



City Research Online

City St George's, University of London

Citation: Wang, X., Yang, X., Zhu, R. & Xue, J-H. (2025). PUAL: A classifier on trifurcate positive-unlabelled data. *Neurocomputing*, 637, 130080. doi: 10.1016/j.neucom.2025.130080

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://city-ref-test.eprints-hosting.org/id/eprint/35160/>

Link to published version: <https://doi.org/10.1016/j.neucom.2025.130080>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

PUAL: A Classifier on Trifurcate Positive-Unlabelled Data

Xiaoke Wang^a, Xiaochen Yang^b, Rui Zhu^{c,*}, Jing-Hao Xue^a

^a*Department of Statistical Science, University College London, Gower Street, London, WC1E 6BT, England, UK*

^b*School of Mathematics & Statistics, University of Glasgow, University Place, Glasgow, G12 8QQ, Scotland, UK*

^c*Bayes Business School, City University of London, 106 Bunhill Row, London, EC1Y 8TZ, England, UK*

Abstract

Positive-unlabelled (PU) learning aims to train a classifier using the data containing only labelled-positive instances and unlabelled instances. However, existing PU learning methods are generally hard to achieve satisfactory performance on trifurcate data, where the positive instances distribute on both sides of the negative instances. To address this issue, firstly we propose a PU classifier with asymmetric loss (PUAL), by introducing a structure of asymmetric loss on positive instances into the objective function of the global and local learning classifier. Then we develop a kernel-based algorithm to enable PUAL to obtain non-linear decision boundary. We show that, through experiments on both simulated and real-world datasets, PUAL can achieve satisfactory classification on trifurcate data.

Keywords: Positive-unlabelled learning, Trifurcate data, Asymmetric loss

1. Introduction

PU learning is to train a classifier from PU data, which only contain labelled-positive instances and unlabelled instances, i.e., the PU data lack labelled-negative instances for training. For example, if we aim to train a classifier using some YouTube backend data to predict if a user likes a YouTuber, subscribers can be treated as positive instances. However, non-subscribers cannot be assumed as negatives, as they may like

*Corresponding author
Email address: rui.zhu@city.ac.uk (Rui Zhu)

7 the YouTuber without subscribing, leading to bias in the classifier and flawed recom-
8 mendation strategies. Recently, there are more and more PU data occurring in practice,
9 such as deceptive review detection [1], text categorization [2], remote sensing classifi-
10 cation [3, 4] and drug-target interaction prediction [5].

11 The main difficulty in PU learning is the lack of labelled-negative instances. A
12 natural way to deal with this issue is to pick the instances highly likely to be negative
13 from the unlabelled set and treat them as negative; then the obtained dataset will con-
14 tain labelled-positive, labelled-negative and unlabelled instances; and thus finally PU
15 learning can be converted to classical semi-supervised learning. The methods follow-
16 ing this idea are often referred to as two-step methods [6, 7, 8, 9], which can also be
17 generalised to have more steps of further iterative training [10, 11, 12, 13]. For ex-
18 ample, [7] introduced Bayesian optimal relabelling to the selection of reliable negative
19 instances, which improves instance-dependent PU learning by dynamically redefining
20 unlabelled samples through probabilistic relabelling; [9] expanded the selection of re-
21 liable negative instances with privileged information, which refines classification accu-
22 racy by incorporating auxiliary contextual data during training, addressing challenges
23 in domains like medical diagnostics.

24 However, the accuracy of such a multi-step method relies heavily on the accuracy
25 of the algorithm applied in the first step to pick reliable negative instances [14]. This
26 drawback motivated studies to train a PU classifier in a single step, termed one-step
27 methods [15, 16], which can be further categorised into inconsistent PU learning meth-
28 ods and consistent PU learning methods, depending on if the objective function is a
29 consistent estimator of an expected loss to classify an unknown instance from the
30 population.

31 A pioneer consistent PU learning method is the unbiased PU learning (uPU) [17].
32 Subsequently, the non-negative PU learning (nnPU) [18] was proposed by taking the
33 absolute value of the estimated average loss on the negative set in the objective func-
34 tion of uPU for better convergence of the classifier training. Furthermore, imbalanced
35 nnPU (imbalancednnPU) [19] was proposed to address imbalanced PU training set.
36 Nevertheless, the training of the consistent methods often required more information
37 out of the PU dataset, e.g., the class prior or the distribution of population.

Early attempt of the inconsistent PU learning methods was the biased support vector machine (BSVM) [20] based on the classic supervised support vector machine (SVM) [21], assigning a high weight to the average loss of the labelled-positive instances and a low weight to the average loss of the unlabelled instances in the objective function. Subsequently, weighted unlabelled samples SVM (WUS-SVM) [22] was proposed to assign a distinct weight to each of the unlabelled instances according to the likelihood of this unlabelled instance to be negative. Then, the biased least squares SVM (BLS-SVM) was proposed in [23]. It replaces the hinge loss with the squared loss in the objective of BSVM in case that too much importance is given to the unlabelled-positive instances treated as negative during the training of classifier. Then the local constraint was introduced to the objective function of BLS-SVM in [24], encouraging the instances to be classified into the same class as its neighborhoods and the proposed method is called the global and local learning classifier (GLLC). Moreover, the large-margin label-calibrated SVM (LLSVM) [25] was proposed to further alleviate the bias by introducing the hat loss to the objective function, where the hat loss measures the gap between the inverse tangent of the predictive score function (without intercept) and a certain threshold.

However, for trifurcate data where the positive set is roughly constituted by two subsets distributed on both sides of negative instances, existing PU learning methods are usually hard to achieve satisfactory performance. An example of the trifurcate PU datasets is illustrated in Fig. 1 by the 2-dimensional projection with the t-distributed stochastic neighbor embedding (t-SNE) [26] of dataset Wireless Indoor Localization (**wifi**) [27].

A classifier with non-linear decision boundary is needed for the classification of trifurcate datasets. However, when we apply kernel trick to obtain the non-linear decision boundary, the original trifurcate datasets may be converted to follow the pattern illustrated in Fig. 2, where the distances from the two positive subsets to the ideal decision boundary, as indicated by a solid blue line, are very different. In this case, using the squared loss (e.g. that used in GLLC) will impose quadratic penalty not only on the instances wrongly classified but also on all the instances correctly classified. Therefore, the labelled-positive instances correctly classified by, but far away from, the ideal

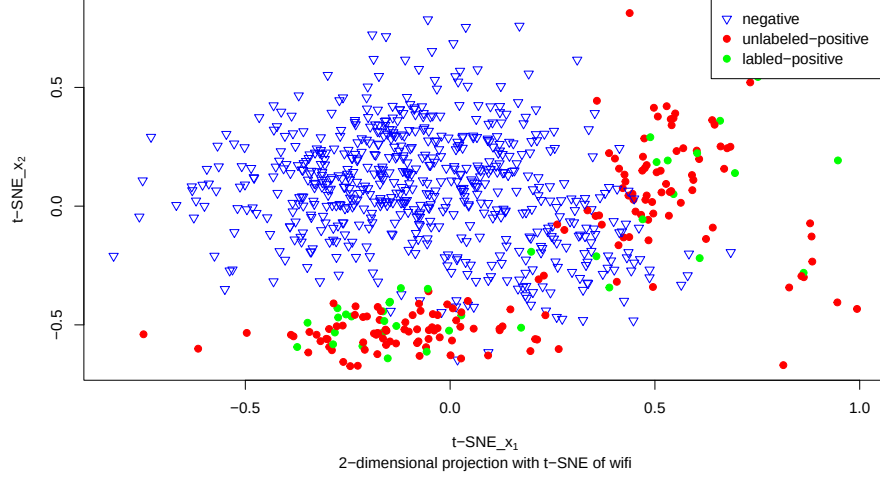


Figure 1: The 2-dimensional projection with t-SNE of dataset **wifi**, where the positive set is roughly constituted by two subsets distributed on both sides of negative instances; the perplexity for the training of t-SNE on **wifi** was set to 750.

69 decision boundary, as circled by the dashed lines in Fig. 2, can unfortunately generate
70 large penalty via the squared loss and hence drag the ideal decision boundary towards
71 the labelled-positive instances far away. This leads to the estimated decision boundary
72 indicated in green solid line, which misclassifies many more instances than the ideal
73 decision boundary.

74 The aim of applying the squared loss in the objective function of GLLC was to
75 ensure the importance given to the unlabelled-positive instances to be negative support
76 vectors to be lower than the importance given to unlabelled-negative instances to be
77 negative support vectors. Hence we note that it is not necessary to also use the squared
78 loss on the labelled-positive set. Furthermore, we note that, as the hinge loss does not
79 penalise the instances lying on the correct side of the margin, if the hinge loss is applied
80 to the labelled-positive set, the instances correctly classified but far away from the ideal
81 decision boundary will not generate any wrong penalty.

82 Therefore, motivated by the above analysis, we proposed a PU classifier with asym-
83 metric loss (PUAL) on positive instances for better classification on the trifurcate PU

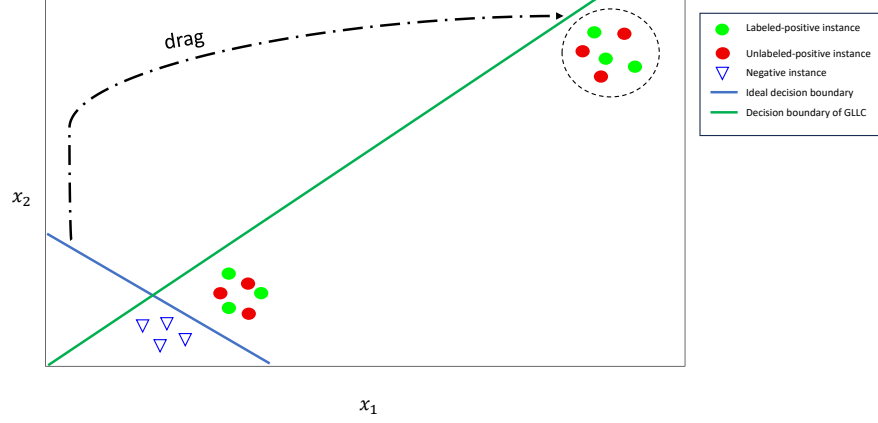


Figure 2: A pattern of the linearly separable space constructed from the original trifurcate PU datasets via the kernel trick; x_1 and x_2 represent the mappings of the features.

84 datasets. The contribution of this paper can be summarised as follows.

85 Firstly, in Section 3.1, we propose the methodology and algorithm of PUAL, which
 86 can generate a linear decision boundary much closer to the ideal decision boundary for
 87 the datasets following the pattern in Fig. 2 but first in the original feature space. This
 88 is achieved by using the hinge loss for the labelled-positive instances and the squared
 89 loss for the unlabelled-instances (and thus the unlabelled-positive instances).

90 Secondly, we develop a kernel-based algorithm to enable PUAL to obtain non-
 91 linear decision boundary in the original feature space, as detailed in Section 3.2.

92 Thirdly, we conduct experiments on both simulated and real-world datasets to
 93 clearly verify the motivation and effectiveness of our proposed methods, as presented
 94 in Section 4.

95 2. Related Work

96 Firstly we provide some general definitions involved in the following related works.
 97 Suppose there are n_p labelled-positive instances and n_u unlabelled instances with m fea-
 98 tures. Let feature matrix $X_{[pu]} = (\mathbf{x}_1, \dots, \mathbf{x}_{n_p}, \dots, \mathbf{x}_{n_p+n_u})^T \in \mathbb{R}^{(n_p+n_u) \times m}$, where the col-
 99 umn vector $\mathbf{x}_i \in \mathbb{R}^{m \times 1}$ denotes the vector of the features of the i th instance. Similarly,

matrix $\mathbf{X}_{[p]} = (\mathbf{x}_1, \dots, \mathbf{x}_{n_p})^T \in \mathbb{R}^{n_p \times m}$ denotes the feature matrix of the labelled-positive set while matrix $\mathbf{X}_{[u]} = (\mathbf{x}_{n_p+1}, \dots, \mathbf{x}_{n_p+n_u})^T \in \mathbb{R}^{n_u \times m}$ denotes the feature matrix of the unlabelled set. The methods listed in this section are all aimed to train the following predictive score function for classification:

$$f = \mathbf{x}^T \boldsymbol{\beta} + \beta_0, \quad (1)$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_m)^T \in \mathbb{R}^{m \times 1}$ and β_0 are the model parameters.

2.1. BSVM

In order to enable SVM to handle PU classification, BSVM [20] treats all the unlabelled instances as negative and assigns the loss of labelled-positive instances and the loss of unlabelled instances with different weights in the objective function. BSVM trains classifiers by solving the following objective function:

$$\begin{aligned} \min_{\boldsymbol{\beta}, \beta_0} & \frac{\lambda}{2} \boldsymbol{\beta}^T \boldsymbol{\beta} + C_p \mathbf{1}_p^T [\mathbf{1}_p - (\mathbf{X}_{[p]} \boldsymbol{\beta} + \mathbf{1}_p \beta_0)]_+ \\ & + C_u \mathbf{1}_u^T [\mathbf{1}_u + (\mathbf{X}_{[u]} \boldsymbol{\beta} + \mathbf{1}_u \beta_0)]_+, \end{aligned} \quad (2)$$

where C_p , C_u , and λ are positive hyper-parameters, $[g(\cdot)]_+$ indicates the column vector of the maximum between each element of $g(\cdot)$ and 0, and $\mathbf{1}_{p,u} = \underbrace{(1, 1, \dots, 1)^T}_k, k = n_p, n_u$.

2.2. BLS-SVM

One weakness of BSVM is that sometimes the hinge loss in the objective function of BSVM in Equation (2) selects more unlabelled-positive instances than unlabelled-negative instances to be the support vectors for the negative class, which constructs a decision boundary tending to misclassify the unlabelled-positive instances as negative. This is more likely to happen when there are many unlabelled-positive instances close to the unlabelled negative instances. BLS-SVM was proposed by [23] to force all training instances to contribute to the construction of the decision boundary of the trained SVM by solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\beta}, \beta_0} & \frac{\lambda}{2} \boldsymbol{\beta}^T \boldsymbol{\beta} + C_p [\mathbf{1}_p - (\mathbf{X}_{[p]} \boldsymbol{\beta} + \mathbf{1}_p \beta_0)]^T [\mathbf{1}_p - (\mathbf{X}_{[p]} \boldsymbol{\beta} + \mathbf{1}_p \beta_0)] \\ & + C_u [\mathbf{1}_u + (\mathbf{X}_{[u]} \boldsymbol{\beta} + \mathbf{1}_u \beta_0)]^T [\mathbf{1}_u + (\mathbf{X}_{[u]} \boldsymbol{\beta} + \mathbf{1}_u \beta_0)], \end{aligned} \quad (3)$$

where the squared loss replaces the hinge loss in BSVM on both labelled-positive set and unlabelled set. The objective function of BLS-SVM makes all the instances contribute to the construction of the decision boundary hence the importance given to the unlabelled-positive instances treated as negative is restricted [28].

2.3. GLLC

The similarities between a training instance and its neighbors can also be treated as a factor for classification, the idea of which is named local learning [29]. It is noted that the gap between PU learning and classical supervised learning on accuracy can be mitigated via GLLC [24], which is a combination of BLS-SVM and local learning. The objective function of GLLC is given as

$$\begin{aligned} \min_{\beta, \beta_0} & \frac{\lambda}{2} \beta^T \beta + C_p [\mathbf{1}_p - (X_{[p]} \beta + \mathbf{1}_p \beta_0)]^T [\mathbf{1}_p - (X_{[p]} \beta + \mathbf{1}_p \beta_0)] \\ & + C_u [\mathbf{1}_u + (X_{[u]} \beta + \mathbf{1}_u \beta_0)]^T [\mathbf{1}_u + (X_{[u]} \beta + \mathbf{1}_u \beta_0)] \\ & + (X_{[pu]} \beta + \mathbf{1}_{pu} \beta_0)^T \mathbf{R} (X_{[pu]} \beta + \mathbf{1}_{pu} \beta_0), \end{aligned} \quad (4)$$

where $\mathbf{R}$ is the similarity matrix for the instances and their neighbors, which can be calculated as follows.

Firstly we need to calculate matrix $\mathbf{W}$ by

$$w_{ij} = \begin{cases} \exp(-\sigma^{-1}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)) & \text{if the } i\text{th and } j\text{th instances are KNN of each other,} \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

where σ is a hyper-parameter to be selected. Then defining $\mathbf{1}_{pu} = \underbrace{(1, 1, \dots, 1)^T}_k, k = n_p + n_u$ and letting $\mathbf{w}_{\cdot i}$ denote the i th column of matrix $\mathbf{W}$ and $\mathbf{W}^*$ denote a diagonal matrix with the i th diagonal element equal to $\mathbf{1}_{[pu]}^T \mathbf{w}_{\cdot i}$, one can obtain

$$\mathbf{R} = \frac{1}{(n_p + n_u)} (\mathbf{W}^* - \mathbf{W}). \quad (6)$$

3. Methodology

3.1. PUAL with Linear Decision Boundary

3.1.1. Objective Function

As discussed in Section 1, we propose to apply the hinge loss to the labelled-positive instances, so as to impose no penalty to the instances correctly classified but far away from the ideal decision boundary. Meanwhile, the squared loss needs to be applied to the unlabelled instances in case that too much importance is given to the unlabelled-positive instances treated as negative during the training of the classifier. Hence we propose a structure of asymmetric loss on positive instances to revise GLLC, so that the new method PUAL is able to address the issue of classification of trifurcate PU data. Moreover, a local constraint term of similarity is helpful for PU classification. Therefore, the unconstrained optimization problem of PUAL can be formulated as

$$\begin{aligned} \min_{\beta, \beta_0} & \frac{\lambda}{2} \beta^T \beta + C_p \mathbf{1}_p^T [\mathbf{1}_p - (X_{[p]} \beta + \mathbf{1}_p \beta_0)]_+ \\ & + C_u [\mathbf{1}_u + (X_{[u]} \beta + \mathbf{1}_u \beta_0)]^T [\mathbf{1}_u + (X_{[u]} \beta + \mathbf{1}_u \beta_0)] \\ & + (X_{[pu]} \beta + \mathbf{1}_{pu} \beta_0)^T R (X_{[pu]} \beta + \mathbf{1}_{pu} \beta_0). \end{aligned} \quad (7)$$

However, the hinge loss term $\mathbf{1}_p^T [\mathbf{1}_p - (X_{[p]} \beta + \mathbf{1}_p \beta_0)]_+$ in the objective function of PUAL in Equation (7) is not always differentiable in the feasible region of the optimization, bringing difficulty to applying the gradient descent directly. To find an alternative way for solving PUAL, the following reformulation of Equation (7) can be considered:

$$\begin{aligned} \min_{\beta, \beta_0, \mathbf{h}} & C_p \mathbf{1}_p^T [\mathbf{h}]_+ + C_u (\mathbf{1}_u + X_{[u]} \beta + \mathbf{1}_u \beta_0)^T (\mathbf{1}_u + X_{[u]} \beta + \mathbf{1}_u \beta_0) \\ & + (X_{[pu]} \beta + \mathbf{1}_{pu} \beta_0)^T R (X_{[pu]} \beta + \mathbf{1}_{pu} \beta_0) + \frac{\lambda}{2} \beta^T \beta \\ \text{s.t. } & \mathbf{h} = \mathbf{1}_p - (X_{[p]} \beta + \mathbf{1}_p \beta_0). \end{aligned} \quad (8)$$

The predictive score function of PUAL for instance $\mathbf{x}$ is the same as the predictive score function of SVM in Equation (1).

3.1.2. Parameter Estimation

The convex objective function in Equation (8) can be regarded as the sum of the functions of (β, β_0) and the function of $\mathbf{h}$ while the constraints in Equation (8) can

159 be regarded as a linear combination of $(\boldsymbol{\beta}, \beta_0)$ and $\mathbf{h}$; this meets the requirement of
 160 ADMM [30, 31], which can decompose a large-scale convex optimization problem
 161 with affine constraints into several simpler sub-problems and update the solution iter-
 162 atively until convergence. Moreover, ADMM is able to converge to modest accuracy
 163 within fewer iterations than the gradient descent. Hence, we adopt it here.

164 The Lagrangian function of problem in Equation (8) is

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) = & C_p \mathbf{1}_p^T [\mathbf{h}]_+ + C_u (\mathbf{1}_u + \mathbf{X}_{[u]} \boldsymbol{\beta} + \mathbf{1}_u \beta_0)^T (\mathbf{1}_u + \mathbf{X}_{[u]} \boldsymbol{\beta} + \mathbf{1}_u \beta_0) \\ & + (\mathbf{X}_{[pu]} \boldsymbol{\beta} + \mathbf{1}_{pu} \beta_0)^T \mathbf{R} (\mathbf{X}_{[pu]} \boldsymbol{\beta} + \mathbf{1}_{pu} \beta_0) + \frac{\lambda}{2} \boldsymbol{\beta}^T \boldsymbol{\beta} \\ & + \mathbf{u}_h^T [\mathbf{1}_p - (\mathbf{X}_{[p]} \boldsymbol{\beta} + \mathbf{1}_p \beta_0) - \mathbf{h}], \end{aligned} \quad (9)$$

165 where $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \beta_0, \mathbf{h}, \mathbf{u}_h\}$ and $\mathbf{u}_h$ is dual variable. Then the augmented Lagrangian
 166 function is given as

$$\mathcal{L}_a(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) + \frac{\mu_1}{2} \|\mathbf{1}_p - (\mathbf{X}_{[p]} \boldsymbol{\beta} + \mathbf{1}_p \beta_0) - \mathbf{h}\|_2^2, \quad (10)$$

167 where μ_1 is the step-size coefficient of the augmented Lagrangian function.

168 It follows that the update of $\boldsymbol{\beta}$ and β_0 is

$$\begin{aligned} (\boldsymbol{\beta}^{(k+1)}, \beta_0^{(k+1)}) = & \arg \min_{\boldsymbol{\beta}, \beta_0} \frac{\lambda}{2} \boldsymbol{\beta}^T \boldsymbol{\beta} \\ & + C_u (\mathbf{1}_u + \mathbf{X}_{[u]} \boldsymbol{\beta} + \mathbf{1}_u \beta_0)^T (\mathbf{1}_u + \mathbf{X}_{[u]} \boldsymbol{\beta} + \mathbf{1}_u \beta_0) \\ & + (\mathbf{X}_{[pu]} \boldsymbol{\beta} + \mathbf{1}_{pu} \beta_0)^T \mathbf{R} (\mathbf{X}_{[pu]} \boldsymbol{\beta} + \mathbf{1}_{pu} \beta_0) \\ & + \mathbf{u}_h^{(k)T} [\mathbf{1}_p - (\mathbf{X}_{[p]} \boldsymbol{\beta} + \mathbf{1}_p \beta_0) - \mathbf{h}^{(k)}] \\ & + \frac{\mu_1}{2} \|\mathbf{1}_p - (\mathbf{X}_{[p]} \boldsymbol{\beta} + \mathbf{1}_p \beta_0) - \mathbf{h}^{(k)}\|_2^2, \end{aligned} \quad (11)$$

169 which is a quadratic optimization with every term differentiable. Let $\mathbf{I}_k, \forall k \in \mathbb{Z}$, denote
 170 a $k \times k$ identity matrix. By defining

$$\begin{aligned} \mathbf{M}_{11} &= \lambda \mathbf{I}_m + 2C_u \mathbf{X}_{[u]}^T \mathbf{X}_{[u]} + 2\mathbf{X}_{[pu]}^T \mathbf{R} \mathbf{X}_{[pu]} + \mu_1 \mathbf{X}_{[p]}^T \mathbf{X}_{[p]}, \\ \mathbf{M}_{12} &= 2C_u \mathbf{X}_{[u]}^T \mathbf{1}_u + 2\mathbf{X}_{[pu]}^T \mathbf{R} \mathbf{1}_{pu} + \mu_1 \mathbf{X}_{[p]}^T \mathbf{1}_p, \\ \mathbf{M}_{21} &= 2C_u \mathbf{1}_u^T \mathbf{X}_{[u]} + 2\mathbf{1}_{pu}^T \mathbf{R} \mathbf{X}_{[pu]} + \mu_1 \mathbf{1}_p^T \mathbf{X}_{[p]}, \\ \mathbf{M}_{22} &= 2C_u n_u + 2\mathbf{1}_{pu}^T \mathbf{R} \mathbf{1}_{pu} + \mu_1 n_p, \\ \mathbf{m}_1 &= -2C_u \mathbf{X}_{[u]}^T \mathbf{1}_u + \mathbf{X}_{[p]}^T \mathbf{u}_h + \mu_1 \mathbf{X}_{[p]}^T (\mathbf{1}_p - \mathbf{h}), \\ \mathbf{m}_2 &= -2C_u n_u + \mathbf{u}_h^T \mathbf{1}_p + \mu_1 (\mathbf{1}_p - \mathbf{h})^T \mathbf{1}_p, \end{aligned} \quad (12)$$

the solution of problem in Equation (11) can be obtained by solving the following linear equation w.r.t. β and β_0 :

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \beta^{(k+1)} \\ \beta_0^{(k+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1 \\ m_2 \end{bmatrix}. \quad (13)$$

Then the update of $\mathbf{h}$ becomes

$$\begin{aligned} \mathbf{h}^{(k+1)} = \arg \min_{\mathbf{h}} & C_p \mathbf{1}_p^T [\mathbf{h}]_+ + \mathbf{u}_h^{(k)T} [\mathbf{1}_p - (\mathbf{X}_{[p]} \beta^{(k+1)} + \mathbf{1}_p \beta_0^{(k+1)}) - \mathbf{h}] \\ & + \frac{\mu_1}{2} \|\mathbf{1}_p - (\mathbf{X}_{[p]} \beta^{(k+1)} + \mathbf{1}_p \beta_0^{(k+1)}) - \mathbf{h}\|_2^2, \end{aligned} \quad (14)$$

which is equivalent to solving the problem

$$\min_{\mathbf{h}} \sum_{i=1}^{n_p} \left\{ \frac{C_p}{\mu_1} [h_i]_+ + \frac{1}{2} \left[1 + \frac{u_{hi}^{(k)}}{\mu_1} - (\mathbf{x}_i^T \beta^{(k+1)} + \beta_0^{(k+1)}) - h_i \right]^2 \right\}. \quad (15)$$

According to [32], for constant $c > 0$, we can obtain

$$\arg \min_x c[x]_+ + \frac{1}{2} \|x - d\|_2^2 = \begin{cases} d - c & d > c, \\ 0 & 0 \leq d \leq c, \\ d & d < 0. \end{cases} \quad (16)$$

Thus, by defining $s_c(d) = \arg \min_x c[x]_+ + \frac{1}{2} \|x - d\|_2^2$, the i th element of $\mathbf{h}^{(k+1)}$ in problem in Equation (14) is solved as

$$h_i^{(k+1)} = s_{\frac{C_p}{\mu_1}} \left(1 + \frac{u_{hi}^{(k)}}{\mu_1} - (\mathbf{x}_i^T \beta^{(k+1)} + \beta_0^{(k+1)}) \right), i = 1, \dots, n_p. \quad (17)$$

According to [31], the update of the dual parameter $\mathbf{u}_h$ can be

$$\mathbf{u}_h^{(k+1)} = \mathbf{u}_h^{(k)} + \mu_1 [\mathbf{1}_p - (\mathbf{X}_{[p]} \beta^{(k+1)} + \mathbf{1}_p \beta_0^{(k+1)}) - \mathbf{h}^{(k+1)}]. \quad (18)$$

3.1.3. Algorithm

The algorithm of PUAL can be summarised in Algorithm 1.

3.2. PUAL with Non-Linear Decision Boundary

In this section, we develop a kernel-based algorithm to enable PUAL to have non-linear decision boundary, so that PUAL can be applied on the non-linear separable datasets including trifurcate PU datasets. The techniques applied in this section are similar to many previous methods [24, 33, 34, 35].

Algorithm 1 PUAL with linear decision boundary

Input: PU dataset, C_p , C_u , λ , σ and μ_1

Output: β and β_0

- 1: Initialise $\beta, \beta_0, \mathbf{h}, \mathbf{u}_h$
 - 2: **while** not converged **do**
 - 3: Update $(\beta^{(k+1)}, \beta_0^{(k+1)}) = \arg \min_{\beta, \beta_0} \mathcal{L}_a(\beta, \beta_0, \mathbf{h}^{(k)}, \mathbf{u}_h^{(k)})$
 - 4: Update $\mathbf{h}^{(k+1)} = \arg \min_{\mathbf{h}} \mathcal{L}_a(\beta^{(k+1)}, \beta_0^{(k+1)}, \mathbf{h}, \mathbf{u}_h^{(k)})$
 - 5: Update $\mathbf{u}_h^{(k+1)} = \mathbf{u}_h^{(k)} + \mu_1[\mathbf{1}_p - (X_{[p]}\beta^{(k+1)} + \mathbf{1}_p\beta_0^{(k+1)}) - \mathbf{h}^{(k+1)}]$
 - 6: **end while**
-

3.2.1. Objective Function

Suppose $\phi(\mathbf{x}) \in \mathbb{R}^{r \times 1}$ be a mapping of the instance vector $\mathbf{x}$. Then let $\phi(X_{[k]}) \in \mathbb{R}^{n_k \times r}$, $k = p, u, pu$ be the mapping of the original data matrix $X_{[k]}$. The i th row of $\phi(X_{[k]})$ is $\phi(\mathbf{x}_i)^T$. According to Equations (12) and (13), using $\phi(X_{[k]})$ as features matrix instead of $X_{[k]}$ for the training of PUAL, we can find the following necessary condition for the optimal solution of β :

$$\mathbf{B}\beta = \phi(X_{[pu]})^T \Omega, \quad (19)$$

where

$$\mathbf{B} = \mathbf{M}_{11} - \frac{\mathbf{M}_{12}\mathbf{M}_{21}}{\mathbf{M}_{22}}, \quad (20)$$

and

$$\Omega = \begin{bmatrix} \mathbf{u}_h - \mu_1 \frac{m_2}{M_{22}} \mathbf{1}_p + \mu_1 (\mathbf{1}_p - \mathbf{h}) \\ -2C_u \mathbf{1}_u - 2 \frac{m_2}{M_{22}} C_u \mathbf{1}_u \end{bmatrix} - 2 \frac{m_2}{M_{22}} \mathbf{R} \mathbf{1}_{pu}. \quad (21)$$

Equation (19) can be regarded as a condition when the objective function reaches its minimum. For $n_p, n_u > m$, $\mathbf{B}$ is symmetric and invertible. In this case, we can obtain

$$\beta = \mathbf{B}^{-1} \phi(X_{[pu]})^T \Omega. \quad (22)$$

Substituting Equation (22) into the objective function in Equation (7), we have the new

197 objective function as

$$\begin{aligned}
& \min_{\Omega, \beta_0} \frac{\lambda}{2} \Omega^T \phi(X_{[pu]}) B^{-1} B^{-1} \phi(X_{[pu]})^T \Omega + C_p \mathbf{1}_p^T [\mathbf{1}_p - (\phi(X_{[p]}) B^{-1} \phi(X_{[pu]})^T \Omega + \mathbf{1}_p \beta_0)]_+ \\
& + C_u [\mathbf{1}_u + \phi(X_{[u]}) B^{-1} \phi(X_{[pu]})^T \Omega + \mathbf{1}_u \beta_0] [\mathbf{1}_u + \phi(X_{[u]}) B^{-1} \phi(X_{[pu]})^T \Omega + \mathbf{1}_u \beta_0]^T \\
& + (\phi(X_{[pu]}) B^{-1} \phi(X_{[pu]})^T \Omega + \mathbf{1}_{pu} \beta_0)^T R (\phi(X_{[pu]}) B^{-1} \phi(X_{[pu]})^T \Omega + \mathbf{1}_{pu} \beta_0).
\end{aligned} \tag{23}$$

198 To prove $\phi(X_{[k]}) B^{-1} \phi(X_{[pu]})^T$ and $\phi(X_{[k]}) B^{-1} B^{-1} \phi(X_{[pu]})^T$ in Equation (23) are
 199 kernel matrices for $X_{[k]}$ and $X_{[pu]}$, we need to introduce the two closure properties for
 200 the construction of kernel functions proved in [36] as follows.

201 **Theorem 1.** *Let $\phi(X), \phi(Z)$ be a mapping of matrices of X, Z and $\kappa_1(\phi(X), \phi(Z))$ be a*
 202 *kernel matrix of $\phi(X)$ and $\phi(Z)$. Then the following two matrices $\kappa_2(X, Z)$ and $\kappa_3(X, Z)$*
 203 *can be regarded as the kernel matrix w.r.t. X, Z :*

- 204 • $\kappa_2(X, Z) = \kappa_1(\phi(X), \phi(Z))$,
- 205 • $\kappa_3(X, Z) = \mathbf{X} F \mathbf{Z}^T$, with F to be a symmetric matrix.

206 Then, according to the closure properties in Theorem 1, we can obtain

$$\begin{aligned}
\phi(X_{[k]}) B^{-1} \phi(X_{[pu]})^T &= \Phi'(\phi(X_{[k]}), \phi(X_{[pu]})) \\
&= \Phi(X_{[k]}, X_{[pu]})
\end{aligned} \tag{24}$$

207 and

$$\begin{aligned}
\phi(X_{[k]}) B^{-1} B^{-1} \phi(X_{[pu]})^T &= \Phi''(\phi(X_{[k]}), \phi(X_{[pu]})) \\
&= \Phi_2(X_{[k]}, X_{[pu]}),
\end{aligned} \tag{25}$$

208 where $\Phi'(\phi(X_{[k]}), \phi(X_{[pu]}))$, $\Phi''(\phi(X_{[k]}), \phi(X_{[pu]}))$ are the kernel matrices for $\phi(X_{[k]})$
 209 and $\phi(X_{[pu]})$, and $\Phi(X_{[k]}, X_{[pu]})$, $\Phi_2(X_{[k]}, X_{[pu]})$ are the kernel matrices for $X_{[k]}$ and
 210 $X_{[pu]}$.

211 Hence, the objective function of PUAL can be reformulated as

$$\begin{aligned}
& \min_{\Omega, \beta_0} \frac{\lambda}{2} \Omega^T \Phi_2(X_{[pu]}, X_{[pu]}) \Omega + C_p \mathbf{1}_p^T [\mathbf{1}_p - (\Phi(X_{[p]}, X_{[pu]}) \Omega + \mathbf{1}_p \beta_0)]_+ \\
& + C_u [\mathbf{1}_u + \Phi(X_{[u]}, X_{[pu]}) \Omega + \mathbf{1}_u \beta_0]^T [\mathbf{1}_u + \Phi(X_{[u]}, X_{[pu]}) \Omega + \mathbf{1}_u \beta_0] \\
& + (\Phi(X_{[pu]}, X_{[pu]}) \Omega + \mathbf{1}_{pu} \beta_0)^T R (\Phi(X_{[pu]}, X_{[pu]}) \Omega + \mathbf{1}_{pu} \beta_0),
\end{aligned} \tag{26}$$

212 whose solution is not related to $X[k], k = p.u.pu$ once the kernel matrices are deter-
 213 mined.

214 The predictive score function for instance $\mathbf{x}^*$ of PUAL can be now transformed to

$$f = \Phi(\mathbf{x}^*, X_{[pu]})\Omega + \beta_0. \quad (27)$$

215 3.2.2. Parameter Estimation

216 In this case, we can update β_0 via

$$\beta_0^{(k+1)} = \frac{m_2}{M_{22}} - \mathcal{Q}_b^{(k+1)}/M_{22}, \quad (28)$$

217 where m_2, M_{22} are not related to $X_{[p]}, X_{[u]}, X_{[pu]}$ and

$$\begin{aligned} \mathcal{Q}_b^{(k+1)} = & 2C_u \mathbf{1}_u^T \Phi(X_{[u]}, X_{[pu]})\Omega^{(k+1)} \\ & + 2\mathbf{1}_{pu}^T \mathbf{R} \Phi(X_{[pu]}, X_{[pu]})\Omega^{(k+1)} \\ & + \mu_1 \mathbf{1}_p^T \Phi(X_{[p]}, X_{[pu]})\Omega^{(k+1)}. \end{aligned} \quad (29)$$

218 The update of $\mathbf{h}, \mathbf{u}_h$ can be reformulated as

$$\begin{aligned} \mathbf{h}_i^{(k+1)} = & s_{\frac{C_p}{\mu_1}} \left(1 + \frac{u_{hi}^{(k)}}{\mu_1} - (\Phi(\mathbf{x}_i, X_{[pu]})\Omega^{(k+1)} + \beta_0^{(k+1)}) \right), i = 1, \dots, n_p, \\ \mathbf{u}_h^{(k+1)} = & \mathbf{u}_h^{(k)} + \mu_1 [\mathbf{1}_p - (\Phi(X_{[p]}, X_{[pu]})\Omega^{(k+1)} + \mathbf{1}_p \beta_0^{(k+1)}) - \mathbf{h}^{(k+1)}]. \end{aligned} \quad (30)$$

219 As $\Phi_2(X_{[k]}, X_{[pu]})$ does not directly appear in the update process for the optimiza-
 220 tion, we only need to determine the form of $\Phi(X_{[k]}, X_{[pu]})$ in practice. Moreover, λ
 221 either does not appear directly in the above update process or it is contained in the ma-
 222 trix $\mathbf{B}$ as a part of $\Phi(X_{[k]}, X_{[pu]})$. Therefore, for convenience, in the case of using the
 223 kernel trick in Section 4.1, we use λ to represent the hyper-parameter(s) of the kernel
 224 matrix $\Phi(X_{[k]}, X_{[pu]})$.

225 3.2.3. Algorithm

226 The algorithm of PUAL with non-linear decision boundary can be summarised in
 227 Algorithm 2.

228 4. Experiments

229 In this section, we conduct experiments on both synthetic data and real-world data
 230 to assess the classification performance of PUAL.

Algorithm 2 PUAL with non-linear decision boundary

Input: PU dataset, Φ , C_p , C_u , λ , σ and μ_1

Output: Ω and β_0

- 1: Initialise Ω , β_0 , $\mathbf{h}$, $\mathbf{u}_h$.
 - 2: **while** not converged **do**
 - 3: Update Ω via Equation (21) w.r.t. $\mathbf{h}^{(k)}$ and $\mathbf{u}_h^{(k)}$
 - 4: Update β_0 via Equation (28)
 - 5: Update $\mathbf{h}$, $\mathbf{u}_h$ via Equation (30)
 - 6: **end while**
-

4.1. Experiments on Synthetic Data

In this section, we conduct experiments on synthetic datasets following the pattern in Figure 2 to verify the superior performance of PUAL over GLLC. We generate positive-negative (PN) data from a multivariate normal distribution and split them into training and test sets under the single-training-set setting. Positive-unlabelled (PU) data are created from the training set, and the model is trained by using Algorithm 1. Hyperparameter tuning is performed via cross-validation, with the F1-score as the evaluation metric.

4.1.1. Generation of Synthetic Positive-Negative (PN) Datasets

The synthetic datasets following the pattern in Fig. 2 were obtained by the following steps:

1. To generate the first subset of the 2-dimensional synthetic positive set, 200 instances were sampled from the multivariate normal distribution with mean vector $(15, 15)$ and the covariance matrix $\begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}$.
2. To generate the second subset of the 2-dimensional synthetic positive set, 200 instances were sampled from the multivariate normal distribution with the mean vector $(\mathbf{mean}_{p2}, \mathbf{mean}_{p2})$ and the covariance matrix $\begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}$.

- 248 3. To generate the 2-dimensional synthetic negative set, 400 instances were sam-
 249 pled from the multivariate normal distribution of mean vector $(0, 0)$ and the co-
 250 variance matrix $\begin{bmatrix} 50 & 0.2 \\ 0.2 & 50 \end{bmatrix}$.
 251 4. Mixing the first subset of the synthetic positive set obtained in Step 1, the second
 252 subset of the synthetic positive set obtained in Step 2 and the synthetic negative
 253 set obtained in Step 3, a simple 2-dimensional synthetic dataset can be eventually
 254 obtained as shown in Fig. 2.

255 In Step 2, $\mathbf{mean}_{p2}$ took value from $(50, 100, 200, 500, 1000)$. For each value of
 256 $\mathbf{mean}_{p2}$, the above steps were repeated 5 times so that we have 5 synthetic datasets for
 257 each value of $\mathbf{mean}_{p2}$.

258 4.1.2. Training-Test Split for the Synthetic PU Datasets

259 It should be noted that both GLLC and PUAL can be applied on the datasets sam-
 260 pled from either single-training-set scenario [37] or case-control scenario [17] as we
 261 set the suitable metric for hyper-parameter tuning in practice. More specifically, the
 262 case-control scenario indicates that the unlabelled training set can be regarded to be
 263 i.i.d. sampled from the population, while the single-training-set scenario indicates that
 264 the whole training set can be regarded to be i.i.d. sampled from the population. In
 265 this case, for more intuitive comparison, we split each of the synthetic dataset gen-
 266 erated in Section 4.1.1 to construct the PU training and test sets consistent with the
 267 single-training-set scenario by the following two steps:

- 268 1. Firstly, to split the dataset into a training set and a test set, 70% of the instances
 269 in the simple synthetic dataset obtained in Section 4.1.1 were randomly selected
 270 as the training set while the test set was constituted by the rest 30% instances.
- 271 2. Secondly, to construct the labelled-positive set and unlabelled-set for training,
 272 25% of the positive instances in the above obtained training set were randomly
 273 selected to form the labelled-positive set for training. The rest of the positive
 274 instances were mixed with the negative set, contributing to the unlabelled set for
 275 training.

276 Then 25 pairs of PU training set and test set were obtained.

277 4.1.3. Model Setting

278 We note that the real value of F1-score on the training dataset is not accessible if
 279 we do not use the label information during model training. Therefore, by fixing C_p to
 280 1 and the number K of the nearest neighbors to 5, C_u , λ , σ in the objective functions
 281 of PUAL and GLLC were determined by 4-fold cross-validation (CV), which reached
 282 the highest average PUF-score proposed in [38] on the validation sets. PUF-score is
 283 similar to F1-score and can be directly obtained from PU data:

$$\text{PUF-score} = \frac{\text{recall}^2}{P[\text{sgn}(f(\mathbf{x})) = 1]}, \quad (31)$$

284 where ‘recall’ can be estimated by computing $\frac{1}{n_p} \sum_{\mathbf{x}_i \in X_{[p]}} \mathbb{I}(\text{sgn}(f(\mathbf{x}_i)) = 1)$ with the in-
 285 dicator function denoted by $\mathbb{I}(\cdot)$, and at the single-training-set scenario $P[\text{sgn}(f(\mathbf{x})) =$
 286 $1]$ can be estimated by computing $\frac{1}{n_p + n_u} \sum_{\mathbf{x}_i \in X_{[pu]}} \mathbb{I}(\text{sgn}(f(\mathbf{x}_i)) = 1)$. Furthermore, λ , σ
 287 were tuned from the set $\{1, 2, 3, 4, 5\} \circ \{0.1, 1, 10, 100\}$; C_u was selected from the set
 288 $\{0.01, 0.02, \dots, 0.5\}$ based on the setting in [24].

289 4.1.4. Results and Analysis

290 Results of the experiments on the constructed synthetic PU datasets are summarised
 291 in Table 1. The performance are measured by the average F1-score, which is a widely-
 292 used metric for the evaluation of PU learning methods [39]. The patterns of the deci-
 293 sion boundary obtained by PUAL and GLLC on the synthetic datasets are illustrated
 294 in Fig. 3. For each value of **mean**_{p2}, we use the first generated synthetic dataset as
 295 example.

296 According to the experimental results in Table 1, PUAL always has better per-
 297 formance than GLLC on the synthetic PU datasets with all the 5 values of **mean**_{p2}.
 298 Furthermore, with the value of **mean**_{p2} increasing, the gap between PUAL and GLLC
 299 becomes increasingly large. This is more clearly in the six plots in Fig. 3, as one of
 300 the positive subset becomes increasingly far away, the decision boundary of GLLC
 301 is dragged to a strange position that more and more instances are misclassified by
 302 the decision boundary of GLLC, while the decision boundary of PUAL is unaffected.
 303 Therefore, it is verified that PUAL can generate much better linear decision boundary
 304 than GLLC on the datasets following the pattern in Fig. 2.

Table 1: Summary of the average F1-score (%) with the standard deviation, from the experiments on the synthetic dataset; the result highlighted in blue is the better one between PUAL and GLLC.

mean_{p2}	PUAL	GLLC
50	95.07 ± 0.78	91.17 ± 1.52
100	94.89 ± 0.61	86.27 ± 1.83
200	93.56 ± 0.75	81.04 ± 2.93
500	92.83 ± 3.22	73.58 ± 2.58
1000	93.47 ± 2.08	71.28 ± 2.37

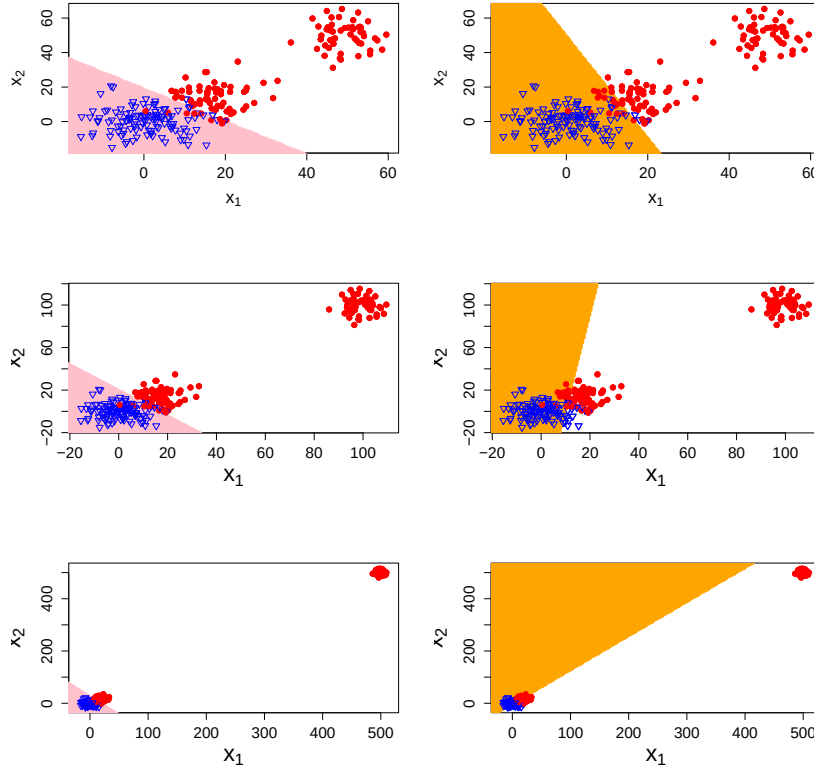


Figure 3: The decision boundaries trained by (left) PUAL and (right) GLLC on the synthetic data with $\text{mean}_{p2} = 50, 100, 500$. Pink area: the negative region of PUAL; orange area: the negative region of GLLC; red dots: unlabelled-positive instances; blue triangles: negative instances; the instances in the plots are from the test sets.

305 4.2. Experiments on Real-World Data

306 In this section, we further assess the classification performance of PUAL on real-
 307 world datasets. We adopt the case-control setting, where ground-truth classes are bi-
 308 narised to construct PU data. The dataset is split into training and test sets, and the
 309 model is trained on the training set using Algorithm 2. Hyperparameters are tuned via
 310 cross-validation, using the F1-score as the evaluation metric.

311 4.2.1. Real-World Datasets

312 Firstly, 16 datasets from the UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/index.php>) were selected to assess the performance of PUAL and
 313 verify our motivation: Accelerometer (**Acc**), **Ecoli**, Pen-Based Recognition of Hand-
 314 written Digits (**Pen**), Online Retail (**OR1**), Online Retail II (**OR2**), Parking Birm-
 315 ingham (**PB**), **wifi**, Sepsis survival minimal clinical records (**SSMCR**), Avila, Raisin
 316 Dataset (**RD**), Occupancy Detection (**OD**), User Knowledge Modelling Data Set (**UMD**),
 317 **Seeds**, Energy efficiency Data Set (**ENB**), Heart Disease (**HD**) and Liver Disorders
 318 (**LD**). The details of these real-world datasets are summarised in Table 2.

Table 2: Summary of the real-world datasets.

Dataset	Positive instances	Negative instances	#Features
Acc	100 red	100 blue	4
Ecoli	116 im & 52 pp	143 cp & 25 om	6
Pen	200 one & 200 eight	400 four	16
OR1	301 UK	301 Germany	4
OR2	500 UK	500 Germany	4
SSMCR	391 alive	109 dead	3
PB	500 Bull Ring	500 BHMBCCMKT01	3
OD	100 occupied	300 not occupied	5
UMD	83 Low	63 high	5
Seeds	70 Kama	70 Rosa	7
ENB	144 TypeII	144 Type III	7
wifi	100 Location 2& 100 Location 4	499 Location 1 & 100 Location 3	7
Avila	300 E	900 A	10
RD	450 Kecimen	450 Besni	7
LD	144 class 1	200 class 2	6
HD	150 absence	119 presence	13

320 4.2.2. Compared Methods

321 GLLC, uPU, nnPU, Robust-PU [40] and T-HOneCls [41] were also trained on the
322 16 real datasets as the compared methods with PUAL. GLLC serves as the baseline
323 of PUAL. uPU and nnPU are two reliable consistent PU learning methods. Robust-
324 PU [40] and T-HOneCls [41] are two recent works of two-step approach and one-step
325 approach for PU classification, respectively.

326 4.2.3. Training-Test Split for the Real PU Datasets

327 Different from the steps in Section 4.1.2, PU training and test sets for the 16 real
328 datasets are constructed under the case-control scenario since we would like to see the
329 performance of PUAL on the datasets with various labelling mechanism. Furthermore,
330 under the case-control scenario, it is possible to do fair comparison between GLLC and
331 the two convincing methods for PU classification, i.e., uPU [17] and nnPU [18], since
332 they were proposed under the case-control scenario. The steps to perform training-test
333 split for the 16 PU real datasets are summarised as follows:

- 334 1. To obtain the binary positive-negative (PN) datasets from the original multi-class
335 dataset, certain classes in each of the original real datasets were treated as pos-
336 itive while some other classes were treated as negative with the rest of classes
337 abandoned.
- 338 2. To construct the labelled-positive set and unlabelled-set, γ' of the positive in-
339 stances in each binary PN dataset obtained in Step 1 were randomly selected to
340 form the labelled-positive set, and the rest of the positive instances were mixed
341 into the unlabelled set, contributing to the unlabelled set of the PU dataset.
- 342 3. To generate the training set and the test set from the constructed PU datasets,
343 the labelled-positive set and 70% of the instances in the unlabelled set were se-
344 lected as the training set while the test set was constituted by the rest 30% of
345 the instances in the unlabelled set; this corresponds to the setting of case-control
346 scenario since the unlabelled training set and the test set can be regarded to be
347 sampled from the same population.

348 During preliminary experiments, we found that the label frequency γ needs to be
349 greater than 20% for the hyper-parameter tuning strategy introduced in Section 4.2.4

to achieve adequate results. Therefore, the value of γ' is set to $\frac{7}{17}, \frac{7}{37}$ so that we have the label frequency $\gamma = \gamma'/(0.3\gamma' + 0.7) = 0.5, 0.25$, which is the fraction of positive instances that are labelled, in the corresponding constructed PU training sets, respectively. Then Step 2 and Step 3 were repeated for 10 times on each of the 16 binary PN datasets and obtained 10 pairs of PU training and test sets for each of the 16 binary PN datasets with a certain value of γ' .

4.2.4. Model Setting

In the case-control scenario, we also use the PUF-score in Equation (31) for hyper-parameter selection. The numerator ‘recall’ can still be estimated by computing $\frac{1}{n_p} \sum_{\mathbf{x}_i \in X_{[p]}} \mathbb{I}(\text{sgn}(f(\mathbf{x}_i)) = 1)$, while the denominator $P[\text{sgn}(f(\mathbf{x})) = 1]$ needs to be estimated by computing $\frac{1}{n_u} \sum_{\mathbf{x}_i \in X_{[u]}} \mathbb{I}(\text{sgn}(f(\mathbf{x}_i)) = 1)$ at the case-control scenario. Therefore, by fixing C_p to 1 and the number K of the nearest neighbors to 5, C_u , λ , and σ in the objective functions of PUAL and GLLC were firstly tuned by 4-fold CV, which reached the highest average PUF-score.

Furthermore, considering the higher complexity of the real datasets compared with the synthetic datasets in Section 4.1, we modified our strategy for hyper-parameter selection, enabling efficient selection of hyper-parameters across a broader range. More specifically, λ , σ were tuned from the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4\}$ and C_u was selected to from the set $\{0.5, 0.3, 0.1, 0.05, 0.01\}$ based on the setting in [24]. Then λ , σ and C_u were continually tuned following the greedy algorithm based on the average PUF-score on the validation sets as follows:

1. Set λ , σ and C_u to the best combination from the grid search.
2. Sequentially update one of hyper-parameters λ , σ and C_u by increasing/decreasing 10% of its current value with the rest of the hyper-parameters unchanged. The optimal scenario on 4-fold CV is set to be the final update of this step.
3. Repeat Step 2 until there is no better scenario appearing.

In addition, the hyper-parameters of uPU and nnPU were fixed as the recommended setting in the open source provided by [18] at <https://github.com/kiryor/nnPUlearning>. The hyper-parameters of Robust-PU were as recommended by [40] at <https://github.com>.

com/woriazzc/robust-pu. The hyper-parameters of T-HOneCls were as recommended by [41] at <https://github.com/Hengwei-Zhao96/T-HOneCls>. Radial Basis Function (RBF) kernel was applied to both PUAL and GLLC. More specifically, we calculated $\exp(-\|x_i - x_j\|^2 / 2\lambda^2)$ as the (i, j) element of $\Phi'(\phi(X_{[pu]}), \phi(X_{[pu]}))$.

4.2.5. Results

The F1-scores of PUAL, GLLC, uPU, nnPU, robust-PU and T-HOneCls on the 16 real datasets are summarised in Table 3. According to the t -tests with p -value lower than 0.05 between the pairwise results of these four methods, there are totally 8 cases where PUAL is the uniquely optimal choice among the six methods. To delve deeper into the results, we compare PUAL with its competitors in detail below.

Table 3: The average F1-score (%) of the classifiers. For each of the 16 original datasets, the average F1-scores and standard deviations in the two rows were obtained under label frequencies $\gamma = 0.5$ (top) and 0.25 (bottom), respectively. In each row, the result highlighted in blue indicates the statistically optimal choice among the six methods according to the t -tests with p -value lower than 0.05.

Dataset	PUAL	GLLC	uPU	nnPU	Robust-PU	T-HOneCls
HD	82.7 ± 2.4	82.0 ± 5.5	71.4 ± 4.2	74.4 ± 2.2	85.3 ± 5.0	83.7 ± 7.9
	81.9 ± 4.0	84.5 ± 4.1	71.0 ± 4.0	75.1 ± 2.4	80.6 ± 1.3	81.4 ± 5.2
Pen	92.5 ± 8.1	88.9 ± 10.2	77.8 ± 31.0	87.5 ± 15.0	93.7 ± 3.8	91.6 ± 2.9
	91.7 ± 9.0	87.0 ± 11.4	72.6 ± 31.0	84.1 ± 16.9	91.8 ± 2.6	88.2 ± 3.5
LD	44.2 ± 5.7	50.8 ± 6.9	11.9 ± 25.8	31.5 ± 27.8	83.9 ± 3.7	86.0 ± 1.5
	36.9 ± 10.0	40.1 ± 8.9	10.2 ± 22.4	20.1 ± 26.3	81.1 ± 6.1	80.9 ± 1.9
OR1	90.1 ± 2.3	85.6 ± 3.8	16.6 ± 33.3	84.1 ± 6.9	85.8 ± 2.3	82.4 ± 6.4
	83.9 ± 5.8	72.9 ± 5.5	20.9 ± 33.1	72.1 ± 7.0	80.6 ± 4.9	78.4 ± 3.7
RD	82.5 ± 2.2	83.0 ± 2.4	70.6 ± 12.9	71.3 ± 13.6	62.8 ± 0.9	60.7 ± 8.4
	77.6 ± 3.8	82.0 ± 2.8	72.9 ± 14.5	73.1 ± 12.8	62.6 ± 2.4	69.8 ± 8.1
Seeds	92.3 ± 4.9	94.6 ± 2.8	92.4 ± 1.5	97.3 ± 3.7	100.0 ± 0.0	100.0 ± 0.0
	89.1 ± 5.5	91.2 ± 4.5	86.9 ± 3.1	93.1 ± 3.9	100.0 ± 0.0	100.0 ± 0.0
wifi	95.1 ± 2.0	90.4 ± 4.7	91.2 ± 4.3	92.2 ± 3.2	93.9 ± 1.8	90.1 ± 4.3
	96.7 ± 1.8	89.1 ± 4.8	87.7 ± 2.9	89.3 ± 2.6	90.3 ± 2.5	89.7 ± 4.7
Avila	55.8 ± 2.9	59.6 ± 4.1	62.7 ± 8.8	63.8 ± 9.1	79.5 ± 3.8	82.3 ± 2.6
	50.1 ± 4.2	55.0 ± 6.2	61.3 ± 9.2	61.0 ± 9.0	76.3 ± 2.4	80.8 ± 4.2
OD	89.0 ± 8.4	100.0 ± 0.0	80.0 ± 42.2	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
	95.7 ± 6.7	100.0 ± 0.0	80.0 ± 42.2	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
OR2	88.9 ± 1.2	86.5 ± 1.4	76.9 ± 4.9	81.6 ± 4.2	84.2 ± 4.7	83.7 ± 5.8
	85.5 ± 3.4	77.1 ± 5.7	74.4 ± 5.5	77.3 ± 3.8	81.0 ± 2.2	80.5 ± 2.5

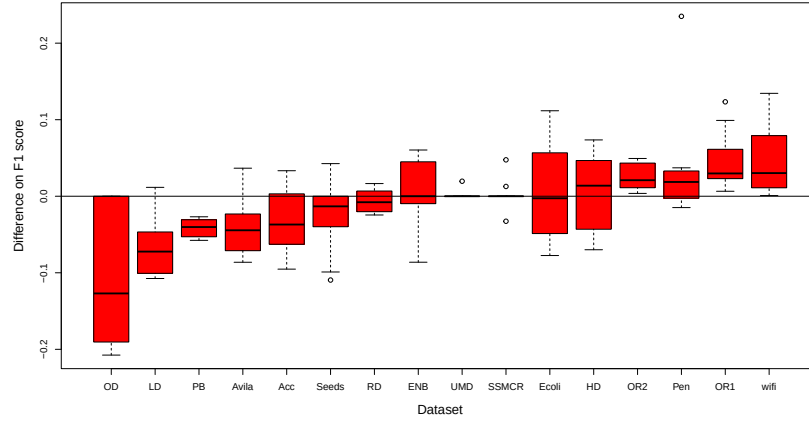
Table 3 – continued from previous page

Dataset	PUAL	GLLC	uPU	nnPU	Robust-PU	T-HOneCls
PB	95.9 $\pm$ 1.1 97.9 $\pm$ 0.7	100.0 $\pm$ 0.0 100.0 $\pm$ 0.0	69.8 $\pm$ 2.6 68.8 $\pm$ 2.6	67.2 $\pm$ 3.2 66.6 $\pm$ 4.1	96.2 $\pm$ 1.7 93.1 $\pm$ 2.5	98.2 $\pm$ 2.7 96.9 $\pm$ 3.2
Acc	65.0 $\pm$ 4.8 66.4 $\pm$ 4.4	68.1 $\pm$ 2.2 64.1 $\pm$ 6.3	20.1 $\pm$ 27.6 22.0 $\pm$ 29.6	20.5 $\pm$ 28.6 23.4 $\pm$ 31.4	72.5 $\pm$ 4.9 69.1 $\pm$ 5.3	76.1 $\pm$ 7.4 71.9 $\pm$ 8.1
Ecoli	90.8 $\pm$ 2.6 88.0 $\pm$ 4.4	90.2 $\pm$ 6.3 88.0 $\pm$ 4.0	84.4 $\pm$ 6.1 84.9 $\pm$ 6.8	85.9 $\pm$ 6.7 86.1 $\pm$ 6.6	86.2 $\pm$ 2.4 83.1 $\pm$ 1.8	88.3 $\pm$ 9.8 85.5 $\pm$ 11.0
SSMCR	87.6 $\pm$ 1.3 87.6 $\pm$ 1.3	87.4 $\pm$ 2.1 87.4 $\pm$ 2.1	85.7 $\pm$ 2.0 85.0 $\pm$ 2.0	87.4 $\pm$ 1.4 86.8 $\pm$ 1.5	62.8 $\pm$ 0.9 62.5 $\pm$ 0.9	60.7 $\pm$ 8.4 69.8 $\pm$ 6.1
ENB	42.8 $\pm$ 4.8 45.8 $\pm$ 7.5	42.7 $\pm$ 4.6 44.2 $\pm$ 6.6	29.6 $\pm$ 22.1 26.1 $\pm$ 30.5	30.2 $\pm$ 23.7 26.9 $\pm$ 31.3	69.3 $\pm$ 5.2 65.1 $\pm$ 3.1	64.8 $\pm$ 3.2 64.0 $\pm$ 7.1
UMD	100.0 $\pm$ 0.0 99.6 $\pm$ 0.9	99.8 $\pm$ 0.6 98.4 $\pm$ 1.8	100.0 $\pm$ 0.0 100.0 $\pm$ 0.0	100.0 $\pm$ 0.0 100.0 $\pm$ 0.0	100.0 $\pm$ 0.0 100.0 $\pm$ 0.0	100.0 $\pm$ 0.0 100.0 $\pm$ 0.0

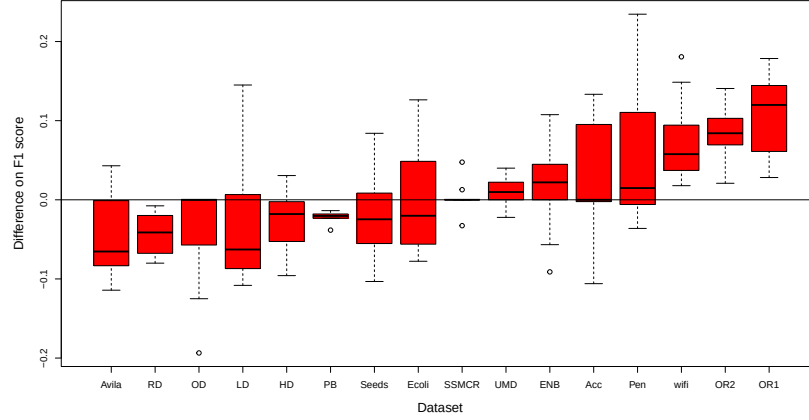
389 *Comparison between PUAL and GLLC.* The results of the difference between the F1-
390 scores of PUAL and GLLC on each pairwise experiments are shown in the boxplots in
391 Fig. 4 with $\gamma = 0.5, 0.25$, respectively. In both figures, the best four datasets for PUAL
392 over GLLC are **wifi**, **OR1**, **OR2**, and **Pen**. Furthermore, with $\gamma = 0.5, 0.25$, three of
393 the worst four datasets for PUAL compared with GLLC are **OD**, **LD**, **Avila**. When
394 $\gamma = 0.5$, the rest one of the worst four datasets is **PB**; when $\gamma = 0.25$, it is **RD**.

395 *Pattern Analysis for the Real Datasets Preferring PUAL to GLLC.* The t-SNE plots of
396 the best four datasets, i.e., **wifi**, **OR1**, **OR2**, and **Pen**, for PUAL compared with GLLC
397 are shown in Fig. 5. According to the four t-SNE plots, the following observations can
398 be made:

- 399 1. The trifurcate pattern of the best four datasets in Fig. 5 is clear that the positive
400 set is constituted by two subsets distributed on both sides of the negative set as
401 discussed in the motivation of PUAL in Section 3.1.
- 402 2. In the t-SNE plot of dataset **OR1**, there are many more instances in the right-
403 hand labelled-positive subset than the instances in the left-hand labelled-positive
404 set. This indicates that the trifurcate pattern does not have to be balanced for
405 PUAL to outperform GLLC.



(a) $\gamma = 0.5$



(b) $\gamma = 0.25$

Figure 4: Boxplots for the difference between F1-scores of PUAL and GLLC on each dataset increasingly ranked by medians; label frequencies $\gamma = 0.5$ (top) and 0.25 (bottom); x-axis: the datasets; y-axis: the difference between PUAL and GLLC in F1-score.

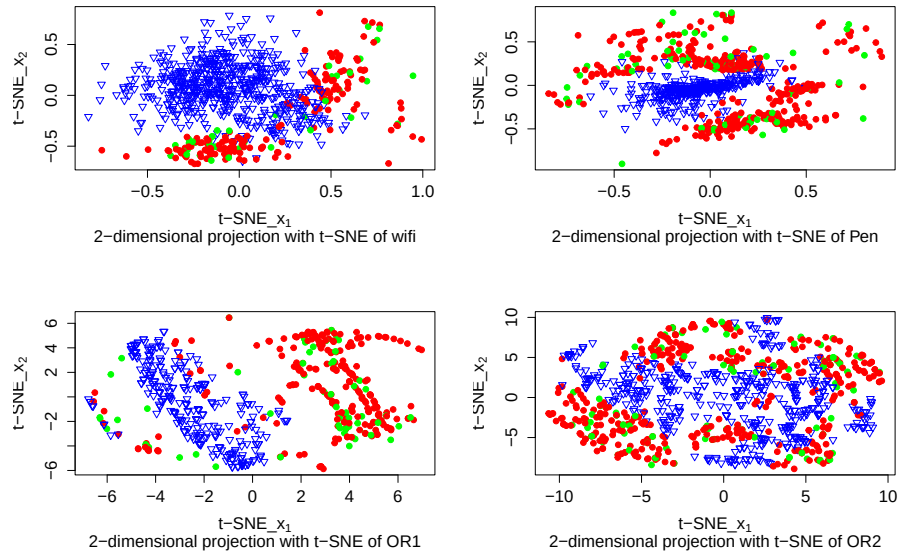


Figure 5: The t-SNE plots of the best four datasets **wifl**, **OR1**, **OR2** and **Pen**; the perplexity for the training of t-SNE on these four datasets was set to 750, 40, 250, 750, respectively; label frequency $\gamma=0.25$; green dots: labelled-positive instances; red dots: unlabelled-positive instances; blue triangles: negative instances.

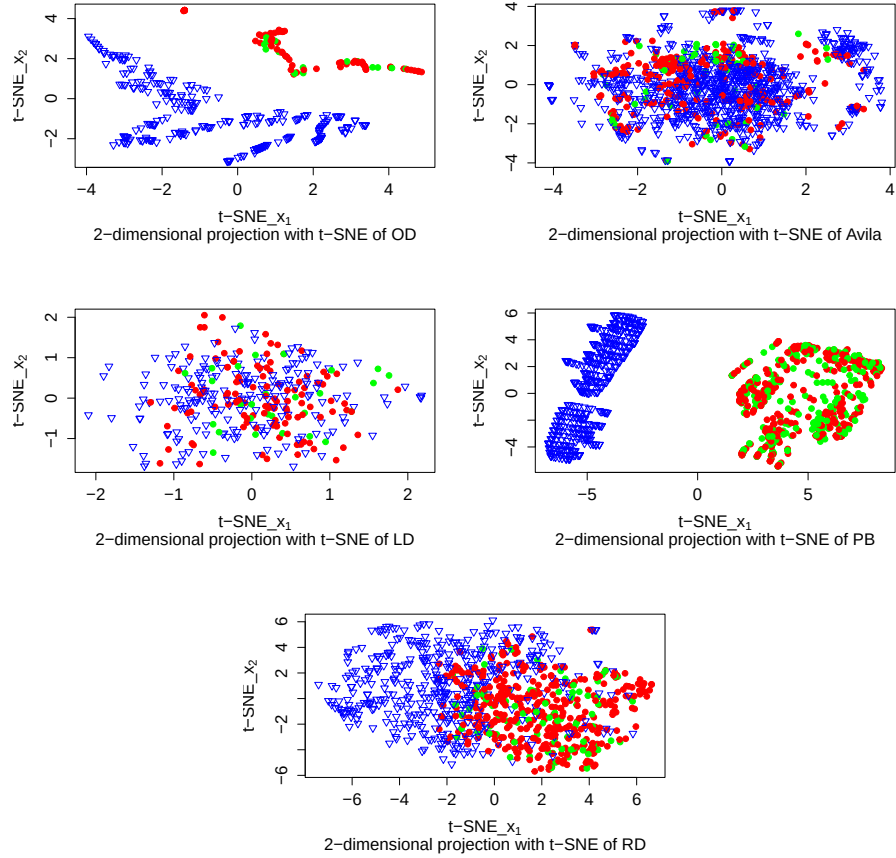


Figure 6: The t-SNE plots of the worst five datasets **OD**, **Avila**, **LD**, **PB**, **RD**; cross entropy loss for training with perplexity = 200, 550, 250, 300, 300; label frequency $\gamma=0.25, 0.25, 0.25, 0.5, 0.25$. The rest of the caption is as in Fig. 5.

406 *Pattern Analysis for the Real Datasets Preferring GLLC to PUAL.* There are overall
 407 five datasets to be the worst four datasets for PUAL compared with GLLC under two
 408 cases of label frequency, i.e., $\gamma = 0.5, 0.25$. The t-SNE plots for them are illustrated in
 409 Fig. 6 and their patterns can be summarised as follows:

- 410 1. According to the t-SNE plots of the two datasets **LD** and **Avila**, the positive
 411 set and the negative set are mixed together, making the dataset challenging to be
 412 separated. In this case, the labelled-positive instances selected as support vectors
 413 by the hinge loss of PUAL are not sufficient to adequately represent the pattern
 414 of the positive set, while the squared loss of GLLC on the labelled-positive set,
 415 which selects all positive instances as the support vectors, can somewhat alleviate
 416 this issue. As a result, GLLC on these two datasets outperforms PUAL.
- 417 2. The t-SNE plots of the three datasets **OD**, **PB**, and **RD** represent the typical
 418 two-class patterns. In this type of datasets, the problem of GLLC mentioned in
 419 Section 3.1 does not exist. Therefore, the optimal combination(s) of the hyper-
 420 parameters for GLLC to outperform PUAL was (were) found in at least one case
 421 of label frequency γ .

422 *Comparison between PUAL, uPU, nnPU, robust-PU and T-HOneCls.* Based on the
 423 results in Table 3, we can obtain the following findings. First, there are in total 13/32
 424 cases where PUAL outperforms the state-of-the-art methods. Secondly, uPU and nnPU
 425 sometimes have much larger standard deviations than PUAL, since their algorithms
 426 based on Adam for the optimization of their non-convex objective functions cannot
 427 always converge to the optimal solution, although nnPU alleviates this issue to some
 428 extent.

429 *Ablation Study of PUAL.* To analyze the impact of different terms in the loss function,
 430 we conduct an ablation study by comparing three variants: PUAL with an RBF kernel
 431 (PUAL), PUAL without the kernel trick (PUAL-linear), and PUAL with an RBF kernel
 432 but without the local constraint term involving **R** (PUAL-global). The F1-scores on 16
 433 UCI datasets are summarised in Table 4. The results show a substantial decrease in F1-
 434 scores without the RBF kernel, suggesting that the classification boundaries in these

real datasets are nonlinear. Additionally, removing the local constraint also leads to lower F1-scores, although the effect is less significant.

Table 4: The average F1-score (%) of the ablation study. The rest of the caption is as in Table 3.

Dataset	PUAL	PUAL-linear	PUAL-global	Dataset	PUAL	PUAL-linear	PUAL-global
HD	82.7 ± 2.4 81.9 ± 4.0	59.2 ± 4.2 57 ± 5.1	76.8 ± 8.3 72.4 ± 9.1	OD	89.0 ± 8.4 95.7 ± 6.7	53.6 ± 14.9 48.2 ± 11.5	83 ± 9.6 81.7 ± 8.3
Pen	92.5 ± 8.1 91.7 ± 9.0	62.6 ± 1.7 58.1 ± 3.0	85.5 ± 7.4 82.1 ± 11.6	OR2	88.9 ± 1.2 85.5 ± 3.4	61.1 ± 3.2 57.3 ± 7.0	82.1 ± 2.7 79.9 ± 5.5
LD	44.2 ± 5.7 36.9 ± 10.0	14.8 ± 6.9 16.3 ± 6.5	41.7 ± 2.1 31.1 ± 2.9	PB	95.9 ± 1.1 97.9 ± 0.7	51.9 ± 12.6 53.4 ± 12.1	91.4 ± 7.9 92.4 ± 5.1
OR1	90.1 ± 2.3 83.9 ± 5.8	59.5 ± 4.8 53.7 ± 5.2	84.9 ± 5.8 82.6 ± 4.8	Ecoli	90.8 ± 2.6 88.0 ± 4.4	65.3 ± 7.2 58.8 ± 10.6	87.5 ± 3.3 83.8 ± 5.8
RD	82.5 ± 2.2 77.6 ± 3.8	55.9 ± 2.1 53.7 ± 2.6	75.9 ± 7.0 74.4 ± 8.2	Acc	65.0 ± 4.8 66.4 ± 4.4	42.5 ± 6.3 40.5 ± 5.8	61 ± 3.9 57.1 ± 3.7
Seeds	92.3 ± 4.9 89.1 ± 5.5	79.1 ± 7.4 70.5 ± 6.6	86.8 ± 1.6 85.6 ± 1.9	SSMCR	87.6 ± 1.3 87.6 ± 1.3	48.0 ± 6.3 42.3 ± 6.1	81.9 ± 6.4 81.9 ± 6.4
wifi	95.1 ± 2.0 96.7 ± 1.8	61.0 ± 3.9 60.3 ± 5.7	90.3 ± 3.2 88.3 ± 5.2	ENB	42.8 ± 4.8 45.8 ± 7.5	36.2 ± 2.5 33.8 ± 3.9	40.3 ± 5.8 41.7 ± 3.9
Avila	55.8 ± 2.9 50.1 ± 4.2	37.2 ± 6.4 32.8 ± 10.9	46.8 ± 8.4 42.5 ± 12.8	UMD	100.0 ± 0.0 99.6 ± 0.9	83.1 ± 6.2 81.4 ± 7.4	89.6 ± 8.0 87.9 ± 12.4

5. Conclusion

In this paper, we propose PUAL for better classification on trifurcate PU datasets, where the positive set is constituted by two subsets distributed on both sides of the negative set. The key novelty of PUAL is an asymmetric structure composed of hinge loss and squared loss for training the PU classifier. Experimental results demonstrate the superiority of PUAL in performing PU classification on trifurcate data. As an SVM-based method, PUAL is still negatively affected by the irrelevant features in the datasets, partly due to the L2 regularization term of model parameters, which cannot compress the coefficients of the irrelevant features to be exactly zero thoroughly [42]. To address this issue, a future work is to explore the L1 regularization term of the model parameters for PUAL.

References

- [1] Y. Ren, D. Ji, H. Zhang, Positive unlabeled learning for deceptive reviews detection, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 488–498.
- [2] H. Li, Z. Chen, B. Liu, X. Wei, J. Shao, Spotting fake reviews via collective positive-unlabeled learning, in: 2014 IEEE International Conference on Data Mining, IEEE, 2014, pp. 899–904.
- [3] W. Li, Q. Guo, C. Elkan, A positive and unlabeled learning algorithm for one-class classification of remote-sensing data, IEEE Transactions on Geoscience and Remote Sensing 49 (2) (2010) 717–725.
- [4] S. Dai, X. Li, Y. Zhou, X. Ye, T. Liu, GradPU: positive-unlabeled learning via gradient penalty and positive upweighting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 7296–7303.
- [5] W. Lan, J. Wang, M. Li, J. Liu, Y. Li, F.-X. Wu, Y. Pan, Predicting drug–target interaction using positive-unlabeled learning, Neurocomputing 206 (2016) 50–57.
- [6] H. Yu, J. Han, K.-C. Chang, PEBL: Web page classification without negative examples, IEEE Transactions on Knowledge and Data Engineering 16 (1) (2004) 70–81.
- [7] F. He, T. Liu, G. I. Webb, D. Tao, Instance-dependent PU learning by Bayesian optimal relabeling, arXiv preprint arXiv:1808.02180 (2018).
- [8] L. Liu, T. Peng, Clustering-based method for positive and unlabeled text categorization enhanced by improved TFIDF, Journal of Information Science & Engineering 30 (5) (2014).
- [9] B. Liu, Q. Liu, Y. Xiao, A new method for positive and unlabeled learning with privileged information, Applied Intelligence 52 (3) (2022) 2465–2479.

- [10] S. Chaudhari, S. Shevade, Learning from positive and unlabelled examples using maximum margin clustering, in: International Conference on Neural Information Processing, Springer, 2012, pp. 465–473.
- [11] Y. He, X. Li, M. Zhang, P. Fournier-Viger, J. Z. Huang, S. Salloum, A novel observation points-based positive-unlabeled learning algorithm, CAAI Transactions on Intelligence Technology 8 (4) (2023) 1425–1443.
- [12] C. Xu, C. Liu, S. Yang, Y. Wang, S. Zhang, L. Jia, Y. Fu, Split-PU: Hardness-aware training strategy for positive-unlabeled learning, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 2719–2729.
- [13] E. Dorigatti, J. Goschenhofer, B. Schubert, M. Rezaei, B. Bischl, Positive-unlabeled learning with uncertainty-aware pseudo-label selection, arXiv preprint arXiv:2201.13192 (2022).
- [14] Q. Liang, M. Zhu, Y. Wang, X. Wang, W. Zhao, M. Yang, H. Wei, B. Han, X. Zheng, Positive distribution pollution: rethinking positive unlabeled learning from a unified perspective, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 8737–8745.
- [15] S. Kong, W. Shen, Y. Zheng, A. Zhang, J. Pu, J. Wang, False positive rate control for positive unlabeled learning, Neurocomputing 367 (2019) 13–19.
- [16] B. Žunkovič, Positive unlabeled learning with tensor networks, Neurocomputing 552 (2023) 126556.
- [17] M. C. Du Plessis, G. Niu, M. Sugiyama, Analysis of learning from positive and unlabeled data, Advances in neural information processing systems 27 (2014) 703–711.
- [18] R. Kiryo, G. Niu, M. C. d. Plessis, M. Sugiyama, Positive-unlabeled learning with non-negative risk estimator, arXiv preprint arXiv:1703.00593 (2017).
- [19] G. Su, W. Chen, M. Xu, Positive-unlabeled learning from imbalanced data, in: International Joint Conference on Artificial Intelligence, 2021, pp. 2995–3001.

- 501 [20] B. Liu, Y. Dai, X. Li, W. S. Lee, P. S. Yu, Building text classifiers using posi-
502 tive and unlabeled examples, in: Third IEEE International Conference on Data
503 Mining, IEEE, 2003, pp. 179–186.
- 504 [21] C. Cortes, V. Vapnik, Support vector machine, Machine learning 20 (3) (1995)
505 273–297.
- 506 [22] Z. Liu, W. Shi, D. Li, Q. Qin, Partially supervised classification: based on
507 weighted unlabeled samples support vector machine, in: Data Warehousing and
508 Mining: Concepts, Methodologies, Tools, and Applications, IGI Global, 2008,
509 pp. 1216–1230.
- 510 [23] T. Ke, H. Lv, M. Sun, L. Zhang, A biased least squares support vector machine
511 based on mahalanobis distance for PU learning, Physica A: Statistical Mechanics
512 and its Applications 509 (2018) 422–438.
- 513 [24] T. Ke, L. Jing, H. Lv, L. Zhang, Y. Hu, Global and local learning from positive
514 and unlabeled examples, Applied Intelligence 48 (8) (2018) 2373–2392.
- 515 [25] C. Gong, T. Liu, J. Yang, D. Tao, Large-margin label-calibrated support vector
516 machines for positive and unlabeled learning, IEEE Transactions on Neural Net-
517 works and Learning Systems 30 (11) (2019) 3471–3483.
- 518 [26] L. van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine
519 Learning Research 9 (2008) 2579–2605.
- 520 [27] R. Bhatt, Wireless indoor localization, UCI Machine Learning Repository, DOI:
521 <https://doi.org/10.24432/C51880> (2017).
- 522 [28] C. Scott, G. Blanchard, Novelty detection: Unlabeled data definitely help, in:
523 Artificial Intelligence and Statistics, PMLR, 2009, pp. 464–471.
- 524 [29] O. Chapelle, B. Scholkopf, A. Zien (Eds.), Semi-Supervised Learning, MIT
525 Press, 2006.

- [30] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, *Computers & Mathematics With Applications* 2 (1) (1976) 17–40.
- [31] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends® in Machine learning* 3 (1) (2011) 1–122.
- [32] G.-B. Ye, X. Xie, Split Bregman method for large scale fused lasso, *Computational Statistics & Data Analysis* 55 (4) (2011) 1552–1569.
- [33] S. Jain, M. White, M. W. Trosset, P. Radivojac, Nonparametric semi-supervised learning of class proportions, *arXiv preprint arXiv:1601.01944* (2016).
- [34] M. Christoffel, G. Niu, M. Sugiyama, Class-prior estimation for learning from positive and unlabeled data, in: *Asian Conference on Machine Learning*, PMLR, 2016, pp. 221–236.
- [35] J. Bekker, J. Davis, Estimating the class prior in positive and unlabeled data through decision tree induction, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, 2018, pp. 2712–2719.
- [36] M. G. Genton, Classes of kernels for machine learning: a statistics perspective, *Journal of Machine Learning Research* 2 (2001) 299–312.
- [37] J. Bekker, J. Davis, Learning from positive and unlabeled data: a survey, *Machine Learning* 109 (4) (2020) 719–760.
- [38] W. S. Lee, B. Liu, Learning with positive and unlabeled examples using weighted logistic regression, in: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, 2003, pp. 448–455.
- [39] Y. Liu, J. Zhao, Y. Xu, Robust and unbiased positive and unlabeled learning, *Knowledge-Based Systems* 277 (2023) 110819.

- 551 [40] Z. Zhu, L. Wang, P. Zhao, C. Du, W. Zhang, H. Dong, B. Qiao, Q. Lin, S. Ra-
552 jmoohan, D. Zhang, Robust positive-unlabeled learning via noise negative sam-
553 ple self-correction, in: Proceedings of the 29th ACM SIGKDD Conference on
554 Knowledge Discovery and Data Mining, 2023, pp. 3663–3673.
- 555 [41] H. Zhao, X. Wang, J. Li, Y. Zhong, Class prior-free positive-unlabeled learning
556 with taylor variational loss for hyperspectral remote sensing imagery, in: Pro-
557 ceedings of the IEEE/CVF International Conference on Computer Vision, 2023,
558 pp. 16827–16836.
- 559 [42] H. T. Nguyen, K. Franke, S. Petrović, On general definition of l_1 -norm support
560 vector machines for feature selection, International Journal of Machine Learning
561 and Computing 1 (3) (2011) 279.