Partial Multi-Label Learning via Credible Label Elicitation

Jun-Peng Fang,[1,2] **Min-Ling Zhang**[1,2,3,*]
[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
[3]Collaborative Innovation Center of Wireless Communications Technology, China
fangjp@seu.edu.cn, zhangml@seu.edu.cn* (corresponding author)

Abstract

In partial multi-label learning (PML), each training example is associated with multiple candidate labels which are only partially 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. To learn from PML training examples, the training procedure is prone to be misled by the false positive labels concealed in candidate label set. In light of this major difficulty, a novel two-stage PML approach is proposed which works by eliciting credible labels from the candidate label set for model induction. In this way, most false positive labels are expected to be excluded from the training procedure. Specifically, 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 with virtual label splitting or maximum a posteriori (MAP) reasoning. Extensive experiments on synthetic as well as real-world data sets clearly validate the effectiveness of credible label elicitation in learning from PML examples.

Introduction

Partial multi-label learning deals with one particular learning framework with inaccurate supervision, where multiple candidate labels are assigned to each training example which are only partially valid. The need to learn from PML examples naturally arises in many real-world scenarios, where accurate supervision information is difficult to be obtained from the collected data (Zhou 2018; Xie and Huang 2018). For instance, in crowdsourcing image tagging (Figure 1), among the set of candidate labels given by crowdsourcing annotators only some of them are valid ones due to potential unreliable annotators. The task of partial multi-label learning is to learn a multi-label predictor from PML training examples which can assign a set of proper labels for the unseen instance.

Formally, let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimensional feature space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ denote the output space with q possible class labels. Furthermore, 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 of partial multi-label learning lies in that the ground-truth labels $\widetilde{Y}_i \subseteq \mathcal{Y}$ 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 $\mathcal{D}$.

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

A straightforward strategy to learn from PML examples is to treat all the candidate labels in Y_i as ground-truth ones, and then apply off-the-shelf multi-label learning algorithms (Zhang and Zhou 2014; Gibaja and Ventura 2015) to induce the desired multi-label predictor. Obviously, the resulting multi-label training procedure will be significantly affected by the labeling noise brought by false positive labels in Y_i. One recent attempt towards PML works by utilizing the confidence of each candidate label being the ground-truth one (Xie and Huang 2018), where confidence scores and predictive model are optimized in an alternative manner by minimizing the confidence-weighted ranking loss between candidate and non-candidate labels. Nonetheless, the estimated confidence scores would be error-prone especially when the proportion of false positive labels is high, which in turn will impact the predictive model due to the alternative optimization procedure.

To deal with the major difficulty that ground-truth labels

are concealed in the candidate label set of PML training examples, a novel approach named PARTICLE, i.e. *PARTIal multi-label learning via Credible Label Elicitation*, is proposed in this paper. 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 with virtual label splitting or maximum a posteriori reasoning. Comprehensive experimental studies show that credible label elicitation serves as an effective strategy to solve the partial multi-label learning problem.

The rest of this paper is organized as follows. Firstly, related works on partial multi-label learning are briefly discussed. Secondly, technical details of the proposed PARTICLE approach are presented. Thirdly, detailed experimental results are reported. Finally, we conclude this paper.

Related Work

Partial multi-label learning is closely related to two popular learning frameworks, namely *multi-label learning* (Zhang and Zhou 2014; Gibaja and Ventura 2015; Zhou and Zhang 2017) and *partial label learning* (Cour, Sapp, and Taskar 2011; Liu and Dietterich 2012; Zhang, Yu, and Tang 2017).

In multi-label learning (MLL), each example is associated with multiple valid labels simultaneously. Based on the order of correlations being exploited for model training, existing MLL approaches can be roughly categorized into three groups including *first-order approaches* (Boutell et al. 2004; Zhang et al. 2018), *second-order approaches* (Fürnkranz et al. 2008; Li, Song, and Luo 2017), and *high-order approaches* (Read et al. 2011; Tsoumakas, Katakis, and Vlahavas 2011; Burkhardt and Kramer 2018). Both MLL and PML aim to induce the predictive model which can assign proper label set 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* (Sun, Zhang, and Zhou 2010; Tan et al. 2018; Wei et al. 2018) which considers the case of missing ground-truth labels w.r.t. the associated label set. 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 negative labels within candidate label set.

In partial label learning (PLL), 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 (Cour, Sapp, and Taskar 2011; Liu and Dietterich 2012; Yu and Zhang 2017; Gong et al. 2018; Chen, Patel, and Chellappa 2018) or transforming partial label learning problem into canonical supervised learning problems (Chen et al. 2014; Zhang, Yu, and Tang 2017; Wu and Zhang 2018). Both PLL and PML learn from training examples with labeling noise 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 single-label predictor needs to be induced from PML training examples.

To solve the partial multi-label learning problem, one most straightforward strategy is to treat all candidate labels as ground-truth ones. In this way, any off-the-shelf multi-label learning algorithms can be applied to induce the desired multi-label predictor. Nevertheless, it is obvious that this straightforward strategy tends to suffer from the false positive labels concealed in candidate label set. Another recent strategy (Xie and Huang 2018) learns from PML examples by estimating the ground-truth labeling confidence of each candidate label, where the estimated confidence scores are incorporated into an alternative optimization procedure for model induction. Due to the alternative nature of optimization, estimation errors on confidence scores will keep impairing the coupled predictive model, especially when the proportion of false positive labels 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 focusing on reliable labeling information.

The PARTICLE Approach

Credible Label Elicitation

In the first stage, PARTICLE elicits credible labels from the candidate label set via an iterative label propagation procedure. 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 and $E = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid i \in \mathcal{N}(\boldsymbol{x}_j), 1 \leq j \leq m\}$ with $\mathcal{N}(\boldsymbol{x}_j)$ being the index set of $\boldsymbol{x}_j$'s k nearest neighbors in $\mathcal{D}$.

For the weight matrix $\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m]^\top$, the weight vector $\boldsymbol{w}_j = [w_{1,j}, w_{2,j}, \ldots, w_{m,j}]^\top$ $(1 \leq j \leq m)$ is optimized by solving the following minimum error reconstruction problem:

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

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 quadratic programming (QP) solver.

Let $\mathbf{H} = \mathbf{W}\mathbf{D}^{-1}$ be the propagation matrix by normalizing the columns of $\mathbf{W}$, where $\mathbf{D} = \mathrm{diag}[d_1, d_2, \ldots, d_m]$ is the diagonal matrix with $d_j = \sum_{i=1}^{m} w_{i,j}$. Furthermore, let $\mathbf{F} = [f_{i,c}]_{m \times q}$ be an $m \times q$ matrix with non-negative entries where $f_{i,c} \geq 0$ is assumed to represent the confidence of y_c being a valid label for $\boldsymbol{x}_i$. Based on PML training examples,

the initial labeling confidence matrix $\mathbf{F}^{(0)}$ is configured as:

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

Therefore, the initial labeling confidence is evenly distributed over the candidate label set. For the t-th 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 labeling information inherited from iterative propagation and the initial labeling confidence $\mathbf{F}^{(0)}$. After that, $\widehat{\mathbf{F}}^{(t)}$ will be re-scaled into $\mathbf{F}^{(t)}$ by normalizing each row w.r.t. the candidate label set:

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

Let $\mathbf{F}^{*}$ denote the final labeling confidence matrix when the iterative label propagation procedure terminates[1], it is feasible to elicit credible labels for each PML training example by identifying candidate labels with high labeling confidence w.r.t $\mathbf{F}^{*}$.

Nonetheless, to reduce the risk of overfitting with label propagation, PARTICLE fulfills the elicitation task by further performing kNN aggregation. For $\boldsymbol{x}_j$ and its k nearest neighbors in $\mathcal{N}(\boldsymbol{x}_j)$, the aggregation weight vector $\boldsymbol{\omega}^j = [\omega_1^j, \omega_2^j, \ldots, \omega_m^j]^{\top}$ is set as:

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

Here, $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ calculates the Euclidean distance between $\boldsymbol{x}_j$ and its neighboring example $\boldsymbol{x}_i$. Then, 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 $\boldsymbol{\omega}^j$:

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

Thereafter, the credible label set $Y_j^{\mathcal{C}}$ for $\boldsymbol{x}_j$ is determined by thresholding $\boldsymbol{\lambda}^j$:[2]

$$Y_j^{\mathcal{C}} = \{y_l \mid \lambda_l^j \ge thr,\ y_l \in Y_j\} \bigcup \{y_{l^*} \mid y_{l^*} = \operatorname*{argmax}_{y_l \in Y_j} \lambda_l^j\} \tag{7}$$

In other words, $Y_j^{\mathcal{C}} \subseteq Y_j$ is formed by credible labels whose labeling confidence are greater than the thresholding parameter $thr \in [0,1]$. The one with highest labeling confidence (i.e. y_{l^*}) also belongs to $Y_j^{\mathcal{C}}$ so as to avoid the potential case of empty credible label set.

[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 $\lambda_l^j = \frac{\lambda_l^j - \min_{1\le l\le q} \lambda_l^j}{\max_{1\le l\le q} \lambda_l^j - \min_{1\le l\le q} \lambda_l^j}$.

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
α:	balancing parameter
thr:	thresholding parameter
$\mathcal{B}$:	binary training algorithm
$mode$:	*virtual label splitting* or *MAP reasoning*
$\boldsymbol{x}$:	unseen instance

Outputs:

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

Process:

Instantiate the weighted graph $G = (V, E, \mathbf{W})$ by solving Eq.(1) with kNN minimum error reconstruction;
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);
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));
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);
Induce binary classifier $g_{uz} \leftarrowtail \mathcal{B}(\mathcal{D}_{uz}^{\mathcal{C}})$;
switch $mode$ **do**
  case *virtual label splitting*
    For each label y_u $(1 \le u \le q)$, generate binary training set $\mathcal{D}_{uV}^{\mathcal{C}}$ according to Eq.(9);
    Induce binary classifier $g_{uV} \leftarrowtail \mathcal{B}(\mathcal{D}_{uV}^{\mathcal{C}})$;
    Return $Y = f(\boldsymbol{x})$ according to Eq.(12) (together with Eq.(10) and Eq.(11));
  case *MAP reasoning*
    For each label y_u $(1 \le u \le q)$, set the counting statistic C_u according to Eq.(13);
    Return $Y = f(\boldsymbol{x})$ according to Eq.(14) (together with Eqs.(15)-(18));
end switch

Predictive Model Induction

In the second stage, PARTICLE aims to induce the multi-label predictive model by utilizing credible labels elicited in the first stage.

Specifically, let $\mathcal{D}^{\mathcal{C}} = \{(\boldsymbol{x}_i, Y_i^{\mathcal{C}}) \mid 1 \le i \le m\}$ 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 the transformed PML training examples with virtual label splitting or maximum a posteriori (MAP) reasoning, where similar techniques have been successfully applied to learn from multi-label data (Zhang and Zhou 2007; Fürnkranz et al. 2008; Zhang and Zhou 2014; Gibaja and Ventura 2015).

The basic idea of pairwise label ranking is to transform the original learning problem into a number of pairwise comparison problems, one for each label pair (y_u, y_z) $(1 \le u < z \le q)$. For each transformed PML training example

Table 2: Characteristics of the PML experimental data sets. 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.

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
image	2,000	294	5	2, 3, 4	1.23
emotions	593	72	6	3, 4, 5	1.86
scene	2,407	294	6	3, 4, 5	1.07
yeast	2,417	103	14	9, 10, 11, 12, 13	4.23
eurlex_dc	8,636	100	15	5, 6, 7, 8, 9, 10, 11, 12, 13, 14	1.01
eurlex_sm	12,679	100	15	5, 6, 7, 8, 9, 10, 11, 12, 13, 14	1.53

$(\boldsymbol{x}_i, Y_i^{\mathcal{C}})$ with $Y_i^{\mathcal{C}} \subseteq Y_i$, let $\bar{Y}_i = \mathcal{Y} \setminus Y_i$ be the complementary set of candidate label set Y_i in $\mathcal{Y}$.

For each label pair (y_u, y_z), one binary training set is generated from $\mathcal{D}^{\mathcal{C}}$ as follows:

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

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

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

In other words, $\boldsymbol{x}_i$ will be regarded as one positive or negative training example when 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 $\mathcal{D}_{uz}^{\mathcal{C}}$.

Given the binary training algorithm $\mathcal{B}$, a total of $\binom{q}{2}$ binary classifiers $g_{uz} : \mathcal{X} \mapsto \mathbb{R}$ can be induced from $\mathcal{D}_{uz}^{\mathcal{C}}$, i.e. $g_{uz} \leftarrow\!\shortmid \mathcal{B}(\mathcal{D}_{uz}^{\mathcal{C}})$. Conceptually, for unseen instance $\boldsymbol{x}$, the binary classifier votes for y_u if $g_{uz}(\boldsymbol{x}) > 0$ and y_z otherwise. Thereafter, based on the $\binom{q}{2}$ binary classifiers, PARTICLE proceeds to predict the set of proper labels for $\boldsymbol{x}$ via virtual label splitting or MAP reasoning.

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

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

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

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

In other words, $\boldsymbol{x}_i$ will be regarded as one positive or negative training example when 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, another set of q binary classifiers $g_{uV} : \mathcal{X} \mapsto \mathbb{R}$ can be induced from $\mathcal{D}_{uV}^{\mathcal{C}}$ as well, i.e. $g_{uV} \leftarrow\!\shortmid \mathcal{B}(\mathcal{D}_{uV}^{\mathcal{C}})$. Furthermore, let r_{uz} and r_{uV} denote 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. Then, for unseen instance $\boldsymbol{x}$, the overall (weighted) votes yielded by $\binom{q}{2} + q$ classifiers 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:

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

MAP Reasoning In this case, a simple counting statistic is utilized to facilitate model prediction. For unseen instance $\boldsymbol{x}$, let C_u denote the statistic which counts the number of binary classifiers which vote for y_u on $\boldsymbol{x}$:

$$C_u = \sum\nolimits_{l=1}^{u-1} [\![g_{lu}(\boldsymbol{x}) \le 0]\!] + \sum\nolimits_{l=u+1}^{q} [\![g_{ul}(\boldsymbol{x}) > 0]\!] \tag{13}$$

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

Let H_u denote the event that y_u is a relevant label for $\boldsymbol{x}$, and $\mathbb{P}(H_u \mid C_u)$ represents the posteriori probability that H_u holds given C_u. Accordingly, $\mathbb{P}(\neg H_u \mid C_u)$ represents 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:

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

Based on Bayes theorem, we have:

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

Table 3: Experimental results of each comparing approach 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	ML-KNN	CLR	LIFT
music_emotion	5.29	.265±.008	**.253±.008**	.266±.006	.277±.008	.305±.004	.270±.007	.255±.007
music_style	6.04	.157±.002	.164±.004	.220±.046	.148±.003	.200±.005	**.153±.004**	.186±.007
mirflickr	3.35	.225±.026	**.115±.073**	.171±.042	.160±.049	.214±.057	.198±.038	.137±.047
image	2	.193±.019	.172±.018	.224±.022	.192±.017	.211±.023	.200±.013	**.156±.020**
	3	.200±.016	**.176±.013**	.286±.018	.209±.019	.247±.018	.233±.018	.201±.018
	4	.262±.016	**.236±.014**	.436±.014	.258±.014	.314±.010	.267±.012	.341±.018
emotions	3	.181±.019	.172±.018	.214±.029	.196±.017	.203±.012	.194±.016	**.166±.010**
	4	.188±.007	**.177±.012**	.209±.020	.221±.036	.255±.016	.220±.026	.230±.036
	5	.269±.034	**.252±.036**	.268±.010	.280±.018	.332±.047	.281±.018	.345±.029
scene	3	.119±.006	.104±.010	.207±.024	.151±.011	.151±.017	.188±.013	**.090±.010**
	4	.149±.012	**.134±.005**	.247±.040	.190±.010	.190±.014	.219±.008	.184±.010
	5	.233±.017	**.195±.013**	.353±.052	.248±.020	.292±.013	.279±.020	.289±.032
yeast	9	.192±.006	.214±.008	.216±.010	.187±.008	.194±.004	.203±.008	**.183±.005**
	10	**.192±.006**	.208±.012	.216±.012	.193±.008	.195±.006	.205±.010	.207±.009
	11	**.199±.006**	.224±.009	.218±.010	.202±.007	.207±.004	.219±.007	.236±.006
	12	**.217±.006**	.230±.005	.225±.008	**.217±.008**	.221±.008	.235±.008	.268±.009
	13	.244±.003	.245±.007	.250±.008	.261±.004	**.234±.004**	.259±.006	.297±.005
eurlex_dc	5	**.045±.003**	.050±.005	.062±.005	.062±.005	.083±.004	.067±.005	.142±.010
	6	**.050±.003**	.058±.002	.063±.004	.063±.004	.091±.007	.066±.006	.151±.014
	7	**.054±.005**	.063±.006	.067±.005	.067±.005	.101±.005	.078±.004	.206±.101
	8	**.053±.003**	.067±.004	.079±.002	.079±.002	.103±.004	.085±.004	.199±.044
	9	**.062±.002**	.071±.003	.089±.005	.089±.005	.117±.008	.098±.004	.206±.015
	10	**.068±.005**	.087±.007	.094±.008	.094±.008	.133±.004	.101±.005	.227±.015
	11	**.073±.003**	.082±.004	.102±.008	.102±.008	.143±.008	.112±.009	.252±.005
	12	**.093±.005**	.105±.006	.111±.005	.111±.005	.167±.004	.119±.005	.278±.005
	13	.115±.006	**.111±.007**	.140±.004	.140±.004	.211±.012	.143±.005	.292±.013
	14	.164±.007	**.151±.006**	.156±.007	.156±.007	.261±.009	.169±.008	.296±.014
eurlex_sm	5	**.102±.004**	**.102±.003**	.268±.006	.133±.005	.121±.001	.144±.005	.166±.024
	6	**.109±.001**	.111±.004	.275±.003	.141±.004	.131±.005	.158±.002	.164±.014
	7	.113±.002	**.112±.004**	.281±.005	.150±.005	.143±.002	.173±.004	.189±.008
	8	**.124±.004**	**.124±.005**	.285±.007	.160±.005	.155±.006	.178±.006	.210±.028
	9	.134±.004	**.133±.004**	.285±.007	.172±.003	.175±.009	.195±.009	.236±.006
	10	.135±.004	**.132±.002**	.271±.005	.172±.006	.188±.006	.194±.008	.255±.020
	11	.146±.002	**.145±.003**	.275±.005	.171±.005	.203±.005	.198±.003	.283±.017
	12	.160±.004	**.148±.004**	.271±.005	.196±.005	.226±.004	.212±.007	.342±.010
	13	.188±.004	**.170±.003**	.262±.006	.199±.006	.253±.004	.212±.008	.348±.010
	14	.231±.004	**.204±.006**	.242±.009	.227±.007	.330±.016	.240±.007	.369±.019

Therefore, to enable MAP reasoning it suffices to compute the four terms $\mathbb{P}(H_u)$, $\mathbb{P}(\neg H_u)$, $\mathbb{P}(C_u \mid H_u)$ and $\mathbb{P}(C_u \mid \neg H_u)$ in Eq.(15).

Specifically, the prior terms $\mathbb{P}(H_u)$ and $\mathbb{P}(\neg H_u)$ can be estimated via relative frequency counting with Laplacian smoothing:

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

Furthermore, two frequency arrays κ_u and $\bar{\kappa}_u$ each with q elements are defined as follows:

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

Here, $\delta_u(\boldsymbol{x}_i) = \sum_{l=1}^{u-1}[\![g_{lu}(\boldsymbol{x}_i) \le 0]\!] + \sum_{l=u+1}^{q}[\![g_{ul}(\boldsymbol{x}_i) > 0]\!]$ counts the number of binary classifiers which vote for y_u on training example $\boldsymbol{x}_i$. Therefore, $\kappa_u[p]$ ($\bar{\kappa}_u[p]$) records the number of training examples which have (don't have) label y_u and receive exactly p votes for y_u from all the binary classifiers.

Then, the likelihood terms $\mathbb{P}(C_u \mid H_u)$ and $\mathbb{P}(C_u \mid \neg H_u)$ can be estimated via relative frequency counting with Laplacian smoothing as well:

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

Table 1 summarizes 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 utilized to induce the multi-label predictive model via virtual label splitting (steps 7-10) or MAP reasoning (steps 11-

Table 4: Experimental results of each comparing approach 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	ML-KNN	CLR	LIFT
music_emotion	5.29	.607±.010	**.611±.011**	.574±.010	.566±.009	.553±.006	.566±.009	.592±.010
music_style	6.04	**.713±.004**	.710±.007	.612±.096	.701±.005	.683±.001	.709±.002	.674±.002
mirflickr	3.35	.671±.027	**.827±.101**	.715±.040	.744±.058	.666±.052	.667±.029	.768±.059
image	2	.790±.024	.789±.024	.736±.022	.769±.013	.763±.026	.771±.012	**.809±.019**
	3	.779±.017	**.781±.014**	.698±.016	.751±.018	.723±.017	.746±.017	.755±.015
	4	.721±.015	.723±.018	.592±.011	.701±.014	.645±.012	.713±.012	.615±.011
emotions	3	**.800±.020**	**.800±.027**	.752±.029	.781±.021	.761±.011	.784±.013	.792±.015
	4	**.803±.017**	.792±.022	.753±.027	.758±.039	.720±.015	.764±.029	.738±.041
	5	.717±.026	**.724±.041**	.664±.021	.708±.025	.645±.038	.712±.020	.626±.035
scene	3	.830±.009	.826±.013	.718±.008	.762±.015	.792±.018	.742±.019	**.853±.012**
	4	**.792±.013**	**.792±.010**	.658±.047	.715±.010	.739±.016	.712±.007	.725±.008
	5	.703±.012	**.712±.019**	.546±.031	.644±.024	.605±.019	.648±.019	.590±.032
yeast	9	**.744±.007**	.722±.007	.713±.013	.738±.011	.733±.008	.733±.007	.737±.004
	10	**.743±.007**	.720±.009	.708±.012	.730±.008	.728±.009	.731±.007	.713±.009
	11	**.738±.006**	.712±.008	.699±.014	.723±.009	.719±.006	.720±.005	.690±.009
	12	**.726±.004**	.699±.007	.686±.005	.709±.001	.705±.005	.710±.004	.659±.009
	13	**.704±.003**	.688±.001	.654±.009	.651±.004	.687±.005	.687±.005	.628±.004
eurlex_dc	5	**.883±.007**	.867±.009	.818±.006	.818±.006	.838±.005	.822±.003	.690±.008
	6	**.877±.007**	.856±.007	.823±.007	.823±.007	.830±.006	.826±.007	.672±.018
	7	**.873±.011**	.851±.013	.809±.008	.809±.008	.812±.008	.801±.010	.595±.133
	8	**.871±.009**	.844±.004	.787±.009	.787±.009	.803±.005	.792±.009	.615±.033
	9	**.857±.002**	.835±.006	.773±.008	.773±.008	.782±.010	.775±.010	.593±.036
	10	**.843±.009**	.812±.009	.772±.006	.772±.006	.749±.008	.772±.008	.550±.034
	11	**.835±.006**	.814±.007	.749±.015	.749±.015	.723±.011	.744±.013	.522±.005
	12	**.794±.010**	.771±.010	.736±.008	.736±.008	.661±.006	.739±.010	.474±.030
	13	**.764±.008**	.749±.008	.696±.010	.696±.010	.572±.015	.710±.011	.482±.025
	14	**.695±.008**	.675±.011	.653±.011	.653±.011	.475±.008	.666±.011	.473±.029
eurlex_sm	5	**.789±.005**	.779±.004	.486±.006	.707±.009	.759±.002	.704±.007	.667±.041
	6	**.777±.005**	.762±.007	.445±.004	.695±.004	.747±.009	.676±.009	.670±.017
	7	**.771±.001**	.759±.006	.417±.009	.690±.007	.732±.007	.667±.006	.652±.010
	8	**.753±.006**	.742±.006	.415±.006	.675±.009	.714±.010	.655±.011	.627±.033
	9	**.739±.006**	.729±.009	.429±.014	.661±.004	.683±.010	.636±.008	.609±.010
	10	**.736±.005**	.728±.005	.446±.008	.658±.006	.659±.006	.634±.012	.542±.032
	11	**.724±.004**	.710±.005	.444±.008	.653±.007	.627±.004	.634±.008	.489±.054
	12	**.704±.002**	.699±.005	.457±.007	.637±.006	.584±.005	.629±.010	.417±.030
	13	**.672±.006**	.665±.005	.475±.008	.607±.004	.512±.008	.612±.008	.361±.018
	14	**.610±.007**	.606±.010	.542±.025	.563±.006	.392±.020	.575±.011	.355±.044

13). In this paper, the two variants of PARTICLE instantiated with virtual label splitting and MAP reasoning are termed as PARTICLE-VLS and PARTICLE-MAP respectively.

Experiments

Experimental Setup

Data Sets To thoroughly evaluate the performance of comparing approaches, a number of synthetic as well as real-world PML data sets have been employed for experimental studies. Table 2 summarizes characteristics of the experimental data sets used in this paper.

Specifically, a synthetic PML data set is generated from one multi-label data set by adding random labeling noise. For each multi-label example, some of its irrelevant labels are randomly chosen to form the candidate label set along with its relevant labels. As shown in Table 2, six benchmark multi-label data sets (Zhang and Zhou 2014) are used to generate synthetic PML data sets, including `image`, `emotions`, `scene`, `yeast`, `eurlex_dc`, and `eurlex_sm`. For each multi-label data set, different settings are considered by varying the average number of candidate labels (avg. #CLs). Accordingly, a total of thirty-four synthetic PML data sets have been generated. Furthermore, three real-world PML data sets including `music_emotion`, `music_style` and `mirflickr` (Huiskes and Lew 2008) are also employed in this paper. For the real-world PML data set, candidate labels are collected from web users which are further examined by human labellers to specify the ground-truth labels.

Comparing Approaches Three well-established multi-label learning algorithms ML-KNN (Zhang and Zhou 2007), CLR (Fürnkranz et al. 2008), and LIFT (Zhang and Wu 2015) are employed as the comparing approaches, which are tailored to learn from PML training examples by treating all candidate labels as ground-truth ones. In addition, two recent counterpart PML algorithms named PML-LC and PML-FP (Xie and Huang 2018) are also employed as the comparing approaches, which learn from PML training examples by optimizing labeling confidence and predictive model alternatively.

Table 5: Win/tie/loss counts of pairwise t-test (at 0.05 significance level) between PARTICLE and each comparing approach.

Evaluation Metric	PARTICLE-VLS against					PARTICLE-MAP against				
	PML-LC	PML-FP	ML-KNN	CLR	LIFT	PML-LC	PML-FP	ML-KNN	CLR	LIFT
Hamming loss	**27/0/10**	**26/0/11**	**30/0/7**	**34/3/0**	**34/3/0**	**30/1/6**	**19/2/16**	**36/0/1**	**37/0/0**	**37/0/0**
One-error	**36/1/0**	**37/0/0**	**37/0/0**	**36/1/0**	**35/0/2**	**36/1/0**	**37/0/0**	**33/1/3**	**30/2/5**	**32/1/4**
Coverage	**35/1/1**	**36/0/1**	**36/1/0**	**37/0/0**	**32/0/5**	**31/1/5**	**31/1/5**	**32/0/5**	**31/1/5**	**31/0/6**
Ranking loss	**34/2/1**	**31/1/5**	**35/0/2**	**35/0/2**	**30/1/6**	**35/0/2**	**32/0/5**	**32/0/5**	**33/1/3**	**32/1/4**
Average precision	**35/1/1**	**36/0/1**	**37/0/0**	**37/0/0**	**33/0/4**	**36/1/0**	**33/0/4**	**33/1/3**	**33/0/4**	**33/0/4**
Total	**167/5/13**	**166/1/18**	**175/1/9**	**179/4/2**	**164/4/17**	**168/4/13**	**152/3/30**	**166/2/17**	**164/4/17**	**165/2/18**

Parameters suggested in respective literatures are used for the comparing approaches, and Libsvm (Chang and Lin 2011) is used as the base learner to instantiate CLR and LIFT. As shown in Table 1, parameters k (number of nearest neighbors considered), α (balancing parameter) and thr (credible label elicitation threshold) for PARTICLE are set to be 10, 0.95 and 0.9 respectively. Furthermore, Libsvm (Chang and Lin 2011) is also utilized to serve as the binary training algorithm $\mathcal{B}$ for PARTICLE.

Five popular multi-label metrics *hamming loss*, *one-error*, *coverage*, *ranking loss* and *average precision* are employed for performance evaluation, whose detailed definitions can be found in (Zhang and Zhou 2014; Gibaja and Ventura 2015). On each data set, five-fold cross-validation is performed where the mean metric value as well as standard deviation are recorded for each comparing approach.

Experimental Results

Tables 3 and 4 report the detailed experimental results of each comparing algorithm in terms of *ranking loss* and *average precision*, while similar observations can be made in terms of other evaluation metrics. For each data set and evaluation metric, pairwise t-test based on five-fold cross-validation (at 0.05 significance level) is conducted to show whether the performance of PARTICLE is significantly different to the comparing approach. Accordingly, Table 5 summarizes the resulting win/tie/loss counts over 37 data sets and 5 evaluation metrics.

Based on the experimental results of comparative studies, it is impressive to observe that:

- Out of 185 statistical tests (37 data sets $\times$ 5 evaluation metrics), PARTICLE-VLS significantly outperforms the counterpart PML approaches PML-LC and PML-FP in 90.2% and 89.7% cases, and significantly outperforms the tailored MLL approaches ML-KNN, CLR and LIFT in 94.6%, 96.7% and 94.5% cases.
- Similarly, PARTICLE-MAP significantly outperforms PML-LC and PML-FP in 90.8% and 82.1% cases, and significantly outperforms ML-KNN, CLR and LIFT in 89.7%, 88.6% and 89.1% cases.
- On the real-world PML data sets `music_emotion`, `music_style` and `mirflickr`, the two variants of PARTICLE achieve optimal performance in almost all cases (except on `music_style` where CLR outperforms PARTICLE in terms of *ranking loss*). Furthermore, the performance advantage of PARTICLE is more pronounced on synthetic PML data sets with large avg. #CLs (`yeast`, `eurlex_dc`, and `eurlex_sm`).

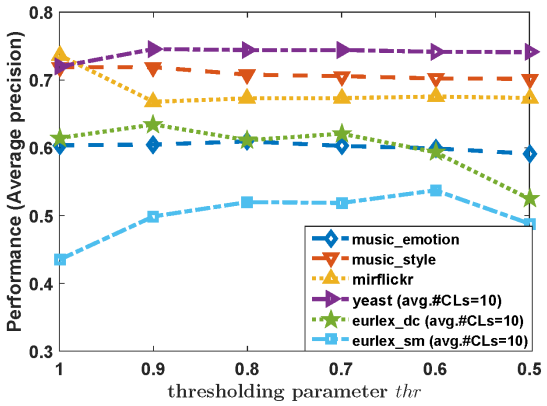

Figure 2: Performance of PARTICLE-VLS changes as parameter thr varies from 1 to 0.5 with an interval of 0.1 (in terms of *average precision*).

As shown in Table 1, thr serves as a crucial parameter which controls the amount of credible labels elicited in the first phase. Figure 2 gives an illustrative example on how the performance of PARTICLE (the virtual label splitting variant) changes as the value of parameter thr varies. It is shown that the performance of PARTICLE becomes relatively stable as thr decrease to 0.9, which is the value used in this paper.

Conclusion

The problem of partial multi-label learning is investigated in this paper, where a novel strategy based on credible label elicitation is proposed to mitigating the negative impact of false positive labels. Based on the elicited labeling information, multi-label predictive model is induced by adapting pairwise label ranking. Extensive experiments over a range of PML data sets clearly validate the effectiveness of credible label elicitation for partial multi-label learning.

Acknowledgement

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Key R&D Program of China (2018YFB1004300), the National Science Foundation of China (61573104), the Fundamental Research Funds for the Central Universities (2242018K40082), and partially supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern Recognition* 37(9):1757–1771.

Burkhardt, S., and Kramer, S. 2018. Online multi-label dependency topic models for text classification. *Machine Learning* 107(5):859–886.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chen, Y.-C.; Patel, V. M.; Chellappa, R.; and Phillips, P. J. 2014. Ambiguously labeled learning using dictionaries. *IEEE Transactios on Information Forensics and Security* 9(12):2076–2088.

Chen, C.-H.; Patel, V. M.; and Chellappa, R. 2018. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(7):1653–1667.

Cour, T.; Sapp, B.; and Taskar, B. 2011. Learning from partial labels. *Journal of Machine Learning Research* 12(May):1501–1536.

Fürnkranz, J.; Hüllermeier, E.; Loza Mencía, E.; and Brinker, K. 2008. Multilabel classification via calibrated label ranking. *Machine Learning* 73(2):133–153.

Gibaja, E., and Ventura, S. 2015. A tutorial on multilabel learning. *ACM Computing Surveys* 47(3):Article 52.

Gong, C.; Liu, T.; Tang, Y.; Yang, J.; Yang, J.; and Tao, D. 2018. A regularization approach for instance-based superset label learning. *IEEE Transactions on Cybernetics* 48(3):967–978.

Huiskes, M. J., and Lew, M. S. 2008. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, 39–43.

Li, Y.; Song, Y.; and Luo, J. 2017. Improving pairwise ranking for multi-label image classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1837–1845.

Liu, L., and Dietterich, T. 2012. A conditional multinomial mixture model for superset label learning. In Bartlett, P.; Pereira, F. C. N.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 25*. Cambridge, MA: MIT Press. 557–565.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine Learning* 85(3):333–359.

Sun, Y.-Y.; Zhang, Y.; and Zhou, Z.-H. 2010. Multi-label learning with weak label. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 593–598.

Tan, Q.; Yu, G.; Domeniconi, C.; Wang, J.; and Zhang, Z. 2018. Multi-view weak-label learning based on matrix completion. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, 450–458.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2011. Random k-labelsets for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering* 23(7):1079–1089.

Wei, T.; Guo, L.-Z.; Li, Y.-F.; and Gao, W. 2018. Learning safe multi-label prediction for weakly labeled data. *Machine Learning* 107(4):703–725.

Wu, X., and Zhang, M.-L. 2018. Towards enabling binary decomposition for partial label learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2868–2974.

Xie, M.-K., and Huang, S.-J. 2018. Partial multi-label learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 4302–4309.

Yu, F., and Zhang, M.-L. 2017. Maximum margin partial label learning. *Machine Learning* 106(4):573–593.

Zhang, M.-L., and Wu, L. 2015. Lift: Multi-label learning with label-specific features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(1):107–120.

Zhang, M. L., and Zhou, Z. H. 2007. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition* 40(7):2038–2048.

Zhang, M.-L., and Zhou, Z.-H. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26(8):1819–1837.

Zhang, M.-L.; Li, Y.-K.; Liu, Y.-Y.; and Geng, X. 2018. Binary relevance for multi-label learning: An overview. *Frontiers of Computer Science* 12(2):191–202.

Zhang, M.-L.; Yu, F.; and Tang, C.-Z. 2017. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering* 29(10):2155–2167.

Zhou, Z.-H., and Zhang, M.-L. 2017. Multi-label learning. In Sammut, C., and Webb, G. I., eds., *Encyclopedia of Machine Learning and Data Mining, 2nd Edition*. Berlin: Springer.

Zhou, Z.-H. 2018. A brief introduction to weakly supervised learning. *National Science Review* 5(1):44–53.