Covariate Shift Adaptation on Learning from Positive and Unlabeled Data

Tomoya Sakai*
NEC Corporation
t-sakai@ah.jp.nec.com

Nobuyuki Shimizu
Yahoo Japan Corporation
nobushim@yahoo-corp.jp

Abstract

The goal of binary classification is to identify whether an input sample belongs to positive or negative classes. Usually, supervised learning is applied to obtain a classification rule, but in real-world applications, it is conceivable that only positive and unlabeled data are accessible for learning, which is called *learning from positive and unlabeled data* (PU learning). Furthermore, in practice, data distributions are likely to differ between training and testing due to, for example, time variation and domain shift. The *covariate shift* is a dataset shift situation, where distributions of covariates (inputs) differ between training and testing, but the input-output relation is the same. In this paper, we address the PU learning problem under the covariate shift. We propose an importance-weighted PU learning method and reveal in which situations the importance-weighting is necessary. Moreover, we derive the convergence rate of the proposed method under mild conditions and experimentally demonstrate its effectiveness.

1 Introduction

The goal of binary classification is to identify whether an input sample belongs to positive or negative classes. To obtain classification rules, we collect labeled data for both positive and negative classes and use supervised learning. However, in real-world applications, it is conceivable that collecting positive data is easy but collecting negative data is relatively difficult or almost impossible. For example, in social networking services, the users' favorite articles (positive data) can be found using a "like" button, but disliked articles (negative data) cannot be explicitly observed unless a "dislike" button is implemented; the articles that the users did not declare to "like" are a mixture of positive and negative data, i.e., *unlabeled* data.

To address such a situation, *learning from positive and unlabeled data* (PU learning) (Letouzey, Denis, and Gilleron 2000; Lee and Liu 2003; Elkan and Noto 2008) has been studied and recently gaining much attention (Xu et al. 2017; Yang, Liu, and Yang 2017; Kiryo et al. 2017; Gong et al. 2018). In PU learning, a classifier is trained with only positive and unlabeled data, and used for identifying whether test samples belong to positive or negative classes. To obtain the classifier, du Plessis, Niu, and Sugiyama (2015) proposed a method based on empirical risk minimization (ERM), which enables us to compute a risk estimator unbiased to the risk in supervised learning by only using PU data. Moreover, the theoretical properties of the ERM-based PU classification method have been studied in various perspective (Niu et al. 2016; Kiryo et al. 2017).

In addition to the PU learning situation, in practice, a distribution for training can be different from testing due to the time variation and domain change for example. As a distribution shift scenario, the *covariate shift* is widely considered and has been studied so far (Shimodaira 2000; Quioñero Candela et al. 2009). In the covariate shift, the input-output relation is assumed to be the same across training and test, but the input distribution in a test phase is different from that in training. For example, when we want to predict whether a patient has a certain disease in some city but the training data were gathered in a different city, the distribution of patients will often change.

To address the covariate shift, an importance-weighted risk minimization approach was proposed and its superior performance was demonstrated (Quioñero Candela et al. 2009). The importance function is the ratio between test and training densities and it is multiplied by a loss, so that an average of weighted losses over training data approximates an average of (non-weighted) losses over test data. Although the effectiveness of the importance-weighted ERM for the covariate shift has been demonstrated in various machine learning tasks, it has not been elucidated in the PU learning setting even if the PU learning situation frequently occurs in real-world applications.

In this paper, we consider the problem of training a classifier from PU data under the covariate shift. To address the problem, we propose a novel risk function and show the practical implementation. The proposed method can reuse positive data from the training distribution; thus we can save annotation cost for obtaining labeled data from test distribution, which is similar to the existing supervised learning methods for covariate shift adaptation. Our analysis reveal the situations where the importance-weighting is necessary for PU learning under the covariate shift. Besides, we discuss the relation between the covariate shift and the *prior probability shift*, which is also often considered in dataset

*Part of this work was done while at the University of Tokyo and RIKEN.

shift. Furthermore, we derive the convergence rate for the learned classifier. Finally, we demonstrate the effectiveness of our proposed method through numerical experiments.

Related work: There was an attempt to apply PU learning techniques to dataset shift. In Xia et al. (2013), a PU classification method was used for finding important samples in the target domain from domain-mixed training data, where samples from the target domain were regarded as positive and samples from the mixed domains are regarded as unlabeled data. After finding the important samples, they applied an importance-weighted supervised classification method, where both positive and negative data were used for training. Although the idea of PU learning is used in the dataset shift scenario, the existing approach is substantially different from the proposed approach in this paper. Unlike the existing work, since we consider classification tasks under the PU learning setting, we can obtain classification rules even if negative data are not collected in training,

2 Background

In this section, we formulate our problem setting and review the existing algorithms.

2.1 Problem Setting

Let $\boldsymbol{x} \in \mathbb{R}^d$ and $y \in \{\pm 1\}$ be covariate and its class label associated with probability density $p(\boldsymbol{x}, y)$, where d is a positive integer. Let $g\colon \mathbb{R}^d \to \mathbb{R}$ be a classifier and the predicted label is obtained by its sign: $\widehat{y} = \mathrm{sign}(g(\boldsymbol{x}))$. Our goal is to obtain a classifier that minimizes the risk over *test* data as small as it can:

$$R^{\mathrm{te}}(g) := \mathrm{E}_{p_{\mathrm{te}}(\boldsymbol{x},y)}[\ell(yg(\boldsymbol{x}))],$$

where $\mathrm{E}_{p_{\mathrm{te}}(\boldsymbol{x},y)}$ denotes the expectation over the test probability distribution $p_{\mathrm{te}}(\boldsymbol{x}, y)$, and $\ell(m)$ is a loss function.

In PU learning, unlike ordinary supervised classification, we are given positive (P) data but not given negative (N) data; instead, we are given unlabeled (U) data:

$$\begin{aligned}
&\{\boldsymbol{x}_i^{\mathrm{Ptr}}\}_{i=1}^{n_{\mathrm{Ptr}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x} \mid y=+1),\\
&\{\boldsymbol{x}_k^{\mathrm{Utr}}\}_{k=1}^{n_{\mathrm{Utr}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}) = \pi_{\mathrm{Ptr}} p_{\mathrm{tr}}(\boldsymbol{x} \mid y=+1)\\
&\qquad\qquad + \pi_{\mathrm{Ntr}} p_{\mathrm{tr}}(\boldsymbol{x} \mid y=-1),
\end{aligned}$$

where $\pi_{\mathrm{Ptr}} := p_{\mathrm{tr}}(y=+1)$, $\pi_{\mathrm{Ntr}} := p_{\mathrm{tr}}(y=-1)$, and p_{tr} denotes the training probability distribution.

In addition to the PU learning setting, we further consider the situation known as the covariate shift (Shimodaira 2000), where the input-output relation is the same but the input distributions are different between training and test:

$$\begin{aligned}
p_{\mathrm{tr}}(y \mid \boldsymbol{x}) &= p_{\mathrm{te}}(y \mid \boldsymbol{x}),\\
p_{\mathrm{tr}}(\boldsymbol{x}) &\neq p_{\mathrm{te}}(\boldsymbol{x}).
\end{aligned}$$

In our setting, we suppose that unlabeled data from *test* distribution are also given:

$$\begin{aligned}
\{\boldsymbol{x}_k^{\mathrm{Ute}}\}_{k=1}^{n_{\mathrm{Ute}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{te}}(\boldsymbol{x}) &= \pi_{\mathrm{Pte}} p_{\mathrm{te}}(\boldsymbol{x} \mid y=+1)\\
&\quad + \pi_{\mathrm{Nte}} p_{\mathrm{te}}(\boldsymbol{x} \mid y=-1),
\end{aligned}$$

where $\pi_{\mathrm{Pte}} := p_{\mathrm{te}}(y=+1)$ and $\pi_{\mathrm{Nte}} := p_{\mathrm{te}}(y=-1)$.

From these three sets of samples $\{\boldsymbol{x}_i^{\mathrm{Ptr}}\}_{i=1}^{n_{\mathrm{Ptr}}}$, $\{\boldsymbol{x}_k^{\mathrm{Utr}}\}_{k=1}^{n_{\mathrm{Utr}}}$, and $\{\boldsymbol{x}_k^{\mathrm{Ute}}\}_{k=1}^{n_{\mathrm{Ute}}}$, we train a classifier that achieves accurate prediction under the covariate shift.

2.2 PU Classification

We here review the PU classification method based on empirical risk minimization (ERM) (du Plessis, Niu, and Sugiyama 2015).

The ordinary supervised learning risk is expressed as

$$\begin{aligned}
R^{\mathrm{tr}}(g) &= \mathrm{E}_{p_{\mathrm{tr}}(\boldsymbol{x},y)}[\ell(yg(\boldsymbol{x}))]\\
&= \pi_{\mathrm{Ptr}}\mathrm{E}_{\mathrm{Ptr}}[\ell(g(\boldsymbol{x}))] + \pi_{\mathrm{Ntr}}\mathrm{E}_{\mathrm{Ntr}}[\ell(-g(\boldsymbol{x}))]\\
&=: R^{\mathrm{tr}}_{\mathrm{PN}}(g), \qquad (1)
\end{aligned}$$

where $\mathrm{E}_{\mathrm{Ptr}}$ and $\mathrm{E}_{\mathrm{Ntr}}$ denote the expectations over $p_{\mathrm{tr}}(\boldsymbol{x} \mid y=+1)$ and $p_{\mathrm{tr}}(\boldsymbol{x} \mid y=-1)$, respectively. We refer to $R^{\mathrm{tr}}_{\mathrm{PN}}(g)$ as the positive-negative risk (the PN risk). From the definition of the marginal density $p_{\mathrm{tr}}(\boldsymbol{x})$, we have

$$\begin{aligned}
\mathrm{E}_{\mathrm{Utr}}[\ell(-g(\boldsymbol{x}))] &= \pi_{\mathrm{Ptr}}\,\mathrm{E}_{\mathrm{Ptr}}[\ell(-g(\boldsymbol{x}))]\\
&\quad + \pi_{\mathrm{Ntr}}\,\mathrm{E}_{\mathrm{Ntr}}[\ell(-g(\boldsymbol{x}))],
\end{aligned}$$

where $\mathrm{E}_{\mathrm{Utr}}$ denotes the expectation over $p_{\mathrm{tr}}(\boldsymbol{x})$. Arranging the above equation, we obtain

$$\begin{aligned}
\pi_{\mathrm{Ntr}}\,\mathrm{E}_{\mathrm{Ntr}}[\ell(-g(\boldsymbol{x}))] &= \mathrm{E}_{\mathrm{Utr}}[\ell(-g(\boldsymbol{x}))]\\
&\quad - \pi_{\mathrm{Ptr}}\,\mathrm{E}_{\mathrm{Ptr}}[\ell(-g(\boldsymbol{x}))].
\end{aligned}$$

Finally, plugging it into the second term of the PN risk in Eq. (1), the risk in PU classification (the PU risk) is given by

$$R^{\mathrm{tr}}_{\mathrm{PU}}(g) := \pi_{\mathrm{Ptr}}\,\mathrm{E}_{\mathrm{Ptr}}[\widetilde{\ell}(g(\boldsymbol{x})] + \mathrm{E}_{\mathrm{Utr}}[\ell(-g(\boldsymbol{x}))], \qquad (2)$$

where $\widetilde{\ell}(m) := \ell(m) - \ell(-m)$. In practice, we use the *empirical* PU risk by using only positive and unlabeled data from training distribution:

$$\widehat{R}^{\mathrm{tr}}_{\mathrm{PU}}(g) := \frac{\pi_{\mathrm{Ptr}}}{n_{\mathrm{Ptr}}}\sum_{i=1}^{n_{\mathrm{Ptr}}}\widetilde{\ell}(g(\boldsymbol{x}_i^{\mathrm{Ptr}})) + \frac{1}{n_{\mathrm{Utr}}}\sum_{k=1}^{n_{\mathrm{Utr}}}\ell(-g(\boldsymbol{x}_k^{\mathrm{Utr}})).$$

We obtain the learned classifier by minimizing the empirical PU risk with, e.g., the ℓ_2-regularizer.

2.3 Covariate Shift Adaptation by Importance Weighting

In this section, we review the importance-weighted risk minimization framework for covariate shift adaptation (Quioñero Candela et al. 2009).

For covariate shift adaptation, the use of the ordinary risk $R^{\mathrm{tr}}(g) := \mathrm{E}_{p_{\mathrm{tr}}(x,y)}[\ell(yg(\boldsymbol{x}))]$ often leads to an inaccurate classifier in practice.[1] Instead, we employ the importance-weighted risk:

$$R^{\mathrm{tr}}_{\mathrm{c}}(g) := \mathrm{E}_{p_{\mathrm{tr}}(\boldsymbol{x},y)}[\ell(yg(\boldsymbol{x}))w(\boldsymbol{x})],$$

[1]To be precise, if we use the correct model for estimating the input-output relation, the covariate shift does not matter. However, in practice, the model is often *misspecified*, i.e., the true function is not included in the model. Thus, the trained model does not work well on test data because the risk on training data, $R^{\mathrm{tr}}(g)$, is biased.

where $w(\boldsymbol{x})$ is the importance function defined as

$$w(\boldsymbol{x}) := \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}.$$

Under the covariate shift, we can show that the importance-weighted risk is equivalent to the risk on test data:

$$\begin{aligned} R_{\mathrm{c}}^{\mathrm{tr}}(g) &= \mathrm{E}_{p_{\mathrm{tr}}(\boldsymbol{x})}[\ell(yg(\boldsymbol{x}))w(\boldsymbol{x})p_{\mathrm{tr}}(y \mid \boldsymbol{x})] \quad (3)\\ &= \mathrm{E}_{p_{\mathrm{te}}(\boldsymbol{x})}[\ell(yg(\boldsymbol{x}))p_{\mathrm{te}}(y \mid \boldsymbol{x})] \quad (4)\\ &= R^{\mathrm{te}}(g), \end{aligned}$$

where $p_{\mathrm{tr}}(y \mid \boldsymbol{x}) = p_{\mathrm{te}}(y \mid \boldsymbol{x})$ and $w(\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x}) = p_{\mathrm{te}}(\boldsymbol{x})$ are used for obtaining Eq. (4) from Eq. (3). By the importance weighting, we obtain a classifier trying to minimize the risk on test distribution.

3 PU Classification under Covariate Shift

In this section, we first propose a risk function computed from PU data for covariate shift adaptation, and analyze the effect of dataset shift from the viewpoint of the proposed risk function. We also describe the practical implementation and the convergence rate of our method.

3.1 Proposed Approach

Let $R_{\mathrm{PU}}^{\mathrm{te}}$ be the PU risk on test distribution p_{te}. If the covariate shift does not occur, both $R_{\mathrm{PU}}^{\mathrm{tr}}$ and $R_{\mathrm{PU}}^{\mathrm{te}}$ are equivalent. This means that the empirical PU risk on training data, $\widehat{R}_{\mathrm{PU}}^{\mathrm{tr}}$, is an estimator unbiased to the PU risk on test data $R_{\mathrm{PU}}^{\mathrm{te}}$, i.e., $\mathrm{E}[\widehat{R}_{\mathrm{PU}}^{\mathrm{tr}}(g)] = R_{\mathrm{PU}}^{\mathrm{tr}}(g) = R_{\mathrm{PU}}^{\mathrm{te}}(g)$.

Under the covariate shift, a use of $\widehat{R}_{\mathrm{PU}}^{\mathrm{tr}}$ does not mean the risk estimator unbiased to $R_{\mathrm{PU}}^{\mathrm{te}}$, i.e., $R_{\mathrm{PU}}^{\mathrm{tr}}(g) \neq R_{\mathrm{PU}}^{\mathrm{te}}(g)$ due to input distribution shift $p_{\mathrm{tr}}(\boldsymbol{x}) \neq p_{\mathrm{te}}(\boldsymbol{x})$. To address this issue, we propose to use the following risk for covariate shift adaptation on PU learning, called the PUc risk, defined as

$$\begin{aligned} R_{\mathrm{PUc}}(g) := \pi_{\mathrm{Ptr}}\,\mathrm{E}_{\mathrm{Ptr}}[\widetilde{\ell}(g(\boldsymbol{x}))w(\boldsymbol{x})] \\ + \mathrm{E}_{\mathrm{Utr}}[\ell(-g(\boldsymbol{x}))w(\boldsymbol{x})]. \quad (5) \end{aligned}$$

The empirical PUc risk on training data is an unbiased estimator to the PU risk on test data, that is, $\mathrm{E}[\widehat{R}_{\mathrm{PUc}}(g)] = R_{\mathrm{PU}}^{\mathrm{te}}(g)$. Since the equivalence of the second term in Eq. (5) is obvious from $p_{\mathrm{tr}}(\boldsymbol{x})w(\boldsymbol{x}) = p_{\mathrm{te}}(\boldsymbol{x})$, we show that the first term in Eq. (5) is equivalent to the first term in Eq. (2):

$$\begin{aligned} \pi_{\mathrm{Ptr}}\mathrm{E}_{\mathrm{Ptr}}[\widetilde{\ell}(g(\boldsymbol{x}))w(\boldsymbol{x})] &= \mathrm{E}_{\mathrm{Utr}}[\widetilde{\ell}(g(\boldsymbol{x}))w(\boldsymbol{x})p_{\mathrm{tr}}(y{=}+1|\boldsymbol{x})]\\ &= \mathrm{E}_{\mathrm{Ute}}[\widetilde{\ell}(g(\boldsymbol{x}))p_{\mathrm{te}}(y=+1|\boldsymbol{x})]\\ &= \pi_{\mathrm{Pte}}\,\mathrm{E}_{\mathrm{Pte}}[\widetilde{\ell}(g(\boldsymbol{x}))], \end{aligned}$$

where $\pi_{\mathrm{Ptr}}p_{\mathrm{tr}}(\boldsymbol{x} \mid y = +1) = p_{\mathrm{tr}}(y = +1 \mid \boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})$ are used in the first line, $p_{\mathrm{tr}}(y \mid \boldsymbol{x}) = p_{\mathrm{te}}(y \mid \boldsymbol{x})$ and $p_{\mathrm{tr}}(\boldsymbol{x})w(\boldsymbol{x}) = p_{\mathrm{te}}(\boldsymbol{x})$ are used for obtaining the second line from the first line.

By replacing the expectations over $p_{\mathrm{tr}}(\boldsymbol{x} \mid y = +1)$ and $p_{\mathrm{te}}(\boldsymbol{x})$ with the corresponding sample averages, we obtain the empirical PUc risk by

$$\begin{aligned} \widehat{R}_{\mathrm{PUc}}(g) := \frac{\pi_{\mathrm{Ptr}}}{n_{\mathrm{Ptr}}}\sum_{i=1}^{n_{\mathrm{Ptr}}}\widetilde{\ell}(g(\boldsymbol{x}_i^{\mathrm{Ptr}}))w(\boldsymbol{x}_i^{\mathrm{Ptr}}) \\ + \frac{1}{n_{\mathrm{Utr}}}\sum_{k=1}^{n_{\mathrm{Utr}}}\ell(-g(\boldsymbol{x}_k^{\mathrm{Utr}}))w(\boldsymbol{x}_k^{\mathrm{Utr}}). \quad (6) \end{aligned}$$

An advantage of the PUc risk is that positive samples drawn from *test* conditional distribution are not required to obtain the PUc risk estimator, unlike the PU risk estimator on test data. This property will be highly useful in practical applications under the covariate shift. In such a case, to apply the PU classification method, we need to collect positive samples from a new environment. In contrast, the PUc classification method can reuse the positive samples obtained from a previous environment. An extra cost for the PUc classification method is collecting unlabeled test data, but the cost is relatively low compared with collecting positive data in addition to unlabeled data from the test environment.

Discussion: In the PUc risk in Eq. (5), the weighted average over unlabeled *training* data can be replaced with non-weighted average over unlabeled *test* data. The use of non-weighted average over unlabeled test data will give a stable estimates of the risk since importance-weighting reduces the number of samples for estimation.[2] However, in our preliminary experiments, we found that the use of weighted average over unlabeled training data achieved more accurate classification performance. A possible reason is that since an estimated importance function is not accurate enough, the weighted loss computed by positive data is biased by the estimated importance. Thus, the loss computed by unlabeled data is better to be biased in the same direction of the weighted loss for the positive class.

3.2 Effect of Covariate Shift

Here, we discuss the effect of the covariate shift on PU learning. In particular, we consider two situations called positive-only and negative-only shifts.

Positive-Only Shift Dataset shift occurs in a domain of positive class but not negative class, i.e.,

$$\begin{aligned} p_{\mathrm{tr}}(\boldsymbol{x} \mid y = +1) &\neq p_{\mathrm{te}}(\boldsymbol{x} \mid y = +1),\\ p_{\mathrm{tr}}(\boldsymbol{x} \mid y = -1) &= p_{\mathrm{te}}(\boldsymbol{x} \mid y = -1). \end{aligned}$$

In the positive-only shift, unless the ratio of class-priors of training and test phases satisfy

$$\frac{\pi_{\mathrm{Ptr}}}{\pi_{\mathrm{Pte}}} = \frac{p_{\mathrm{te}}(\boldsymbol{x} \mid y = +1)}{p_{\mathrm{tr}}(\boldsymbol{x} \mid y = +1)}$$

[2] For instance, suppose 20% of samples are important and 80% of samples are less important from the viewpoint of values of an importance function. In this case, the method ignores 80% of samples and trains a classifier by using only 20% of samples for covariate shift adaptation.

for all $\boldsymbol{x} \in \mathcal{D}_{\mathrm{P}}$, where $\mathcal{D}_{\mathrm{P}}$ is the support of class-conditional density for positive class, the importance function around the positive domain is not equal to one, i.e.,

$$\frac{p_{\mathrm{te}}(\boldsymbol{x}, y=+1)}{p_{\mathrm{tr}}(\boldsymbol{x}, y=+1)} \neq 1.$$

This shows that the weighting is necessary for the PUc risk; otherwise the first term in Eq. (5) is biased.

Negative-Only Shift We assume that data distribution satisfies the following condition:

$$p_{\mathrm{tr}}(\boldsymbol{x} \mid y=+1) = p_{\mathrm{te}}(\boldsymbol{x} \mid y=+1),$$
$$p_{\mathrm{tr}}(\boldsymbol{x} \mid y=-1) \neq p_{\mathrm{te}}(\boldsymbol{x} \mid y=-1).$$

In contrast to the positive-only shift, the weighting is not necessary in the negative-only shift. This is because i) we have

$$p_{\mathrm{te}}(\boldsymbol{x}, y=+1) = \pi_{\mathrm{Pte}} p_{\mathrm{tr}}(\boldsymbol{x} \mid y=+1),$$

and ii) the risk for negative data can be approximated by unlabeled *test* data instead of the weighted average over unlabeled training data, meaning that all we need is class-prior in test instead of training data. In summary, in the negative-only shift, even though dataset shift occurs, the importance-weighting is not necessary and the following risk for PU classification, called the PUc-te risk, can be used:

$$R_{\text{PUc-te}}(g) := \pi_{\mathrm{Pte}} \mathrm{E}_{\mathrm{Ptr}}[\widetilde{\ell}(g(\boldsymbol{x}))] + \mathrm{E}_{\mathrm{Ute}}[\ell(-g(\boldsymbol{x}))]. \quad (7)$$

Unlike the PUc risk, since the weighting is not used, the PUc-te risk is more stable than the PUc risk, similarly to the discussion in supervised learning (Quioñero Candela et al. 2009). Moreover, when applying PU classification on the negative-only shift situation, the positive data can be reused; a laborious labeling task is not necessary. The above observation coincides with an advantage of PU classification discussed in the literature (see, for example, du Plessis, Niu, and Sugiyama (2015)).

In Section 4.3, we experimentally validate this discussion.

Relation to Prior Probability Shift Here, we discuss the relation to the class-prior probability shift.

The prior probability shift is stated that class-conditional densities are fixed but class-prior probabilities are different between training and test phases:

$$p_{\mathrm{tr}}(\boldsymbol{x} \mid y) = p_{\mathrm{te}}(\boldsymbol{x} \mid y),$$
$$p_{\mathrm{tr}}(y) \neq p_{\mathrm{te}}(y).$$

Under the prior probability shift, the empirical risk on training data is not equal to the one on test data.

In supervised learning, the prior probability shift can be adapted by weighting errors of positive and negative classes. The PN risk for the prior probability shift, called the PNw risk, is equivalent to the test risk:

$$\begin{aligned} R^{\mathrm{tr}}_{\mathrm{PNw}}(g) &:= \pi_{\mathrm{Pte}} \mathrm{E}_{\mathrm{Ptr}}[\ell(g(\boldsymbol{x}))] + \pi_{\mathrm{Nte}} \mathrm{E}_{\mathrm{Ntr}}[\ell(-g(\boldsymbol{x}))] \\ &= \pi_{\mathrm{Pte}} \mathrm{E}_{\mathrm{Pte}}[\ell(g(\boldsymbol{x}))] + \pi_{\mathrm{Nte}} \mathrm{E}_{\mathrm{Nte}}[\ell(-g(\boldsymbol{x}))] \\ &= R^{\mathrm{te}}(g). \end{aligned}$$

On the other hand, the PU risk for the prior probability shift, called the PUw risk, can be obtained by

$$R^{\mathrm{tr}}_{\mathrm{PUw}}(g) := \pi_{\mathrm{Pte}} \mathrm{E}_{\mathrm{Ptr}}[\widetilde{\ell}(g(\boldsymbol{x}))] + \mathrm{E}_{\mathrm{Ute}}[\ell(-g(\boldsymbol{x}))], \quad (8)$$

where we used the following relation for derivation:

$$p_{\mathrm{te}}(\boldsymbol{x}) = \pi_{\mathrm{Pte}} p_{\mathrm{tr}}(\boldsymbol{x} \mid y=+1) + p_{\mathrm{te}}(\boldsymbol{x}, y=-1).$$

Thus, the prior probability shift is mitigated by replacing the class-prior and unlabeled data in a training phase with that in a test phase.

The PUw risk in Eq. (8) is equivalent to the PUc-te risk in Eq. (7). A difference is the assumption about the class-conditional density for the negative class, i.e, whether $p(\boldsymbol{x} \mid y=-1)$ is the same or not between training and test phases, implying a relation between the covariate shift and prior probability shift.

3.3 Implementation

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

As a classifier, we use the linear-in-parameter model:

$$g(\boldsymbol{x}) = \sum_{\ell=1}^{b} \beta_\ell \phi_\ell(\boldsymbol{x}) = \boldsymbol{\beta}^\top \boldsymbol{\phi}(\boldsymbol{x}),$$

where $\boldsymbol{\beta} := (\beta_1, \ldots, \beta_b)^\top$ is the vector of parameters, $\boldsymbol{\phi}(\boldsymbol{x}) := (\phi_1(\boldsymbol{x}), \ldots, \phi_b(\boldsymbol{x}))^\top$ is the vector of basis functions, and b is the number of basis functions. In Eq. (6), there are two unknown quantities: the class-prior of training data π_{Ptr} and the importance function $w(\boldsymbol{x})$. In our implementation, we estimate them by using existing methods.

To estimate the class-prior π_{Ptr}, we employ the method based on *kernel mean embedding* (Ramaswamy, Scott, and Tewari 2016).

For the importance function $w(\boldsymbol{x}) = p_{\mathrm{te}}(\boldsymbol{x})/p_{\mathrm{tr}}(\boldsymbol{x})$, we employ *direct* density-ratio estimation methods (Sugiyama, Suzuki, and Kanamori 2012), rather than separately estimating individual densities $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$. We adopt *relative unconstrained least-squares importance fitting* (RuLSIF) (Yamada et al. 2013), which addresses the importance function takes extremely larger values around a low-density region of training data (see Figure 1 in Yamada et al. (2013)). RuLSIF directly estimates the *α-relative density-ratio* of $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$:

$$w_\alpha(\boldsymbol{x}) := \frac{p_{\mathrm{te}}(\boldsymbol{x})}{(1-\alpha)p_{\mathrm{tr}}(\boldsymbol{x}) + \alpha p_{\mathrm{te}}(\boldsymbol{x})},$$

where $0 \leq \alpha \leq 1$ is the mixture rate. The α-relative density-ratio is always bounded by $1/\alpha$ when $\alpha > 0$, and it coincides with the plain importance function when $\alpha = 0$. The α controls efficiency and consistency (Yamada et al. 2013).

Replacing the class-prior and importance function by the estimates $\widehat{\pi}_{\mathrm{Ptr}}$ and $\widehat{w}_\alpha(\boldsymbol{x})$, we obtain a trained classifier $\widehat{g}$ by solving the following optimization problem:

$$\widehat{g} := \operatorname*{argmin}_g \ \widehat{R}_{\mathrm{PUc}}(g) + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta},$$

where $\lambda \geq 0$ is the regularization parameter. In our experiments and theoretical analysis, we use the squared loss function $\ell(m) = (1-m)^2/4$. An advantage of the squared loss is that the optimization problem becomes the convex optimization if we used the linear-in-parameter model (du Plessis, Niu, and Sugiyama 2015), i.e., we solve the following optimization problem:

$$\widehat{\boldsymbol{\beta}} := \operatorname*{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^b} \ -\widehat{\boldsymbol{q}}\boldsymbol{\beta} + \frac{1}{4}\boldsymbol{\beta}^\top \widehat{\boldsymbol{H}}\boldsymbol{\beta} + \frac{1}{2}\widehat{\boldsymbol{h}}^\top \boldsymbol{\beta} + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta},$$

where

$$\widehat{\boldsymbol{q}} := \frac{\widehat{\pi}_{\mathrm{Ptr}}}{n_{\mathrm{Ptr}}} \sum_{i=1}^{n_{\mathrm{Ptr}}} \widehat{w}_\alpha(\boldsymbol{x}_i^{\mathrm{Ptr}}) \boldsymbol{\phi}(\boldsymbol{x}_i^{\mathrm{Ptr}}),$$

$$\widehat{\boldsymbol{H}} := \frac{1}{n_{\mathrm{Utr}}} \sum_{k=1}^{n_{\mathrm{Utr}}} \widehat{w}_\alpha(\boldsymbol{x}_k^{\mathrm{Utr}}) \boldsymbol{\phi}(\boldsymbol{x}_k^{\mathrm{Utr}}) \boldsymbol{\phi}(\boldsymbol{x}_k^{\mathrm{Utr}})^\top,$$

$$\widehat{\boldsymbol{h}} := \frac{1}{n_{\mathrm{Utr}}} \sum_{k=1}^{n_{\mathrm{Utr}}} \widehat{w}_\alpha(\boldsymbol{x}_k^{\mathrm{Utr}}) \boldsymbol{\phi}(\boldsymbol{x}_k^{\mathrm{Utr}}).$$

In practice, we need to determine the hyperparameters such as the regularization parameter. To this end, we use the importance-weighted cross-validation (IWCV) (Sugiyama, Krauledat, and Müller 2007), which computes a score based on the importance-weighted risk. We compute the empirical PUc risk in Eq. (6) with the plain density-ratio estimator, i.e., $\widehat{w}_\alpha(\boldsymbol{x})$ with $\alpha = 0$, similarly to Yamada et al. (2013).

3.4 Convergence Analysis

Here, we present the convergence property of our proposed PUc risk estimator. Our proof follows du Plessis, Niu, and Sugiyama (2015) that is based on the perturbation analysis of optimization problems (Bonnans and Cominetti 1996; Bonnans and Shapiro 1998).

For the sake of simplicity, we focus on the squared loss and the linear-in-parameter model. As in du Plessis, Niu, and Sugiyama (2015), the use of the squared loss and linear-in-parameter model leads to convex optimization problems. Without loss of generality, we assume that the basis function satisfies $0 \leq \phi_\ell \leq 1$ for all $\ell = 1, \ldots, b$ and $\boldsymbol{x} \in \mathbb{R}^d$, and the basis functions are linearly independent over $p_{\mathrm{tr}}(\boldsymbol{x})$. Also, we assume that the parameter vector of the classifier is bounded, i.e., regularized as $\|\boldsymbol{\beta}\| \leq M$ for a constant $M > 0$. Additionally, we assume that the class-prior π_{Ptr} and the importance function w are known.[3]

The PUc and ℓ_2-regularized empirical PUc risk with the squared loss function are respectively defined as

$$\widehat{R}^\lambda_{\mathrm{PUc}}(\boldsymbol{\beta}) = -\widetilde{\boldsymbol{q}}^\top \boldsymbol{\beta} + \frac{1}{4}\boldsymbol{\beta}^\top \widetilde{\boldsymbol{H}}\boldsymbol{\beta} + \frac{1}{2}\widetilde{\boldsymbol{h}}^\top \boldsymbol{\beta} + 1 + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta},$$

$$R_{\mathrm{PUc}}(\boldsymbol{\beta}) = -\boldsymbol{q}^\top \boldsymbol{\beta} + \frac{1}{4}\boldsymbol{\beta}^\top \boldsymbol{H}\boldsymbol{\beta} + \frac{1}{2}\boldsymbol{h}^\top \boldsymbol{\beta} + 1,$$

[3]The convergence property of estimation methods for the class-prior and importance functions were studied in, e.g., Ramaswamy, Scott, and Tewari (2016), du Plessis, Niu, and Sugiyama (2017), and Yamada et al. (2013), respectively.

where

$$\boldsymbol{q} := \pi_{\mathrm{Ptr}} \int w(\boldsymbol{x}) \boldsymbol{\phi}(\boldsymbol{x}) p_{\mathrm{tr}}(\boldsymbol{x} \mid y = +1) \mathrm{d}\boldsymbol{x},$$

$$\boldsymbol{H} := \int w(\boldsymbol{x}) \boldsymbol{\phi}(\boldsymbol{x}) \boldsymbol{\phi}(\boldsymbol{x})^\top p_{\mathrm{tr}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x},$$

$$\boldsymbol{h} := \int w(\boldsymbol{x}) \boldsymbol{\phi}(\boldsymbol{x}) p_{\mathrm{tr}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x},$$

and $\widetilde{\boldsymbol{q}}$, $\widetilde{\boldsymbol{H}}$, and $\widetilde{\boldsymbol{h}}$ are its sample approximation.

Let $\boldsymbol{\beta}^* := \operatorname{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^b} R_{\mathrm{PUc}}(\boldsymbol{\beta})$ be the minimizer of the PUc risk with the squared-loss. We prove the following convergence results (the proof is in Appendix A):

Theorem 1. *As $n_{\mathrm{Ptr}}, n_{\mathrm{Utr}} \to \infty$, we have*

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\| = \mathcal{O}_p\Big(\frac{1}{\sqrt{n_{\mathrm{Ptr}}}} + \frac{1}{\sqrt{n_{\mathrm{Utr}}}}\Big),$$

$$\big|\widehat{R}^\lambda_{\mathrm{PUc}}(\widehat{\boldsymbol{\beta}}) - R_{\mathrm{PUc}}(\boldsymbol{\beta}^*)\big| = \mathcal{O}_p\Big(\frac{1}{\sqrt{n_{\mathrm{Ptr}}}} + \frac{1}{\sqrt{n_{\mathrm{Utr}}}}\Big)$$

provided that $\lambda = \mathcal{O}_p(1/\sqrt{n_{\mathrm{Ptr}}} + 1/\sqrt{n_{\mathrm{Utr}}})$.

Theorem 1 means that the estimated parameter of the classifier converges on the order of $\mathcal{O}_p(1/\sqrt{n_{\mathrm{Ptr}}} + 1/\sqrt{n_{\mathrm{Utr}}})$. Moreover, Theorem 1 implies that even if a model of classifier is misspecified, the parameter of the model converges to the optimal one in the prespecified function class.

4 Experiments

In this section, we show the effectiveness of the proposed PUc classification method.

4.1 Settings

In Section 4.2, we used the linear model $g(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + w_0$. In Sections 4.3 and 4.4, we used the linear-in-parameter model $g(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$ with the Gaussian kernel basis function $\phi_\ell(\boldsymbol{x}) = \exp(-\|\boldsymbol{x} - \boldsymbol{x}_\ell\|^2/(2\sigma^2))$, where $\sigma > 0$ was the bandwidth, the number of basis functions was set at $b = \min(200, n_{\mathrm{Ute}})$, and $\{\boldsymbol{x}_\ell\}_{\ell=1}^b$ was a set of samples selected randomly from $\{\boldsymbol{x}_k^{\mathrm{Ute}}\}_{k=1}^{n_{\mathrm{Ute}}}$. All hyper-parameters were determined by 5-fold IWCV described in Section 3.3. In all the experiments, the class-prior probabilities for training and test were set at $\pi_{\mathrm{Ptr}} = \pi_{\mathrm{Pte}} = 0.5$.

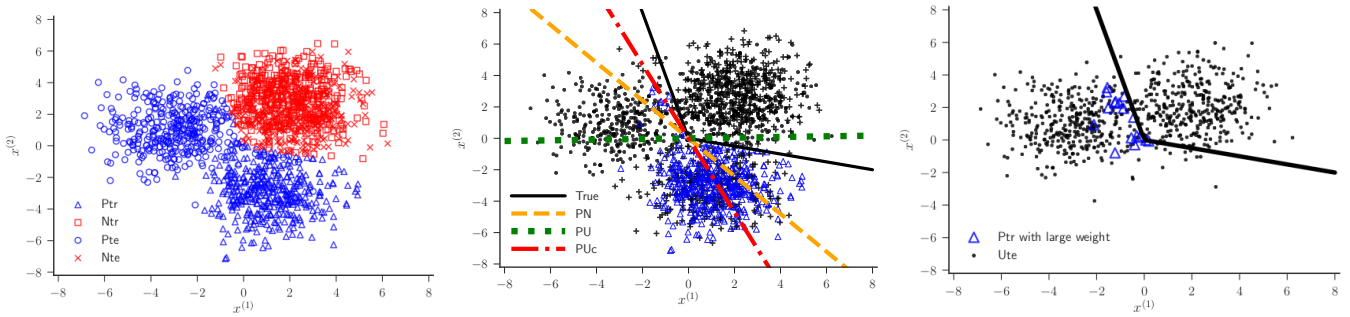
We evaluated the classification performance by misclassification rate, i.e., the empirical risk on test data with the zero-one loss function $\ell_{0\text{-}1}(m) := (1 + \mathrm{sign}(m))/2$.

4.2 Illustration

Firstly, we illustrate the performance of our proposed method on artificial data. The marginal distributions for training and test were respectively specified as

$$p_{\mathrm{tr}}(\boldsymbol{x}) = \frac{1}{2} \underbrace{N\Big(\begin{pmatrix}1\\-3\end{pmatrix}, 2\boldsymbol{I}_2\Big)}_{p_{\mathrm{tr}}(\boldsymbol{x}|y=+1)} + \frac{1}{2} \underbrace{N\Big(\begin{pmatrix}2\\3\end{pmatrix}, 2\boldsymbol{I}_2\Big)}_{p_{\mathrm{tr}}(\boldsymbol{x}|y=-1)},$$

$$p_{\mathrm{te}}(\boldsymbol{x}) = \frac{1}{2} \underbrace{N\Big(\begin{pmatrix}-3\\1\end{pmatrix}, 2\boldsymbol{I}_2\Big)}_{p_{\mathrm{te}}(\boldsymbol{x}|y=+1)} + \frac{1}{2} \underbrace{N\Big(\begin{pmatrix}2\\2\end{pmatrix}, 2\boldsymbol{I}_2\Big)}_{p_{\mathrm{te}}(\boldsymbol{x}|y=-1)},$$



(a) Generated data with its class labels (b) Learned decision boundaries (c) Positive training data having larger weights

Figure 1: (a) The positive training (△) and test (○) samples and negative training (□) and test (×) samples, where the mark of negative samples are hardly visible because of large overlap of training and test distributions of negative data. (b) The obtained decision boundaries from the PUc (- - -), plain PU (· · ·), PNc (- - -) methods, and the true decision boundary (—). The proposed PUc classification method can access the positive training data (△), unlabeled training data (+), and the unlabeled test data (·). The result shows that the decision boundary obtained by the PUc method is corrected compared with that of the plain PU method. (c) The positive training samples (△) with importance weight larger than two $\{\boldsymbol{x}_i^{\mathrm{Ptr}} \mid w(\boldsymbol{x}_i^{\mathrm{Ptr}}) > 2,\ \forall i = 1, \ldots, n_{\mathrm{Ptr}}\}$. The samples distributed around the right bottom region in Fig. 1(b) were almost ignored while the samples given large importance-weights contributed to training a classifier.

where $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the normal distribution with mean vector $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $\boldsymbol{I}_d$ is the d-dimensional identity matrix. The decision boundary was set at $g^*(\boldsymbol{x}) = \max(-4x^{(1)}, -x^{(1)}/4) - x^{(2)}$, where $x^{(t)}$ denotes the t-th element of the feature vector. Then, we collected $n_{\mathrm{Ptr}} = 400$ positive and $n_{\mathrm{Utr}} = 700$ unlabeled data from training distribution, and $n_{\mathrm{Ute}} = 700$ unlabeled data from test distribution. We used the plain PU and proposed PUc classification methods and plotted the decision boundaries. As reference, we also plotted the decision boundary obtained by the PNc method (supervised classification for covariate shift adaptation) with labeled samples of size $n_{\mathrm{P}} = n_{\mathrm{N}} = 400$.

Figure 1(a) depicts the positive training (△) and test (○) samples and negative training (□) and test (×) samples, where the mark of negative samples are hardly visible because of large overlap of training and test distributions of negative data. Figure 1(b) shows the obtained decision boundaries from the PUc (- - -), plain PU (· · ·), PNc (- - -) methods, and the true decision boundary (—). The proposed PUc classification method can access the positive training data (△), unlabeled training data (+), and the unlabeled test data (·). In this example, $\widetilde{g}_{\mathrm{tr}}(g) = -x^{(1)}/4 - x^{(2)}$ is an accurate linear decision function for the training data, and $\widetilde{g}_{\mathrm{te}}(g) = -4x^{(1)} - x^{(2)}$ for the test data (see Fig. 1(b)). The result shows that the boundary obtained by the PUc method was compensated compared with that of the plain PU method.

To see the effect of the importance weights, we plotted the positive training samples (△) with importance weight larger than two, $\{\boldsymbol{x}_i^{\mathrm{Ptr}} \mid w(\boldsymbol{x}_i^{\mathrm{Ptr}}) > 2,\ i = 1, \ldots, n_{\mathrm{Ptr}}\}$, in Fig. 1(c). While the samples distributed around the right bottom region in Fig. 1(b) were almost ignored, the positive samples near $\widetilde{g}_{\mathrm{te}}$ were contributed to training a classifier.

Table 1: Average with standard error of misclassification rates over 100 trials. The boldface indicates the best and comparable methods in terms of average misclassification rate according to the t-test at the significance level of 5%.

Dataset Shift	PUc	PU	PUc-te
positive-only	**31.68 (0.63)**	33.66 (0.59)	47.12 (0.32)
negative-only	30.66 (0.64)	32.89 (0.64)	**19.89 (0.31)**

4.3 Effect of Dataset Shift

In this section, we experimentally confirmed the effect of dataset shift discussed in Section 3.2. That is, under either the positive-only or negative-only shifts, we evaluated the following three methods: the proposed PUc method in Eq. (5), the PU method in Eq. (2), and the PUc-te method in Eq. (7). In the experiment, we drew sets of samples of size $n_{\mathrm{Ptr}} = 100$, $n_{\mathrm{Utr}} = 500$, and $n_{\mathrm{Ute}} = 500$.

We used the *MNIST* dataset (LeCun et al. 1998) and regarded the even numbers as the positive class, and the odd numbers as the negative class. To simulate the positive-only (resp. negative-only) shift, we changed the ratio of samples in subclasses of positive (resp. negative) class. In this experiment, with probability 0.1, positive training samples were drawn from images of "0" and "2", and with probability 0.9, positive training samples were drawn from images of "4", "6", and "8", while the rate was switched in positive *testing* samples i.e., 0.9 and 0.1 of positive testing samples. Similarly, with probabilities 0.1 and 0.9, negative training samples were drawn from images of "1" and "3", and images of "5", "7", and "9", respectively; the rate was also switched in test data.

Table 1 summarizes the results of dataset shift in the positive (resp. negative) class. In the positive-only shift, the

Table 2: Average with standard error of misclassification rates over 100 trials. The boldface indicates the best and comparable methods in terms of average misclassification rate according to the t-test at the significance level of 5%.

Dataset	d	PUc	PU	Reference	
				PNc	PN
banana	2	**24.6 (0.64)**	**26.4 (0.68)**	15.5 (0.68)	19.4 (0.68)
susy	18	**40.1 (0.56)**	**40.1 (0.57)**	31.8 (0.57)	31.7 (0.57)
ijcnn1	22	**46.0 (0.45)**	48.1 (0.29)	41.8 (0.29)	45.0 (0.29)
comp-rec	100	**21.4 (1.42)**	**24.7 (1.57)**	6.4 (1.57)	6.9 (1.57)
comp-sci	100	**22.0 (0.77)**	25.1 (0.89)	14.6 (0.89)	16.1 (0.89)
comp-talk	100	**15.9 (1.28)**	20.3 (1.58)	4.9 (1.58)	5.5 (1.58)
a9a	123	**21.8 (0.22)**	23.5 (0.27)	19.0 (0.27)	20.0 (0.27)

proposed PUc method attained lower misclassification rate than the other approaches. In particular, the PUc-te method performed poorly; this agrees with our discussion in Section 3.2. In contrast, in the negative-only shift, the PUc-te method achieved the lowest misclassification rate. This can be explained as follows: i) since the positive data do not change in the negative-only shift and unlabeled testing data absorbs the change, the PUc-te method properly handled the dataset shift, ii) the importance-weighting approach is known as making learning unstable while the PUc-te method does not require the importance weighting as discussed in Section 3.2.

4.4 Benchmark Data

Finally, we evaluated the performance of our proposed method on the benchmark datasets taken from the website of *LIBSVM* (Chang and Lin 2011), the *IDA Benchmark* (Rätsch, Onoda, and Müller 2001), and the 20 *Newsgroups* (Lang 1995).[4] In this experiment, we split the data set into training and test data based on the median of the feature vector. Specifically, we first constructed the set $\mathcal{C} := \{\boldsymbol{c}_i = \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|\}_{i=1}^n$, where $\|\cdot\|$ is the Euclidean norm and $\bar{\boldsymbol{x}} = (1/n)\sum_{i=1}^n \boldsymbol{x}_i$. We found the median $\boldsymbol{c}_{\mathrm{med}}$ from the set $\mathcal{C}$, and then split the set $\mathcal{C}$ into the first set whose elements were smaller than $\boldsymbol{c}_{\mathrm{med}}$ and the second set whose elements were larger than $\boldsymbol{c}_{\mathrm{med}}$. With probability 0.9 and 0.1, the samples whose indices were in the first set were chosen as training data and test data, respectively. In contrast, the samples whose indices were in the second set were chosen as training data and test data with probability 0.1 and 0.9, respectively.

We compared the proposed PUc classification method against the ordinary PU classification method. As reference, we also report the results obtained by supervised learning (PN) and that with importance-weighting (PNc). We drew sets of samples of size $n_{\mathrm{Ptr}} = 300$, $n_{\mathrm{Ntr}} = 300$, $n_{\mathrm{Utr}} = 700$, and $n_{\mathrm{Ute}} = 700$.

Table 2 summarizes the average with standard error of misclassification rates, showing that the PUc classification method achieved more accurate classification performance than the ordinary PU classification method on many datasets. The difference between the proposed PUc and ordinary PU classification methods was larger than that between the PN and PNc methods. One of the reasons is that since label information for negative class is not available in PU learning, the information of importance function would be highly useful under the covariate shift.

[4]For the 20 Newsgroups, we used a tiny version of the dataset available at https://cs.nyu.edu/~roweis/data.html.

5 Conclusions

In this paper, we considered classification from positive and unlabeled data (PU classification) under the covariate shift. Based on importance-weighted risk minimization, we proposed a PU classification method for covariate shift adaptation. Our analysis revealed that in which situations covariate shift adaptation is effective from both theoretical and empirical viewpoints. Furthermore, we derived the convergence rate of parameters of a classifier. Through numerical experiments, we demonstrated the effectiveness of the proposed method on several benchmark datasets.

Acknowledgments

TS was supported by JSPS KAKENHI 15J09111.

A Proof of Theorem 1

We first prove the following Lemma:

Lemma 2. *Let ϵ be the smallest eigenvalue of $\widetilde{\boldsymbol{H}}$. The second order growth condition holds*

$$R_{\mathrm{PUc}}(\boldsymbol{\beta}) \geq R_{\mathrm{PUc}}(\boldsymbol{\beta}^*) + \epsilon\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2^2.$$

Proof. Given the linearly independent basis functions over $p_{\mathrm{tr}}(\boldsymbol{x})$, $\widetilde{\boldsymbol{H}}$ is positive definite. Thus, $R_{\mathrm{PUc}}(\boldsymbol{\beta})$ is strongly convex with parameter at least ϵ. We then have

$$\begin{aligned} R_{\mathrm{PUc}}(\boldsymbol{\beta}) &\geq R_{\mathrm{PUc}}(\boldsymbol{\beta}^*) + \nabla R_{\mathrm{PUc}}(\boldsymbol{\beta}^*)^\top(\boldsymbol{\beta} - \boldsymbol{\beta}^*) \\ &\quad + \epsilon\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2^2 \\ &\geq R_{\mathrm{PUc}}(\boldsymbol{\beta}^*) + \epsilon\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2^2, \end{aligned}$$

where we used the optimality condition $\nabla R_{\mathrm{PUc}}(\boldsymbol{\beta}^*) = \boldsymbol{0}$ to obtain the second equation. □

Then, let us define a set of perturbation parameters as $\boldsymbol{u} := \{\boldsymbol{u_q} \in \mathbb{R}^b, \boldsymbol{U_H} \in \mathbb{R}^{b\times b}, \boldsymbol{u_h} \in \mathbb{R}^b\}$. The perturbed objective function and the solution are given by

$$\begin{aligned} R_{\mathrm{PUc}}(\boldsymbol{\beta}, \boldsymbol{u}) &:= -(\boldsymbol{q} + \boldsymbol{u_q})^\top\boldsymbol{\beta} + \frac{1}{4}\boldsymbol{\beta}^\top(\boldsymbol{H} + \boldsymbol{U_H})\boldsymbol{\beta} \\ &\quad + \frac{1}{2}(\boldsymbol{h} + \boldsymbol{u_h})^\top\boldsymbol{\beta} + 1, \\ \boldsymbol{\beta}(\boldsymbol{u}) &:= \operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^b} R_{\mathrm{PUc}}(\boldsymbol{\beta}, \boldsymbol{u}). \end{aligned}$$

Apparently, $R_{\mathrm{PUc}}(\boldsymbol{\beta}) = R_{\mathrm{PUc}}(\boldsymbol{\beta}, \boldsymbol{0})$. Then, we obtain the following lemma:

Lemma 3. *Given a sufficiently small neighborhood of $\boldsymbol{\beta}^*$, $R_{\mathrm{PUc}}(\cdot, \boldsymbol{u}) - R_{\mathrm{PUc}}(\cdot)$ is Lipschitz continuous modulus $\omega(\boldsymbol{u}) = \mathcal{O}(\|\boldsymbol{u_q}\|_2 + \|\boldsymbol{U_H}\|_{\mathrm{Fro}} + \|\boldsymbol{u_h}\|_2)$.*

Proof. Let $\mathcal{B}_\delta(\boldsymbol{\beta}^*) := \{\boldsymbol{\beta} \mid \|\boldsymbol{\beta} - \boldsymbol{\beta}\|_2 \leq \delta$ be the δ-ball of $\boldsymbol{\beta}^*$. For any $\boldsymbol{\beta} \in \mathcal{B}_\delta(\boldsymbol{\beta}^*)$, we can easily show $\|\boldsymbol{\beta}\|_2 \leq \|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2 + \|\boldsymbol{\beta}^*\|_2 \leq \delta + M$. In addition, we have

$$\Big\| \frac{\partial}{\partial \boldsymbol{\beta}} \Big(R_{\mathrm{PUc}}(\cdot, \boldsymbol{u}) - R_{\mathrm{PUc}}(\cdot) \Big) \Big\|_2 \leq \|\boldsymbol{u_q}\|_2 + \frac{\delta + M}{2} \|\boldsymbol{U_H}\|_{\mathrm{Fro}} + \frac{1}{2} \|\boldsymbol{u_h}\|_2,$$

meaning that $R_{\mathrm{PUc}}(\cdot, \boldsymbol{u}) - R_{\mathrm{PUc}}(\cdot)$ is Lipschitz continuous on $\mathcal{B}_\delta(\boldsymbol{\beta}^*)$ with a Lipschitz constant of order $\mathcal{O}(\|\boldsymbol{u_q}\|_2 + \|\boldsymbol{U_H}\|_{\mathrm{Fro}} + \|\boldsymbol{u_h}\|_2)$. □

Finally, we prove Theorem 1. According to the *central limit theorem*, we have $\|\boldsymbol{u_q}\|_2 = \mathcal{O}_p(1/\sqrt{n_{\mathrm{Ptr}}})$, $\|\boldsymbol{U_H}\|_{\mathrm{Fro}} = \mathcal{O}_p(1/\sqrt{n_{\mathrm{Utr}}})$, $\|\boldsymbol{u_h}\|_2 = \mathcal{O}_p(1/\sqrt{n_{\mathrm{Utr}}})$ as $n_{\mathrm{Ptr}}, n_{\mathrm{Utr}} \to \infty$. Thus, by using Lemma 2, Lemma 3, and Proposition 6.1 in Bonnans and Shapiro (1998), we have

$$\begin{aligned} \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 &\leq \epsilon^{-1} \omega(\boldsymbol{u}) \\ &= \mathcal{O}(\|\boldsymbol{u_q}|_2 + \|\boldsymbol{U_H}\|_{\mathrm{Fro}} + \|\boldsymbol{u_h}\|_2) \\ &= \mathcal{O}_p(1/\sqrt{n_{\mathrm{Ptr}}} + 1/\sqrt{n_{\mathrm{Utr}}}). \end{aligned}$$

This concludes the first half of Theorem 1.

Next, we prove the second half of the theorem. By using the triangle inequality, we have

$$\begin{aligned} &|\widehat{R}^{\lambda}_{\mathrm{PUc}}(\widehat{\boldsymbol{\beta}}) - R_{\mathrm{PUc}}(\boldsymbol{\beta}^*)| \\ &\quad \leq |\widehat{R}^{\lambda}_{\mathrm{PUc}}(\widehat{\boldsymbol{\beta}}) - \widehat{R}^{\lambda}_{\mathrm{PUc}}(\boldsymbol{\beta}^*)| + |\widehat{R}^{\lambda}_{\mathrm{PUc}}(\boldsymbol{\beta}^*) - R_{\mathrm{PUc}}(\boldsymbol{\beta}^*)|. \end{aligned}$$

We can show $|\widehat{R}^{\lambda}_{\mathrm{PUc}}(\widehat{\boldsymbol{\beta}}) - \widehat{R}^{\lambda}_{\mathrm{PUc}}(\boldsymbol{\beta}^*)| = \mathcal{O}(\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|) = \mathcal{O}_p(1/\sqrt{n_{\mathrm{Ptr}}} + 1/\sqrt{n_{\mathrm{Utr}}})$ and $|\widehat{R}^{\lambda}_{\mathrm{PUc}}(\boldsymbol{\beta}^*) - R_{\mathrm{PUc}}(\boldsymbol{\beta}^*)| = \mathcal{O}_p(1/\sqrt{n_{\mathrm{Ptr}}} + 1/\sqrt{n_{\mathrm{Utr}}})$. As a result, we conclude $|\widehat{R}^{\lambda}_{\mathrm{PUc}}(\widehat{\boldsymbol{\beta}}) - R_{\mathrm{PUc}}(\boldsymbol{\beta}^*)| = \mathcal{O}_p(1/\sqrt{n_{\mathrm{Ptr}}} + 1/\sqrt{n_{\mathrm{Utr}}})$.

References

Bonnans, J. F., and Cominetti, R. 1996. Perturbed optimization in Banach spaces I: A general theory based on a weak directional constraint qualificationn; II: A theory based on a strong directional qualification condition; III: Semiinfinite optimization. *SIAM Journal on Control and Optimization* 34(4):1151–1171, 1172–1189, and 1555–1567.

Bonnans, J. F., and Shapiro, A. 1998. Optimization problems with perturbations: A guided tour. *SIAM Review* 40(2):228–264.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

du Plessis, M. C.; Niu, G.; and Sugiyama, M. 2015. Convex formulation for learning from positive and unlabeled data. In *ICML*, volume 37, 1386–1394.

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

Elkan, C., and Noto, K. 2008. Learning classifiers from only positive and unlabeled data. In *ACM SIGKDD*, 213–220.

Gong, T.; Wang, G.; J.Ye; Xu, Z.; and Lin, M. 2018. Margin based PU learning. In *AAAI*.

Kiryo, R.; Niu, G.; du Plessis, M. C.; and Sugiyama, M. 2017. Positive-unlabeled learning with non-negative risk estimator. In *NIPS*, 1674–1684.

Lang, K. 1995. NewsWeeder: Learning to filter netnews. In *ICML*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Lee, W. S., and Liu, B. 2003. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, 448–455.

Letouzey, F.; Denis, F.; and Gilleron, R. 2000. Learning from positive and unlabeled examples. In *ALT*, 71–85.

Niu, G.; du Plessis, M. C.; Sakai, T.; Ma, Y.; and Sugiyama, M. 2016. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *NIPS*, 1199–1207.

Quioñero Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. D. 2009. *Dataset Shift in Machine Learning*. The MIT Press.

Ramaswamy, H. G.; Scott, C.; and Tewari, A. 2016. Mixture proportion estimation via kernel embedding of distributions. In *ICML*.

Rätsch, G.; Onoda, T.; and Müller, K.-R. 2001. Soft margins for adaboost. *Machine learning* 42(3):287–320.

Shimodaira, H. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90(2):227–244.

Sugiyama, M.; Krauledat, M.; and Müller, K.-R. 2007. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research* 8(May):985–1005.

Sugiyama, M.; Suzuki, T.; and Kanamori, T. 2012. *Density Ratio Estimation in Machine Learning*. Cambridge, UK: Cambridge University Press.

Xia, R.; Hu, X.; Lu, J.; Yang, J.; and Zong, C. 2013. Instance selection and instance weighting for cross-domain sentiment classification via PU learning. In *IJCAI*, 2176–2182.

Xu, Y.; Xu, C.; Xu, C.; and Tao, D. 2017. Multi-positive and unlabeled learning. In *IJCAI*, 3182–3188.

Yamada, M.; Suzuki, T.; Kanamori, T.; Hachiya, H.; and Sugiyama, M. 2013. Relative density-ratio estimation for robust distribution comparison. *Neural Computation* 25(5):1324–1370.

Yang, P.; Liu, W.; and Yang, J. 2017. Positive unlabeled learning via wrapper-based adaptive sampling. In *IJCAI*, 3273–3279.