Dynamic Ensemble Selection for Quantification Tasks

Pablo Pérez-Gállego, Alberto Castaño, José Ramón Quevedo, Juan José del Coz

Artificial Intelligence Center, University of Oviedo, Gijón (Spain)

Abstract

Ensembles are among the most effective and successful methods for almost all supervised tasks. Not long ago, an ensemble approach has been proposed for quantification learning. The idea of such method is to exploit the prior knowledge about quantification tasks, building ensembles in which diversity is achieved by training each model with a different distribution. These training samples are generated taking into account the expected drift in class distribution. This paper extends this method proposing three new quantifier selection criteria particularly devised for quantification problems, where two of them are defined for dynamic ensemble selection. The experiments demonstrate that, in many cases, these selection functions outperform straightforward approaches, like averaging all models and using quantification accuracy to prune the ensemble. Moreover, the results show that performance heavily depends on the combination of the base quantification algorithm and the selection measure.

Keywords: Quantification, Ensembles, Dynamic Ensemble Selection

1. Introduction

Quantification learning is a fairly new supervised task introduced by Forman [1]. It is aimed at obtaining models able to predict an aggregate estimate for a collection of instances, without providing individual predictions for each of them. A typical example is related to sentiment analysis in social

Email address: {pablopg,castanoalberto,quevedo,juanjo}@uniovi.es (Pablo Pérez-Gállego, Alberto Castaño, José Ramón Quevedo, Juan José del Coz)

networks [2], e.g., to estimate the percentage of positive comments about a certain topic of interest during a concrete time period [3].

Interestingly, quantification methods have been lately applied to solve several real-world problems from different application areas. For instance, automatic methods based on quantification have been used to predict the proportion of membrane-intact sperm cells, which it is one of the most important aspects analyzed in fertility studies and was traditionally carried out by human experts [4]. Quantifiers have been also applied to quantify recurrent issues based on the analysis of technical-support call logs recorded by customer services [5]. Early detection of such issues is useful, among others things, to identify problems that may become epidemics. In Epidemiology, [6] proposed a quantification approach to estimate the cause of death distribution in a population using verbal autopsies. A verbal autopsy is a procedure that helps to determine the cause of a death recollecting the deceased's health information/symptoms, and also the events prior to death from relatives, friends or other informants. This method is cheaper because physician reviews are not required. Plankton abundance estimation [7, 8] is another paradigmatic quantification task, whose goal is to estimate the abundance of some taxonomic groups given plankton samples taken from the oceans.

Ensemble learning is undoubtedly one of the most effective learning approaches –is the winner of many challenges– and has been adapted to tackle almost any kind of supervised task. An ensemble consists of a group of models whose predictions are combined to produce a final, single response. Usually, an ensemble method comprises at least two main steps: 1) in the training phase, the ensemble models are obtained using some base learning algorithm; and 2) in the prediction phase, the outputs of these models are combined to produce a collective decision for a given unseen case. This fusion strategy mainly depends on the learning task, being majority voting the predominant rule for classification, and averaging for regression and probability estimation. In any case, ensemble models can be treated equally or not, assigning a different weight to each one that is is usually proportional to the accuracy/precision of the model.

In the past years, an additional intermediate step, consisting in selecting only some models before combining them, has been studied in depth [9]. The goal of this process, named as ensemble pruning [10, 11, 12] or ensemble selection [13, 14, 15], depending on the author, is to improve ensembles' accuracy and complexity by just using some models instead of all [16]. Ensemble model selection can be further divided into two phases: 1) to define a function or criterion for evaluating/ranking the models and 2) to use a search algorithm to find the best group of models. Obviously, the definition of the selection measure is a crucial component of the whole process. Competence and diversity are the usual concepts applied to devise such functions. Besides, ensemble selection methods can be either static or dynamic [9, 12**?** , 18]. In the former case, a fixed subset of models is selected (the rest of models will never be used), while in the latter, a different group of models may be selected for each testing example (no model is discarded). Static selection has the advantage of being computationally more efficient because it needs less memory to store the final models and predictions can be delivered faster. The rationale of using dynamic approaches is that each testing case may require the use of different models, or put another way, each model has its own area of competence within the input space [9]. [14, 15**?**] provide nice reviews on ensemble selection and dynamic selection methods, proposing different taxonomies.

Recently, the use of ensembles has been extended to quantification learning [19]. As we shall discuss in Section 2, one of the main characteristics of quantification tasks is that data distribution changes between training and testing. By problem definition, it is known that at least the class probabilities, $P(y)$, change. The idea in [19] is to exploit this prior knowledge to build ensembles of quantifiers (EoQ) in which each model is trained with a different class distribution. From a conceptual point of view, this approach obeys one of the main ensemble learning principles, which is to build a collection of models with some kind of diversity [10, 21, 22]. In this approach, the concept of diversity is appropriately adapted to quantification problems. The expected behavior is that the ensemble will contain some models that were trained with a class distribution similar to the one observed in a new unseen test sample. In their paper, the authors employ the simplest combination strategy: averaging the prediction of all models. Despite this simplicity, the proposed ensemble algorithm significantly outperforms current state-of-the-art quantification methods. This result suggests that EoQ are already effective for quantification problems. However, their performance could be further improved using more sophisticated fusion strategies like the ones applied for classification and regression tasks.

The main contribution of this paper is to propose three new selection criteria particularly devised for quantification problems and for some concrete quantification algorithms. One of them is static and the other two are defined for dynamic ensemble selection. All of them are appropriate to be used with ranking-based selection methods [14], that is, those that are based on ranking the models according to some evaluation metric and selecting afterwards the best subset following that order. Besides, the paper presents an exhaustive experimental study in which EoQ are analyzed from different perspectives. First, the ensembles are built using five different base quantifiers that follow contrasting approaches, extending the experiments reported in [19]. Then, each of them is applied with four selection measures (the three criteria proposed here and one additional straightforward static criterion based on quantification accuracy). The aim is to study which selection criterion is more appropriate for each base quantifier. Moreover, dynamic ensemble selection methods are compared with the static selection approach. This exhaustiveness allows us to extract some interesting conclusions on the design of ensembles for quantification problems, establishing a solid point of departure for future research in the field of EoQ.

The rest of this manuscript is organized as follows. Section 2 briefly reviews quantification learning, introducing the notation, performance measures and describing some quantification methods, including the one based on EoQ. Section 3 introduces the new quantifier selection measures proposed in the paper. The experiments performed and the results are discussed in Section 4. The paper ends with some conclusions and directions for future work.

2. Quantification learning

Given a training set $D = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, in which $\boldsymbol{x}_i \in \mathcal{X}$ and $y_i \in \{+1, -1\}$, the goal of binary quantification is to induce a model or quantifier

$$\bar{h} : \mathbb{N}^{\mathcal{X}} \longrightarrow [0, 1], \tag{1}$$

able to estimate the prevalence, $\hat{p}$, of the positive class for an unseen test set, T, i.e. $\hat{p} = \bar{h}(T) = \mathbf{P}_T(y = +1)$.

There are two main differences between classification and quantification:

1. A quantifier produces estimates for *bags* or samples (groups of instances), instead of making predictions for individual examples like a classifier does. In the previous definition, $\mathbb{N}^{\mathcal{X}}$ represents a multiset, i.e., a test sample may contain duplicate instances according to the input space $\mathcal{X}$.

2. Many classification algorithms assume that the distribution does not change. On the contrary, the class distribution in quantification tasks changes between training and testing, $\mathbf{P}_D(y=+1) \neq \mathbf{P}_T(y=+1)$, by the own definition of these tasks. Otherwise quantification tasks would be trivial: a quantifier would predict just the prevalence of the positives in the training set: $\hat{p} = \bar{h}(T) = \mathbf{P}_D(y = +1)$.

This last aspect is crucial to differentiate quantification from classification and it is the reason that a classifier trained assuming that the distribution does not change is suboptimal [3, 24] in quantification tasks when the classify and count approach is used (see Section 2.2). In fact, characterizing the expected changes in data distribution is the first step to designing new quantification algorithms. Assuming that there is a drift in the distribution, in symbols $\mathbf{P}_D(\boldsymbol{x}, y) \neq \mathbf{P}_T(\boldsymbol{x}, y)$, and if we represent $\mathbf{P}_T(\boldsymbol{x}, y) = \mathbf{P}(y) \cdot \mathbf{P}(\boldsymbol{x}|y)$, we know that, by the own definition of quantification problems, $\mathbf{P}(y)$ changes, and the main question is to decide whether $\mathbf{P}(\boldsymbol{x}|y)$ changes or not. Most quantification algorithms, as we shall see below, assume that $\mathbf{P}(\boldsymbol{x}|y)$ remains constant.

Notice that changes in the distribution also occur in other learning problems, including *covariate shift*, where $\mathbf{P}(\boldsymbol{x})$ changes but $\mathbf{P}(y|\boldsymbol{x})$ remains constant, and *concept drift* [**?**] where $\mathbf{P}(y|\boldsymbol{x})$ changes but $\mathbf{P}(\boldsymbol{x})$ does not. In general, most problems that deal with data streams mining suffer some kind of drift in the distribution.

2.1. Performance evaluation

The performance of a quantifier is assessed in terms of the error for a random bag or sample. Therefore, the difference with respect to classification is that the unit of evaluation is a sample, not an individual instance. This makes quantifier evaluation a more complex task because we need a representative group of samples, otherwise the performance of a quantifier will not be accurately estimated. Given a collection of testing samples, $\{T_1, \ldots, T_s\}$, the performance of a quantifier, $\bar{h}$, is computed as:

$$Performance(\bar{h}, L, \{T_1, \ldots, T_s\}) = \frac{1}{s} \sum_{j=1}^{s} L(\bar{h}, T_j), \tag{2}$$

in which $L(\cdot, \cdot)$ represents a quantification loss function. In some cases, the experimental design may require adapted versions of well-known validation methods, like cross-validation [23]. Usually L must compare the predicted class distribution and the actual one. In the case of binary quantification, it is enough to compute the difference between the predicted prevalence of the positive class, $\hat{p}$, and the actual prevalence, p. Then, one may use any loss function devised for regression problems, for instance, the absolute error, $AE(\bar{h}, T) = |\hat{p} - p|$, and the squared error, $SE(\bar{h}, T) = (\hat{p} - p)^2$. In this paper we shall employ both to define our two target performance measures for binary quantification, **Mean Absolute Error** (MAE) and **Mean Square Error** (MSE):

$$MAE(\bar{h}, \{T_1, \dots, T_s\}) = \frac{1}{s} \sum_{j=1}^{s} |\hat{p}_j - p_j|, \tag{3}$$

$$MSE(\bar{h}, \{T_1, \dots, T_s\}) = \frac{1}{s} \sum_{j=1}^{s} (\hat{p}_j - p_j)^2. \tag{4}$$

MAE is easier to interpret than MSE, but it is less sensitive to large errors. See [**? ? ? ?**] for other performance measures commonly used in quantification papers .

2.2. Quantification methods

This section just describes the quantification algorithms used in the experiments of the paper (Section 4), for a complete review see [24]. The straightforward approach for quantification learning is the CC (*Classify* and *Count*) method [25]. It works as follows: 1) a classifier is induced from D, 2) such classifier is used to classify a testing sample, T, and 3) the number of examples predicted as positives is counted in order to compute $\hat{p}$. The main drawback of this approach is that the training phase does not take into account the fact that the data distribution will change. In fact, CC usually performs worse when the drift in distribution increases [3].

The AC (*Adjusted Count*) method [25] tries to overcome the issues of the CC approach. It is based on the learning assumption that the class probability distribution $\mathbf{P}(y)$ changes, as discussed before, but $\mathbf{P}(\boldsymbol{x}|y)$ is constant, i.e. $\mathbf{P}_D(\boldsymbol{x}|y) = \mathbf{P}_T(\boldsymbol{x}|y)$. If such is the case, the prevalence perceived by the CC method can be expressed in terms of the actual prevalence p:

$$\hat{p}_{CC} = tpr \cdot \ p + fpr \cdot \ (1 - p), \tag{5}$$

in which tpr and fpr are the *true positive rate* and the *false positive rate* respectively. It is worth noting that these two rates remain constant when $\mathbf{P}(\boldsymbol{x}|y)$ does not change. Thus, the above expression is quite useful because if we estimate tpr and fpr, using for instance cross validation as Forman suggests [25] or a separate validation set, we can compute the actual prevalence solving for p in (5):

$$p = \frac{\hat{p}_{CC} - fpr}{tpr - fpr}. \tag{6}$$

Then, AC comprises four steps: 1) to train a classifier using D, 2) to estimate its rates (tpr and fpr), 3) to classify and count the testing sample, T, and 4) to correct the prevalence obtained in the previous step by applying (6). The key step is the estimation of tpr and fpr, in fact, if both estimates are accurate, AC makes perfect predictions independently of the accuracy of the underlying classifier. However, it is unrealistic to obtain perfect estimates for tpr and fpr because, either the $\mathbf{P}(\boldsymbol{x}|y)$ assumption does not hold, or because there is some bias in the own estimation process. Thus, AC usually produces imperfect quantifications despite its solid theoretical foundations.

In [26], the authors introduce the probabilistic versions of CC and AC methods, denoted here as Probabilistic Classify & Count (PCC) and Probabilistic Adjusted Count (PAC), respectively. The main difference between PCC and CC is that the later uses crisp classifications as we have seen before, and the former requires a probabilistic classifier. Such classifier usually returns the probability of belonging to the positive class, so PCC computes the prevalence of the positives as the average of such probabilities over the testing set. In symbols:

$$\hat{p}_{PCC} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{P}(y_i = +1|\boldsymbol{x}_i). \tag{7}$$

PCC presents similar issues to CC, mainly that ignores the expected changes in the distribution. Then, the same authors propose the PAC quantifier [26] that adjusts the estimated provided by PCC using a probabilistic version of (6):

$$\hat{p}_{PAC} = \frac{\hat{p}_{PCC} - FP^{pa}}{TP^{pa} - FP^{pa}}\ , \tag{8}$$

in which $TP^{pa} = \frac{\sum_{i \in D^+} \mathbf{P}(y_i=+1|\boldsymbol{x}_i)}{|D^+|}$ is the averaged probability of the positive examples and $FP^{pa} = \frac{\sum_{i \in D^-} \mathbf{P}(y_i=+1|\boldsymbol{x}_i)}{|D^-|}$ is the averaged probability of the negatives. Both rates are again estimated using cross validation or a validation set. D^+ and D^- represent the set of positive and negative examples in D, respectively.

The last method, called HDy [27], follows a completely different approach as it is not based on counting and correcting. The key idea consists in representing and comparing training and testing distributions. The acronym merges the Hellinger Distance (HD), employed to measure the similarity between both distributions, and the predicted values, y, used to represent the distributions. HDy starts learning a classifier using D. A probabilistic classifier is preferred because its outputs are bounded in the range [0..1]. This classifier is applied to collect the predicted values for the training instances. The goal is to obtain two distributions, one for the positive and one for the negative examples: the range [0..1] is partitioned into b bins, and two histograms are built in which each training example is assigned to the corresponding bin depending on its probabilistic score. The same procedure is applied over the test set, but in this case just one histogram is obtained because the class values are unknown. Once we have the three distributions (positive, negative and testing), HDy employs a linear search seeking for the value $\hat{p} \in [0..1]$ that makes the combination of the positive distribution and the negative distribution most similar to the test distribution according to the Hellinger Distance. The combination for each value of $\hat{p}$ is obtained using:

$$\frac{|D'_i|}{|D'|} = \frac{\left|D_i^+\right|}{|D^+|} \cdot \hat{p} + \frac{\left|D_i^-\right|}{|D^-|} \cdot (1 - \hat{p}), \tag{9}$$

in which $|D^+|(|D^-|)$ is the number of examples in D belonging to the positive(negative) class and $\left|D_i^+\right|(\left|D_i^-\right|)$ the number of positive(negative) examples in D belonging to the i-th bin. Notice that in this procedure all the bins are uniformly modified using $\hat{p}$ assuming that $\mathbf{P}(\boldsymbol{x}|y)$ does not change.

Similarity between both histograms is measured with the Hellinger Distance defined for discrete distributions:

$$HD(D', T) = \sqrt{\sum_{i=1}^{b} \left(\sqrt{\frac{|D'_i|}{|D'|}} - \sqrt{\frac{|T_i|}{|T|}} \right)^2}, \tag{10}$$

in which D' and T represent the combined distribution of D (9) and the testing distribution, respectively.

2.3. Ensembles for quantification learning

In [19] the authors present the first ensemble approach devised for quantification. The main hypothesis of the paper is that ensembles are a good alternative for those problems in which the data distribution changes, like it occurs in quantification tasks. The reason is that each model of the ensemble can be trained with a training set whose data distribution may be different from the distribution of the original dataset. This fact can be further exploited in learning problems that suffer a characterizable drift in the distribution. The main idea is to generate a collection of training samples (one for training each ensemble model) having different data distributions, thus taking into account the expected drift. The aim is to introduce enough diversity [10, 21, 22] taking advantage of our prior knowledge about the problem. Notice that this approach differs from those methods based on removing/adding models to the ensemble because these ensembles do not need further modifications once they have been trained.

The first step of the method is to generate the training samples for the models. Each sample has a different prevalence (selected according to the expected change in the class distribution) but its examples are chosen using random sampling with replacement to ensure that $\mathbf{P}(\boldsymbol{x}|y)$ does not change (adopting the same assumption that other quantification algorithms discussed before). In the next phase, the models of the ensemble are trained using a base quantification algorithm with the samples generated in the previous step. In the prediction phase, all models are applied over the test set T and the predicted prevalences are aggregated using the arithmetic mean.

3. Selection measures

As it was stated in the Introduction, there are several ways of combining EoQ. In contrast to using all models with the same importance, the main alternatives are ensemble weighting and ensemble selection. The former consists in assigning different weights to the models comprising the ensemble. The idea is that the best models contribute more to the final prediction than the worst ones. The second method is based on selecting the best subset of models, discarding the rest. Notice that they can be used together, selecting the best models but combining them with different weights. From a conceptual point of view, both approaches present a comparable complexity in the sense that they may have the same two elements: a selection/ranking criterion to assess and compare the model contribution to the ensemble and a method able to assign the concrete weight to each model or to select the best subset of models. Despite there may be some interaction between both elements, in the sense that a particular criterion would fit better for a selection method, in the literature these two problems are usually solved separately to reduce the complexity of obtaining optimal ensembles [28]. In fact, assigning optimal weights is a problem hard to solve and the search problem to find the best group of models is NP-complete [29]: the searching space has $2^m - 1$ different non-empty subsets, being m the ensemble size.

Following the above reasoning, this paper just focuses on selection measures/criteria: the main aim is not to obtain the best possible combination strategy with optimal weights or with an optimal selection, but to study what kind of criterion works better for selecting or ranking the ensemble models. Despite we acknowledge that there may exist a possible interaction between both, the selection/ranking criteria can be better analyzed if they work in isolation, without combining them with additional algorithms to select models or to assign weights. Otherwise, the analysis of the combination of both elements would hinder the suitability of a particular ranking criterion given a quantification algorithm, which it is our goal. The analysis of the three elements together (selection criteria, selection methods and quantification algorithms) would make more difficult to extract useful conclusions. Thus, the aim of this paper is to analyze the interaction between selection criteria and state-of-the-art quantification methods. Specifically, we shall focus on the quality of the rankings provided by different criteria studying, for instance, the performance of different subsets of models, increasing the size by including first those models with the highest rankings. Our primal interest is to find out which is the best selection measure for some base quantifiers. As will be discussed later, we are particularly interested in analyzing criteria for dynamic selection in the context of EoQ. For these reasons, the use of different selection methods proposed in the literature, like hill-climbing search[13], meta-learning [**?**], clustering [30], probabilistic models [**?**], genetic algorithms [34], quadratic integer programming [35] or hybrid methods [9], just to cite a few, is out of the scope of the paper.

Two different groups of selection measures are considered in this paper. The first group leads to static ensemble selection, meaning that the same subset of models is used for all testing samples, discarding the rest. The second group allows dynamic selection, that is, applying a different subset for each sample. Static selection presents same advantages, mainly that predictions can be delivered faster and less memory is required to store the models. On the other hand, dynamic ensemble selection allows a flexible structure of the ensemble and an adaptive behavior [9].

3.1. Static selection measures

The most commonly used static selection measures are based on the accuracy/precision of the models. The idea is straightforward: select the best (strongest) models, discarding the worst (weakest) ones. This kind of selection criterion is defined in terms of performance measures. For instance, accuracy is a natural choice for classification problems: select the models with the highest accuracy. We also apply here this approach, named as ACC, but adapted to binary quantification. This means to use a binary quantification measure instead of classification accuracy. In our experiments, the models are ranked and selected according to their MSE scores (4). It is worth noting that, in addition to the selected performance measure, it is important to define the validation process to estimate such errors. As it was discussed in Section 2.1, in quantification, it is essential to employ a representative collection of samples with a suitable range of prevalences. In our experiments, error estimation for each model is made by using the training samples generated for the rest of models.

This paper introduces a new selection criterion that it is related to ACC, but adapted to the characteristics of some paradigmatic quantification algorithms. This selection measure is called MAX, and is especially devised for AC and PAC quantifiers. MAX was inspired by the method of the same name proposed in [25]. The idea is to select those models that maximize the difference $tpr - fpr$ for the AC method and $TP^{pa} - FP^{pa}$ for the PAC method. This means that MAX measure prefers those models with a large value for the denominator in the correction equations, (6) and (8). A smaller value in the denominator implies a bigger and risky correction. It is preferable a smooth correction, meaning that the prediction of CC/PCC is already rather accurate.

Notice that this criterion is somehow similar to ACC, but at the same time, presents subtle differences that become relevant for imbalanced domains in which the class distribution is skewed, since examples of one class (usually the negative class) appear much more frequently. For balanced domains, a model with a higher accuracy will also have a high tpr and a low fpr, resulting in a large value for the difference between both. But in an imbalanced situation, one model may have a high accuracy, but a small value for $tpr-fpr$, for instance if the model predicts always the negative (majority) class ($tpr = 0$, $fpr = 0$).

3.2. Dynamic selection measures

The main characteristic of quantification problems, the fact that data distribution changes, makes that dynamic quantifier ensemble selection [**?**] seems a much more appealing approach than static selection. Our proposal is to select those models that were trained using a training distribution more similar to the distribution of the testing sample. The key issue is how to compare such distributions in a computationally efficient manner, avoiding other complex approaches like, for instance, density estimation [36, 37]. This paper proposes two different criteria.

The first selection criterion assumes that both distributions are similar when the prevalence of the positive class is also similar in both distributions. The problem is that we know the prevalence of the positive class in the training distribution, but we totally ignore the prevalence of the testing sample. This is precisely what we want to estimate. Our idea is to use all the models in the ensemble to obtain a first estimate of the prevalence for the testing sample. Then, in a second round, we will rank the models according to the difference between the prevalence of the their corresponding training samples and the first estimation obtained using all the models. The aim is to select the models that are close to such first estimate, discarding those that are far away.

This criterion, called P_{tr} in the experiments, seems very simplistic at first sight, but is closely related to several formal algorithms, namely AC, PAC and HDy. Actually, P_{tr} makes exactly the same learning assumption that those quantification algorithms, that is, P_{tr} also assumes that $\mathbf{P}(\boldsymbol{x}|y)$ does not change. When such assumption holds, a change in the data distribution is only due to a variation in the prevalence of the classes. Notice that HDy follows a similar idea when applies linear search to obtain the value of the prevalence that makes the combination of the positive distribution and the negative distribution most similar to the testing distribution (9).

The second criterion, called DS (Distribution Similarity), is inspired in the HDy algorithm. The idea is to compare the distribution of the y values for the testing and the training samples. The procedure is analogous to the one described above for the HDy algorithm. First, during the training phase, the training distribution of each ensemble model is summarized computing the histogram of its y values. The same y-histogram is computed for the testing sample, and both histograms are compared using the Hellinger distance. The models are then ranked according to these distances.

Notice that, from a computational point of view, P_{tr} and DS are very similar. Both require to apply all models over the testing sample: in the former case, to calculate a first estimation for the testing sample prevalence, and in the latter, to obtain the distribution of the testing examples output values. Comparing these two approaches with ACC and MAX criteria, presented above, we find that P_{tr} and DS are less efficient in the testing phase, because static approaches do not require to make any process during the testing phase (the best models are selected in the training phase). And for the same reason, ACC and MAX are less efficient in the training phase because they require to estimate quantification accuracy or the difference between some rates (e.g. $tpr - fpr$).

4. Experimental results

The experiments described in this section analyze the performance of the different selection measures discussed in the previous section. The main goal is to prove that selecting a group of models by applying these proposed criteria outperforms the straightforward approach of taking the average prediction of all ensemble models. Additionally, ensembles were built considering a group of representative base quantifiers to extend the significance of the experiments. The idea was to analyze whether the best criterion depends on the base quantifier or, in contrast, one of the criterion prevails over the rest, independently of the used quantification method.

4.1. Experimental setting

Thirty two datasets were used in the experiments, exactly the same employed in [19]. Table 1 describes their main properties. Approximately half of them are multi-class problems (balance, cmc, ctg, iris, pageblocks and wine) in which one of the classes was mapped as the positive class and the others comprised the negative class. The rest are originally binary problems. The most important characteristic for binary quantification problems is the prevalence of the positive class (last column in Table 1). Notice that this value ranges from 2% to 70%, thus providing enough variability.

We considered five different base quantification algorithms to learn the ensemble models: CC, PCC, AC, PAC and HDy (see Section 2.2). The corresponding ensemble versions shall be denoted as ECC, EPCC, EAC, EPAC and EHDy, respectively. In this sense, this paper extends the results reported in [19] by the addition of EPCC and EPAC. Notice that all these

Table 1: Summary of datasets: n is the number of examples, d is the dimension of the input space, $P(N)$ the number of positive (negative) examples and p the prevalence of the positive class

name	n	d	P	N	p
balance.1	625	4	288	337	46%
balance.2	625	4	49	576	8%
balance.3	625	4	288	337	46%
breast-cancer	683	9	444	239	65%
cmc.1	1473	9	629	844	43%
cmc.2	1473	9	333	1140	23%
cmc.3	1473	9	511	962	35%
ctg.1	2126	22	1655	471	78%
ctg.2	2126	22	295	1831	14%
ctg.3	2126	22	176	1950	8%
diabetes	768	8	268	500	35%
german	1000	24	700	300	70%
haberman	306	3	81	225	26%
ionosphere	351	34	126	225	36%
iris.2	150	4	50	100	33%
iris.3	150	4	50	100	33%
mammographic	830	5	403	427	49%
pageblocks.5	5473	10	115	5358	2%
phoneme	5404	5	1586	3818	29%
semeion	1593	256	155	1438	10%
sonar	208	60	97	111	47%
spambase	4601	57	1813	2788	39%
spectf	267	44	55	212	21%
tictactoe	958	9	332	626	35%
transfusion	748	4	178	570	24%
wdbc	569	30	212	357	37%
wine.1	178	13	59	119	33%
wine.2	178	13	71	107	40%
wine.3	178	13	48	130	27%
wine-q-red	1599	11	855	744	53%
wine-q-white	4898	11	3258	1640	67%
yeast	1484	8	429	1055	29%

quantification methods need an underlying classifier. We decided to use probabilistic classifiers for all of them because some of the algorithms, namely PCC, PAC, require a probabilistic classifier, and it is also preferable for others, like HDy. Additionally, this decision allows us to ensure that all the ensembles are formed by the same models for a given dataset. Thus, despite we are just interested in studying the selection criteria for each EoQ separately, the reader could also compare all ensembles methods with the new selection measures.

The probabilistic classifier employed was Logistic Regression [38]. The regularization parameter (C) was selected through a search in the interval $[10^{-3}, \ldots, 10^{3}]$ optimizing the geometric mean using CV5x2 (cross validation, 5 folds and 2 repetitions) over the training examples. Geometric mean was chosen to deal with imbalanced datasets which frequently appear in quantification tasks. Besides, the positive class and the negative class were balanced ($-w$ parameter in LibLinear [38]) to obtain good classifiers even for severe imbalanced cases. Additionally, AC/EAC and PAC/EPAC require estimating the values of (tpr, fpr) in (6) and (TP^{pa}, FP^{pa}) in (8), respectively. This was done by means of a CV10x1 over the training data.

The number of models to build the ensembles was set to $m = 50$. The procedure for generating the training sample or bag for each model was the following: the prevalence of each training sample, p_i, was uniformly selected from $[5\% - 95\%]$ and then, the examples of the new sample were chosen using random sampling with replacement (trying to maintain $\mathbf{P}(\boldsymbol{x}|y)$ constant). The size of the sample was always equal to the size of the original training set.

We studied two aggregation strategies. Firstly, we computed the average prediction of all the models, that is, the strategy employed by [19]. This approach is denoted as ALL in the results. Our goal was to analyze if these results can be significantly improved. The second strategy is to select 50% of the models, just the best models given a new test sample according to the four selection measures discussed in the paper:

1. ACC: selects those models with the highest quantification accuracy, or lowest error measure in terms of mean square error (4),
2. MAX: picks the models that maximize the denominator of equations (6) and (8) depending on the base quantifier used: $tpr - fpr$ for ECC and EAC, and $TP^{pa} - FP^{pa}$ for the probabilistic-oriented methods (EPCC, EPAC and EHDy),
3. P_{tr}: chooses those models that were trained with a sample that had a similar prevalence than that of the testing sample, and
4. DS: selects the models whose training distribution is most similar to the distribution of the testing sample, measured in terms of the Hellinger distance. Probabilities were discretized in 8 bins to compare both distributions.

Results in Section 4.2 and Section 4.3 were computed with CV5x2 experiments. 101 samples were generated with each test fold in order to adequately measure quantification performance. The procedure to obtain these testing bags was the same than that used for generating the training samples described before: the prevalence of each testing bag was in this case uniformly selected in the range from 0% to 100% (101 bags) and the size equal to the corresponding test fold, choosing its examples using again random sampling with replacement. Thus, each reported result in the next sections corresponds to the average of 1010 quantification tests (2 repetitions × 5 folds × 101 testing bags generated per fold). In order to study the behavior of the considered approaches, we have chosen the two measures discussed in Section 2, MAE and MSE.

4.2. Results

Tables 2-6 show the MAE scores for the ensemble method using one particular base quantifier and applying different selection measures. For instance, Table 2 reports the results using CC as the base quantifier. The difference between the scores are just due to the selection strategies, because the ensemble models are the same. It is worth noting that ALL is the worst approach in this case. All selection approaches clearly boost the results of ALL. The best two methods are DS and ACC. In fact, they obtain the best results in 28 out of 32 datasets; ALL is not the best in any.

Table 3 shows the scores when PCC is the base quantifier of the ensemble method. Here, it is clear that ACC is the best performer: not only it ranks first in terms of average ranking, but it also obtains the best score for 25 datasets. The second best approach is MAX: so it seems that in this case the static selection and the criteria based on the performance of the underlying quantifiers are better than dynamic selection and those measures based on distribution similarity. Again, ALL does not obtain the best result in any of the datasets and is outperformed by all selection approaches.

Table 4 contains the MAE scores for the ensemble method based on AC quantifier. Three approaches obtain similar results with this method: DS, ALL and MAX. The difference between them is small, both in terms of average ranking and the number of wins (13, 8 and 10 wins respectively). Interestingly, MAX achieves better performance than ACC, something that did not happen with ECC and EPCC. This could be somehow expected because MAX criterion was inspired by the equation that defines AC.

Table 5 shows the results for EPAC ensemble. MAX is the selection function that achieves the best average ranking, winning in 19 cases. This result is in line with the one described before for EAC ensemble. It makes sense that MAX outperforms other selection measures because it is designed for EPAC ensembles. The results for the rest of the approaches are pretty similar, being DS the second best method.

Table 6 contains the latest scores, those corresponding to EHDy ensembles. As one could expect in this case, DS is the best criterion. However, the difference with respect to ALL and ACC is small. The worst method is MAX, maybe because the correction idea has nothing to do with EHDy ensembles.

Results for MSE are very similar to those previously discussed for MAE. The complete MSE scores have been omitted but the average rankings and their corresponding statistical analysis are included in the next section.

4.3. Statistical analysis

The obtained results may be analyzed using several statistical tests. First, we compare all the methods together taking into account their average ranks [39]. This procedure comprises two steps: 1) a Friedman test to reject/accept the null hypothesis (all approaches perform equally well from a statistical point of view) and 2) a set of pairwise comparisons using the Bergmann-Hommel test to analyze if one particular method is significantly better than other. We are mainly interested in comparing ALL with the proposed selection measures. The average rankings of all ensemble methods using MAE and MSE are in Table 7.

The Friedman test rejects the null hypothesis (so there are significant differences among the methods) except for EPAC ($p = 0.3084$) and EHDy ($p = 0.2248$), both considering MSE. Analyzing the pairwise comparisons using the Bergmann-Hommel test ($\alpha = 0.05$), we can observe that most of the significant differences occur for ECC and EPCC: all the selection measures are significantly better than ALL, except MAX when MAE is the performance measure, and P_{tr} for MSE. For the rest of ensemble models (EAC, EPAC and EHDy), the only significant difference is between ALL and ACC (MSE scores for EAC). There are other cases in which the difference is close to be significant, for instance MAX versus ALL (EAC and MAE, $p = 0.057$).

Regarding the rest of the pairwise comparisons (not involving ALL), we see that ACC is significantly better than other selection functions for ECC and EPCC ensembles, for instance ACC is significantly better than DS, P_{tr} and MAX for EPCC ensembles, but it is worse for EAC, EPAC and EHDy ensembles, e.g. ACC is significantly worse than DS(EAC), MAX(EPAC). This suggests that ACC is a better criterion for those quantifiers that are based on the classify and count approach; more sophisticated quantifiers require other specific selection measures like the ones proposed in the paper.

However, the Bergmann-Hommel test discussed before presents some issues [40], basically because it depends on the set of compared methods: the outcome of the comparison between a pair of approaches depends also on the performance of the rest of the methods included in the study. This problem can be even worse in our experiments because some of the approaches are correlated (ACC and MAX, and DS and P_{tr}). Following [40] we perform multiple comparisons using the Wilcoxon signed-rank test. The p-values are in Table 8. The Wilcoxon tests confirm that all the selection functions are significantly better than ALL for ECC and EPCC both for MAE and MSE.

This includes MAX for CC ensembles, whose difference was not significant using the Bergmann-Hommel test. However, the most interesting results from our point of view are that: 1) MAX significantly outperforms ALL for ensembles based on PAC, 2) DS also significantly improves the scores of ALL for HDy ensembles, and 3) ALL is significantly better than ACC for EAC ensembles.

The first two results point out that ad-hoc criteria, specifically designed to work with a concrete quantifier, can help to boost the performance of the ensemble methods. MAX criterion is defined to prefer those models that maximize $TP^{pa} - FP^{pa}$ which is the denominator of the correction formula (8) of PAC models. Therefore, it is reasonable that the performance of MAX for EPAC is good. Even more interesting is the case of the combination of EHDy and DS. The idea behind HDy quantifiers is to match the test distribution with the training distribution. Thus, it makes sense that best models in the EHDy ensemble are those trained with a distribution similar to the test distribution. In fact EHDy with DS obtains the best overall results. If we perform Wilcoxon signed rank tests between the best method for each ensemble we obtain that: EHDy-DS is significantly better for MAE than: ECC-DS ($p = 5.771e - 06$), EPCC-ACC ($p = 1.603e - 05$), EAC-DS ($p = 0.0011$) and EPAC-MAX ($p = 4.392e - 05$); and for MSE is also better than the rest with the following p-values: ECC-ACC ($p = 2.313e - 06$), EPCC-ACC ($p = 2.313e - 06$), EAC-ALL ($p = 0.00941$) and EPAC-MAX ($p = 0.0009569$). Be aware, however, that these differences are not just motivated by the selection measure, but by the combination of both elements.

4.4. Analysis of the selection strategy

The results discussed in the previous section were obtained selecting half of the models. The main question regarding this policy is what is the behavior of the ensemble methods when this percentage varies. The goal of this section is to graphically analyze this issue. Figures 1-5 show the performance for the ensemble method using one particular base quantifier and varying the percentage of selected models between 2% (using just the best model) and 100% (ALL strategy). The selection measure applied for each ensemble was the best performer in the previous experiment. For instance, Figure 1 depicts the MAE scores using the combination of ECC ensembles and DS selection measure for all the benchmark datasets. The difference between the scores are just due to the ability of the selection measure to rank the best models of the ensemble.

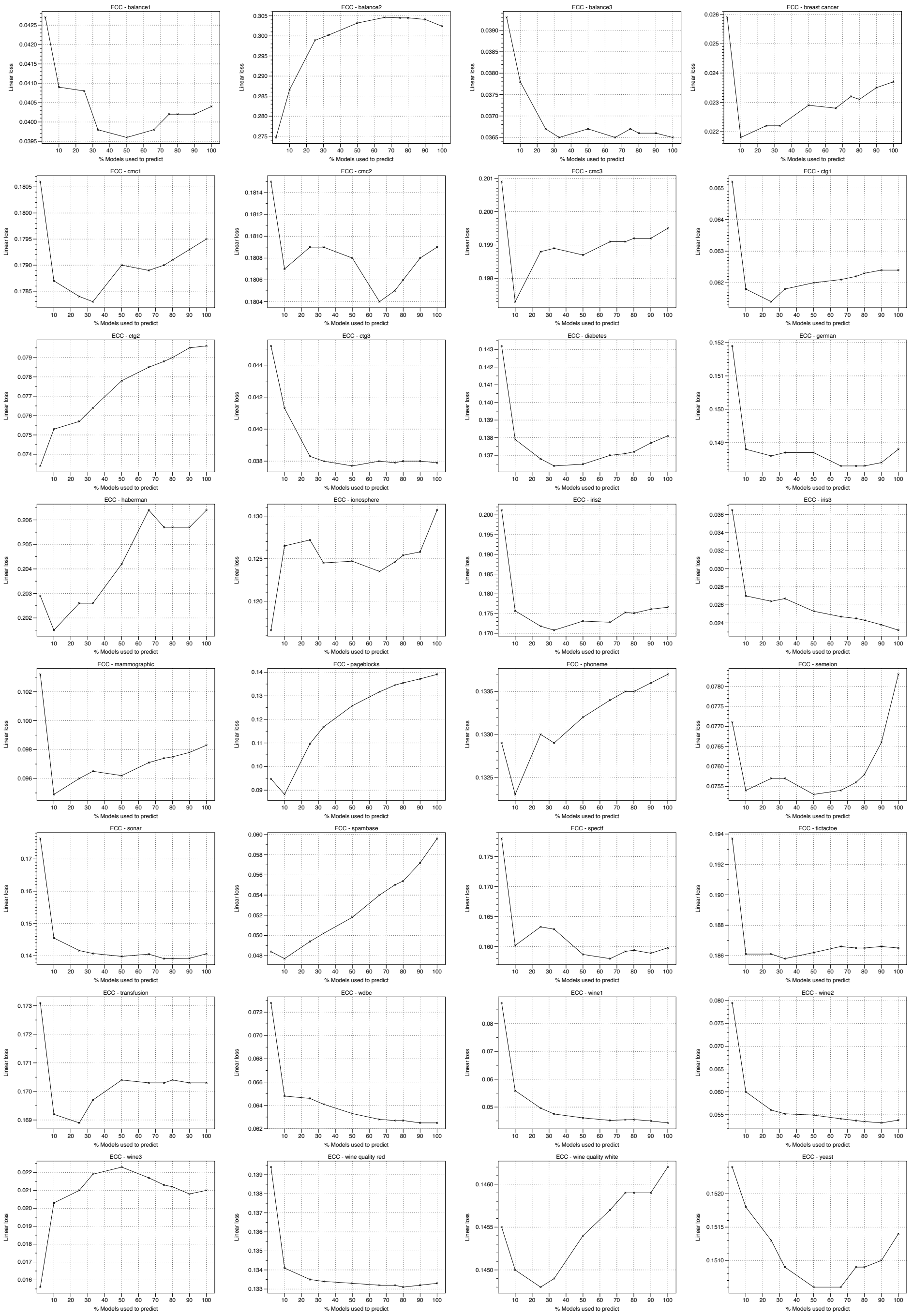

Figure 1: MAE scores selecting a different percentage of models using ECC-DS

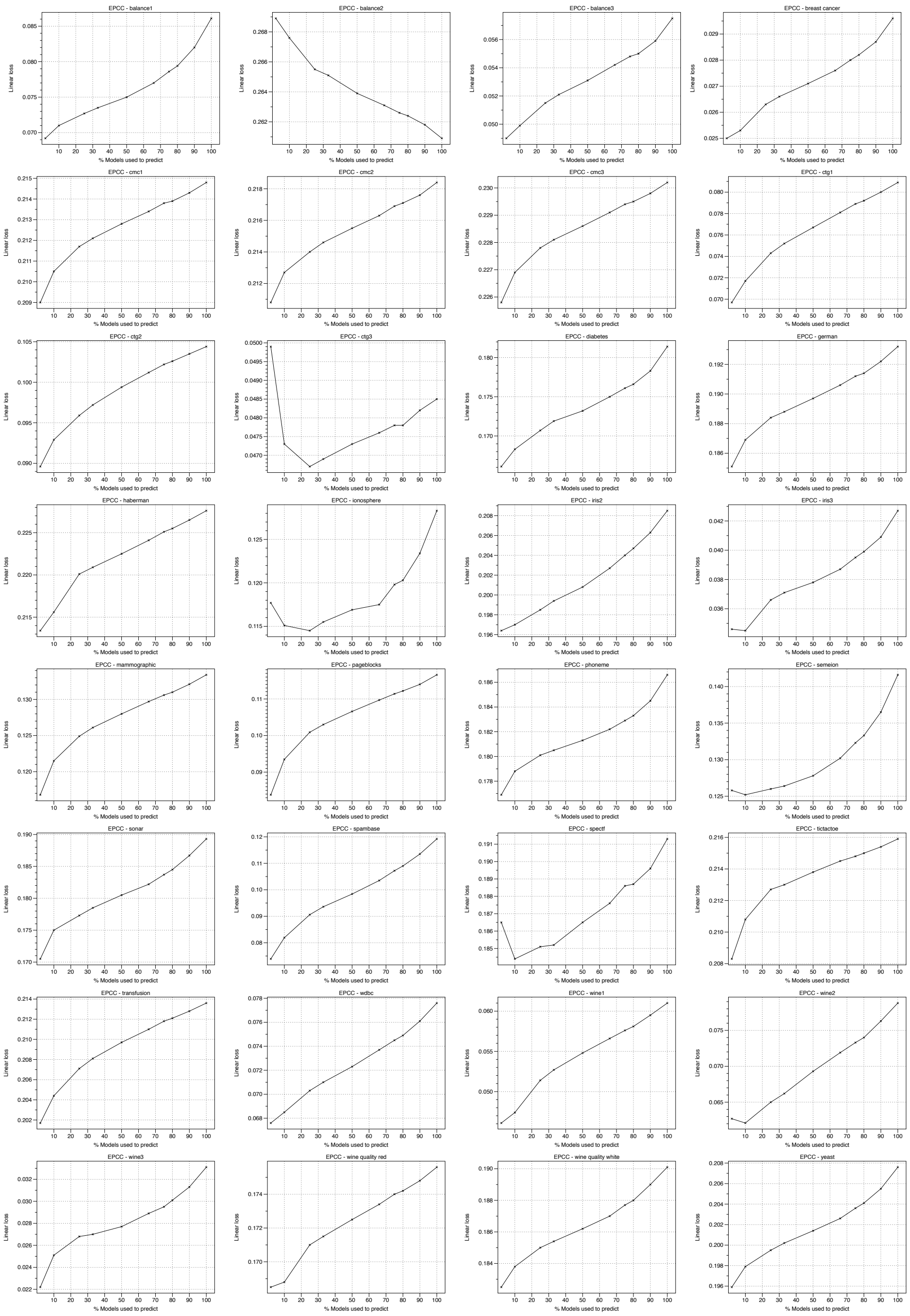

Figure 2: MAE scores selecting a different percentage of models using EPCC-ACC

Analyzing the results for ECC-DS combination (Figure 1) we can observe that taking just the best model (2%) or averaging all (100%) achieves bad results: the best model rule wins just in 4 datasets (balance2, ctg2, ionosphere and wine3) and ALL in 3 cases (iris2, wdbc and wine1). The most usual behavior is that the scores tend to improve when the percentage of selected model increases from 2% until some point in which the minimum (best) score is obtained. From that point the scores degrade again. This pattern occurs in 19 out of 32 datasets. A paradigmatic case is for instance the top left corner figure corresponding to balance1 dataset. The position of the minimum depends on the problem, sometimes is reached early between 10% and 30% (e.g. breast cancer), most of the times in the middle of the range, but it hardly happens for values greater than 70% (such cases mostly correspond for datasets in which ALL wins).

Figure 2 shows the graphs for EPCC-ACC method. These results are surprising and totally different from those observed for the rest ensemble methods. In the most common pattern, the minimum is reached with the best model and the scores degrade from there, almost linearly, when the percentage of selected models increases. This occurs in 26 datasets. There are other 4 cases (ionosphere, iris3, semeion and spectf) in which the pattern is similar, but the difference is that the minimum is reached taking between 10% and 25% of the models. Only 2 datasets show a clearly different behavior (balance2 and ctg3). These results suggest that EPCC ensembles contain many poor PCC models, maybe because the base classifier provides biased probability estimates.

Figure 3 contains the results of the ensemble method based on AC quantifiers selecting the models using the DS criterion. Here, we observe that taking all the models produces good results in more cases than in the two previous methods. ALL rule attains the best result in 6 datasets (cmc1, german, ionosphere, pageblocks, sonar and spambase) and its score is very close to the minimum in another 6 (ctg2, diabetes, haberman, iris2, spectf and wine2). Despite the behavior is somehow unstable (different patterns and none clearly prevails), we can identify a pattern that occurs in near half of the cases. We called it the “L” pattern: it starts with a bad result for the best model (2%), then the scores improve rapidly achieving the minimum around 30% and the curve remains almost flat from that point. This happens for approximately half of the datasets.

Figure 4 depicts the scores of the EPAC methods using the MAX selection function. The performance of ALL is similar to the case of EAC ensembles:

it is the best performer in 4 domains (balance2, german, iris2 and semeion) and achieves very good scores in another 10 datasets (balance3, cmc3, ctg1, diabetes, iris3, phoneme, wdbc, wine1, wine quality red and white). However, here the “L” pattern appears more frequently. This implies that most of the models in an ensemble based on PAC quantifiers should be reasonably good. These results can be explained by the fact that PAC quantifiers should be much better than the corresponding PCC methods due to the correction procedure. Recall that EPCC and EPAC use the same models. Nevertheless, there are also a few domains in which ALL performs quite poorly, noticeably ctg3, pageblocks, spampbase, tictactoe, transfusion and wine3.

Finally, Figure 5 shows the results of EHDy-DS combination. Here, the “L” pattern is again present in more than half of the graphs. As before, this kind of behavior tends to appear more often for the ensembles based on sophisticated quantifiers (AC, PAC and HDy) than for those based on the Classify & Count approach (CC and PCC).

5. Conclusions and future work

This paper extends the still embryonic work on ensembles for quantification learning. Here, three new selection criteria for ensembles of quantifiers are proposed, allowing to implement both static and dynamic ensemble selection methods. Moreover, all the proposed criteria are devised for tackling quantification problems, exploiting their peculiarities. In particular, two of them are especially designed to work in combination with some state-of-the-art quantifiers, namely AC, PAC and HDy.

The experiments reported show that using these criteria with a simple selection scheme based just on ranking improves the performance of the ensembles when all the models are averaged. Our results are in line with those reported in [15] in the sense that most of the times ensemble selection performs better, but not always. Interestingly, these new criteria outperform in some cases the well-know competence measure based on accuracy. Another important conclusion is that the performance strongly depends on an appropriate combination between the selection criterion used and the base quantifier. In this sense, dynamic ensemble selection using both the HDy algorithm and the HDy selection function seems the overall best method. The fact that quantification learning needs to make estimations for sets of examples instead for individual instances, requires to extend the work that has been done in the field of ensemble selection for other learning tasks. Ac-

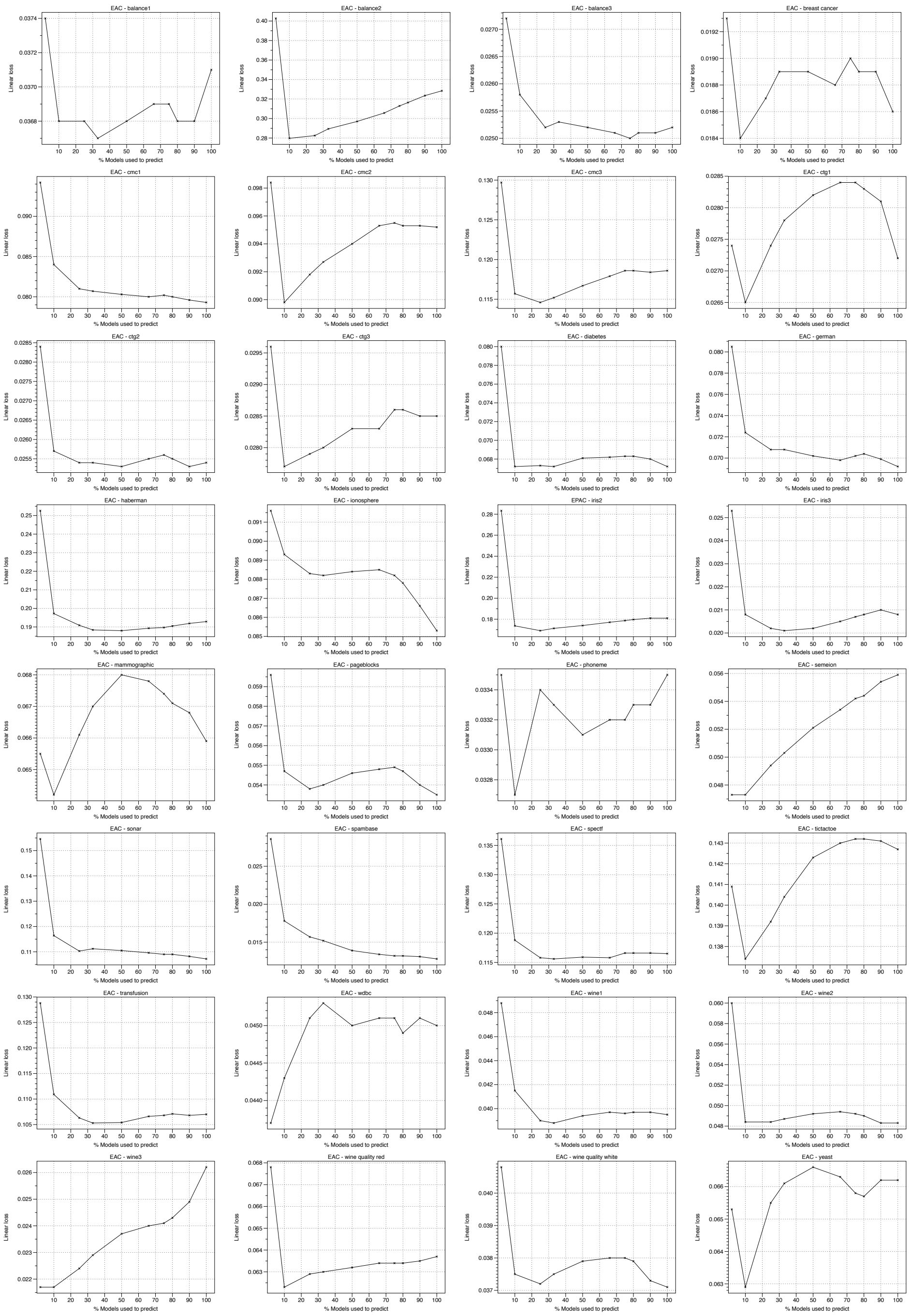

Figure 3: MAE scores selecting a different percentage of models using EAC-DS

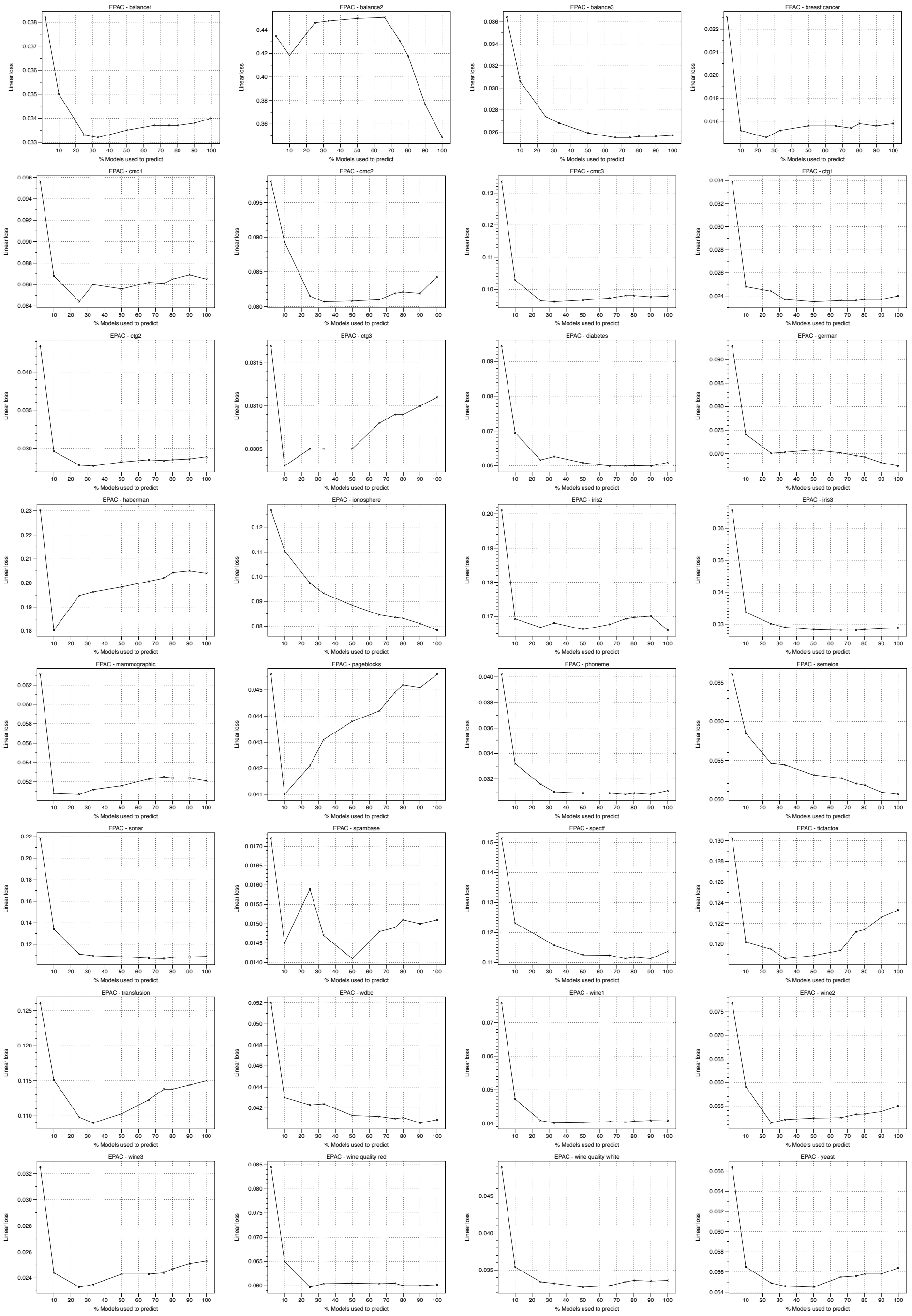

Figure 4: MAE scores selecting a different percentage of models using EPAC-MAX

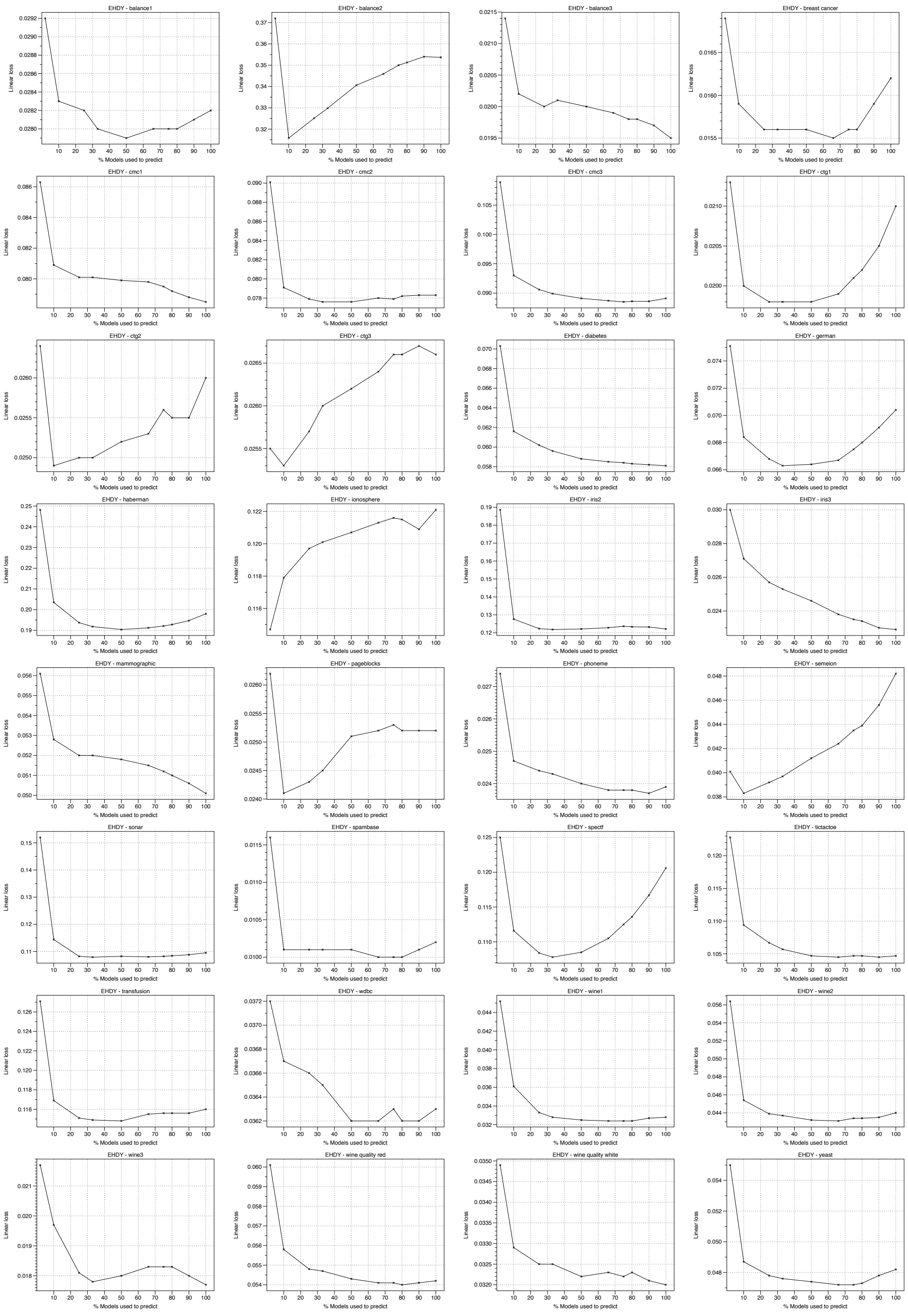

Figure 5: MAE scores selecting a different percentage of models using EHDy-DS

cording to our results, the most promising approach is the one that has into account the similarities between the training and testing distribution.

Finally, several ideas can be considered as future work. The first one is to take into account the interactions between the selected models, trying to employ complementary models in order to boost the final performance. Another interesting idea is to combine somehow static and dynamic criteria, for instance, dynamically selecting from the best models according to a static criterion.

Acknowledgments

This research has been funded by MINECO (the Spanish Ministerio de Economía y Competitividad) and FEDER (Fondo Europeo de Desarrollo Regional), grant TIN2015-65069-C2-2-R (MINECO/FEDER).

References

References

[1] G. Forman, Counting positives accurately despite inaccurate classification, in: Machine Learning: ECML 2005, Springer, 564–575, 2005.

[2] W. Gao, F. Sebastiani, Tweet Sentiment: From Classification to Quantification, in: International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2015), 2015.

[3] P. González, J. Díez, N. Chawla, J. J. del Coz, Why is quantification an interesting learning problem?, Progress in Artificial Intelligence (2016) 1–6.

[4] R. Alaiz-Rodríguez, E. Alegre-Gutiérrez, V. González-Castro, L. Sánchez, Quantifying the proportion of damaged sperm cells based on image analysis and neural networks, in: Proceedings of SMO'08, World Scientific and Engineering Academy and Society (WSEAS), WSEAS Press, 383–388, 2008.

[5] G. Forman, E. Kirshenbaum, J. Suermondt, Pragmatic text mining: minimizing human effort to quantify many issues in call logs, in: Proceedings of ACM SIGKDD'06, ACM, 852–861, 2006.

[6] G. King, Y. Lu, Verbal autopsy methods with multiple causes of death, Statistical Science 23 (1) (2008) 78–91.

[7] P. González, E. Álvarez, J. Barranquero, J. Díez, R. González-Quirós, E. Nogueira, A. López-Urrutia, J. J. del Coz, Multiclass Support Vector Machines With Example-Dependent Costs Applied to Plankton Biomass Estimation, IEEE Transactions on Neural Networks and Learning Systems 24 (11) (2013) 1901–1905.

[8] A. Solow, C. Davis, Q. Hu, Estimating the taxonomic composition of a sample when individuals are classified with error, Mar. Ecol.: Prog. Ser. 216 (2001) 309–311.

[9] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, Q. Zou, LibD3C: ensemble classifiers with a clustering and dynamic selection strategy, Neurocomputing 123 (2014) 424–435.

[10] R. E. Banfield, L. O. Hall, K. W. Bowyer, W. P. Kegelmeyer, Ensemble diversity measures and their application to thinning, Information Fusion 6 (1) (2005) 49–62.

[11] G. Tsoumakas, I. Partalas, I. Vlahavas, An ensemble pruning primer, in: Applications of supervised and unsupervised ensemble methods, Springer, 1–13, 2009.

[12] F. Markatopoulou, G. Tsoumakas, I. Vlahavas, Dynamic ensemble pruning based on multi-label classification, Neurocomputing 150 (2015) 501–512.

[13] R. Caruana, A. Niculescu-Mizil, G. Crew, A. Ksikes, Ensemble selection from libraries of models, in: Proceedings of the twenty-first international conference on Machine learning, ACM, 18, 2004.

[14] G. Tsoumakas, I. Partalas, I. Vlahavas, A taxonomy and short review of ensemble selection, in: Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications, 2008.

[15] A. S. Britto, R. Sabourin, L. E. Oliveira, Dynamic selection of classifiers—A comprehensive review, Pattern Recognition 47 (11) (2014) 3665–3680.

[16] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artificial intelligence 137 (1) (2002) 239–263.

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

[18] A. H. Ko, R. Sabourin, A. S. Britto Jr, From dynamic classifier selection to dynamic ensemble selection, Pattern Recognition 41 (5) (2008) 1718–1731.

[19] P. Pérez-Gállego, J. R. Quevedo, J. J. del Coz, Using ensembles for problems with characterizable changes in data distribution: A case study on quantification, Information Fusion 34 (2017) 87–100.

[20] T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, Machine Learning 40 (2) (2000) 139–157.

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

[22] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, Information Fusion 6 (1) (2005) 5–20.

[23] P. González, E. Álvarez, J. Díez, Á. López-Urrutia, J. J. del Coz, Validation methods for plankton image classification systems, Limnology and Oceanography: Methods (2016) 1–15.

[24] P. González, A. Castaño, C. Nitesh, J. J. del Coz, A Review on Quantification Learning, Tech. Rep., Artificial Intelligence Center, Gijón, Spain., 2017.

[25] G. Forman, Quantifying counts and costs via classification, Data Mining and Knowledge Discovery 17 (2) (2008) 164–206.

[26] A. Bella, C. Ferri, J. Hernández-Orallo, M. J. Ramirez-Quintana, Quantification via probability estimators, in: IEEE International Conference on Data Mining (ICDM'10), 737–742, 2010.

[27] V. González-Castro, R. Alaiz-Rodríguez, E. Alegre, Class Distribution Estimation based on the Hellinger Distance, Information Sciences 218 (2013) 146–164.

[28] A.-R. Ko, R. Sabourin, A. de Souza Britto, Combining diversity and classification accuracy for ensemble selection in random subspaces, in: The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE, 2144–2151, 2006.

[29] C. Tamon, J. Xiang, On the boosting pruning problem, in: European Conference on Machine Learning, Springer, 404–412, 2000.

[30] H. Zhang, L. Cao, A spectral clustering based ensemble pruning approach, Neurocomputing 139 (2014) 289–297.

[31] I. Partalas, G. Tsoumakas, I. P. Vlahavas, Focused Ensemble Selection: A Diversity-Based Method for Greedy Ensemble Selection., in: ECAI, 117–121, 2008.

[32] B. Krawczyk, M. Woźniak, Untrained weighted classifier combination with embedded ensemble pruning, Neurocomputing 196 (2016) 14–22.

[33] I. Partalas, G. Tsoumakas, I. Vlahavas, Pruning an ensemble of classifiers via reinforcement learning, Neurocomputing 72 (7) (2009) 1900–1909.

[34] D. Ruta, B. Gabrys, Classifier selection for majority voting, Information fusion 6 (1) (2005) 63–81.

[35] Y. Zhang, S. Burer, W. N. Street, Ensemble pruning via semi-definite programming, Journal of Machine Learning Research 7 (Jul) (2006) 1315–1338.

[36] B. W. Silverman, Density estimation for statistics and data analysis, vol. 26, CRC press, 1986.

[37] D. W. Scott, Multivariate density estimation: theory, practice, and visualization, John Wiley & Sons, 2015.

[38] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9 (2008) 1871–1874.

[39] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.

[40] A. Benavoli, G. Corani, F. Mangili, Should We Really Use Post-Hoc Tests Based on Mean-Ranks?, Journal of Machine Learning Research 17 (5) (2016) 1–10.

Table 2: Mean absolute errors using different selection functions for the ensemble version of the Classify & Count (CC) quantifier. The best score for each dataset is in bold

ECC - Ensembles of CC

dataset	ALL	ACC	MAX	P$_{tr}$	DS
balance.1	0.0404	0.0417	**0.0396**	0.0397	**0.0396**
balance.2	0.3024	0.3116	0.3032	0.3042	**0.2875**
balance.3	0.0365	0.0373	0.0367	**0.0359**	0.0360
breast-cancer	0.0783	0.0757	0.0753	0.0624	**0.0619**
cmc.1	0.1809	0.1801	0.1808	0.1794	**0.1778**
cmc.2	0.1995	0.1982	0.1987	0.1981	**0.1978**
cmc.3	0.0624	**0.0596**	0.0620	0.0609	0.0605
ctg.1	0.0796	**0.0749**	0.0778	0.0803	0.0792
ctg.2	0.0379	0.0378	0.0377	0.0362	**0.0361**
ctg.3	0.1514	**0.1492**	0.1506	0.1502	0.1508
diabetes	0.1488	0.1470	0.1487	0.1429	**0.1418**
german	0.0983	**0.0948**	0.0962	0.0962	0.0956
haberman	0.2064	**0.2030**	0.2042	0.2090	0.2085
ionosphere	0.1307	**0.1145**	0.1247	0.1283	0.1269
iris.2	0.1766	**0.1671**	0.1731	0.1779	0.1701
iris.3	0.0232	0.0233	0.0253	0.0218	**0.0215**
mammographic	0.0237	0.0228	0.0229	0.0218	**0.0217**
pageblocks.5	0.1391	**0.1218**	0.1258	0.1451	0.1441
phoneme	0.0596	**0.0478**	0.0518	0.0643	0.0615
semeion.8	0.1337	0.1318	0.1332	0.1321	**0.1315**
sonar	0.1406	0.1327	0.1398	**0.1308**	0.1356
spambase	0.1795	**0.1778**	0.1790	0.1781	0.1782
spectf	0.1598	0.1580	0.1587	0.1498	**0.1444**
tictactoe	0.1865	0.1857	0.1862	0.1844	**0.1827**
transfusion	0.1703	**0.1701**	0.1704	**0.1701**	0.1702
wdbc	0.0625	**0.0602**	0.0633	0.0629	0.0631
wine.1	0.0443	**0.0398**	0.0461	0.0449	0.0442
wine.2	0.0538	**0.0496**	0.0549	0.0540	0.0526
wine.3	0.0210	0.0210	0.0223	0.0201	**0.0197**
wine-quality-red	0.1333	0.1326	0.1333	0.1315	**0.1312**
wine-quality-white	0.1462	**0.1435**	0.1454	0.1491	0.1473
yeast	0.1381	**0.1343**	0.1365	0.1362	0.1358
Avg. rank	4.1250	2.2500	3.5781	2.9375	**2.1094**

Table 3: MAE results using different selection functions for the ensemble version of the Probabilistic Classify & Count (PCC). The best score for each dataset is in bold

EPCC - Ensembles of PCC

dataset	ALL	ACC	MAX	P_{tr}	DS
balance.1	0.0861	**0.0750**	0.0755	0.0836	0.0847
balance.2	0.2609	0.2639	0.2631	0.2612	**0.2587**
balance.3	0.0575	**0.0531**	0.0539	0.0553	0.0554
breast-cancer	0.1416	0.1278	0.1278	0.1292	**0.1272**
cmc.1	0.2184	**0.2155**	0.2157	0.2175	0.2158
cmc.2	0.2302	**0.2286**	0.2287	0.2299	0.2293
cmc.3	0.0809	**0.0767**	0.0782	0.0789	0.0785
ctg.1	0.1044	**0.0994**	0.1003	0.1028	0.1020
ctg.2	0.0485	0.0473	0.0477	**0.0462**	**0.0462**
ctg.3	0.2076	**0.2014**	0.2016	0.2054	0.2072
diabetes	0.1932	0.1897	0.1903	0.1907	**0.1885**
german	0.1334	**0.1280**	0.1291	0.1322	0.1319
haberman	0.2276	**0.2225**	0.2226	0.2265	0.2287
ionosphere	0.1283	**0.1169**	0.1285	0.1209	0.1221
iris.2	0.2085	**0.2008**	0.2017	0.2056	0.2093
iris.3	0.0427	**0.0378**	0.0399	0.0393	0.0391
mammographic	0.0296	**0.0271**	0.0274	0.0276	0.0276
pageblocks.5	0.1166	**0.1066**	0.1089	0.1182	0.1168
phoneme	0.1192	**0.0984**	0.1042	0.1248	0.1199
semeion.8	0.1866	**0.1813**	0.1814	0.1841	0.1859
sonar	0.1893	0.1805	0.1829	**0.1772**	0.1835
spambase	0.2148	**0.2128**	**0.2128**	0.2145	0.2134
spectf	0.1913	0.1865	0.1871	0.1825	**0.1817**
tictactoe	0.2159	0.2138	0.2139	0.2154	**0.2133**
transfusion	0.2136	**0.2097**	0.2105	0.2142	0.2131
wdbc	0.0776	**0.0723**	0.0734	0.0753	0.0751
wine.1	0.0610	**0.0548**	0.0587	0.0588	0.0587
wine.2	0.0788	**0.0693**	0.0730	0.0768	0.0757
wine.3	0.0331	**0.0277**	0.0314	0.0312	0.0299
wine-quality-red	0.1756	**0.1725**	0.1731	0.1753	0.1738
wine-quality-white	0.1901	**0.1862**	0.1870	0.1922	0.1902
yeast	0.1814	**0.1732**	0.1761	0.1836	0.1801
Avg. rank	4.5625	**1.4062**	2.5156	3.5625	2.9531

Table 4: Mean absolute errors using different selection functions for the ensemble version of the Adjusted Count (AC) quantifier. The best score for each dataset is in bold

EAC - Ensembles of AC

dataset	ALL	ACC	MAX	P_{tr}	DS
balance.1	0.0371	0.0374	**0.0349**	0.0367	0.0368
balance.2	0.3284	0.3926	0.4067	0.3163	**0.2970**
balance.3	**0.0252**	0.0258	0.0257	**0.0252**	**0.0252**
breast-cancer	0.0559	0.0570	0.0598	0.0524	**0.0521**
cmc.1	0.0952	0.0947	0.0970	0.0950	**0.0940**
cmc.2	0.1186	0.1180	0.1122	0.1176	**0.1167**
cmc.3	0.0272	0.0271	**0.0268**	0.0283	0.0282
ctg.1	0.0254	0.0257	0.0254	0.0254	**0.0253**
ctg.2	0.0285	0.0289	**0.0283**	0.0284	**0.0283**
ctg.3	0.0662	**0.0658**	0.0663	0.0678	0.0666
diabetes	**0.0692**	0.0721	**0.0692**	0.0701	0.0702
german	0.0659	0.0677	**0.0630**	0.0683	0.0680
haberman	0.1929	0.1941	0.1892	0.1910	**0.1880**
ionosphere	**0.0853**	0.0915	0.0947	0.0891	0.0884
iris.2	0.1810	0.1857	**0.1737**	0.1782	0.1740
iris.3	0.0208	0.0213	0.0229	0.0208	**0.0202**
mammographic	**0.0186**	0.0194	0.0188	0.0192	0.0189
pageblocks.5	0.0535	0.0561	**0.0527**	0.0544	0.0546
phoneme	**0.0128**	0.0131	0.0130	0.0143	0.0139
semeion.8	0.0335	0.0341	0.0336	**0.0331**	**0.0331**
sonar	**0.1072**	0.1089	0.1092	0.1134	0.1105
spambase	0.0793	0.0798	**0.0791**	0.0799	0.0803
spectf	0.1165	**0.1109**	0.1193	0.1213	0.1159
tictactoe	0.1427	0.1426	**0.1376**	0.1423	0.1423
transfusion	0.1070	0.1082	0.1106	0.1087	**0.1054**
wdbc	0.0450	0.0455	**0.0447**	0.0452	0.0450
wine.1	0.0395	**0.0389**	0.0398	0.0402	0.0394
wine.2	0.0483	**0.0479**	0.0491	0.0508	0.0492
wine.3	0.0262	0.0261	0.0238	0.0239	**0.0237**
wine-quality-red	0.0637	0.0639	0.0647	0.0633	**0.0632**
wine-quality-white	**0.0371**	**0.0371**	0.0378	0.0377	0.0379
yeast	**0.0672**	0.0697	0.0699	0.0710	0.0681
Avg. rank	2.6875	3.5469	2.8438	3.4531	**2.4688**

Table 5: MAE results using different selection functions for the ensemble version of the Probabilistic Adjusted Count (PAC) quantifier. The best score for each dataset is in bold

	EPAC - Ensembles of PAC				
dataset	ALL	ACC	MAX	P_{tr}	DS
balance.1	0.0340	0.0338	0.0335	**0.0332**	0.0333
balance.2	0.3486	0.4748	0.4497	0.3253	**0.3039**
balance.3	0.0257	0.0262	0.0259	0.0257	**0.0256**
breast-cancer	0.0506	0.0511	0.0531	**0.0475**	0.0476
cmc.1	0.0843	**0.0800**	0.0808	0.0832	0.0840
cmc.2	0.0979	0.0980	**0.0967**	0.0981	0.0980
cmc.3	0.0240	0.0237	**0.0235**	0.0242	0.0241
ctg.1	0.0289	0.0294	**0.0282**	0.0285	0.0284
ctg.2	0.0311	0.0310	**0.0305**	0.0308	0.0307
ctg.3	0.0564	0.0559	**0.0545**	0.0553	0.0554
diabetes	**0.0674**	0.0680	0.0708	0.0681	0.0684
german	0.0521	0.0536	**0.0516**	0.0534	0.0534
haberman	0.2040	0.2096	**0.1984**	0.2023	0.2017
ionosphere	**0.0784**	0.0809	0.0884	0.0829	0.0825
iris.2	0.1660	0.1689	0.1662	0.1662	**0.1652**
iris.3	0.0288	0.0277	0.0283	0.0276	**0.0269**
mammographic	0.0179	0.0181	**0.0178**	0.0182	0.0181
pageblocks.5	0.0456	0.0459	**0.0438**	0.0473	0.0469
phoneme	0.0151	**0.0130**	0.0141	0.0188	0.0163
semeion.8	0.0311	0.0316	**0.0309**	0.0311	0.0311
sonar	0.1088	0.1092	**0.1085**	0.1097	0.1123
spambase	0.0865	0.0888	**0.0856**	0.0861	0.0864
spectf	0.1137	0.1129	**0.1125**	0.1142	0.1145
tictactoe	0.1233	0.1248	**0.1189**	0.1220	0.1221
transfusion	0.1150	0.1181	**0.1103**	0.1153	0.1166
wdbc	0.0409	**0.0401**	0.0413	0.0411	0.0410
wine.1	0.0407	0.0419	**0.0402**	0.0406	0.0405
wine.2	0.0550	0.0549	**0.0524**	0.0575	0.0560
wine.3	0.0253	0.0244	0.0243	0.0236	**0.0234**
wine-quality-red	0.0602	0.0597	0.0605	**0.0590**	0.0591
wine-quality-white	0.0336	0.0335	**0.0327**	0.0342	0.0343
yeast	0.0609	0.0616	**0.0608**	0.0631	0.0623
Avg. rank	3.1406	3.5000	**2.1719**	3.2344	2.9531

Table 6: Mean absolute errors using different selection functions for the ensemble version of the HDy quantifier. The best score for each dataset is in bold

EHDy - Ensembles of HDy

ddataset	ALL	ACC	MAX	P$_{tr}$	DS
balance.1	0.0282	**0.0277**	0.0288	0.0281	0.0279
balance.2	0.3537	0.3573	0.3469	**0.3312**	0.3406
balance.3	**0.0195**	0.0197	0.0205	0.0201	0.0200
breast-cancer	0.0482	0.0478	0.0571	0.0422	**0.0412**
cmc.1	0.0783	0.0764	0.0813	**0.0776**	**0.0776**
cmc.2	0.0891	**0.0889**	0.0906	0.0894	0.0891
cmc.3	0.0210	**0.0197**	0.0223	0.0198	0.0198
ctg.1	0.0260	0.0260	0.0261	**0.0252**	**0.0252**
ctg.2	0.0266	0.0267	0.0268	0.0264	**0.0262**
ctg.3	0.0482	0.0483	0.0483	0.0476	**0.0474**
diabetes	0.0704	0.0673	0.0719	0.0657	**0.0664**
german	**0.0501**	0.0507	0.0506	0.0518	0.0518
haberman	0.1980	0.2107	0.1950	0.1926	**0.1904**
ionosphere	0.1221	**0.0992**	0.1151	0.1234	0.1207
iris.2	**0.1221**	0.1288	0.1267	0.1252	**0.1221**
iris.3	0.0229	0.0236	**0.0227**	0.0253	0.0246
mammographic	0.0162	0.0152	**0.0149**	0.0157	0.0156
pageblocks.5	0.0252	0.0255	0.0252	0.0256	**0.0251**
phoneme	0.0102	0.0101	0.0103	**0.0100**	0.0101
semeion.8	**0.0239**	0.0240	0.0241	0.0240	0.0240
sonar	0.1095	0.1088	0.1164	0.1105	**0.1082**
spambase	0.0785	0.0796	**0.0772**	0.0802	0.0799
spectf	0.1206	0.1207	0.1208	0.1193	**0.1085**
tictactoe	**0.1047**	0.1063	0.1012	0.1049	**0.1047**
transfusion	0.1160	0.1215	0.1101	0.1180	**0.1148**
wdbc	0.0363	**0.0351**	0.0387	0.0364	0.0362
wine.1	0.0328	0.0334	0.0334	**0.0322**	0.0325
wine.2	0.0440	0.0429	**0.0425**	0.0443	0.0432
wine.3	0.0177	**0.0168**	0.0181	0.0191	0.0180
wine-quality-red	0.0542	**0.0543**	0.0552	**0.0543**	**0.0543**
wine-quality-white	**0.0320**	0.0321	0.0325	0.0322	0.0322
yeast	0.0581	**0.0559**	0.0592	0.0582	0.0588
Avg. rank	2.8594	2.8438	3.7656	3.1719	**2.3594**

Table 7: Average ranking for all ensemble methods using MAE (top) and MSE (bottom) as performance measures. Symbol † indicates that the selection function in the corresponding column is significantly better than ALL using a Bergmann-Hommel test ($p < 0.05$). Symbol § indicates the opposite

MAE average rankings

Algorithm	ALL	ACC	MAX	P_{tr}	DS
ECC	4.1250	2.2500 †	3.5781	2.9375†	**2.1094**†
EPCC	4.5625	**1.4062**†	2.5156 †	3.5625†	2.9531 †
EAC	2.6875	3.5469	2.8438	3.4531	**2.4688**
EPAC	3.1406	3.5000	**2.1719**	3.2344	2.9531
EHDy	2.8594	2.8438	3.7656	3.1719	**2.3594**

MSE average rankings

Algorithm	ALL	ACC	MAX	P_{tr}	DS
ECC	4.0156	**2.1250**†	3.5938 †	2.9531	2.3125 †
EPCC	4.6094	**1.4219**†	2.5156 †	3.4844†	2.9688 †
EAC	**2.4531**	3.6250 §	3.1250	3.1875	2.6094
EPAC	2.9219	3.4219	**2.5938**	2.9219	3.1406
EHDy	2.9688	3.1250	3.4688	2.8750	**2.5625**

Table 8: p-values for the comparison between ALL and the proposed selection functions using the Wilcoxon signed-rank test

ECC	ACC	MAX	P_{tr}	DS
MAE	0.000069	0.012080	0.034120	0.000381
MSE	0.000044	0.006810	0.079819	0.000633

EPCC	ACC	MAX	P_{tr}	DS
MAE	0.000002	0.000002	0.000641	0.000008
MSE	0.000005	0.000008	0.002938	0.000031

EAC	ACC	MAX	P_{tr}	DS
MAE	0.004080	0.537180	0.180012	0.658280
MSE	0.004387	0.316767	0.626407	0.834729

EPAC	ACC	MAX	P_{tr}	DS
MAE	0.099776	0.016635	0.975385	0.945307
MSE	0.266549	0.015474	0.903040	0.637021

EHDy	ACC	MAX	P_{tr}	DS
MAE	0.929713	0.263913	0.940368	0.026648
MSE	0.577523	0.745263	0.483749	0.084681