Towards On-Line Analytical Mining in Large Databases *

Jiawei Han

Intelligent Database Systems Research Laboratory

School of Computing Science, Simon Fraser University, British Columbia, Canada V5A 1S6

URL: http://db.cs.sfu.ca/ (for research group) http://db.cs.sfu.ca/DBMiner (for system)

Abstract

Great efforts have been paid in the Intelligent Database Systems Research Lab for the research and development of efficient data mining methods and construction of on-line analytical data mining systems.

Our work has been focused on the integration of data mining and OLAP technologies and the development of scalable, integrated, and multiple data mining functions. A data mining system, `DBMiner`, has been developed for interactive mining of multiple-level knowledge in large relational databases and data warehouses. The system implements a wide spectrum of data mining functions, including characterization, comparison, association, classification, prediction, and clustering. It also builds up a user-friendly, interactive data mining environment and a set of knowledge visualization tools. In-depth research has been performed on the efficiency and scalability of data mining methods. Moreover, the research has been extended to spatial data mining, multimedia data mining, text mining, and Web mining with several new data mining system prototypes constructed or under construction, including `GeoMiner`, `MultiMediaMiner`, and `WebLogMiner`.

This article summarizes our research and development activities in the last several years and shares our experiences and lessons with the readers.

1 Introduction

The research into data mining in our lab started in early 1989, when we proposed an efficient knowledge discovery method, *attribute-oriented induction* [4]. Since then, we have investigated a set of interesting data mining methods for mining relational data, data warehouse data, spatial data, data formed with complex objects, text data, and multimedia data. These include enhancement of attribute-oriented induction [13, 16], automatic generation and adjustment of concept hierarchies [16], mining multi-level association rules [15], meta-rule guided mining of associations [22], incremental and distributed mining of associations [8, 7], constraint pushing in association mining [10, 27], mining periodicity and similarity in time-series data [11, 30], multi-level classification and prediction [23, 6], spatial data cube construction [21], spatial association rule mining [24], OLAP mining [12], Weblog mining [31], etc.

A data mining system, `DBMiner` [16, 14], has been constructed with our years of research and development. The system integrates data mining with on-line analytical processing (OLAP) and implements a spectrum of data mining functions, including characterization, comparison, association, classification, prediction, and clustering. An important goal of the system is to perform *multiple functional*, *on-line analytical mining* in large databases and data warehouses, where the *on-line analytical mining* implies that data mining is performed in a way similar to on-line analytical processing (OLAP) in multi-dimensional databases, i.e., mining can be performed, *interactively* (i.e., by mouse clicking and with quick response) when possible, in different portions of a multi-dimensional database and at different levels of abstraction.

This paper summarizes our work related to the research and development of on-line analytical mining mechanisms. The remaining of the paper is organized as follows. In Section 2, we present the on-line analytical mining mechanisms designed and implemented in the DBMiner system. In Section 3, we introduce our additional research into analytical mining methods. In Section 4, we present our work on mining complex types of data, including spatial data, complex data objects, text data, multimedia data, and Web data. Finally, we summarize our study and point out some future research directions in Section 5.

2 OLAP + Data Mining → On-Line Analytical Mining

On-line analytical processing (OLAP) is a powerful data analysis method for multi-dimensional analysis of data

*Research was supported in part by a research grant and a CRD grant from the Natural Sciences and Engineering Research Council of Canada, a grant NCE:IRIS/Precarn from the Networks of Centres of Excellence of Canada, and grants from B.C. Advanced Systems Institute, MPR Teltech Ltd., National Research Council of Canada, and Hughes Research Laboratories.

warehouses [5]. Motivated by the popularity of OLAP technology, we develop an On-Line Analytical Mining (OLAM) mechanism for multi-dimensional data mining in large databases and data warehouses. We believe this is a promising direction to pursue based on the following observations.

1. Most data mining tools need to work on integrated, consistent, and cleaned data, which requires costly data cleaning, data transformation, and data integration as preprocessing steps [9]. A data warehouse constructed by such preprocessing serves as a valuable source of cleaned and integrated data for OLAP as well as for data mining.

2. Effective data mining needs exploratory data analysis. A users often likes to traverse flexibly through a database, select any portions of relevant data, analyze data at different granularities, and present knowledge/results in different forms. On-line analytical mining provides facilities for data mining on different subsets of data and at different levels of abstraction, by drilling, pivoting, filtering, dicing and slicing on a data cube and on some intermediate data mining results. This, together with data/knowledge visualization tools, will greatly enhance the power and flexibility of exploratory data mining.

3. It is often difficult for a user to predict what kinds of knowledge to be mined beforehand. By integration of OLAP with multiple data mining functions, on-line analytical mining provides flexibility for users to select desired data mining functions and swap data mining tasks dynamically.

However, data mining functions usually cost more than simple OLAP operations. Efficient implementation and fast response is the major challenge in the realization of on-line analytical mining in large databases or data warehouses. Therefore, our study has been focused on the efficient implementation of the on-line analytical mining mechanism. The methods that we developed include the efficient computation of data cubes by integration of MOLAP and ROLAP techniques, the integration of data cube methods with dimension relevance analysis and data dispersion analysis for concept description, and data cube-based multi-level association, classification, prediction and clustering techniques. These methods will be discussed in detail in the following subsections.

2.1 Architecture for on-line analytical mining

An OLAM engine performs analytical mining in data cubes in a similar manner as an OLAP engine performs on-line analytical processing. Therefore, it is suggested to have an integrated OLAM and OLAP architecture as shown in Figure 1, where the OLAM and OLAP engines both accept users' on-line queries (instructions)

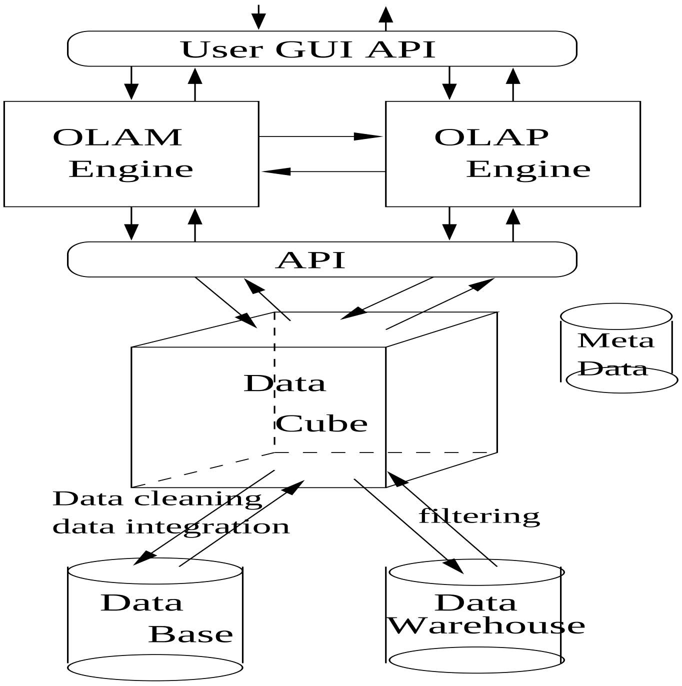

Figure 1: An integrated OLAM and OLAP architecture

and work with the data cube in the analysis. Furthermore, an OLAM engine may perform multiple data mining tasks, such as concept description, association, classification, prediction, clustering, time-series analysis, etc. Therefore, an OLAM engine is more sophisticated than an OLAP engine since it usually consists of multiple mining modules which may interact with each other for effective mining.

Since some requirements in OLAM, such as the construction of numerical dimensions, may not be readily available in the commercial OLAP products, we have chosen to construct our own data cube and build the mining modules on such data cubes. With many OLAP products available on the market, it is important to develop on-line analytical mining mechanisms directly on top of the constructed data cubes and OLAP engines. Based on our analysis, there is no fundamental difference between the data cube required for OLAP and that for OLAM, although OLAM analysis may often involve the analysis of a larger number of dimensions with finer granularities, and thus require more powerful data cube construction and accessing tools than OLAP analyses. Since OLAM engines are constructed either on customized data cubes which often work with relational database systems, or on top of the data cubes provided by the OLAP products, it is suggested to build on-line analytical mining systems on top of the existing OLAP and relational database systems, rather than from the ground up.

2.2 Data cube construction

Data cube technology is essential for efficient on-line analytical mining. There have been many studies on efficient computation and access of multidimensional databases, such as [1, 5, 33].

Our early development of attribute-oriented induction method [13] adopts two generalization techniques: (1) *attribute removal*, which removes attributes which represent low-level data in a hierarchy, and (2) *attribute generalization*, which generalizes attribute values to their corresponding high level ones. Such generalization leads to a new, compressed generalized relation with *count* and/or other aggregate values accumulated. This is similar to the relational OLAP (ROLAP) implementation of the roll-up operation.

For fast response in OLAP and data mining, our later implementation has adopted data cube technology as follows: when data cube contains a small number of dimensions, or when it is generalized to a high level, the cube is structured as compressed sparse array but is still stored in a relational database (to reduce the cost of construction and indexing of different data structures). The cube is precomputed using a chunk-based multiway array aggregation technique similar to [33]. However, when the cube has a large number of dimensions, it becomes very sparse with a huge number of chunks. In this case, a relational structure is adopted to store and compute the data cube, similar to the ROLAP implementation. We believe such a dual data structure technique represents a balance between multidimensional OLAP (MOLAP) and relational OLAP (ROLAP) implementations. It ensures fast response time when handling medium-sized cubes/cuboids and high scalability when handling large databases with high dimensionality.

Notice that even adopting the ROLAP technique, it is still unrealistic to materialize all the possible cuboids for large databases with high dimensionality due to the huge number of cuboids. It is wise to materialize more of the generalized, low dimensionality cuboids besides considering other factors, such as accessing patterns and the sharing among different cuboids.

A 3-D data cube/cuboid can be selected from a high-dimensional data cube and be browsed conveniently using the DBMiner 3-D cube browser as shown in Figure 2, where the size of a cell (displayed as a tiny cube) represents the entry *count* in the corresponding cell, and the brightness of the cell represents another measure of the cell. Pivoting, drilling, and slicing/dicing operations can be performed on the data cube browser with mouse clicking.

2.3 Concept description

Concept/class description plays an important role in descriptive data mining. It consists of two major func-

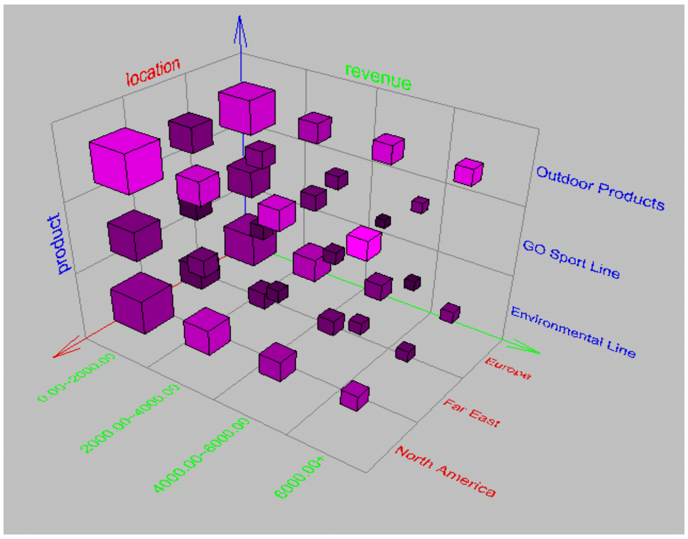

Figure 2: Browsing of a 3-dimensional data cube in DBMiner

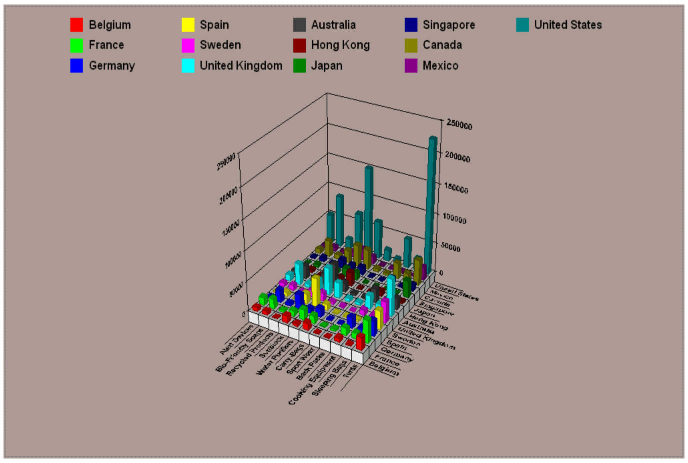

Figure 3: Graphical output of the Characterizer of DBMiner

tions: *data characterization* and *data discrimination* (or *comparison*).

Data characterization summarizes and characterizes a set of task-relevant data by data generalization. Data characterization and its associated OLAP operations, such as drill-down, roll-up (also called drill-up), slice, and dice can be performed on data cubes. Drilling operation facilitates users to examine data characteristics at multiple levels of abstraction.

An output of the DBMiner characterizer is shown in Figure 3.

Data characterization, though can be implemented efficiently using data cube structures, is different from simple OLAP operations in data warehouse in two aspects. First, the data cube approach confines the data types of the dimensions in a data cube to be *simple, nonnumeric data* and the measures to be *simple, aggregated numeric value*, whereas many applications may require the analysis of more complex data types in both dimensions and measures. Second, a simple OLAP operation does not answer some important questions in concept description, such as which dimensions should be included in concept description, and at what level(s) that a generalization process should reach. It is user's responsibility to select appropriate dimensions and decide which level the generalization should reach.

With regard to the first aspect, DBMiner allows numerical attributes to serve as dimensions with a facility of automatic generation of numerical hierarchies based on the value distributions of the numerical attribute in the database. One may choose to generate a numerical hierarchy with naturally segmented, approximately equal-lengthed intervals at each level, or generate a hierarchy with segmentation based on relatively even distribution of certain measure, such as *count* or *sum of sales*, in the database. Also, one may choose to generate hierarchies by applying some sophisticated clustering or segmentation algorithms. Moreover, besides storing *simple, aggregated numeric value* as measures in a data cube, one may store in cube cells pointers to one or a group of (aggregated) objects. For example, the measure in a spatial data cube could be a spatial object pointer, pointing to either a precomputed, merged spatial object or a collection of spatial object identifiers.

The second aspect is handled as follows. A dimension relevance analysis method is used to rank the relevance of the dimensions and only the more relevant dimensions will be included in data characterization. Moreover, instead of performing generalization step by step by repeated mouse clicking in OLAP, characterization generalizes each dimension directly to a desired level controlled by a default or user/expert- specified dimension threshold. Further drill-down or roll-up on the generalized result along a dimension can be performed by the user. The drill-down can be implemented efficiently by saving a *minimally generalized cuboid* or saving a set of cuboids at the levels lower than that of currently generalized cuboid.

Discrimination or *comparison* is to find a set of discriminant features or rules which distinguish the general properties of a target class from that of the contrasting class(es) specified by a user.

Concept discrimination (or comparison) is implemented as follows. First, the set of relevant data in the database is collected by query processing and partitioned respectively into a *target* class and one or a set of *contrasting* class(es). Second, dimension relevance analysis is performed on these classes and only the relevant dimensions are included in the further analysis. Third, generalization is performed on the target class to the level controlled by a user/expert- specified dimension threshold, which results in a *prime target cuboid.* The concepts in the *contrasting* class(es) are generalized to the same level as those in the prime target cuboid, forming the *prime contrasting cuboid(s).* Finally, the resulting classes can be presented in the form of tables, graphics, and rules. Synchronous drilling can be performed on the target and contrasting classes in order to adjust the results to the desired levels.

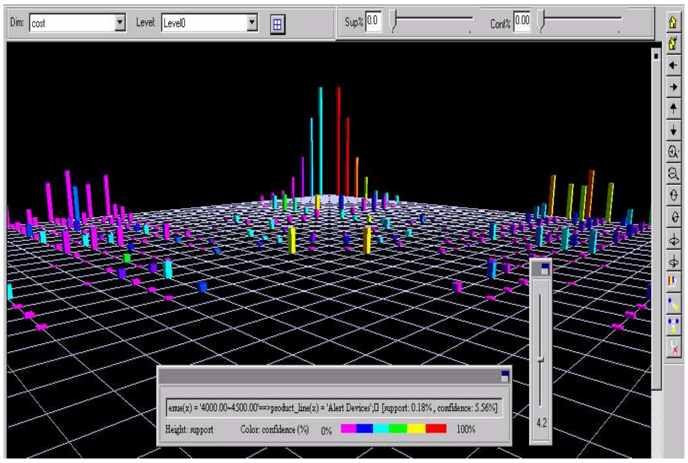

Figure 4: Association plane for visualizing two-dimensional associations in DBMiner

Moreover, analytical descriptive data mining is not confined to characterizing measures of simple aggregate functions, such as *count, average, sum, maximum*, and *minimum.* More comprehensive statistical measures can be included in data characterization and discrimination. For example, one can display the *approximate boxplot* for a combination of two selected dimensions and drill along one dimension to show the behavior of data in regard to both central tendency and dispersion, where the *approximate boxplot* contains the approximations of *the first quartile, median, the third quartile, the whiskers*, and the *potential outliers* or *outlier blocks.* Such *approximate boxplot* can be implemented efficiently in data cubes with numerical dimensions. Similarly, one can construct the *approximate quantile-quantile plot* based on the same principle and compare interesting data distribution properties among different groups of data. Notice that by computing such statistical measures in data cubes, multi-dimensional OLAP analysis, such as interactive drill-down and roll-up along any dimension, can be integrated with statistical analysis to make descriptive data mining an effective and enjoyable process.

2.4 Data cube-based association analysis

Association analysis is an important data mining function. There have been many studies on mining association rules in transaction databases [3, 29, 15]. Data cube offers additional flexibility and efficiency in association rule mining.

Two kinds of associations can be mined in a data cube: *inter-dimension association* and *intra-dimension association*. The former is an association among different dimensions; whereas the latter is an association with regard to one or a set of dimensions (called *reference dimension*) by grouping the remaining set of dimensions into transaction-like sets. In DBMiner, inter-dimension association is called *multi-dimensional association*, whereas intra-dimension association is called *transaction-based association*.

The distinction between the two kinds of associations is illustrated in the following example.

Example 1. Suppose the cube "grading" for a university database contains four dimensions as shown below.

$$grading = \langle student, course, semester, grade \rangle.$$

Inter-dimension association is the association among a set of *distinct* dimensions of a data cube. For example, the association between *course* and *grade*, such as "*the courses in computing science tend to give good grades*", is an inter-dimension association.

Intra-dimension association is the association with regard to one or a set of reference dimensions by grouping the remaining set of dimensions into a transaction-like set. For example, the associations between each student and his/her overall course performance is an intra-dimension association because taking $\langle student \rangle$ as the reference dimension and the *student_id* as the reference level, the remaining set of dimensions, "$\langle$*course, semester, grade*$\rangle$", are grouped into a transaction-like set. A possible association rule could be "*a student taking course A in this semester is likely to take course B in the next* (*semester*)." □

Data cube provides flexibility for mining both kinds of associations. First, it is easy to group data according to one or a set of dimensions using the cube structure. Second, count and other aggregate values may have been computed in data cube which facilitates the association testing and filtering. Moreover, multi-level association can be mined by drilling along any dimension in the data cube with mouse clicking.

Take mining multi-level, inter-dimension association rule as an example. A count cell in a cuboid stores the number of occurrences of the corresponding multi-dimensional data value; whereas the sum of counts of the cells in the whole dimension is also stored in the cuboid. With this structure, it is straightforward to calculate the *support* and *confidence* measures of association rules based on the values in these summary cells. A set of such cuboids, ranging from the minimally generalized one to rather high level ones, facilitate mining of association rules at multiple levels of abstraction.

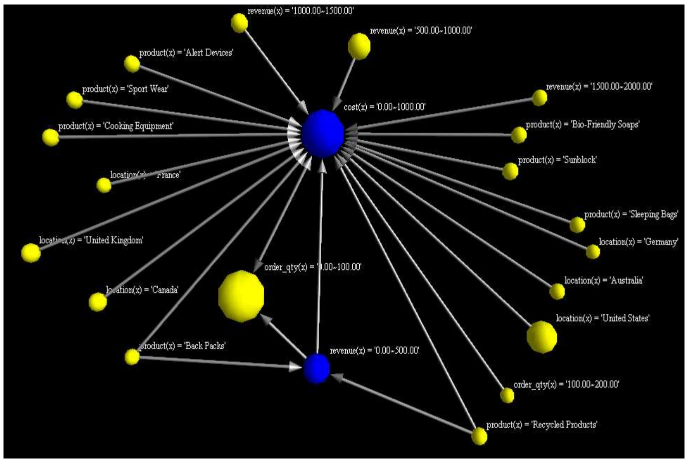

Figure 5: Association rule graph for visualizing multi-dimensional associations in DBMiner

Moreover, it is preferable to push user-specified constraints into the association rule mining process. Such constraints can be specified in a *meta-rule* (or *meta-pattern*) form [22], which confines the search to specific forms of rules. For example, a meta-rule "$P(x, y) \rightarrow Q(x, z, w)$", where P and Q are predicate variables matching different properties in a database, can be used as the *rule template* constraint in the search. With this rule template, one may find a rule like "$major(x, \text{"cs"}) \rightarrow takes(x, \text{"intro\_DBs"}, \text{"3rd\_year"})$", which means if a student x majors in computer_science, he or she is likely to take the course, "*introduction to database systems*", at the third year.

Two-dimensional or two-item association rules can be visualized using association plane as shown in Figure 4. Association rules containing more than two predicates may need the help of an association rule graph, as shown in Figure 5.

2.5 Data cube-based classification

Classification is the process of finding a set of models (or functions) describing data classes or concepts. It is based on the analysis of a set of *training data*, where typically, a *unique* class label for each data object is known. In an ideal world, the model for a given target class would describe all of the objects of that class, and none of the objects from the contrasting classes. In the real world, however, it is recognized that the derivation of such "ideal" models may not be possible.

The data mining view of classification recognizes that although ideal models may be impossible to obtain due to noise or overfitting avoidance, the major factor preventing the creation of ideal models is due to the wide diversity of data in large databases. Given such diversity, it is more probable to assume that a given object may belong to more than one class, particularly when the data have been generalized to high levels of

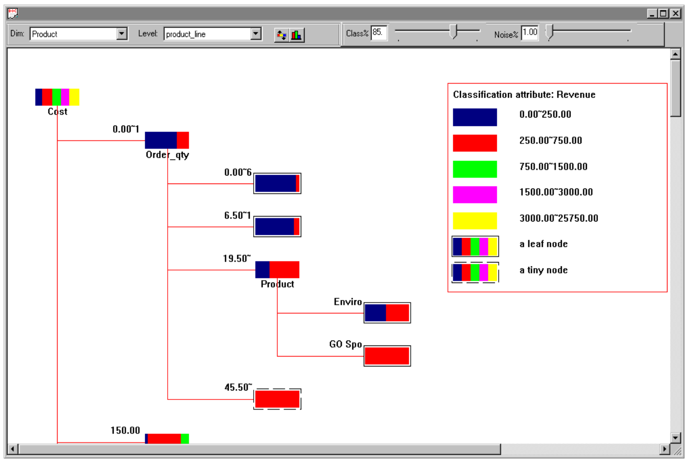

Figure 6: Graphical output of the Classifier of DBMiner

abstraction. Therefore, each model will end up covering most of the objects of the class it represents, while *maximally distinguishing* the properties of the class from that of the other classes. Furthermore, consider the use of such models to classify a given object whose class label is unknown. Such classification models will return a *class probability distribution*, rather than a unique class prediction. This distribution allows the user to view, for each class, the predicted probability that the object can belong to that class.

There have been many classification methods studied, including decision-tree methods, such as ID-3 and C4.5 [28], statistical methods, neural networks, rough sets, as well as some recently proposed database-oriented classification methods [25].

Our classification method consists of four steps: (1) collection of the relevant set of data and partitioning of the data into training and test data, (2) analysis of the relevance of the attributes, (3) construction of classification (decision) tree, and (4) test of the effectiveness of the classification using the test data set.

Attribute relevance analysis is performed based on the analysis of an uncertainty measurement, a measurement which determines how much an attribute is in relevance to the class attribute. Other measurements, such as entropy-based information gain [28] and Gini index [25], can be used for relevance analysis as well. Several top-most relevant attributes are retained for classification analysis; whereas the weakly or irrelevant attributes are not considered in the subsequent classification process.

In the classification process, our classifier adopts a generalization-based decision-tree induction method which integrates data cube technology with a decision-tree induction technique, by first performing minimal generalization on the set of training data, and then performing decision tree induction on the generalized data.

Since a generalized cell comes from the generalization of a number of original cells, the *count* information is associated with each generalized cell and plays an important role in classification. To handle noise and exceptional data and facilitate statistical analysis, two thresholds, *classification threshold* and *exception threshold*, are introduced. The former is used for justification whether it is needed to continue classification on a node if a significant set of the examples of the node belongs to a single class; whereas the latter is used to terminate further classification on a node if the node contains only a negligible number of examples.

With the availability of data cube, drilling can be performed on any dimension as well as on the class attribute, and classification will be performed at the new, corresponding abstraction space.

An output of the classification module of DBMiner is shown in Figure 6.

2.6 Data cube-based prediction

A predictor predicts data values or value distributions on the attributes of interest based on similar groups of data in the database. For example, one may predict the amount of research grants that an applicant may receive based on the data about the similar groups of researchers.

The power of data prediction should be confined to the ranges of numerical data or the nominal data generalizable to only a small number of categories. It is unlikely to give reasonable prediction on one's name or social insurance number based on other persons' data.

For successful prediction, the factors (or attributes) which strongly influence the values of the attributes of interest should be identified first. This can be done by the analysis of data relevance or correlations by statistical methods, decision-tree classification techniques, or be simply based on expert judgement. Similar to the method used in our classifier, we use the uncertainty measurement in the analysis of attribute relevance. This process ranks the relevance of all the attributes selected and only the highly ranked attributes will be used in the prediction process.

After the selection of highly relevant attributes, a generalized linear model has been constructed which can be used to predict the value or value distribution of the predicted attribute.

When a query probe is submitted, the corresponding value distribution of the predicted attribute can be plotted based on the curves or pie charts generated above. The values in the set of highly relevant predictive attributes can be used for trustable prediction.

The prediction output has two forms of presentation: curve graph and pie chart depending whether the predictive attribute is a numeric attribute or a categorical attribute. When the predictive attribute is a numeric one, the output is a set of curves, each indicating the

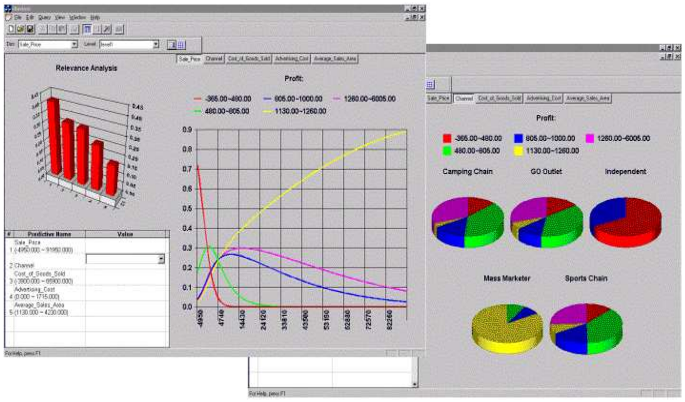

Figure 7: Graphical output of the Predictor of DBMiner: numeric predictive attribute (left) and categorical predictive attribute (right)

trend of likely changes of the value distribution of the predicted attribute, as shown in the left half of Figure 7. When the predictive attribute is a categorical one, the output is a set of pie charts, each indicating the distributions of the value ranges of the predicted attribute, as shown in the right half of Figure 7.

On-line analytical mining for prediction can be performed easily using the data cube structure. Drilling can be performed along any predicted or predicting attribute (or dimension), and prediction will then be performed in the corresponding abstraction space.

2.7 Data cube-based clustering analysis

Data clustering is a process of partitioning a set of data into a set of classes, called *clusters*, with the objects in each cluster sharing some interesting common properties. A good clustering method should produce high quality clusters to ensure that the intra-cluster similarity is high and inter-cluster similarity is low.

Clustering analysis has many interesting applications. For example, it can be used to help marketers discover distinct groups in their customer bases and develop targeted marketing programs.

Data clustering has been studied in statistics, machine learning, image processing, and data mining with different methods and emphases [26, 32]. A data cube-based clustering analyzer must effectively deal with large amount and high dimensionality of data and find interesting clusters. Moreover, most of the existing data clustering methods can only handle numeric data or cannot produce good quality results in the case where categorical domains are present. A data cube-based algorithm should handle both numerical and categorical data and make good use of concept hierarchy information as well.

Based on these considerations, a data cube-based, multi-level clustering module is being developed in DBMiner. The general idea of the method is to grow the clusters from low dimensions to higher ones, and from high abstraction space to low abstraction ones. Also, multi-level hierarchy information is associated with the categorical data in the form of *weight* to quantize the differences among the data in different relative positions in the hierarchy.

On-line analytical mining for cluster analysis allows drilling along different dimensions. By doing so, clustering analysis will then be performed on the corresponding new abstraction space.

3 More on Analytical Mining Methods

Besides the techniques presented above which have been or are being implemented in the DBMiner system, more studies have been performed on efficient and effective on-line analytical mining. These include the design of a data mining language, incremental and distributed mining of association rules, constrained association mining, mining periodic patterns, wavelet technique for similarity-based time-series analysis, intelligent query answering with data mining techniques, and a multi-layer database model.

3.1 Design of a data mining query language

To support ad-hoc and interactive data mining, it is essential to design a good data mining query language. Such a language can be used to serve as the underlying core of different graphical user interfaces of a variety of commercial data mining systems and facilitate the standardization and wide adoption of the technology.

Based on our study of data mining systems, a data mining language, DMQL [18], has been proposed and partially implemented in the DBMiner system. The language adopts an SQL-like syntax and provides primitives for specification of different data mining tasks. Especially, it provides a variety of primitives for the specification of rule templates and query constraints for query-based data mining.

3.2 Incremental and distributed mining of association rules

With huge amounts of data in a database, it is highly preferable to update data mining results incrementally rather than mining from scratch on database updates.

It is straightforward to work out incremental data mining algorithms for concept description since a data cube can be updated incrementally on database updates [13]. However, it is nontrivial to update association rules incrementally.

Let's examine the issue under the Apriori framework [3]. Upon insertion of ΔDB (a set of database tuples) into a DB, some previously large (i.e., frequent) k-itemsets (L_k for any k) may become small (i.e., infrequent), whereas some previously small ones may become large. With the same support threshold, an itemset is *large* in $DB \cup \Delta DB$ if it is large in both DB and ΔDB, and it is *small* in $DB \cup \Delta DB$ if it is small in both. Thus we only need to examine whether an itemset which is large in DB but small in ΔDB may pass the support threshold test (which is an easy task), or whether an itemset which is large in ΔDB but small in DB may pass the test (which needs a scan of DB). Some additional techniques can be used to further speed up the processing. A detailed discussion is presented in [8].

Similar heuristics and some additional techniques can be applied to *parallel and/or distributed mining* of association rules so that locally large itemsets can be mined in each partition, and with minimal message passing, one may compute the globally large itemsets without redistributing data to different sites. A detailed study is presented in [7].

3.3 Constrained association rule mining

It is highly desirable to promote ad-doc query-based data mining since users may like to examine different portions of data with different constraints. Constrained association rule mining is to support constraint-based, human-centered exploratory mining of associations and investigate how user-specified constraints can be pushed deeply into the association mining process to reduce the search space.

Some foundational issues of constrained association mining are examined systematically in [27] by putting a rich set of constraint constructs, including domain, class, and SQL-style aggregate constraints into the same framework and discovering two properties of constraints that are critical to pruning: *anti-monotonicity* and *succinctness*.

For example, suppose a query requires S_l, the set of items at the left hand side of the rule, satisfy the following three constraints: (1) $S_l \subseteq \{milk, bread, cheese\}$, (2) $max(S_l.price) < 1000$, and (3) $avg(S_l.price) < 500$. The first two constraints are *anti-monotonic* in the sense that if a set S does not satisfy the constraint $\mathcal{C}$, adding more items into S will not make it satisfy $\mathcal{C}$. Thus both constraints can be pushed deeply into each iteration under the Apriori framework. However, the third constraint "$avg(S_l.price) < 500$" cannot be pushed in because it is not *anti-monotonic* in the sense that if a set S does not satisfy the constraint, adding more items to S may make it satisfy the constraint.

A systematic study on what kinds of constraints are *anti-monotonic* and/or *succinct* is performed in [27] which also presents an efficient algorithm for constraint pushing.

3.4 Mining periodicity patterns

Many patterns are periodic or approximately periodic in nature, e.g., seasons change periodically by year, temperatures change periodically by day, etc. However, in many cases, some particular points or segments in a sequence could be (approximately) periodic although the whole sequence has no periodicity behavior. For example, Tom watches CBS news at 8:00–8:30am almost everyday but his TV watching habit is "irregular" at other hours.

Can we mine the periodicity of such patterns in large databases? Note that the traditional periodicity detection methods, such as Fast Fourier Transformation, find the periodicity of the whole sequence but not the periodicity of particular point/segment in the sequence as illustrated in the above example.

We examine the problem in two cases: mining periodicity (1) with a *given period* and (2) with an *arbitrary period*, and propose an OLAP-based technique to mine such periodicity [11]. For user-specified given period, such as per day, per week, per quarter, etc., the potential activity patterns can be aggregated with respect to the given period along the time dimension in a data cube. Such aggregation will indicate the patterns which are periodic with respect to the given period, and such patterns may grow from small segments to larger ones by merging its neighborhood cells. For mining periodic patterns with arbitrary periods, similar OLAP-based methods can apply with the augmentation of some cycle merging properties. Notice also with OLAP-based cube manipulation, one can drill-down, roll-up, slice and dice the time-related cuboids to find such periodicity patterns on-the-fly. A detailed study is presented in [11].

3.5 Wavelet technique for similarity-based time-series analysis

Similarity-based time-series analysis is to find similar time-related patterns (trends, segments, etc.) in a large time-series database, such as stock market database.

Traditional trend analysis techniques, such as Fourier transformation, are adopted in most previous analyses of similarity-based time-series [2]. With the popular adoption of wavelet transformation and analysis methods, we examine the wavelet transformation-based similarity mining methods for discovery of trends and/or similar curves or curve segments [30]. Template segments can be specified by users based on the given curve segments or using template primitives. Then wavelet transformation techniques can be used for curve smoothing and approximation, scaling, and translation, and then fragment-based pattern matching analysis. Our study shows that the method is efficient and effective at mining large time-series databases.

3.6 Intelligent query answering with data mining techniques

With data mining techniques available, database queries can be answered intelligently using concept hierarchies, data mining results, or on-line data mining techniques [17]. For example, instead of presenting bulky answers, one can present a summary of answers and allow users to manipulate such a summary by drilling or dicing. One can present related answers or rules in the form of associations or correlations based on association mining results. Moreover, one may add useful dimensions to extend the width of the result table, or add additional (neighborhood) tuples as an extension of the height of the table. OLAP techniques and data mining methods provide useful tools for efficient and effective intelligent query answering.

3.7 A multi-layer database model for heterogeneous databases

A major challenge for cooperating multiple databases is the semantic heterogeneity among different databases. This is difficult to handle due to the autonomy and semantic heterogeneity of the component databases. Methods for schema analysis, transformation, integration, and mediation have been investigated in the database community in order to produce tools to handle this problem. However, schema level analysis may sometimes be too general to solve the problem. Data level analysis, i.e., the analysis of database contents, should be taken into serious consideration.

With generalization-based data mining, a multi-layer database model can be constructed by utilizing some common data access API and generalizing database contents from primitive level to multiple, higher levels [19]. A set of mutually related, generalized databases form a multi-layer database. Such a multi-layer database not only provides a useful architecture for intelligent query answering but also helps information exchange and interoperability among heterogeneous databases. This is because the low-level heterogeneous data can be transformed into high-level, relatively homogeneous information which can be used for effective communication and query/information transformation among multiple databases.

Methods for construction and maintenance of multiple-layer databases and for information exchange among heterogeneous databases are studied in [19].

4 Towards On-Line Analytical Mining of Complex Types of Data

It is challenging to extend the on-line analytical mining method to complex types of data, such as complex data objects, spatial data, text and multimedia data and Web-based data. Here we report our preliminary studies towards this direction.

4.1 Data mining in object-oriented and object-relational databases

Object-oriented and object-relational databases introduce a set of advanced concepts in database systems, including object identity, complex structured objects, methods, class/subclass hierarchies, etc.

A generalization-based data mining method is proposed which generalizes complex objects, constructs a multi-dimensional object cube, and performs analytical mining in such an object cube [20]. Notice that objects with complex structures can be generalized to high-level data with relatively simple structures. For example, an object identifier can be generalized to the class identifier of the lowest class where the object resides. An object with a sophisticated structure can be generalized into several dimensions of data which reflect the structure, the generalized value, or other features of the object. A method associated with an object can be generalized to the data returned by the application of the method to the object. Generalization of a class of objects should be performed in a way similar to the generalization of a data relation, and it results in a generalized class and forms the basis for an object data cube.

4.2 Spatial OLAP and spatial data mining

A spatial database stores both *spatial data* which represents points, lines, and regions, and *nonspatial data* which represents other properties of spatial objects and their nonspatial relationships.

A spatial data cube [21] consists of both spatial and nonspatial dimensions and/or measures and can be modeled by the star or snowflake schema, resembling its relational counterpart [5]. Since a spatial measure may represent a group of aggregated spatial objects, whereas multi-dimensional spatial aggregation may produce a great number of such aggregated spatial objects, it is impossible to precompute and store all of such spatial aggregations. Therefore, selective materialization of aggregated spatial objects is a reasonable tradeoff between storage space and on-line computation time. A method for selective materialization of spatial objects in spatial data cube computation is studied in [5].

Spatial data mining can be performed in a spatial data cube as well as in a spatial database. A spatial OLAP and spatial data mining system prototype, GeoMiner, is constructed based on our studies. Because of the high cost of spatial computation, a *multi-tier computation technique* is adopted in spatial data mining [24]. For example, at mining spatial association rules, one can first apply rough spatial computation, such as minimal bounding rectangle method, to filter out most of the sets of spatial objects which should be excluded from further consideration (e.g., not spatially close enough), and then apply relatively costly, refined spatial computation only to the set of promising candidates.

4.3 Text and multimedia data mining

Text analysis methods and content-based image retrieval techniques play an important role at mining text and multimedia data, respectively. Our method for on-line analytical mining of text and multimedia data follows the same philosophy as we did for others, by first building text/multimedia data cubes and then extending the cube-based relational and/or spatial mining techniques towards mining text and multimedia data.

Currently, our text data mining is performed on a library database and an e-mail database, and our multimedia data mining is experimented on a database of on-line pictures, most of which were fetched from the Internet. A MultiMediaMiner system prototype is being constructed by integration of DBMiner with a content-based image retrieval system, C-Bird, developed in the Multimedia Lab of our School.

4.4 Weblog mining

Because of the complexity of the unstructured and semi-structured data on the Internet, little progress has been made towards our previously planned WebMiner for the construction of data warehouses for the Internet and mining of such data warehouses.

Instead, a WebLogMiner is being constructed to mine the Web access patterns stored in the Web log records [31]. Our approach is similar to our previous work on data cube construction and analytical mining by preprocessing and cleaning Web log records, building multiple dimensions based on the Web access information, such as page start time, duration, user, server, URL, next_page, page_type, and so on, constructing a WebLog cube, and performing time-related, multi-dimensional data analysis and data mining.

5 Conclusions

On-line analytical mining, which integrates on-line analytical processing and data mining, is a promising direction for mining large databases and data warehouses. We summarized our work in this direction, including the research into efficient mining methods, the development of the DBMiner system, and the preliminary study of mining complex types of data.

Our major efforts have been dedicated to the high performance and fast response of on-line analytical mining. Nevertheless, we feel that efficiency is still a major challenge to satisfactory exploratory data mining. However, since data cube technology generalizes a huge amount of data to a controllable size at a high level of abstraction, high performance can be achieved with a trade-off between the response time and mining granularity. With fast increase of computing power (including parallel and distributed processing) and rapid progress on the research into data mining performance issues, one can expect on-line analytical mining will achieve increasingly faster performance and finer mining granularity.

With multiple data mining functions available, one may wonder how to determine which data mining function is the most appropriate one for a particular application. To select an appropriate data mining function, one needs to be familiar with the application problem, data characteristics, and the roles of data mining functions. Sometimes one needs to perform interactive exploratory analysis to observe which function discloses the most interesting features in the database. Therefore, the building of exploratory analysis tools and the construction of an application-oriented semantic layer are two important steps. On-line analytical mining provides an exploratory analysis tool, however, further study should be performed on the *automatic selection* of data mining functions for particular applications.

For effective interpretation of data mining results and interaction with data mining process, visual data mining seems to be quite important. On-line analytical mining should be integrated with visual data mining for effective exploratory mining.

Moreover, our research and development of on-line analytical mining on complex types of data, including spatial, text, multi-media, and Web data, have just started. More work will be reported in this direction in the future.

References

[1] S. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. In *Proc. 1996 Int. Conf. Very Large Data Bases*, pages 506–521, Bombay, India, Sept. 1996.

[2] R. Agrawal, K.-I. Lin, H.S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proc. 21st Int. Conf. Very Large Data Bases*, pages 490–501, Zurich, Switzerland, Sept. 1995.

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 1994 Int. Conf. Very Large Data Bases*, pages 487–499, Santiago, Chile, September 1994.

[4] Y. Cai, N. Cercone, and J. Han. Attribute-oriented induction in relational databases. In G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*, pages 213–228. AAAI/MIT Press, 1991.

[5] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26:65–74, 1997.

[6] S. Cheng. Statistical approaches to predictive modeling in large databases. In *M.Sc. Thesis, Simon Fraser University*, Burnaby, B.C., Canada, January 1998.

[7] D.W. Cheung, J. Han, V. Ng, A. Fu, and Y. Fu. A fast distributed algorithm for mining association rules. In

Proc. 1996 Int. Conf. Parallel and Distributed Information Systems, pages 31–44, Miami Beach, Florida, Dec. 1996.

[8] D.W. Cheung, J. Han, V. Ng, and C.Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In *Proc. 1996 Int'l Conf. Data Engineering*, pages 106–114, New Orleans, Louisiana, Feb. 1996.

[9] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[10] Y. Fu and J. Han. Meta-rule-guided mining of association rules in relational databases. In *Proc. 1st Int'l Workshop Integration of Knowledge Discovery with Deductive and Object-Oriented Databases (KDOOD'95)*, pages 39–46, Singapore, Dec. 1995.

[11] W. Gong. Periodic pattern search in time-related data sets. In *M.Sc. Thesis, Simon Fraser University*, Burnaby, B.C., Canada, November 1997.

[12] J. Han. OLAP mining: An integration of OLAP with data mining. In *Proc. 1997 IFIP Conf. Data Semantics (DS-7)*, pages 1–11, Leysin, Switzerland, Oct. 1997.

[13] J. Han, Y. Cai, and N. Cercone. Data-driven discovery of quantitative rules in relational databases. *IEEE Trans. Knowledge and Data Engineering*, 5:29–40, 1993.

[14] J. Han, J. Chiang, S. Chee, J. Chen, Q. Chen, S. Cheng, W. Gong, M. Kamber, G. Liu, K. Koperski, Y. Lu, N. Stefanovic, L. Winstone, B. Xia, O. R. Zaïane, S. Zhang, and H. Zhu. DBMiner: A system for data mining in relational databases and data warehouses. In *Proc. CASCON'97: Meeting of Minds*, pages 249–260, Toronto, Canada, November 1997.

[15] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proc. 1995 Int. Conf. Very Large Data Bases*, pages 420–431, Zurich, Switzerland, Sept. 1995.

[16] J. Han and Y. Fu. Exploration of the power of attribute-oriented induction in data mining. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 399–421. AAAI/MIT Press, 1996.

[17] J. Han, Y. Huang, N. Cercone, and Y. Fu. Intelligent query answering by knowledge discovery techniques. *IEEE Trans. Knowledge and Data Engineering*, 8:373–390, 1996.

[18] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. in preparation, 1998.

[19] J. Han, R. T. Ng, Y. Fu, and S. Dao. Dealing with semantic heterogeneity by generalization-based data mining techniques. In *M. P. Papazoglou and G. Schlageter (eds.), Cooperative Information Systems: Current Trends & Directions*, pages 207–231, Academic Press, 1998.

[20] J. Han, S. Nishio, H. Kawano, and W. Wang. Generalization-based data mining in object-oriented databases using an object-cube model. In *Data and Knowledge Engineering*, to appear, 1998.

[21] J. Han, N. Stefanovic, and K. Koperski. Selective materialization: An efficient method for spatial data cube construction. In *Proc. 1998 Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD'98)*, Melbourne, Australia, April 1998.

[22] M. Kamber, J. Han, and J. Y. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. In *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD'97)*, pages 207–210, Newport Beach, California, August 1997.

[23] M. Kamber, L. Winstone, W. Gong, S. Cheng, and J. Han. Generalization and decision tree induction: Efficient classification in data mining. In *Proc. of 1997 Int. Workshop Research Issues on Data Engineering (RIDE'97)*, pages 111–120, Birmingham, England, April 1997.

[24] K. Koperski and J. Han. Discovery of spatial association rules in geographic information databases. In *Proc. 4th Int'l Symp. Large Spatial Databases (SSD'95)*, pages 47–66, Portland, Maine, Aug. 1995.

[25] M. Mehta, R. Agrawal, and J. Rissanen. SLIQ: A fast scalable classifier for data mining. In *Proc. 1996 Int. Conf. Extending Database Technology (EDBT'96)*, Avignon, France, March 1996.

[26] R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. 1994 Int. Conf. Very Large Data Bases*, pages 144–155, Santiago, Chile, September 1994.

[27] R. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules. In *submitted for publication*, 1997.

[28] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[29] R. Srikant and R. Agrawal. Mining generalized association rules. In *Proc. 1995 Int. Conf. Very Large Data Bases*, pages 407–419, Zurich, Switzerland, Sept. 1995.

[30] B. Xia. Similarity search in time series data sets. In *M.Sc. Thesis, Simon Fraser University*, Burnaby, B.C., Canada, December 1997.

[31] O. R. Zaïane, M. Xin, and J. Han. Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs. In *Proc. Advances in Digital Libraries Conf. (ADL'98)*, Santa Babara, CA, April, 1998.

[32] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data*, pages 103–114, Montreal, Canada, June 1996.

[33] Y. Zhao, P. M. Deshpande, and J. F. Naughton. An array-based algorithm for simultaneous multidimensional aggregates. In *Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data*, pages 159–170, Tucson, Arizona, May 1997.