Building Data Mining Solutions with SQL Server 2000

Microsoft Corp. Zhaohui Tang · Pyungchul (Peter) Kim

1. Introduction

Data Mining receives more and more attention these days. Data mining, as we use the term, is the exploration and analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns and rules. These patterns and rules will help corporations to improve its marketing, sales, and customer support operations through better understanding of their customers. Through the years, corporations have accumulated very large databases from applications such as ERP, CRM or other operational systems. People believe there are untapped values hidden inside these data, to get these patterns out requires data mining techniques.

SQL Server 2000 has introduced data mining features for the first time. Two scalable data mining algorithms are included: Microsoft Decision Trees and Microsoft Clustering. Both algorithms are developed and patented by Microsoft Research.

Microsoft data mining provider becomes part of Analysis Service, which origins from OLAP Services in SQL Server 7.0. The Analysis Services of SQL Server 2000 has two components: OLAP and data mining. Both data mining and OLAP are important techniques for data analysis, but the related technologies are different. The automatic or semi automatic pattern discovery is one of the differences between data mining and OLAP. Data mining combines techniques developed in artificial intelligence, database and statistics, while OLAP is mainly based on SQL plus certain aggregation techniques. The term often employed in OLAP is multi-dimensional database, or so called data cube. For example, a sales cube can be built on top of the sales table that has a number of dimensions such as Product, Region, and Time. Each cells in the cube gives the aggregated sales value for a particular product, region and time period.

Data mining and OLAP are complementary as they are both analytical tools. For example, in the customer dimension of a sales cube, there are lots of members, which is very difficult to find customer's buying patterns. Data mining techniques can be applied to analyze this dimension to find out what are the clusters among the customers based on customer member properties and measures. SQL Server Analysis Services has these advanced features that bridge data mining and OLAP together. We will not address details of these features in this article.

2. OLE DB for Data Mining

Data mining is a relative young and promising area. However, the data mining industry today is highly fragmented, making it difficult for application software vendors and corporate developers to integrate different knowledge-discovery tools. We can consider the current data mining market similar to the database market before SQL was introduced. Every data mining vendor has its own data mining package, which does not communicate with other products. For example, a customer is interested in decision tree algorithm from Vendor A and has built the data mining application based on Vendor A's package. Later on, the customer finds the time series algorithm from vender B is also very attractive for prediction tasks, he faces difficult situation as product A and B has no common interface and he has to restart the whole project from the very beginning.

Most data mining products are horizontal packages and are difficult to be integrated in user applications such as customer care, CRM, ERP. With the help of OLE DB for DM Specification[4], any data mining algorithms can be accessed through OLE automation, which can be easily embedded into any consumer applications.

Another problem of most commercial data mining products is that data extraction from relational database to an intermediate storage format is necessary. Data porting and transformation are very expensive operations. Why can't do data mining directly on relational database where most data are stored?

To solve these problems, Microsoft has initiated the work of OLE DB for Data Ming(DM) Specification with more than 40 ISVs in the business intelligence fields since last summer. Its goal is to provide an industry standard for data mining so that different data mining algorithms from various data mining ISVs can be easily plug into consumer applications. Those software packages that provide data mining algorithms are called Data Mining Provider, those applications that use data mining features are called Data Mining Consumer. OLE DB for DM specifies the common interface between Data Mining Consumer and Data Mining Provider.

Figure 1 shows the tree basic steps of data mining process: model creation, training and prediction.

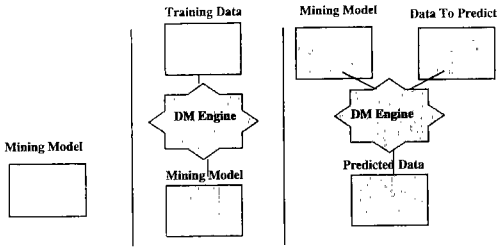

Figure 1 Data Mining Process

2.1 Data Mining Model

Data mining model is a new concept introduced in OLE DB for DM. A data mining model can be considered as a relation table in the sense that it has a list of columns with different data types. Some of these columns are input columns while others are predictable columns. A data mining model is a container. However, data mining model is different to relational table as it doesn't store raw data, rather it stores the patterns that data mining algorithms have discovered in the relational table. A mining model also has to specify the data mining algorithm it is associated and the list parameters if exists any. To create a data mining model, OLE DB for DM adopts the similar table creation syntax in SQL. The following example is to create a mining model to predict credit risk level based on

customer demographic information using Microsoft Decision Trees algorithm:

```
CREATE MINING MODEL CreditRisk
(
 CustomerId     long     key,
 Profession     text     discrete,
 Income         text     disrete,
 Age            long     continuous,
 RiskLevel      text     discrete  predict,)
USING [Microsoft Decision Tree]
```

2.2 Training a Mining Model

When a data mining model is created, it is an empty container. During the training stage, the data mining algorithm analyzes the input cases and populates the patterns it has discovered to the mining model. According to OLE DB for DM Specification, training data can be from any tabular data source as long as there is a proper OLE DB driver. It does not require users to export data from relational source to any special intermediate storage format. This largely simplifies the process of data mining. To be consistent with SQL, OLE DB for DM adopts the syntax of data insertion query. The following sample trains the CreditRisk mining model with the data stored in the customers table of a SQL Server database.

```
INSERT INTO CreditRisk
(
 CustomerId,Profession,Income,
 Age,RiskLevel
)
OPENROWSET('sqloledb', 'sa'; 'mypass'; '' ,
'SELECT CustomerID, Profession, Income,
        Age, Risk
 FROM Customers'
)
```

Openrowset command can access remote data from an OLE DB data source. SQL Server 2000 ships OLE DB drivers for SQL Server, Access and Oracle. Training process may take some time as during this stage, the data mining algorithm goes through all the input cases and does some complicated calculations. After training, the data mining algorithms find the patterns, which are persisted inside the data mining model. Users can browser the mining model to look at the discovered patterns, or use the trained mining model for prediction tasks.

2.3 Prediction

Prediction is an important data mining task. It requires two elements: a trained data mining model and a set of new cases. The result of prediction is a new recordset that contains values for predictable columns as well as other input columns. The overall process is very similar to relational join. Instead of joining two tables, prediction joins a data mining model with an input table. Thus we introduce a new concept called Prediction Join. The following example shows the syntax of a prediction join:

```
SELECT
   Customers.ID,
   CreditRisk.RiskLevel,
   PredictProbability(CreditRisk.RiskLevel)
FROM CreditRisk PREDICTION JOIN
     Customers
ON   CreditRisk.Profession =
Customers.Profession AND
CreditRisk.Income =
Customers.Income AND
CreditRisk.Age =
Customers.Age
```

Usually prediction can be done on the fly. It is also possible to do prediction on single

case instead of a set of new cases. We call these prediction queries singleton queries.

OLE DB for DM has also defined a list of prediction functions that can be included in the select clause of the prediction statement. These functions will return the probability of the predicted value, histogram information about other possible values and related probabilities, top counts, cluster id, etc.

2.4 Schema Rowsets

The schema information specified in OLE DB is based on the assumption that providers support the concepts of a catalog and a schema. Schema information can be retrieved in predefined schema rowsets. In OLE DB for Data Mining, we have predefined a list of schema rowsets. These schema rowsets help applications to dynamically discover the available data mining services and their parameters, existing mining models and their contents. For example, after training a mining model using decision tree algorithms, the content schema rowset contains the tree nodes information. Consumer application can display the tree graphically based on the content schema rowset.

3. Data Mining in SQL Server 2000

3.1 Components Architecture

Microsoft has implemented an OLE DB provider for Data Mining based on the OLE DB for DM Specification. The provider includes two data mining algorithms: Microsoft Decision Tree and Microsoft Clustering. Both algorithms are the result of the start of the art research work by Microsoft Research. Data mining provider is part of the Analysis Services 2000(used to be called OLAP Services in SQL Server 7.0). Similar to Microsoft OLAP Services, the data mining component in SQL Server 2000 is mainly target to DBAs. There is no sophisticated GUI component for data mining and Microsoft is working closely with ISV partners to build these general consumer tools. However, there are a few data mining GUI components for data mining in the Analysis Services. These components include model creation wizard, model editor, model content browser, and DTS task for prediction. Figure 2 shows the major components in Analysis Services 2000.

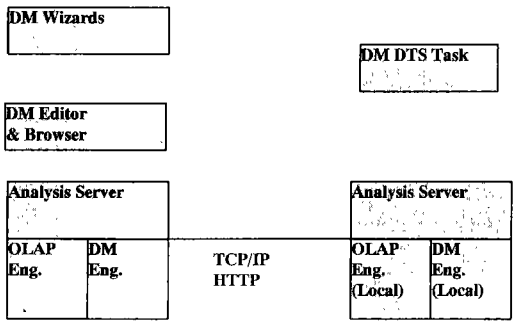

Figure 2 Analysis Services 2000

3.2 Data Mining Algorithms

Microsoft has implemented two representative data mining algorithms: Microsoft Decision Tree and Microsoft Clustering. Decision trees are widely being used for classification tasks. Unlike other classification algorithms such as nearest neighbor, neural networks, regression-based statistical techniques, decision trees can handle high dimensional data and the rules found can be more easily understood.

In order to achieve highly scalable classification over a large database, we implemented the execution module proposed in[2]. The execution module batches execution of multiple queries for the classification

client in a single scan of data to compute statistics, and appropriately stages data from server to client file system and to client main memory. It also has optimization facility to tradeoff the cost of scanning data at the server versus use of in-memory operations depending on the available client memory size.

The Microsoft clustering algorithm is a scalable implementation of Expectation-Maximization(EM) algorithm[1]. Unlike distance-based algorithms such as K-Means, EM constructs proper statistical models of the underlying data source and naturally generalizes to cluster databases containing both discrete-valued and continuous-valued data. The scalable method is based on a decomposition of the basic statistics the algorithm needs: identifying regions of the data that are compressible and regions that must be maintained in memory. The approach operates within the confines of a limited main memory buffer and requires at most a single database scan. Data resolution is preserved to the extent possible based upon the size of the main memory buffer and the fit of the current clustering model to the data.

In case when there are too many attributes involved in training(e.g., more than 255 attributes by default), we apply a feature selection method to filter out less interesting attributes. The interestingness of an attribute is calculated based on the entropy of the attribute found in[3].

4. Building Data Mining Application using Analysis Services

One of the biggest advantages of Microsoft Data Mining solution is that it is based on OLE DB for DM specification. It is fairly easy to use; any database developer can develop applications using data mining feature. The data mining language is very similar to SQL. Microsoft data mining provider is open as it is an OLE DB component. Algorithms from other ISVs can be plugged into the same platform. Data mining services can be invoked from any consumer application through DSO(Decision Support Object) or ADO object.

For example, a bank is developing its loan application software. It would like to add data mining feature to evaluate the loan risk for each applicant. The application is an n-tier intranet application. Loan assistant input customer information through a web form and then by clicking submit button, the associated loan risk indication will be displayed through an Active Server Page. The architecture of the application is displayed in Figure 3:

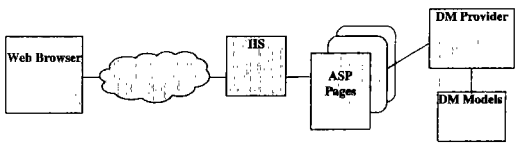

Figure 3 Building data mining applications

Before building the application, the first thing to do is to create a data mining model and train the model. There are a few different ways to do this task. The easiest way is to use the mining model creation wizard of Analysis Manager. The wizard will generate data mining creation and training queries and send this queries to Microsoft data mining provider through OLE DB for DM interface. The other way is to write some VB or C++ code to connect to the data mining provider through ADO or DSO, and then issue the text queries to the provider in a similar way as database developer does with database queries. The following example shows how to connect to the data mining provider through ADO

object in the ASP page.

```
Set con =
server.CreateObject("adodb.connection")
con.open "provider=msolap; Data
Source= myserver;initial catalog=Data
Mining Database"
```

The msolap provider will then initialize the Microsoft data mining provider.

To do prediction, it should first create a prediction query in the ASP page. In our example, as loan assistant only inputs information about one loan candidate at a time, so the prediction join is on singleton cases. The query is shown as the following:

```
QueryText="Select t.CustomerId,
CreditRisk. RiskLevel"
QueryText=QueryText + "From CreditRisk
natural prediction join ("
QueryText = QueryText + "Select 100 as
CustomerId, 'Engineer' as - Profession,
50000 as Income, 30 as Age) as t"
```

To execute the prediction query, it is exactly the same as to do a database query through ado:

```
Set rs = con.Execute (QueryText)
```

The prediction will return a record set, which has two columns: CustomerId and predicted credit risk level.

Analysis Services 2000 has extended its DSO model to support data mining. It is now possible to connect data mining provider through DSO. There are a few advantages of using DSO objects other than simple ADO connection, for example, better security control about using the model, remote mining services, repository support for mining models. However, with DSO programming, more coding is required and the developer needs to specify attributes for each column objects and mining model objects.

Sometimes consumer applications may like to show the content of a mining model. Model content represents the patterns the data mining algorithm found in the dataset. The mining content for a tree model represents the tree graph. Developer can get these information through content schema rowset. The following code shows how to connect to the content schema rowset. The attributes of the content schema rowset are defined in OLE DB for DM specification. The following is the definition of Model Content schema rowset.

```
const
DMSCHEMA_MINING_MODEL_CONTE
NT = "{3add8a76-d8b9-11d2-8d2a-
00e029154fde}"
set rs =
con.OpenSchema(adSchemaProviderSpecif
ic, ,
DMSCHEMA_MINING_MODEL_CONTE
NT )
```

5. Conclusion

With Analysis Services of SQL Server 2000, data mining is no longer a reserved domain for statisticians. The complexity of the data mining algorithms is hidden from the user. Every database developer will be able to create and train data mining models and to embed these advanced features into their consumer applications. Data mining will soon become a widely used analytical technique.

References

[1] Paul Bradley, Usama Fayyad, Cory Reina, Scaling EM (Expectation Maximization) Clustering to Large Databases,

Microsoft Tech. Report MSR-TR-98-35, Microsoft, 1998.
[2] Surajit Chaudhuri, Usama Fayyad, Jeff Bernhardt, Scalable Classification over SQL Databases. ICDE 1999, pp. 470-479.
[3] Huan Liu, Hiroshi Motoda (ed.), Feature Extraction, Construction and Selection:A Data Mining Perspective, Kluwer Academic Publishers, 1998.
[4] Microsoft, OLE DB for Data Mining Specifications, July 2000, www.microsoft.com/data/oledb/dm.

ZhaoHui Tang

1993 University of Versailles (M.S.)
1996 University of Versailles (Ph.D.)
1995~1996 Software developer, TechGnosis
1996~1999 Researcher at Sema Group Corp, Paris
1999~present, Program Manager at Microsoft Corp.
E-mail:zhaotang@microsoft.com

Pyungchul (Peter) Kim

1986 Seoul Nat'l Univ. (B.S.)
1989 KAIST (M.S.)
1994 KAIST (Ph.D.)
1991~1992 Software developer, UniSQL, Inc.
1993~1995. Senior researcher, ETRI
1995 Software dev. manager, Korea Computer Communications, Ltd.
1995~1998. Assist. Prof., Chungnam Nat'l Univ.
1997~1999. Chief software eng., Cyber Database Solutions, Inc.
1999~present. Software design eng., Microsoft Corp.
E-mail:Peterkim@microsft.com