Expert Systems With Applications 201 (2022) 117130

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Improving imbalanced classification using near-miss instances

Akira Tanimoto [a,b,c,*], So Yamada [a], Takashi Takenouchi [d,c], Masashi Sugiyama [c,e], Hisashi Kashima [b,c]

[a] NEC, Shimonumabe 1753, Nakahara-ku, Kawasaki, 211-8666, Japan
[b] Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
[c] RIKEN AIP, Nihonbashi 1-chome Mitsui Building, 15th floor, 1-4-1 Nihonbashi, Chuo-ku, Tokyo, 103-0027, Japan
[d] National Graduate Institute for Policy Studies, 7-22-1 Roppongi, Minato-ku, Tokyo, 106-8677, Japan
[e] The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan

ARTICLE INFO

Keywords:
Imbalanced classification
Learning using privileged information
Generalized distillation

ABSTRACT

The class imbalance is a major issue in classification, i.e., the sample size of a rare class (positive) is often a performance bottleneck. In real-world situations, however, "near-miss" positive instances, i.e., negative but nearly-positive instances, are sometimes plentiful. For example, natural disasters such as floods are rare, while there are relatively plentiful near-miss cases where actual floods did not occur but the water level approached the bank height. We show that even when the true positive cases are quite limited, such as in disaster forecasting, the accuracy can be improved by obtaining refined label-like side-information "positivity" (e.g., the water level of the river) to distinguish near-miss cases from other negatives. Conventional cost-sensitive classification cannot utilize such side-information, and the small size of the positive sample causes high estimation variance. Our approach is in line with learning using privileged information (LUPI), which exploits side-information for training without predicting the side-information itself. We theoretically prove that our method reduces the estimation variance, provided that near-miss positive instances are plentiful, in exchange for additional bias. Results of extensive experiments demonstrate that our method tends to outperform or compares favorably to existing approaches.

1. Introduction

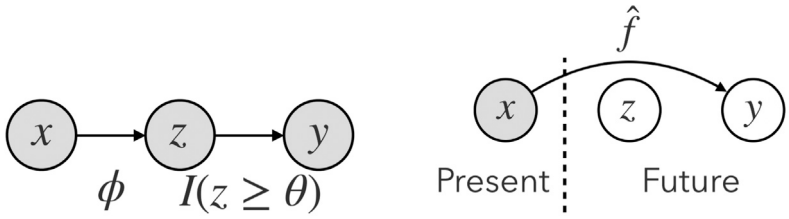
Class imbalance is often a major problem in real-world data analysis (Haixiang et al., 2017; Japkowicz & Stephen, 2002), since the class of interest (i.e., the positive class) often corresponds to rare events, such as disasters, accidents, diseases (Haixiang et al., 2017), abnormalities (Fuqua & Razzaghi, 2020), or conversions in advertisement recommendation tasks (Lee et al., 2012). In such cases, the performance will be limited by the size of the positive training sample. However, among such real-world imbalanced problems, there are cases where "near-miss" instances, i.e., negative but nearly-positive instances, are relatively plentiful.

In flood prediction (Cloke & Pappenberger, 2009), for example, actual floods are rare, while there are relatively many near-miss cases where the water level approached the height of the riverbank. Also, in condition-based maintenance, the condition of each piece of equipment is monitored regularly, and the maintenance is carried out to keep the condition not to reach an alarm-level (Lee et al., 2006). While actual accidents are rare, there are many near-miss incidents where the condition approaches the alarm-level (Li & Nilkitsaranont, 2009; van der Schaaf, 1995). Furthermore, sales forecast for new products such as songs (Herremans et al., 2014) or books (Chang & Lai, 2005) are difficult due to the skewness of the sales distribution (Hendricks & Sorensen, 2009). If one needs only to know whether the sales exceed a threshold, such as a break-even point for deciding to publish, the task would be a classification task. While hit books are rare, we often have plentiful records of near-miss hit books whose sales are slightly below the break-even point.

Exploiting such near-miss data is a well-known heuristic in the field of accident prevention. Heinrich et al. (1980), Jones et al. (1999), and Barach and Small (2000) argued the importance of collecting data not only regarding actual accidents but also regarding near-miss incidents and suggested to take measures to prevent them. To the best of our knowledge, exploiting near-miss data has not yet been sufficiently investigated in machine learning literature. We therefore show that this lesson in accident prevention applies to machine learning, i.e., even when the number of true positive cases is quite limited, the accuracy

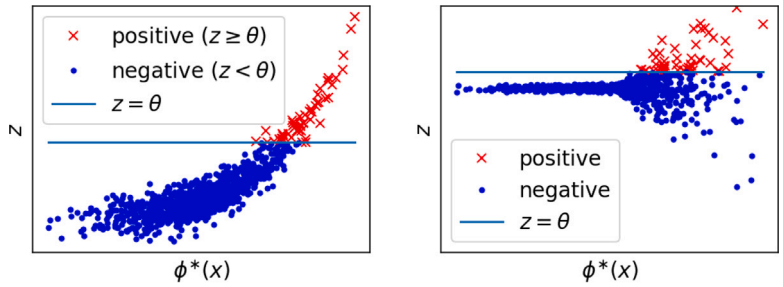
* Corresponding author at: NEC, Shimonumabe 1753, Nakahara-ku, Kawasaki, 211-8666, Japan.
E-mail addresses: a.tanimoto@nec.com (A. Tanimoto), soh-yamada@nec.com (S. Yamada), t-takenouchi@grips.ac.jp (T. Takenouchi), sugi@k.u-tokyo.ac.jp (M. Sugiyama), kashima@i.kyoto-u.ac.jp (H. Kashima).

https://doi.org/10.1016/j.eswa.2022.117130
Received 7 September 2020; Received in revised form 30 November 2021; Accepted 29 March 2022
Available online 10 April 2022



(a) Data generation model (for the training phase). (b) Prediction model (for the inference phase).

Fig. 1. Our assumed graphical models for training and inference. Gray nodes represent observed variables at each phase. (a) Our assumed data generation model. $x \in \mathbb{R}^d$ is a feature vector, $z \in \mathbb{R}$ is a numerical mediator variable that represents "positivity", I is an indicator function, θ is a threshold, and $y := I(z \geq \theta)$ is the binary label. (b) Our employed prediction model. z typically represents a future condition; thus it is not available in the test phase, and need not be predicted. The only prediction target y is whether or not the condition z exceeds a given threshold. Thus, we do not predict z; rather, we predict y directly.



(a) $p(z|x) = \mathcal{N}(\mu(\phi^*(x)), \psi)$. (b) $p(z|x) = \mathcal{N}(\mu, \psi(\phi^*(x)))$.

Fig. 2. Toy examples for the setting illustrated in Fig. 1(a). $\phi^*(x) = w^{*\top}x$ is the true scoring function. (a) Toy data generated by a generalized linear model that we used in our experiments. (b) Another data with heteroscedastic noise, showing how regression- and rank-based approaches may fail.

can be improved by obtaining additional information to identify the near-miss cases.

Such additional information we assume is "positivity" $z \in \mathbb{R}$ given in the training phase as in Fig. 1(a). The label y is defined by whether or not z exceeds a given threshold θ. Fig. 2 shows synthetic examples. Positivity z represents, for example, the future water level in flood prediction, the future condition of equipment in condition-based maintenance, or the sales of the new book. Note that, since z typically denotes some future condition, z is not available in the inference phase.

Since the final goal is to predict the binary label y, a naive approach is to throw away z and train a classifier only from (x, y) pairs.

Imbalanced classification using binary labels has been actively studied (Haixiang et al., 2017; Leevy et al., 2018).

In particular, when the number of positive data is small, cost-sensitive learning (Elkan, 2001) is often used to cancel the estimation bias due to the class imbalance, in which misclassification costs for false positives and false negatives are unequal. While it converges asymptotically to the Bayes optimal solution, estimation variance is high, as we theoretically prove in Section 4 and experimentally demonstrate in Section 5.3.

Many methods have been proposed in this context, including those based on under- and oversampling with synthetic data generation (Barua et al., 2012; Chawla et al., 2002; He et al., 2008; Wei et al., 2020) and hybrid/ensemble methods (Chawla et al., 2003; Kim & Sohn, 2020; Seiffert et al., 2009). We also make comparisons with representative ones of these in Section 5.4.

A tempting approach for avoiding high estimation variance is regression, i.e., estimating the generative model ϕ in Fig. 1(a). While here we never confront the imbalance issue, naive regression methods cannot convey information other than the conditional mean $\mathrm{E}[z|x]$, and fail when the noise level is not constant, as illustrated in Fig. 2(b). Further discussion of this approach and the relation of z and $p(y|x)$ is provided in Section 3.2.

We therefore take a direct modeling approach, as in Fig. 1(b), and exploit z as side-information to alleviate the estimation variance. Then, provided that the near-miss positive instances are relatively plentiful with respect to the real positives, we can increase the effective positive rate by regarding the near-miss positive instances as being partly positive. This makes it possible for our method to enjoy reduced estimation variance, as is proved in Section 4.2, in exchange for additional bias, as in Section 4.3. Experimental results given in Section 5.4 indicate the effectiveness of our approach.

Our main contributions are three-fold. First, we propose a new learning algorithm to exploit the positivity z, which is model-agnostic, i.e., it can be incorporated into many off-the-shelf implementations of classifiers. Second, we derive a non-asymptotic bound, which shows the mechanism that our method can reduce the estimation variance via increasing the effective size of the positive sample with the help of near-miss instances, in exchange for additional bias. The bound of the additional bias also gives a characterization of effective positivity information. Lastly, our extensive experiments illustrate the effectiveness of our method compared to the conventional classification methods and the regression- and rank-based approaches.

2. Problem statement

We want to learn a scoring function (decision function) $g : \mathcal{X} \to \mathbb{R}$ that defines a plug-in binary classifier $\hat{y} = I(g(x) \geq 0)$, where $\mathcal{X} \subset \mathbb{R}^d$ is the feature space and I is the indicator function. Given a task-specific threshold θ we learn from the data set $S = \{d_1, d_2, \ldots, d_N\}$, where N is the sample size, and $d_n = (x_n, z_n)$ consists of a feature vector $x_n \in \mathcal{X}$ and a mediator variable $z_n \in \mathbb{R}$, which we refer to as positivity. Note that the positivity z_n is accessible only in the training phase, and we *cannot* use z_n in the test phase. A class label is determined as

$$y_n = I(z_n \geq \theta).$$

Without loss of generality, we hereafter assume $\theta = 0$ (i.e., let $z_n - \theta$ be the new z_n).

Positivity z_n is considered related to a "probabilistic label (soft-label)" $p_n = p(y = 1|x_n)$; however, p_n itself is not given, which represents

the difference from existing soft-label studies (Nguyen et al., 2014; Nguyen, Valizadegan, Seybert et al., 2011; Peng et al., 2014). A detailed discussion of this is given in Section 3.2.

For the evaluation, we adopt a cost-sensitive metric called the weighted accuracy (WA) (Cohen et al., 2006):

$$\mathrm{WA}_N(g) = \frac{1}{N}\sum_n^N \left\{ C_+ I(z_n \geq 0 \wedge g(x_n) \geq 0) + C_- I(z_n < 0 \wedge g(x_n) < 0) \right\}, \tag{1}$$

where C_+ and C_- are task-specific constants for the positive class and the negative class, respectively, as introduced in the cost-sensitive learning framework (Elkan, 2001; Ling & Sheng, 2008; Vasile et al., 2017), and $\wedge$ represents the logical AND. Since we consider the imbalanced case, the accuracy for the rare positive class is usually emphasized, i.e., $C_- < C_+$. We also consider a special case of WA, letting $C_+ = N/2N_+$ and $C_- = N/2N_-$, where $N_+ := \sum_n^N I(z_n \geq 0)$ and $N_- := \sum_n^N I(z_n < 0)$, as balanced accuracy (BA). Here, $(1 - \mathrm{BA})$ is the balanced error rate (BER), which is often adopted in imbalanced problems (Chen & Wasikowski, 2008). We evaluated the performance of a classifier with respect to BA in our experiments.

$$\mathrm{BA}_N(g) = \frac{1}{N}\sum_n^N \left\{ \frac{N}{2N_+} I(z_n \geq 0 \wedge g(x_n) \geq 0) + \frac{N}{2N_-} I(z_n < 0 \wedge g(x_n) < 0) \right\}. \tag{2}$$

3. Learning with positivity

In this section, we propose a proxy loss, a generalization of the cost-sensitive learning to the case in which positivity is obtained, and compare it with another approach, i.e., the generative modeling.

3.1. Proposed loss function

A naive approach for this problem is the cost-sensitive learning which minimizes the convex relaxation of $(\mathrm{const.} - \mathrm{WA}_N)$ (Dmochowski et al., 2010), i.e., its empirical risk is

$$\hat{L}(g) = \frac{1}{N}\sum_n^N \left\{ C_+ y_n \ell(g(x_n)) + C_-(1 - y_n)\ell(-g(x_n)) \right\}, \tag{3}$$

where $\ell(g)$ is the instance-wise loss such as the hinge loss or the negative log-likelihood. As we prove in Section 4, however, the estimation variance is high, and thus the performance would be poor under the limited size of the positive training sample. To overcome this limitation, we propose the following proxy loss that treats near-miss instances as being partly positive.

$$\hat{L}_T(g) = \frac{1}{N}\sum_n^N \left\{ C_{T,+}\sigma(z_n/T)\ell(g(x_n)) + C_{T,-}\sigma(-z_n/T)\ell(-g(x_n)) \right\}, \tag{4}$$

where $\sigma(a) := 1/(1 + \exp(-a))$ is the sigmoid function, T is a hyper-parameter called temperature, $C_{T,+} := C_+ \frac{N_+}{N_{T,+}}$ and $C_{T,-} := C_- \frac{N_-}{N_{T,-}}$ are rebalanced cost parameters, and $N_{T,+} := \sum_n^N \sigma(z_n/T)$. We refer to $\sigma(z/T)$ as the soft-label. Considering that the soft-label goes to the original hard label in the limit of $T \to 0$, i.e, $\lim_{T\to 0}\sigma(z/T) = y$ (except for $z = 0$), our loss function includes the cost-sensitive learning as the limit of $T \to 0$.

Our loss function (4) can be implemented as instance weighting; namely, we duplicate the whole training set for positive and negative parts with weights $C_{T,+}\sigma(z_n/T)$ and $C_{T,-}\sigma(-z_n/T)$, respectively. Then, any off-the-shelf base learner $\mathcal{A}$ can be trained with duplicated instances and weights. The detailed algorithm for the setting of BER minimization is described in Algorithm 1.

One benefit of introducing the soft-label is increasing the effective positive sample size, i.e., $N_+ < N_{T,+}$ for some proper $T > 0$, as is described in Section 4. By increasing the effective positive sample size $N_{T,+}$ and rebalancing the effective total costs of each class, we can reduce the imbalance of cost parameters $C_{T,+}$ and $C_{T,-}$, which results in the reduction of the estimation variance as we prove theoretically in Section 4.2.

Algorithm 1 Learning with positivity

Input: $D = \{(\mathbf{x}_n, z_n)\}_n$, θ, T, and a base learner $\mathcal{A}$
Output: Trained model M

```
for n = 1 to N do
    s_n ← 1 / (1 + exp(−(z_n − θ)/T))
end for
p_{T,+} ← (1/N) Σ_n^N s_n
D' ← {(x_n, y = 1, weight = s_n / p_{T,+}), (x_n, y = 0, weight = (1 − s_n)/(1 − p_{T,+}))}_n
M ← A(D')
return M
```

3.2. Comparison with the generative modeling approach

In this section, we explain the relationship between the positivity z and the conditional probability $p(y|x)$ and clarify the reason why the naive generative modeling approach is not always suitable.

In a similar and well-studied setting called learning on probabilistic labels, the conditional probability $p_n := p(y = 1|x_n)$ or its estimation is given as the label for each instance. The probabilistic labels are typically given by averaging crowd-sourced labels over annotators. Regression-based (Nguyen, Valizadegan, Hauskrech, 2011; Peng et al., 2014) and rank-based methods (Nguyen, Valizadegan, Hauskrech, 2011; Nguyen, Valizadegan, Seybert et al., 2011; Xue & Hauskrecht, 2016, 2017) are proposed for learning on probabilistic labels.

In our setting, the conditional probability is not directly given, but can be expressed as $p(y = 1|x) = \int I(z \geq 0)p(z|x)\mathrm{d}z$. Thus, it might be tempting to model $\hat{p}(z|x)$ and then plug-in as

$$\hat{p}(y = 1|x) = \int I(z \geq 0)\hat{p}(z|x)\mathrm{d}z. \tag{5}$$

Then, since the positivity z is a continuous variable, one need never confront the imbalance issue.

However, this indirect modeling of z is not always suitable. For example, regression methods with homoscedastic noise (i.e., $\mathrm{Var}[z|x]$ is assumed constant) fail if the assumption is not satisfied, as with the distribution illustrated in Fig. 2(b). In this case, these methods tend to learn a constant model $\hat{p}(z|x) = c$ and the plug-in classification model in (5) also ends up in a constant model $\hat{p}(y = 1|x) = c'$, while the true conditional probability $p(y = 1|x)$ is not constant in x. Modeling conditional variance is not always sufficient, either, due to higher moments of $p(z|x)$. We are particularly interested here in the tails of distributions, and, therefore, the higher moments are often dominant for evaluating $p(z \geq 0|x)$. This is why the direct modeling approach is superior in terms of versatility for distributions. The experimental results in Section 5.4 also support the versatility of the proposed method.

3.3. Choice of the soft labeling function σ and the noise robustness

Here we make a note on the noise in the training data and the choice of the soft labeling function σ. Addressing noise is considered important in the imbalanced classification field (Napierała et al., 2010; Natarajan et al., 2017; Sáez et al., 2015). Generally speaking, our approach is considered to be relatively robust to noise. That is, when the true positivity $z = 0.1$ is observed as $z_{\mathrm{obs}} = -0.1$ as a result of noise on z, the binary label y changes abruptly from 0 to 1, while the soft label $\sigma(z/T)$ in the proposed method only changes from 0.48 to 0.52 under the temperature $T = 1$. Here, even when the noise is added in the input

x, if the degree of noise is small, and if we further assume that the conditional probability $p(y = 1|x)$ is continuous (e.g., in the sense of Lipschitz) in x, it can be regarded as equivalent to a small noise on z.

On the other hand, for the case of severe noise, e.g., a completely negative instance $z = -10$ is sometimes observed as completely positive $z_{\text{obs}} = 10$, the noise robustness of our proposed method is only comparable to that of the conventional cost-sensitive learning. One possible solution for such cases is to incorporate a label smoothing technique in the learning from the binary label setting (Natarajan et al., 2017; Szegedy et al., 2016), in which the label is smoothed from $\{0, 1\}$ to (e.g.) $\{0.05, 0.95\}$. Our approach can incorporate this by, e.g., setting the soft labeling function as $\tilde{\sigma}(z/T) = 0.9\sigma(z/T) + 0.05$. The optimal labeling scheme depends on the joint distribution $p(x, z)$. It is desirable to reduce the variance analyzed in Section 4.2 while minimizing the increase in bias analyzed in Section 4.3. This direction, i.e., improving the soft labeling function to increase noise robustness, is a promising future work.

3.4. Comparison with synthetic oversampling methods

Our method proposed in Section 3.1 extends the cost-sensitive learning, which is called the algorithm-level approach in the imbalanced classification field (Krawczyk, 2016). Another well-studied direction is the data-level approach, i.e., synthetic oversampling of positive instances. This direction was pioneered by the synthetic minority oversampling technique (SMOTE) (Chawla et al., 2002) and has been actively studied (Fernández et al., 2018).

While simple over-sampling of positive instances is equivalent to the cost-sensitive learning at the level of its loss function, SMOTE and its variants are clearly distinguished in that they utilize additional inductive biases. For example, SMOTE treats interpolations of neighboring positive instances as positive, which may reflect the convexity of the support of conditional distribution $p(x|y = 1)$ or the cluster assumption (Chapelle et al., 2006). Also, Ali-Gombe and Elyan (2019) proposed generating positive instances by training a generative adversarial network (GAN) for image data. GANs can incorporate with unlabeled instances for generating realistic images, which highlights a new approach of semi-supervised learning for imbalanced classification. It has been suggested that GANs can utilize some kind of inductive bias common to images (Zhao et al., 2018).

While data augmentation methods have been repeatedly shown to be promising, careful consideration should be given as to whether the inductive biases behind them are still valid in our problem setting. A significant difference may come from the direction of causality. Our typical setting is prediction, i.e., the input feature x causes the outcome y with positivity z observed as a mediator variable as in Fig. 1. This is called a causal setting, as opposed to an anti-causal setting, where the label y causes the feature or image x. Schölkopf et al. (2012) have revealed that incorporating the cluster assumption by semi-supervised learning can be helpful only in anti-causal settings. In causal settings, the marginal distribution of the feature $p(x)$ contains no information about the conditional distribution $p(y|x)$. In fact, our experimental results in Section 5.4 also show that SMOTE and its variants only achieve comparable or inferior performance for the cost-sensitive learning.

The inductive bias we are utilizing is in a different direction from this data augmentation approach in the input space. As we analyze in Section 4.3, we assume that the larger positivity values indicate the larger possibility of being positive, which reflects a kind of continuity assumption of the conditional distribution in the positivity space. Therefore, our approach may not only be effective for the settings where SMOTE and its variant are not effective but may also incorporate with them. Investigating the key success factor of these synthetic oversampling methods and extend them to prediction or regression problems is a promising direction as discussed in Krawczyk (2016).

4. Theoretical analysis

In this section, we describe the performance of the proposed method, which includes the conventional cost-sensitive learning method as a special case.

4.1. Setup

We analyze the excess risk, i.e., the difference in the expected risks of estimated and optimal models, using the population version of the proposed loss (4), namely,

$$L_T(g) = \mathop{\mathbb{E}}_{x,z}\left[C_{T,+}\sigma(z/T)\ell(g(x)) + C_{T,-}\sigma(-z/T)\ell(-g(x))\right]. \tag{6}$$

and the cost-sensitive one,

$$L(g) = \mathop{\mathbb{E}}_{x,y}\left[C_+ y\ell(g(x)) + C_-(1-y)\ell(-g(x))\right]. \tag{7}$$

When ℓ is the hinge loss or the negative log-likelihood, (7) can be seen as a tight convex upper bound of (const. − WA) (Dmochowski et al., 2010), and thus good performance is expected asymptotically. Although, when the size of the positive sample is small and its weight C_+ is set large, the estimation variance is high. Our proposed loss (6) treats near-miss instances as being partly positive through soft-labeling function σ, and relaxes the imbalance between the class weights, resulting in reduced estimation variance, as we prove in this section.

The excess risk with respect to the cost-sensitive loss (7) can be decomposed as

$$\begin{aligned}\mathop{\mathbb{E}}_{S}[L(\hat{g}) - L(g^*)] = &\underbrace{\mathop{\mathbb{E}}_{S}[L(\hat{g}) - L_T(\hat{g})]}_{\text{bias 1}} + \underbrace{\mathop{\mathbb{E}}_{S}[L_T(\hat{g})] - \min_{g\in\mathcal{G}} L_T(g)}_{\text{variance}} \\ &+ \underbrace{\min_{g\in\mathcal{G}} L_T(g) - L_T(g^*)}_{\le 0 \text{ by definition}} + \underbrace{L_T(g^*) - L(g^*)}_{\text{bias 2}},\end{aligned} \tag{8}$$

where S is the training set, $\hat{g} := \operatorname{argmin}_{g\in\mathcal{G}} \hat{L}_T(g)$ is the empirical proxy loss minimizer, which depends on S, and $g^* := \operatorname{argmin}_{g\in\mathcal{G}} L(g)$ is the optimal model in assumed model class $\mathcal{G}$.

Although the proposed method is model-agnostic, we add some technical assumptions here for theoretical analysis.

Assumption 1. $\mathcal{G}$ is a bounded linear class; namely, $\mathcal{G} = \{g : g(x; w) = w^\top x, \|w\|_2 \le B\}$.

Assumption 2. The support of $p(x)$ is bounded; namely, $p(\|x\|_2 \le X) = 1$.

Assumption 3. ℓ is 1-Lipschitz and satisfies $\max_{a,a'\in[-BX,BX]} |\ell(a) - \ell(a')| \le c$.

In addition, we replace the cost parameter settings with the population versions:

$$C_{T,+} = C_+\frac{p_+}{p_{T,+}} \text{ and } C_{T,-} = C_-\frac{p_-}{p_{T,-}}, \tag{9}$$

where p_+ and p_- are the expected positive and negative rates, $p_{T,+}$ and $p_{T,-}$ are the expected effective rates of positive and negative, namely, $p_{T,+} = \mathbb{E}_z\left[\sigma(z/T)\right]$ and $p_{T,-} = \mathbb{E}_z\left[\sigma(-z/T)\right]$. This is because the expectation $\mathbb{E}_S\left[N_+/N_{T,+}\right]$ may not exist. Similarly, when we discuss the BER minimization setting in cost-sensitive learning, we set the cost parameters as

$$C_+ = 1/(2p_+) \text{ and } C_- = 1/(2p_-). \tag{10}$$

4.2. Variance reduction

Let us first evaluate the excess risk for our proxy loss, which is denoted as variance in (8).

Theorem 4.1 (*Proxy Loss Minimization Bound*). *Let $\hat{w}$ be a minimizer of the empirical proxy loss $\hat{L}_T$ (4) with cost parameters (9) and w_T^* be a minimizer of the expected proxy loss L_T. Suppose that $\mathcal{G}$, $p(x)$ and ℓ satisfy Assumptions 1–3. The excess risk for L_T will then be bounded as follows:*

$$\mathop{\mathbb{E}}_{S}\left[L_T(\hat{w}_T) - L_T(w_T^*)\right] \le \frac{2BX}{\sqrt{N}}\sqrt{C_+^2\frac{p_+^2}{p_{T,+}} + C_-^2\frac{p_-^2}{p_{T,-}}}.$$

This is given by element-wise upper bounding of the Rademacher complexity, i.e.,

$$\begin{aligned} R(\ell\circ A) :=& R(\{\left(\ell_1(a_1),\dots,\ell_N(a_N)\right) : \mathbf{a}\in A\subset\mathbb{R}^N\}) \\ \le& R(\{\left(\rho_1 a_1,\dots,\rho_N a_N\right) : \mathbf{a}\in A\}), \end{aligned}$$

where $\mathbf{a} := (a_1,\dots,a_N)$ and ρ_n is the Lipschitz constant of ℓ_n. The detailed proof is given in Appendix A. This element-wise evaluation of the Lipschitz constants is the key for a tighter bound since our loss function consists of a small number of element-wise losses that have a large Lipschitz constant $C_{T,+}$ and a large number of one with a small Lipschitz constant $C_{T,-}$.

For the BER minimization setting (10), the bound is rewritten as follows.

Corollary 4.1.1 (*Balanced Loss Minimization Bound*).

$$\mathop{\mathbb{E}}_{S}\left[L_T(\hat{w}) - L_T(w_T^*)\right] \le \frac{BX}{\sqrt{N}}\sqrt{\frac{1}{p_{T,+}} + \frac{1}{p_{T,-}}}. \tag{11}$$

So long as the effective positive rate is much smaller than the effective negative rate, namely, $p_{T,+} \ll p_{T,-}$, the term $1/p_{T,+}$ is dominant in (11). This is why reducing the imbalance between $p_{T,+}$ and $p_{T,-}$ has a critical impact on the variance reduction. From the definition of the soft-label $\sigma(z/T)$, we observe

$$\lim_{T\to 0} p_{T,+} \to p_+ \text{ and } \lim_{T\to\infty} p_{T,+} \to 1/2.$$

Therefore, by using proper $T > 0$, we can increase the effective positive rate $p_{T,+}$, and can attain variance reduction.

Corollary 4.1.1 is also useful to predict the limitation of conventional cost-sensitive learning. Let us assume that $T \to 0$ (then $Np_{T,+} \to Np_+ \simeq N_+$), the model complexity $B = 1$, and the size of the feature space $X = \sqrt{d}$ (each dimension is normalized). Since $p_+ \ll p_-$ holds, the r.h.s. of (11) would be simplified as follows:

$$\text{r.h.s. of (11)} \simeq \sqrt{d/N_+}. \tag{12}$$

Therefore, when the size of the positive sample is smaller than the feature dimension ($N_+ < d$), the variance term would be larger than 1, which is no longer meaningful as an upper bound of the BER. Assuming the bound is tight enough, this implies that there is plenty of room for performance improvement by tuning T when $N_+ < d$ holds, and also experimental results in Section 5.3 agree to this. That is, the conventional cost-sensitive method significantly underperforms the proposed method when $N_+ < d$.

4.3. Bias bound

We next give an upper bound of the bias terms in (8). To simplify the notation, we introduce a random variable η that depends on the soft label $\sigma(z/T)$ as

$$p(\eta|z) = \text{Bernoulli}(\sigma(z/T)).$$

η can be seen as "a potential label that might have been under the given z", and $p(\eta = y) = 1$ when $T \to 0$. By using η, we can bound the bias as follows:

Proposition 4.2 (*Bound of the Bias of the Proxy Loss*). *Suppose that $\mathcal{G}$, $p(x)$, and ℓ satisfy Assumptions 1–3. The bias terms in (8) in the BER minimization setting (10) is upper-bounded as*

$$\begin{aligned} (\text{bias 1} + \text{bias 2}) \le c\,\big\{&\mathrm{TV}\left(p(x|\eta=1), p(x|y=1)\right) \\ &+ \mathrm{TV}\left(p(x|\eta=0), p(x|y=0)\right)\big\}, \end{aligned}$$

where $\mathrm{TV}(p(x), q(x)) := \frac{1}{2}\int |p(x) - q(x)|\,\mathrm{d}x$ *is the total variation distance.*

If we set $T > 0$, the bias might increase, which is bounded using the TV distances, and which depends on the joint distribution $p(x, z)$ and the temperature T. Differently from the distance between the conditional label probabilities $\mathrm{TV}(p(y|x), p(\eta|x))$, these TV distance terms do not necessarily increase as do $p_{T,+} = p(\eta = 1)$. Thus, in the range of reasonably small T, and provided that a reasonable z is given such that $\sigma(z/T)$ is highly correlated to $p(y = 1|x)$, the proposed method attains reasonable variance reduction in exchange for additional bias. Conversely, if z has no additional information to y, that is, for example, z is determined by y as $z = 2y - 1$, the TV distance terms immediately increase when $T > 0$, and we cannot attain significant variance reduction. Note that the soft-label itself need not necessarily be a good estimator of $p(y = 1|x)$, which is a difference from the probabilistic label $p_n = p(y = 1|x_n)$.

4.4. Connection to the learning using privileged information (LUPI)

Learning using privileged information (LUPI) is a general problem setting that aims to utilize additional information like z. Privileged information was first proposed in Vapnik and Vashist (2009), in which it was assumed that additional features were provided for each training instance and that the features were strongly related to the label but not available in the test phase. They argued that a faster learning rate could be obtained by using privileged information to estimate the slack variables in the SVM. Generalized distillation (GD) (Lopez-Paz et al., 2016) enables model-agnostic learning with privileged information using a similar procedure to the distillation (Hinton et al., 2015). The basic procedure of GD is to first learn a "teacher model" $g_t(z)$ from the privileged features $z \in \mathbb{R}^m$ and the original labels, and then learn a "student model" with the original features $x \in \mathbb{R}^d$ and soft-labels given by the teacher model using the following proxy loss[1]:

$$\hat{L}_T(g) = \frac{1}{N}\sum_n^N \left\{\sigma(g_t(z_n)/T)\ell(g(x_n)) + \sigma(-g_t(z_n)/T)\ell(-g(x_n))\right\}.$$

While those methods are aimed at fast learning rates in terms of the sample size, we utilize soft-labels given by a similar procedure for lessening the imbalance. To the best of our knowledge, this is the first work that utilizes privileged information for imbalanced classification problems. Without cost rebalancing in (9), GD cannot attain the variance reduction analyzed in Section 4.2. The key advantage of privileged information in the application to the class-imbalanced problems comes from the reduction of the instance-wise Lipschitz constants by rebalanced costs, which highlights a new aspect of LUPI.

5. Experiments

In this section, on the basis of extensive experiments on synthetic and real datasets, we demonstrate the characteristics and the performance of the proposed method.

5.1. Experimental setup

Here we describe experimental settings briefly, and the details, including computing infrastructure, preprocess, and the settings of compared methods, are given in Appendix C.

[1] In the original paper, they used a mixed label of the true label y and the teacher label $\sigma(g_t/T)$, by means of a so-called imitation parameter λ. We do not need λ since positivity z includes the whole information of y.

Table 1
Dataset specifications. UCI from Dheeru and Karra Taniskidou (2017), Regression from Torgo (2018), and Keel from Alcalá-Fdez et al. (2011). For the song dataset (SO), the number of instances was reduced by random subsampling.

Abbreviation	Dataset	Source	Instances	Dims
AB	Abalone	UCI	4177	9
AQ	Air quality	UCI	6941	11
BH	Boston housing	UCI	506	13
CA	California	Keel	20 640	8
DI	Diabetes	UCI	442	10
HO	House	Keel	22 784	16
KI	Kinematics	Regression	8192	8
PU	Puma32H	Regression	8192	32
ST	Student performance	UCI	1044	43
SO	Song year prediction MSD	UCI	10 000	90
TO	Toy	Synthesized	3000	100
WI	Wine quality	UCI	6497	12
–	GPU kernel performance	UCI	241 600	14

5.1.1. Datasets

All the datasets we used in our experiments are summarized in Table 1. Since our method (as do regression and rank-based baselines) requires positivity z, we used datasets originally designed for use in regression problems. Each dataset has a numerical target attribute, which we regarded as positivity z, and we set the task-specific threshold θ such that the top-$100p_+$% would be positive.

5.1.2. Evaluation

We used balanced accuracy (BA) for the performance evaluation, as explained in Section 2. For the regression-based methods, we applied the original threshold to the prediction to evaluate BA, i.e., $\hat{y} = I(\hat{z} \geq 0)$. For the rank-based method, we set θ such that the top $100p_+$% predicted scores would be positive.

In the experiments in Sections 5.3 and 5.4, we evaluated BA using nested cross-validation (Varma & Simon, 2006). The outer cross-validation loop was 5-fold, and the inner one for hyperparameter selection was 2-fold. An outer loop split each dataset into 80% for training and 20% for testing, and an inner loop then divided the training set into half for validation in order to select the hyperparameter. We then performed re-training under the selected hyperparameter with both training and validation sets. For the Gaussian process (GP), we applied the maximum likelihood estimation for hyperparameter selection to avoid heavy computation. In both training and test data, the ratio of positive and negative sample was maintained, i.e., stratified sampling was performed. We repeated this process four times, changing the split of the outer loop (thus, there were 20 results for the test data).

5.2. Performance variation in temperature T

First, we investigated the effect of introducing a soft-label using the hyperparameter T. Since a soft-label with a small T goes to a hard label y, the change in metrics for various T values demonstrates the benefit of utilizing positivity information. We used the toy data in Fig. 2(a) and logistic regression with l2 and l1 regularizers. The regularization strength was fixed to 1.0.

Results with respect to BA are shown in Fig. 3. The best T is neither zero nor infinity, which indicates the variance reduction in small T and the bias increase in large T. The difference between the best performance and the performance in $T \to 0$ illustrates the benefit of introducing the soft-label. Also, $T \to \infty$ means treating near-miss and far-miss, i.e., the other negative instances equally, which induces a large bias as analyzed in Section 4.3 and degrades the performance. This illustrates the importance of treating only near-miss instances as being partly positive.

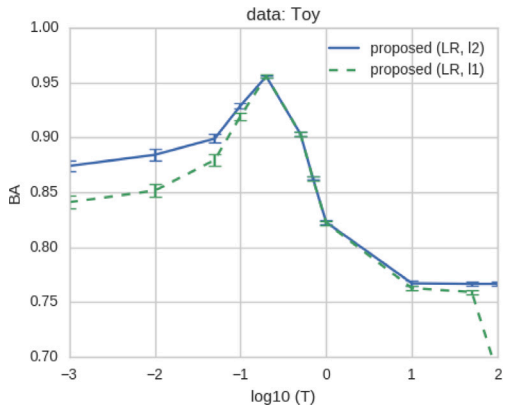


Fig. 3. Performance in various T on toy data in Fig. 2(a). As shown, there exist here some moderate temperatures that perform better than $T \to 0$ or $T \to \infty$.

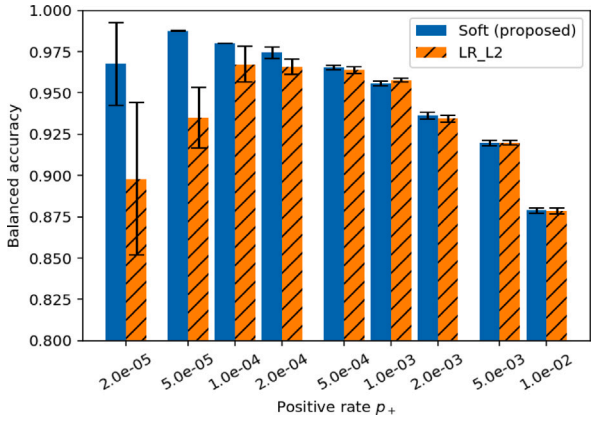


Fig. 4. Performance of the proposed method and conventional cost-sensitive classification with respect to BA for the GPU kernel performance dataset under highly imbalanced conditions. Positive rate $p_+ := \sum I(z \geq 0)/N$ ranged from 2×10^{-5} to 1×10^{-2}. Error bars indicate standard error.

5.3. Comparison with conventional classification under highly imbalanced conditions

To demonstrate the benefit of our method under highly imbalanced conditions, we compared it with conventional cost-sensitive learning for various positive rates p_+ (and thus N_+). The base learner was a logistic regression model with the L2 regularizer.

We used the GPU kernel performance dataset (Ballester-Ripoll et al., 2017; Nugteren & Codreanu, 2015), which is a large-scale dataset with real-valued target attributes. It had 14 features of GPU kernel parameters and four target attributes of elapsed times in milliseconds for four independent runs under the same parameters, and the number of instances was 241.6k. We transformed the problem for elapsed time regression into a classification for finding good parameters, i.e., we used the average speed $z = \frac{4}{\sum y_i}$, where $\{y_i\}_{1:4}$ are the original elapsed times.

The results given in Fig. 4 show that the conventional cost-sensitive logistic regression worsened when highly imbalanced, while the proposed method worked well. The performance gap is particularly large when $p_+ \leq 5{\times}10^{-5}$, which means the size of the positive training sample $N_+ \leq 10 < d$. This is in good agreement with the theoretical prediction in (12). The results with respect to AUC in the same setting and the results in fixed p_+ and various N are also shown in Appendix D, which presents similar trends.

Table 2
Averaged performance comparison over all datasets. In addition to our main metric Balanced Accuracy (2), we also evaluated the geometric mean of precision and recall (G-mean), the area under the ROC curve (ROC-AUC), and the area under the precision–recall curve (Average Precision). Model selection was performed on each metric.

Metric method	Balanced accuracy	G-mean	ROC-AUC	Average precision
Cost-sensitive classification				
LR (L1)	.842 ± .007*	.845 ± .007	.902 ± .006**	.479 ± .019**
LR (L2)	.838 ± .007**	.838 ± .007*	.899 ± .006**	.466 ± .019**
SVM	**.853 ± .007**	.843 ± .008	*.907 ± .007***	.527 ± .019**
Regression-based				
Lasso	.594 ± .010**	.289 ± .021**	.898 ± .006**	.478 ± .020**
Ridge	.766 ± .009**	.722 ± .012**	.904 ± .006**	.470 ± .019**
GP	.678 ± .011**	.533 ± .022**	.893 ± .009	**.568 ± .020**
Rank-based				
Rank-SVM	.699 ± .009**	.334 ± .022**	.882 ± .008**	.428 ± .019**
SMOTE-variants				
SMOTE-LR (L1)	.844 ± .007	.843 ± .007	.901 ± .006**	.471 ± .019**
SMOTE-LR (L2)	.840 ± .007**	.837 ± .007*	.900 ± .006**	.458 ± .018**
SMOTE-SVM	.815 ± .008**	.800 ± .010**	.906 ± .007**	.525 ± .019**
B-SMOTE-LR (L1)	.828 ± .008**	.816 ± .009**	.893 ± .007**	.469 ± .018**
B-SMOTE-LR (L2)	.814 ± .007**	.802 ± .009**	.891 ± .007**	.452 ± .018**
S-SMOTE-LR (L1)	.711 ± .009**	.600 ± .019**	.868 ± .008**	.400 ± .016**
S-SMOTE-LR (L2)	.709 ± .009**	.600 ± .019**	.867 ± .007**	.392 ± .016**
CCR-SMOTE-LR (L1)	.844 ± .007*	.842 ± .007*	.900 ± .006**	.462 ± .018**
CCR-SMOTE-LR (L2)	.837 ± .007**	.836 ± .007**	.900 ± .006**	.463 ± .018**
Ensemble				
RUSBoost	.788 ± .009**	.769 ± .012**	.860 ± .008**	.404 ± .017**
Proposed				
Soft-LR (L1)	.851 ± .006	**.848 ± .007**	.907 ± .006**	.489 ± .020**
Soft-LR (L2)	.849 ± .006	*.847 ± .006*	.905 ± .006**	.484 ± .020**
Soft-SVM	*.851 ± .006*	.846 ± .007	**.921 ± .006**	*.535 ± .019***

*$p < 0.05$ w.r.t. the best method.
**$p < 0.005$ w.r.t. the best method.

5.4. Comparison with various baseline methods and datasets

We are also able to demonstrate the versatility of our proposed method for various datasets. Positive rate p_+ was fixed to 5% since the sample sizes are not so large in most of the datasets we prepared. We compared the proposed method with three base learners (logistic regression with L1 and L2 regularizers, and SVM with an RBF kernel) and baseline methods, namely, the conventional cost-sensitive classification, oversampling-based classification (SMOTE) (Chawla et al., 2002), borderline-SMOTE (B-SMOTE) (Han et al., 2005), safe-level-SMOTE (S-SMOTE) (Bunkhumpornpat et al., 2009), CCR (CCR-SMOTE) (Koziarski & Woźniak, 2017), undersampling ensemble classification (RUSBoost) (Seiffert et al., 2009), regression-based methods (ridge, lasso, and GP with an RBF kernel), and Rank-SVM (with a linear kernel, as proposed in Xue and Hauskrecht (2016)). Following the recommendation in Stapor et al. (2021), the Wilcoxon signed-rank test is used to measure the significance of the best method against other methods in each setting.

Table 2 summarizes the results in various metrics over all datasets. The model selection was performed for each metric. Overall, the proposed method outperformed or was comparable to other methods in most of the metrics. Compared to the cost-sensitive learning and other hard classification methods, our methods have resulted in better performance in the most metrics and base learners. As shown in Section 5.3, the difference would be more significant with higher imbalance ratio, which was fixed to 1:19 here. The regression-based methods performed relatively well in ROC-AUC and average precision, which are metrics for various thresholds, while hard classification methods (cost-sensitive, SMOTE-variants, and ensemble) performed well in balanced accuracy and G-mean. This may reflect the difference between classification methods, which optimize performance for a fixed threshold with respect to the positivity, and regression-based methods, which care various positivity levels. The proposed methods were better or comparable to both of them, which illustrates the versatility of our method. Although SMOTE and its variants are commonly used in imbalanced classification and are still being actively studied, they have rarely shown better performance than that of the cost-sensitive learning. As discussed in Section 3.4, this is reasonable because the inductive biases that were valid in the conventional classification setting, e.g., the cluster assumption (Chapelle et al., 2006), are not valid in our typical setting of prediction where the feature x causes the outcome y.

In average precision (AP), regression-based methods performed well, with GP in particular outperforming the others significantly. AP is especially important to ensure that the instances we flag are mainly positives (Cook & Ramadas, 2020), as in recommendation problems. On the other hand, ROC-AUC is more important for identifying a high percentage of the positives, as in safety-related applications. It should also be noted that AP is more volatile than ROC-AUC (Cook & Ramadas, 2020), as it is greatly influenced by the performance of a small number of instances with high prediction scores. In fact, the difference between GP and other methods comes mainly from a single dataset (WI) as shown in Table 7.

Detailed metrics (precision, recall, and specificity) can also be found in Table 3 for the model selected with the balanced accuracy. The results show that the regression-based methods prefer precision, which means the conservativeness to flag as positive. The reason may be because the predicted values by a regressor tend not to output extreme values, and the proportion of prediction exceeding the original threshold ($\hat{z} \geq \theta$) would be smaller than that of the actual positivity. A possible workaround for mitigating this tendency would be using another threshold θ', which would be another tuning parameter. As with the other methods, we used a fixed threshold to ensure a fair comparison.

Tables 4–10 show the results for each dataset. Our approach outperformed or was at least comparable to the regression and the rank-based baselines for properly chosen base learners, while regression-based approaches failed for some data, including Diabetes (DI) and Puma32H (PU) in BA and G-mean. The student performance dataset (ST) had a quite limited number of instances for its dimensions, which may be a reason why the regression-based baseline worked better.

To investigate the performance on high-dimensional and large-scale data, we also employed the GPU dataset used in Section 5.3 with an

Table 3
Averaged performance on accuracy-based metrics in detail. Model selection was performed on balanced accuracy.

Metric method	Balanced accuracy	Precision	Recall	Specificity
Cost-sensitive classification				
LR (L1)	.842 ± .007*	.248 ± .009**	.861 ± .009**	.833 ± .006**
LR (L2)	.838 ± .007**	.244 ± .009**	.843 ± .009**	.838 ± .005**
SVM	**.853 ± .007**	.260 ± .010**	.844 ± .011**	.852 ± .005**
Regression-based				
Lasso	.594 ± .010**	.366 ± .026**	.193 ± .020**	**.995 ± .000**
Ridge	.766 ± .009**	*.417 ± .018***	.600 ± .017**	.930 ± .005**
GP	.678 ± .011**	**.590 ± .022**	.399 ± .022**	*.992 ± .000***
Rank-based				
Rank-SVM	.699 ± .009**	.412 ± .016**	.459 ± .018**	.969 ± .001**
SMOTE-variants				
SMOTE-LR (L1)	.844 ± .007	.265 ± .010**	.850 ± .009**	.842 ± .006**
SMOTE-LR (L2)	.840 ± .007**	.259 ± .010**	.833 ± .009**	.847 ± .005**
SMOTE-SVM	.815 ± .008**	.272 ± .009**	.793 ± .014**	.860 ± .005**
B-SMOTE-LR (L1)	.828 ± .008**	.284 ± .012**	.791 ± .013**	.866 ± .006**
B-SMOTE-LR (L2)	.814 ± .007**	.298 ± .012**	.744 ± .014**	.884 ± .004**
S-SMOTE-LR (L1)	.711 ± .009**	.326 ± .016**	.470 ± .019**	.951 ± .003**
S-SMOTE-LR (L2)	.709 ± .009**	.336 ± .016**	.470 ± .018**	.950 ± .003**
CCR-SMOTE-LR (L1)	.844 ± .007*	.240 ± .009**	.866 ± .009**	.823 ± .006**
CCR-SMOTE-LR (L2)	.837 ± .007**	.229 ± .008**	.852 ± .009**	.822 ± .006**
Ensemble				
RUSBoost	.788 ± .009**	.283 ± .014**	.711 ± .015**	.865 ± .006**
Proposed				
Soft-LR (L1)	.851 ± .006	.238 ± .009**	*.901 ± .007***	.803 ± .008**
Soft-LR (L2)	.849 ± .006	.232 ± .008**	.900 ± .006**	.800 ± .008**
Soft-SVM	*.851 ± .006*	.228 ± .009**	**.923 ± .007**	.784 ± .009**

*$p < 0.05$ w.r.t. the best method.
**$p < 0.005$ w.r.t. the best method.

Table 4
Results on balanced accuracy.

Dataset method	AB	AQ	BH	CA	DI	HO	KI	PU	SO	ST	TO	WI
Cost-sensitive classification												
LR (L1)	.835	**.963**	.895*	.881**	.745*	.822**	.859**	.868**	.613**	.959*	.944**	.724**
LR (L2)	.836	.959*	.909*	.881**	.736**	.822**	.861**	.866**	.612**	.895**	.950**	.732**
SVM	.827	.946**	**.942**	.891*	.728*	*.893*	*.947*	.875**	.598**	.929**	.932**	.729**
Regression-based												
Lasso	.566**	.877**	.519**	.590**	.501**	.507**	.500**	.500**	.515**	**.972**	.585**	.500**
Ridge	.769**	.932**	.845**	.625**	.667**	.777**	.735**	.681**	.552**	.952	.934**	.728**
GP	.565**	.912**	.677**	.637**	.500**	.740**	.873**	.507**	.511**	.953	.579**	.680**
Rank-based												
Rank-SVM	.671**	.844**	.834**	.702**	.605**	.576**	.714**	.632**	.504**	.878**	.863**	.563**
SMOTE-variants												
SMOTE-LR (L1)	.837	.958**	.892**	.883**	.755	.821**	.859**	.876**	.622*	*.964***	.938**	.724**
SMOTE-LR (L2)	.837	.958*	.900**	.882**	.743	.821**	.863**	.873**	.621**	.914**	.940**	.725**
SMOTE-SVM	.820*	.950**	.885**	*.893*	.727**	.893	.935**	.862**	.583**	.878**	.621**	*.737***
B-SMOTE-LR (L1)	.811**	.951**	.897*	.882**	.708*	.822**	.860**	*.887***	.561**	.953*	.881**	.728**
B-SMOTE-LR (L2)	.815**	.959*	.911	.882**	.707*	.823**	.858**	.864**	.548**	.806**	.872**	.723**
S-SMOTE-LR (L1)	.748**	.881**	.718**	.842**	.522**	.682**	.817**	.822**	.507**	.772**	.712**	.505**
S-SMOTE-LR (L2)	.746**	.888**	.737**	.843**	.542**	.682**	.818**	.781**	.512**	.748**	.713**	.505**
CCR-SMOTE-LR (L1)	**.839**	.958*	.902*	.882**	.735	.822**	.861**	.866**	.643	.959	.945**	.718**
CCR-SMOTE-LR (L2)	*.839*	*.961*	.895*	.882**	.738	.820**	.861**	.867**	.625*	.897**	.944**	.715**
Ensemble												
RUSBoost	.724**	.870**	.892*	.805**	.608**	.863**	.858**	**.921**	.556**	.952	.705**	.702**
Proposed												
Soft-LR (L1)	.833	.958**	*.917**	.881**	**.778**	.822**	.863**	.868**	*.652**	.950*	**.963**	.724**
Soft-LR (L2)	.835	.959*	.914*	.881**	*.757**	.822**	.864**	.867**	**.654**	.941*	*.962**	.736**
Soft-SVM	.833	.956**	.886*	**.895**	.734*	**.894**	**.951**	.878**	.642	.881**	.899**	**.765**

*$p < 0.05$ w.r.t. the best method.
**$p < 0.005$ w.r.t. the best method.

expanded binary feature set of up to second-order interaction terms of the original features. The resulting number of features was 335. Due to the large data size ($N = 241{,}600$), we compared methods excluding the kernel-based and pairwise ranking-based methods. The performance comparison under various positive rate is shown in Table 11. The resulting performance illustrates that our proposed method outperforms baseline methods in a highly imbalanced setting ($p_+ = 0.005\%$) and is comparable in a mildly imbalanced setting ($p_+ = 0.1\%$, which means

Table 5
Results on G-mean.

Dataset method	AB	AQ	BH	CA	DI	HO	KI	PU	SO	ST	TO	WI
Cost-sensitive classification												
LR (L1)	.839	.962	.933	.881**	.730	.821**	.857**	.864**	.638*	.959*	.937**	.721*
LR (L2)	.838	**.962**	.903	.881**	.737	.821**	.858**	.864**	.618**	.906**	.946**	.725*
SVM	.821**	.945**	.902	.891*	.708	.892	.943**	.869**	.576**	.912**	.924**	.733
Regression-based												
Lasso	.362**	.870**	.110**	.425**	.000**	.128**	.000**	.000**	.174**	**.975**	.399**	.032**
Ridge	.757**	.932**	.824**	.501**	.565**	.773**	.702**	.618**	.373**	.954	.938**	.730
GP	.363**	.899**	.843**	.522**	.047**	.698**	.871**	.108**	.174**	.961	.285**	.621**
Rank-based												
Rank-SVM	.187**	.003**	.293**	.015**	.716	.329**	.725**	.782**	.085**	.137**	.731**	.000**
SMOTE-variants												
SMOTE-LR (L1)	.834	.958**	.903	.883**	.748	.819**	.857**	.872**	.627*	.963**	.934**	.723*
SMOTE-LR (L2)	.836	.960*	.900*	.882**	.745	.821**	.858**	.870**	.617**	.897**	.938**	.723*
SMOTE-SVM	.821**	.950**	.888**	.893	.706	.893	.930**	.863**	.554**	.870**	.497**	.733
B-SMOTE-LR (L1)	.805**	.950**	.898*	.881**	.671*	.822**	.860**	.883**	.478**	.952**	.871**	.725*
B-SMOTE-LR (L2)	.809**	.959*	.909	.882**	.688	.822**	.858**	.863**	.439**	.797**	.866**	.726
S-SMOTE-LR (L1)	.732**	.874**	.642**	.834**	.195**	.628**	.809**	.800**	.232**	.706**	.676**	.065**
S-SMOTE-LR (L2)	.732**	.883**	.653**	.834**	.211**	.628**	.809**	.776**	.234**	.699**	.678**	.062**
CCR-SMOTE-LR (L1)	**.839**	.958**	.909	.882**	.717	.821**	.859**	.863**	.641*	.959**	.945**	.715**
CCR-SMOTE-LR (L2)	.838	.962	.895*	.882**	.726	.820**	.859**	.863**	.621**	.906**	.943**	.716**
Ensemble												
RUSBoost	.708**	.862**	.872*	.786**	.473**	.863**	.857**	**.921**	.559**	.949**	.692**	.687**
Proposed												
Soft-LR (L1)	.837	.957**	**.934**	.881**	**.748**	.821**	.861**	.864**	.657	.943**	**.961**	.716**
Soft-LR (L2)	.837	.960*	.907	.881**	.745	.822**	.863**	.864**	**.659**	.949*	.961	.722**
Soft-SVM	.830*	.953**	.888**	**.895**	.711*	**.895**	**.956**	.877**	.652	.854**	.903**	**.742**

*$p < 0.05$ w.r.t. the best method.
**$p < 0.005$ w.r.t. the best method.

Table 6
Results on ROC-AUC.

Dataset method	AB	AQ	BH	CA	DI	HO	KI	PU	SO	ST	TO	WI
Cost-sensitive classification												
LR (L1)	.916	.991*	.981**	.948**	.826*	.891**	.932**	.908**	.654**	**.995**	.988**	.799**
LR (L2)	.916	.991*	.950**	.948**	.833*	.891**	.933**	.900**	.652**	.982**	.988**	.801**
SVM	.912	.991	.976**	**.957**	.805**	.956**	.986**	.910**	.635**	.981**	.975**	.802**
Regression-based												
Lasso	.894**	.991*	.946**	.927**	**.866**	.875**	.919**	.908**	.673**	.994*	.993**	.790**
Ridge	.906*	.992	.977**	.943**	.860	.868**	.930**	.903**	.692**	.994*	.983**	.805**
GP	.915	**.992**	.928*	.956	.572**	**.960**	**.993**	.907**	.695**	.992**	.989**	.815**
Rank-based												
Rank-SVM	.915	.986**	.976**	.869**	.833*	.788**	.932**	.898**	.605**	.993**	.987**	.801**
SMOTE-variants												
SMOTE-LR (L1)	.916	.991	.952**	.948**	.816**	.891**	.932**	.907**	.684*	.994	.986**	.799**
SMOTE-LR (L2)	.916	.991	.949**	.948**	.824**	.891**	.933**	.902**	.675*	.981**	.987**	.801**
SMOTE-SVM	.910	.991	.959**	.956**	.810*	.957**	.986**	.900**	.636**	.980**	.975**	.806**
B-SMOTE-LR (L1)	.913**	.991	.964**	.947**	.781**	.889**	.930**	.907**	.617**	.994	.986**	.792**
B-SMOTE-LR (L2)	.913**	.991	.944**	.947**	.807*	.889**	.931**	.904**	.611**	.977**	.987**	.794**
S-SMOTE-LR (L1)	.873**	.987**	.929**	.946**	.797**	.876**	.930**	.898**	.558**	.965**	.893**	.763**
S-SMOTE-LR (L2)	.875**	.989**	.921**	.946**	.824*	.876**	.930**	.873**	.553**	.956**	.893**	.769**
CCR-SMOTE-LR (L1)	.917*	.991	.925**	.948**	.815**	.891**	.932**	.907**	.696*	.991*	.988**	.799**
CCR-SMOTE-LR (L2)	**.918**	.991	.938**	.948**	.829*	.891**	.932**	.901**	.682**	.981**	.988**	.800**
Ensemble												
RUSBoost	.809**	.956**	.966*	.897**	.714**	.936**	.929**	**.971**	.582**	.985**	.782**	.797**
Proposed												
Soft-LR (L1)	.916	.992	.959**	.948**	.853	.891**	.932**	.908**	.702*	.995	.994	.796**
Soft-LR (L2)	.916	.991	.940**	.948**	.855	.891**	.933**	.902**	.702*	.993**	**.994**	.800**
Soft-SVM	.906*	.989**	**.991**	.957	.819*	.958**	.990**	.920**	**.716**	.972**	.985**	**.850**

*$p < 0.05$ w.r.t. the best method.
**$p < 0.005$ w.r.t. the best method.

the positive sample size of $N_+ = 242$, equivalent to the dimension in terms of order). When the positive rate is 0.1%, the positive sample size would be $N_+ = 241$, which is about the same as the dimensions. Thus, it is thought that the variance was not dominant when the linear model with a regularizer was used, and the difference from the cost-sensitive learning did not appear. It is again confirmed that the proposed method performs as well as the cost-sensitive learning when the estimated variance is not dominant and improves on the cost-sensitive learning when the sample size of positive examples is small, and the estimated variance becomes dominant.

Table 7
Results on average precision.

Dataset method	AB	AQ	BH	CA	DI	HO	KI	PU	SO	ST	TO	WI
Cost-sensitive classification												
LR (L1)	.352	.859**	.773**	.647**	.262	.247**	.426**	.320**	.039**	**.894**	.810**	.116**
LR (L2)	.352	.857**	.784**	.650**	.237*	.247**	.416**	.311**	.040**	.762**	.818**	.115**
SVM	.342	.859**	.878*	.702*	.193**	.574**	.826**	.370**	**.066**	.703**	.682**	*.129***
Regression-based												
Lasso	.328*	.864**	.762**	.630**	**.310**	.240**	.339**	.328**	.042**	.874	.901**	.116**
Ridge	.328	*.866***	.768**	.646**	.280	.236**	.372**	.310**	.041**	*.887*	.795**	.115**
GP	.335	**.903**	.773**	**.709**	.064**	**.626**	**.877**	.377**	.043**	.839**	.838**	**.426**
Rank-based												
Rank-SVM	.352	.747**	.710**	.410**	.252*	.162**	.414**	.298**	.031**	.832**	.807**	.116**
SMOTE-variants												
SMOTE-LR (L1)	.356	.863**	.714**	.644**	.266	.247**	.415**	.325**	.044**	.864	.803**	.111**
SMOTE-LR (L2)	.355*	.856**	.734**	.651**	.257*	.249**	.406**	.311**	.043**	.719**	.811**	.112**
SMOTE-SVM	.343	.856**	*.881*	*.707*	.237	*.593***	.834**	.333**	.039**	.665**	.693**	.121**
B-SMOTE-LR (L1)	**.361**	.857**	.687**	.638**	*.296*	.240**	.420**	.321**	.039**	.847	.798**	.121**
B-SMOTE-LR (L2)	*.360*	.858**	.735**	.637**	.270	.241**	.411**	.321**	.036**	.613**	.807**	.121**
S-SMOTE-LR (L1)	.322**	.825**	.681**	.658**	.237*	.250**	.428**	.304**	.026**	.589**	.369**	.112**
S-SMOTE-LR (L2)	.324**	.848**	.717**	.658**	.230*	.250**	.422**	.252**	.029**	.498**	.367**	.112**
CCR-SMOTE-LR (L1)	.353*	.849**	.691**	.650**	.247*	.246**	.439**	.327**	.043**	.788*	.796**	.117**
CCR-SMOTE-LR (L2)	.354*	.849**	.750**	.650**	.264	.246**	.436**	.309**	.044**	.726**	.807**	.114**
Ensemble												
RUSBoost	.216**	.597**	.729**	.514**	.190*	.459**	.411**	**.725**	.031**	.672**	.211**	.099**
Proposed												
Soft-LR (L1)	.348	.866**	.775**	.659**	.262*	.246**	.419**	.320**	*.051**	.879	*.917*	.120**
Soft-LR (L2)	.350	.866**	.757**	.660**	.273	.246**	.412**	.294**	.049*	.863*	**.918**	.120**
Soft-SVM	.308*	.823**	**.915**	.657**	.258	.588**	*.843***	*.431***	.049**	.630**	.799**	.119**

*$p < 0.05$ w.r.t. the best method.
**$p < 0.005$ w.r.t. the best method.

Table 8
Results on precision.

Dataset method	AB	AQ	BH	CA	DI	HO	KI	PU	SO	ST	TO	WI
Cost-sensitive classification												
LR (L1)	.217**	.458**	.394**	.269**	.152*	.183**	.222**	.190**	.034**	.354**	.433**	.070**
LR (L2)	.217**	.470**	.369**	.269**	.158*	.185**	.220**	.191**	.035**	.335**	.410**	.074**
SVM	.196**	.342**	.509**	.313**	.131**	.289**	.436**	.221**	.034**	.215**	.349**	.080**
Regression-based												
Lasso	**.497**	*.827*	.250**	**.929**	.000**	.206**	.000**	.000**	*.042**	.644*	**1.000**	.000**
Ridge	.288**	.693**	*.761***	.864**	**.223**	.195**	.393**	.310**	.039**	**.681**	.481**	.079**
GP	*.431**	**.828**	**.890**	*.924*	.033**	**.657**	**.818**	**.596**	**.087**	*.669*	.514**	**.636**
Rank-based												
Rank-SVM	.360**	.761**	.698**	.433**	*.210*	.194**	.451**	.313**	.034**	.600**	*.742***	*.152***
SMOTE-variants												
SMOTE-LR (L1)	.223**	.460**	.403**	.278**	.158*	.186**	.225**	.201**	.034**	.434**	.509**	.073**
SMOTE-LR (L2)	.224**	.500**	.391**	.277**	.163*	.188**	.228**	.205**	.035**	.366**	.451**	.075**
SMOTE-SVM	.205**	.358**	.355**	.309**	.126**	*.305***	*.461***	.218**	.035**	.214**	.528**	.074**
B-SMOTE-LR (L1)	.246**	.397**	.490**	.305**	.147**	.189**	.217**	.229**	.034**	.456**	.614**	.082**
B-SMOTE-LR (L2)	.249**	.468**	.458**	.304**	.182*	.191**	.222**	.233**	.036**	.520**	.626**	.082**
S-SMOTE-LR (L1)	.248**	.703**	.611**	.506**	.092*	.291**	.338**	.244**	.028**	.450**	.270**	.135**
S-SMOTE-LR (L2)	.248**	.711**	.735**	.506**	.159	.291**	.338**	.220**	.031**	.385**	.275**	.135**
CCR-SMOTE-LR (L1)	.214**	.435**	.334**	.269**	.131**	.181**	.214**	.186**	.035**	.402**	.415**	.066**
CCR-SMOTE-LR (L2)	.213**	.449**	.330**	.270**	.137**	.182**	.212**	.190**	.035**	.274**	.392**	.068**
Ensemble												
RUSBoost	.159**	.506**	.670**	.243**	.115**	.256**	.250**	*.409**	.027**	.533**	.150**	.082**
Proposed												
Soft-LR (L1)	.191**	.409**	.393**	.268**	.134**	.181**	.196**	.188**	.033**	.338**	.454**	.065**
Soft-LR (L2)	.186**	.433**	.337**	.268**	.141**	.183**	.203**	.188**	.033**	.303**	.447**	.064**
Soft-SVM	.158**	.379**	.382**	.293**	.127**	.273**	.429**	.205**	.032**	.135**	.254**	.065**

*$p < 0.05$ w.r.t. the best method.
**$p < 0.005$ w.r.t. the best method.

6. Summary

In this paper, we have introduced a novel problem setting, imbalanced classification with positivity, and proposed a versatile method for dealing with it, which highlighted the usefulness of the positivity information. The key advantage of our method is exploiting near-miss positive instances, which are specified by positivity, to lessen the class imbalance.

We have investigated the loss theoretically for the proposed method and for conventional cost-sensitive learning in consideration of the degree of imbalance, and have shown that our method lessens the imbalance with the help of near-miss positive instances. Extensive experiments have illustrated that our method outperforms the conventional cost-sensitive classification under highly imbalanced conditions and is more versatile than are existing regression or rank-based approaches.

Table 9
Results on recall.

Dataset method	AB	AQ	BH	CA	DI	HO	KI	PU	SO	ST	TO	WI
Cost-sensitive classification												
LR (L1)	.824**	.984**	*.943*	.890**	.707*	.841**	.879**	.948	.624**	*.986*	.940**	.763**
LR (L2)	.824**	.983**	.887**	.890**	.718*	.838**	.882**	.947	.554**	.893**	.963**	.739**
SVM	.806**	.994	.865**	.885**	.720*	*.904***	*.952***	.911**	.443**	.964	.943**	.739**
Regression-based												
Lasso	.143**	.763**	.048**	.181**	.000**	.017**	.000**	.000**	.033**	.971	.162**	.000**
Ridge	.623**	.886**	.702**	.252**	.373**	.707**	.515**	.404**	.156**	.929*	.930**	.725**
GP	.141**	.816**	.723**	.274**	.022**	.494**	.766**	.015**	.043**	.943*	.153**	.391**
Rank-based												
Rank-SVM	.389**	.777**	.698**	.434**	.228**	.194**	.451**	.313**	.086**	.943*	.742**	.250**
SMOTE-variants												
SMOTE-LR (L1)	.812**	.977**	.887*	.887**	.750*	.834**	.873**	.949	.595**	.971	.915**	.748**
SMOTE-LR (L2)	.812**	.971**	.878**	.886**	.718*	.831**	.875**	.937**	.545**	.864**	.940**	.734**
SMOTE-SVM	.799**	*.994*	.907*	.890**	.685*	.892**	.920**	.898**	.383**	.921*	.257**	.771**
B-SMOTE-LR (L1)	.734**	.980**	.850**	.868**	.670*	.831**	.888**	.940*	.280**	.950*	.792**	.707**
B-SMOTE-LR (L2)	.740**	.977**	.887*	.869**	.560**	.830**	.880**	.882**	.205**	.636**	.770**	.691**
S-SMOTE-LR (L1)	.584**	.778**	.450**	.722**	.082**	.418**	.707**	.771**	.041**	.571**	.507**	.013**
S-SMOTE-LR (L2)	.579**	.794**	.487**	.723**	.123**	.418**	.709**	.691**	.057**	.543**	.507**	.013**
CCR-SMOTE-LR (L1)	.827**	.983**	.917	*.891***	.742	.844**	.895**	*.951*	.622**	.971	.963**	.783**
CCR-SMOTE-LR (L2)	.827**	.987**	.898*	.891**	.740	.839**	.898**	.948	.562**	.907*	.968**	.755**
Ensemble												
RUSBoost	.639**	.794**	.810**	.768**	.360**	.857**	.850**	.913**	.398**	.936*	.587**	.624**
Proposed												
Soft-LR (L1)	.847**	.988*	**.972**	.890**	*.812*	.844**	.932**	.950	*.785*	.957*	.987*	.843**
Soft-LR (L2)	*.853***	.987*	.895**	.890**	**.823**	.841**	.926**	.951	**.795**	.986	*.988*	*.863***
Soft-SVM	**.913**	**.995**	.915	**.906**	.765	**.918**	**.983**	**.951**	.767	**.993**	**.997**	**.977**

*$p < 0.05$ w.r.t. the best method.
**$p < 0.005$ w.r.t. the best method.

Table 10
Results on specificity.

Dataset method	AB	AQ	BH	CA	DI	HO	KI	PU	SO	ST	TO	WI
Cost-sensitive classification												
LR (L1)	.855**	.938**	.912**	.872**	.785**	.802**	.837**	.787**	.652**	.935**	.935**	.683**
LR (L2)	.854**	.942**	.908**	.873**	.791**	.806**	.835**	.788**	.698**	.929**	.924**	.711**
SVM	.838**	.899**	.942**	.897**	.736**	.882**	.934**	.828**	.753**	.877**	.906**	.733**
Regression-based												
Lasso	**.993**	**.992**	**1.000**	**.999**	**.998**	**.997**	**1.000**	**1.000**	*.986***	.980*	**1.000**	**.999**
Ridge	.924**	.979**	.988**	.998**	.920**	.847**	.958**	.953**	.925**	**.984**	.946**	.736**
GP	*.991***	*.991*	*.995***	*.999***	*.989***	*.986***	*.991***	*1.000***	**.991**	*.982*	*.999***	.991**
Rank-based												
Rank-SVM	.966**	.987**	.984**	.970**	.953**	.958**	.971**	.964**	.951**	.978**	.986**	.956**
SMOTE-variants												
SMOTE-LR (L1)	.861**	.939**	.926**	.878**	.776**	.808**	.842**	.800**	.664**	.955**	.953**	.702**
SMOTE-LR (L2)	.862**	.949**	.920**	.878**	.795**	.812**	.844**	.808**	.699**	.939**	.938**	.716**
SMOTE-SVM	.848**	.907**	.889**	.895**	.743**	.893**	.942**	.830**	.790**	.834**	.988**	.698**
B-SMOTE-LR (L1)	.889**	.922**	.946**	.896**	.745**	.812**	.831**	.833**	.843**	.956**	.970**	.748**
B-SMOTE-LR (L2)	.891**	.941**	.936**	.895**	.851**	.815**	.837**	.846**	.891**	.976*	.975**	.756**
S-SMOTE-LR (L1)	.912**	.983**	.989**	.963**	.957**	.946**	.927**	.873**	.973**	.973**	.918**	.997**
S-SMOTE-LR (L2)	.913**	.983**	.990**	.963**	.956**	.946**	.926**	.870**	.966**	.966**	.920**	*.997***
CCR-SMOTE-LR (L1)	.851**	.933**	.889**	.872**	.729**	.800**	.827**	.780**	.663**	.947**	.927**	.654**
CCR-SMOTE-LR (L2)	.850**	.934**	.895**	.873**	.736**	.802**	.824**	.786**	.689**	.886**	.920**	.676**
Ensemble												
RUSBoost	.808**	.957**	.974**	.842**	.848**	.868**	.865**	.929**	.714**	.969**	.823**	.779**
Proposed												
Soft-LR (L1)	.821**	.924**	.905**	.871**	.707**	.800**	.797**	.784**	.542**	.931**	.937**	.616**
Soft-LR (L2)	.814**	.932**	.893**	.872**	.715**	.803**	.808**	.782**	.533**	.917**	.935**	.601**
Soft-SVM	.761**	.914**	.886**	.885**	.684**	.870**	.928**	.805**	.535**	.763**	.820**	.555**

*$p < 0.05$ w.r.t. the best method.
**$p < 0.005$ w.r.t. the best method.

Table 11
Balanced accuracy comparison in the large-scale dataset (GPU kernel performance). The best method is in bold, and the second place is italic and underlined.

Dataset	Cost-sensitive		SMOTE		Ensemble	Regression-based		Proposed	
	LR (l1)	LR (l2)	LR (l1)	LR (l2)	RUSBoost	Lasso	Ridge	LR (l1)	LR (l2)
GPU-interaction-0.1%	*0.961*	0.959	0.958	0.959	**0.969**	0.500	0.500	0.951	0.951
GPU-interaction-0.005%	0.894	0.878	0.887	0.898	0.968	0.505	0.505	*0.986*	**0.986**

CRediT authorship contribution statement

Akira Tanimoto: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft. **So Yamada:** Software, Investigation, Writing – original draft. **Takashi Takenouchi:** Writing – review & editing. **Masashi Sugiyama:** Writing – review & editing. **Hisashi Kashima:** Writing – review & editing, Supervision.

Declaration of competing interest

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

Acknowledgments

TT was partially supported by JSPS KAKENHI Grant Numbers 20K03753 and 19H04071. HK was supported by the JSPS KAKENHI Grant Number 20H04244.

Appendix A. Proof of Theorem 4.1

First, we prepare a lemma to upper bound using the Lipschitz constant of the instance-wise loss function. In the contraction lemma of the Rademacher complexity (Shalev-Shwartz & Ben-David, 2014), the Lipschitz constant with respect to the scoring function value is constant for all instances. However, in the case of cost-sensitive loss, the Lipschitz constant is large (C_+) only for a small number of instances (positive), and it is small (C_-) for most of the instances (negative). Therefore, to get a tighter upper bound, it is preferable to evaluate the Lipschitz constant, instance by instance.

Lemma A.1 (*Element-wise Contraction*). *For each $n \in [N]$, let $\ell_n : \mathbb{R} \to \mathbb{R}$ be a ρ_n-Lipschitz function; namely, for all $\alpha, \beta \in \mathbb{R}$ we have $|\ell_n(\alpha) - \ell_n(\beta)| \leq \rho_n|\alpha - \beta|$. Then,*

$$R(\{\ell_n(a_n)\}) \leq R(\{\rho_n a_n\}),$$

where R is the Rademacher complexity.

Proof. First, we set an upper bound for an instance, $n = 1$. Let $p(\epsilon_n = 1) = 1/2$ and $p(\epsilon_n = -1) = 1/2$ for all $n \in [N]$.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\epsilon}\left[\sup_{\{a_n\}}\{\sum \epsilon_n \ell_n(a_n)\}\right] \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\epsilon_2,\ldots,\epsilon_N}\left[\sup_{\{a_n\}}\{\rho_1\ell(a_1) + \sum_{n=2}^{N}\epsilon_n\ell_n(a_n)\}\right. \\
&\quad \left. + \sup_{\{a_n\}}\{-\rho_1\ell(a_1) + \sum_{n=2}^{N}\epsilon_n\ell_n(a_n)\}\right] \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\epsilon_2,\ldots,\epsilon_N}\left[\sup_{\{a_n\},\{a'_n\}}\{\rho_1(\ell(a_1) - \ell(a'_1)) + \sum_{n=2}^{N}\epsilon_n\ell_n(a_n) + \sum_{n=2}^{N}\epsilon_n\ell_n(a'_n)\}\right] \\
&\leq \frac{1}{2}\mathop{\mathbb{E}}_{\epsilon_2,\ldots,\epsilon_N}\left[\sup_{\{a_n\},\{a'_n\}}\{\rho_1|a_1 - a'_1| + \sum_{n=2}^{N}\epsilon_n\ell_n(a_n) + \sum_{n=2}^{N}\epsilon_n\ell_n(a'_n)\}\right] \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\epsilon_2,\ldots,\epsilon_N}\left[\sup_{\{a_n\},\{a'_n\}}\{\rho_1(a_1 - a'_1) + \sum_{n=2}^{N}\epsilon_n\ell_n(a_n) + \sum_{n=2}^{N}\epsilon_n\ell_n(a'_n)\}\right] \\
&= \mathop{\mathbb{E}}_{\epsilon_1,\ldots,\epsilon_N}\left[\sup_{\{a_n\}}\{\epsilon_1\rho_1 a_1 + \sum_{n=2}^{N}\epsilon_n\ell_n(a_n)\}\right].
\end{aligned}$$

The inequality comes from the definition of the Lipschitz function. By applying this repeatedly for all instances, we get the lemma. □

Next, we provide the proof of the theorem. Let $g = w^\top x$ be the decision function value. Then $\ell(y, g(x))$ is m_n-Lipschitz w.r.t. g, where $m_n := \max\left\{C_+\frac{p_+}{p_{T,+}}\sigma(z/T), C_-\frac{p_-}{p_{T,-}}\sigma(-z/T)\right\}$

$$\begin{aligned}
&\mathop{\mathbb{E}}_{S}\left[L_T(\hat{w})\right] - \inf_{w:\|w\|_2\leq B} L_T(w) \\
&\leq 2\mathop{\mathbb{E}}_{S,\epsilon}\mathbb{E}\left[\sup_{w:\|w\|_2\leq B}\{\frac{1}{N}\sum\epsilon_n\ell(w, x_n, y_n)\}\right] \\
&\leq 2\mathop{\mathbb{E}}_{S,\epsilon}\left[\frac{1}{N}\sup_{\|w\|_2\leq 1}\sum_n \epsilon_n m_n w^\top x\right] && \text{(Lemma A.1)} \\
&= 2\mathop{\mathbb{E}}_{S,\epsilon}\left[\frac{B}{N}\|\sum_n \epsilon_n m_n x_n\|_2\right] \\
&\leq 2\mathop{\mathbb{E}}_{S}\left[\frac{B}{N}\sqrt{\mathop{\mathbb{E}}_{\epsilon}\left[\|\sum_n^N \epsilon_n m_n x_n\|_2^2\right]}\right] && \text{(Jensen's ineq.)} \\
&\leq \frac{2BX}{N}\sqrt{\mathop{\mathbb{E}}_{S}\left[\sum_n^N m_n^2\right]} && \text{(Jensen's ineq.)} \\
&= \frac{2BX}{N}\sqrt{\mathop{\mathbb{E}}_{S}\left[\sum_n^N\left(C_+\frac{p_+}{p_{T,+}}s_n\right)^2 I_+ + \left(C_-\frac{p_-}{p_{T,-}}(1-s_n)\right)^2 I_-\right]},
\end{aligned}$$

where $I_+ := I(C_+\frac{p_+}{p_{T,+}}s_n \geq C_-\frac{p_-}{p_{T,-}}(1-s_n))$ and $I_- := I(C_+\frac{p_+}{p_{T,+}}s_n < C_-\frac{p_-}{p_{T,-}}(1-s_n))$. The first inequality comes form Theorem 26.3 in Shalev-Shwartz and Ben-David (2014).

Since $s_n^2 \leq s_n$, $I_\pm \leq 1$ and $\mathbb{E}_S[\sum_n^N s_n] = Np_{T,+}$, the r.h.s. is bounded as follows:

$$\text{r.h.s.} \leq \frac{2BX}{\sqrt{N}}\sqrt{C_+^2\frac{p_+^2}{p_{T,+}} + C_-^2\frac{p_-^2}{p_{T,-}}},$$

which concludes the proof.

Appendix B. Proof of Proposition 4.2

The additional bias can be rewritten as

$$\begin{aligned}
&(\text{bias1} + \text{bias2}) = \\
&\quad \mathop{\mathbb{E}}_{x}\left[\Delta_+\mathop{\mathbb{E}}_{S}\left[\ell(g^*(x)) - \ell(\hat{g}(x))\right]\right. \\
&\qquad \left. + \Delta_-\mathop{\mathbb{E}}_{S}\left[\ell(-g^*(x)) - \ell(-\hat{g}(x))\right]\right],
\end{aligned} \tag{B.1}$$

where $\Delta_+ := \mathbb{E}_{z|x}\left[C_{T,+}\sigma(z/T) - C_+ I(z \geq 0)\right]$ and $\Delta_- := \mathbb{E}_{z|x}\left[C_{T,-}\sigma(-z/T) - C_- I(z < 0)\right]$.

From Hölder's inequality,

$$\begin{aligned}
\text{r.h.s. of (B.1)} \leq &\mathop{\mathbb{E}}_{x}\left[|\Delta_+|\right]\max_{x:p(x)>0}\left|\mathop{\mathbb{E}}_{S}\left[\ell(g^*(x)) - \ell(\hat{g}(x))\right]\right| \\
&+ \mathop{\mathbb{E}}_{x}\left[|\Delta_-|\right]\max_{x:p(x)>0}\left|\mathop{\mathbb{E}}_{S}\left[\ell(-g^*(x)) - \ell(-\hat{g}(x))\right]\right| \\
\leq &c\left(\mathop{\mathbb{E}}_{x}\left[|\Delta_+|\right] + \mathop{\mathbb{E}}_{x}\left[|\Delta_-|\right]\right),
\end{aligned} \tag{B.2}$$

where $\Delta_+ := \mathbb{E}_{z|x}\left[C_{T,+}\sigma(z/T) - C_+ I(z \geq 0)\right]$ and $\Delta_- := \mathbb{E}_{z|x}\left[C_{T,-}\sigma(-z/T) - C_- I(z < 0)\right]$ are differences in weighted labels. From the definition of η,

$$\begin{aligned}
p(\eta = 1) &= \mathop{\mathbb{E}}_{z}\left[\sigma(z/T)\right] = p_{T,+}, \\
p(\eta = 1|x) &= \mathop{\mathbb{E}}_{z|x}\left[\sigma(z/T)\right].
\end{aligned}$$

And thus,

$$\frac{1}{2p_{T,+}}\mathop{\mathbb{E}}_{z|x}\left[\sigma(z/T)\right]p(x) = \frac{1}{2}\frac{p(\eta = 1|x)p(x)}{p(\eta = 1)} = \frac{1}{2}p(x|\eta = 1).$$

Samely,

$$\frac{1}{2p_+}\mathop{\mathbb{E}}_{z|x}\left[I(z \geq 0)\right]p(x) = \frac{1}{2}p(x|y = 1).$$

Therefore, in the BER minimization setting, i.e., $C_+ = 1/2p_+$ and $C_{T,+} = 1/2p_{T,+}$,

$$\begin{aligned}\mathop{\mathbb{E}}_{x}\left[|\Delta_+|\right] &= \int \left| \mathop{\mathbb{E}}_{z|x}\left[\frac{1}{2p_{T,+}}\sigma(z/T) - \frac{1}{2p_+}I(z \geq 0)\right]\right| p(x)\mathrm{d}x \\ &= \frac{1}{2}\int |p(x|\eta = 1) - p(x|y = 1)|\,\mathrm{d}x \\ &= \mathrm{TV}\left(p(x|\eta = 1), p(x|y = 1)\right).\end{aligned}$$

Samely,

$$\begin{aligned}\mathop{\mathbb{E}}_{x}\left[|\Delta_-|\right] &= \int \left| \mathop{\mathbb{E}}_{z|x}\left[\frac{1}{2p_{T,-}}\sigma(-z/T) - \frac{1}{2p_-}I(z \leq 0)\right]\right| p(x)\mathrm{d}x \\ &= \frac{1}{2}\int |p(x|\eta = 0) - p(x|y = 0)|\,\mathrm{d}x \\ &= \mathrm{TV}\left(p(x|\eta = 0), p(x|y = 0)\right).\end{aligned}$$

By substituting $\mathbb{E}_x\left[|\Delta_+|\right]$ and $\mathbb{E}_x\left[|\Delta_-|\right]$ in (B.2), we get the proposition.

Appendix C. Experimental conditions

C.1. Computing infrastructure

All the experiments were run on a machine with eight CPUs (Intel Xeon E7-8850 2.0 GHz, ten cores) and 1.0TB RAM.

C.2. Data preprocesses

We here describe the preprocesses for real datasets. First, we describe the common preprocesses for all datasets and then describe preprocesses for each dataset.

Common preprocesses: We applied the following preprocesses for all the datasets.

- The standardization, i.e., scaling and shifting so as to $\mathbb{E}[x] = 0$ and $\mathrm{Var}[x] = 1$ for each feature, was applied.
- The binary expansion was applied to categorical features, i.e., a categorical feature that has k categories are expanded into $k - 1$ binary features. The first category in the alphabetical order was not expanded.
- For datasets that has multiple files (wine quality and student datasets) are concatenated, and a categorical feature that represents the source files was added.
- Instances that have missing features were deleted.

Toy: The toy data shown in Fig. 2(a), which we used also in the experiments, was generated as follows. The coefficients w and the features x are drawn from 100-dimensional standard normal distribution, and then, positivity z is drawn as

$$z \sim \mathcal{N}(5\exp(w^\top x/15), 2).$$

Air quality: For the target attribute (CO(GT)), the value -200 means missing and thus removed. Categorical features named Date and Time was removed. In addition, a feature named NMHC(GT) was removed since there exist many missing entries.

Year prediction MSD: We sampled 10k instances at random.

C.3. Compared methods and hyperparameter ranges

The methods compared include conventional classification methods, regression-based methods, and a rank-based method, as listed below. We also describe here the hyperparameter ranges considered.

Hyperparameter ranges: The considered hyperparameter configurations are the following:

- The regularization strength was ranged from 10^{-2} to 10^2.
- T of our proposed method ranged from 10^{-3} to 10^2.
- γ of an RBF kernel $\exp(\gamma\|x - x'\|^2)$ ranged from 10^{-2} to 10^2.
- For the GP, the hyperparameter optimizer was restarted five times.
- For the SMOTE-based methods, the number of neighboring points used to synthesize over-sampled points was optimized from $[3, 5, 8]$.
- For the RUSBoost, the number of estimators was optimized from $[20, 30, 50]$, random state ranged 0–2.
- For the CCR, the energy was ranged from 10^{-2} to 10^2.

Models with our proposed method: As our method is model-agnostic, we performed experiments on different types of base classification learners. We adopted three models: logistic regression (LR) with L1 and L2 regularizers each, and a support vector machine (SVM) with an RBF kernel. The methods in this setting were as follows:

- proposed (base learner: LR (regularization: L1))
- proposed (base learner: LR (regularization: L2))
- proposed (base learner: SVM (kernel: radial basis function))

Conventional classification methods: These models use only the binary label $y \in \{0, 1\}$, not the numerical mediator $z \in \mathbb{R}$. We adopted the same models as those for the proposed method, namely LR with L1 and L2 regularizers each and SVM for the cost-sensitive classification and SMOTE. In a manner similar to that with the proposed method, the sample weights were rebalanced in the cost-sensitive classification. In other words, we learned from the data set $D = \{d_1, d_2, \ldots, d_N\}$, where $d_n = (\mathbf{x}_n, y_n)$ consists of a feature vector and a class label. This setting was normal classification. The models compared in this setting were as follows:

- LR (regularization: L1)
- LR (regularization: L2)
- SVM (kernel: radial basis function)

Also, we compared the RUSBoost, which utilize the boosting method as the base learner.

Regression-based methods: These methods learn and predict $z \in \mathbb{R}$, and then apply the threshold to the prediction. We adopted Lasso regression, Ridge regression, and a Gaussian process with an RBF kernel. In other words, we learned from the data set $D = \{d_1, d_2, \ldots, d_N\}$, where $d_n = (\mathbf{x}_n, z_n)$ consists of a feature vector and a target variable. This setting was normal regression. The models compared in this setting were as follows:

- Lasso regression
- Ridge regression
- Gaussian process (kernel: radial basis function)

Rank-based method: This model is based on a pair-wise ranking method in which the rank information is extracted from z. It learned a ranking function $r(\cdot)$. In $\{(\mathbf{x}_i, \mathbf{x}_j) \mid z_i > z_j\}$, the model was optimized to satisfy the pair-wise rank constraints: $r(\mathbf{x}_i) > r(\mathbf{x}_j)$ or $r(\mathbf{x}_i) - r(\mathbf{x}_j) = 0$, that is $\mathbf{w}^\top(\mathbf{x}_i - \mathbf{x}_j) = 0$. In general, the linear SVM with slack variables is commonly used for the pair-wise ranking method because of its computational-efficiency. The model employed in this setting was as follows:

- Rank-SVM (kernel: linear)

Appendix D. Additional experimental results

We present additional experimental results for Section Section 5.3. Fig. D.5 shows the results in the same setting with respect to BA and ROC-AUC. Fig. D.6 shows the results under various sample sizes with fixed positive rate $p_+ = 1\%$.

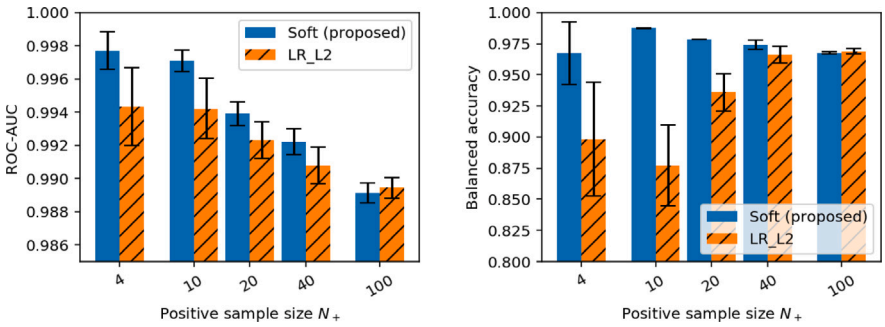


Fig. D.5. Performance of the proposed method and conventional cost-sensitive classification with respect to AUC and BA for the GPU kernel performance dataset under highly imbalanced conditions. Positive rate $p_+ := \sum I(z \geq 0)/N$ ranged from 2×10^{-5} to 1×10^{-2}. Error bars indicate standard error. Note that, the task-specific threshold θ differs depending on p_+ settings, therefore performances in different p_+ cannot be compared with each other.

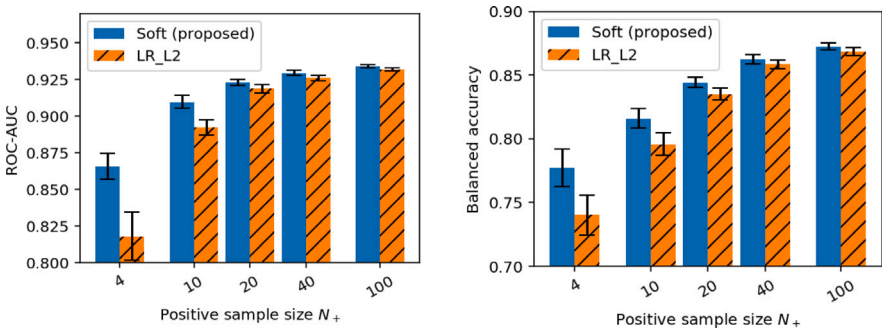


Fig. D.6. Performance of the proposed method and conventional cost-sensitive classification with respect to AUC and BA for the GPU kernel performance dataset under $p_+ = 1\%$ and various training sample sizes. Error bars indicate standard error.

References

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

Ali-Gombe, A., & Elyan, E. (2019). MFC-GAN: Class-imbalanced dataset classification using multiple fake class generative adversarial network. *Neurocomputing*, *361*, 212–221.

Ballester-Ripoll, R., Paredes, E. G., & Pajarola, R. (2017). Sobol tensor trains for global sensitivity analysis. arXiv preprint arXiv:1712.00233.

Barach, P., & Small, S. D. (2000). Reporting and preventing medical mishaps: lessons from non-medical near miss reporting systems. *BMJ: British Medical Journal*, *320*(7237), 759.

Barua, S., Islam, M. M., Yao, X., & Murase, K. (2012). MWMOTE–Majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering*, *26*(2), 405–425.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Pacific-Asia conference on knowledge discovery and data mining (PAKDD)* (pp. 475–482). Springer.

Chang, P.-C., & Lai, C.-Y. (2005). A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book forecasting. *Expert Systems with Applications*, *29*(1), 183–192.

Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning (adaptive computation and machine learning)*. The MIT Press.

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*, 321–357.

Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery (PKDD)* (pp. 107–119). Springer.

Chen, X.-w., & Wasikowski, M. (2008). Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems. In *ACM SIGKDD international conference on knowledge discovery and data mining (KDD)* (pp. 124–132). ACM.

Cloke, H., & Pappenberger, F. (2009). Ensemble flood forecasting: A review. *Journal of Hydrology*, *375*(3–4), 613–626.

Cohen, G., Hilario, M., Sax, H., Hugonnet, S., & Geissbuhler, A. (2006). Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, *37*(1), 7–18.

Cook, J., & Ramadas, V. (2020). When to consult precision-recall curves. *The Stata Journal*, *20*(1), 131–148.

Dheeru, D., & Karra Taniskidou, E. (2017). UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

Dmochowski, J. P., Sajda, P., & Parra, L. C. (2010). Maximum likelihood in cost-sensitive learning: Model specification, approximations, and upper bounds. *Journal of Machine Learning Research*, *11*(Dec), 3313–3332.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence (IJCAI)* (pp. 973–978). Lawrence Erlbaum Associates Ltd.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, *61*, 863–905.

Fuqua, D., & Razzaghi, T. (2020). A cost-sensitive convolution neural network learning for control chart pattern recognition. *Expert Systems with Applications*, *150*, Article 113275.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220–239.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing* (pp. 878–887). Springer.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE international joint conference on neural networks (IJCNN)* (pp. 1322–1328). IEEE.

Heinrich, H. W., Petersen, D. C., Roos, N. R., & Hazlett, S. (1980). *Industrial accident prevention: A safety management approach*. McGraw-Hill Companies.

Hendricks, K., & Sorensen, A. (2009). Information and the skewness of music sales. *Journal of Political Economy*, *117*(2), 324–369.

Herremans, D., Martens, D., & Sörensen, K. (2014). Dance hit song prediction. *Journal of New Music Research*, *43*(3), 291–302.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS deep learning and representation learning workshop*. URL: http://arxiv.org/abs/1503.02531.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429–449.

Jones, S., Kirchsteiger, C., & Bjerke, W. (1999). The importance of near miss reporting to further improve safety performance. *Journal of Loss Prevention in the Process Industries*, *12*(1), 59–67.

Kim, K. H., & Sohn, S. Y. (2020). Hybrid neural network with cost-sensitive support vector machine for class-imbalanced multimodal data. *Neural Networks*, *130*, 176–184.

Koziarski, M., & Woźniak, M. (2017). CCR: A combined cleaning and resampling algorithm for imbalanced data classification. *International Journal of Applied Mathematics and Computer Science*, *27*(4).

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 221–232.

Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., & Liao, H. (2006). Intelligent prognostics tools and e-maintenance. *Computers in Industry*, *57*(6), 476–489.

Lee, K.-c., Orten, B., Dasdan, A., & Li, W. (2012). Estimating conversion rate in display advertising from past erformance data. In *SIGKDD international conference on knowledge discovery and data mining (KDD)* (pp. 768–776).

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, *5*(1), 42.

Li, Y., & Nilkitsaranont, P. (2009). Gas turbine performance prognostic for condition-based maintenance. *Applied Energy*, *86*(10), 2152–2161.

Ling, C. X., & Sheng, V. S. (2008). Cost-sensitive learning and the class imbalance problem. *Encyclopedia of Machine Learning*, *2011*, 231–235.

Lopez-Paz, D., Bottou, L., Schölkopf, B., & Vapnik, V. (2016). Unifying distillation and privileged information. In *International conference on learning representations (ICLR)* (pp. 1–10).

Napierała, K., Stefanowski, J., & Wilk, S. (2010). Learning from imbalanced data in presence of noisy and borderline examples. In *International conference on rough sets and current trends in computing* (pp. 158–167). Springer.

Natarajan, N., Dhillon, I. S., Ravikumar, P., & Tewari, A. (2017). Cost-sensitive learning with noisy labels. *Journal of Machine Learning Research*, *18*(1), 5666–5698.

Nguyen, Q., Valizadegan, H., & Hauskrecht, M. (2011). Learning classification with auxiliary probabilistic information. In *IEEE international conference on data mining (ICDM)* (pp. 477–486). http://dx.doi.org/10.1109/ICDM.2011.84.

Nguyen, Q., Valizadegan, H., & Hauskrecht, M. (2014). Learning classification models with soft-label information. *Journal of the American Medical Informatics Association*, *21*(3), 501–508.

Nguyen, Q., Valizadegan, H., Seybert, A., & Hauskrecht, M. (2011). Sample-efficient learning with auxiliary class-label information. In *AMIA annual symposium proceedings, Vol. 2011* (p. 1004). American Medical Informatics Association.

Nugteren, C., & Codreanu, V. (2015). CLTune: A generic auto-tuner for opencl kernels. In *IEEE international symposium on Embedded multicore/many-core systems-on-chip (MCSoC)* (pp. 195–202). IEEE.

Peng, P., Wong, R. C.-W., & Yu, P. S. (2014). Learning on probabilistic labels. In *SIAM international conference on data mining (SDM)* (pp. 307–315). SIAM.

Sáez, J. A., Luengo, J., Stefanowski, J., & Herrera, F. (2015). SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, *291*, 184–203.

van der Schaaf, T. W. (1995). Near miss reporting in the chemical process industry: An overview. *Microelectronics Reliability*, *35*(9–10), 1233–1243.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., & Mooij, J. (2012). On causal and anticausal learning. In *International conference on machine learning (ICML)* (pp. 1255–1262).

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2009). RUSboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *40*(1), 185–197.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: from theory to algorithms*. Cambridge University Press.

Stapor, K., Ksieniewicz, P., García, S., & Woźniak, M. (2021). How to design the fair experimental classifier evaluation. *Applied Soft Computing*, *104*, Article 107219.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2818–2826).

Torgo, L. (2018). Regression data sets, 2001. URL http://www.liaad.up.pt/~ltorgo/regression/datasets.html.

Vapnik, V., & Vashist, A. (2009). A new learning paradigm: Learning using privileged information. *Neural Networks*, *22*(5–6), 544–557.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, *7*(1), 91.

Vasile, F., Lefortier, D., & Chapelle, O. (2017). Cost-sensitive learning for utility optimization in online advertising auctions. In *International Workshop on Data Mining for Online Advertising (ADKDD)* (p. 8). ACM.

Wei, J., Huang, H., Yao, L., Hu, Y., Fan, Q., & Huang, D. (2020). NI-MWMOTE: AN improving noise-immunity majority weighted minority oversampling technique for imbalanced classification problems. *Expert Systems with Applications*, Article 113504.

Xue, Y., & Hauskrecht, M. (2016). Learning of classification models from noisy soft-labels. In *European Conference on Artificial Intelligence (ECAI)* (pp. 1618–1619). http://dx.doi.org/10.3233/978-1-61499-672-9-1618.

Xue, Y., & Hauskrecht, M. (2017). Efficient learning of classification models from soft-label information by binning and ranking. In *International florida AI research society conference. Florida AI research symposium* (p. 164).

Zhao, S., Ren, H., Yuan, A., Song, J., Goodman, N., & Ermon, S. (2018). Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, *31*.