Labeling Information Enhancement for Multi-label Learning with Low-Rank Subspace

An Tao[1](✉), Ning Xu[2,3], and Xin Geng[2,3]

[1] School of Information Science and Engineering,
Southeast University, Nanjing, China
taoan@seu.edu.cn

[2] School of Computer Science and Engineering,
Southeast University, Nanjing, China
{xning,xgeng}@seu.edu.cn

[3] Key Laboratory of Computer Network and Information Integration,
Southeast University, Ministry of Education, Nanjing, China

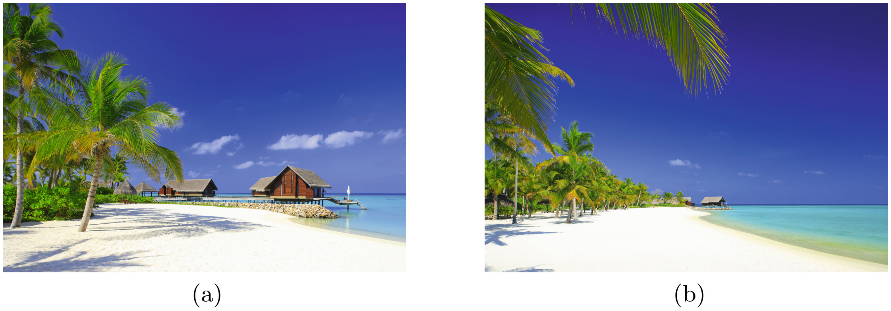
Abstract. In multi-label learning, each training example is represented by an instance while associated with multiple class labels simultaneously. Most existing approaches make use of multi-label training examples by utilizing the logical labeling information, i.e., one class label is either fully relevant or irrelevant to the instance. In this paper, a novel multi-label learning approach is proposed which aims to enhance the labeling information by extending logical labels into numerical labels. Firstly, a stacked matrix is constructed where the feature and the logical label matrix are placed vertically. Secondly, the labeling information is enhanced by leveraging the underlying low-rank structure in the stacked matrix. Thirdly, the multi-label predictive model is induced by the learning procedure from training examples with numerical labels. Extensive comparative studies clearly validate the advantage of the proposed method against the state-of-the-art multi-label learning approaches.

Keywords: Multi-label learning · Label enhancement · Low-rank

1 Introduction

In multi-label learning, there are multiple labels associated to the same instance simultaneously [1,2]. In more formal terms, let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimentional feature space and $\mathcal{Y} = [y_1, \ldots, y_t]$ denote the label set with t possible labels. Given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) | 1 \leq i \leq n\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is the feature vector and $\boldsymbol{y}_i \in \{0, 1\}^t$ is the label vector, the task of traditional multi-label learning is to learn a model built from the feature space to the label space. During the past decade, multi-label learning has been applied successfully to learn data with rich semantics, e.g. text [3,4], image [5,6], audio [7,8], video [9], etc.

X. Geng and B.-H. Kang (Eds.): PRICAI 2018, LNAI 11012, pp. 671–683, 2018.
https://doi.org/10.1007/978-3-319-97304-3_51



(a) (b)

Fig. 1. Two natural scene image examples which are both described by the label set $\mathcal{Y} = \{sky, water, cloud, beach, plant, house\}$ in different value.

The accessible labeling information of multi-label training example is categorical, i.e. each class label $\boldsymbol{y}_i$ is regarded to be either relevant or irrelevant for instance $\boldsymbol{x}_i$. Such label $\boldsymbol{y}_i$ is called *logical label.* Nonetheless, recent studies show that categorical labeling information is actually a simplification of the rich semantics encoded by multi-label training examples [10]. In order to enhance the labeling information, the logical label should be extended to be numerical. This new label is called *numerical label*, which carries more semantic information and describes the instance more comprehensively. An example is shown in Fig. 1. Figure 1(a) and (b) are both annotated with the label set $\mathcal{Y}$=$\{$*sky*, *water*, *cloud*, *beach*, *plant*, *house*$\}$. In order to specify the numerical label, we let the sign of the numerical label denotes whether the label is relevant or irrelevant to the corresponding instance, and let the absolute numerical value denote the degree to which the label describes the instance. To reflect the labeling information with numerical labels, for the *within-instance label variance*, the related label *beach* in Fig. 1(b) should have larger value than the value of the related label *water*, because the former can describe the image more apparently than the latter. Similarly, for the *between-instance label variance*, the value of the related label *house* in Fig. 1(a) should be larger than the value of the same label in Fig. 1(b). The process of recovering numerical labels from logical labels can be called *label enhancement* (LE). LE can be seen as a data preprocessing step which aims to facilitate in learning a better model from the feature space to the label space.

To enhance the labeling information of multi-label examples, we first construct a stacked matrix where the feature and the logical label matrix are placed vertically. Then we assume that the stacked matrix belongs to an underlying low-rank subspace [11,12]. The stacked matrix is therefore an underlying low-rank matrix.

Based on the above assumption, we propose an efficient multi-label method named LIEML, i.e., *Labeling Information Enhancement for Multi-label Learning.* The basic strategy of LIEML is to enhance the labeling information of multi-label examples by leveraging the underlying low-rank structure in the stacked matrix.

Specifically, we minimize the rank of the stacked matrix until obtaining the lowest rank. In order to prevent the values in the stacked matrix from deviating the original values too much, we add two functions on the feature and the label portion of the stacked matrix respectively. The numerical labels are then obtained from the label portion of the stacked matrix. After that, the desired multi-label predictive model is learned from training examples with numerical labels based on tailored multivariate regression techniques. Experimental studies across a wide range of benchmark data sets show that LIEML achieves highly competitive performance against other state-of-the-art multi-label learning approaches.

The rest of this paper is organized as follows. First, existing work related to our proposed approach is discussed in Sect. 2. Then, the details of LIEML are proposed in Sect. 3. After that, the results of comparative studies are reported in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Related Work

Existing multi-label approaches can be roughly grouped into three categories based on the thought of *order of label correlations* [2]. The first-order approaches which assume independence among class labels are the simplest ones [13,14]. Then the multi-label classification becomes a series of binary classification problems. On the contrary, second-order approaches consider the correlations between pairs of class labels [15,16], and the high-order approaches consider the correlations among label subsets or all the class labels [17]. The approaches above all treat the label as the logcial label, representing whether the label is fully relevant or irrelevant to the corresponding instance. In contrast, LIEML enhances the labeling information by transforming logical label to numerical label.

There have been some multi-label works which transform the logical label space to the numerical label space. For example, [18] tries to reduce the computational effort by seeking the principle correlations between labels, especially for the data sets with large numbers of labels. The bases of the numerical space are the combinations of the logical label vectors. Another work [19] projects the feature space and the label space to a new space where the correlation between the projections of the two spaces are maximized. In both cases, the dimensionality of the label space is reduced. However, the meaning of each dimension still remains in LIEML.

Recently, there are also some attempts which aim to facilitate multi-label predictive model induction by manipulating the feature space. In [20], label propagation is conducted over the fully-connected affinity graph specified over the feature space. In [10], the manifold structure of feature space is characterized by the weighted k-nearest neighbor graph defined over training examples. Different to those approaches, LIEML recovers numerical label by exploiting the structure of the stacked matrix via low-rank assumption.

3 The LIEML Algorithm

The learning procedure of LIEML consists of two steps, including label enhancement and predictive model induction. Technical details of these steps are scrutinized as follows.

3.1 Label Enhancement

As shown in Sect. 1, the training set of multi-label learning can be expressed as $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) | 1 \leq i \leq n\}$. Given any instance $\boldsymbol{x}_i \in \mathbb{R}^d$ and the logical label vector $\boldsymbol{y}_i \in \{-1, 1\}^t$, we use $\boldsymbol{u}_i \in \mathbb{R}^t$ to denote the numerical label vector. Note that here we use -1 instead of 0 in the logical label vector to represent irrelevant to the instance.

The goal of LE is to recover the numerical labels from the logical labels in $\mathcal{D}$. To solve the problem, we consider the linear model

$$\boldsymbol{u}_i = \boldsymbol{W}^\top \boldsymbol{x}_i + \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{W} = [\boldsymbol{w}^1, \ldots, \boldsymbol{w}^t]$ is a weight matrix and $\boldsymbol{b} \in \mathbb{R}^t$ is a bias vector. For convenient describing, we set $\hat{\boldsymbol{W}} = [\boldsymbol{W}^\top, \boldsymbol{b}]$. Then the Eq. 1 becomes

$$\boldsymbol{U} = \hat{\boldsymbol{W}}[\boldsymbol{X}; \boldsymbol{1}^\top]. \tag{2}$$

As shown in Sect. 1, we construct a stacked matrix $\boldsymbol{Z} = [\boldsymbol{Y}; \boldsymbol{X}; \boldsymbol{1}^\top]$ and assume that $\boldsymbol{Z}$ is an underlying low-rank matrix, i.e., $\text{rank}(Z) \ll \min(d+t+1, n)$. This assumption succeeds in many component analysis techniques, e.g., principal component analysis (PCA) and Fisher's linear discriminant analysis (FLDA). Because of our model in Eq. 2, an all-1 row is added in $\boldsymbol{Z}$. After minimizing the rank of $\boldsymbol{Z}$, we gain the numerical label matrix $\boldsymbol{U}$ from the position of $\boldsymbol{Y}$ in $\boldsymbol{Z}$. The optimization problem of LE in this paper becomes

$$\begin{aligned} \operatorname*{argmin}_{\boldsymbol{Z} \in \mathbb{R}^{(t+d+1) \times n}} \quad & L(\boldsymbol{Z}) + R(\boldsymbol{Z}) \\ \text{s.t.} \quad & z_{(t+d+1)\cdot} = \boldsymbol{1}^\top. \end{aligned} \tag{3}$$

In order to prevent the label values in $\boldsymbol{Z}$ from deviating the original values too much, for L we choose the logistic loss function as

$$L(\boldsymbol{Z}) = \frac{\lambda}{tn} \sum_{i=1}^{t} \sum_{j=1}^{n} \log(1 + e^{-y_{ij} z_{ij}}), \tag{4}$$

where λ is a positive trade-off weight.

Because rank($\boldsymbol{Z}$) is non-convex and difficult to optimize, we relax rank($\boldsymbol{Z}$) with the convex nuclear norm $\|\boldsymbol{Z}\|_*$. Considering the feature values should not deviate from the orignal values too much, we choose the squared function on the position of $\boldsymbol{X}$ in $\boldsymbol{Z}$. This leads to the following R which we wish to minimize

$$R(\boldsymbol{Z}) = \mu\|\boldsymbol{Z}\|_* + \frac{1}{dn}\sum_{i=1}^{d}\sum_{j=1}^{n}\frac{1}{2}(z_{(i+t)j} - x_{ij})^2, \tag{5}$$

where μ is a positive trade-off weight.

Formulating the LE problem into an optimization framework over Eqs. 4 and 5, the target function T_1 is yielded as

$$T_1(\boldsymbol{Z}) = \frac{\lambda}{tn}\sum_{i=1}^{t}\sum_{j=1}^{n}\log(1+e^{-y_{ij}z_{ij}}) + \mu\|\boldsymbol{Z}\|_* + \frac{1}{dn}\sum_{i=1}^{d}\sum_{j=1}^{n}\frac{1}{2}(z_{(i+t)j} - x_{ij})^2. \tag{6}$$

To optimize the Eq. 6, we modify the Fixed Point Continuation (FPC) method [21], which is to alternate between the gradient descent $\boldsymbol{A}^k = \boldsymbol{Z}^k - \tau g(\boldsymbol{Z}^k)$ and the shrinkage $\boldsymbol{Z}^{k+1} = S_{\tau\mu}(\boldsymbol{A}^k)$. For the gradient descent, τ is a step size whose choice will be discussed next, and $g(\boldsymbol{Z}^k)$ is the matrix gradient of the logistic loss function and the squared function in Eq. 3. We define $g(\boldsymbol{Z}^k)$ as

$$g(\boldsymbol{z}_{ij}) = \begin{cases} \frac{\lambda}{tn}\frac{-y_{ij}}{1+e^{y_{ij}z_{ij}}}, & i \le t \\ \frac{1}{dn}(z_{ij} - x_{(i-t)j}), & t < i \le t+d \\ 0, & i = t+d+1 \end{cases}. \tag{7}$$

For the shrinkage, $S_{\tau\mu}(\cdot)$ is a matrix shrinkage operator. Let $\boldsymbol{A}^k = \boldsymbol{U\Lambda V}^\top$ be the SVD of $\boldsymbol{A}^k$. Then $S_{\tau\mu}(\boldsymbol{A}^k) = \boldsymbol{U}\max(\boldsymbol{\Lambda} - \tau\mu, 0)\boldsymbol{V}^\top$, which reduces the nuclear norm.

To improve the speed of convergence, we begin with a large value μ_1 for μ, and determine the sequence value as $\mu_{k+1} = \max(\mu_k\eta_\mu, \mu_{min})$, $k = 1, ..., L-1$, through a decay parameter η_μ. The sequence is ended with the smallest value μ_L which is equal to μ_{min}.

3.2 Predictive Model Induction

We build the learning model through an adapted regressor based on MSVR [10]. Similar to MSVR, we generalize the 1-D SVR to solve the multi-dimensional case. In addition, our regressor not only concerns the distance between the predicted and the real values, but also the sign consistency of them. The target funciton T_2 we wish to minimize is

$$T_2(\boldsymbol{\Theta}, \boldsymbol{m}) = \frac{1}{2}\sum_{j=1}^{t}\|\boldsymbol{\theta}_j\|^2 + \gamma_1\sum_{i=1}^{n}\varOmega_1(r_i) + \gamma_2\sum_{i=1}^{n}\sum_{j=1}^{t}\varOmega_2(q_{ij}), \tag{8}$$

where $r_i = \|\boldsymbol{e}_i\| = \sqrt{\boldsymbol{e}_i^\top\boldsymbol{e}_i}$, $\boldsymbol{e}_i = \boldsymbol{u}_i - \varphi(\boldsymbol{x}_i)^\top\boldsymbol{\Theta} - \boldsymbol{m}$, $q_{ij} = y_{ij}(\varphi(\boldsymbol{x}_i)^\top\boldsymbol{\theta}_j + m_j)$, $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t]$, $\boldsymbol{m} = [m_1, \ldots, m_t]$. $\varphi(\boldsymbol{x})$ is a nonlinear transformation of $\boldsymbol{x}$ to a higher dimensional feature space.

To consider all dimensions into a unique restriction and yield a single support vector for all dimensions, we set Ω_1 as

$$\Omega_1(r) = \begin{cases} 0, & r < \varepsilon \\ r^2 - 2r\varepsilon + \varepsilon^2, & r \geq \varepsilon \end{cases}. \tag{9}$$

This will create an insensitive zone determined by ε around the estimate, i.e., the value of r less than ε will be ignored.

To make the signs of the numerical label and the logical label as same as possible, we set Ω_2 as

$$\Omega_2(q) = -q\sigma(-q) = \begin{cases} 0, & q > 0 \\ -q, & q \leq 0 \end{cases}, \tag{10}$$

where $\sigma(q)$ is an activation function where the value will be equal to 0 if q is negative, otherwise equal to 1. For the meaning of Eq. 10, if the signs of the predicted numerical label and the logical label are different, the result of Eq. 10 will be positive, otherwise zero.

To minimize $T_2(\mathbf{\Theta}; \boldsymbol{m})$, we use an iterative quasi-Newton method called Iterative Re-Weighted Least Square (IRWLS) [22]. Firstly, we approximate $T_2(\mathbf{\Theta}; \boldsymbol{m})$ by its first order Taylor expansion at the solution of the current k-th iteration, denoted by $\mathbf{\Theta}^{(k)}$ and $\boldsymbol{m}^{(k)}$

$$\Omega_1'(r_i) = \Omega_1(r_i^{(k)}) + \left.\frac{d\Omega_1(r)}{dr}\right|_{r_i^{(k)}} \frac{(\boldsymbol{e}_i^{(k)})^\top}{r_i^{(k)}} \left(\boldsymbol{e}_i - \boldsymbol{e}_i^{(k)}\right), \tag{11}$$

where $\boldsymbol{e}_i^{(k)}$ and $r_i^{(k)}$ are calculated from $\mathbf{\Theta}^{(k)}$ and $\boldsymbol{m}^{(k)}$. Then a quadratic approximation is further constructed

$$\begin{aligned} \Omega_1''(r_i) &= \Omega_1(r_i^{(k)}) + \left.\frac{d\Omega_1(r)}{dr}\right|_{r_i^{(k)}} \frac{r_i^2 - (r_i^{(k)})^2}{2r_i^{(k)}} \\ &= \frac{1}{2}a_i r_i^2 + \nu, \end{aligned} \tag{12}$$

where

$$a_i = \frac{1}{r_i^{(k)}} \left.\frac{d\Omega_1(r)}{dr}\right|_{r_i^{(k)}} = \begin{cases} 0, & r_i^{(k)} < \varepsilon \\ \frac{2\left(r_i^{(k)} - \varepsilon\right)}{r_i^{(k)}}, & r_i^{(k)} \geq \varepsilon \end{cases}, \tag{13}$$

and ν is a constant term that does not depend on either $\mathbf{\Theta}^{(k)}$ or $\boldsymbol{m}^{(k)}$. Combining Eqs. 8, 10 and 12 can get

$$T_2''(\mathbf{\Theta}, \boldsymbol{m}) = \frac{1}{2}\sum_{j=1}^{t} \|\boldsymbol{\theta}_j\|^2 + \frac{1}{2}\gamma_1 \sum_{i=1}^{n} a_i r_i^2 + \gamma_2 \sum_{i=1}^{n}\sum_{j=1}^{t} q_{ij}\sigma(-q_{ij}) + \nu. \tag{14}$$

Algorithm 1. LIEML

```
Input: The training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) | 1 \leq i \leq n\}$, parameters $\mu_{min}$, $\eta_\mu$, $\lambda$, $\gamma_1$,
       $\gamma_2$, step size $\tau$, convergence criterion $\varepsilon$;
Output: Model parameters $\boldsymbol{\beta}$ and $\boldsymbol{m}$;
Initial the stacked matrix $\boldsymbol{Z}^{(0)}$;
Determin $\mu_1 > \mu_2 > \cdots > \mu_L = \mu_{min} > 0$;
for each $\mu \leftarrow \mu_1, \mu_2, ..., \mu_L$ do
    $k \leftarrow 0$;
    repeat
        Compute $\boldsymbol{A}^{(k)} \leftarrow \boldsymbol{Z}^{(k)} - \tau g(\boldsymbol{Z}^{(k)})$;
        Compute SVD of $\boldsymbol{U}^{(k)} \boldsymbol{\Lambda}^{(k)} (\boldsymbol{V}^{(k)})^\top \leftarrow \boldsymbol{A}^{(k)}$;
        Compute $\boldsymbol{Z}^{(k+1)} \leftarrow \boldsymbol{U}^{(k)} \max(\boldsymbol{\Lambda}^{(k)} - \tau\mu, 0)(\boldsymbol{V}^{(k)})^\top$;
        Project $\boldsymbol{Z}^{(k+1)}$ to feasible region $z_{(t+d+1)\cdot} \leftarrow \mathbf{1}^\top$;
        $k \leftarrow k + 1$;
    until $|T_1(\boldsymbol{Z}^{(k)}) - T_1(\boldsymbol{Z}^{(k-1)})| < \varepsilon$;
end
$\boldsymbol{U} \leftarrow z_{(1:t)\cdot}$;
Initial the model parameter $\boldsymbol{\beta}^{(0)}$ and $\boldsymbol{m}^{(0)}$;
Compute target function $T_2(\boldsymbol{\beta}^{(0)}, \boldsymbol{m}^{(0)})$ by Eq. 8;
$k \leftarrow 0$;
repeat
    Compute the descending directions $\boldsymbol{\beta}'$ and $\boldsymbol{m}'$ by Eq. 15;
    repeat
        Compute the model parameter $\boldsymbol{\beta}^{(k+1)}$ through a line search algorithm
        combining $\boldsymbol{\beta}^{(k)}$ and $\boldsymbol{\beta}'$;
        Compute the model parameter $\mathbf{m}^{(k+1)}$ through a line search algorithm
        combining $\mathbf{m}^{(k)}$ and $\mathbf{m}'$;
        Compute $T_2(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{m}^{(k+1)})$ by Eq. 8;
    until $T_2(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{m}^{(k+1)}) < T_2(\boldsymbol{\beta}^{(k)}, \boldsymbol{m}^{(k)})$;
    $k \leftarrow k + 1$;
until $|T_2(\boldsymbol{\beta}^{(k)}, \boldsymbol{m}^{(k)}) - T_2(\boldsymbol{\beta}^{(k-1)}, \boldsymbol{m}^{(k-1)})| < \varepsilon$;
```

The optimum of this piecewise quadratic problem can be integrated as solving a system of linear equations for $j = 1, ..., t$

$$\begin{bmatrix} \gamma_1 \boldsymbol{\Phi}^\top \boldsymbol{D}_a \boldsymbol{\Phi} + \boldsymbol{I} & \gamma_1 \boldsymbol{\Phi}^\top \boldsymbol{a} \\ \gamma_1 \boldsymbol{a}^\top \boldsymbol{\Phi} & \gamma_1 \mathbf{1}^\top \boldsymbol{a} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}'_j \\ m'_j \end{bmatrix} = \begin{bmatrix} \gamma_1 \boldsymbol{\Phi}^\top \boldsymbol{D}_a \boldsymbol{u}_j + \gamma_2 \boldsymbol{\Phi}^\top \boldsymbol{D}_j \boldsymbol{y}_j \\ \gamma_1 \boldsymbol{a}^\top \boldsymbol{u}_j + \gamma_2 (\boldsymbol{\sigma}_j)^\top \boldsymbol{y}_j \end{bmatrix}, \qquad (15)$$

where $\boldsymbol{\Phi} = [\varphi(\boldsymbol{x}_1), ..., \varphi(\boldsymbol{x}_n)]^\top$, $\boldsymbol{a} = [a_1, ..., a_n]^\top$, $(\boldsymbol{D}_a)_{ik} = a_i \delta_{ik}$ (δ_{ik} is the Kronecker's delta function), $(\boldsymbol{D}_j)_{ik} = \sigma(-q_{ij})\delta_{ik}$, $\boldsymbol{\sigma}_j = [\sigma(-q_{1j}), ..., \sigma(-q_{nj})]^\top$, $\boldsymbol{y}_j = [y_{1j}, ..., y_{nj}]^\top$. Then, the direction of the optimal solution ($\boldsymbol{\Theta}'$ and $\boldsymbol{m}'$) of Eq. 15 is used as the descending direction for the optimization of $T_2(\boldsymbol{\Theta}; \boldsymbol{m})$, and the solution for the next iteration ($\boldsymbol{\Theta}^{(k+1)}$ and $\boldsymbol{m}^{(k+1)}$) is obtained via a line search algorithm along this direction.

Table 1. Attributes of the benchmark multi-label data sets

Datasets	$\|S\|$	$dim(S)$	$L(S)$	$F(S)$	$LCard(S)$	$LDen(S)$	$DL(S)$	$PDL(S)$	Domain
cal500	502	68	174	numeric	26.044	0.150	502	1.000	audio
emotion	593	72	6	numeric	1.868	0.311	27	0.046	audio
medical	978	1449	45	nominal	1.245	0.028	94	0.096	text
llog	1460	1004	75	nominal	1.180	0.016	304	0.208	text
enron	1702	1001	53	nominal	3.378	0.064	753	0.442	text
image	2000	294	5	numeric	1.236	0.247	20	0.010	image
scene	2407	294	5	numeric	1.074	0.179	15	0.006	image
yeast	2417	103	14	numeric	4.237	0.303	198	0.082	biology
slashdot	3782	1079	22	nominal	1.181	0.054	156	0.041	text
corel5k	5000	499	374	nominal	3.522	0.009	3175	0.635	image

According to the representor's theorem [23], under fairly general conditions, a learning problem can be expressed as a linear combination of the training examples in the feature space, i.e., $\boldsymbol{\theta}^j = \sum_i \varphi(\boldsymbol{x}_i)\boldsymbol{\beta}_j = \boldsymbol{\Phi}\boldsymbol{\beta}_j$. If we replace this expression into Eq. 15, it will generate the inner product $< \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) >$, and then the kernel trick can be applied. After that the line search algorithm can be expressed in terms of $\boldsymbol{\beta}_j$ and m_j. Therefore the target function can be renamed as $T_2(\boldsymbol{\beta}, \boldsymbol{m})$. The pseudocode of LIEML is given in Algorithm 1.

4 Experiments

4.1 Experiment Configuration

Data Sets. A total of ten benchmark multi-label data sets[1] are employed for performance evaluation. Table 1 summarizes detailed characteristics of the real data sets, which are roughly organized in ascending order of the number of examples $|S|$. As shown in Table 1, the ten data sets cover a broad range of cases with diversified multi-label properties and thus serve as a solid basis for thorough comparative studies. For each multi-label data set S, we use $|S|$, $dim(S)$, $L(S)$, and $F(S)$ to represent its number of examples, number of features, number of class labels and feature type respectively. In addition, several multi-label statistics [24] are further used to characterize properties of the data set, including label cardinality $LCard(S)$, label density $LDen(S)$, distinct label sets $DL(S)$, and proportion of distinct label sets $PDL(S)$. Detailed definitions on these properties can be found in [24].

Comparing Algorithms. In this paper, we choose to compare the performance of LIEML against six well-established multi-label learning algorithms: BR [13], CLR [16], ECC [24], RAKEL [25], LP [20], and ML^2 [10].

[1] The data sets can be downloaded from: http://meka.sourceforge.net/#datasets and http://mulan.sourceforge.net/datasets.html.

Table 2. Performance of each comparing algorithm (mean±std.) on the benchmark multi-label data sets.

Comparing	*Ranking-loss* ↓				
algorithm	cal500	emotion	medical	llog	enron
LIEML	**0.177±0.002(1)**	0.221±0.011(2)	**0.026±0.005(1)**	0.144±0.008(2)	**0.076±0.002(1)**
BR	0.258±0.003(6)	0.233±0.016(6)	0.091±0.005(5)	0.328±0.007(6)	0.312±0.009(7)
CLK	0.239±0.026(5)	0.222±0.014(3)	0.123±0.026(7)	0.190±0.015(5)	0.089±0.002(2)
ECC	0.205±0.004(4)	0.227±0.017(4)	0.032±0.007(2)	0.154±0.009(3)	0.120±0.004(5)
RAKEL	0.444±0.005(7)	0.254±0.020(7)	0.095±0.033(6)	0.412±0.010(7)	0.241±0.005(6)
LP	0.181±0.003(2)	**0.182±0.012(1)**	0.034±0.006(4)	**0.125±0.005(1)**	0.091±0.003(4)
ML^2	0.188±0.002(3)	0.231±0.012(5)	0.032±0.005(2)	0.158±0.005(4)	0.090±0.012(3)
Comparing	*Ranking-loss* ↓				
algorithm	image	scene	yeast	slashdot	corel5k
LIEML	**0.143±0.006(1)**	0.065±0.003(2)	0.170±0.002(2)	**0.093±0.002(1)**	0.121±0.002(2)
BR	0.314±0.014(7)	0.229±0.010(7)	0.190±0.004(4)	0.240±0.008(6)	0.416±0.003(6)
CLK	0.294±0.009(5)	0.127±0.003(4)	0.198±0.003(6)	0.260±0.007(7)	**0.114±0.002(1)**
ECC	0.276±0.005(4)	0.151±0.005(5)	0.190±0.003(4)	0.123±0.004(3)	0.292±0.003(5)
RAKEL	0.311±0.010(6)	0.205±0.008(6)	0.245±0.004(7)	0.190±0.005(5)	0.627±0.004(7)
LP	0.181±0.008(3)	0.087±0.006(3)	0.174±0.004(3)	0.132±0.005(4)	0.145±0.002(3)
ML^2	**0.143±0.007(1)**	**0.064±0.003(1)**	**0.168±0.003(1)**	0.095±0.003(2)	0.163±0.003(4)
Comparing	*One-error* ↓				
algorithm	cal500	emotion	medical	llog	enron
LIEML	**0.119±0.014(1)**	0.350±0.023(2)	**0.166±0.011(1)**	0.766±0.020(3)	**0.225±0.011(1)**
BR	0.921±0.025(7)	0.375±0.027(6)	0.297±0.036(6)	0.884±0.011(6)	0.648±0.019(7)
CLK	0.331±0.111(6)	0.356±0.030(5)	0.688±0.143(7)	0.900±0.019(7)	0.376±0.017(4)
ECC	0.191±0.021(4)	0.353±0.040(4)	0.182±0.019(3)	0.785±0.009(4)	0.424±0.013(6)
RAKEL	0.286±0.039(5)	0.392±0.035(7)	0.208±0.071(4)	0.838±0.014(5)	0.412±0.016(5)
LP	0.120±0.015(2)	**0.303±0.027(1)**	0.213±0.021(5)	0.748±0.011(2)	0.311±0.013(3)
ML^2	0.141±0.016(3)	0.352±0.021(3)	0.179±0.019(2)	**0.683±0.018(1)**	0.258±0.090(2)
Comparing	*One-error* ↓				
algorithm	image	scene	yeast	slashdot	corel5k
LIEML	**0.271±0.010(1)**	0.197±0.006(2)	**0.226±0.009(1)**	0.387±0.009(2)	0.650±0.006(2)
BR	0.538±0.019(7)	0.475±0.014(7)	0.285±0.008(7)	0.734±0.017(6)	0.919±0.006(7)
CLK	0.514±0.014(5)	0.371±0.008(4)	0.270±0.007(6)	0.979±0.003(7)	0.721±0.007(4)
ECC	0.486±0.018(4)	0.373±0.008(5)	0.256±0.007(5)	0.481±0.014(4)	0.699±0.006(3)
RAKEL	0.515±0.017(6)	0.444±0.012(6)	0.251±0.008(4)	0.453±0.005(3)	0.819±0.010(6)
LP	0.353±0.017(3)	0.270±0.016(3)	0.241±0.011(3)	0.558±0.009(5)	0.755±0.005(5)
ML^2	0.272±0.009(2)	**0.194±0.008(1)**	0.228±0.009(2)	**0.382±0.009(1)**	**0.647±0.007(1)**

For LIEML, the first value μ_1 for μ is set to $\sigma\eta_\mu$, where σ is the largest singular value of Z and the decay parameter η_μ is set to 0.25. The last value μ_{min} for μ is set to 10^{-5}. The parameter λ is chosen among $\{10^{-1}, 1, 10, 10^2\}$. The step size τ is set to $\min(\frac{3.8tn}{\lambda}, dn)$. The parameters γ_1 and γ_2 are set to 1 and 0.1 respectively. The convergence criterion ε is set to 10^{-5}. The kernel function in LIEML is radial basis function (RBF). The four algorithms, including BR, CLR, ECC, and RAKEL, are operated on the MULAN multi-label learning package. The base classifier for the four algorithms is logistic regression model. The ensemble size for ECC is set to 30, and for RAKEL is set to $2t$ with $k = 3$. The parameter α in LP is set to 0.5. The number of neighbors K for ML^2 is set to $t + 1$.

Table 3. Performance of each comparing algorithm (mean±std.) on the benchmark multi-label data sets.

Comparing algorithm	*Hamming-loss* ↓				
	cal500	emotion	medical	llog	enron
LIEML	**0.136±0.001(1)**	0.251±0.006(3)	0.015±0.001(3)	**0.015±0.000(1)**	**0.047±0.001(1)**
BR	0.214±0.004(7)	0.265±0.013(5)	0.022±0.003(5)	0.052±0.003(7)	0.105±0.003(7)
CLK	0.165±0.005(5)	0.270±0.011(7)	0.024±0.002(6)	0.019±0.002(5)	0.072±0.002(6)
ECC	0.146±0.002(4)	0.254±0.013(4)	0.013±0.001(2)	0.016±0.000(2)	0.064±0.001(5)
RAKEL	0.138±0.002(2)	0.269±0.011(6)	**0.010±0.003(1)**	0.017±0.001(4)	0.058±0.001(3)
LP	0.167±0.004(6)	**0.223±0.007(1)**	0.017±0.001(4)	0.016±0.000(2)	0.063±0.003(4)
ML^2	0.138±0.002(2)	0.243±0.010(2)	0.283±0.027(7)	0.021±0.001(6)	0.051±0.001(2)

Comparing algorithm	*Hamming-loss* ↓				
	image	scene	yeast	slashdot	corel5k
LIEML	0.160±0.003(2)	0.083±0.002(2)	**0.195±0.003(1)**	**0.040±0.001(1)**	**0.009±0.000(1)**
BR	0.287±0.008(6)	0.184±0.005(7)	0.219±0.003(6)	0.130±0.003(7)	0.027±0.000(7)
CLK	0.305±0.005(7)	0.181±0.004(6)	0.222±0.002(7)	0.058±0.001(5)	0.011±0.001(3)
ECC	0.244±0.005(4)	0.133±0.002(4)	0.216±0.002(5)	0.049±0.001(4)	0.015±0.001(5)
RAKEL	0.286±0.007(5)	0.171±0.005(5)	0.202±0.003(3)	0.048±0.001(3)	0.012±0.001(4)
LP	0.190±0.005(3)	0.127±0.005(3)	0.214±0.004(4)	0.060±0.002(6)	0.024±0.000(6)
ML^2	**0.156±0.004(1)**	**0.076±0.003(1)**	0.196±0.003(2)	0.043±0.001(2)	0.010±0.001(2)

Comparing algorithm	*Coverage* ↓				
	cal500	emotion	medical	llog	enron
LIEML	**0.744±0.007(1)**	0.347±0.010(2)	**0.041±0.006(1)**	**0.149±0.007(1)**	**0.226±0.006(1)**
BR	0.852±0.014(6)	0.363±0.015(6)	0.118±0.007(6)	0.377±0.008(6)	0.601±0.014(7)
CLK	0.794±0.010(5)	0.351±0.016(3)	0.143±0.030(7)	0.225±0.016(5)	0.243±0.006(3)
ECC	0.788±0.008(4)	0.356±0.013(4)	0.048±0.009(2)	0.192±0.010(4)	0.300±0.009(5)
RAKEL	0.971±0.001(7)	0.381±0.019(7)	0.117±0.040(5)	0.459±0.011(7)	0.523±0.008(6)
LP	0.747±0.007(2)	**0.318±0.031(1)**	0.052±0.001(4)	0.159±0.006(2)	0.242±0.005(2)
ML^2	0.780±0.008(3)	0.357±0.009(5)	0.048±0.008(2)	0.162±0.008(3)	0.256±0.017(4)

Comparing algorithm	*Coverage* ↓				
	image	scene	yeast	slashdot	corel5k
LIEML	**0.168±0.006(1)**	**0.068±0.003(1)**	0.454±0.004(2)	**0.109±0.002(1)**	0.276±0.004(2)
BR	0.301±0.012(7)	0.207±0.009(7)	0.474±0.005(4)	0.259±0.009(6)	0.758±0.003(6)
CLK	0.286±0.008(5)	0.120±0.007(3)	0.492±0.006(6)	0.272±0.007(7)	**0.267±0.004(1)**
ECC	0.272±0.005(4)	0.141±0.004(4)	0.476±0.004(5)	0.139±0.004(3)	0.562±0.007(5)
RAKEL	0.298±0.010(6)	0.186±0.006(6)	0.558±0.006(7)	0.212±0.005(5)	0.886±0.004(7)
LP	0.198±0.007(3)	0.171±0.009(5)	**0.451±0.005(1)**	0.148±0.005(4)	0.328±0.005(3)
ML^2	**0.168±0.007(1)**	**0.067±0.003(1)**	0.454±0.004(2)	0.112±0.003(2)	0.372±0.006(4)

Evaluation Metrics. We use five evaluation metrics widely-used in multi-label learning in this paper, i.e., *Ranking-loss*, *One-error*, *Hamming-loss*, *Coverage*, and *Average precision* [2]. For all the five multi-label metrics, their values vary between [0, 1]. Furthermore, for average precision, the larger the values the better the performance. While for the other four metrics, the smaller the values the better the performance. These metrics serve as good indicators for comprehensive comparative studies as they evaluate the performance of the learned models from various aspects.

4.2 Experimental Results

Tables 2, 3, and 4 report the detailed experimental results of each comparing algorithm on the benchmark multi-label data sets. On each data set, 50% examples are randomly sampled without replacement to form the training set, and the

Table 4. Performance of each comparing algorithm (mean±std.) on the benchmark multi-label data sets.

Comparing	*Average precision* ↑				
algorithm	cal500	emotion	medical	llog	enron
LIEML	**0.512±0.003(1)**	0.745±0.011(2)	**0.872±0.011(1)**	0.347±0.014(3)	**0.698±0.008(1)**
BR	0.300±0.005(7)	0.730±0.015(6)	0.762±0.022(5)	0.215±0.009(5)	0.381±0.009(7)
CLK	0.395±0.042(5)	0.742±0.016(3)	0.400±0.062(7)	0.194±0.018(7)	0.610±0.008(4)
ECC	0.463±0.006(4)	0.740±0.021(4)	0.860±0.015(3)	0.342±0.009(4)	0.559±0.008(5)
RAKEL	0.353±0.006(6)	0.717±0.023(7)	0.700±0.234(6)	0.197±0.013(6)	0.539±0.006(6)
LP	0.496±0.005(3)	**0.779±0.012(1)**	0.837±0.018(4)	0.390±0.009(2)	0.661±0.007(3)
ML^2	0.501±0.003(2)	0.737±0.013(5)	0.865±0.014(2)	**0.405±0.013(1)**	0.681±0.053(2)

Comparing	*Average precision* ↑				
algorithm	image	scene	yeast	slashdot	corel5k
LIEML	**0.824±0.006(1)**	0.884±0.004(2)	**0.766±0.005(1)**	0.708±0.006(2)	**0.305±0.003(1)**
BR	0.649±0.012(7)	0.692±0.010(7)	0.734±0.004(5)	0.427±0.014(6)	0.123±0.003(6)
CLK	0.666±0.008(5)	0.778±0.004(4)	0.730±0.003(6)	0.250±0.007(7)	0.274±0.002(3)
ECC	0.685±0.008(4)	0.766±0.005(5)	0.741±0.004(4)	0.628±0.009(3)	0.264±0.003(4)
RAKEL	0.661±0.010(6)	0.713±0.008(6)	0.720±0.005(7)	0.617±0.004(4)	0.122±0.004(7)
LP	0.775±0.009(3)	0.842±0.009(3)	0.753±0.006(3)	0.579±0.009(5)	0.241±0.002(5)
ML^2	**0.824±0.006(1)**	**0.885±0.004(1)**	0.765±0.005(2)	**0.711±0.005(1)**	0.297±0.002(2)

rest 50% examples are used to form the test set. The sampling process is repeated for ten times, and the average predictive performance across ten training/testing trials are recorded. For each evaluation metric, ↓ indicates the smaller the better while ↑ indicates the larger the better. The best performance among the seven comparing algorithms is shown in boldface.

From the result tables we can see, across all the evaluation metrics, LIEML ranks 1*st* in the most cases. Note that ML^2 ranks 1*st* in the second most cases, and the number of cases where LIEML ranks 1*st* is 1.73% larger than the number of cases where ML^2 ranks 1*st*. Thus LIEML achieves rather competitive performance against ML^2. Because the model of LE in LIEML is linear, but nonlinear in ML^2, it is uneasy for LIEML to beat ML^2 with the less efficient linear way. Apart from ML^2, LIEML obviously outdistances the other five well-established multi-label learning algorithms. Therefore, the results of the experiment validate the effectiveness of our LIEML algorithm for multi-label learning.

5 Conclusion

This paper proposes a novel multi-label learning method named LIEML, which enhances the labeling information by extending logical labels into numerical labels. We first construct a stacked matrix where the feature and the label matrix are placed vertically. Then, we enhance the labeling information by leveraging the underlying low-rank structure in the stacked matrix. After that, we induce the multi-label predictive model by the learning procedure from training examples with numerical labels. Experimental results clearly validate the advantage of LIEML against the state-of-the-art multi-label learning approaches. In the future, we will explore if there exist better ways to make use of the labeling information for multi-label learning.

Acknowledgements. This research was supported by the National Key Research & Development Plan of China (No. 2017YFB1002801), the National Science Foundation of China (61622203), the Jiangsu Natural Science Funds for Distinguished Young Scholar (BK20140022), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Collaborative Innovation Center of Wireless Communications Technology.

References

1. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-09823-4_34
2. Zhang, M.-L., Zhou, Z.-H.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2014)
3. Rubin, T., Chambers, A., Smyth, P., Steyvers, M.: Statistical topic models for multi-label document classification. Mach. Learn. **88**(1–2), 157–208 (2012)
4. Yang, B., Sun, J.-T., Wang, T., Chen, Z.: Effective multilabel active learning for text classification. In: Proceedings of 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, pp. 917–926 (2009)
5. Cabral, R., Torre, F., Costeira, J., Bernardino, A.: Matrix completion for multi-label image classification. In: Proceedings of 24th International Conference on Neural Information Processing Systems, Granada, Spain, pp. 190–198 (2011)
6. Wang, H., Huang, H., Ding, C.: Image annotation using multi-label correlated green's function. In: Proceedings of 12th IEEE International Conference on Computer Vision, Kyoto, Japan, pp. 2029–2034 (2009)
7. Lo, H.-Y., Wang, J.-C., Wang, H.-M., Lin, S.-D.: Costsensitive multi-label learning for audio tag annotation and retrieval. IEEE Trans. Multimedia **13**(3), 518–529 (2011)
8. Sanden, C., Zhang, J.-Z.: Enhancing multi-label music genre classification through ensemble techniques. In: Proceedings of 34th ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, pp. 705–714 (2011)
9. Wang, J., Zhao, Y., Wu, X., Hua, X.-S.: A transductive multi-label learning approach for video concept detection. Pattern Recogn. **44**(10–11), 2274–2286 (2011)
10. Hou, P., Geng, X., Zhang, M.-L.: Multi-label manifold learning. In: Proceedings of 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, pp. 1680–1686 (2016)
11. Liu, G., Lin, Z.-C., Yang, S.-C., Sun, J., Yu, Y., Ma, Y.: Robust recovery of subspace structures by low-rank representation. IEEE Trans. Pattern Anal. Mach. Intel. **35**(1), 171–184 (2013)
12. Eriksson, B., Balzano, L., Nowak, R.: High-rank matrix completion. In: Proceedings of 15th International Conference on Artificial Intelligence Statistics, La Palma, Canary Islands, vol. 20, pp. 373–381 (2012)
13. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recogn. **37**(9), 1757–1771 (2004)
14. Zhang, M.-L., Zhou, Z.-H.: ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn. **40**(7), 2038–2048 (2007)
15. Elisseeff, A., Weston, J.: A kernel method for multilabelled classification. In: Proceedings of Advance Neural Information Processing Systems 14, Vancouver, Canada, pp. 681–687 (2001)

16. Frnkranz, J., Hllermeier, E., Menca, E., Brinker, K.: Multilabel classification via calibrated label ranking. Mach. Learn. **73**(2), 133–153 (2008)
17. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multilabel classification. IEEE Trans. Knowl. Data Eng. **23**(7), 1079–1089 (2011)
18. Tai, F., Lin, H.-T.: Multilabel classification with principal label space transformation. Neural Comput. **24**(9), 2508–2542 (2012)
19. Sun, L., Ji, S., Ye, J.: Canonical correlation analysis for multilabel classification: a least-squares formulation, extensions, and analysis. IEEE Trans. Pattern Anal. Mach. Intel. **33**(1), 194–200 (2011)
20. Li, Y.-K., Zhang, M.-L., Geng, X.: Leveraging implicit relative labeling-importance information for effective multi-label learning. In: Proceedings of 15th IEEE International Conference on Data Mining, Atlantic City, NJ, pp. 251–260 (2015)
21. Ma, S.-Q., Goldfarb, D., Chen, L.-F.: Fixed point and bregman iterative methods for matrix rank minimization. Math. Programm. **128**(1–2), 321–353 (2011)
22. Pérez-Cruz, F., Vázquez, A., Alarcón-Diana, P., Artés-Rodríguez, A.: An IRWLS procedure for SVR. In: 10th European Conference on Signal Processing, Tampere, Finland, pp. 1–4 (2000)
23. Schlkopf, B., Smola, A.: Learning with Kernels. The MIT Press, Berlin (2001)
24. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Mach. Learn. **85**(3), 333–359 (2011)
25. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random klabelsets for multilabel classification. IEEE Trans. Knowl. Data Eng. **23**(7), 1079–1089 (2011)