Integration of Prediction Services – A Practical Case Study

Diana Gorea

Abstract—This work investigates a practical use case for the prediction services integration problem. The use case is built upon the heart disease data set of the UCI machine-learning repository. Several prediction models based on various data sets (different schemas or instances) were built using Weka (classifiers, cluster models etc) and exported in PMML or XML. The front end is realized using plug-ins in Spoon interface of the Kettle integration tool. The goal of this work is to demonstrate how combining different classifiers can achieve better performance than single ones and to justify the necessity of integrating semantic schema matching techniques in such a system. Such a system can function as customizable decision making assistant.

Index Terms—prediction methods, decision-making, web technologies, schema integration

I. INTRODUCTION

This work contributes in extending the idea of reuse and sharing of knowledge obtained through the KDD processes in the context of KDD globalization. Therefore knowledge can be further stored, efficiently managed and provided via web services.

As [1] states, the future of KDD lies in achieving overall integration of its process through the use of standards based on XML, such as PMML. During the past several years, the Data Mining Group (DMG) ([6]) has been working in this direction specifying the Predictive Model Mark-up Language or PMML, a standard XML-based language for interchanging knowledge between applications such as data mining tools, database systems, spreadsheets and decision support systems. PMML can be also used for knowledge integration of results obtained by mining distributed data sets.

With PMML, statistical and data mining models can be thought of as first class objects described using XML. Applications or services can be thought of as producing PMML or consuming PMML. A PMML XML file contains enough information so that an application can process and score a data stream with a statistical or data mining model using only the information in the PMML file.

The work presents a system called DeVisa that is based on collecting knowledge in a domain and further providing it as a service to consumer applications. It 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 service oriented architectures.

DeVisa provides several services, whose general description was presented in [2] and [3]. The scoring service represents a mean to provide knowledge as a service in DeVisa. In this context [4] formalizes the main entities in the system and the online scoring problem. In addition the system provides a general context in which composition of data mining models can be achieved within the boundaries of this online scoring system ([5]). A PMML query language, called PMQL, used both for interacting with the consumer and for communication between DeVisa internal components is defined.

II. DESCRIPTION OF THE SYSTEM

A. Functionality

The most important function of the system is that of online scoring. It represents the process through which a data mining consumer requests through a web service call that a certain data set is annotated with prediction values based on one or more serialized data mining models. Producers and consumers of the models are different entities, so that it is possible that the schemas of the two do not correspond, although it might refer to the same entities. Furthermore, the consumer is not aware of the exact models that exist in the repository. In this case a schema matching procedure is integrated in the scoring process.

Briefly, it occurs when a consumer application would like to score a given data set having no knowledge if DeVisa contains a data dictionary (or schema) that matches the schema of its data set. When solving the consumer’s goal containing a laxly specified target DeVisa model, the scoring engine is performing a schema matching to identify an appropriate model or models to score on. During this process, it needs to iterate over the existing DeVisa data dictionaries.
and attempt to perform a mapping. On the other side one can imagine the case in only a subset consumer’s attributes are used in the prediction process, namely the maximal subset that can be mapped to a DeVisa data dictionary and for which an adequate mining schema can be deduced. Nevertheless the degree of coverage of the consumer’s set of attributes is a configurable parameter (i.e support).

One of the main characteristics of the DeVisa system is that it does not store the data the models were built on, but it is meant to enable model and prediction interchange between applications. Therefore some of the schema matching strategies (especially instance based strategies) that employ learning techniques cannot be applied. Nevertheless the consumer’s data that needs to be scored on is available and can be analyzed. On the other side a model expressed in PMML can contain univariate statistics on each of the fields used in the mining model. Therefore a technique that builds a “matching signature” in the manner presented in [15] could be applied for certain data dictionaries. To improve the efficiency of the matching during the model discovery phase of the preprocessing procedure the preprocessing should be done offline.

The schema matching technique is based on a modified version of the cycle canceling max-flow min-cost algorithm that allows integrating additional constraints such as derivability and validity. It also proposes an adaptive similarity measure based on string metrics, Jaccard index in the textual description, field statistics and lexical sense.

One of the important functionalities that DeVisa system provides is the composition of prediction models. In general the composition of prediction models can be realized in various ways. They all have in common the goal of making predictions more reliable. Composing several prediction models means merging the various outputs into a single prediction. Model composition is supported by the PMML specification in two major flavors: sequencing and selection.

**Model sequencing** is the process through which two or more models are combined into a sequence where the results of one model are used as input in another model. Model sequencing is supported partially by the PMML specification ([7]).

Examples of sequencing include:

- The missing values in a regression model can be replaced by a set of rules (or decision tree)
- Several classification models with the same target value can be merged via a voting scheme, i.e the final classification result can be computed as an average of the results of the initial classifiers. The average can be computed by a regression model.
- Prediction results may have to be combined with a cost or profit matrix before a decision can be derived. A mailing campaign model may use tree classification to determine response probabilities per customer and channel. The cost matrix can be appended as a regression model that applies cost weighting factors to different channels, e.g., high cost for phone and low cost for email. The final decision is then based on the outcome of the regression model.

**Model selection** is the process of combining multiple 'embedded models', such as model expressions, into the decision logic that selects one of the models depending on the current input values. For instance, a common method for optimizing prediction models is the combination of segmentation and regression. Data are grouped into segments and for each segment there may be different regression equations. If the segmentation can be expressed by decision rules then this kind of segment based regression can be implemented by a decision tree where any leaf node in the tree can contain an embedded regression model. PMML version 3.2 supports the combination of decision trees and simple regression models. More general variants would be possible and may be defined in future versions of PMML.

### B. Modules

From an structural perspective the system is formed of several important components that are described hereunder.

The **PMML Model Service** is a web service that provides different specialized operations corresponding to the capabilities listed in previous section. 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 [8]) To solve the incoming requests the web service detaches the PMQL fragment to the PMQL Engine.

The Admin PMML Service is based on XMLRPC 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 ([9]) instructions.

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 - DeVisa uses eXist ([10]) for this purpose. 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 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. A scoring
function has a PMQL query plan as input and produces a PMQL query answer.

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

The **Metadata Catalog** contains metadata about the PMML models stored in a specific XML format. The catalog consists of the following types 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. This component is a materialized view on the PMML repository containing information on the PMML models in the repository. In DeVisa the Metadata Catalog is strongly dependent on the underlying XML database indexing system so that the performance of the retrieval process is influenced by the active configuration in a particular database instance.

Because the PMML models can potentially originate from different PMML producer applications, the data dictionaries can be semantically heterogeneous, even if they are part of the same domain. Therefore the Catalog contains a component called **Global Data Dictionary (GDD)**, whose role is to provide unified view on the data dictionaries.

The GDD acts like a mediator for the existing data dictionaries in DeVisa. The mediator architecture conforms to a simplified Global as View (GAV) mediator model. The implementation of the GAV mediator architecture is depicted in Figure 1.

The mediation occurs only at the semantic level, because both the global data dictionary and the local ones are represented in the same XML schema. Therefore the wrapper components are lightweight, their function resuming to seamlessly mapping an entity in the global dictionary to the corresponding local one during the PMQL query processing. The GDD provides a matching between the fields in the data dictionaries, including the derived fields in the transformation dictionaries. For each attribute the matching is annotated with the confidence measure, meaning the degree of similarity between the respective fields. An example of such a matching is depicted in Figure 2. In this figure we have depicted two data dictionaries DD1 and DD2, containing two and one mining schemas respectively. The global data dictionary contains a set of virtual attributes that are connected with the original ones through edges labeled with a similarity measure (a value in the [0,1] interval). If the similarity measure between two attributes is higher than a threshold (that is known a priori), then the two attributes are considered similar and an edge is drawn between the two.

Figure 2. An example of the global data dictionary mediation between two data dictionaries in DeVisa.

### III. ABOUT THE DATA SET

In our tests we used the heart disease data set from the UCI machine learning repository ([12]), which was created by Andras Janosi, William Steinbrunn (University Hospital, Zurich, Switzerland), Matthias Pfisterer (University Hospital, Basel, Switzerland), and Robert Detrano (Cleveland Clinic Foundation).

This database contains 76 attributes, but all published experiments (such as [11], [13]) refer to using a subset of 14 of them. The predicted field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. The heart disease data set is horizontally fragmented into patient data collected from patients in Cleveland, Switzerland, Hungary or Long Beach. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).

There are different flavors of the Cleveland data set (the most used in the publications) in which values are expressed either as symbolic or as numeric.

### IV. THE FRONT END

Pentaho Data Integration (also called Kettle) ([16]) is a software component for the Extract, Transform and Load (ETL) processes. Though ETL tools are most frequently used in data warehouses environments, PDI can also be used for other purposes: migrating data between applications or databases, exporting data from databases to flat files, loading data massively into databases, data cleansing, integrating applications. It offers a graphical interface called Spoon that allows designing and testing of the integration processes through the use of jobs and transformations. Kettle presents an extensible plug-in architecture in which new functionality
can be added to the existing one.

Besides the built-in Kettle functions in our tests we use the following plug-ins: Weka Scoring, Univariate statistics. We created other DeVisa specific plug-ins, such as automatic schema matching, several voting procedures for implementing prediction method selection, global data dictionary mediator, predictive model repository access.

The DeVisa system itself does not have a user interface, all its functionality being available as web services. Therefore such an extensible plug-in based architecture specialized on integration and business intelligence can easily incorporate DeVisa functions and provide the same functionality to a human user. This approach has the advantage of being more flexible since the user can validate the results immediately and try alternative ways of achieving his/her goals.

Kettle also allows plug-ins that call web services, which making it suitable as a remote client (consumer) for DeVisa services.

V. METHODS AND RESULTS

We are interested in integrating heterogeneous prediction models into a common repository that further provides knowledge services to consumers. The heterogeneity can exist either at the schema level (different names of the attributes, different data types, different number of attributes) or at the model type level (cluster model, classification or regression models, association rules).

The general idea is to visually provide methods of obtaining best results by providing tools for mediating model differences and for composition of various classifiers.

As mentioned in section III, the heart disease data set is horizontally fragmented based on geographical localization of the patients. We make a set of basic assumptions:

1. The fields were named differently in different partitions and it is possible that different data types were used to represent the values of the respective fields.

2. In each of the partitions various prediction models have been built that act on vertical fragments of the data set. From the total of 76 attributes, the models can use projections on some subsets of these attributes, which can be overlapping or not.

One of the possible use cases is when we have a set of records belonging to patients in a different location for which prediction models have not been built. One can use a heuristic like “choose the prediction model built on patient data closest to the location of the test data”. However this is not applicable when we do not have the actual location of the patients. Furthermore we can encounter again schema heterogeneity (the new data has its own schema). What we try to do then is apply schema matching techniques and to combine the results of the individual prediction models. Combining the results of various prediction models has several facets:

1. We can apply different models that predict the same attribute on the new data (each of the models can be applied on subsets of the original data) and at the end combine the results through a voting procedure. Before sending the data to a model a matching algorithm might need to be applied.

2. We can apply the models in a sequence to achieve the final prediction, i.e. one model can predict an attribute that can be the input of another model etc. This situation is described in figure 4. This situation requires a more detailed domain database in which correlation between various attributes can be established and represented in the prediction models. In our use case we did not test this situation yet, although it is possible to apply it in DeVisa and it is part of the future work along with enriching the data set and building new prediction models in the same field.

VI. RELATED WORK

In the scientific literature there are numerous works that focus on the online scoring problem. Out of these the one that
resembles the most our system is the web-based system for classifier sharing and fusion named CSF/DC described in [14]. It enables the sharing of classification models, by allowing the upload and download of such models expressed in PMML in 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. One of the purposes of the system is to lead to evolution of user communities that are involved with modeling of the same domain. Furthermore, the availability of classifiers can lead to enhancement and refinement of the classification models themselves through their exposure to many users with various expertise.

Compared to the specified work, 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;
- It provides a XML-based language for expressing the Metadata Catalog;
- DeVisa deals with schema integration aspects in the scoring process and provides a 1:1 schema matching technique and an adaptive similarity measure;
- Uses a functional dependency approach in verifying if a consumer's schema can be derived from the existing schemas in DeVisa;
- DeVisa allows online composition of prediction models either during the scoring process or explicitly;
- 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.

Adding DeVisa web service components in the Kettle frontend can achieve a more complex sharing and integration functionality than in the CSF/DC system.

VII. CONCLUSIONS AND FURTHER WORK

This work presented a practical use case of an integrator of prediction services using the heart disease data set. We presented a scenario in which several prediction models expressed in PMML or XML have been built using Weka and deployed in a XML database system. Although DeVisa functionality is published through web services we have used an interface tool to represent graphically the main components responsible with the computation in DeVisa during the scoring process. The analysis of this scenario concludes that enabling integration facilities within the scoring process provides the premises of getting better results than scoring on individual prediction models. In the future the data set will be further enriched with new data instances as well as new attributes and dependencies between attributes materialized in predictive and statistical models, so that more complex use case can be obtained.

REFERENCES