Mach Learn (2018) 107:767–794
https://doi.org/10.1007/s10994-017-5678-9

Semi-supervised AUC optimization based on positive-unlabeled learning

Tomoya Sakai[1,2] · **Gang Niu**[1,2] · **Masashi Sugiyama**[1,2]

Received: 29 April 2017 / Accepted: 16 September 2017 / Published online: 26 October 2017
© The Author(s) 2017

Abstract Maximizing the area under the receiver operating characteristic curve (AUC) is a standard approach to imbalanced classification. So far, various supervised AUC optimization methods have been developed and they are also extended to semi-supervised scenarios to cope with small sample problems. However, existing semi-supervised AUC optimization methods rely on strong distributional assumptions, which are rarely satisfied in real-world problems. In this paper, we propose a novel semi-supervised AUC optimization method that does not require such restrictive assumptions. We first develop an AUC optimization method based only on positive and unlabeled data and then extend it to semi-supervised learning by combining it with a supervised AUC optimization method. We theoretically prove that, without the restrictive distributional assumptions, unlabeled data contribute to improving the generalization performance in PU and semi-supervised AUC optimization methods. Finally, we demonstrate the practical usefulness of the proposed methods through experiments.

Keywords AUC optimization · Learning from positive and unlabeled data · Semi-supervised learning

Editors: Wee Sun Lee and Robert Durrant.

The original version of this article was revised: Corrections made to title, author affiliations, and equations on pp. 7 and 26.

✉ Tomoya Sakai
sakai@ms.k.u-tokyo.ac.jp

Gang Niu
gang@ms.k.u-tokyo.ac.jp

Masashi Sugiyama
sugi@k.u-tokyo.ac.jp

1 Center for Advanced Intelligence Project, RIKEN, Nihonbashi, Chuo-ku, Tokyo, Japan

2 Graduate School of Frontier Sciences, The University of Tokyo, Kashiwanoha, Kashiwa-shi, Chiba, Japan

1 Introduction

Maximizing the *area under the receiver operating characteristic curve* (AUC) (Hanley and McNeil 1982) is a standard approach to imbalanced classification (Cortes and Mohri 2004). While the misclassification rate relies on the sign of the score of a single sample, AUC is governed by the ranking of the scores of two samples. Based on this principle, various supervised methods for directly optimizing AUC have been developed so far and demonstrated to be useful (Herschtal and Raskutti 2004; Zhao et al. 2011; Rakhlin et al. 2012; Kotlowski et al. 2011; Ying et al. 2016).

However, collecting labeled samples is often expensive and laborious in practice. To mitigate this problem, *semi-supervised* AUC optimization methods have been developed that can utilize unlabeled samples (Amini et al. 2008; Fujino and Ueda 2016). These semi-supervised methods solely rely on the assumption that an unlabeled sample that is "similar" to a labeled sample shares the same label. However, such a restrictive distributional assumption (which is often referred to as the cluster or the entropy minimization principle) is rarely satisfied in practice and thus the practical usefulness of these semi-supervised methods is limited (Cozman et al. 2003; Sokolovska et al. 2008; Li and Zhou 2015; Krijthe and Loog 2017).

On the other hand, it has been recently shown that unlabeled data can be effectively utilized without such restrictive distributional assumptions in the context of *classification from positive and unlabeled data* (PU classification) (du Plessis et al. 2014). Furthermore, based on recent advances in PU classification (du Plessis et al. 2014, 2015; Niu et al. 2016), a novel semi-supervised classification approach has been developed that combines supervised classification with PU classification (Sakai et al. 2017). This approach inherits the advances of PU classification that the restrictive distributional assumptions are not necessary and is demonstrated to perform excellently in experiments.

Following this line of research, we first develop an AUC optimization method from positive and unlabeled data (PU-AUC) in this paper. Previously, a *pairwise ranking* method for PU data has been developed (Sundararajan et al. 2011), which can be regarded as an AUC optimization method for PU data. However, it merely regards unlabeled data as negative data and thus the obtained classifier is biased. On the other hand, our PU-AUC method is unbiased and we theoretically prove that unlabeled data contribute to reducing an upper bound on the generalization error with the optimal parametric convergence rate without the restrictive distributional assumptions.

Then we extend our PU-AUC method to the semi-supervised setup by combining it with a supervised AUC optimization method. Theoretically, we again prove that unlabeled data contribute to reducing an upper bound on the generalization error with the optimal parametric convergence rate without the restrictive distributional assumptions, and further we prove that the variance of the empirical risk of our semi-supervised AUC optimization method can be smaller than that of the plain supervised counterpart. The latter claim suggests that the proposed semi-supervised empirical risk is also useful in the cross-validation phase. Finally, we experimentally demonstrate the usefulness of the proposed PU and semi-supervised AUC optimization methods.

2 Preliminary

We first describe our problem setting and review an existing supervised AUC optimization method.

Let covariate $\boldsymbol{x} \in \mathbb{R}^d$ and its corresponding label $y \in \{\pm 1\}$ be equipped with probability density $p(\boldsymbol{x}, y)$, where d is a positive integer. Suppose we have sets of positive and negative samples:

$$\mathcal{X}_{\mathrm{P}} := \{\boldsymbol{x}_i^{\mathrm{P}}\}_{i=1}^{n_{\mathrm{P}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{P}}(\boldsymbol{x}) := p(\boldsymbol{x} \mid y = +1), \text{ and}$$
$$\mathcal{X}_{\mathrm{N}} := \{\boldsymbol{x}_j^{\mathrm{N}}\}_{j=1}^{n_{\mathrm{N}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{N}}(\boldsymbol{x}) := p(\boldsymbol{x} \mid y = -1).$$

Furthermore, let $g \colon \mathbb{R}^d \to \mathbb{R}$ be a decision function and classification is carried out based on its sign: $\widehat{y} = \operatorname{sign}(g(\boldsymbol{x}))$.

The goal is to train a classifier g by maximizing the AUC (Hanley and McNeil 1982; Cortes and Mohri 2004) defined and expressed as

$$\begin{aligned}\mathrm{AUC}(g) &:= \mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{N}}[I(g(\boldsymbol{x}^{\mathrm{P}}) \geq g(\boldsymbol{x}^{\mathrm{N}}))]\right] \\ &= 1 - \mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\mathrm{N}}\left[I(g(\boldsymbol{x}^{\mathrm{P}}) < g(\boldsymbol{x}^{\mathrm{N}}))\right]] \\ &= 1 - \mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\mathrm{N}}\left[\ell_{0\text{-}1}(g(\boldsymbol{x}^{\mathrm{P}}) - g(\boldsymbol{x}^{\mathrm{N}}))\right]], \end{aligned} \tag{1}$$

where E_{P} and E_{N} be the expectations over $p_{\mathrm{P}}(\boldsymbol{x})$ and $p_{\mathrm{N}}(\boldsymbol{x})$, respectively. $I(\cdot)$ is the indicator function, which is replaced with the *zero-one* loss, $\ell_{0\text{-}1}(m) = (1 - \operatorname{sign}(m))/2$, to obtain the last equation. Let

$$f(\boldsymbol{x}, \boldsymbol{x}') := g(\boldsymbol{x}) - g(\boldsymbol{x}')$$

be a composite classifier. Maximizing the AUC corresponds to minimizing the second term in Eq. (1). Practically, to avoid the discrete nature of the zero-one loss, we replace the zero-one loss with a surrogate loss $\ell(m)$ and consider the following PN-AUC risk (Herschtal and Raskutti 2004; Kotlowski et al. 2011; Rakhlin et al. 2012):

$$R_{\mathrm{PN}}(f) := \mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{N}}))]\right]. \tag{2}$$

In practice, we train a classifier by minimizing the empirical PN-AUC risk defined as

$$\widehat{R}_{\mathrm{PN}}(f) := \frac{1}{n_{\mathrm{P}} n_{\mathrm{N}}} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{j=1}^{n_{\mathrm{N}}} \ell\left(f\left(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_j^{\mathrm{N}}\right)\right).$$

Similarly to the *classification-calibrated* loss (Bartlett et al. 2006) in misclassification rate minimization, the consistency of AUC optimization in terms of loss functions has been studied recently (Gao and Zhou 2015; Gao et al. 2016). They showed that minimization of the AUC risk with a consistent loss function is asymptotically equivalent to that with the zero-one loss function. The squared loss $\ell_{\mathrm{S}}(m) := (1 - m)^2$, the exponential loss $\ell_{\mathrm{E}}(m) := \exp(-m)$, and the logistic loss $\ell_{\mathrm{L}}(m) := \log(1 + \exp(-m))$ are shown to be consistent, while the hinge loss $\ell_{\mathrm{H}}(m) := \max(0, 1 - m)$ and the absolute loss $\ell_{\mathrm{A}}(m) := |1 - m|$ are *not* consistent.

3 Proposed method

In this section, we first propose an AUC optimization method from positive and unlabeled data and then extend it to a semi-supervised AUC optimization method.

3.1 PU-AUC optimization

In PU learning, we do not have negative data while we can use unlabeled data drawn from marginal density $p(\boldsymbol{x})$ in addition to positive data:

$$\mathcal{X}_{\mathrm{U}} := \{\boldsymbol{x}_k^{\mathrm{U}}\}_{k=1}^{n_{\mathrm{U}}} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}) = \theta_{\mathrm{P}} p_{\mathrm{P}}(\boldsymbol{x}) + \theta_{\mathrm{N}} p_{\mathrm{N}}(\boldsymbol{x}), \tag{3}$$

where

$$\theta_{\mathrm{P}} := p(y=+1) \text{ and } \theta_{\mathrm{N}} := p(y=-1).$$

We derive an equivalent expression to the PN-AUC risk that depends only on positive and unlabeled data distributions without the negative data distribution. In our derivation and theoretical analysis, we assume that θ_{P} and θ_{N} are known. In practice, they are replaced by their estimate obtained, e.g., by du Plessis et al. (2017), Kawakubo et al. (2016), and references therein.

From the definition of the marginal density in Eq. (3), we have

$$\begin{aligned}\mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{U}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}}))]\right] &= \theta_{\mathrm{P}}\,\mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\overline{\mathrm{P}}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]\right] + \theta_{\mathrm{N}}\,\mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{N}}))]\right] \\ &= \theta_{\mathrm{P}}\,\mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\overline{\mathrm{P}}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]\right] + \theta_{\mathrm{N}} R_{\mathrm{PN}}(f),\end{aligned}$$

where $\mathrm{E}_{\overline{\mathrm{P}}}$ denotes the expectation over $p_{\mathrm{P}}(\overline{\boldsymbol{x}}^{\mathrm{P}})$. Dividing the above equation by θ_{N} and rearranging it, we can express the PN-AUC risk in Eq. (2) based on PU data (the PU-AUC risk) as

$$R_{\mathrm{PN}}(f) = \frac{1}{\theta_{\mathrm{N}}}\mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{U}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}}))]\right] - \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}}\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\overline{\mathrm{P}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))\right]] := R_{\mathrm{PU}}(f). \tag{4}$$

We refer to the method minimizing the PU-AUC risk as *PU-AUC optimization*. We will theoretically investigate the superiority of R_{PU} in Sect. 4.1.

To develop a semi-supervised AUC optimization method later, we also consider AUC optimization form *negative* and unlabeled data, which can be regarded as a mirror of PU-AUC optimization. From the definition of the marginal density in Eq. (3), we have

$$\begin{aligned}\mathrm{E}_{\mathrm{U}}\left[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{U}}, \boldsymbol{x}^{\mathrm{N}}))]\right] &= \theta_{\mathrm{P}}\,\mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{N}}))]\right] + \theta_{\mathrm{N}}\,\mathrm{E}_{\mathrm{N}}\left[\mathrm{E}_{\overline{\mathrm{N}}}[\ell(f(\boldsymbol{x}^{\mathrm{N}}, \overline{\boldsymbol{x}}^{\mathrm{N}}))]\right] \\ &= \theta_{\mathrm{P}} R_{\mathrm{PN}}(f) + \theta_{\mathrm{N}}\,\mathrm{E}_{\mathrm{N}}\left[\mathrm{E}_{\overline{\mathrm{N}}}[\ell(f(\boldsymbol{x}^{\mathrm{N}}, \overline{\boldsymbol{x}}^{\mathrm{N}}))]\right],\end{aligned}$$

where $\mathrm{E}_{\overline{\mathrm{N}}}$ denotes the expectation over $p_{\mathrm{N}}(\overline{\boldsymbol{x}}^{\mathrm{N}})$. Rearranging the above equation, we can obtain the PN-AUC risk in Eq. (2) based on negative and unlabeled data (the NU-AUC risk):

$$R_{\mathrm{PN}}(f) = \frac{1}{\theta_{\mathrm{P}}}\mathrm{E}_{\mathrm{U}}\left[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{U}}, \boldsymbol{x}^{\mathrm{N}}))]\right] - \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}}\mathrm{E}_{\mathrm{N}}\left[\mathrm{E}_{\overline{\mathrm{N}}}[\ell(f(\boldsymbol{x}^{\mathrm{N}}, \overline{\boldsymbol{x}}^{\mathrm{N}}))]\right] := R_{\mathrm{NU}}(f). \tag{5}$$

We refer to the method minimizing the NU-AUC risk as *NU-AUC optimization*.

3.2 Semi-supervised AUC optimization

Next, we propose a novel semi-supervised AUC optimization method based on positive-unlabeled learning. The idea is to combine the PN-AUC risk with the PU-AUC/NU-AUC risks, similarly to Sakai et al. (2017).[1]

[1] In Sakai et al. (2017), the combination of the PU and NU risks has also considered and found to be less favorable than the combination of the PN and PU/NU risks. For this reason, we focus on the latter in this paper.

First of all, let us define the PNPU-AUC and PNNU-AUC risks as

$$R^{\gamma}_{\mathrm{PNPU}}(f) := (1-\gamma)R_{\mathrm{PN}}(f) + \gamma R_{\mathrm{PU}}(f),$$
$$R^{\gamma}_{\mathrm{PNNU}}(f) := (1-\gamma)R_{\mathrm{PN}}(f) + \gamma R_{\mathrm{NU}}(f),$$

where $\gamma \in [0, 1]$ is the combination parameter. We then define the PNU-AUC risk as

$$R^{\eta}_{\mathrm{PNU}}(f) := \begin{cases} R^{\eta}_{\mathrm{PNPU}}(f) & (\eta \geq 0), \\ R^{-\eta}_{\mathrm{PNNU}}(f) & (\eta < 0), \end{cases} \tag{6}$$

where $\eta \in [-1, 1]$ is the combination parameter. We refer to the method minimizing the PNU-AUC risk as *PNU-AUC optimization*. We will theoretically discuss the superiority of $R^{\gamma}_{\mathrm{PNPU}}$ and $R^{\gamma}_{\mathrm{PNNU}}$ in Sect. 4.1.

3.3 Discussion about related work

Sundararajan et al. (2011) proposed a pairwise ranking method for PU data, which can be regarded as an AUC optimization method for PU data. Their approach simply regards unlabeled data as negative data and the ranking SVM (Joachims 2002) is applied to PU data so that the score of positive data tends to be higher than that of unlabeled data. Although this approach is simple and shown computationally efficient in experiments, the obtained classifier is biased. From the mathematical viewpoint, the existing method ignores the second term in Eq. (4) and maximizes only the first term with the hinge loss function. However, the effect of ignoring the second term is not negligible when the class prior, θ_{P}, is not sufficiently small. In contrast, our proposed PU-AUC risk includes the second term so that the PU-AUC risk is equivalent to the PN-AUC risk.

Our semi-supervised AUC optimization method can be regarded as an extension of the work by Sakai et al. (2017). They considered the misclassification rate as a measure to train a classifier and proposed a semi-supervised classification method based on the recently proposed PU classification method (du Plessis et al. 2014, 2015). On the other hand, we train a classifier by maximizing the AUC, which is a standard approach for imbalanced classification. To this end, we first developed an AUC optimization method for PU data, and then extended it to a semi-supervised AUC optimization method. Thanks to the AUC maximization formulation, our proposed method is expected to perform better than the method proposed by Sakai et al. (2017) for imbalanced data sets.

4 Theoretical analyses

In this section, we theoretically analyze the proposed risk functions. We first derive generalization error bounds of our methods and then discuss variance reduction.

4.1 Generalization error bounds

Recall the composite classifier $f(\boldsymbol{x}, \boldsymbol{x}') = g(\boldsymbol{x}) - g(\boldsymbol{x}')$. As the classifier g, we assume the linear-in-parameter model given by

$$g(\boldsymbol{x}) = \sum_{\ell=1}^{b} w_{\ell}\phi(\boldsymbol{x}) = \boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}),$$

where $^\top$ denotes the transpose of vectors and matrices, b is the number of basis functions, $\boldsymbol{w} = (w_1, \dots, w_b)^\top$ is a parameter vector, and $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \dots, \phi_b(\boldsymbol{x}))^\top$ is a basis function vector. Let $\mathcal{F}$ be a function class of bounded hyperplanes:

$$\mathcal{F} := \{f(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{w}^\top(\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{x}')) \mid \|\boldsymbol{w}\| \le C_{\boldsymbol{w}};\ \forall \boldsymbol{x}\colon \|\boldsymbol{\phi}(\boldsymbol{x})\| \le C_{\boldsymbol{\phi}}\},$$

where $C_{\boldsymbol{w}} > 0$ and $C_{\boldsymbol{\phi}} > 0$ are certain positive constants. This assumption is reasonable because the ℓ_2-regularizer included in training and the use of bounded basis functions, e.g., the Gaussian kernel basis, ensure that the minimizer of the empirical AUC risk belongs to such the function class $\mathcal{F}$. We assume that a surrogate loss is bounded from above by C_ℓ and denote the Lipschitz constant by L. For simplicity,[2] we focus on a surrogate loss satisfying $\ell_{0\text{-}1}(m) \le \ell(m)$. For example, the squared loss and the exponential loss satisfy the condition.[3]

Let

$$I(f) = \mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{N}}[\ell_{0\text{-}1}(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{N}}))]\right]$$

be the generalization error of f in AUC optimization. For convenience, we define

$$h(\delta) := 2\sqrt{2}LC_\ell C_{\boldsymbol{w}} C_{\boldsymbol{\phi}} + \frac{3}{2}\sqrt{2\log(2/\delta)}.$$

In the following, we prove the generalization error bounds of both PU and semi-supervised AUC optimization methods.

For the PU-AUC/NU-AUC risks, we prove the following generalization error bounds (its proof is available in Appendix 1):

Theorem 1 *For any $\delta > 0$, the following inequalities hold separately with probability at least $1 - \delta$ for all $f \in \mathcal{F}$:*

$$I(f) \le \widehat{R}_{\mathrm{PU}}(f) + h(\delta/2)\left(\frac{1}{\theta_{\mathrm{N}}\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{U}})}} + \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}\sqrt{n_{\mathrm{P}}}}\right),$$
$$I(f) \le \widehat{R}_{\mathrm{NU}}(f) + h(\delta/2)\left(\frac{1}{\theta_{\mathrm{P}}\sqrt{\min(n_{\mathrm{N}}, n_{\mathrm{U}})}} + \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}\sqrt{n_{\mathrm{N}}}}\right),$$

where $\widehat{R}_{\mathrm{PU}}$ and $\widehat{R}_{\mathrm{NU}}$ are unbiased empirical risk estimators corresponding to R_{PU} and R_{NU}, respectively.

Theorem 1 guarantees that $I(f)$ can be bounded from above by the empirical risk, $\widehat{R}(f)$, plus the confidence terms of order

$$\mathcal{O}_p\left(\frac{1}{\sqrt{n_{\mathrm{P}}}} + \frac{1}{\sqrt{n_{\mathrm{U}}}}\right) \text{ and } \mathcal{O}_p\left(\frac{1}{\sqrt{n_{\mathrm{N}}}} + \frac{1}{\sqrt{n_{\mathrm{U}}}}\right).$$

Since n_{P} (n_{N}) and n_{U} can increase independently in our setting, this is the optimal convergence rate without any additional assumptions (Vapnik 1998; Mendelson 2008).

For the PNPU-AUC and PNNU-AUC risks, we prove the following generalization error bounds (its proof is also available in Appendix 1):

[2] Our theoretical analysis can be easily extended to the loss satisfying $\ell_{0\text{-}1}(m) \le M\ell(m)$ with a certain $M > 0$.

[3] These losses are bounded in our setting, since the input to $\ell(m)$, i.e., f is bounded.

Theorem 2 *For any $\delta > 0$, the following inequalities hold separately with probability at least $1-\delta$ for all $f \in \mathcal{F}$:*

$$I(f) \le \widehat{R}^{\gamma}_{\mathrm{PNPU}}(f) + h(\delta/3)\left(\frac{1-\gamma}{\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{N}})}} + \frac{\gamma}{\theta_{\mathrm{N}}\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{U}})}} + \frac{\gamma\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}\sqrt{n_{\mathrm{P}}}}\right),$$
$$I(f) \le \widehat{R}^{\gamma}_{\mathrm{PNNU}}(f) + h(\delta/3)\left(\frac{1-\gamma}{\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{N}})}} + \frac{\gamma}{\theta_{\mathrm{P}}\sqrt{\min(n_{\mathrm{N}}, n_{\mathrm{U}})}} + \frac{\gamma\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}\sqrt{n_{\mathrm{N}}}}\right).$$

where $\widehat{R}^{\gamma}_{\mathrm{PNPU}}$ and $\widehat{R}^{\gamma}_{\mathrm{PNNU}}$ are unbiased empirical risk estimators corresponding to $R^{\gamma}_{\mathrm{PNPU}}$ and $R^{\gamma}_{\mathrm{PNNU}}$, respectively.

Theorem 2 guarantees that $I(f)$ can be bounded from above by the empirical risk, $\widehat{R}(f)$, plus the confidence terms of order

$$\mathcal{O}_p\left(\frac{1}{\sqrt{n_{\mathrm{P}}}} + \frac{1}{\sqrt{n_{\mathrm{N}}}} + \frac{1}{\sqrt{n_{\mathrm{U}}}}\right).$$

Again, since n_{P}, n_{N}, and n_{U} can increase independently in our setting, this is the optimal convergence rate without any additional assumptions.

4.2 Variance reduction

In the existing semi-supervised classification method based on PU learning, the variance of the empirical risk was proved to be smaller than the supervised counterpart under certain conditions (Sakai et al. 2017). Similarly, we here investigate if the proposed semi-supervised risk estimators have smaller variance than its supervised counterpart.

Let us introduce the following variances and covariances:[4]

$$\begin{aligned}
\sigma^2_{\mathrm{PN}}(f) &= \mathrm{Var}_{\mathrm{PN}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{N}}))\right],\\
\sigma^2_{\mathrm{PP}}(f) &= \mathrm{Var}_{\mathrm{P\bar{P}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))\right],\\
\sigma^2_{\mathrm{NN}}(f) &= \mathrm{Var}_{\mathrm{N\bar{N}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{N}}, \overline{\boldsymbol{x}}^{\mathrm{N}}))\right],\\
\tau_{\mathrm{PN,PP}}(f) &= \mathrm{Cov}_{\mathrm{PN,P\bar{P}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{N}})), \ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))\right],\\
\tau_{\mathrm{PN,NN}}(f) &= \mathrm{Cov}_{\mathrm{PN,N\bar{N}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{N}})), \ell(f(\boldsymbol{x}^{\mathrm{N}}, \overline{\boldsymbol{x}}^{\mathrm{N}}))\right],\\
\tau_{\mathrm{PU,PP}}(f) &= \mathrm{Cov}_{\mathrm{PU,P\bar{P}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}})), \ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))\right],\\
\tau_{\mathrm{NU,NN}}(f) &= \mathrm{Cov}_{\mathrm{NU,N\bar{N}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}})), \ell(f(\boldsymbol{x}^{\mathrm{N}}, \overline{\boldsymbol{x}}^{\mathrm{N}}))\right].
\end{aligned}$$

Then, we have the following theorem (its proof is available in Appendix 1):

Theorem 3 *Assume $n_{\mathrm{U}} \to \infty$. For any fixed f, the minimizers of the variance of the empirical PNPU-AUC and PNNU-AUC risks are respectively obtained by*

$$\gamma_{\mathrm{PNPU}} = \mathrm{argmin}_{\gamma}\ \mathrm{Var}\left[\widehat{R}^{\gamma}_{\mathrm{PNPU}}(f)\right] = \frac{\psi_{\mathrm{PN}} - \psi_{\mathrm{PP}}/2}{\psi_{\mathrm{PN}} + \psi_{\mathrm{PU}} - \psi_{\mathrm{PP}}}, \tag{7}$$
$$\gamma_{\mathrm{PNNU}} = \mathrm{argmin}_{\gamma}\ \mathrm{Var}\left[\widehat{R}^{\gamma}_{\mathrm{PNNU}}(f)\right] = \frac{\psi_{\mathrm{PN}} - \psi_{\mathrm{NN}}/2}{\psi_{\mathrm{PN}} + \psi_{\mathrm{NU}} - \psi_{\mathrm{NN}}}, \tag{8}$$

4 $\mathrm{Var}_{\mathrm{PN}}$, $\mathrm{Var}_{\mathrm{P\bar{P}}}$, and $\mathrm{Var}_{\mathrm{N\bar{N}}}$ are the variances over $p_{\mathrm{P}}(\boldsymbol{x}^{\mathrm{P}})p_{\mathrm{N}}(\boldsymbol{x}^{\mathrm{N}})$, $p_{\mathrm{P}}(\boldsymbol{x}^{\mathrm{P}})p_{\mathrm{P}}(\overline{\boldsymbol{x}}^{\mathrm{P}})$, and $p_{\mathrm{N}}(\boldsymbol{x}^{\mathrm{N}})p_{\mathrm{N}}(\overline{\boldsymbol{x}}^{\mathrm{N}})$, respectively. $\mathrm{Cov}_{\mathrm{PN,P\bar{P}}}$, $\mathrm{Cov}_{\mathrm{PN,N\bar{N}}}$, $\mathrm{Cov}_{\mathrm{PU,P\bar{P}}}$, and $\mathrm{Cov}_{\mathrm{NU,N\bar{N}}}$ are the covariances over $p_{\mathrm{P}}(\boldsymbol{x}^{\mathrm{P}})p_{\mathrm{N}}(\boldsymbol{x}^{\mathrm{N}})p_{\mathrm{P}}(\overline{\boldsymbol{x}}^{\mathrm{P}})$, $p_{\mathrm{P}}(\boldsymbol{x}^{\mathrm{P}})p_{\mathrm{N}}(\boldsymbol{x}^{\mathrm{N}})p_{\mathrm{N}}(\overline{\boldsymbol{x}}^{\mathrm{N}})$, $p_{\mathrm{P}}(\boldsymbol{x}^{\mathrm{P}})p(\boldsymbol{x}^{\mathrm{U}})p_{\mathrm{P}}(\overline{\boldsymbol{x}}^{\mathrm{P}})$, and $p_{\mathrm{N}}(\boldsymbol{x}^{\mathrm{N}})p(\boldsymbol{x}^{\mathrm{U}})p_{\mathrm{N}}(\overline{\boldsymbol{x}}^{\mathrm{N}})$, respectively.

where

$$\begin{aligned}
\psi_{\mathrm{PN}} &= \frac{1}{n_{\mathrm{P}}n_{\mathrm{N}}}\sigma^2_{\mathrm{PN}}(f),\\
\psi_{\mathrm{PU}} &= \frac{\theta_{\mathrm{P}}^2}{\theta_{\mathrm{N}}^2 n_{\mathrm{P}}{}^2}\sigma^2_{\mathrm{PP}}(f) - \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}^2 n_{\mathrm{P}}}\tau_{\mathrm{PU,PP}}(f),\\
\psi_{\mathrm{PP}} &= \frac{1}{\theta_{\mathrm{N}} n_{\mathrm{P}}}\tau_{\mathrm{PN,PU}}(f) - \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}} n_{\mathrm{P}}}\tau_{\mathrm{PN,PP}}(f),\\
\psi_{\mathrm{NU}} &= \frac{\theta_{\mathrm{N}}^2}{\theta_{\mathrm{P}}^2 n_{\mathrm{N}}{}^2}\sigma^2_{\mathrm{NN}}(f) - \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}^2 n_{\mathrm{N}}}\tau_{\mathrm{NU,NN}}(f),\\
\psi_{\mathrm{NN}} &= \frac{1}{\theta_{\mathrm{P}} n_{\mathrm{N}}}\tau_{\mathrm{PN,NU}}(f) - \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}} n_{\mathrm{N}}}\tau_{\mathrm{PN,NN}}(f).
\end{aligned}$$

Additionally, we have $\mathrm{Var}[\widehat{R}^{\gamma}_{\mathrm{PNPU}}(f)] < \mathrm{Var}[\widehat{R}_{\mathrm{PN}}(f)]$ *for any* $\gamma \in (0, 2\gamma_{\mathrm{PNPU}})$ *if* $\psi_{\mathrm{PN}} + \psi_{\mathrm{PU}} > \psi_{\mathrm{PP}}$ *and* $2\psi_{\mathrm{PN}} > \psi_{\mathrm{PP}}$. *Similarly, we have* $\mathrm{Var}[\widehat{R}^{\gamma}_{\mathrm{PNNU}}(f)] < \mathrm{Var}[\widehat{R}_{\mathrm{PN}}(f)]$ *for any* $\gamma \in (0, 2\gamma_{\mathrm{PNNU}})$ *if* $\psi_{\mathrm{PN}} + \psi_{\mathrm{NU}} > \psi_{\mathrm{NN}}$ *and* $2\psi_{\mathrm{PN}} > \psi_{\mathrm{NN}}$.

This theorem means that, if γ is chosen appropriately, our proposed risk estimators, $\widehat{R}^{\gamma}_{\mathrm{PNPU}}$ and $\widehat{R}^{\gamma}_{\mathrm{PNNU}}$, have smaller variance than the standard supervised risk estimator $\widehat{R}_{\mathrm{PN}}$. A practical consequence of Theorem 3 is that when we conduct cross-validation for hyperparameter selection, we may use our proposed risk estimators $\widehat{R}^{\gamma}_{\mathrm{PNPU}}$ and $\widehat{R}^{\gamma}_{\mathrm{PNNU}}$ instead of the standard supervised risk estimator $\widehat{R}_{\mathrm{PN}}$ since they are more stable (see Sect. 5.3 for details).

5 Practical implementation

In this section, we explain the implementation details of our proposed methods.

5.1 General case

In practice, the AUC risks R introduced above are replaced with their empirical version $\widehat{R}$, where the expectations in R are replaced with the corresponding sample averages.

Here, we focus on the linear-in-parameter model given by

$$g(\boldsymbol{x}) = \sum_{\ell=1}^{b} w_\ell \phi(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}),$$

where $^\top$ denotes the transpose of vectors and matrices, b is the number of basis functions, $\boldsymbol{w} = (w_1, \ldots, w_b)^\top$ is a parameter vector, and $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_b(\boldsymbol{x}))^\top$ is a basis function vector. The linear-in-parameter model allows us to express the composite classifier as

$$f(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{w}^\top \bar{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{x}'),$$

where

$$\bar{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{x}') := \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{x}')$$

is a composite basis function vector. We train the classifier by minimizing the ℓ_2-regularized empirical AUC risk:

$$\min_{\boldsymbol{w}} \widehat{R}(f) + \lambda\|\boldsymbol{w}\|^2,$$

where $\lambda \geq 0$ is the regularization parameter.

5.2 Analytical solution for squared loss

For the squared loss $\ell_S(m) := (1-m)^2$, the empirical PU-AUC risk[5] can be expressed as

$$\begin{aligned}\widehat{R}_{\mathrm{PU}}(f) &= \frac{1}{\theta_{\mathrm{N}} n_{\mathrm{P}} n_{\mathrm{U}}} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{k=1}^{n_{\mathrm{U}}} \ell_{\mathrm{S}}\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_k^{\mathrm{U}})\right) \\ &\quad - \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}} n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'=1}^{n_{\mathrm{P}}} \ell_{\mathrm{S}}\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_{i'}^{\mathrm{P}})\right) + \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}(n_{\mathrm{P}}-1)} \\ &= 1 - 2\boldsymbol{w}^\top \widehat{\boldsymbol{h}}_{\mathrm{PU}} + \boldsymbol{w}^\top \widehat{\boldsymbol{H}}_{\mathrm{PU}} \boldsymbol{w} - \boldsymbol{w}^\top \widehat{\boldsymbol{H}}_{\mathrm{PP}} \boldsymbol{w},\end{aligned}$$

where

$$\begin{aligned}\widehat{\boldsymbol{h}}_{\mathrm{PU}} &:= \frac{1}{\theta_{\mathrm{N}} n_{\mathrm{P}}} \boldsymbol{\Phi}_{\mathrm{P}}^\top \boldsymbol{1}_{n_{\mathrm{P}}} - \frac{1}{\theta_{\mathrm{N}} n_{\mathrm{U}}} \boldsymbol{\Phi}_{\mathrm{U}}^\top \boldsymbol{1}_{n_{\mathrm{U}}}, \\ \widehat{\boldsymbol{H}}_{\mathrm{PU}} &:= \frac{1}{\theta_{\mathrm{N}} n_{\mathrm{P}}} \boldsymbol{\Phi}_{\mathrm{P}}^\top \boldsymbol{\Phi}_{\mathrm{P}} - \frac{1}{\theta_{\mathrm{N}} n_{\mathrm{P}} n_{\mathrm{U}}} \boldsymbol{\Phi}_{\mathrm{U}}^\top \boldsymbol{1}_{n_{\mathrm{U}}} \boldsymbol{1}_{n_{\mathrm{P}}}^\top \boldsymbol{\Phi}_{\mathrm{P}} \\ &\quad - \frac{1}{\theta_{\mathrm{N}} n_{\mathrm{P}} n_{\mathrm{U}}} \boldsymbol{\Phi}_{\mathrm{P}}^\top \boldsymbol{1}_{n_{\mathrm{P}}} \boldsymbol{1}_{n_{\mathrm{U}}}^\top \boldsymbol{\Phi}_{\mathrm{U}} + \frac{1}{\theta_{\mathrm{N}} n_{\mathrm{U}}} \boldsymbol{\Phi}_{\mathrm{U}}^\top \boldsymbol{\Phi}_{\mathrm{U}}, \\ \widehat{\boldsymbol{H}}_{\mathrm{PP}} &:= \frac{2\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}(n_{\mathrm{P}}-1)} \boldsymbol{\Phi}_{\mathrm{P}}^\top \boldsymbol{\Phi}_{\mathrm{P}} - \frac{2\theta_{\mathrm{P}}}{\theta_{\mathrm{N}} n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \boldsymbol{\Phi}_{\mathrm{P}}^\top \boldsymbol{1}_{n_{\mathrm{P}}} \boldsymbol{1}_{n_{\mathrm{P}}}^\top \boldsymbol{\Phi}_{\mathrm{P}}, \\ \boldsymbol{\Phi}_{\mathrm{P}} &:= \left(\boldsymbol{\phi}(\boldsymbol{x}_1^{\mathrm{P}}), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_{n_{\mathrm{P}}}^{\mathrm{P}})\right)^\top, \\ \boldsymbol{\Phi}_{\mathrm{U}} &:= \left(\boldsymbol{\phi}(\boldsymbol{x}_1^{\mathrm{U}}), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_{n_{\mathrm{U}}}^{\mathrm{U}})\right)^\top,\end{aligned}$$

and $\boldsymbol{1}_b$ is the b-dimensional vector whose elements are all one. With the ℓ_2-regularizer, we can analytically obtain the solution by

$$\widehat{\boldsymbol{w}}_{\mathrm{PU}} := (\widehat{\boldsymbol{H}}_{\mathrm{PU}} - \widehat{\boldsymbol{H}}_{\mathrm{PP}} + \lambda \boldsymbol{I}_b)^{-1} \widehat{\boldsymbol{h}}_{\mathrm{PU}},$$

where $\boldsymbol{I}_b$ is the b-dimensional identity matrix.

The computational complexity of computing $\widehat{\boldsymbol{h}}_{\mathrm{PU}}$, $\widehat{\boldsymbol{H}}_{\mathrm{PU}}$, and $\widehat{\boldsymbol{H}}_{\mathrm{PP}}$ are $\mathcal{O}((n_{\mathrm{P}}+n_{\mathrm{U}})b)$, $\mathcal{O}((n_{\mathrm{P}}+n_{\mathrm{U}})b^2)$, and $\mathcal{O}(n_{\mathrm{P}}b^2)$, respectively. Then, solving a system of linear equations to obtain the solution $\widehat{\boldsymbol{w}}_{\mathrm{PU}}$ requires the computational complexity of $\mathcal{O}(b^3)$. In total, the computational complexity of this PU-AUC optimization method is $\mathcal{O}((n_{\mathrm{P}}+n_{\mathrm{U}})b^2 + b^3)$.

As given by Eq. (6), our PNU-AUC optimization method consists of the PNPU-AUC risk and the PNNU-AUC risk. For the squared loss $\ell_S(m) := (1-m)^2$, the empirical PNPU-AUC risk can be expressed as

$$\begin{aligned}\widehat{R}_{\mathrm{PNPU}}^{\gamma}(f) &= \frac{1-\gamma}{n_{\mathrm{P}} n_{\mathrm{N}}} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{j=1}^{n_{\mathrm{N}}} \ell_{\mathrm{S}}\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_j^{\mathrm{N}})\right) + \frac{\gamma}{\theta_{\mathrm{N}} n_{\mathrm{P}} n_{\mathrm{U}}} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{k=1}^{n_{\mathrm{U}}} \ell_{\mathrm{S}}\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_k^{\mathrm{U}})\right) \\ &\quad - \frac{\gamma\theta_{\mathrm{P}}}{\theta_{\mathrm{N}} n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'=1}^{n_{\mathrm{P}}} \ell_{\mathrm{S}}\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_{i'}^{\mathrm{P}})\right) + \frac{\gamma\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}(n_{\mathrm{P}}-1)} \\ &= (1-\gamma) - 2(1-\gamma)\boldsymbol{w}^\top \widehat{\boldsymbol{h}}_{\mathrm{PN}} + (1-\gamma)\boldsymbol{w}^\top \widehat{\boldsymbol{H}}_{\mathrm{PN}} \boldsymbol{w} \\ &\quad + \gamma - 2\gamma \boldsymbol{w}^\top \widehat{\boldsymbol{h}}_{\mathrm{PU}} + \gamma \boldsymbol{w}^\top \widehat{\boldsymbol{H}}_{\mathrm{PU}} \boldsymbol{w} - \gamma \boldsymbol{w}^\top \widehat{\boldsymbol{H}}_{\mathrm{PP}} \boldsymbol{w},\end{aligned}$$

[5] We discuss the way of estimating the PU-AUC risk in Appendix 1.

where

$$\begin{aligned}
\widehat{\boldsymbol{h}}_{\mathrm{PN}} &:= \frac{1}{n_{\mathrm{P}}}\boldsymbol{\Phi}_{\mathrm{P}}^{\top}\mathbf{1}_{n_{\mathrm{P}}} - \frac{1}{n_{\mathrm{N}}}\boldsymbol{\Phi}_{\mathrm{N}}^{\top}\mathbf{1}_{n_{\mathrm{N}}},\\
\widehat{\boldsymbol{H}}_{\mathrm{PN}} &:= \frac{1}{n_{\mathrm{P}}}\boldsymbol{\Phi}_{\mathrm{P}}^{\top}\boldsymbol{\Phi}_{\mathrm{P}} - \frac{1}{n_{\mathrm{P}}n_{\mathrm{N}}}\boldsymbol{\Phi}_{\mathrm{P}}^{\top}\mathbf{1}_{n_{\mathrm{P}}}\mathbf{1}_{n_{\mathrm{N}}}^{\top}\boldsymbol{\Phi}_{\mathrm{N}}\\
&\quad - \frac{1}{n_{\mathrm{P}}n_{\mathrm{N}}}\boldsymbol{\Phi}_{\mathrm{N}}^{\top}\mathbf{1}_{n_{\mathrm{N}}}\mathbf{1}_{n_{\mathrm{P}}}^{\top}\boldsymbol{\Phi}_{\mathrm{P}} + \frac{1}{n_{\mathrm{N}}}\boldsymbol{\Phi}_{\mathrm{N}}^{\top}\boldsymbol{\Phi}_{\mathrm{N}},\\
\boldsymbol{\Phi}_{\mathrm{N}} &:= \left(\boldsymbol{\phi}(\boldsymbol{x}_1^{\mathrm{N}}), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_{n_{\mathrm{N}}}^{\mathrm{N}})\right)^{\top}.
\end{aligned}$$

The solution for the ℓ_2-regularized PNPU-AUC optimization can be analytically obtained by

$$\widehat{\boldsymbol{w}}_{\mathrm{PNPU}}^{\gamma} := \Big((1-\gamma)\widehat{\boldsymbol{H}}_{\mathrm{PN}} + \gamma\widehat{\boldsymbol{H}}_{\mathrm{PU}} - \gamma\widehat{\boldsymbol{H}}_{\mathrm{PP}} + \lambda\boldsymbol{I}_b\Big)^{-1}\Big((1-\gamma)\widehat{\boldsymbol{h}}_{\mathrm{PN}} + \gamma\widehat{\boldsymbol{h}}_{\mathrm{PU}}\Big).$$

Similarly, the solution for the ℓ_2-regularized PNNU-AUC optimization can be obtained by

$$\widehat{\boldsymbol{w}}_{\mathrm{PNNU}}^{\gamma} := \Big((1-\gamma)\widehat{\boldsymbol{H}}_{\mathrm{PN}} + \gamma\widehat{\boldsymbol{H}}_{\mathrm{NU}} - \gamma\widehat{\boldsymbol{H}}_{\mathrm{NN}} + \lambda\boldsymbol{I}_b\Big)^{-1}\Big((1-\gamma)\widehat{\boldsymbol{h}}_{\mathrm{PN}} + \gamma\widehat{\boldsymbol{h}}_{\mathrm{NU}}\Big).$$

where

$$\begin{aligned}
\widehat{\boldsymbol{h}}_{\mathrm{NU}} &:= \frac{1}{\theta_{\mathrm{P}}n_{\mathrm{U}}}\boldsymbol{\Phi}_{\mathrm{U}}^{\top}\mathbf{1}_{n_{\mathrm{U}}} - \frac{1}{\theta_{\mathrm{P}}n_{\mathrm{N}}}\boldsymbol{\Phi}_{\mathrm{N}}^{\top}\mathbf{1}_{n_{\mathrm{N}}},\\
\widehat{\boldsymbol{H}}_{\mathrm{NU}} &:= \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}n_{\mathrm{N}}}\boldsymbol{\Phi}_{\mathrm{N}}^{\top}\boldsymbol{\Phi}_{\mathrm{N}} - \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}n_{\mathrm{N}}n_{\mathrm{U}}}\boldsymbol{\Phi}_{\mathrm{U}}^{\top}\mathbf{1}_{n_{\mathrm{U}}}\mathbf{1}_{n_{\mathrm{N}}}^{\top}\boldsymbol{\Phi}_{\mathrm{N}}\\
&\quad - \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}n_{\mathrm{N}}n_{\mathrm{U}}}\boldsymbol{\Phi}_{\mathrm{N}}^{\top}\mathbf{1}_{n_{\mathrm{N}}}\mathbf{1}_{n_{\mathrm{U}}}^{\top}\boldsymbol{\Phi}_{\mathrm{U}} + \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}n_{\mathrm{U}}}\boldsymbol{\Phi}_{\mathrm{U}}^{\top}\boldsymbol{\Phi}_{\mathrm{U}},\\
\widehat{\boldsymbol{H}}_{\mathrm{NN}} &:= \frac{2\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}(n_{\mathrm{N}}-1)}\boldsymbol{\Phi}_{\mathrm{N}}^{\top}\boldsymbol{\Phi}_{\mathrm{N}} - \frac{2\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}n_{\mathrm{N}}(n_{\mathrm{N}}-1)}\boldsymbol{\Phi}_{\mathrm{N}}^{\top}\mathbf{1}_{n_{\mathrm{N}}}\mathbf{1}_{n_{\mathrm{N}}}^{\top}\boldsymbol{\Phi}_{\mathrm{N}}.
\end{aligned}$$

The computational complexity of computing $\widehat{\boldsymbol{h}}_{\mathrm{PN}}$ and $\widehat{\boldsymbol{H}}_{\mathrm{PN}}$ are $\mathcal{O}((n_{\mathrm{P}}+n_{\mathrm{N}})b)$ and $\mathcal{O}((n_{\mathrm{P}}+n_{\mathrm{N}})b^2)$, respectively. Then, obtaining the solution $\widehat{\boldsymbol{w}}_{\mathrm{PNPU}}$ ($\widehat{\boldsymbol{w}}_{\mathrm{PNNU}}$) requires the computational complexity of $\mathcal{O}(b^3)$. Including the computational complexity of computing $\widehat{\boldsymbol{h}}_{\mathrm{PU}}$, $\widehat{\boldsymbol{H}}_{\mathrm{PU}}$, and $\widehat{\boldsymbol{H}}_{\mathrm{PP}}$, the total computational complexity of the PNPU-AUC optimization method is $\mathcal{O}((n_{\mathrm{P}}+n_{\mathrm{N}}+n_{\mathrm{U}})b^2+b^3)$. Similarly, the total computational complexity of the PNNU-AUC optimization method is $\mathcal{O}((n_{\mathrm{P}}+n_{\mathrm{N}}+n_{\mathrm{U}})b^2+b^3)$. Thus, the computational complexity of the PNU-AUC optimization method is $\mathcal{O}((n_{\mathrm{P}}+n_{\mathrm{N}}+n_{\mathrm{U}})b^2+b^3)$.

From the viewpoint of computational complexity, the squared loss and the exponential loss are more efficient than the logistic loss because these loss functions reduce the nested summations to individual ones. More specifically, for example, in the PU-AUC optimization method, the logistic loss requires $\mathcal{O}(n_{\mathrm{P}}n_{\mathrm{U}})$ operations for evaluating the first term in the PU-AUC risk, i.e., the loss over positive and unlabeled samples. In contrast, the squared loss and exponential loss reduce the number of operations for loss evaluation to $\mathcal{O}(n_{\mathrm{P}} + n_{\mathrm{U}})$.[6] This property is beneficial especially when we handle large scale data sets.

5.3 Cross-validation

To tune the hyperparameters such as the regularization parameter λ, we use the cross-validation.

For the PU-AUC optimization method, we use the PU-AUC risk in Eq. (4) with the zero-one loss as the cross-validation score.

For the PNU-AUC optimization method, we use the PNU-AUC risk in Eq. (6) with the zero-one loss as the score. To this end, however, we need to fix the combination parameter η in the cross-validation score in advance and then, we tune the hyperparameters including the combination parameter. More specifically, let $\overline{\eta} \in [-1, 1]$ be the predefined combination parameter. We conduct cross-validation with respect to $R^{\overline{\eta}}_{\mathrm{PNU}}(f)$ for tuning the hyperparameters. Since the PNU-AUC risk is equivalent to the PN-AUC risk for any $\overline{\eta}$, we can choose any $\overline{\eta}$ in principle. However, when the empirical PNU-AUC risk is used in practice, choice of $\overline{\eta}$ may affect the performance of cross-validation.

Here, based on the theoretical result of variance reduction given in Sect. 4.2, we give a practical method to determine $\overline{\eta}$. Assuming the covariances, e.g., $\tau_{\mathrm{PN,PP}}(f)$, are small enough to be neglected and $\sigma_{\mathrm{PN}}(f) = \sigma_{\mathrm{PP}}(f) = \sigma_{\mathrm{NN}}(f)$, we can obtain a simpler form of Eqs. (7) and (8) as

$$\overline{\gamma}_{\mathrm{PNPU}} = \frac{1}{1 + \theta_{\mathrm{P}}^2 n_{\mathrm{N}} / \left(\theta_{\mathrm{N}}^2 n_{\mathrm{P}}\right)},$$
$$\overline{\gamma}_{\mathrm{PNNU}} = \frac{1}{1 + \theta_{\mathrm{N}}^2 n_{\mathrm{P}} / \left(\theta_{\mathrm{P}}^2 n_{\mathrm{N}}\right)}.$$

They can be computed simply from the number of samples and the (estimated) class-prior. Finally, to select the combination parameter η, we use $\widehat{R}^{\overline{\gamma}_{\mathrm{PNPU}}}_{\mathrm{PNPU}}$ for $\eta \geq 0$, and $\widehat{R}^{\overline{\gamma}_{\mathrm{PNNU}}}_{\mathrm{PNNU}}$ for $\eta < 0$.

6 Experiments

In this section, we numerically investigate the behavior of the proposed methods and evaluate their performance on various data sets. All experiments were carried out using a PC equipped with two 2.60 GHz Intel® Xeon® E5-2640 v3 CPUs.

[6] For example, the exponential loss over positive and unlabeled data can be computed as follows:

$$\begin{aligned}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{k=1}^{n_{\mathrm{U}}} \ell_{\mathrm{E}}\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_k^{\mathrm{U}})\right) &= \sum_{i=1}^{n_{\mathrm{P}}}\sum_{k=1}^{n_{\mathrm{U}}} \exp(-g(\boldsymbol{x}_i^{\mathrm{P}}) + g(\boldsymbol{x}_k^{\mathrm{U}})) \\ &= \sum_{i=1}^{n_{\mathrm{P}}} \exp\left(-g(\boldsymbol{x}_i^{\mathrm{P}})\right) \sum_{k=1}^{n_{\mathrm{U}}} \exp\left(g(\boldsymbol{x}_k^{\mathrm{U}})\right).\end{aligned}$$

Thus, the number of operations for loss evaluation is reduced to $n_{\mathrm{P}} + n_{\mathrm{U}} + 1$ rather than $n_{\mathrm{P}} n_{\mathrm{U}}$.

As the classifier, we used the linear-in-parameter model. In all experiments except text classification tasks, we used the Gaussian kernel basis function expressed as

$$\phi_\ell(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_\ell\|^2}{2\sigma^2}\right),$$

where $\sigma > 0$ is the Gaussian bandwidth, $\{\boldsymbol{x}_\ell\}_{\ell=1}^b$ are the samples randomly selected from training samples $\{\boldsymbol{x}_i\}_{i=1}^n$ and n is the number of training samples. In text classification tasks, we used the linear kernel basis function:

$$\phi_\ell(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{x}_\ell.$$

The number of basis functions was set at $b = \min(n, 200)$. The candidates of the Gaussian bandwidth were $\mathrm{median}(\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\}_{i,j=1}^n) \times \{1/8, 1/4, 1/2, 1, 2\}$ and that of the regularization parameter were $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1\}$. All hyper-parameters were determined by five-fold cross-validation. As the loss function, we used the squared loss function $\ell_{\mathrm{S}}(m) = (1-m)^2$.

6.1 Effect of variance reduction

First, we numerically confirm the effect of variance reduction. We compare the variance of the empirical PNU-AUC risk against the variance of the empirical PN-AUC risk, $\mathrm{Var}[\widehat{R}^\eta_{\mathrm{PNU}}(f)]$ vs. $\mathrm{Var}[\widehat{R}_{\mathrm{PN}}(f)]$, under a fixed classifier f.

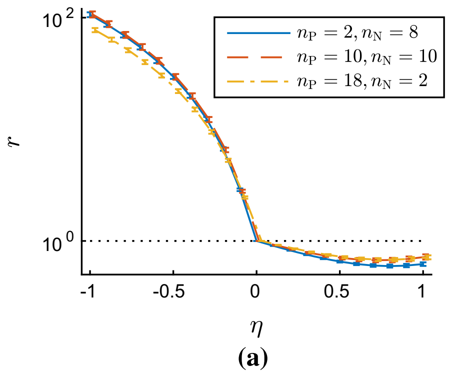
As the fixed classifier, we used the minimizer of the empirical PN-AUC risk, denoted by $\widehat{f}_{\mathrm{PN}}$. The number of positive and negative samples for training varied as $(n_{\mathrm{P}}, n_{\mathrm{N}}) = (2, 8)$, $(10, 10)$, and $(18, 2)$. We then computed the variance of the empirical PN-AUC and PNU-AUC risks with additional 10 positive, 10 negative, and 300 unlabeled samples. As the data set, we used the Banana data set (Rätsch et al. 2001). In this experiment, the class-prior was set at $\theta_{\mathrm{P}} = 0.1$ and assumed to be known.

Figure 1 plots the value of the variance of the empirical PNU-AUC risk divided by that of the PN-AUC risk,

$$r := \frac{\mathrm{Var}\left[\widehat{R}^\eta_{\mathrm{PNU}}(\widehat{f}_{\mathrm{PN}})\right]}{\mathrm{Var}\left[\widehat{R}_{\mathrm{PN}}(\widehat{f}_{\mathrm{PN}})\right]},$$

as a function of the combination parameter η under different numbers of positive and negative samples. The results show that $r < 1$ can be achieved by an appropriate choice of η, meaning that the variance of the empirical PNU-AUC risk can be smaller than that of the PN-AUC risk.

We then investigate how the class-prior affects the variance reduction. In this experiment, the number of positive and negative samples for $\widehat{f}_{\mathrm{PN}}$ are $n_{\mathrm{P}} = 10$ and $n_{\mathrm{N}} = 10$, respectively. Figure 2 showed the values of r as a function of the combination parameter η under different class-priors. When the class-prior, θ_{P}, is 0.1 and 0.2, the variance can be reduced for $\eta > 0$. When the class-prior is 0.3, the range of the value of η that yields variance reduction becomes smaller. However, this may not be that problematic in practice, because AUC optimization is effective when two classes are highly imbalanced, i.e., the class-prior is far from 0.5; when the class-prior is close to 0.5, we may simply use the standard misclassification rate minimization approach.



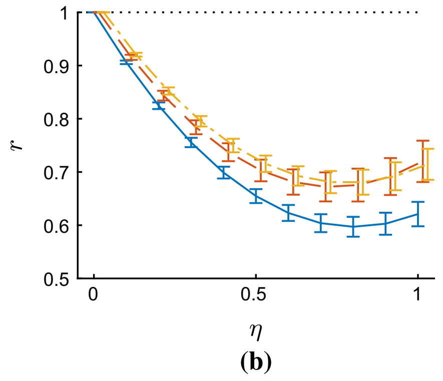
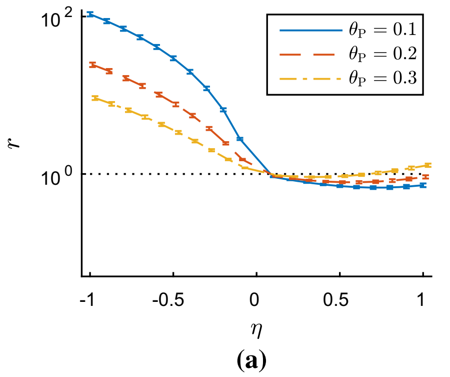


Fig. 1 Average with standard error of the ratio between the variance of the empirical PNU risk and that of the PN risk, $r = \mathrm{Var}[\widehat{R}^{\eta}_{\mathrm{PNU}}(\widehat{f}_{\mathrm{PN}})]/\,\mathrm{Var}[\widehat{R}_{\mathrm{PN}}(\widehat{f}_{\mathrm{PN}})]$, as a function of the combination parameter η over 100 trials on the Banana data set. The class-prior is $\theta_\mathrm{P} = 0.1$ and the number of positive and negative samples varies as $(n_\mathrm{P}, n_\mathrm{N}) = (2, 8)$, $(10, 10)$, and $(18, 2)$. Left: values of r as a function of η. Right: values for $\eta > 0$ are magnified. **a** $-1 \le \eta \le 1$. **b** $\eta > 0$



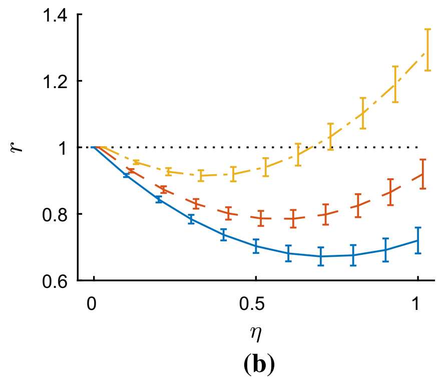


Fig. 2 Average with standard error of the ratio between the variance of the PNU-AUC risk and that of the PN-AUC risk, $r = \mathrm{Var}[\widehat{R}^{\eta}_{\mathrm{PNU}}(\widehat{f}_{\mathrm{PN}})]/\,\mathrm{Var}[\widehat{R}_{\mathrm{PN}}(\widehat{f}_{\mathrm{PN}})]$, as a function of the combination parameter η over 100 trials on the Banana data set. A class-prior varies as $\theta_\mathrm{P} = 0.1, 0.2$, and 0.3. Left: values of r as a function of η. Right: values for $\eta > 0$ are magnified. When $\theta_\mathrm{P} = 0.1, 0.2$, the variance of the empirical PNU-AUC risk is smaller than that of the PN risk for $\eta > 0$. **a** $-1 \le \eta \le 1$. **b** $\eta > 0$

6.2 Benchmark data sets

Next, we report the classification performance of the proposed PU-AUC and PNU-AUC optimization methods, respectively. We used 15 benchmark data sets from the *IDA Benchmark Repository* (Rätsch et al. 2001), the *Semi-Supervised Learning Book* (Chapelle et al. 2006), the *LIBSVM* (Chang et al. 2011), and the *UCI Machine Learning Repository* (Lichman 2013). The detailed statistics of the data sets are summarized in Appendix 1.

6.2.1 AUC optimization from positive and unlabeled data

We compared the proposed PU-AUC optimization method against the existing AUC optimization method based on the ranking SVM (PU-RSVM) (Sundararajan et al. 2011). We

Table 1 Average and standard error of the estimated class-prior over 50 trials on benchmark data sets in PU learning setting

Data set	d	$\theta_P = 0.1$	$\theta_P = 0.2$
Banana	2	0.15 (0.01)	0.24 (0.01)
skin_nonskin	3	0.10 (0.00)	0.20 (0.01)
cod-rna	8	0.32 (0.01)	0.40 (0.02)
Magic	10	0.34 (0.01)	0.39 (0.01)
Image	18	0.13 (0.01)	0.24 (0.01)
Twonorm	20	0.27 (0.00)	0.34 (0.00)
Waveform	21	0.31 (0.00)	0.38 (0.01)
mushrooms	112	0.19 (0.00)	0.28 (0.00)

Table 2 Average and standard error of the AUC over 50 trials on benchmark data sets

Data set	d	$\theta_P = 0.1$		$\theta_P = 0.2$	
		PU-AUC	PU-RSVM	PU-AUC	PU-RSVM
Banana	2	**95.4 (0.1)**	88.5 (0.2)	**95.0 (0.1)**	85.9 (0.2)
skin_nonskin	3	**99.9 (0.0)**	86.2 (0.3)	**99.8 (0.0)**	73.8 (0.5)
cod-rna	8	**98.1 (0.1)**	62.8 (0.3)	**97.6 (0.1)**	59.4 (0.4)
Magic	10	**87.4 (0.2)**	82.8 (0.2)	**86.4 (0.2)**	81.9 (0.1)
Image	18	**97.5 (0.1)**	82.2 (0.2)	**96.7 (0.2)**	77.0 (0.4)
Twonorm	20	**99.6 (0.0)**	85.5 (0.2)	**99.5 (0.0)**	79.2 (0.4)
Waveform	21	**96.6 (0.1)**	73.9 (0.6)	**96.3 (0.1)**	59.5 (1.0)
mushrooms	112	**99.8 (0.0)**	93.9 (0.3)	**99.6 (0.1)**	84.7 (0.3)
#Best/Comp.		8	0	8	0

The boldface denotes the best and comparable methods in terms of the average AUC according to the t test at the significance level 5%. The last row shows the number of best/comparable cases of each method

trained a classifier with samples of size $n_P = 100$ and $n_U = 1000$ under the different class-priors $\theta_P = 0.1$ and 0.2. For the PU-AUC optimization method, the squared loss function was used and the class-prior was estimated by the distribution matching method (du Plessis et al. 2017). The results of the estimated class-prior are summarized in Table 1.

Table 2 lists the average with standard error of the AUC over 50 trials, showing that the proposed PU-AUC optimization method achieves better performance than the existing method. In particular, when $\theta_P = 0.2$, the difference between PU-RSVM and our method becomes larger compared with the difference when $\theta_P = 0.1$. Since PU-RSVM can be interpreted as regarding unlabeled data as negative, the bias caused by this becomes larger when $\theta_P = 0.2$.

Figure 3 summarizes the average computation time over 50 trials. The computation time of the PU-AUC optimization method includes both the class-prior estimation and the empirical risk minimization. The results show that the PU-AUC optimization method requires almost twice computation time as that of PU-RSVM, but it would be acceptable in practice to obtain better performance.

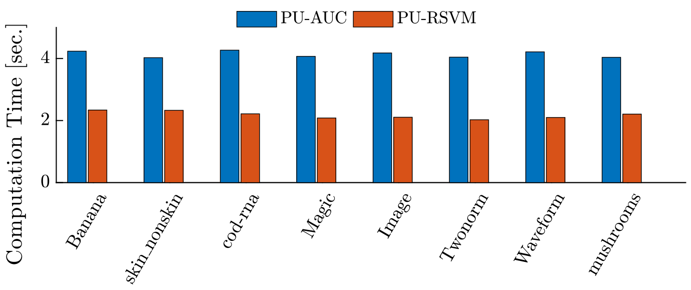


Fig. 3 Average computation time on benchmark data sets when $\theta_P = 0.1$ over 50 trials. The computation time of the PU-AUC optimization method includes the class-prior estimation and the empirical risk minimization

6.2.2 Semi-supervised AUC optimization

Here, we compare the proposed PNU-AUC optimization method against existing AUC optimization approaches: the semi-supervised rankboost (SSRankboost) (Amini et al. 2008),[7] the semi-supervised AUC-optimized logistic sigmoid (sAUC-LS) (Fujino and Ueda 2016),[8] and the optimum AUC with a generative model (OptAG) (Fujino and Ueda 2016).

We trained the classifier with samples of size $n_P = \theta_P \cdot n_L$, $n_N = n_L - n_P$, and $n_U = 1000$, where n_L is the number of labeled samples. For the PNU-AUC optimization method, the squared loss function was used and the candidates of the combination parameter η were $\{-0.9, -0.8, \ldots, 0.9\}$. For the class-prior estimation, we used the energy distance minimization method (Kawakubo et al. 2016). The results of the estimated class-prior are summarized in Table 3.

For SSRankboost, the discount factor and the number of neighbors were chosen from $\{10^{-3}, 10^{-2}, 10^{-1}\}$ and $\{2, 3, \ldots, 7\}$, respectively. For sAUC-LS and OptAG, the regularization parameter for the entropy regularizer was chosen from $\{1, 10\}$. Furthermore, as the generative model of OptAG, we adapted the Gaussian distribution for the data distribution and the Gaussian and Gamma distributions for the prior of the data distribution.[9]

Table 4 lists the average with standard error of the AUC over 50 trials, showing that the proposed PNU-AUC optimization method achieves better performance than or comparable performance to the existing methods on many data sets. Figure 4 summarizes the average computation time over 50 trials. The computation time of the PNU-AUC optimization method includes both the class-prior estimation and the empirical risk minimization. The results show that even though our proposed method involves the class-prior estimation, the computation time is relatively faster than SSRankboost and much faster than sAUC-LS and OptAG. The reason for longer computation time of sAUC-LS and OptAG is that their implementation is

[7] We used the code available at http://ama.liglab.fr/~amini/SSRankBoost/.

[8] This method is equivalent to OptAG without a generative model, which only employs a discriminative model with the entropy minimization principle. To eliminate the adverse effect of the wrongly chosen generative model, we added this method for comparison.

[9] As the generative model, we used the Gaussian distributions for positive and negative classes:

$$p_g\left(\boldsymbol{x}^{\mathrm{P}}; \boldsymbol{\mu}_{\mathrm{P}}\right) \propto \tau_{\mathrm{P}}^{\frac{d}{2}} \exp\left(-\frac{\tau_{\mathrm{P}}}{2}\|\boldsymbol{x}^{\mathrm{P}} - \boldsymbol{\mu}_{\mathrm{P}}\|^2\right),$$

$$p_g\left(\boldsymbol{x}^{\mathrm{P}}; \boldsymbol{\mu}_{\mathrm{N}}\right) \propto \tau_{\mathrm{N}}^{\frac{d}{2}} \exp\left(-\frac{\tau_{\mathrm{N}}}{2}\|\boldsymbol{x}^{\mathrm{N}} - \boldsymbol{\mu}_{\mathrm{N}}\|^2\right),$$

Table 3 Average and standard error of the estimated class-prior over 50 trials on benchmark data sets in semi-supervised learning setting

Data set	n_{L}	$\theta_{\mathrm{P}} = 0.1$	$\theta_{\mathrm{P}} = 0.2$
Banana	50	0.12 (0.01)	0.21 (0.02)
($d = 2$)	100	0.10 (0.01)	0.20 (0.01)
skin_nonskin	50	0.11 (0.01)	0.21 (0.01)
($d = 3$)	100	0.10 (0.01)	0.20 (0.01)
cod-rna	50	0.12 (0.01)	0.22 (0.01)
($d = 8$)	100	0.12 (0.01)	0.21 (0.01)
Magic	50	0.09 (0.01)	0.17 (0.01)
($d = 10$)	100	0.07 (0.01)	0.20 (0.01)
Image	50	0.12 (0.01)	0.22 (0.01)
($d = 18$)	100	0.11 (0.01)	0.20 (0.01)
SUSY	50	0.10 (0.01)	0.20 (0.01)
($d = 18$)	100	0.10 (0.01)	0.19 (0.01)
Ringnorm	50	0.06 (0.00)	0.15 (0.00)
($d = 20$)	100	0.07 (0.00)	0.17 (0.00)
Twonorm	50	0.10 (0.00)	0.20 (0.00)
($d = 20$)	100	0.10 (0.00)	0.20 (0.00)
Waveform	50	0.11 (0.01)	0.20 (0.01)
($d = 21$)	100	0.09 (0.01)	0.19 (0.01)
covtype	50	0.09 (0.01)	0.20 (0.01)
($d = 54$)	100	0.09 (0.01)	0.18 (0.01)
phishing	50	0.10 (0.00)	0.20 (0.00)
($d = 68$)	100	0.10 (0.00)	0.20 (0.00)
a9a	50	0.10 (0.01)	0.20 (0.01)
($d = 83$)	100	0.10 (0.00)	0.21 (0.01)
mushrooms	50	0.10 (0.00)	0.20 (0.00)
($d = 112$)	100	0.10 (0.00)	0.20 (0.00)
USPS	50	0.10 (0.00)	0.20 (0.01)
($d = 241$)	100	0.09 (0.01)	0.19 (0.01)
w8a	50	0.10 (0.00)	0.19 (0.00)
($d = 300$)	100	0.09 (0.00)	0.20 (0.01)

Footnote 9 continued
where τ_{P} and τ_{N} denote the precisions and $\boldsymbol{\mu}_{\mathrm{P}}$ and $\boldsymbol{\mu}_{\mathrm{N}}$ are the means. As the prior of $\boldsymbol{\mu}_{\mathrm{P}}$, $\boldsymbol{\mu}_{\mathrm{N}}$, τ_{P}, and τ_{N}, we used the Gaussian and Gamma distributions:

$$p\left(\boldsymbol{\mu}_{\mathrm{P}}; \boldsymbol{\mu}_{\mathrm{P}}^0\right) \propto \tau_{\mathrm{P}}^{\frac{d}{2}} \exp\left(-\frac{\rho_{\mathrm{P}}^0 \tau_{\mathrm{P}}}{2} \|\boldsymbol{\mu}_{\mathrm{P}} - \boldsymbol{\mu}_{\mathrm{P}}^0\|^2\right),$$

$$p\left(\boldsymbol{\mu}_{\mathrm{N}}; \boldsymbol{\mu}_{\mathrm{N}}^0\right) \propto \tau_{\mathrm{N}}^{\frac{d}{2}} \exp\left(-\frac{\rho_{\mathrm{N}}^0 \tau_{\mathrm{N}}}{2} \|\boldsymbol{\mu}_{\mathrm{N}} - \boldsymbol{\mu}_{\mathrm{N}}^0\|^2\right),$$

$$p\left(\tau_{\mathrm{P}}; a_{\mathrm{P}}^0, b_{\mathrm{P}}^0\right) \propto \tau_{\mathrm{P}}^{a_{\mathrm{P}}^0 - 1} \exp\left(-b_{\mathrm{P}}^0 \tau_{\mathrm{P}}\right),$$

$$p\left(\tau_{\mathrm{N}}; a_{\mathrm{N}}^0, b_{\mathrm{N}}^0\right) \propto \tau_{\mathrm{N}}^{a_{\mathrm{N}}^0 - 1} \exp\left(-b_{\mathrm{N}}^0 \tau_{\mathrm{N}}\right),$$

Table 4 Average and standard error of the AUC over 50 trials on benchmark data sets

Data set	n_L	$\theta_P = 0.1$				$\theta_P = 0.2$			
		PNU-AUC	SSRboost	sAUC-LS	OptAG	PNU-AUC	SSRboost	sAUC-LS	OptAG
Banana	50	**84.6 (1.3)**	61.6(1.0)	**82.0(1.3)**	**84.0 (1.3)**	**86.7 (0.9)**	66.1 (0.9)	83.3 (1.5)	**86.9 (0.7)**
($d = 2$)	100	**88.4 (0.7)**	66.2 (0.7)	84.0 (1.2)	**88.7 (0.5)**	**91.2 (0.4)**	69.1 (0.5)	88.5 (0.6)	89.9 (0.6)
skin_nonskin	50	**96.5 (0.9)**	92.2 (0.8)	96.4 (0.6)	**97.8 (0.6)**	**98.1 (0.6)**	93.7 (0.6)	**98.8 (0.2)**	**99.1 (0.2)**
($d = 3$)	100	98.8 (0.3)	94.5 (0.4)	99.0 (0.1)	**99.5 (0.0)**	**99.1 (0.3)**	95.6 (0.2)	**99.2 (0.3)**	**99.3 (0.2)**
cod-rna	50	**86.7 (1.4)**	79.4 (1.0)	61.6 (1.5)	63.3 (1.4)	**91.2 (1.0)**	86.8 (0.6)	66.0 (1.5)	68.7 (1.3)
($d = 8$)	100	**93.4 (0.6)**	88.7 (0.5)	67.3 (1.5)	67.7 (1.3)	**95.6 (0.5)**	91.7 (0.3)	74.1 (1.5)	75.9 (0.9)
Magic	50	**77.0 (1.1)**	74.8 (0.8)	73.7 (1.6)	**77.4 (0.7)**	**77.2 (1.3)**	**78.2 (0.5)**	76.5 (0.7)	75.4 (1.1)
($d = 10$)	100	**79.5 (0.5)**	**79.3 (0.6)**	74.5 (1.6)	77.1 (1.0)	**81.5 (0.4)**	**81.2 (0.4)**	75.8 (1.0)	78.3 (0.5)
Image	50	**81.6 (1.7)**	70.0 (1.1)	76.5 (1.6)	**80.7 (1.6)**	**86.1 (0.8)**	78.8 (0.6)	81.5 (1.0)	82.9 (1.1)
($d = 18$)	100	**88.3 (0.6)**	80.9 (0.7)	83.4 (1.2)	85.2 (1.0)	**92.1 (0.4)**	87.1 (0.5)	86.5 (0.6)	87.8 (0.4)
SUSY	50	63.0 (1.1)	**67.7 (1.3)**	56.2 (0.9)	56.2 (1.0)	64.7 (0.8)	**70.9 (0.8)**	55.6 (0.9)	57.7 (0.7)
($d = 18$)	100	65.4 (1.1)	**73.4 (0.5)**	59.0 (0.9)	59.0 (0.8)	69.4 (1.0)	**76.2 (0.4)**	58.9 (0.7)	58.3 (0.6)
Ringnorm	50	**98.3 (0.4)**	79.9 (0.7)	**98.7 (0.5)**	**99.0 (0.6)**	98.4 (0.4)	86.6 (0.4)	**99.5 (0.3)**	**99.1 (0.4)**
($d = 20$)	100	98.8 (0.3)	88.0 (0.4)	**99.8 (0.0)**	**99.8 (0.0)**	**99.4 (0.2)**	90.8 (0.3)	**99.5 (0.4)**	**99.6 (0.2)**
Twonorm	50	**96.9 (0.6)**	90.3 (0.4)	94.4 (1.1)	**96.1 (0.3)**	**97.5 (0.5)**	93.2 (0.3)	**97.7 (0.2)**	**97.4 (0.2)**
($d = 20$)	100	**98.6 (0.1)**	94.7 (0.2)	96.6 (0.2)	96.5 (0.2)	**99.0 (0.1)**	96.8 (0.1)	98.3 (0.1)	98.0 (0.2)
Waveform	50	**86.7 (1.3)**	**88.9 (0.4)**	85.5 (1.5)	85.8 (0.8)	**92.0 (0.6)**	**91.3 (0.2)**	88.6 (0.8)	89.4 (0.6)
($d = 21$)	100	**92.8 (0.4)**	91.8 (0.2)	87.8 (1.1)	87.7 (1.2)	**94.7 (0.2)**	93.1 (0.1)	90.1 (0.4)	89.7 (0.5)
covtype	50	57.8 (1.3)	**63.1 (1.0)**	55.7 (0.9)	58.9 (1.1)	60.3 (1.0)	**65.6 (0.8)**	56.4 (0.9)	57.8 (0.9)
($d = 54$)	100	60.7 (1.1)	**66.7 (0.8)**	59.1 (0.9)	60.3 (0.9)	64.2 (0.6)	**70.6 (0.6)**	57.7 (0.9)	60.6 (0.7)
phishing	50	89.8 (1.1)	**91.9 (0.6)**	71.1 (1.6)	74.0 (0.9)	91.8 (0.7)	**94.2 (0.3)**	74.7 (1.5)	77.1 (1.2)

Table 4 continued

Data set	n_L	$\theta_P = 0.1$				$\theta_P = 0.2$			
		PNU-AUC	SSRboost	sAUC-LS	OptAG	PNU-AUC	SSRboost	sAUC-LS	OptAG
($d = 68$)	100	**94.6 (0.2)**	**94.8 (0.2)**	76.3 (1.3)	76.9 (1.2)	94.9 (0.4)	**96.3 (0.1)**	80.5 (1.3)	82.0 (1.0)
a9a	50	69.4 (1.7)	**75.1 (0.9)**	**75.3 (1.4)**	**73.8 (1.4)**	**78.2 (1.0)**	**79.3 (0.5)**	**78.3 (1.1)**	**79.2 (0.7)**
($d = 83$)	100	77.9 (1.1)	**80.1 (0.5)**	**80.3 (0.5)**	**79.5 (0.8)**	82.0 (0.5)	**83.2 (0.3)**	81.5 (0.5)	81.5 (0.7)
mushrooms	50	96.0 (0.7)	96.0 (0.6)	96.6 (0.5)	**97.8 (0.3)**	96.4 (0.7)	**98.0 (0.2)**	**97.2 (0.5)**	**97.6 (0.8)**
($d = 112$)	100	**98.1 (0.5)**	98.2 (0.1)	97.9 (0.3)	**98.9 (0.2)**	**98.7 (0.2)**	**98.7 (0.1)**	98.2 (0.4)	**99.1 (0.4)**
USPS	50	**85.6 (1.0)**	73.9 (0.9)	79.5 (1.4)	82.7 (0.8)	**88.2 (0.7)**	80.5 (0.7)	81.2 (0.9)	85.1 (1.2)
($d = 241$)	100	**90.3 (0.5)**	79.4 (0.7)	84.2 (1.0)	86.0 (1.0)	**93.7 (0.5)**	85.2 (0.5)	83.3 (0.9)	89.7 (0.3)
w8a	50	**69.6 (1.1)**	**70.9 (0.9)**	52.5 (0.9)	52.9 (1.1)	**79.2 (0.9)**	75.4 (0.9)	52.6 (0.9)	55.2 (0.9)
($d = 300$)	100	**79.9 (1.1)**	77.1 (0.7)	53.1 (1.0)	53.9 (1.1)	**85.6 (0.6)**	81.0 (0.5)	55.7 (0.8)	56.2 (0.9)
#Best/Comp.		20	11	5	13	21	13	7	9

The boldface denotes the best and comparable methods in terms of the average AUC according to the t test at the significance level 5%. The last row shows the number of best/comparable cases of each method. SSRboost is an abbreviation for SSRankboost

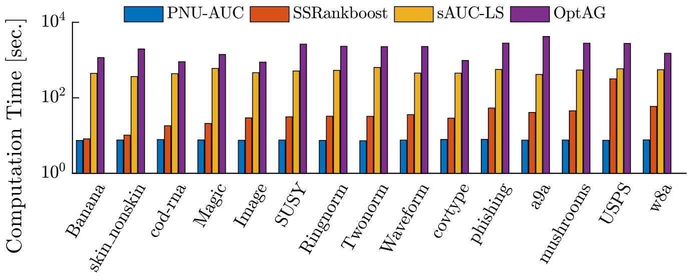


Fig. 4 Average computation time of each method on benchmark data sets when $n_L = 100$ and $\theta_P = 0.1$ over 50 trials

Table 5 Average with standard error of the AUC over 20 trials on the text classification data sets

Data set	d	$\widehat{\theta}_P$	PNU-AUC	SSRankboost	sAUC-LS	OptAG
rcv1	47236	0.21 (0.00)	**91.1** (**0.7**)	76.6 (0.7)	79.6 (2.3)	79.4 (2.0)
amazon2	262144	0.21 (0.01)	**87.2** (**1.3**)	68.7 (1.8)	55.9 (0.4)	56.8 (0.5)
news20	1355191	0.20 (0.00)	**79.8** (**1.1**)	74.2 (2.0)	66.4 (1.6)	71.5 (1.2)

The boldface denotes the best and comparable methods in terms of the average AUC according to the t test at the significance level 5%

based on the logistic loss in which the number of operations for loss evaluation is $\mathcal{O}(n_P n_N + n_P n_U + n_N n_U)$, unlike the PNU-AUC optimization method with the squared loss in which the number of operations for loss evaluation is $\mathcal{O}(n_P + n_N + n_U)$ (cf. the discussion about the computational complexity in Sect. 5).

6.3 Text classification

Next, we apply our proposed PNU-AUC optimization method to a text classification task. We used the *Reuters Corpus Volume I* data set (Lewis et al. 2004), the *Amazon Review* data set (Dredze et al. 2008), and the 20 *Newsgroups* data set (Lang 1995). More specifically, we used the data set processed for a binary classification task: the rcv1, amazon2, and news20 data sets. The rcv1 and news20 data sets are available at the website of LIBSVM (Chang et al. 2011), and the amazon2 is designed by ourselves, which consists of the product reviews of books and music from the *Amazon7* data set (Blondel et al. 2013). The dimension of a feature vector of the rcv1 data set is 47, 236, that of the amazon2 data set is 262,144, and that of the news20 data set is 1, 355, 191.

We trained a classifier with samples of size $n_P = 20$, $n_N = 80$, and $n_U = 10,000$. The true class-prior was set at $\theta_P = 0.2$ and estimated by the method based on energy distance minimization (Kawakubo et al. 2016). For the generative model of OptAG, we employed naive Bayes (NB) multinomial models and a Dirichlet prior for the prior distribution of the NB model as described in Fujino and Ueda (2016).

Table 5 lists the average with standard error of the AUC over 20 trials, showing that the proposed method outperforms the existing methods. Figure 5 summarizes the average computation time of each method. These results show that the proposed method achieves better performance with short computation time.

Footnote 9 continued
where $\boldsymbol{\mu}_P^0$, $\boldsymbol{\mu}^0$, a_P^0, b_P^0, a_N^0, b_N^0, ρ_P^0, andρ_N^0 are the hyperparameters.

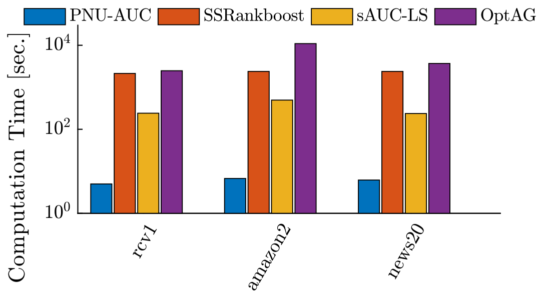


Fig. 5 Average computation time of each method on the text classification data sets

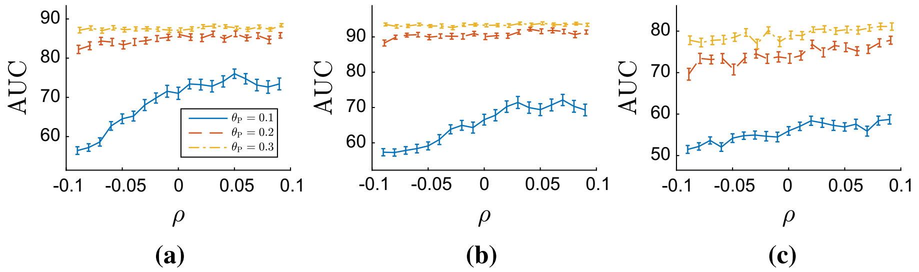


Fig. 6 Average with standard error of the AUC as a function of the noise ρ over 100 trials. The PNU-AUC optimization method used the noisy class-prior $\widehat{\theta}_P = \theta_P + \rho$ in training. The plots show that when $\theta_P = 0.2$ and 0.3, the performance of the PNU-AUC optimization method is stable even when the estimated class-prior has some noise. However, when $\theta_P = 0.1$, as the noise is close to $\rho = -0.09$, the performance largely decreases. Since the true class-prior is small, it is sensitive to the negative bias. **a** Banana ($d = 2$). **b** cod-rna ($d = 8$). **c** w8a ($d = 300$)

6.4 Sensitivity analysis

Here, we investigate the effect of the estimation accuracy of the class-prior for the PNU-AUC optimization method. Specifically, we added noise $\rho \in \{-0.09, -0.08, \dots, 0.09\}$ to the true class-prior θ_P and used $\widehat{\theta}_P = \theta_P + \rho$ as the estimated class-prior for the PNU-AUC optimization method. Under the different values of the class-prior $\theta_P = 0.1, 0.2,$ and 0.3, we trained a classifier with samples of size $n_P = \theta_P \times 50$, $n_N = \theta_N \times 50$, and $n_U = 1000$.

Figure 6 summarizes the average with standard error of the AUC as a function of the noise. The plots show that when $\theta_P = 0.2$ and 0.3, the performance of the PNU-AUC optimization method is stable even when the estimated class-prior has some noise. On the other hand, when $\theta_P = 0.1$, as the noise is close to $\rho = -0.09$, the performance largely decreases. Since the true class-prior is small, it is sensitive to the negative bias. In particular, when $\rho = -0.09$, the gap between the estimated and true class-priors is larger than other values. For instance, when $\rho = -0.09$ and $\theta_P = 0.2$, $\theta_P/\widehat{\theta}_P \approx 1.8$, but when $\rho = -0.09$ and $\theta_P = 0.1$, $\theta_P/\widehat{\theta}_P \approx 10$. In contrast, the positive bias does not heavily affect the performance even when $\theta_P = 0.1$.

6.5 Scalability

Finally, we report the scalability of our proposed PNU-AUC optimization method. Specifically, we evaluated the AUC and computation time while increasing the number of unlabeled samples. We picked two large data sets: the SUSY and amazon2 data sets. The number of positive and negative samples were $n_P = 40$ and $n_N = 160$, respectively.

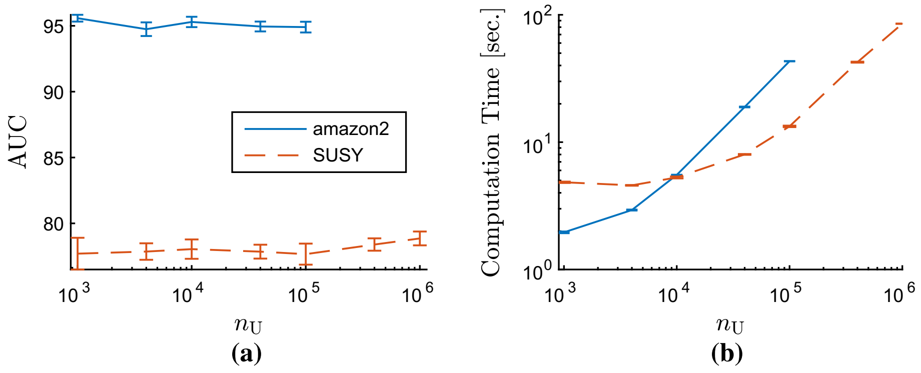


Fig. 7 Average with standard error of the AUC as a function of the number of unlabeled data n_U over 20 trials. **a** AUC. **b** Computation time

Figure 7 summarizes the average with standard error of the AUC and computation time as a function of the number of unlabeled samples. The AUC on the SUSY data set slightly increased at n_U =1,000,000, but the improvement on the amazon2 data set was not noticeable or the performance decreased slightly. In this experiment, the increase of the size of unlabeled data did not significantly improve the performance of the classifier, but it did not affect adversely, i.e., it did not cause significant performance degeneration.

The result of computation time shows that the proposed PNU-AUC optimization method can handle approximately 1,000,000 samples within reasonable computation time in this experiment. The longer computation time on the SUSY data set before n_U =10,000 is because we need to choose one additional hyperparameter, i.e., the bandwidth of the Gaussian kernel basis function, compared with the linear kernel basis function. However, the effect gradually decreases; after n_U =10,000, the matrix multiplication of the high dimensional matrix on the amazon2 data set ($d = 262, 144$) requires more computation time than the SUSY data set ($d = 18$).

7 Conclusions

In this paper, we proposed a novel AUC optimization method from positive and unlabeled data and extend it to a novel semi-supervised AUC optimization method. Unlike the existing approach, our approach does not rely on strong distributional assumptions on the data distributions such as the cluster and the entropy minimization principle. Without the distributional assumptions, we theoretically derived the generalization error bounds of our PU and semi-supervised AUC optimization methods. Moreover, for our semi-supervised AUC optimization method, we showed that the variance of the empirical risk can be smaller than that of the supervised counterpart. Through numerical experiments, we demonstrated the practical usefulness of the proposed PU and semi-supervised AUC optimization methods.

Acknowledgements TS was supported by KAKENHI 15J09111. GN was supported by the JST CREST program and Microsoft Research Asia. MS was supported by JST CREST JPMJCR1403. We thank Han Bao for his comments.

Appendix A: PU-AUC risk estimator

In this section, we discuss the way of estimating the proposed PU-AUC risk. Recall that the PU-AUC risk in Eq. (4) is defined as

$$R_{\mathrm{PU}}(f) = \frac{1}{\theta_{\mathrm{N}}} \mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{U}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}}))]\right] - \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}} \mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\overline{\mathrm{P}}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]\right].$$

If one additional set of positive samples $\{\overline{\boldsymbol{x}}_i^{\mathrm{P}}\}_{i=1}^{n_{\mathrm{P}}}$ is available, we obtain the unbiased PU-AUC risk estimator by

$$\widehat{R}_{\mathrm{PU}}(f) = \frac{1}{\theta_{\mathrm{N}}} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{k=1}^{n_{\mathrm{U}}} \ell\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_k^{\mathrm{U}})\right) - \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}} n_{\mathrm{P}}{}^2} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'=1}^{n_{\mathrm{P}}} \ell\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \overline{\boldsymbol{x}}_{i'}^{\mathrm{P}})\right).$$

We used this estimator in our theoretical analyses because learning is not involved. However, obtaining one additional set of samples is not always possible in practice. Thus, instead of the above risk estimator, we use the following risk estimator in our implementation:

$$\widehat{R}_{\mathrm{PU}}(f) = \frac{1}{\theta_{\mathrm{N}}} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{k=1}^{n_{\mathrm{U}}} \ell\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_k^{\mathrm{U}})\right) - \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}} \left(\frac{1}{n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'=1}^{n_{\mathrm{P}}} \ell\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_{i'}^{\mathrm{P}})\right) - \frac{\ell(0)}{n_{\mathrm{P}}-1}\right).$$

This estimator is also unbiased. To show unbiasedness of this estimator, let us rewrite the second term of the PU-AUC risk without coefficient in Eq. (4) as

$$\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\overline{\mathrm{P}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]\right] = \mathrm{E}_{\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))\right].$$

The unbiased estimator can be expressed as

$$\frac{1}{n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'=1}^{n_{\mathrm{P}}} \ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_{i'}^{\mathrm{P}})) - \frac{\ell(0)}{n_{\mathrm{P}}-1},$$

because the expectation of the above estimator can be computed as follows:

$$\begin{aligned}
&\mathrm{E}_{\boldsymbol{x}_1^{\mathrm{P}},\ldots,\boldsymbol{x}_{n_{\mathrm{P}}}^{\mathrm{P}}}\left[\frac{1}{n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'=1}^{n_{\mathrm{P}}} \ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_{i'}^{\mathrm{P}})) - \frac{\ell(0)}{n_{\mathrm{P}}-1}\right] \\
&= \mathrm{E}_{\boldsymbol{x}_1^{\mathrm{P}},\ldots,\boldsymbol{x}_{n_{\mathrm{P}}}^{\mathrm{P}}}\left[\frac{1}{n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \left(\sum_{i=1}^{n_{\mathrm{P}}} \ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_i^{\mathrm{P}})) + \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'\neq i}^{n_{\mathrm{P}}} \ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_{i'}^{\mathrm{P}}))\right) - \frac{\ell(0)}{n_{\mathrm{P}}-1}\right] \\
&= \mathrm{E}_{\boldsymbol{x}_1^{\mathrm{P}},\ldots,\boldsymbol{x}_{n_{\mathrm{P}}}^{\mathrm{P}}}\left[\frac{1}{n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \left(\sum_{i=1}^{n_{\mathrm{P}}} \ell(0) + \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'\neq i}^{n_{\mathrm{P}}} \ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_{i'}^{\mathrm{P}}))\right) - \frac{\ell(0)}{n_{\mathrm{P}}-1}\right] \\
&= \frac{1}{n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'\neq i}^{n_{\mathrm{P}}} \mathrm{E}_{\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_{i'}^{\mathrm{P}}}\left[\ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_{i'}^{\mathrm{P}}))\right] \\
&= \frac{1}{n_{\mathrm{P}}(n_{\mathrm{P}}-1)} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'\neq i}^{n_{\mathrm{P}}} \mathrm{E}_{\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))\right] \\
&= \mathrm{E}_{\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))\right],
\end{aligned}$$

where we used $f(\boldsymbol{x}, \boldsymbol{x}) = \boldsymbol{w}^\top(\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{x})) = 0$ from the second to third lines. If the squared loss function $\ell(m) = (1-m)^2$ is used, $\ell(0) = 1$ (cf. the implementation with the squared loss in Sect. 5). Therefore, the proposed PU-AUC risk estimator is unbiased.

Appendix B: Proof of generalization error bounds

Here, we give the proofs of generalization error bounds in Sect. 4.1. The proofs are based on Usunier et al. (2006).

Let $\{\boldsymbol{x}_i\}_{i=1}^m$ and $\{\boldsymbol{x}'_j\}_{j=1}^n$ be two sets of samples drawn from the distribution equipped with densities $q(\boldsymbol{x})$ and $q'(\boldsymbol{x})$, respectively. Recall $\mathcal{F}$ be a function class of bounded hyperplanes:

$$\mathcal{F} := \{f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{x}')\rangle \mid \|\boldsymbol{w}\| \le C_{\boldsymbol{w}};\ \ \forall \boldsymbol{x}\colon \|\boldsymbol{\phi}(\boldsymbol{x})\| \le C_{\boldsymbol{\phi}}\},$$

where $C_{\boldsymbol{w}} > 0$ and $C_{\boldsymbol{\phi}} > 0$ are certain positive constants. Then, the AUC risk over distributions q and q' and its empirical version can be expressed as

$$R(f) := \mathrm{E}_{\boldsymbol{x}\sim q}\left[\mathrm{E}_{\boldsymbol{x}'\sim q'}[\ell(f(\boldsymbol{x}, \boldsymbol{x}'))]\right],$$

$$\widehat{R}(f) := \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\ell\left(f(\boldsymbol{x}_i, \boldsymbol{x}'_j)\right).$$

For convenience, we define

$$h(\delta) := 2\sqrt{2}LC_\ell C_{\boldsymbol{w}} C_{\boldsymbol{\phi}} + \frac{3}{2}\sqrt{2\log(2/\delta)}.$$

We first have the following theorem:

Theorem 4 *For any $\delta > 0$, the following inequality holds with probability at least $1-\delta$ for any $f \in \mathcal{F}$:*

$$R(f) - \widehat{R}(f) \le h(\delta)\frac{1}{\sqrt{\min(n, n')}}.$$

Proof By slightly modifying Theorem 7 in Usunier et al. (2006) to fit our setting, for any $\delta > 0$, with probability at least $1-\delta$ for any $f \in \mathcal{F}$, we have

$$R(f) - \widehat{R}(f) \le \frac{2LC_\ell C_{\boldsymbol{w}}\sqrt{\max(n, n')}}{nn'}\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n'}\|\boldsymbol{\phi}(\boldsymbol{x}_i) - \boldsymbol{\phi}(\boldsymbol{x}_j)\|^2} + 3\sqrt{\frac{\log(2/\delta)}{2\min(n, n')}}. \tag{9}$$

Applying the inequality

$$\sum_{i=1}^{n}\sum_{j=1}^{n'}\|\boldsymbol{\phi}(\boldsymbol{x}_i) - \boldsymbol{\phi}(\boldsymbol{x}_j)\|^2 \le n'\sum_{i=1}^{n}\|\boldsymbol{\phi}(\boldsymbol{x}_i)\|^2 + n\sum_{j=1}^{n'}\|\boldsymbol{\phi}(\boldsymbol{x}_j)\|^2$$
$$\le 2nn'C_{\boldsymbol{\phi}}^2,$$

to the first term in Eq. (9), we obtain the theorem. □

By using Theorem 4, we prove the risk bounds of the PU-AUC and NU-AUC risks:

Lemma 1 *For any $\delta > 0$, the following inequalities hold separately with probability at least $1 - \delta$ for any $f \in \mathcal{F}$:*

$$R_{\mathrm{PU}}(f) - \widehat{R}_{\mathrm{PU}}(f) \leq h(\delta/2)\left(\frac{1}{\theta_{\mathrm{N}}\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{U}})}} + \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}\sqrt{n_{\mathrm{P}}}}\right),$$
$$R_{\mathrm{NU}}(f) - \widehat{R}_{\mathrm{NU}}(f) \leq h(\delta/2)\left(\frac{1}{\theta_{\mathrm{P}}\sqrt{\min(n_{\mathrm{N}}, n_{\mathrm{U}})}} + \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}\sqrt{n_{\mathrm{N}}}}\right).$$

Proof Recall that the PU-AUC and NU-AUC risks are expressed as

$$R_{\mathrm{PU}}(f) = \frac{1}{\theta_{\mathrm{N}}} \mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{U}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}}))]\right] - \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}} \mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\overline{\mathrm{P}}}\left[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]\right],$$
$$R_{\mathrm{NU}}(f) = \frac{1}{\theta_{\mathrm{P}}} \mathrm{E}_{\mathrm{U}}\left[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{U}}, \boldsymbol{x}^{\mathrm{N}}))]\right] - \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}} \mathrm{E}_{\mathrm{N}}\left[\mathrm{E}_{\overline{\mathrm{N}}}[\ell(f(\boldsymbol{x}^{\mathrm{N}}, \overline{\boldsymbol{x}}^{\mathrm{N}}))]\right].$$

Based on Theorem 4, for any $\delta > 0$, we have these uniform deviation bounds with probability at least $1 - \delta/2$:

$$\sup_{f\in\mathcal{F}}\left(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\mathrm{U}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}}))]] - \frac{1}{n_{\mathrm{P}}n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{k=1}^{n_{\mathrm{U}}}\ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_k^{\mathrm{U}}))\right) \leq h(\delta/2)\frac{1}{\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{U}})}},$$
$$\sup_{f\in\mathcal{F}}\left(\mathrm{E}_{\mathrm{U}}[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{U}}, \boldsymbol{x}^{\mathrm{N}}))]] - \frac{1}{n_{\mathrm{N}}n_{\mathrm{U}}}\sum_{k=1}^{n_{\mathrm{U}}}\sum_{j=1}^{n_{\mathrm{N}}}\ell(f(\boldsymbol{x}_k^{\mathrm{U}}, \boldsymbol{x}_j^{\mathrm{N}}))\right) \leq h(\delta/2)\frac{1}{\sqrt{\min(n_{\mathrm{N}}, n_{\mathrm{U}})}},$$
$$\sup_{f\in\mathcal{F}}\left(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\overline{\mathrm{P}}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]] - \frac{1}{{n_{\mathrm{P}}}^2}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{i'=1}^{n_{\mathrm{P}}}\ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \overline{\boldsymbol{x}}_{i'}^{\mathrm{P}}))\right) \leq h(\delta/2)\frac{1}{\sqrt{n_{\mathrm{P}}}},$$
$$\sup_{f\in\mathcal{F}}\left(\mathrm{E}_{\mathrm{N}}[\mathrm{E}_{\overline{\mathrm{N}}}[\ell(f(\boldsymbol{x}^{\mathrm{N}}, \overline{\boldsymbol{x}}^{\mathrm{N}}))]] - \frac{1}{{n_{\mathrm{N}}}^2}\sum_{j=1}^{n_{\mathrm{N}}}\sum_{j'=1}^{n_{\mathrm{N}}}\ell(f(\boldsymbol{x}_j^{\mathrm{N}}, \overline{\boldsymbol{x}}_{j'}^{\mathrm{N}}))\right) \leq h(\delta/2)\frac{1}{\sqrt{n_{\mathrm{N}}}},$$

Simple calculation showed that for any $\delta > 0$, with probability $1 - \delta$, we have

$$\begin{aligned}\sup_{f\in\mathcal{F}}\left(R_{\mathrm{PU}}(f) - \widehat{R}_{\mathrm{PU}}(f)\right) &\leq \frac{1}{\theta_{\mathrm{N}}}\sup_{f\in\mathcal{F}}\left(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\mathrm{U}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}}))]] - \frac{1}{n_{\mathrm{P}}n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{k=1}^{n_{\mathrm{U}}}\ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_k^{\mathrm{U}}))\right)\\ &\quad + \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}}\sup_{f\in\mathcal{F}}\left(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\overline{\mathrm{P}}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]] - \frac{1}{{n_{\mathrm{P}}}^2}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{i'=1}^{n_{\mathrm{P}}}\ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \overline{\boldsymbol{x}}_{i'}^{\mathrm{P}}))\right)\\ &\leq h(\delta/2)\left(\frac{1}{\theta_{\mathrm{N}}\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{U}})}} + \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}\sqrt{n_{\mathrm{P}}}}\right), \end{aligned} \tag{10}$$

where we used

$$\sup(x + y) \leq \sup(x) + \sup(y),$$
$$R_{\mathrm{PU}}(f) \leq \frac{1}{\theta_{\mathrm{N}}}\mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\mathrm{U}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}}))]\right] + \frac{\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}}\mathrm{E}_{\mathrm{P}}\left[\mathrm{E}_{\overline{\mathrm{P}}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]\right].$$

Similarly, for the NU-AUC risk, we have

$$\sup_{f\in\mathcal{F}}\left(R_{\mathrm{NU}}(f) - \widehat{R}_{\mathrm{NU}}(f)\right) \leq h(\delta/2)\left(\frac{1}{\theta_{\mathrm{P}}\sqrt{\min(n_{\mathrm{N}}, n_{\mathrm{U}})}} + \frac{\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}\sqrt{n_{\mathrm{N}}}}\right). \tag{11}$$

Equations (10) and (11) conclude the lemma. □

Finally, we give the proof of Theorem 1.

Proof Assume the loss satisfying $\ell_{0\text{-}1}(m) \leq M\ell(m)$. We have $I(f) \leq MR(f)$. When $M = 1$ such as $\ell_{\mathrm{S}}(m)$ and $\ell_{\mathrm{E}}(m)$, $I(f) \leq R(f)$ holds. This observation yields Theorem 1. □

Next, we prove the generalization error bounds of the PNPU-AUC and PNNU-AUC risks in Theorem 2. We first prove the following risk bounds:

Lemma 2 *For any* $\delta > 0$*, the following inequalities hold separately with probability at least* $1 - \delta$ *for all* $f \in \mathcal{F}$*:*

$$R^{\gamma}_{\mathrm{PNPU}}(f) - \widehat{R}^{\gamma}_{\mathrm{PNPU}}(f) \leq h(\delta/3)\Big(\frac{1-\gamma}{\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{N}})}} + \frac{\gamma}{\theta_{\mathrm{N}}\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{U}})}} + \frac{\theta_{\mathrm{P}}\gamma}{\theta_{\mathrm{N}}\sqrt{n_{\mathrm{P}}}}\Big),$$
$$R^{\gamma}_{\mathrm{PNNU}}(f) - \widehat{R}^{\gamma}_{\mathrm{PNNU}}(f) \leq h(\delta/3)\Big(\frac{1-\gamma}{\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{N}})}} + \frac{\gamma}{\theta_{\mathrm{P}}\sqrt{\min(n_{\mathrm{N}}, n_{\mathrm{U}})}} + \frac{\theta_{\mathrm{N}}\gamma}{\theta_{\mathrm{P}}\sqrt{n_{\mathrm{N}}}}\Big).$$

Proof Recall the PNPU-AUC and PNNU-AUC risks:

$$R^{\gamma}_{\mathrm{PNPU}}(f) := (1-\gamma)R_{\mathrm{PN}}(f) + \gamma R_{\mathrm{PU}}(f),$$
$$R^{\gamma}_{\mathrm{PNNU}}(f) := (1-\gamma)R_{\mathrm{PN}}(f) + \gamma R_{\mathrm{NU}}(f).$$

Based on Theorem 4, for any $\delta > 0$, we have these uniform deviation bounds with probability at least $1 - \delta/3$:

$$\sup_{f\in\mathcal{F}}\Big(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{N}}))]] - \frac{1}{n_{\mathrm{P}}n_{\mathrm{N}}}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{j=1}^{n_{\mathrm{N}}}\ell(f(\boldsymbol{x}^{\mathrm{P}}_i, \boldsymbol{x}^{\mathrm{N}}_j))\Big) \leq h(\delta/3)\frac{1}{\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{N}})}},$$

$$\sup_{f\in\mathcal{F}}\Big(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\mathrm{U}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}}))]] - \frac{1}{n_{\mathrm{P}}n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{k=1}^{n_{\mathrm{U}}}\ell(f(\boldsymbol{x}^{\mathrm{P}}_i, \boldsymbol{x}^{\mathrm{U}}_k))\Big) \leq h(\delta/3)\frac{1}{\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{U}})}},$$

$$\sup_{f\in\mathcal{F}}\Big(\mathrm{E}_{\mathrm{U}}[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{U}}, \boldsymbol{x}^{\mathrm{N}}))]] - \frac{1}{n_{\mathrm{N}}n_{\mathrm{U}}}\sum_{k=1}^{n_{\mathrm{U}}}\sum_{j=1}^{n_{\mathrm{N}}}\ell(f(\boldsymbol{x}^{\mathrm{U}}_k, \boldsymbol{x}^{\mathrm{N}}_j))\Big) \leq h(\delta/3)\frac{1}{\sqrt{\min(n_{\mathrm{N}}, n_{\mathrm{U}})}},$$

$$\sup_{f\in\mathcal{F}}\Big(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\overline{\mathrm{P}}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]] - \frac{1}{{n_{\mathrm{P}}}^2}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{i'=1}^{n_{\mathrm{P}}}\ell(f(\boldsymbol{x}^{\mathrm{P}}_i, \overline{\boldsymbol{x}}^{\mathrm{P}}_{i'}))\Big) \leq h(\delta/3)\frac{1}{\sqrt{n_{\mathrm{P}}}},$$

$$\sup_{f\in\mathcal{F}}\Big(\mathrm{E}_{\mathrm{N}}[\mathrm{E}_{\overline{\mathrm{N}}}[\ell(f(\boldsymbol{x}^{\mathrm{N}}, \overline{\boldsymbol{x}}^{\mathrm{N}}))]] - \frac{1}{{n_{\mathrm{N}}}^2}\sum_{j=1}^{n_{\mathrm{N}}}\sum_{j'=1}^{n_{\mathrm{N}}}\ell(f(\boldsymbol{x}^{\mathrm{N}}_j, \overline{\boldsymbol{x}}^{\mathrm{N}}_{j'}))\Big) \leq h(\delta/3)\frac{1}{\sqrt{n_{\mathrm{N}}}}.$$

Combining three bounds from the above, for any $\delta > 0$, with probability $1 - \delta$, we have

$$\begin{aligned}
&\sup_{f\in\mathcal{F}}\Big(R^{\gamma}_{\mathrm{PNPU}}(f) - \widehat{R}^{\gamma}_{\mathrm{PNPU}}(f)\Big)\\
&\leq (1-\gamma)\sup_{f\in\mathcal{F}}\Big(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\mathrm{N}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{N}}))]] - \frac{1}{n_{\mathrm{P}}n_{\mathrm{N}}}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{j=1}^{n_{\mathrm{N}}}\ell(f(\boldsymbol{x}^{\mathrm{P}}_i, \boldsymbol{x}^{\mathrm{N}}_j))\Big)\\
&\quad + \frac{\gamma}{\theta_{\mathrm{N}}}\sup_{f\in\mathcal{F}}\Big(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\mathrm{U}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \boldsymbol{x}^{\mathrm{U}}))]] - \frac{1}{n_{\mathrm{P}}n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{P}}}\sum_{k=1}^{n_{\mathrm{U}}}\ell(f(\boldsymbol{x}^{\mathrm{P}}_i, \boldsymbol{x}^{\mathrm{U}}_k))\Big)
\end{aligned}$$

$$+ \frac{\gamma\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}} \sup_{f\in\mathcal{F}} \Big(\mathrm{E}_{\mathrm{P}}[\mathrm{E}_{\overline{\mathrm{P}}}[\ell(f(\boldsymbol{x}^{\mathrm{P}}, \overline{\boldsymbol{x}}^{\mathrm{P}}))]] - \frac{1}{n_{\mathrm{P}}{}^2} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'=1}^{n_{\mathrm{P}}} \ell(f(\boldsymbol{x}_i^{\mathrm{P}}, \overline{\boldsymbol{x}}_{i'}^{\mathrm{P}})) \Big)$$

$$\leq h(\delta/3)\Big(\frac{1-\gamma}{\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{N}})}} + \frac{\gamma}{\theta_{\mathrm{N}}\sqrt{\min(n_{\mathrm{P}}, n_{\mathrm{U}})}} + \frac{\gamma\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}\sqrt{n_{\mathrm{P}}}} \Big).$$

This concludes the risk bounds of the PNPU-AUC risk.

Similarly, we prove the risk bounds of the PNNU-AUC risk. □

Again, $I(f) \leq R(f)$ holds in our setting. This leads to Theorem 2.

Appendix C: Proof of variance reduction

Here, we give the proof of Theorem 3.

Proof The empirical PNPU-AUC risk can be expressed as

$$\begin{aligned}
\widehat{R}^{\gamma}_{\mathrm{PNPU}}(f) &= (1-\gamma)\widehat{R}_{\mathrm{PN}}(f) + \gamma\widehat{R}_{\mathrm{PU}}(f) \\
&= \frac{1-\gamma}{n_{\mathrm{P}}n_{\mathrm{N}}} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{j=1}^{n_{\mathrm{N}}} \ell\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_j^{\mathrm{N}})\right) + \frac{\gamma}{\theta_{\mathrm{N}}n_{\mathrm{P}}n_{\mathrm{U}}} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{k=1}^{n_{\mathrm{U}}} \ell\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \boldsymbol{x}_k^{\mathrm{U}})\right) \\
&\quad - \frac{\gamma\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}n_{\mathrm{P}}{}^2} \sum_{i=1}^{n_{\mathrm{P}}} \sum_{i'=1}^{n_{\mathrm{P}}} \ell\left(f(\boldsymbol{x}_i^{\mathrm{P}}, \overline{\boldsymbol{x}}_{i'}^{\mathrm{P}})\right).
\end{aligned}$$

Assume $n_{\mathrm{U}} \to \infty$, we obtain

$$\begin{aligned}
\mathrm{Var}\left[\widehat{R}^{\gamma}_{\mathrm{PNPU}}(f)\right] &= \frac{(1-\gamma)^2}{n_{\mathrm{P}}n_{\mathrm{N}}}\sigma^2_{\mathrm{PN}}(f) + \frac{\gamma^2\theta_{\mathrm{P}}^2}{n_{\mathrm{P}}{}^2\theta_{\mathrm{N}}^2}\sigma^2_{\mathrm{PP}}(f) + \frac{(1-\gamma)\gamma}{\theta_{\mathrm{N}}n_{\mathrm{P}}}\tau_{\mathrm{PN,PU}}(f) \\
&\quad - \frac{\gamma^2\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}^2 n_{\mathrm{P}}}\tau_{\mathrm{PU,PP}}(f) - \frac{(1-\gamma)\gamma\theta_{\mathrm{P}}}{\theta_{\mathrm{N}}n_{\mathrm{P}}}\tau_{\mathrm{PN,PP}}(f) \\
&= (1-\gamma)^2\psi_{\mathrm{PN}} + \gamma^2\psi_{\mathrm{PU}} + (1-\gamma)\gamma\psi_{\mathrm{PP}},
\end{aligned}$$

where the terms divided by n_{U} are disappeared. Setting the derivative with respect to γ at zero, we obtain the minimizer in Eq. (7).

For the empirical PNNU-AUC risk, when $n_{\mathrm{U}} \to \infty$, we obtain

$$\begin{aligned}
\mathrm{Var}\left[\widehat{R}^{\gamma}_{\mathrm{PNNU}}(g)\right] &= \frac{(1-\gamma)^2}{n_{\mathrm{P}}n_{\mathrm{N}}}\sigma^2_{\mathrm{PN}}(g) + \frac{\gamma^2\theta_{\mathrm{N}}^2}{n_{\mathrm{N}}{}^2\theta_{\mathrm{P}}^2}\sigma^2_{\mathrm{NN}}(g) + \frac{(1-\gamma)\gamma}{\theta_{\mathrm{P}}n_{\mathrm{N}}}\tau_{\mathrm{PN,NU}}(g) \\
&\quad - \frac{\gamma^2\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}^2 n_{\mathrm{N}}}\tau_{\mathrm{NU,NN}}(g) - \frac{(1-\gamma)\gamma\theta_{\mathrm{N}}}{\theta_{\mathrm{P}}n_{\mathrm{N}}}\tau_{\mathrm{PN,NN}}(g) \\
&= (1-\gamma)^2\psi_{\mathrm{PN}} + \gamma^2\psi_{\mathrm{NU}} + (1-\gamma)\gamma\psi_{\mathrm{NN}}.
\end{aligned}$$

Setting the derivative with respect to γ at zero, we obtain the minimizer in Eq. (8). □

Appendix D: Statistics of data sets

Table 6 summarizes the statistics of the data sets used in our experiments. The class balance is the number of positive samples divided by that of total samples. The sources of data sets

Table 6 The statistics of the data sets

Data set	Dimension	#Samples	Class balance	Source
Banana	2	5,300	0.45	IDA
skin_nonskin	3	245,057	0.21	LIBSVM
cod-rna	8	331152	0.67	LIBSVM
Magic	10	19020	0.35	UCI
Image	18	2,310	0.39	IDA
SUSY	18	5,000,000	0.46	LIBSVM
Ringnorm	20	7,400	0.50	IDA
Twonorm	20	7,400	0.50	IDA
Waveform	21	5,000	0.33	IDA
covtype	54	581,012	0.51	LIBSVM
phishing	68	11,055	0.44	LIBSVM
a9a	83	48,842	0.24	LIBSVM
mushrooms	112	8,124	0.48	LIBSVM
USPS	241	1,500	0.20	SSL
w8a	300	64,700	0.03	LIBSVM
rcv1	47,236	697,641	0.53	LIBSVM
amazon2	262,144	1,149,374	0.18	Amazon7
news20	1,355,191	19,996	0.50	LIBSVM

The source of data sets is as follows: the IDA Benchmark Repository (IDA) (Rätsch et al. 2001), the UCI Machine Learning Repository (UCI) (Lichman 2013), the LIBSVM data sets (LIBSVM) (Chang et al. 2011), the Semi-Supervised Learning Book (SSL) (Chapelle et al. 2006), and the Amazon Review (Amazon7) (Blondel et al. 2013)

are as follows: the IDA Benchmark Repository (IDA) (Rätsch et al. 2001), the UCI Machine Learning Repository (UCI) (Lichman 2013), the LIBSVM data sets (LIBSVM) (Chang et al. 2011), the Semi-Supervised Learning Book (SSL) (Chapelle et al. 2006), and the Amazon Review (Amazon7) (Blondel et al. 2013).

References

Amini, M. R., Truong, T. V., & Goutte, C. (2008). A boosting algorithm for learning bipartite ranking functions with partially labeled data. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 99–106).

Bartlett, P. L., Jordan, M. I., & McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, *101*(473), 138–156.

Blondel, M., Seki, K., & Uehara, K. (2013). Block coordinate descent algorithms for large-scale sparse multiclass classification. *Machine Learning*, *93*(1), 31–52.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. Cambridge: MIT Press.

Cortes, C., & Mohri, M. (2004). AUC optimization vs. error rate minimization. *Advances in Neural Information Processing Systems*, *16*, 313–320.

Cozman, F. G., Cohen, I., & Cirelo, M. C. (2003). Semi-supervised learning of mixture models. In *Proceedings of the 20th international conference on machine learning* (pp. 99–106).

Dredze, M., Crammer, K., & Pereira, F. (2008). Confidence-weighted linear classification. In *Proceedings of the 25th international conference on machine learning* (pp. 264–271).

du Plessis, M. C., Niu, G., & Sugiyama, M. (2014). Analysis of learning from positive and unlabeled data. *Advances in Neural Information Processing Systems*, *27*, 703–711.

du Plessis, M. C., Niu, G., & Sugiyama, M. (2017). Class-prior estimation for learning from positive and unlabeled data. *Machine Learning*, *106*(4), 463–492.

du Plessis, M. C., Niu, G., & Sugiyama, M. (2015). Convex formulation for learning from positive and unlabeled data. In *Proceedings of 32nd international conference on machine learning, JMLR workshop and conference proceedings* (Vol. 37, pp. 1386–1394).

Fujino, A., Ueda, N. (2016). A semi-supervised AUC optimization method with generative models. In *IEEE 16th international conference on data mining* (pp. 883–888).

Gao, W., & Zhou, Z. H. (2015). On the consistency of AUC pairwise optimization. In *International joint conference on artificial intelligence* (pp. 939–945).

Gao, W., Wang, L., Jin, R., Zhu, S., & Zhou, Z. H. (2016). One-pass AUC optimization. *Artificial Intelligence*, *236*(C), 1–29.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), 29–36.

Herschtal, A., & Raskutti, B. (2004). Optimising area under the ROC curve using gradient descent. In *Proceedings of the 21st international conference on machine learning*.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 133–142).

Kawakubo, H., du Plessis, M. C., & Sugiyama, M. (2016). Computationally efficient class-prior estimation under class balance change using energy distance. *IEICE Transactions on Information and Systems*, *E99–D*(1), 176–186.

Kotlowski, W., Dembczynski, K. J, & Huellermeier, E. (2011). Bipartite ranking through minimization of univariate loss. In *Proceedings of the 28th international conference on machine learning* (pp. 1113–1120).

Krijthe, J. H., & Loog, M. (2017). Robust semi-supervised least squares classification by implicit constraints. *Pattern Recognition*, *63*, 115–126.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international machine learning conference*.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, *5*, 361–397.

Li, Y. F., & Zhou, Z. H. (2015). Towards making unlabeled data never hurt. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(1), 175–188.

Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Mendelson, S. (2008). Lower bounds for the empirical minimization algorithm. *IEEE Transactions on Information Theory*, *54*(8), 3797–3803.

Niu, G., du Plessis, M. C., Sakai, T., Ma Y., & Sugiyama, M. (2016). Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., & Garnett, R. (Eds.) *Advances in neural information processing systems* (Vol. 29, pp. 1199–1207)

Rakhlin, A., Shamir, O., & Sridharan, K. (2012). Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th international conference on machine learning* (pp. 449–456).

Rätsch, G., Onoda, T., & Müller, K. R. (2001). Soft margins for adaboost. *Machine Learning*, *42*(3), 287–320.

Sakai, T., du Plessis, M. C., Niu, G., & Sugiyama, M. (2017). Semi-supervised classification based on classification from positive and unlabeled data. In *Proceedings of the 34th international conference on machine learning*.

Sokolovska, N., Cappé, O., & Yvon, F. (2008). The asymptotics of semi-supervised learning in discriminative probabilistic models. In *Proceedings of the 25th international conference on machine learning* (pp. 984–991).

Sundararajan, S., Priyanka, G., & Selvaraj, S. S. K. (2011). A pairwise ranking based approach to learning with positive and unlabeled examples. In *Proceedings of the 20th ACM international conference on information and knowledge management* (pp. 663–672).

Usunier, N., Amini, M., & Patrick, G. (2006). Generalization error bounds for classifiers trained with interdependent data. In: Weiss, Y., Schölkopf, P. B., & Platt J. C. (Eds.) *Advances in neural information processing systems* (Vol. 18, pp. 1369–1376). La Jolla, CA: Neural Information Processing Systems Foundation Inc.

Vapnik, V. N. (1998). *Statistical learning theory*. London: Wiley.

Ying, Y., Wen, L., & Lyu, S. (2016). Stochastic online AUC maximization. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., & Garnett, R. (Eds.) *Advances in neural information processing systems* (Vol. 29, pp. 451–459). La Jolla, CA: Neural Information Processing Systems Foundation Inc.

Zhao, P., Jin, R., Yang, T., & Hoi, S. C. (2011). Online AUC maximization. In *Proceedings of the 28th international conference on machine learning* (pp. 233–240).