DeVisa
Concepts and Architecture of a Data Mining Models Scoring and Management Web System

Diana Gorea
Faculty of Computer Science, University "Al. I. Cuza", 16 G-ral Berthelot, 700483 Iasi, Romania

Keywords: PMML, native XML database, data mining model, knowledge discovery, data integration, XQuery.

Abstract: In this paper we describe DeVisa, a Web system for scoring and management of data mining models. The system has been designed to provide unified access to different prediction models using standard technologies based on XML. The prediction models are serialized in PMML format and managed using a native XML database system. The system provides functions such as scoring, model comparison, model selection or sequencing through a web service interface. DeVisa also defines a specialized PMML query language named PMQL used for specifying client requests and interaction with PMML repository. The paper analyzes the system's architecture and functionality and discusses its use as a tool for researchers.

1 INTRODUCTION

The Knowledge Discovery in Databases (KDD) process is defined to be the non-trivial extraction of implicit, previously unknown and potentially useful patterns from data (Frawley and Piatetsky-Shapiro, 1991). KDD is a field that has evolved from the intersection of research fields such as machine learning, pattern recognition, databases, statistics, AI, knowledge acquisition from expert systems, data visualization and high performance computing. The unifying goal is to obtain high-level knowledge from low-level data. Knowledge can be represented in many forms, such as logical rules, decision trees, fuzzy rules, Bayesian belief networks, and artificial neural networks. The data mining component of KDD relies on techniques from machine learning, pattern recognition or statistics to build patterns or models that are further used to extract knowledge. Thus data mining applies algorithms that -under acceptable computational efficiency limitations- produce the patterns or models.

In general, building data mining models is very expensive because it involves analyzing large amounts of data. In most of the cases the costs are dominated by the cost of pre-processing the data (collecting, cleaning, transforming, preparing). In the case in which the data resides in different heterogeneous sources an integration phase adds up to the process, making it even more expensive. In the modeling phase the complexity of the algorithms generally depend on both the size of the schema and the number of data instances. A systematic overview of data mining algorithms as well as a comprehensive discussion on the computational complexity can be found in (Hand et al., 2001) and (Ian H. Witten, 2005).

The natural consequence is that knowledge has a broader purpose beyond its direct use in the system that has built it. Thus knowledge should be shared and transfered between different systems. During the past several years, the Data Mining Group (DMG, 2007) has been working in this direction specifying the Predictive Model Mark-up Language or PMML (PMML, 2007), a standard XML-based language for interchanging knowledge between applications.

The models expressed in PMML serve predictive and descriptive purposes. The predictive property means that models produced from historical data have the ability to predict future behavior of entities. The descriptive property is when the model itself is inspected to understand the essence of the knowledge found in data. For example, a decision tree can not only predict outcomes, but can also provide rules in a human understandable form. Clustering models are

Gorea D. (2008).
DeVisa - Concepts and Architecture of a Data Mining Models Scoring and Management Web System.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - DISI*, pages 276-281
DOI: 10.5220/0001706202760281
Copyright © SciTePress

not only able to assign a record to a cluster, but also provide a description of each cluster, either in the form of a representative point (the centroid), or as a rule that describes why a record is assigned to a cluster.

This paper proposes a system called DeVisa (DeVisa, 2007) that is intended to provide feasible solutions for dynamic knowledge integration into applications that do not necessarily have to deal with knowledge discovery processes. Nowadays the true value of KDD is not relying on its collection of complex algorithms, but moreover on the practical problems that the knowledge produced by these algorithms can solve. Therefore DeVisa focuses on collecting such knowledge and further providing it as a service in the context of Semantic Web applications. It manages a repository of PMML models as a native XML database and exploits their predictive or descriptive nature through a specialized PMML query language named PMQL, being part of DeVisa. The current work extends some ideas depicted in (Gorea, 2007), in which the means through which the models should be processed were not very clearly specified.

This paper is organized as follows. Section 2 describes the architectural and functional perspective on DeVisa as well as the PMQL language. Section 3 presents the founding technologies and systems on which the proposed system is based, such as PMML, XQuery, eXist, Weka and web services. Section 4 presents the related work in the field and the position of the current work in this context. Section 5 presents an usage example of the system in the micro-RNA discovery problem, while Section 6 lists the conclusions and further work.

2 THE DeVisa SYSTEM

2.1 Functional Perspective

The core functionality of DeVisa is available via web services (Studer et al., 2007). The client application communicates with a DeVisa web service through messages (request, answer) specified in PMQL (see 2.3). The main functional capabilities are described below.

Scoring. The client application wants to score one of the existing models in the repository on a set of instances. The model can be specified by its unique name that includes the producer's namespace, or can be selected by specifying desired characteristics related to freshness, performance measures (an overview of the existing model assessment techniques and measure is available in (Cios et al., 2007)), producer application, complexity etc.

Search. DeVisa provides searching functions in the catalog as selecting the models with desired properties (e.g performance measures, fields statistics, function type, degree of freshness etc.) or full text search in the model repository.

Model Comparison. Two schema compatible models can be compared from a qualitative perspective (e.g accuracy in the case of supervised models) and from a structural perspective (through XML differencing).

Model Composition. Two or more PMML models can be combined into a single composite PMML model. The PMML specification describes two types of composition: model selection (e.g. a decision tree model can be used to select among other compatible models) or model sequencing (e.g. the output of a model can be used as input to another model). A client application can request for a composite model based on existing models in the DeVisa repository.

Statistics. An application can invoke this service to obtain statistics on the models (e.g frequencies per domain, schema, or function type etc.).

2.2 Architectural Perspective

The main architectural components of the DeVisa system are depicted in Figure 1.

The PMML Model Repository is a collection of models stored in PMML format that uses the native XML storage features provided by the underlying XML database system. A PMML document contains one or more models that share the same schema. The models are organized in collections (corresponding to domains) and identified via XML namespace facilities (connecting to the producer application). The documents in the repository are indexed for fast retrieval (structured indexes, full-text indexes and range indexes).

The Catalog contains metadata about the PMML models stored in a specific XML format. The catalog XML Schema can be found in (DeVisa, 2007). The catalog consists of the following type of information: available collections , model schema, model information (algorithm, producer application, upload date), statistics (e.g univariate statistics: mean, minimum, maximum, standard deviation, different frequencies), model performance (e.g precision, accuracy, sensitivity, misclassification rate, complexity), etc.

The PMQL-LIB module is a collection of functions entirely written in XQuery for the purpose of PMML querying. The functions in the PMQL-LIB module are called by the PMQL engine during the

query plan execution phase (See 2.3.3).

The PMML Model Service is a web service that provides different specialized operations corresponding to the capabilities listed in 2.1. It is an abstract computational entity meant to provide access to the aforementioned concrete services. Thus the PMML Model Service receives and returns SOAP messages that contain queries expressed in PMQL (See 2.3). To solve the incoming requests the web service detaches the PMQL fragment to the PMQL Engine.

The PMQL Engine is a component of the DeVisa system that processes and executes a query expressed in PMQL (See 2.3). After syntactic and semantic validation, query rewriting, it executes the query plan by invoking functions in the PMQL-LIB internal module.

The Admin PMML Service is based on XML-RPC or SOAP protocols and consists of methods for storing and retrieving PMML models. DeVisa redefines the basic SOAP store / retrieve web service with customized PMML features. Therefore, when a model is uploaded in the repository, it is validated against the PMML Schema or by using the XSLT based PMML validation script provided by DMG. Then the model is distributed in the appropriate collection (based on the domain / producer) and the catalog is updated with the new model's metadata. Also the service provides features for updating/replacing an existing model with a newer one via XUpdate ((XUpdate, 2003)) instructions.

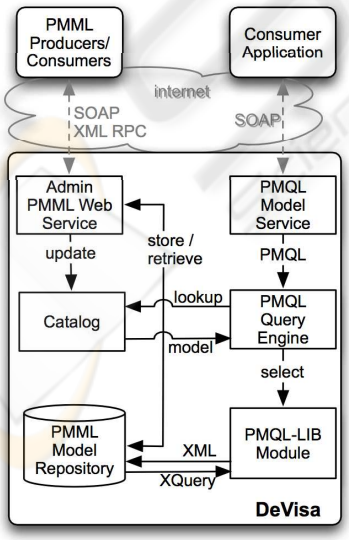

Figure 1: The architecture of DeVisa.

2.3 PMQL

PMQL (Predictive Model Query Language) is a query language with an XML syntax defined by DeVisa for the purpose of interacting with PMML documents. A PMQL query is executed against the DeVisa repository of prediction or description models stored in PMML format. The PMQL schema and specifications can be found in (DeVisa, 2007). A client application can wrap a PMQL query in a SOAP message and invoke a service method. Depending on the task, a PMQL query specifies the target models it is based on, such as the schema, the instances to predict etc. PMQL can express queries that correspond to core DeVisa functionality (See 2.1).

Bellow a simple PMQL query fragment containing a classification task is presented. The full example is listed in the PMQL use cases (DeVisa, 2007).

```
<pmql><request>
 <score function="classification">
  <modelSpec> <schema>
      <field id="01" name="MFE"
           opttype="continuous"
           datatype="xs:float" usage="active">
          <description>
              Minimal free energy
          </description> </field>
      <field id="02" name="BN"
           opttype="continuous"
           datatype="xs:int"  usage="active">
            <description>
              Number of Bulges
            </description> </field> ...
    </schema></modelSpec></score>
    <data> <instance>
            <field idref="#01" value="-20"/>
            <field idref="#02" value="5"/>...
        </instance> ...</data>
</request></pmql>
```

A query expressed in PMQL is executed in DeVisa in several phases described below.

2.3.1 Query Annotation

In this phase DeVisa PMQL engine performs syntactic checking against the PMQL Schema as well as semantic checking.

The semantic checking involves a look-up in the repository catalog for a model that satisfies the query constraints (function type, accuracy, complexity, freshness) and whose schema matches the query schema. The look-up can be strict (exact matching of the schemas) or lax (compatible schemas). If the lax look-up is enabled then the schemas need only to be compatible with respect to the number of fields and field types. Should this be the case the subsequent

phase performs the actual mapping between the fields of the two schemas.

If several models fulfill the query requirements, a sequence of model references is returned. If no model entry is complying with the requested properties found in the catalog, a null reference is returned and the execution of the current query stops.

2.3.2 Query Rewriting

This phase involves solving the schema differences between the query and the sequence of models returned in the first phase, provided that the look-up procedure was lax and it returned something (one model or a sequence of matching models). Basically the goal is to rewrite the query in the terms of the matching models. This procedure solves the type and name differences. Types are resolved by applying the allowed type conversions as specified in the XMLSchema.

Name ambiguity solving is subject to intense research in the context of achieving the Semantic Web desiderates. There are two possible approaches to bypass the naming differences. The first approach is the use of ontologies to mediate between the heterogeneous schemas. One of the promising fields of application of ontologies in Semantic Web is the integration of information from differently organized sources, e.g to interpret data from one source under the schema of another, or to realize unified querying and reasoning over both information sources. DeVisa can refer URI-identified resources in shared ontologies, therefore aiming to conform to Semantic Web model (SW, 2007).

Another approach for name solving is the use of fuzzy matching techniques against the textual description of the fields. This is less precise than the former and can be used as an alternative technique in case no ontology is available. The DeVisa support for name ambiguity solving is currently under development.

The outcome of this phase is a rewritten PMQL query (or a sequence of queries) which is (are) reinterpreted and transformed with respect to the internal schema of the matching PMML model (or models) in the DeVisa repository.

2.3.3 Plan Building and Execution

The query plan contains a reference to the models subject to query processing, a reference to an XML document containing the instances with the schema reinterpreted with respect to the aforementioned model and a reference to a function in the PMQL-LIB. In case there are several models that satisfy the query requirements (see 2.3.2), the query plan is made of a sequence of execution units, corresponding to each model. The query plan is dispatched to the appropriate XQuery function in the PMQL-LIB and the XQuery engine executes the function on the models and the instances. The result of this phase is an XML document containing the query responses.

3 OVERVIEW OF THE UNDERLYING TECHNOLOGIES AND SYSTEMS

PMML. The PMML language provides a declarative approach for defining self-describing data mining models. In consequence it has emerged as an interchange format for prediction models. It provides a clean interface between producers of models, such as statistical or data mining systems, and consumers of models, such as scoring systems or applications that employs embedded analytics. A PMML document can describe one or more models and includes components such as: data dictionary that is common to all models (contains fields used in the models in the document), mining schema (to identify the fields used in a specific model), transformation dictionary (defines derived fields from the fields in the mining schema) and model statistics. The application producer (in our case the system is tested with Weka) can use the custom DeVisa libraries based on SOAP or XMLRPC web services to upload and register PMML models in the system.

Weka. (Weka, 2007) is machine learning and data mining software written in Java and distributed under the GNU Public License, which is used for research, education, and applications. The main features include a comprehensive set of data pre-processing tools, learning algorithms and evaluation methods, a graphical user interface (including data visualization), an environment for experimenting and comparing learning algorithms, and a knowledge flow. Weka includes all the four basic styles of learning that appear in data mining applications: classification, regression, clustering and association (Ian H. Witten, 2005). At the time this material was written, Weka did not provide support for PMML, but it is very likely that it will be integrated in future versions. Such a module (although it does not support all the models in Weka) is part of the DeVisa system. Otherwise Weka does not interact directly with DeVisa, nor does it have any other particular role except for the role of PMML producer (or consumer).

XQuery and eXist. eXist (eXist, 2007) is a native

XML database engine featuring efficient, index-based XQuery processing, automatic indexing (structural, full-text, range), extensions for full-text search, XUpdate support, XQuery update extensions and tight integration with existing XML development tools. eXist provides basic database functions such as storing and retrieving XML or binary resources and XQuery/XPath querying via the XMLDB, XMLRPC or SOAP interfaces.

The core functionality of DeVisa is implemented in XQuery (Boag et al., 2007). XQuery is a functional language designed for querying XML documents and it has become a W3C recommendation as of January 2007. The language has the following distinctive properties: composition, closure, schema conformance, XPath compatibility, completeness, conciseness, and static analysis.

Web Services. The DeVisa system provides its functionality via web services definitions. The Admin PMML Service supports SOAP, XMLRPC and REST-style protocols. The PMML Model Service supports only SOAP messaging. The web services are described using WSDL, are written in Java and deployed using Apache Axis framework ((Axis, 2007)).

4 RELATED WORK

Due to the increasing usage of the web space as a platform (one of the main goals of Web 2.0), there are many proposals of data mining services available on the web (Chieh-Yuan Tsai, 2005). (Grigorios Tsoumakas, 2007) describes a web-based system for classifier sharing and fusion named CSF/DC. It enables the sharing of classification models, by allowing the up- and download of such models expressed in PMML on the system's online classifier repository. It also enables the online fusion of classification models located at distributed sites, which should be a priori registered in the system using a Web form. Unlike CSF/DC, DeVisa only supports processing of local models exposing the functionality as web services.

The idea of storing the PMML models in a database was also depicted in (Chaves et al., 2006), which presents a PMML-based scoring engine named Augustus. Augustus is an open source infrastructure for building and deploying data mining, and statistical models for large data sets and high volume data streams.

DeVisa is particular in the following aspects:

- DeVisa does not store the data the models were built on, but only the PMML models themselves;
- The DeVisa approach leverages the XML native storage and processing capabilities like indexing (structural, full text and range), query optimization, inter-operation with the XML based family of languages and technologies;
- DeVisa defines a XML based query language - PMQL - used for interaction with the DM consumers. PMQL is wrapped in a SOAP message, interpreted within DeVisa and executed against the PMML repository;
- The interoperability with other applications (e.g consumers) is achieved exclusively through the use of web services;
- DeVisa integrates a native XQuery library for processing PMML documents.

5 FUTURE USAGE OF DeVisa

In the last years, the information stored in biological sequence databases grew up exponentially, and new methods and tools have been proposed to get useful information from such data. One of the most actively researched domains in bioinformatics is the micro-RNA discovery, i.e. discovering new types of micro-RNA in genomic sequences. There has been a great amount of published significant results concerning micro-RNA discovery with Decision Tree Classifiers (Ritchie et al., 2007), Support Vector Machines as in (Sung-Kyu, 2006) and (Sewer, 2005) etc., Hidden Markov Models (Nam, 2005) etc., Association Rules, etc.

All these results are independent and, as far as known at this moment, there is no system that integrates all these results in a semantic manner. There is a large availability of prediction models in the scientific community - like the ones listed above, public databases of known micro-RNAs as miRBase, e.g (Griffiths-Jones et al., 2006), public web ontologies as Gene Ontology (GO, 2007), or online services for characterizing genomic sequences (for instance RNAFold that analyses RNA secondary structure). DeVisa will be used to integrate all the knowledge resulting from sparse scientific work concerning the micro-RNA discovery problem into a web knowledge base platform that is easily integrable into the future work in the field.

6 CONCLUSIONS AND FURTHER WORK

This paper presented the DeVisa system, a web architecture for management of data mining models. Besides basic functions such as uploading/downloading

models, DeVisa includes a web service for scoring, composing the models, comparing, searching on the stored models. DeVisa also includes a library that builds PMML based on Weka classifier model classes using the Java Reflection API. It represents work in progress and at the time of writing of this material it provides support only for several classification models (trees, naive bayes, rule set) and association rules. The system will be extended to support most of the models supported by the PMML specification.

The freshness of a model is one of the factors where the accuracy of a model strongly depends on. In the current implementation the freshness of a model is a feature that depends only on a model producer. DeVisa cannot control that aspect, so models can get outdated very easily. However the consumer can specify the freshness and DeVisa can rank the models from that point of view. One possible solution is that DeVisa will trigger the upload of the models using web service interfaces of model producers.

Another item which is part of the DeVisa roadmap is providing a mash-up interface for the scoring services. The system will be able to recommend the appropriate service by selecting the appropriate model using a technique similar with the one used in the look-up phase of the PMQL execution (See 2.3.1).

In the DeVisa system the collection of models together with the related ontologies form a knowledge base. A good research direction is designing a technique of ensuring consistency of the knowledge base. Each upload triggers a check-up against the existing knowledge base and a conflict might occur. There are a few ways of dealing with conflicts: reject the whole knowledge base (FOL approach, but unacceptable under the web setup), determine the maximal consistent subset of the knowledge base, or use a para-consistent logic approach. The knowledge base facilitates that new knowledge is derived from it, so that DeVisa can include in the future rule management and inference engines (RuleML, 2007).

REFERENCES

Axis (2007). Apache axis. http://ws.apache.org/axis/.

Boag, S., Chamberlin, D., Fernandez, M., Florescu, D., Robie, J., and Simeon, J. (2007). Xquery 1.0: An xml query language. http://www.w3.org/TR/xquery/.

Chaves, J., Curry, C., Grossman, R. L., Locke, D., and Vejcik, S. (2006). Augustus: the design and architecture of a pmml-based scoring engine. In *DMSSP '06: Proceedings of the 4th international workshop on Data mining standards, services and platforms*, pages 38 – 46, New York, NY, USA. ACM.

Chieh-Yuan Tsai, M.-H. T. (2005). A dynamic web service based data mining process system. In *The Fifth International Conference on Computer and Information Technology CIT 2005*, pages 1033– 1039.

DeVisa (2007). Devisa. http://devisa.sourceforge.net.

DMG (2007). Data mining group. http://www.dmg.org.

eXist (2007). Exist - open source native xml database. http://www.exist-db.org/.

Frawley, W. and Piatetsky-Shapiro, G. (1991). *Knowledge Discovery In Databases: An Overview*. Knowledge Discovery In Databases. AAAI Press/MIT Press, Cambridge, MA.

GO (2007). The gene ontology.

Gorea, D. (2007). Towards storing and interchanging data mining models. In *Proceedings of the 3rd Balkan Conference in Informatics*, volume 2, pages 229–236.

Griffiths-Jones, S., Grocock, R., van Dongen, S., Bateman, A., and Enright, A. (2006). mirbase: microrna sequences, targets and gene nomenclature. *Nucleic Acids Res.*, 34:140 – 144.

Grigorios Tsoumakas, I. V. (2007). An interoperable and scalable web-based system for classifier sharing and fusion. *Expert Systems with Applications*, 33(3):716–724.

Hand, D., Mannila, H., and Smyth, P. (2001). *Principles of Data Mining*. The MIT Press.

Ian H. Witten, E. F. (2005). *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann series in data management systems. Elsevier, 2nd edition.

Nam, J.-W. (2005). Human microrna prediction through a probabilistic co-learning model of sequence and structure. *Nucleic Acids Research*, 33(11):3570–3581.

PMML (2007). Pmml version 3.2. http://www.dmg.org/pmml-v3-2.html.

Ritchie, W., Legendre, M., and Gautheret, D. (2007). Rna stem-loops: To be or not to be cleaved by rnase iii. *RNA*, 13:457–462.

RuleML (2007). Ruleml. http://www.ruleml.org.

Sewer, A. (2005). dentification of clustered micrornas using an ab initio prediction method. *BMC Bioinformatics*.

Studer, R., Grimm, S., and Abecker, A., editors (2007). *Semantic Web Services. Concepts, Technologies and Applications*, chapter 2. Springer.

Sung-Kyu, K. (2006). mitarget: microrna target-gene prediction using a support vector machine. *BMC Bioinformatics 2006*, 7(1):411.

SW (2007). W3c semantic web activity. http://www.w3.org/2001/sw/.

Weka (2007). Weka 3 - data mining software in java. http://www.cs.waikato.ac.nz/ml/weka.

XUpdate (2003). Xupdate. http://xmldb-org.sourceforge.net/xupdate/.