

	Time	Space	Completeness	Optimality
BFS	b^d	b^d	Y	Y
DFS	b^d	$b \cdot d$	N	N
DFS w Iter Dpng	b^d	$b \cdot d$	Y	Y
Best First Greedy	b^d	b^d	N	N
Best First Astar	b^d	b^d	Y	Y
Minimax	b^d	$b \cdot d$	Y (if finite tree)	Y (if optimal opponent)

Nodes	Cond1	Cond2
Grandpa	Alpha (Max)	Beta (Min)
Parent	Beta (Min)	Alpha (Max)
Condition	Alpha \geq Beta	Beta \leq Alpha
Action	Stop searching under Parent	Stop searching under Parent

Apriori: Frequent itemset has frequent subsets, infrequent itemset has infrequent supersets

- Closed Frequent Itemset: all immediate superset have less support
- Maximal Frequent Itemset: no frequent superset

FP Growth: remove infrequent, order, then grow the tree; **Conditional Pattern Base:** to go from bottom to top to list all parent path of an item.

Best First Search (greedy) $f(n) = g(n) + h(n)$, $g(n) = 0$. if $g(n)$ not 0, A^ .*

$$Support(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{|T|} = P(X \cup Y); \text{ Condidence}(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} = \frac{P(X \cup Y)}{P(X)} = P(Y|X);$$

Two tasks in ARM : 1.Frequent Itemset Generation 2.Rule generation

$$Lift(X, Y) = \frac{Conf(X \rightarrow Y)}{Sup(Y)} = \frac{P(Y|X)}{P(Y)}. \quad (= 1 \Rightarrow \text{independent}; > 1 \Rightarrow \text{posi corr}; < 1 \Rightarrow \text{neg corr})$$

$$** ID3 ** \quad Info(D) = - \sum_{i=1}^m p_i \log_2(p_i); \quad Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} Info(D_j); \quad Gain(A) = Info(D) - Info_A(D)$$

$$** C4.5 ** \quad SplitInfo_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} Info\left(\frac{|D_j|}{|D|}\right); \quad GainRatio_A(D) = \frac{Gain(A)}{SplitInfo_A(D)}$$

$$Gini(D) = 1 - \sum_{i=1}^2 p_i^2; \quad Gini_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} Gini(D_j)$$

$$sensitivity = \frac{TP}{P}; \quad specificity = \frac{TN}{N}; \quad precision = \frac{TP}{TP + FP}; \quad recall = \frac{TN}{TN + FN}$$

$$acc = sensitivity\left(\frac{P}{P + N}\right) + specificity\left(\frac{N}{P + N}\right); \quad F = \frac{2 * precision * recall}{precision + recall}$$

Clustering : Euclidean distance : $d(x, y) = \|x - y\|_2$; Manhattan : $d(x, y) = \|x - y\|_1$;

$$Infinity : \max_{1 \leq j \leq d} |x_j - y_j| = \lim_{t \rightarrow \infty} \left(\sum_{j=1}^d (x_j - y_j)^t \right)^{\frac{1}{t}}$$

BP : output units : $\delta_k = \sigma'(net_k)(t_k - o_k)$; hidden units : $\delta_j = \sigma'(net_j) \sum_k \delta_k w_{kj}$; $\Delta b_j = \eta_b \delta_j$;

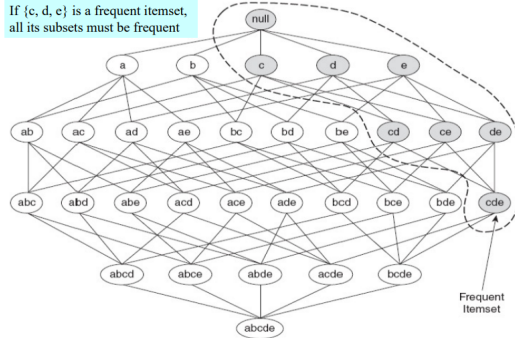
Bayes' Theorem : $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ (under assumption : $P(a_1, \dots, a_d|v_j) = \prod_{i=1}^d P(a_i|v_j)$)

$$PCA : \max_{(\|V\|_2=1)} \frac{1}{N} \sum_{i=1}^N (v^T x_i)^2; \text{ Singular Value Decomposition } A = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T,$$

where U and V are orthogonal and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n), \sigma_1 > \sigma_2 > \dots > \sigma_n > 0$

Illustration 1: Apriori principle

If {c, d, e} is a frequent itemset, all its subsets must be frequent



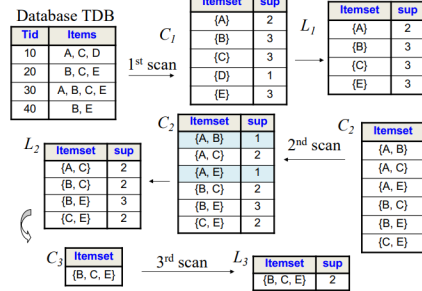
36

The Apriori Algorithm—Example

minsup = 2

C_k : candidate k -itemsets

L_k : frequent k -itemsets.



41

Example: FP-Tree Construction

TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Sort frequent items in frequency descending order, F-list
3. Scan DB again, construct FP-tree

Header Table

Item	frequency	head
f	4	
a	3	
b	3	
m	3	
p	3	

F-list=f-c-a-b-m-p

48

SVM: Margin

- Let the distance of the two hyperplanes, which are parallel to the decision boundary and cross the two closest samples, be

$$\mathbf{w} \cdot \mathbf{x}_a + b = k > 0$$

$$\mathbf{w} \cdot \mathbf{x}_c + b = k' < 0$$

- We can rescale the parameters \mathbf{w} and b to express the two hyperplanes as:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

$$\mathbf{w} \cdot \mathbf{x}_a + b = 1$$

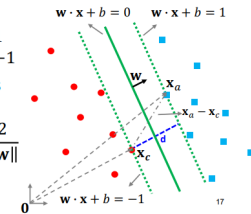
$$\mathbf{w} \cdot \mathbf{x}_c + b = -1$$

- The margin d can be obtained as

$$\mathbf{w} \cdot (\mathbf{x}_a - \mathbf{x}_c) = 2$$

$$\|\mathbf{w}\| \times d = 2 \Rightarrow d = \frac{2}{\|\mathbf{w}\|}$$

- Samples \mathbf{x}_a and \mathbf{x}_c are called support vectors.



Linear SVM Model

- SVM learns the parameters \mathbf{w} and b of the decision boundary to satisfy all training samples, as follows:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 \text{ if } y_i = 1, \Rightarrow y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1,$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \text{ if } y_i = -1, \Rightarrow y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \leq -1, \quad i = 1, 2, \dots, N.$$

- Besides, the margin d of the decision boundary must be maximum:

$$\max d = \frac{2}{\|\mathbf{w}\|} \Rightarrow \min \frac{\|\mathbf{w}\|^2}{2}$$

- The SVM can be learned by solving the following constrained optimization problem with the standard Lagrange multiplier method.

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, N.$$

18

- With the new training samples P and N , the decision boundary D_1 no longer satisfies the original constraints:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N.$$

- The inequality constraints can be relaxed to accommodate the deviation by using a slack variable $\xi_i > 0$:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 - \xi_i \text{ if } y_i = 1,$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 + \xi_i \text{ if } y_i = -1, \Rightarrow y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$$

20

SVM: Non-separable Cases

- Hence, for non-separable cases, SVM learning can be formulated as the following constrained optimization problem with a slack variable:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2} + C \left(\sum_{i=1}^n \xi_i \right)^k$$

$$\text{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N$$