机器学习.

Model + Introduction
- linear, ☆logistic
- ☆SVM (soft Margin)
- ☆NN、CNN
- tree
- Bayesian
- Clustering
- Ensemble

Input | Output
Loss | Optimization
Model

☆ Optimization:

GD: $\beta = \hat{\omega}$, $z = \beta^T x$

$$\frac{\partial L}{\partial \beta} = \frac{\partial L}{\partial 6(z)} \cdot \frac{\partial 6(z)}{\partial z} \cdot \frac{\partial z}{\partial \beta}$$

$$= -\Sigma_i \left[\frac{y_i}{6(z)} - \frac{(1-y_i)}{1-6(z)}\right] \cdot 6(z)(1-6(z)) \cdot X_i$$

$$= -\Sigma_i [y_i - 6(z)] \cdot X_i$$

等价.

⇓

Matrix: $\frac{\partial L}{\partial \beta} = X(6(X^T\beta) - y)$

※注意维度 $X \in R^{d \times m}$ $y \in R^{m \times 1}$

KKT ⇒ 凸时充要, 非凸必要

x^*, λ^*, ν^* 满足: 最优解.

- $g_i(x) \le 0$, $h_i(x) = 0$ } 原问题可行
- $\lambda \ge 0$ → 对偶可行
- $\lambda_i g_i(x) = 0$ → 互补松弛
- $\nabla f(x) + \Sigma \lambda \nabla g(x) + \Sigma \nu \nabla h(x) = 0$ → 稳定性 ($L(x,\lambda,\nu)$ 对 x 导数为0, 鞍点)

⇓ 套到SVM上: $(x_+ - x_-) \cdot \frac{w}{\|w\|}$ 边界法向量

Margin: $\boxed{\frac{2}{\|w\|}}$ $= (1-b-(-1+b)) \cdot \frac{1}{\|w\|}$

L1 Linear

1) Model: ☆有时间看一眼 PPT 非 matrix 的推导

$\hat{y} = \omega^T x + b$

⇒ $\hat{y} = X\hat{W}$ 考试用

$\hat{\omega} = (\omega; b) = \binom{\omega}{b}$, $X = (x, 1)$

2) Loss: m个样本, $X \in R^{m \times d}$

☆MSE Loss: ↗ $\hat{y} = X\hat{\omega}$

$$L = \sum_{i=1}^{m} (y_i - \hat{y})^2$$

⇒本质是MLE.

$$= (y - X\hat{\omega})^T (y - X\hat{\omega})$$

3) Optimization

① Closed Form ☆看下矩阵求导. ⇒一定要检查维度

$$\frac{\partial L}{\partial \hat{\omega}} = 2X^T(X\hat{\omega} - y) = 0$$

⇒ $\hat{\omega} = (X^TX)^{-1} X^T y$ 不可逆 ⇓ 保留整体.

loss 加 $\times \frac{1}{2}$

② Gradient Descent / SGD

$\boxed{\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta)}$ → $X^T(X\hat{\omega} - y)$

- GD: 整个数据集 X、y $\Sigma_i(\theta^T x - y) x_k$
- SGD: 随机取一个样本, 直到收敛. $(\theta^T x - y) x_k$

Loss 有时会 $\times \frac{1}{2}$

☆ Logistic Regression

⇒ { 有概率, 易求解, 无 prior assumption

1) Model:

$$\hat{y} = 6(\omega^T x + b) = \frac{1}{1 + e^{-\hat{\omega}x}}$$

⇒ 本质是让 $\hat{\omega}x \to \ln\frac{y}{1-y}$

$6(x) = \frac{1}{1+e^{-x}}$, $6'(x) = 6(x)(1 - 6(x))$

⇓ softmax要分 i=j, i≠j 讨论.

☆ Loss:

由MLE 导出: likelihood: $l(\omega, b) = \sum_{i=1}^{m} \log p(y_i | x_i, \omega, b)$

↗ loss: $\Sigma - y x_i \hat{\omega} + \ln(1 + e^{x\hat{\omega}})$

$P(y_i | x_i, \omega, b)$ {
- $y_i P_1(x_i; \hat{\omega}) + (1 - y_i) P_0(x_i; \hat{W})$ ⇒本质等价
- $P_1(x_i; \hat{\omega})^{y_i} \cdot P_0(x_i; \hat{\omega})^{1-y_i}$ 形式简单.

{ $P_1(x_i | \hat{\omega}) = \hat{y} = 6(z) = \frac{1}{1 + e^{-\hat{\omega}x}} = \frac{e^{\hat{\omega}x}}{1 + e^{\hat{\omega}x}}$
{ $P_0(x_i | \hat{\omega}) = 1 - \hat{y} = 1 - 6(z) = \frac{1}{1 + e^{x\hat{\omega}}}$

⇓ Cross Entropy

Loss: min $\boxed{\sum_{i=1}^{m} -(y_i \log P_{1i} + (1 - y_i) \log P_{0i})}$

⇓

$$\sum_{i=1}^{m} -y_i \beta^T x_i + \log(1 + e^{\beta^T x_i})$$

L3 & L4 Introduction & Model Selection

1) Generalization

⇒ ultimate goal: performance on unseen patterns (not training set)

2) notation

{ Generalization error, $x \sim D$ 泛化误差(测试集) → 分布
{ Empirical error, $x \in S$ 经验误差(训练集) → samples 采样

⇒ $P(h(x) \ne c(x))$ (hypothesis, concept)

3) Performance Measure

{ Error rate: $E(f; D) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(f(x_i) \ne y_i)$
{ Accuracy: $1 - E(f; D) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(f(x_i) = y_i)$

☆ Confusion Matrix

Ground Truth \ Prediction	P	N
P	TP	FN
N	FP	TN

填数量

Precision = $\frac{TP}{TP + FP}$ 查准率 (prediction)

Recall = $\frac{TP}{TP + FN}$ 查全率

$F_1 = \frac{2PR}{P+R}$ ↘ 全正的有多准

L6 SVM

⇒ Soft Margin + Lagrange.

Lagrange Dual:

原问题: 标准形式
$\min f(x)$
s.t. $g_i(x) \le 0$, $i = 1 \cdots m$
$h_j(x) = 0$, $j = 1 \cdots n$

⇒ $L(x, \lambda, \nu) = f(x) + \Sigma_i \lambda_i g_i(x) + \Sigma_j \nu_j h_j(x)$

⇓ dual $\min_x \max_{\lambda,\nu} \leftrightarrow \max_{\lambda,\nu} \min_x$

$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu)$

⇓ ☆对偶问题: $\boxed{\max g(\lambda, \nu) \quad s.t. \ \lambda \ge 0}$

Slater 条件: 凸 + 严格不等

⇓ $d^* = p^*$ 强对偶.

*SVM 记下求的顺序和表示的方法

1) Model: $\omega^T x + b$ = {
- 0 → 超平面, 分类边界
- =1, 正SV; >1, 正类
- =-1, 负SV; <-1, 负类
- (-1,1), Margin

线性模型.

w 是 $\omega^T x + b = 0$ 超平面法向量.

2) Loss: max 转 min

Hard Margin:

$\arg\min_{\omega, b} \frac{1}{2}\|w\|^2$ → margin 最大

s.t. $y_i(\omega^T x_i + b) \ge 1$ → 全分对 P31

优化变量: ω, b. 看下PPT求解过程

☆ Soft Margin: 引入 slack ⇒ penalty

$$\min_{w,b,\varepsilon} \frac{1}{2}\|w\|^2 + C \Sigma_i \varepsilon_i$$

s.t. $y_i(w^T x + b) \ge 1 - \varepsilon_i$

$\varepsilon_i \ge 0$

$\|w\|^2 = \Sigma w_i^2 = w^T w$

3) Optimization ☆

① $L(\omega, b, \varepsilon, \lambda, \mu)$ ε_i 也是优化变量

$= \frac{1}{2}\|w\|^2 + C\Sigma_i \varepsilon_i + \Sigma \lambda_i (1 - \varepsilon_i - y_i(w^T x_i + b)) - \Sigma \mu_i \varepsilon_i$

② KKT:

$\frac{\partial L}{\partial w} = w - \Sigma \lambda_i y_i x_i = 0$ 看详细推导 假设圈.

$\frac{\partial L}{\partial b} = -\Sigma y_i \lambda_i = 0$ 记得.

$\frac{\partial L}{\partial \varepsilon_i} = C - \lambda_i - \mu_i = 0$ → 因对 ε_i 单个求导可得

③ 代入写 $g(\lambda, \mu)$

☆$g(\lambda, \mu) = \Sigma \lambda_i - \frac{1}{2} \Sigma_i \Sigma_j \lambda_i \lambda_j y_i y_j x_i^T x_j$

⇓ ☆Dual Problem:

$\min -g(\lambda)$

s.t. $C \ge \lambda_i \ge 0$ → 与 hard 不同

$\Sigma y_i \lambda_i = 0$

$\lambda_i + \mu_i = C$ 推出

解出 $\lambda^{(\omega)}$ 求解 SMO不考

用任意SV代入出 b.

※由互补松弛 (Sparsity)

$\lambda_i (y_i(\omega^T x + b) - 1 + \varepsilon_i) = 0$

⇒ $w = \Sigma \lambda_i y_i x_i$ 只与 SV 有关 →对 Hard 也是

$y_i(w^T x + b) = 1 - \varepsilon_i$ 的

kernel function SVM: $f(x) = w^T \phi(x) + b$

⇒ 解决线性不可分数据. model 为: $\sum_{i}^{m} \lambda_i y_i k(x_i, x) + b$.

⇒ 以内积形式 $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ → map.

核矩阵 K, $K_{ij} = k(x_i, x_j)$ kernel function 的线性组合.

⇒ { Linear : $k(x_i, x_j) = x_i^T x_j$

Polynomial : $k(x_i, x_j) = (x_i^T x_j + c)^n$

二次: ⇒ 转出 Φ

$k(x, z) = (x^T z + c)^2$

$= (\sum_i x_i z_i + c)(\sum_j x_j z_j + c)$ 将 x, z 分离.

$= \sum_i \sum_j x_i x_j z_i z_j + 2c \sum_i x_i z_i + c^2$

全排, 换序的也要.

☆ ⇒ 构造 $\Phi(a) = [a_1^2, a_1 a_2, a_1 a_3 \cdots\cdots a_n^2, \sqrt{2c}\, a_1, \cdots \sqrt{2c}\, a_n, c]$ ☆.

L8 Back Propagation. ☆ MLP (CNN, Linear Layer).

⇒ 本质是快速求导, 存下先算的, 加速后算的.

1) forward ⇒ 算 output

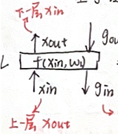
↓

backward ⇒ 从输出向输入层算.   看作业(PR). 看图、画图.

↓

update

2) 方法: 基于链式法则.



→ 上一层传的gradient: $\frac{\partial L}{\partial x_{out}}$

⇒ 对这一层: $\boxed{\frac{\partial L}{\partial w_L} = g_{out} \times \frac{\partial x_{out}}{\partial w_L}}$

↳ 给下一层 gradient: $\frac{\partial L}{\partial x_{in}} = \frac{\partial L}{\partial x_{out}} \cdot \frac{\partial x_{out}}{\partial x_{in}} = g_{out} \cdot W$

$f(x_{in}, W_L) = W_L x_{in}$

更新: $w_L^{t+1} = w_L^t - \eta \cdot \frac{\partial L}{\partial w_L}$.

看题实操.

⇒ 小心激活函数导数.

⇒ 用的是对单项求导, 不是整体求导.

⇒ 一步一步来, 不怕多写.

L9 CNN

⇒ 记住规则, 实操算一下 { Conv / Pooling

1) Conv

⇒ in_channel → 一个 in Feature 一个2D kernel   parameter sharing, 一张图 kernel parameter 同.

out_channel → 一个 out Feature 一个3D kernel.

padding → 单边的大小, $2p = K_{W/H} - 1$ 则不变

stride. → 步幅.

kernel size.

⇒ 大小: $\frac{H/W - K + 2P + S}{S}$

计算: 元素对应相乘再求和, + 偏置 一个2D kernel 一个.

参数要训练. 别忘.

2) Pooling 通道不变, 无须训练.

⇒ padding { max ⇒ 最大值.

stride avg ⇒ 均值(窗内)

kernel size.

→ here.

L11 Decision Tree ⇒ { 非线性 / 可解释 / 易使用.

1) model

条件决策, 从根节点 → 叶节点.

要么结论, 要么下一个节点测试

建树算法: $\boxed{\text{选择划分特征}}$ ☆

↓

每个特征一个节点 { 空: 取此处大多数 / 有: 继续划分.

另: 所有 ∈ 同一 label ⇒ leaf node

所有特征值相同 ⇒ leaf node 大多数.

2) Optimization ☆看题目.

用 Information Gain 来 Feature Select: increase purity

Information Entropy:

$Ent(D) = -\sum_{k=1}^{|y|} p_k \log_2 p_k$ label class 种类; proportion of class k.   ☆ $\log_2$ 要标.

⇒ 低的话 purity 高.

⇒ 全为一类/空时为0, 平均分时最高: $\log|y|$.

feature (最小)   加权.

$\boxed{Gain(D, a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)}$

⇒ Gain > 0 说明 purity ↑.

选 Gain 最大的 Feature a.   ☆一定要看题手算(笔记). 作业

Pruning: { pre-prunning / post-prunning ⇒ 控制模型复杂度, 防止过拟合

⇓ 通过 validation set 看效果

① pre-prunning:

⇒ 训练/建树时

⇒ 尝试 prunning:

判断当前划分能否让 val acc↑ { 能: splitting / 不能: 按照多数成立叶节点 label

⇒ 简单但效果不好.

② post-prunning:

⇒ 在已训练好的树上剪枝(底向上) → 查每个结点(非叶).

⇒ 用多数结点 label 替代子树 看 val acc { ↑ 剪枝 / ↓ 复原.

⇒ 算法开销大, 但泛化好.

Lecture 12 Bayesian Classifier.

1) Model

discriminative model：直接求 $P(c|x)$
generative model：$P(x,c) \to P(c|x)$

Bayes theorem：

$$P(c|x) = \boxed{\frac{P(c)\cdot P(x|c)}{P(x)}}$$

$$= \frac{P(c)\cdot \prod_{i=1}^{d} P(x_i|c)}{P(x)}$$

→ 特征间独立假设 ⇒ Naive Bayes.

2) Loss：

引入分类错误损失：λ_{ij} → 真值为 j，错分为 i 的 risk. $\lambda(i|j)$ action

条件风险 conditional risk

$$\Rightarrow R(c_i|x) = \sum_{j=1}^{N} \lambda_{ij}\cdot P(c_j|x)$$

→ 对一个 sample x，分为 c_i 的 risk

action 分类结果

⇓

Total risk： $h(x)$ → 分类器

$$R(h) = E_x[R(h(x)|x)]$$

⇓

Optimal：

$$h^* = \underset{c\in y}{\operatorname{argmin}}\ R(c|x)$$

⇒ 找到使每个 x loss 最小的 label c

Bayes optimal classifier

$R(h^*)$ ⇒ Bayes Risk

⇒ 机器学习理论最优

3) Optimization

⇒ 概率模型的训练过程就是概率计算/参数估计.（求 $P(c)$、$P(x|c)$）

① Naive Bayes

⇒ prior assumption：特征属性间独立

$$P(c) = \frac{|D_c|}{|D|}$$

$$P(x_i|c) = \frac{|D_{c,x_i}|}{|D_c|}$$，连续时 Gaussian μ_{ci}, σ_{ci}

$$\frac{1}{\sqrt{2\pi}\sigma_{ci}}\cdot \exp\left[-\frac{(x_i-\mu_{ci})^2}{2\sigma_{ci}^2}\right]$$

$$\Rightarrow P(c|x) = \frac{P(c)\cdot P(x|c)}{P(x)} = \frac{P(c)}{P(x)}\cdot\prod_{i=1}^{d} P(x_i|c)$$

$$\Rightarrow h(x) = \underset{c\in y}{\operatorname{argmax}}\ P(c|x)$$

⇒ 后验 p 最大的.

NB是线性分类器.

Tips：

Laplace Smoothing：

⇒ 当某一特征从未在此 label 上出现时，

$P(x_i|c)=0$，其它特征也被忽略，$P(c|x)=0$.

⇒ 做一个平滑：

两个都要平滑

$$\hat{P}(c) = \frac{|D_c|+1}{|D|+N}$$

→ c 类别数.

$$\hat{P}(x_i|c) = \boxed{\frac{|D_{c,x_i}|+1}{|D_c|+N_i}} = \frac{|D_{c,x_i}|+1}{\sum_i (D_{c,x_i}+1)}$$

用求和验证！（如果都算了的话）.

→ c中 x_i 数 ⇒ 注意是整训练集中的都要，此类中未出现也要

PPT P72

种类数.

4) Application

multinomial distribution：

⇒ 类子有放回抽球.

$$P(X_1=x_1 \cdots X_k=x_k)$$ ⇒ 第 k 种颜色

$$= \frac{n!}{x_1!\cdots x_k!}\cdot p_1^{x_1}\cdots p_k^{x_k}$$

$n!$ → 共抽 n 次，→ 次方，有 x_k 个

$(n = \sum_i x_i)$

自是一个分布可代入数据求.

抽一次球到 k 色 Prob

Text classification：看PPT例题.

⇒ Feature：单词出现频率. / c → 文档类型

$$c_{NB} = \underset{c\in y}{\operatorname{argmax}}\ \log P(c) + \sum_{i\in position}\log P(w_i|c)$$

选最大后验

加 log，防溢出

每个位置的相乘

Lecture 13 Ensemble Learning

⇒ 集成"好而不同"的弱学习器，最终产生超强性能

有一定 acc　多样性 ⇒ Ensemble 核心是如何 train 出"好而不同" learner.

⇒ 串行，learner 间有依赖 ⇒ boost
$\boxed{\text{并行}}$，可同时生成 ⇒ bagging

1) Bagging ⇒ Bootstrap AGGregatING（多分类、回归可用）

⇒ ① 在训练集中有放回采样 T 个 m 大小采样集
m个样本　会重复　和 dataset 一样大.

未被采的可当 val 集.

② 用 T 个采样集训 T 个 base learner

③ 结合（voting / avg）⇒ 选最多学习器认同的
回归.

$$H(x) = \underset{c\in y}{\operatorname{argmax}} \sum_{t=1}^{T} \mathbb{I}(h_t = c)$$

⇒ bias 没降 variance 下降.

⇓

1.1) Random Forest

⇒ 除了样本扰动再加上属性扰动

⇒ 对 base learner 每个结点，先在所有可选 feature 中随机选 k 个，再从这 k 个中选最好的

与 bagging 比：

① 训练效率高，只算子集 feature

② 开始精度低，因 learner 少时差。（由于子集，学习器弱）但之后泛化强.

Lecture 14 Clustering

K-means (过)

☆Hierarchical Clustering 层次聚类.

1) AGNES (bottom up):

⇒ ① 初始时每个样本视为一个簇 cluster

② 每一轮将最近的簇融合成一个 → 距离度量

③ 直到剩下预设的 k 个簇. 超参

⇓

Implementation:

Distance Matrix

	a	b	c	d
a	0	d_{ab}	…	
b	d_{ba}	0		
c			0	
d	…			0

⇒

	(a,b)	c	d
(a,b)	0	…	
c	⋮	0	
d			0

☆更新只修改有变化的行列.
其余不变.

☆变化的行列的值可由之前的推.

Ex:

$\Rightarrow \min((a,b),c)$

$= \min(d_{ac}, d_{bc})$

dist( ) 可用: 维度

$$dist(x_i, x_j) = \left(\sum_{u=1}^{n} |x_{iu} - x_{ju}|^p\right)^{\frac{1}{p}}$$

p 常为 2. Euclidean.

距离度量:

minimum: $d = \min_{x \in C_i, y \in C_j} dist(x,y)$ [single linkage]

maximum: $d = \max dist(x,y)$ [complete linkage]

average: $d = \frac{1}{|C_i||C_j|}\sum\sum dist(x,y)$ [average linkage]

⇓

不同策略适合不同数据分布,效果不同.

* k-means

☆Derivative:

① $\frac{\partial x^T A x}{\partial x} = (A + A^T)x$

② $\frac{\partial x^T a}{\partial x} = \frac{\partial a^T x}{\partial x} = a.$

③ $\frac{\partial a^T x b}{\partial x} = a \cdot b^T$

↓ $(a \cdot b^T)^T$

④ $\frac{\partial a^T x^T b}{\partial x} = b a^T$

☆考虑分类讨论,

c>0 …

c<0 …

Gaussian: $x \sim N(\mu, \sigma^2)$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

SVM kernel:

① 先写判别函数: $f(x) = \sum \lambda y x \cdot x + b.$

② 观察. 二次核 ⇒ 椭圆 / 双曲线

高斯 ⇒ 圆.

↓

Ex: $\exp(-r\|x_i - x_j\|^2)$ ⇒ r 越大, 当 $x_i x_j$ 差的大时结果相近 ⇒ SV 多.

⇒ kernel matrix = $\begin{bmatrix} \kappa(x_1,x_1) & \cdots & \\ & & \kappa(x_n,x_n) \end{bmatrix}$

* 正定, 则可当核函数.

Bayesian Decision

① $P(H|X) = \dfrac{P(H)\cdot P(X|H)}{P(X)}$

直接判断.

② X, Ω, A (动作), λ → 损失.

⇒ ☆ $R(\alpha_i|X) = \sum_{j=1}^{c} \lambda(\alpha_i|w_j)\cdot P(w_j|X)$

⇒ Bayes decision rule：损失小的选.

$\alpha(X) = \arg\min_{\alpha_i \in A} R(\alpha_i|X)$.

分类问题中 α_i 为分类的类别.

二分类：$R(\alpha_1|X) < R(\alpha_2|X)$ ⇔ $\lambda_{21}\cdot P(w_1|X) + \lambda_{22}\cdot P(w_2|X)$ （判 → 真）

⇔ $\dfrac{P(X|w_1)}{P(X|w_2)} > \dfrac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}}\cdot\dfrac{P(w_2)}{P(w_1)}$ → 前判后真.

多分类：记得全概率公式 & 条件概率公式

使用0-1损失时：对0错1.

$R(\alpha_i|X) = 1 - P(w_i|X)$.

$\sum_{j=1}^{c}\lambda(\alpha_i|w_j)\cdot P(w_j|X)$

损失小

⇓

w_i if $P(w_i|X) > P(w_j|X)$ 谁大选谁.

$P(X|w_i) \sim N(\mu_i, \sigma_i^2)$

高斯分布：$X \sim N(\mu, \sigma^2)$

$P(x) = \dfrac{1}{\sqrt{2\pi}\,\sigma}\, e^{\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]}$

$E[x] = \int_{-\infty}^{+\infty} x\, p(x) = \mu$

$E(X-EX)^2$.

$Var[x] = \int_{-\infty}^{+\infty} (x-\mu)^2\cdot p(x) = \sigma^2$.

多元 $p(x) = \dfrac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)\right]$

x, μ 向量，Σ 矩阵.

书26页.

⇓

针对Gaussian的判别函数推理 (P28)

$\begin{cases} \Sigma_i = \sigma^2 I \to \text{linear} \\ \Sigma_i = \Sigma \\ \Sigma_i = \text{arbitary} \to \text{quadratic} \end{cases}$

⇒ 带入后二次项无意义. (书上)

$g_i(x) = \ln p(x|w_i) + \ln P(w_i)$

⇔ $g_i(x) = -\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(w_i)$.

Parametric Estimation

Maximum Likelihood

① $P(w_j)$ → count.

② $P(x|w_j)$ → 假设服从某一分布

⇒ $P(D|\theta) = \prod_{k=1}^{n} p(x_k|\theta)$ （dataset）

$\hat{\theta} = \arg\max_{\theta} P(D|\theta)$.

求解：① 加上 $\ln$ （$\Pi \to \Sigma$） ② 对 θ 求导为0 （Σ 还在.） ⇒ 必要条件

③ 检查极值点哪一个最优 { 代入 / 二阶导.

应用：Gaussian { 已知 Σ，估计 μ / Σ, μ 未知　P70

$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k$

$\hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(x_k-\hat{\mu})(x_k-\hat{\mu})^T$

Bayesian Estimation

求 $P(w_j|x, D^*)$.

⇒ 每个类的后验，最大的即结果 ⇒ 可能求到的后验

此处 $P(x|D) = P(x|w_j, D_j)$.

☆ $P(w_j|x, D) = \dfrac{P(w_j)\cdot P(x|w_j, D_j)}{\sum_i P(w_i)\cdot P(x|w_i, D_i)}$

乘法公式+条件概率公式

求解：P74 高斯情况，求 $P(\mu|D)$.

☆ 核心：$P(x|w_j, D_j)$

① $P(x|D) = \int p(x|\theta)p(\theta|D)\, d\theta$.

② $P(\theta|D) = \dfrac{P(D|\theta)p(\theta)}{\alpha}$　$\alpha \Leftarrow \int P(D|\theta)p(\theta)d\theta$.

③ $P(D|\theta) = \prod_{k=1}^{n} p(x_k|\theta)$.

递归 P79.　看题目. 按步骤 + 化简.

HMM：

Notations：

① $\Omega = \{w_1, \cdots\}$ ⇒ state

② $V = \{v_1, \cdots\}$ ⇒ signal/symbol

③ $w^T = \{w(1), \cdots\}$ （时间）⇒ state sequence

④ $V^T = \{v(1) \cdots\}$ ⇒ symbol sequence

⑤ A, B, π.　$\pi_j = P(w(1) = w_j)$

Evaluation：P108

求 $P(V^T|\theta)$

前向 forward. ⇒ 记得求和，从1开始

后向 backward. ⇒ 看准下标，从T开始

F：$\alpha_j(t) = \left[\sum_{i=1}^{c}\alpha_i(t-1)a_{ij}\right] b_{jv(t)}$ （状态；前；后）　⇒ 结果是求和　$\alpha_j(1) = \pi_j b_{jv(1)}$

B：$\beta_j(t) = \sum_{i=1}^{c}\beta_i(t+1)\, a_{ji}\, b_{iv(t+1)}$ （后；前）　$\beta_j(T) = 1$

⇒ $P = \sum_j \pi_j b_{jv(1)}\beta_j(1)$

建议列表做.

☆ Decoding：求 w^T，$\max P(w^T|\theta, V^T)$

① Viterbi

$\delta_j(t) = \max_{w(1)\cdots w(t-1)} P(w(1), \cdots, w(t-1), w(t) = w_j, v(1)\cdots v(t)|\theta)$

⇒ 到 t 的前面 state/symbol 最高的概率 ($w(t) = w_j$ 时)

⇒ 求解：$\delta_j(1) = \pi_j b_{jv(1)}$，$\psi_j(1) = 0$.

$\delta_j(t) = \left[\max_i \delta_i(t-1)a_{ij}\right] b_{jv(t)}$

$\psi_j(t) = \arg\max_i \delta_i(t-1)a_{ij}$ → 最有可能的 w 下标

值是下标.　⇒ 有n个，由后面选 (根据 t 推 t-1)

⇓

☆ $\boxed{\begin{aligned} w(T) &= \arg\max_j \delta_j(T) \\ w(t) &= \psi_{w(t+1)}(t+1) \end{aligned}}$ 从后往前. (后处理).

小数计算很多 别错了.(计算器)

Learning：求 $\theta = \{A, B, \pi\}$，know V^T.

⇒ $\arg\max_{\theta} P(V^T|\theta)$.

前向-后向算法 (P113).

$\hat{\pi}_i$ / $\hat{a}_{ij}$ / $\hat{b}_{ik}$

Bayesian Belief Network

P44.

只有父子相关

联合：$P(a,b,c) = P(a|Pa(a))\cdot P(b|Pa(b))\cdot P(c|\cdots$

边缘：$P(a) = \sum_{b,c} P(a,b,c)$

条件：$P(a|b) = \dfrac{P(a,b)}{P(b)}$

Non-parametric

P134.

① Density estimation

求 $p(x)$ (pdf)：

原理：$\begin{cases}\int \varphi(u)du = 1 \\ \varphi(x) \geq 0\end{cases}$

$p(x) \approx P = \dfrac{E[k]}{n} = \boxed{\dfrac{k/n}{V}}$

1) Parzen Windows (P134)　超立方体 (h_n^d)

☆ h_n 为超参 P136：固定 V (参数)，求 k.　任意pdf　$\varphi(u) = \begin{cases}1, |u_j| \leq \frac{1}{2} \\ 0\end{cases}$

$k_n = \sum_{i=1}^{n}\varphi\left(\frac{x-x_i}{h_n}\right)$ → 样本　☆ 待测样本 (即…　边长

φ 可用任意 pdf

⇒ $\boxed{P_n(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{x-x_i}{h_n}\right)}$，$V_n = h_n^d$

求的相当于时 $P(x|w_k)$　→ 只用了类 k 数据.

再用后验决策：$P(w_k|x) = \dfrac{P(w)\; P(x|w)}{P(x)}$ → 常数

2) P143　数数

☆ ③ k_n - Nearest - Neighbor　别忘了求 $P(w)$ 数数

⇒ 人工定 k_n，在 x 中心变大 V_n 直到 k_n 个样本，返回 V_n 算 $P(x|w_j)$.

$P_n(x|w_j)$

$P_n(x) = \dfrac{k_n/n}{V_n}$ → h_n^d → h_n 是有 k_n 个样本 in 时的最小距离

↓

一类一个！ = $P(x|w_j)$，用 $P(w_j|x)$ 判断!

② Decision ⇒ 直接估计 $P(w|x)$.

1) 最近邻 NN P146 ⇒ 选与测试集最近的样本，看同类.

2) kNN k近邻.　最好时都与Bayes同

⇒ 选最近k个点，其中哪个类多选哪个.

复杂度 $O(nd)$

各种距离：P153

Evaluation 补：

① 列表算 每种可能都有一项

② 后处理是求和，t时刻所有状态概率和为序列概率

先写完整公式，再代入计算！ ↓ 有分.