Robust Extreme Multi-label Learning

Chang Xu* xuchang@pku.edu.cn
Dacheng Tao† Dacheng.Tao@uts.edu.au
Chao Xu* xuchao@cis.pku.edu.cn

Key Laboratory of Machine Perception (Ministry of Education), Peking University*, Beijing, China
Centre for Quantum Computation and Intelligent Systems, University of Technology†, Sydney

ABSTRACT

Tail labels in the multi-label learning problem undermine the low-rank assumption. Nevertheless, this problem has rarely been investigated. In addition to using the low-rank structure to depict label correlations, this paper explores and exploits an additional sparse component to handle tail labels behaving as outliers, in order to make the classical low-rank principle in multi-label learning valid. The divide-and-conquer optimization technique is employed to increase the scalability of the proposed algorithm while theoretically guaranteeing its performance. A theoretical analysis of the generalizability of the proposed algorithm suggests that it can be improved by the low-rank and sparse decomposition given tail labels. Experimental results on real-world data demonstrate the significance of investigating tail labels and the effectiveness of the proposed algorithm.

CCS Concepts

•**Computing methodologies** → *Supervised learning;*

Keywords

Multi-label Learning; Robust Algorithm

1. INTRODUCTION

In contrast to conventional single-label learning, in which each example is assigned only one label, multi-label learning evaluates examples with multiple labels. Many real-world applications use multi-label learning [30] including text categorization and image/video annotation [7]. The rapid evolution of information techniques has fueled the emergence of large-scale multi-label applications with huge numbers of labels. For example, given over a million labels (categories) on Wikipedia, one might wish to build a classifier to annotate a new article or web page with a subset of the most relevant categories. In another example, taking billions of YouTube videos as distinct labels, a task might be to recommend a ranked list of labels to a single user. Hence, extreme multi-label learning with an extremely large number of labels has become an important research focus.

Binary relevance (BR) [23] seeks to independently train a classifier for each label. BR is a straightforward approach for multi-label learning, but due to prohibitive training and prediction costs arising from large numbers of labels, this method becomes less useful. A number of embedding-based approaches have been proposed to overcome this extreme multi-label learning problem that reduce the effective number of labels. The approaches assume that the label matrix is low rank. Different techniques can be used to compress and decompress label vectors, including compressed sensing [14], Bloom filters [9], SVD [22], landmark labels [2, 5], and output codes [31].

The low-rank label matrix assumption in embedding methods is violated in many real-world applications due to the presence of tail labels occurring in a handful of data points. Histograms of the label matrices on the *wiki10*, *Delicious-L* and *Amazon* datasets are shown in Figure 1. There are more than 10^4 labels which occur in at most 2 examples on each dataset, such that they are not well approximated by any linear low-dimensional basis. This tail label issue is frequently and persistently neglected in multi-label learning. Recently, instead of projecting label vectors into a linear low-rank subspace, [4] addressed the tail label problem by learning embeddings that non-linearly captured label correlations by preserving the pairwise distances between label vectors. Although this provides an alternative approach to handling tail labels, we must also ask whether low-rank based approaches really are now redundant.

Here we revisit the classical low-rank principle in multi-label learning and suppress the influence of tail labels. Tail labels can be regarded as label matrix outliers, inspiring us to decompose the label matrix into a low-rank part that depicts label correlations and a sparse part that captures tail labels. Various effective embedding techniques are then applicable. To improve scalability and leverage the growing availability of parallel computing architectures, the divide-and-conquer approach is adopted to optimize the resulting objective function, whose performance can be theoretically guaranteed with high probability. Our theoretical analysis shows that the proposed low-rank and sparse decomposition is useful for improving the generalizability of multi-label learning algorithms. Experiments on real-world data demonstrate the significance of studying tail labels in multi-label learning and the effectiveness of the proposed algorithm.

KDD'16 August 13–17, 2016, San Francisco, CA, USA

DOI: 10.475/123_4

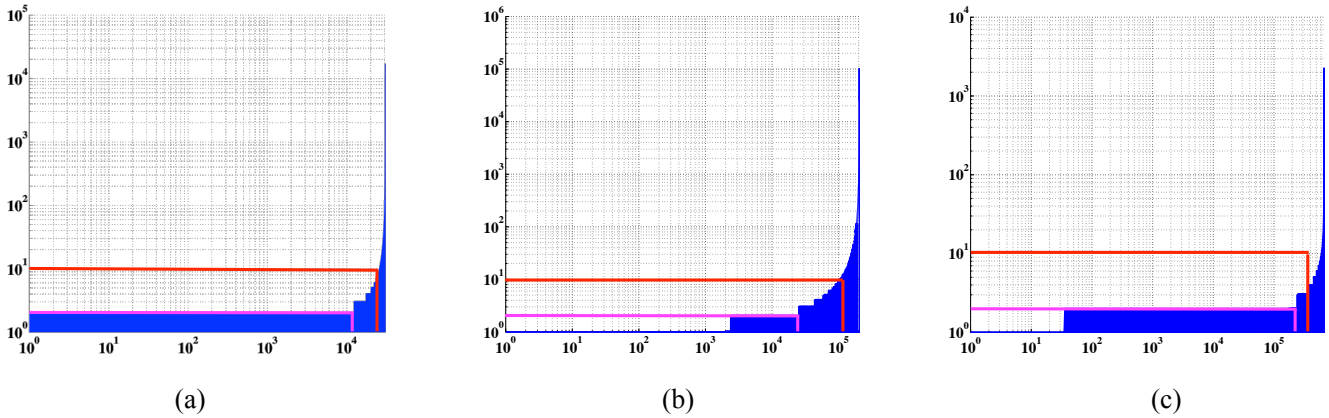

Figure 1: The number of examples in which each label is present on the (a) Wiki10 (b) Delicious-L and (c) Amazon datasets. The horizontal axis indicates the index of label, while the vertical axis indicates the number of associated examples.

2. RELATED WORK

Broadly speaking, existing multi-label learning algorithms can be categorized into two groups: algorithm adaptation and problem transformation methods. Algorithm adaptation methods adapt, extend and customize an existing machine learning algorithms for the task of multi-label learning, while problem transformation methods transform the multi-label learning problem into one or more single-label classification or regression problems. Representative examples include boosting [19, 25, 15], decision trees [24, 16], neural networks [28, 10], support vector machines [12, 13] and k nearest neighbors classifier [29, 20].

The key in multi-label learning is modeling inter-label correlations and using them for label vector prediction. In many applications such as text categorization and functional genomics, the labels are often organized in the form of a tree or directed acyclic graph, so that label relationship can be exploited from the knowledge resources as prior knowledge for multi-label learning [6, 13]. In most real-world tasks, however, prior knowledge of label relationship is often unavailable. It is thus necessary to model the label relationship directly. For example, Sun et al. [21] proposed to employ hypergraphs to exploit the higher-order relations among multiple instances sharing the same label in multi-label learning. They constructed a hyperedge for each label, and included all instances annotated with a common label into one hyperedge, thus capturing their joint similarity. Zhang and Zhang [27] used a Bayesian network to characterize the dependence structure between multiple labels, and learned a binary classifier for each label by treating its parental labels in the dependence structure as additional input features. However, as the number of labels keeps growing, these algorithms are usually computationally infeasible.

Embedding-based approaches are significant for handling extreme multi-label learning problem by reducing the effective number of labels. Generally, they assume that the label matrix is low-rank, and then project label vectors into a low-dimensional subspace. Hence, instead of directly learning to predict the original high-dimensional label vector of each example, the training complexity is largely decreased by first learning the predictor of embedded label vectors, and then a decompression operation is employed to lift the embedded label vectors back to the original label space.

Various compression and decompression techniques have been exploited by existing embedding methods. Hsu et al. [14] addressed classification problem with a large number of labels via a three-step approach. First, random transformation is used to project the high-dimensonal label vector into a low-dimensional space; next, a regression model is trained to predict each dimension of the transformed label vector; finally, for a test example, its predicted label vector in the low-dimensional space is projected back to the original label space. Considering the drawback of random transformation in [14], Tai and Lin [22] proposed the principal label space transformation, which uses principal component analysis (PCA) to accomplish the compression operation on the high-dimensional label vector. Since PCA in the label space only focuses on minimizing the encoding error between high-dimensional feature vectors and their low-dimensional representations [22], Chen and Lin [8] proposed conditional principal label space transformation, which improves [22] by simultaneously considering the label encoding error and training error in the low-dimensional label space. Based on canonical correlation analysis (CCA), Zhang and Schneider [31] also took both feature matrix and label matrix into consideration. After that, a maximum margin formulation was developed to learn an output coding, which is predictive and discriminative so that the codings for different label vectors are easy to predict and significantly different from each other. Instead of using label transformation, Balasubramanian and Lebanon [2] proposed to train only a small subset of the labels, which come from the original labels, so that the difficulty of the learning problems can be decreased. Supposing that the non-selected labels are to be faithfully and easily constructed from the selected ones, a group-sparse learning problem is investigated to discover the optimal label subset [2]. However, the structured sparsity optimization problem in [2] is computationally expensive, especially when there are a lot of labels to select from. Bi and Kwok [5] alleviated this problem by proposing an efficient label selection method based on randomized sampling. Following the assumption in [2], Bi and Kwok [5] designed the sampling probability of each label using its leverage score in the best rank-k subspaces of the label matrix.

Recently, Yu et al. [26] modeled multi-label classification as a general empirical risk minimization (ERM) problem with a low-rank constraint, which generalizes both label and feature dimensionality reduction. Given squared-L_2 loss, LEML algorithm in [26] has a closed form solution, and can be reduced to the conditional principal label space transformation algorithm in [8]. Various loss functions and regularizers are applicable in this ERM framework for preventing overfitting and increasing scalability. However, as suggested by [4], the low-rank assumption in embedding-based approaches is easily violated by tail labels in real-world datasets with a large number of labels. Instead of globally projecting high-dimensional label vectors into a low-rank subspace, SLEEC algorithm in [4] learns low-dimensional embedding which non-linearly capture label correlations by preserving the pairwise distances between only the closet label vectors. Regressors are then trained to predict the embedded label vector for each example. During prediction, rather than using a decomposition matrix, SLEEC uses a k-nearest neighbor (kNN) classifier in the embedding space, which leverages the fact that nearest neighbors have been persevered during training.

3. PROBLEM FORMULATION

Given a training data set $\{(x_1, y_1), \cdots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^d$ is the feature vector of the i-th example, and $y_i \in \{-1, 1\}^L$ is the corresponding label vector, the feature matrix is denoted as $X = [x_1; \cdots; x_n] \in \mathbb{R}^{n\times d}$ and the label matrix is $Y = [y_1; \cdots; y_n] \in \{-1, 1\}^{n\times L}$. If $Y_{ij} = 1$, example x_i will have label-j; otherwise, there is no label-i for example x_j. Multi-label learning aims to learn a hypothesis $f : \mathbb{R}^d \to \{-1, 1\}^L$ that accurately predicts the label vector for a given example.

There are a large number of labels in the extreme multi-label learning setting. Tail labels, which only occur with several examples, cannot be ignored, and a number of rows in the label matrix Y will have plenty of -1-valued entries while occasionally being dotted with several 1-valued entries. Due to the existence of these rows, the classical low-rank assumption on the label matrix is violated. We attempt to modernize the low-rank principle by carefully formulating the tail labels in the extreme multi-label learning problem.

3.1 Robust Extreme Multi-label Learning

One reasonable approach to interpret the label matrix in multi-label learning is to assume that the label matrix can be well approximated using a low-dimensional subspace. This assumption has been well justified in many practical situations. A general model can be written as

$$\begin{aligned} &\min_{\widehat{Y},W} \|Y - \widehat{Y}\|_F^2 + \lambda \ell(f(X, W), \widehat{Y}) \\ &\text{s.t. } rank(\widehat{Y}) \le k. \end{aligned} \tag{1}$$

where $\widehat{Y}$ is the low-rank approximation of Y, and loss function $\ell(\cdot)$ is employed to penalize the loss generated by a multi-label predictor parameterized by W.

Problem (1) has been studied for decades, and various techniques have been used to formulate the low-rank $\widehat{Y}$, the predictor $f(\cdot)$ and the loss function $\ell(\cdot)$. However, as mentioned above, tail labels damage the low-rank assumption over Y and render the estimated $\widehat{Y}$ arbitrarily far from the true Y; the learned multi-label predictor is, therefore, seriously influenced as a result. A method that can extract the low-rank components from Y even in the presence of tail labels would be desirable.

To achieve this, we treat tail labels as outliers and decompose the label matrix to

$$Y = \widehat{Y}_L + \widehat{Y}_S, \tag{2}$$

where $\widehat{Y}_L$ is of low rank and depict label correlations and $\widehat{Y}_S$ is the sparse component capturing the influence of tail labels. These two components can be obtained by solving the following objective:

$$\begin{aligned} &\min_{\widehat{Y}_L, \widehat{Y}_S} \|Y - \widehat{Y}_L - \widehat{Y}_S\|_F^2 \\ &\text{s.t. } rank(\widehat{Y}_L) \le k, \quad card(\widehat{Y}_S) \le s. \end{aligned} \tag{3}$$

Given the low rank and sparse components $\widehat{Y}_L$ and $\widehat{Y}_S$, we expect to learn regression models that predict them using the input features. That is, we require that $\widehat{Y}_L \approx WX$ and $\widehat{Y}_S \approx HX$, where $W, H \in \mathbb{R}^{d\times L}$. Hence, problem (3) can be reformulated as:

$$\begin{aligned} &\min_{W,H} \|Y - XW - XH\|_F^2 \\ &\text{s.t. } rank(XW) \le k, \quad card(XH) \le s. \end{aligned} \tag{4}$$

Since the rank and card constraints tend to increase the optimization complexity, we employ two popular matrix factorization heuristics (to encourage low-rankness) and $L1$-norm minimization (to encourage sparsity) to relax the constraints in problem (4), such that

$$\begin{aligned} \min_{U,V,H} \|Y - XUV - XH\|_F^2 + \lambda_1 \|H\|_F^2 \\ + \lambda_2(\|U\|_F^2 + \|V\|_F^2) + \lambda_3 \|XH\|_1, \end{aligned} \tag{5}$$

where $\{\lambda_1, \lambda_2, \lambda_3\}$ are positive constants, W is supposed to have $W = UV$ given $U \in \mathbb{R}^{d\times k}$ and $V \in \mathbb{R}^{k\times L}$, and an $L2$-regularization has been included for H. It is expected that by solving problem (5), we can obtain a low-rank function (i.e., $W = UV$) and a sparse function (i.e., H), which together are used for multi-label prediction.

4. OPTIMIZATION

In this section, we first present the basic optimization method for problem (5) and then adopt the divide-and-conquer strategy to develop an optimization method applicable to extreme multi-label learning.

4.1 Basic Optimization Method

Problem (5) can be solved by alternatively solving the following three subproblems until convergence:

$$V = \arg\min_V \|Y - XUV - XH\|_F^2 + \lambda_2 \|V\|_F^2 \tag{6a}$$

$$U = \arg\min_U \|Y - XUV - XH\|_F^2 + \lambda_2 \|U\|_F^2 \tag{6b}$$

$$\begin{aligned} H = \arg\min_H \|Y - XUV - XH\|_F^2 + \lambda_1 \|H\|_F^2 \\ + \lambda_3 \|XH\|_1. \end{aligned} \tag{6c}$$

In the following, we illustrate the optimization of these three subproblems respectively.

4.1.1 Solving V

In problem (6a), updating V is simple since each column v_j of V can be independently updated:

$$\min_{v_j} \sum_{i=1}^{n} \|Y_{ij} - x_i U v_j - (XH)_{ij}\|^2 + \lambda_2 \|v_j\|_2^2, \qquad (7)$$

which is easy to solve in a closed form as the dimension of v_j (i.e. k) is generally small. Setting the gradient of Eq. (7) w.r.t. v_j to zero, the optimal v_j^* corresponding to the j-th label can be obtained as

$$v_j^* = \Big[\sum_{i=1}^{n} (x_i U)^T x_i U + \lambda_2 I\Big]^{-1} \Big[\sum_{i=1}^{n} \big(Y_{ij} - (XH)_{ij}\big)(x_i U)^T\Big]$$

4.1.2 Solving U

Problem (6b) is equivalent to

$$\min_{u} \sum_{i=1}^{n} \sum_{j=1}^{L} \|Y_{ij} - u^T \widetilde{X}_{ij} - (XH)_{ij}\|^2 + \lambda_2 \|u\|^2, \qquad (8)$$

where $u \in \mathbb{R}^{dk}$ denotes $vec(U)$, and $\widetilde{X}_{ij} = vec(x_i v_j)$. There is a closed form solution for this problem,

$$u^* = \Big[\sum_{i=1}^{n} \sum_{j=1}^{L} \widetilde{X}_{ij} (\widetilde{X}_{ij})^T + \lambda_2 I\Big]^{-1} \Big[\sum_{i=1}^{n} \sum_{j=1}^{L} \big(Y_{ij} - (XH)_{ij}\big) \widetilde{X}_{ij}\Big]$$

However, it is inefficient to compute the closed form solution for the above problem when d is large due to the huge computational cost of inverting a $dk \times dk$ matrix. Instead, it is more appropriate to employ efficient gradient descent methods (e.g. conjugate gradient descent) for optimization.

4.1.3 Solving H

In problem (6c), given $\widetilde{Y} = Y - XUV$, each column H_j can be independently solved,

$$\min_{H_j} \sum_{i=1}^{n} \|\widetilde{Y}_{ij} - x_i H_j\|_F^2 + \lambda_1 \|H_j\|_F^2 + \lambda_3 \|X H_j\|_1. \qquad (9)$$

The $L1$ norm in problem (9) involves both H_j and X, which makes the problem non-smooth and disallows the standard prox function-based optimization methods. One way to circumvent this difficulty is by introducing an auxiliary variable $Z_j = XH_j$ and transforming problem (9) into

$$\begin{aligned} \min_{H_j, Z_j, \mu} \; & \sum_{i=1}^{n} \|\widetilde{Y}_{ij} - x_i H_j\|_F^2 + \lambda_1 \|H_j\|_F^2 + \lambda_3 \|Z_j\|_1 \\ & + \rho \|X H_j - Z_j + \mu\|_F^2, \end{aligned} \qquad (10)$$

where $\rho > 0$ is the penalty parameter, and μ is the scaled dual variable.

Fixing H_j and μ, problem (10) is reduced to

$$\min_{Z_j} \; \|X H_j - Z_j + \mu\|_F^2 + \frac{\lambda_3}{\rho} \|Z_j\|_1. \qquad (11)$$

The optimal Z_j^* can be obtained via soft thresholding operation [11],

$$Z_j^* = soft(XH_j + \mu, \frac{\lambda_3}{\rho}), \qquad (12)$$

where

$$soft(a, b) = sign(a) \max(|a| - b, 0). \qquad (13)$$

Algorithm 1 Robust Extreme Multi-label Learning

Input: X, Y, $t \geq 1$
For $i = 1, \cdots, t$ **do In Parallel**
  Sample $(Y)_i \subseteq Y$
  repeat
    Solve $(\widehat{V})_i$ from Problem (6a)
    Solve $(\widehat{U})_i$ from Problem (6b)
    Solve $(\widehat{H})_i$ from Problem (6c)
  until Convergence
  $(\widehat{W})_i = (\widehat{U})_i (\widehat{V})_i$
End
ColumnProjection $([(\widehat{W})_1, \cdots, (\widehat{W})_t], (\widehat{W})_1)$

Fixing Z_j and μ, H_j can be solved from the following objective function:

$$\min_{H_j} \sum_{i=1}^{n} \|\widetilde{Y}_{ij} - x_i H_j\|_F^2 + \lambda_1 \|H_j\|_F^2 + \rho \|X H_j - Z_j + \mu\|_F^2. \qquad (14)$$

The gradient w.r.t. H_j is calculated as

$$\begin{aligned} \nabla_{H_j} \mathcal{J} = & 2\big(\sum_{i=1}^{n} x_i^T x_i + \lambda_1 I + \rho X^T X\big) H_j \\ & - 2\big(\sum_{i=1}^{n} x_i^T \widetilde{Y}_{ij} + \rho X^T (Z_j - \mu)\big). \end{aligned} \qquad (15)$$

By setting Eq. (15) to zero, H_j can be easily solved out in a closed form. On the other hand, if the dimension of H_j is great, cheap gradient descent optimization method can be applied. μ can be updated via

$$\mu \leftarrow \mu + XH - Z. \qquad (16)$$

Through alternatively updating H_j, Z_j and μ, the optimal H_j for problem (9) can be achieved.

4.2 Divide-and-Conquer Optimization

The divide-and-conquer strategy can be employed to increase the algorithm's capability for handling extremely large numbers of labels. The original optimization problem is first divided into cheaper sub-problems that can be efficiently solved in parallel. The solutions to these subproblems can then be combined to achieve the final solution. The whole optimization procedure is summarized in Algorithm 1.

Divide Step. Given the label matrix $Y \in \{-1, 1\}^{n \times L}$, we randomly partition it into t m-column sub-matrices $\{(Y)_i\}_{i=1}^{t}$, where we suppose $L = tm$ and each $(Y_i) \in \{-1, 1\}^{n \times m}$. Hence, the original problem is divided into t sub-problems regarding $\{(Y)_1, \cdots, (Y)_t\}$, respectively. The basic optimization method described in Section 4.1 can be adopted to solve these sub-problems in parallel, which outputs the solutions $\big\{\big((\widehat{W})_1, (\widehat{H})_1\big), \cdots, \big((\widehat{W})_t, (\widehat{H})_t\big)\big\}$.

Conquer Step. This conquer step exploits column projection to integrate the solutions of sub-problem solutions. The final approximation W to problem (5) can thus be obtained by projecting $[(\widehat{W})_1, \cdots, (\widehat{W})_t]$ onto the column space of $(\widehat{W})_1$. After obtaining $\widehat{W}$, we can then launch the optimization over each each column of $\widehat{H}$ in parallel to obtain the sparse component of the resulting multi-label predictor.

5. THEORETICAL ANALYSIS

In this section, we prove the upper bounds on the estimation error of the basic and divide-and-conquer optimization methods, respectively. The generalizability of our learning model is also analyzed.

5.1 Estimation Error

Given a feature matrix $X \in \mathbb{R}^{n\times d}$ and the corresponding label matrix $Y \in \{-1,1\}^{n\times L}$ for n training points of L labels, we suppose that

$$Y = XW_0 + Y_S + \epsilon, \tag{17}$$

where $W_0 \in \mathbb{R}^{d\times L}$ is the ground truth weight, Y_S is the sparse component to formulate the influence of tail labels, and ϵ is a data-independent noise term. Given this training data, we aim to estimate W_0 by performing empirical risk minimization:

$$\widehat{W} = \arg\min_W = \|\widetilde{Y} - XW\|_F^2 + \lambda\|W\|_*, \tag{18}$$

where $\widetilde{Y} = Y - Y_S$. Note that although the method in Eq. (4) uses a regularized rank-constrained formulation, we analyze the trace norm-regularized version without the rank constraint for simplicity. Since the class of rank-constrained matrices is smaller than the class of trace norm-constrained matrices, we can in fact expect better theoretical results here.

We generically denote the estimator for Σ_{XX} by X^TX, and the estimator for $\Sigma_{X\widetilde{Y}}$ by $X^T\widetilde{Y}$. We require Σ_{XX} to be positive semidefinite. Thus, the estimator for $\widehat{W}$ naturally becomes,

$$\widehat{W} = \arg\min_W \langle W, \Sigma_{XX}W\rangle - 2\langle\Sigma_{X\widetilde{Y}}, W\rangle + \lambda\|W\|_*. \tag{19}$$

We use the following theorem to show that $\widehat{W}$ solved in Eq. (19) is an approximated estimation of the true W_0.

THEOREM 1. *Suppose the smallest eigenvalue of Σ_{XX} is bounded by $\sigma_{min}(\Sigma_{XX}) \geq \sigma > 0$. The estimation error satisfies*

$$\|\widehat{W} - W_0\|_F \leq \frac{1}{\sigma}\big(2\|\Sigma_{X\widetilde{Y}} - \Sigma_{XX}W_0\|_* - \lambda\big). \tag{20}$$

PROOF. Let $\Delta = \widehat{W} - W_0$. Considering the optimality of $\widehat{W}$ for problem (19), we have

$$\begin{aligned}&\langle W_0+\Delta, \Sigma_{XX}(W_0+\Delta)\rangle - 2\langle\Sigma_{X\widetilde{Y}}, W_0+\Delta\rangle + \lambda\|W_0+\Delta\|_*\\ &\quad\leq\langle W_0, \Sigma_{XX}W_0\rangle - 2\langle\Sigma_{X\widetilde{Y}}, W_0\rangle + \lambda\|W_0\|_*,\end{aligned}$$

which can be rearranged to

$$\langle\Delta, \Sigma_{XX}\Delta\rangle \leq 2\langle\Sigma_{XY} - \Sigma_{XX}W_0, \Delta\rangle - \lambda\|\Delta\|_*. \tag{21}$$

Since the smallest eigenvalue of Σ_{XX} is bounded, we have

$$\langle\Delta, \Sigma_{XX}\Delta\rangle \geq \sigma\|\Delta\|_F^2. \tag{22}$$

Given $\|\Delta\|_* \geq \|\Delta\|_F$, the right hand side of Eq. (21) can be upper-bounded by

$$\begin{aligned}&2\langle\Sigma_{X\widetilde{Y}} - \Sigma_{XX}W_0, \Delta\rangle - \lambda\|\Delta\|_*\\ &\quad\leq 2\|\Sigma_{X\widetilde{Y}} - \Sigma_{XX}W_0\|_*\|\Delta\|_F - \lambda\|\Delta\|_F.\end{aligned} \tag{23}$$

Combining all the above results, we get

$$\sigma\|\Delta\|_F^2 \leq 2\|\Sigma_{X\widetilde{Y}} - \Sigma_{XX}W_0\|_*\|\Delta\|_F - \lambda\|\Delta\|_F. \tag{24}$$

The result then follows. □

According to Theorem 1, the estimation error bound depends on $\Sigma_{X\widetilde{Y}} = X^T(Y - Y_S)$, and thus Y_S can be interpreted as the perturbations of $\Sigma_{X\widetilde{Y}}$. Since the magnitude of Y_S is no greater than that of Y, the tail labels will not overwhelm the estimation over W.

We next analyze the estimation error in the divide-and-conquer optimization method. Suppose the compact singular value decomposition (SVD) of W is $U_W\Sigma V_W^T$, where Σ is diagonal and contains k non-zero singular values of W, and $U_W \in \mathbb{R}^{d\times k}$ and $V_W \in \mathbb{R}^{L\times k}$ are the corresponding left and right singular vectors of W. We assume the true weight W_0 is (μ,k)-coherent, whose definition is given as below,

DEFINITION 1. *Given $\mu_0(V_W) = \frac{d}{k}\max_{1\leq i\leq d}\|(V_W)_i\|^2$ and $\mu_1(W) = \sqrt{\frac{dL}{k}}\max_{i,j}|e_i^TU_WV_W^Te_j|$, for any $\mu > 0$, if $rank(W) = k$, $\max(\mu_0(V_W), \mu_0(V_W)) \leq \mu$ and $\mu_1(W) \leq \mu$, we call W is (μ,k)-coherent.*

We first invoke a lemma from [17] to show that column projection can produce an approximation that is nearly as good as a given rank-k target by sampling a number of columns proportional to the coherence.

LEMMA 1. *[17]. Given a matrix $W \in \mathbb{R}^{d\times L}$ and a rank-k approximation $A \in \mathbb{R}^{d\times L}$, choose $m \geq ck\mu\log(L)\log(1/\delta)/\epsilon^2$, where c is a fixed positive constant, and let $\widetilde{W} \in \mathbb{R}^{d\times m}$ be a matrix of m columns of W sampled uniformly without replacement. Then, with probability at least $1-\delta$,*

$$\|W - \widetilde{W}^{proj}\|_F \leq (1+\epsilon)\|W - A\|_F, \tag{25}$$

where $\widetilde{W}^{proj} \in \mathbb{R}^{d\times L}$ is derived by projecting $\widetilde{W}$ onto the space of W.

Recall that the true weight W_0 has been partitioned into t sub-matrices $\{(W_0)_1, \cdots, (W_0)_t\}$ that are solved by optimizing distinct sub-problems in parallel and correspond to t estimated sub-matrices $\{(\widehat{W})_1, \cdots, (\widehat{W})_t\}$. We employ column projection to derive the approximation of W_0 with the help of $\widehat{W}$, and denote it as $\widehat{W}^{proj}$. The difference between W_0 and $\widehat{W}^{proj}$ can be bounded by the following theorem.

THEOREM 2. *For $m \geq ck\mu\log(L)\log(1/\delta)/\epsilon^2$, where c is a fixed positive constant. The original problem has been divided into t sub-problems. If a basic optimization method yields the estimation error satisfying Theorem 1 for each sub-problem, then with probability at least $1-\delta$, the estimation error in the divide-and-conquer method is bounded by*

$$\|W_0 - \widehat{W}^{proj}\|_F \leq \frac{2+\epsilon}{\sigma}\sum_{i=1}^{t}\big(2\|\Sigma_{X(\widetilde{Y})_i} - \Sigma_{XX}(W_0)_i\|_* - \lambda\big)$$

PROOF. According to Lemma 1, with probability at least $1-\delta$, the following inequality holds:

$$\|\widehat{W} - \widehat{W}^{proj}\|_F \leq (1+\epsilon)\|\widehat{W} - W_0\|_F. \tag{26}$$

By adding $\|\widehat{W} - W_0\|_F$ to both sides of the above inequality, we get

$$\begin{aligned}(2+\epsilon)\|\widehat{W} - W_0\|_F &\geq \|\widehat{W} - \widehat{W}^{proj}\|_F + \|\widehat{W} - W_0\|_F\\ &\geq \|W_0 - \widehat{W}^{proj}\|_F,\end{aligned}$$

which implies that

$$\begin{aligned}\|W_0-\widehat{W}^{proj}\|_F &\le (2+\epsilon)\|\widehat{W}-W_0\|_F \\ &\le (2+\epsilon)\sum_{i=1}^{t}\|(W_0)_i-(\widehat{W})_i\|_F \\ &\le \frac{2+\epsilon}{\sigma}\sum_{i=1}^{t}\big(2\|\Sigma_{X(\widetilde{Y})_i}-\Sigma_{XX}(W_0)_i\|_*-\lambda\big).\end{aligned}$$

This ends the proof. □

Compared to the estimation error bound derived by applying a basic optimization method to estimate W_0 in Theorem 1, Theorem 2 exhibits an approximate recovery error with appropriate probability. It is instructive to note that the divide-and-conquer approach provides a controlled increase in error and a controlled decrease in the probability of success. Users can, therefore, adjust the optimization speed and accuracy.

5.2 Generalization Error

Given n multi-label points sampled i.i.d. from the distribution $\mathcal{Q}=\mathcal{X}\times\mathcal{Y}$, the proposed model aims to learn $(\widehat{W},\widehat{H})\in\mathcal{F}=\mathcal{W}\times\mathcal{H}$ by performing ERM as follows:

$$\inf_{\substack{rank(W)\le k\\ card(XH)\le s}}\widehat{\mathcal{L}}(W,H)=\frac{1}{n}\sum_{i=1}^{n}\ell(y_i,f(x_i,W,H)), \tag{27}$$

where $\widehat{\mathcal{L}}(W,H)$ is the empirical risk of the predictor (W,H). Our goal would be to show that $(\widehat{W},\widehat{H})$ has good generalization properties, that is:

$$\mathcal{L}(\widehat{W},\widehat{H})\le\inf_{\substack{rank(W)\le k\\ card(XH)\le s}}\widehat{\mathcal{L}}(W,H)+\epsilon, \tag{28}$$

where $\mathcal{L}(W,H)=\mathbb{E}_{(x,y)}[\ell(y,f(x,W,H))]$ is the population risk of the predictor.

The Rademacher complexity is an effective way to measure the richness (complexity) of the function class $\mathcal{F}$, based on which the generalization error bound of the learning algorithm can easily be obtained using standard approaches [3].

Definition 2. *Given a sample $S=\{x_1,\cdots,x_n\}\in\mathcal{X}^n$ and a real-valued function class $\mathcal{F}$ defined on a space $\mathcal{X}$, the empirical Rademacher complexity of $\mathcal{F}$ is defined as*

$$\widehat{R}_n(\mathcal{F})=\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\left|\frac{2}{n}\sum_{j=1}^{n}\sigma_j f(x_j)\right|x_1,\cdots,x_n\right],$$

where $\sigma=(\sigma_1,\cdots,\sigma_n)$ are independent uniform $\{\pm1\}$-valued Rademacher random variables. The Rademacher complexity of $\mathcal{F}$ is

$$R_n(\mathcal{F})=\mathbb{E}_s(\widehat{R}_n(\mathcal{F}))=\mathbb{E}_{x,\sigma}\left[\sup_{f\in\mathcal{F}}\left|\frac{2}{n}\sum_{j=1}^{n}\sigma_j f(x_j)\right|\right].$$

The Rademacher complexity of the proposed multi-label learning algorithm can thus be written as:

$$R_n(\mathcal{F})=\frac{2}{n}\mathbb{E}_{x,\sigma}\left[\sup_{(W,H)\in\mathcal{F}}\sum_{i=1}^{n}\sigma_i\sum_{j=1}^{L}x_i(w_j+h_j)\right],$$

where w_j and h_j correspond the j-th columns of W and H, respectively, and together determine the predictor for the j-th label. We denote $\widetilde{x}$ as the weighted summarization $\sum_{i=1}^{n}\sigma_i x_i$. L copies of $\widetilde{x}$ are then stacked into $\widetilde{X}$. Hence, the multi-label Rademacher complexity can be simply written as

$$R_n(\mathcal{F})=\frac{2}{n}\mathbb{E}_{x,\sigma}\left[\sup_{(W,H)\in\mathcal{F}}\langle W+H,\widetilde{X}\rangle\right], \tag{29}$$

whose upper bound can be revealed by the following theorem.

Theorem 3. *The proposed algorithm learns W and H over n training points with L labels i.i.d. sampled from distribution $\mathcal{Q}=\mathcal{X}\times\mathcal{Y}$. For any data point, $\|x\|_2\le\Lambda$. The learning algorithm encourages that $rank(W)\le k$ and $\|XH\|_1\le\Omega$. $\|W\|_F$ is assumed to be upper bounded by Θ. Then, the Rademacher complexity of F is*

$$R_n(\mathcal{F})\le 2\frac{\sqrt{nkL}\Theta\Lambda+L^{3/2}\Omega}{n}. \tag{30}$$

Proof. According to Eq. (29), we have

$$\begin{aligned}R_n(\mathcal{F}) &= \frac{2}{n}\mathbb{E}_{x,\sigma}\sup_{(W,H)\in\mathcal{F}}\Big[\langle W+H,\widetilde{X}\rangle\Big] \\ &= \frac{2}{n}\mathbb{E}_{x,\sigma}\sup_{(W,H)\in\mathcal{F}}\Big[\langle W,\widetilde{X}\rangle+\langle H,\widetilde{X}\rangle\Big] \\ &\le \frac{2}{n}\mathbb{E}_{x,\sigma}\sup_{(W,H)\in\mathcal{F}}\Big[\|W\|_*\|\widetilde{X}\|_F+\|\widetilde{X}H\|_*\Big] \\ &\le \frac{2}{n}\mathbb{E}_{x,\sigma}\sup_{(W,H)\in\mathcal{F}}\Big[\sqrt{k}\|W\|_F\|\widetilde{X}\|_F+\sqrt{L}\|\widetilde{X}H\|_1\Big].\end{aligned}$$

Then the following bounds can be easily obtained,

$$\mathbb{E}_{x,\sigma}\|\widetilde{X}\|_F^2=\mathbb{E}_{x,\sigma}L\|\widetilde{x}\|_2^2=\mathbb{E}_{x,\sigma}L[\sum_{i=1}^{n}\sigma_i x_i]_2^2\le nL\Lambda^2,$$

and

$$\mathbb{E}_{x,\sigma}\|\widetilde{X}H\|_1\le\mathbb{E}_x L\|XH\|_1\le L\Omega.$$

This proves

$$R_n(\mathcal{F})\le 2\frac{\sqrt{nkL}\Theta\Lambda+L^{3/2}\Omega}{n}. \tag{31}$$

□

To make conclusions of Theorem 3, consider a typical algorithm that neglects tail labels and attempts to minimize the trace norm of $W'=W+H$ for multi-label learning and with corresponding Rademacher complexity of $\mathcal{O}(\sqrt{rank(W')/n})$. As shown above, tail labels will significantly violate the low-rank assumption on W', and thus the large $rank(W')$ will lead to a greater Rademacher complexity. In contrast, the bound revealed in Theorem 3 is composed of two components corresponding to the low rank W and sparse XH, respectively. By separating the influence of tail labels, the rank of W (i.e., k) will be smaller. Although tail labels might influence multi-label learning, its significance on the bound can be dramatically decreased by constraining the sparsity of XH (i.e. $\|XH\|_1\le\Omega$). Hence, decomposing the multi-label model into low-rank and sparse parts to handle tail labels is helpful for decreasing the Rademacher complexity of the function class, which in turn improves the generalization error bound of the algorithm.

Table 1: Statistics of the datasets used in experiments.

Dataset	#training	#test	#features	#labels	#card-label	#card-feature
Bibtex	4,880	2,515	1,836	159	2.40	68.74
Delicious-S	12, 920	3,185	500	983	19.03	18.17
Mediamill	30, 993	12,914	120	101	4.38	120.00
Wiki10	14, 146	6,616	101,938	30,938	18.64	673.45
Delicious-L	196,606	100,095	782,585	205,443	75.54	301.17
Amazon	490,449	153,025	135,909	670,091	5.45	75.68

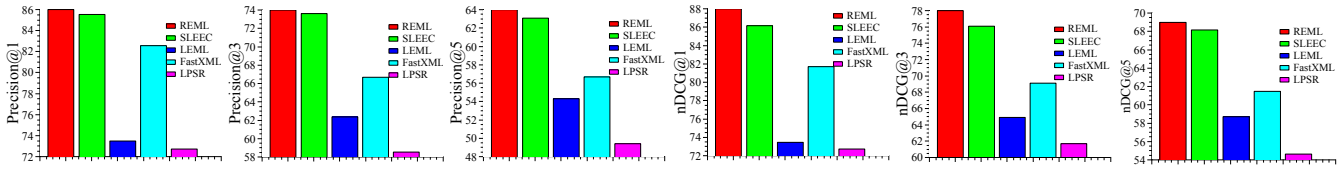

Figure 2: Top k precision and nDCG@k of multi-label learning algorithms on the Wiki10 dataset.

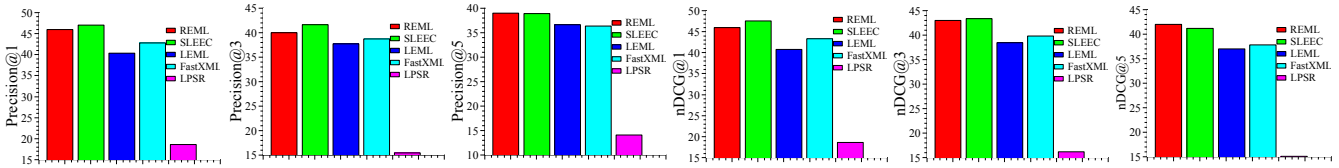

Figure 3: Top k precision and nDCG@k of multi-label learning algorithms on the Delicious-L dataset.

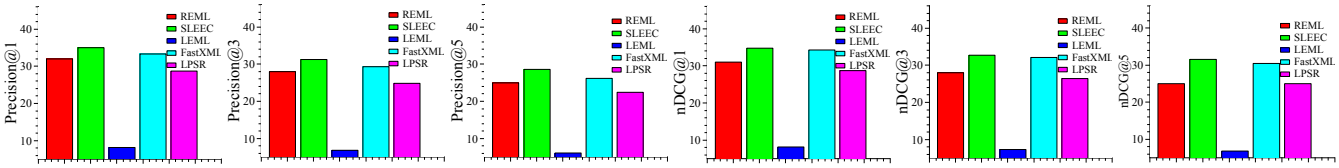

Figure 4: Top k precision and nDCG@k of multi-label learning algorithms on the Amazon dataset.

6. EXPERIMENTS

In this section, we evaluate the proposed algorithm on six benchmark multi-label datasets: *Bibtex*, *Delicious-S*, *Mediamill*, *Wiki10*, *Delicious-L* and *Amazon*. All these datasets were provided by [4], and have already been pre-separated into training and test sets. A summary of the statics of datasets is shown in Table 1. #training is the number of training examples; #test is the number of test examples; #features is the number of features; #labels is the number of labels; #card-label is the average number of positive labels per example; #card-feature is the average number of nonzero features per example. The first three datasets with less than 1,000 labels are regarded as small datasets, while the last three are large datasets. We set the embedding dimension in REML algorithm as $0.8L$ for the small datasets, and 200 for the large datasets. Since the *Delicious-L* and *Amazon* datasets are rather large, in optimizing REML we divided the original problems into 2 and 4 subproblems respectively using the divide-and-conquer strategy. For the other datasets, we directly solve REML using the basic optimization method.

In experiments, we compared our proposed Robust Extreme Multi-label Learning (REML) algorithm with LEML [26], which is the state-of-the-art label embedding method based on the low-rank assumption over label matrix, and SLEEC [4], which is a recently developed method using the neighborhood embedding technique to handle tail labels. Other representative multi-label learning algorithms, such as CS [14], CPLST [8], ML-CSSP [5] and 1-vs-All [1], which are only applicable for small datasets, are included in comparison experiments as well.

We used two metrics to evaluate the multi-label classification performance in the experiments, both of which have been widely used in the fields of multi-label learning and ranking. Precision at k measures the fraction of true positive predictions in the top k scoring labels, given the predicted score vector $\widehat{y} \in \mathbb{R}^L$. nDCG at k measures the usefulness, or gain, of a label based on its position in the predicted label list. We refer readers to [18] for more detailed information.

6.1 Results on Large Datasets

We compare the classification performance of the proposed REML algorithm with those of leading methods on three large datasets: *Wiki10*, *Delicious-L* and *Amazon*. The classification results measured in Precision@k and nDCG@k are presented in Figures 2-4. It can be seen that REML stably performs better than LEML on these large datasets. For example, REML improves over LEML by as much as

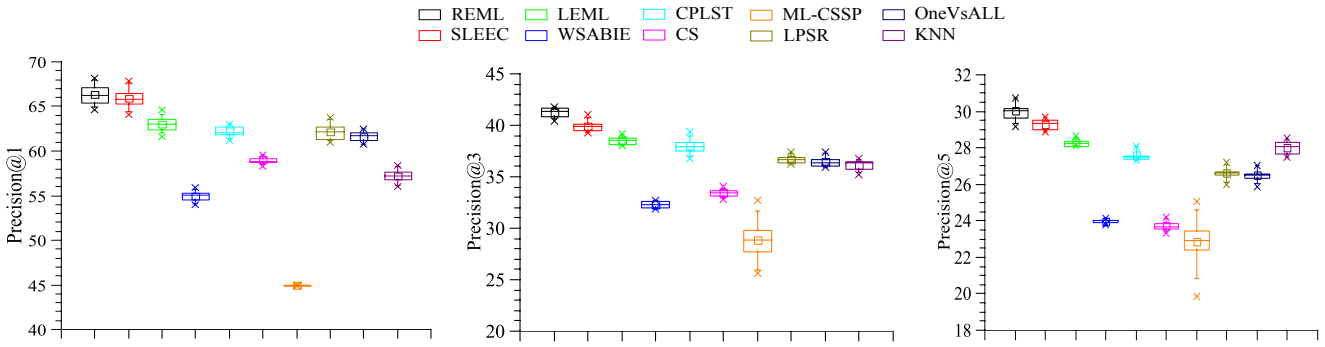

Figure 5: Top k precision of multi-label learning algorithms on the Bibtex dataset.

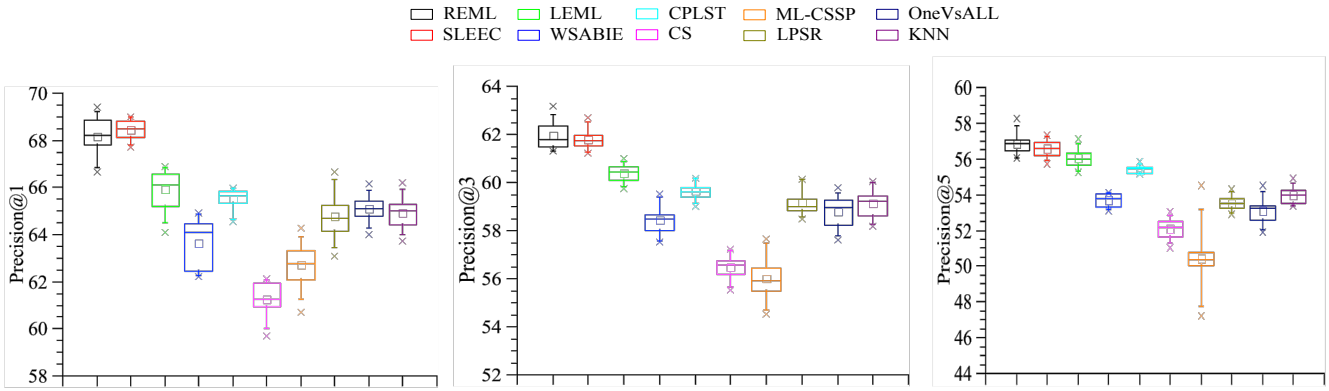

Figure 6: Top k precision of multi-label learning algorithms on the Delicious-S dataset.

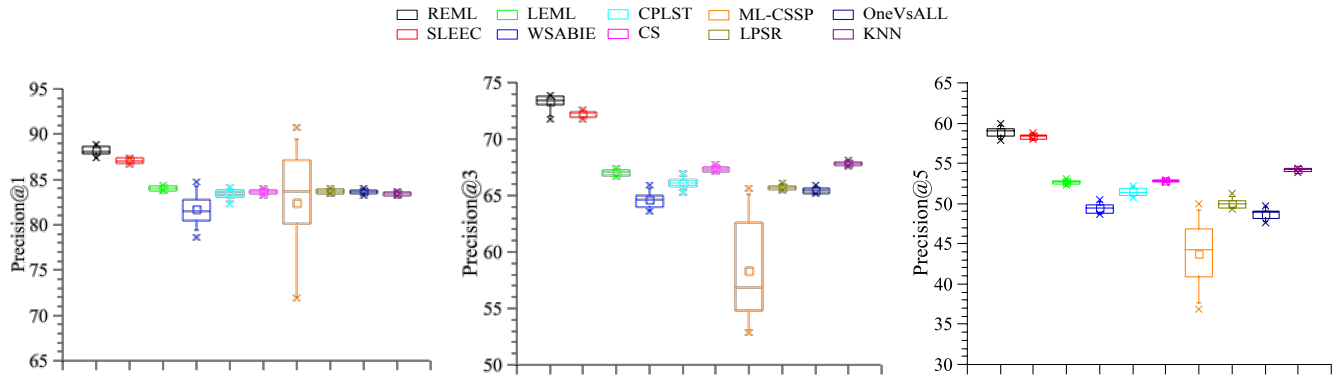

Figure 7: Top k precision of multi-label learning algorithms on the MediaMill dataset.

18% and 20% in terms of Precision@3 and nDCG@3 on the *Wiki10* dataset. This is because that the success of LEML mainly depends on the low-rank assumption, which tends to be violated on the large dataset with lots of tail labels (see Figure 1). REML provides an appropriate approach to preserve the validity of low-rank assumption by elegantly handling the tail labels as outliers. Hence, REML is able to achieve satisfactory classification results when faced with a number of tail labels, which is evidenced by its comparable performance with respect to the SLEEC method.

6.2 Results on Small Datasets

We next conduct multi-label classification over three small datasets: *Bibtex*, *Delicious-S*, *MediaMill*, which can be handled by more leading multi-label learning algorithms, such as CPLST and CS. The classification results in Precision@k are shown in Figures 5-7, while Table 2 presents the results in nDCG@k. Since the tail label problem is not acute on these small datasets, LEML and other comparison algorithms can achieve fine classification results. However, LEML is inferior to SLEEC, which uses neighborhood embedding to get rid of the influence of tail labels. Hence, SLEEC improves LEML

Table 2: nDCG@k of multi-label learning algorithms on the Bibtex, Delicious-S, Mediamill datasets.

	Bibtex			Delicious-S			Mediamill		
Algorithm	nDCG@1	nDCG@3	nDCG@5	nDCG@1	nDCG@3	nDCG@5	nDCG@1	nDCG@3	nDCG@5
KNN	57.81	52.36	54.57	64.80	60.71	57.02	82.59	75.62	72.76
OneVsAll	62.62	59.44	61.73	64.90	60.94	56.54	83.67	73.90	67.75
LPSR	62.98	57.11	58.76	64.46	60.47	56.19	83.65	74.12	69.12
ML-CSSP	56.86	52.54	54.81	63.96	59.07	54.86	83.21	72.67	65.05
CS	59.60	53.08	53.74	61.26	57.85	54.44	83.97	75.29	71.99
CPLST	61.99	57.66	59.71	65.65	61.52	58.00	83.79	74.44	70.49
WSABIE	55.03	50.26	52.33	63.67	59.47	56.37	79.86	72.51	69.31
LEML	63.10	58.84	61.06	64.96	61.80	58.42	83.09	75.23	71.96
SLEEC	64.49	59.90	62.29	67.41	62.46	59.02	86.61	80.04	77.71
REML	65.13	60.01	62.46	66.30	62.65	59.10	86.73	82.67	78.32

Table 3: Time and performance of REML with varying numbers of subproblems.

#subproblems	Time (s)	Precision@1	Precision@3	Precision@5	nDCG@1	nDCG@3	nDCG@5
1	3,739	86.17	74.30	64.37	88.09	78.44	69.38
2	2,190	84.07	72.70	62.92	85.26	76.52	68.94
5	1,495	82.80	71.20	62.78	83.79	75.47	66.23
8	923	78.45	69.53	58.76	81.92	72.07	64.42
10	764	76.68	67.75	56.68	80.01	70.30	63.32

by 4.8% in terms of Precision@1 on the *Bibtex* dataset and 7.8% in terms of Precision@3 on the *MediaMill* dataset. We suggest that the performance gap between LEML and SLEEC can be bridged by the REML algorithm, which is able to suppress the influence of tail labels and activate the low-rank assumption on multiple labels. On the *Delicious-S* dataset, REML is the closest competitor of SLEEC, while on the *Bibtex* dataset, REML takes a lead ahead of SLEEC in terms of Precision@5. This demonstrates that the low-rank assumption is powerful for processing multiple labels, and its performance can be further strengthened by carefully investigating the tail labels as outliers in the low-rank formulation.

6.3 Algorithm Analysis

We next perform experiments on the *Wiki10* dataset, to explore the trade-off between computation and accuracy when using divide-and-conquer technique to optimize the proposed REML algorithm (denoted as 'DC-REML'). Table 3 presents the time required to solve REML with different numbers of subproblems (i.e., t), and the corresponding classification results in Precision@k and nDCG@k. From this table, we find that DC-REML performs comparably to REML for smaller values of t, and the performance gradually degrades for larger subproblem number, which is consistent with the theoretical analysis in Section 5.1. Most importantly, DC-REML have significantly sped up REML by dividing the original problem into 5 subproblems, with an acceptable performance degradation of 4% relative to REML in terms of Precision@3. Hence, given the scalability provided by the divide-and-conquer optimization technique, we can flexibly manage the optimization accuracy and time cost in solving large-scale multi-label learning problems.

7. CONCLUSION

In this paper, we study the tail labels problem in multi-label learning, where the scarce labels associate with limited number of examples. To prevent the damage of tail labels on the low-rank assumption over multiple labels, we treat tail labels as outliers and develop a robust extreme multi-label learning algorithm. Divide-and-conquer approach applied to optimize the resulting objective function is beneficial for improving the scalability, and its advantages in balancing accuracy and computation have been theoretically demonstrated. We analyze the generalization error of the proposed algorithm, and suggest that it can be improved by the low-rank and sparse decomposition given tail labels. Experimental results on real-world datasets demonstrate the significance of investigating tail labels and the promising performance of the proposed algorithm.

References

[1] S. V. N. V. B. Hariharan and M. Varma. Efficient max-margin multi-label classification with applications to zero-shot learning. *Machine Learning*, 2012.

[2] K. Balasubramanian and G. Lebanon. The landmark selection method for multiple output prediction. In *ICML*, 2012.

[3] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.

[4] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pages 730–738, 2015.

[5] W. Bi and J. Kwok. Efficient multi-label classification with many labels. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 405–413, 2013.

[6] W. Bi and J. T. Kwok. Multi-label classification on tree-and dag-structured hierarchies. In *Proceedings of*

the 28th International Conference on Machine Learning (ICML-11), pages 17–24, 2011.

[7] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3):394–410, 2007.

[8] Y.-N. Chen and H.-T. Lin. Feature-aware label space dimension reduction for multi-label classification. In *Advances in Neural Information Processing Systems*, pages 1538–1546, 2012.

[9] M. M. Cisse, N. Usunier, T. Artieres, and P. Gallinari. Robust bloom filters for large multilabel classification tasks. In *Advances in Neural Information Processing Systems*, pages 1851–1859, 2013.

[10] K. Crammer and Y. Singer. A family of additive online algorithms for category ranking. *The Journal of Machine Learning Research*, 3:1025–1058, 2003.

[11] D. L. Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627, 1995.

[12] A. Elisseeff and J. Weston. A kernel method for multilabelled classification. In *Advances in neural information processing systems*, pages 681–687, 2001.

[13] B. Hariharan, L. Zelnik-Manor, M. Varma, and S. Vishwanathan. Large scale max-margin multi-label classification with priors. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 423–430, 2010.

[14] D. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, volume 22, pages 772–780, 2009.

[15] S.-J. Huang, Y. Yu, and Z.-H. Zhou. Multi-label hypothesis reuse. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 525–533. ACM, 2012.

[16] C.-L. Li and H.-T. Lin. Condensed filter tree for cost-sensitive multi-label classification. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 423–431, 2014.

[17] L. W. Mackey, M. I. Jordan, and A. Talwalkar. Divide-and-conquer matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1134–1142, 2011.

[18] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[19] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168, 2000.

[20] E. Spyromitros, G. Tsoumakas, and I. Vlahavas. An empirical study of lazy multilabel classification algorithms. In *Artificial Intelligence: Theories, Models and Applications*, pages 401–406. Springer, 2008.

[21] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 668–676. ACM, 2008.

[22] F. Tai and H.-T. Lin. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.

[23] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2010.

[24] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214, 2008.

[25] R. Yan, J. Tesic, and J. R. Smith. Model-shared subspace boosting for multi-label classification. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 834–843. ACM, 2007.

[26] H.-F. Yu, P. Jain, and I. S. Dhillon. Large-scale multi-label learning with missing labels. In *Proceedings of the twenty-first international conference on Machine learning*, 2014.

[27] M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 999–1008. ACM, 2010.

[28] M.-L. Zhang and Z.-H. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *Knowledge and Data Engineering, IEEE Transactions on*, 18(10):1338–1351, 2006.

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

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

[31] Y. Zhang and J. G. Schneider. Multi-label output codes using canonical correlation analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 873–882, 2011.