Information Sciences 218 (2013) 146–164

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Class distribution estimation based on the Hellinger distance

Víctor González-Castro, Rocío Alaiz-Rodríguez *, Enrique Alegre

Dpto. de Ingeniería Eléctrica y de Sistemas y Automática, University of León, Campus de Vegazana s/n, 24071 León, Spain

ARTICLE INFO

Article history:
Received 6 May 2011
Received in revised form 10 May 2012
Accepted 31 May 2012
Available online 15 June 2012

Keywords:
Class prior probability estimation
Quantification
Class distribution shift
Hellinger distance

ABSTRACT

Class distribution estimation (quantification) plays an important role in many practical classification problems. Firstly, it is important in order to adapt the classifier to the operational conditions when they differ from those assumed in learning. Additionally, there are some real domains where the quantification task is itself valuable due to the high variability of the class prior probabilities. Our novel quantification approach for two-class problems is based on distributional divergence measures. The mismatch between the test data distribution and validation distributions generated in a fully controlled way is measured by the Hellinger distance in order to estimate the prior probability that minimizes this divergence. Experimental results on several binary classification problems show the benefits of this approach when compared to such approaches as counting the predicted class labels and other methods based on the classifier confusion matrix or on posterior probability estimations. We also illustrate these techniques as well as their robustness against the base classifier performance (a neural network) with a boar semen quality control setting. Empirical results show that the quantification can be conducted with a mean absolute error lower than 0.008, which seems very promising in this field.

1. Introduction

Supervised learning aims at computing a classifier with good prediction ability on future unseen data. A set of labeled instances is required in order to train the classifier. Once the classifier is designed, it is assumed that it is applied as-is to new data in order to predict the class to which each individual belongs to.

Most work assumes that training and future (test) data follow the same, although unknown, distribution [1]. In particular, class prior probabilities estimated from the training data set are considered to reflect truly the class distribution of the operational environment. However, time or space class stationarity cannot be assumed in many practical fields [2,3]. For example, if a system for word sense disambiguation is trained using words from a certain domain (i.e. sports news), but it is then used with instances from a different domain (i.e. political news), where the sense priors are different, the classifier will be affected [4]. Remote sensing applications also suffer from this problem since a dataset collected in a given season from a region with different terrains (industrial, hay, wheat, corn, grass, etc.) is usually employed to train the classifier. However, when that classifier is deployed, mismatches in terrain distribution may appear just because of seasonal or location changes [5]. Another illustrative example is direct mail marketing as the target proportion or customer profile may vary from one area to another.

It is well known that a mismatch between the actual class prior probabilities and those for which the classifier has been optimized, leads to suboptimal solutions [1]. Whenever there is such a change, some authors rely on an eventual perfect knowledge of the new conditions by the end user [6], but when this is not possible, estimating this new class proportion

* Corresponding author.
E-mail addresses: victor.gonzalez@unileon.es (V. González-Castro), rocio.alaiz@unileon.es (R. Alaiz-Rodríguez), enrique.alegre@unileon.es (E. Alegre).

0020-0255/$ - see front matter © 2012 Published by Elsevier Inc.
http://dx.doi.org/10.1016/j.ins.2012.05.028

is important in adapting the classifier to the new context [7–9]. Adapting the classifier to the new operating conditions, based on an unlabeled data set, is a problem that has received a lot of attention lately from several perspectives [7–14], with the ultimate goal of improving the individual classification performance. Some techniques include those described in [8,11,15–17]. Wang et al. [18] proposed a method video annotation (which is formulated as a classification task) when there is a large variation in the training data (i.e. the assumed model may change). This method uses an iterative process to update class densities and posterior probabilities, similar to what Saerens et al. [8] did, which based on Bayes rule.

In other applications where the class proportions are subject to high variability, their estimation is itself valuable, in particular when the classes are imbalanced [19]. For instance, artificial insemination techniques in the veterinarian field should guarantee that semen samples are optimal for fertilization. There is a direct relationship between sperm fertility and the state of the acrosome: a sample containing a high percentage of spermatozoa with a damaged acrosome will not be useful for fertilizing purposes [20]. In this case, the class prior probabilities estimated from the labeled training data cannot be considered representative of future samples since they are subject to variation due to factors like the animal/farm variability, or the manipulation and conservation conditions. Quantifying the proportion of damaged cells is traditionally carried out manually, using stains, which makes this process time-consuming, costly and, what is more important, not objective. In this field, then, the aim is to estimate the proportion of damaged cells with no concerns about the individual classification of each one [21].

To the best of our knowledge, only a few works address directly the problem of estimating the class distribution (also known as quantification) in real domains. Quantification has been applied to such domains as quality control [22–24], news categorization [25,26], analysis of technical-support call logs [27] and word text disambiguation [4].

To sum up, estimating the class prior probabilities of an unlabeled dataset plays an important role in supervised learning in order to be able to detect changes in classifier performance due to shifts in class prior probabilities (assuming that class conditional densities are fixed) and in order to adapt the classifier to the new operating conditions whenever it is possible. It also plays an important role in applications where the class distribution shows high variability and its estimation has practical interest.

The quantification techniques proposed in the literature are either based on the classifier confusion matrix [7,4,25,26,9] or on the posterior probability estimates provided by the classifier [4,23,22,24,28]. Forman has also explored a method, Mixture Model, based on the estimation of the class conditional probability densities [26], but when it was evaluated on text classification data sets, found it was outperformed by simple methods that rely on the confusion matrix. There is also a preliminary work based on assessing mismatches between data distributions [24].

Our proposal to estimate the class distributions is based on measuring distributional divergences. We focus on problems where the class conditional densities are assumed to be fixed, but class prior probabilities may vary. It is well known that a shift in class prior probabilities between the training and test sets makes the data distributions, as well as the classifier output distribution, change. Basically, our approach assesses the similarity between distributions, comparing the test data distribution with validation data distributions generated in a fully controlled way from the training data set. Finding the class distribution (a simple for-loop can be used in binary problems as in this work) that achieves the maximum similarity, provides the estimated value. A distributional divergence metric, the Hellinger Distance (HD) [29], may be applied at different stages of the classification process: (i) between data distributions themselves for each input feature (referred to as HDx) and (ii) between the classifier output distributions (referred to as HDy). The HDy proposal is similar to the Mixture Model of [30], but we use the HD to measure the goodness of the fit instead of using the PP-Area metric developed in [30] to compare two cumulative distribution functions.

The goal of this paper is: (a) to explore an information theoretic approach to quantify automatically the class distribution of an unlabeled dataset, (b) to compare it with previously proposed approaches (in 15 applications from the UCI Machine Learning repository with a neural based classifier, Naive Bayes and logistic regression) and (c) to evaluate these quantification methods and check whether or not reliable estimates can be achieved for a real specific application of boar semen analysis. Note that, unlike most prior work that focuses on text classification tasks, here we apply our algorithms to a variety of domains collected in the UCI repository and to a real computer vision application.

The rest of this paper is organized as follows: Section 2 briefly describes previous proposed approaches to this problem and Section 3 presents the theoretical approach and algorithms of the estimation method based on the Hellinger distance proposed in this paper. Empirical evaluation methodology is presented in Section 4, the experimental results are shown in Section 5 and finally, Section 6 summarizes the main conclusions.

2. Quantification: the problem of class distribution estimation

Consider a classification problem with a labeled training data set $S_t = \{(\mathbf{x}^k, d^k), k = 1, \ldots, K\}$ where $\mathbf{x}^k$ is the feature vector of the kth element and d^k is its class label, which takes its value in $\Omega = \{d_0, d_1, \ldots, d_{M-1}\}$.

Let us consider that all the samples $\mathbf{x}^k \in S_t$ have been independently recorded according to the class probability density function $p(\mathbf{x}|d_i)$ and the *a priori* probability of the class d_i in the training data set S_t is denoted by $P_t(d_i)$. Note that hereafter the subscript t will be used for estimates based on the training set S_t.

Now, consider we have an unlabeled test data set $U = \{(\mathbf{x}^l), l = 1, \ldots, N\}$ from which there is no specific interest in knowing the class label of each instance, but the class distribution $P(d_i)$ of the test set U needs to be estimated.

Let us also consider a classification model $\mathbf{f_w}$ whose parameters $\mathbf{w}$ have been adjusted using examples from the training set S_t. This classifier makes decisions in two steps: it first computes a soft output $\hat{\mathbf{y}}^l = \mathbf{f_w}(\mathbf{x}^l)$ and then makes a hard decision $\hat{d}^l \in \Omega$ based on it. It is well known that if the classifier is trained minimizing an appropriate cost function, the soft outputs $\hat{y}_i^l$ will provide an estimation of the *a posteriori* probability of the observation $\mathbf{x}^l$ belonging to class d_i, denoted by $\widehat{P}(d_i|\mathbf{x}^l)$ [31].

The naïve approach to estimate the actual class distribution is based on just counting the labels $\hat{d}^l$ assigned by the classifier. This approach has been referred to as Classify and Count (CC) in [26]. The estimates made by this method will not be reliable since: (i) the classifier performance will drop if there is a difference between $P(d_i)$ and $P_t(d_i)$, and (ii) there is no guarantee that the errors for each class will cancel each other out.

Next, some quantification techniques based on the classifier confusion matrix and relying on the posterior probability estimates provided by the classifier will be described briefly in Sections 2.1 and 2.2, respectively.

2.1. Quantification based on the confusion matrix

The confusion matrix summarizes the performance of a classifier. It is an observation of the number of elements classified as belonging to the class i when they actually belong to the class j. An example of a confusion matrix for a binary problem is shown in Table 1 where class d_1 is also referred to as the positive class and class d_0 as the negative class.

The count of positives P' assigned by the classifier will include both the true and false positives ($P' = TP + FP$), while the number of predicted negatives N' will be the sum of the true and the false negatives ($N' = TN + FN$). Similarly, the number of real positive examples is $P = TP + FN$ while the number of actual negatives is $N = FP + TN$. By using these quantities, the following rates can be computed:

- True Positive rate: $tpr = \widehat{P}(\hat{d}_1|d_1) = TP/P$
- False Positive rate: $fpr = \widehat{P}(\hat{d}_1|d_0) = FP/N$
- False Negative rate: $fnr = \widehat{P}(\hat{d}_0|d_1) = FN/P$
- True Negative rate: $tnr = \widehat{P}(\hat{d}_0|d_0) = TN/N$

As has been derived in [30], the probability that a classifier makes a positive prediction in a binary classification problem is:

$$\begin{aligned}\widehat{P}(\hat{d}_1) &= \widehat{P}(\hat{d}_1|d_1)\cdot\widehat{P}(d_1) + \widehat{P}(\hat{d}_1|d_0)\cdot\widehat{P}(d_0)\\ &= \widehat{P}(\hat{d}_1|d_1)\cdot\widehat{P}(d_1) + \widehat{P}(\hat{d}_1|d_0)\cdot(1-\widehat{P}(d_1))\\ &= tpr\cdot\widehat{P}(d_1) + fpr\cdot(1-\widehat{P}(d_1))\\ &= tpr\cdot\widehat{P}(d_1) + fpr - fpr\cdot\widehat{P}(d_1)\\ &= fpr + \widehat{P}(d_1)\cdot(tpr - fpr)\end{aligned}$$

which leads to the estimation of the *a priori* probability of the class d_1 as:

$$\widehat{P}(d_1) = \frac{\widehat{P}(\hat{d}_1) - fpr}{tpr - fpr} \tag{1}$$

where

$$\widehat{P}(\hat{d}_1) = \frac{P'}{P' + N'} \tag{2}$$

As was mentioned in Section 1, the quantification process assumes that the within class densities $p(\mathbf{x}|d_i)$ do not change from the training to the new data sets [8] and, therefore, there is no fundamental variation in the *fpr* and *tpr* between the distributions for the training set and test sets. The confusion matrix can be estimated by techniques such as stratified k-fold cross-validation where the value of k is recommended to be as high as possible, as suggested in [26].

This method has been referred to as *Adjusted Count* (AC) in [26] and it could be summarized in a binary problem as follows: Firstly, the classifier is trained and its performance (*fpr* and *tpr*) estimated via k-fold cross-validation. Then, when the

Table 1
Confusion matrix for a binary classification problem.

	Prediction	
	$\hat{d}_1$	$\hat{d}_0$
True class		
d_1	TP	FN
d_0	FP	TN

classifier is applied to a new unlabeled set, the probability of predicted positive elements $\widehat{P}(\hat{d}_1)$ is computed according to (2) and finally, the estimation of the true percentage of positives is computed as (1).

If the problem we are dealing with has M classes, a system of M linear equations with respect to $P(\hat{d}_j)$ should be solved in order to estimate all the new class prior probabilities, as can be seen in (3).

$$\widehat{P}(\hat{d}_i) = \sum_{j=0}^{M-1} \widehat{P}_t(\hat{d}_i|d_j)\widehat{P}(d_j), \quad i = 0, 1, \ldots, M-1 \tag{3}$$

where $\widehat{P}(d_j)$ is the estimation of the *a priori* probability of the class j and $\widehat{P}(\hat{d}_i)$ is the observed class probability by looking at the classifier labels $\hat{d}$.

The solution of (1) or (3), however, can be inconsistent with the basic probability laws (i.e, values outside the interval $[0,1]$). In a binary problem, Forman suggests [26] clipping the negative values to zero and fixing the probability of the other class to one. In a multiclass problem, however, there is no straightforward solution.

Based on this Adjusted Count (AC) method, Forman also proposes the *Median Sweep* (MS) method [25]. Briefly, it can be described as follows. First of all, several confusion matrices are computed for different classification thresholds. Next, the method AC is applied for each confusion matrix and finally, the class distribution estimate is computed as the median of the estimates derived from each confusion matrix.

2.2. Quantification based on the posterior probability estimates

Based on a classification model whose outputs provide estimates of posterior probabilities, an algorithm is proposed in [22] to estimate the class distribution of a new dataset for a general multi-class problem. It is inspired by an iterative procedure based on the EM algorithm proposed by Saerens et al. [8] that adjusts the classifier outputs for the new deployment conditions without re-training the classifier, as long as classifier outputs provide class posterior probability estimates. This process indirectly computes the new class prior probabilities, which is the goal in this work.

Consider that the outputs $\hat{\mathbf{y}}^k$ generated by the classifier for the set U are an approximation of the *a posteriori* probabilities of the classes, while the class frequencies in the training set are an estimation of the *a priori* probabilities. Thus, the prior and the posterior probability estimates are initialized by:

$$\widehat{P}^{(0)}(d_i|\mathbf{x}_k) = \hat{y}_i^k \tag{4}$$

$$\widehat{P}^{(0)}(d_i) = \frac{|S_t^i|}{|K|} \tag{5}$$

where K is the total number of training examples and $|S_t^i|$ is the cardinality of the set of training examples from class-i. Consider $\widehat{P}^{(r)}(d_i)$ the estimate of the new *a priori* probabilities and $\widehat{P}^{(r)}(d_i|\mathbf{x}^k)$ the new *a posteriori* probabilities at the rth iteration of the algorithm. These estimates are based on Bayes' decision theory and are given by (6) and (7) respectively.

Regarding the class prior probability, it is well-known that the prior probability of class d_i for a dataset U with N unseen examples can be estimated as an average of the posterior probabilities provided for each example in the set. Therefore, its estimate at the rth iteration is given by

$$\widehat{P}^{(r)}(d_i) = \frac{1}{N}\sum_{l=1}^{N} \widehat{P}^{(r-1)}(d_i|\mathbf{x}^k) \tag{6}$$

Once a new estimate of the class prior probabilities $\widehat{P}^{(r)}(d_i)$ is available, the class posterior probabilities are adjusted accordingly by means of a simple transformation [32] based on the Bayes' theorem:

$$\widehat{P}^{(r)}(d_i|\mathbf{x}^k) = \frac{\frac{\widehat{P}^{(r)}(d_i)}{\widehat{P}^{(0)}(d_i)}\widehat{P}^{(0)}(d_i|\mathbf{x}^k)}{\sum_{j=0}^{M-1}\frac{\widehat{P}^{(r)}(d_j)}{\widehat{P}^{(0)}(d_j)}\widehat{P}^{(0)}(d_j|\mathbf{x}^k)}. \tag{7}$$

where the denominator makes the adapted probabilities for the different classes sum to one.

This procedure is repeated a certain number of iterations, or until the difference between two successive estimates is lower than a certain threshold. It is called the *Posterior Probability* method (PP) in [22,24].

3. Quantification based on the Hellinger distance

In this section we present two quantification techniques (HDx and HDy) based on assessing the Hellinger Distance (HD) between the test data distribution and a validation data distribution. The HDx approach works directly with the feature vectors $\mathbf{x}$ and therefore it does not require any classification model. The proposal HDy works with the outputs $\hat{\mathbf{y}}$ that a classifier (calibrated with instances from the training set S_t) generates for the samples $\mathbf{x}$. Note that the data HDx works with, has a

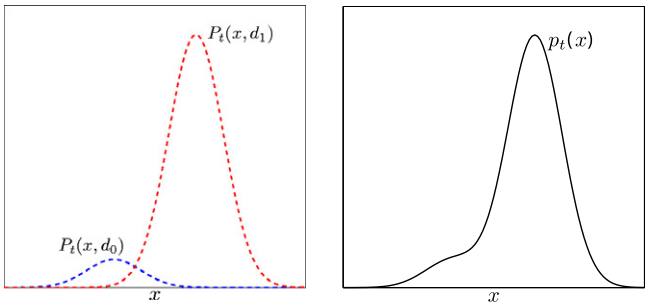

Fig. 1. Training data. Joint probabilities $P_t(x,d_0)$ and $P_t(x,d_1)$ (left) and unconditional density $p_t(x)$ (right) for prior class probabilities $(P_t(d_0), P_t(d_1))$ equal to $(0.1, 0.9)$.

dimensionality equal to n_f (the number of features) regardless of the number of classes. HDy, however, deals with one dimensional vectors if the classification problem is binary.

The remainder of this section is organized as follows: next, the effect of a prior probability shift on data distributions is illustrated and analyzed. Then, Section 3.2 presents the Hellinger distance metric. Finally, the methods HDx and HDy are described in Sections 3.3 and 3.4, respectively.

3.1. Prior probability shift

The prior probability shift has been described as a particular case of dataset shift [33], where the within-class conditional densities $p(\mathbf{x}|d_i)$ do not change between the training and test scenarios, but the class prior probabilities shift after the classification model is generated $(P_t(d_i) \neq P(d_i))$.[1] When this happens, the joint probabilities also vary $(P_t(\mathbf{x}, d_i) \neq P(\mathbf{x}, d_i))$ as does the unconditional density $(p_t(\mathbf{x}) \neq p(\mathbf{x}))$. As we have already pointed out, this is the type of problem we want to tackle.

For instance, assume we have a binary classification problem where each class is defined by a univariate Gaussian distribution. Fig. 1 shows the joint probabilities $P_t(x,d_0)$ and $P_t(x,d_1)$ for the training dataset with class priors $(P_t(d_0), P_t(d_1))$ equal to $(0.1, 0.9)$ and the data density $p_t(x)$. Now, in the test data the class conditional densities remain unchanged, but the test class prior probabilities $(P(d_0), P(d_1)) = (0.6, 0.4)$ have changed $(P(d_0) \neq P_t(d_0), P(d_1) \neq P_t(d_1))$ from the training phase. This shift in class proportions makes the data distribution $p(x)$ significantly different from the training data distribution $p_t(x)$ as it is illustrated in Figs. 1 and 2.

When we deal with a real practical problem, we know $p(\mathbf{x})$. Additionally, from the training dataset we can generate validation datasets V with given prior probabilities and compute the differences between these data distributions (validation data $p_v(\mathbf{x})$ and test data $p(\mathbf{x})$) in order to find the validation distribution that is the most similar to the test data distribution. Therefore, this process allows us to estimate the new class proportions, which are the same as the proportions of the validation set that minimizes that difference. Next, we present the distance metric we suggest in order to assess the similarity between the test and validation distributions.

3.2. Hellinger distance

The Kullback–Leibler divergence as well as the χ^2 measure and the Hellinger Distance (HD) are particular cases of the family of f-Divergences [29]. When it comes to measuring the divergence between distributions, the χ^2 measure and the widely used KL divergence are both asymmetric, and not strictly distance metrics, which makes the Hellinger distance very appealing for our purpose.

Recently, HD has been receiving attention by the machine learning community in order to detect failures in classifier performance due to shifts in data distributions. In particular, Cieslak and Chawla [34] have shown that the measuring the HD is very effective in detecting breakpoints in classifier performance due to shifts in class prior probabilities. Here, we address the problem of class distribution estimation following a HD-based approach.

The Hellinger distance between two probability density functions $q(\mathbf{x})$ and $p(\mathbf{x})$ can be expressed as

$$HD(q,p) = \sqrt{\int \left(\sqrt{q(\mathbf{x})} - \sqrt{p(\mathbf{x})}\right)^2 dx} \tag{8}$$

which is non negative, bounded (it takes values from 0 to $\sqrt{2}$) and symmetric (i.e., $HD(q,p) = HD(p,q)$). Additionally, it is defined for all possible values of $p(\mathbf{x})$ and $q(\mathbf{x})$ and does not make any assumptions about the distributions themselves. For all these reasons, in this setting we propose the HD as the metric to measure the mismatch between distributions.

[1] Note that the subscript t refers to the training set.

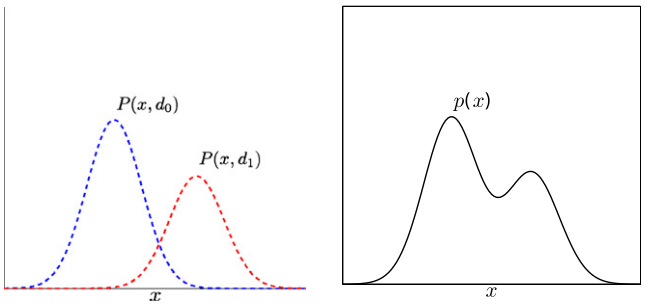

Fig. 2. Test (future) data. Joint probabilities $P(x,d_0)$ and $P(x,d_1)$ (left) and unconditional density $p(x)$ (right) for prior class probabilities ($P(d_0)$, $P(d_1)$) in the test set equal to (0.6,0.4).

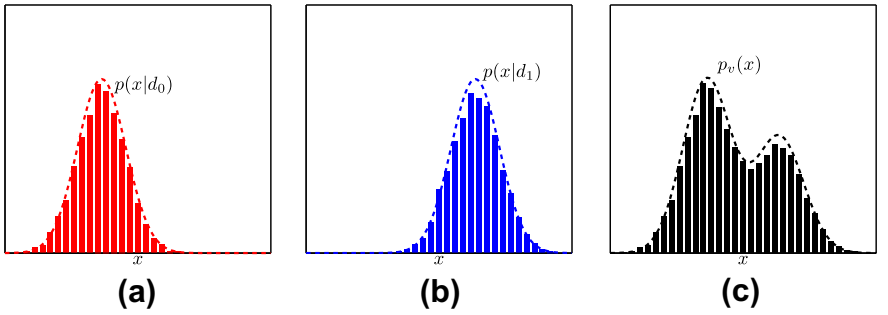

Fig. 3. Binned distributions of the class probability density functions ($p(x|d_0$ and $p(x|d_1)$) used to model a data distribution $p_v(x)$ with $P_v(d_0) = 0.6$ and $P_v(d_1) = 0.4$.

3.3. Quantification method HDx

The method HDx requires measuring the similarity between validation distributions $p_v(\mathbf{x})$ and the test data distribution $p(\mathbf{x})$ where $\mathbf{x}$ is a feature vector with n_f dimensions.

Similarity between two discrete data distributions can also be measured with the Hellinger distance [34,35] by converting them into binned distributions with a probability associated with each of the b bins. Thus, the HD between the test data distribution (with unknown priors $P(d_i)$) and a validation data distribution with a given class distribution $P_v(d_i)$ can be estimated by measuring the HD between the unlabeled test dataset $U = \{(\mathbf{x}^l),\ l = 1, \ldots, N\}$ and a validation dataset V (extracted from the available training data set S_t according to $P_v(d_i)$) as

$$HD(V,U) = \frac{1}{n_f}\sum_{f=1}^{n_f} HD_f(V,U) \tag{9}$$

where n_f is the number of features and $HD_f(V, U)$ represents the (intermediate) Hellinger distance between V and U according to feature f. Note that the final HD is the average of distances for all features.

The Hellinger distance for feature f is computed as

$$HD_f(V,U) = \sqrt{\sum_{i=1}^{b}\left(\sqrt{\frac{|V_{f,i}|}{|V|}} - \sqrt{\frac{|U_{f,i}|}{|U|}}\right)^2} \tag{10}$$

where b is the number of bins used to construct the histogram, $|U|$ the total number of test examples and $|U_{f,i}|$ the number of test examples whose feature f belongs to bin i. Likewise, $|V|$ and $|V_{f,i}|$ correspond to the validation dataset.

Estimating the class prior probabilities of the test set can be stated in a straightforward way as finding the class prior probabilities of the validation dataset $P_v(d_i)$ that minimize the HD with the test set.

In practice, generating validation datasets from the training dataset S_t with different prior probabilities can be conducted by subsampling and/or oversampling. However, this implies discarding and/or replicating instances and thus, losing information or adding no information. This is undesirable especially when the samples are scarce, or when trying to generate datasets where some of the classes have very low prior probabilities.

Our approach deals with this in the following way: instead of modeling the validation data distribution from a validation dataset V, we can construct the histogram for the class-conditional probability density functions $p(\mathbf{x}|d_0)$ and $p(\mathbf{x}|d_1)$

(assumed stationary). Then, a validation data distribution $p_v(\mathbf{x})$ for a given class distribution ($P_v(d_0)$, $P_v(d_1)$) with two classes can be computed as

$$p_v(\mathbf{x}) = p(\mathbf{x}|d_0)P_v(d_0) + p(\mathbf{x}|d_1)P_v(d_1) \tag{11}$$

For the sake of clarity, we will illustrate this with the binary classification problem depicted in Figs. 1 and 2. A binned distribution for each of the class probability density functions ($p(x|d_0)$ and $p(x|d_1)$) can be obtained from the set of training examples that belong to class 0 $\left(S_t^0\right)$ and class 1 $\left(S_t^1\right)$, respectively, as depicted in Fig. 3.

Next, for any class distribution ($P_v(d_0)$, $P_v(d_1)$) we are able to model the data distribution $p_v(x)$ according to (11). Fig. 3c depicts the data distribution for class prior probabilities $P_v(d_0) = 0.6$ and $P_v(d_1) = 0.4$. This way, we use all the data available and there is no need either to replicate or discard samples.

By making the substitution

$$\frac{|V_{f,i}|}{|V|} = \frac{|S_{t,f,i}^0|}{|S_t^0|}P_v(d_0) + \frac{|S_{t,f,i}^1|}{|S_t^1|}P_v(d_1) \tag{12}$$

into (10), the HD between $p(\mathbf{x})$ and $p_v(\mathbf{x})$ according to feature f can be computed based on the available labeled data set S_t. Here $|S_t^0|$ is the total number of training examples that belong to class-0 and $|S_{t,f,i}^0|$ is the number of training examples from class-0 whose feature f belongs to bin i. In the same way, $|S_t^1|$ and $|S_{t,f,i}^1|$ are the equivalent measures for class-1 and $P_v(d_0) = 1 - P_v(d_1)$ in the binary case. Note also that these values can be computed beforehand and stored for later use.

Therefore, it is straightforward to simulate validation data distributions with any probabilities $P_v(d_i)$ and measure their HD with the unlabeled test data distribution according to (9), (10) and (12). Finally, through a search in the probability space (in this work we use a simple for-loop), the estimated *a priori* probability of the test set is the one that minimizes this HD distance.

Let us assume that the test data follow the distribution depicted in Fig. 2b where the unknown class prior probabilities of the test set are $P(d_0) = 0.6$ and $P(d_1) = 0.4$. Fig. 4 plots the Hellinger distance between the test set distribution and several validation distributions with $P_v(d_1)$ that ranges in the interval [0,1]. It can be noticed that the minimum HD is achieved for the class prior probability ($P_v(d_1) = 0.4$) that matches the unknown test class distribution.

Notice that the method HDx requires the computation of a binned distribution. Although preliminary experimental results did not show a high sensitivity to the number of bins b, in order to get a more robust estimate, the algorithm is run for a wide range of bins and the class prior probabilities are estimated taking the median of the individual estimates. Algorithm 1 presents the procedure to get one of these estimates using b bins.

In this work, we have determined the class distribution that minimizes the Hellinger distance, calculating it for each value of class-1 prior probability (we focused on two-class problems) from 0 to 1 in small steps (0.01 in the experimental work) and returning the one that provides the minimum distance value. For a real application or multiclass problems, methods like hill-climbing could be implemented.

Algorithm 1. QuantificationMethod_HDx

Inputs:
Labeled data set $S_t = \{(\mathbf{x}^k, d^k), k = 1, \ldots, K\}$.
Test data set $U = \{(\mathbf{x}^l), l = 1, \ldots, N\}$
Number of bins b
Compute $|S_t^0|$, $|S_t|^1$ and $|U|$
for $f = 1$ to n_f
  for $i = 1$ to b
    Compute $|S_{t,f,i}^0|$, $|S_{t,f,i}^1|$ and $|U_{f,i}|$
  end for
end for
for $P_v(d_1) = 0$ to 1 in small steps
  for $f = 1$ to n_f
    Compute HD_f according to (10), using (12) with $P_v(d_1)$
  end for
  $HD[P_v(d_1)] = \frac{1}{n_f}\sum_{f=1}^{n_f} HD_f(P_v(d_1))$
end for
Outputs:
Prior probability estimation for test set U
$\widehat{P}(d_1) = arg\ min(HD)$, $\widehat{P}(d_0) = 1 - \widehat{P}(d_1)$

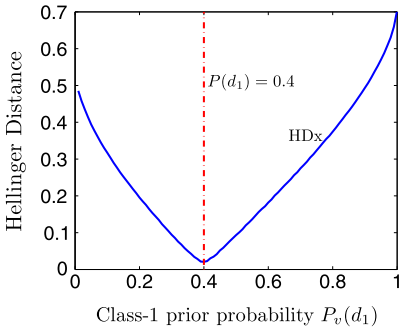

Fig. 4. Hellinger distance between the test data distribution $p(x)$ and different validation data distributions $p_v(x)$ generated for class prior probabilities that vary from $P_v(d_1) = 0$ to $P_v(d_1) = 1$. The dashed vertical line represents the actual class-1 prior probability $P(d_1)$ of the test data set. Data are defined in a one dimensional space ($n_f = 1$).

3.4. Quantification method HDy

The intuition behind the quantification technique HDy is taken from the HDx proposal, but it mainly differs from HDx in the use of a classifier. While HDx works directly with the feature vectors **x**, HDy employs the soft outputs **y** that a classifier, trained with instances from the training dataset, generates for the samples **x** (**y** = $\mathbf{f_w}$(**x**)). Therefore, HDx works with data defined in a n_f dimensional space whereas the data HDy deals with are defined in a space with dimensionality $M - 1$, where M is the number of classes. Note that in a binary classification problem, HDy works with one dimensional data whereas the HDx data has as many dimensions as the number of features.

The main advantage of HDx is that it does not require a classification model, but its main limitation is that its computational complexity increases as the number of features does. Therefore, its use in very high dimensional datasets is unfeasible and HDx is limited to datasets from low to medium dimensionality.

Another issue that has to be taken into account is data sparseness. This is a problem that usually has to be faced in real-world applications, where training data sets are very likely not to be fully representative in all regions of the n_f dimensional space, in particular when the data dimensionality is high. When this happens, the estimated HDx curve in Fig. 4 may be less reliable than that obtained measuring the HD between the classifier output distributions. In this case, the problem is simplified because distributional divergences are measured with data (the classifier outputs) defined in a one dimensional space for a two class problem. A comparison of both is shown in Fig. 5 for a binary classification problem where the data follow Gaussian distributions defined in a space with 20 dimensions ($n_f = 20$). Notice that HDy has a higher convexity so that estimating where its minimum lies is more reliable when the number of instances is limited.

Thus, the HD between the classifier output distribution for test data U and the classifier output distribution for validation data with a given class distribution $P_v(d_i)$ can be estimated as

$$HD(V,U) = \sqrt{\sum_{i=1}^{b}\left(\sqrt{\frac{|V_{y,i}|}{|V|}} - \sqrt{\frac{|U_{y,i}|}{|U|}}\right)^2} \tag{13}$$

where b is the number of bins used to construct the histogram, $|U|$ the total number of test examples and $|U_{y,i}|$ the number of test examples whose output y belongs to bin i. Likewise, $|V|$ and $|V_{y,i}|$ correspond to the validation dataset.

As in the case of the HDx method and in order to avoid subsampling and/or oversampling, we can make the substitution

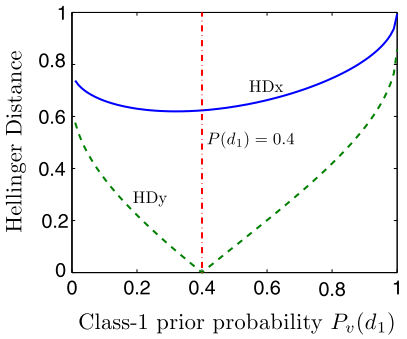

Fig. 5. Hellinger distance between the classifier output distributions (curve HDy) and the data itself (curve HDx) of a test set and different validation settings. Data are defined in a twenty dimensional space ($n_f = 20$).

$$\frac{|V_{y,i}|}{|V|} = \frac{|S^0_{t,y,i}|}{|S^0_t|} P_v(d_0) + \frac{|S^1_{t,y,i}|}{|S^1_t|} P_v(d_1) \tag{14}$$

where $|S^0_t|$ is the total number of training examples that belong to class-0 and $|S^0_{t,y,i}|$ is the number of training examples from class-0 whose output y belongs to bin i. In the same way, $|S^1_t|$ and $|S^1_{t,y,i}|$ are the equivalent measures for class-1 and $P_v(d_0) = 1 - P_v(d_1)$ in the binary case.

This procedure that is referred to as HDy quantification method is summarized in Algorithm 2.

Algorithm 2. QuantificationMethod_HDy

Inputs:
Labeled data set $S_t = \{(\mathbf{x}^k, d^k), k = 1, \ldots, K\}$.
Test data set $U = \{(\mathbf{x}^l), l = 1, \ldots, N\}$
Classifier $\mathbf{f_w}$
Number of bins b
Compute $|S^0_t|$, $|S^1_t|$ and $|U|$
Compute classifier outputs for S_t as $\{y^k = \mathbf{f_w}(\mathbf{x}^k), k = 1, \ldots, K\}$
Compute classifier outputs for U as $\{y^l = \mathbf{f_w}(\mathbf{x}^l), l = 1, \ldots, N\}$
for $i = 1$ to b
  Compute $|S^0_{t,y,i}|$, $|S^1_{t,y,i}|$ and $|U_{y,i}|$
end for
for $P_v(d_1) = 0$–1 in small steps
  Compute $HD[P_v(d_1)]$ according to (13), using (14) with $P_v(d_1)$
end for
Outputs:
Prior probability estimation for test set U
$\widehat{P}(d_1) = arg\ min(HD)$, $\widehat{P}(d_0) = 1 - \widehat{P}(d_1)$

4. Experimental methodology

4.1. Datasets

In this paper several public real-world datasets have been used to assess the performance of the quantification methods, which have also been tested in a real computer-vision-based quality control application of boar sperm.

4.1.1. Publicly available datasets

We have evaluated the performance of the quantification methods on 15 binary datasets from the UCI Machine Learning repository [36], which have a very wide range of size, number of features and class proportions, in order to provide diverse scenarios for the experiments (see details in Table 2). Some of these datasets are not originally binary, so we have converted them to two-classed datasets by grouping the target classes:

- *CMC:* The three target classes in the original dataset have been grouped in two: ''using contraceptive methods'' and ''not using contraceptive methods''.
- *Page blocks:* The original classes (different blocks in pages) have been divided in two target classes: ''Pictures'' and the other blocks.
- *Semeion handwritten digit:* In order to make this dataset a binary one, we have considered as minority class the original class ''digit 8'', and the other class is ''the other digits''.
- *Red and white wine:* In both datasets the target classes have been considered to be ''bad quality'' (interval [0,5] in the original dataset), and ''good quality'' (interval [6,10] in the original dataset).
- *Yeast:* The target classes that we have considered are ''proteins in the nucleus of the cell'' and ''proteins in a different location''.

4.1.2. Sperm cell data set

One of the goals of a seminal quality control application based on computer vision is to estimate automatically the proportion of sperm cells with intact or damaged acrosome. The quantification methods were tested on a real boar semen image dataset. The image data acquisition was conducted at CENTROTEC under the guidance of veterinarian researchers from the University of León.

Using a digital camera connected to a microscope, boar semen images are captured with a magnification of 100× with a resolution of 780 × 580 both under a fluorescent illumination and phase contrast. The former highlights the state of the

Table 2
Datasets description

Datasets	Samples	Features	Minority class (%)
Breast Cancer Wisconsin	699	9	34.48
CMC	1473	9	22.61
Coil	9822	85	5.97
Diabetes	770	8	38.57
German Credits	1000	24	30.00
Letters (G)	20000	16	3.87
Letters (H)	20000	16	3.67
Mammographic mass	830	5	48.55
Page blocks (picture)	5473	10	2.10
Phoneme	5404	5	29.35
Semeion (8)	1593	256	9.73
Spambase	4601	57	39.40
Wine quality (red)	1599	11	3.94
Wine quality (white)	4898	11	3.74
Yeast	1484	8	28.91

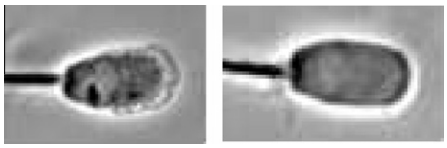

Fig. 6. Grey level images of damaged (left) and intact (right) acrosomes of boar sperm.

acrosome, so it allows the system to label each head. Afterwards, the images are automatically cropped, segmented [37] and described in terms of a set of 20 texture features called Wavelet Co-occurrence features (WCF) [38]. We direct the reader interested in further details about how the images have been described to [39]. The image set has 1849 instances: 904 intact and 945 damaged spermatozoon heads. An example of both classes can be seen in Fig. 6.

4.2. Base classifiers

Except for HDx, the quantification methods evaluated in this article need a classifier. PP also requires that the classifier outputs provide class posterior probability estimates.

Classification was carried out with a back-propagation neural network with one hidden layer and a logistic sigmoid transfer function for the hidden and the output layers. Learning was carried out with a momentum and adaptive learning rate algorithm. It is well known that when the training is carried out minimizing some loss functions such as the Mean Square Error used in this work [31], the outputs provided by this model are estimates of class posterior probabilities. Additionally, we tested the quantification techniques with Naïve Bayes and Logistic Regression.

Data were normalized with zero mean and standard deviation equal to one. The neural network architecture as well as the number of training cycles were determined by 10-fold cross-validation separately for each dataset.

4.3. Performance metrics

The mismatch between the real class distribution and the estimation provided by the quantification methods is measured by means of the Mean Absolute Error (MAE) that focuses on the class of interest – called class-1 in all our experiments – and is defined as the absolute value of the difference between its actual prior probability and the estimated one.

$$MAE(P(d_1), \widehat{P}(d_1)) = |P(d_1) - \widehat{P}(d_1)| \tag{15}$$

The MAE does not include the importance of the error with respect to the true value. For this reason, the Mean Relative Error (MRE), that includes such information, is used as well and computed in % as

$$MRE(P(d_1), \widehat{P}(d_1)) = \frac{|P(d_1) - \widehat{P}(d_1)|}{P(d_1)} 100 \tag{16}$$

4.4. Statistical tests

The Wilcoxon signed-ranks test [40] is the non-parametric alternative to the paired t-test. Therefore, it is a pairwise test that aims to detect significant differences between the performance results of two algorithms. Let a_i be the difference

between the performance scores of the two algorithms on the ith out of N datasets. These differences are ranked according to their absolute values (average ranks are assigned in case of ties). Let R^+ be the sum of the ranks where the difference is positive – i.e. the second algorithm outperforms the first one –, and R^- the sum of ranks for the opposite. Ranks of $a_i = 0$ are split evenly among the sums; one of them would be ignored if there was an odd number of them. This is described in (17) and (18).

$$R^+ = \sum_{a_i>0} rank(a_i) + \frac{1}{2}\sum_{a_i=0} rank(a_i) \tag{17}$$

$$R^- = \sum_{a_i<0} rank(a_i) + \frac{1}{2}\sum_{a_i=0} rank(a_i) \tag{18}$$

Let T be the smaller of the sums, $T = min(R^+, R^-)$. If it is less than, or equal to the value of the Wilcoxon distribution with N degrees of freedom the null hypothesis of equality of the algorithms is rejected. Tables for the Wilcoxon distribution may be included on most books on general statistic, such as [41] When the number of datasets, N, is large (such as $N > 25$), the statistics shown in (19) is distributed approximately normally. The null hypothesis can be rejected if z is smaller than -1.96 with $\alpha = 0.05$ [42].

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \tag{19}$$

5. Experiments and results

In this section, the quantification techniques based on the Hellinger Distance (HDx and HDy) proposed in this work as well as other previous approaches are evaluated on the 15 datasets presented in Section 4.1.1, and then in the context of a real boar semen quality control application described in Section 4.1.2. Comparisons will be carried out with the baseline approach Classify and Count (CC), the method Adjusted Count (AC) by Forman based on the classifier confusion matrix and the Median Sweep (MS) proposal derived from it. The Posterior Probability (PP) technique based on probability estimators will be evaluated as well.

5.1. Quantification based on the Hellinger distance

The aim of the following experiment is to assess the performance of the quantification methods based on the Hellinger Distance (HDx and HDy). Each data set in Table 2 has been divided into two subsets by stratified sampling: 30% of the elements will be used as the test set, U, while the remaining 70% will be used for training the classifier and generating the validation sets. From now on the latter set will be called S_t.

Note that since the test set U is extracted by stratified sampling from the whole data set for each of the problems assessed, its class distribution is given by the class frequency of the original data set. Thus, in the 15 different scenarios evaluated, the proportion of the minority class[2] varies from 2.10% in the Pageblocks application to 48.55% in the Mammographic mass problem (see Table 2 for more details).

In order to face a situation of uncertainty in the class distribution, the classifier has been trained with a balanced dataset, which has all the instances of S_t that belong to the minority class and the same number of the majority class. In this section, the neural classifier is evaluated. The number of hidden nodes and the number of training cycles of each network has been determined experimentally by 10-fold cross-validation separately for each dataset. Table 3 shows this information together with the misclassification rate for each database.

In this work, the number of bins b used in HDx and HDy was chosen from 10 to 110 in steps of 10, and the final estimated *a priori* probability was taken as the median of these 11 estimates.

For each problem, results are the average of 50 randomly extracted test sets. Table 4 shows the MAE achieved by HDx and HDy. While HDx achieves good performance in all problems (in the order of 10^{-2}), it is clearly outperformed by HDy, which achieves lower absolute errors, except in the case of Mammographic mass.

We have also performed the Wilcoxon signed-rank test [40] with $\alpha = 0.05$ with the absolute and the relative errors. The results of this test (see Table 5) show that these differences are statistically significant, so it confirms that HDy clearly outperforms HDx when the neural network is used as classifier.

In theory, the performance of the HDy quantification technique does not depend on the classifier error rate, as long as its outputs are not randomly generated. However, something interesting may be observed in practice. Although HDy may provide good performance regardless of the neural network performance, as it is the case for the Coil problem (MAE = 0.012, neural network error rate = 37.04%), the scatter plot in Fig. 7 shows generally positive correlation between the error in

[2] The minority class of these databases is called class 1.

Table 3
Neural network configurations with each classification problem

Datasets	Training cycles	Hidden nodes	Error rate (%)
Breast Cancer Wisconsin	200	2	4.64
CMC	200	2	37.04
Coil	200	2	30.39
Diabetes	200	2	20.43
German Credits	200	2	32.15
Letters (G)	400	5	7.67
Letters (H)	400	5	11.58
Mammographic mass	200	2	17.85
Page blocks (picture)	300	2	6.60
Phoneme	300	5	19.12
Semeion (8)	200	3	13.60
Spambase	300	3	7.34
Wine quality (red)	300	2	25.91
Wine quality (white)	400	2	28.40
Yeast	400	2	28.73

Table 4
MAE of the quantification methods HDx and HDy on the UCI databases using a neural network.

Dataset	HDx	HDy
Breast Cancer Wisconsin	0.014	0.012
CMC	0.064	0.038
Coil	0.017	0.012
Diabetes	0.050	0.028
German Credits	0.053	0.034
Letters (G)	0.005	0.002
Letters (H)	0.006	0.003
Mammographic mass	0.031	0.032
Page blocks (picture)	0.011	0.004
Phoneme	0.015	0.013
Semeion (8)	0.017	0.016
Spambase	0.013	0.006
Wine quality (red)	0.019	0.012
Wine quality (white)	0.021	0.011
Yeast	0.035	0.026

Table 5
Wilcoxon signed-rank test of the methods HDx and HDy.

Comparison	$R+$	$R-$	p-Value	Null hypothesis of equality
HDy vs HDx	2	120	0.00018	Rejected (HDy outperforms HDx)

the quantification process and the classification error. Therefore, efforts have to be made to tune the classifier parameters so that the classifier performance is optimized.

5.2. Comparison of quantification methods using a neural network classifier

In this section the method HDy (which outperforms HDx, as has been shown in Section 5.1) will be compared with CC (Classify & Count), AC (Adjusted Count), MS (Median Sweep) and PP (Posterior Probability). The comparison has been carried out as it was done in Section 5.1.

The confusion matrices required for the methods AC and MS were estimated from the training set by 50-fold cross-validation separately for each dataset, as suggested in [26]. For these methods, we did not discard any test instance regardless of its classification score. The MS method used nine confusion matrices computed for classification thresholds from 0.1 to 0.9 in steps of 0.1. Likewise, the estimated class proportion provided by MS is taken as the median of all the nine individual estimates, without making any special treatment for any threshold. For the PP method, the maximum number of iterations is set to 30, but it stops if the difference between two consecutive iterations is lower than 0.0001. Table 6 shows the MAE of the quantifications as well as their ranks. The final tier of the table shows the average rank over the 15 problems.

Results highlight the importance of using an estimation method rather than relying on just counting the results of the classification, as the method CC does (with MAE up to 0.31). The Hellinger distance based procedure is more reliable than

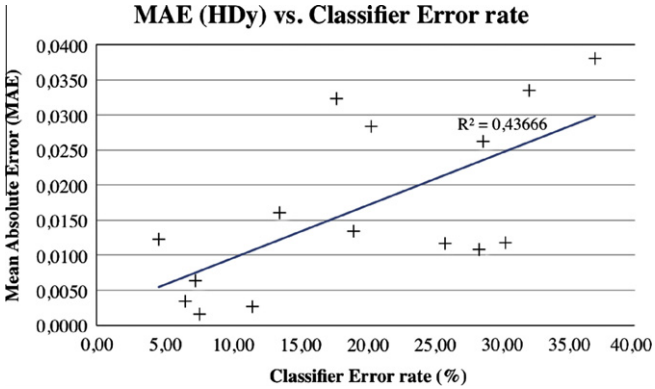

Fig. 7. Mean Absolute Error (MAE) for HDy against the neural network error rate.

Table 6
MAE of the quantification methods HDy, CC, AC, MS and PP on the UCI databases using a neural network classifier.

Dataset	CC	AC	MS	PP	HDy
Breast Cancer Wisconsin	0.022 (5)	0.011 (1.5)	0.011 (1.5)	0.014 (4)	0.012 (3)
CMC	0.215 (5)	0.069 (2)	0.110 (4)	0.078 (3)	0.038 (1)
Coil	0.311 (5)	0.091 (3)	0.067 (2)	0.207 (4)	0.012 (1)
Diabetes	0.052 (5)	0.039 (3)	0.048 (4)	0.034 (2)	0.028 (1)
German-Credits	0.142 (5)	0.090 (3)	0.097 (4)	0.083 (2)	0.034 (1)
Letters (G)	0.074 (5)	0.008 (2)	0.010 (3)	0.030 (4)	0.002 (1)
Letters (H)	0.105 (5)	0.008 (2)	0.011 (3)	0.038 (4)	0.003 (1)
Mammographic mass	0.026 (1)	0.039 (4)	0.044 (5)	0.032 (2.5)	0.032 (2.5)
Page blocks (picture)	0.070 (5)	0.012 (3)	0.009 (2)	0.018 (4)	0.004 (1)
Phoneme	0.132 (5)	0.018 (3)	0.020 (4)	0.017 (2)	0.013 (1)
Semeion (8)	0.090 (5)	0.017 (3)	0.016 (1.5)	0.045 (4)	0.016 (1.5)
Spambase	0.015 (5)	0.008 (3)	0.010 (4)	0.007 (2)	0.006 (1)
Wine quality (red)	0.269 (5)	0.089 (3)	0.087 (2)	0.224 (4)	0.012 (1)
Wine quality (white)	0.231 (5)	0.052 (3)	0.037 (2)	0.153 (4)	0.011 (1)
Yeast	0.155 (5)	0.043 (3)	0.075 (4)	0.038 (2)	0.026 (1)
Avg. rank	4.733	2.767	3.067	3.167	1.267

the other quantification methods. It is noticeable that the MAE of HDy is never too high, even in the case of very imbalanced datasets. The MAE of HDy is always lower than 0.040 and it outperforms the other methods in most cases. In general terms, it has the smallest average rank.

In order to determine whether these differences are significant we have carried out a statistical comparison between the performance of the methods. We have specifically selected the Wilcoxon signed-ranks test [40] with $\alpha = 0.05$. We have made comparisons between the algorithm which has achieved the best average rank (HDy) and the others to determine if there are significant differences. These tests, whose results can be seen in Table 7 show that there are statistically significant differences between HDy and the other methods.

5.3. Quantification with different base classifiers

In this section the quantification techniques are assessed on the UCI datasets for other base classifiers: Naïve Bayes and Logistic Regression. Experiments are conducted as indicated in Section 5.2.

Regarding the quantification methods based on the Hellinger distance, the MAE for HDy is shown in Tables 8 and 9 for the Logistic Regression model and Naïve Bayes, respectively. Results for HDx (no classifier required) were previously presented

Table 7
Wilcoxon signed-rank test for the method HDy against the others using a neural network classifier.

Comparison	R^+	R^-	p-Value	Null hypothesis of equality
HDy vs CC	1	119	0.00012	Rejected (HDy outperforms CC)
HDy vs AC	2	118	0.00018	Rejected (HDy outperforms AC)
HDy vs MS	3	117	0.00031	Rejected (HDy outperforms MS)
HDy vs PP	1	119	0.00012	Rejected (HDy outperforms PP)

Table 8
MAE of the quantification methods HDy, CC, AC, MS and PP on the UCI databases.

Dataset	Logistic Regression				
	CC	AC	MS	PP	HDy
Breast Cancer Wisconsin	0.013 (1.5)	0.014 (3.5)	0.014 (3.5)	0.013 (1.5)	0.019 (5)
CMC	0.226 (5)	0.066 (3)	0.125 (4)	0.055 (2)	0.038 (1)
Coil	0.326 (5)	0.091 (2)	0.094 (3)	0.233 (4)	0.013 (1)
Diabetes	0.041 (4)	0.034 (3)	0.044 (5)	0.030 (1)	0.032 (2)
German Credits	0.115 (5)	0.06 (3)	0.092 (4)	0.049 (2)	0.040 (1)
Letters (G)	0.194 (5)	0.012 (2)	0.035 (3)	0.06 (4)	0.006 (1)
Letters (H)	0.238 (5)	0.015 (2)	0.037 (3)	0.049 (4)	0.008 (1)
Mammographic mass	0.022 (1)	0.033 (4)	0.036 (5)	0.031 (3)	0.029 (2)
Pageblocks (picture)	0.100 (5)	0.035 (3)	0.034 (2)	0.090 (4)	0.006 (1)
Phoneme	0.146 (5)	0.024 (3)	0.073 (4)	0.022 (2)	0.019 (1)
Semeion (8)	0.337 (4.5)	0.124 (2)	0.126 (3)	0.337 (4.5)	0.018 (1)
Spambase	0.012 (4)	0.010 (3)	0.018 (5)	0.008 (2)	0.007 (1)
Wine quality (red)	0.259 (5)	0.042 (3)	0.031 (2)	0.117 (4)	0.016 (1)
Wine quality (white)	0.235 (5)	0.023 (3)	0.020 (2)	0.037 (4)	0.012 (1)
Yeast	0.120 (5)	0.056 (3)	0.093 (4)	0.047 (2)	0.043 (1)
Avg. Rank	4.333	2.833	3.500	2.933	1.400

Table 9
MAE of the quantification methods HDy, CC, AC, MS and PP on the UCI databases.

Dataset	Naive Bayes				
	CC	AC	MS	PP	HDy
Breast Cancer Wisconsin	0.028 (4.5)	0.016 (2.5)	0.016 (2.5)	0.028 (4.5)	0.014 (1)
CMC	0.306 (5)	0.072 (2)	0.110 (3)	0.293 (4)	0.045 (1)
Coil	0.381 (5)	0.114 (2)	0.124 (3)	0.376 (4)	0.034 (1)
Diabetes	0.049 (3)	0.059 (4)	0.060 (5)	0.043 (2)	0.037 (1)
German Credits	0.139 (5)	0.073 (2.5)	0.074 (2.5)	0.115 (4)	0.046 (1)
Letters (G)	0.222 (5)	0.011 (2)	0.011 (3)	0.156 (4)	0.005 (1)
Letters (H)	0.266 (5)	0.012 (2)	0.021 (3)	0.158 (4)	0.005 (1)
Mammographic mass	0.029 (1)	0.041 (5)	0.038 (4)	0.037 (3)	0.034 (2)
Page blocks (picture)	0.021 (5)	0.011 (3)	0.01 (2)	0.012 (4)	0.008 (1)
Phoneme	0.201 (5)	0.026 (2)	0.066 (3)	0.141 (4)	0.018 (1)
Semeion (8)	0.283 (5)	0.027 (1.5)	0.027 (1.5)	0.278 (4)	0.032 (3)
Spambase	0.147 (4.5)	0.017 (2.5)	0.017 (2.5)	0.147 (4.5)	0.015 (1)
Wine quality (red)	0.248 (5)	0.060 (2)	0.061 (3)	0.174 (4)	0.014 (1)
Wine quality (white)	0.133 (5)	0.024 (2)	0.033 (3)	0.061 (4)	0.010 (1)
Yeast	0.665 (4)	0.163 (1)	0.172 (2)	0.672 (5)	0.251 (3)
Avg. Rank	4.467	2.400	2.867	3.933	1.333

in Table 4. The Wilcoxon signed-rank tests with $\alpha = 0.05$ showed that the methods HDx and HDy are statistically equivalent when the classifier used is either Naïve Bayes or Logistic Regression. Recall that HDy together with a neural network outperformed HDx, what indicates that the classifier choice may be an important issue.

The MAE achieved when the class distribution is estimated with CC, AC, MS, PP and HDy (results equivalent to HDx) using Logistic Regression is shown in Table 8. Note that HDy gets the lowest average rank over the 15 UCI datasets. The Wilcoxon signed-rank tests (see Table 10) where HDy is compared with the other quantification techniques indicate that the difference is statistically significant. The same analysis was carried out with the Naïve Bayes baseline classifier (see Tables 9 and 11) and it leads to the same conclusion regarding the superiority of HDy against the other methods.

5.4. Quantification of damaged acrosomes

The performance of the quantification methods has also been assessed on an application of semen quality control (see Section 4.1.2).

The training set has 70% of the minority class examples from the whole dataset, and the same number of elements from the majority class. Test sets have fixed size of 280 instances. Both sets are mutually exclusive, and randomly selected. We have evaluated 10 scenarios where the proportion of damaged acrosomes in the test phase (operational environment) varies from 0.05 to 0.50. We have explored this wide range of deployment conditions in order to evaluate the algorithms. However, only the samples with proportion of damaged acrosomes equal or lower than 0.20 have interest from the point of view of veterinarian practice.

Table 10
Wilcoxon signed-rank test for the method HDy against the others using Logistic Regression.

Logistic Regression				
Comparison	R^+	R^-	p-Value	Null hypothesis of equality
HDy vs CC	5	115	0.00061	Rejected (HDy outperforms CC)
HDy vs AC	4	116	0.00042	Rejected (HDy outperforms AC)
HDy vs MS	1	119	0.00012	Rejected (HDy outperforms MS)
HDy vs PP	8	112	0.00152	Rejected (HDy outperforms PP)

Table 11
Wilcoxon signed-rank test for the method HDy against the others using Naïve Bayes.

Naive Bayes				
Comparison	R^+	R^-	p-Value	Null hypothesis of equality
HDy vs CC	1	119	0.00012	Rejected (HDy outperforms CC)
HDy vs AC	19	101	0.01721	Rejected (HDy outperforms AC)
HDy vs MS	19	101	0.01806	Rejected (HDy outperforms MS)
HDy vs PS	0	120	0.00006	Rejected (HDy outperforms PS)

Table 12
Sperm cell data set. MAE of the quantification methods for 10 different test scenarios using a neural classifier.

$P(d_1)$	CC	AC	MS	PP	HDx	HDy
0.05	0.036	0.011	0.013	0.005	0.062	0.007
0.10	0.032	0.010	0.012	0.008	0.056	0.008
0.15	0.028	0.009	0.011	0.007	0.050	0.007
0.20	0.023	0.010	0.011	0.008	0.041	0.008
0.25	0.020	0.009	0.010	0.008	0.032	0.008
0.30	0.016	0.008	0.009	0.008	0.024	0.008
0.35	0.013	0.009	0.009	0.008	0.020	0.008
0.40	0.010	0.009	0.009	0.008	0.018	0.009
0.45	0.008	0.008	0.008	0.008	0.022	0.008
0.50	0.008	0.008	0.008	0.008	0.026	0.008

Table 13
Sperm cell data set. MRE of the quantification methods for 10 different test scenarios using a neural classifier.

$P(d_1)$	CC	AC	MS	PP	HDx	HDy
0.05	71.63	21.18	25.53	10.78	124.08	13.92
0.10	32.10	10.31	12.40	7.46	56.28	7.48
0.15	18.50	6.15	7.39	4.71	33.49	4.88
0.20	11.59	4.78	5.52	4.10	20.24	4.20
0.25	7.87	3.73	3.94	3.02	12.67	3.22
0.30	5.42	2.79	3.04	2.64	7.89	2.72
0.35	3.75	2.48	2.52	2.29	5.59	2.41
0.40	2.53	2.24	2.20	2.05	4.58	2.24
0.45	1.70	1.83	1.80	1.68	4.94	1.76
0.50	1.49	1.66	1.64	1.56	5.17	1.55

We have used a neural network with the lowest error rate estimated by 10-fold cross-validation (3.93%). It has three neurons in the hidden layer and trained with 400 cycles. The rest of the design of the experiment is identical to the one described in Sections 5.1 and 5.2.

Table 12 shows the MAE of HDx and HDy, as well as CC, AC, MS and PP for each of the 10 scenarios. A Wilcoxon signed-rank test between the method that achieves the lowest MAE and the others has been performed. To carry out the test, we used in this case the 50 performance values [43] from the experiment. The best methods (statistically equivalent) for each scenario are underlined in Table 12. Table 13 shows the same information with respect to the MRE metric.

For test environments with proportions of damaged acrosomes between 0.05 and 0.25, HDy and PP outperform the others, whereas they are statistically equivalent. What is more important, their estimates lead to a very low MAE in these scenarios (from 0.005 to 0.008), which are promising results in the application field.

PP performance strongly depends on the quality of the estimates of the class *a posteriori* probabilities provided by the classifiers. In order to get good *a priori* probability estimates it is necessary that the *a posteriori* probabilities are well approximated [8]. A low rate of misclassifications (in this case 3.93%) does not guarantee the success of the PP method, but the classifier outputs should be well calibrated. When this is not the case, the classifier scores can be scaled in the probability space by methods like Isotonic or Logistic Regression (for a review of different methods and details see [44]). The HDy method, however, does not require output calibration.

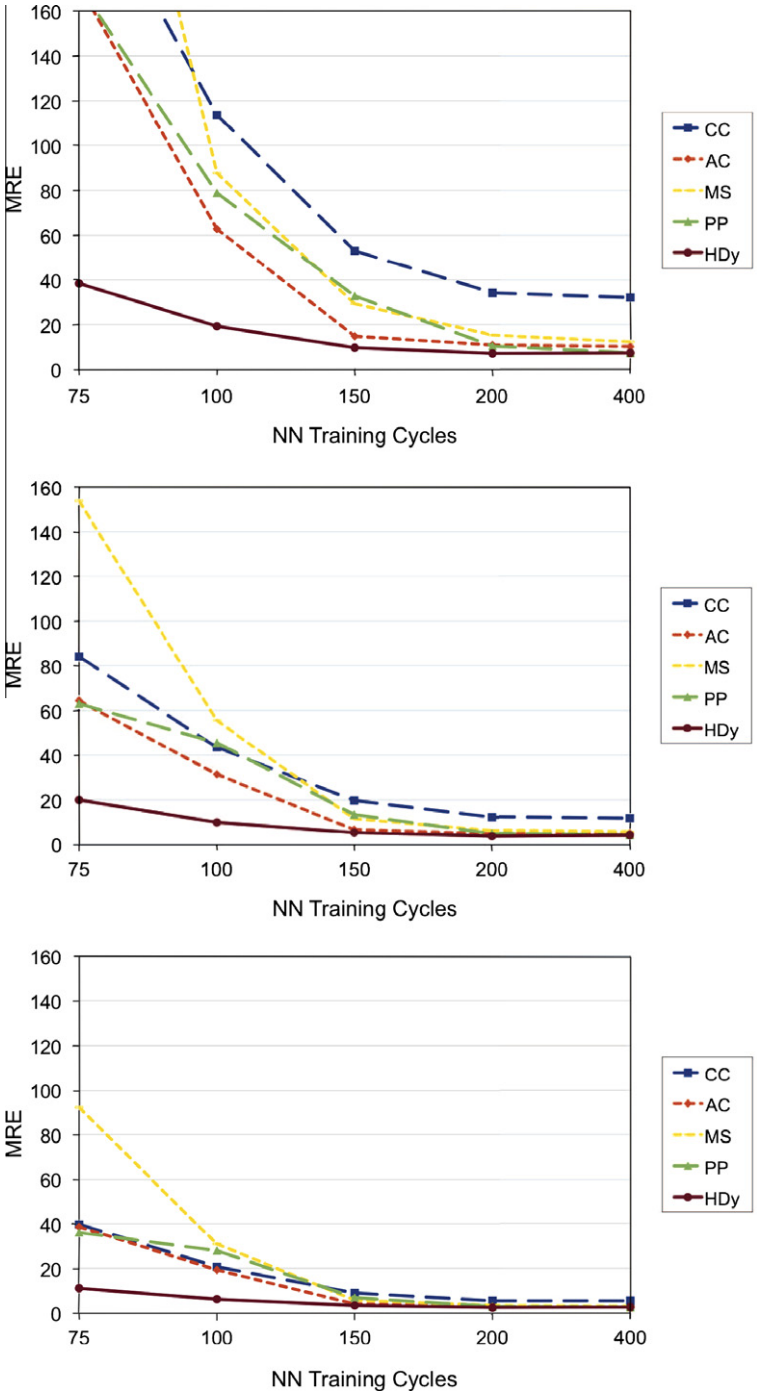

Fig. 8. MRE for CC, AC, MS, PP and HDy using neural networks trained with different number of cycles. Test set with class-1 *a priori* probability equal to 0.10, 0.20 and 0.30 (from top to bottom).

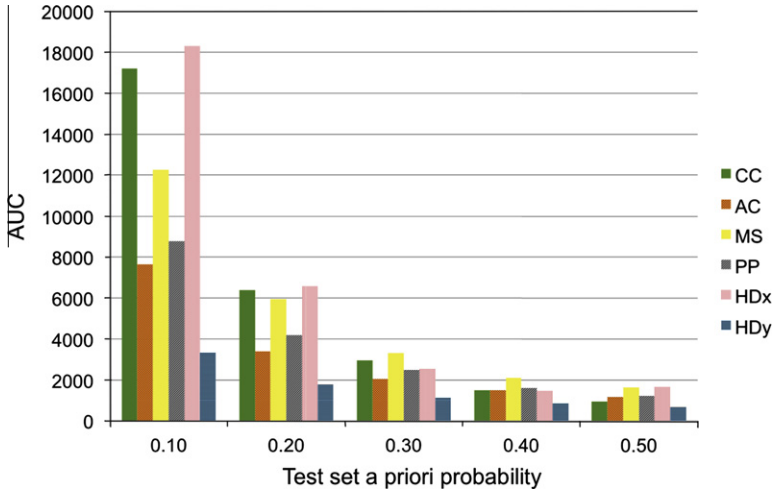

Fig. 9. Area under the curves (for different number of training cycles) of the quantification methods.

Looking at the whole picture, it can be observed that HDy and PP are always the best methods in any scenario of the boar semen application. We also show that HDx is not suitable for this application. With respect to AC and MS, both have similar performance and are only competitive with HDy and PP for deployment conditions with relatively high proportion of damaged cells (30% and higher, which is out of the range of practical interest). Finally, once again results show the benefits from using an estimation method instead of relying on the naïve CC. For instance, in the estimation of a proportion of 0.10, the MRE can be reduced from 32.1% (CC) to 7.4% (HDy and PP).

We have also compared the performance of the quantification methods with that of a veterinarian expert that uses the same grey level images as our computer vision technique. The images obtained through fluorescence that require stains were employed as the ground truth and results show that the expert global error rate in classification is 12.7%. Regarding the quantification task, the empirical study shows that the quantification method HDy far surpasses the human level of performance. For instance, for a test scenario with a proportion of damaged cells equal to 0.10, the MAE of HDy is 0.008 while the MAE for the expert is extremely high (0.250). In the same way, when the proportion is 0.15, HDy achieves an MRE of 4.88% whereas the human MRE goes to 96.30%. The computer-based quantification method has the additional advantages of saving time, money and not requiring specialized staff.

5.4.1. Robustness

In this section, we study the robustness of the estimation methods with respect to the classifier performance. The aim of this experiment is to explore how the quantification is affected by the performance of the underlying classifier and, thus, how important the tuning of the classifier in the estimation is. In order to evaluate it, we use several neural networks which were trained with 75, 100, 150 and 200 cycles, and have three neurons in the hidden layer. We did not use the neural net used in Section 5.4. Their error rates estimated by 10-fold cross-validation were 32.10%, 15.40%, 6.47% and 4.15%, respectively.

Fig. 8 shows the evolution of the MRE (in %) for different number of training cycles on three scenarios with different *a priori* probability of class-1 (damaged cells): 0.10, 0.20, 0.30. All subfigures are plotted with the same axis limits, so that visually we can appreciate how the performance is affected.

Regarding the test scenario with a proportion of damaged cells equal to 0.10, the AC and MS methods have a maximum MRE around 12% for the optimal classifier (that one trained with 400 cycles) while they reach values of 80% when the network is trained with 100 cycles and higher than 170% for 75 cycles. In the same way, the results show that the PP method achieves the best estimates for an optimal classifier, but its performance strongly depends on the training conditions. Thus, when the network is trained with 75 cycles, its MRE may be higher than 160%, while the maximum MRE when the number of training cycles is 400 is lower than 7%. With respect to the HDy method, its performance degradation is smaller than that for other methods when the base classifier performance worsens. For example, the maximum MRE of HDy is around 7% when the number of training cycles is 200, and it rises to 19% when the network is trained with 100 cycles, while in the case of PP, this increase goes from 10% to 78%.

HDy clearly outperforms the other quantification methods when the classifier performance is poor – e.g. 75, 100, 150 and 200 training cycles. This is especially noticeable when the proportion of class-1 elements in the test set is low (at the top graph in Fig. 8).

Moreover, it is observed that HDy is much more robust than the other methods for imbalanced test sets. In order to illustrate this fact, the area under the curves (AUC) in Fig. 8 has been computed and shown in Fig. 9 for test set class distributions from 0.10 to 0.50. As the curves have been plotted by joining the relative errors of the five values of training cycles, the AUC

has been computed by means of numerical integration based on the Trapezoidal rule. In the case of HDx, a horizontal line is considered since there is no classifier dependence.

Fig. 9 confirms high dependence on the class prior probability of the test set for CC, PP, AC and especially for MS and HDx. The method HDy is more robust than the others with respect to the deployment conditions.

6. Conclusions and future work

In this work we have addressed the problem of automatically estimating the class distribution of an unlabeled dataset (also known as quantification).

Our proposal is based on the Hellinger distance in order to measure distributional divergences between a new dataset and validation sets with known proportions and find the most similar one (quantification method HDx). Likewise, a classifier can be used and its outputs for the data examples can be used instead (quantification method HDy).

These methods have been compared with the naïve approach of just counting the outputs of the classifier and other methods proposed in the literature (AC, MS and PP), both using public databases from the UCI, and data from a real boar semen quality control application.

Results show that the naïve approach of Classify and Count provide very poor estimates, especially when the datasets are imbalanced. HDy appears to be a very appealing method for this task as either it outperforms or matches the other methods when evaluated on several datasets, consistently providing estimates with low deviations from the true value, even if the datasets are imbalanced. Moreover, HDy does not require a classifier that provides estimates of the class posterior probabilities and its dependence on the base classifier performance when a neural network is used is not as strong as the other quantification methods.

Note that the quantification task, although related, is different to supervised classification since we are not interested in the individual prediction for each instance in the test set. Quantification is itself interesting in many applications with non-stationary distributions, a very frequent problem in real applications. Supervised classification in environments with imprecise class distributions may also benefit from quantification. It allows us to detect and prevent a drop in classifier performance due to shifts in the class prior probabilities and allows us to adapt the classifier to the new operational conditions. An adaptive approach in order to avoid this performance deterioration can be followed based on estimating the class prior probabilities in the environment where the classifier is going to be deployed. Once this quantification is available the decision threshold can be adapted using the ROC curve as indicated in [15] or if the classifier provides posterior probability estimates, these outputs can be adjusted accordingly as in [8,11].

Although the quantification methods based on the Hellinger Distance (HDx and HDy) are originally multiclass, the search method we propose would only be suitable for binary classification problems. Our immediate future work is to extend the methods based on the Hellinger distance to the multiclass case addressing the problem directly (instead of using 1-vs-1 or 1-vs-all approaches), modifying the search in the probability space technique.

We will also tackle the design of a committee system that combines or merges different quantification information so that estimates can be improved. Designing a committee of quantifiers where each member uses a different baseline classifier or where each member has been trained with different class prior probabilities are some alternatives to explore. Other possibilities include using the CC or PP method to make a first estimate and then, applying the HDy method with a base classifier optimized for a class distribution close to the estimate given by CC (and chosen from a pool of classifiers previously trained).

Acknowledgments

This work has been partially supported by the research projects DPI2009-08424 and TEC2011-22480 from the Spanish Ministry of Education and Science.

The authors thank CENTROTEC for providing us the semen samples and for their collaboration in the acquisition of the sperm images.

References

[1] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley and Sons, 2001.
[2] L.L. Minku, A.P. White, X. Yao, The impact of diversity on online ensemble learning in the presence of concept drift, IEEE Transactions on Knowledge and Data Engineering 22 (5) (2010) 730–742.
[3] S. Gu, Y. Tan, X. He, Recentness biased learning for time series forecasting, Information Sciences, http://dx.doi.org/10.1016/j.ins.2010.09.004.
[4] Y.S. Chan, H.T. Ng, Estimating class priors in domain adaptation for word sense disambiguation, in: ACL-44: Proc. of the 21st Int. Conf. on Computational Linguistics, 2006, pp. 89–96.
[5] A. Guerrero-Curieses, R. Alaiz-Rodríguez, J. Cid-Sueiro, Cost-sensitive and modular land-cover classification based on posterior probability estimates, International Journal of Remote Sensing 30 (22) (2009) 5877–5899.
[6] C. Drummond, R.C. Holte, Cost curves: an improved method for visualizing classifier performance, Machine Learning 65 (1) (2006) 95–130.
[7] S. Vucetic, Z. Obradovic, Classification on data with biased class distribution, in: Proceedings of the 12th European Conference on Machine Learning (ECML, Freiburg), 2001, pp. 527–538.
[8] M. Saerens, P. Latinne, C. Decaestecker, Adjusting a classifier for new a priori probabilities: a simple procedure, Neural Computation 14 (2002) 21–41.
[9] J.C. Xue, G.M. Weiss, Quantification and semi-supervised classification methods for handling changes in class distribution, in: KDD '09: Proceedings of the 15th Int. Conf. on Knowledge Discovery and Data Mining, 2009, pp. 897–906.

[10] R. Alaiz-Rodríguez, N. Japkowicz, Assessing the impact of changing environments on classifier performance, in: Proceedings of the Canadian Society for Computational Studies of Intelligence, 21st Conference on Advances in Artificial Intelligence, Canadian AI'08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 13–24.
[11] 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 (16) (2011) 2614–2623.
[12] A. Fernández, M.J. del Jesus, F. Herrera, Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets, International Journal of Approximate Reasoning 50 (2009) 561–577.
[13] A. Fernández, M.J. del Jesus, F. Herrera, On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets, Information Sciences 180 (8) (2010) 1268–1291.
[14] C. Seiffert, T.M. Khoshgoftaar, J.V. Hulse, A. Folleco, An empirical study of the classification performance of learners on imbalanced and noisy software quality data, Information Sciences, 2011, http://dx.doi.org/10.1016/j.ins.2010.12.016.
[15] F. Provost, T. Fawcett, Robust classification systems for imprecise environments, Machine Learning 42 (3) (2001) 203–231.
[16] M.-C. Chen, L.-S. Chen, C.-C. Hsu, W.-R. Zeng, An information granulation based data mining approach for classifying imbalanced data, Information Sciences 178 (16) (2008) 3214–3227.
[17] J. Liu, Q. Hu, D. Yu, A weighted rough set based method developed for class imbalance learning, Information Sciences 178 (4) (2008) 1235–1256.
[18] M. Wang, X.-S. Hua, T. Mei, R. Hong, G. Qi, Y. Song, L.-R. Dai, Semi-supervised kernel density estimation for video annotation, Computer Vision and Image Understanding 113 (3) (2009) 384–396.
[19] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.
[20] R. Yanagimachi, Mammalian Fertilization, 2nd ed., vol. 1, Raven Press, 1994.
[21] E. Alegre, V. González-Castro, S. Suárez, M. Castejón, Comparison of supervised and unsupervised methods to classify boar acrosomes using texture descriptors, in: Proc. Int. Symp. ELMAR '09, 2009, pp. 65–70.
[22] R. Alaiz-Rodríguez, E. Alegre, V. González-Castro, L. Sánchez, Quantifying the proportion of damaged sperm cells based on image analysis and neural networks, in: Proc. of the 8th Conference on Simulation, Modelling and Optimization, 2008, pp. 383–388.
[23] L. Sánchez, V. González, E. Alegre, R. Alaiz, Classification and quantification based on image analysis for sperm samples with uncertain damaged/intact cell proportions, in: Proc. of the 5th International Conf., ICIAR 2008, vol. 5112, Lecture Notes in Computer Science, 2008, pp. 827–836.
[24] V. González-Castro, R. Alaiz-Rodríguez, L. Fernández-Robles, R. Guzmán-Martínez, E. Alegre, Estimating class proportions in boar semen analysis using the Hellinger Distance, in: Trends in Applied Intelligent Systems, vol. 6096, Lecture Notes in Computer Science, 2010, pp. 284–293.
[25] G. Forman, Quantifying trends accurately despite classifier error and class imbalance, in: Principles and Practice of Knowledge Discovery in Databases, 2006, pp. 157–166.
[26] G. Forman, Quantifying counts and costs via classification, Data Mining and Knowledge Discovery 17 (2) (2008) 164–206.
[27] G. Forman, E. Kirshenbaum, J. Suermondt, Pragmatic text mining: minimizing human effort to quantify many issues in call logs, in: KDD '06: Proc. of the 12th Int. Conf. on Knowledge Discovery and Data Mining, 2006, pp. 852–861.
[28] A. Bella, C. Ferri, J. Hernandez-Orallo, M.J. Ramirez-Quintana, Quantification via probability estimators, IEEE International Conference on Data Mining (2010) 737–742.
[29] I. Csiszar, P. Shields, Information Theory and Statistics: A Tutorial (Foundations and Trends in Communications and Information), Now Publishers Inc., 2004.
[30] G. Forman, Counting positives accurately despite inaccurate classification, in: ECML, 2005, pp. 564–575.
[31] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1996.
[32] L. Tarassenko, N.C.A. Forum, A guide to neural computing applications, A Hodder Arnold Publication, Arnold, 1998.
[33] J.G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recognition 45 (1) (2012) 521–530.
[34] D.A. Cieslak, N.V. Chawla, A framework for monitoring classifiers performance: when and why failure occurs?, Knowledge and Information Systems 18 (1) (2009) 83–108
[35] G. Ditzler, R. Polikar, Hellinger distance based drift detection for nonstationary environments, in: 2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), 2011, pp. 41–48.
[36] A. Frank, A. Asuncion, UCI Machine Learning Repository, 2010. <http://archive.ics.uci.edu/ml>.
[37] V. González-Castro, E. Alegre, P. Morala-Argello, S.A. Suarez, A combined and intelligent new segmentation method for boar semen based on thresholding and watershed transform, International Journal of Imaging 2 (S09) (2009) 70–80.
[38] S. Arivazhagan, L. Ganesan, Texture classification using wavelet transform, Pattern Recognition Letters 24 (9–10) (2003) 1513–1521.
[39] M. González, E. Alegre, R. Alaiz, L. Sánchez, Acrosome integrity classification of boar spermatozoon images using dwt and texture techniques, in: VipIMAGE – Computational Vision and Medical Image Processing, 2007, pp. 165–168.
[40] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (1945) 80–83.
[41] J.H. Zar, Biostatistical Analysis, fifth ed., Prentice-Hall, Inc., 2007.
[42] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[43] J.G. Moreno-Torres, X. Llorà, D.E. Goldberg, R. Bhargava, Repairing fractures between data using genetic programming-based feature extraction: a case study in cancer diagnosis, Information Sciences, http://dx.doi.org/10.1016/j.ins.2010.09.018.
[44] M. Gebel, C. Weihs, Calibrating classifier scores into probabilities, in: Advances in Data Analysis, Studies in Classification, Data Analysis, and Knowledge Organization, Springer, Berlin Heidelberg, 2007, pp. 141–148.