多因素实验设计（正交实验设计）：<https://wenku.baidu.com/view/59608555443610661ed9ad51f01dc281e53a56ec.html>

(d) 比较各列 $\overline{K_{1,i}}, \overline{K_{2,i}}, \dots, \overline{K_{n,i}}$, 选出最佳的因素水平组合。

正交实验得到的最优组合结果的取值空间为整个全面试验, 可以不包含在正交实验的方案设计中, 这是正交实验最大的优越性。

23.1.1 正交实验实例

试验号	(A) 温度 °C	空列	(B) 酯化时间 h	(C) 催化剂种类	y(乳化能力)
1	1 (130)	1	1 (3)	1 (甲)	0.56
2	1 (130)	2	2 (2)	2 (乙)	0.74
3	1 (130)	3	3 (4)	3 (丙)	0.57
4	2 (120)	1	2 (2)	3 (丙)	0.87
5	2 (120)	2	3 (4)	1 (甲)	0.85
6	2 (120)	3	1 (3)	2 (乙)	0.82
7	3 (110)	1	3 (4)	2 (乙)	0.67
8	3 (110)	2	1 (3)	3 (丙)	0.64
9	3 (110)	3	2 (2)	1 (甲)	0.66
$\overline{K_1}$	0.623	0.700	0.673	0.690	
$\overline{K_2}$	0.847	0.743	0.757	0.743	
$\overline{K_3}$	0.657	0.683	0.697	0.693	
极差 R	0.224	0.060	0.084	0.053	
因素显著性	$A \rightarrow B \rightarrow C$ 温度—酯化时间—催化剂种类				
最佳水平组	$A_2B_2C_2$ 温度 120°C, 酯化时间 2h, 催化剂种类: 乙				

23.2 卷积计算及其性质

$$x(t) \otimes y(t) = \int_{-\infty}^{\infty} x(\tau)y(t - \tau)d\tau$$

交换律 $x(t) \otimes y(t) = y(t) \otimes x(t)$

分配律 $z(t) \otimes [x(t) + y(t)] = z(t) \otimes x(t) + z(t) \otimes y(t)$

结合律 $z(t) \otimes [x(t) \otimes y(t)] = [z(t) \otimes x(t)] \otimes y(t)$

微分 $\frac{d}{dt}[x(t) \otimes y(t)] = \left[\frac{d}{dt}x(t) \right] \otimes y(t) = x(t) \otimes \left[\frac{d}{dt}y(t) \right]$

积分 $\int_{-\infty}^t x(\tau) \otimes y(\tau)d\tau = \int_{-\infty}^t x(\tau)d\tau \otimes y(t) = x(t) \otimes \int_{-\infty}^t y(\tau)d\tau$

多重微积分 $x(t)^{(m)} \otimes y(t)^{(n)} = [x(t) \otimes t(t)]^{(m+n)}$

延时 $x(t) \otimes y(t) = f(t) \implies x(t - t_1) \otimes y(t - t_1) = f(t - t_1 - t_2)$

几个特殊函数
的卷积: $f(t) \otimes \delta(t - t_0) = f(t - t_0)$

$$f(t) \otimes \delta^{(n)}(t - t_0) = f^{(n)}(t - t_0)$$

$$f(t) \otimes \varepsilon(t) = \int_{-\infty}^t f(\tau)d\tau$$

23.3 特殊矩阵运算介绍

Moore-Penrose 广义逆 又名“穆尔-彭罗斯广义逆”或简写为“M-P 广义逆”，是传统矩阵逆运算的推广。由 Moore 和 Penrose 分别于 1920 和 1955 年独立提出。记 A, G 分别为 $s \times n$ 和 $n \times s$ 复矩阵，若 A, G 满足：

$$AGA = A, \quad GAG = G, \quad (AG)^H = AG, \quad (GA)^H = GA \quad (\text{Moore-Penrose 广义逆})$$

则称矩阵 A, G 互为 M-P 广义逆矩阵，记为 $G = A^\dagger$ （或 $A = G^\dagger$ ）。上式中 X^H 称为矩阵 X 的共轭转置（hermitian），表示为对矩阵 X 的每一元素求共轭再对整体求转置。对于任意矩阵 A ，其 M-P 广义逆矩阵 A^\dagger 均存在且唯一。**M-P 广义逆的提出统一了最小二乘最优解的形式**——对于线性回归问题 $y = Ax$ ，其最佳最小二乘解写为 $x = A^\dagger y$ ，是在范数约束下的最佳逼近；

Kronecker 积 将一个矩阵的所有元素与另一个矩阵相乘并拼接得到更大的矩阵。记矩阵 $A \in \mathbb{R}_{I \times J}, B \in \mathbb{R}_{K \times L}$ ，则矩阵 A, B 的 Kronecker 积为一个大小为 $(IK) \times (JL)$ 的矩阵，写为 $A \otimes B$

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1J}B \\ a_{21}B & a_{22}B & \cdots & a_{2J}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}B & a_{I2}B & \cdots & a_{IJ}B \end{bmatrix} \quad (\text{Kronecker 积})$$

Khatri-Rao 积 定义为两个矩阵对应列向量 Kronecker 积组成的矩阵。记矩阵 $A \in \mathbb{R}_{I \times K}, B \in \mathbb{R}_{J \times K}$ ，则矩阵 A, B 的 Khatri-Rao 积为一个大小为 $(IJ) \times K$ 的矩阵，写为 $A \odot B$

$$A \odot B = \begin{bmatrix} A_1 \otimes B_1 & A_2 \otimes B_2 & \cdots & A_K \otimes B_K \end{bmatrix} \quad (\text{Khatri-Rao 积})$$

上式中 A_k, B_k 分别表示矩阵 A, B 的列向量。向量间的 Khatri-Rao 积与 Kronecker 积完全等价；

Hadamard 积 两个尺寸相同矩阵对应元素相乘。记矩阵 $A, B \in \mathbb{R}_{I \times J}$ ，则矩阵 A, B 的 Hadamard 积仍为一个大小为 $I \times J$ 的矩阵，写为 $A * B$

$$A * B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix} \quad (\text{Hadamard 积})$$

运算性质 上述几种矩阵运算具有如下性质：

1. $(A^\top)^\dagger = (A^\dagger)^\top, (A^\top A)^\dagger = A^\dagger (A^\dagger)^\top;$
2. $(A \otimes B)(C \otimes D) = AC \otimes BD;$
3. $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger;$
4. $(A \odot B) \odot C = A \odot (B \odot C);$
5. $(A \odot B)^\top (A \odot B) = A^\top A * B^\top B;$
6. $(A \odot B)^\dagger = ((A^\top A) * (B^\top B))^\dagger (A \odot B)^\top.$

23.4 矩阵求导

矩阵求导是标量求导与向量求导的延伸，在统计学、控制论、机器学习等领域有广泛的应用。具体可分为标量对矩阵求导和矩阵对矩阵求导两类，前者已有明确的数学定义和求解方法，而后的定义则具有多个版本，且尚无绝对优劣^{2,3}。

²矩阵求导术（上）：<https://zhuanlan.zhihu.com/p/24709748>

³矩阵求导术（下）：<https://zhuanlan.zhihu.com/p/24863977>

23.4.1 标量对矩阵求导

1. 假设标量函数 f 以矩阵 X 为自变量，则 f 对 X 求导本质上即是多元函数求偏导，在数学上定义为 $\frac{\partial f}{\partial X} = \left[\frac{\partial f}{\partial X_{ij}} \right]$ ，其结果为与 X 尺寸相同的矩阵。然而在实际计算中，一般不按定义逐元素求偏导，因为计算过于复杂，不符合使用矩阵以简化数学表达的初衷；

2. 为整体计算函数 f 对矩阵 X 的导数，即需要同时对所有元素 X_{ij} 求偏导，注意到按全微分公式，有

$$df = \sum_{ij} \frac{\partial f}{\partial X_{ij}} dX_{ij}$$

又因为对于尺寸相同的矩阵 A, B ，有 $\sum_{ij} A_{ij} B_{ij} = \text{tr}(A^T B) = \text{tr}(AB^T)$ ， $\text{tr}(\cdot)$ 表示矩阵的迹，则上式可写为矩阵形式

$$df = \sum_{ij} \frac{\partial f}{\partial X_{ij}} dX_{ij} = \text{tr} \left(\left(\frac{\partial f}{\partial X} \right)^T dX \right)$$

上式建立了矩阵导数与微分间的关系，故可将矩阵求导问题转化为微分问题。具体地，对于任意函数 $f(X)$ ，只需对其求微分并整理为形如 $df = \text{tr}(A^T dX)$ 的形式，即有 $\frac{\partial f}{\partial X} = A$ ；

3. 进一步介绍矩阵形式下的若干基本微分法则：

- 矩阵加减： $d(X \pm Y) = dX \pm dY$ ；
- 矩阵乘法： $d(XY) = (dX)Y + X(dY)$ ；
- 矩阵转置： $d(X^T) = (dX)^T$ ；
- 迹运算： $d(\text{tr}(X)) = \text{tr}(dX)$ ；
- 矩阵求逆： $d(X^{-1}) = -X^{-1}dXX^{-1}$ ，推导可通过对等式 $XX^{-1} = I$ 两边同时求微分实现；
- 行列式运算： $d(|X|) = \text{tr}(X^* dX)$ ，式中 X^* 为 X 的伴随矩阵；当 X 可逆时又有 $d(|X|) = |X|\text{tr}(X^{-1} dX)$ ；
- 矩阵逐元素乘法： $d(X * Y) = (dX) * Y + X * (dY)$ ， $*$ 为 Hadamard 积，定义为矩阵逐元素相乘；
- 矩阵逐元素函数： $d\sigma(X) = \sigma'(X) * dX$ ， $\sigma(\cdot)$ 为作用于矩阵各元素的函数，有 $\sigma(X) = [\sigma(X_{ij})]$ ， $\sigma'(\cdot)$ 为其导函数。

4. 因为求微分后还需整理为形如 $df = \text{tr}(A^T dX)$ 的形式才可知 $\frac{\partial f}{\partial X}$ 的结果，故还需介绍若干迹运算技巧 (trace trick) 以方便形式整理：

- 标量套上述： $x = \text{tr}(x)$ ；
- 矩阵加减： $\text{tr}(X \pm Y) = \text{tr}(X) \pm \text{tr}(Y)$ ；
- 矩阵转置： $\text{tr}(X^T) = \text{tr}(X)$ ；
- 矩阵乘法交换： $\text{tr}(XY) = \text{tr}(YX) = \sum_{ij} X_{ij} Y_{ji}$ ，其中 X 与 Y^T 尺寸相同；
- 矩阵乘法/Hadamard 乘法交换： $\text{tr}(X^T(Y * Z)) = \text{tr}((X * Y)^T Z) = \sum_{ij} X_{ij} Y_{ij} Z_{ij}$ ，其中 X, Y, Z 尺寸相同。

5. 基于上述矩阵微分法则及迹技巧即可求标量对矩阵的导数。以统计学中方差的极大似然估计为算例：



算例：总体分布协方差矩阵的极大似然估计

记样本 $x_1, \dots, x_N \sim \mathcal{N}(\mu, \Sigma)$ ，其中 x_i 为向量， Σ 为协方差矩阵（对称正定阵）。则欲求 Σ 的极大似然估计，即是求解下式对数似然函数 l 的驻点 $\frac{\partial l}{\partial \Sigma} = 0$

$$l = \ln |\Sigma| + \frac{1}{N} \sum_i (x_i - \bar{x})^T \Sigma^{-1} (x_i - \bar{x})$$

按照标量对矩阵求导的方法，首先对上式求微分。因为 Σ 可逆，且行列式 $|\Sigma|$ 为标量，则等式右侧第一项的微分有

$$d \ln |\Sigma| = \frac{1}{|\Sigma|} d|\Sigma| = \text{tr}(\Sigma^{-1} d\Sigma)$$

而等式右侧第二项的微分有

$$d \left(\frac{1}{N} \sum_i (x_i - \bar{x})^T \Sigma^{-1} (x_i - \bar{x}) \right) = \frac{1}{N} \sum_i (x_i - \bar{x})^T d(\Sigma^{-1}) (x_i - \bar{x})$$

$$\begin{aligned}
&= -\frac{1}{N} \sum_i (x_i - \bar{x})^T \Sigma^{-1} d\Sigma \Sigma^{-1} (x_i - \bar{x}) \\
&= -\frac{1}{N} \sum_i \text{tr} ((x_i - \bar{x})(x_i - \bar{x})^T \Sigma^{-1} d\Sigma \Sigma^{-1}) \\
&= \text{tr} \left(-\frac{1}{N} \sum_i S_i \Sigma^{-1} d\Sigma \Sigma^{-1} \right) = \text{tr} \left(\left(-\frac{1}{N} \sum_i S_i \right) \Sigma^{-1} d\Sigma \Sigma^{-1} \right)
\end{aligned}$$

式中 $S_i = (x_i - \bar{x})(x_i - \bar{x})^T$, 注意到 $\frac{1}{N} \sum_i S_i = S$ 即为样本协方差矩阵, 则对数似然函数 l 的微分有

$$dl = \text{tr}(\Sigma^{-1} d\Sigma) - \text{tr}(S \Sigma^{-1} d\Sigma \Sigma^{-1}) = \text{tr}((\Sigma^{-1} - S \Sigma^{-1} \Sigma^{-1}) d\Sigma) \implies \frac{dl}{d\Sigma} = (\Sigma^{-1} - S \Sigma^{-1} \Sigma^{-1})^T$$

注意到因为 S, Σ 均为 N 阶对称阵, 此时矩阵乘法交换律成立, 故上式并不唯一, 也可写为 $\frac{dl}{d\Sigma} = (\Sigma^{-1} - \Sigma^{-1} S \Sigma^{-1})^T = (\Sigma^{-1} - \Sigma^{-1} \Sigma^{-1} S)^T$ 。令 $\frac{dl}{d\Sigma} = 0$, 即可得到 Σ 的极大似然估计为 S 。

23.4.2 矩阵对矩阵求导

1.

23.5 离散概率分布采样算法

算法	初始化时间复杂度	采样时间复杂度	空间复杂度
查表法	$O(\prod_{i=1}^n d_i)$	$O(1)$	$O(\prod_{i=1}^n d_i)$
轮盘赌法	$O(n)$	$O(\ln n)$	$O(n)$
别名采样 (Alias Method)	$O(n)$	$O(1)$	$O(n)$

- 在各类算法设计中, 如何从给定概率分布中采样是一个常见的问题。随着机器学习的兴起, 抽样的场景、目标日益复杂, 对于大量复杂的分布如何设计高效抽样算法也愈加受到关注。本节主要讨论一类基础的抽样问题——离散概率分布的高效抽样^{4,5}。更为一般的针对复杂分布的抽样算法可参考 MCMC 方法 (详见第 27.1 节);
- 查表法是一种非常简单、直观的抽样算法, 适用于离散概率分布非常简单的情况。其思路是预先构造一个频数分布与被采样概率分布相一致的样本集, 再从该样本集中随机均匀采样。考虑一个简单的离散抽样问题: 假设 A、B、C、D 四种情况对应概率分别为 0.3、0.1、0.1、0.5, 则查表法需预构造序列 [A, A, A, B, C, D, D, D, D], 再从该序列中随机均匀抽取一个元素即可实现按 0.3、0.1、0.1、0.5 的概率采样。因为最终抽样时仅需按均匀分布随机抽样, 故查表法的时间复杂度仅为 $O(1)$, 与离散概率分布包含的情况数无关。但需预先构造频数分布与概率分布一致的样本集并储存, 这一过程具有较高的时间复杂度和空间复杂度;
- 轮盘赌法是一种基础的通用抽样算法, 适用于连续或离散概率分布抽样。进行离散抽样时算法不需要复杂的初始化过程, 空间复杂度也较低, 但无法实现 $O(1)$ 的时间复杂度。其思路是基于概率密度分布生成 0-1 之间单调递增的累计概率分布, 再生成 0-1 之间的随机数, 根据随机数落入的区间确定抽样的样本。假设离散概率分布涉及 n 种情况, 显然生成累计概率分布的时间和空间复杂度为 $O(n)$, 显著优于查表法。但生成随机数后为确定随机数落入的区间, 即需要找到随机数在 n 个临界点间的正确插入。若按顺序比较需要 $O(n)$ 的时间复杂度, 而基于二分法插入则需要 $O(\ln n)$ 的时间复杂度;
- 别名采样 (Alias sampling) 是一种非常高效的离散概率抽样算法, 其在采样时可以实现仅 $O(1)$ 的时间复杂度, 而准备阶段虽较轮盘赌法复杂, 但时间、空间复杂度仍保持在 $O(n)$ 。算法将按概率的离散抽样问题分解为一次等概率均匀抽样和一次二项分布抽样, 因为均匀抽样和二项分布抽样的时间复杂度均

⁴时间复杂度为 $O(1)$ 的抽样算法——别名采样 (alias sample): <https://zhuanlan.zhihu.com/p/111885669>

⁵《统计模拟》序列 7——离散随机变量别名 (alias method) 抽样法: <https://zhuanlan.zhihu.com/p/370581286>

为 $O(1)$, 从而保证了整体 $O(1)$ 的抽样时间复杂度。具体地, 假设离散概率分布满足 $\sum_i^n P(i) = 1$, 则别名采样首先令 $B_i = nP(i)$, 显然有 $\sum_i^n B_i = n$ 。进一步地算法要求构造概率序列 $\{\hat{P}(j)\}$ 满足

$$\sum_j^n \hat{P}(j) = \sum_i^n B_i = n, \quad \hat{P}(j) = \beta_{ij} B_i + \beta_{i'j} B_{i'} = 1, \quad \beta_{ij}, \beta_{i'j} \in [0, 1], \quad \sum_j^n \beta_{ij} = 1, \quad \forall i, i', j = 1, \dots, n$$

概率序列 $\{\hat{P}(j)\}$ 中每一事件 j 的概率均等, 故可基于等概率随机抽样得到一个事件 j , 而每一事件 j 均由至多两种情况 i, i' 构成, 概率分别为 $\beta_{ij} B_i, \beta_{i'j} B_{i'}$, 因此再生成一个 0-1 间的随机数即可采样 i , 又因为 β_{ij} 满足 $\sum_j^n \beta_{ij} = 1$, 故上述采样过程满足 $i \sim P(i)$ 。以上即为别名采样的流程。而为构造 $\{\hat{P}(j)\}$, 需引入别名采样算法的理论基础——对于任意离散概率分布 $\sum_i^n P(i) = 1$, 必然存在以下引理:

- $\exists i$ 使得 $P(i) \leq \frac{1}{n}$ 。证明: 反证法易证;
- 对于 $P(i) \leq \frac{1}{n}$, $\exists i' \neq i$ 使得 $P(i) + P(i') > \frac{1}{n}$ 。证明: 反证法易证。

上述引理指出, 对于任意 $0 < B_i < 1$, 则必然可以从另一个 $B_{i'}$ (满足 $B_i + B_{i'} > 1$) 中挖去一部分构造

$$\hat{P}(1) = B_i + \frac{1 - B_i}{B_{i'}} \cdot B_{i'} = 1, \quad B_i < 1, \quad B_i + B_{i'} > 1$$

同时更新 $B_{i'} \leftarrow B_{i'} - (1 - B_i)$, $B_i \leftarrow 0$ 。更新后若 $B_{i'} < 1$, 则寻找另一个 $B_{i''}$ (满足 $B_{i'} + B_{i''} > 1$) 构造 $\hat{P}(2)$; 若 $B_{i'}$ 仍大于 1, 则同样寻找另一个 $0 < B_{i''} < 1$ 构造 $\hat{P}(2)$; 直至不再存在 $0 < B_i < 1$, 则 $\{\hat{P}(j)\}$ 构造完成。

23.6 概率分布的特征函数 (*Characteristic function*)

$$\varphi_X(t) = E(e^{itX})$$

1. 在很多时候, 具体的概率分布可由其概率密度函数 (probability density function, $f(x)$) 表征, 概率密度函数相同, 则分布相同。特征函数 (Characteristic function, $\varphi(t)$) 是除此之外的另一类表示方法, 特征函数相同, 分布同样相同⁶;

2. 顾名思义, 特征函数是概率分布的特征的函数。某一概率分布可以由多个特征, 如期望、方差、偏度 (Skewness)、峰度 (Kurtosis) 等等, 如果随机变量 X, Y 的特征越相似, 则两者服从的分布也越接近, 如果两变量具有完全相同的特征, 则认为两者服从统一概率分布;

3. 观察发现, 上述期望、方差、偏度 (Skewness)、峰度 (Kurtosis) 等特征均与随机变量的矩有关:

- 期望 $\mu = E(X)$ ($E(X)$: 一阶距);
- 方差 $\sigma^2 = E(X^2) - [E(X)]^2$ ($E(X^2)$: 二阶距);
- 偏度 $Skewness = \frac{1}{\sigma^3} [E(X^3) - 3\mu\sigma^2 - \mu^3]$ ($E(X^3)$: 三阶距);
-

由此也可以以随机变量的各阶矩作为特征;

4. 对随机变量 X , 定义其特征函数 $\varphi_X(t)$ 如下, 并对指数项进行泰勒展开

$$\begin{aligned} \varphi_X(t) &= E(e^{itX}) = E(1 + itX + \frac{(itX)^2}{2!} + \dots + \frac{(itX)^n}{n!}) \\ &= 1 + itE(X) + \frac{(it)^2}{2!} E(X^2) + \dots + \frac{(it)^n}{n!} E(X^n) \end{aligned}$$

可以看到特征函数 $\varphi_X(t)$ 本质上是随机变量 X 的各阶矩的函数。若两随机变量的各阶矩完全相同, 则具有相同的特征函数, 则服从相同的概率分布。

⁶如何理解统计中的特征函数? http://www.360doc.com/content/18/0706/11/15930282_768244807.shtml

23.7 经典概率分布

23.7.1 理解 Poisson 分布

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

- ⁷首先考虑一个排队问题：某服务窗口在工作时间 $[0, T]$ 区间内迎来 X 位访客， X 是一个服从某一分布的随机变量。显然该窗口需要保证一定的服务能力以满足大多数情况下的服务需求，为此需要根据 X 的概率密度曲线得到某一分位值作为窗口服务能力设计的参照。
- 显然，在区间 $[0, T]$ 内， X 的具体取值由单位时间内访客是否出现决定。也就是说只需假定单位时间内访客出现所服从的分布，即可得到随机变量 K 的分布。考虑到乘客是否出现是一个 0-1 变量，自然地，假设单位时间内访客出现服从概率为 p 的伯努利二项分布，此时定义 K 服从泊松分布。
- 具体地，将区间 $[0, T]$ 均分为长度为 T/n 的 n 段子区间，假设每一子区间内出现访客的概率为 p ，则 $[0, T]$ 区间内 X 的期望为 np ，而出现 $X=k$ 位访客的概率

$$P(X = k) = C_n^k p^k (1-p)^{n-k} = \frac{n(n-1)\cdots(n-k+1)}{k!} p^k (1-p)^{n-k}, \quad k \leq n$$

令 $\lambda = np$ ，并将时间轴无穷细分至单位时间，即 $n \rightarrow \infty$ 时即可描述实际的访客到来情况

$$\begin{aligned} P(X = k) &= \lim_{n \rightarrow \infty} \frac{n(n-1)\cdots(n-k+1)}{k!} \left(\frac{\lambda}{n}\right)^k \left(1 - \frac{\lambda}{n}\right)^{n-k} \\ &= \frac{\lambda^k}{k!} \cdot \lim_{n \rightarrow \infty} \frac{n}{n} \cdot \frac{n-1}{n} \cdots \frac{n-k+1}{n} \cdot \lim_{n \rightarrow \infty} \left(1 - \frac{\lambda}{n}\right)^{n-k} \end{aligned}$$

易知， k 为常数，有

$$\lim_{n \rightarrow \infty} \frac{n}{n} \cdot \frac{n-1}{n} \cdots \frac{n-k+1}{n} = 1, \quad \lim_{n \rightarrow \infty} \left(1 - \frac{\lambda}{n}\right)^{n-k} = e^{-\lambda}$$

综上，泊松分布表达式如下。显然， n 很大且 p 很小时，二项分布将趋近为泊松分布。

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

23.7.2 多元高斯分布

- 定义随机变量 x 为一 d 维向量，对应 d 个特征，若每一特征均服从高斯分布，则称 x 服从 d 维高斯分布，记为 $x \sim N(x|\mu, \Sigma)$ ，其概率密度函数 $f(x)$ 为

$$f(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left[-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right] \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}, \mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \cdots & \sigma_{dd} \end{bmatrix}$$

其中 d 维向量 μ 为随机变量 x 的均值， $d \times d$ 维矩阵 Σ 为随机变量 x 的协方差矩阵；

- 多元高斯分布有一条很重要的性质，即 d 维随机变量 x 的任意子随机变量 x' 同样服从多元高斯分布，且给定总体的均值向量 μ 、协方差矩阵 Σ 和两个子随机变量 $x = (x_a, x_b)^T$ 的均值向量 μ_a, μ_b 和协方差矩阵 Σ_a, Σ_b ，可以按下式确定条件概率分布函数 $f(x_a|x_b)$ ⁸

$$\begin{cases} \mu_{a|b} = \mu_a + \Sigma_{ab} \Sigma_{bb}^{-1} (x_b - \mu_b) \\ \Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ba} \end{cases} \quad x = \begin{bmatrix} x_a \\ x_b \end{bmatrix}, \mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix} \quad (\text{条件分布})$$

⁷如何通俗理解泊松分布？https://blog.csdn.net/ccnt_2012/article/details/81114920

⁸多元条件高斯分布的均值和方差的数学推导（Bishop: Pattern Recognition and Machine Learning 第二章）：https://blog.csdn.net/qq_38402294/article/details/102467418

3. 进一步地证明上式。首先仅考虑 $f(x)$ 的指数项，将 x, μ, Σ 拆解为 $x_a, x_b, \mu_a, \mu_b, \Sigma_{aa}, \Sigma_{ab}, \Sigma_{bb}$ ，有

$$\begin{aligned} -\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu) &= -\frac{1}{2}(x_a - \mu_a)^T \Lambda_{aa}(x_a - \mu_a) - \frac{1}{2}(x_b - \mu_b)^T \Lambda_{ba}(x_a - \mu_a) \\ &\quad - \frac{1}{2}(x_a - \mu_a)^T \Lambda_{ab}(x_b - \mu_b) - \frac{1}{2}(x_b - \mu_b)^T \Lambda_{bb}(x_b - \mu_b) \\ &= -\frac{1}{2}(x_a - \mu_a)^T \Lambda_{aa}(x_a - \mu_a) - (x_a - \mu_a)^T \Lambda_{ab}(x_b - \mu_b) - \frac{1}{2}(x_b - \mu_b)^T \Lambda_{bb}(x_b - \mu_b) \end{aligned}$$

式中 Λ 为协方差矩阵 Σ 的逆矩阵

$$\Lambda = \begin{bmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{bmatrix} = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}^{-1}$$

假设 x_b 已知，则根据拆解结果显然有

$$\begin{aligned} f(x_a|x_b) &\sim \exp \left[-\frac{1}{2}(x_a - \mu_a)^T \Lambda_{aa}(x_a - \mu_a) - (x_a - \mu_a)^T \Lambda_{ab}(x_b - \mu_b) - \frac{1}{2}(x_b - \mu_b)^T \Lambda_{bb}(x_b - \mu_b) \right] \\ &= \exp \left[-\frac{1}{2}x_a^T \Lambda_{aa} x_a + x_a^T [\Lambda_{aa}\mu_a - \Lambda_{ab}(x_b - \mu_b)] + \text{const} \right] \end{aligned}$$

const 为与随机变量 x_a 无关的常量。而根据定义，条件概率 $f(x_a|x_b)$ 有

$$\begin{aligned} f(x_a|x_b) &\sim \exp \left[-\frac{1}{2}(x_a - \mu_{a|b})^T \Sigma_{a|b}^{-1}(x_a - \mu_{a|b}) \right] \\ &= \exp \left[-\frac{1}{2}x_a^T \Sigma_{a|b}^{-1} x_a + x_a^T \Sigma_{a|b}^{-1} \mu_{a|b} + \text{const} \right] \end{aligned}$$

显然只需要通过对比系数即可以确定条件概率参数 $\mu_{a|b}, \Sigma_{a|b}$

$$\begin{cases} \Sigma_{a|b}^{-1} = \Lambda_{aa} \\ \Sigma_{a|b}^{-1} \mu_{a|b} = \Lambda_{aa}\mu_a - \Lambda_{ab}(x_b - \mu_b) \end{cases} \implies \begin{cases} \Sigma_{a|b} = \Lambda_{aa}^{-1} \\ \mu_{a|b} = \mu_a - \Lambda_{aa}^{-1} \Lambda_{ab}(x_b - \mu_b) \end{cases}$$

最后只需要将上式的 $\Lambda_{aa}, \Lambda_{ab}$ 替换为 $\Sigma_{aa}, \Sigma_{ab}, \Sigma_{ba}, \Sigma_{bb}$ 即可。由分块矩阵求逆法则，有

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} M & -MBD^{-1} \\ -D^{-1}CM & D^{-1} + D^{-1}CMBD^{-1} \end{bmatrix} \quad M = (A - BD^{-1}C)^{-1}$$

则 $\Lambda_{aa} = (\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}$, $\Lambda_{ab} = -(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}\Sigma_{ab}\Sigma_{bb}^{-1}$, 代入即可得到 $\mu_{a|b}, \Sigma_{a|b}$ 的表达式；

4. 多元高斯分布的另一个重要性质即是其共轭先验 (conjugate prior) 特性——即假设观测样本 $\{x_1, \dots, x_n\}$ 服从 $N(\mu, \Sigma)$ 分布，其中 Σ 已知，且假设参数 μ 服从高斯先验 $N(\mu_0, \Sigma_0)$ ，则参数 μ 的后验分布仍服从高斯分布 $N(\mu_n, \Sigma_n)$ ，且有

$$\mu_n = \Sigma_0 \left(\Sigma_0 + \frac{1}{n} \Sigma \right)^{-1} \bar{x} + \frac{1}{n} \Sigma \left(\Sigma_0 + \frac{1}{n} \Sigma \right)^{-1} \mu_0, \quad \Sigma_n = \frac{1}{n} \Sigma_0 \left(\Sigma_0 + \frac{1}{n} \Sigma \right)^{-1} \Sigma \quad (\text{后验分布})$$

5. 进一步给出上式推导过程。基于贝叶斯公式得到 μ 的后验分布 $p(\mu|x_1, \dots, x_n)$ 有

$$\begin{aligned} p(\mu|x_1, \dots, x_n) &\propto p(x_1, \dots, x_n|\mu)p(\mu) \\ &= \prod_i N(x_i|\mu, \Sigma)N(\mu|\mu_0, \Sigma_0) \propto \exp \left\{ -\frac{1}{2} \sum_i (x_i - \mu)^T \Sigma^{-1} (x_i - \mu) \right\} \exp \left\{ -\frac{1}{2} (\mu - \mu_0)^T \Sigma_0^{-1} (\mu - \mu_0) \right\} \end{aligned}$$

仅考虑指数项内部部分对其展开，有

$$-\frac{1}{2} \left[(\mu - \mu_0)^T \Sigma_0^{-1} (\mu - \mu_0) + \sum_i (x_i - \mu)^T \Sigma^{-1} (x_i - \mu) \right]$$

$$= -\frac{1}{2} \left[\mu^\top \Sigma_0^{-1} \mu - 2\mu_0^\top \Sigma_0^{-1} \mu + \mu_0^\top \Sigma_0^{-1} \mu_0 + \sum_i x_i^\top \Sigma^{-1} x_i - 2n\mu^\top \Sigma^{-1} \bar{x} + n\mu^\top \Sigma^{-1} \mu \right]$$

因为此处讨论 μ 的后验分布，故可删去上式中与 μ 无关的项，从而得到

$$\begin{aligned} p(\mu|x_1, \dots, x_n) &\propto N(\mu_n, \Sigma_n) \propto \exp \left\{ -\frac{1}{2} [\mu^\top \Sigma_0^{-1} \mu - 2\mu_0^\top \Sigma_0^{-1} \mu + 2n\mu^\top \Sigma^{-1} \bar{x} + n\mu^\top \Sigma^{-1} \mu] \right\} \\ &= \exp \left\{ -\frac{1}{2} [\mu^\top (\Sigma_0^{-1} + n\Sigma^{-1}) \mu - 2\mu^\top (\Sigma_0^{-1} \mu_0 + n\Sigma^{-1} \bar{x})] \right\} \end{aligned}$$

又因为 $N(\mu_n, \Sigma_n) \propto \exp\{-\frac{1}{2}(\mu - \mu_n)^\top \Sigma_n^{-1}(\mu - \mu_n)\} \propto \exp\{-\frac{1}{2}(\mu^\top \Sigma_n^{-1} \mu - 2\mu^\top \Sigma_n^{-1} \mu_n)\}$ ，对比系数易知

$$\Sigma_n = (\Sigma_0^{-1} + n\Sigma^{-1})^{-1}, \quad \mu_n = \Sigma_n (\Sigma_0^{-1} \mu_0 + n\Sigma^{-1} \bar{x})$$

进一步代入 $(A^{-1} + B^{-1})^{-1} = A(A+B)^{-1}B = B(A+B)^{-1}A$ ，最终得到

$$\begin{aligned} \Sigma_n &= (\Sigma_0^{-1} + n\Sigma^{-1})^{-1} = \Sigma_0 \left(\Sigma_0 + \frac{1}{n} \Sigma \right)^{-1} \frac{1}{n} \Sigma \\ \mu_n &= \Sigma_n (\Sigma_0^{-1} \mu_0 + n\Sigma^{-1} \bar{x}) = \Sigma_0 \left(\Sigma_0 + \frac{1}{n} \Sigma \right)^{-1} \frac{1}{n} \Sigma (\Sigma_0^{-1} \mu_0 + n\Sigma^{-1} \bar{x}) = \Sigma_0 \left(\Sigma_0 + \frac{1}{n} \Sigma \right)^{-1} \bar{x} + \frac{1}{n} \Sigma \left(\Sigma_0 + \frac{1}{n} \Sigma \right)^{-1} \mu_0 \end{aligned}$$

一元高斯分布与多元高斯分布的 KL 散度

高斯分布假设是大量统计学模型和机器学习模型的基础，而 KL 散度（见第 23.9.6 节）又是一种通用的概率分布间距离的度量函数。下文将先后考虑一元高斯分布与多元高斯分布的情况，推导高斯分布间 KL 散度的数学形式^a。首先考虑一元高斯分布的情况。记 $N_1(x|\mu_1, \sigma_1^2), N_2(x|\mu_2, \sigma_2^2)$ 分别表示两个高斯分布，则两者的 KL 散度按定义计算为

$$D_{KL}(N_1||N_2) = \int_{-\infty}^{\infty} N_1(x) \ln \frac{N_1(x)}{N_2(x)} dx = \int_{-\infty}^{\infty} N_1(x) \ln N_1(x) dx - \int_{-\infty}^{\infty} N_1(x) \ln N_2(x) dx$$

首先考虑第一项，按高斯分布概率密度函数展开有

$$\begin{aligned} \int_{-\infty}^{\infty} N_1(x) \ln N_1(x) dx &= \int_{-\infty}^{\infty} N_1(x) \left(\ln \frac{1}{\sqrt{2\pi}\sigma_1} - \frac{(x-\mu_1)^2}{2\sigma_1^2} \right) dx \\ &= -\frac{1}{2} \ln (2\pi\sigma_1^2) - \frac{1}{2\sigma_1^2} \int_{-\infty}^{\infty} N_1(x)(x-\mu_1)^2 dx = -\frac{1}{2} [\ln (2\pi\sigma_1^2) + 1] \end{aligned}$$

为快速计算上式，一方面用到了 $\int_{-\infty}^{\infty} N_1(x) dx = 1$ 的结论，另一方面还需注意到 $\int_{-\infty}^{\infty} N_1(x)(x-\mu_1)^2 dx$ 实际上就是计算随机变量 $x \sim N_1(x|\mu_1, \sigma_1^2)$ 的方差，显然有 $\int_{-\infty}^{\infty} N_1(x)(x-\mu_1)^2 dx = \sigma_1^2$ 。进一步计算 $D_{KL}(N_1||N_2)$ 展开式的第二项

$$\begin{aligned} \int_{-\infty}^{\infty} N_1(x) \ln N_2(x) dx &= \int_{-\infty}^{\infty} N_1(x) \left(\ln \frac{1}{\sqrt{2\pi}\sigma_2} - \frac{(x-\mu_2)^2}{2\sigma_2^2} \right) dx \\ &= -\frac{1}{2} \ln (2\pi\sigma_2^2) - \frac{1}{2\sigma_2^2} \int_{-\infty}^{\infty} N_1(x)(x-\mu_1+\mu_1-\mu_2)^2 dx \\ &= -\frac{1}{2} \ln (2\pi\sigma_2^2) - \frac{1}{2\sigma_2^2} \int_{-\infty}^{\infty} N_1(x)(x-\mu_1)^2 dx - \frac{\mu_1-\mu_2}{\sigma_2^2} \int_{-\infty}^{\infty} N_1(x)(x-\mu_1) dx - \frac{(\mu_1-\mu_2)^2}{2\sigma_2^2} \\ &= -\frac{1}{2} \ln (2\pi\sigma_2^2) - \frac{1}{2} - \frac{(\mu_1-\mu_2)^2}{2\sigma_2^2} = -\frac{1}{2} \left[\ln (2\pi\sigma_2^2) + 1 + \frac{(\mu_1-\mu_2)^2}{\sigma_2^2} \right] \end{aligned}$$

综上得到一元高斯分布的 KL 散度 $D_{KL}(N_1||N_2)$

$$D_{KL}(N_1||N_2) = -\frac{1}{2} \left[\ln (2\pi\sigma_1^2) + 1 - \left(\ln (2\pi\sigma_2^2) + 1 + \frac{(\mu_1-\mu_2)^2}{\sigma_2^2} \right) \right] = \frac{1}{2} \ln \frac{\sigma_2}{\sigma_1} + \frac{(\mu_1-\mu_2)^2}{2\sigma_2^2} \quad (\text{一元高斯分布 KL 散度})$$

进一步扩展至多元高斯分布的情况。记 $N_1(x|\mu_1, \Sigma_1), N_2(x|\mu_2, \Sigma_2)$ 分别表示两个 n 维高斯分布，其中 $\Sigma = \mathbb{E}_N[(x-\mu)(x-\mu)^\top] \in \mathbb{R}^{n \times n}$ 为协方差矩阵，则同理先考虑 KL 散度的第一项

$$\int_{-\infty}^{\infty} N_1(x) \ln N_1(x) dx = \int_{-\infty}^{\infty} N_1(x) \left(\ln \frac{1}{\sqrt{2\pi^n |\Sigma_1|}} - \frac{(x-\mu_1)^\top \Sigma_1^{-1} (x-\mu_1)}{2} \right) dx = -\frac{1}{2} \ln (2\pi^n |\Sigma_1|) - \frac{1}{2} \mathbb{E}_{N_1} [(x-\mu_1)^\top \Sigma_1^{-1} (x-\mu_1)]$$

注意到 $x^\top Ax$ 计算结果为标量，有 $x^\top Ax = \text{tr}(x^\top Ax) = \text{tr}(Ax x^\top)$ ，其中 $\text{tr}(\cdot)$ 表示迹运算，又因为迹运算为加和线性运算，有 $\mathbb{E}[\text{tr}(Ax x^\top)] = \text{tr}(\mathbb{E}[Ax x^\top]) = \text{tr}(A \mathbb{E}[xx^\top])$ ，故上式可进一步简化为

$$\int_{-\infty}^{\infty} N_1(x) \ln N_1(x) dx = -\frac{1}{2} \ln (2\pi^n |\Sigma_1|) - \frac{1}{2} \mathbb{E}_{N_1} [(x-\mu_1)^\top \Sigma_1^{-1} (x-\mu_1)]$$

$$\begin{aligned}
&= -\frac{1}{2} \ln(2\pi^n |\Sigma_1|) - \frac{1}{2} \text{tr}(\Sigma_1^{-1} \mathbb{E}_{N_1} [(x - \mu_1)(x - \mu_1)^\top]) \\
&= -\frac{1}{2} \ln(2\pi^n |\Sigma_1|) - \frac{1}{2} \text{tr}(\Sigma_1^{-1} \Sigma_1) = -\frac{1}{2} \ln(2\pi^n |\Sigma_1|) - \frac{n}{2}
\end{aligned}$$

再考虑 KL 散度的第二项

$$\int_{-\infty}^{\infty} N_1(x) \ln N_2(x) dx = -\frac{1}{2} \ln(2\pi^n |\Sigma_2|) - \frac{1}{2} \mathbb{E}_{N_1} [(x - \mu_2) \Sigma_2^{-1} (x - \mu_2)^\top] = -\frac{1}{2} \ln(2\pi^n |\Sigma_2|) - \frac{1}{2} \text{tr}(\Sigma_2^{-1} \mathbb{E}_{N_1} [(x - \mu_2)(x - \mu_2)^\top])$$

上式中

$$\begin{aligned}
\mathbb{E}_{N_1} [(x - \mu_2)(x - \mu_2)^\top] &= \mathbb{E}_{N_1} [(x - \mu_1 + \mu_1 - \mu_2)(x - \mu_1 + \mu_1 - \mu_2)^\top] \\
&= \mathbb{E}_{N_1} [(x - \mu_1)(x - \mu_1)^\top] + \mathbb{E}_{N_1} [(x - \mu_1)(\mu_1 - \mu_2)^\top] + (\mu_1 - \mu_2) \mathbb{E}_{N_1} [(x - \mu_1)^\top] + (\mu_1 - \mu_2)(\mu_1 - \mu_2)^\top \\
&= \Sigma_1 + (\mu_1 - \mu_2)(\mu_1 - \mu_2)^\top
\end{aligned}$$

故 KL 散度的第二项整理为

$$\begin{aligned}
\int_{-\infty}^{\infty} N_1(x) \ln N_2(x) dx &= -\frac{1}{2} \ln(2\pi^n |\Sigma_2|) - \frac{1}{2} \text{tr}(\Sigma_2^{-1} \mathbb{E}_{N_1} [(x - \mu_2)(x - \mu_2)^\top]) \\
&= -\frac{1}{2} \ln(2\pi^n |\Sigma_2|) - \frac{1}{2} \text{tr}(\Sigma_2^{-1} (\Sigma_1 + (\mu_1 - \mu_2)(\mu_1 - \mu_2)^\top)) \\
&= -\frac{1}{2} \ln(2\pi^n |\Sigma_2|) - \frac{1}{2} \text{tr}(\Sigma_2^{-1} \Sigma_1) - \frac{1}{2} (\mu_1 - \mu_2)^\top \Sigma_2^{-1} (\mu_1 - \mu_2)
\end{aligned}$$

综上得到 n 维高斯分布的 KL 散度 $D_{KL}(N_1 || N_2)$

$$\begin{aligned}
D_{KL}(N_1 || N_2) &= -\frac{1}{2} \ln(2\pi^n |\Sigma_1|) - \frac{n}{2} - \left(-\frac{1}{2} \ln(2\pi^n |\Sigma_2|) - \frac{1}{2} \text{tr}(\Sigma_2^{-1} \Sigma_1) - \frac{1}{2} (\mu_1 - \mu_2)^\top \Sigma_2^{-1} (\mu_1 - \mu_2) \right) \\
&= \frac{1}{2} \left[\ln \frac{|\Sigma_2|}{|\Sigma_1|} + \text{tr}(\Sigma_2^{-1} \Sigma_1) + (\mu_1 - \mu_2)^\top \Sigma_2^{-1} (\mu_1 - \mu_2) - n \right]
\end{aligned} \quad (\text{n 维高斯分布 KL 散度})$$

^a多元高斯分布间的 KL 散度及其 Pytorch 实现: https://zhuanlan.zhihu.com/p/640473373?utm_id=0

23.7.3 威沙特分布 (wishart distribution) 与高斯-威沙特分布 (normal-wishart distribution)

1. 威沙特分布以统计学家约翰·威沙特命名, 是一种半正定矩阵随机分布, 在多变量分析中至关重要;
2. 假设 n 个随机变量独立同服从于 p 维正态分布 $X_1, \dots, X_n \sim N(0, \Sigma)$ (X_i 记为行向量)。记矩阵 $X = [X_1, \dots, X_n]^\top$ 为 $n \times p$ 维矩阵, 则称矩阵 $W = \sum_i X_i^\top X_i = X^\top X$ 服从威沙特分布 (wishart distribution), 记为 $W \sim W_p(n, \Sigma)$ 。威沙特分布与常见的卡方分布、伽马分布具有密切联系:
 - 卡方分布定义为多个独立服从标准正态分布随机变量的平方和;
 - 伽马分布可视为多个独立服从以 0 为期望的正态分布随机变量的平方和;
 - 威沙特分布定义为多个独立服从以 0 为期望的多维正态分布随机变量的平方 (外积) 和。
3. 根据定义可知, 威沙特分布的数学期望为 $\mathbb{E}(W) = n\Sigma$, 推导如下

$$\mathbb{E}(W) = \mathbb{E}\left(\sum_i X_i^\top X_i\right) = \sum_i \mathbb{E}((X_i - 0)^\top (X_i - 0)) = \sum_i COV(X_i) = n\Sigma$$

由此可知, 威沙特分布的一个重要作用即是描述多元正太分布样本的协方差。假设独立同分布样本服从 p 维正态分布 $X_1, \dots, X_n \sim N(\mu, \Sigma)$, 则样本协方差矩阵 S 有

$$(n-1)S \sim W_p(n-1, \Sigma), \quad S = \frac{1}{n-1} \sum_i (X_i - \bar{X})^\top (X_i - \bar{X})$$

4. 另外可写出威沙特分布的概率密度函数如下

$$f(W) = \frac{|W|^{(n-p-1)/2} \exp\{-\text{trace}(\Sigma^{-1} W)/2\}}{2^{np/2} |\Sigma|^{n/2} \Gamma_p(n/2)}, \quad \Gamma_p\left(\frac{n}{2}\right) = \pi^{p(p-1)/4} \prod_{j=1}^p \Gamma\left(\frac{n+1-j}{2}\right)$$

上式中 $\Gamma_p(\cdot)$ 为多元伽马函数;

5. 基于威沙特分布描述样本协方差，又以高斯分布描述样本期望，则可描述样本分布，相应的建模思路即为高斯-威沙特分布 (**normal-wishart distribution**)。具体地，若均值 μ 满足如下高斯分布 $\mu \sim N(\mu_0, (\lambda\Lambda)^{-1})$ ，而 Λ 又满足威沙特分布 $\Lambda \sim W(v, W)$ ，则称 (μ, Λ) 服从高斯-威沙特分布，记为 $(\mu, \Lambda) \sim NW(\mu_0, \lambda, W, v)$ ；
6. 在统计建模中，高斯-威沙特分布一般作为期望与精度矩阵（协方差矩阵的逆）未知的多元正态分布的共轭先验。其概率密度函数有

$$f(\mu, \Lambda | \mu_0, \lambda, W, v) = N(\mu | \mu_0, (\lambda\Lambda)^{-1}) \cdot W(\Lambda | v, W)$$

7. 根据定义，若 (μ, Λ) 服从高斯-威沙特分布，则 Λ 的边际分布 (**marginal distribution**) 为威沙特分布， μ 的边际分布为多元 t 分布， Λ 确定时 μ 的条件分布 (**conditional distribution**) 为多元高斯分布；
8. 高斯-威沙特分布的一个重要性质是其满足共轭先验 (**conjugate prior**)——即若假设超参先验服从高斯-威沙特分布 $(\mu, \Lambda) \sim NW(\mu_0, \lambda, W, v)$ ，则在获得多维正态分布样本集 $X = [X_1, \dots, X_n]^\top$ 后超参后验仍服从高斯-威沙特分布 $(\mu, \Lambda) \sim NW(\mu_n, \lambda_n, W_n, v_n)$ ，且有

$$\mu_n = \frac{\lambda\mu_0 + n\bar{X}}{\lambda + n}, \quad \lambda_n = \lambda + n, \quad W_n^{-1} = W_0^{-1} + \sum_i^n (X_i - \bar{X})^\top (X_i - \bar{X}) + \frac{n\lambda}{n + \lambda} (\bar{X} - \mu_0)^\top (\bar{X} - \mu_0), \quad v_n = v + n$$

9. 根据定义，采样生成服从高斯-威沙特分布的样本也并不复杂——首先基于参数 v, W 生成 v 个独立服从多元高斯分布 $N(0, W)$ 的样本 X_i 以计算 $\Lambda = \sum_i X_i^\top X_i$ ，而后基于参数 μ_0 和协方差 $(\lambda\Lambda)^{-1}$ 生成 μ 即可得到样本 (μ, Λ) 。

23.7.4 指数族分布 (**exponential family distribution**)

1. 指数族分布 (**exponential family distribution**) 的标准表达式如下

$$p(x|\eta) = h(x) \exp(\eta^T \Phi(x) - A(\eta)) = \frac{1}{\exp(A(\eta))} h(x) \exp(\eta^T \Phi(x))$$

$$\exp(A(\eta)) = \int h(x) \exp(\eta^T \Phi(x)) dx$$

式中 η 为自然参数 (**natural parameter**)、 $A(\eta)$ 称为对数配分函数 (**log partition function**) ($\exp(A(\eta))$ 称为配分函数)、 $\Phi(X) = \sum_i \Phi(x_i)$ 称为 η 的充分统计量 (**sufficient statistic**)⁹；

2. 指数族分布具有充分统计量、共轭先验、最大熵三类性质：

- 指数族分布中 $\Phi(X) = \sum_i \Phi(x_i)$ 为 η 的充分统计量，即统计量 $\Phi(X)$ 包含了随机变量 X 的所有信息，此时在基于样本集 $D = \{x_1, \dots, x_n\}$ 得到统计量 $\Phi(X)$ 后即可抛去样本集直接由 $\Phi(X)$ 推断参数 η ，进而确定概率分布形式。进一步地证明可由统计量 $\Phi(X) = \sum_i \Phi(x_i)$ 直接推断 η 的极大似然估计值 $\hat{\eta}_{MLE}$

$$\begin{aligned} \hat{\eta}_{MLE} &= \arg \max_{\eta} \log \prod_{i=1}^n p(x_i|\eta) = \arg \max_{\eta} \sum_{i=1}^n \log p(x_i|\eta) \\ &= \arg \max_{\eta} \sum_{i=1}^n (\log h(x_i) + \eta^T \Phi(x_i) - A(\eta)) \\ &= \arg \max_{\eta} \sum_{i=1}^n (\eta^T \Phi(x_i) - A(\eta)) = \arg \max_{\eta} \left(\eta^T \sum_{i=1}^n \Phi(x_i) - nA(\eta) \right) \end{aligned}$$

$\hat{\eta}_{MLE}$ 即为上式导函数的零点

$$\frac{\partial}{\partial \eta} \left(\eta^T \sum_{i=1}^n \Phi(x_i) - nA(\eta) \right) = \sum_{i=1}^n \Phi(x_i) - nA'(\eta) = 0 \implies A'(\hat{\eta}_{MLE}) = \frac{1}{n} \sum_{i=1}^n \Phi(x_i) = \frac{1}{n} \Phi(X)$$

显然，在基于全体样本得到统计量 $\Phi(X)$ 后即可推断出参数 η ，从而确定整体的分布情况，而不再需要原始样本数据的信息，因此 $\Phi(X)$ 为充分统计量；

⁹注意 $\Phi(X)$ 不同于 $\Phi(x)$ ，前者是关于随机变量 X 的函数，后者是关于样本 x 的函数。

- 对于任意指数族分布，均存在自然参数 η 的共轭先验 (**conjugate prior**)。在贝叶斯公式中，后验分布 $p(\eta|X)$ 、似然 $p(X|\eta)$ 与先验分布 $p(\eta|\tau)$ 满足如下关系

$$p(\eta|X) \propto p(X|\eta)p(\eta|\tau)$$

上式中 τ 为超参。若先验分布与后验分布属于同类分布，则称先验分布与后验分布为共轭分布，而先验分布被称为似然函数的共轭先验。假设随机变量 X 服从指数族分布，则似然函数 $p(X|\eta)$ 为

$$p(X|\eta) = \left(\prod_{i=1}^n h(x_i) \right) \exp \left\{ \eta^T \sum_{i=1}^n \Phi(x_i) - nA(\eta) \right\}$$

为使得先验分布为似然函数的共轭先验，只需令 $p(\eta|\tau)$ 也取指数族分布形式 $p(\eta|\tau) \propto \exp\{\eta^T \tau_1 - \tau_2 A(\eta)\}$ ，有

$$\begin{aligned} p(\eta|X) &\propto \left(\prod_{i=1}^n h(x_i) \right) \exp \left\{ \eta^T \sum_{i=1}^n \Phi(x_i) - nA(\eta) \right\} \exp\{\eta^T \tau_1 - \tau_2 A(\eta)\} \\ &\propto \exp \left\{ \eta^T \left(\sum_{i=1}^n \Phi(x_i) + \tau_1 \right) - (n + \tau_2)A(\eta) \right\} \end{aligned}$$

可以看到，后验分布同样具有指数族分布形式，与先验分布为共轭分布；

- 指数族分布满足最大熵的思想，即对于经验分布利用最大熵原理导出的分布就是指数族分布。对于样本集 $D = \{x_1, \dots, x_n\}$ ，可以基于样本集分布直方图得到经验分布 $\hat{p}(x)$

$$\hat{p}(x) = \frac{\text{Count}(x)}{n}$$

最大熵思想指出，拟合的最优分布函数 $p(x)$ 应在满足 $\hat{p}(x)$ 的基础上使得熵最大，即不确定性最大，即除了样本集之外不再引入其它信息

$$\begin{aligned} p(x) &= \arg \min_p \sum_x p(x) \ln p(x) \\ \text{s.t. } &\sum_x p(x) = 1 \\ &\mathbb{E}_p[f(x)] = \mathbb{E}_{\hat{p}}[f(x)] \end{aligned}$$

上式第二道约束中 $f(X)$ 为关于随机变量 X 的任意函数，即概率 p 下得到的 $f(X)$ 的期望应该等于经验概率 \hat{p} 下的 $f(X)$ 的期望。基于拉格朗日乘数法求解上述规划问题

$$\begin{aligned} p(x) &= \arg \min_p L(x, \lambda_1, \lambda_2) = \arg \min_p \left\{ \sum_x p(x) \ln p(x) + \lambda_1 \left(1 - \sum_x p(x) \right) + \lambda_2 (\mathbb{E}_{\hat{p}}[f(x)] - \mathbb{E}_p[f(x)]) \right\} \\ \implies \frac{\partial L}{\partial p(x)} &= \ln p(x) + 1 - \lambda_1 - \lambda_2 f(x) = 0 \implies p^*(x) = \exp(\lambda_2 f(x) - (1 - \lambda_1)) \end{aligned}$$

可以看到，满足最大熵原理的最优概率分布为 $h(x) = 1, \eta = \lambda_2, A(\eta) = 1 - \lambda_1$ 的指数族分布。

3. 指数族分布的特殊性质使其广泛应用于机器学习各项任务中，是广义线性模型、概率图模型、变分推断等模型或方法的核心：

- 指数族分布是唯一拥有有限充分统计量的分布族，此时可将海量数据压缩至有限大小的充分统计量而不丢失数据信息，有助于在线学习任务（在线学习任务中观测的数据量将随时间而增长，此时无需记录所有样本，仅需计算相应的充分统计量）；
- 指数族分布是唯一存在共轭先验的分布，此时后验分布和先验分布具有相同的分布形式，有助于直接计算后验分布。LDA (Latent Dirichlet Allocation) 文档主题生成模型是一个三层贝叶斯概率模型，为基于贝叶斯公式计算待估参数的后验分布，模型采用属于指数族分布的狄利克雷分布作为先验分布，利用指数族分布的共轭先验特性确保后验分布依然服从狄利克雷分布，便于计算；
- 当样本满足指数族分布时，可方便计算期望和方差 ($\mathbb{E}[X] = A'(\eta), \mathbb{D}[X] = A''(\eta)$)。在变分推断中，经常要计算期望，通过这个性质，便可以将期望计算转化成求导计算。

典型分布转化

1. Bernoulli 分布

若 X 服从参数为 π 的 Bernoulli 分布, 有

$$\begin{aligned} p(x|\pi) &= \pi^x(1-\pi)^{1-x} = \exp\{x \ln \pi + (1-x) \ln(1-\pi)\} \\ &= \exp\left\{x \ln \frac{\pi}{1-\pi} + \ln(1-\pi)\right\} = \exp\left\{x \ln \frac{\pi}{1-\pi} - \ln\left(1+e^{\ln \frac{\pi}{1-\pi}}\right)\right\} \\ \implies h(x) &= 1 \quad T(x) = x \quad \eta = \ln \frac{\pi}{1-\pi} \quad A(\eta) = \ln(1-e^\eta) \end{aligned}$$

2. Poisson 分布

若 X 服从参数为 λ 的 Poisson 分布, 有

$$\begin{aligned} p(x|\lambda) &= \frac{\lambda^x e^{-\lambda}}{x!} = \frac{1}{x!} \exp\{x \ln \lambda - \lambda\} = \frac{1}{x!} \exp\{x \ln \lambda - e^{\ln \lambda}\} \\ \implies h(x) &= \frac{1}{x!} \quad T(x) = x \quad \eta = \ln \lambda \quad A(\eta) = e^\eta \end{aligned}$$

3. Gaussian 分布

若 X 服从参数为 μ, σ^2 的 Gaussian 分布, 有

$$\begin{aligned} p(x|\mu, \sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x - \left(\frac{\mu^2}{2\sigma^2} + \frac{1}{2}\ln\sigma^2\right)\right\} \\ &= \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x - \left(-\frac{1}{4}\frac{\mu^2}{\sigma^4} - \frac{1}{2}\ln\left(-2\frac{1}{2\sigma^2}\right)\right)\right\} \\ \implies h(x) &= \frac{1}{\sqrt{2\pi}} \quad T(x) = [x^2, x]^T \quad \eta = [\eta_1, \eta_2]^T = \left[-\frac{1}{2\sigma^2}, \frac{\mu}{\sigma^2}\right]^T \quad A(\eta) = -\frac{\eta_2^2}{4\eta_1} - \frac{1}{2}\ln 2\eta_1 \end{aligned}$$

23.7.5 幂律分布 (Power law distribution)

$$p(x) \propto x^{-\alpha} \quad \alpha > 0$$

^{10,11} 概率密度函数满足上式的概率分布即称为幂律分布, 式中 α 称为尺度参数 (scaling parameter), 一般介于 (2, 3)。因为幂律分布属于重尾分布, 在重尾分布中 α 决定了尾部的具体形状, 故又称为尾部指数 (tail-index)。在实际应用中, 很少有完全符合幂律分布的数据, 幂律分布更多地用于分析一组数据中高于特定阈值 (x_{\min}) 的部分, 即拟合某一概率分布的尾部。对于服从连续和离散幂律分布的随机变量, 其概率密度函数具体如下:

$$p(x) = \begin{cases} P(x < X < x + dx) = Cx^{-\alpha} = \frac{\alpha-1}{x_{\min}} \left(\frac{x}{x_{\min}}\right)^{-\alpha} & \text{Continuous} \\ P(x = X) = Cx^{-\alpha} = \frac{x^{-\alpha}}{\sum_{n=0}^{\infty} (n+x_{\min})^{-\alpha}} & \text{Discrete} \text{ (仅考虑 } x \text{ 为整数)} \end{cases}$$

上式中 x_{\min} 不是样本的最小值, 而是相应幂律分布定义域的左极限。当随机变量完全服从幂律分布时抽样的最小值不一定是 x_{\min} ; 当随机变量不服从幂律分布, 但属于重尾分布 (如 T 分布) 时, 因为幂律分布也属于重尾分布, 因此常用幂律分布拟合其它重尾分布的尾部, 以进行极值理论 (extreme value theory) 分析, 此时需要从样本中筛去小于 x_{\min} 的观察值。 $\sum_{n=0}^{\infty} (n+x_{\min})^{-\alpha}$ 可写作 $\zeta(\alpha, x_{\min})$, 称为赫尔维茨函数 (Hurwitz zeta function)。同样地, 可计算其累积概率函数:

$$P(x) = \int_x^{\infty} p(u)du = \begin{cases} \left(\frac{x}{x_{\min}}\right)^{-\alpha+1} & \text{Continuous} \\ \frac{\sum_{n=0}^{\infty} (n+x_{\min})^{-\alpha}}{\sum_{n=0}^{\infty} (n+x_{\min})^{-\alpha}} = \frac{\zeta(\alpha, x)}{\zeta(\alpha, x_{\min})} & \text{Discrete} \text{ (仅考虑 } x \text{ 为整数)} \end{cases}$$

¹⁰ 幂律分布 <http://wiki.swarma.net/index.php?title=%E5%B9%82%E5%BE%8B%E5%88%86%E5%B8%83&oldid=19296#.E5.85.B6.E4.BB.96.E5.BD.A2.E5.BC.8F.EF.BC.88Variants.EF.BC.89>

¹¹ Clauset A, Shalizi C R, Newman M E, et al. Power-Law Distributions in Empirical Data[J]. Siam Review, 2009, 51(4): 661-703.

与正态分布类似，幂律分布同样是自然界中的一种常见现象。不同之处在于正态分布反映了世界的同质性——大多数个体都是相似且平凡的，极端个体明显偏少；而幂律分布反映了世界的异质性——极少数的个体掌握了绝大部分的资源，而剩余绝大部分个体分享剩余的一小部分资源。第一个为幂律分布命名的是经济学家帕累托。他发现，在19世纪的意大利，极少数的富人赚走了绝大部分的钱，大部分家庭的收入都很低。他的这一发现被后人称为“帕累托法则”，也叫“二八定律”，也就是20%的人获得了80%的收入。

幂律分布参数估计

1. 幂律分布的主要参数即其尾部指数 α 。考虑到幂律分布概率密度在双对数图下呈线性的特点，可以简单地基于最小二乘法(least-squares)进行线性拟合，得到的斜率的绝对值即为 α 。然而，因为幂律分布为重尾分布，在尾部的抽样具有较大的波动性，使得估计结果存在较大误差。因此在参数估计时存在若干技巧：

不采用最小二乘估计，选择最大似然估计(maximum likelihood estimate, MLE)。因为幂律分布又分连续模型与离散模型，两类模型的最大似然估计量计算式不同。以下简单给出服从幂律分布的连续随机变量的尾部指数 α 的最大似然估计量 $\hat{\alpha}$ 的公式推导：

$$\begin{aligned} p(x) &= \frac{\alpha-1}{x_{\min}} \left(\frac{x}{x_{\min}}\right)^{-\alpha} \implies p(x|\alpha) = \prod_{i=1}^n \frac{\alpha-1}{x_{\min}} \left(\frac{x_i}{x_{\min}}\right)^{-\alpha} \\ &\implies L(x|\alpha) = \ln p(x|\alpha) = \sum_{i=1}^n \left[\ln(\alpha-1) - \ln x_{\min} - \alpha \ln \frac{x_i}{x_{\min}} \right] \\ &\implies \frac{\partial L}{\partial \alpha} = \sum_{i=1}^n \left[\frac{1}{\alpha-1} - \ln \frac{x_i}{x_{\min}} \right] = \frac{n}{\alpha-1} - \sum_{i=1}^n \ln \frac{x_i}{x_{\min}} \end{aligned}$$

令上式 $\frac{\partial L}{\partial \alpha} = 0$ ，有

$$\hat{\alpha} = 1 + n \left[\sum_{i=1}^n \ln \frac{x_i}{x_{\min}} \right]^{-1} \quad \text{Continuous}$$

以上估计量正好为Hill估计量(Hill estimator)，是重尾分布尾部指数最常用的估计量。当样本为离散样本时 α 的最大似然估计量推导较为复杂，以下直接给出计算结果，同样地假设随机变量为整数

$$\hat{\alpha} \cong 1 + n \left[\sum_{i=1}^n \ln \frac{x_i}{x_{\min} - \frac{1}{2}} \right]^{-1} \quad \text{Discrete}$$

注意到连续变量与离散变量下的 α 估计公式并不完全相同，应避免混用。

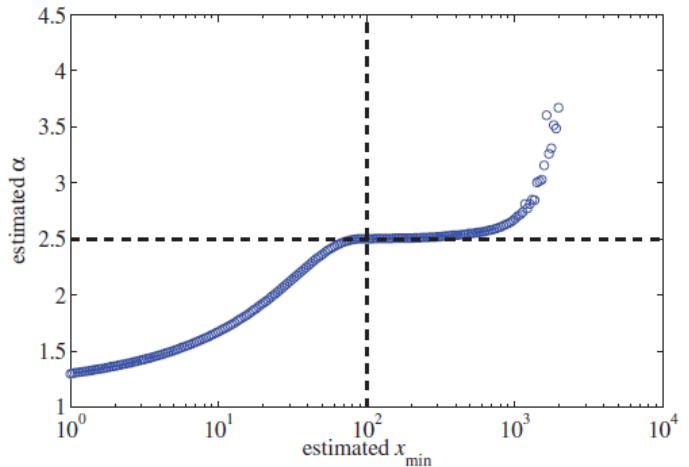
对累积概率函数而非概率密度函数进行线性拟合。注意到幂律分布的累积概率函数在双对数图下同样呈线性。通过逐项累积缓解了重尾分布尾部抽样稳定性差的特点，因此累积概率函数图往往比概率密度函数图更加稳定。此时可以不采用最大似然估计，仅由最小二乘拟合即可得到较准确的估计结果。

2. 观察以上 α 的估计公式，注意到需要首先估计阈值 x_{\min} 。特别是在分析尾部数据时，选取的 x_{\min} 偏小会使得截取的样本不服从幂律分布， x_{\min} 过大又会因为丢失过多信息引入更多误差。 α 估计量本质上是Hill估计量，因此对应的 x_{\min} 选取问题也就属于Hill估计量的阈值选择问题。以下介绍几种方法：

Hill-plot估计这是一种经典的Hill估计量阈值定性估计方法。注意到 $\hat{\alpha}$ 与 x_{\min} 存在函数对应关系，代入一系列 x_{\min} 即可得到相应的 $\hat{\alpha}$ 。将点对 $(x_{\min}, \hat{\alpha})$ 可视化即为Hill-plot。选择曲线最平稳部分的起点所对应的 x_{\min} 即为阈值的估计量。这是因为只要在 $x > \hat{x}_{\min}$ 时随机变量服从幂律分布，则对任意 $x_{\min} > \hat{x}_{\min}$ ，同样有 $x > x_{\min}$ 服从幂律分布，且具有相同的尾部指数 α ，因此计算的 $\hat{\alpha}$ 不应该有很大的波动性；

拟合优度检验其思路是选择最合适的 \hat{x}_{\min} ，使得样本与理论幂律分布模型具有最大的拟合优度。对于非正态分布的定量数据，最常用的拟合优度检验为**KS检验**。此时使得统计量 D_n 最小的 \hat{x}_{\min} 即达到最高的拟合优度。

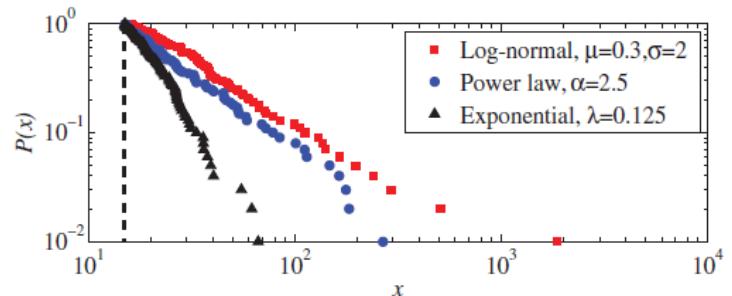
图 23.1 实验中假设数据服从 $x_{\min} = 100$, $\alpha = 2.5$ 的幂律分布，并基于概率模型抽样，样本总量为 2500，绘制 Hill-plot，可以看到曲线有一明显的平稳段，且大致以 $x_{\min} = 100$ 为起点，此时也可得到正确的估计量 $\hat{\alpha} = 2.5$ 。



幂律分布假设检验

- 已有的统计研究很少严谨地检验数据是否服从幂律分布，往往仅凭双对数图定性推断。然而需要说明的是：频数（概率）分布在双对数图上呈近似线性仅是数据服从幂律分布的必要条件，而非充要条件，受参数、抽样或样本数的影响，服从其它概率分布的样本也可能在双对数图上表现出线性；

图 23.2 部分概率分布在双对数图下的概率分布，包含对数正态分布、幂律分布和指数分布。样本数 100，属于小样本。



- 幂律分布检验的目的即是为了评价观测样本与拟合的概率模型是否具有相同的分布特征，同样可以选择 **KS 拟合优度检验**。假设观测样本量为 n ，则同样地生成大小为 n 的理论样本，并比较两者分布的一致性。以下简单介绍几点细节：

理论样本集生成 需要说明的是，为了同时评价 $\hat{\alpha}$, \hat{x}_{\min} 的估计效果，当观测样本分布仅在尾部服从幂律分布，则生成的理论样本不能只包括 $x > \hat{x}_{\min}$ 的幂律分布部分，还要求在 $x < \hat{x}_{\min}$ 部分与观测样本也具有相同的分布规律。以下简单介绍满足以上要求的理论样本集生成方法。假设观测样本中满足 $x > \hat{x}_{\min}$ 的样本量为 n_{tail} ，则在生成理论样本集时，以 $\frac{n_{tail}}{n}$ 的概率生成服从幂律分布的样本，以 $\frac{n - n_{tail}}{n}$ 的概率从满足 $x > \hat{x}_{\min}$ 的观察样本中随机抽取一组样本，往复循环 n 次即得到大小为 n 的理论样本；

零假设拒绝条件 为避免查表，可通过比较 KS 检验的 p 值与预设阈值（显著水平）的大小关系选择拒绝或接受零假设。 p 值越高说明样本分布越接近幂律分布。一般地设置显著水平 0.05，即当 $p < 0.05$ 时认为最优幂律分布无法有效拟合观测样本，反之则认为样本可由幂律分布拟合。在一些苛刻的条件下，可将阈值调高至 0.1；

理论样本集数量 生成以组理论样本集，即可与观测样本集比较得到一个 p 值，然而因为理论样本集的生成带有随机性，为了更有力地证明显著性检验的结论，往往生成多组理论样本集并以多个 p 值的均值作为判别依据。理论样本集数量越多，显著性检验的结论就越精确。一般地，如果希望平均 p 值与真值的误差不超过 ϵ ，则需要生成至少 $\frac{1}{4\epsilon^2}$ 组理论样本集。

指数截止的幂律分布 (power law with exponential cutoff)

指数截止的幂律分布即是在幂率的基础上乘上一个指数函数，即

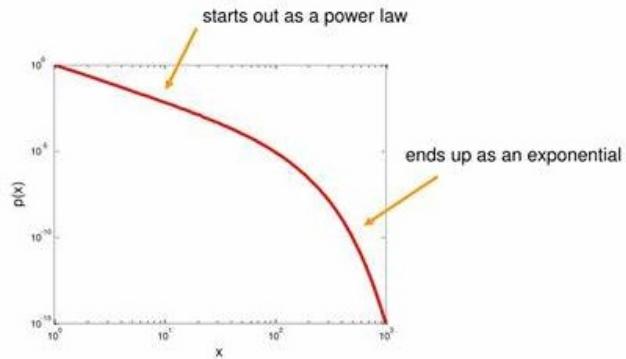
$$p(x) = \frac{\lambda^{1-\alpha}}{\Gamma(1-\alpha, \lambda x_{\min})} x^{-\alpha} e^{-\lambda x} \Rightarrow \frac{\partial \ln p(x)}{\partial \ln x} = -\alpha - \lambda x$$

可以看到，在双对数坐标轴下，图象的斜率由小逐渐增大。随着 x 的增加，指数项会在某个时刻超过正常的幂律分布，此时概率密度函数会以指数衰减，类似于“截断”的效果。这种分布是渐近幂律分布的常见替代方法，因为它考虑了有限大小的影响，适用于拟合后小部分数据不服从幂律分布的情况。对于另一类常见情况，即前小部分数据不服从幂律分布时，一类简单且行之有效的方法是仅分析其最大的部分数据，如取最大的 \sqrt{n} 或 $\frac{n}{10}$ 组数据。

Another common distribution: power-law with an exponential cutoff

■ $p(x) \sim x^{-\alpha} e^{-k/x}$

图 23.3 指数截止的幂律分布。在前半段表现出幂律分布的特征，后半段表现出负指数分布的特征。因为负指数分布的衰减速度快于幂律分布，因而表现出“截断”效果。



23.8 混合高斯模型与 EM 算法

23.8.1 混合高斯模型 (gaussian Mixture Model, GMM)

$$p(X=x) = \sum_{k=1}^K p(k)p(x|k) = \sum_{k=1}^K \pi_k N(x|\mu_k, \Sigma_k) \quad \sum_{k=1}^K \pi_k = 1$$

- 正态分布又称高斯分布，利用高斯分布描述事件的模型称为高斯模型。当模型仅含一个高斯分布时，称为单高斯模型，单高斯模型包含 μ, Σ 两组参数，其一般概率密度函数 $f(x)$ 如下，式中 d 为随机变量 x 的维度

$$f(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left[-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right] \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}, \mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \cdots & \sigma_{dd} \end{bmatrix}$$

- 在其它一些时候，事件的分布为两个或以上个高斯分布之和，此时即需要选择混合高斯模型。考虑以下实例：研究人员将研究一批志愿者身高所服从的分布，志愿者中有男有女，显然不同性别的人群的身高服从两个不同的高斯分布，则整体的身高分布为以上两高斯分布的加权和，对应的权重为志愿者中男女所占的比例；
- 因此，混合高斯模型除了包含 μ_k, Σ_k 两组参数外，还需另一组表示各模型权重的参数 π_k ，并满足 $\sum_{k=1}^K \pi_k = 1$ ， K 为模型中所包含的高斯模型的个数。得到 μ_k, Σ_k, π_k 的最优估计量即可得到模型的表达式；

¹²EM 算法和混合高斯模型（一）：<https://www.jianshu.com/p/008025aaad25>

4. 极大似然估计量是一类常用的估计量，可由极大似然估计法得到（似然的概念见 1*VI）。然而极大似然估计法在估计 GMM 模型的最优参数时将存在问题。假设共有 N 个样本点，则由对数似然函数 $\ln L(\mu, \Sigma, \pi)$ 得到 $\hat{\mu}_k, \hat{\Sigma}_k, \hat{\pi}_k$ 的过程如下：

$$\ln L(\mu, \Sigma, \pi) = \ln \left[\prod_{n=1}^N \sum_{k=1}^K \pi_k N(x_n | \mu_k, \Sigma_k) \right] = \sum_{n=1}^N \ln \sum_{k=1}^K \pi_k N(x_n | \mu_k, \Sigma_k) \quad \begin{cases} \frac{\partial L(\mu, \Sigma, \pi)}{\partial \hat{\mu}_k} = 0 \\ \frac{\partial L(\mu, \Sigma, \pi)}{\partial \hat{\Sigma}_k} = 0 \\ \frac{\partial L(\mu, \Sigma, \pi)}{\partial \hat{\pi}_k} = 0 \end{cases}$$

上述对数似然函数 $\ln L(\mu, \Sigma, \pi)$ 中对数项含求和式，将为偏导解方程带来困扰。

23.8.2 EM 算法 (expectation maximum)

- ¹³具体而言，只要模型中包含多个分布（混合模型），直接对对数似然函数求偏导解得极大似然估计量的方法均会产生上述问题，其根本原因在于数据的不完整性。再次以身高分布的例子为例，预测某人身高实际包含两个步骤：
 - 基于直接获取的自变量判断其性别；
 - 再根据相关性别的身高分布实现预测。
 步骤一将会得到一中间变量，用以判断数据源于混合模型中的哪个模型，这一中间变量被称为隐变量，似然函数略去隐变量而直接建立了自变量与似然值之间的联系，从而造成数据的不完整；
- EM 算法用于估计混合模型参数的极大似然估计量。算法引入隐变量，并采用逐次迭代逼近的方法替代直接求导。算法包含两个步骤：E-step (expectation-step, 期望步) 和 M-step (maximum-step, 最大化步)；
- 假设 m 个独立样本 $x = (x_1, \dots, x_m)$ ，对应有 m 个隐变量 $z = (z_1, \dots, z_m)$ ，此时 (x, z) 组成完全数据，未知的隐变量 z 同样需要优化，因此待优化的目标函数由 $\ln L(\theta)$ 变为 $\ln L(\theta, z)$

$$\theta, z = \arg \max_{\theta, z} \ln L(\theta, z) = \arg \max_{\theta, z} \sum_{i=1}^m \ln \sum_{z_i} P(x_i, z_i | \theta)$$

构造 $P(x_i, z_i | \theta) = Q_i(z_i) \frac{P(x_i, z_i | \theta)}{Q_i(z_i)}$ ， $Q_i(z_i)$ 为一未知分布，满足 $\sum_{z_i} Q_i(z_i) = 1$ ， $0 \leq Q_i(z_i) \leq 1$ ，相当于身高分布例子中男女比例的分布。因为对数函数为严格凹函数，由 Jensen 不等式¹⁴即可得到上式的下界函数 $B(\theta, Q_i(z_i))$

$$\sum_{i=1}^m \ln \sum_{z_i} P(x_i, z_i | \theta) = \sum_{i=1}^m \ln \sum_{z_i} Q_i(z_i) \frac{P(x_i, z_i | \theta)}{Q_i(z_i)} \geq \sum_{i=1}^m \sum_{z_i} Q_i(z_i) \ln \frac{P(x_i, z_i | \theta)}{Q_i(z_i)} = B(\theta, Q_i(z_i))$$

$\sum_{z_i} Q_i(z_i) \ln \frac{P(x_i, z_i | \theta)}{Q_i(z_i)}$ 可以看做变量 $\ln \frac{P(x_i, z_i | \theta)}{Q_i(z_i)}$ 的期望，这也就是期望步的由来；

- 当等式成立时，下界函数 $B(\theta, Q_i(z_i))$ 即与对数似然函数 $\ln L(\theta, z)$ 同时取得最大值，充要条件为随机变量为常数

$$\frac{P(x_i, z_i | \theta)}{Q_i(z_i)} = c \implies P(x_i, z_i | \theta) = c \cdot Q_i(z_i) \implies \sum_{z_i} P(x_i, z_i | \theta) = c \sum_{z_i} Q_i(z_i) = c$$

因此有

$$Q_i(z_i) = \frac{P(x_i, z_i | \theta)}{c} = \frac{P(x_i, z_i | \theta)}{\sum_{z_i} P(x_i, z_i | \theta)} = \frac{P(x_i, z_i | \theta)}{P(x_i | \theta)} = P(z_i | x_i, \theta)$$

以上推导说明 $Q_i(z_i)$ 的本质为已知样本和模型参数下的隐变量分布；

- 在最大化步中，通过最大化下界函数 $B(\theta, Q_i(z_i))$ 实现对数似然函数 $\ln L(\theta, z)$ 的最大化，即最大化步；

¹³EM 算法原理：<https://www.cnblogs.com/coshaho/p/9573367.html>

¹⁴Jensen 不等式：如果函数 $f(x)$ 为凹函数， x 为随机变量，则存在 $f(E(x)) \geq E(f(x))$ ，当 $f(x)$ 为严格凹函数时，当且仅当 x 为常量时等号成立。

6. 具体地, EM 算法计算流程如下:

- 期望步: 初始化模型参数 θ , 基于样本计算隐变量的后验分布 $Q_i(z_i) = P(z_i|x_i, \theta)$, 此时 $B(\theta, Q_i(z_i))$ 达到最大, 有 $B(\theta, Q_i(z_i)) = \ln L(\theta, z)$;
- 最大化步: 固定 $Q_i(z_i)$, 此时在隐变量分布已知的情况下即可由最大似然估计得到新的模型参数值 $\theta = \arg \max_{\theta} \ln L(\theta, z)$;
- 循环以上两步直至 θ 收敛。

23.9 距离(相似性)度量

23.9.1 点间距离——闵可夫斯基距离 (Minkowski distance)

1. 闵可夫斯基距离又称闵氏距离, 是一组距离的定义式。假设 n 维样本点 $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n)$, 则 x, y 的闵氏距离 $D(x, y)$ 定义为

$$D(x, y) = \left(\sum_{k=1}^n |x_k - y_k|^p \right)^{1/p}$$

2. 以上定义式类似于 p -范数的定义, 当 $p = 1$ 时表示曼哈顿距离、 $p = 2$ 时表示欧式距离、 $p = \infty$ 时表示切比雪夫距离。当样本各维度间存在尺度差异时, 可首先将样本标准化再计算其闵氏距离, 但闵氏距离假设各维度相互独立, 故样本各维度间存在相关性时闵氏距离将不适用。

23.9.2 点间距离——马氏距离 (Mahalanobis Distance)

1. 马氏距离由马哈拉诺比斯 (P. C. Mahalanobis) 提出, 是欧式距离的推广, 适用于样本各维度间存在线性相关性时的距离计算, 且马氏距离无需考虑各维度的尺度差异;

2. 对 n 维样本点 $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n)$, x 距原点的马氏距离 $D(x)$ 和 x, y 的马氏距离 $D(x, y)$ 定义为

$$D(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \quad D(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)}$$

上式中 $\mu = (\mu_1, \dots, \mu_n)$ 为多维随机变量的均值向量, Σ 为多维随机变量的协方差矩阵, 即 $\Sigma_{ij} = \text{Cov}\{X_i, X_j\} = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]$, 其中 X_i, X_j 表示随机变量的第 i, j 个维度分量;

3. 马氏距离本质上是对样本作 **cholesky** 变换, 将线性相关变量转化为线性无关变量后再求欧式距离, 将线性相关变量转化为线性无关变量的过程同 PCA 算法一致。当各维度相互独立时, Σ 对角线元素为方差的对角阵, 马氏距离退化为标准化的欧式距离;

4. 考虑到马氏距离所基于的 **cholesky** 变换为线性变换, 故算法无法有效处理样本各维度存在非线性相关时的情况。联系马氏距离与 PCA 降维算法的联系, 可考虑基于非线性降维算法实现样本各维度存在非线性相关时的距离计算问题。例如基于 t-SNE 算法首先解析样本点之间的相似性并将其嵌入低维空间, 再基于一般的距离计算公式求解样本点间的距离。

23.9.3 序列间距离——Hausdorff 距离

1. Hausdorff 距离是描述两组点集之间相似性的一种度量, 可用于计算轨迹与轨迹之间的距离;

2. 假设两条轨迹 $X = \{x_1, \dots, x_m\}, Y = \{y_1, \dots, y_n\}$, 则定义从轨迹 X 到轨迹 Y 的单向 Hausdorff 距离 $h(X, Y)$, 从轨迹 Y 到轨迹 X 的单向 Hausdorff 距离 $h(Y, X)$ 和双向 Hausdorff 距离 $H(X, Y)$ 分别如下

$$h(X, Y) = \max_{x \in X} \min_{y \in Y} \|x - y\| \quad h(Y, X) = \max_{y \in Y} \min_{x \in X} \|y - x\| \quad H(X, Y) = \max\{h(X, Y), h(Y, X)\}$$

式中 $\|x - y\|$ 表示 x, y 的欧式距离。可以看到, 双向 Hausdorff 距离可解释为两轨迹间最近点距离的最大值, 距离越小则两轨迹越接近;

3. 因为 Hausdorff 距离计算时考虑了轨迹的每一个样本点, 故原始 Hausdorff 距离易受异常点的影响, 因此 Huttenlocher 于 1933 年通过引入分位数提出了部分 Hausdorff 距离的概念。定义分位数 K , 则单向部分

Hausdorff 距离 $h_K(X, Y)$ 和双向部分 Hausdorff 距离 $H_K(X, Y)$ 定义为

$$h_K(X, Y) = K^{\text{th}} \max_{x \in X} \min_{y \in Y} \|x - y\| \quad H_K(X, Y) = \max\{h_K(X, Y), h_K(Y, X)\}$$

23.9.4 序列间距离——EDR (Edit Distance on Real sequence) 距离

1. Hausdorff 距离是一种基于空间距离的轨迹相似性度量方法, EDR 距离则是一种基于计数 (count-based) 的轨迹相似性度量方法, 定义为需要对轨迹 X 进行插入、删除、替换使其变为轨迹 Y 的次数, 其思路类似于量化概率分布相似性的推土机距离 (见 23.9.7 节);
2. 基于动态规划的 EDR 距离 $D(X, Y)$ 计算公式如下

$$D(X, Y) = \begin{cases} n & m = 0 \\ m & n = 0 \\ \min \left\{ \begin{array}{l} D(\text{Rest}(X), \text{Rest}(Y) + \text{subcost}) \\ D(\text{Rest}(X), Y) + 1 \\ D(X, \text{Rest}(Y)) + 1 \end{array} \right\} & \text{otherwise} \end{cases}$$

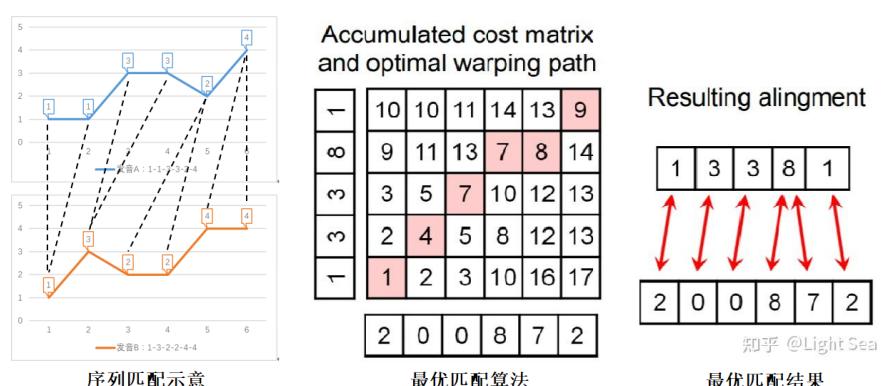
$$\text{subcost} = \begin{cases} 0 & D(\text{Head}(X), \text{Head}(Y)) \leq \varepsilon \\ 1 & \text{otherwise} \end{cases}$$

式中 $\text{Head}(X)$ 表示轨迹序列的第一个元素, $\text{Rest}(X)$ 表示除第一个元素后的剩余元素集合, ε 为预设的最小距离阈值, 两点之间距离小于该值时将被认为是同一点。EDR 距离对异常值相对不敏感, 但实际应用时难以确定合适的 ε 值。

23.9.5 序列间距离——动态时间规整 (dynamic time warping, DTW)

1. 动态时间规整 (dynamic time warping, DTW) 是上世纪 60 年代由日本学者 Itakura 提出的一种量化序列间距离的算法。算法最早应用于语音识别, 是针对孤立词识别的一种经典算法, 如今常用于量化时间序列间的相似性¹⁵;
2. 语音信号具有很强的随机性, 不同的发音习惯, 发音时所处的环境不同, 心情不同都会导致发音持续时间长短不一的现象。表现在序列结构上即是两串信息相同的序列因为时间轴的无规律扭曲而在形态上存在差异 (如序列 [1, 2, 1] 和 [1, 2, 2, 1], 后者可理解为前者在“2”处被“拉长”的结果)。为识别形态不同但实质相同的两条序列, 不同于一般的距离度量方法直接量化对象间的差异, DTW 会首先“还原”序列沿时间轴的无规律扭曲, 寻找两个序列的最优匹配, 再计算匹配后的距离。因为寻找两个序列最优匹配的过程并非简单地对时间轴作线性变换, 而是非线性伸缩, 这即算法名称中“warp (扭曲)”的由来;
3. 所谓序列最优匹配实际上即是将两个序列的索引作一对一或一对多匹配, 以使得两者间的总距离最小。匹配时遵循以下基本原则:

图 23.4 动态时间规整算法示意。自左至右分别展示序列匹配的基本原理、基于累计距离矩阵的最优匹配计算方法和最优匹配结果的解读。



¹⁵崔岩的笔记——动态时间规整算法 (Dynamic Time Warping, DTW): https://blog.csdn.net/qq_43587949/article/details/127307459

- 序列的每个索引必须与令一个序列的一个或多个索引匹配，反之亦然；
- 序列的首个索引必须至少匹配另一个序列的首个索引；
- 序列索引和另一个序列索引的匹配必须是单调递增的，反之亦然。记第一个序列的两个索引 $j > i$ 和第二个序列的两个索引 $l > k$ ，则不可出现 j 匹配 k 而 i 匹配 l 。

总而言之，序列匹配时必须保证头尾对齐，且内部不能交叉匹配或遗漏；

4. 进一步介绍算法的具体实现。记两序列分别为 $A \in \mathbb{R}^M$, $B \in \mathbb{R}^N$ 。首先需自定义距离公式以量化两序列间任意元素的距离 $d(A_i, B_j)$ 。距离公式也是算法中唯一需要人为设定的内容。基于 $d(A_i, B_j)$ 构建累计距离矩阵 (accumulated cost matrix) $D \in \mathbb{R}^{N \times M}$ 。以 $D_{1,1}$ 表示左下角元素， $D_{M,N}$ 表示右上角元素，则任意元素 $D_{i,j}$ 满足如下递推关系

$$D_{i,j} = d(A_i, B_j) + \min \{D_{i-1,j}, D_{i-1,j-1}, D_{i,j-1}\}, \quad \forall i = 1, \dots, M, j = 1, \dots, N$$

可以看到矩阵 D 的元素 $D_{i,j}$ 由两部分组成——一部分为元素 A_i, B_j 的距离；另一部分为 $D_{i,j}$ 左侧相邻元素 $D_{i-1,j}$ 、下方相邻元素 $D_{i,j-1}$ 、和左下方元素 $D_{i-1,j-1}$ 的最小值；

5. 基于累计距离矩阵 D 也可快速计算最优规整路径 (optimal warping path)，表示两个序列间的最优匹配方案。具体计算方法是从矩阵右上角元素 $D_{M,N}$ 出发，在左下角三个元素 $\{D_{M-1,N}, D_{M-1,N-1}, D_{M,N-1}\}$ 中选择最小元素作为下一节点，依次类推直达左下角元素 $D_{1,1}$ ，即为最优规整路径。假设路径中包含的元素为 $\{D_{1,1}, D_{2,2}, D_{3,3}, D_{4,4}, D_{5,4}, D_{6,5}\}$ ，表示最优匹配方案为 $\{(A_1, B_1), (A_2, B_2), (A_3, B_3), (A_4, B_4), (A_5, B_4), (A_6, B_5)\}$ ；
6. 记最优规整路径包含 K 个元素，每个元素为 $D_{i^*(k), j^*(k)}$ ，则序列 A, B 的 DTW 距离为

$$DTW(A, B) = \frac{D_{M,N}}{K} = \frac{1}{K} \sum_k D_{i^*(k), j^*(k)}$$

7. 以上即为动态时间规整算法的具体流程，本质上是一种动态规划算法。

23.9.6 概率分布间的距离——从信息量到 KL 散度与交叉熵

$$I(x_i) = -\ln p(X = x_i) \tag{23.1}$$

$$H(X) = -\sum_i^n p(X = x_i) \ln p(X = x_i) \tag{23.2}$$

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \ln p(x, y) \tag{23.3}$$

$$H(X|Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \ln p(x|y) \tag{23.4}$$

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \ln \frac{p(x, y)}{p(x)p(y)} \tag{23.5}$$

$$D_{KL}(p||q) = \sum_i^n p(X = x_i) \ln \frac{p(X = x_i)}{q(X = x_i)} \tag{23.6}$$

$$H(p, q) = -\sum_i^n p(X = x_i) \ln q(X = x_i) \tag{23.7}$$

1. ¹⁶信息量、信息熵、相对熵 (KL 散度) 与交叉熵等均是香农提出的信息论中的重要概念。信息论是运用概率论与数理统计的方法研究信息、信息熵、通信系统、数据传输、密码学、数据压缩等问题的应用数学学科；
2. 信息量 (amount of information) 是对信息多少的度量。信息论中认为信源输出的消息是随机的。即在未收到消息之前，是不能肯定信源到底发送什么样的消息。而通信的目的也就是要使接收者在接收到

¹⁶一文搞懂交叉熵在机器学习中的使用，透彻理解交叉熵背后的直觉：<https://blog.csdn.net/tsyccnh/article/details/79163834>

消息后，尽可能多的解除接收者对信源所存在的疑义（不定度），因此这个被解除的不定度实际上就是在通信中所要传送的信息量，根据定义，显然事件 $X = x_i$ 信息量 $I(x_i)$ 的大小与其概率有关

$$I(x_i) = -\ln p(X = x_i)$$

3. 信息量是对事件 $X = x_i$ 所含信息多少的度量，而信息熵 (**information entropy**) $H(X)$ 则是对随机变量 X 所含信息多少的度量，显然 $H(X)$ 为 $I(x_i)$ 的数学期望

$$H(X) = -\sum_i^n p(X = x_i) \ln p(X = x_i)$$

一般情况下，默认信息熵计算公式中以 2 为底，计算单位为比特 (bit)，也可换算成以 e 为底，对应单位为奈特 (nat)：

4. 上述信息量和信息熵仅关注一个随机变量 X 。参考条件概率和联合概率的定义，对于两个随机变量 X, Y ，也存在条件熵 (**conditional entropy**) $H(X|Y)$ 和联合熵 $H(X, Y)$ 。联合熵的定义和物理意义与信息熵基本一致，仅将概率分布替换为联合概率分布

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \ln p(x, y)$$

条件熵 $H(X|Y)$ 则定义为在 Y 给定的情况下 X 的条件概率分布的信息熵关于 Y 的期望

$$H(X|Y) = \sum_{y \in Y} p(y) H(X|Y = y) = -\sum_{y \in Y} p(y) \sum_{x \in X} p(x|y) \ln p(x|y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \ln p(x|y)$$

条件熵的物理意义是在得知某一确定随机变量的基础上获取另一变量所需的期望信息量。显然条件熵、联合熵和信息熵之间满足如下关系

$$H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$$

5. 根据条件熵的物理意义易知 $H(X) - H(X|Y) \geq 0$ ，且若 $H(X) - H(X|Y) = 0$ 表示随机变量 Y 对确定 X 毫无作用，即 X, Y 相互独立，而 $H(X) - H(X|Y)$ 越大则表示 X, Y 之间相关性越高。因此可基于 $H(X) - H(X|Y)$ 量化随机变量 X, Y 之间的距离，在信息论中定义为互信息 (**mutual information**) $I(X, Y)$

$$\begin{aligned} I(X, Y) &= H(X) - H(X|Y) = -\sum_{x \in X} p(x) \ln p(x) + \sum_{x \in X} \sum_{y \in Y} p(x, y) \ln p(x|y) \\ &= -\sum_{x \in X} \left(\sum_{y \in Y} p(x, y) \right) \ln p(x) + \sum_{x \in X} \sum_{y \in Y} p(x, y) \ln p(x|y) \\ &= \sum_{x \in X} \sum_{y \in Y} p(x, y) \ln \frac{p(x|y)}{p(x)} = \sum_{x \in X} \sum_{y \in Y} p(x, y) \ln \frac{p(x, y)}{p(x)p(y)} \end{aligned}$$

由上式可知，互信息具有对称性和非负性，其物理意义为已知一个随机变量后另一随机变量不确定性的降低程度；

6. 相对熵 (**relative entropy**) 又称 KL 散度 (**Kullback-Leibler divergence**, KLD) ($D_{KL}(p||q)$)，定义为由概率分布 q 拟合随机变量 X 的概率分布 p 时所产生的信息损耗

$$D_{KL}(p||q) = \sum_i^n p(X = x_i) \ln \frac{p(X = x_i)}{q(X = x_i)}$$

相对熵为非负，且 $D_{KL}(p||q)$ 与 $D_{KL}(q||p)$ 不一定相等，当且仅当 $q = p$ 时 $D_{KL}(p||q) = D_{KL}(q||p) = 0$ ；

7. 交叉熵 (**cross entropy**) $H(p, q)$ 的定义来源于相对熵，对相对熵的定义式展开如下

$$D_{KL}(p||q) = \sum_i^n p(X = x_i) \ln \frac{p(X = x_i)}{q(X = x_i)} = -H_p(X) + \left[-\sum_i^n p(X = x_i) \ln q(X = x_i) \right]$$

因为 p 为随机变量 X 的真实分布, 显然 $H_p(X)$ 为常量, 概率分布 q 为自变量, 因此只需计算后面一项即可评估概率分布 q 相对 p 造成的信息增益, 即是交叉熵的定义

$$H(p, q) = - \sum_i^n p(X = x_i) \ln q(X = x_i)$$

8. ¹⁷在利用神经网络训练分类器时, 常常使用交叉熵而非均方误差 (**mean-square error, MSE**) 作为模型的损失函数, 而后者多用于回归器中, 具体原因与两者应用于最速下降法时的不同效果有关。假设神经网络层数为 m 、激励函数 f 为 sigmoid、预测的结果为 $\{y'_1, \dots, y'_n\}$, 而真实值为 $\{y_1, \dots, y_n\}$, 则用均方误差和交叉熵分别表示的分类结果误差 J 如下

$$\begin{cases} MSE: & J_1 = \frac{1}{2n} \sum_i^n (y_i - y'_i)^2 \\ cross\ entropy: & J_2 = - \sum_i^n y_i \ln y'_i \end{cases}$$

以误差函数对权值 w_{ij}^{m-1} 的偏导为例说明均方误差函数所存在的缺陷 (求导相关推导可参考章节 24.1)

$$\frac{\partial J}{\partial w_{ij}^{m-1}} = \frac{\partial J}{\partial y'_i} \frac{\partial y'_i}{\partial u'_i} \frac{\partial u'_i}{\partial w_{ij}^{m-1}} = \frac{\partial J}{\partial y'_i} f' y_j^{m-1} = \frac{\partial J}{\partial y'_i} y'_i (1 - y'_i) y_j^{m-1} = \begin{cases} -\frac{1}{n} (y_i - y'_i) y'_i (1 - y'_i) y_j^{m-1} & J = J_1 \\ -y_i (1 - y'_i) y_j^{m-1} & J = J_2 \end{cases}$$

因为为分类器, 则 y_i 只能取 0 或 1, 当 $J = J_1$ 时上式如下

$$\frac{\partial J_1}{\partial w_{ij}^{m-1}} = -\frac{1}{n} (y_i - y'_i) y'_i (1 - y'_i) y_j^{m-1} = \begin{cases} -\frac{1}{n} y'_i (1 - y'_i)^2 y_j^{m-1} & y_i = 1 \\ -\frac{1}{n} y_i^2 (1 - y'_i) y_j^{m-1} & y_i = 0 \end{cases}$$

可以看到无论 $y_i = 0$ 或 1 上式都包含结构 $A^2(1 - A)$, $A \in [0, 1]$, 显然随着 A 的增大该结构先增后减, 这即反映出均方误差函数用于分类器时缺点: 当预测值 y'_i 非常接近 0 或 1 时误差函数的梯度将非常小, 即所谓的**梯度消失**, 从而降低学习效率, 而换用交叉熵则可以在不改变激励函数的情况下提升分类器训练效果。

23.9.7 概率分布间的距离——Wasserstein 距离

$$W[P(x_1), Q(x_2)] = \min_{\lambda} \int \int \lambda(x_1, x_2) d(x_1, x_2) dx_1 dx_2 \quad \int \lambda(x_1, x_2) dx_1 = Q(x_2) \quad \int \lambda(x_1, x_2) dx_2 = P(x_1)$$

1. ¹⁸Wasserstein 距离 (简称 W 距离) 又称 Kantorovich-Rubinstein 距离, 是一种度量两种概率分布相似性的方式。W 距离还有一个更形象的称呼为推土机 (Earth-Mover, EM) 距离, 因为 W 距离可以理解为将概率 $P(x)$ “搬到” 概率 $Q(x)$ 所需要的最小代价。与 KL 散度等常见的度量方式相比, W 距离具有更大的适用范围, 主要表现为其可以很方便地比较完全不同的两个概率分布;
2. 假设随机变量 X_1, X_2 分别服从 $P(x_1), Q(x_2)$ 的概率分布, 则 W 距离 $W[P(x_1), Q(x_2)]$ 如上所示。上式中 $\lambda(x_1, x_2)$ 为任意满足边缘概率分布为 $P(x_1), Q(x_2)$ 的二维联合概率分布。如果将概率分布理解为质量分布, 则 $\lambda(x_1, x_2)$ 表示从 $x = x_1$ 处移动至 $x = x_2$ 处的质量大小, 这也是推土机距离的由来;
3. x_1, x_2 为基于 $\lambda(x_1, x_2)$ 采样得到的样本 (可以是标量也可以是向量), 则 $d(x_1, x_2)$ 可理解为样本 x_1, x_2 的距离 (相似度), 一般地可由 l 范数计算, 如

$$d(x_1, x_2) = \sum_{i=1}^n |x_{1i} - x_{2i}| \quad or \quad d(x_1, x_2) = \sqrt{\sum_{i=1}^n |x_{1i} - x_{2i}|^2} \quad or \quad d(x_1, x_2) = \sum_{i=1}^n |x_{1i} - x_{2i}|^2$$

x_{1i}, x_{2i} 分别为向量 x_1, x_2 的元素;

¹⁷为什么均方差 (MSE) 不适合分类问题? 交叉熵 (cross-entropy) 不适合回归问题? https://blog.csdn.net/weixin_41888969/article/details/89450163

¹⁸从 Wasserstein 距离、对偶理论到 WGAN: <https://www.spaces.ac.cn/archives/6280>

4. 在工程中一般更关心 \mathbf{W} 距离的离散形式:

$$W[P(x_1), Q(x_2)] = \min_{\Lambda} \sum_i \sum_j \lambda(x_{1i}, x_{2j}) d(x_{1i}, x_{2j}) = \min_{\Lambda} \langle \Lambda, D \rangle$$

$\langle \Lambda, D \rangle$ 表示矩阵点积 (对应元素相乘), 矩阵 Λ, D 分别为 $\lambda(x_{1i}, x_{2j}), d(x_{1i}, x_{2j})$ 的元素集合;

5. 可以看到, 计算 \mathbf{W} 距离最大的难度在于最优 Λ 的计算。一般地此类问题可通过梯度下降算法求解, 但因为问题是线性规划问题, 梯度下降时收敛速度较慢, 而且结果不稳定, 易受样本影响;
6. 目前最主流的求解方法为 **Sinkhorn** 算法。算法在标准 \mathbf{W} 距离的基础上加上一熵正则项, 提高求解速度和稳定性¹⁹。定义 Λ 的熵正则项 $H(\Lambda)$

$$W[P(x_1), Q(x_2)] = \min_{\Lambda} [\langle \Lambda, D \rangle - \varepsilon H(\Lambda)] \quad H(\Lambda) = - \sum \Lambda(\ln \Lambda - 1)$$

讨论 $H(\Lambda)$ 加入的效果:

- 因为 $\Lambda \leq 1$, $\sum \Lambda = 1$, 有 $0 < H(\Lambda)$, 而且当且仅当所有 $\lambda(x_{1i}, x_{2j})$ 相等时 $-\varepsilon H(\Lambda)$ 最小。当 Λ 各元素相近时, 每一元素都很小, 因此熵正则项的加入起到鼓励多股小流量传输的效果;
- 另外, $H(\Lambda)$ 的引入使得目标函数为 ε - 强凸 (**strong convex**) 函数²⁰, 因而具有唯一解, 且具有较高的收敛速度;
- 再观察系数 ε . ε 越小, 计算的解就越接近标准 \mathbf{W} 距离, Λ 稀疏性较大, 少数元素取得较大值, 但此时计算稳定性也不高。

23.10 超球体积与超球坐标系

$$V_n = \frac{\pi^{\frac{n}{2}} R^n}{\Gamma(\frac{n}{2} + 1)}$$

$$\left\{ \begin{array}{l} x_1 = r \cos(\varphi_1) \\ x_2 = r \sin(\varphi_1) \cos(\varphi_2) \\ x_3 = r \sin(\varphi_1) \sin(\varphi_2) \cos(\varphi_3) \\ \dots \\ x_{n-1} = r \sin(\varphi_1) \cdots \sin(\varphi_{n-2}) \cos(\varphi_{n-1}) \\ x_n = r \sin(\varphi_1) \cdots \sin(\varphi_{n-2}) \sin(\varphi_{n-1}) \end{array} \right.$$

1. 假设超球体体积为 V_n , 根据积分定义其在直角坐标系下可表示为:

$$V_n = \underbrace{\int \cdots \int}_{x_1^2 + \cdots + x_n^2 \leq R^2} dx_1 \cdots dx_n$$

2. 最简单的积分方法是将直角坐标系下积分变换至超球坐标系下积分, 计算雅克比行列式可知两坐标系微分变换公式如下

$$dx_1 \cdots dx_n = r^{n-1} \sin^{n-2}(\varphi_1) \sin^{n-3}(\varphi_2) \cdots \sin(\varphi_{n-2}) dr d\varphi_1 d\varphi_2 \cdots d\varphi_{n-1}$$

继而积分表达式变换如下

$$V_n = \int_0^R r^{n-1} dr \int_0^\pi \sin^{n-2} \varphi_1 d\varphi_1 \int_0^\pi \sin^{n-3} \varphi_2 d\varphi_2 \cdots \int_0^\pi \sin \varphi_{n-2} d\varphi_{n-2} \int_0^{2\pi} d\varphi_{n-1}$$

¹⁹ 最优传输-熵正则化 (第八篇): [https://blog.csdn.net/Utterly_Bonkers/article/details/89546491?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommentFromBaidu-2.channel_param](https://blog.csdn.net/Utterly_Bonkers/article/details/89546491?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommentFromBaidu-2.channel_param&depth_1=utm_source=distribute.pc_relevant.none-task-blog-BlogCommentFromBaidu-2.channel_param)

²⁰ 对于一可微函数 $f(x)$, 若满足 $f(x) \geq f(x_0) + \nabla f(x_0)^T(x - x_0) + \frac{\mu}{2}(x - x_0)^2$, 则称函数为 u - 强凸 (u -strong convex)。与凸函数相比, 强凸函数在定义上多了一项二次项。强凸函数是保证很多基于梯度下降的算法的线形收敛速率的条件之一。

$$= \frac{2\pi R^n}{n} \int_0^\pi \sin^{n-2} \varphi d\varphi_1 \int_0^\pi \sin^{n-3} \varphi_2 d\varphi_2 \cdots \int_0^\pi \sin \varphi_{n-2} d\varphi_{n-2}$$

记 $I_i = \int_0^{\frac{\pi}{2}} \sin^i \varphi d\varphi$, 则上式进一步简化为

$$V_n = \frac{2^{n-1}\pi R^n}{n} \prod_{i=1}^{n-2} I_i$$

又因为

$$I_i = \begin{cases} \frac{(i-1)!!}{i!!} \cdot \frac{\pi}{2} & i = 2k \\ \frac{(i-1)!!}{i!!} & i = 2k-1 \end{cases}$$

当 n 为偶数时, 有 $\Gamma(\frac{n}{2} + 1) = (\frac{n}{2})!$, 此时

$$V_n = \frac{2^{n-1}\pi R^n}{n} \prod_{i=1}^{n-2} I_i = \frac{2^{n-1}\pi R^n}{n} \left(\prod_{k=1}^{\frac{n}{2}-1} I_{2k} \right) \left(\prod_{k=1}^{\frac{n}{2}-1} I_{2k-1} \right) = \frac{(2\pi)^{\frac{n}{2}} R^n}{n!!} = \frac{\pi^{\frac{n}{2}} R^n}{(\frac{n}{2})!} = \frac{\pi^{\frac{n}{2}} R^n}{\Gamma(\frac{n}{2} + 1)}$$

当 n 为奇数时, 有 $\Gamma(\frac{n}{2} + 1) = \sqrt{\pi} \frac{n!!}{2^{(n+1)/2}}$, 此时

$$V_n = \frac{2^{n-1}\pi R^n}{n} \prod_{i=1}^{n-2} I_i = \frac{2^{n-1}\pi R^n}{n} \left(\prod_{k=1}^{\frac{n-3}{2}} I_{2k} \right) \left(\prod_{k=1}^{\frac{n-1}{2}} I_{2k-1} \right) = \frac{2(2\pi)^{\frac{n-1}{2}} R^n}{n!!} = \frac{\pi^{\frac{n}{2}} R^n}{\Gamma(\frac{n}{2} + 1)}$$

3. 转换至超球坐标系积分本身难度不大, 除超球体积外也适合其它与超球有关的积分计算, 但需要计算雅克比行列式。除了以上方法也可在直角坐标系内直接计算超球体积。作线性缩放:

$$V_n = \underbrace{\int \cdots \int}_{x_1^2 + \cdots + x_n^2 \leq R^2} dx_1 \cdots dx_n \xrightarrow{x_i = Ru_i} R^n \underbrace{\int \cdots \int}_{u_1^2 + \cdots + u_n^2 \leq 1} du_1 \cdots du_n = R^n U_n$$

式中 U_n 为半径为 1 的 n 阶超球的体积, 显然只需计算 U_n , 进一步地逐项积分:

$$U_n = \underbrace{\int \cdots \int}_{u_1^2 + \cdots + u_n^2 \leq 1} du_1 \cdots du_n = \int_{-1}^1 du_n \underbrace{\int \cdots \int}_{u_1^2 + \cdots + u_{n-1}^2 \leq 1 - u_n^2} du_1 \cdots du_{n-1} \xrightarrow{u_i = \sqrt{1-u_n^2} \mu_i} \int_{-1}^1 (1-u_n^2)^{\frac{n-1}{2}} du_n \underbrace{\int \cdots \int}_{\mu_1^2 + \cdots + \mu_{n-1}^2 \leq 1} d\mu_1 \cdots d\mu_{n-1}$$

显然存在 U_n 递推式

$$U_n = U_{n-1} \int_{-1}^1 (1-u^2)^{\frac{n-1}{2}} du \xrightarrow{u = \sin \varphi} U_{n-1} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^n \varphi d\varphi = 2U_{n-1} \int_0^{\frac{\pi}{2}} \cos^n \varphi d\varphi = 2U_{n-1} I_n$$

递推同样可得超球体积公式。

23.11 图像特征提取

23.11.1 梯度方向直方图 (histogram of oriented gradient, HOG) 特征

1. 顾名思义, HOG 特征提取了图片像素点的梯度强度和方向信息, 并通过直方图统计相邻像素点的梯度信息起到信息稠密化的效果。HOG 特征描述了图片各区域的梯度强度和方向的分布情况, 能很好的提取物体的轮廓和纹理信息;
2. 为避免颜色和光照信息影响轮廓识别, 首先需将图片灰度化并标准化 Gamma 颜色空间。Gamma 标准化属于图像增强技术, 其本质是对图片的像素值作幂函数映射, 可矫正图片的亮度信息。假设灰度化后的图像像素值为 $I(x, y) \in [0, 1]$, 则 Gamma 标准化公式如下:

$$I(x, y) = I(x, y)^\gamma \quad (23.8)$$

上式一般有 $\gamma = 0.5$, 又因为 $I(x, y) \in [0, 1]$, 结合幂函数性质可以看到, Gamma 标准化可以提高阴影区域的分辨率同时压缩曝光区域的信息, 从而减少图片光照和阴影, 突出物体轮廓信息;

3. 定义 (x, y) 处像素点的梯度强度为 $G(x, y)$ 、梯度方向为 $\alpha(x, y) \in [-\pi/2, \pi/2]$, 则定义式如下:

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad \alpha(x, y) = \arctan\left(\frac{G_x(x, y)}{G_y(x, y)}\right) \quad (23.9)$$

$$G_x(x, y) = I(x+1, y) - I(x-1, y) \quad G_y(x, y) = I(x, y+1) - I(x, y-1)$$

4. 定义向量 $P(x, y)$ 同时表示像素梯度强度和方向信息。 $P(x, y)$ 的维度为 9, 对应于 $\alpha(x, y)$ 值域的均匀的 9 个子区间, 若 $\alpha(x, y)$ 落入区间 i , 则有 $P_i(x, y) = G(x, y)$, 否则则令 $P_i(x, y) = 0$ 。叠加多个像素点的 $P(x, y)$ 即为所谓的“梯度方向直方图 (HOG)”;

5. 最后介绍基于各像素点 $P(x, y)$ 向量得到图片 HOG 特征的方法:

- 以 8×8 个像素点为单位组成一个元胞 (cell), 叠加元胞 C 内的 64 个像素点的梯度方向向量 $P(x, y)$ 得到梯度方向直方图 $\sum_{(x,y) \in C} P(x, y)$, 此时即将 $8 \times 8 \times 2$ 的信息量压缩为 1×9 的向量 ($8 \times 8 \times 2$ 中 2 指梯度大小和方向 2 维信息);
- 以 2×2 个元胞为单位组成一个块 (block), 假设元胞间不重叠, 则一个块内包含 4 个直方图。将 4 个直方图合并为 1×36 的向量并对其作归一化, 称为 block 的 HOG 特征描述子。统计图像中所有 block 的 HOG 特征描述子即为最终的图像 HOG 特征。图像的 HOG 特征是一个高维的长向量。

23.11.2 Gabor 特征

1. Gabor 特征对于图像的边缘敏感, 能够提供良好的方向选择和尺度选择特性, 而且对于光照变化不敏感, 能够提供对光照变化良好的适应性, 因此 Gabor 特征广泛应用于视觉信息理解。Gabor 特征得名于 Gabor 变换, 是指对图像进行 Gabor 变换后再压缩得到的特征。Gabor 变换又称短时傅里叶变换或窗口傅立叶变换 (见 21.14 节), 由 D.Gabor 于 1946 年提出, 与小波变换并称为两类经典的时域-频域综合分析方法;
2. 对于一维信号 $f(t)$, 定义窗口函数 $g_a(t-b)$, b 为位置参数、 a 为尺度参数。一般地令 $g_a(t-b)$ 为高斯函数, 则对信号 $f(t)$ 对 Gabor 变换公式如下

$$G(a, b, \omega) = \int_{-\infty}^{\infty} f(t) g_a(t-b) e^{-i\omega t} dt \quad g_a(t-b) = \frac{1}{2\sqrt{\pi a}} e^{-\frac{t^2}{4a}}$$

上式中 $g_a(t-b)e^{-i\omega t}$ 定义为 Gabor 滤波器。可以看到, 以上变换实际上是对信号 $f(t)$ 局部平稳化 (加窗) 后再进行傅立叶变换。 a, b 决定了窗口函数的尺度特征和位置特征, 调整 a, b 即可调整提取的信号 $f(t)$ 的特征。若 $g_a(t-b)$ 为常数, 则 Gabor 变换退化为傅立叶变换;

3. 当应用于二维图像处理时, 窗口函数即为二维高斯函数, 而且此时不仅在两个维度有各自的位置参数和尺度参数, 还有旋转参数。另外在二维场景下, Gabor 滤波器中的正弦函数 $e^{-i\omega t}$ 也为二维空间正弦函数。二维 Gabor 滤波器定义式为

$$\begin{aligned} & \left[\frac{1}{ab} \exp \left\{ -\pi (a^2 x'^2 + b^2 y'^2) \right\} \right] \cdot \left[\exp \left\{ -2\pi i(u_0 x + v_0 y) \right\} \right] \\ & \begin{cases} x' = (x - x_0) \cos \theta + (y - y_0) \sin \theta \\ y' = -(x - x_0) \sin \theta + (y - y_0) \cos \theta \end{cases} \quad \theta = \arctan \frac{u_0}{v_0} \end{aligned} \quad (23.10)$$

4. 需要说明的是, 因为虚数的影响, 频域变换后会得到幅度谱和相位谱 (或实部和虚部) 两类信息。另外, 因为一次 Gabor 变换只能采集到图像的一类特征, 所以需要设置多组滤波器参数进行多次 Gabor 变换以提取图像不同尺度的特征, 同时多次变换也会得到多张 Gabor 特征图。为降低内存需对多种尺度多种方向下的 Gabor 特征进行融合, 包括方向融合和尺度融合两类:

- 前者融合 Gabor 特征在同一尺度上的方向信息 (使得同一尺度上的所有实部和虚部之比相同);
- 后者融合 Gabor 特征在不同尺度上的幅度信息 (使得所有 Gabor 特征具有相同的实部和虚部)。

最终得到与原图尺寸相等, 但体现多种尺度和方向特征的 Gabor 特征图;

5. 进行 Gabor 变换后得到的图像虽然保留了纹理信息, 但仍与原图具有相同的尺寸。为提升信息密度, 对图像分块 (block) 并统计每一块的能量 $\sum_{(x,y) \in B} \|Gabor(x, y)\|^2$ 。将图像中所有块的能量组合成向量, 即得到最终的 Gabor 特征。

23.11.3 尺度不变特征 (scale-invariant feature transform, SIFT)

1. SIFT 特征得名于 SIFT 算法, 由 David Lowe 于 1999 年最初发表, 并于 2004 年最终完善。算法将图像数据转换为相对于局部特征的尺度不变坐标, 即为 SIFT 特征。顾名思义, SIFT 特征对图片大小和旋转不敏感, 而且对光照、噪声等影响的抗击能力也非常优秀。相比于之前的图像特征提取算法, SIFT 在性能和适用范围方面较于之前的算法有着质的改变;
2. SIFT 算法的流程可分为尺度空间极值检测 (scale-space extrema detection)、关键点定位 (key-point localization)、方向匹配 (orientation assignment) 和关键点描述 (key-point description) 共四步。

尺度空间极值检测 (scale-space extrema detection)

1. 第一步中“尺度”的定义基本等同于 Gabor 变换的尺度参数, 尺度越大对应越宏观 (模糊) 的特征。“尺度空间极值”即为一定尺度空间内的极大值点和极小值点。这些点在相应尺度空间内较为突出, 是候选关键点。因此首先需构建尺度空间。算法基于高斯核函数 $G(x, y, \sigma)$ 对原图像 $I(x, y)$ 进行卷积变换得到相应尺度下的尺度图像 $L(x, y, \sigma)$

$$L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y) \quad G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} \quad (23.11)$$

σ 即为尺度参数, \otimes 表示卷积运算。选取等比的多组尺度参数得到多组尺度图像, 得到原图像的尺度空间;

2. 尺度的增大必然导致图像信息量的减小, 为降低储存空间, 将图像的尺度空间与图像金字塔 (image pyramid) 技术结合。以连续的 3 或 5 层尺度图像作为金字塔的一阶 (octave)。每一阶的最上一层尺度图像经过降采样 (downsampled, 或称下采样 (subsampling), 或称池化 (pooling)) 将图像尺寸缩减一半得到下一阶的最下层尺度图像的尺寸;
3. 为保证整个金字塔范围内的尺度连续, 记 s 为一阶中的层数, $k = 1/s$ 为尺度变化因子, 则第 n 阶的第 m 层尺度图像的尺度应为 $nk^{m-1}\sigma$ 。此时无论相邻两层尺度图像位于同一阶或是相邻两阶, 两者的尺度差均为 $k\sigma$;
4. 在金字塔所代表的尺度空间中检测极值点, 可类比 HOG 特征以尺度梯度极值表示。沿梯度轴对金字塔同一阶内的相邻两层尺度图像作差分, 得到高斯差分尺度图像 $D(x, y, \sigma)$ 组成的高斯差分尺度空间 (DOG scale-space)

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] \otimes I(x, y) \quad (23.12)$$

对于 DOG 图像中的任意一点, 包括上下相邻两层 DOG 图像, 其临域内共有 $8 + 9 + 9 = 26$ 个点。若其是该临域内的极大值或极小值, 则该点即为尺度空间极值点, 也是候选的关键点.

关键点定位 (key-point localization)

1. 算法的第二步即是剔除候选关键点中的干扰点, 得到最终的关键点。算法考虑两类干扰点, 第一类为噪声点, 第二类为边界点²¹——在一个方向 DOG 剧烈变化而在正交方向变化和缓;
2. 噪声点在 DOG 空间内的幅值一般较小, 可通过预设阈值 (设为 0.03) 直接去除。需要注意的是, 因为 DOG 空间为离散空间, 所得到的候选关键点 $D(z^*, y^*, \sigma)$ 未必就是领域内的极值点。为得到更精确的数值解, 对函数 $D(x, y, \sigma)$ 于所有关键点 $z^* = (x^*, y^*)$ 处进行二阶泰勒展开, 求一阶导得到更精确的极值点 \hat{z}^* 和极值 $D(\hat{z}^*, \sigma)$, 并滤除 $|D(\hat{z}^*, \sigma)| < 0.03$ 的点。

$$\begin{aligned} D(z, \sigma) \approx D(z^*, \sigma) + \frac{\partial D}{\partial z} \Big|_{z^*} z + \frac{1}{2} z^T \frac{\partial^2 D}{\partial z^2} \Big|_{z^*} z \Rightarrow \hat{z}^* &= -\frac{\partial^2 D}{\partial z^2} \Big|_{z^*} \cdot \frac{\partial D}{\partial z} \Big|_{z^*} \\ \Rightarrow D(\hat{z}^*, \sigma) &\approx D(z^*, \sigma) + \frac{1}{2} \frac{\partial D}{\partial z} \Big|_{z^*} \hat{z}^* \end{aligned} \quad (23.13)$$

3. 对于第二类干扰点, 曲线变化的剧烈程度可通过曲率表征, 理想的关键点应该在任意方向都具有相近的曲率。但在二维曲面内寻找最大和最小的曲率较为复杂。注意到二维曲面某点的 Hessian 矩阵实际上刻画

²¹在 SIFT 算法中边界点属于干扰点, 在理想情况下关键点应该为角点或者端点。以四边形为例, 其关键点即为其四个顶点, 无论如何变换该图形, 仍可由四个顶点描述四边形的特征, 而边界上的点对临域特征的贡献其实不大。

了其临域的变化趋势。Hessian 矩阵的特征值描述了该点在特征方向上的凹凸性，特征值越大凸性越强，越接近零则越平缓，两个特征向量则为该点临域内凹凸性最强的两个方向。因此以差分 Hessian 矩阵 H 的两个特征值替代两个曲率，记两个特征值为 $\alpha \geq \beta$ ，并有 $\alpha = \gamma\beta$ 。 H 的迹 $\text{Tr}(H)$ 和行列式 $\text{Det}(H)$ 与两个特征值 α, β 具有如下关系

$$\begin{cases} \text{Tr}(H) = \alpha + \beta = (1 + \gamma)\alpha \\ \text{Det}(H) = \alpha\beta = \gamma\alpha^2 \end{cases} \implies \frac{\text{Tr}^2(H)}{\text{Det}(H)} = \frac{(1 + \gamma)^2}{\gamma} \quad \gamma \geq 1 \quad (23.14)$$

4. 因为候选关键点 (x^*, y^*) 为临域 DOG 空间内的极值，不可能出现 $\gamma < 0$ 的情况，而在 $\gamma \geq 1$ 的情况下 $\frac{\text{Tr}^2(H)}{\text{Det}(H)}$ 单调递增，因此欲滤除 $\gamma > \gamma_0$ 的候选关键点可等价于滤除 $\frac{\text{Tr}^2(H)}{\text{Det}(H)} > \frac{(1 + \gamma_0)^2}{\gamma_0}$ 的点（设 $\gamma_0 = 10$ ）。

方向匹配 (orientation assignment)

1. 为保证算法得到的 SIFT 特征具有尺度和方向不变性（对图像大小和旋转不敏感），在算法的第三步中需计算得到的关键点的方向。该步骤较为简单；
2. 对于关键点 (x^*, y^*, σ) ，其方向 θ 由相应尺度图像 $L(x, y, \sigma)$ 内的临域其它点的梯度方向和梯度强度共同决定。对于任意坐标点 (x, y, σ) ，其梯度方向 $\theta(x, y, \sigma)$ 和梯度强度 $m(x, y, \sigma)$ 如下

$$\theta(x, y, \sigma) = \arctan \frac{L(x, y + 1, \sigma) - L(x, y - 1, \sigma)}{L(x + 1, y, \sigma) - L(x - 1, y, \sigma)} \quad \theta \in [0, 2\pi] \quad (23.15)$$

$$m(x, y, \sigma) = \sqrt{(L(x, y + 1, \sigma) - L(x, y - 1, \sigma))^2 + (L(x + 1, y, \sigma) - L(x - 1, y, \sigma))^2}$$

3. 类似于 HOG 特征，计算关键点临域内所有点的梯度方向和强度后统计梯度方向直方图，不同之处在于此时梯度方向的取值范围为 $[0, 2\pi]$ ，并进行 36 等分。梯度方向直方图中峰值最大的方向即为关键点的方向 θ 。如果同时存在多个最大方向，则设置多个关键点，每个关键点具有相同的位置和尺度，但方向不同；
4. 临域内不同点的梯度方向和强度对最终关键点的方向应该具有不同的贡献，距关键点越远的点贡献越小，因此在统计梯度方向直方图时可进行高斯函数加权，对每一点的梯度强度乘上相应的高斯权重。高斯权重函数的尺度 $\sigma_0 = 1.5\sigma$ ；
5. 最后介绍临域范围确定方法。因为采用高斯函数加权，由 3σ 原则，距关键点距离大于 $3\sigma_0$ 的点对于关键点的方向的贡献可忽略，因此设临域半径 $r = 3\sigma_0$ ，临域内共 $4r^2$ 个样本点。

关键点描述 (key-point description)

1. 算法的最后一步为描述关键点临域内的局部特征，并保证局部特征具有方向不变性，得到最终的 SIFT 特征；
2. 该步骤同样类似于 HOG 算法。首先同样是确定关键点的临域半径 $r = 8$ ，得到关键点在相应尺度图像上的临域²²。对临域进一步划分为 $d \times d$ 的子区域（推荐 $d = 4$ ），对每个子区域统计起梯度方向直方图，直方图的长度设为 l （原文中 $l = 8$ ）。将一个临域内所有子区域的梯度方向直方图合并即为该关键点的局部特征向量，记为 \mathbf{h} ，长度为 $(2r)^2/d^2 \times l = 128$ 。为去除光照的影响，将特征向量 \mathbf{h} 归一化，得到最终的 SIFT 特征。下式中 $\mathbb{D}[\mathbf{h}]$ 表示方差

$$\mathbf{h} = \frac{\mathbf{h}}{\sqrt{\mathbb{D}[\mathbf{h}]}} \quad (23.16)$$

特征向量 \mathbf{h} 包含了临域像素的梯度方向信息。若将图像旋转一定角度，梯度方向也会改变相应角度，进而改变 SIFT 特征，因此此时特征是不满足方向不变性的；

3. 第三步得到的关键点方向 θ 在描述方向信息的同时也提供了相应尺度下关键点临域的基准方向，在组合特征之前，应首先将尺度图像旋转 $-\theta$ 角，使得关键点的方向与图像 x 轴方向一致。此后再进行组合得到的 SIFT 特征即可满足方向不变性。需要说明的是，旋转后的像素点的坐标很可能不位于临域内的网格点处，此时需基于双线性插值得到网格点的值。

²²注意此时的临域大小不等于上一步方向匹配中的临域大小。

23.12 反函数的数值计算

1. 对于定义在 $x \in C$ 上的函数 $y = f(x)$, 若存在函数 $g(y)$ 满足 $x = g(y), y \in C$, 则称 $g(y)$ 为 $f(x)$ 的反函数 (**inverse function**), 可记为 $x = g(y) = f^{-1}(y)$, 而 $f(x)$ 则称为原函数或直接函数。原函数与反函数的形状关于 $y = x$ 对称, 并且两者的导数具有如下关系:

$$\frac{dy}{dx} = \frac{1}{\frac{dx}{dy}}$$

2. 借助原函数和反函数可以得到两个相关变量的双向映射关系, 然而在很多时候往往无法同时得到两个函数的解析形式, 而且大量的函数不存在严格意义上的反函数 (大部分偶函数不存在反函数)。然而在实际工作中往往需要对一定范围内的映射关系求反函数以实现反向搜索, 因此便需要通过数值计算方法确定原函数的反函数;



级数反演法 (series revision)

级数反演又称为序列反演, 指一对序列或级数可以相互表示的互反关系, 联系这种互反关系的公式称为反演公式。级数反演的定义类似于反函数, 具体地, 若两个级数或序列 $g(n), f(n)$ 满足以下关系, 则称以下两式为反演公式

$$g(n) = \sum_{r=0}^n c_{n,r} f(r) \iff f(n) = \sum_{r=0}^n d_{n,r} g(r), \quad n \in N$$

上式等价于系数矩阵 $C = \{c_{n,r}\}$ 与 $D = \{d_{n,r}\}$ 互逆。对于任意高阶多项式

$$y = a_1 x + a_2 x^2 + a_3 x^3 + \dots$$

则存在以下相应系数 A_1, A_2, A_3, \dots 使得反演关系成立^a

$$x = A_1 y + A_2 y^2 + A_3 y^3 + \dots$$

$$A_1 = a_1^{-1}$$

$$A_2 = -a_1^{-3} a_2$$

$$A_3 = a_1^{-5} (2a_2^2 - a_1 a_3)$$

$$A_4 = a_1^{-7} (5a_1 a_2 a_3 - a_1^2 a_4 - 5a_2^3)$$

$$A_5 = a_1^{-9} (6a_1^2 a_2 a_4 + 3a_1^2 a_3^2 + 14a_2^4 - a_1^3 a_5 - 21a_1 a_2^2 a_3)$$

$$A_6 = a_1^{-11} (7a_1^3 a_2 a_5 + 7a_1^3 a_3 a_4 + 84a_1 a_2^3 a_3 - a_1^4 a_6 - 28a_1^2 a_2 a_3^2 - 42a_2^5 - 28a_1^2 a_2^2 a_4)$$

$$A_7 = a_1^{-13} (8a_1^4 a_2 a_6 + 8a_1^4 a_3 a_5 + 4a_1^4 a_2^4 + 120a_1^2 a_2^3 a_4 + 180a_1^2 a_2^2 a_3^2 + 132a_2^6 - a_1^5 a_7 - 36a_1^3 a_2 a_5$$

$$- 72a_1^3 a_2 a_3 a_4 - 12a_1^3 a_3^3 - 330a_1 a_2^4 a_3)$$

进一步地, 对任意 n , 存在以下系数 A_n 的通项公式

$$A_n = \frac{1}{na_1^n} \sum_{s,t,u,\dots} (-1)^{s+t+u+\dots} \frac{n(n+1)\cdots(n-1+s+t+u+\dots)}{s!t!u!\dots} \left(\frac{a_2}{a_1}\right)^s \left(\frac{a_3}{a_1}\right)^t \cdots$$

上式中参数 s, t, u, \dots 满足 $s + 2t + 3u + \dots = n - 1$, 可以通过递推依次计算 s, t, u, \dots 。

^aSeries Reversion: <https://mathworld.wolfram.com/SeriesReversion.html>

3. 基于上述所提的级数反演法 (**series revision**), 可以给出数值计算任意可微函数 $f(x), x \in [a, b]$ 的反函数的一个可行思路²³——基于泰勒展开将 $f(x)$ 展开为多项式形式, 基于多项式级数的反演公式即可写出 $f(x)$ 的反函数 $g(y)$)

$$y = f(x) \approx f(c) + f'(c)(x - c) + \frac{1}{2}f''(c)(x - c)^2 + \frac{1}{6}f'''(c)(x - c)^3 + O((x - c)^4), \quad c = \frac{a+b}{2}$$

²³Numerical algorithm for finding the inverse of a function: <https://math.stackexchange.com/questions/3693157/numerical-algorithm-for-finding-the-inverse-of-a-function>

$$\Rightarrow x = g(y) \approx c + \frac{1}{f'(c)}(y - f(c)) - \frac{f''(c)}{2f'(c)^3}(y - f(c))^2 + \frac{3f''(c)^2 - f'''(c)f'(c)}{6f'(c)^5}(y - f(c))^3 + O((y - f(c))^4)$$

23.13 瑞利商 (*Rayleigh quotient*) 与广义瑞利商 (*generalized Rayleigh quotient*)

1. 瑞利商 (*Rayleigh quotient*) 是一种算子，在矩阵论、统计学及多种机器学习算法中被广泛应用。记 n 维对称阵 $A \in \mathbb{R}^{n \times n}$ 和 n 维向量 $x \in \mathbb{R}^{n \times 1}$ ，则瑞利商 $R(A, x)$ 定义为

$$R(A, x) = \frac{x^\top Ax}{x^\top x}$$

由定义易知，瑞利商满足 $R(A, x) = R(A, cx)$ ，其中 c 为标量；

2. 记对称阵 A 的最大特征值和最小特征值分别为 $\lambda_{\min}, \lambda_{\max}$ ，则在 A 固定的情况下瑞利商 $R(A, x)$ 具有如下重要性质

$$\lambda_{\min} \leq R(A, x) \leq \lambda_{\max} \iff \min_x R(A, x) = \lambda_{\min}, \quad \max_x R(A, x) = \lambda_{\max}$$

而上述对 $R(A, x)$ 求最大或最小值的优化问题又被称为瑞利商问题；

3. 证明上述瑞利商性质的方法有多种，以下介绍基于运筹优化的方法。以证明 $R(A, x) \geq \lambda_{\min}$ 为例，显然只需要解 $\min_x R(A, x)$ ，不妨构建如下优化问题

$$\min_x x^\top Ax, \quad \text{s.t. } x^\top x = 1$$

上式利用了瑞利商 $R(A, x) = R(A, cx)$ 的性质，故假设 x 为单位向量并不会影响优化的结果。基于拉格朗日乘子法求解上述问题，构造拉格朗日函数有

$$L(x, \lambda) = x^\top Ax + \lambda(1 - x^\top x) \implies \begin{cases} \frac{\partial L}{\partial x} = 2Ax - 2\lambda x = 0 \\ \frac{\partial L}{\partial \lambda} = 1 - x^\top x = 0 \end{cases} \implies \begin{cases} Ax = \lambda x \\ x^\top x = 1 \end{cases}$$

上述 KKT 条件表明，对称阵 A 的每一个特征向量 x' 均为瑞利商 $R(A, x)$ 的驻点，且对应的 $R(A, x')$ 为相应的特征值 λ' 。而为求解 $R(A, x)$ 的最小值，将上述 KKT 条件代入最初构建的优化问题，则问题变为

$$\min_x J = x^\top Ax = x^\top \lambda x = \lambda, \quad \text{s.t. } x^\top x = 1, \quad Ax = \lambda x$$

由此易知 $R(A, x) \geq \lambda_{\min}$ ，当且仅当 x^* 为 λ_{\min} 对应的特征向量时等号成立。同理可证 $R(A, x) \leq \lambda_{\max}$ 并得到等号成立条件；

4. 在瑞利商的基础上还可进一步定义广义瑞利商 (*generalized Rayleigh quotient*)。记 n 维对称阵 $A, B \in \mathbb{R}^{n \times n}$ 和 n 维向量 $x \in \mathbb{R}^{n \times 1}$ ，则广义瑞利商 $R(A, B, x)$ 定义为

$$R(A, B, x) = \frac{x^\top Ax}{x^\top Bx}$$

同样地广义瑞利商满足 $R(A, B, cx) = R(A, B, x)$ ，其中 c 为标量。并且令向量 $y = B^{\frac{1}{2}}x$ ，则 $R(A, B, x)$ 显然有

$$R(A, B, x) = \frac{x^\top Ax}{x^\top Bx} = \frac{(B^{-\frac{1}{2}}y)^\top A (B^{-\frac{1}{2}}y)}{(B^{-\frac{1}{2}}y)^\top B (B^{-\frac{1}{2}}y)} = \frac{y^\top B^{-\frac{1}{2}}AB^{\frac{1}{2}}y}{y^\top y} = R(B^{-\frac{1}{2}}AB^{\frac{1}{2}}, y), \quad y = B^{\frac{1}{2}}x$$

由此易知广义瑞利商 $R(A, B, x)$ 的上下限及不等号取等条件。

23.14 优劣解距离法 (*Technique of order preference by similarity to an ideal solution, TOPSIS*)

1. TOPSIS 算法国内一般称为优劣解距离法或理想解法，由 C.L.Hwang 和 K.Yoon 于 1981 年首次提出，是一种经典的多指标方案综合比选方法²⁴；

²⁴TOPSIS 法 (优劣解距离法) 介绍及 python3 实现: <https://zhuanlan.zhihu.com/p/37738503>

23.14. 优劣解距离法 (TECHNIQUE OF ORDER PREFERENCE BY SIMILARITY TO AN IDEAL SOLUTION, TOPSIS)

2. 如何比选包含多维评价指标的多个方案是一个经典且被广泛研究的问题。为进行方案比选必然要求基于构造一个综合表征多个维度指标的单一维度评分，而最直接的思路是基于给定权重将多维评价指标加权聚合，由此衍生了如专家打分、层次分析法等一系列确定权重的方法；
3. **TOPSIS 算法提供了另一种确定评分的方法——首先基于样本构建理论最优与最差方案，再以各方案距理论最优和最差方案的距离作为评分。**记原始数据集为 $X = [x_{ij}]$ ，其中任意元素 x_{ij} 表示方案 i 第 j 维指标的评分。以下具体介绍 TOPSIS 算法的具体计算流程：
4. 首先是对原始数据集 X 进行预处理，具体包括两个步骤——指标属性同向化与归一化。指标属性同向化的目的是使得各维度指标的数值大小关系具有相同的意义。一般选择正向化，要求所有指标都是取值越大越好。为此考虑原始指标可能的以下四种类型：
- **极大型（效益型）指标：** 取值越大越好，无需额外正向化 $x'_{ij} = x_{ij}$ ；
 - **极小型（成本型）指标：** 取值越小越好，其正向化多采用下式

$$x'_{ij} = M_j - x_{ij} \quad \text{or} \quad x'_{ij} = \frac{1}{x_{ij}} \quad (x_{ij} > 0)$$

上式中 M_j 表示指标 j 可能的最大取值；

- **中间型指标：** 取值越接近中间的某个值越好，如水体 PH 值，其正向化多采用下式

$$x'_{ij} = \begin{cases} \frac{2(x_{ij} - m_j)}{M_j - m_j}, & m_j \leq x_{ij} \leq \frac{M_j + m_j}{2} \\ \frac{2(M_j - x_{ij})}{M_j - m_j}, & \frac{M_j + m_j}{2} \leq x_{ij} \leq M_j \end{cases}$$

上式中 m_j 表示指标 j 可能的最小取值；

- **区间型指标：** 取值位于特定区间越好，如体温，其正向化多采用下式

$$x'_{ij} = \begin{cases} 1 - \frac{a_j - x_{ij}}{a_j - m_j}, & x_{ij} < a_j \\ 1, & a_j \leq x_{ij} \leq b_j \\ 1 - \frac{x_{ij} - b_j}{M_j - b_j}, & x_{ij} > b_j \end{cases}$$

上式中 $[a_j, b_j]$ 表示指标 j 的最优区间。

对同向化后的数据集 $X' = [x'_{ij}]$ 进行归一化得到矩阵 $Z = [z_{ij}]$, $z_{ij} = \frac{x'_{ij}}{\sqrt{\sum_i (x'_{ij})^2}}$ ，数据预处理流程完成；

5. 算法的第二步是确定理论最优与最差方案，分别记为 $Z^+ = (z_1^+, \dots, z_m^+)^T$ 和 $Z^- = (z_1^-, \dots, z_m^-)^T$ ，其中元素 z_j^+, z_j^- 分别表示最优与最差方案中指标 j 的取值。在完成同向化后 Z^+, Z^- 可按下式易得

$$z_j^+ = \max_i z_{ij}, \quad z_j^- = \min_i z_{ij}$$

6. 最后即是基于各方案距最优与最差方案的距离确定方案间的优劣顺序。首先基于任意距离度量函数计算各方案距最优与最差方案的距离，以欧氏距离为例

$$D_i^+ = \sqrt{\sum_j \omega_j (z_j^+ - z_{ij})^2}, \quad D_i^- = \sqrt{\sum_j \omega_j (z_j^- - z_{ij})^2}$$

上式中 D_i^+, D_i^- 分别表示方案 i 距最优与最差方案的距离； ω_j 则表示距离计算时指标 j 的权重，由研究者自行确定。基于 D_i^+, D_i^- 即可得到方案 i 与最优方案的贴近程度 C_i 以作为方案 i 的综合评分

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-}, \quad C_i \in [0, 1]$$

显然 C_i 越接近 1 表示方案 i 距理论最优方案越近，综合评分也越高。

23.15 模糊数学 (**Fuzzy mathematics**) 基础——模糊集与模糊数

- 模糊数学 (fuzzy mathematics) 是现代数学中一个非常年轻的分支, 由查德 (L. A. Zadeh) 教授于 1965 年建立。与概率论类似, 模糊数学同样关注数学中的不确定性 (uncertainty) 现象。但不同于概率论关注事件发生的可能性 (probability), 模糊数学讨论的“模糊 (ambiguity)” 现象是指事件发生的程度。因此在模糊数学提出后, 现代数学即可大体划分为三个流派——研究精确与确定性现象的传统数学, 研究随机现象的概率论, 和研究模糊现象的模糊数学;
- 需要指出的是, 模糊数学自提出后一直面临着大量争议, 大量概率学者不认可模糊数学理论的合理性, 在上世纪下半页双方针对此展开了数十年的论战。例如 Lindley 强调“只有概率才是对不确定性的合理描述”²⁵。笔者认为上述争议的原因大体包括以下几方面:
 - 首先模糊数学是一门非常年轻的理论体系, 至今其理论仍未完善, 而当时同样研究不确定性的概率论已经具备完备的理论基础和广泛的学界认可;
 - 同时模糊数学理论的野心极大, 对概率论乃至现代数学的理论基础集合论都提出了相当程度的否定;
 - 另外上世纪下半页正好处于第三次数学危机结束, 理论数学基本完善, 同时第三次工业革命如火如荼, 以运筹学、控制论、人工智能为代表的应用数学快速发展的时期。模糊数学的提出挑战了概率论于不确定性数学的应用生态。

如今模糊数学的应用领域包括模糊聚类、模糊模式识别、模糊决策、模糊线性规划等, 其中以模糊综合评价应用最为广泛。而在如今最主要的应用数学领域——人工智能与机器学习则是概率模型占主导地位。

23.15.1 模糊集 (fuzzy set) 及其运算性质

- 模糊数学建立的标志是 1965 年模糊集 (fuzzy set) 概念的提出。集合论是包括概率论在内的现代数据理论的基础, 也是第三次数学危机的导火索。而 Zadeh 直接扩展了经典集合的定义, 提出模糊集的概念以作为模糊数学的理论基础, 由此反映了模糊数学理论的野心。模糊集的定义如下: 对于论域 U 内的任意一个元素 x , 若存在映射 $A(\cdot)$ 使得 $A(x) \in [0, 1]$, 则称 A (也可记为 $A(U)$) 为 U 上的模糊集, 同时 $A(\cdot)$ (也可记为 $\mu_A(\cdot)$) 称为 x 对 A 的隶属度函数 (membership function)。隶属度 $A(x)$ 越接近 1 表明 x 属于模糊集 A 的程度越大 (注意是“程度”不是“可能性”), 使得 $A(x) = 0.5$ 的点 x 称为过渡点, 此处模糊性最强^{26,27};
- 按照模糊集的定义, 论域 U 内的任意元素 x 均在不同程度上隶属于模糊集 A , 因此无法用传统的集合符号表示模糊集。模糊集的表示方法并不唯一, Zadeh 表示法最为常用。记 $U = \{u_1, u_2, \dots, u_n\}$ 为有限集, 则 U 上的模糊集 $A(U)$ 按 Zadeh 表示法记为

$$A(U) = \sum_{i=1}^n \frac{A(u_i)}{u_i} = \frac{A(u_1)}{u_1} + \frac{A(u_2)}{u_2} + \dots + \frac{A(u_n)}{u_n}$$

上式中 $\sum, +$ 不表示加法, 分式也不表示除法, 仅作为记录模糊集中各元素及其对应隶属度的符号。类似的, 若 U 为无限集, 则模糊集 $A(U)$ 按 Zadeh 表示法记为

$$A(U) = \int_{u \in U} \frac{A(u)}{u}$$

同理上式中 \int 也不表示积分, 只是一种标记符号;

- 模糊集所支持的运算与经典集合基本相同, 其运算结果同样为模糊集。模糊集 A, B, C 有基本运算:
 - 包含运算:** 若对 $\forall x \in U$ 均有 $A(x) \leq B(x)$, 则称 $A \subseteq B$;
 - 等于运算:** 若对 $\forall x \in U$ 均有 $A(x) = B(x)$, 则称 $A = B$;
 - 补集运算:** 若对 $\forall x \in U$ 均有 $A(x) = 1 - B(x)$, 则称 $A = B^c$;

²⁵“Only probability is a sensible description of uncertainty”, 出自 Lindley, Dennis V. “Scoring Rules and the Inevitability of Probability.” *International Statistical Review / Revue Internationale de Statistique*, vol. 50, no. 1, 1982, pp. 1-11. JSTOR, <https://doi.org/10.2307/1402448>.

²⁶模糊数学 (Fuzzy Mathematics) 理论基础: https://blog.csdn.net/weixin_60737527/article/details/125726451

²⁷模糊数学基础及应用-2 (1) -模糊集合: https://zhuanlan.zhihu.com/p/627041613?utm_id=0

- **交集运算:** 若对 $\forall x \in U$ 均有 $C(x) = A(x) \wedge B(x)$, 则称 $C = A \cap B$, 其中 $a \wedge b = \min\{a, b\}$;
- **并集运算:** 若对 $\forall x \in U$ 均有 $C(x) = A(x) \vee B(x)$, 则称 $C = A \cup B$, 其中 $a \vee b = \max\{a, b\}$;
- **笛卡尔积:** 若对 $\forall x_1, \dots, x_n \in U$ 均有 $B(x_1, \dots, x_n) = \bigwedge_i A_i(x_i)$, 则称 $B = A_1 \times \dots \times A_n = \prod_i A_i$ 。

与此同时, 模糊集运算时也支持如下运算律:

- **交换律:** $A \cap B = B \cap A$, $A \cup B = B \cup A$;
- **结合律:** $(A \cap B) \cap C = A \cap (B \cap C)$, $(A \cup B) \cup C = A \cup (B \cup C)$;
- **分配律:** $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$, $(A \cup B) \cap C = (A \cap c) \cup (B \cap C)$;
- **对偶律:** $(A \cap B)^c = A^c \cup B^c$, $(A \cup B)^c = A^c \cap B^c$;
- **还原律:** $(A^c)^c = A$;
- **0-1律:** $A \cup U = U$, $A \cap U = A$, $A \cup \emptyset = A$, $A \cap \emptyset = \emptyset$ 。

上述模糊集支持的运算律与经典集合的运算律几乎完全相同, **唯一不同在于模糊集不支持排中律, 即 $A \cup A^c \neq U$, $A \cap A^c \neq \emptyset$** 。这一根本性的松弛使得模糊数学方法在理论上具有更强的不确定性建模能力。因为概率论同样建立在经典集合论的基础上, 满足 $P(A \cap A^c) = P(\emptyset) = 0$, 而在模糊数学视角下, 则可以实现 $P(A \cap A^c) \geq 0$;

4. 虽然模糊集合能够反映现实中存在的模糊概念, 但在处理实际问题的过程中, 特别是最后决策时, 往往又需要将模糊集合化成各种不同的普通集合, 这就要求模糊集和普通集能够相互转化。为此介绍模糊集的 λ 截集 (λ -cut) 的概念。对于论域 U 上的任意模糊集 A , 记 $\lambda \in [0, 1]$, 称集合 $A_\lambda = \{x | x \in U, A(x) \geq \lambda\}$ 为 A 的 λ 截集。 λ 截集为经典集合, 也可视为隶属度函数仅为 0 和 1 的模糊集, 其运算满足如下性质:

- 对 U 上的两个模糊集 A, B , 有 $(A \cup B)_\lambda = A_\lambda \cup B_\lambda$, $(A \cap B)_\lambda = A_\lambda \cap B_\lambda$, 证明如下

$$\begin{aligned} x \in (A \cup B)_\lambda &\iff (A \cup B)(x) = A(x) \vee B(x) \geq \lambda \iff x \in A_\lambda \text{ or } x \in B_\lambda \iff x \in A_\lambda \cup B_\lambda \\ x \in (A \cap B)_\lambda &\iff (A \cap B)(x) = A(x) \wedge B(x) \geq \lambda \iff x \in A_\lambda \text{ and } x \in B_\lambda \iff x \in A_\lambda \cap B_\lambda \end{aligned}$$

- 对 U 上的多个模糊集 $\{A^{(i)} | i = 1, \dots, n\}$, 有 $(\bigcup_i A^{(i)})_\lambda \supseteq \bigcup_i (A^{(i)})_\lambda$, $(\bigcap_i A^{(i)})_\lambda = \bigcap_i (A^{(i)})_\lambda$, 先证后式

$$x \in \bigcap_i (A^{(i)})_\lambda \iff \forall i, A^{(i)}(x) \geq \lambda \iff \left(\bigcap_i A^{(i)} \right)(x) = \bigwedge_i A^{(i)}(x) \geq \lambda \iff x \in \left(\bigcap_i A^{(i)} \right)_\lambda$$

再证前式

$$x \in \bigcup_i (A^{(i)})_\lambda \iff \exists j, A^{(j)}(x) \geq \lambda \iff \left(\bigcup_i A^{(i)} \right)(x) = \bigvee_i A^{(i)}(x) \geq \lambda \iff x \in \left(\bigcup_i A^{(i)} \right)_\lambda$$

上式说明对于 $\forall x \in \bigcup_i (A^{(i)})_\lambda$ 均有 $x \in (\bigcup_i A^{(i)})_\lambda$, 但反之并不成立, 根源在于当 $n \rightarrow \infty$ 时 $(\bigcup_i A^{(i)})(x) \geq \lambda$ 并不一定能推出 $\exists j, A^{(j)}(x) \geq \lambda$ 。举如下反例说明。令 $A^i(x) = \frac{i-1}{i}$, 则显然有 $(\bigcup_i^\infty A^{(i)}) = U$, 令 $\lambda = 1$, 则 $(\bigcup_i^\infty A^{(i)})_\lambda = U$, $\bigcup_i^\infty (A^{(i)})_\lambda = \emptyset$, 因此只能说 $(\bigcup_i A^{(i)})_\lambda \supseteq \bigcup_i (A^{(i)})_\lambda$ 而非 $(\bigcup_i A^{(i)})_\lambda = \bigcup_i (A^{(i)})_\lambda$ 。

之所以基于截集沟通模糊数学与确定性数学, 在于**截集反映了模糊集某一水平切面的性质**, 而根据微积分思想, 若已知模糊集的所有截集则可以反求模糊集本身。这一特征被称为**分解定理 (decomposition theorem)**, 为模糊数学的核心定理之一

$$A = \bigcup_{\lambda \in [0, 1]} \lambda A_\lambda \quad \text{or} \quad A(x) = \bigvee_{\lambda \in [0, 1]} \lambda \wedge A_\lambda(x) = \bigvee \{\lambda \in [0, 1] | x \in A_\lambda\} \quad (\text{分解定理})$$

上式中 λA_λ , $\lambda \wedge A_\lambda(x)$ 即是将 A_λ 视为模糊集, 其隶属度函数仅取 0 或 1;

5. 上文先后介绍了模糊集的定义、运算律和与经典集合的联系, 最后介绍**模糊集的映射法则**, 借此可将经典集合论的所有运算推广至模糊集合。对针对模糊集的映射 f , 记 $B = f(A)$, 则模糊数 B 与 A 的隶属度函数之间存在如下关系

$$B(y) = \bigvee_{y=f(x)} A(x) \implies f^{-1}(B)(x) = B(f(x)) \quad (\text{扩张原理})$$

上式被称为扩张原理 (**extension principle**)。其意义非常清晰——若 f 为单映射 (即映射后 y, x 一一对应)，则映射不会改变元素 x 的隶属度，仅将其位置映射至 y ；若存在多个 $\{x_1, \dots, x_n\}$ 与 y 对应，则映射后 y 处的隶属度显然应该由 $\{x_1, \dots, x_n\}$ 中的最大隶属度决定。当映射 f 包含多个自变量时 (即 $B = f(A_1, \dots, A_n)$)，扩张原理进一步拓展为多元扩张原理

$$B(y) = \bigvee_{y=f(x_1, \dots, x_n)} \left[\bigwedge_{x_i} A(x_i) \right] \quad (\text{多元扩张原理})$$

注意到多元映射本质上即是多个集合的笛卡尔积构成的集合的一元映射，故多元扩张原理可由扩张原理与笛卡尔积公式导出。

23.15.2 模糊数 (fuzzy number) 及其运算性质

1. 在模糊集的基础上进一步延伸出了模糊数 (fuzzy number) 的概念²⁸。模糊数是经典的“数”的概念的延伸，是一种特殊的模糊集。设 A 为定义在实数域 R 上的模糊集，若 A 同时满足：

- 正规模糊集合 (normal fuzzy set): $\exists x \in R$ 使得 $A(x) = 1$ ；
- 凸模糊集合 (convex fuzzy set): $\forall r \in (0, 1]$ 使得截集 $A_r = \{x|x \in R, A(x) \geq r\}$ 为凸集。

则称 A 为一个模糊数。以上为模糊数的严格定义。模糊数同时具有如下性质：若 A 为模糊数则当且仅当存在 $a \leq b$ 使得

- $A(x) = 1, \forall x \in [a, b]$ ；
- 在 $(-\infty, a)$ 上 $A(x)$ 为右连续的增函数且 $0 \leq A(x) < 1, \lim_{x \rightarrow -\infty} A(x) = 0$ ；
- 在 (b, ∞) 上 $A(x)$ 为左连续的减函数且 $0 \leq A(x) < 1, \lim_{x \rightarrow \infty} A(x) = 0$ 。

2. 除了集合间的运算，模糊数也支持基本的四则运算，计算结果也是一个模糊数。记两个模糊数 A, B , * 表示任意四则运算符，则可基于多元扩张原理将 $A * B$ 写为

$$(A * B)(x) = \bigvee_{x_1 * x_2 = x} (A(x_1) \wedge B(x_2)) = \max \{ \min\{A(x_1), B(x_2)\} \mid \forall x_1 * x_2 = x \}$$

$A * B$ 的表达式也可基于分解定理推导，有

$$(A * B)(x) = \bigvee_{\lambda \in [0, 1]} \lambda \wedge (A * B)_\lambda(x)$$

上式中 $(A * B)_\lambda = A_\lambda * B_\lambda$ ，由此可将模糊数的四则运算转化为区间的四则运算问题。记 $E = [a, b], F = [c, d]$ 表示两个实数区间，两者间的四则运算法则如下：

- $E \pm F = [a \pm c, b \pm d]$ ；
- $E \times F = [\min\{ad, ae, bc, bd\}, \max\{ad, ae, bc, bd\}]$ ；
- $E/F = [a, b] \times [1/d, 1/c]$ 。

3. 在模糊数的概念提出后，只需将一般算法或模型中的数换成模糊数即可得到基于模糊数学理论的相应算法或模型。但在实际应用中，往往最终需要计算确定性结果而非模糊数，为此则需要对模糊数进行去模糊化 (defuzzification) (类似于概率统计中的参数估计)，从模糊数表征的模糊集中提取最能表征模糊数性质的数值。需要说明的是，目前学界并不存在统一的去模糊化方法：

- 最常用的量化方法是基于隶属度函数的重心确定 (类似于概率论中的期望)，被称为重心去模糊法 (center-of-gravity defuzzification method)

$$COG(A) = \frac{\int x A(x) dx}{\int A(x) dx}$$

- 重心去模糊法是在论域上求积分，Chen 等提出的梯级平均积分表示法 (graded mean integration

²⁸模糊数学笔记：八、模糊数及其运算性质：<https://blog.csdn.net/cauchy7203/article/details/107492577>

representation, GMIR) 则是在隶属度维度上求积分²⁹，也是一种非常常用的去模糊化方法。由分解定理可知，任意模糊数可由其所有截集表示，而模糊数的截集为一区间，可由其中心表征，由此可通过所有截集的中心表征模糊数，这就是 GMIR 法的基本思想。定义 $L_A(\lambda), R_A(\lambda)$ 分别表征模糊数截集 A_λ 的左边界和右边界，严格的数学定义为

$$L_A(\lambda) = \begin{cases} \inf\{x|x \in A_\lambda\} & \lambda \in (0, 1] \\ \inf\{x|x \in \text{Supp}(A)\} & \lambda = 0 \end{cases}, \quad R_A(\lambda) = \begin{cases} \sup\{x|x \in A_\lambda\} & \lambda \in (0, 1] \\ \sup\{x|x \in \text{Supp}(A)\} & \lambda = 0 \end{cases}$$

上式中 \inf, \sup 分别表示下确界和上确界； $\text{Supp}(A) = \{x|x \in U, A(x) > 0\}$ 称为模糊集 A 的支集 (**support set**)。基于 $L_A(\lambda), R_A(\lambda)$ 即可得到任意截集 A_λ 的中点 $\frac{L_A(\lambda) + R_A(\lambda)}{2}$ ，从而类比重心的计算方法得到 GMIR 法的表征结果 $P(A)$ 为

$$P(A) = \int_0^1 \lambda \left(\frac{L_A(\lambda) + R_A(\lambda)}{2} \right) d\lambda / \int_0^1 \lambda d\lambda$$

- 与 GMIR 法的思想类似，Delgado 等人提出了更一般的模糊数定量表征方法³⁰。论文中同时提出了两个指标以分别表征模糊数的值 (**value**) 与模糊度 (**ambiguity**)。前者被记为 $V_s(A)$ ，与截集的中心位置有关；后者被记为 $A_s(A)$ ，与截集的大小有关

$$V_s(A) = \int_0^1 s(\lambda)[L_A(\lambda) + R_A(\lambda)]d\lambda, \quad A_s(A) = \int_0^1 s(\lambda)[R_A(\lambda) - L_A(\lambda)]d\lambda$$

上式中 $s(\lambda)$ 为关于 $\lambda \in [0, 1]$ 的单调增函数，且满足 $s(0) = 0, s(1) = 1$ 。通过选择不同形式的 $s(\lambda)$ ，可以在不同程度上滤去低隶属度的区间。当 $s(\lambda) = \lambda$ 时 $V_s(A)$ 在数值上即等于 GMIR 法的结果。

4. 最后介绍几种常用的模糊数：

- 三角模糊数 (**triangular fuzzy number**)：三角模糊数是最常用的模糊数，顾名思义，是指隶属度函数为三角形的模糊数，由三个参数表示，一般记为 $A = (a_1, a_2, a_3)$ ，表明隶属度函数在 $[a_1, a_2]$ 间从 0 线性递增至 1，在 $[a_2, a_3]$ 间从 1 线性递减至 0，而在其它区间取 0。当 $a_1 > 0$ 时称其为正三角模糊数 (**positive triangular fuzzy number**)；当 $a_3 < 0$ 时称其为负三角模糊数 (**negative triangular fuzzy number**)；当 $0 \in [a_1, a_3]$ 时称其为部分负三角模糊数 (**partial negative triangular fuzzy number**)。按重心去模糊法与 GMIR 法分别表征三角模糊数，有

$$COG(A) = \frac{a_1 + a_2 + a_3}{3}, \quad P(A) = \frac{a_1 + 4a_2 + a_3}{6}$$

三角模糊数具有如下运算特征：

- 相加（减）也是三角模糊数： $A \pm B = (a_1 \pm b_1, a_2 \pm b_2, a_3 \pm b_3)$ ；
- 相乘不是三角模糊数，但可近似为三角模糊数：

$$A \times B \simeq (\min\{a_1b_1, a_1b_3, a_3b_1, a_3b_3\}, a_2b_2, \max\{a_1b_1, a_1b_3, a_3b_1, a_3b_3\})$$

- 作商不是三角模糊数，但可近似为三角模糊数：

$$A/B = A \times \frac{1}{B}, \quad \frac{1}{B} = \left(\min \left\{ \frac{1}{b_1}, \frac{1}{b_2}, \frac{1}{b_3} \right\}, \text{medin} \left\{ \frac{1}{b_1}, \frac{1}{b_2}, \frac{1}{b_3} \right\}, \max \left\{ \frac{1}{b_1}, \frac{1}{b_2}, \frac{1}{b_3} \right\} \right)$$

除了近似计算，也可基于上文介绍的模糊数四则运算法则推导三角模糊数的解析相乘（除）结果，一般是通过截集 $(A \times B)_\lambda = A_\lambda \times B_\lambda$ 表示。三角模糊数的截集按定义写为

$$A_\lambda = [\underline{a}(\lambda), \bar{a}(\lambda)] = [(a_2 - a_1)\lambda + a_1, -(a_3 - a_2)\lambda + a_3]$$

²⁹Some properties of graded mean integration representation of LR type fuzzy numbers[J]. Tamsui Oxford Journal of Mathematical Sciences, 2006, 22(2): 185-208. https://www.researchgate.net/profile/Shu-Chang-13/publication/239281343_Some_Properties_of_Graded_Mean_Integration_Representation_of_LR_Type_Fuzzy_Numbers/links/564ed86308aefef619b0ff3d4/Some-Properties-of-Graded-Mean-Integration-Representation-of-LR-Type-Fuzzy-Numbers.pdf

³⁰Delgado M, Vila M A, Voxman W. On a canonical representation of fuzzy numbers[J]. Fuzzy sets and systems, 1998, 93(1): 125-135.

再根据上文介绍的区间四则运算法则，可以推导出三角模糊数相乘的解析结果

$$(A \times B)_{\lambda} = A_{\lambda} \times B_{\lambda} = [\min\{\underline{a}(\lambda)\underline{b}(\lambda), \underline{a}(\lambda)\bar{b}(\lambda), \bar{a}(\lambda)\underline{b}(\lambda), \bar{a}(\lambda)\bar{b}(\lambda)\}, \\ \max\{\underline{a}(\lambda)\underline{b}(\lambda), \underline{a}(\lambda)\bar{b}(\lambda), \bar{a}(\lambda)\underline{b}(\lambda), \bar{a}(\lambda)\bar{b}(\lambda)\}]$$

- 梯形模糊数 (trapezoidal fuzzy number): 隶属度函数为梯形的模糊数，由四个参数表示，记为 $A = (a_1, a_2, a_3, a_4)$ ，表明隶属度函数在 $[a_1, a_2]$ 间从 0 线性递增至 1，在 $[a_2, a_3]$ 间恒为 1，在 $[a_3, a_4]$ 间从 1 线性递减至 0，而在其它区间取 0。按重心去模糊法与 GMIR 法分别表征梯形模糊数有

$$COG(A) = \frac{1}{3} \left(a_1 + a_2 + a_3 + a_4 + \frac{a_3 a_4 - a_1 a_2}{a_1 + a_2 - a_3 - a_4} \right), \quad P(A) = \frac{a_1 + 2a_2 + 2a_3 + a_4}{6}$$

23.16 复变函数理论

23.16.1 约当引理 (Jordan lemma)

- 考虑一个实变函数的无穷积分问题 $\int_{-\infty}^{\infty} g(x)dx$ ($g(x)$ 在实轴上连续)，除了按一般方法计算外还可考虑将其转化为一个复变函数积分的问题

$$\int_{-\infty}^{\infty} g(x)dx = \lim_{R \rightarrow \infty} \int_{L_R} g(z)dz = \lim_{R \rightarrow \infty} \oint_{\Gamma_R} g(z)dz - \lim_{R \rightarrow \infty} \int_{C_R} g(z)dz$$

式中 L_R 表示复空间内从 $(-R, 0)$ 到 $(R, 0)$ 的直线（位于实轴上）， C_R 表示以原点为圆心、 R 为半径的上半圆圆弧（逆时针方向）， $\Gamma_R = L_R + C_R$ 即为逆时针的闭合曲线。闭合曲线积分 $\oint_{\Gamma_R} g(z)dz$ 可由留数定理转化为闭合曲线内极点留数计算问题而无需计算积分，因此当 $g(x)$ 的原函数计算较复杂时也可考虑将无穷积分问题 $\int_{-\infty}^{\infty} g(x)dx$ 转化为计算 $\lim_{R \rightarrow \infty} \int_{C_R} g(z)dz$ ；

2. 约当引理 (Jordan lemma)³¹ 适用于 $g(x) = f(x)e^{lix}$ (或 $f(x) \sin(\lambda x)$, $f(x) \cos(\lambda x)$)，且 $\lim_{x \rightarrow \infty} f(x) = 0$ 的情况。具体地，约当引理指出——设 $f(z)$ 在 $C_{\rho} : z = \rho e^{i\theta}$, $Im(z) \geq 0$ 上连续，且 $\lim_{z \rightarrow \infty, Im(z) \geq 0} f(z) = 0$ ，则有

$$\lim_{\rho \rightarrow +\infty} \int_{C_{\rho}} f(z)e^{\lambda iz} dz = 0, \quad \forall \lambda > 0, \quad Im(z) \geq 0 \quad (\text{约当引理})$$

其中 C_{ρ} 即为以原点为圆心、 ρ 为半径的上半圆圆弧（逆时针方向）

3. 进一步证明约当引理。将弧线积分转化为直线积分有

$$\lim_{\rho \rightarrow +\infty} \int_{C_{\rho}} f(z)e^{\lambda iz} dz = \lim_{\rho \rightarrow +\infty} \int_0^{\theta_0} f(z)e^{\lambda iz} d(\rho e^{i\theta}) = i \int_0^{\theta_0} \lim_{\rho \rightarrow +\infty} f(z)ze^{\lambda iz} d\theta$$

显然只需积分内的极限为 0 则约当引理得证。又因为 $\lim_{\rho \rightarrow +\infty} f(z) = 0$ ，则只需 $\lim_{\rho \rightarrow +\infty} ze^{\lambda iz} < \infty$ ，即 $\lim_{\rho \rightarrow +\infty} |ze^{\lambda iz}| < +\infty$

$$|ze^{\lambda iz}| = \rho |e^{i\theta} e^{\lambda \rho i(\cos \theta + i \sin \theta)}| = \rho |e^{\lambda \rho i \cos \theta - \lambda \rho \sin \theta + i\theta}| = \rho e^{-\lambda \rho \sin \theta} |e^{i(\lambda \rho \cos \theta + \theta)}| = \rho e^{-\lambda \rho \sin \theta}$$

又因为 $Im(z) \geq 0$ ，则 $\sin \theta \geq 0$ ，则显然有 $\lim_{\rho \rightarrow +\infty} |ze^{\lambda iz}| = \lim_{\rho \rightarrow +\infty} \rho e^{-\lambda \rho \sin \theta} = 0$ ，故引理得证。

23.16.2 柯西幅角原理 (principle of argument)

1. 幅角原理是复变函数论中基于留数定理的又一个重要定理，是线性控制的 Nyquist 稳定判据的理论基础³²。设 $f(z)$ 为域 U 上的亚纯函数³³， $\gamma \subset U$ 为一条简单正向封闭曲线，且在 U 中可连续地缩成一个点。记 $f(z)$ 在 γ 内有有限的 $N(f, \gamma)$ 个零点和 $P(f, \gamma)$ 个极点 (k 阶零、极点算 k 个)。令复数 $w = f(z)$ 。则幅角原理指出， z 沿 γ 正向绕行一圈造成 w 幅角的变化量 $\Delta_{\gamma}(Arg(w))$ 有

$$N(f, \gamma) - P(f, \gamma) = \frac{1}{2\pi} \Delta_{\gamma}(Arg(w)) \quad (\text{幅角原理})$$

³¹ 小大圆弧/Jordan 引理-复变函数笔记：<https://zhuanlan.zhihu.com/p/546412110/Jordan->

³² 【复变函数】1. 幅角原理：https://www.cnblogs.com/alphiy/c/p/complex_1.html

³³ 亚纯函数，即一个在域 $U \subseteq C$ 上有定义，并在除一个或若干个孤立点集合之外的区域处处解析的函数，这些孤立点称为极点。

上式中 $\frac{1}{2\pi} \Delta_\gamma(\operatorname{Arg}(w))$ 的物理意义为 w 绕复平面原点逆时针（正方向）转动的圈数；

2. 进一步证明幅角原理。记 $f(z)$ 在 γ 内各有 n, p 个独立的零、极点（高阶零、极点算一个），并记零点 z_k ($k = 1, 2, \dots, n$) 的阶数为 n_k ，极点 u_j ($j = 1, 2, \dots, p$) 的阶数为 p_j 。以每个零点为圆心作园 γ_k ($k = 1, 2, \dots, n$) 使 γ_k 在 γ 内部且互不相交；又以每个极点为圆心作园 C_j ($j = 1, 2, \dots, p$) 使 C_j 在 γ 内部且互不相交，则构造如下积分，并由 Cauchy 积分公式有

$$\frac{1}{2\pi i} \oint_{\gamma} \frac{f'(z)}{f(z)} dz = \frac{1}{2\pi i} \left(\sum_k \oint_{\gamma_k} \frac{f'(z)}{f(z)} dz + \sum_j \oint_{C_j} \frac{f'(z)}{f(z)} dz \right)$$

因为 z_k 为 $f(z)$ 的 n_k 阶零点，则 $f(z)$ 可写为

$$f(z) = (z - z_k)^{n_k} h_k(z) \implies \frac{f'(z)}{f(z)} = \frac{n_k(z - z_k)^{n_k-1} h_k(z) + (z - z_k)^{n_k} h'_k(z)}{(z - z_k)^{n_k} h_k(z)} = \frac{n_k}{z - z_k} + \frac{h'_k(z)}{h_k(z)}$$

其中 $h_k(z)$ 在 z_k 邻域内不为零，则基于留数定理有

$$\frac{1}{2\pi i} \oint_{\gamma_k} \frac{f'(z)}{f(z)} dz = \frac{1}{2\pi i} \oint_{\gamma_k} \frac{n_k}{z - z_k} + \frac{h'_k(z)}{h_k(z)} dz = \frac{1}{2\pi i} \oint_{\gamma_k} \frac{n_k}{z - z_k} dz = n_k$$

同理关注 $f(z)$ 的 p_j 阶极点 u_j ，将 $f(z)$ 写为

$$f(z) = (z - u_j)^{-p_j} g_j(z) \implies \frac{f'(z)}{f(z)} = \frac{-p_j(z - u_j)^{-p_j-1} g_j(z) + (z - u_j)^{-p_j} g'_j(z)}{(z - u_j)^{-p_j} g_j(z)} = -\frac{p_j}{z - u_j} + \frac{g'_j(z)}{g_j(z)}$$

同样地 $g_j(z)$ 在 u_j 邻域内不为零，基于留数定理有

$$\frac{1}{2\pi i} \oint_{C_j} \frac{f'(z)}{f(z)} dz = \frac{1}{2\pi i} \oint_{C_j} -\frac{p_j}{z - u_j} + \frac{g'_j(z)}{g_j(z)} dz = -\frac{1}{2\pi i} \oint_{C_j} \frac{p_j}{z - u_j} dz = -p_j$$

则代入最初构造的积分式有

$$\frac{1}{2\pi i} \oint_{\gamma} \frac{f'(z)}{f(z)} dz = \frac{1}{2\pi i} \left(\sum_k \oint_{\gamma_k} \frac{f'(z)}{f(z)} dz + \sum_j \oint_{C_j} \frac{f'(z)}{f(z)} dz \right) = \sum_k n_k - \sum_j p_j = N - P$$

最后只需再证等式左边积分 $\oint_{\gamma} \frac{f'(z)}{f(z)} dz$ 表示 $w = f(z)$ 幅角的变化量 $\Delta_\gamma(\operatorname{Arg}(w))$ 。记 γ 在 $w = f(z)$ 的映射为 Γ ，并以极坐标形式将 w 记为 $w = \rho e^{i\varphi}$ ， φ 即为 w 的幅角，则按复域积分的定义³⁴有

$$\oint_{\gamma} \frac{f'(z)}{f(z)} dz = \oint_{\Gamma} \frac{1}{w} dw = \oint_{\Gamma} \frac{1}{w} d(\rho e^{i\varphi}) = \oint_{\Gamma} \frac{1}{\rho e^{i\varphi}} (e^{i\varphi} d\rho + i\rho e^{i\varphi} d\varphi) = \int_{\rho_1}^{\rho_2} \frac{1}{\rho} d\rho + \int_{\varphi_1}^{\varphi_2} i d\varphi = \ln \frac{\rho_2}{\rho_1} + i(\varphi_2 - \varphi_1)$$

因为 γ 为简单闭合围线，则 Γ 必然也为闭合围线，有 $\rho_1 = \rho_2$ 、 $\varphi_2 - \varphi_1$ 为 2π 的整数倍，则幅角原理得证

$$\oint_{\gamma} \frac{f'(z)}{f(z)} dz = i(\varphi_2 - \varphi_1) = i\Delta_\gamma(\operatorname{Arg}(w))$$

³⁴复变函数的积分：<https://wuli.wiki/online/CpxInt.html>

赌书消得泼茶香 当时只道是寻常

第七部分

人工智能



第 24 章

人工神经网络 (Artificial Neural Network, ANN)

24.1 人工神经网络初入

1. 人工神经网络是对自然神经网络 (**Nature neural network, NNN**) 的模拟。人脑由一千多亿 ($10^{11} - 10^{14}$) 个神经元组成，神经元约有 1000 种类型，每一个神经元大约与 $10^3 - 10^4$ 个其它神经元相连，形成极其复杂又多变的网状结构。神经元可以接收来自其他神经元或外界的刺激，根据接收的刺激的不同，神经元具有兴奋和抑制两种状态；
2. 人工神经元的一般数学模型如下：

$$y_i = f(u_i) \quad u_i = \sum_{j=1}^n w_{ij}x_j + \theta_i = \mathbf{W} \cdot \mathbf{X}$$

上式表示神经元 i 接收刺激 \mathbf{X} 输出 y_i 的过程如下：

- (a) 首先对刺激 \mathbf{X} 经过线性运算得到 u_i 。 \mathbf{W} 中 w_{ij} 为对应刺激的权重，其中包括使神经元 i 表现为兴奋或抑制的阈值 θ_i ，是偏置 1 的权重；
- (b) 对线性运算的结果 u_i 代入非线性函数 $f(u)$ ，得到 y_i 。 $f(u)$ 称为激励函数 (**activation function**)。常见的激励函数如下：
 - sigmoid 函数， $y = f(u) = \frac{1}{1+e^{-u}}$ ，第一个用于神经网络的激励函数，为一个值域在 $(0, 1)$ 区间的 S 型曲线；
 - tanh 函数， $y = f(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$ ，为一个值域在 $(-1, 1)$ 区间的 S 型曲线；
 - softmax 函数， $y_i = f(u_i) = \frac{e^{u_i}}{\sum_i e^{u_i}}$ ，常用于多分类问题，将多个输出 u_i 映射至取值范围为 $(0, 1)$ 区间的 $\sum_i y_i = 1$ ，且满足 $\sum_i y_i = 1$ 。可以计算得 softmax 的导数如下：

$$\frac{\partial y_i}{\partial u_j} = y_i(\delta_{ij} - y_j) = \begin{cases} y_i(1 - y_j) & i = j \\ -y_i y_j & i \neq j \end{cases}$$

- ReLU 函数， $y = f(u) = \max(0, u)$ ，全称为 Rectified Linear Units，为一个在原点左侧取 0、原点右侧斜率为 1 的直线的折线，常用于卷积神经网络；
- Linear 函数， $y = f(u) = u$ ，在实际中极少用，因为会降低非线性分类的效果，一般仅用于输出结果取值范围为 $(-\infty, +\infty)$ 的情况。

通过调整权重及选择合适的激励函数，即可控制神经网络的输出；

3. 人工神经网络的结构包括前馈型和反馈型两类；
4. 过拟合 (**overfitting**) 是所有机器学习模型都需要面对的问题，而这一问题在人工神经网络中往往较为严重，这是因为人工神经网络往往具有大量的神经元，每一神经元都可用于特征提取，从而获得极强的特征提取能力，不可避免地包括样本中的噪声和特例信息，从而出现过拟合现象，因此降低结构复杂度可有效克服过拟合。过拟合的具体表现是：随着训练的推进，模型在训练集上的误差逐渐减少、正确

率逐渐提高，而在验证集上的误差先减少后增加，验证集误差增加的过程就是出现过拟合的过程。在训练过程中克服过拟合的具体方法如下：

- **提前终止 (Early stopping):** 即在发现验证集误差自减少开始转为增加后终止神经网络训练；
- **Dropout:** 这是一种神经网络训练过程中常用的克服过拟合的方法。即在每一轮训练中，随机地关闭一些神经元，这种方法在一定程度上减少了参数的数量，减小了过拟合的风险；
- **增加参数惩罚项:** 这一方法在经典机器学习中也有应用（如岭回归），即在已有的损失函数 J 后面增加表示模型复杂度的惩罚项，称为正则化因子，有“L1 参数范数惩罚项”和“L2 参数范数惩罚项”两类

$$L1 : J' = J + \frac{\lambda}{n} \sum_w |w| \quad L2 : J' = J + \frac{\lambda}{2n} \sum_w w^2$$

式中 n 为参数 w 的个数， λ 称为正则化系数或惩罚系数。新的损失函数中第一项表示“经验风险”，后半部分表示“结构风险”。 λ 一般较小，越大使得 w 越趋向于 0，即网络越简洁。

5. 对于复杂的人工神经网络（如深度神经网络），往往含有大量的神经元参数 w, b ，这些如此多的参数之间又有着复杂的关系，以至于早已失去了统计学上的意义，无法得到清晰的物理解释，也就无法进行有效的逆向研究。所以人工神经网络模型往往被视为“黑匣子”，这一点遭到大量学者的诟病，但又因为其在实验中表现的良好的泛化性能而依然得到广泛关注。

24.1.1 BP (backward propagation) 算法与 BP 神经网络

1. BP 神经网络是一种基本的前馈型神经网络，一般由一个输入层、一个输出层及至少一个隐藏层组成，每一个神经元连接下一层的所有神经元（图 24.1）。BP 神经网络得名于 BP 算法，是一种将神经网络输出值与实际值的误差向后传递逐层修改每一神经元权重 w_{ij} ，直至输出值与实际值的误差低于可接受程度的算法；

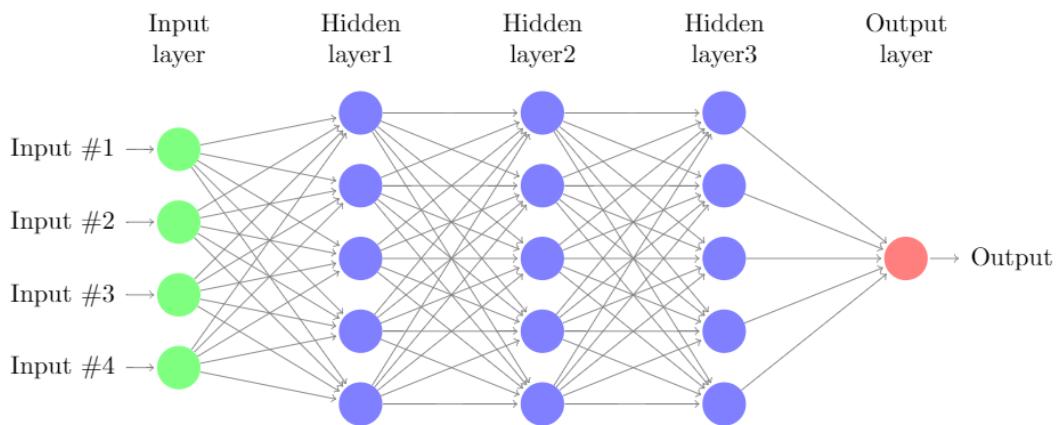


图 24.1 BP 神经网络基本图



Kolmogorov 定理

给定任意 $\varepsilon > 0$ ，对于任意的 L_2 型连续函数 $f : [0, 1]^n \rightarrow \mathbb{R}^m$ ，存在一个三层 BP 神经网络，其输入层有 n 个神经元，中间层有 $2n+1$ 个神经元，输入层有 m 个神经元，它可以在任意 ε^2 误差精度内逼近 f 。

2. Kolmogorov 定理保证了 BP 神经网络具有很好的逼近特性、较强的泛化能力和较好的容错性，而 BP 算法则提供了实现途径。BP 算法分为正向传播和反向传播两部分，正向传播即是根据输入的特征预测标签，反向传播即是根据预测值与实际值的误差反向逐层修正每一神经元的权重；
3. 以下重点介绍反向传播的算法。假设第 $k (k \leq m)$ 层第 i 个神经元输出的结果为 y_i^k ， y_i^m 即为输出层神经元

的输出结果，假设实际值为 y_{si} ，则构造预测值与实际值的误差函数 J 如下

$$J = \frac{1}{2} \sum_i (y_i^m - y_{si})^2 \quad y_i^k = f(u_i^k) = f\left(\sum_j w_{ij}^{k-1} y_j^{k-1}\right)$$

式中系数 $1/2$ 为简化求导结果所设。采用梯度下降算法对逐层权重 w_{ij}^{k-1} 进行优化使误差函数 J 最小。设定权重 w_{ij}^{k-1} 的学习步长 ε （一般取 0.5），则优化过程如下

$$w_{ij,new}^{k-1} = w_{ij}^{k-1} + \Delta w_{ij}^{k-1} \quad \Delta w_{ij}^{k-1} = -\varepsilon \frac{\partial J}{\partial w_{ij}^{k-1}}$$

可以看到，优化的关键在于 $\frac{\partial J}{\partial w_{ij}^{k-1}}$ 的计算

$$\frac{\partial J}{\partial w_{ij}^{k-1}} = \frac{\partial J}{\partial y_i^k} \frac{\partial y_i^k}{\partial u_i^k} \frac{\partial u_i^k}{\partial w_{ij}^{k-1}} = \frac{\partial J}{\partial y_i^k} f'_{i,k}(u_i^k) y_j^{k-1} = d_i^k y_j^{k-1}$$

对输出层

$$\frac{\partial J}{\partial y_i^k} = \frac{\partial J}{\partial y_i^m} = y_i^m - y_{si} \implies d_i^m = \frac{\partial J}{\partial y_i^m} f'_{i,m}(u_i^m) = (y_i^m - y_{si}) f'_{i,m}(u_i^m)$$

对隐藏层及输入层

$$\frac{\partial J}{\partial y_i^k} = \sum_j \frac{\partial J}{\partial u_j^{k+1}} \frac{\partial u_j^{k+1}}{\partial y_i^k} = \sum_j d_j^{k+1} w_{ij}^k \implies d_i^k = \frac{\partial J}{\partial y_i^k} f'_{i,k}(u_i^k) = f'_{i,k}(u_i^k) \sum_j d_j^{k+1} w_{ij}^k$$

综上所述

$$\Delta w_{ij}^{k-1} = -\varepsilon \frac{\partial J}{\partial w_{ij}^{k-1}} = -\varepsilon d_i^k y_j^{k-1} \quad d_i^k = \begin{cases} (y_i^m - y_{si}) f'_{i,m}(u_i^m), & k = m \\ f'_{i,k}(u_i^k) \sum_j d_j^{k+1} w_{ij}^k, & k < m \end{cases}$$

- 以上即为 BP 算法的推导。BP 神经网络的优点非常突出，但其缺点也很明显：收敛速度慢、易陷入局部最优、难以确定隐藏层数目及层内节点数目、层数过多时容易出现梯度消失或梯度爆炸、容易过拟合等。

24.2 循环神经网络 (RNN) 与长短期记忆 (LSTM)、GRU 结构

- 循环神经网络 (recurrent neural network, RNN) 是一类反馈型神经网络，其最大特点即是建立了同一层神经元之间的连接，每一神经元的输出与上一层神经元的输出和同一层中的上一个神经元的输出均有关，这一特点使得 RNN 可广泛用于序列预测、语言分析等与顺序有关的任务，RNN 也可用于图像、视频识别等部分分类任务。与其它神经网络一样，RNN 也可以有多个隐藏层，然而 RNN 的参数过于复杂，使得多层 RNN 训练时效率太慢且容易过拟合，大多数情况下只需一个隐藏层即可满足需求，一般不会超过两层；
- 经典的 RNN 结构如图 24.2，以单层 RNN 为例，每一单元基于输入的 x_t 及上一单元的 h_{t-1} 经过线性及非线性变化得到 h_t ， h 称为隐藏状态，隐藏状态 h_t 再经过一次线性和非线性变化才得到神经网络的输出结果 o_t

$$h_t = \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{t-1} + b_{hh}) = \phi(U \cdot x_t + W \cdot h_{t-1})$$

$$o_t = \varphi(V \cdot h_t)$$

RNN 的参数包括向量 U, W, V ，参数的学习过程同样是误差 J 的向后传递，即计算 $\frac{\partial J}{\partial U}, \frac{\partial J}{\partial W}, \frac{\partial J}{\partial V}$ ，因为神经元的输出结果与之前神经元的隐藏状态有关，因此求导时与 BP 算法存在微小差异，称为 BPTT 算法 (backward propagation through time)；

- 同一般前馈型网络层数过多时易发生梯度消失一样，RNN 长度过长时同样也存在梯度消失的现象，表现为序列前端的元素对后端元素影响逐渐减小（即记忆力差），从而降低了神经网络对序列全局信息的利用能力，于 1997 年提出的长短期记忆 (long-short term memory, LSTM) 结构即是解决上述问题的一类有效方法；

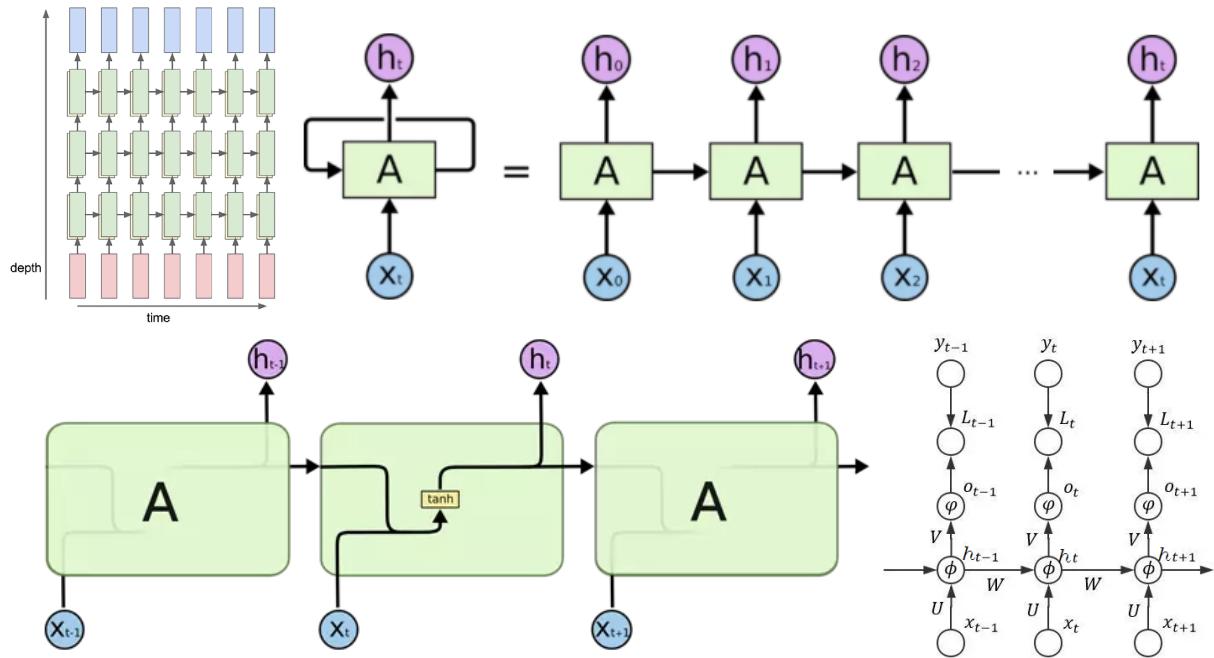


图 24.2 经典 RNN 一般结构

4. LSTM 结构在隐藏状态 h_t 之外引入了细胞状态 c_t (cell state), c_t 用于保留序列信息中有助于预测的信息, 代表长期记忆, 与 h_t 所代表的短期记忆共同组成长-短期记忆。另外, 引入遗忘门、输入门和输出门建立 c_t, h_t, x_t 之间的联系:

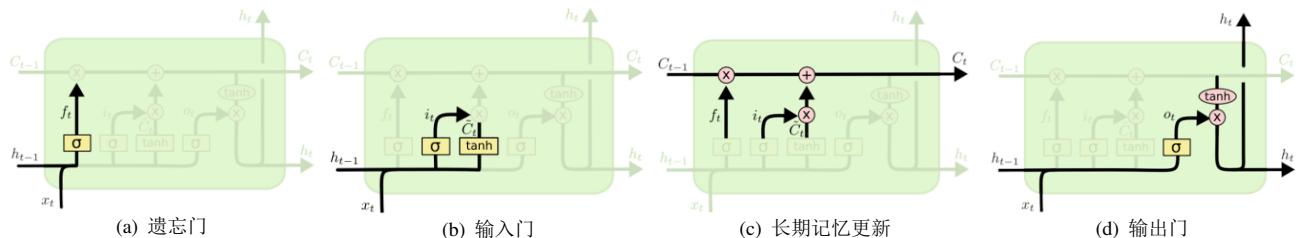


图 24.3 LSTM 结构

- (a) 遗忘门结构如图 24.3(a), 用于决定应该在多大程度上遗忘细胞状态 c_{t-1} (长期记忆)。式中 σ 为 sigmoid 激励函数; $f_t \in (0, 1)$, 表示遗忘 c_{t-1} 的程度, 越接近 1 表示遗忘程度越低;

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf})$$

- (b) 输入门结构如图 24.3(b), 用于决定应该多大程度地将新的信息加入细胞状态 c_{t-1} (长期记忆)

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi})$$

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg})$$

遗忘门和输入门的结果共同决定细胞状态的更新结果 c_t 。式中 \odot 表示哈达玛积 (Hadamard product)¹;

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

- (c) 输出门结构如图 24.3(d), 基于更新的细胞状态 c_t 、输入值 x_t , h_{t-1} 输出当前的隐藏状态 h_t

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho})$$

$$h_t = o_t \odot \tanh(c_t)$$

¹一种特殊的矩阵乘法, 两矩阵对应元素相乘, 得到新的矩阵。

5. LSTM 结构的引入有效地解决了一般 RNN 网络的梯度消失问题，然而多出的细胞状态、遗忘门、输入门和输出门意味着更多地训练参数和更长的训练时间，为此有多种 LSTM 的变体以解决上述问题，2014 年提出的 GRU 结构即是其中最著名的一种变体。**GRU (Gated Recurrent Unit)** 仅包含重置门 (**reset gate**) 和更新门 (**update gate**)，并且将细胞状态与隐藏状态进行合并，与 LSTM 相比，GRU 结构更为简洁、训练时间更短而又往往能取得接近的结果

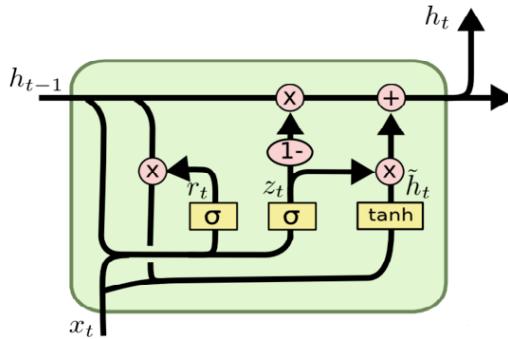


图 24.4 GRU 结构

(a) 重置门用于控制忽略前一时刻的状态信息的程度，重置门的值越小说明忽略得越多

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr})$$

(b) 更新门用于控制前一时刻的状态信息被带入到当前状态中的程度，更新门的值越大说明前一时刻的状态信息带入越多

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz})$$

(c) 基于重置门和更新门的结果 r_t , z_t 与 x_t , h_{t-1} 得到当前的隐藏状态 h_t

$$n_t = \tanh[W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{t-1} + b_{hn})]$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot n_t$$

24.3 面向序列数据的卷积模型

- 在自注意力机制提出前，主流的神经网络结构可简单分为卷积结构和循环结构。其中卷积模型在计算机视觉任务占垄断地位，而循环模型（RNN 及其变体）则主要聚焦序列数据处理；
- 在深度学习中，序列数据一般包含时序数据和自然语言数据两类。一个微小的差别在于时序数据存在严格的时间因果关系（时间不可逆），而自然语言的前后因果关系则不完全（后一个单词未必与之前的单词有因果关系，而是由整体句子的意思决定）；
- 尽管循环神经网络在结构上天然适用于序列数据建模，但在实际应用中也因其循环结构使得无法并行优化。相比之下，卷积运算则可通过并行优化显著减少训练时间。因此在自注意力机制提出前，学者曾尝试设计可媲美 RNN 的针对序列数据建模的卷积神经网络模型，部分研究至今仍有相当的应用价值。

24.3.1 门控线性单元 (gated linear unit, GLU)

- GLU 模块由 Yann N. Dauphin 于 2016 年提出^{2,3}，是一种基于卷积的序列数据建模方法。**GLU 模块抛弃了经典的 RNN 循环结构，通过卷积运算并行化挖掘上下文语义特征，从而在语言模型的建模方法上更具有竞争力。**与 RNN 结构相比，其复杂度从 $O(N)$ 降低到 $O(N/k)$ ，其中 N, k 分别表示文本上下文集合和卷积核宽度；

²Yann N. Dauphin, Angela Fan, Michael Auli, David Grangier. Language Modeling with Gated Convolutional Networks: <https://arxiv.org/abs/1612.08083>

³FLASH: 高效 Transformer 解析 (1)—GLU(Gated Linear Unit, 门控线性单元): <https://zhuanlan.zhihu.com/p/486055017>

2. GLU 的结构非常简单。记输入时间序列（或文本向量）为 $X \in \mathbb{R}^{N \times m}$, 其中 N, m 分别表示序列长度和序列中单个元素的表示向量维度。GLU 模块基于两个平行的卷积核挖掘 X 的语义信息

$$h_l(X) = (X * W + b) \odot \sigma(X * V + c)$$

上式中 $*$ 表示卷积运算; \odot 表示 Hadamard 积; W, V 表示两个独立的卷积核; W, V, b, c 均为待学习参数; $\sigma(\cdot)$ 表示 sigmoid 激励函数。可以看到, GLU 模块对 X 作两次卷积后, 将其中一次卷积的结果进行 sigmoid 归一化作为门控依据对另一次卷积的结果进行过滤, 这也是名称中“门控”的由来。而“线性”是因为 $X * W + b$, $X * V + c$ 运算皆为线性, 且 sigmoid 函数在中间部分（近 0 端）表现近似于线性, 所以整个模型的复杂度基本近似于线性;

3. 与 CNN 网络一样, GLU 模块的感受野由卷积核尺寸决定, 通过堆叠多个 GLU 模块可以扩大感受野, 挖掘上下文相关信息。另外, 为避免 sigmoid 可能导致的梯度弥散, 堆叠 GLU 模块时还可加入残差结构。

24.3.2 因果卷积模型 (causal convolutional network)

1. 因果卷积最初于 2016 年被提出以处理信号问题⁴, 并成为 2018 年提出的时序卷积神经网络模型 (temporal convolutional network, TCN) 的核心⁵。“因果卷积”指卷积过程中考虑序列的先后依赖关系;
2. 序列预测任务所遵循的最基本的因果关系即是未来状态应由过去及当前状态决定 (即格兰杰因果), 其通式可写为

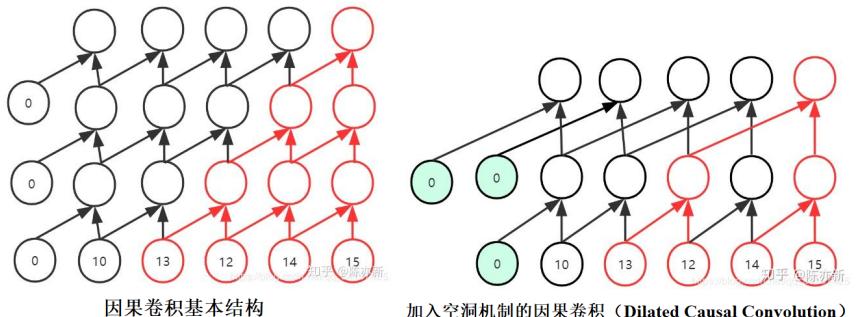
$$y_{t+1} = f(x_t, x_{t-1}, \dots, x_1)$$

对于人工神经网络而言, RNN 结构天然地满足上述形式, 而全连接及卷积则并不满足。因为输出的靠前的 (前一时刻) 神经元与输入的靠后的 (后一时刻) 神经元相互产生了连接。解决这一问题的思路也很简单, 只需去除全连接结构中从未来神经元往过去神经元的连接, 此时得到最基本的因果卷积结构;

3. 与一般的卷积模型一致, 为扩大感受野 (reception field) 以捕捉序列的长程相关性从而提升预测精度, 需增加因果卷积的卷积核大小或增加卷积层数目。前者将导致模型忽略序列的短程相关性, 后者也将显著增加模型复杂度, 且无论何种做法均只能在线性时间尺度上扩大对历史信息的捕捉;
4. 为使得在网络不太深的情况下捕捉尽可能多的信息历史, 一般采用空洞因果卷积 (dilated causal convolution) 又称膨胀因果卷积或扩张因果卷积。空洞因果卷积是在因果卷积时引入空洞机制 (dilatation)。一般的卷积核是连续的, 从而集计输入数据中局部连续区域的特征 (假设卷积核尺寸为 2, 则卷积时考虑的数据为 x_t, x_{t-1} 、感受野为 2)。而空洞机制则是在卷积核中加入空洞, 从而在单次卷积时隔去部分数据, 进而在卷积核尺寸不变的前提下扩大感受野 (假设卷积核尺寸为 2、空洞大小为 2, 则卷积时考虑的数据为 x_t, x_{t-2} 、感受野为 3)。一维场景下的扩张因果卷积数学形式表示为

$$F(t) = (x *_d f)(t) = \sum_{i=0}^{k-1} f(i) \cdot x_{t-d \cdot i}$$

图 24.5 因果卷积模型。若卷积核大小为 2, 欲使序列预测模型的感受野为 4, 则最基本的因果卷积模型需要 3 层卷积层, 而引入空洞机制后可仅凭更少的卷积层实现更大的感受野。



⁴Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio (2016): <https://arxiv.org/abs/1609.03499>

⁵Bai, S., Kolter, J. Z. & Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling (2018): <https://arxiv.org/pdf/1803.01271.pdf>

上式中 d 表示空洞参数 (dilation parameter), $d = 1$ 表示卷积核中无空洞; $*_d$ 即表示空洞参数为 d 的因果卷积; f 表示卷积核, 其大小为 k , $f(i)$ 为卷积核的第 i 个参数。另外一般令空洞大小随网络深度增加以快速扩大感受野, 对于第 l 层空洞因果卷积层, 常令 $d = 2^{l-1}$;

- 在加入空洞机制后, 为使得卷积后的维度与输入相同, 每一层卷积的 padding 应设置为 $(k - 1) \times d$ 。

24.4 注意力机制 (Attention mechanism)

- 注意力 (Attention) 机制**⁶是一种借鉴生物视觉逻辑提出的深度学习模型结构, 其基本目标是加强深度学习网络对输入数据中重要部分的关注度。注意力机制最早于 2014 年由 Google Mind 针对视觉相关任务提出, 而后逐渐被应用于 NLP 等其它领域;
- 早期的注意力机制仅作为主流神经网络模型 (CNN、RNN 等) 内部的补充结构, 且具体的计算方法仍处于探索阶段。直至 2017 年 Google 机器翻译团队发表的 *Attention is All You Need* 中, 完全抛弃了 RNN 和 CNN 等网络结构, 而仅仅采用注意力机制来进行机器翻译任务取得了很好的效果, 注意力机制的具体形式才基本确定, 并使其成为与 CNN、RNN 等并列的主流神经网络结构;
- 现如今最经典的注意力机制包括自注意力机制、通道注意力机制和空间注意力机制三类。在此基础上发展出的神经网络模型已替代 RNN 成为最主流的 NLP 模型; 而在计算机视觉领域, 基于注意力机制的模型也已达到了媲美 CNN 的效果;
- 注意力机制模型较 CNN、RNN 模型具有参数少、复杂度低的优点; 而且解决了 RNN 不能并行计算的问题, 和 CNN 均可通过并行计算提升效率。注意力机制是目前学习长距离信息的最有效方法。模型基于注意力机制对输入序列的不同位置分配不同的权重, 从而在处理每个序列元素时专注于最相关的部分。

24.4.1 自注意力 (self-attention) 与多头自注意力 (multi-head self-attention)

- 自注意力 (self-attention) 和多头自注意力 (multi-head self-attention) 机制是最经典的注意力机制模型, 于 *Attention is All You Need* (2017) 中最初针对 NLP 问题提出。所谓“自注意力”即是关注输入序列两两元素间的相关性;
- 记模型输入序列为矩阵 X , 其中第 i 个列向量 x_i 表示输入序列第 i 个元素的原始属性向量; X 经自注意力模块编码后的输出为矩阵 Y , 每一列向量 y_i 为 x_i 的对应输出。因为考虑了两两元素间的相关性, 则输出结果 y_i 应既与 x_i 的固有特征相关, 又考虑输入序列中其它元素特征对 x_i 的影响, 因此不妨记

$$y_i = \sum_j \hat{\alpha}_{ij} v_j \iff Y = V \hat{A}, \quad V = W^v X$$

上式中列向量 v_j 表示基于 x_j 提取的特征向量, 表示元素 x_j 的固有特征; V 为所有 v_j 构成的矩阵; 标量 $\hat{\alpha}_{ij}$ 为注意力, 表示元素 x_j 对 x_i 的影响; \hat{A} 为所有 $\hat{\alpha}_{ij}$ 构成的矩阵。自注意力机制的核心即是计算 $\hat{\alpha}_{ij}$;

- 为量化元素 x_j 对 x_i 的影响以计算 $\hat{\alpha}_{ij}$, 显然需要首先构造描述 x_i, x_j 的特征向量, 且该向量应不同于 v_i, v_j 。以输入序列“赌书消得泼茶香”为例, 每个字对应 X 的一个列向量。则 v_0 应侧重于描述“赌”的固有含义 (字、词信息), 而 $\hat{\alpha}_{01}$ 应侧重于解读动宾结构“赌书”中动词“赌”与名词“书”配对对“赌”含义的影响 (搭配、语法信息)。因此需要构造关于 X 的至少另一组特征编码 (记为 $K = W^k X$) 以计算 $\hat{\alpha}_{ij}$, 例如

$$\hat{\alpha}_{ij} = \frac{\exp\{s(k_i, k_j)\}}{\sum_j \exp\{s(k_i, k_j)\}}$$

上式中 s 为任意二元相似度 (相关性) 函数, 经 softmax 归一化后即得到 $\hat{\alpha}_{ij}$;

- 上述逻辑具有一定合理性, 但按此设计的注意力机制模型将存在复杂性过低的问题。因为大多数场景中要求 $\hat{\alpha}_{ij} \neq \hat{\alpha}_{ji}$ 。若只基于一组特征编码 K , 则只能通过限制相似度函数 s 的形式满足不对称约束。因此, 为保证模型的灵活性和通用性, 应至少构造关于 X 的两组特征编码以计算 $\hat{\alpha}_{ij}$

$$\hat{\alpha}_{ij} = \frac{\exp\{s(q_i, k_j)\}}{\sum_j \exp\{s(q_i, k_j)\}}, \quad Q = W^q X, \quad K = W^k X$$

⁶注意力机制综述 (图解完整版附代码): <https://zhuanlan.zhihu.com/p/631398525>

5. 以上即为自注意力机制的基本结构，其中基于 X 提取的三类向量 Q, K, V 分别称为 **query** 向量、**key** 向量和 **value** 向量，而常用的相似度函数 s 包括：
- 点乘： $s(q, k) = q^\top k$ ；
 - cos** 相似度： $s(q, k) = \frac{q^\top k}{\|q\| \cdot \|k\|}$ ；
 - 串联拼接： $s(q, k) = \text{Concat}(q, k)$ ；
 - 全连接层： $s(q, k) = \tanh(Wq + Uk)$ 。

6. 多头自注意力 (**multi-head self-attention**) 机制是自注意力机制的变体。类似于 CNN 模型中采用多个不同尺寸的卷积核可提取数据的不同尺度信息，对数据应用多次自注意力模块有助于挖掘不同类别的注意力信息，从而增强模型的表达能力和泛化能力。具体地，在对 X 编码提取 Q, K, V 后进一步细化

$$Q^{(\eta)} = W^{q,\eta} Q, \quad K^{(\eta)} = W^{k,\eta} K, \quad V^{(\eta)} = W^{v,\eta} V, \quad \eta = 1, \dots, n$$

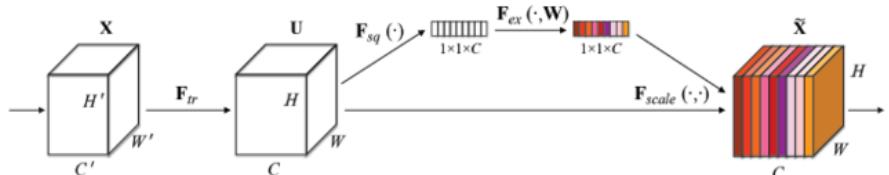
上式中 $\{(W^{q,\eta}, W^{k,\eta}, W^{v,\eta}) | \eta = 1, \dots, n\}$ 为模型待优化参数。每一组独立的 $Q^{(\eta)}, K^{(\eta)}, V^{(\eta)}$ 被称为一个“自注意力头 (**self-attention head**)”，这正是“多头自注意力”的由来。基于每一组自注意力头输出一组 $Y^{(\eta)}$ ，按列拼接所有 $Y^{(\eta)}$ 并聚合即可得到多头自注意力模块的最终输出

$$Y = W^y \text{Concat}(Y^{(1)}, \dots, Y^{(n)})$$

24.4.2 通道注意力 (**channel attention**) 与空间注意力 (**spatial attention**)

- 由名可知，通道注意力 (**channel attention**) 与空间注意力 (**spatial attention**) 均是针对视觉模型提出。视觉模型的处理对象一般为三维张量 $C \times H \times W$ ，其中 C 表示图像的通道数， H, W 分别表示图像的高和宽。通道注意力模块即是旨在捕捉输入图像各通道的重要度，而空间注意力模块则捕捉图像中不同区域的重要度，两者也可联合使用。不同于自注意力，目前尚无绝对标准的计算通道注意力和空间注意力的方法，相关模块一般也不作为单独的网络结构，而是作为卷积模块的补充；
- 首先介绍几种经典的通道注意力模型。Squeeze-and-Excitation Networks (SENet) 是由自动驾驶公司 Momenta 在 2017 年公布的一种全新的图像识别结构，通过引入通道注意力结构（原文称为 SE block）获得 2017 ILSVR 竞赛的冠军。如下图所示，张量 X 为输入，经卷积层后输出张量 U ，后续的部分即为模型中的通道注意力模块。模块包括一个 Squeeze 函数 $F_{sq}(\cdot)$ 和一个 Excitation 函数 $F_{ex}(\cdot)$ 。其中 $F_{sq}(\cdot)$ 通过全局平均池化操作将每个通道的特征图转化为一个标量值，从而将 U 压缩为 $1 \times 1 \times C$ 的张量，再输入 $F_{ex}(\cdot)$ 中。 $F_{ex}(\cdot)$ 通过全连接层网络和激励函数挖掘 $1 \times 1 \times C$ 张量信息，得到各个通道的重要度（即通道注意力），同样表示为 $1 \times 1 \times C$ 张量。将通道注意力与原张量 U 点乘（即 $F_{scale}(\cdot)$ 函数），最终得到考虑通道重要度差异的输出 \tilde{X} 。SENet 中额外加入的通道注意力模块计算量显著小于卷积运算，因此在一般 CNN 模型中加入该结构有助于在计算量未显著增加的前提下提升学习效果；

图 24.6 SENet 模型中的通道注意力模块

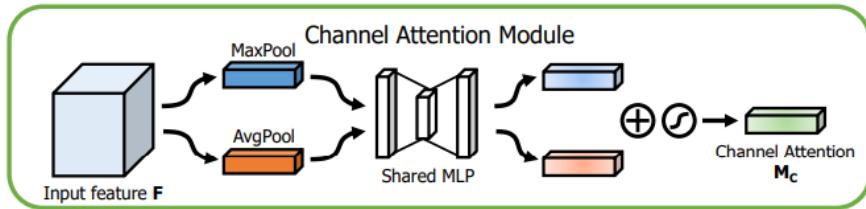


- Convolutional Block Attention Module (CBAM) 发表于 ECCV 2018，是一种混合了通道注意力和空间注意力的针对 CNN 模型的注意力模块。其通道注意力的计算方法与上述 SENet 模型并不完全一致。如下图所示，模型在压缩各通道空间信息时除了平均池化外还考虑了最大值池化，而后通过全连接神经网络和激励函数提取通道注意力信息

$$M_c = \sigma(W_1 \text{ReLU}(W_0 F_{avg}^c)) + \sigma(W_1 \text{ReLU}(W_0 F_{max}^c))$$

上式中 F_{avg}^c, F_{max}^c 分别表示基于平均池化和最大值池化压缩的各通道信息； W_0, W_1 表示两个全连接层的权重； σ 表示 sigmoid 激励函数；

图 24.7 CBAM 模型中的通道注意力模块

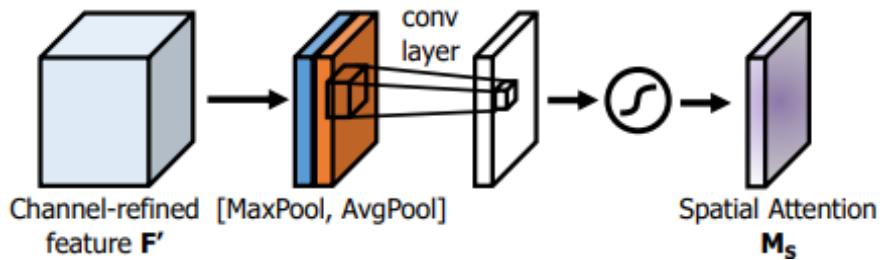


4. 进一步介绍 CBAM 模块中的空间注意力计算方法。如下图所示，同样采用平均池化和最大值池化，但此时是压缩各空间位置处的信息。将两次池化的结果堆叠为 $2 \times H \times W$ 的张量后基于卷积层和激励函数提取 $1 \times H \times W$ 空间注意力信息

$$M_s = \sigma(f^{7 \times 7}([F_{avg}^s; F_{max}^s]))$$

上式中 F_{avg}^s, F_{max}^s 分别表示基于平均池化和最大值池化压缩的各空间位置信息； $f^{7 \times 7}$ 表示卷积核大小为 7×7 的卷积操作； σ 表示 sigmoid 激励函数；

图 24.8 CBAM 模型中的空间注意力模块



24.5 Word2Vec 工具包

Word2Vec 是 2013 年 Google 开源的一款用于词向量计算的工具包⁷，一经问世直接将 NLP 领域带到了一个新的高度。在 2018 年 BERT（见第 24.7.2 节）被提出之前，Word2Vec 一直是 NLP 算法工程师追捧的预训练词向量模型。与之前的词嵌入方法相比，Word2Vec 可以在百万数量级的词典和上亿的数据集上进行高效地训练；其次，该工具得到的训练结果——词向量 (word embedding)，可以很好地度量词与词之间的相似性。需要说明的是，Word2Vec 只是一个计算词向量的工具包，底层的计算方法包括两个模型——Skip-Gram 模型与 CBOW 模型。前者的目标是基于输入的词向量预测上下文内容，而后者是根据上下文内容预测中间词的词向量。Word2Vec 的成功启发了后续大量工作，甚至包括 NLP 领域以外的研究（如推荐系统、图学习等等）。尽管取得突破性的进展，Word2Vec 依然存在明显缺陷：1) 学习词嵌入时仅考虑局部上下文窗口中的信息，忽略了词语间的长程相关性；2) 使用唯一的词向量表征词汇，导致无法很好的处理多义词。尽管在 BERT 模型提出后 Word2Vec 即被认为是“过时”的工具，但其仍具有相当的工程应用及研究借鉴价值。

24.5.1 文本向量化 (text vectorization) 与词嵌入 (word embedding) 基础

- 截至目前，所有的机器学习模型均无法如人类般真正理解文本。但正如人类的所有思维在底层上均是神经电信号和神经递质的传递，机器学习模型也可通过底层的数学运算在一定程度上学习人类语言的结构。而为进行数学运算需要提前将文本转换为数值张量，这一过程被称为文本向量化 (text vectorization)；
- 文本向量化实际上是对文本的基本单元作向量化。常考虑的文本基本单元包括单词、字符、和 n-gram。n-gram 是 n 个连续单词或字符的集合 (n-gram 间可重叠)。将上述基本单元统称为标记 (token)，而将文本划分为标记的过程被称为分词 (tokenization)；
- 为实现标记至向量的映射，一般考虑以下几种思路：
 - 独热编码 (one-hot encoding)：是将标记转换为向量的最常用、最基本的方法。它将每个标记与一个唯一的整数索引相关联，然后将这个整数索引 i 转换为长度为 N 的二进制向量 (N 是词表大小)，

⁷Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013). <https://arxiv.org/pdf/1301.3781.pdf>

这个向量只有第 i 个元素是 1，其余元素都为 0。**one-hot** 编码得到的向量是二进制的，且具有稀疏、高维、忽视标记间相关性的缺点；

- 标记嵌入 (**token embedding**)：通常只用于单词，叫作词嵌入 (**word embedding**)。词嵌入是目前最强大的文本向量化方法，其旨在将高维离散的词表嵌入至低维稠密的浮点数向量空间，并使得语义相近的词也具有相似的词向量。常见的词向量维度是 256、512 或 1024（one-hot 编码的词向量维度通常在 20000 以上）。基于文本数据学习词嵌入的过程又可分为两种方法：

- 在完成主任任务（比如文档分类或情感预测）的同时学习词嵌入。此时词向量为模型部分隐藏层的输出。得到的词嵌入可学习得适配于主任任务的语义信息，但无法保证词嵌入结果的可迁移性；
- 设计专门的机器学习模型预算算好词嵌入，再根据需求作为下游任务的输入，这些词嵌入被称为预训练词嵌入 (**pretrained word embedding**)。词向量同样为模型部分隐藏层的输出。其结果理论上可捕捉更全面的语义信息，具有较高的可迁移性。Word2Vec 即属于这一类型。

在 Word2Vec 提出之前，几乎不存在除独热编码以外的成熟的预训练词向量生成方案，且当时的神经网络模型也难以在 NLP 任务中充分捕捉文本的语义结构信息从而得到高质量的词嵌入。

24.5.2 Skip-Gram 模型

1. Skip-Gram 模型^{8,9}的目标是基于输入的词向量预测上下文内容。如果模型能准确预测上下文信息，意味着模型可较好地理解各类词语的语义信息，则其隐藏层即可被视为一种良好的词嵌入；
2. 直观地可将上下文预测任务建模为一个经典的多分类问题——给定一个包含所有单词的字典，输入一个单词，则模型输出字典中每个单词作为输入单词上下文的概率；
3. 首先准备训练集，每个训练样本为每个目标单词 (**target word**) 和一个相应上下文单词 (**context word**) 的配对。目标单词为分类模型的输入，而上下文单词则为理论的最优分类结果。训练样本由语料库 (**corpus**) 中的文本或句子生成。Skip-Gram 模型采用滑动窗口采样法 (**sliding window sampling**)。采样时涉及两个超参——*skip_window* 和 *num_skips*。*skip_window* 表示从目标单词左侧或右侧选词的数量。以句子 “The dog barked at the mailman” 为例，若以 “dog” 为目标单词，并设 *skip_window*=2，则 “dog” 两侧窗口中包含 “the”、“barked”、“at” 共三个单词。*num_skips* 表示从窗口中选取的上下文单词的数量，即样本数。若令 *num_skips*=2 则可得到两组训练样本如 (“dog”, “the”)、(“dog”, “barked”);
4. 基于训练集搭建神经网络模型进行多分类训练。模型在结构上非常简单，是一个仅包含一层线性全连接隐藏层的浅层神经网络。因为模型训练前并无良好的词嵌入方式，故模型的输入为单词的独热编码。记输入为 $x \in \mathbb{R}^V$ ，其中 V 表示单词表中的单词数目， x 中只有一个元素为 1，其它均为 0。模型的隐藏层通过线性变换（无激励函数）将 x 嵌入低维稠密隐空间

$$h = Wx, \quad h \in \mathbb{R}^N, \quad W \in \mathbb{R}^{N \times V}$$

上式中 h 即为 x 对应单词的词向量； N 为自定义的词向量维度； W 为全连接隐藏层权重。因为输入 x 为独热编码，记第 i 个元素 $x_i = 1$ ，则 Wx 实际上即是取 W 的第 i 列 w_i ，有 $h = Wx = w_i$ ，故也可称 w_i 为单词 i 的向量表示，这一过程被称为查表 (**lookup**)。模型的输出层将 h 重新映射回 \mathbb{R}^V 空间，并经 softmax 标准化得到词表中每个词作为 i 的上下文单词的概率

$$y = \text{softmax}(u) = \text{softmax}(W'h) \iff y_j = \frac{\exp\{u_j\}}{\sum_{j'} \exp\{u_{j'}\}} = \frac{\exp\{w'_j w_i\}}{\sum_{j'} \exp\{w'_{j'} w_i\}}, \quad y, u \in \mathbb{R}^V, \quad W' \in \mathbb{R}^{V \times N}$$

上式中 y_j 为输出向量的第 j 个元素，表示单词 j 作为 i 的上下文单词的概率； W' 为全连接输出层权重， w'_j 为其第 j 行向量。从隐藏层与输出层的对称关系不难发现， w_i, w'_i 均可作为单词 i 的向量表征，前者是作为目标单词时的嵌入结果，后者是作为上下文单词的嵌入结果，只是 word2vec 一般只考虑前者。这一特性与矩阵分解的向量理解一致，因此也揭示了经典词嵌入模型与矩阵分解的同源性；

⁸Skip-gram 模型：<https://ultipa.cn/document/ultipa-graph-analytics-algorithms/skip-gram/v4.3>

⁹Skip-gram 模型优化：<https://ultipa.cn/document/ultipa-graph-analytics-algorithms/skip-gram-optimization/v4.3>

5. 记每一单词均考虑了 C 个上下文单词生成训练样本，则可基于极大对数似然估计设计 Skip-Gram 模型针对单词 i 嵌入的目标函数

$$(\min) \quad \mathcal{L}_i = -\ln \prod_{c=1}^C y_{j_c} = -\sum_{c=1}^C \ln \frac{\exp\{w'_{j_c} w_i\}}{\sum_{j'} \exp\{w'_{j'} w_i\}} = -\sum_{c=1}^C w'_{j_c} w_i + C \cdot \ln \sum_{j'} \exp\{w'_{j'} w_i\}$$

式中 j_c 表示上下文单词 c 在词表的索引。可以看到上述目标函数由两项组成——第一项旨在最大化 $w'_{j_c} w_i$ ，即最大化目标单词 i 与上下文单词 c 嵌入结果的相关性，而第二项最小化 $w'_{j'} w_i$ ，即最小化目标单词 i 与其它无关单词嵌入结果的相关性。该形式目标函数也将在后续各领域嵌入任务（如图嵌入）中大量沿用。将 \mathcal{L}_i 扩展至词表中的所有单词即可得到原始的 Skip-Gram 模型的目标函数：

6. 上述的原始 Skip-Gram 模型可处理小规模的词嵌入问题，但在实际海量词表嵌入应用时则受计算量的困扰。因为随着单词数量和嵌入维数的增加 W, W' 的规模将变得非常庞大，而误差反向传播时 W, W' 的绝大多数参数又与目标词和上下文词无关，导致梯度下降过程噪声较大、更新缓慢。与此同时，输出层采用的 softmax 激励函数的归一化分母涉及了词汇表中的所有单词，导致目标函数 \mathcal{L}_i 中的绝大多数信息都是与 i 无关的单词，同样增加计算并引入噪声。为此一般在原始 Skip-Gram 模型的基础上引入二次采样 (subsampling) 和负采样 (negative sampling) 技术；
7. 二次采样的目标在于减少训练集规模，仅保留高价值的样本，从而减少计算量并提升学习效果。一般认为语料库中的高频词（如“the”、“and”、“is” 等等）语义价值越有限，训练样本中大量的高频词会造成“不平衡分类”问题，弱化其它语义更清晰的词语的信息。因此对于训练集中的单词按概率进行二次采样，适当丢弃高频词。二次采样中单词 i 被保留的概率按下式计算

$$P(i) = \left(\sqrt{\frac{f_i}{\alpha}} + 1 \right) \frac{\alpha}{f_i}$$

式中 f_i 表示单词 i 的频率； α 是影响概率分布的超参，默认为 0.001；

8. 负采样的目标是减少词嵌入目标函数中无关单词的干扰，其思想最初来源于 Noise-Contrastive Estimation 算法。在负采样时，每次采样一个上下文单词作为正样本时也将随机选择 K 个其它无关单词作为负样本，而其它无关的单词不再被考虑。负采样的基本原则是优先考虑语料库中的高频词，负采样分布可定义为

$$P_n(i) = \frac{f_i^{3/4}}{\sum_j f_j^{3/4}}$$

$P_n(\cdot)$ 中的下标“n”表示噪声，故 $P_n(\cdot)$ 也被称为噪声分布 (noise distribution)。研究表明，对于小型数据集 K 可取 5-20，而对于大型数据集 K 可减小至 2-5。加入负采样后再将模型输出层的激励函数改为 sigmoid（可理解为多分类改二分类，正样本应输出 1 而负样本应输出 0，且正负样本相互独立），即

$$y_j = \sigma(u_j) = \frac{1}{1 + \exp\{-w'_j w_i\}}$$

式中 y_j 同样理解为单词 j 作为 i 的上下文单词的概率。记单词 c 为 i 的上下文单词（正样本）。考虑了负采样后的 Skip-Gram 模型针对单词 i 嵌入的目标函数同样按极大对数似然法设计为

$$(\min) \quad \mathcal{L}_i = -\ln \left(y_{j_c} \prod_{k=1}^K (1 - y_{j_k}) \right) = -\ln \sigma(w'_{j_c} w_i) - \sum_{k=1}^K \ln (1 - \sigma(w'_{j_k} w_i)) = -\ln \sigma(w'_{j_c} w_i) - \sum_{k=1}^K \ln \sigma(-w'_{j_k} w_i)$$

上式中 j_c 表示上下文单词（正样本）的索引，而 j_k 表示其它无关单词（负样本）的索引。因为 \mathcal{L}_i 只和 w_i, w'_{j_c}, w'_{j_k} 有关，因此梯度反向传播时只有矩阵 W 的一个列向量和矩阵 W' 的 $K+1$ 个行向量需要更新，更新参数量大大减少，从而得到适用于计算实际大规模词嵌入的 Skip-Gram 模型。

24.5.3 CBOW (continuous bag of words) 模型

1. 与 Skip-Gram 模型相反，连续词袋 (continuous bag of words, CBOW) 模型¹⁰的目标是基于上下文内容预测中心词，模型训练后的隐藏层即为最终的词嵌入。“词袋 (bag-of-words)”指由句子所有 token

¹⁰word2vec 详解 (CBOW, skip-gram, 负采样, 分层 Softmax): <https://zhuanlan.zhihu.com/p/53425736>

组成但忽略前后顺序的集合，是深度学习模型成熟前 NLP 领域的主流特征工程工具，适用于轻量级机器学习模型。CBOW 模型的建模思路和模型结构与 Skip-Gram 均高度相似，下文将重点介绍其特性；

2. 原始的 CBOW 模型同样将中心词预测任务建模为多分类问题。模型的输出为字典中每个词作为中心词的概率，但不之处在于模型的输入为多个前后文单词的独热向量。记字典中的单词数为 V ，词嵌入的维度为 N ，每个样本中包含 C 个前后文单词，则模型隐藏层聚合每个单词的嵌入信息

$$h = \frac{1}{C} \sum_{c=1}^C Wx_c = \frac{1}{C} \sum_{c=1}^C w_c, \quad h, w_c \in \mathbb{R}^N, \quad x_c \in \mathbb{R}^V, \quad W \in \mathbb{R}^{N \times V}$$

上式中 x_c 为单词 c 的独热编码； W 为全连接隐藏层的参数矩阵； w_c 为 W 的第 c 列向量，因为 x_c 为独热编码，则 $w_c = Wx_c$ 即为单词 c 的词嵌入；集计隐藏层状态 h 时不考虑前后文单词间的前后顺序，这便是“词袋”的由来。模型的输出层将 h 重新映射回 \mathbb{R}^V 空间，并经 softmax 标准化得到词表中每个词作为中心词的概率

$$y = \text{softmax}(u) = \text{softmax}(W'h) \iff y_j = \frac{\exp\{u_j\}}{\sum_{j'} \exp\{u_{j'}\}} = \frac{\exp\{w'_j h\}}{\sum_{j'} \exp\{w'_{j'} h\}}, \quad y, u \in \mathbb{R}^V, \quad W' \in \mathbb{R}^{V \times N}$$

式中 y_j 为输出向量的第 j 个元素，表示单词 j 作为中心词的概率； W' 为全连接输出层权重， w'_j 为其第 j 行向量。上式与原始 Skip-Gram 的输出层数学形式几乎一致，唯一的差别即在于隐藏层状态 h 的定义不同。令 j 为中心词，则参考原始 Skip-Gram 的目标函数给出原始 CBOW 模型目标函数如下

$$(\min) \quad \mathcal{L} = -w'_j h + \ln \sum_{j'} \exp\{w'_{j'} h\} = -\frac{1}{C} \sum_{c=1}^C w'_j w_c + \ln \sum_{j'} \exp \left\{ \frac{1}{C} \sum_{c=1}^C w'_{j'} w_c \right\}$$

尽管数学形式相似，仍可看出 CBOW 模型目标函数反向传播时与 Skip-Gram 的不同——对于一条样本，CBOW 每次可学习 C 个单词的词嵌入信息，而 Skip-Gram 每次仅能学习一条词嵌入；

3. 同样可在原始 CBOW 模型的基础上引入二次采样 (subsampling) 和负采样 (negative sampling) 技术进一步提升模型训练效率。二次采样与降采样的实施方式与 Skip-Gram 模型一致。加入负采样后同样将模型输出层的激励函数改为 sigmoid

$$y_j = \sigma(u_j) = \frac{1}{1 + \exp\{-w'_j h\}}$$

式中 y_j 同样理解为单词 j 作为中心词的概率。记负样本数目为 K ，进一步得到考虑负采样后的 CBOW 模型的目标函数

$$(\min) \quad \mathcal{L} = -\ln \sigma(w'_j h) - \sum_{k=1}^K \ln \sigma(-w'_{j_k} h)$$

对上述目标函数的解释也与 Skip-Gram 模型一致；

4. 因为 CBOW 模型每次可学习多个词的嵌入信息，而 Skip-Gram 每次仅聚焦一个单词，导致了两者在实际应用层面的差别：

- CBOW 模型训练效率更高，更适用于大型数据集。CBOW 的时间复杂度与上下文单词的数目无关，仅与训练集大小有关，为 $O(V)$ ；而 Skip-Gram 模型则为 $O(C \cdot V)$ ；
- Skip-Gram 模型对生僻词的学习效果更高。Skip-Gram 每次仅聚焦一个单词的语义信息，当考虑多个上下文单词时会多次更新同个单词的词嵌入，故而对词语义信息的理解更充分；而 CBOW 模型学习时误差梯度会被所有上下文单词分担，噪声较大。

24.6 Seq2Seq 模型

1. 顾名思义，seq2seq 模型是一种基于序列生成序列的模型，于 2014 年由 Google Brain 团队和 Yoshua Bengio 团队独立提出，主要面向自然语言处理中的文本生成和翻译问题，并适用于任意序列预测问题¹¹。在后续 Transformer 模型（见第 24.7.1 节）出现前，seq2seq 是自然语言处理领域最主流的模型；

¹¹清晰理解 Seq2seq: <https://zhuanlan.zhihu.com/p/714326774>

2. 在 seq2seq 模型出现之前, LSTM 等 RNN 变体 (见第 24.2 节) 便已被广泛应用于时间序列等各类序列建模。但语言生成和翻译任务相比其它传统序列建模任务的一个重要难点在于输入序列和输出序列的长度均不固定。单独的 RNN 结构可处理模型输入长度不固定的问题, 但对于输出长度不固定则无能为力。seq2seq 模型通过引入两个 RNN 模型分别作为编码器 (学习输入序列) 和解码器 (生成输出序列) 成功解决这一问题。因此尽管基于 RNN 架构的 seq2seq 模型如今已不再是主流, 但其采用的解码器-编码器架构仍被 Transformer 等方法继承;
3. 本节主要介绍 seq2seq 模型的整体结构。编码器模型与一般的 RNN 变体模型无异, 多采用单层或多层的 LSTM 和 GRU 模型, 以语料序列作为输入。解码器以编码器的最终输出作为初始隐状态。在训练阶段, 解码器的输入为后移一个单位的目标序列。具体地, 记目标序列为 (y_1, y_2, \dots, y_n) , 则解码器的输入-输出数据对依次为 $(\langle \text{start} \rangle, \hat{y}_1), (\hat{y}_1, \hat{y}_2), \dots, (\hat{y}_{n-1}, \hat{y}_n), (\hat{y}_n, \langle \text{end} \rangle)$, 其中 $\langle \text{start} \rangle$ 和 $\langle \text{end} \rangle$ 为起止符, 该过程被称为 teacher forcing。基于交叉熵 (见第 23.9.6 节) 建模训练损失函数

$$J = -\ln p(\hat{y}_1) - \ln p(\hat{y}_2) - \dots - \ln p(\hat{y}_n) - \ln p(\langle \text{end} \rangle)$$

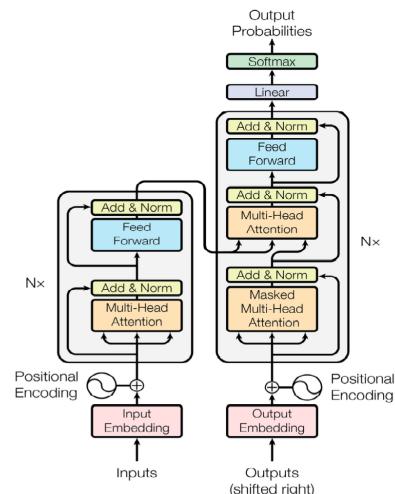
在测试阶段, 解码器以自身上一阶段的输出为输入, 输入-输出数据对依次为 $(\langle \text{start} \rangle, \hat{y}_1), (\hat{y}_1, \hat{y}_2), \dots$, 直至输出结束符 $\langle \text{end} \rangle$, 该过程被称为 free running。理论上解码器在训练阶段也可按 free running 模式生成输出序列。但因为初始状态下该模式将造成大量误差累计, 不利于梯度学习, 故采用 teacher forcing 提升学习效率;

4. 然而解码器在训练和测试阶段的输入不同又可能导致过拟合问题, 使得模型在测试阶段因误差累积效果直线下降。为此进一步提出计划采样 (scheduled sampling) 训练方法。即要求解码器在训练阶段以概率 p 基于上一步输出进行预测, 而以概率 $1 - p$ 基于真实数据进行预测, 并令 p 随训练由 0 升至 1。

24.7 Transformer 系列

24.7.1 标准 Transformer 模型

图 24.9 标准 Transformer 模型结构。由编码器和解码器组成, 解码器和编码器各包含若干个串联的 Transformer 模块。经典的 seq2seq 模型以 RNN 为基础, 在训练时解码器基于编码器的输入预测输出。而 Transformer 则以自注意力机制为基础, 且训练时解码器的输入除编码器结果外还包括目标输出。以经典的翻译任务为例, 假设编码器的输入源文本为“我爱中国”, 则解码器基于编码器的结果和目标文本“I love China”为输入, 仍输出“I love China”。



- Transformer 模型由 Google 机器翻译团队于 2017 年与自注意力机制同步提出 (见第 24.4 节)。提出后迅速取代了经典 seq2seq 模型在生成式自然语言模型中的垄断地位, 并针对视觉任务发展出了诸多变体, 奠定了多模态大语言模型的基础¹²;
- 模型采用编码器-解码器结构。解码器和编码器由若干个 Transformer 模块串联组成 (一般为 6 个)。一个 Transformer 编码模块由多头自注意力层和前向神经网络组成, 而一个 Transformer 解码模块则包含掩码多头自注意力层、解码多头自注意力层和前向神经网络。下游编码模块的输入为上游编码模块的输出, 而下游解码模块的输入为上游解码模块的输出和编码器的整体编码结果;

¹²Transformer 模型详解: <https://zhuanlan.zhihu.com/p/681604237>

3. 首先介绍 Transformer 编码模块。记编码模块的输入为 X , 其行向量 x_i 表示源文本的第 i 个 token 的于文本中的向量表示, 由 token 的词向量 (token embedding) \hat{x}_i 和位置编码 (position embedding) p_i 相加而成。其中词向量可采用 word2vec (见第 24.5 节) 等预训练模型的结果, 也可随机初始化。位置编码按如下经验公式生成

$$p_{ij} = \begin{cases} \sin\left(\frac{i}{10000^{2k/d}}\right) & j = 2k \\ \cos\left(\frac{i}{10000^{2k/d}}\right) & j = 2k + 1 \end{cases}$$

式中 p_{ij} 表示位置编码向量 p_i 的第 j 个元素, d 为 \hat{x}_i, p_i 的维度。引入位置编码的原因是因为多义词的影响, token 的语义需要在特定的语序环境下才能确定, 而 Transformer 模型不含 RNN 结构, 无法识别输入向量的前后关系, 故需要在向量编码中显式引入位置信息。 X 先后经多头自注意力层和前馈神经网络, 并引入残差结构和标准化层后得到单个 Transformer 编码模块的输出。经多个编码模块编码, 最终得到源文本的编码结果 C ;

4. 进一步介绍 Transformer 解码模块。对于目标文本, 首先在其前、后端分别插入起始符 “</s>” 和终止符 “</e>”, 考虑包括起始符、终止符在内的所有 token 词向量及位置编码, 得到目标文本的向量化表示 Y 作为解码模块的输入。插入起始符的目的是为了在训练后的推理阶段解码器可基于起始符和编码器结果生成后续内容, 而终止符则有助于模型自主判定生成结束。解码模块的核心为掩码多头自注意力层和解码多头自注意力层:

- Y 首先经掩码多头自注意力层 (masked-multi-head self-attention) 提取特征信息 Z 。所谓“掩码”即是在 Q, K, V 计算完成后对基于 Q, K 计算的自注意力矩阵 A 取掩码。矩阵 A 的元素 a_{ij} 表示目标文本中 token*i* 对 token*j* 的注意力。为使解码器仅基于上文内容预测下文, 应使 A 为下三角矩阵, 即当 $j > i$ 时令 $a_{ij} = 0$, 同时仍保证 A 的行向量和为 1;
- 解码多头自注意力层 (encoder-decoder multi-head self-attention) 旨在融合编码器与掩码多头自注意力层的结果。不同于传统的自注意力计算方法, Y 经掩码多头自注意力层提取的特征信息 Z 仅用于计算解码多头自注意力层 Q 矩阵, 而编码器的编码结果 C 则用于计算解码多头自注意力层 K, V 矩阵。由此计算的注意力可表征源文本编码结果与目标文本的相关关系, 从而用于目标文本生成。这一机制被称为交叉注意力 (cross attention)。

解码模块同样存在前馈神经网络、残差结构和标准化层。经若干解码模块堆叠后最终由 softmax 激励函数预测字典库中各词于目标文本各位置的概率;

5. 最后介绍模型训练完成后的推理过程。编码器仍以源文本为输入得到文本嵌入结果, 而解码器则仅基于起始符 “</s>” 和编码结果预测首个 token, 拼接起始符与首个预测 token 再次输入解码器预测下一个 token, 循环往复直至预测出终止符 “</e>” 为止表示文本生成结束。

24.7.2 BERT (bidirectional encoder representations from transformer) 模型

1. 在 Transformer (第 24.7.1 节) 的基础上, 谷歌团队进一步于 2018 年提出了著名的 BERT 模型¹³, 直接刷新了多个 NLP 领域的任务, 并开启了 NLP 领域的预训练-微调 (pretrain+fine-tune) 范式, 在此之前大规模预训练不仅仅只在 CV 中 (如 Yolo 等)。BERT 模型在 NLP 中取得的成功标志着 NLP 进入了深度模型时代;
2. 顾名思义, BERT 模型以 Transformer 模型的编码器为基础的表征学习模型, 其输出为输入文本的逐 token 向量表征。模型与 Transformer 编码器的差异主要为:
 - BERT 模型较标准 Transformer 编码器更深。标准 Transformer 编码器由 6 个编码模块串联而成, 而基准 BERT 模型采用 12 个 Transformer 编码模块, 后续又进一步扩展至 24 个编码模块, 成为名副其实的“深度模型”;

¹³Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018). <https://arxiv.org/abs/1810.04805>

- BERT 模型的输入样本固定为由两个句子组成的句子对，Transformer 编码器的输入文本则无限制；
- 标准 Transformer 编码器的输入向量为 token 词向量和位置编码向量相加而成的文本表征向量，而 BERT 模型的输入则是在上述文本表征向量的基础上再加上段表征向量 (**segment embedding**)。段向量用于刻画文本的全局语义信息，每段文本对应一个段向量，其取值在模型训练过程中自动学习。另外 BERT 模型的每句输入文本还在起、止位置插入起始符和终止符，与 Transformer 解码器一致。

3. BERT 模型的核心在于预训练，基于无监督学习范式学习大规模、与特定 NLP 任务无关的文本语料信息。文章作者设计了预训练阶段的两类无监督学习任务——掩码语言模型 (*masked language model*) 和下句预测 (*next sentence prediction*)：

- 掩码语言模型任务指随机抹去输入源文本的若干个 **token**，并要求 BERT 模型对此进行预测补全。具体地，作者随机选择 15% 的词汇用于预测，其中 80% 采用掩码替换，10% 采用任意其余 token 替换，剩余 10% 情况下保持原 token 不变。由此使得模型并不知道输入对应位置的 token 是否正确，从而迫使模型更多地依赖于上下文信息去预测 token 信息。因为自注意力机制下 BERT 模型可基于上下文信息预测目标 **token**，而非传统的 RNN 结构只能基于前文单向预测后文，这便是模型名称中“B”（双向）的来源；
- 下句预测任务是在掩码语言模型的基础上预测语料库中随机选择的两句话是否是前后句关系，是一个典型的二分类问题。每一句的起始符的对应输出向量被认为可表征句子的全局信息，故基于其进行下句预测。需要说明的是，该任务后被证明对于模型来说并没有太大的作用，可能是因为 BERT 学习能力太强，同时这个任务的难度较低，所以后续以 BERT 为基础的改进模型都将这个功能去掉了，要么直接不用类似的任务，要么使用这个任务的改进版——语句顺序预测 (*sentence order prediction*)。

BERT 模型通过对掩码语言模型任务和下句预测任务进行联合预训练，使模型输出的每个字/词的向量表示都能尽可能全面、准确地刻画输入文本（单句或语句对）的整体信息，为后续的微调任务提供更好的模型参数初始值；

4. 在完成预训练任务后再根据下游任务对模型进行微调。具体地以预训练模型参数进行初始化，再根据具体任务需求在 BERT 模型表征学习结果的基础上增加相应的前馈结构。

24.7.3 sentence-BERT 模型

1. BERT 模型（见第 24.7.2 节）的成功迅速启发了大量语言模型研究，在预训练 BERT 模型的基础上进一步针对下游任务提出了一系列 BERT 变体模型。2019 年提出的 **sentence-BERT** 模型即是聚焦于句子整体的向量化表征任务，构建句子的向量化编码以反映不同语句语义信息的相似度，是现阶段最具代表性的句向量嵌入模型之一¹⁴；
2. 预训练 BERT 模型可输出特定语境中每个 token 的表征结果，理论上基于该结果可直接得到句子整体的向量表征，常见的思路包括：
 - 以句子起始符的表征向量作为句子整体的表征向量，BERT 模型预训练时即是基于该信息预测句子间的上下文关系；
 - 均值池化：使用句子所有 token 的表征向量的均值作为句子整体的表征向量；
 - 最大值池化：提取句子所有 token 的表征向量各维度的最大值作为句子整体的表征向量。
 然而实验结果表明在不对 BERT 模型进行微调的情况下按上述方法得到的句向量并不能有效反映句子的语义信息，根源在于 BERT 模型的预训练任务更多聚焦于 **token** 级别构建，其学习结果并不能很好地反映句子的整体信息；
3. 为此作者基于孪生网络 (*siamese network*) 结构设计了面向句向量嵌入的 sentence-BERT 模型。不同于原生 BERT 模型以包含两个句子的样本作为输入，孪生网络结构下两个句子并行输入两个共享权重的 BERT 模型，使得模型专注于单个句子的特征。需要说明的是，根据具体需求孪生 BERT 网络也可包括三个及以上的共享权重的 BERT 模型；

¹⁴Nils Reimers, Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (2019). <https://arxiv.org/abs/1908.10084>

4. 记样本 i 中的句子 A_j^i 输入 BERT 模型后由均值池化得到的句向量为 u_j^i （也可选择最大值池化或起始符表征向量）。作者进一步根据不同的训练集类型提出了三种训练 sentence-BERT 模型的目标函数：

分类目标函数 若训练集提供了具有不同关系（如矛盾关系、推论关系、无关系等）的句子对样本，则 sentence-BERT 模型训练可建模为多分类问题，模型输出句子对 A_1^i, A_2^i 所属关系的概率 σ^i ，并基于交叉熵损失优化

$$\sigma^i = \text{softmax} (W [u_1^i; u_2^i; |u_1^i - u_2^i|])$$

式中 W 表示输出层的权重； $[u_1^i; u_2^i; |u_1^i - u_2^i|]$ 表示 $u_1^i, u_2^i, |u_1^i - u_2^i|$ 的拼接向量；

回归目标函数 若训练集提供了多个级别相关性的句子对样本，则 sentence-BERT 模型训练可建模为回归问题，直接计算 u_1^i, u_2^i 间的余弦相似度以表征句子对 A_1^i, A_2^i 的相关性，并基于均方误差损失优化；

三元组目标函数 若训练集仅提供相关的和不相关的句子对样本时，除建模为二分类问题外还可参考语言模型中常用的负采样技术（如第 24.5 节中 word2vec 的 Skip-Gram 和 CBOW 模型），选择相关的句子对 A_1^i, A_2^i 作为正样本，同时随机抽取无关的句子 A_3^i 作为负样本，对应的 BERT 模型句向量分别为 u_1^i, u_2^i, u_3^i ，并最小化如下目标函数

$$\max \{ \|u_1^i - u_2^i\| - \|u_1^i - u_3^i\| + \varepsilon, 0\}$$

上式中 ε 为超参（论文中设为 1）， $\|\cdot\|$ 为距离度量。该目标函数要求正样本 A_2^i 较 A_1^i 的嵌入差异至少应该比负样本 A_3^i 较 A_1^i 的嵌入差异小 ε 。

5. 在预训练 BERT 模型的基础上，选择适当的微调任务即可得到可有效识别句子整体语言并生成通用表示向量的 sentence-BERT 模型。

24.7.4 GPT-1 (generative pre-training) 模型

1. 2018 年 Google 团队基于 Transformer 的编码器模型提出了大名鼎鼎的 BERT 模型（见第 24.7.2 节），同年早些时候 OpenAI 团队则基于 Transformer 的解码器模型提出了初代 GPT (generative pre-training) 模型¹⁵，顾名思义 GPT-1 模型是一个生成式语言模型，与 BERT 同遵循预训练-微调的应用范式；
2. **因为 BERT 模型基于 Transformer 编码器设计，故模型可基于上下文信息预测中心词；而 GPT-1 模型基于 Transformer 解码器，只能按标准语言模型基于上文预测下文。**因此在当时学界普遍认为 GPT-1 模型更适用于自然语言生成 (nature language generation, NLG) 任务，而 BERT 模型更适用于自然语言理解 (nature language understanding, NLU) 任务。并在一系列其它因素的影响下，学界普遍认为 BERT 模型较 GPT-1 具有更大的开发潜力。直至 OpenAI 在 GPT-1 的基础上不断迭代提出标志性的 GPT-3.5 (即 ChatGPT) 模型，颠覆了 BERT 模型开启的基于预训练-微调的学习范式，引导自然语言处理领域进入通用大语言模型时代，GPT-1 模型才重回学界视野，并引发学界对原有认识的一系列反思；
3. 首先介绍 GPT-1 模型的结构。模型的输入与 Transformer 模型一致——由文本各 token 的词向量和位置编码向量相加而成（BERT 的输入还额外考虑了反映文本全局特征的段向量）。记模型的视野为 k ，即基于前 k 个 token $U = (u_{-k}, \dots, u_{-1})$ 预测下一个 token u ；又记模型包含 n 个 Transformer 解码模块（GPT-1 模型与基准 BERT 模型相同均由 $n = 12$ 个 Transformer 模块组成），有

$$P(u) = \text{softmax}(h_n W), \quad h_l = \text{TB}_l(h_{l-1}), \quad h_0 = UW_e + PW_p, \quad \forall l \in \{1, \dots, n\}$$

式中 W_e, W_p 分别表示词向量和位置向量编码矩阵，不同于标准 Transformer 采用确定性的位置编码方法，**GPT-1** 模型的位置编码为随机初始化并随训练自动学习； h_0 为输入模型的文本向量表征； $\text{TB}_l(\cdot)$ 表示第 l 个 Transformer 解码模块，其输出为 h_l ；最后以一个 Transformer 解码模块的输出 h_n 经线性层和 softmax 激励函数后计算下一个 token u 的概率。**GPT-1** 模型的 Transformer 解码模块与标准 Transformer 模型的解码模块不完全相同，因前者不含 Transformer 编码器，故去除了标准 Transformer 解码模块中与编码器相关的部分——解码多头自注意力层，仅包含掩码多头自注意力层和前馈神经网络两部分；

¹⁵Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. Improving Language Understanding by Generative Pre-Training (2018): https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

4. 进一步介绍模型的预训练过程。模型的预训练目标函数与标准语言模型一致。记语料库为无标签 token 序列 $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$, 则模型的预训练目标是最大化如下对数似然函数

$$(\max) \quad L_1(\mathcal{U}; \theta) = \sum_i \ln P(u_i | u_{i-k}, \dots, u_{i-1}; \theta) \quad (\text{GPT-1 预训练})$$

式中 θ 表示模型权重参数;

5. 根据下游任务的需求, 模型的微调过程一般为监督学习任务。记模型的一条输入样本为 $\{u_1, u_2, \dots, u_k\}$, 对应标签为 y , 则首先搭建针对性的输出层如下

$$P(y|u_1, \dots, u_k; \theta) = \text{softmax}(h_{nk} W_y)$$

式中 h_{nk} 为最后一个 Transformer 解码模块输出 h_n 中对应于 u_k 的分量, 从而设计如下微调目标函数

$$(\max) \quad L_3(\mathcal{U}; \theta) = L_2(\mathcal{U}; \theta) + \lambda L_1(\mathcal{U}; \theta), \quad L_2(\mathcal{U}; \theta) = \sum_{(u,y)} \ln P(y|u_1, \dots, u_k; \theta) \quad (\text{GPT-1 微调})$$

文中上式 $\lambda = 0.5$, 微调目标函数中仍保留语言模型 $L_1(\mathcal{U}; \theta)$ 有助于提升监督模型的泛化能力并加快收敛;

6. 需要说明的是, 与 sentence-BERT 模型 (见第 24.7.3 节) 一致, 针对不同类型的下游任务, 需要针对性地调整模型的输入输出形式:

- 对于分类任务, 以单个句子作为样本, 并在句子起、止处插入起始符和结束符。后续模型结构和输出层结构与上文一致;
- 对于句子关系任务, 在与 BERT 模型类似以两个句子的拼接作为样本, 在句子拼接处插入分隔符, 而在样本止处插入起始符和结束符。后续模型结构和输出层结构与上文一致;
- 对于文本相似性任务, 与句子关系任务类似, 但需考虑 “Text1-Text2” 和 “Text2-Text1” 两种拼接方式, 保证相似性的对称性。因此构造与 sentence-BERT 模型一致的孪生网络结构, 将 “Text1-Text2” 和 “Text2-Text1” 两样本并行输入两个共享参数的 GPT-1 模型, 将两个样本对应的表示 h_{nk} 相加再由全连接网络计算相似性;
- 对于多项选择任务, 以问答为例, 给定问题 Q 要求在若干候选回答 $\{A_1, \dots, A_N\}$ 中选择最合适的回答。考虑从 $Q - A_1$ 到 $Q - A_N$ 的 N 种拼接方式, 构造包含 N 个共享参数的 GPT-1 模型, 并行输入 N 个样本后对 N 个输出结果再作 softmax 计算概率。

24.8 通用大语言模型

24.8.1 GPT-2/GPT-3

GPT-2 与零样本学习 (zero-shot learning)

1. 在 2018 年 Google 团队基于 Transformer 编码器架构提出 BERT 模型 (第 24.7.2 节) 后, 大量 NLP 领域相关研究便转向 BERT 技术路线, 而同年稍早前由 OpenAI 团队基于 Transformer 解码器架构提出的 GPT-1 模型 (第 24.7.4 节) 则遭到冷落。OpenAI 团队为证明 GPT 架构不逊于 BERT, 基于更大规模数据训练了一个更大规模的模型, 但较 BERT 仍无明显优势, 随即将模型卖点转向零样本预测 (zero-shot), 于 2019 年提出 GPT-2 模型¹⁶, 标志着通用大语言模型的开端;
2. GPT-1 和 BERT 模型的相继提出开启了 NLP 领域的“预训练-微调”范式, 即首先基于大量无标签数据预训练模型, 再根据下游特定任务需求进行有监督的预训练。然而该范式存在以下缺陷: 1) 对于不同的下游任务需训练不同的模型; 2) 微调时需要带标签数据; 3) 大量研究表明模型完成微调后将丧失对分布外样本 (**out-of-distribution, OOD**) 的泛化能力, 即零样本预测能力;
3. 人工智能研究以通用人工智能为根本目标, 即训练一个可应对多种任务、具有极强泛化能力的单一智能模型。然而因为模型学习能力的限制该目标长期以来显得遥遥无期。研究者已习惯针对各类任务设计不

¹⁶Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. <https://d4mucfpksyv.wv.cloudfront.net/better-language-models/language-models.pdf>

同结构的模型，以适应现实任务的多样性与复杂性。**自然语言的通用性使得通用的语言模型成为可能。**语言模型一般被建模为关于输入-输出文本的概率模型 $P(\text{output} | \text{input})$ 。以往的语言模型仅针对特定任务，即 $P_{\text{task}}(\text{output} | \text{input})$ 。而构建通用的语言模型，即要求构建 $P(\text{output} | \text{input}, \text{task})$ 。因为模型输入和模型任务均可由自然语言描述，故理论上 $P(\text{output} | \text{input}, \text{task})$ 与 $P(\text{output} | \text{input})$ 是一致的；

4. 基于该思路 GPT-2 模型应运而生。相比于包含 12 个 Transformer 解码层的 GPT-1 模型，GPT-2 分别使用了 24、36、48 层，在细节层面则无过多创新。在数据层面，模型采用了更大规模的数据，并为了确保模型对多任务的通用性将具体任务引入训练集中。例如翻译任务的训练样本为“（翻译为法语，中文文本，法语文本）”，而阅读理解任务的样本为“（回答问题，阅读文本，文本，答案）”。其中用于描述任务的文本便是日后大名鼎鼎的提示词 (**prompt**)。理论上在经过海量包含多任务的文本预训练后，只需提供任务文本和输入文本，模型即可直接生成合理输出，从而实现零样本预测。

GPT-3 与上下文学习 (**in-context learning**)

1. GPT-2 模型尽管在无样本预测层面取得可喜的结果，其效果仍无法与经监督学习微调的语言模型相比。然而模型的成功已经从概念上支撑了构建无微调通用语言模型的可行性，并且在测试中发现模型效果与模型规模间存在对数线性关系，意味着可以通过构造更大规模的模型取得更好的无监督学习效果。与此同时，当模型超过一定规模后传统的监督学习微调将不可避免地面临模型过拟合的风险，而 GPT-2 模型坚持的无微调技术路线可能使模型具有更高的鲁棒性；
2. 因此，OpenAI 团队坚持 GPT-2 模型的基本架构和训练方法，于 2020 年进一步推出 GPT-3 模型¹⁷。GPT-3 模型的参数量较 GPT-2 增加了两个量级，参与预训练的样本规模也增加了一个量级。然而前者相较于后者最关键的调整在于**放弃了对零样本预测的坚持，而是转向少样本学习 (few-shot)**。所谓少样本学习即是在提示词中除任务描述文本外还引入少量的带标签任务样本。以翻译任务为例，零样本学习的提示词仅包括翻译的语言类型，随后便直接要求模型翻译输入样本，而少样本学习还会在提示词中额外提供若干个单词的翻译示例。根据提供的带标签样本数目，少样本学习还可进一步分为单样本学习 (**one-shot**) 和少样本学习。零样本、单样本和少样本学习统称上下文学习 (**in-context learning**)。尽管设计带标签样本，少样本学习与传统监督学习仍具有明显差异：

- 传统范式中监督学习用于微调模型参数，而少样本学习仅是将样本加入提示词，预训练后参数不变；
- 模型监督学习微调时需要较多带标签样本避免过拟合，而少样本学习仅需极少量的标签样本。

GPT-3 模型选择少样本学习而非零样本学习或监督学习符合直觉——监督学习需要大量带标签样本，而人类往往仅需少量示范即可快速掌握下游任务；零样本学习完全不提供下游任务示范样本信息，同样与人类的学习模式不符；

3. 模型在细节层面也较 GPT-2 作了若干改进以适应更大的参数规模。其中最主要的调整在于将 Transformer 解码层中的自注意力机制改为稀疏自注意力 (**sparse attention**)。传统的自注意力机制属于稠密注意力 (**dense attention**)，即每个 token 编码需要与其它所有 token 编码两两计算注意力，时间复杂度为 $O(n^2)$ ；而稀疏注意力需引入超参 k ，每个 token 编码仅需与相对距离不超过 k 和相对距离为 $2k, 3k, \dots$ 的 token 编码计算注意力，时间复杂度为 $O(n \ln n)$ 。由此稀疏注意力可减少模型复杂度，并聚焦于较近的上下文关系，而对于较远的上下文保持稀疏关注；
4. GPT-3 模型的测试结果指出，随着模型规模的增加上下文学习范式的效益逐渐显现，模型可无需微调阶段而掌握各类任务知识，甚至具备一定的自主学习能力，能够简单处理没有见过的任务和识别一些简单的模式；且与零样本和单样本学习相比，少样本学习可使模型具有更高的预测效果，在给出足够的样例后可超过以往微调后的主流模型。至此，GPT-3 模型完全冲击了传统的预训练-微调学习范式并动摇了 BERT 模型在 NLP 领域的统治地位，通用大语言模型的时代即将到来。

¹⁷Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901: <https://arxiv.org/pdf/2005.14165.pdf>

24.8.2 GPT-3.5 (InstructGPT) 与基于人类反馈的强化学习 (reinforcement learning from human feedback, RLHF)

1. 尽管 GPT-3 模型已经表现出较为出色的语言理解和多任务适应能力，但因为自然语言的灵活性，语料库中不可能包含所有形式的语言表达，使得模型难以理解不属于 GPT-3 范式的人类指令。另外因为完全抛弃监督学习微调，使得模型尽管具有流畅的语言表达能力却难以判断其生成答案相对于提问的质量；
2. 为此，OpenAI 团队重新将监督学习引入大语言模型训练，具体包括有监督微调 (supervised fine-tuning, SFT) 和偏好对齐 (preference alignment) 两个阶段，从而得到 GPT-3.5 模型（也称 InstructGPT），为 ChatGPT 的基础¹⁸。至此无监督预训练-监督微调-偏好对齐便成为大语言模型训练的通用范式。需要说明的是，模型引入的有监督学习不同于传统语言模型训练中的监督微调，后者的目的在于训练模型实现特定的下游任务，而前者旨在进一步提升模型的多任务学习效果。
3. 有监督微调的过程相对直观。为进一步提升 GPT-3 模型对不同人类指令的理解能力和对生成内容质量的评估能力，需要额外构建带标签的优质 $\langle \text{prompt}, \text{answer} \rangle$ 数据集。具体地从测试用户提交的 prompt 中随机抽取一批，然后请专业的标注人员为这些 prompt 给出高质量答案，进而引导 GPT-3 模型生成与标注答案相似的文本；
4. 受标注成本的限制，团队不可能构建满足大语言模型训练需求规模的高质量标签语料库，为此团队创造性地提出了偏好对齐方法以进一步提升模型性能。所谓“偏好对齐”，顾名思义即是使得模型在各指令下生成的文本符合人类的偏好，为此便需要包含人类偏好的标签数据。同样因为标注成本的限制，要求使用尽可能少的偏好数据，为此研究基于强化学习进行偏好对齐。强化学习无需标签数据集，由奖励函数作为伪标签引导模型学习最优策略，而先前构造的偏好数据集则用于构造奖励函数，这一过程被称为基于人类反馈的强化学习 (reinforcement learning from human feedback, RLHF)，是首个成熟的偏好对齐算法。需要说明的是，RLHF 算法早在大语言模型发展前便被提出，2017 年谷歌¹⁹提出该算法以通过人工反馈提升强化学习在模拟机器人等任务中的效果；
5. 首先介绍偏好数据集的构建。随机抽样一批 prompt 指令集，基于微调后的 GPT-3 模型针对每一 prompt 生成 K ($4 \leq K \leq 9$) 个回答并两两配对得到 C_K^2 组结果。标注人员根据相关性、信息性和有害信息等标准，区分每一组样本两个回答的优劣。记 prompt 为 x ，标注人员偏好的回答为 y_w ，标注人员不偏好的回答为 y_l ，则得到偏好数据集 $\{(x, y_w, y_l)\}$ ；
6. 进一步介绍奖励函数。方法并不显式地设计奖励函数形式，而是训练一个奖励模型 (reward model, RM) 进行拟合。模型以不包含输出层的微调后 GPT-3 模型作为初始化，以 (x, y) 数据对（即 $\langle \text{prompt}, \text{answer} \rangle$ 数据对）为输入，输出反映答案质量的评分 $r_\theta(x, y)$ 。其中 θ 为模型参数。奖励函数训练的目标函数为

$$(\min) \quad L(\theta) = -\frac{1}{C_K^2} \mathbb{E}_{(x, y_w, y_l) \in D} [\ln \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))] \quad (\text{奖励模型训练目标})$$

式中 D 表示批训练集， $\sigma(\cdot)$ 为 sigmoid 激励函数。上述目标旨在最大化标注人员偏好与不偏好的回答评分的差值，从而使奖励模型学习人类偏好；

7. 最后介绍强化学习微调过程。研究最主要的贡献即在于将强化学习算法引入大语言模型预训练。所采用的强化学习算法借鉴了该团队于 2017 年提出的 PPO 算法。记强化学习微调的 GPT-3 模型参数为 ϕ ，则目标函数为

$$(\max) \quad O(\phi) = \mathbb{E}_{(x, y) \in D_{\pi_\phi}} \left[r_\theta(x, y) - \beta \ln \frac{\pi_\phi(y|x)}{\pi^{\text{SFT}}(y|x)} \right] + \gamma \mathbb{E}_{x \in D_{\text{pretrain}}} [\ln \pi_\phi(x)] \quad (\text{强化学习微调目标})$$

上式由三项构成， β, γ 为算法超参。前两项源于经典 PPO 算法，第一项旨在最大化模型输出结果的偏好评分，第二项的统计意义为强化学习微调模型输出与监督微调模型输出的 KL 散度。强化学习过程中模型输出可能因参数更新而显著改变，而奖励模型是基于仅监督微调的模型输出训练的。引入上式第二项

¹⁸Ouyang, Long, et al. "Training language models to follow instructions with human feedback." *Advances in neural information processing systems* 35 (2022): 27730-27744: <https://arxiv.org/abs/2203.02155>

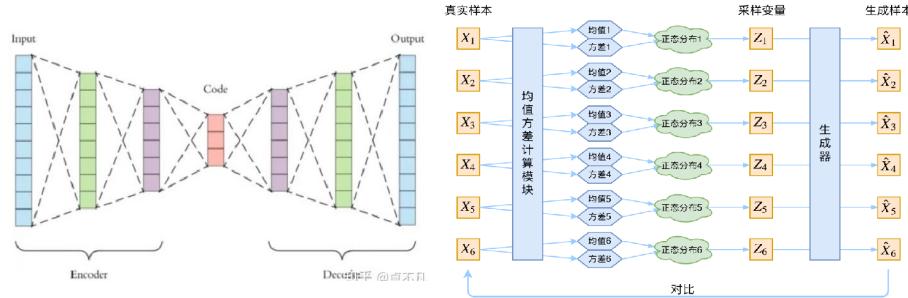
¹⁹Christiano, Paul F., et al. "Deep reinforcement learning from human preferences." *Advances in neural information processing systems* 30 (2017): https://proceedings.neurips.cc/paper_files/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html

可避免强化学习微调后模型输出较监督微调模型显著变化，以确保奖励模型评分的可靠性。目标函数的第三项为通用的语言模型训练目标，其目的是避免偏好对齐降低模型在通用 NLP 任务上的性能。

24.9 自编码器框架

24.9.1 经典自编码器 (autoencoder, AE) 与降噪自编码器 (denoising autoencoder, DAE)

图 24.10 自编码器模型（左）与变分自编码器模型（右）结构。由编码器、压缩表示和解码器组成。以重建输入数据为训练目标。理论上编码器可用于降维任务，解码器可用于生成任务。



- 从自编码器 (auto-encoder, AE) 到变分自编码器 (variational auto-encoder, VAE) 的历史也是人工神经网络模型从一众机器学习模型中脱颖而出并发展为几乎覆盖人工智能所有领域的通用范式的历史。1986 年 Hinton 首次提出了非线性的多层感知机模型和相应的 BP 反向传播算法，开启了神经网络研究的第二次热潮。在这段仅持续 5 年的热潮中，诞生了一大批当时尚未得到重视而如今名扬四海的模型，其中既包括 CNN 和 RNN，也包括最早的自编码器模型；
- 自编码器的诞生甚至早于 CNN 和 RNN 模型。在 BP 网络提出的同年（1986），Rumelhart、Hinton 和 Williams 即构建了基于 BP 模型的自编码器，作为人工神经网络模型于无监督学习的早期尝试。自编码器模型最初针对降维任务设计，也被称为表示学习 (representation learning)，旨在进行特征提取、数据压缩、数据降噪等经典任务；
- 自编码器的结构非常简单，主要由编码器 (encoder)、压缩表示 (code) 和解码器 (decoder) 三部分组成。其中解码器将输入压缩为潜在空间表示；解码器再将压缩表示重构回原始尺寸；压缩表示的维度一般低于原始输入，从而起到过滤冗余信息和降噪的效果。记 x 为输入数据， W_e, W_d 分别表示编码器、解码器待学习参数， \hat{x} 为解码输出，则自编码器的目标是最小化 x, \hat{x} 间的误差。以 L2 范数为例

$$(\min) \quad \frac{1}{2} \|x - \hat{x}\|_2^2, \quad \hat{x} = g_d(h, W_d), \quad h = g_e(x, W_e) \quad (\text{自编码器})$$

上式中 g_e, g_d 分别表示代表编码器和解码器的神经网络结构， h 为编码的压缩表示。自编码器模型设计的底层逻辑是——如果解码器能通过训练将压缩表示还原回原本的高维输入数据，意味着生成低维压缩表示的编码器模型可良好地挖掘并保留输入数据的有价值信息，同时滤取冗余和噪声信息；

- 针对数据类型的不同可采用不同的神经网络结构设计不同的自编码器。最简单的自编码器为线性自编码器，指模型完全由全连接层组成。而对于图像和视频数据则可训练卷积自编码器以挖掘图像和视频信息。卷积自编码器的编码器结构是传统的卷积网络，而解码器结构则需采用转置卷积 (transpose convolution) 和反池化 (unpooling) 运算；
- 自编码器与 PCA、NMF 等主流降维算法相同均是以数据重建为目标进行训练。若舍去线性自编码器中的非线性激活函数，则其本质上就是一个矩阵分解模型。但得益于神经网络模型的多层结构和非线性拟合能力，自编码器模型较传统的 PCA、NMF 等模型具有以下优势：

- 可通过多层非线性激活函数捕捉非线性相关性，实现非线性降维；
- 可通过变换神经网络类型针对性地学习特定数据的特征，如使用卷积层以适应视频、图像数据；
- 可利用来自另一个模型的预训练层通过迁移学习来增强编码器/解码器。

而且因为自编码属于以数据重建为目标的降维模型，除降维外模型理论上还可适用于其它任务：

- 仅保留训练后的编码器结构用于降维和表示学习任务；
- 同时保留训练后的编码器和解码器用于降噪任务；

- 仅保留训练后的解码器结构用于数据生成任务。
6. 尽管自编码器模型在理论上具有突出优势并适用于包括降维、降噪和数据生成在内的多项任务，但在应用中因为自编码器本身的不完善而未能迅速获得认可：
- 降维任务一般服务于特征工程和可视化两类目的。其中特征工程一般仅要求线性降维以降低数据的线性相关性，而非线性特征则由各类下游的非线性预测模型处理。因此具备非线性降维能力的自编码器模型与线性降维算法相比并不占优，反而面临可能因非线性特征挖掘能力过强导致的过拟合和降维结果可解释性不足的问题。可视化任务则要求数据降维后尽可能保留高维环境下的分布特征，而自编码器模型的目标函数并不包含低维表示 h 与输入 x 的分布的比较，故而也无法保证其结果可视化时可反映数据的原始分布结构；
 - 与其它降噪模型相比，自编码器的降噪效果缺乏理论支撑。模型并不能显式地定义、识别噪声，而是简单地认为输入 x 与重构输出 \hat{x} 的误差即为噪声。这将导致降噪效果受模型过拟合或欠拟合程度的影响，而因为模型以最小化 $\|x - \hat{x}\|_2^2$ 为目标使得无法在训练过程中判别并控制模型的欠拟合或过拟合程度，从而无法定量化地评价模型的降噪效果；
 - 数据生成是自编码器最有潜力的领域，因为具有非线性解码器，与线性的 PCA、NMF 相比在理论上可生成更复杂的数据结构。**但在实际应用中因为自编码器模型仅以最小化重构误差为训练目标而忽视了数据的分布规律，使得解码器难以正确重构训练集以外的低维向量表示。这一时期的自编码器完全不具备可泛化的生成能力，而 PCA、NMF 等模型反而得益于其线性特性具有更好的鲁棒性。因为神经网络模型易梯度消失的缺陷，其第二次研究热潮在 90 年代初即早早结束，针对自编码器模型的研究和改进也陷入低潮。直至 2006 年 Hinton 初步解决梯度消失的解决方案开启持续至今的神经网络第三次研究热潮，自编码器模型才因其理论上的优势重获关注；
7. **为提升自编码器模型的竞争，根本方式是引导模型于最小化重构误差的同时学习数据的分布规律以避免过拟合。**降噪自编码器 (denoising autoencoder, DAE) 即是其中具有阶段代表性的自编码器变体。模型于 2008 年由 Vincent P 等提出²⁰，针对降噪任务设计。模型的改进思路非常简单

$$(\min) \quad \frac{1}{2} \|x - \hat{x}\|_2^2, \quad \hat{x} = g_d(h, W_d), \quad h = g_e(x + \epsilon, W_e) \quad (\text{降噪自编码器})$$

为引导自编码器学习数据的分布规律，模型的输入不再是 x 而是 $x + \epsilon$ ，其中 ϵ 表示随机高斯噪声，而训练目标仍是最小化重建误差，从而强迫模型辨识输入中的噪声信息，进而学习数据的分布规律。另外在训练时还加入 dropout 机制进一步避免过拟合。经此改进，自编码器的泛化性和特征识别能力得到显著提升，开始在降维、降噪等任务（特别是针对图像、视频数据）中崭露头角，也奠定了其后续发展出生成能力的基础：

24.9.2 变分自编码器 (variational autoencoder, VAE)

1. 使自编码器具有可靠的生成能力的关键是确保输入数据被编码至尽可能平滑、连续的空间。这意味着两组相近的样本应具有相近的编码向量，而相似的编码向量则应解码出相近的结果，使得给定随机的编码向量，解码器可将其解码为尽可能接近训练样本的结果，以最大程度避免无意义结果。基于这一思路并参考降噪自编码器的做法，不难设计出如下形式的自编码器

$$(\min) \quad \frac{1}{2} \|x - \hat{x}\|_2^2, \quad \hat{x} = g_d(h + \epsilon, W_d), \quad h = g_e(x, W_e)$$

上式中 ϵ 同样表示随机噪声。按此思路设计的自编码器可以在一定程度上保证嵌入向量 h 周围邻域空间内解码结果的合理性，从而形成初步的生成能力。但这一思路仍存在以下问题：

- 最直接的问题是仅考虑单个样本嵌入结果的局部邻域空间内的解码合理性，而仍忽略了相近样本间嵌入结果的相似性，即无法保证相近样本 $\{x^{(i)}, x^{(j)}\}$ 具有相近的编码向量 $\{h^{(i)}, h^{(j)}\}$ ；

²⁰Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (pp. 1096-1103). (至 2024 年 3 月 5 日论文引用量 8747 次)

- 另一个问题是在于随机噪声 ϵ 所属分布的方差不应与样本的局部分布结构无关。例如 $x^{(i)}$ 在样本集中存在大量相似样本，则对 $h^{(i)}$ 扰动时应设置方差相对较小的噪声 $\epsilon^{(i)}$ 以区分 $h^{(i)}$ 与邻近嵌入间的差异性；而对于相对离群的样本 $x^{(j)}$ 则应设置方差较大的扰动噪声 $\epsilon^{(j)}$ 以使 $h^{(j)}$ 的影响覆盖较大的区域。

沿着这一思路进一步改进，即得到著名的变分自编码模型：

2. 2013 年末提出的变分自编码器模型将自编码器家族带到了全新的高度，至此自编码器模型真正意义上获得了生成能力^{21,22}。变分自编码器是自编码器家族中唯一的“生成模型”，与 GAN（2014 年）、Diffusion（2015 年提出，2020 年后成熟）并称为目前生成式深度学习领域的三大主流范式。一般认为，VAE 与 GAN 各有优劣。前者训练过程更稳定，潜在空间连续性及生成样本丰富度更高，但在数据质量上处于劣势；而后者则更善于生成更高质量、更逼真的数据；
3. 不再机械地人为构造噪声，变分自编码器创造性地将输入 x 编码至概率密度空间而非确定性的向量空间以使模型自适应地构造并克服噪声的影响从而学习数据结构。具体地，对于任意样本 $x^{(i)}$ ，编码器认为其编码 $h^{(i)}$ 服从多元高斯分布 $N(\mu^{(i)}, (\sigma^{(i)})^2 I)$ ，其中 $\mu^{(i)}, \sigma^{(i)}$ 是编码器的编码结果。而解码器的输入则是对 $N(\mu^{(i)}, (\sigma^{(i)})^2 I)$ 的采样，从而得到模型的重构结果 $\hat{x}^{(i)}$

$$\hat{x}^{(i)} = g_d(z^{(i)}, W_d), \quad z^{(i)} \sim N(\mu^{(i)}, (\sigma^{(i)})^2 I), \quad \mu^{(i)} = g_{e,\mu}(x^{(i)}, W_{e,\mu}), \quad \ln(\sigma^{(i)})^2 = g_{e,\sigma}(x^{(i)}, W_{e,\sigma})$$

可以看到此时自编码器中包含两套独立的神经网络模型 $g_{e,\mu}, g_{e,\sigma}$ 以分别编码高斯分布的均值 μ 和标准差 σ ，又因为 σ 非负，为减少约束模型以 $\ln(\sigma^{(i)})^2$ 为编码目标；

4. 进一步介绍变分自编码器模型的训练目标。因为模型自行控制噪声的生成，如果仅以最小化重构误差为目标将很可能导致模型令 $\sigma \rightarrow 0$ 以去除噪声从而降低学习难度，此时模型将退化为传统的自编码器，失去生成能力。为保留编码时的随机性，模型假设 $N(\mu^{(i)}, (\sigma^{(i)})^2 I)$ 服从 $N(0, I)$ 的标准高斯先验，基于 $N(\mu^{(i)}, (\sigma^{(i)})^2 I)$ 与 $N(0, I)$ 之间的 KL 散度为重构误差引入正则项

$$(\min) \quad \frac{1}{2} \|x^{(i)} - \hat{x}^{(i)}\|_2^2 + \frac{1}{2} \sum_{k=1}^K \left(\mu_k^{(i)2} + \sigma_k^{(i)2} - \ln \sigma_k^{(i)2} - 1 \right) \quad (\text{变分自编码器})$$

上式中 K 表示编码空间的维数。多元高斯分布间 KL 散度的推导详见第 23.7.2 节。引入 KL 散度正则项的具体作用机理为：

- 一方面引导所有样本 $\{x^{(i)}\}$ 的期望编码 $\mu^{(i)}$ 均趋于 0，即令各样本的编码在整体上尽可能接近。在此大约束下即可自然地使得相似样本具有相近嵌入，同时排斥差异较大的样本；
- 另一方面自适应学习各样本对应的扰动标准差 $\sigma^{(i)}$ 。当重构误差较大时则会自然地减少 $\sigma^{(i)}$ 以降低随机性进而降低学习难度，当重构误差减小至一定限度后又会增大 $\sigma^{(i)}$ 以提升生成器的生成能力。

5. “变分自编码器”的所谓“变分”得名于“变分推断（variational inference，见第 27.2 节）”，但截至目前在推导时并未显式说明变分自编码器与变分推断之间的关系。因此进一步地将从概率图模型的视角出发重新理解变分自编码器模型。在概率图模型的视角下称编码器为推断模型（inference model），将样本 x 映射至高斯概率空间，记为 $q_\phi(z|x)$ ，其中 ϕ 为编码器参数。并假设因变量 z 关于 x 的真实后验分布为 $p(z|x)$ ，则编码器的学习目标是通过简单的高斯分布 $q_\phi(z|x)$ 拟合难以计算的复杂后验分布 $p(z|x)$ ，即最小化 $q_\phi(z|x)$, $p(z|x)$ 间的 KL 散度，按变分推断建模该目标函数等价于最大化证据下界（ELBO）

$$\begin{aligned} (\min) \quad -\mathcal{L}_{ELBO}(x) &= - \int_z q_\phi(z|x) [\ln p(z, x) - \ln q_\phi(z|x)] dz \\ &= - \int_z q_\phi(z|x) [\ln p(x|z) + \ln p(z) - \ln q_\phi(z|x)] dz = -\mathbb{E}_{q_\phi(z|x)} [\ln p(x|z)] + D_{KL}(q_\phi(z|x) \| p(z)) \end{aligned}$$

显然 $q_\phi(z|x)$ 即为变分推断中的变分分布，并按经典的“坐标上升变分推断”算法假设其服从各分量无关的多元高斯分布 $N(\mu^{(i)}, (\sigma^{(i)})^2 I)$ ； $p(z)$ 为隐变量 z 的先验分布。可以看到，若条件概率 $p(x|z)$ 已知，则无

²¹ Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*. <https://arxiv.org/abs/1312.6114> (至 2024 年 3 月 5 日论文引用量 33634 次)

²² 变分自编码器（一）：原来是这么一回事：<https://kexue.fm/archives/5253>

需解码器即可训练编码器模型学习 x 的概率空间嵌入 z 。而在 $p(x|z)$ 未知的情况下则需构建解码器模型拟合 $p_\theta(x|z) \simeq p(x|z)$, 其中 θ 为解码器模型参数。在概率图模型视角下称编码器为生成模型 (**generative model**)。则上述目标函数进一步写为

$$(\min) -\mathcal{L}_{ELOB}(x) = -\mathbb{E}_{q_\phi(z|x)} [\ln p(x|z)] + D_{KL}(q_\phi(z|x)\|p(z)) \simeq -\mathbb{E}_{q_\phi(z|x)} [\ln p_\theta(x|z)] + D_{KL}(q_\phi(z|x)\|p(z))$$

上式目标函数涉及了编码器和解码器的参数 ϕ, θ , 则由误差反向传播即可优化编码器和解码器。注意到 $D_{KL}(q_\phi(z|x)\|p(z))$ 即为上文讨论的为避免编码器模型丧失随机性而引入的 KL 散度正则项, 则对应的 $\mathbb{E}_{q_\phi(z|x)} [\ln p_\theta(x|z)]$ 应与重建误差有关, 故进一步讨论其统计学意义。注意到上述期望无法解析求解, 则由蒙特卡罗方法估计有

$$\mathbb{E}_{q_\phi(z|x)} [\ln p_\theta(x|z)] = \frac{1}{L} \sum_{l=1}^L \ln p_\theta \left(x^{(l)} \middle| z^{(l)} \right), \quad z^{(l)} \sim q_\phi \left(z \middle| x^{(l)} \right)$$

上式中 $x^{(l)}$ 为随机抽样的样本, L 为随机抽样的样本数。对于连续变量 x , 一般假设其服从高斯分布, 则解码器对应的生成模型 $p_\theta(x|z)$ 同样为高斯分布, 则有

$$\mathbb{E}_{q_\phi(z|x)} [\ln p_\theta(x|z)] = \frac{1}{L} \sum_{l=1}^L \ln p_\theta \left(x^{(l)} \middle| z^{(l)} \right) \propto -\frac{1}{2L} \sum_{l=1}^L \frac{(x^{(l)} - \hat{x}^{(l)})^2}{\sigma_{x^{(l)}}^2}$$

式中 \hat{x}, σ_x 为解码器输出, 表示样本 x 服从的高斯分布的期望和标准差。显然只需令 $L = 1$, 并规定 $\sigma_x = 1$ 只重建 \hat{x} , 则 $-\mathbb{E}_{q_\phi(z|x)} [\ln p_\theta(x|z)]$ 等价于自编码器框架中的重建误差。当 x 为连续变量时假设其服从高斯分布, 则重建误差可建模为均方误差的形式; 而 x 为离散变量时可假设其服从伯努利分布或泊松分布等, 则对应的重建误差将有其它的形式;

6. 最后介绍变分自编码器模型训练时的重参数化技巧 (**reparameterization trick**)。因为解码器的输入 $z^{(i)}$ 是高斯分布的采样结果, 而采样操作不可导, 导致模型无法通过反向传播更新编码器参数。为此不妨参考上文介绍的朴素的为编码结果引入噪声的做法, 令

$$z^{(i)} = \mu^{(i)} + \sigma^{(i)} \cdot \epsilon, \quad \epsilon \sim N(0, I)$$

此时 $z^{(i)}$ 成为一个可微的服从高斯分布 $N(\mu^{(i)}, (\sigma^{(i)})^2 I)$ 的随机变量, 其随机性由独立的标准高斯分布噪声 ϵ 提供。这种转移随机性以保证变量可微的技巧即被称为重参数化。

24.9.3 VaDE 模型 (**variational deep encoding, IJCAI-17**)

- 变分自编码器模型的提出是自编码器框架发展的关键节点。将样本嵌入概率空间的创新使得自编码器模型首次获得了生成能力, 也意味着模型可产生更为健壮的嵌入结果。因此在此之后的自编码器模型研究主要以变分自编码器框架为基础并大体分为两个流派——一类旨在提升模型的生成能力, 而另一类则重新回归至传统的嵌入任务中, 旨在进一步提升编码器嵌入结果的可解释性。由北京理工大学、腾讯和 Hulu 团队于 2016 年最早提出的 VaDE 模型即属于后者²³;
- 训练后的变分自编码器模型可以将样本嵌入到相应的高斯概率空间中, 并通过解码器将高斯嵌入重新还原为原本的样本。因此理论上编码器生成的嵌入向量有效保留了原始样本的特征信息, 可替代原样本服务于下游任务。然而这一思路最大的问题在于编码器的编码结果是为解码器服务的, 未显式要求嵌入空间分布区分不同类别样本, 反而可能不利于其它模型学习。为此便要求编码器编码向量时提供更丰富且可解释性的信息:
 - 一部分研究是在自编码框架中引入以编码结果为输入的其它模块以及相应地学习误差, 通过更丰富和针对性的误差函数提升嵌入结果的可解释性, 而且引入的额外模块也可使得模型同时学习编码和其它下游任务;

²³Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2017. Variational deep embedding: an unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. AAAI Press, 1965–1972. <https://arxiv.org/pdf/1611.05148.pdf>

- 另一部分研究直接从编码器着手，根据具体的可解释性目标和下游任务需求规定编码的概率空间的具体形式。**VaDE** 模型即是将编码器的编码结果由高斯分布修改为混合高斯分布，编码结果可显式地区分原始分布中不同簇的样本，同时实现降维和聚类两大无监督学习任务。

3. 详细介绍 VaDE 模型的模型设计和数学细节。记 x, z, c 分别表示原始样本、样本嵌入和样本的聚类结果， K 为预设的聚类数目。首先介绍模型的解码器生成过程，有

$$p(x, z, c) = p(x|z)p(z|c)p(c), \quad c \sim \text{Cat}(\pi), \quad z \sim N(\mu_c, \sigma_c^2 I), \quad \hat{x} = f_\theta(z) \quad (\text{VaDE 模型生成模型})$$

具体地，解码器基于概率分布为 π 的离散分布先验 $p(c)$ 生成样本类别 c ，再从相应类别 c 的正态分布 $N(\mu_c, \sigma_c^2 I)$ 中抽样嵌入向量 z ，最后以 z 为输入由参数为 θ 的神经网络模型 $f_\theta(\cdot)$ 重构 \hat{x} ，旨在拟合后验分布 $p(x|z)$ 。因为解码器抽样时不同类别对应不同的高斯概率空间，因此解码器是从混合高斯分布空间进行抽样。当 x 为连续变量时模型输出 \hat{x} 服从混合高斯分布。 π, μ_c, σ_c 不是解码器或生成器的参数或输出，而是 **VaDE** 模型的可学习参数；

4. 模型的编码器在编码 z 时考虑了其所属的类别 c ，其推断过程则建模为

$$q(z, c|x) = q(z|x)q(c|x), \quad q(z|x) = N(z|\tilde{\mu}, \tilde{\sigma}^2 I), \quad \tilde{\mu}, \tilde{\sigma} = g_\phi(x) \quad (\text{VaDE 模型推断模型})$$

式中 $g_\phi(\cdot)$ 表示参数为 ϕ 的编码器神经网络，在数学上与变分自编码器一致将 x 嵌入高斯概率空间。观察 VaDE 模型中编码-解码关系发现，VaDE 模型进一步打断了编码器与解码器的连接关系——变分自编码器的编码器和解码器并不完全连接，但解码器的输入仍是基于编码器的输出随机化后得到；而 **VaDE** 模型中解码器的输入和编码器的输出间再无直接联系，但通过样本重构学习也可自然地使得编码器的输出 $\tilde{\mu}, \tilde{\sigma}$ 根据 x 所属的类别 c 收敛至 μ_c, σ_c 。由此带来两项优势：

- VaDE 模型并不需要显式地搭建神经网络分类模块以拟合 $q(c|x)$ 。 $q(c|x)$ 的估计将于后文中介绍；
- 编程时仍按一般的重参数化方法基于编码器输出随机化作为解码器的输入，而通过将 π, μ_c, σ_c 作为 VaDE 模型的可学习参数自然地引导编码器将 x 嵌入混合高斯分布空间。

5. 基于生成模型和推断模型，遵循变分自编码器框架推导 VaDE 模型目标函数。最大化证据下界 (ELOB)

$$(\min) -\mathcal{L}_{ELOB}(x) = -\mathbb{E}_{q(z, c|x)} \left[\ln \frac{p(x, z, c)}{q(z, c|x)} \right] = -\mathbb{E}_{q(z, c|x)} [\ln p(x|z) + \ln p(z|c) + \ln p(c) - \ln q(z|x) - \ln q(c|x)]$$

逐项讨论上式。在变分自编码器目标函数理解中已指出上式第一项 $\mathbb{E}_{q(z, c|x)}[\ln p(x|z)]$ 等价于经典自编码器框架中的重构误差，由蒙特卡洛方法估计。而其它各项有：

- 对 $\mathbb{E}_{q(z, c|x)}[\ln p(z|c)]$ ，因为 $q(z, c|x)$ 为关于 z, c 的联合分布，则按期望定义展开为关于 z, c 的二重积分

$$\mathbb{E}_{q(z, c|x)}[\ln p(z|c)] = \int_z \sum_{c=1}^k q(z, c|x) \ln p(z|c) dz = \sum_{c=1}^k q(c|x) \int_z q(z|x) \ln p(z|c) dz$$

式中 $q(z|x)$, $p(z|c)$ 均为多元高斯分布， $\int_z q(z|x) \ln p(z|c) dz$ 表示两个高斯分布的交叉熵（见第 23.9.6 节）。交叉熵的定义来源于 KL 散度。根据多元高斯分布的 KL 散度的相关推导（见第 23.7.2 节），得到其交叉熵并代入上式有

$$\begin{aligned} \mathbb{E}_{q(z, c|x)}[\ln p(z|c)] &= -\sum_{c=1}^k q(c|x) \left[\frac{J}{2} \ln(2\pi) + \frac{1}{2} \sum_{j=1}^J \left(\ln \sigma_{c,j}^2 + \frac{\tilde{\sigma}_j^2}{\sigma_{c,j}^2} + \frac{(\tilde{\mu}_j - \mu_{c,j})^2}{\sigma_{c,j}^2} \right) \right] \\ &= -\frac{J}{2} \ln(2\pi) - \frac{1}{2} \sum_{c=1}^k q(c|x) \sum_{j=1}^J \left(\ln \sigma_{c,j}^2 + \frac{\tilde{\sigma}_j^2}{\sigma_{c,j}^2} + \frac{(\tilde{\mu}_j - \mu_{c,j})^2}{\sigma_{c,j}^2} \right) \end{aligned}$$

式中 J 表示编码向量 z 的维数。可以看到上式中包含 $\frac{\tilde{\sigma}_j^2}{\sigma_{c,j}^2}$, $(\tilde{\mu}_j - \mu_{c,j})^2$ 两项，意味着通过优化改式可使得编码器的输出 $\tilde{\mu}, \tilde{\sigma}$ 根据 x 所属的类别 c 收敛至 μ_c, σ_c ；

- 对 $\mathbb{E}_{q(z, c|x)}[\ln p(c)]$ 同样按期望定义展开有

$$\mathbb{E}_{q(z, c|x)}[\ln p(c)] = \int_z \sum_{c=1}^k q(z, c|x) \ln p(c) dz = \sum_{c=1}^k q(c|x) \ln p(c) \int_z q(z|x) dz = \sum_{c=1}^k q(c|x) \ln \pi_c$$

- 对 $\mathbb{E}_{q(z,c|x)}[\ln q(z|x)]$ 同样按期望定义展开有

$$\mathbb{E}_{q(z,c|x)}[\ln q(z|x)] = \int_z \sum_{c=1}^k q(z, c|x) \ln q(z|x) dz = \int_z q(z|x) \ln q(z|x) dz \sum_{c=1}^k q(c|x) = \int_z q(z|x) \ln q(z|x) dz$$

$\int_z q(z|x) \ln q(z|x) dz$ 表示高斯分布的信息熵（见第 23.9.6 节），其推导同样见第 23.7.2 节中多元高斯分布 KL 散度的推导，有

$$\mathbb{E}_{q(z,c|x)}[\ln q(z|x)] = \int_z q(z|x) \ln q(z|x) dz = -\frac{J}{2} \ln(2\pi) - \frac{1}{2} \sum_{j=1}^J (1 + \ln \tilde{\sigma}_j^2)$$

- 对 $\mathbb{E}_{q(z,c|x)}[\ln q(c|x)]$ 同样按期望定义展开有

$$\mathbb{E}_{q(z,c|x)}[\ln q(c|x)] = \int_z \sum_{c=1}^k q(z, c|x) \ln q(c|x) dz = \int_z q(z|x) dz \sum_{c=1}^k q(c|x) \ln q(c|x) = \sum_{c=1}^k q(c|x) \ln q(c|x)$$

将 $q(c|x)$ 简记为 γ_c ，整合以上各项得到 VaDE 模型的目标函数有

$$(\min) -\mathbb{E}_{q(z,c|x)}[\ln p(x|z)] + \frac{1}{2} \sum_{c=1}^k \gamma_c \sum_{j=1}^J \left(\ln \sigma_{c,j}^2 + \frac{\tilde{\sigma}_j^2}{\sigma_{c,j}^2} + \frac{(\tilde{\mu}_j - \mu_{c,j})^2}{\sigma_{c,j}^2} \right) - \frac{1}{2} \sum_{j=1}^J (1 + \ln \tilde{\sigma}_j^2) + \sum_{c=1}^k \gamma_c \ln \frac{\gamma_c}{\pi_c}$$

6. 只需再估计 $\gamma_c = q(c|x)$ 即可计算 VaDE 模型目标函数。重写 $\mathcal{L}_{ELOB}(x)$ 的形式有

$$\begin{aligned} \mathcal{L}_{ELOB}(x) &= \mathbb{E}_{q(z,c|x)} \left[\ln \frac{p(x, z, c)}{q(z, c|x)} \right] = \int_z \sum_c q(c|x) q(z|x) \left[\ln \frac{p(x|z)p(z)}{q(z|x)} + \ln \frac{p(c|z)}{q(c|x)} \right] dz \\ &= \int_z q(z|x) \ln \frac{p(x|z)p(z)}{q(z|x)} dz - \int_z q(z|x) \sum_c q(c|x) \ln \frac{q(c|x)}{p(c|z)} dz \\ &= \int_z q(z|x) \ln \frac{p(x|z)p(z)}{q(z|x)} dz - \int_z q(z|x) D_{KL}(q(c|x) \| p(c|z)) dz \end{aligned}$$

上式中第一项只与 z 有关，而第二项与 z, c 相关，且因为 KL 散度非负，则第二项非负。因此为最大化 $\mathcal{L}_{ELOB}(x)$ ，必然要求 $D_{KL}(q(c|x) \| p(c|z)) = 0$ ，即 $q(c|x) = p(c|z)$ 。因此可以用 $p(c|z)$ 近似估计 $q(c|x)$ ，从而将估计 $q(c|x)$ 的问题转化为估计 $p(c|z)$ 的问题，并由贝叶斯公式有

$$\gamma_c = q(c|x) \simeq p(c|z) = \frac{p(z|c)p(c)}{\sum_{c'} p(z|c')p(c')} = \frac{N(z|\mu_c, \sigma_c^2 I) \pi_c}{\sum_{c'} N(z|\mu_{c'}, \sigma_{c'}^2 I) \pi_{c'}}$$

因为分布 $p(z|c)$, $p(c)$ 均已知，故 $q(c|x)$ 可估计，从而可最终计算 VaDE 模型的目标函数；

7. 最后介绍 VaDE 模型的预训练过程。VaDE 模型的目标函数远比变分自编码模型复杂。因为引入了更多的正则项，在 VaDE 模型训练初期重构误差 $-\mathbb{E}_{q(z,c|x)}[\ln p(x|z)]$ 于总误差中的占比将远小于变分自编码器模型，导致模型可能早早陷入局部最小点而在后续训练中难以进一步优化重构误差。因此 VaDE 模型训练前需首先进行预训练。预训练具有两项目的——初始化 VaDE 模型的参数使其具有较好的重构误差，同时初始化可学习参数 π, μ_c, σ_c 。具体流程为：

- 搭建一个与 VaDE 模型结构一致、但不考虑随机嵌入的经典自编码器模型，仅考虑重构误差进行少量预训练（如 10 轮）；
- 预训练后的自编码器模型参数用于初始化 VaDE 模型的相关参数，同时对预训练自编码器模型的编码结果 z 作混合高斯分布建模，得到 VaDE 模型中可学习参数 π, μ_c, σ_c 的初始值。

24.9.4 DCC 模型 (deep continuous clustering)

1. 与 VaDE 模型类似，2018 年提出的 DCC 模型²⁴同样基于自编码器框架同时降维和聚类，从而实现对高维样本（图像、文本）等的精确聚类。不同于 VaDE，DCC 模型在降维层面基于自编码器框架而非变分自编码器；在聚类层面也非直接分配样本类别，而是参考谱聚类的思路将聚类问题转化为图分割问题。另外 DCC 模型无需指定聚类数目，而是类似 DBSCAN 算法基于预设的样本距离阈值自动区分类内样本；

²⁴Shah, S.A., Koltun, V., 2018. Deep continuous clustering. arXiv preprint arXiv:1803.01449. <https://arxiv.org/abs/1803.01449>

2. 由名可知, DCC 模型聚焦于“连续聚类”。模型认为为联合降维聚类, 理想情况应将降维与聚类的模型参数融合入单一目标函数中, 并通过误差反向传播同时优化降维与聚类参数。但聚类过程天然具有离散性——聚类中心、类别数目等参数均可能随优化过程跳变, 造成目标函数和对应梯度不连续, 不利于降维与聚类参数的联合更新。“连续聚类”则要求目标函数在优化过程中完全固定, 从而保证误差梯度不随聚类结果的改变而跳变。为此, **模型构造的目标函数并不显式包含与样本聚类有关的参数, 而是仅引导相似样本生成相近的低维表示;**
3. 具体地, 令 $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ 表示原始的高维样本数据集, N 为样本数目; 则模型同时优化样本的两种低维表示 $\mathbf{Y}, \mathbf{Z} \in \mathbb{R}^{d \times N}$ ($d \ll D$), 其中 \mathbf{Y} 为自编码器生成的隐变量, \mathbf{Z} 则是基于 \mathbf{Y} 进一步构造的用于聚类的低维表示。模型目标函数如下

$$(\min) \quad \mathcal{L}(\mathbf{Z}) = \frac{1}{D} \|\mathbf{X} - G_\omega(\mathbf{Y})\|_F^2 + \frac{1}{d} \left(\sum_i \rho_1(\|\mathbf{z}_i - \mathbf{y}_i\|_2; \mu_1) + \lambda \sum_{(i,j) \in \mathcal{E}} w_{ij} \rho_2(\|\mathbf{z}_i - \mathbf{z}_j\|_2; \mu_2) \right), \quad \mathbf{Y} = F_\theta(\mathbf{X})$$

式中 $G_\omega(\cdot)$, $F_\theta(\cdot)$ 分别表示自编码模型的编码器和解码器; λ, μ_1, μ_2 为算法超参; \mathcal{E} 为预构造的连接各样本点的图结构, 基于原始高维数据 \mathbf{X} 按 KNN 近邻规则生成连接, 边权 $w_{ij} = \frac{\sum_{k=1}^n n_k}{N \sqrt{n_i n_j}}$, n_i 表示样本 i 的连边数; $\rho_1(\cdot), \rho_2(\cdot)$ 为下调型 M 估计量 (redescending M-estimator), 研究采用 Geman-McGlure 函数

$$\rho(x; \mu) = \frac{\mu x^2}{\mu + x^2}, \quad \mu > 0$$

可以看到, 当 $x \rightarrow 0$ 时 $\rho(x; \mu) \rightarrow 0$, 当 $x \rightarrow \infty$ 时 $\rho(x; \mu) \rightarrow \mu$ 。这是统计学中的一种稳健估计方法, 将其加入误差函数可使得误差函数对异常大的偏差不敏感, 提升估计的鲁棒性。因为目标函数仅包含样本的低维表示, 故其梯度不受聚类结果的影响, 从而实现连续聚类;

4. 上述目标函数由三项误差组成。第一项为自编码器的重构误差; 第二项为数据低维表示 \mathbf{Y}, \mathbf{Z} 之间的误差, 希望同一样本的两项低维表示 $\mathbf{y}_i, \mathbf{z}_i$ 一致; 第三项为样本对低维表示误差, 旨在使相近的样本具有相似的表示, 直接服务样本聚类。理论上模型可仅优化自编码器生成的低维表示 \mathbf{Y} , 去掉目标函数中的第二项, 并将第三项样本对误差中的 \mathbf{z}_i 替换为 \mathbf{y}_i 。但通过区分自编码器低维表示 \mathbf{Y} 和用于聚类的低维表示 \mathbf{Z} , 并引入下调型 M 估计量, 可克服原始数据潜在噪声的影响, 得到更鲁棒的聚类结果;
5. 进一步介绍模型参数设置细节。模型的自编码器采用堆叠降噪自编码器 (stacked denoising autoencoder, SDAE), 并在初始化时令 $\mathbf{Z} = \mathbf{Y} = F_\theta(\mathbf{X})$ 。同时初始化超参 $\lambda = \frac{\|\mathbf{Y}\|_2}{\|\mathbf{A}\|_2}$, $\mathbf{A} = \sum_{(i,j) \in \mathcal{E}} w_{ij} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top$, 其中 $\|\cdot\|_2$ 表示矩阵的谱范数 (spectral norm), 即最大奇异值, $\mathbf{e}_i \in \mathbb{R}^d$ 为第 i 个元素为 1 的单位向量。 λ 的取值不随后续优化变化。重点介绍下调型 M 估计量超参 μ_1, μ_2 的取值方法。若 μ_1, μ_2 过大, 则目标函数更易收敛, 但对噪声敏感, 且容易导致所有样本聚为一类; 反之则相反。因此研究首先令 μ_1, μ_2 取较大值, 并随迭代逐渐减小, 直至达到下限 $\frac{\delta_1}{2}, \frac{\delta_2}{2}$ 。其中 δ_1 定义为所有 \mathbf{y}_i 到 \mathbf{Y} 中点的距离的平均值; δ_2 定义为 \mathcal{E} 中节点距离的 1% 分位数;
6. 样本量过大时可采用批数据集训练模型。为照顾目标函数中的第三项数据对误差, 采样随机边采样的方法从连边集 \mathcal{E} 中提取批样本集 \mathcal{B} , 得到批训练目标函数 $\mathcal{L}_{\mathcal{B}}$

$$(\min) \quad \mathcal{L}_{\mathcal{B}}(\mathbf{Z}) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} w_i \left(\frac{\|\mathbf{x}_i - g_\omega(\mathbf{y}_i)\|_2^2}{D} + \frac{\rho_1(\|\mathbf{z}_i - \mathbf{y}_i\|_2)}{d} \right) + \frac{\lambda}{|\mathcal{B}|} \sum_{(i,j) \in \mathcal{E}_{\mathcal{B}}} \frac{w_{ij}}{d} \rho_2(\|\mathbf{z}_i - \mathbf{z}_j\|_2), \quad \mathbf{y}_i = f_\theta(\mathbf{x}_i), \quad w_i = \frac{n_i^{\mathcal{B}}}{n_i}$$

式中 w_i 是批训练中额外引入的权重参数。因为目标函数的前两项是以样本 (节点) 为单位, 而后一项是以样本对 (连边) 为单位, 则基于边采样构造批训练集时可能出现样本对不重复但样本重复采样的情况, 造成目标函数的前两项权重偏高, 故而引入 w_i 进行调整。 $n_i^{\mathcal{B}}$ 为子图 $\mathcal{E}_{\mathcal{B}}$ 中节点 i 的连边数;

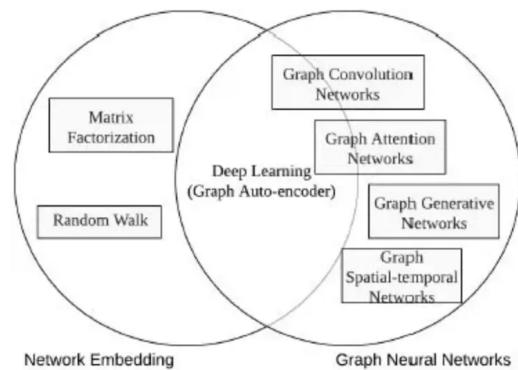
7. 最后介绍模型训练的收敛判别和聚类生成方法。当 μ_1, μ_2 均收敛至最小值后, 每次迭代模型均基于 \mathbf{Z} 构造图结构 \mathcal{G} , 当 $\|\mathbf{z}_i - \mathbf{z}_j\| < \delta_2$ 时表示样本 i, j 相连。此时一个连通子图表示同属一类的样本。若某次迭代后 \mathcal{G} 的连边变化量小于 0.1%, 则模型训练完成。

第 25 章

图机器学习 (Graph Machine Learning)

不同于传统机器学习模型只适用于处理如表格、图像 (image)、文本、音频等欧式空间数据，图机器学习 (**graph machine learning, Graph ML**) 是一类用于处理图结构 (graph) 数据的机器学习范式。图结构数据中的实体以节点的形式表示，实体之间的关系以边的形式表示。图数据的复杂性对传统的机器学习算法提出了重大挑战。由于图可能是不规则的，因此图可能具有可变大小的无序节点，并且来自图中的节点可能具有不同数量的邻居，导致一些重要的操作（例如卷积）在图像域中很容易计算，但是难以应用于图域。此外，现有机器学习算法的一个核心假设是各个实例相互独立。这个假设不再适用于图数据，因为每个实例（节点）都通过各种类型的链接（例如引用、友谊和交互）与其他实例相关联。

图机器学习的目标是从图结构数据中学习有用的表现，并利用这些表示进行各种任务，例如节点分类、图分类、链接预测等。现阶段的图机器学习研究大体可分为两个领域——图神经网络 (**graph neural network**) 与图表示学习 (**graph representation learning**)。其中图神经网络本质上是一种针对图的预测模型，尝试将传统的针对欧式空间数据的神经网络结构迁移至图数据；而图表示学习又称为图嵌入 (**graph embedding**)，本质上是一种针对图的降维方法，尝试将复杂高维的图拓扑结构嵌入欧式向量空间。目前，最先进的图神经网络分为四类，即递归图神经网络、卷积图神经网络、图自动编码器和时空图神经网络。而图表示学习范式则可分为三类，即基于因子分解的方法、基于随机游走的方法和基于深度学习的方法。



: Network Embedding v.s. Graph Neural Networks.

25.1 初代图神经网络 (**Vanilla graph neural network, Vanilla GNN**)

- 正如同神经网络模型早在深度学习兴起之前即已被提出，图神经网络概念的提出也早于图深度学习的流行。图神经网络的概念最早在 2004 年提出，并于 2009 年由 Scarselli 在其论文中定义了图神经网络的理论基础¹，形成了最基础的图神经网络模型。但该时期仍处于神经网络研究的低潮时期，图神经网络同样不温不火。直至 2012 年以卷积神经网络 (CNN) 为代表的深度学习模型成为主流，并于 2013 年将卷积运算迁移至图结构得到图卷积神经网络 (GCN) 后，图神经网络与图深度学习才开始逐渐兴起。本节主要介绍图学习的基本概念以及最基本的图神经网络；
- 图领域的应用通常分为 **graph-focused** 和 **node-focused** 两类。顾名思义，graph-focused 任务更关注图的整体结构，图分类和图生成即属于经典的 **graph-focused** 任务； node-focused 任务更关注图结构中局部节点

¹Scarselli F, Gori M, Tsoi A C, et al. The graph neural network model[J]. IEEE transactions on neural networks, 2009, 20(1): 61-80. <https://ro.uow.edu.au/cgi/viewcontent.cgi?article=10501&context=infopapers>

的特征，经典的 **node-focused** 任务包括节点分类和链接预测等。用数学语言描述，则 graph-focused 任务可建模为构造一个映射 f 使得 $f(G) \in \mathbb{R}^m$ ；而 node-focused 任务可建模为构造一个映射 f 使得 $f(G, n) \in \mathbb{R}^m$ 。上文中 G, n 分别表示图和图中的一个节点。因为图的整体特征实际上是各节点处局部结构特征的宏观体现，故 **graph-focused** 任务在建模时可还原回 **node-focused** 任务。只需增加一个与图中所有节点相连的“超节点 (super node)”或对所有节点的映射结果进行进一步映射即可；

3. 在应用图神经网络时又可将图结构分为 **nonpositional graph** 和 **positional graph** 两类。前者定义为节点的邻居节点没有顺序关系，可以任意排列的图；反之则属于 **positional graph**。大部分图结构均属于前者，而后者多见于异质图，例如综合交通网络中任意节点的邻居节点包括地面路网节点和公交网络节点，两类节点的特征组成、影响机制完全不同，不可任意调换位置；
4. 进一步介绍最基本的图神经网络模型。记节点 n 属性 (feature) 为 x_n ，连边 e 属性为 x_e ，表征节点 n 局部结构特征的隐状态 (state embedding) 为 h_n 。图神经网络的核心思想是通过消息传递机制 (**message-passing mechanism**) 聚合节点的邻居信息，并更新节点的隐状态。用数学语言表示为

$$h_n^{k+1} = f_\omega(x_n, x_{co[n]}, x_{ne[n]}, h_{ne[n]}^k)$$

上式中 f_ω 称为局部转移函数 (**local transition function**)，其参数为 ω ； $co[n], ne[n]$ 分别表示连接节点 n 的连边与节点的集合。对于 non-positional graph，因为各相邻节点没有顺序关系，则上式也可写为

$$h_n^{k+1} = \sum_{m \in ne[n]} f_\omega(x_n, x_{(m,n)}, x_m, h_m^k)$$

上述消息传递过程通常会进行多轮迭代，使得每个节点的隐状态包含其邻居和更远节点的信息。假设经过 K 轮消息传递后得到每一节点的隐状态 h_n^K ，则最终每个节点的输出 o_n 计算为

$$o_n = g_\omega(h_n^K, x_n)$$

上式中 g_ω 称为局部输出函数 (**local output function**)。令 h, x, o 分别表示堆叠了所有节点隐状态、属性、输出的向量，又令 \tilde{x} 表示堆叠了所有节点和连边属性的向量，则上述消息传递和输出过程可简写为

$$h^{k+1} = F_\omega(h^k, \tilde{x}), \quad o = G_\omega(h^{k+1}, x)$$

式中 F_ω, G_ω 分别称为全局转移函数 (**global transition function**) 和全局输出函数 (**global output function**)，一般由神经网络模型实现；

5. 上文指出多轮消息传递即可使得每个节点的隐状态包含其相邻及更远节点的信息。而一个合理的期望即是每个节点的隐状态会随网络间消息传递的进行而逐渐收敛，此时意味着每个节点均“充分”接收了网络其它各节点的信息，这便是早期图神经网络的理论基础——不动点理论 (**fixed point theorem**)，具体地是指巴拿赫不动点理论 (**Banach's fixed point theorem**)。不动点理论指出，只要 F_ω 是一个压缩映射 (**contraction mapping**)，无论初值 h^0 是什么经若干轮迭代后 h^k 均会收敛至某个固定值 h^K ，该点即为不动点。进一步介绍压缩映射的定义。对于任意函数 $F(\cdot)$ ，其是压缩映射等价于 $\exists c \in [0, 1)$ 使得 $d(F(x), F(y)) \leq c \cdot d(x, y)$ ，其中 $d(x, y)$ 表示距离函数。以 $L2$ 范数表示距离，则该式又等价于

$$\frac{\|F(y) - F(x)\|_2}{\|y - x\|_2} \leq c \implies \frac{\|F(x + \Delta) - F(x)\|_2}{\|\Delta\|_2} \leq c \implies \left\| \frac{\partial F}{\partial x} \right\|_2 \leq c$$

6. 根据压缩映射的要求，记误差函数为 e_ω ，则基础图神经网络模型的训练目标如下

$$\min_{\omega} e_\omega = loss, \quad \text{s.t. } \left\| \frac{\partial F_\omega}{\partial h^K} \right\|_2 \leq c$$

以经典的监督学习为例， $loss$ 为基于图神经网络模型最终输出 o 和真实标签计算的误差。若按上式优化意味着 F_ω 不能为任意映射函数，但因 F_ω, G_ω 均为神经网络模型，而限制神经网络模型的性质难度较大，因此基于罚函数法将约束转移至目标函数，从而将神经网络训练松弛为无约束优化问题

$$\min_{\omega} e_\omega = loss + \lambda \cdot \max \left\{ \left\| \frac{\partial F_\omega}{\partial h^K} \right\|_2 - c, 0 \right\}, \quad \lambda > 0 \quad (\text{GNN 优化目标})$$

上式中 λ 为算法超参。经罚函数松弛后 F_ω, G_ω 即可取任意神经网络模型同时使得 h^k 近似收敛至不动点；

图 25.1 初代图神经网络模型的误差反向传播训练算法

```

MAIN
    initialize  $w$ ;
     $x = \text{FORWARD}(w)$ ;
    repeat
         $\frac{\partial e_w}{\partial w} = \text{BACKWARD}(x, w)$ ;
         $w = w - \lambda \cdot \frac{\partial e_w}{\partial w}$ ;
         $x = \text{FORWARD}(w)$ ;
    until (a stopping criterion);
    return  $w$ ;
end

FORWARD( $w$ )
    initialize  $x(0), t = 0$ ;
    repeat
         $x(t+1) = F_w(x(t), l)$ ;
         $t = t + 1$ ;
    until  $\|x(t) - x(t-1)\| \leq \varepsilon_f$ ;
    return  $x(t)$ ;
end

BACKWARD( $x, w$ )
     $o = G_w(x, l_N)$ ;
     $A = \frac{\partial F_w}{\partial x}(x, l)$ ;
     $b = \frac{\partial e_w}{\partial o} \cdot \frac{\partial G_w}{\partial x}(x, l_N)$ ;
    initialize  $z(0), t=0$ ;
    repeat
         $z(t) = z(t+1) \cdot A + b$ ;
         $t=t-1$ ;
    until  $\|z(t-1) - z(t)\| \leq \varepsilon_b$ ;
     $c = \frac{\partial e_w}{\partial o} \cdot \frac{\partial G_w}{\partial x}(x, l_N)$ ;
     $d = z(t) \cdot \frac{\partial F_w}{\partial w}(x, l)$ ;
     $\frac{\partial e_w}{\partial w} = c + d$ ;
    return  $\frac{\partial e_w}{\partial w}$ ;
end

```

7. 最后推导误差反向传播过程，即计算 $\frac{\partial e_\omega}{\partial \omega}$ 。根据链式法则有

$$\frac{\partial e_\omega}{\partial \omega} = \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial \omega} + \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial h^K} \frac{\partial h^K}{\partial \omega}$$

上式中 ω, h^K 分别为神经网络 G_ω 的参数和输入，偏导数 $\frac{\partial G_\omega(h^K, x)}{\partial \omega}, \frac{\partial G_\omega(h^K, x)}{\partial h^K}$ 计算较为方便，主要难点在于计算收敛后的隐状态 h^K 关于 ω 的偏导 $\frac{\partial h^K}{\partial \omega}$ ，为此需建立 h^K 与 ω 之间的隐函数关系。根据不动点的性质，若 h^K 收敛至不动点，则

$$h^K = F_\omega(h^K, \tilde{x}) \implies \frac{\partial h^K}{\partial \omega} = \frac{\partial F_\omega(h^K, \tilde{x})}{\partial \omega} + \frac{\partial F_\omega(h^K, \tilde{x})}{\partial h^K} \frac{\partial h^K}{\partial \omega} \implies \frac{\partial h^K}{\partial \omega} = \left(I - \frac{\partial F_\omega(h^K, \tilde{x})}{\partial h^K} \right)^{-1} \frac{\partial F_\omega(h^K, \tilde{x})}{\partial \omega}$$

代入 $\frac{\partial e_\omega}{\partial \omega}$ 的表达式有

$$\frac{\partial e_\omega}{\partial \omega} = \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial \omega} + \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial h^K} \left(I - \frac{\partial F_\omega(h^K, \tilde{x})}{\partial h^K} \right)^{-1} \frac{\partial F_\omega(h^K, \tilde{x})}{\partial \omega}$$

显然偏导 $\frac{\partial e_\omega}{\partial \omega}$ 的结果由 $\frac{\partial e_\omega}{\partial o}, \frac{\partial G_\omega(h^K, x)}{\partial \omega}, \frac{\partial G_\omega(h^K, x)}{\partial h^K}, \frac{\partial F_\omega(h^K, \tilde{x})}{\partial h^K}, \frac{\partial F_\omega(h^K, \tilde{x})}{\partial \omega}, \frac{\partial F_\omega(h^K, \tilde{x})}{\partial \omega}$ 决定，各偏导的计算均相对简单。上式也可进一步简化以消去求逆运算，不妨记 z 满足

$$\begin{aligned} z &= \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial h^K} \left(I - \frac{\partial F_\omega(h^K, \tilde{x})}{\partial h^K} \right)^{-1} \implies z \left(I - \frac{\partial F_\omega(h^K, \tilde{x})}{\partial h^K} \right) = \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial h^K} \\ &\implies z = \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial h^K} + z \frac{\partial F_\omega(h^K, \tilde{x})}{\partial h^K} \end{aligned}$$

因此不妨构造序列 $\{z(n)|n = 0, -1, -2, \dots\}$ 满足

$$z(n) = z(n+1) \frac{\partial F_\omega(h^K, \tilde{x})}{\partial h^K} + \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial h^K}$$

因为加入罚函数后可使得 F_ω 最终收敛为压缩函数，则根据不动点理论序列 $\{z(n)|n = 0, -1, -2, \dots\}$ 也将最终收敛至 z ，从而得到最终的 GNN 模型误差梯度

$$\begin{aligned} \frac{\partial e_\omega}{\partial \omega} &= \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial \omega} + z \frac{\partial F_\omega(h^K, \tilde{x})}{\partial \omega} && (\text{GNN 误差梯度}) \\ z &= \lim_{n \rightarrow -\infty} z(n), \quad z(n) = z(n+1) \frac{\partial F_\omega(h^K, \tilde{x})}{\partial h^K} + \frac{\partial e_\omega}{\partial o} \frac{\partial G_\omega(h^K, x)}{\partial h^K} \end{aligned}$$

8. 作为图神经网络的开山之作，GNN 模型突破了传统神经网络模型无法应用于非欧空间数据的瓶颈，具有重大的意义。其以不动点理论为核心，通过结点信息的传播使整张图达到收敛，再在其基础上进行预测的思想也具有充分的创新性与启发性。但不动点理论作为 GNN 模型的内核，也限制了其更广泛的应用：

- 只将边作为一种传播手段，但并未区分不同边的功能。虽然可以为不同类型的边赋予不同的特征，但相比于其他输入，边对结点隐藏状态的影响实在有限。并且也没有为边设置独立的可学习参数，意味着无法通过模型学习到边的某些特性；
- 不动点理论限制了 GNN 模型于图表示场景的应用。这主要是因为基于不动点的收敛会导致结点之间的隐藏状态间存在较多信息共享，从而导致结点的状态太过光滑 (*over smooth*)，并且属于结点自身的特征信息匮乏 (*less informative*)。

25.2 谱域图卷积神经网络 (*Spectral graph convolutional neural network*)

1. 作为图神经网络的开山之作，传统的基于不动点理论的图神经网络模型在提出后其本身并未得到充分关注，如今已不再是图神经网络模型的主流类型。这其中除了不动点理论本身造成的模型缺陷外，另一个关键原因是图神经网络作为传统神经网络模型于图结构的应用，天然地存在复现传统深度学习模型的趋势。在此过程中既可扩充已有神经网络模型的生命力，也可吸收深度学习领域的最新成果。与卷积、循环、注意力机制等主流神经网络结构相比，基于不动点理论的图神经网络模型显得过于面向图结构，从而削弱了研究者对其的热情；
2. 在 2009 年 Scarselli 首次提出基于不动点理论的图神经网络模型后，真正引发学界对图神经网络关注的是 2013 年 Joan Bruna 等 (Yann LeCun 的学生) 提出的图卷积神经网络模型²。当时正值卷积神经网络模型 (AlexNet) 研究与应用的高潮阶段，而相关研究也是最早期的图卷积神经网络模型；
3. **卷积神经网络模型应用于图结构的主要难点在于图卷积的定义与计算**。传统卷积神经网络模型适用于欧式空间数据，具有固定大小的卷积核；然而在非欧图空间中，节点间的连接属于不规则结构，难以选取固定的卷积核以适应图的不规则性，包括邻居节点数量的不确定性与节点顺序的不确定性；
4. 为设计适用于图结构的卷积核及卷积运算规则，图卷积神经网络的研究逐渐形成两大范式——谱域 (**spectral domain**) 图卷积与空域 (**vertex domain & spatial domain**) 图卷积。谱域卷积又称为频域卷积，基于谱图理论在谱空间中定义图卷积规则，理论基础较为扎实，最早的图卷积神经网络模型即属于此类。空域卷积又称为顶点域卷积，是在图结构上直接定义卷积运算，提出较晚且缺乏理论基础，但因为计算过程直观、与传统卷积神经网络相似度高、模型设计灵活性强，也得到了广泛的关注；
5. **图卷积运算被理解为一种提取节点局部结构特征的方法**。谱域图卷积方法依托于谱图理论，故各类谱域图卷积模型间往往存在清晰的主线脉络，与其它图神经网络模型也存在明显差别。而空域图卷积方法则不存在核心理论，研究者依托不同的猜想与理论定义了海量的空域图卷积方法。实际上可以将所有不依托于图谱理论但又可捕捉图局部信息的图神经网络模型统称为空域图卷积神经网络模型。为不使本节内容过于臃肿，本节后续内容将仅讨论谱域图模型，而部分代表性的空域模型将在后续章节中单独介绍。

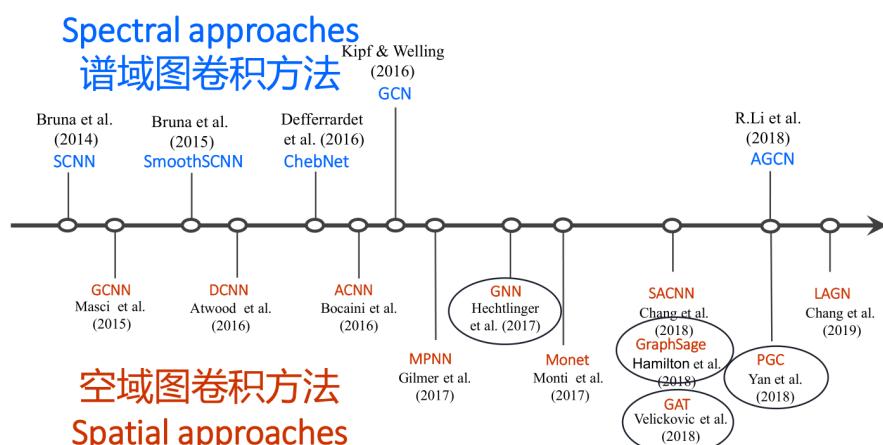


图 25.2 部分代表性的谱域及空域图卷积神经网络模型发展时间线。

²Bruna J, Zaremba W, Szlam A, et al. Spectral networks and locally connected networks on graphs[J]. arXiv preprint arXiv:1312.6203, 2013. <https://arxiv.org/abs/1312.6203>

25.2.1 Spectral CNN

1. 2013 年 Joan Bruna 等提出的最早期的图卷积神经网络模型也是最基础的基于谱域卷积的图神经网络模型，被称为 Spectral CNN。模型对图卷积作出了非常巧妙的定义——因为信号在时域的卷积等于频域上的乘积，故不妨将图卷积运算定义为图信号在图的频域的乘积。第 18.5 节介绍的图傅里叶变换定义了图信号在传统顶点域与谱域间切换的方法，且指出图的谱域可等效为传统信号处理中的“频域”。综上，对于定义在图节点上的信号 x (n 维列向量)，Spectral CNN 模型定义图卷积核为 g (n 维列向量)，并通过图傅里叶变换定义 g, x 间的卷积运算

$$g * x = U(U^\top g \odot U^\top x) = Ug_\theta U^\top x, \quad g_\theta = \text{diag}(U^\top g)$$

上式中 $*$, \odot 分别表示卷积运算与 Hadamard 积； U 为图拉普拉斯矩阵的单位特征向量矩阵， $U^\top g, U^\top x$ 分别表示对 g, x 进行图傅里叶变换，而 $U(U^\top g \odot U^\top x)$ 则表示对 $U^\top g \odot U^\top x$ 进行逆图傅里叶变换。模型定义 $g_\theta = \text{diag}(U^\top g)$ ，训练时以 g_θ 整体作为可学习参数进行误差反向传播更新；

2. 进一步地，考虑图信号为多维矩阵 $X \in \mathbb{R}^{n \times f}$ ，则 Spectral CNN 模型的第 k 个卷积层写为

$$x_j^{k+1} = \sigma \left(U \sum_{i=1}^{f_k} F_{ij}^k U^\top x_i^k \right), \quad j = 1, \dots, f_{k+1} \quad (\text{Spectral CNN 卷积层})$$

上式中 $\sigma(\cdot)$ 为激励函数； $X^k \in \mathbb{R}^{n \times f_k}$ 为第 k 个卷积层的输入，表示每个节点的特征为 f_k 维向量， x_i^k 为 X^k 的第 i 维特征；相应地 $X^{k+1} \in \mathbb{R}^{n \times f_{k+1}}$ 为第 $k+1$ 个卷积层的输入， x_j^{k+1} 为 X^{k+1} 的第 j 维特征； $F_{ij}^k \in \mathbb{R}^{n \times n}$ 为第 k 个卷积层中实现第 i 维输入特征向第 j 维输出特征映射的可学习参数（即为前述的 g_θ ），为对角阵；

3. 作为初代图卷积神经网络，Spectral CNN 在理论层面较为完备，但其在应用过程中缺点也较为突出：

- **计算复杂度大：** 模型需要对拉普拉斯矩阵进行特征值分解，需要 $O(n^3)$ 的计算复杂度。如果每个样本均为一个图，当样本量及图的规模很大时计算负担较大；
- **非局部性连接：** 传统的 CNN 模型在卷积时仅针对像素点邻域内的有限范围，一方面可使得卷积运算时捕捉局部特征，另一方面也可控制卷积核参数数目；而 Spectral CNN 模型在卷积时同时处理图中所有节点的信息，既缺乏物理意义，也使得模型每一层卷积需要 $n \times f_k \times f_{k+1}$ 个参数，造成模型参数过多。

25.2.2 Chebyshev CNN (ChebNet)

1. 在 Spectral CNN 的基础上，Michaël Defferrard 等人于 2016 年提出了基于切比雪夫多项式的图卷积神经网络模型³，有效解决了 Spectral CNN 模型计算复杂度大、非局部连接的问题，标志了谱域图卷积神经网络模型走向成熟。至今模型仍被广泛应用；
2. Spectral CNN 模型的缺点源于其定义了过于通用的可学习卷积核 g_θ ，既导致模型参数过多，也造成图卷积运算形式无法进一步简化以减少计算复杂度。**Chebyshev CNN** 模型的主要创新点即在于其规定了卷积核 g_θ 的具体类型从而巧妙地简化图卷积运算。因为 $g_\theta = \text{diag}(U^\top g)$ ，而 U 显然又是拉普拉斯矩阵特征值矩阵 Λ 的函数，故 g_θ 可视为关于 Λ 的函数 $g_\theta = g_\theta(\Lambda)$ 。进一步地模型假设 g_θ 为关于 Λ 的多项式函数，则 g, x 间的卷积运算表示为

$$g * x = Ug_\theta U^\top x = Ug_\theta(\Lambda)U^\top x = U \left(\sum_{k=0}^K \theta_k \Lambda^k \right) U^\top x = \sum_{k=0}^K \theta_k U \Lambda^k U^\top x$$

由上式可知，假设 g_θ 为关于 Λ 的多项式所带来的最直观地便利即是减少了卷积层的参数——原本 g_θ 具有 n 个参数（即图节点数）；而令 $g_\theta = \sum_{k=0}^K \theta_k \Lambda^k$ 后仅剩 $K+1$ 个参数， K 为模型超参，由研究者定义。然而上述卷积运算还可进一步简化。因为 U 为单位列正交矩阵，有 $U^\top U = I$ ，则

$$g * x = \sum_{k=0}^K \theta_k U \Lambda^k U^\top x = \sum_{k=0}^K \theta_k (U \Lambda U^\top)^k x = \sum_{k=0}^K \theta_k L^k x$$

³Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering[J]. Advances in neural information processing systems, 2016, 29. https://papers.nips.cc/paper_files/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html

可以发现，假设 g_θ 为关于 Λ 的多项式使得原本的卷积运算变为关于拉普拉斯矩阵 L 的多项式，从而完全避免了特征值分解，降低计算复杂度。而且由第 18.5 节介绍的拉普拉斯矩阵的定义可知，拉普拉斯矩阵 L 左乘图信号 x 可提取节点邻域的局部结构特征，而随着 k 的增大 $L^k x$ 所挖掘的局部特征的范围也相应增大。因而只需定义合适的 K ，则按上式进行图卷积即可有效捕捉不同尺度下图的结构信息；

- 综上所述，只需假设 g_θ 为关于 Λ 的多项式即可有效解决 Spectral CNN 模型的主要缺陷。但作者在此基础上进一步引入切比雪夫多项式，模型也由此得名。切比雪夫多项式源于对余弦函数倍角公式的探索。注意到 $\cos 2x = 2\cos^2 x - 1$, $\cos 3x = 4\cos^3 x - 3\cos x$ ，且对于 $\cos nx$ ，其均可展开为 $\cos x$ 的多项式形式，有

$$\cos nx = \begin{vmatrix} \cos x & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 2\cos x & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 2\cos x & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2\cos x & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 2\cos x \end{vmatrix}_{n \times n}$$

对上述行列式按最后一行展开可以得到递推式

$$\cos(n+1)x = 2\cos nx \cdot \cos x - \cos(n-1)x$$

则称满足上述递推式的多项式为第一类切比雪夫多项式

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad T_0(x) = 1, \quad T_1(x) = x, \quad x \in [-1, 1]$$

切比雪夫多项式具有诸多有趣性质，在数值计算中具有重要意义。在 Chebyshev CNN 论文中，作者主要关注切比雪夫多项式的正交性

$$\int_{-1}^1 \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx = \begin{cases} 0 & n \neq m \\ \pi & n = m = 0 \\ \pi/2 & n = m \neq 0 \end{cases}$$

前文中假设卷积核 g_θ 为关于 Λ 的多项式 $g_\theta = \sum_{k=0}^K \theta_k \Lambda^k$ ，而切比雪夫多项式又正好具有正交性，故作者进一步以第一类切比雪夫多项式为正交基 $\{T_k(x)\}_{k=0}^K$ ，对多项式 $g_\theta = \sum_{k=0}^K \theta_k \Lambda^k$ 进行正交分解分解（与傅里叶变换中以简谐函数为正交基分解信号一致），则有

$$g_\theta = \sum_{k=0}^K \theta'_k T_k(\tilde{\Lambda}), \quad \tilde{\Lambda} = \frac{2\Lambda}{\lambda_{\max}} - I$$

此时模型卷积层的参数变为 θ'_k ，其数学意义为正交分解的系数； $\tilde{\Lambda}$ 为缩放平移变换后的拉普拉斯矩阵的特征值矩阵，以满足第一类切比雪夫多项式中要求自变量 $x \in [-1, 1]$ 的约束。将上式代入图卷积运算公式，得到最终 Chebyshev CNN 的卷积层为

$$y = \sigma \left(\sum_{k=0}^K \theta'_k U T_k(\tilde{\Lambda}) U^\top x \right) = \sigma \left(\sum_{k=0}^K \theta'_k T_k(\tilde{L}) x \right), \quad \tilde{L} = \frac{2L}{\lambda_{\max}} - I \quad (\text{Chebyshev CNN 卷积层})$$

按上式计算需估计拉普拉斯矩阵的最大特征值 λ_{\max} 以估计 \tilde{L} 。但估计 λ_{\max} 无需特征值分解，可由幂迭代法 (power iteration) 得到；

- 最后需要补充的是，切比雪夫多项式并非唯一具有正交性的多项式，也不是正交性最优的多项式；而且将一般多项式 $g_\theta = \sum_{k=0}^K \theta_k \Lambda^k$ 正交分解至切比雪夫多项式也并不会带来计算复杂度的进一步改善。选择切比雪夫多项式更多是出于作者团队对其更为熟悉，在实际应用中引入切比雪夫多项式并不一定会带来更多的优势（因为对于神经网络而言优化参数 θ_k 或 θ'_k 并无差别）。在实践中，引入切比雪夫多项式的一个微弱的优势在于其可使得模型对系数扰动更稳定，因为一般多项式 $\sum_{k=0}^K \theta_k \Lambda^k$ 当 k 较大时可能存在 Λ^k

过大的情况，导致梯度爆炸；而引入切比雪夫多项式后对 Λ 作变换使得 $\tilde{\Lambda} \in [-1, 1]$ ，有助于避免梯度爆炸的问题⁴。

25.2.3 一阶 ChebNet (1stChebNet, GCN)

- 在 2016 年 ChebNet 提出标志谱域图卷积神经网络模型走向成熟后，同年 Kipf 等人仅对 ChebNet 作了简单修改后即得到了最具代表性、也是狭义上认为的图卷积神经网络模型——GCN⁵。GCN 因其结构简单、效果极佳的优势进一步启发了更多关于图卷积神经网络模型的研究，同时也使其自身成为目前最常用的图神经网络模型之一；
- 由 GCN 模型的原名“一阶 ChebNet”模型可知，其相比于 ChebNet 模型最直接的改进在于图卷积运算时仅假设卷积核为关于拉普拉斯矩阵 L 的一阶多项式，即令图卷积层超参 $K = 1$ ，此时图卷积运算变为

$$g * x = \theta_0 T_0(\tilde{L})x + \theta_1 T_1(\tilde{L})x, \quad \tilde{L} = \frac{2L}{\lambda_{\max}} - I$$

又因为切比雪夫多项式有 $T_0(x) = 1, T_1(x) = x$ ，则上式进一步写为

$$g * x = \theta_0 x + \theta_1 \tilde{L}x, \quad \tilde{L} = \frac{2L}{\lambda_{\max}} - I$$

对拉普拉斯矩阵 L 进行对称标准化（详见第 18.5 节），则其特征值上限为 2，故令 $\lambda_{\max} = 2$ ，上式有

$$g * x = \theta_0 x + \theta_1 (L^{\text{sym}} - I)x = \theta_0 x + \theta_1 (I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} - I)x = \theta_0 x - \theta_1 D^{-\frac{1}{2}}AD^{-\frac{1}{2}}x$$

在假设为一阶多项式后图卷积层仅剩 2 个参数 θ_0, θ_1 ，再令其共享参数 $\theta = \theta_0 = -\theta_1$ ，此时图卷积层仅剩 1 个参数，图卷积运算也变为

$$g * x = \theta \left(I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \right) x$$

进一步理解上述过程。由拉普拉斯矩阵的统计学性质可知，**令图卷积核仅为拉普拉斯矩阵的一阶多项式可使得图卷积层仅挖掘各节点的一阶邻域结构特征**，与传统 CNN 模型的卷积层效果一致。具体到上式可知， $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}x$ 挖掘了各节点的一阶邻域节点的信息，但因矩阵 $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ 对角线为 0 其无法提取节点本身的信息，该信息由 Ix 补充；

- 作者又进一步介绍了另一种数学形式更简单的图卷积运算方法，可同时挖掘各节点自身及其一阶邻域节点的信息

$$g * x = \theta \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} x, \quad \tilde{A} = A + I, \quad \tilde{D} = D + I = \sum_j \tilde{A}_{ij}$$

上式的本质是对图中的每一节点加上“自环”，使得 $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ 主对角线元素大于 0，从而使其可同时挖掘各节点自身及其一阶邻域节点的信息，与 $I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ 实现相似的效果。需要强调的是，**两种图卷积运算在数学上并不等价，仅是具有相似的统计意义**；

- 对比矩阵 $I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ 与 $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ 发现，前者的特征值介于 $[0, 2]$ 之间，应用于多层神经网络时可能导致计算结果不稳定，引发梯度爆炸；而后的特征值介于 $[0, 1]$ 之间，可视为对前者的重标准化 (renormalization trick)，计算稳定性更强（判断矩阵特征值上下限的方法可参考第 18.5 节中证明对称标准化拉普拉斯矩阵特征值上限的过程）。综上得到 GCN 模型第 l 层卷积层的计算式为

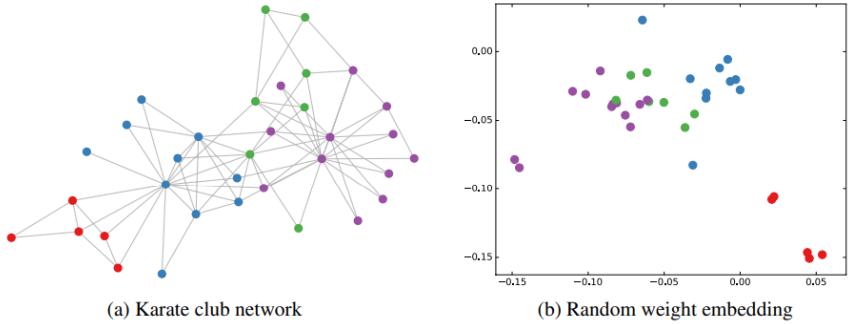
$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} \Theta^{(l)} \right), \quad \tilde{A} = A + I, \quad \tilde{D} = D + I = \sum_j \tilde{A}_{ij} \quad (\text{GCN 卷积层})$$

上式中 $\sigma(\cdot)$ 为非线性激励函数； $H^{(l)} \in \mathbb{R}^{n \times f_l}$ 为第 l 层图卷积层输入特征； $H^{(l+1)}$ 为第 l 层图卷积层输出特征； $\Theta^{(l)} \in \mathbb{R}^{f_l \times f_{l+1}}$ 为图卷积层可学习参数，由参数 θ 组成； n, f_l 分别表示节点数和第 l 层图卷积层中节点向量表示的维数。**得益于 GCN 模型极佳的图局部信息挖掘能力，作者论文中指出，即使不训练仅基于随机初始化参数的 GCN 模型即可提取出非常有效的信息，这是传统 CNN 模型和以往图神经网络模型无法比拟的巨大优势**；

⁴why the Chebyshev polynomials is needed? https://github.com/mdefe/cnn_graph/issues/35

⁵Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016. <https://arxiv.org/pdf/1609.02907.pdf>

图 25.3 使用一个俱乐部会员的关系网络，基于随机初始化的 GCN 模型进行特征提取，得到各节点的嵌入信息。可以发现，原数据中同类别的节点，经随机初始化 GCN 模型提取出的节点嵌入信息已经在空间上自动实现聚类了。



5. 作为 ChebNet 模型的改进版本，GCN 模型继承了前者参数少、局部连接、计算复杂度低的优点，而且因为卷积核仅为拉普拉斯矩阵的一阶多项式使得模型感受野正比于卷积层层数，即随着卷积层的增多模型挖掘图节点邻域特征的范围随之扩大，该特性同样与传统 CNN 模型一致。但 GCN 模型仍存在以下缺陷：
 - **扩展性差：**模型训练时需输入图的邻接矩阵 A ，这意味着训练完成后的模型无法应用于存在新节点的图，具有此类缺陷的模型被称为直推式 (*transductive*) 模型；
 - **局限于浅层：**作者发现在实验中 2 层 GCN 效果取得最好，进一步增加层数无法提升性能，甚至还会削弱模型效果；
 - **不适用于有向图：**GCN 模型仍属于谱域图卷积模型，而谱域图卷积依托于拉普拉斯矩阵，有向图的拉普拉斯矩阵不具备对称性，无法进行特征分解等谱域分析。

6. 从最初的 Spectral CNN 开始，谱域卷积模型日益成熟，至 GCN 提出后达到了顶峰。回顾三代谱域卷积模型发现，尽管其发展轨迹存在清晰的脉络，但在经过诸多简化后最终的 **GCN** 模型已与图傅里叶变换产生了明显的偏离。其卷积运算甚至已不再包含拉普拉斯矩阵，反而是在空域上有清晰的统计学意义——GCN 中所谓的谱域卷积实际上在空域上即是对节点特征矩阵基于边权重作简单的线性加和后再加上一层非线性激励函数。因此，作为谱域卷积模型研究的顶峰，GCN 的提出反而引发了空域卷积的热潮，后续研究者开始从图信号思路角度跳脱出来，直接从邻居节点信息融合角度定义图卷积算子，而谱域卷积的研究也随之转向低潮。

25.2.4 扩散图卷积 (ICLR-18)

1. GCN 提出后图神经网络进入百花齐放的发展阶段，大量跳脱于传统谱域卷积背景的图卷积方法被提出。2018 年提出的扩散图卷积方法即不再基于图傅里叶变换解释图卷积，而是建模为扩散过程 (*diffusion process*)⁶。但扩散图卷积又不同于 GraphSAGE 等空域卷积（见第 25.4.1 节），其仍需图的邻接矩阵作为输入，与经典谱域卷积模型同属于直推式模型；
2. 模型认为图节点信息的空间相关性是信息沿连边扩散导致的，并将扩散建模为带重启的图随机游走 (*random walk with restart*)，即信息从任意节点以概率 α 游走到下一个邻居节点，而以概率 $1 - \alpha$ 留在原节点，概率 α 即为重启概率 (*restart probability*)。同时信息扩散到相邻任意节点的概率与边权成正比，对应的概率转移矩阵为 $D_O^{-1}W$ ，其中 W 为邻接矩阵， D_O 为出度对角阵。上述随机游走属于典型的马尔科夫过程，存在稳态分布 (*stationary distribution*) $P \in \mathbb{R}^{N \times N}$

$$P = \sum_{k=0}^{\infty} \alpha(1-\alpha)^k (D_O^{-1}W)^k$$

式中 k 表示随机游走的轮数， P 的行向量 $P_{i,:} \in \mathbb{R}^N$ 表示节点 i 的信息最终扩散至其它各节点的概率。显然该扩散过程和稳态分布的概念与初代图神经网络（第 25.1 节）所讨论的消息传递和不动点具有相似性；

3. 基于上述图扩散机制设计扩散卷积算子，只需假设扩散总次数 K 有限，并将系数 $\alpha(1-\alpha)^k$ 视为可学习参

⁶Li, Y., Yu, R., Shahabi, C., Liu, Y., 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting, *the 6th International Conference on Learning Representations*, Vancouver Convention Center, Vancouver, BC, Canada. <https://openreview.net/forum?id=SJiHKGWAZ>

数 θ_k , 则扩散卷积定义为

$$f_\theta * x = \sum_{k=0}^{K-1} \left(\theta_{k,1} (D_O^{-1} W)^k + \theta_{k,2} (D_I^{-1} W^\top)^k \right) x, \quad x \in \mathbb{R}^N \quad (\text{扩散图卷积算子})$$

式中 f_θ 表示扩散卷积核; $\theta \in \mathbb{R}^{K \times 2}$ 为卷积核参数; 除了出度对角阵 D_O , 上式进一步引入入度对角阵 D_I 以同时建模沿边方向和逆向的扩散过程;

4. 可以看到, 扩散图卷积是关于 $D_O^{-1}W$ 和 $D_I^{-1}W^\top$ 的多项式, 特别是对于无向图有 $D_O^{-1}W = D_I^{-1}W^\top = D^{-1}W$ 。由第 18.5 节介绍可知, $L^{rw} = I - D^{-1}W$ 被称为随机游走标准化拉普拉斯矩阵, 故无向图的扩散卷积也是关于随机游走标准化拉普拉斯矩阵的多项式, 而经典的切比雪夫图卷积则被建模为关于对称标准化拉普拉斯矩阵的多项式, 因此扩散图卷积模型可视为对经典谱域卷积方法的继承和发展;
5. 最后考虑多通道图节点特征 $X \in \mathbb{R}^{N \times P}$, P 为输入特征维数。计 Q 为输出特征维数, 则扩散图卷积层为

$$H_{:,q} = \sigma \left(\sum_{p=1}^P f_{\Theta_{q,p},:} * X_{:,p} \right), \quad \forall q = 1, \dots, Q \quad (\text{扩散图卷积层})$$

式中 $\Theta \in \mathbb{R}^{Q \times P \times K \times 2}$ 为卷积层参数; $\sigma(\cdot)$ 为非线性激励函数; $H_{:,q}$, $X_{:,p}$ 分别表示第 q 维输入特征和第 p 维输入特征。

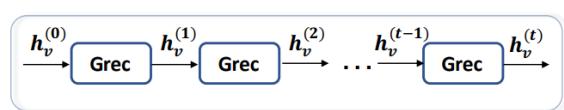
25.3 图递归神经网络 (Graph recurrent neural network, RecGNN)

1. 图递归神经网络是目前主流图神经网络框架中与 2009 年 Scarselli 提出的初代图神经网络模型 (详见第 25.1 节) 最接近的一类。如今不少研究者会把初代图神经网络模型归类为图递归神经网络, 但现代的图递归神经网络模型已与其大不相同;
2. 在 2013 年 Joan Bruna 等提出最早的图卷积神经网络 (第 25.2 节) 后, 初代图神经网络所依托的不动点理论即被证明可以舍去。不同于大批涌向图卷积模型的学者, 另一批学者注意到在去掉初代图神经网络模型中的不动点假设后, 模型的层数 (即前向传播的次数) 即可以取固定值 (原本需重复传播直至收敛到不动点), 并且也不再要求转移函数为压缩映射, 故若再令转移函数为递归神经网络模块, 则模型将自然地转变为定义在图结构上的递归神经网络模型;
3. 与传统的递归神经网络 (第 24.2 节) 一致, 图递归神经网络模型适用于处理时序图结构数据, 也可输出定义在图结构上的序列 (如图上的路径等等)。同样地也因为基本的 RNN 结构存在梯度消失问题, 故现代的图递归神经网络普遍引入 LSTM、GRU 等门控结构 (gating)。但不同于传统递归神经网络模型中门控结构一般用于处理时序相关性, 图递归神经网络中既可包含针对时序信息的门控结构 (time gating), 也可构造针对节点和连边间信息流动的门控结构 (spatial gating)。

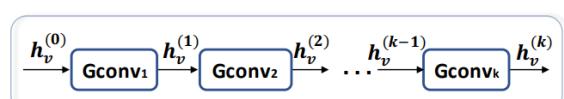
25.3.1 门控图序列神经网络 (gated graph sequence neural network, GGNN)

1. GGNN 模型是最早、也是最具代表性的现代图递归神经网络模型, 由 Li 等于 2015 年提出⁷;

图 25.4 图递归与图卷积神经网络的结构差别。图卷积神经网络 (图 (b)) 中每一卷积层具有独立的结构与参数。图递归神经网络 (图 (a)) 中的所有循环神经网络模块则共享相同的参数, 与初代图神经网络模型类似——初代模型基于同一个传输函数多次迭代。



(a) Recurrent Graph Neural Networks (RecGNNs). RecGNNs use the same graph recurrent layer (Grec) in updating node representations.



(b) Convolutional Graph Neural Networks (ConvGNNs). ConvGNNs use a different graph convolutional layer (Gconv) in updating node representations.

⁷Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks[J]. arXiv preprint arXiv:1511.05493, 2015. <https://arxiv.org/pdf/1511.05493.pdf>

2. 在递归神经网络模型中，隐状态的更新大体由上一步的隐状态和新的输入决定。而在 GGNN 模型中，以每一节点自身的状态为隐状态，而以其邻居节点状态作为输入。具体地，记 $h_v^{(t-1)}$ 表示第 t 次循环前节点 $v \in V$ 的隐状态，则第 t 次循环中节点 v 的输入 $a_v^{(t)}$ 可写为

$$a_v^{(t)} = A_v^\top \begin{bmatrix} h_1^{(t-1)\top} & \dots & h_{|V|}^{(t-1)\top} \end{bmatrix}^\top + b$$

上式旨在聚合节点 v 邻居节点的信息，其中 A_v 表示图邻接矩阵中对应于节点 v 的行向量；

3. 节点隐状态 $h_v^{(t)}$ 的具体更新法则套用循环神经网络中的 GRU 结构（第 24.2 节）。GRU 结构中上一步的隐状态和新输入的信息经过重置门（reset gate）和更新门（update gate）提取过滤后得到新的隐状态。首先介绍重置门，用于控制忽略前一时刻的状态信息的程度，重置门输出结果 $r_v^{(t)}$ 越小说明忽略越多

$$r_v^{(t)} = \sigma(W^r a_v^{(t)} + U^r h_v^{(t-1)}), \quad r_v^{(t)} \in [0, 1]$$

上式中 W^r, U^r 为重置门参数， σ 为 sigmoid 激励函数。基于重置门结果 $r_v^{(t)}$ 滤去上一步隐状态 $h_v^{(t-1)}$ 的无效信息后再与节点 v 的输入信息 $a_v^{(t)}$ 聚合得到初步更新后的节点隐状态 $\tilde{h}_v^{(t)}$

$$\tilde{h}_v^{(t)} = \tanh(W a_v^{(t)} + U(r_v^{(t)} \odot h_v^{(t-1)}))$$

上式中 W, U 同样为 GRU 结构参数。将 $\tilde{h}_v^{(t)}$ 与历史信息 $h_v^{(t-1)}$ 加权聚合即得到节点 v 的最终更新后的隐状态 $h_v^{(t)}$ ，而加权权重 $z_v^{(t)}$ 由更新门确定

$$h_v^{(t)} = (1 - z_v^{(t)}) \odot h_v^{(t-1)} + z_v^{(t)} \odot \tilde{h}_v^{(t)}, \quad z_v^{(t)} = \sigma(W^z a_v^{(t)} + U^z h_v^{(t-1)}), \quad z_v^{(t)} \in [0, 1]$$

上式中 W^z, U^z 为更新门参数， σ 同样为 sigmoid 激励函数。以上即为 GGNN 模型的核心内容。

25.4 经典空域图卷积模型 (*Spatial graph convolutional neural network*)

1. 图卷积模型研究的难点在于图上卷积运算的定义。在图卷积模型研究初期，研究者多聚焦于谱域卷积模型，因为可将卷积运算通过图傅里叶变换变为谱域乘法运算，从而绕开了显式的空域卷积核的定义；
2. 随着图卷积研究热点逐渐转向空域卷积模型，意味着研究者需要直面非欧图结构上卷积运算定义的难点，具体包括以下三方面问题：
 - **邻域的定义：** 欧式数据中邻域的定义非常直观，但对于非欧图结构数据，特别是对于带权有向图，节点邻域的定义还存在诸多讨论空间；
 - **邻域集合尺寸可变的处理：** 传统 CNN 模型的卷积核尺寸是固定的，因为每一元素的邻域集合尺寸固定，而在图结构中每一节点的邻居节点数目往往不固定。如何定义邻域大小不固定下的卷积法则是构建空域图卷积模型所面临的关键问题；
 - **邻域节点位置不固定的处理：** 传统 CNN 模型中的卷积运算本质上是对邻域集合内各元素按固定顺序的加权求和，但对于大多数图模型其任意节点的邻居节点间是不存在相对位置关系的。如何在邻域节点相对位置随机的情况下定义加权求和权重是空域卷积模型设计中最容易被忽略的问题。

需要说明的是，并非所有的空域图卷积模型均考虑了上述三个问题。现阶段大量广受认可的模型仅重点讨论第二个问题。

25.4.1 GraphSAGE 模型 (graph sample and aggregate)

1. GraphSAGE 模型由 Hamilton 等于 2017 年提出⁸，是 GCN 之后另一种非常经典且成功的图卷积神经网络模型，也同 GCN 一样是目前应用最广泛的图神经网络模型之一。不同之处在于 GCN 模型为谱域卷积模型，只适用于无向图，且为直推式（transductive）模型，无法应用于新增节点（详见第 25.2 节）；而 GraphSAGE 模型为空域卷积模型，兼容有向图与无向图，且为归纳式（inductive）模型，模型训练完成后可用于包含新节点的新图；

⁸Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in neural information processing systems, 2017, 30. <https://arxiv.org/pdf/1706.02216.pdf>

2. GraphSAGE 模型将空域图卷积理解为采样 (sample) 和集计 (aggregate) 两步，这也是模型名称的由来。其中“采样”用于解决图卷积时节点邻域集合尺寸可变的问题，而“集计”用于解决邻域节点位置不固定的问题；
3. 首先介绍模型的“采样”环节。模型处理图卷积时节点邻域集合尺寸可变的问题的方式非常直接——效仿传统 CNN 模型选择固定大小的卷积核，再从节点的一阶邻域节点中通过随机采样构建与卷积核尺寸相同的邻域集合用于卷积运算。记模型包含 K 层卷积层，则 GraphSAGE 需要通过采样构建每一节点的 K 阶邻域连通子图。具体地，以节点 u 为中心，首先在其邻居节点中通过随机采样构建一阶邻域节点，再以被采样节点为中心随机采样构建二阶邻域节点，重复进行 K 次得到以 u 为中心的 K 阶邻域连通子图。另外介绍采样的原理，当采样数少于节点邻居数时采用无放回均匀采样，而采样数大于节点邻居数时采用有放回均匀采样；
4. 进一步介绍模型的“集计”方法。针对卷积时邻域节点位置不固定的问题，模型同样作了非常直接的处理——认为邻居节点不需要进行排序，故模型要求集计函数不受输入顺序约束。具体地论文介绍了三种符合要求的集计方法：
 - **Mean Aggregator:** 对邻域内采样的节点特征取平均值，显然平均计算的结果与输入顺序无关；
 - **Pooling Aggregator:** 最大化运算显然也是与输入顺序无关；
 - **LSTM Aggregator:** 因为均值运算和最大化运算都不具备非线性特征提取能力，作者也进一步提出了基于 LSTM 的集计函数，基于 LSTM 模型编码邻居的特征，但需随机打乱各邻居节点的顺序作为输入以使得编码结果与节点顺序无关。
5. 综合上述采样与集计环节，得到 GraphSAGE 模型的空域卷积层如下

$$h_u^k = \sigma(W^k \cdot \text{CONCAT}(h_u^{k-1}, h_{N(u)}^k)), \quad h_{N(u)}^k = \text{AGG}_k(\{h_v^{k-1}, \forall v \in N_k(u)\}) \quad (\text{GraphSAGE 卷积层})$$

上式中集合 $N_k(u)$ 表示经采样得到的节点 u 的 k 阶邻域节点集合； W^k 为第 k 层卷积层的卷积核参数； h_u^k 为第 k 层卷积层提取的节点 k 的特征，可在卷积后加入归一化层作为第 k 层卷积的最终输出 $h_u^k = \frac{h_u^k}{\|h_u^k\|_2}$ ；

6. GraphSAGE 模型适用于有监督问题和无监督问题。若针对有监督问题则可选用常规的误差函数（如交叉熵、均方差等）进行训练；而无监督问题主要指图嵌入 (graph embedding)，旨在基于局部拓扑关系和节点自身特征计算每一节点的向量表示，从而将非欧图结构嵌入欧式向量空间。与一般降维算法的目标一致，图嵌入时的目标函数应使得相近的节点有相似的嵌入，同时相距较远节点的嵌入应有较大差异。为此 GraphSAGE 模型设计了如下图嵌入目标函数

$$J_G(z_u) = -\ln(\sigma(z_u^\top z_v)) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} [\ln(\sigma(-z_u^\top z_{v_n}))]$$

可以看到上式由两部分组成，前项量化了两个相近节点嵌入向量间的距离，后项则欲使得相距较远节点的嵌入向量应有较大差异。其中 z_u 表示节点 u 的嵌入向量，也是模型的最终输出； v 为一个与 u 相近的节点，以 u 为起点作固定步的随机行走得到； v_n 为与 u 较远的节点，被称为负样本 (negative sample)；采样生成 v_n 的分布函数 P_n 被称为负采样分布 (negative sampling distribution)； Q 为负样本数目 (文中设为 10)； σ 为 sigmoid 激励函数。

25.4.2 图注意力神经网络模型 (graph attention network, GAT)

1. 图注意力机制本质上是针对图节点的自注意力机制（详见第 24.4 节），由“深度学习三巨头”之一的 Yoshua Bengio 团队于 2017 年首次提出，并发表于 ICLR 2018⁹。得益于注意力机制在深度学习领域的流行，GAT 模型自提出后迅速引发广泛关注，在不少的任务上都取得了 state of art 的效果。GAT 模型与 GraphSAGE 模型同属于经典的归纳式 (inductive) 空域图卷积模型；
2. 自注意力机制旨在捕捉输入各元素间的两两相关性。而因为图结构中连边的存在天然划定了节点间的相关性，图注意力的运算可考虑两种方式：

⁹ Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *stat*, 1050(20), 10-48550. <https://arxiv.org/abs/1710.10903> (至 2024 年 3 月 7 日论文引用量达 13806)

- 全局图注意力 (**global graph attention**)：计算图中所有节点间的两两相关性，类似于初代谱域图卷积模型 Spectral CNN 在卷积时考虑了所有节点的信息（详见第 25.2 节）。但因为空域图卷积不依托于图邻接矩阵，意味着全局图注意力卷积完全丢弃了图本身的结构信息，可能反而降低学习效果，且运算量大；
- 掩码图注意力 (**mask graph attention**)：只计算节点与其一阶邻域节点间（含自身）的相关性，与如今主流的图卷积模型一致。目前的图注意力模型多采用掩码图注意力的方法。

- 空域卷积设计的关键在于如何处理节点邻域集合尺寸可变的问题。相比于 **GraphSAGE** 模型通过抽样强制构造尺寸一致的邻域集合，**GAT** 模型基于自注意力机制提出了更自然地处理方法。首先自注意力计算时每次仅考虑两个节点，因此邻域集合大小并不干扰计算；另外计算后的自注意力需经 softmax 归一化，因此基于自注意力聚合邻域节点信息本质上是加权平均，同样与邻域集合大小无关；
- 进一步介绍 GAT 模型中基于自注意力机制设计的空域卷积方法。记 $h_i^k \in \mathbb{R}^F$ 表示输入第 k 个卷积层的节点 i 向量表示，卷积后输出为 $h_i^{k+1} \in \mathbb{R}^{F'}$ ，则

$$h_i^{k+1} = \sigma \left(\sum_{j \in N_i} \alpha_{ij} W h_j^k \right), \quad \alpha_{ij} = \frac{\exp \{ \text{LeakyReLU}(a^\top [W h_i^k; W h_j^k]) \}}{\sum_{j' \in N_i} \exp \{ \text{LeakyReLU}(a^\top [W h_i^k; W h_{j'}^k]) \}} \quad (\text{GAT 自注意力卷积层})$$

上式中 N_i 表示节点 i 和其一阶邻居节点构成的集合； $\sigma(\cdot)$ 表示 sigmoid 激励函数； $\alpha_{ij} \in \mathbb{R}$ 表示节点 i 对节点 j 的注意力； $a \in \mathbb{R}^{2F'}, W \in \mathbb{R}^{F' \times F}$ 表示待学习的全连接层参数。显然上式的核心在于节点自注意力 α_{ij} 的计算。首先基于全连接层对每一节点向量表示作线性变换 $W h_i^k \in \mathbb{R}^{F'}$ 以挖掘特征信息，再拼接节点 i, j 的特征信息并线性聚合 $a^\top [W h_i^k; W h_j^k] \in \mathbb{R}$ ，最后通过 LeakyReLU 激励函数（超参设为 0.2）和 softmax 归一化得到节点自注意力 α_{ij} 。由上式可知，虽然图注意力机制被定义为节点间的自注意力机制，但其建模较经典的自注意力模型简单，只包含一个双层全连接网络，而无需三次映射构造 **Q**、**K**、**V**；

- 类比自注意力机制与多头自注意力机制，也可设计针对图结构的多头自注意力 GAT 模型进行空域卷积运算。假设共考虑 n 个自注意力头，则考虑多头自注意力的 **GAT** 模型的每个卷积层包含 n 个自注意力模块，且每个模块的待学习参数 $a^\eta \in \mathbb{R}^{2F'}, W^\eta \in \mathbb{R}^{F' \times F}$ 相互独立

$$h_i^{k+1} = \sigma \left(\frac{1}{n} \sum_{\eta=1}^n \sum_{j \in N_i} \alpha_{ij}^\eta W^\eta h_j^k \right) \quad (\text{GAT 多头自注意力卷积层})$$

25.4.3 ECC (edge-conditioned graph convolution) 模型

- 在针对图的统计和机器学习研究中，图结构更多被作为数据的非欧空间分布的一种表示方法，其中图节点对应数据，连边表示数据间的非均匀联系。因此大多数图模型也更聚焦于节点特征，仅将连边作为消息传递的通道。但在某些细分场合中（如知识图谱、分子化合结构图等等）连边除了反映连接的强度外还具有更丰富的特征属性，这就要求图模型不仅能挖掘节点和拓扑特征，也应建模连边的属性；
- 尽管知名度远不如 GCN、GraphSAGE、GAT 等模型，但 2017 年提出的 **ECC (edge-conditioned graph convolution)** 模型具有图卷积时同时考虑节点属性、连边属性和拓扑结构的优势¹⁰。作为一种空域卷积模型，GCN 采用了 GAT 类似的应对图邻居节点数目可变的方法——令卷积核仅提取相邻的单组节点对的信息，再将共用一个节点的所有节点对信息加权聚合完成图卷积。但不同于 GAT 模型基于连边相邻的两节点的属性提取加权聚合系数，ECC 模型卷积时的聚合系数由连边属性决定，这也是“**edge-conditioned convolution**”的含义；
- 记任意有向图或无向图为 $G = (V, E)$ ，其中节点数为 n ，连边数为 m 。记第 l 层图卷积输入的节点特征为 $X^l \in \mathbb{R}^{n \times d_l}$ ，其中 d_l 表示第 l 层图卷积时节点向量表示的维数。记连边特征为 $L \in \mathbb{R}^{m \times s}$ ，其中 s 表示连边特征维数。ECC 模型卷积时只更新节点特征而连边特征保持不变，但因为连边特征用于估计加权聚合参数，因此模型理论上仍可同时得到节点和连边的嵌入结果。进一步介绍连边特征的提取方法。定义用于第 l

¹⁰Simonovsky, M., & Komodakis, N. (2017). Dynamic edge-conditioned filters in convolutional neural networks on graphs. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3693-3702). https://openaccess.thecvf.com/content_cvpr_2017/html/Simonovsky_Dynamic_Edge-Conditioned_Filters_CVPR_2017_paper.html (至 2024 年 3 月 7 日论文引用量达 1381)

层图卷积的多层全连接网络模型 $F^l : \mathbb{R}^s \rightarrow \mathbb{R}^{d_{l+1} \times d_l}$, 网络参数记为 W^l , 则 ECG 图卷积写为

$$X_i^{l+1} = \frac{1}{|N(i)|} \sum_{j \in N(i)} \Theta_{ji}^l X_j^l + b^l, \quad \Theta_{ji}^l = F^l(L_{ji}|W^l) \quad (\text{ECC 卷积层})$$

上式中可学习参数 W^l, b^l 记为 GCC 模型的卷积核参数; N_i 表示节点 i 和其一阶邻居节点构成的集合。

25.5 时空图模型 (*Spatial-temporal graph model*)

以图卷积为代表的图深度学习模型的目标在于学习非欧空间中的数据结构。时变图是一种特殊的图结构, 一方面图节点位于非欧图空间中, 另一方面节点的属性又沿平坦的时间维度变化。建模此类数据结构时既需要考虑节点间的非欧分布特征, 有需要考虑时间维度上的先后因果关系。大约从 2017 年开始, 随着图神经网络模型基本成熟, 学者开始扩展图卷积模型至时变图场景。此类模型一般被称为时空图模型, 其并非一个明确的模型或一类固定的框架, 而是泛指能同时捕捉图相关性和时间相关性的图卷积模型。时空图模型中学习图相关性的机制与一般图模型无异, 而学习时间相关性的机制则主要参考已有的时序神经网络建模方法, 大体包含循环结构 (第 24.2 节)、时域卷积 (第 24.3.2 节)、和注意力 (第 24.4 节) 等三种方法。

25.5.1 STGCN (spatial-temporal graph convolutional network, IJCAI-18)

1. 本小节主要介绍 2018 年北京大学和北京大数据研究院团队提出的面向交通预测的 STGCN 模型¹¹, 在当时达到了 SOTA 的效果。模型适用于处理图结构不随时间改变, 但节点属性随时间改变的时变图模型;
2. 模型的核心是时空卷积结构, 可堆叠多个时空卷积结构深度挖掘网络状态的时空信息。每个时空卷积结构包含两个时域卷积模块和一个图卷积模块。时域卷积模块和图卷积模块呈三明治结构堆叠。时变图特征首先经时域卷积模块后输入图卷积模块, 最后再经一次时域卷积完成时空特征提取。记第 l 个时空卷积结构的输入为 $v^l \in \mathbb{R}^{M \times n \times C^l}$, 其中 M 表示时间序列长度, n 表示节点数, C^l 表示输入数据通道数, 则时空卷积的数学形式表示为

$$v^{l+1} = \Gamma_1^l *_{\mathcal{T}} \text{ReLU} \left(\Theta^l *_{\mathcal{G}} \left(\Gamma_0^l *_{\mathcal{T}} v^l \right) \right) \in \mathbb{R}^{(M-2K_t+2) \times n \times C^{l+1}} \quad (\text{时空卷积结构})$$

上式中 $*_{\mathcal{T}}$ 表示时域卷积, 两次时域卷积的卷积核分别为 Γ_0^l, Γ_1^l ; $*_{\mathcal{G}}$ 表示图卷积, 卷积核为 Θ^l ;

3. 进一步分别介绍模型的时域卷积模块和图卷积模块:

- 图卷积模块可使用任意图卷积模型。因研究时间背景关系, 作者选择了当时最成熟的 GCN 模型 (即一阶切比雪夫图卷积, 详见第 25.2 节)。图卷积对象为整个图结构, 假设节点于各时间片的属性相互独立;
- 时域卷积模块同样可使用任意针对时间序列的卷积模型。若抛去卷积的限制, 也可选择如 RNN、自注意力等其它适用于时序数据建模的神经网络模型。因研究时间背景关系, 作者结合当时相对先进的 GLU 模块 (详见第 24.3.1 节) 与因果卷积模型 (详见第 24.3.2 节) 搭建门控时序卷积 (**temporal gated convolution**) 模块。具体地, 是将 GLU 模块中的全连接层替换为一维因果卷积运算。记每一节点的时序特征为 $Y \in \mathbb{R}^{M \times C_i}$, 其中 C_i 表示输入时序特征通道数, 又令因果卷积核为 $\Gamma \in \mathbb{R}^{K_t \times C_i \times 2C_o}$, 其中 K_t 表示卷积核长度, C_o 表示卷积输出时序特征通道数, 在没有 padding 的情况下卷积可得到矩阵 $[P \ Q] \in \mathbb{R}^{(M-K_t+1) \times 2C_o}$, 再平分为 $P, Q \in \mathbb{R}^{(M-K_t+1) \times C_o}$ 两部分替代 GLU 模块中的全连接映射结果, 得到如下门控时序卷积形式

$$\Gamma *_{\mathcal{T}} Y = P \odot \sigma(Q) \in \mathbb{R}^{(M-K_t+1) \times C_o} \quad (\text{门控时序卷积})$$

式中 $\sigma(\cdot)$ 表示 sigmoid 激励函数, \odot 表示 Hadamard 积。上式的由来见第 24.3.1 节。时域卷积对象为节点的时序特征向量, 图中各节点独立运算。

¹¹Bing Yu, Haoteng Yin, Zhanxing Zhu. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting (2018): <https://arxiv.org/abs/1709.04875v4>

25.5.2 ASTGCN (attention based spatial-temporal graph convolutional network, AAAI-19)

1. ASTGCN 模型可视为上文 STGCN 模型的改进版本, 发表于 AAAI 2019¹²。其继承了后者基于图卷积和时域卷积学习时空图特征的基本思路, 并在模型结构和输入数据两个层面进行优化设计;
2. 顾名思义, 模型在结构层面的主要创新点在于将注意力机制引入时空图学习过程。具体地, 模型由若干个时空学习单元堆叠组成, 每一个时空学习单元依次包含时空注意力和时空卷积两部分。模型的时空注意力更多作为时空卷积的补充, 其中空间注意力旨在跳出图拓扑结构的约束捕捉任意节点对间的动态交互, 时间注意力则是捕捉节点任意时刻状态间的相互关联;
3. 记第 r 个时空学习单元的输入为 $X_h^{(r-1)} = (X_1, X_2, \dots, X_{T_{r-1}}) \in \mathbb{R}^{N \times C_{r-1} \times T_{r-1}}$ 。其中 N 表示图节点个数; C_{r-1} 表示第 r 个时空学习单元输入的特征通道数; T_{r-1} 表示第 r 个时空学习单元输入的时间切片数, 特别地当 $r=1$ 时令 $T_0 = T_h$, 表示输入的历史数据步长;
4. 首先介绍模型的时间注意力机制

$$\begin{aligned}\widehat{X}_h^{(r-1)} &= (X_1, X_2, \dots, X_{T_{r-1}}) E' \in \mathbb{R}^{N \times C_{r-1} \times T_{r-1}}, \\ E'_{t,\tau} &= \frac{\exp(E_{t,\tau})}{\sum_{\tau=1}^{T_{r-1}} \exp(E_{t,\tau})}, \quad E = V_e \cdot \sigma((X_h^{(r-1)\top} U_1) U_2 (U_3 X_h^{(r-1)}) + b_e)\end{aligned}\quad (\text{时间注意力})$$

其中 $E' \in \mathbb{R}^{T_{r-1} \times T_{r-1}}$ 表示归一化的时间注意力得分, 任意元素 $E'_{t,\tau}$ 表示任意节点 t 时刻状态与 τ 时刻状态的归一化相关性; $V_e, b_e \in \mathbb{R}^{T_{r-1} \times T_{r-1}}$, $U_1 \in \mathbb{R}^N$, $U_2 \in \mathbb{R}^{C_{r-1} \times N}$, $U_3 \in \mathbb{R}^{C_{r-1}}$ 为可学习参数; $\sigma(\cdot)$ 为激励函数。将归一化的时间注意力得分 E' 直接作用于输入 $X_h^{(r-1)}$ 得到隐状态 $\widehat{X}_h^{(r-1)}$ 以强化数据的时间相关性, 服务于后续时空卷积;

5. 模型的空间注意力机制与时间注意力类似, 但归一化空间注意力得分矩阵 $S' \in \mathbb{R}^{N \times N}$

$$S'_{i,j} = \frac{\exp(S_{i,j})}{\sum_{j=1}^N \exp(S_{i,j})}, \quad S = V_s \cdot \sigma((X_h^{(r-1)} W_1) W_2 (W_3 X_h^{(r-1)})^\top + b_s) \quad (\text{空间注意力})$$

式中 $S'_{i,j}$ 表示节点 i, j 状态间的归一化相关性; $V_s, b_s \in \mathbb{R}^{N \times N}$, $W_1 \in \mathbb{R}^{T_{r-1}}$, $W_2 \in \mathbb{R}^{C_{r-1} \times T_{r-1}}$, $W_3 \in \mathbb{R}^{C_{r-1}}$ 为可学习参数。空间注意力得分 S' 并不直接作用于输入数据, 而是补充至邻接矩阵 A 以强化图卷积模型的空间学习能力。以切比雪夫图卷积 (见第 25.2 节) 为例, 即是将卷积核中的 $T_k(\tilde{L})$ 替换为 $T_k(\tilde{L}) \odot S'$ 。其中 \odot 表示矩阵元素相乘, $T_k(\tilde{L})$ 表示经切比雪夫多项式变换的归一化图拉普拉斯矩阵;

6. 完成上述时空注意力后再进行时空卷积, 即构成一个时空学习单元。时空卷积可参考上文 STGCN 模型;
7. 模型的另一个创新在于输入数据层面——除历史数据 $X_h^0 \in \mathbb{R}^{N \times C_0 \times T_h}$, 模型进一步考虑历史先验 $X_d^0 \in \mathbb{R}^{N \times C_0 \times T_d}$, $X_w^0 \in \mathbb{R}^{N \times C_0 \times T_w}$ 作为输入, 分别表示日周期 (daily-periodic) 先验和周周期 (weekly-periodic) 先验。记预测窗口长度为 T_p , 则 T_d, T_w 应为 T_p 的整数倍。以 $\frac{T_d}{T_p} = \frac{T_w}{T_p} = 2$ 为例, 则 X_d^0 记录了预测时段于前两天的状态, X_w^0 记录了预测时段于 7 天和 14 天前的状态。自适应融合 X_h^0, X_d^0, X_w^0 分别经时空学习 (结构相同, 参数不同) 后得到的结果 $\widehat{Y}_h, \widehat{Y}_d, \widehat{Y}_w \in \mathbb{R}^{N \times C_0 \times T_p}$, 得到最终的预测标签

$$\widehat{Y} = W_h \odot \widehat{Y}_h + W_d \odot \widehat{Y}_d + W_w \odot \widehat{Y}_w$$

其中 W_h, W_d, W_w 均为可学习参数;

8. 另外, 论文作者进一步指出可去除模型中的时空注意力机制, 并称仅保留时空卷积和先验输入的简化模型为 MSTGCN (multi-component spatial-temporal graph convolution network) 模型。

25.5.3 GMAN (graph multi-attention network, AAAI-20)

1. GMAN 模型发表于 AAAI 2020¹³。顾名思义, 模型完全基于注意力机制学习图特征的时空相关性, 而上文的 STGCN 模型则完全采用卷积运算。GMAN 模型的另一个特点是其是少有的继承自经典随机游走图

¹²Guo, S., Lin, Y., Feng, N., Song, C., Wan, H., 2019. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 33(01), 922-929. <https://ojs.aaai.org/index.php/AAAI/article/view/3881>

¹³Zheng, C., Fan, X., Wang, C., & Qi, J. (2020). GMAN: A Graph Multi-Attention Network for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01), 1234-1241. <https://ojs.aaai.org/index.php/AAAI/article/view/5477>

嵌入模型 (第 25.7 节) 的新兴图模型。模型基于 Node2Vec 模型 (第 25.7.2 节) 而非图神经网络捕捉图拓扑特征, 因此模型中不存在图卷积或图递归运算;

2. 不同于时空图神经网络一般以时空图结构 \mathcal{G} 整体作为输入, GMAN 模型基于图嵌入技术, 将输入分为节点特征 $X \in \mathbb{R}^{P \times N \times C}$ 和时空嵌入 (ST embedding) $E \in \mathbb{R}^{(P+Q) \times N \times D}$ 两部分。其中 N, C 分别表示图节点数目和节点特征维数 (通道数); P, Q 则分别表示输入和预测的时间步长数; 超参 D 为隐变量特征维数。时空嵌入编码与节点的具体状态无关, 仅与节点邻域拓扑和时间戳有关。另计模型输出为 $\hat{Y} \in \mathbb{R}^{Q \times N \times C}$;
3. 模型采用输入层-编码器-转换注意力单元 (transform attention block)-解码器-输出层结构。其中解码器和编码器均由 L 个时空注意力单元 (ST-attention block) 组成。记第 l 个注意力单元的输入、输出隐变量分别为 $H^{(l-1)}, H^{(l)}$ 。若其位于编码器, 则 $H^{(l-1)}, H^{(l)} \in \mathbb{R}^{P \times N \times D}$; 若位于解码器, 则 $H^{(l-1)}, H^{(l)} \in \mathbb{R}^{Q \times N \times D}$ 。转换注意力单元的输入、输出隐变量则分别为 $H^{(L)} \in \mathbb{R}^{P \times N \times D}, H^{(L+1)} \in \mathbb{R}^{Q \times N \times D}$ 。输入层和输出层均为全连接模型, 分别将模型输入 X 和解码器输出 $H^{(2L+1)}$ 映射为编码器输入 $H^{(0)}$ 和模型输出 \hat{Y} ;
4. 首先介绍时空嵌入编码 E 的生成方法, 涉及空间编码和时间编码两部分。记节点 v_i 和时间戳 t_j 的编码分别为 $e_{v_i}^S, e_{t_j}^T \in \mathbb{R}^D$, 则相应的时空编码

$$e_{v_i, t_j} = e_{v_i}^S + e_{t_j}^T \quad (\text{节点时空编码向量})$$

空间编码 $e_{v_i}^S$ 由 Node2Vec 模型生成并经双层全连接网络映射得到。时间编码 $e_{t_j}^T$ 则考虑了周内时间和日内时间, 经独热编码拼接将任意时间戳编码为 \mathbb{R}^{T+7} 维向量, 并同样并经双层全连接网络映射得到 \mathbb{R}^D 维向量, 其中 T 为单日时间步长数。时空嵌入编码 E 将被应用于后续的时空注意力和转换注意力计算;

5. 进一步介绍时空注意力单元, 包括多头空间注意力、多头时间注意力和门控融合三种机制。假设注意力头数为 K , 则单个注意力头的特征编码维数 $d = D/K$:

- 空间注意力旨在实时捕捉任意两节点间的相关性, 既与节点实时状态有关, 又与固有特征 (时空编码) 有关。记第 l 个时空注意力单元对节点 v_i 于时间 t_j 特征的空间注意力编码为 $hs_{v_i, t_j}^{(l)} \in \mathbb{R}^d$, 有

$$hs_{v_i, t_j}^{(l)} = \left\| \sum_{k=1}^K \alpha_{v_i, v}^{(k)} \cdot f_{s, 3}^{(k)}(h_{v_i, t_j}^{(l-1)}) \right\|, \quad \alpha_{v_i, v}^{(k)} = \frac{\exp(s_{v_i, v}^{(k)})}{\sum_{v_r \in \mathcal{V}} \exp(s_{v_i, v_r}^{(k)})}, \quad s_{v_i, v}^{(k)} = \frac{\langle f_{s, 1}^{(k)}(h_{v_i, t_j}^{(l-1)} \| e_{v_i, t_j}), f_{s, 2}^{(k)}(h_{v_i, t_j}^{(l-1)} \| e_{v_i, t_j}) \rangle}{\sqrt{d}} \quad (\text{多头空间注意力})$$

式中 $s_{v_i, v}^{(k)} \in \mathbb{R}^d$ 表示第 k 个注意力头计算的节点 v_i 与任意节点 v 于 t_j 时刻的注意力评分, 将其归一化得到注意力权重 $\alpha_{v_i, v}^{(k)}$; $f_{s, 1}^{(k)}(\cdot), f_{s, 2}^{(k)}(\cdot), f_{s, 3}^{(k)}(\cdot)$ 为第 k 个注意力头的非线性映射函数, 由单层全连接网络和非线性激励组成; $\langle \cdot, \cdot \rangle$ 表示向量内积; $\|\cdot\|$ 表示向量拼接; \mathcal{V} 表示网络所有节点集合。上述空间注意力计算时考虑了图中任意节点间的相关性, 而非图神经网络仅考虑低阶邻居节点;

- 时间注意力旨在捕捉节点状态于不同时间的相关性, 既与节点实时状态有关, 又与固有特征 (时空编码) 有关。记第 l 个时空注意力单元对节点 v_i 于时间 t_j 特征的时间注意力编码为 $ht_{v_i, t_j}^{(l)} \in \mathbb{R}^d$, 有

$$ht_{v_i, t_j}^{(l)} = \left\| \sum_{t \in \mathcal{N}_{t_j}} \beta_{t_j, t}^{(k)} \cdot f_{t, 3}^{(k)}(h_{v_i, t}^{(l-1)}) \right\|, \quad \beta_{t_j, t}^{(k)} = \frac{\exp(u_{t_j, t}^{(k)})}{\sum_{t_r \in \mathcal{N}_{t_j}} \exp(u_{t_j, t_r}^{(k)})}, \quad u_{t_j, t}^{(k)} = \frac{\langle f_{t, 1}^{(k)}(h_{v_i, t_j}^{(l-1)} \| e_{v_i, t_j}), f_{t, 2}^{(k)}(h_{v_i, t}^{(l-1)} \| e_{v_i, t}) \rangle}{\sqrt{d}} \quad (\text{多头时间注意力})$$

式中 $u_{t_j, t}^{(k)} \in \mathbb{R}^d$ 表示第 k 个注意力头计算的 t_j 时刻节点 v_i 状态关于 t 时刻状态的注意力评分, 将其归一化得到注意力权重 $\beta_{t_j, t}^{(k)}$; $f_{t, 1}^{(k)}(\cdot), f_{t, 2}^{(k)}(\cdot), f_{t, 3}^{(k)}(\cdot)$ 为第 k 个注意力头的非线性映射函数, 由单层全连接网络和非线性激励组成; \mathcal{N}_{t_j} 表示位于 t_j 时刻之前的时间戳集合;

- 基于门控机制自适应融合空间注意力编码 $H_S^{(l)}$ 与时间注意力编码 $H_T^{(l)}$, 得到时空注意力的输出 $H^{(l)}$

$$H^{(l)} = z \odot H_S^{(l)} + (1 - z) \odot H_T^{(l)} + H^{(l-1)}, \quad z = \sigma(H_S^{(l)} W_{z, 1} + H_T^{(l)} W_{z, 2} + b_z) \quad (\text{门控融合})$$

式中 $W_{z, 1}, W_{z, 2}, b_z$ 为可学习参数; $\sigma(\cdot)$ 为 sigmoid 激励函数; \odot 表示矩阵元素级乘法。

6. 上文说明, GMAN 模型在计算空间注意力时不受图拓扑结构的约束, 考虑了任意节点对状态间的相关性。这意味着模型需要计算 N^2 次空间注意力评分, 当图节点数较多时需要的计算资源极大。为此模型针对大规模图结构进一步提出了分组空间注意力计算方法。将图结构随机均分为 G 组子图, 每组节点数 $M = N/G$, 则空间注意力运算被拆分为组内空间注意力 (*intra-group spatial attention*) 和组间空间注意力 (*inter-group spatial attention*) 两个环节:

- 组内空间注意力并行计算各子图内部各节点间的多头空间注意力, 各组共享注意力模型参数;
- 对各子图按组内空间注意力得到各节点局部隐状态后基于最大值池化提取每一子图的全局向量。基于各组子图的全局向量进行注意力计算, 得到各子图的表示向量, 这一过程即为组间空间注意力;
- 将各子图表示向量与子图内部各节点局部隐向量相加, 最终得到考虑全图特征的隐状态 $H_S^{(l)}$ 。

以上即为分组空间注意力的计算方法, 无论组内或组间空间注意力均参考前文多头空间注意力机制计算。按上述分组计算方法, 共需计算 $GM^2 + G^2 = NM + (N/M)^2$ 次注意力评分。对该式取关于 M 的极小值, 可知当 $M = \sqrt[3]{2N}$ 时总计算复杂度最低;

7. 最后介绍 GMAN 模型的转换注意力机制。转换注意力本质上也属于时间注意力, 其旨在捕捉预测窗口内各时间戳与历史窗口内时间戳的两两相关性。记转换注意力单元对节点 v_i 于时间 t_j ($t_j = t_{p+1}, \dots, t_{p+Q}$) 特征的转换注意力编码为 $h_{v_i, t_j}^{(L+1)} \in \mathbb{R}^D$, 有

$$h_{v_i, t_j}^{(L+1)} = \left\| \sum_{k=1}^K \left\{ \sum_{t=t_1}^{t_p} \gamma_{t_j, t}^{(k)} \cdot f_{t, 3}^{(k)}(h_{v_i, t}^{(L-1)}) \right\} \right\|, \quad \gamma_{t_j, t}^{(k)} = \frac{\exp(\lambda_{t_j, t}^{(k)})}{\sum_{t_r=t_1}^{t_p} \exp(\lambda_{t_j, t_r}^{(k)})}, \quad (\text{多头转换注意力})$$

$$\lambda_{t_j, t}^{(k)} = \frac{\langle f_{t, 1}^{(k)}(e_{v_i, t_j}), f_{t, 2}^{(k)}(e_{v_i, t}) \rangle}{\sqrt{d}}$$

对比转换注意力与时间注意力的计算机制可知, 转换注意力评分 $\lambda_{t_j, t}^{(k)}$ 仅与时间戳 t_j, t 和节点 v_i 固有属性有关, 而与节点实时 (隐) 状态无关, 从而缓解长时段预测过程中的误差累积;

8. 以上即为 GMAN 模型的基本结构。模型包含时空注意力单元数目 $2L$ 、注意力头数 K 和单个注意力头的维数 d 共三项超参。作者于实验中发现令 $L = 3, K = 8, d = 8$ 时效果最好。

25.5.4 AGCRN (adaptive graph convolutional recurrent network, NeurIPS-20)

1. AGCRN 模型发表于 NIPS 2020¹⁴, 与上文的 STGCN 和 ASTGCN 模型相比, AGCRN 模型在实验中往往具有更强的时空图预测效果。顾名思义, 模型基于循环神经网络结构学习时间相关性, 而主要创新在于图学习模块——包括自适应节点嵌入 (**node adaptive parameter learning**) 和自适应图生成 (**data adaptive graph generation**) 两部分;

2. 模型的图学习机制可视为经典图嵌入和图卷积技术的融合。考虑经典的 GCN 模型 (第 25.2 节)

$$Z = \sigma \left(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} X \Theta + b \right)$$

式中 $X \in \mathbb{R}^{N \times C}$, $Z \in \mathbb{R}^{N \times F}$ 分别表示图卷积的输入和输出特征矩阵; C, F 分别为输入特征和输出特征的维数; 而 $\Theta \in \mathbb{R}^{C \times F}$, $b \in \mathbb{R}^F$ 则是卷积核可学习参数。研究认为经典图卷积普遍存在以下缺陷:

- 图卷积核只能捕捉节点的共有模式, 忽视了节点的特有模式。表现为卷积核参数只和节点特征有关, 但与节点本身无关, 其背后的假设是图节点间的相互作用只由节点特征和邻接关系决定。但在实际应用中, 具有一致拓扑连接的节点仍可能因为其它未考虑的因素表现出不同的互作模式;
- 预定义的图结构无法捕捉节点间的所有相互作用。目前的图节点连接判定方法主要由距离函数或相似度函数决定, 这一过程必然存在信息丢失, 而图的拓扑特征 $\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$ 并不随卷积过程动态更新, 从而只能得到次优解。

3. 为使得图卷积过程可捕捉节点的特有模式, 最直观思路是扩大图卷积核的规模使其能为每个节点引入一组单独的参数 $\Theta \in \mathbb{R}^{N \times C \times F}$ 。然而该设置将显著增加参数量和过拟合风险¹⁵。为此, AGCRN 模型借鉴张量

¹⁴Bai, L., Yao, L., Li, C., Wang, X., Wang, C., 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc., Vancouver, BC, Canada, p. Article 1494. <https://arxiv.org/abs/2007.02842>

¹⁵需要说明的是此时 $X\Theta$ 将不再是矩阵乘法而是张量运算, 具体的运算为 $[X\Theta]_{ik} = \sum_j x_{ij}\theta_{ijk}$ 。

分解的思路提出如下图卷积算法

$$Z = \sigma \left(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} X E_G W_G + E_G b_G \right) \quad (\text{自适应节点嵌入 GCN})$$

$E_G \in \mathbb{R}^{N \times d}$, $W_G \in \mathbb{R}^{d \times C \times F}$, $b_G \in \mathbb{R}^{d \times F}$ 为卷积核可学习参数, 其中 E_G 为节点嵌入矩阵 (node embedding matrix), W_G 为权重池化矩阵 (weight polling), b_G 为偏置池化矩阵 (bias polling), 超参 $d \ll N$ 。上式的基本思路是通过低秩分解实现 $E_G W_G \in \mathbb{R}^{N \times C \times F}$, $E_G b_G \in \mathbb{R}^{N \times F}$, 既保证图卷积可区分不同节点的固有模式, 又避免因参数量过大导致的过拟合问题。上式也因此被称为自适应节点嵌入图卷积;

4. 为自适应捕捉节点间的相互作用, 核心在于自适应更新图结构。又因为图结构应随学习过程动态调整, 故可令拓扑关系与卷积核参数一同作为可学习对象, 而无需显式构造图结构。AGCRN 模型借鉴经典的基于矩阵分解的图嵌入 (见第 25.6 节) 思想, 提出如下自适应图生成图卷积算法

$$Z = \sigma \left(\text{softmax} \left(\text{ReLU} \left(E_A E_A^\top \right) \right) X \Theta + b \right) \quad (\text{自适应图生成 GCN})$$

上式本质上即是令 $\text{softmax} \left(\text{ReLU} \left(E_A E_A^\top \right) \right)$ 近似 $\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$, 其中 $E_A \in \mathbb{R}^{N \times d_e}$ 为可学习的节点嵌入矩阵, 从而动态更新图的拓扑关系;

5. 结合上述自适应节点嵌入和自适应图生成 GCN 计算过程, 令 $E_G = E_A = E$, 并引入 GRU 结构 (见第 24.2 节) 捕捉时序特征, 最终得到具有时空学习能力的 AGCRN 模型如下

$$\begin{aligned} \widetilde{A} &= \text{softmax} \left(\text{ReLU} \left(E E^\top \right) \right), \quad z_t = \sigma \left(\widetilde{A} [X_{:,t}, h_{t-1}] E W_z + E b_z \right), \quad r_t = \sigma \left(\widetilde{A} [X_{:,t}, h_{t-1}] E W_r + E b_r \right), \\ \hat{h}_t &= \tanh \left(\widetilde{A} [X_{:,t}, r \odot h_{t-1}] E W_h + E b_h \right), \quad h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t \end{aligned} \quad (\text{AGCRN})$$

式中 $X_{:,t}$ 为第 t 个时间片的输入节点特征, $[::]$ 表示矩阵拼接;

6. 以上即为 AGCRN 模型的具体结构。与上文介绍的 GMAN 模型相比, AGCRN 模型同样无需显示构造图结构, 而是通过节点嵌入向量表征拓扑特征。但 GMAN 模型完全放弃了图卷积的思路, 而是基于空间自注意力机制学习图的拓扑关系。AGCRN 模型的图学习机制则源于经典的图卷积运算, 因此可视其为经典图嵌入和图卷积技术的融合。

25.6 基于矩阵分解的图嵌入模型

如本章引言所述, 图嵌入 (又称图表示学习) 和图神经网络属于图机器学习研究中不同但相关的两个方向。两者均要求挖掘数据于非欧图空间中的分布特征, 但前者是为了将非欧数据嵌入至欧式空间中, 而后者的目标是直接实现端到端预测。在图学习研究的早期阶段, 因为图神经网络乃至一般的神经网络模型尚未成熟, 图嵌入方法涉及多种技术路线, 将图数据嵌入欧式空间后即可基于其它成熟的预测模型进行预测。而随着图神经网络模型的发展, 模型在图特征挖掘方面的优势愈加明显, 图嵌入的技术路线逐渐被图神经网络垄断 (参考第 25.2 节介绍的无需专门训练即可有效嵌入节点信息的 GCN 模型及第 25.4.1 节介绍的面向图嵌入设计的 GraphSAGE 模型目标函数)。

图嵌入在机器学习领域中属于降维问题, 而最主流的降维技术路线为矩阵分解 (matrix factorization)。受此影响, 在图嵌入最早期的研究中涌现了一批基于矩阵分解的模型。这类模型往往以重建图的邻接矩阵为目标, 学习各个节点的表示。如今看来, 这些模型有较大的局限性, 但其中仍不乏启发性的研究。

25.6.1 图因子分解 (graph factorization)

1. 图因子分解模型由 Google 于 13 年提出¹⁶。方法的思想非常直接——对图的邻接矩阵作矩阵分解, 从而得到节点的向量表征;
2. 记 y_{ij} 表示连边 (i, j) 的权重, h_i 表示待求解的节点 i 的向量表示, 则模型的目标函数如下

$$(\min) \mathcal{L} = \frac{1}{2} \sum_{(i,j) \in E} (y_{ij} - \langle h_i, h_j \rangle)^2 + \frac{\lambda}{2} \sum_i \|h_i\|^2$$

¹⁶Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., & Smola, A. J. (2013). Distributed large-scale natural graph factorization. *Proceedings of the 22nd International Conference on World Wide Web*. <https://sci-hub.se/10.1145/2488388.2488393>

其中 $\langle h_i, h_j \rangle$ 表示向量内积。上式基于欧式距离量化 y_{ij} 与 $\langle h_i, h_j \rangle$ 的误差，本质上是假设 y_{ij} 服从 $\langle h_i, h_j \rangle$ 为期望的高斯分布。实际应用中可根据 y_{ij} 的实际情况选择合适的分布假设从而保证向量表示的质量；

3. 上式的求解也非常简单，对目标函数求梯度有

$$\frac{\partial \mathcal{L}}{\partial h_i} = - \sum_{j \in N(i)} (y_{ij} - \langle h_i, h_j \rangle) h_j + \lambda h_i$$

上式中 $N(i)$ 表示节点 i 的邻居节点。基于上式可直接逐节点进行梯度下降优化，但计算时每个节点需要被重复访问，故复杂度并不正比于节点数，而是与连边数正相关。作者提出直接以边为单位进行迭代更新的算法

$$h_i \leftarrow h_i + \eta [(y_{ij} - \langle h_i, h_j \rangle) h_j + \lambda h_i], \quad (i, j) \in E$$

按上式更新时每次仅选取连接 i 的一条边 (i, j) 估计梯度，因此上式属于序列随机梯度下降算法 (**sequential stochastic gradient descent**)；

4. 图因子分解模型的建模与一般的矩阵分解基本一致（如概率矩阵分解，见第 20.5 节），甚至更为简单。一般矩阵分解会将原矩阵分解为至少两个矩阵，而图因子分解模型对 Y 分解后仅得到一个矩阵 H 。因为模型嵌入时只计算节点的一个向量表征，意味着模型认为节点 i 在连边 (i, j) 和连边 (j, i) 中发挥着相同的作用，故模型仅适用于无向图嵌入。另外，模型仅适用于边权重 y_{ij} 为标量的情况，而且因为只考虑邻接矩阵的分解而忽视了节点间的高阶关系。

25.6.2 GraRep 算法

1. GraRep 算法提出于 2015 年¹⁷。此时图机器学习领域已愈发被关注，而图神经网络模型尚未完全成熟，正处于图嵌入技术路线最丰富的时期。尽管属于技术路线最为传统的矩阵分解流派，GraRep 算法远较早期的图因子分解算法成熟，且效果不逊于同期大热的 DeepWalk 算法；
2. 与其它图嵌入方法相比，GraRep 算法的一个重要特点在于其表示学习区分了节点的高阶邻接特征。具体地，算法提出了一个将节点的 k 阶信息映射到独立子空间上的方法，并通过拼接不同阶信息的表示向量最终捕捉一个节点的所有 K 阶信息的表达。记 A, D 分别表示图的邻接矩阵，则算法首先定义归一化的邻接矩阵 \tilde{A} 为单步概率转移矩阵 (**1-step probability transition matrix**)

$$\tilde{A} = D^{-1}A \iff \tilde{A}_{ij} = \frac{A_{ij}}{\sum_{j'} A_{ij'}}$$

显然元素 \tilde{A}_{ij} 可表示为节点 i 经过一步转移到节点 j 的概率，也可记为 $p(j|i)$ 。 \tilde{A} 和 A 一样体现了图的一阶连接关系，基于矩阵分解重构 \tilde{A} 即可学习这一关系。而为了学习图的 k 阶连接关系，则只需重构图的 k 步概率转移矩阵 (**k -step probability transition matrix**)

$$\tilde{A}^k = \underbrace{\tilde{A} \cdots \tilde{A}}_k \iff \tilde{A}_{ij}^k = \sum_{j'} \tilde{A}_{ij'}^{k-1} \tilde{A}_{j'j}$$

显然元素 \tilde{A}_{ij}^k 可表示为节点 i 经过 k 步转移到节点 j 的概率，也可记为 $p_k(j|i)$ ；

3. 不同于图因子分解模型，GraRep 算法基于两个矩阵 W, C 还原图的 k 阶邻接信息 \tilde{A}^k ，与一般的矩阵分解模型一致。但算法并不直接对 \tilde{A}^k 作矩阵分解，而是参考当时已广泛流行的词嵌入技术设计如下目标函数

$$(\min) J_k = \sum_{i \in V} J_k(i), \quad J_k(i) = - \left(\sum_{j \in V} p_k(j|i) \ln \sigma(w_i^\top c_j) \right) - \lambda \mathbb{E}_{j' \sim p_k(V)} [\ln \sigma(-w_i^\top c_{j'})]$$

上式中 σ 为 sigmoid 激励函数； w_i, c_i 分别表示矩阵 W, C 的第 i 个列向量，均可视为节点 i 的向量表示，其中 w_i 是节点 i 为源节点时的表达， c_i 是节点 i 为其它节点的上下文节点 (**context node**) 时的表达； $J_k(i)$

¹⁷ Shaosheng Cao, Wei Lu, and Qiongkai Xu (2015). *GraRep: Learning graph representations with global structural information*. In Proceedings of the 24th ACM international conference on information and knowledge management. https://www.researchgate.net/publication/301417811_GraRep

表示学习节点 i 的 k 阶邻接表达时的误差函数，同样的形式也应用于 GraphSAGE 模型的图嵌入任务中（见第 25.4.1 节）； $J_k(i)$ 由两部分组成，前项旨在最小化 i 与其 k 阶邻居节点 j 嵌入向量间的距离，后项则欲使 i 与其它节点 j' 的嵌入向量有较大差异； j' 被称为负样本（negative sample），采样生成 j' 的分布函数 $p_k(V)$ 被称为负采样分布（negative sampling distribution），超参 λ 为负样本数目。需要说明的是，负采样时并不要求负样本 j' 一定不是 i 的 k 阶邻居。按期望定义展开 $J_k(i)$ 的第二项有

$$\mathbb{E}_{j' \sim p_k(V)} [\ln \sigma(-w_i^\top c_{j'})] = \sum_{j' \in V} p_k(j') \ln \sigma(-w_i^\top c_{j'}) , \quad p_k(j') = \sum_{i'} q(i') p_k(j'|i')$$

负采样时节点 j' 被采样的概率 $p_k(j')$ 定义为从任意节点出发经 k 步后到达 j' 的概率； $q(i')$ 为节点 i' 被作为 k 次转移的源节点的概率。令 $q(i') = \frac{1}{N}$ 并将 j' 改写为 j 重新整理 $J_k(i)$ 的表达式有

$$\begin{aligned} J_k(i) &= - \left(\sum_{j \in V} p_k(j|i) \ln \sigma(w_i^\top c_j) \right) - \lambda \mathbb{E}_{j \sim p_k(V)} [\ln \sigma(-w_i^\top c_j)] \\ &= \sum_{j \in V} (-p_k(j|i) \ln \sigma(w_i^\top c_j) - \lambda p_k(j) \ln \sigma(-w_i^\top c_j)) \\ &= \sum_{j \in V} \left(-\tilde{A}_{ij}^k \ln \sigma(w_i^\top c_j) - \frac{\lambda}{N} \sum_{i' \in V} \tilde{A}_{i'j}^k \ln \sigma(-w_i^\top c_j) \right) = \sum_{j \in V} J_k(i, j) \end{aligned}$$

式中 $J_k(i, j)$ 被称为局部损失（local loss），表示给定任意节点对 (i, j) 的表示学习误差。当向量 w_i, c_j 的维度足够大，则可近似认为各节点对点积 $w_i^\top c_j$ 结果相互独立，即 $J_k(i, j)$ 相互独立，此时优化 J_k 即可简化为独立地优化 $J_k(i, j)$ ；

- 作者巧妙地将 $J_k(i, j)$ 优化任务建模为矩阵分解问题以快速求解 W, C 。首先求 $J_k(i, j)$ 关于 $w_i^\top c_j$ 的极值点

$$\frac{\partial J_k(i, j)}{\partial w_i^\top c_j} = 0 \implies w_i^\top c_j = \ln \frac{N \cdot \tilde{A}_{ij}^k}{\lambda \left(\sum_{i'} \tilde{A}_{i'j}^k \right)}$$

当 $w_i^\top c_j$ 等于上式时 $J_k(i, j)$ 取最优。因此只需构造矩阵 $X^k \in \mathbb{R}^{N \times N}$ 满足 $X_{ij}^k = \ln \frac{N \cdot \tilde{A}_{ij}^k}{\lambda \left(\sum_{i'} \tilde{A}_{i'j}^k \right)}$ ，再对 X^k 作分解 $X^k \simeq W^\top C$ 即可得到 W, C 。作者具体采用低秩 SVD 分解（low-rank SVD decomposition）

$$W^k = \left[U_d^k (\Sigma_d^k)^{1/2} \right]^\top, \quad C^k = (\Sigma_d^k)^{1/2} V_d^k, \quad X^k \simeq U_d^k \Sigma_d^k V_d^k$$

上式中 $W^k, C^k \in \mathbb{R}^{N \times d}$ 为对图的 k 阶邻接特征的表征结果，超参 d 表示图节点表示向量的维数； $U_d^k \in \mathbb{R}^{N \times d}, \Sigma_d^k \in \mathbb{R}^{d \times d}, V_d^k \in \mathbb{R}^{d \times N}$ 为 X^k 的低秩 SVD 分解结果，分别为 X^k 左奇异矩阵、对角奇异值矩阵、右奇异矩阵的前 d 列，是在秩 d 约束下对 X^k 的最优拟合结果；

- 最后为避免 SVD 分解受异常值的影响，一般在 SVD 分解前令 $X^k = \max\{0, X^k\}$ ，其背后的逻辑是如果节点对 (i, j) 之间的 k 阶转移概率对节点 i 来说占比非常少（即 $\tilde{A}_{ij}^k / \left(\sum_{i'} \tilde{A}_{i'j}^k \right)$ 趋于 0），则一概将其赋值为一个足够小的固定值 λ/N 。

25.6.3 LINE (large-scale information network embedding) 模型

- 2015 年提出的 LINE 模型可视为图因子分解和 GraRep 算法的折中。因为简单而合理的目标函数设计取得极好的表现，成为图嵌入研究早期阶段可与大热的 DeepWalk、Node2Vec 模型并列的经典模型¹⁸。不同于 GraRep 算法旨在考虑图的高阶邻接信息，LINE 模型仅区分节点的一阶相似度（first-order proximity）和二阶相似度（second-order proximity），且一阶相似度模型和二阶相似度模型在数学上并不完全一致；

¹⁸Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015, May). Line: Large-scale information network embedding. In Proceedings of the 24th international conference on world wide web (pp. 1067-1077). <https://arxiv.org/pdf/1503.03578.pdf>. (至 2024 年 3 月 4 日，论文引用量超 6000 次，远超同期的图因子分解和 GraRep 模型，约为 DeepWalk 和 Node2Vec 的 1/2 至 2/3.)

2. 考虑一阶相似度的 LINE 模型对标图因子分解模型，对于每一节点 i 仅学习一个向量表征 v_i ，因而也只适用于无向图嵌入。一阶相似度聚焦节点对的一阶邻接关系。对于节点对 (i, j) ，若存在直连连边，则一阶相似度大于 0，且边权 w_{ij} 越大一阶相似度越接近 1，向量表征结果 v_i, v_j 也应越接近。故定义给定向量表征 v_i, v_j 下节点对 (i, j) 存在直连连边的联合概率 $p_1(v_i, v_j)$ 和基于边权 w_{ij} 估计的经验概率 $\hat{p}_1(i, j)$

$$p_1(v_i, v_j) = \frac{1}{1 + \exp\{-v_i^\top v_j\}}, \quad \hat{p}_1(i, j) = \frac{w_{ij}}{\sum_{(i', j') \in E} w_{i' j'}}$$

上式中下标“1”表示一阶相似度的 LINE 模型。 $p_1(v_i, v_j)$ 与向量内积 $v_i^\top v_j$ 正相关，因向量内积即是一种经典的计算向量相似度的方法，再由 sigmoid 函数将其映射为 0-1 之间的概率。采用 sigmoid 函数意味着计算时仅考虑节点对 (i, j) 间的邻接关系而不考虑 i 的其它邻居，这便是一阶相似度的内涵。自然地一阶相似度模型应使得概率 $p_1(v_i, v_j)$ 与经验概率 $\hat{p}_1(i, j)$ 一致。基于 KL 散度量化概率间距离，得到一阶相似度 LINE 模型的目标函数为

$$(\min) \quad \mathcal{L}_1 = - \sum_{(i, j) \in E} w_{ij} \ln p_1(v_i, v_j) \quad (\text{一阶相似度模型})$$

3. 考虑二阶相似度的 LINE 模型适用于无向图或有向图嵌入，因此对于每一节点 i 仅学习两个向量表征 v_i, v'_i ，前者表征 i 的邻域结构信息，后者则是在 i 作为其它节点的邻居时使用。一般默认图为有向图。“二阶相似度”旨在表征两个节点的邻居间的关系。记 i 与图中所有节点的边权为 $p_i = (w_{i0}, \dots, w_{i|V|})$ ，则 p_i 与 p_j 越接近则表示两节点的二阶相似度越高，则两者的嵌入结果也应越接近。对于有向边 (i, j) ，同样令 w_{ij} 与 $v'_j^\top v_i$ 正相关，则给定 v_i 条件下节点 j 被 i 连接的条件概率 $p_2(v'_j | v_i)$ 和经验条件概率 $\hat{p}_2(j|i)$ 定义为

$$p_2(v'_j | v_i) = \frac{\exp\{v'_j^\top v_i\}}{\sum_{k \in V} \exp\{v'_k^\top v_i\}}, \quad \hat{p}_2(j|i) = \frac{w_{ij}}{\sum_{k \in V} w_{ik}}$$

上式中下标“2”表示二阶相似度的 LINE 模型。 $p_2(v'_j | v_i)$ 采用 softmax 函数而非 sigmoid 归一化正是因为 softmax 函数可考虑节点 i 与所有邻居的关系而不再仅孤立地聚焦于一组节点对，从而正确反映 i 的邻域结构信息，便于计算二阶相似度。二阶相似度模型应使得概率 $p_2(v'_j | v_i)$ 与 $\hat{p}_2(j|i)$ 一致。同样基于 KL 散度得到二阶相似度 LINE 模型的目标函数为

$$(\min) \quad \mathcal{L}_2 = - \sum_{(i, j) \in E} w_{ij} \ln p_2(v'_j | v_i) \quad (\text{二阶相似度模型})$$

因为 softmax 计算非常低效，参考 Skip-Gram（详见第 24.5 节）引入负采样技术优化目标函数为

$$(\min) \quad \mathcal{L}_2 = - \sum_{(i, j) \in E} w_{ij} \left(\ln \sigma(v'_j^\top v_i) + \sum_{k=1}^K \mathbb{E}_{p_n} [\ln \sigma(-v'_k^\top v_i)] \right) \quad (\text{负采样二阶相似度模型})$$

上式中 p_n 表示采样负样本的噪声分布，令每个节点被采样的概率与节点的出度 d_i 正相关 $p_n(i) \propto d_i^{3/4}$ ； K 表示负样本数。注意到在引入负采样技术后上述目标函数与本节上文介绍的 GraRep 模型在数学形式上基本一致，故也可将其建模为矩阵分解问题：

4. 注意到一阶和二阶相似度的 LINE 模型的目标函数中均存在各边边权 w_{ij} 作为系数，导致在梯度更新时每一个参数因对应 w_{ij} 取值的差异而具有不同的更新幅度，从而难以确定适用于所有节点的学习率。因此模型进一步提出的边采样 (edge sampling) 优化技巧。以 w_{ij} 为依据采样连边，得到仅保留部分连边的子图，且将被采样连边的边权统一设为 1，将图简化为无权图。应用边采样技巧不仅避免了学习率取值的问题，还可减小问题规模，提升计算效率。为高效采样模型采用 Alias 算法（见第 23.5 节）；
5. 以上即为 LINE 模型的核心贡献。对有向图则只适用于二阶相似度模型，对无向图则可分别基于一阶和二阶相似度模型嵌入后拼接两模型的结果服务于后续任务，从而进一步提升图邻接特征捕捉效果。

25.7 基于随机游走 (RandomWalk) 的图嵌入模型

本章序言中已指出，图嵌入方法包括三个范式——基于矩阵分解、基于随机游走和基于神经网络。基于矩阵分解的图嵌入思想源于“深度学习中的嵌入任务广义上属于机器学习的降维任务”的事实，而基于随机游走

的图嵌入则是受到流行一时的词嵌入研究的启发。在 2013 年 word2vec 首次提出成熟高效的大规模词嵌入解决方案后（见第 24.5 节），包括推荐系统、图学习在内的多个其它领域开始借鉴这一思路。**随机游走 (random walk)** 提供了一种建立图嵌入与词嵌入关系的巧妙方法。通过在图上任意节点出发进行随机游走，可以采样得到由图中各节点组成的路径信息，且该路径蕴含了图中各节点的邻接关系。将节点视为单词、节点路径视为语句，则可迁移词嵌入领域的成熟方法研究节点和图的嵌入模型。使用随机游走还可带来两个好处：

- **并行化：** 可同时在网络中的多个节点开始随机游走生成大量序列，减少采样时间；
- **适应性：** 随机游走得到的序列仅涉及网络的局部结构，因此网络的局部变化也只会对有限的随机游走序列产生影响，而无需每次都重新计算整个网络的随机游走。这一特性使得基于随机游走的图模型虽然名义上是直推式 (**transductive**) 模型，但可较好的适应归纳式 (**inductive**) 的任务。

基于随机游走的图嵌入研究大体与基于矩阵分解的方法处于同一时期（2013-2017），但因为可迁移最新的词嵌入方法，前者较后者更为流行。在图神经网络成熟后（2016 年 GCN 模型提出）基于随机游走的图模型便被迅速取代。因为与直接学习网络结构相比，基于随机游走的研究一方面需预先生成大量节点序列，计算效率较低；另一方面是从图采样序列的过程不可避免地存在图拓扑结构的丢失，影响随机游走图模型的学习上限。

25.7.1 DeepWalk 模型

1. 受 2013 年 word2vec 方法的启发，2014 年 B. Perozzi 等人即提出面向图节点嵌入的 DeepWalk 模型¹⁹。模型非常简单，但因其是基于随机游走图嵌入研究的开山之作，巧妙地结合随机游走方法与自然语言处理方法为图嵌入研究引入全新的灵感，因此成为最经典的图嵌入模型之一，也是 GCN 提出前图机器学习模型的代表；
2. 模型首先基于随机游走在图上采样，将节点邻接结构映射成序列结构。随机游走时基于均匀采样 (**uniform sampling**) 从节点邻居选择下一个节点，直至采样序列长度达到给定最大长度。记图的节点数为 $|V|$ ，以每个节点为起点生成 k 条随机游走序列，每次随机游走的长度为 l ，则完成整张图的随机游走的时间复杂度为 $O(kl|V|)$ ；
3. DeepWalk 模型的下一个模块是直接套用 word2vec 中的 Skip-Gram 模型学习随机游走序列中每一节点的低维向量表示。如今复现 DeepWalk 模型时一般选择引入负采样改进的 Skip-Gram 模型（详见第 24.5 节）；
4. 以上即是 DeepWalk 的全部工作。作者通过多标签分类任务测试模型性能。与当时已有的方法相比，其优势在于适用于大规模网络，并且可在稀疏标记的环境中取得良好的性能。但 DeepWalk 更大的价值在于创造性地开辟了全新的图嵌入技术路线。

25.7.2 Node2Vec 模型

1. 2016 年提出的 Node2Vec 模型聚焦于对 DeepWalk 模型中随机游走的改进²⁰。随机游走图嵌入研究的核心在于通过随机游走生成节点序列以反映图拓扑信息，故随机游走序列的质量决定了图嵌入质量的上限；
2. DeepWalk 采用的完全随机的游走时空复杂度较小。但 Node2Vec 认为完全随机游走忽略了节点间的连接“强度”，也无法考虑节点的结构等价性 (**structural equivalence**) 和同质性 (**homophily**)：
 - 所谓结构等价的节点是指在网络中承担相似“功能”的节点，如不同簇的中心节点，一般相距较远；
 - 所谓同质节点是指网络中局部结构相似的节点，如同属一个簇的节点，一般相距较近。
 在此基础上 Node2Vec 模型认为若随机游走倾向于广度优先搜索 (**breadth-first-search, BFS**)，则生成的序列会更倾向于在起点的附近进行探索，从而捕捉节点间的同质性；若随机游走倾向于深度优先搜索 (**depth-first-search, DFS**)，则生成的序列会探索图中更大的区域，从而有机会捕捉节点间的结构相似性。而 DeepWalk 的随机游走过程完全随机，无法控制其 BFS 或 DFS 倾向性；
3. 为控制随机游走的 BFS 或 DFS 倾向性，作者提出了有偏二阶随机游走模型 (**biased second order**

¹⁹B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations", in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 701-710.

²⁰Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 855-864. <https://doi.org/10.1145/2939672.2939754>

random walk), 所谓“二阶”是指游走的下一步节点 x 不仅取决于当前的节点 v , 也取决于上一步节点 t 。定义游走转移概率为 π_{vx} , 有

$$\pi_{vx} = \frac{\alpha_{pq}(t, x) \cdot w_{vx}}{\sum_{x' \in N(v)} \alpha_{pq}(t, x') \cdot w_{vx'}}, \quad \alpha_{pq}(t, x) = \begin{cases} 1/p & d_{tx} = 0 \\ 1 & d_{tx} = 1 \\ 1/q & d_{tx} = 2 \end{cases}$$

上式中 $N(v)$ 表示节点 v 的一阶邻居集合; w_{vx} 表示连边 (v, x) 的权重; d_{tx} 表示节点 t 与 x 间的距离 (不考虑边权); p, q 为控制随机游走 BFS、DFS 倾向性的超参, 显然 p 越大越不容易返回上一步节点 t , 即鼓励模型游走的更远, 而 q 越大则倾向于在上一步节点 t 周围探索, 当 $p = q = 1$ 时模型退化为 DeepWalk;

4. DeepWalk 模型的完全随机行走中每一步行走都是等概率随机采样, 故每一步行走的时间复杂度为 $O(1)$, 总的随机行走时间复杂度为 $O(kl|V|)$ 。在考虑有偏二阶随机游走后, Node2Vec 模型随机行走时每一步为离散概率采样, 需采用 Alias 采样策略 (见第 23.5 节) 才能实现 $O(1)$ 的时间复杂度。考虑到 Alias 采样策略需要额外进行初始化, 则最终 **Node2Vec** 模型整体随机行走的时间复杂度为 $O(kl|V| + 2|E|)$, 其中 $|E|$ 为图的连边数;
5. 以上即为 Node2Vec 模型的核心创新点, 在节点嵌入部分与 DeepWalk 一致采用 Skip-Gram 模型。

25.7.3 Struc2Vec 模型

1. Struc2Vec 模型提出于 2017 年, 是一类晚期且非典型的随机行走图嵌入模型。与经典的 DeepWalk、Node2Vec 相比, **Struc2Vec** 模型针对随机游走过程作了颠覆性的修改——不直接在待嵌入的图结构上作随机游走, 而是构建一个全新的图在其上作随机游走, 所构建的新图要求反映原图内部的结构相似性^{21,22};
2. Struc2Vec 认为, 尽管 Node2Vec 模型通过有偏随机行走兼顾了远距离节点间的结构相似性, 但行走时并未刻意寻找结构相似的节点, 而且因为每一步随机行走仅在节点的一阶邻域内作选择, 在随机行走总长度的约束下无法保证结构相似的两个节点能被捕捉在同一个窗口中, 从而导致无法习得相近的向量表达。由名称可知, **Struc2Vec** 是一种完全聚焦于节点结构相似度学习的图嵌入模型。其思路是在原图的基础上构建新的拓扑链接, 使得随机行走时有机会跳到远距离的结构相似的节点中, 充分探索图中各区域的结构特征, 保证结构相似的节点具有相近的向量表示;
3. 模型以度分布作为图节点的结构特征。具体地模型认为: 如果两个节点具有相同的度, 那么它们可能是结构相似的; 若它们的邻居也有着相同的度, 那么它们在结构上则更加相似。因此比较两节点各阶邻居的度分布即可全面地评估两节点的结构相似度。模型量化节点 k 阶邻域结构相似性 $w_k(u, v)$ 如下

$$w_k(u, v) = \exp\{-f_k(u, v)\}, \quad f_k(u, v) = f_{k-1}(u, v) + g(S(R_k(u)), S(R_k(v))), \quad k \geq 0, |R_k(u)|, |R_k(v)| > 0, f_{-1}(u, v) = 0$$

上式中 $f_k(u, v)$ 为量化节点 u, v 的第 k 阶邻域结构特征的距离函数; $R_k(u)$ 表示节点 u 的第 k 阶邻居的集合; $S(R_k(u))$ 为节点 u 的第 k 阶邻居节点的度组成的有序序列 (从小到大排序); $g(\cdot, \cdot)$ 表示序列间距离的度量函数, 模型采用动态时间规整 (dynamic time warping, DTW) 度量 (见第 23.9.5 节)。应用 DTW 算法时需预设点间距离计算公式 $d(x, y)$, 模型定义为 $d(x, y) = \frac{\max\{x, y\}}{\min\{x, y\}} - 1$;

4. 类似于 GraRep 算法 (见第 25.6 节) 为考虑节点的结构特征对图的各阶邻接关系分别构建向量表示, Struc2Vec 模型也要求分别计算任意 u, v 间的各阶邻域结构相似性 $\{w_0(u, v), w_1(u, v), \dots, w_{K-1}(u, v)\}$ 以全面挖掘图的结构特征。以 $\{w_0(u, v) | \forall u \neq v\}$ 为边权构建第零层无向完全图以反映各节点的零阶结构相关性; 再以 $\{w_1(u, v) | \forall u \neq v\}$ 为边权构建第一层无向完全图以反映各节点的一阶结构相关性; 以此类推共构建 K 层无向完全图;
5. 将构建的 K 层无向完全图连接起来得到多层带权图 (**multilayer weighted graph**) 即可全面反映原图的结构相似性。仅对不同层之间的同一个节点建立有向连接, 连边权重经验地设计为

$$w(u_k, u_{k-1}) = 1, \quad w(u_k, u_{k+1}) = \ln(\Gamma_k(u) + e), \quad \Gamma_k(u) = \sum_{v \in V} \mathbb{I}(w_k(u, v) > \bar{w}_k)$$

²¹图神经网络学习笔记之三 (struc2vec): <https://zhuanlan.zhihu.com/p/417132945>

²²社交网络分析 (六)-Struc2Vec: <https://zhuanlan.zhihu.com/p/88317960>

式中 e 表示自然系数，因此显然有 $w(u_k, u_{k+1}) > w(u_k, u_{k-1}) = 1$ 。其背后的思想是随着 k 的增加 $w_k(u, v)$ 对节点 u, v 间结构相似度的区分能力越强，因此设计 $w(u_k, u_{k+1}) > w(u_k, u_{k-1})$ 引导随机游走时更倾向于向更高层跳转可保证最终模型对图的高阶结构特征的学习能力。 $\Gamma_k(u)$ 的设计同理。其定义为节点 u 在第 k 层图的连边中边权大于该层平均边权的数目。因为边权 $w_k(u, v)$ 越大表示节点 u, v 的结构相似度越高，如果节点 u 在第 k 层图中存在大量连边的边权高于该层平均边权，则表明该层不足以充分区分节点 u 与其它节点的结构相似度，故赋予 $w(u_k, u_{k+1})$ 较大的取值以引导随机游走至 u 处向更高层跳转；

6. 最后就是在由 K 层完全图组成的大图中作随机游走采样。每一步游走时预设一个概率 q 以确定是否留在当前层，再基于层内跳转概率 $p_k(u, v)$ 和层间跳转概率 $p_k(u_k, u_{k+1}), p_k(u_k, u_{k-1})$ 确定下一步节点

$$p_k(u, v) = \frac{w_k(u, v)}{\sum_{v' \neq u} w_k(u, v')}, \quad p_k(u_k, u_{k+1}) = \frac{w(u_k, u_{k+1})}{w(u_k, u_{k+1}) + w(u_k, u_{k-1})}, \quad p_k(u_k, u_{k-1}) = 1 - p_k(u_k, u_{k+1})$$

7. 以上即为 Struc2Vec 模型的核心贡献。从其对经典模型中随机游走环节的大刀阔斧修改中不难看出研究的野心。但一方面其多层图构建的计算复杂度过大，尽管作者在论文后半部分作诸多优化并声称可显著减少计算量，但仍难以应用于大型图；另一方面论文提出时 GCN 模型已取得巨大成功，图机器学习方法论转移至各类图卷积模型，基于随机游走的技术路线迅速失势，导致相关思路的继承和优化基本中断。

25.8 深度图生成模型 (*Deep graph generative model*)

在 2016 年 GCN 提出之后，图神经网络表现出了远超其它技术路线的图拓扑特征挖掘能力，自然而然地成为图嵌入领域的主流技术路线。传统深度学习领域相关模型的发展史又进一步指出——除了服务于传统的降维和预测任务外，嵌入信息的另一个潜力巨大的应用方向在于数据生成。因此将图模型推广至图生成任务即成为一个自然而然的方向。然而与传统的音频、文本、图像、视频等数据的生成相比，图结构的生成具有更大的挑战。理想的图生成模型应该具备生成包含大规模且不固定数目节点与连边的任意连通图结构的能力。然而作为一种离散结构，图的迭代构建过程是不可微的，这为基于反向传播的图深度学习模型提出了巨大的挑战。目前绝大部分的图生成模型仅针对有限的场景设计，根据采用的深度生成模型技术，可大致分为基于 VAE 的模型、基于 GAN 的模型和深度自回归模型²³。

25.8.1 图自编码器 (**graph autoencoder, GAE**) 与变分图自编码器 (**variational graph autoencoder, VGAE**)

1. 与 GCN 提出同年，GCN 的提出者 Thomas N. Kipf 便尝试将 GCN 应用于变分自编码器框架（见第 24.9 节），首次提出变分图自编码器 (**variational graph autoencoder, VGAE**) 模型，是图自编码器模型的开山之作²⁴。受图模型发展背景的影响，VGAE 模型在编码阶段仅接受无向图（基本形式的 GCN 模型仅适用于无向图），而在解码阶段模型仅能生成无向无权图，且因当时的图模型尚不适用于归纳式任务，故 VGAE 模型只能学习一个图，更多是在节点集不变的前提下用于连边预测任务。另外模型训练后的编码器也可用于嵌入任务；
2. **GAE、VGAE** 与经典自编码器结构的一大差异在于 **GAE、VGAE** 的编码器和解码器并不具备对称结构。首先介绍简单的 GAE 模型。模型的编码器为一个双层 GCN 模型，而解码器以连边预测（重构 0-1 邻接矩阵）为目标，但不具被神经网络结构

$$\hat{A} = \sigma(ZZ^\top), \quad Z = GCN(A, X), \quad \hat{A}, A \in \mathbb{R}^{|V| \times |V|}, \quad X \in \mathbb{R}^{|V| \times k}, \quad Z \in \mathbb{R}^{|V| \times d} \quad (\text{GAE})$$

式中 Z 为编码器基于图邻接矩阵 A 和各节点特征矩阵 X 得到的各节点编码； \hat{A} 为解码器的重构的邻接矩阵，基于 sigmoid 函数直接预测各节点的邻接概率。当输入数据 A 为无权图时训练目标即简单地最小化 \hat{A} 的重构误差；当 A 为带权图时则可尝试将其权重压缩为 0-1 之间或者基于预设阈值将带权图“砍”为无权图；

²³论文导读 | 深度图生成模型简介：<https://zhuanlan.zhihu.com/p/267284870>

²⁴Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*. <https://arxiv.org/abs/1611.07308>

3. 进一步介绍 VGAE 模型的细节，其思路同样与 VAE 完全一致。为将图节点嵌入至高斯分布空间，模型的编码器包括两个双层 GCN 模型（第一层权重共享）分别编码高斯分布的均值 μ 和标准差 σ 。解码器与 GAE 模型的解码器一致，但输入为从嵌入高斯分布空间的抽样数据。最终的训练目标为最小化重构误差（交叉熵）和嵌入高斯分布空间的先验误差（KL 散度）

$$(\min) \quad -\mathbb{E}_{q(Z|X,A)} [\ln p(A|Z)] + KL [q(Z|X,A) \| p(Z)], \quad (\text{VGAE})$$

$$\begin{aligned} \text{Encoder: } q(Z|X,A) &= \prod_{i=1}^{|V|} q(z_i|X,A), \quad q(z_i|X,A) = N(z_i|\mu_i, \text{diag}(\sigma_i^2)), \quad \mu = GCN_\mu(A,X), \quad \ln \sigma = GCN_\sigma(A,X) \\ \text{Decoder: } p(A|Z) &= \prod_{i=1}^{|V|} \prod_{j=1}^{|V|} p(A_{ij}|z_i, z_j), \quad p(A_{ij} = 1|z_i, z_j) = \sigma(z_i^\top, z_j) \end{aligned}$$

式中 $p(Z) = \prod_{i=1}^{|V|} N(z_i|0, I)$ 为多个独立标准高斯分布先验的联合先验。

25.8.2 GraphVAE 模型 (ICANN 2018)

1. 作为基于变分自编码器框架图生成的开山之作，VGAE 只具备非常有限的图生成能力——只能学习一种图结构，并在顶点集给定的前提下生成无向图，且无法生成丰富的连边特征等等。后续大量学者延续变分自编码器的框架展开进一步研究，2018 年提出的 GraphVAE 模型²⁵便是其中的代表性成果；
2. 与 VGAE 模型相比，GraphVAE 模型更为复杂，但也获得了更强的图生成能力——可学习多种图结构并生成节点数可变、邻接关系任意、且具有丰富连边特征的无向或有向图结构。为使节点数可变，模型实际上是预设最大的节点数，从而将任意图生成的问题简化为从给定节点集中选择节点并确定连接关系的问题（类似于文本生成，字典表即为单词选择的上限）。而且因为计算复杂度较高，模型无法预设过大的节点集，故模型只适用于小规模图结构的生成（论文指出最多不超过 38 个节点），但已可满足化合物结构预测等不少领域的研究需求；
3. 记图 $G = (A, E, F, y)$ 。其中 $A \in \{0, 1\}^{n \times n}$ 为邻接矩阵（对角线元素设为 1，相当于存在自环）； $E \in \mathbb{R}^{n \times n \times d_e}$ 表示连边特征， d_e 表示连边特征维数； $F \in \mathbb{R}^{n \times d_n}$ 表示节点特征， d_n 表示节点特征维数； $y \in \mathbb{R}^{d_y}$ 表示图的类别标签向量。GraphVAE 模型的结构是由编码器基于 A, E, F, y 将图 G 嵌入到连续概率空间 $z \in \mathbb{R}^c$ ，再由解码器基于输入的图类别标签 y 和概率空间抽样 z 构造 $\tilde{A} \in [0, 1]^{k \times k}$, $\tilde{E} \in \mathbb{R}^{k \times k \times d_e}$, $\tilde{F} \in \mathbb{R}^{k \times d_n}$ ，并要求 $\tilde{G} = (\tilde{A}, \tilde{E}, \tilde{F}, y)$ 应尽可能接近 $G = (A, E, F, y)$ 。 $k \geq n$ 表示预设的最大节点数， \tilde{A} 表示概率连接矩阵，其中对角线元素 $\tilde{A}_{a,a}$ 对应候选节点 a 被选择的概率，非对角线元素 $\tilde{A}_{a,b}$ 指连边 (a, b) 的生成概率；
4. 再介绍图类别标签 y 的在编码与解码时的处理方法。在编码前， y 将并入节点特征 F 构造 $\mathbb{R}^{n \times (d_n + d_y)}$ 矩阵输入编码器。而在解码前 y 将并入概率空间向量 z 构造 $\mathbb{R}^{(c+d_y)}$ 向量输入解码器。引入图类别标签 y 可使得 GraphVAE 模型更好地区分不同类型图的结构，从而提升生成结果的针对性；
5. 为同时编码图的拓扑结构 A 、节点属性 F 和连边属性 E ，作者采用 ECC 图卷积模型（第 25.4.3 节）作为 GraphVAE 的编码器，并构建多层全连接网络作为解码器。遵循变分自编码器的统一架构，GraphVAE 模型训练的目标函数如下

$$(\min) \quad \mathcal{L}(\phi, \theta, G) = -\mathbb{E}_{q_\phi(z|G,y)} [\ln p_\theta(G|z,y)] + KL [q_\phi(z|G,y) \| p(z)]$$

上式中的第一项指解码结果 \tilde{G} 与原始输入 G 的重构误差 (reconstruction loss)， ϕ, θ 分别指编码器和解码器参数；目标函数第二项为概率嵌入 z 所属高斯分布与标准高斯先验 $p(z)$ 的距离，为变分自编码器的核心。模型训练的难点在于如何在尺寸不一致 ($k \neq n$) 的情况下量化 \tilde{G}, G 之间的重构误差；

6. 为计算 \tilde{G}, G 之间的重构误差，需首先确定从 \tilde{G} 到 G 的映射。定义二项分配矩阵 (binary assignment matrix) $X \in \{0, 1\}^{k \times n}$ 表示从 \tilde{G} 到 G 的映射，当且仅当节点 $a \in \tilde{G}$ 被映射至节点 $i \in G$ 时有 $X_{a,i} = 1$ 。基于

²⁵Simonovsky, M., & Komodakis, N. (2018). GraphVAE: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning-ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I* 27 (pp. 412-422). Springer International Publishing. <https://arxiv.org/abs/1802.03480>

X 便可实现 \tilde{G}, G 的相互映射

$$A' = XAX^\top \in \{0, 1\}^{k \times k}, \quad \tilde{F}' = X^\top \tilde{F} \in \mathbb{R}^{n \times d_n}, \quad \tilde{E}'_{\cdot, \cdot, l} = X^\top \tilde{E}_{\cdot, \cdot, l} X \in \mathbb{R}^{n \times n}$$

其中 A' 是原图的邻接关系在预测图 \tilde{G} 空间的映射; \tilde{F}', \tilde{E}' 则是预测的节点和连边特征在原图 G 空间的映射; $\tilde{E}_{\cdot, \cdot, l}$ 表示 $\tilde{E} \in \mathbb{R}^{k \times k \times d_e}$ 于第 $l \leq d_e$ 维特征的矩阵切片。对比 $(A', \tilde{A}), (E, \tilde{E}'), (F, \tilde{F}')$ 即可量化 $\ln p_\theta(G|z, y)$

$$\ln p_\theta(G|z, y) = \lambda_A \ln p_\theta(A'|z, y) + \lambda_F \ln p_\theta(F|z, y) + \lambda_E \ln p_\theta(E|z, y),$$

$$\ln p_\theta(A'|z, y) = \frac{1}{k} \sum_a \left(A'_{a,a} \ln \tilde{A}_{a,a} + (1 - A'_{a,a}) \ln (1 - \tilde{A}_{a,a}) \right) + \frac{1}{k(k-1)} \sum_{a \neq b} \left(A'_{a,b} \ln \tilde{A}_{a,b} + (1 - A'_{a,b}) \ln (1 - \tilde{A}_{a,b}) \right),$$

$$\ln p_\theta(F|z, y) = \frac{1}{n} \sum_i \ln \left(F_{i,\cdot}^\top \tilde{F}'_{i,\cdot} \right), \quad \ln p_\theta(E|z, y) = \frac{1}{\|A\|_1 - n} \sum_{i \neq j} \ln \left(E_{i,j}^\top \tilde{E}'_{i,j} \right)$$

式中 $\lambda_A, \lambda_F, \lambda_E$ 为模型预设超参。基于 $\ln p_\theta(G|z, y)$ 即可由蒙特卡洛法 (见第 27.1 节) 近似估计重构误差 $-\mathbb{E}_{q_\theta(z|G, y)} [\ln p_\theta(G|z, y)]$ 。可以看到计算重构误差时 A', \tilde{A} 与 $E, \tilde{E}', F, \tilde{F}'$ 处于不同的空间。在比较图拓扑结构 A', \tilde{A} 的重构误差时是在编码结果 \tilde{G} 的空间内比较, 由此同时考虑了原本图 G 空间中存在和不存在的节点和连边集, 为 \tilde{G} 结构的预测提供了更精细的指导, 故而在进一步细化比较节点和连边属性时便仅在原图 G 的空间内作比较。至此模型的难点最终转移至二项分配矩阵 X 的估计;

7. 一般将图的最优匹配问题建模为一个整数二项规划 (**integer quadratic programming, IQP**) 问题。为得到最优分配矩阵 X , 需首先分别建模图 \tilde{G} 中任意连边 (a, b) 、节点 a 与图 G 中任意连边 (i, j) 、节点 i 的相似度。定义相似度函数

$$S((i, j), (a, b)) = \begin{cases} \left(E_{i,j}^\top \tilde{E}_{a,b} \right) A_{i,j} \tilde{A}_{a,b} \tilde{A}_{a,a} \tilde{A}_{b,b} & i \neq j \wedge a \neq b \\ \left(F_{i,\cdot}^\top \tilde{F}_{a,\cdot} \right) \tilde{A}_{a,a} & i = j \wedge a = b \end{cases}$$

当 $i \neq j$ 且 $a \neq b$ 时上式计算连边相似度, 当 $i = j$ 且 $a = b$ 时上式计算节点相似度。显然最优匹配方案应使得匹配后连边、节点相似度最高, 从而得到匹配问题的目标函数为

$$f(x) = \sum_{X_{a,i}=1, X_{b,j}=1} S((i, j), (a, b)) = x^\top S x, \quad x = \text{vec}(X) \in \{0, 1\}^{kn}, \quad S \in \mathbb{R}^{kn \times kn}$$

上式中 x 为 X 的按列向量化结果 (**column-wise vectorized replica**); S 为相似度矩阵, 其对角线元素为节点相似度, 非对角线元素为连边相似度。再考虑 x 的其它约束建模从 \tilde{G} 到 G 的最优匹配问题如下

$$x^* = \arg \max_x x^\top S x, \quad \text{s.t. } x = \text{vec}(X) \in \{0, 1\}^{kn}, \quad \sum_{i=1}^n X_{a,i} \leq 1, \quad \sum_{a=1}^k X_{a,i} \leq 1 \quad (\text{标准图匹配问题})$$

式中 $\sum_{i=1}^n X_{a,i} \leq 1, \sum_{a=1}^k X_{a,i} \leq 1$ 常被称为匹配约束 (**matching constraints**) 或映射约束 (**mapping constraints**)。因为整数二项规划问题是一个 NP 难问题, 上式难以直接求解。大量研究均选择松弛上述问题的部分约束。GraphVAE 模型参考 Cho 等的工作²⁶采用最大池化匹配 (**max-pooling match, MPM**) 算法近似估计 x^* 。具体地, 模型完全松弛了上述问题的整数约束和匹配约束, 再额外增加了对 x 的 L2 范数约束构造松弛图匹配问题

$$x^* = \arg \max_x x^\top S x, \quad \text{s.t. } x = \text{vec}(X) \in [0, 1]^{kn}, \quad \|x\|_2 = 1 \quad (\text{松弛图匹配问题})$$

注意到若忽略 $x \in [0, 1]^{kn}$ 的约束, 则实际上即是求解瑞利商 (**Rayleigh quotient**) 问题, 最优解为矩阵 S 的最大特征值对应的特征向量 (证明见第 23.13 节)。又因为 S 为非负矩阵, 则基于 Perron-Frobenius 定理其最大特征值对应的特征向量必然非负, 故而自然满足 $x \in [0, 1]^{kn}$ 的约束。综上, 直接求解矩阵 S

²⁶M. Cho, J. Sun, O. Duchenne and J. Ponce, "Finding Matches in a Haystack: A Max-Pooling Strategy for Graph Matching in the Presence of Outliers," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 2091-2098, doi: 10.1109/CVPR.2014.268. https://openaccess.thecvf.com/content_cvpr_2014/papers/Cho_Finding_Matches_in_2014_CVPR_paper.pdf

最大特征值对应的特征向量作为上述松弛图匹配问题的最优解 x^* , 以常用的幂法 (power method) 为例迭代式如下

$$x \leftarrow \frac{Sx}{\|Sx\|_2} \iff X_{a,i} \leftarrow \frac{1}{\|Sx\|_2} \left(X_{a,i}S((i,i),(a,a)) + \sum_{j \in N(i)} \sum_{b \in N(a)} X_{b,j}S((i,j),(a,b)) \right)$$

上式中 $N(i), N(a)$ 分别指图 G, \tilde{G} 中节点 i, a 的邻居集合。因为 GraphVAE 模型生成的图 \tilde{G} 的拓扑结构 \tilde{A} 是以概率形式建模的, 故计算 $\sum_{b \in N(a)} X_{b,j}S((i,j),(a,b))$ 时实际上是考虑了图 \tilde{G} 中所有可能与 a 相连的节点, 这种计算方式可认为是求和池化 (sum-pooling) 或均值池化 (average-pooling)。在其它匹配任务中发现其缺点在于可能过多地考虑了实际上与 a 无关节点的影响, 降低解的质量。只考虑最可能与 a 相连的节点即可避免这一问题, 从而得到基于最大池化的匹配方法

$$X_{a,i} \leftarrow \frac{1}{\|Sx\|_2} \left(X_{a,i}S((i,i),(a,a)) + \sum_{j \in N(i)} \max_{b \in N(a)} \{X_{b,j}S((i,j),(a,b))\} \right)$$

按上式迭代直至收敛即得到松弛图匹配问题的最优解 $X^* \in [0, 1]^{k \times n}$ 。最后还需将其离散化得到满足匹配约束 $\sum_{i=1}^n X_{a,i} \leq 1$, $\sum_{a=1}^k X_{a,i} \leq 1$ 的最终映射矩阵 $X^* \in \{0, 1\}^{k \times n}$ 。论文作者发现, 基于匈牙利算法 (Hungarian algorithm) 可取得良好的离散化效果。

25.9 知识图谱 (***Knowledge graph***) 与知识表示学习 (***Knowledge representation learning***) 基础

1. 知识表示学习 (***knowledge representation learning***) 是图表示学习中一个特殊的领域。其研究对象为知识图谱 (***knowledge graph***), 研究目标是系统学习客观知识^{27,28}。与一般的图表示学习任务相比, 知识表示学习的特殊性源于知识图谱的特殊性;
2. 知识图谱的起源可追溯至上世纪 50 年代人工智能发展的早期阶段。符号人工智能 (***symbolic artificial intelligence***) 是当时主流的人工智能研究范式 (50 年代中期至 80 年代末期)。区别于后期兴起的数据和算力驱动的机器学习范式, 符号人工智能的目标是研究人类知识在计算机系统中的符号化、结构化表示, 以此模拟人的思考、推理过程。Richens 在这一时期 (1956 年) 提出了语义网络 (***semantic net***) 模型作为一种知识表示的方法, 主要用于自然语言理解领域。这也是用图结构描述人类知识的最早源头。尽管在医药、游戏等领域取得初步成果, 但受限于数据、算力和模型拟合能力, 符号人工智能追求的“学习人类知识”目标在当时仍是遥不可及, 故而逐渐被更实用的机器学习流派取代。随着 21 世纪以来数据、算力及方法论等各方面领域的突破, 2012 年 Google 正式提出“知识图谱”的概念, 并发布了包含 570 亿实体的大规模知识图谱和基于知识图谱的搜索引擎, 使得这一人工智能早期阶段的理想化愿景重新回到聚光灯下。知识图谱逐渐成为一个独立的研究领域, 得到学术界和工业界的极大重视;
3. 知识图谱的本质是一个有向图, 图中的节点表示实体 (***entity***)、连边表示实体间的关系 (***relation***)。但因其“承载知识”的特殊目的, 知识图谱较一般的图结构 (交通网络图、社交关系图、科学引文图、蛋白质关系图等等) 具有明显的特殊性, 具体表现为强烈的异质性——图中的实体与关系不仅可能具有不同的属性, 往往还存在不同的类别。知识图谱在数学上一般表示为 $\mathcal{G}(\mathcal{E}, \mathcal{R}, \mathcal{F})$ 。其中 \mathcal{E}, \mathcal{R} 分别表示实体和关系的集合, 而 \mathcal{F} 则是事实 (***fact***) 的集合, 一个事实被表示为一组由头实体、关系、尾实体组成的知识三元组 $(h, r, t) \in \mathcal{F}$ 。知识三元组是知识图谱的基本组成单元, 每一组知识三元组对应一种结构化的知识;
4. 知识图谱目前的研究方向可以大致分为四类: 知识表示学习 (***knowledge representation learning***)、知识获取 (***knowledge acquisition***)、时序知识图谱 (***temporal knowledge graph***) 和知识应用 (***knowledge-aware applications***)。其中知识表示学习承载着学习知识图谱表征的结构化知识的任务, 是知识图谱相关研究的基础和热点;

²⁷ 知识图谱入门——认识知识图谱: <https://zhuanlan.zhihu.com/p/396516565>

²⁸ 自动化知识图谱表示学习: 从三元组到子图。 <https://zhuanlan.zhihu.com/p/511199700>

5. 与一般的图表示学习一致，知识表示学习的目标同样是寻找知识图谱在欧式低维稠密向量空间中的向量表达。一般的图结构的重点在于节点，连边仅作为表示数据在非欧空间分布模式的工具，因此一般图表表示学习的关键在于节点的表示学习；而知识图谱的关键在于描述结构化知识的三元组，三元组中既包含节点也包含连边，因此为正确习得“知识”，知识表示模型往往要求同时学习节点和连边的向量表示；
6. 正是因为知识图谱和知识三元组的特殊意义，使得知识表示学习成为如今图表示学习中少有的方法论不被图神经网络垄断的领域。具体地，按学习目标如今的知识表示学习范式大体可分为三类：
- **基于三元组的模型：**聚焦于对单体知识的理解，关键在于建模具有对称性或相反性的语义，具有计算效率高的优势。在实际落地中应用较为广泛，也仍是研究的热点。缺点是难以建模高阶、复合的知识信息。按方法论又可细分为翻译距离模型 (*translational distance model*, TDM)、双线性模型 (*bilinear model*, BLM)、基于矩阵分解的模型和基于神经网络的模型，以前两者最为常用。此类研究可类比基于矩阵分解的图表示研究，均是以复现节点间邻接关系为主要目标；
 - **基于关系路径的模型：**单个知识三元组仅能表达有限的、基础的知识。将三元组中的头实体和尾实体通过图中的路径相连，则在保持三元组的同时还可得到更加高阶、复合的知识。因此基于关系路径的模型侧重于建模复合语义，具有更强的解释性。其计算复杂度依赖于路径的数量。显然此类研究可类比基于随机游走的图表示研究；
 - **基于子图的模型：**直接基于图神经网络模型学习知识图谱的向量表示。模型侧重于建模高阶语义，使用复杂逻辑规则来达到可解释性的目标，这一类模型的计算复杂度由抽取的子图决定。可类比基于图神经网络的图表示研究。该领域是一个很有潜力的研究方向，目前主要面临着计算效率的挑战。

25.10 基于翻译距离 (*Translational distance*) 的知识三元组表示学习 (Trans 系列模型)

2013	TransE	Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. <i>Advances in neural information processing systems</i> , 26. https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf
2014	TransH	Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014, June). Knowledge graph embedding by translating on hyperplanes. In <i>Proceedings of the AAAI conference on artificial intelligence</i> (Vol. 28, No. 1). https://ojs.aaai.org/index.php/AAAI/article/view/8870
2015	TransR & CTransR	Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015, February). Learning entity and relation embeddings for knowledge graph completion. In <i>Proceedings of the AAAI conference on artificial intelligence</i> (Vol. 29, No. 1). https://ojs.aaai.org/index.php/AAAI/article/view/9491
	TransD	Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015, July). Knowledge graph embedding via dynamic mapping matrix. In <i>Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing</i> (volume 1: Long papers) (pp. 687-696). https://aclanthology.org/P15-1067.pdf
	TransA	Xiao, H., Huang, M., Hao, Y., & Zhu, X. (2015). TransA: An adaptive approach for knowledge graph embedding. <i>arXiv preprint arXiv:1509.05490</i> . https://arxiv.org/abs/1509.05490
	TransG	Xiao, H., Huang, M., Hao, Y., & Zhu, X. (2015). TransG: A generative mixture model for knowledge graph embedding. <i>arXiv preprint arXiv:1509.05488</i> . https://arxiv.org/abs/1509.05488
2016	KG2E	He, S., Liu, K., Ji, G., & Zhao, J. (2015, October). Learning to represent knowledge graphs with gaussian embedding. In <i>Proceedings of the 24th ACM international on conference on information and knowledge management</i> (pp. 623-632). https://dl.acm.org/doi/abs/10.1145/2806416.2806502
	TranSparse	Ji, G., Liu, K., He, S., & Zhao, J. (2016, February). Knowledge graph completion with adaptive sparse transfer matrix. In <i>Proceedings of the AAAI Conference on Artificial Intelligence</i> (Vol. 30, No. 1). https://ojs.aaai.org/index.php/AAAI/article/view/10089
	TransA+	Jia, Y., Wang, Y., Lin, H., Jin, X., & Cheng, X. (2016, February). Locally adaptive translation for knowledge graph embedding. In <i>Proceedings of the AAAI Conference on Artificial Intelligence</i> (Vol. 30, No. 1). https://ojs.aaai.org/index.php/AAAI/article/view/10091

⋮	⋮	⋮
---	---	---

- 翻译距离模型 (**translational distance model**, TDM) 又称为 Trans 系列模型, 是知识表示学习中最为经典的模型²⁹。该系列模型以发表于 NIPS2013 的 TransE 模型为基础, 在不到五年内快速发展出了包含 TransE、TransH、TransR、CTransR、TransD、TransA、TransG、KG2E、TranSparse、TransA+ 等等一系列三元组表示模型。其中贡献最大的 TransE 变体为 TransH 和 TransR 模型 (至 2024 年 3 月 2 日两篇对应论文的应用量约为 TransE 的一半, 其它变体均不足 TransE 的十分之一);;
- 所谓“翻译距离”源于 TransE 模型的论文名: *Translating Embeddings for Modeling Multi-relational Data*, 其基本思想是把知识三元组中的关系 (relation) 视为从头实体 (head) 到尾实体 (tail) 的翻译。记 h, r, t 分别为一个知识三元组中头实体、关系和尾实体的向量表征, 则 TransE 模型认为正确的表征结果应该满足 $h + r = t$, 而对于不满足三元组关系的 h', r, t' 则 $h' + r$ 与 t' 的差距应尽可能大, 这即是模型追求的“翻译”过程。基于此可自然地给出 TransE 模型的目标函数

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{F}} \left(\sum_{(h',r,t') \in \mathcal{F}'_{(h,r,t)}} [\gamma + d(h+r, t) - d(h'+r, t')]_+ \right), \quad \mathcal{F}'_{(h,r,t)} = \{(h', r, t) | h' \neq h \in \mathcal{E}\} \cup \{(h, r, t') | t' \neq t \in \mathcal{E}\}$$

式中 $d(\cdot)$ 为自定义的距离度量函数; $[\cdot]_+$ 等价于 ReLU 函数; $\gamma > 0$ 为间隔距离超参数, 要求 $d(h+r, t)$ 应至少比 $d(h'+r, t')$ 小 γ ; $\mathcal{F}'_{(h,r,t)}$ 表示针对 (h, r, t) 的负样本集合, 由对真实三元组的头实体或尾实体替换后的样本组成。因此 TransE 的计算过程也非常简单, 只需预先随机初始化所有实体和关系的表征向量, 再优化更新 h, r, t, h', t' 即可。记实体与关系的嵌入维数为 k , 则初始化向量认为服从 $(-6/\sqrt{k}, 6/\sqrt{k})$ 间的均匀分布, 均匀采样后需对所有实体和关系向量作归一化 $e \leftarrow e/\|e\|$, $r \leftarrow r/\|r\|$, 且在每一步更新后还需对实体向量作归一化。为减少计算量优化时一般采用随机梯度下降, 对每一组三元组 (h, r, t) 仅随机构造一组负样本 (h', r, t') 。作为知识表示学习的划时代模型, TransE 模型因其简单、直观、高效的特点取得了巨大的成功。但在实践中发现 TransE 模型对简单关系的建模效果显著, 但是对复杂关系的建模效果却十分不理想。所谓复杂关系一般包括两类:

- **自反关系:** 如同时存在 (h, r, t) 和 (t, r, h) 两种知识, 则 TransE 模型会倾向于令 $h = t$, $r = 0$, 意味着关系 r 无效, 显然不一定符合实际;
- **多关联关系:** 如在实体 h, t 不变的情况下同时存在 $(h, r_1, t), \dots, (h, r_n, t)$ 共 n 种知识, 则 TransE 模型会倾向于令 $r_1 = \dots = r_n$, 意味着 h, t 间只能有一种关系, 显然也不一定符合实际。

上述问题的直接原因是 TransE 在嵌入三元组时要求向量 h, r, t 三者共面。TransE 的一系列变体均是为解决这一问题而提出的。TransE 模型的另一项缺点是易生成假负样本。随机替换掉 (h, r, t) 中的头实体或尾实体生成的三元组 $(h', r, t), (h, r, t')$ 可能是客观成立的。又因为知识图谱往往不完整, 无法捕捉真实世界的所有知识, 从而导致很多假负样本无法被识别。

- 2014 年提出的 TransH 模型是一种代表性的、且被广泛认可的 TransE 变体。模型巧妙地补充了 TransE 对“翻译”的定义从而提升了对复杂关系的表示效果。TransH 认为从 h 到 t 的“翻译”不应仅与实体属性 h, t 有关, 还应考虑关系 r 的类别。“TransH”中的“H”指“超平面 (hyperplane)”。模型为每一种关系 r 对应一个专门的超平面, 从 h 到 t 的“翻译”不再在整个超空间中, 而是在特定的超平面上, 故而模型的目标从 $h + r = t$ 变为 $h_r + r = t_r$, 其中 h_r, t_r 分别表示 h, t 在 r 所处的超平面上的投影。记 r 所处的超平面的单位法向量为 w_r , 易知任意实体向量 e 的投影关系为 $e_r = e - w_r^\top e w_r$, 则可写出 TransH 模型的目标函数, 并基于随机梯度更新所有实体、关系、和关系超平面法向量的嵌入 e, r, w_r

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{F}} \left(\sum_{(h',r,t') \in \mathcal{F}'_{(h,r,t)}} [\gamma + d(h_r + r, t_r) - d(h'_r + r, t'_r)]_+ \right) + C \left\{ \sum_{e \in \mathcal{E}} [\|e\|_2^2 - 1]_+ + \sum_{r \in \mathcal{R}} \left[\frac{(w_r^\top r)^2}{\|r\|_2^2} - \epsilon^2 \right]_+ \right\}$$

²⁹知识表示学习 Trans 系列梳理 (论文 + 代码): <https://zhuanlan.zhihu.com/p/144412694>

上述目标函数由三部分组成。第一项旨在实现 $h_r + r = t_r$, 后两项为通过罚函数法引入的软约束, 包括实体向量 e 的归一化约束和 r, w_r 间的正交化约束。另外 TransH 还设定 w_r 为单位法向量, 因此在每一步更新后还需对 w_r 作归一化。在实际应用中发现正交化约束对知识嵌入效果无显著影响, 故常去掉以提升计算效率。**TransH** 模型相较 **TransE** 的另一项改进在于负样本集 $\mathcal{F}'_{(h,r,t)}$ 的生成方式。为减少假负样本生成的概率, 模型不再完全随机地替换头实体或尾实体, 而是对于一对多的关系更多地替换其头实体, 多对一的关系则更多地替换尾实体。对于关系 r , 统计两个参数 tph, hpt , 分别表示该关系中每个头实体对应的尾实体数 (tails per head) 和每个尾实体对应的头实体数 (heads per tail), 则头实体和尾实体被替换的概率分别为 $\frac{tph}{tph + hpt}, \frac{hpt}{tph + hpt}$;

4. 在 TransH 之后, 2015 年 **TransR** 模型的提出标志着基于翻译距离的知识三元组表示学习进入成熟和高峰阶段。回顾 TransH 模型提出的超平面投影变换 $e_r = e - w_r^\top e w_r$, 其本质就是对 e 作以 w_r 为参数的映射 $e_r = f(e|w_r)$, 又结合 TransH 实际应用中发现 r, w_r 的正交约束对知识嵌入效果无显著影响, 不妨推断——解决复杂关系建模的根本在于将实体 e 和关系 r 嵌入不同的向量空间, 从 h 到 t 的“翻译”应在关系空间 (relation space) 内完成。“TransR”中的“R”即指“关系空间 (relation space)”。模型为每一种关系 r 对应一个专门的关系空间, 规定任意实体向量 e 到 r 所处关系空间的映射为线性映射 $e_r = M_r e$ 。由此可写出 TransR 模型的目标函数

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{F}} \left(\sum_{(h',r,t') \in \mathcal{F}'_{(h,r,t)}} [\gamma + d(h_r + r, t_r) - d(h'_r + r, t'_r)]_+ \right), \quad \text{s.t. } \|e\|_2 \leq 1, \|r\|_2 \leq 1, \|M_r e\|_2 \leq 1$$

式中 $e \in \mathbb{R}^k, r \in \mathbb{R}^d, M_r \in \mathbb{R}^{d \times k}$ 。TransR 模型中负样本集 $\mathcal{F}'_{(h,r,t)}$ 的生成方式与 TransH 一致。基于公开数据集的验证结果指出, 与 TransE、TransH 和其它更早期的方法相比, TransR 模型在关系预测、三元组分类等任务上具有明显的优势。但与 TransE、TransH 相比, **TransR** 模型引入的关系空间映射矩阵 $M_r \in \mathbb{R}^{d \times k}$ 参数量过大, 计算时间复杂度大大提高。

5. TransD 模型针对 TransR 提出的关系空间映射方法作进一步修改。名称中的“D”指代动态映射矩阵 (dynamic mapping matrix)。模型认为——同一个关系对应的头、尾两实体可能在类型上存在巨大的差异 (如三元组“美国”-“有”-“总统”, 其中头实体“美国”与尾实体“总统”不具备对等关系), 故将实体向量映射至关系空间时映射矩阵不应只与关系 r 有关, 还应考虑具体的头、尾实体属性 h, t , 这便是所谓的“动态映射”。自然地映射过程由 $e_r = M_r e$ 变为

$$e_r = M_{re} e = (r_p e_p^\top + I) e, \quad e, e_p \in \mathbb{R}^n, \quad e_r, r, r_p \in \mathbb{R}^m, \quad M_{re}, I \in \mathbb{R}^{m \times n}$$

上式中 M_{re} 表示将 e 映射至 r 所处关系空间的矩阵。为避免参数量过多, 模型认为 M_{re} 可由两个映射向量 (projection vector) r_p, e_p 表征, 加上单位阵 I 以保证 M_{re} 满秩。综上 **TransD** 模型学习知识三元组时实际上对每一个实体和关系学习两个向量表征 $\{e, e_p, r, r_p\}$, 其中 e, r 分别捕捉实体和关系的本征含义, 而 e_p, r_p 也与相应实体、关系的含义相关, 但专用于构建映射矩阵。这一思路与各类矩阵分解模型和自注意力机制中的 Q, K, V (见第 24.4 节) 有异曲同工之妙。TransD 模型的目标函数和约束与 TransR 基本一致

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{F}} \left(\sum_{(h',r,t') \in \mathcal{F}'_{(h,r,t)}} [\gamma + d(h_r + r, t_r) - d(h'_r + r, t'_r)]_+ \right), \quad \text{s.t. } \|e\|_2 \leq 1, \|r\|_2 \leq 1, \|M_{re} e\|_2 \leq 1$$

6. TransSparse 模型直接继承自 TransR 和 TransD 模型, 聚焦于关系空间映射矩阵 M_r 的进一步修改。模型同样认为 TransR 中采用的映射矩阵复杂度过高, 但不同于 TransD 采用低秩矩阵降低复杂度, **TransSparse** 选择令映射矩阵为稀疏矩阵, 学习时只更新非零元素。模型名称中的“Sparse”即指“sparse transfer matrix”。模型进一步注意到, 尽管“映射矩阵应随关系 r 变化”已成共识, 但 TransR 和 TransD 均未考虑到映射矩阵的复杂度也应随 r 变化。基于此 TransSparse 引入了参数量正比于知识库中关系 r 复杂度的映射矩阵 M_r , 以适应知识表示学习的异质性 (heterogeneous) 和不平衡性 (unbalanced) 问题:

- 异质性指知识图谱中往往某些关系与大量的实体连接, 而另一些关系的连接实体数则极少。显然知识库中连接数越多的关系反映的知识也就越丰富, 需要更多的参数学习以避免欠拟合, 而对于连接

数较少的关系则应减少学习参数量以防过拟合。由此定义关系空间稀疏转移矩阵为

$$M_r = M_r(\theta_r) \in \mathbb{R}^{n \times n}, \quad \theta_r = 1 - \frac{(1 - \theta_{\min})N_r}{\max_r\{N_r\}}$$

上式中 $\theta_r \in [0, 1]$ 表示矩阵 M_r 的稀疏度, M_r 的非零元素数目为 $\lfloor \theta_r \times n \times n \rfloor$, 其中 $\lfloor \cdot \rfloor$ 表示下界取整; θ_{\min} 为预设超参; N_r 表示关系 r 连接的实体对数目。采样上式定义的转移矩阵的 TransSparse 模型被称为 TransSparse (share), 意指同属一个关系 r 的所有实体共用一个 M_r ;

- 不平衡性指知识图谱中某些关系涉及更多的头实体, 而另一些关系所关联的尾实体则更多。为考虑不平衡性则转移矩阵的参数量也应与具体的实体 e 相关

$$M_r^e = M_r^e(\theta_r^e) \in \mathbb{R}^{n \times n}, \quad \theta_r^e = 1 - \frac{(1 - \theta_{\min})N_r^e}{\max_{r,e}\{N_r^e\}}$$

采样上式定义的转移矩阵的 TransSparse 模型被称为 TransSparse (separate), 同 TransD 模型一致同时区分了实体 e 对转移矩阵的影响。

进一步介绍基于稀疏度 θ 的稀疏矩阵 $M(\theta)$ 构建方法。为保证 $M(\theta)$ 满秩, 优先将其对角线元素设为非零元素。若 $\lfloor \theta_r \times n \times n \rfloor > n$, 则剩余的非零元素考虑两种排列方法:

- 结构化稀疏矩阵 (**structured sparse matrix**): 非零元素均匀沿副对角线依次分布的对称阵, 便于矩阵和向量相乘 (**matrix-vector products**) 运算;
- 非结构化稀疏矩阵 (**unstructured sparse matrix**): 非零元素无规律均匀分布, 计算不便, 但一般性更强。

以上即为 TransSparse 模型的核心贡献, 模型的目标函数为

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{F}} \left(\sum_{(h',r,t') \in \mathcal{F}'_{(h,r,t)}} [\gamma + d(h_r + r, t_r) - d(h'_r + r, t'_r)]_+ \right), \quad \text{s.t. } \|e\|_2 \leq 1, \|r\|_2 \leq 1, \|M(\theta)e\|_2 \leq 1$$

基于公开数据集的验证结果指出, 在连接预测和知识三元组分类任务中 TransSparse 模型的表现整体优于 Trans(E,H,R,D) 等对照模型, 而且 TransSparse (separate) 略优于 TransSparse (share), 而非结构化稀疏矩阵建模略优于结构化稀疏矩阵建模;

7. 由 TransD 提出团队进一步推出的 KG2E 模型则对 TransE 以来定义的“翻译”过程作了更颠覆性的理解。翻译距离模型普遍选择将实体与关系嵌入点向量空间, 其背后的假设是认为已有的知识三元组均为正确且确定性的。而 **KG2E** 模型考虑了实体和关系本身可能存在的不确定性, 将实体与关系嵌入概率密度空间, 此时的距离度量 $d(h + r, t)$ 也从点间距离变为概率分布间的距离。具体地模型选择多元高斯分布作为嵌入空间 $e \sim N(\mu_e, \Sigma_e)$, $r \sim N(\mu_r, \Sigma_r)$, 则模型优化参数也由实体和关系的嵌入向量变为相应高斯分布的参数 $\mu_e, \mu_r \in \mathbb{R}^k$, $\Sigma_e, \Sigma_r \in \mathbb{R}^{k \times k}$ 。假设三元组 (h, r, t) 的嵌入已知, 则正确的翻译应使得 $t - h \sim N(\mu_t - \mu_h, \Sigma_t + \Sigma_h)$ 与分布 $N(\mu_r, \Sigma_r)$ 尽可能一致。考虑两种满足对称性的概率分布距离度量方法:

- 基于 **KL 散度**: 因为 KL 散度本身不满足对称性, 则定义

$$d(h + r, t) = \frac{1}{2} [D_{KL}(N_{t-h} \| N_r) + D_{KL}(N_r \| N_{t-h})]$$

$$D_{KL}(N_{t-h} \| N_r) = \frac{1}{2} \left[\ln \frac{|\Sigma_r|}{|\Sigma_t + \Sigma_h|} + \text{tr} (\Sigma_r^{-1} (\Sigma_t + \Sigma_h)) + (\mu_t - \mu_h - \mu_r)^\top (\Sigma_t + \Sigma_h)^{-1} (\mu_t - \mu_h - \mu_r) - k \right]$$

其中多元高斯分布及其 KL 散度推导详见第 23.7.2 节;

- 基于期望似然 (**expected likelihood**): 即 $h + r - t = 0$ 的概率密度, 为方便计算再取负对数

$$d(h + r, t) = -\ln N(0 | \mu_h + \mu_r - \mu_t, \Sigma_h + \Sigma_r + \Sigma_t)$$

$$= \frac{1}{2} [k \ln 2\pi + \ln |\Sigma_h + \Sigma_r + \Sigma_t| + (\mu_t - \mu_h - \mu_r)^\top (\Sigma_h + \Sigma_r + \Sigma_t)^{-1} (\mu_t - \mu_h - \mu_r)]$$

综上得到 KG2E 模型的目标函数。因为模型并未显式地区分实体和关系的嵌入空间, 故其目标函数在形式上与 TransE 基本一致

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{F}} \left(\sum_{(h',r,t') \in \mathcal{F}'_{(h,r,t)}} [\gamma + d(h + r, t) - d(h' + r, t')]_+ \right), \quad \text{s.t. } \|\mu\|_2 \leq 1, c_{\min} I \leq \Sigma \leq c_{\max} I$$

25.10.1 KR-EAR (knowledge representation learning with entities, attributes and relations, IJCAI-16)

1. KR-EAR 模型由清华大学研究团队于 2016 年提出³⁰。不同于传统知识表示学习聚焦的实体-关系嵌入，KR-EAR 模型研究实体-属性-关系嵌入。在具体的嵌入方法层面，模型任遵循经典翻译距离模型的思路；
 2. 第 25.9 节已指出，与一般图表示学习相比，知识表示学习的特点在于额外考虑了关系（连边）的嵌入。而本节上文进一步指出关系表示学习的难点在于如何表示复杂关系（自反关系、一对多、多对一等等）。除了设计更复杂的知识表示算法外，另一部分研究认为复杂关系存在的根源在于经典的实体-关系三元组 (h, r, t) 建模过于粗糙，应进一步细化知识三元组的类型；
 3. KR-EAR 模型即认为应将知识三元组分为两类：
 - 关系三元组 (relational triples): $S \subseteq E \times R \times E$, 其中 S, E, R 分别表示知识三元组、实体和关系集合。此类三元组即是经典认为的 (h, r, t) 三元组，其中关系 r 包括父母、首都、作者等等，连接头尾两个实体，多为一对一连接；
 - 属性三元组 (attributional triples): $Y \subseteq E \times A \times V$, 其中 Y, E, A, V 分别表示属性三元组、实体、属性和属性值集合。此类三元组可写为 (e, a, v) , 其中属性 a 多包括国籍、性别、种族、区域等等，连接一个实体和对应的一个特征，且往往大量实体对应同一个特征。模型主要考虑离散属性。
- 通过区分关系三元组 S 与属性三元组 Y , 模型的目标函数自然变为最大化联合条件概率分布 $P(S, Y | \mathbf{X})$

$$(\max) \quad P(S, Y | \mathbf{X}) = P(S | \mathbf{X})P(Y | \mathbf{X}) = \prod_{(h,r,t) \in S} P((h, r, t) | \mathbf{X}) \prod_{(e,a,v) \in Y} P((e, a, v) | \mathbf{X})$$

式中 \mathbf{X} 表示关系三元组和属性三元组的嵌入向量。后续为避免混淆将用斜体表示变量，而正体加粗表示变量对应的向量。如 e 表示某实体， \mathbf{e} 表示实体 e 对应的表示向量；

4. 实现上式第一部分 $\prod_{(h,r,t) \in S} P((h, r, t) | \mathbf{X})$ 的模型被称为关系三元组编码器 (relational triplet encoder)，对应一般的知识表示模型。但已有研究一般不整体嵌入三元组 (h, r, t) ，而是区分实体与关系，即研究

$$\begin{aligned} (\max) \quad & \prod_{(h,r,t) \in S} P(h|r, t, \mathbf{X}) \cdot P(t|h, r, \mathbf{X}) \cdot P(r|h, t, \mathbf{X}) \\ &= \prod_{(h,r,t) \in S} \left(\frac{\exp\{g(\mathbf{h}, \mathbf{r}, \mathbf{t})\}}{\sum_{\hat{h} \in E} \exp\{g(\hat{\mathbf{h}}, \mathbf{r}, \mathbf{t})\}} \right) \cdot \left(\frac{\exp\{g(\mathbf{h}, \mathbf{r}, \mathbf{t})\}}{\sum_{\hat{t} \in E} \exp\{g(\mathbf{h}, \hat{\mathbf{r}}, \mathbf{t})\}} \right) \cdot \left(\frac{\exp\{g(\mathbf{h}, \mathbf{r}, \mathbf{t})\}}{\sum_{\hat{r} \in R} \exp\{g(\mathbf{h}, \mathbf{r}, \hat{\mathbf{t}})\}} \right) \end{aligned}$$

式中 $g(\cdot)$ 为量化三元组 (h, r, t) 嵌入效果的评分函数，可选择任意翻译距离模型，如 TransE 和 TransR 等

$$g(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \begin{cases} -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\| + b_1, & \text{TransE} \\ -\|\mathbf{hM}_r + \mathbf{r} - \mathbf{tM}_r\| + b_1, & \text{TransR} \end{cases}$$

5. 实现目标函数第二部分 $\prod_{(e,a,v) \in Y} P((e, a, v) | \mathbf{X})$ 的模型被称为属性三元组编码器 (attributional triplet encoder)，是 KR-EAR 模型的核心贡献。实体 e 的表示已由关系三元组编码器得到，故属性三元组编码器的目标在于确定属性 a 和属性值 v 的编码。记 $Y(e) = \{(e, \hat{a}, \hat{v}) | (e, \hat{a}, \hat{v}) \in Y\}$ 表示实体 e 除 (e, a, v) 之外的所有属性三元组集合，则 $P((e, a, v) | \mathbf{X})$ 可分解为

$$P((e, a, v) | \mathbf{X}) \propto P(v | e, a, \mathbf{X}) \cdot P((e, a, v) | Y(e))$$

- $P(v | e, a, \mathbf{X})$ 表示给定实体 e 和属性 a 条件下属性值为 v 的概率，可建模为一个关于属性预测的多分类问题：

$$P(v | e, a, \mathbf{X}) = \frac{\exp\{h(\mathbf{e}, \mathbf{a}, \mathbf{v})\}}{\sum_{\hat{v} \in V_a} \exp\{h(\mathbf{e}, \mathbf{a}, \hat{v})\}}, \quad h(\mathbf{e}, \mathbf{a}, \mathbf{v}) = -\|f(\mathbf{eW}_a + \mathbf{b}_a) - \mathbf{v}\| + b_2$$

³⁰Lin, Y., Liu, Z., Sun, M., 2016. Knowledge representation learning with entities, attributes and relations, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, New York, New York, USA, pp. 2866–2872.

式中 V_a 为属性 a 的所有取值集合； $h(\cdot)$ 为量化属性三元组 (e, a, v) 表示效果的评分函数，其量化方法是将实体表示 \mathbf{e} 由全连接网络映射至属性域，比较映射结果 $f(\mathbf{eW}_a + \mathbf{b}_a)$ 与属性值嵌入 \mathbf{v} 的差距； $\mathbf{W}_a, \mathbf{b}_a$ 为属性 a 对应的全连接层参数。上式仅优化属性值 v 的表示 \mathbf{v} ，而属性 a 的表示 \mathbf{a} 则由 $P((e, a, v)|Y(e))$ 建模；

- $P((e, a, v)|Y(e))$ 表示在实体 e 的其它属性确定的条件下属性三元组 (e, a, v) 存在概率，用于捕捉属性间的相关性（例如英国籍的作者母语往往是英文）。同样地基于 softmax 函数建模

$$P((e, a, v)|Y(e)) = \frac{\exp\{z(\mathbf{e}, \mathbf{a}, \mathbf{v}, Y(e))\}}{\sum_{\hat{v} \in V_a} \exp\{z(\mathbf{e}, \mathbf{a}, \hat{v}, Y(e))\}}, \quad z(\mathbf{e}, \mathbf{a}, \mathbf{v}, Y(e)) \propto \sum_{(e, \hat{a}, \hat{v}) \in Y(e)} P((a, v)|(\hat{a}, \hat{v})) \cdot (\mathbf{a}^\top \hat{\mathbf{a}})$$

式中 $z(\cdot)$ 为量化不同属性相关性的评分函数，通过不同属性嵌入向量的点积 $\mathbf{a}^\top \hat{\mathbf{a}}$ 可使得相关的属性具有相近的嵌入结果；条件概率 $P((a, v)|(\hat{a}, \hat{v}))$ 基于经验数据确定。

6. 最后考虑表示向量的模长约束，得到 KR-EAR 模型的目标函数如下

$$\begin{aligned} (\max) \quad O(\mathbf{X}) &= \ln P(S, Y|\mathbf{X}) + \gamma C(\mathbf{X}) && \text{(KR-EAR 目标函数)} \\ C(\mathbf{X}) &= \sum_{e \in E} [\|\mathbf{e}\| - 1]_+ + \sum_{r \in R} [\|\mathbf{r}\| - 1]_+ + \sum_{e \in E} \sum_{i=1}^{|A|} [\|\mathbf{eW}_i + b_i\| - 1]_+ + \sum_{v \in V} [\|\mathbf{v}\| - 1]_+ \end{aligned}$$

25.11 融合大语言模型的图深度学习

25.11.1 GLEM 框架 (graph and language learning by expectation-maximization, ICLR 2023)

1. 文本属性图 (text-attributed graph, TAG) 是指节点特征为文本的图结构。捕捉文本属性图的特征要求同时捕捉节点文本描述的语义特征和图结构的拓扑特征；
2. 语言模型和图神经网络分别是目前学习语义信息和拓扑信息的主流工具。以节点分类问题为例讨论融合语言模型与图模型的文本属性图学习架构。记图结构 $G_S = (V, A, s_V)$ ，其中 V 表示节点集合， A 为邻接矩阵， $s_V = \{s_n | \forall n \in V\}$ 为节点文本特征集合，则节点分类的目标是基于图中的带标签节点 y_L ， $L \subset V$ 预测其它无标签节点的标签 y_U ， $U = V \setminus L$ ：
 - 语言模型可直接基于每一节点的描述文本进行节点分类

$$p_\theta(y_n|s_n) = \text{softmax}(\text{MLP}_{\theta_2}(h_n)), \quad h_n = \text{SeqEnc}_{\theta_1}(s_n)$$

式中 $\text{SeqEnc}_{\theta_1}(\cdot)$ 表示参数为 θ_1 的文本编码模型，可将自然语言 s_n 编码为表示向量 h_n ，再经线性映射及激励函数即可预测节点 n 的标签 y_n 。但语言模型无法捕捉图拓扑信息，在节点文本描述信息有限时无法准确预测标签；

- 具有 L 层结构的图模型可基于图的拓扑结构预测节点分类

$$p_\phi(y_n|A) = \text{softmax}(h_n^L), \quad h_n^l = \sigma(\text{AGG}_\phi(\text{MSG}_\phi(h_{NB_n}^{l-1}), A))$$

式中 ϕ 表示图模型参数； h_n^l , $l = \{1, \dots, L\}$ 为图模型中第 l 层结构的输出中对应于节点 n 的表示向量； $\text{AGG}_\phi(\cdot)$, $\text{MSG}_\phi(\cdot)$ 分别表示图模型的信息聚合和传递模块； $\sigma(\cdot)$ 表示激励函数。图模型的缺陷在于无法处理文本信息 s_n ，需要外部输入合理的初始节点表示 h_n^0 。

综上，最直观的挖掘文本属性图节点特征的方法便是串联语言模型和图模型——基于语言模型提取节点初始向量表征，再基于图模型进行消息传递预测节点标签；

3. 然而，前沿的语言模型往往具有极大的规模，当图结构较大时串联语言模型和图模型的学习范式将产生巨大的训练时间成本。研究的目标便是提出一种基于 EM 算法（见第 23.8 节）的新型语言模型-图模型联合学习框架 (GLEM)。方法为语言模型和图模型设计单独的目标函数，从而允许两模型独立训练，减小空间复杂度；又通过相互提供伪标签的方式实现语言模型和图模型的交互，以保证语义信息和拓扑信息间的有效传递³¹；

³¹ZZhao, J., Qu, M., Li, C., Yan, H., Liu, Q., Li, R., Xie, X., Tang, J., 2023. Learning on large-scale text-attributed graphs via variational inference, *The 11th International Conference on Learning Representations*, Kigali Rwanda, pp. 1-13. <https://arxiv.org/abs/2210.14709>

4. 以节点分类任务为例具体介绍算法细节。易知节点分类的目标为极大化带标签节点的标签的对数似然

$$(\max) \quad \ln L(y_L|s_V, A)$$

然而因为隐变量 y_U (即未知标签节点的标签) 的存在, 上述优化问题不容易求解。EM 算法 (见第 23.8 节) 便是求解带隐变量极大似然问题的经典方法。方法通过交替执行期望步 (expectation step) 和最大化步 (maximization step) 求解此类优化问题;

5. 期望步的目标在于估计隐变量的后验分布 $p(y_U|s_V, A, y_L)$, 可由变分推断近似 (见第 27.2 节)。变分推断的目标是基于简单的变分分布拟合复杂分布。此处选择变分分布为 $q(y_U|s_U)$, 则期望步的目标在于通过优化 $q(y_U|s_U)$ 最小化其与后验分布 $p(y_U|s_V, A, y_L)$ 间的 KL 散度

$$(\min) \quad D_{KL}(q(y_U|s_U) \| p(y_U|s_V, A, y_L)) \quad \text{or} \quad D_{KL}(p(y_U|s_V, A, y_L) \| q(y_U|s_U))$$

变分推断的优化对象为变分分布 $q(y_U|s_U)$ 。若以上述左式为优化目标, 则按 KL 散度定义展开后两项均与 $q(y_U|s_U)$ 有关; 若以右式为优化目标, 则展开后仅一项与 $q(y_U|s_U)$ 有关。故选择右式作为期望步的目标函数。注意到变分分布 $q(y_U|s_U)$ 本质上是仅基于节点描述文本预测节点标签, 故可由语言模型拟合; 而后验分布 $p(y_U|s_V, A, y_L)$ 是基于节点特征和拓扑结构预测节点标签, 可由图模型拟合

$$q(y_U|s_U) \simeq q_\theta(y_U|s_U) = \prod_{n \in U} q_\theta(y_n|s_n),$$

$$p(y_U|s_V, A, y_L) \simeq p_\phi(y_U|s_V, A, y_L) = \prod_{n \in U} p_\phi(y_n|s_V, A, y_L) \simeq \prod_{n \in U} p_\phi(y_n|s_V, A, y_L, \hat{y}_{U \setminus n})$$

式中 $\hat{y}_{U \setminus n}$ 表示由语言模型预测的除节点 n 以外的其它无标签节点的伪标签。引入语言模型生成的伪标签 \hat{y}_U 是为使后续图模型训练时不仅考虑真实标签 \hat{y}_L , 也考虑语言模型所学习的知识, 实现知识蒸馏。将上式代入前述 KL 散度 $D_{KL}(p(y_U|s_V, A, y_L) \| q(y_U|s_U))$ 有

$$D_{KL}(p(y_U|s_V, A, y_L) \| q(y_U|s_U)) = -\mathbb{E}_{p(y_U|s_V, A, y_L)} [\ln q(y_U|s_U)] + \text{const} = -\sum_{n \in U} \mathbb{E}_{p_\phi(y_n|s_V, A, y_L, \hat{y}_{U \setminus n})} [\ln q_\theta(y_n|s_n)] + \text{const}$$

综上得到 GLEM 框架下期望步的目标函数 $O(\theta)$ 为

$$(\max) \quad O(\theta) = \alpha \sum_{n \in U} \mathbb{E}_{p_\phi(y_n|s_V, A, y_L, \hat{y}_{U \setminus n})} [\ln q_\theta(y_n|s_n)] + (1 - \alpha) \sum_{n \in L} \ln q(y_n|s_n) \quad (\text{GLEM 期望步})$$

因为变分推断的优化对象为变分分布 $q(y_U|s_U)$, 故上式为语言模型 θ 的目标函数, 而图模型 ϕ 不参与优化。式中第一项为变分推断的优化目标, 要求语言模型拟合图模型的预测伪标签 $y_n, \forall n \in U$, 故第一项可使语言模型 θ 学习图模型 ϕ 的知识, 从而捕捉拓扑特征。第二项为补充的监督学习目标, 要求语言模型正确预测真实标签 $y_n, \forall n \in L$ 。 α 为超参;

6. 进一步介绍最大化步。EM 算法的最大化步要求固定隐变量的后验分布 (即经变分推断拟合的变分分布 $q(y_U|s_U)$) 并最大化对数联合似然

$$(\max) \quad \ln L(y_L, y_U|s_V, A) = \mathbb{E}_{q(y_U|s_u)} \left[\ln \frac{p(y_L, y_U|s_V, A)}{q(y_U|s_u)} \right] = \mathbb{E}_{q(y_U|s_u)} [\ln p(y_L, y_U|s_V, A)] + \text{const}$$

式中联合分布 $p(y_L, y_U|s_V, A)$ 表示给定节点特征和拓扑结构下节点的标签概率, 可由图模型 ϕ 拟合

$$\mathbb{E}_{q(y_U|s_u)} [\ln p(y_L, y_U|s_V, A)] \simeq \mathbb{E}_{q_\theta(y_U|s_U)} \left[\sum_{n \in V} \ln p_\phi(y_n|s_V, A, y_L, \hat{y}_{U \setminus n}) \right]$$

再由蒙特卡洛方法 (见第 27.1 节) 近似上述期望, 从而得到 GLEM 框架下最大化步目标函数 $O(\phi)$ 为

$$(\max) \quad O(\phi) = \beta \sum_{n \in U} \ln p_\phi(\hat{y}_n|s_V, A, y_L, \hat{y}_{U \setminus n}) + (1 - \beta) \sum_{n \in L} \ln p_\phi(y_n|s_V, A, y_L, \hat{y}_{U \setminus n}) \quad (\text{GLEM 最大化步})$$

显然上式为图模型模型 ϕ 的目标函数，而语言模型 θ 不参与优化。式中第一项为基于蒙特卡洛方法的近似期望 $\mathbb{E}_{q_\theta(y_U|s_U)} [\sum_{n \in V} \ln p_\phi(y_n|s_V, A, y_L, \hat{y}_{U \setminus n})]$ ，以采样的伪标签 $\hat{y}_U \sim q_\theta(y_U|s_U)$ 为预测目标，故第一项可使图模型 ϕ 学习语言模型 θ 的知识，从而捕捉语义特征。第二项为监督学习目标，要求图模型正确预测真实标签 $y_n, \forall n \in L$ 。 β 为超参；

7. 通过交替执行上述的期望步和最大化步以交替优化语言模型和图模型，可实现语言模型和图模型知识间的双向传递，最终使得语言模型和图模型均可同时捕捉语义信息和拓扑信息，因此两类模型训练完成后均可用于推理预测；
8. 以上便为 GLEM 框架的主要内容，实验表明所提框架在文本属性图学习上具有显著的效率和性能优势。

25.11.2 Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning (ICLR 2024)

1. 文本属性图 (text-attributed graph, TAG) 是指节点特征为文本的图结构。捕捉文本属性图的特征要求同时捕捉节点文本描述的语义特征和图结构的拓扑特征。语言编码和图神经网络分别是目前学习语义信息和拓扑信息的最主流工具。
2. 早期的文本属性图学习范式基于 word2vec (第 24.5 节) 等浅层语言编码方法生成节点特征文本的表征向量，再将其作为图神经网络的输入实现图学习。然而此类浅层语言编码仅具有有限的语义理解能力，从而限制了下游图模型的学习效果；
3. 后续的研究则尝试将浅层语言编码模型替换为更复杂的语言模型。以 BERT (第 24.7.2 节) 为代表的深度语言模型开启了自然语言处理领域的预训练-微调范式——首先基于海量语料库预训练模型，再根据下游具体任务对其进行微调从而得到具有更强语义理解能力的模型。在应用于文本属性图学习时，可直接以微调后的固定参数的语言模型替代浅层语言编码模型，也可将文本属性图学习任务作为微调任务同步优化语言模型与图模型 (如第 25.11.1 节介绍的 GLEM 框架)。此类学习范式往往需要复杂的设计和较高的计算复杂度，从而无法支持具有更大规模与更强语义理解能力的大语言模型；
4. 近年来通用大语言模型的成功为自然语言处理领域提供了全新的“预训练-提示-预测”学习范式。不同于之前的“预训练-微调”范式，大语言模型通过采用更复杂的模型结构并在更大规模的预料中预训练学习通用知识，使得无需针对下游任务微调，只需提供包含任务需求的提示词即可直接输出解决方案。然而大语言模型的应用也面临一系列挑战：
 - 一方面目前最成功的大语言模型多为闭源模型，用户无法获得内部的文本编码仅能得到生成文本，同时无法对模型进行微调；
 - 对于开源大语言模型，因为模型的巨大规模也难以对其成功微调，并需消耗大量计算资源。

因此学界提出“语言模型即服务 (language model as a service, LMaaS)”的概念，以期在无需额外计算资源和专业知识的前提下利用大语言模型；

5. 综上，研究遵循“语言模型即服务”的目标，提出了一种融合大语言模型的文本属性图学习框架³²。方法无需微调大语言模型或获取其文本编码，而是提出“解释即特征 (explanations as features)”的概念，基于大语言模型生成对原始节点描述文本的解释文本以提供更丰富的语义信息。再由传统的小规模语言模型编码原始文本和解释文本生成图神经网络的输入。由此小规模语言模型作为沟通大语言模型和图模型的翻译器；
6. 以引文关系图的节点分类任务为例。记图结构 $G_S = (V, A, s_V)$ 。其中 V 表示节点集合，每一节点对应一篇论文； A 为邻接矩阵； $s_V^{orig} = \{s_n^{orig} | \forall n \in V\}$ 为节点的原始文本特征集合，其中 s_i^{orig} 记录了论文 i 的题目和摘要信息。则节点分类的目标是基于图中的带标签节点 $y_L, L \subset V$ 预测其它无标签节点的标签 $y_U, U = V \setminus L$ ；
7. 具体介绍算法步骤：
 - 首先基于大语言模型生成解释文本和预测文本。对于每篇论文 i ，输入其题目和摘要信息 s_i^{orig} ，要求大语言模型按概率从高到低预测其所属的类别，并解释预测原因，记解释文本为 s_i^{expl} 。注意到大语

³²He, X., Bresson, X., Laurent, T., Perold, A., LeCun, Y., Hooi, B., 2024. Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning, *The Twelfth International Conference on Learning Representations*, Vienna Austria. [https://openreview.net/forum?id=RXFVcynVe1&referrer=%5Bthe%20profile%20of%20Yann%20LeCun%5D\(%2Fprofile%3Fid%3D~Yann_LeCun1\)](https://openreview.net/forum?id=RXFVcynVe1&referrer=%5Bthe%20profile%20of%20Yann%20LeCun%5D(%2Fprofile%3Fid%3D~Yann_LeCun1))

言模型的预测为零样本 (zero-shot) 预测，模型完全基于其预训练阶段学习的通用知识预测论文所属的类别；

- 进一步构建并微调小型语言模型以提取 s_i^{orig}, s_i^{expl} 语义特征。语言模型 LM 提取表示向量 h_i^{orig}, h_i^{expl}

$$h_i^{orig} = LM(s_i^{orig}) \in \mathbb{R}^d, \quad h_i^{expl} = LM(s_i^{expl}) \in \mathbb{R}^d$$

式中 d 表示文本表示向量的维数。可构建不同的语言模型分别提取 h_i^{orig}, h_i^{expl} ，但实验表明也可由同一个语言模型实现。在 h_i^{orig}, h_i^{expl} 的基础上接入线性映射和 softmax 函数即可预测论文 i 所属的类别

$$y_i^{orig} = \text{softmax}(\text{MLP}(h_i^{orig})) \in \mathbb{R}^C, \quad y_i^{expl} = \text{softmax}(\text{MLP}(h_i^{expl})) \in \mathbb{R}^C$$

式中 C 表示总类别数。以交叉熵作为目标函数即可实现对小型语言模型的微调；

- 另外对于大语言模型生成的预测信息，按独热编码表示为 $p_{i1}, \dots, p_{ik} \in \mathbb{R}^C$ 。其中 p_{il} 表示大语言模型预测的论文 i 最可能属于的类别， p_{ik} 表示论文 i 第 k 可能属于的类别。进而生成预测表示向量 h_i^{pred}

$$h_i^{pred} = \text{MLP}(p_i) \in \mathbb{R}^{dp}, \quad p_i = [p_{i1} : \dots : p_{ik}] \in \mathbb{R}^{kC}$$

式中 p_i 表示由 p_{i1}, \dots, p_{ik} 按顺序合并而成的向量。上述 MLP 模型也按小型语言模型的方法进行微调，最终得到经大语言模型增强后的节点 i 表示向量 $h_i = \{h_i^{orig}, h_i^{expl}, h_i^{pred}\}$ ；

- 最后固定语言模型参数，以 h_i 为输入训练图模型完成图节点分类。作者针对 $h_i^{orig}, h_i^{expl}, h_i^{pred}$ 分别构建结构相同的图模型 $f_{orig}, f_{expl}, f_{pred}$ ，集成三个模型的结果预测节点 i 的类别 \hat{y}_i

$$\hat{y}_i = \text{mean}(\hat{y}_i^{orig}, \hat{y}_i^{expl}, \hat{y}_i^{pred}) \in \mathbb{R}^C, \quad \hat{y}_i^{orig/expl/pred} = f_{orig/expl/pred}(h_i^{orig/expl/pred}, A) \in \mathbb{R}^C$$

8. 作者也提供了所提方法有效性的理论条件。记随机变量 E 表示大语言模型生成的解释文本， Z_L, Z 分别表示大语言模型和小型语言模型对原始文本的编码， y 表示标签， $H(\cdot|\cdot)$ 表示条件熵 (conditional entropy，见第 23.9.6 节)。所提方法基于 E, Z 预测 y ，在 E, Z 已知时 y 的不确定性可以由 $H(y|Z, E)$ 量化，越小表示越有助于预测。为证明所提方法有效，只需证 $H(y|Z, E) < H(y|Z)$ 。易证其成立的一个充分条件为

$$H(y|Z) - H(y|Z, Z_L) > H(Z_L|E) \quad (\text{所提方法有效的充分条件})$$

式中 $H(Z_L|E) > 0$ 定义为 E 已知时大语言模型编码的不确定性，可理解为解释文本信息相较于模型认知的损失，越小表示生成的解释文本 E 可越充分地反映 Z_L 的信息；同理 $H(y|Z) - H(y|Z, Z_L)$ 的物理意义大语言模型编码 Z_L 相较于小规模语言模型编码 Z 对于标签预测的效益， $H(y|Z) - H(y|Z, Z_L) = 0$ 表示大语言模型相较于小规模语言模型无法提供额外信息。因此，**所提方法有效的充分条件可解释为大语言模型相较于小规模语言模型所额外提供的信息量大于其生成解释文本的信息损失**。进一步给出证明过程。根据条件熵的性质，有

$$I(y, Z_L|Z, E) = H(y|Z, E) - H(y|Z, Z_L, E) = H(Z_L|Z, E) - H(Z_L|Z, y, E)$$

式中 $I(y, Z_L|Z, E)$ 为条件互信息 (conditional mutual information，见第 23.9.6 节)，表示随机变量 Z, E 已知时 y, Z_L 之间的相关性，具有非负性。由此有

$$H(y|Z, E) \leq H(y|Z, Z_L, E) + H(Z_L|Z, E) \leq H(y|Z, Z_L) + H(Z_L|E)$$

式中第二次放缩基于条件熵的性质——减少条件变量后条件熵增加或不变。最后代入充要条件命题得证

$$H(y|Z, E) < H(y|Z, Z_L) + [H(y|Z) - H(y|Z, Z_L)] = H(y|Z)$$

25.11.3 Exploring the potential of large language models (LLMs) in learning on graphs (ACM SIGKDD Explorations Newsletter, 2024)

1. 文本属性图 (text-attributed graph, TAG) 是一类常见的图结构。在数学上记图 G_S 的节点集为 \mathcal{V} ，邻接矩阵为 $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ ，则文本属性图的特点是任意节点 $v_i \in \mathcal{V}$ 的特征由文本表示，记为 s_i 。引文关系图、产品关系图等均属于此类型；

2. 自然地，构建面向文本属性图的学习模型便要求同时捕捉图结构的拓扑信息和节点特征的语义信息。已有的研究大多简单地基于词袋模型、word2vec（见第 24.5 节）等浅层文本编码算法将节点属性向量化，再由图模型进行消息传递捕捉图拓扑信息。然而上述浅层文本编码算法无法捕捉词语在复杂语境下的动态语义，从而影响下游图模型学习效果；
3. 与早期的浅层文本编码算法相比，大语言模型表现出了更强的语境感知与语义理解能力，由此引发出了第一个猜想——**能否以大语言模型替代浅层语言模型编码节点特征文本，弥补图学习模型无法感知语义信息的缺陷？**另一方面，目前的大语言模型已经在推荐、排序等具有隐式图结构的问题上取得了巨大成功，从而自然好奇**大语言模型能否在不依赖图模型的情况下理解显式图结构，独立完成图学习任务？**基于此作者提出了两种融合大语言模型的图深度学习范式：
 - 将大语言模型用于特征提取及增强 (**LLMs-as-Enhancers**)：基于大语言模型提取节点描述文本语义信息，再由图模型进行图学习；
 - 将大语言模型直接用于图学习预测 (**LLMs-as-Predictors**)：完全依赖大语言模型捕捉拓扑特征和节点描述文本语义特征，独立进行图学习。该范式的难点在于如何设计有效的提示词引导大语言模型学习图拓扑特征。
4. 定义不同类型的大语言模型。基于模型的规模可分为以下三类：

预训练语言模型 (pre-trained language models, PLMs) 指较早期的相对规模较小的语言模型，如 BERT（见第 24.7.2 节）、Deberta 等等。此类模型已经完成预训练具有一定的语言理解能力，但还需根据下游任务进行后续微调；

深度语句嵌入模型 (deep sentence embedding models) 指在预训练语言模型的基础上进行针对性微调，具有完备的语句理解能力并可生成合理表示向量的模型，如 sentence-BERT（见第 24.7.3 节）等等。在应用前一般无需再对其微调；

通用大语言模型 (large language models, LLMs) 狹义上的大语言模型，较预训练语言模型具有更大的规模和更强的语义理解能力，即使不微调也能胜任绝大多数通用任务。可进一步分为开源和闭源两类。开源通用大语言模型（如 LLaMA）可生成文本的向量编码，并可由用户进一步微调，但微调难度大；而闭源模型（如 ChatGPT）则只能得到生成的文本回答，且无法微调。

上述预训练语言模型、深度语句嵌入模型和开源通用大语言模型均属于嵌入可见大语言模型 (**embedding-visible LLMs**)，闭源通用大语言模型则属于嵌入不可见大语言模型 (**embedding-invisible LLMs**)；
5. 作者聚焦于文本属性图的节点分类任务。考虑 CORA、PUMED、OGBN-ARXIV、OGBN-PRODUCTS 共四个常用的文本属性图数据集。对于 OGBN-ARXIV 和 OGBN-PRODUCTS 采用官方提供的训练集-测试集划分方法，对于 CORA 和 PUMED 则考虑两种训练集-测试集划分方法：

低标签率 (low-labeling-rate) 划分 对每一类别随机选择 20 个样本（节点）组成训练集，从剩余节点中随机选择 500 个节点组成验证集，最后再随机选择 1000 个节点组成测试集。该划分方法可保证训练集中包含尽可能丰富的样本，但每类样本数量有限；

高标签率 (high-labeling-rate) 划分 随机选择 60% 的样本组成训练集，20% 样本组成验证集，剩余 20% 组成测试集。该划分方法可生成与全样分布一致的训练集，使模型尽可能充分地学习优势类别的样本，但存在数据不平衡的问题。

LLMs-as-Enhancers

1. 考虑到大语言模型更全面的语言理解能力，作者提出了特征级 (**feature-level enhancement**) 和文本级 (**text-level enhancement**) 两种基于大语言模型的语义增强范式。前者基于节点描述文本 s_i 生成更能表征语义信息的表示向量 $h_i \in \mathbb{R}^d$ ；后者则是首先在初始描述文本 s_i 的基础上生成更全面的增强文本 s_i^{Aug} ，再对 s_i , s_i^{Aug} 向量化 $h_i, h_i^{Aug} \in \mathbb{R}^d$ 。对于特征级文本语义增强，作者又考虑了顺序串联型 (**cascading structure**) 和并行交互型 (**iterative structure**)（即第 25.11.1 节所介绍的 GLEM 训练架构）的两种大语言模型-图模型耦合结构；
2. 首先讨论基于大语言模型的特征级文本语义增强。作者考虑了浅层语言编码器 (TF-IDF、word2vec)、固

定参数和可学习的预训练语言模型 (Deberta)、固定参数的深度语句嵌入模型 (Sentence-BERT、e5-large、text-ada-embedding-002、Palm-Cortex-001) 和固定参数的通用大语言模型 (LLama) 等多种具备不同语义理解能力的语言模型，并结合多种图模型设计了大量节点分类实验比较不同模型和方法的效果，结果表明：

- 无论采用何种语言编码方法，引入图模型后均可显著提升节点分类效果；
- 当采用低标签率生成的训练集时，基于微调的语言模型可能无法取得可接受的效果。因为训练集中每一类别样本有限，微调语言模型可能陷入过拟合；
- 只需选择深度语句嵌入模型并与下游图模型简单串联便可取得很好的效果，即使固定参数的深度语句嵌入模型在预训练时并未学习拓扑特征；
- 简单的扩大语言模型规模（选择通用大语言模型）并不一定能提升节点分类效果，这可能与预训练的目标函数有关。

除比较节点分类效果外进一步设计实验评估不同语言模型和训练方法的可扩展性，具体关注语言模型和图模型训练阶段的耗时和内存占用指标：

- 尽管在训练样本充足的情况下 GLEM 框架表现出了更优的分类效果，但与简单的串联架构相比前者的训练时长和空间占用也显著更多；
- 在多种语言模型中，选择固定参数的深度语句嵌入模型具有显著的训练效率优势。

3. 特征级文本语义增强要求语言模型输出表征语义信息的表示向量，然而目前语义理解能力最先进的通用大语言模型（如 ChatGPT）多为闭源模型，无法生成向量编码仅能输出文本，故有必要讨论基于大语言模型的文本级文本语义增强方法。核心是基于通用大语言模型（实验中选择 GPT3.5）在原始节点文本特征的基础上生成信息量更丰富的增强文本。考虑两种文本增强方法：

- 一类是借鉴第 25.11.2 节所提出的方法，简称为“TAPE”方法。其中“TA”表示图节点的原始文本特征；“P”、“E”分别表示基于通用大语言模型预测的节点伪标签和解释文本，即为增强文本。为全面评价“TAPE”文本增强方法的有效性，后续实验中将分别评估伪标签和解释文本的效果。该方法的潜在缺点在于增强文本“P”、“E”的质量受大语言模型认知的影响，当大语言模型做出错误预测时相应的增强文本可能无助于下游图模型的学习；
- 作者又提出了基于知识提升的文本增强 (knowledge-enhanced augmentation, KEA) 方法。方法不要求通用大语言模型生成预测和解释文本，仅要求其识别原始文本中的科学术语并生成更丰富的术语说明作为增强文本。生成的增强文本可插入原文本中共同编码输入图模型，也可分别编码。与 TAPE 方法相比，KEA 方法并不要求大语言模型进行预测，故其增强文本的质量具有更高的鲁棒性。

4. 对于原始特征文本和补充的增强文本，基于小规模语言模型进行编码后输入串联的下游图模型进行学习。其中小规模语言模型可为预训练语言模型（实验中选择 Deberta），与图模型共同训练；也可为固定参数的深度语句嵌入模型（实验中选择 e5-large），不参与后续训练。实验结果表明：

- TAPE 方法的有效性主要源于解释文本提供的信息，相比之下预测伪标签的信息质量则受大语言模型预测准确性影响较大，其效果在不同数据集中具有较大差异；
- KEA 方法生成的增强文本有助于提升下游图模型的学习效果，且增强文本与原始文本共同编码或分别编码均可产生相近的效益；
- 无论是 TAPE 还是 KEA 方法，采用固定参数的深度语句嵌入模型而非待微调的预训练语言模型均可使下游的图模型取得更优的效果；
- 对于不同的数据集，KEA 方法与 TAPE 方法优劣不同。前者的效果更鲁棒，因其不需要通用大语言模型进行预测，而后者则大语言模型预测准确性较高的数据集中表现更为突出。

LLMs-as-Predictors

1. 上文讨论为基于大语言模型的文本属性图节点特征提取和增强方法更多聚焦于大语言模型的语义理解能力，而较少发挥其推理能力。目前通用大语言模型已经在多项任务中证明了出色的推理能力，并胜任多项具有隐式结构的任务。故本小节关注通用大语言模型是否可单独执行图结构上的预测任务。后文选择 GPT3.5 作为通用大语言模型，考虑 CORA、CITESEER、PUMED、OGBN-ARXIV、OGBN-PRODUCTS

共五个常用的文本属性图数据集：

- 首先关注大语言模型能否在不依赖图结构的前提下进行文本属性图节点分类（即将图节点分类问题简化为文本分类问题）。因为大语言模型 API 的访问时间成本无法将所有测试节点输入大语言模型，故仅从测试集中随机选择 200 个节点进行测试，并选择以下几种策略：

- 零样本提示词 (**zero-shot prompts**)：即不依赖任何训练样本，直接在提示词中提供节点特征要求大模型预测节点的标签；
- 少样本提示词 (**few-shot prompts**)：提示词中处待预测节点的特征外，还从训练集中随机选择若干个训练样本将其特征及标签也加入提示词；
- 带思维链的零样本提示词 (**zero-shot prompts with chain-of-thoughts**)：思维链 (**chain-of-thoughts**, CoT)³³在大量推理任务中表现了极佳的效果。大量实验指出，只需在提示词中加入“think it step by step”引导大语言模型按步推理便可显著提高推理结果的质量。其中逐步的推理逻辑便是思维链；
- 带思维链的少样本提示词 (**few-shot prompts with chain-of-thoughts**)：首先为大语言模型提供少量训练样本并加入“think it step by step”引导大语言模型生成思维链，再将生成的思维链与测试样本特征加入提示词进行预测。

实验发现在提示词中加入太多信息反而会降低大语言模型预测质量，故在小样本提示词中限制提供的训练样本最多为 2 个。为便于通过代码自动读取大语言模型生成的预测文本，可在提示词中要求大语言模型将其预测结果以 python 列表形式输出。实验结果指出：

- 通用大语言模型在部分数据集中表现了初步的预测可行性。在不依赖拓扑信息的前提下，其在大部分数据集上的表现与图模型仍有一定差距，但在 PUBMED 数据集上表现更好；
- 尽管存在较高的错误率，但其部分错误预测存在合理性；
- 思维链技巧在文本属性图预测中未能提供性能提升。因为不同于数理推理存在清晰的思维链，文本属性图预测可能存在多种各异但合理的思维链；
- 具有相似语义的提示词可能产生截然不同的效果，其中在提示词中加入过多的信息可能反而会降低预测质量。

- 进一步探索大语言模型能否利用图结构以提升文本属性图的预测效果。因为提示词的规模存在限制，故仅将测试节点邻域的抽样节点信息加入提示词。注意到图模型多为两层结构，意味着基于二阶邻居信息即可进行节点预测，故实验中考虑测试节点的二阶邻居节点，并从中随机抽取 5 个邻居作为训练样本输入至提示词中。实验结果表明：

- 对于同构图 (**homophily graph**) 在提示词中加入邻居信息有助于提升大语言模型预测效果；
- 对于异构图 (**heterophily graph**) 在提示词中加入邻居信息反而可能降低大语言模型预测效果。因为异构图中节点倾向于与其它类别的节点产生联系，引入邻居信息反而会增加噪声。

- 不同于大多数机器学习模型，上述实验证明通用大语言模型在真实图结构上具有初步的零样本预测效果，但其 API 调用与推理过程较一般图模型具有更高的时间和金钱成本。因此一个合理的方案是基于知识蒸馏的思路由通用大语言模型生成伪标签以训练小型图模型。该思路面临的关键问题包括：

- 量化大语言模型生成的伪标签的置信度。作者尝试要求大语言模型直接输出置信度，但往往均输出 1。作者也不再考虑其它更复杂的设计；
- 选择合适的训练样本用于生成伪标签。图结构中往往存在某些节点较其它节点更重要，将其作为训练样本生成正确的伪标签可使图模型学得更丰富的知识。为简单起见作者选择低标签率随机抽样策略。具体地对于每一类别随机选择 20 个样本，并以其中的 75% 作为训练集，剩余作为验证集。

实验结果表明，通用大语言模型的伪标签可提供相对满意的信息量。又因为不同数据集中大语言模型的零样本预测精度存在差异，导致基于伪标签训练的图模型的精度存在差异；

- 最后考虑图学习中的分布外样本 (**out-of-distribution**) 问题。当测试样本的分布相对训练样本存在偏移时已有图模型的表现往往会显著下滑，而通用大语言模型则展示了其面对分布外语言样本的鲁棒

³³ 大语言模型的预训练：思维链 (Chain-of-thought, CoT) 原理详解：<https://baijiahao.baidu.com/s?id=1771926480948343579&wfr=spider&for=pc>

性。对于存在分布偏移的文本属性图，忽略拓扑信息的初步实验同样指出大语言模型预测器具有显著的鲁棒性，其在验证集和测试集中的预测表现几乎不存在差别。

25.12 物理/知识驱动的图深度学习

第 26 章

集成学习 (Ensemble Learning)

集成学习与深度学习、统计学习等并列，同为主流的机器学习模式。集成学习思想最早提出于上世纪 90 年代初，与其它模式不同，其通过构建并结合多个学习器（又称基学习器）来完成学习任务，有时也被称为多分类器系统、基于委员会的学习等。集成学习中最常见的基学习器为决策树模型，此时称相应的算法为基于决策树的集成学习算法。除此之外也常采用神经网络作为基学习器，此时集成学习更多地是作为避免神经网络过拟合的一种手段。

当算法中采用多种类型的基学习器时称为异质学习，属于集成学习的 **Stacking** 算法和 **Blending** 算法（**Blending** 的结构与 **Stacking** 基本一致）。当算法的基学习器均为同种类型时称为同质学习，是最常见的集成学习模式，分为 **Bagging** 算法和 **Boosting** 算法两类（理论上 **Stacking** 和 **Blending** 算法也可用于同质学习，但在实际任务中一般设置多类学习器以提高集成效果）。四类算法诞生的先后顺序为 **Boosting**(1990)、**Stacking**(1992)、**Bagging**(1996)、**Blending**(2009)。**Bagging** 和 **Boosting** 在训练和预测过程的差别如下：

1. 在训练方面，**bagging** 是通过对训练集的不同采样同时训练多个基学习器，基学习器之间无强依赖；**boosting** 则是串行学习，首先训练前一个基学习器，而后训练下一个基学习器修正上一学习器的预测误差；
2. 在预测方面，**bagging** 采用 **vote**（投票）的方式，采用得票数最多的决策作为决策结果，从而起到减小方差的效果；**boosting** 采用 **stack**（堆叠）的方式，将基学习器按照训练顺序叠加输出结果，形成最终的决策，每一个基分类器都会重点修正上一个基分类器的错误和不足，从而起到减小偏差的效果。

26.1 决策树模型 (**Decision tree**)

决策树算法起源于 1966 年 E.B.Hunt 等人的论文 *Experiments in Induction*。1979 年 Quinlan（罗斯·昆兰）提出 **ID3** 算法，掀起了决策树研究的热潮，决策树也成为机器学习的主流算法。决策树模型与线性模型、支持向量机模型等同属于统计模型，具有较强的可解释性，而且决策树算法是基于逻辑的（其它一般统计模型大多是基于非逻辑的，仅考虑统计关系）。

一棵决策树由一个根节点、若干个内部节点和若干个叶子结点组成。叶子结点位于树的最底端，对应决策结果。根节点与内部节点对应样本的各项属性（特征），根节点是树的第一个节点，与内部节点共同组成一系列判据。根节点中包含全体样本集，基于一系列判据各样本最终落入特定的叶子节点中，形成针对各样本的决策。生成决策树的过程就是从样本中选取合适的属性作为根节点与内部节点并确定相应分裂判据的过程。

26.1.1 决策树生成算法——ID3、C4.5、CART

ID3 算法

1. **ID3** 算法由 Quinlan 于 1979 年提出，是最经典的决策树算法。在算法中，节点的划分 (split) 基于信息增益 $I(X, Y)$ 确定，信息增益越大的特征越适合用来分类¹，首先将介绍信息增益的概念；
2. 第 23.9.6 节介绍了信息熵 (information entropy) $H(X)$ 的概念，表示对随机变量 X 所含信息量的度量

$$H(X) = - \sum_i^n p(X = x_i) \ln p(X = x_i)$$

¹根据监督学习任务的不同，决策树又分分类树和回归树。分类树与回归树一般具有不同的划分判据。在讨论 **ID3** 算法与 **C4.5** 算法时，一般默认相应的决策树为分类树，与回归树有关的内容将在介绍 **CART** 模型时讨论。

在信息熵的基础上进一步引入条件熵 $H(X|Y)$, 表示在随机变量 Y 确定后对随机变量 X 所含信息多少的度量

$$H(X|Y) = - \sum_i^n p(X = x_i, Y = y_i) \ln p(X = x_i | Y = y_i) = - \sum_i^n p(Y = y_i) H(X|Y = y_i)$$

基于信息熵和条件熵, 即可得到随机变量 Y 所带来的信息增益 (information gain) $I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$, $I(X, Y)$ 又称互信息;

3. 显然若样本的某一特征具有较大的信息增益, 意味着以其作为划分依据可以将原样本集划分为信息量较少的多个子样本集, 意味着纯度提升;
4. 假设样本集包含 m 项特征 X_1, \dots, X_m , 且标签 Y 与所有特征 X_i 均为分类数据 (可以是二分类, 也可以是多分类), 则 ID3 算法如下:
 - 确定根节点对应的特征。对于全体样本, 计算标签 Y 的信息熵 $H(Y)$ 和条件熵 $H(Y|X_i)$, 从而计算每一特征 X_i 的信息增益 $I(X_i, Y)$, 选择 $X_k = \arg \max_i I(X_i, Y)$ 作为根节点对应的特征, 基于 X_k 将样本集分为若干个子样本集 (X_k 是几分类变量, 即分为几个子样本集);
 - 确定内部节点对应的特征。对于经根节点划分的样本集, 针对每一子样本集基于相同的方法确定新的内部节点, 进一步细分;
 - 在无额外约束 (深度等) 的情况下, 当所有特征均用于生成根节点或内部节点, 或得到完全纯粹子样本集时, 算法结束, 得到的树状结构即为目标分类树。
5. 可以看到, 基于 ID3 算法的分类树并不一定是二叉树, 且算法具有如下缺点:
 - 算法假设所有特征及标签均为分类数据, 不适用于含连续数据的样本;
 - 算法优先采用信息增益大的特征建立节点, 而在相同条件下分类更多的特征具有更大的信息增益, 造成算法在建立节点时偏好于分类更多的特征;
 - 另外, 算法也无法处理存在缺失值的数据。

C4.5 算法

1. Quinlan 于 1993 年在 ID3 算法的基础上提出 C4.5 算法, 改进了 ID3 算法的主要缺点;
2. C4.5 算法对连续数据进行分类离散化, 为简化运算仅作二元分类。考虑一个由 n 组样本组成的样本集, 存在连续特征 X , 则 X 在样本集中共有 n 个取值, 将其从小到大排列得序列 $\{x_1, \dots, x_n\}$ 。为对序列进行二分类, 算法对相邻两样本点 x_j, x_{j+1} 取平均数 t_j , 从而得到 $n-1$ 个候选划分点。分别计算每一个点作为二元划分点时特征 X 的信息增益, 选择使信息增益最大的 t_j 作为划分点, 从而实现二元划分。与离散特征不同, 若当前结点划分特征为连续特征, 该特征依然参与后代结点的划分;
3. C4.5 算法在信息增益 $I(X, Y)$ 的基础上引入新指标信息增益比 $I_R(X, Y) = \frac{I(X, Y)}{H(X)}$, 将信息熵 $H(X)$ 作为分母可以矫正 ID3 算法在建立节点时偏好于分类更多的特征的缺点。然而若直接以 $I_R(X, Y)$ 替代 $I(X, Y)$ 作为节点生成依据, 会使算法偏好于分类较少的特征, 因此算法采用一个启发式算法: 首先从候选特征中选出 $I(X, Y)$ 高于平均值的特征, 再从中选出 $I_R(X, Y)$ 最大的特征用于生成节点;
4. C4.5 算法进一步设计了针对带缺失值数据的处理规则, 具体地包括以下两个问题:
 - 若样本集中的部分特征存在缺失值, 如何确定相应特征的信息增益等指标。此时, 基于相应特征的不含缺失值的子集计算其信息增益或信息增益比。以 $I(X, Y)$ 为例, 假设 X 为样本集中含缺失值的一个特征、 \tilde{X} 为 X 中不含缺失值的子集, 则有 $I(X, Y) = \rho_X \times I(\tilde{X}, \tilde{Y})$, 其中 ρ_X 为无缺失值样本所占的比例。此时无论特征是否含缺失值, 都可计算其信息增益等指标, 并用于节点划分;
 - 在确定了用于分类的特征后, 若某样本的该特征为缺失值, 如何对其进行分类。此时, 该样本以不同的权重划入所有节点。同样地, 假设当前节点对应的特征为 X 、 \tilde{X} 为 X 中不含缺失值的子集, 且 X 可划分为 k_X 类。对于该特征为缺失值的样本 x , 假设其原本的权重为 w_x , 对于 X 下分的 $\forall i = 1, \dots, k_X$ 类, 样本划入时的权重为 $\rho_i \times w_x$, 其中 ρ_i 为划入第 i 类的无缺失值样本占 \tilde{X} 的比例。
5. 在 C4.5 算法中, 对连续数据采用二分类, 而多分类数据仍采用多分类, 因此算法得到的分类树同样不一定是二叉树。

分类回归树模型 (classification and regression tree, CART)

1. ID3 与 C4.5 算法均是基于信息论的决策树生成算法, 涉及大量对数运算。CART 模型于 1984 年由 Breiman 等人提出, 是另一种广泛应用的决策树模型。与 ID3 与 C4.5 不同, CART 算法不涉及对数运算, 而且算法每次对特征进行切分后只会产生两个子节点, 生成的决策树为二叉树。顾名思义, CART 模型适用于分类和回归任务, 以下将分别进行介绍;
2. 对于 CART 分类树, 算法以基尼系数 (gini index) 替代信息增益。基尼系数代表了模型的不纯度, 越小则不纯度越低, 特征越好。对于给定的样本 Y , 假设其有 K 个类别, 第 k 类的样本为 Y_k , 则样本的基尼系数为

$$\text{Gini}(Y) = 1 - \sum_{k=1}^K \left(\frac{\|Y_k\|}{\|Y\|} \right)^2$$

其中 $\|Y\|$ 表示 Y 的样本量。若基于特征 X 将样本 Y 划分为 Y_1, Y_2 两部分, 则有

$$\text{Gini}(Y, X) = \frac{\|Y_1\|}{\|Y\|} \text{Gini}(Y_1) + \frac{\|Y_2\|}{\|Y\|} \text{Gini}(Y_2)$$

3. 对于 CART 回归树, 每一片叶子所含训练集元素的均值即为该叶子的输出值, 因此尽管是回归问题, 树模型的输出空间依然是离散的。算法以均方误差 (mean square error, MSE) 或平均偏差 (mean absolute error, MAE) 作为节点划分指标;
4. 因为 CART 分类树和回归树都为二叉树, 因此不同于 ID3 与 C4.5 算法, 即使是面对多分类特征 X , 算法也是遍历 X 的所有取值, 选择使得 $\text{Gini}(Y, X)$ 或 $MSE(MAE)$ 最小的取值进行二分类;
5. 在分类问题中, CART 模型与 ID3、C4.5 模型的主要区别在于节点划分指标和划分数目两方面; 而在回归问题中, 仅存在节点划分数目的差异——若生成的决策树为二叉树则为 CART 树, 反之则不是。相比于增加划分数目, 增加二叉树的深度同样可以起到提升模型精度的效果, 因此一般仅由 CART 树模型解决回归问题, 而不考虑 ID3、C4.5 等用于回归的情况, 因为不存在本质差异。

26.1.2 决策树剪枝 (pruning)

1. 由前文可知, 构造决策树时节点划分是以提升训练集样本纯度为目标进行的, 而训练集样本纯度降低也意味着模型对训练集的预测效果得到提升。但若基于某一特征的划分只能提升训练集精度但无法提升测试集精度 (即模型泛化性能), 则相关分支是无意义的。决策树剪枝就是指将决策树的部分无意义的内部节点 (或根节点) 替换为叶子结点的过程, 是降低模型复杂度并避免过拟合的有效手段;
2. 根据剪枝与构造的先后顺序, 剪枝策略可分为预剪枝 (pre-pruning) 和后剪枝 (post-pruning) 两类:
 - 预剪枝与决策树构造同时进行, 可直接生成已经完成剪枝的决策树。此时在确定用于划分的特征后, 将首先比较划分前后模型对测试集的预测效果 (即模型泛化性能), 若测试精度得到提升则进行划分, 否则将该节点标记为叶子结点, 不再进行划分。预剪枝策略降低了模型复杂度和过拟合风险, 也能有效提升模型训练速度, 但也要注意到预剪枝是一种贪心策略——当前划分或许无助于提升测试精度, 但不代表后续划分也是无意义的, 因此策略同时增大了模型欠拟合的风险;
 - 后剪枝则是在决策树构造之后再进行, 首先生成一棵完整的决策树, 再自下而上地对所有非叶子结点进行考察, 若将该节点替换为叶子结点有助于提升模型泛化性能, 则舍去该节点以下的子树。同样地, 也是基于测试集判断模型泛化性能。与预剪枝策略相比, 基于后剪枝得到的决策树往往保留较多的分支、欠拟合程度更低、泛化性能较高, 但时间开销显著增大。

26.2 AdaBoost (Adaptive boosting, 自适应增强)

1. Boosting 是最早成型的集成学习思想, 几乎可应用于所有机器学习算法中并加强原算法的预测精度。Boosting 起源于 Valiant 于 1984 年提出的可能近似正确 (probably approximately correct, PAC) 学习框架, 框架中定义了弱学习和强学习两种概念, 前者是指准确率仅比随机猜测略高的算法, 而后者则具有更高的准确率并能在多项式时间内完成学习。1988 年, Kearns 与 Valiant 提出 PAC 框架中强学习与弱学习等价性的问题——即任意给定一个弱学习器, 能否将其提升为强学习器;

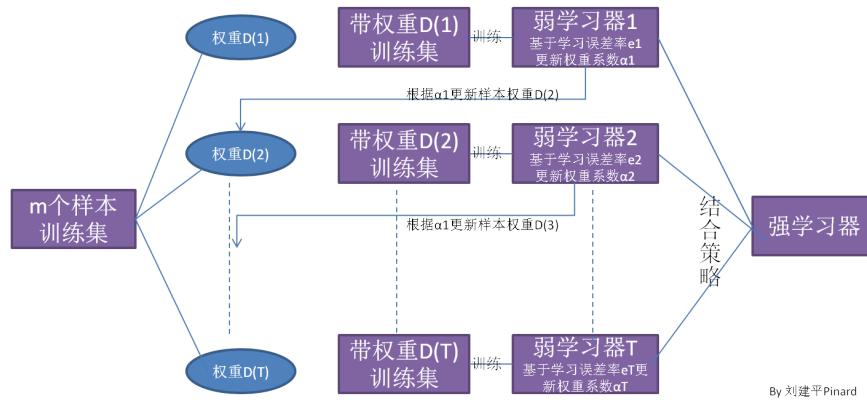


图 26.1 Boosting 算法结构

2. 在这一背景下，1990 年 Schapire 提出最初的 Boosting 算法。但早期的几类 Boosting 算法均要求预知弱学习器学习正确率的下限。1995 年，Freund 和 Schapire 提出 AdaBoost 算法，算法与已有 Boosting 算法具有相同的效率，但不需任何关于弱学习器的先验知识，因而取得巨大的成功，成为最经典的 Boosting 算法。AdaBoost 算法的提出标志着集成学习成为机器学习的重要方法论，在此之后直至深度学习框架流行，集成学习和 SVM 并列为最主流的监督学习算法；
3. AdaBoost 算法通过串行学习的方式生成多个弱学习器，即在第 K 轮学习时已训练得 $K - 1$ 个学习器，并基于前 $K - 1$ 个学习器的预测效果训练第 K 个学习器，最后集成所有基学习器的预测结果得到预测结果。需要说明的是，AdaBoost 是一种元算法框架，并不要求基学习器为决策树；
4. 理解 AdaBoost 算法，核心是理解两个权重的概念：
 - **样本权重**：在 AdaBoost 算法中，所有基学习器都是基于同一训练集生成的。可以预见，如果训练集中的所有样本在每轮训练中保持相同的权重，生成的基学习器也必然相同。因此为了起到集成的效果，在每轮训练后都会基于生成的基学习器的预测效果更新训练集各样本的权重。一般地，前一个学习器预测错误的样本在下一轮训练时权重将被提升，使得生成的基学习器更关注之前学习器预测错误的特征；
 - **学习器权重**：各基学习器关注点不同，预测效果也不同，因此在集成各学习器预测结果时，应赋予预测精度更高的基学习器更高的权重。
5. 假设训练集 X 共包含 n 组样本，记 w_{ij} 为第 i 轮训练时样本 j 的权重、 $D_i = \{w_{i1}, \dots, w_{in}\}$ 为第 i 轮训练时的训练集样本权重的集合、第 i 轮训练得到的基学习器为 $G_i(x)$ 、 $G_i(x)$ 的权重为 α_i 。

26.2.1 AdaBoost 分类

基本思路

1. 考虑二分类问题，不失一般性地记两类的标签为 $\{-1, 1\}$ ；
2. 在第 1 轮训练时，所有样本的权重都相同，即 $w_{1j} = \frac{1}{n}$ ，得到基分类器 $G_1(x)$ ；
3. 基于 D_i ，第 i 轮训练得到分类器 $G_i(x) : X \rightarrow \{-1, 1\}$ ，其权重 α_i 表示为

$$\alpha_i = \frac{1}{2} \ln \frac{1 - e_i}{e_i} \quad e_i = P(G_i(x_j) \neq y_j) = \sum_{j=1}^n w_{ij} I(G_i(x_j) \neq y_j)$$

式中 e_i 定义为分类器 $G_i(x)$ 的分类误差率，即被 $G_i(x)$ 误分类的样本的权重之和。显然 e_i 越小则分类器精度越高、权重 α_i 越大；

4. 基于 $D_i, \alpha_i, G_i(x)$ 更新用于第 $i + 1$ 轮训练的样本权重集合 D_{i+1}

$$w_{i+1,j} = \frac{w_{ij}}{Z_i} \exp\{-\alpha_i y_i G_i(x_j)\} \quad Z_i = \sum_{j=1}^n w_{ij} \exp\{-\alpha_i y_i G_i(x_j)\}$$

式中 Z_i 为规范化因子，使得 D_{i+1} 能成为一概率分布，避免 $e_{i+1} > 1$ 使 α_{i+1} 不可解。可以看到，若 $\alpha_i > 0$ ，即 $e_i < 0.5$ （若明分类器的分类效果优于随机猜测），且 $y_i G_i(x_j) > 0$ （说明分类器对样本 x_j 的分类正确），则 $w_{i+1,j}$ 的权重减小，而被错误分类的样本权重则增大；

- 假设最终生成 m 个基分类器，则将其加权组合得到最终的强分类器 $G(x)$

$$G(x) = \text{sign}\{f(x)\} = \text{sign}\left\{\sum_{i=1}^m \alpha_i G_i(x)\right\}$$

- 为防止分类器过拟合，在实际应用中常引入学习率（learning rate） ν 进行正则化（也称为 shrinkage），此时有

$$f_i(x) = f_{i-1}(x) + \nu \alpha_i G_i(x)$$

损失函数优化

- 在理解了两个权重的概念后，AdaBoost 分类算法的思路并不难理解。上文直接给出 α_i, w_{ij} 的计算式，需要指出的是其并非经验公式，而是由算法的损失函数推导而来；
- AdaBoost 分类算法可以更系统地理解为一个模型为加法模型、损失函数为指数函数、学习算法为前向分步学习算法的二分类学习方法：
 - 所谓加法模型是因为强分类器可表示为若干个基分类器的线性组合 $f(x) = \sum_{i=1}^n \alpha_i G_i(x)$ ；
 - 所谓指数损失函数是指形如 $\exp\{-y f(x)\}$ 的损失函数，可以看到若 $f(x)$ 与 y 同号则损失小于 1，且 $y f(x)$ 越大损失越小；
 - 所谓前向分步学习算法是因为强分类器 $f_{i-1}(x)$ 和 $f_i(x)$ 之间具有如下关系

$$f_i(x) = f_{i-1}(x) + \alpha_i G_i(x)$$

代入损失函数，得到第 i 轮训练时的优化目标为

$$\alpha^*, G^*(x) = \arg \min_{\alpha, G} \sum_{j=1}^n \exp\{-y_j f_i(x_j)\} = \arg \min_{\alpha, G} \sum_{j=1}^n \exp\{-y_j f_{i-1}(x_j) - \alpha y_j G(x_j)\} = \arg \min_{\alpha, G} \sum_{j=1}^n w'_{ij} \exp\{-\alpha y_j G(x_j)\}$$

式中 $w'_{ij} = \exp\{-y_j f_{i-1}(x_j)\}$ 为第 $i-1$ 轮训练时的损失函数于样本 j 的分量，与第 i 轮训练无关。因此在 AdaBoost 算法中每步仅优化一组 $\alpha^*, G^*(x)$ ，逐步优化逐步逼近，即为前向分步学习算法。

基于以上解释，证明 AdaBoost 分类算法中 α_i, w_{ij} 的计算公式；

- 首先观察 w'_{ij} ，注意到 w'_{ij} 越大意味着 $f_{i-1}(x)$ 对样本 j 的预测效果越差，显然 w'_{ij} 的意义与第 i 轮训练时的样本 j 权重 w_{ij} 具有相似性。因为 $w'_{ij} = \exp\{-y_j f_{i-1}(x_j)\}$ ，且 $f_i(x) = f_{i-1}(x) + \alpha_i G_i(x)$ ，得到 w'_{ij} 的递推式

$$w'_{i+1,j} = w'_{ij} \exp\{-\alpha_i y_j G_i(x_j)\}$$

对比 w_{ij} 的递推式可以进一步明确两者的相似性。由数学归纳法可证 $w_{ij} = \frac{w'_{ij}}{\sum_j w'_{ij}}$ ；

- 进一步地化简目标函数从而分离变量 $\alpha, G(x)$

$$\begin{aligned} \sum_{j=1}^n w'_{ij} \exp\{-\alpha y_j G(x_j)\} &= e^{-\alpha} \sum_{j=1}^n w'_{ij} I(y_j = G(x_j)) + e^\alpha \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j)) \\ &= e^{-\alpha} \sum_{j=1}^n w'_{ij} - e^{-\alpha} \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j)) + e^\alpha \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j)) \\ &= e^{-\alpha} \sum_{j=1}^n w'_{ij} + (e^\alpha - e^{-\alpha}) \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j)) \end{aligned}$$

- 计算 α^* ，对上式求 α 的偏导并令其为 0

$$\frac{\partial}{\partial \alpha} \left[e^{-\alpha} \sum_{j=1}^n w'_{ij} + (e^\alpha - e^{-\alpha}) \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j)) \right] = 0$$

$$\begin{aligned}
 & \Rightarrow -e^{-\alpha} \sum_{j=1}^n w'_{ij} + (e^\alpha + e^{-\alpha}) \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j)) = 0 \\
 & \Rightarrow e^\alpha \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j)) = e^{-\alpha} \left(\sum_{j=1}^n w'_{ij} - \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j)) \right) \\
 & \Rightarrow \alpha^* = \frac{1}{2} \ln \frac{\sum_{j=1}^n w'_{ij} - \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j))}{\sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j))} = \frac{1}{2} \ln \frac{1 - e_i}{e_i} \quad e_i = \frac{\sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j))}{\sum_{j=1}^n w'_{ij}} = \sum_{j=1}^n w_{ij} I(y_j \neq G(x_j))
 \end{aligned}$$

显然 α_i 即为最优 α ;

6. 最后计算 $G^*(x)$

$$G^*(x) = \arg \min_G \sum_{j=1}^n w'_{ij} I(y_j \neq G(x_j))$$

基于 w'_{ij} 与 w_{ij} 的关系, 同样可以认为 $G^*(x)$ 是使得误分类样本权重之和最小的分类器, 同样为所求 $G_i(x)$;

7. 以上给出了 AdaBoost 二分类算法细节的理论解释。

26.2.2 AdaBoost 回归

- AdaBoost 算法最初是应用于二分类问题, 为扩展至回归问题衍生出诸多变种算法。在应用于回归问题时仍采用与分类算法相同的基本思想, 仅样本权重、基学习器权重的计算公式和最终基学习器的集成方式存在不同。以下介绍常见的 AdaBoost R2 回归算法;
- 第 i 轮学习得到的弱回归器 $G_i(x)$ 的权重 α_i

$$\alpha_i = \frac{e_i}{1 - e_i} \quad e_i = \sum_{j=1}^n w_{ij} e_{ij}$$

式中 e_i 为回归误差率, 为各样本的相对误差 e_{ij} 的线性组合。样本相对误差有多种计算方式:

- 如果是线性误差, 则 $e_{ij} = |y_j - G_i(x_j)|/E_i$;
- 如果是平方误差, 则 $e_{ij} = [y_j - G_i(x_j)]^2/E_i^2$;
- 如果是指数误差, 则 $e_{ij} = 1 - \exp\{-|y_j - G_i(x_j)|/E_i\}$ 。

其中 $E_i = \max_j |y_j - G_i(x_j)|$ 为训练集各样本的最大误差;

3. 用于第 $i+1$ 轮训练的样本权重 D_{i+1}

$$w_{i+1,j} = \frac{w_{ij} \alpha_j^{1-e_{ij}}}{Z_i} \quad Z_i = \sum_{j=1}^n w_{ij} \alpha_j^{1-e_{ij}}$$

4. 得到 m 个弱回归器后对其进行集成, 但不是通过对所有弱回归器进行线性组合

$$f(x) = \left[\sum_{i=1}^m \left(\ln \frac{1}{\alpha_i} \right) \right] g(x)$$

式中 $g(x) = \alpha_k G_k(x)$, 其中 k 为 $1, \dots, m$ 的中位数。

26.3 梯度提升树 (Gradient boosting decision tree, GBDT)

- 顾名思义, GBDT 是基于决策树模型的梯度提升 (gradient boosting) 算法实例。与 AdaBoost 相似, 梯度提升也是一种元算法, 可支持多种基学习器, 基于梯度提升算法的学习器称为梯度提升机 (gradient boosting machine, GBM);
- 梯度提升与 AdaBoost 同属 Boosting 算法族, 由 Friedman 受 AdaBoost 启发于 2000 年提出, 是 AdaBoost 的推广 (也可以说 AdaBoost 是梯度提升的特例)。前文已说明, 在应用于分类问题时, AdaBoost 本质上是一类基于加法模型和指数损失函数的前向分步学习算法; 在应用于回归问题时设计了不同的权重计算公式, 其核心是采用了不同的损失函数。作为 AdaBoost 的推广, 梯度提升算法支持更多类型的可微损失函数, 极大提升了算法的应用范围 (分类、回归、排序²) 和对特定问题的应用效果。发展至今, 梯度提升

²在有监督学习中, 排序一般指推荐系统的排序任务。任务要求基于训练集计算预测集各样本的推荐值, 并基于推荐值高低向用户进行个性化推荐。

算法已成为目前最主流的 Boosting 框架, 近年来在浅层学习大放异彩的 XGBoost、LightGBM、CatBoost 等都是基于该算法;

3. 梯度提升可视为梯度下降 (gradient descent) 与 boosting 的结合。不同于一般通过更新参数实现梯度下降, 梯度提升算法通过叠加弱学习器 (即 **boosting**) 实现梯度下降;
4. 为方便起见, 以下以回归问题为例介绍 GBDT 算法的基本思想。假设回归问题样本集为 $D_0 = \{(x_1, y_1), \dots, (x_n, y_n)\}$, 所有样本权重相同, 定义误差函数为 L :

- 第 i 轮训练时, 训练第 i 棵回归树 $h_i(x)$, 得到强学习器 $F_i(x) = \sum_{k=1}^i h_k(x) = F_{i-1}(x) + h_i(x)$ 。当考虑学习率 ν 时, 则 $F_i(x) = F_{i-1}(x) + \nu h_i(x)$ 。可以看到, 在组合弱学习器时也不存在学习器的权重差异;
- GBDT 算法训练弱回归树时也不存在样本权重差异, 而是通过改变每轮训练的训练集实现各弱回归树的个性化。记回归树 $h_i(x)$ 的训练集为 $D_i = \{(x_1, r_{i1}), \dots, (x_n, r_{in})\}$, 其中 $r_{ij} = y_j - F_{i-1}(x_j)$ 为学习器 $F_{i-1}(x)$ 对样本 j 预测的残差。可以看到, 每棵回归树 $h_i(x)$ 的目标都是为了修正前 $i-1$ 轮得到的学习器 $F_{i-1}(x)$ 的误差³, 使得 $F_i(x)$ 具有更好的拟合效果, 思路与 AdaBoost 相似。通过以上思路即可拟合样本集 D_0 , 称为提升树 (boosting decision tree, BDT) 算法;
- GBDT 算法与 BDT 的唯一差别在于 r_{ij} , GBDT 算法中 r_{ij} 定义为误差函数在 $F_{i-1}(x_j)$ 处的负梯度

$$r_{ij} = -\frac{\partial L(y_j, F_{i-1}(x_j))}{\partial F_{i-1}(x_j)}$$

此时 r_{ij} 为样本残差的近似值, 称为伪残差, 当且仅当采用均方误差函数时 r_{ij} 才等于残差。因为负梯度是误差函数在 $F_{i-1}(x)$ 处的最速下降方向, 以其作为 $h_i(x)$ 的拟合对象可以起到加速收敛的效果, 这就是梯度下降思想在梯度提升中的体现;

- 综上所述, 可以得到 GBDT 算法中 $F_i(x)$ 与 $F_{i-1}(x)$ 的递推关系, 与梯度下降算法中的参数更新公式无异

$$F_i(x) = F_{i-1}(x) + \nu h_i(x) \simeq F_{i-1}(x) - \nu \cdot \frac{\partial L(y, F_{i-1}(x))}{\partial F_{i-1}(x)}$$

5. 与 AdaBoost 相比, 梯度提升算法避免了经验化的权重设计, 通过支持多种类型的误差函数以更合理直接地应用于多种任务, 以下介绍常用的误差函数:

- **均方差**, 用于回归问题: $L(y, f(x)) = \frac{1}{2}(y - f(x))^2$;
- **绝对损失**, 用于回归问题: $L(y, f(x)) = |y - f(x)|$;
- **Huber 损失**, 用于回归问题, 均方差和绝对损失的结合, 对于远离中心的异常点采用绝对损失, 而中心附近的点采用均方差。预设阈值 δ 一般为某个分位数点

$$L(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & |y - f(x)| \leq \delta \\ \delta \left(|y - f(x)| - \frac{\delta}{2} \right) & |y - f(x)| > \delta \end{cases}$$

需要注意的是, 当采用 Huber 损失时有 $L(y, y) = -0.5\delta^2 \neq 0$, 因此 δ 的取值非常关键, 不宜过大;

- **分位数损失**, 用于分位数回归问题⁴, 其中 θ 为预设分位数

$$L(y, f(x)) = \sum_{y \geq f(x)} \theta |y - f(x)| + \sum_{y < f(x)} (1 - \theta) |y - f(x)|$$

- **指数损失**, 用于分类问题: $L(y, f(x)) = \exp\{-yf(x)\}$;
- **对数损失**, 用于二分类问题: $L(y, f(x)) = \ln(1 + \exp\{-yf(x)\})$;
- **交叉熵损失**, 用于多分类问题, 若 x 属于第 k 类则 $y_k(x) = 1$, $p_k(x)$ 为 x 属于第 k 类的概率

$$L(y, f(x)) = -\sum_{k=1}^K y_k(x) \ln p_k(x) \quad p_k(x) = \frac{\exp\{f_k(x)\}}{\sum_k \exp\{f_k(x)\}}$$

³生成 CART 回归树时常以均方误差作为节点划分标准, 但在 GBDT 算法中, 回归树 $h_i(x)$ 的节点划分标准与 GBDT 算法的误差函数保持一致, 同样为 L 。

⁴分位数回归是一般回归问题的推广, 一般的回归任务关注自变量和因变量均值 (期望) 的关系, 而分位数回归则关注自变量和因变量的特定分位数的关系, 分位数回归还可起到抗噪声的效果 (如用中位数回归替代均值回归)。

26.4 XGBoost (*Extreme gradient boosting*, 极端梯度提升)

1. XGBoost 是陈天奇等人基于传统梯度提升算法开发的开源机器学习项目，最初版本提出于 2014 年。XGBoost 本质上仍属于梯度提升算法，但在算法和工程上做出诸多改进。同样地，XGBoost 支持包括决策树、线性分类器在内的多种基学习器。大量实验证明，XGBoost 是目前浅层学习领域最优秀的算法之一，在特定任务上甚至能与 LSTM 等深度学习模型媲美；
2. AdaBoost、GBDT 和 XGBoost 均基于 boosting 思想串行训练基学习器，每一轮训练的基学习器旨在修正已有学习器的偏差。而为实现这一目标，上述三种算法采取不同的思路：
 - 在 AdaBoost 算法中，每轮训练的目标函数相同、训练集相同，但训练集中样本权重不同，从而使不同的基学习器关注不同的特征。另外基学习器集成时的权重也不同，预测正确率更高的基学习器对最终预测结果的影响越大；
 - 在 GBDT 算法中，每轮训练的目标函数相同、训练集样本权重相同，但训练集不同。每轮训练的训练集被直接设定为已有学习器的伪残差，为已有学习器误差的一阶导数，从而使新的基学习器沿梯度最大方向修正已有学习器的偏差。另外基学习器集成时权重相同；
 - 在 XGBoost 算法中，每轮训练的训练集样本权重相同，但训练集和目标函数均不同。算法支持对样本随机采样生成训练集（借鉴随机森林）增强基学习器泛化性和训练速度。同时算法为每轮训练构造不同的目标函数，不仅考虑了已有学习器误差的一阶导数还引入二阶导数，提升算法的收敛效果。另外基学习器集成时权重相同。
3. 可以看到，XGBoost 算法最大的特点在于训练基学习器时的目标函数设计。定义算法误差函数为 L 、训练集样本数为 n ，假设基学习器 $f(x)$ 为决策树，则目标函数由三部分组成

$$f_i(x) = \arg \min_{f(x)} \left\{ \left[\sum_{j=1}^n L(y_j, F_{i-1}(x_j) + f(x_j)) \right] + \Omega(f(x)) + C \right\} = \arg \min_{f(x)} \{L(y, F_{i-1}(x) + f(x)) + \Omega(f(x)) + C\}$$

4. 目标函数的第一项为误差项，也是最主要的优化目标。XGBoost 算法对其进行泰勒展开，保留至二阶导数项

$$\begin{aligned} L(y_j, F_{i-1}(x_j) + f(x_j)) &\approx L(y_j, F_{i-1}(x_j)) + \frac{\partial L(y_j, F_{i-1}(x_j))}{\partial F_{i-1}(x_j)} f(x_j) + \frac{1}{2} \frac{\partial^2 L(y_j, F_{i-1}(x_j))}{\partial F_{i-1}^2(x_j)} f^2(x_j) \\ &= L(y_j, F_{i-1}(x_j)) + g_{ij}f(x_j) + \frac{1}{2}h_{ij}f^2(x_j) \quad g_{ij} = \frac{\partial L(y_j, F_{i-1}(x_j))}{\partial F_{i-1}(x_j)}, \quad h_{ij} = \frac{\partial^2 L(y_j, F_{i-1}(x_j))}{\partial F_{i-1}^2(x_j)} \end{aligned}$$

5. 目标函数的第二项为正则项，表示决策树模型 $f(x)$ 的复杂度。记 $f(x)$ 的叶子数为 K 、第 k 片叶子的输出值为 c_k 、 γ, λ 为超参，则

$$\Omega(f(x)) = \gamma K + \frac{1}{2} \lambda \sum_{k=1}^K c_k^2$$

6. 目标函数的第三项为常数项。重新组合三项，得到

$$f_i(x) \approx \arg \min_{f(x)} \left\{ \left[\sum_{j=1}^n L(y_j, F_{i-1}(x_j)) + g_{ij}f(x_j) + \frac{1}{2}h_{ij}f^2(x_j) \right] + \left(\gamma K + \frac{1}{2} \lambda \sum_{k=1}^K c_k^2 \right) + C \right\}$$

舍去与优化对象 $f(x)$ 无关的项 $L(y_j, F_{i-1}(x_j)), C$

$$\begin{aligned} f_i(x) &\approx \arg \min_{f(x)} \left\{ \left[\sum_{j=1}^n g_{ij}f(x_j) + \frac{1}{2}h_{ij}f^2(x_j) \right] + \left(\gamma K + \frac{1}{2} \lambda \sum_{k=1}^K c_k^2 \right) \right\} \\ &= \arg \min_{f(x)} \left\{ \sum_{k=1}^K \left[c_k \sum_{j \in J_k} g_{ij} + \frac{1}{2}c_k^2 \left(\lambda + \sum_{j \in J_k} h_{ij} \right) \right] + \gamma K \right\} \\ &= \arg \min_{f(x)} \left\{ \sum_{k=1}^K \left[c_k G_{ik} + \frac{1}{2}c_k^2 (\lambda + H_{ik}) \right] + \gamma K \right\} \quad G_{ik} = \sum_{j \in J_k} g_{ij}, \quad H_{ik} = \sum_{j \in J_k} h_{ij} \end{aligned}$$

上式中 J_k 表示落入第 k 片叶子的样本集合；

7. 假设决策树模型的结构及各内部节点的划分标准确定，则易知叶子结点的输出值 c_k

$$\frac{\partial}{\partial c_k} \left[c_k G_{ik} + \frac{1}{2} c_k^2 (\lambda + H_{ik}) \right] = 0 \implies c_k^* = -\frac{G_{ik}}{\lambda + H_{ik}}$$

将 c_k^* 代入上式，得到目标函数最优值

$$\min_{f(x)} \left\{ \sum_{k=1}^K \left[c_k G_{ik} + \frac{1}{2} c_k^2 (\lambda + H_{ik}) \right] + \gamma K \right\} = \gamma K - \frac{1}{2} \sum_{k=1}^K \frac{G_{ik}^2}{\lambda + H_{ik}}$$

只需遍历所有树结构，使得上式最小的树即为本轮优化的决策树模型。综上所述，XGBoost 算法的核心在于以误差函数二阶泰勒展开为目标函数的决策树 $f_i(x)$ 生成算法。集成过程同 GBDT——考虑学习率 ν ，有 $F_i(x) = F_{i-1}(x) + \nu f_i(x)$ 。

26.4.1 误差函数二阶泰勒展开的意义——与 AdaBoost、GBDT 算法的理论联系

1. XGBoost 算法最大的特点既是将误差函数二阶泰勒展开作为决策树训练的目标函数。本小节将讨论 XGBoost 算法与 AdaBoost 算法和 GBDT 算法的联系；
2. 仅考虑目标函数中与误差函数有关的部分，有

$$\begin{aligned} f_i(x) &= \arg \min_{f(x)} \left\{ \sum_{j=1}^n L(y_j, F_{i-1}(x_j) + f(x_j)) \right\} \simeq \arg \min_{f(x)} \left\{ \sum_{j=1}^n L(y_j, F_{i-1}(x_j)) + g_{ij} f(x_j) + \frac{1}{2} h_{ij} f^2(x_j) \right\} \\ &= \arg \min_{f(x)} \left\{ \sum_{j=1}^n L(y_j, F_{i-1}(x_j)) + \frac{1}{2} h_{ij} \left[f(x_j) + \frac{g_{ij}}{h_{ij}} \right]^2 - \frac{g_{ij}^2}{2h_{ij}} \right\} \end{aligned}$$

而其中 $L(y_j, F_{i-1}(x_j))$, g_{ij} , h_{ij} 均与优化对象 $f(x)$ 无关，则舍去后有

$$f_i(x) \simeq \arg \min_{f(x)} \left\{ \sum_{j=1}^n h_{ij} \left[f(x_j) + \frac{g_{ij}}{h_{ij}} \right]^2 \right\}$$

3. 此时目标函数本质上为一加权最小平方和，类比最小平方误差的概念可以发现，在不考虑正则项的情况下第 i 轮优化本质上是在拟合样本集 $\{(x_j, -g_{ij}/h_{ij})\}$ ，而在 GBDT 算法中第 i 轮的拟合对象为 $\{(x_j, -g_{ij})\}$ 。 h_{ij} 同时也是本轮优化中样本 x_j 的权重，在 AdaBoost 算法中也会在每轮训练时基于已有预测误差更新样本权重，不同之处在于 AdaBoost 采用 $L(y_j, F_{i-1}(x_j))$ 作为样本误差更新公式；
4. 综上所述，尽管 AdaBoost、GBDT、XGBoost 等算法在表面上存在明显差异，但在理论上均可溯源至梯度提升算法。

26.4.2 XGBoost 决策树生成算法

1. 首先介绍决策树的节点划分标准，可基于上式可以得到。考虑当前决策树第 k 片叶子结点的样本集合 J_k ，则该片叶子的不纯度为 $\gamma - \frac{G_{ik}^2}{2(\lambda+H_{ik})}$ ，不纯度越大效果越差。若基于某特征将集合 J_k 划分为 J_{k1}, J_{k2} 两部分，则可计算该特征带来的增益

$$\text{Gain} = \left(\gamma - \frac{1}{2} \frac{G_{ik}^2}{\lambda + H_{ik}} \right) - \left(\gamma - \frac{1}{2} \frac{G_{ik1}^2}{\lambda + H_{ik1}} + \gamma - \frac{1}{2} \frac{G_{ik2}^2}{\lambda + H_{ik2}} \right) = \frac{1}{2} \left(\frac{G_{ik1}^2}{\lambda + H_{ik1}} + \frac{G_{ik2}^2}{\lambda + H_{ik2}} - \frac{G_{ik}^2}{\lambda + H_{ik}} \right) - \gamma$$

使得上述增益 Gain 最大的节点划分方式即为该叶子结点是分裂方式；

2. 在计算划分增益之前，需首先确定节点划分阈值，而阈值选择需依次遍历多个特征的多个取值。XGBoost 支持三种分裂点选择算法：

- *Exact greedy algorithm for split finding*: 算法的遍历过程与经典 CART 模型的遍历过程一致，即依次遍历所有特征的所有取值，并选择使得当前增益最大的划分方式作为节点分裂方式。此时所得结果为最优值，但因需要遍历所有取值效率较低；

- *Approximate algorithm for split finding:* 算法基于加权分位数寻找近似最优分裂点，此时算法不再遍历某个特征的所有取值，而是仅遍历对应于特定分位数的特征取值，遍历开销大大减少。需要注意的是，样本 x_j 对应的分位数 $\text{rank}(x_j)$ 不仅取决于其在集合 J_k 中的大小排序，还应考虑权重 h_{ij} 的影响

$$\text{rank}(x_j) = \frac{\sum_{x \in J_k, x < x_j} h}{\sum_{x \in J_k} h}$$

- *Sparsity-aware split finding:* 旨在处理存在缺失值的数据。若训练集存在缺失值，算法先对非缺失的样本进行排序，对该特征缺失的样本先不处理，然后在遍历每个分裂点时，将这些缺失样本分别划入左子树和右子树来计算损失然后求最优。若预测集存在缺失值，相应样本会被默认分到右子树。

3. 当且达到以下限制时停止决策树生长：

- 树模型达到最大深度（超参 `max_depth`）；
- 叶子结点样本权重和小于特定阈值（超参 `min_child_weight`）；
- 分裂带来的增益小于特定阈值（超参 γ ）。

4. 综上所述，XGBoost 算法中决策树采用 CART 模型，但在叶子结点取值 c_k 、节点划分等算法细节上与经典 CART 模型略有差异。

26.5 LightGBM

LightGBM 是 2016 年微软提出的一种新的 boosting 学习框架。LightGBM 大体采用 XGBoost 框架，但在部分算法细节上进行优化，在保持同等水平准确率的前提下全面改善训练速度和内存占用情况，因此得名“轻型梯度提升机”。本节将主要介绍 LightGBM 相对 XGBoost 的算法优化。

26.5.1 LightGBM 的决策树模型优化

1. 不同于大多数决策树模型或集成学习框架中采用的按层生长的 (**level-wise**) 决策树生成算法，LightGBM 算法创造性地采用带深度限制的按叶子生长的 (**leaf-wise**) 决策树生成算法：
 - 在不考虑预剪枝或因节点样本数过少停止生长的情况下，采用 level-wise 生长策略的决策树每次生长时会同时分裂所有叶子结点，此时叶子结点数 num_leaves 与深度 $depth$ 满足 $\text{num_leaves} = 2^{depth}$ 。
level-wise 策略可避免过拟合，同时方便进行多线程优化，但不加区分地对待所有叶子结点可能带来不必要的开销，即并非所有叶子结点的分裂都可带来可观的增益；
 - LightGBM 采用的 leaf-wise 策略则是每次仅选择所有叶子结点中分裂增益最大的节点进行分裂，此时决策树叶子数与深度之间不再存在直接映射关系。与 level-wise 策略相比，在分裂次数相同的情况下 **leaf-wise** 策略具有更高的精度，缺点是可能生成更深的树，陷入过拟合。因此 LightGBM 对决策树显式地加上最大深度限制 `max_depth`，同时要求 $\text{num_leaves} < 2^{\text{max_depth}}$ 。
2. XGBoost 为加速节点分裂提出基于加权分位数的近似搜索算法，搜索时无需遍历所有样本仅需遍历对应于特定分位数的样本，但仍需保存所有样本的排序。LightGBM 则采用直方图算法 (**histogram algorithm**)，顾名思义，算法对连续数据进行分箱，从而将大规模数据以直方图的形式进行保存，全面改善节点分裂速度与内存占用情况。算法具有如下特点：
 - 算法不再需要对样本数据进行排序，因为只需划定相应的阈值即可获得全部样本的分箱情况，计算效率大大提升；
 - 搜索时仅需遍历相应的直方图所代表的数据，储存时也仅需保存相应的分箱情况，内存占用大大降低；
 - 在节点分裂时可以非常方便地进行直方图加速：已知父节点和任意子节点的直方图，可直接通过作差得到另一子节点的直方图，计算效率大大提升；
 - 连续数据离散化不可避免地损失了部分特征，可能会导致精度下降，但分箱本身又起到正则化效果，合理分箱有助于避免过拟合。

3. LightGBM 的决策树模型支持直接输入类别特征，而无需提前将类别特征转化为 **one-hot** 编码。**one-hot** 编码是机器学习中最常用的对类别特征的编码方式，但在二叉树模型中，**one-hot** 编码会降低树模型对高维分类特征的学习效果。因为二叉树模型在节点划分时采用 **one vs rest** 方式（即每次仅划分一个类别），若特征类别数特别多，会使得划分样本时存在明显的不平衡问题，且导致样本被划分至很多零碎的小空间中。**LightGBM** 的树模型将离散的类别特征嵌入至实数域中，从而有助于实现 **many vs many** 方式的划分（即可单次划分多个类别）：

- 首先基于类别特征对样本进行分箱，每一个箱代表一种类别，并删掉样本量较少的箱；
- 对于第 k 个类别对应的箱，记落入内部的样本集合为 J_k ，按下式计算该类别的嵌入值

$$\frac{G_k}{\lambda + H_k} = \frac{\sum_{j \in J_k} g_{jk}}{\lambda + \sum_{j \in J_k} h_{jk}}$$

上式中 λ 为正则化系数（超参数 `cat_smooth`）、 g_{jk}, h_{jk} 分别为样本 x_j 对应的误差一阶导和二阶导。可以看到上式即为第26.4节中 XGBoost 树模型叶子结点输出值 c_k 的绝对值。基于该嵌入值对所有箱排序；

- LightGBM 定义超参数 `max_cat_to_onehot`（默认为 4），即若特征的类别数不超过 4 类，则仍按 **one vs rest** 方式扫描逐个箱选择最优划分点；反正则采用 **many vs many** 方式进行划分，此时算法分别从左至右和从右至左搜索 `max_num_cat`（默认为 32）个箱（并不搜索所有的分箱）得到最优划分阈值，阈值的两侧即为两个 **many** 集合。

26.5.2 LightGBM 的样本规模压缩

1. 上一小节主要介绍了 LightGBM 对决策树模型所做的优化，其中直方图算法的提出极大地改善了节点划分时的运算效率和内存占用，但在具体计算增益时，仍需遍历每一样本每一特征的误差一阶导数 g_i 和二阶导数 h_i ，对于大规模数据或高维特征的样本集仍然具有较大的计算压力。因此 LightGBM 进一步提出了单边梯度采样 (**Gradient-based One-Side Sampling**, GOSS) 算法与互斥特征绑定 (**Exclusive Feature Bundling**, EFB) 算法，以直接压缩原样本集的规模⁵；

2. 顾名思义，GOSS 是一种样本采样算法，即只采样部分样本用于训练，起到压缩样本量的效果。采样的依据在于权重，尽管 GBDT 算法并未显式地定义样本权重的概念，但第26.4节已经指出样本误差或误差梯度实际上即起到权重的效果，LightGBM 采用样本梯度作为样本权重：

- 对样本按梯度排序，并定义百分数 a, b ；
- 抽取前 a 个大梯度样本，并在之后的 $1 - a$ 个剩余样本中随机抽取 b 个，组成新样本集；
- 因为上述采样过程改变了原样本的分布，因此需放大后续随机采样的 b 个样本的权重，放大倍数为 $\frac{1-a}{b}$ ；
- 将新样本集用于训练，与原样本集相比，数据量大大降低。

3. 不同于 GOSS 算法旨在压缩样本量，EFB 算法通过合并若干个特征压缩样本维度，将稀疏矩阵转化为稠密矩阵。算法需解决两个问题：如何确定参与合并的特征，以及如何实现特征的合并：

- 理想情况下，若两个特征完全互斥（即两个特征不同时取非 0 值），则两个特征即可进行合并而不丢失任何信息，如果两个特征不完全互斥，则称两者存在冲突，定义冲突数 C 为两者同时取非 0 值的次数。若要求只有完全互斥的特征才能进行合并，则特征合并过程可转化为经典的图着色问题 (**graph coloring problem**)⁶，每一特征代表图中的节点，若特征间存在冲突则节点间存在连边，且连边的权重为冲突数。通过定义最大冲突数 C_{\max} ，可以实现不完全互斥的特征的合并，从而进一步压缩特征维度；
- 考虑一个包含 K 个类别的类别特征，通过 one-hot 编码可得到 K 个完全互斥的新特征（取 1 表示属于该特征，取 0 表示不属于），显然 EFB 算法会将其重新合并为一个特征，而且应该能清晰地区分

⁵需要说明的是，直方图算法并非 LightGBM 首创，其它 GBDT 框架发展时已有提出（包括 XGBoost），LightGBM 在效率上最大的优化实际上即为 GOSS 和 EFB 算法。

⁶图着色问题是经典的图论问题，属于 NP 难问题。给定一个无向图 $G = (V, E)$ ，图着色问题即要求将节点集 V 分为 K 个颜色组，每个组形成一个独立集，其中没有相邻的顶点。其优化版本是希望获得最小的 K 值。

K 个类别（取 1 表示属于类别 1，取 2 表示属于类别 2……）。类比这一过程，可以得到 EFB 算法中特征合并的思路——只需确保不同的特征具有不同的取值范围即可保证合并后仍能根据取值区分原本的多个特征。假设特征 A 的取值范围为 $[0, 10]$ 、特征 B 的取值范围为 $[0, 20]$ ，则可为特征 B 的取值加上适当偏置使其取值范围为 $[10, 30]$ 则可与特征 A 合并，形成新的取值范围为 $[0, 30]$ 的特征。此时在采用直方图算法时将不会混淆两类特征的数据。

26.6 CatBoost (*Gradient boosting with categorical features support*)

1. CatBoost 是俄罗斯的搜索巨头 Yandex 于 2017 年开源的机器学习库，与 XGBoost、LightGBM 并称为目前三大 GBDT 实现框架；

26.7 随机森林 (*Random forest*)

第 27 章

贝叶斯机器学习

27.1 马尔可夫链蒙特卡罗方法 (**Markov Chain Monte Carlo, MCMC**)

27.1.1 蒙特卡罗方法 (**Monte Carlo method**)

- 蒙特卡罗方法得名于摩纳哥的蒙特卡罗赌场，又称统计模拟法。方法起源于 1777 年法国数学家布丰 (Comte de Buffon) 的投针实验。在 20 世纪 40 年代中期，由于计算机的发明结合概率统计理论的指导，冯诺依曼和乌拉姆等人正式将其总结为一种数值计算方法。蒙特卡罗方法并非一种具体的算法，而是一类基于随机数进行数值估计的方法的总称；
- 蒙特卡罗方法本质上是一种非参数方法，其核心思想是：对于解析上难以分析的分布，通过产生服从该分布的大量样本对分布的各类性质进行数值估计。方法的理论支撑来源于格里文科定理 (Glivenko-Cantelli theorem) 与大数定律 (law of large numbers)，两者分别指出基于大量样本得到的经验分布函数和样本均值收敛于真实的总体分布函数和总体期望；
- 根据上述描述可知，蒙特卡罗方法本身并不属于贝叶斯统计或机器学习方法，其应用范围也远不止贝叶斯方法，但其出现为贝叶斯方法在计算上带来巨大的便利，因此对贝叶斯方法至关重要。贝叶斯方法的核心是贝叶斯公式

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int_{\theta} p(X|\theta)p(\theta)d\theta}$$

上式中 $p(\theta)$ 为研究对象的先验分布，由研究者假定； $p(X|\theta)$ 为似然，基于样本 X 和先验估计 $p(\theta)$ 计算；而融合了先验估计和似然修正的后验分布 $p(\theta|X)$ 是大量贝叶斯方法的分析基础。可以看到，后验分布 $p(\theta|X)$ 的计算过程非常复杂，即使得到了 $p(\theta|X)$ 的解析形式也可能难以进一步得到其解析性质（如期望、方差等）。蒙特卡罗方法即适用于待研究分布解析形式已知的情况：

- 因为蒙特卡罗方法是通过生成大量服从特定分布的样本对分布性质进行数值估计的方法，故首先讨论蒙特卡洛法关注的一类基础问题：若分布 $p(x)$ 的解析形式已知，如何按概率生成相应的样本 $x_i \sim p(x)$ ：
 - 基于累积概率分布函数采样：适用于 $p(x)$ 的累积概率分布函数 $F(x)$ 可计算的情况。注意到 $F(x)$ 为值域从 0 至 1 的单调不减函数，则服从 $p(x)$ 分布的样本 x_i 可按下式生成

$$x_i = F^{-1}(u_i), \quad u_i \sim U(0, 1)$$

上式中 $U(0, 1)$ 为 $[0, 1]$ 区间上的均匀分布， F^{-1} 为函数 F 的反函数。如果 F^{-1} 的解析形式无法计算，也可按第 23.12 节所述方法数值估计：

- 接受拒绝采样法 (**acceptance-rejection sampling**)：适用于大多数 $p(x)$ 解析形式已知的情况。当 $p(x)$ 的累积概率分布 $F(x)$ 无法计算时，选择任意简单的概率分布 $g(x)$ （可直接采样，如均匀分布、正态分布等）与合适的常数 c ，使得恒有 $c \cdot g(x) \geq p(x)$ ，若样本 x_i 满足以下条件即可认为其服

从 $p(x)$ 分布

$$x_i \sim g(x) \quad \& \quad u_i \leq \frac{p(x_i)}{c \cdot g(x_i)}, \quad u_i \sim U(0, 1)$$

用文字解释上式，即是分别从概率分布 $g(x)$ 与均匀分布 $U(0, 1)$ 采样生成样本 x_i, u_i ，若 $u_i \leq \frac{p(x_i)}{c \cdot g(x_i)}$ 则保留 x_i ，最终保留的集合 $\{x_i\}$ 即服从 $p(x)$ 分布。显然 $g(x), c$ 的选择对算法效率至关重要，若使得接受率过低，则将导致大量无效采样。

上述采样算法统称为**蒙特卡罗采样**，是一类基于易采样分布得到难采样分布样本的算法。基于服从分布 $p(x)$ 的大量样本，即可根据上文所述的格里文科定理与大数定律基于样本信息估计真实总体分布信息；

5. 接受拒绝采样法是一种在理论上非常简洁巧妙的算法，但在实际应用中既要保证构造的 $c \cdot g(x) \geq p(x)$ 恒成立，又要使得拒绝率尽可能小，仍要求研究者具备相当的数学分析功底。为此引出**蒙特卡罗方法除采样之外的另一类基本应用：无需生成严格服从 $p(x)$ 分布的样本，直接基于任意易采样分布样本估计 $p(x)$ 的性质**。注意到概率分布 $p(x)$ 的大量性质如期望、方差等均可统一建模为关于函数 $f(x)$ 的定积分问题，故以定积分问题为例

$$\theta = \int_a^b f(x) dx$$

当 $f(x)$ 解析形式已知但定积分难以计算时，蒙特卡罗法即是一种可行的数值估计方法：

- **随机投点法：**实际上即是布丰投针实验估计圆周率 π 的办法。当 a, b 为实数且 $f(x)$ 在区间 $[a, b]$ 内有界时，则可通过在二维平面内随机均匀投点估计 θ 。算法非常简单，但效率较低，且不适用于 a, b 取 ∞ 或 $f(x)$ 无界的情况；
- **平均值法（重要性采样）：**适用于 a, b 可能取 ∞ 或 $f(x)$ 可能无界的情况。将定积分可写为级数形式

$$\theta = \int_a^b f(x) dx = \lim_{n \rightarrow \infty} \frac{b-a}{n} \sum_{i=1}^n f(x_i), \quad x_i = a + \frac{(b-a)i}{n}$$

上式中 $\{x_i\}$ 即为 $[a, b]$ 间均匀分布的大量样本。故当 a, b 为实数时， θ 可直接基于均匀分布样本估计

$$\theta = \int_a^b f(x) dx \approx \frac{b-a}{n} \sum_{i=1}^n f(x_i), \quad x_i \sim U(a, b)$$

然而当 a, b 可能取 ∞ 时，因为均匀分布无法生成无穷区间内的样本，故上式无法直接应用。假设概率密度函数 $q(x)$ 定义于无穷区间，则又注意到

$$\theta = \int_a^b f(x) dx = \int_a^b \frac{f(x)}{q(x)} q(x) dx = \mathbb{E}_{q(x)}[X], \quad X_i = \frac{f(x_i)}{q(x_i)}, \quad x_i \sim q(x)$$

可知 $f(x)$ 在 $[a, b]$ 上的定积分等价于随机变量 X 在概率分布 $q(x)$ 上的期望，因此 θ 可基于服从 $q(x)$ 分布的样本估计

$$\theta = \int_a^b f(x) dx = \int_a^b \frac{f(x)}{q(x)} q(x) dx \approx \frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{q(x_i)}, \quad x_i \sim q(x)$$

27.1.2 马尔可夫链采样

1. 上文已指出，蒙特卡罗方法是一类基于易采样分布的大量样本进行数值估计的过程。方法一方面要求构造的分布足够简单以方便采样，另一方面又要求分布具有相当近似性以使得样本可覆盖待研究问题的多种场景。然而对于高维空间问题，往往难以构造满足需求的即近似又简单易采样的分布，由此即引出**马尔可夫链模型**；
2. 马尔可夫链是一种经典的随机过程模型。模型假设系统某一时刻状态转移的概率只依赖于其前一个状态，即 $P(X_{t+1}|X_0, \dots, X_t) = P(X_{t+1}|X_t)$ 。因此，若系统任意两种状态间相互转移的概率 P_{ij} 已知，构造概率转移矩阵 $P = [P_{ij}]$ ，则系统任意 t 时刻的状态可简单计算得 $X_t = P^t X_0$ ；

3. 之所以引入马尔可夫链模型可便利高维空间下的蒙特卡罗方法，是因为马尔可夫链的收敛性质——如果一个非周期的马尔可夫链有状态转移矩阵 P ，且系统任意两状态连通，则 $\lim_{n \rightarrow \infty} P_{ij}^n$ 与 i 无关，具体地

$$\lim_{n \rightarrow \infty} P_{ij}^n = \pi_j, \quad \lim_{n \rightarrow \infty} P^n = \begin{bmatrix} \pi_1 & \cdots & \pi_j & \cdots \\ \vdots & \ddots & \vdots & \vdots \\ \pi_1 & \cdots & \pi_j & \cdots \end{bmatrix}, \quad \pi_j = \sum_{i=0}^{\infty} \pi_i P_{ij}$$

式中 P_{ij} 严格写作 $P(j \rightarrow i)$ ；行向量 π 被称为马尔可夫链的平稳分布 (**stationary distribution**)，是方程 $\pi P = \pi$ 的唯一非负解。上述性质适用于状态数有限或无限场景，但要求马尔可夫链非周期且各状态连通。其中马尔可夫链非周期的数学判据为对于任意状态 i ，令 d 为集合 $\{n | n \geq 1, P_{ii}^n > 0\}$ 的最大公约数，若 $d = 1$ 则为非周期。而各状态连通则是指任意一个状态可以通过有限步到达其他所有状态，即 P^n 中任意元素均大于 0；

4. 马尔可夫链的收敛性质为蒙特卡罗方法中的特定分布样本采样方法提供了新的思路。记 n 步后状态转移矩阵 P 收敛至平稳分布 π 。对于高维特征样本 X_0 ，基于状态转移矩阵 P 依次得到新样本 X_1, X_2, \dots ，则样本 X_n, X_{n+1}, \dots 即为同属于平稳分布 π 的样本，上述过程即被称为马尔可夫链采样。**基于马尔可夫链采样生成服从任意分布的样本，关键在于反推相应的概率转移矩阵 P ；**
5. 为解决上述问题，进一步介绍马尔可夫链的细致平稳条件 (**detailed balance condition**)——若非周期马尔可夫过程对应的状态转移概率矩阵 P 与概率分布 π 满足

$$\pi_i P_{ij} = \pi_j P_{ji}, \quad \forall i, j$$

则 π 为相应马尔可夫过程的平稳分布。对于二维概率转移矩阵 P ，细致平稳条件是其存在平稳分布的充要条件；对于更高维概率转移矩阵，细致平稳条件仅是其存在平稳分布的充分条件。因此，**欲基于马尔可夫链采样生成服从任意分布 π 的样本，可通过构造满足细致平稳条件的概率矩阵 P 得到以 π 为平稳分布的马尔可夫过程；**

6. 综上所述，马尔可夫链采样过程关键分为以下两步：1) 基于任意目标分布 $\pi(x)$ 构造满足细致平稳条件的概率转移矩阵 P ；2) 基于上一步采样 x_t 与条件概率 $P(x|x_t)$ 采样得到新样本 x_{t+1} 。连续多次抽样后的样本即可逐渐收敛于目标分布 $\pi(x)$ 。

27.1.3 MCMC 采样与估计

- 顾名思义，马尔可夫链蒙特卡洛采样 (MCMC 采样) 是马尔可夫链采样方法与蒙特卡罗采样方法的组合。具体地，**MCMC 采样方法是一种基于蒙特卡罗方法改进的马尔可夫链采样方法；**
- 上文已指出，马尔可夫链采样通过马尔可夫过程克服了蒙特卡罗采样方法难以直接应用于高维分布采样的问题。在上述过程中，可能存在概率转移矩阵 P 难以构造的问题，且构造得到的 P 也可能不易于采样，特别是状态连续的情况。为此**引入蒙特卡罗方法的基本思想——基于易采样但不符合细致平稳条件的概率转移矩阵 Q 采样得到符合细致平稳条件但不易采样的概率转移矩阵 P 的样本；**
- 类比蒙特卡罗方法中的接受拒绝采样法，若易采样的概率转移矩阵 Q 不满足细致平稳条件 $\pi(i)Q(j \rightarrow i) \neq \pi(j)Q(i \rightarrow j)$ ，则不妨引入接受率 $\alpha(j \rightarrow i) \in [0, 1]$ 使得细致平稳条件成立

$$\pi(i)Q(j \rightarrow i)\alpha(j \rightarrow i) = \pi(j)Q(i \rightarrow j)\alpha(i \rightarrow j)$$

令 $\alpha(j \rightarrow i) = \pi(j)Q(i \rightarrow j)$ ，从而得到满足细致平稳条件的概率转移矩阵 $P(j \rightarrow i) = Q(j \rightarrow i)\alpha(j \rightarrow i)$ 。进而得到最基本的 MCMC 采样方法：

Algorithm 27.1 基本 MCMC 采样算法

输入：任意易采样马尔可夫状态转移概率矩阵 Q ，目标分布 $\pi(x)$ ，状态转移次数阈值 n_1 ，需要的样本数 n_2 ，初始化样本 x_0

- 1: **for** $t = 0, \dots, n_1 + n_2 - 1$ **do**
- 2: 从状态转移概率矩阵 Q 采样 $x^* \sim Q(x_t \rightarrow x)$

```

3: 从 0-1 均匀分布采样  $u \sim U(0, 1)$ 
4: 定义接受率  $\alpha(x^* \rightarrow x_t) = \pi(x^*)Q(x_t \rightarrow x^*)$ 
5: if  $u < \alpha(x^* \rightarrow x_t)$  then
6:   接受  $x^*$ ,  $x_t = x^*$ 
7: else
8:   拒绝  $x^*$ , 重新采样  $t = \max\{t - 1, 0\}$ 
9: end if
10: end for
11: 最后  $n_2$  个采样样本即为服从目标分布  $\pi(x)$  的样本

```

算法在理论层面上实现了基于任意马尔可夫过程采样得到目标马尔可夫过程的目标，但在实际应用中并不常用，因为按定义构造的接受率 $\alpha(j \rightarrow i) = \pi(j)Q(i \rightarrow j)$ 往往偏低，导致执行时存在频繁拒绝的情况；

4. 为解决 MCMC 采样时接受率可能过低的问题，一个自然的思路为等比例放大接受率 $\alpha(j \rightarrow i)$ 使得细致平稳条件依然被满足。记放大系数为 K ，放大后的接受率为 $\hat{\alpha}(j \rightarrow i)$ ，显然放大的极限为 $\hat{\alpha}(j \rightarrow i) = 1$ ，此时 $K = \frac{1}{\alpha(j \rightarrow i)} = \frac{1}{\pi(j)Q(i \rightarrow j)}$ 。为满足细致平稳条件

$$\hat{\alpha}(i \rightarrow j) = K\alpha(i \rightarrow j) = \frac{\pi(i)Q(j \rightarrow i)}{\pi(j)Q(i \rightarrow j)} \leq 1$$

综上可得到等比例放大后的接受率 $\hat{\alpha}(j \rightarrow i)$ 定义式 $\hat{\alpha}(j \rightarrow i) = \min\left\{\frac{\pi(j)Q(i \rightarrow j)}{\pi(i)Q(j \rightarrow i)}, 1\right\}$ 。将其替换原本 MCMC 算法中接受率的定义式，即得到经典的 Metropolis-Hastings 算法（又称 M-H 算法），由 Metropolis 提出并被 Hastings 改进

Algorithm 27.2 Metropolis-Hastings 采样算法

输入：任意易采样马尔可夫状态转移概率矩阵 Q ，目标分布 $\pi(x)$ ，状态转移次数阈值 n_1 ，需要的样本数 n_2 ，初始化样本 x_0

```

1: for  $t = 0, \dots, n_1 + n_2 - 1$  do
2:   从状态转移概率矩阵  $Q$  采样  $x^* \sim Q(x_t \rightarrow x)$ 
3:   从 0-1 均匀分布采样  $u \sim U(0, 1)$ 
4:   定义接受率  $\alpha(x^* \rightarrow x_t) = \min\left\{\frac{\pi(x^*)Q(x_t \rightarrow x^*)}{\pi(x_t)Q(x^* \rightarrow x_t)}, 1\right\}$ 
5:   if  $u < \alpha(x^* \rightarrow x_t)$  then
6:     接受  $x^*$ ,  $x_t = x^*$ 
7:   else
8:     拒绝  $x^*$ , 重新采样  $t = \max\{t - 1, 0\}$ 
9:   end if
10: end for
11: 最后  $n_2$  个采样样本即为服从目标分布  $\pi(x)$  的样本

```

M-H 采样算法作为最经典的 MCMC 算法之一，现阶段更多应用于低维分布采样。当分布维数过高时，一方面接受率 $\alpha(j \rightarrow i)$ 的计算时间增大，且无法避免拒绝采样的情况；另一方面目标分布的联合概率分布 $\pi(i)$ 的确定难度增大，仅能获得各维度特征的条件概率分布，此时算法将不适用；

5. 为避免 MCMC 采样时拒绝采样的情况，要求接受率 α 恒为 1，又考虑到接受率 α 的引入是为了解决细致平稳条件一般不成立的情况，故而需针对性地设计马尔可夫状态转移概率矩阵 Q ，既易于采样且细致平稳条件成立。先讨论二维分布的场景，记二维随机变量 $x = (u_1, u_2)$ ，联合概率分布为 $\pi(x)$ ，条件概率分布为 $\pi(u_2|u_1)$ （或 $\pi(u_1|u_2)$ ）。观察第一个特征维度相等的两个样本 $x^* = (u_1^*, u_2^*)$, $x' = (u_1^*, u_2')$ ，显然有

$$\begin{cases} \pi(x^*) \cdot \pi(u'_2|u_1^*) = P(u_1 = u_1^*) \cdot \pi(u_2^*|u_1^*) \cdot \pi(u'_2|u_1^*) \\ \pi(x') \cdot \pi(u_2^*|u_1^*) = P(u_1 = u_1^*) \cdot \pi(u'_2|u_1^*) \cdot \pi(u_2^*|u_1^*) \end{cases} \implies \pi(x^*) \cdot \pi(u'_2|u_1^*) = \pi(x') \cdot \pi(u_2^*|u_1^*)$$

上式指出，若令转移概率 $Q(x^* \rightarrow x) = \pi(u_2|u_1^*)$ ，则直线 $u_1 = u_1^*$ 上任意两点的转移均满足细致平稳条件；同理，若令转移概率 $Q(x^* \rightarrow x) = \pi(u_1|u_2^*)$ ，则直线 $u_2 = u_2^*$ 上任意两点的转移也满足细致平稳条件。综上

即可构造如下适用于二维分布的马尔可夫链状态转移概率矩阵 Q 使得细致平稳条件成立

$$Q(x_i \rightarrow x_j) = \begin{cases} \pi(u_{2j}|u_{1i}) & \text{if } u_{1j} = u_{1i} \\ \pi(u_{1j}|u_{2i}) & \text{if } u_{2j} = u_{2i} \\ 0 & \text{else} \end{cases}$$

可以看到，按上式构造马尔可夫过程进行采样，一方面无需引入接受率从而提升采样效率，另一方面仅需目标平稳分布的条件分布而无需联合分布，从而降低计算与采样难度。以上算法即为二维 Gibbs 采样算法，是现阶段最主流的 MCMC 采样算法。将算法扩展至更高维度场景，得到一般的 Gibbs 采样如下

Algorithm 27.3 N 维 Gibbs 采样算法 (N 不小于 2)

输入：目标分布条件概率分布 $\pi(u_i|u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_N), \forall i = 1, \dots, N$ ，状态转移次数阈值 n_1 ，需要的样本数 n_2 ，初始化样本 $x^{(0)} = (u_1^{(0)}, \dots, u_N^{(0)})$

- 1: **for** $t = 0, \dots, n_1 + n_2 - 1$ **do**
- 2: **for** $i = 1, \dots, N$ **do**
- 3: 从条件概率分布 $\pi(u_i|u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_N)$ 采样第 i 维特征取值 $u_i^{(t+1)} \sim \pi(u_i|u_1^{(t+1)}, \dots, u_{i-1}^{(t+1)}, u_{i+1}^{(t)}, \dots, u_N^{(t)})$
- 4: **end for**
- 5: 得到第 $t+1$ 个采样样本 $x^{(t+1)} = (u_1^{(t+1)}, \dots, u_N^{(t+1)})$
- 6: **end for**
- 7: 最后 n_2 个采样样本即为服从目标分布 $\pi(x)$ 的样本

6. 基于上述采样算法可得到服从分布 $\pi(x)$ 的大量样本 $\{x_i\}$ 。基于该样本即可对随机变量 x 进行点估计与区间估计。然而某些问题只需得到点估计结果，基于上述 MCMC 采样算法求解此类问题意味着需要额外生成大量样本，造成计算资源浪费。迭代条件峰值 (iterated conditional modes, ICM) 算法适用于解决此类问题。算法名字中的 mode 并非指“模式”而是指“众数”。顾名思义，算法通过迭代最大化条件概率实现随机变量估计，本质上是通过坐标上升法 (coordinate ascent) 实现极大后验估计，与 Gibbs 算法非常相似

Algorithm 27.4 ICM (iterated conditional modes) 算法

输入：目标分布条件概率分布 $\pi(u_i|u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_N), \forall i = 1, \dots, N$ ， $t = 0$ ，初始化样本 $x^{(0)} = (u_1^{(0)}, \dots, u_N^{(0)})$

- 1: **repeat**
- 2: **for** $i = 1, \dots, N$ **do**
- 3: 极大条件概率分布 $\pi(u_i|u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_N)$ 估计第 i 维特征取值 $u_i^{(t+1)} \leftarrow \arg \max_{u_i} \pi(u_i|u_1^{(t+1)}, \dots, u_{i-1}^{(t+1)}, u_{i+1}^{(t)}, \dots, u_N^{(t)})$
- 4: **end for**
- 5: $x^{(t+1)} = (u_1^{(t+1)}, \dots, u_N^{(t+1)})$
- 6: $t \leftarrow t + 1$
- 7: **until** $x^{(t)}$ 收敛

27.2 贝叶斯变分推断 (Variational Bayesian Inference)

27.2.1 问题建模

1. 贝叶斯推断的核心是后验分布的估计，由此可大体将贝叶斯推断方法分为精确推断与近似推断两大类。精确推断方法要求严格基于贝叶斯公式推导后验分布解析式进而进行各类估计任务，计算开销较大；而近似推断方法则旨在对后验分布进行逼近，在现代贝叶斯推断框架中更为常用；
2. 贝叶斯近似推断框架又可大致分为两类。一类是以 Metropolis-Hastings 采样、Gibbs 采样为代表的 MCMC 方法（第 27.1 节），具有计算精度高、计算效率低的特点。在计算资源无约束的情况下，MCMC 方法最终可以得到完全收敛于后验分布的样本。另一类则是变分推断 (variational inference, VI) 方法，可作为 MCMC 算法的高效替代，但精度存在上限；
3. 不同于 MCMC 方法通过构造马尔可夫过程拟合目标分布，变分推断方法的基本思路是通过已知的简单分布逼近目标分布，其中用于逼近的简单分布被称为变分分布 (variational distribution)，“变分推断”由此得名；

4. 记后验分布为 $P(Z|X)$, 变分分布为 $Q(Z)$ 。令 KL 散度 (见第 23.9.6 节) 量化分布间的相似度, 有

$$\begin{aligned} D_{KL}(Q(Z)|P(Z|X)) &= \sum_Z Q(Z) \ln \frac{Q(Z)}{P(Z|X)} = \sum_Z Q(Z) [\ln Q(Z) - \ln P(Z, X)] + \ln P(X) \sum_Z Q(Z) \\ &= \sum_Z Q(Z) [\ln Q(Z) - \ln P(Z, X)] + \ln P(X) \\ \implies \ln P(X) &= D_{KL}(Q(Z)|P(Z|X)) + \sum_Z Q(Z) [\ln P(Z, X) - \ln Q(Z)] \\ \implies \ln P(X) &\geq \sum_Z Q(Z) [\ln P(Z, X) - \ln Q(Z)] \end{aligned}$$

上式中 X 为观测值; $P(X) = \sum_Z P(X|Z)P(Z)$ 与 $Q(Z)$ 无关, 被称为 evidence; 而 $\sum_Z Q(Z) [\ln P(Z, X) - \ln Q(Z)]$ 作为 $\ln P(X)$ 的下界, 又被称为 ELBO (evidence lower boundary)。欲构造 $Q(Z)$ 逼近 $P(Z|X)$, 有

$$\arg \min_Q D_{KL}(Q(Z)|P(Z|X)) \iff \arg \max_Q \sum_Z Q(Z) [\ln P(Z, X) - \ln Q(Z)]$$

综上, 变分推断被建模为最大化 ELBO 的优化问题, 而优化的结果取决于变分分布 $Q(Z)$ 的形式。

27.2.2 坐标上升变分推断 (coordinate ascent variational inference, CAVI)

1. 一种简单而有效的变分分布族为平均场变分族。此类变分族基于“平均场 (mean field)”理论¹, 假设 Z 的各分量 Z_1, \dots, Z_m 间相互独立, 即 $Q(Z) = \prod_j Q_j(Z_j)$ 。将 ELBO 改写为关于 Z 的分量的形式, 注意到

$$\begin{aligned} \sum_Z f(Z) &= \sum_{Z_1} \sum_{Z_2} \cdots \sum_{Z_m} f(Z) \\ \sum_Z Q(Z)f(Z_j) &= \sum_{Z_j} Q_j(Z_j)f(Z_j) \left(\sum_{Z_{\neq j}} Q_{\neq j}(Z_{\neq j}) \right) = \sum_{Z_j} Q_j(Z_j) \cdot f(Z_j) \\ \sum_Z Q(Z)f(Z) &= \sum_{Z_j} Q_j(Z_j) \left(\sum_{Z_{\neq j}} Q_{\neq j}(Z_{\neq j})f(Z) \right) = \sum_{Z_j} Q_j(Z_j) \cdot \mathbb{E}_{Q_{\neq j}}[f(Z)] \end{aligned}$$

则 ELBO 改写为

$$ELBO = \sum_Z Q(Z) [\ln P(Z, X) - \ln Q(Z)] = \sum_{Z_j} Q_j(Z_j) \cdot \mathbb{E}_{Q_{\neq j}}[\ln P(Z, X)] - \sum_{Z_j} Q_j(Z_j) \ln Q_j(Z_j) + \text{const}$$

其中 const 表示与 Z_j 无关的项。注意到上式中 $\mathbb{E}_{Q_{\neq j}}[\ln P(Z, X)]$ 是对除 Z_j 以外的其它分量求期望, 故其结果为 Z_j 的表达式。令一个新的分布 $\ln \hat{P}_j(Z_j, X) = \mathbb{E}_{Q_{\neq j}}[\ln P(Z, X)] + \text{const}$, 则

$$ELBO = - \sum_{Z_j} Q_j(Z_j) \frac{\ln Q_j(Z_j)}{\ln \hat{P}_j(Z_j, X)} + \text{const} = -D_{KL}\left(Q_j(Z_j) \mid \hat{P}_j(Z_j, X)\right) + \text{const}$$

上式被称为 ELBO 的平均场近似 (mean field approximation)。

2. 在将 ELBO 改写为关于 Z 的分量的形式后即可基于坐标上升算法 (详见第 17.7 节) 进行优化。显然有

$$\arg \max_{Z_j} ELBO \implies \arg \min_{Z_j} D_{KL}\left(Q_j(Z_j) \mid \hat{P}_j(Z_j, X)\right) \implies Q_j^*(Z_j) = \hat{P}_j(Z_j, X) \propto \exp\{\mathbb{E}_{Q_{\neq j}}[\ln P(Z, X)]\}$$

上式即为坐标上升变分推断中每一次沿坐标轴方向的更新公式;

3. 坐标上升变分推断算法基于各分量独立的平均变分族近似 Z 的后验分布, 看似存在假设较强的问题, 但在实际应用中算法具有较好的扩展性, 可通过将坐标上升算法改为块坐标上升算法解决参数 Z 中部分分量存在相关性的问题。

¹ 平均场理论又称平均场近似, 最早提出于统计物理学领域, 如今在多个领域都有应用, 其本质是一种将多体问题分解为单体问题的近似方法。在多体问题, 不同个体会相互影响, 从而极大地增加了分析的复杂性。此时当个体数量非常多时, 可首先研究多个个体相互作用所形成的场, 即为平均场, 再对处于平均场中的单个物体进行分析, 从而降低问题复杂性。举例分析, 在鱼群中, 每条鱼的行为决策都会受周围鱼行为的影响, 同时也会影响周围鱼的行为决策, 此时在一个大规模鱼群中分析单条鱼的行为将非常困难, 因此可以首先分析鱼群整体的行为, 再根据单条鱼在鱼群中的位置估算其行为。

27.2.3 黑盒变分推断 (black box variational inference, BBVI)

- 为最大化 ELBO，坐标上升变分推断算法假设了变分分布 $Q(Z)$ 的特征，不可避免地导致算法无法准确拟合某些形式的后验分布。2014 年 Rajesh Ranganath 等人提出了黑盒变分推断 (black box variational inference, BBVI) 以解决上述问题²；
- 顾名思义，算法无需额外假定变分分布 $Q(Z)$ 的任何特征，对于任意形式的变分分布 $Q(Z)$ ，记其超参为 θ ，算法通过优化 θ 最大化 ELBO

$$\begin{aligned}\frac{\partial}{\partial \theta} \text{ELBO} &= \frac{\partial}{\partial \theta} \left(\sum_z Q(Z|\theta) [\ln P(Z, X) - \ln Q(Z|\theta)] \right) \\ &= \sum_z \frac{\partial}{\partial \theta} Q(Z|\theta) \cdot [\ln P(Z, X) - \ln Q(Z|\theta)] + \sum_z Q(Z|\theta) \cdot \frac{\partial}{\partial \theta} [\ln P(Z, X) - \ln Q(Z|\theta)] \\ &= \sum_z \frac{\partial}{\partial \theta} Q(Z|\theta) \cdot [\ln P(Z, X) - \ln Q(Z|\theta)] - \sum_z Q(Z|\theta) \cdot \frac{1}{Q(Z|\theta)} \cdot \frac{\partial}{\partial \theta} Q(Z|\theta) \\ &= \sum_z \frac{\partial}{\partial \theta} Q(Z|\theta) \cdot [\ln P(Z, X) - \ln Q(Z|\theta)] - \frac{\partial}{\partial \theta} \left(\sum_z Q(Z|\theta) \right) \\ &= \sum_z \frac{\partial}{\partial \theta} Q(Z|\theta) \cdot [\ln P(Z, X) - \ln Q(Z|\theta)] = \mathbb{E}_{Q(Z|\theta)} \left[\frac{\partial}{\partial \theta} \ln Q(Z|\theta) \cdot [\ln P(Z, X) - \ln Q(Z|\theta)] \right]\end{aligned}$$

上式即为 ELBO 关于 θ 导数的解析形式。但因为分布 $P(X, Z), Q(Z|\theta)$ 的概率密度函数形式可能较为复杂，且为了避免穷举 Z ，上式可通过蒙特卡罗方法（见第 27.1 节）进一步近似如下

$$\frac{\partial}{\partial \theta} \text{ELBO} \simeq \frac{1}{S} \sum_{s=1}^S \frac{\partial}{\partial \theta} \ln Q(Z^{(s)}|\theta) \cdot [\ln P(Z^{(s)}, X) - \ln Q(Z^{(s)}|\theta)], \quad Z^{(s)} \sim Q(Z|\theta)$$

上式中 $\{Z^{(s)}|s=1, \dots, S\}$ 为从变分分布 $Q(Z|\theta)$ 采样的样本。基于上述梯度信息，即可基于梯度上升法迭代更新 θ 以最大化 ELBO

$$\theta_{t+1} \leftarrow \theta_t + \rho_t \left[\frac{1}{S} \sum_{s=1}^S \frac{\partial}{\partial \theta} \ln Q(Z_t^{(s)}|\theta_t) \cdot [\ln P(Z_t^{(s)}, X) - \ln Q(Z_t^{(s)}|\theta_t)] \right], \quad \text{s.t. } \sum_t^\infty \rho_t = \infty, \quad \sum_t^\infty \rho_t^2 < \infty$$

因为梯度信息是由有限样本近似估计的，故上述梯度上升过程为随机梯度上升 (stochastic gradient ascent)，基于上述过程进行变分推断即为最基本的黑盒变分推断算法；

- 随机梯度上升中的梯度信息是由有限样本估计的，不可避免引入额外噪声，影响算法收敛速度。为此包括原文在内的大量研究尝试控制梯度估计时产生的方差以提升收敛速度。其中一类改进思路是引入在深度学习中广泛应用的重参数化技巧 (reparameterization trick)，方法可为无梯度的采样行为中额外引入梯度信息，从而增加估计的确定性。重参数化的基本思想在于将随机的采样行为 $x \sim p(x|\theta)$ 转化为确定性计算过程 $x = f(\varepsilon, \theta)$ ，而将原本采样过程的随机性转移至函数 f 的自变量 ε 中， ε 为随机变量 $\varepsilon \sim q(\varepsilon)$ 。基于重参数化技巧重新推导 ELBO 对 θ 的导数有

$$\begin{aligned}\frac{\partial}{\partial \theta} \text{ELBO} &= \frac{\partial}{\partial \theta} \left(\sum_z Q(Z|\theta) [\ln P(Z, X) - \ln Q(Z|\theta)] \right) \\ &= \sum_\varepsilon q(\varepsilon) \cdot \frac{\partial}{\partial \theta} [\ln P(Z, X) - \ln Q(Z|\theta)] \\ &= \sum_\varepsilon q(\varepsilon) \cdot \frac{\partial}{\partial Z} [\ln P(Z, X) - \ln Q(Z|\theta)] \cdot \frac{\partial}{\partial \theta} f(\varepsilon, \theta), \quad Z = f(\varepsilon, \theta), \quad \varepsilon \sim q(\varepsilon)\end{aligned}$$

再基于蒙特卡罗方法估计上式有

$$\frac{\partial}{\partial \theta} \text{ELBO} \simeq \frac{1}{S} \sum_{s=1}^S \frac{\partial}{\partial Z} [\ln P(Z^{(s)}, X) - \ln Q(Z^{(s)}|\theta)] \cdot \frac{\partial}{\partial \theta} f(\varepsilon^{(s)}, \theta), \quad Z^{(s)} = f(\varepsilon^{(s)}, \theta), \quad \varepsilon^{(s)} \sim q(\varepsilon)$$

²Rajesh Ranganath, Sean Gerrish, David Blei. Black Box Variational Inference. Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, PMLR 33:814-822, 2014. <https://proceedings.mlr.press/v33/ranganath14.html>

这一简单的技巧可以使得从一个噪声分布中进行采样从而获得对梯度的无偏随机估计。更重要的是，在连续变量模型中方法比其他估计方法具有更小的方差。

27.3 朴素贝叶斯模型 (Naive Bayesian Model, NBM)

- 朴素贝叶斯算法是贝叶斯分类算法中最简单常见的一类分类算法。贝叶斯分类算法以贝叶斯定理为基础。假设样本集 $D = \{(x_1, y_1), \dots, (x_m, y_m)\}$ ，样本标签 $\{y_1, \dots, y_n\}$ 包含 K 个类别，样本特征 $x_i = \{x_{i1}, \dots, x_{in}\}$ 为 n 维向量。则对于任意样本特征 x_i ，其属于第 k 类的概率可由贝叶斯公式表示为：

$$P(y_i = k|x_i) = \frac{P(x_i|k)P(k)}{P(x_i)} = \frac{P(x_i|k)P(k)}{\sum_k P(x_i|k)P(k)} \propto P(x_i|k)P(k)$$

上式中 $P(y_i = k|x_i)$ 称为后验 (the posterior)， $P(x_i|k)$ 称为似然 (likelihood)， $P(k)$ 称为先验 (the prior)。基于上式求解分类问题的算法统称贝叶斯分类算法；

- 可以看到算法将求解未知量 $P(k|x_i)$ 转化为计算 $P(x_i|k)$, $P(k)$, $P(x_i)$ 的问题，分别表示样本集中标签为 k 的样本中特征为 x_i 的概率、样本集中标签为 k 的样本的出现概率和样本集中特征为 x_i 的样本的出现概率，而以上三者均可由训练集直接推算。而且因为分母 $P(x_i)$ 本质上为归一化因子，只需计算 $P(x_i|k)$, $P(k)$ 即可求得 $P(k|x_i)$ ；

- 特别的，若样本特征的各维度分量互相独立，则由概率乘法公式有

$$P(x_i|k) = \prod_{j=1}^n P(x_{ij}|k)$$

显然结合上式可进一步降低计算难度，而假定各维度特征互相独立即是“朴素贝叶斯”中“朴素”的来源。以贝叶斯定理为基础，假定各维度特征互相独立的贝叶斯分类算法即为朴素贝叶斯分类算法；

- 进一步地介绍 $P(x_i|k)$, $P(k)$ 的具体计算方法。在分类问题中，样本标签为离散值，样本特征则可能为连续值或离散值。对于不同的特征类型以上三类概率的计算方法不完全相同，因此可将朴素贝叶斯算法分为多项式、伯努利和高斯三种典型模型：

- 多项式模型适用于离散数据下的概率计算。为避免测试集因存在训练集中不存在的特征或标签类别导致计算的后验概率 $P(k|x_i) = 0$ 的情况，采用如下公式计算相关概率（又称拉普拉斯平滑）

$$P(x_{ij}|k) = \frac{M_{x_{ij},k} + \lambda}{M_k + S_j \lambda} \quad P(k) = \frac{M_k + \lambda}{M + K \lambda}$$

式中 M , M_k , $M_{x_{ij},k}$ 分别表示训练集样本量、训练集中标签为 k 的样本量、和训练集中标签为 k 且第 j 个特征为 x_{ij} 的样本量； S_j 表示第 j 个特征的类别数； λ 为平滑超参数，一般取 1。

- 伯努利模型适用于二值数据下的概率计算。因为二值数据属于特殊的离散数据，因此伯努利模型的概率计算公式与多项式模型基本一致。当特征或标签本身即为二值数据时，可直接基于多项式模型公式计算；当特征或标签不是二值数据时，算法基于提前预设的阈值将特征或标签二值化，再直接基于多项式模型公式计算。

- 高斯模型适用于连续数据下的概率计算。因分类问题中类别标签必然为离散变量，因此高斯模型仅针对存在连续样本特征的情况。记第 j 个特征为连续变量，假设样本标签为 k 时 x_{ij} 服从高斯分布 $p(x_{ij}|k) \sim N(\mu_{k,j}, \sigma_{k,j}^2)$ ，其中 $\mu_{k,j}, \sigma_{k,j}$ 分别为标签为 k 的样本集合的第 j 个特征的均值和标准差，则 x_{ij} 的条件概率密度 $p(x_{ij}|k)$ 计算式为

$$p(x_{ij}|k) = \frac{1}{\sqrt{2\pi\sigma_{k,j}^2}} \exp \left\{ -\frac{(x_{ij} - \mu_{k,j})^2}{2\sigma_{k,j}^2} \right\}$$

27.4 贝叶斯线性回归

- 对应于分类问题中的朴素贝叶斯算法，贝叶斯线性回归是贝叶斯推断于回归问题中的最简单应用³。不过贝叶斯线性回归以估计模型参数 ω 为目标，而不直接预测标签值 Y ，因此不需要假设训练集各特征之间

³ 回归的多种写法：线性回归-贝叶斯线性回归-高斯过程回归（理论篇）：https://zhuanlan.zhihu.com/p/350389546?ivk_sa=1024320u

相互独立。考虑基本的线性回归模型如下，其中 $\varepsilon \sim N(0, \sigma^2)$ 为随机误差， ω 为模型待求参数

$$Y = X^T \omega + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

基于频率学派理论的传统线性回归模型普遍从点估计出发，将 ω 视为未知常数，从而将 ω 估计转化为优化问题。而贝叶斯推断理论则将 ω 视为未知随机变量，首先基于贝叶斯公式确定 ω 的后验概率分布（即 **inference** 部分），进而确定在自变量 x^* 确定下回归模型预测目标 y^* 的后验概率分布（即 **prediction** 部分）；

- 首先推断 ω 的后验概率分布 $P(\omega|X, Y)$ 。基于贝叶斯公式， $P(\omega|X, Y)$ 可以由似然 $P(Y|\omega, X)$ 和先验分布 $P(\omega)$ 确定

$$P(\omega|X, Y) = \frac{P(Y|\omega, X)P(\omega)}{\sum_w P(Y|w, X)P(w)} \propto P(Y|\omega, X)P(\omega)$$

假设先验分布服从正态分布 $\omega \sim N(0, \Sigma_p)$ ，并展开似然计算式 $P(Y|\omega, X) = \prod_i P(y_i|\omega, x_i)$ ，显然 $y_i|\omega, x_i \sim N(x_i^T \omega, \sigma^2)$ ，因此有

$$\begin{aligned} P(Y|\omega, X) &= \prod_i P(y_i|\omega, x_i) \propto \prod_i \exp \left\{ -\frac{1}{2\sigma^2} (y_i - x_i^T \omega)^2 \right\} \\ &= \exp \left\{ -\frac{1}{2\sigma^2} \sum_i (y_i - x_i^T \omega)^2 \right\} = \exp \left\{ -\frac{1}{2\sigma^2} (Y^T - \omega^T X)(Y - X^T \omega) \right\} \sim N(X^T \omega, \sigma^2 I) \end{aligned}$$

因为似然 $P(Y|\omega, X)$ 和先验分布 $P(\omega)$ 均服从正态分布，因此后验分布同样服从正态分布 $P(\omega|X, Y) \sim N(\mu_\omega, \Sigma_\omega)$ ，参考第23.7.2节采用的对比系数法，同样可以求解 $\mu_\omega, \Sigma_\omega$ 。对于任意服从正态分布的随机变量 $Z \sim N(\mu, \Sigma)$ ，其概率分布函数指数项展开后有

$$P(Z) \propto \exp \left\{ -\frac{1}{2} (Z^T - \mu^T) \Sigma^{-1} (Z - \mu) \right\} = \exp \left\{ -\frac{1}{2} (Z^T \Sigma^{-1} Z - 2Z^T \Sigma^{-1} \mu + \mu^T \Sigma^{-1} \mu) \right\}$$

而同样展开后验分布 $P(\omega|X, Y)$ 的指数项有

$$\begin{aligned} P(\omega|X, Y) &\propto P(Y|\omega, X)P(\omega) = \exp \left\{ -\frac{1}{2} (Y^T - \omega^T X) \sigma^{-2} I (Y - X^T \omega) - \frac{1}{2} \omega^T \Sigma_p^{-1} \omega \right\} \\ &= \exp \left\{ -\frac{1}{2} [\omega^T (\sigma^{-2} X X^T + \Sigma_p^{-1}) \omega - 2\sigma^{-2} \omega^T X Y + \sigma^{-2} Y^T Y] \right\} \end{aligned}$$

对比系数有

$$\begin{cases} \Sigma_\omega^{-1} = \sigma^{-2} X X^T - \Sigma_p^{-1} \\ \Sigma_\omega^{-1} \mu_\omega = \sigma^{-2} X Y \end{cases} \implies \begin{cases} \Sigma_\omega = (\sigma^{-2} X X^T + \Sigma_p^{-1})^{-1} \\ \mu_\omega = \sigma^{-2} (\sigma^{-2} X X^T + \Sigma_p^{-1})^{-1} X Y \end{cases}$$

- 求解 $\mu_\omega, \Sigma_\omega$ 确定后验分布 $P(\omega|X, Y)$ 后，即可基于给定自变量 x^* 预测 y^* 的概率分布：

- 当不考虑随机误差时（即 $y^* = x^{*T} \omega$ ），有 $y^* \sim N(x^{*T} \mu_\omega, x^{*T} \Sigma_\omega x^*)$ ；
- 当考虑随机误差时（即 $y^* = x^{*T} \omega + \varepsilon, \varepsilon \sim N(0, \sigma^2)$ ），有 $y^* \sim N(x^{*T} \mu_\omega, x^{*T} \Sigma_\omega x^* + \sigma^2 I)$ 。

27.5 高斯过程回归 (Gaussian Process Regression, GPR)

27.5.1 从高斯过程到高斯过程回归——函数空间视角 (function-space)

- 高斯过程 (Gaussian process) 是基于统计理论和贝叶斯推断方法发展出来的一种机器学习方法，与贝叶斯线性回归、KNN、支持向量机等方法存在紧密联系，并可与神经网络技术结合；
- 理论上高斯过程适用于非线性数据的分类和回归问题，但在具体实施过程中多应用于高斯过程回归 (Gaussian process regression, GPR)，而分类领域则更多采用类似但效果更好的支持向量机模型；
- 首先介绍高斯过程。高斯过程可以理解为多维高斯分布推广至无限维的场景。对于一个 n 维高斯分布，每一次采样即可得到 n 维特征的一组取值；当 $n \rightarrow \infty$ 时，则一次采样即对应无限维特征的一组取值，也可理解为一次采样对应一个函数，函数曲线的任意一点对应无限维特征的一个分量，该点的取值对应该分量的取值。该无限维高斯分布即为高斯过程，对应了一个函数的分布；

4. 高斯过程正式地定义为：对于所有 $\mathbf{x} = [x_1, \dots, x_n]$, 若存在一个映射 f , 使得 $f(\mathbf{x}) = [f(x_1), \dots, f(x_n)]$ 均服从多元高斯分布, 则称 f 为一个高斯过程, 表示为

$$f(\mathbf{x}) \sim N(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}))$$

上式中 $\mu(\mathbf{x})$ 为均值函数 (mean function), 返回各个维度的均值; $\kappa(\mathbf{x}, \mathbf{x})$ 为协方差函数 (covariance function), 返回两个向量各个维度的协方差矩阵。在机器学习中, $\kappa(\mathbf{x}, \mathbf{x})$ 也称为核函数 (kernel function)⁴;

5. 核函数 $\kappa(\mathbf{x}, \mathbf{x})$ 是高斯过程模型的核心。根据定义, 对于任意坐标点 x_i , 其标签 $y_i = f(x_i)$ 服从以 $\mu(x_i)$ 为均值、 $\kappa(x_i, x_i)$ 为标准差的正态分布, 因此核函数 $\kappa(\mathbf{x}, \mathbf{x})$ 的对角线元素定义了高斯过程所对应的函数分布于各个坐标点的取值范围。同样, 对于任意坐标点 $x_i \neq x_j$, 两者标签 $f(x_i), f(x_j)$ 分布的协方差为 $\kappa(x_i, x_j)$, 协方差越大意味着函数于 x_i, x_j 两点处的取值相关性越大, 因此核函数 $\kappa(\mathbf{x}, \mathbf{x})$ 的周围元素定义了高斯过程所对应的函数分布沿坐标轴变化的平滑程度。综上, 选择不同的核函数类型和核函数超参取值, 即可改变高斯过程模型的拟合效果。介绍几类常用的核函数如下:

- 线性核函数: $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$;
- 多项式核函数: $\kappa(\mathbf{x}, \mathbf{x}') = (l\mathbf{x}^T \mathbf{x}' + r)^d$
- 高斯噪声核函数: $\kappa(\mathbf{x}, \mathbf{x}') = \sigma^2 \delta_{\mathbf{x}, \mathbf{x}'}$;
- 高斯核函数/径向基函数 (radial basis function, RBF) (最常用): $\kappa(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2l^2}\right\}$;
- Matern 核函数 (RBF 函数的泛化): $\kappa(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{x}-\mathbf{x}'\|}{l}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}\|\mathbf{x}-\mathbf{x}'\|}{l}\right)$, K_ν 为修正的赛贝尔函数;
- 周期核函数: $\kappa(\mathbf{x}, \mathbf{x}') = \exp\left\{-2 \frac{\sin^2(\|\mathbf{x}-\mathbf{x}'\|/2)}{l^2}\right\}$;
- sigmoid 核函数: $\kappa(\mathbf{x}, \mathbf{x}') = \tanh(l\mathbf{x}^T \mathbf{x}' + r)$ 。

上式中 l, r, d, σ, ν 均为核函数超参;

6. 相比于传统机器学习模型, 基于高斯过程的机器学习模型在训练时不以优化参数取值为目标, 而是优化高斯过程对应的函数分布, 因此训练后的高斯过程模型可显式地描述不确定性。另一方面, 传统机器学习模型在训练时往往要求样本服从正态分布, 若样本分布有偏则往往会造成学习效果的偏差, 而高斯过程模型训练时则不要求训练样本服从正态分布, 因为任意的样本分布均可视为一个无限维高斯分布的一次随机采样的结果, 训练样本有偏可以反映为高斯过程模型的不确定度;
7. 进一步地基于高斯过程理论导出高斯过程回归 (Gaussian process regression, GPR)。初始化一个高斯过程先验 f , 假设训练集 (\mathbf{x}, \mathbf{y}) 满足高斯过程, 则有 $\mathbf{y} = f(\mathbf{x}) + \varepsilon$ 。注意到高斯过程可视为无穷维高斯分布, 因此可以假设测试集与训练集 $[f^*(\mathbf{x}^*), \mathbf{y}]^T$ 共同满足上述高斯过程, 其中 f^* 为训练得到的高斯过程后验, 从而实现对测试集的预测 $\mathbf{y}^* = f^*(\mathbf{x}^*)$ 。上述过程即为高斯过程回归。因为高斯过程的特点, 理论上, 高斯过程回归可拟合任意函数;
8. 进一步推导高斯过程回归的基本公式。首先假设高斯过程先验 $f(\mathbf{x}) \sim N(\mu_f, K_{ff})$, 对于训练集 (\mathbf{x}, \mathbf{y}) 和测试集 \mathbf{x}^* , 假设 $[f^*(\mathbf{x}^*), \mathbf{y}]^T$ 共同满足上述高斯过程

$$\begin{bmatrix} f^* \\ \mathbf{y} \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_f \\ \mu_y \end{bmatrix}, \begin{bmatrix} K_{ff} & K_{fy} \\ K_{fy}^T & K_{yy} + \sigma_n^2 I \end{bmatrix}\right) \quad K_{ff} = \kappa(\mathbf{x}^*, \mathbf{x}^*), \quad K_{fy} = \kappa(\mathbf{x}^*, \mathbf{x}), \quad K_{yy} = \kappa(\mathbf{x}, \mathbf{x})$$

根据多元高斯分布的条件分布公式 (见第23.7.2节), 可以得到高斯过程后验 f^*

$$f^* | \mathbf{x}, \mathbf{y}, \mathbf{x}^* = N(\mu_f + K_{fy}(K_{yy} + \sigma_n^2 I)^{-1}(\mathbf{y} - \mu_y), K_{ff} - K_{fy}(K_{yy} + \sigma_n^2 I)^{-1}K_{fy}^T)$$

一般地, 常令先验均值函数 $\mu = \mathbf{0}$, 则上式可简化为

$$f^* | \mathbf{x}, \mathbf{y}, \mathbf{x}^* = N(K_{fy}(K_{yy} + \sigma_n^2 I)^{-1}\mathbf{y}, K_{ff} - K_{fy}(K_{yy} + \sigma_n^2 I)^{-1}K_{fy}^T)$$

最后, 若考虑预测噪声 $\mathbf{y}^* = f^*(\mathbf{x}^*) + \varepsilon$, 则上式变为

$$f^* | \mathbf{x}, \mathbf{y}, \mathbf{x}^* = N(K_{fy}(K_{yy} + \sigma_n^2 I)^{-1}\mathbf{y}, K_{ff} - K_{fy}(K_{yy} + \sigma_n^2 I)^{-1}K_{fy}^T + \sigma_n^2 I)$$

⁴核与内积、距离等名词的意思一致, 均表示相似度。核函数的计算值越小, 说明两个点的相似度越小, 反之则越大。

27.5.2 从 KNN 到高斯过程回归——无参数机器学习

1. 在最终预测时，一般直接以高斯回归后验的均值函数 $\mu^* = K_{fy}(K_{yy} + \sigma_n^2 I)^{-1} y$ 作为预测结果 y^* ，因此讨论均值函数 μ^* 以进一步理解高斯过程回归的原理。假设训练集和预测集的维度分别为 n, m ：

- 令 $z = (K_{yy} + \sigma_n^2 I)^{-1} y$ ，显然 z 为 $n \times 1$ 向量，并将矩阵乘法展开为元素乘积加和的形式，则可以得到测试集样本 x_i^* 的预测值 $y_i^* = \sum_j^n \kappa(x_i^*, x_j) z_j$ 。显然 y_i^* 的预测结果为训练集所有样本 $x_j (j = 1, \dots, n)$ 的加权平均，训练集样本 x_j 的权重与 $\kappa(x_i^*, x_j)$ 正相关——若 x_i^*, x_j 越接近（相似度越大），则权重越大；
- 进一步展开 z_j ，为简化考虑忽视噪声 σ_n ，则 $z_j = \sum_k^n \lambda(x_j, x_k) y_k, \Lambda = K^{-1}$ 。显然在预测 y_i^* 时，并不是直接基于训练集样本所有样本 $x_j (j = 1, \dots, n)$ 与测试集样本 x_i^* 的相似度作为权重对 $y_j (j = 1, \dots, n)$ 进行加权，而是提前对 $y_k (k = 1, \dots, n)$ 进行加权得到 z_j ，再加权计算 y_i^* 。而加权计算 z_j 时的权重与 $\kappa(x_j, x_k)$ 有关——即同时考虑训练集样本 j 的标签值 y_j 和其它与样本 j 相近的训练集样本 k 的标签值 y_k 共同确定 z_j 。

2. 上述讨论进一步论证了核函数 $\kappa(x, x)$ 在高斯过程回归中的核心地位。同时也建立了高斯过程回归模型与经典 KNN 算法的相似性：

- KNN 算法仅考虑训练集中与测试集样本 x_i^* 最近的 K 个样本的标签 $y_j, j = 1, \dots, K$ 加权预测 x_i^* 的标签 y_i^* ，其它训练集样本的权重则视为 0；而高斯过程回归则考虑了训练集的所有样本，并以相似度作为权重区分不同训练集样本对预测的贡献度。从这个角度考虑，可以说 KNN 为仅考虑邻近样本的“硬”加权，而高斯过程回归为考虑所有样本的“软”加权；
- KNN 算法预测时直接在训练集标签 y_j 的基础上进行加权预测，而高斯过程回归则首先对所有训练集样本加权构造一个新的标签 $z_j = \sum_k^n \lambda(x_j, x_k) y_k$ ，再进行第二轮加权预测测试集样本标签。在构造 z 的过程中，可认为核函数 $\kappa(x, x)$ 起到平滑的效果，因此可以说 KNN 为直接在原始样本的基础上进行加权，而高斯过程回归模型则是在平滑后的训练集样本基础上进行加权。

3. 综上所述，高斯过程回归算法与经典 KNN 算法同属于无参数方法，不通过预设函数形式并优化超参数的方式拟合训练集，而是直接基于训练集样本与测试集样本的相似度直接进行预测，这也等价于函数空间视角的解释——高斯过程回归直接优化一个函数分布；

4. 至此可以总结高斯过程回归的优缺点。其中优点如下：

- 高斯过程回归为非参数模型，理论上可以拟合任意形式函数，因此支持非线性建模；
- 高斯过程回归天然支持对不确定性的建模，可以直接得到预测值的概率分布。

同时模型也具有如下缺点：

- 因为为非参数模型，每次推断优化都需要对所有训练数据点进行矩阵求逆， n 个样本点的时间复杂度为 n^3 ，空间复杂度为 n^2 ，在数据量大时变得无法求解；
- 高斯过程回归中假设先验为高斯分布，同时要求似然满足高斯分布，因此在似然不服从高斯分布的问题中，需要进行近似。

27.5.3 从贝叶斯线性回归到高斯过程回归——权重空间视角 (weight-space)

1. 在前两小节的讨论中，从两个角度解释了核函数于高斯过程模型中的作用——作为物理意义为相似度的权重和作为平滑算子。但在前面的讨论中并未明显地体现高斯过程回归作为贝叶斯机器学习方法与其它贝叶斯模型的联系。本小节从贝叶斯线性回归（第27.4节）出发推导高斯过程回归基本公式，同时也引出核函数的另一种解释：

2. 注意到贝叶斯线性回归最大的问题在于无法拟合非线性数据，借鉴支持向量机的思路，一个可行的方式即将低维空间下的非线性数据升维至高维空间并线性化，再在高维空间下进行贝叶斯线性回归。定义一个从低维空间到高维空间的非线性映射 φ ，此时贝叶斯线性回归模型变为

$$y = \Phi^T \omega + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2), \quad \Phi = (\varphi(x_1), \dots, \varphi(x_n))$$

直接代入贝叶斯线性回归的结果，有

$$y^* = f(x^*) \sim N\left(\varphi(x^*)^T \sigma^{-2} (\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1})^{-1} \Phi y, \varphi(x^*)^T (\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1})^{-1} \varphi(x^*)\right)$$

3. 进一步基于矩阵代数公式对上式进行变换。首先变换均值 $\mu = \varphi(\mathbf{x}^*)^T \sigma^{-2} (\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1})^{-1} \Phi \mathbf{y}$ 。注意到

$$\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1} = (\sigma^{-2} \Phi \Phi^T \Sigma_p \Phi + \Phi) \Phi^{-1} \Sigma_p^{-1} = \sigma^{-2} \Phi (\Phi^T \Sigma_p \Phi + \sigma^2 I) \Phi^{-1} \Sigma_p^{-1}$$

由 $(AB)^{-1} = B^{-1}A^{-1}$ 得

$$\begin{aligned} \mu &= \varphi(\mathbf{x}^*)^T \sigma^{-2} (\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1})^{-1} \Phi \mathbf{y} = \varphi(\mathbf{x}^*)^T \sigma^{-2} \left(\sigma^2 \Sigma_p \Phi (\Phi^T \Sigma_p \Phi + \sigma^2 I)^{-1} \Phi^{-1} \right) \Phi \mathbf{y} \\ &= \varphi(\mathbf{x}^*)^T \Sigma_p \Phi (\Phi^T \Sigma_p \Phi + \sigma^2 I)^{-1} \mathbf{y} \end{aligned}$$

再变换方差 $\Sigma = \varphi(\mathbf{x}^*)^T (\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1})^{-1} \varphi(\mathbf{x}^*)$ 。对于其中的 $(\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1})^{-1}$ 采用另一种变换方法。由伍德伯里矩阵恒等式 (Woodbury matrix identity)

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

则令 $\Sigma_p^{-1} = A$, $\Phi = U$, $\sigma^{-2}I = C$, $\Phi^T = V$, 方差 Σ 可变换为

$$\begin{aligned} \Sigma &= \varphi(\mathbf{x}^*)^T (\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1})^{-1} \varphi(\mathbf{x}^*) = \varphi(\mathbf{x}^*)^T \left(\Sigma_p - \Sigma_p \Phi (\sigma^2 I + \Phi^T \Sigma_p \Phi)^{-1} \Phi^T \Sigma_p \right) \varphi(\mathbf{x}^*) \\ &= \varphi(\mathbf{x}^*)^T \Sigma_p \varphi(\mathbf{x}^*) - \varphi(\mathbf{x}^*)^T \Sigma_p \Phi (\sigma^2 I + \Phi^T \Sigma_p \Phi)^{-1} \Phi^T \Sigma_p \varphi(\mathbf{x}^*) \end{aligned}$$

4. 观察变换后的均值 μ 和方差 Σ 公式, 可以发现公式中频繁出现以下结构 $\varphi(\mathbf{x}^*)^T \Sigma_p \Phi$, $\Phi^T \Sigma_p \Phi$, $\varphi(\mathbf{x}^*)^T \Sigma_p \varphi(\mathbf{x}^*)$, $\Phi^T \Sigma_p \varphi(\mathbf{x}^*)$, 其中 $\Phi, \varphi(\mathbf{x}^*)$ 分别表示高维空间下的训练集和测试集。上述结构可进一步统一形式, 定义

$$\kappa(x, x') = \varphi(x)^T \Sigma_p \varphi(x') = (\Sigma_p^{1/2} \varphi(x))^T (\Sigma_p^{1/2} \varphi(x')) = \langle \Sigma_p^{1/2} \varphi(x), \Sigma_p^{1/2} \varphi(x') \rangle$$

则有 $K_{fy} = \kappa(\mathbf{x}^*, \mathbf{x}) = \varphi(\mathbf{x}^*)^T \Sigma_p \Phi$, $K_{yy} = \kappa(\mathbf{x}, \mathbf{x}) = \Phi^T \Sigma_p \Phi$, $K_{ff} = \kappa(\mathbf{x}^*, \mathbf{x}^*) = \varphi(\mathbf{x}^*)^T \Sigma_p \varphi(\mathbf{x}^*)$, $K_{yf} = \kappa(\mathbf{x}, \mathbf{x}^*) = \Phi^T \Sigma_p \varphi(\mathbf{x}^*)$, 因而可进一步改写高维空间下贝叶斯线性回归的结果

$$\begin{aligned} \mathbf{y}^* &= f(\mathbf{x}^*) \sim N \left(\varphi(\mathbf{x}^*)^T \sigma^{-2} (\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1})^{-1} \Phi \mathbf{y}, \varphi(\mathbf{x}^*)^T (\sigma^{-2} \Phi \Phi^T + \Sigma_p^{-1})^{-1} \varphi(\mathbf{x}^*) \right) \\ &= N \left(\varphi(\mathbf{x}^*)^T \Sigma_p \Phi (\Phi^T \Sigma_p \Phi + \sigma^2 I)^{-1} \mathbf{y}, \varphi(\mathbf{x}^*)^T \Sigma_p \varphi(\mathbf{x}^*) - \varphi(\mathbf{x}^*)^T \Sigma_p \Phi (\sigma^2 I + \Phi^T \Sigma_p \Phi)^{-1} \Phi^T \Sigma_p \varphi(\mathbf{x}^*) \right) \\ &= N \left(K_{fy} (K_{yy} + \sigma^2 I)^{-1} \mathbf{y}, K_{ff} - K_{fy} (\sigma^2 I + K_{yy})^{-1} K_{yf} \right) \end{aligned}$$

5. 显然上式与高斯过程回归的公式完全一致, 因此高维空间下的贝叶斯线性回归等价于高斯过程回归, 所定义的 $\kappa(x, x')$ 即为高斯过程模型中的核函数。在权重空间视角的理解下, 高斯过程模型中的核函数与支持向量机中的核函数完全一致, 是一种高维向量内积的简便计算方法。在高斯过程回归和支持向量机模型中, 均需要将低维空间下的非线性数据非线性映射至高维空间使其线性化, 并计算高维空间内向量的距离 (相似度)。而通过定义核函数 $\kappa(x, x') = (\Sigma_p^{1/2} \varphi(x))^T (\Sigma_p^{1/2} \varphi(x')) = \langle \Sigma_p^{1/2} \varphi(x), \Sigma_p^{1/2} \varphi(x') \rangle$ 为 x, x' 的函数, 可以绕过高维非线性映射函数 φ 而直接在低维空间下计算高维空间下的相似度, 起到减小计算量的效果。

27.5.4 超参数优化

- 在高斯过程模型中, 核函数 κ 至关重要, 对此已在前文中充分讨论。需要注意的是, 核函数中存在超参, 为了使得高斯过程回归的结果最优, 需要提前对核函数的超参进行估计;
- 频率学派一般基于极大似然估计 (maximum likelihood estimation, MLE) 估计模型参数, 而贝叶斯学派则一般基于最大后验分布估计 (maximum a posteriori estimation, MAP) 估计模型参数。在高斯过程回归的超参估计中, MLE 和 MAP 估计的表达式完全一致;
- 以 MLE 估计为例, 记核函数超参为 θ , 注意到似然函数 $p(\mathbf{y}|\mathbf{x}, \theta)$ 可以写为

$$p(\mathbf{y}|\mathbf{x}, \theta) = \int p(\mathbf{y}|f, \mathbf{x}, \theta) p(f|\mathbf{x}, \theta) df$$

在统计学中, 形如上式的似然函数称为边际似然函数 (marginal likelihood, or integrated likelihood)。式中 $p(f|\mathbf{x}, \theta)$ 为高斯过程 f 的先验, 显然有 $p(f|\mathbf{x}, \theta) = N(0, K_\theta)$; $p(\mathbf{y}|f, \mathbf{x}, \theta)$ 为从函数分布中抽样得一个函数 f 后的似然, 根据 $\mathbf{y} = f(\mathbf{x}) + \varepsilon$, 显然有 $p(\mathbf{y}|f, \mathbf{x}, \theta) = N(f(\mathbf{x}), \sigma_n^2 I)$;

4. 为最大化边际似然函数, 最直接的思路即直接计算上述积分, 但也存在更快的办法。注意到 $p(\mathbf{f}|\mathbf{x}, \theta), p(\mathbf{y}|\mathbf{f}, \mathbf{x}, \theta)$ 均为高斯分布, 则 $p(\mathbf{y}|\mathbf{x}, \theta)$ 同样服从高斯分布。又因为

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) + \boldsymbol{\varepsilon} \quad \mathbf{f} \sim N(0, K_\theta), \quad \boldsymbol{\varepsilon} \sim N(0, \sigma_n^2 I)$$

因此显然有 $p(\mathbf{y}|\mathbf{x}, \theta) = N(0, K_\theta + \sigma_n^2 I)$, 从而有

$$\ln p(\mathbf{y}|\mathbf{x}, \theta) = -\frac{1}{2} (\mathbf{y}^T (K_\theta + \sigma_n^2 I)^{-1} \mathbf{y} + \ln \|K_\theta + \sigma_n^2 I\| + n \ln 2\pi)$$

只需最大化上式, 即可得到超参数 θ 的取值;

5. 需要说明的是, 在高斯过程回归模型的边际似然函数为非凸, 因此优化过程中无法保证得到全局最优解, 往往为局部最优解。另外, 除了传统的估计方法, 如 MCMC 等现代估计方法也可也可用于优化高斯过程模型的超参数。

27.6 贝叶斯优化 (*Bayesian Optimization*)

- 贝叶斯优化理论由 Jonas Mockus 于 20 世纪 70-80 年代提出, 是一种无梯度全局优化算法;
- 从优化的角度而言, 其属于智能优化算法; 而从机器学习的角度而言, 方法与强化学习框架一致, 均是旨在解决序贯决策 (sequential decision) 问题的常见算法 (见28.1节);
- 为解决序贯决策问题, 贝叶斯优化方法与强化学习框架具有相似的框架——包括一个预测结构和一个决策结构, 预测结构用于建模待优化对象, 而决策结构基于当前的预测结果寻找可能的最优解。在贝叶斯优化理论中, 分别称上述两结构为概率代理模型 (probabilistic surrogate model) 和采集函数 (acquisition function)。不同于强化学习框架中将预测结构 (也可能包括决策结构) 建模为不可解析的黑箱过程, 贝叶斯优化理论中概率代理模型和采集函数相对简单;
- 与强化学习相比, 因为概率代理模型和采集函数较为简单, 贝叶斯优化方法的优化效果总体弱于强化学习, 但实现难度更小计算效率更高; 而相比梯度下降或元启发算法等纯粹以优化为目标的算法, 贝叶斯优化方法借助概率代理模型在优化的同时也可对目标函数进行建模, 理论上具有更优的泛化能力;
- 进一步分别介绍贝叶斯优化的核心——概率代理模型和采集函数的技术细节。考虑一个典型的优化问题

$$\begin{aligned} & \min f(\mathbf{x}) \\ \text{s.t. } & g_j(\mathbf{x}) \leq 0, \quad h_k(\mathbf{x}) = 0 \end{aligned}$$

当目标函数 f 的形式不可知 (黑箱模型) 或过于复杂时, 即需要包括贝叶斯优化算法在内的智能优化算法进行求解。贝叶斯优化算法首先构造可解析的概率代理模型 F 以近似原目标函数 f , 且作为一种无梯度优化算法, F 完全基于目标函数于采样点的取值确定;

- 构造简化模型近似复杂模型从而方便求解的思路与传统优化算法中的信赖域方法具有一定的相似性 (见17.3.4节), 两者的差别主要体现为:
 - 信赖域优化中的近似为局部近似。构造的简化模型仅要求在以当前采样点为中心的信赖域内近似原目标函数, 其优点在于不要求简化模型具有过强的拟合能力 (常为二项式模型), 但缺点在于难以得到全局最优解, 而且无法利用之前的所有采样点的结果;
 - 而作为全局优化算法, 贝叶斯优化中的近似为全局近似。要求概率代理模型尽可能利用之前所有采样的结果, 对复杂目标函数进行全局拟合。为此将贝叶斯推断过程应用于概率代理模型建模过程中——在每一次采样前视已有的概率代理模型为先验, 而每一次采样的结果作为似然, 从而更新概率代理模型后验。另外全局近似的需求也要求概率代理模型具有较强的拟合能力, 高斯过程模型 (见27.5节) 为最常用的概率代理模型, 除此之外也可使用随机森林、神经网络等非参数模型和贝塔-伯努利 (Beta-Bernoulli) 模型、线性模型、广义线性模型等参数模型。
- 每一轮采样后基于概率代理模型后验确定下一采样点, 采样点应该为“最有可能”使得目标函数下降的点。注意到所谓“最有可能”应该从以下两方面考虑:

- 对于采样空间内采样密度高或者距离采样点近的部分，概率代理模型的确定性较高，因此相关区域概率代理模型的极小值点即属于可能使得目标函数下降的点；
- 对于采样空间内采样密度低或者距离采样点远的部分，概率代理模型的不确定性较高，因此相关区域概率代理模型的置信区间下界的极小值点同样属于可能使得目标函数下降的点。

8. 在贝叶斯优化方法中，通过最大化采集函数实现上述采样过程。采集函数 $\alpha(\mathbf{x})$ 为定义在采样空间内关于决策变量 \mathbf{x} 的函数， $\alpha(\mathbf{x}^*)$ 量化了采样 $\mathbf{x} = \mathbf{x}^*$ 处后使得目标函数下降的潜在贡献。通过设置不同类型的采集函数可实现不同的采样效果：

- 概率提升 (*probability of improvement, PI*) 法：

PI 法是最早被提出的一种采集函数，也是最简单的一种。方法直接将 $\alpha(\mathbf{x})$ 定义为采样点 \mathbf{x} 能使得目标函数下降的概率

$$\alpha(\mathbf{x}) = p(F(\mathbf{x}) \leq y_{best} - \xi)$$

上式中 F 即为概率代理模型， y_{best} 为目前以采样的目标函数的最低点， ξ 为模型超参用于鼓励探索。当概率代理模型 F 为高斯过程模型时，由27.5节可知，对于任意点 \mathbf{x} ， $F(\mathbf{x})$ 服从以 $\mu(\mathbf{x})$ 为均值、 $\sigma(\mathbf{x})$ 为标准差的正态分布，则显然有

$$\alpha(\mathbf{x}) = p(F(\mathbf{x}) \leq y_{best} - \xi) = \Phi\left(\frac{y_{best} - \xi - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)$$

其中 Φ 为标准正态分布的概率累积函数。而新的采样点 \mathbf{x}^* 的确定依据即为

$$\mathbf{x}^* = \arg \max \alpha(\mathbf{x}) = \arg \max \Phi\left(\frac{y_{best} - \xi - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)$$

PI 法最大的缺点在于其构造采集函数时仅考虑了采样点对优化的提升概率，而未考虑提升的幅度；

- 期望提升 (*expected improvement, EI*) 法：

EI 法是目前最常用的采集函数。函数一方面保持了 PI 法简单的特点，另一方面又在考虑提升概率的基础上进一步考虑了提升幅度，两者结合，将 $\alpha(\mathbf{x})$ 定义为采样点 \mathbf{x} 能使得目标函数下降的期望

$$\alpha(\mathbf{x}) = \mathbb{E}(\min\{0, F(\mathbf{x}) - y_{best} + \xi\}) = \int_{-\infty}^{y_{best}-\xi} p(F(\mathbf{x})) \cdot (y_{best} - \xi - F(\mathbf{x})) d\mathbf{x}$$

同样地，若概率代理模型 F 为高斯过程模型，则

$$\alpha(\mathbf{x}) = \mathbb{E}(\min\{0, F(\mathbf{x}) - y_{best} + \xi\}) = \begin{cases} (y_{best} - \xi - \mu(\mathbf{x}) + \sigma(\mathbf{x})) \Phi\left(\frac{y_{best} - \xi - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right), & \sigma(\mathbf{x}) > 0 \\ 0, & \sigma(\mathbf{x}) = 0 \end{cases}$$

9. 以上即为贝叶斯优化的基本原理。综上所述，**贝叶斯优化适用于目标函数为黑箱，且一次采样需消耗较大计算资源的优化问题**。有研究指出贝叶斯优化的计算效率优于元启发算法。目前贝叶斯优化最广泛应用于的场景为模型超参数优化。

27.7 基于贝叶斯推断的矩阵与张量分解

自 2007 年 Andriy Mnih 等人提出概率矩阵分解算法开始，大量研究尝试基于贝叶斯推断框架复现各经典矩阵与张量分解问题，形成了以概率矩阵分解 (probabilistic matrix factorization, 详见第 20.5 节)、贝叶斯概率矩阵分解 (bayesian probabilistic matrix factorization, 详见第 20.6 节)、贝叶斯非负矩阵分解 (bayesian non-negative matrix factorization, 详见第 20.7 节)、贝叶斯张量分解 (bayesian tensor factorization, 详见第 20.8 节) 等为代表的一系列贝叶斯机器学习降维算法。

基于贝叶斯推断理论复现上述经典的矩阵/张量分解算法不仅证明并扩展了传统贝叶斯统计流派的生命力，而且得益于贝叶斯推断理论长久以来对于“分布”整体的关注，使得上述基于贝叶斯推断的分解算法较传统的基于运筹优化的算法可捕捉更充分的不确定信息，也更适用于非凸问题建模。

第 28 章

强化学习 (Reinforcement Learning)

28.1 强化学习概述

1. 强化学习的主要目的是研究并解决机器人（智能体）序贯决策 (sequential decision) 问题。序贯决策是指机器人（智能体）在一定的环境下不断自我学习并最终得到该环境下最合理的行为决策。需要说明的是，并非所有序贯决策问题都只能由强化学习实现，只是强化学习具有更强的泛化能力；
2. 首先介绍强化学习的基本概念：
 - **Agent:** 智能体，强化学习过程中的行为主体；
 - **Environment:** 智能体所处的环境，可以是游戏环境、仿真环境或其它更复杂的环境；
 - **Observation (State):** 智能体观测的环境信息，由人为判断，用于表示智能体所处环境的某种状态；
 - **Action:** 智能体所做的动作，动作的发生可能引起状态 (O_t 或 S_t) 的改变（如车辆换道与否将影响下一刻的交通流状态），也可能与状态无关（如车辆换道与否与天气的变化无关）；
 - **Policy:** 策略，智能体所做的动作 A_t 应该和所观测的环境 S_t 有关，而策略即用于建立两者的关系，一般有以下两种表达形式：

$$A_t = \pi(s|\theta) \iff \pi(a|s) = P(A_t = a|S_t = s)$$

- **Behavior policy:** 行为策略，指智能体在和环境交互并得到数据过程中所采用的策略；
- **Target policy:** 目标策略，指利用行为策略得到的数据不断学习、优化并最终用于应用的策略。
- **Episode:** 状态序列，指智能体在一个环境中执行某种策略直至结束的全过程（如游戏结束、仿真结束等等）的所有状态；
- **Reward:** 奖励值，为强化学习过程中的超参 (hyperparameter)，需人为预设，用于评价智能体在完成一个动作后的效果，越大表示鼓励，反之表示抑制；

$$R_t = R(a|s) = R(\pi(s|\theta)|s)$$

- **Evaluation Function:** 评价函数，指在确定的环境下，对某一策略 π 的评价值。因为在某一环境下可能会存在多种状态，而同一策略对不同的状态会有不同的奖励值，因此评价函数往往是关于奖励值或状态的数学期望；

$$E[R(\pi(s|\theta)|s)] = \frac{1}{N} \sum_i R(a|s_i)\pi(a|s_i)P(s_i)$$

- **Value Function:** 值函数，分为状态值函数 $v_\pi(s)$ 和动作值函数 $q_\pi(s, a)$ 。状态值函数指在策略 π 确定的情况下，状态 $S_t = s$ 的估值，估值越高该状态越好。因为在一定的策略下，状态会随时间发生多次变化，得到多个奖励值，因此值函数是关于一定时间区间内的多个状态或奖励值的数学期望，并考虑到距离当前时间越远的状态或奖励值对值函数的影响应该越小

$$v_\pi(s) = E_\pi[\gamma^0 R_t + \gamma^1 R_{t+1} + \gamma^2 R_{t+2} + \dots | S_t = s] = E_\pi[G_t | S_t = s]$$

式中 $\gamma = 0 \sim 1$ 。类似的，定义动作值函数为对动作 $A_t = a$ 的估值

$$q_\pi(s, a) = E_\pi[\gamma^0 R_t + \gamma^1 R_{t+1} + \gamma^2 R_{t+2} + \dots | S_t = s, A_t = a] = E_\pi[G_t | S_t = s, A_t = a]$$

显然两者满足 $v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a)$ 。

3. 强化学习可以分为如下两个过程：

- (a) 预测问题：给定强化学习的几个要素，以及策略 π ，求解该策略下的状态值函数 $v_\pi(s)$ ；

- (b) 控制问题：求解最优的策略 π^* 及最优化函数 $(v_{\pi^*}(s), q_{\pi^*}(s, a))$ 。

4. 强化学习的过程可以概括为：在特定的环境下，智能体于 t 时刻所处的状态为 S_t ，并基于状态 S_t 执行动作 A_t ，得到奖励值 R_t 和新的状态 S_{t+1} 。经过多次迭代，智能体即可学得当前环境下的最优决策；

$$S_t \rightarrow A_t \rightarrow R_t \rightarrow S_{t+1} \quad A_t = \pi(S_t | \theta) \quad R_t = R(\pi(S_t | \theta) | S_t)$$

5. 得到最优策略的过程可以有 **policy-based**、**value-based** 和 **actor-critic** 三种方法：

- **policy-based**（直接法）：直接法是对评价函数进行优化，目标是得到最优策略 $\pi(s|\theta)$ ，其思路是：最优的策略应该使得一个 episode 后总奖励值最大。其优化参数为策略函数 $\pi(s|\theta)$ 中的参数 θ 。**policy-based** 方法存在以下缺点：
 - 只有在一个 episode 结束后才能对策略进行评估，因而更新较慢，并且有很高的偏差；
 - 通常得到的都是局部最优解。
- **value-based**（间接法）：间接法是对值函数进行优化，目标是得到任意状态 s 下的最优动作 a ，其思路是：对于任意环境，如果智能体能准确估计每一状态 s 及动作 a 的值函数 $q(s, a)$ ，则对于任意状态 s ，之需执行使得 $q(s, a)$ 最大的动作 a ，即是最优的控制策略。这一方法存在以下三个缺点：
 - 对高维度和连续动作的处理能力不足。动作空间过大时，需要正确地对大量 $q(s, a)$ 进行估值，会消耗大量的计算及存储资源，且难以收敛；
 - 对受限状态下的问题处理能力不足。当环境过于复杂、状态空间过大、而智能体的感知空间受限时（例如地图游戏，智能体不可能感知全幅地图的信息），两个不同的状态可能建模后拥有相同的状态特征，进而导致无法得到正确的值函数估值；
 - 无法解决随机策略问题。**value-based** 方法的动作选取是使动作值函数最大的动作，然而有些问题的最优策略却具有随机性（例如棋类游戏，最优的落子往往是有多种选择的）。
- **actor-critic**：**actor-critic** 是 **policy-based** 和 **value-based** 的结合，分为 actor 和 critic 两部分。actor 采用 **policy-based**，每次根据 s 得到 a ，而为了改进 **policy-based** 收敛速度慢的缺点加入 critic 部分，critic 采用 **value-based**，根据 actor 选择的 a 计算值函数 $q(s, a)$ ，从而对 actor 进行评价。actor 和 policy 两部分协同更新，直至收敛。
 - 优点：critic 的引入使得 actor 可以逐步更新，比纯粹的 **policy-based** 更快；
 - 缺点：actor 的正确更新取决于 critic 的正确估值，在 critic 收敛之前 actor 存在较大误差。

6. model-based 与 model-free：

- **model-based**：基于模型的强化学习方法，是指智能体通过多次观察其在状态 s 下执行的动作 a 得到奖励值 r 和下一个状态 s' 的过程后对当前环境建立模型，再基于已建立的模型运用规划方法解决问题的学习过程。这一方法的优点在于可以引用熟悉的监督学习方法，且模型建立后智能体对环境具有了一定的理解能力，而非仅限于最大化奖励本身。但这一过程需要涉及两个学习过程：对模型的学习和值函数的学习，会带来双重误差；
- **model-free**：无模型的强化学习方法，是指智能体不对环境本身进行建模，而是直接对动作值函数进行预测，从而选择使得值函数最大的动作的学习过程。相对于基于建模的学习方法，无模型的学习方法更为简单，但对样本利用效率较低。

7. on-policy learning 与 off-policy learning：

- **on-policy learning**：同步策略学习，当目标策略等于行为策略时即称为同步策略学习，即首先采用行为策略获得数据，并基于得到的数据对原策略进行更新，再用新的策略产生新的数据。该方法会遭遇探索-利用的矛盾，光利用目前已知的最优选择，可能学不到最优解，收敛到局部最优；

- off-policy learning: 异步策略学习，当目标策略不等于行为策略时即称为异步策略学习，行为策略用于生成数据，而用于优化的目标策略与行为策略不完全相等¹，使得行为策略未必是最优的，此时行为策略起到探索的效果，从而确保了数据全面性，但收敛慢。

28.2 动态规划 (*Dynamic programming, DP*) 与马尔科夫决策过程 (*Markov decision process, MDP*)

1. 动态规划过程具有如下特点：

- **最优子结构**: 指一个子问题的最优解是可以得到的；
- **重复子结构**: 指一个大问题可以分解为多个子问题，而每个子问题的最优解已知，从而求解不同大问题时均可重复用上子问题的结论，降低计算量。

2. 动态规划过程中，强化学习的两个问题是如下定义的：

- (a) 预测问题，给定强化学习的 6 个要素：状态集 S , 动作集 A , 模型状态转化概率矩阵 P , 即时奖励 R , 衰减因子 γ 和策略 π , 求解该策略的状态价值函数 $v_\pi(s)$;
- (b) 控制问题，给定强化学习的 5 个要素：状态集 S , 动作集 A , 模型状态转化概率矩阵 P , 即时奖励 R , 衰减因子 γ , 求解最优的状态价值函数 $v_{\pi^*}(s)$ 和最优策略 π^*

3. 马尔科夫过程是指未来状态仅由当前状态决定而与之前状态无关的过程，则可以记从状态 s 转移至状态 s' 的概率为 $P_{ss'}$ 。MDP 是当前强化学习理论推导的基础，通过 MDP，强化学习的交互流程可以很好地以概率论的形式表示出来，解决强化学习问题的关键定理也可以依此表示出来；

4. 马尔科夫决策过程属于动态规划过程，而强化学习又称拟动态规划，两者的区别在于动态规划假设 MDP 模型的参数为已知 (Model-based)，而强化学习可以使模型参数未知；

5. Bellman 公式建立了相邻状态 s, s' 的状态值函数 $v_\pi(s), v_\pi(s')$ 和动作值函数 $q_\pi(s, a), q_\pi(s', a')$ 之间的关系，Bellman 公式是 MDP 中最核心的内容，也是后续各种强化学习算法的基础。由 $v_\pi(s)$ 的定义：

$$\begin{aligned} v_\pi(s) &= E_\pi[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots | S_t = s] \\ &= E_\pi[R_t + \gamma(R_{t+1} + \gamma R_{t+2} + \dots) | S_t = s] \\ &= E_\pi[R_t | S_t = s] + \gamma E_\pi[R_{t+1} + \gamma R_{t+2} + \dots | S_t = s] \\ &= R_{s,\pi} + \gamma \sum_{s'} P_{ss'} v_\pi(s') = \sum_a \pi(a|s) \left(R_{s,\pi}^a + \gamma \sum_{s'} P_{ss'}^a v_\pi(s') \right) \end{aligned} \quad (28.1)$$

可以看到，策略 π 下状态 s 的估值由两部分组成：当前获得的奖励值及下一阶段所有可能状态的估值的加权和。假设当前环境一共有 n 种状态，则上式也可写成如下矩阵形式：

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} R(1) \\ \vdots \\ R(n) \end{bmatrix} + \gamma \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{11} & \dots & P_{1n} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} \quad (28.2)$$

同样地可以推导关于动作值函数的公式：

$$q_\pi(s, a) = R_{s,\pi}^a + \gamma \sum_{s'} P_{ss'}^a v_\pi(s') = R_{s,\pi}^a + \gamma \sum_{s'} P_{ss'}^a \sum_{a'} \pi(a'|s') q_\pi(s', a') \quad (28.3)$$

以上两式即是 Bellman 公式；

6. 动态规划问题的目标在于求解最优策略。策略迭代法 (policy iteration) 和值迭代法 (value iteration) 是两种经典的动态规划最优策略求解方法，两者均基于 Bellman 公式，以下将分别介绍。

¹要求行为策略必须包含目标策略。

28.2.1 策略迭代法 (policy iteration)

1. 策略迭代法的基本思路如下:
 - (a) 首先假定策略为 π , 计算在该策略下的状态值函数 $v_\pi(s)$, 该步骤称为策略评估 (policy evaluation);
 - (b) 基于计算的状态值函数 $v_\pi(s)$ 确定更优的策略 π^* , 该步骤称为策略提升 (policy improvement);
 - (c) 将更新的策略 π^* 回代入第一步, 多次循环直至策略收敛, 此时收敛的策略即为最优策略。
2. 策略评估中计算特定策略的状态值函数 $v_\pi(s)$ 的方法基于式 28.1 (28.2):
 - (a) 首先根据策略初始化状态转移概率矩阵 P , 并随机初始化各状态的值函数 $v_0(1), \dots, v_0(n)$;
 - (b) 将 $v_0(1), \dots, v_0(n)$ 代入式 28.2, 得到新的状态值函数 $v_1(1), \dots, v_1(n)$;
 - (c) 重复上一步直至状态值函数收敛, 整个循环中策略及转移概率矩阵均不变。
3. 策略提升中基于上一步得到的状态值函数 $v_\pi(s)$ 确定新策略的步骤如下:
 - (a) 根据当前策略 π 及概率转移矩阵 P , 将状态值函数代入式 28.3 得到动作值函数 $q_\pi(s, a)$, 新策略 π_1 由使得动作值函数最优的动作决定 (本质为贪心算法):

$$\pi_1 = \arg \max_a q_\pi(s, a) \quad (28.4)$$
 - (b) 将更新的策略 π_1 回代入第一步, 多次循环直至策略收敛为 π^* , 整个循环过程中状态值函数不变。
4. 策略迭代算法的最大缺点为: 需要使得某一策略的状态值函数收敛, 才能进行策略的更新, 这使得一个迭代结构内需要进行策略评估和策略提升两次迭代, 从而增加计算量。

28.2.2 值迭代法 (value iteration)

1. 值迭代法可以有效改进策略迭代的缺点。因为最优的策略必然使得值函数达到最优, 因此只需要使得值函数最优, 也就可以得到最优的策略;
2. 回归策略迭代中策略提升的过程, 将式 28.3 代入式 28.4 得到策略的优化方法:

$$\pi_1 = \arg \max_a \left(R_{s,\pi}^a + \gamma \sum_{s'} P_{ss'}^a v_\pi(s') \right)$$

因为策略与值函数同时达到最优, 从而得到值函数的更新公式 (式 28.5):

$$v_{\pi_1}(s) = \max_a \left(R_{s,\pi}^a + \gamma \sum_{s'} P_{ss'}^a v_\pi(s') \right) \quad (28.5)$$

3. 在值迭代过程中, 重复执行式 28.5 至状态值函数收敛, 此时状态值函数达到最优, 对应的策略也是最优。

28.3 蒙特卡洛法 (Monte Carlo method)

1. 蒙特卡洛法, 也叫统计模拟法, 由上世纪 40 年代美国数学家冯·诺依曼和乌拉姆为解决原子弹制造中的中子扩散问题而提出的一种以概率统计理论为指导的非常重要的数值计算方法。蒙特卡洛法使用随机数解决多种计算问题, 与之对应的是确定性计算方法。蒙特卡洛的核心思想是大数定律: 通过反复试验统计, 当统计次数足够多的时候频率值即收敛于概率值, 并用于替代模型中参数的真实值;
2. 采用前述的动态规划法得到最优策略的过程需要假定环境符合 MDP 模型, 为一种 model-based 算法, 但很多环境并不满足, 从而降低了动态规划法的适用性。蒙特卡洛通过大量统计直接估计某策略 π 下不同状态 s 的状态值函数 $v_\pi(s)$, 跳过了建模过程, 也不需要模型参数 $P_{ss'}$, 是一种 model-free 算法。蒙特卡洛法中对强化学习两个问题的定义如下:
 - (a) 预测问题, 给定强化学习的 5 个要素: 状态集 S , 动作集 A , 即时奖励 R , 衰减因子 γ 和策略 π , 求解该策略的状态价值函数 $v_\pi(s)$;
 - (b) 控制问题, 给定强化学习的 5 个要素: 状态集 S , 动作集 A , 即时奖励 R , 衰减因子 γ , 和探索率 ϵ (用于策略优化的参数, 见下文) 求解最优的动作值函数 ($q_{\pi^*}(s, a)$) 和最优策略 π^*

3. 蒙特卡洛法中得到策略 π 的状态值函数 $v_\pi(s)$ 的方法有首次访问蒙特卡洛策略估值 (first-visit Monte-Carlo Policy Evaluation) 和每次访问蒙特卡洛策略估值 (every-visit Monte-Carlo Policy Evaluation) 两种方法：

- 首次访问蒙特卡洛策略估值：设置计数器 $N(s)$ 和总体奖励值 G_t 累积函数 $S(s)$ 分别记录不同状态 s 出现的次数和总体奖励累计值，在每一次仿真得到的状态序列中，若状态 s 为第一次出现，则执行如下操作

$$N(s) = N(s) + 1 \quad S(s) = S(s) + G_t$$

大量重复试验得到大量的状态序列后，即可得到状态值函数的估值

$$v_\pi(s) = \frac{S(s)}{N(s)}$$

以上两式可以统一成下式，式中超参 $\alpha = \frac{1}{N(s)} = 0 \sim 1$ ，表示“记忆”周期， α 设置越大，表示周期越短，遗忘率越高，估值变化越快、越剧烈；

$$v_\pi(s) = v_\pi(s) + \alpha(G_t - v_\pi(s)) \quad (28.6)$$

- 每次访问蒙特卡洛策略估值：与前者非常相似，差别仅在于对一个状态序列中的每一个状态无论其是否出现过都进行计算。

4. 蒙特卡洛法中通过 $q_\pi(s, a)$ 更新策略 π 。与动态规划法中采用贪心算法不同，蒙特卡洛法采用 ε -贪心算法 (ε -greedy) 更新策略，更新流程如下：

- 采用与估算 $v_\pi(s)$ 相同的方法估算 $q_\pi(s, a)$ ；
- 设置探索率 $\varepsilon = 0 \sim 1$ ，以 $1 - \varepsilon$ 的概率贪婪地选择使得 $q_\pi(s, a)$ 最大的 a ，以 ε 的概率随机地选择 a ，从而实现决策的更新；
- 重复循环以上两步，直至决策收敛。在循环过程中，宜使得 ε 自大变小，即前期迭代时鼓励更多的探索，而后期迭代以贪心为主。

5. 蒙特卡洛法的缺点是：需要等到一个状态序列结束时才能估计每一状态的值函数，而无法没遇到一个状态就进行更新，因为总体奖励值 G_t 只有在一个状态序列结束后才能得到。

28.4 时间差分 (Time difference, TD)

蒙特卡洛法要求一个完整的状态序列结束后才能对策略进行更新，而时间差分法实现了策略与状态的同步更新。SARSA 算法和 Q-learning 算法是两类时间差分算法。

28.4.1 SARSA 算法

1. SARSA 算法是序列 $S \rightarrow A \rightarrow R \rightarrow S' \rightarrow A'$ 的缩写。SARSA 算法中更新动作值函数 $Q_\pi(s, a)$ 的算法为

$$Q_\pi(S, A) \leftarrow Q_\pi(S, A) + \alpha[R + \gamma Q_\pi(S', A') - Q_\pi(S, A)] \quad (28.7)$$

Algorithm 28.1 SARSA 算法流程

输入： α, γ ，初始化任意状态 S 和动作 A 的动作值函数 $Q(s, a)$ ，令任何一个终止状态的 Q 值都为 0

- 1: **for** 每个 Episode **do**
- 2: 设置初始状态 S
- 3: 根据 Q 及 S 选择动作 A (可按 ε -贪心算法)
- 4: **for** Step in Episode **do**
- 5: 完成动作 A ，观察奖励 R 及后续状态 S'
- 6: 根据 Q 及 S' 选择动作 A' (可按 ε -贪心算法)
- 7: 赋值： $Q_\pi(S, A) \leftarrow Q_\pi(S, A) + \alpha[R + \gamma Q_\pi(S', A') - Q_\pi(S, A)]$
- 8: 赋值： $S \leftarrow S', A \leftarrow A'$
- 9: **end for**
- 10: **end for**

2. SARSA 算法是一种 on-policy 算法，而 ε -贪心算法中动态 ε 的设置可以改善 on-policy 算法面临的探索-利用矛盾。

28.4.2 Q-Learning 算法

1. Q-Learning 算法是强化学习中最有效、最具普适性的算法，其收敛速度往往快于 SARSA 算法。与 SARSA 算法不同，Q-Learning 算法是一种 off-policy 算法。两者的算法流程基本相似，仅在动作值函数 $Q(s, a)$ 的更新上存在差异

$$Q_{\pi}(S, A) \leftarrow Q_{\pi}(S, A) + \alpha[R + \gamma \max_{a'} Q_{\pi}(S', a') - Q_{\pi}(S, A)] \quad (28.8)$$

2. Q-Learning 算法中，选择当前状态 S 后所能转移到的所有状态 S' 可能做的所有动作 a' 中的最大动作值函数为当前动作 $Q(s, a)$ 的估值，这一算法可以加速收敛，但会过高地估计了当前动作的估值，造成过估计 (overestimating)，并且过估计的严重程度会随动作空间的增加而增加，这是 Q-Learning 算法的一大缺点；
3. Q-Learning 算法的另一缺点是几乎无法实现输出动作为连续值的场景（这一问题同样存在于 SARSA 算法中），这是因为当动作为连续值是很难遍历每一个动作的值函数，从而难以进行动作值函数的更新。

28.5 深度强化学习简介

1. 前述的所有方法均为传统强化学习方法，适用于处理较低维度的状态信息，而在实际问题中，随着期望处理的状态信息的复杂化（如一整张图片、一个连续的视频流）时传统的方法即显得捉襟见肘，而深度学习则能很好的处理这一情况，因此将深度学习方法应用于强化学习过程中，即深度强化学习；
2. 深度学习是指基于深度神经网络的机器学习解决方案，基于深度神经网络的优势，模型往往能取得过人的表现能力，具体表现为较强的对分类、回归问题的拟合能力和较高的泛化能力。相较于传统的机器学习方法，深度学习的优势主要表现在：
- 降低特征提取门槛。在采用传统机器学习方法进行分类回归时，需要首先人工地选择并提取数据特征，特征中可能出现的噪点、多重共线性等问题都会对后续机器学习的效果产生负面影响。而在深度学习时，由于大量线性分类器的堆积、卷积神经网络的应用和对噪声的忍耐能力，降低了对数据特征提取的要求；
 - 可有效处理线性不可分问题。
3. 深度学习方法无法解决强化学习的建模问题，也无法提升强化学习的效果，而是在给出状态 s 和值函数 $Q(s, a)$ 后借助神经网络实现 $s \rightarrow Q(s, a)$ 的映射关系，在经典强化学习方法中，这一过程往往是通过数据库索引等方法实现的，效率较低。

28.6 Deep Q Network (DQN) 算法

28.6.1 NIPS DQN 与 Nature DQN

1. Deep Q Network (DQN) 算法是最早出现的一类深度强化学习算法，是深度学习与 Q-Learning 算法的结合，于 2013 年最早被 DeepMind 团队提出，又在 2015 年进行重大改进并发表于 Nature。上述两个 DQN 版本是学术界公认的最早的使用深度学习技术解决强化学习问题的手段，它们也各自以其最初发表的刊物（会议）名称被命名为 NIPS DQN 与 Nature DQN；
2. Q-Learning 算法的核心是 $Q(S, A)$ 的更新算法（式 28.8），算法需要维护一个巨大的 $Q(S, A)$ 数据库以查询 $Q_{\pi}(s, a)$ 和 $\max_{a'} Q_{\pi}(s', a')$ 。DQN 算法的思路是以神经网络替代数据库，即维护一神经网络，输入状态 S 即可输入动作值函数矩阵 $Q(s, A; \theta) = \{Q(S = s, A = a_1; \theta), \dots, Q(S = s, A = a_n; \theta)\}$ ，式中 θ 为神经网络的参数。神经网络的引入使得 DQN 可以处理更大的状态空间，因此可以将整幅图像作为状态输入，但动作空间的大小依然受限。NIPS DQN 算法是其它改进 DQN 算法的基础；

Algorithm 28.2 NIPS DQN 算法流程

输入: episode 迭代次数 E 、批数据尺寸 M 、学习率 LR 、衰减因子 γ 、状态数 NS 和动作数 NA , 初始化记忆矩阵 D 并指定矩阵最大容量 N , 初始化神经网络 Q_θ 及其网络参数 θ

- 1: **for** episode in E **do**
- 2: 设置初始状态 s_t
- 3: **for** step in episode **do**
- 4: 赋值: $Q(s_t, A; \theta) \leftarrow Q_\theta(s_t)$
- 5: 赋值: $a_t \leftarrow \arg \max_A Q(s_t, A; \theta)$ (ε -greedy)
- 6: 根据 s_t, a_t 计算奖励 r_t 及下一状态 s_{t+1} , 并将序列 $[s_t, a_t, r_t, s_{t+1}]$ 存入 D (超出最大容量时替换旧序列)
- 7: 自 D 随机提取批序列 mini-batch
- 8: **for** $[s_i, a_i, r_i, s_{i+1}]$ in mini-batch **do**
- 9: 赋值: $Q(s_i, A; \theta) \leftarrow Q_\theta(s_i)$
- 10: 赋值: $targetQ(s_i, A; \theta) \leftarrow r_i$ **if** s_i is terminal **else** $r_i + \gamma \max_A Q(s_{i+1}, A; \theta)$
- 11: 赋值: $loss \leftarrow [targetQ(s_i, A; \theta) - Q(s_i, A; \theta)]^2$
- 12: 误差反向传递更新神经网络
- 13: **end for**
- 14: 赋值: $s_t \leftarrow s_{t+1}$
- 15: **end for**
- 16: **end for**

3. 对比 NIPS DQN 与 Q-Learning 算法, 一个明显地不同在于略去了遗忘率 α , 这是因为 DQN 算法中引入记忆矩阵 Replay Memory (D), 矩阵保存每一组 $[s_t, a_t, r_t, s_{t+1}]$ Transition 序列, 且矩阵的容量为定值, 新的序列会替换掉矩阵中最老的序列, 从而起到遗忘的效果;
4. 另外, Replay Memory 为随机抽取 mini batch 提供了可能。神经网络的学习过程本质上是监督学习的过程, 而监督学习要求样本之间互不相关, 然而传统的强化学习所产生的序列往往具有较强的相关性, **Replay Memory** 的引入即可缓和样本的相关性, 从而提升了模型的收敛和泛化能力;
5. NIPS DQN 算法中只有一个网络, 这一网络同时用于 $Q(s_t, A; \theta)$ 及其标签值 $targetQ(s_t, A; \theta)$ 的估值, 并每一次基于误差对网络参数 θ 进行更新。然而在监督学习中, 一般标签均为定值, 而 **NIPS DQN** 算法中因为神经网络一直变化也就使得对标签的估值也一直变换, 不利于算法收敛;
6. Nature DQN 算法是在 NIPS DQN 的基础新增了目标网络 (Target Network), 专门用于目标值的估值, 而原有的网络称为主网络 (Main Network)。两个网络具有完全相同的结构, 且初始化时的参数也完全相同。主网络对动作值函数进行估值, 并基于其与目标值的误差进行更新。目标网络不主动进行更新, 而是每一定间隔以主网络的参数进行更新。目标网络的引入使得值函数目标值具有更高的稳定性, 提升了主网络的收敛速度。

28.6.2 Double DQN

1. Double DQN 的概念同样提出于 2015 年, 其原型为 2010 年提出的 Double Q-Learning。Q-Learning 算法因为动作估值中 max 的存在会产生普遍的过估计问题, 以之为基础的 DQN 算法也存在同样的问题。而 Double Q-Learning 与 Double DQN 则调整了动作估值的算法, 有助于缓解过估计的情况;
2. Double DQN 中的“Double”指模型中存在两个结构相同的神经网络 Q^A, Q^B , 对任意状态 s , 两神经网络均可输出对不同动作的估值 $Q(s, A; \theta^A), Q(s, A; \theta^B)$ 。算法的精髓部分在于动作值函数标签值 $targetQ(s_i, A; \theta)$ 的更新过程, 将动作选择和标签值的估算两步解耦:

$$\begin{cases} targetQ(s_i, A; \theta^A) = r_i + \gamma Q(s_i, a^*; \theta^B), & a^* = \arg \max_A Q(s_{i+1}, A; \theta^A) \\ targetQ(s_i, A; \theta^B) = r_i + \gamma Q(s_i, a^*; \theta^A), & a^* = \arg \max_A Q(s_{i+1}, A; \theta^B) \end{cases}$$

上式中的第一道表示更新网络 Q^A 、第二道表示更新网络 Q^B , 对每一步循环, 只需随机更新其中一个网络;

3. 对于状态 s , 为得到序列 $[s, a, r, s']$ 需要进行动作选择。此时动作将基于另一个未被更新的网络的值函数估值进行选择。假设上一步更新网络 Q^A , 则 $a = \arg \max_A Q(s_i, A; \theta^B)$;
4. 以上即是 Double DQN 的特点: 在学习估值的时候用 \max 进行评估, 在选择输出动作时用另一个没有做 \max 评估的权重集合进行选择。

28.6.3 Dueling DQN

1. Dueling DQN 提出于 2016 年, 同样是针对 DQN 算法的改进, 算法对神经网络的结构进行调整: 与大多数 DQN 算法的神经网络直接输出动作值函数 $Q(s, a; \theta)$ 不同, Dueling DQN 对 $Q(s, a; \theta)$ 作出如下分解:

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + A(s, a; \theta, \alpha)$$

即认为动作值函数由状态值函数 $V(s; \theta, \beta)$ 与动作优势值函数 $A(s, a; \theta, \alpha)$ 构成。神经网络首先独立评估 $V(s; \theta, \beta)$ 与 $A(s, a; \theta, \alpha)$, 再输出 $Q(s, a; \theta, \alpha, \beta)$;

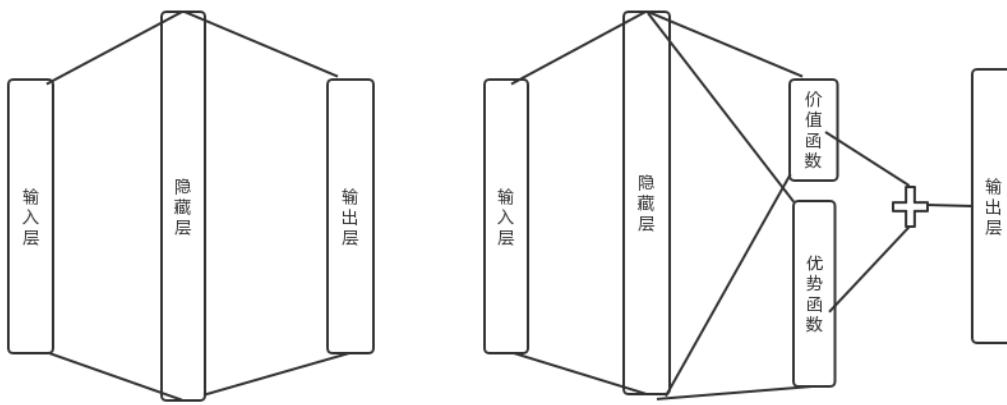


图 28.1 Dueling DQN 与一般 DQN 的对比

2. 以上调整非常符合实际: 存在一些极好的状态, 此时无论做什么动作都无伤大雅; 而对于某些较差的状态, 合适的动作选择即显得尤为重要。通过这种巧妙的调整, 实现动作本身价值和状态本身价值的解耦;
3. 然而如果直接采用上式会使得学习效率偏差。因为神经网络优化的过程也是动作值函数 $Q(s, a; \theta, \alpha, \beta)$ 收敛的过程, 然而依靠收敛的 $Q(s, a; \theta, \alpha, \beta)$ 并不能反向确定 $V(s; \theta, \beta)$ 和 $A(s, a; \theta, \alpha)$ 的估值, 这是由于缺少约束决定的, 这一问题称为低辨识度问题 (unidentifiable);
4. 为解决上述问题, 需要对公式人为进行调整, 本质上是增加一个约束:

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left[A(s, a; \theta, \alpha) - \max_{a'} A(s, a'; \theta, \alpha) \right]$$

上式强制指定选择的最优的动作的优势值函数为 0, 即可由动作值函数的估值计算状态值函数的估值 $V(s; \theta, \beta) = Q(s, a^*; \theta, \alpha, \beta)$, $a^* = \arg \max_{a'} A(s, a'; \theta, \alpha)$, 进一步也能得到动作优势值函数的估值。除了减去最大动作优势值函数, 也可以减去平均动作优势值函数, 两者具有相似的效果:

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left[A(s, a; \theta, \alpha) - \frac{1}{N} \sum_{a'} A(s, a'; \theta, \alpha) \right]$$

5. 经过调整后, 虽然状态值函数与动作优势值函数的定义发生改变, 但本质含义并未改变, 且增加了识别性。

28.6.4 优先回放 DQN (Priority Replay DQN)

1. “优先经验回放” (priority experience replay) 的概念同样提出于 2016 年, 这一概念即可应用于 DQN, 也可用于 Q-Learning 中, 是一种旨在提高模型学习速率、加速收敛的算法;

2. 在一般 DQN 算法中, $[s, a, r, s']$ 序列储存在记忆矩阵 replay memory 中, 神经网络训练时随机且均匀地从矩阵中抽取批序列进行训练。训练的过程实际上是使得 $Q(s, a)$ 收敛的过程, 在学习过程中必然存在部分 $Q(s, a)$ 收敛的更快而其它则更慢的现象, 此时为了提高学习效率可以更关注那些欠收敛 $Q(s, a)$ 的学习, 即是优先回放 DQN 的基本思路;
3. 在优先回放 DQN 中, 批序列的抽取不再服从均匀分布, 而是对每一序列 i 引入优先级权重 w_i , 权重越大序列被抽取的概率也就越大, 而这一权重也被用于损失函数 J 的计算中, 使得神经网络误差反向传递时会更关注权重高的样本

$$J = \frac{1}{N} \sum_i w_i [target Q(s_i, A; \theta) - Q(s_i, A; \theta)]^2$$

4. 以上即是优先回放 DQN 的主要特点, 其余部分与其它 DQN 算法一致。

28.7 深度确定性策略梯度法 (*Deep deterministic policy gradient, DDPG*)

28.7.1 随机策略梯度 (stochastic policy gradient, SPG)

1. 前述的蒙特卡洛、SARSA、Q-Learning、DQN 等方法都是通过计算动作值函数 $q(s, a)$ 的方法获取最优策略, 即 value-based 方法。尽管神经网络的引入极大地改善了学习效果, 但无法避免 value-based 的固有问题 (P 476), 其中最主要的是动作空间受限;
2. 策略梯度法是一类 policy-based 方法, 算法直接对策略 $\pi(s|\theta)$ 的评价函数进行优化, 对任意状态 s 可直接输出动作 a 。策略梯度得名于在优化评价函数时, 需要计算其对策略的参数 θ 的梯度以实现梯度上升。根据策略的随机性或确定性, 策略梯度又分为随机策略梯度 (SPG) 和确定性策略梯度 (DPG):
 - 随机性策略: 给定动作 s 和参数 θ , 输出每一动作的概率 $\pi_\theta(a|s) = P[a|s, \theta]$;
 - 确定性策略: 给定动作 s 和参数 θ , 输出特定的动作 $a = \mu_\theta(s)$ 。
3. 首先以随机策略为例详细介绍策略梯度的推导。假设在策略 π_θ 下智能体经历了一整个 episode, 得到一个状态序列 (或轨迹) $\tau = (s_0, a_0, r_0, \dots, s_t, a_t, r_t)$, 显然对该策略的评价 $J(\theta)$ 应该与其得到的所有奖励值有关, 可表示如下

$$J(\theta) = E[r_0 + r_1 + \dots + r_n | \pi_\theta] = E_{\tau \sim p_\theta(\tau)} \left[\sum_t r(s_t, a_t) \right] = E_{\tau \sim p_\theta(\tau)} [r(\tau)]$$

式中 $\tau \sim p_\theta(\tau)$ 表示轨迹 τ 服从概率分布 $p_\theta(\tau)$, θ 为参数; $r(\tau) = \sum_t r(s_t, a_t)$ 。假设 τ 的分布函数 $p_\theta(\tau)$ 是可微的, 则根据期望的定义, 评价函数 $J(\theta)$ 可以进一步表示为积分形式, 从而求微分

$$J(\theta) = \int p_\theta(\tau) r(\tau) d\tau \implies \nabla_\theta J(\theta) = \int \nabla_\theta p_\theta(\tau) r(\tau) d\tau, \quad \nabla_\theta = \left[\frac{\partial}{\partial \theta_1}, \dots, \frac{\partial}{\partial \theta_n} \right]^T$$

又因为 $\nabla_\theta p_\theta(\tau) = p_\theta(\tau) \frac{\nabla_\theta p_\theta(\tau)}{p_\theta(\tau)} = p_\theta(\tau) \nabla_\theta \ln p_\theta(\tau)$, 代入上式得

$$\nabla_\theta J(\theta) = \int p_\theta(\tau) \nabla_\theta \ln p_\theta(\tau) r(\tau) d\tau = E_{\tau \sim p_\theta(\tau)} [\nabla_\theta \ln p_\theta(\tau) r(\tau)]$$

从而策略评价函数的梯度 $\nabla_\theta J(\theta)$ 也可以表示为数学期望的形式, 也就可以不计算解析解而是采用抽样平均的方法计算。进一步地, 以更为方便得到的 $\pi_\theta(a|s)$ 计算 $\nabla_\theta \ln p_\theta(\tau)$

$$\begin{aligned} \because p_\theta(\tau) &= p(s_1) \prod_{t=1}^T \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t) \\ \therefore \ln p_\theta(\tau) &= \ln p(s_1) + \sum_{t=1}^T [\ln \pi_\theta(a_t|s_t) + \ln p(s_{t+1}|s_t, a_t)] \\ \therefore \nabla_\theta \ln p_\theta(\tau) &= \sum_{t=1}^T \nabla_\theta \ln \pi_\theta(a_t|s_t) \end{aligned}$$

$$\therefore \nabla_{\theta} J(\theta) = E_{\tau \sim p_{\theta}(\tau)} \left[\left(\sum_{t=1}^T \nabla_{\theta} \ln \pi_{\theta}(a_t | s_t) \right) r(\tau) \right] = \frac{1}{N} \sum_{i=1}^N \left[\left(\sum_{t=1}^T \nabla_{\theta} \ln \pi_{\theta}(a_{i,t} | s_{i,t}) \right) \left(\sum_{t=1}^T r(s_{i,t}, a_{i,t}) \right) \right]$$

4. 对上式还可做出以下改进：

- 引入因果关系。上式中存在两个并列求和项相乘的结构 $[\sum_{t=1}^T \nabla_{\theta} \ln \pi_{\theta}(a_{i,t} | s_{i,t})][\sum_{t=1}^T r(s_{i,t}, a_{i,t})]$, 而这一结构并不符合因果关系： t 时刻所做的策略 π_{θ} 应该只与 $t' \geq t$ 时刻的奖励值 r 有关；
- 考虑到奖励随时间的衰减，可引入衰减因子 γ 。

经过调整后得到最终的随机策略的评价函数梯度公式，策略参数 θ 的更新公式即为 $\theta = \theta + \alpha \nabla_{\theta} J(\theta)$

$$\begin{aligned} \nabla_{\theta} J(\theta) &= \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \left[\nabla_{\theta} \ln \pi_{\theta}(a_{i,t} | s_{i,t}) \left(\sum_{t'=t}^T \gamma^{t-t'} r(s_{i,t'}, a_{i,t'}) \right) \right] = \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \ln \pi_{\theta}(a_{i,t} | s_{i,t}) Q(s_{i,t}, a_{i,t}) \\ &= \int_S \rho^{\pi}(s) \int_A \nabla_{\theta} \pi_{\theta}(a | s) Q(s, a) da ds = E_{s \sim \rho^{\pi}, a \sim \pi_{\theta}} [\nabla_{\theta} \ln \pi_{\theta}(a | s) Q(s, a)] \end{aligned} \quad (28.9)$$

5. 以上 SPG 的推导公式早在 1999 年即由 Sutton 给出，而在很长时间里 DPG 的公式均被认为不存在，直至 2014 年被证明²。因为 SPG 需要给出每一动作的概率，当动作空间极大时效率极低，所以 DPG 的提出非常关键。

28.7.2 确定性策略梯度 (deterministic policy gradient, DPG)

1. 在给出 DPG 的详细推导前，首先定义状态 s' 在策略 μ_{θ} 下发生的概率为 $\rho^{\mu_{\theta}}(s')$ ，而 $\rho^{\mu_{\theta}}(s')$ 可做如下分解

$$\rho^{\mu_{\theta}}(s') = \int_S \sum_{t=0}^{\infty} \gamma^{t-1} p_1(s) p(s \rightarrow s', t, \mu_{\theta}) ds$$

式中 γ 为衰减系数； $p_1(s)$ 表示初状态为 s 的概率； $p(s \rightarrow s', t, \mu_{\theta})$ 表示在策略 μ_{θ} 下自状态 s 经过 t 次转变至状态 s' 的概率；

2. 随后给出 DPG 的详细推导。记策略 μ_{θ} 的评价函数为 $J(\theta)$

$$J(\theta) = \int_S p_1(s) Q(s, \mu_{\theta}(s)) ds = E_{s \sim \rho^{\mu}} [Q(s, \mu_{\theta}(s))]$$

因为为确定性策略，在给定的 s 下即有唯一的 a ，所以上式无需对 a 求积分，且有 $Q(s, \mu_{\theta}(s)) = V_{\mu}(s)$ 。易知 $\nabla_{\theta} J(\theta) = \int_S p_1(s) \nabla_{\theta} Q(s, \mu_{\theta}(s)) ds$ ，所以关键在于计算 $\nabla_{\theta} Q(s, \mu_{\theta}(s))$ 。根据 Bellman 公式（式 28.3）

$$\begin{aligned} \nabla_{\theta} Q(s, \mu_{\theta}(s)) &= \nabla_{\theta} \left[r(s, \mu_{\theta}(s)) + \int_S \gamma p(s' | s, \mu_{\theta}(s)) V_{\mu}(s') ds' \right] \\ &= \nabla_{\theta} \left[r(s, \mu_{\theta}(s)) + \int_S \gamma p(s' | s, \mu_{\theta}(s)) Q(s', \mu_{\theta}(s')) ds' \right] \\ &= \nabla_{\theta} \mu_{\theta}(s) \nabla_a r(s, a) \Big|_{a=\mu_{\theta}(s)} + \int_S \gamma \left[\nabla_{\theta} \mu_{\theta}(s) \nabla_a p(s' | s, a) \Big|_{a=\mu_{\theta}(s)} Q(s', \mu_{\theta}(s')) + p(s' | s, \mu_{\theta}(s)) \nabla_{\theta} Q(s', \mu_{\theta}(s')) \right] ds' \\ &= \nabla_{\theta} \mu_{\theta}(s) \nabla_a \left[r(s, a) + \int_S \gamma p(s' | s, a) V_{\mu}(s') ds' \right] \Big|_{a=\mu_{\theta}(s)} + \int_S \gamma p(s' | s, \mu_{\theta}(s)) \nabla_{\theta} Q(s', \mu_{\theta}(s')) ds' \\ &= \nabla_{\theta} \mu_{\theta}(s) \nabla_a Q(s, a) \Big|_{a=\mu_{\theta}(s)} + \int_S \gamma p(s \rightarrow s', 1, \mu_{\theta}) \nabla_{\theta} Q(s', \mu_{\theta}(s')) ds' \end{aligned}$$

以上给出了 $\nabla_{\theta} Q(s, \mu_{\theta}(s))$ 与 $\nabla_{\theta} Q(s', \mu_{\theta}(s'))$ 的递推式。注意到

$$\int_S \gamma p(s \rightarrow s', 1, \mu_{\theta}) \int_S \gamma p(s' \rightarrow s'', 1, \mu_{\theta}) \nabla_{\theta} Q(s'', \mu_{\theta}(s'')) ds'' ds' = \int_S \gamma^2 p(s \rightarrow s', 2, \mu_{\theta}) \nabla_{\theta} Q(s', \mu_{\theta}(s')) ds'$$

按状态序列 $s \rightarrow s' \rightarrow s'' \rightarrow \dots$ 迭代上述递推式得

$$\nabla_{\theta} Q(s, \mu_{\theta}(s)) = \nabla_{\theta} \mu_{\theta}(s) \nabla_a Q(s, a) \Big|_{a=\mu_{\theta}(s)} + \int_S \gamma p(s \rightarrow s', 1, \mu_{\theta}) \nabla_{\theta} Q(s', \mu_{\theta}(s')) ds'$$

²Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]// 2014.<http://proceedings.mlr.press/v32/silver14.pdf> (附录：<https://www.taodocs.com/p-50316246.html>)

$$\begin{aligned}
&= \nabla_{\theta} \mu_{\theta}(s) \nabla_a Q(s, a) \Big|_{a=\mu_{\theta}(s)} \\
&\quad + \int_S \gamma p(s \rightarrow s', 1, \mu_{\theta}) \nabla_{\theta} \mu_{\theta}(s') \nabla_a Q(s', a) \Big|_{a=\mu_{\theta}(s')} ds' \\
&\quad + \int_S \gamma^2 p(s \rightarrow s', 2, \mu_{\theta}) \nabla_{\theta} Q(s', \mu_{\theta}(s')) ds' \\
&= \nabla_{\theta} \mu_{\theta}(s) \nabla_a Q(s, a) \Big|_{a=\mu_{\theta}(s)} + \int_S \sum_{t=1} \gamma^t p(s \rightarrow s', t, \mu_{\theta}) \nabla_{\theta} \mu_{\theta}(s') \nabla_a Q(s', a) \Big|_{a=\mu_{\theta}(s')} ds' \\
&= \int_S \sum_{t=0} \gamma^{t-1} p(s \rightarrow s', t, \mu_{\theta}) \nabla_{\theta} \mu_{\theta}(s') \nabla_a Q(s', a) \Big|_{a=\mu_{\theta}(s')} ds'
\end{aligned}$$

以上即是动作值函数对参数 θ 的梯度，代入评估函数的梯度 $\nabla_{\theta} J(\theta)$

$$\begin{aligned}
\nabla_{\theta} J(\theta) &= \int_S p_1(s) \nabla_{\theta} Q(s, \mu_{\theta}(s)) ds \\
&= \int_S p_1(s) \int_S \sum_{t=0} \gamma^{t-1} p(s \rightarrow s', t, \mu_{\theta}) \nabla_{\theta} \mu_{\theta}(s') \nabla_a Q(s', a) \Big|_{a=\mu_{\theta}(s')} ds' ds \\
&= \int_S \nabla_{\theta} \mu_{\theta}(s') \nabla_a Q(s', a) \Big|_{a=\mu_{\theta}(s')} \int_S p_1(s) \sum_{t=0} \gamma^{t-1} p(s \rightarrow s', t, \mu_{\theta}) ds' ds \\
&= \int_S \rho^{\mu_{\theta}}(s) \nabla_{\theta} \mu_{\theta}(s) \nabla_a Q(s, a) \Big|_{a=\mu_{\theta}(s)} ds = E_{s \sim \rho^{\mu}} \left[\nabla_{\theta} \mu_{\theta}(s) \nabla_a Q(s, a) \Big|_{a=\mu_{\theta}(s)} \right] \quad (28.10)
\end{aligned}$$

28.7.3 深度确定性策略梯度 (deep deterministic policy gradient, DDPG)

- 顾名思义，DDPG 即是将深度学习的方法与策略梯度法结合。其中连续性策略梯度是为了解决连续动作输出问题，又为了解决策略梯度法更新速度慢的问题而采用 A-C 模式，其中基于 s 输出 a 的过程（即 actor，表示策略）和基于 s, a 输出 $q(s, a)$ 的过程（即 critic）分别由两个神经网络 $\mu(s|\theta^{\mu})$, $Q(s, a|\theta^Q)$ 完成，另外设置对应的两个目标网络 $\mu'(s|\theta^{\mu'})$, $Q'(s, a|\theta^{Q'})$ 实现平稳更新；

Algorithm 28.3 DDPG 算法流程

输入：episode 迭代次数 E 、批数据尺寸 M 、学习率 LR 、衰减因子 γ 、更新率 τ 、状态数 NS 和动作数 NA ，初始化记忆矩阵 D 并指定矩阵最大容量 N ，初始化神经网络 $\mu(s|\theta^{\mu})$, $Q(s, a|\theta^Q)$, $\mu'(s|\theta^{\mu'})$, $Q'(s, a|\theta^{Q'})$

- for** episode in E **do**
- 设置初始状态 s
- 初始化噪声 n
- for** step in episode **do**
- 执行 actor 环节：根据噪声及当前策略得到动作 $a_t = \mu(s_t|\theta^{\mu}) + n_t$
- 根据 s_t, a_t 计算奖励 r_t 及下一状态 s_{t+1} ，并将序列 $[s_t, a_t, r_t, s_{t+1}]$ 存入 D （超出最大容量时替换旧序列）
- 自 D 随机提取批序列 mini-batch
- 执行 critic 环节：根据批序列中的 s_i, a_i 由主网络计算值函数 $Q(s_i, a_i|\theta^Q)$
- 根据批序列中的 r_i, s_{i+1} 由目标网络计算值函数的目标值 $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
- 计算 critic 环节误差并更新网络参数 θ^Q ：

$$L_Q = \frac{1}{M} \sum_i [y_i - Q(s_i, a_i|\theta^Q)]^2, \quad \theta^Q = \theta^Q - LR \cdot \nabla_{\theta^Q} L_Q$$

- 计算 actor 环节效果并更新网络参数 θ^{μ} ：

$$J_{\mu} = \frac{1}{M} \sum_i Q(s_i, \mu(s_i|\theta^{\mu})|\theta^Q), \quad \nabla_{\theta^{\mu}} J_{\mu} = \frac{1}{M} \sum_i \nabla_a Q(s_i, a|\theta^Q) \Big|_{a=\mu(s_i|\theta^{\mu})} \nabla_{\theta^{\mu}} \mu(s_i|\theta^{\mu}), \quad \theta^{\mu} = \theta^{\mu} + LR \cdot \nabla_{\theta^{\mu}} L_Q$$

- 对目标网络参数进行软更新 (soft replacement)

$$\theta^{Q'} = \tau \theta^Q + (1 - \tau) \theta^{Q'}, \quad \theta^{\mu'} = \tau \theta^{\mu} + (1 - \tau) \theta^{\mu'}$$

- end for**

- end for**
-

2. 除了 A-C 模式的引入外，上述 DDPG 伪代码同前述其它 value-based 方法（特别是 DQN）还存在以下差异：

- 引入自适应噪声替代 ε -greedy。因为 DDPG 采用确定性策略，策略直接输出动作而非动作概率，因此不存在采用 greedy 算法的可能性，又因为动作是连续的，为了提升学习的探索效果在输出动作后加上噪声项。噪声服从均值为 0 的正态分布，随着学习的深入方差逐渐减小，起到前期探索后期贪心的效果；
- 引入更新率 τ 对目标网络进行软更新 (**soft replacement**) 而非按频率更新。Nature DQN 中引入目标网络是，目标网络不随主网络变换，而是每迭代一定次数才更新一次，从而改善了主网络学习的稳定性。而 DDQG 中引入更新率 τ 对目标网络进行更新，目标网络随主网络而更新 ($\theta' = \tau\theta + (1-\tau)\theta'$)， τ 一般较小（如 0.001），使得目标网络每次的更新都比较小，同样保证了主网络学习的稳定性。

28.8 异步优势 AC 算法 (**Asynchronous advantage actor-critic, A3C**)

1. 顾名思义，异步优势 AC 算法 (A3C) 是 AC 算法的改进，由 asynchronous、advantage 和 actor-critic 三部分组成。A3C 算法与 AC 算法一样均为 policy-based 和 value-based 的结合，对以上两类方法（如 Q-Learning 和 PG）均可适用。actor-critic 模式已于 P 476 介绍，以下将先后讨论 advantage 和 asynchronous 的含义；
2. advantage 即为 Dueling DQN 中介绍的动作优势值函数 $A(s, a) = Q(s, a) - V(s)$ ， $A(s, a)$ 的引入实现了动作本身价值和状态本身价值的解耦。以 SPG 为例，在 AC 模式中，actor 环节的参数更新梯度见式 28.9，而 critic 环节的误差即是其估值结果同 Bellman 公式的误差

$$\text{loss} = \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T_n} \left[r_t^n + \gamma \max_{a_{t+1}^n} Q^{\pi_\theta}(s_{t+1}^n, a_{t+1}^n) - Q^{\pi_\theta}(s_t^n, a_t^n) \right]^2$$

可以看出无论是 actor 还是 critic，其参数更新均是基于 $Q(s, a)$ ，**advantage actor-critic (A2C)** 即是以 $A(s, a)$ 替代 $Q(s, a)$ ，可以理解为为 $Q(s, a)$ 增加一基线，从而得到新的 actor 和 critic 的参数更新公式

$$\begin{cases} \nabla_\theta J(\theta) = \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^T \nabla_\theta \ln \pi_\theta(a_t^n | s_t^n) \left[Q^{\pi_\theta}(s_t^n, a_t^n) - V^{\pi_\theta}(s_t^n) \right] \\ \text{loss} = \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T_n} \left[r_t^n + \gamma \max_{a_{t+1}^n} A^{\pi_\theta}(s_{t+1}^n, a_{t+1}^n) - A^{\pi_\theta}(s_t^n, a_t^n) \right]^2 \end{cases}$$

可以看到上式需要同时对 $Q(s, a)$ 和 $V(s)$ 估值，即需要训练两套神经网络，增加了计算量，因此可代入 Bellman 公式 $Q(s, a) = r + \gamma V(s')$ (式 28.3)，得到 A2C 的参数更新公式

$$\begin{cases} \nabla_\theta J(\theta) = \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^T \nabla_\theta \ln \pi_\theta(a_t^n | s_t^n) \left[r_t^n + \gamma V^{\pi_\theta}(s_{t+1}^n) - V^{\pi_\theta}(s_t^n) \right] \\ \text{loss} = \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T_n} \left[r_t^n + \gamma V^{\pi_\theta}(s_{t+1}^n) - V^{\pi_\theta}(s_t^n) \right]^2 \end{cases} \quad (28.11)$$



n-step return

n-step return 的概念来自于 Bellman 公式。在前述所有强化学习算法中，评估函数对值函数 (q, v, a) 的估值都是仅利用当前步和下一步的信息，即 1-step return，而实际上根据值函数的定义，可以利用更多步的信息，即 n-step return。

$$(1\text{-step return}): V(s_0) = r + \gamma V(s_1) \implies (n\text{-step return}): V(s_0) = r + \gamma r_1 + \cdots \gamma^{n-1} r_{n-1} + \gamma^n V(s_n)$$

n-step return 的优点在于增大了误差函数中奖励值（即真值）的权重，从而放大了评估函数的误差，有助于评估函数的学习。但计算时所包含的步骤越多也就意味着方差越大，这反而不利于评估函数收敛。

3. A3C 是在 A2C 的基础上加上 asynchronous (“异步的”), 其思路是希望通过多个进程或线程的 agent，在不同策略的指导下不断在环境中试探，从而“博采众长”对策略进行更新的方法。多进程的应用最直接的优点即是可以实现效率的提升；

4. A3C 相比于 A2C 的另一个小优化点是在更新 actor 的参数时在策略梯度的基础上增加了策略 π 的熵项 $H(\pi(s))$ (式 23.2), 系数为 c_{reg} 。策略 π 的熵越大, 意味着其所包含的不确定性信息越多、策略的随机性越大 (P 365), 因此在更新 actor 参数时考虑策略熵起到了增加策略随机性和探索效果的作用;

$$\theta = \theta + \alpha \nabla_{\theta} [J(\theta) + c_{reg} H(\pi(s))] = \theta + \alpha \nabla_{\theta} \frac{1}{n} \sum_{i=1}^n [A(s_i, a_i) \ln \pi(s_i, a_i) - c_{reg} \pi(s_i, a_i) \ln \pi(s_i, a_i)]$$

Algorithm 28.4 A3C 算法中单个进程的算法流程

```

输入: 全局 AC 网络参数  $\theta_a, \theta_c$ ; 全局迭代轮数  $T$  和全局最大迭代轮数  $T_{max}$ 
输入: 进程 AC 网络参数  $\theta'_a, \theta'_c$ ; 进程内单次迭代时间序列的最大轮数  $t_{max}$ 
1: 初始化进程内单次迭代时间序列轮数  $t \leftarrow 1$ 
2: repeat
3:   重置全局 AC 网络参数梯度更新量  $d\theta_a \leftarrow 0, d\theta_c \leftarrow 0$ 
4:   同步进程 AC 网络参数  $\theta'_a \leftarrow \theta_a, \theta'_c \leftarrow \theta_c$ 
5:    $t_{start} \leftarrow t$ 
6:   获得状态  $s_t$ 
7:   repeat
8:     基于策略  $\pi(a_t|s_t, \theta'_a)$  得到动作  $a_t$ 
9:     观察奖励值  $r_t$  和新状态  $s_{t+1}$ 
10:     $t \leftarrow t + 1, T \leftarrow T + 1$ 
11:    until  $s_t$  为终止状态 or  $t - t_{start} == t_{max}$ 
12:     $R = \begin{cases} 0 & \text{terminal } s_t \\ V(s_t|\theta'_c) & \text{else} \end{cases}$ 
13:    for  $i \in \{t - 1, \dots, t_{start}\}$  do
14:       $R \leftarrow r_i + \gamma R$ 
15:      更新全局 actor 网络参数梯度  $d\theta_a \leftarrow d\theta_a + \nabla_{\theta'_a} [\ln \pi(a_i|s_i, \theta'_a)(R - V(s_i|\theta'_c)) + c_{reg} H(\pi(s_i|\theta'_a))]$ 
16:      更新全局 critic 网络参数梯度  $d\theta_c \leftarrow d\theta_c + \nabla_{\theta'_c} [R - V(s_i|\theta'_c)]^2$ 
17:    end for
18:    更新全局 AC 网络参数  $\theta_a \leftarrow \theta_a + \alpha d\theta_a, \theta_c \leftarrow \theta_c - \alpha d\theta_c$ 
19: until  $T > T_{max}$ 
  
```

5. A3C 算法与前述所有深度强化学习算法的另一处明显不同在于其取消了自 DQN 引入的 Replay Memory。Replay Memory 旨在缓和样本之间的相关性, 从而便于采用监督学习的思路进行深度学习, 但其应用又会消耗计算机储存资源。A3C 取消了 Replay Memory 是因为算法中全局网络参数的更新源于不同进程的贡献, 而每一进程所面临的情况又是独立同分布的, 因此多进程的应用本身即可达到缓和样本相关性的效果。

赌书消得泼茶香 当时只道是寻常

第八部分

计算机科学

第 29 章

计算机网络

29.1 概述

1. 计算机网络是通信技术与计算机技术紧密结合的产物，计算机网络本身就是一种特殊的通信网络，网络的发出端和接收端为计算机。称网络中的节点（计算机）为主机（hosts），主机间由通信链路（如光纤、电缆、无线）互相联系，当网络规模过大、主机数量过多时需引入交换网络，交换网络同样为网络，其节点称为交换节点（如路由器 routers 或交换机 switches），主机与交换节点由通信链路联系，再通过交换网络实现主机间的互联，在此过程中交换节点起到数据分组交换的作用；
2. 仅由硬件无法组成计算机网络，网络中的数据交换还需遵守事先约定的规则，即网络协议（network protocol），简称协议。协议规定了通信实体间所交换的消息的格式、意义、顺序以及针对收到信息或发生的事件所采取的动作。语法、语义、时序是协议的三要素；