Prior Shift Estimation for Positive Unlabeled Data Through the Lens of Kernel Embedding

Jan Mielniczuk [1,3]
[1]Polish Academy of Sciences, Warsaw, Poland

Wojciech Rejchel [2]
[2]Nicolaus Copernicus University, Toruń, Poland

Paweł Teisseyre[1,3]
[3]Warsaw University of Technology, Warsaw, Poland

Abstract

We study estimation of a class prior for unlabeled target samples which possibly differs from that of source population. Moreover, it is assumed that the source data is partially observable: only samples from the positive class and from the whole population are available (PU learning scenario). We introduce a novel direct estimator of the class prior which avoids estimation of posterior probabilities in both populations and has a simple geometric interpretation. It is based on a distribution matching technique together with kernel embedding in Reproducing Kernel Hilbert Space and is obtained as an explicit solution to an optimisation task. We establish its asymptotic consistency as well as an explicit non-asymptotic bound on its deviation from the unknown prior, which is calculable in practice. We study finite sample behaviour for synthetic and real data and show that the proposal works consistently on par or better than its competitors.

1 INTRODUCTION

Positive Unlabeled (PU) learning (Elkan and Noto, 2008; Bekker and Davis, 2020) is an active research topic that has attracted a lot of interest in the machine learning community in recent years due to a common occurrence of data which is only partially observable. The goal is to build a binary classification model based on training data that contains solely positive cases and unlabeled ones, which can be either positive or negative. Typical example is a scenario, when patients with a confirmed diagnosis of a disease are treated as positive cases, while patients with no diagnosis are considered as unlabeled observations, since this group may include both sick and healthy individuals. PU data occurs frequently in many fields, such as bioinformatics (Li et al., 2021), image and text classification (Li and Liu, 2003; Fung et al., 2006), and survey research (Sechidis et al., 2017).

Most state-of-the-art learning algorithms for PU data, such as nnPU (Kiryo et al., 2017), VAE-PU (Na et al., 2020) and others (Chen et al., 2020; Zhao et al., 2022; Luo et al., 2021) require knowledge of the class prior, i.e. the probability of the positive class. Knowledge of the class prior can be used to either obtain calculable representation of a risk function, or to modify a threshold of a classification rule learned on source data. Since the class prior is usually unknown, there is an important line of research aimed at developing methods for estimating it from PU data, see Jain et al. (2016); Ramaswamy et al. (2016); Bekker and Davis (2018) for representative examples. The task is non-trivial, because in the case of PU data we do not have direct access to negative observations, but only to an unlabeled sample which is a mixture of positive and negative observations. This implies that the prior is not identifiable in general and one needs to impose assumptions to ensure its uniqueness. Moreover, most of the existing methods assume that the class prior is the same for the source (training) data and the target (test) data. This assumption is not fulfilled in many situations. For example, imagine that the source data is collected in the period before the outbreak of an epidemic, where the percentage of people with the considered disease is small, while the target data is gathered during an epidemic, where the prevalence of the disease may be much higher (Roland et al., 2022). The source and target data may also be collected in different climatic zones, which naturally differ in the prevalence of diseases. In such situations, it is necessary to estimate the class prior probability not only for the source PU data but also for new unlabeled target data, for which

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

we only observe features whereas the labels remain unknown. Figure 1 illustrates the discussed situation. In the example, the source class prior $\pi = 0.2$ whereas the target class prior is $\pi' = 0.8$.

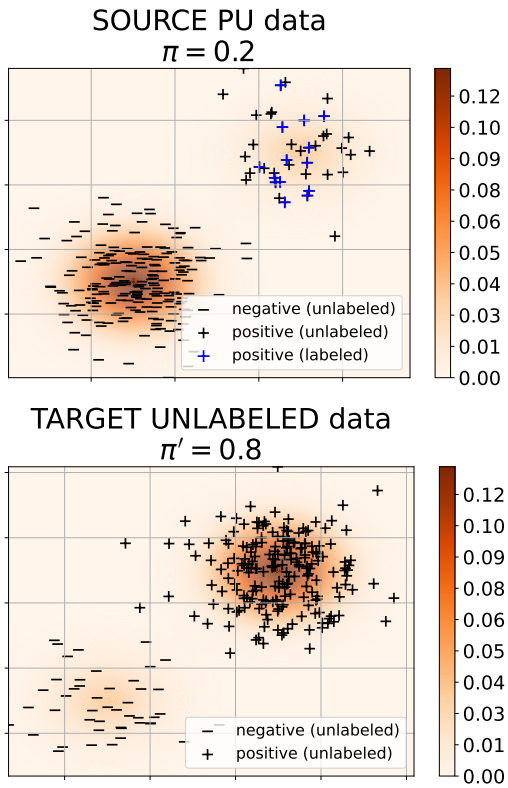

Figure 1: Label shift visualization for PU data. Source (training) data contains positive (blue) and unlabeled (grey) observations. Target (test) data contains only unlabeled observations. The class priors differ between source data ($\pi = 0.2$) and target data ($\pi' = 0.8$). The goal is to estimate the target class prior π' using source PU data and target unlabeled data.

The problem of inference under class prior shift, known in the literature as label shift, has been extensively studied and several methods have been developed to estimate the probability of the class prior for the target data as well as to modify the classifier to take the shift into account (Saerens et al., 2002; Lipton et al., 2018; Garg et al., 2020; Iyer et al., 2014; Vaz et al., 2019). We note that in business applications evaluation of proportion of each label on unlabeled data set (known as quantification task) is frequently even more needed than classification itself. This is particularly important for applications tracking trends (see Forman (2008) and González et al. (2017) for the review). However, methods developed for label shift problem require fully labeled source data and thus cannot be directly applied to the problem considered here. Whence the important endeavour is to estimate the target class prior directly, possibly avoiding label prediction for source and target samples. Surprisingly, despite its obvious importance, this problem has been rarely discussed. The only example up to now is Nakajima and Siguyama (2023) where an approach based on estimation of ratio of densities for positive and mixture distribution is proposed. Its authors, however, focus mainly on properties of the ensuing classifier, not on the properties of label shift estimator.

The main contribution of this work is the proposal of a new estimator of the target class prior π' called **TCPU** for which label prediction for target population is not necessary. Our approach is based on employing the distributions matching technique and the kernel method. The kernel method (Fukumizu et al., 2007; Gretton et al., 2012) is a powerful approach successfully applied for many problems of machine learning due to related universality property. For shifted prior estimation when the data is fully observable it has been used in Iyer et al. (2014). Here, we provide a rigorous treatment of proposed estimator, including its consistency, i.e. convergence in probability to the true target class prior. Even more importantly, we provide a *non-asymptotic* bound on the approximation error. Additionally, in the paper we show how to adapt the popular KM estimator (Ramaswamy et al., 2016), being a state-of-the-art class prior estimator for PU data, to the case of class prior shift. Our experiments, conducted on artificial and real data for different class prior shift schemes, confirm the effectiveness of proposed method TCPU.

2 LABEL SHIFT FOR PU LEARNING

We first introduce relevant notation. Let $X \in \mathcal{X}$ be a random variable corresponding to a feature vector, $Y \in \{-1, 1\}$ be a true class indicator and P_{XY} their joint distribution (called a source distribution). We consider a problem of modeling Positive Unlabeled data in case-control setting, which means that only samples coming from the positive class and samples coming from the overall population X are available. More formally, let P_X be the distribution of X and $P_+ = P_{X|Y=1}$, $P_- = P_{X|Y=-1}$ the distributions of samples from the positive class and the negative class, respectively. We denote by $X_1, \ldots, X_n$ independent samples generated according to P_X and $X_1^+, \ldots, X_m^+$ generated according to P_+.

Moreover, we consider the second vector (X', Y') such that its distribution $P'_{X'Y'}$ (called a target distribution) is a label shifted distribution of (X, Y), which means that the marginal distribution of Y' is different

from that of Y, i.e.

$$\pi' = P'(Y' = 1) \neq P(Y = 1) = \pi,$$

however, the covariate distributions in both the positive and the negative class remain the same:

$$P'_{X'|Y'=i} = P_{X|Y=i} \quad \text{for} \quad i = \pm 1. \tag{1}$$

Thus, we have that a distribution of X' satisfies $P'_{X'} = \pi' P_+ + (1-\pi')P_-$ and is different from $P_X = \pi P_+ + (1-\pi)P_-$.

Denote by $X'_1, \ldots, X'_{n'}$ independent samples generated from $P'_{X'}$. In the paper, we consider the problem of estimation of label shifted probability π'. Note that the problem is nontrivial as the class indicators corresponding to shifted samples are not available. However, we have at our disposal data from the positive class and unlabeled X observations corresponding to a different prior π.

We note that if π is known the distribution P_- is uniquely determined and the problem is well defined in general. When π is unknown, the specific assumptions are needed under which it is identifiable, e.g. an assumption that P_- is *not* a convex combination of P_+ and other probability measure (see e.g. Ramaswamy et al. (2016)).

Finally, it is worth mentioning that estimation of π' is crucial to define a classification rule for the target set which involves a modified threshold based on π' (Lipton et al., 2018):

$$P'(Y' = 1|X') > 0.5 \iff$$
$$P(Y = 1|X) > \frac{(1-\pi)\pi'}{(1-\pi')\pi + (1-\pi)\pi'}.$$

A natural approach is to train PU classifier, such as nnPU (Kiryo et al., 2017), on the source data, to estimate $P(Y = 1|X)$, and then apply the above decision rule.

3 CLASS PRIOR ESTIMATION FOR TARGET DATA

3.1 TCPU: a novel kernel-based estimator

In this section we introduce a novel method of estimating π', which will be called **TCPU** (a **T**arget **C**lass prior estimator for **P**ositive-**U**nlabled data under a label shift).

Let $K(\cdot,\cdot)$ be a kernel function (i.e. a symmetric, continuous and semi-positive function defined on $\mathcal{X} \times \mathcal{X}$) and let $\mathcal{H}$ be a Reproducing Kernel Hilbert Space (RHKS) induced by $K(\cdot,\cdot)$ (see e.g. Fukumizu et al. (2007)). We denote an associated kernel transform by $\phi(x) = K(x,\cdot) \in \mathcal{H}$. A scalar product in $\mathcal{H}$ of $\phi(x_1)$ and $\phi(x_2)$ is defined as $< \phi(x_1), \phi(x_2) >_{\mathcal{H}} = K(x_1, x_2)$ for $x_1, x_2 \in \mathcal{X}$ and is naturally extended for general elements of $\mathcal{H}$. Next, let $\Phi(P_X) = \mathbb{E}\phi(X) = \int_{\mathcal{X}} \phi(s)\, P_X(ds) \in \mathcal{H}$ be a mean functional of P_X with a norm $||\Phi(P_X)||^2_{\mathcal{H}} = \mathbb{E}K(X_1, X_2)$, where X_1 and X_2 are two independent random vectors following a distribution P_X. The mean functionals $\Phi(P_+)$ and $\Phi(P'_{X'})$ are defined analogously. Recall that when kernel is universal we have that $\Phi(P) = \Phi(Q)$ is equivalent to the fact that distributions P and Q coincide (Fukumizu et al., 2007).

Let $\hat{P}_X$, $\hat{P}_+$ and $\hat{P}'_{X'}$ be empirical distributions corresponding to observable samples. In the following we will omit indices X and X' in P_X and $P'_{X'}$, respectively, and the same convention is applied to their empirical counterparts. We note that due to the fact that $\hat{P}$ is a discrete distribution with mass n^{-1} at each observation X_i we have that $\Phi(\hat{P}) = n^{-1}\sum_{i=1}^n \phi(X_i)$ and analogous representations hold for $\Phi(\hat{P}_+)$ and $\Phi(\hat{P}')$.

In the above setup we have that

$$\begin{aligned}(1-\pi')(P - \pi P_+) &= (1-\pi')(1-\pi)P_- \\ &= (1-\pi)(P' - \pi' P_+).\end{aligned}$$

Therefore, in order to determine π', it is natural to substitute γ for π' and minimize the following objective function

$$\mathcal{L}(\gamma) = ||(1-\gamma)A - (1-\pi)B(\gamma)||^2_{\mathcal{H}}, \tag{2}$$

$A = [\Phi(P) - \pi\Phi(P_+)]$ and $B(\gamma) = [\Phi(P') - \gamma\Phi(P_+)]$.

Lemma 1. *Suppose that a kernel K is universal, $P_+ \neq P_-$ and $\pi < 1$. Then π' is unique minimizer of $\mathcal{L}(\gamma)$.*

Proof. Denote by $Crit(\gamma)$ the function given by

$$Crit(\gamma) = |\pi' - \gamma| \times (1-\pi)||\Phi(P_-) - \Phi(P_+)||_{\mathcal{H}}. \tag{3}$$

Then by noting that

$$\Phi(P) - \pi\Phi(P_+) = (1-\pi)\Phi(P_-),$$
$$\Phi(P') - \gamma\Phi(P_+) = (\pi' - \gamma)\Phi(P_+) + (1-\pi')\Phi(P_-)$$

we have that $\mathcal{L}(\gamma) = [Crit(\gamma)]^2$ and the minimiser of $Crit(\gamma)$ is obviously π'. □

In the proposed method, we consider the empirical version of (2) given as

$$\widehat{\mathcal{L}}(\gamma) = ||(1-\gamma)\widehat{A} - (1-\pi)\widehat{B}(\gamma)||^2_{\mathcal{H}}, \tag{4}$$

$\widehat{A} = [\Phi(\hat{P}) - \pi\Phi(\hat{P}_+)]$ and $\widehat{B}(\gamma) = [\Phi(\hat{P}') - \gamma\Phi(\hat{P}_+)]$

and the estimator of π' is defined as its minimizer

$$\widehat{\pi}' = \text{argmin}_\gamma \widehat{\mathcal{L}}(\gamma). \tag{5}$$

The estimator defined above will be called **TCPU** further on. Figure 2 illustrates the behavior of the objective function and TCPU estimator.

For theoretical results it will be assumed that π is known. This assumption is plausible when the large data base corresponding to source distribution is available.

In the following lemma we show that the proposed estimator can be explicitly calculated using a simple algebraic formula.

Lemma 2. *Let $\widehat{\pi}'$ be defined by (5). Then we have*

$$\widehat{\pi}' = \frac{< \Phi(\hat{P}) - \Phi(\hat{P}_+), \Delta >_{\mathcal{H}}}{||\Phi(\hat{P}) - \Phi(\hat{P}_+)||^2_{\mathcal{H}}} = \tag{6}$$
$$1 - \frac{(1-\pi) < \Phi(\hat{P}) - \Phi(\hat{P}_+), \Phi(\hat{P}') - \Phi(\hat{P}_+) >_{\mathcal{H}}}{||\Phi(\hat{P}) - \Phi(\hat{P}_+)||^2_{\mathcal{H}}},$$

where $\Delta = \Phi(\hat{P}) - \pi\Phi(\hat{P}_+) - (1-\pi)\Phi(\hat{P}')$.

Remark 1. *In view of the first equality in (6), $\widehat{\pi}'$ has a simple geometric interpretation: it is a coefficient of projection of a function Δ on $\Phi(\hat{P}) - \Phi(\hat{P}_+)$ in $\mathcal{H}$.*

Proof. Simple calculations show that the expression under the norm in (4) equals

$$-\gamma[\Phi(\hat{P}) - \Phi(\hat{P}_+)] + \Phi(\hat{P}) - \pi\Phi(\hat{P}_+) - (1-\pi)\Phi(\hat{P}')$$
$$= -\gamma[\Phi(\hat{P}) - \Phi(\hat{P}_+)] + \Delta.$$

Thus, the squared norm equals

$$\gamma^2||\Phi(\hat{P}) - \Phi(\hat{P}_+))||^2_{\mathcal{H}} +$$
$$2\gamma < \Phi(\hat{P}_+) - \Phi(\hat{P}), \Delta >_{\mathcal{H}} + ||\Delta||^2_{\mathcal{H}}$$

and the first form of $\widehat{\pi}'$ follows by calculating a minimizer of the above function. The second formula follows by simple algebraic manipulations. □

The kernel approach is a core of the Maximum Mean Discrepancy (MMD) method which matches the distributions based on the features in RHKS induced by a kernel K. It is frequently used in machine learning and statistics to compare a data distribution with a specific distribution, in a two-sample problem and covariate shift detection (see Gretton et al. (2012)). It has been also applied for the label shift problem in the classical framework. Namely, for the case when distribution P_{XY} is observable the approach relies on the equality (Zhang et al., 2014)

$$\pi' = \tag{7}$$
$$\text{argmin}_{\lambda\in[0,1]}||\Phi(P_{X'}) - [\lambda\Phi(P_+) + (1-\lambda)\Phi(P_-)]||^2_{\mathcal{H}}$$

(see also Iyer et al. (2014) and Dussap et al. (2023)). In the considered scenario, a direct application of (7) to estimate π' is clearly infeasible, as observations from the negative class are not available. The estimator (5) can be considered as a modification of the MMD approach applied for a label shift in PU setting.

3.2 Asymptotic consistency and non-asymptotic error bounds for the proposed estimator

We let $N = \min(n, m, n')$. In the next result we establish asymptotic and non-asymptotic error bounds for $\widehat{\pi}'$. In its proof we also state conditions, which guarantee that $||\Phi(\hat{P}) - \Phi(\hat{P}_+)||_{\mathcal{H}} > 0$, which is implicitly assumed in (6). The proofs of all results below are given in the supplement.

Theorem 1. *Suppose that a kernel K is universal, $P_+ \neq P_-$ and $\pi < 1$.*
(i) Moreover, assume $\mathbb{E}K(X,X) < \infty$ for $X \sim P$ and analogous conditions hold for $X^+ \sim P_{X|Y=1}$ and $X' \sim P'$. Then for $N \to \infty$ we have $\hat{\pi}' \to \pi'$ in probability.
(ii) Assume that $M = \sup_x K(x,x) < \infty$. Moreover, fix $\alpha \in (0,1)$ and $\delta \leq \exp(-(\sqrt{2}+1)^2/2)$ and let

$$N \geq \frac{16M\log(1/\delta)}{(1-\alpha)^2(1-\pi)^2||\Phi(P_-) - \Phi(P_+)||^2_{\mathcal{H}}}. \tag{8}$$

Then we have

$$P\left(|\widehat{\pi}' - \pi'| \leq \frac{4\sqrt{\frac{M}{N}\log(1/\delta)}}{\alpha(1-\pi)||\Phi(P_-) - \Phi(P_+)||_{\mathcal{H}}}\right) \geq 1-3\delta. \tag{9}$$

We note that consistency of $\widehat{\pi}'$ is proved in Theorem 1(i) under weak conditions. Indeed, dropping the conditions $P_+ \neq P_-$ and $\pi \neq 1$ makes the problem ill-posed. Finally, the restrictions imposed on a kernel are satisfied, for instance, for a gaussian kernel.

The claim of Theorem 1 (ii) is stronger (it is a nonasymptotic result implying asymptotic consistency), so it needs more restrictive assumptions as well. However, these conditions are reasonable as in (i). For instance, a gaussian kernel satisfies the assumption in (ii) with $M = 1$. The dependence of an error bound on $N, \delta, \pi, ||\Phi(P_-) - \Phi(P_+)||_{\mathcal{H}}$ is stated explicitly in (9).

Theorem 2. *Assume that $M = \sup_x K(x,x) < \infty$. Then for any $\delta \leq \exp(-(\sqrt{2}+1)^2/2)$ we have that*

$$P\left(|\widehat{\pi}' - \pi'| \leq \frac{4\sqrt{\frac{M}{N}\log(1/\delta)}}{||\Phi(\hat{P}) - \Phi(\hat{P}_+)||_{\mathcal{H}}}\right) \geq 1-3\delta. \tag{10}$$

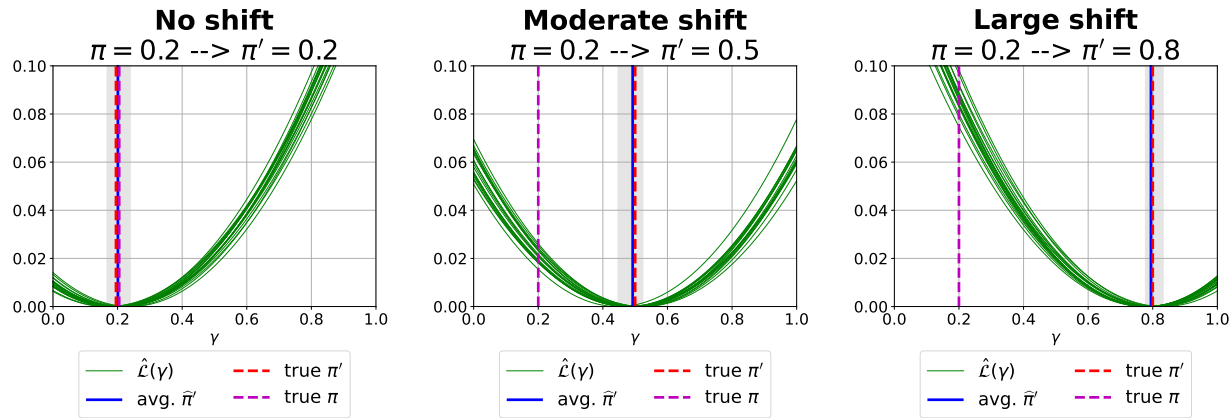

Figure 2: Visualization of the objective function behavior for $\pi = 0.2$. Three cases are considered: $\pi' = 0.2$ (no shift), $\pi' = 0.5$ (moderate shift) and $\pi' = 0.8$ (large shift). The TCPU estimator is defined as $\widehat{\pi}' = \arg\min_\gamma \hat{\mathcal{L}}(\gamma)$. The grey area indicates the range of estimator's values for 20 runs.

We stress the differences between Theorems 2 and 1. Probability inequality (10) does not require assumption (8) and it yields a bound on an estimation error which depends on $||\Phi(\hat{P}) - \Phi(\hat{P}_+)||_{\mathcal{H}}$. Notice that this bound can be calculated in practice. On the other hand, the error bound in (9) is nonrandom and explicitly establishes the dependence on π and a distance between P_- and P_+.

Till now we have assumed that source prior π is known. In the experiments we estimate π using a well-known KM2 method (Ramaswamy et al., 2016) described below and then plug-in it into (4). We stress that this requires additional assumptions implying that π is identifiable, which are imposed for all PU-based approaches. The most common one is that the distribution of negative samples is *not* a mixture of distribution of positive samples and some other probability distribution (Ramaswamy et al., 2016; Blanchard et al., 2010).

4 RELATED WORK

4.1 Method DRPU

The most related method is DRPU (Nakajima and Siguyama, 2023), which in the label-shift setting considered here, constructs empirical Bayes classifier for samples from label shifted population. Other methods consider similar but different scenarios such as PU covariate shift (Sakai and Shimizu, 2019) and positive data shift (Hammoudeh and Lowd, 2020). DRPU involves estimator of ratio r of densities of distributions P_+ and P for training data and of both prior probabilities π and π'. Estimator $\hat{r}$ of r is a minimiser of expected Bregman divergence functional (cf Section 2.5 in Nakajima and Siguyama (2023)) and both prior estimators are based on an observation (cf Blanchard et al. (2010)) that π and π' can be recovered in PU setting by minimising $P(A)/P_+(A)$ (respectively $P'(A)/P_+(A)$) over all sets A for which $P_+(A) > 0$. In Nakajima and Siguyama (2023), ratio $\hat{P}'(A)/\hat{P}_+(A)$ is minimised over sets being the level sets of $\hat{r}$. The estimator based on the above method will be denoted simply as **DRPU**.

4.2 Adapting the KM method to label shift

As a benchmark, we also introduce a modification of KM2 estimator (Ramaswamy et al., 2016) for label-shift PU setting. The KM2 estimator is based on the observation that distribution of negative class P_- can be written as $P_- = \lambda P + (1-\lambda)P_+$, where $\lambda = 1/(1-\pi)$ and thus, analogously to (7), estimator of λ can be constructed by projecting $\gamma\phi(P) + (1-\gamma)\phi(P_+)$ on a convex $\mathcal{C}$ hull of $\phi(X_1), \ldots, \phi(X_n), \phi(X_1^+), \ldots, \phi(X_m^+)$ and defining $\hat{\lambda}$ as γ yielding the smallest distance. This leads to two estimators of π, KM1 and KM2 investigated in Ramaswamy et al. (2016). As unavailable positive samples for test population have the same distribution as positive observations from training distribution, we can apply approach from Ramaswamy et al. (2016) to samples $X'_1, \ldots, X'_{n'}, X_1^+, \ldots, X_m^+$ and obtain KM2 estimator of π' also investigated below. The adaptation will be called **KM2-LS** in the following. Note that as KM2-LS does not use available observations from P at all, it is not expected to work well; this observation will be confirmed in our experiments.

5 EXPERIMENTS

5.1 Methods

We empirically evaluate the effectiveness of TCPU [1] to recover the true target class prior π'. As baselines we used DRPU (Nakajima and Siguyama, 2023) and KM2-LS methods described in Section 4. In the case of DRPU, we utilised the implementation made publicly available by the authors. In the case of KM2-LS, we applied the code of the standard KM2 method (Ramaswamy et al., 2016) and adopted it to our setting as described in Section 4. Note that TCPU requires knowledge of π. Since typically π remains unknown, we estimate it using KM2 estimator (Ramaswamy et al., 2016). Importantly, the KM2-LS method does not require the π estimation. The DRPU method is the only one that requires learning a parametric model. As in the original work, we considered Multi-Layer Perceptron (MLP) to this end. Technical details about the model used in DRPU and the selection of hyperparameters are discussed in the supplement. For TCPU, we used Gaussian kernel $K(x, y) = \exp(-\tau||x - y||^2)$ with the default value of parameter $\tau = 1/p$, where p is the number of features.

5.2 Datasets

The experiments were conducted on 11 datasets, including one synthetic dataset, 3 image datasets: CIFAR-10, MNIST, and FASHION (Paszke et al., 2019) and 7 tabular datasets from the UCI repository: Diabetes, Spambase, Segment, Waveform, Vehicle, Yeast and Banknote. For the synthetic dataset, negative observations are generated from a 10-dimensional normal distribution $N(0, I)$, and positive observations are generated from $N(a, I)$, where $a = (1, \ldots, 1)$. The characteristics of the UCI and image datasets are provided in Table 1. Tabular datasets with multiple classes were transformed into binary classification datasets, where the most common class is treated as the positive class, and the remaining classes are combined into the negative class. For image datasets, the binary class variable is defined according to the specific dataset, following methods used in other PU learning papers (Kiryo et al., 2017; Gong et al., 2021; Nakajima and Siguyama, 2023). For MNIST, even digits form the positive class, and odd digits form the negative class. In CIFAR-10, vehicles form the positive class, and animals form the negative class. For FASHION, clothing items worn on the upper body are marked as positive cases, and the remaining items are assigned to the negative class. For image data, we use a pre-trained deep neural network, ResNet18, to extract the feature vector. For each image, the feature vector, with a dimension of 512, is the output of the average pooling layer. From the extracted 512-dimensional feature vector, we select the 30 most correlated features with the class variable to reduce the dimensionality of the problem.

[1] The source code: `https://github.com/teisseyrep/TCPU`

Table 1: Statistics of the considered data sets.

Dataset	n	p	$P(Y=1)$	type
Diabetes	768	8	0.35	tab
Spambase	4601	57	0.39	tab
Segment	2310	19	0.14	tab
Waveform	5000	40	0.34	tab
Yeast	1484	8	0.31	tab
Vehicle	846	18	0.26	tab
Banknote	1347	4	0.44	tab
CIFAR10	50000	-	0.4	img
MNIST	60000	-	0.49	img
Fashion	60000	-	0.5	img

5.3 Experimental settings

First, each dataset is split into a source and a target dataset in equal proportions. For image data, we use the splits defined in the PyTorch library (Paszke et al., 2019). For synthetic datasets, we generate observations for fixed values of π and π'. For real datasets, we simulate a label shift scenario using the downsampling technique. Specifically, we randomly remove observations from one of the classes in both the source and target datasets to control the class priors π and π', respectively. Finally, based on the source data, we artificially create a PU dataset by selecting some positive observations for the labeled subset, while the unlabeled subset consists of a mixture of positive and negative observations. We follow a known procedure to control the size of the source data and the labeling frequency c, which represents the percentage of labeled observations among all positive observations. The details of the procedure are provided in the supplement.

In the experiments, we consider $c = 0.25, 0.5$ for sythetic and image datasets and $c = 0.5$ for UCI datasets. The entire target dataset is treated as unlabeled. The considered methods take as input the source PU data and unlabeled target data. For each method, we calculate the absolute estimation error $|\pi' - \widehat{\pi}'|$, where $\widehat{\pi}'$ is the estimator returned by the method. We perform 20 repetitions of the above procedure and analyze the distributions of the errors as well as the distributions of the estimators themselves.

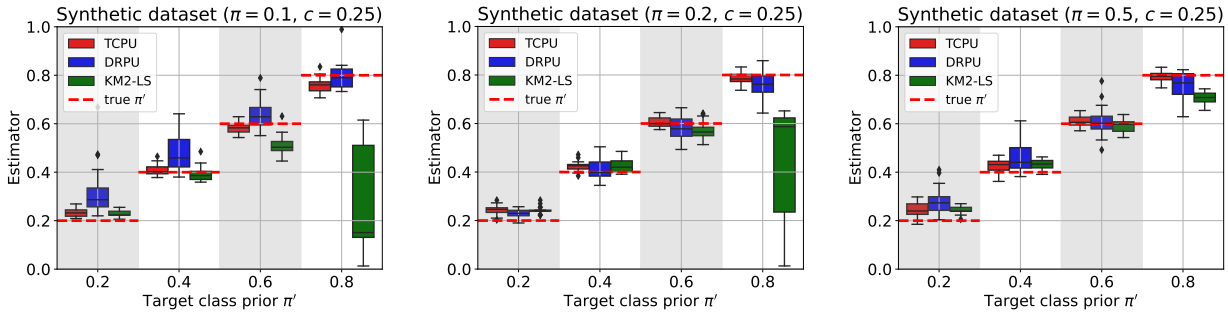

Figure 3: Distribution of estimators (red line indicates the true π'). Size of the source data and the target data is 2000.

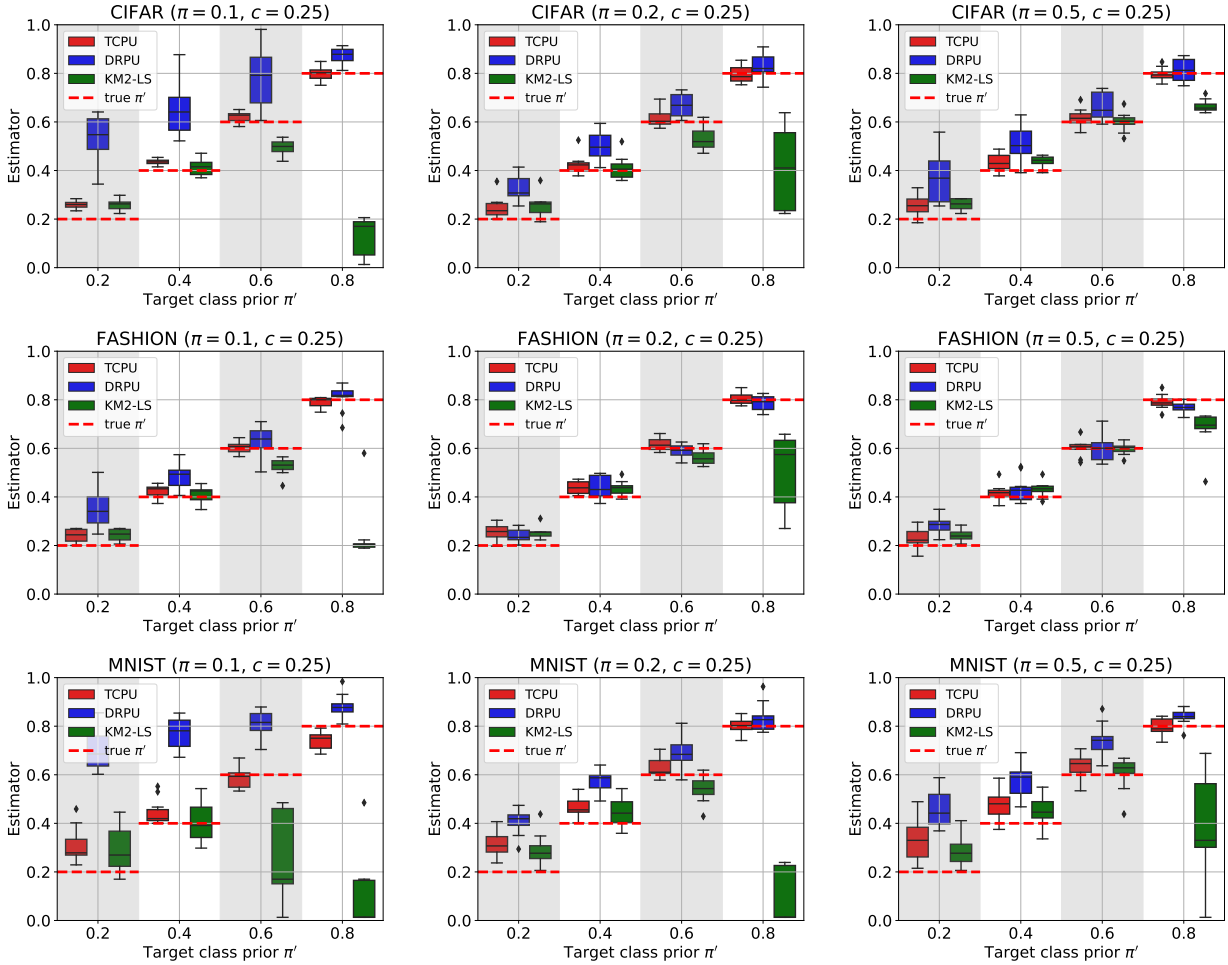

Figure 4: Distribution of estimators (red line indicates the true π') for image datasets for $c = 0.25$.

5.4 Discussion

In the main part of the paper we present selected results, while the full results can be found in the supplement.

Figures 3-4 in the main paper show the distributions of estimators for synthetic and image datasets when $c = 0.25$. In the supplement, Figures 7-10 show the results for other parameter settings, whereas Tables 3-6 present the mean absolute value estimation errors and their standard deviations for image and UCI datasets. Since our primary interest lies in analyzing estimation errors for small or moderate data samples, we exhibit the boxplots for the case where the source and target

datasets consist of randomly chosen samples of 2000 observations. When the total number of observations in the original dataset is less than 4000, we split the data into source and target datasets in equal proportions.

Experiments indicate that the values of π and π' significantly influence the quality of the π' estimation, with the impact differing across methods. The boxplots clearly indicate that KM2-LS significantly underestimates the true value of π' when π' is large. This effect becomes pronounced for low c and small π values, which is often the case in many practical applications. A similar problem with the KM2-LS method occurs for UCI datasets, where we observe very large estimation errors for $\pi = 0.8$ (see Tables 5 and 6). Importantly, in the discussed situation, the proposed TCPU performs much better and allows for more accurate estimation of π', regardless of the values of π and c. The performance of DRPU depends on the particular dataset. For the synthetic dataset as well as for the FASHION dataset, DRPU performs quite stably. However, for MNIST and CIFAR datasets, we see that DRPU significantly overestimates the true value of π', especially for $\pi = 0.1$ and $c = 0.25$. In these cases, the proposed TCPU method returns estimators that are significantly closer to the theoretical values of π'. Tables 3-6, containing estimation errors, reveal that the TCPU method returns the smallest estimation errors or errors that are not significantly different from the winning method for most datasets and parameter settings.

Since the proposed TCPU estimator requires π as input, we also investigated the impact of the value π on the quality of the TCPU estimation of π'. As expected (Figures 5 and 12), the smallest errors are obtained when the TCPU estimator uses a known π (this situation is marked with a blue dashed line). When the estimator $\hat{\pi}$ deviates from the true value of π, we obtain larger prediction errors for π'. Furthermore, we observe larger errors for $\pi = 0.5$ than for small values of $\pi = 0.1$ or $\pi = 0.2$.

We also analyzed the TCPU method's robustness to violations of the label shift assumption (1), indicating that the feature distribution for a given value of the target variable is the same in both the source and target datasets. To this end, we modified the synthetic data generation method as follows. As in previous experiments, negative observations are generated from a 10-dimensional normal distribution $N(0, I)$. However, we change the way positive observations are generated. In the source dataset, we assume that positive observations are generated from $N(a, I)$, where $a = (1, \ldots, 1)$, whereas in the target dataset, the positive observations are generated from $N(a + g, I)$, where g is the

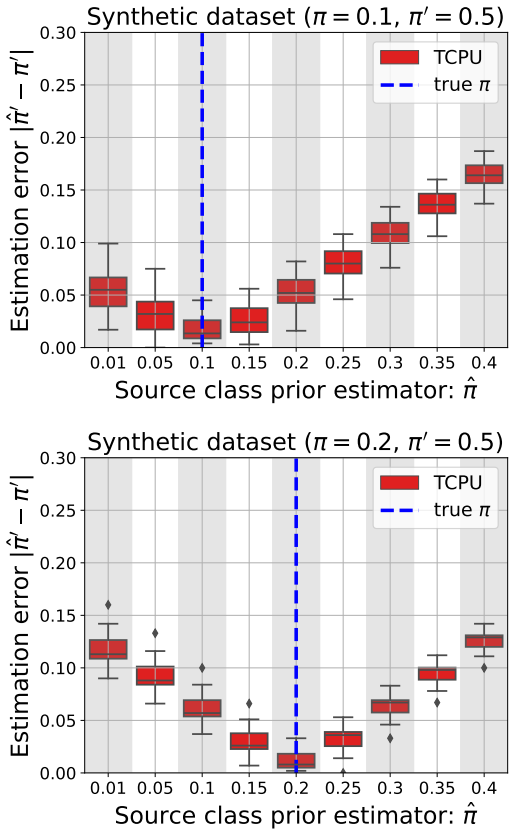

Figure 5: The impact of π estimation on the performance of the TCPU estimator. The boxplot shows estimation errors for TCPU target class prior estimator $|\widehat{\pi}' - \pi'|$, for different source class prior estimators $\hat{\pi}$.

disturbance parameter. The value $g = 0$ means that the assumption of invariance of the feature distribution within classes is met, while values other than zero indicate that it is violated. Figure 6 in the main paper and Figure 13 in the supplement show the distributions of TCPU estimation errors for different values of the parameter g. As expected, the smallest errors usually occur for $g = 0$. It can be seen that for small values of the disturbance parameter g, the method is quite robust and returns small errors. Larger disturbances, in line with intuition, lead to a deterioration in the method's performance. A more pronounced deterioration is observed for negative values of g than for positive ones.

The effect of dataset size on the quality of the estimation is shown in Figure 11 in the supplement. DRPU returns values close to zero for small sample sizes, which results in large estimation errors, while for larger sample sizes it performs comparable or only slightly worse than TCPU. Estimation errors for the KM2-LS method decrease quite slowly with the increase in the

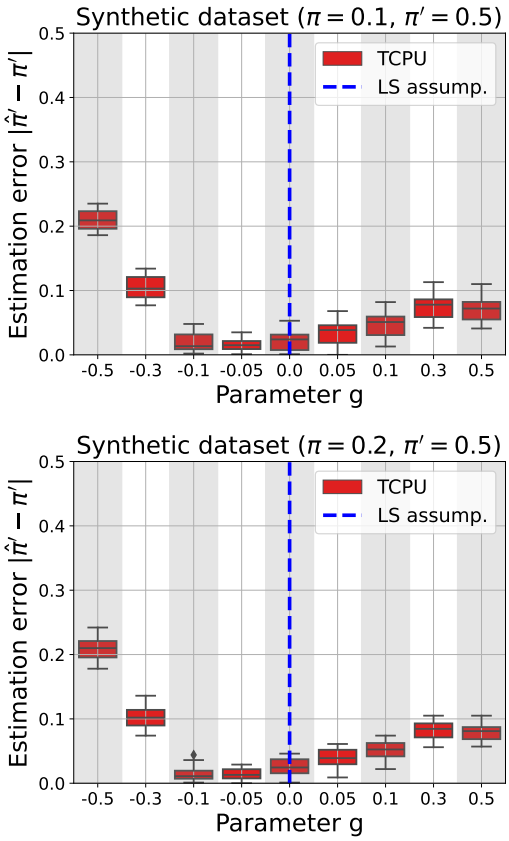

Figure 6: Robustness to violation of the Label Shift (LS) assumption (1). The blue vertical line corresponds to the situation when the assumption is met, and non-zero values of the parameter g indicate a violation of the assumption.

sample size. The advantage of TCPU over DRPU and KM2-LS is clear for small or moderate data sizes.

We analyzed the impact of the Gaussian kernel parameter τ on the estimation quality. The value $\tau = 1/p$ corresponds to the default setting in the function `pairwise_kernels` from the popular `scikit-learn` library in Python. This choice turned out to be quite reasonable in our experiments. For example, for the CIFAR data, with $\pi = 0.1$ and $\pi' = 0.8$, the averaged estimation errors $|\pi' - \widehat{\pi}'|$ were $0.04, 0.02, 0.07$ for $\tau = 0.5/p, 1/p, 2/p$, respectively. We conjecture that there is room for further improvement. For instance, if one could establish the asymptotic variance of TCPU, a natural approach would be to choose τ that minimizes the estimated asymptotic variance. Minimisation can be performed over quantiles of empirical distribution of $\{||X_i - X_j||^{-2}, i \neq j\}$, see Vaz et al. (2019).

Finally, we also analyzed the computation times for the individual methods (Table 2 in the supplement). DRPU is the only method considered that requires model training on the source data, which usually leads to longer computation times. Computing our TCPU estimator requires calculation of the kernel matrix for all pairs of observations, and this is the dominant cost of the entire algorithm. The cost of this step grows quadratically with the number of observations in the data. For example, for the synthetic dataset considered in the paper, the computation times were: 2.81, 5.97, 83.58, 331.56 and 3277.25 seconds for sample sizes of 500, 1K, 2K, 5K, and 10K respectively. However, as the times reported in Table 2 show, our method is still faster than the competing methods (DRPU and KM2-LS). Naturally, speed-up can be achieved using multithreaded computation: computing the kernel matrix is easy to parallelize, and such an option is available in popular functions like pairwise kernels in the scikit-learn library. Moreover, in order to reduce quadratic computational costs of U-statistics which are needed to calculate TCPU estimator, one can use incomplete U-statistics based on random sample of pairs sampled from the set of all pairs (Schrab et al., 2022). In the manuscript, we focused on smaller datasets (up to 2000 observations) because class-prior estimation is most demanding in that setting.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a novel estimator, TCPU, for estimating the target class prior π' under a label shift in the context of positive unlabeled (PU) learning. Our approach, which leverages distribution matching and kernel embedding, provides an explicitly expressed estimate of the target class prior without estimating posterior probabilities via classifier training. We proved that the TCPU estimator is asymptotically consistent and, in Theorem 2, established a calculable non-asymptotic error bound. Experimental results on synthetic and real datasets show that TCPU outperforms existing methods, particularly in cases of small source class prior π values. Importantly, TCPU avoids the problem of significant underestimation or overestimation of the target class prior, which is a drawback of both competing methods DRPU and KM2-LS. A natural direction for future research is to further investigate the effect of π and π' estimation on classification accuracy and to develop testing methods for occurrence of label shift what is of interest for the quantification problems. Also, generalisation of the proposed method to not necessarily binary nominal response by considering the extension of a PU scenario to a noisy data model (Zhang and Sabuncu (2018)) is of interest.

References

Bekker, J. and Davis, J. (2018). Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, pages 1–8.

Bekker, J. and Davis, J. (2020). Learning from positive and unlabeled data: a survey. *Machine Learning*, 109:719–760.

Blanchard, G., Lee, G., and Scott, C. (2010). Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11:2973–3009.

Chen, X., Chen, W., Chen, T., Yuan, Y., Gong, C., Chen, K., and Wang, Z. (2020). Self-PU: Self boosted and calibrated positive-unlabeled training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20.

Dussap, B., Blanchard, G., and Chérif-Abdellatif, B.-E. (2023). Label shift quantification with robustness guarantees via distribution feature matching. In *Proceedings of the European Conferencce on Machine Learning*.

Elkan, C. and Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 213–220.

Forman, G. (2008). Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17:164–206.

Fukumizu, K., Gretton, A., Sun, X., and Schölkopf, B. (2007). Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems*, volume 20.

Fung, G. P. C., Yu, J. X., Lu, H., and Yu, P. S. (2006). Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):6–20.

Garg, S., Wu, Y., Balakrishnan, S., and Lipton, Z. C. (2020). A unified view of label shift estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS' 20, pages 1–11.

Gong, C., Wang, Q., Liu, T., Han, B., You, J., Yang, J., and Tao, D. (2021). Instance-dependent positive and unlabeled learning with labeling bias estimation. *IEEE Trans Pattern Anal Mach Intell*, pages 1–16.

González, P., Castaño, A., Chawla, N., and Coz, J. (2017). A review on quantification learning. *ACM Comput. Surv.*, 50(5).

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773.

Hammoudeh, Z. and Lowd, D. (2020). Learning from positive and unlabeled data with arbitrary positive shift. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Iyer, A., Nath, S., and Sarawagi, S. (2014). Maximum mean discrepancy for class ratio estimation: convergence bounds and kernel selection. In *Proceedings of the 31th International Conferencce on Machine Learning*, IMLR W & CP vol. 32.

Jain, S., White, M., and Radivojac, P. (2016). Estimating the class prior and posterior from noisy positives and unlabeled data. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 2693–2701.

Kiryo, R., Niu, G., du Plessis, M. C., and Sugiyama, M. (2017). Positive-unlabeled learning with non-negative risk estimator. In *Proceedings of the International Conference on Neural Information Processing Systems*, NIPS'17, pages 1674–1684.

Li, F., Dong, S., Leier, A., Han, M., Guo, X., Xu, J., Wang, X., Pan, S., Jia, C., Zhang, Y., Webb, G., Coin, L. J. M., Li, C., and Song, J. (2021). Positive-unlabeled learning in bioinformatics and computational biology: a brief review. *Briefings in Bioinformatics*, 23(1).

Li, X. and Liu, B. (2003). Learning to classify texts using positive and unlabeled data. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, page 587–592.

Lipton, Z. C., Wang, Y., and Smola, A. J. (2018). Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, ICML' 18, pages 3128–3136.

Luo, C., Zhao, P., Chen, C., Qiao, B., Du, C., Zhang, H., Wu, W., Cai, S., He, B., Rajmohan, S., and Lin, Q. (2021). Pulns: Positive-unlabeled learning with effective negative sample selector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35 of *AAAI'21*, pages 8784–8792.

Mc Diarmid, C. (1989). On the method of bounded differences. *Survey in Combinatorics*, pages 148–188.

Mielniczuk, J. and Wawrzenczyk, A. (2024). Single-sample versus case-control sampling scheme for Positive Unlabeled data: the story of two scenarios. *Fundamenta Informaticae*, 191:1–17.

Na, B., Kim, H., Song, K., Joo, W., Kim, Y., and Moon, I. (2020). Deep generative positive-unlabeled

learning under selection bias. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1155–1164.

Nakajima, S. and Siguyama, M. (2023). Positive-unlabeled classification under class-prior shift: a prior-invariant approach based on density ratio estimation. *Machine Learning*, 112:889–919.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, NIPS'19, pages 8024–8035.

Ramaswamy, H., Scott, C., and Tewari, A. (2016). Mixture proportion estimation via kernel embeddings of distributions. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 2052–2060.

Roland, T., Bock, C., Tschoellitsch, T., Maletzky, A., Hochreiter, S., Meier, J., and Klambauer, G. (2022). Domain shifts in machine learning based covid-19 diagnosis from blood tests. *Journal of Medical Systems*, 46(5):1–12.

Saerens, M., Latinne, P., and Decaestecker, C. (2002). Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural Comput.*, 14(1):21–41.

Sakai, T. and Shimizu, N. (2019). Covariate shift adaptation on learning from positive and unlabeled data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4838–4845.

Schrab, A., Kim, I., Guedj, B., and Gretton, A. (2022). Efficient aggregated kernel tests using incomplete u-statistics. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS'22.

Sechidis, K., Sperrin, M., Petherick, E. S., Luján, M., and Brown, G. (2017). Dealing with under-reported variables: An information theoretic solution. *International Journal of Approximate Reasoning*, 85:159 – 177.

Tolstikhin, I., Sriperumbudur, B. K., and Muandet, K. (2017). Minimax estimation of kernel mean embeddings. *Journal of Machine Learning Research*, 18(86):1–47.

Vaz, A., Izbicki, R., and Stern, R. (2019). Quantification under prior probability shift: the ratio estimator and its extensions. *Journal of Machine Learning Research*, 20:1–33.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. (2014). Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conferencce on Machine Learning*.

Zhang, Z. and Sabuncu, M. (2018). Generalizec cross entropy loss for training neural networks with noisy labels. In *NIPS'18*, pages 8792 – 8802.

Zhao, Y., Xu, Q., Jiang, Y., Wen, P., and Huang, Q. (2022). Dist-pu: Positive-unlabeled learning from a label distribution perspective. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, CVPR'22, pages 14461–14470.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Jan Mielniczuk [1,3], Wojciech Rejchel [2], Paweł Teisseyre[1,3]

Prior shift estimation for positive unlabeled data through the lens of kernel embedding: Supplementary Materials

A Proofs

A.1 Proof of Theorem 1

The proof of (i) follows from Chebyshev's inequality applied for

$$\Phi(\hat{P}) = n^{-1}\sum_{i=1}^{n}\Phi(X_i)$$

as well as $\Phi(\hat{P}_+)$ and $\Phi(\hat{P}')$: fix $\varepsilon > 0$, then

$$P(||\Phi(\hat{P}) - \Phi(P)||_{\mathcal{H}} > \varepsilon) \leq \frac{\mathbb{E}||\Phi(\hat{P}) - \Phi(P)||^2_{\mathcal{H}}}{\varepsilon^2}. \tag{11}$$

Now we focus on bounding the numerator in (11). First, we obtain

$$\mathbb{E}||\Phi(\hat{P}) - \Phi(P)||^2 = \mathbb{E}||\Phi(\hat{P})||^2_{\mathcal{H}} - 2\mathbb{E} < \Phi(\hat{P}), \Phi(P) >_{\mathcal{H}} + ||\Phi(P)||^2_{\mathcal{H}}. \tag{12}$$

Next, we consider the two first terms on the right-hand side of (12). For the first one, we have:

$$\begin{aligned}
\mathbb{E}\left|\left|\frac{1}{n}\sum_{i=1}^{n}\phi(X_i)\right|\right|^2_{\mathcal{H}} &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} < \phi(X_i), \phi(X_j) >_{\mathcal{H}} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} K(X_i, X_j) \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}K(X_i, X_i) + \frac{1}{n^2}\sum_{1\leq i\neq j\leq n}\mathbb{E}K(X_i, X_j) \\
&= \frac{1}{n}\mathbb{E}K(X_1, X_1) + \left(1 - \frac{1}{n}\right)||\Phi(P)||^2_{\mathcal{H}}.
\end{aligned}$$

Moreover, we have

$$\mathbb{E} < \Phi(\hat{P}), \Phi(P) >_{\mathcal{H}} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E} < \phi(X_i), \Phi(P) >_{\mathcal{H}} = \mathbb{E} < \phi(X_1), \Phi(P) >_{\mathcal{H}} . \tag{13}$$

From the reproducing property we obtain $< \phi(x_1), \Phi(P) >_{\mathcal{H}} = [\Phi(P)](x_1) = \mathbb{E}K(X, x_1)$. Therefore, the right-hand side of (13) equals $||\Phi(P)||^2_{\mathcal{H}}$. Finally, (12) is $\frac{1}{n}[\mathbb{E}K(X, X) - ||\Phi(P)||^2_{\mathcal{H}}]$, so the right-hand side of (11) tends to zero as n goes to infinity. This fact implies that $\Phi(\hat{P})$ tends to $\Phi(P)$ in probability. Obviously, the analogous properties hold for $\Phi(\hat{P}_+)$ and $\Phi(\hat{P}')$. In particular, we have that $||\Phi(\hat{P}) - \Phi(\hat{P}_+)||_{\mathcal{H}}$ is positive with probability tending to one as $N \to \infty$. It follows from continuity of a norm and the assumptions that

$$||\Phi(\hat{P}) - \Phi(\hat{P}_+)||_{\mathcal{H}} \to_P ||\Phi(P) - \Phi(P_+)||_{\mathcal{H}} = (1 - \pi)||\Phi(P_-) - \Phi(P_+)||_{\mathcal{H}}.$$

Thus, the second equality in Lemma 2 (main paper) and continuity of a scalar product give

$$\hat{\pi}' \to_P 1 - \frac{(1 - \pi) < \Phi(P) - \Phi(P_+), \Phi(P') - \Phi(P_+) >_{\mathcal{H}}}{||\Phi(P) - \Phi(P_+)||^2_{\mathcal{H}}} = \pi'.$$

Next, we focus on the proof of (ii). From the first equality in (6) and the Cauchy–Schwarz inequality we have

$$|\widehat{\pi}' - \pi'| = \frac{< \Phi(\hat{P}) - \Phi(\hat{P}_+), \Delta - \pi'[\Phi(\hat{P}) - \Phi(\hat{P}_+)] >_{\mathcal{H}}}{||\Phi(\hat{P}) - \Phi(\hat{P}_+)||^2_{\mathcal{H}}} \leq \frac{||\Delta - \pi'[\Phi(\hat{P}) - \Phi(\hat{P}_+)]||_{\mathcal{H}}}{||\Phi(\hat{P}) - \Phi(\hat{P}_+)||_{\mathcal{H}}}. \tag{14}$$

Notice that

$$\Delta - \pi'[\Phi(\hat{P}) - \Phi(\hat{P}_+)] = (1-\pi')\Phi(\hat{P}) - (1-\pi)\Phi(\hat{P}') + (\pi' - \pi)\Phi(\hat{P}_+)$$

and

$$(1-\pi')\Phi(P) - (1-\pi)\Phi(P') + (\pi' - \pi)\Phi(P_+) = 0,$$

which imply that

$$||\Delta - \pi'[\Phi(\hat{P}) - \Phi(\hat{P}_+)]||_{\mathcal{H}} \leq (1-\pi')||\Phi(\hat{P}) - \Phi(P)||_{\mathcal{H}} + (1-\pi)||\Phi(\hat{P}') - \Phi(P')||_{\mathcal{H}} + |\pi' - \pi| \times ||\Phi(\hat{P}_+) - \Phi(P_+)||_{\mathcal{H}}.$$

Now we use Lemma 3 (given below) and consider the event on which all three inequalities in this lemma hold. In this case

$$||\Delta - \pi'[\Phi(\hat{P}) - \Phi(\hat{P}_+)]||_{\mathcal{H}} \leq 4\sqrt{\frac{M}{N}\log(1/\delta)} \tag{15}$$

and

$$||\Phi(\hat{P}) - \Phi(\hat{P}_+)||_{\mathcal{H}} \geq ||\Phi(P) - \Phi(P_+)||_{\mathcal{H}} - ||\Phi(\hat{P}) - \Phi(P)||_{\mathcal{H}} - ||\Phi(\hat{P}_+) - \Phi(P_+)||_{\mathcal{H}},$$

which again can be bounded by

$$(1-\pi)||\Phi(P_-) - \Phi(P_+)||_{\mathcal{H}} - 4\sqrt{\frac{M}{N}\log(1/\delta)}.$$

From the assumption (8) the above expression is not smaller than $\alpha(1-\pi)||\Phi(P_-) - \Phi(P_+)||_{\mathcal{H}}$. In particular, we have that $||\Phi(\hat{P}) - \Phi(\hat{P}_+)||_{\mathcal{H}}$ is positive with high probability. Since the numerator and the denominator on the right-hand side of (14) are bounded, the proof of (ii) is finished.

A.2 Proof of Theorem 2

The proof is even simpler than the proof of Theorem 1 and involves bounding the numerator on the right-hand side of (14) done in (15).

A.3 Lemma 3 and its proof

Lemma 3. *Suppose that $M = \sup_x K(x,x) < \infty$ and $\delta \leq \exp(-(\sqrt{2}+1)^2/2)$ is arbitrary. Then with probability at least $1 - 3\delta$ the following three inequalities simultaneously hold*

$$||\Phi(\hat{P}) - \Phi(P)||_{\mathcal{H}} \leq 2\sqrt{\frac{M}{n}\log(1/\delta)}\,, \qquad ||\Phi(\hat{P}') - \Phi(P')||_{\mathcal{H}} \leq 2\sqrt{\frac{M}{n'}\log(1/\delta)}\,,$$

$$||\Phi(\hat{P}_+) - \Phi(P_+)||_{\mathcal{H}} \leq 2\sqrt{\frac{M}{m}\log(1/\delta)}\,.$$

Proof. Fix $\delta \leq \exp(-(\sqrt{2}+1)^2/2)$. We focus on the first inequality in the lemma. The initial step is McDiarmid's inequality (Mc Diarmid, 1989) applied for the function

$$F(X_1, \ldots, X_n) = ||\Phi(\hat{P}) - \Phi(P)||_{\mathcal{H}} = \left|\left|\frac{1}{n}\sum_{i=1}^{n}\phi(X_i) - \Phi(P)\right|\right|_{\mathcal{H}}.$$

We are to bound a difference

$$|F(X_1,\ldots,X_n) - F(X_1,\ldots,X_i',\ldots,X_n)| \leq \frac{1}{n}||\phi(X_i) - \phi(X_i')||_{\mathcal{H}} \leq \frac{2\sqrt{M}}{n} \ .$$

Therefore, we obtain that with probability at least $1-\delta$

$$||\Phi(\hat{P}) - \Phi(P)||_{\mathcal{H}} \leq \mathbb{E}||\Phi(\hat{P}) - \Phi(P)||_{\mathcal{H}} + \sqrt{\frac{2M\log(1/\delta)}{n}}.$$

Applying Jensen's inequality to the expectation above, we get

$$\mathbb{E}||\Phi(\hat{P}) - \Phi(P)||_{\mathcal{H}} \leq \sqrt{\mathbb{E}||\Phi(\hat{P}) - \Phi(P)||^2_{\mathcal{H}}}.$$

The expression under the square root has been already considered in the Proof of Theorem 1 and was bounded there by $\frac{1}{n}[\mathbb{E}K(X,X) - ||\Phi(P)||^2_{\mathcal{H}}]$, which is smaller than $\frac{M}{n}$. Note that for the chosen δ we have $1+\sqrt{2\log(1/\delta)} \leq 2\sqrt{\log(1/\delta)}$, which finishes the proof. $\square$

The similar concentration inequalities are proven, for instance, in (Tolstikhin et al., 2017, Proposition A.1 and Remark A.2).

B Computational times

All experiments are executed in an isolated manner on the same machine with access to 32Gb RAM and 8 cores of an Intel(R) Xeon(R) CPU E31270 3.40GHz.

Table 2: **Computational times (in seconds)**, for example parameter setting $\pi = 0.2$ and $\pi' = 0.6$.

Dataset	TCPU	DRPU	KM2-LS
Synthetic	83.582 ± 0.473	**48.72 ± 0.213**	162.818 ± 1.855
MNIST	**155.415 ± 2.28**	440.129 ± 18.567	280.33 ± 5.337
CIFAR	**78.873 ± 0.997**	151.749 ± 5.435	160.437 ± 1.658
Fashion	**38.371 ± 0.319**	48.768 ± 0.291	80.313 ± 0.384
Diabetes	**0.043 ± 0.009**	0.293 ± 0.014	0.173 ± 0.019
Spambase	**0.015 ± 0.004**	0.023 ± 0.005	0.102 ± 0.018
Segment	0.052 ± 0.012	**0.039 ± 0.015**	0.194 ± 0.033
Waveform	0.07 ± 0.004	0.087 ± 0.009	**0.036 ± 0.006**
Vehicle	**0.073 ± 0.015**	0.091 ± 0.015	0.318 ± 0.016
Yeast	**0.117 ± 0.016**	0.22 ± 0.02	0.249 ± 0.018
Banknote	0.041 ± 0.011	**0.026 ± 0.006**	0.16 ± 0.033

C Technical details about the DRPU method

The DRPU method requires learning a parametric model. As in the original work, we used 5-layer Multi-Layer Perceptron (MLP) : $p - 300 - 300 - 300 - 1$ (with p being the size of the feature vector) with ReLU activation, and trained by Adam with the default momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and ℓ_2 regularization parameter 5×10^3. Training was performed for 1000 epochs with the batch size 100 and learning rate 2×10^{-5}. The DRPU method requires setting the hyperparameter α for non-negative correction. In the experiments, for image datasets we used the values proposed in Nakajima and Siguyama (2023): $\alpha = 0.475, 0.425, 0.6$, for MNIST, CIFAR10 and FashionMNIST, respectively. For the remaining datasets, we set the value $\alpha = \hat{\pi}$, where $\hat{\pi}$ is the KM estimator, using the fact that α should satisfy $0 \leq \alpha \leq \pi$. This is usually a sensible choice, for image datasets there is usually no significant difference between $\alpha = \hat{\pi}$ and the default values given above.

D Controlling the size of the source data and the labeling frequency.

We follow the procedure described e.g. in Mielniczuk and Wawrzenczyk (2024) to control the size of the source data and the labeling frequency c, which represents the percentage of labeled observations among all positive observations. Let n denote the total number of observations in the source dataset. We pick $c \times \pi \times n$ observations from positive class and $(1-c) \times n$ from the whole data set. Thus $c \times \pi \times n + \pi \times (1-c) \times n = \pi \times n$ is an expected number of observations from the positive class in the sample and fraction c of them will be labeled on average. In order to ensure that the chosen sample has size equal to n, both sizes should be increased $A = (1 - c(1-\pi))^{-1}$ times i.e. size of chosen labeled sample should be $A \times c \times \pi \times n$ and $A \times (1-c) \times n$ for the unlabeled one. Note that both samples are not necessarily disjoint.

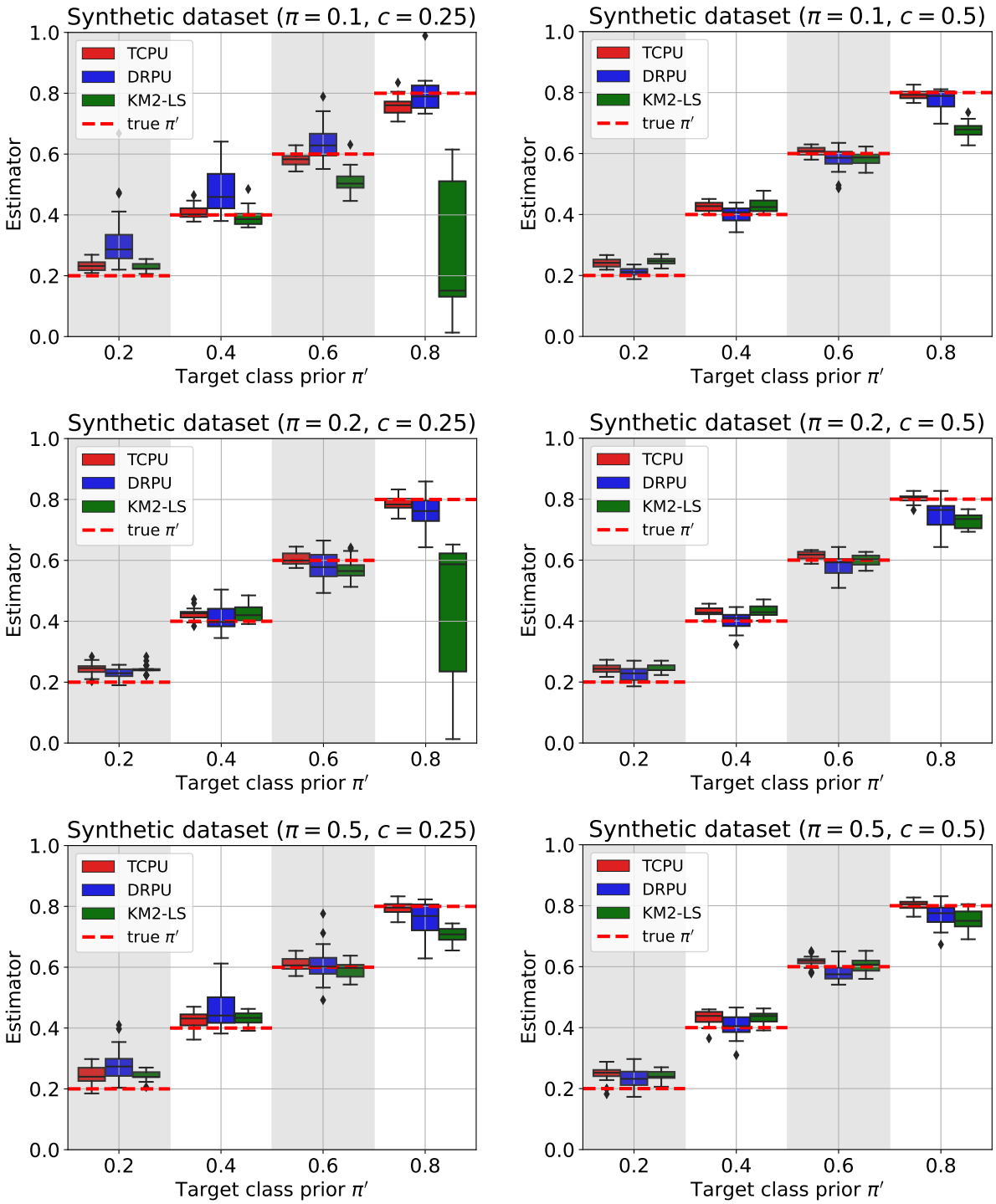

Figure 7: Distribution of estimators (red line indicates the true π'). Size of the source data and the target data is 2000.

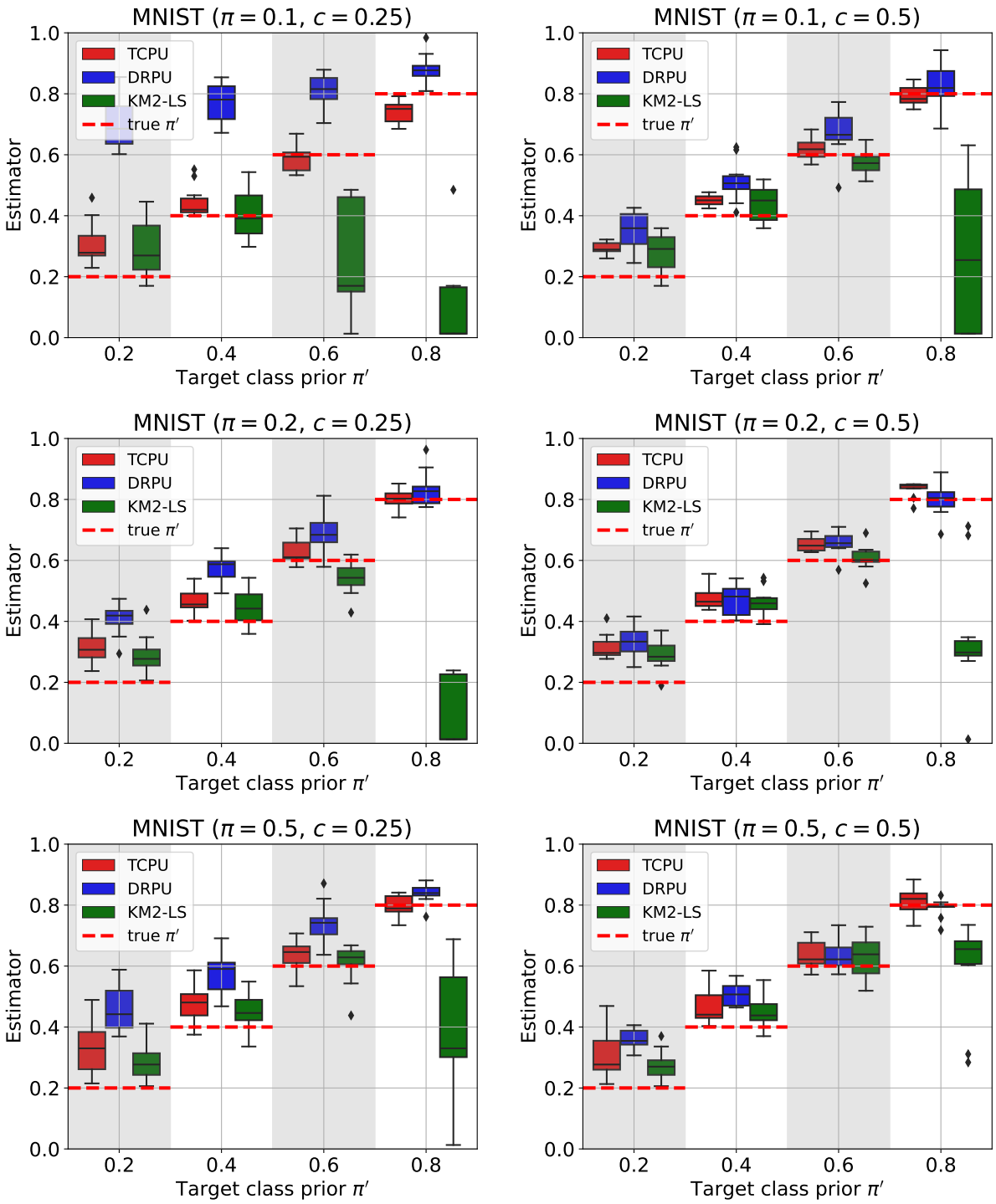

Figure 8: Distribution of estimators (red line indicates the true π') for MNIST dataset.

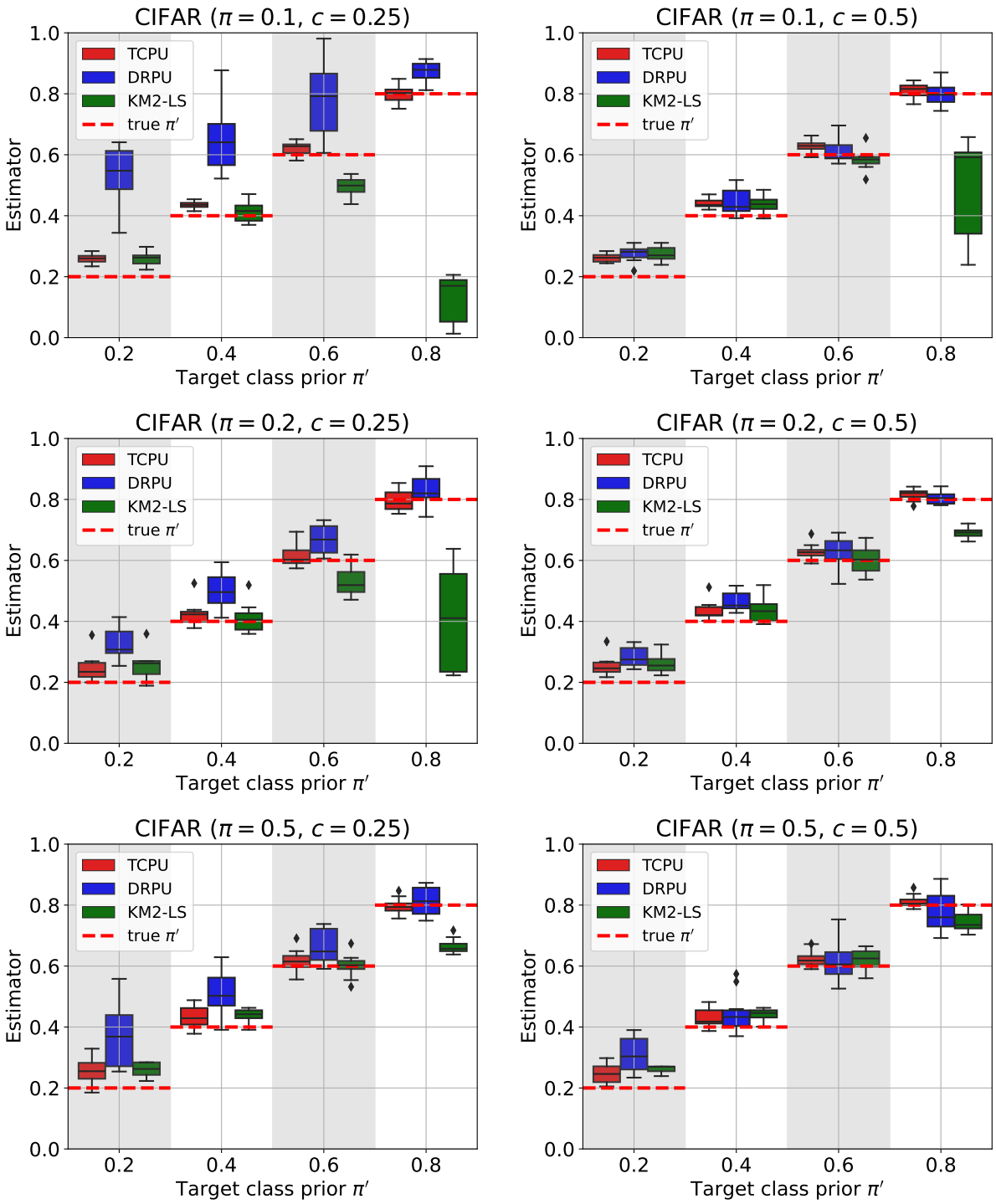

Figure 9: Distribution of estimators (red line indicates the true π') for CIFAR dataset.

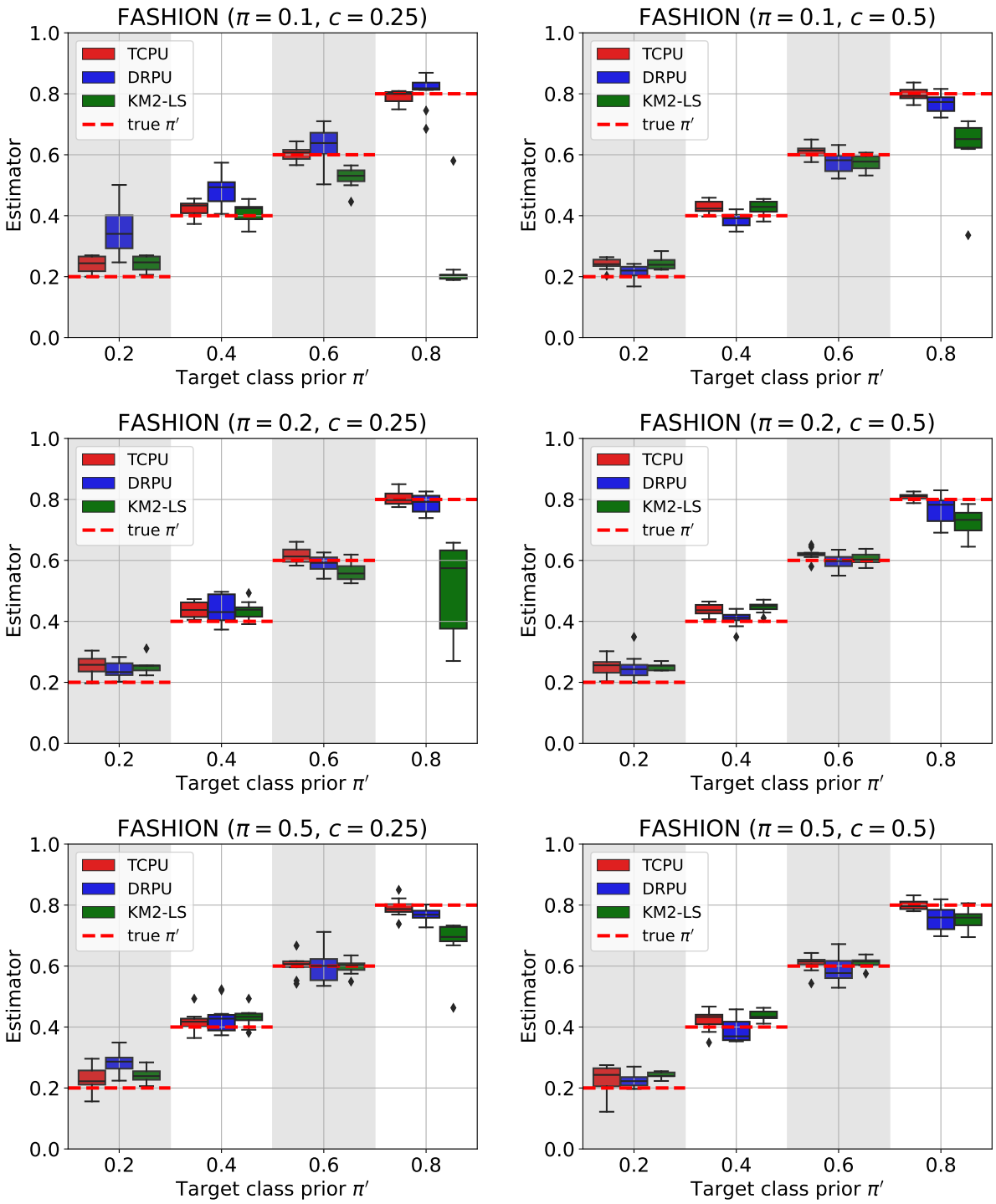

Figure 10: Distribution of estimators (red line indicates the true π') for FASHION dataset.

Table 3: **Estimation errors** for image datasets for $c = 0.25$. The winning method and the method whose error does not differ from the winning method by more than 0.01 are bolded.

Dataset	π	π'	TCPU	DRPU	KM2-LS
MNIST	0.1	0.2	0.109 ± 0.021	0.507 ± 0.027	**0.095 ± 0.026**
		0.4	**0.044 ± 0.016**	0.371 ± 0.019	0.065 ± 0.014
		0.6	**0.037 ± 0.008**	0.209 ± 0.017	0.332 ± 0.054
		0.8	**0.058 ± 0.011**	0.083 ± 0.014	0.695 ± 0.045
	0.2	0.2	0.116 ± 0.015	0.207 ± 0.016	**0.089 ± 0.02**
		0.4	0.067 ± 0.013	0.172 ± 0.013	**0.056 ± 0.015**
		0.6	**0.034 ± 0.011**	0.094 ± 0.016	0.062 ± 0.015
		0.8	**0.024 ± 0.006**	0.047 ± 0.015	0.702 ± 0.033
	0.5	0.2	0.13 ± 0.027	0.263 ± 0.023	**0.083 ± 0.019**
		0.4	0.082 ± 0.016	0.172 ± 0.021	**0.064 ± 0.012**
		0.6	**0.05 ± 0.009**	0.143 ± 0.02	**0.053 ± 0.013**
		0.8	**0.032 ± 0.005**	0.046 ± 0.006	0.396 ± 0.062
CIFAR	0.1	0.2	**0.06 ± 0.005**	0.334 ± 0.029	**0.059 ± 0.007**
		0.4	**0.035 ± 0.003**	0.262 ± 0.038	**0.029 ± 0.006**
		0.6	**0.027 ± 0.004**	0.183 ± 0.037	0.104 ± 0.01
		0.8	**0.022 ± 0.006**	0.073 ± 0.01	0.666 ± 0.025
	0.2	0.2	**0.046 ± 0.013**	0.126 ± 0.016	0.057 ± 0.013
		0.4	**0.033 ± 0.01**	0.101 ± 0.019	**0.036 ± 0.009**
		0.6	**0.026 ± 0.008**	0.07 ± 0.014	0.074 ± 0.011
		0.8	**0.03 ± 0.005**	0.049 ± 0.011	0.39 ± 0.053
	0.5	0.2	**0.06 ± 0.012**	0.169 ± 0.031	**0.062 ± 0.007**
		0.4	**0.04 ± 0.009**	0.115 ± 0.021	**0.04 ± 0.006**
		0.6	**0.03 ± 0.008**	0.067 ± 0.016	**0.027 ± 0.008**
		0.8	**0.02 ± 0.005**	0.041 ± 0.007	0.135 ± 0.008
FASHION	0.1	0.2	**0.04 ± 0.008**	0.149 ± 0.024	**0.042 ± 0.008**
		0.4	**0.03 ± 0.006**	0.085 ± 0.015	**0.032 ± 0.004**
		0.6	**0.02 ± 0.004**	0.055 ± 0.011	0.074 ± 0.011
		0.8	**0.018 ± 0.006**	0.043 ± 0.01	0.56 ± 0.036
	0.2	0.2	0.054 ± 0.009	**0.042 ± 0.008**	0.053 ± 0.007
		0.4	**0.039 ± 0.008**	0.047 ± 0.012	**0.037 ± 0.008**
		0.6	**0.025 ± 0.007**	**0.024 ± 0.006**	0.042 ± 0.007
		0.8	**0.019 ± 0.004**	**0.025 ± 0.006**	0.289 ± 0.048
	0.5	0.2	**0.042 ± 0.008**	0.084 ± 0.011	**0.042 ± 0.007**
		0.4	**0.03 ± 0.007**	0.045 ± 0.012	**0.037 ± 0.007**
		0.6	**0.024 ± 0.007**	0.044 ± 0.01	**0.016 ± 0.005**
		0.8	**0.024 ± 0.006**	**0.032 ± 0.006**	0.119 ± 0.024

Table 4: **Estimation errors** for image datasets for $c = 0.5$. The winning method and the method whose error does not differ from the winning method by more than 0.01 are bolded.

Dataset	π	π'	TCPU	DRPU	KM2-LS
MNIST	0.1	0.2	**0.094 ± 0.006**	0.148 ± 0.019	**0.085 ± 0.016**
		0.4	**0.05 ± 0.005**	0.114 ± 0.02	**0.059 ± 0.012**
		0.6	**0.03 ± 0.007**	0.094 ± 0.014	**0.039 ± 0.008**
		0.8	**0.029 ± 0.004**	0.06 ± 0.016	0.538 ± 0.076
	0.2	0.2	0.116 ± 0.013	0.133 ± 0.015	**0.092 ± 0.014**
		0.4	0.078 ± 0.011	**0.069 ± 0.015**	**0.065 ± 0.013**
		0.6	0.053 ± 0.007	0.064 ± 0.008	**0.029 ± 0.009**
		0.8	**0.039 ± 0.004**	**0.042 ± 0.011**	0.45 ± 0.061
	0.5	0.2	0.107 ± 0.023	0.16 ± 0.01	**0.075 ± 0.015**
		0.4	**0.068 ± 0.017**	0.108 ± 0.012	**0.059 ± 0.015**
		0.6	**0.047 ± 0.013**	**0.045 ± 0.015**	0.065 ± 0.012
		0.8	0.037 ± 0.008	**0.019 ± 0.008**	0.209 ± 0.048
CIFAR	0.1	0.2	**0.061 ± 0.004**	0.076 ± 0.008	0.073 ± 0.007
		0.4	**0.04 ± 0.005**	0.046 ± 0.013	**0.042 ± 0.008**
		0.6	**0.03 ± 0.005**	0.03 ± 0.009	**0.029 ± 0.007**
		0.8	**0.021 ± 0.004**	0.033 ± 0.007	0.302 ± 0.051
	0.2	0.2	**0.053 ± 0.01**	0.082 ± 0.01	**0.058 ± 0.009**
		0.4	**0.035 ± 0.01**	0.065 ± 0.01	**0.039 ± 0.012**
		0.6	**0.031 ± 0.007**	0.047 ± 0.009	**0.036 ± 0.007**
		0.8	**0.022 ± 0.003**	**0.018 ± 0.004**	0.11 ± 0.005
	0.5	0.2	**0.049 ± 0.01**	0.109 ± 0.017	0.061 ± 0.004
		0.4	**0.034 ± 0.008**	0.057 ± 0.017	**0.042 ± 0.006**
		0.6	**0.026 ± 0.008**	0.054 ± 0.015	**0.033 ± 0.006**
		0.8	**0.016 ± 0.006**	0.061 ± 0.009	0.056 ± 0.01
FASHION	0.1	0.2	0.041 ± 0.005	**0.022 ± 0.005**	0.045 ± 0.006
		0.4	**0.029 ± 0.006**	**0.021 ± 0.006**	0.031 ± 0.005
		0.6	**0.019 ± 0.004**	0.033 ± 0.008	**0.028 ± 0.007**
		0.8	**0.018 ± 0.004**	0.035 ± 0.008	0.171 ± 0.033
	0.2	0.2	0.051 ± 0.009	**0.049 ± 0.013**	**0.05 ± 0.003**
		0.4	0.039 ± 0.006	**0.02 ± 0.005**	0.047 ± 0.005
		0.6	0.025 ± 0.004	**0.021 ± 0.005**	**0.015 ± 0.003**
		0.8	**0.012 ± 0.002**	0.04 ± 0.011	0.073 ± 0.013
	0.5	0.2	0.049 ± 0.008	**0.025 ± 0.006**	0.041 ± 0.004
		0.4	**0.036 ± 0.005**	**0.035 ± 0.004**	0.038 ± 0.005
		0.6	**0.023 ± 0.005**	0.037 ± 0.007	**0.016 ± 0.003**
		0.8	**0.014 ± 0.003**	0.05 ± 0.01	0.048 ± 0.009

Table 5: **Estimation errors** for benchmark datasets for $c = 0.5$ (part 1). The winning method and the method whose error does not differ from the winning method by more than 0.01 are bolded.

Dataset	π	π'	TCPU	DRPU	KM2-LS
Diabetes	0.1	0.2	**0.072 ± 0.013**	0.2 ± 0.0	**0.071 ± 0.023**
		0.4	**0.055 ± 0.019**	0.4 ± 0.0	0.149 ± 0.018
		0.6	**0.104 ± 0.014**	0.6 ± 0.0	0.404 ± 0.021
		0.8	**0.157 ± 0.014**	0.8 ± 0.0	0.771 ± 0.015
	0.2	0.2	0.129 ± 0.011	0.239 ± 0.037	**0.103 ± 0.011**
		0.4	**0.072 ± 0.013**	0.417 ± 0.016	0.088 ± 0.015
		0.6	**0.044 ± 0.011**	0.563 ± 0.035	0.281 ± 0.017
		0.8	**0.065 ± 0.016**	0.735 ± 0.061	0.627 ± 0.052
	0.5	0.2	**0.121 ± 0.027**	0.576 ± 0.018	0.132 ± 0.011
		0.4	**0.058 ± 0.011**	0.428 ± 0.014	**0.055 ± 0.014**
		0.6	**0.043 ± 0.009**	0.293 ± 0.014	0.173 ± 0.019
		0.8	**0.045 ± 0.011**	0.138 ± 0.012	0.645 ± 0.056
Spambase	0.1	0.2	**0.021 ± 0.006**	0.067 ± 0.013	**0.024 ± 0.004**
		0.4	**0.02 ± 0.005**	0.061 ± 0.011	0.078 ± 0.015
		0.6	**0.021 ± 0.007**	0.044 ± 0.008	0.299 ± 0.022
		0.8	**0.031 ± 0.008**	**0.036 ± 0.01**	0.592 ± 0.012
	0.2	0.2	**0.023 ± 0.005**	0.046 ± 0.007	**0.02 ± 0.003**
		0.4	**0.021 ± 0.004**	0.05 ± 0.01	0.039 ± 0.006
		0.6	**0.014 ± 0.003**	0.037 ± 0.01	0.161 ± 0.024
		0.8	**0.016 ± 0.003**	0.029 ± 0.003	0.469 ± 0.02
	0.5	0.2	**0.025 ± 0.01**	0.057 ± 0.015	**0.022 ± 0.005**
		0.4	**0.025 ± 0.005**	0.059 ± 0.012	**0.02 ± 0.005**
		0.6	**0.015 ± 0.004**	**0.023 ± 0.005**	0.102 ± 0.018
		0.8	**0.017 ± 0.004**	0.046 ± 0.012	0.378 ± 0.026
Segment	0.1	0.2	0.019 ± 0.004	**0.007 ± 0.002**	0.032 ± 0.008
		0.4	**0.022 ± 0.004**	**0.019 ± 0.004**	0.12 ± 0.013
		0.6	**0.021 ± 0.006**	**0.029 ± 0.007**	0.18 ± 0.028
		0.8	**0.022 ± 0.004**	0.042 ± 0.01	0.499 ± 0.025
	0.2	0.2	**0.017 ± 0.005**	**0.008 ± 0.003**	0.036 ± 0.006
		0.4	**0.022 ± 0.004**	**0.022 ± 0.006**	0.114 ± 0.013
		0.6	**0.02 ± 0.005**	0.037 ± 0.013	0.182 ± 0.032
		0.8	**0.019 ± 0.003**	0.047 ± 0.013	0.504 ± 0.017
	0.5	0.2	0.101 ± 0.021	**0.028 ± 0.008**	0.052 ± 0.008
		0.4	0.086 ± 0.021	**0.043 ± 0.012**	0.121 ± 0.01
		0.6	0.052 ± 0.012	**0.039 ± 0.015**	0.194 ± 0.033
		0.8	**0.026 ± 0.006**	0.041 ± 0.013	0.493 ± 0.024
Waveform	0.1	0.2	0.156 ± 0.006	**0.105 ± 0.01**	0.165 ± 0.006
		0.4	0.107 ± 0.006	0.101 ± 0.018	**0.091 ± 0.008**
		0.6	0.047 ± 0.005	0.08 ± 0.012	**0.026 ± 0.007**
		0.8	**0.016 ± 0.004**	0.039 ± 0.007	0.445 ± 0.027
	0.2	0.2	**0.165 ± 0.005**	0.215 ± 0.015	0.176 ± 0.006
		0.4	0.117 ± 0.007	0.156 ± 0.014	**0.103 ± 0.007**
		0.6	0.063 ± 0.004	0.104 ± 0.016	**0.026 ± 0.006**
		0.8	**0.017 ± 0.003**	0.07 ± 0.007	0.525 ± 0.092
	0.5	0.2	**0.169 ± 0.005**	0.187 ± 0.02	**0.169 ± 0.008**
		0.4	0.123 ± 0.006	0.156 ± 0.02	**0.107 ± 0.007**
		0.6	0.07 ± 0.004	0.087 ± 0.009	**0.036 ± 0.006**
		0.8	**0.017 ± 0.004**	0.054 ± 0.01	0.19 ± 0.008

Table 6: **Estimation errors** for benchmark datasets for $c = 0.5$ (part 2). The winning method and the method whose error does not differ from the winning method by more than 0.01 are bolded.

Dataset	π	π'	TCPU	DRPU	KM2-LS
Yeast	0.1	0.2	0.087 ± 0.024	0.2 ± 0.0	**0.057 ± 0.019**
		0.4	**0.043 ± 0.016**	0.4 ± 0.0	0.164 ± 0.015
		0.6	**0.072 ± 0.017**	0.6 ± 0.0	0.423 ± 0.027
		0.8	**0.102 ± 0.016**	0.8 ± 0.0	0.723 ± 0.032
	0.1	0.2	0.194 ± 0.045	0.462 ± 0.023	**0.141 ± 0.029**
		0.4	0.142 ± 0.028	0.37 ± 0.015	**0.1 ± 0.015**
		0.6	**0.095 ± 0.019**	0.216 ± 0.017	0.272 ± 0.022
		0.8	**0.057 ± 0.015**	0.102 ± 0.011	0.641 ± 0.047
	0.1	0.2	0.238 ± 0.035	0.43 ± 0.025	**0.164 ± 0.042**
		0.4	0.161 ± 0.022	0.356 ± 0.019	**0.146 ± 0.026**
		0.6	**0.117 ± 0.016**	0.22 ± 0.02	0.249 ± 0.018
		0.8	**0.055 ± 0.015**	0.065 ± 0.015	0.543 ± 0.04
Vehicle	0.1	0.2	**0.035 ± 0.006**	0.2 ± 0.0	0.07 ± 0.009
		0.4	**0.055 ± 0.012**	0.4 ± 0.0	0.227 ± 0.014
		0.6	**0.065 ± 0.017**	0.6 ± 0.0	0.496 ± 0.031
		0.8	**0.088 ± 0.021**	0.8 ± 0.0	0.729 ± 0.028
	0.2	0.2	**0.028 ± 0.006**	0.099 ± 0.026	0.047 ± 0.006
		0.4	**0.042 ± 0.006**	0.088 ± 0.024	0.163 ± 0.009
		0.6	**0.047 ± 0.007**	**0.038 ± 0.011**	0.344 ± 0.029
		0.8	0.051 ± 0.011	**0.032 ± 0.007**	0.655 ± 0.044
	0.5	0.2	0.088 ± 0.016	0.18 ± 0.029	**0.016 ± 0.006**
		0.4	**0.078 ± 0.02**	0.172 ± 0.027	0.174 ± 0.015
		0.6	**0.073 ± 0.015**	0.091 ± 0.015	0.318 ± 0.016
		0.8	0.078 ± 0.013	**0.066 ± 0.014**	0.677 ± 0.045
Banknote	0.1	0.2	**0.02 ± 0.003**	0.2 ± 0.0	0.063 ± 0.014
		0.4	**0.02 ± 0.003**	0.4 ± 0.0	0.204 ± 0.025
		0.6	**0.045 ± 0.008**	0.6 ± 0.0	0.372 ± 0.028
		0.8	**0.052 ± 0.006**	0.8 ± 0.0	0.584 ± 0.041
	0.2	0.2	**0.021 ± 0.005**	**0.022 ± 0.004**	0.04 ± 0.009
		0.4	**0.019 ± 0.004**	**0.027 ± 0.012**	0.11 ± 0.025
		0.6	**0.03 ± 0.009**	**0.024 ± 0.008**	0.309 ± 0.024
		0.8	**0.023 ± 0.005**	**0.021 ± 0.007**	0.487 ± 0.033
	0.5	0.2	0.07 ± 0.019	**0.016 ± 0.006**	0.028 ± 0.006
		0.4	**0.05 ± 0.016**	**0.041 ± 0.018**	0.117 ± 0.026
		0.6	0.041 ± 0.011	**0.026 ± 0.006**	0.16 ± 0.033
		0.8	**0.033 ± 0.009**	**0.031 ± 0.01**	0.38 ± 0.048

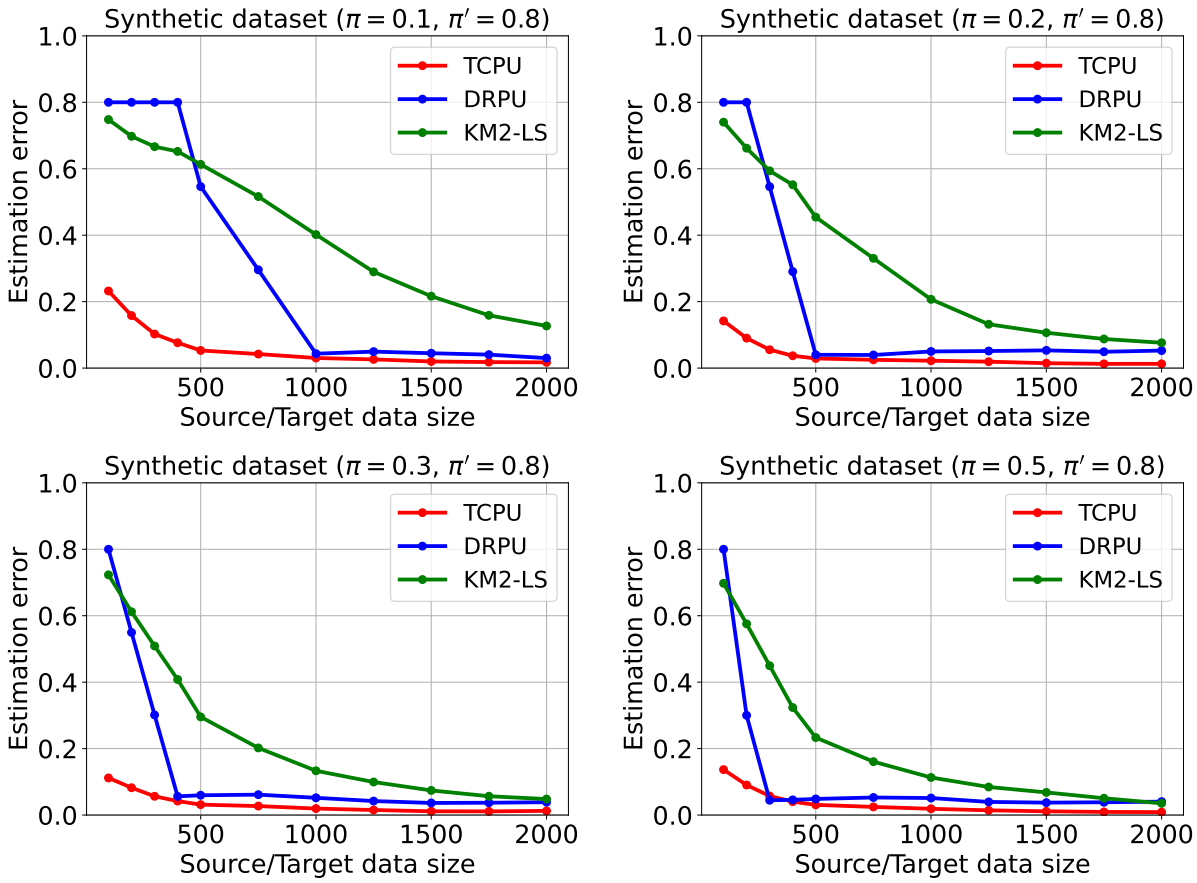

Figure 11: Estimation errors wrt size of the source data for synthetic dataset and $c = 0.5$. We assume that the size of the target data is equal to the size of the source data.

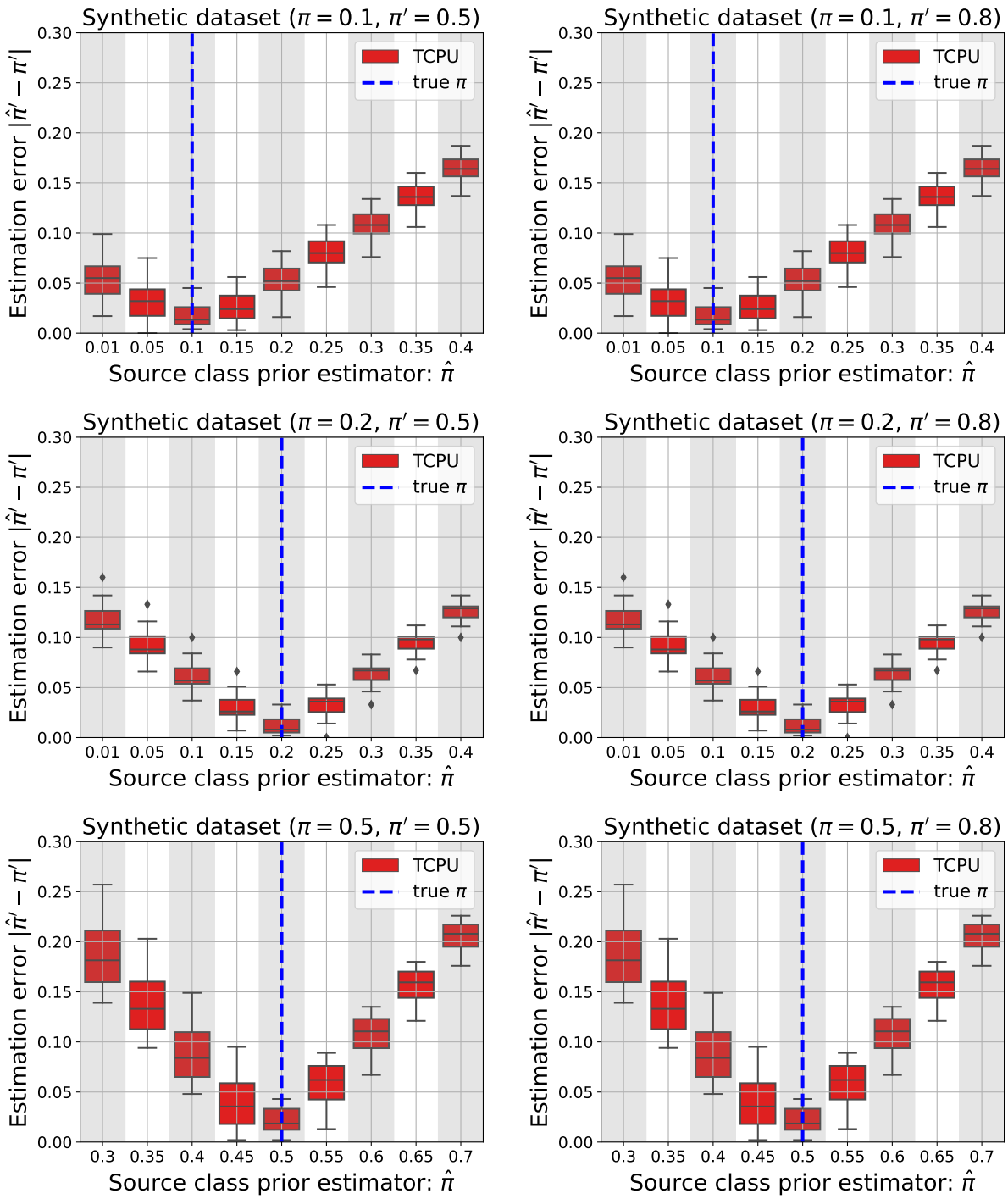

Figure 12: The impact of π estimation on the performance of the TCPU estimator. The boxplots show estimation errors for TCPU target class prior estimator $|\widehat{\pi}' - \pi'|$, for different source class prior estimators $\hat{\pi}$.

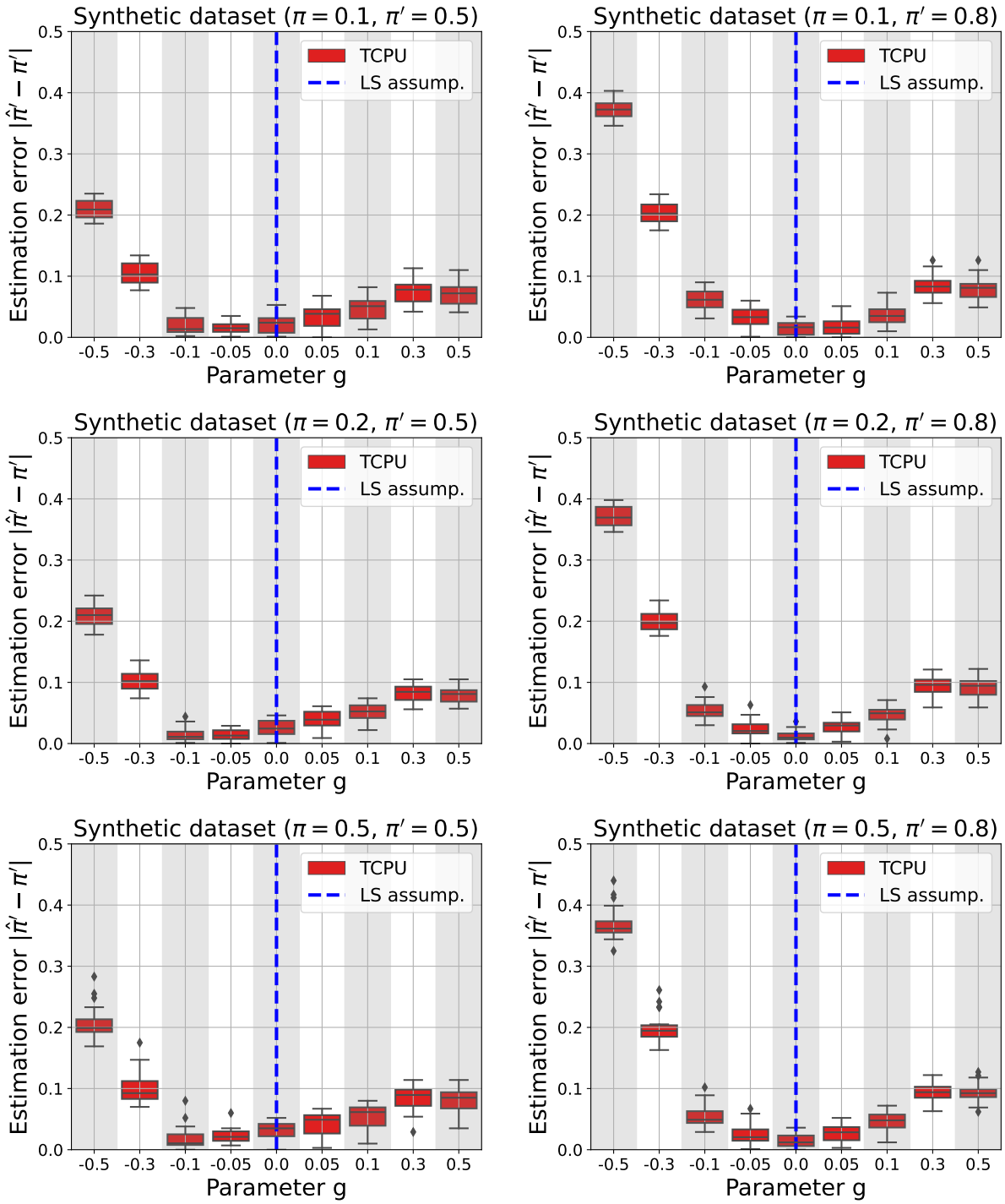

Figure 13: Robustness to violation of the Label Shift (LS) assumption. The blue vertical line corresponds to the situation when the assumption is met, and non-zero values of the parameter g indicate a violation of the assumption. Boxplots show the distributions of TCPU estimation errors for different values of the parameter g.