Pattern Anal Applic (2008) 11:269–280
DOI 10.1007/s10044-007-0087-5

On the *k*-NN performance in a challenging scenario of imbalance and overlapping

V. García · R. A. Mollineda · J. S. Sánchez

Received: 6 January 2007 / Accepted: 10 July 2007 / Published online: 28 September 2007

Abstract A two-class data set is said to be imbalanced when one (minority) class is heavily under-represented with respect to the other (majority) class. In the presence of a significant overlapping, the task of learning from imbalanced data can be a very difficult problem. Additionally, if the overall imbalance ratio is different from local imbalance ratios in overlap regions, the task can become in a major challenge. This paper explains the behaviour of the *k*-nearest neighbour (*k*-NN) rule when learning from such a complex scenario. This local model is compared to other machine learning algorithms, attending to how their behaviour depends on a number of data complexity features (global imbalance, size of overlap region, and its local imbalance). As a result, several conclusions useful for classifier design are inferred.

Keywords Imbalanced data · Nearest neighbour rule · Class overlap · Local and global learning · Overall imbalance ratio · Local imbalance ratio

1 Introduction

The class imbalance problem has received considerable attention in areas such as machine learning and pattern recognition. A two-class data set is said to be imbalanced when one of the classes (the minority one) is heavily under-represented in comparison to the other class (the majority one). This issue is particularly important in real-world applications where it is costly to misclassify examples from the minority class, such as the diagnosis of rare diseases, the detection of fraudulent telephone calls, insurance claims, among others. Because of examples of the minority and majority classes usually represent the presence and absence of rare cases, respectively, they are also known as positive and negative examples.

The research in this topic has mainly focused on a number of solutions for learning from imbalanced data, which can be divided into three categories (that can also be combined):

1. Cost-sensitive learning [1–3].
2. To resample the original training set, either by over-sampling the minority class and/or under-sampling the majority class, until the classes are approximately equally represented [4, 5].
3. Internally biasing the discrimination-based process so as to compensate for the class imbalance [6, 7].

Many other studies on the behaviour of several standard classifiers in imbalance domains have shown that significant loss of performance is mainly due to skew of class distributions. However, recent investigations also suggest that there are other factors that contribute to such performance degradation, for example, size of the data set, class imbalance level, small disjuncts, density, and overlap complexity [8–12]. With respect to the latter, results on C4.5 and Fuzzy classifiers show that the overlap affects more than imbalance [11, 13]. It would be interesting to identify the degree of influence of each factor (and their interdependences) in the operation of each classifier.

V. García (✉)
Laboratorio de Reconocimiento de Patrones, Instituto
Tecnológico de Toluca. Av. Tecnológico s/n,
Metepec 52140, México
e-mail: vgarciaj@hotmail.com

R. A. Mollineda · J. S. Sánchez
Departament de Llenguatges i Sistemes Informàtics,
Universitat Jaume I. Av. Vicent Sos Baynat s/n,
12071 Castelló, Spain

The *k*-nearest neighbour (*k*-NN) [14] rule has been one of the most exhaustively analysed classifiers in the pattern recognition literature. Among other important results, experiments have concluded that *k*-NN is extremely sensitive to data complexity, mainly to imperfections in the data sets and to class density [15, 16]. It is well-known that this method requires high densities in order to achieve reliable class estimates, which is directly related to class sizes. This is the reason why *k*-NN is very sensitive to the imbalance level: as more represented (or denser) is a class, the expected *k*-NN results are better [17, 18]. Actually, this dependence is only critical in overlap regions, where decisions between two (or more) classes depends on the relation between their local densities.

In a two-class imbalance problem, where the majority class is generally the more represented in the overlap region, the *k*-NN classification results are usually better in the majority class than in the minority class. As *k* and/or imbalance increase, this effect is more accentuated along with a progressive reduction of the influence of the minority class. This is the general and expected behaviour but, is it possible to find situations where the *k*-NN classifier benefit the minority class? In this sense, is it possible to say that, due to its local nature, imbalance in the overlap region is more significant for *k*-NN performance than the volume (or size) of the overlap region? Finally, and considering that *k*-NN operation depends on local densities, can be asserted that the overall imbalance is much less important for *k*-NN than for other classifiers with a less local nature?

This paper aims to answer the previous questions through the analysis of the performances of *k*-NN classifiers in two series of experiments with opposite complexities in overlap regions: (a) the imbalance ratio in the overlap region is similar to the overall imbalance ratio (general case), and (b) the imbalance ratio in the overlap region is inverse to the overall one, that is, the minority class is denser than the majority class in the overlap region (atypical case). For such a purpose, we have designed two studies over two-class synthetic data sets with a fixed overall imbalance ratio in order to make results not dependent on this parameter.

The first experiment defines a series of data sets in which both the imbalance in the overlap region and the overall imbalance are identical while overlapping changes. This will establish a baseline to analyse the results of the next part. The second experiment operates on a different series of data sets where the minority class is locally denser than the majority class in the overlap region. In this case, two different studies were carried out: while the first one deals with moderately imbalanced data, the second data set exhibits a minority/majority ratio of 1:50 in order to verify the experimental results in much higher imbalanced scenarios. Complementary experiments were carried out with four other machine learning algorithms, with the purpose of comparing their results with those obtained from *k*-NN in the experiments described above. The analysis is addressed to demonstrate the effects of overall imbalance in their results, attending to their natures.

The structure of the paper is as follows. Section 2 presents the learning algorithms. In Section 3, we briefly describe the performance measures used to evaluate and compare classifiers. Section 4 consists of experiments on synthetic data sets and an exhaustive discussion of results. Finally, we will conclude the main remarks and outline some directions for future work in Section 5.

2 Machine learning algorithms

In this section, we briefly describe the classifiers selected for the subsequent experiments, each one presenting different inductive biases. More specifically, the algorithms are a naïve Bayes classifier, a MLP neural network, a C4.5 decision tree, a radial basis function (RBF) network, and a nearest neighbour classifier. The choice of these classifiers responds to their different properties, mainly in terms of data representation and local-versus-global learning. The aim is to analyse the relationship between class imbalance and these two learning strategies. Whereas the nearest neighbour classifier constitutes a representative of local learning, different models (a probabilistic classifier, two artificial neural networks and a decision tree) have been adopted as examples of global learning.

2.1 Naïve bayes classifier

The naïve bayes (NBS) classifier [19] is arguably one of the simplest probabilistic schemes, following from Bayesian decision theory. The model constructed by this algorithm is a set of probabilities. Each member of this set corresponds to the probability that a specific feature f_i appear in the instances of class c, i.e. $P(f_i \mid c)$. These probabilities are estimated by counting the frequency of each feature value in the instances of a class in the training set. Given a new instance, the classifier estimates the probability that the instance belongs to a specific class, based on the product of the individual conditional probabilities for the feature values in the instance.

The NBS algorithm is based on class independence, that is, all the attributes are independent given the value of the class variable. The conditional independence assumption is rarely true in most real-world applications. Despite this strong assumption, the algorithm tends to perform well in many scenarios. Experimental studies suggest that NBS tends to learn more rapidly than most induction algorithms.

2.2 Multilayer perceptron

The most popular class of multilayer feedforward networks is the multilayer perceptron (MLP) [20]. MLP usually comprises one input layer, one or more hidden layers, and one output layer. In general, input nodes correspond to features, hidden layers are used for computations and output layers correspond to the classes to be recognised. Each individual neuron is the elemental unit of each layer. It computes the weighted sum of its inputs, adds a bias term and drives the result thought a generally nonlinear activation function to produce a single output. The most common activation function is the sigmoid activation function, also used in the present study.

There are several training algorithms for MLP. The most common is the backpropagation algorithm, which takes a set of training instances for the learning process. For the given feedforward network, the weights are initialised to small random numbers. Each training instance is passed through the network and the output from each unit is computed. The target output is compared with the output computed by the network to calculate the error and this error value is fed back through the network. To adjust the weights, backpropagation uses gradient descent to minimise the squared error between the target output and the computed output. At each unit in the network, starting from the output unit and moving down to the hidden units, its error value is used to adjust weights of its connections so as to reduce the error. This process of adjusting the weights using training instances is iterated for a fixed number of times or until the error is small or cannot be reduced.

2.3 C4.5 decision tree

The C4.5 algorithm [21] employs a greedy technique to induce decision trees for classification. A decision-tree model is built by analysing training data and the model is used to classify unseen data. The nodes of the tree evaluate the existence or significance of individual features. Following a path from the root to the leaves of the tree, a sequence of such tests is performed resulting in a decision about the appropriate class of new objects.

The decision trees are constructed in a top–down fashion by choosing the most appropriate attribute each time. An information-theoretic measure is used to evaluate features, which provides an indication of the “classification power” of each feature. Once a feature is chosen, the training data are divided into subsets, corresponding to different values of the selected feature, and the process is repeated for each subset, until a large proportion of the instances in each subset belong to a single class. Decision tree induction is an algorithm that normally learns a high accuracy set of rules.

2.4 Radial basis function

The RBF [22] neural network, which has three layers, can be seen as a special class of multilayer feedforward networks. Each unit in the hidden layer employs a RBF, such as Gaussian kernel, as the activation function. The output units implement a weighted sum of hidden unit outputs. The input into an RBF network is nonlinear. The output is linear. The RBF (or kernel) is centered at the point specified by the weight vector associated with the unit. Both the positions and the widths of these kernels are learned from training instances. Each output unit implements a linear combination of these RBFs.

An RBF is trained to learn the centers and widths of the Gaussian function for hidden units, and then to adjust weights in the regression model that is used at the output unit. To learn the centers of the Gaussian functions the k-means clustering algorithm can be used that clusters the training instances to obtain k Gaussian functions for each attribute in the instance. After the parameters for the Gaussian function at the hidden units have been found, the weights from these units to the output unit are adjusted using linear regression.

Because of the combination of their nonlinear characteristics, RBF networks are commonly used in complex applications and are considered superior to MLP networks. In several practical cases, perceptrons require many neurons, computational power and time in order to calculate the hyperplanes which distinguish the problem classes.

2.5 The nearest neighbour classifier

One of the most widely studied non-parametric classification approaches corresponds to the k-NN decision rule [14]. In brief, given a set of n previously labeled examples (training set), say $\mathcal{X} = \{(x_1, \omega_1), (x_2, \omega_2), , (x_n, \omega_n)\}$, the k-NN classifier consists of assigning a new input sample $\mathbf{x}$ to the class most frequently represented among the k closest instances in the training set, according to a certain dissimilarity measure (generally, the Euclidean distance metric). A particular case is when $k = 1$, in which an input sample is decided to belong to the class indicated by its closest neighbour.

Several properties make the k-NN classifier quite attractive, including the fact that the asymptotic risk (i.e. when $n \rightarrow \infty$) converges to the optimal Bayes risk as $k \rightarrow \infty$ and $k/n \rightarrow 0$ [23]. If $k = 1$, the upper bound of the classification error rate is approximately twice the optimal Bayes error under the assumption of an infinite number of training examples [14]. The optimal behaviour of this rule in asymptotic classification performance along with a conceptual and implementational simplicity make it a powerful classification technique capable of dealing with arbitrarily complex problems, provided that there is a large enough number of training instances available.

However, in many practical situations, such a theoretical maximum can hardly be achieved due to certain inherent weaknesses that significantly reduce the effective applicability of *k*-NN classifiers. For example, the performance of these rules, as with most non-parametric classification approaches, is extremely sensitive to data complexity. In particular, class overlapping, class density, high data dimensionality, and incorrectness or imperfections in the training set can negatively affect the behaviour of these classifiers [15, 16]. Also, the class imbalance (i.e. high differences in class distributions) has been reported as an obstacle on applying distance-based algorithms to real-world problems [8, 17, 18].

Analogously, accuracy of *k*-NN classifiers significantly drops in domains where many data attributes are irrelevant [24]. Such attributes inappropriately affect the values returned by most dissimilarity metrics. On the other hand, these classifiers cannot be straightforwardly employed in domains with missing attributes [25] because most distance metrics can only be used if each example can be interpreted as a point in the feature space [26]. Another problem using the *k*-NN rule refers to the seeming necessity of a lot of memory and computational resources (especially, in applications with a huge number of training examples), since it is necessary to search the entire training set to identify the nearest neighbours to the test sample [15].

Friedman [27] establishes that the combination of the bias and variance components of the estimation error can be more significant for classification than the probabilities themselves. This analysis is supported by an evaluation of the *k*-NN classifier, showing that certain types of very high bias produced by the curse-of-dimensionality can be compensated by a low variance to produce good classification results. In particular, the experiments study class density and dimensionality in a controlled domain with a very simple decision boundary and no overlapping.

Nevertheless, it is well known that class overlapping negatively affects the performance of the *k*-NN classifiers, and this has been widely proved in many empirical studies (e.g. see [28]). Analogously, the effect of feature space dimensionality on the *k*-NN performance has also been extensively investigated in many works. For example, Beyer et al. [29] showed that under certain broad conditions, as dimensionality increases, the distance of the nearest neighbour approaches the distance of the farthest neighbour, that is, the contrast in distances to different data points becomes nonexistent.

3 Performance measures in class imbalance problems

Most of performance measures for two-class problems are built over a 2×2 confusion matrix as illustrated in Table 1. From this, four simple measures can be directly obtained: TP and TN denote the number of positive and negative cases correctly classified, while FP and FN refer to the number of misclassified positive and negative examples, respectively.

The most widely used metrics for measuring the performance of learning systems are the error rate and the accuracy, which can be computed as (TP + TN)/(TP + FN + TN + FP). Nevertheless, researchers have demonstrated that, when the prior class probabilities are very different, these measures are not appropriate because they do not consider misclassification costs, are strongly biased to favor the majority class, and are sensitive to class skews [30–33].

Thus, several metrics that measure the classification performance on positive and negative classes independently can be derived from Table 1. The true positive rate, also referred to as recall or sensitivity, TP rate = TP/(TP + FN), is the percentage of correctly classified positive examples. The *true negative rate* (or specificity), TN rate = TN/(TN + FP), is the percentage of correctly classified negative examples. The *false positive rate*, FP rate = FP/(FP + TN) is the percentage of misclassified positive examples. The *false negative rate*, FN rate = FN/(TP + FN) is the percentage of misclassified negative examples. Finally, the precision (or purity), Precision = TP/(TP + FP), is defined as the proportion of positive cases that are actually correct.

A way to combine the TP and FP rates is by using the ROC curve. The ROC curve is a two-dimensional graph to visualise, organise and select classifiers based on their performance. It also depicts trade-offs between benefits (true positives) and costs (false positives) [30, 34]. In the ROC curve, the TP rate is represented on the *Y*-axis and the FP rate on the *X*-axis. Several points on a ROC graph should be noted. The lower left point (0, 0) represents that the classifier labeled all examples as negative the upper right point (1, 1) is the case where all examples are classified as positive, the point (0, 1) represents perfect classification, and the diagonal line $y = x$ defines the strategy of randomly guessing the class. To assess the overall performance of a classifier, one can measure the

Table 1 Confusion matrix for a two-class problem

	Predicted positive	Predicted negative
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

fraction of the total area that falls under the ROC curve (AUC) [31]. AUC varies between 0 and +1. Larger AUC values indicate generally better classifier performance.

Kubat and Matwin [5] use the geometric mean (*g*-mean) of accuracies measured separately on each class, $\text{g-mean} = \sqrt{\text{recall} \times \text{specificity}}$. This measure relates to a point on the ROC curve and the idea is to maximise the accuracy on each of the two classes while keeping these accuracies balanced. An important property of the g-mean is that it is independent of the distribution of examples between classes. Another property is that it is nonlinear, that is, a change in recall (or specificity) has a different effect on this measure depending on the magnitude of recall (or specificity). An alternative metric that does not take care of the performance on the majority class corresponds to the geometric mean of precision and recall, which is defined as $\text{gpr} = \sqrt{\text{precision} \times \text{recall}}$. Like the g-mean, this measure is higher when both precision and recall are high and balanced.

4 Experimental results and discussion

In this section, a number of experiments on two series of artificial data sets, whose characteristics can be fully controlled, are carried out and discussed. Pseudo-random bivariate patterns have been generated following a uniform distribution in a square of length 100. There are 400 negative examples and 100 positive patterns in all cases, keeping the overall majority/minority ratio equal to 4. It should be pointed out, although only one-dimension appears as discriminant, the inclusion of two-dimensions is with the aim of making easier the interpretation of the results in overlap regions of greater sizes (or volumes).

The experiments are divided into two scenarios. The first constitutes a typical class imbalance problem with overlapping, in the sense that imbalance equally affects to the whole representation space. The second one refers to a more challenging situation, where the imbalance ratio in the overlap region is inverse to the overall imbalance ratio, that is, the majority and minority classes have interchanged their roles. We expect that this singular case allows us to better explain the dependence of *k*-NN on overall imbalance, local imbalance in overlap region, and the size of overlap region. A third experiment using a group of artificial data sets with a much higher imbalance ratio (1:50) is included just to corroborate the conclusions obtained in the second scenario.

We have adopted a tenfold cross-validation method: each data set was divided into ten equal parts, using ninefolds as the training set and the remaining block as an independent test set. This process has been repeated ten times, that is, tentimes tenfold cross-validation and average the results. The experiments consist of computing the performance metrics reported in Section 3, when using several classifiers of distinct natures: a *k*-NN classifier, a MLP, a NBS classifier, a RBF, and a C4.5 decision tree, taken all of them of WEKA toolkit [35].

4.1 Experiment I

The first experiment has been over a collection of six data sets with increasing class overlap. In all cases, the positive examples are defined on the *X*-axis in the range [50–100], while those belonging to the majority class are generated in [0–50] for 0% of class overlap, [10–60] for 20%, [20–70] for 40%, [30–80] for 60%, [40–90] for 80%, and [50–100] for 100% of overlap. Note that the overall imbalance ratio matches the imbalance ratio corresponding to the overlap region, what could be accepted as a common case. Their results should establish a baseline, useful to better interpret the results of the second series. Figure 1 illustrates two examples of these data sets.

4.1.1 Behaviour of k*-NN algorithm*

This section is limited to analyse the behaviour of *k*-NN, which has been defined for values of *k* ranging from 1 to 15 and for the Euclidean distance. For the sack of simplicity

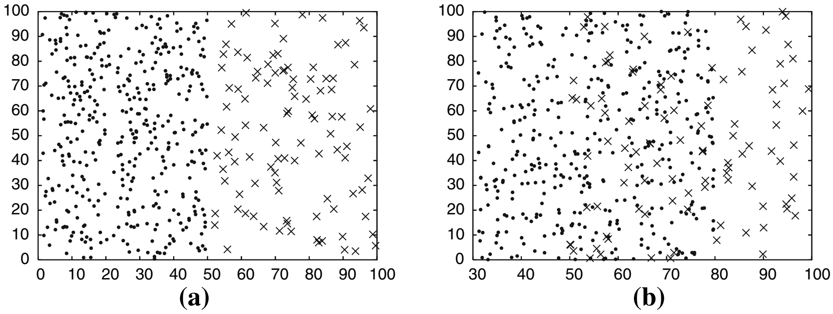

Fig. 1 Two different levels of class overlapping: **a** 0% and **b** 60%

and clarity, we have included only the values of 1, 3, 7, 9 and 13 in the figures.

Figure 2 shows six performance measures for all versions of *k*-NN while overlap increases. From the first two measures, TP rate and TN rate, two main issues can be remarked. First, when overlapping increases, the recognition rate of positive examples drops significantly faster than that of the negative examples (for any fixed *k*). Second, as *k* increases (being the rule less local), the TP rate also drops while TN rate raises to nearly 100% (for any fixed degree of overlapping).

With respect to the first issue, as overlapping increases, the amount of examples from both classes in the confusion region increases, but keeping the difference between densities (negative class is denser than positive class). Thus, it is much more likely that *k*-NN missclassifies positive examples than negative ones. Attending to the second issue, with the increase of *k*, the local volume to make a decision becomes larger, along with the probability to find neighbourhoods with a majority of negative examples and to missclasify positive examples.

As a preliminary conclusion, it can be said the increase in overlapping or in the value of *k* in the presence of a homogeneous imbalance affects more the (overall) minority class. It is also worth mentioning that this wide range of results was obtained for a fixed ratio of imbalance, what seems to suggest that overall imbalance by itself does not necessarily affects the *k*-NN operation.

Attending to global performance measures about *k*-NN (see Fig. 2 (c-f)), three of them (Precision, gpr and AUC) produce better results for higher values of *k* (for almost all degrees of overlapping). In other words, for these three measures, the greater the value of *k*, the better the *k*-NN classifier is. However, for the geometric mean (g-mean) between the two class recognition rates TPR and TNR, the 1-NN is the most discriminant scheme. The reason is that TPR of 1-NN is slightly better with respect to other *k*-NN than TNR of *k*-NN with respect to 1-NN. But just relatively, because a 1% of TPR corresponds to one positive example, while the same fraction of TNR corresponds to four negative examples. In this sense, the plain accuracy, computed as the weighted mean of TNR and TPR, clearly confirms that *k*-NN is a better classifier as *k* increases (for any overlapping).

4.1.2 k-*NN versus other machine learning approaches*

In this section, *k*-NN is compared to four other machine learning algorithms (Section 2), when classifying the same series of imbalance data sets. These four methods are MLP, naïve Bayes, RBF, and C4.5. For *k*-NN, only three values of *k* (1, 3, 9) have been used because they seem sufficient to show the trends of this local scheme.

Results are conceptually similar to those of the previous section: the more local schemes tend to be better at classifying the minority class (particularly for higher overlapping), while models based on a more global learning are more robust at classifying the majority class (Fig. 3a, b). In more details, the 1-NN rule clearly achieves higher recognition rates on positive examples than other models in the three more overlapped scenarios. Meanwhile, MLP, which performs a global learning, appears as the best classifiers on the negative class. In this partial task, *k*-NN accuracy increases along with the *k* value, that is, as *k*-NN becomes less local.

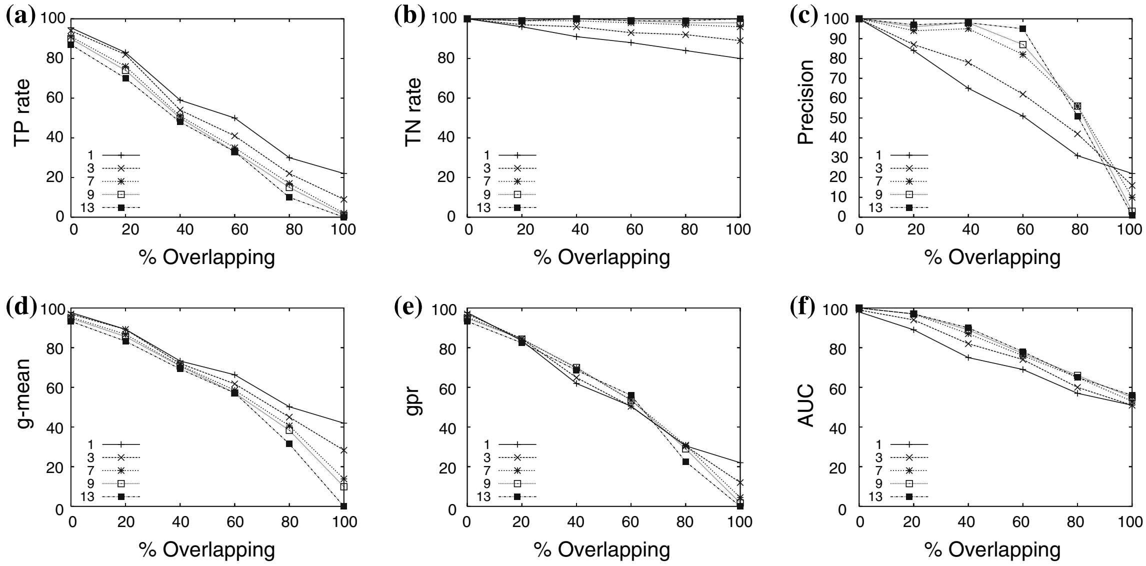

Fig. 2 Performance metrics in *k*-NN rule for Experiment I: **a** TP rate, **b** TN rate, **c** Precision, **d** g-mean, **e** gpr and **f** AUC

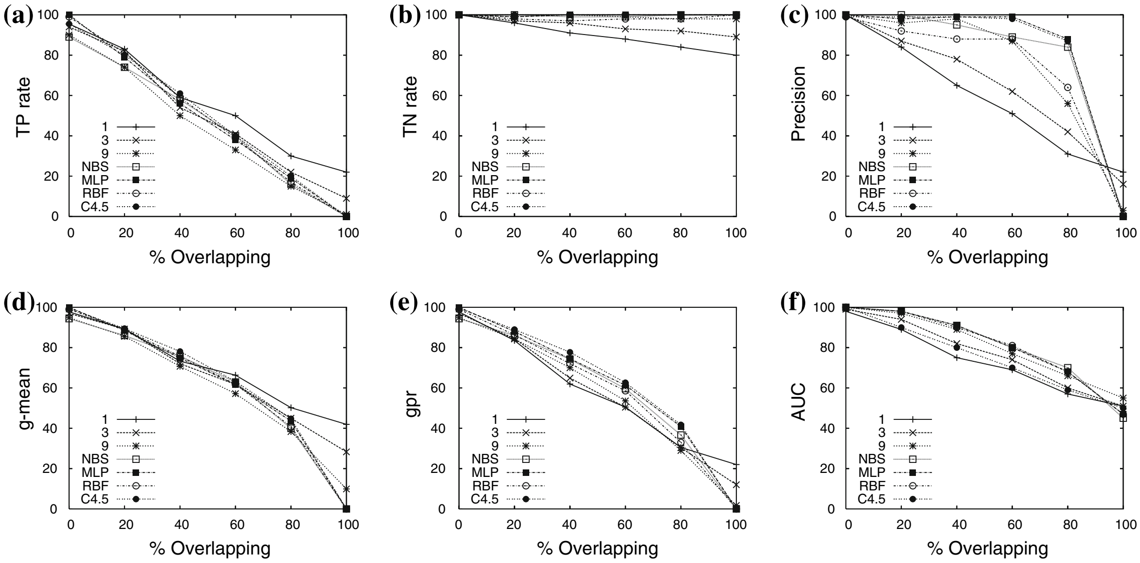

Fig. 3 Performance metrics in *k*-NN rule and other learning algorithms for Experiment I: **a** TP rate, **b** TN rate, **c** Precision, **d** g-mean, **e** gpr and **f** AUC

Because of TN rates are significantly higher than TP rates for all classifiers, the methods that better classify the negative class are also those which generally produce higher overall performances. They are the classifiers with less local learning (Fig. 3c, e, f). In particular, MLP seems to be the most consistent classifier while 1-NN is the one with worst performance for these three measures. The exception is again g-mean (Fig. 3d), where 1-NN overcomes the rest of classifiers in the more overlapped scenarios. Note that these results are obtained with measures which are appropriated for imbalance problems. In these sense, the use of general-purpose plain accuracy conducts to the same results but with more remarked distances among classifiers.

As a second preliminary conclusion, it can be said that the majority class tends to be better discriminated by classifiers based on a global learning, while the minority class tends to be better discriminated by local classifiers. Note that, as in the previous section, this conclusion is limited to situations with homogeneous imbalance.

4.2 Experiment II

The second experiment has been carried out over a collection of five artificial imbalanced data sets in which the overall minority class becomes the majority in the overlap region. To this end, the 400 negative examples have been defined on the *X*-axis to be in the range [0–100] in all data sets, while the 100 positive cases have been generated in the ranges [75–100], [80–100], [85–100], [90–100], and [95–100]. The number of elements in the overlap region varies from no local imbalance in the first case, where both classes have the same (expected) number of patterns and density, to a critical inverse imbalance in the fifth case, where the 100 minority examples appears as majority in the overlap region along with about 20 expected negative examples. Figure 4 illustrates two examples of these data sets. Summarising: the overall minority class (the positive one) turns the most represented class in the overlapped region. Situations like this one, in which overall imbalance configuration is different from those of some local

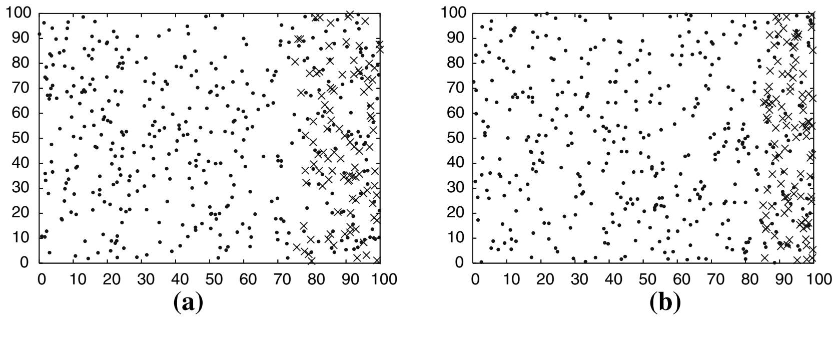

Fig. 4 Two different cases in experiment II: [75–100] and [85–100]. For this latter case, note that in the overlap region, the majority class is under-represented in comparison to the minority class

region(s), will be referred to as heterogeneous imbalance. The aim of this experiment is to analyse the importance of the overall imbalance ratio, the size of the overlap region, and its local imbalance ratio, in the behaviour of a variety of machine learning algorithms when dealing with this challenging task.

4.2.1 *Behaviour of* k-*NN algorithm*

This section is devoted to study only the *k*-NN behaviour on such atypical imbalanced scenario. The *k* values and the distance are those used in Section 4.1.1.

Figure 5 shows six performance measures for all versions of *k*-NN as a function of overlapping (from greater to lower size of the overlap region). By focusing our attention on the first two measures, TP rate and TN rate, two main observations can be made. First, as *k* increases, the recognition rate of positive examples raises for the three most significant overlapping cases while the recognition rate of negative examples slightly drops for most of cases. Second, as the size of overlap region decreases along with the increase of the density of positive examples and the decrease of the number of negative examples in such region, the TP rate raises while TN rate keeps almost constant.

With respect to the first observation, unlike the Experiment I with homogeneously imbalanced data sets, the increase of *k* benefits the overall minority class. However, like the Experiment I, this class is the more represented in the overlap region. The opposite analysis can be performed with the overall majority class. It confirms that *k*-NN is much more dependent on the local imbalance in the overlap region than on the overall imbalance.

With respect to the second observation, the reduction of the size of the overlap region affects both classes in different ways. The density of the positive class increases because the overlap region encloses all positive examples, while the density of negative class keeps constant due to its unchanging distribution. As a result, *k*-NN improves its accuracy on positive class (the overall minority one), while produces almost-stable accuracy curves on negative class (the majority one), for all *k* value. The explanation of the result on positive class is straightforward: the increase of density of the positive class with respect to the negative class, raises the probability of correct *k*-NN decision on positive examples. In the case of results on the negative class, each new reduction of the overlapping size leaves more negative examples out of the overlap region and a less number of negative examples coexisting with the whole positive class. Most of the former will be well classified while most of the latter have an increasing probability to be misclassified. Both effects compensate each other leading to almost-stable accuracy curves, which are clearly flat in the three central overlapping cases (85, 90 and 95–100) where the size of the overlap region is reduced by half.

This interesting result could have been also achieved by fixing the overlapping region with all the positive examples and appropriately putting/removing negative examples to/from the overlap region (keeping both classes uniformly distributed). In other words, this (constant) *k*-NN behaviour can be obtained without any dependence on the overlapping size. This example seems to reveal that, when the overlapped data is not balanced, the imbalance ratio in overlapping can be more important for *k*-NN performance than the overlapping size.

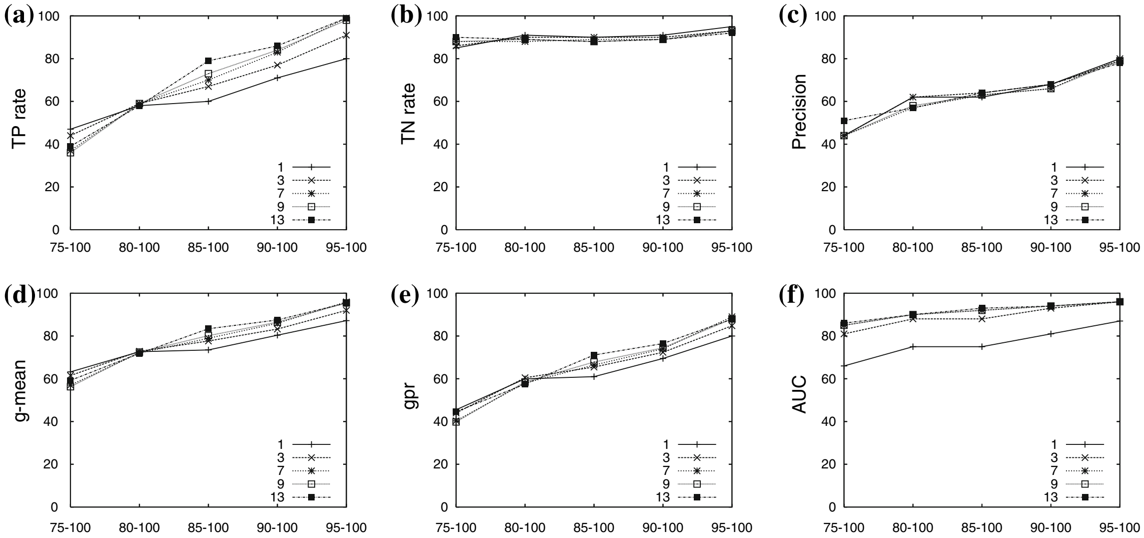

Fig. 5 Performance metrics in *k*-NN rule for Experiment II: **a** TP rate, **b** TN rate **c** Precision, **d** g-mean, **e** gpr and **f** AUC

Finally, three global performance measures, g-mean, gpr, and AUC (Fig. 5c–f), show, as a tendency, better performances as *k* increases for most of overlapping cases. A review of previous discussion and a close look to TN and TP rates at Fig. 5a, b, allow to explain it.

4.2.2 k-*NN versus other machine learning approaches*

As in Section 4.1.2, 1, 3, 9-NN are compared to MLP, naïve Bayes, RBF, and C4.5, when classifying the heterogeneously imbalanced data sets.

In the previous experiments, the 1-NN, the most local version of *k*-NN, has been the best classifier for the class less represented in the overlap region, that were the overall minority class in Section 4.1, and the overall majority class in Section 4.2.1. Meanwhile, the classifiers based on more global learning have been the most discriminant for the class more represented in the overlap region, in particular, the overall majority class in Section 4.1 and the overall minority class in Section 4.2.1.

New results of this section agree with the previous summary. The classifiers based on more global learning, MLP, NB, and C4.5, attain greater TP rates (Fig. 6a) than *k*-NN family and RBF, while the latter, the models based on a more local learning, obtain better TN rates than the former (Fig. 6b). These results are more visible for the three central overlapping cases (85, 90 and 95–100), which exclude the two extreme cases (no imbalance and very intense imbalance).

Because the greater differences between global-learning and local-learning classifiers are in TP rates with respect to TN rates, those with higher TP rates are the ones with higher values in the overall performance measures (Fig. 6 d–f). In particular, MLP, NB and C4.5 are notably better than *k*-NN family considering the measures g-mean and gpr, whereas all of them are very close in the two other overall measures, precision and AUC.

In Section 4.1.2, where a similar experiment was performed on the series of homogeneously imbalanced data sets, a preliminary conclusion interrelated the learning global/local nature of classifiers with the overall majority/minority condition of classes. That conclusion is not valid in the presence of a heterogeneous imbalance, as previous analysis has demonstrated. A more general conclusion should connect learning nature and local majority/minority condition in overlap regions. In this sense, it can be said that the class more represented in overlap regions tends to be better classified by methods based on global learning, while the class less represented in such regions tends to be better classified by methods based on local learning.

4.3 Experiment III

The third experiment has been carried out over a collection of five artificial imbalanced data sets, following the idea of the experiments in Section 4.2. In this case, two concentric classes have been randomly generated, both centred at [0, 0]. The majority class consists of 5,000 examples lying within a radius equal to 50 in all data sets. The 100 positive cases were defined considering a different radius (0.20, 0.40, 0.60, 0.80, 1.00) for each data set. Thus, in the case of a radius equal to 1.00, there exists approximately the same number of positive and negative examples in the overlap

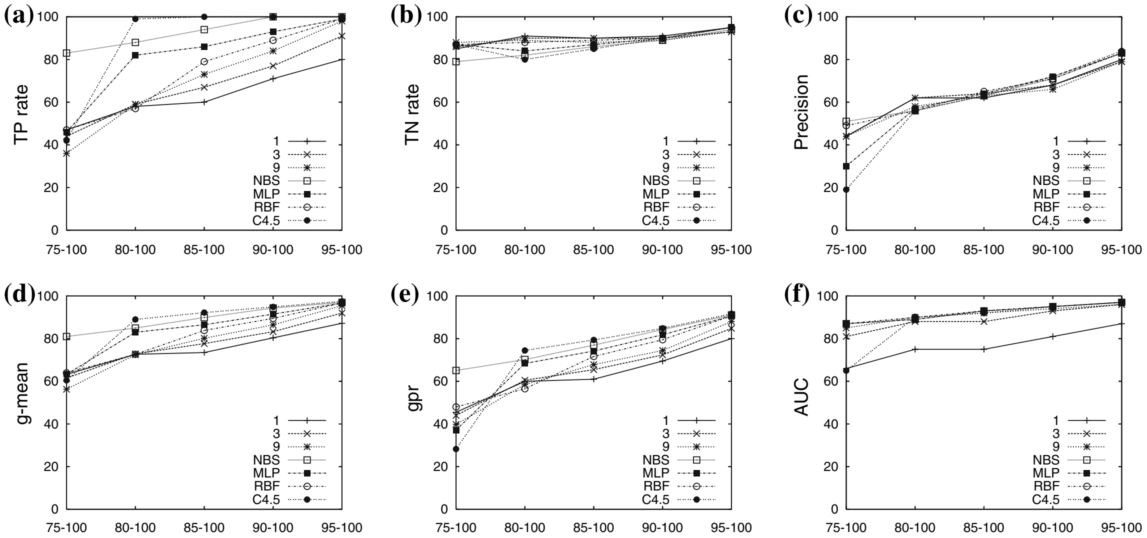

Fig. 6 Performance metrics in *k*-NN rule and other learning algorithms for Experiment II: **a** TP rate, **b** TN rate, **c** Precision, **d** g-mean, **e** gpr and **f** AUC

region (in all data sets, this is defined by the minority class).

As already mentioned, the aim of this experiment is to validate the results obtained in Section 4.2, but in a much higher imbalanced scenario. Figure 7 compares several learning algorithms using the performance measures employed in previous sections.

Considering the *k*-NN classifiers, the results of TP and TN rates confirm that the local imbalance has a more important influence than the overall imbalance. Note that classification of positive examples improves as the value of *k* increases in data sets where the minority class is denser than the majority class in the overlap region (radius equal to 0.20, 0.40 and 0.60).

In the case of of MLP, RBF, NBS and C4.5, the results are similar to those given in Section 4.2, although with some differences. For example, the MLP misclassifies the positive cases in all databases, thus showing low values for precision, g-mean, gpr and AUC. This behaviour can be due to the complexity of the boundary decision and also to the high imbalance ratio. On the other hand, NBS and C4.5 generally achieve performance rates greater than *k*-NN and RBF.

5 Conclusions

The class imbalance problem have been mostly studied from an homogeneous perspective, that is, assuming an uniform imbalance ratio over the whole distribution of data. This paper designs challenging two-class data sets where the imbalance ratio in the overlap region is inverse to the overall imbalance ratio. That is, the overall majority and minority classes have interchanged their roles in the overlap region.

Given such atypical scenario, the analysis was primarily addressed to find out the relative significance for *k*-NN behaviour of three data complexity features: the overall imbalance ratio, the size of overlap region, and its local imbalance ratio. According to experiment results, *k*-NN behaviour seems to be more dependent on changes in the local imbalance ratio in the overlap region, than on changes in the size of the overlap region. Experiments also confirm that such a local imbalance ratio and the size of the overlap region are more important than the overall imbalance ratio. Identical behaviour has been observed in the case of highly imbalanced data.

The *k*-NN rule was also compared to four other machine learning algorithms. Experiments focused on the relation between the majority/minority condition of classes and the global/local nature of classifiers. Results show that the class more represented in overlap regions tends to be better classified by methods based on global learning, while the class less represented in such regions tends to be better classified by local methods. In this sense, as the value of *k* of the *k*-NN rule increases, along with a weakening of its local nature, it was progressively approaching the behaviour of global models.

This complementarity between global and local classifiers suggests a direction for future works on learning from imbalance data. It consists in the design of multiclassifier systems defined by a proper combination of both types of classifiers, what should be more suitable for managing imbalance than individual schemes.

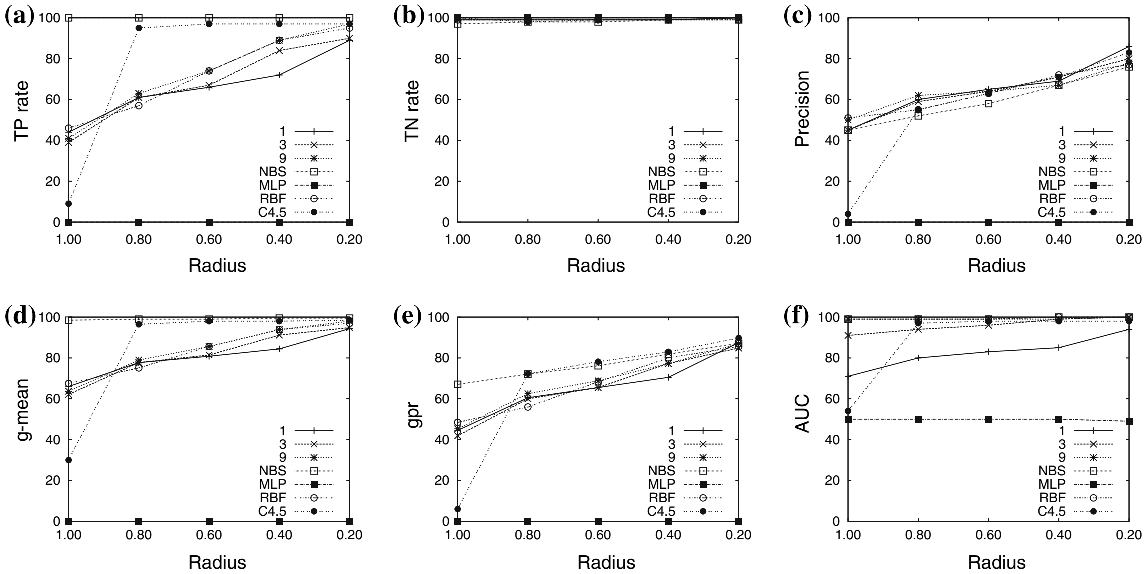

Fig. 7 Performance metrics in *k*-NN rule and other learning algorithms for Experiment III: **a** TP rate, **b** TN rate, **c** Precision, **d** g-mean, **e** gpr and **f** AUC

6 Originality and contribution

The originality of this paper consists of a comparative analysis of the behaviours of *k*-NN and four other classifiers (of different global/local natures), when managing a challenging data set in which the overall imbalance is inverse to local imbalance in overlapping. The contribution is a number of interesting conclusions about the relative importance of imbalance and overlapping in the behaviour of the machine learning algorithms involved, and ideas on how to use some of them in classifier design.

Acknowledgments This work has been partially supported by grants DPI2006-15542 from the Spanish CICYT, CSD2007-00018 from the Spanish Ministry of Science and Education and SEP-2003-C02-44225 from the Mexican CONACyT.

References

1. Domingos P (1999) Metacost: a general method for making classifiers cost-sensitive. In: Proceedings of the 5th international conference on knowledge discovery and data mining, pp 155–164
2. Gordon DF, Perlis D (1989) Explicitly biased generalization. Comput Intell 5:67–81
3. Pazzani M, Merz C, Murphy P, Ali K, Hume T, Brunk C (1994) Reducing misclassification costs. In: Proceedings 11th international conference on machine learning, pp 217–225
4. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
5. Kubat M, Matwin S (1997) Adressing the curse of imbalanced training sets: one-sided selection. Proceedings of the 14th international conference on machine learning, pp 179–186
6. Barandela R, Sánchez JS, García V, Rangel E (2003) Strategies for learning in class imbalance problems. Pattern Recognit 36:849–851
7. Fawcett T, Provost F (1996) Adaptive fraud detection. Data Mining Knowl Discov 1:291–316
8. Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intell Data Anal J 6(5):429–450
9. Jo T, Japkowicz N (2004) Class imbalances versus small disjuncts. SIGKDD Explor 6:40–49
10. Weiss GM (2003) The effect of small disjuncts and class distribution on decision tree learning. PhD thesis, Rutgers University
11. Prati RC, Batista GE, Monard MC (2004) Class imbalance versus class overlapping: an analysis of a learning system behavior. In: Proceedings of the 3rd Mexican international conference on artificial intelligence, pp 312–321
12. Orriols A, Bernardó E (2005) The class imbalance problem in learning classifier systems: a preliminary study. In: Proceedings of conference on genetic and evolutionary computation, pp 74–78
13. Visa S, Ralescu A (2003) Learning from imbalanced and overlapped data using fuzzy sets. In: Proceedings of ICML-2003 workshop: learning with imbalanced data sets II, pp 97–104
14. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13:21–27
15. Dasarathy BV (1991) Nearest neighbor norms: NN pattern classification techniques. IEEE Computer Society Press, Los Alamos
16. Devijver PA, Kittler J (1992) Pattern recognition: a statistical approach. Prentice Hall, Englewood Cliffs
17. Hand DJ, Vinciotti V (2003) Choosing *k* for two-class nearest neighbour classifiers with unbalanced classes. Pattern Recognit Lett 24:1555–1562
18. Zhang J, Mani I (2003) kNN approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of workshop on learning from imbalanced datasets II, pp 42–48
19. Duda RO, Hart PE, Stork DG (2001) Pattern classification and scene analysis. Wiley, New York
20. Bishop C (1995) Neural networks for pattern recognition. Oxford University Press, USA
21. Quinlan JR (1993) C4.5: Programs for machine learning. Morgan Kaufmann, San Mateo
22. Buhmann M, Albowitz M (2003) Radial basis functions: theory and implementations. Cambridge University Press, USA
23. Cover TM (1968) Estimation by the nearest neighbor rule. IEEE Trans Inf Theory 14:50–55
24. Okamoto S, Yugami N (2003) Effects of domain characteristics on instance-based learning algorithms. Theor Comput Sci 298:207–233
25. Little RJA, Rubin DB (2002) Statistical analysis with missing data. Wiley, New York
26. Kubat M, Chen WK (1998) Weighted projection in nearest-neighbor classifiers. In: Proceedings of 1st southern symposium on computing, pp 27–34
27. Friedman JH (1997) On bias, variance, 0/1-loss, and the curse of dimensionality. Data Mining Knowl Discov 1:55–77
28. Sánchez JS, Barandela R, Marqués AI, Alejo R, Badenas J (2003) Analysis of new techniques to obtain quality training sets. Pattern Recognit Lett 24:1015–1022
29. Beyer KS, Goldstein J, Ramakrishnan R, Shaft U (1999) When is "nearest neighbor" meaningful?. In: Proceedings of 7th international conference on database theory, pp 217–235
30. Provost F, Fawcett T (1997) Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In: Proceedings of 3rd international conference on knowledge discovery and data mining, pp 43–48
31. Huang J, Ling CX (2005) Using AUC and accuracy in evaluating learning algorithms. IEEE Trans Knowl Data Engng 17:299–310
32. Daskalaki S, Kopanas I, Avouris N (2006) Evaluation of classifiers for an uneven class distribution problem. Appl Artif Intell 20:381–417
33. Landgrebe TCW, Paclick P, Duin RPW (2006) Precision-recall operating characteristic (P-ROC) curves in imprecise environments. In: Proceedings of 18th international conference on pattern recognition, pp 123–127
34. Fawcett T (2006) ROC graphs with instance-varying costs. Pattern Recognit Lett 27:882–891
35. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques. Morgan Kaufmann, USA

Author Biographies

Vicente García Jiménez is currently a PhD student at the Universitat Jaume I (Castelló de la Plana, Spain) and the Instituto Tecnológico de Toluca (México) since 2005. He received a BSc in Computer Science from the Instituto Tecnológico de Villahermosa (México), in 2000 and a MSc in Computer Science from the Instituto Tecnológico de Toluca in 2002. His current research interests lie in the areas of pattern recognition and machine learning, including nonparametric classifications, ensembles of classifiers and clustering.

Ramón A. Mollineda Cárdenas is an Associate Professor in the Department of Programming Languages and Information Systems at Universitat Jaume I (Castelló de la Plana, Spain) since 2003, and currently belongs to the Pattern Analysis and Learning Group. He received a BSc in Computer Science from the Universidad Central de Las Villas, Cuba, in 1995 and a PhD in Computer Science from the Universidad Politécnica de Valencia in 2001. He is a member of IAPR and AERFAI (Spanish Association of Pattern Recognition and Image Analysis). He is author or co-author of more than 30 scientific publications, and he has been a member of review committees of several journals and conferences. His current research interests lie in the areas of pattern recognition and machine learning, including nonparametric classification, ensembles of classifiers, data analysis and string matching.

J.S. Sánchez is an Associate Professor in the Department of Programming Languages and Information Systems at Universitat Jaume I (Castelló de la Plana, Spain) since 1992, and he is currently the head of the Pattern Analysis and Learning Group. He received a BSc in Computer Science from the Universidad Politécnica de Valencia in 1990 and a PhD in Computer Science Engineering from Universitat Jaume I in 1998. He is the author or co-author of more than 100 scientific publications, co-editor of two books and guest editor of several special issues in international journals. He is a member of IEEE and IAPR. He serves as an Associate Editor for the Pattern Analysis and Applications Journal. His current research interests lie in the areas of pattern recognition and machine learning, including nonparametric classification, feature and prototype selection, ensembles of classifiers, and clustering.