API Standardization Efforts for Data Mining

Jaroslav Zendulka
Brno University of Technology, Czech Republic

INTRODUCTION

Data mining technology just recently became actually usable in real-world scenarios. At present, the data mining models generated by commercial data mining and statistical applications are often used as components in other systems in such fields as customer relationship management, risk management or processing scientific data. Therefore, it seems to be natural that most data mining products concentrate on data mining technology rather than on the easy-to-use, scalability, or portability. It is evident that employing common standards greatly simplifies the integration, updating, and maintenance of applications and systems containing components provided by other producers (Grossman, Hornick, & Meyer, 2002). Data mining models generated by data mining algorithms are good examples of such components.

Currently, established and emerging standards address especially the following aspects of data mining:

- **Metadata:** for representing data mining metadata that specify a data mining model and results of model operations (CWM, 2001).
- **Application Programming Interfaces (APIs):** for employing data mining components in applications.
- **Process:** for capturing the whole knowledge discovery process (CRISP-DM, 2000).

In this paper, we focus on standard APIs. The objective of these standards is to facilitate integration of data mining technology with application software. Probably the best-known initiatives in this field are OLE DB for Data Mining (OLE DB for DM), SQL/MM Data Mining (SQL/MM DM), and Java Data Mining (JDM).

Another standard, which is not an API but is important for integration and interoperability of data mining products and applications, is a Predictive Model Markup Language (PMML). It is a standard format for data mining model exchange developed by Data Mining Group (DMG) (PMML, 2003). It is supported by all the standard APIs presented in this paper.

BACKGROUND

The goal of data mining API standards is to make it possible for different data mining algorithms from various software vendors to be easily plugged into applications. A software package that provides data mining services is called *data mining provider* and an application that employs these services is called *data mining consumer*. The data mining provider itself includes three basic architectural components (Hornick et al., 2002):

- ***API* the End User Visible Component:** An application developer using a data mining provider has to know only its API.
- **Data Mining Engine (or Server):** the core component of a data mining provider. It provides an infrastructure that offers a set of data mining services to data mining consumers.
- **Metadata Repository:** a repository that serves to store data mining metadata.

The standard APIs presented in this paper are not designed to support the entire knowledge discovery process but the data mining step only (Han & Kamber, 2001). They do not provide all necessary facilities for data cleaning, transformations, aggregations, and other data preparation operations. It is assumed that data preparation is done before an appropriate data mining algorithm offered by the API is applied.

There are four key concepts that are supported by the APIs: a data mining model, data mining task, data mining technique, and data mining algorithm. The *data mining model* is a representation of a given set of data. It is the result of one of the *data mining tasks*, during which a *data mining algorithm* for a given *data mining technique* builds the model. For example, a decision tree as one of the classification models is the result of a run of a decision tree-based algorithm.

The basic data mining tasks that the standard APIs support enable users to:

1. *Build* a data mining model. This task consists of two steps. First the data model is defined, that is, the source data that will be mined is specified, the source data structure (referred to as *physical schema*) is mapped on inputs of a data mining algorithm (referred to as *logical schema*), and the algorithm used to build the data mining model is specified. Then, the data mining model is built from training data.

2. *Test* the quality of a mining model by applying testing data.
3. *Apply* a data mining model to new data.
4. *Browse* a data mining model for reporting and visualization applications.

The APIs support several commonly accepted and widely used techniques both for predictive and descriptive data mining (see Table 1). Not all techniques need all the tasks listed above. For example, association rule mining does not require testing and application to new data, whereas classification does.

The goals of the APIs are very similar but the approach of each of them is different. OLE DB for DM is a language-based interface, SQL/MM DM is based on user-defined data types in SQL:1999, and JDM contains packages of data mining oriented Java interfaces and classes.

In the next section, each of the APIs is briefly characterized. An example showing their application in prediction is presented in another article in this encyclopedia.

MAIN THRUST

OLE DB for Data Mining

OLE DB for DM (OLE DB, 2000) is Microsoft's API that aims to become the industry standard. It provides a set of extensions to OLE DB, which is a Microsoft's object-oriented specification for a set of data access interfaces designed for record-oriented data stores. It employs SQL commands as arguments of interface operations. The approach in defining OLE DB for DM was not to extend OLE DB interfaces but to expose data mining interfaces in a language-based API.

OLE DB for DM treats a data mining model as if it were a special type of "table:" (a) Input data in the form of a set of cases is associated with a data mining model and additional meta-information while defining the data mining model. (b) When input data is inserted into the data mining model (it is "populated"), a mining algorithm builds an abstraction of the data and stores it into this special table. For example, if the data model represents a decision tree, the table contains a row for each leaf node of the tree (Netz et al., 2001). Once the data mining model is populated, it can be used for prediction, or it can be browsed for visualization.

OLE DB for DM extends syntax of several SQL statements for defining, populating, and using a data mining model – see Figure 1.

SQL/MM Data Mining

SQL/MM DM is an international ISO/IEC standard (SQL, 2002), which is part of the SQL Multimedia and Application Packages (SQL/MM) (Melton & Eisenberg, 2001). It is based on SQL:1999 and its structured user-defined data types (UDT). The structured UDT is the fundamental

Table 1. Supported data mining techniques

Technique	OLE DB for DM	SQL/MM DM	JDM
Association rules	X	X	X
Clustering (segmentation)	X	X	X
Classification	X	X	X
Sequence and deviation analysis	X		
Density estimation	X		
Regression		X	
Approximation			X
Attribute importance			X

Figure 1. Extended SQL statements in OLE DB for DM