ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240737

Partial Label Learning via Cost-Guided Retraining

Zhaoyuan Zhang[a], **Zhenbing Liu**[a] **and Haoxiang Lu**[a,*]

[a]Computer and Information Security, Guilin University of Electronic Technology, Guilin, China
ORCID (Haoxiang Lu): https://orcid.org/0000-0003-2284-5154

Abstract. In partial label learning, each training sample corresponds to a set of candidate labels. The ground-truth label, hidden within this set, cannot be directly obtained during the training phase. The key to solving the partial label learning problem is to obtain ground-truth labels through label disambiguation. Existing works often rely on the label averaging assumption and do not fully investigate the class imbalance. Tail ground-truth labels are often overwhelmed by head pseudo-labels. The incorrectly identified labels could have contagiously negative impacts on the final predictions. In this paper, we propose a cost-guided retraining strategy, which achieves guidance and correction of disambiguation results, and provides instance-based class imbalance concerns for candidate labels. This approach significantly enhances the algorithm's ability to handle class imbalance problems. The superiority of our method is demonstrated using 8 real-world datasets and 5 evaluation metrics. Code is available at https://github.com/DerrickZzyR/PL-CGR

1 Introduction

In supervised learning, each training sample is associated with an exact label. In real-world tasks, the high-quality labels are expensive and time-consuming. To solve the problem, **P**artial **L**abel **L**earning (PLL) [42, 29] has been proposed. In PLL, an instance is associated with a set of candidate labels, among which only one is the ground-truth label. Specifically, $\mathcal{X} = \mathbb{R}^d$ is defined as the feature space with dimension d, and $\mathcal{Y} = \{0, 1\}^q$ represents the label space with q labels. The PLL dataset is defined as $\mathcal{D} = \{(x_i, Y_i) \,|1 \leq i \leq m\}$, where $x_i \in \mathcal{X}$ is the i-th instance, $Y_i \subseteq \mathcal{Y}$ is the corresponding set of candidate labels and m is the number of training instances. The PLL aims to learn a multi-class classifier $f : \mathcal{X} \to \mathcal{Y}$ that accurately identifies ground-truth labels based on the data set $\mathcal{D}$.

Label disambiguation, a crucial method for addressing PLL, mainly includes average disambiguation strategy and identification disambiguation strategy. Existing algorithms assume the same number of different labels. However real-world data often suffers from class imbalance. Self-guided retraining has achieved good results as the identification disambiguation strategy. As shown in Figure 1 (a), SURE [8] is based on label mutual exclusion and achieves label disambiguation by continuously increasing the algorithm's attention to potentially ground-truth labels. However, due to class imbalance, head labels naturally receive more attention, even if these labels may be incorrect. When the tail samples are similar to the head samples and have the same common candidate labels, the tail samples are easily misclassified. As shown in Figure 1 (b) and (c). The proportion of head samples predicted by SURE is higher than real and does not perform as well on the tail labels.

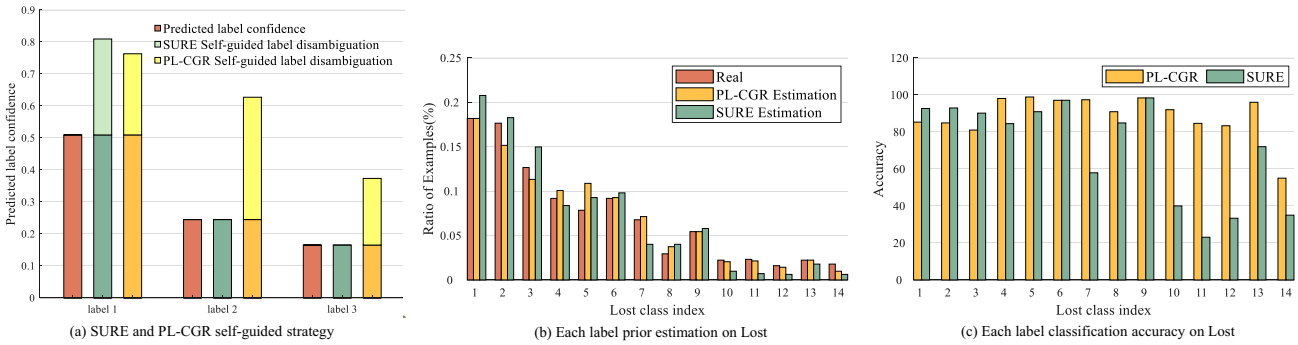
The issue of multi-class imbalance [39, 32] has been extensively explored. In the data-level, existing approaches employ under-/over-sampling techniques [6] to modify the **I**mbalance **R**atio (IR) [16]. At the algorithmic-level, cost sensitivity [33, 2] is embedded into the classification model. By minimum cost loss, this approach effectively reduces label misclassification and improves algorithm performance. Existing approaches to multi-class imbalance problems rely on accurate label information. It makes them unsuitable for PLL.

To address the aforementioned challenges, a novel approach named PL-CGR, i.e. **P**artial **L**abel Learning via **C**ost-**G**uided **R**etraining is proposed. Specifically, we construct prototype samples by intra- and inter-class scatter, which serve as label prior information for establishing label thresholds and label attention mechanisms. Label thresholds are used to guide and correct label disambiguation. Additionally, label thresholds and label attention mechanisms achieve instance-based class imbalance attention. This approach aims to mitigate the negative impact of class imbalance leading to incorrect label identification on the final prediction. Our main contributions are delineated across three domains: 1) We propose a label disambiguation correction based on label thresholds to reduce situations where the ground-truth label is overwhelmed by the head label; 2) We improve the performance of the algorithm in solving the class imbalance problem by utilizing label thresholds and label attention mechanisms to provide instance-based class imbalance attention; 3) We propose a cost-sensitive strategy for PLL that aims to solve the class imbalance problems in self-guided retraining strategies. Extensive experiments validate the effectiveness of the strategy.

2 Related work

Partial label learning is one of the important weakly supervised learning frameworks [28, 34]. In PLL, each instance is associated with a set of candidate labels, among which only one is valid. PLL is already a challenge because the algorithms do not have direct access to ground-truth label during the training phase. To overcome this, the key approach is label disambiguation, which includes average-based disambiguation [11, 4], identification-based disambiguation [30, 24, 23]. In average-based disambiguation, each candidate label is treated equally by the algorithm, and model predictions are generated by averaging the outcomes. This strategy is intuitive and straightforward, but pseudo-labels easily overshadow the ground-truth label. In Identification-based disambiguation, the ground-truth label, considered as the latent variable, is identified by iterative optimization.

* Corresponding Author. Email: hxlu1005@163.com.



(a) SURE and PL-CGR self-guided strategy (b) Each label prior estimation on Lost (c) Each label classification accuracy on Lost

Figure 1. (a) Differences between SURE and PL-CGR self-guided strategies. PL-CGR gives different attention to candidate labels rather than focusing only on the label with the highest confidence. Labels 1, 2, and 3 correspond to the head, mid-quantity, and tail labels, respectively. (b) The real/estimated label distribution of the Lost dataset. (c) Comparison of the performance of SURE and PL-CGR on different labels of the Lost dataset.

The multi-class imbalance problem has been well studied, and some methods have been applied to address class imbalance in PLL. Wang et al. proposed three resampling approaches to mitigate the effects of the class imbalance problem [32]. The method mitigates the effects of class imbalance at the data level, but it does not deal well with extreme class imbalance (IR $\geq$ 50). Wang et al. proposed Solar [31] to improve the performance of the algorithm on tail labels by constraining the head labels. However, over-ignoring head labels may lead to degradation of algorithm performance. Cost-sensitive learning is one of the important algorithms for solving class imbalance by minimizing the cost of classification errors rather than minimizing the number of errors [27]. Cost-sensitive support vector machines introduce cost-sensitive coefficients to improve the performance of the algorithm on tail label samples [9]. Zhou et al. embedded the cost of misclassification into the logistic regression model and proposed a cost-sensitive logistic regression [43] for multi-classification, which was successfully applied to the field of face recognition.

In this paper, we employ cost sensitivity to solve class imbalance, which is different from traditional cost-sensitive methods. In PLL, accurate label information cannot be obtained, and the misclassification cannot be accurately known during the training phase. Additionally, the instance is considered as the positive sample for candidate labels. To the best of our knowledge, our work is the first to employ a cost-sensitive strategy in addressing class imbalance within a self-guided retraining framework.

3 The proposed approach

Following the notations in the introduction, $X = [x_1, x_2, \cdots, x_m]^\top \in \mathbb{R}^{m\times d}$ is denoted as the instance matrix and $Y = [y_1, y_2, \cdots, y_m]^\top \in \{0,1\}^{m\times q}$ is denoted as the partial label matrix, where $y_{ij} = 1$ indicates that the j-th label is a candidate label for the instance x_i, while $y_{ij} = 0$ implies the j-th label is a non-candidate label. $P \in \mathbb{R}^{m\times d}$ is the prediction label confidence matrix. cs and $\mathcal{A}$ are the cost-sensitive coefficients and label attention coefficients.

3.1 Prototype-based cost coefficient construction

Existing methods for constructing cost coefficients rely on accurate label information. However, the ground-truth label is hidden in the candidate labels, which makes it more challenging. Inspired by [5, 40, 44], firstly each class is roughly divided into positive set $\mathcal{P}$ and negative set $\mathcal{N}$. For the Y_j, the positive and negative sample sets are divided as follows:

$$\begin{aligned}\mathcal{P}_j &= \left\{x_i \middle| (x_i, p_j) \in \mathcal{D}, p_j \geq \overline{y}_j\right\} \\ \mathcal{N}_j &= \left\{x_i \middle| (x_i, p_j) \in \mathcal{D}, p_j < \overline{y}_j\right\}.\end{aligned} \tag{1}$$

where $\overline{y}_j$ is the average confidence threshold for Y_j, i.e. $\overline{y}_j = 1/|Y_{\cdot j}| \times \sum_{i=1}^{m} p_{ij}$. The head-label instances are typically richer in feature information and therefore tend to be associated with higher label confidence. In contrast, tail-label instances generally have lower label confidence. The uniform confidence threshold can incorrectly classify tail-label positive instances as negative. This approach helps to generate prototype instances more fairly and effectively. Based on the set of positive instances, we propose a prototype instances generation method that adapts to class imbalance. Specifically, we divided each label positive sample set into multiple subgroups. The number of subgroups for the Y_j is shown below:

$$k_j = \lceil r \times |\mathcal{P}_j| \rceil, (1 \leq j \leq q). \tag{2}$$

where r is a hyperparameter controlling the number of subgroups [5]. Since outlier instances are not representative, we construct prototype instances using k-means clustering [38, 21] and the cluster centers are denoted as prototype instances $\mathcal{I} = \{I_1, I_2, \cdots, I_q\}$, where $I_j = \left(i_j^1, i_j^2, \cdots, i_j^{k_j}\right) \in \mathbb{R}^{k_j, d} (1 \leq j \leq q)$ is denoted the set of prototype instances for Y_j. Clustering, as an unsupervised learning algorithm, is commonly used to discover natural groups. This method has the following three advantages: 1) The prototype instances inherit the attributes of the sample space distribution; 2) Multiple prototype instances can ensure the reliability of positive instances, while more accurately reflecting the data distribution and facilitating the construction of reliable cost coefficients; 3) Prototype instances are considered as the prior knowledge, which can reduce the algorithm's dependence on a large number of precise labels. The cost of x_i being identified as Y_j is the mean Euclidean distance between x_i and $\mathcal{I}_j$. It can be expressed as follows:

$$cs_{ij} = \begin{cases} \dfrac{\sum_{n=1}^{k_j} \sum_{x\in\mathcal{P}_j} \left\|x_i - i_j^n\right\|_2^2}{|\mathcal{P}_j|}, (i \neq j, 1 \leq i, j \leq q) \\ 0, \qquad\qquad\qquad\qquad (i = j, 1 \leq i, j \leq q), \end{cases} \tag{3}$$

The higher the cs value, the lower the probability that x_i belongs to Y_j. Eq. (3) aims to construct cost coefficients that combine intra-class scatter and inter-class scatter. To unify the cost coefficients with label confidence sizes and address empty labels in real-world tasks, we introduce the slack variable ε. Inspired by the max-min normalization, the cost-sensitive coefficients are normalized and are expressed as follows:

$$cs_{ij} = \frac{cs_{ij} + \varepsilon}{\max\left(cs_{i\cdot} + \varepsilon * q\right).} \tag{4}$$

In response to the inability to refine the misclassification, inspired by [35], we transform q classification into q binary classification. The model predictions are used to determine whether the sample is positive. The specific transformation is shown in Figure 2 (a). The j-th row of the cost matrix represents the cost c_{01} of the instance of Y_j being identified as another label, also known as a false negative (FN). The j-th column of the cost matrix represents the cost c_{10} of other label instances being identified as Y_j, also called false positive (FP). The cost matrix for the Y_j is as follows:

Table 1. Cost matrix corresponding to Y_j in partial label learning.

	Actual positive	Actual negative
Predicted positive	$c^j_{11} = 0$	$c^j_{01} = \sum_{j=1}^{q} cs_{\cdot j}$
Predicted negative	$c^j_{10} = \sum_{i=1}^{q} cs_{i\cdot}$	$c^j_{00} = 0$

The instance x_i will be identified as a positive instance of Y_j only when the following conditions are satisfied [15]:

$$p\left(Y_j^+|x_i\right)\cdot c^j_{11} + p\left(Y_j^-|x_i\right)\cdot c^j_{01} < p\left(Y_j^+|x_i\right)\cdot c^j_{10} + p\left(Y_j^-|x_i\right)\cdot c^j_{00}, \tag{5}$$

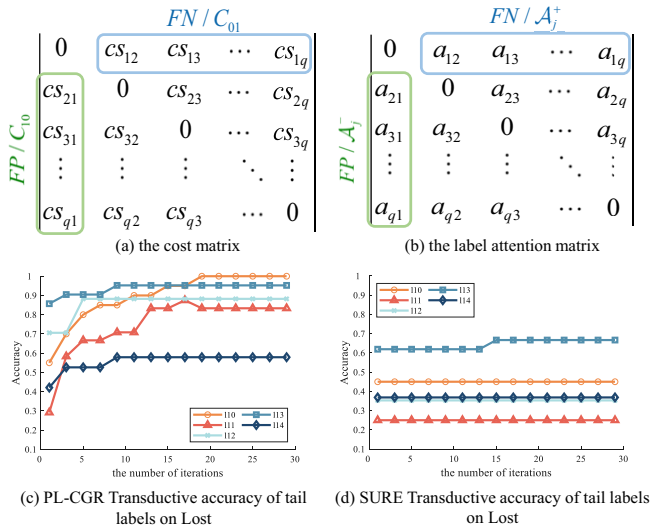
Here, $p\left(Y_j^+|x_i\right) = f_j\left(x_i\right)$ is defined as the probability that x_i belongs to Y_j. The function $f_j\left(x_i\right)$ is denoted by the label confidence of the model in assigning Y_j to x_i. $p\left(Y_j^-|x_i\right) = \sum_{z\neq j} f_z\left(x_i\right)$ is defined as the probability that x_i does not belong to Y_j. Given that $\sum f(x_i) = p\left(Y_j^+|x_i\right) + p\left(Y_j^-|x_i\right) = 1$, Eq. (5) can be reformulated as follows:

$$\begin{aligned} p\left(Y_j^+|x_i\right)\cdot c^j_{10} &> p\left(Y_j^-|x_i\right)\cdot c^j_{01} \\ p\left(Y_j^+|x_i\right) &> p^*_j = \frac{c^j_{01}}{c^j_{10} + c^j_{01}}. \end{aligned} \tag{6}$$

$P^* = \left[p^*_1, p^*_2, \cdots, p^*_q\right] \in \mathbb{R}^q$ is defined as the label threshold. The label threshold determines the direction of the label update. The label attention matrix, calculated as $a_{ij} = 1 - cs_{ij}$, is calculated from the cost matrix and determines the label update step size. The smaller the cs_{ij}, the more attention the algorithm pays to Y_j when misclassified. Similarly to the construction of the cost coefficient, the update step size for Y_j is specified as follows:

$$\begin{aligned} \mathcal{A}^+_j &= \frac{\sum_{z\neq j} a_{jz}}{\sum_{z\neq j} a_{jz} + \sum_{z\neq j} a_{zj}}, if f_j\left(x_i\right) < p^*_j, y_j \in Y_i \\ \mathcal{A}^-_j &= \frac{\sum_{z\neq j} a_{zj}}{\sum_{z\neq j} a_{jz} + \sum_{z\neq j} a_{zj}}, if f_z\left(x_i\right) < p^*_j, y_j \notin Y_i. \end{aligned} \tag{7}$$

$\mathcal{A}^+_j$ is defined as the label attention received in the case of a false negative case, and $\mathcal{A}^-_j$ is defined as the label attention received in the case of a false positive case.



(a) the cost matrix (b) the label attention matrix

(c) PL-CGR Transductive accuracy of tail labels on Lost (d) SURE Transductive accuracy of tail labels on Lost

Figure 2. (a) Showing the construction of the cost matrix for Y_j. (b) Showing the construction of the label attention coefficient for Y_j. (c) PL-CGR transductive accuracy of each tail label on Lost. (d) SURE transductive accuracy of each tail label on Lost.

3.2 Cost-guided strategy

Inspired by [25, 36], the partial label cost-guided loss is specified as follows:

$$R\left(x_i, p^*_j, \mathcal{A}_j\right) = \begin{cases} \mathcal{A}^+_j \max\left\{0, f_j\left(x_i\right) - p^*_j\right\}, y_j \in Y_i \\ \mathcal{A}^-_j \max\left\{0, p^*_j - f_j\left(x_i\right)\right\}, y_j \notin Y_i, \end{cases} \tag{8}$$

In PLL, the non-candidate label confidence is 0. Each instance is considered as the positive instance for candidate labels. We set the parameter λ to refine the algorithm's attention to labels, the loss function (8) can be transformed into:

$$\begin{aligned} R\left(x_i, p^*_j, \mathcal{A}_j\right) &= \lambda \times \left(\mathcal{A}^+_j \odot Y_i \odot \left(p_i - p^*_j\right)\right) \\ &= \lambda \times \eta_i. \end{aligned} \tag{9}$$

where $\eta_i = \mathcal{A}^+ \odot Y_i \odot \left(p_i - p^*\right) \in \mathbb{R}^q$. $\odot$ is the Hadamard product operation. Eq. (9) implements instance-based class imbalance attention via label thresholds and label attention coefficients. When the label confidence exceeds the label threshold, the algorithm distributes more attention to other labels, thus avoiding misidentification and reducing the problem of tail labels being ignored. As shown in Figure 2 (c) and (d), compared with the baseline method, PL-CGR can pay more attention to the tail labels and mitigate the effect of class imbalance on the algorithm.

3.3 PL-CGR

In this subsection, we introduce PL-CGR in detail. The specific form is as follows:

$$\min \sum_{i=1}^{m} \left(\ell\left(x_i, p_i, f\right) - R\left(p_i, p^*, \mathcal{A}\right)\right) + \mu\Omega\left(f\right) \tag{10}$$

where ℓ indicates label confidence loss, R denotes cost-guided loss, and Ω avoids model overfitting. μ is a hyperparameter that adjusts the weight of two PL-CGR terms. We use the squared loss to fit

Table 2. Details of the Real-world data sets.

Dataset	#instance	#features	#label	#Avg-Las	#IR	Data Domain
FG-NET	1002	262	78	7.48	47.00	Facial age estimation
Lost	1122	108	16	2.23	11.33	Automatic face naming
Mirflicker	2780	1536	14	2.76	392.00	Web image classification
MSRCv2	1758	48	23	3.16	85.00	Object classification
Soccer Player	17472	279	171	2.09	954.33	Automatic face naming
Yahoo! News	22991	163	219	1.91	308.79	Automatic face naming
Italian	21878	519	90	1.6	3544.00	POS Tagging
Malagasy	5303	384	44	8.35	278.50	POS Tagging

the label confidence, i.e. $\ell(x_i, p_i, f) = \left\| x_i W + b^\top - p_i \right\|_2^2$, where W and b are classifier parameters. To control the model complexity, we use the squared Frobenius norm, i.e. $\Omega(f) = \|W\|_2^2$. To sum up, the optimisation problem (10) can be transformed into:

$$\begin{aligned} &\min \sum_{i=1}^{m} \left(\left\| x_i W + b^\top - p_i \right\|_2^2 - \lambda \eta_i \odot p_i \right) + \mu \|W\|_F^2 \\ &s.t. \quad 0 \le p_{ij} \le Y_{ij} \, (1 \le i \le m, 1 \le j \le q), \sum_{j=1}^{q} p_{ij} = 1. \end{aligned} \tag{11}$$

The first constraint term guarantees that the ground-truth label must be in the candidate label set, while the non-candidate label confidence must be 0. The second constraint term normalizes the candidate label confidence. It is convenient to distinguish the probability of different candidate labels as ground-truth labels, i.e. selecting the label with the highest probability.

4 Optimization

In the previous section, we proposed the problem (11) with the convex property [12]. We solve this optimization problem by the alternating method. Specifically, the classifier parameters are updated with the candidate label confidence fixed, and the candidate label confidence is updated with the classifier fixed.

4.1 Updating classifier parameters

Fixing the label confidence, problem (11) can be expressed as:

$$\min_{W,b} \left\| XW + 1b^\top - P \right\|_2^2 + \mu \|W\|_F^2, \tag{12}$$

where $1 \in \mathbb{R}^m$ is the vector with all components 1. Closed solutions can be easily obtained by setting the gradients of W and b to zero:

$$\begin{aligned} W &= \left(X^\top X + \mu I - \frac{X^\top 1 1^\top X}{m} \right)^{-1} \left(X^\top P - \frac{X^\top 1 1^\top P}{m} \right) \\ b &= \frac{1}{m} \left(P^\top 1 - W^\top X^\top 1 \right), \end{aligned} \tag{13}$$

To handle the nonlinear case, the linear learning model can be easily extended to a kernel-based nonlinear model. We achieve this by using the feature mapping $\phi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{\mathcal{H}}$ to map the original feature space to some higher-dimensional Hilbert space. According to the representation theorem, W can be represented as a linear combination of the input variables, i.e. $W = \phi(X)^\top A$, where the combination weights of the stored instances in $A \in \mathbb{R}^{m \times q}$. Hence $\phi(X) W = KA$ where $K \in \phi(X) \phi(X)^\top$ is defined as kernel matrix, with each element denotes by $k_{ij} = \phi(x_i)^\top \phi(x_i) = k(x_i, x_j)$. In PL-CGR, we use Gaussian kernel function $k(x_i, x_j) = exp\left(- \|x_i - x_j\|_2^2 / \left(2\sigma^2\right) \right)$, where σ is set to the average pairwise distance among the instances. By kernel extension, the optimisation problem (13) can be expressed as follows:

$$\min \left\| KA + 1b^\top - P \right\|_2^2 + \mu tr\left(A^\top K A \right), \tag{14}$$

where $tr(\cdot)$ is the trace operator. By setting the gradient of A and b to 0, the closed-form solutions are expressed as follows:

$$\begin{aligned} A &= \left(K + \mu I - \frac{1 1^\top K}{m} \right)^{-1} \left(P - \frac{1 1^\top P}{m} \right) \\ b &= \frac{1}{m} \left(P^\top 1 - A^\top K^\top 1 \right). \end{aligned} \tag{15}$$

4.2 Label confidence updates

With A and b fixed, the model prediction label confidence $Q = [q_1, q_2, \ldots, q_m] \in \mathbb{R}^{m \times q}$ is denoted as $Q = KA + 1^\top b$. The problem (11) can be expressed as:

$$\begin{aligned} &\min \sum_{i=1}^{m} \left(\|p_i - q_i\|_2^2 - \lambda \eta_i \odot p_i \right) \\ &s.t. 0 \le p_{ij} \le y_{ij} \, (1 \le i \le m, 1 \le j \le q), \sum_{j=1}^{q} p_{ij} = 1, \end{aligned} \tag{16}$$

Problem (16) can be expressed in detail as

$$\begin{aligned} OP(P) &= \sum_{i=1}^{m} \sum_{j=1}^{q} \left((p_{ij} - q_{ij})^2 - \lambda \eta_{ij} p_{ij} \right) \\ &= \sum_{i=1}^{m} \sum_{j=1}^{q} \left(\left(p_{ij}^2 + q_{ij}^2 - 2 p_{ij} \times q_{ij} \right) - \lambda \eta_{ij} p_{ij} \right) \\ &= \sum_{i=1}^{m} \sum_{j=1}^{q} \left(p_{ij}^2 - 2 p_{ij} \times q_{ij} - (2 q_{ij} - \lambda \eta_{ij} p_{ij}) \right) + C, \end{aligned} \tag{17}$$

where $C = \sum_{i=1}^{m} \sum_{j=1}^{q} q_{ij}^2$ is a constant. To reduce the complexity of the algorithm, let $\widetilde{q} = vec(q) \in \mathbb{R}^{mq \times d}$, where $vec(\cdot)$ is the vectorisation operator. Likewise, $\widetilde{p} = vec(P) \in [0,1]^{mq \times d}$, $\widetilde{y} = vec(Y) \in \{0,1\}^{mq \times d}$ and $\widetilde{\eta} = vec(\eta) \in [0,1]^{mq \times d}$. Then minimizing function (17) is equivalent to solving the following function:

$$OP(\widetilde{p}) = \frac{1}{2} \widetilde{p}^\top H \widetilde{p} - (2\widetilde{q} - \lambda \widetilde{\eta}) \widetilde{p}, \tag{18}$$

Table 3. Accuracy (mean±std) and Precision (mean±std) of each comparing algorithm on the real-world partial label data sets. Note: •/○ indicates if PL-CGR's performance on each data set is statistically superior or inferior to the comparative algorithm (pairwise Wilcoxon Signed-Rank Test at 0.05 significance level).

Accuracy							
	PL-CGR	PLCL	PL-AGGD	SURE	IPAL	LALO	PL-KNN
Lost	0.813±0.041	0.775±0.049•	0.778±0.053•	0.785±0.057	0.720±0.027•	0.750±0.046•	0.595±0.036•
MSRCv2	0.590±0.052	0.503±0.054•	0.502±0.057•	0.480±0.058•	0.527±0.059•	0.481±0.062•	0.449±0.047•
FG-NET	0.091±0.014	0.083±0.014	0.088±0.027	0.079±0.018•	0.067±0.011•	0.080±0.012•	0.057±0.017•
Mirflicker	0.670±0.022	0.658±0.024•	0.669±0.025	0.669±0.025	0.533±0.024•	0.664±0.020•	0.549±0.022•
Malagasy	0.720±0.026	0.652±0.027•	0.655±0.040•	0.645±0.049•	0.633±0.020•	0.659±0.032•	0.610±0.031•
Soccer Player	0.567±0.013	0.553±0.013•	0.544±0.012•	0.534±0.011•	0.548±0.011•	0.540±0.010•	0.518±0.012•
Yahoo! News	0.667±0.006	0.653±0.008•	0.652±0.008•	0.635±0.010•	0.667±0.008	0.639±0.011•	0.587±0.011•
Italian	0.671±0.008	0.680±0.012	0.681±0.011	0.636±0.011•	0.582±0.013•	0.674±0.010	0.474±0.008•
Precision							
	PL-CGR	PLCL	PL-AGGD	SURE	IPAL	LALO	PL-KNN
Lost	0.663±0.058	0.566±0.078•	0.555±0.071•	0.535±0.073•	0.560±0.064•	0.520±0.078•	0.477±0.092•
MSRCv2	0.422±0.063	0.349±0.054•	0.350±0.060•	0.318±0.098•	0.428±0.058	0.302±0.084•	0.320±0.067•
FG-NET	0.026±0.014	0.025±0.015•	0.025±0.013•	0.020±0.007•	0.033±0.010	0.023±0.013	0.021±0.014•
Mirflicker	0.506±0.016	0.462±0.026•	0.457±0.017•	0.458±0.015•	0.451±0.019•	0.467±0.029•	0.548±0.023○
Malagasy	0.332±0.018	0.299±0.029•	0.278±0.023•	0.262±0.020•	0.298±0.018•	0.265±0.020•	0.242±0.015•
Soccer Player	0.410±0.039	0.361±0.025•	0.318±0.011•	0.267±0.023•	0.370±0.031•	0.004±0.001•	0.217±0.023•
Yahoo! News	0.659±0.024	0.635±0.017•	0.606±0.016•	0.573±0.022•	0.665±0.024	0.598±0.016•	0.549±0.024•
Italian	0.272±0.011	0.258±0.019•	0.269±0.017	0.267±0.009•	0.239±0.010•	0.282±0.015○	0.209±0.010•

where $H \in \mathbb{R}^{mq \times mq}$ is defined as follows:

$$H = \begin{bmatrix} T & 0_{q\times q} & \cdots & 0_{q\times q} \\ 0_{q\times q} & T & \cdots & 0_{q\times q} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{q\times q} & 0_{q\times q} & \cdots & T \end{bmatrix}, \tag{19}$$

where $T \in \mathbb{R}^{q\times q}$ is denoted the unit matrix with all 2 diagonal elements. $0_{q\times q}$ is denoted the all-zero matrix. The problem (16) is equivalent to the following equation:

$$\begin{aligned} &\min \frac{1}{2}\widetilde{p}H\widetilde{p} - (2\widetilde{q} - \lambda\widetilde{\eta})\,\widetilde{p} \\ &s.t. 0_{mq} \le \widetilde{p} \le \widetilde{y}, \sum_{j=1, j/m=i}^{mq} \widetilde{p}_j = 1\,(\forall 0 \le i \le m-1)\,. \end{aligned} \tag{20}$$

Obviously, the optimization problem (20) is a standard quadratic programming problem that can be solved by any off-the-shelf QP toolbox. After the optimization process is complete, the ground-truth labels for unknown instances are determined via the following equation:

$$\widetilde{y} = \arg\max_{j} \sum_{i=1}^{m} a_{ij} k\,(x^*, x_i) + b_j. \tag{21}$$

For label initialization, we initially train the mode based on the partial label matrix. Labels with the highest confidence within the candidate set are considered potential ground-truth labels and are assigned an additional attention weight of 0.2. The model prediction confidence is considered as the result of label initialization. Details can be found in [8]. The maximum number of iterations is 30.

5 Experiments

In this section, we evaluate PL-CGR on 8 real-world partial label datasets with 5 evaluation metrics on a total of 40 tasks. Firstly, we introduce the experimental setup. Secondly, we report detailed experimental results and statistical performance comparisons.

5.1 Experimental setup

Datasets. The 8 real-world partial label datasets[1] include FG-NET [26] for facial age estimation, Lost [3], Soccer Player [37] and Yahoo! News [13] for automatic face naming, MSRCv2 [22] for object classification, Mirflickr [17] for web image classification, Italian [20] and Malagasy [10] for POS tagging. In the automatic face naming task, faces cropped from images are treated as instances, with names extracted from associated descriptions or subtitles as corresponding candidate labels. In the object classification task, image segmentation is considered as instances and other objects appearing in the same image are the corresponding candidate labels. In the task of facial age estimation, each face is an instance, with ages annotated by ten crowdsourced labelers and the ground-truth age as candidate labels. For web image classification, images are represented as instances, and labels extracted from web pages are considered as candidate labels. For the POS label task, each word with contextual information can be considered as an instance and all possible POS labels are candidate labels. Table 2 records the details of each real-world partial label dataset, including the average number of candidate labels (Avg-Las), the imbalance ratio (IR), and other metrics. In experiments, we keep empty labels and extreme imbalance labels. All algorithms are compared under the condition that the training sets and test sets are randomly divided.

Comparing Methods. To prove the effectiveness of PL-CGR, we compared it with 6 partial label learning algorithms. Each comparison algorithm is set up according to the suggested parameters in the literature. The compared state-of-the-art PLL algorithms include: PLCL [19]: A partial label learning algorithm based on complementary classification. (Suggest configuration: $k = 10$, $\lambda = 0.03$ $\gamma, \mu, \alpha, \beta \in \{0.001, 0.01, 0.1, 0.2, 0.5, 1, 1.5, 2, 4\}$); PL-AGGD

[1] These datasets are publicly available at: https://palm.seu.edu.cn/zhangml

Table 4. Recall (mean±std) and F-measure (mean±std) of each comparing algorithm on the real-world partial label data sets. Note: •/○ indicates if PL-CGR's performance on each data set is statistically superior or inferior to the comparative algorithm (pairwise Wilcoxon Signed-Rank Test at 0.05 significance level).

Recall							
	PL-CGR	PLCL	PL-AGGD	SURE	IPAL	LALO	PL-KNN
Lost	0.617±0.063	0.507±0.050•	0.504±0.053•	0.510±0.064•	0.520±0.056•	0.477±0.040•	0.393±0.060•
MSRCv2	0.407±0.051	0.325±0.045•	0.322±0.043•	0.290±0.046•	0.388±0.059	0.303±0.053•	0.311±0.046•
FG-NET	0.035±0.014	0.030±0.016	0.033±0.017	0.031±0.012•	0.027±0.011•	0.031±0.013•	0.021±0.010•
Mirflicker	0.543±0.018	0.452±0.024•	0.453±0.025•	0.444±0.019•	0.454±0.016•	0.451±0.019•	0.414±0.020•
Malagasy	0.329±0.019	0.257±0.017•	0.240±0.019•	0.237±0.022•	0.301±0.011•	0.235±0.021•	0.241±0.012•
Soccer Player	0.167±0.013	0.115±0.010•	0.098±0.007•	0.077±0.007•	0.152±0.010•	0.005±0.001•	0.070±0.009•
Yahoo! News	0.493±0.021	0.456±0.014•	0.429±0.016•	0.390±0.022•	0.540±0.026	0.419±0.020•	0.472±0.026•
Italian	0.263±0.012	0.240±0.015•	0.230±0.010•	0.234±0.011•	0.240±0.010•	0.236±0.010•	0.159±0.010•
F-measure							
	PL-CGR	PLCL	PL-AGGD	SURE	IPAL	LALO	PL-KNN
Lost	0.619±0.054	0.512±0.059•	0.507±0.061•	0.504±0.066•	0.521±0.056•	0.475±0.051•	0.406±0.066•
MSRCv2	0.379±0.056	0.300±0.050•	0.296±0.053•	0.268±0.062•	0.372±0.054	0.271±0.060•	0.276±0.050•
FG-NET	0.023±0.008	0.023±0.013	0.024±0.012	0.019±0.006•	0.026±0.008	0.022±0.011•	0.018±0.009•
Mirflicker	0.511±0.015	0.439±0.017•	0.442±0.018•	0.437±0.015•	0.436±0.016•	0.441±0.015•	0.404±0.017•
Malagasy	0.318±0.016	0.265±0.020•	0.244±0.021•	0.240±0.022•	0.290±0.014•	0.239±0.021•	0.231±0.014•
Soccer Player	0.216±0.016	0.159±0.013•	0.137±0.006•	0.109±0.009•	0.197±0.015	0.004±0.001•	0.070±0.009•
Yahoo! News	0.541±0.021	0.508±0.014•	0.408±0.016•	0.442±0.022•	0.570±0.022○	0.471±0.019•	0.479±0.026•
Italian	0.246±0.009	0.233±0.013•	0.229±0.010•	0.231±0.009•	0.221±0.008•	0.240±0.009•	0.170±0.009•

[30]: A Graph-Based disambiguation method by using adaptive graph construction to generate label confidence for label disambiguation. (Suggest configuration: k = 10. T = 20, λ = 1, μ = 1, γ = 0.05); SURE [8]: Self-guided retraining baseline. (Suggest configuration: $\lambda, \beta \in \{0.001, 0.01, 0.05, 0.1, 0.3, 0.5, 1\}$); LALO [7]: A disambiguation method by utilizing the latent label distribution for label identification (Suggest configuration: k = 10, λ = 0.05, μ = 0.005); IPAL [41]: A Graph-Based disambiguation method by considering the instance similarity. (Suggest configuration: α=0.95, k = 10, T = 100); PL-KNN [18]: K-nearest neighbour PLL method based on average disambiguation strategy. (Suggest configuration: k = 10).

For each dataset, the validity of the algorithm is checked by a ten-fold cross-check and the average prediction accuracy and standard deviation are recorded. Furthermore, to determine whether PL-CGR is superior/inferior (win/loss) to comparing algorithms in all experiments, we used a Wilcoxon Signed-Rank test at 0.05 significance level for two independent samples.

5.2 Experimental results

The Classification accuracy. The performance of PL-CGR and the comparison methods are evaluated on eight real datasets, the details of which are shown in Tables 3. The average number of candidate labels in FG-NET is quite large, which could cause the extremely low classification accuracy of all algorithms. The following conclusions can be obtained from the performance indicators:

- When compared to the self-guided retraining baseline SURE, PL-CGR significantly outperforms the comparison algorithm in 75% of the cases. And PL-CGR outperforms SURE in all real-world tasks in terms of accuracy.
- When compared to other approaches, PL-CGR achieved better performance than PLCL, PL-AGGD, IPAL, LALO and PL-KNN in 75%, 63%, 88%, 88% and 100% of the cases respectively.
- In ten-fold cross-validation experiments on eight real-world tasks, PL-CGR achieved the best accuracy on seven datasets. Performance on the Italian dataset was slightly lower than PL-AGGD. Performance results on the Lost, MSRCv2 and Malagasy datasets are impressive.

The class imbalance indicator. To more accurately assess the performance of PL-CGR in addressing class imbalance issues, we employed 4 class imbalance evaluation metrics [32], including Average Precision, Average Recall, Average F-measure, and MAUC, for algorithm validation.

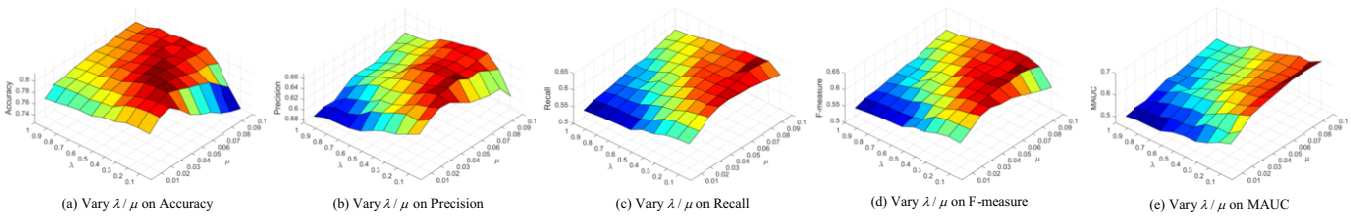
- Average Precision: $Avg\mathcal{P} = \frac{1}{q}\sum_{j=1}^{q}\mathcal{P}_j = \frac{1}{q}\sum_{j=1}^{q}\frac{c_{jj}}{\sum_{k=1}^{q}c_{kj}}$
- Average Recall: $Avg\mathcal{R} = \frac{1}{q}\sum_{j=1}^{q}\mathcal{R}_j = \frac{1}{q}\sum_{j=1}^{q}\frac{c_{jj}}{n_j}$
- Average F-measure: $Avg\mathcal{F} = \frac{1}{q}\sum_{j=1}^{q}\mathcal{F}_j = \frac{1}{q}\sum_{j=1}^{q}\frac{2\mathcal{P}_j\times\mathcal{R}_j}{\mathcal{P}_j+\mathcal{R}_j}$
- MAUC: $MAUC = \frac{2}{(q(q-1))}\sum_{1\le j<k\le q}\frac{A_{jk}+A_{kj}}{2}$

Here, n_j denotes the number of instances belonging to y_j. $C = [c_{jk}] \in \mathbb{R}^{q\times q}$ is the count matrix, where c_{jk} represents the number of instances misclassified as y_j by the classifier , but whose ground-truth labels are y_k. The main diagonal element of C shows the number of correctly classified for each label. A_{jk} is defined as the area under the ROC curve [14, 1] between y_j and y_k calculated from $F_{\cdot j}$. Table 3-5 show the results of ten-fold cross-validation for each algorithm across 4 evaluation metrics and 8 datasets. Table 6 shows the counts of wins/ties/losses for each partial label learning algorithm under 32 statistical tests. The excellence of the PL-CGR algorithm can be shown in the following three points:

- When compared to the self-guided retraining baseline SURE, PL-CGR significantly outperforms the comparison algorithm in 97% of the cases.
- When compared to other approaches, PL-CGR achieved better performance than PLCL, PL-AGGD, IPAL, LALO and PL-KNN in 91%, 84%, 53%, 91% and 91% of the cases respectively.
- On the Lost dataset, PL-CGR outperforms the comparison algorithm in all cases. On the Malagasy, Mirflickr and Soccer Player datasets, PL-CGR outperforms the comparison algorithms in 96% of the cases.

Table 5. MAUC (mean±std) of each comparing algorithm on the real-world partial label data sets. Note: •/○ indicates if PL-CGR's performance on each data set is statistically superior or inferior to the comparative algorithm (pairwise Wilcoxon Signed-Rank Test at 0.05 significance level).

	MAUC						
	PL-CGR	PLCL	PL-AGGD	SURE	IPAL	LALO	PL-KNN
Lost	0.609±0.083	0.442±0.088•	0.425±0.082•	0.409±0.090•	0.558±0.140•	0.407±0.079•	0.443±0.098•
MSRCv2	0.515±0.067	0.374±0.075•	0.352±0.059•	0.321±0.080•	0.669±0.064○	0.328±0.076•	0.440±0.064•
FG-NET	0.088±0.015	0.078±0.008	0.079±0.010	0.057±0.010	0.205±0.032	0.087±0.011	0.124±0.022
Mirflicker	0.721±0.001	0.434±0.071•	0.393±0.001•	0.414±0.042•	0.725±0.001○	0.431±0.051•	0.587±0.001•
Malagasy	0.222±0.014	0.186±0.022•	0.172±0.021•	0.145±0.021•	0.246±0.013	0.155±0.015•	0.179±0.019•
Soccer Player	0.343±0.042	0.204±0.017•	0.160±0.015•	0.122±0.014•	0.373±0.043	0.087±0.003•	0.128±0.015•
Yahoo! News	0.500±0.032	0.451±0.020•	0.408±0.019•	0.360±0.027•	0.621±0.036○	0.396±0.021•	0.597±0.043○
Italian	0.246±0.016	0.198±0.024•	0.187±0.020•	0.215±0.019•	0.389±0.024○	0.226±0.015•	0.172±0.011•



(a) Vary λ / μ on Accuracy (b) Vary λ / μ on Precision (c) Vary λ / μ on Recall (d) Vary λ / μ on F-measure (e) Vary λ / μ on MAUC

Figure 3. Experiments on parameter analysis in the PL-CGR on Lost. (a) Experiment on parameter analysis of λ/μ on Accuracy; (b) Experiment on parameter analysis of λ/μ on Precision; (c) Experiment on parameter analysis of λ/μ on Recall; (d) Experiment on parameter analysis of λ/μ on F-measure; (e) Experiment on parameter analysis of λ/μ on MAUC.

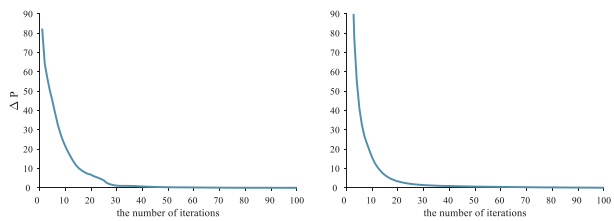
Table 6. Win/tie/loss Counts of PL-CGR's classification performance against each compared method on controlled Real-word data sets (pairwise Wilcoxon Signed-Rank Test at 0.05 significance level).

	PL-CGR against					
	PLCL	PL-AGGD	SURE	IPAL	LALO	PL-KNN
Precision	8/0/0	6/2/0	8/0/0	5/3/0	6/1/1	7/0/1
Recall	7/1/0	7/1/0	8/0 /0	6/2/0	8/0/0	8/0/0
f-measure	7/1/0	7/1/0	8/0/0	5/2/1	8/0/0	8/0/0
MAUC	7/1/0	7/1/0	7/1/0	1/2/5	7/1/0	6/1/1
sum	29/3/0	27/5/0	31/1/0	17/9/6	29/2/1	29/1/2

The Soccer Player dataset has only one head label, so the algorithm performs poorly on this dataset with the class imbalance metrics.

5.3 *Further analysis*

The two PL-CGR parameters λ and μ are also worth investigating. The performance of PL-CGR on the Lost dataset, across various evaluation metrics and parameter configurations, is shown in Figure 3. The algorithm performs better under Accuracy, Precision, and F-measure evaluation metrics when $\lambda \in \{0.4, 0.5\}$,$\mu \in \{0.04, 0.05, 0.06\}$. The algorithm performs best in Recall performance when $\lambda \in \{0.3, 0.4\}$. The algorithm performs sub-optimally on MAUC when $\lambda = 0, 4$ and $\mu = 0.5$. Therefore, for PL-CGR, we set $\lambda \in \{0.4, 0.5\}$ and $\mu \in \{0.04, 0.05, 0.06\}$. And we demonstrate PL-CGR's convergence by analyzing changes in the optimization variable P between iterations ($\triangle P = \left\| P^{t+1} - P^{t} \right\|_2^2$). Figure 4 (a) and (d) present the convergence curves for the Lost and MSRCv2 datasets, respectively. It illustrates the algorithm's convergence.



(a) Convergence on Lost (b) Convergence on MSRCv2

Figure 4. (a) Convergence analysis on Lost. (b) Convergence analysis on MSRCv2.

6 Conclusion

In this paper, we propose a novel cost-guided method PL-CGR to solve the class imbalance in the self-guided retraining strategy. Unlike previous work, it guides label disambiguation by label thresholds and determines the step size of label disambiguation by label attention coefficients. The efficacy of PL-CGR is comprehensively validated under 8 real-world datasets and 5 evaluation metrics. In the future, we will investigate further research on refining label misclassification cases, improving the accuracy of label thresholds, and enhancing the reliability of prototype instances.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under grants (82272075), the Guangxi Science and Technology Project (AB21220037), and the Open Project of Guangdong Provincial Key Laboratory of Artificial Intelligence in Medical Image Analysis and Application (2022MKF01).

References

[1] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

[2] N. Charoenphakdee, Z. Cui, Y. Zhang, and M. Sugiyama. Classification with rejection based on cost-sensitive classification. In *International Conference on Machine Learning*, pages 1507–1517. PMLR, 2021.

[3] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *2009 IEEE conference on computer vision and pattern recognition*, pages 919–926. IEEE, 2009.

[4] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011.

[5] R.-J. Dong, J.-Y. Hang, T. Wei, and M.-L. Zhang. Can label-specific features help partial-label learning? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7432–7440, 2023.

[6] D. Elreedy, A. F. Atiya, and F. Kamalov. A theoretical distribution analysis of synthetic minority oversampling technique (smote) for imbalanced learning. *Machine Learning*, pages 1–21, 2023.

[7] L. Feng and B. An. Leveraging latent label distributions for partial label learning. In *IJCAI*, pages 2107–2113, 2018.

[8] L. Feng and B. An. Partial label learning with self-guided retraining. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3542–3549, 2019.

[9] J. Gan, J. Li, and Y. Xie. Robust svm for cost-sensitive learning. *Neural Processing Letters*, pages 1–22, 2022.

[10] D. Garrette and J. Baldridge. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147, 2013.

[11] X. Gong, J. Yang, D. Yuan, and W. Bao. Generalized large margin k nn for partial label learning. *IEEE Transactions on Multimedia*, 24: 1055–1066, 2021.

[12] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical methods of operations research*, 66(3):373–407, 2007.

[13] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 634–647. Springer, 2010.

[14] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45:171–186, 2001.

[15] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[16] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[17] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 39–43, 2008.

[18] E. Hüllermeier and J. Beringer. Learning from ambiguously labeled examples. In *Intelligent Data Analysis*, volume 10, pages 419–439, 2006.

[19] Y. Jia, C. Si, and M.-L. Zhang. Complementary classifier induced partial label learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 974–983, 2023.

[20] B. Johan, C. Bosco, A. Mazzei, et al. Converting a dependency treebank to a categorial grammar treebank for italian. In *Proceedings of the Eight international workshop on treebanks and linguistic theories (TLT8)*, pages 27–38. Educatt, 2009.

[21] J. Li, D. Azizov, L. Yang, and S. Liang. Contrastive continual learning with importance sampling and prototype-instance relation distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13554–13562, 2024.

[22] L. Liu and T. Dietterich. A conditional multinomial mixture model for superset label learning. *Advances in neural information processing systems*, 25, 2012.

[23] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama. Progressive identification of true labels for partial-label learning. In *International conference on machine learning*, pages 6500–6510. PMLR, 2020.

[24] G. Lyu, S. Feng, T. Wang, and C. Lang. A self-paced regularization framework for partial-label learning. *IEEE Transactions on Cybernetics*, 52(2):899–911, 2022.

[25] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach: a case study in intensive care monitoring. Technical report, Technical Report, 1999.

[26] G. Panis, A. Lanitis, N. Tsapatsoulis, and T. F. Cootes. Overview of research on facial ageing using the fg-net ageing database. *Iet Biometrics*, 5(2):37–46, 2016.

[27] H. Shao, Q. Xu, Z. Yang, P. Wen, G. Peifeng, and Q. Huang. Weighted roc curve in cost space: Extending auc to cost-sensitive learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[28] V. Shukla, Z. Zeng, K. Ahmed, and G. Van den Broeck. A unified approach to count-based weakly supervised learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[29] C. Si, Z. Jiang, X. Wang, Y. Wang, X. Yang, and W. Shen. Partial label learning with a partner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15029–15037, 2024.

[30] D.-B. Wang, M.-L. Zhang, and L. Li. Adaptive graph guided disambiguation for partial label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8796–8811, 2021.

[31] H. Wang, M. Xia, Y. Li, Y. Mao, L. Feng, G. Chen, and J. Zhao. Solar: Sinkhorn label refinery for imbalanced partial-label learning. *Advances in neural information processing systems*, 35:8104–8117, 2022.

[32] J. Wang and M.-L. Zhang. Towards mitigating the class-imbalance problem for partial label learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2427–2436, 2018.

[33] L.-Y. Wen, X. Wang, and F. Min. Cost-sensitive microbial data augmentation through matrix factorization. *Applied Intelligence*, 53(10): 12684–12700, 2023.

[34] Z. Xu, T. Xiao, W. He, Y. Wang, Z. Jiang, S. Chen, Y. Xie, X. Jia, D. Yan, and Y. Zhou. Spatial-logic-aware weakly supervised learning for flood mapping on earth imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22457–22465, 2024.

[35] X.-R. Yu, D.-B. Wang, and M.-L. Zhang. Partial label learning with emerging new labels. *Machine Learning*, 113(4):1549–1565, 2024.

[36] A. Zaoui, C. Denis, and M. Hebiri. Regression with reject option and application to knn. *Advances in Neural Information Processing Systems*, 33:20073–20082, 2020.

[37] Z. Zeng, S. Xiao, K. Jia, T.-H. Chan, S. Gao, D. Xu, and Y. Ma. Learning by associating ambiguously labeled images. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 708–715, 2013.

[38] B. Zhang, X. Li, Y. Ye, Z. Huang, and L. Zhang. Prototype completion with primitive knowledge for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3754–3762, 2021.

[39] C. Zhang, J. Bi, S. Xu, E. Ramentol, G. Fan, B. Qiao, and H. Fujita. Multi-imbalance: An open-source software for multi-class imbalance learning. *Knowledge-Based Systems*, 174:137–143, 2019.

[40] M.-L. Zhang and L. Wu. Lift: Multi-label learning with label-specific features. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):107–120, 2014.

[41] M.-L. Zhang and F. Yu. Solving the partial label learning problem: An instance-based approach. In *IJCAI*, pages 4048–4054, 2015.

[42] Y. Zhang, G. Yang, S. Zhao, P. Ni, H. Lian, H. Chen, and C. Li. Partial label learning via generative adversarial nets. In *ECAI 2020*, pages 1674–1681. IOS Press, 2020.

[43] W. Zheng and H. Zhao. Cost-sensitive hierarchical classification for imbalance classes. *Applied Intelligence*, 50(8):2328–2338, 2020.

[44] L. Zhou, Y. Zhang, J. Zhang, X. Qian, C. Gong, K. Sun, Z. Ding, X. Wang, Z. Li, Z. Liu, et al. Prototype learning guided hybrid network for breast tumor segmentation in dce-mri. *IEEE Transactions on Medical Imaging*, 2024.