Bagging Propensity Weighting: A Robust method for biased PU Learning

Sander De Block SANDER.DEBLOCK02@GMAIL.COM
Jessa Bekker JESSA.BEKKER@KULEUVEN.BE
Department of Computer Science, Leuven.AI, KU Leuven, Belgium

Editor: Nuno Moniz, Paula Branco, Luís Torgo, Nathalie Japkowicz, Michał Woźniak and Shuo Wang.

Abstract

Propensity weighting enables learning from positive and unlabeled data (PU learning) in the face of labeling bias. PU learning aims to train a binary classification model when only positive and unlabeled data is available to learn from. This problem setting arises commonly in practice. Often, PU data suffers from a labeling bias, where the labeled examples are a biased sample from the positive examples. The probability for a positive example to get selected to be labeled is called its propensity score. Weighting PU datasets using propensity scores, allows to learn an unbiased model from biased PU data. However, this method has a strong downside of being rather unstable. This paper proposes a robust method for learning from biased PU data based on bagging. We show that the proposed method remains unbiased, while it reduces the variance and hence increases robustness. Our experiments confirms this by showing that our method has lower variance and classification error than plain propensity weighting as well as another method that was proposed for variance reduction.

Keywords: PU Learning, labeling bias, propensity scores, propensity weighting, Semi-supervised learning, label noise, label imbalance

1. Introduction

The field of Learning from Positive and Unlabeled data (PU learning) aims to train binary classification models from training data that consists of positive and unlabeled examples, where the unlabeled examples can be either positive or negative. This setting arise often in practice. Consider recommendation systems that aims to recommend products of interest and are learned from click data. When someone clicks on a product, then they are definitely interested in the product (positive), but if they do not click on it (unlabeled) there might be a reason different from disinterest for not clicking on it. Another example is prediction of product defects, based on products that were returned. Not all defect products will be returned because the defect might not have been noticed yet, the defect could be light and not worth the effort of returning it, or the customer may be too lazy to return it.

In both previous examples, the labeled examples are a biased subset of the positive examples. One is more likely to click on a product if they are more interested in it, they are more exposed to it, or if it is better located on a page. Similarly, products with stronger defects or more expensive products may be more likely to be returned. The past 5 years, the PU learning field has shown an increasing interest in learning in face of a labeling bias (Bekker et al., 2019; Kato et al., 2019; Saito et al., 2020; Gong et al., 2022; Gupta et al., 2021; Gerych et al., 2022; Schouterden et al., 2022). The labeling mechanism decides which positive examples get labeled and is quantified by propensity scores, which are the labeling probabilities. The propensity scores can be used to perform propensity weighting and so learn an unbiased model using the unbiased labels (Bekker et al., 2019). While the propensity weighting method is unbiased, it suffers from instability because of its potential high variance (Saito et al., 2020). Saito et al. (2020) propose a variance-reduction method based on clipping propensity scores, however this method reduces the variance at the cost of introducing a bias.

This work aims to make propensity weighting more robust by reducing its variance while keeping it unbiased. Concretely, we (i) propose a new method for propensity weighting based on bagging, (ii) proof that the ensemble models are unbiased, and (iii) empirically show that the proposed method indeed reduces the variance as well as the classification error, also outperforming the method proposed by Saito et al. (2020).

This paper is structured as follows. Section 2 discusses the related work. Section 3 explains the preliminaries, including an explanation of propensity weighting, its variance, and the bias-variance decomposition of loss functions. Section 4 introduces our proposed method and motivates it by providing insight on the instability issue in propensity weighting and why bagging may help to resolve this. Section 5 contains the empirical evaluation. Finally, section 6 concludes.

2. Related work

Most PU learning methods operate either under the Selected Completely At Random (SCAR) assumption, where the labeled examples are i.i.d. sampled from the positive examples, or, they rely on the separability of the classes. For learning under the SCAR assumption, the methods can broadly be categorized as class prior incorporation methods and biased learning methods. Class prior incorporation explicitly takes the true class prior into account when modeling the learning problem (Elkan and Noto, 2008; Du Plessis et al., 2015; Kiryo et al., 2017), where the class prior could be estimated from the PU data itself (Elkan and Noto, 2008; Scott, 2015; Ramaswamy et al., 2016; Bekker and Davis, 2018). Biased learning methods consider the labeled examples as negative, but assign different costs to the two classes (Liu et al., 2003; Lee and Liu, 2003; Mordelet and Vert, 2014; Claesen et al., 2015). Learning with separable classes is usually tackled with two-step techniques, where the first step identifies reliable negative examples as the unlabeled examples that are very different from any labeled examples and then performs standard (semi-)supervised learning using the reliable negative examples in the second step (Liu et al., 2002; Yu et al., 2002; Liu et al., 2003). For an overview of these methods, we refer to the survey of Bekker and Davis (2020). This paper, in contrast, makes neither of these assumptions. It assumes that the labeled examples are a biased sample of the positive examples.

Limited work has been done on PU learning with a labeling bias. The methods for this setting are largely divided into two categories: (a) the labeling mechanism is understood or (b) the labeling mechanism is unknown. This work falls in category (a). Bekker and Davis (2020) introduced this setting and proposed the propensity weighting method, which is built upon in this work. Similar methods have been proposed in the context of recommendation systems (Saito et al., 2020; Gupta et al., 2021) and knowledge base completion (Schouterden et al., 2022). Methods that assume an unknown labeling mechanism, typically simultaneously optimize two models: one for the classification and one for the labeling mechanism (Kato et al., 2019; Bekker et al., 2019; Gong et al., 2022; Gerych et al., 2022).

Saito et al. (2020) has shown that the propensity-weighted method can have a large variance in the presence of low propensity scores. They proposed to address this by clipping low propensity scores. This paper also aims to improve the robustness of propensity weighting by reducing its variance. In contrast to the clipping approach, our method does not introduce a bias in the effort of reducing the variance.

Our proposed method is a bagging ensemble approach. Bagging techniques have been shown to be beneficial under the SCAR assumption. Li and Zhang (2008) showed that bagging PU decision trees (Denis et al., 2005) improves their robustness and classification accuracy. Mordelet and Vert (2014) proposed bagging SVMs where all the labeled examples are always used as positive examples but different subsets of the unlabeled examples were sampled as the negative examples. The reasoning behind this is that the unlabeled examples can be considered contaminated negative examples, and by sampling them the contamination differs per learned model. This showed significantly better results over standard class-weighted SVMs (Liu et al., 2003). Claesen et al. (2015) further extended this approach by also sampling from the labeled set, which showed improvements if the labeled set also contained contamination.

3. Preliminaries

Let x be an example characterized by its features and y the indicator for its class: $y = 1$ means that x is positive, $y = 0$ that x is negative. s indicates whether x is labeled ($s = 1$) or not ($s = 0$). $\mathbf{x}$,$\mathbf{y}$, and $\mathbf{s}$ indicate sets of examples, classes and labels. A PU dataset D of size n consists of labeled examples $\langle x, s = 1\rangle$ and unlabeled examples $\langle x, s = 0\rangle$. The true class y is hidden, but from the PU property, it is known that all labeled examples are positive $\Pr(y = 1|s = 1, x) = 1$. This paper considers the single-training set scenario, where the dataset D is an i.i.d. sample of the population P (Elkan and Noto, 2008).[1] PU learning aims to train a binary classification model $f(x) = \hat{y}$ from D that approximates the true class y as close as possible.

Which of the positive examples are selected to be labeled depends on the stochastic labeling mechanism, characterized by the examples' propensity score $e(x) = \Pr(s = 1|y = 1, x)$. The propensity score of an example x is the probability that it would get labeled if the example were positive. Many PU learning approaches make the simplifying Selected Completely At Random (SCAR) assumption, where the labeling mechanism is constant, i.e., independent of the features $e(x) = \Pr(s = 1|y = 1, x) = \Pr(s = 1|y = 1) = c$ (Elkan and Noto, 2008). In contrast, this work generalizes to the Selected At Random (SAR) assumption, where the labeling mechanism depends on the features x and hence can be biased (Bekker et al., 2019).

[1] In contrast, in the case-control scenario, the set of unlabeled examples $D|s = 0$ is an i.i.d. sample of the population P and the set of labeled examples $D|s = 1$ is a (possibly biased) sample of the positive population $P|y = 1$.

3.1. Propensity Weighting for PU Learning

Propensity weighting is a method based on empirical risk minimization. Empirical-risk-minimization approaches aim to train a model that minimizes the empirical risk $R(\hat{\mathbf{y}})$, which is calculated from labeled data as follows:

$$R(\hat{\mathbf{y}}) = \frac{1}{n} \sum_{\langle x_i, y_i \rangle \in D} \delta_{y_i}(\hat{y}_i) = \frac{1}{n} \sum_{i=1}^{n} y_i \delta_1(\hat{y}_i) + (1 - y_i)\delta_0(\hat{y}_i), \tag{1}$$

where function $\delta_y(\hat{y})$ represents the cost for predicting $\hat{y}$ when the class is y. In this work, we consider the zero-one loss as the cost: $\delta_y(\hat{y}) = [\![\hat{y} = y]\!]$, where $[\![\hat{y} = y]\!]$ is an Iverson bracket evaluating to 1 iff the predicted class $\hat{y}$ equals the true class y, and 0 otherwise.

The empirical risk cannot directly be calculated in PU data, because there is no access to the true classes y. Bekker et al. (2019) proposed an estimator for the empirical risk that can be calculated using the propensity scores:

$$\hat{R}_{\text{PW}}(\hat{\mathbf{y}}|\mathbf{x}, \mathbf{s}) = \frac{1}{n} \sum_{\langle x_i, s_i \rangle \in D} s_i \left(\frac{1}{e(x_i)} \delta_1(\hat{y}_i) + (1 - \frac{1}{e(x_i)}) \delta_0(\hat{y}_i) \right) + (1 - s_i)\delta_0(\hat{y}_i). \tag{2}$$

This estimator is unbiased, i.e., $\mathbb{E}_{s\sim e}\left[\hat{R}_{\text{PW}}(\hat{\mathbf{y}}|\mathbf{x}, \mathbf{s})\right] = R(\hat{\mathbf{y}})$ (Bekker et al., 2019). Calculating $\hat{R}_{\text{PW}}(\hat{\mathbf{y}}|\mathbf{x}, \mathbf{s})$ corresponds to calculating the normal risk (Equation 1) on a weighted dataset derived from the PU dataset. In this weighted dataset, each labeled example is added once as a positive example with weight $\frac{1}{e(x)}$ and once as a negative example with weight $1 - \frac{1}{e(x)}$, each unlabeled example is added as a negative example with weight 1. This way, any risk-minimization based learner can be turned into a PU Learner by using the propensity-weighted dataset. Note, however, that the weight $1 - \frac{1}{e(x)}$ is negative, which in practice prevents some learning methods from being applicable.

The intuition behind the weighting scheme can be understood as follows: Each positive example x is selected to be labeled with probability $e(x)$, meaning that for each labeled example, there are expected to be $\frac{1}{e(x)}$ positive examples. The weighted labeled examples now correctly represent the positive distribution. The negative distribution is the difference between the total distribution and the positive distribution. The total distribution is correctly represented by the original dataset, i.e. each labeled and unlabeled example with weight 1. From these, the positive distribution needs to be substracted, i.e. each labeled example with weight $\frac{1}{e(x)}$.

Obtaining propensity scores Propensity weighting takes the propensity scores $e(x)$ of the labeled examples as input. However, in practice, it is unlikely that the exact propensity scores are known. Methods have been proposed to estimate the propensity scores from PU data simultaneously with learning the classification model (Kato et al., 2019; Bekker et al., 2019; Gong et al., 2022; Gerych et al., 2022). The underlying assumption is then that, either the model biases will be strong enough to separate the labeling mechanism from the classification (Bekker et al., 2019; Gong et al., 2022), that the labeling mechanism preserves the order induced by the class posterior (Gong et al., 2022; Gerych et al., 2022), or that there is a clear separation between the two classes (Gerych et al., 2022). Saito et al. (2020) takes a different approach and uses a proxy that can be calculated from the data to approximate the propensity scores. In their recommendation systems setting, they use relative item popularity as the proxy. Propensity score estimation is an area of research that deserves more attention, as good propensity estimates are required as input for propensity weighting. However, this goes beyond the scope of this work.

3.2. Variance of Propensity Weighting

While the propensity-weighted estimator is unbiased, it can suffers from high variance. Specifically when there are low propensity scores, as the variance depends on $\frac{1}{e(x)}$ (Saito et al., 2020):

$$\operatorname*{Var}_{s\sim e}\left[\hat{R}_{\mathrm{PW}}(\hat{\mathbf{y}}|\mathbf{x},\mathbf{s})\right]=\frac{1}{|D|^2}\sum_{\langle x_i,s_i\rangle\in D}\Pr(y=1|x_i)\left(\frac{1}{e(x_i)}-\Pr(y=1|x_i)\right)\left(\delta_1(\hat{y}_i)-\delta_0(\hat{y}_i)\right)^2. \tag{3}$$

To reduce the variance of the estimator, Saito et al. (2020) proposed to use clipped propensity scores $e'(x)=\max\big(e(x),M\big)$, with clip value M:

$$\hat{R}_{\mathrm{cl}}(\hat{\mathbf{y}}|\mathbf{x},\mathbf{s})=\frac{1}{n}\sum_{\langle x_i,s_i\rangle\in D}s_i\left(\frac{1}{e'(x_i)}\delta_1(\hat{y}_i)+(1-\frac{1}{e'(x_i)})\delta_0(\hat{y}_i)\right)+(1-s_i)\delta_0(\hat{y}_i). \tag{4}$$

This reduces the variance at the cost of an increased bias (proofed by Saito et al. (2020)):

$$\operatorname*{\mathbb{E}}_{s\sim e}\left[\hat{R}_{\mathrm{cl}}(\hat{\mathbf{y}}|\mathbf{x},\mathbf{s})\right]=R(\hat{\mathbf{y}})+\left|\frac{1}{n}\sum_{\langle x_i,s_i\rangle\in D}[\![e(x_i)>M]\!]\Pr(y=1|x_i)\left(\frac{e(x_i)}{M}-1\right)\left(\delta_1(\hat{y}_i)-\delta_0(\hat{y}_i)\right)\right| \tag{5}$$

$$\operatorname*{Var}_{s\sim e}\left[\hat{R}_{\mathrm{cl}}(\hat{\mathbf{y}}|\mathbf{x},\mathbf{s})\right]=\frac{1}{|D|^2}\sum_{\langle x_i,s_i\rangle\in D}\Pr(y=1|x_i)\left(\frac{1}{e'(x_i)}-\Pr(y=1|x_i)\right)\left(\delta_1(\hat{y}_i)-\delta_0(\hat{y}_i)\right)^2. \tag{6}$$

3.3. Bias-Variance Decomposition

To study whether the variance is indeed reduced using our method, we use the bias-variance decomposition. The expected loss of a learner on a set of datasets D sampled from a certain population P can be decomposed into three components: noise, bias, and variance (Domingos, 2000). The **noise** $N(x)$ is the unavoidable component of the loss. It is due to the difference between the optimal prediction y_* and true class y. The optimal prediction y_* for an example with features x is the prediction that minimizes the expected loss for examples with features x: $y_*=\operatorname{argmin}_{y_*}\mathbb{E}_P\left[\delta_y(y_*)\right]$. In case of zero-one loss, this is most common class y for an example with features x. The **bias** $B(x)$ is the component that specifies the average difference between the learned models and the optimal model. It is due to the difference between the main y_m and optimal prediction y_*. The main prediction y_m for an example with features x is the expected predicted class by models trained on different datasets $D \in \mathbf{D}$: $y_m = \text{argmin}_{y_m} \mathbb{E}_{\mathbf{D}}[\delta_y(y_m)]$. In case of zero-one loss, this is the most common prediction $\hat{y}$ for an example with features x. The **variance** V(x) specifies how much the models vary from being trained on one training set D to another in $\mathbf{D}$. It is due to the difference between a specific model's prediction $\hat{y}$ and main prediction y_m.

For a broad class of loss functions, including zero-one loss, the loss decomposes as follows:

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{D}}[\delta_y(\hat{y})] &= c_1 \cdot \mathop{\mathbb{E}}_{P}[\delta_y(y_*)] \quad &+ \quad \delta_{y_*}(y_m) \quad &+ \quad c_2 \cdot \mathop{\mathbb{E}}_{\mathbf{D}}[\delta_{y_m}(\hat{y})] \qquad (7)\\ &= c_1 \cdot N(x) \quad &+ \quad B(x) \quad &+ \quad c_2 \cdot V(x), \qquad (8)\end{aligned}$$

where, if the loss function is symmetric (as is the case for zero-one loss), $c_1 = 2\,\text{Pr}_{\mathbf{D}}(\hat{y} = y_*) - 1$ and $c_2 = 1$ if $y_m = y_*$ and $c_2 = -1$ otherwise. The average loss over all examples x is the sum of the noise, the average bias and the *net variance* $\mathbb{E}_x[c_2 V(x)]$:

$$\mathop{\mathbb{E}}_{\mathbf{D},\mathbf{x}}[\delta_y(\hat{y})] = \quad = \mathop{\mathbb{E}}_{x}[c_1 N(x)] \quad + \quad \mathop{\mathbb{E}}_{x}[B(x)] \quad + \quad \mathop{\mathbb{E}}_{x}[c_2 V(x)]\,, \qquad (9)$$

4. Robust Propensity Weighing through Bagging

This paper proposes a bagging-based method to enable more robust PU learning based on propensity weighting. The next section provides insight on the instability of plain propensity weighting and argues why bagging is a good way of addressing it. Section 4.2 details the proposed method.

4.1. The Instability Issue and Bagging Potential

Propensity weighting provides an unbiased estimator for the risk (Bekker et al., 2019). However, due to the variance of the estimator, the actual risk being optimized can still significantly differ from the true risk (Saito et al., 2020), leading to potentially learning bad models. The source of the variance lies with positive examples with low propensity scores. Small perturbations in the training set, as a result of different numbers of examples being labeled due to a low propensity score, thus result in large changes in the resulting model. This is illustrated by the following example.

Example 1 (Instability of propensity weighting) *Consider a dataset with a region consisting of* 40 *positive and 20 negative examples, and the propensity score of this region being* $e(x) = 0.05$. *Propensity weighting counts each labeled example as positive with* $1/0.05 = 20$. *If the number of labeled example in this region is the expected number* 2, *then weighting those 2 examples with 20 indeed results in the correct 40 positive examples. However, the probability of having exactly 2 examples labeled is only* 28%. *If by chance only 1 example gets labeled (with probability* 27%*), then only 20 positive examples will be counted in this region. With probability* 19% *3 examples get labeled and hence this region will be counted as completely positive. These three situations with very similar likelihoods of happening will thus result in completely different models.*

Breiman (1996) introduced the concept of bagging ensembles to decrease the variance of unstable learning algorithms. A learning algorithm is unstable if small perturbations in the training set result in large changes in the resulting model. This is exactly the issue of propensity-weighted learners.

To deal with the instability, bagging trains an ensemble of models, where each model is trained on a different dataset i.i.d. sampled from the training data. Because of the instability of the models, the variance between models might be large. When the models are used to make a prediction, the predictions of all the models are aggregated. This aggregation reduces the variance. This reduction in variance does not in general come at the cost of increasing the bias, because each of the models in the ensemble has the same expected bias as a model trained on the training set.

When sampling subsets of PU data, each subset of the data would have a varying proportion of labeled/unlabeled examples in the various data regions. If the proportion for the same region differs a lot between models, then combining the models will make sure that no extreme predictions are done for this region.

Grandvalet (2004) argued that bagging works because it equalizes the influence over all training instances. Datasets typically contain some leverage points, which have a large impact on the learned model. By taking samples of the original training dataset, not all leverage points are in the sampled set, which gives room for other data points to have an influence on this model of the ensemble and thus the resulting ensemble. In propensity-weighted PU Learning, low propensity labeled examples are such leverage points.

4.2. Bagging Propensity-Weighted Learners

To apply bagging to the propensity-weighted learner, an ensemble of k models is trained. Algorithm 1 details the process. Each model is trained on a subset of the examples: an i.i.d. sample D_s of the labeled examples with sample probability p_s, and i.i.d. sample $D_{\bar{s}}$ of the unlabeled samples with sample probability $p_{\bar{s}}$. Each of the models is trained by applying a learner that minimizes $\hat{R}_{\text{BPW}}$, a slight adaptation of the propensity-weighted risk estimator $\hat{R}_{\text{PW}}$. The models' predictions are combined using majority voting to make a final prediction.

```
**Algorithm 1:** Bagging Propensity-Weighted Learners (BPW)
**Input:** $D = \{\langle x_i, s_i \rangle\}^{i=1...n}$, $e(x)$, $k$, $p_s$, $p_{\bar{s}}$
**Output:** Ensemble of classification models $\mathbf{M}$
$\mathbf{M} \leftarrow \{\}$
**repeat** $k$ **times**
    $D_s \sim \{\langle x_i, s_i \rangle \in D | s_i = 0\}$, sampled with probability $p_s$
    $D_{\bar{s}} \sim \{\langle x_i, s_i \rangle \in D | s_i = 0\}$, sampled with probability $p_{\bar{s}}$
    $M \leftarrow$ train model from $D_s$ and $D_{\bar{s}}$ that minimizes $\hat{R}_{\text{BPW}}$ (Equation 10)
    $\mathbf{M} \leftarrow \mathbf{M} \cup M$
**end**
```

The sample probabilities p_s and $p_{\bar{s}}$ should be chosen such that there is enough variation between the different resulting datasets. If the sample probability is too large (too close to 1), then the low propensity examples will still appear in most subsamples, still carrying too much influence. The sample probability should not be too small either, so that each individual model can still learn something valuable. Due to the typically very low labeled-unlabeled example ratio in PU data, a suitable sample probability for unlabeled examples might be unsuitable for labeled examples and vice versa. Therefore, we propose to sample the labeled and unlabeled examples separately, using two hyper parameters p_s and $p_{\bar{s}}$.

Because the positive and unlabeled examples are sampled independently, the resulting sampled dataset is not an i.i.d. sample of the training data; the labeled-unlabeled ratio differs. Therefore, the propensity-weighted risk estimator $\hat{R}_{\text{PW}}$ (Equation 2) needs to be adjusted to reweight the examples using the sample probabilities to preserve their respective importance. This leads to the following risk estimator:

$$\hat{R}_{\text{BPW}}(\hat{\mathbf{y}}|\mathbf{e},\mathbf{s}) = \frac{1}{n}\sum_{\langle x_i,s_i\rangle\in D_s\cup D_{\bar{s}}} s_i\frac{1}{p_s}\left(\frac{1}{e(x_i)}\delta_1(\hat{y}_i) + (1-\frac{1}{e(x_i)})\delta_0(\hat{y}_i)\right) + (1-s_i)\frac{1}{p_{\bar{s}}}\delta_0(\hat{y}_i). \tag{10}$$

Proposition 1 (The BPW risk estimator is unbiased) *Let D_{PN} be a binary classification dataset of size n with tuples $\langle x, y\rangle$ where x are the feature vectors and y the true class labels. Let D be a PU dataset of size n corresponding to D_{PN} with tuples $\langle x, s\rangle$, where s indicates whether the example is labeled. y and s relate as follows: $\Pr(s = 1|y = 0, x) = 0$ and $\Pr(s = 1|y = 1, x) = e(x)$. Let D_s be an i.i.d. sample of the examples $\langle x, s = 1\rangle \in D$ with sample probability p_s and $D_{\bar{s}}$ an i.i.d. sample of the examples $\langle x, s = 0\rangle \in D$ with sample probability $p_{\bar{s}}$. Then $\hat{R}_{BPW}(\hat{\mathbf{y}}'|\mathbf{e}',\mathbf{s}')$, where $\hat{\mathbf{y}}',\mathbf{e}',\mathbf{s}'$ are the subsets of $\hat{\mathbf{y}},\mathbf{e},\mathbf{s}$ that are in $D_s \cup D_{\bar{s}}$, is an unbiased estimator for $R(\hat{\mathbf{y}}|\mathbf{y})$ calculated over D_{PN}:*

$$\mathop{\mathbb{E}}_{D_{PN}}\left[\hat{R}_{BPW}(\hat{\mathbf{y}}'|\mathbf{e}',\mathbf{s}')\right] = R(\hat{\mathbf{y}}|\mathbf{y}) \tag{11}$$

Proof

$$\begin{aligned}
&\mathop{\mathbb{E}}_{D_{PN}}\left[\hat{R}_{BPW}(\hat{\mathbf{y}}'|\mathbf{e}',\mathbf{s}')\right]\\
&\quad= \frac{1}{n}\sum_{\langle x_i,y_i\rangle\in D_{PN}} \Pr(s_i = 1|y_i, x_i)\Pr(x_i \in D_s|s_i = 1)\frac{1}{p_s}\left(\frac{1}{e(x_i)}\delta_1(\hat{y}_i) + (1-\frac{1}{e(x_i)})\delta_0(\hat{y}_i)\right)\\
&\qquad\qquad + \Pr(s_i = 0|y_i, x_i)\Pr(x_i \in D_{\bar{s}}|s_i = 0)\frac{1}{p_{\bar{s}}}\delta_0(\hat{y}_i)\\
&\quad= \frac{1}{n}\sum_{\langle x_i,y_i\rangle\in D_{PN}} y_i e(x_i) p_s\frac{1}{p_s}\left(\frac{1}{e(x_i)}\delta_1(\hat{y}_i) + (1-\frac{1}{e(x_i)})\delta_0(\hat{y}_i)\right)\\
&\qquad\qquad + (y_i(1-e(x_i)+(1-y_i))\, p_{\bar{s}}\frac{1}{p_{\bar{s}}}\delta_0(\hat{y}_i)\\
&\quad= \frac{1}{n}\sum_{\langle x_i,y_i\rangle\in D_{PN}} y_i e(x_i)\left(\frac{1}{e(x_i)}\delta_1(\hat{y}_i) + (1-\frac{1}{e(x_i)})\delta_0(\hat{y}_i)\right)\\
&\qquad\qquad + (y_i(1-e(x_i)+(1-y_i))\,\delta_0(\hat{y}_i)\\
&\quad= \frac{1}{n}\sum_{\langle x_i,y_i\rangle\in D_{PN}} y_i\delta_1(\hat{y}_i) + (1-y_i)\delta_0(\hat{y}_i) \quad = R(\hat{\mathbf{y}}|\mathbf{y})
\end{aligned}$$

■

Table 1: Datasets

Dataset name	#instances	#features	Pr(y=1)
Breast Cancer	683	9	0.35
Image Segmentation	2310	18	0.43
Mushroom	8124	111	0.48
Splice	3175	60	0.52
20 Newsgroups	8870	200	0.55

Limitations Aside from improved robustness, bagging propensity-weighted learners suffers from the same limitations as a single bagging propensity-weighted learner. The two largest limitations are (1) that the propensity scores are required as input, and (2) that the learner needs to be able to handle negative weights (as $1 - \frac{1}{e(x_i)} \leq 0$). Bagging introduces two additional limitations: (3) There are additional hyperparameters k, p_s and $p_{\bar{s}}$ to be set. Since, to the best of our knowledge, there has not yet been introduced a SAR PU score metric, it is not clear how to tune for them. (4) The training time takes k times longer than for training a single propensity-weighted learner.

5. Empirical Evaluation

This section shows that the method proposed in this paper indeed learns more robust models by bagging propensity weighted learners. To this end, it answers the following experimental questions:

Q1 Does our proposed bagging approach lead to a variance reduction w.r.t. the original propensity-weighted estimator, and in which propensity score settings is this more prominent?

Q2 Does the reduction in variance also lead to a reduction in classification error and in which settings?

Q3 Does our proposed bagging approach outperform the clipped estimator in terms of the bias, variance and classification error?

5.1. Experimental Setup

To answer the experimental questions, different learning methods are evaluated in different propensity score settings based on their empirical variance, bias and classification error (zero-one loss). To estimate the variance, bias and classification error, several instances of PU datasets are constructed from binary classification datasets, on which models are trained using different learning methods. The details are discussed below.

Data The datasets used for this experiments are 5 binary classification datasets that were also used by Bekker et al. (2019), using the same preprocessing[2]. The datasets are summarized in table 1.

[2] https://github.com/ML-KULeuven/SAR-PU

Table 2: Propensity score settings

Setting	Interval	Propensity scores per cluster
Low	$0.1 - 0.3$	$[0.11, 0.13, 0.15, 0.17, 0.19, 0.21, 0.23, 0.25, 0.27, 0.29]$
Varied	$0.1 - 0.95$	$[0.11, 0.2, 0.29, 0.38, 0.47, 0.56, 0.65, 0.74, 0.83, 0.92]$
High	$0.7 - 0.9$	$[0.7, 0.72, 0.74, 0.76, 0.78, 0.80, 0.82, 0.84, 0.86, 0.88]$

Constructing PU Datasets To turn the binary classification datasets into PU datasets, a labeling mechanism is constructed and applied to the datasets. The same type of labeling mechanisms are used for all datasets and 3 variants are proposed to enable studying the effect of the proposed methods in different settings. The dataset is clustered into 10 clusters using k-means clustering and to each cluster, a propensity score is assigned. I.e., there are 10 distinct propensity scores $[e_1, e_2, \ldots, e_{10}]$, each associated e_i associated with a cluster i. A positive example belonging to a cluster i is then selected to be labeled with probability e_i. The propensity score values are varied to induce the following 3 settings: low, varied and high propensity settings. For each setting, 10 propensity scores are defined, evenly spaced within the interval. 10 variations of each of the settings are introduced, by assigning the clusters to each of the scores in different ways using cyclic permutations of the propensity scores. The settings are described in table 2. For each propensity score assignment, 100 PU datasets are generated, i.e. 1000 PU datasets per propensity setting.

Learning methods Models are learned using the generated datasets, using the following methods:

- *BPW.* The method proposed in this work: A bagging ensemble of learners that minimize the propensity-weighted empirical risk estimator $\hat{R}_{\text{BPW}}$ (Equation 10). The hyperparameters are set as $p_s = 0.4$, $p_{\bar{s}} = 0.2$, $k = 50$.
- *PW.* A learner that minimizes the propensity-weighted empirical risk estimator $\hat{R}_{\text{PW}}$ (Equation 2) (Bekker et al., 2019). It uses the whole PU dataset and the propensity score function e to create its model.
- *Clipped.* A learner that minimizes the clipped propensity-weighted empirical risk estimator $\hat{R}_{\text{cl}}$ (Equation 4), with a clip value of $M = 0.2$ (Saito et al., 2020).
- *Sup.* A supervised learner that is trained using the real underlying classes y of the data. It can be considered as an upper bound on the possible performance.
- *Naive.* A naive learner that assumes that all labeled examples are positive and all unlabeled examples are negative.

All of the above methods use logistic regression as their learner, using the Scikit-Learn implementation with default parameters, except max_iter, which is set to 1000.

Evaluation The learning methods are evaluated on their classification error (zero-one loss) and its decomposition into bias and net variance. To this end, 100 random training-test splits are performed on each binary classification dataset, assigning 80% as training and 20% as test data. Each of the training sets is turned into a 30 PU dataset using the construction method described above, i.e. 1 PU dataset per propensity score assignment. The models trained on the PU training sets are evaluated on their corresponding test set. For each propensity score setting and dataset combination, there are thus 1000 training and test sets on which a model is trained and evaluated for each learning method.

The average classification error (zero-one loss) and its decomposition into the average bias and net variance are reported for each combination of propensity score setting, dataset and learning method, averaged over the test set predictions of 1000 models. The bias is the loss incurred by the main prediction (the most common prediction of the 1000 models) relative to the optimal prediction. However, since the optimal prediction is unknown, we will (naively) assume that the optimal prediction is the correct prediction y. The bias thus includes the noise as well. The variance is the average loss incurred by predictions relative to the main prediction. To obtain the net variance, the variance is multiplied by -1 if the main prediction was incorrect. The classification error is the sum of the bias and the net variance.

5.2. Results and Discussion

Table 3 shows the results, reporting the bias, net variance and classification error for each combination of the propensity score settings, datasets and learning methods. The results show that, as expected, the original PW method results in the lowest bias but has the highest variance. For all the methods that incorporate the bias, it can be observed that both the bias and variance consistently increase with lower propensity scores.

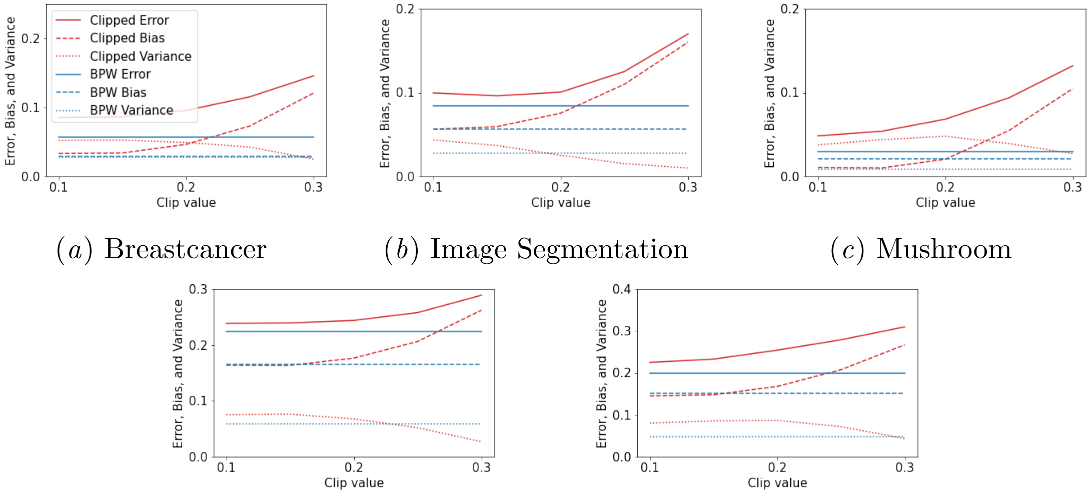
To answer **Q1**, the variance of the BPW and PW methods are compared. Q1 is answered affirmatively as BPW consistently has a lower variance than PW, with only 1 exception for Splice in the high propensity score setting. The difference between the variances becomes more prominent as the propensity scores are lower. The lower variance of the BPW does come at the cost of an increase in the bias. Q2 assesses whether the increase in bias is worth the decrease in variance.

To answer **Q2**, the classification error of the BPW and PW methods are compared. Q2 is answered affirmatively as both BPW has a lower classification error than PW in most cases. The classification error is lower for all datasets in the low and varied propensity score settings and for 3 out of 5 datasets in the high propensity score setting. The improvement is again more prominent as the propensity scores are lower.

To answer **Q3**, the bias, variance and classification error of the BPW and Clipped methods are compared. Here, the high propensity score setting can be omitted, as none of the propensity scores are clipped and hence no variance reduction method is applied. Q3 is answered affirmatively as BPW has lower bias in 4/5 datasets in both the varied and low propensity score settings, its variance is lower in 4/5 datasets in the low and all datasets in the varied propensity score settings, and its classification error is consistently lower for all datasets in both settings. That the bias would increase for the clipped method was expected, as the clipped estimator for the risk is a biased estimator. The reduction in variance depends on the clip value being used. Figure 1 shows the influence of this parameter in the low propensity score setting, by varying it between 0.1 and 0.3. As the clip value increases, the variance decreases as expected, sometimes going below the variance

Table 3: The “bias + net variance = classification error” for each combination of the propensity score settings, datasets and learning methods. The minimum (best) value among the methods that incorporate the propensity scores is formatted in bold. While the original propensity weighting method (PW) shows the least bias, the proposed bagging approach (BPW) reduces the variance, especially when low propensity scores are present, which results in a lower classification error.

Propensity Score Setting	Dataset	BPW	PW	Clipped	Sup	Naive
Low	Breastcancer	**.030**+**.028**=**.058**	.033+.052=.085	.046+.049=.096	.031+.001=.032	.349-.005=.344
	Image Segment.	.057+.031=**.088**	**.056**+.043=.100	.075+**.025**=.100	.048-.001=.047	.428-.001=.428
	Mushroom	.022+**.008**=**.030**	**.011**+.037=.048	.020+.048=.068	.000+.000=.000	.482-.001=.481
	Splice	.167+**.059**=**.225**	**.164**+.075=.239	.176+.067=.244	.151+.004=.155	.518-.005=.513
	20 Newsgroups	.153+**.047**=**.200**	**.145**+.080=.225	.168+.087=.254	.129+.002=.131	.549-.008=.541
Varied	Breastcancer	**.029**+**.014**=**.043**	.030+.031=.061	.038+.025=.063	.031+.001=.032	.205-.001=.204
	Image Segment.	.054+**.014**=**.068**	**.049**+.023=.072	.061+.015=.076	.048-.001=.047	.260+.001=.261
	Mushroom	.015+**.006**=**.021**	**.005**+.026=.032	.018+.023=.041	.000+.000=.000	.231+.002=.233
	Splice	.162+**.035**=**.197**	**.158**+.044=.203	.162+.036=**.197**	.151+.004=.155	.365-.003=.363
	20 Newsgroups	.145+**.035**=**.180**	**.139**+.062=.201	.150+.058=.208	.129+.002=.131	.388-.000=.388
High	Breastcancer	**.028**+**.004**=**.032**	.030+.009=.039	.030+.009=.039	.031+.001=.032	.067+.008=.074
	Image Segment.	.052+**.002**=.053	**.046**+.003=**.049**	.046+.003=.049	.048-.001=.047	.086+.003=.088
	Mushroom	.005+**.003**=**.008**	**.002**+.010=.011	.002+.010=.011	.000+.000=.000	.000+.001=.002
	Splice	.157+.008=.165	**.155**+**.007**=**.162**	.155+.007=.162	.151+.004=.155	.184+.010=.194
	20 Newsgroups	.137+**.013**=**.150**	**.133**+.024=.157	.133+.024=.157	.129+.002=.131	.182+.015=.196



(*a*) Breastcancer (*b*) Image Segmentation (*c*) Mushroom

(*d*) Splice (*e*) 20 Newsgroups

Figure 1: The bias, net variance and classification error of the BPW and clipped methods for different clip values on different datasets in the low propensity score setting.

of BPW. However, the biases increases at a faster rate, leading to a worse classification error, which is always above the BPW classification error.

6. Conclusion

This paper introduced bagging propensity weighting, which was shown to be more robust than both plain and clipped propensity weighting. Low propensity scores make plain propensity weighting very unstable, due to a small difference in the number observed positive examples making a large difference in the presumed number of actual positive examples. Bagging methods have been shown to improve the robustness of unstable learners, while not affecting their bias. Indeed, bagging propensity weighting is unbiased and result in both a lower variance and classification error than plain and clipped propensity weighting. As expected, the improvement gets more prominent as the propensity scores decrease.

Acknowledgments

This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.

References

Jessa Bekker and Jesse Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence 2018*, 2018.

Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Machine Learning*, 109(4):719–760, 2020.

Jessa Bekker, Pieter Robberechts, and Jesse Davis. Beyond the Selected Completely At Random Assumption for Learning from Positive and Unlabeled Data. In *Proceedings of the 2019 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 71–85, 2019.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Marc Claesen, Frank De Smet, Johan A. K. Suykens, and Bart De Moor. A robust ensemble approach to learn from positive and unlabeled data using svm base models. *Neurocomputing*, 160:73–84, 2015.

François Denis, Rémi Gilleron, and Fabien Letouzey. Learning from positive and unlabeled examples. *Theoretical Computer Science*, 348(1):70–83, 2005.

Pedro Domingos. A unified bias-variance decomposition and its applications. In *Proceedings of International Conference on Machine Learning*, pages 231–238, 2000.

Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *Proceedings of International Conference on Machine Learning*, pages 1386–1394, 2015.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

Walter Gerych, Thomas Hartvigsen, Luke Buquicchio, Emmanuel O. Agu, and Elke A. Rundensteiner. Recovering the propensity score from biased positive unlabeled data. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 2022.

Chen Gong, Qizhou Wang, Tongliang Liu, Bo Han, Jane Jia You, Jian Yang, and Dacheng Tao. Instance-dependent positive and unlabeled learning with labeling bias estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:4163–4177, 2022.

Yves Grandvalet. Bagging equalizes influence. *Machine Learning*, 55:251–270, 2004.

Shantanu Gupta, Hao Wang, Zachary C. Lipton, and Yuyang Wang. Correcting exposure bias for link recommendation. In *Proceedings of International Conference on Machine Learning*, 2021.

Masahiro Kato, Takeshi Teshima, and Junya Honda. Learning from positive and unlabeled data with a selection bias. In *Proceedings of International Conference on Learning Representations*, 2019.

Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Proceedings of Neural Information Processing Systems*, pages 1675–1685, 2017.

Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of International Conference on Machine Learning*, pages 448–455, 2003.

Chen Li and Yang Zhang. Bagging one-class decision trees. In *Proceedings of Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 2, pages 420–423. IEEE, 2008.

Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. Partially supervised classification of text documents. In *Proceedings of International Conference on Machine Learning*, pages 387–394. Citeseer, 2002.

Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of Third IEEE International Conference on Data Mining*, pages 179–186, 2003.

Fantine Mordelet and J-P Vert. A bagging svm to learn from positive and unlabeled examples. *Pattern Recognition Letters*, 37:201–209, 2014.

Harish Ramaswamy, Clayton Scott, and Ambuj Tewari. Mixture proportion estimation via kernel embedding of distributions. In *Proceedings of International Conference on Machine Learning*, 2016.

Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020.

Jonas Schouterden, Jessa Bekker, Jesse Davis, and Hendrik Blockeel. Unifying knowledge base completion with pu learning to mitigate the observation bias. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 4137–4145, 2022.

Clayton Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Proceedings of Artificial Intelligence and Statistics*, pages 838–846, 2015.

Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. PEBL: positive example based learning for web page classification using svm. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–248, 2002.