Information Sciences 505 (2019) 32–64

Contents lists available at ScienceDirect

Information Sciences

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

A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance

Dina Elreedy*, Amir F. Atiya

Computer Engineering Department, Cairo University, Giza, Egypt

ARTICLE INFO

Article history:
Received 5 November 2018
Revised 17 July 2019
Accepted 19 July 2019
Available online 19 July 2019

Keywords:
Unbalanced data
Minority class
Over-sampling
Data level
SMOTE

ABSTRACT

Imbalanced classification problems are often encountered in many applications. The challenge is that there is a minority class that has typically very little data and is often the focus of attention. One approach for handling imbalance is to generate extra data from the minority class, to overcome its shortage of data. The Synthetic Minority over-sampling TEchnique (SMOTE) is one of the dominant methods in the literature that achieves this extra sample generation. It is based on generating examples on the lines connecting a point and one its *K*-nearest neighbors. This paper presents a theoretical and experimental analysis of the SMOTE method. We explore the accuracy of how faithful it emulates the underlying density. To our knowledge, this is the first mathematical analysis of the SMOTE method. Moreover, we analyze the effect of the different factors on generation accuracy, such as the dimension, size of the training set and the considered number of neighbors *K*. We also provide a qualitative analysis that examines the factors affecting its accuracy. In addition, we explore the impact of SMOTE on classification boundary, and classification performance.

1. Introduction

Imbalanced learning is one of the most challenging problems in data mining. In imbalance learning we have one or more classes with very little data, and some others with sufficient amount of data. Imbalanced data sets are particularly prevalent in “anomaly-type” classification problems. These are a broad category of problems that classify between a “normal” class and an “anomaly” class, where anomaly can denote a variety of possible meanings. Examples are medical diagnosis [37] and machine fault detection [28], and fraud detection [6]. In these types of problems the minority class deserves special attention, because it represents the phenomenon that we seek to predict, from among a plethora of majority class patterns denoting normal operation.

Standard classifiers are designed to minimize the overall classification error irrespective of the class distribution. Therefore, such classifiers’ performance becomes biased towards the majority class examples while sacrificing minority class accuracy. The approaches for handling the data imbalance problem are mainly classified into the following categories: the cost sensitive approach, the algorithm level approach, and the data level approach.

The cost sensitive approach uses cost matrices to set misclassification costs according to the importance of the class and degree of imbalance. Examples of work on the cost sensitive approach include AdaCost [15] and the work in [40].

* Corresponding author at: Faculty of Engineering, Cairo University Street, Giza, Egypt, 12613.
E-mail addresses: dinaelreedy@email.wustl.edu (D. Elreedy), amir@alumni.caltech.edu (A.F. Atiya).

https://doi.org/10.1016/j.ins.2019.07.070
0020-0255/

The algorithm level approach tailors the classification algorithm's specifics to take into account the imbalance issue. For example, there is work on modifying the K nearest neighbor classifier (KNN) [13], other work on tailoring decision trees [31], and some approaches that modify support vector machines (SVM) [25,44], all in a way that puts special focus on the minority class(es).

The data level approach is based on modifying the data itself by attempting to rebalance the minority and the majority classes. This can be performed either by removing some instances of the majority class (under-sampling), or by increasing the number of minority class instances (over-sampling). It is the most dominant imbalance approach, because of its simplicity and because it is a general approach which can be applied independent of the classifier being used. It is simply an overlay of sampling, with the classifier in tact underneath it. This approach is the focus of this work.

Under-sampling is performed by removing less important patterns, either by random selection or by using some heuristic rules. For example, the condensed nearest neighbor rule [16] and one-sided selection [2] are some types of under-sampling. However, under-sampling is risky as potential important information could be lost. The other probably more successful alternative is over-sampling [2]. It is accomplished either by randomly replicating minority class patterns, or by generating new minority class patterns. Sampling algorithms are then likely combined with ensemble learning. This means that we generate multiple sample sets, and for each set we design a classifier. Subsequently, we combine the classification of these classifiers. Examples of such models are SMOTEBoost [8] and Ranked Minority over-sampling in Boosting (RAMOBoost), which combine ensembles with over-sampling [9]. On the other hand, other techniques like RUSBoost [38], EasyEnsemble, and BalanceCascade [32] are based on creating an ensemble of data consisting of under-sampled majority class. The advantage of over-sampling is that it compensates for the shortage of data of the minority class by generating extra data. The problem, however, is that the underlying distribution is not known, and the challenge is to emulate this distribution as much as possible when generating new data.

A very successful technique for generating new data is the so-called "Synthetic Minority Over-sampling Technique", or SMOTE [7]. It is based on sampling data from the minority class by simply generating data points on the line segment connecting a randomly selected data point and one of its K-nearest neighbors. This approach is very simple, and extremely successful in practice, therefore it became very wide-spread. The only problem with SMOTE [7] is that it is not grounded on a solid mathematical theory. The purpose of this work is to address this shortcoming and provide an in-depth analysis of the SMOTE procedure.

In this work, we present an analysis of the distribution characteristics of the synthetic samples generated by SMOTE. This helps in evaluating the quality of the data, in the sense of how well do the generated data emulate the true underlying distribution. In our experiments we evaluate how the oversampled patterns deviate from the original distribution. It is important to focus on the distributional aspects, because the distributions of the classes dictate the shape of the classification boundary. This is given explicitly in the case of the Bayes classifier, and implicitly for the other classifiers. Even though SMOTE is not explicitly designed to reproduce the underlying distribution, the distribution plays a fundamental role in shaping the classification boundary. In addition to our presented distributional analysis, we provide an analysis of the impact of SMOTE on classification performance, since the classification performance is the ultimate goal of using SMOTE. Specifically, our goals are the following:

- Develop a mathematical analysis of SMOTE, and test the degree of its emulation of the underlying distribution (by checking its moments). The presented theoretical analysis is general, and applies for any distribution.
- Apply the general theoretical analysis to two distributions: multivariate Gaussian, and multivariate Laplacian distribution, in order to obtain simplified closed-form formulas for mean, and covariance of the distribution of the oversampled patterns.
- Provide a detailed experimental study of SMOTE, exploring the factors that affect its accuracy (in mimicking the distribution). For example we found, both theoretically and empirically, that the accuracy deteriorates when the number of original minority patterns decreases, when the dimension increases, and when the number of neighbors used to apply SMOTE sampling increases.
- Explore the performance of SMOTE for some classifiers, both theoretically and empirically, by also investigating the effect of different factors on their performance.
- Present a comprehensive empirical analysis of SMOTE, and three popular SMOTE extensions (Borderline SMOTE1, Borderline SMOTE2, and Adasyn), which aims to evaluate these oversampling methods in terms of both distribution, and classification performance.

The paper is organized as follows: Section 2 presents a literature review. Section 3 introduces the SMOTE technique stating its advantages and potential drawbacks. Section 4 analyzes the distribution of the patterns generated using SMOTE by deriving its mean and covariance matrix. Then, experimental analysis is performed to analyze SMOTE in Section 5. After that, Section 6 analyzes the experimental and theoretic findings and discusses some of the issues and pitfalls of SMOTE. Finally, Section 7 concludes the paper and mentions potential future work.

2. Related work

2.1. SMOTE Analysis and Comparative Studies

Although there are comprehensive studies in the literature discussing sampling methods for the class imbalance problem (see the reviews [21,26,33]), there is little work, if any, that provides a theoretical analysis of these methods.

Among the empirical analysis work is the study by Luengo et al. [34]. They analyze the behavior of different sampling methods, including SMOTE, one of its extensions called SMOTE-ENN, and an evolutionary under-sampling method EUSCHC [19], by evaluating their effect on the "shape" of the resulting data. For example, they provide an analysis of the degree of pattern overlap between the different classes and an analysis of class separability and its geometrical properties. However, these measures do not consider distributional issues of the generated examples and instead they are more concerned with the characterization of the datasets. On the other hand, in this work, we are interested in investigating the distribution characteristics of SMOTE generated patterns, and to what degree they diverge from the original distribution. The following papers provide comparative studies of different sampling methods. García et al. [20] investigate two factors affecting re-sampling methods' performance: the employed classifier and the ratio between the numbers of minority and majority class examples. Their results show that for datasets having low or moderate class imbalance ratio, over-sampling outperforms under-sampling using local classifiers, i.e. those which depend on local neighborhood patterns in their classification such as the K-nearest neighbor classifier. However, some under-sampling methods outperform over-sampling when using classifiers having global learning schemes such as neural networks.

In [14], the authors perform an empirical analysis of different under-sampling methods, over-sampling methods, and combinations thereof. Additionally, they investigate the use of different over-sampling and under-sampling rates in a study of Alzehimer's disease classification. Their experimental analysis includes random over-sampling, SMOTE, random under-sampling, and K-Medoids under-sampling. K-Medoids under-sampling is a clustering-based method, whereby N clusters for the majority class patterns are created, where N is the size of minority class patterns. Then, the final dataset would be the final cluster centers of the majority class clusters and all of the minority class patterns. Their results show that the more "engineered" sampling methods, i.e. SMOTE and K-Medoids, outperform random over-sampling and random under-sampling.

The authors of [39] study extreme class imbalance, where the number of minority class samples is very scarce. In their work, the authors demonstrate that the SMOTE over-sampling method typically exhibits poor performance when there are only a few minority class patterns since the potential information of the minority patterns would be insufficient to generate representative synthetic samples. Therefore, they develop a new oversampling method, named Sampling WIth the Majority (SWIM), which makes use of the majority class distribution for generating more distribution-oriented synthetic minority class patterns. The SWIM method basically estimates the majority class distribution from the plentiful majority samples. Then it generates synthetic minority patterns with the same relative Mahalanobis distance, as the original minority patterns with respect to the estimated majority class distribution.

In this work, we study the impact of the number of minority patterns on the SMOTE distribution characteristics, and on the classification performance as well. We perform a theoretical analysis in Section 4, and an empirical analysis in Section 5. The results of our provided analyses support the findings inferred in [39], that the SMOTE performance deteriorates as the number of minority patterns decreases.

2.2. Studying SMOTE's impact on Classification Boundary

A very important aspect to be considered when studying synthetic oversampling is the impact of the generated patterns on the decision boundary. Wallace et al. [42] present a study of the impact of SMOTE generated samples on the decision boundary, and their study concludes that the SMOTE oversampling method incurs a bias towards minority class. They perform this analysis based on cost-sensitive learning, where different costs are imposed on minority and majority classes. Then, risk minimization is applied to find the optimal class separator.

Another study, presented in [45], investigates the impact of unbalanced data on linear discriminant analysis (LDA) classifier. This paper performs an empirical study of some basic over-sampling and under-sampling methods including SMOTE. It shows the superiority of over-sampling approach over under-sampling for the LDA classifier.

In this work, we also consider the impact of SMOTE on the classification boundary of LDA, among investigating other classifiers as well.

2.3. Application of SMOTE to Complex datasets

With researchers dealing with ever more difficult classification tasks, problems with complex datasets come up frequently. For example, high dimensional data, problems with hierarchical subcategories, problems where the data lie on a manifold, etc are very often encountered in practice. There has been some research that studies the application of SMOTE to these complex-type datasets, including sparse, and high dimensional datasets. The work presented in [41] studies the impact of sparseness and high dimensionality on class imbalance problem for behavioral data, which are data related to actions or interactions between certain objects or users. In their study, the authors adapt several over-sampling, under-sampling, and cost-sensitive learning methods to work on sparse and high dimensional unbalanced behavioral data. The authors of

[35] provide some theoretical analysis that studies the effect of SMOTE on the distances, and the correlations between the samples. They also analyze the application of SMOTE to high dimensional datasets, and show that SMOTE has performance limitations when applying it to high-dimensional data. Bellinger et al. consider the application of synthetic over-sampling to high dimensional problems [3]. The paper considers datasets that have manifold property where high-dimensional data can be effectively presented in lower-dimensional spaces. The authors argue that SMOTE is not effective for high dimensional data. Consequently, they propose new manifold based over-sampling methods to cope with high-dimensional datasets having the manifold property. First, they apply data transformation using principal component analysis (PCA), or autoencoder. Then, synthetic over-sampling is applied in the manifold space. Finally, the generated samples are mapped back to the original feature space.

In this work, we investigate the application of SMOTE to high dimensional artificial and real world datasets. However, we do not only study the classification accuracy, but we also evaluate distributional characteristics of the synthetic samples generated by SMOTE as well.

2.4. SMOTE Extensions

The aforementioned papers focus on the analysis of sampling methods and their influencing parameters. There has been much work on modifying SMOTE to make it more effective. We will be brief in the review of these works, as the purpose of this paper is to provide an analysis, rather than propose new modifications or new sampling methods. Some of the SMOTE modification methods are borrowed from the accepted idea that classification performance tends to be better if the classifier puts more emphasis on patterns near the classification boundary. For example, two variations of Borderline SMOTE are presented in [22]: Borderline-SMOTE1 and Borderline-SMOTE2. In these methods only the minority examples near the borderline are over-sampled. The first approach, Borderline-SMOTE1, identifies borderline examples using the K nearest neighbor rule, and then applies SMOTE only to these borderline examples. The second approach, Borderline-SMOTE2, allows for interpolating with majority class neighbors and sets the generated example closer to the minority class when the chosen nearest neighbor is a majority class example.

Another model, the so-called Safe-Level-SMOTE algorithm [5], uses an argument contrary to Borderline SMOTE. It seeks to generate synthetic examples in safe areas of minority class, rather than class overlap areas. It takes a risk-averse approach, which avoids messing with regions of the majority class.

SMOTE extensions using local neighborhood characteristics include the Modified Synthetic Minority Over-sampling TEchnique (MSMOTE), presented in [24], which labels the minority class examples as safe, border and noisy instances according to their neighborhood characteristics. It then applies SMOTE in case of safe examples, discards noisy examples and chooses the nearest neighbor as a candidate for interpolation in case of border examples.

Another model is the so-called Adaptive Synthetic Sampling Approach for Imbalanced Learning (ADASYN) [23]. It uses a weighted distribution for different minority class examples according to their level of difficulty in learning. A well-known approach that has extended ADASYN is the Ranked Minority over-sampling in Boosting (RAMOBoost) introduced in [9]. RAMOBoost combines the generation of new minority examples with the AdaBoost.M2 boosting algorithm [17].

In this work, to explore the behavior of SMOTE on classification problems. We have also tested other variants of SMOTE, including the two methods of Borderline-SMOTE1, and Borderline-SMOTE2 presented in [22], and Adasyn [23].

From the above discussion of the contributions related to SMOTE, we can observe that most of the studies in the literature are only empirical. To make up for this deficit, we seek to provide a mathematical analysis of the SMOTE method in this work. This is performed by investigating the distribution of SMOTE generated patterns, and by studying how it diverges from the distribution of the original samples. In addition, we aim to extend the existing experimental work in the literature by exploring how SMOTE's accuracy is influenced by the number of minority data points, the dimension of the problem, and the K pertaining to the K-nearest neighbor used for generating SMOTE samples. This analysis also sheds some light into the behavior of SMOTE in high dimensions. Furthermore, we provide an empirical analysis of SMOTE's effect on the decision boundary, by providing a mathematical analysis (Section 4.4), in addition to an experimental analysis (Section 5.3). Furthermore, we perform a comprehensive experimental analysis of SMOTE, and three popular oversampling methods that are extensions of SMOTE, and evaluate their classification, and distribution performance on real datasets. The details of this experimental analysis are presented in Section 5.5.

3. Introduction to SMOTE

The SMOTE over-sampling procedure consists of the following simple steps:

- For each pattern X_0 from the minority class do the following:
  - Pick one of its K nearest neighbors X (belonging to the minority class also).
  - Create a new pattern Z on a random point on the line segment connecting the pattern and the selected neighbor, as follows:

$$Z = X_0 + w(X - X_0) \tag{1}$$

where w is a uniform random variable in the range [0,1].

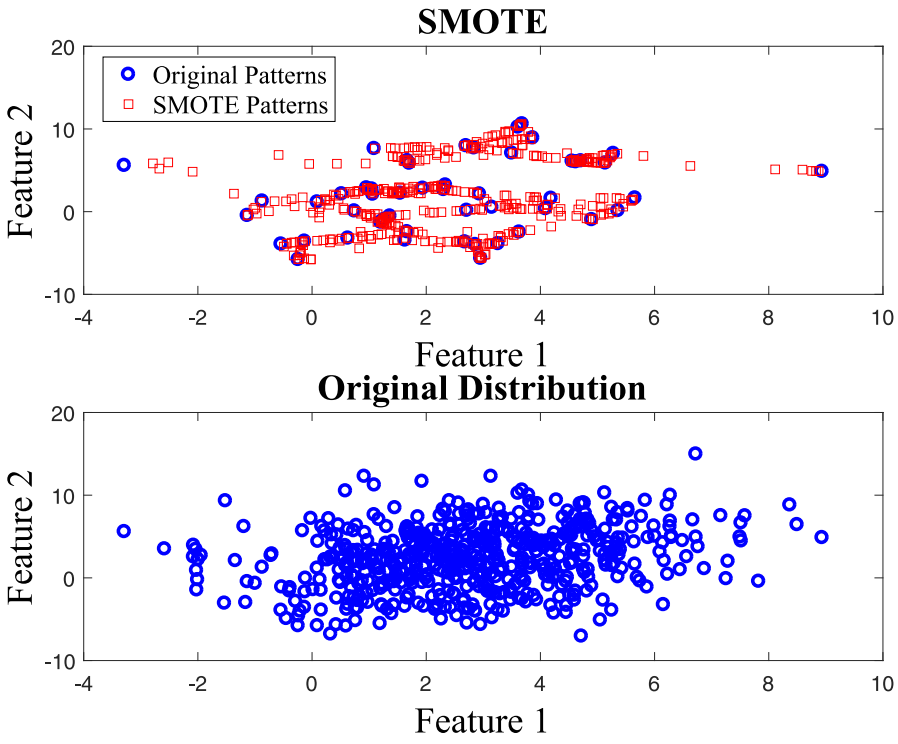

Fig. 1. SMOTE Generated vs. Original Examples.

Fig. 1 shows an example of patterns generated by SMOTE. As a comparison, this figure also shows extra patterns generated from the original distribution. One can observe that the SMOTE generated patterns are more contracted than the patterns generated from the true distribution. This is because the SMOTE generation process by linear interpolation causes them to be inward-placed as shown in Fig. 1. The other problem of SMOTE is that patterns are allocated only on the line segments connecting the K neighbors, creating an unrealistic graph shape, where edges are flooded with data points and internal portions rarely have samples. This problem is accentuated even more in higher dimensions. Fig. 1 shows how SMOTE generated patterns cluster around some paths, with some empty spaces around them.

As discussed in [5], another encountered problem is that SMOTE may erroneously generate synthetic examples invading the majority class decision region especially in case of overlapping classes. Another source of inaccuracy for SMOTE occurs when there is a clique of two or three patterns that are K nearest neighbors of each other. In such case the examples generated using these patterns will be over-represented, leading to some statistical bias. A similar issue with the SMOTE method is related to the within-class imbalance problem. In case the minority class has different clusters as small disjuncts [27,43], SMOTE could erroneously create synthetic examples connecting between the different clusters. Majority Weighted Minority Oversampling TEchnique (MWMOTE), is a SMOTE based technique that alleviates this problem. MWMOTE first identifies hard-to-learn minority class examples, then it assigns them weights according to their distance from majority class examples. After that, MWMOTE generates new minority class examples according to the assigned weights using agglomerative clustering. Douzas et al. propose another over-sampling method based on SMOTE and k-means clustering [11]. Their method aims to reduce the generation of noisy patterns and handling not only imbalance between classes but within class imbalance as well.

The underlying objective of applying SMOTE is to shift the classification decision boundary in favor of minority class decision region. To achieve that, the main idea of the SMOTE over-sampling method is to generate minority class samples that are similar to the original samples. This claimed similarity is based on the assumption of similarity in the density among K nearest neighbors. However, this assumption is not mathematically exact as we will prove in our theoretical and empirical analysis, presented in Section 4, and Section 5, respectively. The divergence of the SMOTE-generated patterns from the original distribution is large in case of small number of original minority samples, since the aforementioned similarity assumption may not be accurate, especially for complex datasets that are sparse, and have sub-concepts. Moreover, for sparse, and high dimensional data, the density similarity assumption may not hold as addressed in [3]. This discrepancy of the SMOTE-generated patterns from the original data distribution, especially for high-dimensional, sparse, or small number of original minority samples, could essentially hurt classification performance as will be discussed in Section 4, and Section 5.

4. Theoretical analysis of SMOTE

4.1. The case of general distribution

To provide some mathematical basis of the SMOTE method we will develop here a theoretical analysis. The success of SMOTE as a valid sampling algorithm hinges on its ability to generate patterns obeying a distribution close to the true one. We will investigate this issue here. The mean vector and the covariance matrix are the two major parameters characterizing any distribution. We derive approximate formulas for the mean and the covariance matrix of patterns generated according to SMOTE, and compare them with the true values.

Let $\Delta = X - X_0$, then:

$$Z = X_0 + w\Delta \tag{2}$$

where w is a uniformly generated number in [0, w^*]. The parameter w^*, typically greater than or equal one, allows us to both extrapolate and interpolate on the line connecting the pattern X_0 and its randomly selected neighbor X. If $w^* = 1$ then this reverts back to the original SMOTE (applying only interpolation). If $w^* > 1$, then we can go beyond point X_0, i.e. we are allowing some level of extrapolation.

The derivations are lengthy and are presented in Appendix A. The final approximations of the mean and covariance matrix of the generated pattern vector Z are given by the following Eq. (3) and Eq. (4) respectively. It is worth noting that the following Eqs. (3) and (4), and their corresponding derivations presented in A.1, and A.2, apply for any given distribution of minority class examples $p(X_0)$. As we will see next two subsections, we can even simplify the formulas further, and obtain closed-form solutions for some well-known distribution examples, such as the multivariate Gaussian distribution, and the multivariate Laplacian distribution.

$$E[Z] \approx \mu_{X_0} + \frac{Cw^*}{2} \int_{X_0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X} \, dX_0 \tag{3}$$

where $[\frac{\partial p(X)}{\partial X}]^T = (\frac{\partial p(X)}{\partial x_1}, \ldots., \frac{\partial p(X)}{\partial x_d})$.

$$\begin{aligned} \Sigma_Z = \Sigma_{X_0} &+ \frac{Cw^{*2}}{3} \int_{X_0} p(X_0)^{1-\frac{2}{d}} \, dX_0 I - \frac{C^2 w^{*2}}{3} \int_{X_0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X} \, dX_0 \int_{X_0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X}^T \, dX_0 \\ &+ \frac{Cw^*}{2} \left[\int_{X_0} p(X_0)^{-\frac{2}{d}} \frac{\partial p(X_0)}{\partial X} [(X_0 - \mu_{X_0})^T] \, dX_0 + \int_{X_0} p(X_0)^{-\frac{2}{d}} (X_0 - \mu_{X_0}) \frac{\partial p(X_0)}{\partial X}^T \, dX_0 \right] \end{aligned} \tag{4}$$

where d is the dimension of the pattern vector, N is the number of original patterns of the considered class, μ_{X_0} is the true mean vector of the class, Σ_{X_0} is the true covariance function, $p(X_0)$ is the class-conditional density at point X_0, I is the identity matrix, and C is calculated as follows:

$$C = \frac{N! \Gamma\left(1 + \frac{2}{d}\right)^{\frac{2}{d}} \Gamma\left(K + \frac{2}{d} + 1\right)}{\pi K! (d+2) \Gamma\left(N + \frac{2}{d} + 1\right)} \tag{5}$$

4.2. The case of multivariate Gaussian distribution

If the true probability density is multivariate Gaussian, then the approximations can be simplified further to the following:

$$E[Z] \approx \mu_{X_0} \tag{6}$$

$$\Sigma_Z = \Sigma_{X_0} + \left[(2\pi)^{\frac{1-d}{2}} \frac{Cw^{*2}}{3} \det^{\frac{1-d}{2d}} (\Sigma_{X_0}) \left(\frac{d}{2d-1} \right)^{\frac{d}{2}} - 2\pi Cw^* \det^{\frac{1}{d}} (\Sigma_{X_0}) \left(\frac{d}{d-2} \right)^{\frac{d+2}{2}} \right] I \tag{7}$$

where det denotes the determinant. The previous formula (7) is valid for dimensions $d > 2$ because of some convergence condition. From Eq. (7), since the fraction $\frac{d}{d-2}$ is greater than one for any $d > 2$, hence the third term of the generated examples' covariance matrix Σ_Z would be negative. It can be observed from Eq. (7) that the second term is less than the third term in magnitude except for large w^* (for standard SMOTE, $w^* = 1$). Accordingly, the covariance matrix Σ_Z of the patterns generated by SMOTE would be more contracted than that of original minority class examples Σ_{X_0}. This is because one subtracts the identity matrix times a positive value, and hence the diagonal elements become smaller.

From the above formulas one can observe the following:

- The mean vector of SMOTE-generated patterns is very close to the true one.
- The covariance matrix has some discrepancy. It is more contractive than the true one. This agrees with the intuitive argument discussed last section, which says that the generation process places the patterns more inwards.

Even though these results focus on the distributional characteristics of SMOTE, this would also impact the classification performance in an indirect way. According to Bayes rule, the posterior probability of the class $p(Y|X)$ is a function of the underlying class-conditional densities $p(X|Y)$. While most classifiers do not explicitly use these distributions, their classification results are implicitly impacted by the underlying distributions of the data, because the distributional properties of the data used in the design affect the classifier.

4.3. Case of other distributions: Multivariate laplace

We have used another example, where the true probability density follows multivariate Laplace distribution [29], which is defined as follows:

$$p(X) = \left(\frac{\alpha}{2}\right)^d e^{-\alpha \sum_{i=1}^{d} |X_i - \mu_i|} \tag{8}$$

where α is a distribution parameter, μ is the original mean of the distribution, and d is number of dimensions. Then, the approximations of mean and covariance of the SMOTE generated patterns can be derived to obtain the following (see A.4 for derivation details):

$$E[Z] \approx \mu_{X_0} \tag{9}$$

$$\Sigma_Z = \Sigma_{X_0} + \frac{4Cw^*}{\alpha^2}\left(\frac{d}{d-2}\right)^d \left[\frac{w^*}{3} - \left(\frac{d}{d-2}\right)\right] I \tag{10}$$

where C is calculated as defined in Eq. (5).

Similar to the Gaussian distribution case, the fraction $\frac{d}{d-2}$ is greater than one for any $d > 2$, hence the third term of the generated examples' covariance matrix Σ_Z would be negative and since for standard SMOTE, $w^* = 1$, the covariance matrix Σ_Z of the patterns generated by SMOTE would be more contracted than the original covariance matrix Σ_{X_0}.

4.4. Effect on the classification boundary

In this section, we study the impact of the SMOTE oversampling method on classification decision boundary. As an example of a classifier, we consider the Fisher linear discriminant analysis (LDA) classifier. The reason for this choice is that it can treat both majority and minority classes with equal footing, by weighing both covariance matrices equally in the formula.

To simplify matters, we perform a preliminary derivation for the discrepancy caused by SMOTE sampling. Following this, we perform an empirical study to show SMOTE's effect on classification performance for this LDA classifier and other classifiers.

In our analysis, we assume that the SMOTE generated examples' distribution obey the mean and covariance matrix as given by Eq. (3) and Eq. (4), respectively. Fisher LDA equations of w and w_0, for $w^T x + w_0 = 0$ decision boundary are defined as:

$$w = (\Sigma_{X0} + \Sigma_1)^{-1}(\mu_1 - \mu_{X0}) \tag{11}$$

$$w_0 = -\frac{1}{2}(\mu_{X0} + \mu_1)^T (\Sigma_{X0} + \Sigma_1)^{-1}(\mu_1 - \mu_{X0}) \tag{12}$$

where μ_1 and Σ_1 are the mean and covariance of the majority class distribution, and μ_{X0} and Σ_{X0} are the mean and covariance of the minority class distribution. The previous equation for w_0 can be expressed in terms of w as follows:

$$w_0 = -\frac{1}{2}(\mu_{X0} + \mu_1)^T w \tag{13}$$

After applying the SMOTE oversampling method, we can substitute with Eq. (3) and Eq. (4) in the Fisher linear discriminant (Eq. (11) and Eq. (12)):

$$w_z = (\Sigma_Z + \Sigma_1)^{-1}(\mu_1 - \mu_Z) \tag{14}$$

$$w_{0z} = -\frac{1}{2}(\mu_Z + \mu_1)^T (\Sigma_Z + \Sigma_1)^{-1}(\mu_1 - \mu_Z) \tag{15}$$

To measure the difference in decision boundary imposed by SMOTE, we evaluate two terms $\Delta_w = ||w_z - w||$ and $\Delta_{w_0} = ||w_{0z} - w_0||$.

$$\Delta_w = ||(\Sigma_Z + \Sigma_1)^{-1}(\mu_1 - \mu_Z) - (\Sigma_{X0} + \Sigma_1)^{-1}(\mu_1 - \mu_{X_0})|| \tag{16}$$

Let $A = \Sigma_{X0} + \Sigma_1$ and $b = \mu_1 - \mu_{X0}$. Also, let $T = \Sigma_Z - \Sigma_{X0}$ which encapsulates the second, third, and fourth terms in Eq. (4) and represents the discrepancy in the covariance matrix due to SMOTE. Let $h = \mu_Z - \mu_{X0}$, the second term in Eq. (3). Then, Δ_w is evaluated as:

$$\Delta_w = ||(A+T)^{-1}(b-h) - A^{-1}b|| \tag{17}$$

Using the following matrix inversion identity:

$$(A+BCD)^{-1} = A^{-1} - A^{-1}BCDA^{-1}(I+BCDA^{-1})^{-1} \tag{18}$$

Let $B = D = I$, where I is the identity matrix, since $A = \Sigma_{X0} + \Sigma_1$, then we assume that these covariance matrices are positive definite, and hence invertible. We obtain:

$$(A+T)^{-1} = A^{-1} - A^{-1}TA^{-1}(I+TA^{-1})^{-1} \tag{19}$$

Then Δ_w can be evaluated as:

$$\Delta_w = ||(A^{-1} - A^{-1}TA^{-1}(I+TA^{-1})^{-1})(b-h) - A^{-1}b|| \tag{20}$$

which can be simplified to:

$$\begin{aligned}\Delta_w &= ||-A^{-1}h - A^{-1}TA^{-1}(I+TA^{-1})^{-1})(b-h)|| \\ &= ||A^{-1}h + A^{-1}TA^{-1}(I+TA^{-1})^{-1})(b-h)||\end{aligned} \tag{21}$$

According to the theoretical analysis shown above in the case of multivariate Gaussian and multivariate Laplace distributions, the mean of the SMOTE generated patterns μ_Z is very close to the mean of original distribution μ_{X_0}, and hence $h \approx 0$. Accordingly, Eq. (21) would be as follows:

$$\begin{aligned}\Delta_w &= ||A^{-1}TA^{-1}(I+TA^{-1})^{-1})b|| \\ &= ||A^{-1}(AT^{-1})^{-1}(I+TA^{-1})^{-1}b|| \\ &= ||A^{-1}[(I+TA^{-1})AT^{-1})]^{-1}b||\end{aligned} \tag{22}$$

Hence, Δ_w is calculated as follows:

$$\Delta_w = ||A^{-1}(AT^{-1}+I)^{-1}b|| \tag{23}$$

Similar to Δ_{w_0}, since w_0 is a scalar, we express the difference in it using absolute difference:

$$\Delta_{w_0} = \frac{1}{2}|w_z^T(\mu_Z + \mu_1) - w^T(\mu_{X_0} + \mu_1)| \tag{24}$$

Let $\Delta = w_z - w$, then:

$$\Delta_{w_0} = \frac{1}{2}|(w+\Delta)^T(\mu_{X_0} + h + \mu_1) - w^T(\mu_{X_0} + \mu_1)| \tag{25}$$

Simplifying Eq. (25):

$$\Delta_{w_0} = \frac{1}{2}|(w+\Delta)^T(\mu_{X_0} + h + \mu_1) - w^T(\mu_{X_0} + \mu_1)| \tag{26}$$

$$\Delta_{w_0} = \frac{1}{2}|w^Th + \Delta^T(\mu_{X_0} + h + \mu_1)| = \frac{1}{2}|w^Th + \Delta^T(\mu_Z + \mu_1)| \tag{27}$$

Similar to Δ_w derivation, assuming $h \approx 0$ results in the following:

$$\Delta_{w_0} = \frac{1}{2}|\Delta^T(\mu_{X_0} + \mu_1)| \tag{28}$$

Accordingly, Eq. (23), and Eq. (28) represent the difference in LDA classifier resulting from the SMOTE generated patterns. It can be observed from Eq. (23) that Δ_w depends on T^{-1}, where T is the discrepancy between covariance matrices of SMOTE generated samples Σ_Z, and the covariance matrix of the original distribution Σ_{X_0}. From the theoretical analysis presented in Section 4.1, and the empirical results in Section 5, it could be concluded that if the difference between Σ_Z, and Σ_{X0}, is large, then because of the double inversion the value of Δ_w would be large (however the discrepancy is limited to some maximum value). The converse is true when the covariance matrices are close to each other.

Another factor of Δ_w is the overlap between the minority, and majority class regions expressed in terms of the distance of the minority class mean μ_{X_0}, and the majority class mean μ_1, denoted as b. For challenging problems of large overlap between classes' regions, b would be small, and hence, Δ_w would not be large.

We further discuss the impact of SMOTE on the LDA classification boundary in Section 5. We can observe from Fig. 15 that there is no substantial change of decision boundary before, and after applying SMOTE, which supports the theoretical findings discussed above in this section.

5. Experiments

5.1. Experiments for testing the distribution

To have a more in-depth understanding of the quality of SMOTE sampling and its influencing factors, we have performed a simulation study. In the first group of experiments, we generate artificial datasets from multivariate Gaussian distributions, apply SMOTE over-sampling, then estimate the distribution of the examples generated by SMOTE and compare it to the original distribution. The advantage of using artificial data is that in this case we know the ground truth distribution, and hence we can obtain an accurate estimate in the error of the generated distribution. In the second group of experiments, we consider real world data sets and explore the accuracy of SMOTE for these data sets.

For the first experiments on the artificial datasets, in order to have the analysis general enough, we consider 20 different distributions with different parameters. In all cases we consider the zero mean case, because the mean constitutes a shift in the center of operations and will therefore be inconsequential. However, we consider a variety of 20 different covariance matrices Σ_{X_0} varying between diagonal matrices and matrices containing off-diagonal terms.

We studied the effect of the following influencing parameters: the number of original minority examples N, the dimension d, and the K parameter of the K-nearest neighbor. We have separately varied each of the influencing factors, while fixing the others, and in each case we documented the accuracy in the distribution of the generated points. While varying each parameter, the others are set at their "default values", which are as follows: $N = 100$, $d = 10$, and $K = 5$. Additionally, in these experiments, we have set $w^* = 1$ as used in the standard SMOTE, since we are interested in analyzing original SMOTE method.

In order to estimate the mean and the covariance of the SMOTE generated patterns, we apply the following procedure:

- Repeat the following M times:
  - Randomly generate N patterns from the original distribution.
  - Repeat the following L times:
    - Apply SMOTE to the generated N original patterns.
    - Obtain the sample mean vector and the covariance matrix of the SMOTE generated examples.
- - Average the sample mean vector and covariance matrix over the L inner runs.
- Average the mean and the covariance matrix estimates over the M outer runs to get final estimates for the generated patterns' distribution.
- Compare the final estimated SMOTE-generated patterns' distribution to the original distribution using distribution distance metrics, described below.

In our experiments, we use $M = 1000$ and $L = 1000$. In addition to the simulation framework proposed above, we also apply the derived theoretical formula, equations 6 and (7) for the multivariate Gaussian case, and equations 9, and (10) for the multivariate Laplace case, in order to garner additional evidence concerning the behavior of SMOTE. All experiments essentially showed that the mean vector of the SMOTE generated points almost coincides with the true mean, producing almost zero error. This agrees with the theoretical analysis. Therefore, we focus our simulations on the covariance matrix, which indeed shows some discrepancy.

In order to measure how the covariance matrix of the SMOTE-generated patterns diverges from the original covariance matrix, we define the "Total Variances Difference" (TVD) measure. This measure helps us learn the amount and the polarity of the difference between the synthetic and the original covariance matrices. It is defined as the difference between the sum of component (or feature) variances for the two covariance matrices. It is normalized by the sum of component variances for the original covariance matrix, in order to have it as a relative measure. Since the sum of variances equals the trace of the covariance matrix, the TVD can be written as:

$$TVD(\Sigma_1, \Sigma_2) = \frac{\text{trace}(\Sigma_2) - \text{trace}(\Sigma_1)}{\text{trace}(\Sigma_1)}$$

Furthermore, in order to measure difference in the off-diagonal elements of the covariance matrix, we use the Frobenius norm of the difference matrix. Similar to TVD, we normalize it by the Frobenius norm of the original covariance matrix to be a relative measure.

$$Frob.Norm(\Sigma_1, \Sigma_2) = \frac{||\Sigma_2 - \Sigma_1||_F}{||\Sigma_1||_F}$$

where the Frobenius norm of a real matrix A is defined as:

$$||A||_F = \sqrt{\text{trace}(A^T A)}$$

In addition, we use the Kullback–Leibler Divergence (KL), which is an asymmetric distance measure between two probability distributions P and Q such that $D_{KL}(P||Q)$ measures the information lost when Q is used to approximate P [30]. It is defined as follows:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \ln\left(\frac{p(x)}{q(x)}\right) dx$$

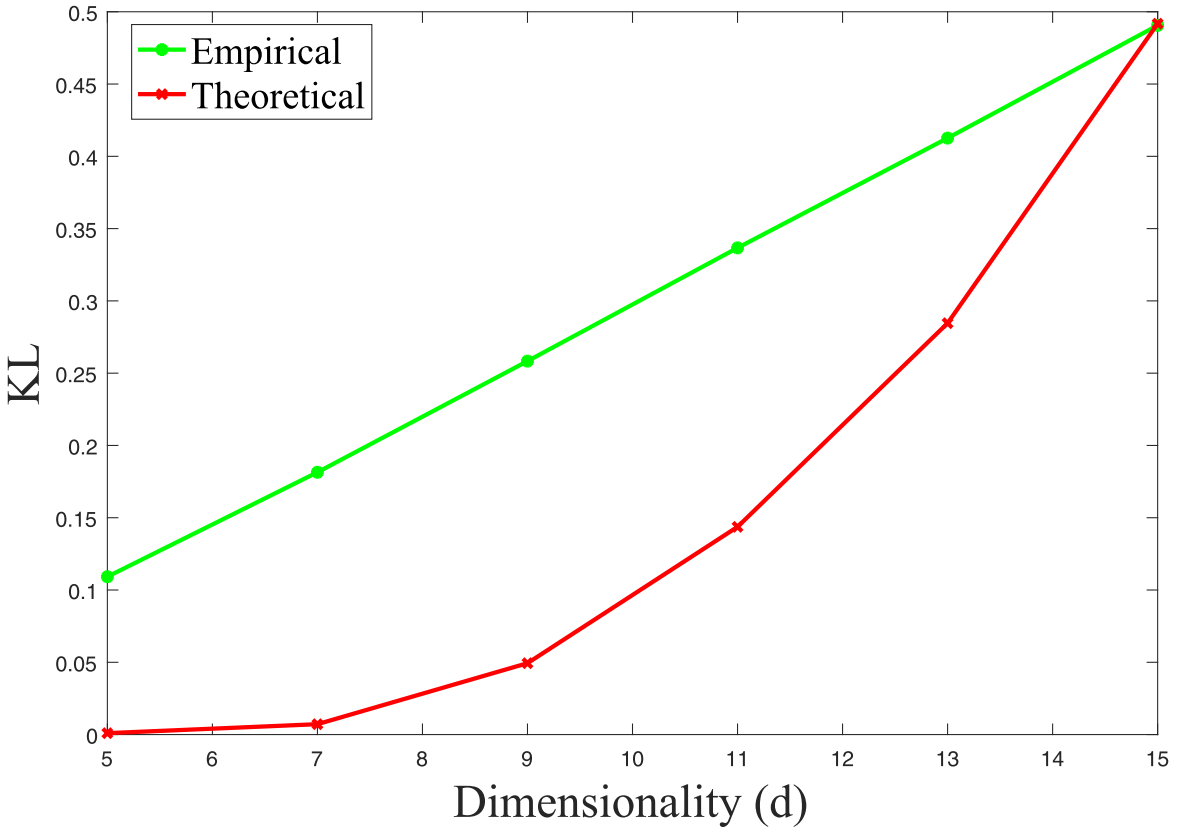

Fig. 2. KL for the empirical and the theoretical estimates (measuring the discrepancy between the distribution of SMOTE-generated examples and the true distribution), versus dimensionality d, using multivariate Gaussian distribution generated data.

where $p(x)$ and $q(x)$ are the probability density functions to be compared. For multivariate Gaussian distributions, $p(X) \sim N(\mu_1, \Sigma_1)$ and $q(X) \sim N(\mu_2, \Sigma_2)$, the KL-distance can be evaluated as follows:

$$D_{KL}(P||Q) = \frac{1}{2}\left[\text{trace}(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T {\Sigma_2}^{-1}(\mu_2 - \mu_1) - d + \ln\left(\frac{\det(\Sigma_2)}{\det(\Sigma_1)}\right)\right]$$

where d is the dimension. We note that while the metric TVD shows the polarity and amount of difference, the KL metric shows only the amount of difference and is always nonnegative.

Figs. 2–4 show the three distance metrics (between the distribution of the SMOTE-generated patterns and the true distribution), when exploring the effect of the dimension d. We show the distance metrics based on both the experimental study and the derived theoretical study (all applied to the same problems). As mentioned before, we fix all other factors at their default values, while varying the dimension. Similarly, Figs. 5–7 show the KL, the Frobenius norm and the TVD metrics respectively for the case of varying the number of minority samples N. Also, Figs. 8–10 show the KL, the Frobenius norm and the TVD metrics respectively for varying the K parameter of the KNN (that is used as part of the construction of the SMOTE method). One can observe that the covariance matrix of the SMOTE-generated patterns becomes less accurate as the number of minority patterns N decreases, and as the dimension d increases, and as K (of the KNN) increases.

We have also applied the same set of experiments using the Laplace distribution. The empirical distribution results for Laplace distribution are very similar to the represented results of the multivariate Gaussian distribution. For space limitations, we did not include the Laplacian distribution empirical results.

5.2. Distribution-related experiments using real data

In the other set of experiments, we have applied a similar set-up as discussed on three real world UCI datasets. This provides a test for realistic situations and for situations where the distribution is not necessarily Gaussian or Laplacian. The analysis using real datasets is undoubtedly important in order to justify that the findings apply to more complex situations, since real datasets could be noisy, and they could have sub-concepts for the minority class patterns.

We considered datasets that are originally large. This is in order to have an accurate estimate of the true mean and true covariance matrix. However, since SMOTE is used primarily for smaller datasets, we consider only a small subset (like 50 or

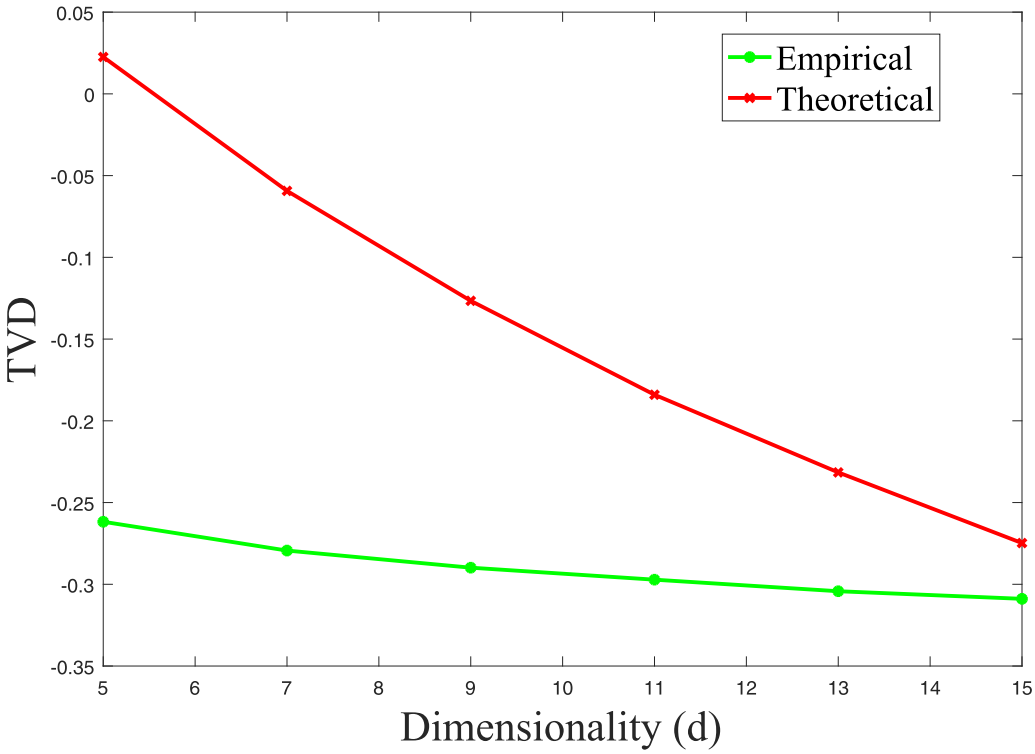

Fig. 3. TVD for the empirical and the theoretical estimates (measuring the discrepancy between the distribution of SMOTE-generated examples and the true distribution), versus dimensionality d, using Multivariate Gaussian distribution generated data.

Table 1
Real world Datasets Description.

Dataset	Number of minority class patterns	Dimensions
Adult	22,654	14
Client Default	23,364	23
Credit Card	284,315	28

100) of the data, and perform the sampling using these. For example, assume that the dataset has about 10,000 points. We compute the mean and covariance matrix from the 10,000 points and assume these to be approximately the true ones (due to the large number of points). Consider that we test the case of number of patterns $N = 100$. In such a situation we select 100 patterns randomly from the 10,000 original data points. We perform the SMOTE generation experiments on these 100 selected points. Then, we repeat with a different selection of the $N = 100$ data points M times, thus implementing the outer loop of the simulation experiment along the lines discussed above for the artificial data sets.

Regarding the experiment where we vary the dimension (d), for real datasets, we choose the most informative features that have the highest principal components. Upon using this criterion, small dimensions will be subsumed by higher dimensions which establishes a fair comparison of performance among dimensions. In other words, when we discard features to test lower dimensionality, we discard the least informative ones. This is by virtue of their order according to highest to lowest principal components. This will minimize the confounding effect of a varying information content of the tested feature sets. Table 1 shows the sizes and the dimensions of the considered datasets. Adult and Default datasets are UCI datasets [12] and the third dataset, credit card, is a Kaggle dataset developed by [10]. Fig. 11 shows the empirical estimates of the Total Variance Difference (TVD) metric for varying the dimension d. Fig. 12 shows the empirical estimates of the Total Variance Difference (TVD) metric for varying the number of patterns N. In addition, Fig. 13 demonstrates the empirical estimates of the Total Variance Difference (TVD) metric for varying the K parameter of KNN in SMOTE. Only empirically estimated TVD has been computed. The theoretical estimates as in Eq. (4) are hard to compute because the underlying density function $p(X_0)$ is unknown and probability densities are very hard to estimate with a reasonable error, especially for high dimensions, even in case of large data sets.

5.3. Classification related experiments

The previous experiments analyzed the distributional discrepancy of the SMOTE generated patterns. In these sets of experiments we explore how SMOTE impacts the classification performance and classification boundaries. We consider prob-

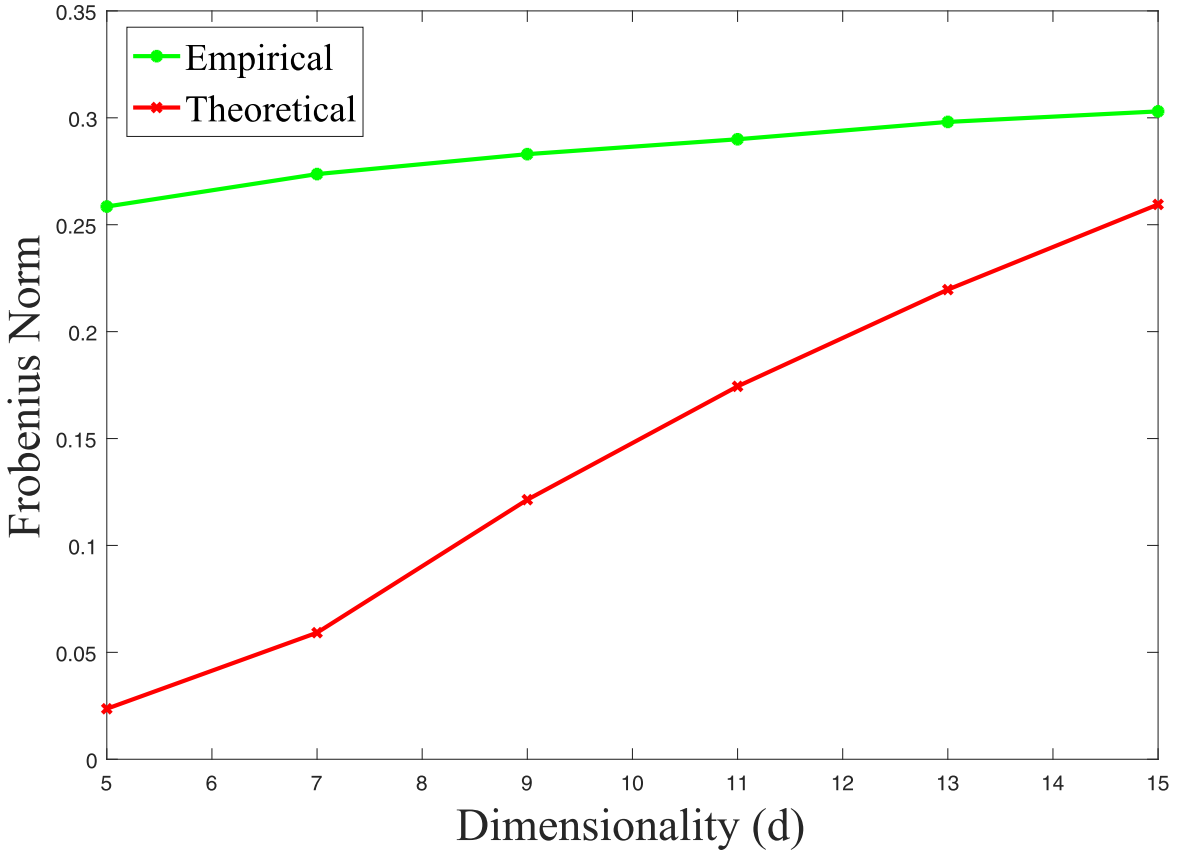

Fig. 4. Frobenius Norm for the empirical and the theoretical estimates (measuring the discrepancy between the distribution of SMOTE-generated examples and the true distribution), versus dimensionality d, using multivariate Gaussian distribution generated data.

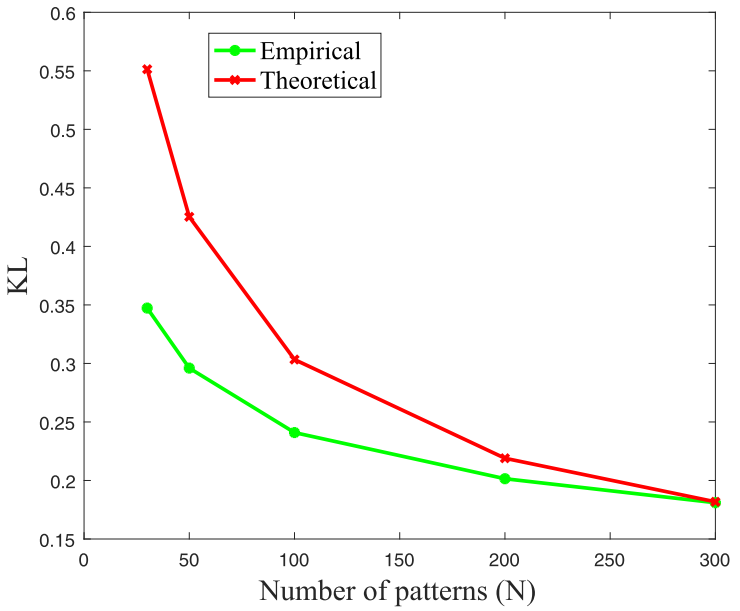

Fig. 5. KL for the empirical and the theoretical estimates (measuring the discrepancy between the distribution of SMOTE-generated examples and the true distribution), versus number of patterns N, using multivariate Gaussian distribution generated data.

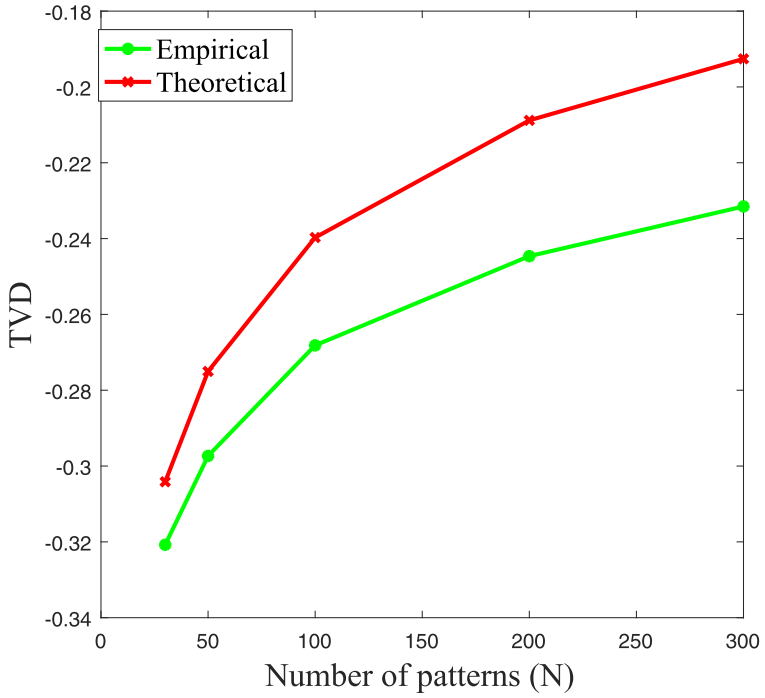

Fig. 6. TVD for the empirical and the theoretical estimates (measuring the discrepancy between the distribution of SMOTE-generated examples and the true distribution), versus number of patterns *N*, using multivariate Gaussian distribution generated data.

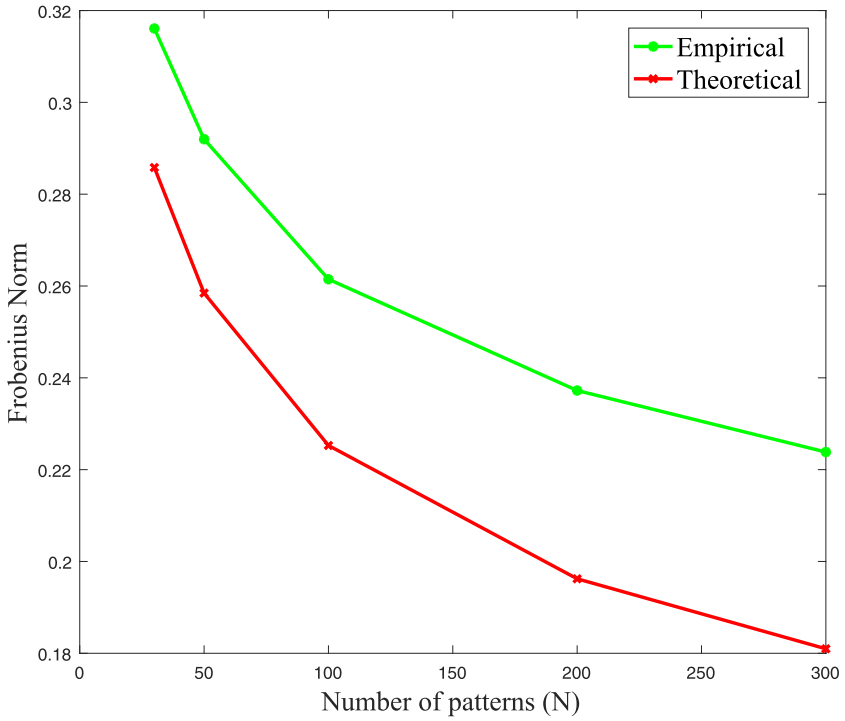

Fig. 7. Frobenius Norm for the empirical and the theoretical estimates (measuring the discrepancy between the distribution of SMOTE-generated examples and the true distribution), versus number of patterns *N*, using multivariate Gaussian distribution generated data.

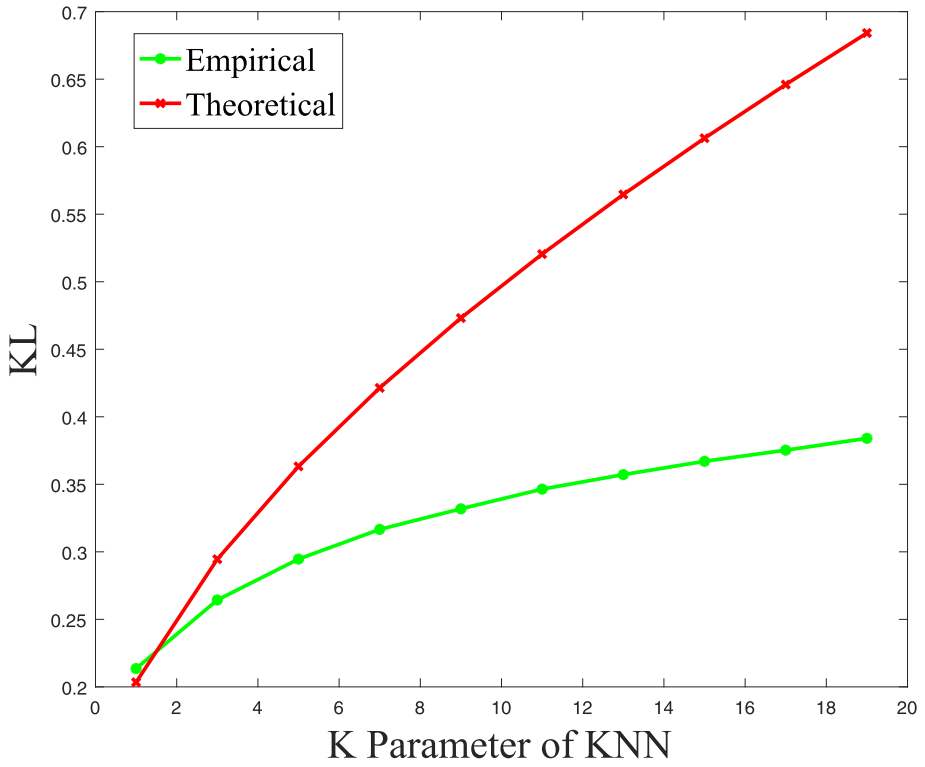

Fig. 8. KL for the empirical and the theoretical estimates (measuring the discrepancy between the distribution of SMOTE-generated examples and the true distribution), versus K parameter of KNN in SMOTE, using multivariate Gaussian distribution generated data.

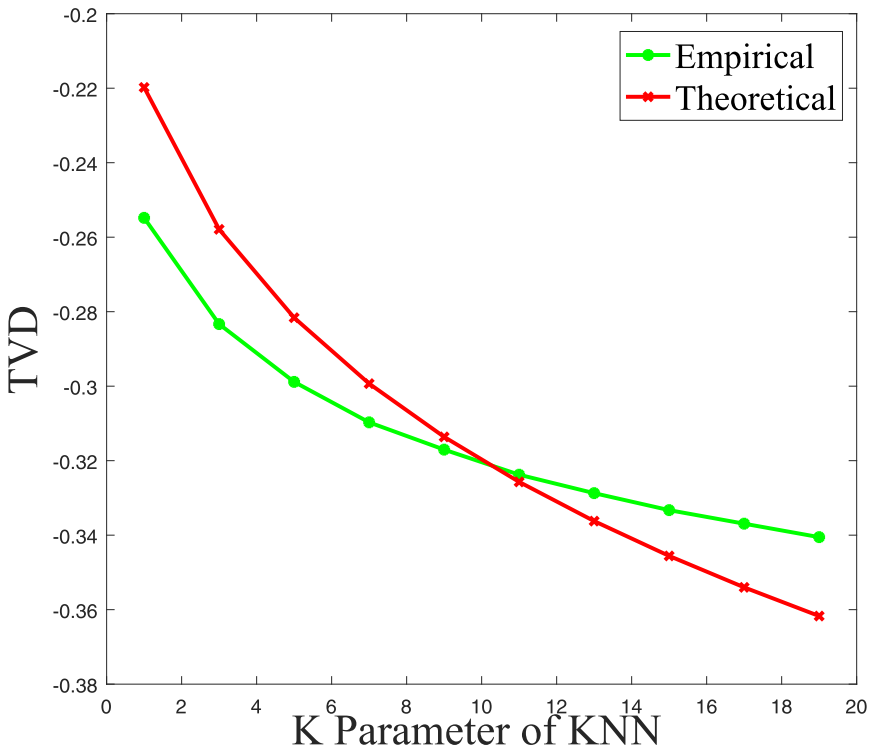

Fig. 9. TVD for the empirical and the theoretical estimates (measuring the discrepancy between the distribution of SMOTE-generated examples and the true distribution), versus K parameter of KNN in SMOTE, using multivariate Gaussian distribution generated data.

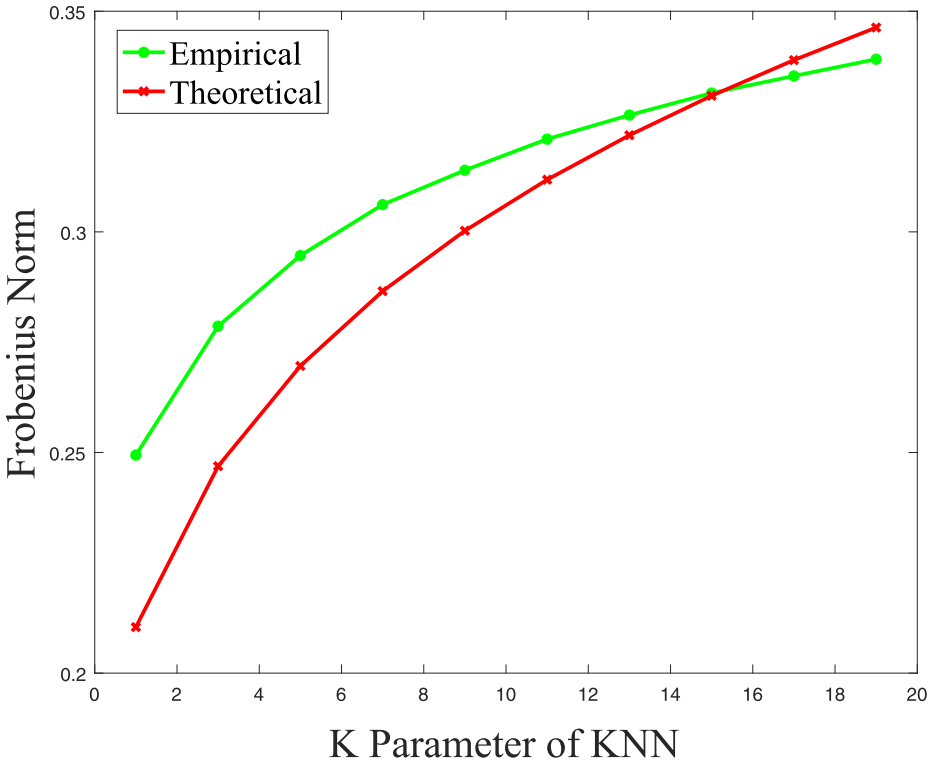

Fig. 10. Frobenius Norm for the empirical and the theoretical estimates (measuring the discrepancy between the distribution of SMOTE-generated examples and the true distribution), versus K parameter of KNN in SMOTE, using multivariate Gaussian distribution generated data.

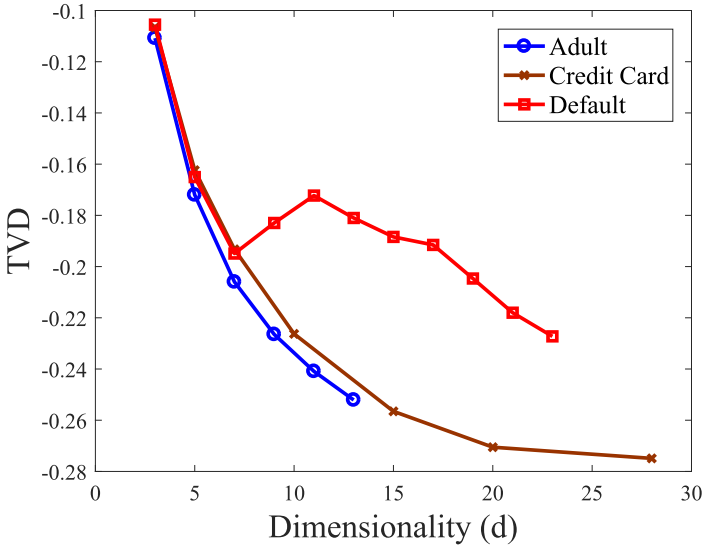

Fig. 11. TVD for SMOTE versus dimensionality d for the three real datasets.

lems with considerable unbalance, but where the minority class holds equal importance to the majority class, and cannot be "swamped" by a majority-dominated classifier. For this purpose, we use the Gmean performance metric, which treats both classes on equal footing:

$$Gmean = \sqrt{Acc_{min} \times Acc_{maj}} \tag{29}$$

where $Acc_{\min}$ and Acc_{maj} are the accuracies of minority class and majority class respectively.

The following experiment shows the impact of SMOTE on the classification boundary. Fig. 14 shows decision boundaries for the SVM classifier for an unbalanced artificial dataset for three cases. In the first case the classifier uses all the training

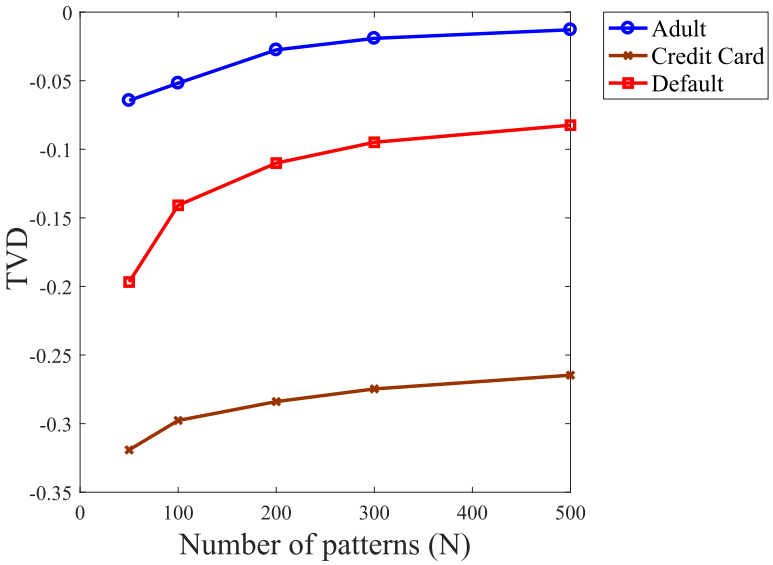

Fig. 12. TVD for SMOTE versus number of patterns *N* for the three real datasets.

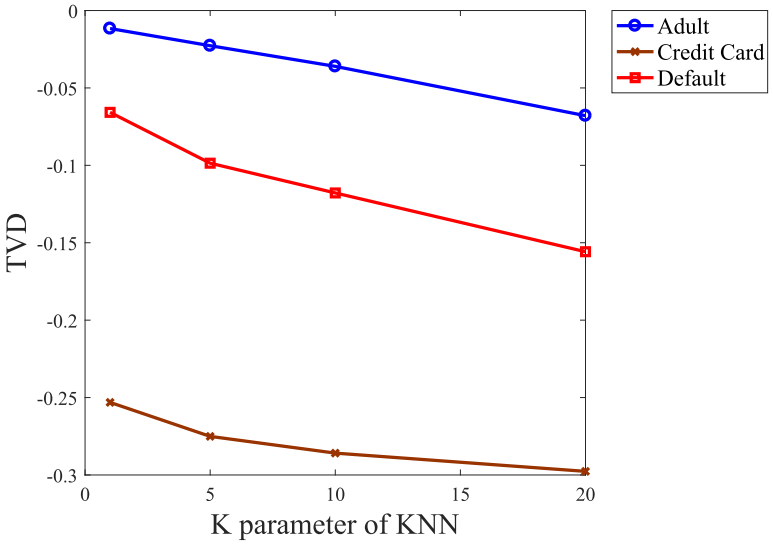

Fig. 13. TVD for SMOTE versus *K* parameter of KNN in SMOTE for the three real datasets.

patterns with no regard to boosting the minority samples presence (the "before-SMOTE" classifier in the figure). Obviously, this is an insensible classifier, and leads to the majority class region pushing back on the minority class region. In the second classifier, the minority set is boosted by adding SMOTE generated samples until they are equal in number to those of the majority (the "after-SMOTE" classifier in the figure). The third classifier, is the ideal case, where we assume that we know the true distribution of the minority class, generate from it samples so that both classes have equal size data (equal to the size of the majority's), and then design the classifier (the "balanced-boundary" classifier in the figure). Fig. 14 shows the classification boundaries of the three classifiers. As can be seen, the "after-SMOTE" classifier provides an improvement over the "before-SMOTE" classifier. As we mentioned in Section 1, standard classifiers aim to maximize overall accuracy, which would hurt classification performance of minority class. For example, for SVM classifier, the number of support vectors of majority class is greater than that of minority class which causes bias towards majority class as shown in Fig. 14. To remedy this issue some of the researchers suggested the following solutions. The authors of [1] propose using different penalty parameters *C* for minority and majority class examples in the objective function of support vector machines. Also, another method named z-SVM is proposed in [25], boosts the emphasis of minority class support vectors by multiplying their coefficients by a factor z to decrease SVM's bias towards majority class. Chang et. al [44] use kernel transformation to

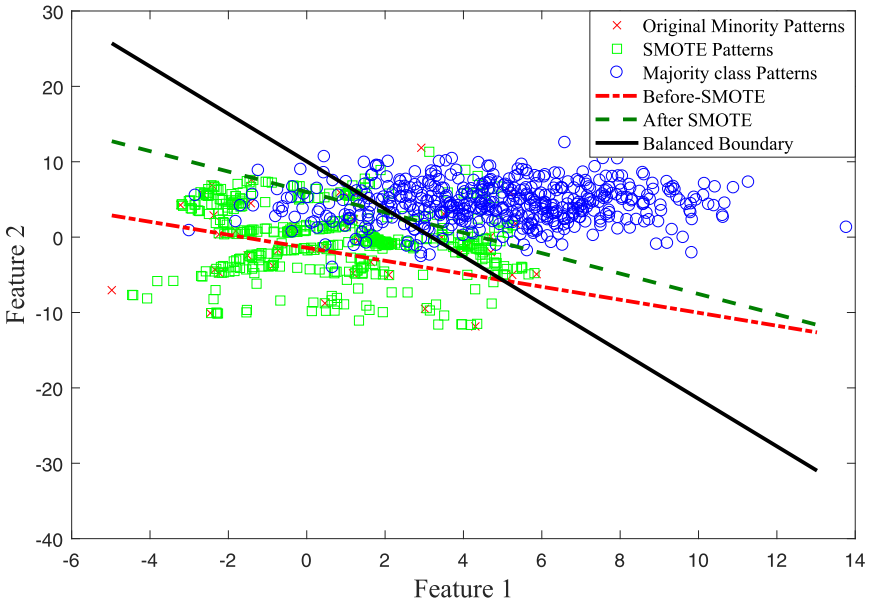

Fig. 14. SVM decision boundaries before SMOTE, after SMOTE, and using a balanced class distribution.

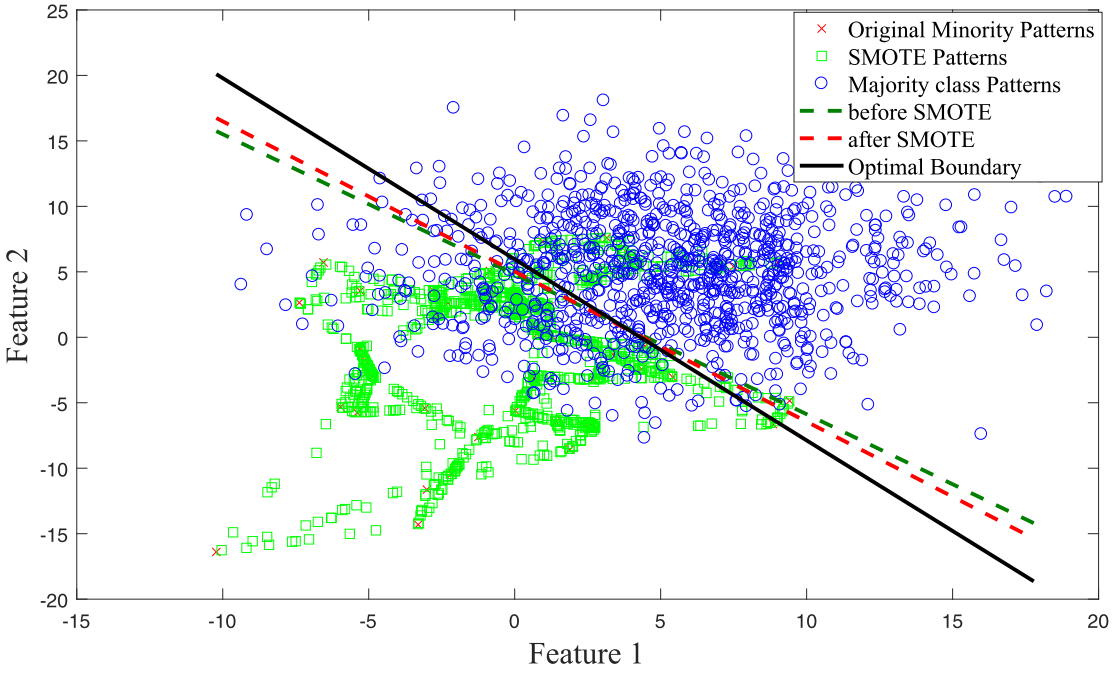

Fig. 15. Decision boundaries using LDA classifier: before SMOTE, after SMOTE, and the optimal Bayes decision boundary.

handle class imbalance by modifying the kernel matrix according in order to widen the classification region around minority class.

In the next experiment, we consider the Fisher linear discriminant classifier which has been analyzed in a previous section. We build upon the derived formulas by experimentally exploring the effect of SMOTE on the classification regions/boundaries. Fig. 15 shows a 2-D example for a dataset generated using multivariate Gaussian distribution. The figure shows the optimal Bayes decision boundary using the true distribution parameters and the LDA decision boundary before and after applying SMOTE. It can be observed from the figure that there is not much difference in decision boundaries before and after applying SMOTE. Therefore, applying SMOTE does not yield a significant improvement for the linear discriminant analysis classifier. The previous two experiments show that SMOTE can impact different classifiers differently. While in SVM SMOTE had a positive impact on the classifier, for LDA it made little difference. This is because of the construction of LDA,

Table 2
Gmean for KNN classifier using different number of minority class patterns N.

Method	Gmean (N=50)	Gmean (N=100)
Before Applying SMOTE	60.80 %	68.47%
After Applying SMOTE	71.01 %	75.35%
Using Balanced Distribution	87.01%	86.30%

Table 3
Gmean for KNN classifier using different dimensionality d.

Method	Gmean (d=10)	Gmean (d=20)
Before Applying SMOTE	45.29%	39.21%
After Applying SMOTE	62.89%	63.21%
Using Balanced Distribution	82.25%	75.55 %

Table 4
Gmean for KNN classifier using different values of K parameter used in SMOTE generation.

Method	Gmean (K=5)	Gmean (K=10)
Before Applying SMOTE	59.04%	60.47%
After Applying SMOTE	69.44%	74.90%
Using Balanced Distribution	86.43%	86.90%

Table 5
Gmean for SVM classifier using different number of minority class patterns N.

Method	Gmean (N=50)	Gmean (N=100)
Before Applying SMOTE	56.74%	70.06%
After Applying SMOTE	76.36%	82.48%
Using Balanced Distribution	89.60%	89.33%

which relies mainly on mean and covariance matrix estimates. The covariance matrix estimates using the original samples or SMOTE generated samples would probably be close. The reason is that the sample covariance estimate (using the original samples) is known from statistics theory to be optimal, in the maximum likelihood sense. So, SMOTE generated samples stand little chance to improve the estimate. The authors of [35] empirically show that SMOTE is not beneficial for linear discriminant analysis classifier, and this agrees with our observation too.

5.4. *Effect of* N, d, *and* K *on classifier performance*

In the next set of experiments we consider two different classifiers in our experiments: support vector machines (SVM) with Radial Basis Function kernel (RBF), and K nearest neighbor classifier (KNN). The purpose is to explore the effect of varying the number of minority samples N, the dimension d, and the K used for KNN sample generation in SMOTE on classifier performance. In each experiment, we generate an unbalanced artificial dataset generated from multivariate Gaussian distribution. Then, we generate samples using SMOTE from this unbalanced set, and design the classifier. The performance of the classifier is then tested using a large balanced test pattern set $N = 10,000$ (in order to obtain an accurate performance measurement). Again, we use the gmean metric to test the performance.

The default number of minority class patterns is set to be $N = 50$ (fixed at that when varying other factors). Also the default dimension and K are respectively $d = 7$, and $K = 5$. For all runs we use the number of majority class patterns to be 1000. We repeat each experiment for 200 runs, each run with a different generated unbalanced dataset, and then we average the results. The classifiers' key parameters (the K in case of KNN classifier, and the C and γ in case of the SVM classifier) are tuned using five fold cross validation. The benchmark classifier with which we compare against is a balanced problem with minority class patterns equal in number to the 1000 majority class patterns. The reason for using artificial data is that in only this case we know the ground truth of the distributions, can repeat the experiment in a Monte Carlo fashion, and can measure classifier performance accurately.

Tables 2–6 show the results as to how does SMOTE affect classification performance, as we vary the major factors N, d, and K. One can observe that applying SMOTE does improve classification performance, due to the extra boosting of minority patterns. However, it can be observed that even after applying SMOTE, there is a significant difference between that and the case when we use original minority patterns of equal number as the majority ones. It can be observed from Table 5 that increasing N, i.e. having more original minority class patterns, leads to better classification accuracy, which conforms with

Table 6
Gmean for SVM classifier using different dimensionality d.

Method	Gmean (d=10)	Gmean (d=20)
Before Applying SMOTE	62.94%	31.10%
After Applying SMOTE	70.85%	38.24%
Using Balanced Distribution	95.32%	92.27%

Table 7
Gmean for SVM classifier using different values of K parameter used in SMOTE generation.

Method	Gmean (K=5)	Gmean (K=10)
Before Applying SMOTE	56.21 %	57.45%
After Applying SMOTE	75.20%	76.79%
Using Balanced Distribution	89.09%	89.83%

Table 8
Real world Datasets Description used for Classification and Distribution Analysis.

Dataset	Number of minority class patterns	Number of majority class patterns	Dimensions
SUSY	2,287,827	2,712,173	18
Hepmass	523,733	524,842	27
EyeState	6723	8257	14
ElecGrid	3620	6380	12
NewsPopularity	18,490	21,154	58

our findings in the distribution analysis section that increasing the number of patterns enhances the proximity of SMOTE generated patterns to the original distribution. Similarly, Table 6 shows that increasing the dimension reduces classification performance (especially for the SVM classifier), which conforms with our the distribution analysis results. There is a natural deterioration of performance due to decreasing N and increasing d, due to respectively small sample behavior and the curse of dimensionality. But this goes beyond that, and adds an extra underperformance, due to deteriorating SMOTE sampling accuracy.

As of the effect of K, one can see that it has little influence on the performance. This is in contrast with the distribution-based results, which favor small K. The reason is that there is a sample dependence effect in the case of small K that counterbalances its aforementioned advantage. This is explained in detail in the fifth item of Section 6.

5.5. Comprehensive analysis of distribution and classification on real datasets

In order to have a better understanding of the relation between classification, and distribution performance, we perform an additional set of experiments on five real-world UCI datasets [12]. The details of these datasets are described in Table 8. We did not use the datasets described in Table 1 in this experiment because we needed larger datasets. To investigate whether the obtained analysis applies to other variants of SMOTE, we extend our analysis to include popular SMOTE extensions: Borderline SMOTE with its two versions [22], and Adasyn [23].

Through this set of experiments, we perform a dual empirical analysis of classification, and distribution performance on real datasets using SMOTE, and its three variants. For the distribution analysis, we use the same methodology, and parameters described in Section 5.2. However, since some oversampling methods rely on majority class patterns, we estimate the majority class distribution, and we set the number of majority class patterns in the Monte Carlo simulation procedure described in Section 5.1 to 1000. The default number of minority class samples N is set to 50, the default value for K parameter is 5, and the default value for the dimension d is the original dimensionality of each considered dataset. We use the same default values for the classification experiment as well. By default we mean the basic value that is to be fixed when varying one of the other factors.

In our classification analysis, we compare several methods: the unbalanced distribution case, SMOTE, and its considered extensions, and the fully balanced case. For the latter we have a large and equal number of samples per class (20,000 samples per class, or however many originally exists in the smallest size class if it is less than 20,000). We evaluate the fully-balanced classification performance in order to have a baseline for judging the classification performance of the different over-sampling methods. This is the gold standard benchmark towards which the other sampling methods vie to have similar or close performance.

We follow the same methodology as presented in Section 5.4. In the experiments, we consider equal sized training, and testing sets. Then, we consider the K-nearest neighbor classifier (KNN), and the support vector classifier (SVM), with RBF kernel, and apply them to the training set. Again, we tune the parameters using 5-fold cross validation. Finally, we evaluate the classification performance by applying the trained model to the test set, and measuring the gmean metric.

Table 9
TVD for SMOTE, Borderline SMOTE1, Borderline SMOTE2, and Adasyn, on real datasets, using different values of number of minority class patterns N (N/A means the method did not generate enough data).

Dataset	N value	SMOTE	Bordeline-SMOTE 1	Bordeline-SMOTE 2	Adasyn
SUSY	20	-0.3658	-0.3394	-0.3551	-0.3466
SUSY	50	-0.2990	-0.2227	-0.2536	-0.3350
SUSY	100	-0.2568	-0.1989	-0.2413	-0.3317
EyeState	20	-0.3850	N/A	N/A	-0.2901
EyeState	50	-0.2626	-0.8408	-0.8408	-0.4185
EyeState	100	-0.1511	N/A	N/A	-0.4705
Hepmass	20	-0.3830	-0.4062	-0.4183	-0.3447
Hepmass	50	-0.3212	-0.3395	-0.3385	-0.3170
Hepmass	100	-0.3128	-0.2929	-0.2972	-0.3220
ElecGrid	20	-0.4026	-0.5632	-0.5672	-0.2419
ElecGrid	50	-0.2179	-0.1833	-0.0709	-0.1345
ElecGrid	100	-0.1887	-0.1584	-0.1034	-0.1464
NewsPopularity	20	-0.3156	-0.6237	-0.6044	-0.0855
NewsPopularity	50	-0.1681	-0.3347	-0.2764	-0.0362
NewsPopularity	100	-0.1159	-0.2150	-0.1644	-0.0272

Table 10
TVD for SMOTE, Borderline SMOTE1, Borderline SMOTE2, and Adasyn, on real datasets, using different values of K parameter of synthetic data generation (N/A means the method did not generate enough data).

Dataset	K value	SMOTE	Bordeline-SMOTE 1	Bordeline-SMOTE 2	Adasyn
SUSY	5	-0.3003	-0.2268	-0.2584	-0.3330
SUSY	10	-0.3167	-0.2321	-0.2516	-0.3403
SUSY	20	-0.3430	-0.2931	-0.2985	-0.3573
EyeState	5	-0.2310	N/A	N/A	-0.4063
EyeState	10	-0.2781	N/A	N/A	-0.3481
EyeState	20	-0.3170	N/A	N/A	-0.2037
Hepmass	5	-0.3227	-0.3548	-0.3634	-0.3257
Hepmass	10	-0.3388	-0.3325	-0.3180	-0.3273
Hepmass	20	-0.3329	-0.3152	-0.3094	-0.3302
ElecGrid	5	-0.2649	-0.2677	-0.1881	-0.1687
ElecGrid	10	-0.3059	-0.2757	-0.1540	-0.1879
ElecGrid	20	-0.3472	-0.3297	-0.1892	-0.2160
NewsPopularity	5	-0.1701	-0.3162	-0.2510	-0.0336
NewsPopularity	10	-0.2335	-0.3296	-0.2110	-0.0594
NewsPopularity	20	-0.3370	-0.4081	-0.2249	-0.0956

In both of the distribution, and classification analyses for SMOTE and its variants, we study different values for each of the three main factors: the number of minority class patterns N, the dimension d, and the number of K neighbors used in oversampling generation of SMOTE. The objective of this study, is to analyze the impact of varying these parameters on real datasets with respect to distribution, and classification, since real datasets often have different complexities than artificial datasets such as: sub-concepts, overlapping classes, noisy patterns, and outliers. For this presented set of experiments, we evaluate the distribution behavior exactly as described in Section 5.2. We have chosen very large datasets so that we can accurately evaluate the true distribution parameters, and in order to evaluate the classification performance in case of fully balanced distribution. Tables 9–11 show the total variances difference (TVD) for the experiments that analyze the distribution accuracy of SMOTE and its variants for the three factors N, K, and d, respectively. Tables 12–14 show the classification performance for the case of using the KNN classifier. In other words, they show gmean classification accuracy measure of SMOTE and its variants for respectively the three factors N, K, and d. Also, Tables 15–17 show the classification performance (the gmean) for the case of using the SVM classifier for SMOTE and its variants for respectively the three factors N, K, and d.

By analyzing SMOTE performance, we observe from Tables 12–17 that the SMOTE over-sampling method effectively promotes the classification performance over the unbalanced distribution. This is because it restores the influence of the minority samples by boosting their numbers. However, SMOTE performance is still far from the true balanced performance as can be observed from the aforementioned tables.

For SMOTE extensions, we find from Tables 12–14 that they are promising over-sampling approaches in terms of classification performance. For the distribution case there is no pioneer method having best distribution accuracy. This seems logical, since all of the popular SMOTE extensions mainly focus on promoting the classification performance, so for example the two versions of Borderline SMOTE, and Adasyn essentially favor hard-to-classify minority class patterns that are close to decision boundary. Therefore, the resulting distribution may deviate from the original class distribution.

For the classification results, we can observe from Table 12, and Table 15 that as the number of patterns N increases, the classification performance in terms of gmean values improves, and this holds for SMOTE, and its extensions. For the

Table 11
TVD for SMOTE, Borderline SMOTE1, Borderline SMOTE2, and Adasyn, on real datasets, using different values of dimensionality *d*.

Dataset	*d* value	SMOTE	Bordeline-SMOTE 1	Bordeline-SMOTE 2	Adasyn
SUSY	5	-0.2645	-0.2675	-0.2133	-0.1760
SUSY	10	-0.2727	-0.3249	-0.3581	-0.3894
SUSY	18	-0.2887	-0.3199	-0.3736	-0.4008
EyeState	5	-0.1767	-0.5430	-0.5649	-0.5906
EyeState	10	-0.2029	-0.4937	-0.5479	-0.5632
EyeState	14	-0.2259	-0.4356	-0.5046	-0.5245
Hepmass	5	-0.2551	-0.2909	-0.2138	-0.1450
Hepmass	10	-0.2983	-0.2833	-0.2463	-0.2512
Hepmass	20	-0.3154	-0.3203	-0.3043	-0.2938
Hepmass	27	-0.3272	-0.3160	-0.3144	-0.3265
ElecGrid	5	-0.2748	-0.3517	-0.2602	-0.1421
ElecGrid	10	-0.3071	-0.3039	-0.2608	-0.2497
ElecGrid	12	-0.3192	-0.3223	-0.2662	-0.2553
NewsPopularity	5	-0.2500	-0.1564	-0.1078	-0.2055
NewsPopularity	10	-0.2973	-0.2701	-0.2634	-0.3054
NewsPopularity	20	-0.3026	-0.2853	-0.2949	-0.3327
NewsPopularity	58	-0.3337	-0.3401	-0.3745	-0.3870

Table 12
Gmean for SMOTE, Borderline SMOTE1, Borderline SMOTE2, and Adasyn, on real datasets, using KNN classifier, and using different values of number of minority class patterns *N*.

Dataset	*N* value	Unbalanced	SMOTE	Bordeline-SMOTE 1	Bordeline-SMOTE 2	Adasyn	Balanced
SUSY	20	32.77%	46.40%	36.20%	37.00%	51.55%	75.45%
SUSY	50	45.17%	55.53%	50.70%	56.28%	63.71%	75.45%
SUSY	100	52.29%	60.34%	59.05%	65.19%	66.94%	75.45%
EyeState	20	34.17%	52.82%	41.67%	36.93%	51.81%	94.05%
EyeState	50	48.92%	65.30%	60.50%	58.37%	68.85%	94.03%
EyeState	100	61.20%	72.95%	71.41%	70.23%	75.47%	94.05%
Hepmass	20	34.73%	59.71%	48.32%	51.50%	67.32%	90.14%
Hepmass	50	51.51%	70.25%	66.34%	74.21%	79.51%	90.13%
Hepmass	100	62.29%	75.61%	74.23%	82.43%	82.09%	90.14%
ElecGrid	20	25.48%	47.71%	35.48%	32.98%	50.99%	86.53%
ElecGrid	50	37.92%	58.70%	52.81%	53.53%	64.85%	86.50%
ElecGrid	100	49.69%	66.08%	63.76%	68.47%	72.57%	86.57%
NewsPopularity	20	17.76%	32.04%	21.39%	22.19%	41.60%	63.42%
NewsPopularity	50	26.41%	42.71%	38.10%	40.79%	49.47%	63.43%
NewsPopularity	100	36.55%	48.92%	47.05%	49.73%	53.02%	63.43%

Table 13
Gmean for SMOTE, Borderline SMOTE1, Borderline SMOTE2, and Adasyn, on real datasets, using KNN classifier, and using different values of *K* parameter of synthetic data generation.

Dataset	*K* value	Unbalanced	SMOTE	Bordeline-SMOTE 1	Bordeline-SMOTE 2	Adasyn	Balanced
SUSY	5	43.87%	47.67%	44.48%	44.13%	50.61%	75.43%
SUSY	10	44.88%	55.69%	50.96%	56.98%	63.78%	75.45%
SUSY	20	44.61%	58.59%	56.51%	64.44%	65.97%	75.44%
EyeState	5	47.83%	55.40%	48.20%	47.84%	55.36%	94.03%
EyeState	10	48.94%	65.34%	60.83%	58.11%	68.14%	94.05%
EyeState	20	47.99%	67.87%	65.16%	63.27%	69.55%	94.05%
Hepmass	5	51.84%	60.94%	51.46%	51.87%	64.42%	90.14%
Hepmass	10	52.26%	70.20%	66.32%	75.76%	79.97%	90.13%
Hepmass	20	53.05%	73.39%	72.15%	82.51%	81.79%	90.13%
ElecGrid	5	37.10%	49.43%	36.61%	37.90%	49.59%	86.58%
ElecGrid	10	37.32%	58.49%	53.04%	53.56%	64.05%	86.54%
ElecGrid	20	38.77%	62.31%	59.87%	66.01%	70.99%	86.49%
NewsPopularity	5	27.06%	37.18%	26.06%	27.37%	40.92%	63.44%
NewsPopularity	10	26.81%	42.00%	36.96%	40.05%	49.01%	63.44%
NewsPopularity	20	27.28%	43.96%	41.85%	47.34%	52.68%	63.43%

distribution results, Table 9 demonstrates that as number of minority patterns *N* increases, the divergence of the SMOTE-type generated patterns from the original distribution diminishes. Thus, both of classification, and distribution empirical results agree regarding the impact of the number of minority class patterns, which supports the provided analysis in the previous sections.

Table 14
Gmean for SMOTE, Borderline SMOTE1, Borderline SMOTE2, and Adasyn, on real datasets, using KNN classifier, and using different values dimensionality d.

Dataset	d value	Unbalanced	SMOTE	Bordeline-SMOTE 1	Bordeline-SMOTE 2	Adasyn	Balanced
SUSY	5	35.82%	51.14%	46.04%	49.32%	57.89%	71.11%
SUSY	10	44.05%	56.11%	51.70%	56.35%	63.62%	76.10%
SUSY	18	45.28%	55.28%	50.77%	56.99%	63.69%	75.45%
EyeState	5	32.63%	49.90%	43.17%	40.35%	52.89%	66.76%
EyeState	10	44.87%	61.16%	56.29%	54.07%	64.38%	84.03%
EyeState	14	48.13%	64.36%	59.66%	57.53%	67.49%	94.05%
Hepmass	5	47.69%	63.56%	59.08%	66.99%	74.82%	87.04%
Hepmass	10	43.42%	56.92%	51.72%	60.40%	70.52%	85.96%
Hepmass	20	45.93%	62.32%	57.34%	66.45%	73.99%	87.20%
Hepmass	27	51.80%	70.23%	66.34%	73.84%	79.79%	90.14%
ElecGrid	5	33.37%	47.20%	42.09%	46.89%	54.94%	65.81%
ElecGrid	10	35.81%	54.80%	48.80%	49.12%	59.55%	76.38%
ElecGrid	12	38.22%	58.97%	53.27%	53.64%	65.38%	86.51%
NewsPopularity	5	49.42%	50.18%	40.94%	37.72%	50.17%	40.28%
NewsPopularity	10	21.87%	36.58%	28.13%	26.71%	41.40%	52.59%
NewsPopularity	20	24.22%	39.95%	32.61%	35.40%	47.31%	59.23%
NewsPopularity	58	26.48%	42.13%	36.87%	39.61%	49.00%	63.41%

Table 15
Gmean for SMOTE, Borderline SMOTE1, Borderline SMOTE2, and Adasyn, on real datasets, using SVM classifier, and using different values of number of minority class patterns N.

Dataset	N value	Unbalanced	SMOTE	Bordeline-SMOTE 1	Bordeline-SMOTE 2	Adasyn	Balanced
SUSY	20	14.66%	31.14%	25.07%	30.87%	34.00%	79.21%
SUSY	50	20.98%	40.35%	39.30%	48.51%	54.42%	79.30%
SUSY	100	36.34%	47.93%	45.80%	56.86%	51.84%	79.21%
EyeState	20	5.17%	45.84%	48.42%	49.42%	49.90%	58.45%
EyeState	50	8.42%	34.88%	55.70%	51.44%	49.11%	58.45%
EyeState	100	12.44%	39.39%	28.02%	51.96%	51.59%	58.45%
Hepmass	20	10.86%	19.75%	18.93%	28.32%	58.05%	91.43%
Hepmass	50	19.81%	17.02%	33.70%	66.82%	70.88%	91.28%
Hepmass	100	30.06%	40.45%	38.54%	83.23%	81.24%	91.39%
ElecGrid	20	24.31%	42.05%	50.74%	46.12%	49.98%	95.25%
ElecGrid	50	37.90%	65.87%	66.12%	62.95%	71.49%	95.19%
ElecGrid	100	52.87%	78.76%	76.34%	77.53%	77.88%	95.28%
NewsPopularity	20	8.99%	6.24%	4.88%	4.49 %	14.89%	66.49%
NewsPopularity	50	20.57%	16.34%	11.70%	12.71 %	54.54%	66.41%
NewsPopularity	100	29.20%	27.01%	31.72%	50.20%	57.74%	66.41%

Table 16
Gmean for SMOTE, Borderline SMOTE1, Borderline SMOTE2, and Adasyn, on real datasets, using SVM classifier, and using different values of K parameter of synthetic data generation.

Dataset	K value	Unbalanced	SMOTE	Bordeline-SMOTE 1	Bordeline-SMOTE 2	Adasyn	Balanced
SUSY	5	30.83%	42.80%	41.92%	47.50%	45.47%	79.37%
SUSY	10	24.69%	41.04%	41.83%	62.16%	57.15%	79.31%
SUSY	20	27.36%	44.33%	41.38%	56.16%	61.10%	79.29%
EyeState	5	6.45%	22.02%	12.55%	45.00%	45.38%	58.45%
EyeState	10	0.00%	53.73%	49.87%	50.43%	51.28%	58.45%
EyeState	20	11.54%	54.63%	52.84%	56.09%	53.76%	58.45%
Hepmass	5	22.26%	23.94%	20.99%	44.35%	83.14%	91.31%
Hepmass	10	20.13%	28.30%	27.16%	80.15%	75.65%	91.40%
Hepmass	20	17.14%	26.79%	26.19%	73.73%	66.03%	91.28%
ElecGrid	5	39.82%	59.80%	39.40%	39.82%	58.60%	95.33%
ElecGrid	10	38.12%	70.10%	73.73%	60.69%	66.76%	95.33%
ElecGrid	20	38.75%	62.66%	62.40%	63.72%	75.25%	95.33%
NewsPopularity	5	13.99%	15.76%	8.37%	15.05%	54.64%	66.57%
NewsPopularity	10	14.62%	16.83%	14.75%	22.45%	54.07%	66.21%
NewsPopularity	20	13.98%	22.91%	21.75%	56.33%	56.06%	66.21%

Like for the case of artificial datasets classification performance, the results presented in Table 4, and Table 7, regarding the impact of K on classification performance are similar for the real datasets' experiments. So expanding the number of neighbors K enhances the classification performance since it considers diverse original patterns for synthetic generation. On the other hand, for the distribution performance (Table 10), similar to the artificial datasets' results, having small values for K is generally better. The reason for the discrepancy between the distribution results and the classification results is

Table 17
Gmean for SMOTE, Borderline SMOTE1, Borderline SMOTE2, and Adasyn, on real datasets, using SVM classifier, and using different values of dimensionality d.

Dataset	d value	Unbalanced	SMOTE	Bordeline-SMOTE 1	Bordeline-SMOTE 2	Adasyn	Balanced
SUSY	5	17.59%	56.61%	51.19%	52.06%	58.54%	72.42%
SUSY	10	28.48%	49.17%	49.42%	56.02%	52.63%	78.16%
SUSY	18	26.93%	36.93%	35.09%	46.90%	44.99%	79.25%
EyeState	5	0.00%	46.14%	48.09%	47.60%	49.20%	48.52%
EyeState	10	2.54%	38.50%	45.34%	45.35%	47.08%	59.77%
EyeState	14	10.81%	35.69%	42.52%	43.95%	41.85%	58.45 %
Hepmass	5	14.06%	75.01%	71.60%	72.22%	78.51%	87.16%
Hepmass	10	23.26%	52.49%	54.47%	63.30%	63.75%	87.43%
Hepmass	20	23.74%	35.85%	34.44%	61.57%	69.28%	90.25%
Hepmass	27	17.91%	26.31%	18.13%	61.42%	74.08%	91.39%
ElecGrid	5	17.70%	54.43%	51.41%	54.42%	57.20%	68.72%
ElecGrid	10	32.14%	53.71%	56.54%	58.78%	59.59%	77.95%
ElecGrid	12	40.06%	65.18%	67.93%	65.38%	69.87%	95.28%
NewsPopularity	5	0.00%	46.95%	46.95%	44.92%	44.84%	50.51%
NewsPopularity	10	6.47%	37.51%	31.24%	31.94%	42.69%	54.63%
NewsPopularity	20	14.67%	27.25%	23.62%	30.22%	38.73%	61.11%
NewsPopularity	58	10.93%	16.91%	7.49%	14.26%	54.13%	66.17%

discussed in Section 6. In summary, this is because for small K one generates samples that are nearby, due to the proximity of closest neighbors, and at the locations of these generated points we would have similar distribution as the minority point in question. However, there is a correlation factor that increases with smaller K, and this impacts the classification negatively. More will be discussed in the commentary section, Section 6.

Concerning the impact of the dimension d, we observe from Table 11 that for SMOTE, and its extensions that generally increasing the dimension, increases the error in the distribution, due to the curse of dimensionality as will be discussed in the following section (see Section 6). For the classification performance when varying the dimension d, there are two opposing factors. The first one is the deterioration of the accuracy of the distribution (compared to the true one) with increasing dimension. Of course distribution accuracy impacts classification performance (though in an indirect way). The other factor is the fact that increasing the number of features adds useful information that helps classification performance. From Table 14 one can observe that the latter factor is generally winning for the KNN classifier, and therefore increased d leads to better performance. Consider for example the EyeState problem. For $d = 5,\ 10,$ and 14, we get a gmean for the balanced case of 66.76%, 84.03%, and 94.05% respectively. So there is a natural improvement with adding features. On the other hand, when applying SMOTE to an imbalanced situation of the same problem, we get for the same values of d a gmean of 49.90%, 61.16%, and 64.36% respectively. One can see that even though there is an improvement with increasing d, the rate of improvement is much less than for the pure unbalanced case. This means that the distribution accuracy impact took a toll on the results, but not quite to overcome the other factor. For the SVM classifier we observe an opposite effect to the KNN classifier. The distribution accuracy issue has a bigger weight, and results in a worse overall effect on the classification performance, when increasing d.

6. Commentary on the results

6.1. The accuracy of SMOTE

We observe that the experimental results agree to a reasonable extent with the approximate theoretical results, especially in the direction of changes in relation to the variables of influence. Moreover, the results of the experiments with real data also agree to a good extent with those of the artificial datasets. We can summarize the findings as follows:

- We find that the TVD is always negative, indicating the contractive nature of SMOTE. This means that SMOTE samples are more shrunk inwards.
- The accuracy (of how the distribution of SMOTE samples emulates the true distribution) generally deteriorates with higher dimension d. This is consistent with a general observation for estimation-type problems, the so-called curse of dimensionality. The higher the dimension, the worse is any estimator, whether it is parameter, density, or classifier estimation.
- The accuracy improves as the number of minority examples N is higher. The reason is that for higher N, the K-nearest neighbor patterns become closer to each other. This means that we are dealing with a region of similar density function value. Going too far means reaching out to regions of markedly different density values and hence less “representative” generated patterns. In fact, the deterioration as N becomes smaller is rather sharp and steep. However, as mentioned, for high N the accuracy is reasonable, indicating a reasonable ability to emulate the true distribution. Ironically, this is precisely the situation where we have the least need to apply SMOTE, because we have sufficient data for this case.

- The classification performance, when generating patterns by SMOTE, also deteriorates with decreasing N and increasing d, agreeing to a large extent with the results on distribution accuracy.
- The faithfulness improves with smaller K (of the KNN), becoming the best at having a single neighbor $K = 1$ (case of nearest neighbor). This is because of the argument mentioned in the third point above: far away patterns, corresponding to large K, are less representative. However, for the nearest neighbor a detrimental effect happens that is not apparent in the covariance estimation numbers. We make use of only a single neighbor and all the generated points will be clustered on the line connecting the point in question with that neighbor. There is therefore strong correlation between the generated samples. This will make any generated sample generally less worth (as a piece of it is already duplicated in nearby examples). This observation applies also for small K. As a general guide, selecting K in the range of 5 to 10 seems to be a sensible choice. This would be a trade-off to avoid the high errors of large K and the correlation issue for very small K.
- The conflicting factors mentioned last point generally tilt the balance towards larger K being somewhat better when considering overall classification performance.
- In the distribution experiments, for the less favorable range of factors (i.e. large d, small N and large K) the TVD is between -0.2 and -0.3. This indicates about 20% to 30% discrepancy, which is somewhat high.
- Unlike the covariance matrix, the mean of SMOTE-generated patterns is very close to the true mean. The theoretical analysis approximates the error in the means as zero in case of Gaussian distribution., while the experimental results report an error of about 10^{-3} (for problems whose variances vary in the range from 1 to 10).
- All three sets of experiments for the distribution case (empirical artificial data, theoretical and real-world data) agree to a great extent with the obtained findings.
- The study of the effect of SMOTE on classifier performance reveals that SMOTE provides a benefit because of the boost it provides for minority patterns. However, this benefit varies from one classifier type to another. This applies to both synthetic classification problems and real world problems.
- However, this boost goes only half way compared to the case of using balanced data sets to begin with. This is due to the deficit of SMOTE in emulating true distributions. This deficit increases with decreasing N and increasing d, as expected, and consistent with our earlier results.

For the classification performance on real datasets nearly all of the oversampling methods do help in classification, but they are also far from the ideal (the fully balanced case). Also, increasing N and increasing K helps the performance. Concerning the performance when d increases, there are two conflicting factors. The first one is the deterioration of the performance due to less accurate distribution-effect of SMOTE with increasing dimension. The other one is the fact that increasing the dimension adds more informative features that probably help classification performance. The latter factor generally wins for the KNN classifier only for the real classification problems we tested. However, this also tells us that when encountering severe imbalance, one has to be more inclined to reduce the number of features (by a feature selection step) than for the case of a balanced problem.

ADASYN seems to be the best performing method among the oversampling variants in terms of classification results. It produces consistently better performance.

The concept of SMOTE is based on the idea that the K-nearest neighbors around a point will obey a density function close in value to that of the considered point. Possible future enhancements of SMOTE could be by targeting neighborhoods that have closer density function value. The work by Magdon-Ismail and Atiya [36] shows that there is an interesting dual nature between density estimation and random variate generation.

6.2. On the relation with bias and variance

In any classifier the error can be decomposed into two factors: the bias and the variance. This decomposition is a fundamental concept that exists for all types of classifiers (and all types of machine learning models, see Ben Taieb and Atiya [4]). Essentially, the bias originates from the misspecification of the model. A high bias means that the classifier is not highly attune to the real underlying classification regions and boundaries (for example using a linear classifier when the true boundary is highly nonlinear).

The variance, on the other hand, originates from the variation of the classification model's output due to the variability of the locations of the patterns (from one training set realization to another). For a smaller training set the variance is the more prominent component. In such a case each training pattern's location has a disproportionately high influence on classifier parameters and the estimation error (in classifier parameters) is high due to the small sample available. As the training set size increases the variance shrinks and at some point the bias will become the dominant component.

Our theoretical SMOTE analysis bears only on the bias component. For example, the SMOTE generated examples obey a more shrinking covariance matrix than the true one. This is some form of bias that is going to negatively affect the classifier performance. However, starting from a small training set, generating more data will improve the variance considerably, in a way that will outweigh the negative effect of the misspecification of the distribution (i.e. the bias). This is due to lower parameter estimation error when the sample size increases. However, the generated patterns reach a point whereby generating more points will have less potency in reducing the variance (variance reduction is higher for smaller training sets). At this point, the bias effect may surpass the variance effect and it is time to stop generating more patterns. In other

words, having a misspecified distribution (i.e. a SMOTE generation that is off the true distribution) does not annul the benefit of generating more data, but it keeps a lid on the amount of data that can be beneficially generated.

7. Conclusion

In this paper, we provide a comprehensive analysis of Synthetic Minority over-sampling TEchnique (SMOTE) method. In the presented analysis, we considered both aspects: theory and experimental analysis. SMOTE is an effective method that generates extra examples from the minority class, in an attempt to have its dataset size match that of the majority class, to combat the existing imbalance. SMOTE is also a general method that can be used for regular balanced classification problems of small-sized data. Boosting the data size should improve classification performance and alleviate the detrimental small-sample effects. In this work, we investigated how close the distribution of the patterns generated by SMOTE is from the original distribution. In addition, we determined how the different factors, such as dimension, the number of original minority class patterns, and the number of neighbors affect the generated patterns' distribution. The theoretical and the experimental results generally agree and they should be a useful guide to the user of the SMOTE generation. One of the findings is that for large sized data (large N), SMOTE has better accuracy, so its use is not limited to small sized data or minority class data. We also provide an analysis as to how SMOTE and the other parameters affect classification performance.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

A.1. Expectation derivation

The SMOTE over-sampling technique generates synthetic patterns by interpolation according to Eq. (1). Let $\Delta = X - X_0$, Then:

$$Z = X_0 + w\Delta$$

Then, the expectation is given by:

$$E[Z] = E[(1-w)X_0 + wX]$$

We break down the problem as follows. We first compute the above expectation, given a fixed base pattern X_0, a fixed chosen neighbor k among the K nearest neighbors, and a fixed distance $r = |X - X_0|$ between the neighbor pattern and the base pattern. After this conditional expectation is evaluated, we will take the expectation with respect to r, k and X_0 one by one, to obtain the unconditional expectation.

The first step is given by:

$$E[Z|X_0, r, k] = X_0 + wE[\Delta|X_0, r]$$

By approximating the probability density function using Taylor series around the point X_0 as proposed in [18], we obtain:

$$p(X|X_0) = p(X_0) + \frac{\partial p(X_0)}{\partial X}^T (X - X_0) \tag{A.1}$$

$$E[\Delta|X_0, r] = \int_{S(X_0)} \Delta p(\Delta|X_0, r)\, \mathrm{d}\Delta = \int_{S(X_0)} \Delta \frac{p(\Delta|X_0)}{p(|X - X_0| = r)}\, \mathrm{d}\Delta$$

where $S(X_0)$ is the spherical shell around X_0 containing its k^{th} neighbor that is a distance r away from X_0 and $p(|X - X_0| = r)$ denotes the probability density of distance r between X and X_0. Using Eq. (A.1), $p(\Delta|X_0, r)$ is evaluated as:

$$\int_{S(X_0)} p(\Delta|X_0, r)\, \mathrm{d}X = \int_{S(X_0)} \left[p(X_0) + \frac{\partial p(X_0)}{\partial X}^T (X - X_0) \right] \mathrm{d}X$$

Due to symmetry, $\int_{S(X_0)} (X - X_0) = 0$. Accordingly, $p(|X - X_0| = r)$ would be evaluated as the area of n-dimesional sphere of radius r times $p(X_0)$, as follows:

$$\int_{S(X_0)} p(|X - X_0| = r)\, \mathrm{d}X = \int_{S(X_0)} p(X_0) \mathrm{d}X = p(X_0) \frac{2\pi^{\frac{d}{2}} r^{d-1}}{\Gamma\left(\frac{d}{2}\right)} \tag{A.2}$$

Therefore, the expectation $E[\Delta|X_0, r]$ will be evaluated as:

$$E[\Delta|X_0, r] = \frac{\int_{S(X_0)} [\Delta p(X_0) + \Delta\Delta^T \frac{\partial p(X_0)}{\partial X}]\, \mathrm{d}\Delta}{p(X_0) \frac{2\pi^{\frac{d}{2}} r^{d-1}}{\Gamma\left(\frac{d}{2}\right)}} \tag{A.3}$$

We now evaluate the numerator of Eq. (A.3). The first term $\int_{S(X_0)} \Delta \, \mathrm{d}\Delta$ equals zero due to the symmetry of the sphere $S(X_0)$. For a given length of the Δ vector equal to r, the radius of the sphere around X_0, the second term can be evaluated as:

$$\int_{S(X_0)} \Delta\Delta^T \, \mathrm{d}\Delta = \frac{\pi^{\frac{d}{2}} r^{d+1} I}{\Gamma\left(\frac{d}{2}+1\right)} \tag{A.4}$$

where d is the dimension of the space and I is the identity matrix. The previous equation is proved below. Over the sphere there are no nonzero correlation terms, so the above integral in Eq. (A.4) has to be diagonal. Moreover, by symmetry, because we cannot favor any direction over the other, all diagonal elements are equal, so we obtain:

$$\int_{S(X_0)} \Delta\Delta^T \, \mathrm{d}\Delta = \alpha I$$

for some α. Let:

$$J = \int_{S(X_0)} \Delta\Delta^T \, \mathrm{d}\Delta \tag{A.5}$$

Then, taking the trace of both sides of Eq. (A.5):

$$\mathrm{tr}(J) = \int_{S(X_0)} \mathrm{tr}(\Delta\Delta^T) \, \mathrm{d}\Delta = \mathrm{tr}(\alpha I)$$

Using the fact that $\mathrm{tr}(\Delta\Delta^T) = \mathrm{tr}(\Delta^T\Delta) = r^2$, we get:

$$\mathrm{tr}(\alpha I) = \alpha d = \int_{S(X_0)} r^2 \, \mathrm{d}\Delta = \frac{r^2 \; d\pi^{\frac{d}{2}} r^{d-1}}{\Gamma\left(\frac{d}{2}+1\right)}$$

Then $\alpha = \frac{\pi^{\frac{d}{2}} r^{d+1}}{\Gamma\left(\frac{d}{2}+1\right)}$. Accordingly, Eq. (A.4) is verified. Then, substituting from Eq. (A.4) into Eq (A.3):

$$E[\Delta | X_0, r] = \frac{r^2 \frac{\partial p(X_0)}{\partial X}}{dp(X_0)} \tag{A.6}$$

Then, for a certain chosen neighbor k of certain r euclidean distance to X_0, expectation of Z would be :

$$E[Z | X_0, r, k] = X_0 + \frac{wr^2 \frac{\partial p(X_0)}{\partial X}}{dp(X_0)}$$

Taking the expectation over r, we eliminate the conditioning on r, so we get:

$$E[Z | X_0, k] = \int_0^\infty p(r) E[Z | X_0, r, k] \, \mathrm{d}r$$

We can obtain an approximation of $p(r)$, i.e. the probability density of the distance to the k^{th} neighbor. We use the concept of coverage of S_{X_0}, introduced in [18]. In their work, the authors of [18] define the following quantity:

$$u = \int_{B_k(X_0)} p(X) dX \tag{A.7}$$

where $B_k(X_0)$ is the ball around X_0, reaching out to the k^{th} neighbor. This quantity u follows a beta distribution: $u \sim \beta(k, N - k + 1)$ where N is the total number of examples. Let the volume of the ball be denoted by v, as follows:

$$v = \frac{\pi^{\frac{d}{2}} r^d}{\Gamma\left(\frac{d}{2}+1\right)}$$

Using the Taylor series approximation in Eq. (A.7), we get:

$$u = p(X_0) \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)} r^d + \frac{\partial p(X_0)}{\partial X}^T \int_{S(X_0)} \Delta \, \mathrm{d}\Delta$$

Since the second term $\int_{S(X_0)} \Delta \, \mathrm{d}\Delta$ of the equation above, evaluates to zero, then:

$$u = p(X_0) \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)} r^d$$

Let $\lambda = p(X_0) \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)}$, then $r = (u/\lambda)^{(1/d)}$.

Then, the expectation $E[Z|X_0, k]$ can be evaluated as:

$$E[Z|X_0, k] = X_0 + \frac{w\frac{\partial p(X_0)}{\partial X}}{dp(X_0)} \int_{r=0}^{\infty} p(r) r^2 \, \mathrm{d}r \tag{A.8}$$

$$= X_0 + \frac{w\frac{\partial p(X_0)}{\partial X}}{dp(X_0)} E_u\left[\frac{u}{\lambda}\right]^{\frac{2}{d}} \tag{A.9}$$

But, since $p(u)$ is Beta$(k, N-k+1)$, the integrand in E_u can be simplified into another beta function, that can be integrated easily. We get after some simplification:

$$E[Z|X_0, k] = X_0 + \frac{wN!\Gamma\left(1+\frac{2}{d}\right)^{\frac{2}{d}}\Gamma\left(k+\frac{2}{d}\right)\frac{\partial p(X_0)}{\partial X}}{\pi d(k-1)!\Gamma\left(N+\frac{2}{d}+1\right)p(X_0)^{(1+\frac{2}{d})}} \tag{A.10}$$

Taking expectation over X_0:

$$E[Z|k] = \mu_{X_0} + \frac{wN!\Gamma\left(1+\frac{2}{d}\right)^{\frac{2}{d}}\Gamma\left(k+\frac{2}{d}\right)}{\pi d(k-1)!\Gamma\left(N+\frac{2}{d}+1\right)} \int_{X_0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X} \, \mathrm{d}X_0 \tag{A.11}$$

where μ_{X_0} is the mean of the class distribution. In the next step, we compute the expectation over k. Since the K neighbors are all equally likely to be selected for interpolation, then

$$E[Z] = \sum_{k=1}^{K} \frac{1}{K} E[Z|k] \tag{A.12}$$

$$E[Z] = \mu_{X_0} + \frac{wN!\Gamma\left(1+\frac{2}{d}\right)^{\frac{2}{d}}}{\pi dK\Gamma\left(N+\frac{2}{d}+1\right)} \int_{X_0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X} \, \mathrm{d}X_0 \sum_{k=1}^{K} \frac{\Gamma\left(k+\frac{2}{d}\right)}{(k-1)!} \tag{A.13}$$

After algebraic manipulation, it can be shown that:

$$\sum_{k=1}^{K} \frac{\Gamma\left(k+\frac{2}{d}\right)}{(k-1)!} = \frac{d\Gamma\left(K+1+\frac{2}{d}\right)}{(K-1)!(d+2)} \tag{A.14}$$

Then, the expectation of Z can be evaluated as:

$$E[Z] = \mu_{X_0} + wC \int_{X_0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X} \, \mathrm{d}X_0 \tag{A.15}$$

where C is the following constant:

$$C = \frac{N!\Gamma\left(1+\frac{2}{d}\right)^{\frac{2}{d}}\Gamma\left(K+\frac{2}{d}+1\right)}{\pi K!(d+2)\Gamma\left(N+\frac{2}{d}+1\right)} \tag{A.16}$$

The generated point is interpolated uniformly on the line connecting X_0 and X, so w is typically selected from a uniform distribution in [0,1]. As a generalization, we consider the generation of w as uniform in [0, w^*] instead. Taking the expectation over w, we get:

$$E[Z] = \mu_{X_0} + C \int_{X_0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X} \, \mathrm{d}X_0 \int_{w=0}^{w^*} \frac{w}{w^*} \, \mathrm{d}w = \mu_{X_0} + \frac{Cw^*}{2} \int_{X_0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X} \, \mathrm{d}X_0 \tag{A.17}$$

This is the final formula for the expectation. It can be argued that the integral on the right side is very small or negligible. For well-behaved densities, such as unimodal densities, the gradient $\frac{\partial p(X_0)}{\partial X}$ will be pointing towards the center or the mode of the density. When we sum up all these during the process of integrating over the space, all these radially pointing vectors will tend to cancel out. Simulation results confirm that this term is negligible. Moreover, analysis of the case of multivariate Gaussian probability density $p(X_0)$ (as given in A.3) confirms this finding. Hence, the expectation of the generated examples $E[Z]$ tends to be very close to the true mean of the distribution μ_{X_0}. As such, that aspect is favorably preserved.

A.2. Covariance matrix derivation

Here, we derive the covariance matrix of the patterns generated using SMOTE. Similar to the expectation part of the proof, the covariance matrix of Z is evaluated given X_0, k and r. The resulting expectation is denoted as $\Sigma_{(Z|k,X_0,r)}$. Then we will perform expectation over the random variables r, X_0, then k respectively, to obtain a general expectation of the covariance matrix. The covariance is given by:

$$\Sigma_Z = E[(Z-\mu_Z)(Z-\mu_Z)^T] \tag{A.18}$$

Using Eq. (2) :

$$\Sigma_Z = E\Big[\big((X_0 - \mu_{X_0}) + w(\Delta - \mu_\Delta)\big)\big((X_0 - \mu_{X_0}) + w(\Delta - \mu_\Delta)\big)^T\Big] \tag{A.19}$$

$$\Sigma_Z = \Sigma_{X_0} + w^2 E[\Delta\Delta^T] - w^2 \mu_\Delta {\mu_\Delta}^T + wE[(X_0 - \mu_{X_0})(\Delta - \mu_\Delta)^T] + wE[(\Delta - \mu_\Delta)(X_0 - \mu_{X_0})^T] \tag{A.20}$$

For a given example X_0 and for a certain chosen neighbor k whose euclidean distance from X_0 is r, we will evaluate the terms of Eq. (A.20) successively. The expectation $E[\Delta\Delta^T|X0, r, k]$ will be:

$$E[\Delta\Delta^T|X_0, r, k] = \int_{S(X_0)} \frac{\Delta\Delta^T p(\Delta|X_0, r, k)}{p(|X - X_0| = r)}\, \mathrm{d}\Delta \tag{A.21}$$

Using Taylor approximation and substituting from Eq. (A.1):

$$E[\Delta\Delta^T|X_0, r, k] = \frac{\int_{S(X_0)} \Delta\Delta^T [p(X_0) + \frac{\partial p(X_0)}{\partial X}^T \Delta]\, \mathrm{d}\Delta}{p(|X - X_0| = r)} \tag{A.22}$$

Accordingly, substituting from Eq. (A.2) and Eq. (A.4):

$$E[\Delta\Delta^T|X_0, r, k] = \frac{r^2}{d} I + \frac{\Gamma\left(\frac{d}{2}\right)}{2\pi^{\frac{d}{2}} r^{d-1} p(X_0)} \int_{S(X_0)} \Delta\Delta^T \frac{\partial p(X_0)}{\partial X}^T \Delta\, \mathrm{d}\Delta \tag{A.23}$$

The integration $\int_{S(X_0)} \Delta\Delta^T \frac{\partial p(X_0)}{\partial X}^T \Delta\, \mathrm{d}\Delta$ can be proved to be zero using spherical co-ordinates transformation. Accordingly, substituting into Eq. (A.23):

$$E[\Delta\Delta^T|X_0, r, k] = \frac{r^2}{d} I \tag{A.24}$$

In the next step, we take the expectation with respect to r. Similar to the expectation derivation, Eq. (A.8), the integration $\int_r p(r) r^2\, \mathrm{d}r$ is computed, and $E[\Delta\Delta^T|X_0, k]$ becomes:

$$E[\Delta\Delta^T|X_0, k] = \frac{N!\Gamma\left(1 + \frac{2}{d}\right)^{\frac{2}{d}} \Gamma\left(k + \frac{2}{d}\right) I}{\pi d (k-1)! \Gamma\left(N + \frac{2}{d} + 1\right) p(X_0)^{\frac{2}{d}}} \tag{A.25}$$

In the next step we integrate out X_0, as follows:

$$E[\Delta\Delta^T|k] = \frac{N!\Gamma\left(1 + \frac{2}{d}\right)^{\frac{2}{d}} \Gamma\left(k + \frac{2}{d}\right)}{\pi d (k-1)! \Gamma\left(N + \frac{2}{d} + 1\right)} I \int_{X_0} p(X_0)^{1 - \frac{2}{d}}\, \mathrm{d}X_0 \tag{A.26}$$

Then, we take the expectation with respect to k, similar to Eq. (A.12):

$$E[\Delta\Delta^T] = CI \int_{X_0} p(X_0)^{1 - \frac{2}{d}}\, \mathrm{d}X_0 \tag{A.27}$$

Now we evaluate the fourth and fifth terms of Eq. (A.20) (they are transposes of each other). We obtain the following:

$$E[(\Delta - \mu_\Delta)(X_0 - \mu_{X_0})^T] = E[\Delta(X_0 - \mu_{X_0})^T] - E[\mu_\Delta (X_0 - \mu_{X_0})^T] \tag{A.28}$$

$$E[\Delta(X_0 - \mu_{X_0})^T|X_0, r, k] = \int_{S(X_0)} p(\Delta|X_0, r, k) \Delta (X_0 - \mu_{X_0})^T\, \mathrm{d}\Delta \tag{A.29}$$

From Eq. (A.6), we obtain :

$$E[\Delta(X_0 - \mu_{X_0})^T|X0, r, k] = \frac{r^2 \frac{\partial p(X_0)}{\partial X} (X_0 - \mu_{X_0})^T}{d p(X_0)} \tag{A.30}$$

Then, taking expectation over r, similar to Eq. (A.8), we get:

$$E[\Delta(X_0 - \mu_{X_0})^T|X_0, k] = \frac{N!\Gamma\left(1 + \frac{2}{d}\right)^{\frac{2}{d}} \Gamma\left(k + \frac{2}{d}\right)}{\pi d (k-1)! \Gamma\left(N + \frac{2}{d} + 1\right)} \times \frac{\partial p(X_0)}{\partial X} \frac{[(X_0 - \mu_{X_0})^T]}{p(X_0)^{1 + \frac{2}{d}}}$$

Then, we take the expectation with respect to X_0, like in Eq. (A.11):

$$E[\Delta(X_0 - \mu_{X_0})^T|k] = \frac{N!\Gamma\left(1 + \frac{2}{d}\right)^{\frac{2}{d}} \Gamma\left(k + \frac{2}{d}\right)}{\pi d (k-1)! \Gamma\left(N + \frac{2}{d} + 1\right)} \times \int_{X_0} p(X_0)^{-\frac{2}{d}} \frac{\partial p(X_0)}{\partial X} [(X_0 - \mu_{X_0})^T]\, \mathrm{d}X_0$$

Taking the expectation over k, similar to Eq. (A.12), we get:

$$E[\Delta(X_0-\mu_{X_0})^T]=C\int_{X_0}p(X_0)^{-\frac{2}{d}}\frac{\partial p(X_0)}{\partial X}[(X_0-\mu_{X_0})^T]\,dX_0 \tag{A.31}$$

Since $E[\mu_\Delta(X_0-\mu_{X_0})^T]=\mu_\Delta E[(X_0-\mu_{X_0})^T]=0$, then substituting from Eq. (A.31) in Eq. (A.28) results in:

$$E[(\Delta-\mu_\Delta)(X_0-\mu_{X_0})^T]=C\int_{X_0}p(X_0)^{-\frac{2}{d}}\frac{\partial p(X_0)}{\partial X}[(X_0-\mu_{X_0})^T]\,dX_0 \tag{A.32}$$

From equations (2) and (A.17), $E[\Delta]$ is evaluated as:

$$E[\Delta]=C\int_{X_0}p(X_0)^{\frac{-2}{d}}\frac{\partial p(X_0)}{\partial X}\,dX_0 \tag{A.33}$$

Hence, substituting from Eqs. (A.27), (A.33), and (A.32) into Eq. (A.20):

$$\begin{aligned}\Sigma_Z=\;&\Sigma_{X_0}+w^2CI\int_{X_0}p(X_0)^{1-\frac{2}{d}}\,dX_0-w^2C^2\int_{X_0}p(X_0)^{\frac{-2}{d}}\frac{\partial p(X_0)}{\partial X}\,dX_0\int_{X_0}p(X_0)^{\frac{-2}{d}}\frac{\partial p(X_0)}{\partial X}^T\,dX_0\\&+\frac{wC}{2}\left[\int_{X_0}p(X_0)^{-\frac{2}{d}}\frac{\partial p(X_0)}{\partial X}[(X_0-\mu_{X_0})^T]\,dX_0+\int_{X_0}p(X_0)^{-\frac{2}{d}}(X_0-\mu_{X_0})\frac{\partial p(X_0)}{\partial X}^T\,dX_0\right]\end{aligned} \tag{A.34}$$

Taking expectation over w, where w is a uniform random variable $[0, w^*]$:

$$\begin{aligned}\Sigma_Z=\;&\Sigma_{X_0}+\frac{Cw^{*2}}{3}\int_{X_0}p(X_0)^{1-\frac{2}{d}}\,dX_0I-\frac{C^2w^{*2}}{3}\int_{X_0}p(X_0)^{\frac{-2}{d}}\frac{\partial p(X_0)}{\partial X}\,dX_0\int_{X_0}p(X_0)^{\frac{-2}{d}}\frac{\partial p(X_0)}{\partial X}^T\,dX_0\\&+\frac{Cw^*}{2}\left[\int_{X_0}p(X_0)^{-\frac{2}{d}}\frac{\partial p(X_0)}{\partial X}[(X_0-\mu_{X_0})^T]\,dX_0+\int_{X_0}p(X_0)^{-\frac{2}{d}}(X_0-\mu_{X_0})\frac{\partial p(X_0)}{\partial X}^T\,dX_0\right]\end{aligned} \tag{A.35}$$

A.3. Case of Gaussian multivariate distribution

Assuming that the original minority class patterns X_0 follow a multivariate Gaussian distribution with mean μ_{X_0} and covariance matrix Σ_{X_0}. In this section, we will derive the mean and covariance of the synthetic examples generated by SMOTE. The multivariate Gaussian distribution is given by

$$p(X_0)=\frac{1}{(2\pi)^{\frac{d}{2}}\det^{\frac{1}{2}}(\Sigma_{X_0})}e^{\frac{-1}{2}(X_0-\mu_{X_0})^T\Sigma_{X_0}^{-1}(X_0-\mu_{X_0})}$$

Then, the gradient of Gaussian density would be:

$$\frac{\partial p(X_0)}{\partial X}=-p(X_0){\Sigma_{X_0}}^{-1}(X_0-\mu_{X_0}) \tag{A.36}$$

Substituting from Eq. (A.36) into Eq. (A.15):

$$E[Z]=\mu_{X_0}+wC\int_{X_0}-p(X_0)^{1-\frac{2}{d}}{\Sigma_{X_0}}^{-1}(X_0-\mu_{X_0})\,dX_0 \tag{A.37}$$

We will show here that the integral:

$I_a\equiv\int_{X_0}-p(X_0)^{1-\frac{2}{d}}{\Sigma_{X_0}}^{-1}(X_0-\mu_{X_0})\,dX_0$ evaluates to zero. Assume that $d\neq2$ (otherwise the integral does not converge). Then,

$$I_a=-{\Sigma_{X_0}}^{-1}\times\int_{X_0}\frac{e^{-\frac{(d-2)}{2d}(X_0-\mu_{X_0})^T\Sigma_{X_0}^{-1}(X_0-\mu_{X_0})}}{(2\pi)^{\frac{(d-2)}{2}}(det(\Sigma_{X_0}))^{\frac{d-2}{2d}}}(X_0-\mu_{X_0})\,dX_0 \tag{A.38}$$

Let $\Sigma_1=\frac{d}{d-2}\Sigma_{X_0}$, which implies that $\Sigma_1^{-1}=\frac{d-2}{d}\Sigma_{X_0}^{-1}$ and $\det(\Sigma_1)=\left(\frac{d}{d-2}\right)^d\det(\Sigma_{X_0})$. Then,

$$I_a=-2\pi(\det^{\frac{1}{d}}(\Sigma_{X_0}))\left(\frac{d}{d-2}\right)^{\frac{d}{2}}{\Sigma_{X_0}}^{-1}\int_Y p(Y)(Y-\mu_Y)\,dY \tag{A.39}$$

$$=-2\pi(\det^{\frac{1}{d}}(\Sigma_{X_0}))\left(\frac{d}{d-2}\right)^{\frac{d}{2}}{\Sigma_{X_0}}^{-1}E[Y-\mu_Y]=0 \tag{A.40}$$

since $Y\sim N(\mu_{X_0},\Sigma_1)$. Then, $\mu_\Delta=0$ and substituting into Eq. (A.37) results in:

$$E[Z]=\mu_{X_0} \tag{A.41}$$

We now estimate the covariance matrix of the generated examples in case of Gaussian density assumption. The third term of covariance matrix in Eq. (A.35) evaluates to zero since $E[\Delta] = 0$. The integrals of the second term and the last two terms in Eq. (A.35) will be evaluated as follows:

Let $I_1 = \int_{X_0} p(X_0)^{1-\frac{2}{d}}\, dX_0$, and $I_2 = \int_{X_0} p(X_0)^{-\frac{2}{d}} \frac{\partial p(X_0)}{\partial X} (X_0 - \mu_{X_0})^T\, dX_0$.

For I_1, similar to the expectation derivation of Eq. (A.39), and using $Y \sim N(\mu_{X_0}, \Sigma_2)$ where $\Sigma_2 = \frac{d}{d-2}\Sigma_{X_0}$, we obtain:

$$I_1 = (2\pi)\det^{\frac{1}{d}}(\Sigma_{X_0})\left(\frac{d}{d-2}\right)^{\frac{d}{2}} \int_Y p(Y)\, dY$$

Since $\int p(Y)\, dY = 1$, we obtain:

$$I_1 = (2\pi)\det^{\frac{1}{d}}(\Sigma_{X_0})\left(\frac{d}{d-2}\right)^{\frac{d}{2}} \tag{A.42}$$

Concerning I_2, we substitute the formula for $\frac{\partial p(X_0)}{\partial X}$:

$$I_2 = -\int_{X_0} p(X_0)^{1-\frac{2}{d}} {\Sigma_{X_0}}^{-1}(X_0 - \mu_{X_0})(X_0 - \mu_{X_0})^T\, dX_0$$

which gives:

$$I_2 = -2\pi \det^{\frac{1}{d}}(\Sigma_{X_0})\left(\frac{d}{d-2}\right)^{\frac{d+2}{2}} I \tag{A.43}$$

Substituting from Eq. (A.42), and Eq. (A.43) into Eq. (A.35):

$$\Sigma_Z = \Sigma_{X_0} + \left[(2\pi)^{\frac{1-d}{2}} \frac{Cw^{*2}}{3} \det^{\frac{1-d}{2d}}(\Sigma_{X_0})\left(\frac{d}{2d-1}\right)^{\frac{d}{2}} - 2\pi Cw^* \det^{\frac{1}{d}}(\Sigma_{X_0})\left(\frac{d}{d-2}\right)^{\frac{d+2}{2}}\right] I$$

A.4. Case of Laplace multivariate distribution

Assuming that the original minority class patterns X_0 follow a multivariate Laplace distribution with mean μ_{X_0}. In this section, we will derive the mean and covariance of the synthetic examples generated by SMOTE. The probability density of the multivariate Laplace distribution is given by:

$$p(X_0) = \left(\frac{\alpha}{2}\right)^d e^{-\alpha \sum_{i=1}^{d} |X_0(i) - \mu_{X_0}(i)|} \tag{A.44}$$

where α is the distribution parameter, $X_0(i)$ denotes the i^{th} element of the vector X_0, μ_{X_0} is the original mean of distribution, and Σ_{X_0} is a diagonal matrix where its diagonal entries$=\frac{2}{\alpha^2}$. Then, the gradient of the density, $\frac{\partial p(X_0)}{\partial X}$ would be:

$$\frac{\partial p(X_0)}{\partial X} = -\frac{\alpha^{d+1}}{2^d}[sign(X_0(1) - \mu_{X_0}(1), \ldots . sign(X_0(d) - \mu_{X_0}(d)]^T \tag{A.45}$$

where $sign(x)$ denotes the sign function of x, which evaluates to positive one if the x if positive or zero and evaluates to negative one if the x is negative.

We can simplify Eq. (A.45) to be:

$$\frac{\partial p(X_0)}{\partial X} = -\frac{\alpha^{d+1}}{2^d} Sign(X_0 - \mu_{X_0}) \tag{A.46}$$

where $Sign(X)$ is a vectorized version of $sign(x)$ that takes a vector x and calculates element-wise sign of its elements. Substituting from Eq. (A.44) and Eq. (A.45) into Eq. (3).

$$E[Z] = \mu_{X_0} + \frac{Cw^*}{2} \int_{X_0} -\alpha \left(\frac{\alpha}{2}\right)^{d-2} e^{-\alpha' \sum_{i=1}^{d} |X_0(i) - \mu_{X_0}(i)|} Sign(X_0 - \mu_{X_0})\, dX_0 \tag{A.47}$$

where $\alpha' = \alpha \frac{(d-2)}{d}$.

Since the variables of integration in Eq. (A.47) are uncorrelated, so we can evaluate it separately per dimension. Let I_b be the integration over the first dimension be as follows:

$$I_b = \int_{X_0(1)} -\alpha \left(\frac{\alpha}{2}\right)^{d-2} e^{-\alpha' \sum_{i=1}^{d} |X_0(i) - \mu_{X_0}(i)|} sign(X_0(1) - \mu_{X_0}(1)) \tag{A.48}$$

Accordingly:

$$I_b = -\frac{\alpha}{\alpha'} \left(\frac{\alpha}{2}\right)^{d-2} e^{-\alpha' \sum_{i=1}^{d} |X_0(i) - \mu_{X_0}(i)|} = 0 \tag{A.49}$$

since all X_0 elements vary from $-\infty$ to ∞.

Accordingly, substituting from Eq. (A.49) into Eq. (A.47), the mean of generated patterns $E[Z]$ will be as follows:

$$E[Z] \approx \mu_{X_0} \tag{A.50}$$

Regarding the covariance matrix of the generated patterns, Σ_Z, using Eq. (4), it can be observed that there are three main integrals to be evaluated:

Let $I_1 = \int_{X_0} p(X_0)^{1-\frac{2}{d}}\, dX_0$.

$$I_1 = \int_{X_0} \left(\frac{\alpha}{2}\right)^{d-2} e^{-\alpha' \sum_{i=1}^{d} |X_0(i)-\mu_{X_0}(i)|}\, dX_0 \tag{A.51}$$

where $\alpha' = \alpha \frac{d-2}{d}$. Accordingly:

$$I_1 = \left(\frac{\alpha}{2}\right)^{d-2} \left(\frac{2}{\alpha'}\right)^{d} \int_{X_0} \left(\frac{\alpha'}{2}\right)^{d} e^{-\alpha' \sum_{i=1}^{d} |X_0(i)-\mu_{X_0}(i)|}\, dX_0 \tag{A.52}$$

However, the integration $\int_{X_0} (\frac{\alpha'}{2})^d e^{-\alpha' \sum_{i=1}^{d} |X_0(i)-\mu_{X_0}(i)|}\, dX_0$ evaluates to 1 since it is probability density of a multivariate Laplace distribution of α' parameter. Then:

$$I_1 = \left(\frac{\alpha}{2}\right)^{d-2} \left(\frac{2}{\alpha'}\right)^{d} = \frac{4}{\alpha^2} \left(\frac{d}{d-2}\right)^{d} \tag{A.53}$$

$$I_2 = \int_{X_0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X}\, dX_0 = 0 \tag{A.54}$$

This integral I_2 evaluates to zero as we proved in the derivation of the mean μ_Z.

Let $I_3 = \int_{X_0} p(X_0)^{-\frac{2}{d}} (X_0 - \mu_{X_0}) \frac{\partial p(X_0)}{\partial X}^T\, dX_0$.

$$I_3 = \int_{X_0} -\alpha \left(\frac{\alpha}{2}\right)^{d-2} e^{-\alpha' \sum_{i=1}^{d} |X_0(i)-\mu_{X_0}(i)|} (X_0 - \mu_{X_0}) (Sign(X_0 - \mu_{X_0}))^T\, dX_0 \tag{A.55}$$

It can be observed from Eq. (A.55) that the integral I_3 evaluates to a matrix. We will evaluate two general integrals that constitute I_3: I_{ii} is an integral for the i^{th} diagonal element, and I_{ij} an integral for off-diagonal element indexed by row i and column j.

$$I_{ii} = \int_{X_0} -\alpha \left(\frac{\alpha}{2}\right)^{d-2} e^{-\alpha' \sum_{i=1}^{d} |X_0(i)-\mu_{X_0}(i)|} (X_0(i) - \mu_{X_0}(i)) sign(X_0(i) - \mu_{X_0}(i))\, dX_0 \tag{A.56}$$

To simplify notation let $X_0(i)$ be x and $\mu_{X_0}(i)$ be μ.

$$I_{ii} = \int_{X_0(j)} e^{-\alpha' \sum_{j\neq i} |X_0(j)-\mu_{X_0}(j)|}\, dX_0(j) \int_x -\alpha \left(\frac{\alpha}{2}\right)^{d-2} e^{-\alpha'|x-\mu|} (x-\mu) sign(x-\mu)\, dx \tag{A.57}$$

Accordingly:

$$I_{ii} = -\alpha \left(\frac{\alpha}{2}\right)^{d-2} \int_{X_0(j)} e^{-\alpha' \sum_{j\neq i} |X_0(j)-\mu_{X_0}(j)|}\, dX_0(j) \int_x e^{-\alpha'|x-\mu|} |x-\mu|\, dx \tag{A.58}$$

Evaluating the integral $I_4 = \int_x e^{-\alpha'|x-\mu|} |x-\mu|\, dx$, and let $z = x - \mu$, accordingly:

$$I4 = \int_{x=-\infty}^{\infty} e^{-\alpha'|x-\mu|} |x-\mu|\, dx = 2 \int_{z=0}^{\infty} e^{-\alpha' z} z\, dz \tag{A.59}$$

Solving $\int_{z=0}^{\infty} e^{-\alpha' z} z\, dz$ using integration by parts results in:

$$I4 = \frac{2}{\alpha'^2} \tag{A.60}$$

Substitute with Eq. (A.60) into Eq. (A.58):

$$I_{ii} = -\frac{2\alpha}{\alpha'^2} \left(\frac{\alpha}{2}\right)^{d-2} \int_{X_0(j)} e^{-\alpha' \sum_{j\neq i} |X_0(j)-\mu_{X_0}(j)|}\, dX_0(j) \tag{A.61}$$

Evaluating one of the remaining multiple integrals, for example w.r.t. $X_0(k)$, and let $X_0(k) = x$ and $\mu_{X_0}(k) = \mu_k$.

$$I_{ii} = -\frac{2\alpha}{\alpha'^2} \left(\frac{\alpha}{2}\right)^{d-2} \int_{X_0(j)} e^{-\alpha' \sum_{j\neq k} |X_0(j)-\mu_{X_0}(j)|}\, dX_0(j) \int_x e^{-\alpha'|x-\mu_k|}\, dx \tag{A.62}$$

Let $I_5 = \int_x e^{-\alpha'|x-\mu_k|}\,dx$.

$$I_5 = \int_x e^{-\alpha'|x-\mu_k|}\,dx = \frac{2}{\alpha'}\int_x \frac{\alpha'}{2} e^{-\alpha'|x-\mu_k|}\,dx = \frac{2}{\alpha'} \tag{A.63}$$

Since $\int_x \frac{\alpha'}{2} e^{-\alpha'|x-\mu_k|}\,dx$ is an integral over probability density function of one-dimensional Laplace distribution as defined in Eq. (A.44), so it evaluates to 1. Accordingly:

$$I_{ii} = -\frac{4\alpha}{\alpha'^3}\left(\frac{\alpha}{2}\right)^{d-2}\int_{X_0(j)} e^{-\alpha'\sum_{j\neq k}|X_0(j)-\mu_{X_0}(j)|}\,dX_0(j) \tag{A.64}$$

Evaluating the integrals for all j, where $j \neq i$ results in the following:

$$I_{ii} = -4\alpha^2\left(\frac{d}{d-2}\right)^{d+1} \tag{A.65}$$

Evaluating the off-diagonal integral I_{ij}:

$$I_{ij} = \int_{X_0} -\alpha\left(\frac{\alpha}{2}\right)^{d-2} e^{-\alpha'\sum_{i=1}^{d}|X_0(i)-\mu_{X_0}(i)|}(X_0(i)-\mu_{X_0}(i))sign(X_0(j)-\mu_{X_0}(j))\,dX_0 \tag{A.66}$$

To simplify notation let $X_0(i)$ be x, $\mu_{X_0}(i)$ be μ_x, and $X_0(j)$ be y, and $\mu_{X_0}(j)$ be μ_y, then:

$$I_{ij} = \alpha\left(\frac{\alpha}{2}\right)^{d-2}\int_{X_0(i)_{i\notin\{j,k\}}} -e^{-\alpha'\sum_{i\notin\{j,k\}}|X_0(i)-\mu_{X_0}(i)|} \times \int_x\int_y e^{-\alpha'(x-\mu_x)}e^{-\alpha'(y-\mu_y)}(x-\mu_x)sign(y-\mu_y)\,dx\,dy \tag{A.67}$$

Let $I_6 = \int_x\int_y e^{-\alpha'(x-\mu_x)}e^{-\alpha'(y-\mu_y)}(x-\mu_x)sign(y-\mu_y)\,dx\,dy$.

Evaluating I_6 w.r.t. y first.

$$I_6 = \int_x e^{-\alpha'(x-\mu_x)}(x-\mu_x)\int_y e^{-\alpha'(y-\mu_y)}sign(y-\mu_y)\,dy\,dx \tag{A.68}$$

However, the integral $\int_{y=-\infty}^{\infty} e^{-\alpha'(y-\mu_y)}sign(y-\mu_y)\,dy$ evaluates to zero as we mentioned previously in the expectation's derivation.

Accordingly, both I_6, and I_{ij} evaluates to zero. So, the integral I_3 is a diagonal matrix, and since the integral I_{ii} does not depend on the value of i (see Eq. (A.65)), then:

$$I_3 = -4\alpha^2\left(\frac{d}{d-2}\right)^{d+1} I \tag{A.69}$$

where I is the identity matrix. Substituting from Eq. (A.53), Eq. (A.54) and Eq. (A.69) into Eq. (4):

$$\Sigma_Z = \Sigma_{X_0} + \frac{4Cw^*}{\alpha^2}\left(\frac{d}{d-2}\right)^{d}\left[\frac{w^*}{3} - \left(\frac{d}{d-2}\right)\right]I \tag{A.70}$$

References

[1] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, in: European conference on machine learning, Springer, 2004, pp. 39–50.
[2] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 20–29.
[3] C. Bellinger, C. Drummond, N. Japkowicz, Manifold-based synthetic oversampling with manifold conformance estimation, Mach. Learn. 107 (3) (2018) 605–637.
[4] S. Ben Taieb, A.F. Atiya, A bias and variance analysis for multistep-ahead time series forecasting, IEEE Trans. Neural Netw. Learn. Syst. 27 (1) (2016) 62–76.
[5] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-Level-SMOTE : Safe-Level-Synthetic Minority Over-Sampling TEchnique, in: Advances in Knowledge Discovery and Data Mining, Springer, 2009, pp. 475–482.
[6] P. Chan, W. Fan, A. Prodromidis, S. Stolfo, Distributed data mining in credit card fraud detection, Intell. Syst. their Appl., IEEE 14 (6) (1999) 67–74.
[7] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (1) (2002) 321–357.
[8] N. Chawla, A. Lazarevic, L. Hall, K. Bowyer, SMOTEBoost: Improving prediction of the minority class in boosting, Knowledge Discovery in Databases: PKDD 2003 (2003) 107–119.
[9] S. Chen, H. Haibo, E. Garcia, RAMOBoost : Ranked minority oversampling in boosting, IEEE Trans. Neural Netw. 21 (10) (2010) 1624–1642.
[10] A. Dal Pozzolo, O. Caelen, R.A. Johnson, G. Bontempi, Calibrating probability with undersampling for unbalanced classification, in: IEEE Symposium Series on Computational Intelligence, 2015, pp. 159–166.
[11] G. Douzas, F. Bacao, F. Last, Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE, Inf. Sci. 465 (2018) 1–20.
[12] D. Dua, C. Graff, UCI machine learning repository, 2017, http://archive.ics.uci.edu/ml.
[13] H. Dubey, V. Pudi, Class Based Weighted K-Nearest Neighbor over Imbalance Dataset, in: Advances in Knowledge Discovery and Data Mining, Springer, 2013, pp. 305–316.
[14] R. Dubey, J. Zhou, Y. Wang, P. Thompson, J. Ye, Analysis of sampling techniques for imbalanced data: an n= 648 ADNI study, NeuroImage 87 (2014) 220–241.
[15] W. Fan, S. Stolfo, J. Zhang, P. Chan, Adacost: misclassification cost-sensitive boosting, in: ICML, 99, 1999, pp. 97–105.
[16] H. Fayed, A. Atiya, A novel template reduction approach for the-Nearest neighbor method, IEEE Trans. Neural Netw. 20 (5) (2009) 890–896.

[17] Y. Freund, R. Schapire, Experiments with a new boosting algorithm, in: ICML, 96, 1996, pp. 148–156.
[18] K. Fukunaga, L. Hostetler, Optimization of k nearest neighbor density estimates, IEEE Trans. Inf. Theory 19 (3) (1973) 320–326.
[19] S. García, F. Herrera, Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy, Evol. Comput. 17 (3) (2009) 275–306.
[20] V. García, J. Sánchez, R. Mollineda, Exploring the performance of resampling strategies for the class imbalance problem, Trends Appl. Intell. Syst. (2010) 541–549.
[21] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: review of methods and applications, Expert Syst.Appl. (2016).
[22] 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), Lecture Notes in Computer Science, 3644, Springer, 2005, pp. 878–887.
[23] H. He, Y. Bai, E. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: IEEE International Joint Conference on Computational Intelligence, IJCNN 2008, IEEE, 2008, pp. 1322–1328.
[24] S. Hu, Y. Liang, L. Ma, Y. He, MSMOTE: Improving classification performance when training data is imbalanced, in: 2009 Second International Workshop on Computer Science and Engineering, 2009, pp. 13–17.
[25] T. Imam, K. Ting, J. Kamruzzaman, z-SVM: an SVM for improved classification of imbalanced data, in: Australian conference on artificial intelligence, Springer, 2006, pp. 264–273.
[26] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (5) (2002) 429–449.
[27] T. Jo, N. Japkowicz, Class imbalances versus small disjuncts, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 40–49.
[28] K. Kerdprasop, N. Kerdprasop, Data Preparation Techniques for Improving Rare Class Prediction 2 Intelligent Methods for Predicting, in: Proceedings of the 13th WSEAS international conference on mathematical methods, 2011, pp. 204–209.
[29] S. Kotz, T. Kozubowski, K. Podgorski, The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance, New York: Springer Science & Business Media, 2012.
[30] S. Kullback, R.A. Leibler, On information and sufficiency, Annals of Math. Stat. 22 (1) (1951) 79–86.
[31] W. Liu, N. Chawla, A Robust Decision Tree Algorithm for Imbalanced Data Sets, in: SIAM International Conference on Data Mining, 10, 2010, pp. 766–777.
[32] X. Liu, J. Wu, Z. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Trans. Syst., Man, Cybernetics, Part B (Cybernetics) 39 (2) (2009) 539–550.
[33] 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, Inf. Sci. 250 (2013) 113–141.
[34] J. Luengo, A. Fernández, S. García, F. Herrera, Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling, Soft Comput. 15 (10) (2011) 1909–1936.
[35] L. Lusa, et al., Class prediction for high-dimensional class-imbalanced data, BMC Bioinformatics 11 (1) (2010) 523.
[36] M. Magdon-Ismail, A. Atiya, Density estimation and random variate generation using multilayer networks, IEEE Trans. Neural Netw. 13 (3) (2002) 497–520.
[37] R. Rao, S. Krishnan, R. Niculescu, Data mining for improved cardiac care, ACM SIGKDD Explorations Newsletter 8 (1) (2006) 3–10.
[38] C. Seiffert, T. Khoshgoftaar, J. Hulse, A. Napolitano, RUSBoost: A hybrid approach to alleviating class imbalance, IEEE Trans. Syst., Man, and Cybern 40 (1) (2010) 185–197.
[39] S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane, N. Japkowicz, Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 447–456.
[40] Y. Sun, M. Kamel, A. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognit. 40 (12) (2007) 3358–3378.
[41] J. Vanhoeyveld, D. Martens, Imbalanced classification in sparse and large behaviour datasets, Data Mining Knowl. Discov. (2017) 1–58.
[42] B.C. Wallace, K. Small, C.E. Brodley, T.A. Trikalinos, Class imbalance, redux, in: 2011 IEEE 11th International Conference on Data Mining, IEEE, 2011, pp. 754–763.
[43] G.M. Weiss, Mining with rarity: a unifying framework, ACM SIGKDD Explorations Newsletter 6 (1) (2004) 7–19.
[44] G. Wu, E. Chang, Class-Boundary Alignment for Imbalanced Dataset Learning, in: ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC, 2003, pp. 49–56.
[45] J. Xie, Z. Qiu, The effect of imbalanced data sets on LDA: a theoretical and empirical analysis, Pattern Recognit. 40 (2) (2007) 557–562.