Meta-Learning Based Framework for Helping Non-expert Miners to Choice a Suitable Classification Algorithm: An Application for the Educational Field

Marta Zorrilla(✉) and Diego García-Saiz

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

Abstract. One of the most challenging tasks in the knowledge discovery process is the selection of the best classification algorithm for a data set at hand. Thus, tools which help practitioners to choose the best classifier along with its parameter setting are highly demanded. These will not only be useful for trainees but also for the automation of the data mining process. Our approach is based on meta-learning, which relies on the application of learning algorithms on meta-data extracted from data mining experiments in order to better understand how these algorithms can become flexible in solving different kinds of learning problems. This paper presents a framework which allows novices to create and feed their own experiment database and later, analyse and select the best technique for their target data set. As case study, we evaluate different sets of meta-features on educational data sets and discuss which ones are more suitable for predicting student performance.

Keywords: Meta-learning · Regression · Student performance

1 Introduction

Currently, the possibility of automatising the Knowledge Discovery Process (KDD) is still an open problem. As it is well-known, the KDD process [4] consists in several phases (preprocessing, modeling, mining and testing) and each one, in turn, includes a large number of tasks which should be performed. As every complex problem, a way to deal with it is to follow a divide and rule strategy. That is why this paper is focused on the mining phase, that means, the step in which the data miners have to choose the best algorithm for their data set at hand.

Rice [15] was the one who first formulated this issue and since then different approaches have been proposed, for instance: a) a traditional approach based on a costly trial-and-error procedure; b) an approach based on meta-learning, able to automatically provide guidance on the best alternative from a set of meta-features; or c) the use of ensemble methods to obtain better predictive performance.

M. Núñez et al. (Eds.): ICCCI 2015, Part II, LNCS 9330, pp. 431–440, 2015.
DOI: 10.1007/978-3-319-24306-1_42

As derived from the *no-free-lunch theorem*, no learning algorithm can be specified as outperforming on the set of all real-world problems [20], we therefore search a mechanism which allows us to characterise the algorithms from the meta-features of the data sets which they classify with better accuracy and use these to build an algorithm recommender, that means, we rely on meta-learning.

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 [19] or to assist the user to determine the most suitable learning algorithm(s) for a problem at hand [7], among others.

In particular, we provide a framework which recommends practitioners the set of algorithms which should be applied to a concrete data set according to its characteristics and show its feasibility carrying out an experiment with data sets from the educational arena. Concretely, we address one of the oldest and best-known problems in educational data mining [16] that is predicting students performance. Furthermore, we discuss which meta-features, according to our experimentation, are more suitable for this challenging problem. We utilise regression techniques in this case study, unlike this paper [21] in which classification algorithms were applied with the aim of selecting the best classifier.

This paper is organised as follows: Section 2 briefly describes the different elements which comprise our framework, introducing previously the meta-learning field. Section 3 describes the methodology used in our case study and the setting of our experiment. Section 4 presents and discusses the results obtained showing the feasibility of our proposal. Finally, conclusions and future works are outlined in Section 5.

2 Background and an Overview of our Framework

Our aim, as previously mentioned, is to provide novice miners with a tool which help them to analyze the behaviour of different machine learners on different data sets and enable them to choose the more suitable algorithm for their data set at hand.

Meta-learning aims at learning the relationship between the meta-features extracted from the data sets and the algorithms performance applied on them. Thus, a meta-learning system 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. In other occasions, instead of selecting an algorithm, a ranking of algorithms is provided [2]. Different approaches for building the predictor are found, mainly based on classification [10,13,21] and regression [8,14]. Recently,

a new approach called meta-learning template [9] has arisen with the aim of recommending a hierarchical combination of algorithms.

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 [17,19].
- Information theoretic features used for characterising data sets containing categorical attributes such as class entropy or noise to signal ratio [5].
- Model-based meta-features, which collect the structural shape and size of a decision tree trained on the data sets [11].
- Landmarkers, which are meta-features calculated as the performance measures achieved by using simple classifiers [12].
- Complexity features, that characterize the apparent complexity of data sets for supervised learning [6]. These are provided by DCoL (data complexity library) [1,21].
- Contextual features, i.e., characteristics related to data set domain [21].

Figure 1 depicts our proposal graphically. As can be observed, the framework basically makes use of four workflows, one for extracting meta-features of the data sets, another one for loading the descriptive information of each experiment performed on each data set into the database; the third one, responsible for building a regressor for each type of algorithm used in the training phase; and the last one, responsible for carrying out the predictive phase, that means, reading a new data set, extracting its meta-features, applying this meta-data set to the regressors previously built and showing the algorithms ranked according to the value of accuracy predicted by themselves. Accuracy is directly calculated by running each regressor.

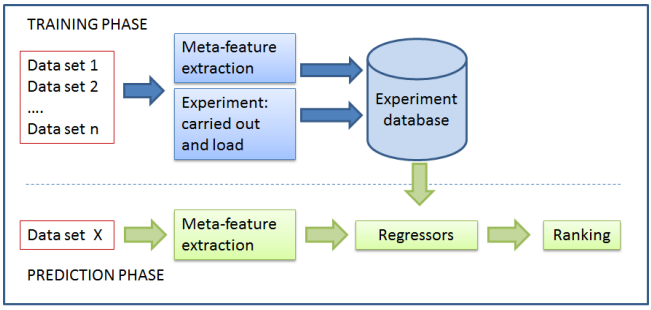

Fig. 1. Overview of our framework

The database schema used to gather the experiments is shown in Figure 2. We designed this based on the one proposed in [18] which gathers machine learning experiments. But this had to be extended to collect meta-features of each data set and the set of meta-features which comprise each training data set. A complete description of this database model is found in [3].

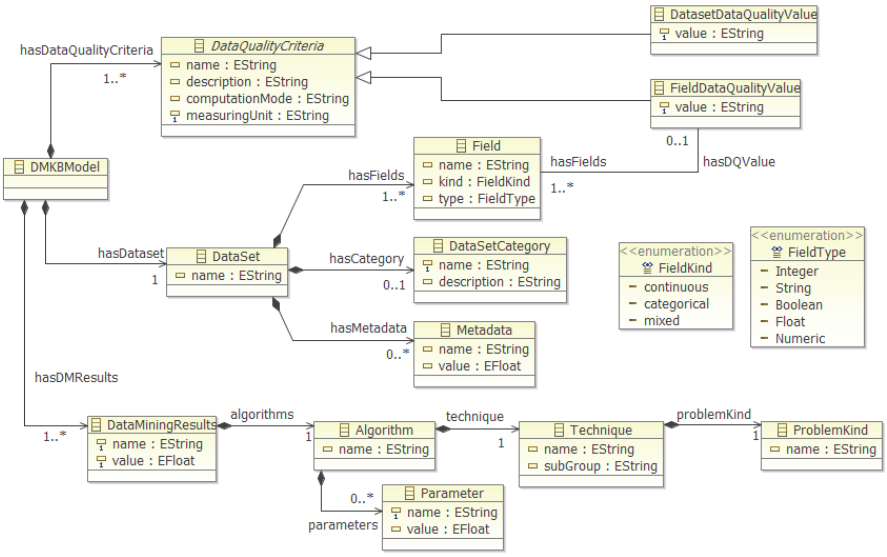

Fig. 2. Database model

3 Experiment Design

Next, we explain how the experiment has been carried out to show the feasibility of our approach.

The data sets used came from educational arena, in particular, they gather the activity performed by students in thirty different e-learning and blended courses hosted in a Moodle platform. This activity is measured by means of several metrics such as the total number of sessions open by each student in the course and in each tool of the course (tests, contents, forum,...), the number of self-tests performed, the number of messages posted and answered in the forum, among others. All attributes are numeric except the class attribute which collects if the learner failed (positive class) or passed (negative class) the course. Next, we extracted their meta-features. Concretely, we used the following ones:

– Simple meta features: number of instances, number of attributes, number of instances of the positive class (fail) and number of instances of the negative class (pass).

- Statistical measures: min, max and average value of the skewness and kurtosis of all the attributes of the data set calculated by means of the MATH3-apache Java library.
- Landmarkers. In this case we included as meta-features the accuracy achieved by the following weak classifiers: LinearDiscriminant (LD), BestNode with gain-ratio criterion (BN), RandomNode (RN), NaïveBayes (NB) and 1-N, all available in Weka or RapidMiner.
- Complexity meta features: we used the fourteen features offered by DCoL software that are the maximum Fisher's discriminant ratio (F1), the directional-vector maximum Fisher's discriminant ratio (F1v), the overlap of the per-class bounding boxes (F2), the maximum (individual) feature efficiency (F3), the collective feature efficiency (sum of each feature efficiency)(F4), the fraction of points on the class boundary (N1), the ratio of average intra/inter class nearest neighbor distance (N2), the training error of a linear classifier (N3), the fraction of maximum covering spheres (T1), the average number of points per dimension (ratio of the number of examples in the data set to the number of attributes)(T2), the leave-one-out error rate of the one-nearest neighbor classifier (L1), the minimized sum of the error distance of a linear classifier (L2) and the nonlinearity of a linear classifier (L3).

As our data sets only have numeric features, no information-theory measures were used. In a future work, we will include these meta-features by using multi-class data sets along with nominal predictor attributes.

As can be observed in Tables 1,2 and 3 meta-features extracted from training data sets take a wide range of values.

Table 1. Range of values of the complexity meta-features

F1	F1v	F2	F3	F4	L1	L2
0.04-29.46	0.03-370.82	0-0.2	0.02-0.88	0.05-1	0.27-0.92	0.09-0.45
L3	N1	N2	N3	N4	T1	T2
0.07-0.5	0.05-0.93	0.25-1.23	0-0.67	0-0.49	0.6-1	0.82-33.62

Table 2. Range of values of the simple and statistical meta-features

#N Ins.	#N Att.	#N Fail	#N Pass
13-504	3-28	5-220	3-433

Max. Skew.	Min Skew.	Avg. Skew.	Max. Kurt	Min. Kurt	Avg. Kurt
8.02-500.12	(-)1.51-15.22	2.99-131.43	1.43-22.33	(-)13.48-3.23	(-)2.3-10-36

Next, we run five classifiers on these thirty data sets using their default setting and 10-fold cross-validation. These were C4.5 (J48 version from Weka),

Table 3. Range of values of the landmarkers

Acc. NB	Acc. LD	Acc. BN	Acc. RN	Acc. 1NN
23.33-95.35	35-97.5	23.08-100	31.33-100	40-100

RandomForest, RIPPER (JRip version from Weka), k-NearestNeighbourds with auto-selection of the k number o neighbours (iBk in Weka), and LogisticRegression. As can be observed, each one follows a different learning paradigm. The accuracy achieved by each classifier on each data set during the training was stored in the data base.

Then, we generated five meta-data sets, one for each classifier. Each data set contained the meta-features of the training data sets along with the accuracy achieved by that specific classifier. For the sake of studying the behaviour of each group of meta-features, we built different linear regression models by using different combinations of meta-features:

1. Using all the meta features available.
2. Using only the meta-features which belong to each group (simple, statistical, complexity or landmarkers) separately.
3. Using 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. Tree thresholds were used, 10%, 40% and 70% to choose features according to their relevance.

All the linear regression models were generated using leave-one-out strategy as evaluation process. The RMSE of each regression model computed was also stored in the database.

Finally, we used two new data sets for testing our approach. We compared the predicted accuracy achieved by our regressors (recommenders) with the real accuracy achieved by the same classifiers. This allows us to measure at what extent our proposal actually shows a reliable ranking.

4 Results and Discussion

First of all, we show a summary of the results achieved in the regressors building phase. Table 4 displays, in the first column, "Avg. Acc.", the RMSE which results as a consequence of averaging the accuracy achieved for each classifier on all data sets. This will be used as base result in our experiment. The rest of the columns gather the RMSE obtained in the building of each linear regression model by using all meta-features ("All"), only the meta-features of a concrete group, or by applying a leave-one-out feature selection (FS) with a threshold of 10%, 40% and 70% ("FS 10%","FS 40%","FS 70%").

Reading this table, one notices that using all the meta-features to build the regressors does not lead to a better result than using the average accuracy

Table 4. RMSE of the regressors built

	Avg. Acc.	Simple	Statistic	Complexity	Landmark	All	FS 10%	FS 40%	FS 70%
RF	0.108	0.118	0.110	0.274	0.056	0.153	0.153	0.090	0.060
Jrip	0.105	0.113	0.104	0.333	0.063	0.261	0.099	0.050	0.056
J48	0.098	0.118	0.110	0.336	0.052	0.321	0.091	0.067	0.051
LR	0.122	0.135	0.139	0.252	0.078	0.435	0.296	0.122	0.061
iBk	0.141	0.126	0.121	0.265	0.066	0.279	0.074	0.067	0.062
Avg.	0.115	0.122	0.117	0.292	0.063	0.289	0.143	0.079	0.058

directly. This is mainly due to the fact that there are several meta-features that hinder the building of a good model. This is clearly endorsed by the results obtained when the meta-features with a relevance lower than 10% are removed from the meta-data set. In this case the regression models based on iBk, J48 and Jrip performed better than the base case. The results improve further when more meta-features are dropped. In fact, when the meta-data set only include the features with a relevance higher than 70%, all regressors performed better than the base case.

Regarding using only a meta-features group to generate the regression models, the results show that the landmakers are good predictors since the RMSE achieved is close to the one obtained with "FS 70%". In fact, the model built using RamdomForest with only landmakers has the lowest RMSE. The use of the remaining groups of meta-features individually does not seem to get more accurate models. The results thus yield that the best way to achieve a good ranking system based on meta-learning is applying a feature-selection algorithm with a high removal threshold on the meta-features data set.

Table 6 displays the number of times that each meta-feature selected by "FS 70%" was used in the building of our five regressors (min of 1, max of 5).

Table 5. Times that the meta-features are selected with "FS 70%"

F1	F1v	F2	F3	F4	T1	T2	skMin	kurtMin	1NN	BN	RN	LD	NB
5	2	1	2	2	1	1	1	2	5	5	5	3	1

The landmarkers calculated as the accuracy achieved by the 1-NN, BN and LD algorithms were always selected. This fact is aligned with our first conclusion which stated that the landmarkers are the best meta-features for our purpose. Nevertheless, there are also other meta-features which show a high relevance. That is the case of the complexity measure labelled as F1, which was chosen 5 times too. Furthermore, as a result of the fact that regressors built from the "FS 70%" meta-data sets performed better in 4 out of 5 techniques, we can say that the most suitable strategy to build an algorithm ranking system is to calculate all meta-features and use the most relevant. Although using only landmarkers would also lead to good results, as was concluded after carrying out

a paired t-test with a significance level of 0.01 between the case base regressor and landmakers-based regressors.

Regarding the predictive phase, we tested our proposal with two data sets from two different courses, with the aim of discovering the ranking of techniques which suggests us for each one. We used the "FS 70%" regression model of each one of the algorithms. The real (R.Acc.) and predicted (Pr.Acc.) accuracy for each data set and classifier are shown in Table 6, as well as the ranking of the classifiers (1 to 5 from better to worse performance).

Table 6. Output of our recommender system for the two test data sets

	Dataset1		Dataset2	
	Pr.Acc.(rank)	R.Acc.(Rank)	Pr.Acc.(rank)	R.Acc.(Rank)
RF	0.81 (3)	0.9(1)	0.67 (3)	0.64 (3)
Jrip	0.82 (2)	0.87(3)	0.75 (1)	0.70 (1)
J48	0.94 (1)	0.88(2)	0.65 (5)	0.64 (3)
LR	0.81 (3)	0.85 (4)	0.73 (2)	0.70 (1)
iBk	0.80 (5)	0.78 (5)	0.66 (4)	0.67 (2)

As can be observed, the recommendation for the first data set would be to use the J48 algorithm, since it has the higher predicted accuracy. This classifier achieved the second real higher accuracy, so that although our recommender did not recommend the best classifier, its accuracy is very close to it. Regarding the second data set, our proposal recommended the two better classifiers, Jrip and Logistic Regression. So, we can conclude that our approach works fine and can be a useful tool for helping novice data miners to decide which algorithms to utilise for their data set at hand.

5 Conclusions

This paper describes a framework which allows practitioners to discover what algorithm is more suitable for applying on certain data set. In fact, it offers a ranking of algorithms, thus the data miners can also evaluate what other techniques could well be used if their difference in accuracy with respect to the first one is not very large. Our proposal relies on meta-learning, that means, the use of a database which collects the results of different learning experiments along with the meta-features of the data sets involved with the aim of building a recommender which informs us about the better technique for a problem at hand.

The experimentation carried out shows that the use of linear regression for building this recommender works suitably and that the most significative meta-features for our purpose are landmakers, although, as demonstrated, the most effective recommender was built by applying a feature selection algorithm on all meta-features establishing a high threshold.

This work has a few limitations that will be addressed in a near future. On the one hand, we should extend our experimentation by including multiclass data sets and data sets with both numeric and nominal attributes. Furthermore, we should feed our database with more experiments applying more mining algorithms and registering different performance measures. Finally, we should build recommenders with other regression techniques and evaluate which one is the most suitable.

References

1. Cavalcanti, G., Ren, T., Vale, B.: Data complexity measures and nearest neighbor classifiers: a practical analysis for meta-learning. In: 2012 IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI), vol. 1, pp. 1065–1069, November 2012
2. Romero, C., Olmo, J.L., Ventura, S.: A meta-learning approach for recommending a subset of white-box classification algorithms for Moodle datasets. In: Proc. 6th Int. Conference on Educational Data Mining, pp. 268–271 (2013)
3. Espinosa, R., García-Saiz, D., Zorrilla, M.E., Zubcoff, J.J., Mazón, J.: Development of a knowledge base for enabling non-expert users to apply data mining algorithms. In: Accorsi, R., Ceravolo, P., Cudré-Mauroux, P. (eds.) Proceedings of the 3rd International Symposium on Data-driven Process Discovery and Analysis, Riva del Garda, Italy, August 30, 2013. CEUR Workshop Proceedings, vol. 1027, pp. 46–61. CEUR-WS.org (2013). http://ceur-ws.org/Vol-1027/paper4.pdf
4. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The kdd process for extracting useful knowledge from volumes of data. Commun. ACM **39**(11), 27–34 (1996)
5. Hilario, M., Kalousis, A.: Building algorithm profiles for prior model selection in knowledge discovery systems. Engineering Intelligent Systems **8**, 956–961 (2002)
6. Ho, T.K.: Geometrical complexity of classification problems (2004). CoRR cs.CV/0402020
7. 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)
8. 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)
9. Kordík, P., Cerný, J.: On performance of meta-learning templates on different datasets. In: IJCNN, pp. 1–7. IEEE (2012)
10. 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)
11. Peng, Y.H., Flach, P.A., Soares, C., Brazdil, P.B.: Improved dataset characterisation for meta-learning. In: Lange, S., Satoh, K., Smith, C.H. (eds.) DS 2002. LNCS, vol. 2534, pp. 141–152. Springer, Heidelberg (2002)
12. Pfahringer, B., Bensusan, H., Giraud-carrier, C.: Meta-learning by landmarking various learning algorithms. In: Proceedings of the 17th International Conference on Machine Learning, pp. 743–750. Morgan Kaufmann (2000)
13. 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)

14. Reif, M., Shafait, F., Goldstein, M., Breuel, T., Dengel, A.: Automatic classifier selection for non-experts. Pattern Analysis and Applications **17**(1), 83–96 (2014). http://dx.doi.org/10.1007/s10044-012-0280-z
15. Rice, J.: The algorithm selection problem. Adv. Comput. **15**, 65–118 (1976)
16. Romero, C., Ventura, S.: Data mining in education. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **3**(1), 12–27 (2013)
17. Segrera, S., Pinho, J., Moreno, M.N.: Information-theoretic measures for meta-learning. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI), vol. 5271, pp. 458–465. Springer, Heidelberg (2008)
18. Vanschoren, J., Blockeel, H., Pfahringer, B., Holmes, G.: Experiment databases. Machine Learning **87**(2), 127–158 (2012). http://dx.doi.org/10.1007/s10994-011-5277-0
19. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial Intelligence Review **18**, 77–95 (2002)
20. Wolpert, D.H.: The supervised learning no-free-lunch theorems. In: Proc. 6th Online World Conference on Soft Computing in Industrial Applications, pp. 25–42 (2001)
21. Zorrilla, M., 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., Raś, Z.W. (eds.) RSEISP 2014. LNCS, vol. 8537, pp. 285–292. Springer, Heidelberg (2014). http://dx.doi.org/10.1007/978-3-319-08729-0_28