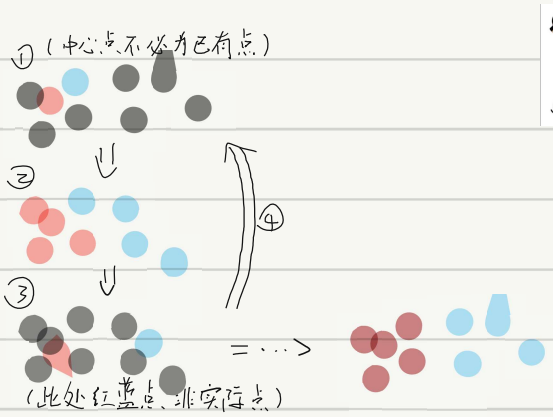
人工智能求解与实践（注：非期末小抄）

K-Means ①（随机）选取K个中心点

②将每个点分配到最近的中心点

③重新计算中心点

④回到②直到中心点收敛



Spectual Clustering

据说变光笔显示有问题
反正画的都是圆点

无向图切图：for $G(V,E)$，子图点集

设交 $\bigcup A_i = V$，$A_i \cap A_j = \phi$

for set $A, B \subset V$, def 权 $W(A,B) = \sum_{i\in A, j\in B} W_{ij}$

(Ncut) def $cut(A_1, A_2, \cdots, A_k) = \frac{1}{2}\sum_{i=1}^{k} W(A, \bar{A_i})$，$\bar{A_i}$：补集 $(V \setminus A_i)$

考虑平衡：$Ncut(A_1, A_2 \cdots A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \bar{A_i})}{vol(A)}$

其中 $vol(A) = \sum_{i\in A} d_i$，$d_i = \sum_{j=1}^{n} w_{ij}$（与一点相连边权和）

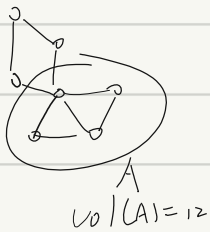
将 $vol(A)$ 替换为 $|A|$，得到 Ratiocut，性能不如 Ncut

卢老师上课以 $N=2$ 为例，且边权都为1，$W(A_1, A_2) = |\{(i,j)\in E; i\in A_1, j\in A_2\}|$

故上述 d_i 为一点的边数

$vol(A)$ 即为2倍A点间的边 + 经过A边界的边

（注，当然A只是点集，但将A看作区域更为直观



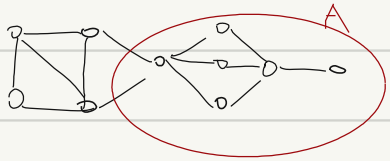
代回 Ncut，$Ncut(A_1, A_2) = \frac{W(A_1, A_2)}{vol(A_1)} + \frac{W(A_1, A_2)}{vol(A_2)}$

$vol(A_1) + vol(A_2) = 2|E|$，

于是卢老师实际给出的简化优化对象为

$$\text{argmin}\ \frac{\text{连结区域A,B的边数}}{\text{较小区域的vol}} = \frac{|\{(i,j)\in E, i\in A, j\notin A\}|}{\min(vol(A), 2m - vol(A))}$$

例



$vol(A) = 16$　$2m - vol(A) = 12$

结果 $\phi = \frac{2}{12} = \frac{1}{6}$

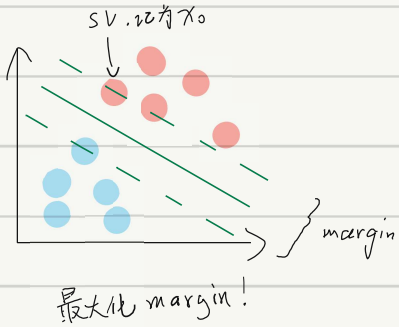
SVM　寻找 $n-1$ 维超平面将 n 个 M 维样本分为2类

卢老师上课时仍先以二分类为例

推导大致过程：设平面 $w_1x_1 + w_2x_2 + b = 0$

经过两个支持向量的平面为 $w_1x_1 + w_2x_2 + b = \pm c$

归一化：三式均除以 c，直线不变，参数变化



最大化 margin！

本笔记为复习期间回顾，独立撰写。如与上课不同，以上课为准

得表达式 $w_1'x_1 + w_2'x_2 + b' = 0$，±1 三个平面

简化表达，以向量形式记为 $w^Tx + b = 0$（注，此处 $w = \begin{pmatrix} \frac{w_1}{c} \\ \frac{w_2}{c} \end{pmatrix}$）

距离公式以 $\frac{w^Tx+b}{\|w\|}$，于是分类 y 为 $\begin{cases} 1, & \frac{w^Tx+b}{\|w\|} \geq d \\ -1, & \frac{w^Tx+b}{\|w\|} \leq -d \end{cases}$，d 为 $\frac{1}{2}$ margin

注意到 $d = \frac{w^Tx_0+b}{\|w\|}$ 未知，而 $\|w\|$ 可调

$w^Tx_0 + b = \|w\| d = \pm 1$，故 $\begin{cases} w^Tx+b \geq 1, & y=1 \\ w^Tx+b \leq -1, & y=-1 \end{cases}$

即 $y(w^Tx_0+b) \geq 1$ ← confidence

SVM 欲使 margin 最大，而 $margin = 2d = 2\frac{|w^Tx+b|}{\|w\|} = 2\frac{y(w^Tx+b)}{\|w\|} = \frac{2}{\|w\|}$

故优化目标 $\arg\max \frac{1}{\|w\|}$ s.t. $y(w^Tx+b) \geq 1, \forall x \in D$

求 $\arg\min \|w\|$ 或 $\arg\min \|w\|^2$

此处进入 SVM 精华部分，然卢老师止步于此（即考试不考）

忆（我考砸了的）数分 Lagrange 乘子法

对优化问题 $\min f(x)$，s.t. $g_n(x) = 0, n = 1, 2 \cdots l$（此 f, g 为向量值函数）

令 $L(x, \lambda) = f(x) + \sum_{n=1}^{l} \lambda_n g_n(x)$

可能极值点为 $\frac{dL}{dx_i}, \frac{dL}{d\lambda_i}$ 的零点

于是 SVM 优化目标 $\min f(w) = \min \|w\|^2$ s.t. $g_i(w) = 1 - y(x_i + b) \leq 0$

记 $L(w, \lambda) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{l} \lambda_i g_i(w)$

要求 $\min \frac{1}{2}\|w\|^2$，由 $g_i(w) \leq 0$，即求 $\min_w \max_\lambda L(w, \lambda)$ s.t. $\lambda_i \geq 0$

由强对偶性，化为 $\max_\lambda \min_w L(w, \lambda)$

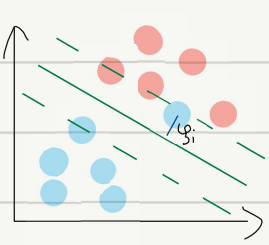
偏导可得 $w = \sum_{i=1}^{l} \lambda_i x_i y_i$，$\sum_{i=1}^{l} \lambda_i y_i = 0$

代回，SMO 求解，此处略去

以上称为 Hard Margin SVM，不可解决 inseparable 问题

引入 slack variable ξ_i，目标改为 $\arg\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i$ s.t. $g_i(w) - \xi_i \leq 0$，$\xi_i \geq 0$

C 为错误分类惩罚因子，ξ_i 为错误程度



要求 $\max_{\lambda,\mu} \min_{w,b,\xi} L(w, b, \xi, \lambda, \mu)$，求法同上

当维度相当低，不足以寻找平面（如  ）时，

升维或利用核函数转化 Kernal trick，代 $x = \varphi(x)$

因升维并不能加快，故用核函数可简化大量计算

如升维 $x' = (x_1^2, \cdots x_n^2, \cdots, \sqrt{2}x_1, \cdots, \sqrt{2}x_n, 1)$

相当于 $\varphi(x) = (x \cdot y + 1)^2$

Spectual Clustering

补充求解过程，方方使，权记为1

先得到邻接(adjacency)矩阵 W 和度矩阵 D，Laplacian 矩阵 $L=D-W$

引入指示向量矩阵，$h_{ij}=\begin{cases}0, & v_i \notin A_j \\ \frac{1}{\sqrt{vol(A_i)}}, & v_i \in A_j\end{cases}$

列向量 h_i，$h_i^T L h_i = \frac{cut(A_i, \bar{A_i})}{vol(A_i)}$，$Ncut(A_1, A_2 \cdots A_i) = tr(H^T L H)$

卢老师的二分类情况，用了更为简单的表达

令 $x_i=\begin{cases}1, & i \in A \\ -1, & i \notin A\end{cases}$，$x$ 为一些由 ± 1 组成的向量

但二元化使同一类所有点贡献相同，不能很好区分中间点和边界点。

于是不妨 $x_i > 0$，$i \in A$

优化 $\arg\min_x f(x)=\sum_{(i,j)\in E}(x_i-x_j)^2$ s.t. $\sum x_i^2=0$（归一化条件），$\sum x_i=0$

化简 $f(x)=\sum_{(i,j)\in E}(x_i^2+x_j^2-2x_i x_j)$，结合度的定义

$=\sum_{i=1}^{n} D_{ii}x_i^2 - \sum_{(i,j)\in E} 2x_i x_j$，再由 W 定义

$=\sum_{i=1}^{n}(D_{ii}x_i^2 - W_{ij}x_i x_j) = \sum_{i=1}^{n} L_{ij}x_i x_j$

$= X^T L X$，st $x^T x = 1$，$\sum x_i = 0$

即优化 $\arg\min_x \frac{x^T L x}{x^T x}$ s.t. $\sum x_i = 0$

之后用线代知识求解，略

Expectation - Maximization

发现时间不足，开始简略

EM 在 K-Means 加入 covariance σ^2

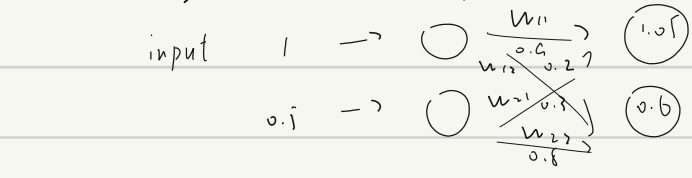
求解 Caussian(μ, σ^2)，流程与 K-Means 一致

补充知识点：最大似然估计

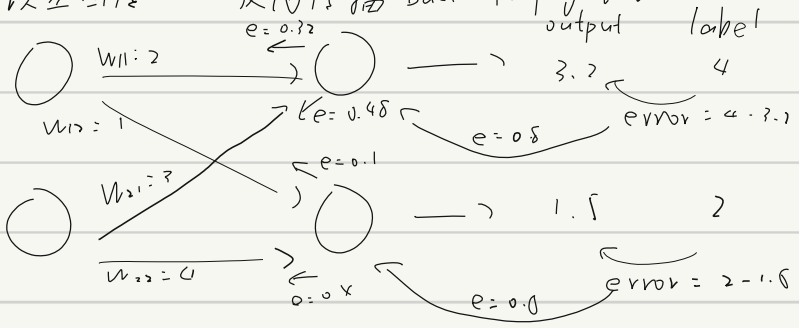
Neural Networks

还不会的，埋个吧

FC 计算过程 $a^{(n+1)} = \sigma(W a^{(n)} + b)$（此处用 sigmoid $y=\frac{1}{1+e^{-x}}$）由上一层算下一层 前向传播



② 误差回传　反向传播 Back Propagation



③ 更新权重

误差对权求导. $\frac{\partial E}{\partial w_{ij}} = \frac{\partial (l_j - o_j)^2}{\partial w_{ij}} = -2(l_j - o_j)\frac{\partial o_j}{\partial w_{ij}}$

$= -2(l_j - o_j) \cdot \frac{\partial \text{sigmoid}(\sum_i w_{ij} \cdot o_i)}{\partial w_{ij}}$

又 $\sigma'(x) = \sigma(x)(1-\sigma(x))$

于是 $\frac{\partial E}{\partial w_{ij}} = -2(l_j - o_j) \cdot \sigma(\sum_i w_{ij} \cdot o_i)(1 - \sigma(\sum_i w_{ij} \cdot o_i)) o_i$

$\sim e_j \sigma(\sum_i w_{ij} o_i)(1 - \sigma(\sum_i w_{ij} o_i)) o_i$

更新即 new w = old w $- \alpha \frac{\partial E}{\partial w}$. 其中 α 称为学习率

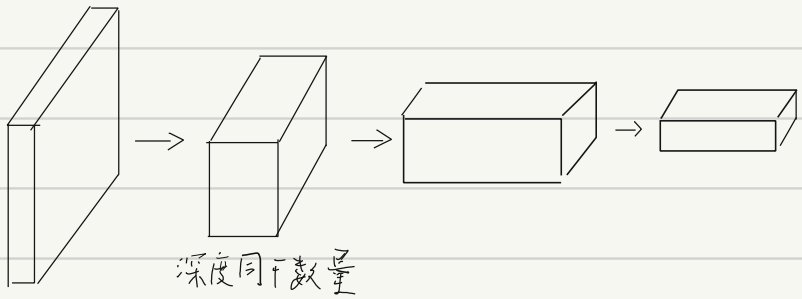
(卷积) CNN

CONV

对应块与Filter值相乘

再套ReLU

池化层：降采样 (2×2取 max)



常用w初始化：W = np.random.randn(in_dim, out_dim)/np.sqrt(in_dim)

应用：LeNet, AlexNet, VGG net,

DenseNet 比 ResNet 节约显存

Naive Bayes

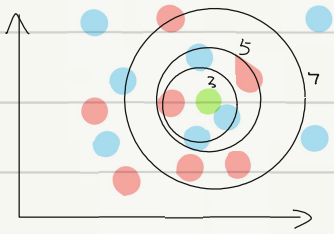
Theorem: $P(B|A) = \frac{P(AB)}{P(A)} = \frac{P(A|B)P(B)}{P(A)}$

So $p(C|x_1, x_2 \cdots x_n) = \frac{P(C) \cdot P(x_1, x_2 \cdots x_n|C)}{P(x_1, x_2 \cdots x_n)}$

$\arg\max_C P(x_1, x_2 \cdots x_n|C)P(C) = \arg\max_C P(C) \prod_i P(x_i|C)$

if features are strongly independent

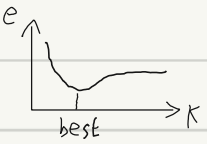
K-Nearest-Neighbours



Green dot is class Blue for K 3.5

Red for K 7

Best K means min error.



Search with K-d Tree

Decision Tree (C4.5)

$Entrophy = -\sum P(i)\log_2 P(i)$ (label为i; $\log_2$ 或换为 $\log_{10}$)

设 data: lable = 0 5个, label = 1 5个,

$H(P) = -(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}) = 1$, 熵大, 不确定程度高. 信息量大

1个9个, $H(P) = -(0.1\log_2 0.1 + 0.9\log_2 0.9) \approx 0.47$

Let $Info(Y) = H(P)$, Y is a result and A is an attribute

$Info_A(Y) = \sum P(i) \cdot Info(Y)$

$Gain(A) = Info(Y) - Info_A(Y)$

$SplitInfo(A) = -\sum P(A)\log P(A)$

$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$

PPT例子挺明白的

建树: ① max info gain attribute

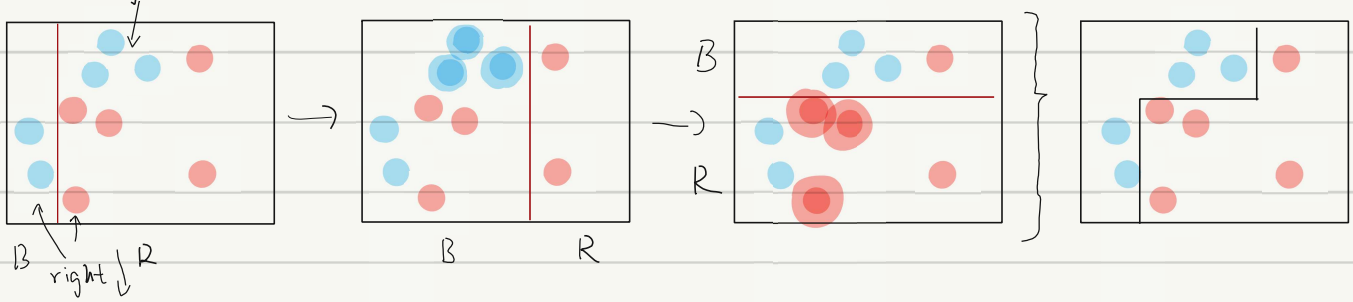
② delete selected new node

③ recalculate

(实际上, 先挑出ratio高于平均值的, 再算最高的)

Adaboost: 与bagging (RF所属算法) 并列, 增强 (boosting) 之一

思想: 相信弱分类器堆料可以成为强分类器 (AMD战胜Intel)

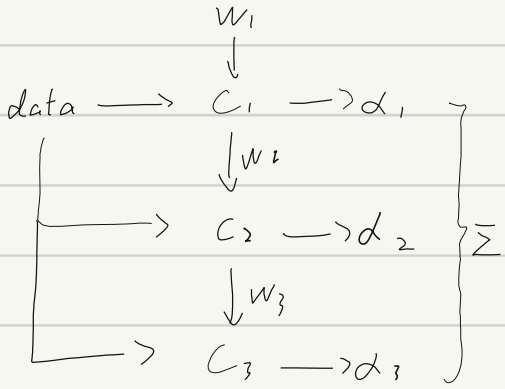


① initialize $w_1 = \frac{1}{N}$, equally contribute

② for range (Num_of_c)

fit c to all data with w

$$error = \frac{\sum_{i=1}^{N} w_i I(\text{wrong ans of } c)}{\sum_{i=1}^{N} w_i}$$



if error > 0.5 abort it

else compute $\alpha = \log((1-err)/err)$

if wrong prediction for dot x_i, $w_i * = e^{\alpha m}$ ↑

if prediction correct for dot x_j. $w_1 * = e^{-\alpha m}$ ↓

③ output = $\Sigma \alpha C(x)$

PageRank

PR(A) 指，每人的堂均分. A得票数，L(A)为A投给的人数

$$PR(A) = \Sigma \frac{PR(x)}{L(x)}$$

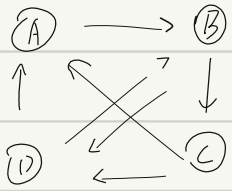
修正系数 d，以免全变成0等情况

$$PR(A) = \left(\Sigma \frac{PR(x)}{L(x)}\right) d + \frac{1-d}{N}$$

化为矩阵形式 $P_{i+1} = dSP_i + (1-\alpha)P_0$。

S为边权矩阵，如例

$$S = \begin{pmatrix} 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$



$$P_0 = \frac{\vec{e}}{N} = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{pmatrix}$$

S列之和为1，P_0和为1

令 $A = dS + (1-\alpha)e\frac{e^T}{N}$ （P_i列之和也为1）

于是 $P = \lim_{n\to\infty} A^n P_0$，由线代知识. $P = AP$，解特征向量即可

基础知识 list

① 监督与非监督

② 分类与回归

③ Train: samples used to train validation: calculate error (k-fold validation)

test: see final performance

④ distance