Improving Meta-learning for Algorithm Selection by Using Multi-label Classification: A Case of Study with Educational Data Sets

Juan Luis Olmo [1] , Cristóbal Romero [1] , Eva Gibaja[1] , Sebastián Ventura[1 2]

[1] *Department of Computer Science and Numerical Analysis,*
University of Córdoba, 14071 Córdoba, Spain
E-mail: {jlolmo,cromero,egibaja,sventura}@uco.es
[2] *Department of Computer Science, King Abdulaziz University,*
Jeddah 22254, Saudi Arabia

Received 4 November 2014

Accepted 6 October 2015

Abstract

Recommending classification algorithms is an open research problem the solution to which is of tremendous value for practitioners and non-experts data mining users such as educators. This paper proposes a new meta-learning framework for educational domains based on the use of multi-label learning for selecting the best classification algorithms in order to predict students' performance. In short, the framework considers an offline phase where statistical tests are performed to find the subset of algorithms that achieves the best performance over the repository of educational data sets. The subset of algorithms along with the meta-features extracted from the training data are used to generate a multi-label data set. A multi-label classifier is then trained and, in an online phase, this model is used to recommend the most suitable classification algorithms to be applied to new unseen data sets. This new multi-label meta-learning approach has been applied to a repository of educational data sets generated from Moodle usage data. The results obtained show significant improvement compared with a previous nearest neighbor proposal, demonstrating the suitability of the new framework.

Keywords: Meta-learning, Multi-label classification, Educational data mining, Students' performance

1. Introduction

Classification algorithm selection is a very important and challenging issue. As the classification task of data mining (DM) [1] is a much studied field, resulting in a high number of available algorithms of different paradigms, a user can apply multiple choices to a given classification problem, with considerably different levels of performance [2]. In fact, it is generally accepted that no single learning algorithm can dominate another algorithm over all possible learning problems (this is also known as the no free lunch theorem [3]). If we extrapolate this reasoning to a specific domain where users do not necessarily have to be familiar with DM techniques, such as education, then recommending appropriate algorithms for their specific data becomes even more important. As an alternative, an educator could simply run all available classification algorithms and choose the one with the best performance. However, it may be computationally prohibitive to run all possible algorithms. Not all models obtained by the previous classification algorithms are equally interpretable, either. In fact, classification algorithms can be grouped in black and white box models [4]. Black models normally obtain high classificational accuracy, but their explanation of the results is obscure and difficult to understand. On the other hand, white box models, such as decision trees and rule-based algorithms, are more useful since they normally provide a set of IF-THEN classification rules that are one of the most popular ways of representing knowledge thanks to their simplicity and comprehensibility.

The importance of comprehensible classification models is often a prerequisite for users to trust the model's predictions and follow the recommendations associated with those predictions [5]. For example, the need for trusting computational predictions to be particularly strong in educational applications. In education, the prediction models obtained should be comprehensible/interpretable for instructors [6] in order that these models could be used directly for decision making [7] and provide an explanation for the classification. Finally, one of the objectives of educational data mining (EDM) [8,9] is to design easy-to-use tools for educators and non-expert users of DM. Nowadays, general DM tools range from commercial (such as DBMiner, SPSS Clementine, SAS Enterprise Miner, or IBM Intelligent Miner) to open-source solutions (such as WEKA and RapidMiner). Unfortunately, not all of these tools are specifically designed for educational purposes, and educators may be overwhelmed by the high number of algorithms and configuration options that these tools present.

A way to address the aforementioned problem is to employ meta-learning for classification algorithm recommendation [2]. It consists of a framework developed in the field of supervised machine learning to automatically predict algorithm performance, helping users with the algorithm selection process [10].

As a set of algorithms may be recommended, the meta-learning problem can be addressed from a multi-label learning (MLL) [11,12] perspective. In contrast with classical classification (a.k.a. single-label), where only one class label can be assigned to an instance, in MLL multiple target labels can be associated to each instance. Typical examples of MLL problems are classification of text [13] and multimedia [14] or bioinformatics [15]. Even new applications such as drug discovery [16], sentiment analysis [17], social network mining [18] or direct marketing [19] are continuing to emerge. However, MLL has not been widely used in the EDM field [20]. We can only cite two applications: the automatic tagging of learning objects [21] and the classification of learning styles from the learner profile [22].

In addition, although meta-learning has been widely applied to determine the best classification algorithm for a given data set [23,24,25,26], as far as we know, MLL techniques have not previously been used. Only work by Kanda et al. [27] can be cited, where selecting the most promising algorithm for the travelling salesman problem (TSP) is considered as a multi-label problem that involves only a few basic transformations of the data sets.

In this paper, we address an EDM-related problem that has not previously been tackled by MLL: the use of meta-learning for recommending classification algorithms for educational data. We first carry out an experimental study to analyze the application of the different state-of-the-art MLL algorithms to generate the classifier for making a recommendation. A second experimental study compares the results obtained by the current proposal with those obtained under another framework [28], which used a nearest-neighbor approach to determine the outputs from the recommendation. We demonstrate that the new multi-label framework is more suitable for this problem than the previous one, obtaining significant improvements in the recommendation.

The rest of this paper is organized as follows. In the next section we briefly review the two mainstream approaches, namely meta-learning and multi-label learning, providing a taxonomy. The proposed framework is described in detail in Section 3. Section 4 details the step by step application of the framework proposed for a specific repository of EDM data sets. Section 5 presents the experimental studies carried out, discussing the results obtained. Finally, some concluding remarks are outlined in Section 6.

2. Related work

Predicting students' performance is one of the oldest and most studied problems in EDM [6]. The goal of the prediction is to infer a categorical target or single aspect of the data (predicted variable) from some combination of other aspects of the data (predictor variables). This is a supervised learning predictive task that can either be addressed as classification, if the predicted variable is a categorical attribute, or as regression, if a numerical value is predicted. In the educational domain the objective is to estimate the unknown value of a student's performance, knowledge, score, or mark. Many different approaches and algorithms have been applied within classification and prediction tasks of DM to solve this problem [29]: decision trees [30], classification and regression trees [31], neural networks [32], bayesian networks [33], support vector machines [34], genetic algorithms [35], genetic programming [36], swarm programming [37], etc. As we can see, there is no general consensus on which algorithm or technique is the best option for an educator to be applied for classification or prediction over a given data set. In fact, as stated by the No-Free-Lunch theorem [3], there is no single classifier that performs best on all data sets, and thus, selecting and identifying the most adequate algorithm for a new data set is a difficult task.

For this reason, meta-learning can be used to help instructors to select the technique to be applied to classify their own data. Specifically, meta-learning can be defined as learning about learning, by taking results produced by learning as inputs and generalizing over them [38,39]. Within the machine learning community, meta-learning can be used in a variety of tasks that match the previous definition.

One of these tasks consists of learning how to combine the predictions of several classifiers. To this end, the predictions of each different model together with the correct class values constitute a meta-level dataset that is given as input for a meta-level classifier, the output of which is the final class. Several MLL approaches based on stacking [40] have been developed for this task. The target was to apply meta-learning to induce the dependencies between the labels while maintaining a linear complexity with the number of labels [41,42,43,44,45,46,47,48].

Another task concerns learning how to select the most appropriate learner for a certain problem according to a certain criterion (e.g. predictive accuracy). This is the type of meta-learning addressed in this paper. To carry out this task, several domains that can either be single or multi-label are described by a set of meta-features that are relevant to the performance of learning algorithms. This description, together with the performance of algorithms in these domain, constitutes a meta-domain or meta-learning data set to which a meta-learner is applied. This meta-learning data set may be single-label (the target is the best algorithm), multi-label (the target consists of several algorithms) or multi-score (the target is the performance of each algorithm). Finally, the output of the meta-learner may be single-label, multi-label, a ranking or a score for each algorithm. Table 1 shows a summary of the developed approaches in the field of meta-learning and MLL, classifying them according to the type of domain used in each stage of learning.

In [24], Chekina et al. proposed a meta-learning approach to recommend the best multi-label algorithm to be used over a certain domain. The domain was multi-label, but as only the best algorithm was recommended, the meta-learning data set was single-label. Kanda et al. [27] developed a meta-learning approach to recommend five different optimization meta-heuristics for solving TSP problems. The output of the meta-domain was the heuristic or set of heuristics that were able to find the best solution, thus producing a multi-label dataset. In this work, only three basic multi-label methods were applied; two of them (copy and ignore) have thus far received little consideration in the literature owing to its drawbacks, loss of information and low performance [50]. Later, in [49], multi-layer perceptrons were used to obtain a ranking of meta-heuristics. A meta-example was labelled with the performance on the five meta-heuristics and the output layer had five neurons that identified the ranks of the five meta-heuristics for the TSP instance provided in the input.

To our knowledge a complete framework that recommends a set of algorithms for a single-label classification problem by using a multi-label meta-learner does not exist in the literature. This meta-

REFERENCE	DOMAIN	META-LEARNING DATA SET	RECOMMENDATION
Chekina et al. 2011 [24]	multi-label	single-label	single-label
Kanda et al. 2011 [27]	single-label	multi-label	multi-label
Kanda et al. 2012 [49]	single-label	multi-score	ranking

Table 1: Classification of meta-learning and multi-label learning proposals according to the type of domain used in each stage of learning

learner is trained with statistical, complexity, and domain meta-features, and the multi-label target is obtained by using statistical evidence. It is also worth highlighting that the current work includes a full experimental study to determine which multi-label meta-learner of the state-of-art in multi-label learning is more suitable, and is also the first time that such an approach has been applied to EDM. In fact, meta-learning is used mainly in general domain and publicly available data sets such as those available at the UCI machine learning repository [51], but its application to EDM is quite limited, and only few works can be cited. The first one is focused on using meta-learning to support the selection of parameter values in a J48 classifier using several educational data sets [52]. A second work [28] proposed the employment of several classification measures to evaluate classifiers' performance. Non-parametric statistical tests were then performed to identify significant differences among algorithms for each data set in the repository. The meta-features of these data sets were also extracted. Once the educator has a new data set, after extracting its meta-features, a one nearest neighbor (1-NN) algorithm was used to detect the closest data set in the repository (using the meta-features previously extracted). The set of algorithms recommended for the new data set then coincided with the set of algorithms of its nearest neighbor. A more recent work that applied meta-learning to EDM was the paper by Zorrilla and Garcia [53], were meta-learning is used to build a recommender that help instructors (as non-expert data miners) in applying the right DM algorithm on their data sets. It is also worth noting the work by Zapata et al. [54], where meta-learning techniques are used in the field of learning objects recommendation in order to automatically obtain or predict the final ratings.

3. The proposed multi-label meta-learning framework

This section outlines the new multi-label meta-learning framework proposed for recommending classification algorithms for educational data. The approach can be split into two phases, as shown in Figure 1.

In the training or offline phase, the final goal is to generate a multi-label data set from educational data sets. To this end, several steps need to be addressed. In Step 1, a set of classification algorithms are executed over the original single-label educational data set so that several classification measures are calculated. Note that the algorithms selected at this point are those that will be recommended at the end of the process.

In Step 2, the algorithms that perform best must be found for each data set. The multiple-comparison Friedman or Iman&Davenport statistical tests can be employed for this purpose [55]. Both tests compare the mean ranks of k algorithms over N evaluation measures. These ranks indicate which algorithm obtains the best results considering all the measures studied. To calculate them, a rank of 1 is assigned to the algorithm with the highest value in the first measure studied, the algorithm with the next highest value in this measure is given a rank of 2, and so on. The same procedure is then carried out for the other evaluation metrics involved in the study. According to the test performed, it is possible to find out if the algorithms present significant differences in performance among themselves, according to the classification evaluation measures studied. If there are significant differences, a post-hoc test must then be performed to reveal such performance differences. Several statistical tests can be used at this point, such as Bonferroni-Dunn, Holm's and Hochberg's methods [55]. As result, we will know the subset

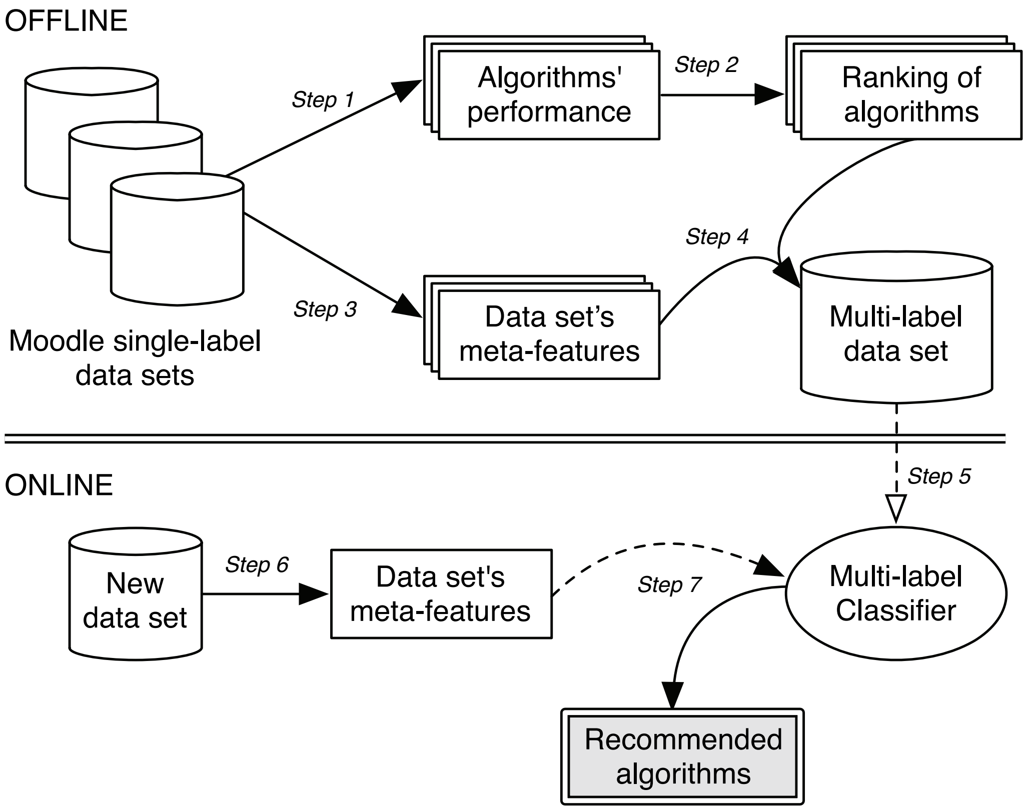


Figure 1: Multi-label meta-learning methodology

of algorithms recommended for each particular data set. Note that the algorithms recommended will not present significant differences among themselves regarding the classification metrics evaluated.

On the other hand, the meta-features of each original data set in the repository are extracted in Step 3, such as statistical, complexity and domain features. Then, in Step 4, the meta-features extracted for a given data set will become part of an instance of the multi-label data set. Specifically, the meta-features extracted will correspond to the predictive attributes, while the value of the labels will come from the subset of algorithms recommended for the same data set. The number of instances in the multi-label data set is equal to the number of single-label educational data sets in the repository, and the number of labels is equal to the number of algorithms employed in Step 1. Note that, for a given instance, a value of 1 will be set in a label if the algorithm associated belonged to the subset of algorithms recommended in Step 2, and 0 otherwise.

Finally, Step 5 consists in training a multi-label classifier using the multi-label data set generated as training data. Any kind of multi-label classification algorithm can be employed to generate the classifier.

In the prediction or online phase, given a new educational data set, its meta-features must first be extracted in Step 6. The same meta-features used to generate the multi-label data set should be extracted. Then, in Step 7, the values of these meta-features can be used as input for the multi-label classifier, which will generate the algorithms recommended for the new data set by prediction.

4. Experimental study on educational data

The goal of the experimental study carried out in this paper aims to demonstrate the validity of the multi-label meta-learning framework proposed, using a specific repository of educational data sets. The repository consisted of 32 classification higher education data sets used to predict students' performance [56] that were generated from Moodle usage data [8]. Moodle is a free learning management system that allows powerful, flexible and engaging online courses to be created [57]. The data were collected during the six-year period between 2007 and 2012 and relate to university Computer Science students. As predictive attributes, these data sets comprise various information about students' interaction in the Moodle learning platform during a course, and the class to be predicted is related to the final mark obtained by students in the course.

This section focuses on describing the experiment performed, explaining the different steps carried out until the multi-label data set is generated, the cross-validation model used, the multi-label metrics calculated for each classifier, and the configuration used for the multi-label algorithms executed.

4.1. Step 1: Evaluating the algorithms' performance

The target audience of this work includes teachers or professional instructors who might not be familiar with the knowledge discovery process [58]. For this reason, it was assumed that the algorithms that should be recommended to these domain experts are those that allow the creation of models that can be easily understood and used directly in the decision-making process. In this context, high-level representation techniques such as decision trees and decision rules [59] are especially interesting, since they allow the user to interpret and understand the knowledge extracted. The algorithms considered in this work are thus restricted to decision trees and rule-based algorithms, although other paradigms could have been taken into account. In particular, we have employed the following rule-based classifiers and decision trees provided by the Weka tool [60]*: ConjunctiveRule [60], DecisionTable [61], DTNB [62], JRIP [63], NNge [64], OneR [65], PART [66], Ridor [67], ZeroR [60], BFTree [68], DecisionStump [69], J48 [70], J48graft [71], LADTree [72], LMT [73], NBTree [74], RandomForest [75], RandomTree [60], REPTree [76], SimpleCart [77].

In Step 1, we run each algorithm over each specific data set in our repository. More specifically, we carried out a stratified ten-fold cross validation procedure, where each data set was randomly split into ten mutually exclusive folds, with each fold consisting of approximately the same proportion of classes as in the original data set. Each algorithm was executed ten times, with a different fold left out as the test set each time, the other nine being used for training. This gives a total of 6400 executions (20 algorithms × 32 data sets × 10 folds). Several classification measures were used to compare algorithms' performance [78]:

- Sensitivity (Sen), a.k.a. recall, which indicates the ability of the model to detect positive instances:

$$\frac{T_P}{T_P + F_N} \tag{1}$$

- Precision (Prec), which indicates the number of positive instances correctly identified among all the instances predicted as positives.

$$\frac{T_P}{T_P + F_P} \tag{2}$$

- F-measure (F-M), which is the harmonic mean between sensitivity and precision.

$$2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3}$$

- Kappa (Kap), which is an alternative measure to accuracy since it compensates for random hits [79]. It evaluates the merit of the classifier, i.e. the actual hits that can be attributed to the classifier and not by mere chance. Cohen's kappa statistic ranges from -1 (total disagreement) to 0 (random classification) to 1 (total agreement). It is calculated by means of the confusion matrix as follows:

$$Kappa = \frac{N\sum_{i=1}^{k} x_{ii} - \sum_{i=1}^{k} x_{i.}x_{.i}}{N^2 - \sum_{i=1}^{k} x_{i.}x_{.i}} \tag{4}$$

where x_{ii} is the count of cases in the main diagonal of the confusion matrix, N is the number of examples, and $x_{.i}$ and $x_{i.}$ are the column and row total counts, respectively. The kappa rate also penalizes

*The Weka machine learning software is publicly available at `http://www.cs.waikato.ac.nz/ml/index.html`

Table 2: Algorithms' performance for dataset12

Algorithm	Sen	Prec	F-M	Kap	AUC
ConjunctiveRule	0.778	0.800	0.770	0.5404	0.735
DecisionTable	0.790	0.809	0.783	0.5668	0.720
DTNB	0.802	0.828	0.795	0.591	0.717
JRip	0.765	0.790	0.756	0.514	0.689
NNge	0.691	0.690	0.690	0.374	0.686
OneR	0.716	0.742	0.701	0.409	0.698
PART	0.753	0.794	0.738	0.485	0.729
Ridor	0.741	0.785	0.723	0.458	0.721
ZeroR	0.543	0.295	0.382	0.000	0.444
BFTree	0.778	0.800	0.770	0.540	0.715
DecisionStump	0.728	0.763	0.712	0.433	0.701
J48	0.765	0.802	0.753	0.512	0.727
J48graft	0.778	0.810	0.767	0.538	0.729
LADTree	0.753	0.772	0.744	0.489	0.676
LMT	0.778	0.810	0.767	0.538	0.744
NBTree	0.790	0.819	0.781	0.565	0.734
RandomForest	0.765	0.782	0.758	0.516	0.692
RandomTree	0.728	0.752	0.715	0.436	0.615
REPTree	0.741	0.762	0.730	0.463	0.705
SimpleCart	0.778	0.800	0.770	0.540	0.701

all-positive or all-negative predictions, especially in imbalanced data problems. Kappa is very useful for multiclass problems, measuring classifier's accuracy while compensating for random success.

- Area under the ROC curve (AUC), which is a commonly used evaluation measure in imbalanced classification [80]. ROC curve presents the tradeoff between the true positive rate and the false positive rate. It ranges from 0.5 (random classifier) to 1.0 (perfect classifier), and is computed by using the entries of the confusion matrix:

$$\begin{aligned} AUC &= \frac{1 + TPrate - FPrate}{2} \\ &= \frac{1 + \dfrac{T_P}{T_P + F_N} - \dfrac{F_P}{F_P + T_N}}{2} \qquad (5) \end{aligned}$$

Table 2 shows the average values obtained for these measures by each algorithm on dataset12.

4.2. Step 2: Ranking single-label algorithms

In Step 2, we outlined the algorithms that did not present significant differences when classifying a given data set, taking the five classification measures into account simultaneously. To this end, the performance differences among algorithms were statistically evaluated. The process followed is then detailed for the sample table referred to above. The multiple-comparison Iman&Davenport [55] test was carried out. According to this test, we state that all algorithms are equivalent if the null hypothesis is accepted. In contrast, if the null hypothesis is rejected, we will state that there are differences between the algorithms. With a significance level of $\alpha = 0.1$ the Iman&Davenport statistic of average rankings distributed according to the F-distribution rejected the null-hypothesis in the 32 cases studied, indicating the existence of significant differences among the classifiers.

To reveal such performance differences, we applied the Bonferroni-Dunn post-hoc test, the focus being on all the possible pairwise comparisons among the algorithms. The critical value revealed by this test at the same significance level of $\alpha = 0.1$ was 9.533. Those algorithms whose rank belonged to the interval between the value of the highest rank and this latter plus the critical value, were the subset of algorithms recommended for that particular data set, given that there were no significant differences among them.

For instance, following the example of Table 2, after carrying out the Iman&Davenport and subsequent Bonferroni-Dunn non-parametric tests, the

Table 3: Subset of recommended algorithms for educational dataset12 and their ranking values

ALGORITHM	RANK
DTNB	2.333
NBTree	2.667
DecisionTable	3.667
LMT	5.250
ConjunctiveRule	5.333
J48graft	5.833
BFTree	6.667
SimpleCart	7.083
J48	9.000
PART	10.917
RandomForest	11.000
JRip	11.167

subset of classifiers that would be recommended for dataset12 is shown in Table 3, where the critical interval was [2.333, 2.333 + 9.533]. The remaining eight algorithms were not recommended since their rank exceeded the upper limit of the critical interval.

4.3. Step 3: Extracting meta-features from Moodle data sets

Following Step 3 of the framework proposed, the meta-features of the data sets were extracted. Sixteen features were considered from each data set which can be categorized into one of the following three groups: statistical, complexity, and domain, as shown in the section under the label "meta-features" in Table 4.

The statistical features comprise the number of instances or students (Ni), the number of numerical attributes (Nna), the number of categorical attributes (Nca), the number of classes or labels of the mark attribute such as Pass/Fail, High/Medium/Low, etc. (Nc), and the imbalance ratio (IR), which is the ratio between instances of the majority and minority classes.

On the other hand, taking into account the fact that the classification ability of classifiers depends on the specifics of the data [81], the complexity of each particular Moodle data set was analyzed. Different data complexity measures have been proposed to characterize the difficulty of a classification problem quantitatively [82], divided into three categories: measures of overlap of the feature values from different classes, measures of separability of classes, and measures of geometry, topology and density of manifolds. These measures take into account the geometrical regularities and irregularities of a data set, assessing different degrees of difficulty related to the boundary complexity. The extraction of complexity measures to be used as meta-features in meta-learning frameworks has been considered in several works [83,84]. The measures extracted as meta-features in this work were the following:

- F1, maximum Fisher's discriminant ratio. It computes the maximum discriminant power of each feature. For a given feature, it computes how spread are the classes with respect to a specific feature, comparing the difference between class means with the sum of class variances.
- F2, overlap of the per-class bounding boxes. It measures the overlap of the tails of the distributions defined by the instances of each class.
- F3, maximum (individual) feature efficiency. It computes the discriminative power of individual features and returns the value of the attribute that can discriminate the largest number of training instances.
- F4, collective feature efficiency. It follows the same idea of the previous F3 measure, but considering the discriminative power of all features.
- L1, minimized sum of the error distance of a linear classifier. It evaluates to what extent data is linearly separable, computing the sum of the difference between the prediction of a linear classifier and the actual class value.
- L2, training error of a linear classifier. This mea-

sure computes the error rate of a linear classifier defined for L1 on the original training set.

- N1, fraction of points on the class boundary. It provides the percentage of nodes that link different classes in a minimum spanning tree constructed over the data set, counting the number of points incident on an edge going across the two classes.
- N2, ratio of average intra/inter class nearest neighbor distance. For each instance x_i, we calculate the minimum distance to a neighbor instance belonging to the same class ($intraDist(x_i)$), as well as the mininum distance to a neighbor instance of any other class ($interDist(x_i)$). The result of this metric is assessed as the ratio of the sum of the intra-class distances to the sum of the inter-class distances for each instance.
- N3, leave-one-out error rate of 1-NN classifier. This is simply the error rate of a nearest neighbor classifier using a leave-one-out method. It denotes how close the examples of different classes are.
- L3, non-linearity of a linear classifier. It is a measure of non-linearity described first in [85]. It is computed by creating a test set by linear interpolation between points of the same class chosen randomly from the training set. The value returned is the error rate produced by the linear classifier over the test set.
- N4, non-linearity of the 1-NN classifier. It follows the same procedure of the L3 metric, but using a 1-NN classifier.
- T1, fraction of maximum covering spheres. This measure is based on the concept of adherence subset [86], which is a sphere centred on an instance of the data set which is grown as much as possible until reaching any instance of any other class. Therefore, an adherence subset contains a set of instances of the same class and cannot grow more without including instances of other classes. The T1 measure considers only the biggest adherence subsets, removing all those that are included in others. Then, it returns the number of adherence subsets normalized by the total number of instances.
- T2, average number of points per dimension. It is the ratio between the number of instances in the data set and the number of attributes. It is a rough indicator of sparseness of the data set.

Note that in order to obtain their complexity values for each data set, the DCoL library was used [87].

Finally, the last meta-feature in Table 4 is the source of the data set, which is specific of the educational domain, and in our case can take one of the following values, depending on the Moodle's source:

- Report: a report is a general summary about each student's interactions in a Moodle course. Moodle provides a flexible array of course activities, resources and assignments. Some examples of variables that we have stored in this type of data set are: the total time spent on the course, the number of sessions/times the course was accessed, the number of activities visited, the number of assignments performed, the average score obtained in the assignments, the total time spent on resources and activities, the total time spent on assignments, etc.
- Quiz: a specific summary about the interaction of each student with Moodle quizzes or tests. Moodle provides a large variety of question types, including multiple choice, true-false, and short answer questions. Some examples of variables that we have stored in this type of data set are the total time spent on all quizzes of the course and each quiz done, the total number of quizzes completed, the number of quizzes passed and failed, the average score obtained in quizzes, the number of questions correctly/incorrectly answered, etc.
- Forum: is a specific summary about the interaction of each student with Moodle forums. Moodle provides different types of forums for exchanging ideas by posting comments, which can be graded by the teacher or other students. Some examples of variables that we have stored in this type of data set are: the total time spent in a forum, the number of messages sent, the number of messages read, the number of threads created, the number of replies received, the average score obtained in a forum, etc.

Table 4: Multi-label meta-data set. The left side corresponds to statistical, complexity and domain features of single-label Moodle data sets for predicting students' performance. The right side corresponds to the classes, having one label per algorithm considered.

	STATISTICAL					COMPLEXITY			DOMAIN	ALGORITHMS				
DATA SET	Ni	Nna	Nca	Nc	IR	F1	...	T2	Source	Alg1	Alg2	Alg3	...	Alg20
dataset1	98	4	0	2	1.08	0.046	···	48.5	Report	0	1	1	...	1
dataset2	194	0	4	2	1.39	1.452	...	48.25	Report	0	1	1	...	0
dataset3	786	6	0	3	9.8	1.499	...	297.66	Quiz	0	1	1	...	1
dataset4	658	0	6	3	9.1	0.005	...	298	Quiz	1	1	0	...	0
dataset5	67	40	0	2	1.23	0.101	...	1.675	Quiz	1	0	1	...	0
dataset6	922	6	0	3	19.27	2.745	...	153.66	Quiz	0	1	0	...	1
dataset7	910	0	6	3	19.24	0.018	...	153.16	Quiz	1	1	0	...	0
dataset8	114	0	11	2	1.19	1.198	...	10.364	Forum	0	0	0	...	1
dataset9	42	0	11	2	6	1.029	...	3.818	Forum	1	0	0	...	0
dataset10	103	0	11	2	1.53	0.648	...	10.364	Forum	0	1	1	...	1
dataset11	114	11	0	2	1.43	0.079	...	10.364	Forum	1	0	0	...	1
dataset12	98	0	6	2	1.91	0.872	...	13.5	Forum	0	0	0	...	1
dataset13	81	6	0	2	1.19	0.094	...	13.5	Forum	1	1	1	...	0
dataset14	33	0	12	2	32	26.43	...	2.75	Forum	0	1	1	...	1
dataset15	82	0	12	2	3.1	0.11	...	6.833	Forum	0	0	0	...	0
dataset16	113	40	0	4	23.5	0.037	...	2.825	Quiz	0	0	0	...	1
dataset17	105	41	0	3	1.06	0.055	...	2.488	Quiz	1	1	1	...	1
dataset18	123	0	10	4	3.89	2.482	...	10.3	Quiz	0	1	1	...	0
dataset19	102	10	0	3	1.06	0.168	...	10.2	Quiz	1	0	0	...	1
dataset20	75	0	8	2	2.12	0.517	...	9.375	Report	0	1	0	...	1
dataset21	52	0	4	2	1.89	2.274	...	13	Report	0	0	0	...	1
dataset22	208	10	0	2	3.25	0.023	...	20.7	Report	1	0	1	...	0
dataset23	438	0	10	4	15.41	0.398	...	43.8	Report	1	0	0	...	0
dataset24	421	10	0	4	14.2	0.065	...	43.8	Report	0	1	1	...	1
dataset25	84	6	0	4	5.43	0.042	...	14	Report	0	0	0	...	0
dataset26	168	6	0	4	11.25	0.003	...	28	Report	1	1	0	...	1
dataset27	136	6	0	4	11.5	-1	...	22.667	Report	0	0	0	...	0
dataset28	283	0	10	2	1.67	0.842	...	28.2	Report	1	0	1	...	0
dataset29	155	0	10	2	1.21	0.18	...	15.5	Report	0	0	1	...	0
dataset30	72	6	0	4	11	-1	...	12	Report	1	1	1	...	1
dataset31	40	0	10	2	1.2	0.476	...	2	Quiz	1	0	0	...	0
dataset32	48	10	0	2	1.8	2.327	...	2	Quiz	1	1	0	...	1

$\underbrace{\qquad\qquad\qquad\qquad}_{meta-features}$ $\underbrace{\qquad\qquad}_{labels}$

Table 5: Features of the EDM multi-label data set

METRIC	VALUE
Number of instances	32
Number of attributes	20
Number of labels	20
LCard	12.68
LDen	0.63
DL	32

4.4. Step 4: Generating the multi-label meta-learning data set

The next step of the offline phase is the construction of the multi-label data set from the information extracted in the previous steps. Specifically, this presupposes the extraction of meta-knowledge that relates the meta-features obtained from the Moodle data sets in Step 3 to the algorithms that statistically perform best over those data sets, obtained as results in Step 2. In our study, this meta-knowledge is represented in a multi-label data set that presents 20 labels, one per algorithm considered in the framework. On the other hand, it has one instance per Moodle data set in the original repository. The predictive attributes of a given instance correspond to the meta-features extracted in Step 3 for the data set concerned, while each label has a value of 1 if the algorithm belongs to the subset of recommended algorithms for the data set in Step 2, or a value of 0 otherwise. The structure of the multi-label meta-learning data set is shown in Table 4. We have published this data set so that it is available to other researchers[†].

Table 5 summarizes the main features of the multi-label meta-learning data set. The number of instances corresponds to the number of data sets in the repository. The number of attributes is the number of meta-features extracted for each data set in the repository. The number of labels is the number of algorithms executed over each original data set. The label cardinality (LCard) is the average number of labels per pattern while the label density (LDen) is the cardinality divided by the total number of labels, and is used to compare data sets with different numbers of labels. Both measures are defined in [88]. Finally, the distinct labelsets (DL) [89] are described as the number of different label combinations in the data set. Table 6 shows the number of counts for each label along with their relative frequency.

4.5. Step 5: Training the multi-label classifier

This step links the offline and online phases. More specifically, it consists in generating a classifier by using any MLL algorithm, training the classifier by using the multi-label meta-learning data set shown in Table 4, which was generated in the previous step. The induced classifier can be later used to make predictions given a new unseen data set. In particular, it will generate a set of 1s and 0s as its output, one per label/algorithm, where a value of 1 will mean that this algorithm is suitable for use over the new data set, and 0 otherwise.

4.6. Step 6: Extracting meta-features from a new unseen educational data set

As regards the online phase, given a new unseen educational data set, it is first necessary to extract its meta-features. Exactly the same meta-features used in Step 3 for the repository of Moodle data sets have to be extracted and calculated for the new data set. In particular, statistical features can easily be extracted by the final user by counting the number of instances, the number of numerical attributes, the number of categorical attributes, the number of classes, and the imbalance ratio. Complexity characteristics can easily be extracted by using the open source DCoL library. Finally, the domain feature has to be indicated by the instructor or final user.

All these characteristics will make up a new unlabeled instance, i.e., an instance where the labels are unknown (these values will be predicted by the classifier). Note that this instance has the original number of attributes of the multi-label meta-learning data set (see Table 4).

4.7. Step 7: Recommending a set of suitable algorithms for the educational data set

Once an unlabeled instance has been generated from the meta-features of the new educational data set, the last step of the framework proposed consists in making a prediction, using this instance as input for the multi-label classifier trained in Step 5. As output, the classifier will generate a prediction. In this, those labels where a value of 1 is predicted are indicative of the fact that the corresponding white-box algorithm is a good option to be used for classifying the original data set. Otherwise, that algorithm is not appropriate to be used over the original data. Note that, owing to the multi-label nature of the classifier, a set of suitable algorithms can be recommended, since a value of 1 can be predicted for several labels.

As an example, we next show the output for a new unseen data set identified as dataset33. This new data set, not used for training the classifier, considers 28 university students participating in a Moodle forum for predicting the students' success or failure in a course about human-computer interaction. The interpretation and use of the classification model obtained can be very useful for the course instructor for detecting new students' final performance over time (since data were collected from the

[†] `http://www.uco.es/grupos/kdis/kdiswiki/index.php/EDMMultiLabelMetaLearning`

Table 6: Label frequency in the EDM multi-label data set

Label	Count	Rel. freq.
ZeroR	6	0.187
DecisionStump	11	0.343
RandomForest	13	0.406
ConjunctiveRule	15	0.469
NNge	16	0.500
OneR	17	0.531
JRip	19	0.594
REPTree	19	0.593
Ridor	20	0.625
DecisionTable	23	0.719
BFTree	23	0.719
LADTree	23	0.719
SimpleCart	23	0.719
PART	24	0.750
J48	25	0.781
LMT	25	0.781
RandomTree	25	0.781
DTNB	26	0.812
J48graft	26	0.812
NBTree	27	0.843

forum halfway through of the course), and to decide how to help students predicted to FAIL [6]. Then, after extracting the meta-features of dataset33 as explained in Section 4.6, the unlabeled instance that was generated was then used as input for the classifier (that was trained in this example using the Ensemble of Pruned Sets multi-label algorithm). As output of the classifier, several algorithms were recommended. Any of them could be used, but let's say the user, who is an educator in our domain, selects the J48graft algorithm [71], which is a grafted version of the well known C4.5 decision tree algorithm. This algorithm generates the decision tree shown in Table 7 when used over the dataset33. As we can see, the decision tree generated by the algorithm is a highly interpretable model of the students' performance, which is crucial for the instructor. It consists of a set of rules with the form IF-ELSE-IF, in which the THEN operator is indicated by the symbol ":". In this specific decision tree, students are divided into two major leaves, depending on whether students have a centrality (measure of a student's prominence in the forum) higher or lower than 0.019. Students having a centrality lower or equal to 0.019 are predicted to FAIL the course. Among those students having a centrality over 0.019, the model differentiates between those having more than 2 replies to their messages, who are predicted to PASS the course, and those that receive less than 2. This subgroup is divided into those students who write more than 107 words and are predicted to PASS the course, and those that write fewer words. These latter are divided into those that sent a number of messages lower than 4.5 and are predicted to FAIL the course, and those with more than 4.5, who are predicted to PASS the course.

On the other hand, to estimate the validity of the framework proposed for educational data, we have carried out an experimental study where we have followed a leave-one-out cross-validation procedure. Since the performance of the model can vary depending on the MLL classifier used, it is important to analyze which one gives the best results. Therefore, for each multi-label algorithm used, 32 classifiers were trained using the multi-label meta-learning data set induced in Step 5 as training data with the particularity of excluding one instance each time. That instance represents the meta-features of the educational data set excluded, and is used as input in Step 7 to make a multi-label prediction and to recommend the algorithms for such a data set.

Because the data set generated in the offline phase is a multi-label data set, specific performance metrics must be used to consider multiple outputs. In particular, two different types of metrics can be differentiated: label-based and example-based met-

Table 7: Decision tree generated by J48graft over dataset33

```
Centrality <= 0.019: FAIL
Centrality > 0.019
|    Replies <= 2
|    |    Words <= 107
|    |    |    Messages <= 4.5: FAIL
|    |    |    Messages > 4.5: PASS
|    |    Words > 107: PASS
|    Replies > 2: PASS
```

rics [88]. Both kinds were used in our experimental study (see Table 8).

Label-based metrics are calculated for each label based on the number of true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn), and any binary evaluation metric can be used with this type of approach. Because there are several confusion matrices, the average of the metrics can be calculated in two different ways: the macro and the micro approaches. The former is the arithmetic average of the measure over all the categories, while the latter considers predictions from all instances together and then calculates the measure across all labels. There is no consensus about using a macro or micro approach. According to [90,91], macro-averaged scores give equal weight to every category, while micro-averaged scores give equal weight to every example. Pestian et al. [92] pointed out that the macro approach is more appropriate when the system is required to perform consistently across all classes regardless of the frequency of the class, while the micro approach may be more appropriate if the density of the class is considerable. In this paper we thus follow the macro approach, in order to consider the same weight for each category. The following example-based measures have been used in the experiments:

- Recall calculates the percentage of relevant labels that are correctly predicted.
- Precision gives us the percentage of predicted labels that are relevant.
- Specificity is related to the ability to identify negative results, and is defined as the proportion of negative outputs that are actually negative.
- The macro F-measure is the harmonic mean of macro precision and macro recall.

Alternatively, example-based metrics are calculated for each test example and then averaged across the test set. Metrics to evaluate rankings and bipartitions are included in this group of metrics. Another group of example-based metrics originating from the information retrieval area [42] that are commonly used in MLL are precision, recall, specificity and F-measure, which were used as metrics to evaluate bipartitions (see Table 8). To define the metrics used, let $T = (x_i, Y_i) 1 \leqslant i \leqslant t$ be a multi-label test set with t instances, Y_i and Z_i the set of true and predicted labels for an instance, and let τ be the predicted ranking for an instance. The following example-based metrics were used:

- Coverage [13] is the metric that evaluates how far on average a learning algorithm needs to go down the ordered list of prediction to cover all the true labels of an instance. The smaller the value, the better the performance.
- Ranking loss [93] evaluates the average fraction of label pairs that are misordered for the instance. The lower the value of the metric, the better the performance. Note that $|E|$ is called in [94] *error-set-size*.
- Hamming loss [93] evaluates on average how many times an example-label pair is misclassified. This metric takes into account both prediction errors (an incorrect label is predicted) and omission errors (a correct label is not predicted) normalized over the total number of classes and the total number of examples. The lower the value, the better

Table 8: Metrics used in experiments

LABEL-BASED METRICS
$recall_{mac} = \frac{1}{q}\sum_{i=1}^{q}\frac{tp_i}{tp_i+fn_i}$
$precision_{mac} = \frac{1}{q}\sum_{i=1}^{q}\frac{tp_i}{tp_i+fp_i}$
$specificity_{mac} = \frac{1}{q}\sum_{i=1}^{q}\frac{tn_i}{tn_i+fp_i}$
$F-measure_{mac} = 2 \times \frac{precision_{mac} \times recall_{mac}}{precision_{mac}+recall_{mac}}$
EXAMPLE-BASED METRICS
$coverage = \frac{1}{t}\sum_{i=1}^{t}\max_{\lambda \in Y_i}\tau_i(\lambda) - 1$
$error-set-size = \lvert E \rvert =$
$\{(\lambda,\lambda')\lvert\tau_i(\lambda) > \tau_i(\lambda'),(\lambda,\lambda') \in Y_i \times \overline{Y_i}\}$
$ranking\ loss = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{\lvert Y_i\rvert\lvert\overline{Y_i}\rvert}\lvert E\rvert$
$Hamming\ loss = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{q}\lvert Z_i \Delta Y_i\rvert$
$recall = \frac{1}{t}\sum_{i=1}^{t}\frac{\lvert Z_i \cap Y_i\rvert}{\lvert Y_i\rvert}$
$precision = \frac{1}{t}\sum_{i=1}^{t}\frac{\lvert Z_i \cap Y_i\rvert}{\lvert Z_i\rvert}$
$specificity = \frac{1}{t}\sum_{i=1}^{t}\frac{\lvert \overline{Z_i} \cap \overline{Y_i}\rvert}{\lvert \overline{Z_i}\rvert}$
$F-measure = 2 \times \frac{precision \times recall}{precision+recall}$

the performance of the classifier. Δ stands for the symmetric difference of two sets.

4.8. Experimental setup

Several multi-label algorithms were used in this experimental study to induce the classifier of Step 5 of the framework. Their implementation is freely available at the MULAN library[‡] and all of them have been tested using a leave-one-out cross-validation, running each one 32 times in total. Regarding problem transformation methods, which transform a multi-label problem into one or more single-label ones to apply any classic machine learning algorithm, the following methods were used: Binary Relevance (BR) [88], Label Powerset (LP) [88], Ensemble of Pruned Sets (EPS) [95], Calibrated Label Ranking (CLR) [96], RAndom *k*-labELsets (RA*k*EL) [89], and Ensemble of Classifier Chains (ECC) [97]. Regarding algorithm adaptation methods, which adapt a single-label algorithm in order to deal directly with multi-label data, the following algorithms were employed: AdaBoost.MH [93], Multi-label k-nearest neighbour (ML-*k*NN), Instance Based Learning by Logistic Regression (IBLR) [98], and Backpropagation for Multilabel Learning (BP-MLL) [15].

Regarding the algorithms setup, BR, LP and CLR transformations were run with the classical J48 as a base algorithm. RA*k*EL was run with its default parameter configuration, consisting in an LP with the J48 algorithm as base classifier, a subset size of 3, a number of models equal to twice the number of labels and 0.5 as threshold value. BP-MLL was run with a 0.05 learning rate, 100 epochs and the number of hidden units equal to 20% of the input units, the configuration recommended in [15]. The number of neighbors in ML-kNN was set to 10 and the smoothing factor to 1 as recommended in [99]. As for IBLR, this used 10 nearest neighbors as recommended by the authors in [98]. AdaBoost.MH used the default configuration established in MULAN without parameters. ECC also used the default configuration with J48 as base classifier, 10 models, using confidences and sampling with replacement. Finally, EPS used the default MULAN's configuration, consisting of 10 models in the ensemble, strategy A (keeping the top b=2 ranked subsets), 66%

[‡] MULAN library for multi-label learning can be reached at `http://mulan.sourceforge.net/`

Table 9: Multi-label classification results

MEASURE	ML*k*NN	IBLR	BPMLL	AB.MH	RA*k*EL	BRJ48	LPJ48	CLR	EPS	ECC
Macro F-measure	0.6313	0.6063	0.6266	0.6594	0.6469	0.6469	0.6734	0.6563	**0.6922**	0.6750
Example based F-measure	0.7227	0.6718	0.6451	0.7789	0.7226	0.7226	0.7471	0.7362	**0.7801**	0.7485
Macro precision	0.6313	0.6063	0.6266	0.6594	0.6469	0.6469	0.6734	0.6563	**0.6922**	0.6750
Example based precision	0.6934	0.7089	0.6289	0.6558	0.7211	0.7211	**0.7351**	0.7198	0.7142	0.7305
Macro recall	0.6313	0.6063	0.6266	0.6594	0.6469	0.6469	0.6734	0.6563	**0.6922**	0.6750
Example based recall	0.7736	0.6601	0.7015	**0.9775**	0.7434	0.7434	0.7818	0.7756	0.8785	0.7864
Macro specificity	0.6313	0.6063	0.6266	0.6594	0.6469	0.6469	0.6734	0.6563	**0.6922**	0.6750
Example based specificity	0.3935	**0.5363**	0.4839	0.1033	0.4726	0.4726	0.4617	0.4469	0.3522	0.4795
Hamming loss	0.3687	0.3938	0.3734	0.3406	0.3531	0.3531	0.3266	0.3438	**0.3078**	0.3250
Coverage	17.3125	17.9063	**16.5313**	17.5000	17.9688	17.9688	17.0313	17.2500	16.8125	17.2813
Error set size	31.7188	33.9688	**27.8125**	37.1875	31.7813	31.7813	28.5938	29.5313	28.6250	28.2188
Ranking loss	0.3532	0.3675	**0.2991**	0.4227	0.3612	0.3612	0.3225	0.3349	0.3346	0.3245
Iman&Davenport Ranking	7.25	8.5833	6.5833	5.75	6.5417	6.5417	3.0833	5.0833	**2.75**	2.8333

of data to sample (original paper used 63%), J48, a threshold of 0.5 and pruning labelsets that occurred less than p=3 times.

5. Results and discussion

This section presents and interprets the results obtained in the experimental study. It is divided into two different parts: the first one compares the results obtained by the different multi-label algorithms for making the predictions, while the second one presents a comparison between the framework proposed in this paper and another meta-learning framework not founded on the use of multi-label learning.

5.1. Comparison of multi-label classifiers

The results of the experimental study are shown in Table 9, where there is a row for the average results of each measure obtained by the multi-label algorithms. Values in bold indicate the algorithm that attains the best result for a specific measure. Note that, regarding the last four measures, the lower their value the better the performance of the classifier.

To analyze these results statistically, we proceeded by carrying out the Iman&Davenport multiple-comparison non-parametric test [55], comparing the average ranking obtained by the 10 algorithms over the 12 measures studied. The last row of Table 9 shows the ranking calculated for each multi-label algorithm. As can be observed, the lowest value corresponds to the EPS algorithm and, therefore, we can assert that this algorithm presents the best performance. In any case, further analysis should be carried out to find out whether the algorithms present significant differences in performance among themselves. For this purpose, it is necessary to check whether the test rejects the hypothesis of equivalence of means, i.e., if the computed value for the Iman&Davenport statistic for average results distributed according to the F-distribution does not belong to the critical interval at a given significance level. Assuming the $\alpha = 0.1$ significance level, the critical interval of the F-distribution is $C_0 = [0, (F_F)_{0.1,9,99} = 0, 1.696]$. The value of the statistic is 8.798, which exceeds the upper limit of the interval. Hence, there exist significant differences.

Specific differences can be detected by applying the Bonferroni-Dunn post-hoc statistical procedure. The critical difference of Bonferroni-Dunn considering the same level of significance $\alpha = 0.1$ is equal to 3.138. At this significance level, EPS obtains statistically better results than RA*k*EL, BRJ48, BPMLL, ML*k*NN and IBLR algorithms.

5.2. Comparison with the nearest-neighbor version

This section compares the framework proposed in this paper with a previous one [28]. As a result of the offline phase, the previous framework did not actually consider the creation of a meta-data set, but kept the meta-features and the algorithms ranking per data set separated. Then, in the online phase, to make a recommendation for a new data set, its meta-features were first extracted. Secondly, a nearest neighbor approach was used to find the closest data set in the repository to the new one, comparing its meta-features against those extracted from the Moodle data sets in the offline phase. And, finally, the subset of recommended algorithms for the new data set corresponded to those previously obtained for its nearest neighbor.

Since the previous approach recommended the set of algorithms of the nearest data set given a test instance, the information retrieval precision, recall and F-measure metrics were used. To carry out a fair comparison, then, we have restricted the study to these example-based measures. More specifically, we focused on the results obtained by the multi-label algorithms for these metrics, starting with the EPS algorithm, since it attains the highest performance, as proved in Section 5.1. We have also considered the results of the algorithm with the lowest ranking which does not present significant differences with EPS, which was the AdaBoost.MH algorithm. Moreover, we are interested in comparing the results of the multi-label approach considering the multi-label classifier which has the poorest performance, and hence we also show the results of the IBLR algorithm.

Figure 2 shows the notched boxplots obtained by the multi-label meta-learning framework, using either the EPS algorithm to extract the classifier, the AdaBoost.MH, or the IBLR algorithm, against those of the nearest-neighbor model. To assist in judging differences between sample medians, a notch can be used to show the 95% confidence interval for the median, given by $m \pm 1.58 \times IQR/\sqrt{n}$ [100].

As can be seen in Figure 2(a), the mean-value for the F-measure following the multi-label approach is considerably higher than that of the nearest-neighbor one. There are actually significant differences between the medians in the case of EPS and AB.MH algorithms, since the notches do not overlap. Moreover, even using IBLR to extract the classifier, which is the multi-label algorithm that performed most poorly, the result for this metric is better than using the nearest-neighbor approach.

As regards the precision metric, the mean is also higher in the multi-label approach, except for the AdaBoost.MH algorithm, as shown in Figure 2(b).

Finally, focusing on recall results of Figure 2(c), the medians for the multi-label approach using both EPS and AdaBoost.MH algorithms as classifiers seem to differ significantly, because their notches do not overlap with the notch of the nearest neighbor approach. Moreover, using the IBLR algorithm for extracting the classifier, these results also outperform those of the nearest neighbor methodology.

6. Concluding remarks

This paper proposed a multi-label meta-learning framework for classification algorithm recommendation in EDM. Roughly speaking, this framework makes use of MLL to generate a data set of meta-knowledge from a repository of single-label educational data sets. A multi-label classifier can then be trained from the data set generated, and then applied to new unseen data sets to recommend the best algorithms for them.

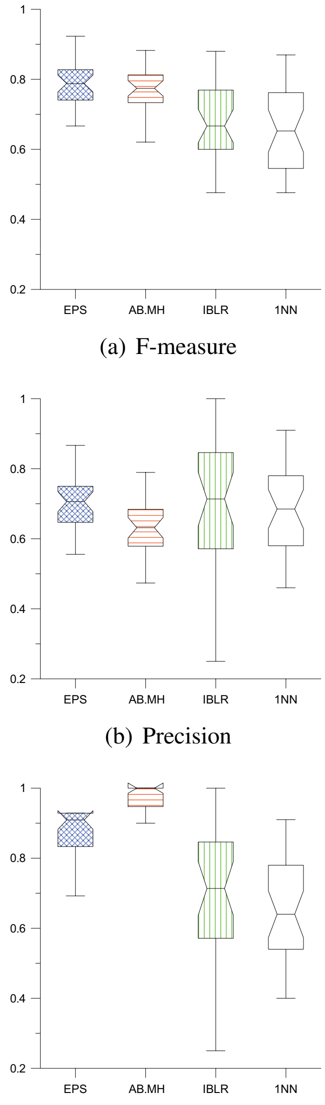
One of the contributions of this work is the way MLL is considered a principal component of the meta-learning framework. In addition, results obtained on a thorough experimental study using a repository of Moodle data sets proved the suitability of this framework. Actually, regardless of the multi-label algorithm employed for inducing the classifier, the new framework always outperforms a nearest-neighbour-based approach, which did not make use of MLL.

Educators could take greater advantage of this work if specific educational DM tools consider the inclusion of this framework, allowing them to apply it directly over their own repository of educational data sets.

Testing the effectiveness of the proposed framework with other types of educational data sets from environments such as intelligent tutoring systems, adaptive and hypermedia systems or massive open online courses would make for interesting future research. This would mean that results could be generalized to other domains than just learning management systems. In this sense, it could be interesting to consider including other domain specific measures to be extracted as meta-features.

Acknowledgements

This research is supported by projects of the Spanish Ministry of Science and Technology (TIN2014-55252-P), and the Deanship of Scientific Research (DSR), King Abdulaziz University, under grant No. (2-611-35-HiCi).



(a) F-measure

(b) Precision

(c) Recall

Figure 2: Box plots for a sample of n=32 data sets. The box bounds the IQR divided by the median, and Tukey-style whiskers extend to a maximum of 1.5 x IQR beyond the box.

References

1. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kauffman, 2006.
2. J. W. Lee and C. Giraud-Carrier, "Automatic selection of classification learning algorithms for data mining practitioners," *Intelligent Data Analysis*, vol. 17, no. 4, pp. 665–678, 2013.
3. D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, 1997.
4. W. Klosgen and J. Zytkow, *Handbook of data mining and knowledge discovery*. Oxford University Press, 2002.
5. A. A. Freitas, "Comprehensible classification models: A position paper," *SIGKDD Exploration Newsletter*, vol. 15, no. 1, pp. 1–10, 2014.
6. C. Romero, P. G. Espejo, A. Zafra, J. R. Romero, and S. Ventura, "Web usage mining for predicting final marks of students that use Moodle courses," *Computer Applications in Engineering Education*, vol. 21, no. 1, pp. 135–146, 2013.
7. A. A. Kardan and H. Sadeghi, "A decision support system for course offering in online higher education institutes," *International Journal of Computational Intelligence Systems*, vol. 6, no. 5, pp. 928–942, 2013.
8. C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," *Expert Systems with Applications*, vol. 33, no. 1, pp. 135 – 146, 2007.
9. M. Jovanovic, M. Vukicevic, M. Milovanovic, and M. Minovic, "Using data mining on student behavior and cognitive style data for improving e-learning systems: a case study," *International Journal of Computational Intelligence Systems*, vol. 5, no. 3, pp. 597–610, 2012.
10. N. Bhatt, A. Thakkar, and A. Ganatra, "A survey & current research challenges in meta learning approaches based on dataset characteristics," *International Journal of Soft Computing and Engineering*, vol. 2, no. 1, pp. 239–247, 2012.
11. E. Gibaja and S. Ventura, "Multi-label learning: a review of the state of the art and ongoing research," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 6, pp. 411–444, 2014.
12. E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, pp. 52:1–52:38, 2015.
13. R. E. Schapire and Y. Singer, "BoosTexter: A Boosting-based System for Text Categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.
14. K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label Classification of Music into Emotions," in *International Conference on Music Information Retrieval (ISMIR)*, 2008.
15. M.-L. Zhang and Z.-H. Zhou, "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 10, pp. 1338–1351, 2006.
16. K. Kawai and Y. Takahashi, "Identification of the Dual Action Antihypertensive Drugs Using TFS-Based Support Vector Machines," *Chem-Bio Informatics Journal*, vol. 4, pp. 44–51, 2009.
17. T. Sobol-Shikler and P. Robinson, "Classification of Complex Information: Inference of Co-Occurring Affective States from Their Expressions in Speech," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 7, pp. 1284–1297, 2010.
18. A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," in *ACM international conference on Web Search and Data Mining (ACM WSDM)*, pp. 173–182, ACM, 2012.
19. Y. Zhang, S. Burer, W. N. Street, K. Bennett, and E. Parrado-hern, "Ensemble Pruning Via Semidefinite Programming," *Journal of Machine Learning Research*, vol. 7, pp. 1315–1338, 2006.
20. A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," *Expert Systems with Applications*, vol. 41, no. 4, Part 1, pp. 1432 – 1462, 2014.
21. V. F. López, F. de la Prieta, M. Ogihara, and D. D. Wong, "A model for multi-label classification and ranking of learning objects," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8878 – 8884, 2012.
22. E. Özpolat and G. B. Akar, "Automatic detection of learning styles for an e-learning system," *Comput. Educ.*, vol. 53, pp. 355–367, 2009.
23. P. B. Brazdil, C. Soares, and J. P. da Costa, "Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results," *Machine Learning*, vol. 50, 2003.
24. L. Chekina, L. Rokach, and B. Shapira, "Meta-learning for selecting a multi-label classification algorithm," in *IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 220–227, 2011.
25. S. Y. Sohn, "Meta analysis of classification algorithms for pattern recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 11, pp. 1137–1144, 1999.
26. R. Vilalta, C. Giraud-Carrier, P. Brazdil, and C. Soares, "Using meta-learning to support data mining," *International Journal of Computer Science & Applications*, vol. 1, no. 1, pp. 31 – 45, 2004.
27. J. Kanda, A. Carvalho, E. Hruschka, and C. Soares, "Selection of algorithms to solve traveling salesman

problems using metalearning," *International Journal of Hybrid Intelligent Systems*, vol. 8, 2011.

28. C. Romero, J. L. Olmo, and S. Ventura, "A meta-learning approach for recommending a subset of white-box classification algorithms for Moodle datasets," in *International Conference on Educational Data Mining (EDM)*, pp. 268–271, 2013.
29. J. Lara, D. Lizcaino, M. Martinez, J. Pazos, and T. Riera, "A system for knowledge discovery in e-learning environments within the european higher education area - application to student data from Open University of Madrid, UDIMA," *Computers & Education*, vol. 72, no. 0, pp. 23 – 36, 2014.
30. C. Romero, M. Lopez, J. Luna, and S. Ventura, "Predicting students' final performance from participation in on-line discussion forums," *Computers & Education*, vol. 68, no. 0, pp. 458 – 472, 2013.
31. Z. Kovacic, "Predicting student success by mining enroment data," *Research in Higher Education Journal*, vol. 15, pp. 1–20, 2012.
32. S. Huang and N. Fang, "Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models," *Computers & Education*, vol. 61, pp. 133–145, 2013.
33. G. Zhang, T. J. Anderson, M. W. Ohland, and B. R. Thorndyke, "Identifying factors influencing engineering student graduation: A longitudinal and cross-institutional study," *Journal of Engineering Education*, vol. 93, no. 4, pp. 313–320, 2004.
34. S. Agarwal, G. N. Pandey, and M. D. Tiwari, "Data mining in education: Data classification and decision tree approach," *International Journal of e-Education, e-Business, e-Management and e-Learning*, vol. 2, no. 2, pp. 140–144, 2012.
35. B. Minaei-Bidgoli, D. Kashy, G. Kortemeyer, and W. Punch, "Predicting student performance: an application of data mining methods with an educational web-based system," in *ASEE/IEEE Frontiers in Education Conference*, pp. 13–18, 2003.
36. C. Márquez-Vera, A. Cano, C. Romero, and S. Ventura, "Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data," *Applied Intelligence*, vol. 38, no. 3, pp. 315–330, 2013.
37. J. L. Olmo, J. R. Romero, and S. Ventura, "Swarm-based metaheuristics in automatic programming: a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 6, pp. 445–469, 2014.
38. S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?," *Machine Learning*, vol. 54, no. 3, pp. 255–273, 2004.
39. R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, 2002.
40. D. H. Wolpert, "Stacked Generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.
41. M. Antenreiter, R. Ortner, and P. Auer, "Combining Classifiers for Improved Multilabel Image Classification," in *Proceedings of the 1st workshop on learning from multilabel data (MLD) held in conjunction with ECML/PKDD*, pp. 16–27, 2009.
42. S. Godbole and S. Sarawagi, "Discriminative Methods for Multi-Labeled Classification," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 22–30, 2004.
43. K. Kurach, K. Pawlowski, L. Romaszko, M. Tatjewski, A. Janusz, and H. S. Nguyen, "An ensemble approach to multi-label classification of textual data," in *Advanced Data Mining and Applications*, LNCS, pp. 306–317, Springer, 2012.
44. T. Li, C. Zhang, and S. Zhu, "Empirical Studies on Multi-label Classification," in *ICTAI '06: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pp. 86–92, 2006.
45. F. Markatopoulou, V. Mezaris, and I. Kompatsiaris, "A comparative study on the use of multi-label classification techniques for concept-based video indexing and annotation," in *Multimedia Modeling*, vol. 8325 of *LNCS*, pp. 1–12, Springer, 2014.
46. E. Montañes, R. Senge, J. Barranquero, J. R. Quevedo, J. J. del Coz, and E. Hüllermeier, "Dependent binary relevance models for multi-label classification," *Pattern Recognition*, vol. 47, no. 3, pp. 1494 – 1508, 2014.
47. F. Pachet and P. Roy, "Improving Multilabel Analysis of Music Titles: A Large-Scale Validation of the Correction Approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 335–343, 2009.
48. G. Tsoumakas, A. Dimou, E. Spyromitros, V. Mezaris, I. Kompatsiaris, and I. Vlahavas, "Correlation-Based Pruning of Stacked Binary Relevance Models for Multi- Label Learning," in *Proceedings of the 1st International Workshop on Learning from Multi-Label Data (MLD'09)* (G. Tsoumakas, M. Zhang, and Z. e. Zhou, eds.), pp. 101–116, 2009.
49. J. Kanda, C. Soares, E. Hruschka, and A. de Carvalho, "A meta-learning approach to select meta-heuristics for the traveling salesman problem using mlp-based label ranking," in *Proceedings of the 19th International Conference on Neural Information Processing - Volume Part III*, ICONIP'12, pp. 488–495, Springer, 2012.
50. J. Read, *Scalable Multi-label Classification*. PhD thesis, University of Waikato, Sept. 2010.
51. K. Bache and M. Lichman, "UCI machine learning repository," 2013.

52. M. D. M. Molina, C. Romero, S. Ventura, and J. M. Luna, "Meta-learning approach for automatic parameter tuning: A case of study with educational datasets," in *International Conference on Educational Data Mining (EDM)*, pp. 180–183, 2012.
53. M. Zorrilla and D. Garcia-Saiz, "Meta-learning: Can it be suitable to automatise the kdd process for the educational domain?," in *Rough Sets and Intelligent Systems Paradigms* (M. Kryszkiewicz, C. Cornelis, D. Ciucci, J. Medina-Moreno, H. Motoda, and Z. Ras, eds.), vol. 8537 of *Lecture Notes in Computer Science*, pp. 285–292, Springer International Publishing, 2014.
54. A. Zapata, V. Menendez, M. Prieto, and C. Romero, "Evaluation and selection of group recommendation strategies for collaborative searching of learning objects," *International Journal of Human-Computer Studies*, vol. 76, pp. 22–39, 2015.
55. J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
56. S. Natek and M. Zwilling *Expert Systems with Applications*, vol. 41, no. 14, pp. 6400 – 6407, 2014.
57. W. Rice, *Moodle 2.0 E-Learning Course Development*. Community experience distilled, Packt Publishing Ltd, 2011.
58. K. Cios, W. Pedrycz, R. Swiniarski, and L. Kurgan, *Data Mining: A Knowledge Discovery Approach*. Springer, 2010.
59. S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: a review of classification and combining techniques," *Artificial Intelligence Reviews*, vol. 26, pp. 159–190, 2006.
60. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.
61. R. Kohavi, "The power of decision tables," in *European Conference on Machine Learning (ECML)*, pp. 174–189, 1995.
62. M. Hall and E. Frank, "Combining naive bayes and decision tables," in *International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pp. 318–319, 2008.
63. W. Cohen, "Fast Effective Rule Induction," in *International Conference on Machine Learning (ICML)*, pp. 115–123, 1995.
64. B. Martin, "Instance-based learning: Nearest neighbor with generalization," Master's thesis, University of Waikato, New Zealand, 1995.
65. R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Machine Learning*, vol. 11, no. 1, pp. 63–90, 1993.
66. E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," in *International Conference on Machine Learning (ICML)*, pp. 144–151, 1998.
67. D. Richards, "Two decades of ripple down rules research," *The Knowledge Engineering Review*, vol. 24, pp. 159–184, 6 2009.
68. R. T. Jerome Friedman, Trevor Hastie, "Additive logistic regression : A statistical view of boosting," *Annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.
69. J. J. Oliver and D. Hand, "Averaging over decision stumps," in *Machine Learning: ECML-94* (F. Bergadano and L. Raedt, eds.), vol. 784 of *LNCS*, pp. 231–241, Springer, 1994.
70. J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
71. G. I. Webb, "Decision tree grafting from the all-tests-but-one partition," in *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 702–707, 1999.
72. G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, and M. Hall, "Multiclass alternating decision trees," in *European Conference on Machine Learning (ECML)*, pp. 161–172, 2002.
73. N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.
74. R. Kohavi, "Bayes rule based and decision tree hybrid classifier," 2001. US Patent 6,182,058.
75. L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
76. T. Elomaa and K. Kaariainen, "An analysis of reduced error pruning," *Journal of Artificial Intelligence Research*, vol. 15, pp. 163–187, 2001.
77. L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.
78. T. Pang-Ning, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison-Wesley, 2nd ed., 2014.
79. A. Cano, A. Zafra, and S. Ventura, "Weighted data gravitation classification for standard and imbalanced data," *Cybernetics, IEEE Transactions on*, vol. 43, no. 6, pp. 1672–1687, 2013.
80. T. Fawcett, "An introduction to roc analysis," *Pattern Recogniticon Letters*, vol. 27, no. 8, pp. 861–874, 2006.
81. "Shared domains of competence of approximate learning models using measures of separability of classes," *Information Sciences*, vol. 185, no. 1, pp. 43–65, 2012.
82. T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 3, pp. 289–300, 2002.
83. J. M. Sotoca, R. A. Mollineda, and J. S. Sánchez, "A meta-learning framework for pattern classification by means of data complexity measures," *Revista Iberoamericana de Inteligencia Artificial*, vol. 10, no. 29, pp. 31–38, 2006.

84. E. Leyva, A. Gonzalez, and R. Perez, "Knowledge-based instance selection: A compromise between efficiency and versatility," *Knowledge-Based Systems*, vol. 47, no. 0, pp. 65 – 76, 2013.
85. A. Hoekstra and R. Duin, "On the nonlinearity of pattern classifiers," in *Proceedings of the 13th International Conference on Pattern Recognition (ICPR)*, vol. 4, pp. 271–275, 1996.
86. L. Frank and E. Hubert, "Pretopological approach for supervised learning," in *Proceedings of the 13th International Conference on*, vol. 4, pp. 256–260 vol.4, 1996.
87. N. Macià, *Data complexity in supervised learning: A far-reaching implication*. PhD thesis, La Salle - Universitat Ramon Llull, October 2011.
88. G. Tsoumakas, I. Katakis, and I. Vlahavas, *Data Mining and Knowledge Discovery Handbook, Part 6*, ch. Mining Multi-label Data, pp. 667–685. Springer, 2010.
89. G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-Labelsets for Multi-Label Classification," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23(7), pp. 1079–1089, 2010.
90. Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," *Information Retrieval*, vol. 1, pp. 69–90, 1999.
91. Y. Yang and X. Liu, "A re-examination of text categorization methods," in *International SIGIR Conference on Research & Development on Information Retrieval*, pp. 42–49, 1999.
92. J. Pestian, C. Brew, P. Matykiewicz, D. Hovermale, N. Johnson, K. Cohen, and W. Duch, "A shared task involving multi-label classification of clinical free text," in *Proceedings of ACL BioNLP*, 2007.
93. R. E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-rated Predictions," *Machine Learning*, vol. 37(3), pp. 297 – 336, 1999.
94. K. Crammer and Y. Singer, "A family of additive online algorithms for category ranking," *Journal of Machine Learning Research*, vol. 3, pp. 1025–1058, 2003.
95. J. Read, B. Pfahringer, and G. Holmes, "Multi-label Classification Using Ensembles of Pruned Sets," in *IEEE International Conference on Data Mining (ICDM)*, vol. 0, pp. 995–1000, 2008.
96. K. Brinker, J. Fürnkranz, and E. Hüllermeier, "A Unified Model for Multilabel Classification and Ranking," in *European Conference on Artificial Intelligence (ECAI)*, pp. 489–493, 2006.
97. J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, pp. 1–27, 2011.
98. W. Cheng and E. Hüllermeier, "Combining instance-based learning and logistic regression for multilabel classification," *Machine Learning*, vol. 76, no. 2-3, pp. 211–225, 2009.
99. M.-L. Zhang and Z.-H. Zhou, "A k-Nearest Neighbor Based Algorithm for Multi-label Classification," in *IEEE International Conference on Granular Computing (IEEE GrC)*, vol. 2, pp. 718–721, 2005.
100. R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *The American Statistician*, vol. 32, no. 1, pp. 12–16, 1978.