Dynamic Ensemble Selection with Probabilistic Classifier Chains

Anil Narassiguin[1,2], Haytham Elghazel[1], and Alex Aussem[1]

[1] LIRIS UMR CNRS 5205, Université Lyon 1, 69622 Villeurbanne, France,
{haytham.elghazel, aaussem}@univ-lyon1.fr
[2] EASYTRUST, 71 Boulevard National, 92250 La garenne colombes, France
anil.narassiguin@easytrust.com

Abstract. Dynamic ensemble selection (DES) is the problem of finding, given an input $\mathbf{x}$, a subset of models among the ensemble that achieves the best possible prediction accuracy. Recent studies have reformulated the DES problem as a multi-label classification problem and promising performance gains have been reported. However, their approaches may converge to an incorrect, and hence suboptimal, solution as they don't optimize the true - but non standard - loss function directly. In this paper, we show that the label dependencies have to be captured explicitly and propose a DES method based on Probabilistic Classifier Chains. Experimental results on 20 benchmark data sets show the effectiveness of the proposed method against competitive alternatives, including the aforementioned multi-label approaches. This study is reproducible and the source code has been made available online.[3]

Keywords: Dynamic ensemble selection, Multi-label learning, Probabilistic Classifier Chains

1 Introduction

The ubiquity of ensemble models in several interesting machine learning problems stems primarily from their potential to significantly increase prediction accuracy over individual classifier models [10, 29, 18]. Ensemble methods can be divided into two categories, depending on how they generate the committee of the classifiers. When the same classification algorithm is used to generate all the models of the ensemble, the ensemble method is called *homogeneous*, otherwise it is called *heterogeneous*. In the last decade, there has been a great deal of research focused on the problem of selecting good subensembles of base classifiers prior to combination in order to improve generalization and prediction efficiency.

The process of selecting a subset of classifiers is called *ensemble selection* or *ensemble pruning*. When the same subset of models is selected for all test instances, the process is referred to as *static selection* [14]. In that case, the simplest idea is to select the ensemble members from a set of individual classifiers that are subject to less resource consumption and response time with accuracy that performs at least as good as the original ensemble. A natural follow-up is

[3] https://github.com/naranil/pcc_des

to determine this subset dynamically, i.e. according to the current input feature $\mathbf{x}$. This process is referred to as *dynamic ensemble selection* (DES).

Several DES methods have been recently proposed in the literature. A comprehensive coverage of *individual-based* and *group-based* DES methods is provided in [3]. In individual-based methods, the selection of a subset of models for each test instance is done by estimating the competence level of the base classifiers *individually*, that is, without taking their dependency structure of the model errors into account. *Group-based* methods make one step further by modeling the error co-occurrences.

As noted in [16, 17, 20], DES may be cast as a distinct special case of multi-label classification (MLC) problem with a specific zero-one error expressing the fact that at least, half of the base classifiers selected for inclusion of the sub-ensemble should be correct for the overall class to be correct (i.e. $precision > 1/2$, yes or no?). The question raised by these authors was: What should be the properties of the MLC algorithm to minimize this non-standard loss? This question was addressed from an experimental point of view only, pointing out that *precision* was found experimentally a good surrogate loss candidate for the success of DES. Yet, many loss functions has been proposed in the literature and it is now well understood that a MLC method performing optimally for one loss is likely to perform suboptimally for another loss [8]. For simple loss functions, analytic expressions of the Bayes (optimal) classifier can be derived. For example, the Hamming loss minimizer coincides with the marginal modes of the conditional distribution of the class labels given an instance. Conversely, for the subset 0/1 loss, the risk minimizer is given by the joint mode of the conditional distribution, for which *individual-based* methods might not be good choices. For more complex multi-label loss functions like the one associated with the DES problem, the Bayes (optimal) classifier is unknown and the minimization of such losses requires more involved procedures. In this paper, we show that the minimization of the true loss function necessitates the modeling of dependencies between labels (i.e. co-occurrence of errors) and we use Probabilistic Classifier Chains (PCC), with Monte Carlo sampling, as a "plug-in rule approach" for optimizing this loss directly.

The rest of the paper is organized as follows: Section 2 reviews recent studies on DES and introduces our contribution in this context using MLC. Experiments using relevant benchmarks data sets are presented in Section 3. Finally, Section 4 concludes with a summary of our contributions and raises issues for future work.

2 Problem statement and contribution

In this section, we first survey and appraise the recent literature reporting the use of machine learning techniques devoted to the DES problem, giving some prominence to the use of multi-label classification methods. We then present our proposed DES approach based on Probabilistic Classifier Chains.

2.1 Dynamic ensemble selection (DES)

The key assumption on which DES methods hinge is that each model in the ensemble has distinct prediction abilities on different subsets of the input space. A criterion to measure the level of competence of a base classifier (*e.g.* accuracy) is needed. The literature reports several of DES methods, considers the classifiers either *individually* or in *groups* [3]. In the first category, the classifiers are selected based on their individual competence on the whole or on a local region of the feature space using a validation set. Most of the methods proposed for this purpose are based either on the nearest neighbors algorithm [26, 12] or on clustering techniques [13]. Regarding the number of classifiers they select, these individual-based selection procedures are organized into two groups: *dynamic selection*, for the methods that only select the best classifier; and *dynamic combination*, for the methods that are not restricted in the number of classifiers they select.

In the (*group-based*) DES category, the selection procedures decide for the appropriate subset of the initial ensemble by taking into account the dependencies between the classification errors of the individual models. The most famous methods in this category are *meta-learning* based procedures. Recently, the authors in [6] proposed a DES "meta-learning" framework: instead of using only single criterion to estimate the competence level of the classifiers, several meta-features are used to capture distinct desirable "properties" characterizing the behavior of the base classifiers. These meta-features are extracted from the training data and used by the meta-classifier to decide whether a base classifier is competent on a given input sample $\mathbf{x}$.

2.2 DES as a multi-label classification problem

The DES problem has recently been reformulated as a multi-label classification (MLC) problem [16, 17, 20]. The multi-label training set is constructed on a validation set. The labels are 0-1 indicator random variables indicating whether the corresponding model has made an error on input sample $\mathbf{x}$. The transformation process is illustrated in Table 1. This formulation allows us to cast the DES problem as a standard MLC problem, which can be efficiently solved using standard MLC techniques. The IBEP-MLC method in [16, 17] was the first framework to use MLC approaches for DES: ML-KNN [27] and Calibrated Label Ranking (CLR) [11]. Significant improvements in accuracy have been reported using a heterogeneous ensemble of 200 classifiers. Another recent proposal, called CHADE (for CHAined Dynamic Ensemble) algorithm [20] is based the classifier chain (CC) technique [22]. This approach was evaluated on a bagging ensemble of 100 decision stumps using a large set of classification data sets.

However, the literature leaves open the question of deciding what MLC algorithm should work best, and more importantly how to exploit the dependencies between the labels, implicitly giving the misleading impression that any MLC method could solve the DES task. The benefit of exploiting label dependence is known to be closely depend on the type of loss to be minimized. Rather than proposing yet another MLC algorithm, the aim of this paper is to elaborate more closely on the idea of exploiting label dependence to solve the DES task.

Table 1. Problem transformation

Validation set		Classifier predictions				Multi-label metabase			
$\mathbf{X}_{val}$	$\mathbf{Y}_{val}$	c_1	c_2	c_3		$\mathbf{X}_{val}$	$\hat{\mathbf{Y}}_{val}$		
$\mathbf{x}_1$	0	1	1	0	$\Rightarrow$	$\mathbf{x}_1$	0	0	1
$\mathbf{x}_2$	1	0	1	1		$\mathbf{x}_2$	0	1	1
...		...				...		...	
$\mathbf{x}_n$	0	0	1	0		$\mathbf{x}_n$	1	0	1

2.3 DES loss function

When the multi-label training set is constructed for an ensemble of classifiers $\Psi = \{\psi_1, \ldots; \psi_n\}$, the goal is to output a subset $\Psi_\mathbf{x}$ of classifiers ($\Psi_\mathbf{x} \subset \Psi$) using a multi-label classifier for a given test instance $\mathbf{x}$. A natural question is what should be learned from the labels dependency structure to solve the DES task, and what is the appropriate loss function for training the MLC method to obtain a "good" subset of classifiers.

Let's denote the subset of classifiers that correctly classify $\mathbf{x}$ as $\Phi_\mathbf{x}$ and suppose that $\mathbf{h}_\mathbf{x} = (h_i)_{i=1}^n$ ($h_i \in \{0,1\}$) and $\mathbf{w}_\mathbf{x} = (w_i)_{i=1}^n$ ($w_i \in \{0,1\}$) are the binary representations for respectively $\Psi_\mathbf{x}$ and $\Phi_\mathbf{x}$, an intuitive way of obtaining a correct final prediction in a two-class classification task is to have at least 50% of the classifiers from $\Psi_\mathbf{x}$ to be in $\Phi_\mathbf{x}$ [16, 17]. This condition can be written in different ways:

$$\frac{|\Psi_\mathbf{x} \cap \Phi_\mathbf{x}|}{|\Psi_\mathbf{x}|} > 0.5 \Leftrightarrow \frac{\mathbf{h}_\mathbf{x}.\mathbf{w}_\mathbf{x}}{\mathbf{h}_\mathbf{x}.\mathbf{h}_\mathbf{x}} > 0.5 \Leftrightarrow \frac{\sum_{i=1}^{n} h_i.w_i}{\sum_{i=1}^{n} h_i} > 0.5$$

This yields the following actual loss function (also referred to as task loss),

$$Task\_loss(\mathbf{h}_\mathbf{x}, \mathbf{w}_\mathbf{x}) = \begin{cases} 0, & \text{if } \dfrac{\mathbf{h}_\mathbf{x}.\mathbf{w}_\mathbf{x}}{\mathbf{h}_\mathbf{x}.\mathbf{h}_\mathbf{x}} > 0.5 \\ 1, & \text{otherwise.} \end{cases} = 1 - [[\frac{\mathbf{h}_\mathbf{x}.\mathbf{w}_\mathbf{x}}{\mathbf{h}_\mathbf{x}.\mathbf{h}_\mathbf{x}} > 0.5]] \qquad (1)$$

Unfortunately, there is no closed-form of the Bayes optimal multi-label classifier, that is, a mapping $\mathbf{h}^*$ from the input features $\mathcal{X}$ to the labels $\mathcal{Y}$ that minimizes the expected loss (or risk) L of the model h, defined as:

$$R_L(\mathbf{h}) = \mathbb{E}_{\mathbf{XY}} L(\mathbf{Y}, \mathbf{h}(\mathbf{X})) = \sum_{\mathbf{x},\mathbf{y} \in \mathcal{X} \times \mathcal{Y}} P(\mathbf{x}, \mathbf{y}) L(\mathbf{y}, \mathbf{h}(\mathbf{x})) \qquad (2)$$

The optimal classifier, $\mathbf{h}^*$, commonly referred to as Bayes classifier, minimizes the risk conditioned on $\mathbf{x}$: $\mathbf{h}^*(\mathbf{x}) = arg\min_h \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y}|\mathbf{x}) L(\mathbf{y}, \mathbf{h}(\mathbf{x}))$. Finding $\mathbf{h}^*(\mathbf{x})$ directly by brute force search leads to intractable optimization problems and only very few loss functions have a (known) closed-form solution. For simple

loss functions, analytic expressions of the Bayes optimal classifier have been derived in [8]. For example, the Hamming loss minimizer was shown to coincide with the marginal modes of the conditional distribution of the labels given an instance $\mathbf{x}$, and methods such as Binary Relevance (BR), perform particularly well in this case. Conversely, for the subset 0/1 loss, the risk minimizer was proven to be the joint mode of the conditional distribution, for which methods such as the Label Powerset classifier (LP) is a good choice. Further results have been established for the ranking loss [8], and more recently for the F-measure loss [7]. However, as far as we know, there is no closed-form expression of the Bayes classifier that minimizes the DES task loss. In such situations, the true loss is usually replaced by a surrogate loss that is easier to cope with.

2.4 MLC approaches to the DES problem

With the above difficulty in mind, Markatopoulou et al. [16, 17], used the precision loss as surrogate loss:

$$Precision\_loss(\mathbf{h_x}, \mathbf{w_x}) = 1 - \frac{\mathbf{h_x}.\mathbf{w_x}}{\mathbf{h_x}.\mathbf{h_x}} = 1 - Pr(\mathbf{h_x}, \mathbf{w_x}) \tag{3}$$

To solve the problem, two multi-label learning algorithms (ML-KNN [27] and CLR [11]) were used. Each algorithm outputs a score vector for each label. There were used in tandem with a thresholding strategy as an attempt to optimize the task loss. Despite the performance improvements reported, we shall see next that a method performing optimally for the precision loss may not perform well for the DES task loss, even upon tuning the threshold value. More problematic is the fact that the standard version of ML-KNN does not consider the correlation between labels and, as such, is devoted to minimize the Hamming loss (L_H) [8]:

$$L_H(\Psi_\mathbf{x}, \Phi_\mathbf{x}) = \frac{|(\Psi_\mathbf{x} \cap \Phi_\mathbf{x}) \cup (\overline{\Psi_\mathbf{x}} \cap \overline{\Phi_\mathbf{x}})|}{|\Psi|}, \quad L_H(\mathbf{h_x}, \mathbf{w_x}) = \frac{1}{n}\sum_{i=1}^{n}[[h_i = w_i]] \tag{4}$$

Tuning automatically the threshold via cross-validation was performed to overcome the theoretical shortcomings of the base MLC approaches. Clearly, choosing higher confidence thresholds for inclusion in the final pool tends to reduce the precision loss. Threshold values greater than 0.75 have been considered in their work.

In [20], the Classifier Chains (CC) [22] classifier was used to take the correlation between labels into account. However Dembczynski et al. [8] argued that CC is more appropriate for the subset 0/1 loss as it tends to approximate the joint mode of the conditional distribution of label vectors in a greedy manner. The 0/1 loss is given by:

$$L_{0/1}(\Psi_\mathbf{x}, \Phi_\mathbf{x}) = [[\forall \psi \in \Psi_\mathbf{x}, \psi \in \Phi_\mathbf{x}]], \quad L_{0/1}(\mathbf{h_x}, \mathbf{w_x}) = [[\mathbf{h_x} = \mathbf{w_x}]] \tag{5}$$

The above methods have several shortcomings. Consider the simple DES example in Table 2. The ensemble consists of 4 models, each having a mean accuracy exceeding 50%. The joint conditional distribution $P(y_1, \ldots, y_4 \mid \mathbf{x})$ is displayed.

Table 2. A DES example cast as a multi-label problem: different loss functions yield distinct minimizers.

y_1	y_2	y_3	y_4	$P(y_1, \ldots, y_4 \mid \mathbf{x})$
1	1	0	1	3 / 7
1	1	1	0	2 / 7
1	0	1	1	1 / 7
0	0	1	1	1 / 7

It is easy to show that in this toy example, the optimal solution for the Hamming loss, 0/1 loss, DES task loss and Precision loss respectively are given by $\mathbf{h}^*_{\mathbf{hl}} = (1,1,1,1)$, $\mathbf{h}^*_{\mathbf{0/1}} = (1,1,0,1)$, $\mathbf{h}^*_{\mathbf{DEStaskloss}} \in \{(0,1,1,1),(1,0,1,1)\}$ and $\mathbf{h}^*_{\mathbf{Precisionloss}} = (1,0,0,0)$. This illuminating toy example is important to caution the hurried researcher against using "off-the-shelf" MLC techniques to solve the DES problem. Indeed, IBEP-MLC which minimizes the Hamming loss implicitly, would select all the classifiers, whereas CHADE, based on CC that attempts to minimize the 0/1 loss, would output $\{c_1, c_2, c_4\}$. As may be observed, both methods fail to recover the optimal solution for the DES actual loss function, $\{c_2, c_3, c_4\}$ or $\{c_1, c_3, c_4\}$. It is also worth noting that the thresholding strategy based on the marginal label probabilities is unable cope with this problem. In fact, some information on the label dependency structure has to be captured to optimize the DES actual loss function. The following result shows that the precision loss tends to favor the best performing model,

Lemma 1. *The mapping* $\boldsymbol{h(.)} = (h_1(.), \ldots, h_n(.))$ *defined by:*

$$\begin{cases} h_k(\mathbf{x}) = 1, \ k = \arg\max_{i \in \{1,\ldots,n\}} \boldsymbol{P}(Y_i = 1|\mathbf{x}). \\ \\ h_j(\mathbf{x}) = 0, \ j \neq k \end{cases} \tag{6}$$

minimizes the expected precision score loss.

Proof. Minimizing the expected precision loss is equivalent to maximizing the expected precision which can easily be bounded above:

$$\mathbb{E}_{\mathbf{Y}|\mathbf{X}} Pr(\mathbf{h}, \mathbf{Y}) = \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y}|\mathbf{x}) \frac{\sum_{i=1}^{n} h_i . y_i}{\sum_{i=1}^{n} h_i} = \frac{\sum_{i=1}^{n} h_i P(y_i = 1|\mathbf{x})}{\sum_{i=1}^{n} h_i} \leq max_i \ P(y_i = 1|\mathbf{x})$$

The mapping $\mathbf{h(.)}$ defined above reaches this bound and is thus Bayes optimal for the expected precision. This concludes the proof.

Therefore, picking the label having the highest confidence is a Bayes optimal solution to the MLC problem under the precision loss. However, we have just seen that on a toy problem that the best performing model is not always a good solution to the DES problem even if it is straightforward to identify. We may

conclude that Precision loss is not a valid surrogate loss for this task. In this paper we focus on a general technique capable of minimizing the DES actual loss function based on a combination of Probabilistic Classifier Chains and Monte Carlo sampling. A similar approach was successfully applied to maximize the F-measure in [7]. This constitutes our main contribution.

2.5 Probabilistic classifier chains & Monte Carlo inference

We have seen that some information on the joint conditional distribution $P(\mathbf{Y} \mid \mathbf{x})$ has to be captured to minimize the DES task loss. Brute-force search is however intractable as the number of possible labels permutations grows as $\mathcal{O}(2^n)$. One idea to cope with this issue is to infer a label combination probability in a step-wise manner using the chain rule of probability. Given a test instance $\mathbf{x}$, the joint conditional probability of the labels $\mathbf{y} = (y_1, \ldots, y_n)$ can be expressed by the chain rule of probability :

$$P_{\mathbf{x}}(\mathbf{y}) = P(\mathbf{y}|\mathbf{x}) = P(y_1|\mathbf{x}) \cdot \prod_{i=2}^{n} P(y_i|\mathbf{x}, y_1, \ldots y_{i-1}) \tag{7}$$

The rationale behind *Probabilistic Classifier Chains* [5] (PCC) is to estimate the joint conditional probability using this chain rule. PCC is the probabilistic counterpart of the Classifier Chain [22] algorithm. The method goes as follows: n probabilistic classifiers are used to estimate the probability distributions $P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1})$ for each label $i = 1, \ldots, n$. Therefore, the i^{th} classifier h_i is trained on a training data set composed of the original training data $\mathbf{X}_{tr}$ and $(y_{tr_1}, \ldots, y_{tr_{i-1}})$. While the training stage is rather straightforward, several approaches have been proposed in the literature for performing inference during the testing stage. CC is the simplest approach: each h_i predicts in sequential fashion the label y_i with the highest marginal conditional probability, taking as input the current input $\mathbf{x}$ and the previous predicted labels $(\hat{y}_1, \ldots, \hat{y}_{i-1})$. Therefore, CC may be regarded as a greedy approximation of PCC, focusing on the 0/1 loss minimization as the method estimates the mode of the joint distribution in a greedy fashion. In contrast, inference with PCC amounts to explore exhaustively the probability tree to estimate the Bayes optimal solution for any type of loss. This approach called Exhaustive Search (ES) estimates the true risk minimizer at the cost of extensive computation time since the tree diagram grows exponentially with n. Several methods have been proposed to reduce the computational burden of ES: ϵ-Approximation, Beam Search and Monte Carlo sampling (MC) (see for instance [19] and references therein for further details and experimental comparisons). However, ϵ-Approximation and Beam Search also tend to minimize of the 0/1 loss instead of the DES task loss. In this paper, we use Monte Carlo MC sampling technique [21] due to its ability to minimize arbitrary loss functions. The procedure is rather straightforward: given a new unlabeled instance $\mathbf{x}$, the labels are sampled in sequence, by taking the previously sampled labels $\hat{y}_1, \ldots, \hat{y}_i$ as input to the classifier h_i in order to estimate the marginal conditional probability of the next label y_{i+1}. Finally, the label combination $\hat{\mathbf{y}}_{pcc}$ that exhibits the lowest DES task loss value among the n_{MC} samples is chosen

as the final prediction. Note that the DES task loss minimizer is estimated over a subset of n_{MC} samples drawn randomly instead of the whole set of possible labels, in order to keep the computational burden as low as possible. Once the n_{MC} samples are drawn, the search for the DES task loss minimizer requires $O(n_{MC}^2)$ further operations (calls to the loss functon) which can be prohibitive for large values of n_{MC}. Of course, the preference for smaller values of n_{MC} should be traded off against the prediction performance of the selected classifiers. In our experiments, we set $n_{MC} = 1000$. The PCC + Monte Carlo method applied to DES is termed PCC-DES in the sequel.

3 Experiments

In this section, we report on the experiments performed to evaluate the use of the proposed PCC-DES method on several data sets and we compare its predictive performance against other multi-label based DES methods. The following experiments were performed on 20 binary classification data sets primarily selected from the UCI Machine Learning Repository [2] and some other online repositories, covering a wide variety of topics including health, education, business, science etc., and exhibiting various dimensionalities as described in Table 3.

Table 3. Characteristics of the data sets used in the study

Datasets	# Instances	# Features	# Classes	**Ref.**
Adult	48842	14	2	[2]
AutoMoto	1980	2159	2	[23]
BaseHock	1993	4862	2	[28, 23]
Breast cancer wisconsin (original)	699	9	2	[2]
Colic	368	27	2	[2]
Colon	62	2000	2	[1]
Credit Approval	690	15	2	[2]
EleCrypt	1973	2514	2	[23]
German credit	1000	24	2	[2]
GunMid	1847	2917	2	[23]
Hepatitis	155	19	2	[2]
Ionosphere	351	34	2	[2]
Chess (Krvskp)	3196	36	2	[2]
Madelon	2600	500	2	[2]
Ovarian	54	1536	2	[25]
PcMac	1943	3289	2	[28, 23]
RelAthe	1427	4322	2	[28, 23]
Connectionist Bench (Sonar)	208	60	2	[2]
Spambase	4601	57	2	[2]
Congressional Voting Records (Vote)	435	16	2	[2]

3.1 Ensemble generation

In order to make fair comparisons, we used two ensemble generation techniques that appeared in the literature and investigated the performance of PCC-DES against other multi-label based DES techniques.

The *First (ensemble) generation* was used in [16, 17]. An heterogeneous ensemble of 200 classifiers was constructed consisting of: (1) 40 *multilayer perceptrons* (**MLPs**) with {1, 2, 4, 8, 16} hidden units, momentum varying in {0, 0.2, 0.5, 0.9} and two learning rates: 0.3 and 0.6, (2) 60 *k nearest neighbors* (***k*NNs**) with 20 values for k evenly distributed between 1 and the number of training

observations, 3 weighting methods: no weights, inverse-weighting and similarity-weighting, (3) 80 *support vector machines* (**SVMs**) composed of 16 polynomial SVMs with a kernel of degree 2 and 3 and a complexity parameter C varying from 10^{-5} to 10^{2} in steps of 10, and 64 radial SVMs with the same values of C and a width γ in {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2}, and (4) 20 *decision trees* (**DTs**), half of which are trained using Gini and half using entropy as split criteria; five values of the maximum depth pruning option 1, 2, 3, 4 and None, 8 decision trees using also Gini and entropy, varying the number of features to consider when looking for the best split (square root, log2, 50% and 100%) of the total number of features, and 2 decision trees using Gini and 2 values for the minimum number of samples per leaf 2, 3.

The *Second (ensemble) generation* was used in [4]. A pool of 200 heterogeneous models was constructed consisting of: (1) 50 *bagged trees* (**BAG-DTs**) using 25 trees for each splitting criterion (Gini and entropy), (2) 50 *random subspace trees* (**RSM-DTs**) consisting of 25 trees per splitting criterion, (3) 8 *Boosting decision trees* (**BST-DTs**) obtained by boosting a decision tree for each splitting criterion (Gini and entropy) and since Boosting can overfit, boosted DTs were added to the pool after 2, 4, 8, 16 steps of boosting, (4) 14 *Boosting stumps* (**BST-STMP**) obtained by boosting single level decision trees with both splitting criteria, each boosted 2, 4, 8, 16, 32, 64, 128 steps, (5) 24 *multilayer perceptrons* (**MLPs**) with {1, 2, 4, 8, 32, 128} hidden units and a momentum varying in {0, 0.2, 0.5, 0.9}, and (6) 54 *support vector machines* (**SVMs**) composed of 6 linear SVMs with complexity parameter C varying from 10^{-3} to 10^{2} in steps of 10, 48 radial SVMs with the same values of C and a width γ in {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2}.

These two strategies have many classifiers (**MLPs** and **SVMs**) in common. Yet, the *second generation* is expected to perform better as more powerful models (**BAG-DTs, RSM-DTs, BST-DTs, BST-STMP**) are generated. The overall mean error rate, averaged over the 20 data sets, is 0.340 with the *first generation* and 0.288 with the *second generation*. This should be kept in mind when analyzing the results.

3.2 Compared methods & Evaluation protocol

To gauge the practical relevance of our PCC-DES method, we compared its performance to four multi-label based DES methods in terms of accuracy improvements.

- BR-DES: Binary Relevance based DES method. BR resolves the MLC problem by training a classifier for each label separately. It is tailored for the Hamming loss [8].
- LP-DES: Label Powerset based DES method. LP reduces the MLC problem to multi-class classification, considering each label subset as a distinct meta-class. LP is tailored for the subset 0/1 loss [8].
- PM-DES: *Precision loss* minimizer based DES technique. As discussed in Section 2, this approach attempts to select the best classifier in the pool, given $\mathbf{x}$.

- CHADE: CHAined Dynamic Ensemble algorithm [20]. It is based on the classifier chain (CC) technique. CC tailored for the subset 0/1 loss [8].
- BEST: the classifier with the highest accuracy in the validation data is selected (static method) [24].
- ENSEMBLE: the complete ensemble is classically used (baseline method).

Following [8], the logistic regression chosen as the base classifier of the MLC methods in our experiments. As noted earlier, a set of $n_{MC} = 1000$ samples was considered during the MC inference stage. The performance of the models was tested using a 5-fold cross-validation experiment. At each step of the cross-validation, 75% of the training data set was used to train the ensemble and the remaining 25% as a validation set to train the meta-learners for DES. This process was repeated 5 times for each DES method. The overall accuracy was computed by averaging over those 25 iterations.

3.3 Results and discussion

The average accuracies of the compared methods for all 20 data sets using the first and the second generation strategies are reported respectively in Tables 4 and 5. We follow in this study the methodology proposed by [9] for the comparison of several algorithms over multiple data sets. In this study, the non-parametric Friedman test is firstly used to determine if there is a statistically significant difference between the rankings of the compared techniques. The Friedman test reveals here statistically significant differences ($p < 0.05$) for each ensemble generation strategy. Next, as recommended by Demsar [9], we perform the Nemenyi post hoc test with average rank diagrams. These diagrams are given on Figure 1. The ranks are depicted on the axis, in such a manner that the best ranking algorithms are at the rightmost side of the diagram. The algorithms that do not differ significantly (at $p = 0.05$) are connected with a line. The critical difference (CD) is shown above the graph (CD=2.0139 here). As may be observed from CD plots and the results in Tables 4 and 5 PCC-DES outperform the other models most of the time.

As far as the *first ensemble generation* is concerned (*c.f.* Table 4 and Figure 1), the performances of PCC-DES are not statistically distinguishable from the performances of the single best classifier in the ensemble (BEST). As mentioned before, the first generation produces a pool containing several weak classifiers. Selecting the best single model from this pool yields remarkably good performance. The nonparametric statistical tests we used are very conservative. To further support these rank comparisons, we compared the 25 accuracy values obtained over each data set split for each pair of algorithms according to the paired t-test (with $p = 0.05$). The results of these pairwise comparisons are depicted in the last row of Table 4 in terms of "win/tie/loss" statuses of all methods against PCC-DES; the three values respectively indicate how times many the corresponding approach is significantly better/not significantly different/significantly worse than PCC-DES. Inspection of this win/tie/loss values reveals that DES using PCC (PCC-DES) is the only MLC-based DES method able to outperform the best single model BEST. The win/tie/loss values triples

Table 4. Means and standard deviations of accuracy for compared algorithms on the benchmark data sets with the *first ensemble generation* strategy

Data set	ENSEMBLE	PM-DES	BR-DES	LP-DES	CC-DES	PCC-DES	BEST
Adult	0.752±0.06•	0.781±0.04•	0.798±0.06•	0.755±0.06•	0.790±0.06•	0.803±0.04	0.791±0.04•
Auto Moto	0.631±0.16•	0.872±0.04•	0.852±0.04•	0.774±0.06•	0.818±0.06•	0.902±0.05	0.845±0.04•
BaseHock	0.643±0.19•	0.911±0.02•	0.867±0.07•	0.808±0.06•	0.824±0.11•	0.933±0.03	0.912±0.03•
Breast-Cancer	0.960±0.02•	0.965±0.02	0.970±0.02	0.961±0.02•	0.970±0.02	0.970±0.02	0.968±0.02
Colic	0.678±0.03•	0.812±0.05	0.737±0.05•	0.709±0.05•	0.735±0.05•	0.822±0.04	0.821±0.06
Colon	0.684±0.20•	0.781±0.13	0.794±0.15	0.774±0.16•	0.791±0.17	0.813±0.14	0.779±0.15
Credit Approval	0.828±0.06•	0.852±0.03•	0.871±0.03	0.831±0.05•	0.870±0.03	0.872±0.04	0.866±0.03
Elecrypt	0.774±0.23•	0.909±0.02•	0.882±0.05•	0.818±0.07•	0.833±0.10•	0.938±0.02	0.918±0.03•
German Credit	0.700±0.04•	0.727±0.05•	0.736±0.05	0.722±0.04•	0.724±0.04•	0.745±0.05	0.733±0.05
Gunmid	0.582±0.11•	0.768±0.04•	0.756±0.05•	0.715±0.05•	0.738±0.06•	0.806±0.04	0.784±0.04•
Hepatitis	0.794±0.13•	0.806±0.11	0.795±0.13•	0.795±0.13•	0.795±0.13•	0.808±0.12	0.815±0.12
Ionosphere	0.641±0.19•	0.909±0.05	0.766±0.15•	0.661±0.19•	0.765±0.14•	0.919±0.04	0.927±0.05°
Krvskp	0.662±0.12•	0.946±0.02•	0.916±0.05•	0.801±0.09•	0.912±0.06•	0.966±0.02	0.952±0.03•
Madelon	0.501±0.05•	0.584±0.05	0.574±0.04	0.546±0.04•	0.563±0.04	0.590±0.05	0.580±0.06
Ovarian	0.369±0.37•	0.778±0.15	0.833±0.09	0.745±0.13•	0.833±0.10	0.823±0.04	0.771±0.16
PcMac	0.602±0.15•	0.838±0.03•	0.802±0.07•	0.725±0.06•	0.759±0.11•	0.882±0.03	0.836±0.03•
Relathe	0.562±0.06•	0.849±0.04•	0.801±0.06•	0.789±0.06•	0.674±0.07•	0.888±0.03	0.855±0.04•
Sonar	0.093±0.18•	0.438±0.13	0.298±0.14•	0.220±0.19•	0.303±0.14•	0.465±0.12	0.396±0.14•
Spambase	0.756±0.06•	0.890±0.03	0.854±0.03•	0.754±0.06•	0.812±0.05•	0.898±0.03	0.883±0.03•
Vote	0.945±0.04	0.928±0.05	0.940±0.05	0.930±0.05•	0.939±0.05	0.945±0.04	0.938±0.05
(Win/Tie/Loss)	0/0/19	0/9/10	0/6/13	0/0/20	0/5/14		1/8/10

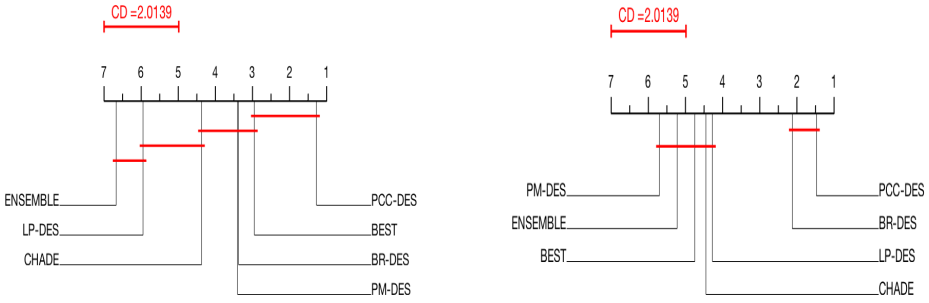

Fig. 1. Average rank diagrams of the compared DES methods using the first (left) and second (right) ensemble generation strategies.

are statistically better with PCC-DES on 10 data sets, poorer on 1 data set only, and not significant on 8 data sets. Overall, PCC-DES compares more favorably to the other approaches, sometimes by a noticeable margin, in terms of accuracy.

Regarding the *second ensemble generation* strategy, here again PCC-DES outperforms the other algorithms, except BR-DES (*c.f.* Table 5 and Figure 1). PCC-DES ranks first as well. Yet, it is not statistically better than BR-DES according the post hoc test. On the other hand, the win/tie/loss counts in Table 5 are statistically better for PCC-DES on 4 data sets and not significant on 16 data sets.

For a better understanding of the behavior of PCC-DES in comparison with the others DES approaches, we explored in the sequel the relation between the diversity-accuracy of the ensemble and the performance of the dynamic ensemble selection. To measure the diversity within the ensemble, we consider the kappa metric (κ) used in [15]. κ evaluates the level of agreement between two classi-

Table 5. Means and standard deviations of accuracy for compared algorithms on the benchmark data sets with the *second ensemble generation* strategy

Data set	ENSEMBLE	PM-DES	BR-DES	LP-DES	CC-DES	PCC-DES	BEST
Adult	0.950±0.03	0.947±0.04	0.952±0.03	0.948±0.03	0.950±0.03	0.952±0.03	0.952±0.03
Automoto	0.878±0.05$^{\bullet}$	0.859±0.04$^{\bullet}$	0.905±0.04	0.879±0.05$^{\bullet}$	0.893±0.04$^{\bullet}$	0.908±0.04	0.856±0.04$^{\bullet}$
BaseHock	0.883±0.06$^{\bullet}$	0.904±0.03$^{\bullet}$	0.921±0.04	0.903±0.03$^{\bullet}$	0.868±0.04$^{\bullet}$	0.930±0.03	0.898±0.04$^{\bullet}$
Breast-Cancer	0.963±0.02	0.951±0.03$^{\bullet}$	0.966±0.02	0.964±0.02	0.963±0.02	0.964±0.02	0.942±0.03$^{\bullet}$
Colic	0.825±0.04$^{\bullet}$	0.808±0.05$^{\bullet}$	0.832±0.03$^{\bullet}$	0.825±0.05$^{\bullet}$	0.823±0.04$^{\bullet}$	0.847±0.04	0.807±0.05$^{\bullet}$
Colon	0.784±0.15$^{\bullet}$	0.765±0.14$^{\bullet}$	0.846±0.13	0.854±0.12	0.842±0.12	0.844±0.11	0.797±0.14$^{\bullet}$
Credit Approval	0.898±0.03	0.858±0.03$^{\bullet}$	0.902±0.02	0.877±0.03$^{\bullet}$	0.902±0.02	0.905±0.02	0.874±0.03$^{\bullet}$
Elecrypt	0.899±0.03$^{\bullet}$	0.899±0.03$^{\bullet}$	0.917±0.03	0.911±0.02$^{\bullet}$	0.897±0.03$^{\bullet}$	0.922±0.02	0.912±0.02
German Credit	0.722±0.05$^{\bullet}$	0.696±0.04$^{\bullet}$	0.744±0.05	0.735±0.04	0.731±0.05$^{\bullet}$	0.748±0.04	0.717±0.05$^{\bullet}$
Gunmid	0.747±0.05$^{\bullet}$	0.747±0.04$^{\bullet}$	0.807±0.05	0.772±0.04$^{\bullet}$	0.780±0.05$^{\bullet}$	0.806±0.04	0.776±0.05$^{\bullet}$
Hepatitis	0.815±0.11	0.790±0.11$^{\bullet}$	0.823±0.10	0.818±0.10	0.812±0.11	0.831±0.09	0.788±0.15$^{\bullet}$
Ionosphere	0.910±0.06$^{\bullet}$	0.891±0.05$^{\bullet}$	0.910±0.06$^{\bullet}$	0.907±0.06$^{\bullet}$	0.910±0.06$^{\bullet}$	0.920±0.05	0.891±0.07$^{\bullet}$
Krvskp	0.952±0.03$^{\bullet}$	0.954±0.02	0.960±0.02	0.956±0.02	0.953±0.02	0.959±0.03	0.958±0.02
Madelon	0.548±0.05$^{\bullet}$	0.540±0.05$^{\bullet}$	0.592±0.04	0.563±0.05$^{\bullet}$	0.573±0.05$^{\bullet}$	0.599±0.04	0.553±0.05$^{\bullet}$
Ovarian	0.762±0.15$^{\bullet}$	0.740±0.15$^{\bullet}$	0.841±0.08	0.738±0.15$^{\bullet}$	0.820±0.08$^{\bullet}$	0.845±0.07	0.764±0.14$^{\bullet}$
PcMac	0.828±0.03$^{\bullet}$	0.847±0.03$^{\bullet}$	0.886±0.02	0.836±0.03$^{\bullet}$	0.859±0.04$^{\bullet}$	0.894±0.02	0.847±0.04$^{\bullet}$
Relathe	0.815±0.05$^{\bullet}$	0.850±0.03$^{\bullet}$	0.863±0.04$^{\bullet}$	0.844±0.04$^{\bullet}$	0.830±0.05$^{\bullet}$	0.879±0.03	0.867±0.05
Sonar	0.323±0.13$^{\bullet}$	0.477±0.11°	0.382±0.09$^{\bullet}$	0.392±0.08	0.340±0.12$^{\bullet}$	0.415±0.08	0.467±0.13°
Spambase	0.900±0.02	0.886±0.03$^{\bullet}$	0.903±0.02	0.900±0.02	0.898±0.02	0.906±0.02	0.882±0.03$^{\bullet}$
Vote	0.950±0.03	0.947±0.04	0.952±0.03	0.948±0.03	0.950±0.03	0.952±0.03	0.952±0.03
(Win/Tie/Loss)	0/6/14	1/3/16	0/16/4	0/9/11	0/8/12		1/5/14

fier outputs. The plots in figure 2 are representative examples of the effects of individual classifier average error and diversity (respectively) on the ability of DES methods for accuracy improvement under the *first generation* and *second generation* strategies.

A closer inspection of plots in this figure reveals the following: (1) not surprisingly, as the individual classifiers become less accurate (respectively more diverse), the dynamic ensemble selection becomes crucial for ensemble learning, (2) a significant accuracy gain was obtained with large values of errors (respectively diversity) with PCC-DES compared to the others MLC-based DES techniques, especially for ensemble models obtained using the *first generation* strategy.

In Table 6, the average number of models selected by BR-DES, LP-DES, CHADE and PCC-DES across all test instances and for all data sets is displayed. Our prime conclusion is that PCC-DES is a promising approach to DES. Concentrating on the actual DES task loss pays off in terms of performance. Compared to all others DES approaches, it appears that PCC-DES selects a far smaller number of models on average, especially with the first ensemble generation strategy containing weaker models as well (*c.f.* Figure 3). We also plotted in Figure 4 the overall accuracy on the 20 data sets as a function of the size of the ensemble, varying from 100 to 500. This confirms that our conclusions are rather insensitive to the size of the original ensemble. The running times are not shown here due to space restrictions. We would like to stress again that the MC inference step with PCC-DES requires $O(n_{MC}^2)$ calls to the loss function. On Madelon, for instance, PCC-DES takes about 90 seconds to label as single test example. This computational overhead prevents PCC-DES from being used in real-time.

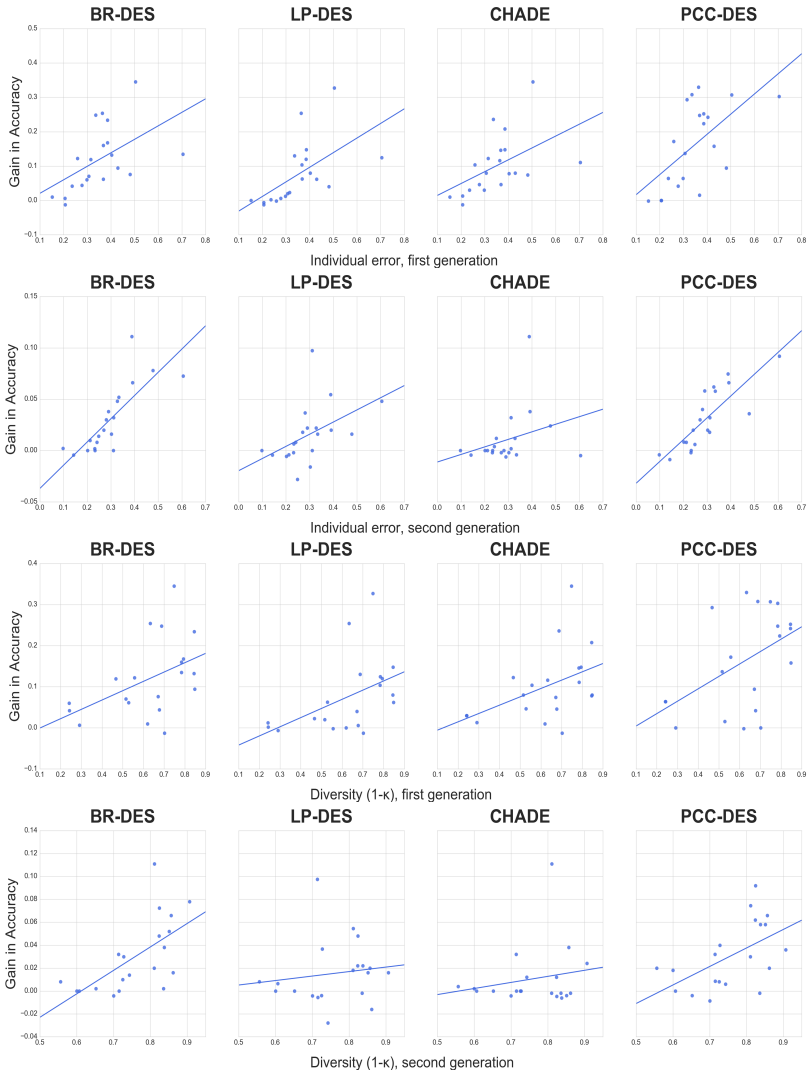

Fig. 2. Gain in accuracy of PCC-DES over the other DES methods vs. individual classifier average error (top plots) and diversity ($1 - \kappa$, lower plots) with the first and second ensemble generation strategies.

4 Conclusion

In this work, we reformulated the dynamic ensemble selection (DES) problem as a multi-label classification problem and derived the actual multi-label loss associated to the DES problem. Contrary to other approaches that use state-of-art multi-label classification methods, we addressed the problem of optimizing the non-standard actual loss directly, since an analytic expression (or characterization) of the Bayes classifier that minimizes the actual DES loss is missing. We

Table 6. Average number of classifiers selected by DES methods for the first and second ensemble generation strageies.

	first generation				*second generation*			
Data set	BR-DES	LP-DES	CHADE	PCC-DES	BR-DES	LP-DES	CHADE	PCC-DES
Adult	187 +/- 40	200 +/- 6	185 +/- 49	27 +/- 20	193 +/- 16	189 +/- 27	196 +/- 21	63 +/- 43
Auto Moto	125 +/- 36	122 +/- 40	118 +/- 36	106 +/- 19	161 +/- 23	138 +/- 35	165 +/- 27	136 +/- 30
BaseHock	139 +/- 42	128 +/- 46	130 +/- 44	107 +/- 24	164 +/- 24	140 +/- 38	168 +/- 27	137 +/- 24
Breast-Cancer	176 +/- 33	182 +/- 30	177 +/- 32	159 +/- 47	190 +/- 10	191 +/- 9	191 +/- 10	164 +/- 45
Colic	160 +/- 54	172 +/- 47	161 +/- 54	84 +/- 44	164 +/- 31	140 +/- 55	183 +/- 30	107 +/- 42
Colon	172 +/- 34	153 +/- 49	172 +/- 35	161 +/- 39	154 +/- 38	144 +/- 45	155 +/- 38	145 +/- 32
Credit Approval	159 +/- 37	166 +/- 40	161 +/- 38	110 +/- 63	173 +/- 23	153 +/- 39	178 +/- 25	111 +/- 43
Elecrypt	135 +/- 40	128 +/- 46	135 +/- 41	102 +/- 42	167 +/- 20	140 +/- 45	181 +/- 20	122 +/- 38
German Credit	166 +/- 64	187 +/- 43	169 +/- 68	21 +/- 12	171 +/- 42	145 +/- 56	179 +/- 52	51 +/- 33
Gunmid	135 +/- 36	120 +/- 37	129 +/- 37	90 +/- 33	149 +/- 24	124 +/- 35	150 +/- 36	82 +/- 19
Hepatitis	195 +/- 20	194 +/- 30	196 +/- 21	94 +/- 61	195 +/- 10	159 +/- 53	198 +/- 1	126 +/- 60
Ionosphere	173 +/- 44	188 +/- 22	172 +/- 48	39 +/- 17	191 +/- 15	187 +/- 25	196 +/- 8	121 +/- 63
krvskp	144 +/- 46	151 +/- 49	149 +/- 45	82 +/- 24	167 +/- 18	154 +/- 29	172 +/- 21	151 +/- 23
Madelon	115 +/- 50	99 +/- 59	116 +/- 65	47 +/- 27	102 +/- 27	100 +/- 32	112 +/- 39	55 +/- 9
Ovarian	125 +/- 45	118 +/- 44	121 +/- 43	103 +/- 37	161 +/- 31	140 +/- 27	162 +/- 31	123 +/- 25
PcMac	144 +/- 33	120 +/- 40	139 +/- 33	100 +/- 24	162 +/- 23	131 +/- 36	163 +/- 24	125 +/- 12
Relathe	154 +/- 54	133 +/- 58	170 +/- 46	88 +/- 17	170 +/- 28	139 +/- 39	185 +/- 27	122 +/- 23
Sonar	149 +/- 46	149 +/- 48	147 +/- 52	65 +/- 33	163 +/- 31	142 +/- 44	182 +/- 29	86 +/- 43
Spambase	172 +/- 41	198 +/- 13	180 +/- 36	63 +/- 28	189 +/- 14	180 +/- 29	198 +/- 4	112 +/- 36
Vote	168 +/- 26	166 +/- 31	168 +/- 26	160 +/- 40	185 +/- 13	185 +/- 16	186 +/- 12	165 +/- 32
Mean	152 +/- 48	152 +/- 52	153 +/- 51	85 +/- 50	168 +/- 32	151 +/- 44	175 +/- 34	112 +/- 49

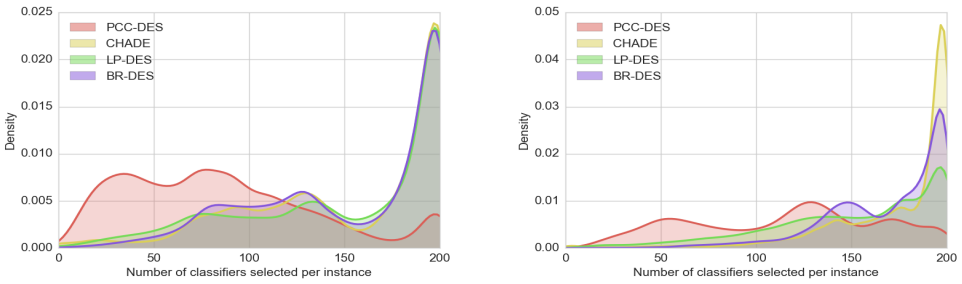

Fig. 3. Histogram of the number of classifiers selected per instance, by each DES method with the first and second ensemble generation strategies.

showed that the dependencies of the errors made by each model in the ensemble have to be exploited to optimize this loss. As the problem is intractable for realistic ensemble sizes, we discussed a more sophisticated multi-label procedure based on Probabilistic Classifier Chains and Monte Carlo sampling capable that allows to minimize the actual loss function directly. The experimental results on 20 benchmark data sets demonstrated the effectiveness of the proposed method against competitive alternatives using standard "off-the-shelf" multi-label learning techniques. Our experimental results show that optimizing the actual DES loss pays off in terms of performance. Compared to all others DES approaches, the proposed method was found to select a significantly smaller number of models, especially in the presence of many weak models. Future work should aim

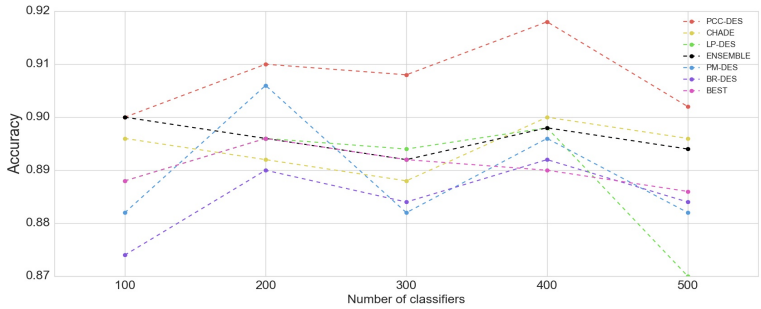

Fig. 4. Accuracy averaged over 20 data sets, as a function of the ensemble size.

to characterize the Bayes classifier for the actual DES loss in order to reduce the computational burden of the training phase and to increase the performance further.

Bibliography

[1] Uri Alon, Naama Barkai, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *National Academy of Sciences*, 1999.

[2] C.L Blake and C.J Merz. Uci repository of machine learning databases, 1998.

[3] Alceu S. Britto, Jr., Robert Sabourin, and Luiz E. S. Oliveira. Dynamic selection of classifiers-a comprehensive review. *Pattern Recogn.*, 47(11):3665–3680, 2014.

[4] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *ICML*, 2004.

[5] Weiwei Cheng, Eyke Hüllermeier, and Krzysztof J Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 279–286, 2010.

[6] Rafael MO Cruz, Robert Sabourin, George DC Cavalcanti, and Tsang Ing Ren. Meta-des: A dynamic ensemble selection framework using meta-learning. *Pattern Recognition*, 48(5):1925–1935, 2015.

[7] Krzysztof Dembczynski, Arkadiusz Jachnik, et al. Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, volume 28, pages 1130–1138, 2013.

[8] Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1):5–45, 2012.

[9] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006.

[10] Thomas G. Dietterich. Ensemble methods in machine learning. In *First International Workshop on Multiple Classifier Systems*, pages 1–15, 2000.

[11] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[12] Albert H.R. Ko, Robert Sabourin, and Alceu Souza Britto Jr. K-nearest oracle for dynamic ensemble selection. In *ICDAR*, pages 422–426, 2007.

[13] Ludmila I. Kuncheva. Clustering-and-selection model for classifier combination. In *KES*, pages 185–188, 2000.

[14] Nan Li, Yang Yu, and Zhi-Hua Zhou. Diversity regularized ensemble pruning. In *ECML PKDD*, pages 330–345, 2012.

[15] Dragos D. Margineantu and Thomas G. Dietterich. Pruning adaptive boosting. In *ICML*, pages 211–218, 1997.

[16] Fotini Markatopoulou, Grigorios Tsoumakas, and other. Instance-based ensemble pruning via multi-label classification. In *ICTAI*, 2010.

[17] Fotini Markatopoulou, Grigorios Tsoumakas, and Ioannis P. Vlahavas. Dynamic ensemble pruning based on multi-label classification. *Neurocomputing*, 150:501–512, 2015.

[18] Gonzalo Martínez-Muñoz, Daniel Hernández-Lobato, and Alberto Suárez. An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):245–259, 2009.

[19] Deiner Mena, José Ramón Quevedo, Elena Montañés, and Juan José del Coz. A heuristic in a* for inference in nonlinear probabilistic classifier chains. *Knowl.-Based Syst.*, 126:78–90, 2017.

[20] Fábio Pinto, Carlos Soares, and João Mendes-Moreira. Chade: Metalearning with classifier chains for dynamic combination of classifiers. In *ECML PKDD*, pages 410–425, 2016.

[21] Jesse Read, Luca Martino, and David Luengo. Efficient monte carlo methods for multi-dimensional learning with classifier chains. *Pattern Recognition*, 47(3):1535–1546, 2014.

[22] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

[23] J Rennie. Newsgroups data set, sorted by date, 2000.

[24] Dymitr Ruta and Bogdan Gabrys. Classifier selection for majority voting. *Information Fusion*, 6(1):63–81, 2005.

[25] Michèl Schummer, WaiLap V Ng, Roger E Bumgarner, et al. Comparative hybridization of an array of 21 500 ovarian cdnas for the discovery of genes overexpressed in ovarian carcinomas. *Gene*, 238(2):375–385, 1999.

[26] Kevin Woods, W. Philip Kegelmeyer, and Kevin Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE transactions on pattern analysis and machine intelligence*, 1997.

[27] Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[28] Z. Zhao, F. Morstatter, et al. Advancing feature selection research–asu feature selection repository. *School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe*, 2010.

[29] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: Many could be better than all. *Artif. Intell.*, 137(1-2):239–263, 2002.