Chapter8

DATA MINING PRIMITIVES, LANGUAGES,

AND SYSTEM ARCHITECTURES

8 DATA MINING PRIMITIVES, LANGUAGES,

AND SYSTEM ARCHITECTURES

8.1 Introduction

A popular misconception about mining is to expect that data mining can autonomously dig out of the valuable knowledge that is embedded in a given large database, without human intervention or guidance. Although it may at first sound appealing to have an autonomous data mining system, in practice such system would uncover an overwhelmingly large set of patterns. The entire set of generated patterns may easily surpass the size given database. To let a data mining system "run loose" in its of patterns, without providing it with any indication regarding the portions of the database that the user wants to prove or the kinds of patterns the user would find interesting is to let loose a data mining "monster". Most patterns discovered would be irrelevant to the analysis task, may be difficult to understand or lack validity novelty, or utility-making them uninteresting. Thus, it is neither realistic nor desirable to generate, store or present all of the patterns that could be discovered from a given database.

A more realistic scenario is to expect that users can communicate with the data mining system using a set of data mining primitive designed in order to facilitate efficient and fruitful knowledge discovery. Such primitives include the specification of the portions of the database or the set of data in which the user is interested, the kinds of knowledge to be mined, background knowledge useful in guiding the discovery process, interestingness measures for pattern evaluation, and how the discovered knowledge should be visualized. These primitives allow the user to interactively communicate with the data mining system during discovery in order to examine the finding from different angles or depths, and direct the mining process.

A data mining query language can be designed to incorporate these primitives, allowing users to flexibly interact, with data mining systems. Having a data mining query language provides a foundation on which user-friendly graphical interfaces can be built.

8.2 Data mining primitives: what defines a data mining task?

Each user will have a data mining task in mind that is some form of data analysis that she would like to have performed. A data-mining task can be specified in the form of a data-mining query, which is input to the data mining system. A data-mining query is defined in terms of the following primitives:

Task-relevant data: this is the database portion to be investigated. For example suppose that you are a manager of ABCompany in charge of sales in the India and Sri Lanka. In particular, you would like to study the buying trends of customers in Sri Lanka. Rather them mining the entire database, or can specify that only the data relating to customer purchasing Sri Lanka need to be retrieved, along with the related customer profile information, you can also specify the attributes of interest to be considered in the mining process. These are referred as relevant attributes. For example, if you are interested only in studying possible relationships between, say, the items purchased and customer annual income and age, then the attribute name of relation items , and income and age of the relation customer , can be specifies as the relevant attributes for mining.

The kinds of knowledge to be mined: this specifies the data mining functions to be performed, such as characterization, discrimination, association, classification, clustering, or revolution analysis. For instance it studying the buying habits of profiles and the items that these customers like to buy.

Background Knowledge: users can specify the background the knowledge or knowledge about domain to be mined. This knowledge is useful for guiding the knowledge discovery process for evaluating the patterns found. There are several kinds of background knowledge. We focus our discussion on a popular from background knowledge known as concept hierarchies. Concept hierarchies are useful in that they allow being data to be minded multiple levels of abstraction. Other examples include user beliefs regarding relationships in the data. These can be used to evaluate the discovered patterns according to their degree of unexpectedness (where unexpected patterns are deemed interesting) or expectedness.(where patterns that confirm a user hypothesis are considered interesting).

Interestingness measures: these functions are used to separate uninteresting patterns from knowledge. They may be used to guide the mining process or , after discovery , to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interesting measures for association rules include support (the percent of task-relevant data tuples for which the rule patterns appears) and confidence (an estimate of the strength of the implication of the rule). Rules whose support and confidence values are below user –specified thresholds are considered uninteresting.

Presentation and visualization of discovered patterns: this refers to the form in which discovered patterns are to be displayed. User can choose from different forms of the knowledge presentation, such as rules, charts, graphs, decision trees, cubes.

8.3Task relevant data:

The first primitive is the specification of the data on which mining is to be performed. Typically a user interested in only a subset of the data base. it is impractical to indiscriminately mined the entire the database particularly since the number of patterns generated could be exponential with respect to the database size. Further more many of the patterns found would be relevant to the interest of the user.

In a relational database, the set of task- relevant data can be collected via a relational query involving operations like selection, projection, join and aggregation . this retrieval of the data can be thought of as a “sub task” of the data mining task the data collection process results in new data relation called the initial data relation. Initial data relation can be ordered or grouped or according to the conditions specified in query. The data may be cleaned or transformed (e.g., aggregated on certain attributes) prior to applying data mining analysis. The initial relation may or not corresponds to a physical relation in the data base. Since virtual relations are called views in the field of data base , the set of task relevant data for mining is called a mixable view.

If the data task is to study association between items frequently at ABCompany by customers in Sri Lanka , the task relevant data can specified by providing the following information.

- The name of data base or data warehouse to be used(e .g,ABCompany-db)
- The name of the tables or data cubes containing the relevant data (e. g.,item, customer , purchases, item sold)
- Conditions for selecting the relevant data (e.g., retrieve data pertaining to purchases made in Sri Lanka for the current year.
- The attributes are dimensions .(e.g., name and price from the item table and income and age from the customer table)

In addition, the user may specify that the data retrieved be grouped by certain attributes, such as “group by” data. Given the information, an SQL query can be used to retrieve the task re Levant data.

In a data warehouse, data are typically stored in a multidirectional database, known as data cube, which can be implemented using a multidimensional array structure, relational structure, or a combination of both. The set of task-relevant data can be specified by the condition-based data filtering, slicing (extraction of data for a given attribute value or “slice”) or dicing (extracting the intersecting of several) of the data cube.

Notice that in a data mining query conditions provided for data selection can be at a level that is conceptually highest then the data in the database or data warehouse. For example a user may specify a selection on items at ABCompany using the concept type “home entertainment” even through individual items in the database may not be stored according to type, but rather, at a lower conceptual, such as “TV “, “CD player” or “VCR”. A conceptual level, composed of the lower level concepts {“TV “, “CD player” or “VCR”}, can be used in the collection of the task-relevant data.

Specification of the relevant attributes or dimensions can be difficult task for users. A user may have only a rough idea of what the interesting attributes for exploration. Furthermore, when specifying the data to be mined the user may overlook additional relevant data having strong semantic links to them. For example, the sales of Halloween, or to particular groups of customers, yet these factors may not be included in the general data analysis request. For such cases, mechanisms can be used that help given a more precise specification of the task-relevant data. These include function to evaluate and rank attributes according to their relevance with respect to the operation specified. In addition, techniques that search for attributes with strong semantic ties can be used to enhance the initial data set specified by the user.

8.4 The kind of knowledge to be mined:

It is important to specify the kind of knowledge to be mined, as this determines the data mining function to be performed. The kinds of knowledge include concept description, (characterization and discrimination), association, classification, prediction, clustering and evolution analysis.

In addition to specifying the kind of knowledge to be mined for a give data mining task, the user can be more specific and provide pattern templates that all discovered patterns must match. These templates or metapatterns (also called metarules or metaqueries), can be used to guide the discovery process.

5.5 Background knowledge: Concept Hierarchies

Background knowledge is information about the domain to be useful in the discovery process./ in this section, we focus our attention on a simple yet powerful from of background knowledge at multiple levels of abstraction.

A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level more general concepts. Concept hierarchies are a useful form of Background knowledge in that they allow raw data to be handled at higher, generalized levels of abstraction. Generalization of the data, or rolling up, is achieved buy replacing primitive-level data (such as continents for location, or numerical values for age) by higher-level data (such as continents, or ranges like "20….30", "40….59", "60+" for age). This allows the user to view the data at more meaningful and explicit abstractions, and makes the discovered patterns to easier to understand. Generalization has an added advantage of compressing the data. Mining on a compressed data set will require fewer input/output operations and be more efficient then mining on a larger, uncompressed data set.

If the resulting data appear over-generalized, concept hierarchies also allow specialization, or drilling down, whereby concept values are replaced by lower-level concepts, by rolling up and drilling down, user can view the data from different perspectives, gaining further into hidden data relation ship. The mappings are typically data-or application specific concept hierarchies can often be automatically discovered or dynamically refined based on statistical of the data distribution.

There may be more than one concept hierarchy for a given attribute or dimension based on different user viewpoints. For example, a regional sales manager of ABCompany who is interested in studying habits of the customers at different location may prefer the concept hierarchy. A marketing manager, however, may prefer to see location organized with respect to linguistic lines facilitate the distribution of commercial ads.

There are four major type of concept hierarchies. The most common types schema hierarchies and set-grouping hierarchies in addition, we also study operation derived hierarchies and rule-based hierarchies.

Schema hierarchies: a schema hierarchy (or more rigorously, a schema defined hierarchy) is a total or partial order among the attribute in the database schema. Schema hierarchies may formally express exiting semantic relationship between attributes. Typically, a schema hierarchy specifies a data warehouse dimension.

Given the schema of a relation for address containing the attributes street, city province_or_state, and country, we can define a location schema hierarchy by the following total order.

Street<city<province_or_state<country

This means that street is at a conceptually lower level than city, which is lower than province_or_state, which is conceptually lower than country. A schema hierarchy provides metadata information, that is, data about the data. Its specification in terms of a total or partial order among attributes is more cosines than an equalant definition that lists all instances of streets, provinces or streets, and countries.

Set-grouping hierarchies: a set-grouping hierarchy organizes values for a given attribute or dimension into grouping of constants or range values. A total or partial order can be defined among the groups. Set-grouping hierarchies can be used to refine or enrich schema defined hierarchies, when the two types of hierarchies are combined. They are typically used for defining small sets of object relationships.

5.6 Interestingness measures: Although specification of the task task-relevant data and of the kind of the knowledge to be mined (e.g., characterization, association, etc.) may subsequently reduce the number of patterns generated, a data mining process may still generate a large number of patterns. Typically, only a small fraction of these patterns will actually be of interest to the given user. Thus, we need to further confine the number of uninteresting patterns returned by the process. This can be achieved by specifying interestingness measures that estimate the simplicity, certainty, utility, and novelty of patterns.

In this section, we study some objective measures of pattern interestingness. Such objective measures are based on the structure of patterns and the statistics underlying them. In general each measure is associated with a threshold that can be controlled by the user. Rules that do not match the threshold are considered unintrestingness and hence are not presented to the user as knowledge.

Simplicity: a factor contributing to the interestingness of a pattern is the patterns over all simplicity for human comprehension. Objective measures of pattern simplicity can be viewed as functions of the pattern structure, defined in terms of the pattern size in bits, or the number of attributes or operators appearing in the pattern. For example the more complex the structure of a rule is, the more difficult it is to interpret, and hence, the less interestingness it is likely to be. Rule length, for instance is a simplicity measure. For rules expressed in conjugative normal from (i.e., as a set of conjunctive predicates), rule length is typically defined as the number if conjuncts in the rule. Association, discrimination, or classification, rules whose lengths exceed a user-defined threshold can be considered uninteresting. For patterns expressed as decision trees, simplicity may be a function of the number of tree leaves of tree leaves or tree nodes.

Certainty: each discovered pattern should have a measure of trustworthiness associated with it.

#_tuples containing A and B

Confidence (A=>B) = ---------------------------------------

#_ tuples containing A

A confidence value of 100% or 1, indicates that the rule is always

Correct on the rule is always correct on the data analyzed. Such rules are called exact.

For classification rules, the above equation can easily adapted to act as a measure of certainty referred to as reliability and accuracy. Classification rules purpose model for distinguishing objects, or tuples, of a target class 9 say, big spenders) from objects of contrasting classes (say, budget Spenders). A low reliability value indicates that the rule in question incorrectly classifies a large number of the contrasting class objects as target class objects. Rule reliability is also known as rule strength, rule quali9ty, certainty factor, and discriminating weight.

Utility: the potential usefulness of a pattern is a factor defining its interestingness. It can be estimated by a utility function, such as support. The support of an association pattern refers to the percentage of task-relevant data tuples (or transactions) for which the pattern is true. For association rules of the form “A ъ B” where A and B are sets of items, it is defined as

#_tuples_contanins_both_A and _B

Support (A=>B) = --

total_#_of_tuples

suppose that the task of task-relevant data of transactions from the computer department of ABCompany. A support of 30% for the association rule above means that 30% of all customers in the department purchased both a computer and software.

Association rules that satisfy both a user-specified minimum confidence threshold and user-specified minimum support threshold are referred to as strong association rules and are considered interesting. Rules with low support likely represent noise, or rare or exceptional cases.

Novelty: Novel patterns are those that contribute new information or increased performance to the given pattern set. For example, a date exception may be considered novel in that differs from that expected based on statistical model or user beliefs. Another strategy for detecting novelty is to remove redundant patterns. If a discovered rule can be implied by another rule it is already in the knowledge base or in the derived rule set, then either it should be reexamined in order to remove the potential redundancy.

Mining with concept hierarchies can result in a large number of redundant rules. For example, suppose that the following association rules were mined from the ABC company database, using the concept hierarchy in Figure 4.3 for location.

Location(X, “Sri Lanka”) = busy (X, ONIDA_TV”) [8%, 70%]

Location(X, “Colombo”) = busy (X, ONIDA_TV”) [2%, 71%]

Suppose that the above rule has 8% support and 70% confidence. We might expect the rule to have a confidence of around 70% as well, since all the tupelos representing data objects for Colombo are also data objects for Sri Lanka rule, is more general than the rule for Colombo, and therefore, we would expect the former rule to occur more frequently than the latter. Consequently, the two rules should not have the same support. Suppose that about one quarter of all sales in Sri Lanka come from Colombo. We would then expect the support of the rule involving Colombo to be one quarter of the support of the rule involving Sri Lanka. In other words, we expect the support of the rule to be 8% x = 2%. If the actual confidence and support of the next rule are as expected, then rule is considered redundant since it does not offer any additional information and is less general than the rule for Sri Lanka.

Data Mining systems should allow users to flexibly and interactively specify, test, and modify interestingness measures and their respective thresholds. There are many other objective measures, apart from the basic ones studied above. Subjective measures exist as well, which consider user beliefs regarding relationships in the data, in addition to objective statistical measures. Interestingness measures are discussed in grater detail throughout the book

8.7 Presentation and Visualization of Discovered Patterns

For data mining to be effective, data mining systems should be able to display the discovered patterns in multiple forms, such as rules, tables, cross tabs (Cross tabulations), pie or bar charts, decision trees, cubes, or other visual representations.

Allowing the visualization of the discovered patterns in various forms can help users with different backgrounds to identify patterns of interest and to interact or guide the system in further discovery. A user should be able to specify the forms of presentation to be used for displaying the discovered patterns.

The use of concept hierarchies plays an important role in aiding the user to visualize the discovered patterns. Mining with concept hierarchies allows the representation of discovered knowledge in high-level concepts, which may be more understandable to user than rules expressed in terms of primitive (i.e., raw) data, such as functional or multi-valued dependency rules, on integrity constraints. Furthermore, data mining system should employ concept hierarchies to implement drill-down and roll-up operations, so those users may inspect discovered patterns at multiple levels of abstraction. In addition, pivoting (or rotating), slicing, and dicing operations aid the user in viewing generalized data and knowledge from different perspectives. A data mining system should provide such interactive operations for any dimension, as well as for individual values of each dimension.

Some representation forms may be better suited than others for particular kinds of knowledge. For example, generalized relations and their corresponding cross tabs or pie/bar charts are good for presenting characteristic descriptions, whereas decision trees are a common choice for classification. Interestingness can be displayed for each discovered pattern, in order to help users identify those patterns representing useful knowledge.

8.8 Review Question

1 Expalin Data mining primitives and what defines a data mining task

2 Explain Concept Hierarchies

3. Explain about Task Relavant data?

4. Discuss about Presentation and Visualization of Discovered Patterns

8.9 References

[1]. Data Mining Techniques, Arun k pujari 1st Edition

[2] .Data warehousung,Data Mining and OLAP, Alex Berson ,smith.j. Stephen

[3].Data Mining Concepts and Techniques ,Jiawei Han and MichelineKamber

[4]Data Mining Introductory and Advanced topics, Margaret H Dunham PEA

[5] The Data Warehouse lifecycle toolkit , Ralph Kimball Wiley student Edition