Towards the development of a classification service for predicting students' performance

Diego García-Saiz
Department of Mathematics, Statistics and Computation
University of Cantabria
Avda. Los Castros s/n, Santander, Spain
diego.garcia@unican.es

Marta Zorrilla
Department of Mathematics, Statistics and Computation
University of Cantabria
Avda. Los Castros s/n, Santander, Spain
marta.zorrilla@unican.es

ABSTRACT

Choosing a suitable classifier for a given data set is an important part of a data mining process. Since a large variety of classification algorithms are proposed in literature, non-experts, as teachers, do not know which method should be used in order to achieve a good pattern. Hence, a recommender service which guide on the process or automatize it is welcome. In this paper, we rely on meta-learning in order to predict the best algorithm for a data set given. More specifically, our work analyses what meta-features are more suitable for the problem of predicting student performance and also evaluates the viability of the recommender.

1. INTRODUCTION

One of the most important tasks in the process of knowledge discovery (KDD) is the selection of the algorithm that gets the best performance to solve a given problem. An approach based on meta-learning, able to automatically provide guidance on the best alternative from a set of meta-data, can be followed to achieve this goal.

We consider that this approach is suitable for educational context in which most teachers are not experts in data mining, but they do need to have objective information that allows them to enhance the teaching-learning process. Our ultimate goal is to automate the whole KDD process so that teachers should only be concerned to define the data set and a software service, supported by a recommender of algorithms, which generates the most accurate classification model based on the more relevant features of the data set.

In our work, we understand *meta-learning* as the automatic process of generating knowledge that relates the performance of machine learning algorithms to the characteristics of the data sets. We propose a number of features and discard others for their use in educational field. In our case study we generated 81 data sets from 2 virtual courses taught in the University of Cantabria and build over 700 classifiers using twelve different classification algorithms. Then we created three meta-data sets with the intrinsic characteristics extracted from each original data sets and define the algorithm with higher accuracy as class attribute. One of these data sets was used to generate our recommender.

There are different approaches about what features can be used as meta-data. In most cases measurable properties of data sets and algorithms are chosen. For instance, some authors [3] utilize general, statistical and information-theoretical measuresextracted from data sets whereas others as use landmarkers as in [4].

This paper is not intended to design a database to store data mining processes as there is already one available [1], but its main aim is to assess the feasibility of our proposal and propose a set of measurable features on educational data sets which can help us to choose automatically the classification algorithm with certain reliability.

We must also mention several research projects have targeted meta-learning in recent years, as e-LICO project [2].

2. EXPERIMENTATION

In our experiments, we used data from 2 virtual courses: a multimedia course taught during three academic years (2008-2010) hosted in Blackboard and a programming course taught in 2009 hosted in Moodle. All data sets gather the activity performed by learners in each course with their corresponding numeric mark.

In order to have enough data sets for our experimentation, we generated 81 data sets from them. First we created 3 data sets with data from multimedia course establishing the class attribute with values pass or fail, and another one as the union of these three. The same process was carried out with the programming course. Next, we generated 4 discretized data sets from the previous bi-class data sets using PKIDiscretize from Weka, and 4 data sets more but these partially discretized. Besides, we created two data sets with 4 classes (fail, pass, good, excellent) and one with 5 classes (drop-out, fail, pass, good, and excellent).

Next, we generated 60 data sets by adding to all original data sets a 10, 20, 30 and 40% of missing values. And finally, we created 4 data sets more by applying SMOTE algorithm on 2 of our original data sets with the following proportion of balancing class: 80-20%, 85-15%, 70-10% and 90-10%.

Models generation was performed by applying 12 classification algorithms on 63 data sets. The algorithms chosen were: NaiveBayes, BayesNet, NearestNeighbours, AdaBoost, OneR, Jrip, Ridor, NNge, J48, RandomForest, OneR and SimpleCart. We selected as features the number of attributes and instances in the data set, the number of categorical and numerical attributes, the type of data in the data set (numeric, nominal or mixed) and the number of classes. Regarding quality, we chosen completeness (percentage of null values) and finally, we used class entropy in order to establish if the class was balanced or not.

Next, we generated three data sets with the mentioned meta-features as attributes and the algorithm which achieved better performance as class value. The "md1" meta-data set contains an instance per data set, we only considered the best algorithm. If two or more algorithms achieved the same accuracy, all of them were included (84 instances). The "md2" meta-data set follows the same criteria as "md1" but in this case, we only used the models obtained by J48, JRIP, NaïveBayes and BayesNet. Finally, "md3" meta-data set contains as many instances as models achieved an accuracy whose statistical difference assessed by t-test was lower than 5%, with OneR as base algorithm.

In order to evaluate what meta-features are more useful to build the recommender, we applied a filtering algorithm offered by Weka, ClassifierSubSetEval. This algorithm returns how important the features are to perform a prediction task. It requires a base classifier as parameter, so it focuses on what attributes are more useful for a single classifier. Since we are focused on offering a recommender to non-experts in data mining, the base classifier used by ClassifierSubSetEval should be a prediction model easy to understand. We chosen two algorithms, J48 and NaïveBayes with the aim of testing two different approaches. The results when ClassifierSubSetEval was run are shown in Table 1, using as search algorithm LinearForwardSelection . The importance of the features is measured by means of an scale from 0 (useless) to 10 (very useful).

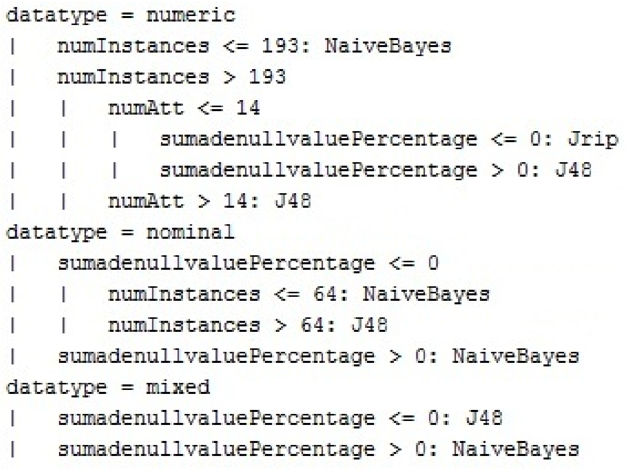
Analysing the results we can say that the degree of class imbalance, the number of instances and the completeness have a high significance. The other feature that seems to be important is the number of attributes. The rest of features are significant depending on the classification algorithm. For instance, the type of data is quite important for J48, but meaningless for NaïveBayes.

Table 1: Recommended features by ClassifierSubSetEval

	md1		md2		md3	
	NB	J48	NB	J48	NB	J48
#N Instances	9	5	8	10	8	8
#N Attributes	7	9	9	2	6	8
#N Numeric att.	1	3	0	4	5	5
#N Nominal att.	9	3	0	0	1	6
Completeness	10	10	8	10	7	9
#Type att.	7	3	6	4	5	4
#N Classes.	6	2	1	9	3	5
Is_balanced?	9	8	9	7	3	3

Next, we built models using the three meta-data sets generated for this experimentation. The more accurate model was achieved with "md2". This is due to the class attribute has 4 possible values in a data set with 80 instances, whereas, in "md1" and "md3", it has 12 different values with a slightly higher number of instances. Moreover, most models built from "md1" and "md3" were over-fitted.

Figure 1 depicts a model built with "md2" using J48. As expected, according to previous features analysis, it uses the type of data, the number of instances, the number of attributes and the completeness to build the model. From 81 data sets, 63 were used for building our recommender and the rest for testing. It achieved an accuracy of 68.75% which is a little lower than those obtained by classifiers built for this experimentation (range from 55% to 76%).



```
datatype = numeric
|   numInstances <= 193: NaiveBayes
|   numInstances > 193
|   |   numAtt <= 14
|   |   |   sumadenullvaluePercentage <= 0: Jrip
|   |   |   sumadenullvaluePercentage > 0: J48
|   |   numAtt > 14: J48
datatype = nominal
|   sumadenullvaluePercentage <= 0
|   |   numInstances <= 64: NaiveBayes
|   |   numInstances > 64: J48
|   sumadenullvaluePercentage > 0: NaiveBayes
datatype = mixed
|   sumadenullvaluePercentage <= 0: J48
|   sumadenullvaluePercentage > 0: NaiveBayes
```

Figure 1: J48 Recommender

3. CONCLUSIONS

We have analysed which features are more suitable for describing educational data sets aimed at predicting student performance. We have also shown that construction of a recommender system following a meta-learning approach is feasible.

In a near future we will work with other kind of meta-characteristics such as the mentioned landmarkers and setting parameters of the algorithms. Of course, other quality measures of model, in addition to accuracy, will be considered.

4. REFERENCES

[1] H. Blockeel and J. Vanschoren. Experiment databases: Towards an improved experimental methodology in machine learning. In J. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenic, and A. Skowron, editors, *Knowledge Discovery in Databases: PKDD 2007*, volume 4702 of *Lecture Notes in Computer Science*, pages 6–17. Springer Berlin / Heidelberg, 2007.

[2] M. Hilario. e-lico annual report 2010. Technical report, Université de Geneve, 2010.

[3] D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, editors. *Machine learning, neural and statistical classification.* Ellis Horwood, Upper Saddle River, NJ, USA, 1994.

[4] B. Pfahringer, H. Bensusan, and C. Giraud-carrier. Meta-learning by landmarking various learning algorithms. In *in Proceedings of the 17th International Conference on Machine Learning*, pages 743–750. Morgan Kaufmann, 2000.