Une version corrigée de l'algorithme des plus proches voisins pour l'optimisation de la F-mesure dans un contexte déséquilibré

Rémi Viola, Rémi Emonet, Amaury Habrard, Guillaume Metzler, Sébastien Riou, Marc Sebban

Une version corrigée de l'algorithme des plus proches voisins pour l'optimisation de la F-mesure dans un contexte déséquilibré.

Rémi Viola[1,2], Rémi Emonet[1], Amaury Habard[1], Guillaume Metzler[1,3], Sébastien Riou[2], et Marc Sebban[1,2]

[1]Univ. Lyon, UJM Saint-Etienne, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, Saint-Etienne, France
[2]Direction Générances des Finances Publiques, Ministère de l'Economie et des Finances, France
[3]Blitz inc., France

29 mai 2019

Résumé

Dans le présent papier, nous proposons une approche basée sur l'algorithme des plus proches voisins pour de l'apprentissage dans un contexte déséquilibré. Dans un tel contexte, les exemples de la classe minoritaire sont au centre de l'attention et nécessitent des critères d'optimisation spécifiques pour nous permettre de les détecter, comme la F-mesure. Reposant sur des fondements géométriques, nous présentons un algorithme qui pondère la distance entre un nouvel exemple et les exemples positifs de la classe minoritaire. Cela entraîne une modification des régions de Voronoï et donc de la frontière de décision. Une analyse théorique de cette pondération explique comment il est possible de réduire le taux de faux négatifs tout en contrôlant le taux de faux positifs. Les expériences menées sur plusieurs jeux de données publiques, ainsi que sur de grands jeux de données du Ministère de l'Economie et des Finances sur la détection de fraude à l'impôt, mettent en évidence l'efficacité de la méthode en dépit de sa simplicité. En outre, elle se révèle d'autant plus intéressante et performante lorsque qu'elle est combinée à des méthodes d'échantillonage.

Mots-clef : Plus proches voisins, F-mesure, Apprentissage dans un contexte déséquilibré.

1 Introduction

Intrusion detection, health care insurance or bank fraud identification, and more generally anomaly detection, *e.g.* in medicine or in industrial processes, are tasks requiring to address the challenging problem of learning from imbalanced data [Agg17, CBK09]. 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 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) or the *F-measure* to cite a few [FHOM09, Ste07]. The main pitfalls related to such a strategy concern the difficulty to directly optimize non smooth, non separable and non convex measures. 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 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 [FHS$^+$17] with a smooth surrogate of the Mean-AP. The second family of methods is based on the modification of the

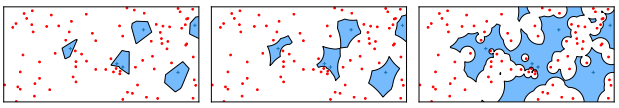

FIGURE 1 – Toy imbalanced dataset : On the left, the Voronoi regions around the positives are small. The risk to generate false negatives (FN) at test time is large. On the right : by increasing too much the regions of influence of the positives, the probability to get false positives (FP) grows. In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

distribution of the training data using sampling strategies [FGHC18]. This is typically achieved by removing examples from the majority class, as done, *e.g.*, in *ENN* or *Tomek's Link* [Tom76], and/or by adding examples from the minority class, as in SMOTE [CBHK02] and its variants, or by resorting to generative adversarial models [GPM$^+$14]. One peculiarity of imbalanced datasets can be interpreted from a geometric perspective. As illustrated in Fig. 1 (left) which shows the Voronoi cells on an artificial imbalance 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 is high, leading to a low F-measure, the criterion we focus on in this paper, defined as the harmonic mean of the $Precision = \frac{TP}{TP+FP}$ and the $Recall = \frac{TP}{TP+FN}$, where FP (resp. FN) is the number of false positives (resp. negatives) and TP the number of true positives. Note that increasing the regions of influence of the positives would allow us to reduce FN and improve the F-measure. However, not controlling the expansion of these regions may have a dramatic impact on FP, and so on the F-Measure, as illustrated in Fig. 1 (right).

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 low F-Measure. A natural way to increase the influence of positives may consist in using generative models (like GANs [GPM$^+$14]) to sample new artificial examples, mimicking the negative training samples. However, beyond the issues related to the parameter tuning, the computation burden and the complexity of such a method, using GANs to optimize the precision and recall is still an open problem (see [SBL$^+$18] for a recent paper on this topic). We show in this paper that a much simpler strategy can be used by modifying the distance exploited in a k-nearest neighbor (NN) algorithm [CH67] which enjoys many interesting advantages, including its simplicity, its capacity to approximate asymptotically any locally regular density, and its theoretical rootedness [LB04, KW15, KSU16]. k-NN also benefited from many algorithmic advances during the past decade in the field of metric learning, aiming at optimizing under constraints the parameters of a metric, typically the Mahalanobis distance, as done in LMNN [WS09] or ITML [DKJ$^+$07] (see [BHS15] 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, 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 formally, we show in this paper that the optimization of the F-Measure is facilitated by weighting 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.

The rest of this paper is organized as follows. Section 2 is dedicated to the introduction of our notations. The related work is presented in Section 3. Section 4 is devoted to the presentation of our method. We perform an extensive experimental study in Section 5 on many imbalanced datasets, including non public data from the French Ministry of Economy and Finance on a tax fraud detection task. We give evidence about 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 as-

sume 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. Two measures are usually used : the *Recall* or *True Positive Rate* which measures the capacity of the model to recall/detect positive examples, and the *Precision* which is the confidence in the prediction of a positive label :

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

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 : the *F-measure* [Rij79] (or F_1 score), which is widely used in fraud and anomaly detection, and more generally in imbalanced classification [Gee14].

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}}.$$

Note that F_1 considers the Precision and Recall equally.

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 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 [Dud76] 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 $\text{kNN}(\mathbf{x})$ stands for the set of the k nearest neighbors of $\mathbf{x}$.

In [BSGR03], the authors account both the label and the distance to the neighbors $(\mathbf{x}_i, y_i)$ to define a weighted 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 weighting the distance to the examples according to their label. However, three main differences justify why our method will be better in the experiments : (i) d' is fixed in advance while we will adapt the weight that optimizes the F- 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 this will be intrinsically captured in our method by optimizing the weight given the p-dimensional space ; (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 weight differently 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 : m_-/m_+. Thus, it can be seen as a *uniform* over-sampling 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*. LMNN [WS09] or ITML [DKJ+07] 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 semidefinite (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 weighting method is a specific case of metric learning which looks for a diagonal matrix - applied only when comparing a query to a positive example - and that behaves well in terms of F-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 [FGHC18] 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 [CBHK02] (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 [HWM05] 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 points are generated if the number n of negatives in the k-neighborhood is such that $k/2 \leq n \leq k$.

The Adaptive Synthetic [HBGL08] (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 [Wil72] (ENN) algorithm on the top of SMOTE. After SMOTE has generated data, the ENN algorithm removes data that are misclassified by their k nearest neighbors. The second one combines SMOTE with Tomek's link [Tom76]. A Tomek's link 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 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 stay in this line of research. Rather than generating new examples, that would have a negative impact from a complexity perspective, we locally modify the density around the positive points. This is achieved by rescaling the distance between a test sample and the positive training examples. We will 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 A Corrected $k-$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 propose 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_γ that we propose is defined, using an underlying distance d (e.g. the euclidean distance) as follows :

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

As we will tune the γ 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 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 γ is simply tuned such as maximizing the non convex F-Measure. Lastly, and most importantly, it is applied only when comparing the query to positive examples. As such, d_γ is not a proper distance, however, it is exactly this which allows it to compensate for the class imbalance. In the binary setting, there is no need to have a γ 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 γ.

Before formalizing the $\gamma k-$NN algorithm that will leverage the distance d_γ, we illustrate in Fig. 2, on 2D data, the decision boundary induced by a nearest neighbor binary classifier that uses d_γ. 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

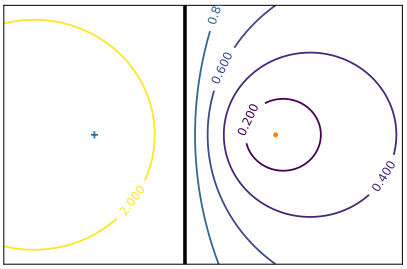

FIGURE 2 – Evolution of the decision boundary based on d_γ, for a 1-NN classifier, on a 2D dataset with one positive (resp. negative) instance represented by a blue cross (resp. orange point). The value of γ is given on each boundary ($\gamma = 1$ on the thick line).

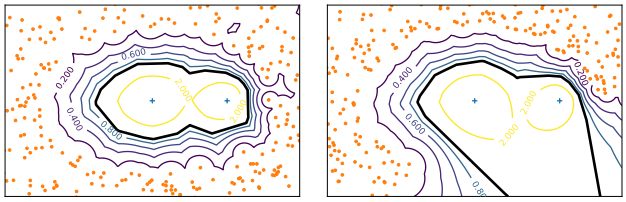

FIGURE 3 – Behavior of the decision boundary according to the γ value for the 1-NN classifier on two toy datasets. The positive points are represented by blue crosses and the negatives by orange points. The black line represents the standard decision boundary for the 1-NN classifier, i.e. when $\gamma = 1$.

the value of γ below 1 brings the decision boundary closer to the negative point, and eventually tends to surround it very closely. In Fig 3, two more complex datasets are shown, each with two positive points and several negative examples. As intuited, we see that the γ parameter allows to control how much we want to push the boundary towards negative examples.

We can now introduce the $\gamma k-$NN algorithm (see Algo 1) that is parameterized by a γ parameter. It has the same overall complexity as $k-$NN. The first step to classify 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 γ, to obtain d_γ. 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_γ does not define a proper distance, we can still use any existing fast nearest neighbor search algorithm, because the actual search is done (twice but) only using the original distance d.

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 γ, 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

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

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

$y \leftarrow +$ if $\left|\mathcal{NN}_\gamma \cap \mathcal{NN}^+\right| \geq \frac{k}{2}$ else $-$ // 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 the γ parameter of our corrected version of the $k-$NN algorithm. To this aim, we study what impact γ 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 a k-NN.

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 $\gamma \leq 1$,*

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

Proof 1 *(sketch of proof) Let ϵ 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,*

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

while

$$FN(\mathbf{z}) = \left(1 - P(\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{z}))\right)^{m_+}. \tag{2}$$

Solving (1) $\leq$ (2) 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.(1) and (2) 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 γ.

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 $\gamma \geq 1$,*

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

Proof 2 *(sketch of proof) Using the same idea as before, we get :*

$$\begin{aligned} FP_\gamma(\mathbf{z}) &= \prod_{\mathbf{x}' \in S_-} \left(1 - P(\mathbf{x}' \in \mathcal{S}_{\gamma\epsilon}(\mathbf{z}))\right), \\ &= \left(1 - P(\mathbf{x}' \in \mathcal{S}_{\gamma\epsilon}(\mathbf{z}))\right)^{m_-} \qquad (3) \end{aligned}$$

while

$$FP(\mathbf{z}) = \left(1 - P(\mathbf{x}' \in \mathcal{S}_{\epsilon}(\mathbf{z}))\right)^{m_-}. \qquad (4)$$

Solving (3) $\leq$ (4) 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 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 $\gamma \in]0, 1[$. As we will see in the experimental section, the more imbalance the datasets, the smaller the optimal γ, confirming the previous conclusion.

5 Experiments

In this section, we present an experimental evaluation of our method on public and real private datasets with comparisons to classic distance-based methods and state of the art sampling strategies able to deal with imbalanced data. All results are reported using $k = 3$. The experiments with $k = 1$, fully reported in the supplementary material, follow the same trends.

5.1 Experimental setup

For the experiments, we use several public datasets from the classic UCI [1] and KEEL [2] repositories. We also use eleven real fraud detection datasets provided by the General Directorate of Public Finances (DGFiP) which is part of the French central public administration related to the French Ministry for the Economy and Finance. These private datasets correspond to data coming from tax and VAT declarations of French companies and are used for tax fraud detection purpose covering declaration of over-valued, fictitious or prohibited charges, wrong turnover reduction or particular international VAT frauds such as "VAT carousels". The DGFiP performs about 50,000 tax audits per year within a panel covering more than 3,000,000 companies. Being able to select the right companies to control each year is a crucial issue with a potential high societal impact. Thus, designing efficient imbalance learning methods is key. The main properties of the datasets are summarized in Table 5.1, including the imbalance ratio (IR).

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 F-measure F_1. In a first series of experiments, we compare our method, named $\gamma k-$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) [Dud76],
- the class weighted version of $k-$NN (cw$k-$NN) [BSGR03],
- the $k-$NN version where each positive is duplicated according to the IR of the dataset (dup$k-$NN),
- the metric learning method LMNN [WS09].

1. `https://archive.ics.uci.edu/ml/datasets.html`
2. `https://sci2s.ugr.es/keel/datasets.php`

DATASETS	SIZE	DIM	%+	%−	IR
BALANCE	625	4	46.1	53.9	1.2
AUTOMPG	392	7	37.5	62.5	1.7
IONOSPHERE	351	34	35.9	64.1	1.8
PIMA	768	8	34.9	65.1	1.9
WINE	178	13	33.1	66.9	2
GLASS	214	9	32.7	67.3	2.1
GERMAN	1000	23	30	70	2.3
VEHICLE	846	18	23.5	76.5	3.3
HAYES	132	4	22.7	77.3	3.4
SEGMENTATION	2310	19	14.3	85.7	6
ABALONE8	4177	10	13.6	86.4	6.4
YEAST3	1484	8	11	89	8.1
PAGEBLOCKS	5473	10	10.2	89.8	8.8
SATIMAGE	6435	36	9.7	90.3	9.3
LIBRAS	360	90	6.7	93.3	14
WINE4	1599	11	3.3	96.7	29.2
YEAST6	1484	8	2.4	97.6	41.4
ABALONE17	4177	10	1.4	98.6	71.0
ABALONE20	4177	10	0.6	99.4	159.7
DGFIP 19 2	16643	265	35.1	64.9	1.9
DGFIP 9 2	440	173	24.8	75.2	3
DGFIP 4 2	255	82	20.8	79.2	3.8
DGFIP 8 1	1028	255	17.8	82.2	4.6
DGFIP 8 2	1031	254	17.9	82.1	4.6
DGFIP 9 1	409	171	16.4	83.6	5.1
DGFIP 4 1	240	76	16.2	83.8	5.2
DGFIP 16 1	789	162	10.3	89.7	8.7
DGFIP 16 2	786	164	9.9	90.1	9.1
DGFIP 20 3	17584	294	5	95	19
DGFIP 5 3	19067	318	3.9	96.1	24.9

TABLE 1 – Information about the studied datasets sorted by imbalance ratio. The first part refers to the public datasets, the second one describes the *DGFiP* private datasets.

We set the number of nearest neighbors to $k = 3$ for all methods. The hyperparameter μ of *LMNN*, weighting the impact of impostor constraints (see [WS09] for more details), is tuned in the range $[0, 1]$ using a step of 0.1. Our γ parameter is tuned in the range $[0, 1]$ [3] using a step of 0.1.

In a second series of experiments, we compare our method to the 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 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. Other parameters of these methods are the default ones used by the package *ImbalancedLearn* of *Scikit-learn*.

5.2 Results

The results on the public datasets using distance-based methods are provided in Table 5.2. Overall, our $\gamma k-$NN approach performs much better than its competitors by achieving an improvement of at least 3 points on average, compared to the 2nd best method (DUP$k-$NN). The different $k-$NN versions fail globally to provide models efficient whatever the imbalance ratio. The metric learning approach LMNN is competitive when IR is smaller than 10 (although algorithmically more costly). Beyond, it faces some difficulties to find a relevant projection space due to the lack of positive data. The efficiency of $\gamma k-$NN is not particularly sensitive to the imbalance ratio.

The results for our second series of experiments, focusing on sampling strategies, are reported on Fig. 4. We compare each of the 5 sampling methods with the average performances of $3-$NN and γk-NN obtained over the 19 public datasets reported in Table 5.2. Additionally, we also use $\gamma k-$NN on the top of the sampling methods to evaluate how both strategies are complementary. However, in this scenario, we propose to learn a different γ value to be used with the synthetic positives. Indeed, some of them may be generated in some true negative areas and in this situation it might be more appropriate to decrease the influence of such synthetic examples. The γ parameter for these examples is then tuned in the range $[0, 2]$ using a step of 0.1. If one can easily observe that all the oversampling strategies improve the classic $k-NN$, none of them is better than our γk-NN method showing that our approach is able to deal efficiently with imbalanced data. Moreover, we are able to improve the efficiency of γk-NN when it is coupled with an oversampling strategy. The choice of the oversampler does not really influence the results. The gains obtained by using a sampling method with γk-NN for each dataset is illustrated in Fig. 6 (left).

To study the influence of using two γ parameters when combined with an oversampling strategy, we show an illustration (Fig. 5 (top)) of the evolution of the F-measure with respect to the γ values for synthetic and real positive instances. The best F-measure is achieved when the γ on real positives is smaller than 1 and when the γ on synthetic positives is greater than 1, justifying the interest of using two parameterizations of γ. In Fig. 5 (bottom), we show how having two γ values gives the flexibility to independently control the increased influence of real positives and the one of artificial

3. We experimentally noticed that using a larger range for γ 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.

DATASETS	3−NN	DUPk−NN	Wk−NN	CWk−NN	LMNN	γk−NN
BALANCE	0.954(0.017)	0.954(0.017)	0.957(0.017)	0.961(0.010)	**0.963**(0.012)	0.954(0.029)
AUTOMPG	0.808(0.077)	0.826(0.033)	0.810(0.076)	0.815(0.053)	0.827(0.054)	**0.831**(0.025)
IONOSPHERE	0.752(0.053)	0.859(0.021)	0.756(0.060)	0.799(0.036)	0.890(0.039)	**0.925**(0.017)
PIMA	0.500(0.056)	0.539(0.033)	0.479(0.044)	0.515(0.037)	0.499(0.070)	**0.560**(0.024)
WINE	0.881(0.072)	0.852(0.057)	0.881(0.072)	0.876(0.080)	**0.950**(0.036)	0.856(0.086)
GLASS	0.727(0.049)	0.733(0.061)	0.736(0.052)	0.717(0.055)	0.725(0.048)	**0.746**(0.046)
GERMAN	0.330(0.030)	0.449(0.037)	0.326(0.030)	0.344(0.029)	0.323(0.054)	**0.464**(0.029)
VEHICLE	0.891(0.044)	0.867(0.027)	0.891(0.044)	0.881(0.021)	**0.958**(0.020)	0.880(0.049)
HAYES	0.036(0.081)	0.183(0.130)	0.050(0.112)	0.221(0.133)	0.036(0.081)	**0.593**(0.072)
SEGMENTATION	0.859(0.028)	0.862(0.018)	0.877(0.028)	0.851(0.022)	**0.885**(0.034)	0.848(0.025)
ABALONE8	0.243(0.037)	0.318(0.013)	0.241(0.034)	0.330(0.015)	0.246(0.065)	**0.349**(0.018)
YEAST3	0.634(0.066)	0.670(0.034)	0.634(0.066)	**0.699**(0.015)	0.667(0.055)	0.687(0.033)
PAGEBLOCKS	0.842(0.020)	0.850(0.024)	0.849(0.019)	0.847(0.029)	**0.856**(0.032)	0.844(0.023)
SATIMAGE	0.454(0.039)	0.457(0.027)	0.454(0.039)	0.457(0.023)	**0.487**(0.026)	0.430(0.008)
LIBRAS	**0.806**(0.076)	0.788(0.187)	**0.806**(0.076)	0.789(0.097)	0.770(0.027)	0.768(0.106)
WINE4	0.031(0.069)	**0.090**(0.086)	0.031(0.069)	0.019(0.042)	0.000(0.000)	**0.090**(0.036)
YEAST6	0.503(0.302)	0.449(0.112)	0.502(0.297)	0.338(0.071)	0.505(0.231)	**0.553**(0.215)
ABALONE17	0.057(0.078)	**0.172**(0.086)	0.057(0.078)	0.096(0.059)	0.000(0.000)	0.100(0.038)
ABALONE20	0.000(0.000)	0.000(0.000)	0.000(0.000)	**0.067**(0.038)	0.057(0.128)	0.052(0.047)
MEAN	0.543(0.063)	0.575(0.053)	0.544(0.064)	0.559(0.046)	0.560(0.053)	**0.607**(0.049)

TABLE 2 – Results for 3−NN on the public datasets. The values correspond to the mean F-measure F_1 over 5 runs. The standard deviation is indicated between brackets. The best result on each dataset is indicated in bold.

positives.

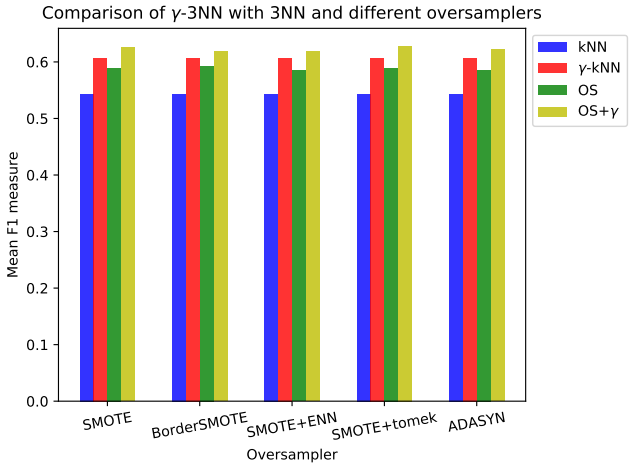

FIGURE 4 – Comparison of different sampling strategies averaged over the 19 public datasets. *OS* refers to the results of the corresponding sampling strategy and $OS+\gamma$ to the case when the sampling strategy is combined with γk-NN. k−NN and γk−NN refers to the results of these methods without oversampling as obtained in Table 5.2. (numerical values for these graphs are provided in supplementary material)

We now propose a study on the influence of the imbalance ratio on the optimal γ-parameter. We consider the *Balance* dataset which has the smallest imbalance

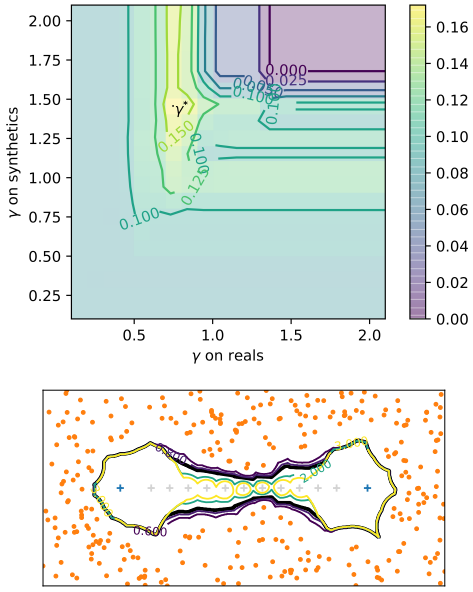

FIGURE 5 – (Top) An example of heatmap that shows the best couple of γ for the OS+γk−NN strategy on the yeast6 dataset with SMOTE and Tomek's link. (Bottom) Illustration, on a toy dataset, of the effect of varying the γ for generated positive points (in grey) while keeping a fixed $\gamma = 0.4$ for real positive points.

ratio that we increase by iteratively randomly under-

sampling the minority class over the training set. We report the results on Fig. 6 (right). As expected, we can observe that the optimal γ value decreases when the imbalance increases. However, note that from a certain IR (around 15), γ stops decreasing to be able to keep a satisfactory F-Measure.

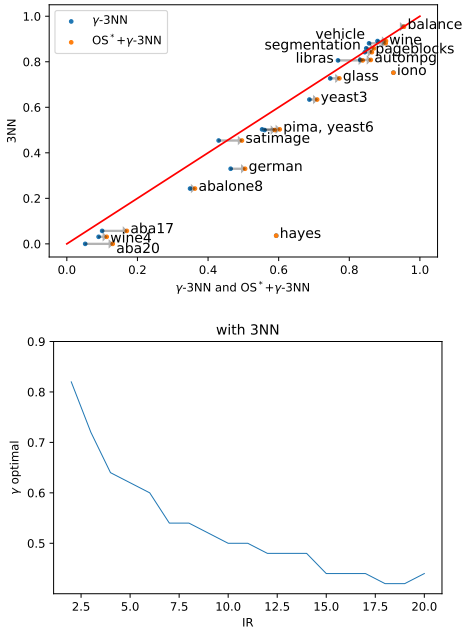

FIGURE 6 – (Top) Comparison of k-NN with (i) $\gamma k-$NN (points in blue) and (ii) $\gamma k-$NN coupled with the best sampling strategy (OS*) (points in orange) for each dataset and for $k = 3$. Points below the line $y = x$ means that k-NN is outperformed. (Bottom) Evolution of the optimal γ value with respect to the IR for $k = 3$.

The results for the real datasets of the DGFiP are available in Table 3. Note that only the SMOTE algorithm is reported here since the other oversamplers have comparable performances. The analysis of the results leads to observations similar as the ones made for the public datasets. Our $\gamma - k$NN approach outperforms classic $k-$NN and is better than the results obtained by the SMOTE strategy. Coupling the SMOTE sampling method with our distance correction γk-NN allows us to improve the global performance showing the applicability of our method on real data.

6 Conclusion

In this paper, we have proposed a new strategy that addresses the problem of learning from imbalanced datasets, based on the $k-$NN algorithm and that modifies the distance to the positive examples. It has been shown to outperform its competitors in term of F_1-measure. Furthermore, the proposed approach is complementary to oversampling strategies and can even increase their performance. Our $\gamma k-$NN algorithm, despite its simplicity, is highly effective even on real data sets.

Two lines of research deserve future investigations. We can note that tuning γ is equivalent to building a diagonal matrix (with γ^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 metric learning algorithm dedicated to optimizing a PSD matrix under F-Measure-based constraints. If one can learn such a matrix, the second perspective will consist in deriving generalization guarantees over the learned matrix. In addition, making γ non-stationary (a $\gamma(\mathbf{x})$ that smoothly varies in $\mathcal{X}$) would increase the model flexibility.

Références

[Agg17] Charu C. Aggarwal. *Outlier Analysis*. Springer International Publishing, 2017.

[BHS15] Aurélien Bellet, Amaury Habrard, and Marc Sebban. *Metric Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2015.

[BSGR03] Ricardo Barandela, José Salvador Sánchez, V Garca, and Edgar Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36, 2003.

[CBHK02] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote : synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 2002.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection : A survey. *ACM Comput. Surv.*, 2009.

[CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13, 1967.

[DKJ+07] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.

DATASETS	3−NN	$\gamma k-$NN	SMOTE	SMOTE+$\gamma k-$NN
DGFIP19 2	0,454(0.007)	0,528(0.005)	0,505(0.010)	**0,529**(0.003)
DGFIP9 2	0,173(0.074)	0,396(0.018)	0,340(0.033)	**0,419**(0.029)
DGFIP4 2	0,164(0.155)	0,373(0.018)	0,368(0.057)	**0,377**(0.018)
DGFIP8 1	0,100(0.045)	**0,299**(0.010)	0,278(0.043)	**0,299**(0.011)
DGFIP8 2	0,140(0.078)	0,292(0.028)	**0,313**(0.048)	0,312(0.021)
DGFIP9 1	0,088(0.090)	0,258(0.036)	0,270(0.079)	**0,288**(0.026)
DGFIP4 1	0,073(0.101)	0,231(0.139)	0,199(0.129)	**0,278**(0.067)
DGFIP16 1	0,049(0.074)	0,166(0.065)	0,180(0.061)	**0,191**(0.081)
DGFIP16 2	0,210(0.102)	0,202(0.056)	0,220(0.043)	**0,229**(0.026)
DGFIP20 3	0,142(0.015)	0,210(0.019)	0,199(0.015)	**0,212**(0.019)
DGFIP5 3	0,030(0.012)	0,105(0.008)	**0,110**(0.109)	0,107(0.010)
MEAN	0,148(0.068)	0,278(0.037)	0,271(0.057)	**0,295**(0.028)

TABLE 3 – Results for 3−NN on the DGFiP datasets. The values correspond to the mean F-measure F_1 over 5 runs. The best result on each dataset is indicated in bold while the second is underlined.

[Dud76] Sahibsingh A Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, 4, 1976.

[FGHC18] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data : Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61, 2018.

[FHOM09] César Ferri, José Hernández-Orallo, and R Modroiu. An experimental comparison of performance measures for classification. *PRL*, 30, 2009.

[FHS+17] Jordan Fréry, Amaury Habrard, Marc Sebban, Olivier Caelen, and Liyun He-Guelton. Efficient top rank optimization with gradient boosting for supervised anomaly detection. In *ECML-PKDD*, 2017.

[Gee14] Sunder Gee. *Fraud and fraud detection : a data analytics approach.* 2014.

[GPM+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[HBGL08] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn : Adaptive synthetic sampling approach for imbalanced learning. In *IJCNN*, 2008.

[HWM05] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote : a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, 2005.

[KSU16] Aryeh Kontorovich, Sivan Sabato, and Ruth Urner. Active nearest-neighbor learning in metric spaces. In *NIPS*. 2016.

[KW15] Aryeh Kontorovich and Roi Weiss. A Bayes consistent 1-NN classifier. In *ICAIS*, volume 38, 2015.

[LB04] Ulrike von Luxburg and Olivier Bousquet. Distance-based classification with lipschitz functions. *JMLR*, 5, 2004.

[Rij79] C. J. Van Rijsbergen. *Information Retrieval.* 1979.

[SBL+18] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *NeurIPS*. 2018.

[Ste07] Harald Steck. Hinge rank loss and the area under the roc curve. In Joost N. Kok, Jacek Koronacki, Raomon Lopez de Mantaras, Stan Matwin, Dunja Mladenič, and Andrzej Skowron, editors, *Machine Learning : ECML 2007*, 2007.

[Tom76] Ivan Tomek. Two modifications of cnn. In *IEEE Transactions on Systems Man and Communications*, 1976.

[Wil72] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 3, 1972.

[WS09] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10, 2009.