Deep Ensembles for Imbalanced Classification

Nataliia Kozlovskaia
Yandex.Taxi, HSE
Email: natalikozlovskaya@gmail.com

Alexey Zaytsev
Skoltech, IITP RAS
Email: a.zaytsev@skoltech.ru

***Abstract*—Most of the standard classification algorithms perform poorly when dealing with the case of imbalanced classes i.e. when there is a class to which the overwhelming majority of samples belong. There are many approaches that deal with this problem, among which SMOTE and SMOTE boosting, the common approach prefers overly simplistic models that lead to degradation of performance. Recent advances in statistical learning theory provide more adequate complexity penalties for weak classifiers, which stem from the Rademacher complexity terms in the ensemble generalization bounds. By adopting these advances and introducing a sample weight correction based on the classification margin at each iteration of boosting we get more precise models for imbalanced classification problems.**

I. Introduction

Many practical classification problems suffer from severe class imbalance: the amount of observations that belong to a major class is significantly higher than the number of observations that belong to a minor class [1]. In medical diagnostics the number of patients with a rare condition is significantly lower than the number of patients without it [2]. Similar imbalance is observed in malware detection [3], churn analysis [4], aircraft fault prediction [5] and so on [6], [7], [8], [9]. Another broad field that contains imbalanced classification problems is anomaly detection [10], as typically number of anomalies in the training sample is small, while correct detection of anomalies is crucial in applications.

Without any modification, classifiers trained to attain high accuracy are tuned to the objects of major class and thus have high false-negative rate with respect to the minor class. In many applications the costs of false-positives are much less than that of false-negatives, especially in healthcare, where falsely classifying an ill patient as healthy can have dire consequences.

It has been observed [11] that common classification algorithms typically perform poorly in imbalanced classification problems. An anecdotal example of such poor performance is illustrated in Table I, which compares Gradient Boosting classifier [12] with SMOTE boosting, which is the state-of-the-art approach for imbalanced classification [13]. It can be seen, that on three common imbalanced datasets the F1-scores obtained using cross-validation are significantly better for SMOTE boosting than for Gradient Boosting. This example shows that in a typical imbalanced classification problem generic approaches can fail to construct classifiers of good quality.

Dataset	Gradient boosting	SMOTE boosting
yeast3	0.7829	**0.8024**
pocker-8-9_vs_5	0.1738	**0.2855**
winequal-red-8_vs_6-7	0.2371	**0.3040**

TABLE I
F1-SCORES FOR DIFFERENT IMBALANCED DATASETS FOR GRADIENT BOOSTING AND SMOTE BOOSTING. BEST VALUES ARE IN BOLD.

Another popular idea used in many imbalanced classification methods is to augment the training sample by adding examples to the sample (Oversampling) or dropping examples from the sample (Undersampling) in order to level the class balance in it [14], [15]. One of the most used approaches is SMOTE [16], which adds new synthetic objects to the training sample by deriving them from the examples in the minority class.

A large family of approaches to imbalanced classification relies on boosting of weak classifiers. Boosting has well studied theoretical properties [17], [18] and exhibits superior performance compared to many other algorithms [19]. There exists a SMOTE-inspired modification of boosting [13], which does a resampling step before adding a new weak classifier to the ensemble. For a more detailed review of usage of ensembles in imbalanced classification see [20] and references therein.

Advances in development of new boosting schemes rely on selection of the right term for model complexity penalty, based on the generalization upper bound and the Rademacher complexity of weak classifier model class [21]. Recently the naive upper bound for classification error provided by Koltchinskii [22] was improved by Cortes et al. and used in Deep Boosting algorithm, which picks ensemble elements that minimize the corresponding learning bound at each step of boosting [23].

In this paper we improve the quality of SMOTE boosting for imbalanced classification problems by enhancing the procedure with insights from Deep Boosting and introducing a correction of sampling probability for objects using the classification margin for boosting algorithms. We also prove an upper bound for the Rademacher complexity for the SMOTE oversampling approach and examine how to get better models with more precise estimates for the Rademacher complexity.

The paper is structures as follows: we begin by describing the proposed approaches and the related theoretical properties, and then discuss the results of the numerical experiments applying the proposed approaches to the typical imbalanced classification problems. Appendix contains additional experi-

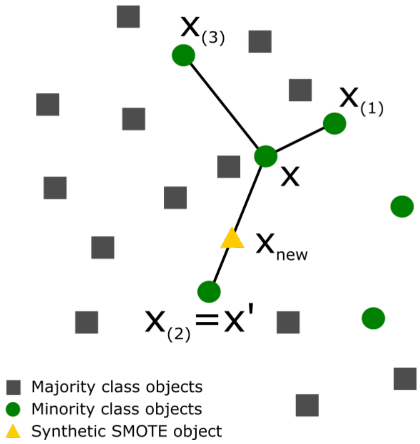

Fig. 1. Generation of $\mathbf{x}_{\texttt{new}}$ using SMOTE.

ments and proofs.

II. Algorithms

We consider an imbalanced binary classification problem: there is a training sample $S = \{(\mathbf{x}_i, y_i = y(\mathbf{x}_i))\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$, and the goal is to construct a classifier $f(\mathbf{x}) : \mathbb{R}^d \mapsto \{-1, 1\}$ such that $f(\mathbf{x}) \approx y(\mathbf{x})$. Furthermore let $n_{\texttt{maj}} = \sum_{i=1}^n [y_i = -1]$ be the number of objects in the major class, and $n_{\texttt{min}} = \sum_{i=1}^n [y_i = 1]$ – the number of objects in the minor class. Since the problem is imbalanced, the imbalance ratio $IR(S) = \frac{n_{\texttt{maj}}}{n_{\texttt{min}}} \gg 1$. Let $\{w_{\mathbf{x}_i}\}_{i=1}^n$ denote the sample weights of S.

A. SMOTE boosting

One of the most popular approaches to dealing with imbalanced classification problem is SMOTE (Synthetic Minority Oversampling Technique) [13]. The key idea is to introduce additional synthetic objects to the minor class. Each new object is generated through the following steps (see Figure 1 for an illustration):

1) Pick a random object $\mathbf{x}$ from the minor class ($y(\mathbf{x}) = 1$, $(\mathbf{x}, y(\mathbf{x})) \in S$).
2) Uniformly sample one object $\mathbf{x}'$ from the set of the k nearest neighbours of $\mathbf{x}$ ($\{\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(k)}\}$) within the same class ($y(\mathbf{x}_{(j)}) = 1$) of the original sample S (previously generated objects do not contribute to the synthesis).
3) Synthesize a new object $\mathbf{x}_{\texttt{new}} = a\mathbf{x} + (1-a)\mathbf{x}'$, where a is a random variate from the uniform distribution over $[0, 1]$. Set the sample weight of this object (for training new weak learner) to $w_{\mathbf{x}_{\texttt{new}}} = a w_{\mathbf{x}} + (1-a) w_{\mathbf{x}'}$.
4) Add the object $(\mathbf{x}_{\texttt{new}}, 1)$ with weight $w_{\mathbf{x}_{\texttt{new}}}$ to the training sample.

There are two parameters of the algorithm: the size of the neighbourhood k and the number of synthetic objects added to the sample at each step of boosting. The number of synthetic objects is often derived from the desired resampling ratio $r = \frac{IR(S')}{IR(S)}$, where S' is the sample S with synthetic samples generated using SMOTE. The size of the neighbourhood k and the resampling ratio r are usually chosen through cross-validation.

To perform boosting on the basis of SMOTE we generate new synthetic objects before constructing a new weak learner and select weights for them according to the procedure above.

B. Deep Boosting

Let us consider a family of classifiers H, e.g. the set of all decision trees with the number of nodes not exceeding m. The empirical *Rademacher complexity* of H over a sample S of size n measures the richness of H in terms of accurately the best classifier from H correlates with the random noise. In particular:

$$\widehat{\mathcal{R}}_S(H) = \frac{1}{n} \mathrm{E}_\sigma \left[\sup_{h \in H} \sum_{i=1}^n \sigma_i h(\mathbf{x}_i) \right],$$

where $\sigma = \{\sigma_i\}_{i=1}^n$ are the Rademacher variables: i.i.d. random variables taking values in $\{-1, 1\}$ with equal probability and independent from the sample S.

The *Rademacher complexity* of the family H for i.i.d. samples of size n from a distribution D on $(\mathbf{x}, y)$ is

$$\mathcal{R}_n(H) = \mathrm{E}_{S \sim D^n} \left[\widehat{\mathcal{R}}_S(H) \right].$$

We consider an ensemble of the form $f(\mathbf{x}) = \sum_{t=1}^T \alpha_t h_t(\mathbf{x})$ with each $h_t(\mathbf{x}) \in H_{k_t}$ and $k_t \in \{1, \ldots, N\}$. Thus the weak classifiers are picked from one of the complexity families $H_1, \ldots, H_N$.

We want to bound the theoretical binary misclassification error $R(f)$ with the ρ-empirical error $\hat{R}_{S,\rho}(f)$, given by

$$R(f) = \frac{1}{n} \sum_{i=1}^n [y_i \neq f(\mathbf{x}_i)] = \mathrm{E}_{(x,y) \sim D} \left[1_{yf(x) \leq 0} \right],$$
$$\widehat{R}_{S,\rho}(f) = \mathrm{E}_{(x,y) \sim S} \left[1_{yf(x) \leq \rho} \right].$$

The theorem below provides an upper bound for the misclassification error:

Theorem 1 ([23]): For fixed $\rho > 0$ for each $\delta > 0$ with probability at least $(1-\delta)$ over the draws of $S \sim D^n$ for each $f = \sum_{t=1}^T \alpha_t h_t \in \mathcal{F}$ it holds that:

$$R(f) \leq \widehat{R}_{S,\rho}(f) + \frac{4}{\rho} \sum_{t=1}^T \alpha_t \mathcal{R}_n(H_{k_t}) + \\ + \frac{2}{\rho} \sqrt{\frac{\log N}{n}} + \sqrt{\frac{\log N}{n} \left\lceil \frac{4}{\rho^2} \log \left[\frac{\rho^2 n}{\log N} \right] \right\rceil + \frac{\log \frac{2}{\delta}}{2n}}.$$

Essentially, the Deep Boosting minimizes an tractable approximation of this generalization upper bound, [23].

C. Deep SMOTE boosting

To apply Deep Boosting with SMOTE resampling at each step for the ensemble construction we need to estimate the empirical Rademacher complexity $\widehat{\mathcal{R}}_S(H)$ for the family of weak learners H over the sample S' generated by the SMOTE procedure, sec.II-A.

The following theorem provides an upper bound for the Rademacher complexity for the family of decision trees with fixed number of nodes, while similar results can be obtained for any weak classifier with known VC dimension.

Theorem 2: The Rademacher complexity of a decision tree for a sample of size n and number of new synthetic objects $\tilde{n}$ can be upper bounded by

$$R_n(H) \leq \sqrt{\frac{(4m+2)\log_2(d+2)\log(n+\tilde{n}+1)}{n+\tilde{n}}},$$

where m is the number of nodes for decision trees in H and d is the input dimension.

The proof follows the similar proof in section 4 of [23]. The VC-dimension of $\mathcal{T}_{m,d}$ — a family of all decision trees with m nodes and input dimension d can be upper bounded by $2(m+1)\log_2(d+1)$, see e.g. [24], [25]. For any class of functions H it holds that $\mathcal{R}_n(H) \leq \sqrt{\frac{2\mathrm{VC-dim}(H)\log(n+1)}{n}}$. As we consider a sample of size $n+\tilde{n}$ we get the desired upper bound for the Rademacher complexity.

This modification of the upper bounds of the Rademacher complexity is naive, because after resampling the new dataset is no longer i.i.d. It is desirable to further improve the upper bound for the family of decision trees taking into account the violation of i.i.d. and imbalanced nature of the resampled dataset S'.

D. Improvement of Deep Boosting using margin

The classification margin for a boosting algorithm for some object $(\mathbf{x}, y)$ is defined as follows:

$$M(\mathbf{x}, y) = \frac{y\sum_{t=1}^{T}\alpha_t h_t(\mathbf{x})}{\sum_{t=1}^{T}\alpha_t},$$

where $h_1, ..., h_T$ are the base classifiers returned by a boosting algorithm on the training set.

Margin-based Deep SMOTE boosting differs from common Deep SMOTE boosting algorithm in that it specifies a new scheme for sampling minor class objects during the synthesis of new objects. In Deep SMOTE boosting algorithm a minor class object for SMOTE is selected in a uniformly random way among all examples of the minor class: the probability that an object is chosen is

$$p(\mathbf{x}_i) = \begin{cases} \frac{1}{n_{\mathtt{min}}}, & \text{if } y_i = 1, \\ 0, & \text{otherwise.} \end{cases}$$

In margin-based Deep SMOTE boosting the selection probability is based on the classification margin:

$$p(\mathbf{x}_i) = \begin{cases} \frac{M(\mathbf{x}_i, y_i)}{\sum_{j=1}^{n}[y_j=1]M(\mathbf{x}_j, y_j)}, & \text{if } y_i = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Therefore objects from the minor class which are misclassified and have a strong margin are chosen for SMOTE-synthesis more often. Due to this adjustment of the selection probabilities, the SMOTE procedure generates more objects in the general region where the current ensemble has the largest misclassification error, thereby improving the class balance and the ensemble performance after this boosting iteration in the area.

III. Experiments

In this section we discuss the results of the numerical experiments aimed at comparing the proposed Margin-based Deep SMOTE boosting and common Deep SMOTE boosting algorithms against the state-of-the alternatives: SMOTE boosting, Deep boosting, AdaBoosting and Gradient boosting using typical datasets for testing in imbalanced classification problems.

The code for the Margin-based and common Deep SMOTE boosting, as well as the Deep boosting algorithms is available at the Github repository https://github.com/natalikozlo/margin_deep_smote. We use `xgboost` [12] implementation of the Gradient Boosting and perform standard tuning for handling of imbalanced classification problems for it.

A. Quality measures

Typically in imbalanced classification problems we want to increase the *recall*, given by the ratio $\frac{\mathtt{TP}}{\mathtt{TP}+\mathtt{FP}}$, while keeping *Precision*, $\frac{\mathtt{TP}}{\mathtt{TP}+\mathtt{FN}}$ at a fixed level, where TP, TN, FP, and FN are the number of true positives, true negatives, false positives and false negatives in the test sample, respectively. This trade-off in the imbalanced classification problems is better captured by the *F1 score*, which is the harmonic mean of *Precision* and *Recall*

$$F1\ score = 2\frac{Precision \cdot Recall}{Precision + Recall}.$$

B. Datasets

We use the following datasets of various input dimension d, sample size n and imbalanced ratio IR in our experiments:

1) Datasets from KEEL repository [26], which is the most popular repository for datasets with imbalanced classes. In this paper we use datasets with relatively high sample size, which are listed in Table II.
2) Datasets that are often used for benchmarking of imbalanced classification algorithms, and that are frequently used in papers on SMOTE are given in Table III.
3) OCR-17 dataset, which was used for benchmarking in Deep Boosting algorithm. The dataset was altered by dropping a significant part of the objects with label "7", so as to change the class balance in the training sample in favour of objects with label "1" (see Table III).

C. Benchmarking of the Deep SMOTE boosting

We compare the following algorithms: AdaBoost, SMOTE boosting (based on AdaBoost), the Deep boosting and the Deep SMOTE boosting on datasets from Table III to test Deep boosting algorithm in combination with SMOTE on datasets frequently used in papers on SMOTE. The resampling approaches which generally performed worse than SMOTE boosting according to experiments in Section A are not tested here.

The optimal hyperparameters of each algorithm are selected through grid search with cross-validation. The number of

Dataset	d	n	IR
yeast-1-2-8-9_vs_7	8	947	30.57
yeast-0-2-5-6_vs_3-7-8-9	8	1004	9.14
winequal-white-3-9_vs_5	11	1482	58.28
yeast6	8	1484	41.40
yeast5	8	1484	32.73
winequal-red-8_vs_6-7	11	855	46.50
yeast4	8	1484	28.10
yeast-0-2-5-7-9_vs_3-6-8	8	1004	9.14
winequal-white-3_vs_7	11	900	44.00
poker-8-9_vs_5	10	2075	82.00
yeast3	8	1484	8.10
page-blocks0	10	5472	8.79
segment0	19	2308	6.02
vehicle0	18	846	3.25

TABLE II
SELECTED DATASETS FROM KEEL REPOSITORY

Dataset	d	n	IR
OCR-17 (imb.)	784	8597	9.14
Mammography	6	11183	2.30
Phoneme	6	5404	2.90
Satimage	36	6435	9.70

TABLE III
DATASETS FOR TESTING OF SMOTE

synthetic objects are picked from $[100, 200, 300, 400, 500]$, the number of neighbours – from $[3, 5, 7, 9]$, and the depth of trees vary between 2 and 12. Parameters λ and β for the Deep Boosting and the Deep SMOTE boosting are selected from $\{10^{-3}, 10^{-4}, \ldots 10^{-7}\}$.

The results for the best hyperparameters are provided in Table IV. Figure 2 depicts the dependence of the *F1 score* on number of boosting rounds for the "Phoneme" dataset.

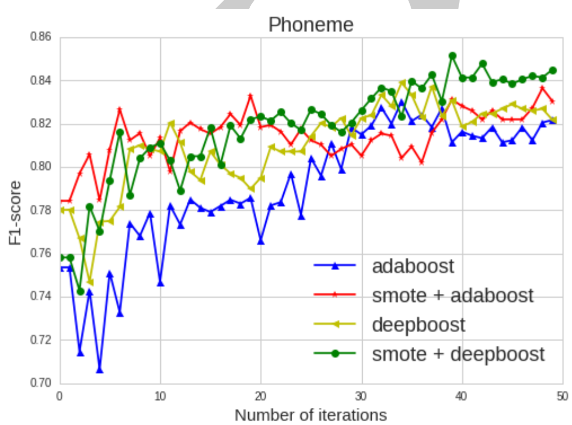

Fig. 2. Dependence of F1-score on number of trees in the ensemble for the dataset Phoneme

Provided results suggest that the Deep SMOTE boosting is better than the existing alternatives for imbalanced classification: it performs better than SMOTE boosting, one of the best existing approaches to imbalanced datasets, and the Deep boosting.

Dataset	Ada(Boost)	AdaSMOTE	Deep	DeepSMOTE
Mammography	0.6625	0.6687	0.6799	**0.6903**
Phoneme	0.8436	0.8472	0.8514	**0.8608**
Satimage	0.5642	0.5923	0.6086	**0.6091**
OCR-17 (imb.)	0.9833	0.9861	0.9859	**0.9916**

TABLE IV
COMPARISON OF F1-SCORES FOR DIFFERENT BOOSTING APPROACHES

Dataset	Gradient	SMOTE	SMOTE+Margin
yeast3	0.7829	0.8150	**0.8234**
page-blocks0	0.8885	**0.8914**	0.8805
segment0	0.9909	0.9955	**0.9970**
vehicle0	0.9401	**0.9600**	0.9556
yeast-1-2-8-9_vs_7	0.3462	0.4291	**0.4510**
yeast-0-2-5-6_vs_3-7-8-9	0.6008	0.6689	**0.6785**
winequal-white-3-9_vs_5	0.2381	**0.4054**	0.4000
yeast6	0.5404	0.6309	**0.6324**
yeast5	0.7973	**0.8521**	0.8317
winequal-red-8_vs_6-7	0.2371	0.3632	**0.4000**
yeast4	0.3964	0.5355	**0.5526**
yeast-0-2-5-7-9_vs_3-6-8	0.8190	0.8460	**0.8557**
winequal-white-3_vs_7	0.3333	0.5033	**0.5156**
poker-8-9_vs_5	0.1738	0.3300	**0.3492**

TABLE V
COMPARISON OF GRADIENT BOOSTING (GRADIENT), THE DEEP SMOTE BOOSTING (SMOTE) AND THE MARGIN-BASED DEEP SMOTE BOOSTING (SMOTE+MARGIN) BY *F1 score*

D. *Introducing margin into the Deep SMOTE boosting*

The aim of this set of experiments is to test if the margin-based selection probabilities improve the resampling procedure for the Deep SMOTE boosting and to compare it with state-of-the-art realization of boosting `xgboost` [12] on imbalanced datasets from Table II.

The setup is as follows: optimal tree depth is chosen with grid-search from 3 to 12, the Deep Boosting parameters λ and β are fixed to 10^{-6}, because these values are nearly the optimal ones for most problems, and the number of neighbours k is set to 5 according to the recomendations in the state-of-the-art. Resampling multiplier r vary from 1.5 to 8.5.

We optimize the hyperparameters for the `xgboost` by grid-search over the depth of tress ($[3, 12]$) and the learning rate ($[0.05, 0.1, 0.5, 0.7, 1, 1.5]$). We use the logistic regression loss and logistic regression loss function before logistic transformation, and we re-weight the sample according to the imbalance ratio of each dataset. Number of boosting rounds was fixed to 100. To deal with imbalanced classification we vary the *max delta step* boosting parameter, as suggested in documentation of `xgboost` [12].

Results of cross-validation are presented in Table V. Dependence of the *F1 score* on the resampling multiplier for the "winequal-white-3_vs_7" is illustrated in Figure 3.

We see that the Margin-based Deep SMOTE boosting is better than both the Deep SMOTE boosting, which is designed to tackle imbalanced classification problems, and Gradient boosting.

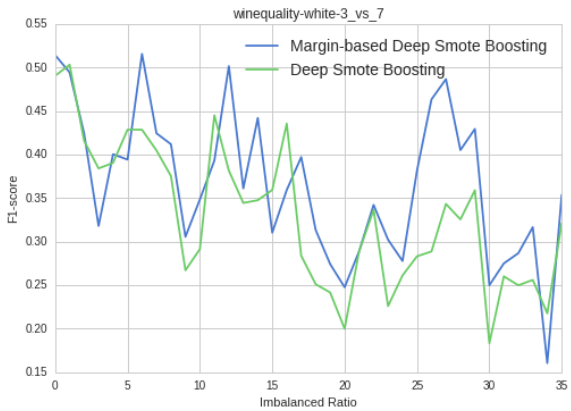

Fig. 3. Dependence of the *F1 score* on the imbalance ratio of the augmented sample

E. Properties of trees in ensembles

Tables VI and VII present properties of the decision trees in the final ensemble for two datasets: "Mammography" and "Satimage". We select the optimal hyperparameters using cross-validation and compare them across 4 different algorithms: AdaBoost, the SMOTE boosting, the Deep boosting and the Deep SMOTE boosting. It turns out that the mean depth of the decision tree is larger for "deep" counterparts of algorithms: trees constructed with the Deep boosting are deeper than trees from AdaBoost, and trees from the Deep SMOTE boosting are generally deeper than trees for the simple SMOTE boosting. Also "deep" counterparts require less boosting rounds T (trees in the ensemble), while offering better *F1 scores*.

Metric	Ada(Boost)	SMOTE	Deep	DeepSMOTE
F1 score	0.6625	0.6687	0.6799	**0.6903**
T	50	46	13	26
Mean depth	4	4	6.23	10.62
Max depth	4	4	7	12

TABLE VI
PROPERTIES OF TREES IN ENSEMBLES FOR DATASET "MAMMOGRAPHY"

Metric	Ada(Boost)	SMOTE	Deep	DeepSMOTE
F1 score	0.5642	0.5923	0.6086	**0.6091**
T	50	50	42	38
Mean depth	4	8	7.5	8.915
Max depth	4	8	8	12

TABLE VII
PROPERTIES OF TREES IN ENSEMBLES FOR DATASET "SATIMAGE"

IV. CONCLUSION

We introduced two new approaches to tackle the issue of class imbalance in the binary classification problems: the Deep SMOTE and the Margin-based Deep SMOTE boosting. These approaches offer significant improvement of the classification quality over the state-of-the-art algorithms. The presented experimental evidence suggests that the main sources of improvement are the use of more complex base learners and the adoption of the margin-based selection probabilities for SMOTE resampling procedure.

In addition we provide more accurate upper bounds fro the Rademacher complexity for decision trees, as common estimates appear to be too loose. Complexity penalties based on these estimates lead to ensembles of models better suited for imbalanced classification, as they take into account class imbalance.

ACKNOWLEDGMENTS

The authors would like to thank Evgeny Burnaev for his constant help during the work on this project, Nikita Zhivotovskiy for useful discussions and Ivan Nazarov for the proofreading of the article and another round of discussions. The research in Section III of this paper was supported by the RFBR grants "16-01-00576 A" and "16-29-09649 ofi_m"; the research, presented in other sections, was supported solely by the Russian Science Foundation grant (project "14-50-00150").

APPENDIX

COMPARISON OF SMOTE WITH UNDER AND OVER SAMPLING

Here we compare SMOTE with the classic undersampling and oversampling techniques for the Deep Boosting ensembles.

For the experiment the depth of the decision trees is chosen by grid-search over values from 3 to 12. The hyperparameters λ and β of the Deep Boosting algorithm are 10^{-6}, and the number of neighbours k fro SMOTE resampling is 5.

Resampling multiplier r is adjusted to keep the imbalance ratio in the augmented sample in the range 1.5 to 8.5.

- Undersampling: we remove $\frac{r-1}{r} n_{\mathtt{maj}}$ objects of the major class
- Oversampling: we add $(r-1)n_{\mathtt{min}}$ objects of the minor class
- SMOTE: similar to Oversampling.

Dataset	SMOTE	Oversampling	Undersampling
yeast3	**0.8052**	0.7945	0.7671
page-blocks0	0.8909	0.9115	0.8596
vehicle0	**0.9873**	0.9744	0.9250
yeast-1-2-8-9_vs_7	**0.7273**	0.6667	0.6000
yeast-0-2-5-6_vs_3-7-8-9	0.7692	0.8108	0.7568
yeast6	0.8333	0.8571	0.8333
yeast5	**0.9474**	0.9474	0.8000
winequal-red-8_vs_6-7	**0.6667**	0.5714	0.4000
yeast4	**0.6316**	0.6207	0.5714
yeast-0-2-5-7-9_vs_3-6-8	**0.9500**	0.9268	0.9048

TABLE VIII
F1 score FOR SMOTE, OVERSAMPLING, UNDERSAMPLING

SMOTE is better than Oversampling and Undersampling approaches. Undersampling is worse almost on every dataset due to the small size of an updated sample.

IR SMOTE i.i.d.	0.5 no	0.1 no	0.1 yes not i.i.d	0.1 yes i.i.d
poker-8-9_vs_5	0.8841	0.9807	0.9065	0.9659
winequal-red-8_vs_6-7	0.7660	0.9360	0.9115	0.9286
winequal-white-3_vs_7	0.7382	0.9222	0.9058	0.9202
yeast-0-2-5-6_vs_3-7-8-9	0.7184	0.9427	0.9157	0.9359
yeast-1-2-8-9_vs_7	0.7263	0.9469	0.9208	0.9378
yeast6	0.6417	0.9217	0.8983	0.9155

TABLE IX
ESTIMATES OF THE RADEMACHER COMPLEXITY

ESTIMATION OF THE RADEMACHER COMPLEXITY

The Rademacher complexity is a lose estimate of the model complexity, moreover often we need to use some upper bounds of the Rademacher complexity in applied problems. Also the Rademacher complexity doesn't take into account an intrinsic nature of the data e.g. imbalance of classes or non-i.i.d distribution of class labels in the sample. In this section we examine how these two problems affect the accuracy of classifiers constructed using penalties based on the Rademacher complexity.

For real datasets we use the following workflow:

1) Select imbalanced ratio IR (0.1 or 0.5)
2) Generate new objects using SMOTE to get the selected imbalanced ratio.
3) Generate labels for new objects for estimation of the Rademacher complexity or according to non-i.i.d. nature of the data: if we have a synthetic object generated from two objects $\mathbf{x}$, $\mathbf{x}'$, then generate a label for this object σ according to the labels of $\mathbf{x}$ and $\mathbf{x}'$ when estimating the Rademacher complexity.

So, we consider 4 setups for the experiment: either SMOTE non-i.i.d generation of labels or i.i.d generation of labels, and either generation of the Rademacher random variables or generation of binomial random variables with probability to get 1 selected using imbalance ratio.

For each case we train trees with different depths (in range $[3, 15]$) and select the best depth, to estimate the Rademacher complexity we generate labels $\{\sigma_i\}$ 100 times.

In Table IX there are Rademacher complexity estimates for the described 4 cases. We can see that using SMOTE we increase the modified Rademacher complexity, and for non-i.i.d. data with synthetic objects the Rademacher complexity is lower.

REFERENCES

[1] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.

[2] Y. Tang, Y. Zhang, N. Chawla, and S. Krasser, "Svms modeling for highly imbalanced classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 281–288, 2009.

[3] E. Burnaev and D. Smolyakov, "One-class svm with privileged information and its application to malware detection," in *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 273–280.

[4] J. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4626–4636, 2009.

[5] S. Alestra, C. Bordry, C. Brand, E. Burnaev, P. Erofeev, A. Papanov, and C. Silveira-Freixo, "Application of rare event anticipation techniques to aircraft health management," in *Advanced Materials Research*, vol. 1016. Trans Tech Publ, 2014, pp. 413–417.

[6] Y. Sun, A. Wong, and M. Kamel, "Classification of imbalanced data: A review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

[7] S. Kotsiantis, D. Kanellopoulos, and P. e. a. Pintelas, "Handling imbalanced datasets: A review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 25–36, 2006.

[8] V. Ishimtsev, I. Nazarov, A. Bernstein, and E. Burnaev, "Conformal k-nn anomaly detector for univariate data streams," *arXiv preprint arXiv:1706.03412*, 2017.

[9] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," *arXiv preprint arXiv:1305.1707*, 2013.

[10] E. Burnaev, P. Erofeev, and D. Smolyakov, "Model selection for anomaly detection," *arXiv preprint arXiv:1707.03909*, 2017.

[11] E. Burnaev, P. Erofeev, and A. Papanov, "Influence of resampling on accuracy of imbalanced classification," in *Eighth International Conference on Machine Vision*. International Society for Optics and Photonics, 2015, pp. 987 521–987 521.

[12] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[13] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," *Knowledge Discovery in Databases: PKDD 2003*, pp. 107–119, 2003.

[14] C. Drummond and R. e. a. Holte, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11. Citeseer Washington DC, 2003.

[15] R. Batuwita and V. Palade, "Efficient resampling methods for training support vector machines with imbalanced datasets," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010, pp. 1–8.

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

[17] E. Burnaev and P. Prikhodko, "On a method for constructing ensembles of regression models," *Automation and Remote Control*, vol. 74, no. 10, pp. 1630–1644, 2013.

[18] R. Schapire, "The boosting approach to machine learning: An overview," pp. 149–171, 2003.

[19] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *J. Mach. Learn. Res*, vol. 15, no. 1, pp. 3133–3181, 2014.

[20] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2012.

[21] S. Boucheron, O. Bousquet, and G. Lugosi, "Theory of classification: A survey of some recent advances," *ESAIM: probability and statistics*, vol. 9, pp. 323–375, 2005.

[22] V. Koltchinskii and D. Panchenko, "Empirical margin distributions and bounding the generalization error of combined classifiers," *Annals of Statistics*, pp. 1–50, 2002.

[23] C. Cortes, M. Mohri, and U. Syed, "Deep boosting," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1179–1187.

[24] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2012.

[25] Y. Mansour, "Pessimistic decision tree pruning based on tree size," in *ICML*, 1997, pp. 195–201.

[26] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, 2011.