A ALGORITHM DETAILS

Algorithm 1 Detailed algorithm of PLRL

Input: the partial label training set $\mathcal{D}$, $\mathbf{F}^{(\mathbf{0})} = \mathbf{P}$ as given in Eq.(3), parameters $\alpha, T, \mu, \lambda, \eta, \beta$.
Output: predictor g.

for epoch = 1,2,... **do**
  Get the weight matrix $\mathbf{W}$ by Eq.(1) from pretrained neural network
  for $t = 1$ to T **do**
    Set $\tilde{\mathbf{F}}^{(t)}$ by Eq.(4).
    Normalize $\tilde{\mathbf{F}}^{(t)}$ by row into $\mathbf{F}^{(t)}$ as given in Eq.(5).
  end for
  Return $\mathbf{F} = \mathbf{M} \circ \tilde{\mathbf{F}}^{(T)}$ as given in Eq.(6).
  Update prediction model $g(\boldsymbol{x})$ by Eq.(9).
  Calculate the negative reward by Eq.(2).
  Update parameters θ as given in Eq.(10).
end for
Predict y^* for unseen instance $\boldsymbol{x}^*$ by (11).

B COMPARING METHODS

Graph-based methods:

- AGGD (Wang et al., 2019): the disambiguation is done by alternative optimization, and the prediction is based on kernel regression. [Configuration: $k = 10, T = 10, \lambda = 1, \mu = 1, \gamma = 0.05$];
- IPAL(Zhang & Yu, 2015): the similarity graph $\mathbf{W}$ is built with k-NN, and the unseen instances are predicted based on the minimum error reconstructed by its nearest neighbors. [Configuration: $\alpha = 0.95, k = 10, T = 100$];
- PL-KNN (Hüllermeier & Beringer, 2006): predict by averaging the labels of neighbors obtained using k-NN. [configuration: $k = 10$].
- PLRL: Our proposed algorithm. [Configuration: $\alpha = 0.95, T = 100, \lambda = 0.05, \mu = 1, \eta = 0.5, \beta = 0.05$].

Other methods:

- SURE (Feng & An, 2019a): the self-guided retraining method to balance the minimum approximation loss and the maximum infinity norm of the outputs. [Configuration: $\lambda = 0.5, \beta = 0.05$];
- LSB-CMM (Liu & Dietterich, 2012): the maximum likelihood approach to conduct identification-based disambiguation. [Configuration: $\sigma^2 = 1, K = 80, \alpha = 0.05$];
- CLPL (Cour et al., 2011): use SVM with the squared hinge loss to transfer the PLL problems to binary learning problems by feature mapping.
- PL-SVM (Nguyen & Caruana, 2008): optimize the margin-based objective function. [Configuration: $\lambda = 0.01$];

C DATASETS DESCRIPTION

C.1 SIMULATION PARAMETER

The simulation data features are generated from several distributions: Gaussian, uniform, and binomial. The detailed generation rules are shown in Table 7.

Table 7: Simulation data feature generation rules

feature dimension	1	2	3	4	5	6	sample size
class i	$N(a_i, 5)$	$N(b_i, 7)$	$U[c_i, d_i]$	$U[e_i, f_i]$	$B(25, p_i)$	$B(35, p_i)$	n_i

$a_i \sim U[0, 10],\ b_i \sim U[10, 30],\ c_i, e_i \sim U[10, 20],\ d_i, f_i \sim U[20, 40],$
$p_i \sim U(0, 1),\ \sum_i n_i = n,\ i = 1, ..., q.$

C.2 UCI AND REAL-WORLD DATASETS

Characteristics of the controlled UCI datasets and real-world datasets are summerized in Table 8 and Table 9 respectively.

Table 8: Characteristics of the controlled UCI datasets

Data set	#Examples	#Features	#Class Labels
vehicle	846	18	4
sensor	5456	24	4
dermatology	366	33	6
steel	1941	27	7
segment	2310	18	7
Satimage	6435	36	7
wine	1599	11	10
abalone	4177	7	29

Table 9: Characteristics of the real-world data sets

Data set	#Examples	#Features	#Class Labels	Avg. #CLs
Lost	1122	108	16	2.23
MSRCv2	1758	48	23	3.16
BirdSong	4998	38	13	2.18
Soccer Player	17,472	279	171	2.09
Yahoo! News	22,991	163	219	1.91

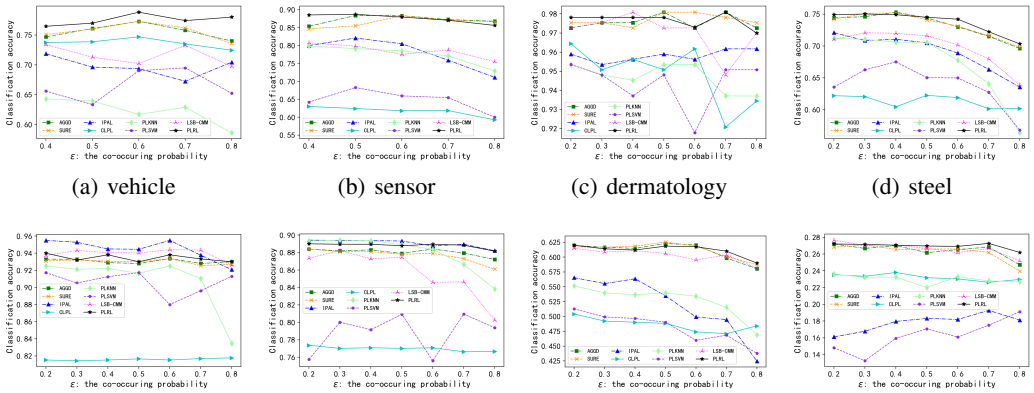
D CONTROLLED EXPERIMENTS

In Figure 3, those considered algorithms are compared with respect to their classification accuracy as ϵ ranges from 0.2 to 0.8 (from 0.4 to 0.8 for the first two datasets) when $p = 1$ and $r = 1$ (Configuration (I)). In this setting, a specific label is selected as the coupled label that co-occurs with the ground-truth label with probability ϵ, and any other label would be randomly chosen to co-occur with y with equal probabilities. The prediction accuracy is summarized in Figure 4 for each algorithm as r varies with its range increasing with the total number of categories (Configuration (II)). As shown in Figures 3 and 4, PLRL performs consistently well in most scenarios, especially when r and ϵ are large, that is, when the degree of confusion is high.

E FURTHER ANALYSIS

E.1 PARAMETER SENSITIVITY

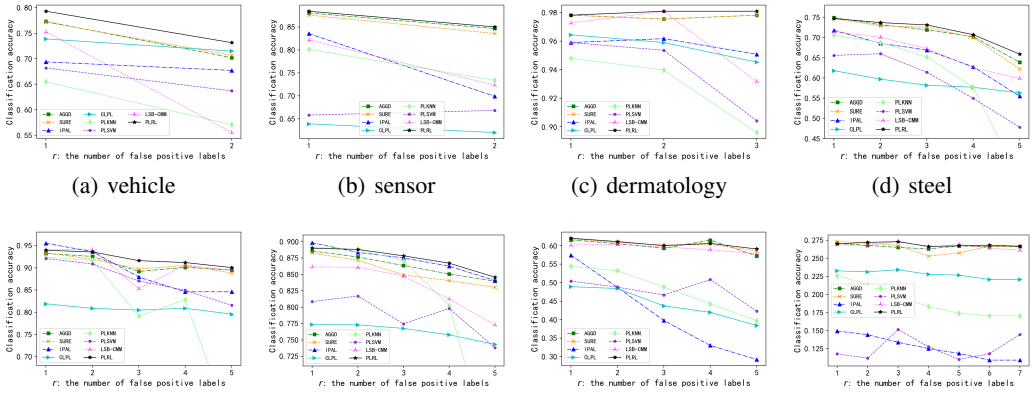
We conduct a sensitivity analysis for PLRL by changing three important regularization coefficients, μ, η and β in the reward function. We fix the other hyperparameters as $\alpha = 0.95, T = 100, \lambda = 0.05$ following the same settings in Zhang & Yu (2015); Feng & An (2019a). Figure 5 visualizes the performance of PLRL under different parameter configurations on two selected datasets, Lost and BirdSong. According to the experimental results, we suggest that the practical choices of three regularization coefficients are $\mu = 1, \eta = 0.5, \beta = 0.05$.



(a) vehicle (b) sensor (c) dermatology (d) steel

(e) segment (f) Satimage (g) wine (h) abalone

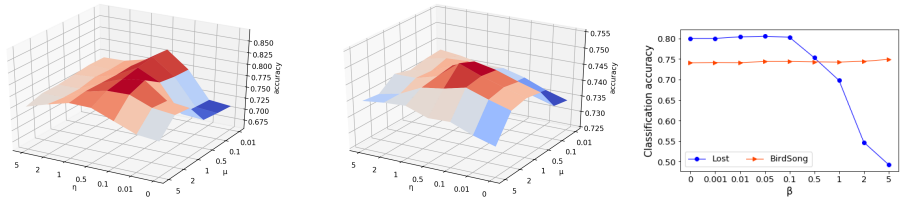
Figure 3: Classification performance on controlled UCI datasets with ϵ (co-occurring probability of the coupling label) and one false positive candidate label ($r = 1$)



(a) vehicle (b) sensor (c) dermatology (d) steel

(e) segment (f) Satimage (g) wine (h) abalone

Figure 4: Classification performance on controlled UCI datasets with r (the number of false positive candidate label).



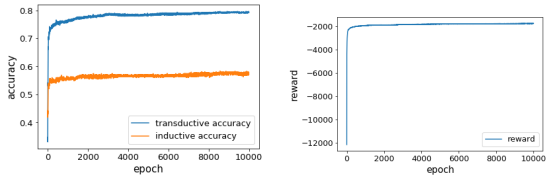
(a) Varying μ and η on Lost (b) Varying μ and η on BirdSong (c) Varying β

Figure 5: Parameter sensitivity analysis for PLRL in terms of classification accuracy on Lost and BirdSong by varying μ, η and β.

E.2 Algorithm Convergence

Figures 6(a) and 6(b) give the training curves of the prediction accuracy and the reward score by PLRL on MSRCv2, respectively. We can see that the reward converges as the number of training epochs increases, and eventually falls into a small range after 1000 epochs. On the other hand, the accuracy curves for both the training and test sets have the same trends as the reward curve and jointly

reach a high level in the end. The convergence analysis shows that the RL agent gradually learns to model the instance similarities and improves its prediction power.



(a) Classification accuracy on MSRCv2

(b) Reward on MSRCv2

Figure 6: Convergence analysis for PLRL. (a) Convergence curves of transductive and inductive classification accuracy on MSRCv2. (b) Convergence curves of reward on MSRCv2.

F Related work

In PLL, the true labels of most training instances are unobserved while a small set of candidate labels are provided instead. One fundamental problem PLL tries to solve is how to utilize the limited number of ground-truth labels to disambiguate the partially labelled instances, i.e. selecting the most possible one from a set of candidates. Extensive efforts have been made to develop label disambiguation strategies, which can be divided into three main categories. The most naive way is to fairly treat all candidate labels and obtain the final prediction by simply taking the average of the predicted probabilities over the candidate classes.

Candidate labels are disambiguated by averaging the output probabilities using k-NN method in Gong et al. (2017). Boosting learning adopted by Tang & Zhang (2017) improves the classifier by adapting the weights of training instances and the ground-truth confidence of candidate labels. The averaged modeling output of the candidate labels is distinguished from that of the non-candidates by Cour et al. (2011). Some non-parametric methods, such as Zhang & Yu (2015), make predictions for the testing examples by voting among all candidate labels of its neighboring instances. The main disadvantage of these voting methods is that the ground truth can be overwhelmed by other confounded incorrect labels, which leads to negative impact on the disambiguation performance.

The second category of disambiguation methods is identification-based. Traditional machine learning models are directly trained to build the classifiers (Jin & Ghahramani, 2002; Nguyen & Caruana, 2008; Liu & Dietterich, 2012; Zhou et al., 2016). Latent semantic differences are maximized by Feng & An (2019b) between any two instances whose unknown true labels are ensured to be different. SURE (Feng & An, 2019a) uses a self-guided retraining method to balance the minimum approximation loss and the maximum infinity norm of the outputs. The main drawback of the identification-based methods is that the identified label might turn out to be false-positive due to the model limits.

Different from the first two categories of methods that focus on the label disambiguation, the third category of methods is graph-based. Some recent studies try to utilize the instance similarities by discovering some underlying connective graph using some unsupervised approaches, such as PL-KNN (Hüllermeier & Beringer, 2006) and IPAL (Zhang & Yu, 2015). Other methods like LALO (Feng & An, 2018) adopts a feature-aware approach which facilitates the mutual adaption of the model training and the constrained label propagation by simultaneously estimating the latent label distributions at the training stage. PL-LEAF (Zhang et al., 2016) makes use of the local manifold structure in feature space to help disambiguate the candidate label set. AGGD (Wang et al., 2019) considers using an adaptive graph, but the complex optimization proposed by them is difficult to solve in practice. The key problem of the graph-based methods is that the learned instance similarity is weakly related to the classification task since the graph structure is independently estimated without well using the instance features and the label information. Moreover, the voting strategy they use is usually empirically non-robust. Our PLRL algorithm addresses these issues by using an end-to-end model, which makes the learned underlying connective graph more related to the classification task and its empirical performance is significantly better than all the existing two-stage and alternative graph-based methods.