Posistive-Unlabeled Learning via Optimal Transport and Margin Distribution

Nan Cao , **Teng Zhang*** , **Xuanhua Shi** and **Hai Jin**
National Engineering Research Center for Big Data Technology and System
Services Computing Technology and System Lab, Cluster and Grid Computing Lab,
School of Computer Science and Technology, Huazhong University of Science and Technology, China
{nan_cao, tengzhang, xhshi, hjin}@hust.edu.cn

Abstract

Positive-unlabeled (PU) learning deals with the circumstances where only a portion of positive instances are labeled, while the rest and all negative instances are unlabeled, and due to this confusion, the *class prior* can not be directly available. Existing PU learning methods usually estimate the class prior by training a nontraditional probabilistic classifier, which is prone to give an overestimation. Moreover, these methods learn the decision boundary by optimizing the *minimum margin*, which is not suitable in PU learning due to its sensitivity to label noise. In this paper, we enhance PU learning methods from the above two aspects. More specifically, we first explicitly learn a transformation from unlabeled data to positive data by entropy regularized *optimal transport* to achieve a much more precise estimation for class prior. Then we switch to optimizing the *margin distribution*, rather than the minimum margin, to obtain a label noise insensitive classifier. Extensive empirical studies on both synthetic and real-world data sets demonstrate the superiority of our proposed method.

1 Introduction

In ordinary supervised learning, training a classifier usually requires labeled instances from all classes, which may be impractical or need additional steps which will incur unaffordable cost in many real-world scenarios. For example, in personalized advertising tasks, the pages that are visited and clicked by the user can be treated as positive instances. However, the rest huge amount of unexplored pages may not be uninteresting, thus they can not be simply labeled as negative instances. Such problems lead to the research of *positive-unlabeled* (PU) learning [Elkan and Noto, 2008; Bekker and Davis, 2020], where training a classifier can only access a portion of positive instances while the remaining unlabeled data could be either positive or negative.

Due to the importance of such framework, it has attracted lots of interest from the machine learning community. So far, the development of PU learning can be roughly classified into three stages. The algorithms in the first stage [Liu *et al.*, 2002; Li and Liu, 2003; Tao *et al.*, 2008; Chaudhari and Shevade, 2012] take the separability and smoothness assumptions, and adopt a two-step strategy in which reliable negative instances are identified first, followed by a traditional (semi)-supervised learning approach. The algorithms in the second stage [Mordelet and Jean Philippe, 2014; Shao *et al.*, 2015; Du Plessis *et al.*, 2015b] directly treat PU learning as a biased learning problem, in which the unlabeled instances are viewed as noisy negative instances. In the last stage, the algorithms [Bekker and Davis, 2017; Kiryo *et al.*, 2017; Zhang *et al.*, 2019; Chang *et al.*, 2021] usually incorporate class prior to learn an unbiased classifier.

Existing PU learning algorithms either suffer an overestimation of class prior or simply assume it is given ahead of time. To obtain a more precise class prior, ℓ_1 divergence between the positive data and the whole data is employed to penalize the overestimation of class prior [Du Plessis *et al.*, 2015a]. Besides, kernel mean embedding has also been introduced to characterize the matching degree between these two distributions [Chang *et al.*, 2021]. Different from these methods, we explicitly learn a transformation from unlabeled data to positive data by entropy regularized *optimal transport* (OT) [Kantorovitch, 1958; Frogner *et al.*, 2015; Peyré *et al.*, 2019], which brings us a much more precise estimation of class prior. Moreover, most PU learning methods learn the decision boundary through optimizing the minimum margin, which is not suitable in PU learning because it is very sensitive to label noise. In this paper, we optimize the *margin distribution* (MD) [Zhang and Zhou, 2014], rather than the minimum margin, to obtain a label noise insensitive classifier enjoying much better generalization performance. To summarize, our contributions are

- It achieves a much more precise estimation of class prior by leveraging the entropy regularized OT.
- It utilizes the margin distribution optimization to alleviate the inevitable label noise in PU learning.
- It shows significantly better generalization performance on both synthetic and real-world data sets.

The rest of the paper is organized as follows. We first introduce some basic preliminaries, and then present the proposed methods. After that, we detail the optimization procedure, followed by the empirical studies. Finally we conclude the paper with future works.

*Contact Author

2 Preliminaries

We first present some notations utilized throughout the paper. The scalars are denoted by normal font letters (e.g., x, X). The boldface letters (e.g., $\boldsymbol{x}$, $\mathbf{X}$) denote the vectors and matrices respectively. The upper case letters with mathcal font (e.g., $\mathcal{S}$) indicate the sets. In particular, $[m]$ is defined as the integer set $\{1, 2, \ldots, m\}$.

Without loss of generality, for PU learning data set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[p]} \cup \{\boldsymbol{x}_j\}_{j\in[m]\setminus[p]}$, we assume the first p instances are positive, i.e., $y_i = 1$ for any $i \in [p] \triangleq \mathcal{P}$, while the remaining $u = m - p$ instances are unlabeled, i.e., y_j is unknown for any $j \in [m] \setminus [p] \triangleq \mathcal{U}$. The class prior is denoted by $\pi = \mathbb{P}(y = 1)$ in unlabeled data, and the mixed class-wise unlabeled data density can be formulated as

$$\mathbb{P}(\boldsymbol{x}; \pi) = \pi\mathbb{P}(\boldsymbol{x}|y = 1) + (1 - \pi)\mathbb{P}(\boldsymbol{x}|y = -1) \quad (1)$$

2.1 Optimal Transport

OT is a useful mathematical tool to evaluate the difference between pairs of probability distributions. It is the cost of transforming the source distribution to the target distribution. It has also been proven to be a well defined distance and can integrate with metric learning [Song *et al.*, 2017] to better exploit the rich geometric structure on the space of probability distributions, which has been applied in many machine learning methods [Ho *et al.*, 2017; Blondel *et al.*, 2018; Titouan *et al.*, 2019]. To make the computation efficient, the entropy regularized version of OT is proposed. It encourages dense transport coefficients and can help distinguish the instances sampling from different class distributions.

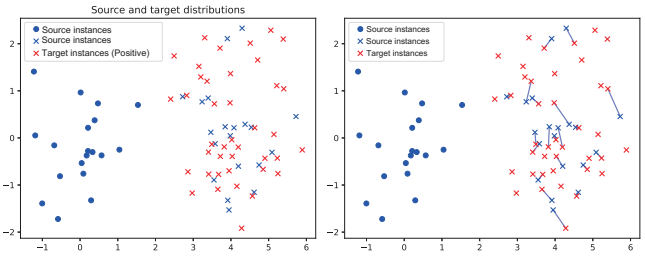
As shown in Figure 1(a), dots and crosses are from two class distributions. We randomly choose some red crosses as empirical target distributions and the rest with all dots as empirical source distributions, then utilize entropy regularized OT to transport source to target. In Figure 1(b), we connect the source instance with the target instance for which the largest transport coefficient exceeds the prespecified threshold. It can be seen that the blue crosses are connected with the red crosses while the blue dots are not. Thus we can identify the underlying crosses by checking the largest transport coefficient.

2.2 Optimal Margin Distribution Learning

Margin is one of the most essential concepts in machine learning. It indicates the confidence of the prediction results. Recent studies on margin theory [Gao and Zhou, 2013] demonstrate that margin distribution is crucial to generalization, and gives rise to a novel statistical learning framework named *optimal margin distribution machine* (ODM) [Zhang and Zhou, 2019]:

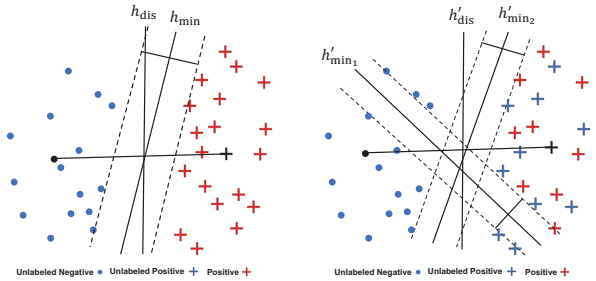
$$\min_{\boldsymbol{w},\xi_i,\epsilon_i} \frac{1}{2}\|\boldsymbol{w}\|^2 - \beta\bar{\gamma} + \alpha \sum_{i\in[m]} \frac{\xi_i^2 + \epsilon_i^2}{m}$$
$$\text{s.t. } \bar{\gamma} - \xi_i \le \gamma(\boldsymbol{x}_i, y_i) \le \bar{\gamma} + \epsilon_i, \ \forall i \in [m]$$

where $\gamma(\boldsymbol{x}_i, y_i)$ is the margin of $\boldsymbol{x}_i$, $\bar{\gamma}$ is the margin mean, α and β are trading-off hyperparameters. The first regularization term $\|\boldsymbol{w}\|^2/2$ controls the model complexity. The summation of slack variables ξ_i and ϵ_i in the last term is exactly the margin variance.



(a) Original data distribution (b) OT transportation plan

Figure 1: An illustration of entropy regularized optimal transport



(a) Supervised learning task (b) PU learning task

Figure 2: An illustration of optimizing minimum margin and margin distribution in PU learning task

Since more stable performance than the minimum margin based counterparts, ODM has been extended to many general learning settings [Zhang and Zhou, 2017; Zhang and Zhou, 2018a; Zhang and Zhou, 2018b; Zhang *et al.*, 2020; Zhang and Jin, 2021; Cao *et al.*, 2021]. Figure 2 illustrates the differences between ODM and SVM when solving the supervised and PU learning tasks respectively. For supervised learning, both h_{min} and h_{dis} achieve good performance. However, for PU learning, optimizing minimum margin only focuses on a small proportion of instances, which may result in multiple low-density decision boundaries, e.g., h'_{min_1} and h'_{min_2}, and a wrong selection will lead to a degenerated performance. Meanwhile, optimizing the whole margin distribution can help us avoid such a dilemma.

3 Class Prior Estimation by Regularized OT

The proposed estimation algorithm consists of two steps. In the first step, we utilize entropy regularized OT to identify the possible positive and negative instances in unlabeled data. Specifically, we treat positive instances as empirical target distribution $\boldsymbol{p}_t$, while unlabeled instances as empirical source distribution $\boldsymbol{p}_s$. All instances are assigned with equal mass density. Assume $\mathbf{C} = [C_{ij}]_{u\times p}$ is the cost matrix with the (i, j)-th entry $C_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\delta^2)$ indicating the cost of transporting one unit mass from $[\boldsymbol{p}_s]_i$ to $[\boldsymbol{p}_t]_j$, where δ is the width hyperparameter. Then the transportation from unlabeled instances to positive instances with minimum cost can be formalized as:

$$\min_{\mathbf{T}} \ \text{tr}(\mathbf{C}^\top\mathbf{T}) - \eta \cdot \Omega(\mathbf{T}) \quad \text{s.t. } \mathbf{T}\mathbf{1} = \boldsymbol{p}_s, \ \mathbf{T}^\top\mathbf{1} = \boldsymbol{p}_t \quad (2)$$

where $\mathbf{T} = [T_{ij}]_{u\times p}$ is the optimal transport matrix, $\Omega(\mathbf{T}) = -\sum_{ij} T_{ij}(\log T_{ij} - 1)$ is the entropy regularization term, and η is the trading-off hyperparameter. The problem of Eq. (2) have already been well studied and the unique solution can be obtained by the Sinkhorn-Knopp algorithm [Cuturi, 2013]. When we obtain transport matrix $\mathbf{T}$, we set a threshold σ to identify the reliable instances. Specifically, for each unlabeled instance $\boldsymbol{x}_i$, we treat it as a candidate positive instance if the maximal transport coefficient $\max_j T_{ij} \geq \sigma$, and vice versa. The intuition is that the underlying positive instances of unlabeled set and the positive data share the same distribution, thus the most likely positive instances will be firstly and concentratedly transported to the nearest positive instances.

In the second step, we utilize the discovered possible positive and negative instances to estimate the class prior. For PU learning, the distribution of unlabeled data can be approximately represented by a convex combination of positive and negative distributions as Eq. (1). Suppose the candidate positive instances are $\{\boldsymbol{x}_j \mid j \in \mathcal{C}_p\}$ and the candidate negative instances are $\{\boldsymbol{x}_k \mid k \in \mathcal{C}_n\}$. Then the class prior π can be estimated by minimizing the following distance between the unlabeled instance mean and the mixture of candidate positive and negative instance mean:

$$\min_{\hat{\pi}} \left\| \frac{1}{u}\sum_{i\in\mathcal{U}} \boldsymbol{x}_i - \frac{\hat{\pi}}{|\mathcal{C}_p|}\sum_{j\in\mathcal{C}_p} \boldsymbol{x}_j - \frac{1-\hat{\pi}}{|\mathcal{C}_n|}\sum_{k\in\mathcal{C}_n} \boldsymbol{x}_k \right\|^2 \tag{3}$$

Eq. (3) can also be optimized in some *reproducing kernel Hilbert space* (RKHS), i.e., all instances are embedded to this RKHS first via the inducing kernel mapping $\phi(\cdot)$, and then minimize the distance in this new feature space. In addition, Eq. (3) is a convex quadratic programming (QP), which can be easily solved via off-the-shelf QP solvers. The overall optimization procedure is summarized in Alg. 1. It is worth noting that we also return the candidate labels $\boldsymbol{y}_u^{(0)}$ for unlabeled instances to initialize the later optimization procedure.

4 Training with Margin Distribution

In this section, we detail the training of classifier.

4.1 Proposed Method

We first build a kNN-based graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set consisted of instances in $\mathcal{S}$, and $\mathcal{E}$ is the edge set representing the similarity between pairs of instances. We utilize $\mathcal{G}(\mathcal{V}, \mathcal{E})$ to exploit the relation between feature space and label space, and identify the ground-truth labels. We denote $\mathbf{A} = [A_{ij}]_{m\times m}$ as adjacency matrix of $\mathcal{G}(\mathcal{V}, \mathcal{E})$, and $A_{ij} = \exp(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\delta^2)$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_i$ are linked, and $A_{ij} = 0$ otherwise, and $\bar{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ as normalized adjacency matrix, where $D_{ii} = \sum_j A_{ij}$. We use $\bar{\mathbf{A}}$ to exploit relations between instances, and obtain:

$$\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon},\boldsymbol{y}} \frac{\|\boldsymbol{w}\|^2}{2} + \sum_{i\in[m]} \alpha_i \frac{\xi_i^2 + \mu\epsilon_i^2}{m(1-\theta)^2} + \tau \sum_{i,j\in[m]} \bar{A}_{ij}(y_i - y_j)^2$$

$$\text{s.t. } 1 - \theta - \xi_i \leq y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \leq 1 + \theta + \epsilon_i,\ i \in \mathcal{P} \tag{4}$$

$$1 - \theta - \xi_j \leq |y_j \boldsymbol{w}^\top \phi(\boldsymbol{x}_j)| \leq 1 + \theta + \epsilon_j \tag{5}$$

Algorithm 1 Class prior estimation

1: **Input:** PU data set $\mathcal{S}$, regularization hyperparameter η, threshold σ.
2: Calculate the cost matrix $\mathbf{C}$.
3: Obtain $\mathbf{T}$ by solving Eq. (2).
4: Find the candidate instances in unlabeled data.
5: Calculate the class prior $\hat{\pi}$ by solving Eq. (3).
6: **Output:** class prior $\hat{\pi}$, candidate labels $\boldsymbol{y}_u^{(0)}$.

$$\sum_j \mathbb{I}(y_j = 1) = u\hat{\pi},\ j \in \mathcal{U}$$

where only $\{y_i\}_{i\in\mathcal{U}}$ are optimization variables while $\{y_i\}_{i\in\mathcal{P}}$ are fixed, $\{\alpha_i\}_{i\in\mathcal{P}}$ and $\{\alpha_i\}_{i\in\mathcal{U}}$ indicate trade-off parameters for labeled and unlabeled data, $\mathbb{I}(\cdot)$ is the indicator function. In the above formulation, the second term as well as the Eq. (4) restrict margin variance of labeled positive instances, Eq. (5) imposes on the unlabeled instances to be accurately classified with less margin deviations. The final term is a smooth regularization letting the similar instances be more likely to have similar labels. Moreover, the final constraint ensures a proportion $\hat{\pi}$ of unlabeled instances to be positive.

4.2 Optimizing Classifier

If the labels of unlabeled data are available, our method reduces to the binary classification ODM [Zhang and Zhou, 2019]. Furthermore, because $\boldsymbol{y}$ is integer variable, it is difficult to optimize it together with $\boldsymbol{w}$, $\boldsymbol{\xi}$, $\boldsymbol{\epsilon}$. Thus we resort to the alternating optimization method.

Notice that $\boldsymbol{y}$ appears both in the objective function and the constraints, which makes the optimization difficult. To handle this problem, we apply the variable splitting technique by introducing an auxiliary variable $\boldsymbol{q} = [q_1, \ldots, q_m]$, and Eq. (4) becomes

$$\begin{aligned}\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon},\boldsymbol{y},\boldsymbol{q}} & \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i\in[m]} \alpha_i \frac{\xi_i^2 + \mu\epsilon_i^2}{m(1-\theta)^2} \\ & + \tau \sum_{i,j\in[m]} \bar{A}_{ij}(q_i - q_j)^2 + \lambda \sum_{i\in[m]} \mathbb{I}(y_i \neq q_i) \quad (6) \\ \text{s.t. } & 1 - \theta - \xi_i \leq y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \leq 1 + \theta + \epsilon_i \\ & \sum_{i\in[m]} \mathbb{I}(y_i = 1) = u\hat{\pi} + p,\ i \in [m]\end{aligned}$$

where the last term is set to make $\boldsymbol{y}$ and $\boldsymbol{q}$ as similar as possible and λ is a balancing parameter. Then, we need to solve the three subproblems in terms of $\boldsymbol{w}$, $\boldsymbol{y}$, and $\boldsymbol{q}$ respectively.

Optimizing $\boldsymbol{w}$. Fixing $\boldsymbol{\xi}$, $\boldsymbol{\epsilon}$, $\boldsymbol{y}$ and $\boldsymbol{q}$, we have

$$\begin{aligned}\min_{\boldsymbol{w}} & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{s.t. } & 1 - \theta - \xi_i \leq y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \leq 1 + \theta + \epsilon_i,\ i \in [m]\end{aligned} \tag{7}$$

Notice that the Eq. (7) is a convex quadratic programming (QP), which can be easily solved via off-the-shelf QP solvers.

Optimizing $\boldsymbol{\xi}$, $\boldsymbol{\epsilon}$ and $\boldsymbol{y}$. Fixing $\boldsymbol{w}$ and $\boldsymbol{q}$, we have

$$\begin{aligned}\min_{\boldsymbol{y},\boldsymbol{\xi},\boldsymbol{\epsilon}} & \sum_{i\in[m]} \alpha_i \frac{\xi_i^2+\mu\epsilon_i^2}{m(1-\theta)^2} + \lambda \sum_{i\in[m]} \mathbb{I}(y_i \neq p_i) \\ \text{s.t. } & 1-\theta-\xi_i \leq y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) \leq 1+\theta+\epsilon_i \\ & \sum_{i\in[m]} \mathbb{I}(y_i=1) = u\hat{\pi}+p,\ i\in[m]\end{aligned} \tag{8}$$

First we denote $\xi_{i,y_i} = \max(0, 1-\theta-y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i))$, $\epsilon_{i,y_i} = \max(0, y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i)-1-\theta)$, and $l_{i,y_i} = \xi_{i,y_i}^2 + \mu\epsilon_{i,y_i}^2$, then we define the cost matrix $\mathbf{B} = [B_{ij}]_{m\times 2}$ to distinguish the loss of positive and unlabeled data. For positive data, we set the i-th row $[B_{i1}, B_{i2}] = [E, l_{i,1}]$ where E is a large constant such as 10^3 specified in advance, while for unlabeled data, we set $[B_{i1}, B_{i2}] = [l_{i,-1}, l_{i,1}]$. Furthermore, we denote $\mathbf{Y} = [Y_{ij}]_{m\times 2}$ as the one-hot encoding of $\boldsymbol{y}$. We also define $\mathbf{Q} = [Q_{ij}]_{m\times 2}$ in the same way. Then Eq. (8) can be rewritten as

$$\begin{aligned}\min_{\mathbf{Y}} & \sum_{i\in[m]}\sum_{j\in[2]} \frac{\alpha_i}{m} Y_{ij}\left(\frac{\alpha_i}{m}B_{ij} - \lambda Q_{ij}\right) \\ \text{s.t. } & \sum_{j\in[2]} Y_{ij} = 1,\ i\in[m] \\ & \sum_{i\in[m]} Y_{i2} = u\hat{\pi}+p,\ Y_{ij}\in\{0,1\}\end{aligned} \tag{9}$$

Eq. (9) can be converted to a linear programming (LP) by relaxing $Y_{ij} \in \{0,1\}$ as $Y_{ij} \in [0,1]$ and solved by off-the-shelf LP solvers. Then the optimal y_i can be set as 1 if $Y_{i1} < Y_{i2}$ and -1 otherwise.

Optimizing $\boldsymbol{q}$. Fixing $\boldsymbol{w}$, $\boldsymbol{\xi}$, $\boldsymbol{\epsilon}$ and $\boldsymbol{y}$, we have:

$$\min_{\boldsymbol{q}} \ \tau \sum_{i,j\in[m]} \bar{A}_{ij}(q_i-q_j)^2 + \lambda \sum_{i\in[m]} \mathbb{I}(y_i \neq q_i)$$

Notice that the indicator function is discontinuous and difficult to optimize, we use ℓ_2 norm as a surrogate, and we have:

$$\min_{\boldsymbol{q}} \ \tau \sum_{i,j\in[m]} \bar{A}_{ij}\|q_i-q_j\|^2 + \lambda \sum_{i\in[m]} \|y_i-q_i\|^2 \tag{10}$$

The closed form solution of Eq. (10) is

$$\mathbf{Q} = \frac{\lambda}{\tau+\lambda}\left(\mathbf{I} - \frac{\tau}{\tau+\lambda}\bar{\mathbf{A}}\right)^{-1}\mathbf{Y}$$

We can obtain q_i according to Q_{ij} in a similar way as y_i.

Alg. 2 shows the whole optimization procedures.

Algorithm 2 Classifier training

Input: PU data set $\mathcal{S}$, kNN based graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, ODM hyperparameters μ, θ, α_i, trading-off hyperparameters λ, τ, estimated class prior $\hat{\pi}$, maximum iteration number T.
Initialize: $t=0$, $\boldsymbol{w}=\mathbf{0}$, $\boldsymbol{y}_u = \boldsymbol{q}_u = \boldsymbol{y}_u^{(0)}$, $[B_{i,1}, B_{i,2}] = [E, 0]$ for any $i \in \mathcal{P}$, $[B_{i,1}, B_{i,2}] = [1/2, 1/2]$ for any $i \in \mathcal{U}$.
while $t < T$ and not converge **do**
  Update $\boldsymbol{w}$, $\boldsymbol{\xi}$ and $\boldsymbol{\epsilon}$ by fixing $\mathbf{Y}$, $\mathbf{Q}$ and solve Eq. (7).
  Update coefficient matrix $\mathbf{B}$.
  Update $\mathbf{Y}$ and $\mathbf{Q}$ by solving Eq. (9) and Eq. (10).
  if converge **then**
    Break.
  end if
end while
Output: $\boldsymbol{w}$.

5 Experiments

In this section, we empirically show the effectiveness of our proposed method.

5.1 Data Sets and Settings

We perform experiments on both synthetic and real-world data sets. For real-world data sets, we utilize eight data sets from the UCI Machine Learning Repository. Their basic statistics are listed in Table 1.

Data sets	#Ins.	#Fea.	#Pos.	#Neg.
Australian	753	14	370	383
Diabetes	768	8	268	500
Banknote	1,372	5	762	610
Kr-vs-kp	3,196	37	1,669	1,527
Spambase	4,601	58	1,813	2,788
Musk	6,598	178	1,017	5,581
Mushroom	8,124	23	3,916	4,208
House	20,640	9	8,914	11,726

Table 1: Experimental data sets with their basic statistics

For synthetic data sets, we construct two Gaussian distributions centers set at $(0,0)$ and $(3,3)$, and the covariance matrixes of them are set to $\left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix}1 & -0.8\\ -0.8 & 1\end{smallmatrix}\right]$. It contains 400 labeled positive instances and 1000 unlabeled instances.

To demonstrate the superiority of our proposed PUOTMD, we compare it with three PU learning methods which also integrate class prior estimation algorithms EN [Elkan and Noto, 2008], PE [Du Plessis Marthinus and Sugiyama, 2014], and CAPU [Chang *et al.*, 2021], and six more PU learning methods WLR [Lee and Liu, 2003], PULD [Zhang *et al.*, 2019], UPU [Du Plessis *et al.*, 2015b], nnPU [Kiryo *et al.*, 2017],

π	EN	PE	CAPU	PUOTMD
0.3	.582±.037	.379±.024	.389±.103	**.307±.031**
	.917±.015•	.953±.017•	.951±.019•	**.978±.014**
0.5	.679±.241	.582±.016	.551±.022	**.519±.037**
	.901±.011•	.922±.014•	.938±.017•	**.949±.018**
0.7	.803±.047	.763±.019	**.719±.015**	.753±.011
	.890±.035•	.908±.021•	.943±.011	**.949±.018**

Table 2: Class prior estimation and classification accuracy on synthetic data sets. The best results are shown in bold. •/○ indicates that the performance of PUOTMD is significantly better/worse than the compared method (pairwise t-test at 0.05 significance level).

Data set	π	EN	PE	CAPU	PUOTMD	Data set	π	EN	PE	CAPU	PUOTMD
Australian	0.3	.459±.024 .787±.029•	.388±.018 .807±.024•	**.352±.015** .827±.017•	.355±.013 **.838±.019**	Spambase	0.3	.558±.027 .739±.029•	.406±.017 .789±.016○	**.383±.015** **.872±.013**	.388±.023 .871±.019
	0.5	.648±.027 .755±.031•	.599±.024 .801±.021•	.588±.014 .807±.016•	**.574±.016** **.819±.021**		0.5	.679±.022 .766±.028•	.586±.018 .765±.021•	.565±.013 .838±.025•	**.513±.017** **.897±.015**
	0.7	.781±.018 .708±.019•	.761±.023 .724±.028•	.739±.015 .758±.021•	**.715±.019** **.761±.022**		0.7	.815±.019 .711±.026•	.769±.016 .757±.023•	.785±.014 .825±.024•	**.684±.032** **.863±.024**
Diabetes	0.3	.589±.032 .507±.029•	.438±.027 .634±.022•	.331±.013 .739±.018•	**.325±.014** **.755±.019**	Musk	0.3	.449±.029 .871±.032•	**.337±.024** .890±.018•	.348±.013 .915±.019	.258±.018 **.924±.019**
	0.5	.735±.021 .622±.024•	.613±.012 .642±.022•	.609±.026 .702±.017•	**.567±.027** **.722±.024**		0.5	.627±.025 .851±.027•	.574±.021 .876±.028•	.553±.021 .873±.023•	**.521±.013** **.915±.029**
	0.7	.894±.014 .624±.025•	.759±.017 .657±.023•	.767±.023 .672±.019•	**.754±.014** **.701±.026**		0.7	.798±.023 .821±.019•	**.753±.017** .842±.022•	.779±.019 .838±.024•	.769±.024 **.864±.031**
Banknote	0.3	.503±.036 .881±.037•	.413±.024 .928±.021•	.358±.028 **.949±.019**	**.315±.018** .943±.016	Mushroom	0.3	.397±.014 .918±.023•	.331±.021 .921±.019•	**.294±.019** .928±.015	.281±.019 **.937±.021**
	0.5	.688±.031 .823±.029•	.597±.027 .879±.027•	.552±.017 .902±.029•	**.479±.027** **.934±.019**		0.5	.593±.023 .887±.019•	.544±.014 .902±.023•	**.531±.018** **.912±.026**	.442±.017 .901±.018
	0.7	.869±.034 .837±.029•	.752±.016 .893±.021	.776±.017 .889±.024•	**.674±.027** **.897±.019**		0.7	.819±.028 .869±.027•	.753±.019 .883±.022•	.749±.013 .901±.021	**.732±.025** **.903±.019**
Kr-vs-kp	0.3	.505±.034 .783±.029•	.389±.021 .816±.017•	.369±.019 .821±.023•	**.351±.015** **.841±.019**	House	0.3	.491±.019 .907±.013•	**.337±.022** **.923±.017**	.387±.027 .911±.016•	.358±.017 .911±.013
	0.5	.617±.027 .734±.034•	.576±.022 .772±.017•	.579±.019 .789±.024•	**.543±.021** **.801±.026**		0.5	.632±.038 .838±.024•	.597±.017 .849±.025•	.562±.019 .870±.022•	**.538±.014** **.887±.023**
	0.7	.834±.019 .718±.031•	.786±.019 .744±.017•	**.754±.021** **.759±.018**	.779±.028 .738±.026		0.7	.869±.028 .801±.027•	.798±.019 .838±.023•	**.742±.013** .849±.018•	.776±.024 **.863±.020**

Table 3: Class prior estimation and classification accuracy on real-world data sets. For each class prior, the first row shows estimated class prior value, while the second row is the classification accuracy. The best results are shown in bold. •/○ indicates that the performance of PUOTMD is significantly better/worse than the compared method (pairwise t-test at 0.05 significance level).

PUSB [Kato *et al.*, 2018], and LDCE [Shi *et al.*, 2018], which adopt different strategies to deal with PU learning problems and they all require a precisely given class prior to train the classifiers. The experiments are conducted with class prior from $\{0.3, 0.5, 0.7\}$. All data sets are randomly divided into training and test set with ratio 7:3, and we randomly select labeled and unlabeled instances according to *select completely at random* (SCAR) assumption [Bekker and Davis, 2020], i.e., the labeled instances are i.i.d. sampled from positive data distribution. We repeat the experiments ten times and record the average value and standard deviations of performance.

All hyperparameters of the baselines are tuned in the same way reported in the references. For CAPU and LDCE, their two trade-off hyperparameters are tuned from $\{2^{-5}, \ldots, 2^5\}$ and $\{10^{-3}, \ldots, 10^3\}$ respectively. The balancing hyperparameters in PULD are tuned from $\{10^{-3}, \ldots, 10^3\}$. As for our proposed method, we set the number of nearest neighbors $N = 10$, and the width of RBF kernel is selected from $\{2^{-10}, 2^{-9}, \ldots, 2^3\}$. We tune ODM hyperparameters μ and θ from the set $\{0.1, 0.2, \ldots, 0.9\}$, the threshold σ is set as $\min\{1, 10/p\}$, and the balancing hyperparameters are selected from $\{10^{-3}, \ldots, 10^3\}$.

5.2 Results

We first compare it with three PU learning methods EN, PE, and CAPU on synthetic data sets without giving class prior. The results are shown in Table 2. It can be seen that when class prior is 0.3 and 0.5, PUOTMD not only achieves a more precise class prior estimation but also produces a better classifier. When class prior is 0.7, PUOTMD only loses to CAPU in terms of class prior estimation, but still achieves the best classification accuracy.

We also conduct experiments on real-world data sets. Again we compare it with EN, PE, and CAPU without providing class prior. The results are shown in Table 3. It can be seen that EN is very prone to overestimate the class prior and thus getting the worst performance among the baselines. Our proposed PUOTMD achieves a more accurate class prior, which verifies the effectiveness of our proposed entropy regularized OT estimation method. Moreover, it outperforms other baselines on Diabetes in all class prior settings. In most cases, PUOTMD achieves both better class prior estimation and classification accuracy, but under the circumstance of less accurate estimation of class prior, PUOTMD can also achieve better classification accuracy, e.g., on House when class prior is 0.7.

Table 4 exhibits the classification accuracy of nine methods on eight data sets with given class priors. It can be seen that on Banknote, Spambase, and Musk, PUOTMD outperforms other methods in all class prior settings, and on other data sets, PUOTMD also performs better in most cases. Compared to PULD and EN, which adopt minimum margin framework, PUOTMD wins on Diabetes and Australian under all class prior settings, while on other data sets, it performs significantly better than EN in almost all the class prior settings, and beats PULD when class prior is 0.7. These

Data set	π	WLR	PULD	UPU	nnPU	CAPU	PUSB	EN	LDCE	PUOTMD
Australian	.3	.773±.022•	.826±.015•	.841±.009	.822±.009•	**.854±.013**	.795±.007•	.824±.009•	.811±.017•	.843±.004
	.5	.731±.021•	.790±.022•	.815±.016•	.816±.015	.811±.021	.779±.009•	.818±.007•	.779±.022 •	**.829±.007**
	.7	.677±.027•	.746±.017•	.769±.015•	.781±.017•	.802±.025•	.721±.015•	.775±.011•	.732±.027•	**.812±.007**
Diabetes	.3	.705±.019•	.741±.019•	.719±.013•	.707±.007•	**.775±.008**○	.743±.013•	.742±.013•	.732±.018•	.763±.009
	.5	.679±.022•	.722±.011•	.681±.019•	.689±.013•	.743±.012	.718±.017•	.683±.017•	.709±.016•	**.752±.011**
	.7	.631±.027•	.708±.023•	.643±.018•	.677±.009•	.711±.015•	.689±.028•	.639±.017•	.652±.022•	**.724±.013**
Banknote	.3	.952±.012•	.959±.013•	.955±.009•	.969±.011	.965±.027•	.951±.007•	.964±.009•	.966±.017	**.971±.009**
	.5	.924±.015•	.945±.006	.931±.014•	.940±.007•	.938±.017•	.927±.016•	.933±.013•	.937±.008•	**.953±.007**
	.7	.891±.017•	.908±.005•	.897±.013•	.906±.005•	.903±.031•	.891±.015•	.909±.017•	.902±.013•	**.929±.016**
Kr-vs-kp	.3	.813±.018•	.849±.015	.832±.014•	.824±.012•	.847±.022	.827±.019•	.837±.014•	.822±.017•	**.856±.013**
	.5	.796±.021•	**.826±.012**	.811±.011•	.803±.015•	.819±.017	.813±.016•	.811±.019•	.803±.015•	.824±.011
	.7	.778±.020•	.801±.009•	.781±.018•	.783±.014•	.798±.016•	.783±.024•	.789±.021•	.782±.022•	**.809±.016**
Spambase	.3	.879±.024•	.902±.011•	.889±.017•	.873±.016•	.906±.004•	.876±.007•	.821±.017•	.891±.017	**.912±.007**
	.5	.841±.029•	.887±.016	.807±.029•	.828±.012•	.883±.009•	.852±.009•	.801±.021•	.853±.019•	**.901±.005**
	.7	.802±.023•	.872±.010•	.784±.027•	.809±.015•	.841±.011•	.817±.007•	.772±.029•	.831±.027•	**.873±.012**
Musk	.3	.938±.014•	.938±.009	.925±.009•	.932±.007•	.922±.008•	.931±.014•	.933±.011•	.901±.011•	**.947±.009**
	.5	.921±.017	.911±.014•	.907±.013•	.901±.011•	.899±.014•	.914±.013•	.901±.008•	.874±.023•	**.929±.005**
	.7	.877±.016•	.881±.012•	.878±.018•	.872±.015	.878±.012•	.871±.021•	.883±.011•	.841±.017•	**.891±.011**
Mushroom	.3	.924±.013•	.952±.005	.923±.011•	.945±.009	.952±.015	.938±.014•	.947±.017•	.934±.013•	**.958±.005**
	.5	.901±.011•	.939±.012•	.911±.012•	.921±.007•	**.941±.009**	.925±.011	.931±.017	.917±.014•	.933±.003
	.7	.889±.019•	.923±.016•	.883±.014•	.902±.007•	.917±.013•	.912±.011•	.909±.013•	.891±.011•	**.922±.007**
House	.3	.917±.019•	.941±.009	.915±.011•	.908±.021•	.935±.028	.948±.015	.932±.007•	.922±.009•	**.958±.013**
	.5	.882±.015•	**.933±.014**	.873±.017•	.881±.018•	.898±.026•	.917±.009•	.909±.011•	.875±.021•	.929±.012
	.7	.841±.023•	.883±.013•	.822±.021•	.839±.024•	.906±.024•	.885±.013•	.897±.012	.831±.017•	**.908±.016**
w/t/l	.3	8/0/0	4/4/0	7/1/0	6/2/0	3/4/1	7/1/0	8/0/0	6/2/0	
	.5	7/1/0	4/4/0	8/0/0	7/1/0	4/4/0	7/1/0	7/1/0	8/0/0	
	.7	8/0/0	8/0/0	8/0/0	7/1/0	8/0/0	8/0/0	7/1/0	8/0/0	

Table 4: Classification accuracy on eight data sets. In each class prior scenario, the best result on each data set is shown in bold. •/○ indicates that the performance of PUOTMD is significantly better/worse than the compared method (pairwise t-test at 0.05 significance level). The win/tie/loss counts for PUOTMD are summarized in the last three rows.

verify that PUOTMD possesses better generalization performance than the minimum margin based PU learning methods. As for other baselines, PUOTMD only loses to CAPU in Diabetes when class prior is 0.3, and performs significantly better in most cases.

From the above experimental results, we can see that, in general, PU learning tasks where class prior is not given, our entropy regularized OT method estimates class prior more accurately, and achieves better classification performance. When class prior is available, our PUOTMD gets better classification accuracy in most cases, and shows better generalization performance than minimum margin based methods.

6 Conclusions

In this paper, we propose a novel PU learning method named PUOTMD, which consists of estimating class priors and training classifiers. Specifically, we first leverage the entropy regularized OT to adaptively discover the reliable positive and negative instances in unlabeled data, then utilize the mixture proportion estimation to produce an accurate class prior. Next, we jointly optimize the margin distribution and the labels of unlabeled data. Finally, we perform extensive experiments to verify its superiority. In the future, we will conduct theoretical analysis on our proposed method.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2020AAA0108501, the National Natural Science Foundation of China under Grant 62006088 and the Key R&D Program of Hubei under Grant 2020BAA020.

References

[Bekker and Davis, 2017] Jessa Bekker and Jesse Davis. Positive and unlabeled relational classification through label frequency estimation. In *ILP*, pages 16–30, 2017.

[Bekker and Davis, 2020] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: A survey. *Mach. Learn.*, 109(4):719–760, 2020.

[Blondel *et al.*, 2018] Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In *AISTATS*, pages 880–889, 2018.

[Cao *et al.*, 2021] Nan Cao, Teng Zhang, and Hai Jin. Partial multi-label optimal margin distribution machine. In *IJCAI*, pages 2198–2204, 2021.

[Chang *et al.*, 2021] Shizhen Chang, Bo Du, and Liangpei Zhang. Positive unlabeled learning with class-prior approximation. In *IJCAI*, pages 2014–2021, 2021.

[Chaudhari and Shevade, 2012] Sneha Chaudhari and Shirish Shevade. Learning from positive and unlabelled examples using maximum margin clustering. In *ICONIP*, pages 465–473, 2012.

[Cuturi, 2013] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS 26*, pages 2292–2300, 2013.

[Du Plessis *et al.*, 2015a] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Class-prior estimation for learning from positive and unlabeled data. In *ACML*, pages 221–236, 2015.

[Du Plessis *et al.*, 2015b] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *ICML*, pages 1386–1394, 2015.

[Du Plessis Marthinus and Sugiyama, 2014] Christoffel Du Plessis Marthinus and Masashi Sugiyama. Class prior estimation from positive and unlabeled data. *IEICE Trans. Inf. Syst.*, 97(5):1358–1362, 2014.

[Elkan and Noto, 2008] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *SIGKDD*, pages 213–220, 2008.

[Frogner *et al.*, 2015] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In *NIPS 28*, pages 2053–2061, 2015.

[Gao and Zhou, 2013] Wei Gao and Zhi-Hua Zhou. On the doubt about margin explanation of boosting. *Artif. Intell.*, 203:1–18, 2013.

[Ho *et al.*, 2017] Nhat Ho, XuanLong Nguyen, Mikhail Yurochkin, Hung Hai Bui, Viet Huynh, and Dinh Phung. Multilevel clustering via wasserstein means. In *ICML*, pages 1501–1509, 2017.

[Kantorovitch, 1958] Leonid Kantorovitch. On the translocation of masses. *Manage. Sci.*, 5(1):1–4, 1958.

[Kato *et al.*, 2018] Masahiro Kato, Takeshi Teshima, and Junya Honda. Learning from positive and unlabeled data with a selection bias. In *ICLR*, 2018.

[Kiryo *et al.*, 2017] Ryuichi Kiryo, Gang Niu, Marthinus du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *ICONIP*, pages 1674–1684, 2017.

[Lee and Liu, 2003] Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, pages 448–455, 2003.

[Li and Liu, 2003] Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI*, pages 587–592, 2003.

[Liu *et al.*, 2002] Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. Partially supervised classification of text documents. In *ICML*, pages 387–394, 2002.

[Mordelet and Jean Philippe, 2014] Fantine Mordelet and Vert Jean Philippe. A bagging SVM to learn from positive and unlabeled examples. *Pattern Recognit. Lett.*, 37:201–209, 2014.

[Peyré *et al.*, 2019] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Found. Trends Mach. Learn.*, 11(5-6):355–607, 2019.

[Shao *et al.*, 2015] Yuan-Hai Shao, Wei-Jie Chen, Li-Ming Liu, and Nai-Yang Deng. Laplacian unit-hyperplane learning from positive and unlabeled examples. *Inf. Sci.*, 314:152–168, 2015.

[Shi *et al.*, 2018] Hong Shi, Shaojun Pan, Jian Yang, and Chen Gong. Positive and unlabeled learning via loss decomposition and centroid estimation. In *IJCAI*, pages 2689–2695, 2018.

[Song *et al.*, 2017] Kun Song, Feiping Nie, Junwei Han, and Xuelong Li. Parameter free large margin nearest neighbor for distance metric learning. In *AAAI*, pages 2555–2561, 2017.

[Tao *et al.*, 2008] Peng Tao, Wanli Zuo, and Fengling He. SVM based adaptive learning method for text classification from positive and unlabeled documents. *Knowl. Inf. Syst.*, 16(3):281–301, 2008.

[Titouan *et al.*, 2019] Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *ICML*, pages 6275–6284, 2019.

[Zhang and Jin, 2021] Teng Zhang and Hai Jin. Optimal margin distribution machine for multi-instance learning. In *IJCAI*, pages 2383–2389, 2021.

[Zhang and Zhou, 2014] Teng Zhang and Zhi-Hua Zhou. Large margin distribution machine. In *SIGKDD*, pages 313–322, 2014.

[Zhang and Zhou, 2017] Teng Zhang and Zhi-Hua Zhou. Multi-class optimal margin distribution machine. In *ICML*, pages 4063–4071, 2017.

[Zhang and Zhou, 2018a] Teng Zhang and Zhi-Hua Zhou. Optimal margin distribution clustering. In *AAAI*, pages 4474–4481, 2018.

[Zhang and Zhou, 2018b] Teng Zhang and Zhi-Hua Zhou. Semi-supervised optimal margin distribution machines. In *IJCAI*, pages 3104–3110, 2018.

[Zhang and Zhou, 2019] Teng Zhang and Zhi-Hua Zhou. Optimal margin distribution machine. *IEEE Trans. Knowl. Data Eng.*, 32(6):1143–1156, 2019.

[Zhang *et al.*, 2019] Chuang Zhang, Dexin Ren, Tongliang Liu, Jian Yang, and Chen Gong. Positive and unlabeled learning with label disambiguation. In *IJCAI*, pages 4250–4256, 2019.

[Zhang *et al.*, 2020] Teng Zhang, Peng Zhao, and Hai Jin. Optimal margin distribution learning in dynamic environments. In *AAAI*, pages 6821–6828, 2020.