Quantification-oriented learning based on reliable classifiers

Jose Barranquero, Jorge Díez, Juan José del Coz*

Artificial Intelligence Center (University of Oviedo), Campus de Viesques s/n, 33204, Spain

Abstract

Real-world applications demand effective methods to estimate the class distribution of a sample. In many domains, this is more productive than seeking individual predictions. At a first glance, the straightforward conclusion could be that this task, recently identified as quantification, is as simple as counting the predictions of a classifier. However, due to natural distribution changes occurring in real-world problems, this solution is unsatisfactory. Moreover, current quantification models based on classifiers present the drawback of being trained with loss functions aimed at classification rather than quantification. Other recent attempts to address this issue suffer certain limitations regarding reliability, measured in terms of classification abilities. This paper presents a learning method that optimizes an alternative metric that combines simultaneously quantification and classification performance. Our proposal offers a new framework that allows the construction of binary quantifiers that are able to accurately estimate the proportion of positives, based on models with reliable classification abilities.

Keywords: Quantification, Class distribution estimation, Performance metrics, Reliability, Multivariate predictions

1. Introduction

Any data scientist who had tackled real-world problems knows that there exist classification domains that are inherently complex, it being very difficult to obtain accurate predictions when focusing on each specific example; i.e., to achieve high classification accuracy. However, it is not so strange to require estimations about the characteristics of the overall sample instead,

[*] Corresponding author. J.J. del Coz (juanjo@aic.uniovi.es). Phone/Fax: +34 985182501/985182125

Email addresses: barranquero@aic.uniovi.es (Jose Barranquero), jdiez@aic.uniovi.es (Jorge Díez), juanjo@aic.uniovi.es (Juan José del Coz)

mainly with respect to data distribution. Tentative application scopes include opinion mining [1], network-behavior analysis [2], remote sensing [3], quality control [4], word-sense disambiguation [5], monitoring of support-call logs [6], credit scoring [7] and adaptive fraud-detection [8], among others.

For instance, in order to measure the success of a new product, there is an increasing demand for methods for tracking overall consumer opinion, superseding classical approaches aimed at individual perceptions. To answer questions like *how many clients are satisfied with our new product?*, we need effective algorithms focused on estimating the distribution of classes from a sample. This has emerging relevance when dealing with the tracking of trends over time [9], such as early detection of epidemics and endangered species, risk prevalence, market and ecosystem evolution, or any other kind of distribution change in general.

In many business, scientific and medical applications, it is sufficient, and sometimes even more relevant, to obtain estimations at an aggregated level in order to properly plan strategies. Companies could obtain greater returns on investment if they are able to accurately estimate the proportion of *events* that will involve higher costs or benefits. This will avoid wasting resources in guessing the class of each specific event; a task that usually reveals itself as complex, expensive and error-prone. For example, the estimation of the proportion of policy holders that will be involved in accidents during the next year, or the estimation of overall consumer satisfaction with respect to any specific product, service or brand.

In machine learning, the task of quantification is *to accurately estimate the number of positive cases (or class distribution) in a test set, using a training set that may have a substantially different distribution* [10]. Despite having many potential applications, this problem has barely been addressed within the community, and has yet to be properly standardized in terms of error measurement, experimental setup and methodology in general. Unfortunately, quantification has attracted little attention due to the mistaken belief of it being somewhat trivial. The key problem is that it is not as simple as classifying and counting the examples of each class, seeing as different distributions of train and test data can have a huge impact on the performance of state-of-the-art classifiers. The general assumption made by classification methods is that the samples are representative [11], which implies that the within-class probability densities, $Pr(\boldsymbol{x}|y)$, and the a priori class distribution, $Pr(y)$, do not vary.

The influence of different changing environments on classification and the performance of knowledge-based systems has been analyzed in several studies (see, for instance, [7, 12, 13]), suggesting that addressing distribution drifts is a complex and critical problem. Moreover, many papers focus on addressing distribution changes for classification, offering different views of what is subject to change and what is assumed to be constant. As in previous quantification-related papers, we focus only on studying changes in the a priori class distribution, while maintaining within-class probability densities constant. Domains of this kind are identified as $Y \rightarrow X$ problems by Fawcett and Flach [14]. Provided that we use stratified sampling [15], an example of situations where $Pr(\boldsymbol{x}|y)$ does not change is when the number of examples of one or both classes is conditioned by the costs associated with obtaining and labeling them [16]. The explicit study of other types of distribution shifts, as well as $X \rightarrow Y$ domains, fall outside the scope of this paper (for further reading, we refer the reader to [17, 18, 19, 20]).

Receiver Operating Characteristic (ROC) analysis is quite a popular technique for the graphical analysis of classification models [21]. A classifier may be trained for one particular operating condition, defined by one class distribution and cost proportion, but might then be deployed on a different condition. ROC curves visualize how the true positive rate (TPR) and the false positive rate (FPR) evolve for the same classifier for a range of thresholds. The threshold is the element to adapt a classifier to a given operating condition. ROC-based methods [8, 22] and cost curves [23] have been successfully applied to adjust the classification threshold, given that new class priors are known in advance. However, as already stated by Forman [10], these approaches are not useful for estimating class distributions from test sets. Similarly, if these new priors are unknown, two main approaches have been followed in the literature. On the one hand, most published papers focus on adapting the deployed models to the new conditions [24, 25, 26, 27, 28]. On the other hand, the alternative view is mainly concerned with enhancing *robustness* in order to learn models that are more resilient to changes in class distribution [29]. Whatever the case may be, the aim of these methods, although related, is quite different from that of quantification, as adapting a classifier for improving individual classification performance does not imply obtaining better quantification predictions, as we shall discuss later. Moreover, there exists a natural connection with imbalance-tolerant methods, mainly those based on preprocessing of data [30]. Actually, quantification was originally designed to deal with highly imbalanced datasets [10]; however, these preprocessing techniques are not directly applicable in changing environments.

The main approach that has been studied in the literature for learning an explicit binary-quantification model is based on standard classifiers, following a two-step training procedure. The first step is to train a classifier optimizing a classification metric, usually accuracy. The next step is then to study some relevant properties of this classifier. The aim of this second step is to correct the quantification prediction obtained from aggregating classifier estimates [10, 31].

An open question is whether it may be more effective to learn a classifier optimizing a quantification metric, instead of a classification performance measure. Conceptually, this alternative strategy is more formal, because the learning process takes into account the target performance measure. The main contribution of this paper is to explore this approach in detail.

The idea of optimizing a pure quantification metric during learning was introduced by Esuli and Sebastiani [1], although these authors neither implement nor evaluate it. Their proposal is based on learning a binary classifier with optimum quantification performance. We argue that this method has a pitfall. The key problem that arises when optimizing a pure quantification measure is that the resulting hypothesis space contains several global optimums. In practice, however, these optimum hypotheses are not equally good due to the fact that they differ in terms of the quality of their future quantification predictions. This paper claims that the robustness of a quantifier based on an underlying classifier is directly related to the reliability of such classifier. For instance, given several models showing equivalent quantification performance during training, the learning method should prefer the best one in terms of its potential for generalization. As we shall analyze later, this factor is closely related with their classification abilities.

This lead us to further explore Esuli and Sebastiani's approach with the aim of building a learning method able to induce more robust quantifiers based on classifiers that are as reliable as possible. In order to accomplish this goal, we introduce a new metric that combines both factors. That is, a metric that combines classification performance with quantification performance, resulting in better quantification models.

As occurs with any other quantification metric, our proposal measures performance from an aggregated perspective, taking into account the whole sample. The difficulty involved in optimizing such functions is that they are not decomposable as a linear combination of the individual errors. Hence, not all binary learners are capable of optimizing them directly, requiring a more advanced learning machine. In this paper we adapt Joachim's multivariate SVMs [32] to implement our proposal and the idea presented by Esuli and Sebastiani. In order to validate these two approaches, another key contribution is to perform an exhaustive study in which we compare them, along with several state-of-the-art quantifiers, by means of benchmark datasets from the UCI Machine Learning repository [33].

The paper is organized as follows. Section 2 introduces binary quantification as a learning task. Core concepts, notation and performance metrics for binary quantification are presented first. Then, a brief review of available quantification methods is provided, including those approaches based on adjusted classification (Section 2.2.2) and threshold selection policies (Section 2.2.3). Quantification-oriented learning is analyzed in depth in Section 3. First, we describe the idea proposed by Esuli and Sebastiani. Then, we discuss a possible pitfall in their approach. Finally, we introduce our method (Section 3.3), based on a new quantification measure called $Q\text{-}measure$. For a better understanding of our proposal, we describe $Q\text{-}measure$, both conceptually and graphically, in comparison with other performance measures. Section 4 reports the experiments performed, including the experimental setup, datasets, algorithms and statistical tests employed. The results are discussed in terms of different quantification measures. The paper ends by drawing some conclusions in Section 5.

2. Binary quantification

From a statistical point of view, the aim of a binary quantification task is to estimate the prevalence of an event or property within a sample. During the learning stage, we have a training set with examples labeled as positives or negatives; formally, $D = \{(\boldsymbol{x}_i, y_i) : i = 1 \dots S\}$, in which $\boldsymbol{x}_i$ is an object of the input space $\mathcal{X}$ and $y_i \in \mathcal{Y} = \{-1, +1\}$. This dataset shows a specific distribution that can be summarized with the actual proportion of positives or prevalence. The learning goal is to obtain a model able to predict the prevalence (p) of another sample, usually identified as the test set, that may show a markedly different distribution of classes. Thus, the input data is equivalent to that of traditional classification problems, but the focus is on the estimated prevalence (p') of the sample, rather than on the class assigned to each individual example. Notice that we use p and p' to identify the actual and estimated prevalences of any sample; these variables are not tied to training or test sets in any way.

Table 1 summarizes the notation that we shall employ throughout the paper. First, an algorithm is applied over the training set in order to learn a classifier. Then, we take the test set, where P represents the count of actual positives and N the count of actual negatives. Once the classifier is applied over this second set to predict its classes, we have that P' is the count of

Table 1: Contingency table for binary problems

	P	N
P'	TP	FP
N'	FN	TN
$(S = P + N = P' + N')$		

individuals predicted as positives, N' the count of predicted negatives, while TP, FN, TN and FP represent the count of *true positives*, *false negatives*, *true negatives* and *false positives*. We can then obtain the actual and estimated prevalences as $p = P/S$ and $p' = P'/S$, respectively. Notice again that these values can be computed for any set of examples, provided we use the classifier to predict their classes, even for the training set itself.

2.1. Performance measures for binary quantification

This section presents a brief review of several quantification loss functions that have been applied in previous quantification papers.

2.1.1. Estimation bias

According to Forman [10], the estimation bias is a natural error metric for quantification, which is computed as the estimated percentage of positives minus the actual percentage of positives

$$bias = p' - p = \frac{P' - P}{S} = \frac{FP - FN}{S}. \tag{1}$$

When a method outputs more FP than FN, it shows a positive bias, and vice-versa. Thus, this metric measures whether the model tends to overestimate or underestimate the proportion of the positive class. However, this metric is not useful for evaluating the overall performance in terms of average error (for a collection of sets), for the reason that negative and positive biases are neutralized. That is, as Forman points out, *a method that guesses 5% too high or too low equally will often have zero bias* on average.

2.1.2. Absolute and squared errors

Forman proposed [10, 34, 35] the *Absolute Error* (AE) between actual and predicted positive prevalence as a standard loss function for quantification, that is simple, interpretable and directly applicable:

$$AE = |p' - p| = \frac{|P' - P|}{S} = \frac{|FP - FN|}{S}. \tag{2}$$

As an alternative to AE, Bella et al. [31] proposed the *Squared Error* (SE):

$$SE = (p' - p)^2 = \left(\frac{P' - P}{S}\right)^2 = \left(\frac{FP - FN}{S}\right)^2. \tag{3}$$

Actually, *Mean Absolute Error* (MAE) and *Mean Squared Error* (MSE) are probably the most commonly used loss functions in regression problems. The concept of computing the absolute or squared error of real value estimations can be extended to any problem based on a continuos variable, like p. However, in the case of quantification, averaging among samples with different actual prevalence or from different domains has some implications that should be carefully taken into account [10]. Note, for instance, that having a 5% AE for a test set with 45% of positive examples may not be equivalent to obtaining the same error over a test set with only 10% of positive examples.

2.1.3. Kullback-Leibler Divergence

Kullback-Leibler Divergence (KLD), also known as normalized cross-entropy (see [1, 10]), can be applied in the context of quantification. Assuming that we have only two classes, the final equation is:

$$KLD = \frac{P}{S} \cdot \log\left(\frac{P}{P'}\right) + \frac{N}{S} \cdot \log\left(\frac{N}{N'}\right). \tag{4}$$

This metric determines the error made in estimating the predicted distribution (P'/S, N'/S) with respect to the true distribution (P/S, N/S).

The main advantages of KLD are that it may be more appropriate to average over different test prevalences and more suitable for extending the quantification task for multiclass problems. However, a drawback of KLD is that it is less interpretable than other measures, like AE. Moreover, we also need to define its output for those cases in which P, N, P' or N' are zero (see Section 3.4.2).

2.2. Quantification methods: state-of-the-art

The task of quantification has been formally addressed in a limited number of papers in recent years, with several complementary approaches having been proposed. Here we present a brief review of some of them.

2.2.1. Classify and count

The most simple method for building a quantifier is to learn a classifier, use the resulting model to label the instances of the sample and count the proportions of each class. This method is taken as a baseline by Forman [10], identifying it as *Classify & Count* (CC). Actually, it is straightforward to conclude that a perfect classifier would lead to a perfect quantifier. The key problem is that developing a perfect classifier is unrealistic, getting instead imperfect classifiers in real-world environments. This also implies that the quantifier will inherit the bias of the underlying classifier.

For instance, given a binary classification problem in which the learned classifier tends to misclassify some positive examples, then the derived quantifier will underestimate the proportion of the positive class. This effect becomes even more problematic in a changing environment, in which the test distribution is usually substantially different from that of the training set. Following the previous example, when the proportion of the positive class goes up uniformly in the test set, then the number of misclassified positive instances increases and the quantifier will underestimate the positive class even more. Forman highlighted and studied this behavior for binary quantification, proposing several methods to tackle such bias.

2.2.2. Quantification via adjusted classification

With the aim of correcting classification bias, Forman [34] proposed a method termed *Adjusted Count* (AC), in which the process is to train a classifier and estimate its tpr (true positive rate) and fpr (false positive rate) characteristics:

$$tpr = \frac{TP}{P} \qquad \text{and} \qquad fpr = \frac{FP}{N}\ , \tag{5}$$

through cross-validation over the training set. That is, for each fold we compute TP, FP, P and N to average tpr and fpr across all folds. The next step is then to count the positive predictions of the classifier over the test examples (i.e., just like the CC method) and adjust this value via the following formula

$$p'' = \frac{p' - fpr}{tpr - fpr}\ , \tag{6}$$

where p'' denotes the adjusted proportion of positive test examples and p' is the estimated proportion obtained by counting the classifier outputs over the test set. In some cases, this leads to infeasible estimates of p, requiring a final step in order to clip the estimation into the range $[0, 1]$.

Bearing in mind that the values of tpr and fpr are also estimates, we obtain an approximation, p'', of the actual proportion, p. These two rates are crucial in understanding quantification methods as proposed by Forman because they are designed under the assumption that the a priori class distribution, $Pr(y)$, changes, but the within-class probability densities, $Pr(\boldsymbol{x}|y)$, do not. This in turn ensures that both classifier characteristics, tpr and fpr, are independent of changes in class distribution (see [14]).

Note that due to (5), only the tpr fraction of any shift in P will be perceived by the already-trained classifier ($TP = tpr \cdot P$). Moreover, the fpr fraction of N is misclassified as false positives ($FP = fpr \cdot N$). In line with these observations, Forman [10] states the following theorem and its corresponding proof:

Theorem 2.1 (Forman's Theorem). *For an imperfect classifier, the CC method will underestimate the true proportion of positives p in a test set for $p > p^*$, and overestimate for $p < p^*$, where p^* is the particular proportion at which the CC method estimates correctly; i.e., the CC method estimates exactly p^* for a test set having p^* positives.*

The overall conclusion is that a non-adjusted classifier tends to underestimate the prevalence of the positive class when it increases, and vice-versa.

2.2.3. Quantification via threshold selection policies

Given that the AC method allows any base classifier to be used to build a quantifier, the underlying learning process has attracted little attention. Much of the effort is once again due to Forman, who proposed a collection of methods based on training a linear SVM classifier, employing a posterior calibration of its threshold. The main difference between these methods is the threshold selection policy employed, aimed at alleviating some drawbacks of AC correcting formula from alternative perspectives.

A key problem related to the AC method is that its performance mainly depends on the degree of imbalance in the training set, worsening when the positive class is scarce [35]. In this case, the underlying classifier tends to minimize the false positive errors, which usually implies a low tpr (see [22]) and a small denominator in Equation (6). This fact produces high vulnerability to fluctuations in the estimation of tpr or fpr.

For highly imbalanced situations, the main intuition is that selecting a threshold that allows more true positives, even at the cost of many more false positives, could afford better quantification performance. The goal is to choose those thresholds where the estimates of tpr and fpr have less variance or where the denominator in Equation (6) is large enough to be more resistant to estimation errors. For instance, the *Max* method selects a threshold that maximizes the difference between tpr and fpr, while the *X* method chooses the threshold where fpr equals $1 - tpr$, avoiding the tails of both curves. In line with this last idea and assuming that positives constitute the minority class, the T50 method selects the threshold with $tpr = 50\%$, avoiding only the tails of tpr curve.

Notwithstanding, there is another problem related with all these methods arising from the fact that the estimation of tpr and fpr can differ significantly from the real values. Forman thus proposed a more advanced method, *Median Sweep* (MS), based on estimating the prevalence for all thresholds during testing, in order to compute their median. This strategy is comparatively consistent, smoothing over estimation errors like in bootstrap-based algorithms and showing promising empirical results in practice.

2.2.4. Quantification via probability estimators

Bella et al. [31] have recently developed a family of methods they call *probability estimation & average*. Their core proposal is to develop a probabilistic version of AC. First they introduce a simple method called *Probability Average* (PA), which is clearly aligned with CC. The key difference is that the classifier learned is probabilistic in this case. Once the probability predictions are obtained from the test dataset, the average of these probabilities is computed for the positive class as follows:

$$p' = \hat{\pi}^{PA}_{Test}(\oplus) = \frac{1}{S}\sum_{i=1}^{S} Pr(y_i = +1|\boldsymbol{x}_i). \tag{7}$$

As might be expected, when the proportion of positives changes between training and test, then PA will underestimate or overestimate as occurs with CC. These authors thus propose an enhanced version of this method, called *Scaled Probability Average* (SPA). Similar to CC and AC, the estimation p' obtained from Equation (7) is corrected according to a simple scaling formula:

$$p'' = \hat{\pi}^{SPA}_{Test}(\oplus) = \frac{p' - FP_{pa}}{TP_{pa} - FP_{pa}} \ , \tag{8}$$

where TP_{pa} and FP_{pa} are values estimated from the training set, defined respectively as *TP probability average* or *positive probability average of the positives*

$$TP_{pa} = \hat{\pi}_{Train_{\oplus}}(\oplus) = \frac{\sum_{\{i|y_i=+1\}} Pr(y_i = +1|\boldsymbol{x}_i)}{\#\{y_i = +1\}},$$

and *FP probability average* or *positive probability average of the negatives*

$$FP_{pa} = \hat{\pi}_{Train_{\ominus}}(\oplus) = \frac{\sum_{\{i|y_i=-1\}} Pr(y_i = +1|\boldsymbol{x}_i)}{\#\{y_i = -1\}}.$$

The expression defined in Equation (8) yields a probabilistic version of Forman's adjustment defined in Equation (6). In their experiments, the SPA method outperforms CC, AC and T50; although they do not compare their proposal with other methods based on threshold selection policies like Max, X or MS.

3. Quantification-oriented learning

Esuli and Sebastiani [1] suggest the first training approach explicitly designed to learn a binary quantifier, in the context of a sentiment quantification task. However, a key limitation is that they neither implement nor validate it. This paper presents the first experiment results based on such an approach. Moreover, in this section we point out a possible pitfall in their idea and propose an alternative based on a new quantification measure, called $Q\text{-}measure$.

3.1. Idea proposed by Esuli and Sebastiani

Although the training method that these authors describe is also based on building a classifier, in this case the learning process optimizes the quantification error, without taking into consideration the classification performance of the model. Essentially, as their focus is on binary quantification problems, they argue that compensating the errors between both classes provides the means for obtaining better quantifiers. Therefore, the key idea is to optimize a metric derived from the expression $|FP - FN|$. That is, a perfect quantifier should simply counterbalance all false positives with the same amount of false negative errors. In fact, all loss functions reviewed in Section 2.1 reach their optimum when this difference is equal to 0.

One difficulty in implementing this idea is that not all binary learners are capable of optimizing this kind of metric, because such functions are not decomposable as a linear combination of the individual errors. Hence, this approach requires a more advanced learning machine, like SVM^{Δ}_{multi} [32], which provides an efficient base algorithm for optimizing non-linear functions computed from the contingency table (see Table 1). However, the straightforward benefit is that these methods address the quantification problem from an aggregated perspective, taking into account the performance over whole samples, which seems more appropriate for the problem in general.

Therefore, rather than learning a traditional classification model like

$$h : \mathcal{X} \to \mathcal{Y},$$

the core idea of SVM^{Δ}_{multi} is to transform the learning problem into one of multivariate prediction. That is, the goal is to induce a hypothesis, $\bar{h}$, that maps all feature vectors of a sample $\bar{\boldsymbol{x}} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_S)$ to a tuple $\bar{y} = (y_1, \ldots, y_S)$ of S labels

$$\bar{h} : \bar{\mathcal{X}} \to \bar{\mathcal{Y}},$$

in which $\bar{\boldsymbol{x}} \in \bar{\mathcal{X}} = \mathcal{X}^S$ and $\bar{y} \in \bar{\mathcal{Y}} = \{-1, +1\}^S$. This multivariate mapping is implemented via a linear discriminant function

$$\bar{h}_{\boldsymbol{w}}(\bar{\boldsymbol{x}}) : \underset{\bar{y}' \in \bar{\mathcal{Y}}}{\arg\max} \; \{\langle \boldsymbol{w}, \Psi(\bar{\boldsymbol{x}}, \bar{y}')\rangle\},$$

where $\bar{h}_{\boldsymbol{w}}(\bar{\boldsymbol{x}})$ yields the tuple $\bar{y}' = (y'_1, \ldots, y'_S)$ of S predicted labels with a higher score according to the linear function defined by the parameter vector, $\boldsymbol{w}$. The joint feature map, Ψ, describes the match between a tuple of inputs and a tuple of outputs. For the quantification-oriented methods presented in this paper, we use the same form proposed by Joachims for binary classification

$$\Psi(\bar{\boldsymbol{x}}, \bar{y}') = \sum_{i=1}^{S} \boldsymbol{x}_i y'_i.$$

This setup allows the learner to consider the predictions for all the examples and, in turn, optimize a sample-based loss function, Δ. The optimization problem for obtaining $\boldsymbol{w}$ given a non-negative Δ is as follows

$$\begin{aligned} \min_{\boldsymbol{w},\xi\geq 0} \quad & \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w}\rangle + C\xi \\ \text{s.t.} \quad & \langle \boldsymbol{w}, \Psi\left(\bar{\boldsymbol{x}}, \bar{y}\right) - \Psi\left(\bar{\boldsymbol{x}}, \bar{y}'\right)\rangle \geq \Delta(\bar{y}, \bar{y}') - \xi, \quad \forall \bar{y}' \in \bar{\mathcal{Y}} \setminus \bar{y}. \end{aligned} \tag{9}$$

Notice that the constraint set of this optimization problem is extremely large, including one constraint for each tuple $\bar{y}'$. Solving this problem directly is intractable due to the exponential size of $\bar{\mathcal{Y}}$. Instead, we obtain an approximate solution applying Algorithm 1 described in [32]. The key idea of this algorithm is to iteratively construct a *sufficient subset* of the set of constraints. In each iteration, the most violated constraint is added to the active subset of constraints. The search for this constraint depends on the target loss function. Given any metric computed from the contingency table, such as the quantification loss functions defined previously, Algorithm 2 [32] efficiently returns the most violated constraint. Note that the non-negativity condition imposed on Δ implies that estimation bias cannot be optimized because it may return negative values.

3.2. Discussion

The two major frameworks described up to this point may present some drawbacks under specific conditions, as occurs with all learning paradigms. On the one hand, Forman's methods provide estimations that are obtained in terms of modified classification models, optimized to improve their classification accuracy, instead of training them to reduce their quantification error. Although these algorithms showed promising quantification performance in practice, it seems more orthodox to build quantifiers by optimizing a quantification metric, as stated by Esuli and Sebastiani.

However, their proposal does not take classification accuracy into account as long as the quantifier balances the number of errors between both classes, even at the cost of obtaining a rather poor classifier. That is, Esuli and Sebastiani propose that the learning method should optimize a quantification measure that simply deteriorates with $|FP - FN|$. We strongly believe that it is also important for the learner to consider the classification performance as well. Our claim is that this aspect is crucial to ensure a minimum level of confidence for the deployed models. The key issue is that pure quantification measures do not take into account the classification abilities of the model, producing several optimum points within the hypothesis search space (any that fulfills $FP = FN$). However, some of these hypotheses are less reliable than others.

In order to analyze this issue we shall use the example in Figure 1, which represents all instances of the iris dataset. This training set contains three classes, with the same percentage for each of them. The learning task is to obtain a quantifier, not a classifier, for class 3 (i.e., class 3 is the positive class) while the negative class comprises classes 1 and 2 and we need a model to predict the prevalence of class 3. The figure depicts two hypotheses: $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$; the former classifies all examples of class 1 as positives, while the latter predicts the majority of examples of class 3 as positives. Both hypotheses are perfect quantifiers for class 3 according to the training data. It is important to recall that all classes of the dataset have the same number of examples. For that reason, hypothesis $\boldsymbol{w}_1$ is a perfect quantifier for class 3 because it predicts the exact prevalence of class 1, which is the same prevalence as that of class 3. Any learning method that only takes quantification performance into account is not able to distinguish between $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. Our claim is that $\boldsymbol{w}_2$ should be prefered, because it is the better classifier, being more robust to changes in class distribution. Actually, $\boldsymbol{w}_1$ will quantify any change in the proportion of class 3 in the opposite direction due to the fact that the hyperplane defined by $\boldsymbol{w}_1$ is irrelevant in the distinction between positive and negative examples. That is, using $\boldsymbol{w}_1$, any increment in the proportion of class 3 results in a decrement in the quantification of that class, and vice-versa. In contrast, the estimations of $\boldsymbol{w}_2$ increase or decrease in the same direction as these changes.

Interestingly enough, the strength of Forman's approach is the weakness of the proposal presented by Esuli and Sebastiani, and vice-versa. While the latter approach emphasizes quantification ability during optimization, the former concentrates on building and characterizing classifiers in order to apply them as quantifiers. In this respect, our proposal may be able to soften these drawbacks, considering both classification and quantification performance during learning and thus producing more reliable and more robust quantifiers.

In fact, reliability is always a key issue when applying machine learning methods in practice. The question to be answered is how to measure the reliability that a quantifier offers, or whether it is reasonable for it not to be able to classify a minimum number of examples correctly.

The formal approach to obtain such quantifiers is to design a metric that somehow combines classification and quantification abilities and then apply a learning algorithm able to select a model that optimizes such a metric. This is the core idea of our proposal, which we shall introduce in the next section.

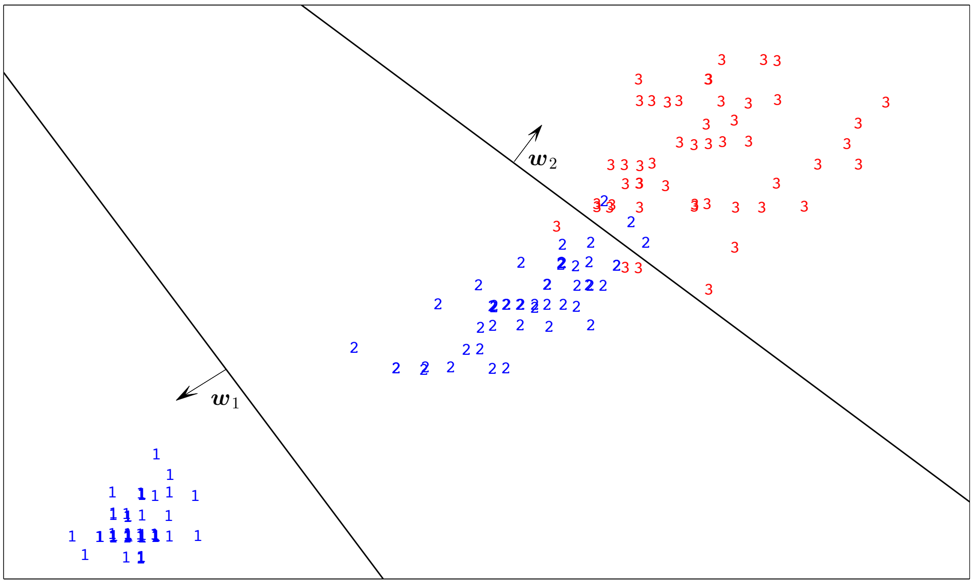


Figure 1: Graphical display of two conflicting perfect quantifiers

3.3. Our Proposal

Conceptually, the strategy of merging two complementary learning objectives is not new; we find the best example in information retrieval. The systems developed for such tasks are trained to balance two goals: retrieving as many relevant documents as possible, but discarding non-relevant ones. The metric that allows assessing how close these complementary goals are to being accomplished is $F\text{-}measure$ [36]. Actually, this metric emerges from the combination of two ratios: $recall$ (TP/P), which was already defined as tpr in (5), and $precision$ (TP/P'). In a certain respect, we face a similar problem in quantification.

The first element of our proposal is a new family of score functions, inspired by the aforementioned $F\text{-}measure$. We need two core ingredients, a metric for quantification and another for classification. The additional advantage of this approach is flexibility, in the sense that almost any combination of measures can be potentially selected by practitioners. This new family is mainly aimed at guiding model selection during the learning stage. But, to a certain extent, it also allows the comparison of quantifiers trained with different approaches, whether or not they are based on these ideas. Evaluating quantifiers from this twofold perspective assists us in analyzing their reliability.

3.4. Q-measure: balancing quantification and classification

All the above leads us to present a new metric, called $Q\text{-}measure$, which simultaneously balances quantification and classification performance. The first point worth noting is that quantification is mostly explored for binary problems, in which the positive class is usually more relevant and must be correctly quantified. Thus, the design of $Q\text{-}measure$ described in this paper is focused on a binary quantification setting.

In summary, our approach is based on a similar concept to the standard classification metric $F\text{-}measure$

$$F_\beta = (1+\beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} , \tag{10}$$

which balances an adjustable tradeoff between $precision$ and $recall$. Analogously, we suggest $Q\text{-}measure$, defined as

$$Q_\beta = (1+\beta^2) \cdot \frac{\mathit{cperf} \cdot \mathit{qperf}}{\beta^2 \cdot \mathit{cperf} + \mathit{qperf}} . \tag{11}$$

The β parameter allows weighting *cperf* and *qperf* measures, providing an AND-like behavior. Note that *cperf* and *qperf* stand for *classification performance* and *quantification performance*, respectively. The selection of these metrics depends on the final learning goal, bearing in mind that they should be bounded between 0 and 1 in order to be effectively combined, representing the worst and best case, respectively.

We now explore some alternatives through graphical representations. The motivation behind Figures 2, 3 and 4 is to enable us to analyze the behavior of different loss functions with respect to all combinations of values for FP and FN; both under balanced (2a, 3a and 4a) and unbalanced (2b, 3b and 4b) training conditions. Each of the 2D plots is the xy-projection of its lower 3D graph. Darker colors mean better scores. Notice also that 3D views are rotated over the z-axis in order to make it easier to visualize the surfaces and that the x-axis ranges are different between balanced and unbalanced cases. Intuitively, a well-conceived learning procedure should tend to move towards those models whose scores fall within the darker areas. In other words, these graphs illustrate the hypothesis search space of each metric.

3.4.1. Classification performance

In Figure 2 we review some candidate classification metrics. In line with the binary quantification setting introduced previously, a natural choice for *cperf* is $accuracy$, defined as $(TP + TN)/S$. However, this choice has some drawbacks, because quantification is usually applied over an unbalanced binary problem, in which negatives are the majority class, resulting from a combination of several related classes (*one-vs-all*).

Other standard alternatives are F_1, defined in Equation (10), and the geometric mean of tpr ($recall$) and tnr (*true negative rate*), defined as $GM = \sqrt{TP/P \cdot TN/N}$; i.e., the geometric mean of *sensitivity* and *specificity*. GM is particularly useful when dealing with unbalanced problems in order to mitigate the bias towards the majority class during learning [37].

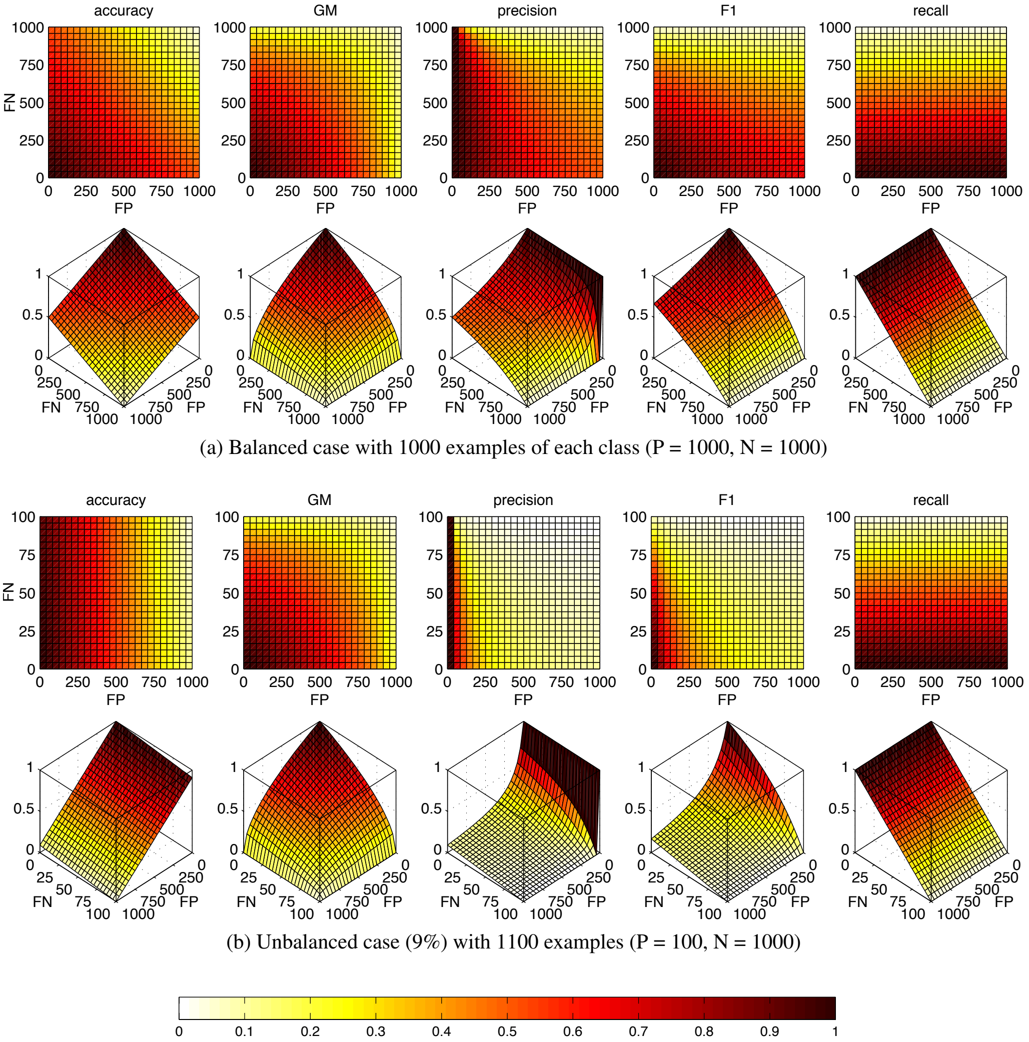
An interesting property of both tpr and tnr is that their respective search spaces are only defined over one of the two classes, and hence they are invariant to changes in the dimension of the other. Notice that the graphical representation of tnr is equivalent to tpr or $recall$ in Figure 2, though rotated 90º over the z-axis. That is why GM also shows a constant shape between balanced (Figure 2a) and unbalanced cases (Figure 2b), with a proper scaling for the y-axis. It is also worth noting that $accuracy$ approaches tnr when the size of the positive class is negligible ($(TP + TN)/S \approx TN/N$, when $P \to 0$).

Therefore, we believe that $accuracy$ may be appropriate only in those cases in which we are dealing with problems where both classes have a similar size, so we discard it. Regarding F_1 and GM, although both could be appropriate, we finally focus on $recall$ for our study. A potential benefit of maximizing $recall$ is that this may lead to a greater denominator in Equation (6), providing more stable corrections. The fact that this metric is included in $F\text{-}measure$ and GM is also of interest, in order to weight the relevance of the positive class accordingly. Thus, this decision is also supported by the fact that the goal of the applications described in quantification literature focuses on estimating the prevalence of the positive class, which is usually more relevant.

In practical terms, $Q\text{-}measure$ is able to discard pointless $qperf$ optimums thanks to the use of $recall$. The key aspect is that $recall$ acts as a *hook*, forcing the quantifier to avoid incoherent classification predictions over the positive class. This reduces the amount of FN errors, consequently restricting the search space for the quantification part in $Q\text{-}measure$. Notice also that pure quantification metrics tend to overlook positive class relevance in unbalanced scenarios.

3.4.2. Quantification performance

We considered several alternatives for $qperf$, starting from the standard measures described in Section 2. Unfortunately, none of the reviewed metrics fulfill all the requirements imposed by the design of $Q\text{-}measure$. Hence, we also analyze the normalized versions of AE and SE.



(a) Balanced case with 1000 examples of each class (P = 1000, N = 1000)

(b) Unbalanced case (9%) with 1100 examples (P = 100, N = 1000)

Figure 2: Graphical representation of all possible values for different classification loss functions, varying FP and FN between 0 and their maximum value, and with a fixed size for both P and N (see inner captions). Darker colors mean better scores.

Figure 3 provides a graphical representation to assist in the interpretation and discussion of these functions. However, it is worth mentioning that the decision regarding $qperf$ does not depend on whether we need to estimate the prevalence of one or both classes, because both values are complementary in binary problems ($p = 1 - n$, where n is the proportion of negatives or N/S).

Estimation bias is inappropriate because it can yield negative predictions. We also discard KLD because it is not properly bounded and it yields unwieldy results when estimated proportions are near 0% or 100%, like infinity or indeterminate values. According to [10], this problem can be resolved by backing off by half a count, which in our case means substituting the estimated proportion by $|p' - 0.5/S|$, when $p' \in \{0, 1\}$. Moreover, as can be observed in Figure 3, we also have to crop its range after subtracting from 1. These adjustments are not exempt from controversy, so we have focused on other alternatives.

We consider AE and SE, defined in Section 2.1.2, to be the most suitable candidates because both are bounded between 0 and 1. However, they do not reach a value of 1 for almost any possible class proportion, except for $p \in \{0, 1\}$, moving further away from 1 in correlation with the degree of imbalance (notice that the AE and SE values are substracted from 1 in Figure 3). This may result in an awkward behavior when combining these metrics with $cperf$ in Equation (11). Observe in Figure 2 that both components of $F\text{-}measure$ cover the whole range between 0 (worst) and 1 (best case), and as required by $Q\text{-}measure$.

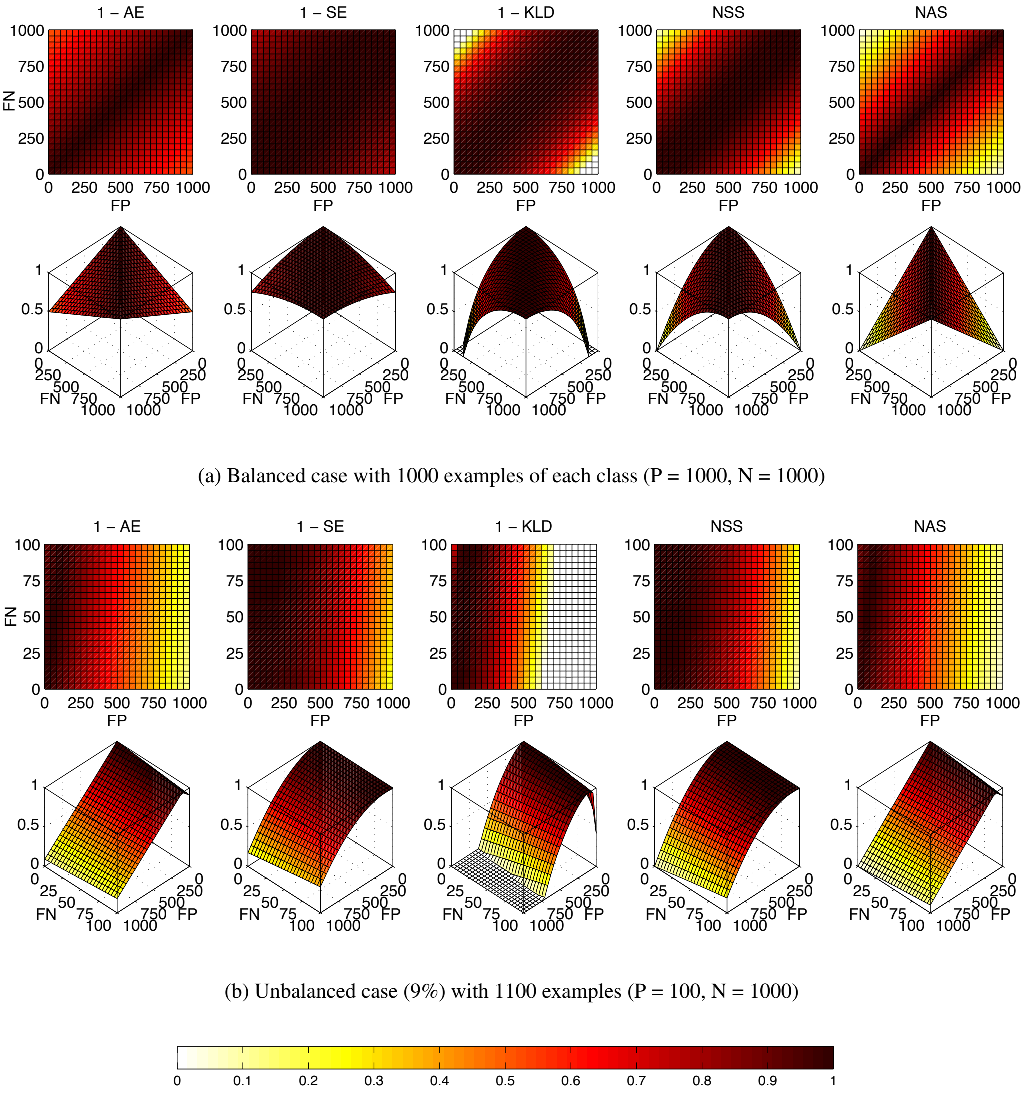
Looking at Equations (2) and (3) in more detail, we can see that, given a particular value for p, their effective upper bounds are $\max(p, n)$ and $\max(p, n)^2$, respectively. Therefore we need to normalize them. Moreover, as they are defined as loss functions, with the optimum at 0, we also need to redefine them as score functions. Taking into account these factors, we obtain two derived measures for quantification, denoted as *Normalized Absolute Score* (NAS)

$$NAS = 1 - \frac{|p' - p|}{\max(p, n)} = 1 - \frac{|FN - FP|}{\max(P, N)} \ , \tag{12}$$

and *Nomalized Squared Score* (NSS)

$$NSS = 1 - \left(\frac{p' - p}{\max(p, n)}\right)^2 = 1 - \left(\frac{FN - FP}{\max(P, N)}\right)^2 . \tag{13}$$

Figure 3 shows that NAS and NSS are uniform and easily interpretable, presenting equivalent shapes to those offered by standard quantification loss functions. For instance, NSS is



(a) Balanced case with 1000 examples of each class (P = 1000, N = 1000)

(b) Unbalanced case (9%) with 1100 examples (P = 100, N = 1000)

Figure 3: Graphical representation of all possible values for different quantification loss functions, varying FP and FN between 0 and their maximum value, and with a fixed size for both P and N (see inner captions). Darker colors mean better scores.

quite similar to 1-KLD. From Figure 3a, we can see that when the problem is balanced, then all functions return the best scores on the diagonal. This represents where the FP and FN values neutralize each other, i.e., where $|FP - FN|$ cancels out. Figure 3b, on the other hand, provides an example of an unbalanced problem. Once again, the optimal region lies above the line where these values cancel each other out, as may be expected.

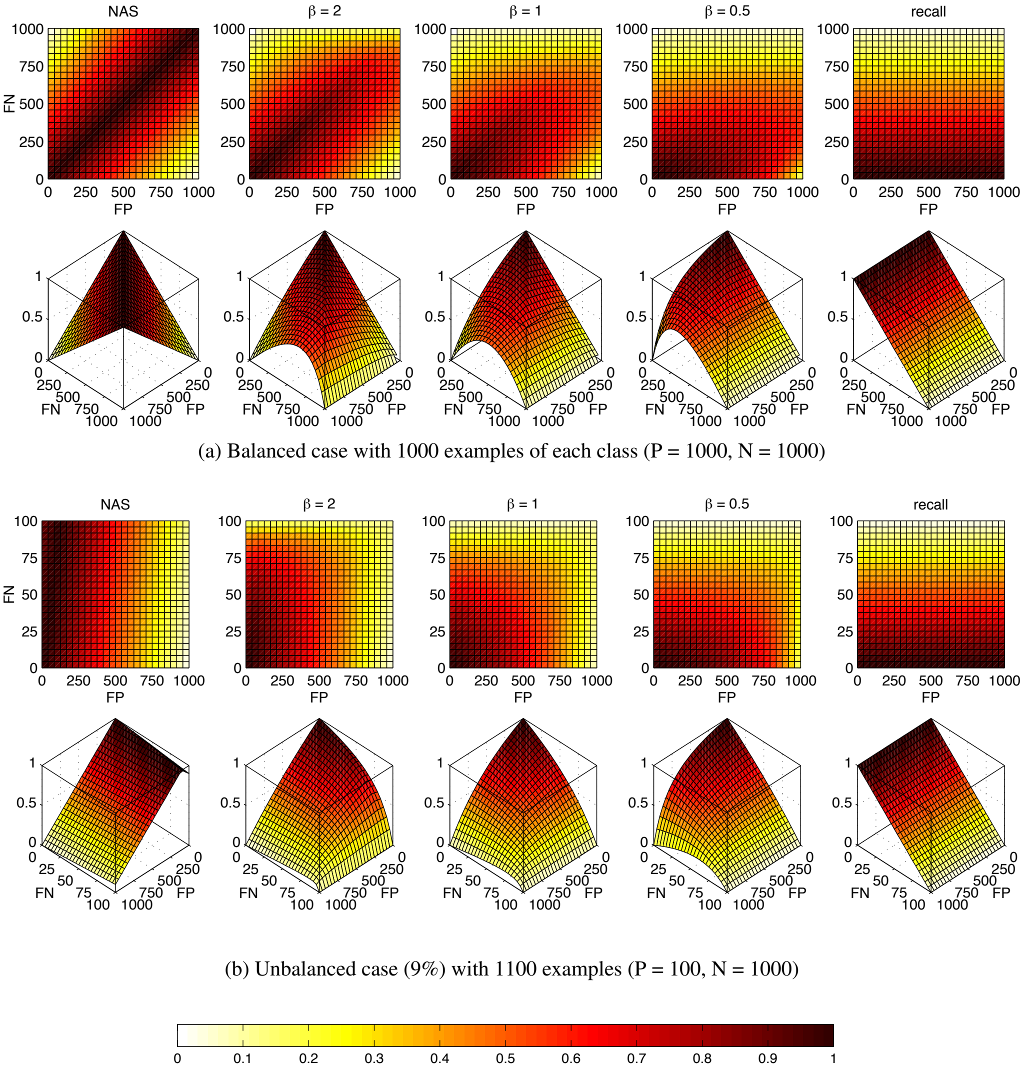
For the sake of simplicity, we only focus on NAS in our study. If we look for the maximum possible value of $|FP - FN|$, we conclude that it is always the number of individuals in the majority class. Assuming that N is greater than P, as is usual, the proof is that the worst quantification score is obtained when all the examples of the minority class are classified correctly ($TP = P$ and $FN = 0$), but all the examples of the majority class are misclassified ($TN = 0$ and $FP = N$), and thus Equation (12) evaluates to 0. With such a simple metric, we can see that the $|FP - FN|$ count is weighted in terms of the predominant class (denominator), forcing the output on the whole range between 0 and 1.

3.4.3. Graphical analysis of Q-$measure$

The graphical representation in Figure 4 provides an intuitive view to understand the behavior of Q-$measure$, selecting $recall$ as $cperf$ and NAS as $qperf$ for Equation (11). Its interpretation is exactly the same as in previous figures. Once again, we present two alternative learning conditions, balanced on the top (Figure 4a) and unbalanced on the bottom (Figure 4b). From left to right, we show different search spaces obtained from five target measures: first NAS, then those obtained from three different β values (Q_2, Q_1 and $Q_{0.5}$) , and finally $recall$. Notice that $recall$ and NAS are equivalent to Q_0 and Q_∞, respectively. When the value of β is 1 (on the middle graph), both the classification and quantification performance measures are equally weighted; when its value decreases to 0, then Q-$measure$ tends to be more similar to $cperf$; and when it rises above 1, it tends to resemble $qperf$. Obviously, for the intermediate values of β, the obtained search spaces are significantly different from those of the seminal metrics.

In summary, $recall$ drives the model to yield accurate predictions over the positive class, minimizing FN. Whereas, on the other hand, NAS evaluates the compensation between FP and FN. Hence, we have that Q-$measure$ degrades when $|FP - FN|$ is high, but we are also penalizing those models with high FN.

Observing Figure 4, we can foresee that the search space defined by $\beta = 2$ will produce competitive quantifiers. An interesting property of this learning objective is that Q_2 preserves



(a) Balanced case with 1000 examples of each class (P = 1000, N = 1000)

(b) Unbalanced case (9%) with 1100 examples (P = 100, N = 1000)

Figure 4: Graphical representation for the proposed loss function $Q\text{-}measure$, varying FP and FN between 0 and their maximum value, and with a fixed size for both P and N (see inner captions). Darker colors mean better scores. Each row shows the progression from NAS ($\beta \to \infty$) to $recall$ ($\beta = 0$) through different values of β.

the general shape of the optimal region defined by NAS, while degrading these optimums in consonance with $recall$. That is, it offers the benefits of a quantification-oriented target, avoiding incoherent optimums (see Section 3.2).

We can also observe that, with $\beta = 1$, we are forcing the learning method to obtain models in the proximities of the lower values of FP and FN. Specifically, in Figure 2b and Figure 4b, we see that the shape of Q_1 is reminiscent of that of GM when the dataset is unbalanced. This similarity arises from the fact that both share $recall$ as one of their components, while NAS is similar to tnr on highly unbalanced datasets. In the extreme case, when the positive class is minimal, the score $1 - AE$ is similar to NAS, $accuracy$ and tnr

$$1 - \frac{|FP - FN|}{N + P} \approx 1 - \frac{FP}{N} = \frac{TN}{N} \approx \frac{(TP + TN)}{N + P}, \text{ when } P \rightarrow 0.$$

Therefore, the main motivation for mixing in $recall$ is that using a pure quantification metric could imply optimizing a similar target to that of $accuracy$ or tnr on highly unbalanced problems. In fact, as we shall analyze in the following section, the empirical results obtained from our experiments suggest that the behavior of a model learned though NAS is very similar to that of CC, which is a classifier trained with $accuracy$. In balanced cases, we believe that the contribution of $recall$ to Q-$measure$ also offers a more coherent learning objective, providing more robust quantifiers in practice.

3.5. Learning algorithm

We use the same algorithms as described in Section 3.1 to implement a learning method for optimizing Q-$measure$. Actually, any metric obtained from the contingency table can be optimized with these algorithms. This includes any variation of Q-$measure$ based on different seminal metrics for $cperf$ and $qperf$.

4. Experiments

The main objective of this section is to study the behavior of the quantification methods presented in this paper, comparing their performance with other state-of-the-art approaches. The main difference with respect to the first experimental designs followed for quantification is that our empirical analysis neither focuses on a particular domain, nor on a specific range of train or test prevalences. We aim to cover a broader or more general scope, following the methodology that we have previously applied with success in [38]. Specifically, the experiments are designed to answer the following questions:

1. Do the empirical results support the use of a learner optimizing a quantification loss function instead of a classification performance measure?
2. Do we obtain any clear benefit by considering both classification and quantification simultaneously during learning?

The rest of the section is organized as follows. First we describe the experimental setup, including datasets, algorithms and statistical tests. We then present the results obtained from the experiments, evaluating them in terms of AE and KLD. Finally, we discuss these results, with the aim of providing answers to the aforementioned questions.

4.1. Experimental setup

As we introduced in [38], the required experiment methodology for quantification is relatively uncommon and has yet to be properly standardized and validated. The key difference with respect to traditional classification methodologies is that we need to evaluate performance over whole sets, rather than via individual classification outputs. Moreover, quantification assessment requires evaluating performance over a broad spectrum of test sets with different class distributions, instead of using a single test set.

We use benchmark datasets with known positive prevalences for performance measurement and comparison purposes, applying a variation of stratified 10-fold cross-validation. This setup preserves the original prevalence in all training iterations. Once a model has been trained with nine of the folds, the remaining one is used to generate 11 different random test sets with specific positive proportions ranging from 0% to 100%, in steps of 10%, by means of stratified sampling [15]. This setup ensures that the within-class distributions, $Pr(\boldsymbol{x}|y)$, are maintained between training and test, as stated in Section 2.2.2, seeing that random resampling is uniform and stratified.

We presume that this variation in the testing conditions may be rather unnatural, requiring more appropriate data collections. Changes in training and test conditions should be extracted directly from different snapshots of the same population, showing natural shifts in their distribution. As yet, however, we have not been able to find suitable collections of publicly available

Table 2: Summary of datasets

Dataset	Identifier	*Size*	*Attrs.*	*Pos.*	*Neg.*	*%pos.*
Balance Scale Weight & Distance (left)	balance.1	625	4	288	337	46%
Balance Scale Weight & Distance (balanced)	balance.2	625	4	49	576	8%
Balance Scale Weight & Distance (right)	balance.3	625	4	288	337	46%
Contraceptive Method Choice (no use)	cmc.1	1473	9	629	844	43%
Contraceptive Method Choice (long term)	cmc.2	1473	9	333	1140	23%
Contraceptive Method Choice (short term)	cmc.3	1473	9	511	962	35%
Cardiotocography Data Set (normal)	ctg.1	2126	22	1655	471	78%
Cardiotocography Data Set (suspect)	ctg.2	2126	22	295	1831	14%
Cardiotocography Data Set (pathologic)	ctg.3	2126	22	176	1950	8%
Haberman's Survival Data	haberman	306	3	81	225	26%
Johns Hopkins University Ionosphere Database	ionosphere	351	34	126	225	36%
Iris Plants Database (setosa)	iris.1	150	4	50	100	33%
Iris Plants Database (versicolour)	iris.2	150	4	50	100	33%
Iris Plants Database (virginica)	iris.3	150	4	50	100	33%
Sonar, Mines vs. Rocks	sonar	208	60	97	111	47%
SPECTF Heart Data	spectf	267	44	55	212	21%
Tic-Tac-Toe Endgame Database	tictactoe	958	9	332	626	35%
Blood Transfusion Service Center Data Set	transfusion	748	4	178	570	24%
Wisconsin Diagnostic Breast Cancer	wdbc	569	30	212	357	37%
Wine Recognition Data (1)	wine.1	178	13	59	119	33%
Wine Recognition Data (2)	wine.2	178	13	71	107	40%
Wine Recognition Data (3)	wine.3	178	13	48	130	27%

datasets offering these specific features.

4.1.1. Datasets

In order to enable a fair comparison between all methods, we select a collection of datasets from the UCI Machine Learning Repository [33], aiming to follow an unbiased criterion: problems with ordinal or continuous features with, at the most, three classes and ranges from 150 to 2,500 examples. The summary of the 22 datasets fulfilling these constraints is presented in Table 2. As the percentage of positive examples ranges between 8% and 78%, this offers the possibility of evaluating the methods over significantly different training conditions. For datasets that originally have more than two classes, we follow a one-vs-all decomposition approach.

4.1.2. Algorithms

We take CC, AC, Max, X, T50 and MS as state-of-the-art quantifiers from Forman's proposals, considering CC as the baseline. The underlying classifier for all these algorithms is a linear SVM from the *libsvm* library [39], with default parameters. The process of learning and threshold characterization, discussed in Sections 2.2.2 and 2.2.3, is common to all these models, reducing the total time of the experiment and guaranteeing an equivalent root SVM for them all. Moreover, as Forman points out, the MS method may behave oddly when the denominator in Equation (6) is too small, making it advisable to discard any threshold with $tpr - fpr < 1/4$. However, he does not make any recommendation in the case where there is no threshold that avoids said restriction. We therefore decided to fix these missing values with the values obtained by the Max method, which provides the threshold with the greatest value for that difference.

The group of models based on learning a classifier by optimizing a quantification metric consists of two approaches. On the one hand, there is our proposal, using $recall$ and NAS as seminal metrics (see Section 3.4). We consider three $Q\text{-}measure$ variants: Q0.5, Q1 and Q2, representing models that optimize Equation (11) with β at 0.5, 1 and 2, respectively. On the other hand, we also include a method called NAS, which represents the approach suggested by Esuli and Sebastiani [1], using NAS as the target measure. The reason for choosing NAS instead of any other quantification loss function is that we believe that both approaches should use the same quantification metric, only differing in the fact that our proposal combines such metric with $recall$. This guarantees a fair comparison. All these systems are learned by means of SVM^{Δ}_{multi} [32], described in Section 3.1.

4.1.3. Estimation of tpr and fpr

The estimations of *tpr* and *fpr* for quantification correction, defined in Equation (6), are obtained through a standard 10-fold cross-validation after learning the root model. Other alternatives like 50-fold or LOO are discarded because they are much more computationally expensive and are prone to yield biased estimations, producing uneven corrections in practice.

It is also worth noting that we do not apply this correction for Q0.5, Q1, Q2 or NAS. Hence, their end models just count how many items are predicted as positive, like in the CC method. This decision is supported by the fact that our main objective is to evaluate the performance of models obtained from the optimization of these metrics, isolated from any other factor. Moreover, given that these systems are based on SVM^{Δ}_{multi}, the estimation of *tpr* and *fpr* is much more expensive and it did not show a clear improvement in our preliminary experiments.

In fact, although the theory behind Equation (6) is well founded, in practice there exist cases where this correction involves a greater quantification error. However, these issues fall outside the scope of this study, offering an interesting opportunity to perform a more detailed analysis in future studies.

4.1.4. Adaptation of the Friedman-Nemenyi statistical test

Following Demšar [40], several two-step statistical test procedures were carried out. In each of these procedures, the first step consists of a Friedman test of the null hypothesis that all approaches perform equally in terms of a specific score or error metric. When this hypothesis is rejected, a Nemenyi post-hoc test is then conducted to compare the methods in a pairwise way. Both steps are based on the average of the ranks. The comparisons include 10 algorithms over 22 datasets or domains, evaluated over 11 different prevalences, resulting in 242 measurements per model.

Moreover, as Demšar notes, there are variations of the Friedman test which can consider multiple repetitions per dataset, provided that the observations are independent. However, since each collection of 11 test sets is sampled from the same fold, we cannot guarantee the assumption of independence among them. Thus, in order to take into account the differences between algorithms over several test prevalences from the same dataset, we first obtain their ranks for each test prevalence and then compute an average rank per dataset, which is used to rank algorithms on that domain. Therefore, we only consider the original number of datasets to calculate the *critical difference* (CD), rather than using all test cases, resulting in a more conservative value. The reason for this is not only the fact that the assumption of independence is not fulfilled, but also that the number of test cases is not bound. Otherwise, simply taking a wider range of prevalences to test would imply a lower CD value, which appears to be unjustified from a statistical point of view and can be prone to distorted conclusions. Thus, we consider that the 10 algorithms are compared over 22 domains, regardless of the number of prevalences that are tested for each of them, resulting in a CD of 2.8883 for the Nemenyi test at the 5% significance level.

It should be stressed that we afford equal weight to all test prevalences. However, the methodology that we propose is open to other interpretations, where the experimental design could assign larger weights to some prevalences or even the criterion followed to distribute the test prevalences may be neither linear nor uniform. This will depend mainly on the final aim of the experiment.

4.2. Results

This section presents the experimental results in terms of two standard quantification measures: AE and KLD. Each of these measures provides a different perspective. In summary, we collect results from 22 datasets, applying a stratified 10-fold cross-validation for them all and assessing the performance of the resulting model with 11 test sets generated from the remaining fold (see Section 4.1). Recall that only the quantification outputs provided by AC, X, Max, T50 and MS are adjusted by means of Equation (6).

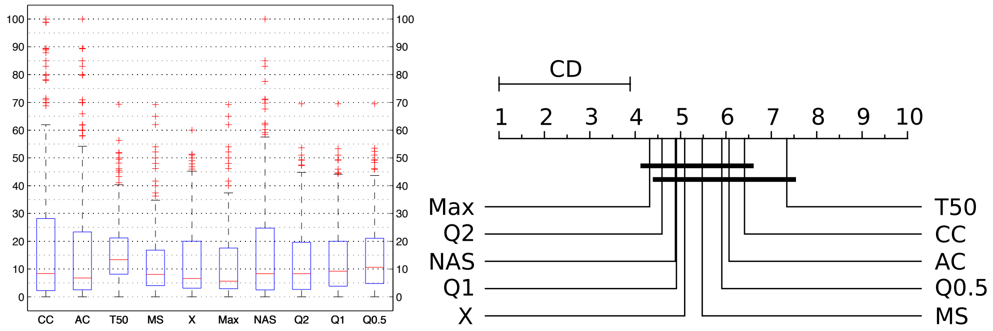
4.2.1. Analysis of AE measurements

The first approach that we follow is to represent the results for all test conditions in all datasets with a boxplot for each method under study. The idea is to show, in one single graph, the range of errors for a given metric of all the compared approaches. For instance, Figure 5a shows the ranges for AE measurements. Each box represents the first and third quartile by means of the lower and upper side, respectively, and the median or second quartile by means of the inner red line. The whiskers extend to the most extreme results not considered outliers, while outliers are plotted individually. In this case, we consider any point greater than the third quartile plus 1.5 times the inter-quartile range as an outlier. In this representation, it is better for a method to have lower quartile values, without outliers.

We distinguish three main groups in Figure 5a according to the learning procedure followed. The first one, including CC and AC, shows strong discrepancies between actual and estimated prevalences of up to 100%. These systems appear to be very unstable under specific circumstances. The second group includes T50, MS, X and Max, all of which are based on threshold selection policies (see Section 2.2.3). The T50 method stands out as the worst approach in this group due to the upward shift of its box. The final group comprises the $SVM^{\triangle}_{multi}$ models: Q0.5, Q1, Q2 and NAS. The Q_{β} versions of this last group seems more stable than NAS, without extreme values over 70 and showing more compact boxes.

Friedman's null hypothesis is rejected at the 5% significance level. The overall results of the Nemenyi test are shown in Figure 5b, in which each system is represented by a thin line, linked to its name on one side and its average rank on the other. The thick horizontal segments connect non significantly different methods at a confidence level of 5%. This plot suggests that Max and our proposal, represented by Q2, are the methods that perform best in this experiment in terms of AE score comparison for Nemenyi's test. In this setting, we have no statistical evidence of differences between the two approaches. Neither do they show clear differences with other systems. We can only appreciate that Max is significantly better than T50.

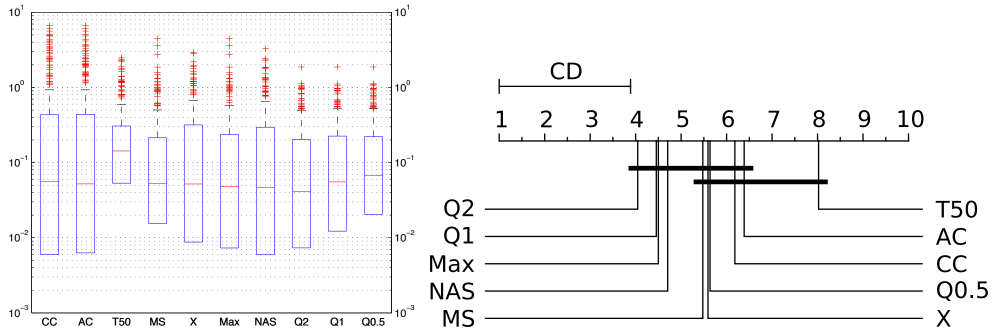
It is worth noting that the results of the Friedman-Nemenyi test are exactly the same for AE and NAS. The reason is that, given any two systems, their ranking order is equal in both metrics.



(a) Boxplots of all systems

(b) Nemenyi at 5% (CD = 2.8883)

Figure 5: Statistical comparisons in terms of AE results



(a) Boxplots of all systems

(b) Nemenyi at 5% (CD = 2.8883)

Figure 6: Statistical comparisons in terms of KLD results

The mathematical proof is straightforward. Note that this is not fulfilled for other metrics, like KLD.

4.2.2. Analysis of KLD measurements

Although the analysis of AE results could be sufficient in most cases to discriminate an appropriate model for a specific real-world task, we also provide a complementary analysis of our experiments in terms of KLD. From Figure 5, we can see that the differences between some systems are quite subtle in terms of AE, while in Figure 6 we observe that these differences are evidenced slightly more. For instance, Max and MS show larger outliers in terms of KLD, due to the fact that KLD is similar to a quadratic error (see Figure 3).

Analyzing the results of the Nemenyi test in Figure 6b, our approach obtains the best rank, represented again by Q2, which is designed to give more weight to the quantification metric during learning. However, except for T50, this system is not significantly better than other models.

Q1, Max and NAS are also statistically differentiable from T50.

4.3. Discussion

In order to make the discussion of the results clearer, we now aim to answer the questions raised at the beginning of this section:

1. Do the empirical results support the use of a learner optimizing a quantification loss function instead of a classification performance measure?

   The fact is that the best ranks are dominated by these kinds of methods, in conjunction with Max. However, the differences with respect to other systems are not statistically significant in general.

   In any case, our approach, initially suggested by Esuli and Sebastiani, is theoretically well-founded and is not based on any heuristic rule. From this point of view, we strongly believe that the methods presented here should be considered for future studies in the field of quantification. At the very least, they offer a different learning bias with respect to current approaches, which can produce better results in some domains.

   Moreover, it should also be stressed that none of the quantification methods evaluated in this experiment are corrected by means of Equation (6), as discussed in Section 4.1.3. Thus, these methods may be considered variants of CC, which can be further improved with similar strategies to those applied in AC, Max, X, MS and T50.

2. Do we obtain any clear benefit by considering both classification and quantification simultaneously during learning?

   As we suspected, our variant obtains better results than the original proposal by Esuli and Sebastiani in terms of pure quantification performance (see AE results in Figure 5 and KLD results in Figure 6).

   In some cases, NAS induces very poor classification models, despite benefiting from the definition of the optimization problem of SVM_{multi}^{Δ}, presented in Equation (9). Note that the constraints on the optimization problem are established with respect to the actual class of each example $(\Psi(\bar{\boldsymbol{x}}, \bar{y}) - \Psi(\bar{\boldsymbol{x}}, \bar{y}'))$, which would be produced by the *perfect classifier*. Thus, the algorithm is biased to those models similar to the perfect classifier even when the target loss function is not. In practice, however, this learning bias is not able to overcome the drawbacks derived from the intrinsic design of pure quantification metrics, which assigns an equal score to any model that simply neutralizes false positive errors with the same amount of false negative errors. Actually, our first intuition was that their proposal should provide even worse classifiers due to this fact. As we discuss in Section 3.2, the key problem is that pure quantification metrics produce several optimum points within the hypothesis search space, contrary to what occurs with other metrics, in which there is only one.

In summary, not only does our approach provides better quantification results than NAS, but we also consider it to be more reliable in general. Moreover, it is more flexible, allowing the practitioner to adjust the weight of both components of $Q\text{-}measure$ taking into account the specific requirements of the problem under study by means of the β parameter. In fact, provided that when $\beta \to \infty$ our method optimizes only the quantification component, it includes NAS as a particular case. This calibration is not needed in general and can be fixed via the experimental design. As a rule of thumb, we suggest $\beta = 2$, because, in line with the discussion of Figure 4 and the analysis of the empirical results, it effectively combines the best features of both components.

5. Concluding remarks

Esuli and Sebastiani point out that state-of-the art quantification algorithms do not optimize the loss function applied during model selection or comparison. Following their line of research, we claim that optimizing only a quantification metric during model training does not sufficiently address the problem, as we could obtain quantifiers with poor quantification behavior due to an incoherent underlying model in terms of classification abilities. In this regard, the most important question behind our study is whether it is actually advisable to rely on quantification models that do not distinguish between positives and negatives at an individual level. But, how could this issue be mitigated during quantifier training? Formally, the way to solve any machine learning problem comprises two steps: define a suitable metric and design an algorithm that optimizes it. The combination of $Q\text{-}measure$ and the multivariate algorithm proposed by Joachims offers a formal solution for quantifier learning.

Our main contributions are: i) the study of the first quantification-oriented learning approach, i.e., the first algorithm that optimizes a quantification metric; and ii) the definition of a parametric loss function for quantification. This proposal is not only theoretically well-founded, but also offers competitive performance on benchmark datasets compared with state-of-the-art quantifiers.

Acknowledgments

This work was supported in part by the Spanish Ministerio de Economía y Competitividad, under research project TIN2011-23558. The contribution of Jose Barranquero was also supported by FPI grant BES-2009-027102.

References

[1] A. Esuli, F. Sebastiani, Sentiment quantification, IEEE Intelligent Systems 25 (2010) 72–75.

[2] L. Tang, H. Gao, H. Liu, Network quantification despite biased labels, in: Proceedings of the 8th Workshop on Mining and Learning with Graphs, ACM, pp. 147–154.

[3] A. Guerrero-Curieses, R. Alaiz-Rodriguez, J. Cid-Sueiro, Cost-sensitive and modular land-cover classification based on posterior probability estimates, International Journal of Remote Sensing 30 (2009) 5877–5899.

[4] L. Sánchez, V. González-Castro, E. Alegre-Gutiérrez, R. Alaiz-Rodríguez, Classification and quantification based on image analysis for sperm samples with uncertain damaged/intact cell proportions, in: Image Analysis and Recognition, LNCS 5112, Springer, 2008, pp. 827–836.

[5] Y. Chan, H. Ng, Estimating class priors in domain adaptation for word sense disambiguation, in: Proc. of the 21st International Conference on Computational Linguistics, ACL, pp. 89–96.

[6] 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, pp. 852–861.

[7] D. Hand, Classifier technology and the illusion of progress, Statistical Science 21 (2006) 1–14.

[8] F. Provost, T. Fawcett, Robust classification for imprecise environments, Machine Learning 42 (2001) 203–231.

[9] T. Rakthanmanon, E. Keogh, S. Lonardi, S. Evans, MDL-based time series clustering, Knowledge and Information Systems 33 (2012) 371–399.

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

[11] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, John Wiley & Sons, 2001.

[12] M. Kelly, D. Hand, N. Adams, The impact of changing populations on classifier performance, in: Proceedings of ACM SIGKDD'99, pp. 367–371.

[13] R. Alaiz-Rodríguez, N. Japkowicz, Assessing the impact of changing environments on classifier performance, in: Advances in AI, LNCS 5032, Springer-Verlag, 2008, pp. 13–24.

[14] T. Fawcett, P. Flach, A response to Webb and Ting's on the application of ROC analysis to predict classification performance under varying class distributions, Machine Learning 58 (2005) 33–38.

[15] J. Fleiss, B. Levin, M. Paik, Statistical Methods for Rates and Proportions, Wiley Series in Probability and Statistics, John Wiley & Sons, 2003.

[16] G. Weiss, F. Provost, Learning when training data are costly: The effect of class distribution on tree induction, Journal of Artificial Intelligence Research 19 (2003) 315–354.

[17] G. Webb, K. Ting, On the application of ROC analysis to predict classification performance under varying class distributions, Machine Learning 58 (2005) 25–32.

[18] R. Holte, Elaboration on two points raised in "classifier technology and the illusion of progress", Statistical Science 21 (2006) 24–26.

[19] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, N. Lawrence (Eds.), Dataset Shift in Machine Learning, The MIT Press, 2009.

[20] J. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recognition 45 (2012) 521–530.

[21] A. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (1997) 1145–1159.

[22] T. Fawcett, ROC graphs: Notes and practical considerations for researchers, Machine Learning 31 (2004) 1–38.

[23] C. Drummond, R. Holte, et al., Cost curves: An improved method for visualizing classifier performance., Machine Learning 65 (2006) 95–130.

[24] P. Latinne, M. Saerens, C. Decaestecker, Adjusting the outputs of a classifier to new a priori probabilities may significantly improve classification accuracy: Evidence from a multi-class problem in remote sensing, in: Proceedings of ICML'01, M. Kaufmann, 2001, pp. 298–305.

[25] S. Vucetic, Z. Obradovic, Classification on data with biased class distribution, in: Proceedings of ECML'01, Springer-Verlag, pp. 527–538.

[26] M. Saerens, P. Latinne, C. Decaestecker, Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure, Neural Computation 14 (2002) 21–41.

[27] J. Xue, G. Weiss, Quantification and semi-supervised classification methods for handling changes in class distribution, in: Proceedings of ACM SIGKDD'09, ACM, pp. 897–906.

[28] R. Alaiz-Rodríguez, A. Guerrero-Curieses, J. Cid-Sueiro, Class and subclass probability re-estimation to adapt a classifier in the presence of concept drift, Neurocomputing 74 (2011) 2614–2623.

[29] R. Alaiz-Rodríguez, A. Guerrero-Curieses, J. Cid-Sueiro, Minimax regret classifier for imprecise class distributions, Journal of Machine Learning Research 8 (2007) 103–130.

[30] E. Ramentol, Y. Caballero, R. Bello, F. Herrera, SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory, Knowledge and Information Systems 33 (2012) 245–265.

[31] A. Bella, C. Ferri, J. Hernández-Orallo, M. Ramírez-Quintana, Quantification via probability estimators, in: Proc. of the 10th IEEE International Conference on Data Mining, pp. 737–742.

[32] T. Joachims, A support vector method for multivariate performance measures, in: Proceedings of ICML'05, ACM, pp. 377–384.

[33] K. Bache, M. Lichman, UCI machine learning repository, 2013.

[34] G. Forman, Counting positives accurately despite inaccurate classification, in: Proceedings of ECML'05, Springer-Verlag, 2005, pp. 564–575.

[35] G. Forman, Quantifying trends accurately despite classifier error and class imbalance, in: Proceedings of ACM

SIGKDD’06, ACM, pp. 157–166.

[36] C. van Rijsbergen, Foundations of evaluation, Journal of Documentation 30 (1974) 365–373.

[37] R. Barandela, J. Sánchez, V. García, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognition 36 (2003) 849–851.

[38] J. Barranquero, P. González, J. Díez, J. J. del Coz, On the study of nearest neighbour algorithms for prevalence estimation in binary problems, Pattern Recognition 46 (2013) 472—482.

[39] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 1–27.

[40] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.