What's PMML and What's New in PMML 4.0?

Rick Pechter
MicroStrategy, Inc.
Carlsbad, CA, USA
rpechter@microstrategy.com

Abstract

The Predictive Model Markup Language (PMML) data mining standard has arguably become one of the most widely adopted data mining standards in use today. Two years in the making, the latest release of PMML contains several new features and many enhancements to existing ones. This paper provides a primer on the PMML standard and its applications along with a description of the new features in PMML 4.0 which was released in May 2009.

Categories and Subject Descriptors

H.4 [**Database Management**]: Database Applications – Data Mining

General Terms

Data Mining, Business Intelligence, Predictive Analysis, Knowledge Discovery

Keywords

Data Mining, PMML, Database, Business Intelligence

1.0 Introduction

The Predictive Model Markup Language (PMML) is one of the industry's most widely supported standards for the representation and exchange of data mining models. While other standards exist, such as the Java Data Mining API [1] or Microsoft's OLEDB for Data Mining [2], PMML has features that make it a compelling solution for many organizations:

- Since PMML represents models in XML, it is platform independent, vendor agnostic and supported in a wide variety of operating environments.
- PMML is supported by many of the leading vendors in the industry.
- PMML contains a rich set of features that allows a wide range of predictive models to be represented and applied in applications.

As the lingua franca of the data mining world, PMML has become an enabler of predictive applications that span across different systems and technologies, even different constituencies. With PMML, it's possible to take, say, a credit risk predictive model created using a data mining workbench and reliably deploy that model to a business intelligence application from a different vendor. That same model could be deployed to other systems, perhaps used to generate scores in real-time within a database application or within a software-as-a-service cloud computing application. PMML can even be used to document the internal workings of the model so that government regulators can be assured that its credit risk predictions don't discriminate inappropriately (e.g., on the basis of ethnicity or religious affiliation).

PMML can meet these needs because of its breadth in model support, its depth in useful features and universality across vendors and applications. Complete details about the standard can be found on the website of the consortium that controls the PMML standard, the Data Mining Group [3]. This paper discusses that newest changes available in the latest version of the standard, PMML 4.0, released in

MAY 2009. But before discussing what's new in PMML 4.0, it's useful to describe the workings of the standard and its application.

1.1 PMML Structure

PMML is an XML-based standard and therefore its structure is described by its XML Schema Definition (XSD) [3]. Put simply, PMML has this general structure:

- Header
- Mining Build Task
- Data Dictionary
- Transformation Dictionary
- Model

These elements are common to all types of models.

The first four elements apply to all models described by the PMML document:

- The **Header** is a required element that contains about the application that generated the model (e.g., name, version) including a time stamp.
- The **Mining Build Task** is an optional element that can contain vendor specific information about how the model was built. It has no effect on model scoring and is not commonly used.
- The **Data Dictionary** is a required element that contains details about the variables, called Data Fields that participate in the model. These can be thought of representing the actual data used to develop the model, including information such the name, the type of data (e.g., string, numeric) and how it's used (e.g., is it a continuous numeric value, a categorical value, etc.).
- The **Transformation Dictionary** describes how to manipulate the Data Fields from the Data Dictionary into variables that exist within the PMML definition. These manipulations include normalizing continuous values, discretizing categorical values and applying mathematical functions. The resulting variables are called Derived Fields.

Next is the **Model** itself. Currently, the standard supports just one model element per PMML. This model element describes a specific data mining algorithm. These are the currently supported types of model types, including their model specific features:

- Association Rules: Items, item sets and rules that relate item sets.
- Clustering: Clustering Fields and Clusters
- General Regression: Parameter, Factor and Covariate Lists along with matrices that relate parameters to each other.
- Model Composition: Decision Trees and Regression models that can be used for model sequencing or model selection.
- Multiple Models (new in 4.0) : Extends to Model Composition to include segmentation
- Naïve Bayes: Counts pairing input values to output values
- Neural Network: Neurons and neural layers
- Regression: Regression table(s) that relate the input to the predicted values.
- Ruleset: Rules combined with a rule selection method
- Sequences: Items, Itemsets, Sequences and Sequence Rule that relate sequences.
- Support Vector Machines: Support vectors and coefficients along with a kernel function.
- Text: Dictionary of terms, corpus of text documents, document-term matrix along with data about the model's normalization and similarity
- Time Series (new in 4.0): Time Series information and Exponential Smoothing coefficients
- Trees: Structure of Tree Nodes

Beyond these model-specific features, each model can contain these common elements:

- Model
    - Mining Schema
    - Outputs
    - Model Statistics
    - Targets
    - Model Explanation
    - Local Transformations
    - Model Verification

These elements are common to all model types, provide the same functionality independent of the type of model and describe the details about a particular model:

- There is one **Mining Schema** required per model that describes the fields the user has to provide in order to score the model.
- **Outputs** describe the results that can be generated from the model, including the actual prediction, the confidence in that prediction as well as details about how the prediction was determined. Outputs were introduced in PMML 3.0 and are considered optional since in most cases the result to be generated is obvious (such as the result of a regression equation). But outputs allow the model to deliver any result in a reliable, precise and visible manner.
- The **Model Statistics** are often a byproduct of the data mining process and, while they tend to be description in nature, they can be useful to capture the nature of the data used to build the model and also for determining model validity.
- **Targets** allow handling of data fields and derived fields so that particular outcomes of the model are clear (such as taking "Response" and creating two targets "Response=Yes" and "Response=No".
- The **Model Explanation** element contains data that is not required for scoring the model but can help in understanding the model's quality and behavior. This optional element can include descriptive features such as lift charts, confusion matrices and ROC curves.
- The **Local Transformations** are transformations that are specific to the model element they are contained within and are only valid within the scope of that model. Contrast this with the Transformation Dictionary whose transformations are valid for all models specified in the PMML document.
- **Model Verification** is a dataset of records containing inputs and their corresponding model outputs, capturing and placed into the PMML document by the application creating the model. When deployed to another system, this dataset can be scored in the deployed environment to verify that results in the new environment match those from the environment where the model was created.

1.2 PMML Features

The structure of a PMML document is a close parallel to the structure one would expect from a predictive model. But there are other data mining features that are supported within this structure. While some of these capabilities can be considered advanced or atypical, they become essential if the standard is to have the depth of features necessary to be applicable to the widest variety of predictive applications.

1.2.1 Transformations

Transformations allow data to be converted from one meaning to another. Operations like this are common in predictive models since the underlying algorithms often depend on data having meeting certain characteristics. As described above, global transformations which apply to the entire PMML definition are contained in the Transformation Dictionary element. And transformations that are applicable only with the scope of a particular model or sub-model are contained in the Local Transformations element.

But transforms have also been included in certain model specific elements which, by their nature, involve manipulation of data values. A good example is the ClusteringField elements.

In clustering, items grouped together in the same cluster are supposed to have maximal similarity to each other over those items in other clusters. Imagine push-pins on a wall map; pins that are close to each other form a cluster. Hence, there's usually concept of "distance" in such models. While our wall map example was two dimensional, clustering models usually have many more dimensions. For the clustering algorithm to work properly, these dimensions have to be on similar scales. Consider using these four variables in a clustering model:

- Age (less than 100),
- Income (up to 100,000),
- Gender (only two valid choices), and
- Marital Status (three valid choices: Single, Married, Formerly Married).

One would want to normalize all of these to the same scale; otherwise income (with values ranging orders of magnitude larger than the other variables) will tend to dominate the result of the clustering

algorithm. Usually, a scale of 0 to 1 is used and these four data fields would be transformed in this way:

- Age_ClusteringField = Age/100
- Income_ClusteringField = Income/100000
- Gender_ClusteringField = {If Gender = Female then 1 else 0}
- Single_ClusteringField = {If MaritalStatus = Single then 1 else 0}
- Married_ClusteringField = {If MaritalStatus = Married then 1 else 0}
- Divorced_ClusteringField = {If MaritalStatus = Divorced then 1 else 0}

The transformation of Age and Income are normalizations of continuous values and PMML contains a **NormContinuous** element for this type of transformation. The transform of categorical variables like gender and marital status is simple and PMML contains a **NormDiscrete** element to do this. Note it would take three NormDiscrete operations to properly represent our Marital Status variable and the end result is a clustering model that uses six dimensions, not four, to group items into clusters. All six dimensions are on the same 0 to 1 scale so no variable will dominate the analysis.

In addition to NormContinuous and NormDiscrete, PMML contains several other transformations:

- **Discretization**: maps continuous values to discrete values.
- **MapValues**: map discrete values to discrete values.
- **Functions**: derive a value by applying a function to one or more parameters.
- **Aggregation**: summarize or collect groups of values, e.g., compute average.
- **Constant**: Represents a fixed value.

1.2.2 Data Preparation

Another common type of data manipulation is handling special cases that can occur in real world data. Predictive models depend on variables that are well-behaved and consistent. A model is only as good as the data used to build it. To avoid Garbage-in/Garbage-out, models not only need to distinguish valid values, they need to gracefully process values that are invalid. In fact, sometimes these exceptional values have significant predictive power.

Recent PMML releases have included many data preparation features and enhancements. The can generally be grouped into three areas:

- Missing Value Handling
- Invalid Value Handling
- Outlier Handling

All three areas require that such values be identified and handled accordingly.

Missing values are a common occurrence in data. Survey responders may decline to answer a question, some products may not get sold in a given place at a given time and data isn't always collected properly. While it may seem counter-intuitive, data is often missing for a reason and that reason can have predictive power. For example, declining to provide one's age on a questionnaire may say a lot about a person's tendency to use certain products or services. PMML allows models to specify what precisely is a missing value. Beyond the trivial null value, PMML allows particular values to be treated as missing. For instance, a numeric field may encode -999 to indicate that a missing record. Once a missing value is identified, the model needs to know what to do with it. PMML provide missing value handling within many key elements, including data fields, logical expressions and transformations. Missing values can be ignored, imputed with a constant or a data dependent value (mean, mode, median, etc.) or have another variable act as a surrogate.

Values that aren't missing aren't always valid. A person can't have a negative age, US standard Zip Codes are five digits, someone entered a bogus product code. PMML allows data to be specified as valid or invalid. Invalid values can cause results to be properly flagged as invalid. Or, invalid values can be treated as missing values which are subject to the missing value handling protocols.

Finally, values can have validity but lie beyond the range tolerated by the model. Good models can make predictions based on data that it has seen in the past. But models can be suspect when the applied to data that wasn't available when the model was developed. For example, while cutting a product's cost in half may double the demand, one can't assume that a ten-fold decrease will increase sales ten times if there aren't that many buyers in the market. So, models can specify when values exceed key ranges. And when these outliers are detected, they can be forced to the nearest valid extreme, or be subject to missing value handling protocols.

1.2.3 Taxonomies and Hierarchies

Data values often live within larger structures. A location can be in a city or a country, a product can be part of a category within a product line and a transaction takes place on a certain day in a certain month in a certain year. PMML allows these structures to be defined within its models.

1.2.4 Statistics

While PMML is about predictive modeling, data mining algorithms often generate useful descriptive statistics as by-products. This information is very useful to analysts and model developers since understanding the behavior of variables is often the key to building good predictive models.

PMML supports a wide range of univariate statistics, including:

- Variable Importance
- Total Count
- Missing Count
- Invalid Count
- Cardinality (new in 4.0)
- Minimum
- Maximum
- Mean
- Median
- Mode
- Inter-quartile Range
- Standard Deviation
- Histograms
- ANOVA (new in 4.0)

1.2.5 Outputs and Targets

Data mining models differ from basic functions or formulae since a model usually doesn't provide just one output. We most often thing of models as generating scores, such as if a customer is a good credit risk or if a prospect will respond to an offer. But often, it's more important to know the probability or confidence in that prediction so one can say how likely is a customer to default or respond.

In order to give model producers the flexibility to define the form and format of results and model consumers a reliable and consistent way to produce those results, PMML contains output and target elements.

Outputs define the different types of results that can be generated by the model. This includes things like names and data types, as well as rules for selecting specific result features. While not all types of models support all outputs, PMML provides for a wide variety of them, including:

- **Predicted Value**: The raw result of a model.
- **Predicted Display Value**: The result of a model formatted for human consumption.
- **Probability**: The confidence in the predicted value.
- **Residual**: For regression-type models, this is actual value minus the predicted value. For classification-type models, this is the difference in probabilities between the predicted and target values.
- **Standard Error**: The standard error of the predicted numeric value.
- **Cluster ID**: For clustering models, this identifier for the selected cluster.
- **Cluster Affinity**: For clustering models, this is the distance or similarity to a particular cluster of interest.
- **Entity ID**: Generic version of Cluster ID, indicates that the ID of the predicted cluster, tree node, neuron or rule.
- **Warning**: Provision for generating warning messages such as "Too many missing values."

PMML also supports mechanisms for defining the targets of a predictive model. Since these can come from various elements in different types of models, the Target elements provide a consistent way of referencing target across all model types.

1.2.6 Conformance

One of the main objectives for PMML is to facilitate the exchange of models from one environment to another. Exchanging predictive models between different products or environments requires a common understanding of the PMML specification. This understanding can be less than perfect, especially since PMML contains over 800 language elements. By leverages the mutual reliance producers and consumers have on the PMML:

Producers need to generate PMML that is valid and consumers need to deploy models accurately [4]. The DMG has provided several resources to assist in this area via its web site [3]:

- Sample models are available on the DMG website's sample page.
- Users can have their models validated using the DMG website.
- Information collected from users are summarized into a PMML coverage application which shows which PMML language features are in use, including in what types of models and applications.

1.2.7 Extensions

PMML also includes the ability for model producer to extend their PMML models with additional information not defined in the standard. This powerful mechanism allows application specific information to be included in the model, such as formatting specification and references to keys or object IDs. Ideally, extensions should not be required for another application to score the model. But if proprietary algorithms require non-standard features, extensions allow PMML to be used within these vendor-specific environments.

1.3 Products that support PMML

As evidenced by a recent survey which shows a three-fold increase in the use of PMML to deploy models [5], more models than ever before are being deployed using PMML. This is due in large part to the increasing number of applications that support PMML. These include IBM®, KNIME®, KXEN®, MicroStrategy®, R / Rattle®, Salford Systems®, SAS®, SPSS® and Zementis®. For a complete list, please see the DMG web site [3].

2.0 What's New in PMML 4.0?

Since PMML 1.0 was released in August 1999, the DMG has released seven major updates [3]:

- PMML 1.1: August 2000
- PMML 2.0: August 2001
- PMML 2.1: March 2003
- PMML 3.0: October 2004
- PMML 3.1: December 2005
- PMML 3.2: May 2007
- PMML 4.0: April 2009

The latest release is PMML 4.0 which includes these enhancements:

- **Association Rules**: These models now include scoring procedures that enable making specific recommendations or associations.
- **Model Explanation**: All models include this new element which contains optional (non-scoring) features that capture information about the model itself, including Lift/Gains Charts, Confusion Matrices and ROC Curves. Also included are statistics that assess the quality of predictive models, including error statistics (mean, absolute and squared), R^2 statistics as well as clustering model quality statistics (SSE and SSB). A variety of field correlations can now be represented including Pearson's correlation coefficient, Spearman's rank correlation coefficient, Kendall's τ, Contingency tables, Chi Square test, Cramer's V and Fisher's exact test.
- **General Regression**: Now allows Cox survival models using a new model sub-type called CoxRegression, including the ability to specify related time, event and hazard variables. Support also added for generating confidence intervals and new scoring algorithms with examples.
- **Mining Schema**: Now allows non-scoring weighting options that increase descriptive information about the model.
- **Model Composition and Multiple Models**: ModelComposition was deprecated in this release and replaced by the new Multiple Models type of model. Multiple models extend the previous approach by allowing more flexibility in defining and processing ensembles of models.
- **Outputs**: New outputs were added to support the various results possible with Association Rules models, allowing ranking using rule selection based on confidence, lift or support.
- **Statistics**: Univariate statistics in PMML now support weighted counts and cardinality as well as ANOVA.

- **Support Vector Machines**: This model type was enhanced to include new optional attributes for threshold and classification method.
- **Time Series**: PMML 4.0 now supports Time Series models. This first release covers Exponential Smoothing and the groundwork has been laid for additional time series methods to be added in the future.

3.0 Conclusion

Over the past decade, PMML has evolved to become the industry's leading standard for predictive model interoperability. Its rich set of features and robust implementation allow for an increasingly wider variety of applications to be built using the specification. As a result, more vendors are support PMML and, more importantly, more customers are using PMML to deploy predictive analytics beyond the silo of analysts and statisticians that build them. PMML 4.0 adds many new features that will increase its utility and applicability. With the continued support of the leading business intelligence and analytical providers, it's easy to predict that adoption and use of PMML will grow more in the future.

Acknowledgements

The author would like to thank all the colleagues in the industry who commit their valuable time and energy crafting and promoting standards for data mining. Though too numerous to name here, it has been a privilege working with and learning from the brilliant people who, in addition to their "day jobs," are able to lend their expertise to these collaborations.

References

[1] JSR 73: Data Mining API. See http://jcp.org/en/jsr/detail?id=73

[2] OLE DB for Data Mining. See http://msdn.microsoft.com/en-us/library/ms146608.aspx

[3] Data Mining Group Website. http://www.dmg.org

[4] R. Pechter. Conformance Standard for the Predictive Model Markup Language. *KDD-2006 Workshop on Data Mining Standards, Services and Platforms*. August, 2006.

[5] KDNuggets 2009 n9, item 1, *Poll: Data mining deployment grows, especially PMML and Cloud options*. May 12, 2009. See http://www.kdnuggets.com/news/2009/n09-/1i.html