Discovering Latent Class Labels for Multi-Label Learning

Jun Huang\textsuperscript{1,2,*}, Linchuan Xu\textsuperscript{1}, Jing Wang\textsuperscript{2}, Lei Feng\textsuperscript{4,*} and Kenji Yamanishi\textsuperscript{1}

\textsuperscript{1}Graduate School of Information Science and Technology, The University of Tokyo.
\textsuperscript{2}School of Computer Science and Technology, Anhui University of Technology.
\textsuperscript{3}School of Computing and Mathematical Sciences, University of Greenwich.
\textsuperscript{4}School of Computer Science and Engineering, Nanyang Technological University.
\{jun_huang,linchuan_xu,jing_wang,yamanishi\}@mist.i.u-tokyo.ac.jp, feng0093@e.ntu.edu.sg

Abstract
Existing multi-label learning (MLL) approaches mainly assume all the labels are observed and construct classification models with a fixed set of target labels (known labels). However, in some real applications, multiple latent labels may exist outside this set and hidden in the data, especially for large-scale data sets. Discovering and exploring the latent labels hidden in the data may not only find interesting knowledge but also help us to build a more robust learning model. In this paper, a novel approach named D\textsubscript{LCL} (i.e., Discovering Latent Class Labels for MLL) is proposed which can not only discover the latent labels in the training data but also predict new instances with both the latent and known labels simultaneously. Extensive experiments show a competitive performance of D\textsubscript{LCL} against other state-of-the-art MLL approaches.

1 Introduction
MLL [Zhang and Zhou, 2014; Gibaja and Ventura, 2015] deals with data examples with multiple class labels simultaneously. Many well-known approaches have been proposed to solve different problems of MLL, such as partial MLL [Wang et al., 2019; Xu et al., 2019], extreme MLL [Wei et al., 2019; Jain et al., 2019; Chen et al., 2019], missing labels [Yu et al., 2014; Yang et al., 2016; Tan et al., 2018], and multi-view MLL [Xing et al., 2018; Wu et al., 2019]. It is noted that existing approaches mainly assume that all the class labels are observed in advance. However, in some applications, some latent labels might be completely unobserved and hidden in the data, and below is a summary of two possible reasons:

1. Labeling Cost. In the big data era, it is difficult to provide a complete label set for a data, especially a large-scale data with an extreme number of labels. Labelling efforts usually focus on the given set of target labels, while labels outside this set will not be considered.

2. Limitation of knowledge. For example, in medical diagnosis, possible diseases will be predicted according to the patient’s symptoms by the model constructed on the history data [Zhang et al., 2018]. However, complicated diseases may definitely exist but have not been discovered due to the limitation of human’s knowledge.

Discovering and exploring the class labels hidden in the data may not only find interesting knowledge but also improve the performance on known labels [Pham et al., 2015; Zhu et al., 2017a]. Therefore, it is important to construct a robust model for MLL which can not only discover the latent labels but also could predict new data examples with both the known and latent labels simultaneously.

Approaches have been proposed to solve data classification with an unfixed label set, such as online learning for single-label learning (SLL) [Kuzborskij et al., 2013; Nguyen et al., 2016; Zhu et al., 2017b] and MLL [Hua and Qi, 2008; Xioufis et al., 2011; Zhu et al., 2018; Zhang et al., 2020]. However, in the settings of online learning, novel labels are only induced by new instances, and they will not be discovered if they are hidden in the existing training data. MIMLNC [Pham et al., 2015] and D\textsubscript{MNL} [Zhu et al., 2017a] are two highly related studies on discovering new label(s) for multi-instance multi-label learning (MIMLL). MIMLNC is a probabilistic model to identify novel instances for IMILLL, but it assumes that all novel instances belong to a single new label. D\textsubscript{MNL} tries to discover multiple novel labels for MIMLL. It assumes that there are \( k \) novel labels, and the problem is formulated as a non-negative orthogonal constrained optimization problem which has a bag-dependent loss term and a bag-independent clustering regularization term. However, these two approaches cannot be applied to general single-instance MLL problems directly.

In this paper, a novel approach named D\textsubscript{LCL} is proposed for MLL which can not only discover the latent labels in the training data but also predict new instances with both the latent and known labels. On the one hand, we try to improve the performance of known labels by exploring the information provided by the discovered latent labels. On the other hand, we exploit the knowledge of known labels to guide the discovery of latent labels.

2 Proposed Method
Let \( \mathcal{X} \subset \mathbb{R}^d \) be the feature space, and \( \hat{\mathcal{Y}} = \mathcal{Y} \cup \tilde{\mathcal{Y}} \) be the label set, where \( \mathcal{Y} = \{y_1, \ldots, y_q\} \) and \( \tilde{\mathcal{Y}} = \{y_{q+1}, \ldots, y_{q+k}\} \) indicate the
observed and latent labels respectively. \( X \in \mathbb{R}^{n \times d} \) indicates the data matrix, and \( Y = [ Y, \hat{Y} ] \in \{ 0, 1 \}^{n \times l} \) represents the label matrix, where \( l = q + k \). \( Y \in \{ 0, 1 \}^{n \times q} \) and \( \hat{Y} \in \{ 0, 1 \}^{n \times k} \) indicates the observed and latent label matrices respectively. If \( x_i \) belongs to \( y_j \), then \( y_{ij} = 1 \); otherwise \( y_{ij} = 0 \).

2.1 Discovering Latent Class Labels

Problem Definition (Discovering Latent Class Labels for MLL). Given an MLL data set with \( q \) known labels, the problem of discovering latent class labels for MLL is to detect previously unknown labels (e.g., \( k \) latent labels) for each instance in the training set, and build a model which can predict unseen data examples with both the known and latent labels.

The aim is to construct a MLL model \( h : \mathcal{X} \to \mathcal{Y} \), but \( \hat{Y} \) is unknown at first. Motivated by previous work on clustering based matrix factorization [Hu and Chen, 2019], we try to learn an approximate representation \( U \in \{ 0, 1 \}^{n \times l} \) for the completed label matrix \( Y = [ Y, \hat{Y} ] \). Since \( Y \) is known in advance, and thus the results of the first \( q \) columns of \( U \) should be consistent with that of known class labels. Therefore, the optimization problem can be defined as

\[
\min_{U, V} \frac{1}{2} \| X - UV \|_F^2 + \frac{\lambda_1}{2} \| U - \hat{Y} \|_F^2 \tag{1}
\]

\[
\text{s.t. } U \in \{ 0, 1 \}^{n \times l}
\]

where \( V \in \mathcal{R}^{l \times d} \) is the coefficient, and \( P \in \mathcal{R}^{l \times q} \) is a projection matrix which is composed of the first \( q \) columns of an \( l \times l \) identity matrix. Once \( U \) is obtained, we can initialize \( \hat{Y} \) according to \( U \), i.e., \( U_{(\cdot, (q+1):l)} \). Then, we can construct a MLL model with a squared loss as

\[
\min_{W, U, V} \frac{1}{2} \| X - \hat{Y} \|_F^2 + \frac{\lambda_1}{2} \| X - UV \|_F^2 + \frac{\lambda_2}{2} \| UP - Y \|_F^2 \tag{2}
\]

\[
\text{s.t. } U \in [0, 1]^{n \times l}
\]

where \( W = [ w_1, \ldots, w_l ] \in \mathbb{R}^{d \times l} \) is the coefficient matrix, and \( l \)-norm regularization is used to learn sparse label-specific features [Zhang and Wu, 2015; Huang et al., 2016; Huang et al., 2018; Wei et al., 2019; Wu et al., 2019; Huang et al., 2019]. For the simplicity of optimization, the discrete constraint on the values of matrix \( U \) is relaxed to continuous, i.e., \( \forall u_{ij} \in [0, 1] \).

It is worth noting that the latent labels may have correlations with known labels more or less, and thus it was expected that the performance on known labels and latent labels will be both boosted by exploiting the correlations between them. Thus, let \( C \) be the label correlation matrix. Each element \( c_{ij} \) indicates the value of correlation between the \( i \)-th and \( j \)-th labels, and is estimated by calculating the cosine similarity between \( Y_{i,:} \) and \( \hat{Y}_{j,:} \). Since \( \hat{Y} \) is unknown, the calculated correlations between known and latent labels may not be reliable enough. Thus, we introduce an extra matrix \( R \in \mathbb{R}^{l \times l} \), and each element \( r_{ij} \) indicates the confidence of \( c_{ij} \) as

\[
r_{ij} = \begin{cases} 1 & \text{if } 1 \leq i, j \leq q \\ \alpha & \text{otherwise; } \alpha \in [0, 1] \end{cases}
\]

Then, we try to exploit the pairwise label correlation by modeling the Euclidean distance between any pair of model coefficient vectors. Specifically, if \( y_i \) and \( y_j \) have a strong correlation, their corresponding coefficients \( w_i \) and \( w_j \) will be similar, and thus the distance (i.e., \( d_{ij} = \| w_i - w_j \|_2 \)) will be small. Otherwise, the distance will be large. The fourth term of (2) was utilized to model pairwise label correlation.

3 Optimization

An alternating optimization strategy is adopted to solve problem (2), and \( \mathcal{L} \) represents the objective function of it.

3.1 Update W

With \( U \) and \( V \) fixed, problem (2) reduces to

\[
\min_W \frac{1}{2} \| XW - \hat{Y} \|_F^2 + \frac{\lambda_1}{2} \| W \|_1 \tag{4}
\]

where \( L \) is graph Laplacian of the weighted correlation matrix \( C \odot R \). Thus, the gradient w.r.t \( W \) can be calculated as

\[
\nabla_W \mathcal{L} = X^T X W - X^T \hat{Y} + \lambda_3 W \tag{5}
\]

The \( \ell_1 \)-norm regularization w.r.t \( W \) can be solved by the element-wise soft-threshold operator. According to the proximal gradient descend algorithm [Beck and Teboulle, 2009], \( W \) can be updated by

\[
W_{l+1} = \text{prox}_{\frac{\alpha_1}{L_f}} ( W_l ) - \frac{1}{L_f} \nabla_W \mathcal{L} ( \mathbf{P}, (W_l, U) ) \tag{6}
\]

where \( W_l = W_l + \frac{\alpha_{l-1} - 1}{\alpha_l} ( W_l - W_{l-1} ) \). \( L_f \) is the Lipschitz constant, and an upper bound of it is shown in Theorem 1. For a sequence \( \alpha_l \), it should satisfy the condition of \( \alpha_l^2 - \alpha_t \leq \alpha_{l-1}^2 \), and \( \text{prox}_r (a) \) is the element-wise operator which is defined as

\[
\text{prox}_r (a) = \text{sign}(a) \max(|a| - \epsilon, 0) \tag{7}
\]

Theorem 1 (Lipschitz Continuous Gradient). Given two arbitrary distinct parameters \( W_1 \) and \( W_2 \), we have

\[
\| \nabla_W \mathcal{L}(W_1) - \nabla_W \mathcal{L}(W_2) \|_F^2 \leq \gamma \| \Delta W \|_F^2 \tag{8}
\]

where \( \gamma = 2 \| X^T X \|_F^2 + 2 \| \lambda_3 L \|_2^2 \) and \( \Delta W = W_1 - W_2 \), and an approximate Lipschitz constant can be calculated by

\[
L_f = \sqrt{2 \| X^T X \|_F^2 + 2 \| \lambda_3 L \|_2^2} \tag{9}
\]

Proof. Given \( W_1 \) and \( W_2 \), according to Eq. (5), we have

\[
\| \nabla_W \mathcal{L}(W_1) - \nabla_W \mathcal{L}(W_2) \|_F^2
\]

\[
= \| X^T (X W_1 + \lambda_3 W_1 L - X^T X W_2 - \lambda_3 W_2 L) \|_F^2
\]

\[
= \| X^T (X W_1 - W_2 + \lambda_3 (W_1 - W_2) L) \|_F^2
\]

\[
= \| X^T (X \Delta W + \lambda_3 \Delta W L) \|_F^2
\]

\[
\leq 2 \| X^T X \|_F^2 + 2 \| \lambda_3 \Delta W L \|_F^2
\]

\[\leq 2 \| X^T X \|_F^2 + 2 \| \lambda_3 L \|_2^2 \| \Delta W \|_F^2
\]

\[= (2 \| X^T X \|_F^2 + 2 \| \lambda_3 L \|_2^2) \| \Delta W \|_F^2
\]
Algorithm 1 Training of Dlcl.

Input: Training data: $X \in \mathbb{R}^{n \times d}$, label matrix $Y \in \mathbb{R}^{n \times q}$; Parameter: The non-negative weighting parameters $\lambda_1$, $\lambda_2$, $\lambda_3$, and $\lambda_4$, the number of latent class labels $k$, and $\alpha$;
Output: $W$, $U$, and $V$
1: $\alpha_1 = 1$.
2: Initialize $W$, $U$, and $V$ with random value.
3: $Y = U(:,(q+1):i)$.
4: while stop criterion not reached do
5: Calculate $C$ based on $Y$.
6: Let $C = C \odot R$, and then calculate $L$.
7: calculate $L_f$ according to Eq. (8).
8: update $W$ according to Eq. (6).
9: update $U$ according to Eq. (10).
10: update $V$ according to Eq. (13).
11: search $\beta^*$ according to (14).
12: update $\tilde{Y}$ according to Eq. (15).
13: $\alpha_{t+1} = \frac{1}{2} + \frac{1}{2} \sqrt{1 + \frac{4\alpha_t^2 + \alpha_t}{2}}$.
14: end while
15: return $W$, $U$, and $V$

3.2 Update $U$

With $W$ and $V$ fixed, problem (2) becomes
\[
\min_U \frac{\lambda_1}{2} \|X - UV\|_F^2 + \frac{\lambda_2}{2} \|UP - Y\|_F^2 (9)
\]
Thus, the gradient w.r.t $U$ can be calculated as
\[
\nabla_U \mathcal{L} = \lambda_1 (UVV^T - XV^T) + \lambda_2 (UPP^T - YP^T)
\]
Therefore, we can obtain a closed-form solution for $U$ as
\[
U = (\lambda_1 XV^T + \lambda_2 YP^T)(\lambda_1 VV^T + \lambda_2 PP^T)^{-1} (10)
\]
Then, $\forall u_{ij} \in [0, 1]$ can be achieved by $U = \max(U, 0)$ and the min-max normalization over each column of $U$.

3.3 Update $V$

With $U$ and $W$ fixed, problem (2) is simplified as
\[
\min_V \frac{\lambda_1}{2} \|X - UV\|_F^2 (11)
\]
Consequently, the gradient w.r.t $V$ can be calculated by
\[
\nabla_V \mathcal{L} = \lambda_1 U^TUV - \lambda_1 U^TX
\]
Then, a closed-form solution for $V$ can be obtained as
\[
V = (U^TU)^{-1}U^TX (13)
\]

3.4 Update $\tilde{Y}$

In (2), $\tilde{Y} = [\tilde{Y}, \tilde{Y}]$ is the full label matrix, and $\tilde{Y}$ indicates the latent label matrix and is unknown in advance. Therefore, we need to update $\tilde{Y}$ after each iteration of the optimization. As aforementioned, $U$ is an approximate representation of $\tilde{Y}$. Besides, we can also obtain a result from the classifier, i.e., $XW$, and it is expected that the useful information (i.e., label correlation) induced from the known label can lead to a good prediction on the latent labels. Therefore, we plan to update $\tilde{Y}$ according to both to $U$ and $XW$ with a balance weight $\beta \in (0, 1)$. Since $Y$ is known, we can search for a proper value for $\beta$ from $\{0.05, 0.1, \ldots, 0.95\}$ according to (14).
\[
\beta^* = \arg \min_{\beta} \|Y - (\beta UP + (1 - \beta) XW)\|_1 (14)
\]
Then, $\tilde{Y}$ can be updated by
\[
\tilde{Y} = \beta^* U(:,(q+1):i) + (1 - \beta^*) XW(:,(q+1):i) (15)
\]

4 Experiment

4.1 Experimental Setting

By surveying previous work on MLL, we noted that there is no previous work doing the same topic like us. MMLNC [Pham et al., 2015] and Dnst. [Zhu et al., 2017a] are the only two highly related studies on discovering novel labels, but these two approaches are tailored for MIMLL, and cannot be applied to general single-instance MLL directly. In order to verify the effectiveness of our proposed method, we compare Dlcl with four state-of-the-art approaches in terms of their performance on known labels. Detailed configurations of them are summarized as: 1) Bk [Boutell et al., 2004]: Binary Relevance. It learns a binary classifier (one-vs-rest) for each label independently, and Linear Regression is utilized as the base binary learner for it, and the regularization parameter is tuned in $\{10^i | i = -2, \ldots, 2\}$. 2) MLNN [Zhang and Zhou, 2007]: A lazy learning approach to MLL, the number of n-nearest neighbors $k$ is searched in $\{7, \ldots, 17\}$. 3) LLSF [Huang et al., 2016]: Learning label-specific features for MLL, the regularization parameters are tuned in $\{2^i | i = -10, \ldots, 10\}$. 4) KRAM [Jia and Zhang, 2019]: Multi-dimensional classification via knn feature augmentation. The number of nearest neighbors $k$ is searched in $\{7, \ldots, 17\}$. 5) DLCL: The proposed approach in this paper. Parameters $\lambda_1$ and $\lambda_2$ are tuned in $\{10^i | i = 2, \ldots, 6\}$, $\lambda_3$ is tuned in $\{2^i | i = -2, \ldots, 4\}$, $\lambda_4$ is tuned in $\{10^i | i = -2, \ldots, 1\}$, and $\alpha$ is tuned in $\{0.4, 0.5, 0.6\}$. Parameter tuning for each of them is based on a 5-fold cross validation over the training data of each data set.

Table 1 shows a summarization of the twelve experimental data sets. We adopt six common evaluation metrics [Zhang

\footnote{code: http://palm.seu.edu.cn/zhangml/files/KRAM.rar}

\footnote{code: http://palm.seu.edu.cn/zhangml/files/ML-kNN.rar}

\footnote{code: http://palm.seu.edu.cn/zhangml/files/ML-kNN.rar}
## Table 2: Experimental results (mean) of all the comparing approaches on known labels

| Data sets          | BR | HLSN | LLSF | KRAM | DCLL |
|--------------------|----|------|------|------|------|
| arts               | 0.010 | 0.090 | 0.075 | 0.077 | 0.071 |
| bitext             | 0.017 | 0.018 | 0.020 | 0.017 | 0.014 |
| corel1k1           | 0.030 | 0.028 | 0.030 | 0.029 | 0.028 |
| corel1k2           | 0.029 | 0.028 | 0.028 | 0.027 | 0.027 |
| education          | 0.063 | 0.063 | 0.059 | 0.054 | 0.058 |
| medical            | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 |
| chemistry          | 0.019 | 0.021 | 0.018 | 0.018 | 0.021 |
| cooking            | 0.011 | 0.011 | 0.010 | 0.011 | 0.011 |
| philosophy         | 0.014 | 0.013 | 0.012 | 0.013 | 0.012 |

| Data sets          | Average Precision ($q = 70\%$, $k = 70\%$) | Average Precision ($q = 80\%$, $k = 20\%$) | Average Precision ($q = 90\%$, $k = 10\%$) |
|--------------------|---------------------------------|---------------------------------|---------------------------------|
| arts               | 0.811                           | 0.764                           | 0.709                           |
| bitext             | 0.693                           | 0.580                           | 0.503                           |
| corel1k1           | 0.742                           | 0.674                           | 0.611                           |
| corel1k2           | 0.734                           | 0.667                           | 0.614                           |
| education          | 0.656                           | 0.566                           | 0.504                           |
| medical            | 0.567                           | 0.471                           | 0.412                           |
| chemistry          | 0.545                           | 0.454                           | 0.397                           |
| cooking            | 0.539                           | 0.449                           | 0.392                           |
| philosophy         | 0.544                           | 0.454                           | 0.398                           |

| Data sets          | One Error ($q = 70\%$, $k = 70\%$) | One Error ($q = 80\%$, $k = 20\%$) | One Error ($q = 90\%$, $k = 10\%$) |
|--------------------|---------------------------------|---------------------------------|---------------------------------|
| arts               | 0.329                           | 0.286                           | 0.272                           |
| bitext             | 0.348                           | 0.307                           | 0.296                           |
| corel1k1           | 0.356                           | 0.319                           | 0.307                           |
| corel1k2           | 0.345                           | 0.308                           | 0.300                           |
| education          | 0.343                           | 0.305                           | 0.295                           |
| medical            | 0.344                           | 0.306                           | 0.297                           |
| chemistry          | 0.338                           | 0.302                           | 0.293                           |
| cooking            | 0.347                           | 0.310                           | 0.299                           |
| philosophy         | 0.352                           | 0.315                           | 0.303                           |

| Data sets          | Coverage ($q = 70\%$, $k = 70\%$) | Coverage ($q = 80\%$, $k = 20\%$) | Coverage ($q = 90\%$, $k = 10\%$) |
|--------------------|---------------------------------|---------------------------------|---------------------------------|
| arts               | 0.108                           | 0.103                           | 0.103                           |
| bitext             | 0.118                           | 0.113                           | 0.113                           |
| corel1k1           | 0.115                           | 0.110                           | 0.110                           |
| corel1k2           | 0.105                           | 0.100                           | 0.100                           |
| education          | 0.104                           | 0.093                           | 0.093                           |
| medical            | 0.103                           | 0.098                           | 0.098                           |
| chemistry          | 0.103                           | 0.098                           | 0.098                           |
| cooking            | 0.106                           | 0.101                           | 0.101                           |
| philosophy         | 0.107                           | 0.102                           | 0.102                           |

| Data sets          | MacRO C ($q = 70\%$, $k = 70\%$) | MacRO C ($q = 80\%$, $k = 20\%$) | MacRO C ($q = 90\%$, $k = 10\%$) |
|--------------------|---------------------------------|---------------------------------|---------------------------------|
| arts               | 0.757                           | 0.735                           | 0.725                           |
| bitext             | 0.819                           | 0.784                           | 0.770                           |
| corel1k1           | 0.814                           | 0.782                           | 0.769                           |
| corel1k2           | 0.815                           | 0.782                           | 0.769                           |
| education          | 0.817                           | 0.783                           | 0.770                           |
| medical            | 0.817                           | 0.783                           | 0.770                           |
| chemistry          | 0.817                           | 0.783                           | 0.770                           |
| cooking            | 0.818                           | 0.784                           | 0.770                           |
| philosophy         | 0.819                           | 0.785                           | 0.770                           |

| Data sets          | MacRO A ($q = 70\%$, $k = 70\%$) | MacRO A ($q = 80\%$, $k = 20\%$) | MacRO A ($q = 90\%$, $k = 10\%$) |
|--------------------|---------------------------------|---------------------------------|---------------------------------|
| arts               | 0.757                           | 0.735                           | 0.725                           |
| bitext             | 0.819                           | 0.784                           | 0.770                           |
| corel1k1           | 0.814                           | 0.782                           | 0.769                           |
| corel1k2           | 0.815                           | 0.782                           | 0.769                           |
| education          | 0.817                           | 0.783                           | 0.770                           |
| medical            | 0.817                           | 0.783                           | 0.770                           |
| chemistry          | 0.817                           | 0.783                           | 0.770                           |
| cooking            | 0.818                           | 0.784                           | 0.770                           |
| philosophy         | 0.819                           | 0.785                           | 0.770                           |
and Zhou, 2014], i.e., Hamming Loss, Average Precision, One Error, Ranking Loss, Coverage, and Macro AUC, to evaluate the performance of the comparing algorithms on known labels. To evaluate the performance of our method on discovered latent labels, we adopt the following metric proposed in [Zhu et al., 2017a],

$$F_{\text{novel}} = \frac{1}{k} \sum_{i=1}^{k} \max\{F(Y_{i,q}, G_{i,q+j}), j \in \{1, ..., k\}\}$$

where $F(\cdot)$ is the function of F-measure, and $G$ indicates the ground-truth label matrix. $F_{\text{novel}}$ measures the average performance on detected multiple latent labels on the ground-truth label that best matches.

### 4.2 Experimental Results

For each data set, 80% of it are randomly generated as the training part and 20% for testing, which is repeated 10 times. Following the settings in previous work on discovering new labels for MIMLL [Pham et al., 2015; Zhu et al., 2017a], the first 70%, 80% and 90% labels are set be to known labels and the rest are taken as latent ones respectively. The average results of each comparing algorithm on the known labels are shown in Figure 1. $\uparrow$ (\downarrow) indicates the larger (smaller) the value, the better the performance. Best results are highlighted in bold face.

#### Results on Known Labels.

Friedman test [Demšar, 2006] is employed to conduct performance analysis among the comparing approaches, and the result of it is shown in Table 3. As shown in Table 3, the null hypothesis that all the comparing algorithms perform equivalently is clearly rejected in terms of all the evaluation metrics at significance level $\alpha = 0.05$. Consequently, the Nemenyi test [Demšar, 2006] is adopted to analyse the relative performance among them. For Nemenyi test, the critical difference $\text{CD} = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}} = 1.0167$ ($k=5$, $N=36$) with $q_{\alpha}=2.725$ at significance level $\alpha=0.05$, where $k$ is the number of algorithms and $N (16 \times 3)$ is the number of data sets. The CD diagrams of DLCL w.r.t to the comparing algorithms on each evaluation metric are shown in Figure 1. In each sub-figure, any comparing algorithm whose average rank is within one CD to that of DLCL is connected. Otherwise, any algorithm not connected with DLCL is considered to have significant different performance between them. According to these experimental results, the following observations can be made: 1) The proposed method DLCL significantly outperforms the comparing algorithms in terms of ranking loss, average precision, coverage, and AUC. Besides, DLCL statistically outperforms the comparing algorithms in terms of hamming loss and one error. These results definitely demonstrate the effectiveness of our method on MLL. 2) KRAM and MLANN algorithms are all constructed based on the information of $k$ nearest neighbors of each instance. It is worth noting that these two algorithms achieve worse performance on those data with a large number of labels (e.g., $l>=100$) than on those data with a small number of labels. One possible reason might be that the $k$ nearest neighbors can not provide sufficient information for model construction when many labels are hidden in the data. These results verify the importance of discovering latent class labels for MLL.

#### Results on Latent Labels.

Table 4 shows the results of DLCL on discovered latent class labels according to $F_{\text{novel}}$. It is clearly indicated that DLCL can discover the latent labels for MLL, and DLCL significantly outperforms RS(Randomly Setting) when $q = 70\%$ and $k = 30\%$. For some data sets, the results of $F_{\text{novel}}$ differ extremely under different ratios of latent labels. The possible reason might be that the difficulties of prediction of different labels are different. On the other hand, we want to know what latent labels have we discov-
The number of Latent labels \( k \) and correlation confidence \( \alpha \). For \textit{arts}, the first 20 and the rest 6 labels are set as known and latent labels respectively. Figure 2 shows the average results of DLCL over 10 repetitions with different values of \( k \) and \( \alpha \). The result (i.e., \textit{Macro AUC}) on known labels is improved by discovering latent labels (i.e., \( k > 0 \)), and the \( P_{\text{novel}} \) decreases with the increasing of \( k \). The larger the number of latent labels, the harder it is to discover them. Therefore, we could set \( k \) to be a relative small value and run DLCL multiple times. It is also noted that the performance of latent labels can be significantly improved with the help of known labels and the improvement on known labels is slightly with the help of latent labels by exploiting their correlations (i.e., \( \alpha > 0 \)). The known labels are observed in advance and can be considered as a teacher to guide the prediction on latent labels.

Analysis on regularization parameters. The average results (i.e, \textit{Average Precision}) of DLCL with different values of \( \lambda_1, \lambda_2, \lambda_3, \) and \( \lambda_4 \) over \textit{stackex-chemistry} are shown in Figure 3, and similar results were also obtained for the other data sets. It is noted that the performance of DLCL is insensitive to the parameters, and also the optimal performance is usually achieved at some intermediate values of each parameter.

Convergence. Figure 4 demonstrates two examples of the convergence curves of DLCL. It is noted that the values of the objective function are non-increasing and drop sharply around 15 iterations on \textit{corel5k} and \textit{stackex-cs} data sets.

5 Conclusion

In this paper, a novel approach named DLCL is proposed for MLL which can not only discover the latent labels in the training data but also predict new instances with these latent labels and known labels simultaneously. The experimental results demonstrate that the performance of latent labels can be significantly improved with the help of known labels and the performance of known labels can be improved with the help of latent labels by exploiting their correlations. Extensive experiments with other state-of-the-art MLL approaches have show a competitive performance of DLCL.

Acknowledgement

This work is supported by JST KAKENHI: 19H00000190, JST-AIP: JPMJCR19U4, and NSFC: 61806005.
References

[Beck and Teboulle, 2009] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[Boutell et al., 2004] M. R. Boutell, J.-B. Luo, X.-P. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognit.*, 37(9):1757–1771, 2004.

[Chen et al., 2019] C. Chen, H. Wang, W. Liu, X. Zhao, T. Hu, and G. Chen. Two-stage label embedding via neural factorization machine for multi-label classification. In *AAAI*, pages 3304–3311, 2019.

[Demsar, 2006] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, 2006.

[Gibaja and Ventura, 2015] E. Gibaja and S. Ventura. A tutorial on multilabel learning. *ACM Comput. Surv.*, 47(3):52:1–52:38, 2015.

[Huang et al., 2016] J. Huang, G. Li, Q. Huang, and X. Wu. Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Trans. Knowl. Data Eng.*, 28(12):3309–3323, 2016.

[Huang et al., 2018] J. Huang, G. Li, Q. Huang, and X. Wu. Joint feature selection and classification for multilabel learning. *IEEE Trans. Cybern.*, 48(3):876–889, 2018.

[Huang et al., 2019] J. Huang, F. Qin, X. Zheng, Z. Cheng, Z. Yuan, W. Zhang, and Q. Huang. Improving multi-label classification with missing labels by learning label-specific features. *Inf. Sci.*, 492:124–146, 2019.

[Jain et al., 2019] H. Jain, V. Balasubramanian, B. Chunduri, and M. Varma. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *WSDM*, pages 528–536, 2019.

[Jia and Zhang, 2019] B.-B. Jia and M.-L. Zhang. Multidimensional classification via knn feature augmentation. In *AAAI*, pages 3975–3982, 2019.

[Kuzborskij et al., 2013] I. Kuzborskij, F. Orabona, and B. Caputo. From n to n+1: Multiclass transfer incremental learning. In *CVPR*, pages 3358–3365, 2013.

[Nguyen et al., 2016] V. Nguyen, T. D. Nguyen, T. Le, S. Venkatesh, and D. Phung. One-pass logistic regression for label-drift and large-scale classification on distributed systems. In *ICDM*, pages 1113–1118, 2016.

[Pham et al., 2015] A. Pham, R. Raich, X. Fern, and J. P. Arriaga. Multi-instance multi-label learning in the presence of novel class instances. In *ICML*, pages 2427–2435, 2015.

[Tan et al., 2018] Q. Tan, G. Yu, C. Domeniconi, J. Wang, and Z. Zhang. Incomplete multi-view weak-label learning. In *IJCAI*, 2018.

[Wei et al., 2019] T. Wei, W.-W. Tu, and Y.-F. Li. Learning for tail label data: A label-specific feature approach. In *IJCAI*, pages 3842–3848, 2019.

[Wu et al., 2019] X. Wu, Q. Chen, Y. Hu, D. Wang, X. Chang, X. Wang, and M.-L. Zhang. Multi-view multi-label learning with view-specific information extraction. In *IJCAI*, pages 3884–3890, 2019.

[Xing et al., 2018] Y. Xing, G. Yu, C. Domeniconi, J. Wang, and Z. Zhang. Multi-label co-training. In *IJCAI*, 2018.

[Xioufis et al., 2011] E. S. Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas. Dealing with concept drift and class imbalance in multi-label stream classification. In *IJCAI*, pages 1583–1588, 2011.

[Xu et al., 2019] N. Xu, J. Lv, and X. Geng. Partial label learning via label enhancement. In *AAAI*, pages 3838–3845, 2019.

[Yang et al., 2016] H. Yang, J. T. Zhou, and J. Cai. Improving multi-label learning with missing labels by structured semantic correlations. In *ECCV*, pages 835–851, 2016.

[Yu et al., 2014] H. Yu, P. Jain, P. Kar, and I. S. Dhillon. Large-scale multi-label learning with missing labels. In *ICML*, pages 593–601, 2014.

[Zhang and Wu, 2015] M.-L. Zhang and L. Wu. Lift: Multi-label learning with label-specific features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(1):107–120, 2015.

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

[Zhang and Zhou, 2014] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.*, 26(8):1819–1837, 2014.

[Zhang et al., 2018] Y. Zhang, R. Henao, Z. Gan, Y. Li, and L. Carin. Multi-label learning from medical plain text with convolutional residual models. In *MLHC*, pages 280–294, 2018.

[Zhang et al., 2020] Y. Zhang, Y. Wang, X.-Y. Liu, S. Mi, and M.-L. Zhang. Large-scale multi-label classification using unknown streaming images. *Pattern Recognit.*, page 107100, 2020.

[Zhu et al., 2017a] Y. Zhu, K.-M. Ting, and Z.-H. Zhou. Discover multiple novel labels in multi-instance multi-label learning. In *AAAI*, pages 2977–2983, 2017.

[Zhu et al., 2017b] Y. Zhu, K.-M. Ting, and Z.-H. Zhou. New class adaption via instance generation in one-pass class incremental learning. In *ICDM*, pages 1207–1212, 2017.

[Zhu et al., 2018] Y. Zhu, K.-M. Ting, and Z.-H. Zhou. Multi-label learning with emerging new labels. *IEEE Trans. Knowl. Data Eng.*, 30(10):1901–1914, 2018.