CHAPTER-23

MINING COMPLEX TYPES OF DATA

23.MINING COMPLEX TYPES OF DATA

23.1 Introduction

Our previous studies on data mining techniques have focused on mining relational data-bases, transactional databases, and data warehouses formed by the transformation and integration of structured data. Vast amount of data in various complex forms (e.g., structured and unstructured, hypertext and multimedia) have been growing explosively owing to the rapid progress of data collection tolls, advanced database system technologies and World –Wide Web (WWW) technologies. Therefore, an increasingly important task in data mining is to mine complex types of data, including complex objects, spatial data, multimedia data, time-series data, text data, and the World Wide Web.

In this chapter, we examine how to further develop the essential data mining techniques (such as characterization, association, classification and clustering), and how to develop new ones to cope with complex types of data and perform fruitful knowledge mining in complex information repositories. Since search into mining such complex databases has been evolving at a hasty pace, our discussion covers only some preliminary issues. We expect that many books dedicated to the mining of complex kinds of data will become available in the future.

23.2 Multidimensional Analysis and Descriptive Mining of Complex Data Objects

A major limitation of many commercial data warehouse and OLAP tools for multidimensional database analysis is their restriction on the allowable data types for dimensions and measures. Most data cube implementations confine dimensions to nonnumeric data and measures to simple aggregated values. To introduce data mining and multidimensional data analysis for complex objects, this section examines how to perform generalization on complex structured objects and construct object cubes for OLAP and mining in object databases.

The storage and access of complex structured data have been studied in object-relational and object-oriented database systems. These systems organize a large set of complex data objects into classes, which are in turn organized into class/subclass hierarchies. Each object in a class is associated with

1) An object-identifier
2) A set of attributes that may contain sophisticated data structures, set- or list-valued data, class composition and hierarchies, multimedia data and so on &
3) A set of methods that specify the computational routines or rules associated with the object class.

To facilitate generalization and induction in object-relational and object-oriented databases, it is important to study how the generalized data can be used for multidimensional data and analysis and data mining.

23.3 Generalization of Structured Data

An important feature of object-relational and object-oriented databases is their capability of storing, accessing and modeling complex structure-valued data, such as set-valued and list-valued data end data with nested structures.

Let's start by having a look at the generalization of set-valued and list-valued attributes.

A set-valued attribute may be homogenous or heterogeneous type. Typically, set-valued data can be generalized by

(1) Generalization of each value in the set into Us corresponding higher-level concepts or
(2) Derivation of general behavior of the set, such as the number of elements in the set, or the weighted average for numerical data. Moreover, applying different generalization operators to explore alternative generalization paths can perform generalization. In this case the result of generalization is a heterogeneous set.

A set-valued attribute may be generalized into a set-valued or single-valued attribute; a single-valued attribute may be generalized into a set valued attribute if the values form a lattice or "hierarchy" or the generalization follows different paths. Further generalizations on such a generalized set-valued attribute should follow the generalization path of each value in the set.

A list-valued or sequence valued attribute can be generalized in a manner similar to that for set valued attributes except that the order of the elements in the sequence should be observed in the generalization. Each value in the list can be generalized into its corresponding higher level concept. Alternatively a list can be generalized according to its general behavior, such as the length of the list, the type of list elements, the value range, the weighted average value for numerical data, or by dropping unimportant elements in the list. A list may be generalized into a list, set, or a single value.

A complex structure-valued attribute may contain sets, tuples, lists, trees, records and so on, and their combinations where one structure may be nested in another at any level. In general, a structure-valued attribute can be generalized in several ways, such as

(1) Generalizing each attribute in the structure while maintaining the shape of the structure,
(2) Flattening the structure and generalizing the flattened structure,
(3) Summarizing the low-level structure by high-level concepts or aggregation &
(4) Returning the type or an overview of the structure.

23.4 Aggregation and Approximation in Spatial and Multimedia Data Generalization

Aggregation and approximation should be considered another important means of generalization, which is especially useful for generalizing attributes with large sets of values, complex structures and spatial or multimedia data.

Let's take spatial data as an example. We would like to generalize detailed geographic points into clustered regions, such as business, residential, industrial, or agricultural areas, according to land usage. Such generalization often requires the merge of a set of geographic areas by spatial operations, such as spatial union or spatial clustering methods. Aggregation and approximation are important techniques for this form of generalization. In a spatial merge, it is necessary to not only merge the regions of similar

types within the same general class but also compute the total areas, average density, or other aggregate functions while ignoring some scattered regions with different types if they are unimportant to the study. Other spatial operators, such as spatial-union, spatial-overlapping, and spatial-intersection, which may require the merging of scattered small regions into large, clustered regions, can also use spatial aggregation and approximation as data generalization operators.

Example:

Suppose that we have different pieces of land for various purposes of agricultural usage, such as the planting of vegetables, grains, and fruits. These pieces can be merged or aggregated into one large piece of agricultural land by a spatial merge. However, such a piece of agricultural land may contain highways, houses, small stores, and so on. If the majority of the land is used for agriculture, the scattered regions for other purposes can be ignored, and the whole region can be claimed as an agricultural area by approximation.

A multimedia database may contain complex texts, graphics, images, video fragments, maps, voice, music, and other forms of audio/video information. Multimedia data are typically stored as sequences of bytes with variable lengths, and segments of data are linked together or indexed in a multidimensional way for easy reference.

Recognition and extraction of the essential features and/or general patterns of such data can perform generalization on multimedia data. There are many ways to extract such information. For an image, aggregation and/or approximation can extract the size, color, shape, texture, orientation, and relative positions and structures of the contained objects or regions in the image. For a segment of music, its melody can be summarized based on its tone, tempo, or the major musical instruments played. For an article, its abstract or general organizational structure (e.g., the table of contents, the subject and index terms that frequently occur in the article, etc.) may serve as its generalization.

In general, it is a challenging task to generalize spatial data and multimedia data in order to extract interesting knowledge implicitly stored in the data. Technologies developed in spatial databases and multimedia databases such as spatial data accessing and analysis techniques and content based image retrieval and multidimensional indexing methods should be integrated with data generalization and data mining techniques to achieve satisfactory results. Techniques for mining such data are further discussed in following sessions.

23.5 Generalization of Object Identifiers and Class/Subclass Hierarchies

At first glance, it may seem impossible to generalize an object identifier. It remains unchanged even after structural reorganization of the data. However, since objects in an object-oriented database are organized into classes, which in turn are organized into class/subclass hierarchies, referring to its associated hierarchy can perform the generalization of an object. Thus, an object identifier can be generalized as follows. First, the object identifier is generalized to the identifier of the lowest subclass to which the object belongs. The identifier of this subclass can then, in turn, be generalized to a higher-level class/subclass identifier by climbing up the class/subclass hierarchy. Similarly, a class or a subclass can be generalized to its corresponding, super class(es) by climbing up its associated class/subclass hierarchy.

Since object-oriented databases are organized into class/subclass hierarchies, some attributes or methods of an object class are not explicitly specified in the class itself but are inherited from higher-level classes of the object. Some object-oriented database systems allow multiple inheritances, where properties can be inherited from more than one super class when the class/subclass "hierarchy" is organized in the shape of a lattice, the inherited properties of an object can be derived by query processing in the object-oriented database. From the data generalization point of view, it is unnecessary to distinguish which data are stored within the class and which are inherited from its super class. As long as the set of relevant data are collected by query processing, the data mining process will treat the inherited data in the same manner as the data stored in the object class, and perform generalization accordingly.

Methods are an important component of object-oriented databases. Many behavioral data of objects can be derived by the application of methods. Since a method is usually defined by a computational procedure/function or by a set of deduction rules, it is impossible to perform generalization on the method itself. However, generalization can be performed on the data derived by application of the method. That is, once the set of task-relevant data is derived by application, of the method, generalization can then be performed on these data.

23.6 Generalization of Class Composition Hierarchies

An attribute of an object may be composed of or described by another object, some of whose attributes maybe in turn composed of or described by other objects, thus forming a class composition, hierarchy. Generalization on a class composition hierarchy can be viewed as generalization on a set of nested structured data (which are possibly infinite, if the nesting is recursive).

In principle, the reference to a composite object may traverse via a long sequence of references along the corresponding class composition hierarchy. However, in most cases, the longer the sequence of references traversed, the weaker the semantic linkage between the original object and the referenced composite object. For example, an attribute vehicles owned of an object class student could refer to another object class car, which may contain an attribute auto-dealer, which may refer to attributes describing the dealer's manager and children. Obviously it is unlikely that any interesting general regularities exist between a student and her car dealer's manager's children. Therefore, generalization on a class of objects should be performed on the descriptive attribute values, and methods of the class, with limited reference to its closely related components via its closely related linkages in the class composition hierarchy. That is, in order to discover interesting knowledge, generalization should be performed on the objects in the class composition hierarchy that are closely related in semantics to the currently focused class(es), but not on those, that have only remote and rather weak semantic linkages.

23.7 Construction and Mining of Object Cubes

In an object database, data generalization and multidimensional analysis are not applied to individual objects but to classes of objects. Since a set of objects in a class may share many attributes and methods, and the generalization of each attribute and method may apply a sequence of generalization operators, the major issue becomes how to make the generalization processes cooperate among different attributes and methods in the class(es) .

For class-based generalization, the attribute-oriented induction method for mining characteristics of relational databases can be extended to mine data characteristics in object databases. Consider that a generalization-based data mining process can be viewed as the application of a sequence of class-based generalization operators on different attributes. Generalization can continue until the resulting class contains a small number of generalized objects that can be summarized as a concise, generalized rule in high-level terms. For efficient implementation, examining each attribute (or dimension), generalizing each attribute to simple-valued data, and constructing a multidimensional data cube, called an object cube can perform the generalization of multidimensional attributes of a complex object class. Once an object cube is constructed, multidimensional analysis and data mining can be performed on it in a manner similar to that for relational data cubes.

Notice that from the application point of view, it is not always desirable to generalize a set of values to single-valued data. Consider the attribute keyword, which may contain a set of keywords describing a book. It does not make much sense to generalize this set of keywords to one single value. In this context, it is difficult to construct an object cube containing the keyword dimension. We will address some progress in this direction in the next section when discussing spatial data cube construction. However, it remains a challenging research issue to develop techniques for handling set-valued data effectively in object cube construction and object-based data mining.

19.8 Generalization-Based Mining of Plan Databases by Divide-and-Conquer

To show how generalization can play an important role in mining complex data-bases, we examine a case of mining significant patterns of successful actions in a plan database using a divide-and-conquer strategy.

A plan **consists** of a variable sequence of actions. A plan database, or **simply** a planbase, is a large collection of plans. Plan mining is the task of mining significant patterns or knowledge from a planbase. Plan mining can be used to discover travel patterns\ of business passengers in an air flight database, or find significant patterns from the sequences of actions in the repair of automobiles. Plan mining is different from sequential pattern mining, where a large number of frequently occurring sequences are mined at a very detailed level. Instead, plan mining is the extraction of important or significant generalized (sequential) patterns from a planbase.

What we would like to find is a small number of general(sequential) patterns that cover a substantial portion of the plans, and then we can divide, our search efforts based on such mined sequences. The key to mining such patterns is to generalize the plans in the planbase to a sufficiently high level, a multidimensional database model, can be used to facilitate such plan generalization. Since low-level information may never share enough commonality to form succinct plans, we should do the following:

(1) Generalize the planbase in different directions using the multidimensional model,

(2) Observe when the generalized plans share common, interesting, sequential patterns having substantial support, and

(3) Derive high-level, concise plans.

Let's examine this planbase. By combining tuples with the **same** plan number, the sequences of actions (shown in terms of airport codes) may appear as follows:

ALB – JFK – ORD – LAX – SAN

SPI – ORD – JFK – SYR

These sequences may look very different. However, they can be generalized in multiple dimensions. When they are generalized based on the airport size dimension, we observe some interesting sequential patterns like S-L-L-S, where L represents a large airport (i.e., a hub), and 5 represents a relatively small regional airport.

The generalization of a large number of air travel plans may lead to some rather general but highly regular patterns. This is often the case if the merge and optional operators are applied to the generalized sequences, where the former merges (and collapses) consecutive identical symbols into one using the transitive closure notation "+" to represent a sequence of actions of the same type whereas the latter uses the notation "I]" to indicate that the object or action inside the square brackets "[‘]"is optional.

By merging and collapsing similar actions, we can derive generalized sequential patterns. In other words, the travel pattern consists of flying first from possibly a small airport, hopping through one to many large airports, and finally reaching a large (or possibly, a small) airport.

After a sequential pattern is found with sufficient support, it can be used to partition the planbase. We can then mine each partition to find common characteristics.

This example demonstrates a divide-and-conquer strategy, which first finds interesting, high-level concise sequences of plans by multidimensional generalization of a planbase, and then partitions the planbase based on mined patterns to discover the corresponding characteristics of sub-planbases. This mining approach can be applied to many other applications.

The plan mining technique can be further developed in several aspects. For instance, a minimum support threshold

similar to that in association rule mining can be used to determine the level of generalization and ensure that a pattern covers a sufficient number of cases. Additional operators in plan mining can be explored, such as less than. Other variations include extracting associations from sub sequences, or mining sequence patterns involving multidimensional attributes, for example, the patterns involving both airports and countries. Such dimension-combined mining also requires the generalization of each dimension to a high level before examination of the combined sequence patterns.

23.9 Review Questions

1 Explain about Multidimensional Analysis and Descriptive Mining of Complex Data Objects

2 Explain about Generalization of Structured Data

3 Explain about Aggregation and Approximation in Spatial and Multimedia Data Generalization

4 Explain about Generalization of Object Identifiers and Class/Subclass Hierarchies

5 Explain about Generalization of Class Composition Hierarchies

6 Explain about Construction and Mining of Object Cubes

7 Explain about Generalization-Based Mining of Plan Databases by Divide-and-Conquer

23.10 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