K-Means (K clusters, unlabeled input)

Algorithm: Randomly choose K centroids

Repeat:

for each data:

choose closest centroid

for each cluster:

update centroid with mean

Until converage.

Spectral (谱线) Clustering

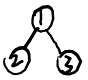
Goal: Divide into 2 disjoint groups

* Good Cluster? Cut less edge & sides balanced

Minimize: $\phi(A) = \frac{|E|}{\min(vol(A), vol(B))}$

Notice: $vol(A) + vol(B) = 2m$, vol: degrees remained

Simplification



A	1	2	3
1	0	1	1
2	1	0	0
3	1	0	0

A: adjacency matrix

D: degree matrix

L: Laplacian matrix

D	1	2	3
1	2	0	0
2	0	1	0
3	0	0	1

* $L = D - A$
* $n \times n$ symmetric matrix
* $L \cdot \begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix} = 0 = 0 \cdot \begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix} \Rightarrow \lambda_1 = 0$

L	1	2	3
1	2	-1	-1
2	-1	1	0
3	-1	0	1

* Eigenvalues: $\lambda \geq 0$, $\lambda \in \mathbb{R}$

Eigenvectors: $\vec{x} \in \mathbb{R}^n$, $(\vec{x_i}, \vec{x_j}) = 0$

Target: $\vec{x} = \underset{x_i \in \mathbb{R}}{\arg\min} \sum_{(i,j) \in E} (x_i - x_j)^2$ (for minimization)

Limit: $\sum x_i^2 = 0$, $\sum x_i = 0$ (for balance)

Finally: $\vec{x} = \underset{x_i \in \mathbb{R}}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{n} L_{ij} x_i x_j$

Minimize: $x^T L x$ $(x^T x = 0)$

$\min_x \frac{x^T L x}{x^T x} = \min_y \frac{y^T \Lambda y}{y^T y}$, $\Lambda = P^T L P$ (正交变换)

$= \min_y \sum_{i=1}^{n} \lambda_i y_i^2$ Λ为特征值对角阵

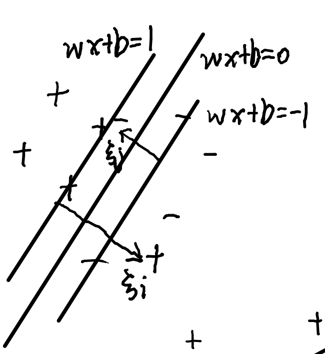
$= \lambda_{min}$ (最小非零特征值)

此时 $y = \varepsilon_i$, $x = P^{-1} y$ 即为所求

Support Vector Machine

Line L: $w \cdot x + b = 0$, Point A

$d(A, L) = \frac{|wA + b|}{\|w\|}$



Split: L: $w \cdot x + b = 0$

Perdiction: $\text{sign}(w \cdot x + b) \overset{def}{=} y$

Confidence: $(w \cdot x + b) \cdot y$ (>0)

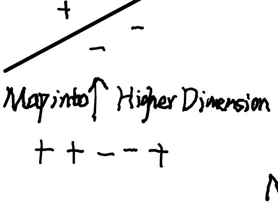
Margin: $\gamma \overset{def}{=} \left(\frac{w \cdot x + b}{\|w\|}\right) \cdot y$

We define $|w \cdot x + b| = 1$ (Reasonable, don't care)

Thus $\gamma = \frac{1}{\|w\|}$, we should maximize it

We may as well minimize $\|w\|$, namely

$\min_w \frac{1}{2}\|w\|^2$ s.t. $\forall i$: $y_i(w \cdot x_i + b) \geq 1$



Not seperable? slack variables ξ_i

$\min_{w, b, \xi_i \geq 0} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i$ s.t. $\forall i: y_i(w x_i + b) \geq 1 - \xi_i$

C is regularization parameter. Or: Map into higher dimension

C4.5 (Classification, Trained)

Which attribute differs best? 「Gain Ratio」 determines

For $\{Y_i\}$ with m classes and distribution $\{p_i\}$:

Entropy (熵) $H(P) = -\sum_{i=1}^{m} p_i \ln p_i = -\sum_{i=1}^{m} \frac{|Y_i|}{|Y|} \ln \frac{|Y_i|}{|Y|}$

Information $Info(Y) = H(P)$ (越大越混乱)

After classification by attribute A with k classes:

$Info_A(Y) = \sum_{j=1}^{k} \frac{|Y_j|}{|Y|} Info(Y_j)$

$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} = \frac{Info(Y) - Info_A(Y)}{Info(A)}$

For example: $Info(out) = -\frac{9}{14}\ln\frac{9}{14} - \frac{5}{14}\ln\frac{5}{14} = 0.940$

$Info_{wind}(out) = -\frac{8}{14}\left(\frac{2}{8}\ln\frac{2}{8} + \frac{6}{8}\ln\frac{6}{8}\right) - \frac{6}{14}\left(\frac{3}{6}\ln\frac{3}{6} + \frac{3}{6}\ln\frac{3}{6}\right) = 0.892$

$SplitInfo(wind) = -\frac{8}{14}\ln\frac{8}{14} - \frac{6}{14}\ln\frac{6}{14} = 0.985$

$GainRatio(wind) = \frac{0.940 - 0.892}{0.985} = 0.049$

Then build the tree: 1. Select maximal GainRatio attribute from pool

2. Remove it from pool 3. Generate new node

4. For each partition, if not completed, back to 1

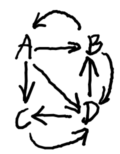
PageRank $PR(C) = \frac{PR(A)}{L(A)} + \frac{PR(B)}{L(B)}$

* Random Surfur: $PR(u) = \alpha \sum_{v \to u} \frac{PR(v)}{L(v)} + \frac{1-\alpha}{N}$

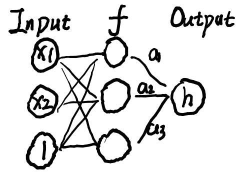
Let $P_0 = \frac{e}{N}$, $P = (PR(1), PR(2), \cdots, PR(n))$, S: transition matrix

We have $P_{i+1} = \alpha S P_i + (1-\alpha) P_0$

Example: $S = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 1 & 0 \end{pmatrix}$, $P_0 = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{pmatrix}$



Deep Learning (Neural Network)



前向传播 $a_1 = f(w_{11}x_1 + w_{12}x_2 + b_1)$

$a_2 = f(w_{21}x_1 + w_{22}x_2 + b_2)$

$h(x) = f(w'_{11}a_1 + w'_{12}a_2 + b'_1)$

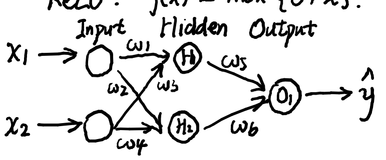
Or we say: $z_i^{(n+1)} = \sum w_{ij}^{(n)} x_j + b_i^{(n)}$

$a_i^{(n)} = f(z_i^{(n)})$

激活函数 BP: Sigmoid: $f(z) = \frac{1}{1+e^{-z}}$, $f'(z) = f(z)(1 - f(z))$

ReLU: $f(x) = \max\{0, x\}$.

反向传播:



$E = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(y - O_1)^2$

$O_1 = f(o_1)$, $do_1 = \frac{\partial E}{\partial O_1} \cdot \frac{\partial O_1}{\partial o_1}$

$\therefore do_1 = -(y - O_1) \cdot f(o_1) \cdot (1 - f(o_1))$

$o_1 = w_5 H_1 + w_6 H_2 + b$, $d\omega_5 = \frac{\partial E}{\partial o_1} \cdot \frac{\partial o_1}{\partial \omega_5} = do_1 \cdot H_1$

$db = \frac{\partial E}{\partial o_1} \cdot \frac{\partial o_1}{b} = do_1$, $d\omega_6 = \frac{\partial E}{\partial o_1} \cdot \frac{\partial o_1}{\partial \omega_6} = do_1 \cdot H_2$

$dH_1 = \frac{\partial E}{\partial o_1} \cdot \frac{\partial o_1}{\partial H_1} = do_1 \cdot \omega_5$, $d\omega_1 = \frac{\partial E}{\partial H_1} \cdot \frac{\partial H_1}{\partial \omega_1} = dH_1 \cdot x_1$

同理有 $dH_2, d\omega_2, d\omega_3, d\omega_4$

Update: $\omega_i += \eta \cdot d\omega_i$, η: Learning Rate

KNN (Supervised, Classification)

Algorithm: Find distance to each point

Sort all the distanc

Take closest K points

$\hat{Y}(x) = \frac{1}{K} \sum_{x_i \in N_k(x)} y_i$

With K-D Tree optimization: $O(\log N) \sim O(N)$

Naive Bayes (Data Mining)

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

$\hat{v} = \underset{v_j \in V}{\arg\max} P(v_j | a_1 a_2 \cdots a_n) = \underset{v_j \in V}{\arg\max} P(a_1 a_2 \cdots a_n | v_j) P(v_j)$

$= \underset{v_j \in V}{\arg\max} P(v_j) \prod_i P(a_i | v_i)$ (独立性)