Data Mining (2)
Credits :3   theoretical: 2 hours   Practical: 2hours
Lecturer: Dr. Lamia AbedNoor

Syllabus:
Extracting Rules from Groups

Decision Trees

Splitting criteria

Classification

Linear simple regression

Multiple linear regression

Classification and regression trees

Logistic Regression

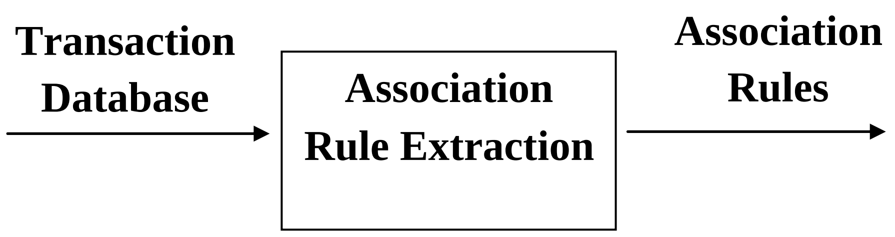
Neural Networks

Time series data mining

Case study (1), Case study (2), Case study (3), Case study (4), Case study (5)

Association Rule Mining

Association rule mining, one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Association rules are widely used in various areas such as telecommunication networks, market and risk management, inventory control etc.



Transaction data

each transaction is a list of items purchased by a customer in a visit. Each transaction has no. that is known Transaction identifier(i.e. TID) and set of items.

Let $I = \{i_1, i_2, \ldots, i_m\}$: a set of *items*.

- Transaction *t* :
  - *t* a set of items(i), and $t \subseteq I$.
- Transaction Database *T*: a set of transactions $T = \{t_1, t_2, \ldots, t_n\}$.
- An itemset is a set of items.

Example:

Transaction data: supermarket data, Market basket transactions:

TID	Produce
1	MILK, BREAD, EGGS
2	BREAD, SUGAR
3	BREAD, CEREAL
4	MILK, BREAD, SUGAR
5	MILK, CEREAL
6	BREAD, CEREAL
7	MILK, CEREAL
8	MILK, BREAD, CEREAL, EGGS
9	MILK, BREAD, CEREAL

Frequent Pattern Analysis?

Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
□ First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of frequent itemsets and association rule mining

Rules:is a popular symbolic representation of knowledge derived from data;

- Natural and easy form of representation → possible inspection by human and their interpretation.
- Standard form of rules

IF *Conditions* THEN *Class*

The other representation can be given in the form:

antecedent → **consequent**

There are various types of rules in data mining such as Decision / classification rules, Association rules and other

Lab.

1- Save data in file using the following command
   Let "po" a variable that want to save in file name is"mydata" , we can use the following command:

```
>save(po,file='mydata')
```

2- Load the data from file
   To retrieve the data that saved in previous file that was mentioned in step 1, and put in variable name 'qq' ,we can use the following command:

```
    >qq=load('mydata')
```

Homework

Construct data frame and put in variable name(bb) and then save it in filename(rr), then loading this data again

Converting transaction database

The transaction data base can be converted in a flat table using binary representation for the attributes; where each item can be represented by binary representation.

Example(1)

TID	Products
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, C

TID	A	B	C	D	E
1	1	1	0	0	
2	0	1	0	1	0
3	0	1	1	0	0
4	1	1	0	1	0
5	1	0	1	0	0
6	0	1	1	0	0
7	1	0	1	0	0
8	1	1	1	0	1
9	1	1	1	0	0

Rule- Strength Measures

There are two important basic measures for association rules, support(s) and confidence(c). The two thresholds are called minimal support and minimal confidence respectively.

Support(s)

Support(s) of an association rule is defined as the percentage/fraction of records that contain $X \cup Y$ to the total number of records in the database(no. of transactions). Suppose the support of an item is 0.1%, it means only 0.1 percent of the transactions contain purchasing of this item.

Support(XY)= count(XY)/count(total transactions)

count(XY): count of transaction contains XY togather

Example(2)

As relating example(1): Let X=A, Y=C Then
S=count (AB)/count(total)
=4/9

Another example: Let X=C, Y=BE, Then
S= count (CBE)/count(total)
=1/9

Confidence(c)

Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain $X \cup Y$ to the total number of records that contain X. Confidence is a measure of strength of the association rules, suppose the confidence of the association rule $X \Rightarrow Y$ is 80%, it means that 80% of the transactions that contain X also contain Y together.

c=count(XY)/count(X)
count(XY): number of transactions contains XY together
count(X): number of transactions contains X

Example(3)
As related example(1), Let X=A, Y=C, then:

Confidence of $A \Rightarrow C$ is computed = count(AB)/count(A)
= 4/6

Association Rules Mining Algorithm

- **Goal:** Find all rules that satisfy the user-specified *minimum support* (minsup) and *minimum confidence* (minconf).
- In general, association rule algorithm first generate the candidate k-itemsets. A set of items (such as the antecedent or the consequent of a rule) is called an itemset. The number of items in an itemset is called the length of an itemset. Itemsets of some length k are referred to as k-itemsets.
- Supports for the candidate k-itemsets are generated by a pass over the database. Itemsets that do not have the minimum support are discarded and the remaining itemsets are called large k-itemsets.
- Select rules with high confidence (using a threshold).

Association Rules Mining Algorithm Types

There are several algorithms that were suggest in this field, one of the popular algorithms that are called "**APRIORI**".

- Find the *frequent itemsets*: the sets of items that have minimum support
  - A subset of a frequent itemset must also be a frequent itemset
    - i.e., if $\{AB\}$ is a frequent itemset, both $\{A\}$ and $\{B\}$ should be a frequent itemset
  - Iteratively find frequent itemsets with cardinality from 1 to *k (k*-itemset*)*
- Use the frequent itemsets to generate association rules.

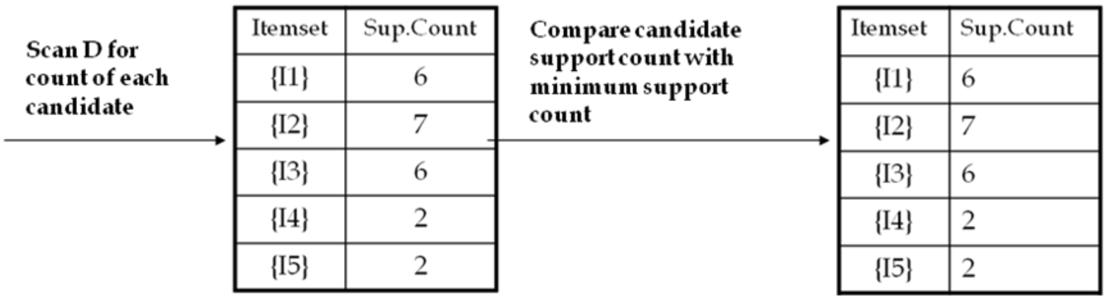
Example

- Consider a database, D , consisting of 9 transactions.
- Suppose min. support count required is 2 (i.e. min_sup = 2/9 = 22 %)
- Let minimum confidence required is 70%.
- We have to first find out the frequent itemset using Apriori algorithm.
- Then, Association rules will be generated using min. support & min. confidence.

TID	**List of Items**
T100	I1, I2, I5
T100	I2, I4
T100	I2, I3
T100	I1, I2, I4
T100	I1, I3
T100	I2, I3
T100	I1, I3
T100	I1, I2 ,I3, I5
T100	I1, I2, I3

First step:

- In the first iteration of the algorithm, each item is a member of the set of candidate.
- The set of frequent 1-itemsets, L_1 , consists of the candidate 1-itemsets satisfying minimum support.



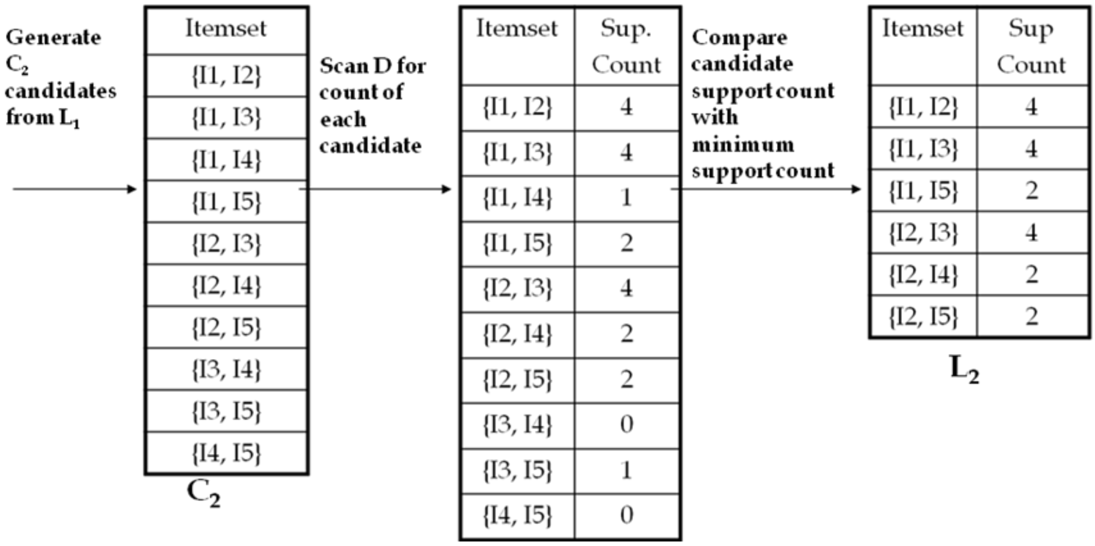
Scan D for count of each candidate

Itemset	Sup.Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count

Itemset	Sup.Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Step 2: Generating 2-itemset Frequent Pattern



Generate C_2 candidates from L_1

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

C_2

Scan D for count of each candidate

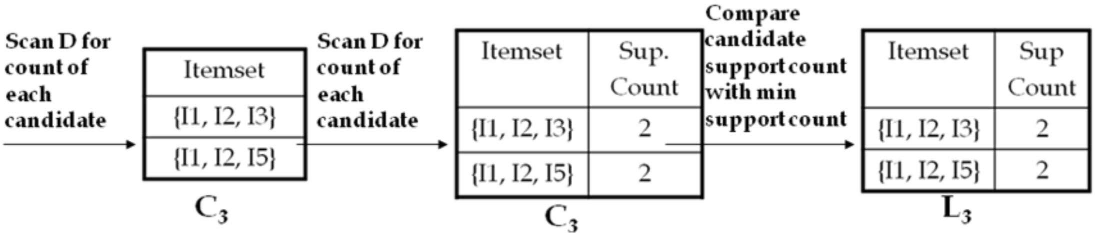
Itemset	Sup. Count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count

Itemset	Sup Count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

L_2

Step 3: Generating 3-itemset Frequent Pattern


Scan D for count of each candidate
Scan D for count of each candidate
Compare candidate support count with min support count
C_3
C_3
L_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

C_3

Itemset	Sup. Count
{I1, I2, I3}	2
{I1, I2, I5}	2

C_3

Itemset	Sup Count
{I1, I2, I3}	2
{I1, I2, I5}	2

L_3

Step 4: Generating 4-itemset Frequent Pattern

- The algorithm uses L_3 *Join* L_3 to generate a candidate set of 4-itemsets, C_4. Although the join results in {{I1, I2, I3, I5}}, this itemset is pruned since its subset {{I2, I3, I5}} is not frequent.

Step 5: Generating Association Rules from Frequent Itemsets

- Procedure:
  - For each frequent itemset ***"l"***, generate all nonempty subsets of ***l.***
  - For every nonempty subset ***s*** of ***l***, output the rule **"s → (l-s)"** if

support_count(l) / support_count(s) >= min_conf where min_conf is minimum confidence threshold.

- Back To Example:

We had L = {{I1}, {I2}, {I3}, {I4}, {I5}, {I1,I2}, {I1,I3}, {I1,I5}, {I2,I3}, {I2,I4}, {I2,I5}, {I1,I2,I3}, {I1,I2,I5}}.

○ Lets take ***l*** = {I1,I2,I5}.

Its all nonempty subsets are {I1,I2}, {I1,I5}, {I2,I5}, {I1}, {I2}, {I5}

- Let minimum confidence threshold is , say 70%.
- The resulting association rules are shown below, each listed with its confidence.
  - ○ R1: I1 ^ I2 → I5
    - Confidence = sc{I1,I2,I5}/sc{I1,I2} = 2/4 = 50%
    - R1 is Rejected.
  - ○ R2: I1 ^ I5 → I2

    - Confidence = sc{I1,I2,I5}/sc{I1,I5} = 2/2 = 100%
    - R2 is Selected.
- ○ R3: I2 ^ I5 → I1
    - Confidence = sc{I1,I2,I5}/sc{I2,I5} = 2/2 = 100%
    - R3 is Selected.

Practical Lab.

1-Association Rules: The package used is (**arules).** The first step is to load this package:

```
>library("arules")
```

2- Loading the data to be performed association rules, for example " **Groceries**" in the following command

```
> data("Groceries")
```

3- To see the summary of this data, we can use the following command

```
> summary(Groceries)
```

4- we mine association rules using the Apriori algorithm implemented in arules.

```
> rules <- apriori(Groceries, parameter=list(support=0.001, confidence=0.5))
```

The results save in variable name(rules)

5- If we need to display the results, we can give the following command

```
>rules
```

Homework

في محاضرة النظري مع الاخذ **Example(1)** طبق الخطوات اعلاه على المثال
بنظر الاعتبار (minsup=0.25) and (minconf=0.2).