Solutions Advanced Data Mining

Date: 11-11-2010

Time: 13.30-16.30

Question 1 Multiple Choice (16 points)

For the following questions, zero or more answers may be true.

a) 1,2.

b) 1,4.

c) 2,4.

d) 1,3.

Question 2 Frequent Itemset Mining (25 points)

a) Level 1:

candidate	support	frequent?
A	3	Y
B	5	Y
C	2	Y
D	3	Y
E	2	Y

All 1-itemsets are frequent, so all 2-itemsets are candidates at level 2:

candidate	support	frequent?
AB	3	Y
AC	1	N
AD	1	N
AE	0	N
BC	2	Y
BD	3	Y
BE	2	Y
CD	1	N
CE	1	N
DE	2	Y

All subsets of BDE are frequent, so this is a candidate at level 3:

candidate	support	frequent?
BDE	2	Y

There are no level 4 candidates.

b) An itemset is a generator if it is frequent and has no subset with the same support. Level 1:

candidate	support	generator?
A	3	Y
B	5	Y
C	2	Y
D	3	Y
E	2	Y

All 1-itemsets are generators, so all 2-itemsets are candidates at level 2:

candidate	support	generator?
AB	3	N
AC	1	N
AD	1	N
AE	0	N
BC	2	N
BD	3	N
BE	2	N
CD	1	N
CE	1	N
DE	2	N

Now we have all generators. The next step is to compute their closure. The closure has the same support as the generator. The result is the set of all closed frequent itemsets.

generator	closure	support
A	AB	3
B	B	5
C	BC	2
D	BD	3
E	BDE	2

c) KRIMP codetable (order on size, then on support and finally lexicographically)

itemset	usage	codelength
BDE	2	$-\log \frac{2}{8} = \log 4 = 2$ bits
AB	3	$-\log \frac{3}{8} = 1.415$ bits
BD	0	none
BC	0	none
B	0	none
A	0	none
D	1	$-\log \frac{1}{8} = \log 8 = 3$ bits
C	2	$-\log \frac{2}{8} = \log 4 = 2$ bits
E	0	none

d) Yes. This constraint has the Apriori property that if an itemset satisfies this constraint, then so do all of its subsets. Or, conversely, if an itemset does not satisfy this constraint, then neither does any of its supersets. Hence we can prune any itemset that does not satisfy the constraint in an Apriori style levelwise search.

Question 3 Undirected Graphical Models (15 points)

a) From the definition of pairwise independence for undirected graphs:

1. $A \perp\!\!\!\perp D \mid B, C$
2. $A \perp\!\!\!\perp C \mid B, D$
3. $C \perp\!\!\!\perp D \mid A, B$

Using separation in the graph, these can be strengthened to:

1. $A \perp\!\!\!\perp D \mid B$
2. $A \perp\!\!\!\perp C \mid B$
3. $C \perp\!\!\!\perp D \mid B$

b)
1. $\hat{n}(A, B) = n(A, B)$
2. $\hat{n}(B, C) = n(B, C)$
3. $\hat{n}(B, D) = n(B, D)$

c)

$$\hat{n}(A,B,C,D) = \frac{n(A,B)n(B,C)n(B,D)}{n(B)^2}$$

Question 4 Bayesian Network Classifiers (20 points)

a) The probability estimates produced by naive Bayes may be off, but this does not necessarily harm classification performance. For example, in the binary case, all that matters is whether the class probability estimate is on the correct side of 0.5. Furthermore, since naive Bayes only has few parameters, they can be estimated with relatively high precision.

b) Each attribute except the root has 2 parents: the class label and another attribute. This gives $2m$ parent configurations, and for each parent configuration a single probability has to be estimated. Since there are $k-1$ such attributes this gives $2m(k-1)$ parameters. The attribute with only the class as parent has m probabilities to be estimated, and for the class label itself there are $m-1$. Hence, the total is $2mk-1$.

c) The independence properties of a directed graph are the same as that of its moral graph if no marrying of parents is required (no V-structures). In a TAN each attribute (except the root) has two parents: the class label and another attribute. Since the class label is connected to every attribute, these two parents are already married (no V-structures).

d) The score function of standard structure learning algorithms is

$$\text{BIC}(M|D) = \frac{\ln n}{2}\dim(M) - \mathcal{L}(M|D)$$

where

$$\mathcal{L}(M|D) = \sum_{j=1}^{n} \log P_M(C^{(j)}|A^{(j)}) + \sum_{j=1}^{n} \log P_M(A^{(j)})$$

From the viewpoint of classification only accurate modeling of $P(C|A)$ matters, so the second part of the score, modeling the marginal distribution $P(A)$ of the attributes is irrelevant. When there are many attributes, the probability of each single combination of attribute values becomes small, leading to large negative values in the loglikelihood. On the other hand, the conditional probabilities $P(C|A)$ remain of the same order of magnitude regardless of the number of attributes. Hence the irrelevant component of the score function starts to dominate when the number of attributes becomes large: we run the risk of accurately modeling $P(A)$ at the expense of $P(C|A)$.

e) 1. Compute the conditional mutual information between each pair of attributes.
2. Make a complete graph with the attributes as nodes, and put the mutual information as a weight on the edge between two attributes.
3. Compute a maximum weight spanning tree on this graph.
4. Choose one of its nodes as root, and let the edges point away from it.
5. Add the class variable and edges from the class variable to each attribute.

Question 5 Bayesian Networks (10 points)

a) Neighbours C and D are equivalent:

(A)
$$\begin{tikzcd} 1 \arrow[r] & 2 \arrow[r] & 3 \end{tikzcd}$$

(B)
$$\begin{tikzcd} 1 \arrow[r] & 2 & 3 \arrow[l] \end{tikzcd}$$

(C)
$$\begin{tikzcd} 1 \arrow[r] \arrow[rr, bend left] & 2 & 3 \end{tikzcd}$$

(D)
$$\begin{tikzcd} 1 \arrow[r] & 2 & 3 \arrow[ll, bend right] \end{tikzcd}$$

(E)
$$\begin{tikzcd} 1 & 2 \arrow[l] & 3 \end{tikzcd}$$

(F)
$$\begin{tikzcd} 1 & 2 & 3 \end{tikzcd}$$

b) The current model has 10 parameters: Node 1: 2, Node 2: 6, Node 3: 2. We count the number of parameters of the neighbours:

(A) 2+6+6=14.

(B) 2+18+2=22.

(C) 2+6+6=14.

(D) 6+6+2=14.

The penalty per parameter is $\frac{\ln 100}{2} \approx 2.3$. So, for (A), (C) and (D), the increase should be more than $4 \times 2.3 = 9.2$ and for (B) the increase should be more than $12 \times 2.3 = 27.6$.

Question 6 Classification Trees (14 points)

a) $i(t_1) = 0.4 \times 0.6 = 0.24$, $i(t_2) = 0.75 \times 0.25 = 0.1875$, $i(t_3) = 0.5 \times 0.5 = 0.25$, $\Delta i = 0.24 - 0.4 \times 0.1875 - 0.6 \times 0.25 = 0.015$.

b) T_1 is obtained by pruning in t_7: the resulting tree has the same resubstitution error as $T_{\max}$. Then we have

$$g(t_1) = \frac{0.4 - 0.15}{3} \approx 0.08, \quad g(t_2) = \frac{0.1 - 0.05}{1} = 0.05, \quad g(t_3) = \frac{0.3 - 0.1}{1} = 0.2$$

The minimum is $g(t_2)$, so we prune in t_2. Then we recompute the g values. $g(t_3)$ doesn't have to be recomputed, and

$$g(t_1) = \frac{0.4 - 0.2}{2} = 0.1$$

which is smaller than $g(t_3)$, so we prune in the root.

Summarizing: T_1 is obtained by pruning $T_{\max}$ in t_7. This is the best tree for $\alpha \in [0, 0.05)$. T_2 is obtained by pruning T_1 in t_2. This is the best tree for $\alpha \in [0.05, 0.1)$. Then we prune in the root, which is the best tree for $\alpha \in [0.1, \infty)$.