Planning for Data Mining Tool (PDM)
Demo Storyboard

Javier Ortiz and **Rubén Suárez** and **Tomás de la Rosa**
Susana Fernández and **Fernando Fernández** and **Daniel Borrajo**
Departamento de Informática
Universidad Carlos III de Madrid,
Avda. de la Universidad 30
28911 Leganés (Madrid), Spain

David Manzano
Ericsson Research Spain
Madrid, Spain

Abstract

This document presents a storyboard for the demonstration of the PDM Tool in ICAPS 2010. PDM is a distributed architecture for automating data mining and knowledge discovery processes based on AI Planning. The demo comprises showing the interfaces for building and executing PMML files, which hold the data mining tasks that will be plan and executed in WEKA. We first show a single execution for a dataset, and then repeat the execution to show the improvement of the system behaviour when the planning process is aided with experience of previous executions.

Introduction

For a more detailed description of PDM Tool refer to extended abstract prepared for this event. The demo has two main parts. The first one in the presentation of a standard execution of the PDM Tool. The second part is the refinement of PDM estimations using the learning component.

The first part comprises:

1. A brief explanation of the architecture, with some references for people wanting to know further details.
2. The creation of a PMML file using as input a data source in the ARFF format from the UCI Repository.
3. The execution of the PDM Tool using the PMML file previously created.
4. The presentation of results and intermediate relevant files: the PDDL problem, the WEKA Knowledge Flow, a classification model from the output and summary results.

The second part comprises:

1. The presentation of the ARFF files containing summarized results from previous PDM executions using different datasets. The executions and results gathering from datasets will be done off-line. This process takes time and it will not be possible to executed it during the demo.
2. A new execution of the PDM Tool. In this case, using a regression model generated from previous execution information. The regression model allows us to update some planning fluents in order to acquire better estimations in terms of time and accuracy metrics.
3. Visualization of new results: Difference between PDDL files, the set of top N plans that will be a better and different set from the first execution.

Demo Presentation

We will present in two slides the goal of the application (automating data mining tasks) and the architecture used in the implementation as shown in Figure 1

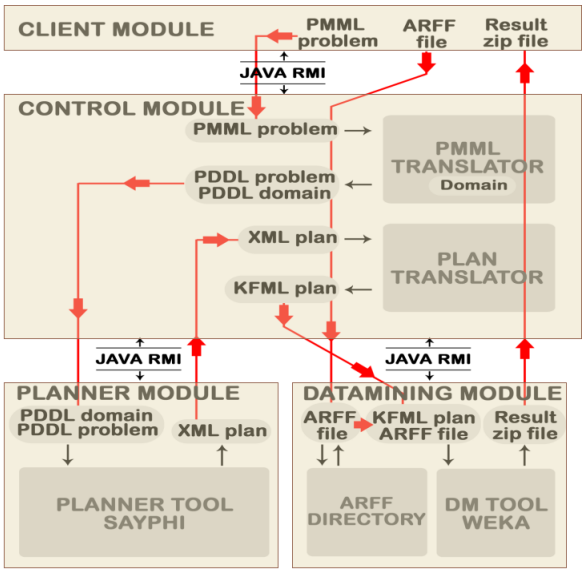

Figure 1: Screen Shot

PDM Execution

This section illustrates a execution of PDM from a user perspective. The first step is the creation of a PMML file using the PDM user interface. The PMML-Builder is the PDM module that facilitates the creation of the PMML file. Figure 2 shows the PMML-Builder GUI. From this interface we can select the input dataset, the set of data mining (DM) tasks (filters and algorithms) and define the constraints for the execution (time, minimum expected accuracy or error, etc).

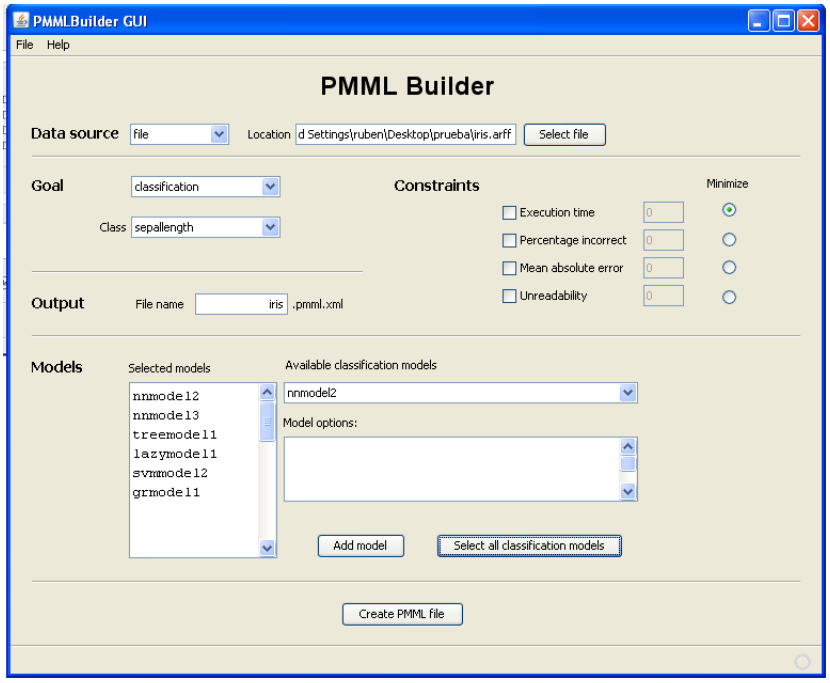

Figure 2: PMML-Builder screen-shot

Figure 3: PDM user interface screen-shot

The next step is the execution of the system from the user interface shown in Figure 3. From this interface we select the previously created PMML file, its corresponding dataset in ARFF format and the client configuration file for some extra settings.

PDM is a distributed architecture intended for running in different machines. For the demo purposes the application runs different modules (i.e., Control, Planning and Data Mining modules) in the same machine, calling each module in a separate console. While PDM is planning and executing different chains of DM tasks, we briefly explain what is happening at each console (see Figure 4).

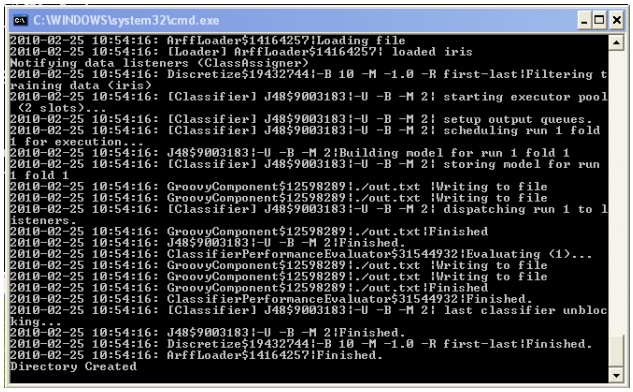

Figure 4: The Data Mining module console executing tasks in WEKA

The planner will generate different options for combining data mining tasks. Each plan is translated into a WEKA knowledge flow. The user can consult a particular flow generated from a particular plan. This knowledge flow can be also executed from outside the architecture. We present as an example a knowledge flow of one of the already executed plans, using the WEKA interface for visualizing it (see Figure 5).

The user can consult the classification models created by different knowledge flows. This is particularly interesting when the model has a readable or interpretable format, such as a decision tree presented in Figure 6.

The list of different executed plans are collected and shown in the PDM GUI (see Figure 7). We enumerate a couple of plans to show different choices of algorithm, parameters, evaluation methods and therefore, results.

Finally, results from all executions are stored in a single directory which contains, PDDL files, plans in XML format, classification models, and summarized results. We show and explain the directory contents as presented in Figure 8. Results for a particular plan are stored in a sub-directory labeled with the plan number.

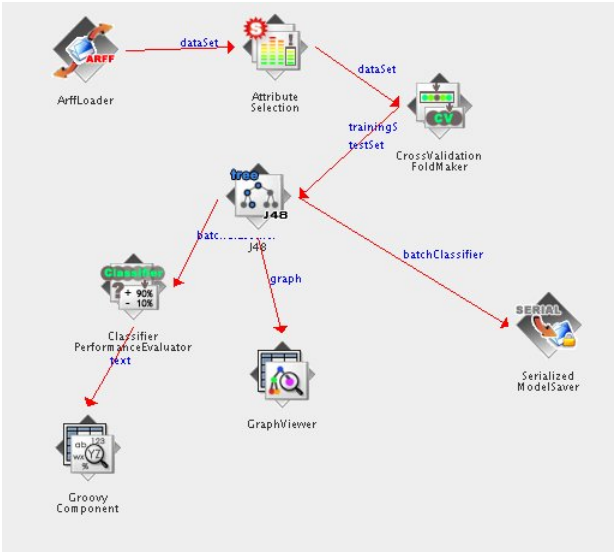

Figure 5: WEKA knowledge flow screen-shot

```
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)
```

Figure 6: An example of a decision tree generated by WEKA (Iris dataset).

PDM Improvement by Learning

This section explains an additional execution in which we want to show the improvement of the system when it has previously executed datasets. Summarized results in the ARFF format from different datasets are collected in a single directory. Using these sets of ARFF files we build a regression model that allows us to update planning fluents that affect algorithm costs in terms of time or accuracy.

The fluents are directly updated in the PDDL file used for planning DM tasks. We show which type of changes can be observed in a PDDL file such as the ones presented in Figure 9.

Results from a new execution can be observed in the PDM GUI as presented in Figure 7. In this case, we analyze the differences between plans rankings. The set of N better plans can change since now we expect to get better estimations regarding the real costs of executing DM tasks.

Acknowledgments

This work has been partially supported by the Spanish MCyT under project TIC2002-04146-C05-05, MEC project TIN2005-08945-C06-05, and CAM-UC3M project UC3M-INF-05-016.

model	function	algorithm	model-instance	parameters	evaluation	time	accuracy	mse
TREEMODEL	classification	J48	TREEMODEL1	'-U -B -M 2'	training	0.32	98	0.108
TREEMODEL	classification	J48	TREEMODEL1	'-U -B -M 2'	training	0.32	96.6667	0.1323
NEURALNET...	classification	RBFNetwork	NNMODEL3	'-B 2 -S 1 -R 1...	training	0.351	97.3333	0.1297
NEURALNET...	classification	MultilayerPerc...	NNMODEL2	'-L 0.3 -M 0.2 ...	training	0.901	98.6667	0.0935
TREEMODEL	classification	J48	TREEMODEL1	'-U -B -M 2'	split	0.233333333...	94.1176	0.1888
NEURALNET...	classification	RBFNetwork	NNMODEL3	'-B 2 -S 1 -R 1...	split	0.24	94.1176	0.2035
NEURALNET...	classification	MultilayerPerc...	NNMODEL2	'-L 0.3 -M 0.2 ...	split	0.474	94.1176	0.1973
TREEMODEL	classification	J48	TREEMODEL1	'-U -B -M 2'	cross-validation	0.0611	92	0.2087
NEURALNET...	classification	RBFNetwork	NNMODEL3	'-B 2 -S 1 -R 1...	cross-validation	0.0791	94	0.18
NEURALNET...	classification	MultilayerPerc...	NNMODEL2	'-L 0.3 -M 0.2 ...	cross-validation	0.573800000...	92	0.2159
TREEMODEL	classification	J48	TREEMODEL1	'-U -B -M 2'	training	0.32	98	0.108
NEURALNET...	classification	RBFNetwork	NNMODEL3	'-B 2 -S 1 -R 1...	training	0.41	97.3333	0.1066
NEURALNET...	classification	RBFNetwork	NNMODEL3	'-B 2 -S 1 -R 1...	training	0.361	96	0.1226
NEURALNET...	classification	MultilayerPerc...	NNMODEL2	'-L 0.3 -M 0.2 ...	training	0.58	96.6667	0.1263
TREEMODEL	classification	J48	TREEMODEL1	'-U -B -M 2'	split	0.266666666...	96.0784	0.1579
NEURALNET...	classification	RBFNetwork	NNMODEL3	'-B 2 -S 1 -R 1...	split	0.240666666...	96.0784	0.1455
NEURALNET...	classification	MultilayerPerc...	NNMODEL2	'-L 0.3 -M 0.2 ...	split	0.279999999...	96.0784	0.1342

Figure 7: Different plans (DM Tasks) executed in the PDM Tool

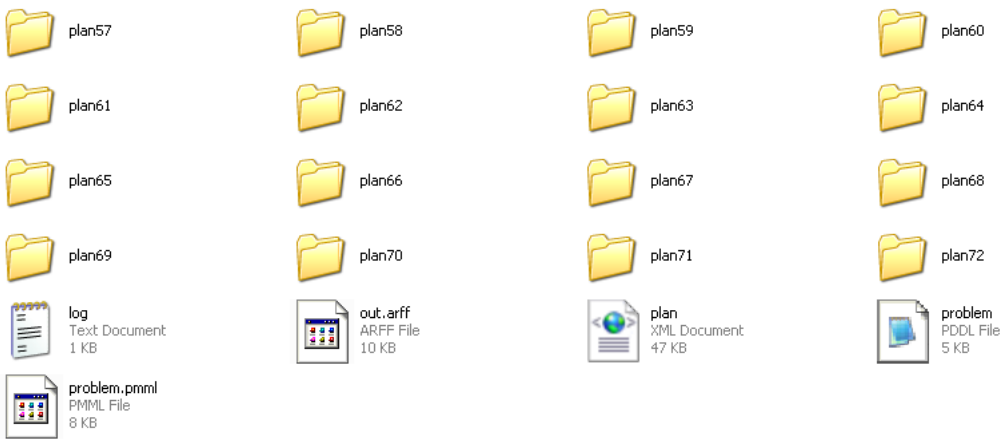

Figure 8: A directory with PDM results from dataset execution.

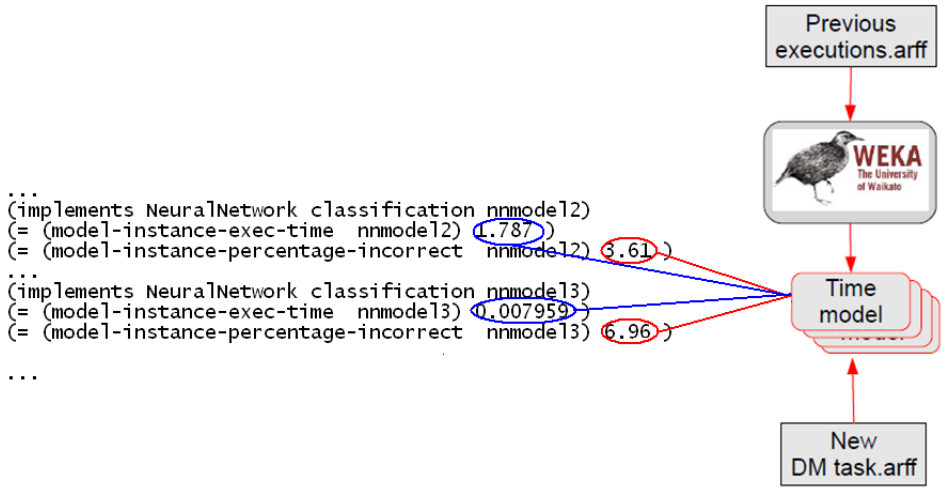

Figure 9: A PDDL fragment where fluents are updated for better time or accuracy estimations.