International Journal on Artificial Intelligence Tools


# A Nearest Neighbor Algorithm for Imbalanced Classification

Rémi Viola[1,2], Rémi Emonet[1], Amaury Habrard[1], Guillaume Metzler[1], Sébastien Riou[2] and Marc Sebban[1]

[1] *Hubert Curien Laboratory UMR 5516,*
*University of Lyon, UJM-Saint-Etienne, CNRS,Institute of Optics Graduate School,*
*Hubert Curien Laboratory UMR 5516, Saint-Etienne, France*
*first-name.name@univ-st-etienne.fr*

[2] *Direction Générale des Finance Publiques,*
*French Ministry of Economy and Finance, Paris, France*
*remi.viola@dgfip.finances.gouv.fr; sebastien-2.riou@dgfip.finances.gouv.fr*

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Due to the inability of the accuracy-driven methods to address the challenging problem of learning from imbalanced data, several alternative measures have been proposed in the literature, like the Area Under the ROC Curve (AUC), the Average Precision (AP), the F-measure, the G-Mean, etc. However, these latter measures are neither smooth, convex nor separable, making their direct optimization hard in practice. In this paper, we tackle the challenging problem of imbalanced learning from a nearest-neighbor (NN) classification perspective, where the minority examples typically belong to the class of interest. Based on simple geometrical ideas, we introduce an algorithm that rescales the distance between a query sample and any positive training example. This leads to a modification of the Voronoi regions and thus of the decision boundaries of the NN classifier. We provide a theoretical justification about this scaling scheme which inherently aims at reducing the False Negative rate while controlling the number of False Positives. We further formally establish a link between the proposed method and cost-sensitive learning. An extensive experimental study is conducted on many public imbalanced datasets showing that our method is very effective with respect to popular Nearest-Neighbor algorithms, comparable to state-of-the-art sampling methods and even yields the best performance when combined with them.

*Keywords*: Machine Learning; Nearest Neighbor Algorithm; Imbalanced Classification.

## 1. Introduction

While the machine learning community can benefit nowadays from larger and larger datasets for optimizing provably accurate classifiers, many real world applications still suffer from a lack of data, especially in imbalanced learning, where the positive examples are very scarce compared with the number of negative samples [1–3]. This is typically the case in intrusion detection, health care insurance or bank fraud identification, and more generally anomaly detection, *e.g.*, in medicine or in industrial

processes. In such a setting, the training set is composed of a few positive examples (*e.g.*, the frauds) and a huge amount of negative samples (*e.g.*, the genuine transactions). Standard learning algorithms struggle to deal with this imbalance scenario because they are typically based on the minimization of (a surrogate of) the 0-1 loss. Therefore, a trivial solution consists in assigning the majority label to any test query, leading to a high performance from an accuracy perspective but completely missing the (positive) examples of interest. To overcome this issue, several strategies have been developed over the years. The first one consists in the optimization of loss functions based on measures that are more appropriate for this context such as the *Area Under the ROC Curve* (AUC), the *Average Precision* (AP), the *G-mean* (GM), the Balanced-Accuracy (BA) or the *F-measure* to cite a few [4–6]. The main pitfalls related to such a strategy concern the difficulty to directly optimize non smooth, non separable and non convex measures (see [7] for the specific case of the *F-measure*). A simple and usual solution to fix this problem consists in using off-the-shelf learning algorithms (maximizing the accuracy) and a posteriori pick the model with the highest AP, GM, BA or *F-measure*. Unfortunately, this might be often suboptimal. A more elaborate solution aims at designing differentiable versions of the previous non-smooth measures and optimizing them, *e.g.*, as done by gradient boosting in [8] with a smooth surrogate of the Mean-AP. The second family of methods is based on the modification of the distribution of the training data using sampling strategies [9]. This is typically achieved by removing examples from the majority class, as done, *e.g.*, in *ENN* or *Tomek's Link* [10], and/or by adding examples from the minority class, as in SMOTE [11] and its variants, or by resorting to generative adversarial models [12]. One peculiarity of imbalanced learning can be interpreted from a geometric perspective. As illustrated in Fig. 1 (left) which shows the Voronoi cells on an artificial imbalanced dataset (where two adjacent cells have been merged if they concern examples of the same class), the regions of influence of the positive examples are much smaller than that of the negatives. This explains why at test time, in imbalanced learning, the risk to get a false negative (*e.g.*, a fraud that is wrongly classified as a genuine transaction) is high. A large number of false negatives ($FN$) leads to a dramatic decrease of the aforementioned measures that all rely on a fine balance between $FN$ and the number of false positives $FP$, building blocks of the so-called $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$ where $TP$ is the number of true positives. Note that increasing the regions of influence of the positives would mechanically reduce $FN$. However, not controlling the expansion of these regions, as illustrated in Fig. 1 (right), may have a dramatic impact on $FP$, and thus on the previous performance measures.

The main contribution of this paper is about the problem of finding the appropriate trade-off (Fig. 1 (middle)) between the two above-mentioned extreme situations (large $FP$ or $FN$, both leading to a poor performance at test time). A natural way to increase the influence of positives may consist in using generative models (like GANs [12]) to sample new artificial examples, mimicking the negative training samples. However, beyond the issues related to the parameter tuning, the computation

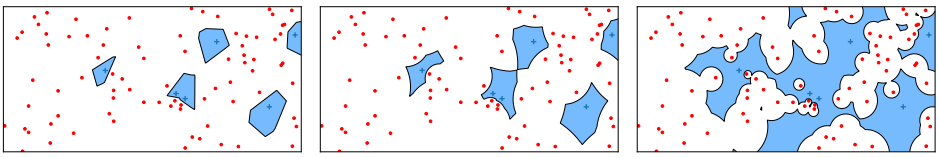

Fig. 1. Toy imbalanced dataset: On the left, the Voronoi regions around the positives are small. The risk to generate false negatives (FN) at test time is large. On the right: by increasing too much the regions of influence of the positives, the probability to get false positives (FP) grows. In the middle: an appropriate trade-off between the two previous situations.

burden and the complexity of such a method, using GANs to optimize the precision and recall is still an open problem (see [13] for a recent study on this topic). We show in this paper that a much simpler strategy can be used by modifying the distance exploited in a $k$-nearest neighbor ($k-$NN) algorithm [14] which enjoys many interesting advantages, including its simplicity, its capacity to approximate asymptotically any locally regular density, and its theoretical rootedness [15–17]. $k-$NN also benefited from many algorithmic advances during the past decade in the field of metric learning, aiming at optimizing under constraints the parameters of a metric, typically the Mahalanobis distance, as done in LMNN [18] or ITML [19] (see [20, 21] for a survey). Unfortunately, existing metric learning methods are dedicated to enhance the $k-$NN accuracy and do not focus on the optimization of criteria, like the F-measure or G-mean, in scenarios where the positive training examples are scarce. A geometric solution to increase, at a very low cost, the region of influence of the minority class consists in modifying the distance when comparing a query example to a positive training sample. More specifically, we formally show in this paper that the optimization of any ($FN$,$FP$)-based performance measure, which are well suited to deal with imbalanced scenarios, is facilitated by scaling the distance to any positive by a coefficient $\gamma \in [0, 1]$ leading to the expansion of the Voronoi cells around the minority examples. An illustration is given in Fig. 1 (middle) which might be seen as a good compromise that results in the reduction of $FN$ while controlling the risk to increase $FP$. Note that our strategy boils down to modifying the local density of the positive examples. For this reason, we claim that it can be efficiently combined with SMOTE-based sampling methods whose goal is complementary and consists in generating examples on the path linking two (potentially far) positive neighbors. Our experiments will confirm this intuition.

This paper improves substantially on our previous work [22], both with increased details and new algorithmic and experimental contributions: (i) we show an explicit link between the proposed method, called $\gamma k-$NN, and cost-sensitive learning, (ii) we present a local version of our method that uses clustering to adapt the parameters to the different regions of the input space, and (iii) we rework and extend the experimental study to incorporate new performance measures and to give a qualitative analysis on the well-known image dataset MNIST.

The rest of the paper is organized as follows. Section 2 is dedicated to the

introduction of our notations and an overview of the main performance measures that will be used to evaluate the compared methods. The related work is presented in Section 3. Section 4 is devoted to the presentation of our method $\gamma k-$NN. This section includes a theoretical analysis of our method as well as a presentation of a local extension aiming at capturing local specificities of the feature space. We finally establish a link between $\gamma k-$NN and cost-sensitive learning. The last part of this paper is dedicated to an extensive experimental study performed on 28 imbalanced datasets (see Section 5). In this comparative analysis, we give evidence of the complementarity of our method with sampling strategies. We finally conclude in Section 6.

## 2. Notations and Evaluation Measures

We consider a training sample $S = \{(\mathbf{x}_i, y_i), i = 1, ..., m\}$ of size $m$, drawn from an unknown joint distribution $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^p$ is the feature space and $\mathcal{Y} = \{-1, 1\}$ is the set of labels. Let us assume that $S = S_+ \cup S_-$ with $m_+$ positives $\in S_+$ and $m_-$ negatives $\in S_-$ where $m = m_+ + m_-$.

Learning from imbalanced datasets requires to optimize appropriate measures that take into account the scarcity of positive examples. Several of them rely on the following two quantities: the Recall (also called *True Positive Rate* ($TPR$) or *sensitivity*) which measures the capacity of the model to recall/detect positive examples, and the Precision (also called *Positive Prediction Value* ($PPV$)) which is the confidence in the prediction of a positive label. They are defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{and} \quad \text{Precision} = \frac{TP}{TP + FP},$$

where $FP$ (resp. $FN$) is the number of false positives (resp. negatives) and $TP$ is the number of true positives. Since one can arbitrarily improve the Precision if there is no constraint on the Recall (and vice-versa), they are usually combined into a single measure.

For instance, the *F-measure* [23] (or $F_\beta$ score), which is widely used in fraud and anomaly detection [24], is defined as the harmonic mean of the Recall and Precision:

$$F_\beta = (1 + \beta^2) \frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}},$$

where $\beta$ is set such that the Recall is considered $\beta$ times as important as the Precision. Note that $F_1$ (*i.e.* $\beta = 1$) considers the Precision and Recall equally.

The *G-measure* ($G_1$) can also be used for imbalanced data classification [25]. Unlike $F_1$, it rather considers the geometric mean of Precision and Recall:

$$G_1 = \sqrt{\text{Precision} \times \text{Recall}}.$$

While $F_\beta$ and $G_1$ consider both Recall and Precision, the *G-mean* ($GM$) [26] rather makes use of the Recall (or TPR) and the False Negative Rate (TNR) as follows:

$$GM = \sqrt{TPR \times TNR} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}.$$

In other words, it computes the geometric mean of $TPR$ and $TNR$. Thus, it gives a higher importance to the negative class, compared to the previous measures. Without being exhaustive, a last performance measure that can be used in an imbalanced setting is the *Balanced Accuracy* ($BA$) [4] which also relies on TPR and TNR and is defined as the mean accuracy of the two classes:

$$BA = (TPR + TNR)/2$$

In the experimental section of this paper, we will resort to these widely used measures to asses the efficiency of our proposed method to overcome the problem of scarcity of positive samples.

## 3. Related Work

In this section, we present the main strategies that have been proposed in the literature to address the problem of learning from imbalanced datasets. We first present methods specifically dedicated to enhance a $k-$NN classifier. Then, we give an overview of the main sampling strategies used to balance the two classes. All these methods will be used in the experimental comparison in Section 5.

### 3.1. *Distance-based Methods*

Several strategies have been devised to improve $k-$NN. The oldest method is certainly the one presented in [27] which consists in associating to each neighbor a voting weight that is inversely proportional to its distance to a query point $\mathbf{x}$. The assigned label $\hat{y}$ of $\mathbf{x}$ is defined as:

$$\hat{y} = \sum_{\mathbf{x}_i \in \text{kNN}(\mathbf{x})} y_i \times \frac{1}{d(\mathbf{x}, \mathbf{x}_i)},$$

where kNN($\mathbf{x}$) stands for the set of the $k$ nearest neighbors of $\mathbf{x}$. A more refined version consists in taking into account both the distances to the nearest neighbors and the distribution of the features according to the class $p(\mathbf{x}_i \mid y_i)$ [28]. Despite these modifications in the decision rule, the sparsity of the positives remains problematic and it is possible that no positives fall in the neighborhood of a new query $\mathbf{x}$. To tackle this issue, a variant of $k-$NN, called $kP$NN [29], is to consider the region of the space around a new query $\mathbf{x}$ which contains exactly $k$ positive examples. By doing so, the authors are able to use the density of the positives to estimate the probability of belonging in the minority class.

A more recent version has been shown to perform better than the two previous approaches: $kR$NN [30]. If the idea remains similar (*i.e.* estimating the local sparsity of minority examples around a new query), the posterior probability of belonging in the minority class is adjusted so that it considers both the local and global disequilibrium for the estimation. In [31], the authors use both the label and the distance to the neighbors $(\mathbf{x}_i, y_i)$ to define a scaled metric $d'$ from the Euclidean

distance $d$, as follows:

$$d'(\mathbf{x}, \mathbf{x}_i) = \left(\frac{m_i}{m}\right)^{1/p} d(\mathbf{x}, \mathbf{x}_i),$$

where $m_i$ is the number of examples in the class $y_i$. As we will see later, this method falls in the same family of strategies as our contribution, aiming at scaling the distance to the examples according to their label. However, three main differences explain the superiority of our method, observed in the experiments: (i) $k$RNN fixes $d'$ in advance while we will automatically adapt the scaling factor to optimize the considered performance measure; (ii) because of (i), $d'$ needs to take into account the dimension $p$ of the feature space (and so will tend to $d$ as $p$ grows) while our method captures the intrinsic dimension of the space by selecting the best weight; (iii) $d'$ is useless when combined with sampling strategies (indeed, $\frac{m_i}{m}$ would tend to be uniform) while our method will allow us to scale differently the distance to the original positive examples and the ones artificially generated.

Another way to assign weights to each class, which is close to the sampling methods presented in the next section, is to duplicate the positive examples according to the Imbalance Ratio $IR = m_-/m_+$. Thus, it can be seen as a *uniform* oversampling technique, where all positives are replicated the same number of times. However, note that this method requires to work with $k > 1$.

A last family of methods that try to improve $k-$NN is related to *metric learning* [20, 21]. LMNN [18] or ITML [19] are two famous examples which optimize under constraints a Mahalanobis distance $d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}_i) = \sqrt{(\mathbf{x} - \mathbf{x}_i)^\top \mathbf{M}(\mathbf{x} - \mathbf{x}_i)}$ parameterized by a positive semi-definite (PSD) matrix $\mathbf{M}$. Such methods seek a linear projection of the data in a latent space where the Euclidean distance is applied. As we will see in the following, our scaling method is a special case of metric learning which looks for a diagonal matrix (but applied only when comparing a query to a positive example) and which behaves well whatever the considered performance measure.

### 3.2. *Sampling Strategies*

One way to overcome the issues induced by the lack of positive examples is to compensate artificially the imbalance between the two classes. Sampling strategies [9] have been proven to be very efficient to address this problem. In the following, we overview the most used methods in the literature.

The Synthetic Minority Over-sampling Technique [11] (SMOTE) over-samples a dataset by creating new synthetic positive data. For each minority example $\mathbf{x}$, it randomly selects one of its $k$ nearest positive neighbors and then creates a new random positive point on the line between this neighbor and $\mathbf{x}$. This is done until some desired ratio is reached.

Borderline-SMOTE [32] is an improvement of the SMOTE algorithm. While the latter generates synthetic points from all positive points, BorderLine-SMOTE only focuses on those having more negatives than positives in their neighborhood.

More precisely, new samples are generated if the number $n$ of negatives in the $k$-neighborhood is such that $k/2 \leq n \leq k$.

The Adaptive Synthetic [33] (ADASYN) sampling approach is also inspired from SMOTE. By using a weighted distribution, it gives more importance to classes that are more difficult to classify, *i.e.* where positives are surrounded by many negatives, and thus generates more synthetic data for these classes.

Two other strategies combine an over-sampling step with an under-sampling procedure. The first one uses the Edited Nearest Neighbors [34] (ENN) algorithm on the top of SMOTE. After SMOTE has generated data, the ENN algorithm removes samples that are misclassified by their $k$ nearest neighbors. The second one combines SMOTE with Tomek's link [10]. The latter is a pair of points $(\mathbf{x}_i, \mathbf{x}_j)$ from different classes for which there is no other point $\mathbf{x}_k$ verifying $d(\mathbf{x}_i, \mathbf{x}_k) \leq d(\mathbf{x}_i, \mathbf{x}_j)$ or $d(\mathbf{x}_k, \mathbf{x}_j) \leq d(\mathbf{x}_i, \mathbf{x}_j)$. In other words, $\mathbf{x}_i$ is the nearest neighbor of $\mathbf{x}_j$ and vice-versa. If so, one removes the example of $(\mathbf{x}_i, \mathbf{x}_j)$ that belongs to the majority class. Note that both strategies tend to eliminate the overlapping between classes.

Interestingly, we can note that all the previous sampling methods try to overcome the problem of learning from imbalanced data by resorting to the notion of $k$-neighborhood. This is justified by the fact that $k-$NN has been shown to be a good estimate of the density at a given point in the feature space.

In our contribution, we also leverage $k-$NN but with a different approach. Instead of generating (many) new examples (which would have a negative impact from a complexity perspective), we locally modify the density around the positive points. We achieve this by rescaling the distance between a test sample and the positive training examples. We show that such a strategy can be efficiently combined with sampling methods, whose goal is complementary, by potentially generating new examples in regions of the space where the minority class is not present.

## 4. Proposed Approach

In this section, we present our $\gamma k-$NN method which works by scaling the distance between a query point and positive training examples by a factor.

### 4.1. *An Adjusted $k-$NN algorithm*

Statistically, when learning from imbalanced data, a new query $\mathbf{x}$ has more chance to be close to a negative example due to the rarity of positives in the training set, even around the mode of the positive distribution. We have seen two families of approaches that can be used to counteract this effect: (i) creating new synthetic positive examples, and (ii) changing the distance according to the class. The approach we propose falls into the second category.

We suggest to modify how the distance to the positive examples is computed, in order to compensate for the imbalance in the dataset. We artificially bring a new query $\mathbf{x}$ closer to any positive data point $\mathbf{x}_i \in S_+$ in order to increase the effective area of influence of positive examples. The new measure $d_\gamma$ that we propose is

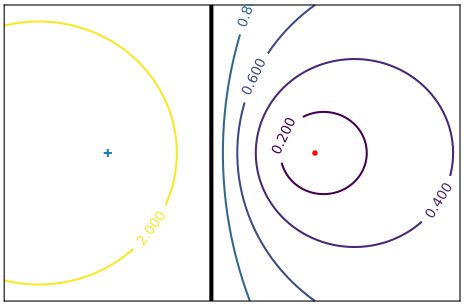


Fig. 2. Evolution of the decision boundary based on $d_\gamma$, for a 1-NN classifier, on a 2D dataset with one positive (resp. negative) instance represented by a blue cross (resp. red point). The value of $\gamma$ is given on each boundary ($\gamma = 1$ on the thick line).

defined using an underlying distance $d$ (*e.g.*, the Euclidean distance) as follows:

$$d_\gamma(\mathbf{x}, \mathbf{x}_i) = \begin{cases} d(\mathbf{x}, \mathbf{x}_i) & \text{if } \mathbf{x}_i \in S_-, \\ \gamma \cdot d(\mathbf{x}, \mathbf{x}_i) & \text{if } \mathbf{x}_i \in S_+. \end{cases} \tag{1}$$

As we will tune the $\gamma$ parameter, this new way to compute the similarity to a positive example is close to a Mahalanobis-distance learning algorithm, looking for a PSD matrix, as previously described. However, the matrix $\mathbf{M}$ is restricted here to be $\gamma^2 \cdot \mathbf{I}$, where $\mathbf{I}$ refers to the identity matrix. Moreover, while metric learning typically works by optimizing a convex loss function under constraints, our $\gamma$ is simply tuned such as maximizing the non convex performance measure. Lastly, and most importantly, it is applied only when comparing the query to positive examples. As such, $d_\gamma$ is not a proper distance. However, this is what allows it to compensate for the class imbalance. In the binary setting, there is no need to have a $\gamma$ parameter for the negative class, since only the relative distances are used. In the multi-class setting with $K$ classes, we would have to tune up to $K - 1$ values of $\gamma$.

Before formalizing the $\gamma k-$NN algorithm that will leverage the distance $d_\gamma$, we illustrate in Fig. 2, on 2D data, the decision boundary induced by a nearest neighbor binary classifier that uses $d_\gamma$. We consider an elementary dataset with only two points, one positive and one negative. The case of $\gamma = 1$, which is a traditional 1-NN is shown in a thick black line. Lowering the value of $\gamma$ below 1 brings the decision boundary closer to the negative point, and eventually tends to surround it very closely. Fig 3 shows, with more complex (toy) datasets, that $\gamma$ controls how much we want to push the boundary towards negative examples. Fig 3 (right) should be imagined as a zoomed-in boundary between the classes, where one class is 20 times less represented. It shows that, due to sampling, the 1-NN boundary wrongly causes regions of false negatives, while $\gamma k-$NN is able to correct the bias.

We can now present $\gamma k-$NN (see Algorithm 1) that is parameterized by the $\gamma$ parameter. It has the same overall complexity as $k-$NN. The first step to classify

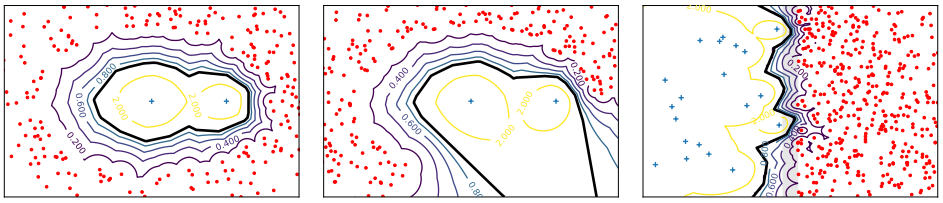

Fig. 3. Behavior of the decision boundary according to the $\gamma$ value for the 1-NN classifier on toy datasets. Positive points are shown as blue crosses and negatives ones as red dots. The black line represents the standard decision boundary for the 1-NN classifier, *i.e.* when $\gamma = 1$.

a query $\mathbf{x}$ is to find its $k$ nearest negative neighbors and its $k$ nearest positive neighbors. Then, the distances to the positive neighbors are multiplied by $\gamma$, to obtain $d_\gamma$. These $2k$ neighbors are then ranked and the $k$ closest ones are used for classification (with a majority vote, as in $k-$NN). It should be noted that, although $d_\gamma$ does not define a proper distance, we can still use any existing fast nearest neighbor search algorithm, because the actual search is done only using the original distance $d$ (but twice, once for $S_+$, once for $S_-$).

**Algorithm 1:** Classification of a new example with $\gamma k-$NN.

**Input :** a query $\mathbf{x}$ to be classified, a set of labeled samples $S = S_+ \cup S_-$, a number of neighbors $k$, a positive real value $\gamma$, a distance function $d$

**Output:** the predicted label of $\mathbf{x}$

$\mathcal{NN}^-, \mathcal{D}^- \leftarrow nn(k, \mathbf{x}, S_-)$ *// nearest negative neighbors with their distances*

$\mathcal{NN}^+, \mathcal{D}^+ \leftarrow nn(k, \mathbf{x}, S_+)$ *// nearest positive neighbors with their distances*

$\mathcal{D}^+ \leftarrow \gamma \cdot \mathcal{D}^+$

$\mathcal{NN}_\gamma \leftarrow firstK\left(k, sortedMerge((\mathcal{NN}^-, \mathcal{D}^-), (\mathcal{NN}^+, \mathcal{D}^+))\right)$

$y \leftarrow \oplus$ if $\left|\mathcal{NN}_\gamma \cap \mathcal{NN}^+\right| \geq \frac{k}{2}$ else $\ominus$ *// majority vote based on* $\mathcal{NN}_\gamma$

**return** $y$

### 4.2. *Theoretical analysis*

In this section, we formally analyze what could be a good range of values for $\gamma$ in our $\gamma k-$NN algorithm. To this aim, we study what impact $\gamma$ has on the probability to get a false positive (and false negative) at test time and explain why it is important to choose $\gamma < 1$ when the imbalance in the data is significant. The following analysis is made for $k = 1$ but note that the conclusion still holds for $k > 1$.

**Proposition 1.** *(False Negative probability) Let $d_\gamma(\mathbf{x}, \mathbf{x}_+) = \gamma d(\mathbf{x}, \mathbf{x}_+)$, $\forall \gamma > 0$, be our modified distance used between a query $\mathbf{x}$ and any positive training example $\mathbf{x}_+$, where $d(\mathbf{x}, \mathbf{x}_+)$ is some distance function. Let $FN_\gamma(\mathbf{z})$ be the probability for a positive example $\mathbf{z}$ to be a false negative using Algorithm (1). The following result*

*holds: if* $\gamma \leq 1$,

$$FN_\gamma(\mathbf{z}) \leq FN(\mathbf{z})$$

**Proof.** (sketch of proof) Let $\epsilon$ be the distance from $\mathbf{z}$ to its nearest-neighbor $N_\mathbf{z}$. $\mathbf{z}$ is a false negative if $N_\mathbf{z} \in S_-$ that is all positives $\mathbf{x}' \in S_+$ are outside the sphere $\mathcal{S}_{\frac{\epsilon}{\gamma}}(\mathbf{z})$ centered at $\mathbf{z}$ of radius $\frac{\epsilon}{\gamma}$. Therefore,

$$\begin{aligned} FN_\gamma(\mathbf{z}) &= \prod_{\mathbf{x}' \in S_+} \left(1 - P(\mathbf{x}' \in \mathcal{S}_{\frac{\epsilon}{\gamma}}(\mathbf{z}))\right), \\ &= \left(1 - P(\mathbf{x}' \in \mathcal{S}_{\frac{\epsilon}{\gamma}}(\mathbf{z}))\right)^{m_+} \end{aligned} \tag{2}$$

while

$$FN(\mathbf{z}) = \left(1 - P(\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{z}))\right)^{m_+}. \tag{3}$$

Solving (2) $\leq$ (3) implies $\gamma \leq 1$. □

This result means that satisfying $\gamma < 1$ allows us to increase the decision boundary around positive examples (as illustrated in Fig. 3), yielding a smaller risk to get false negatives at test time. An interesting comment can be made from Eq.(2) and (3) about their convergence. As $m_+$ is supposed to be very small in imbalanced datasets, the convergence of $FN(\mathbf{z})$ towards 0 is pretty slow, while one can speed-up this convergence with $FN_\gamma(\mathbf{z})$ by increasing the radius of the sphere $\mathcal{S}_{\frac{\epsilon}{\gamma}}(\mathbf{z})$, that is taking a small value for $\gamma$.

**Proposition 2.** *(False Positive probability) Let* $FP_\gamma(\mathbf{z})$ *be the probability for a negative example* $\mathbf{z}$ *to be a false positive using Algorithm (1). The following result holds: if* $\gamma \geq 1$,

$$FP_\gamma(\mathbf{z}) \leq FP(\mathbf{z})$$

**Proof.** (sketch of proof) Using the same idea as before, we get:

$$\begin{aligned} FP_\gamma(\mathbf{z}) &= \prod_{\mathbf{x}' \in S_-} \left(1 - P(\mathbf{x}' \in \mathcal{S}_{\gamma\epsilon}(\mathbf{z}))\right), \\ &= \left(1 - P(\mathbf{x}' \in \mathcal{S}_{\gamma\epsilon}(\mathbf{z}))\right)^{m_-} \end{aligned} \tag{4}$$

while

$$FP(\mathbf{z}) = \left(1 - P(\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{z}))\right)^{m_-}. \tag{5}$$

Solving (4) $\leq$ (5) implies $\gamma \geq 1$. □

As expected, this result suggests to take $\gamma > 1$ to increase the distance $d_\gamma(\mathbf{z}, \mathbf{x}_+)$ from a negative test sample $\mathbf{z}$ to any positive training example $\mathbf{x}_+$ and thus reduce the risk to get a false positive. It is worth noticing that while the two conclusions from Propositions 1 and 2 are contradictory, the convergence of $FP_\gamma(\mathbf{z})$ towards 0

Table 1. Cost matrix for a binary classification task.

| | Predicted positive | Predicted negative |
|---|---|---|
| Actual positive | $c_{TP}$ | $c_{FN}$ |
| Actual negative | $c_{FP}$ | $c_{TN}$ |

is much faster than that of $FN_\gamma(\mathbf{z})$ because $m_- >> m_+$ in an imbalance scenario. Therefore, fulfilling the requirement $\gamma > 1$ is much less important than satisfying $\gamma < 1$. For this reason, we will impose our Algorithm (1) to take $\gamma \in [0, 1]$.

### 4.3. *Link with cost-sensitive learning*

In this section, we show that it is possible to establish a link between the cost-sensitive learning framework [35] and our algorithm $\gamma k-$NN. The goal of cost-sensitive learning is to assign different costs to each entry of the confusion matrix as depicted in Table 1 for a binary setting where we will denote the 4 costs as $c_{TP}$, $c_{FN}$, $c_{FP}$ and $c_{TN}$. Cost sensitive methods are widely used, including in imbalanced scenarios, to give more importance (*i.e.* higher weights/costs) to the examples of the positive/minority class. By doing so, a learned classifier will focus more on decreasing the loss associated to the positive samples. We show here that, despite not being learned by optimizing a loss function, $\gamma k-$NN can still be seen in the lens of cost-sensitive learning.

Let us assume that the correct predictions are not penalized, *i.e.* $c_{TN} = c_{TP} = 0$ and that $c_{FP}$ and $c_{FN}$ are such that $c_{FP} + c_{FN} = 1$ (without loss of generality, as only their relative values matter here). Let $\mathbf{x}^-$ (resp. $\mathbf{x}^+$) be the nearest negative (resp. positive) neighbor of an example $\mathbf{x}$. Suppose that we have a model $\eta(\mathbf{x}) = \mathbb{P}(y = 1 \mid \mathbf{x})$ that gives the probability for $\mathbf{x}$ to be positive. Then the positive label will be assigned to $\mathbf{x}$ if $\eta(\mathbf{x}) > 1/2$, without considering the costs of miss-classification. Taking these latter into account changes the classification rule. Indeed, to minimize the cost-sensitive risk, an example $\mathbf{x}$ must be predicted positive if:

$$\begin{aligned} c_{FP}\ \mathbb{P}(y = 0 \mid \mathbf{x}) &\leq c_{FN}\ \mathbb{P}(y = 1 \mid \mathbf{x}), \\ \Leftrightarrow c_{FP}\ (1 - \eta(\mathbf{x})) &\leq c_{FN}\ \eta(\mathbf{x}), \\ \Leftrightarrow \eta(\mathbf{x}) &\geq c_{FP}. \end{aligned}$$

On the other hand, our algorithm $\gamma k-$NN classifies an example $\mathbf{x}$ as positive if $d(\mathbf{x}, \mathbf{x}^-) > \gamma d(\mathbf{x}, \mathbf{x}^+)$. Given this classification rule, we can show that $\gamma k-$NN resorts to an approximation $\hat{\eta}(\mathbf{x})$ of $\eta(\mathbf{x})$ for a given weighted problem, as follows:

$$\begin{aligned} d(\mathbf{x}, \mathbf{x}^-) &> \gamma d(\mathbf{x}, \mathbf{x}^+), \\ \Leftrightarrow d(\mathbf{x}, \mathbf{x}^-)(1 + \gamma) &> \gamma(d(\mathbf{x}, \mathbf{x}^+) + d(\mathbf{x}, \mathbf{x}^-)), \\ \Leftrightarrow \frac{d(\mathbf{x}, \mathbf{x}^-)}{d(\mathbf{x}, \mathbf{x}^+) + d(\mathbf{x}, \mathbf{x}^-)} &> \frac{\gamma}{1 + \gamma}. \end{aligned}$$

Setting $c_{FP} = \dfrac{\gamma}{\gamma + 1}$ (and therefore $c_{FN} = 1 - \dfrac{\gamma}{\gamma + 1} = \dfrac{1}{\gamma + 1}$) and $\hat{\eta}(\mathbf{x}) =$

$\frac{d(\mathbf{x},\mathbf{x}^-)}{d(\mathbf{x},\mathbf{x}^+)+d(\mathbf{x},\mathbf{x}^-)}$ finishes to establish the link between $\gamma k-$NN and cost-sensitive learning. Note that if $\gamma = 1$ then $c_{FP} = c_{FN} = \frac{1}{2}$ implying that we retrieve a standard $k-$NN classifier which treats positive and negative samples equally without cost sensitivity.

The reader interested in cost-sensitive $k-$NN classifiers can refer to [36, 37].

### 4.4. *Towards a local approach of $\gamma k-$NN*

In what have been presented so far, we consider a single $\gamma$ for the whole input space. While this has the advantage of having a single parameter to tune, it removes the ability to capture non-stationary class imbalance. Indeed, it is possible that a $\gamma$ value is optimal in one part of the space but not in another.

We thus propose a non-stationary version of our algorithm, called local-$\gamma k-$NN. Conceptually, we could have a $\gamma_{\mathbf{x}}$ for every position $\mathbf{x}$ in the space. However, such an over parameterized model would loose the simplicity of the proposed approach and increase the risk of overfitting. To deal with these two issues, we rather partition the input space into $q \in \mathbb{N}^\star$ sub-spaces, $\{C_j\}_{j=1}^q$, using a clustering algorithm (*e.g.*, k-means). Then a value $\gamma_j$, for all $j = 1, ..., q$ is tuned according to the performance measure of interest and using only the available data in the subspace $C_j$. To classify a test query that falls in cluster $j$ we use $\gamma k-$NN (with $\gamma_j$) in this cluster. We will show in the experimental Section 5.4 two possible variants of this local approach.

## 5. Experiments

This part is devoted to an extensive experimental evaluation of $\gamma k-$NN on public datasets with comparisons to classic distance-based methods and state-of-the-art sampling strategies able to deal with imbalanced data. All the results are reported for nearest neighbor classification with $k = 1$ and 3 by considering the four different evaluation measures introduced in Section 2 ($F_1$, $G_1$, $GM$ and $BA$). We also conduct, in Section 5.3, a qualitative analysis on the behavior of our approach on the famous MNIST image dataset [38]. Finally, we conclude our experimental study by an evaluation of the performance of the local version of $\gamma k-$NN (in Section 5.4).

### 5.1. *Experimental setup*

For these experiments, we use 28 public datasets from the well-known UCI[a] and KEEL[b] repositories. The main properties of these datasets are summarized in Table 2, including the imbalance ratio IR defined as: IR$= m_-/m_+$.

[a] *https://archive.ics.uci.edu/ml/datasets.html*

[b] *https://sci2s.ugr.es/keel/datasets.php*

Table 2. Information about the studied public datasets sorted by imbalance ratio IR. The *target* column refers to the label chosen as the minority class (*i.e.* positive examples) in the dataset. The short name of each dataset is given first and will be used, for the sake of readability, in some graphs of this study. (*) The target for YEAST is *ME2 vs MIT,ME3,EXC,VAC,ERL*.

| DATASETS | SIZE | DIM | TARGET | IR | DATASETS | SIZE | DIM | TARGET | IR |
|---|---|---|---|---|---|---|---|---|---|
| BAL - BALANCE | 625 | 4 | *L* | 1.2 | PAG - PAGEBLOCKS | 5472 | 10 | *2,3,4,5* | 8.8 |
| AUT - AUTOMPG | 392 | 7 | *2,3* | 1.7 | SAT - SATIMAGE | 6435 | 36 | *4* | 9.3 |
| ION - IONOSPHERE | 351 | 34 | *b* | 1.8 | YEA - YEAST-0-5-6-7-9VS4 | 528 | 8 | (*) | 9.35 |
| PIM - PIMA | 768 | 8 | *positive* | 1.87 | LIB - LIBRAS | 360 | 90 | *1* | 14 |
| GLA - GLASS | 214 | 9 | *1* | 2.1 | Y17 - YEAST-1VS7 | 459 | 7 | *VAC vs NUC* | 14.3 |
| GER - GERMAN | 1000 | 23 | *2* | 2.3 | ARR - ARRHYTHMIA | 452 | 278 | *6* | 17 |
| YE1 - YEAST1 | 1484 | 8 | *NUC* | 2.46 | SOL - SOLAR-FLARE-M0 | 1389 | 32 | *M0* | 19 |
| HAB - HABERMAN | 306 | 3 | *positive* | 2.78 | OIL - OIL | 937 | 49 | *minority* | 22 |
| VE3 - VEHICLE3 | 846 | 18 | *Class 3 Opel* | 2.99 | YE4 - YEAST4 | 1484 | 8 | *ME2* | 28.1 |
| HAY - HAYES | 132 | 4 | *3* | 3.4 | WI4 - REDWINEQUALITY4 | 1599 | 11 | *4* | 29.2 |
| SEG - SEGMENTATION | 2310 | 19 | *WINDOW* | 6 | YE5 - YEAST5 | 1484 | 8 | *ME1* | 32.73 |
| AB8 - ABALONE8 | 4177 | 10 | *8* | 6.4 | YE6 - YEAST6 | 1484 | 8 | *EXC* | 41.4 |
| YE3 - YEAST3 | 1484 | 8 | *ME3* | 8.1 | A17 - ABALONE17 | 4177 | 10 | *17* | 71 |
| EC3 - ECOLI3 | 336 | 7 | *imU* | 8.6 | A20 - ABALONE20 | 4177 | 10 | *20* | 159.7 |

All the datasets are normalized using a min-max normalization such that each feature lies in the range $[-1, 1]$. We randomly draw 80%-20% splits of the data to generate the training and test sets respectively. Hyperparameters are tuned with a 10-fold cross-validation over the training set. We repeat the process over 5 runs and average the results in terms of the four performance measures. In a first series of experiments, we compare our method $\gamma k-$NN to 6 other distance-based baselines:

- the classic $k-$Nearest Neighbor algorithm ($k-$NN),
- the weighted version of $k-$NN using the inverse distance as a weight to predict the label (w$k-$NN) [27],
- the class weighted version of $k-$NN (cw$k-$NN) [31],
- the $k-$NN version where each positive is duplicated according to the IR of the dataset (dup$k-$NN),
- $kR$NN where the sparsity of minority examples is taken into account [30] by modifying the way the posterior probability of belonging to the positive class is computed.
- the metric learning method LMNN [18].

The hyperparameter $\mu$ of *LMNN*, weighting the impact of impostor constraints (see [18] for more details), is tuned in the range $[0, 1]$ using a step of 0.1. Our $\gamma$ parameter is tuned in the range $[0, 1]$[c] using a step of 0.1. For $kR$NN, we use the parameters values as described in [30].

In a second series of experiments, we compare our method to five oversampling strategies described in Section 3.2: SMOTE, Borderline-SMOTE, ADASYN, SMOTE with ENN, SMOTE with Tomek's link. The number of generated positive

[c]We experimentally noticed that using a larger range for $\gamma$ leads in fact to a potential decrease of performances due to overfitting phenomena. This behavior is actually in line with the analysis provided in Section 4.2.

examples is tuned over the set of ratios $\frac{m_+}{m_-} \in \{0.1, 0.2, ..., 0.9, 1.0\}$ and such that the new ratio is greater than the original one before sampling. The other parameters of these methods are the default ones used by the package *ImbalancedLearn* of *Scikit-learn*. We report the performance of the best oversampler that we denote as OS$^\star$. In order to evaluate how both strategies are complementary, we also combine $\gamma k-$NN with oversamplers, and use the notation (OS+$\gamma k-$NN)$^\star$ to indicate the best combinaison obtained by a 10-cross validation. In this latter scenario, we propose to learn a different $\gamma$ value to be used with the synthetic positives. Indeed, some of the synthetic examples may be generated in some true negative areas and, in this situation, it might be more appropriate to decrease their influence. The $\gamma$ parameter for these examples is tuned in the range $[0, 2]$ using a step of 0.1. Note the upper bound of the range is now set to 2. This allows $\gamma k-$NN to adapt to the different sampling strategies of the oversamplers and enables the possibility to move synthetic positive examples away from dense regions of negatives by selecting $\gamma > 1$.

### 5.2. *Analysis of the results*

The results on the public datasets using the six baselines are provided in Tables 4,5,6 and 7 for the four different performance measures $F_1$, $BA$, $GM$ and $G_1$ respectively. These tables report the complete results when $k = 1$ (in $k-$NN) and provide only the mean results over the 28 datasets for $k = 3$, for the sake of concision and because the behavior for this latter value is similar. Overall, our $\gamma k-$NN approach performs much better than its competitors by achieving an improvement of 0.7 to 5 points on average, compared to the other state-of-the-art algorithms when $k = 1$. It is worth noticing that the results are even better when $k = 3$. But the certainly most striking result comes from the capacity of $\gamma k-$NN associated with the Balanced Accuracy ($BA$) in Table 5 and G-mean ($GM$) in Table 6 to address large imbalanced learning tasks. While the other methods struggle to get good results, $\gamma k-$NN with $BA$ and $GM$ gets the best performance 19 and 20 times respectively over the 22 largest imbalanced datasets (from YEAST1 to ABALONE20). Even the metric learning algorithm LMNN fails to be competitive while it optimizes a representation of the data specifically dedicated to deal with nearest neighbor classification. Indeed, LMNN suffers from the lack of positive data to learn an efficient projection when dealing with highly imbalanced tasks. On the other hand, $\gamma k-$NN does not seem particularly sensitive to the imbalance ratio.

The second series of experiments focuses on the use of sampling strategies and the potential interest of combining $\gamma k-$NN with a synthetic generation of additional positive examples. Fig. 4 compactly summarizes, for the four measures of interest and for both $k = 1$ (on the left) and $k = 3$ (on the right), the impact of sampling strategies. Two main comments can be made from these results. First, $\gamma k-$NN is complementary to the oversamplers. Indeed, for both $k = 1$ and $k = 3$ and for 3 out of 4 measures ($G_1$ excluded), using $\gamma k-$NN in addition with a sampler leads

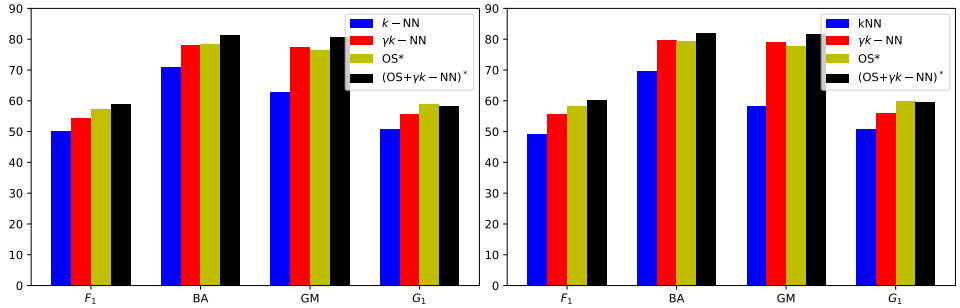


Fig. 4. Comparison of $k-$NN, $\gamma k-$NN, the best oversampler among SMOTE, BorderSmote, SMOTE+ENN, SMOTE+Tomek's links and ADASYN, and the best coupling oversampler + $\gamma k-$NN, in terms of mean of F-measure ($F_1$), Balanced Accuracy ($BA$), Geometric Mean ($GM$) and G-measure ($G_1$) over all the datasets, for $k = 1$ (left) and $k = 3$ (right).

to better results and gives evidence of the fact that $\gamma k-$NN and oversamplers do not work the same way, focusing on different subparts of the feature space. While $\gamma k-$NN aims at expanding the decision boundaries in favor of the positives in the neighborhood of the test query, oversamplers rather tend to fill in the empty parts of the space by generating synthetic positive examples. Second, $\gamma k-$NN (red bars) alone is shown to be very competitive while benefiting from its simplicity. Indeed, we remind the reader that the performance of OS$^\star$ (resp. (OS+$\gamma k-$NN)$^\star$) are obtained from the costly selection of the best oversampler (resp. $\gamma k-$NN + oversampler) for each dataset. Therefore, the green and black bars in Fig. 4 give an optimistic usage of an oversampling strategy because it is generated from the average obtained over a large set of oversamplers (SMOTE, Borderline-SMOTE, ADASYN, SMOTE+ ENN and SMOTE with Tomek's link) that can be seen as an additional hyperparameter. On the other hand, in $\gamma k-$NN, only one parameter ($\gamma$) is required to be tuned.

Fig. 5 illustrates, for the $F_1$ and $GM$ measures and $k = 1$, a dataset-wise view of the advantage of combining $\gamma k-$NN with an oversampler compared to a standard $k-$NN. A point (representing one of the 28 datasets) below the line $y = x$ means that $k-$NN is outperformed. Moreover, a move of a point (illustrated by a right arrow) from left to right illustrates that the joint approach leads to better results. We can see that even for the least favorable measure (*i.e.* $F_1$ on the left), most of the datasets are below the line and benefit from $\gamma k-$NN associated to an oversampler.

In Fig 6 (left), we illustrate how having two $\gamma$ parameters ($\gamma$ on reals and $\gamma$ on synthetics) gives the flexibility to independently control the influence of the actual and artificial positives respectively. The other figures (center and right) represent two examples of heatmaps of the F-measure (note that the trend is the same for the other 3 measures). We can note that while the $\gamma$ parameter tuned for the real positives tends to be smaller than 1 (according to the analysis of Section 4.2), the $\gamma$ parameter required to deal with the synthetic positives is sometimes smaller (right), sometimes greater than 1 (center), depending on the underlying density and the pe-

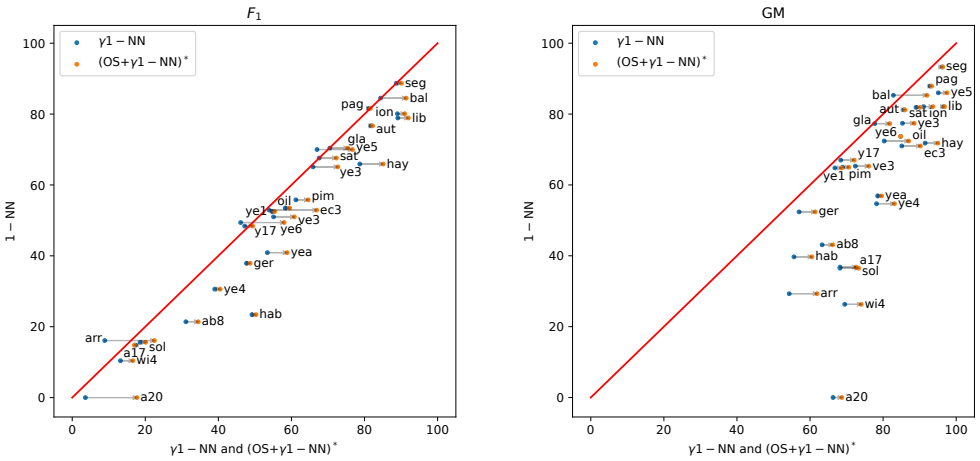


Fig. 5. Comparison of $k-$NN with (i) $\gamma k-$NN (points in blue) and (ii) $\gamma k-$NN coupled with the best sampling strategy (points in orange) for each dataset, in terms of F-measure (left) and Geometric Mean (right) and for $k = 1$. Points below the line $y = x$ means that $k-$NN is outperformed. A move from left to right illustrates that the joint approach is better.

Table 3. Comparison of $\gamma k-$NN and $k-$NN on MNIST for $k = 3$.

| | $F_1$ | $BA$ | $GM$ | $G_1$ |
|---|---|---|---|---|
| $k-$NN | 97.18 | 98.31 | 98.30 | 97.19 |
| $\gamma k-$NN | **97.21** | **98.97** | **98.97** | **97.21** |

culiarity of the feature space.

Recall that Propositions 1 and 2 in Section 4.2 tell us that selecting a $\gamma$ parameter smaller than 1 for the real positives should tend to reduce the false negative (FN) rate while still optimizing the performance measure. To illustrate our theoretical study, we plot in Fig. 7 the percentage of FN generated by the 7 compared methods. As expected, we can note that whatever the performance measure and the value of $k$ ($k = 1$ on the left and $k = 3$ on the right), the number of FN is much smaller than that of the competitors explaining why $\gamma k-$NN gets the best results.

### 5.3. *A qualitative analysis on the MNIST dataset*

In order to visualize the qualitative impact of $\gamma k-$NN, we conduct in this section some additional experiments on the MNIST dataset. To generate a minority class, we build 10 datasets $MNIST_i$ (one for each digit, $i = 0, ..., 9$) from the original one by considering the label $i$ as the minority class and all the other classes representing the remaining digits as the majority class.

As previously done, a 10-fold CV is performed to find the optimal value $\gamma$. The mean results of the comparison of $\gamma k-$NN with $k-$NN are reported in Table 3

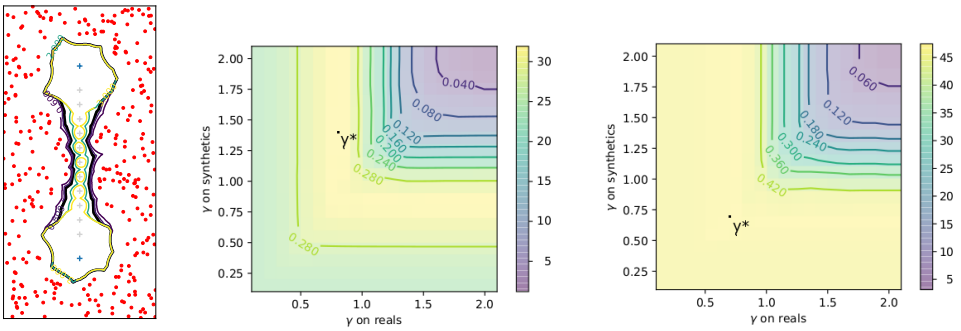


Fig. 6. Left: Illustration on a toy dataset of the effect of varying $\gamma$ for the generated positive points (in grey) while keeping a fixed $\gamma = 0.4$ for the real positives. Center and Right: Two examples of heatmap for the F-Measure that show the pair of $\gamma$ (on real and synthetic positives) corresponding to the best joint approach (OS+ $\gamma k-$NN)* on Abalone8 (center) and German (right).

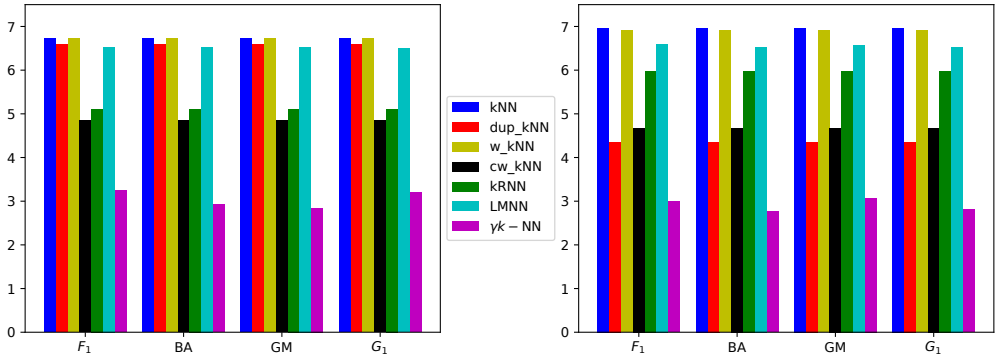


Fig. 7. Percentage of false negatives (FN) generated by the 7 compared methods w.r.t. the four performance measures with 1−NN (left) and 3−NN (right).

where $k = 3$. We can see that whatever the performance measure, $\gamma k-$NN allows us to outperform $k-$NN. As expected, the gain on this well-known MNIST dataset is not significant due to the already very high accuracy reached by the standard $k-$NN. However, the main objective here is elsewhere. We aim at showing the quality of both the space and the neighborhoods induced by $\gamma k-$NN. To illustrate this purpose and visualize how using $d_\gamma$ (as defined in Eq.(1)) bends the feature space, we leverage t-SNE to embed the $MNIST_i$ points in 2D. Note that even if $d_\gamma$ is not an actual distance (the symmetry is not satisfied), it can still be used with t-SNE that only embeds points while preserving relative pair-wise dissimilarities.

Following the definition of $d_\gamma$, we scale the Euclidean distance when the second point in the pair is a positive one. Fig. 8 compares, on the $MNIST_2$ dataset, the output of t-SNE when using $d$ (left) and $d_\gamma$ (right). The analysis of this embedding shows that $d_\gamma$ is able to gather minority examples together in a denser cluster while the Euclidean distance leads to a space where the positive samples are more

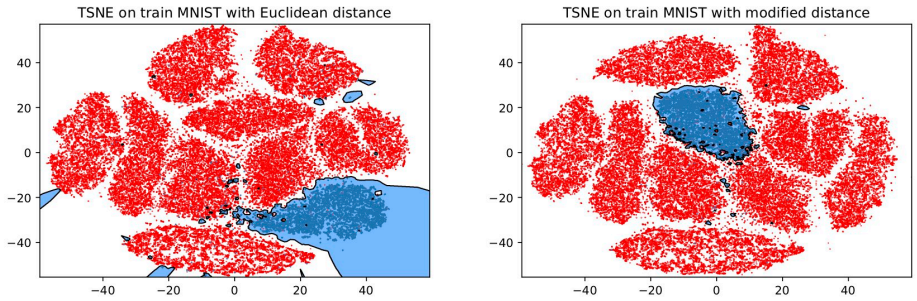


Fig. 8. Visualization on $MNIST_2$ of the influence of the Euclidean distance $d$ (left) and $d_\gamma$ (right) with t-SNE. The red (resp. blue) points correspond to negatives (resp. positives). The blue areas represent the subparts of the space leading to a positive classification by a 3−NN.

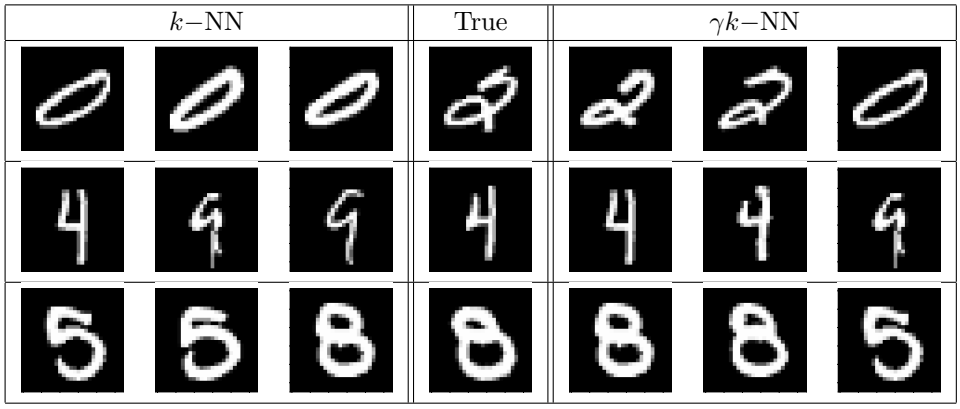


Fig. 9. First three columns: 3 nearest neighbors using $k-$NN; fourth column: the test query; last three figures: 3 nearest neighbors using $\gamma k-$NN.

scattered, some being in the middle of negative regions. This impact of $\gamma k-$NN on the decision boundaries, that we see with this t-SNE experiment, is also illustrated in Fig. 9 which shows some examples for which, the $\gamma k-$NN predictions are different from that of $k-$NN according to their 3-nearest neighbors (on the original dataset).

### 5.4. *On local-$\gamma k-$NN using clustering*

We now evaluate our local algorithm local-$\gamma k-$NN (as presented in Section 4.4), which partitions the input space into $q$ clusters $(C_1, C_2, ..., C_q)$ and uses a parameter $\gamma_j$ for each cluster $j$. The partitioning is performed using k-means, run on the training set. Note that we consider two ways of obtaining the $\gamma_j$ values. The first version (V1) consists in applying the 10 fold cross-validation (CV) procedure in each cluster $C_j$ to tune $\gamma_j$. At test time, a new point $\mathbf{x}'$ is first assigned to the

nearest cluster $C_k$ based on the closest centroid using the Euclidean distance, and the corresponding $\gamma_k$ value is used to scale the distances to the positives.
We propose a second version (V2) to compensate for the fact that V1 is at risk of generating very different values of $\gamma$ for two neighboring clusters. While the test decision is similar to V1, the $\gamma_j$ values are obtained differently, by computing several clusterings. In V2, the 10 fold CV also includes the clustering, so 10 additional partitionings are performed. Each training point $\mathbf{x}$ will thus fall in 9 clusters (in the 9 different clusterings for which the point is not in the validation fold). Each point thus has 9 "best" $\gamma$ values that we average to get a single value $\gamma_{\mathbf{x}}$ for every single point. In the end, $\gamma_j$ is computed as $\frac{1}{|C_j|}\sum_{\mathbf{x}\in C_j}\gamma_{\mathbf{x}}$, *i.e.* the average $\gamma$ value of the training points falling into cluster $j$.

The results are provided for the 4 performance measures in Fig. 10, 11, 12 and 13. Despite an inherent increase of the time complexity, it is worth noting that V2 is better than the original $\gamma k-$NN (on average over the 28 datasets), while V1 does not lead to improvements probably due to overfitting phenomena. Note also that in a huge majority of the datasets (around 90%), the V2 version of local-$\gamma k-$NN equals or outperforms $\gamma k-$NN.

## 6. Conclusion

In this paper, we have proposed a new approach, $\gamma k-$NN, that addresses the problem of learning from imbalanced datasets. It is based on the $k-$NN algorithm but it modifies the distance to the positive examples by expanding the decision boundaries around these minority samples. It has been shown to outperform its competitors in terms of several performance measures. Furthermore, we gave evidence of the complementarity of $\gamma k-$NN with oversampling strategies. Our algorithm, despite its simplicity, is highly effective and its local version local-$\gamma k-$NN has shown to be even more efficient by taking the spatial specificity of the distributions into account.

Two main lines of research deserve future investigations. First, we plan to extend the idea of the local variant of $\gamma k-$NN by proposing a multi-view learning approach, where the different results of $\gamma k-$NN obtained with different subsets of features (the different views) would be combined in some way. Second, we can note that tuning $\gamma$ is equivalent to building a diagonal matrix (with $\gamma^2$ in the diagonal) and applying a Mahalanobis distance only between a query and a positive example. This comment opens the door to a new family of metric learning algorithms dedicated to optimizing a PSD matrix under $(FP, FN)$-based constraints that could leverage recent metric learning approaches for imbalanced data [39].

## Acknowledgments

This work was supported by the following projects: AURA project TADALoT (Pack Ambition 2017, 17 011047 01), ANR project LIVES (ANR-15-CE23-0026) and IDEXLYON project ACADEMICS (ANR-16-IDEX-0005).

Table 4. Results for $k = 1$ with $F_1$ as performance measure over 5 runs. The standard deviation is indicated after the $\pm$ sign and the best results on each dataset is indicated in bold. Only the mean value when $k = 3$ is shown in the last line.

| DATASETS | $k-$NN | DUP$k-$NN | W$k-$NN | CW$k-$NN | KRNN | LMNN | $\gamma k-$NN |
|---|---|---|---|---|---|---|---|
| BALANCE | 84.5±2.2 | 84.5±2.2 | 84.5±2.2 | 84.4±1.7 | **88.2±1.2** | 84.1±4.6 | 84.4±1.7 |
| AUTOMPG | 76.7±7.4 | 76.7±7.4 | 76.7±7.4 | 76.2±6.2 | **82.5±2.8** | 77.5±3.1 | 81.8±3.0 |
| IONOSPHERE | 80.1±4.2 | 80.1±4.2 | 80.1±4.2 | 83.3±3.0 | 0.83±3.0 | 81.3±3.3 | **89.0±4.5** |
| PIMA | 55.8±4.7 | 55.8±4.7 | 55.8±4.7 | 61.1±3.7 | **62.3±3.9** | 55.9±2.8 | 61.2±5.4 |
| GLASS | 70.4±8.7 | 70.4±8.7 | 70.4±8.7 | 73.2±6.4 | **76.2±8.6** | 68.9±7.7 | 70.5±7.5 |
| GERMAN | 37.9±5.0 | 37.9±5.0 | 37.9±5.0 | 41.1±3.6 | 43.7±4.0 | 41.0±3.8 | **47.7±1.9** |
| YEAST1 | 52.5±2.6 | 52.5±2.6 | 52.5±2.6 | 53.3±3.6 | 52.5±2.1 | 51.3±3.7 | **54.8±3.8** |
| HABERMAN | 23.4±6.7 | 23.4±6.7 | 23.4±6.7 | 35.5±10 | 33.2±7.6 | 24.6±6.9 | **49.2±4.4** |
| VEHICLE3 | 51.0±3.4 | 51.0±3.4 | 51.0±3.4 | 51.2±3.7 | **56.1±3.3** | 54.7±3.6 | 55.1±3.4 |
| HAYES | 65.9±10 | 65.9±10 | 65.9±10 | **87.5±4.9** | 76.6±8.3 | 76.8±14 | 78.7±10 |
| SEGMENTATION | 88.7±2.9 | 88.7±2.9 | 88.7±2.9 | 88.9±2.8 | 87.2±1.6 | **91.2±2.3** | 88.7±2.9 |
| ABALONE8 | 21.4±1.3 | 21.4±1.3 | 21.4±1.3 | **31.3±0.9** | 23.8±1.1 | 21.9±1.7 | 31.1±2.0 |
| YEAST3 | 65.1±2.5 | 65.1±2.5 | 65.1±2.5 | 63.0±2.6 | **69.4±1.2** | 63.6±1.0 | 65.9±2.3 |
| ECOLI3 | 52.9±9.7 | 52.9±9.7 | 52.9±9.7 | 54.1±7.8 | **61.2±5.8** | 57.2±11 | 53.9±7.0 |
| PAGEBLOCKS | **81.6±2.6** | **81.6±2.6** | 81.6±2.6 | 81.1±2.4 | 81.3±4.4 | 81.5±3.2 | 81.2±2.2 |
| SATIMAGE | 67.6±3.6 | 67.6±3.6 | 67.6±3.6 | 68.0±3.4 | 68.8±2.7 | **69.0±4.5** | 67.6±3.6 |
| YEAST-0-5-6-7-9VS4 | 40.9±11 | 40.9±11 | 40.9±11 | 49.7±4.1 | 51.9±7.3 | 45.5±15 | **53.4±8.3** |
| LIBRAS | 78.9±8.7 | 78.9±8.7 | 78.9±8.7 | 78.9±8.7 | 73.7±6.0 | 78.8±5.4 | **89.1±8.1** |
| YEAST-1VS7 | 48.4±6.0 | 48.4±6.0 | 48.4±6.0 | 23.8±5.3 | **49.1±8.8** | 40.4±16.6 | 47.2±5.0 |
| ARRYTHMIA | 16.1±20 | 16.1±20 | 16.1±20 | 15.6±20 | 15.6±20 | **20.2±21** | 8.9±12 |
| SOLAR-FLARE-M0 | 15.6±5.6 | 15.6±5.6 | 15.6±5.6 | 18.4±1.9 | **21.4±6.3** | 15.3±10 | 18.6±2.1 |
| OIL | 53.5±11.2 | 53.5±11 | 53.5±11 | 57.2±9.7 | 55.5±5.2 | **61.3±12** | 58.3±12 |
| YEAST4 | 30.6±11.3 | 30.6±11 | 30.6±11 | 29.2±1.9 | **41.4±4.0** | 32.4±12 | 39.0±8.4 |
| REDWINEQUALITY4 | 10.4±5.7 | 10.4±5.7 | 10.4±5.7 | 12.4±2.6 | 12.9±7.0 | 12.0±6.9 | **13.2±5.6** |
| YEAST5 | 70.0±11 | 70.0±11 | 70.0±11 | 56.4±8.2 | 62.6±8.3 | **70.1±12** | 67.0±9.8 |
| YEAST6 | 49.4±13 | 49.4±13 | 49.4±13 | 26.2±2.3 | **49.9±8.6** | 47.1±19 | 46.1±10 |
| ABALONE17 | 14.8±9.7 | 14.8±9.7 | 14.8±9.7 | 10.5±4.0 | 16.3±6.9 | 14.3±6.7 | **17.3±9.4** |
| ABALONE20 | 00.0±0.0 | 00.0±0.0 | 00.0±00.0 | 05.2±3.5 | **06.6±6.7** | 00.0±0.0 | 03.6±4.7 |
| MEAN (K=1) | 50.1 | 50.1 | 50.1 | 50.6 | 53.7 | 51.4 | **54.4** |
| MEAN (K=3) | 49.3 | 54.0 | 50.0 | 52.2 | 53.2 | 51.9 | **55.8** |

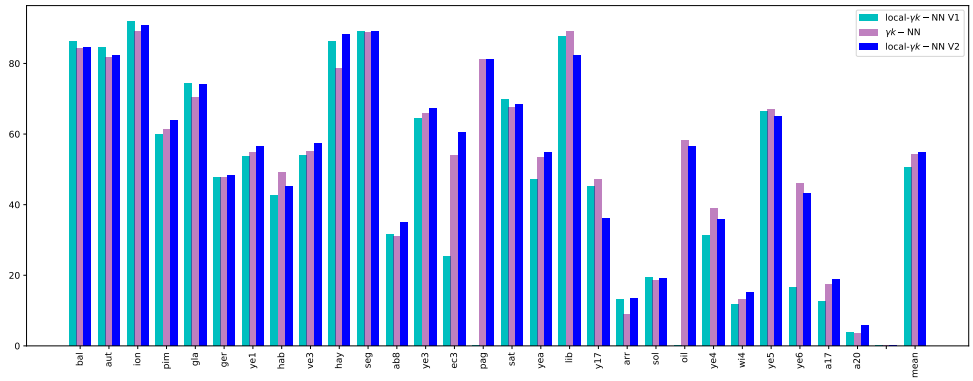


Fig. 10. Comparison of $\gamma 1-$NN with the two versions of local-$\gamma 1-$NN, in terms of F-measure.

Table 5. Results for $k = 1$ with $BA$ as performance measure over 5 runs. The standard deviation is indicated after the $\pm$ sign and the best results on each dataset is indicated in bold. Only the mean value when $k = 3$ is shown in the last line.

| DATASETS | $k-$NN | DUP$k-$NN | W$k-$NN | CW$k-$NN | KRNN | LMNN | $\gamma k-$NN |
|---|---|---|---|---|---|---|---|
| BALANCE | 85.4±2.1 | 85.4±2.1 | 85.4±2.1 | 84.2±1.9 | **88.9±1.2** | 85.5±4.1 | 84.2±1.9 |
| AUTOMPG | 81.6±6.1 | 81.6±6.1 | 81.6±6.1 | 81.1±5.2 | **86.4±2.6** | 81.9±2.6 | 86.0±2.9 |
| IONOSPHERE | 83.6±3.0 | 83.6±3.0 | 83.6±3.0 | 85.9±2.1 | 85.6±2.1 | 84.6±2.4 | **91.2±3.6** |
| PIMA | 66.6±3.4 | 66.6±3.4 | 66.6±3.4 | 69.4±3.3 | **70.4±3.4** | 66.7±1.7 | 69.7±3.7 |
| GLASS | 78.2±6.5 | 78.2±6.5 | 78.2±6.5 | 80.3±4.8 | **83.3±6.7** | 77.1±6.4 | 76.5±4.3 |
| GERMAN | 57.1±3.0 | 57.1±3.0 | 57.1±3.0 | 58.0±2.4 | **59.0±3.1** | 58.6±2.6 | 57.7±3.0 |
| YEAST1 | 66.6±1.9 | 66.6±1.9 | 66.6±1.9 | 66.5±3.2 | 66.2±1.7 | 65.8±2.6 | **67.2±3.8** |
| HABERMAN | 49.8±5.4 | 49.8±5.4 | 49.8±5.4 | 52.7±9.0 | 53.3±6.2 | 50.4±5.4 | **61.1±4.7** |
| VEHICLE3 | 67.3±2.3 | 67.3±2.3 | 67.3±2.3 | 67.4±2.8 | 71.2±2.7 | 70.1±3.2 | **72.6±1.1** |
| HAYES | 75.7±6.0 | 75.7±6.0 | 75.7±6.0 | 90.7±4.6 | 82.4±5.4 | 82.9±8.7 | **91.9±4.2** |
| SEGMENTATION | 93.5±2.2 | 93.5±2.2 | 93.5±2.2 | 95.1±2.1 | 95.5±0.9 | 94.9±1.5 | **96.2±0.9** |
| ABALONE8 | 54.5±0.7 | 54.5±0.7 | 54.5±0.7 | 61.3±0.8 | 55.7±0.7 | 54.9±0.9 | **62.6±1.8** |
| YEAST3 | 79.4±2.9 | 79.4±2.9 | 79.4±2.9 | 85.5±2.4 | 83.4±2.8 | 80.4±2.5 | **85.7±2.9** |
| ECOLI3 | 74.4±6.9 | 74.4±6.9 | 74.4±6.9 | 82.3±5.9 | 81.2±5.2 | 73.3±3.5 | **85.5±8.1** |
| PAGEBLOCKS | 88.5±1.5 | 88.5±1.5 | 88.5±1.5 | 91.4±2.1 | 90.4±2.3 | 88.7±1.9 | **92.9±1.3** |
| SATIMAGE | 83.0±2.0 | 83.0±2.0 | 83.0±2.0 | 87.5±1.7 | 86.6±1.6 | 83.8±1.8 | **89.1±1.2** |
| YEAST-0-5-6-7-9VS4 | 65.4±5.2 | 65.4±5.2 | 65.4±5.2 | 78.3±3.7 | 71.6±3.3 | 68.1±7.4 | **79.3±3.6** |
| LIBRAS | 83.9±5.0 | 83.9±5.0 | 83.9±5.0 | 83.9±5.0 | 83.4±4.8 | 83.9±4.7 | **96.7±3.4** |
| YEAST-1VS7 | 71.7±3.2 | 71.7±3.2 | 71.7±3.2 | 67.5±7.7 | **73.1±5.3** | 68.1±1.0 | 72.5±7.0 |
| ARRYTHMIA | 57.5±16 | 57.5±16 | 57.5±16 | 56.8±17 | 57.2±16 | **59.7±16** | 54.7±6.5 |
| SOLAR-FLARE-M0 | 55.1±2.5 | 55.1±2.5 | 55.1±2.5 | 65.1±1.8 | 58.2±2.9 | 55.0±4.3 | **67.3±4.2** |
| OIL | 76.6±8.8 | 76.6±8.8 | 76.6±8.8 | 80.7±6.2 | 83.4±4.0 | 79.3±9.7 | **84.6±4.4** |
| YEAST4 | 64.9±8.3 | 64.9±8.3 | 64.9±8.3 | 78.7±2.1 | 77.5±3.5 | 66.6±8.5 | **79.3±3.5** |
| REDWINEQUALITY4 | 53.5±2.7 | 53.5±2.7 | 53.5±2.7 | 58.7±3.9 | 55.7±4.4 | 54.3±3.2 | **69.2±5.8** |
| YEAST5 | 87.2±7.4 | 87.2±7.4 | 87.2±7.4 | 91.4±5.5 | 90.9±5.7 | 86.2±6.7 | **95.1±2.8** |
| YEAST6 | 77.7±10 | 77.7±10 | 77.7±10 | 84.9±9.3 | **85.7±9.3** | 79.0±14 | 79.2±7.0 |
| ABALONE17 | 56.8±4.9 | 56.8±4.9 | 56.8±4.9 | 64.2±7.3 | 63.2±6.8 | 57.7±3.7 | **67.0±4.2** |
| ABALONE20 | 49.7±0.1 | 49.7±0.1 | 49.7±0.1 | 58.8±6.9 | 55.5±6.4 | 49.7±0.1 | **68.8±11** |
| MEAN (K=1) | 70.9 | 70.9 | 70.9 | 75.3 | 74.8 | 71.7 | **78.0** |
| MEAN (K=3) | 69.6 | 75.4 | 69.9 | 75.5 | 74.2 | 71.7 | **79.7** |

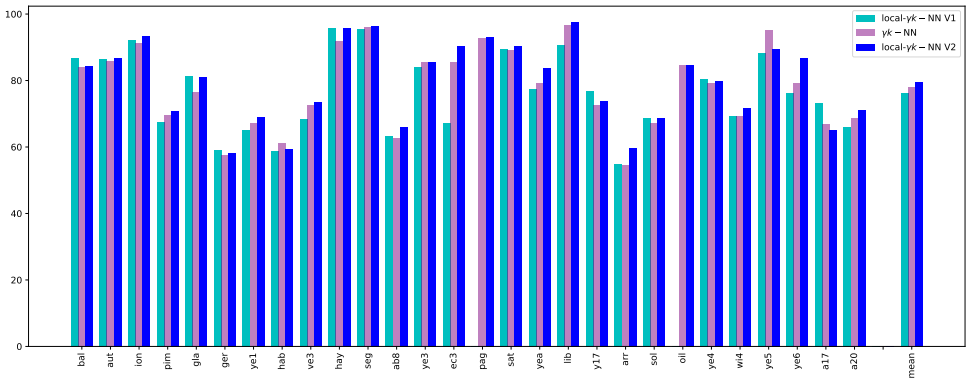


Fig. 11. Comparison of $\gamma 1-$NN with the two versions of local-$\gamma 1-$NN, in terms of Balanced Accuracy (BA).

Table 6. Results for $k = 1$ with $GM$ as performance measure over 5 runs. The standard deviation is indicated after the $\pm$ sign and the best results on each dataset is indicated in bold. Only the mean value when $k = 3$ is shown in the last line.

| DATASETS | $k-$NN | DUP$k-$NN | W$k-$NN | CW$k-$NN | KRNN | LMNN | $\gamma k-$NN |
|---|---|---|---|---|---|---|---|
| BALANCE | 85.3±2.1 | 85.3±2.1 | 85.3±2.1 | 82.8±2.2 | **88.8±1.2** | 85.4±4.2 | 82.8±2.2 |
| AUTOPMG | 81.2±6.4 | 81.2±6.4 | 81.2±6.4 | 80.7±5.4 | **86.2±2.5** | 81.6±2.6 | 85.7±3.1 |
| IONOSPHERE | 82.1±3.5 | 82.1±3.5 | 82.1±3.5 | 84.9±2.3 | 84.7±2.3 | 84.1±2.2 | **91.1±3.7** |
| PIMA | 65.0±3.8 | 65.0±3.8 | 65.0±3.8 | 69.2±3.2 | **70.3±3.4** | 65.1±2.3 | 68.9±3.9 |
| GLASS | 77.3±7.2 | 77.3±7.2 | 77.3±7.2 | 79.7±5.3 | **82.6±7.2** | 76.1±6.8 | 77.7±6.0 |
| GERMAN | 52.4±4.2 | 52.4±4.2 | 52.4±4.2 | 55.4±3.0 | **57.6±3.4** | 55.1±3.2 | 57.0±3.0 |
| YEAST1 | 64.8±1.8 | 64.8±1.8 | 64.8±1.8 | 66.3±3.1 | 65.6±1.7 | 63.8±2.7 | **66.8±4.2** |
| HABERMAN | 39.7±6.4 | 39.7±6.4 | 39.7±6.4 | 51.4±9.8 | 49.6±6.8 | 40.9±5.9 | **55.6±7.9** |
| VEHICLE3 | 65.3±3.1 | 65.3±3.1 | 65.3±3.1 | 66.6±3.1 | 70.5±3.2 | 68.4±4.1 | **72.4±1.6** |
| HAYES | 71.8±8.6 | 71.8±8.6 | 71.8±8.6 | 90.2±5.1 | 80.6±6.6 | 80.3±11.9 | **91.5±4.6** |
| SEGMENTATION | 93.3±2.3 | 93.3±2.3 | 93.3±2.3 | 95.0±2.2 | 95.5±0.9 | 94.9±1.5 | **96.2±0.9** |
| ABALONE8 | 43.1±1.6 | 43.1±1.6 | 43.1±1.6 | 59.6±0.9 | 47.2±1.6 | 43.6±2.1 | **63.3±2.2** |
| YEAST3 | 77.4±3.9 | 77.4±3.9 | 77.4±3.9 | 85.2±2.9 | 82.3±3.6 | 78.9±3.4 | **85.3±3.5** |
| ECOLI3 | 71.0±9.0 | 71.0±9.0 | 71.0±9.0 | 81.9±6.5 | 79.9±6.1 | 73.2±9.7 | **85.1±8.5** |
| PAGEBLOCKS | 87.9±1.6 | 87.9±1.6 | 87.9±1.6 | 91.2±2.2 | 90.1±2.5 | 88.1±2.0 | **92.9±1.3** |
| SATIMAGE | 81.9±2.3 | 81.9±2.3 | 81.9±2.3 | 87.3±1.8 | 86.2±1.7 | 83.2±2.7 | **89.0±1.2** |
| YEAST-0-5-6-7-9VS4 | 56.9±9.2 | 56.9±9.2 | 56.9±9.2 | 77.5±4.4 | 67.1±4.7 | 58.9±14.2 | **78.5±4.2** |
| LIBRAS | 82.1±6.0 | 82.1±6.0 | 82.1±6.0 | 82.1±6.0 | 81.8±5.7 | 82.0±5.8 | **96.6±3.6** |
| YEAST-1VS7 | 67.0±5.0 | 67.0±5.0 | 67.0±5.0 | 65.8±9.4 | **69.0±7.5** | 59.8±16.5 | 68.4±10.5 |
| ARRYTHMIA | 29.3±36.7 | 29.3±36.7 | 29.3±36.7 | 29.2±36.6 | 29.2±36.6 | 37.9±34.1 | **54.3±8.4** |
| SOLAR-FLARE-M0 | 36.5±6.1 | 36.5±6.1 | 36.5±6.1 | 63.9±1.9 | 43.5±5.9 | 36.3±8.6 | **68.2±4.3** |
| OIL | 72.4±12.4 | 72.4±12.4 | 72.4±12.4 | 78.6±7.9 | **82.3±5.0** | 75.8±12.7 | 80.3±2.2 |
| YEAST4 | 54.7±13.5 | 54.7±13.5 | 54.7±13.5 | 78.1±2.4 | 75.3±4.4 | 57.6±14.1 | **78.2±4.3** |
| REDWINEQUALITY4 | 26.3±14.0 | 26.3±14.0 | 26.3±14.0 | 50.1±7.1 | 35.1±17.9 | 28.8±15.4 | **69.5±6.6** |
| YEAST5 | 86.0±8.4 | 86.0±8.4 | 86.0±8.4 | 91.1±6.0 | 90.4±6.2 | 85.0±7.3 | **95.1±2.8** |
| YEAST6 | 73.7±13.7 | 73.7±13.7 | 73.7±13.7 | 83.7±11.1 | 84.1±11.2 | 73.7±21.1 | **84.8±6.2** |
| ABALONE17 | 36.8±11.3 | 36.8±11.3 | 36.8±11.3 | 56.9±11.2 | 52.4±12.1 | 39.4±9.4 | **68.2±6.4** |
| ABALONE20 | 00.0±0.0 | 00.0±0.0 | 00.0±0.0 | 45.4±11.8 | 27.6±23.4 | 00.0±0.0 | **66.3±12.6** |
| MEAN (K=1) | 62.9 | 62.9 | 62.9 | 72.5 | 69.8 | 64.2 | **77.5** |
| MEAN (K=3) | 58.4 | 71.6 | 59.1 | 70.71 | 67.6 | 61.7 | **78.9** |

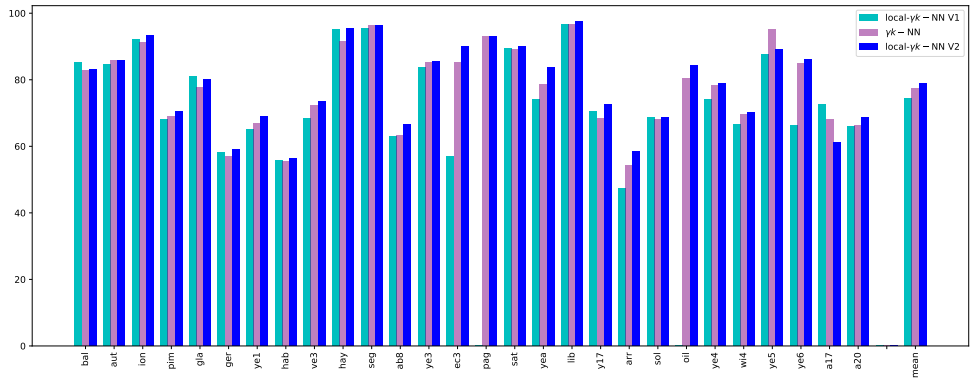


Fig. 12. Comparison of $\gamma 1-$NN with the two versions of local-$\gamma 1-$NN, in terms of Geometric Mean (GM).

Table 7. Results for $k = 1$ with $G_1$ as performance measure over 5 runs. The standard deviation is indicated after the $\pm$ sign and the best results on each dataset is indicated in bold. Only the mean value when $k = 3$ is shown in the last line.

| DATASETS | $k$−NN | DUP$k$−NN | W$k$−NN | CW$k$−NN | KRNN | LMNN | $\gamma k$−NN |
|---|---|---|---|---|---|---|---|
| BALANCE | 84.5±2.2 | 84.5±2.2 | 84.5±2.2 | 85.4±1.5 | **88.4±1.2** | 84.2±4.6 | 85.4±1.5 |
| AUTOPMG | 76.9±7.4 | 76.9±7.4 | 76.9±7.4 | 76.4±6.3 | **82.9±2.9** | 77.7±3.1 | 82.3±3.2 |
| IONOSPHERE | 81.4±3.9 | 81.4±3.9 | 81.4±3.9 | 84.2±3.0 | 83.8±2.9 | 82.0±3.1 | **89.2±4.3** |
| PIMA | 56.1±4.6 | 56.1±4.6 | 56.1±4.6 | 61.3±3.6 | **62.6±3.8** | 56.1±2.6 | 61.8±4.9 |
| GLASS | 71.0±8.2 | 71.0±8.2 | 71.0±8.2 | 73.8±5.9 | **77.4±7.9** | 69.6±7.6 | 68.5±5.0 |
| GERMAN | 38.1±5.0 | 38.1±5.0 | 38.1±5.0 | 41.1±3.6 | 43.8±4.0 | 41.1±3.8 | **55.1±1.1** |
| YEAST1 | 52.6±2.6 | 52.6±2.6 | 52.6±2.6 | 53.8±3.5 | 52.9±2.0 | 51.4±3.6 | **56.7±2.4** |
| HABERMAN | 23.8±7.0 | 23.8±7.0 | 23.8±7.0 | 36.0±10.3 | 33.3±7.6 | 25.1±7.4 | **51.4±4.9** |
| VEHICLE3 | 51.1±3.4 | 51.1±3.4 | 51.1±3.4 | 51.5±3.7 | 56.5±3.6 | 54.8±3.7 | **59.1±1.2** |
| HAYES | 68.9±8.6 | 68.9±8.6 | 68.9±8.6 | **88.0±4.6** | 77.9±7.6 | 79.1±11.3 | 75.6±10.2 |
| SEGMENTATION | 88.8±2.9 | 88.8±2.9 | 88.8±2.9 | 89.0±2.8 | 87.5±1.5 | **90.9±2.9** | 88.8±2.9 |
| ABALONE8 | 21.4±1.3 | 21.4±1.3 | 21.4±1.3 | 33.1±0.9 | 23.9±1.2 | 21.9±1.7 | **38.5±1.2** |
| YEAST3 | 65.4±2.3 | 65.4±2.3 | 65.4±2.3 | 64.9±1.8 | **69.7±1.0** | 63.5±1.2 | 67.0±2.6 |
| ECOLI3 | 53.0±9.8 | 53.0±9.8 | 53.0±9.8 | 56.7±8.0 | **61.7±6.2** | 54.2±10.0 | 54.9±10.7 |
| PAGEBLOCKS | **81.7±2.6** | **81.7±2.6** | **81.7±2.6** | 81.3±2.5 | 81.3±4.4 | 81.5±3.2 | 81.2±2.2 |
| SATIMAGE | 67.7±3.6 | 67.7±3.6 | 67.7±3.6 | 69.0±3.2 | **69.3±2.7** | 69.0±4.5 | 67.2±4.0 |
| YEAST-0-5-6-7-9VS4 | 42.2±10.4 | 42.2±10.4 | 42.2±10.4 | 51.9±4.3 | **52.7±7.5** | 46.5±14.8 | 47.6±14.4 |
| LIBRAS | 80.2±8.5 | 80.2±8.5 | 80.2±8.5 | 80.2±8.5 | 74.3±6.0 | 75.9±4.5 | **88.0±8.3** |
| YEAST-1VS7 | 48.7±5.8 | 48.7±5.8 | 48.7±5.8 | 29.2±7.3 | 49.3±8.8 | **50.0±12.5** | 47.9±7.1 |
| ARRYTHMIA | 17.1±21.7 | 17.1±21.7 | 17.1±21.7 | 16.7±21.4 | 16.7±21.4 | **20.5±22.0** | 17.1±21.7 |
| SOLAR-FLARE-M0 | 16.5±6.8 | 16.5±6.8 | 16.5±6.8 | 24.3±1.7 | 21.5±6.4 | 11.7±7.9 | **26.7±3.2** |
| OIL | 54.6±10.8 | 54.6±10.8 | 54.6±10.8 | 58.0±9.4 | 57.1±4.8 | **66.6±10.0** | 59.7±10.1 |
| YEAST4 | 31.2±11.6 | 31.2±11.6 | 31.2±11.6 | 35.7±2.1 | **43.6±4.1** | 35.0±13.2 | 40.2±8.7 |
| REDWINEQUALITY4 | 10.8±5.8 | 10.8±5.8 | 10.8±5.8 | 15.2±3.7 | 13.3±7.1 | 12.2±7.0 | **19.2±5.5** |
| YEAST5 | 70.5±11.0 | 70.5±11.0 | 70.5±11.0 | 60.2±8.1 | 64.9±8.1 | **72.0±7.4** | 69.7±9.6 |
| YEAST6 | 50.0±13.9 | 50.0±13.9 | 50.0±13.9 | 35.3±5.3 | **52.9±9.9** | 46.6±18.1 | 47.9±11.5 |
| ABALONE17 | 15.0±9.7 | 15.0±9.7 | 15.0±9.7 | 15.0±5.7 | **18.3±7.8** | 17.0±7.2 | 14.7±2.4 |
| ABALONE20 | 00.0±0.0 | 00.0±0.0 | 00.0±0.0 | **8.3±5.2** | 7.6±7.5 | 00.0±0.0 | 1.2±2.4 |
| MEAN (K=1) | 50.7 | 50.7 | 50.7 | 52.7 | 54.5 | 52.0 | **55.8** |
| MEAN (K=3) | 50.8 | 55.2 | 51.4 | 54.0 | 54.6 | 52.7 | **55.9** |

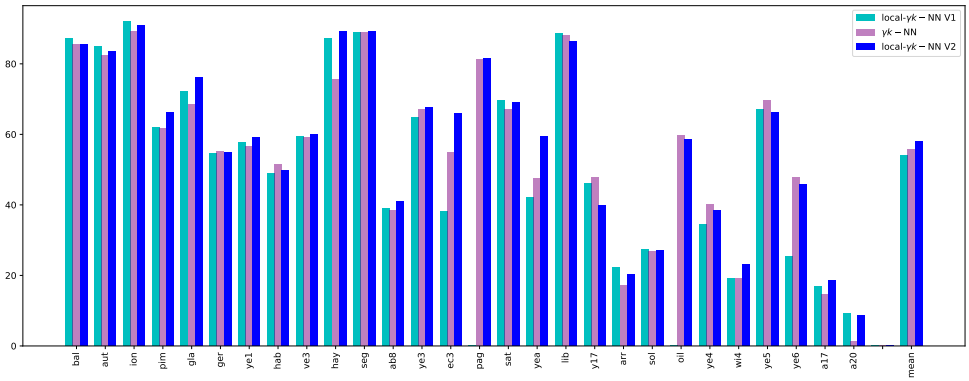


Fig. 13. Comparison of $\gamma 1$−NN with the two versions of local-$\gamma 1$−NN, in terms of G-measure ($G_1$).

## References

1. C. C. Aggarwal, *Outlier Analysis* (Springer International Publishing, 2017).
2. V. Chandola, A. Banerjee and V. Kumar, Anomaly detection: A survey, *ACM Comput. Surv.* (2009).
3. R. A. Bauder, T. M. Khoshgoftaar and T. Hasanin, Data sampling approaches with severely imbalanced big data for medicare fraud detection, in *2018 IEEE 30th international conference on tools with artificial intelligence (ICTAI)* (IEEE, 2018), pp. 137–142.
4. K. H. Brodersen, C. S. Ong, K. E. Stephan and J. M. Buhmann, The balanced accuracy and its posterior distribution, in *20th International Conference on Pattern Recognition* (IEEE, 2010).
5. C. Ferri, J. Hernández-Orallo and R. Modroiu, An experimental comparison of performance measures for classification, *Pattern Recognition Letters* **30** (2009).
6. H. Steck, Hinge rank loss and the area under the roc curve, in *Machine Learning: ECML 2007*, eds. J. N. Kok, J. Koronacki, R. L. d. Mantaras, S. Matwin, D. Mladenič and A. Skowron (Springer, 2007).
7. K. Bascol, R. Emonet, É. Fromont, A. Habrard, G. Metzler and M. Sebban, From cost-sensitive to tight f-measure bounds, in *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019*, eds. K. Chaudhuri and M. Sugiyama **89**, (PMLR, 2019), pp. 1245–1253.
8. J. Fréry, A. Habrard, M. Sebban, O. Caelen and L. He-Guelton, Efficient top rank optimization with gradient boosting for supervised anomaly detection, in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part I* (Springer, 2017).
9. A. Fernández, S. Garcia, F. Herrera and N. V. Chawla, Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary, *Journal of Artificial Intelligence Research* **61** (2018).
10. I. Tomek, Two modifications of cnn., *IEEE Transactions on Systems Man and Communications* **6** (1976) 769–772.
11. N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, *Journal of artificial intelligence research* **16** (2002).
12. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville and Y. Bengio, Generative adversarial nets, in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada* ( 2014).
13. M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet and S. Gelly, Assessing generative models via precision and recall, in *Advances in Neural Information Processing Systems 31* (NeurIPS, 2018)
14. T. Cover and P. Hart, Nearest neighbor pattern classification, *IEEE Transactions on Information Theory* **13** (1967).
15. U. v. Luxburg and O. Bousquet, Distance-based classification with lipschitz functions, *Journal of Machine Learning Research* **5** (2004).
16. A. Kontorovich and R. Weiss, A Bayes consistent 1-NN classifier, in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* **38**, (PMLR, 2015).
17. A. Kontorovich, S. Sabato and R. Urner, Active nearest-neighbor learning in metric spaces, in *Advances in Neural Information Processing Systems 29* (NIPS, 2016)
18. K. Q. Weinberger and L. K. Saul, Distance metric learning for large margin nearest neighbor classification, *Journal of Machine Learning Research* **10** (2009).
19. J. V. Davis, B. Kulis, P. Jain, S. Sra and I. S. Dhillon, Information-theoretic metric

learning, in *ICML* (ACM, 2007).
20. A. Bellet, A. Habrard and M. Sebban, *Metric Learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Vol. 2015 (Morgan & Claypool Publishers, 2015).
21. A. Bellet, A. Habrard and M. Sebban, A survey on metric learning for feature vectors and structured data, *CoRR* **abs/1306.6709** (2013).
22. R. Viola, R. Emonet, A. Habrard, G. Metzler, S. Riou and M. Sebban, An adjusted nearest neighbor algorithm maximizing the f-measure from imbalanced data, in *In Proceedings of the 31st International Conference on Tools with Artificial Intelligence (ICTAI-2019)* (IEEE, 2019).
23. C. J. V. Rijsbergen, *Information Retrieval* 1979.
24. S. Gee, *Fraud and fraud detection: a data analytics approach* 2014.
25. R. Espíndola and N. Ebecken, On extending f-measure and g-mean metrics to multi-class problems, *WIT Transactions on Information and Communication Technologies* **35** (2005).
26. M. Kubat and S. Matwin, Addressing the curse of imbalanced training sets: One-sided selection, in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, ed. D. H. Fisher (Morgan Kaufmann, 1997), pp. 179–186.
27. S. A. Dudani, The distance-weighted k-nearest-neighbor rule, *IEEE Transactions on Systems, Man, and Cybernetics* **4** (1976).
28. W. Liu and S. Chawla, Class confidence weighted knn algorithms for imbalanced data sets, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (Springer, 2011), pp. 345–356.
29. X. Zhang and Y. Li, A positive-biased nearest neighbour algorithm for imbalanced classification, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (Springer, 2013), pp. 293–304.
30. X. Zhang, Y. Li, R. Kotagiri, L. Wu, Z. Tari and M. Cheriet, Krnn: k rare-class nearest neighbour classification, *Pattern Recognition* **62** (2017) 33 – 44.
31. R. Barandela, J. S. Sánchez, V. Garca and E. Rangel, Strategies for learning in class imbalance problems, *Pattern Recognition* **36** (2003).
32. H. Han, W.-Y. Wang and B.-H. Mao, Borderline-smote: a new over-sampling method in imbalanced data sets learning, in *International conference on intelligent computing* (Springer, 2005).
33. H. He, Y. Bai, E. A. Garcia and S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (IEEE, 2008).
34. D. L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, *IEEE Transactions on Systems, Man, and Cybernetics* **3** (1972).
35. C. Elkan, The foundations of cost-sensitive learning, in *International joint conference on artificial intelligence* (Morgan Kaufmann, 2001), pp. 973–978.
36. Z. Qin, A. T. Wang, C. Zhang and S. Zhang, Cost-sensitive classification with k-nearest neighbors, in *International Conference on Knowledge Science, Engineering and Management* (Springer, 2013), pp. 112–131.
37. S. Zhang, Cost-sensitive knn classification, *Neurocomputing* **391** (2019) 234–242.
38. Y. LeCun and C. Cortes, MNIST handwritten digit database (2010).
39. R. Viola, R. Emonet, A. Habrard, G. Metzler and M. Sebban, Learning from few positives: a provably accurate metric learning algorithm to deal with imbalanced data (ijcai), in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (ijcai.org, 2020).