Machine Learning (2021) 110:279–301
https://doi.org/10.1007/s10994-020-05913-4

LoRAS: an oversampling approach for imbalanced datasets

Saptarshi Bej[1] · Narek Davtyan[1] · Markus Wolfien[1] · Mariam Nassar[1] · Olaf Wolkenhauer[1]

Received: 20 August 2019 / Revised: 21 July 2020 / Accepted: 8 September 2020 /
Published online: 12 November 2020

Abstract

The Synthetic Minority Oversampling TEchnique (SMOTE) is widely-used for the analysis of imbalanced datasets. It is known that SMOTE frequently over-generalizes the minority class, leading to misclassifications for the majority class, and effecting the overall balance of the model. In this article, we present an approach that overcomes this limitation of SMOTE, employing Localized Random Affine Shadowsampling (LoRAS) to oversample from an approximated data manifold of the minority class. We benchmarked our algorithm with 14 publicly available imbalanced datasets using three different Machine Learning (ML) algorithms and compared the performance of LoRAS, SMOTE and several SMOTE extensions that share the concept of using convex combinations of minority class data points for oversampling with LoRAS. We observed that LoRAS, on average generates better ML models in terms of F1-Score and Balanced accuracy. Another key observation is that while most of the extensions of SMOTE we have tested, improve the F1-Score with respect to SMOTE on an average, they compromise on the Balanced accuracy of a classification model. LoRAS on the contrary, improves both F1 Score and the Balanced accuracy thus produces better classification models. Moreover, to explain the success of the algorithm, we have constructed a mathematical framework to prove that LoRAS oversampling technique provides a better estimate for the mean of the underlying local data distribution of the minority class data space.

Keywords Imbalanced datasets · Oversampling · Synthetic sample generation · Data augmentation · Manifold learning

Editor: Nathalie Japkowicz.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10994-020-05913-4) contains supplementary material, which is available to authorized users.

✉ Olaf Wolkenhauer
olaf.wolkenhauer@uni-rostock.de
https://www.sbi.uni-rostock.de/

Extended author information available on the last page of the article

1 Introduction

Imbalanced datasets are frequent occurrences in a large spectrum of fields, where Machine Learning (ML) has found its applications, including business, finance and banking as well as bio-medical science. Oversampling approaches are a popular choice to deal with imbalanced datasets (Chawla et al. 2002; Han et al. 2005; Haibo et al. 2008; Bunkhumpornpat et al. 2009; Barua et al. 2014). We here present Localized Randomized Affine Shadowsampling (LoRAS), which produces better ML models for imbalanced datasets, compared to state-of-the art oversampling techniques such as SMOTE and several of its extensions. We use computational analyses and a mathematical proof to demonstrate that drawing samples from a locally approximated data manifold of the minority class can produce balanced classification ML models. We validated the approach with 12 publicly available imbalanced datasets, comparing the performances of several state-of-the-art convex-combination based oversampling techniques with LoRAS. The average performance of LoRAS on all these datasets is better than other oversampling techniques that we investigated. In addition, we have constructed a mathematical framework to prove that LoRAS is a more effective oversampling technique since it provides a better estimate for local mean of the underlying data distribution, in some neighbourhood of the minority class data space.

For imbalanced datasets, the number of instances in one (or more) class(es) is very high (or very low) compared to the other class(es). A class having a large number of instances is called a majority class and one having far fewer instances is called a minority class. This makes it difficult to learn from such datasets using standard ML approaches. Oversampling approaches are often used to counter this problem by generating synthetic samples for the minority class to balance the number of data points for each class. SMOTE is a widely used oversampling technique, which has received various extensions since it was published by Chawla et al. (2002). The key idea behind SMOTE is to randomly sample artificial minority class data points along line segments joining the minority class data points among k of the minority class nearest neighbors of some arbitrary minority class data point. In other words, SMOTE produces oversamples by generating random convex combinations of two close enough data points.

The SMOTE algorithm, however has several limitations for example: it does not consider the distribution of minority classes and latent noise in a data set (Hu et al. 2009). It is known that SMOTE frequently over-generalizes the minority class, leading to misclassifications for the majority class, and effecting the overall balance of the model (Puntumapon and Waiyamai 2012). Several other limitations of SMOTE are mentioned in Blagus and Lusa (2013). To overcome such limitations, several algorithms have been proposed as extensions of SMOTE. Some are focusing on improving the generation of synthetic data by combining SMOTE with other oversampling techniques, including the combination of SMOTE with Tomek-links (Elhassan et al. 2016), particle swarm optimization (Gao et al. 2011; Wang et al. 2014), rough set theory (Ramentol et al. 2012), kernel based approaches (Mathew et al. 2015), Boosting (Chawla et al. 2003), and Bagging (Hanifah et al. 2015). Other approaches choose subsets of the minority class data to generate SMOTE samples or cleverly limit the number of synthetic data generated (Santoso et al. 2017). Some examples are Borderline1/2 SMOTE (Han et al. 2005), ADAptive SYNthetic (ADASYN) (Haibo et al. 2008), Safe Level SMOTE (Bunkhumpornpat et al. 2009), Majority Weighted Minority Oversampling TEchnique (MWMOTE) (Barua et al. 2014), Modified SMOTE (MSMOTE), and Support Vector Machine-SMOTE (SVM-SMOTE) (Suh et al. 2017) (see Table 1) (Hu et al. 2009). Another recent method, G-SMOTE, generates synthetic samples

Table 1 Popular algorithms built on SMOTE

Extension	Description
Borderline1/2 SMOTE (Han et al. 2005)	Identifies borderline samples and applies SMOTE on them
ADASYN (Haibo et al. 2008)	Adaptively changes the weights of different minority samples
SVM-SMOTE (Suh et al. 2017)	Generates new minority samples near borderlines with SVM
Safe-level-SMOTE (Bunkhumpornpat et al. 2009)	Generates data in areas that are completely safe
MWMOTE (Barua et al. 2014)	Identifies and weighs ambiguous minority class samples

in a geometric region of the input space, around each selected minority instance (Douzas and Bacao 2019). Voronoi diagrams have also been used in recent research for improving classification tasks for imbalanced datasets. Because of properties inherent to Voronoi diagrams, a newly proposed algorithm V-synth identifies exclusive regions of feature space where it is ideal to create synthetic minority samples (Young et al. 2015; Carvalho and Prati 2018).

Related research and novelty A more recent trend in the research on imbalanced datasets is to generate synthetic samples, aiming to approximate the latent data manifold of the minority class data space. In Bellinger et al. (2018), a general framework for manifold-based oversampling, especially for high dimensional datasets, is proposed for synthetic oversampling. The method has been successfully applied in Bellinger et al. (2016) to deal with gamma-ray spectra classification. It produces a synthetic set S of n instances in the manifold-space by randomly sampling n instances from the PCA-transformed reduced data space. In order to produce unique samples on the manifold, they apply i.i.d. additive Gaussian noise to each sampled instance prior to adding it to the synthetic set S, controlling the distribution of the noise through the Gaussian distribution parameters. The synthetic Gaussian instances are then mapped back to the feature space to produce the final synthetic samples (Bellinger et al. 2018). Another scheme, using auto-encoders to oversample from an approximated manifold, has also been discussed in Bellinger et al. (2018). This approach selects random minority class samples by adding Gaussian noise to them, and using the auto-encoder framework first maps them non-orthogonally off the manifold and then maps them back orthogonally on the manifold (Bellinger et al. 2018). It remains unclear from this research how the approach would perform in terms of improving F1-Scores of imbalanced classification models as it focuses on relative improvement in the Area Under the (ROC) Curve (AUC) as a performance measure. According to Saito and Rehmsmeier (2015), AUC of the Receiver Operating Characteristic Curve (ROC) curve might not be informative enough for imbalanced datasets. This issue has also been addressed in Davis and Goadrich (2006). Unlike the work of Bellinger et al. (2018) LoRAS relies on locally approximating the manifold by generating random convex combination of noisy minority class data points. Our oversampling strategy LoRAS, rather aims at improving the precision-recall balance (F1-Score) and class wise average accuracy (Balanced accuracy) of the ML models used. The F1-Score can measure how well the classification model handled the minority class classification, whereas Balanced accuracy provides us with a measure of how both majority and minority classes were handled by the classification model. Thus, these two measures together can give us a holistic understanding of a classifier performance on a dataset.

Notably, in the pre-SMOTE era of research, related to oversampling there has been works aiming to enrich minority classes of imbalanced datasets by adding Gaussian noise (Lee 2000) and using the noisy data itself, as oversampled data. The strategy of generating oversamples with convex combinations of minority class samples is also well known, SMOTE itself being an example of such a strategy. Our oversampling strategy LoRAS leverages from a combination of these two strategies. Unlike Lee (2000), we generate Gaussian noise in small neighbourhoods around the minority class samples and create our final synthetic data with convex combinations of multiple noisy data points (shadowsamples) as opposed to SMOTE based strategies, that consider combination of only two minority class data points. Adding the shadowsamples allows LoRAS to produce a better estimate for local mean of the latent minority class data distribution.

We also provide a mathematical framework to show that convex combinations of multiple shadowsamples can provide a proper estimate for the local mean of a neighbourhood in the minority class data space. To be specific, an LoRAS oversample is an unbiased estimator of the mean of the underlying local probability distribution, followed by a minority class sample (assuming that it is some random variable) such that the variance of this estimator is significantly less than that of a SMOTE generated oversample, which is also an unbiased estimator of the mean of the underlying local probability distribution, followed by a minority class sample. In addition to this, LoRAS provides an option of choosing the neighbourhood of a minority class data point by performing prior manifold learning over the minority class using t-Stochastic Neighbourhood Embedding (t-SNE) (van der Maaten and Hinton 2008). t-SNE is a state-of the art algorithm used for dimension reduction maintaining the underlying manifold structure in a sense that, in a lower dimension t-SNE can cluster points, that are close enough in the latent high dimensional manifold. It uses a symmetric version of the cost function used for it's predecessor technique Stochastic Neighbourhood Embedding (SNE) and uses a Student-t distribution rather than a Gaussian to compute the similarity between two points in the low-dimensional space. t-SNE employs a heavy-tailed distribution in the low-dimensional space to alleviate both the crowding problem and the optimization problems of SNE (van der Maaten and Hinton 2008; Hinton and Roweis 2003).

Till date there are at least eighty five extension models built on SMOTE (Kovács 2019). Considering a large number of benchmark datasets explored in our study, it was necessary to shortlist certain oversampling algorithms for a comparative study. We found quite a few studies that have applied or explored SMOTE and extension of SMOTE such as Borderline1/2 SMOTE models, ADASYN, and SVM-SMOTE (Suh et al. 2017; Ah-Pine and Soriano-Morales 2016; Adiwijaya and Saonard 2017; Chiamanusorn and Sinapiromsaran 2017; Wang et al. 2014; Le et al. 2019). Moreover all these oversampling strategies are focused on oversampling from the convex hull of small neighbourhoods in the minority class data space, a similarity that they share with our proposed approach. Considering these factors, we choose to focus on these five oversampling strategies for a comparative study with our oversampling technique LoRAS.

2 LoRAS: localized randomized affine shadowsampling

In this section we discuss our strategy to approximate the data manifold, given a dataset. A typical dataset for a supervised ML problem consists of a set of *features* $F = \{f_1, f_2, \dots\}$, that are used to characterize patterns in the data and a set of *labels* or ground truth. Ideally, the number of instances or samples should be significantly greater than the number of features. In order to maintain the mathematical rigor of our strategy we propose the following definition for a *small dataset*.

Definition 1 Consider a class or the whole dataset with n samples and $|F|$ features. If $\log_{10}(\frac{n}{|F|}) < 1$, then we call the dataset, a *small dataset*.

The LoRAS algorithm is designed to learn from a dataset by approximating the underlying data manifold. Assuming that F is the best possible set of features to represent the data and all features are equally important, we can think of a data oversampling model to be a function $g : \prod_{i=1}^{l} R^{|F|} \to R^{|F|}$, that is, g uses l parent data points (each with $|F|$ features) to produce an oversampled data point in $R^{|F|}$.

Definition 2 We define a *random affine combination* of some arbitrary vectors as the affine linear combination of those vectors, such that the coefficients of the linear combination are chosen randomly. Formally, a vector v, $v = \alpha_1 u_1 + \dots + \alpha_n u_m$, is a random affine combination of vectors $u_1, \dots, u_m$, ($u_j \in R^{|F|}$) if $\alpha_1 + \dots + \alpha_m = 1$, $\alpha_j \in R^+$ and $\alpha_1, \dots, \alpha_m$ are the coefficients of the affine combination chosen randomly from a Dirichlet distribution.

The simplest way of augmenting a data point would be to take the average (or random affine combination with positive coefficients as defined in Definition 2) of two data points as an augmented data point. But, when we have $|F|$ features, we can assume that the hypothetical manifold on which our data lies is $|F|$-dimensional. An $|F|$-dimensional manifold can be locally approximated by a collection of $(|F| - 1)$-dimensional planes.

Given $|F|$ sample points we could exactly derive the equation of an unique $(|F| - 1)$ -dimensional plane containing these $|F|$ sample points. Note that, a small neighbourhood of a dataset can itself be considered as a small dataset. A small neighbourhood of k points around a data point in a dataset, given sufficiently small k, satisfies Definition 1, that is k and $|F|$ satisfies, $\log_{10}(\frac{k}{|F|}) < 1$. Thus, considering k to be sufficiently small we can assume that this small neighbourhood is a small dataset. To enrich this small dataset, we create *shadow data points* or *shadowsamples* from our k parent data points in the minority class data point neighbourhood. Each shadow data point is generated by adding noise from a normal distribution, $\mathcal{N}(0, h(\sigma_f))$ for all features $f \in F$, where $h(\sigma_f)$ is some function of the sample variance σ_f for the feature f. For each of the k data points we can generate m shadow data points such that, $k \times m \gg |F|$. Now it is possible for us to choose $|F|$ shadow data points from the $k \times m$ shadow data points even if $k < |F|$. We choose $|F|$ shadow data points as follows: we first choose a random parent data point p and then restrict the domain of choice to the shadowsamples generated by the parent data points in N_k^p.

For high dimensional datasets, choosing k-nearest neighbours of data point using simple Euclidean, Manhattan or general Minkowski distance measures can be misleading in terms of approximating the latent data manifold. To avoid this, we propose to adopt a manifold learning based strategy. Before choosing the k-nearest neighbours of a data point, we perform a dimension reduction on the data points of the minority class using the well-known dimension reduction and manifold learning technique t-SNE (van der Maaten and Hinton 2008). Once we have a two dimensional t-embedding of the minority class data, we choose the k-nearest neighbours of a particular data point consistent to its k-nearest neighbours (measured as per usual distance metrics) in the 2-dimensional t-SNE embedding of the minority class.

Once we choose our neighbourhood and generate the shadowsamples, we take a random affine combination with positive co-efficients (Convex combination) of the $|F|$ chosen shadowsamples to create one augmented Localized Random Affine Shadowsample or a LoRAS sample as defined in Definition 2. Considering the arbitrary low variance that we can choose for the Normal distribution from which we draw our shadowsamples, we assume that our shadowsamples lie in the latent data manifold itself. It is a practical assumption, considering the stochastic factors leading to small measurement errors. Now, there exists an unique $(|F| - 1)$-dimensional plane, that contains the $|F|$ shadowsamples, which we assume to be an approximation of the latent data manifold in that small neighbourhood. Thus, a LoRAS sample is an artificially generated sample drawn from an $(|F| - 1)$-dimensional plane, which locally approximates the underlying hypothetical $|F|$-dimensional data manifold. It is worth mentioning here, that the effective number of features in a dataset is often less than $|F|$. In high dimensional data there are often correlated features or features with low variance. Thus, for practical use of LoRAS one might consider generating convex combinations of effective number of features which might be less than $|F|$.

Algorithm 1: Localized Random Affine Shadowsample (LoRAS) Oversampling

```
Inputs:
  C_maj:        Majority class parent data points
  C_min:        Minority class parent data points
Parameters:
  k:            Number of nearest neighbors to be considered per parent data point
                (default value : 30 if $|C_{\min}| >= 100$, 5 otherwise)
  |Sp|:         Number of generated shadowsamples per parent data point
                $\left(\text{default value : max}\left(\left\lceil \frac{2|F|}{\mathrm{k}} \right\rceil, 40\right)\right)$
  L_σ:          List of standard deviations for normal distributions for adding noise to each feature
                (default value : $[0.005, \ldots, 0.005]$)
  N_aff:        Number of shadow points to be chosen for a random affine combination
                (default value : $|F|$)
  N_gen:        Number of generated LoRAS points for each nearest neighbors group
                $\left(\text{default value : } \frac{|C_{\mathrm{maj}}|-|C_{\mathrm{min}}|}{|C_{\mathrm{min}}|}\right)$
  embedding:    Type of Embedding used to choose minority class neighbourhood (regular or t-embedding)
                (default value : 'regular' )
  perplexity:   Perplexity of t-embedding (applicable only if embedding='t-embedding')
                (default value : 30)
Constraint:
  $\mathtt{N_{aff}} < k * |\mathtt{S_p}|$
Initialize loras_set as an empty list
For each minority class parent data point p in C_min do
    neighborhood ← calculate k-nearest neighbors of p, as per selected Embedding parameter and append p
    Initialize neighborhood_shadow_sample as an empty list
    For each parent data point q in neighborhood do
        shadow_points ← draw |Sp| shadowsamples for q drawing noises from normal distributions with
          corresponding standard deviations L_σ containing elements for every feature
        Append shadow_points to neighborhood_shadow_sample
    Repeat
        selected_points ← select N_aff random shadow points from neighborhood_shadow_sample
        affine_weights ← create and normalize random weights for selected_points
        generated_LoRAS_sample_point ← selected_points · affine_weights
        Append generated_LoRAS_sample_point to loras_set
    Until N_gen resulting points are created;
Return resulting set of generated LoRAS data points as loras_set
```

In this article, all imbalanced classification problems that we deal with are binary classification problems. For such a problem, there is a minority class C_{min} containing a relatively less number of samples compared to a majority class C_{maj}. We can thus consider the minority class as a small dataset and use the LoRAS algorithm to oversample. For every data point p we can denote a set of shadowsamples generated from p as S_p. In practice, one can also choose $2 \leq N_{\mathrm{aff}} \leq |F|$ shadowsamples for an affine combination and choose a desired number of oversampled points N_{gen} to be generated using the algorithm. We can look at LoRAS as an oversampling algorithm as described in Algorithm 1.

The LoRAS algorithm thus described, can be used for oversampling of minority classes in case of highly imbalanced datasets. Note that the input variables for our algorithm are: number of nearest neighbors per sample k, number of generated shadow points per parent data point $|\mathrm{S_p}|$, list of standard deviations for normal distributions for adding noise to every feature and thus generating the shadowsamples $\mathtt{L}_\sigma$, number of shadowsamples to be chosen for affine combinations $\mathtt{N}_{\mathrm{aff}}$, number of generated points for each nearest neighbors group $\mathtt{N}_{\mathrm{gen}}$ and embedding strategy embedding. There is a conditional input variable perplexity which takes a positive numerical value if one chooses a t-embedding. The perplexity parameter of the t-SNE algorithm is quite crucial. The perplexity parameter can influence the t-Embedding calculated by the t-SNE algorithm. There have been several studies that address the issue on finding a right perplexity parameter for a given problem (Kobak and Berens 2019). That is why, we recommend careful choice of this parameter in order to leverage more from our algorithm. Another important parameter of our algorithm is the $\mathtt{N}_{\text{aff}}$. For this parameters an ideal choice would be the number of effective features in a dataset since this number would be a reasonable approximation to the dimension of the underlying data manifold. One could employ a feature selection technique to find out a good estimate for this. A simple random grid search is also very helpful to get reasonably good estimates of these parameters. We have mentioned all the default values of the LoRAS parameters in Algorithm 1, showing the pseudocode for the LoRAS algorithm. As an output, our algorithm generates a LoRAS dataset for the oversampled minority class, which can be subsequently used to train a ML model (Fig. 1).

3 Case studies

For testing the potential of LoRAS as an oversampling approach, we designed benchmarking experiments with a total of 14 datasets which are either highly imbalanced, high dimensional or with a small number of data points. With this number of diverse case studies we should have a comprehensive idea of the advantages of LoRAS over the other oversampling algorithms of our interest.

3.1 Datasets used for validation

Here we provide a brief description of the datasets and the sources that we have used for our studies.

Scikit-learn imbalanced benchmark datasets The `imblearn.datasets` package is complementing the `sklearn.datasets` package. It provides 27 pre-processed datasets, which are imbalanced. The datasets span a large range of real-world problems from several fields such as business, computer science, biology, medicine, and technology. This collection of datasets was proposed in the `imblearn.datasets` python library by Lemaître et al. (2017) and benchmarked by Ding (2011). Many of these datasets have been used in various research articles on oversampling approaches (Ding 2011; Saez et al. 2016). A statistically reliable benchmarking analysis of all 27 datasets in a stratified cross validation framework involves a lot of computational effort. We thus choose 11 datasets out of these two depending on two criteria:

- *Highly imbalanced* We choose datasets with imbalance ratio more than 25:1. This category includes abalone_19, letter_image, mammography, ozone_level, webpage, wine_quality, yeast_me2 datasets.
- *High dimensional* We choose the datasets with more than 100 features. This category includes arrhythmia, isolet, scene, webpage and yeast_ml8.

Note that the `webpage` dataset is common in both the criteria, giving us a total of 11 datasets. We choose these two categories because they are of special interest in research related to imbalanced datasets and have received extensive attention in this research area (Anand et al. 2010; Hooda et al. 2018; Jing et al. 2019; Blagus and Lusa 2013).

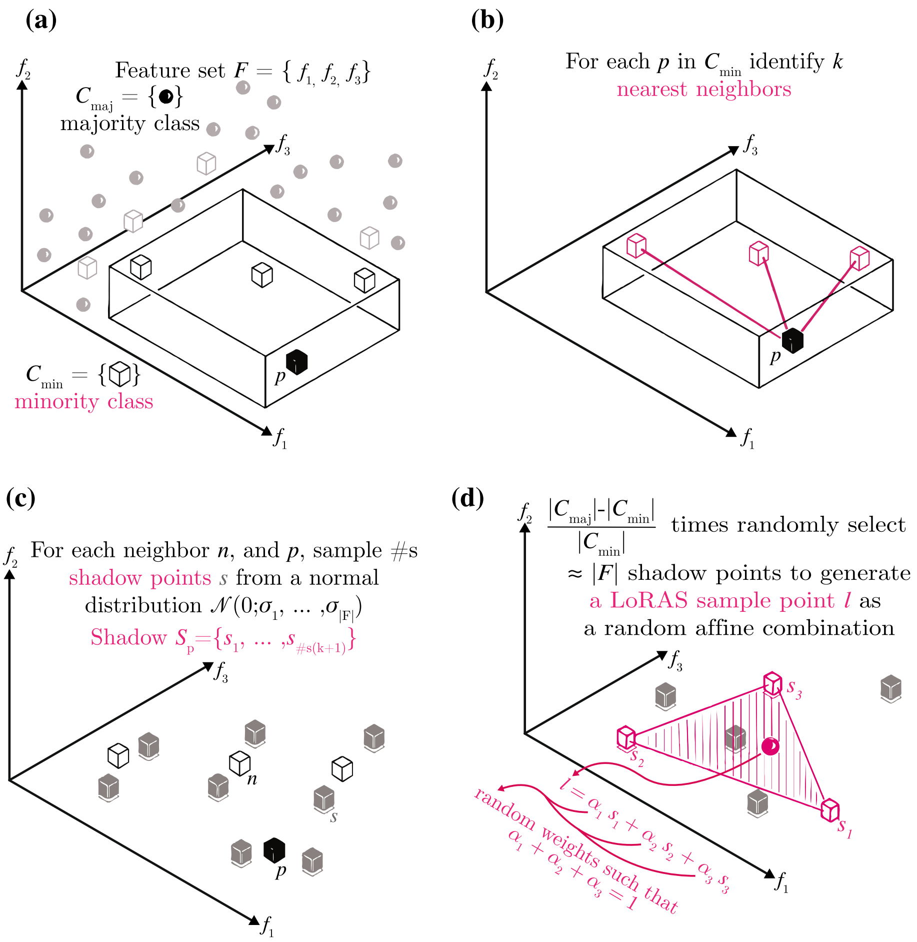
(a)
Feature set F = { f1, f2, f3}
C maj = {●}
majority class
C min = {⬡}
minority class
p
f1
f2
f3
(b)
For each p in C min identify k
nearest neighbors
p
f1
f2
f3
(c)
For each neighbor n, and p, sample #s
shadow points s from a normal
distribution 𝒩(0;σ1, ... ,σ|F|)
Shadow Sp={s1, ... ,s#s(k+1)}
n
s
p
f1
f2
f3
(d)
(|Cmaj|-|Cmin|)/|Cmin| times randomly select
≈ |F| shadow points to generate
a LoRAS sample point l as
a random affine combination
s1
s2
s3
l = α1 s1 + α2 s2 + α3 s3
random weights such that
α1 + α2 + α3 = 1
f1
f2
f3

Fig. 1 Visualization of the workflow demonstrating a step-by-step explanation for LoRAS oversampling. **a** Here, we show the parent data points of the minority class points $C_{\min}$. For a data point p we choose three of the closest neighbors (using knn) to build a neighborhood of p, depicted as the box, **b** extracting the four data points in the closest neighborhood of p (including p), **c** drawing shadow points from a normal distribution, centered at these parent data point n, **d** we randomly choose three shadow points at a time to obtain a random affine combination of them (spanning a triangle). We finally generate a novel LoRAS sample point from the neighborhood of a single data point p

Credit card fraud detection dataset We obtain the description of this dataset from the website. https://www.kaggle.com/mlg-ulb/creditcardfraud. "The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where there are 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.00172 percent of all transactions. The dataset contains only numerical input variables, which are the result of a PCA transformation. Feature variables $f_1, \dots, f_{28}$ are the principal components obtained with PCA, the only features that have not been transformed with PCA are the 'Time' and 'Amount'. The feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature ‘Amount’ consists of the transaction amount. The labels are encoded in the ‘Class’ variable, which is the response variable and takes value 1 in case of fraud and 0 otherwise” (Pozzolo et al. 2017).

Small datasets We were also interested to check the performance of LoRAS on small datasets. We obtained two such datasets: ar1, ar3. Both of these datasets have very few data points and less than 10 points in the minority class.

Thus, in total we benchmark our oversampling algorithms against the existing algorithms on a total of 14 datasets. We provide relevant statistics on these datasets in Table 2

3.2 Methodology

For every dataset we have analyzed, we used a consistent workflow. Given a dataset, for every machine learning model, we judge the model performances based on a 5 × 10-fold stratified cross validation framework. However, for the two small datasets ar1 and ar3 we use a 5 × 3-fold stratified cross validation framework, since there are less than 10 samples in the minority class. First we randomly scuffle the dataset. For a given dataset, we first split the dataset into tenfolds, each one distinct from the other maintaining the imbalance ratio for every fold. We then train the machine learning models on the dataset without any oversampling with tenfold cross validation. This means that we train and test the model 10 times, each time considering a fold as a test fold and rest ninefolds as training folds. However, while training the ML models with oversampled data, we oversample only on the training folds and leave the test fold as they are for each training session. For each dataset we repeat the whole process five times to avoid the stochastic effects as much as possible.

For the oversampling algorithms, for a given dataset, we chose the same neighbourhood size for every oversampling model. If there were less than 100 data points in the minority class the neighbourhood size was chosen to be 5. Otherwise we chose a neighbourhood

Table 2 Table showing some statistics for the datasets we study in this article

Dataset	Imbalance ratio	Number of samples	Number of features
Abalone_19	**130:1**	4177	10
Arrythmia	17:1	452	**278**
Isolet	12:1	7797	**617**
Letter-img	**26:1**	20,000	16
Mammography	**42:1**	11183	6
Scene	13:1	2407	**294**
Ozone_level	**34:1**	2536	72
Webpage	**33:1**	34,780	**300**
Wine-quality	**26:1**	4898	11
Yeast-me2	**28:1**	1484	8
Yeast-ml8	13:1	2417	**103**
Credit fraud	**577:1**	284,807	28
ar1	12.44:1	**121**	30
ar3	6.8:1	**63**	30

For each dataset, we mark in bold the feature of the dataset that led us to its choice for our study size of 30. Given a large number of datasets we are analyzing, we did not customize this for every dataset and rather chose to stick to the above mentioned general rule. For LoRAS oversampling however, we performed a preliminary study to find out customized parameter values for every dataset, since the LoRAS algorithm is highly parametrized in nature. We tried several combinations of parameters $\mathtt{N}_{\mathrm{aff}}$, `embedding` and `perplexity` employing random grid search. For our initial study involving the parameter optimization of LoRAS, given a dataset, we performed a simple train-test split of the dataset (1:1 train-test split ratio), and then applied LoRAS with parameter grids on the training data to oversample and test the classifier performances on the test data. The training set is kept relatively small, so that the classifier does not gain much experience on the data while parameter estimation and gets prone to overfitting. This study was kept completely independent from our main cross-validation based results so that the samples from the test sets of our cross validation have minimum effect on parameter tuning. For parameter $\mathtt{N}_{\mathrm{aff}}$ the grid interval is $[2, |F|]$, $|F|$ being the number of features. We choose five numbers while forming a search grid from this interval. Three of them are randomly chosen and the numbers 2 and $|F|$ are always included in this set of 5 numbers. For parameter `embedding` we the grid values are the two possible entries that the parameter adopts. For the `perplexity` parameter, we used grid values [0.01, 0.1, 1, 10, 30, 100].

We emphasize here, that for all the algorithms including LoRAS, for a given dataset, we keep the neighbourhood size for every oversampling model fixed. For every oversampling model that we considered, the neighbourhood size for the oversampling model is the parameter that the model is highly sensitive to, since it contributes the most in determining the distribution of the oversampled minority class. For LoRAS, there are three (out of seven parameters in total) parameters designed to better model/approximate the minority class data manifold (for example: the ones involving the t-SNE on the minority class), which are tuned to show the applicability of manifold approximation to improve convex combination based oversampling. However, as suggested, we keep all parameters related to the original distribution of the minority class, for all oversampling models fixed for all comparisons.

However, considering the philosophy of LoRAS and a comparatively large number of parameters it use, we take liberty to tune the other parameters for LoRAS, since the other parameters are the key to a proper approximation or modelling of the minority class data manifold, which we argue to be the key factor behind the success of LoRAS.

For LoRAS oversampling every dataset we use an unique value for $\mathtt{N}_{\mathrm{aff}}$ as presented in Table 3. For individual ML models we use different settings for the LoRAS parameters `embedding` and `perplexity` which we mention explicitly in our supplementary materials while presenting the results for each ML model for each dataset. To ensure fairness of comparison, we oversampled such that the total number of augmented samples generated from the minority class was as close as possible to the number of samples in the majority class as allowed by each oversampling algorithm. Speaking of other parameters of the LoRAS algorithm, for $\mathtt{L}_{\sigma}$, we chose a list consisting of a constant value of .005 for each dataset and for the parameter $\mathtt{N}_{gen}$ we chose the value as: $\frac{|C_{\mathrm{maj}}|-|C_{\mathrm{min}}|}{|C_{\mathrm{min}}|}$. We provide a detailed list of parameter settings used by us for the oversampling algorithms in Table 3.

To choose ML models for our study we first did a pilot study with ML classifiers such as k-nearest neighbors (knn), Support Vector Machine (svm) (linear kernel), Logistic regression (lr), Random forest (rf), and Adaboost (ab). As inferred in Blagus and Lusa (2013) we found that knn was quite effective for the datasets we used. We also noticed that lr and svm performed better compared to rf and ab in most cases. We thus

Table 3 In this table we present the details of parameter settings for the oversampling algorithms used by us for our experiment

Dataset	Minority samples	Oversampling nbd	LoRAS $\mathtt{N}_{\text{aff}}$
Abalone19	32	5	10
Arrythmia	25	5	100
Isolet	600	30	179
Letter-img	734	30	16
Mammography	260	30	6
Scene	177	30	2
Ozone_level	73	5	10
Webpage	981	30	94
Wine-quality	183	30	2
Yeast-me2	51	5	2
Yeast-ml8	178	30	3
Credit fraud	492	30	30
ar1	9	3	30
ar3	8	3	10

The second column is the size of the oversampling neighbourhood and we have chosen the same size for all the oversampling models for each dataset in our analysis

The last three columns are specific to LoRAS parameters

chose knn, svm and lr for our final studies. We used lbfgs solver for our logistic regression model and a linear kernel for our svm models. For our knn models, we choose 10 nearest neighbours for our prediction if there are less than 100 samples in the minority class and 30 nearest neighbours otherwise. For ‘arrhythmia’, ‘abalone-19’, ‘ar1’ and ‘ar3’ however we use only 5 nearest neighbours for the knn model since it has only 25, 32, 9 and 8 minority class samples respectively. We choose this parameter to be consistent to the neighbourhood size of the oversampling models, since the neighbourhood size directly influences the distribution of the training data and hence the model performance.

In our analysis we take special notice of the credit card fraud detection dataset. This dataset is not included in the `imblearn.datasets` Python library. However, the main reason why we want to pay a special attention to this dataset is that, it is by far the most imbalanced publicly available dataset that we have come across. The extreme imbalance ratio of 577:1 is incomparable to any of the datasets in `imblearn.datasets`. Also, this dataset has received special attention of researchers attempting to use ML in Credit fraud detection (Varmedja et al. 2019). In this article we see that lr and rf have good prediction accuracies on the dataset. Thus we chose these two ML models for the credit fraud dataset. Varmedja et al. (2019) has also not provided cross validated analysis of their models, while our models have been trained and tested with the usual tenfold cross validation framework as discussed before. Also, for two small datasets with a critically small minority class, we used only knn and lr classifiers, with parameter settings as specified before. The reason is, for all the 12 other datasets, svm did not stand out to be the best performer in terms of F1-Score in any of them.

For computational coding, we used the `scikit-learn (V 0.21.2)`, `numpy (V 1.16.4)`, `pandas (V 0.24.2)`, and `matplotlib (V 3.1.0)` libraries in `Python (V 3.7.4).`

4 Results

For imbalanced datasets there are more meaningful performance measures than *Accuracy*, including *Sensitivity* or *Recall*, *Precision*, and *F1-Score* (*F-Measure*), and *Balanced accuracy* that can all be derived from the *Confusion Matrix*, generated while testing the model. For a given class, the different combinations of recall and precision have the following meanings:

- High Precision & High Recall: The model handled the classification task properly
- High Precision & Low Recall: The model cannot classify the data points of the particular class properly, but is highly reliable when it does so
- Low Precision & High Recall: The model classifies the data points of the particular class well, but misclassifies high number of data points from other classes as the class in consideration
- Low Precision & Low Recall: The model handled the classification task poorly

F1-Score, calculated as the harmonic mean of precision and recall and, therefore, balances a model in terms of precision and recall. These measures have been defined and discussed thoroughly by Elrahman and Abraham (2013). Balanced accuracy is the mean of the individual class accuracies and in this context, it is more informative than the usual accuracy score. High Balanced accuracy ensures that the ML algorithm learns adequately for each individual class.

In our experiments we have noticed an interesting behaviour of oversampling models in terms of their average F1-Score and Balanced accuracy. Once we present our experiment results, we will discuss why considering F1-Score and Balanced accuracy can give us a clearer idea about model performances. We will use the above mentioned performance measures wherever applicable in this article.

Selected model performances for all datasets We provide the detailed results of our experiments for all machine learning models as supplementary material. To be precise, for every combination of datasets, ML models and oversampling strategies we provide the mean and variance of the tenfold cross validation process over 5 repetitions. For judging the performance of the oversampling models we follow the following scheme:

- First, for a given dataset, we choose the ML model trained on that dataset that provides the highest average F1-Score over all the oversampling models and training without oversampling. The F1-Score reflects the balance between precision and recall and considered as a reliable metric for imbalanced classification task.
- We then consider the Balanced accuracy and F1- score of the chosen model as an evaluation of how well the oversampling model performs on the considered dataset. Following this evaluation scheme we present our results in Table 4.

Calculating average performances over all datasets, LoRAS has the best Balanced accuracy and F1-Score. As expected, SMOTE improved Balanced accuracy compared to model training without any oversampling. Surprisingly, it lags behind in F1-Score, for quite a few datasets with high baseline F1-Score such as letter_image, isolet, mammography, webpage and credit fraud. Interestingly, the oversampling approaches SVM-SMOTE and Borderline1 SMOTE also improved the average F1-Score compared to SMOTE, but compromised

Table 4 Table showing balanced accuracy/F1-score for several oversampling strategies (Baseline, SMOTE, SVM-SMOTE, Borderline1 SMOTE, Borderline2 SMOTE, ADASYN and LoRAS column-wise respectively) for all 14 datasets of interest for ML learning models producing best average F1 score over all oversampling strategies and baseline training for respective datasets

Dataset	ML	Baseline	SMOTE	Bl-1	Bl-2	SVM	ADASYN	LoRAS
Abalone19	knn	.534/.000	.644/.054	.552/.044	.552/.044	.556/.045	.571/.055	**.675/.059**
Arrythmia	lr	.679/.37	.666/.345	.672/.352	**.709**/.307	.679/.350	.667/.362	.694/**.380**
Isolet	lr	.900/**.826**	.898/.806	.899/.802	.906/.693	**.911**/.799	.898/.806	.904/.809
Letter-img	knn	.927/**.915**	.988/.781	.984/.768	.977/.687	.986/.724	.985/.732	**.989**/.833
Mammography	knn	.703/**.549**	**.911**/.413	.909/.414	.899/.326	.909/.467	.905/.353	.896/.511
Scene	lr	.551/.168	.616/.222	.619/.230	**.620**/.223	.616/**.235**	**.620**/.224	.616/.226
Ozone_level	lr	.517/.062	.800/.190	.777/.212	.781/.183	.738/**.215**	.803/.192	**.809**/.207
Webpage	knn	.805/**.711**	.906/.267	.901/.274	.903/.287	.904/.267	.903/.264	**.923**/.613
Wine-quality	lr	.517/.067	.718/.179	.715/.182	.711/.171	.712/**.216**	.721/.180	**.734**/.197
Yeast-ml8	knn	.500/.000	.558/.152	.561/.153	.563/.153	**.572/.158**	.558/.151	.559/.152
Yeast-me2	knn	.523/.074	.834/.331	.797/.373	.79/.304	.785/**.388**	.825/.315	**.842**/.354
Credit fraud	rf	.669/.775	.922/.359	.919/.645	.919/.556	.913/.741	.**.923**/.350	.904/**.820**
ar1	knn	.340/.071	.561/.306	.549/.298	**.594**/.338	.550/.324	.583/.320	.563/**.349**
ar3	rf	.634/.259	.810/.531	.809/**.584**	.819/.582	.755/.479	781/.457	**.823**/.563
Average	–	.636/.338	.775/.352	.764/.380	.771/.346	.759/.386	.777/.340	**.783/.433**
Average rank	–	6.53/4.64	3.57/4.75	4.35/3.46	3.39/5.10	4.07/3.17	3.5/4.71	**2.57/2.14**

The bold values in the table denote the oversampling model that leads to the best classifier performance for each dataset

for a lower Balanced accuracy. On the other hand, applying ADASYN increased the Balanced accuracy compared to SMOTE, but again compromises on the F1-Score. In contrast, our LoRAS approach produces the best Balanced accuracy on average by maintaining the highest average F1-Score among all oversampling techniques. We want to emphasize that, even considering stochastic factors, LoRAS can improve both the Balanced accuracy and F1-Score of ML models significantly compared to SMOTE, which makes it unique.

Datasets with high imbalance ratio To verify the performance of LoRAS on highly imbalanced datasets we present average of the selected model performances for the datasets with highest imbalance ratios (among the ones we have tested) in Table 5.

From our results we observe that LoRAS oversampling can significantly improve model performances for highly imbalanced datasets. LoRAS provides the highest F1-Score and Balanced accuracy among all the oversampling models. The results here show similar properties for SMOTE, Borderline-1 SMOTE, SVM SMOTE, ADASYN and LoRAS

Table 5 Table showing the average balanced accuracy/F1-score of the selected models for datasets with the highest imbalance ratios and high dimensional datasets separately

Average	Baseline	SMOTE	Bl-1	Bl-2	SVM	ADASYN	LoRAS
Highly imbalanced datasets	.662/.381	.840/.321	.819/.364	.817/.319	.814/.382	.841/.305	**.846/.449**
High dimensional datasets	.687/.415	.728/.358	.730/.362	**.740**/.332	.736/.361	.729/.361	.739/**.436**

The bold values in the table denote the oversampling model that leads to the best classifier performance for each dataset as discussed before. Note that, for the credit fraud dataset, which is the most imbalanced among all, LoRAS has significant success over the other oversampling models in terms of Balanced accuracy. For the webpage dataset as well it improves the Balanced accuracy significantly, compromising minimally on the baseline F1-Score. The same trend follows for the letter_image dataset. Notably, these three datasets have the highest number of overall samples as well, implying that with more data LoRAS can significantly outperform compared convex combination based oversampling models.

High dimensional datasets It is also of interest to us to check how LoRAS performs on high dimensional datasets. We therefore select five datasets with highest number of features among our tested datasets and present the performances of the selected ML methods in Table 5 From our results for high dimensional datasets, we observe that LoRAS produces the best F1-Score and second best Balanced accuracy on average among all oversampling models as Borderline-2 SMOTE beats LoRAS marginally. SMOTE improves both Balanced accuracy with respect to the baseline score here. Borderline-1 SMOTE and SVM SMOTE further increases SMOTE's performance both in terms of F1-Score and Balanced accuracy. Borderline-2 SMOTE, although improves the Balanced accuracy of SMOTE compromises on the F1-Score. Note that, even excluding the webpage dataset, where LoRAS has an overwhelming success, LoRAS still has the best average F1-Score and third highest Balanced accuracy marginally behind SVM-SMOTE and Borederline-2 SMOTE. We thus conclude, that for high dimensional datasets LoRAS can outperform the compared oversampling models in terms of F1-Score, while compromising marginally for Balanced accuracy.

Small datasets For the two small datasets (with less than 10 samples in minority class) we have explored, we observed that LoRAS performs reasonably well. For the 'ar1', LoRAS produces the best F1-Score and third best Balanced accuracy.For the 'ar2' dataset LoRAS produces the best Balanced accuracy and the third best F1-Score. Note that LoRAS performs quite well for the 'abalone' and 'arrhythmia' datasets, which also have a small number of data points in the minority class.

Statistical analysis Following Tarawneh et al. (2020), we use the Wilcoxon's signed rank test to compare LoRAS against the other convex-combination based oversampling algorithms, in terms of both the performance measures we have used: F1-Score and Balanced accuracy. Tarawneh et al. (2020) chose this test for comparative studies since it is safer than parametric tests as it refrains from assuming homogeneity or normal distribution of data. Therefore, it can be applied to any classifier evaluation measure. Tarawneh et al. (2020) further confirms: 'The Wilcoxon test aims to find if a null hypothesis is true or not. The null hypothesis H_0 assumes that there is no significant difference between the classification results (observations) obtained from two different methods. We assume that the null hypothesis is rejected if the p-value of the Wilcoxon test is less than $\alpha = 0.05$'(Tarawneh et al. 2020).

From Table 6 we observe that the p-values for all the paired tests are less than 0.05 for the F1-Score, and therefore, the H_0 is rejected for all the paired tests in case of the F1-Score. Thus, the F1-Scores LoRAS produce have a big enough difference compared

Table 6 Table showing p-values for comparison of LoRAS against the other oversampling algorithms, in terms of both the performance measures we have used: F1-score and balanced accuracy

Measure	Baseline	SMOTE	Bl-1	Bl-2	SVM	ADASYN
F1-Score	0.0303	0.0009	0.0479	0.0035	0.0479	0.0009
Balanced accuracy	0.0009	0.0354	0.0258	0.5095	0.0382	0.1670

to the other compared algorithms, to be statistically significant. For Balanced accuracy, the algorithms Borderline-2 SMOTE and ADASYN do not show significant statistical difference to LoRAS. However, since F1-Score is a more reliable and widely used metric for imbalanced datasets, we conclude that overall results generated by LoRAS are significantly different from the compared oversampling algorithms.

Tarawneh et al. (2020) also remarks that the p-value alone is informative enough and does not provide information about the relationship strength between variables. The p-values do not reveal whether the results are significantly different in favour of LoRAS or against LoRAS. For that following Tarawneh et al. (2020) we use the metrics W_+, W_- and R. These are calculated using the following steps:

- For each data pair (involving LoRAS and some other oversampling algorithm) of model predictions , the difference between both predictions is calculated and stored in a vector D, excluding the zero difference values.
- The signs of the difference is recorded in a sign vector S.
- The entries in $|D|$ are ranked, forming a vector R'. In case of tied ranks, an average ranking scheme is adopted. This means, after ranking the entries of $|D|$ are ranked using integers and then, in case of tied entries the average of the integer ranks are considered as the average rank for all the respective tied entries with a specific tied value.
- Component-wise product of S and R' provides us with the vector W, the vector of the signed ranks. The sum of absolute values of the positive entries in W is W_+ and the sum of absolute values of the negative entries in W is W_-. After this we define, $W_R = min\{W_+, W_-\}$
- Then the test statistic Z is calculated by the equation

$$Z = \frac{W_R - \frac{n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24} - \frac{\Sigma t^3 - \Sigma t}{48}}} \tag{1}$$

  where n is the number of components in D and t is the number of times some i-th entry occurs in R', summed over all such repeated instances.
- Finally R is calculated using $R = \frac{|Z|}{\sqrt{N}}$, where N is the total number of datasets compared, which is 14 in our case.

Note that a higher value W_+ for LoRAS indicates towards a superior performance of LoRAS and the value of R indicates towards how superior(with a higher W_+)/ inferior(with a higher W_-) the performance of LoRAS is, compared to the other oversampling model for the tested datasets. Tarawneh et al. (2020) have considered ranges of $R \leq 0.1$, $0.1 < R \leq 0.5$ and $R > 0.5$ to be indicators for small, medium and high degree of change (improvement or deterioration) in the predictive performance respectively.

From Table 7 we note that, LoRAS has a higher W_+ value for both F1 Score and Balanced accuracy in comparison to each of the other convex combination based oversampling methods in consideration. Moreover for the F1 Score measure, the R value is also more than 0.5, indicating a high degree of improvement in F1-Score for LoRAS, over the considered oversampling models. Similarly, for Balanced accuracy, we find high degree of improvement for LoRAS, over all considered oversampling models except the Borderline-2 SMOTE, for which there is a medium degree of improvement. Overall, we thus conclude

Table 7 Table showing $W_+/W_-/R$ for comparison of LoRAS against the other oversampling algorithms, in terms of both the performance measures we have used: F1-score and balanced accuracy

Measure	Baseline	SMOTE	Bl-1	Bl-2	SVM	ADASYN
F1-score	95/10/.713	105/0/.880	90/15/.629	102/3/.830	80/15/.629	105/0/.880
Balanced accuracy	105/0/.880	102/3/.830	95/10/.715	69/36/.286	95/10/.722	95/10/.837

that LoRAS provides a significant improvement in performance over the compared convex combination based oversampling methods.

5 Discussion

We have constructed a mathematical framework to prove that LoRAS is a more effective oversampling technique since it provides a better estimate for the mean of the underlying local data distribution of the minority class data space. Let $X = (X_1, \dots, X_{|F|}) \in C_{\min}$ be an arbitrary minority class sample. Let N_k^X be the set of the k-nearest neighbors of X, which will consider the neighborhood of X. Both SMOTE and LoRAS focus on generating augmented samples within the neighborhood N_k^X at a time. We assume that a random variable $X \in N_k^X$ follows a shifted t-distribution with k degrees of freedom, location parameter μ, and scaling parameter σ. Note that here σ is not referring to the standard deviation but sets the overall scaling of the distribution (Simon 2009), which we choose to be the sample variance in the neighborhood of X. A shifted t-distribution is used to estimate population parameters, if there are less number of samples (usually, ≤ 30) and/or the population variance is unknown. Since in SMOTE or LoRAS we generate samples from a small neighborhood, we can argue in favour of our assumption that locally, a minority class sample X as a random variable, follows a t-distribution. Following Blagus and Lusa (2013), we assume that if $X, X' \in N_k^X$ then X and X' are independent. For $X, X' \in N_k^X$, we also assume:

$$\begin{aligned} \mathbf{E}[X] &= \mathbf{E}[X'] \\ &= \mu = (\mu_1, \dots, \mu_{|F|}) \\ \mathrm{Var}[X] &= \mathrm{Var}[X'] \\ &= \sigma^2\left(\frac{k}{k-2}\right) \\ &= \sigma'^2 = (\sigma_1'^2, \dots, \sigma_{|F|}'^2) \end{aligned} \tag{2}$$

where, $\mathbf{E}[X]$ and $\mathrm{Var}[X]$ denote the expectation and variance of the random variable X respectively. However, the mean has to be estimated by an estimator statistic (i.e. a function of the samples). Both SMOTE and LoRAS can be considered as an estimator statistic for the mean of the t-distribution that $X \in C_{\min}$ follows locally.

Theorem 1 *Both SMOTE and LoRAS are unbiased estimators of the mean μ of the t-distribution that X follows locally. However, the variance of the LoRAS estimator is less than the variance of SMOTE given that $|F| > 2$.*

Proof A shadowsample S is a random variable $S = X + B$ where $X \in N_k^X$, the neighborhood of some arbitrary $X \in C_{\min}$ and B follows $\mathcal{N}(0, \sigma_B)$.

$$\begin{aligned}\mathbf{E}[S] &= \mathbf{E}[X] + \mathbf{E}[B]\\ &= \mu\\ \mathrm{Var}[S] &= \mathrm{Var}[X] + \mathrm{Var}[B]\\ &= \sigma'^2 + \sigma_B^2\end{aligned} \tag{3}$$

assuming S and B are independent. Now, a LoRAS sample $L = \alpha_1 S^1 + \cdots + \alpha_{|F|} S^{|F|}$, where $S^1, \dots, S^{|F|}$ are shadowsamples generated from the elements of the neighborhood of X, N_k^X, such that $\alpha_1 + \cdots + \alpha_{|F|} = 1$. The affine combination coefficients $\alpha_1, \dots, \alpha_{|F|}$ follow a Dirichlet distribution with all concentration parameters assuming equal values of 1 (assuming all features to be equally important). For arbitrary $i,j \in \{1, \dots, |F|\}$,

$$\begin{aligned}\mathbf{E}[\alpha_i] &= \frac{1}{|F|}\\ \mathrm{Var}[\alpha_i] &= \frac{|F|-1}{|F|^2(|F|+1)}\\ \mathrm{Cov}(\alpha_i, \alpha_j) &= \frac{-1}{|F|^2(|F|+1)}\end{aligned}$$

where $\mathrm{Cov}(A, B)$ denotes the covariance of two random variables A and B. Assuming α and S to be independent,

$$\mathbf{E}[L] = \mathbf{E}[\alpha_1]\mathbf{E}[S^1] + \cdots + \mathbf{E}[\alpha_{|F|}]\mathbf{E}[S^{|F|}] = \mu \tag{4}$$

Thus L is an unbiased estimator of μ. For $j, k, l \in \{1, \dots, |F|\}$,

$$\begin{aligned}\mathrm{Cov}[\alpha_k S_j^k, \alpha_l S_j^l] &= \mathbf{E}[\alpha_k S_j^k \alpha_l S_j^l] - \mathbf{E}[\alpha_k S_j^k]\mathbf{E}[\alpha_l S_j^l]\\ &= \mathbf{E}[\alpha_k \alpha_l]\mu_j^2 - \frac{\mu_j^2}{|F|^2}\\ &= \left[\mathrm{Cov}(\alpha_k, \alpha_l) + \frac{1}{|F|^2}\right]\mu_j^2 - \frac{\mu_j^2}{|F|^2} = \mu_j^2 \mathrm{Cov}(\alpha_k, \alpha_l)\end{aligned} \tag{5}$$

since $\alpha_k \alpha_l$ is independent of $S_j^k S_j^l$. For an arbitrary j, j-th component of a LoRAS sample L_j

$$\begin{aligned}\mathrm{Var}(L_j) &= \mathrm{Var}(\alpha_1 S_j^1 + \cdots + \alpha_{|F|} S_j^{|F|})\\ &= \mathrm{Var}(\alpha_1 S_j^1) + \cdots + \mathrm{Var}(\alpha_{|F|} S_j^{|F|}) + \Sigma_{k=1}^{|F|} \Sigma_{l=1, l\neq k}^{|F|} \mathrm{Cov}(\alpha_k S_j^k, \alpha_l S_j^l)\\ &= \frac{\mu_j^2(|F|-1) + 2(\sigma_j'^2 + \sigma_{Bj}^2)|F|}{|F|(|F|+1)} - \frac{\mu_j^2(|F|-1)}{|F|(|F|+1)}\\ &= \frac{2(\sigma_j'^2 + \sigma_{Bj}^2)}{(|F|+1)}\end{aligned} \tag{6}$$

For LoRAS, we take an affine combination of $|F|$ shadowsamples and SMOTE considers an affine combination of two minority class samples. Note, that since a SMOTE generated oversample can be interpreted as a random affine combination of two minority class samples, we can consider, $|F| = 2$ for SMOTE, independent of the number of features. Also, from Eq. 4, this implies that SMOTE is an unbiased estimator of the mean of the local data distribution. Thus, the variance of a SMOTE generated sample as an estimator of μ would be $\frac{2\sigma'^2}{3}$ (since $B = 0$ for SMOTE). But for LoRAS as an estimator of μ, when $|F| > 2$, the variance would be less than that of SMOTE. □

This implies that, locally, LoRAS can estimate the mean of the underlying t-distribution better than SMOTE. To visualize the key aspects of LoRAS oversampling, we provide the PCA plots for oversampled data from the ozone_level dataset several oversampling methods we have studied in Fig. 2. From Fig. 2 we can observe that SMOTE and ADASYN oversamples highly on the neighbourhood of the outliers, depicted by a blue box in each subplot. While this is somewhat controlled in Borderline1-SMOTE and SVM SMOTE, they still generate some synthetic samples in this neighbourhood. LoRAS on the other hand refrains, leveraging on its strategy to produce a better estimate for local mean of the underlying local data distribution. This enables LoRAS to ignore the outliers and to oversample more uniformly resulting in a better approximation of the data manifold. Note that, the average F1-Scores of the oversampling models as presented in Table 4 has a direct correlation to how the oversampling strategy oversamples in this neighbourhood. SMOTE and ADASYN generates the lowest F1-Scores and show a tendency of oversampling excessively from this neighbourhood. Borderline-SMOTE and SVM improves the F1-Score compared to SMOTE and ADASYN, again,

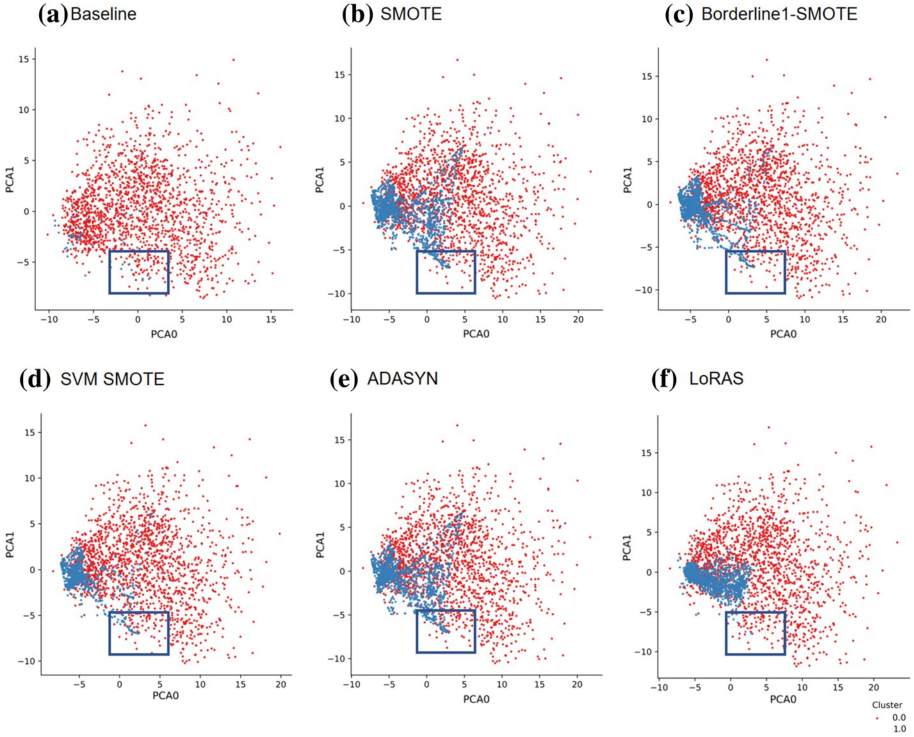

Fig. 2 Figure showing for principal component analysis plot of ozone dataset for baseline data and oversampled data with several oversampling strategies for the ozone_level dataset. The boxed region in each subplot shows a neighbourhood of outliers and how each oversampling stategy generates synthetic samples in that neighbourhood

consistent to their behaviour of oversampling lesser in this neighbourhood. LoRAS, has the highest average F1-Score and oversampling very sparsely from this neighbourhood.

6 Conclusions

Oversampling with LoRAS produces comparatively balanced ML model performances on average, in terms of Balanced Accuracy and F1-Score among the compared convex-combination strategy based oversampling techniques. This is due to the fact that, in most cases LoRAS produces lesser misclassifications on the majority class with a reasonably small compromise for misclassifications on the minority class. From our study we infer that for tabular high dimensional and highly imbalanced datasets our LoRAS oversampling approach can better estimate the mean of the underlying local distribution for a minority class sample (considering it a random variable) and can improve Balanced accuracy and F1-Score of ML classification models. However, the scope of such convex combination based strategies including LoRAS, might be limited for heterogeneous image based imbalanced datasets.

The distribution of both the minority and majority class data points is considered in the oversampling techniques such as Borderline1 SMOTE, Borderline2 SMOTE, SVM-SMOTE, and ADASYN (Gosain and Sardana 2017). SMOTE and LoRAS are the only two techniques, among the oversampling techniques we explored, that deal with the problem of imbalance just by generating new data points, independent of the distribution of the majority class data points. Thus, comparing LoRAS and SMOTE gives a fair impression about the performance of our novel LoRAS algorithm as an oversampling technique, without considering any aspect of the distributions of the minority and majority class data points and relying just on resampling. Other extensions of SMOTE such as Borderline1 SMOTE, Borderline2 SMOTE, SVM-SMOTE, and ADASYN can also be built on the principle of LoRAS oversampling strategy. According to our analyses LoRAS already reveals great potential on a broad variety of applications and evolves as a true alternative to SMOTE, while processing highly unbalanced datasets.

Acknowledgements We thank Prof. Ria Baumgrass from Deutsches Rheuma-Forschungszentrum (DRFZ), Berlin for enlightening discussions on small datasets occuring in her research related to cancer therapy, that led us to the current work. We thank the German Network for Bioinformatics Infrastructure (de.NBI) and Establishment of Systems Medicine Consortium in Germany e:Med for their support, as well as the German Federal Ministry for Education and Research (BMBF) Programs (FKZ 01ZX1709C) for funding us.

Funding Open Access funding enabled and organized by Projekt DEAL.

Code availability A preliminary implementation of the algorithm in `Python (V 3.7.4)` and an example `Jupyter Notebook` for the credit card fraud detection dataset is provided on the GitHub repository https://github.com/sbi-rostock/LoRAS. This version does not yet include the t-embedding parameter. In our computational code, $|S_p|$ corresponds to `num_shadow_points`, L_σ corresponds to `list_sigma_f`, N_{aff} corresponds to `num_aff_comb`, N_{gen} corresponds to `num_generated_points`.

References

Aditsania, A., & Saonard, A. L. (2017). Handling imbalanced data in churn prediction using ADASYN and backpropagation algorithm. In *2017 3rd international conference on science in information technology (ICSITech)* (pp. 533–536). https://doi.org/10.1109/ICSITech.2017.8257170.

Ah-Pine, J., & Soriano-Morales, E.-P. (2016). A study of synthetic oversampling for Twitter imbalanced sentiment analysis. In *Workshop on interactions between data mining and natural language processing (DMNLP 2016)* (Vol. 1646, pp. 17–24).

Anand, A., Pugalenthi, G., & Gary Suganthan, P. (2010). An approach for classification of highly imbalanced data using weighting and undersampling. *Amino Acids*, *39*, 1385–1391. https://doi.org/10.1007/s00726-010-0595-2.

Barua, S., Islam, M. M., Yao, X., & Murase, K. (2014). Mwmote—majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering*, *26*(2), 405–425. https://doi.org/10.1109/TKDE.2012.232.

Bellinger, C., Drummond, C., & Japkowicz, N. (2016). Beyond the boundaries of smote. In P. Frasconi, N. Landwehr, G. Manco, & J. Vreeken (Eds.), *Machine learning and knowledge discovery in databases* (pp. 248–263). Cham: Springer.

Bellinger, C., Drummond, C., & Japkowicz, N. (2018). Manifold-based synthetic oversampling with manifold conformance estimation. *Machine Learning*, *107*, 605–637. https://doi.org/10.1007/s10994-017-5670-4.

Blagus, R., & Lusa, L. (2013). Smote for high-dimensional class-imbalanced data. *BMC Bioinformatics*, *14*(1), 106. https://doi.org/10.1186/1471-2105-14-106.

Bunkhumpornpat, C., Sinapiromsaran, K. & Chidchanok, L. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Advances in knowledge discovery and data mining, Lecture notes in computer science* (Vol. 5476, pp. 475–482). Springer.https://doi.org/10.1007/978-3-642-01307-2_43. ISBN: 978-3-642-01307-2.

Carvalho, A.M. & Prati, R.C. (2018). Improving knn classification under unbalanced data. A new geometric oversampling approach. In *2018 international joint conference on neural networks (IJCNN)* (pp. 1–6). https://doi.org/10.1109/IJCNN.2018.8489411.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–335. https://doi.org/10.1613/jair.953.

Chawla, N. V., Lazarevic, A. H., Lawrence, O., & Bowyer, K. (2003). SMOTEBoost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery* (pp. 107–119). https://doi.org/10.1007/978-3-540-39804-2_12. ISBN: 978-3-540-39804-2.

Chiamanusorn, C., & Sinapiromsaran, K. (2017) Extreme anomalous oversampling technique for class imbalance. In *Proceedings of the 2017 international conference on information technology, ICIT 2017* (pp. 341–345). New York, NY, USA: ACM. https://doi.org/10.1145/3176653.3176671. ISBN 978-1-4503-6351-8.

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on machine learning, ICML '06* (pp. 233–240). New York, NY, USA: ACM. https://doi.org/10.1145/1143844.1143874. ISBN 1-59593-383-2.

Ding, Z. (2011). Diversified ensemble classifiers for highly imbalanced data learning and its application in bioinformatics. Ph.D. thesis. Atlanta, GA, USA: Georgia State University. ISBN: 978-1-267-04661-1.

Douzas, G., & Bacao, F., (2019). Geometric smote a geometrically enhanced drop-in replacement for smote. *Information Sciences*, *501*, 118–135. https://doi.org/10.1016/j.ins.2019.06.007.

Elhassan, T., Aljurf, M., Al-Mohanna, F., & Shoukri, M. (2016). Classification of imbalance data using Tomek link (T-Link) combined with random under-sampling (RUS) as a data reduction method. *Global Journal of Technology and Optimization*, *1*, 2–11. https://doi.org/10.21767/2472-1956.100011.

Elrahman, S. M. A., & Abraham, A. (2013). A review of class imbalance problem. *Journal of Network and Innovative Computing*, *1*, 332–340.

Gao, M., Hong, X., Chen, S., & Harris, C. J. (2011). On combination of SMOTE and particle swarm optimization based radial basis function classifier for imbalanced problems. In *The 2011* *international joint conference on neural networks* (pp. 1146–1153). IEEE. https://doi.org/10.1109/IJCNN.2011.6033353. ISBN: 978-1-4244-9635-8.

Gosain, A., & Sardana S. (2017). Handling class imbalance problem using oversampling techniques: A review. In *2017 international conference on advances in computing, communications and informatics (ICACCI)* (pp. 79–85). https://doi.org/10.1109/ICACCI.2017.8125820. ISBN: 978-1-5090-6367-3.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Advances in intelligent computing. ICIC* (Vol. 3644, pp. 878–887). Berlin, Heidelberg: Springer. https://doi.org/10.1007/1153805_91. ISBN: 978-3-540-31902-3.

Hanifah, F. S., Wijayanto, H., & Kurnia, A. (2015). SMOTE bagging algorithm for imbalanced dataset in logistic regression analysis (Case: Credit of Bank X). *Applied Mathematical Sciences*, *9*(138), 6857–6865. https://doi.org/10.12988/ams.2015.58562.

He, H., Yang, B., Garcia, E., & Shutao, L. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks*. https://doi.org/10.1109/IJCNN.2008.4633969. ISBN: 2161-4393.

Hinton, G. E., & Roweis, S. T. (2003). Stochastic neighbor embedding. In S. Becker, S. Thrun, & K. Obermayer (Ed.), *Advances in neural information processing systems* (Vol. 15, pp. 857–864). MIT Press. http://papers.nips.cc/paper/2276-stochastic-neighbor-embedding.pdf.

Hooda, N., Bawa, S., & Rana, P. S. (2018). B2fse framework for high dimensional imbalanced data: A case study for drug toxicity prediction. *Neurocomputing*, *276*, 31–41. https://doi.org/10.1016/j.neucom.2017.04.081.

Hu, S., Liang, Y., Ma, L., & He, Y. (2009). MSMOTE: Improving classification performance when training data is imbalanced. In *Second international workshop on computer science and engineering* (Vol. 2, pp. 13–17). https://doi.org/10.1109/WCSE.2009.756. ISBN: 978-0-7695-3881-5.

Jing, X., Zhang, X., Zhu, X., Wu, F., You, X., Gao, Y., et al. (2019). Multiset feature learning for highly imbalanced data classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,. https://doi.org/10.1109/TPAMI.2019.2929166.

Kobak, D., & Berens, P. (2019). Visualizing data using t-SNE. *Nature Communications*,. https://doi.org/10.1038/s41467-019-13056-x.

Kovács, G. (2019). Smote-variants: A python implementation of 85 minority oversampling techniques. *Neurocomputing*, *366*, 352–354. https://doi.org/10.1016/j.neucom.2019.06.100.

Le, T., Vo, M. T., Vo, B., & Lee, Y., & Baik, W., (2019). A hybrid approach using oversampling technique and cost-sensitive learning for bankruptcy prediction. *Complexity*, *2019*, 03. https://doi.org/10.1155/2019/8460934.

Lee, S. S. (2000). Noisy replication in skewed binary classification. *Computational Statistics & Data Analysis*, *34*(2), 165–191. https://doi.org/10.1016/S0167-9473(99)00095-X.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*, 559–563.

Mathew, J., Luo, M., Pang, C., & Chan, H. L. (2015). Kernel-based smote for svm classification of imbalanced datasets. In *IECON 2015—41st annual conference of the IEEE industrial electronics society* (pp. 001127–001132). IEEE.https://doi.org/10.1109/IECON.2015.7392251. ISBN: 978-1-4799-1762-4.

Pozzolo, A. D., Boracchi, G. C., Olivier, A. C., & Bontempi, G. (2017). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, *29*, 1–14. https://doi.org/10.1109/TNNLS.2017.2736643.

Puntumapon, K., & Waiyamai, K. (2012). A pruning-based approach for searching precise and generalized region for synthetic minority over-sampling. In *Advances in knowledge discovery and data mining* (pp. 371–382). Berlin, Heidelberg: Springer.

Ramentol, E., Verbiest, N., Bello, R., Caballero, Y., Cornelis, C., & Herrera, F. (2012). Smote-frst: A new resampling method using fuzzy rough set theory. In *World scientific proceedings series on computer engineering and information science* (Vol. 7, pp. 800–805). https://doi.org/10.1142/9789814417747_0128. ISBN: 9789814417730.

Saez, J. A., Krawczyk, B., & Woźniak, M. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, *57*, 164–178. https://doi.org/10.1016/j.patcog.2016.03.012.

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*, *10*, 1–21. https://doi.org/10.1371/journal.pone.0118432.

Santoso, B., Wijayanto, H., Notodiputro, K. A., & Sartono, B. (2017). Synthetic over sampling methods for handling class imbalanced problems: A review. *IOP Conference Series: Earth and Environmental Science*, *58*, 012–031. https://doi.org/10.1088/1755-1315/58/1/012031.

Simon, D. (2009). *Jackman*. WILEY: Bayesian Analysis for the Social Sciences. https://doi.org/10.1002/9780470686621. ISBN 9780470011546.

Suh, Y., Jaemyung, Yu., Mo, J., Song, L., & Kim, C. (2017). A comparison of oversampling methods on imbalanced topic classification of Korean news articles. *Journal of Cognitive Science*, *18*, 391–437.

Tarawneh, A. S., Hassanat, A. B. A., Almohammadi, K., Chetverikov, D., & Bellinger, C. (2020). Smotefuna: Synthetic minority over-sampling technique based on furthest neighbour algorithm. *IEEE Access*, *8*, 59069–59082.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019). Credit card fraud detection—machine learning methods. In *2019 18th international symposium INFOTEH-JAHORINA (INFOTEH)* (pp. 1–5). https://doi.org/10.1109/INFOTEH.2019.8717766.

Wang, K.-J., Makond, B., Chen, K.-H., & Wang, K.-M. (2014). A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients. *Applied Soft Computing*, *20*, 15–24. https://doi.org/10.1016/J.ASOC.2013.09.014.

Young, W. A., Ii, N., Scott, L., Weckman, G. R., & Chelberg, D. M. (2015). Using Voronoi diagrams to improve classification performances when modeling imbalanced datasets. *Neural Computing and Applications*, *26*(5), 1041–1054. https://doi.org/10.1007/s00521-014-1780-0.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Saptarshi Bej[1] · Narek Davtyan[1] · Markus Wolfien[1] · Mariam Nassar[1] · Olaf Wolkenhauer[1]

Saptarshi Bej
saptarshi.bej@uni-rostock.de

Narek Davtyan
narek.davtyan@uni-rostock.de

Markus Wolfien
markus.wolfien@uni-rostock.de

Mariam Nassar
mariam.nassar@uni-rostock.de

[1] Department of Systems Biology and Bioinformatics, University of Rostock, Universitätsplatz 1, 18051 Rostock, Germany