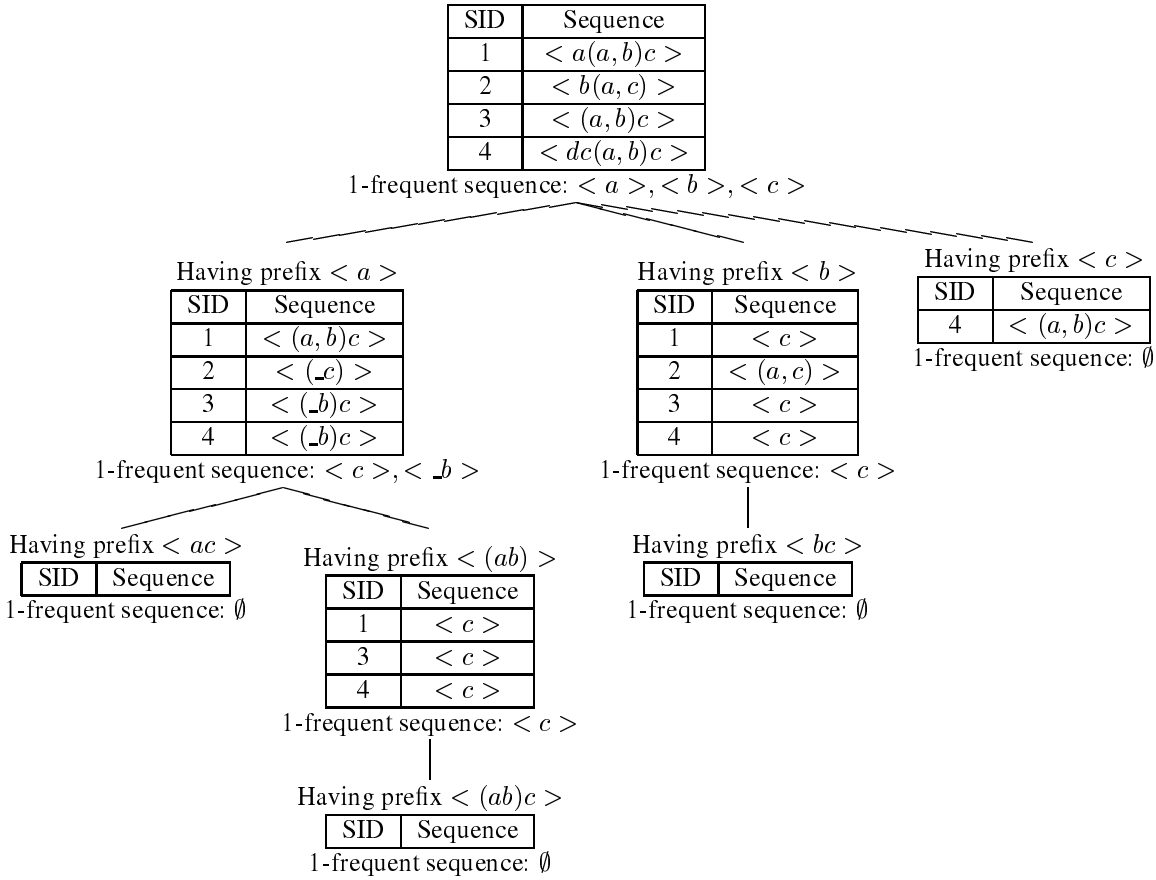
Mid-Term Exam for Data Mining (Spring of 2008)
Instructor: Kyuseok Shim

1. Consider the following sequence database. Assume we have the minimum support of 2 sequences.

Sequence ID	Sequence
1	$< a(a,b)c >$
2	$< b(a,c) >$
3	$< (a,b)c >$
4	$< dc(a,b)c >$

- Find all frequent sequence using PrefixSpan algorithm ran for the above sequence data. Show the detailed steps. [10 pts.]

SOL: PrefixSpan generates the following projected databases for the given input and the minimum support of 2.



Thus, there are 7 frequent sequences, $< a >, < b >, < c >, < ac >, < (ac) >, < bc >$ and $< (ab)c >$.

2. Consider the following market basket database. Assume we have the minimum support of 2 transactions.

Transaction ID	Transaction
1	(a, c, d, f)
2	(b, c, e)
3	(a, b, c, e)
4	(b, e)
5	(a, f)

- Find all frequent itemsets using Apriori-style algorithm ran for the above sequence data. Show the detailed steps. [10 pts.]
- Find all association rules from the frequent itemsets obtained by your answer. Show the detailed steps. Assume that minimum confidence is 50%. [5 pts.]

SOL: Apriori-style algorithm are conducted like the following for the given sequence data and the minimum support of 2.

TID	Transaction
1	(a, c, d, f)
2	(b, c, e)
3	(a, b, c, e)
4	(b, e)
5	(a, f)

$\xrightarrow{\text{1st scan}}$

Itemset	Support
a	3
b	3
c	3
d	1
e	3
f	2

C_1

$\xrightarrow{\text{Remove infrequent itemsets}}$

Itemset	Support
a	3
b	3
c	3
e	3
f	2

L_1

$\xrightarrow{\text{Generate } C_2}$

Itemset
ab
ac
ae
af
bc
be
bf
ce
cf
ef

C_2

$\xrightarrow{\text{2nd scan}}$

Itemset	Support
ab	1
ac	2
ae	1
af	2
bc	2
be	3
bf	0
ce	2
cf	1
ef	0

C_2

$\xrightarrow{\text{Remove infrequent itemsets}}$

Itemset	Support
ac	2
af	2
bc	2
be	3
ce	2

L_2

$\xrightarrow{\text{Generate } C_3}$

Itemset
bce

C_3

$\xrightarrow{\text{3rd scan}}$

Itemset	Support
bce	2

L_3

Thus, there are 11 frequent itemsets, $\{a\}, \{b\}, \{c\}, \{e\}, \{f\}, \{a, c\}, \{a, f\}, \{b, c\}, \{b, e\}, \{c, e\}$ and $\{b, c, e\}$. For the frequent itemsets with at least 2 of length among them, Apriori-style algorithm generates the following association rules with minimum confidence of 50%.

(a) For $\{a, c\}$, we have two candidates of association rule.

$a \Rightarrow c$ with confidence $= \frac{\text{support}\{a,c\}}{\text{support}\{a\}} = \frac{2}{3} > 50\%$.

$c \Rightarrow a$ with confidence $= \frac{\text{support}\{a,c\}}{\text{support}\{c\}} = \frac{2}{3} > 50\%$.

Thus, both candidates are real association rules.

(b) For $\{a, f\}$, we have two candidates of association rule.

$a \Rightarrow f$ with confidence $= \frac{\text{support}\{a,f\}}{\text{support}\{a\}} = \frac{2}{3} > 50\%$.

$f \Rightarrow a$ with confidence $= \frac{\text{support}\{a,f\}}{\text{support}\{f\}} = \frac{2}{3} > 50\%$.

Thus, both candidates are real association rules.

(c) For $\{b, c\}$, we have two candidates of association rule.

$b \Rightarrow c$ with confidence $= \frac{\text{support}\{b,c\}}{\text{support}\{b\}} = \frac{2}{3} > 50\%$.

$c \Rightarrow b$ with confidence $= \frac{\text{support}\{b,c\}}{\text{support}\{c\}} = \frac{2}{3} > 50\%$.

Thus, both candidates are real association rules.

(d) For $\{b, e\}$, we have two candidates of association rule.

$b \Rightarrow e$ with confidence $= \frac{\text{support}\{b,e\}}{\text{support}\{b\}} = \frac{3}{3} > 50\%$.

$e \Rightarrow b$ with confidence $= \frac{\text{support}\{b,e\}}{\text{support}\{e\}} = \frac{3}{3} > 50\%$.

Thus, both candidates are real association rules.

(e) For $\{c, e\}$, we have two candidates of association rule.

$c \Rightarrow e$ with confidence $= \frac{\text{support}\{c,e\}}{\text{support}\{c\}} = \frac{2}{3} > 50\%$.

$e \Rightarrow c$ with confidence $= \frac{\text{support}\{c,e\}}{\text{support}\{e\}} = \frac{2}{3} > 50\%$.

Thus, both candidates are real association rules.

(f) For $\{b, c, e\}$, we have three candidates of association rule with 1 length consequence.

$\{b, c\} \Rightarrow e$ with confidence $= \frac{\text{support}\{b,c,e\}}{\text{support}\{b,c\}} = \frac{2}{2} > 50\%$.

$\{b, e\} \Rightarrow c$ with confidence $= \frac{\text{support}\{b,c,e\}}{\text{support}\{b,e\}} = \frac{2}{3} > 50\%$.

$\{c, e\} \Rightarrow b$ with confidence $= \frac{\text{support}\{b,c,e\}}{\text{support}\{c,e\}} = \frac{2}{2} > 50\%$.

Thus, all the 3 candidates are real association rules and then we can generate 3 candidates of association rules with 2 length consequence.

$b \Rightarrow \{c, e\}$ with confidence $= \frac{\text{support}\{b,c,e\}}{\text{support}\{b\}} = \frac{2}{3} > 50\%$.

$c \Rightarrow \{b, e\}$ with confidence $= \frac{\text{support}\{b,c,e\}}{\text{support}\{c\}} = \frac{2}{3} > 50\%$.

$e \Rightarrow \{b, c\}$ with confidence $= \frac{\text{support}\{b,c,e\}}{\text{support}\{e\}} = \frac{2}{3} > 50\%$.

Thus, all the 3 candidates are real association rules.

Overall, we have 16 association rules.

3. Consider the following database. Assume we have the minimum support of 2 transactions.

Salary	Company	Class Label
40000	KTF	GOOD
15000	KTF	BAD
75000	SKT	GOOD
18000	SKT	GOOD

- Show the decision tree generated by building phase. Show the detailed steps including gini indexes. [10 pts.]
- Compute the MDL cost of the generated decision tree. Assume that the cost of pure leaf node with a single class label is ZERO. [5 pts.]

SOL: We first build a sorted attribute list for each attribute and find the best split point(condition).

For the attribute Salary, we have the following attribute list.

Salary	Class Label	RID
15,000	BAD	2
18,000	GOOD	4
40,000	GOOD	1
75,000	GOOD	3

Then, we have 4 split points, Salary $< 15,000$, Salary $> 15,000$, Salary $> 18,000$ and Salary $> 40,000$.

For Salary $< 15,000$, gini index $= \frac{0}{4} \times \{1 - (0^2 + 0^2)\} + \frac{4}{4} \times \{1 - (0.25^2 + 0.75^2)\} > 0$.
For Salary $> 15,000$, gini index $= \frac{1}{4} \times \{1 - (1^2 + 0^2)\} + \frac{3}{4} \times \{1 - (0^2 + 1^2)\} = 0$.
For Salary $> 18,000$, gini index $= \frac{2}{4} \times \{1 - (0.5^2 + 0.5^2)\} + \frac{2}{4} \times \{1 - (0^2 + 1^2)\} > 0$.
For Salary $> 40,000$, gini index $= \frac{3}{4} \times \{1 - (0.33^2 + 0.67^2)\} + \frac{1}{4} \times \{1 - (0^2 + 1^2)\} > 0$.

Among them, the best one is Salary $> 15,000$ whose gini index is 0.

For the attribute Company, we have the following attribute list.

Company	Class Label	RID
KTF	GOOD	1
KTF	BAD	2
SKT	GOOD	3
SKT	GOOD	4

Then, we have 2 split conditions, Company $\in \{KTF, SKT\}$ and Company $\in \{KTF\}$.

For Company $\in \{KTF, SKT\}$, gini index $= \frac{0}{4} \times \{1 - (0^2 + 0^2)\} + \frac{4}{4} \times \{1 - (0.25^2 + 0.75^2)\} > 0$.
For Company $\in \{KTF\}$, gini index $= \frac{2}{4} \times \{1 - (0.5^2 + 0.5^2)\} + \frac{2}{4} \times \{1 - (0^2 + 1^2)\} > 0$.

Since both conditions have larger gini index than 0, finally the best split condition is Salary $> 15,000$ with gini index of 0. Then, we partition attribute lists using this split condition and we have the following lists.

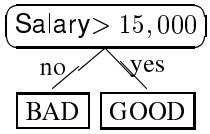
Salary	Class Label	RID
15,000	BAD	2

Company	Class Label	RID
KTF	BAD	2

Salary	Class Label	RID
18,000	GOOD	4
40,000	GOOD	1
75,000	GOOD	3

Company	Class Label	RID
KTF	BAD	2
SKT	GOOD	4
KTF	GOOD	1
SKT	GOOD	3

Since each partition is pure, building phase stops here and we get the following decision tree.



Then, the MDL cost of this decision tree is

$$\begin{aligned} & 3 && \text{: tree describing cost as 100} \\ + \quad & \log_2 2 + \log_2(4-1) && \text{: split condition decribing cost} \\ + \quad & 0 && \text{: data describing cost. Since both leaf nodes are pure, it need 0 bits.} \\ = \quad & 4 + \log_2 3 \\ \cong \quad & 5.58 \\ = \quad & 6bits \end{aligned}$$

4. Consider the following database. For k=2, show the detailed steps of K-means algorithm using L_2 distance. Assume that initial two centers (means) were selected (1,1) and (6.6). [10 pts.]

Point
(1,1)
(1,2)
(5,5)
(5,6)
(6,6)

SOL: Let C_1 and C_2 be two initial clusters centered at (1,1) and (6,6), respectively. K-means algorithm first assigns data points to closer center between (1,1) and (6,6). Then, we have the following result.

Point	Distance from (1,1)	Distance from (6,6)	Assigned Cluster
(1,1)	0	$\sqrt{50}$	C_1
(1,2)	1	$\sqrt{41}$	C_1
(5,5)	$\sqrt{32}$	$\sqrt{2}$	C_2
(5,6)	$\sqrt{41}$	1	C_2
(6,6)	$\sqrt{50}$	0	C_2

Then, we compute new centers for both clusters and we get (1, 1.5) for C_1 and (16/3, 17/3) for C_2. We again assign data points to new centers and then we get the following result.

Point	Distance from (1,1.5)	Distance from (16/3,17/3)	Assigned Cluster
(1,1)	0.5	$\sqrt{365/9}$	C_1
(1,2)	0.5	$\sqrt{290/9}$	C_1
(5,5)	$\sqrt{28.25}$	$\sqrt{5/9}$	C_2
(5,6)	$\sqrt{36.25}$	$sqrt2/9$	C_2
(6,6)	$\sqrt{45.25}$	$sqrt5/9$	C_2

Since we have the same clusters with in the previous step, the algorithm stops.

5. Consider the data [1,2,3,4,5,6,7,8].

   - Show the detailed steps of Haar wavelet computation and give the Haar wavelet coefficients. [10 pts.]
   - Assuming we are allowed to select only 2 coefficients, which coefficients should we choose for L_2 distance? [10 pts.]
   - Assuming we are allowed to select only 2 coefficients, which coefficients should we choose for L_∞ distance? [10 pts.]

   SOL: We compute Haar wavelet coefficients for the data [1,2,3,4,5,6,7,8] as follows.

Resolution	Averages	Detail Coefficients
3	[1, 2, 3, 4, 5, 6, 7, 8]	
2	[1.5, 3.5, 5.5, 7.5]	[-0.5, -0.5, -0.5, -0.5]
1	[2.5, 6.5]	[-1, -1]
0	[4.5]	[-2]

   Thus, Haar wavelet coefficients are [4.5, -2, -1, -1, -0.5, -0.5, -0.5, -0.5].

   Then, we normalize this coefficients to get the best 2 coefficients for L_2 distance. The normalized coefficients are $[9\sqrt{2}, -4\sqrt{2}, -2, -2, -\sqrt{2}/2, -\sqrt{2}/2, -\sqrt{2}/2, -\sqrt{2}/2]$. Since the best 2 coefficients are the largest 2 normalized absolute values, they are $9\sqrt{2}$ and $-4\sqrt{2}$. That is, 4.5 and 2 are them in un-normalized Haar wavelet coefficients.

   Finally, in order to get the best 2 coefficients for L_∞, we have to use a dynamic programming algorithm. The detailed algorithm appears in "Wavelet Synopses with Error Guarantees" written by Minos Garafalatis and Philip B. Gibbons. The solution is to select 4.5 and -2.

6. Consider the following time series database. We will use only two wavelet coefficients to represent each sequence and L_2 distance is used.

Sequence ID	Time series
1	$< 1, 2, 3, 4 >$
2	$< 2, 2, 4, 4 >$
3	$< 5, 5, 6, 6 >$
4	$< 5, 5, 7, 7 >$

   - For a query sequence $< 2, 3, 3, 3 >$, show the the detailed steps of how each sequence is represented with dimensionality reduction and how the sequences within ϵ=2 is discovered. [20 pts.]

SOL: First, we compute the wavelet coefficients for each sequences, and normalize them.

For $< 1, 2, 3, 4 >$, through the following steps

Resolution	Averages	Detail Coefficients
2	[1, 2, 3, 4]	
1	[1.5, 3.5]	[-0.5, -0.5]
0	[2.5]	[-1]

we get $[2.5, -1, -0.5, -0.5]$ and if we normalize them, we have $[5, -2, -\sqrt{2}/2, -\sqrt{2}/2]$.

For $< 2, 2, 4, 4 >$, through the following steps

Resolution	Averages	Detail Coefficients
2	[2, 2, 4, 4]	
1	[2, 4]	[0, 0]
0	[3]	[-1]

we get $[3, -1, 0, 0]$ and if we normalize them, we have $[6, -2, 0, 0]$.

For $< 5, 5, 6, 6 >$, through the following steps

Resolution	Averages	Detail Coefficients
2	[5, 5, 6, 6]	
1	[5, 6]	[0, 0]
0	[5.5]	[-0.5]

we get $[5.5, -0.5, 0, 0]$ and if we normalize them, we have $[11, -1, 0, 0]$.

For $< 5, 5, 7, 7 >$, through the following steps

Resolution	Averages	Detail Coefficients
2	[5, 5, 7, 7]	
1	[5, 7]	[0, 0]
0	[6]	[-1]

we get $[6, -1, 0, 0]$ and if we normalize them, we have $[12, -2, 0, 0]$.

Then, we choose the first two coefficients for each data even though there are many possible selections and represent each data using those two coefficients with dimensionality reduction.

Now, we are ready to answer for the given query $< 2, 3, 3, 3 >$. We do the same things to the query. That is, through the following steps,

Resolution	Averages	Detail Coefficients
2	[2, 3, 3, 3]	
1	[2.5, 3]	[-0.5, 0]
0	[2.75]	[-0.25]

we get $[2.75, -0.25, -0.5, 0]$ as Haar wavelet coefficients and if we normalize them, we have $[5.5, -0.5, -sqrt2/2, 0]$. Then, if we select the first two coefficients, we get 5.5 and -0.5. Using these two coefficients, we compute L_2 distance for each data point.

For $< 1, 2, 3, 4 >$, the L_2 distance is $\sqrt{(5 - 5.5)^2 + (-2 + 0.5)^2} \cong 1.58 < 2$. Thus, $< 1, 2, 3, 4 >$ may be within L_2 distance of 2.

For $< 2, 2, 4, 4 >$, the L_2 distance is $\sqrt{(6 - 5.5)^2 + (-2 + 0.5)^2} \cong 1.58 < 2$. Thus, $< 2, 2, 4, 4 >$ may be within L_2 distance of 2.

For $< 5, 5, 6, 6 >$, the L_2 distance is $\sqrt{(11 - 5.5)^2 + (-1 + 0.5)^2} \cong 5.52 > 2$. Thus, $< 5, 5, 6, 6 >$ can not be within L_2 distance of 2.

For $< 5, 5, 7, 7 >$, the L_2 distance is $\sqrt{(12 - 5.5)^2 + (-2 + 0.5)^2} \cong 6.67 > 2$. Thus, $< 5, 5, 7, 7 >$ can not be within L_2 distance of 2.

Finally, we compute the real L_2 distances between the selected data points and query point and check wheter they are within 2.

For $< 1, 2, 3, 4 >$, the L_2 distance is $\sqrt{(1-2)^2 + (2-3)^2 + (3-3)^2 + (4-3)^2} = \sqrt{3} < 2$. For $< 2, 2, 4, 4 >$, the L_2 distance is $\sqrt{(2-2)^2 + (2-3)^2 + (4-3)^2 + (4-3)^2} = \sqrt{3} < 2$.

Thus, we found $< 1, 2, 3, 4 >$ and $< 2, 2, 4, 4 >$ are within L_2 distance of 2 from the query point $< 2, 3, 3, 3 >$.