Fast algorithm for mining association rules

Presenter: Linda
Discusser :Massih
Instructor : Rachel Pottinger
University of British Columbia

Overview

- Introduction of data Ming (application of mining)
- Association rule mining
- Apriori algorithm
- Comparison of different algorithms
- Performance comparison

Introduction of data mining

- What is of data mining ?

  Data mining (DM) is the process of automatically searching large volumes of data for patterns such as association rules.

- A Real-World Example:

  a supermarket chain who, through analysis of transactions over a long period of time, found that beer and diapers were often bought together



Introduction of data mining

What are the uses of data mining ?

- *Market segmentation* - Identify common characteristics of customers who buy the same products from your company.
- *Customer churn* - Predict which customers are likely to leave and go to a competitor.
- *Fraud detection* - Identify which transactions are likely to be fraudulent.
- *Direct marketing* - Identify which prospects should be included in a mailing list to obtain the highest response rate.
- *Interactive marketing* - Predict what each individual accessing a Web site is most likely interested in seeing.
- *Market basket analysis* - Understand what products or services are commonly purchased together; e.g., beer and diapers.
- *Trend analysis* - Reveal the difference between a typical customer this month and last.

Introduction of data mining

- **How does data mining works ?**

  Data mining software analyzes ***relationships and patterns*** in stored transaction data based on open-ended user queries.

- **What types of relationships are sought?**
  - ***Classes***: Stored data is used to locate data in predetermined groups.
    example: a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
  - ***Clusters***: Data items are grouped according to logical relationships or consumer preferences.
    example, data can be mined to identify market segments or consumer affinities.

  (cont.)

Relationship Types (cont.)

- ***Sequential patterns***: Data is mined to anticipate behavior patterns and trends.

  example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

- ***Associations:*** Data can be mined to identify associations.

  The beer-diaper example is an example of associative mining.

Association rules mining

Itemset
- A set of itemsets is referenced to as itemset
- An itemset containing k items is called k-itemset

Associate rule
- X➔Y (X and Y are disjoint sets of item set)
- Support of an association rule is the percentage of the relevant data transaction for the which the rule is true.

IF A➔B

$$\text{support}(A \rightarrow B) = \frac{\#\ \text{tuples containig both A and B}}{\text{total\_\#\_of\_tuples}}$$

- Confidence of an association is the measure of certainty associated with each discovered pattern.

IF A➔B

$$\text{confidence}(A \rightarrow B) = \frac{\#\ \text{tuples containig both A and B}}{\text{total\_\#\_of\_tuples}}$$

Association rule mining

- Find all **frequent** itemsets
- Generate **strong association rules** from the frequent itemsets
- The **problem** of association rule mining is: efficiently find all rules with support > Minsup , confidence > Minconf
- Example

Example

- **Itemset acm = {a, c,m}**
- **Support of itemset Sup(acm) = 3**
- **Given min_sup = 3, acm is a frequent patten**
- **Frequent pattern mining find all frequent patterns in a data base**

Transaction database TDB

TID	Items bought
100	F, a, c, d, g, l, m, p
200	A, b, c, f, l, m, o
300	B, f, h, j, o
400	B, c, k, s, p
500	A, f, c, e, l, p, m, n

Problem decomposition

- Discover large itemsets
- Use the largest itemsets to generate the desired rules

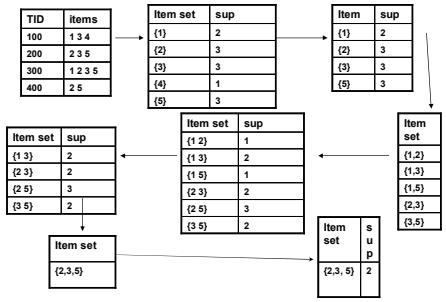
Algorithm Apriori

Find large items

F_1 = {frequent 1-item sets};
k = 2;
while(F_{k-1} is not empty) {
    C_k = Apriori_generate(F_{k-1});
    for all transactions t in T {
        Subset(C_k, t);
    }
    F_k = { c in C_k s.t. c.count >= minimum_support};
}
Answer = union of all sets F_k;

Find candidate pairs, count them ➔ large triplets of items

The Apriori algorithm-- Example



AprioriTid

- Uses the database only once.
- Builds a storage set $C^_k$
  - Members has the form < TID, $\{X_k\}$ >
    - X_k are potentially large k-items in transaction TID.
    - For k=1, $C^_1$ is the database.
- Uses $C^_k$ in pass k+1.

Example Of AprioriTid

Database

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$\overline{C}_1$

TID	Set-of-Itemsets
100	{ {1}, {3}, {4} }
200	{ {2}, {3}, {5} }
300	{ {1}, {2}, {3}, {5} }
400	{ {2}, {5} }

L_1

Itemset	Support
{1}	2
{2}	3
{3}	3
{5}	3

C_2

Itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

$\overline{C}_2$

TID	Set-of-Itemsets
100	{ {1 3} }
200	{ {2 3}, {2 5}, {3 5} }
300	{ {1 2}, {1 3}, {1 5}, {2 3}, {2 5}, {3 5} }
400	{ {2 5} }

L_2

Itemset	Support
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

Itemset
{2 3 5}

$\overline{C}_3$

TID	Set-of-Itemsets
200	{ {2 3 5} }
300	{ {2 3 5} }

L_3

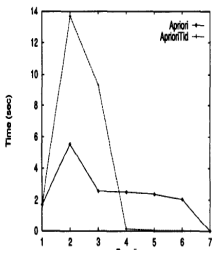
Itemset	Support
{2 3 5}	2

Figure 3: Example

Other algorithms

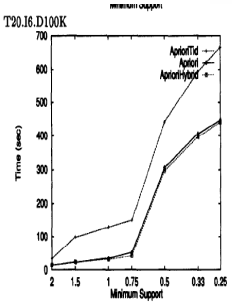
Apriori VS AprioriTid

- In the earlier passes, Apriori does better than AprioriTid.
- AprioriTid beats Apriori in later passes.



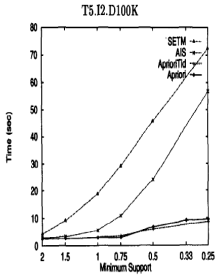
AprioriHybrid

- Uses Apriori in the initial passes and switches to AprioriTid when it expects that the set C_k at the end of the pass will fit in memory



Other algorithms

- AIS
- SETM
- Apriori and AprioriTid algorithms are much better than the AIS algorithm.
- AIS always did considerably better than SETM.



summary

- Association rules are an important tool in analyzing databases.
- We've seen an algorithm which finds all association rules in a database.
- The algorithm has better time results then previous algorithms.
- The algorithm maintains it's performances for large databases.

Discussion

- What are challenges of data mining
- Scalability
- Dimensionality
- Complex and heterogeneous data
- Data quality
- Data ownership and distribution
- Streaming data