Machine Learning Handbook

Xinhe Liu

2018-2-28

Contents

I High-level Views **1**

1 Math Review **2**

1.1 Linear Algebra 2

1.2 Probability 3

1.3 Information Theory 4

1.4 Optimization 5

1.4.1 Optimization Theory 5

1.4.2 Optimization Methods 6

1.4.3 Optimization Algorithms in Machine Learning . . . 6

1.5 Formal Logic 7

2 Statistics **8**

2.1 Concepts 8

2.1.1 Basic 8

2.1.2 Estimator and Estimation 8

2.1.3 Model Selection 11

2.1.4 Hypothesis Testing 11

2.2 Theorems 13
2.3 Important Distributions 13
2.4 Practice/Examples 15

3 Computational Learning Theory 16

4 Model Evaluation and Model Selection 17

4.1 Performance Metrics 18

5 Feature Engineering 20

5.1 Data Wrangling 20

6 Sampling 21

II Supervised Learning 22

7 Regression 23

7.1 Overview 23
7.1.1 Type of Models 23
7.1.2 The Key Questions 24
7.2 Linear Regression 25
7.2.1 Assumptions 25
7.2.2 Resolutions of Assumption Violations 26
7.2.3 Interpretation 27
7.2.4 Model Selection 29
7.2.5 Regularization, Ridge, Lasso 29

7.3 Nonlinear Regression Models 30

7.4 Generalized Additive Models 31

7.5 Generalized Linear Model 31

7.5.1 Logistic Regression 31

7.5.2 Extension: Softmax 33

7.6 Practice/Examples 33

8 Classic Statistical Learning Models 34

8.1 Support Vector Machine 34

8.1.1 Model and Assumptions 34

8.1.2 Kernel Function 36

8.1.3 Soft Margin, Slack Variable and Regularization . . . 37

9 Tree Models and Ensemble Learning 38

9.0.1 Bagging and Random Forest 38

9.0.2 Boosting and GBDT 38

10 Dimension Deduction 39

III Probabilistic Graphical Models 40

11 Naive Bayes 41

11.1 Bayesian Decision Theory 42

11.2 Model and Assumption 42

11.2.1 Semi-naive Bayesian Classifier 44

12 Max Entropy Model 45

13 Hidden Markov Model 46

14 Conditional Probabilistic Field 47

IV Unsupervised Learning 48

15 Clustering 49

16 Gaussian Mixture Model 50

17 Topic Model 51

18 Dimension Reduction 52

18.1 Principal Component Analysis(PCA) 52

18.2 LDA 53

V Deep Learning 54

19 Feedforward Neural Network 55

19.1 Multi-layer Perceptron 55

19.2 Convolutional Neural Network(CNN) 55

19.3 Deep Residual Network(ResNet) 55

19.4 self organizing feature map(SOMNet) 55

19.5 Restricted Boltzman Machine(RBM) 55

19.6 Model Optimization/Regularization 55

20 Recurrent Neural Network(RNN) 56

21 Generative Adversarial Networks(GAN) 57

22 Reinforcement Learning 58

23 Legacy 59

23.1 Neural Networks 59

23.1.1 History Class 59

23.1.2 Neural Networks 59

23.2 Deep Learning 61

23.2.1 Optimization in Deep Learning 61

23.2.2 Hyperparameter Tuning Methods 63

23.3 Key Questions and Status Quo 63

Part I

High-level Views

Chapter 1

Math Review

1.1 Linear Algebra

Concepts:

- scalar, vector, matrix, tensor(n-rank tensor, matrix is a rank 2 tensor)
- Gaussian Elimination, rank
- p-norm
  $$|X|_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$$
- inner product$< x_i, y_i >$, outer product
- orthogonaldimension, basis, orthogonal basis
- linear transformation $Ax = y$
- eigenvalue, eigenvector $Ax = \lambda x$ (transformation and speed)
- vector space, linear space(with summation, scalar production), inner product space(inner product space)

1.2 Probability

Concepts:

- Classic Probability Model: Frequentist
- Bayesian Probability Theory
- Random variable, continuous RV, discrete RV, probability mass function, probability density function, cumulative density function
- expectation, moments, variance, covariance, correlation coefficient

Theorems:

- Law of Total Probability
- Bayes' Rule
  $$P(H|D) = \frac{P(D|H)P(H)}{P(D)}$$
  $P(H)$-prior probability, $P(D|H)$-likelihood, $P(H|D)$-posterior probability,

Important Distributions:

1. Bernoulli distribution
2. Binomial distribution(n,p)
   $$P(X = k) = \binom{N}{k} p^k (1-p)^{(n-k)}$$
3. Poisson distribution
   $$P(X = k) = \lambda^k \frac{e^{-\lambda}}{k!}$$
4. Normal Distribution, See next chapter
5. Bernoulli Distribution

6. Uniform Distribution,
7. Exponential distribution
$$e^{-\frac{x}{\theta}}\theta$$
$$P(x > s + t | X > s) = P(x > t)$$
,
8. Poisson Distribution
9. normal distribution
10. t-distribution

Moment Generating Functions:

1.3 Information Theory

Concepts:

- Information
$$h(A) = -log_2 p(A)$$
(bit)
- (Information Source) Entropy
$$H(X) = -\sum_{i=1}^{n} p(a_i) log_2 p(a_i) \leq log_2 n$$
Maximize under equal probability
- Conditional Entropy
$$H(Y|X) = -\sum_{i=1}^{n} p(x_i) H(Y|X = x_i) = -\sum_{i=1}^{n} p(x_i) \sum_{j=1}^{n} p(y_j|x_i) log_2 p(y_j|x_i)$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{n} p(x_i, y_j) log_2 p(y_j|x_i)$$

- Mutual Information/Information Gain
$$I(X;Y) = H(Y) - H(Y|X)$$

- Kullback-Leibler Divergence (K-L) Divergence
$$D_{KL}(P||Q) = \sum_{i=1}^{n} p(x_i) log_2 \frac{p(x_i)}{q(x_i)} \neq D_{KL}(Q||P)$$
$$D_{KL}(f, \hat{f})) = \int_{-\infty}^{\infty} log(\frac{f_X(x)}{f(\hat{x})}) f_X(x) dx$$
K-L Divergence Measures the Distance of two distributions. The optimal encoding of information has the same bits as the entropy. Measures the extra bits if the real distribution is q rather than p. (Using P to approximate Q) K-L divergence plays an important role in both information theory and MLE theory. MLE $\hat{\theta}$ is actually finding the closest K-L Distance approximation of $f(x;\theta)$ to sample distribution.

Theorems:

- The Maximum Entropy Principle. Without extra assumption, max entropy/equal probability has the minimum prediction risk.

1.4 Optimization

1.4.1 Optimization Theory

- Objective function/Evaluation function, constrained/unconstrained optimizationFeasible Set, Optimal Solution, Optimal Value, Binding Constraints, Shadow Price, Infeasible Price, Infeasibility, Unboundedness

- Linear Programming

- Lagrange Multiplier
$$L(x, y, \lambda) = f(x, y) + \lambda \varphi(x, y)$$

- Convex Set, Convex Function $f : S \rightarrow R$ is convex if and only if $\bigtriangledown^2 f(\mathbf{x})$ is positive semidefinite

1.4.2 Optimization Methods

- Linear Search Method: Direction First, Step Size second
    - Gradient Descent: Batch Processing(Use all samples) vs Stochastic Gradient Descent(Use one sample)

      $$\theta = \theta - \alpha \frac{\partial J}{\partial \theta}$$

    - Newton's Method: Use Curvature Information

      $$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^{t} - (\frac{\partial^2 Loss(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T})^{-1} \frac{\partial Loss(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

- Trust Region: Step first, direction second. Find optimal direction of second-order approximation. If the descent size is too small, make step size smaller.
- Heuristics Method
    - Genetic Algorithm
    - Simulated Annealing
    - Partical Swarming/Ant Colony Algorithm

Quadratic programming (QP)

- Sequential Minimal Optimization(SMO)

1.4.3 Optimization Algorithms in Machine Learning

Loss Function Entropy K-L Distance Regularization Methods EM Algorithm Gradient Descent Stochastic Gradient Descent Batch Gradient Descent Momentum AdaGrad Adam Backward Propagation Gradient Checkling

1.5 Formal Logic

Concepts

- Generative Expert System: Rule+Facts+Deduction Engine
- Godel's incompleteness theorems

Chapter 2

Statistics

2.1 Concepts

2.1.1 Basic

- parameter(constant for probability model), statistic (model of sample data), data, sample, population
- point estimation, interval estimation, Confidence Interval($P(L \leq \theta \leq U)$, notice: θ is not random, L, U is random! (We repeat constructing confidence interval a n times, α percent of the times, it will contain *theta*.

2.1.2 Estimator and Estimation

- Method of Moments: $E(X^k)$ based on LOLN.
  If We have p parameters, we can use p moments to form a system of equations to solve $\theta_1, ...\theta_p$

  $$\sum_{i=1}^{n} X_i^j = E(X^j)$$

  , for j = 1,...,p

- Maximum Likelihood Estimation. Multiply p.m.f/p.d.f since every sample is independent. Maximize the likelihood of finding samples. If $X_1, ...X_n \overset{i.i.d}{\sim} f_x(x, \theta)$,

  $$l(\theta) = \prod_{i=1}^{n} f_{X_i}(x_i; \theta), L(\theta) = log l(\theta)$$

  $$\hat{\theta}_{MLE} = argmax_\theta f_x(x; \theta) = argmax_\theta L(\theta)$$

  Analytical or Numerically solved.

  $$\frac{\partial}{\partial\theta}[log L(\theta)] = 0, \frac{\partial^2}{\partial\theta^2}[log L(\theta)] < 0$$

  , for multiple parameters, we need the Hessian matrix to be negative definite $x^t H x < 0, \forall x$

- Properties of MLE

  1. Invariance $\hat{\theta}$ is MLE of θ, then $g(\hat{\theta})$ is MLE of $g(\theta)$
  2. Consistency

     $$P(\hat{\theta} - \theta) \to 0$$

     as $n \to 0, \forall\epsilon > 0$ Under the conditions

     (a) $X_1, ...X_n \overset{i.i.d}{\sim} f_x(x|\theta)$
     (b) parameters are identifiable, $\theta \neq \theta', f_x(x|\theta) \neq f_x(x|\theta')$
     (c) densities $f_x(x|\theta)$ has common support(set of x with positive density/probability), $f_x(x|\theta)$ is differentiable at θ
     (d) parameter space Ω contains open set ω where true θ_0 is an interior point
  3. Asymptotic Normality

     $$\sqrt{n}(\hat{\theta}_{MLE} - \theta_0) \to N(0, I^{-1}(\theta_0))$$

     $$I(\theta_0) = E(-(\frac{\partial}{\partial\theta}[log f(x, \theta)])^2) = E(-\frac{\partial^2}{\partial\theta^2}[log f(x, \theta)])$$

     called the Fisher Information

     $$\hat{\theta}_{MLE} \approx N(\theta_0, \frac{1}{nI(\theta_0)})$$

     $$nI(\theta_0) = E(-\frac{\partial^2}{\partial\theta^2} log L(\theta))$$

  - So the Variance of MLE($1/E(-\frac{\partial^2}{\partial\theta^2}logL(\theta))$) is the reciprocal of amount of curvature at MLE. Usually, We can just use the *observed Fisher Information* (curvature near $\hat{\theta_{MLE}}$) instead. ($I(\hat{\theta_{MLE}})$)
    $\frac{1}{nI(\theta_0)}$ is called Cramer-Rao Lower Bound.
    Under Multi-dimensional Case,

    $$I(\theta_0)_{ij} == E(-\frac{\partial^2}{\partial\theta_i\partial\theta_j}[logf(x,\theta)])$$

    $Hessian \approx nI(\theta_0)$ $Hessian^{-1} \approx nI(\theta_0)$ when we use numerical approach.
  - Under the above four conditions plus
    (a) $\forall x \in \chi$, $f_x(x|\theta)$ is three times differentiable with respect to θ, and third derivative is continuous at θ, and $\int f_x(x|\theta)dx$ can be differentiated three times under integral sign
    (b) $\forall\theta \in \Omega, \exists c, M(x)$ (both depends on θ_0) such that

    $$\frac{\partial^3}{\partial\theta^3}[logf(x,\theta)] \leq M(x), \forall x \in \chi, \theta_0 - c < \theta < \theta_0 + c, E_{\theta_0}[M(x)] < \infty$$

- Δ -Method: $g(\hat{theta_{MLE}})$ is approximately

  $$N(g(\theta),(g'(\theta))^2\frac{1}{nI(\theta)})$$

  if asymptotic normality is satisfied.
  In Multivariate Case:

  $$\hat{\theta} \sim N(\theta, \Sigma/n), \theta, \hat{\theta} \in R^p$$

  $$g : R^p \to R^m$$

  $$g(\hat{\theta}) \sim N(g(\theta), G\Sigma G^T/n)$$

  $$G = \begin{pmatrix} \frac{\partial g_1(\theta)}{\partial\theta_1} & \cdots & \frac{\partial g_1(\theta)}{\partial\theta_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m(\theta)}{\partial\theta_1} & \cdots & \frac{\partial g_m(\theta)}{\partial\theta_p} \end{pmatrix}$$

- Estimation criteria

  - Unbiased $E(\hat{\theta}) = \theta$
  - Minimum Variance (MVUE, minimum variance unbiased estimator) $Var(\hat{\theta}) < Var(\theta')$
  - Efficient
  - Coherent

2.1.3 Model Selection

AIC - Akaike Information Criterion

By K-L Distance

$$D_{KL}(f, \hat{f})) = \int_{-\infty}^{\infty} log(\frac{f_X(x)}{\hat{f(x)}}) f_X(x) dx$$

$$= const + \frac{1}{2} \int (-2log\hat{f}(x)) f(x) dx = const + AIC$$

$$A(f, \hat{f}) = -2logL(\theta) + 2p(\frac{n}{n-p+1})$$

2.1.4 Hypothesis Testing

- Hypothese, Test Statistic(T), Rejection Region
- p-value (chance of rejecting, largest choice of α that we would fail to reject H_0)
- type-I error(wrongly reject), type-II error(wrongly accept)

Hypothesis Testings (Based on the distribution of $\hat{\theta}$)

- Wald Test

$$T = \frac{\hat{\theta} - \theta_0}{Se(\hat{\theta})}$$

$$\hat{\theta}_{MLE} \approx N(\theta_0, \frac{1}{nI(\theta_0)})$$

$$T = \frac{\hat{\theta} - \theta_0}{\sqrt{\frac{1}{nI(\theta_0)})}}$$

- Likelihood Ratio Test
- Score Test

*Computation-based hypothesis testing approach

- Permutation tests:
  Test $X_1, ..X_n \sim F, Y_1, ...Y_n \sim G, if F = G$. Use
  $T = Mean(X_i) - Mean(Y_i)$, each time scramble X and V labels and should not not change the distributions of vectors $X_1, ...X_n, Y_1..., Y_n$
- Bootstrapping:
  $X_1, ...X_n \sim F$ with $T = T(X_1, ..., X_n)$, to get the distribution of T, **sample with replacement.** The belief is $(\hat{\theta} - \theta)$ should behave the same as $(\theta * -\hat{theta})$. The first quantity can be treated like a pivot. (use $(\theta *_1 - \hat{\theta}_1), ...(\theta *_n - \hat{\theta}_n)$ to test.

Multiple Testing

- Family-wise Error Rate(FWER) the probability of rejecting at least one of at least one null hypothesis
  Under independence, the probability of making mistake when all null are true: P(any type I mistake) = 1-P(no type I mistake for all) = $1 - (1 - \alpha)^M = \beta$)
- Bonferroni correction, assuming independence

  $$P(\bigcup_{i=1}^{n} \text{typeI mistake}) \leq \sum_{i=1}^{n} P(\text{typeI mistake}) \leq M\alpha$$

  ,control at $\alpha = \frac{\alpha}{M}$
  α being to small will impact power of the individual tests!
- False Discovery Rate(FDR): bound the fraction of type-I errors. R be the total number of hypotheses rejected. V be the number of rejected hypotheses that were actually null. Let FDR = V/max(R,1), control $E(FDR) \leq \alpha$.

2.2 Theorems

- Law of Large Number
- Central Limit Theorem
- Bias/Variance decomposition (error = bias + variance + noise)

$$MSE(\mu(X)) = E[(Y - \hat{\mu}(X))^2] == E[(Y - f(x) + f(x) - \hat{\mu}(X))^2]$$

$$= E[(Y - f(x)]^2 + 2E[(Y - f(x))(f(x) - \hat{\mu}(X))] + E[(f(x) - \hat{\mu}(X))^2]$$

$$= E[(Y - f(x)]^2 + 2E[(Y - f(x))(f(x) - \hat{\mu}(X))] + (f(x) - \hat{\mu}(X))^2$$

$$= \sigma_x^2 + Bias(\hat{\mu}(X))^2 + Var(\hat{\mu}(X))$$

2.3 Important Distributions

1. Normal Distribution, $X_1, ...X_n \sim N(\mu, \sigma^2)$ then
   (a) $\bar{X}$ and s^2 are independent
   (b) $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \sim N(0,1)$
   (c) $\frac{(n-1)s^2}{\sigma^2} \sim \chi^2_{n-1}$
   (d) $\frac{\bar{X}-\mu}{s/\sqrt{n}} = \frac{\frac{\bar{X}-\mu}{\sigma/\sqrt{n}}}{\frac{(n-1)s^2}{\sigma^2}\frac{1}{\sqrt{n-1}}} \sim t_{n-1}$

2. Multi-variate normal distribution

$$f_x(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$

   (a) $X_1, ...X_n$ normal $\Leftarrow (X_1, ...X_n)$ is multivariate normal. (Not equivalent)
   (b) $E(X) = \mu, Var(X) = \Sigma$
   (c) Linear transformations $AX + b \sim N(A\mu + b, A\Sigma A^T)$ remain multivariate normal
   (d) Marginals are multivariate normal, each sub-vector is multivariate normal, the parameters are just sub-matrices.
   (e) All conditionals are multivariate normal

3. t-distribution: like normal distribution, but heavier tails
   (a) $Z \sim N(0,1), Y \sim \chi^2_\nu$, Z, Y independent,
   $$X = Z/\sqrt{Y/\nu} \sim t_\nu$$
   (b) pdf has polynomial tails (decays much slower than exponential ones)
   (c) $\nu = 1$, it is the **Cauchy Distribution**, with very heavy tails (no expectation)
   (d) The MCF not exist. $E(|X|^k) < \infty$ for $k < \nu$, $E(|X|^k) = \infty$ for $k > \nu$
   (e) $X \sim t_\nu, E(X) = 0, Var(X) = \frac{\nu}{\nu-2}$
   $$f_X(x) = \frac{1}{\pi(1+x^2)}$$

4. χ^2 distribution
   $$f_x(x) = \frac{1}{(2^{k/2}\Gamma(k/2)}x^{\frac{k}{2}-1}e^{-\frac{x}{2}}, x \in [0,\infty) \sim Gamma(\frac{k}{2}, \frac{1}{2})$$
   (a) $E(X) = k, Var(X) = 2k, M_X(t) = (\frac{1}{1-2^t})^{k/2}$
   (b) $X \sim N(0,1) \Rightarrow X^2 \sim \chi^2$, $X_1, ...X_n \sim N(0,1) i.i.d \Rightarrow \sum X_i^2 \sim \chi^2$,
   $$f_X(x) = \frac{1}{\pi(1+x^2)}$$

5. F-Distribution

More Generalized Distributions

1. Generalized Error Distribution (symmetric)
2. Non-standard t-distribution (shift and scaling, heavy tailed, symmetric)
3. Theodossious skewed t-distribution
4. Theodossious skewed t-distribution plus shift

2.4 Practice/Examples

1. sample mean($\bar{X}$) is unbiased. Sample variance ($\frac{1}{n-1}\sum_{i=1}^{n} x_i^n$) is unbiased. But sample std is not unbiased. $SE(\bar{X}) = \frac{\sigma^2}{n}$
2. $\hat{C}ov(X.Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{Y})$ is unbiased
3. Distributions with Expectation not exist? (Cauchy)
4. Common Confidence Intervals:
   μ: $P(-t_{\alpha/2,n-1} \leq \frac{\bar{X}-\mu}{s/\sqrt{n}} \leq t_{\alpha/2,n-1}) = 1 - \alpha$,
   σ: $P(a \leq \frac{(n-1)s^2}{\sigma^2} \leq b) = 1 - \alpha$
5. Solve MLE/MOM for beta, exponential ($n/\sum X_i$, normal
6. * prove Asymptotic Normality of MLE(hint: using Taylor Expansion for $\theta, \hat{\theta}$)
7. * Use t^{th} quantile to approximate c.d.f, what's the distribution? ($Y_n = \frac{1}{n}\sum I(X_i < x)$, a Bernoulli distribution with $p = F_x(x)$,
   $\sqrt{n}[Y_n(x) - F_x(x)] \sim N(0, F(x)(1 - F(x)))$.
8. $X_1, ...X_n \sim Binomial(n, p)$, What's the MLE for p and Fisher Information? ($\hat{p} = \frac{x_i}{n}, I(p) = 1/p(1-p), var(p) = \frac{p(1-p)}{n}$)
9. $(x_i, y_i) \sim N(\mu_i, \sigma^2)$, find MLE for σ ($\frac{1}{4N}\sum(x_i - y_i)$)
10. How can you get N(0,1) random variables from U[0,1]? (Method1: Inverse Transformation, Method2; Use
    $SumZ_i^2 \sim \chi_k^2, k = 2, F^{-1}(u) = -2log(1-u)$,
    $R^2 \sim \chi^2, Z_1 = Rcos\theta, Z_2 = Rsin\theta, \theta \in [0, 2\pi]$
11. (Permutation test) how can you test $X_1, ..., X_n \sim F$, how can you test if F is symmetric? (Multiply -1 on all two form two sample groups)
12. Draw a bootstrap sample, what fraction of original data points appear in this sample on average?
    Define I be the indicator is it is in the sample.
    $E(\frac{1}{n}\sum I_i = E(I_i) = P(\text{ith point shows up}) = 1 - (1 - \frac{1}{n})^n$

Chapter 3

Computational Learning Theory

Chapter 4

Model Evaluation and Model Selection

- Hold-out: Separate to training/test(dev) set.
- Sampling Methods: Important for holding out. eg. Stratified Sampling
- Cross-Validation: Leave-One-Out and k-fold
- Bootstrapping: with m sampling with replacement:

  $$\lim_{m\to\infty}(1-\frac{1}{m})=\frac{1}{e}\approx 0.368$$

  Use $D \setminus D'$ as testing set
- Hypothesis Test for Cross Validation
  - Binomial test generalized error rate for one CV

    $$P(\hat{\epsilon},\epsilon)=\binom{m}{\hat{\epsilon}m}\epsilon^{\hat{\epsilon}m}(1-\epsilon)^{m-\hat{\epsilon}m}$$

  - t test for multiple CVs

    $$\mu=\frac{1}{k}\sum\hat{\epsilon}_i,\sigma=\frac{1}{k-1}\sum(\hat{\epsilon}_i-\mu)^2$$

  - Paired t-test for two Classifiers A and B (Permutation test or normal t test on $|\frac{\sqrt{k}\mu}{\sigma}|$
  - McNemar Test (χ^2 test)
  - Friedman Test and Nemenyi Post-Test (On MUltiple Learners)

4.1 Performance Metrics

Supervised Learning-Regression and Classification

- Confusion Matrix
  - Accuracy = $\int_{x\in D} \mathbb{I}(f(x) \neq y)p(x)dx$
  - Precision = TP/(TP+FP)
  - Recall = True Positive Rate = TP/(TP+FN)
  - False Positive Rate = FP/(FP+TN)
  - P-R Curve
  - $F - \beta$ Measure

$$\frac{1}{F_\beta} = \frac{1}{1+\beta^2}(\frac{1}{P} + \frac{\beta^2}{R})$$

$$F_\beta = \frac{(1+\beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

  - Specificity = 1- FPR
  - ROC (Receiver Operating Characteristic) Curve

$$\begin{aligned} AUC &= \int_{-\infty}^{+\infty} TPR(t)(FPR(t))' dt \\ &= \int_{-\infty}^{+\infty} \int_{t}^{+\infty} f_1(x)dx f_0(t)dt \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mathbb{1}_{x>T} f_1(x) f_0(t) dxdt \\ &= \mathbb{P}(S_1 > S_0) = 1 - Loss_{rank} \end{aligned}$$

f are densities for 0,1 class data

- Cost-Sensitive Loss: With unequal loss for FP and FN
- Cost Curve : Used to measure Cost-sensitive error rate: Use P(+)cost as horizontal and normalized cost as vertical.

Model Evaluation Performance Metrics A/B Test Bias, variance, Overfitting and Underfitting Hyperparameters Selection

Chapter 5

Feature Engineering

5.1 Data Wrangling

Basic Transformations

- Box-Cox power Transformation -useful when response is strictly postitive

  $$y = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, \text{ if } \lambda \neq 0 \\ log(y), \text{ if } \lambda = 0 \end{cases}$$

  λ could be selected via MLE

- Yeo-Johnson Transformation

  $$y = \begin{cases} \frac{(y+1)^{\lambda}-1}{\lambda}, \text{ if } \lambda \neq 0, y \geq 0 \\ log(y+1), \text{ if } \lambda = 0 \text{ if } \lambda = 0, y \geq 0 \\ \frac{(-y+1)^{2-\lambda}-1}{\lambda}, \text{ if } \lambda \neq 2, y < 0 \\ log(-y+1), \text{ if } \lambda = 0 \text{ if } \lambda = 20, y < 0 \end{cases}$$

  Optimal Transformation can be found with the MLE Method

Feature Engineering Discretization and Normalization Feature Combination Feature Selection Word Embedding

Chapter 6

Sampling

Part II

Supervised Learning

Chapter 7

Regression

7.1 Overview

7.1.1 Type of Models

All Basic Models begins with **Linear Regression** Because

- Linear relationship is the simplest relationship other than constant relationship or ″null″ model (average)
- It's a global model
- Data Invariance: Simple linear model don't do any pre-processing or transformation on the covariants.
- Very Explainable, limited interpretation power.

So, the alternation/improvements also focuses on these aspects

- Nonlinear features-Introduction of basis function
  - Polynomial Regression
  - Spline Models(eg. Cubic Spline, Smoothing Spline)

- Nonlinear parameters: Parameters Self-adjusting.(activation function is an example of basis function as well)
  - Neutral-Network
- global nonlinear: global nonlinear on both parameters and features achieved by linkage function, extends regression models to classification.
  - Generalized Linear Model
- Change the global model to a local model
  - Local Regression (Regression + KNN)
  - Nonparametric Regression
  - Kernel Function
  - Distance Based Learning
- Data Preprocessing (Transformation) and Dimension Reduction
  - PCA
  - LDA
  - Manifold Learning
- Improve Generalization Capability from outside (not from inside the model)
  - Regularization Methods(eg. Ridge, Lasso)
  - Ensemble Learning(Stacking, Aggregating): Random Forest, Boosting(GBDT), Deep Learning...

7.1.2 The Key Questions

- What assumptions are the model making
- How will we access the validity of those assumptions
- How can we be confident about out-of-sample fitting (overfitting problem)
- How do we make predictions and quantify the uncertainty in models?

7.2 Linear Regression

Common Terms

1. Independent Variable, Features, Covariates, Predictors
2. Dependent Variable, Response, Output (variable)
3. Scaling - transform a variable to have mean zero and variance one

7.2.1 Assumptions

Classic Assumptions for Statistics:

1. Linear Relationship between covariates and dependent variable
2. $E(\varepsilon) = 0$
3. $Var(\varepsilon) = \sigma^2$: Homoscedasticity
4. ε is independent with covariates
5. x is observed without error (and no perfect multicollinearity in multivariate case)
6. (optional, Gauss-Markov Theorem) ε is normal - when it is, OLS and MLE agrees and to be BLUE(Best Linear Unbiased Estimator)

Testing the Assumptions of Linear Regression

- Scatter Plot
  Linear Relationship and Outliers
- Residual Analysis $\hat{\varepsilon} = y - \hat{y}$
  Diagnostic Plots:
  1. Plot of Residuals vs. Fitted Values
  2. Normal Probability Plot
  3. Plot Residuals versus time (see any trend of fit)

- Cook's Distance
$$D_j = \frac{\sum_{i=1}^n (\hat{y}_i - \hat{y}_{i(-j)})^2}{(p+1)\hat{\sigma}^2}$$
Test Against $F_{(p+1),(n-p-1}$ degrees of freedom, over 50th percentile will definitely become a problem

- Detect Multicollinearity (two or more predictors are strongly related to one) - Use **Variance Inflation Factor**
$$VIF_k = \frac{1}{1 - R_k^2}$$
fit feature k against other predictors. Note VIF does not give any information of specific predictors

7.2.2 Resolutions of Assumption Violations

- Verify the Linear Relationships again. (non-linear regression, generalized linear models)
- Transformations (for outliers, heteroskesticities, etc)
- Use different models on different periods/data
- Weighted **Least Squares regression**, (for outliers, heteroskesticity)
- **Robust Regression** and Huber Loss Function
$$\sum_{i=1}^n \rho(\frac{y_i - x_i^T \beta}{\sigma})$$
Huber Loss Function
$$\rho(x) = \begin{cases} x^2 \text{ ,if } |x| < k \\ k(2(|x| - k), \text{ otherwise} \end{cases}$$
(default k=1.345) (when k=0, it is an L1-regression, $K \to \infty$, the regression goes back to a linear regression model. It is effective in down-weighting the extreme examples.

Special Situations

- Inputs are discrete variable - Factor Inputs (discrete features): a factor of k levels adds k-1 terms into the regression function.(k-1 different *beta*s)

7.2.3 Interpretation

Under Normal Condition, we have

$$y \sim N(\beta_0 + \beta x_i, \sigma^2), L(\theta) = (\frac{1}{\sqrt{2\pi}\sigma)}^{n} exp(-\frac{\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i))^2}{2\sigma^2})$$

Equivalent to minimize

$$RSS(\theta) = \sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i))^2$$

$$\partial_{\beta_i} RSS = 0, i = 1, 2$$

, we get

$$r_{xy} = \frac{s_{xy}}{s_x s_y}, \beta_1 = r_{xy}\frac{s_y}{s_x} = \frac{s_{xy}}{s_x^2}, \beta_0 = \bar{y} - \hat{\beta}\bar{x}$$

In Multi-variate Case:

$$f(x) = \mathbf{w}^T \mathbf{x} = \sum_{i=1}^{n} w_i x_i$$

$$\mathbf{w}* = \arg\min_{\hat{\mathbf{w}}} (\mathbf{y}) - \mathbf{X}\hat{\mathbf{w}})^T (\mathbf{y}) - \mathbf{X}\hat{\mathbf{w}})$$

$$\frac{\partial E}{\partial \hat{\mathbf{w}}} = 2\mathbf{X}^T(\mathbf{X}\hat{\mathbf{w}} - \mathbf{y})$$

$$\mathbf{w}* = (\mathbf{X^T X})^{-1}\mathbf{X^T y}$$

Assuming noise is normal, maximize

$$p(\mathbf{x_1}, \mathbf{x_2}...\mathbf{x_n}|\mathbf{w}) = \prod_k \frac{1}{\sqrt{2\pi}\sigma} exp[-\frac{1}{2\sigma^2}(y_k - \mathbf{w_t x_k})^2]$$

Another matrix representation

$$f(\beta) = min(Y - X\beta)^T(Y - X\beta), f'(\beta) = 2X^T(Y - X\hat{\beta}) = 0$$

to solve $\hat{\beta}$

$$min||y_k - \mathbf{w^T x}_k||^2 + \lambda||\mathbf{w}||_1$$

Variance Error In Prediction

$$V(\hat{y^*} - y^*) = \sigma^2 + \sigma^2[\frac{1}{n} + \frac{x^* - \bar{x})^2}{(n-1)s_x^2}]$$
$$= V(E(y^*) - y^*) + V(\hat{y^*} - E(y^*)) + 2cov(\hat{y^*} - y^*, \hat{y^*} - y^*)$$

The cross term is zero, the first term is variance with ε^*, second term is variance in β.

The confidence interval is $\hat{y^*} \pm t_{\alpha/2,n-2}SE(\hat{y^*})$.

R^2, the coefficient of determination: The proportion of the sum of squared response which is accounted by the model relative to the model with no covariance. (take mean of response)

$$R^2 = 1 - \frac{\sum(y_i - \hat{y_i})^2}{\sum(y_i - \bar{y})}$$

Note that $0 \leq R^2 \leq 1$ It only tells predictive power if the model is a good fit.

Adjusted R^2: R^2 + penalty P

Hat Matrix: The relationship of predicted value and response

$$Y = H\hat{Y}$$
$$H = X(X^TX)^{-1}X^T$$

The Diagonal Entires h_{ii} are the Leverages.

7.2.4 Model Selection

- Exhaustive Search by AIC or BIC (more stable than LOOCV)
- Stepwise Regression/Stepwise Variable Selection (At each step one covariate is added or dropped)
- Cross-Validation
  Leave-one-Out cross Validation of Linear Regression: Prediction Error Sum of Squares

  $$PRESS = \frac{\sum(y_i - \hat{y}_{-i})^2}{n}$$

  $$y_i - \hat{y}_{-i} = \frac{\hat{\varepsilon}_i}{1 - h_{ii}}$$

  h is the leverage (hat matrix)

7.2.5 Regularization, Ridge, Lasso

Ridge

$$Rss + \lambda \sum_{i=1}^{p} \beta^2$$

λ is the regularization parameter. The result of Ridge is a **Shrinkage** of $\hat{\beta}$ towards zero.

Note

1. No penalty for $\beta + 0$ or b.
2. The predictors should usually be standardized prior to fitting
3. Choose λ by cross-validation

Lasso(Least Absolute Shrinkage and Selection Operator)

(Tibshirani)

$$Rss + \lambda \sum_{i=1}^{p} |\beta|$$

Can be extended to

$$-loglikelihood + \lambda \sum_{i=1}^{p} |\beta|$$

Group Lasso

group predictors together to be either included or excluded.

Elastic Net

$$Rss + \lambda \sum_{i=1}^{p} (\alpha|\beta_j| + (1-\alpha)\beta_j^2)$$

$, 0 \leq \alpha \leq 1$

7.3 Nonlinear Regression Models

- Nonparametric Regression: Complexity controlled by the smoothing parameter (bandwidth). model complexity interpreted in **Degrees of Freedom/Effective degrees of freedom/equivalent degrees of freedom**
  Residual Degrees of freedom is n minus model degrees of freedom.

- Local polynomial Regression: only fit a **neighborhood** of a target point. parameter α to control the span-traditionally, 0.5. When weighting the data in the neighborhood, Fit by weighted sum of squares

$$\sum_{i=1}^{n} w_i(y_i - (\beta_0 + \beta_1 x_i))^2$$

$$w_i == \begin{cases} (1 - |\frac{x_i - x_0}{\text{max dist}}|^3)^3 & \text{,if } x_i \text{ is in the neighborhood} \\ 0, \text{ otherwise} \end{cases}$$

- Splines

- Penalized (Smoothing) Splines: find twice differentiable x to minimize

  $$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int [f^{(2)}(x)]^2 dx$$

  λ penalty for wiggy function. search of x can be a combination of **basis functions** (n + 4 basis functions, n is the knots)
- Cubic Splines

7.4 Generalized Additive Models

Additive Model : no interactions/cross terms

7.5 Generalized Linear Model

7.5.1 Logistic Regression

Sigmoid/ Log Probability Function

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-w^T x}}$$

Loss Function

$$J(z) = -ylogy + (1-y)log(1-y)$$

Training of the Model

MLE of w based on a Bernoulli Distribution

$$L(\mathbf{w}|\mathbf{x}) = \prod_{i=1}^{N}[p(y=1|\mathbf{x},\mathbf{w})]^{y_i}[1-p(y=1|\mathbf{x},\mathbf{w})]^{1-y_i}$$

Take Log to get the Loss Function

$$logL(\mathbf{w}|\mathbf{x}) = \sum_{i=1}^{N} -y_i logy_i + (1-y_i)log(1-y_i)$$

$$l(\mathbf{w}|\mathbf{x}) = logL(\mathbf{w}|\mathbf{x}) = \sum_{i=1}^{N}(y_i\mathbf{w}^T\mathbf{x}_i - log(1+e^{\mathbf{w}^T\mathbf{x}_i}))$$

Intuition

- The log odds (logit function)

$$ln\frac{y}{1-y} = \mathbf{w}^T\mathbf{x} + b$$

$$p(y=1|x) = \frac{e^{\mathbf{w}^T\mathbf{x}}}{1+e^{\mathbf{w}^T\mathbf{x}}}$$
$$p(y=0|x) = \frac{1}{1+e^{\mathbf{w}^T\mathbf{x}}}$$

- Exponential Family
- Maximum Entropy (See Loss function) of the exponential family

$$2\sum_{i]1}^{N} -y_i log\frac{y_i}{\hat{p}_i} + (1-y_i)log(\frac{1-y_i}{1-\hat{p}_i})$$

Regularization techniques:

- With L-1 or L-2 norm (Frobenius Norm)

$$J(z) = \frac{1}{m}\sum_{i=1}^{m} L(y_i, \hat{y}_i) + \frac{\lambda}{2m}\|\mathbf{w}\|_F^2$$

Another Intuition about minimizing the loss function is to minimize the **K-L Divergence** with Maximum-Entropy Model

*Connection with Naive Bayesian

- Naive Bayesian assumes $p(x_i|Y=y_k)$ follows a normal distribution. Then the posterior probability is

$$P(Y=0|x) = \frac{P(Y=0)P(X|Y=0)}{P(Y=0)P(X|Y=0)+P(Y=0)P(X|Y=1)}$$

$$= \frac{1}{1 + exp(ln\frac{P(Y=0)P(X|Y=1)}{P(Y=0)P(X|Y=0)})}$$

$$= \frac{1}{exp(ln\frac{1-p_0}{p_0} + \sum(\frac{\mu_{i1}-\mu_{i0}}{\sigma_i^2}X_i + \frac{\mu_{i0}^2-\mu_{i1}^2}{2\sigma_i^2}))}$$

- Though the solution follows the exact same pattern, Logistic Regression does not have the assumption of independence. When assumptions differ, the results differ. Generally, logistic regression results less bias, more variance(more flexible)
- The rate of convergence is also different, logistic regression needs more data feeding to perform better.

7.5.2 Extension: Softmax

$$P(Y = k|x) = \frac{e^{w^T x}}{\sum_{i=1} Ke^{w^T x}}$$

7.6 Practice/Examples

1. What is Anscombe's Quartet

Chapter 8

Classic Statistical Learning Models

8.1 Support Vector Machine

8.1.1 Model and Assumptions

Find an hyperplane can separate all the samples:

$$\begin{cases} \mathbf{w}^T\mathbf{x} + b \geq +1, y = +1 \\ \mathbf{w}^T\mathbf{x} + b \leq -1, y = -1 \end{cases}$$

The vectors make ″=″ are the support vectors.

The margin is

$$\gamma = \frac{2}{\|\mathbf{w}\|}$$

($\frac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$ is the point distance to plane)

So the problem is

$$\arg\max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|}$$
$$s.t.(\mathbf{w}^T\mathbf{x}_i + b)y_i \geq 1$$

Equivalent to

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$
$$s.t.(\mathbf{w}^T\mathbf{x}_i + b)y_i \geq 1$$

Lagrange Multiplier

$$L = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i(1 - (\mathbf{w}^T\mathbf{x}_i + b)y_i)$$

With first-order condition for **w** and b we can have

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i, b = \sum_{i=1}^{m} \alpha_i y_i$$

Then we get the dual problem

$$\arg\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$$
$$s.t \sum_{i=1}^{m} \alpha_i y_i = 0$$
$$\alpha_i \geq 0$$

When Satisfy K.K.T (Karush-Kuhn-Tucker) condition

$$\begin{cases} \alpha_i \geq +1, y = +1 \\ y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0 \\ \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1) \geq 0 \end{cases}$$

See Optimization, could be solved using SMO(Sequential Minimal Optimization)

8.1.2 Kernel Function

For Linear Un-separable problems, we can project to higher-dimensions

$$\arg\max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|}$$
$$s.t.(\mathbf{w}^T\phi(\mathbf{x}_i)+b)y_i \geq 1$$

$$\arg\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$
$$s.t \sum_{i=1}^{m} \alpha_i y_i = 0$$
$$\alpha_i \geq 0$$

Kernel

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$

We can find the solution by

$$f(x) = (\mathbf{w}^T\phi(\mathbf{x}_i)+b) = \sum_{i=1}^{m} \alpha_i y_i \kappa(\mathbf{x}, \mathbf{x}_i) + b$$

Theorem:

When a symmetric function has semi-positive definite kernel matrix

$$\begin{pmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_m) \\ \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m, \mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_m, \mathbf{x}_m) \end{pmatrix}$$

it can be a kernel function. Common Kernels are

- Linear Kernel
$$\kappa(\mathbf{x}_1, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$$
- Polynomial Kernel
$$\kappa(\mathbf{x}_1, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + c)^d$$
- Gaussian Kernel
$$\kappa(\mathbf{x}_1, \mathbf{y}) = exp(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2})$$
- Laplace Kernel
$$\kappa(\mathbf{x}_1, \mathbf{y}) = exp(-\frac{\|\mathbf{x} - \mathbf{y}\|}{\sigma})$$
- sigmoid Kernel
$$\kappa(\mathbf{x}_1, \mathbf{y}) = tanh(\beta\mathbf{x}^T\mathbf{y} + \theta)$$

8.1.3 Soft Margin, Slack Variable and Regularization

Chapter 9

Tree Models and Ensemble Learning

9.0.1 Bagging and Random Forest

9.0.2 Boosting and GBDT

Chapter 10

Dimension Deduction

Part III

Probabilistic Graphical Models

Chapter 11

Naive Bayes

Bayes' Rule

$$f_{Y|X}(y|x) = \frac{f_{(X,Y)}(x,y)}{f_X(x)} = \frac{f_{(X,Y)}(x,y)}{\int f(x|y)f(y)dy}$$

Byesian Inference:
All parameters are random variables,

$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta}$$

$$\pi(\theta|x) \sim f(x|\theta)\pi(\theta)$$

$\pi(\theta)$ is the prior distribution, $\pi(\theta|x)$ is the posterior distribution for θ given x.

Bayes Estimator

$$\hat{\theta}_{Bayes} = E(\theta|X) = \int \theta\pi(\theta|X)d\theta$$

Conjugate Distribution: $f(x), \pi$ is called conjugate distributions if model $\pi(\theta|x), \pi(\theta)$ follows the same Distribution

eg. Bernoulli(θ) and Beta(α, β), ($\pi(\theta|x) \sim$ Beta($\alpha + \sum X_i, \beta + n - \sum X_i$) ($f(x|\theta) = \prod_{i=1}^n f_{X_i}(X_i|\theta)$)

$$\hat{\theta}_{Bayes} = E(\theta|X) = \frac{\alpha + \sum X_i}{\alpha + \beta + n}$$

$$= \frac{\sum X_i}{n} \frac{n}{\alpha+\beta+n} + \frac{\alpha}{\alpha+\beta} \frac{\alpha+\beta}{\alpha+\beta+n}$$

The prior mean (second term) influences less as n grows.

Poisson(θ) and $Gamma(\alpha + \sum X_i, \beta + n)$

11.1 Bayesian Decision Theory

In State ω we take action $a \in A$, incurr Loss $L(\omega, a)$, how to choose a to minimize Risk:

$$R(a|x) = \sum_{j=1}^{k} L(\omega_j, a) P(\omega_j|\mathbf{x})$$

Decision Rule $d \in A$

$$d^*(x) = \arg\min_{a \in A} R(a|\mathbf{x})$$

$d^*(x)$ here is a Bayes optimal classifier here. and $R(d^*(x)|\mathbf{x})$ is the Bayes Risk.

11.2 Model and Assumption

When we have a 0-1 loss

$$L == \begin{cases} 0 \text{ , if i=j} \\ 1\text{, otherwise} \end{cases}$$

the risk become

$$R(a|x) = 1 - P(\omega_j|\mathbf{x})$$

$$d^*(x) = \arg\max_{a \in A} P(a|\mathbf{x})$$

If we build model around $P(a|\mathbf{x})$ directly, this is a **Discriminative Model**. If We try to model the joint distribution $P(\mathbf{x}, a)$, this is a **Generative Model**. Same as we get the Bayesian estimator, we try to find

$$P(a|\mathbf{x}) = \frac{P(a)P(\mathbf{x}|c)}{P(\mathbf{x})}$$

Naive Bayes made the important **assumption of attribute conditional independence** to write

$$\frac{P(a)P(\mathbf{x}|c)}{P(\mathbf{x})} = \frac{P(a)}{P(\mathbf{x})}\prod_{i=1}^{n} P(x_i|a)$$

We just need to count dataset to get

$$\hat{P}(x_i|a) = \frac{|D_{a,x_i}|}{|D|}$$
$$\hat{P}(a) = \frac{|D_a|}{|D|}$$

For continuous data, we can use probability density function to get the estimates.

In most cases, the smoothing **Laplacian Correction** is needed:

$$\hat{P}(x_i|a) = \frac{|D_{a,x_i} + 1|}{|D| + n}$$
$$\hat{P}(a) = \frac{|D_a| + 1}{|D| + n}$$

It can be proven, When we using the conjugate distribution of multinomial distribution to be the prior distribution and correct the parameter for Dirichlet Distribution to be $N_i + \alpha$ is equivalent for the Laplace Correction.

$$Dir(\mathbf{ff}) = \frac{\Gamma(\sum \alpha_i)}{\prod_{i=1}^{K} \gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \sum_{i=1}^{K} x_i = 1$$

11.2.1 Semi-naive Bayesian Classifier

Assume certain dependencies between attributes. The most common case is ”One-Dependent Estimator”. Such

- Super-Parent ODE
- Tree Augmented Naive Bayes: Use the Maximum Weighted Spanning Tree. Weighted by mutual information (conditional entropy), Build a complete graph on attributes.
- Average One-Dependent Estimator: Ensemble on the SPODE Models

Chapter 12

Max Entropy Model

Chapter 13

Hidden Markov Model

Chapter 14

Conditional Probabilistic Field

Part IV

Unsupervised Learning

Chapter 15

Clustering

Hierarchical Clustering K-means

Chapter 16

Gaussian Mixture Model

Chapter 17

Topic Model

Latent Dirichlet Analysis(LDA)

Chapter 18

Dimension Reduction

18.1 Principal Component Analysis(PCA)

$$\mathbf{x}_i = (x_{i1}, x_{i2}, x_{ip})^T$$

find u to maximize variance $v_i = \sum_{j=1}^{p} u_j x_{ij}$

subject to

$$\sum_{j=1}^{p} u_j^2 = 1$$

u are like the new axes

$$v_i = \sum_{j=1}^{p} u_j x_{ij}$$

same as finding the eigenvalue $Si\hat{g}ma$

X is samples stacking in columns,

$$\mathbf{X} = \mathbf{X} - \mathbf{1}\boldsymbol{\mu}^T$$

be X shifted to mean zero

Then $\mathbf{v} = \mathbf{X}\mathbf{u}$ is a transformation (new position in axis)

$$\mathbf{v}^T 1 = (\mathbf{X}\mathbf{u})^T \mathbf{1} = \mathbf{u}^T \mathbf{X}^T 1 = \mathbf{u}^T \mathbf{0} = 0$$

the mean of v entries is zero

The variance of entries of v :

$$\frac{1}{n}\mathbf{v}^{\mathbf{t}}\mathbf{v} = \frac{1}{n}\mathbf{u}^T \mathbf{X}^T \mathbf{X}\mathbf{u}$$

equivalent to maximizing

$$\frac{\mathbf{u}^T \mathbf{X}^T \mathbf{X}\mathbf{u}}{\mathbf{u}^T \mathbf{u}}$$

$$\mathbf{X}^T \mathbf{X}$$

is the covariance matrix, this is equivalent to find the first eigenvector of covariance matrix if we scale X to have unit standard deviation, the problem is finding eigenvalues for correlation matrix

18.2 LDA

Part V

Deep Learning

Chapter 19

Feedforward Neural Network

19.1 Multi-layer Perceptron

19.2 Convolutional Neural Network(CNN)

19.3 Deep Residual Network(ResNet)

19.4 self organizing feature map(SOMNet)

19.5 Restricted Boltzman Machine(RBM)

19.6 Model Optimization/Regularization

Batch Normalization Dropout Activation sigmoid softmax tanh ReLu

Chapter 20

Recurrent Neural Network(RNN)

RNN LSTM, GRU Attention Model Seq2Seq

Chapter 21

Generative Adversarial Networks(GAN)

Chapter 22

Reinforcement Learning

Chapter 23

Legacy

23.1 Neural Networks

23.1.1 History Class

- MP Neuron

  $$y = \phi(\sum_{i=1}^{N} w_i x_i)$$

  $$w_i(t+1) = w_i(t) + \eta[d_j - y_j(t)]x_{j,i}$$

  can only solve linear-separable problems

- Multilayer Perceptron: Including one hidden layer. The first Feedforward Neural Network. Errors passes by Back Propagation. Its it proven that a single-hidden layer multilayer perceptron can approximate any continuous functions at arbitrary error level.(universal approximation)

23.1.2 Neural Networks

Radial Basis Function Network

In the hidden layer, use activation function as activation function

Radial Basis Function

$$\rho(\mathbf{x}, \mathbf{w_i}, \sigma) = exp(\frac{-\|\mathbf{x} - \mathbf{w_i}\|^2}{2\sigma^2})$$

As long as a feature's distance to the center vector (here $\mathbf{w_i}$) the same, the function value is the same. $\mathbf{w_i}$ separates different hidden unit band with bandwidth σ

The Gaussian function $exp(-\|\mathbf{x} - \mathbf{u_i}\|^2)$(like Kernel) can help to transform linear inseparable case as-if projecting to a high-dimension space (same as SVM), to a linear separable case.

Alternatively, treat an RBF as a interpolation solution. It tries to data hyperplane. It reduces the noise by interpolation among the data points. The interpolated hyperplane still passes all data points.

Training of RBF

1. Initialization of $\mathbf{w_i}$ by random initialization or **unsupervised learning** like K-means.
   Usually, we have $\sigma = d_m ax/\sqrt{2K}$, $d_m ax$ is the maximum distance between centers. (make sure bandwidth is not too small or too big)
2. Training $\mathbf{w_i}$. Use Recursive Least Square
   $$\mathbf{R}(n)\hat{\mathbf{w}}(n) = \mathbf{r}(n)$$
   $\mathbf{R}(n)$ is the covariance matrix between hidden layer outputs ($\hat{y}$),
   $\mathbf{r}(n)$ is the covariance vector between hidden layer outputs ($\hat{y}$) and model response.
   Training by solving $\mathbf{R}^{-1}(n)$
3. After training, use Back propagation to train all parameters one more time. (train the whole network after training layers)

Compare with Neural Network: both can achieve universal approximation, while RBF network uses a local approximation approach.

Deep Learning sees most results in supervised Learning.

Feedforward neural network

23.2 Deep Learning

Characteristics of Deep-Learning

- Advantage of Deep-learning is significant mostly with large data set.
- traditional bias-variance trade-off can largely be overcome by adding more data (reducing variance) and training a larger network(reducing bias) cycle when data is sufficient.
- Optimization becomes more crucial in the training process. Dataset normalization, gradient checking are needed. Initialization carefully to avoid
    - Gradient Vanishing/Exploding

Common Regularization Techniques

- L-1 or L-2 regularization (notice: also affects backward propagation-cause **weight decay**.) Reduces variance.

$$J(z) = \frac{1}{m}\sum_{i=1}^{m} L(y_i, \hat{y}_i) + \sum_{l=1}^{L} \frac{\lambda}{2m} \|\mathbf{w}^{[l]}\|_F^2$$

- Dropout: randomly shutting down (with certain probability) nodes during every training. Still predicting with the whole network. Reduces over-reliance on a certain node.
- Data augmentation: adding transformed data to expand training set.
- Early stoping: early stopping during the optimization of parameters to avoid overfit.

23.2.1 Optimization in Deep Learning

Optimization Algos used in ML?Deep Learning Includes

- Mini-batch gradient descent: Use one batch (subset) of sample to compute the gradient each time. (one epoch) (one batch size =1, it is Stochastic gradient descent)

- Momentum Method: Smooth the gradient series with EWMA (Exponentially weighted averages

$$V_{dw} = \beta_1 V_{dw} + (1-\beta_1)dw, V_{dw}/ = (1-\beta_1^t)$$

$$V_{db} = \beta_1 V_{db} + (1-\beta_1)dw, V_{db}/ = (1-\beta_1^t)$$

- Root-Mean Square Prop (RMSProp)

$$S_{dw} = \beta S_{dw} + (1-\beta)dw^2$$

$$S_{db} = \beta S_{db} + (1-\beta)dw^2$$

$$w := w - \alpha \frac{dw}{\sqrt{sdw}}, b := b - \alpha \frac{db}{\sqrt{sdb}}$$

- Adam(Adaptive Moment Estimation) Algorithm L Combine RMSProp and Momentum

$$V_{dw} = \beta_1 V_{dw} + (1-\beta_1)dw, V_{dw}/ = (1-\beta_1^t)$$

$$V_{db} = \beta_1 V_{db} + (1-\beta_1)dw, V_{db}/ = (1-\beta_1^t)$$

$$S_{dw} = \beta_2 S_{dw} + (1-\beta_2)dw^2$$

$$S_{db} = \beta_2 S_{db} + (1-\beta_2)dw^2$$

$$w := w - \alpha \frac{v_{dw}}{\sqrt{sdw}+\epsilon}, b := b - \alpha \frac{v_{db}}{\sqrt{sdb}+\epsilon}$$

$$w := w - \alpha \frac{dw}{\sqrt{sdb}+\epsilon}, b := b - \alpha \frac{db}{\sqrt{sdb}+\epsilon}$$

- Learning Rate Decay

$$\alpha = \frac{1}{1 + \text{decay rate} \times \text{epoch num}}$$

23.2.2 Hyperparameter Tuning Methods

- Grid method
- Batch Normalization

  $$z_{norm} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

  $$\tilde{z} = \gamma z_{norm} + \beta$$

  Can speed up learning and add some noise to avoid overfitting. (Similar to dropout). In test time, usually use the EWMA across mini-batches on the mean and variance series to normalize the use trained β, γ to transform.

Performance Evaluation

23.3 Key Questions and Status Quo

- Scarce Data, Learning on Small Example (eg. Transfer Learning)
- Meta-Learning
- Knowledge from Hand-made features and Neural Network Combined Together