Applied Soft Computing 142 (2023) 110298

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Privileged information learning with weak labels

Yanshan Xiao [a,*], Zexin Ye [a], Liang Zhao [a], Xiangjun Kong [a], Bo Liu [b], Kemal Polat [c], Adi Alhudhaif [d]

[a] School of Computers, Guangdong University of Technology, China
[b] School of Automation, Guangdong University of Technology, China
[c] Department of Electrical and Electronics Engineering, Bolu Abant Izzet Baysal University, Turkey
[d] Department of Computer Science, College of Computer Engineering and Sciences in Al-kharj, Prince Sattam bin Abdulaziz University, P.O. Box 151, Al-Kharj 11942, Saudi Arabia

ARTICLE INFO

Article history:
Received 24 October 2022
Received in revised form 30 March 2023
Accepted 2 April 2023
Available online 18 April 2023

Keywords:
Privileged information learning
Weak label
Support vector machine
Heuristic framework
Labeler re-weighting

ABSTRACT

Privileged information learning is proposed to construct the classifier by incorporating privileged knowledge. At present, most of the privileged information learning methods assume that the instance is accurately labeled. However, in real-world applications, an instance may be weakly labeled. In this paper, we propose a novel privileged information learning method with weak labels (PLWB). The hypothesis of our work is that an instance may be annotated by a number of labelers and different labelers may give different labels to this instance due to distinct professional knowledge and subjective factors. It leads to ambiguous labels of instances, namely weak labels. To solve this problem, our methodology is to give each labeler a weight and incorporate these weights into a privileged information learning model. Our technique is to employ a heuristic framework to optimize the labeler weights and the privileged information learning model jointly. The existing privileged information learning methods do not consider the weak label problem, and assign an equal or random weight to each labeler. Our work is different from these methods. The novelty and theoretical contribution is that this is the first work to deal with the weak label problem in privileged information learning. The merit is that we assign an unknown weight to each labeler and solve the optimal values of these weights in the optimization process, such that the performance of the learning model can be improved with the optimal labeler weights. In the experiments, the tool that we use is MATLAB, in which we implement our algorithm. The experimental datasets include one handwritten categorization dataset, two image classification datasets (i.e., Animals-with-Attributes dataset and Caltech-101 dataset), and one disease diagnosis dataset (i.e., Alzheimer's Disease Neuroimaging Initiative dataset), in which the number of instances used is 2000, 6180, 8677 and 202, respectively. The obtained results are that: (1) by optimizing the labeler weights, the proposed PLWB method obtains explicitly higher classification accuracy than the existing privileged information learning methods; (2) PLWB has relatively higher training time since it needs to solve the labeler weights in the optimization process.

1. Introduction

1.1. Background

Traditional supervised learning uses only the training data to construct a model. However, in real-world applications, we may collect some extra information that is available only in the training process, but not available in the testing process. Consider an example of facial expression detection in video surveillance. In the laboratory environment, we can obtain not only the low-resolution facial expression images but also high-resolution facial expression images. The utilization of high-resolution facial expression images can effectively refine the capability of classifiers. However, in the practical scenarios of video surveillance, we usually get only the low-resolution images that are collected by the surveillance cameras. The high-resolution images are available only in the training process, but not in the testing process. Although the high-resolution images are not obtainable during the testing process, they can be incorporated into the training process to reinforce the classifier from the low-resolution images. Here, the additional information which is available only in the training process, but not available in the testing process is called privileged information [1]. Privileged information learning

* Corresponding author.
E-mail addresses: xiaoyanshan@gdut.edu.cn (Y. Xiao), 2112005039@mail2.gdut.edu.cn (Z. Ye), 1112005007@mail2.gdut.edu.cn (L. Zhao), XiangjunK@gdut.edu.cn (X. Kong), csbliu@gdut.edu.cn (B. Liu), kpolat@ibu.edu.tr (K. Polat), a.alhudhaif@psau.edu.sa (A. Alhudhaif).

https://doi.org/10.1016/j.asoc.2023.110298
1568-4946/

aims to incorporate privileged information into improving the classification performance of classifiers.

Privileged information learning is an important research field in machine learning and computer vision. Considerable works have been done on privileged information learning. For example, Vapnik and Vashist [1] proposed a learning paradigm, called privileged information support vector machine (SVM+). SVM+ boosts the classification performance of SVM by utilizing privileged data. [2–4] introduce privileged information into the multi-label learning problem. Since training an SVM+ classifier is computationally expensive, several studies [5–8] are put forward to speed up the training efficiency of SVM+.

1.2. Motivations

Although privileged information learning has made much progress, most of the existing works [9–12] assume that the instances in privileged information learning are well labeled. Nevertheless, in practice, the instance labels may be ambiguous, which is considered as a weak label learning problem. On the one hand, annotating data is usually a time-consuming labor process. If the amount of data is large, correctly labeling all the data may be challenging for the labelers. On the other hand, an instance may be labeled by multiple labelers. Considering that different labelers may have distinct professional knowledge and subjective factors, they may not give exactly the same label to the instance. Most of the existing privileged information algorithms [13–16] assume that the instance is accurately labeled, and privileged information learning with weak labels has not been considered.

1.3. Novelty

In this paper, we solve the problem of weak labels in privileged information learning where the training data and privileged data are associated with ambiguous labels, rather than accurate labels. To tackle this problem, we put forward a new privileged information learning method with weak labels (PLWB). In the proposed method, considering that each instance is annotated by multiple labelers, we assign each labeler a weight and use the weighted labels to represent the instance label. These labeler weights are then incorporated into the SVM+ model. A heuristic framework is put forward to learn the privileged information learning model and update the labeler weights alternately. The main contributions of our work are shown as follows.

- The privileged information learning problem with weak labels is introduced. To the best of our knowledge, this is the first attempt to deal with the weak labels in privileged information learning problems.
- Rather than randomly or equally assigning the weight to each labeler, we update the weight of each labeler in our heuristic framework. These labeler weights are incorporated in the training phase to boost the performance of classifiers.
- We conduct experiments on the handwritten categorization, image classification and disease diagnosis datasets. The numerical results have demonstrated that compared to the existing privileged information learning methods, PLWB achieves improved classification performance.

The rest of this paper is organized as follows. The existing work on privileged information learning is discussed in Section 2. The details of the proposed PLWB method are presented in Section 3. Experiments are conducted in Section 4. The conclusion and future work of this paper are given in Section 5.

2. Related work

2.1. Learning privileged information

(1) *State-of-the-art techniques*

Privileged information provides guidance for the learning of classifiers. It is obtainable only in the training process, and not in the testing process. Vapnik and Vashist [1] showed that privileged information can help to improve the performance of a classifier. Then, they propose a privileged information learning algorithm, called SVM+. The SVM+ model is widely used in practical applications. Sabeti et al. [17] leveraged SVM+ and uncertainty labels to detect acute respiratory distress syndrome. Sharmanska et al. [18] put forward a ranking SVM+, which can transfer the similarity of original data to privileged data by using the privileged information from image tags or bounding boxes. Due to the expensive computation of the L2-norm SVM+ model, Niu and Wu [19] proposed a novel SVM+ model, which replaces the L2-norm with an extended L1-norm. The optimization problem is formulated as a linear programming (LP) problem, and the computational cost is less than L2-norm SVM+. Sarafianos et al. [20] extended SVM+ to domain adaptation and proposed the adaptive SVM+ in which the privileged information of the source domain is transferred to the target domain. Lapin et al. [21] tried to find the connection between the SVM+ solution and weighted SVM. They have found that the information of privileged features can be represented by the weights of instances. It can help us to clearly understand the limitations of SVM+, and the link between the SVM+ algorithm and weighted SVM.

(2) *Implications from the literature*

Despite much progress on privileged information learning, the existing privileged information learning algorithms are proposed based on the assumption that all the training instances and privileged instances are accurately labeled. That is to say, different labelers give the same label to one instance and there is no ambiguity in the instance labels. They build up the classifier directly using these labels and the obtained classifier is then used to predict new instances.

(3) *Research gaps identified*

However, in real-world applications, an instance may be annotated by a number of labelers. Due to distinct professional knowledge and subjective factors, different labelers may give different labels to the instance. It leads to ambiguous labels for instances, namely weak labels. The existing privileged information learning works do not take weak labels into account. In this paper, we put forward a new privileged information learning method with weak labels. It can handle the training data and privileged data which are in weak labels.

2.2. Learning with weak labels

(1) *State-of-the-art techniques*

In real-world applications, we may collect a large number of instances to train the classifier. Correctly labeling all the instances is time-consuming for the labelers. Thus, it may be difficult for the labelers to give an accurate label to each instance, which leads to the ambiguity of instance labels. Sultani and Shah et al. [22] annotated multiple instances of the same action in a weakly labeled video. Liu et al. [23] incorporated the incomplete multi-view data into weak label learning and developed a novel model which can simultaneously learn from missing labels and incomplete views. Choi et al. [24] proposed a framework which uses a local detector and global classifier to detect the weakly labeled acoustic event. To tackle the multi-instance multi-label learning problem with weak labels, Yang et al. [25] assumed that if the labels are highly correlated, they should have common instances. Then, the margin among the class means of bags is maximized.

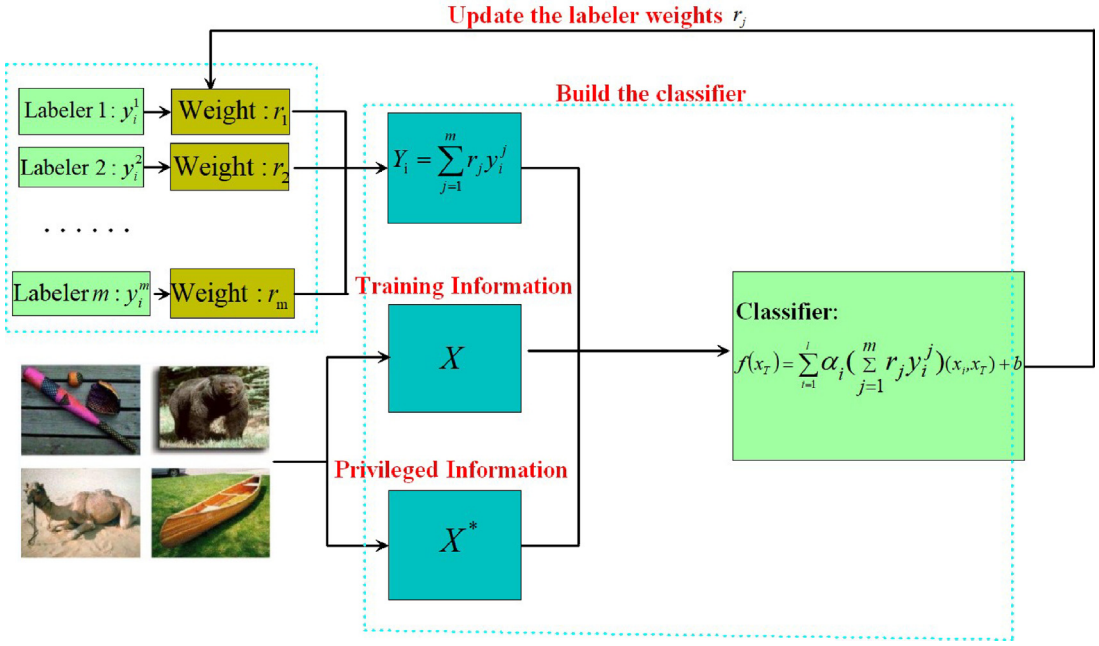

Fig. 1. The schematic diagram of PLWB.

(2) *Implications from the literature*

Most of the existing weak label learning methods [25–28] assume that only the training instances are available during the training stage. They learn the classifier on the training instances and predict the new instances using the obtained classifier. However, in practice, we may get some auxiliary classification information (namely privileged instances), in addition to the training instances. These privileged instances contain some kind of classification information and can be used to boost the classifier learnt from the training instances.

(3) *Research gaps identified*

The existing weak label learning methods use only the training instances to build the classifier, and the privileged instances have not been taken into account. Introducing the privileged instances into the training process can help to refine the classifier and boost the classification accuracy. It motivates the work in this paper. Therefore, we put forward a novel privileged information learning method with weak labels.

3. SVM+ with weak labels

3.1. SVM

Support vector machine (SVM) is a traditional binary class classification method. Consider a binary class dataset $D = \{(x_i, y_i)|i = 1, \ldots, l\}$, where x_i is the ith instance and $y_i \in \{1, -1\}$ is the label of x_i. SVM aims to learn a hyper-plane which separates the positive instances and negative instances, and the margin of the two classes is then maximized. Let $f(x) = (w, x) + b$ be the hyper-plane, where w is the norm vector and b is the bias. To learn this hyper-plane, the objective function of SVM is as follows:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i(w, x_i) + b \geq 1 - \xi_i, \\ & \xi_i \geq 0, \end{aligned} \tag{1}$$

where C denotes the penalty parameter; ξ_i is the slack variable. From the learning problem (1) of SVM, we have the following observations. On the one hand, SVM utilizes only the training data x_i to build up the classifier, and the privileged data cannot be utilized to boost the performance of classifiers. On the other hand, SVM assumes that all the instances are accurately labeled and the ground-truth labels y_i are known. It does not take the weak label problem into account.

3.2. PLWB formulation

Different from SVM, the proposed PLWB method considers the weak label problem and incorporates the privileged information into improving the classifier. Let the training dataset be $D = \{(x_i, x_i^*, Y_i)|i = 1, \ldots, l\}$, where l denotes the number of instances. x_i is the input training instance and x_i^* is the privileged information of x_i. Y_i is the label of x_i. In the training process, both the training information and privileged information are available to construct the classifier. In the testing process, only the testing instance is available and the privileged information is unavailable. If it has $Y_i \geqslant 0$, the instance is labeled as positive. If it has $Y_i < 0$, the instance is labeled as negative. In practice, the instance x_i may be annotated by multiple labelers and Y_i is determined by labels $\{y_i^1, y_i^2, \ldots, y_i^m\}$, where m is the number of labelers; $y_i^j \in \{-1, 1\}$, $j = 1, 2, \ldots, m$, is the label given by the jth labeler. The goal of our method is to learn a privileged information learning model on the data with weak labels.

Fig. 1 shows the schematic diagram of PLWB. In privileged information learning, the training set contains the training instances X and privileged instances X^*. Each training instance x_i is annotated by m labelers. The labels given by the m labelers are y_i^1, y_i^2, $\ldots$, y_i^m, respectively. Due to distinct professional knowledge and subjective factors, different labelers may give different labels. That is to say, the m labels y_i^1, y_i^2, $\ldots$, y_i^m may not be the same, which leads to the ambiguity of instance labels, namely weak labels. To deal with the weak label problem, we assign each labeler a weight r_j, and represent the label of instance x_i as $Y_i = \sum_{j=1}^{m} r_j y_i^j$. To optimize the labeler weight r_j, a heuristic framework is adopted. In this framework, we first initialize the labeler weight r_j and train the classifier $f(x_T)$. Then, we fix the classifier $f(x_T)$ and update the labeler weight r_j. These two steps are conducted

alternately until the algorithm stops. When the algorithm stops, the labeler weight r_j and the classifier $f(x_T)$ are outputted to predict new instances.

Our method is explicitly different from the existing privileged information learning methods. The existing methods do not consider the weak label problem. They assume that the training instances are accurately labeled. That is to say, different labelers give the same label to one instance, i.e., $y_i^1, y_i^2, \ldots, y_i^m$ being the same. There is no ambiguity in the instance labels. Different from the existing methods, our method takes the weak label problem into account. Due to distinct professional knowledge and subjective factors, different labelers may give different labels to one instance, i.e., $y_i^1, y_i^2, \ldots, y_i^m$ may not being the same. There exists ambiguity in the instance labels. To deal with the label ambiguity, the PLWB method is proposed.

Let $f(x) = (w, x) + b$ be the hyper-plane in the training space, and $f^*(x) = (w^*, x) + b^*$ be the correcting function. In Eq. (2), we construct the slack ξ_i which is comprised with the error term ξ_i^* and the smooth function $(w^*, x_i^*) + b^*$. To make sure that the slack ξ_i is larger than 0, we let the error term ξ_i^* and the smooth function $(w^*, x_i^*) + b^*$ in Eq. (2) larger than 0, as shown in (3).

$$\xi_i = [(w^*, x_i^*) + b^*] + \xi_i^*, \quad i = 1, \ldots, l \tag{2}$$

$$(w^*, x_i^*) + b^* \geq 0, \quad \xi_i^* \geq 0, \quad i = 1, \ldots, l \tag{3}$$

Based on this, the learning problem of PLWB can be formulated as

$$\begin{aligned} \min_{\xi_i^* \geq 0} \quad & \frac{1}{2}\|w\|^2 + \frac{\gamma}{2}\|w^*\|^2 + C\sum_{i=1}^{l}[(w^*, x_i^*) + b^*] + \theta C \sum_{i=1}^{l} \xi_i^* \\ s.t. \quad & \sum_{j=1}^{m}(r_j y_i^j)[(w, x_i) + b] \geq 1 - [(w^*, x_i^*) + b^*] - \xi_i^*, \\ & (w^*, x_i^*) + b^* \geq 0, \\ & \sum_{j=1}^{m} r_j = 1, \\ & 0 \leq r_j \leq 1, \end{aligned} \tag{4}$$

- where w and b are the corresponding norm vector and bias referring to the training data. w^* and b^* are the corresponding norm vector and bias referring to the privileged data. $C \geq 0$ is the regularized parameter. ξ_i^* is an error term, which is added to make the correction function smoother. $\theta \geq 0$ is used to tradeoff the slack variable ξ_i^* and the other terms in problem (4). $\gamma \geq 0$ controls the influence of privileged information on the model. r_j is the weight of the jth labeler.
- The term $\frac{\gamma}{2}\|w^*\|^2$ is used to confine the capacity of the function space containing $f^*(x)$. The traditional SVM uses slack variables to balance the distance from the instances to the hyper-plane. Distinctively, our method uses privileged data to adjust the distance from the instances to the hyper-plane. Thus, the constraint $(w^*, x_i^*) + b^* \geq 0$ is imposed. When it has $\gamma = 0$ and $\theta = 1$, the privileged information will have no effect on the model, and problem (4) is degraded into a standard SVM model with weak labels.
- $y_i^j \in \{1, -1\}$ is the label of instance x_i, which is annotated by the jth labeler. It is available to us. r_j is the weight of the jth labeler, which is unknown to us and needed to be optimized. We calculate the label Y_i of instance x_i by $Y_i = \sum_{j=1}^{m} r_j y_i^j$. In traditional supervised learning, each labeler is given a weight equally or randomly, which leads to the ambiguity of instance labels. Different from the traditional supervised learning methods, we incorporate the labeler weight in the model and optimize it in the learning process.

3.3. PLWB optimization

The variables w, w^*, b, b^*, ξ^* and r_j are unknown to us, and problem (4) is difficult to resolve. In the following, a heuristic framework will be employed to calculate these unknown variables.

Specifically, the heuristic framework is comprised with two steps. In the first step, we fix the weight $r_j = \overline{r_j}$, and obtain the values of w, w^*, ξ^*, b and b^* by solving the problem (5). Then, we fix w, w^*, b and b^*, and get the values of r_j by solving the problem (14). The above two steps repeat alternately until the termination criterion is met. In the following, we show the two steps in details.

3.3.1. Fix the weight r and optimize the classifier

We fix r_j to be $\overline{r_j}$. Then, the first set of constraints in problem (4) is transformed into $\sum_{j=1}^{m}(\overline{r} y_i^j)[(w, x_i) + b] \geq 1 - [(w^*, x_i^*) + b^*] - \xi_i^*$. Based on these, the objective function (4) is changed into

$$\begin{aligned} \min_{\xi_i^* \geq 0} \quad & \frac{1}{2}\|w\|^2 + \frac{\gamma}{2}\|w^*\|^2 + \theta C \sum_{i=1}^{l} \xi_i^* + C\sum_{i=1}^{l}[(w^*, x_i^*) + b^*] \\ s.t. \quad & \sum_{j=1}^{m}(\overline{r_j} y_i^j)[(w, x_i) + b] \geq 1 - [(w^*, x_i^*) + b^*] - \xi_i^*, \\ & (w^*, x_i^*) + b^* \geq 0. \end{aligned} \tag{5}$$

When the weight r_j is fixed, the above function is a quadratic optimization (QP) problem. It can be resolved by applying the Lagrange method. The Lagrange function of problem (5) can be given as follows.

$$\begin{aligned} L(w, w^*, b, b^*, \alpha, \beta, \sigma) = & \\ & \frac{1}{2}\|w\|^2 + \frac{\gamma}{2}\|w^*\|^2 + C\sum_{i=1}^{l}((w^*, x_i^*) + b^*) + \theta C \sum_{i=1}^{l} \xi_i^* \\ & - \sum_{i=1}^{l} \alpha_i [(\sum_{j=1}^{m} \overline{r} y_i^j)((w, x_i) + b) - 1 \\ & + ((w^*, x_i^*) + b^*) + \xi_i^*] \\ & - \sum_{i=1}^{l} \beta_i [(w^*, x_i^*) + b^*] - \sum_{i=1}^{l} \sigma_i \xi_i^* \end{aligned} \tag{6}$$

where $\alpha_i \geq 0$, $\beta_i \geq 0$ and $\sigma_i \geq 0$ are Lagrangian multiples. We differentiate the Lagrange function (6) with respect to w, w^*, b, b^* and ξ_i^* and set the derivatives to be zero. The following equations can be obtained.

$$w = \sum_{i=1}^{l} \alpha_i (\sum_{j=1}^{m} r_j y_i^j) x_i, \tag{7}$$

$$w^* = \frac{1}{\gamma}(\sum_{i=1}^{l} \alpha_i x_i^* + \sum_{i=1}^{l} \beta_i x_i^* - C\sum_{i=1}^{l} x_i^*), \tag{8}$$

$$\sum_{i=1}^{l} \alpha_i (\sum_{j=1}^{m} r_j y_i^j) = 0, \tag{9}$$

$$\sum_{i=1}^{l} C - \sum_{i=1}^{l} \alpha_i - \sum_{i=1}^{l} \beta_i = 0, \tag{10}$$

$$\theta C - \alpha_i - \sigma_i = 0. \tag{11}$$

By substituting Eqs. (7)–(11) into problem (6), the dual form (12) can be obtained.

$$\begin{aligned}\max_{\alpha,\beta,\sigma}\ & \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j(\sum_{j_1=1}^{m}\overline{r_{j_1}}y_i)(\sum_{j_2=1}^{m}\overline{r_{j_2}}y_j)(x_i,x_j)\\ & +\frac{1}{2\gamma}\sum_{i,j=1}^{l}\alpha_i\alpha_j(\alpha_i+\beta_i-C)(\alpha_j+\beta_j-C)(x_i^*,x_j^*)\\ s.t.\ & \sum_{i=1}^{l}(\alpha_i+\beta_i-C)=0,\\ & \sum_{i=1}^{l}\alpha_i(\sum_{j=1}^{m}\overline{r_j}y_i^j)=0,\\ & 0\le\alpha_i\le\theta C,\quad 0\le\beta_i,\quad 0\le\sigma_i.\end{aligned}\tag{12}$$

We can obtain the Lagrange multipliers α, β and σ by solving the dual form (12). A new instance x_T can be predicted by using the following function.

$$f(x_T)=\sum_{i=1}^{l}\alpha_i(\sum_{j=1}^{m}r_jy_i^j)(x_i,x_T)+b.\tag{13}$$

3.3.2. Fix the classifier and update the weight r

We can acquire the Lagrange multipliers by resolving the dual form (12). Based on these Lagrange multipliers, we can calculate w, w^*, b and b^*. In problem (4), after w, w^*, b and b^* are known, we can optimize the weight r.

The weight r can be updated by optimizing the classification errors ξ^*. Hence, we obtain the following problem:

$$\begin{aligned}\min_{r,\xi}\ & \sum_{i=1}^{l}\xi_i^*\\ s.t.\ & \sum_{j=1}^{m}(r_jy_i^j)[(w,x_i)+b]\ge 1-[(w^*,x_i^*)+b^*]-\xi_i^*,\\ & \sum_{j=1}^{m}r_j=1,\\ & 0\le r_j\le 1.\end{aligned}\tag{14}$$

Since the values of w, w^*, b and b^* are available, (14) is a linear programming (LP) problem. We can resolve it by using off-the-shelf LP solvers.

3.3.3. Heuristic framework

To solve problem (4), we put forward a heuristic framework. Our algorithm includes two steps: (1) obtaining the classifier when the weight r is fixed; (2) fixing the classifier to update the weight r. We conduct the above two steps alternately until the algorithm stops. The pseudo codes of our method are illustrated in Algorithm 1. Specifically, we input the training instances, privileged instances, as well as the parameters γ, C, θ and ϵ. Here, the parameters γ, C and θ tradeoff the different terms in problem (4), and ϵ determines the termination of the PLWB algorithm. Let $F_{val}(t)$ be the objective function value of problem (5) and t be the number of iterations. Firstly, initialize $F_{val}(t)=\infty$ and $t=0$. Then, the iteration begins. We obtain w, w^*, b and b^* by solving problem (5). If it has $t=1$, namely in the first iteration, we initialize the labeler weights r. Otherwise, we update r based on problem (14) by fixing w, w^*, b and b^*. After the labeler weights r are obtained, we fix them and solve problem (5) to obtain w, w^*, b and b^*. Define F_{max} be the maximum value between $|F_{val}(t-1)|$ and $|F_{val}(t)|$, where $|F_{val}(t-1)|$ and $|F_{val}(t)|$ are the absolute values of $F_{val}(t-1)$ and $F_{val}(t)$, respectively. $|F_{val}(t)|-|F_{val}(t-1)|$ is the difference of F_{val} in two consecutive iterations. If the proportion of $|F_{val}(t)|-|F_{val}(t-1)|$ and F_{max} is smaller than a threshold ϵ, i.e., $|F_{val}(t)|-|F_{val}(t-1)|<\epsilon F_{max}$, the PLWB algorithm terminates. The threshold ϵ is usually set to be a small value. As in [29], we set ϵ to be 0.1 in the experiments. Lastly, we output the decision functions $f(x)=(w,x)+b$ for the training instances and $f^*(x)=(w^*,x)+b^*$ for the privileged instances.

Algorithm 1 PLWB algorithm

Input: Training instances, privileged instances, γ, C, θ and ϵ
Output: $f(x)$ and $f^*(x)$.
1: $t=0$;
2: Initialize $F_{val}(t)=\infty$;
3: **repeat**
4: $t=t+1$;
5: **if** $t=1$ **then**
6: Initialize r;
7: **else**
8: Update r based on (14) by fixing w, w^*, b and b^*;
9: **end if**
10: Substitute r and solve problem (5);
11: Compute w and w^* according to Eqs. (7)-(8), respectively;
12: Let $F_{val}(t)$ be the decision function value of problem (5);
13: Let $F_{max}=\max\{|F_{val}(t-1)|,|F_{val}(t)|\}$;
14: **until** $|F_{val}(t)-F_{val}(t-1)|<\epsilon F_{\max}$
15: Return $f(x)=(w,x)+b$ and $f(x^*)=(w^*,x)+b^*$.

4. Experiments

We compare PLWB with six state-of-the-art algorithms on four real-world datasets — Handwritten (HW), Caltech-101, Alzheimer's Disease Neuroimaging Initiative (ADNI) and Animals-with-Attributes (AWA).

4.1. Dataset description

To verify the effectiveness of PLWB, experiments are conducted on several real-world datasets, including Handwritten (HW) [30], Caltech-101 [31], Alzheimer's Disease Neuroimaging Initiative (ADNI) and Animals-with-Attributes (AWA) [32]. The collection and preprocessing of these datasets are described as follows.

4.1.1. Data collection

Handwritten (HW) dataset: It is available in the UCI machine learning repository. It contains "0" to "9" handwritten digits, and has 10 classes. Each class has 200 handwritten digits which are transformed into binary images. The sample digits can be seen in Fig. 2.

Animals-with-Attributes (AwA) dataset: It contains 50 animal classes. The number of images in the 50 classes is 30 475 in total. Since there are too many classes, we pick up 20 classes from this dataset to perform the experiments. The selected classes include "weasel", "beaver", "bobcat", "tiger", "collie", "horse", "lion" and so on. These 20 classes contain 6180 images which are used in the experiment. Fig. 3. shows the sample images in the 20 classes.

Caltech-101 dataset: It is provided by [31] and is for the task of object recognition. This dataset is comprised of 101 classes and 8677 images. Since the Caltech-101 dataset is highly imbalanced, we conduct experiments on the 10 top classes, i.e., "Boat", "Tree", "Bass", "Crab", "Cup", "Dog", "Sofa", "Car", "Bus" and "Cow". The sample images from the Caltech-101 dataset are shown in Fig. 4.

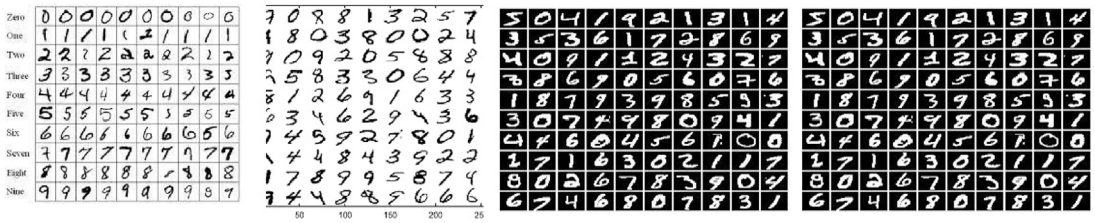

Fig. 2. Sample images from the Handwritten (HW) dataset.

Fig. 3. Sample images from the Animals-with-Attributes (AWA) dataset.

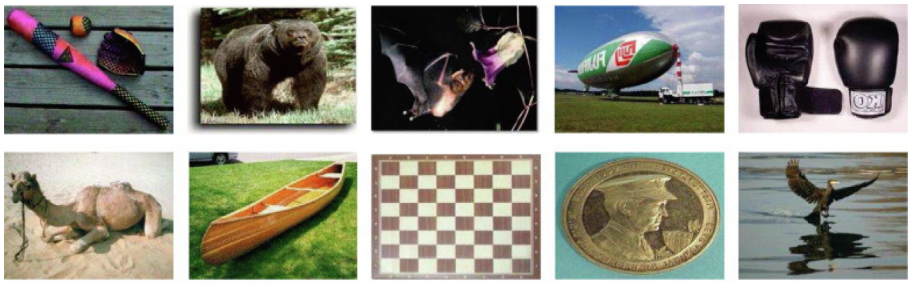

Fig. 4. Sample images from the Caltech-101 dataset.

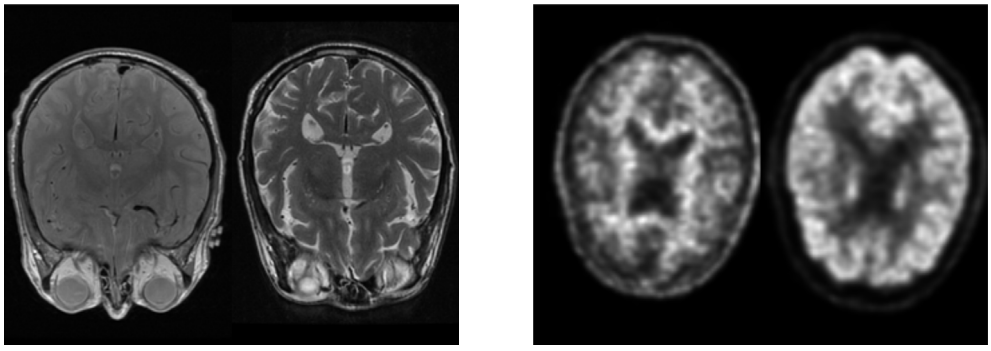

Fig. 5. Sample images from the ADNI database.

Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset: It consists of magnetic resonance imaging (MRI) and positron emission tomography (PET) images from 202 ADNI participants which include 51 Alzheimer's dementia (AD) patients, 99 mild cognitive impairment (MCI) patients, and 52 normal controls (NC). The sample images of MRI and PET from the ADNI dataset are shown in Fig. 5.

4.1.2. Data preprocessing

The above datasets are multi-class classification datasets. We take the following procedures to transform these multi-class classification datasets into weakly labeled binary class classification datasets. Firstly, we transform the multi-class class classification datasets into binary class classification datasets. For a dataset with K classes, one class is chosen as the positive

Table 1
Details of the datasets used in the experiments.

Dataset	Total Inst.	Train Inst.	Test Inst.	Privileged Inst.	Train Dim.	Privileged Dim.
HW	2000	1800	200	1800	76	64
AwA	6180	5562	618	5562	256	252
Caltech-101	8677	867	867	7803	81	252
ADNI	200	20	20	180	93	93

Table 2
Classification accuracy on the Handwritten (HW), Caltech-101 and ADNI datasets.

Dataset	SVM	SVM+	R-CTSVM+	L2- SVM+	R-SVM+	S-ISCN+	PLWB
HW0	88.71 ± 1.29	91.11 ± 0.63	90.72 ± 0.71	91.74 ± 0.64	91.83 ± 0.59	90.17 ± 0.66	**96.95 ± 0.24**
HW1	83.17 ± 1.36	84.25 ± 0.54	88.75 ± 0.67	89.37 ± 0.81	90.55 ± 0.77	87.33 ± 0.71	**94.94 ± 0.32**
HW2	86.13 ± 1.13	90.57 ± 0.71	88.33 ± 0.92	90.71 ± 1.04	90.47 ± 0.93	87.12 ± 0.83	**95.72 ± 0.22**
HW3	87.65 ± 1.38	90.28 ± 1.24	88.47 ± 0.75	92.53 ± 0.63	92.32 ± 0.94	89.43 ± 0.69	**96.47 ± 0.48**
HW4	88.19 ± 1.36	90.72 ± 1.08	89.93 ± 1.21	91.17 ± 0.72	90.73 ± 0.78	90.22 ± 0.74	**94.92 ± 0.57**
HW5	85.13 ± 1.22	91.28 ± 0.96	89.74 ± 0.87	91.42 ± 0.93	90.63 ± 1.04	88.63 ± 0.82	**94.25 ± 0.49**
HW6	88.92 ± 1.05	90.17 ± 1.23	91.15 ± 0.87	91.34 ± 0.63	91.97 ± 0.74	91.37 ± 0.91	**95.15 ± 0.38**
HW7	84.15 ± 0.87	89.21 ± 0.91	89.77 ± 1.24	88.43 ± 0.86	87.52 ± 0.75	86.39 ± 0.93	**93.35 ± 0.37**
HW8	87.42 ± 0.92	90.53 ± 0.88	89.36 ± 0.73	91.25 ± 1.23	90.34 ± 0.74	88.96 ± 0.87	**94.72 ± 0.42**
HW9	87.42 ± 1.16	91.35 ± 0.97	89.56 ± 0.85	92.17 ± 1.04	91.71 ± 0.77	90.15 ± 0.93	**96.15 ± 0.56**
Cal101.boat	74.12 ± 0.93	78.56 ± 0.92	80.53 ± 0.84	81.25 ± 1.03	79.55 ± 0.78	81.43 ± 0.69	**86.31 ± 0.44**
Cal101.tree	73.52 ± 0.87	77.42 ± 0.83	75.34 ± 0.71	77.98 ± 0.96	76.32 ± 1.02	78.21 ± 0.85	**83.25 ± 0.45**
Cal101.bass	80.14 ± 0.99	83.36 ± 1.21	82.55 ± 1.16	82.97 ± 0.87	83.16 ± 0.95	82.94 ± 0.83	**88.78 ± 0.52**
Cal101.crab	77.44 ± 0.97	79.58 ± 0.82	78.41 ± 1.08	80.75 ± 1.13	81.35 ± 0.67	82.33 ± 0.77	**86.96 ± 0.42**
Cal101.cup	76.13 ± 1.12	77.42 ± 0.89	80.82 ± 0.94	80.79 ± 0.85	79.57 ± 0.72	80.17 ± 0.95	**84.36 ± 0.52**
Cal101.dog	74.26 ± 0.81	77.14 ± 0.74	76.26 ± 0.61	77.35 ± 1.07	76.47 ± 0.76	75.68 ± 0.79	**81.64 ± 0.44**
Cal101.sofa	71.25 ± 0.77	72.53 ± 0.86	73.71 ± 0.75	72.95 ± 0.96	73.06 ± 0.72	72.91 ± 0.81	**80.65 ± 0.55**
Cal101.car	70.25 ± 0.92	73.63 ± 1.16	71.95 ± 0.73	73.46 ± 0.87	72.46 ± 0.69	74.11 ± 0.88	**80.23 ± 0.57**
Cal101.bus	77.55 ± 0.94	80.78 ± 0.86	81.35 ± 0.77	80.76 ± 0.65	81.11 ± 1.04	81.36 ± 0.94	**85.35 ± 0.49**
Cal101.cow	80.27 ± 0.72	82.34 ± 1.13	81.75 ± 0.87	82.34 ± 0.89	81.97 ± 0.91	82.37 ± 0.88	**86.13 ± 0.56**
ADNI.AD	82.27 ± 0.62	88.64 ± 0.73	82.95 ± 0.47	88.24 ± 0.59	84.77 ± 0.61	85.33 ± 0.75	**90.32 ± 0.46**
ADNI.NC	86.71 ± 0.82	87.43 ± 0.93	86.54 ± 0.72	88.14 ± 0.69	88.27 ± 0.73	85.96 ± 0.91	**91.27 ± 0.49**
ADNI.MCI	84.23 ± 0.53	87.41 ± 0.63	85.51 ± 0.67	87.64 ± 0.92	88.57 ± 0.71	86.87 ± 0.72	**93.32 ± 0.65**

class and the other classes are regarded as the negative class. K sub-datasets can be formed. In this way, a number of binary class sub-datasets from the HW, Caltech-101, ADNI and AWA datasets can be obtained, as shown in Tables 2 and 3. Specifically, the HW dataset contains 10 classes and thus we obtain 10 sub-datasets, i.e., HW0–HW9. The Caltech-101 dataset has 10 classes and we get 10 sub-datasets, i.e., Cal101.boat, Cal101.tree, Cal101.bass and so on. The ADNI dataset consists of 3 classes and 3 sub-datasets, i.e., ADNI.AD, ADNI.NC and ADNI.MCI, are attained. The AWA dataset is comprised of 20 classes and 20 sub-datasets, i.e., AwA.bat, AwA.beaver, AwA.cow and so on, are achieved, as shown in Table 3.

Secondly, we follow the routine in [33] to form the weak label for each image in the sub-datasets. Specifically speaking, we set the number of labelers to be ten, and randomly allocate a weight r_j to each labeler. This weight indicates the importance of each labeler. Then, we assign the label for each instance. If the instance's ground-truth label is positive, we let six labelers assign y_k^j as $+1$, and the other labelers assign y_k^j as -1. If the instance's ground-truth label is negative, we let six labelers assign y_k^j as $+1$, and the other labelers assign y_k^j as $+1$. Lastly, we use $Y_k = \sum_{j=1}^{m} r_j y_k^j$ to calculate the label of an instance. The instance is relabeled as positive if it has $Y_k \geq 0$. Otherwise, it is relabeled as negative. In the above procedure, every labeler gives a label to each instance, and this label is associated with a random weight. The initial label of an instance is computed by $Y_k = \sum_{j=1}^{m} r_j y_k^j$. Let r_{WB} denote the percentage of correctly labeled instances. In Section 4.4, we set r_{WB} to be 80%. That is to say, 80% of the training instances are correctly labeled according to their ground-truth labels. In Section 4.5, we let r_{WB} be 80%, 60%, 40% and 20%, respectively, and investigate the performance of our method and baselines with different percentages of weak labels.

Lastly, we extract the training features and privileged features from each image in the sub-datasets. For the HW sub-datasets, we use the Fourier coefficients of the character shapes (FOU) feature with 76 attributes as the training feature and the Karhunen-love coefficients (KAR) feature with 64 attributes as the privileged feature. For the AwA sub-datasets, the color histogram feature with 256 attributes is extracted as the training feature and the histogram of oriented gradients feature (HOG) with 252 attributes is as the privileged feature. For the Caltech-101 sub-datasets, we utilize the global image descriptor (GIST) feature with 81 attributes as the training feature and the HOG feature with 252 attributes as the privileged feature. For the ADNI sub-datasets, we extract the gray matter of 93 regions of interest (ROI) from the MRI image as the training feature and the average intensity of each ROI in the PET image as the privileged feature. Table 1 shows the dimension numbers of the training features and privileged features for the experimental datasets.

4.1.3. Data output

After the above data preprocessing, we obtain 10 sub-datasets from the HW dataset, 20 sub-datasets from the AWA dataset, 3 sub-datasets from the ADNI dataset and 10 sub-datasets from the Caltech-101 dataset. Tables 2 and 3 show all the obtained sub-datasets. In each sub-dataset, the instances are weakly labeled and contain both the training features and privileged features. Table 1 presents the total instance number (Total Inst.), training instance number (Train Inst.), testing instance number (Test Inst.), privileged instance number (Privileged Inst.), dimension of training instances (Train Dim.), and dimension of privileged instances (Privileged Dim.) in the experimental datasets. The subsequent experiments will be conducted on these sub-datasets.

4.2. Baselines and parameter setting

4.2.1. Baselines

We compare PLWB with six baselines, i.e., SVM, SVM+ [1], robust capped L1-norm twin support vector machine with privileged information (R-CTSVM+) [34], L2-SVM+ [5], $L_{\frac{1}{2}}$-norm-regularization-based sparse ISCN+ (S-ISCN+) [35] and robust

Table 3
Classification accuracy on the Animals-with-Attributes (AWA) dataset.

Dataset	SVM	SVM+	R-CTSVM+	L2-SVM+	R-SVM+	S-ISCN+	PLWB
AwA.bat	63.74 ± 0.92	66.56 ± 0.77	70.45 ± 0.83	72.15 ± 0.95	71.35 ± 0.87	72.33 ± 0.87	**76.24 ± 0.45**
AwA.beaver	72.14 ± 1.05	80.55 ± 1.12	79.15 ± 1.22	81.56 ± 0.94	82.38 ± 0.82	81.92 ± 1.03	**86.33 ± 0.52**
AwA.cow	65.15 ± 0.97	68.75 ± 0.92	70.58 ± 0.74	61.26 ± 0.82	69.36 ± 1.04	68.31 ± 0.83	**75.66 ± 0.53**
AwA.weasel	67.83 ± 0.97	60.68 ± 0.95	69.18 ± 0.81	70.36 ± 0.83	70.98 ± 0.74	67.34 ± 0.91	**74.67 ± 0.43**
AwA.collie	63.91 ± 0.84	65.85 ± 1.21	68.11 ± 0.98	69.27 ± 0.71	70.78 ± 0.95	65.33 ± 1.18	**74.56 ± 0.51**
AwA.tiger	68.25 ± 1.23	70.77 ± 0.98	61.25 ± 0.72	70.92 ± 0.86	61.16 ± 0.77	69.33 ± 0.97	**75.93 ± 0.35**
AwA.deer	65.16 ± 0.92	68.38 ± 0.75	71.98 ± 0.86	75.37 ± 0.98	75.65 ± 1.23	71.23 ± 0.78	**80.66 ± 0.42**
AwA.fox	74.11 ± 1.02	77.46 ± 1.23	76.16 ± 0.75	78.36 ± 0.86	77.25 ± 0.92	78.12 ± 0.86	**83.64 ± 0.46**
AwA.panda	75.46 ± 0.85	76.83 ± 0.81	79.45 ± 0.99	76.13 ± 1.21	78.94 ± 0.94	77.36 ± 0.85	**83.43 ± 0.52**
AwA.gorilla	74.35 ± 0.91	75.56 ± 0.85	79.37 ± 1.22	79.16 ± 0.96	75.37 ± 1.21	77.32 ± 0.98	**84.77 ± 0.33**
AwA.horse	72.17 ± 0.93	73.56 ± 0.83	75.44 ± 0.98	76.95 ± 0.99	74.12 ± 0.75	76.93 ± 0.91	**81.98 ± 0.42**
AwA.whale	77.55 ± 0.96	79.26 ± 0.94	79.13 ± 0.87	77.63 ± 1.06	78.41 ± 0.99	79.61 ± 0.93	**84.67 ± 0.42**
AwA.moose	70.46 ± 0.88	76.51 ± 0.91	75.12 ± 1.23	73.33 ± 1.04	75.15 ± 0.81	76.53 ± 0.89	**81.57 ± 0.55**
AwA.otter	82.35 ± 0.78	88.77 ± 0.97	86.13 ± 0.85	83.68 ± 1.24	87.55 ± 1.03	86.34 ± 0.92	**91.26 ± 0.33**
AwA.lion	83.15 ± 0.85	84.66 ± 0.98	86.54 ± 0.97	87.59 ± 0.63	88.35 ± 0.75	86.71 ± 0.74	**92.75 ± 0.42**
AwA.bear	80.55 ± 0.86	81.75 ± 0.71	87.34 ± 0.88	88.55 ± 0.67	87.19 ± 1.06	85.69 ± 0.73	**92.43 ± 0.31**
AwA.ox	80.95 ± 0.76	81.54 ± 0.84	87.51 ± 0.92	86.54 ± 0.91	86.45 ± 0.75	84.32 ± 0.89	**93.66 ± 0.42**
AwA.zebra	83.35 ± 1.15	87.56 ± 0.65	86.57 ± 0.92	86.16 ± 0.78	86.47 ± 0.84	87.11 ± 0.73	**91.87 ± 0.42**
AwA.sheep	81.86 ± 0.94	87.26 ± 0.79	87.55 ± 0.85	85.14 ± 1.05	89.26 ± 0.77	87.63 ± 0.88	**93.65 ± 0.44**
AwA.skunk	81.46 ± 0.95	82.57 ± 0.86	88.17 ± 0.97	84.16 ± 1.03	87.17 ± 1.15	85.71 ± 0.92	**91.35 ± 0.45**
AwA.pig	80.46 ± 0.84	81.53 ± 1.21	84.28 ± 0.95	87.97 ± 0.87	84.16 ± 1.06	86.13 ± 0.99	**93.66 ± 0.51**
AwA.walrus	84.25 ± 0.93	86.56 ± 0.89	85.17 ± 0.79	88.36 ± 1.12	87.27 ± 0.86	87.91 ± 0.91	**92.35 ± 0.42**
AwA.wolf	81.28 ± 0.95	83.27 ± 0.89	87.16 ± 0.78	86.18 ± 1.06	83.38 ± 0.97	85.99 ± 0.92	**90.58 ± 0.31**

support vector machine with privileged information (R-SVM+) [36]. Among these baselines, SVM is a typical binary classification method, which does not utilize privileged information. SVM+, R-SVM+, L2-SVM+ and R-CTSVM+ are SVM-based privileged information learning methods. S-ISCN+ is a network-based privileged information learning method.

1. SVM: It is the standard SVM, which trains the classifier by using only the training data and does not utilize the privileged data.

2. SVM+[1]: It is a typical privileged information learning method, which introduces the privileged information into SVM and modifies the slack variables according to the privileged information.

3. L2-SVM+[5]: It applies L2-loss in the ρ-SVM framework and replaces the slack variables with correcting functions that are obtained from the privileged information.

4. R-CTSVM+[34]: It is a robust capped L1-norm twin SVM, which considers the abnormal points by introducing the capped L1-norm and introduces the privileged information in the learning process.

5. R-SVM+[36]: It is a robust SVM+ algorithm, which addresses the potential noises in the training data and privileged data.

6. S-ISCN+[35]: It is a network-based method, which involves the privileged information into learning the stochastic configuration network (SCN).

4.2.2. Parameter setting

In the experiment, the linear kernel is utilized. For the baselines, we follow the same settings in their corresponding papers to tune the parameters. SVM has one parameter C, which is chosen in $2^{[-2,-1,\ldots,3,4]}$. SVM+ and L2-SVM+ have two parameters C and γ. Parameter C is selected in $2^{[-2,-1,\ldots,3,4]}$. Parameter γ is picked up from $10^{[-2,-1,\ldots,3,4]}$. R-SVM+ has four parameters C, λ, γ and σ. Parameter C is selected in $2^{[-2,-1,\ldots,1,2]}$. Parameter γ is picked up from $10^{[-2,-1,\ldots,1,2]}$. Parameter λ is chosen from $10^{[-5,-4,\ldots,0,1]}$. As in [1], σ should be a relatively large value and is fixed to be 1000. R-CTSVM+ has five parameters $C_1, C_2, \lambda, \epsilon_1$ and ϵ_2. We let $C_1 = C_2$ and $\epsilon_1 = \epsilon_2$. Parameters C_1 and C_2 are selected in $10^{[-4,\ldots,-2,-1]}$. Parameter λ is picked up from $10^{[-5,-4,\ldots,0,1]}$. Parameters ϵ_1, and ϵ_2 are chosen in $\{0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2\}$. S-ISCN+ has three parameters η, γ and μ. Parameters η and μ are selected in $10^{[-6,-5,\ldots,5,6]}$. γ is picked up from the range of $10^{[-5,-4,\ldots,-1,0]}$. For PLWB, ϵ controls the convergence of our algorithm. Following the same routine in [29], ϵ is set to be 0.1. θ adjusts the penalty between the correcting function and the training error ξ^*. As suggested in [1], the value of θ should be large enough and we fix it to be 1000. Hence, there are two parameters C and γ needed to be tuned in PLWB. C and γ are picked up from $2^{[-2,\ldots,3,4]}$ and $10^{[-2,\ldots,2,3]}$, respectively.

As in [37], we employ two-fold cross validation to determine the optimal parameters. Specifically, each experimental sub-dataset is split into two halves. One half is considered as the training set and the other half is treated as the testing set. The classification accuracy is recorded. The parameter combination which leads to the highest classification accuracy is selected as the optimal parameters. It is noted that the above procedure is applied only once for each sub-dataset. Once the parameters are determined, they are used in all the subsequent experiments. After the optimal parameters are fixed, ten-fold cross validation is applied to the sub-datasets. The average result with ten-fold cross validation is reported as the final one.

4.3. Implementation setup and working environment

We execute the experiments in Matlab on the laptop with a 2.2-GHz processor and 4-GB memory. During the optimization process, the QP problems in SVM, SVM+, R-CTSVM+, L2-SVM+, R-SVM+ and PLWB are solved via the QP toolbox in MATLAB. The codes of our method can be available from https://github.com/yzx2god/PLWB.git.

4.4. Experimental results

Tables 1 and 2 exhibit the classification accuracy of SVM, SVM+, L-2 SVM+, R-CTSVM+, R-SVM+, S-ISCN+ and PLWB on the sub-datasets. From Tables 1 and 2, we can see that PLWB performs better than the baselines. For example, on the HW0 sub-dataset which considers the digital "0" as positive and the other digitals as negative, the classification accuracy of PLWB is 96.95, which is explicitly higher than SVM (88.71), SVM+ (91.11), R-CTSVM+ (90.72), L2-SVM+ (91.74), R-SVM+ (91.83) and S-ISCN+ (90.17). Among all the HW sub-datasets, PLWB obtains a minimum of 2.83 and up to 11.77 improvements, compared with SVM, SVM+, R-CTSVM+, L2-SVM+, R-SVM+ and S-ISCN+.

In Tables 1 and 2, it can be seen that SVM+, R-CTSVM+, L2-SVM+, RSVM+ and S-ISCN+ have better performance than SVM. On the Cal101.bass sub-dataset, the corresponding accuracy of SVM+, R-CTSVM+, L2-SVM+, R-SVM+ and S-ISCN+ is 83.36, 82.55, 82.97, 83.16 and 82.94, which is higher than SVM (80.14). Among all the Caltech-101 sub-datasets, SVM+, R-CTSVM+, L2-SVM+, R-SVM+ and S-ISCN+ obtain a minimum of 1.28 and up to 7.31 improvements, compared with SVM. The reason is that SVM uses only the training information to learn the classifier, while SVM+, R-SVM+, R-CTSVM+, L2-SVM+ and S-ISCN+ are privileged information learning methods, which can incorporate both the training information and privileged learning into constructing the classifier. The privileged data can provide additional information which can be used to refine the classifier and boost the classification result. Moreover, it is observed that PLWB obtains explicitly better classification accuracy than SVM+, R-SVM+, R-CTSVM+, L2-SVM+ and S-ISCN+. For example, the accuracy of PLWB is 83.64 on the AWA.fox sub-dataset, which is higher than SVM+ (74.11), R-SVM+ (77.25), R-CTSVM+ (76.16), L2-SVM+ (78.36) and S-ISCN+ (78.12) at 9.53, 6.39, 7.48, 5.28 and 5.52, respectively. Among all the AWA sub-datasets, PLWB achieves a minimum of 2.49 and up to 14.77 improvements, compared with SVM+, R-CTSVM+, L2-SVM+, R-SVM+ and S-ISCN+. As presented in Section 4.1.2, every labeler assigns a label to each instance, and this label is associated with a random weight. The initial label of an instance is computed by $Y_k = \sum_{j=1}^{m} r_j y_k^j$. However, SVM+, R-CTSVM+, L2-SVM+, R-SVM+ and S-ISCN+ do not consider the weak label learning problem. Hence, they learn the classifiers directly on the initial labels in which the weights of labelers are randomly assigned and do not change in the subsequent training. Different from these methods, PLWB takes the weak label learning problem into account and continuously updates the weights of labelers to obtain the overall optimization. The better performance of PLWB over SVM+, R-SVM+, R-CTSVM+, L2-SVM+ and S-ISCN+ confirms the effectiveness of our method in refining the weights of labelers.

Furthermore, the standard deviations on the sub-datasets are shown in Tables 1 and 2. It is observed from Tables 1 and 2 that PLWB has the smallest standard deviation among all the methods. Taking the HW9 sub-dataset as an example, the standard deviations of SVM, SVM+, R-CTSVM+, L-2 SVM+, R-SVM+ and S-ISCN+ are 1.16, 0.97, 0.85, 1.04, 0.77 and 0.93, respectively, while that of our method is 0.56. It is seen that PLWB can achieve a more stable performance than the baselines. The baseline methods assume that the instance is accurately labeled and their learning models are constructed based on this assumption. However, when there exist weak labels in the data, the learnt models may be more sensitive to weak labels and less stable for re-sampling. By contrast, PLWB considers the weak label learning problem. Our model is less sensitive to weak labels and more stable for re-sampling.

4.5. Performance with different percentage of correctly labeled instances

We evaluate our method by setting different percentage r_{WB} of the correctly labeled instances. Fig. 6 presents the classification accuracy of our method and the baselines when the percentage r_{WB} of correctly labeled instances varies from 20% to 80%. In Fig. 6, the x-axis is the percentage of correctly labeled instances and the y-axis is the classification accuracy. In Fig. 6, when the percentage of correctly labeled instances increases, the classification accuracy of all methods goes up synchronously. This is because as the increase of correctly labeled instances, the dataset contains less weakly labeled information and the performance of classifiers improves. Moreover, PLWB obtains consistently better classification performance than the baselines — SVM, SVM+, R-CTSVM+, L2-SVM+, R-SVM+ and S-ISCN+. The baselines do not consider the weak label learning problem and the weakly labeled data limits their classification accuracy. Distinctively, PLWB adopts a heuristic framework to update the weight of each labeler and incorporates these weights into boosting the performance of classifiers.

4.6. Parameter sensitivity analysis

We investigate the influence of parameters C and γ on our model. The parameter C determines the importance of the correction function which is obtained from the privileged data to the whole model. The parameter γ balances the two margins, i.e., $\|w\|^2$ and $||w^*||^2$. In the following, we take two sub-datasets (i.e., HW0 and AwA.tiger) as examples to illustrate the sensitivity of these two parameters. In Fig. 7(a) and (b), we fix the parameter γ to be the optimal value and vary C from 0.25 to 16. In Fig. 7(c) and (d), we fix the parameter C to be the optimal value and vary γ from 0.01 to 1000. In these figures, we can observe that satisfactory accuracy can be obtained when the values of C and γ are relatively large. This may be because both C and γ are related to the privileged data. When the values of C and γ are relatively large, the correction function $f^*(x_i^*) = (w^*, x_i^*) + b^*$ and the margin $\|w^*\|^2$ will greatly affect the performance of the model. It implies that the utilization of privileged data can effectively enhance the classification performance.

4.7. Time analysis

In the following, we analyze the time of learning parameters, building models and predicting new instances for the proposed PLWB method and the privileged information learning methods (i.e., SVM+, L2-SVM+, R-CTSVM+, R-SVM+ and S-ISCN+).

The time of learning parameters is discussed. Before the time of learning parameters is presented, we introduce the parameters which are needed to be tuned. In SVM+ and L2-SVM+, there are two parameters C and γ. In R-SVM+, there are three parameters C, γ, and λ.[1] In R-CTSVM+, let $C_1 = C_2$ and $\epsilon_1 = \epsilon_2$, and thus there are three parameters C_1, ϵ_1 and λ. In S-ISCN+, there are three parameters γ, η and μ. In PLWB, there are two parameters C and γ.[2] The details of parameter setting can refer to Section 4.2.2. Fig. 8 shows the parameter learning time of PLWB and the baselines. We have the following observations from Fig. 8. Firstly, L2-SVM+ has the least parameter learning time. It is a least-squares-based method which does not need to solve the inequality constraints, but equality constraints instead. Therefore, L2-SVM+ is the fastest method. It is noted that the proposed PLWB method can be modified into the least squares form which can greatly improve the efficiency of our method. Secondly, PLWB takes more time than SVM+ and R-SVM+. This may be because SVM+ and R-SVM+ do not take the weak label problem into account and directly utilize the initial labels to build the privileged information learning classifier. Different from these methods, we consider the weak label problem and employ a heuristic framework to update the weights of labelers, such that the classification accuracy can be improved by optimizing the instance labels. Thirdly, R-CTSVM+ has a higher parameter

[1] In R-SVM+, the parameter σ should be set to be a relatively large value, as suggested in [1]. Thus, σ is fixed to be 1000 and does not need to be tuned.

[2] In PLWB, the parameter θ is fixed to be 1000, as done in R-SVM+. Moreover, the parameter ϵ controls the termination of our algorithm, which should be set to be a small value. Thus, ϵ is fixed to be 0.1 and does not need to be tuned.

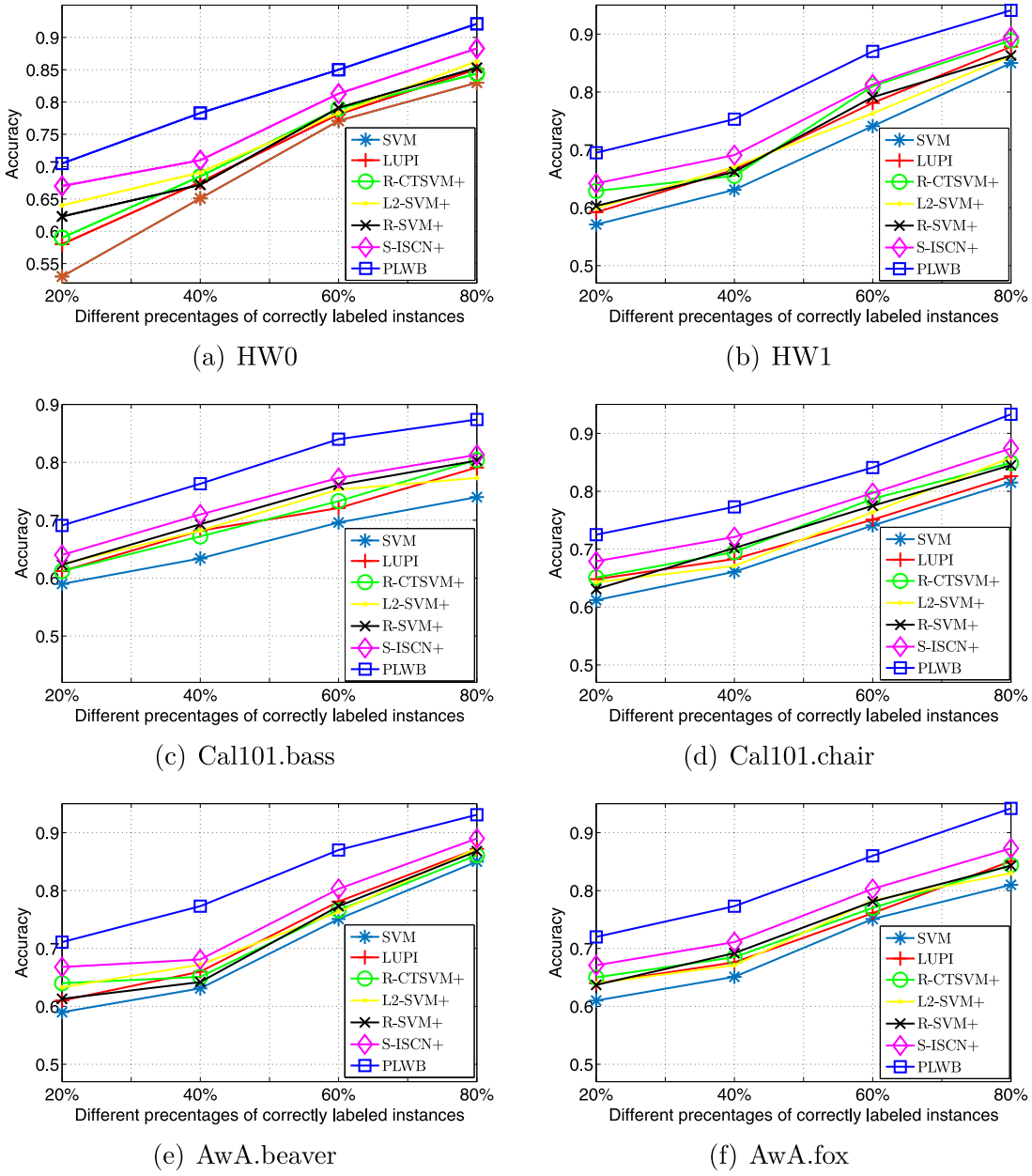

Fig. 6. Performance with different percentages of correctly labeled instances.

learning time than the other SVM-based methods (i.e., L2-SVM+, SVM+, R-SVM+ and PLWB). It is an iterative learning method and has three parameters to be tuned, while the other SVM-based methods have only two parameters. Lastly, S-ISCN+ has the highest parameter learning time among all the methods. It needs to tune there parameters and is a network-based method. Thus, it has the highest parameter learning time.

Moreover, the time of building models is investigated. The average time of building models is shown in Fig. 9. On the one hand, PLWB is relatively slower than L2-SVM+, SVM+ and R-SVM+. This is because PLWB takes the weak label problem into account, and extra time is needed to optimize the labeler weights. Different from PLWB, L2-SVM+, SVM+ and R-SVM+ do not consider the weak label problem. Although they have a faster model building time, their classification accuracy may be limited. For example, on the AwA.wolf sub-dataset, the accuracy of PLWB is 90.58, while that of L2-SVM+, SVM+ and R-SVM+ is 86.18, 83.27 and 83.38, respectively. PLWB has higher classification accuracy than L2-SVM+, SVM+ and R-SVM+ at 4.4, 7.31 and 7.2, respectively. On the other hand, PLWB is faster than R-CTSVM+. R-CTSVM+ adopts an iterative framework and involves the matrix inversion which makes it have higher model building time than PLWB.

Lastly, the time of predicting new instances is studied. Fig. 10 shows the average time of predicting new instances. It can be seen that SVM+, R-SVM+ and PLWB have similar predicting time. They learn a hyper-plane to predict new instances and only a small part of the training instances (i.e., support vectors) are included in predicting instances. Thus, the time of predicting new instances is relatively low. Moreover, the time of L2-SVM+ and R-CTSVM+ is higher than SVM+, R-SVM+ and PLWB. This is because L2-SVM+ is a least-squares-based method and all the training instances are included in predicting new instances. Hence, it has relatively high predicting time. R-CTSVM+ is a twin-SVM-based method which learns two nonparallel hyper-planes.

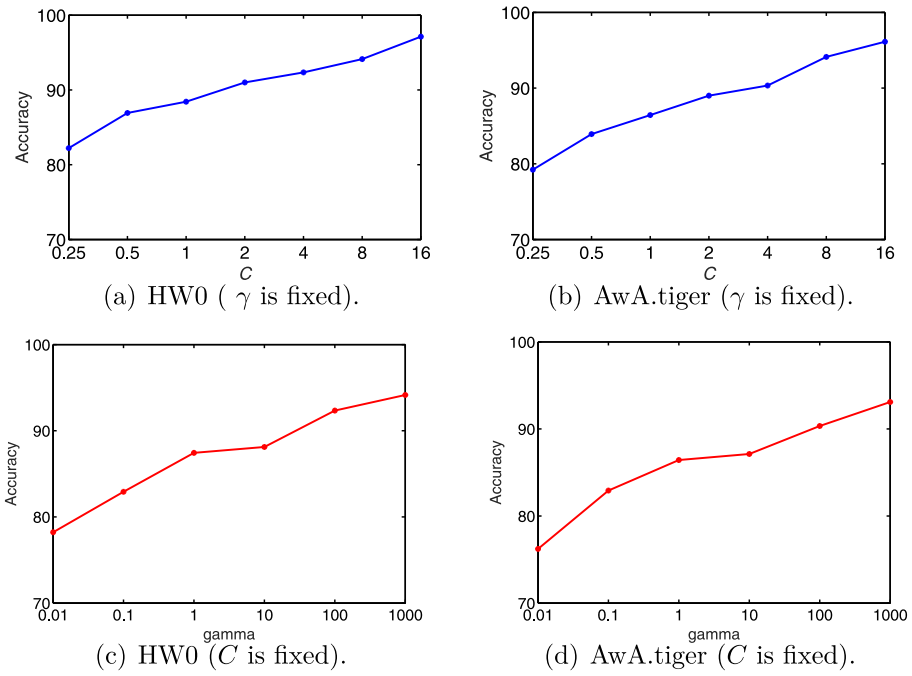

Fig. 7. Performance with different parameter values.

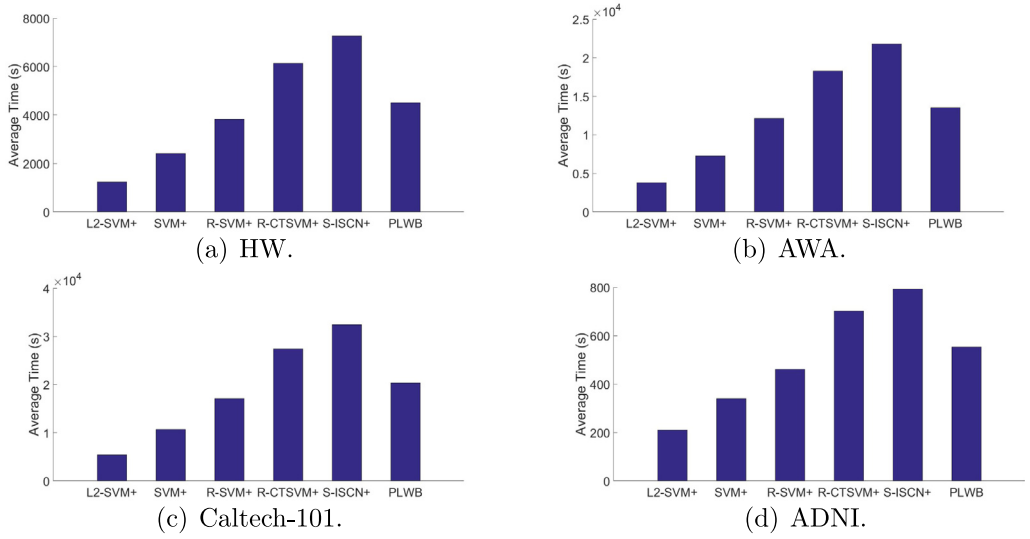

Fig. 8. Time of selecting the optimal parameters.

Thus, each new instance should go through two hyper-planes to obtain the final label and the predicting time is relatively high. At last, ISCN+ has the highest time to predict new instances since it is a network-based method and each new instance should go through a number of hidden nodes to obtain the label. Hence, ISCN+ has higher predicting time than the SVM-based methods, i.e., SVM+, R-SVM+, L2-SVM+, R-CTSVM+ and PLWB.

5. Conclusions and future work

5.1. Conclusions

The existing privileged information learning methods assume that the instances are accurately labeled. They do not consider the weak label problem. Different from these methods, in this paper, we put forward a novel privileged learning method with weak labels. To the best of our knowledge, this is the first attempt to deal with the weak labels in privileged information learning problems. Our method is verified on real-world datasets, including Handwritten (HW) categorization, Animals-with-Attributes (AWA), Caltech-101, and Alzheimer's Disease Neuroimaging Initiative (ADNI) datasets. The experimental results demonstrate that by considering the weak label problem and optimizing the labeler weights, our method can get superior classification performance in comparison with the existing privileged information learning algorithms.

5.2. Limitations

The limitation of our method is that we have relatively higher training time than L2-SVM+, SVM+ and R-SVM+. This is because we consider the weak label problem and need extra time to optimize the labeler weights. Moreover, our training time is still lower than R-CTSVM+ and S-ISCN+.

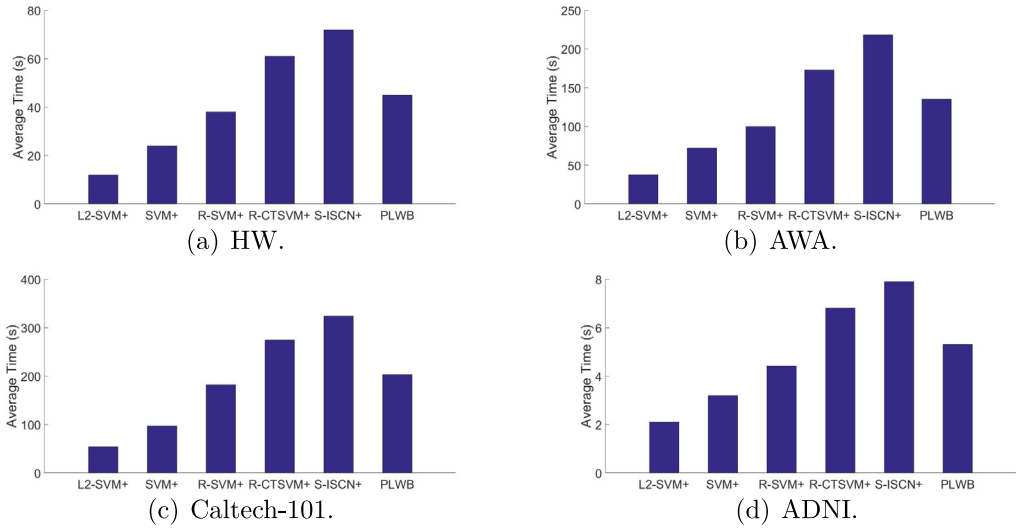

Fig. 9. Time of building models.

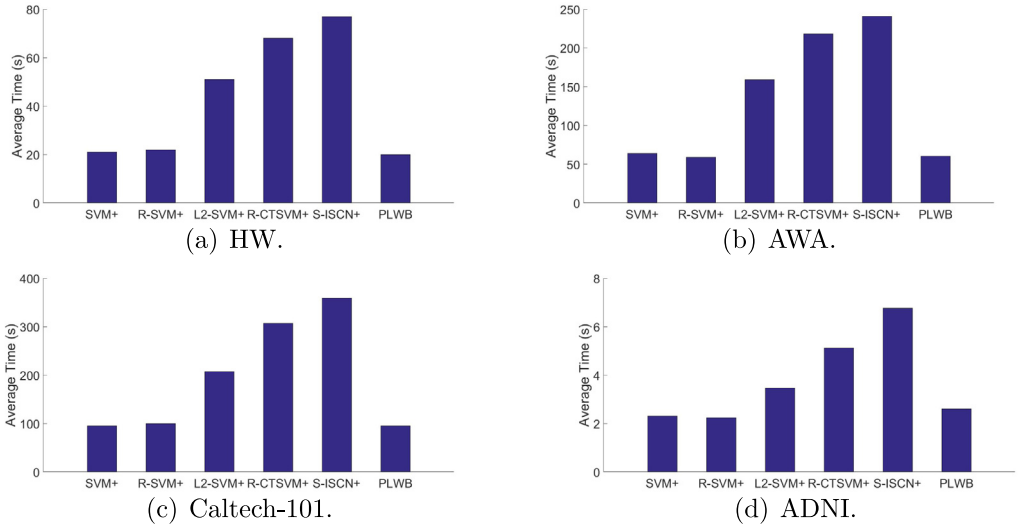

Fig. 10. Time of predicting new instances.

5.3. Future work

In the future, firstly, we would like to extend our method to an on-line style and handle the dynamic data, such that the learning cost and performance can have a better tradeoff. Secondly, correlating our method with the current technologies, such as communications, networks and Cloud, is also a valuable consideration for our future work. Thirdly, we will adapt our method to data aggregation, transfer learning and domain adaptation, and apply it on the data related to explainable AI and disease. Lastly, we would like to apply our method to specific application areas, such as investment risk evaluation, electricity consumption forecasting and so on, and discuss the related policy implications combined with these application areas.

CRediT authorship contribution statement

Yanshan Xiao: Conceptualization, Methodology, Formal analysis, Visualization, Writing – review & editing. **Zexin Ye:** Data curation, Software, Validation, Writing. **Liang Zhao:** Conceptualization, Writing – review & editing. **Xiangjun Kong:** Data curation, Software, Writing – review & editing. **Bo Liu:** Conceptualization, Methodology, Writing – review & editing. **Kemal Polat:** Supervision, Project administration, Writing – review & editing. **Adi Alhudhaif:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the reviewers for their very useful comments and suggestions. This work was supported in part by the Natural Science Foundation of China under Grant 62076074, in part by Guangdong Natural Science Foundation under Grant 2023A1515012560.

References

[1] V. Vapnik, A. Vashist, A new learning paradigm: learning using privileged information, Neural Netw. 22 (5/6) (2009) 544–557.
[2] S. Wang, S. Chen, T. Chen, X. Shi, Learning with privileged information for multi-label classification, Pattern Recognit. 81 (2018) 60–70.
[3] H. Yang, J.T. Zhou, J. Cai, Y.S. Ong, MIML-FCN+: Multi-instance multi-label learning via fully convolutional networks with privileged information, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 5996–6004.
[4] Y. Shan, X. Chang, W. Yunhe, X. Chao, T. Dacheng, Privileged multi-label learning, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 3336–3342.
[5] D. Pechyony, V. Vapnik, Fast optimization algorithms for solving svm+, in: Statistical Learning and Data Science, Chapman and Hall, Boca Raton, FL, 2011, pp. 2258–2266.
[6] W. Li, D. Dai, M. Tan, D. Xu, L. Van Gool, Fast algorithms for linear and kernel SVM+, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016.
[7] X. Li, B. Du, Y. Zhang, C. Xu, D. Tao, Iterative privileged learning, IEEE Trans. Neural Netw. Learn. Syst. 31 (8) (2020) 2805–2817.
[8] J. Tang, Y. He, Y. Tian, D. Liu, G. Kou, F.E. Alsaadi, Coupling loss and self-used privileged information guided multi-view transfer learning, Inform. Sci. 551 (2021) 245–269.
[9] H. Sun, W. Zhai, Y. Wang, L. Yin, F. Zhou, Privileged information-driven random network based non-iterative integration model for building energy consumption prediction, Appl. Soft Comput. 108 (2021) 107438.
[10] X. Zhou, Y. Ao, X. Wang, X. Guo, W. Dai, Learning with privileged information for short-term photovoltaic power forecasting using stochastic configuration network, Inform. Sci. 619 (2023) 834–848.
[11] S. Sun, M. Li, S. Wang, C. Zhang, Multi-step ahead tourism demand forecasting: The perspective of the learning using privileged information paradigm, Expert Syst. Appl. 210 (2022) 118502.
[12] Y. Shu, Q. Li, C. Xu, S. Liu, G. Xu, V-SVR+: Support vector regression with variational privileged information, IEEE Trans. Multimed. 24 (2022) 876–889.
[13] R. Xu, H. Wang, Multi-view learning with privileged weighted twin support vector machine, Expert Syst. Appl. 206 (2022) 117787.
[14] Y. Li, H. Sun, W. Yan, Domain adaptive twin support vector machine learning using privileged information, Neurocomputing 469 (2022) 13–27.
[15] Y. Li, H. Sun, W. Yan, Domain adaptive twin support vector machine learning using privileged information, Neurocomputing 469 (2022) 13–27.
[16] Y. Shu, Q. Li, L. Liu, G. Xu, Privileged multi-task learning for attribute-aware aesthetic assessment, Pattern Recognit. 132 (2022) 108921.
[17] E. Sabeti, J. Drews, N. Reamaroon, E. Warner, M.W. Sjoding, J. Gryak, K. Najarian, Learning using partially available privileged information and label uncertainty: Application in detection of acute respiratory distress syndrome, IEEE J. Biomed. Health Inf. 25 (3) (2021) 784–796.
[18] V. Sharmanska, N. Quadrianto, C.H. Lampert, Learning to rank using privileged information, in: 2013 IEEE International Conference on Computer Vision, 2013, pp. 825–832.
[19] L. Niu, J. Wu, Nonlinear L-1 support vector machines for learning using privileged information, in: 2012 IEEE 12th International Conference on Data Mining Workshops, 2012, pp. 495–499.
[20] N. Sarafianos, M. Vrigkas, I.A. Kakadiaris, Adaptive SVM+: Learning with privileged information for domain adaptation, in: 2017 IEEE International Conference on Computer Vision Workshops, ICCVW, 2017, pp. 2637–2644.
[21] M. Lapin, M. Hein, B. Schiele, Learning using privileged information: SVM+ and weighted SVM, Neural Netw. 53 (2014) 95–108.
[22] W. Sultani, M. Shah, Automatic action annotation in weakly labeled videos, Comput. Vis. Image Underst.: CVIU 161 (2017) 77–86.
[23] X. Liu, L. Sun, S. Feng, Incomplete multi-view partial multi-label learning, Appl. Intell. (52–3) (2022).
[24] I. Choi, S.H. Bae, S.J. Cheon, W.I. Cho, N.S. Kim, Weakly labeled acoustic event detection using local detector and global classifier, in: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017, pp. 1735–1738.
[25] S.-J. Yang, Y. Jiang, Z.-H. Zhou, Multi-instance multi-label learning with weak label, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, 2013, pp. 1862–1868.
[26] H. Dong, Y. Li, Z. Zhou, Learning from semi-supervised weak label data, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 2926–2933.
[27] Q. Wang, L. Yang, Y. Li, Learning from weak-label data: A deep forest expedition, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, pp. 6251–6258.
[28] Q. Tan, G. Yu, C. Domeniconi, J. Wang, Z. Zhang, Multi-view weak-label learning based on matrix completion, in: International Conference on Data Mining, 2018, pp. 450–458.
[29] C. Jixu, L. Xiaoming, Transfer learning with one-class data, Pattern Recognit. Lett. 37 (Feb.1) (2014) 32–40.
[30] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[31] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories, in: 2004 Conference on Computer Vision and Pattern Recognition Workshop, 2004, p. 178.
[32] C.H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 951–958.
[33] Y. Xiao, F. Liang, B. Liu, A transfer learning-based multi-instance learning method with weak labels, IEEE Trans. Cybern. 52 (1) (2022) 287–300.
[34] Y. Li, H. Sun, W. Yan, Q. Cui, R-CTSVM+: robust capped l1-norm twin support vector machine with privileged information, Inf. Sci. 574 (2021) 12–32.
[35] J. Lu, J. Ding, A novel stochastic configuration network with iterative learning using privileged information and its application, Inform. Sci. 613 (2022) 953–965.
[36] X. Li, B. Du, C. Xu, Y. Zhang, L. Zhang, D. Tao, R-SVM+: robust learning with privileged information, in: International Joint Conference on Artificial Intelligence, 2018, pp. 2411–2417.
[37] Y. Chen, J.Z. Wang, Image categorization by learning and reasoning with regions, J. Mach. Learn. Res. 5 (2004) 913–939.