Partial Multi-Label Learning via Credible Label Elicitation

Min-Ling Zhang, *Senior Member, IEEE* and Jun-Peng Fang

Abstract—Partial multi-label learning (PML) deals with the problem where each training example is associated with an overcomplete set of candidate labels, among which only some candidate labels are valid. The task of PML naturally arises in learning scenarios with inaccurate supervision, and the goal is to induce a multi-label predictor which can assign a set of proper labels for unseen instance. The PML training procedure is prone to be misled by false positive labels concealed in the candidate label set, which serves as the major modeling difficulty for partial multi-label learning. In this paper, a novel two-stage PML approach is proposed which works by eliciting credible labels from the candidate label set for model induction. In the first stage, the labeling confidence of candidate label for each PML training example is estimated via iterative label propagation. In the second stage, by utilizing credible labels with high labeling confidence, multi-label predictor is induced via pairwise label ranking coupled with virtual label splitting or maximum a posteriori (MAP) reasoning. Experimental studies show that the proposed approach can achieve highly competitive generalization performance by excluding most false positive labels from the training procedure via credible label elicitation.

Index Terms—Machine learning, multi-label learning, partial label learning, candidate label set, credible label elicitation

1 Introduction

Partial multi-label learning corresponds to one specific learning framework with inaccurate supervision, where a set of candidate labels are assigned to each training example which are only partially valid. For instance, in crowdsourcing image tagging (Fig. 1), among the set of candidate labels given by crowdsourcing annotators only some of them are valid ones due to potentially unreliable annotators. Actually, the need to learn from PML examples naturally arises in many real-world applications, where accurate supervision is difficult to be obtained from the collected data [15], [16], [21], [30], [32], [38].

Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimensional feature space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ denote the label space with q possible class labels. Given the PML training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq m\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector and $Y_i \subseteq \mathcal{Y}$ is the set of *candidate* labels associated with $\boldsymbol{x}_i$. The key assumption adopted by partial multi-label learning lies in that the ground-truth label set $\widetilde{Y}_i \subseteq \mathcal{Y}$ for $\boldsymbol{x}_i$ reside in the candidate label set, i.e., $\widetilde{Y}_i \subseteq Y_i$, and are not directly accessible to the learning algorithm. Accordingly, the task of PML is to induce a multi-label predictor $f : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ from $\mathcal{D}$, which can assign a set of proper labels for the unseen instance.

One straightforward strategy to learn from PML training examples is to treat all candidate labels in Y_i as ground-truth ones, and then employ off-the-shelf multi-label learning algorithms [11], [36], [40] to induce the multi-label predictor. However, the labeling noise brought by false positive labels in Y_i will significantly compromise the resulting multi-label training procedure. Another strategy is to estimate the confidence of each candidate label being the ground-truth one, where the confidence scores and predictive model are optimized in an iterative manner by confidence-weighted ranking loss minimization [30] or low-rank confidence matrix approximation [21], [32]. Nonetheless, the estimated confidence scores could be error-prone especially when the proportion of false positive labels is high, which in turn would impact the predictive model due to the iterative optimization procedure.

In this paper, a novel approach named Particle, i.e., *PARTIal multi-label learning via Credible Label Elicitation*, is proposed to learning from PML training examples. The basic idea of Particle is to mitigate the negative impact of false positive labels by eliciting credible labels from candidate label set, which will be treated as reliable labeling information for subsequent model induction. Briefly, in the first stage, credible labels with high labeling confidence are identified via iterative label propagation. In the second stage, by making use of the identified credible labels, multi-label predictor is induced via pairwise label ranking coupled with virtual label splitting or maximum a posteriori reasoning. Extensive experimental studies show that credible label elicitation serves as an effective strategy to tackle the major PML modeling difficulty of ground-truth labels being concealed in the candidate label set.

The rest of this paper is organized as follows. Firstly, related works on partial multi-label learning are briefly discussed. Secondly, the technical details of the proposed Particle approach are presented. Thirdly, detailed results of comparative studies are reported. Finally, we conclude and indicate several issues for future work.

- *The authors are with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and also with the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China. E-mail: {zhangml, fangjp}@seu.edu.cn.*

Manuscript received 10 July 2019; revised 22 Mar. 2020; accepted 30 Mar. 2020. Date of publication 6 Apr. 2020; date of current version 2 Sept. 2021.
(Corresponding author: Min-Ling Zhang.)
Recommended for acceptance by M. Varma.
Digital Object Identifier no. 10.1109/TPAMI.2020.2985210

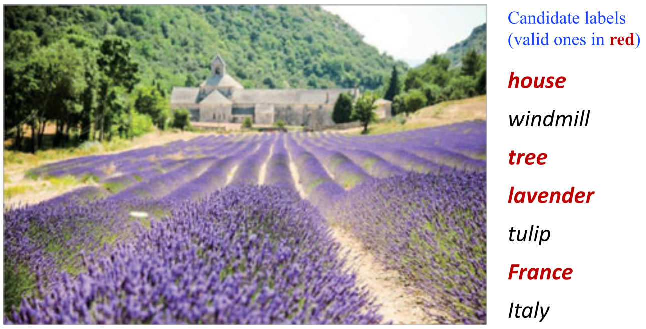


Fig. 1. An exemplary partial multi-label learning scenario. In crowdsourcing image tagging, among the set of 7 candidate labels given by crowdsourcing annotators, only 4 of them are valid ones including *house*, *tree*, *lavender*, and *France*.

2 Related Work

Conceptually speaking, partial multi-label learning is closely related to two popular learning frameworks, i.e., *multi-label learning* [11], [36], [40] and *partial label learning* [7], [18], [35].

Multi-label learning (MLL) deals with the problem where each example is associated with multiple valid labels simultaneously. Existing MLL approaches can be roughly categorized into three groups based on the order of label correlations exploited for model induction, including *first-order approaches* assuming independence among class labels [1], [33], *second-order approaches* assuming pairwise label correlations [10], [14], [17], and *high-order approaches* assuming label correlations among a subset of or whole set of class labels [2], [19], [20], [22], [25]. MLL and PML share the same goal of inducing predictive model which can assign a set of proper labels for unseen instance. Nonetheless, the task of PML is more challenging than MLL as the ground-truth labeling information is not directly accessible to PML learning algorithm. There are also studies on *weak label learning* [23], [24], [27] where there are ground-truth labels missing from the associated label set. Therefore, weak label learning and PML can be viewed as dual variants of MLL with noisy labeling, where weak label learning assumes false negative labels within irrelevant label set while PML assumes false positive labels within candidate label set.

Partial label learning (PLL) deals with the problem where each example is associated with multiple candidate labels among which only one is valid. The task of partial label learning is to induce a multi-class predictive model which can assign one proper label for unseen instance, where existing PLL approaches work by disambiguating the candidate label set [4], [7], [12], [18], [31] or transforming partial label learning problem into canonical supervised learning problems [5], [29], [35]. PLL and PML share similar setting of learning from noisy training examples where false positive labels reside in the candidate label set. Nonetheless, the task of PML is more challenging than PLL as a multi-label predictor rather than a single-label predictor needs to be induced from PML training examples.

The most straightforward strategy to tackle the problem of PML modeling is to treat all candidate labels as ground-truth ones. Thereafter, any off-the-shelf multi-label learning algorithms can be applied to induce the desired multi-label predictor. Nevertheless, it is obvious that the effectiveness of this straightforward strategy tends to suffer from the false positive labels concealed in candidate label set. On the other hand, one can choose to disambiguate candidate label set by estimating the confidence of each candidate label being the ground-truth one. An iterative procedure is thus employed to alternatively optimize the confidence scores and predictive model via confidence-weighted ranking loss minimization [30], low-rank confidence matrix approximation [21], [32], or discriminative modeling based on quadratic programming (QP) [13], [26]. Due to alternative nature of the optimization procedure, estimation errors on confidence scores may keep accumulating over the optimization iterations and thus impair the coupled predictive model, especially when the proportion of false positive labels in candidate label set is high.

In the next section, a two-stage partial multi-label learning strategy based on credible label elicitation will be introduced, which aims to mitigate the negative impact of false positive labels by exploiting reliable labeling information.

3 The Proposed Approach

The proposed Particle approach consists of two basic stages, i.e., *credible label elicitation* aiming to identify reliable labeling information from candidate label set, and *predictive model induction* aiming to make use of the identified information for follow-up model training. Technical details of Particle are scrutinized as follows.

3.1 Credible Label Elicitation

In the first stage, to elicit credible labels from the candidate label set, Particle works by adapting the label propagation procedure based on weighted graph over training instances. In this way, the structural information in feature space is utilized to facilitate identifying reliable labeling information in label space.

Given the PML training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq m\}$, a weighted directed graph $G = (V, E, \mathbf{W})$ is instantiated based on kNN minimum error reconstruction. Here, $V = \{\boldsymbol{x}_i \mid 1 \leq i \leq m\}$ corresponds to the set of training instances, $E = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid i \in \mathcal{N}(\boldsymbol{x}_j), 1 \leq j \leq m\}$ corresponds to the set of directed edges with $\mathcal{N}(\boldsymbol{x}_j)$ being the index set of $\boldsymbol{x}_j$'s k nearest neighbors in $\mathcal{D}$, and $\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m]^\top$ corresponds to the weight matrix with $\boldsymbol{w}_j = [w_{1,j}, w_{2,j}, \ldots, w_{m,j}]^\top$ $(1 \leq j \leq m)$ being the weight vector w.r.t. $\boldsymbol{x}_j$.

Specifically, $\boldsymbol{w}_j$ is optimized by solving the following minimum error reconstruction problem:

$$\begin{aligned} \min_{\boldsymbol{w}_j} \quad & \left\| \boldsymbol{x}_j - \sum_{i=1}^{m} w_{i,j} \cdot \boldsymbol{x}_i \right\|_2^2 \\ \text{s.t.} \quad & w_{i,j} \geq 0 \quad (i \in \mathcal{N}(\boldsymbol{x}_j)) \\ & w_{i,j} = 0 \quad (i \notin \mathcal{N}(\boldsymbol{x}_j)). \end{aligned} \tag{1}$$

Conceptually, the goal of Eq. (1) is to minimize the loss of reconstructing $\boldsymbol{x}_j$ from its k nearest neighbors with non-negative weights. Accordingly, the solution to the linear least square problem of Eq. (1) can be obtained by applying off-the-shelf QP solver.

Let $\mathbf{D} = \mathrm{diag}[d_1, d_2, \ldots, d_m]$ be the diagonal matrix with $d_j = \sum_{i=1}^m w_{i,j}$. Then, the propagation matrix $\mathbf{H} = \mathbf{W}\mathbf{D}^{-1}$ is formed by normalizing the columns of $\mathbf{W}$. Furthermore, let

$\mathbf{F} = [f_{i,c}]_{m \times q}$ denote the $m \times q$ matrix with non-negative entries where $f_{i,c} \geq 0$ corresponds to the confidence of y_c being a valid label for $\boldsymbol{x}_i$.

The initial labeling confidence matrix $\mathbf{F}^{(0)}$ is set w.r.t. the PML training examples as follows:

$$\forall\ 1 \leq i \leq m: \qquad f_{i,c}^{(0)} = \begin{cases} \frac{1}{|Y_i|}, & \text{if} \;\; y_c \in Y_i \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

Specifically, Particle chooses to evenly distribute the initial labeling confidence over candidate label set. For the tth iteration, $\mathbf{F}$ is updated by propagating current labeling confidence over $\mathbf{H}$

$$\widehat{\mathbf{F}}^{(t)} = \alpha \cdot \mathbf{H}^{\top}\mathbf{F}^{(t-1)} + (1-\alpha) \cdot \mathbf{F}^{(0)}. \tag{3}$$

Here, the parameter $\alpha \in [0,1]$ controls the proportion of labeling information inherited from iterative propagation and initial labeling confidence $\mathbf{F}^{(0)}$. Afterwards, Particle rescales $\widehat{\mathbf{F}}^{(t)}$ into $\mathbf{F}^{(t)}$ by normalizing each row w.r.t. the candidate label set

$$\forall\ 1 \leq i \leq m: \qquad f_{i,c}^{(t)} = \begin{cases} \frac{\hat{f}_{i,c}^{(t)}}{\sum_{y_l \in Y_i} \hat{f}_{i,l}^{(t)}}, & \text{if} \;\; y_c \in Y_i \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

As the iterative label propagation procedure terminates, we use $\mathbf{F}^*$ to denote the final labeling confidence matrix.[1]

Based on $\mathbf{F}^*$, it is feasible to elicit credible labels for each PML training example by identifying candidate labels with high labeling confidence. To reduce the risk of overfitting with label propagation, Particle fulfills the elicitation task by further performing kNN aggregation. Given $\boldsymbol{x}_j$ and its k nearest neighbors indexed in $\mathcal{N}(\boldsymbol{x}_j)$, an aggregation weight vector $\boldsymbol{\omega}^j = [\omega_1^j, \omega_2^j, \ldots, \omega_m^j]^{\top}$ is set as

$$\begin{aligned} &\forall\ 1 \leq i \leq m: \\ &w_i^j = \begin{cases} 1 - \frac{\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sum_{r \in \mathcal{N}(\boldsymbol{x}_j)} \mathrm{dist}(\boldsymbol{x}_r, \boldsymbol{x}_j)}, & \text{if} \;\; i \in \mathcal{N}(\boldsymbol{x}_j) \\ 0, & \text{otherwise} \end{cases}. \end{aligned} \tag{5}$$

Here, $\mathrm{dist}(\cdot,\cdot)$ calculates the euclidean distance between two instances. Accordingly, the resulting labeling confidence vector $\boldsymbol{\lambda}^j = [\lambda_1^j, \lambda_2^j, \ldots, \lambda_q^j]^{\top}$ for $\boldsymbol{x}_j$ is obtained by aggregating $\mathbf{F}^*$ with $\boldsymbol{\omega}^j$

$$\boldsymbol{\lambda}^j = \mathbf{F}^{*\top} \cdot \boldsymbol{\omega}^j. \tag{6}$$

The set of credible labels $Y_j^{\mathcal{C}}$ for $\boldsymbol{x}_j$ is then identified by thresholding $\boldsymbol{\lambda}^j$:[2]

$$Y_j^{\mathcal{C}} = \{y_l \mid \lambda_l^j \geq thr,\ y_l \in Y_j\} \bigcup \{y_{l^*} \mid y_{l^*} = \mathop{\arg\max}_{y_l \in Y_j} \lambda_l^j\}. \tag{7}$$

Therefore, $Y_j^{\mathcal{C}} \subseteq Y_j$ is formed by candidate labels whose labeling confidence are greater than the specified threshold $thr \in [0,1]$. In addition, $Y_j^{\mathcal{C}}$ contains at least the candidate label with highest labeling confidence (i.e., y_{l^*}) so as to avoid empty credible label set.

3.2 Predictive Model Induction

In the second stage, Particle makes use of the credible labels elicited in the first stage for inducing the multi-label predictive model.

Let $\mathcal{D}^{\mathcal{C}} = \{(\boldsymbol{x}_i, Y_i^{\mathcal{C}}) \mid 1 \leq i \leq m\}$ denote the transformed PML training set, where each training example $\boldsymbol{x}_i$ is associated with the credible label set $Y_i^{\mathcal{C}}$ other than the original candidate label set Y_i. Pairwise label ranking is tailored to learn from $\mathcal{D}^{\mathcal{C}}$, where similar techniques have been successfully applied to learn from multi-label data [10], [11], [36]. The results yielded by pairwise label ranking are further coupled with virtual label splitting or maximum a posteriori (MAP) reasoning to accomplish model induction.

For each transformed PML training example $(\boldsymbol{x}_i, Y_i^{\mathcal{C}})$ with $Y_i^{\mathcal{C}} \subseteq Y_i$, let $\bar{Y}_i = \mathcal{Y} \setminus Y_i$ denote the complementary set of candidate label set Y_i in $\mathcal{Y}$. Pairwise label ranking works by transforming the original learning problem into a number of binary learning problems, one for each label pair (y_u, y_z) $(1 \leq u < z \leq q)$. Specifically, one binary training set w.r.t (y_u, y_z) is generated from $\mathcal{D}^{\mathcal{C}}$ as follows:

$$\begin{aligned} \mathcal{D}_{uz}^{\mathcal{C}} = &\{(\boldsymbol{x}_i, \varphi(Y_i^{\mathcal{C}}, \bar{Y}_i, y_u, y_z)) \mid \\ &\tau(Y_i^{\mathcal{C}}, \bar{Y}_i, y_u, y_z) = \text{true}, 1 \leq i \leq m\} \quad \text{where} \end{aligned} \tag{8}$$

$$\tau(Y_i^{\mathcal{C}}, \bar{Y}_i, y_u, y_z) = \begin{cases} \text{true}, & \text{if} \;\; (y_u, y_z) \in Y_i^{\mathcal{C}} \times \bar{Y}_i \;\text{or} \\ & \qquad\; (y_z, y_u) \in Y_i^{\mathcal{C}} \times \bar{Y}_i \\ \text{false}, & \text{otherwise} \end{cases}$$

$$\varphi(Y_i^{\mathcal{C}}, \bar{Y}_i, y_u, y_z) = \begin{cases} +1, & \text{if} \;\; (y_u, y_z) \in Y_i^{\mathcal{C}} \times \bar{Y}_i \\ -1, & \text{if} \;\; (y_z, y_u) \in Y_i^{\mathcal{C}} \times \bar{Y}_i \end{cases}.$$

In other words, $\boldsymbol{x}_i$ will be utilized as a binary training example if y_u and y_z have different assignment w.r.t. $Y_i^{\mathcal{C}}$ and $\bar{Y}_i$. Otherwise, $\boldsymbol{x}_i$ will not contribute to the generation of binary training set $\mathcal{D}_{uz}^{\mathcal{C}}$.

Thereafter, a total of $\binom{q}{2}$ binary classifiers $g_{uz}: \mathcal{X} \mapsto \mathbb{R}$ can be induced from $\mathcal{D}_{uz}^{\mathcal{C}}$ by invoking some binary learning algorithm $\mathcal{B}$, i.e., $g_{uz} \leftarrow \mathcal{B}(\mathcal{D}_{uz}^{\mathcal{C}})$. Based on the modeling outputs yielded by $\binom{q}{2}$ binary classifiers, Particle proceeds to predict the set of proper labels for unseen instance $\boldsymbol{x}$ via virtual label splitting or MAP reasoning.

3.2.1 *Virtual Label Splitting*

In this case, one virtual label y_V is introduced to serve as an artificial splitting point between credible labels and non-candidate labels. Accordingly, for each class label y_u $(1 \leq u \leq q)$, one extra binary training set is generated from $\mathcal{D}^{\mathcal{C}}$ as follows:

$$\begin{aligned} \mathcal{D}_{uV}^{\mathcal{C}} = &\{(\boldsymbol{x}_i, \psi(Y_i^{\mathcal{C}}, \bar{Y}_i, y_u)) \mid \\ &\zeta(Y_i^{\mathcal{C}}, \bar{Y}_i, y_u) = \text{true}, 1 \leq i \leq m\} \quad \text{where} \end{aligned} \tag{9}$$

$$\zeta(Y_i^{\mathcal{C}}, \bar{Y}_i, y_u) = \begin{cases} \text{true}, & \text{if} \;\; y_u \in Y_i^{\mathcal{C}} \bigcup \bar{Y}_i \\ \text{false}, & \text{otherwise} \end{cases}$$

$$\psi(Y_i^{\mathcal{C}}, \bar{Y}_i, y_u) = \begin{cases} +1, & \text{if} \;\; y_u \in Y_i^{\mathcal{C}} \\ -1, & \text{if} \;\; y_u \in \bar{Y}_i \end{cases}.$$

[1] The iterative label propagation procedure terminates when $\mathbf{F}^{(t)}$ does not change or the maximum number of iterations (1,000 in this paper) is reached.

[2] To facilitate the thresholding operation, $\boldsymbol{\lambda}^j$ is further normalized to [0,1] with $\lambda_l^j = \frac{\lambda_l^j - \min_{1 \leq l \leq q} \lambda_l^j}{\max_{1 \leq l \leq q} \lambda_l^j - \min_{1 \leq l \leq q} \lambda_l^j}$.

In other words, $\boldsymbol{x}_i$ will be utilized as a binary training example if y_u belongs to $Y_i^{\mathcal{C}}$ or $\bar{Y}_i$. Otherwise, $\boldsymbol{x}_i$ will not contribute to the generation of binary training set $\mathcal{D}_{uV}^{\mathcal{C}}$.

Accordingly, a total of q extra binary classifiers $g_{uV}:\mathcal{X}\mapsto\mathbb{R}$ can be induced from $\mathcal{D}_{uV}^{\mathcal{C}}$ by invoking $\mathcal{B}$ as well, i.e., $g_{uV}\leftarrow\mathcal{B}(\mathcal{D}_{uV}^{\mathcal{C}})$. Furthermore, let r_{uz} and r_{uV} be the empirical accuracy of g_{uz} and g_{uV} in classifying binary training examples in $\mathcal{D}_{uz}^{\mathcal{C}}$ and $\mathcal{D}_{uV}^{\mathcal{C}}$ respectively. Given all these $\binom{q}{2}+q$ classifiers, for unseen instance $\boldsymbol{x}$, their (weighted) votes on each class label y_u $(1\le u\le q)$ and the virtual label y_V correspond to

$$\Gamma(\boldsymbol{x},y_u)=\sum\nolimits_{l=1}^{u-1} r_{lu}\cdot[\![g_{lu}(\boldsymbol{x})\le 0]\!]+ \sum\nolimits_{l=u+1}^{q} r_{ul}\cdot[\![g_{ul}(\boldsymbol{x})>0]\!]+r_{uV}\cdot[\![g_{uV}(\boldsymbol{x})>0]\!] \tag{10}$$

$$\Gamma(\boldsymbol{x},y_V)=\sum\nolimits_{l=1}^{q} r_{lV}\cdot[\![g_{lV}(\boldsymbol{x})\le 0]\!]. \tag{11}$$

Here, $[\![\pi]\!]$ returns 1 if predicate π holds and 0 otherwise. Thereafter, the predicted label set for $\boldsymbol{x}$ is determined as

$$f(\boldsymbol{x})=\{y_u\,|\,\Gamma(\boldsymbol{x},y_u)>\Gamma(\boldsymbol{x},y_V),1\le u\le q\}. \tag{12}$$

3.2.2 MAP Reasoning

In this case, a simple counting statistic is utilized to enable model prediction based on MAP reasoning. For unseen instance $\boldsymbol{x}$, let C_u denote the statistic which counts the average number of binary classifiers which vote for y_u on the k nearest neighbors of $\boldsymbol{x}$ indexed in $\mathcal{N}(\boldsymbol{x})$

$$C_u=\left\lceil\frac{1}{k}\cdot\sum_{r\in\mathcal{N}(\boldsymbol{x})}\left(\sum_{l=1}^{u-1}[\![g_{lu}(\boldsymbol{x}_r)\le 0]\!]+\sum_{l=u+1}^{q}[\![g_{ul}(\boldsymbol{x}_r)>0]\!]\right)\right\rceil. \tag{13}$$

Here, $0\le C_u\le q-1$ as among the $\binom{q}{2}$ binary classifiers generated by pairwise label ranking, $q-1$ of them are related to y_u.

We use H_u to denote the event that y_u is a relevant label for $\boldsymbol{x}$. Accordingly, let $\mathbb{P}(H_u\,|\,C_u)$ be the posteriori probability that H_u holds given C_u, and $\mathbb{P}(\neg H_u\,|\,C_u)$ be the posteriori probability that H_u does not hold given the same condition. Thereafter, the predicted label set for $\boldsymbol{x}$ is determined by the MAP rule

$$f(\boldsymbol{x})=\{y_u\,|\,\mathbb{P}(H_u\,|\,C_u)>\mathbb{P}(\neg H_u\,|\,C_u),1\le u\le q\}. \tag{14}$$

Based on Bayes theorem, we have

$$\frac{\mathbb{P}(H_u\,|\,C_u)}{\mathbb{P}(\neg H_u\,|\,C_u)}=\frac{\mathbb{P}(H_u)\cdot\mathbb{P}(C_u\,|\,H_u)}{\mathbb{P}(\neg H_u)\cdot\mathbb{P}(C_u\,|\,\neg H_u)}. \tag{15}$$

To carry out MAP reasoning, it suffices to compute the four terms $\mathbb{P}(H_u)$, $\mathbb{P}(\neg H_u)$, $\mathbb{P}(C_u\,|\,H_u)$ and $\mathbb{P}(C_u\,|\,\neg H_u)$ in Eq. (15).

For the prior terms $\mathbb{P}(H_u)$ and $\mathbb{P}(\neg H_u)$, their values are estimated via relative frequency counting with Laplacian smoothing

$$\mathbb{P}(H_u)=\frac{1+\sum_{i=1}^{m}[\![y_u\in Y_i]\!]}{2+m},\mathbb{P}(\neg H_u)=1-\mathbb{P}(H_u). \tag{16}$$

For the likelihood terms $\mathbb{P}(C_u\,|\,H_u)$ and $\mathbb{P}(C_u\,|\,\neg H_u)$, two frequency arrays κ_u and $\bar{\kappa}_u$ each with q elements are defined as follows:

TABLE 1
The Pseudo-Code of Particle

```
Inputs:
$\mathcal{D}$:       PML training set $\{(\boldsymbol{x}_i,Y_i)\mid 1\le i\le m\}$
          $(\boldsymbol{x}_i\in\mathcal{X},Y_i\subseteq\mathcal{Y},\mathcal{X}=\mathbb{R}^d,\mathcal{Y}=\{y_1,y_2,\ldots,y_q\})$
$k$:       number of nearest neighbors considered
$\alpha$:       balancing parameter
$thr$:     thresholding parameter
$\mathcal{B}$:       binary learning algorithm
$mode$:    virtual label splitting or MAP reasoning
$\boldsymbol{x}$:       unseen instance

Outputs:
$Y$:       predicted label set for $\boldsymbol{x}$

Process:
1: Instantiate the weighted graph $G=(V,E,\mathbf{W})$ by solving Eq.(1)
   with $k$NN minimum error reconstruction;
2: Initialize $\mathbf{F}^{(0)}$ according to Eq.(2) and obtain the final labeling
   confidence matrix $\mathbf{F}^*$ by conducting iterative label propagation
   according to Eq.(3) and Eq.(4);
3: Identify the credible label set $Y_j^{\mathcal{C}}$ for each example $\boldsymbol{x}_j$ $(1\le j\le m)$
   according to Eq.(7) (together with Eq.(5) and Eq.(6));
4: For each label pair $(y_u,y_z)$ $(1\le u<z\le q)$, generate binary
   training set $\mathcal{D}_{uz}^{\mathcal{C}}$ according to Eq.(8);
5: Induce binary classifier $g_{uz}\leftarrow\mathcal{B}(\mathcal{D}_{uz}^{\mathcal{C}})$;
6: switch $mode$ do
7:   case virtual label splitting
8:     For each label $y_u$ $(1\le u\le q)$, generate binary training set
       $\mathcal{D}_{uV}^{\mathcal{C}}$ according to Eq.(9);
9:     Induce binary classifier $g_{uV}\leftarrow\mathcal{B}(\mathcal{D}_{uV}^{\mathcal{C}})$;
10:    Return $Y=f(\boldsymbol{x})$ according to Eq.(12) (together with Eq.(10)
       and Eq.(11));
11:  case MAP reasoning
12:    For each label $y_u$ $(1\le u\le q)$, set the counting statistic $C_u$
       according to Eq.(13);
13:    Return $Y=f(\boldsymbol{x})$ according to Eq.(14) (together with Eqs.(15)-
       (18));
14: end switch
```

$$\begin{aligned}&\forall\ 0\le p\le q-1:\\ &\kappa_u[p]=\sum\nolimits_{i=1}^{m}[\![y_u\in Y_i]\!]\cdot[\![\delta_u(\boldsymbol{x}_i)=p]\!]\\ &\bar{\kappa}_u[p]=\sum\nolimits_{i=1}^{m}[\![y_u\notin Y_i]\!]\cdot[\![\delta_u(\boldsymbol{x}_i)=p]\!].\end{aligned} \tag{17}$$

Accordingly, $\delta_u(\boldsymbol{x}_i)=\lceil\frac{1}{k}\cdot\sum_{r\in\mathcal{N}(\boldsymbol{x}_i)}\big(\sum_{l=1}^{u-1}[\![g_{lu}(\boldsymbol{x}_r)\le 0]\!]+\sum_{l=u+1}^{q}[\![g_{ul}(\boldsymbol{x}_r)>0]\!]\big)\rceil$ counts the average number of binary classifiers which vote for y_u on the k nearest neighbors of $\boldsymbol{x}_i$. Therefore, $\kappa_u[p]$ ($\bar{\kappa}_u[p]$) records the number of training examples which have (don't have) label y_u and whose k nearest neighbors receive an average of p votes for y_u from all the binary classifiers.

Thereafter, $\mathbb{P}(C_u\,|\,H_u)$ and $\mathbb{P}(C_u\,|\,\neg H_u)$ can be estimated via relative frequency counting with Laplacian smoothing as well

$$\begin{aligned}\mathbb{P}(C_u\,|\,H_u)&=\frac{1+\kappa_u[C_u]}{q+\sum_{p=0}^{q-1}\kappa_u[p]}\\ \mathbb{P}(C_u\,|\,\neg H_u)&=\frac{1+\bar{\kappa}_u[C_u]}{q+\sum_{p=0}^{q-1}\bar{\kappa}_u[p]}.\end{aligned} \tag{18}$$

To summarize, Table 1 gives the complete procedure of the proposed Particle approach. In the first stage, credible labels are elicited from the candidate label set for each PML training example via iterative label propagation (steps 1-3). In the second stage, a total of $\binom{q}{2}$ binary classifiers are generated by pairwise label ranking (steps 4-5), which are in turn

TABLE 2
Characteristics of the PML Experimental Data Sets

Data Set	#Examples	#Features	#Class Labels	avg. #CLs	avg. #GLs
music_emotion	6,833	98	11	5.29	2.42
music_style	6,839	98	10	6.04	1.44
mirflickr	10,433	100	7	3.35	1.77
yeastBP	560	5,548	217	30.43	21.56
emotions	593	72	6	3, 4, 5	1.86
scene	2,407	294	6	3, 4, 5	1.07
yeast	2,417	103	14	5, 7, 9, 11, 13	4.23
reference	5,411	860	14	5, 7, 9, 11, 13	1.15
eurlex_dc	8,636	100	15	5, 7, 9, 11, 13	1.01
eurlex_sm	12,679	100	15	5, 7, 9, 11, 13	1.53
health	8,109	1,483	20	5, 7, 9, 11, 13	1.64
entertainment	8,166	545	20	5, 7, 9, 11, 13	1.43
eurlex_ed	9,792	100	25	5, 7, 9, 11, 13	3.07
computer	11,235	880	25	5, 7, 9, 11, 13	1.51
CAL500	502	68	100	25, 30, 35, 40, 45	23.91

*For each PML data set, the average number of candidate labels (**avg. #CLs**) and the average number of ground-truth labels (**avg. #GLs**) are also recorded.*

coupled with virtual label splitting (steps 7-10) or MAP reasoning (steps 11-13) to induce the multi-label predictive model. Specifically, the two variants of Particle instantiated with virtual label splitting and MAP reasoning are termed as Particle-Vls and Particle-Map respectively.

4 Experiments

4.1 Experimental Setup

4.1.1 Data Sets

A total of fifteen synthetic as well as real-world PML data sets have been employed for comparative studies in this paper, whose characteristics are summarized in Table 2.

Specifically, a synthetic PML data set can be generated from one multi-label data set by adding random labeling noise. Given the multi-label example, some of its irrelevant labels are randomly drawn to form the candidate label set along with its relevant labels. As shown in Table 2, eleven benchmark multi-label data sets [36] are used to generate synthetic PML data sets, including emotions, scene, yeast, reference, eurlex_dc, eurlex_sm, health, entertainment, eurlex_ed, computer and CAL500. For each multi-label data set, different configurations are considered by varying the average number of candidate labels (avg. #CLs). Accordingly, a total of fifty-one synthetic PML data sets have been generated. Furthermore, four real-world PML data sets, including music_emotion, music_style, mirflickr and yeastBP, are also employed in this paper. The first three real-world PML data sets are derived from the image retrieval task [15], where candidate labels are collected from web users and further examined by human labelers to specify the ground-truth labels. The fourth real-world PML data set is derived from the protein-protein interaction prediction task [32], where candidate labels correspond to the biological process annotations of Yeast proteins archived on different periods from the Gene Ontology[3] with those annotations available in history but absent in recent periods as false positive labels.

4.1.2 Learning Approaches

On the one hand, two well-established multi-label learning approaches Ml-knn [37], and Lift [34] are employed for comparative studies, which are tailored to learn from PML training examples by treating all candidate labels as ground-truth ones. On the other hand, several state-of-the-art PML approaches which work by iteratively optimizing labeling confidences and predictive model are also employed for comparative studies, including Pml-lc and Pml-fp [30] based on confidence-weighted ranking loss minimization, Fpml [32] and Pml-lrs [21] based on low-rank confidence matrix approximation.

For compared learning approaches, parameters suggested in respective literature are used and Libsvm [3] is employed as the base learner to instantiate Lift. As shown in Table 1, parameters for Particle are set as follows: $k = 10$ (number of nearest neighbors considered), $\alpha = 0.95$ (balancing parameter) and $thr = 0.9$ (credible label elicitation threshold). Furthermore, Libsvm [3] is also employed as the binary learning algorithm $\mathcal{B}$ for Particle.

For performance evaluation, five widely-used multi-label metrics *hamming loss*, *one-error*, *coverage*, *ranking loss* and *average precision* are employed whose detailed definitions can be found in [11], [36], [40]. For the first four metrics, the smaller the metric value the better the performance. For *average precision*, the larger the metric value the better the performance. On each data set, ten-fold cross-validation is performed where the mean metric value as well as standard deviation are recorded for each learning approach.

4.2 Experimental Results

Tables 3, 4, 5, 6, and 7 report the detailed experimental results of each learning approach in terms of each evaluation metric. For brevity, among all the synthetic PML data sets, detailed results on some of the synthetic configurations are given, i.e., avg. #CLs being 3 and 5 for emotions and scene, 5 and 13 for yeast, reference, eurlex_dc, eurlex_sm, health, entertainment, eurlex_ed, computer, and 25 and 45 for CAL500.

[3] http://www.geneontology.org

TABLE 3
Experimental Results of Each Learning Approach on Synthetic as well as Real-World PML Data Sets in Terms of *Hamming Loss*, Where the Best Performance (the smaller the better) is Shown in Bold Face

Data Set	avg. #CLs	Particle-Vls	Particle-Map	Pml-lc	Pml-fp	Fpml	Pml-lrs	Ml-knn	Lift
music_emotion	5.29	**.215 ± .003**	.222 ± .006	.274 ± .004	.268 ± .004	.401 ± .019	.333 ± .005	.360 ± .010	.340 ± .004
music_style	6.04	**.120 ± .002**	.209 ± .013	.212 ± .004	.162 ± .003	.855 ± .002	.833 ± .004	.842 ± .008	.847 ± .005
mirflickr	3.35	.173 ± .004	**.160 ± .006**	.236 ± .006	.220 ± .005	.222 ± .006	.209 ± .006	.221 ± .006	.221 ± .005
yeastBP	30.43	**.042 ± .008**	.044 ± .008	.059 ± .009	.054 ± .008	.093 ± .009	.134 ± .008	.088 ± .009	.049 ± .009
emotions	3	**.200 ± .015**	.228 ± .018	.259 ± .031	.247 ± .019	.329 ± .018	.329 ± .024	.257 ± .026	.271 ± .027
	5	.359 ± .030	**.275 ± .023**	.299 ± .020	.281 ± .013	.688 ± .015	.688 ± .015	.687 ± .015	.687 ± .015
scene	3	.120 ± .008	**.119 ± .018**	.299 ± .013	.226 ± .008	.394 ± .123	.300 ± .009	.170 ± .021	.260 ± .009
	5	.398 ± .010	**.182 ± .013**	.378 ± .015	.273 ± .010	.821 ± .001	.819 ± .002	.820 ± .001	.820 ± .001
yeast	5	**.208 ± .008**	.215 ± .009	.250 ± .009	.392 ± .009	.239 ± .008	.302 ± .005	.198 ± .008	.207 ± .010
	13	.697 ± .008	**.266 ± .010**	.343 ± .007	.399 ± .008	.697 ± .008	.302 ± .008	.697 ± .008	.697 ± .008
reference	5	**.085 ± .002**	.148 ± .005	.145 ± .002	.154 ± .015	.092 ± .001	.118 ± .005	.088 ± .003	.090 ± .002
	13	.867 ± .003	.197 ± .022	**.173 ± .004**	.174 ± .004	.917 ± .001	.294 ± .015	.917 ± .001	.917 ± .001
eurlex_dc	5	**.025 ± .001**	.036 ± .002	.089 ± .001	.089 ± .001	.081 ± .007	.075 ± .004	**.025 ± .001**	.058 ± .018
	13	.848 ± .008	**.065 ± .002**	.108 ± .002	.108 ± .002	.924 ± .005	.744 ± .021	.932 ± .000	.920 ± .037
eurlex_sm	5	.066 ± .001	.079 ± .004	.161 ± .001	.111 ± .002	.112 ± .005	.115 ± .004	**.060 ± .002**	.111 ± .024
	13	.850 ± .007	**.102 ± .003**	.195 ± .003	.130 ± .002	.724 ± .241	.679 ± .014	.897 ± .000	.897 ± .000
health	5	**.075 ± .001**	.123 ± .003	.127 ± .001	.098 ± .002	.081 ± .002	.096 ± .001	.077 ± .002	.083 ± .010
	13	**.094 ± .004**	.148 ± .004	.130 ± .003	.108 ± .001	.917 ± .001	.255 ± .013	.917 ± .001	.917 ± .001
entertainment	5	.069 ± .002	.134 ± .009	.159 ± .002	.137 ± .002	.070 ± .002	.087 ± .004	**.068 ± .002**	.070 ± .002
	13	**.087 ± .003**	.162 ± .007	.168 ± .002	.137 ± .001	.927 ± .002	.190 ± .013	.926 ± .001	.927 ± .001
eurlex_ed	5	**.056 ± .000**	.080 ± .003	.109 ± .001	.085 ± .001	.070 ± .002	.070 ± .002	.085 ± .001	.083 ± .024
	13	**.057 ± .001**	.090 ± .002	.133 ± .001	.094 ± .002	.286 ± .221	.218 ± .014	.633 ± .012	.847 ± .027
computer	5	.054 ± .001	.124 ± .003	.093 ± .001	.103 ± .012	.057 ± .001	.072 ± .003	**.050 ± .001**	.057 ± .001
	13	**.064 ± .001**	.135 ± .004	.095 ± .001	.097 ± .010	.939 ± .001	.181 ± .013	.557 ± .015	.551 ± .052
CAL500	25	**.239 ± .005**	.231 ± .005	.282 ± .005	.282 ± .005	.263 ± .005	.282 ± .000	.253 ± .005	.268 ± .004
	45	**.231 ± .008**	**.231 ± .008**	.273 ± .007	.272 ± .006	.241 ± .008	.259 ± .001	.242 ± .006	.272 ± .008

TABLE 4
Experimental Results of Each Learning Approach on Synthetic as well as Real-World PML Data Sets in Terms of *Ranking Loss*, Where the Best Performance (the smaller the better) is Shown in Bold Face

Data Set	avg. #CLs	Particle-Vls	Particle-Map	Pml-lc	Pml-fp	Fpml	Pml-lrs	Ml-knn	Lift
music_emotion	5.29	.261 ± .007	**.245 ± .006**	.268 ± .005	.276 ± .007	.331 ± .008	.284 ± .007	.302 ± .008	.276 ± .008
music_style	6.04	.161 ± .005	.161 ± .006	.268 ± .011	**.146 ± .005**	.224 ± .007	.175 ± .005	.199 ± .006	.202 ± .010
mirflickr	3.35	.207 ± .006	**.093 ± .007**	.143 ± .007	.126 ± .011	.159 ± .018	.102 ± .006	.175 ± .019	.114 ± .009
yeastBP	30.43	.935 ± .037	**.283 ± .040**	.393 ± .047	.363 ± .041	.416 ± .057	.406 ± .048	.408 ± .060	.316 ± .054
emotions	3	.184 ± .024	**.170 ± .016**	.215 ± .037	.189 ± .017	.472 ± .031	.333 ± .042	.181 ± .028	.197 ± .032
	5	.261 ± .026	**.222 ± .019**	.247 ± .016	.262 ± .040	.469 ± .043	.464 ± .032	.323 ± .031	.330 ± .046
scene	3	.115 ± .014	.104 ± .015	.318 ± .021	.141 ± .010	.185 ± .020	**.094 ± .012**	.142 ± .013	.082 ± .009
	5	.217 ± .013	.192 ± .015	.430 ± .029	.237 ± .014	.409 ± .038	**.171 ± .016**	.283 ± .023	.265 ± .033
yeast	5	.193 ± .009	.180 ± .011	.200 ± .011	.421 ± .014	.212 ± .011	.380 ± .011	**.173 ± .007**	.176 ± .009
	13	.235 ± .012	.252 ± .014	.324 ± .014	.391 ± .012	**.229 ± .012**	.470 ± .019	.237 ± .012	.242 ± .016
referece	5	.272 ± .017	.282 ± .011	**.244 ± .011**	.330 ± .092	.248 ± .019	.282 ± .019	.254 ± .014	.250 ± .013
	13	.356 ± .017	.354 ± .009	.344 ± .012	.349 ± .014	**.275 ± .014**	.371 ± .019	.283 ± .016	.274 ± .020
eurlex_dc	5	**.045 ± .004**	.056 ± .004	.061 ± .004	.061 ± .004	.243 ± .036	.073 ± .004	.080 ± .010	.155 ± .061
	13	**.113 ± .005**	.122 ± .006	.137 ± .008	.137 ± .008	.326 ± .067	.160 ± .010	.201 ± .020	.283 ± .024
eurlex_sm	5	**.102 ± .007**	.105 ± .008	.402 ± .012	.133 ± .008	.311 ± .039	.138 ± .008	.119 ± .007	.248 ± .030
	13	.186 ± .006	**.171 ± .007**	.305 ± .005	.198 ± .003	.361 ± .015	.201 ± .005	.249 ± .006	.364 ± .032
health	5	.154 ± .006	.197 ± .007	.347 ± .008	.136 ± .007	.131 ± .006	.174 ± .010	**.128 ± .005**	.132 ± .005
	13	.217 ± .006	.267 ± .007	.229 ± .008	.183 ± .003	**.148 ± .008**	.220 ± .007	.163 ± .008	.180 ± .108
entertainment	5	.201 ± .009	.228 ± .008	.520 ± .013	.586 ± .008	.162 ± .005	.215 ± .006	**.154 ± .005**	.158 ± .006
	13	.283 ± .009	.292 ± .009	.408 ± .006	.582 ± .011	**.172 ± .009**	.268 ± .009	.187 ± .008	.180 ± .033
eurlex_ed	5	**.134 ± .004**	.150 ± .005	.445 ± .014	.166 ± .007	.326 ± .017	.172 ± .008	.169 ± .005	.304 ± .041
	13	**.174 ± .007**	.189 ± .007	.498 ± .026	.235 ± .009	.397 ± .021	.212 ± .006	.252 ± .010	.358 ± .024
computer	5	**.129 ± .007**	.240 ± .007	.317 ± .010	.373 ± .183	.128 ± .008	.200 ± .012	.257 ± .006	.131 ± .005
	13	.198 ± .005	.324 ± .010	.199 ± .004	.301 ± .103	.140 ± .010	.260 ± .007	.327 ± .007	**.139 ± .008**
CAL500	25	.479 ± .015	**.258 ± .008**	.363 ± .013	.363 ± .013	.261 ± .007	.272 ± .002	.268 ± .009	.264 ± .006
	45	.350 ± .011	**.260 ± .006**	.325 ± .009	.325 ± .010	.266 ± .007	.261 ± .001	.285 ± .005	.266 ± .008

TABLE 5
Experimental Results of Each Learning Approach on Synthetic as well as Real-World PML Data Sets in Terms of *One-Error*, Where the Best Performance (the smaller the better) is Shown in Bold Face

Data Set	avg. #CLs	Particle-Vls	Particle-Map	Pml-lc	Pml-fp	Fpml	Pml-lrs	Ml-knn	Lift
music_emotion	5.29	**.473 ± .019**	.474 ± .018	.532 ± .019	.540 ± .018	.592 ± .012	.556 ± .012	.544 ± .019	.554 ± .021
music_style	6.04	**.370 ± .016**	.450 ± .034	.593 ± .021	.406 ± .017	.404 ± .012	.404 ± .012	.384 ± .015	.406 ± .014
mirflickr	3.35	**.141 ± .013**	.145 ± .016	.393 ± .023	.312 ± .049	.317 ± .068	.179 ± .014	.456 ± .116	.314 ± .042
yeastBP	30.43	**.906 ± .054**	.912 ± .054	.926 ± .038	.922 ± .036	.980 ± .015	.970 ± .014	.953 ± .048	.913 ± .119
emotions	3	**.232 ± .041**	.269 ± .036	.389 ± .070	.331 ± .077	.554 ± .055	.530 ± .053	.288 ± .033	.318 ± .060
	5	**.322 ± .056**	.389 ± .048	.471 ± .029	.408 ± .083	.576 ± .097	.552 ± .042	.503 ± .068	.477 ± .059
scene	3	**.245 ± .035**	.289 ± .048	.509 ± .033	.364 ± .016	.468 ± .038	.287 ± .029	.300 ± .019	.243 ± .009
	5	**.363 ± .028**	.447 ± .031	.709 ± .023	.531 ± .031	.728 ± .064	.484 ± .041	.583 ± .029	.574 ± .045
yeast	5	**.221 ± .023**	.248 ± .028	.244 ± .026	.370 ± .025	.248 ± .027	.417 ± .031	.240 ± .025	.235 ± .025
	13	**.252 ± .030**	.397 ± .035	.586 ± .033	.516 ± .028	.255 ± .031	.636 ± .058	.254 ± .031	.259 ± .028
referece	5	**.536 ± .017**	.679 ± .021	.572 ± .016	.610 ± .096	.554 ± .019	.614 ± .024	.543 ± .021	.553 ± .020
	13	.654 ± .021	.815 ± .021	.765 ± .022	.764 ± .026	.573 ± .017	.739 ± .022	.572 ± .020	**.552 ± .018**
eurlex_dc	5	**.154 ± .012**	.213 ± .013	.260 ± .016	.260 ± .016	.729 ± .083	.304 ± .014	.203 ± .016	.489 ± .095
	13	**.288 ± .016**	.374 ± .013	.394 ± .017	.394 ± .017	.812 ± .102	.433 ± .016	.567 ± .017	.696 ± .131
eurlex_sm	5	**.228 ± .017**	.267 ± .017	.595 ± .008	.347 ± .016	.708 ± .058	.371 ± .014	.268 ± .015	.609 ± .137
	13	**.338 ± .009**	.404 ± .019	.843 ± .010	.464 ± .016	.815 ± .043	.471 ± .019	.611 ± .009	.821 ± .038
health	5	**.430 ± .016**	.451 ± .034	.524 ± .018	.445 ± .018	.488 ± .020	.462 ± .027	.465 ± .014	.494 ± .038
	13	**.456 ± .011**	.478 ± .025	.675 ± .017	.520 ± .017	.495 ± .016	.533 ± .013	.488 ± .017	.515 ± .080
entertainment	5	**.571 ± .021**	.610 ± .024	.910 ± .010	.689 ± .007	.672 ± .007	.602 ± .021	.618 ± .013	.657 ± .018
	13	**.665 ± .018**	.675 ± .010	.983 ± .003	.701 ± .019	.682 ± .014	.686 ± .018	.667 ± .019	.697 ± .062
eurlex_ed	5	**.394 ± .012**	.498 ± .018	.732 ± .012	.548 ± .020	.746 ± .044	.563 ± .022	.685 ± .013	.782 ± .134
	13	**.459 ± .021**	.577 ± .015	.908 ± .008	.615 ± .021	.831 ± .044	.601 ± .022	.578 ± .013	.731 ± .077
computer	5	**.390 ± .015**	.736 ± .021	.496 ± .015	.710 ± .101	.470 ± .011	.463 ± .017	.419 ± .012	.470 ± .014
	13	**.454 ± .010**	.898 ± .014	.554 ± .014	.617 ± .095	.474 ± .011	.599 ± .014	.453 ± .013	.474 ± .011
CAL500	25	.102 ± .042	.114 ± .038	.366 ± .057	.366 ± .057	.114 ± .038	**.080 ± .000**	.114 ± .040	.120 ± .042
	45	.350 ± .060	**.111 ± .044**	.366 ± .044	.366 ± .044	.116 ± .045	.200 ± .000	.114 ± .044	.130 ± .046

TABLE 6
Experimental Results of Each Learning Approach on Synthetic as well as Real-World PML Data Sets in Terms of *Coverage*, Where the Best Performance (the smaller the better) is Shown in Bold Face

Data Set	avg. #CLs	Particle-Vls	Particle-Map	Pml-lc	Pml-fp	Fpml	Pml-lrs	Ml-knn	Lift
music_emotion	5.29	.411 ± .005	**.409 ± .008**	.438 ± .004	.434 ± .004	.484 ± .006	.442 ± .006	.466 ± .007	.434 ± .005
music_style	6.04	**.203 ± .006**	.218 ± .008	.323 ± .011	.203 ± .007	.288 ± .010	.234 ± .006	.263 ± .005	.264 ± .012
mirflickr	3.35	.268 ± .005	**.207 ± .005**	.243 ± .005	.228 ± .005	.261 ± .014	.216 ± .005	.266 ± .011	.219 ± .006
yeastBP	30.43	.800 ± .075	**.419 ± .079**	.480 ± .063	.499 ± .072	.650 ± .050	.604 ± .038	.645 ± .048	.580 ± .038
emotions	3	**.305 ± .023**	.311 ± .019	.320 ± .030	.314 ± .025	.588 ± .028	.451 ± .049	.320 ± .028	.334 ± .036
	5	.366 ± .019	**.352 ± .018**	.370 ± .027	.378 ± .027	.542 ± .031	.531 ± .024	.433 ± .022	.437 ± .045
scene	3	.099 ± .013	.102 ± .013	.283 ± .019	.133 ± .009	.170 ± .018	**.093 ± .011**	.135 ± .011	.083 ± .008
	5	.170 ± .011	.177 ± .014	.368 ± .023	.214 ± .012	.356 ± .031	**.156 ± .014**	.250 ± .019	.236 ± .028
yeast	5	.465 ± .011	.473 ± .015	.498 ± .015	.780 ± .021	.489 ± .014	.652 ± .012	**.463 ± .013**	.466 ± .011
	13	**.541 ± .013**	.575 ± .019	.595 ± .016	.660 ± .012	.544 ± .017	.750 ± .021	.559 ± .011	.564 ± .026
referece	5	**.260 ± .018**	.289 ± .013	.255 ± .013	.338 ± .083	.261 ± .020	.291 ± .019	.266 ± .014	.262 ± .012
	13	.322 ± .017	.355 ± .008	.346 ± .012	.350 ± .014	.285 ± .013	.372 ± .019	.293 ± .015	**.284 ± .020**
eurlex_dc	5	**.039 ± .003**	.054 ± .005	.059 ± .004	.059 ± .004	.229 ± .034	.070 ± .004	.077 ± .009	.147 ± .056
	13	**.092 ± .005**	.116 ± .005	.131 ± .008	.131 ± .008	.307 ± .062	.152 ± .010	.191 ± .018	.268 ± .023
eurlex_sm	5	**.149 ± .008**	.162 ± .009	.460 ± .012	.192 ± .009	.364 ± .036	.194 ± .009	.174 ± .009	.312 ± .031
	13	**.222 ± .006**	.231 ± .008	.355 ± .005	.263 ± .004	.424 ± .016	.267 ± .006	.312 ± .007	.427 ± .034
health	5	.211 ± .008	.264 ± .009	.420 ± .009	.208 ± .014	.197 ± .010	.243 ± .012	**.191 ± .006**	.195 ± .008
	13	.276 ± .008	.341 ± .010	.295 ± .010	.256 ± .006	**.214 ± .011**	.296 ± .006	.238 ± .012	.249 ± .119
entertainment	5	.226 ± .010	.267 ± .010	.544 ± .013	.608 ± .010	.199 ± .006	.255 ± .008	**.193 ± .007**	.196 ± .007
	13	.293 ± .010	.330 ± .009	.431 ± .008	.604 ± .009	**.210 ± .008**	.310 ± .010	.230 ± .008	.219 ± .035
eurlex_ed	5	**.177 ± .005**	.207 ± .006	.509 ± .015	.217 ± .008	.390 ± .015	.222 ± .009	.226 ± .005	.376 ± .046
	13	**.216 ± .009**	.250 ± .010	.544 ± .014	.291 ± .012	.471 ± .023	.269 ± .009	.323 ± .012	.432 ± .028
computer	5	.174 ± .009	.291 ± .007	.378 ± .011	.435 ± .169	.180 ± .009	.256 ± .012	**.173 ± .008**	.181 ± .007
	13	.246 ± .008	.374 ± .010	.252 ± .004	.365 ± .094	.197 ± .012	.316 ± .006	.374 ± .009	**.194 ± .009**
CAL500	25	.941 ± .007	**.856 ± .018**	.943 ± .012	.943 ± .012	.882 ± .008	.865 ± .008	.885 ± .013	.878 ± .006
	45	.926 ± .011	**.860 ± .012**	.912 ± .011	.910 ± .010	.887 ± .016	.865 ± .007	.899 ± .015	.885 ± .014

TABLE 7
Experimental Results of Each Learning Approach on Synthetic as well as Real-World PML Data Sets in Terms of *Average Precision*, Where the Best Performance (the larger the better) is Shown in Bold Face

Data Set	avg. #CLs	Particle-Vls	Particle-Map	Pml-lc	Pml-fp	Fpml	Pml-lrs	Ml-knn	Lift
music_emotion	5.29	.605 ± .006	**.614 ± .007**	.564 ± .010	.567 ± .010	.520 ± .008	.562 ± .007	.554 ± .007	.569 ± .009
music_style	6.04	**.716 ± .010**	.677 ± .015	.551 ± .016	.703 ± .009	.654 ± .007	.680 ± .006	.683 ± .009	.665 ± .008
mirflickr	3.35	.690 ± .009	**.858 ± .008**	.748 ± .010	.780 ± .008	.771 ± .034	.836 ± .008	.701 ± .036	.795 ± .017
yeastBP	30.43	.086 ± .019	.158 ± .016	.141 ± .026	.143 ± .021	.095 ± .020	.086 ± .016	.110 ± .029	**.169 ± .058**
emotions	3	**.807 ± .027**	.791 ± .020	.758 ± .042	.783 ± .027	.550 ± .026	.630 ± .027	.783 ± .026	.761 ± .032
	5	**.734 ± .029**	.728 ± .023	.698 ± .016	.724 ± .038	.554 ± .040	.565 ± .019	.642 ± .034	.651 ± .030
scene	3	**.828 ± .023**	.824 ± .027	.635 ± .024	.774 ± .012	.707 ± .024	.829 ± .018	.802 ± .012	.855 ± .008
	5	.713 ± .019	**.714 ± .021**	.492 ± .024	.655 ± .020	.493 ± .044	.708 ± .025	.615 ± .023	.625 ± .032
yeast	5	.751 ± .014	.743 ± .018	.729 ± .015	.563 ± .012	.702 ± .016	.584 ± .012	**.755 ± .012**	.750 ± .016
	13	**.710 ± .016**	.655 ± .019	.549 ± .016	.527 ± .013	.689 ± .019	.454 ± .015	.685 ± .019	.677 ± .020
referece	5	**.557 ± .015**	.473 ± .012	.548 ± .009	.493 ± .087	.552 ± .018	.510 ± .018	.556 ± .016	.550 ± .018
	13	.457 ± .015	.361 ± .011	.391 ± .016	.390 ± .020	.524 ± .015	.395 ± .016	.521 ± .014	**.533 ± .015**
eurlex_dc	5	**.882 ± .008**	.849 ± .009	.821 ± .009	.821 ± .009	.468 ± .061	.793 ± .009	.844 ± .014	.645 ± .069
	13	**.762 ± .011**	.725 ± .008	.703 ± .013	.703 ± .013	.381 ± .092	.669 ± .013	.581 ± .020	.471 ± .074
eurlex_sm	5	**.783 ± .013**	.764 ± .013	.441 ± .007	.708 ± .013	.428 ± .050	.694 ± .012	.764 ± .012	.509 ± .072
	13	**.669 ± .008**	.652 ± .012	.363 ± .004	.608 ± .009	.341 ± .031	.597 ± .011	.515 ± .005	.337 ± .027
health	5	**.637 ± .011**	.491 ± .018	.479 ± .013	.645 ± .012	.621 ± .013	.622 ± .016	.481 ± .011	.618 ± .021
	13	.590 ± .009	.382 ± .014	.456 ± .011	.577 ± .010	.606 ± .010	.554 ± .008	**.605 ± .011**	.583 ± .090
entertainment	5	.538 ± .017	**.539 ± .019**	.196 ± .008	.336 ± .006	.504 ± .008	.524 ± .013	**.539 ± .011**	.513 ± .012
	13	.451 ± .011	**.501 ± .009**	.173 ± .002	.333 ± .013	.488 ± .011	.445 ± .013	.492 ± .013	.477 ± .041
eurlex_ed	5	**.633 ± .006**	.578 ± .010	.324 ± .010	.545 ± .013	.349 ± .025	.535 ± .014	.564 ± .010	.347 ± .069
	13	**.571 ± .013**	.508 ± .010	.188 ± .009	.469 ± .014	.273 ± .031	.489 ± .013	.475 ± .012	.345 ± .040
computer	5	**.653 ± .010**	.402 ± .013	.504 ± .013	.385 ± .120	.614 ± .008	.591 ± .016	.401 ± .009	.607 ± .011
	13	.576 ± .007	.262 ± .015	.507 ± .007	.454 ± .092	**.600 ± .009**	.472 ± .009	.301 ± .010	.596 ± .013
CAL500	25	.431 ± .013	.523 ± .013	.382 ± .013	.382 ± .013	.527 ± .013	**.541 ± .002**	.515 ± .015	.526 ± .015
	45	.432 ± .012	**.526 ± .012**	.403 ± .010	.403 ± .010	.522 ± .011	.505 ± .001	.501 ± .011	.523 ± .014

Furthermore, *Friedman test* [8] is utilized as the statistical test to analyze the relative performance among learning approaches. Given n learning approaches and N data sets, let r_i^j denote the rank of the jth approach on the ith data set where mean ranks are shared in case of ties. Let $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j$ denote the average rank for the jth algorithm, under the null hypothesis of all algorithms having indistinguishable performance, the following Friedman statistic F_F will be distributed according to the F-distribution with $n-1$ numerator degrees of freedom and $(n-1)(N-1)$ denominator degrees of freedom

$$F_F = \frac{(N-1)\chi_F^2}{N(n-1)-\chi_F^2}, \quad \text{where}$$
$$\chi_F^2 = \frac{12N}{n(n+1)}\left[\sum_{j=1}^{n} R_j^2 - \frac{n(n+1)^2}{4}\right].$$

For Particle-Vls and Particle-Map, Tables 8 and 9 summarize the Friedman statistics F_F and the corresponding critical value respectively in terms of each evaluation metric (# learning approaches $n = 7$, # data sets $N = 55$). As shown in Tables 8 and 9, at 0.05 significance level, the null hypothesis of indistinguishable performance among the learning approaches is clearly rejected for all evaluation metrics.

Thereafter, to show the relative performance among the learning approaches, *Bonferroni-Dunn test* [8] is employed as the post-hoc test by treating Particle-Vls or Particle-Map as the control approach. Here, the difference between the average ranks of control approach and one learning approach is calibrated with the *critical difference* (CD)

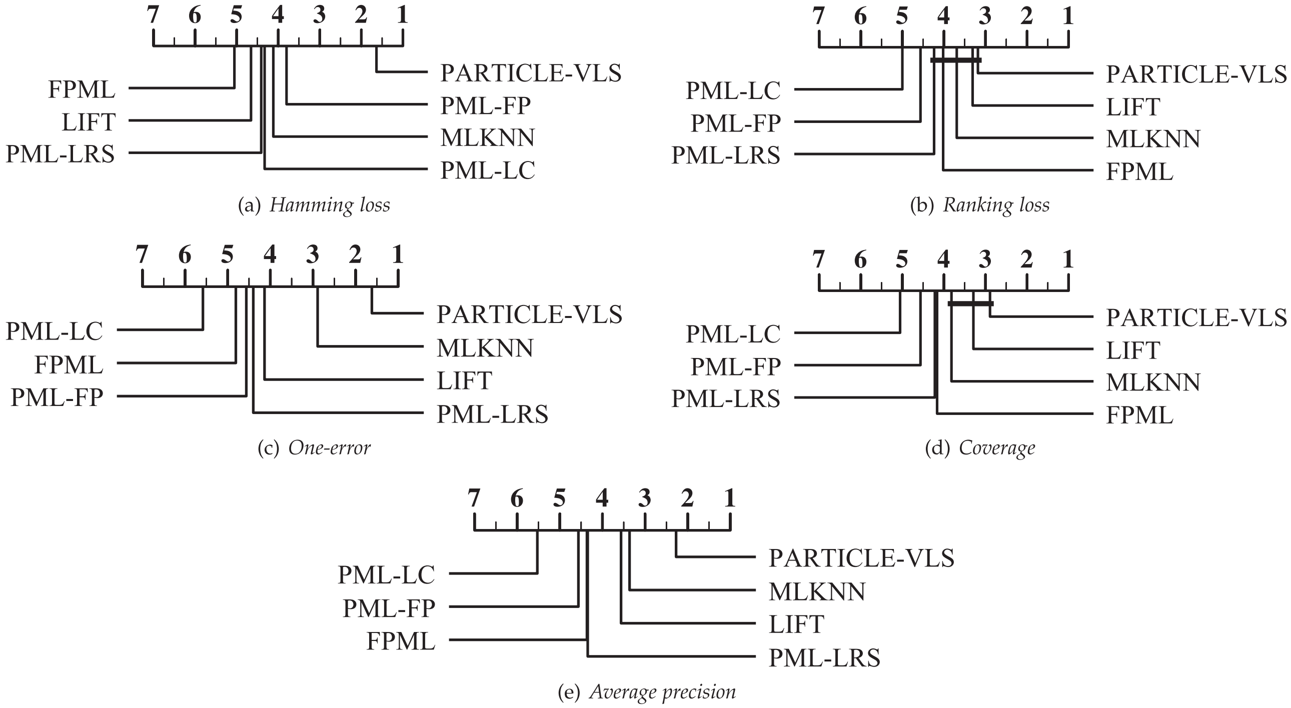
$$CD = \kappa \cdot \sqrt{\frac{n(n+1)}{6N}}. \tag{19}$$

TABLE 8
Summary of the Friedman Statistics F_F in Terms of Each Evaluation Metric and the Critical Value at 0.05 Significance Level for Particle-Vls (# Learning Approaches $n = 7$, # Data Sets $N = 55$)

Evaluation metric	F_F	critical value
Hamming loss	12.3060	2.1266
Ranking loss	4.1607	
One-error	12.8526	
Coverage	15.1603	
Average precision	9.5029	

TABLE 9
Summary of the Friedman Statistics F_F in Terms of Each Evaluation Metric and the Critical Value at 0.05 Significance Level for Particle-Map (# Learning Approaches $n = 7$, # Data Sets $N = 55$)

Evaluation metric	F_F	critical value
Hamming loss	10.4953	2.1266
Ranking loss	3.6695	
One-error	13.3240	
Coverage	14.4883	
Average precision	8.9515	



(a) *Hamming loss*

(b) *Ranking loss*

(c) *One-error*

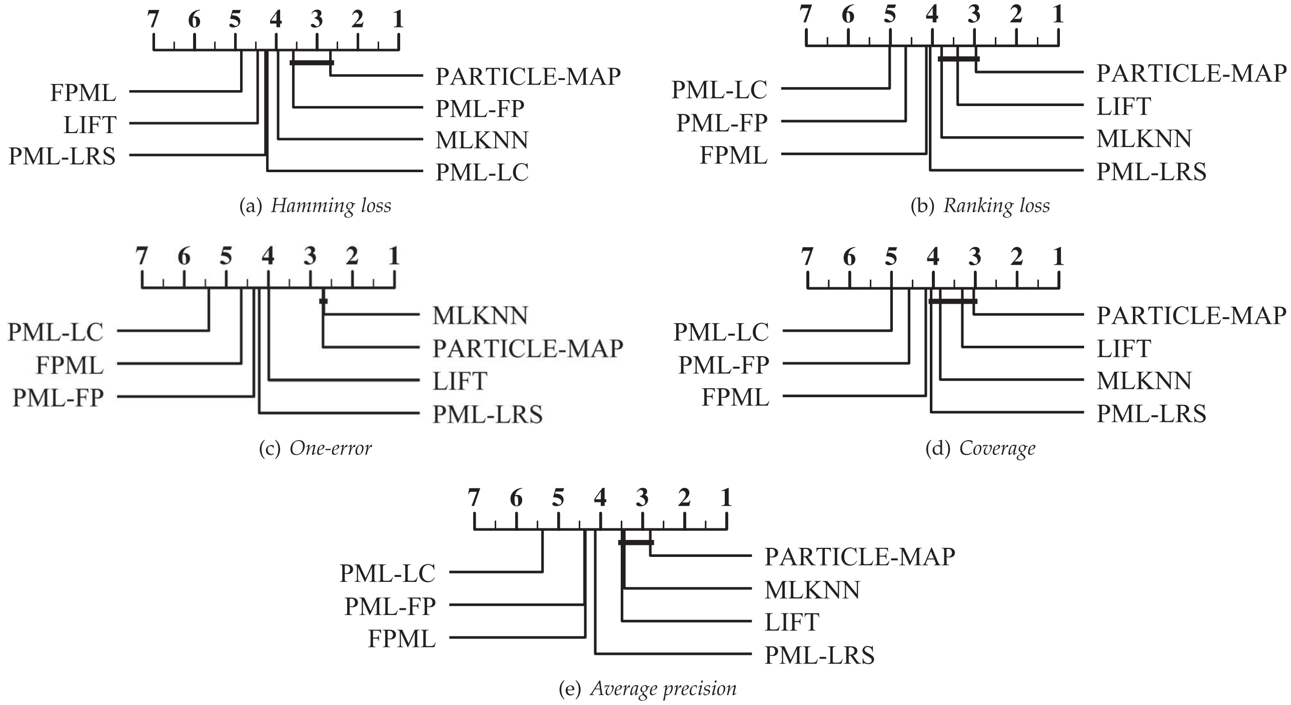
(d) *Coverage*

(e) *Average precision*

Fig. 2. Comparison of Particle-Vls (control approach) against other learning approaches with the *Bonferroni-Dunn test*. Approaches not connected with Particle-Vls in the CD diagram are considered to have significantly different performance from the control approach (CD=1.0867 at 0.05 significance level).

Here, $\kappa = 2.638$ at 0.05 significance level and thus CD = 1.0867 in this paper. Accordingly, the performance between control approach and one learning approach is deemed to be significantly different if their average ranks differ by at least one CD.

Figs. 2 and 3 illustrate the CD diagrams [8] on each evaluation metric by treating Particle-Vls or Particle-Map as the control approach respectively. Here, the average rank of each learning approach is marked along the axis with lower ranks to the right. In each subfigure, any learning approach whose average rank is within one CD to that of the control approach is interconnected to each other with a thick line. Otherwise, it is considered to have significantly different performance against the control approach.



(a) *Hamming loss*

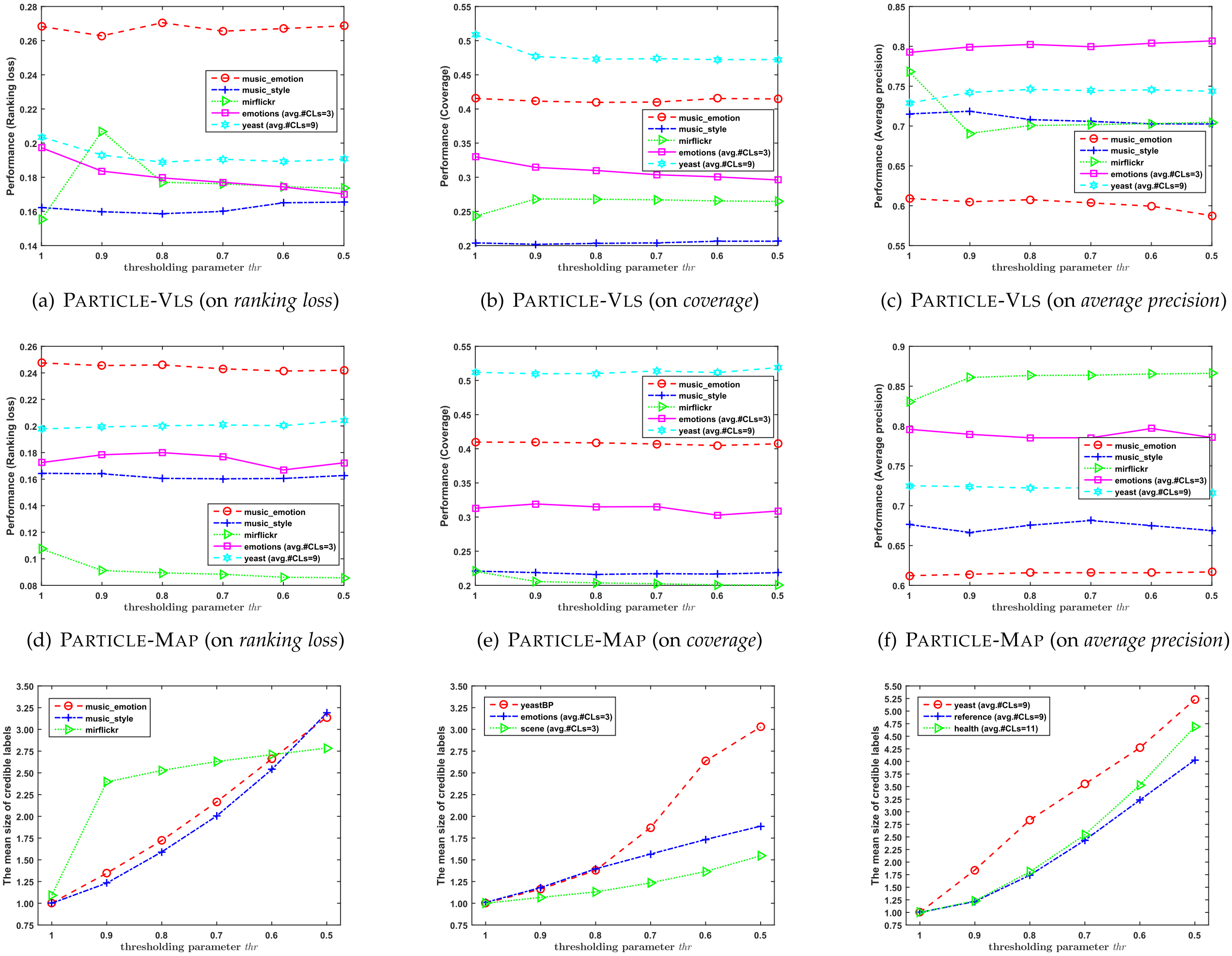
(b) *Ranking loss*

(c) *One-error*

(d) *Coverage*

(e) *Average precision*

Fig. 3. Comparison of Particle-Map (control approach) against other learning approaches with the *Bonferroni-Dunn test*. Approaches not connected with Particle-Map in the CD diagram are considered to have significantly different performance from the control approach (CD=1.0867 at 0.05 significance level).



(a) PARTICLE-VLS (on *ranking loss*)

(b) PARTICLE-VLS (on *coverage*)

(c) PARTICLE-VLS (on *average precision*)

(d) PARTICLE-MAP (on *ranking loss*)

(e) PARTICLE-MAP (on *coverage*)

(f) PARTICLE-MAP (on *average precision*)

(g) mean size of credible labels elicited by PARTICLE (on music_emotion, music_style, mirflickr)

(h) mean size of credible labels elicited by PARTICLE (on yeastBP, emotions and scene)

(i) mean size of credible labels elicited by PARTICLE (on yeast, reference and health)

Fig. 4. Properties of PARTICLE-VLS and PARTICLE-MAP change as parameter thr varies from 1 to 0.5 with an interval of 0.1. **First row**: the predictive performance of PARTICLE-VLS on *ranking loss*, *coverage*, and *average precision*; **Second row**: the predictive performance of PARTICLE-MAP on *ranking loss*, *coverage*, and *average precision*; **Third row**: the mean size of credible labels elicited by PARTICLE.

Based on the reported experimental results, the following observations of the comparative studies can be made:

- As shown in Fig. 2, it is impressive that PARTICLE-VLS has the lowest average rank on each evaluation metric. Regarding the tailored multi-label learning approaches, the performance of PARTICLE-VLS is statistically comparable to ML-KNN and LIFT on *ranking loss* and *coverage*, and superior to both of them in the rest cases. Regarding the existing PML approaches, the performance of PARTICLE-VLS is statistically comparable to FPML and PML-LRS on *ranking loss*, and superior to PML-LC, PML-FP, FPML and PML-LRS in the rest cases.
- As shown in Fig. 3, PARTICLE-MAP has the second lowest average rank on *one-error* and has the lowest average rank on the other evaluation metrics. Regarding the tailored multi-label learning approaches, the performance of PARTICLE-MAP is statistically superior to ML-KNN on *hamming loss*, superior to LIFT on *hamming loss* and *one-error*, and comparable to ML-KNN and LIFT in the rest cases. Regarding the existing PML approaches, the performance of PARTICLE-MAP is statistically comparable to PML-FP on *hamming loss*, comparable to PML-LRS on *coverage*, and superior to PML-LC, PML-FP, FPML and PML-LRS in the rest cases.
- As shown in Tables 3, 4, 5, 6, and 7, on the four real-world PML data sets music_emotion, music_style, mirflickr and yeastBP, the two variants of PARTICLE achieve optimal performance in almost all cases (except on music_style where PML-FP outperforms PARTICLE on *ranking loss*, on yeastBP where LIFT outperforms PARTICLE on *average precision*). Furthermore, the performance advantage of PARTICLE is more pronounced on synthetic PML data sets with a moderate number of features including emotions, yeast, eurlex_dc, eurlex_sm, eurlex_ed and CAL500. One potential reason lies in that the weighted kNN graph constructed by PARTICLE would be more reliable in low-dimensional feature space, which coincides with the observation that dimensionality reduction works well for learning with noisy labeling information [28].

TABLE 10
Summary of Algorithmic Complexity of Each Learning Approach

Learning Approach	Algorithmic Complexity	
	Train	Test
PARTICLE-VLS	$\mathcal{O}\left(m^2(d+k+Tq)+m\cdot\mathcal{F}_{\mathrm{QP}}(m,m)+q^2(\mathcal{F}_{\mathcal{B}}(m,d)+m\cdot\mathcal{F}'_{\mathcal{B}}(d))\right)$	$\mathcal{O}\left(q^2\cdot\mathcal{F}'_{\mathcal{B}}(d)\right)$
PARTICLE-MAP	$\mathcal{O}\left(m^2(d+k+Tq)+m\cdot\mathcal{F}_{\mathrm{QP}}(m,m)+q^2(\mathcal{F}_{\mathcal{B}}(m,d)+mk\cdot\mathcal{F}'_{\mathcal{B}}(d))\right)$	$\mathcal{O}\left(mq(d+k+q)+q^2k\cdot\mathcal{F}'_{\mathcal{B}}(d)\right)$
PML-LC	$\mathcal{O}\left(q^2m+T(\mathcal{F}_{\mathrm{QP}}(dq,mq^2)+\mathcal{F}_{\mathrm{QP}}(mq,mq))\right)$	$\mathcal{O}\left(dq\right)$
PML-FP	$\mathcal{O}\left(mdq+T(\mathcal{F}_{\mathrm{QP}}(dq,mq^2)+\mathcal{F}_{\mathrm{QP}}(mq,mq))\right)$	$\mathcal{O}\left(dq\right)$
FPML	$\mathcal{O}\left(T(\mathcal{F}_{\mathrm{NMF}}(q,n,k')+\mathcal{F}_{\mathrm{AGD}}(dq))\right)$	$\mathcal{O}\left(dq\right)$
PML-LRS	$\mathcal{O}\left(T(dq^2+d^2(m+d+q))\right)$	$\mathcal{O}\left(dq\right)$
ML-KNN	$\mathcal{O}\left(m^2d+qmk\right)$	$\mathcal{O}\left(md+qk\right)$
LIFT	$\mathcal{O}\left(q(\mathcal{F}_{\mathrm{KM}}(m,d,K)+mdk+\mathcal{F}_{\mathcal{B}}(m,K))\right)$	$\mathcal{O}\left(q(dk+\mathcal{F}'_{\mathcal{B}}(K))\right)$

In summary, these results clearly validate the effectiveness of the two-stage credible label elicitation strategy for learning from PML examples.

4.3 Further Analysis

4.3.1 Parameter Sensitivity

As shown in Table 1, thr serves as the crucial parameter which controls the number of credible labels elicited in the first stage. Fig. 4 gives an illustrative example on how the properties of PARTICLE change as the value of parameter thr varies, including the predictive performance in terms of *ranking loss*, *coverage* and *average precision* (first and second rows) and the mean size of credible labels (third row).

As shown in Figs. 4a, 4b, 4c, 4d, 4e, and 4f, the predictive performance of PARTICLE becomes relatively stable as thr decrease to 0.9, which is the value used in this paper. Furthermore, as shown in Figs. 4g, 4h, and 4i, the mean size of elicited credible labels increases in linear or sub-linear rate as thr decreases.

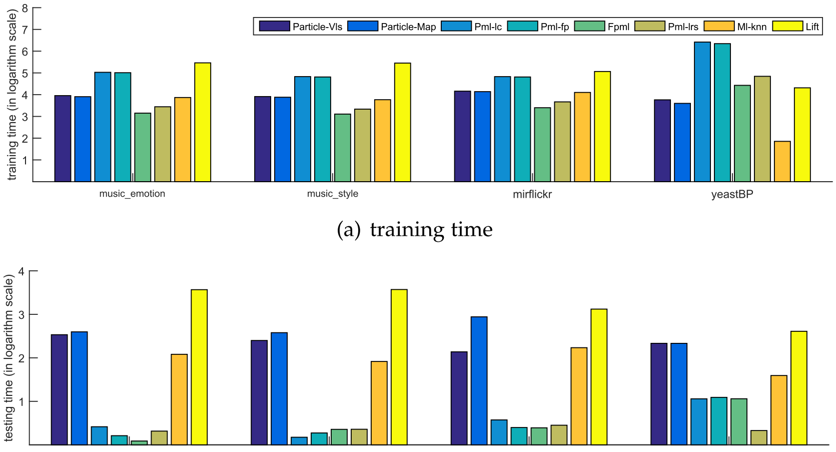
4.3.2 Algorithmic Complexity

Table 10 summarizes the (worst-case) algorithmic complexity of each learning approach considered in this paper w.r.t. several common factors, i.e., m (# training examples), d (# features), q (# class labels) and T (# iterations). Furthermore, to account for specific algorithmic components employed by each learning approach, the following terms are introduced in complexity characterization: a) $\mathcal{F}_{\mathcal{B}}(m,d)$ and $\mathcal{F}'_{\mathcal{B}}(d)$ represent the training and testing complexity for binary learning algorithm $\mathcal{B}$ respectively; b) $\mathcal{F}_{QP}(a,b)$ represents the time complexity of solving a QP problem with a variables and b constraints; c) $\mathcal{F}_{NMF}(a,b,k')$ represents the time complexity of solving the non-negative matrix factorization problem $\mathbf{V}=\mathbf{AB}$ with $\mathbf{V}$, $\mathbf{A}$ and $\mathbf{B}$ being the $a\times b$, $a\times k'$ and $k'\times b$ non-negative matrices respectively, and $\mathcal{F}_{\mathrm{AGD}}(a)$ represents the time complexity of minimizing the a-variate function based on accelerated gradient descent techniques; d) $\mathcal{F}_{\mathrm{KM}}(m,d,K)$ represents the time complexity of invoking the K-Means clustering procedure over m d-dimensional feature vectors.

Furthermore, Fig. 5 illustrates the training time and testing time of each learning approach on the four real-world PML data sets. Generally, the empirical training time of PARTICLE is relatively comparable to the other learning approaches. On the other hand, the empirical testing time of PARTICLE is higher than the other four PML learning approaches due to its quadratic complexity w.r.t. q.

5 CONCLUSION

Partial multi-label learning is an emerging weakly supervised learning framework which considers a specific setting of inaccurate supervision, where each example is associated



(a) training time

(b) testing time

Fig. 5. Running time (train/test) of each learning approach on the four real-world PML data sets (for histogram illustration, the y-axis corresponds to the value of $\log t + 2$ with t being the running time measured in seconds).

with a set of candidate labels which are only partially valid. In this paper, an extension to our earlier research [9] is presented which tackles PML problem by eliciting credible labels from the candidate label set. In this way, the negative impact of false positive labels on model induction is mitigated by exploiting credible labeling information. Extensive comparative studies against state-of-the-art approaches show that credible label elicitation serves as an effective strategy to deal with the labeling noise in PML examples.

The PARTICLE approach proposed in this paper corresponds to one feasible implementation towards credible label elicitation and utilization with iterative label propagation and pairwise label ranking.[4] In the future, it is interesting to investigate other ways to elicit credible labeling information and induce predictive model with credible labels. Furthermore, the effectiveness of disambiguating the candidate label set of PML training example can be enhanced by trying to leverage auxiliary information such as domain knowledge [39], multi-view representation [6], etc.

ACKNOWLEDGMENTS

The authors would like to thank the associate editor and anonymous reviewers for their insightful comments and suggestions. They would also like to thank Dr. Guoxian Yu in helping share the `yeastBP` data set. This work was supported by the National Key R&D Program of China (2018YFB1004300), the National Science Foundation of China (61573104), and was also supported in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.

[2] S. Burkhardt and S. Kramer, "Online multi-label dependency topic models for text classification," *Mach. Learn.*, vol. 107, no. 5, pp. 859–886, 2018.

[3] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[4] C.-H. Chen, V. M. Patel, and R. Chellappa, "Learning from ambiguously labeled face images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 7, pp. 1653–1667, Jul. 2018.

[5] Y.-C. Chen, V. M. Patel, R. Chellappa, and P. J. Phillips, "Ambiguously labeled learning using dictionaries," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2076–2088, Dec. 2014.

[6] Z.-S. Chen, X. Wu, Q.-G. Chen, Y. Hu, and M.-L. Zhang, "Multi-view partial multi-label learning with graph-based disambiguation," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020.

[7] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *J. Mach. Learn. Res.*, vol. 12, no. 4, pp. 1501–1536, 2011.

[8] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, 2006.

[9] J.-P. Fang and M.-L. Zhang, "Partial multi-label learning via credible label elicitation," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 3518–3525.

[10] J. Fürnkranz, E. Hüllermeier, E. LozaMencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Mach. Learn.*, vol. 73, no. 2, pp. 133–153, 2008.

[11] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Comput. Surv.*, vol. 47, no. 3, 2015, Art. no. 52.

[12] C. Gong, T. Liu, Y. Tang, J. Yang, J. Yang, and D. Tao, "A regularization approach for instance-based superset label learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 967–978, Mar. 2018.

[13] S. He, K. Deng, L. Li, S. Shu, and L. Liu, "Discriminatively relabel for partial multi-label learning," in *Proc. 19th IEEE Int. Conf. Data Mining*, 2019, pp. 280–288.

[14] S.-J. Huang, G. Li, W.-Y. Huang, and S.-Y. Li, "Incremental multi-label learning with active queries," *J. Comput. Sci. Technol.*, vol. 35, no. 2, pp. 234–246, 2020.

[15] M. J. Huiskes and M. S. Lew, "The MIR Flickr retrieval evaluation," in *Proc. 1st ACM Int. Conf. Multimedia Inf. Retrieval*, 2008, pp. 39–43.

[16] L. Jie and F. Orabona, "Learning from candidate labeling sets," in *Proc. 23rd Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1504–1512.

[17] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1837–1845.

[18] L. Liu and T. Dietterich, "A conditional multinomial mixture model for superset label learning," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 557–565.

[19] X. Pan, Y.-X. Fan, J. Jia, and H.-B. Shen, "Identifying RNA-binding proteins using multi-label deep learning," *Sci. China Inf. Sci.*, vol. 62, 2019, Art. no. 19103.

[20] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, 2011.

[21] L. Sun, S. Feng, T. Wang, C. Lang, and Y. Jin, "Partial multi-label learning by low-rank and sparse decomposition," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 5016–5023.

[22] L. Sun, H. Ge, and W. Kang, "Non-negative matrix factorization based modeling and training algorithm for multi-label learning," *Front. Comput. Sci.*, vol. 13, no. 6, pp. 1243–1254, 2019.

[23] Y.-Y. Sun, Y. Zhang, and Z.-H. Zhou, "Multi-label learning with weak label," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 593–598.

[24] Q. Tan, G. Yu, C. Domeniconi, J. Wang, and Z. Zhang, "Multi-view weak-label learning based on matrix completion," in *Proc. SIAM Int. Conf. Data Mining*, 2018, pp. 450–458.

[25] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multi-label classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

[26] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, and G. Chen, "Discriminative and correlative partial multi-label learning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3691–3697.

[27] T. Wei, L.-Z. Guo, Y.-F. Li, and W. Gao, "Learning safe multi-label prediction for weakly labeled data," *Mach. Learn.*, vol. 107, no. 4, pp. 703–725, 2018.

[28] J.-H. Wu and M.-L. Zhang, "Disambiguation enabled linear discriminant analysis for partial label dimensionality reduction," in *Proc. 25th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2019, pp. 416–424.

[29] X. Wu and M.-L. Zhang, "Towards enabling binary decomposition for partial label learning," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2868–2974.

[30] M.-K. Xie and S.-J. Huang, "Partial multi-label learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4302–4309.

[31] F. Yu and M.-L. Zhang, "Maximum margin partial label learning," *Mach. Learn.*, vol. 106, no. 4, pp. 573–593, 2017.

[32] G. Yu *et al.*, "Feature-induced partial multi-label learning," in *Proc. 18th IEEE Int. Conf. Data Mining*, 2018, pp. 1398–1403.

[33] M.-L. Zhang, Y.-K. Li, Y.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: An overview," *Front. Comput. Sci.*, vol. 12, no. 2, pp. 191–202, 2018.

[34] M.-L. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 107–120, Jan. 2015.

[35] M.-L. Zhang, F. Yu, and C.-Z. Tang, "Disambiguation-free partial label learning," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2155–2167, Oct. 2017.

[36] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[37] M. L. Zhang and Z. H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

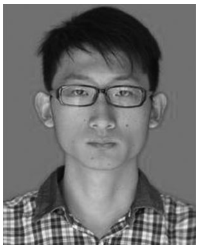
[38] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2018.

[4] Code: http://palm.seu.edu.cn/zhangml/files/PARTICLE.rar

[39] Z.-H. Zhou, “Abductive learning: Towards bridging machine learning and logical reasoning,” *Sci. China Inf. Sci.*, vol. 62, no. 7, 2019, Art. no. 076101.

[40] Z.-H. Zhou and M.-L. Zhang, “Multi-label learning,” in *Encyclopedia of Machine Learning and Data Mining*, 2nd Edition, C. Sammut and G. I. Webb, Eds., Berlin, Germany: Springer, 2017.

Min-Ling Zhang (Senior Member, IEEE) received the BSc, MSc, and PhD degrees in computer science from Nanjing University, China, in 2001, 2004 and 2007, respectively. Currently, he is a professor with the School of Computer Science and Engineering, Southeast University, China. His main research interests include machine learning and data mining. In recent years, he has served as the general co-chairs of ACML’18, program co-chairs of PAKDD’19, CCF-ICAI’19, ACML’17, CCFAI’17, PRICAI’16, senior PC member or area chair of AAAI 2017-2020, IJCAI 2017-2020, ICDM 2015-2019, etc. He is also on the editorial board of the *ACM Transactions on Intelligent Systems and Technology*, the *Neural Networks*, the *Science China Information Sciences*, the *Frontiers of Computer Science*, etc. He is the steering committee member of ACML and PAKDD, secretary-general of the CAAI Machine Learning Society, standing committee member of the CCF Artificial Intelligence & Pattern Recognition Society. He is also a distinguished member of CCF, CAAI, and a senior member of ACM.



Jun-Peng Fang received the BSc degree in computer science from North China Electric Power University, China, in 2017. Currently, he is working toward the master’s degree at Southeast University, China. His main research interests include machine learning and data mining, especially in learning from multi-label data.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.