Chapter 7 Association Analysis

7.1 Introduction

Many business enterprises accumulate huge amount of data from their day-to-day transaction records. For example, supermarket's scanner data, bank customer's account transaction etc. **Association rule discovery** or **market-basket analysis** (**mba**) is to find association rules like this:

(item set A) => (item set B)

Or even more complicit rule like:

(item set A and item set B) => (item set C)

Let us consider a simple example with the following transaction records:

```
ID  Items
1   {Bread, Milk}
2   {Bread, Diapers, Beer, Eggs}
3   {Milk, Diapers,Beer, Cola}
4   {Bread, Milk, Diapers, Beer}
5   {Bread, Milk, Diapers, Cola}
```

In this example, there are totally 5 transactions and 6 different items. An alternative binary representation of this example is:

ID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

From this, we can produce the **Co-occurrence matrix**:

```
         beer bread cola diapers eggs milk
beer        3     2    1       3    1    2
bread       2     4    1       3    1    3
cola        1     1    2       2    0    2
diapers     3     3    2       4    1    3
eggs        1     1    0       1    1    0
milk        2     3    2       3    0    4
```

Item frequency:

```
"bread" "diapers" "milk" "beer" "cola" "eggs"
"4"     "4"       "4"    "3"    "2"    "1"
```

Association rules:

```
     "prod1"   "prod2"   "freq"
      "diapers" "beer"    "3"
      "diapers" "bread"   "3"
      "milk"    "bread"   "3"
      "milk"    "diapers" "3"
      "bread"   "beer"    "2"
      "milk"    "beer"    "2"
      "diapers" "cola"    "2"
      "milk"    "cola"    "2"
      "cola"    "beer"    "1"
      "eggs"    "beer"    "1"
      "cola"    "bread"   "1"
      "eggs"    "bread"   "1"
      "eggs"    "diapers" "1"
      "eggs"    "cola"    "0"
      "milk"    "eggs"    "0"
```

There are 3 transactions with *diapers* and *beer* bought together; … ; only 1 transaction with *eggs* and *diapers. Eggs* and *cola* or *milk* and *eggs* were never being bought together. Note that the combinations (*diapers, beer*), (*diapers, bread*), (*milk, bread*) and (*milk, diapers*) are most popular, but we cannot distinguish between the rule (*diapers => beer*) and (*beer => diapers*). To study these rules more closely, we need the concepts of **support, confidence** and **lift** of these rules. Let us illustrate these by the combination (*diapers, beers*). The two-way contingence table is:

	Diapers		
Beer	No	Yes	Total
No	1	1	2
Yes	0	3	3
Total	1	4	5

Support: *S(Beer => Diapers) = Pr{Beer and Diapers} = 3/5 = 0.6*
Confidence: *C(Beer => Diapers) = Pr{Diapers|Beer} = 3/3 = 1*
Confidence: *C(Diapers => Beer) = Pr{Beer|Diapers} = 3/4 = 0.75*

In general, *S(A=>B) = #(A and B) / N; C(A=>B) = #(A and B) / #(A)*
Lift: *L(Beer => Diapers) = C(Beer =>Diapers) / P(Diapers) = 1/0.8 = 1.25*
Lift: *L(Diapers => Beer) = C(Diapers => Beer) / P(Beer) = 0.75/0.6 = 1.25*

The interpretation is that if *Beer* and *Diapers* were independent, then the probability of a customer buys *Diapers* is 4/5=0.8; and *C(Beer =>Diapers) / P(Diapers)* is the ratio of the confidence of this rule compared with the probability of buying *Diapers* if these two items were independent. Similarly, the probability of a customer buys *Beer* is 3/5=0.6; and the lift *L(Diapers => Beer) = C(Diapers => Beer) / P(Beer)* is the ratio of the confidence of this rule compared with the probability of buying *Beer* if these two items were independent. Note that the lift can also be computed using (3x5)/(4x3) = 1.25.

7.2 Association rule using R

The ***apriori()*** function in the ***arules*** library will give these association rules. Let us use the titanic dataset as an example. First, we need to change the dataset d to ***transaction*** object x:

```
> d<-read.csv("titanic.csv")      # read in data
> library(arules)                 # load library arules
> x<-as(d,"transactions")         # change d to transactions
> summary(x)                      # display summary
transactions as itemMatrix in sparse format with
 2201 rows (elements/itemsets/transactions) and
 10 columns (items) and a density of 0.4
most frequent items:
  Age=adult    Sex=male  Survive=no  Class=crew Survive=yes     (Other)
       2092        1731        1490         885         711        1895
```

Now we use the ***apriori()*** function to find the association rules:

```
> rules<-apriori(x)               # save assoc rules to x
Parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target   ext
        0.8    0.1    1 none FALSE            TRUE     0.1      1     10  rules FALSE
> summary(rules)                  # display summary
set of 27 rules
rule length distribution (lhs + rhs):sizes
 1  2  3  4
 1 10 11  5
mining info:
 data ntransactions support confidence
    x          2201     0.1        0.8
```

Note that the default value for support is 0.1 and confidence is 0.8 and there are totally 27 rules with ***support***>=*0.1* and ***confidence***>=*0.8*. We can change these parameters by:

```
> rules<-apriori(x, parameter=list(support=0.2,confidence=0.7))  # min support=0.2 and min conf=0.7
Parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target   ext
        0.7    0.1    1 none FALSE            TRUE     0.2      1     10  rules FALSE
> summary(rules)                  # display summary
set of 30 rules
rule length distribution (lhs + rhs):sizes
 1  2  3  4
 2 12 13  3
mining info:
 data ntransactions support confidence
    x          2201     0.2        0.7
```

To display the first 10 rules with largest ***lift*** value:

```
> inspect(head(sort(rules,by="lift"),n=10))     # display 10 rules with largest lift value
   lhs             rhs          support confidence lift
1  {Class=crew,
    Survive=no} => {Sex=male}     0.304      0.996 1.27
2  {Class=crew,
    Age=adult,
    Survive=no} => {Sex=male}     0.304      0.996 1.27
3  {Class=crew} => {Sex=male}     0.392      0.974 1.24
4  {Class=crew,
    Age=adult}  => {Sex=male}     0.392      0.974 1.24
5  {Age=adult,
    Sex=male}   => {Survive=no}   0.604      0.797 1.18
6  {Age=adult,
    Survive=no} => {Sex=male}     0.604      0.924 1.18
7  {Survive=no} => {Sex=male}     0.620      0.915 1.16
8  {Sex=male}   => {Survive=no}   0.620      0.788 1.16
9  {Class=crew,
    Sex=male}   => {Survive=no}   0.304      0.777 1.15
10 {Class=crew,
    Age=adult,
    Sex=male}   => {Survive=no}   0.304      0.777 1.15
```

Suppose we want to find the rules with Survive=no on the right hand side,

```
> r1<-subset(rules, subset = rhs %in% "Survive=no")  # select rules with Survive=no on rhs
> inspect(r1)                                        # display r1
  lhs             rhs          support confidence lift
1 {Class=3rd}  => {Survive=no}   0.240      0.748 1.10
2 {Class=crew} => {Survive=no}   0.306      0.760 1.12
3 {Sex=male}   => {Survive=no}   0.620      0.788 1.16
4 {Class=3rd,
   Age=adult}  => {Survive=no}   0.216      0.759 1.12
5 {Class=crew,
   Sex=male}   => {Survive=no}   0.304      0.777 1.15
6 {Class=crew,
   Age=adult}  => {Survive=no}   0.306      0.760 1.12
7 {Age=adult,
   Sex=male}   => {Survive=no}   0.604      0.797 1.18
8 {Class=crew,
   Age=adult,
   Sex=male}   => {Survive=no}   0.304      0.777 1.15
```

Or we can find the rules with Class=3rd on the left hand side,

```
> r2<-subset(rules, subset = lhs %in% "Class=3rd")   # select rules with Class=3rd on lhs
> inspect(r2)                                        # display r2
  lhs             rhs          support confidence  lift
1 {Class=3rd}  => {Survive=no}   0.240      0.748 1.105
2 {Class=3rd}  => {Sex=male}     0.232      0.722 0.919
3 {Class=3rd}  => {Age=adult}    0.285      0.888 0.934
4 {Class=3rd,
   Survive=no} => {Age=adult}    0.216      0.902 0.948
5 {Class=3rd,
   Age=adult}  => {Survive=no}   0.216      0.759 1.121
6 {Class=3rd,
   Sex=male}   => {Age=adult}    0.210      0.906 0.953
7 {Class=3rd,
   Age=adult}  => {Sex=male}     0.210      0.737 0.937
```

7.3 Simpson's Paradox

It is important to interpreting the association between variables when the observed relationship may be influenced by the presence of other confounding factors, i.e., hidden variables that are not included in the analysis. Let us illustrate this by the following simple example:

C(HDTV=>EM) = 99/180 = 0.55
C(~HDTV=>EM) = 54/120 = 0.45

Buy HDTV	Buy Exercise Machine		
	Yes	No	Total
Yes	99	81	180
No	54	66	120
Total	153	147	300

This suggests that customers who buy *HDTV* are more likely to buy *Exercise Machine* than those who do not buy *HDTV*. But if we break down the table by a hidden variable: Customer group: {*College Students*, *Working Adult*}, then

For college students:
C(HDTV=>EM) = 1/10 = 0.10
C(~HDTV=>EM) = 4/34 = 0.118

For working adult:
C(HDTV=>EM) = 98/170 = 0.577
C(~HDTV=>EM) = 50/86 = 0.581

Customer Group	Buy HDTV	Buy Exercise Machine		
		Yes	No	Total
College Students	Yes	1	9	10
	No	4	30	34
Working Adult	Yes	98	72	170
	No	50	36	86

The rules suggest that, for each group, customers who do not buy *HDTV* are more likely to buy *Exercise Machine*, which contradict the previous conclusion when data from these two groups are combined. This is known as **Simpson's paradox**. Mathematically speaking, let *a/b* and *p/q* represent the confidence of the rule (*A=>B*) in two different strata; while *c/d* and *r/s* represent the confidence of the rule (*~A=>B*) in the two strata. Simpson's paradox occurs when $a/b < c/d$ and $p/q < r/s$, but for the combined data, $(a+p)/(b+q) > (c+r)/(d+s)$.
The lesson here is that proper stratification is needed to avoid Simpson's paradox. For example, market basket data from a major supermarket chain should be stratified according to store locations, while medical records from various patients should be stratified according to confounding factors such as age and gender.

Other extensions

The co-occurrence matrix can be extended to higher dimension. Therefore we can consider the rule: (*A and B => C*) or (*A and B and C => D*) in a similar manner. The concepts of support, confidence and lift can also be applied similarly. Other extension is **sequence analysis**, where the items or events are given a time sequence. Then we can consider the rules such as (*A => B => C => D*) etc. Another extension is **dissociation analysis** where we discover infrequent items that they never appear together. Then we can consider rules like: (*A=> ~B*) or (*~C => D*) etc.

Reference:

Chapters 6 and 7 of Introduction to Data Mining by Tan, Steinbach and Kumar, Addison Wesley.