

An Overview of PMML Version 3.0

Stefan Raspl

Abstract

This paper gives an overview of some of the changes in Version 3.0 of the Predictive Model Markup Language (PMML), which is expected to be released in 2004. PMML Version 3.0 adds several new models, including models for rule sets and text mining. It also adds the ability to compose certain data mining operations. For example, in PMML Version 3.0 the outputs of regression models can be used as the inputs to other models (model sequencing) and a decision tree or regression model can be used to combine the outputs of several embedded models (model selection).

1. Introduction

PMML is an application and system independent interchange format for statistical and data mining models. More precisely, the goal of PMML is to encapsulate a model in an application and system independent fashion so that two different applications (the PMML Producer and Consumer) can use it. PMML is developed by a vendor led working group, which is part of the Data Mining Group [1].

Here is a simple example: Assume that a data mining system can export PMML. Then a model developed by a statistician using the data mining system (the PMML Producer) can export the model so that a scoring system embedded in a CRM application (the PMML Consumer) can read the model and use it to score a list of prospects on the likelihood that they will respond to a mailing. The PMML Producer can be a windows application, while the PMML consumer can be a Linux application.

PMML 3.0, which is expected to be released in 2004, includes three new models and important changes to the infrastructure, including supporting the composition of data mining operations [3]

Overview of PMML

Here is a quick overview of PMML following [2].

PMML consists of the following components:

1. **Data Dictionary.** The data dictionary defines the fields that are the inputs to models and specifies the type and value range for each field.
2. **Mining Schema.** Each model contains one mining schema, which lists the fields used in the model. These fields are a subset of the fields in the Data Dictionary. The mining schema contains information that is specific to a certain model, while the data dictionary contains data definitions that do not vary with the model. For example, the Mining

Schema specifies the usage type of an attribute, which may be active (an input of the model), predicted (an output of the model), or supplementary (holding descriptive information and ignored by the model).

3. **Transformation Dictionary.** The Transformation Dictionary defines derived fields. Derived fields may be defined by normalization, which maps continuous or discrete values to numbers; by discretization, which maps continuous values to discrete values; by value mapping, which maps discrete values to discrete values; or by aggregation, which summarizes or collects groups of values, for example by computing averages.
4. **Model Statistics.** The Model Statistics component contains basic univariate statistics about the model, such as the minimum, maximum, mean, standard deviation, median, etc., of numerical attributes.
5. **Model Parameters.** PMML also specifies the actual parameters defining the statistical and data mining models. Models in PMML Version 3.0 include regression models, cluster models, trees, neural networks, Bayesian models, association rules, sequence models, support vector machines, rule sets, and text models.

Figure 1 illustrates the relationship of the Data Dictionary, Mining Schema and Transformation Dictionary. Note that inputs to models can be defined directly from the Mining Schema or indirectly as derived attributes using the Transformation Dictionary.

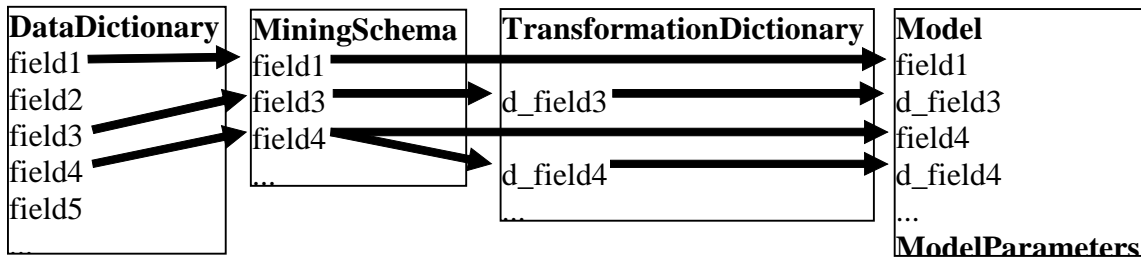


Figure 1. This figure illustrates how the inputs to a model are of two types: basic attributes which are directly specified by the Mining Schema and derived attributes, which are defined in terms of transformations from the Transformation Dictionary applied to attributes in the mining schema.

2. New Models

PMML Version 3.0 adds three new models: rule sets, support vector machines, and text models.

Rule Set: Ruleset models can be thought of as flattened decision tree models, but cover areas where decision trees are not handy or are too limited. Rulesets can be applied to new instances to

derive predictions and associated confidences (scoring). They are not meant to replace decision trees, but rather are designed to meet the requirements of a common use case.

Support Vector Machine: Over the past several years, there has been a significant amount of research on support vector machines and today support vector machine applications are becoming more common. In essence, support vector machines define hyperplanes, which try to separate the values of a given target field. The hyperplanes are defined using kernel functions. The most popular kernel types are supported: linear, polynomial, radial basis and sigmoid. Support Vector Machines can be used for both, classification and regression.

Text: Version 3.0 also adds a text model consisting of the following components:

- Dictionary of terms or text dictionary that contains the terms in the model.
- Corpus of text documents: This element identifies the actual texts that are covered by this model. Only references are given, not the actual texts.
- Document-term matrix: This element specifies what terms are used in which document.
- Text model normalization: This element defines one of several possible normalizations of the document term matrix.
- Text model similarity: This element defines the similarity used to compare two vectors representing documents.

3. New Infrastructure

Model Composition: Using simple models as transformations is one of the major additions to PMML 3.0. It now offers the possibility to combine multiple conventional models into a single new one, using individual models as building blocks. This can result in models being used in sequence, where the result of each model is the input for the next one. This approach, called model sequencing, is not only useful for building more complex models, but can also be put to good use for data preparation.

Another form of model composition is also supported: the result of a model can be used to select which model should be applied next. For example, a decision tree can now have an embedded regression model in each leaf node.

Both model sequencing and model selection can be combined to develop quite complex models.

Built-in and user defined functions. PMML 3.0 now supports functions that can be used to perform preprocessing steps on the input data. A number of predefined built-in functions for simple arithmetic operations like sum, difference, product, division, square root, logarithm, etc., for numeric input fields, as well as functions for string handling, such as functions for trimming blanks or choosing substrings.

In addition, a mechanism to define custom functions was introduced to handle cases where the built-in functions do not suffice. In this way, models can include more sophisticated preprocessing. Users can define functions that, for example, extract the number of days since the year started, out of a given date.

Model verification. The addition of a mechanism for model verification will now greatly increase the compatibility of models between different vendors' applications consuming PMML. A verification model provides a mechanism for attaching a sample data set with sample results so that a PMML consumer can verify that a model has been implemented correctly. This will make model exchange a lot more transparent for users and inform them in advance in case compatibility problems might arise.

Output fields. All models can now have output fields. The output fields describe a set of result values that can be computed by the model. In particular, the output fields specify names, types and rules for selecting specific result features. This information can be used while writing an output table. The Output section in the model specifies default names for columns in an output table that might be different from names used locally in the model. Furthermore, they describe how to compute the corresponding values.

4. Other Changes to Models

PMML Version 3.0 also contains a number of other changes, some of which we quickly describe in this section.

All models: Derived fields can be used for preprocessing inputs prior to usage in the actual model.

Association: A lift attribute has been added. Lift is a popular measure of interestingness of a rule.

Clustering: Missing value weights were added for extended missing value handling. By this, the impact of a missing value in each individual input field can be controlled.

Regression: The attributes `modelType`, `targetField` and `mean` were removed because this functionality was now provided elsewhere. For example, 'mean' was basically used for missing value handling, but that can be done in the `MiningSchema` as well. New normalization methods `probit`, `logit`, `cloglog` and `exp` were added, in order to cover popular normalization methods. A new element `PredictorTerm` has been added, containing one or more fields that are combined by multiplication. That is, 'interaction terms' are now supported as well. Finally, binary classification and logistic regression with ordinal target fields are now supported.

5. Other Changes to Infrastructure

General structure: Sparse arrays have been added. This is a method to write sparsely filled arrays in a much more compact manner. This is especially useful for models such as support vector machines or text models, which make heavy use of array structures. It makes them more readable and prevents models from becoming unnecessarily bloated.

Data dictionary: Version 3.0 adds new data types: `timeSeconds[]`, `dateDaysSince[]` and `dateTimeSecondsSince[]`. These additional types are supported in PMML because mining models often convert input values into numbers. After date and time values have been converted into

numbers they can be used easily in comparisons and other mathematical computations such as differences. For example, the date 2003-04-01 can be converted to the value 15796 of type `dateDaysSince[1960]`. These type casts are analogous to, e.g., casting an integer to a double or vice versa.

Mining schema: Version 3.0 adds attributes 'optype' and 'importance'. 'optype' overrides the corresponding value in the `DataField`. That is, a `DataField` can be used with different optypes in different models. For example, a 0/1 indicator could be used as a numeric input field in a regression model while the same field is used as a categorical field in a tree model. 'importance' states the relative importance of the field. This indicator is typically used in prediction models in order to rank fields by their predictive contribution.

Transformations: In version 3.0, one can define a replacement for missing values via the attribute 'mapMissingTo' in the transformations `NormDiscrete`, `Discretize` and `MapValues`. In the same way, default values can now be defined via 'defaultValue' to cover cases where the input is a missing value in `Discretize` and `MapValues`.

Target. In previous releases, the possible class labels of classification models were specified differently, varying between the different types of models. For example, the target categories in regression models were specified in the `RegressionTable` elements, while the `TreeModel` defines them within `Node` elements. Naive Bayes models on the other hand specify them in `TargetValueCounts`. The new PMML element `Target` for targets provides a common syntax for all models. This can also be used to provide additional information like display names for the class label or prior probabilities.

6. Summary

Perhaps the most significant changes to PMML 3.0 is the support for model composition through model sequencing and model selection. Together with the improved support for built-in functions and user-defined functions, Version 3.0 of PMML now provides a much more powerful platform for data preparation.

PMML 3.0 also adds several new model types: support vector machines, text models, and rule sets.

References

[1] The PMML Working Group is part of the Data Mining Group. See www.dmg.org.

[2] Robert Grossman, Mark Hornick, and Gregor Meyer, Data Mining Standards Initiatives, Communications of the ACM, Volume 45-8, 2002, pages 59-61

[3] PMML documentation can be found on the web site.: sourceforge.net/projects/pmml/