Applied Soft Computing Journal 76 (2019) 380–389

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

An alternative SMOTE oversampling strategy for high-dimensional datasets

Sebastián Maldonado [a,*], Julio López [b], Carla Vairetti [a]

[a] Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Monseñor Álvaro del Portillo 12455, Las Condes, Santiago, Chile
[b] Facultad de Ingeniería y Ciencias, Universidad Diego Portales, Av. Ejército 441, Santiago, Chile

HIGHLIGHTS

- Novel SMOTE oversampling approach for imbalanced data sets.
- A novel distance metric is proposed for dealing with high-dimensionality.
- This metric improves the definition of neighborhoods in SMOTE.
- Best classification performance is achieved in experiments on benchmark datasets.

ARTICLE INFO

Article history:
Received 14 October 2017
Received in revised form 14 November 2018
Accepted 14 December 2018
Available online 27 December 2018

Keywords:
Data resampling
SMOTE
High-dimensional datasets
Feature selection
Imbalanced data classification

ABSTRACT

In this work, the Synthetic Minority Over-sampling Technique (SMOTE) approach is adapted for high-dimensional binary settings. A novel distance metric is proposed for the computation of the neighborhood for each minority sample, which takes into account only a subset of the available attributes that are relevant for the task. Three variants for the distance metric are explored: Euclidean, Manhattan, and Chebyshev distances, and four different ranking strategies: Fisher Score, Mutual Information, Eigenvector Centrality, and Correlation Score. Our proposal was compared with various oversampling techniques on low- and high-dimensional datasets with the presence of class-imbalance, including a case study on Natural Language Processing (NLP). The proposed oversampling strategy showed superior results on average when compared with SMOTE and other variants, demonstrating the importance of selecting the right attributes when defining the neighborhood in SMOTE-based oversampling methods.

1. Introduction

Binary classification can be challenging when confronting class-imbalanced datasets. When the class distribution is too skewed, machine learning methods tend to generate classifiers that favor the majority class, assigning the most frequent label to most test samples. Although this fact usually leads to high overall accuracy, it may cause poor decision-making since the minority class is likely to have higher misclassification costs compared to the majority class in most applications [1–3]. Some common tasks that face this issue are churn prediction [4], text categorization [5], and bioinformatics [6–8], among others.

Data resampling has proved to be very effective for dealing with class-imbalance [2,9]. In particular, SMOTE oversampling is arguably the most frequently used technique when few samples are available [1,9,10]. In such cases, undersampling could lead to an important loss of information. The success of SMOTE oversampling and its variants can be explained by their simplicity, computational efficiency, and superior performance [2,10,11].

In simple terms, SMOTE generates new samples from the minority class artificially by interpolating pre-existing ones. These pre-existing instances are chosen by defining a neighborhood, identifying the k nearest neighbors for each minority sample [12]. In high-dimensional settings, however, the various samples are almost uniformly distant from each other, negatively affecting the proper definition of neighborhood [7].

In this work, we study the effect of high-dimensionality on SMOTE oversampling by formalizing a novel distance metric based on only the relevant attributes for the problem. The identification of relevant variables is done via feature ranking methods, such as the Fisher Score (FS), Mutual Information (MI), Eigenvector Centrality (EC), and the Correlation Score (CFS). Furthermore, we explore the influence of distance metrics that are more suitable for high-dimensional problems than the Euclidean norm, such as the Manhattan and Chebyshev distances.

This paper has the following structure: Section 2 introduces the issue of class-imbalance, presenting SMOTE and other variations that are relevant for this study. Strategies for dealing with

[*] Corresponding author.
E-mail addresses: smaldonado@uandes.cl (S. Maldonado), julio.lopez@udp.cl (J. López), cvairetti@uandes.cl (C. Vairetti).

https://doi.org/10.1016/j.asoc.2018.12.024
1568-4946/© 2018 Elsevier B.V. All rights reserved.

high-dimensionality under class-imbalance conditions are also discussed in that section. The proposed oversampling approach is presented in Section 3. Experimental results using high-dimensional datasets are discussed in Section 4. Finally, this study is summarized in Section 5, where its main conclusions are provided.

2. Literature review

In this section, the relevant models for this study are presented. First, SMOTE oversampling and some well-known extensions are discussed. Subsequently, solutions for handling high-dimensionality and class-imbalance jointly are detailed, presenting the feature ranking methods that are used in the proposed SMOTE variation.

2.1. Imbalanced data classification and SMOTE

There are several approaches for dealing with the class-imbalance problem. The issue can be tackled independently from the classifier by balancing the training set artificially before model construction. This strategy is known as *data resampling*. Alternatively, classifiers can be modified to handle class-imbalance by biasing the model toward a better prediction of the minority class (*cost-sensitive learning*) or by training them with only one of the classes (*one-class learning*) [2,10,11]. These strategies are not reviewed in this work because they fall outside the scope of our proposal.

Data resampling can be done by either downsizing the majority class through discarding instances, an approach known as *undersampling*, or by adding new samples to the minority class, which is known as *oversampling*. The first strategy is particularly useful in large datasets in which the loss of information caused by discarding samples is marginal [2,10]. In this work, however, we focus on small-sized, high-dimensional datasets, such as microarray data, and therefore only oversampling methods are discussed.

Oversampling can be performed by simply replicating the existing elements of the minority class on the training set. This strategy, however, is known to be prone to overfitting [10,13]. To avoid this risk, the new samples can be created artificially by respecting the distribution of the minority class. One such approach is the Synthetic Minority Over-sampling Technique (SMOTE) [12], which has two main steps: First, a neighborhood is defined for each element of the minority class, identifying the k nearest neighbors. Usually, k is set to 5, and the distance metric used is the Euclidean norm. Next, $N < k$ elements of the neighborhood are randomly selected and used to construct new samples via interpolation [12].[1] Given a sample $\mathbf{x}_i$ from the minority class, and N randomly chosen samples from its neighborhood $\mathbf{x}_i^p$, with $p = 1, \ldots, N$, a new synthetic sample $\mathbf{x}_i^{*p}$ is obtained with the following expression:

$$\mathbf{x}_i^{*p} := \mathbf{x}_i + u\left(\mathbf{x}_i^p - \mathbf{x}_i\right), \tag{1}$$

where u is a randomly generated number between 0 and 1. This method has the advantages of being fast to compute and successful at providing balanced and accurate classification performance. Since SMOTE is independent of the classifier, it can be used with any classification technique [10,13].

One issue that SMOTE suffers from is over-generalization. Since the majority class is ignored by the method, synthetic points can be created over the majority class, increasing class overlap [2,3,13]. For example, there could be a large distance between $\mathbf{x}_i^p$ and $\mathbf{x}_i$ if the distribution of the minority samples is very sparse, and $\mathbf{x}_i^{*p}$ could be created in a zone where the majority class, rather than the minority class, is dense [13,14].

To overcome the issue of over-generalization, some extensions have been proposed. Borderline-SMOTE (SMOTE-B, [15]), for example, aims at creating examples from the minority class that are close to the borderline between the two classes. Later, Safe-level SMOTE (SMOTE-SL, [16]) was proposed, which defines a 'safe level' for each minority sample, creating new instances closer to this safe level by introducing weights in the computation of the new samples. This technique, however, is prone to overfitting since the synthetic examples are designed to be far from the classification function. Another method that assigns weights to the samples is the Adaptive Synthetic Sampling Approach (ADASYN, [17]). The idea is to increase the chance of being oversampled for examples that are hard to learn, following the reasoning behind well-known adaptive ensemble methods such as Adaboost [18].

SMOTE oversampling is still a fruitful field of research, and recent developments include MWMOTE [19], a two-step weighted approach that extends Borderline-SMOTE and ADASYN by using the information of the majority instances that lie close to the borderline; A-SUWO [13], a clustering-based algorithm designed to identify groups of minority samples that are not overlapped with clusters from the majority class; or CURE-SMOTE [20], that also uses clustering on the minority class but with the goal of denoising and removing outliers before oversampling.

All the SMOTE variants discussed in this section aim at improving predictive performance by enhancing separability. Some of these strategies remove noise in the minority class [20], while others take the majority class into account to reduce the issue of over-generalization [13,19]. The inclusion of the majority class in the oversampling process clearly provides additional information that helps create better synthetic samples, but it also requires describing the majority class, which can be very time-consuming, especially in large datasets.

The current state of the art tends to ignore the fact that SMOTE, to the best of our knowledge, defines the neighborhood using all the variables and weighting them equally. This assumption has been questioned in k-NN classification since it usually leads to poor prediction performance in high-dimensional settings with high levels of noise and redundancy [21]. Our proposal presents a simple and efficient strategy for defining this neighborhood based only on an adequate subset of variables, without significantly increasing the complexity of the SMOTE approach.

2.2. Dealing with high-dimensionality under class-imbalance

The class-imbalance problem is often accompanied by the issue of high-dimensionality. Therefore, the identification of the relevant variables becomes a key task for reducing class-overlap [10]. In this context, both resampling and cost-sensitive strategies have been used together with feature selection methods in high-dimensional, class-imbalanced settings [5,22–24]. Alternatively, the effect of high-dimensionality can be mitigated via feature extraction and manifold learning. For example, Principal Component Analysis (PCA) was used in Martin-Felez and Mollineda [25]. Feature extraction, however, is beyond the scope of this study.

The taxonomy for feature selection methods follows the same logic as the strategies for dealing with class-imbalance. Feature selection can be performed independently of the classification approach, filtering out irrelevant information before applying it (*filter methods*). Alternatively, it can be performed together with the process of classifier construction (*wrapper/embedded methods*).

The following filter methods for finding a subset of relevant variables as an input for our distance metric are used in our proposal: Fisher Score, Mutual Information, Eigenvector Centrality, and Correlation Score. The *Fisher Score* [26] computes the absolute

[1] The authors actually suggested multiples of 100 as values for N, but then they divided N by hundred.

difference between the means of the two classes, normalized with a joint standard deviation, as follows:

$$FS(j) = \frac{|\mu_{j1} - \mu_{j2}|}{\sigma_{j1}^2 + \sigma_{j2}^2}, \tag{2}$$

where μ_{jl} is the mean value for the jth attribute and class l, $l = 1, 2$, while σ_{jl} is its respective standard deviation.

Another suitable measure for assessing relevance is *Mutual Information*, which computes the amount of information about one attribute that can be gained by observing another one, as follows:

$$MI(j) = \sum_{y \in \mathbf{y}} \sum_{x \in \mathbf{x}_j} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right), \tag{3}$$

where x and y are the various levels of attribute $\mathbf{x}_j$ and the target vector $\mathbf{y}$, respectively; while $p(x)$ and $p(y)$ are their marginal probability distributions, with $p(x, y)$ being their joint distribution. As can be noted, this approach assumes that the covariates are nominal variables, unlike Fisher Score. Mutual Information, however, can be used with numerical variables after binning them [27].

Both the Fisher Score and Mutual Information approaches assess the dependency between the covariates and the label vector. However, there are other filter approaches whose goal is to reduce the redundancy between variables. The *Correlation Score* [28], for example, computes the Pearson correlation $\rho_{j,j'}$ for each pair of attributes j and j', and subsequently computes the lowest absolute correlation, as follows:

$$CFS(j) = \min_{j'} |\rho_{j,j'}|. \tag{4}$$

Additionally, metrics can be combined to assess relevancy and redundancy together. *Eigenvector Centrality* [29] combines the Fisher Score, the Mutual Information, and the covariates' standard deviation to construct an adjacency matrix A, and feature importance is assessed by computing the eigenvector related to the largest eigenvalue of A. The edges of A can be seen as the influence of two attributes that are used together for classification based on the metrics mentioned previously.

There are several filter strategies that can be used directly on class-imbalance problems. Feature Assessment by Sliding Thresholds (FAST) [30], for example, filters out irrelevant variables by computing the area under the curve (AUC) of each attribute, removing those with values close to 0.5. Density Based Feature Selection (DBFS) [31] follows a similar idea, although it uses Information Gain as the contribution measure instead of AUC. Alternatively, filter methods have been developed to assess redundancy among the variables, and used for dealing with the class-imbalance issue.

There are also some studies on model-based feature selection under class-imbalance conditions [32]. Villar et al. [33], for example, proposed a genetic algorithm that constructs a fuzzy rule-based classification system and selects the relevant attributes simultaneously. This is done via backward elimination using AUC as the contribution measure. Maldonado et al. [34] also proposed a backward elimination approach which is designed for class-imbalanced classification with Support Vector Machines [35]. For this method, balanced accuracy was used to assess the feature importance: The attribute whose removal leads to the largest improvement in this measure is eliminated in an iterative fashion.

Finally, the SMOTE oversampling approach has been used on high-dimensional settings. In Deepa and Punithavalli [36], the authors propose E-SMOTE, using genetic algorithms for feature selection and SMOTE oversampling for data resampling on the resulting subset of relevant variables. Qazi and Raza [37] explore the influence of two feature selection approaches (a genetic algorithm based on redundancy and information gain), and two resampling techniques (random undersampling and SMOTE) on a network intrusion detection dataset. Finally, Blagus and Lusa [7] acknowledge the issue of using SMOTE on high-dimensional settings, not being able to mitigate the bias toward the majority class. The authors compared two techniques for correcting the class-imbalance issue (adjusted classification threshold and SMOTE) using various classifiers and on low- and high-dimensional settings. They concluded that SMOTE achieves either similar or worse performance compared to no class-imbalance correction on high-dimensional datasets with few samples. None of the previously mentioned studies propose an algorithmic solution for SMOTE oversampling when dealing with high-dimensional datasets, which is the main contribution of this study.

3. Proposed SMOTE for high-dimensional datasets

The SMOTE algorithm computes the distance between training points from the minority class to define a neighborhood, from which examples are selected for the creation of new synthetic points. These distances are usually computed by using the Euclidean distance. Two important issues can be identified for this step when facing high-dimensional datasets: First, the Euclidean distance is not a suitable norm on high-dimensional settings because the concept of proximity is ill-defined, with all points being approximately equidistant from each other [38]. Furthermore, the Euclidean distance assumes that all attributes are equally important for the definition of a neighborhood in the SMOTE algorithm, but high-dimensional datasets usually have a high percentage of redundant and/or irrelevant variables that introduce noise in the algorithm.

The advantage of our approach over performing feature selection and SMOTE oversampling is that our strategy can be used with wrapper/embedded feature selection methods, which could improve predictive performance. We are postponing the decision of performing feature selection after the resampling process because the right number of attributes for the SMOTE oversampling may not be the same number as that for the classifier. Notice that SMOTE is based on k-NN, which is not able to weight attributes differently. Classifiers that can perform feature selection and classification simultaneously may be more effective than filter strategies with this step. Strictly speaking, our method does not perform feature selection since all classifiers used in this study consider all the variables, and the output of the proposal is an oversampled minority class that also includes all the variables.

In this work, we approach these two issues by redefining the concept of distance within the SMOTE algorithm. Two directions are explored: First, a new metric based on the Minkowski distance [39] is presented, which allows the use of alternatives to the Euclidean distance that are more suitable for high-dimensional datasets, such as the Chebyshev and the Manhattan distance. Secondly, this new metric redefines the original Minkowski distance by considering only a subset of the available attributes in the computation of the distance. This subset is proposed to be constructed via feature ranking strategies, and four filter methods described in the previous section are explored: Fisher Score, Mutual Information, Correlation Score, and Eigenvector Centrality.

Let us consider training samples $\mathbf{x}_i \in \Re^n$, $i = 1, \ldots, m$, their respective labels $y_i \in \{-1, +1\}$, and a set $\mathcal{T}$ that contains all the samples from the minority class. Following the notation presented in Song et al. [40], let $\mathcal{S}$ be the full set of attributes, and $\mathcal{S}^\dagger \subseteq \mathcal{S}$ be a subset of potentially relevant variables of cardinality r. The inputs for the proposed algorithm follow:

- *SMOTE Parameters*: Similar to the SMOTE algorithm, our approach computes the k nearest neighbors for each minority sample, and selects $N < k$ of these neighbors to construct the synthetic samples. The values for k and N need to be defined a priori. In this work, we use $k = 5$ and $N \in \{2, 4\}$, as suggested in Chawla et al. [12]. The N parameter is interpreted as the amount of oversampling; $N = 2$ and $N = 4$ correspond to 200% and 400% oversampling, respectively. Additionally, the interpolation step for the construction of synthetic samples considers a random number u that should be between 0 and 1.

- *Redefined Distance Metric*: Our proposal requires the selection of a subset of r variables of $\mathcal{S}$, using a feature ranking strategy *FR*. We also extend the SMOTE algorithm by using the Minkowski distance instead of the traditional Euclidean distance, exploring $q \in \{1, 2, \infty\}$ (Manhattan, Euclidean, and Chebyshev distances, respectively). The proposed distance between two samples from the minority class i and i' follows:

$$d_{\mathcal{S}^\dagger,q}(\mathbf{x}_i, \mathbf{x}_{i'}) = \|\mathbf{x}_i - \mathbf{x}_{i'}\|_{q,\mathcal{S}^\dagger} = \left(\sum_{j\in\mathcal{S}^\dagger} |x_{i,j} - x_{i',j}|^q\right)^{1/q}, \tag{5}$$

  for $q \geq 1$. The proof that the proposed distance metric satisfies the various properties required for being a distance measure is presented in Appendix A. The values for r and q, and the feature ranking strategy *FR* that leads to $\mathcal{S}^\dagger$ need to be defined a priori.
- *Feature Ranking Methods*: The feature ranking methods used to determine the subset of relevant variables $\mathcal{S}^\dagger$ were formalized in Section 2.2: Fisher Score (cf. Eq. (2)), Mutual Information (cf. Eq. (3)), Eigenvector Centrality, and Correlation Score (cf. Eq. (4)).

We refer to our proposal as the SMOTE-Subset of Features (SMOTE-SF) method. The proposed algorithm follows:

Algorithm 1 SMOTE-SF for high-dimensional datasets.

Input: The full set of attributes $\mathcal{S}$; Minority class sample set $\mathcal{T}$; Amount of oversampling N; Number of nearest neighbors k; Number of selected attributes r, value of q for the distance.
Output: A relevant set of attributes $\mathcal{S}^\dagger$; Oversampled minority class sample set $\mathcal{T}^\dagger$

```
1.  $\mathcal{T}^\dagger \leftarrow \mathcal{T}$.
2.  for $j \in \mathcal{S}$
3.    $FR(j) \leftarrow$ Score each attribute according to its relevance using $FR$.
4.  end for
5.  $\mathcal{S}^\dagger \leftarrow$ Take the $r$ largest values of $FR(j)$.
6.  for $i \in \mathcal{T}$
7.    for $i' \in \mathcal{T},\ i \neq i'$
8.      $d_{\mathcal{S}^\dagger,q}(\mathbf{x}_i, \mathbf{x}_{i'}) \leftarrow \left(\sum_{j\in\mathcal{S}^\dagger} |x_{i,j} - x_{i',j}|^q\right)^{1/q}$.
9.    end for
10.   $\mathcal{TK} \leftarrow \arg\min_{\mathcal{TK}} \sum_{i'\in\mathcal{TK}} d_{\mathcal{S}^\dagger,q}(\mathbf{x}_i, \mathbf{x}_{i'}),\ \mathcal{TK} \subseteq \mathcal{T}\backslash\{i\},\ |\mathcal{TK}| = k$.
11.   for $n \leftarrow 1$ to $N$
12.     $\mathbf{x}_i^n \leftarrow$ Select a random sample from $\mathcal{TK}$.
13.     $\mathbf{x}_i^{*n} \leftarrow \mathbf{x}_i + u \cdot (\mathbf{x}_i^n - \mathbf{x}_i)$.
14.     $\mathcal{T}^\dagger \leftarrow (\mathcal{T}^\dagger, \mathbf{x}_i^{*n})$.
15.     $\mathcal{TK} \leftarrow \mathcal{TK} \setminus \{\mathbf{x}_i^n\}$.
16.   end for
17. end for
```

The first five steps represent the initialization of the algorithm: The new set of minority samples $\mathcal{T}^\dagger$ is first defined as $\mathcal{T}$, the original minority samples (Step 1); and the subset of selected attributes $\mathcal{S}^\dagger$ is defined as the r variables with the highest ranking according to *FR*.

Table 1
Imbalance Ratio (IR), Number of attributes, number of samples, and percentage of samples in each class for all ten datasets.

Dataset	IR	#attributes	#samples	%class(min.,maj.)
Low-dimensional datasets				
Ecoli	8.6	7	336	(10.4,89.6)
Abalone	9.7	8	4177	(9.4,90.6)
CarEval	11.9	6	1728	(7.8,92.2)
Solar	19.4	10	1389	(4.9,95.1)
Yeast	28.1	8	1484	(3.4,96.6)
High-dimensional datasets				
Burczynski	3.88	22,283	127	(20.4,79.6)
Lung	4.85	12,533	181	(17.1,82.9)
Glioma	6.14	4434	50	(14.0,86.0)
SRBCT	6.55	2308	83	(13.3,86.7)
Lung2	9.15	3312	203	(9.8,90.2)
Bullinger	11.25	17,404	98	(8.2,91.8)
CAR	14.8	9182	174	(6.3,93.7)

Steps 6 to 17 correspond to the main loop for the development of synthetic examples: For each object of the minority class i, the distances between i and all the other elements from this class are computed using Eq. (5) (steps 7 to 9). Then, the k objects that are closest to i are identified (set $\mathcal{TK}$, step 10). Finally, N objects from $\mathcal{TK}$ are randomly selected and used for creating new samples for the minority class, including them in $\mathcal{T}^\dagger$ (steps 11 to 16). For this step, our method is similar to SMOTE oversampling.

4. Experimental evaluation

The proposed SMOTE-SF algorithm was applied in twelve class-imbalanced datasets to assess its performance compared to well-known oversampling strategies described in Section 2.1. Additionally, our proposal was applied to a Natural Language Processing (NLP) project that consists of three large, sparse datasets based on TripAdvisor comments. This data was collected by us for analyzing the factors that influence positive and negative reviews for Chilean Restaurants. The results are presented at the end of this section.

Our proposal is compared with the traditional SMOTE [12], Borderline SMOTE (SMOTE-B) [15], and Safe Level SMOTE (SMOTE-SL) [16] using various classification techniques, namely k-nearest neighbors (k-NN), logistic regression (LR), Naïve Bayes (NB), and linear Support Vector Machines (SVM). A detailed description of these methods can be found in [21]. These classifiers were chosen since they have been used widely in previous studies on data resampling for dealing with the class-imbalance problem [21]. In contrast to machine learning methods, such as neural networks or random forest, the selected techniques have few free parameters to calibrate, simplifying the model selection task [12,37].

4.1. Experimental setting and datasets

Of the twelve benchmark datasets studied in this work (excluding the NLP datasets), five are low-dimensional applications from the UCI data repository [41], while the rest are high-dimensional microarray datasets. Additionally, four datasets are binary classification problems, while the remaining eight are adapted multiclass classification tasks, in which the majority class was constructed by grouping all the labels except the minority class, as described in [42]. The relevant information is presented in Table 1 for each dataset:

It can be observed in Table 1 that the datasets studied are very diverse in terms of imbalance ratio (from 3.88 to 28.1), and number of attributes (from 6 to 22283). For the multiclass datasets, the following groups of labels were created. The use of multiclass datasets is justified by the fact that there are few binary classification datasets available that are class-imbalanced and high-dimensional, and that identifying a particular type of cancer over others can be a compelling challenge for the use of microarray data [32]. Notice that the performance of other label combinations was not assessed for this study.

- **Abalone**: This dataset studies 29 types of abalone, and type 7 was used as the minority class while all other abalone types were used as the majority class.
- **CarEval**: The Car Evaluation dataset considers four levels of acceptability of various used cars, of which class 3 (good) and class 4 (very good) were studied together as minority class.
- **Solar**: The Solar Flare dataset studies different types of solar flares on a given day. For this study we focus on M-class flares (moderate flares), of which two classes were constructed by studying the occurrence of zero M-class flares in 24 h versus one or more in the same time period.
- **Yeast**: This dataset studies the protein localization problem, and class ME2 (membrane protein with cleaved signal) was studied as the minority class.
- **Glioma**: For this microarray dataset, the label 'cancer oligodendrogliomas' was used as the minority class.
- **SRBCT**: For this microarray dataset, the label 'Burkitt lymphoma' was used as the minority class.
- **Lung2**: For this microarray dataset, the label 'small-cell lung carcinomas' was used as the minority class.
- **CAR**: For this microarray dataset, the label 'kidney cancer' was used as the minority class.
- **Burczynski**: For the Burczynski dataset, the label 'ulcerative colitis' was used as the minority class.

The low-dimensional datasets were used for benchmarking cost-sensitive SVM formulations in [43], while the microarray datasets were studied in [34,44] to compare feature selection algorithms.

The methodological procedure follows: 10-fold and leave-one-out cross validation were conducted for the low and high-dimensional datasets, respectively, using AUC (Area Under the Curve) as the performance metric. This metric has been used widely in class-imbalance classification [24,30,32]. Alternatively, suitable metrics for class-imbalance classification are the F measure, g-mean, or balanced accuracy [2,34].

While logistic regression and Naïve Bayes do not require parameter setting in their formulations, the values for k and C have to be defined for k nearest neighbors and SVMs, respectively. We used $k = 5$ and $C = 1$ for these methods, since they are suggested in the literature as good default values for these methods [21,45]. For all oversampling methods, the number of neighbors was set to $k = 5$, as suggested in the original paper by Chawla et al. [12]. We selected $N = 2$ and $N = 4$ objects from these five neighbors (200% and 400% oversampling, respectively). For the proposed method, the following values for r (the subset of selected attributes) were studied: $r \in \{1, 3, 5, n\}$ and $r \in \{20, 50, 100, 250, 500, 1000\}$ for the low- and high-dimensional datasets, respectively.

The datasets included in this study have no missing values. Our method shares the same characteristics with SMOTE on this issue. Therefore, data imputation is recommended for computing the distances between observations properly in the presence of missing examples.

4.2. Comparison among methods and running times

Next, a comparison between the best configuration of SMOTE-SF and the alternative oversampling techniques (SMOTE, SMOTE-B, and SMOTE-SL) is presented in Tables 2 and 3 for the low- and high-dimensional datasets, respectively. For each dataset, the best performance in terms of AUC×100 is highlighted in bold type. For our proposal, the best configuration is reported. The AUC values reported in Tables 2 and 3 are averages of ten and n runs, respectively (ten-fold and LOO cross-validation, respectively).

Table 2
Predictive performance summary (AUC×100) for the various oversampling methods. Low-dimensional datasets.

Method	NO-RS	RUS	SMOTE	SMOTE-B	SMOTE-SL	SMOTE-SF
Ecoli						
k-NN	79.25	89.34	87.92	85.68	87.08	88.42
LOGIT	70.42	85.26	87.75	87.42	88.42	90.75
NB	88.01	89.17	89.01	88.09	88.42	**91.26**
SVM	50	87.34	89.42	87.17	89.42	90.92
Abalone						
k-NN	56.94	77.38	70.52	71.10	70.76	72.32
LOGIT	50.29	77.91	77.27	78.29	**78.86**	77.48
NB	76.83	77.15	77.11	77.30	76.93	77.92
SVM	50	75.38	76.31	77.98	78.65	77.08
CarEval						
k-NN	50	88.68	94.53	88.91	78.32	49.06
LOGIT	95.63	97.70	98.37	98.72	96.18	98.78
NB	66.12	80.43	76.67	81.58	73.94	79.68
SVM	90.69	95.33	99.18	99.18	98.12	**99.18**
Solar						
k-NN	51.96	65.76	60.69	63.97	62.39	68.59
LOGIT	53.46	70.06	64.57	69.02	66.51	66.39
NB	68.85	70.40	72.21	72.65	70.51	**73.62**
SVM	50	72.52	61.22	69.27	64.58	66.42
Yeast						
k-NN	55.48	82.00	74.10	79.05	73.62	77.75
LOGIT	57.72	78.20	77.05	81.24	81.31	78.08
NB	67.48	79.84	79.97	79.45	76.88	80.94
SVM	50	**82.82**	71.56	81.25	80.90	72.49

It can be observed in Tables 2 and 3 that the largest AUC is usually achieved with the proposed SMOTE-SF, although no method is able to outperform all the others. The proposed method achieves the best AUC in eight of the twelve datasets (Ecoli, CarEval, Solar, Lung, SRBCT, Lung2, Bullinger, and CAR). These positive results can be explained by the fact that the metric used to define the neighborhood is more suitable than the Euclidean norm, especially in high-dimensional settings, confirming the virtues of our proposal. SMOTE-SL performed best on the Abalone and Yeast datasets; and SMOTE-B achieved the best result for the Glioma dataset. For these datasets, we infer that the use of the majority class for preventing over-generalization explains this result.

For the CarEval dataset (see Table 2), our proposal performs significantly worse than the rest when k-NN is used. This classification technique, however, is not able to achieve good enough performance with any of the oversampling techniques, showing that it is not a suitable classifier for this particular dataset. But our proposal achieves the best AUC for this dataset when SVM is used.

It can be noticed that an AUC of 1 can be achieved on some datasets. Some microarray datasets seem to be linearly separable when a regularized classifier such as SVM is used. Since linear SVM with $C = 1$ is used, there is a low risk of overfitting.

These differences were studied further with the Friedman and Holm tests, which have been used frequently for multiple comparisons between classification methods since their use was suggested by Demšar for this purpose in [46]. These approaches compute the average rank for each method on all datasets given their predictive performance, assessing the differences between methods statistically [46]. First, the Friedman test computes an F statistic under the null hypothesis that all ranks are equal. The Iman–Davenport correction is applied to the Friedman test, as suggested in [46].

Table 3
Predictive performance summary (AUC×100) for the various oversampling methods. High-dimensional datasets.

	NO-RS	RUS	SMOTE	SMOTE-B	SMOTE-SL	SMOTE-SF
Burczynski						
k-NN	74.83	66.57	66.34	73.76	67.82	67.82
LOGIT	49.87	55.64	55.31	57.73	49.05	72.73
NB	69.61	76.64	60.55	65.33	66.32	64.39
SVM	78.85	**81.85**	78.85	78.85	78.85	78.85
Lung						
k-NN	100	86.81	97.92	95.83	97.22	98.61
LOGIT	95.83	82.58	91.98	82.89	94.07	99.31
NB	95.45	**100**	86.36	77.27	86.36	86.36
SVM	**100**	**100**	**100**	**100**	**100**	**100**
Glioma						
k-NN	95.45	86.20	98.47	99.31	98.77	99.08
LOGIT	84.52	82.32	98.47	91.98	91.47	99.69
NB	77.27	94.84	50	86.36	72.73	50
SVM	90.91	94.53	90.91	**100**	90.91	90.91
SRBCT						
k-NN	92.86	82.56	93.02	93.02	94.19	**94.19**
LOGIT	57.48	57.64	57.48	56.31	69.27	79.90
NB	77.41	78.90	50	78.57	78.57	50
SVM	75.08	80.07	75.08	75.08	75.08	75.08
Lung2						
k-NN	**100**	99.18	99.73	99.73	99.73	**100**
LOGIT	90.59	98.36	96.41	99.45	95.59	99.73
NB	99.73	99.73	50	99.73	99.73	50
SVM	**100**	99.18	**100**	**100**	**100**	**100**
Bullinger						
k-NN	76.25	47.64	79.44	78.19	82.78	82.78
LOGIT	70.69	80.97	72.36	68.47	65.00	85.42
NB	81.25	81.25	81.25	81.25	81.25	81.25
SVM	93.75	86.67	93.75	93.75	93.75	**93.75**
CAR						
k-NN	82.26	90.32	99.67	96.44	99.67	**100**
LOGIT	76.43	83.71	76.26	81.10	80.38	91.49
NB	96.44	96.39	93.55	96.44	96.77	95.16
SVM	98.39	**100**	98.39	98.39	98.39	98.39

Table 4
Holm's post-hoc test for pairwise comparisons. Various oversampling methods.

Method	Ranking		AUC×100		p value	$\alpha/(j-1)$	Action
	Mean	Std.	Mean	Std.			
SMOTE-SF	2.67	1.49	82.54	17.5	–	–	not reject
SMOTE-B	3.10	1.90	83.80	12.1	0.25	0.0500	not reject
RUS	3.22	1.14	83.19	15.3	0.14	0.0250	not reject
SMOTE-SL	3.48	1.31	83.03	12.1	0.03	0.0167	not reject
SMOTE	3.94	1.44	80.77	12.6	0.0009	0.0125	reject
NO-RS	4.59	1.23	76.26	14.9	0.0001	0.0100	reject

Next, the Holm post-hoc test is used for pairwise comparisons, assessing the differences between the method with the highest rank and the remaining ones [46].

The result for the Friedman test with Iman–Davenport correction is $F = 8.42$, with a p value below 0.001, rejecting the null hypothesis of equal ranks. The results for the Holm's test are presented in Table 4 for the various oversampling methods. For each method, this table reports the mean rank, the mean AUC, the p value for the Holm's test, the significance threshold, and the conclusion for the pairwise test. A method is outperformed by the one with the best ranking if the p value is below the significance threshold (Action = reject), with a significance level of $\alpha = 5\%$, and $j = 1, \ldots, 5$ being the overall ranking for a given method.

Table 5
Running times, in seconds, for all datasets and SMOTE variants. The following methods are presented (from left to right): the proposed SMOTE-SF using Fisher Score, Mutual Information, Eigenvector Centrality, and the Correlation Score; standard SMOTE, Borderline-SMOTE, and Safe-level SMOTE.

Dataset	S.-SF (FS)	S.-SF (MI)	S.-SF (EC)	S.-SF (CFS)	S.	S.-B	S.-SL
Ecoli	<**0".01**	0".03	<**0".01**	<**0".01**	<**0".01**	0".14	0".16
Abalone	0".03	0".09	0".06	**0".02**	0".20	4".38	5".00
CarEval	**0".02**	**0".02**	0".05	0".06	0".19	1".64	1".25
Solar	**0".02**	0".05	0".09	0".06	0".06	1".84	0".73
Yeast	<**0".01**	0".03	0".02	<**0".01**	<**0".01**	0".55	0".69
Burczynski	**2".91**	6".67	105".77	30".03	2".98	3".97	4".39
Lung	**1".75**	4".70	21".77	9".63	2".25	4".39	4".45
Glioma	0".58	1".00	4".47	1".64	**0".03**	0".13	0".14
SRBCT	0".34	0".56	1".42	0".55	0".19	0".64	**0".16**
Lung2	0".42	1".30	3".16	1".41	**0".41**	1".28	1".50
Bullinger	2".02	4".45	36".52	16".23	**0".30**	6".84	6".94
CAR	1".13	3".06	12".42	5".23	**0".23**	2".36	2".30

It can be seen in Table 4 that our proposal has the best overall rank, with an average of 2.67. SMOTE-SF is able to outperform SMOTE statistically, although SMOTE-B, SMOTE-SL, and random undersampling are not significantly worse than the proposed approach. It can be concluded that this proposal is an excellent alternative for low- and high-dimensional datasets with the presence of irrelevant attributes, allowing a better definition of a neighborhood for each minority sample before the construction of synthetic objects.

Finally, Table 5 provides a comparison for each SMOTE variation in terms of running times. The analysis was performed on an HP Envy dv6 with 16 GB RAM, 750 GB SSD, a i7-2620M processor with 2.70 GHz, and using Microsoft Windows 8.1 Operating System (64-bits). The best performance in terms of running times is highlighted in bold type.

In Table 5, it can be seen that our approach using Fisher Score (second column) is almost as fast as SMOTE oversampling, and clearly faster than the SMOTE-B and SMOTE-SL on the datasets that have the largest dimensionality. The use of Eigenvector Centrality (fourth column), however, leads to the longest running times. Fisher Score and Mutual Information are therefore recommended, given the positive results in terms of performance and computational complexity.

4.3. Detailed SMOTE-SF performance

In this section, the proposed SMOTE-SF is further analyzed in terms of its performance when varying the different ranking methods (the Fisher Score, Mutual Information, the Correlation Score, and Eigenvector Centrality) and distance metrics (Chebyshev or $q = \infty$, Manhattan or $q = 1$, and Euclidean or $q = 2$). The Holm's test discussed in the previous section is applied to SMOTE-SF to determine whether or not the differences in terms of performance are significant.

The results for the Friedman tests with Iman–Davenport correction are $F = 2.58$ and $F = 1.17$ for the ranking methods and distance metrics, respectively, with p values of 0.06 and 0.56. Therefore, the hypothesis of equal ranks cannot be rejected. The results for the Holm's test are reported in Tables 6 and 7 for the various ranking methods and distance metrics, respectively. The detailed results for each dataset are presented in Appendix B: Tables B.1 and B.2 report the performance for the various ranking methods on low- and high-dimensional datasets, respectively, while Tables B.3 and B.4 present the predictive performance for the various distance metrics on low- and high-dimensional datasets, respectively.

It can be observed clearly in Tables 6 and 7 that, in both cases, the differences in terms of mean rank and performance are very small, and no strategy is able to outperform the others statistically. It can be concluded that neither the ranking method used to assess relevance nor the variation of the Minkowski distance

Table 6
Holm's post-hoc test for pairwise comparisons. Various ranking methods related to SMOTE-SF.

Method	Ranking		AUC×100		p value	$\alpha/(j-1)$	Action
	Mean	Std.	Mean	Std.			
MI	2.35	1.09	80.97	15.9	–	–	not reject
CFS	2.37	1.04	80.83	15.8	0.93	0.05	not reject
FS	2.60	1.02	80.67	15.8	0.19	0.025	not reject
EC	2.68	1.02	80.70	15.9	0.08	0.0167	not reject

Table 7
Holm's post-hoc test for pairwise comparisons. Various distance metric related to SMOTE-SF.

Method	Ranking		AUC×100		p value	$\alpha/(j-1)$	Action
	Mean	Std.	Mean	Std.			
$p=2$	1.98	0.60	81.06	15.8	–	–	not reject
$p=1$	1.99	0.58	81.01	15.7	0.91	0.05	not reject
$p=\infty$	2.03	0.58	81.05	15.7	0.75	0.025	not reject

have a strong influence on the final performance. Therefore, the good results obtained by SMOTE-SF can be explained largely by the construction of a subset of attributes when defining a neighborhood for the minority instances. Finally, regarding the influence of parameter r (the subset of selected attributes used with the novel metric), we observed that our method performed best with approximately 50% of the attributes for the low-dimensional datasets $r=5$, and 100 to 250 attributes for the microarray datasets.

4.4. Experiments on large NLP datasets

To demonstrate evaluating large, sparse datasets, we present a Natural Language Processing (NLP) project based on a TripAdvisor collection. TripAdvisor is a travel website that provides rich travel-related information in reviews detailing travelers' experiences with hotels, restaurants, and tourist spots. In particular, we explored perceptions of Chilean restaurants collected from the period between January 3, 2009 and December 31, 2014. It contains 37,801 comments made on 757 Chilean restaurants.

Each entry consists of structured data, such as the restaurant and user ID, and the user's evaluation of the restaurant based on a [1–5] point rating; and unstructured data: the title and the body of the comment. The dataset was cast into a binary classification problem, in which we used the rating as the target variable by defining a negative comment as $y=+1$ when $rating \leq 2$, and a positive/neutral comment $y=-1$ when $rating \geq 2$. This leads to an IR of 8.04, or, equivalently, to 11.06% of negative comments.

Data preprocessing plays a very important role in our study. The techniques used in the preprocessing stage were: Extraction, Stop Words Elimination (articles, prepositions, and pro-nouns, etc. that do not provide meaning for the documents), Word Singularization, and Verb Stemming and Lemmatization.

After the preprocessing stage, the apriori algorithm was used for n-gram construction [47]. Similar to the traditional apriori approach [48], this strategy identifies the terms that are frequent in the dataset. Next, sequences of words of length n, called n-grams, are generated by combining terms that are frequent using a forward approach, until no further items can be combined. This algorithm uses two input parameters: the maximum length of the n-grams (MaxNGramSize), and a lower bound on the number of occurrences of a word (MinFrequency).

In our study, we explored MinFrequency ∈ {200, 300, 500} and MaxNGramSize = 5, leading to three sparse datasets with a total of 37,801 comments. The relevant metadata is presented in Table 8.

It can be observed in Table 8 that the three datasets are high-dimensional, ranging from 2,939 to 19,081 columns (the n-grams).

Table 8
Metadata for the NLP datasets.

MinFreq.	1-gram	2-gram	3-gram	4-gram	5-gram	Total	Samples
200	656	5971	8678	3461	315	19,081	37,801
300	477	3236	3506	954	45	8218	37,801
500	316	1426	1034	163	–	2939	37,801

Table 9
Results for the NLP datasets.

MinFreq.	NO-RS	RUS	SMOTE	SMOTE-SF
200	80.06	79.88	79.91	**80.50**
300	77.76	79.76	80.50	**81.37**
500	77.49	81.35	82.27	**82.71**

By construction, a low MinFrequency parameter implies a large number of n-grams since there are more combinations of terms that are frequent when the minimum word appearance is 200 when compared to a cutoff of 500. We also observe a larger number of 2-grams and 3-grams when comparing with the remaining n-grams.

Next, a comparison between SMOTE-SF, SMOTE, no resampling, and random undersampling is presented in Table 9. For SMOTE-SF, we ran experiments for only the FS and MI methods since EC and CFS are computationally too expensive for large datasets. Similarly, SMOTE-B and SMOTE-SL were not used because they were intractable in terms of running times. For each NLP dataset, the best performance in terms of AUC×100 is highlighted in bold type. We used $r \in \{100, 250, 500, 1000, 5000, 10000\}$ for MinFrequency = 200, and $r \in \{20, 50, 100, 250, 500, 1000\}$ for MinFrequency = {200, 300, 500}.

It can be observed in Table 9, that the largest AUC is achieved with the proposed SMOTE-SF for the three datasets. We conclude that our proposal is suitable even for large datasets in terms of number of samples, in contrast to SMOTE variations that analyze the majority class, and achieving best predictive results when compared to SMOTE and random undersamping.

5. Conclusion

In this work, a novel extension for the SMOTE oversampling approach is presented for dealing with the class-imbalance problem in binary classification. SMOTE generates synthetic examples from the minority class by first identifying the k nearest neighbors within this class, and then interpolating the reference sample with a randomly selected object from its neighborhood. The reasoning behind the proposal is that the definition of this neighborhood should be done with only a subset of the available attributes to avoid the curse of dimensionality. The use of filter methods is proposed for this task. A redefined Minkowski distance is presented and formalized as a distance metric with the corresponding proof.

Experiments on benchmark datasets demonstrated the virtues of the proposed SMOTE-SF method, which achieved the best overall performance, outperforming SMOTE oversampling statistically. The most important gains are observed in high-dimensional settings, such as the microarray datasets used in this study. These positive results were confirmed on large NLP datasets, in which SMOTE-SF achieved the best performance over SMOTE, no resampling, and random undersampling. This result confirms that oversampling can be useful even in cases when the sample size of the minority class is relatively large.

Four ranking methods for feature selection (the Fisher Score, Mutual Information, the Correlation Score, and Eigenvector Centrality) and three variations of the Minkowski distance (Chebyshev, Manhattan, and Euclidean) were studied empirically, yielding the conclusion that no significant differences in terms of performance are observed for all these variations in performance. Therefore, the benefit in terms of performance can be associated with the use of a subset of relevant attributes in the construction of the neighborhood rather than with a particular ranking strategy, or the use of a distance measure different from the Euclidean distance.

Regarding computational complexity, SMOTE-SF is similar to SMOTE with two differences: it requires an additional feature ranking step prior to the definition of the neighborhood, and it uses fewer attributes for the computation of the distances. Fortunately, the feature ranking step can be performed in a very efficient way using measures such as the Fisher Score or Mutual Information. For the second difference, important savings can be achieved in terms of training times for large datasets when fewer computations are required to obtain the distances between two minority samples, compensating for the computational effort of the additional feature ranking step. Our experiments on time complexity demonstrate that our method is as efficient as SMOTE, and faster than SMOTE-B and SMOTE-SL. The SMOTE-B and SMOTE-SL models become intractable for large datasets, such as the NLP data used in this study.

This approach represents an initial effort toward finding a better definition of the neighborhood in SMOTE oversampling, which has been developed only for the original version of the approach. Therefore, it has the limitations of SMOTE when compared with recent variants: the majority class is ignored in the construction of synthetic samples, and thus there is a risk of over-generalization.

There are various opportunities for future research. The proposed strategy can be applied to other SMOTE variations that deal with the issue of over-generalization, such as SMOTE-B or SMOTE-SL. Additionally, other distance metrics can be explored, such as the Gower distance [49]. This measure is suitable for mixed data types, such as numerical and categorical variables. Another possible research opportunity is linking the feature ranking step with the classification method, using the latter to assess the contribution of each attribute in the classification task. Methods such as logistic regression or decision trees are able to derive a feature ranking automatically, and embedded feature selection methods have been developed for methods such as SVM which are not able to assess feature relevance naturally. One example is the Recursive Feature Elimination SVM (RFE-SVM) [50]. Finally, the proposed framework can be useful in business Analytics and other domains in which the class-imbalance problem is fairly common. Applications in business Analytics with such a condition are credit scoring, fraud detection, and churn prediction, among others [4,51,52].

Acknowledgments

This research was partially funded by FONDECYT, Chile projects 1160738 and 1160894, and by the Complex Engineering Systems Institute, Chile (CONICYT - PIA - FB0816). The authors are grateful to the anonymous reviewers who contributed to improving the quality of the original paper.

Appendix A. Proof for the proposed distance metric

In this section, we present the proof that the redefined Minkowski distance proposed in Eq. (5) corresponds to a metric. First, let us recall the proposed norm for two samples $\mathbf{x}_i$ and $\mathbf{x}_{i'}$, with $i \neq i'$, $q \geq 1$, and set $\mathcal{S}^\dagger$ fixed:

$$d_{\mathcal{S}^\dagger,q}(\mathbf{x}_i,\mathbf{x}_{i'}) = \left(\sum_{j\in\mathcal{S}^\dagger} |x_{i,j} - x_{i',j}|^q\right)^{1/q}.$$

The proof for the redefined Minkowski distance follows:

Table B.1
Predictive performance (AUC × 100) for the various feature ranking methods related to SMOTE-SF. Low-dimensional datasets.

	200% Oversampling				400% Oversampling			
	k-NN	LR	NB	SVM	*k*-NN	LR	NB	SVM
Ecoli								
FS	87.1	87.9	90.1	89.6	87.8	90.8	89.9	89.4
MI	85.8	88.1	89.8	89.3	87.9	90.6	90.7	90.9
EC	86.6	87.8	89.0	89.3	86.3	90.6	90.3	89.4
CFS	86.9	88.1	**90.9**	89.4	88.4	90.8	**91.3**	90.1
Abalone								
FS	67.6	68.4	76.9	50.0	70.8	77.3	77.1	77.0
MI	69.4	68.1	76.9	50.0	72.3	77.4	77.1	77.1
EC	69.0	68.0	76.9	50.0	71.0	77.5	77.1	76.9
CFS	69.4	68.0	**77.9**	50.0	71.9	77.1	**77.9**	76.7
CarEval								
FS	49.0	98.5	75.2	98.9	48.4	98.8	77.0	98.8
MI	49.1	98.4	74.6	**99.2**	48.4	98.8	77.0	98.9
EC	48.6	98.5	77.1	99.1	47.4	98.5	79.7	**99.0**
CFS	48.4	98.5	75.3	99.1	47.6	98.7	76.2	98.9
Solar								
FS	65.1	63.6	70.6	61.3	66.0	63.6	71.4	64.2
MI	63.6	63.7	71.5	61.3	64.6	66.1	72.4	66.4
EC	62.6	63.9	71.3	61.3	68.6	64.7	72.9	66.1
CFS	62.7	65.3	**73.0**	61.2	65.6	66.4	**73.6**	64.3
Yeast								
FS	73.8	70.0	79.9	50.0	76.8	77.1	79.9	71.6
MI	**76.7**	70.0	78.0	50.0	76.8	78.1	80.7	71.4
EC	73.7	70.1	78.9	50.0	77.0	77.1	**80.9**	71.5
CFS	74.6	70.2	**80.1**	50.0	77.8	77.9	**80.9**	72.5

- In the case that $\mathbf{x}_i = \mathbf{x}_{i'}$, it is clear that $d_{\mathcal{S}^\dagger,q}(\mathbf{x}_i,\mathbf{x}_i) = 0$ since
$$\left(\sum_{j\in\mathcal{S}^\dagger} |x_{i,j} - x_{i,j}|^q\right)^{1/q} = 0.$$
- Suppose that $\mathbf{x}_i \neq \mathbf{x}_{i'}$. Then, there is a dimension $j \in \mathcal{S}^\dagger$ that $x_{i,j} \neq x_{i',j}$. Thus, $|x_{i,j} - x_{i',j}| > 0$, and therefore $d_{\mathcal{S}^\dagger,q}(\mathbf{x}_i,\mathbf{x}_{i'}) > 0$.
- It is easy to see that the redefined Minkowski distance is symmetrical since
$$\left(\sum_{j\in\mathcal{S}^\dagger} |x_{i,j} - x_{i',j}|^q\right)^{1/q} = \left(\sum_{j\in\mathcal{S}^\dagger} |x_{i',j} - x_{i,j}|^q\right)^{1/q}.$$
- Finally, it holds that $d_{\mathcal{S}^\dagger,q}(\mathbf{x}_i,\mathbf{x}_{i'}) \leq d_{\mathcal{S}^\dagger,q}(\mathbf{x}_i,\mathbf{x}_{i''}) + d_{\mathcal{S}^\dagger,q}(\mathbf{x}_{i''},\mathbf{x}_{i'})$ for any sample $\mathbf{x}_i, \mathbf{x}_{i'}, \mathbf{x}_{i''}$, by applying Minkowski's Inequality with $\mathbf{w} = \mathbf{x}_i - \mathbf{x}_{i''}$ and $\mathbf{z} = \mathbf{x}_{i''} - \mathbf{x}_{i'}$.

The detailed results for each benchmark dataset are presented in this appendix. Tables B.1 and B.2 report the performance for the various ranking methods on low- and high-dimensional datasets, respectively, while Tables B.3 and B.4 present the predictive performance for the various distance metrics on low- and high-dimensional datasets, respectively. For each dataset and resampling technique, the best performance in terms of AUC × 100 is highlighted in bold type.

Lemma 1 (*Minkowski's Inequality*). *For any* $\mathbf{w}, \mathbf{z} \in \Re^n$, *and* $q \geq 1$, *one has that*

$$\|\mathbf{w}+\mathbf{z}\|_{q,\mathcal{S}^\dagger} \leq \|\mathbf{w}\|_{q,\mathcal{S}^\dagger} + \|\mathbf{z}\|_{q,\mathcal{S}^\dagger}.$$

Proof.

$$\|\mathbf{w}+\mathbf{z}\|^q_{q,\mathcal{S}^\dagger} = \sum_{j\in\mathcal{S}^\dagger} |w_j + z_j|\,|w_j + z_j|^{q-1}$$

Table B.2
Predictive performance (AUC× 100) for the various feature ranking methods related to SMOTE-SF. High-dimensional datasets.

	200% Oversampling				400% Oversampling			
	k-NN	LR	NB	SVM	*k*-NN	LR	NB	SVM
Burczynski								
FS	67.8	64.5	63.0	**78.8**	65.3	64.4	63.0	**78.8**
MI	67.3	67.8	64.4	**78.8**	64.4	70.3	63.0	**78.8**
EC	66.3	67.4	63.9	**78.8**	64.4	72.7	59.1	**78.8**
CFS	65.8	69.7	63.0	**78.8**	64.9	61.6	63.0	**78.8**
Lung								
FS	98.6	97.9	86.4	**100**	97.9	98.6	86.4	**100**
MI	98.6	99.3	86.4	**100**	97.9	97.2	86.4	**100**
EC	98.6	97.2	86.4	**100**	98.6	97.9	86.4	**100**
CFS	98.6	99.3	86.4	**100**	98.6	98.6	86.4	**100**
Glioma								
FS	98.8	97.5	50.0	90.9	98.5	98.2	50.0	90.9
MI	98.8	97.9	50.0	90.9	98.2	**99.7**	50.0	90.9
EC	**99.1**	97.5	50.0	90.9	98.2	96.6	50.0	90.9
CFS	98.8	96.9	50.0	90.9	97.9	98.8	50.0	90.9
SRBCT								
FS	**94.2**	79.9	50.0	75.1	93.0	75.2	50.0	75.1
MI	**94.2**	78.7	50.0	75.1	93.0	78.7	50.0	75.1
EC	**94.2**	75.2	50.0	75.1	**94.2**	74.1	50.0	75.1
CFS	**94.2**	76.4	50.0	75.1	93.0	74.1	50.0	75.1
Lung2								
FS	**100**	99.7	50.0	**100**	99.7	99.5	50.0	**100**
MI	99.7	99.2	50.0	**100**	99.7	99.5	50.0	**100**
EC	99.7	99.7	50.0	**100**	99.7	99.2	50.0	**100**
CFS	99.7	99.7	50.0	**100**	99.7	99.5	50.0	**100**
Bullinger								
FS	82.8	80.3	81.3	**93.8**	79.4	83.2	81.3	**93.8**
MI	82.8	84.3	81.3	**93.8**	80.0	85.4	81.3	**93.8**
EC	82.2	79.7	81.3	**93.8**	78.9	84.9	81.3	**93.8**
CFS	82.8	81.4	81.3	**93.8**	81.1	84.3	81.3	**93.8**
CAR								
FS	99.7	85.9	95.2	98.4	99.7	91.5	95.2	98.4
MI	**100**	88.9	95.2	98.4	99.7	87.7	95.2	98.4
EC	**100**	87.3	95.2	98.4	**100**	86.9	95.2	98.4
CFS	99.3	88.8	95.2	98.4	99.7	86.6	95.2	98.4

$$
\begin{aligned}
&\leq \sum_{j\in\mathcal{S}^\dagger} |w_j|\,|w_j+z_j|^{q-1} + \sum_{j\in\mathcal{S}^\dagger} |z_j|\,|w_j+z_j|^{q-1}\\
&\leq \|\mathbf{w}\|_{q,\mathcal{S}^\dagger}\left(\sum_{j\in\mathcal{S}^\dagger} |w_j+z_j|^{(q-1)q'}\right)^{1/q'}\\
&+ \|\mathbf{z}\|_{q,\mathcal{S}^\dagger}\left(\sum_{j\in\mathcal{S}^\dagger} |w_j+z_j|^{(q-1)q'}\right)^{1/q'}\\
&= (\|\mathbf{w}\|_{q,\mathcal{S}^\dagger} + \|\mathbf{z}\|_{q,\mathcal{S}^\dagger})\|\mathbf{w}+\mathbf{z}\|_{q,\mathcal{S}^\dagger}^{q-1},
\end{aligned}
$$

where the first inequality follows from triangle inequality, and the second one from Hölder's Inequality. The desired inequality holds thanks to the last relation. □

Lemma 2 (*Hölder's Inequality, [53]*)**.** *For any* $\mathbf{w},\mathbf{z}\in\Re^n$*, and* $q\geq 1$*, one has that* $\sum_{j\in\mathcal{S}^\dagger}|w_j z_j| \leq \|\mathbf{w}\|_{q,\mathcal{S}^\dagger}\|\mathbf{z}\|_{q',\mathcal{S}^\dagger}$*, where* $\frac{1}{q}+\frac{1}{q'}=1$*.*

Appendix B. Performance summary in terms of AUC

It is quite evident from Tables B.1 and B.2 that there is no feature ranking method that outperform others in terms of AUC, since each method achieves the best performance at least a few times, and the differences are usually very small. Similarly, the same conclusion can be drawn for *p*, the Minkowski distance parameter, as can be observed in Tables B.3 and B.4.

Table B.3
Predictive performance (AUC× 100) for the various distance metric parameters related to SMOTE-SF. Low-dimensional datasets.

	200% Oversampling				400% Oversampling			
	k-NN	LR	NB	SVM	*k*-NN	LR	NB	SVM
Ecoli								
	k-NN	LR	NB	SVM	*k*-NN	LR	NB	SVM
$p=\infty$	87.1	88.1	**90.9**	89.3	87.4	90.6	**91.3**	90.9
$p=1$	85.9	87.8	90.1	89.6	88.4	90.8	90.7	89.8
$p=2$	86.9	88.1	89.8	89.4	87.9	90.6	91.1	90.1
Abalone								
$p=\infty$	69.1	68.1	77.6	50.0	71.9	77.5	**77.9**	76.7
$p=1$	69.0	68.1	77.8	50.0	72.3	77.4	**77.9**	77.1
$p=2$	**69.4**	68.4	77.9	50.0	71.3	77.3	77.8	76.9
CarEval								
$p=\infty$	49.0	98.5	75.3	98.7	48.4	98.8	79.0	98.9
$p=1$	49.1	98.5	76.8	99.1	48.4	98.8	79.4	**99.0**
$p=2$	49.0	98.5	77.1	**99.2**	48.3	98.7	79.7	**99.0**
Solar								
$p=\infty$	63.9	63.9	**73.0**	61.3	68.6	65.7	73.3	66.1
$p=1$	65.1	62.9	72.2	61.3	65.8	66.4	**73.6**	66.1
$p=2$	63.6	65.3	70.9	61.2	67.4	66.1	73.5	66.4
Yeast								
$p=\infty$	74.6	70.0	**79.9**	50.0	77.8	78.0	80.3	71.5
$p=1$	76.7	70.2	79.0	50.0	76.1	78.1	80.7	71.6
$p=2$	73.1	70.0	80.1	50.0	77.0	77.9	**80.9**	72.5

Table B.4
Predictive performance (AUC× 100) for the various distance metric parameters related to SMOTE-SF. High-dimensional datasets.

	200% Oversampling				400% Oversampling			
	k-NN	LR	NB	SVM	*k*-NN	LR	NB	SVM
Burczynski								
$p=\infty$	**81.3**	**81.3**	63.0	63.0	**93.8**	**93.8**	78.8	78.8
$p=1$	**81.3**	**81.3**	64.4	63.0	**93.8**	**93.8**	78.8	78.8
$p=2$	**81.3**	**81.3**	64.4	63.0	**93.8**	**93.8**	78.8	78.8
Lung								
$p=\infty$	98.6	99.3	86.4	**100**	98.6	97.9	86.4	**100**
$p=1$	98.6	99.3	86.4	**100**	97.9	98.6	86.4	**100**
$p=2$	98.6	97.9	86.4	**100**	98.6	98.6	86.4	**100**
Glioma								
$p=\infty$	**99.1**	97.9	50.0	90.9	98.2	98.8	50.0	90.9
$p=1$	**99.1**	97.5	50.0	90.9	97.9	97.9	50.0	90.9
$p=2$	98.8	97.5	50.0	90.9	98.5	**99.7**	50.0	90.9
SRBCT								
$p=\infty$	**94.2**	79.9	50.0	75.1	**94.2**	78.7	50.0	75.1
$p=1$	**94.2**	76.4	50.0	75.1	91.9	76.4	50.0	75.1
$p=2$	**94.2**	75.2	50.0	75.1	93.0	76.4	50.0	75.1
Lung2								
$p=\infty$	**100**	99.7	50.0	**100**	99.7	98.6	50.0	**100**
$p=1$	99.7	99.7	50.0	**100**	99.7	99.5	50.0	**100**
$p=2$	99.7	99.7	50.0	**100**	99.7	99.5	50.0	**100**
Bullinger								
$p=\infty$	**82.2**	80.0	66.3	64.9	84.3	83.2	67.8	70.3
$p=1$	**82.8**	81.1	67.3	64.9	81.4	82.6	67.4	70.3
$p=2$	**82.8**	79.4	67.8	65.3	83.8	**85.4**	69.7	72.7
CAR								
$p=\infty$	**100**	86.5	95.2	98.4	**100**	87.3	95.2	98.4
$p=1$	**100**	88.9	95.2	98.4	99.7	91.5	95.2	98.4
$p=2$	**100**	87.3	95.2	98.4	99.7	86.6	95.2	98.4

References

[1] A. Fernández, S. del Río, N.V. Chawla, F. Herrera, An insight into imbalanced big data classification: outcomes and challenges, Complex Intell. Syst. 3 (2) (2017) 105–120.

[2] H. He, E. García, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (2009) 1263–1284.

[3] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, Inform. Sci. 250 (2013) 113–141.

[4] S. Maldonado, A. Flores, T. Verbraken, B. Baesens, R. Weber, Profit-based feature selection using support vector machines - general framework and an application for customer churn prediction, Appl. Soft Comput. 35 (2015) 740–748.

[5] Z. Zheng, X. Wu, R. Srihari, Feature selection for text categorization on imbalanced data, SIGKDD Explor. 6 (2004) 80–89.

[6] A. Al-shahib, R. Breitling, D. Gilbert, Feature selection and the class imbalance problem in predicting protein function from sequence, Appl. Bioinformatics 4 (2005) 195–203.

[7] R. Blagus, L. Lusa, SMOTE for high-dimensional class-imbalanced data, BMC Bioinformatics 14 (2013) 106.

[8] K.-J. Wang, B. Makond, K.-H. Chen, K.-M. Wang, A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients, Appl. Soft Comput. 20 (2014) 15–24.

[9] N. Verbiest, E. Ramentol, C. Cornelis, F. Herrera, Preprocessing noisy imbalanced datasets using SMOTE enhanced with fuzzy rough prototype selection, Appl. Soft Comput. 22 (2014) 511–517.

[10] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explor. 6 (2004) 1–6.

[11] Y.M. Sun, M.S. Kamel, A.K.C. Wong, Classification of imbalanced data: A Review, Int. J. Pattern Recognit. Artif. Intell. 23 (2009) 687–719.

[12] N.V. Chawla, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer, SMOTE: Synthetic minority oversampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.

[13] I. Nekooeimehr, S.K. Lai-Yuen, Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets, Expert Syst. Appl. 46 (2016) 405–416.

[14] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, Handling imbalanced datasets: A review, GESTS Int. Trans. Comput. Sci. Eng. 30 (2006) 25–36.

[15] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing ICIC 2005: Advances in Intelligent Computing, in: Lecture Notes in Computer Science, vol. 3644, Springer-Verlag Berlin Heidelberg, 2005, pp. 878–887.

[16] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem, in: PAKDD 2009: Advances in Knowledge Discovery and Data Mining, in: Lecture Notes in Computer Science, vol. 5476, Springer-Verlag Berlin Heidelberg, 2009, pp. 475–482.

[17] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: Proceedings of the IEEE International Joint Conference on Computational Intelligence IJCNN 2008, 2008, pp. 1322–1328.

[18] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. System Sci. 55 (1) (1997) 119–139.

[19] S. Barua, M. Islam, X. Yao, K. Murase, MWMOTE - majority weighted minority oversampling technique for imbalanced data set learning, IEEE Trans. Knowl. Data Eng. 26 (2) (2014) 405–425.

[20] L. Ma, S. Fan, CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests, BMC Bioinformatics 18 (2017) 169.

[21] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Technique, 3 edition, Morgan Kaufmann, Waltham, MA, USA, 2011.

[22] R. Blagus, L. Lusa, Class prediction for high-dimensional class-imbalanced data, BMC Bioinformatics 11 (2010) 523.

[23] A.A. Shanab, T.M. Khoshgoftaar, R. Wald, J. Van Hulse, Comparison of approaches to alleviate problems with high-dimensional and class-imbalanced data., 2011 IEEE Int. Conf. Inf. Reuse Integr. (IRI) (2011) 234–239.

[24] J. Van Hulse, T.M. Khoshgoftaar, A. Napolitano, R. Wald, Feature selection with high-dimensional imbalanced data, in: Proceedings of the IEEE International Conference on Data Mining Workshops, 2009, pp. 507–514.

[25] R. Martín-Félez, R.A. Mollineda, On the suitability of combining feature selection and resampling to manage data complexity, in: P. Meseguer, L. Mandow, R.M. Gasca (Eds.), Current Topics in Artificial Intelligence, CAEPIA 2009, in: Lecture Notes in Computer Science, vol. 5988, Springer, Berlin, Heidelberg, 2010, pp. 141–150.

[26] R.O. Duda, P.E. Hard, D.G. Stork, Pattern classification, Wiley-Interscience Publication, 2001.

[27] J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (2014) 175–186.

[28] M.A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: Proceedings of the Seventeenth International Conference on Machine Learning, 2000, pp. 359–366.

[29] G. Roffo, S. Melzi, New frontiers in mining complex patterns, fifth international workshop, nfmcp2016, in: A. Appice, M. Ceci, C. Loglisci, E. Masciari, Z.W. Ras (Eds.), chapter Ranking to Learn: Feature Ranking and Selection via Eigenvector Centrality, in: Lecture Notes in Computer Science, Springer, 2017, pp. 19–35.

[30] X. Chen, M. Wasikowski, FAST: a roc-based feature selection metric for small samples and imbalanced data classification problems, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD 2008, 2009, pp. 124–132.

[31] M. Alibeigi, S. Hashemi, A. Hamzeh, DBFS: An effective density based feature selection scheme for small sample size and high dimensional imbalanced data sets, Data Knowl. Eng. 81–82 (2012) 67–103.

[32] S. Maldonado, J. López, Dealing with high-dimensional class-imbalanced datasets: embedded feature selection for SVM classification, Appl. Soft Comput. 67 (2018) 94–105.

[33] P. Villar, A. Fernandez, R.A. Carrasco, F. Herrera, Feature selection and granularity learning in genetic fuzzy rule-based classification systems for highly imbalanced data-sets., Internat. J. Uncertain. Fuzziness Knowledge-Based Systems 20 (3) (2012) 369–397.

[34] S. Maldonado, R. Weber, F. Famili, Feature selection for high-dimensional class-imbalanced data sets using support vector machines, Inform. Sci. 286 (2014) 228–246.

[35] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[36] T. Deepa, M. Punithavalli, An E-SMOTE technique for feature selection in high-dimensional imbalanced dataset, in: Proceedings of the 3rd International Conference on Electronics Computer Technology, ICECT, 2011.

[37] N. Qazi, K. Raza, Effect of feature selection, SMOTE and under sampling on class imbalance classification, in: Proceedings of the 14th International Conference on Computer Modelling and Simulation, UKSim, 2012.

[38] A.K. Pal, P.K. Mondal, A.K. Ghosh, High dimensional nearest neighbor classification based on mean absolute differences of inter-point distances, Pattern Recognit. Lett. 74 (2016) 1–8.

[39] J.P. Van de Geer, Some Aspects of Minkowski Distance, in: Department of Data Theory, University of Leiden: Esearch report, Leiden University, Department of Data Theory, 1995.

[40] L. Song, A. Smola, A. Gretton, J. Bedo, K. Borgwardt, Feature selection via dependence maximization, J. Mach. Learn. Res. 13 (2012) 1393–1434.

[41] A. Asuncion, D.J. Newman, UCI machine learning repository, 2007, http://archive.ics.uci.edu/ml/.

[42] J. Alcalá-Fernández, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: A software tool to assess evolutionary algorithms to data mining problems, Soft Comput. 13 (3) (2009) 307–318.

[43] S. Maldonado, J. López, Imbalanced data classification using second-order cone programming support vector machines, Pattern Recognit. 47 (5) (2014) 2070–2079.

[44] K. Yang, Z. Cai, J. Li, G. Lin, A stable gene selection in microarray data analysis, BMC Bioinformatics 7 (2006) 228.

[45] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[46] J. Demšar, Statistical comparisons of classifiers over multiple data set, J. Mach. Learn. Res. (2006) 1–30.

[47] J. Fürnkranz, A study using n-gram features for text categorization, Austrian Res. Inst. Artif. Intell. 3 (1998) (1998) 1–10.

[48] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proc. 20th int. conf. very large data bases, VLDB, vol. 1215, 1994, pp. 487–499.

[49] J.C. Gower, A general coefficient of similarity and some of its properties, Biometrics 27 (1) (1971) 857–874.

[50] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature Extraction, Foundations and Applications, Springer, Berlin, 2006.

[51] B. Baesens, Analytics in a Big Data World, John Wiley and Sons, 2014.

[52] K.B. Schebesch, R. Stecking, Using multiple SVM models for unbalanced credit scoring data sets, in: chapter Data Analysis, Machine Learning and Applications, Springer, Berlin Heidelberg, 2008, pp. 515–522.

[53] A. Brown, C. Pearcy, An Introduction to Analysis, Springer-Verlag New York, 1995.