An Experimental Perspective on Sampling Methods for Imbalanced Learning from Financial Databases

Luis Eduardo Boiko Ferreira, Jean Paul Barddal,
Fabrício Enembreck
Programa de Pós-Graduação em Informática (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba, Brazil
{luis.boiko, jean.barddal, fabricio}@ppgia.pucpr.br

Heitor Murilo Gomes
Télécom ParisTech
Université Paris-Saclay
Paris, France
heitor.gomes@telecom-paristech.fr

***Abstract*—The financial market is one of the major consumers of data mining techniques, and the main reason is their efficiency to analyze complex data. One important trait shared between most financial applications is *class imbalance*. Since traditional classification methods assume nearly balanced classes and equal misclassification costs, they usually fail to deal with imbalanced data. However, in financial contexts, problems are usually imbalanced, and instances from the minority class are known for deficits of millions of dollars every year, e.g., credit card frauds, money laundering transactions and so forth. Over the years, several techniques for dealing with class imbalance have been developed, such as sampling techniques and algorithm adaptations. In this study, we analyze how different sampling techniques impact the performance of different classification systems on financial applications. Results show that, for the given datasets, sampling techniques allow the improvement of prediction performance of the minority class while also improving overall classification rates. Nevertheless, their use often deteriorates the performance in predicting the majority class.**

I. Introduction

The financial market is one of the major consumers of data mining (DM) techniques. Since the credit crisis in 2008 [1], financial institutions have been investigating their datasets with the goal of improving decision-making during, for instance, credit scoring [2], [3], [4], [5], fraud detection [6], [7], [8] and product recommendation [9], [10]. Other aspects of the financial spectrum also benefit from the analysis of their data, such as investigate money laundering [11], [12] and bankruptcy prediction [13].

One of the reasons that led to the adoption of DM techniques is due to the complexity involved in the data to be processed. The application of these techniques in problem-solving, besides the efficiency in generating models, is because people are prone to make mistakes when doing certain analyzes or possibly trying to find relationships between multiple characteristics, which usually does not occur with the DM techniques [14].

One important trait shared between most financial applications is *class imbalance*. For instance, the number of non-creditworthy requests in credit scoring and the number of fraudulent transactions in fraud detection schemes is negligible compared to the total number of events. However, these events are of the utmost importance for financial institutions, as these are known for deficits of millions of dollars [15] every year. Even though analyses of class imbalance on financial datasets have been discussed throughout the years [16], [17], [18], they often (i) use datasets that are not public, (ii) refrain on evaluating their proposals on different datasets, and (iii) examine a small portion of existing sampling methods (or even none). Sampling methods are often divided into two categories: under- and over-sampling. The former promotes class balance by decreasing the number of instances in the majority class, while the latter synthesizes new instances belonging to the minority class.

The goal of this paper is to fill the gap of the studies mentioned above in two aspects: (i) an analysis of multiple datasets, and (ii) with different sampling methods. This paper is divided as follows. Section II describes the classification task with a focus on imbalanced scenarios, while Section III reviews the sampling techniques evaluated. Section IV introduces the datasets used, their objectives, and its main traits. Section V presents and discusses our analysis and the results obtained. Finally, Section VI concludes this study and states envisioned future studies.

II. Classification on Imbalanced Scenarios

In this work, we focus on binary classification for imbalanced financial datasets. Binary classification is assumed here since most of the publicly available datasets for financial applications share these characteristics. Details about the datasets used in this experiment are provided in Section IV.

Classification aims at learning a labeling model $f : X \to Y$ which maps d-dimensional instances $\vec{x} \in X$ to a set of classes Y. We limit this study to scenarios where Y assumes two possible values in the $\{0, 1\}$ domain. For the sake of brevity, we denote Y_{min} as the set of instances belonging to the minority class ($Y = 1$), while Y_{maj} is the set of instances of the majority class ($Y = 0$), also that $|Y_{maj}| \gg |Y_{min}|$.

By default, most of the classification methods, such as Logistic Regression [19], Naive Bayes [20], and Decision Trees [21], fail when working on imbalanced datasets. For instance, Logistic Regression has its model intercept affected, which results all probabilities to be skewed [22]. Similarly, Naive Bayes is affected by class imbalance as the minority

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

class assumes smaller weights, while Decision Trees select features based on information theoretic formulas, e.g. Entropy and Information Gain, that are known for being biased towards the majority class. Fortunately, a decent amount of effort has been put on developing techniques to overcome class imbalance, and are categorized into (i) classifier adaptations, and (ii) sampling techniques, which is the focus of the current work.

Sampling techniques are divided into two groups: under-sampling and over-sampling [23]. under-sampling methods promote class balancing by decreasing the number of instances that belong to the majority class. In contrast, over-sampling acts in the opposite direction, i.e., it increases the number of instances of the minority class by synthesizing data samples based on the existing samples. In the next section, we survey both under- and over-sampling techniques used during our investigation. All sampling techniques are then combined with the Logistic Regression, Naive Bayes, and Decision Tree classifiers and evaluated on multiple datasets in Section V. One of the goals of this analysis is to verify whether each type of classifier works best with one specific type of sampling technique or if we can find a more generic behavior given different classifiers and sampling techniques across the financial domain.

III. Literature Review on Sampling Techniques

In this section, we discuss relevant sampling techniques that have been largely applied to deal with class imbalance. Sampling techniques are designed to help classifiers by modifying the dataset before training. The techniques presented and evaluated in this study were chosen based on the number of citations and are divided into **over-** and **under-sampling** techniques.

A. Under-sampling

under-sampling techniques promote class balance by removing instances from Y_{maj} while keeping Y_{min} intact.

Random under-sampling (RU). This technique randomly selects a sub-sample of instances $E \subset Y_{maj}$ and excludes them from the dataset so that $|Y_{min}| \approx |Y_{maj}|$ holds after removal. This is one of the most naive methods to promote class balance, and it often leads to information loss. This occurs because there is no criterion to define which instances will be removed, and thus, there is no guarantee that representative instances will not be incorrectly removed from Y_{maj}.

Near Miss (NM). Near Miss uses a k-nearest neighbors classifier [24] to flag instances for removal. Three variants for Near Miss have been proposed, namely Near Miss-1, Near Miss-2 and Near Miss-3. The first discards instances from Y_{maj} such that the mean distance from a given number of neighbors from Y_{min} is the lowest. The second approach acts oppositely, discarding a user-defined number of farthest neighbors from Y_{min}. Lastly, the third variant removes a given number of Y_{maj} instances, ensuring that every instance from Y_{min} is surrounded by some instances from Y_{maj}.

Tomek Links (TK). Tomek Links has the goal of removing instances that cause class overlap. Given a pair of instances (x_i, x_j), where $x_i \in Y_{min}, x_j \in Y_{maj}$ and $d(x_i, x_j)$ is the distance between x_i and x_j, the pair of instances can be considered as a Tomek Link if there is no instance x_k that $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_i, x_j)$. Following this definition, if a pair of instances forms a Tomek Link, one of them is either noise or they are close to a decision boundary. Therefore, Tomek links can be considered a cleaning technique, which removes instances to decrease overlaps between classes.

Cluster Centroids (CC). One of the biggest problems of the under-sampling techniques is that it may lead to information loss once we remove instances from Y_{maj}. To solve this problem, this technique combines the usage of K-means clustering algorithm [25] with the RU technique. CC starts by dividing Y_{maj} in K clusters. The instances in an i^{th} cluster are named k_i. The ratio between the number of samples from Y_{maj} for the total number of the samples in the last cluster is defined as $r_i = \frac{|k_i|}{|Y_{maj}|}, 1 \leq i \leq k$. The final number of instances to be kept in the dataset and that is also the closest to each centroid is given by $k_i \times r_i$, meaning that the exceeding instances are removed.

B. Over-sampling

In contrast to under-sampling techniques, over-sampling has the goal of synthesizing data from Y_{min} to promote class balance.

SMOTE (SM). This is a powerful technique that demonstrated great success in many applications [26]. SMOTE was designed to first retrieve the k-neighbors for each instance $x_i \in Y_{min}$. Then, one of the k-neighbours $\hat{x}_i$ is randomly selected and its distance to x_i is multiplied by a random number $\delta \in [0, 1]$ in order to generate a new vector (x_{new}) that is between x_i and the k-neighbours as described in Equation 1.

$$x_{new} = x_i + \delta(\hat{x}_i - x_i) \tag{1}$$

One of the major drawbacks of SMOTE is that it can induce to class overlap problem. This is due to the process that generates synthetic instances, once it does not consider the neighborhood of instances and class overlaps.

SMOTE with Tomek Links (SMTK). This technique is a two-step process that combines the original SMOTE with Tomek Links [27]. Initially, SMOTE is applied, followed by a cleanup process performed by Tomek Links to remove instances in class-overlapping regions.

ADASYN (ADA). This method uses a model to create different amounts of synthetic samples based in its distribution [28]. The number of synthetic samples to be generated is calculated using Equation 2.

$$G = (|S_{maj}| - |S_{min}|) \times \beta \tag{2}$$

where $\beta \in [0, 1]$ is a parameter used to specify the desired balance level after the generation of synthetic samples. For each

TABLE I
DATABASES DESCRIPTION

Identifier	# of Instances	# of Attributes	Y_{maj}	Y_{min}
DCCC	30000	24	23364	6636
GERMAN	1000	20	700	300
SANTANDER	76020	371	73012	3008

$X_i \in S_{min}$, the k-nearest neighbours are selected according the Euclidean distance, then the relation Γ_i is computed as follows in Equation 3.

$$\Gamma_i = \frac{1}{Z} \times \frac{\Delta_i}{K}, i = 1, ..., S_{min} \tag{3}$$

where Δ_i represent the number of samples in the k-nearest neighbours of X_i from Y_{maj}, and Z is a normalization constant such that Γ_i is a distribution function $\sum \Gamma_i = 1$. The number of synthetic samples that is generated for each $X_i \in S_{min}$ is given by Eq. 4.

$$g_i = \Gamma_i \times G \tag{4}$$

Finally, $\forall x_i \in S_{min}, g_i$ synthetic samples are generated according to Equation 1, as the regular SM.
The key of this method is to use the density distribution Γ as a criterion to determine the number of synthetic samples to be generated for each sample from Y_{min}, by adapting the weights from different samples from Y_{min} in order to compensate the unequal distribution of the classes.

IV. Financial Datasets

In this study, we work with three publicly available financial-related datasets. In Table I we report their main traits, including the number of instances, attributes and instances per class. We highlight the Santander as being the biggest, and it has been downloaded from a Kaggle challenge[1]. The others are from UCI [29], where all attributes are integers, boolean or real. Some attributes were categorical in the original datasets but were converted into boolean ones using one-hot encoding for processing in scikit-learn [30], version 0.18. Each of the datasets is briefly described below.

Default of Credit Card Clients (DCCC). The goal of this dataset is to build a default prediction model for a bank in Taiwan, i.e., a customer is creditworthy or not. This dataset contains 30,000 instances and 24 anonymized features.

German (GERMAN). This dataset depicts the problem of determining whether a customer is creditworthy or not given 20 features. Very few is known about this dataset since its source and features are confidential.

Santander Customer Satisfaction (SANTANDER). In contrast to the aforementioned datasets, the goal of this dataset is to identify the dissatisfied customer from the Santander bank. This dataset contains 371 anonymized features to predict whether a customer is satisfied or dissatisfied with their banking experience.

V. Analysis

In this paper, we test the sampling techniques described in Section III to handle class imbalance to improve classification rates in the financial context. We present results using the standard classifiers listed in Section II.

This analysis follows the framework proposed in [27]. This framework is organized in two steps: **(i)** an intra-family evaluation to determine the best performing approaches, and **(ii)** an inter-family comparison to find out the fittest approach for this P2P lending dataset.

A. Experimental Protocol

As previously discussed, we analyze the sampling approach to handle class imbalance. Also, we include results using the same classifiers without sampling, and these are assumed to be our *baseline* models.

Evaluating imbalanced datasets is not a simple task since the use of traditional metrics in imbalanced domains can lead to sub-optimal classification models and produce misleading conclusions [23]. The standard metric used to evaluate classification models is Accuracy. One of its drawbacks is that it depends on label distribution. For instance, in a binary classification task, if Y_{min} represents only 5% of the data, and we have a classifier that only guesses Y_{maj}, we have 95% of accuracy, yet the classifier would not correctly classify a single instance from Y_{min}. To avoid this type of issue, we proceed with the Area Under the ROC curve (AUROC), specificity and sensibility as measures of classification quality, since they known to be more suitable for imbalanced datasets [23]. The *baseline* was stipuled with AUROC.

We split the data into two stratified datasets: a training set X_{train} and a test set X_{test}, with 70% and 30% of the data, respectively. Our validation process using X_{train} and X_{test} is detailed as follows:

- X_{train}: This dataset is used to optimize the parameters of each of the previously mentioned methods. The tuning process performed adopts a 5-fold stratified cross-validation scheme. Tuning was performed to optimize both the parameters for classifiers and sampling techniques. The metric chosen for tuning classifiers is AUROC since it accounts for the classification rates of both classes.
- X_{test}: Given the tuned versions of the classifiers and techniques obtained from the training set, these are then used in another 5-fold stratified cross-validation scheme over X_{test}. The results listed in the following sections are the averages obtained during this step.

B. Classifiers and Sampling Methods

Three types of classifiers were evaluated in this study, namely NB, LG, and DT. The list of tuned parameters are listed in Table II. Also, the tuned parameters for sampling are listed in Table III.

Finally, it is important to mention that all of the classification and sampling techniques used follow the implementation

[1] https://www.kaggle.com/c/santander-customer-satisfaction

TABLE II
PARAMETERS FOR GRID SEARCH OF BASELINE CLASSIFIERS

Classifier	Parameter	Values
DT	criterion	gini, entropy
	splitter	best, random
	min_samples_split	2, 10, 20
	max_depth	None, 2, 5, 10
	min_samples_leaf	1, 5, 10
	max_leaf_nodes	None, 5, 10, 20
LG	C	0.001, 0.01, 0.1, 1, 10, 100, 1000
	solver	newton-cg, lbfgs, liblinear, sag
NB	-	None

The NB classifier has no tuning parameters.

TABLE III
PARAMETERS FOR GRID SEARCH FOR SAMPLING

Method	Parameter	Values
RU	ratio	0.8, 09, 1.0
SM	kind	regular, borderline1, borderline2
	ratio	0.8, 09, 1.0
ADA	n_neighbors	1, 3, 5, 7
	ratio	0.8, 09, 1.0
NM	version	1, 2, 3
	ratio	0.8, 09, 1.0
TK	-	-

TK has no tuning parameters

provided by scikit-learn[2] and imbalanced-learn[3] Python packages. The experiment scripts are available at [4].

C. Results – Santander dataset

In this experiment, the most stable classifiers were NB for baseline and DT + NM for sampling, scoring 0.51 and 0.70 at AUROC, respectively (Table IV. The NB has no parameters to tune, but DT was tuned by the gridsearch with *criterion = gini*, *splitter = random*, *min_samples_split = 20*, *max_depth = 5*, *min_samples_leaf = 20* and *max_leaf_nodes = None*. As the grid search was tuned for the best AUROC, the fact that the optimal parameter for *max_depth* was 5, gives evidence that the more the tree grows, the more it overfits to majority class. Not only the classifier was tuned, but also, the NM was tuned. The optimal parameters were *ratio = 0.9* and the *version = 3*. The version 3 of NM basically applies a RU with the heuristic of ensuring that every instance from Y_{min} is surrounded by a given number of Y_{maj} samples, which makes it slightly better than the RU itself.

This dataset was the biggest in the number of instances and attributes, also have the highest imbalance ratio, with approximately $1 : 24$. The best AUROC and Specificity for this dataset were acquired with NM and CC, respectively, both under-sampling techniques. Both approaches show significant improvements over the baseline.

TABLE IV
RESULTS FOR SANTANDER DATASET

Classifier	Sampling	Sensibility	Specificity	AUROC
DT	Baseline	0.94	0.01	0.47
DT	RU	0.75	0.61	0.68
DT	NM	0.69	0.72	**0.70**
DT	SM	0.91	0.40	0.66
DT	SMTK	0.89	0.30	0.60
DT	TK	**1.00**	0.01	0.50
DT	CC	0.10	**0.98**	0.54
DT	ADA	0.99	0.02	0.51
NB	Baseline	0.99	0.03	0.51
NB	RU	0.99	0.06	**0.53**
NB	NM	0.80	**0.25**	**0.53**
NB	SM	0.99	0.04	0.52
NB	SMTK	0.99	0.03	0.51
NB	TK	0.99	0.03	0.51
NB	CC	**1.00**	0.01	0.50
NB	ADA	0.99	0.03	0.51
LG	Baseline	0.92	0.03	0.48
LG	RU	0.69	0.69	**0.69**
LG	NM	0.72	0.32	0.52
LG	SM	0.64	0.69	0.66
LG	SMTK	0.64	0.69	0.66
LG	TK	**1.00**	0.00	0.50
LG	CC	0.18	**0.92**	0.55
LG	ADA	0.65	0.71	0.68

Values listed in bold are the best results obtained for each classifier.

D. Results – German

This was the smallest dataset regarding instances and attributes. Also, it has the lowest imbalance ratio, with approximately 1:2. In Table V it can be seen that the baseline for this dataset was stipulated with LG, scoring 0.64 on AUROC. The LG also outperform the other tested classifiers stability when combined with sampling techniques, scoring the best AUROC in combination with RU or ADA (0.7). For the RU the optimal *ratio* was 0.8, while for ADA it was 1.0. Also, for the ADA technique, the optimal *n_neighbors* parameter was 3. For the base classifier (LG), the optimal parameters were *solver = newton-cg* and *C = 1* in both cases.

As with the Santander dataset, the best specificity at German dataset was obtained with NB, but in combination with RU or TK (0.97), both tuned with *ratio = 0.8*. While the performance was satisfactory at Y_{min} with this combinations, it has a high miss-classification at Y_{maj}.

For all the tested classifiers with this dataset, the combination with sampling techniques produced interesting results, improving the AUROC and Specificity baselines by up to 13% and 54%, respectively. The DT classifier seems to be more affected by the sampling techniques, once the data distribution affects directly the splits made by this classifier, what can lead to better generalization rates on imbalanced scenarios.

E. Results – Default of Credit Card Clients (DCCC)

This dataset has the imbalance ratio of approximately 1:4. Unlike the previous experiments, the baseline was stipuled with DT (0.60), and the best specificity was obtained with over-sampling techniques, more specifically, SM and SMTK

[2] http://scikit-learn.org/stable/

[3] http://contrib.scikit-learn.org/imbalanced-learn/

[4] https://github.com/lestatwa/sac2018

TABLE V
RESULTS FOR GERMAN DATASET

Classifier	Sampling	Sensibility	Specificity	AUROC
DT	Baseline	0.77	0.22	0.50
DT	RU	0.60	0.67	**0.63**
DT	NM	0.59	0.64	0.62
DT	SM	0.78	0.38	0.58
DT	SMTK	0.77	0.31	0.54
DT	TK	**0.89**	0.20	0.54
DT	CC	0.42	**0.76**	0.59
DT	ADA	0.85	0.26	0.55
NB	Baseline	0.22	0.96	0.59
NB	RU	0.21	**0.97**	0.59
NB	NM	0.32	0.87	0.60
NB	SM	0.23	0.96	0.60
NB	SMTK	0.25	0.94	0.59
NB	TK	0.22	**0.97**	0.59
NB	CC	0.22	0.96	0.59
NB	ADA	**0.64**	0.63	**0.64**
LG	Baseline	0.86	0.42	0.64
LG	RU	0.74	**0.67**	**0.70**
LG	NM	0.81	0.50	0.66
LG	SM	0.73	0.62	0.68
LG	SMTK	0.78	0.59	0.69
LG	TK	**0.91**	0.38	0.65
LG	CC	0.73	0.59	0.66
LG	ADA	0.74	0.65	**0.70**

Values listed in bold are the best results obtained for each classifier.

TABLE VI
RESULTS FOR DEFAULT OF C. C. CLIENTS DATASET

Classifier	Sampling	Sensibility	Specificity	AUROC
DT	Baseline	0.83	0.37	0.60
DT	RU	0.80	0.57	**0.68**
DT	NM	0.83	0.51	0.67
DT	SM	0.87	0.48	0.67
DT	SMTK	0.86	0.50	**0.68**
DT	TK	**0.93**	0.39	0.66
DT	CC	0.60	**0.69**	0.65
DT	ADA	0.90	0.45	0.67
NB	Baseline	0.15	0.90	0.52
NB	RU	0.23	0.90	0.56
NB	NM	0.18	0.88	0.53
NB	SM	0.15	**0.94**	0.54
NB	SMTK	0.15	**0.94**	0.54
NB	TK	0.21	0.90	0.55
NB	CC	0.15	0.90	0.53
NB	ADA	**0.30**	0.85	**0.57**
LG	Baseline	0.65	0.49	0.57
LG	RU	0.86	0.49	0.67
LG	NM	0.79	0.53	0.66
LG	SM	0.71	0.60	0.66
LG	SMTK	0.76	0.58	0.67
LG	TK	**0.94**	0.36	0.65
LG	CC	0.52	**0.74**	0.63
LG	ADA	0.89	0.46	**0.68**

Values listed in bold are the best results obtained for each classifier.

(0.94), both with NB classifier, as can be seen in Table VI. The SM and SMTK optimal parameters *ratio = 1.0* and SM was also *kind = regular*.

The most stable combinations were DT+RU, DT+SMTK and LG+ADA, as both scored 0.68 for AUROC. The optimal ratio for RU, SMTK and ADA were 0.9, 0.8 and 0.8, respectively. Also, the DT was tuned with *criterion = gini*, *splitter = random*, *min_samples_split = 10*, *max_depth = None*, *min_samples_leaf = 10* and *max_leaf_nodes = 10*, and the LG with *C = 0.1 solver = newton-cg*. The DT parameter *max_depth = None* gives evidence that the tree overfitted, but this issue seems to affect the model in a positive way, once it scored the best AUROC. It can be concluded that the usage of sampling techniques increase the baselines for all the tested classifiers.

F. Discussion

The three tested datasets, despite being financial, have different characteristics. The balance ratio varies from 1:2 at the $German$ dataset, which is also the smallest regarding instances and attributes, to 1:24 at the $Santander$, which has the highest number of instances and attributes. The diversity observed regarding balance allows the visualization of the impact on the quality of the instances generated in high quantity, such as in $Santander$, in counterpoint to more balanced datasets.

Despite the different characteristics of the datasets, there is a trend on sampling techniques. The under-sampling technique RU and the over-sampling technique ADA are the best-tested sampling techniques if the miss-classification costs for both classes are equal since it provides the best AUROC. The LG+ADA proves to be the best combination since it scored the best results in two of the three tested datasets.

One of the drawbacks in the use of sampling techniques is the performance deterioration that occurs in the majority class. In general, under-sampling techniques can induce information loss, such as oversampling techniques can induce class overlap. The tested techniques exhibit a relation between the performance increase in the minority class and the deterioration in the opposite class. However, the gains at the minority class, in general, are more substantial than the introduced losses.

VI. CONCLUSION

In this paper, we conducted an empirical analysis of different sampling techniques in the context of improving the recognition of the class of interest in financial datasets. This analysis evaluated both over- and under-sampling techniques and their impact when associated with different types of classifiers.

Even though results show that LG + ADA combination is the best performing for nearly all scenarios evaluated in this work, the NB classifier is the most stable regarding correctly classified instances at Y_{min}, yet, at the cost of compromising the performance at Y_{maj}. Also, the DT classifier seems to be the most affected by the sampling techniques, improving the AUROC and Specificity baselines by up to 23% and 97%, respectively, for the $Santander$ dataset.

Regarding the sampling results, similar evidence was found. Each sampling technique shows a different behavior when used on top of each dataset. However, results show that not only the ratio between classes is important, but also the raw number of instances in each class. For instance, in the $German$ and *Default of C. C. Clients* datasets, most of the under-sampling techniques were less effective, which can be argued as the result of having too little data from the minority class to learn something, even when removing instances from the majority class.

Finally, the results obtained by SMTK in *Default of C. C. Clients*, shows how beneficial a cleanup process can be, since it allows the removal of noisy and borderline instances, which are often responsible for hurting learners.

In general, the overall performance scores obtained across datasets are similar, where we see improvements with under-sampling techniques. We believe this is due to the nature of the problems. When working with intrinsic imbalance and a fair amount of data, e.g., such as financial data, Y_{min} often tends to be represented by a dense region in a large feature space. When over-sampling is applied, it is expected that most of the new synthetic data will be generated in the same region. Despite being able to increase the number of samples from Y_{min}, this kind of technique contributes too little, or even nothing, to classifiers since no new interesting classification patterns would be discovered.

Future works envision the development of a sampling technique that analyzes the underlying "trend" of data, i.e., how it "grows" in the feature space. We hypothesize that the generation of synthetic instances should expand the known boundaries of a class in the feature space. Intuitively, this would allow new patterns to be discovered by the classifier despite the fact that we do not necessarily have dense regions representing the minority class in the original data.

References

[1] Y. Son, H. Byun, and J. Lee, "Nonparametric machine learning models for predicting the credit default swaps: An empirical study," *Expert Systems with Applications*, vol. 58, pp. 210–220, 2016.

[2] Y.-C. Lee, "Application of support vector machines to corporate credit rating prediction," *Expert Systems with Applications*, vol. 33, no. 1, pp. 67–74, 2007.

[3] K. Tsai, S. Ramiah, and S. Singh, "Peer lending risk predictor," 2014.

[4] M. Malekipirbazari and V. Aksakalli, "Risk assessment in social lending via random forests," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4621–4631, 2015.

[5] V. Kumar, S. Natarajan, S. Keerthana, K. Chinmayi, and N. Lakshmi, "Credit risk analysis in peer-to-peer lending system," in *Knowledge Engineering and Applications (ICKEA), IEEE International Conference on*. IEEE, 2016, pp. 193–196.

[6] W. Fitzgerald, "Machine learning for fraud detection," Sep. 21 2017, uS Patent App. 15/070,138.

[7] A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques," *International Journal of System Assurance Engineering and Management*, vol. 8, no. 2, pp. 937–953, 2017.

[8] Y. Wang and W. Xu, "Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud," *Decision Support Systems*, vol. 105, pp. 87–95, 2018.

[9] R. Sharma and S. Kapoor, "A review on the page recommendation model using machine learning approaches," 2017.

[10] N. T. James and K. Rajkumar, "Product recommendation systems based on hybrid approach technology," 2017.

[11] M. A. Villalobos and E. Silva, "A statistical and machine learning model to detect money laundering: an application," 2017.

[12] J. A. Álvarez-Jareño, E. Badal-Valero, J. M. Pavía *et al.*, "Using machine learning for financial fraud detection in the accounts of companies investigated for money laundering," Tech. Rep., 2017.

[13] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, vol. 83, pp. 405–417, 2017.

[14] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[15] N. Carneiro, G. Figueira, and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," *Decision Support Systems*, 2017.

[16] J. A. Sanz, D. Bernardo, F. Herrera, H. Bustince, and H. Hagras, "A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 973–990, 2015.

[17] I. Dutta, S. Dutta, and B. Raahemi, "Detecting financial restatements using data mining techniques," *Expert Systems with Applications*, vol. 90, pp. 374–393, 2017.

[18] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with dte-sbd: Decision tree ensemble based on smote and bagging with differentiated sampling rates," *Information Sciences*, vol. 425, pp. 76–91, 2018.

[19] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 215–242, 1958.

[20] T. Bayes, "Naive bayes classifier," 1968.

[21] J. F. Magee, *Decision trees for decision making*. Harvard Business Review, 1964.

[22] Z. Qiu, H. Li, H. Su, G. Ou, and T. Wang, "Logistic regression bias correction for large scale data with rare events," in *Part II of the Proceedings of the 9th International Conference on Advanced Data Mining and Applications - Volume 8347*, ser. ADMA 2013. New York, NY, USA: Springer-Verlag New York, Inc., 2013, pp. 133–144. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-53917-6_12

[23] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 31:1–31:50, Aug. 2016. [Online]. Available: http://doi.acm.org/10.1145/2907070

[24] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.

[25] N. S. Kumar, K. N. Rao, A. Govardhan, K. S. Reddy, and A. M. Mahmood, "Undersampled k-means approach for handling imbalanced distributed data," *Progress in Artificial Intelligence*, vol. 3, no. 1, pp. 29–38, 2014.

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[27] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, Jun. 2004. [Online]. Available: http://doi.acm.org/10.1145/1007730.1007735

[28] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008, pp. 1322–1328.

[29] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.