Metalearning-based recommenders: towards automatic classification algorithm selection

Diego García-Saiz and Marta Zorrilla

Department of Computer Science and Electronics, University of Cantabria
Avenida de los Castros s/n, 39005, Santander, Spain
{diego.garcias,marta.zorrilla}@unican.es

Abstract. Selecting an appropriate mining algorithm for a certain dataset in hand is still today one of the open challenges in the data mining field. Solving this issue is highly complicated but extremely important in order to wrap mining services to be consumed by non-expert users such as teachers involved in virtual teaching. This work relies on meta-learning for the building of an algorithm recommender, describing the process to be followed for its implementation, and analysing the behaviour of different sets of meta-features which could be used for our proposal. Finally, the paper shows the feasibility of our prototype built for supporting the process of obtaining student performance prediction models.

Keywords: Meta-learning, Classification, Non-expert miners, Educational Data Mining, Student performance

1 Introduction

Nowadays we are involved in the so-named datification process [14] which allows citizens and organizations to share their data and take advantage from its analysis. Most users would like to be able to manage and gain insights from this data to make more accurate decisions in their daily work, but regrettably they lack the data mining skills needed to deal with this issue.

Most data mining tools are addressed to mining specialists and consequently non-experts users cannot benefit from their use. This is the case of teachers involved in virtual education who develop their teaching activity supported by an e-learning platform such as Moodle or Blackboard. These professionals know that all the interaction held between learners as well as the activity performed by students is collected in repositories (generally relational databases) which suitably managed could help them to a large extent to uncover how the teaching-learning process takes place.

E-learning Web Miner (ElWM) [4] is a tool developed with the aim of helping teachers involved in virtual education to analyse and discover mining models that allow them to gain insight into the teaching-learning process. One of its main advantages is that it has been designed to be used by non-expert users in data mining, that means, the tool wraps all the mining process. It can directly be used by uploading a file or tuned for reading data from the e-learning platform repository. This web application was designed so as to answer a certain set of frequent questions that most teachers are interested in knowing. For instance, discovering student profiles, visualising patterns about the activity performed by learners or getting performance and/or drop-out prediction models.

For answering each question, a data mining algorithm was fixed before doing a wide experimentation. Now, we are interested in replacing this algorithm with one that is more suitable for the dataset under evaluation. We therefore search a mechanism which allows us to characterise the algorithms from the meta-features extracted from the datasets and models built with them.

This paper thus describes the process to be followed for the implementation of an algorithm recommender, and analyses the behaviour of different sets of meta-features which could be used for our proposal. Finally, the paper shows the feasibility of our approach assessing the performance of a recommender built for supporting the process of obtaining student performance prediction models.

This paper is organised as follows: Section 2 summarises the related works in both research areas involved, meta-learning and student performance prediction. Section 3 briefly explains the method followed for developing our proposal. Section 4 describes the experiment carried out so as to compare the usefulness of each set of meta-features and discusses the outcomes. Section 5 describes how our prototype was implemented and shows how well this works. Finally, conclusions and future works are outlined in Section 6.

2 State of art

The section is divided in two subsections: firstly we provide background on meta-learning and its application for algorithm selection, secondly we outline the most significant contributions in student performance prediction topic.

2.1 Meta-learning: an overview

Meta-learning is a subfield of machine learning that aims at applying learning algorithms on meta-features extracted from machine learning experiments in order to better understand how these algorithms can become flexible in solving different kinds of learning problems, hence to improve the performance of existing learning algorithms [21] or to assist the user to determine the most suitable learning algorithm(s) for a problem at hand [8], among others.

Meta-learning thus aims at learning the relationship between the meta-features extracted from the data sets and the algorithms performance applied on them. Therefore, the algorithm selection process based on meta-learning consists of two main stages: a training phase and a prediction phase. In the training stage, data sets are first characterized by a set of measurable characteristics and next, a set of algorithms are executed on these data sets and their performance evaluations such as accuracy, f-measure, error rate, etc. are linked to the characteristics of the involved data set. Later, a learning algorithm is trained on the collected meta-data, which will yield a model which will be used to predict which the best algorithm to be applied on a new data set is. Different approaches for building the recommender have been proposed, mainly based on classification [12, 17, 22] and regression [9, 1].

Regarding the kind of meta-features that these systems generally use, these can be classified in:

- Simple or general features, such as the number of attributes, the number of instances, the type of attributes (numerical, categorical or mixed), the number of values of the target attribute and dimensionality of the data set, i.e., the ratio between the number of attributes and the number of instances.
- Statistical features, like skew, kurtosis among others which measure the distribution of attributes and their correlation [21, 20].
- Information theoretic features used for characterising data sets containing categorical attributes such as class entropy or noise to signal ratio [6].
- Model-based meta-features, which collect the structural shape and size of a decision tree trained on the data sets [15].
- Landmarkers, which are meta-features calculated as the performance measures achieved by using simple classifiers [16].
- Contextual features, i.e., characteristics related to data set domain [22].

Recently, some works [2, 22] have used a set of data complexity metrics provided by DCOL tool [7] which measures characteristics of the data independently of the learning method as meta-features. These metrics have been also used recently by [11] to obtain the domains of competence of a classifier, which allows to predict if any data set will be suitable for such learning method or not.

2.2 Predicting the student performance

Data mining techniques have been widely used since the last decade in the educational arena, being the prediction of student performance one of the most frequently studied problems. As stated in the "Handbook of Educational Data Mining" [19], there is not a single classification algorithm which performs better than the others in all scenarios, leading to many researchers to study which one is the best for this particular problem.

For instance, in [10], the authors applied seven different classification algorithms on a set of datasets from e-learning courses so as to predict the student performance, concluding that Bayesian techniques were the most suitable for educational datasets, which generally have a quite low number of instances. In this work [18], twenty one different techniques were used with the same purpose, and the authors concluded that no single classifier performed better in all cases. Dekker et al. [3] presented a case study to predict student dropout and demonstrated the effectiveness of several classification techniques, turning out that rather simple and intuitive classifiers such as decision trees give a successful outcome with accuracies that range from 75 to 80%. Finally, this paper [5] stated that Naïve Bayes outperforms on datasets with a low number of instances but when this number grows, other classifiers such us C4.5 or Bayesian Networks, lead to more accurate prediction models.

As far as we know, there are very few works in which meta-learning has been applied for our purpose. . One of these is the work published by Molina et al.[13] which uses meta-learning with the aim of setting two parameters of the J48 algorithm in order to increase the accuracy of the prediction model, and the other one, [22], which builds an algorithm recommender using only the classifier that outperformed in the training phase of the meta-learning process.

3 Recommender design

Next, we explain the schema that we suggest in order to build an algorithm recommender. For the sake of a better understanding, we graphically depict a modular schema of our proposal (see Figure 1).

Our goal is to implement a software artifact that offers a ranking of classifiers according to their expected accuracy for every dataset under analysis, in such a way that the service that wraps it can show this ranking and allow end-user (novice data miner) to choose the algorithm or automatically build the model with the algorithm on the top (for non-expert miners).

As can be observed in the image, the recommender system is built from a set of datasets (D1,D2,..., Dn), in our case, with student activity and performance data from e-learning courses. Then, a meta-feature extraction process is run. This, in turn, comprises several tasks, one, at least, for each group of meta-features mentioned in Sect. 2.

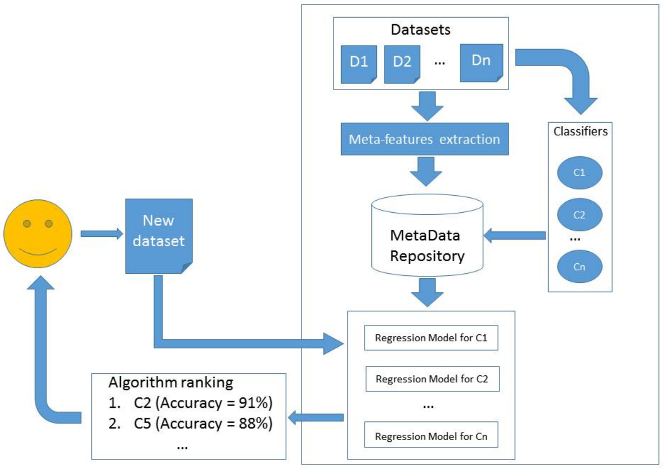

Fig. 1. Recommender modular schema

In parallel, a set of classification algorithms are executed on these datasets and their performance measures are stored in the metadata repository along with the extracted meta-features. For every classification algorithm (C1,C2,..., Cn), a regression model is built, taking the meta-features of the datasets processed as predictor attributes and the accuracy obtained by the classifier as predicted value. When a new dataset is loaded, all regressors are run and a list of algorithms ranked according to their predicted accuracy is shown.

4 Experimental meta-features comparison

One of the main concerns when we deal with the building of this recommender is to know which meta-features are the most effective for our purpose. Therefore, we carried out the following experimentation.

We selected thirty different e-learning and blended courses hosted in a Moodle platform and extracted the activity performed by learners and their outcome (passed or failed the course). The activity is measured by means of several metrics such as the number of tests carried out, the time spent, the number of messages written in the forum and so on. All attributes are numeric except the class.

Regarding meta-features, we extracted the following ones: the number of attributes, the number of instances and dimensionality from the set of simple meta-features; minimum, maximum and average value of the skewness and kurtosis of all attributes of the dataset calculated by means of the MATH3-apache Java library as representation of statistical measures; as landmakers we used the accuracy achieved by the following weak classifiers: LinearDiscriminant (LD), BestNode with gain-ratio criterion (BN), RandomNode (RN), NaïveBayes (NB) and 1-NN, all available in Weka or RapidMiner, and finally, the fourteen features offered by DCoL software. Due to the fact that our datasets have no nominal attributes, no information-theory measures were used. The model-based measures neither had been included since these kind of meta-features are highly dependent of the type of the classification algorithm used to generate them, which is most proposal is a decision tree. Thus, it is needed a complementary study to establish the effect of these meta-features in the prediction of the performance of classifiers of different paradigms, like bayesians or rule-based, which may be approached in a future.

Next, we apply eleven classifiers on these thirty datasets using their default setting: C4.5 (J48 version from Weka), RandomForest, RIPPER (JRip version from Weka), NNge, Ridor, BayesNet, SimpleCart, LogisticRegression, AdaBoost and Bagging with DecisionStump as base classifier. These were selected because each one follows a different learning paradigm and the models that generate are easy to interpret for a non-expert user. The performance measures were evaluated by a leave-one-out cross-validation, due to mainly the reduced number of datasets available.

Then, we generated eleven meta-datasets, one for each classifier. Each dataset contained the meta-features of the training datasets along with the accuracy achieved by this classifier. With the aim of studying the behaviour of each group of meta-features, we built different linear regression models by using different combinations of meta-features:

1. All the meta features available.
2. Only the meta-features belonging to each group (simple, statistical, complexity or landmarkers) separately.
3. Only the most relevant meta-features chosen by a feature-selection algorithm. For this purpose, we used the ClassifierSubSet algorithm, offered by Weka, with the BestFirst algorithm as search method and linear regression as base classifier. The leave-one-out method was used for its evaluation. The thresholds used, to choose features according to their relevance, range from 10% to 95% .

All the regression models were generated with Linear Regression (Weka) using a leave-one-out evaluation strategy. For a more exhaustive evaluation, the root mean square error (RMSE) is used to assess the confidence interval of the predicted accuracy.

The results achieved in this experiment are shown in Table 1. Each column gathers the RMSE obtained by each one of the generated regression models by using all the meta-features ("all"), only complexity ones ("com"), only simple ones ("sim"), only landmarkers ("ldm") and only statistical features ("sta"). The columns marked with "$*$" show the RMSE obtained after applying the feature selection algorithm with a threshold of 95% on the same meta-datasets).

Table 1. RMSE of predicted accuracies of the regression models built for different meta-features groups with and without feature selection

	all .	all* .	com .	com* .	ldm .	ldm* .	sim .	sim* .	sta .	sta* .
AdaBoost	0.143	0.069	0.164	0.093	0.045	0.043	0.128	0.117	0.136	0.115
Bagging	0.12	0.074	0.171	0.073	0.049	0.048	0.118	0.099	0.102	0.097
BayesNet	0.217	0.064	0.195	0.115	0.113	0.094	0.203	0.17	0.197	0.18
J48	0.177	0.069	0.111	0.069	0.051	0.049	0.124	0.101	0.103	0.096
Jrip	0.195	0.042	0.132	0.074	0.079	0.073	0.165	0.126	0.141	0.124
LogisticRegression	0.292	0.05	0.106	0.05	0.081	0.062	0.154	0.147	0.149	0.133
NNge	0.213	0.037	0.104	0.049	0.043	0.041	0.145	0.129	0.128	0.115
OneR	0.335	0.051	0.108	0.072	0.058	0.051	0.143	0.114	0.131	0.11
RandomForest	0.195	0.047	0.051	0.045	0.053	0.047	0.127	0.109	0.095	0.089
Ridor	0.205	0.048	0.098	0.074	0.052	0.052	0.14	0.119	0.12	0.113
SimpleCart	0.259	0.091	0.197	0.096	0.14	0.125	0.216	0.164	0.173	0.167

Analysing the table, the first conclusion that is drawn is that using all measures does not lead to a good model since the RMSE of predicted accuracies of the regression models built is high for all classifiers. The same happens when simple measures, statistical measures and complexity measures are used separately, with a RMSE higher than 0.1 and even 0.2 in some cases. Only the complexity measures and the landmarks seem to generate better regression models, with a low RMSE for most, but not all, of the classification algorithms. When the feature selection process is applied, the outcomes are significatively better. The improvement can be observed in most models, even when only a group of meta-features is used.

Another statement is that using simple or statistical meta-features alone, with or without feature selection, gives poor results. However, the complexity measures perform better when feature selection is carried out, obtaining models with a RMSE lower than 0.05. Moreover, the regression models based on landmarkers, even having a good RMSE without applying feature selection, their performance improve.

Nevertheless, the most significant improvement is achieved when all measures are used and feature selection is applied. In this case, 8 out of 11 classifiers achieved the lowest RMSE being this lower than 0.1. Only the landmarks-based models built for AdaBoost, Bagging and J48 are significantly better than the rest, with a RMSE of 0.04, 0.05 and 0.05 respectively.

To sum up, this experimentation concludes that landmarkers and complexity measures have a good behaviour as predictors, however using as many as possible meta-features and applying feature selection previously to build the regressor leads to the best result.

5 Evaluation of our proposal

As our end goal is to wrap an algorithm recommender in our ElWM tool, next we assess how well this works. For this purpose, we have selected the meta-regression model with lowest RMSE for each algorithm. This means, the landmarkers-based model with feature selection for predicting the accuracy of AdaBoost, Bagging and J48 and the regression models built with all meta-features and application of feature selection for the rest of the classifiers.

Then, in order to ensure the feasibility of our approach, we test our recommender with the same thirty datasets, following a leave-one-out process. Each dataset is evaluated with the previously mentioned regression models generated by using only the remaining twenty nine datasets, in such a way that the dataset under evaluation is considered as a new one to the framework so as to get a recommendation. Table 2 shows how many times the best classification algorithm is recommended among the thirty datasets, and how many times the classification algorithm is in the first quartile of the ranking, ie, one of the three first positions (we should remember that we worked with eleven algorithms). As can be observed, the best classification algorithm is recommended a 23.33% of times and one of the three top algorithms in a 56.67% (17/30) of times. Moreover, the 83.33% of times the chosen algorithm is in the the 50th percentile. Only a 6.67% (2/30) of times, a classifier ranked in the 10 or 11 position (four quartile) is recommended.

Next, we show the ranking offered by our recommender when one of these datasets was loaded (see Table 3). Column "Pred. Acc." refers to predicted accuracy by our recommender whereas "Real Acc." means real accuracy achieved by the classifier when directly applied on it. As can be observed, the first ranked classifier, RandomForest, is the second classifier with the best real accuracy,

Table 2. Number # and % of times that the recommended classifier is the best. Additionally the result by quartiles is shown

	# times (over 30)	% times
best classifier	7	23.33
first quartile	17	56.67
second quartile	8	26.67
third quartile	3	10.00
fourth quartile	2	6.67

88.083, getting a predicted accuracy of 88.237%. Moreover, Bagging, with the same real accuracy, is the second ranked classifier.

Table 3. Ranking based on the predicted accuracy for one of the datasets

Classifier	Rank.	Pred. Acc.	Real Acc.	Diference
RandomForest	1	88.237	88.083	0.154
Bagging	2	87.054	88.083	-1.029
SimpleCart	3	86.927	87.565	-0.638
AdaBoost	4	86.768	86.528	0.240
J48	5	86.596	88.083	-1.487
Jrip	6	86.565	88.601	-2.036
NNge	7	86.562	86.568	0.034
BayesNet	8	86.058	88.063	-2.025
OneR	9	86.034	84.456	1.578
LogisticRegression	10	82.464	82.902	-0.438
Ridor	11	80.824	85.492	-4.668

One of the issues that affects to this ranking is the high value of the error obtained in the prediction of the accuracy of Jrip, -2.036. Since the difference of accuracy for most of the classifiers is lower than 2, and even lower than 1, the fact that the best classification algorithm has a high error in comparison to the others has affected the ranking to the point that, instead of JRip, the selected classifier has been the second better, RandomForest. On the other hand, the difference of real accuracy among the 8 first ranked classifiers is not so high, making more difficult to our system to predict the best classifier. Despite that fact, the selected classifier, RandomForest, has not only the second best real accuracy, but its difference with respect to the real accuracy of the best classifier, Jrip, is very low, 0.518%. Moreover, the three worst classifiers in terms of real accuracy have, at the same time, the worst predicted accuracy.

At this point we can conclude that, by using the algorithm automatic selection system, ElWM will offer, mostly times, more accurate models than using always the prefixed algorithm, in our case J48.

6 Conclusions

One of the challenges that is still open in the prediction arena is to choose the best algorithm for a certain dataset. Automatising a process for solving this issue is currently needed in order to build tools which enable non-expert users in data mining, to take advantages from their data.

This paper provides a proposal addressed to fix this issue. On the one hand, we explain the process for building an algorithm recommender based on meta-learning. Likewise, we enumerate the meta-features which can be used and compare the performance that these achieve in a case study. We concluded that landmarkers and complexity measures have a good behaviour as predictors, but it is much better using as many as possible meta-features and, instead of selecting all, applying a feature selection process previously to build the regressor. Finally, we assess the feasibility of our proposal. As a result, our approach predicts one of the best algorithms to be applied.

Nevertheless, a more wide experimentation must be carried out in order to determine the best setting for each problem domain. This will be accomplished in five-folds: i) using a larger number datasets from different fields; ii) assessing the performance of information theoretic and model-based features as meta-features and, iii)building recommenders by applying more complex regression techniques; iv) adding more classifiers to the study; and v) weighting the meta-features by measuring its relevance (weight) in the regression model with a feature selection technique.

7 Acknowledgments

This work has been partially funded by the Spanish Government under grant TIN2014-56158-C4-2-P (M2C2).

References

1. Automatic classifier selection for non-experts. Pattern Analysis and Applications 17(1) (2014)
2. Cavalcanti, G., Ren, T., Vale, B.: Data complexity measures and nearest neighbor classifiers: A practical analysis for meta-learning. In: Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on. vol. 1, pp. 1065–1069 (Nov 2012)
3. Dekker, G.W., Pechenizkiy, M., Vleeshouwers, J.M.: Predicting students drop out: A case study. International Working Group on Educational Data Mining (2009)
4. Diego García-Saiz, Camilo Palazuelos, M.Z.: Educational Data Mining, Studies in Computational Intelligence, Vol. 524, vol. 524, chap. Data Mining and Social Network Analysis in the Educational Field: An Application for Non-expert Users. Springer (2014)
5. García-Saiz, D., Zorrilla, M.E.: Comparing classification methods for predicting distance students' performance. In: Diethe, T., Balcázar, J.L., Shawe-Taylor, J., Tirnăucă, C. (eds.) WAPA. pp. 26–32 (2011)

6. Hilario, M., Kalousis, A.: Building algorithm profiles for prior model selection in knowledge discovery systems. Engineering Intelligent Systems 8, 956 – 961 (2002)
7. Ho, T.K.: Geometrical complexity of classification problems. CoRR cs.CV/0402020 (2004)
8. Kalousis, A., Hilario, M.: Model selection via meta-learning: a comparative study. In: Proc. 12th IEEE International Conference on Tools with Artificial Intelligence. pp. 406–413 (2000)
9. Köpf, C., Taylor, C., Keller, J.: Meta-analysis: From data characterisation for meta-learning to meta-regression. In: Proceedings of the PKDD-00 Workshop on Data Mining, Decision Support,Meta-Learning and ILP (2000)
10. Kotsiantis, S., Pierrakeas, C., Pintelas, P.: Predicting students' performance in distance learning using machine learning techniques. Applied Artificial Intelligence 18(5), 411–426 (2004), http://dx.doi.org/10.1080/08839510490442058
11. Luengo, J., Herrera, F.: An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. Knowl. Inf. Syst. 42(1), 147–180 (2015), http://dx.doi.org/10.1007/s10115-013-0700-4
12. Molina, M.M., Luna, J.M., Romero, C., Ventura, S.: Meta-learning approach for automatic parameter tuning: A case study with educational datasets. In: Proc. 5th International Conference on Educational Data Mining. pp. 180–183 (2012)
13. Molina, M., Luna, J., Romero, C., Ventura, S.: Meta-learning approach for automatic parameter tuning: A case study with educational datasets. International Educational Data Mining Society (2012)
14. Newell, S., Marabelli, M.: Strategic opportunities (and challenges) of algorithmic decision-making: A call for action on the long-term societal effects of datification. The Journal of Strategic Information Systems (2015)
15. Peng, Y., Flach, P., Soares, C., Brazdil, P.: Improved dataset characterisation for meta-learning. In: Lange, S., Satoh, K., Smith, C. (eds.) Discovery Science, LNCS, vol. 2534, pp. 193–208. Springer Berlin / Heidelberg (2002)
16. Pfahringer, B., Bensusan, H., Giraud-carrier, C.: Meta-learning by landmarking various learning algorithms. In: in Proceedings of the 17th International Conference on Machine Learning. pp. 743–750. Morgan Kaufmann (2000)
17. Reif, M., Leveringhaus, A., Shafait, F., Dengel, A.: Predicting classifier combinations. In: Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods. INSTICC, SciTePress (2013)
18. Romero, C., Espejo, P.G., Zafra, A., Romero, J.R., Ventura, S.: Web usage mining for predicting final marks of students that use moodle courses. Computer Applications in Engineering Education 21(1)
19. Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.: Handbook of educational data mining. CRC Press (2010)
20. Segrera, S., Pinho, J., Moreno, M.N.: Information-theoretic measures for meta-learning. In: Proc. 3rd international workshop on Hybrid Artificial Intelligence Systems. pp. 458–465. HAIS '08, Springer-Verlag, Berlin, Heidelberg (2008)
21. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial Intelligence Review 18, 77–95 (2002)
22. Zorrilla, M.E., García-Saiz, D.: Meta-learning: Can it be suitable to automatise the KDD process for the educational domain? In: Kryszkiewicz, M., Cornelis, C., Ciucci, D., Medina-Moreno, J., Motoda, H., Ras, Z.W. (eds.) Rough Sets and Intelligent Systems Paradigms - Second International Conference, RSEISP 2014. Lecture Notes in Computer Science, vol. 8537, pp. 285–292. Springer (2014), http://dx.doi.org/10.1007/978-3-319-08729-0