Association Rules
Apriori Algorithm

- Machine Learning Overview
- Sales Transaction and Association Rules
- Aprori Algorithm
- Example

Machine Learning

- Common ground of presented methods
  - Statistical Learning Methods (frequency/similarity based)
- Distinction
  - Data are represented in a vector space or symbolically
  - Supervised learning or unsupervised
  - Scalable, works with very large data or not

Extracting Rules from Examples

- Apriori Algorithm
- FP-Growth
  - **Large** quantities of stored data (**symbolic**)
    - Large means often extremely large, and it is important that the algorithm gets scalable
  - **Unsupervised learning**

Cluster Analysis

- K-Means
- EM (Expectation Maximization) COBWEB Clustering
    - Assesment
- KNN (k Nearest Neighbor)

    - Data are represented in a **vector space**
    - **Unsupervised learning**

Uncertain knowledge

- Naive Bayes
- Belief Networks (Bayesian Networks)
    - Main tool is the probability theory, which assigns to each item numerical degree of belief between *0* and *1*

    - Learning from Observation (symbolic data)
    - ***Unsupervised Learning***

Decision Trees

- ID3
- C4.5 cart
  - Learning from Observation (symbolic data)
  - **Unsupervised Learning**

Supervised Classifiers - artificial Neural Networks

- Feed forward Networks
  - with one layer: Perceptron
  - With several layers: Backpropagation Algorithm
- RBF Networks
- Support Vector Machines
  - Data are represented in a **vector space**
  - **Supervised learning**

Prediction

- Linear Regression
- Logistic Regression

Sales Transaction Table

- We would like to perform a basket analysis of the set of products in a single transaction
- Discovering for example, that a customer who buys shoes is likely to buy socks

$$Shoes \Rightarrow Socks$$

SALES

TX#	CUST#	TIMESTAMP	PRODUCT
TX1	C1	d1	Shoes
TX1	C1	d1	Socks
TX1	C1	d1	Tie
TX2	C2	d2	Shoes
TX2	C2	d2	Socks
TX2	C2	d2	Tie
TX2	C2	d2	Belt
TX2	C2	d2	Shirt
TX3	C3	d2	Shoes
TX3	C3	d2	Tie
TX4	C2	d3	Shoes
TX4	C2	d3	Socks
TX4	C2	d3	Belt

Transactional Database

- The set of all sales transactions is called the population
  - We represent the transactions in one record per transaction
  - The transaction are represented by a data tuple

TX1	Shoes,Socks,Tie
TX2	Shoes,Socks,Tie,Belt,Shirt
TX3	Shoes,Tie
TX4	Shoes,Socks,Belt

$$Socks \Rightarrow Tie$$

- Sock is the rule antecedent
- Tie is the rule consequent

Support and Confidence

- Any given association rule has a support level and a confidence level
- **Support** it the percentage of the population which satisfies the rule
- If the percentage of the population in which the antendent is satisfied is *s*, then the **confidence** is that percentage in which the consequent is also satisfied

Transactional Database

$$Socks \Rightarrow Tie$$

- Support is 50% (2/4)
- Confidence is 66.67% (2/3)

TX1	Shoes,***Socks*,Tie**
TX2	Shoes,***Socks*,Tie**,Belt,Shirt
TX3	Shoes,Tie
TX4	Shoes,*Socks*,Belt

Apriori Algorithm

- Mining for associations among items in a large database of sales **transaction** is an important database mining function
- For example, the information that a customer who purchases a keyboard also tends to buy a mouse at the same time is represented in association rule below:

- Keyboard ⇒Mouse
- [support = 6%, confidence = 70%]

Association Rules

- Based on the types of values, the association rules can be classified into two categories: Boolean Association Rules and Quantitative Association Rules
- Boolean Association Rule:

  Keyboard ⇒ Mouse

  [support = 6%, confidence = 70%]
- Quantitative Association Rule:

  (Age = 26 ...30) ⇒(Cars =1, 2)

  [support 3%, confidence = 36%]

Minimum Support threshold

- The support of an association pattern is the percentage of task-relevant data transactions for which the pattern is true

$$A \Rightarrow B$$

$$\text{support}(A \Rightarrow B) = P(A \cup B)$$

$$\text{support}(A \Rightarrow B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{total\_\#\_of\_tuples}$$

Minimum Confidence Threshold

- Confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern

$$A \Rightarrow B$$

$$\text{confidence}(A \Rightarrow B) = P(B \mid A)$$

  - The probability of B given that all we know is A

$$\text{confidence}(A \Rightarrow B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{\#\_tuples\_containing\_A}$$

Itemset

- A set of items is referred to as **itemset**
- An itemset containing *k* items is called **k-itemset**
- An **itemset** can be seen as a conjunction of items (or a presdcate)

Frequent Itemset

- Suppose *min_sup* is the minimum support threshold
- An itemset satisfies minimum support if the occurrence frequency of the itemset is greater or equal to *min_sup*
- If an itemset satisfies minimum support, then it is a frequent itemset

Strong Rules

- Rules that satisfy both a *minimum support* threshold and a *minimum confidence* threshold are called **strong**

Association Rule Mining

- Find all frequent itemsets
- Generate strong association rules from the frequent itemsets

- Apriori algorithm is mining frequent itemsets for Boolean associations rules

Apriori Algorithm

- Lelel-wise search
  - k-itemsets (itensets with k items) are used to explore (k+1)- itemsets from transactional databases for Boolean association rules
    - First, the set of frequent 1-itemsets is found (denoted L_1)
    - L_1 is used to find L_2, the set of frquent 2-itemsets
    - L_2 is used to find L_3, and so on, until no frequent k-itemsets can be found
- Generate strong association rules from the frequent itemsets

Example

$Sup_{min} = 2$

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

- The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent items
- Employs an iterative approach known as level-wise search, where k-items are used to explore k+1 items

Apriori Property

- Apriori property is used to reduce the search space
- Apriori property: All nonempty subset of frequent items must be also frequent
  - Anti-monotone in the sense that if a set cannot pass a test, all its supper sets will fail the same test as well

Apriori Property

- Reducing the search space to avoid finding of each L_k requires one full scan of the database (L_k set of frequent k-itemsets)
- If an itemset *I* does not satisfy the minimum support threshold, *min_sup*, the *I* is not frequent, $P(I) < min\_sup$
- If an item *A* is added to the itemset *I*, then the resulting itemset cannot occur more frequent than *I*, therfor $I \cup A$ is not frequent,
  $P(I \cup A) < \text{min\_sup}$

Scalable Methods for Mining Frequent Patterns

- The downward closure property of frequent patterns
  - Any subset of a frequent itemset must be frequent
  - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
  - i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}
- Scalable mining methods: Three major approaches
  - Apriori (Agrawal & Srikant@VLDB'94)
  - Freq. pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - Vertical data format approach (Charm—Zaki & Hsiao @SDM'02)

Algorithm

1. Scan the (entire) transaction database to get the support *S* of each 1-itemset, compare *S* with *min_sup*, and get a set of frequent 1-itemsets, L_1
2. Use L_{k-1} join L_{k-1} to generate a set of candidate k-itemsets. Use Apriori property to prune the unfreqset k-itemset
3. Scan the transaction database to get the support *S* of each candidate k-itemset in the final set, compare *S* with *min_sup*, and get a set of frequent k-itemsets, L_k
4. Is the candidate set empty, if not goto 2

5 For each frequent itemset *l*, generate all nonempty subsets of *l*

6 For every nonempty subset *s* of *l*, output the rule $s \Rightarrow (I - s)$ if its confidence $C > min\_conf$

- *l={A1,A2,A5}*

$A1 \wedge A2 \Rightarrow A5 \quad A1 \wedge A5 \Rightarrow A2 \quad A2 \wedge A5 \Rightarrow A1$

$A1 \Rightarrow A2 \wedge A5 \quad A2 \Rightarrow A1 \wedge A5 \quad A5 \Rightarrow A1 \wedge A2$

Example

- Five transactions from a supermarket

TID	List of Items
1	Beer,Diaper,Baby Powder,Bread,Umbrella
2	Diaper,Baby Powder
3	Beer,Diaper,Milk
4	Diaper,Beer,Detergent
5	Beer,Milk,Coca-Cola

(diaper=fralda)

Step 1

- Min_sup 40% (2/5)

C1 ➔ L1

Item	Support
Beer	"4/5"
Diaper	"4/5"
Baby Powder	"2/5"
Bread	"1/5"
Umbrella	"1/5"
Milk	"2/5"
Detergent	"1/5"
Coca-Cola	"1/5"

Item	Support
Beer	"4/5"
Diaper	"4/5"
Baby Powder	"2/5"
Milk	"2/5"

Step 2 and Step 3

- C2 ➔ L2

Item	Support
Beer, Diaper	"3/5"
Beer, Baby Powder	"1/5"
Beer, Milk	"2/5"
Diaper,Baby Powder	"2/5"
Diaper,Milk	"1/5"
Baby Powder,Milk	"0"

Item	Support
Beer, Diaper	"3/5"
Beer, Milk	"2/5"
Diaper,Baby Powder	"2/5"

Step 4

- C3 ➔ empty

Item	Support
Beer, Diaper,Baby Powder	"1/5"
Beer, Diaper,Milk	"1/5"
Beer, Milk,Baby Powder	"0"
Diaper,Baby Powder,Milk	"0"

  - Min_sup 40% (2/5)

Step 5

- *min_sup=40%   min_conf=70%*

Item	Support(A,B)	Suport A	Confidence
Beer, Diaper	60%	80%	75%
Beer, Milk	40%	80%	50%
Diaper,Baby Powder	40%	80%	50%
Diaper,Beer	60%	80%	75%
Milk,Beer	40%	40%	100%
Baby Powder, Diaper	40%	40%	100%

Results

$Beer \Rightarrow Diaper$

- support 60%, confidence 70%

$Diaper \Rightarrow Beer$

- support 60%, confidence 70%

$Milk \Rightarrow Beer$

- support 40%, confidence 100%

$Baby\_Powder \Rightarrow Diaper$

- support 40%, confidence 70%

Interpretation

- Some results are belivable, like Baby Powder ➔ Diaper
- Some rules need aditional analysis, like Milk ➔ Beer
- Some rules are unbelivable, like Diaper ➔ Beer

- This example could contain unreal results because of the small data

---

- Machine Learning Overview
- Sales Transaction and Association Rules
- Aprori Algorithm
- Example

- How to make Apriori faster?
- FP-growth