Data mining, Spring 2010. Exercises 1, solutions

1. An educational psychologist wants to use association analysis to analyze test results. The test consists of 100 multiple choice questions with four possible answers each. How would you convert this data into a form suitable for association analysis?

One option would be giving unique values to each possible selection. There would be then 100x4=400 values altogether. In this case the transaction would be formed as a row with 100 items.

Question no/ Response no	1	2	3	4	5	6	7	...	100
1	C1	D2	D3	B4	A5	A6	B7	...	B100
2	C1	B2	A3	D4	C5	A6	D7	...	C100
3	A1	C2	C3	B4	C5	A6	B2	...	C100

2. Consider the market basket data in the Table 1 below.

Transaction ID	**Items bought**
1	**Milk,Beer,Diapers**
2	**Bread, Butter, Milk**
3	**Milk, Diapers, Cookies**
4	**Bread, Butter, Cookies**
5	**Beer, Cookies, Diapers**
6	**Milk, Diapers, Bread, Butter**
7	**Bread, Butter, Diapers**
8	**Beer, Diapers**
9	**Milk, Diapers, Bread, Butter**
10	**Beer, Cookies**

Table 1: Marker basket transactions

(a) What is the maximum number of association rules that can be extracted from this data (including rules with zero support)?

(b) What is the maximum size (k) of frequent k-itemsets in this data, assuming minsup > 0.

a) 6 unique items, 3^d-2^(d+1)+1 = 3□-2□+1 = 602 possible rules

b) itemset {Milk, Diapers, Bread, Butter} has the biggest size with 4 items and is frequent (sup({Milk, Diapers, Bread, Butter})=2)

3. From the data of Table 1,

(a) Find a itemset with 2 or more items that has the largest support.

(b) Find a pair of items a and b, such that the rules a → b and b → a have the same confidence.

initial supports (in descending order)

Item	**support**
Diapers	7
Milk	5
Bread	5
Butter	5
Beer	4
Cookies	4

All 6 itemsets with one item are frequent.

Sets with two items:

Itemset	support
{Diapers, Milk}	4
{Diapers, Bread}	3
{Diapers, Butter}	3
{Diapers, Beer}	3
{Diapers, Cookies}	2

{Milk, Bread}	3
{Milk, Butter}	2
{Milk, Beer}	1
{Milk, Cookies}	1
{Bread, Butter}	5
{Bread, Beer}	0
{Bread, Cookies}	1
{Butter, Beer}	0
{Butter, Cookies}	1
{Beer, Cookies}	2

Altogether 13 frequent itemsets. Two sets can be pruned out as well as bigger sets with these items.

a) We only need to watch the supports of itemsets with 2 items, since adding items will never grow the support. The support will stay either the same or decrease.

Based on the tables in previous exercise, {Bread, Butter} and {Diapers, Milk} have the largest support.

b) The formula for confidence is:

conf(A->B) = sup({A,B})/sup({A})

It can easilly been seen from the tables in previous exercise that Bread and Butter are having the same initial support and therefore rules Bread → Butter and Butter → Bread are having the same confidence. Proofing this with math :

conf(Bread → Butter) = sup({Bread, Butter})/sup({Bread}) = 5/5 = 1

and

conf(Butter → Bread) = sup({Bread, Butter})/sup({Butter}) = 5/5 = 1

Actually if the denominators are the same then the confidence has to be the same.

Based on this also Milk and Bread, Milk and Butter, Beer and Cookies suffice the condition.

4. Consider the following set of frequent 3-itemsets: {1, 2, 3}, {1, 2, 4}, {1, 2, 5}, {1, 3, 4},

{1, 3, 5}, {2, 3, 4}, {2, 3, 5}, {3, 4, 5}.
Assume that there are only five items in the dataset.

List all candidate 4-itemsets obtained by the Fk−1 × F1 candidate generation method.

Supports for 1-itemsets:

Item	support
1	5

2	5
3	6
4	4
5	4

Candidate 4-itemsets (having lexicographical order pruning):

{1,2,3}: {1,2,3,4}, {1,2,3,5}

{1,2,4}: {1,2,4,5}

{1,2,5}: none, not possible to extend

{1,3,4}: {1,3,4,5}

{1,3,4} : no new ones

{2,3,4} : {2,3,4,5}

{2,3,5} : no new ones

{3,4,5}:none, not possible to extend

5. Consider the same set of frequent 3-itemsets as above. List all candidate 4-itemsets obtained by the Fk−1 × Fk−1 candidate generation method.

When considering merging, only pairs that share first k-2
items are considered
(items in lexicographical order).

{1,2,3}x{1,2,4} : {1,2,3,4}

{1,2,3}x{1,2,5} : {1,2,3,5}

{1,2,4}x{1,2,5} : {1,2,4,5}

{1,3,4}x{1,3,5} : {1,3,4,5}

{2,3,4}x{2,3,5} : {2,3,4,5}

6. –7. Consider the following set of candidate 3-itemsets: {1, 2, 3},{1, 2, 6},{1, 3, 4}, {2, 3, 4}, {2, 4, 5}, {3, 4, 6}, {4, 5, 6},

Construct a hash tree for the itemsets, using a hash function that sends odd-numbered items to the left and even-numbered items to the right.

A candidate itemset is inserted by hashing each successive item in the candidate and following the appropriate branch in the hash tree.

Once a leaf node is reached the candidate is inserted according to the following conditions:

Condition 1: if the depth of leaf equals k (root is on level 0) the candidate is inserted regardless of how many itemsets are already stored at the node.

Condition 2: if the depth of the node is less than k, then the candidate is inserted as long as there are less than maxsize = 2 itemsets already stored at the node.

Condition 3: if the depth of node is less than k and the number of itemsets in the node is maxsize = 2, convert the leaf into an internal node. New leafs are created as the children of the node and the new itemset as well as itemsets previously stored in the node are hashed into the children using the hash function.

Has function h(p) = p mod 2

inserting {1,2,3}

```
                               *
                            /       \
                       {1,2,3}
```

inserting {1,2,6}

```
                               *
                            /       \
                        {1,2,3}
                        {1,2,6}
```

inserting {1,3,4}, new leafs created, new itemset and existing ones hashed again

```
                               *
                          /        \
                         *
                      /       \
                 {1,3,4}   {1,2,3}
                           {1,2,6}
```

inserting {2,3,4}

```
                     *
               /          \
              *          {2,3,4}
         /        \
  {1,3,4}      {1,2,3}
               {1,2,6}
```

inserting {2,4,5}

```
                     *
               /          \
              *          {2,3,4}
         /        \      {2,4,5}
  {1,3,4}      {1,2,3}
               {1,2,6}
```

inserting {3,4,6}, new leafs created, and new itemset as well as the old ones hashed again

```
                      *
               /              \
              *               {2,3,4}
      /             \         {2,4,5}
{1,3,4}              *
               /           \
         {1,2,3}          {1,2,6}
                          {3,4,6}
```

inserting {4,5,6}, new leafs created and new itemset as well as the old ones hashed again

```
                      *
               /                     \
              *                             *
      /             \                    /         \
{1,3,4}              *              {2,3,4}     {2,4,5}
               /           \        {4,5,6}
         {1,2,3}          {1,2,6}
                          {3,4,6}
```