Data Warehousing and Data Mining

4th Class

Second course

L3

Edited By

Dr. Khalil I. Ghathwan

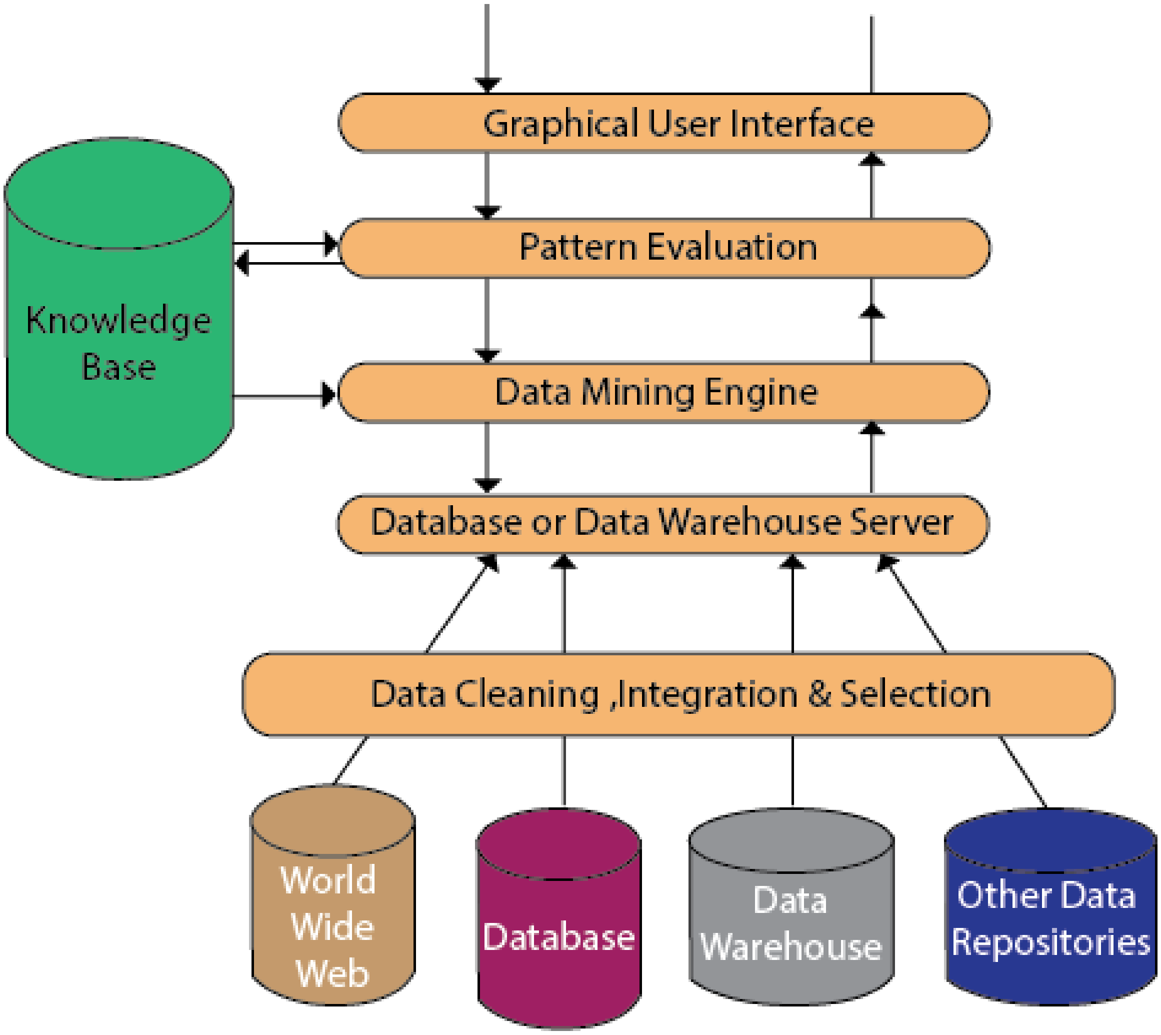
2019-2020

Introduction

Data mining is a significant method where previously unknown and potentially useful information is extracted from the vast amount of data. The data mining process involves several components, and these components constitute a data mining system architecture.

Data Mining Architecture

The significant components of data mining systems are a data source, data mining engine, data warehouse server, the pattern evaluation module, graphical user interface, and knowledge base.



Data Source:

The actual source of data is the Database, data warehouse, World Wide Web (WWW), text files, and other documents. You need a huge amount of historical data for data mining to be successful. Organizations typically store data in databases or data warehouses. Data warehouses may comprise one or more databases, text files spreadsheets, or other repositories of data. Sometimes, even plain text files or spreadsheets may contain information. Another primary source of data is the World Wide Web or the internet.

Different processes:

Before passing the data to the database or data warehouse server, the data must be cleaned, integrated, and selected. As the information comes from various sources and in different formats, it can't be used directly for the data mining procedure because the data may not be complete and accurate. So, the first data requires to be cleaned and unified. More information than needed will be collected from various data sources, and only the data of interest will have to be selected and passed to the server. These procedures are not as easy as we think. Several methods may be performed on the data as part of selection, integration, and cleaning.

Database or Data Warehouse Server:

The database or data warehouse server consists of the original data that is ready to be processed. Hence, the server is cause for retrieving the relevant data that is based on data mining as per user request.

Data Mining Engine:

The data mining engine is a major component of any data mining system. It contains several modules for operating data mining tasks, including association, characterization, classification, clustering, prediction, time-series analysis, etc.

In other words, we can say data mining is the root of our data mining architecture. It comprises instruments and software used to obtain insights and knowledge from data collected from various data sources and stored within the data warehouse.

Pattern Evaluation Module:

The Pattern evaluation module is primarily responsible for the measure of investigation of the pattern by using a threshold value. It collaborates with the data mining engine to focus the search on exciting patterns.

This segment commonly employs stake measures that cooperate with the data mining modules to focus the search towards fascinating patterns. It might utilize a stake threshold to filter out discovered patterns. On the other hand, the pattern evaluation module might be coordinated with the mining module, depending on the implementation of the data mining techniques used. For efficient data mining, it is abnormally suggested to push the evaluation of pattern stake as much as possible into the mining procedure to confine the search to only fascinating patterns.

Graphical User Interface:

The graphical user interface (GUI) module communicates between the data mining system and the user. This module helps the user to easily and efficiently use the system without knowing the complexity of the process. This module cooperates with the data mining system when the user specifies a query or a task and displays the results.

Knowledge Base:

The knowledge base is helpful in the entire process of data mining. It might be helpful to guide the search or evaluate the stake of the result patterns. The knowledge base may even contain user views and data from user experiences that might be helpful in the data mining process. The data mining engine may receive inputs from the knowledge base to make the result more accurate and reliable. The pattern assessment module regularly interacts with the knowledge base to get inputs, and also update it.

Data Mining Applications

1- In Sales/Marketing Data mining enables businesses to understand the hidden patterns inside historical purchasing transaction data, thus helping in planning and launching new marketing campaigns in prompt and cost effective way.

2- In Banking / Finance several data mining techniques, e.g., distributed data mining have been researched, modeled and developed to help credit card fraud detection.

3- In Health Care and Insurance.

4- In Transportation.

5- In Medicine

Advantages of Data Mining

1- Predict future trends, customer purchase habits.

2- Help with decision making.

3- Improve company revenue and lower costs.

4- Market basket analysis.

5- Fraud detection.

Disadvantages of Data Mining

1- User privacy/security.

2- Amount of data is overwhelming.

3- Great cost at implementation stage.

4- Possible misuse of information.

5- Possible in accuracy of data.

A data mining algorithms

A data mining algorithm is a set of heuristics and calculations that creates a data mining model from data. To create a model, the algorithm first analyzes the data you provide, looking for specific types of patterns or trends.

Choosing the best algorithm to use for a specific analytical task can be a challenge. While you can use different algorithms to perform the same business task, each algorithm produces a different result, and some algorithms can produce more than one type of result.

Choosing an Algorithm by Type

Analysis Services includes the following algorithm types:

- **Classification algorithms** predict one or more discrete variables, based on the other attributes in the dataset.
- **Regression algorithms** predict one or more continuous variables, such as profit or loss, based on other attributes in the dataset.
- **Segmentation algorithms** divide data into groups, or clusters, of items that have similar properties.
- **Association algorithms** find correlations between different attributes in a dataset. The most common application of this kind of algorithm is for creating association rules, which can be used in a market basket analysis.
- **Sequence analysis** algorithms summarize frequent sequences or episodes in data, such as a Web path flow.

Overview of Association Rule algorithms

Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based on the concept of association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat.

Basic terminology:

1. Tuples are transactions, attribute-value pairs are items.

2. Association rule: {A,B,C,D,...} => {E,F,G,...}, where A,B,C,D,E,F,G,... are items.

3. Confidence (accuracy) of A => B: P (B|A) = (# of transactions containing both A and B) / (# of transactions containing A).

4. Support (coverage) of A => B: P (A, B) = (# of transactions containing both A and B) / (total # of transactions)

5. We looking for rules that exceed pre-defined support (minimum support) and have high confidence.

$$support = \frac{(X \cup Y).count}{n} \qquad confidence = \frac{(X \cup Y).count}{X.count}$$

t1: Beef, Chicken, Milk

t2: Beef, Cheese

t3: Cheese, Boots

t4: Beef, Chicken, Cheese

t5: Beef, Chicken, Clothes, Cheese, Milk

t6: Chicken, Clothes, Milk

t7: Chicken, Milk, Clothes

Transaction data

Assume: minsup = 30%

minconf = 80%

An example frequent itemset:

{Chicken, Clothes, Milk} [sup = 3/7]

Association rules from the itemset:

Clothes → Milk, Chicken [sup = 3/7, conf = 3/3]

… …

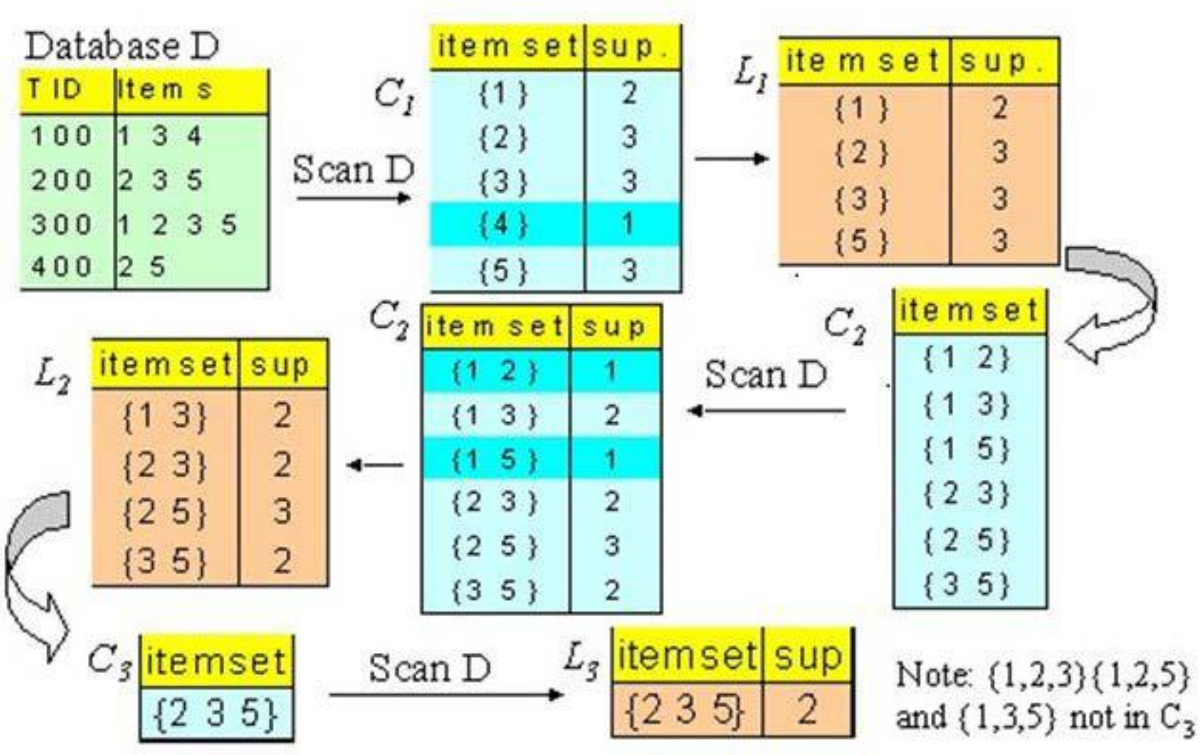
Clothes, Chicken → Milk, [sup = 3/7, conf = 3/3]

Apriori Algorithm

1. Candidate itemsets are generated using only the large itemsets of the previous pass without considering the transactions in the database.
2. The large itemset of the previous pass is joined with itself to generate all itemsets whose size is higher by 1.
3. Each generated itemset that has a subset which is not large is deleted. The remaining itemsets are the candidate ones.

Example: The database of transactions consist of the sets {1,3,4},{2,3,5},{1,2,3,5},{2,5}. We appointed the minimum support level as "1" for this example.

Solution: The first step of Apriori is to count up the frequency (support) of each number (item) separately. Therefore, {4} is not frequent. So, we remove {4} from the first candidate item-set C1. The next step is to generate C2 that is the item-set of all 2-pairs of the frequent items. In the same way, we remove {1, 2}, {1, 5} whose frequencies are "1". And then, we generate the tree of all possible sets. The third candidate item-set is {{2, 3, 5}}. Therefore, because of {2, 3, 5}'s support is 2, the last frequent item-set L3 is {{2, 3, 5}}. As a note, in the last scan, we did not include {1,2,3}, {1,2,5} and {1,3,5} in the C3 because in the previous scan we identified {1,2}, {1,5} items as infrequent, and {1,2}⊂{1,2,3}, {1,2}⊂{1,2,5}, {1,5}⊂{1,2,5}, and {1,5}⊂{1,3,5}. Now we can identify a association rule between the items of the last item-set.



Database D

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

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

L_3

itemset	sup
{2 3 5}	2

Note: {1,2,3}{1,2,5} and {1,3,5} not in C_3