Dynamic Ensemble Pruning Based on Multi-Label Classification

Fotini Markatopoulou[b], Grigorios Tsoumakas[a,*], Ioannis Vlahavas[a]

[a] *Dept. of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece*
[b] *Information Technologies Institute, CERTH, 57001 Thessaloniki, Greece*

Abstract

Dynamic (also known as instance-based) ensemble pruning, selects a (potentially) different subset of models from an ensemble during prediction based on the given unknown instance with the goal of maximizing prediction accuracy. This paper models dynamic ensemble pruning as a multi-label classification task, by considering the members of the ensemble as labels. Multi-label training examples are constructed by evaluating whether ensemble members are accurate or not on the original training set via cross-validation. We show that classification accuracy is maximized when learning algorithms that optimize example-based precision are used in the multi-label classification task. Results comparing the proposed framework against state-of-the-art dynamic ensemble pruning approaches in a variety of datasets using a heterogeneous ensemble of 200 classifiers, show that it leads to significantly improved accuracy.

Keywords: ensemble pruning, ensemble selection, multi-label classification, dynamic classifier fusion

1. Introduction

Supervised ensemble methods are concerned with the production and the combination of multiple predictive models. One dimension along which we could

*Corresponding author
Email addresses: markatopoulou@iti.gr (Fotini Markatopoulou), greg@csd.auth.gr (Grigorios Tsoumakas), vlahavas@csd.auth.gr (Ioannis Vlahavas)

categorize such methods is based on the number of models that affect the final decision. Usually all models are taken into consideration. When models are classifiers, this is called *classifier fusion*. Some methods, however, select just one model from the ensemble. When models are classifiers, this is called *classifier selection*. A third option, standing in between of these two, is to select a subset of the ensemble's models. This is mainly called *ensemble pruning* or *ensemble selection* (Tsoumakas et al., 2009).

Ensemble pruning methods can be either *static*, meaning that they select a fixed subset of the original ensemble for all test instances, or *dynamic*, also called *instance-based*, where a different subset of the original ensemble may be selected for each different test instance. The rationale of using dynamic ensemble pruning approaches is that different models have different areas of expertise in the instance space. Therefore, static approaches that are forced to select a fixed subset prior to seeing an unclassified instance may have a theoretical disadvantage compared to dynamic ones. On the other hand, static approaches lead to improved space complexity as they typically retain a small percentage of the original ensemble, in contrast to dynamic approaches that need to retain the complete original ensemble.

We propose a new approach to the instance-based ensemble pruning problem by modeling it as a multi-label learning task (Tsoumakas et al., 2010). Labels correspond to classifiers and multi-label training examples are formed based on the ability of each classifier to correctly classify each original training example. This way we can take advantage of recent advances in the area of multi-label learning and attack collectively, instead of separately, the problems of predicting whether each classifier will classify correctly a given unclassified instance. This paper builds upon our previous work (Markatopoulou et al., 2010) and extends it in the following main directions: a) it approximately doubles the number of datasets of the empirical comparison, providing further evidence of the effectiveness of the proposed algorithm and b) it employs a thresholding strategy that automatically computes the threshold that optimizes precision, leading to a fairer comparison against the state-of-the-art.

The rest of this paper is organized as follows. Section 2 reviews related work on dynamic ensemble pruning and Section 3 discusses the proposed approach. Section 4 presents the experimental setup. Section 5 presents the empirical study and Section 6 discusses the conclusions of this work.

2. Related Work

Many approaches deal directly with instance-based ensemble pruning (Tsymbal, 2000; Fan et al., 2002; Ko et al., 2007; Hernández-Lobato et al., 2009). These are presented in Section 2.2. However, there are also some that deal with dynamic approaches to classifier selection and fusion (Woods et al., 1997; Giacinto and Roli, 1997; Puuronen et al., 1999b; Ortega et al., 2001), which can be considered as extreme cases of ensemble pruning. In addition, some dynamic classifier selection approaches, may in some cases (e.g. ties) take into consideration more than one model. We therefore discuss in Section 2.1 such methods too. Section 2.3 discusses the issues of time complexity improvement and diversity in the context of dynamic ensemble pruning methods. For ease of reference and navigation of this section, Table 1 shows the category (i.e. selection, fusion, pruning) of each method that is discussed, using either its acronym where available (e.g. KNORA) or its citation.

2.1. Dynamic Classifier Selection and Fusion

The approach in (Woods et al., 1997) starts with retrieving the k nearest neighbors of a given test instance from the training set. It then classifies this test

Table 1: Summary of Dynamic Classifier Selection, Fusion and Pruning methods

Classifier Selection		DS, OLA, LCA, MCB, (Ortega et al., 2001), (Giacinto and Roli, 1997)
Ensemble Pruning	k-NN-based	DVS, KNORA, (Xiao et al., 2010)
	clustering-based	(Kuncheva, 2000)
	ordering-based	(Li et al., 2013), (Yan et al., 2013), (Hernández-Lobato et al., 2009), (Fan et al., 2002)
	other	(Lysiak et al., 2014)
Classifier Fusion		DV

instance using the most competent classifier within this local region. In case of ties, majority voting is applied. The local performance of classifiers is assessed using two different metrics. The first one, called *overall local accuracy* (OLA), measures the percentage of correct classifications of a model for the examples that exist in the local region. The second one, called *local class accuracy* (LCA), measures the percentage of correct classifications of a model within the local region too, but only for those examples where the model had given the same output as the one it gives for the current unlabeled instance being considered. A very similar approach to this one was proposed independently at the same time (Giacinto and Roli, 1997), also taking the distance of the k nearest neighbors into account.

The *dynamic selection* (DS) and *dynamic voting* (DV) approaches in (Puuronen et al., 1999b,a) are in the same spirit as (Woods et al., 1997; Giacinto and Roli, 1997). A kNN approach is initially used to find the most similar training instances with the given test instance. DS selects the classifier with the least error within the local area of the neighbors weighted by distance. In fact, DS, is very similar to the weighted version of OLA presented in (Giacinto and Roli, 1997). DV is different, as it is a classifier fusion approach. It combines all models weighted by their local competence.

Yet another approach along the same lines is (Giacinto and Roli, 2001). After finding the k nearest neighbors of the test instance, this approach further filters the neighborhood based on the similarity of the predictions of all models for this instance and each neighbor. In this sense, this approach is similar to the LCA variation in (Woods et al., 1997). It finally selects the most competent classifier in the reduced neighborhood. The predictions of all models for an instance, are in this paper called collectively *multiple classifier behavior* (MCB).

The approach in (Ortega et al., 2001) estimates whether the ensemble's models will be correct/incorrect with respect to a given test instance, using a learning algorithm, trained from the k-fold cross-validation performance of the models on the training set. It can be considered as a generalization of the approaches we have seen so far in this subsection, where a nearest neighbor approach was

specifically used instead. The approach we propose in this paper, is based on the same principle, with the difference that multi-label learning algorithms are employed and therefore the binary tasks of predicting correct/incorrect decision for each model are viewed in a collective way.

2.2. Dynamic Ensemble Selection

Similar with the dynamic Classifier Selection methods, the majority of the Dynamic Ensemble Selection methods start with retrieving the k nearest neighbors of a given test instance from the training set, in order to construct a new set of instances known as local region of competence (Xiao et al., 2010). The selection algorithms decide for the appropriate subset of the initial ensemble based on different properties (e.g. accuracy, diversity) of the base classifiers in this local region.

Dynamic Voting with Selection (DVS) (Tsymbal, 2000; Tsymbal and Puuronen, 2000) is an approach that stands in between the DS and DV algorithms that were mentioned in the previous subsection. First, about half of the models in the ensemble, those with local errors that fall into the upper half of the error range of the committee, are discarded. Then, the rest are combined using DV. Since this variation, eventually selects a subset of the original models, we can consider it as an instance-based ensemble pruning approach.

The primary goal of k-nearest-oracles (KNORA) (Ko et al., 2007) is improving the accuracy compared to the complete ensemble. Four different versions of the basic KNORA algorithm are proposed, all based on an initial stage of identifying the k nearest neighbors of a given unclassified instance. KNORA-ELIMINATE selects those classifiers that correctly classify *all* k neighbors. In case none such exists, the k value is decreased until at least one is found. KNORA-UNION selects those classifiers that correctly classify at least *one* of the k neighbors. KNORA-ELIMINATE-W and KNORA-UNION-W are variations that weight the votes of classifiers according to their Euclidean distance to the unclassified instance.

While the above methods only consider the accuracy of the ensemble within

the local region, the method proposed by (Xiao et al., 2010) simultaneously considers both the accuracy and the diversity of the pruned ensemble. Specifically, this method utilizes the symmetric regularity criterion to measure the accuracy of the ensemble and the double-fault measure to estimate the diversity. Finally, a GMDH-based neural network, describes the relationship between the class labels of the local region of competence and the test instance.

We can distinguish two more categories of dynamic ensemble selection methods that do not consider the k-nearest neighbors of test instances: Clustering based methods and Ordering based methods. Clustering based methods use clustering algorithms (k-means, Gaussian Mixture Models etc.) in order to estimate the local regions (Kuncheva, 2000). In contrast with the k nearest neighbors based methods that generate the local regions of competence during the test phase, clustering based methods estimate them offline during the training phase. Only the selection of a winning local region and the appropriate classifier ensemble is selected during the test phase.

Ordering based methods utilize statistical or probabilistic measures in order to produce a decreasing order of the base classifiers from the most suitable for a given test instance to the less suitable. The method proposed by (Li et al., 2013) assumes that base classifiers not only make a classification decision but also return a confidence score that shows their belief that their decision is correct. Dynamic ensemble selection is performed by ordering the base classifiers according to the confidence scores and fusion is performed using weighted voting. The method proposed by (Yan et al., 2013) is a two-step approach. In the first step classifiers are ordered based on their diversity using the Fleiss's statistic, in the second step classifiers in this rank are selected until a confidence threshold is reached.

Recently, a statistical approach has been proposed for instance-based pruning of homogeneous ensembles, with the provided that the models are produced via independent applications of a randomized learning algorithm on the same training data, and that majority voting is used (Hernández-Lobato et al., 2009). It is based on the observation that given the decisions made by the classifiers

that have already been queried, the probability distribution of the remaining class predictions can be calculated via a Polya urn model. During prediction, it samples the ensemble members randomly without replacement, stopping when the probability that the predicted class will change is below a specified level. As this approach assumes homogeneous models, it aims at speeding up the classification process without significant loss in accuracy. In contrast, our approach is directed at heterogeneous ensembles and aims primarily at improving the predictive performance compared to the full ensemble.

Another approach with the same goal as (Hernández-Lobato et al., 2009) is (Fan et al., 2002). In that work a first stage of static pruning takes place, followed up by a second stage of dynamic pruning, called dynamic scheduling. In the dynamic stage, classifiers are considered one by one in a decreasing order of total benefit (the authors focused on cost-sensitive applications). This iterative process stops when the difference of the current ensemble and the complete ensemble, as estimated assuming normal distribution with parameters calculated on the training set, is small.

Other approaches that cannot been included to any of the aforementioned categories have been proposed. For example, in (Lysiak et al., 2014) the dynamic selection problem is treated as an optimization problem and the proposed method uses the simulated annealing algorithm to solve it.

2.3. Time Complexity and Diversity

Since the output of dynamic ensemble pruning methods depends on just a subset of the models of the original ensemble, one would expect that there are always gains in terms of time complexity. However, this is not necessarily true. Some methods (e.g. LCA variant in (Woods et al., 1997)) first query all models and then select a subset of those to combine. Others might require computations that are more expensive compared to querying all models. For example, locating the 10 nearest neighbors of a new unclassified instance and then querying one decision tree, might be more expensive compared to directly querying 100 decision trees. Therefore, only computationally simple approaches

to pruning, such as (Hernández-Lobato et al., 2009) can lead to improved time complexity in the general case, sometimes at the expense of accuracy. The primary goal of most of the other methods is improving the accuracy compared to the complete ensemble. The proposed approach falls into this latter category of methods. It aims at improved accuracy and cannot guarantee time complexity improvements as it first needs to query a multi-label learner.

A potential point of criticism against most of the presented dynamic classifier combination methods, as well as against the proposed framework, could be that they are focused on accuracy and ignore diversity. The question is, should diversity be a desired property of the ensembles selected by instance-based ensemble pruning methods? The concept of diversity has been studied extensively (Kuncheva and Whitaker, 2003) and it is generally accepted that diversity and accuracy are both required for a successful ensemble. Diversity is desired, because models that err in different parts of the input space can help each other correct their mistakes through a voting process. In instance-based approaches however, we are interested in the performance for a specific example, not the whole input space in general, or a part of it. In this case we would want to choose only those models that are as accurate as possible for this example, and not those models that may make mistakes in it. Our approach is based solely on accuracy to select the appropriate subset of the ensemble. Note, however, that diversity is still a necessary property of the full ensemble.

3. Our Approach

The main idea of this work is that instance-based ensemble pruning can be modeled as a multi-label classification task, whose input space is the same as the input space of the classification task at hand and whose label space contains one label for each classifier in the ensemble. Given a new test instance, a multi-label classification model for this task would output a subset of classifiers, which is exactly what dynamic ensemble pruning is about. Formally, consider an ensemble of t models $h_j : \mathcal{X} \to \mathcal{Y}, j = 1 \dots t$, where $\mathcal{X} = \mathbb{R}^d$ is the input space

and $\mathcal{Y} = \{y_1, \ldots, y_k\}$ is the domain of the target variable of the classification task at hand. The multi-label classification task aims to learn a model h_m : $X \rightarrow h_1, \ldots, h_t$.

To construct a training set for this task, we should have knowledge of which of the classifiers in the ensemble are correct/incorrect for each training example of the classification task at hand. This could be achieved by letting each member of the ensemble classify each training example and comparing its output with the true class of the example. If the prediction of a classifier is correct then a positive class value is used for the corresponding label, while if the prediction is incorrect then a negative class value is used. As the predictions of a classifier on its training set are biased, we suggest the use of cross-validation for constructing the multi-label training set. Formally, consider a training set $D = (\vec{x}^{(i)}, y^{(i)}), i = 1 \ldots n$, where $\vec{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$, and the predictions $o_j^{(i)}$, obtained via cross-validation on D from the t different learning algorithms and/or different parameterizations of the same algorithms that were used to produce the models $h_j, j = 1 \ldots t$. The multi-label training set is then $M = (\vec{x}^{(i)}, Y^{(i)}), i = 1 \ldots n$, where $Y^{(i)} = \{I\left(o_j^{(i)} = y^{(i)}\right)\}, j = 1 \ldots t$, where $I(true) = 1$ and $I(false) = 0$. Figure 1 exemplifies the process of constructing the multi-label training set for the image segmentation dataset (segment) of the European project Statlog (Feng et al., 1993). The target attribute of this dataset has 6 values: *brickface*, *sky*, *foliage*, *cement*, *window*, *path*, *grass*.

Figure 1: Creating the multi-label training set

training set		classifier predictions				multi-label training set				
$\vec{x}$	y	o_1	o_2	$\ldots$	o_t	$\vec{x}$	o_1	o_2	$\ldots$	o_t
$\vec{x}^{(1)}$	sky	path	sky	$\ldots$	cement	$\vec{x}^{(1)}$	$-$	$+$	$\ldots$	$-$
$\vec{x}^{(2)}$	window	sky	window	$\ldots$	window	$\vec{x}^{(2)}$	$-$	$+$	$\ldots$	$+$
$\ldots$			$\ldots$						$\ldots$	
$\vec{x}^{(n)}$	foliage	foliage	grass	$\ldots$	path	$\vec{x}^{(n)}$	$+$	$-$	$\ldots$	$-$

Given an unlabeled instance, the multi-label classifier h_m is first queried,

outputting a subset of models $Z \in \{h_1, \ldots, h_t\}$ that it considers will correctly classify this instance. Note that the empty set is a feasible output for certain multi-label learning algorithms. However, in our case, an empty set is meaningless, as it means that all classifiers are pruned and that no prediction can be given. Therefore, for the proposed multi-label learning task, multi-label classifiers should be augmented with the constraint of non-empty predictions ($Z \neq \emptyset$). Then, each of the models in the subset is queried and their decisions are combined via *plurality* voting, often called *simple majority* voting. Let's denote the subset of classifiers that give correct predictions as W. Assuming a two-class classification task, the final prediction will be correct if more than half of the models in Z are correct, or in other words if $|Z \cap W|/|Z| > 0.5$. The left part of this inequality is actually the definition of the *precision* of a multi-label prediction (Godbole and Sarawagi, 2004; Tsoumakas et al., 2010). For problems with more than two class values, a lower precision value than 0.5 could in some cases suffice for a correct final decision through plurality voting. This analysis suggests that learning algorithms that optimize precision should be used for the proposed multi-label training task. We could therefore consider algorithms that minimize the following loss functions:

$$loss(Z, W) = \begin{cases} 0 & \text{if } |Z \cap W|/|Z| > 0.5 \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

$$loss(Z, W) = 1 - \frac{|Z \cap W|}{|Z|} \tag{2}$$

However, not many multi-label learning algorithms accept a specific loss function as a parameter. Furthermore, some multi-label learning algorithms learn models that can only output a score vector for each label (Crammer and Singer, 2003; Hüllermeier et al., 2008), so they cannot be used without modifications in our case, where a bipartition of the labels into relevant (correct classifiers) and irrelevant (incorrect classifiers) is required. In addition, most state-of-the-art multi-label classification algorithms (i.e those that do output a bipartition) of the recent literature (Zhang and Zhou, 2006; Tsoumakas et al.,

2011; Read et al., 2008, 2009), actually learn models that output a score vector primarily and employ a thresholding method in order to be able to output bipartitions (Ioannou et al., 2010). Given that high precision is crucial for the success of our approach, all the above learning algorithms could be used in conjunction with a thresholding method that optimizes the losses presented above.

4. Experimental Setup

4.1. Datasets

We experimented on 40 data sets from the UCI Machine Learning repository (Asuncion and Newman, 2007). Table 2 presents the details of these data sets.

4.2. Ensemble Construction

We constructed a heterogeneous ensemble of 200 models, by using different learning algorithms with different parameters on the training set. The WEKA machine learning library (Witten and Frank, 2005) was used as the source of learning algorithms. We trained 40 multilayer perceptrons (MLPs), 60 k Nearest Neighbors (kNNs), 80 support vector machines (SVMs) and 20 decision trees (DT) using the C4.5 algorithm. The different parameters used to train the algorithms were the following (default values were used for the rest of the parameters):

- MLPs: we used 5 values for the nodes in the hidden layer $\{1, 2, 4, 8, 16\}$, 4 values for the momentum term $\{0.0, 0.2, 0.5, 0.9\}$ and 2 values for the learning rate $\{0.3, 0.6\}$.
- kNNs: we used 20 values for k distributed evenly between 1 and the plurality of the training instances. We also used 3 weighting methods: no-weighting, inverse-weighting and similarity-weighting.
- SVMs: we used 8 values for the complexity parameter $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 100\}$, and 10 different kernels. We used 2 polynomial kernels (of degree 2 and 3) and 8 radial kernels (gamma $\in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2\}$).

Table 2: Details of data sets: Folder in UCI server, number of instances, classes, continuous and discrete attributes, percentage of missing values

id	UCI Folder	Instances	Classes	Continuous	Discrete	Missing
d1	anneal	798	6	9	29	0.00
d2	arrhythmia	452	16	206	73	0.32
d3	audiology	226	24	0	69	20.33
d4	autos	205	7	15	10	11.51
d5	balance-scale	625	3	4	0	0.00
d6	breast-cancer	286	2	0	9	0.35
d7	breast-w	699	2	0	2	0.00
d8	car	1728	4	0	6	0.00
d9	cmc	1473	3	2	7	0.00
d10	colic	368	2	7	15	23.80
d11	credit-a	690	2	6	9	0.65
d12	credit-g	1000	2	7	13	0.00
d13	dermatology	366	6	1	33	0.00
d14	diabetes	768	2	8	0	0.00
d15	ecoli	336	8	7	0	0.00
d16	eucalyptus	736	5	14	5	3.20
d17	flags	194	8	2	27	0.00
d18	glass	214	7	9	0	0.00
d19	haberman	306	2	3	0	0.00
d20	heart-c	303	5	6	7	0.18
d21	heart-h	294	5	6	7	20.46
d22	heart-statlog	270	2	13	0	0.00
d23	heart-v	200	5	6	7	26.85
d24	hepatitis	155	2	6	13	5.67
d25	hill	607	2	100	0	0.00
d26	hypothyroid	3772	4	7	30	5.4
d27	ionosphere	351	2	34	0	0.00
d28	kr-vs-kp	3196	2	0	36	0.00
d29	liver-disorders	345	2	6	0	0.00
d30	lymph	148	4	3	15	0.00
d31	primary-tumor	339	2	0	17	0.00
d32	segment	2310	7	19	0	0.00
d33	sick	3772	2	7	23	5.40
d34	sonar	195	2	60	0	0.00
d35	soybean	683	19	0	35	0.00
d36	tic-tac-toe	958	2	0	9	0.00
d37	vehicle	846	4	18	0	0.00
d38	vote	435	2	0	16	5.63
d39	wine	178	3	13	0	0.00
d40	zoo	101	7	1	16	0.00

- Decision trees: We constructed 10 trees using post-pruning with 5 values for the confidence factor {0.1, 0.2, 0.3, 0.5 } and 2 values for Laplace smoothing {true, false}, 8 trees using reduced error pruning with 4 values for the number of folds {2, 3, 4, 5} and 2 values for Laplace smoothing {true, false}, and 2 unpruned trees using 2 values for the minimum objects per leaf {2, 3}.

4.3. Dynamic Ensemble Pruning Methods

We compare the proposed approach against most of the methods that were presented in Section 2 and deal with the issue of accuracy improvement: OLA, LCA, MCB, DV, DS, DVS and KNORA. These methods start by finding the k nearest neighbors of a test instance in the training set. We set the number of neighbors to 10 and used Euclidean distance to estimate the nearest neighbors. The ELIMINATE version of KNORA was used, as that was found as producing the overall best results in (Ko et al., 2007). We also calculate the performance of the complete ensemble of 200 models using simple majority voting for model combination (MV) as a baseline method.

We did not compare the proposed approach against (Hernández-Lobato et al., 2009; Fan et al., 2002), which are methods with the different goal of speeding up the classification process without significant loss in accuracy. As our approach is based solely on accuracy to select the appropriate subset of the ensemble, we further refrained from comparing against approaches that consider diversity as well (Lysiak et al., 2014; Xiao et al., 2010; Yan et al., 2013). This would be an interesting future work direction. Finally, as we already mention in Section 2, the approach by Ortega et al. Ortega et al. (2001), is a generalized version of OLA: In OLA, kNN is used, while in (Ortega et al., 2001) any learning algorithm could be used. In this sense, we are comparing to that work too.

We instantiate the proposed approach with two multi-label learning algorithms. The first one is ML-kNN (Zhang and Zhou, 2007), which was selected because it follows an instance-based approach, similarly to the rest of the competing methods, apart from MV. The number of neighbors is also set to 10 here,

which happens to be the default setting too. The second one is Calibrated Label Ranking (CLR) (Fürnkranz et al., 2008), a multi-label learner that is known to achieve high precision. The C4.5 learning algorithm with default settings is used for the binary classification tasks considered by CLR.

Although the above methods do output bipartitions, they primarily output a score vector containing one score for each label. ML-kNN's scores are probability estimates and it uses a 0.5 threshold to output the bipartition. CLR uses an artificial calibration label to break the ranking of the labels and produce the bipartition. Apart from using these default methods to obtain the bipartition, we also pair the algorithms with a simple thresholding strategy, called *OneThreshold*, which considers a label as positive if its score is higher than a single threshold t used for all labels (Ioannou et al., 2010).

4.4. Evaluation Methodology

We use 10-fold cross validation to evaluate the different approaches. For each of the 10 train/test splits, we perform an inner 10-fold cross-validation on the training set in order to gather meta-data concerning the performance of the algorithms, which are required by all dynamic ensemble pruning approaches. Threshold tuning is again based only on the training data, so at no point are test data seen by any of the approaches.

5. Experiments

Section 5.1 investigates how thresholding affects the precision of the multi-label learners and consequently the accuracy of the proposed approach. Then, Section 5.2 investigates the relationship of the optimal threshold for the proposed approach with four dataset properties (difficulty (via the accuracy of our approach), size in samples, missing values (percentage) and number of classes). Section 5.3 presents the results of comparing the proposed approach with other state-of-the-art approaches. Section 5.4 comments on the number and type of selected models for the competitive dynamic selection approaches. Finally, Sec-

tion 5.5 examines the relalionship of the relative performance of the proposed method on different dataset sizes.

5.1. Thresholding the Multi-Label Learners

Tables 3 and 4 present the accuracy and standard deviation of the framework instantiated with CLR and ML-kNN respectively, for all datasets (column 1) under the following thresholding schemes: a) their default thresholding strategy (column 2), b) OneThreshold with threshold values ranging from 0.5 to 0.9 using a step of 0.05 (columns 3 to 11), and c) OneThreshold using 5-fold cross-validation to automatically compute the threshold value that optimizes the loss function in Equation 2 (column 12). The last column shows the actual threshold value that was selected. The highest accuracy at each dataset is underlined. As suggested in (Demsar, 2006), we base the discussion of the results on the average ranks of the different versions of the proposed approach (last row of the tables).

For CLR, we notice that threshold values higher than 0.55 lead to better average ranks compared to the default thresholding strategy. Higher threshold values lead to the greatest improvement, with 0.80, 0.85 and 0.90 being the three values with the best performance compared to the rest. This is in line with the analysis in Section 3, as higher thresholds improve the precision of the learners. Automatically selecting a threshold via cross-validation performs relatively well, but not better than a fixed high threshold.

As already discussed, the default version of ML-kNN uses an implicit 0.5 threshold to obtain the bipartition. Therefore the first two columns are identical. We notice that all threshold values examined lead to better accuracy compared to the standard threshold. Again, higher thresholds lead to better results, but the best performance is achieved for a threshold value of 0.75. As in the case of CLR, automatically selecting a threshold via cross-validation performs relatively well, but not better than the fixed threshold values of 0.75, 0.85 and 0.90.

The general conclusion is that tuning the threshold improves the performance of the multi-label learners. This comes from the improvement of the precision of the multi-label learners, which in turn comes from the fact that

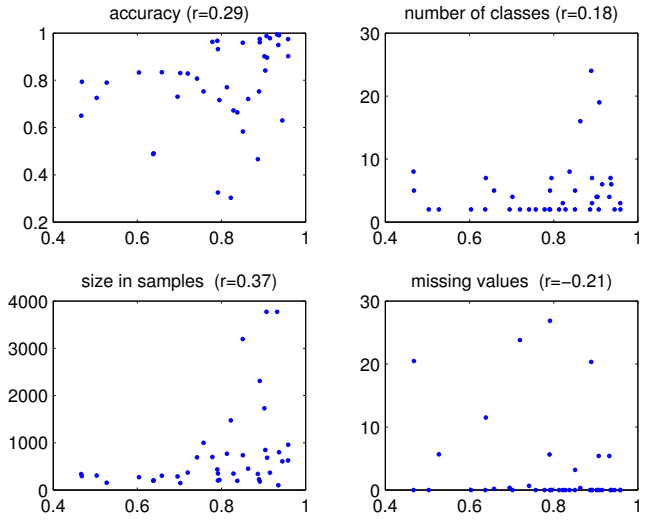

Figure 2: Scatterplots and correlation coefficients for the optimal threshold of our approach instantiated with CLR. The x axis corresponds to the optimal threshold and the y axis to the four different dataset properties.

the learners are forced to select less classifiers, but classifiers that we are more confident about their correctness. This assumes correct probability estimations and hence correct ranking of the classifiers. This of course is not always the case. That is why we do not see the accuracy of our approach monotonically increasing with the threshold values.

5.2. Relationship of Optimal Threshold with Dataset Properties

Next we investigate whether the optimal threshold of our approach is correlated with properties of the given dataset, such as its difficulty (via the accuracy of our approach), size in samples, missing values (percentage) and number of classes. Figures 2 and 3 show scatterplots contrasting the optimal threshold and each of the four dataset properties mentioned above for our approach instantiated with CLR and ML-kNN respectively.

No strong relationships are noticed, with the exception of a 0.85 correlation between the accuracy of our approach using ML-kNN and the optimal threshold.

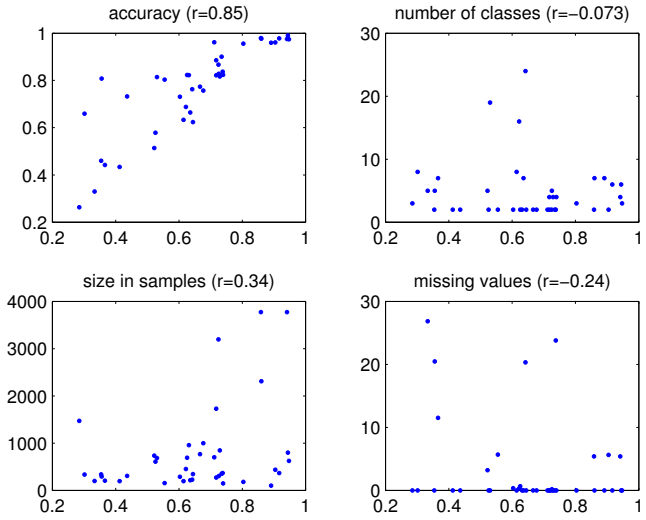

Figure 3: Scatterplots and correlation coefficients for the optimal threshold of our approach instantiated with ML-kNN. The x axis corresponds to the optimal threshold and the y axis to the four different dataset properties.

This means that the easier the learning task (i.e. the higher the accuracy of our approach), the higher the optimal threshold of our approach. This suggests that ML-kNN's confidence in the accuracy of the ensemble's members is a bit optimistic in easier datasets, which is not surprising. This is less pronounced in the case of CLR with a 0.29 correlation. Easier problems means many more positive examples per label than negative ones, e.g. a class imbalance which is opposite to the typical one found in multi-label learning problems, where the positive examples are scarce. We hypothesize that CLR is better at handling this class imbalance compared to ML-kNN and does not require setting a much higher threshold.

In conclusion, the optimal threshold does not appear to be predictable from dataset properties in the general case. We therefore suggest tuning the threshold via a validation set.

5.3. Accuracy

Table 5 shows the accuracy percentage and standard deviation of the competing methods in all datasets. The highest accuracy at each dataset is underlined. We, again, start the analysis of the results based on the average ranks, as suggested in (Demsar, 2006). We notice that the best overall results are achieved by the proposed approach using CLR. This is in accordance with our prior knowledge that CLR is an algorithm that achieves good precision performance. On the other hand, using ML-kNN led to slightly worse results and an overall rank of 4.1. Almost the same performance was achieved by OLA with an average rank of 4.2, while close to this were also MCB and KNORA. DV and DVS did not perform that well, while the worst results were that of the static classifier fusion approach of majority voting. Note that automatic thresholding is used in these experiments by our approach for a fair comparison with competing approaches.

We also comment the results in terms of the wins, ties and losses for each pair of algorithms, which are shown in Table 6. We refrain from reporting only the statistically significant wins and losses, following the recommendation in (Demsar, 2006). We notice again the high performance of CLR, which loses in 14 datasets maximum (35% of all datasets) from any method. Notable is the fact that this maximum number of losses happens only when CLR is compared with our second approach, ML-kNN. ML-kNN presents high performance too, losing in 26 datasets maximum from any method. Once again the main competitor seems to be the OLA algorithm.

We also performed the Wilcoxon signed ranks test between our approach using CLR and its main competitor, namely OLA. The test gave a p-value of 0.0022, indicating that our approach is significantly better than OLA at a 99% level.

5.4. Number and Type of Selected Models

Table 7 shows the average number of models selected by our approach (with CLR and ML-kNN), KNORA and DVS across all test instances for all datasets.

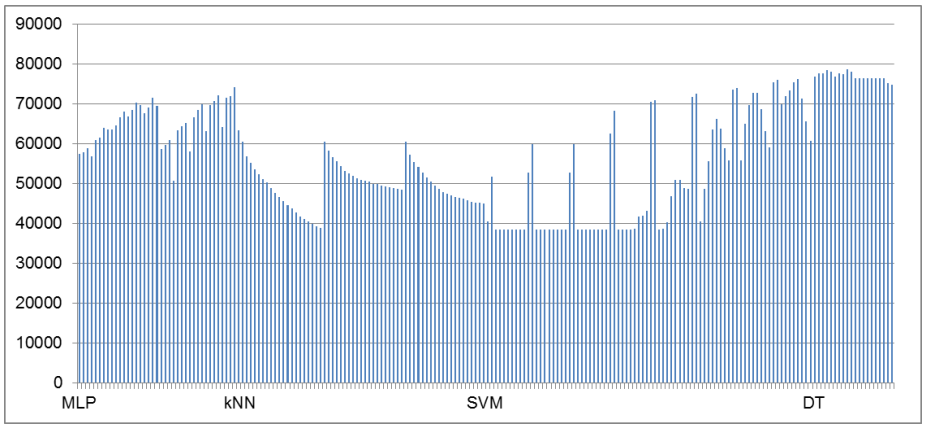

Figure 4: Model frequencies.

The mean across all datasets can be seen in the last row of the table. The lowest mean number of selected models, 36, is attained by our approach with CLR. It is therefore clear that one of the reasons for the success of CLR within our approach is the selection of a small number of accurate models. ML-kNN on the other hand outputs a larger number of models, 113 on average, which justifies its lower performance. The average number of models selected by KNORA is 85, while that of DVS is even larger, 125, justifying its respective lower performance.

Figure 4 shows the frequency of selection of each member of the ensemble across all datasets and test examples. We notice that decision tree models are mostly selected, followed by good versions of SVMs (the higher the cost parameter, the better) and good versions of MLPs (the larger the number of hidden layer nodes, the better). kNNs are selected with lower frequency and we notice that the lower the number of nearest neighbours the better the results.

5.5. Relationship of Improvement with Respect to Dataset Size

In this section we investigate if the relative performance of our approach is correlated with the data set size in samples. The relative performance of our approach is measured via its rank in each of the 40 different datasets when compared to the other nine competitive methods, presented on table 5. Figure

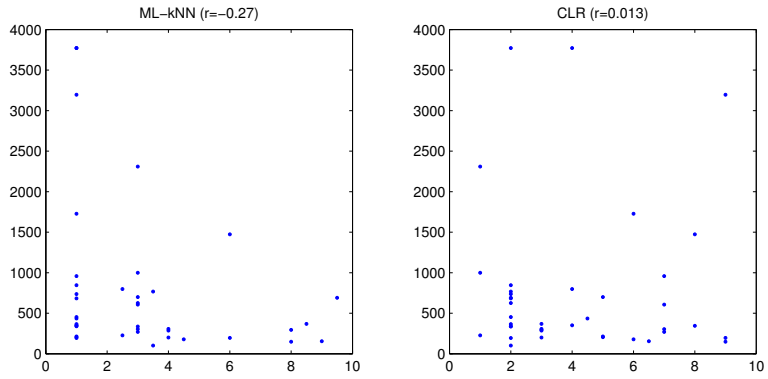

Figure 5: Scatterplots and correlation coefficients for the optimal threshold of our approach instantiated with ML-kNN (left) and CLR (right). The x axis corresponds to the rank of our method and the y axis to the dataset size in samples.

5 shows scatter plots contrasting the rank and the dataset size for our approach instantiated with ML-kNN (left) and CLR (right) respectively.

No relationship can be noticed with respect to our approach using CLR. On the other hand, a weak negative correlation is noticed when using ML-kNN. This means that the larger the dataset size, the better the performance of our approach using ML-kNN (the smaller the rank of the proposed method). This finding is justified as follows. ML-kNN searches for the nearest neighbors of a test instance in the training set. A larger training set represents the real data distribution more accurately and the algorithm is able to generalize better to unknown instances.

6. Conclusions and Future work

This paper presented a new approach for instance-based ensemble pruning based on multi-label learning. Each classifier of an ensemble corresponds to a label and a multi-label model is learned to output the *correct* classifiers for a given instance. We have shown theoretically that in order to achieve a correct prediction after a plurality voting process among the subset of classifiers output by the multi-label model, this model should exhibit high example-based

precision.

We instantiated this framework with two multi-label learning algorithms and investigated the effect of thresholding on the accuracy of the framework. As expected, the use of higher thresholds leads to improved precision and higher accuracy. We further investigated the use of a strategy that automatically determines an appropriate threshold that optimizes the precision of the models.

The performance of the proposed framework was compared with a variety of state-of-the-art competing methods. Based on the results, we reached the conclusion that the proposed framework performs significantly better in the large collection of classification datasets analyzed in this work. We would also like to stress that the performance of the framework depends on the multi-label learner and the thresholding strategy used. Therefore, it has the potential to achieve even better results than those presented here, which are based on the CLR and ML-kNN algorithms and the OneThreshold strategy.

We also investigated if the optimal threshold of our approach is correlated with properties of the given dataset, such as its difficulty (via the accuracy of our approach), size in samples, missing values (percentage) and number of classes. No strong relationships are noticed, with the exception of a strong correlation between the accuracy of our approach using ML-kNN and the optimal threshold.

In our future work it would be interesting to compare the proposed approach against methods that simultaneously consider both the diversity and the accuracy of the pruned ensemble. In addition, we would like to examine the benefit of our approach against popular homogeneous ensemble method, like boosting or random forests. However, for such an experiment to be fair we would also be adding the random forest in our heterogeneous ensemble of models, and hence we expect our framework to do better than random forest by itself. Furthermore, we would like to examine the benefit of the proposed approach on homogeneous ensembles. Finally, we would like to verify the conclusion of previously published papers that dynamic selection methods perform better than static selection methods by comparing our approach with methods that statically select subsets from the initial ensemble or statically select a single best model from

the initial ensemble. Such experiments could also reveal what properties make dynamic selection methods perform better than their static counterparts.

Acknowledgements

We would like to thank the anonymous reviewers for their useful comments and suggestions.

References

Asuncion, A., Newman, D., 2007. UCI machine learning repository. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Crammer, K., Singer, Y., 2003. A family of additive online algorithms for category ranking. Journal of Machine Learning Research 3, 1025–1058.

Demsar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30.

Fan, W., Chu, F., Wang, H., Yu, P. S., 2002. Pruning and dynamic scheduling of cost-sensitive ensembles. In: Eighteenth national conference on Artificial intelligence. American Association for Artificial Intelligence, pp. 146–151.

Feng, C., Sutherland, A., King, R., Muggleton, S., Henery, R., 1993. Comparison of machine learning classifiers to statistics and neural networks. In: Proceedings of the Third International Workshop in Artificial Intelligence and Statistics. pp. 41–52.

Fürnkranz, J., Hüllermeier, E., Mencia, E. L., Brinker, K., 2008. Multilabel classification via calibrated label ranking. Machine Learning 73 (2), 133–153.

Giacinto, G., Roli, F., 1997. Adaptive selection of image classifiers. In: Proc. 9th International Conference on Image Analysis and Processing (ICIAP'97) - Volume I. pp. 38–45.

Giacinto, G., Roli, F., 2001. Dynamic classifier selection based on multiple classifier behavior. Pattern Recognition 34 (9), 1879–1881.

Godbole, S., Sarawagi, S., 2004. Discriminative methods for multi-labeled classification. In: Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2004). pp. 22–30.

Hernández-Lobato, D., Martínez-Muñoz, G., Suárez, A., 2009. Statistical instance-based pruning in ensembles of independent classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2), 364–369.

Hüllermeier, E., Fürnkranz, J., Cheng, W., Bringer, K., 2008. Label ranking by learning pairwise preferences. Artificial Intelligence.

Ioannou, M., Sakkas, G., Tsoumakas, G., Vlahavas, I., 2010. Obtaining bipartitions from score vectors for multi-label classification. In: Proceedings 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2010). IEEE Computer Society, Los Alamitos, CA, USA, pp. 409–416.

Ko, A. H., Sabourin, R., Alceu Souza Britto, J., 2007. From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition 41, 1718–1731.

Kuncheva, L., 2000. Clustering-and-selection model for classifier combination. In: Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on. Vol. 1. pp. 185–188.

Kuncheva, L. I., Whitaker, C. J., May 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning 51 (2), 181–207.

Li, L., Zou, B., Hu, Q., Wu, X., Yu, D., 2013. Dynamic classifier ensemble using classification confidence. Neurocomputing 99 (0), 581 – 591.

Lysiak, R., Kurzynski, M., Woloszynski, T., 2014. Optimal selection of ensemble classifiers using measures of competence and diversity of base classifiers. Neurocomputing 126 (0), 29 – 35.

Markatopoulou, F., Tsoumakas, G., Vlahavas, I., 2010. Instance-based ensemble pruning via multi-label classification. In: Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on. Vol. 1. pp. 401 –408.

Ortega, J., Koppel, M., Argamon, S., 2001. Arbitrating among competing classifiers using learned referees. Knowledge and Information Systems 3 (4), 470–490.

Puuronen, S., Terziyan, V. Y., Tsymbal, A., 1999a. A dynamic integration algorithm for an ensemble of classifiers. In: Ras, Z. W., Skowron, A. (Eds.), ISMIS. Vol. 1609 of Lecture Notes in Computer Science. Springer, pp. 592–600.

Puuronen, S., Terziyan, V. Y., Tsymbal, A. K. A., 1999b. Dynamic integration of multiple data mining techniques in a knowledge discovery management system. In: Proc. SPIE Conference on Data Mining and Knowledge Discovery. pp. 128–139.

Read, J., Pfahringer, B., Holmes, G., 2008. Multi-label classification using ensembles of pruned sets. In: Proc. 8th IEEE International Conference on Data Mining (ICDM'08). pp. 995–1000.

Read, J., Pfahringer, B., Holmes, G., Frank, E., 2009. Classifier chains for multi-label classification. In: Proc. 20th European Conference on Machine Learning (ECML 2009). pp. 254–269.

Tsoumakas, G., Katakis, I., Vlahavas, I., 2010. Mining multi-label data. In: Maimon, O., Rokach, L. (Eds.), Data Mining and Knowledge Discovery Handbook, 2nd Edition. Springer, Ch. 34, pp. 667–685.

Tsoumakas, G., Katakis, I., Vlahavas, I., 2011. Random k-labelsets for multi-label classification. IEEE Transactions on Knowledge and Data Engineering 23, 1079–1089.

Tsoumakas, G., Partalas, I., Vlahavas, I., 2009. An ensemble pruning primer. In: Okun, O., Valentini, G. (Eds.), Supervised and Unsupervised Methods and

Their Applications to Ensemble Methods (SUEMA 2009). Springer Verlag, pp. 1–13.

Tsymbal, A., 2000. Decision committee learning with dynamic integration of classifiers. In: Proc. 2000 ADBIS-DASFAA Symposium on Advances in Databases and Information Systems,.

Tsymbal, A., Puuronen, S., 2000. Bagging and boosting with dynamic integration of classifiers. In: Proc. 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2000). Lyon, France, pp. 195–206.

Witten, I. H., Frank, E., 2005. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Woods, K., Kegelmeyer, Jr., W. P., Bowyer, K., 1997. Combination of multiple classifiers using local accuracy estimates. IEEE Trans. Pattern Anal. Mach. Intell. 19 (4), 405–410.

Xiao, J., He, C., Jiang, X., Liu, D., 2010. A dynamic classifier ensemble selection approach for noise data. Information Sciences 180 (18), 3402 – 3421.

Yan, Y., Yin, X.-C., Wang, Z.-B., Yin, X., Yang, C., Hao, H.-W., 2013. Sorting-Based Dynamic Classifier Ensemble Selection. In: 12th International Conference on Document Analysis and Recognition. IEEE, pp. 673–677.

Zhang, M.-L., Zhou, Z.-H., 2006. Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18 (10), 1338–1351.

Zhang, M.-L., Zhou, Z.-H., 2007. Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition 40 (7), 2038–2048.

Table 3: Accuracy percentage and standard deviation of our approach instantiated with CLR and the following thresholding strategies: a) default, b) OneThreshold using thresholds ranging from 0.5 to 0.9 with a step of 0.05 and using 5-fold cross-validation to automatically compute the threshold that optimizes precision.

id	default	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	optimized result	optimized threshold
d1	98.78±00.92	98.89±00.99	99.22±00.71	99.22±00.71	99.33±00.74	99.33±00.74	99.56±00.74	99.33±00.74	99.11±00.67	99.22±00.71	99.11±00.97	0.94±00.04
d2	69.71±07.41	60.43±09.77	62.65±09.52	65.75±09.16	67.06±09.00	68.40±09.36	69.71±07.38	71.04±07.15	71.91±04.59	71.03±06.01	72.13±05.33	0.86±00.01
d3	76.64±15.07	71.82±12.47	74.01±12.39	74.45±12.93	77.10±12.38	77.96±11.64	77.97±13.45	78.42±12.66	75.34±15.73	74.45±15.97	75.32±16.38	0.89±00.03
d4	44.64±19.23	47.60±24.30	47.64±24.18	48.60±24.25	48.12±23.75	48.60±23.87	48.62±22.87	48.17±22.21	48.67±20.75	48.12±18.69	49.10±23.09	0.64±00.10
d5	87.05±03.32	88.97±03.10	88.81±03.23	88.97±03.26	89.29±03.53	89.93±03.81	91.06±04.20	93.78±04.57	96.33±03.11	97.13±03.16	97.44±02.77	0.96±00.01
d6	74.85±06.78	74.85±06.96	74.50±07.22	74.14±06.91	73.78±07.13	72.72±07.41	72.39±06.98	71.70±07.61	72.04±06.89	72.75±07.37	73.07±07.15	0.70±00.07
d7	96.14±04.29	96.57±03.68	96.28±03.74	96.28±03.74	96.28±03.74	96.28±03.68	96.14±03.62	96.28±03.68	96.28±03.68	96.57±03.51	96.28±03.68	0.78±00.08
d8	83.10±06.31	83.90±08.50	84.77±08.76	84.89±09.14	84.66±09.31	85.35±08.85	86.51±08.46	87.50±08.49	88.08±08.10	88.60±06.99	90.16±06.57	0.90±00.04
d9	27.72±19.25	26.56±21.88	27.38±22.01	28.19±21.89	29.28±21.74	29.82±20.99	30.98±20.45	30.91±20.32	31.86±19.11	32.75±19.14	30.30±20.93	0.82±00.07
d10	85.34±05.00	84.53±05.78	83.99±06.76	83.43±06.55	83.69±06.42	83.69±05.71	84.23±06.19	83.68±07.01	84.76±05.80	83.67±06.91	82.87±06.21	0.72±00.08
d11	81.45±17.69	80.87±17.80	81.16±17.77	81.02±17.76	81.02±18.14	81.02±18.47	81.16±18.48	81.16±18.45	81.01±18.35	81.30±18.62	80.73±18.50	0.74±00.10
d12	74.70±04.34	75.00±04.56	74.70±04.69	75.00±05.42	75.30±05.60	75.20±05.42	75.10±05.36	75.90±05.59	74.90±05.91	75.30±05.22	75.30±05.60	0.76±00.04
d13	96.44±03.04	97.00±01.89	97.28±02.10	97.55±01.90	97.56±01.89	97.56±01.89	97.56±01.89	97.56±01.89	97.55±01.90	98.10±01.73	97.83±02.03	0.92±00.03
d14	75.65±07.64	77.21±06.33	77.20±06.32	76.94±05.71	77.34±06.10	77.20±05.59	76.82±05.71	77.47±05.72	76.30±05.55	75.91±05.44	77.08±05.92	0.81±00.04
d15	65.02±33.45	65.04±33.62	65.94±31.98	67.76±30.29	68.37±29.00	69.88±27.39	68.07±28.47	67.18±28.54	68.07±28.47	69.28±27.33	65.04±33.78	0.47±00.12
d16	53.85±11.89	56.31±14.19	57.13±14.84	56.58±14.37	56.72±14.02	57.26±14.47	57.52±14.05	56.97±14.55	56.83±13.44	58.47±12.77	58.33±13.04	0.85±00.03
d17	63.84±09.30	62.82±08.62	62.84±06.29	63.37±07.27	62.84±07.45	63.87±07.93	66.45±07.90	67.47±07.82	66.42±10.11	65.90±10.13	66.45±10.05	0.84±00.04
d18	74.35±10.73	68.79±11.11	69.74±10.46	71.15±11.40	74.87±11.36	73.46±11.08	72.55±11.63	71.62±11.78	70.67±12.33	70.17±12.58	71.62±12.35	0.79±00.03
d19	72.90±08.47	72.89±08.99	72.89±08.99	72.56±08.78	73.23±08.93	72.24±08.64	72.25±08.38	71.61±08.19	71.28±08.05	71.93±09.29	72.55±09.26	0.50±00.17
d20	84.16±06.78	84.47±05.99	83.47±06.00	83.47±06.00	83.48±05.97	83.47±06.54	83.15±06.44	83.47±06.71	82.80±06.00	83.46±04.55	83.47±06.54	0.66±00.07
d21	79.75±17.67	80.44±16.59	80.44±16.59	80.78±16.60	81.13±17.30	80.78±18.04	80.44±17.90	81.13±16.75	82.16±17.27	81.82±15.85	79.40±17.09	0.47±00.09
d22	83.34±03.80	83.34±04.46	83.34±04.46	83.34±03.80	83.33±03.42	82.59±04.70	82.59±04.70	82.22±04.63	82.59±04.70	80.00±04.45	83.33±04.76	0.60±00.07
d23	34.00±08.89	36.00±08.89	36.00±08.31	35.00±07.42	35.00±09.22	36.00±08.89	34.50±09.60	35.50±09.07	35.00±09.75	32.50±08.14	32.50±07.50	0.79±00.12
d24	79.71±09.90	79.00±12.40	79.67±11.89	80.33±11.70	80.33±11.70	81.04±08.90	82.37±09.62	82.38±08.11	81.00±10.67	83.00±09.61	79.00±13.10	0.53±00.14
d25	54.62±02.86	55.28±03.00	55.12±02.62	56.44±02.42	56.94±02.53	58.01±04.03	59.49±04.13	60.73±03.73	61.97±04.39	63.37±04.14	63.04±04.13	0.94±00.01
d26	99.39±00.29	93.96±01.35	94.09±01.27	94.27±01.30	94.75±01.39	95.94±01.09	97.99±00.67	99.47±00.17	99.42±00.31	99.42±00.31	99.42±00.31	0.93±00.01
d27	88.91±09.55	89.78±08.40	89.49±08.19	90.34±07.29	90.63±07.27	91.48±07.37	91.76±06.38	92.33±06.45	93.17±04.95	94.59±04.31	93.17±04.95	0.79±00.06
d28	94.43±05.01	93.83±06.03	94.27±05.82	94.68±05.58	95.12±05.18	95.65±04.51	95.90±04.38	96.09±03.80	95.96±04.05	95.90±04.14	95.90±04.14	0.85±00.04
d29	62.69±19.98	58.87±22.41	60.34±22.28	61.51±21.25	60.63±19.61	60.92±19.31	63.55±18.48	63.24±17.06	65.82±15.07	67.87±12.02	67.26±11.66	0.83±00.06
d30	81.05±05.91	80.38±08.18	81.05±08.39	81.72±07.43	82.38±06.93	83.05±06.99	81.05±08.39	83.05±09.19	81.71±09.53	81.71±08.01	83.05±09.66	0.70±00.08
d31	40.98±07.39	44.81±06.40	45.99±06.44	46.87±06.18	47.48±04.86	46.86±05.86	46.28±05.66	46.57±06.06	47.47±04.97	47.16±06.63	46.59±05.22	0.89±00.02
d32	97.53±01.42	96.23±01.79	96.71±01.87	96.97±01.68	97.36±01.67	97.40±01.63	97.62±01.39	97.53±01.24	97.40±01.33	97.71±01.38	97.49±01.52	0.89±00.02
d33	98.59±00.37	97.14±00.87	97.38±00.88	97.59±00.80	97.83±00.72	98.07±00.70	98.30±00.78	98.59±00.60	98.73±00.71	98.75±00.55	98.75±00.55	0.91±00.01
d34	43.88±16.78	46.74±15.94	47.21±14.70	47.69±14.94	47.74±15.97	49.71±16.44	50.62±15.35	50.60±16.26	53.48±14.34	55.88±12.63	48.74±16.14	0.64±00.13
d35	63.15±14.79	84.30±10.77	86.50±09.77	87.38±09.04	88.26±08.60	89.58±08.41	89.29±08.60	89.58±09.19	89.73±08.98	89.87±09.65	89.58±09.54	0.91±00.01
d36	64.52±33.88	67.03±31.42	69.65±28.17	72.15±26.94	74.13±26.89	75.49±26.73	76.01±26.39	77.79±25.62	82.92±25.85	87.51±24.39	90.23±24.85	0.96±00.05
d37	80.14±03.68	79.19±04.06	79.91±04.13	80.73±03.29	80.85±03.61	81.21±03.34	81.57±03.64	82.27±03.06	83.57±03.45	83.92±04.30	84.16±04.26	0.90±00.02
d38	96.77±02.59	96.77±02.38	96.77±02.38	96.77±02.38	96.77±02.38	96.77±02.38	96.77±02.38	96.77±02.38	96.54±02.39	96.54±02.39	96.77±02.38	0.79±00.03
d39	95.00±05.24	95.56±04.84	95.56±04.84	95.56±04.84	96.11±04.34	96.11±04.34	96.11±04.34	95.56±04.84	95.56±04.84	95.56±04.84	96.11±05.00	0.89±00.07
d40	95.00±06.71	96.00±04.90	95.00±05.00	95.00±05.00	95.00±05.00	95.00±05.00	94.00±06.63	94.00±08.00	95.00±06.71	95.00±06.71	95.00±06.71	0.93±00.03
rank	7.7	8.1	7.7	6.9	5.7	5.4	5.5	4.7	5.3	4.4	4.8	

Table 4: Accuracy percentage and standard deviation of our approach instantiated with ML-kNN the following thresholding strategies: a) default, b) OneThreshold using thresholds ranging from 0.5 to 0.9 with a step of 0.05 and using 5-fold cross-validation to automatically compute the threshold that optimizes precision.

id	default	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	optimized result	optimized threshold
d1	97.22±01.33	97.22±01.33	97.89±01.16	98.00±01.09	98.33±00.74	98.44±00.89	98.67±00.97	99.00±01.05	98.89±01.31	99.33±00.74	99.00±01.05	0.94±00.01
d2	67.05±07.57	67.05±07.57	67.50±08.80	70.80±07.11	70.13±09.17	69.69±07.64	69.91±07.52	70.36±07.22	70.14±07.40	70.14±07.40	68.81±08.07	0.62±00.03
d3	74.49±14.82	74.49±14.82	74.01±12.28	73.12±11.98	73.12±10.51	73.10±09.42	74.82±10.02	74.41±09.52	74.41±09.12	73.95±09.08	76.29±13.69	0.64±00.04
d4	44.19±16.63	44.19±16.63	43.74±15.55	44.74±13.15	42.74±11.25	42.71±12.78	44.24±11.06	41.76±12.35	42.24±12.66	43.19±14.19	44.24±17.53	0.37±00.06
d5	88.18±03.73	88.18±03.73	88.18±03.38	88.18±03.38	88.18±03.38	88.33±03.59	89.14±03.71	91.86±04.39	93.93±04.24	96.01±02.77	97.45±01.45	0.95±00.00
d6	73.79±04.85	73.79±04.85	74.14±06.53	74.84±07.17	73.81±07.67	72.74±07.34	73.10±07.46	70.65±07.53	69.98±06.77	69.27±07.24	73.09±07.36	0.60±00.07
d7	96.57±04.00	96.57±04.00	96.43±04.05	96.43±04.05	96.57±03.68	96.57±03.68	96.43±03.63	96.14±04.38	95.86±04.26	95.71±03.61	96.14±04.38	0.71±00.06
d8	87.38±07.22	87.38±07.22	88.07±07.59	87.79±07.14	88.65±06.62	88.77±06.54	89.00±06.75	88.54±07.12	88.42±06.70	88.88±06.58	88.53±07.39	0.72±00.13
d9	32.28±15.33	32.28±15.33	32.76±14.83	33.23±14.87	33.57±14.96	32.90±14.44	32.56±14.46	32.69±14.42	32.76±14.36	32.76±14.36	26.29±22.11	0.29±00.05
d10	83.42±05.63	83.42±05.63	83.69±05.83	83.69±05.95	83.70±05.92	83.42±06.24	84.26±05.98	83.98±05.70	84.24±06.40	83.43±06.72	83.70±05.92	0.74±00.07
d11	82.32±17.23	82.32±17.23	82.03±17.68	82.03±17.68	82.32±17.81	82.18±17.77	82.47±17.64	82.61±17.54	82.47±17.51	82.18±17.27	82.32±17.71	0.63±00.06
d12	74.60±05.06	74.60±05.06	75.50±05.22	75.60±04.69	75.70±03.77	74.60±04.32	75.30±03.69	73.70±05.27	73.30±05.00	72.50±05.18	75.70±04.15	0.68±00.02
d13	96.73±01.64	96.73±01.64	97.00±01.45	97.00±01.45	97.00±01.45	97.00±01.45	97.27±01.71	97.27±01.71	97.55±02.25	98.10±02.12	97.82±02.03	0.92±00.02
d14	76.95±05.58	76.95±05.58	77.20±05.94	76.82±06.13	76.68±06.21	77.99±06.22	77.21±05.18	76.30±05.38	77.60±05.29	77.73±05.19	77.33±06.25	0.67±00.03
d15	66.24±32.56	66.24±32.56	65.04±33.31	65.04±33.31	65.34±33.27	65.34±33.08	65.66±31.89	65.65±32.85	66.25±32.47	65.95±32.17	65.92±33.66	0.30±00.08
d16	53.14±09.48	53.14±09.48	50.27±10.40	54.11±10.18	53.43±11.59	53.43±11.08	53.29±11.52	53.56±11.46	53.70±11.46	53.70±11.46	51.37±11.43	0.52±00.03
d17	64.40±08.70	64.40±08.70	63.32±09.21	63.84±08.72	62.84±08.20	63.32±08.90	64.34±07.16	62.29±08.74	62.29±08.74	62.29±08.74	63.32±09.21	0.61±00.04
d18	70.63±11.59	70.63±11.59	67.81±11.43	69.22±11.34	66.41±09.61	68.79±11.89	68.36±12.31	66.97±11.65	66.51±11.50	66.51±11.50	66.43±11.56	0.64±00.03
d19	72.90±09.20	72.90±09.20	72.58±08.35	72.58±07.42	71.93±08.81	72.25±07.73	71.95±08.42	71.28±09.12	69.97±10.15	69.96±10.48	73.24±08.80	0.44±00.11
d20	83.83±05.66	83.83±05.66	83.17±06.03	83.17±06.03	83.51±05.74	83.17±05.84	82.83±05.76	82.49±05.82	81.48±06.43	80.17±07.34	82.84±05.72	0.73±00.06
d21	80.79±14.62	80.79±14.62	81.15±14.95	80.46±14.70	81.15±14.38	81.15±13.79	80.13±14.84	80.46±15.56	79.77±14.33	80.12±14.19	80.78±15.94	0.36±00.06
d22	82.59±03.33	82.59±03.33	82.59±03.33	82.59±03.33	82.59±03.33	82.22±03.63	82.60±03.72	82.22±03.99	82.97±03.78	80.00±05.55	82.22±03.63	0.72±00.08
d23	34.00±05.39	34.00±05.39	35.50±07.23	34.50±06.87	34.50±06.87	34.50±06.87	35.00±06.71	35.00±06.71	35.00±06.71	35.00±06.71	33.00±07.81	0.33±00.06
d24	78.33±13.20	78.33±13.20	79.00±13.10	81.71±08.01	82.37±07.55	81.04±09.85	78.37±12.91	79.71±10.76	79.04±11.33	80.38±10.55	80.37±11.36	0.55±00.12
d25	54.30±02.47	54.30±02.47	57.93±05.76	61.15±05.81	62.54±05.00	63.86±06.26	64.43±05.70	64.76±06.00	65.09±05.72	65.10±05.28	57.85±04.40	0.53±00.04
d26	93.61±01.43	93.61±01.43	93.69±01.33	93.72±01.35	93.93±01.45	94.01±01.41	94.14±01.36	94.41±01.42	94.80±01.61	95.79±01.26	97.54±00.87	0.94±00.00
d27	88.07±08.82	88.07±08.82	88.64±08.55	89.50±08.35	89.78±07.65	89.78±07.65	90.07±07.44	89.50±07.85	90.34±08.23	91.47±07.31	90.07±07.44	0.73±00.05
d28	83.33±16.89	83.33±16.89	84.39±16.17	85.52±14.74	86.36±14.10	86.55±14.12	86.80±14.34	88.18±11.81	89.21±10.65	89.96±10.64	86.71±14.21	0.72±00.11
d29	60.65±16.57	60.65±16.57	61.82±16.05	63.59±15.26	65.03±12.54	63.23±09.49	63.23±07.73	63.24±09.31	62.36±09.23	62.36±09.23	62.39±14.54	0.64±00.05
d30	82.43±08.55	82.43±08.55	82.43±08.55	83.09±07.49	83.09±07.49	83.09±07.49	82.38±07.55	83.71±06.99	81.72±07.43	82.43±09.06	82.38±07.55	0.74±00.05
d31	44.54±05.10	44.54±05.10	45.41±05.63	44.82±05.73	44.82±05.73	43.92±06.86	43.33±06.29	43.03±06.69	43.33±06.53	43.33±06.53	45.99±05.73	0.35±00.02
d32	96.67±01.81	96.67±01.81	96.80±01.66	97.06±01.57	96.80±01.59	96.88±01.52	97.06±01.39	97.23±01.46	97.58±01.27	97.71±01.13	97.71±01.13	0.86±00.01
d33	96.21±01.08	96.21±01.08	96.42±01.02	96.63±01.23	96.71±01.14	96.93±01.10	97.32±01.16	97.69±00.85	97.77±00.85	97.96±00.81	97.96±00.97	0.86±00.03
d34	47.67±17.08	47.67±17.08	47.21±17.65	47.24±17.84	47.22±15.97	50.57±15.27	50.10±14.98	48.19±15.47	48.64±15.73	51.07±13.62	43.38±16.82	0.41±00.11
d35	81.70±10.26	81.70±10.26	80.67±10.73	80.52±10.77	81.12±10.46	81.55±10.10	81.70±09.73	81.56±09.94	81.85±10.06	80.53±09.15	81.41±09.85	0.53±00.08
d36	82.39±24.50	82.39±24.50	82.39±24.63	82.80±24.63	83.74±24.18	84.16±24.05	85.20±24.24	85.41±24.30	85.41±24.35	86.13±24.53	82.28±24.85	0.63±00.14
d37	78.84±04.22	78.84±04.22	79.55±04.74	80.37±04.57	80.37±04.60	81.20±05.60	81.09±04.90	82.63±04.18	82.14±04.30	82.26±03.58	81.68±04.86	0.73±00.02
d38	95.62±03.20	95.62±03.20	96.08±02.96	96.31±03.15	95.84±03.57	96.08±03.76	95.84±03.57	95.84±03.57	96.31±02.98	96.31±02.58	96.08±03.12	0.90±00.02
d39	94.44±04.30	94.44±04.30	95.00±04.61	95.00±04.61	95.00±04.61	96.11±04.34	96.11±04.34	96.11±04.34	96.11±04.34	96.11±04.34	95.55±04.16	0.80±00.15
d40	96.00±04.90	96.00±04.90	96.00±04.90	96.00±04.90	96.00±04.90	96.00±04.90	96.00±04.90	96.00±04.90	96.00±04.90	95.00±06.71	96.00±04.90	0.89±00.04
rank	7.5	7.5	7.0	5.9	5.9	5.6	5.0	5.7	5.2	5.4	5.5	

Table 5: Accuracy percentage and average rank of the competing approaches for all datasets.

id	DS	DV	DVS	KNORA	LCA	MCB	CLR	ML-kNN	MV	OLA
d1	98.89±01.11	96.44±01.55	96.10±02.18	99.22±00.71	97.78±01.22	98.89±01.31	99.11±00.97	99.00±01.05	91.10±02.42	99.11±01.09
d2	68.60±08.36	58.19±10.16	57.52±10.33	64.58±08.01	60.87±09.72	55.99±10.25	72.13±05.33	68.81±08.07	54.21±10.76	64.17±08.06
d3	73.62±11.63	69.17±14.12	49.59±13.99	75.32±08.63	54.49±10.03	63.85±13.75	75.32±16.38	76.29±13.69	25.30±11.39	64.31±14.25
d4	39.88±15.03	27.64±10.80	23.45±13.55	46.14±17.73	44.33±21.32	39.38±17.17	49.10±23.09	44.24±17.53	44.26±24.75	39.38±16.85
d5	96.48±02.64	86.42±03.73	86.90±05.31	97.60±02.04	91.05±01.88	94.09±04.09	97.44±02.77	97.45±01.45	86.25±03.73	92.33±04.97
d6	70.65±08.76	74.84±07.65	71.72±07.42	71.66±08.21	74.51±07.03	72.37±07.44	73.07±07.15	73.09±07.36	71.70±06.03	72.37±07.44
d7	96.00±02.98	95.86±05.25	96.43±02.49	95.28±03.73	96.43±04.00	96.14±03.67	96.28±03.68	96.14±04.38	95.14±05.72	96.14±03.38
d8	88.83±07.12	78.17±09.40	83.44±08.59	89.00±06.32	83.32±08.82	90.04±06.83	90.16±06.57	88.53±07.39	70.01±12.90	89.46±07.04
d9	34.88±12.31	18.71±13.62	29.76±19.20	32.16±11.36	31.87±19.33	31.87±16.65	30.30±20.93	26.29±22.11	13.50±10.13	31.26±17.30
d10	81.80±06.84	83.14±05.82	82.87±06.97	85.58±05.25	83.41±05.22	83.99±05.44	82.87±06.21	83.70±05.92	82.87±05.48	83.14±07.01
d11	81.16±15.55	81.31±17.61	81.60±17.66	80.72±14.33	82.76±16.21	82.03±17.99	80.73±18.50	82.32±17.71	82.17±15.70	81.88±17.95
d12	72.50±04.90	75.40±05.30	73.20±05.27	73.30±02.93	74.60±04.18	74.70±04.24	75.30±05.60	75.70±04.15	70.00±04.65	74.20±04.66
d13	95.63±02.50	97.56±03.30	86.90±04.01	96.45±03.00	92.60±04.78	96.46±03.01	97.83±02.03	97.82±02.03	93.15±05.39	95.92±02.49
d14	75.14±05.76	73.69±04.69	74.35±05.86	75.00±06.46	77.34±06.28	77.08±05.41	77.08±05.92	77.33±06.25	72.00±04.48	76.82±05.67
d15	66.86±30.96	58.08±41.15	59.65±36.67	64.11±35.60	61.76±35.11	61.42±36.02	65.04±33.78	65.92±33.66	56.00±41.88	64.15±33.17
d16	49.04±12.02	39.69±17.75	40.49±13.78	48.47±14.01	47.72±11.67	49.06±12.73	58.33±13.04	51.37±11.43	35.87±15.18	50.27±11.82
d17	62.34±10.14	29.34±11.05	40.53±13.33	60.76±08.03	57.68±12.33	57.66±09.60	66.45±10.05	63.32±09.21	47.34±11.92	59.18±09.27
d18	66.38±15.08	55.80±17.41	58.05±11.20	70.17±12.11	62.25±11.07	67.34±12.20	71.62±12.35	66.43±11.56	53.01±18.44	68.29±10.10
d19	72.20±05.94	73.55±09.47	68.29±10.27	69.94±07.22	72.25±08.00	71.90±07.52	72.55±09.26	73.24±08.80	73.55±09.47	71.59±07.58
d20	79.82±07.31	84.49±05.76	82.18±04.20	79.84±07.69	83.19±07.75	83.45±05.86	83.47±06.54	82.84±05.72	83.51±05.93	83.45±05.86
d21	80.47±12.88	76.78±11.68	79.47±14.10	76.73±14.07	80.50±11.90	81.84±14.37	79.40±17.09	80.78±15.94	80.09±18.09	80.83±12.34
d22	78.89±03.33	83.34±03.80	83.33±04.76	80.00±05.79	81.48±04.38	82.96±03.78	83.33±04.76	82.22±03.63	82.96±02.96	83.70±03.78
d23	33.50±08.96	15.00±05.92	25.00±09.49	38.50±10.01	29.50±08.79	31.50±09.76	32.50±07.50	33.00±07.81	30.00±07.07	31.50±09.76
d24	80.42±09.62	81.71±08.54	81.04±08.90	77.79±09.83	80.38±09.67	83.04±07.60	79.00±13.10	80.37±11.36	79.67±12.96	83.04±07.60
d25	64.76±05.25	53.31±03.41	52.89±03.51	63.86±04.32	60.15±03.85	59.99±05.66	63.04±04.13	57.85±04.40	53.06±03.56	62.54±04.04
d26	96.32±01.43	93.22±01.30	94.67±01.24	97.48±00.92	96.10±01.62	94.67±01.14	99.42±00.31	97.54±00.87	92.29±01.42	96.40±01.46
d27	90.89±05.95	85.21±11.13	86.63±08.63	91.79±07.45	88.36±08.50	88.35±08.60	93.17±04.95	90.07±07.44	78.36±15.20	89.50±07.75
d28	94.21±05.83	89.36±12.22	92.87±05.58	94.27±04.86	93.03±06.51	95.03±04.99	95.90±04.14	86.71±14.21	68.90±20.13	95.59±04.49
d29	66.45±09.72	57.69±23.86	64.61±13.57	62.93±09.58	63.52±10.63	66.14±10.48	67.26±11.66	62.39±14.54	57.39±23.61	66.71±10.20
d30	85.71±05.87	84.43±06.10	83.76±06.86	85.10±05.14	83.76±05.42	84.38±07.53	83.05±09.66	82.38±07.55	81.71±06.12	84.38±07.53
d31	43.34±06.05	32.45±07.09	29.80±07.30	42.44±06.16	40.12±05.43	38.04±09.56	46.59±05.22	45.99±05.73	28.58±07.71	41.88±05.94
d32	94.59±02.07	92.82±02.58	83.38±02.94	97.53±01.29	97.19±01.21	97.45±01.05	97.49±01.52	97.71±01.13	91.73±02.98	97.36±01.17
d33	97.62±00.91	95.92±01.09	96.02±01.12	98.14±00.65	97.83±00.72	97.77±00.57	98.75±00.55	97.96±00.97	93.88±00.96	98.12±00.70
d34	50.60±15.66	56.26±11.21	54.86±16.30	47.67±16.12	48.24±19.19	51.53±13.96	48.74±16.14	43.38±16.82	35.14±18.97	54.45±13.70
d35	59.63±14.48	34.02±23.69	37.09±18.31	50.70±14.07	46.90±17.08	71.93±10.92	89.58±09.54	81.41±09.85	25.91±19.15	76.55±16.27
d36	88.25±24.33	70.45±34.49	87.72±24.44	86.35±24.56	77.88±25.22	85.94±23.86	90.23±24.85	82.28±24.85	63.96±44.02	86.15±23.86
d37	80.97±04.24	72.22±05.89	64.20±07.54	80.37±03.89	76.12±04.43	81.32±05.71	84.16±04.26	81.68±04.86	69.62±05.37	81.56±05.54
d38	96.07±02.57	95.85±03.86	96.08±02.55	95.84±02.91	95.38±03.45	96.31±02.38	96.77±02.38	96.08±03.12	95.39±03.44	96.31±02.38
d39	93.33±11.60	92.78±06.11	87.22±12.68	97.22±04.48	96.11±05.00	97.22±04.48	96.11±05.00	95.55±04.16	87.19±10.53	97.22±04.48
d40	97.00±06.40	93.00±07.81	82.00±13.27	95.00±09.22	76.00±12.00	93.00±07.81	95.00±06.71	96.00±04.90	79.00±13.00	93.00±07.81
rank	5.1	6.9	7.4	4.9	5.9	4.8	3.1	4.1	8.6	4.2

Table 6: Wins, ties and losses (w:t:l) for all pairs of methods.

	CLR	DS	DV	DVS	KNORA	LCA	MCB	MLkNN	MV	OLA
CLR	-	30:0:10	30:0:10	32:1:7	29:2:9	30:1:9	31:1:8	26:0:14	35:0:5	30:0:10
DS	10:0:30	-	30:0:10	30:0:10	21:0:19	26:0:14	18:0:22	13:0:27	33:0:7	15:0:25
DV	10:0:30	10:0:30	-	20:0:20	12:0:28	12:0:28	10:1:29	8:0:32	34:1:5	8:1:31
DVS	7:1:32	10:0:30	20:0:20	-	11:0:29	8:0:32	5:0:35	10:0:30	29:0:11	3:0:37
KNORA	9:2:29	19:0:21	28:0:12	29:0:11	-	28:0:12	20:1:19	17:0:23	33:0:7	20:1:19
LCA	9:1:30	14:0:26	28:0:12	32:0:8	12:0:28	-	14:0:26	14:0:26	33:0:7	9:0:31
MCB	8:1:31	22:0:18	29:1:10	35:0:5	19:1:20	26:0:14	-	16:0:24	35:1:4	13:8:19
MLkNN	14:0:26	27:0:13	32:0:8	30:0:10	23:0:17	26:0:14	24:0:16	-	36:0:4	23:0:17
MV	5:0:35	7:0:33	5:1:34	11:0:29	7:0:33	7:0:33	4:1:35	4:0:36	-	4:0:36
OLA	10:0:30	25:0:15	31:1:8	37:0:3	19:1:20	31:0:9	19:8:13	17:0:23	36:0:4	-

Table 7: Average number of models queried by dynamic ensemble selection methods.

dataset id	DVS	ML-kNN	CLR	KNORA
d1	116	155	11	87
d2	79	93	15	44
d3	106	54	21	46
d4	135	100	67	103
d5	134	60	8	91
d6	124	167	56	104
d7	101	173	35	41
d8	95	138	15	48
d9	102	96	30	45
d10	144	150	51	91
d11	98	131	47	53
d12	151	127	41	127
d13	105	116	16	56
d14	119	127	29	55
d15	113	120	108	29
d16	112	50	20	31
d17	168	47	21	134
d18	107	48	30	60
d19	136	191	101	114
d20	172	142	57	153
d21	131	171	107	51
d22	176	144	73	159
d23	137	68	30	43
d24	94	173	94	56
d25	167	72	8	91
d26	112	138	14	57
d27	134	147	33	103
d28	120	100	26	134
d29	134	68	21	141
d30	141	140	52	114
d31	145	69	14	96
d32	99	115	15	25
d33	141	182	15	139
d34	156	112	64	129
d35	102	26	15	48
d36	89	108	8	78
d37	131	65	12	107
d38	177	155	33	173
d39	111	78	18	50
d40	79	115	13	109
mean	125	113	36	85