On the study of nearest neighbor algorithms for prevalence estimation in binary problems

José Barranquero[1,], Pablo González[1], Jorge Díez[1], Juan José del Coz[1]

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

Abstract

This paper presents a new approach for solving binary quantification problems based on nearest neighbor (NN) algorithms. Our main objective is to study the behavior of these methods in the context of prevalence estimation. We seek for NN-based quantifiers able to provide competitive performance while balancing simplicity and effectiveness. We propose two simple weighting strategies, PWK and PWK$^{\alpha}$, which stand out among state-of-the-art quantifiers. These proposed methods are the only ones that offer statistical differences with respect to less robust algorithms, like CC or AC. The second contribution of the paper is to introduce a new experiment methodology for quantification.

Keywords: quantification, prevalence estimation, nearest neighbor, methodology

1. Introduction

There is growing interest within the machine learning community regarding the accurate estimation of the distribution of classes from a sample. This relatively new task, termed *quantification*, deals with the prediction of the prevalence of the positive class over a specific dataset. In practical terms, the key objective

*Corresponding author. Phone: +34 985 18 2501, Fax: +34 985 182125

is to estimate the class distribution of a test set, provided that we have a training set in which this distribution may be noticeably different. Intuitively, this task is directly related to tracking of trends over time, such as early detection of epidemics, endangered species, market and ecosystem evolution, and other kinds of distribution changes in general.

However, quantification has been an unattractive problem that has barely been addressed in machine learning research due to the mistaken belief that it is somewhat trivial. Nevertheless, this is not necessarily true, because different distributions of training and test data can have a huge impact on the performance of traditional machine learning algorithms, which usually assume that both samples are obtained from identical populations.

In this paper we present an extensive study, analyzing the experimental results from alternative perspectives. The aim is to explore the applicability of *nearest neighbor* (NN) algorithms for binary quantification, using standard benchmark datasets from different domains [**?**]. Similar NN approaches have been successfully applied in a wide range of learning tasks, providing simple and competitive algorithms for classification [**?**], regression [**?**], ordinal regression [**?**], clustering [**?**], preference learning [**?**] and multi-label [**?**] problems, among others.

The motivational intuition beyond this work is that the inherent behavior of NN algorithms should yield appropriate quantification results based on the assumption that they may be able to *remember* details of the topology of the data, independently of the presence of distribution changes between training and test. Moreover, bearing in mind that once the distance matrix has been constructed we are able to compute many different estimations in a straightforward way, we shall explain why we consider that these methods offer a cost-effective alternative for

this problem. At the very least, they reveal themselves to be competitive baseline approaches, providing performance results that challenge more complex methods proposed in previous papers.

In summary, we seek for a quantification approach with competitive performance that could offer simplicity and robustness. Earlier proposals are mostly based on SVM classifiers [**?** **?**], which are one of the most effective state-of-the-art learners. These previous quantification methods showed promising empirical results due to theoretical developments aimed at correcting the aggregation of individual classifier outputs. Thus, our main hypothesis is whether we could apply the aforementioned theoretical foundations with simpler classifiers, such as NN-based algorithms, in order to stress the relevance of corrections of this kind over the use of any specific family of classifiers as base learners for quantification. The second objective of the paper is to develop a new experiment methodology for the task of quantification based on the widespread 10-fold cross-validation (CV) procedure and the two step Friedman-Nemenyi statistical test. This methodology is adapted to the inherent requirements of quantification, which demand evaluating performance over whole sets rather than by means of individual classification outputs. Moreover, quantification assessment also requires evaluating performance over a broad spectrum of test distributions in order for it to be representative.

Quantification is introduced in Section **??** and the NN algorithms used in this paper are presented in Section **??**. We describe our experiment setup and the empirical results in Section **??**, analyzing these in detail. Finally, we discuss the main conclusions and future research paths in Section **??**.

2. Binary quantification

From a statistical point of view, this task is aimed at estimating the prevalence of an event or feature within a sample. During learning stage, we have a training set with examples labeled as positive or negative, showing a specific distribution that can be summarized with the proportion of positives or prevalence (p). The learning objective is to obtain a model being able to predict the prevalence of other samples that may show a remarkably different distribution of classes. Thus, the input data is equivalent to that of traditional classification problems, but the focus is stressed over the estimated prevalence of the sample (p'), rather than the predicted class of each example.

It is worth noting that quantification methods are currently based on classification algorithms. After a surface exploration of the problem, the first intuition tends to emerge as a straightforward solution based on counting the predictions of each class. This method is identified as *Classify & Count* (CC) by George Forman [**?**]. Provided that we use a classifier offering state-of-the-art performance, we could be tempted to consider this method to be both effective and competitive. However, this is not the case unless we have access to a perfect classifier, providing zero misclassified outputs. Unfortunately, the fact is that this scenario is unrealistic for real-world problems.

For instance, given a binary quantification task in which the learned classifier tends to misclassify some examples mostly from the positive class, then the derived quantifier will certainly underestimate the proportion of that class. Furthermore, when the prevalence of the positive class increases uniformly in a test set, then the number of misclassified positive instances also increases and the quantifier will yield a greater negative bias in the estimation of the proportion of positive

class. This effect becomes even more troublesome in a changing environment, in which the test distribution is usually substantially different from that of the training set. Appropriately addressing this issue is crucial for solving quantification problems. Forman pointed out and studied this behavior for binary quantification, proposing several methods to undertake this classification bias [**?**].

The notation that we shall employ throughout the paper is as follows: given a test sample, S represents its size, P the count of actual positives and N the count of actual negatives. Once trained a classifier, we have that P' is the count of individuals of that sample 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* of that model, respectively.

There are two main issues to note about the equations behind the actual prevalence and the predicted prevalence:

$$p = \frac{P}{S} = \frac{TP + FN}{S}, \quad \text{and} \quad p' = \frac{P'}{S} = \frac{TP + FP}{S}. \tag{1}$$

On the one hand, they only differ with respect to one term, being FN and FP respectively. This means that both FN and FP values may play an important role during performance evaluation, as we shall cover in Section **??**. On the other hand, p' comprises both TP and FP, closely related to the *true positive rate* and the *false positive rate*, defined as

$$tpr = \frac{TP}{P} \quad \text{and} \quad fpr = \frac{FP}{N}. \tag{2}$$

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, $P(y)$, changes, but the within-class densities, $P(\boldsymbol{x}|y)$, do not. This implies in turn that tpr and fpr are independent of shifts in class distribution.

These assumptions are fulfilled, for instance, when the changes in class priors are obtained by means of stratified sampling [**? ?**].

2.1. Quantification via adjusted classification

From (**??**), we know that p' depends exclusively on TP and FP. Thus, due to (**??**), only the tpr fraction of any change in P will be perceived by the classifier. Moreover, the fpr fraction of N will be misclassified by CC as positives. According to these observations, Forman [**?**] states the following theorem and proof:

Theorem 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.*

Proof. The expected prevalence p' of classifier outputs over the test set, written as a function of the actual positive prevalence p, is

$$p'(p) \;=\; tpr \cdot p \;+\; fpr \cdot (1-p) \tag{3}$$

Given that $p'(p^*) = p^*$, then for a strictly different prevalence $p^* + \Delta$, where $\Delta \neq 0$, CC does not produce the correct prevalence

$$p'(p^* + \Delta) = tpr \cdot (p^* + \Delta) \;+\; fpr \cdot (1 - (p^* + \Delta)) = p^* + (tpr - fpr) \cdot \Delta.$$

Moreover, since Forman's theorem assumes an imperfect classifier, then we have that $(tpr - fpr) < 1$, and thus

$$p'(p^* + \Delta) \begin{cases} < p^* + \Delta & \text{if } \Delta > 0 \\ > p^* + \Delta & \text{if } \Delta < 0. \end{cases}$$

□

Therefore, the CC method underestimates when the prevalence increases, and overestimates when it decreases. With the aim of correcting this bias, Forman proposed [**?**] a new method termed *Adjusted Count* (AC). The process consists in training a classifier and estimating its tpr and fpr characteristics through cross-validation over the training set. The next step is then to count the positive predictions of the classifier over the test examples (as in the CC method), but adjusting this estimation by means of the following formula derived from Equation (**??**)

$$p = \frac{p'(p) - fpr}{tpr - fpr}, \tag{4}$$

Since tpr and fpr are estimated through cross-validation, we obtain an approximation of the actual proportion. Hence, the accuracy of this adjusted estimation is strongly influenced by the accuracy in the estimation of these rates. In some cases, this leads to infeasible estimates of p, requiring a final step in order to clip the estimation into the range $[0, 1]$.

2.2. Threshold selection policies

A key problem related to the AC method is that its performance depends mostly on the degree of imbalance of the training set, degrading when the positive class is scarce [**?**]. This happens because its natural threshold usually tries to minimize the false positive errors by keeping a very low tpr, resulting in a small denominator in Equation (**??**). This fact produces a high vulnerability to variations in the estimation of tpr or fpr.

Therefore, Forman also proposed alternative imbalance-tolerant methods based on the selection of classifier thresholds. The main intuition is that selecting a threshold that allows more true positives, at the cost of many more false positives,

could provide better corrections and hence more accurate quantification. The objective is to choose those thresholds where the estimates of tpr and fpr present less variance or where the denominator in Equation (**??**) is big enough to be more robust with respect to estimation errors. In this study we assess the same threshold selection policies as in Forman's experiment [**?**]. The first one is Max, which chooses the threshold where the denominator ($tpr - fpr$) is maximized. The second one is the X policy, which takes the threshold where fpr equals $1 - tpr$, avoiding the tails of both curves. Finally, T50 eludes the tails of the tpr curve by selecting the threshold where 50% of positives are correctly estimated.

However, there is a drawback underlying all these threshold selection policies related to the fact that the estimation of tpr and fpr may differ significantly from the actual values. Hence, with the aim of enhancing the robustness of these approaches, Forman proposed the *Median Sweep* (MS) method. In this case, rather than selecting a specific threshold, the tpr and fpr information from all thresholds is exploited. During testing, this ensemble model is used to estimate the corrected prevalence with all available thresholds, using their median as the final output.

2.3. Learning methodology

The learning procedure established by Forman does not involve the calibration of the underlying SVM parameters. He states [**?**] that the focus is no longer on the accuracy of individual outputs, but on the correctness of the aggregated estimations. Thus, in some sense, the *goodness* of the original classifier is not relevant, as long as its predictions are correctly adjusted.

However, the estimations of tpr and fpr obtained from calibrated SVM models, previously adjusting the regularization parameter C, are more robust and provide better quantification results in practice. Moreover, this improvement is

also noticed for the CC method, which does not involve any kind of correction. Therefore, our proposed learning process starts by selecting the best value for the regularization parameter through a grid-search procedure (see Section **??**). Once this optimized model has been obtained, its default threshold is varied over the spectrum of raw training outputs, and the tpr and fpr values for each of these thresholds are estimated through cross-validation. After collecting all this information, several threshold selection policies can be applied in order to prepare the classifier for the following step, as already set out in Section **??**. Each of these strategies provides a derived model which is ready to be used and compared.

3. Nearest neighbor quantification

The goal of the paper is to study the behavior of nearest neighbor (NN) algorithms for prevalence estimation in binary problems. It is well-known that each learning paradigm presents a specific learning bias, which is best suited for some particular domains. As it happens in other machine learning tasks, we expect that NN approaches should outperform other methods in some quantification domains. Our first intuition is that the inherent behavior of NN algorithms should yield appropriate quantification results based on the assumption that they may be able to remember details of the topology of the data.

Furthermore, NN approaches present significative advantages in order to build an AC-based quantifier. In fact, they allow to implement more efficient methods for estimating tpr and fpr, which are required to compute the quantification correction defined in Equation (**??**). The standard procedure for the computations of these rates is cross-validation [**?**]. When working with SVM as base-learner for AC, we have to re-train a model for each partition, while NN approaches allow

us to compute the distance matrix once and use it for all partitions. Thus, we can estimate tpr and fpr at a small computational cost, even applying a *leave-one-out* (LOO) procedure, which may provide a better estimation for some domains.

3.1. k-nearest neighbor algorithm

One of the best known NN-based methods is the k-nearest neighbor (KNN) algorithm. Despite its simplicity, it has been demonstrated to yield very competitive results in many real world situations. In fact, Cover and Hart [**?**] pointed out that the probability of error of the NN rule is upper bounded by twice the Bayes probability of error.

Given a binary problem, represented by a collection of labels $Y = (y_1, ..., y_n)$ and their corresponding predictor features $\boldsymbol{X} = (\boldsymbol{x_1}, ..., \boldsymbol{x_n})$, with $y_i \in \{+1, -1\}$, then, for a test example $\boldsymbol{x}_j$, the resulting output $\hat{y}_j$ for KNN is computed as

$$\hat{y}_j = \text{sign}\left(\sum_{i \sim j}^{k} y_i \right); \tag{5}$$

where $i \sim j$ denotes the k-nearest neighbors of the test example $\boldsymbol{x}_j$.

Regarding the selection of k, Hand and Vinciotti [**?**] pointed out that, as the number of neighbors determines the bias versus variance tradeoff of the model, the value assigned to k should be smaller than the smallest class. This is especially relevant with unbalanced datasets, which is the common case in many domains. Another widely cited study, by Enas and Choi [**?**], proposes $n^{2/8}$ or $n^{3/8}$ as heuristic values, arguing that the optimal k is a function of the dimension of the sample space, the size of the space, the covariance structure and the sample proportions. In practice, however, this optimal value is usually determined empirically through a standard cross-validation procedure. Moreover, the selection of an appropriate

metric or distance is also decisive and complex, in which the Euclidean norm is usually the default option (known as *vanilla* KNN). For our study we decided to simplify all these decisions where possible, limiting our search to selecting the k value that leads to better empirical performance through a grid-search procedure (see Section **??**), and using the Euclidean distance.

3.2. Weight-based k-nearest neighbor

Although KNN has provided competitive quantification results in our experiments, Forman states that quantification models should be ready to learn from highly imbalanced datasets, like in one-vs-all multiclass scenarios or in narrowly defined categories. This gave us the idea of complementing it with weighting policies, mainly those depending on class proportions, in order to counteract the bias towards the majority class.

The main drawback when addressing the definition of a suitable strategy for any weight-based method is the broad range of weighting alternatives depending on the focus of each problem or application. Two major directions for assigning weights in NN-based approaches are identified by Kang and Cho [**?**]. On the one hand, we can assign weights to features or attributes before distance calculation, usually through specific kernel functions or flexible metrics [**?**]. On the other hand, we can assign weights to each neighbor after distance calculation. We have focused our efforts on the latter approach.

This problem has already been studied by Tan [**?**], as the core of neighbor-weighted k-nearest neighbor (NWKNN) algorithm, mostly aimed at unbalanced text problems. Tan's method is based on assigning two complementary weights for each test document: one based on neighbour distributions and another based on similarities between documents. The former assigns higher relevance to smaller

classes and the latter adjusts the contribution of each neighbor by means of its relative distance to the test document. Similarly as in (**??**), for a binary problem and given a test example $\boldsymbol{x}_j$, the estimated output can be obtained as

$$\hat{y}_j = \text{sign}\left(\sum_{i\,\sim\,j}^{k} \text{sim}(\boldsymbol{x}_i, \boldsymbol{x}_j)\; y_i\; w_{y_i}\right). \tag{6}$$

We discarded similarity score for our study,

$$\hat{y}_j = \text{sign}\left(\sum_{i\,\sim\,j}^{k} y_i\; w_{y_i}\right), \tag{7}$$

simplifying the notation and the guidelines for computing the class weights described by Tan. In summary, he proposes class weights that balance the relevance between classes, compensating the natural influence bias of bigger classes in multi-class scenarios. He also includes an additional parameter, which can be interpreted as a shrink factor: when this parameter grows, the penalization of bigger classes is softened progressively. In this paper, we use α to identify this parameter. We compute each class weight during training as the adjusted quotient between the cardinalities of that class (N_c) and the minority class (M)

$$w_c^{(\alpha)} = \left(\frac{N_c}{M}\right)^{-1/\alpha}, \text{with } \alpha \geq 1 \tag{8}$$

Therefore, the bigger the class size observed during training, the smaller its weight. To illustrate this fact, Table **??** shows the weights assigned to one of the classes, varying its prevalence from 1% to 99% for different values of α. Note that when we compute the weight of the minority class, or when the problem is balanced (50%), we always get a weight of 1; i.e., there is no penalization. However, when we compute the weight for the majority class, we get a penalizing weight

Table 1: PWK$^{\alpha}$ weights w.r.t. different training prevalences (binary problem)

α	1%	$\cdots$	50%	60%	70%	80%	90%	99%
1	1	$\cdots$	1	0.67	0.43	0.25	0.11	0.01
2	1	$\cdots$	1	0.82	0.65	0.50	0.33	0.10
3	1	$\cdots$	1	0.87	0.75	0.63	0.48	0.22
4	1	$\cdots$	1	0.90	0.81	0.71	0.58	0.32
5	1	$\cdots$	1	0.92	0.84	0.76	0.64	0.40

ranging from 0 to less than 1. The simplified algorithm defined by (**??**) and (**??**) is renamed as the proportion-weighted k-nearest neighbor (PWK$^{\alpha}$) algorithm.

As an alternative to Equation (**??**), we propose the following class weight

$$w_c = 1 - \frac{N_c}{S}, \tag{9}$$

which produces equivalent weights for $\alpha = 1$. This expression makes it easier to see that each weight w_c is inversely proportional to the size of the class c, with respect to the total size of the sample, denoted by S.

Theorem 2. *For any binary problem, the prediction rule in Equation* (**??**) *produces the same results regardless of whether class weights are calculated using Equation* (**??**) *or Equation* (**??**)*, fixing* $\alpha = 1$.

Proof. Let c_1 be the minority class and c_2 the majority class, then the idea is to prove that weights $w_{c_1}^{(1)}$ and $w_{c_2}^{(1)}$, computed by means of (**??**), are equal to their respective w_{c_1} and w_{c_2}, computed by means of (**??**), when they are divided by a unique constant, which happens to be equal to w_{c_1}. For the majority class:

$$w_{c_2}^{(1)} = \frac{N_{c_1}}{N_{c_2}} = \frac{N_{c_1}/S}{N_{c_2}/S} = \frac{1 - N_{c_2}/S}{1 - N_{c_1}/S} = \frac{w_{c_2}}{w_{c_1}}.$$

Given that by definition $w_{c_1}^{(1)} = 1$, we can rewrite it as $w_{c_1}^{(1)} = w_{c_1} \,/\, w_{c_1}$. Thus, if we fix $\alpha = 1$ in (**??**) and divide all the weights obtained from (**??**) by the mi-

nority class weight, w_{c_1}, the weights obtained from both equations are equivalent and prediction results are found to be equal. □

The combination of (**??**) and (**??**) is identified as PWK in our experiments. We initially considered this simplified PWK method as a naïve baseline for weighted NN approaches. However, despite their simplicity, the resulting models have shown competitive results in our experiments.

The key benefit of PWK$^\alpha$ over PWK is that the former provides additional flexibility to further adapt the model to each dataset through its α parameter, usually increasing precision when α grows, but decreasing recall. Conversely, PWK$^\alpha$ requires a more expensive training procedure due to the calibration of this free parameter. Our experiments in Section **??** suggest no statistical difference between both, so the final decision for a real-world application should be taken in terms of the specific needs of the problem, the constraints of the environment, or the complexity of the data, among others.

It is worth noting that for binary problems when α tends to infinity Equation (**??**) produces a weight of 1 for both classes, and given that PWK$^\alpha$ is equivalent to PWK when $\alpha = 1$, then KNN and PWK can be interpreted as particular cases of PWK$^\alpha$. The parameter α can be thus reinterpreted as a tradeoff between traditional KNN and PWK.

The exhaustive analysis of alternative weighting approaches for KNN is beyond the scope of our study. A succinct review of weight-based KNN proposals is given in [**?**], including attractive approaches for quantification like weighting examples in terms of their classification history [**?**], or accumulating the distances to k neighbors from each of the classes in order to assign the class with the smallest sum of distances [**?**]. Tan has also proposed further evolutions of his NWKNN,

such as the *DragPushing* strategy [**?**], in which the weights are iteratively refined taking into account the classification accuracy of previous iterations.

4. Empirical assessment

The required experiment methodology for quantification is relatively uncommon and has yet to be properly standardized. It differs significantly from traditional classification methodology because we have to evaluate performance over whole sets, rather than by means of 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. In this regard, we follow the global guidelines already established by Forman [**?**].

4.1. Experiment methodology

For performance measurement and comparison purposes we selected standard datasets with known positive prevalence for our experiments. We also adapted the stratified 10-fold cross-validation procedure, taking into account specific requirements for quantification, while preserving the original prevalence in all training iterations. In summary, once a model is 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%. Notice that this approach guarantees that all the examples are tested at least once, because when we test for 0% and 100% positive proportions, we are using all the negative and positive test examples of that fold, respectively. This setup also guarantees that the within-class distributions $P(\boldsymbol{x}|y)$ are maintained between training and test, as stated in Section **??**, due to the fact that resampling processes are uniformly randomized and stratified [**? ?**].

We presume that this variation in the testing conditions may be rather unnatural, requiring more appropriate collections of data. Changes in training and test conditions should be extracted directly from different snapshots of the same population, showing natural shifts in their distribution. However, for the time being we have not been able to find suitable collections of datasets offering these features.

4.1.1. Datasets

The main objective is to evaluate state-of-the-art quantification techniques, comparing them with simpler quantification models based on classical NN rules over different training distributions. In order to compare these models fairly, we selected a collection of datasets from the UCI Machine Learning Repository [**?**], taking problems with ordinal or continuous features with at the most three classes, and ranges from 100 to 2,500 examples. The summary of the 24 datasets meeting these criteria is presented in Table **??**.

Notice that the percentage of positive examples goes from 8% to 78%. This fact offers the possibility of evaluating the methods over significantly different training conditions. For datasets that originally have more than two classes, we followed a one-vs-all decomposition approach. We also extracted two different datasets from *acute*, which provides two alternative binary labels.

For datasets with positive class over 50%, *ctg.1* in this experiment, an alternative approach when using T50 method is to reverse the labels between both classes. We have tried both setups, but we have found no significant differences. Therefore, we decided to preserve the actual labeling, because we consider that it is crucial to perform the comparisons between systems under the same conditions.

Table 2: Summary of datasets

Dataset	Identifier	*Size*	*Attrs.*	*Pos.*	*Neg.*	*%pos.*
Acute Inflammations (urinary bladder)	acute.a	120	6	59	61	49%
Acute Inflammations (renal pelvis)	acute.b	120	6	50	70	42%
Balance Scale Weight & Distance Database (left)	balance.1	625	4	288	337	46%
Balance Scale Weight & Distance Database (balanced)	balance.2	625	4	49	576	8%
Balance Scale Weight & Distance Database (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%

4.1.2. Evaluation of quantification performance

Forman proposed the *Absolute Error* (AE) between actual and predicted positive prevalence as default loss function for quantification [**?**], which is simple, interpretable and directly applicable:

$$AE = |p' - p| = \frac{|P' - P|}{S} = \frac{|FP - FN|}{S}. \tag{10}$$

Moreover, as Esuli and Sebastiani [**?**] suggest, a function must simply deteriorate with $|FP - FN|$ in order to be considered an appropriate quantification metric, which is fulfilled by AE.

Kullback-Leibler Divergence (KLD) or normalized cross-entropy is also used for evaluating quantifiers in some contexts [**? ?**]. 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):

$$KLD = \frac{P}{S} \cdot \log\left(\frac{P}{P'}\right) + \frac{N}{S} \cdot \log\left(\frac{N}{N'}\right). \tag{11}$$

However, KLD is recommended when used to average across different test distributions, which is not the focus of our current experiments. This is because, predicting 7% for a test set with 10% positives is not equivalent to predicting 42% for a test set with 45% positives, although both cases yield 3% for AE. Averaging AE values in those cases is discouraged.

On the other hand, a clear advantage of using AE is that it has a real meaning for the practitioner. Moreover, KLD is not properly bounded, obtaining undesirable results, like infinity or indeterminate values, when the actual or estimated proportions are near 0% or 100%, needing further corrections to be applicable [**?**].

4.1.3. Algorithms and parameters

As one of the experiment baselines, we selected a dummy method that always predicts the distribution observed in training data, irrespective of the test distribution, which is denoted by BL. This allows us to verify the degree of improvement provided by other methods, that is, the point upon which they learn something significant. Although this baseline can be considered a *non-method*, it is able to highlight deficiencies in some algorithms. As we shall discuss later in Section **??**, there are no significant differences between BL and other methods.

We chose CC, AC, Max, X, T50 and MS as state-of-the-art quantifiers from Forman's proposals, considering CC as primary baseline. The underlying classifier for all these algorithms is a linear SVM from the *LibSVM* library [**?**].

The process of learning and threshold characterization, discussed in Section **??**, is common to all these models, reducing the total experiment time and guaranteeing an equivalent base SVM for them all. As regards the MS method, we found cases in which there exists no threshold providing a denominator greater than 1/4. Since Forman does not make any recommendation to overcome this problem, we decided to fix these missing values with the Max method, which provides the threshold with the greatest value for that difference.

The group of NN-based algorithms consists of KNN, PWK and PWK$^{\alpha}$. For the sake of simplicity, we always use the standard Euclidean distance and perform a grid-search procedure to select the best k value, as discussed in Section **??**. It is worth noting that we apply Forman's correction defined in (**??**) for all these NN algorithms. The main objective is to verify whether we can obtain competitive results with instance-based methods, while taking into account the formalisms already introduced by Forman. In contrast with threshold quantifiers, those based on NN rules do not calibrate any threshold after learning the classification model.

We use a grid-search procedure for parameter configuration, consisting of a 2×5 cross-validation [**?** **?**]. The loss function applied for discriminating the best values is the geometric mean (GM) of tpr and tnr (*true negative rate*, defined as TN/N), i.e., *sensitivity* and *specificity*. This measure is particularly useful when dealing with unbalanced problems in order to alleviate the bias towards the majority class during learning [**?**]. For those algorithms that use SVM as base learner, the search space for the regularizer parameter C is $\{0.01, 0.1, 1, 10, 100\}$. For NN-based quantifiers, the range for k parameter is $\{1, 3, 5, 7, 11, 15, 25, 35, 45\}$. In the case of PWK$^{\alpha}$, we also adjust parameter α over the integer range from 1 to 5. The grid-search for NN models is easily optimizable, because once the distance matrix

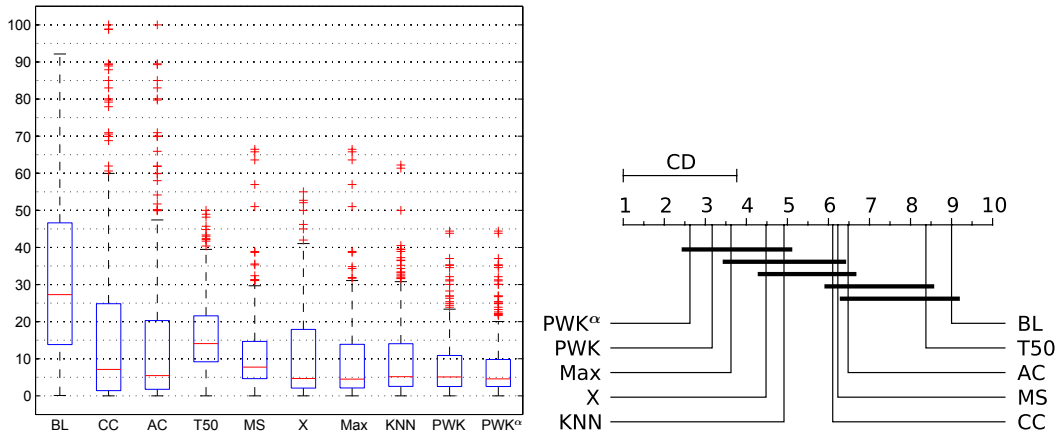
has been constructed and sorted, the computations with different values of k can be obtained almost straightforwardly.

The estimations of tpr and fpr for quantification corrections are obtained through a standard 10-fold cross-validation in all cases. Other alternatives like 50-fold CV or LOO are discarded because they are much more computationally expensive for SVM-based models. In the case of NN-based algorithms, the straightforward method for estimating these rates is by means of the distance matrix, applying a LOO procedure. However, we finally decided to use only one common estimation method for all competing algorithms for fairer comparisons.

4.2. Experimental results

In summary, we collected results from 24 datasets, applying a stratified 10-fold cross-validation for them all, preserving their original class distribution. After each training, we always assess the performance of the resulting model with 11 test sets generated from the remaining fold, varying the positive prevalence as described in Section **??**. We therefore performed 240 training processes and 2,640 tests for every system we evaluated. All quantification outputs are adjusted by means of Equation (**??**), except for BL and CC. This setup generates 264 cross-validated results for each algorithm, that is, 24 datasets $\times$ 11 test distributions. We obtained equivalent results with AE and KLD (see supplementary material). For the sake of readability this section only analyzes AE scores.

One of the key drawbacks that we encountered during the analysis of these experiments is the broad range of standpoints that can be adopted, in addition to the information overload with respect to traditional classification methodologies. Therefore, we consider that coherent and meaningful summaries of this information are crucial to understand, analyze and discuss the results properly.



(a) Boxplots of all AE results

(b) Nemenyi at 5% (CD = 2.7654)

Figure 1: Statistical comparisons among all systems under study

4.2.1. Overview analysis

The first approach that we followed is to represent the AE results for all 11 test conditions in all 24 datasets by means of a box-plot of each system under study. Thus, in Figure **??** we can observe the range of errors for every system. 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 that are not considered outliers, while the outliers are plotted individually with crosses. In this case, we consider as outliers any point greater than the third quartile plus 1.5 times the inter-quartile range. Note that we are not discarding the outliers for any computation, we are simply plotting them individually.

We distinguish four main groups in Figure **??** according to the learning procedure followed. The first one comprises only BL, covering a wide range of the spectrum of possible errors. This is probably due to the varying training conditions

of each dataset, given that this system always predicts the proportion observed during training. The second group, including CC and AC, shows strong discrepancies between actual and estimated prevalences of up to 100% in some outlier cases. These systems appear to be quite unstable under specific circumstances, which we shall analyze later. The third group includes T50, MS, X and Max, all of which are based on threshold selection policies (see Section **??**). However, as we shall also discuss later, the T50 method stands out as the worst approach in this group due to the evident upward shift of its box. The final group comprises NN-based algorithms: KNN, PWK and PWK$^{\alpha}$. The weighted versions of this last group offer the most stable results, with the third quartile below 15% in all cases. The weight-based versions present maximum outlier values below 45%.

Figure **??** provides other helpful insights regarding the algorithms under study. Taking into account the main elements of each box, we can observe that PWK and PWK$^{\alpha}$ stand out as the most compact systems in terms of the inter-quartile range. Both of them have their third quartile, their median and their first quartile around 10%, 5% and 2.5%, respectively. Note also that most of the models have a median AE of around 5%, meaning that 50% of the tests over those systems appear to yield competitive quantification predictions. Once again, however, the major difference is highlighted by the upper tails of the boxes, including the third quartile, the upper whisker and the outliers. From the shape and position of the boxes, KNN, Max, X and MS also appear to be noteworthy.

4.2.2. Friedman-Nemenyi statistical test

Following Demšar's proposal [**?**], a two-step statistical test procedure was carried out. The first step consists of a Friedman test of the null hypothesis that all approaches perform equally. When this hypothesis is rejected, a Nemenyi post-

hoc test is then conducted to compare the methods in a pairwise way. Both tests are based on the average of the ranks. The comparison includes 10 algorithms over 24 datasets or domains, tested over 11 different prevalences, resulting in 264 test cases per algorithm. 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.

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. As an alternative, averaging the AE results over the 11 prevalences that are tested for each dataset suffers the problem of how to handle large outliers and the inconsistency of averaging AE values from different test prevalences (see Section **??**), so we do not average AE results. 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 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 algorithms are compared over 24 domains, regardless of the number of prevalences that are tested for each of them.

Friedman's null hypothesis is rejected at the 5% significance level and the CD for the Nemenyi test with 24 datasets and 10 algorithms is 2.7654. The overall results of the Nemenyi test are shown in Figure **??**, in which each system is rep-

resented by a thin line, linked to its name on one side and to its average rank on the other. The thick horizontal segments connect models that are not significantly different at a confidence level of 5%. Therefore, this plot suggests that PWK$^{\alpha}$ and PWK are the models that perform best in this experiment in terms of AE loss 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 KNN, Max or X. We can only appreciate that PWK$^{\alpha}$ and PWK are significantly better than CC, AC, MS, T50 and BL; Max is still connected with CC and MS, while X and KNN are also connected with AC. It is worth noting that neither AC nor T50 show clear differences with respect to BL, suggesting a lack of consistency in the results provided by the former systems.

4.2.3. Pair-wise comparisons with PWK$^{\alpha}$

Since PWK$^{\alpha}$ appears to be the algorithm that yields the lowest values for AE in general, obtaining the best average rank in the Nemenyi test, from now on we shall use it as a *pivot* model so as to compare it to all the other systems under study. Thus, in Figure **??** we present pair-wise comparisons of each system with respect to PWK$^{\alpha}$. Each point represents the cross-validated AE values of the compared system on the y-axis and of PWK$^{\alpha}$ on the x-axis, for the same dataset and test prevalence. The red diagonal depicts the boundary where both systems perform equally. Therefore, when the points are located above the diagonal, PWK$^{\alpha}$ yields a lower AE value, and vice-versa. It should be noted that as we are using PWK$^{\alpha}$ as a pivot model for all comparisons, there is always the same number of points at each value of the x-axis. Thus, the movement of these points along the y-axis, among all the comparisons, provides visual evidence of which systems are more competitive with respect to PWK$^{\alpha}$.

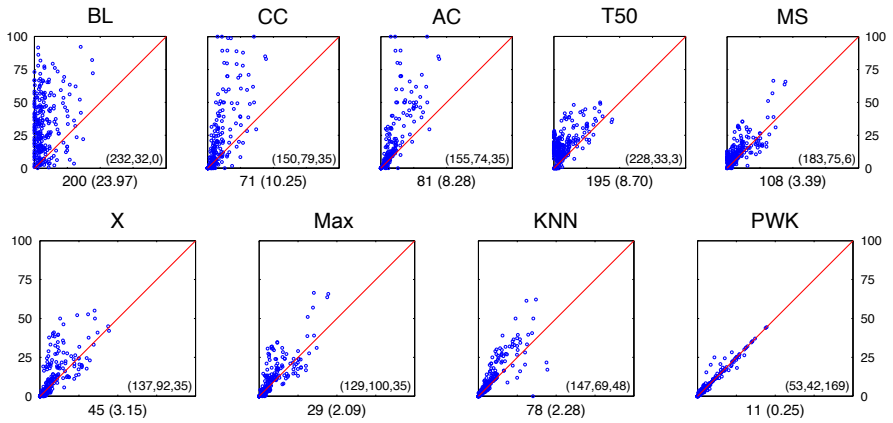


Figure 2: Pair-wise comparisons of each algorithm with respect to PWK$^{\alpha}$, in terms of AE. The results over all test prevalences are aggregated into a single plot, where each one represents 264 cross-validated results. The inner triplet shows the number of wins, losses and ties of PWK$^{\alpha}$ versus the compared system. The numbers below each plot reveal the difference between wins and losses (DWL), and within parentheses the mean of the differences between AE values ($MDAE$).

We also include several metrics within each plot. The numbers below each plot reveal the difference between wins and losses (DWL), and within parentheses the mean of the differences between AE values of both algorithms ($MDAE$). Positive values of DWL and $MDAE$ indicate better results for PWK$^{\alpha}$, though they are only conceived for clarification purposes during visual interpretation. The aim of the DWL metric is to show the degree of *competitiveness* between two systems, values close to zero indicating that they are less differentiable, in terms of wins and losses, than systems with higher values. Moreover, $MDAE$ can also be used as a measure of the *symmetry* of both models. Note that being symmetric in this context does not refer to similarity of results, but to compensation of errors. This means that systems with an $MDAE$ value close to zero are less differentiable

in terms of differences of errors.

From the shape drawn by the plots in Figure **??**, we can observe some interesting interactions between models, always with respect to PWK$^{\alpha}$. As expected, the comparison with PWK, for example, shows a clear connection between both systems; all points present a strong trend towards the diagonal. Moreover, DWL indicates that PWK is the most competitive approach, while $MDAE$ shows that the average difference of errors is only 0.26, being highly symmetric.

The points in KNN's plot are not so close to the diagonal, being mainly situated slightly upwards. This behavior suggests that KNN is less competitive (78) and less symmetric (2.28) than PWK. Nevertheless, in general, NN-based algorithms present the best performance.

Although Max, X, MS and T50 are all based on threshold selection policies, the DWL and $MDAE$ values differ noticeably among them. As already observed in Figure **??**, Max seems to outperform the others, both in competitiveness (29) and symmetry (2.09), while T50 stands out as the less competitive approach among these quantification models.

The distribution of errors in Figure **??** for BL, CC and AC is once again evidenced in Figure **??**. The presence of outliers in CC and AC is emphasized through high values of $MDAE$, combined with intermediate values of DWL. As regards BL, this algorithm shows the worst values in Figure **??** for competitiveness (200) and symmetry (23.97). This poor behavior can be also observed in Figure **??**.

4.2.4. Analysis of results by test prevalence

Although Figures **??**, **??** and **??** provide interesting evidence, they fail to show other important issues. For instance, we cannot properly analyze the performance

Figure 3: Pair-wise comparisons of each algorithm with PWK$^{\alpha}$, in terms of AE. The results over different test prevalences are plotted individually (by rows), where each plot represents the cross-validated results over 24 datasets. See caption of Figure **??** for further details about the metrics placed below each graph.

of each system with respect to specific prevalences. Furthermore, they only offer a general overview of the limits and distribution of AE values, without taking into account the magnitude of the error with respect to the actual test proportions.

Figure **??** follows the same guidelines as those introduced for Figure **??**; however, in this case we split each plot into eleven subplots, placed by rows. Each of these subplots represents the comparative results of a particular system with respect to PWK$^{\alpha}$ for a specific test prevalence. This decision is again supported by the fact that PWK$^{\alpha}$ appears to be the system that performs best in terms of AE metric. Moreover, despite the overload of information available, this summarization allows us to represent the values of all systems with fewer plots, to simplify the comparison of every system with respect to the best of our proposed models, and to visualize the degree of improvement among systems, all at the same time. The axes of those comparisons where DWL has negatives values are highlighted in red, while ties in DWL values are visualized by means of a gray axis. Notice that there are also cases where values of DWL and $MDAE$ have a different sign.

The average training prevalence among all datasets is 34.22%; hence, test prevalences at 30% and 40% are the closest to the original training distribution for the average case. This can be observed in Figure **??** through the BL results, which always predict the proportion observed during training. As expected, when the test distribution resembles that of the training, it yields competitive results, although the performance is significantly degraded to the worst case when the test proportions are different from those observed during training. Taking the plots of BL as reference, we observe that the behavior of PWK$^{\alpha}$ seems to be heading in the right direction in terms of both DWL and $MDAE$. Notice that the $MDAE$ values in this column rise and fall in keeping with changes in test prevalence.

The CC method performs well over low prevalence conditions, obtaining the best DWL results for 10% and 20%. However, it apparently tends to increasingly underestimate for higher proportions of positives, as evidenced by the $MDAE$ values. This supports the conclusions regarding uncalibrated quantifiers drawn by Forman [**?**]. On the other hand, we expected a more decisive improvement of AC over CC results in general. Actually, when the positive class becomes the majority class, for test prevalences greater than 50%, the AC correction produces an observable improvement in terms of DWL, and especially for $MDAE$. From a general point of view, however, the results that we have obtained with this experiment show that simply adjusting SVM outputs may not be sufficient, providing even worse results than traditional uncalibrated classifiers, mainly when testing low prevalence scenarios. This fact is mostly highlighted by the $MDAE$ results of CC and AC over prevalences below 50%.

The most promising results among state-of-the-art quantifiers are obtained by Max and X, although the former provides more competitive results for the average case. The greatest differences between $MDAE$ results are observed for test prevalences below 50%, where Max yields lower values. These differences are softened in favor of X for higher prevalences. We suspect that these threshold selection policies could entail an intrinsic compensation of the underlying classification bias shown by CC, which tends to overestimate the majority class. This intuition is supported by the observation that they still perform worse than CC for low test prevalences, as they may tend to overestimate the minority class.

Additionally, both provide better DWL and $MDAE$ results than CC or AC for prevalences higher than 40%. T50 presents the worst results of this family of algorithms, showing surprisingly good performance in test prevalence at 0%. Con-

versely, MS shows an intermediate behavior, performing appealingly in $MDAE$ but discouragingly in DWL, obtaining competitive results when the test prevalence is 100%. This good performance for extreme test prevalences could be due to the fact that corrected values are clipped into the feasible range after applying Equation (**??**), as described in Section **??**. Therefore, this kind of behavior is not representative, unless it is reinforced with more stable results in near test prevalences. Moreover, Figures **??**, **??** and **??** highlight cases where Max and MS share some results. As described in Section **??**, this is due to missing values in the latter method, which happens to be linked with outlier cases in Max. This suggests a possible connection between the complexity of these cases and their lack of thresholds where the denominator in (**??**) is big enough, being less robust with respect to estimation errors in tpr and fpr.

At first glance, KNN yields interesting results. Excluding CC, it improves DWL below 30% with respect to SVM-based models. Actually, both CC and KNN are the most competitive models over lowest prevalences, probably because they tend to misclassify the minority class, so that they are biased to overestimate the majority class. Thus, when the minority class shrinks, the quantification error also decreases. Notwithstanding, KNN behaves more consistently, providing stable $MDAE$ results over higher prevalences. Comparing KNN with AC, we also observe that, in general, KNN also appears to be more robust in terms of $MDAE$. This suggests that KNN produces AE results with lower variance and less outliers than CC and AC, as previously observed in Figures **??** and **??**.

As already mentioned, the red (black) color in Figure **??** represent cases where the compared system yields better (worse) DWL than PWK$^{\alpha}$, while ties are depicted in gray. Hence, these plots reinforce the conclusion that PWK$^{\alpha}$ is usu-

ally the algorithm that performs best, with a noticeable dominance in terms of $MDAE$. Apparently, adding relatively simple weights offers an appreciable improvement, which is clearly observable when compared with traditional KNN. With the exception of PWK, there exists only one case where both DWL and $MDAE$ produce negative values in Figure **??**, corresponding to CC at a test prevalence of 10%. This is probably caused by the fact that CC is supposed to yield exact results over a specific prevalence, identified as p^* in Forman's theorem. Therefore, this result is not relevant in terms of global behavior. Furthermore, except for PWK over prevalences higher than 50%, the values for the $MDAE$ metric are positive in all cases. This implies that AE values provided by PWK$^\alpha$ and PWK are generally lower and have less variance than those of all the other systems.

The resemblance between PWK$^\alpha$ and PWK is once again emphasized through low values of $MDAE$ over all test prevalences. However, previous figures failed to shed light on a very important issue. Observing the last column in Figure **??**, it appears that PWK$^\alpha$ is more conservative and robust over lower prevalences, while PWK is more competitive over higher ones. These differences are softened towards intermediate prevalences. This behavior is supported by the fact that, although PWK$^\alpha$ and PWK use weights based on equivalent formulations, the parameter α in PWK$^\alpha$ tends to weaken the influence of these weights when it increases. Moreover, as already stated in Section **??**, since these weights are designed to compensate the bias towards the majority class, when the parameter α grows, the recall decreases, and vice-versa.

5. Conclusions

This paper establishes a new approach for dealing with prevalence estimation in binary problems. The main objective is to study the behavior of NN methods in the context of quantification. We seek for an instance-based approach able to provide competitive performance while balancing simplicity and effectiveness. Although other potential alternatives exist, we have limited our experiments to those settings conforming to this scope.

After a brief discussion of the general background related to quantification, as established by Forman in [**?**], we describe our main proposals based on traditional NN rules. These NN-based algorithms include the well-known KNN and two simple weighting strategies, identified as PWK and PWK$^{\alpha}$.

We have found that, in general, weighted NN-based algorithms offer the best performance. The conclusions drawn from the Nemenyi test summary presented in Figure **??** suggest that PWK and PWK$^{\alpha}$ stand out as the best approaches, without statistical differences between the two, but offering clear statistical differences with respect to less robust models. Thus, these experiments do not provide any discriminative indicator regarding which of these two algorithms is more recommendable for real-world applications. The final decision should be taken in terms of the specific needs of the problem, the constraints of the environment, or the complexity of the data, among other factors. Notwithstanding, taking into account the observations discussed in Section **??**, it appears that PWK could be more appropriate when the minority class is much more relevant, while PWK$^{\alpha}$ seems to behave more conservatively with respect to the majority class. Furthermore, PWK is simpler, its weights are more easily interpretable and it only requires calibrating the number of neighbors.

Possible future directions for NN-based quantification could involve the selection of parameters through grid-search procedures, optimizing metrics with respect to equivalent rules as those applied for Max, X or T50, or even using these rules to calibrate the weights of each class during learning. Finally, appropriate collections of data, extracted directly from different snapshots of the same populations and showing natural shifts in their distributions, are required in order to further analyze the quantification problem from a real-world perspective.

Acknowledgment

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

References

[] A. Frank, A. Asuncion, UCI machine learning repository, University of California, Irvine, 2010. http://archive.ics.uci.edu/ml/.

[] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1967) 21–27.

[] W. Hardle, Applied nonparametric regression, Cambridge University Press, Cambridge, 1992.

[] K. Hechenbichler, K. Schliep, Weighted k-nearest-neighbor techniques and ordinal classification, Technical Report 399 (SFB 386), Ludwig-Maximilians University, Munich, 2004.

[] M. Wong, T. Lane, A kth nearest neighbour clustering procedure, Journal of the Royal Statistical Society, Series B, Methodological (1983) 362–368.

[] P. Broos, K. Branting, Compositional instance-based learning, in: Proceedings of the 12th AAAI National Conference, volume 1, pp. 651–656.

[] M. Zhang, Z. Zhou, ML-KNN: A lazy learning approach to multi-label learning, Pattern Recognition 40 (2007) 2038–2048.

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

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

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

[] 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.

[] G. Forman, Counting positives accurately despite inaccurate classification, in: Proceedings of the 16th ECML, Springer, 2005, pp. 564–575.

[] G. Forman, Quantifying trends accurately despite classifier error and class imbalance, in: Proceedings of the 12th SIGKDD, ACM, 2006, pp. 157–166.

[] D. Hand, V. Vinciotti, Choosing k for two-class nearest neighbour classifiers with unbalanced classes, Pattern Recognition Letters 24 (2003) 1555–1562.

[] C. Enas Sung, G. Gregory, Choice of the smoothing parameter and efficiency of k-nearest neighbor classification, Computers & Mathematics with Applications 12 (1986) 235–244.

[] P. Kang, S. Cho, Locally linear reconstruction for instance-based learning, Pattern Recognition 41 (2008) 3507–3518.

[] C. Domeniconi, J. Peng, D. Gunopulos, Locally adaptive metric nearest-neighbor classification, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 24 (2002) 1281–1285.

[] S. Tan, Neighbor-weighted k-nearest neighbor for unbalanced text corpus, Expert Systems with Applications 28 (2005) 667 – 671.

[] S. Cost, S. Salzberg, A weighted nearest neighbor algorithm for learning with symbolic features, Machine Learning 10 (1993) 57–78.

[] K. Hattori, M. Takahashi, A new nearest-neighbor rule in the pattern classification problem, Pattern recognition 32 (1999) 425–432.

[] S. Tan, An effective refinement strategy for KNN text classifier, Expert Systems with Applications 30 (2006) 290–298.

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

[] E. Alpaydm, Combined 5×2 cv F test for comparing supervised classification learning algorithms, Neural computation 11 (1999) 1885–1892.

[] T. G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Computation 10 (1998) 1895–1923.

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

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