Large-scale Multi-label Learning with Missing Labels

Hsiang-Fu Yu ROFUYU@CS.UTEXAS.EDU
Department of Computer Science, University of Texas at Austin

Prateek Jain PRAJAIN@MICROSOFT.COM
Purushottam Kar T-PURKAR@MICROSOFT.COM
Microsoft Research India, Bangalore

Inderjit S. Dhillon INDERJIT@CS.UTEXAS.EDU
Department of Computer Science, University of Texas at Austin

Abstract

The multi-label classification problem has generated significant interest in recent years. However, existing approaches do not adequately address two key challenges: (a) scaling up to problems with a large number (say millions) of labels, and (b) handling data with missing labels. In this paper, we directly address both these problems by studying the multi-label problem in a generic empirical risk minimization (ERM) framework. Our framework, despite being simple, is surprisingly able to encompass several recent label-compression based methods which can be derived as special cases of our method. To optimize the ERM problem, we develop techniques that exploit the structure of specific loss functions - such as the squared loss function - to obtain efficient algorithms. We further show that our learning framework admits excess risk bounds even in the presence of missing labels. Our bounds are tight and demonstrate better generalization performance for low-rank promoting trace-norm regularization when compared to (rank insensitive) Frobenius norm regularization. Finally, we present extensive empirical results on a variety of benchmark datasets and show that our methods perform significantly better than existing label compression based methods and can scale up to very large datasets such as a Wikipedia dataset that has more than 200,000 labels.

1. Introduction

Large scale multi-label classification is an important learning problem with several applications to real-world problems such as image/video annotation (Carneiro et al., 2007; Wang et al., 2009) and query/keyword suggestions (Agrawal et al., 2013). The goal in multi-label classification is to predict a label vector $\boldsymbol{y} \in \{0,1\}^L$ for a given data point $\boldsymbol{x} \in \mathbb{R}^d$. This problem has been studied extensively in the domain of structured output learning, where the number of labels is assumed to be small and the main focus is thus, on modeling inter-label correlations and using them to predict the label vector (Hariharan et al., 2010).

Due to several motivating real-life applications, recent research on multi-label classification has largely shifted its focus to the other end of the spectrum where the number of labels is assumed to be extremely large, with the key challenge being the design of scalable algorithms that offer real-time predictions and have a small memory footprint. In such situations, simple methods such as 1-vs-all or Binary Relevance (BR), that treat each label as a separate binary classification problem fail miserably. For a problem with (say) 10^4 labels and 10^6 features, which is common in several applications, these methods have a memory footprint of around 100 Gigabytes and offer slow predictions.

A common technique that has been used to handle the label proliferation problem in several recent works is "label space reduction". The key idea in this technique is to reduce the dimensionality of the label-space by using either random projections or canonical correlation analysis (CCA) based projections (Chen & Lin, 2012; Hsu et al., 2009; Tai & Lin, 2012; Kapoor et al., 2012). Subsequently, these methods perform prediction on the smaller dimensional label-space and then recover the original labels by projecting back onto the high dimensional label-space. In particular, (Chen & Lin, 2012) recently proposed an efficient algorithm with both label-space and feature-space compression via a CCA type method with some orthogonality constraints. However, this method is relatively rigid and cannot handle several important issues inherent to multi-label problems; see Section 2.1 for more details.

In this paper we take a more direct approach by formulating the problem as that of learning a low-rank linear model $Z \in \mathbb{R}^{d \times L}$ s.t. $\boldsymbol{y}^{pred} = Z^T \boldsymbol{x}$. We cast this learning prob-

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

lem in the standard ERM framework that allows us to use a variety of loss functions and regularizations for Z. This framework unifies several existing dimension reduction approaches. In particular, we show that if the loss function is chosen to be the squared-L_2 loss, then our proposed formulation has a closed form solution, and surprisingly, the conditional principal label space transformation (CPLST) method of (Chen & Lin, 2012) can be derived as a *special case*. However, the flexibility of the framework allows us to use other loss functions and regularizers that are useful for preventing overfitting and increasing scalability.

Moreover, we can extend our formulation to handle missing labels; in contrast, most dimension reduction formulations (including CPLST) cannot accommodate missing labels. The ability to learn in the presence of missing labels is crucial as for most real-world applications, one cannot expect to accurately obtain (either through manual or automated labeling) all the labels for a given data point. For example, in image annotation, human labelers tag only prominent labels and typically miss out on several objects present in the image. Similarly, in online collections such as Wikipedia, where articles get tagged with categories, human labelers usually tag only with categories they know about. Moreover, there might be considerable noise in the labeling.

In order to solve for the low-rank linear model that results from our formulation, we use the popular alternating minimization algorithm that works well despite the nonconvexity of the rank constraint. For general loss functions and trace-norm regularization, we exploit subtle structures present in the problem to design a fast conjugate gradient based method. For the special case of squared-L_2 loss and trace-norm regularization, we further exploit the structure of the loss function to provide a more efficient and scalable algorithm. As compared to direct computation, our algorithm is $O(\bar{d})$ faster, where $\bar{d}$ is the average number of nonzero features in an instance.

On the theoretical side, we perform an excess risk analysis for the trace-norm regularized ERM formulation with missing labels, assuming labels are observed uniformly at random. Our proofs do not follow from existing results due to missing labels and require a careful analysis involving results from random matrix theory. Our results show that while in general the low-rank promoting trace-norm regularization does not provide better bounds than learning a full-rank matrix (e.g. using Frobenius norm regularization), for several interesting data distributions, trace-norm regularization does indeed give significantly better bounds. More specifically, for isotropic data distributions, we show that trace-norm based methods have excess risk of $O(\frac{1}{\sqrt{nL}})$ while full-rank learning can only guarantee $O(\frac{1}{\sqrt{n}})$ excess risk, where n is the number of training points.

Finally, we provide an extensive empirical evaluation of our method on a variety of benchmark datasets. In particular, we compare our method against three recent label compression based methods: CPLST (Chen & Lin, 2012), Bayesian-CS (Kapoor et al., 2012), and WSABIE (Weston et al., 2010). On almost all datasets, our method significantly outperforms these methods, both in the presence and absence of missing labels. Finally, we show the scalability of our method by applying it to a recently curated Wikipedia dataset (Agrawal et al., 2013), that has 881,805 training samples and 213,707 labels. The results show that our method not only provides reasonably accurate solutions for such large-scale problems, but that the training time is orders of magnitude lesser than several existing methods.

Related Work. Typically, Binary Relevance (BR), which treats each label as an independent binary classification task, is quite accurate for multi-label learning. However, for a large number of labels, this method becomes infeasible due to increased model size and prediction time. Recently, techniques have been developed that either reduce the dimension of the labels, such as the Compressed Sensing Approach (Hsu et al., 2009), PLST (Tai & Lin, 2012), CPLST (Chen & Lin, 2012), and Bayesian CS (Kapoor et al., 2012), or reduce the feature dimension, such as (Sun et al., 2011), or both, such as WSABIE (Weston et al., 2010). Most of these techniques are tied to a specific loss function (e.g., CPLST and BCS cater only to the squared-L_2 loss, and WSABIE works with the weighted approximate ranking loss) and/or cannot handle missing labels.

Our framework models multi-label classification as a general ERM problem with a low-rank constraint, which not only generalizes both label and feature dimensionality reduction but also brings in the ability to support various loss functions and allows for rigorous generalization error analysis. We show that our formulation not only retrieves CPLST, which has been shown to be fairly accurate, as a special case, but substantially enhances it by use of regularization, other loss functions, allowing missing labels etc.

Paper Organization. We begin by studying a generic low-rank ERM framework for multi-label learning in Section 2. Next, we propose efficient algorithms for the framework in Section 3 and analyze their generalization performance for trace-norm regularization in Section 4. We present empirical results in Section 5, and conclude in Section 6.

2. Problem Formulation

In this section we present a generic ERM-style framework for multi-label classification. For each training point, we shall receive a feature vector $\boldsymbol{x}_i \in \mathbb{R}^d$ and a corresponding label vector $\boldsymbol{y}_i \in \{0,1\}^L$ with L labels. For any $j \in [L]$, $\boldsymbol{y}_i^j = 1$ will denote that the l^{th} label is "present" or "on" whereas $\boldsymbol{y}_i^j = 0$ will denote that the label is "absent" or "off". Note that although we focus mostly on the binary classification setting in this paper, our methods easily extend to the multi-class setting where $\boldsymbol{y}_i^j \in \{1, 2, \ldots, C\}$.

Our predictions for the label vector shall be parametrized as $f(\boldsymbol{x}; Z) = Z^T \boldsymbol{x}$, where $Z \in \mathbb{R}^{d \times L}$. Although we have adopted a linear parametrization here, our results can easily be extended for non-linear kernels as well. Let $\ell(\boldsymbol{y}, f(\boldsymbol{x}; Z)) \in \mathbb{R}$ be the loss function that computes the discrepancy between the "true" label vector and the prediction. We assume that the loss function is decomposable, i.e., $\ell(\boldsymbol{y}, f(\boldsymbol{x}; Z)) = \sum_{j=1}^{L} \ell(\boldsymbol{y}^j, f^j(\boldsymbol{x}; Z))$.

The motivation for our framework comes from the observation that although the number of labels in a multi-label classification problem might be large, there typically exist significant label correlations, thus reducing the effective number of parameters required to model them to much less than $d \times L$. We capture this intuition by restricting the matrix Z to learn only a small number of "latent" factors. This constrains Z to be a low rank matrix which not only controls overfitting but also gives computational benefits.

Given n training points our training set will be (X, Y) where $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^T$ and $Y = [\boldsymbol{y}_1\ \boldsymbol{y}_2\ \ldots\ \boldsymbol{y}_n]^T$. Using the loss function ℓ, we propose to learn the parameters Z by using the canonical ERM method, i.e.,

$$\hat{Z} = \arg\min_{Z} J(Z) = \sum_{i=1}^{n} \sum_{j=1}^{L} \ell(Y_{ij}, f^j(\boldsymbol{x}_i; Z)) + \lambda \cdot r(Z), \quad s.t.\ \operatorname{rank}(Z) \leq k, \tag{1}$$

where $r(Z) : \mathbb{R}^{d \times L} \to \mathbb{R}$ is a regularizer. If there are missing labels, we compute the loss over the known labels:

$$\hat{Z} = \arg\min_{Z} J_{\Omega}(Z) = \sum_{(i,j) \in \Omega} \ell(Y_{ij}, f^j(\boldsymbol{x}_i; Z)) + \lambda \cdot r(Z), \quad s.t.\ \operatorname{rank}(Z) \leq k, \tag{2}$$

where $\Omega \subseteq [n] \times [L]$ is the index set that represents "known" labels. Note that in this work, we assume the standard missing value setting, where each label can be either on, off (i.e., $Y_{ij} = 1$ or 0), or missing ($Y_{ij} =?$); several other works have considered another setting where only positive labels are known and are given as 1 in the label matrix, while negative or missing values are all denoted by 0 (Agrawal et al., 2013; Bucak et al., 2011).

Note that although the above formulation is NP-hard in general due to the non-convex rank constraint, for convex loss functions, one can still utilize the standard alternating minimization method. Moreover, for the special case of L_2 loss, we can derive closed form solutions for the full-label case (1) and show connections to several existing methods.

We would like to note that while the ERM framework is well known and standard, most existing multi-label methods for large number of labels motivate their work in a relatively ad-hoc manner. Using our approach, we can show that existing methods like CPLST (Chen & Lin, 2012) are in fact a special case of our generic ERM framework (see next section). Furthermore, having this framework also helps us in studying generalization error bounds for our methods and identifying situations where the methods can be expected to succeed (see Section 4).

2.1. Special Case: Squared-L_2 loss

In this section, we study (1) and (2) for the special case of squared L_2 loss function, i.e., $\ell(\boldsymbol{y}, f(\boldsymbol{x}; Z)) = \|\boldsymbol{y} - f(\boldsymbol{x}; Z)\|_2^2$. We show that in the absence of missing labels, the formulation in (1) can be solved optimally for the squared L_2 loss using SVD. Furthermore, by selecting an appropriate regularizer $r(Z)$ and λ, our solution for L_2 loss is exactly the same as that of CPLST (Chen & Lin, 2012).

We first show that the unregularized form of (1) with $\ell(\boldsymbol{y}, f(\boldsymbol{x}; Z)) = \|\boldsymbol{y} - Z^T \boldsymbol{x}\|_2^2$ has a closed form solution.

Claim 1. *If $\ell(\boldsymbol{y}, f(\boldsymbol{x}; Z)) = \|\boldsymbol{y} - Z^T \boldsymbol{x}\|_2^2$ and $\lambda = 0$, then*

$$V_X \Sigma_X^{-1} M_k = \arg \min_{Z : \operatorname{rank}(Z) \leq k} \|Y - XZ\|_F^2, \tag{3}$$

where $X = U_X \Sigma_X V_X^T$ is the thin SVD decomposition of X, and M_k is the rank-k truncated SVD of $M \equiv U_X^T Y$.

See Appendix A for a proof of Claim 1. We now show that this is exactly the solution obtained by (Chen & Lin, 2012) for their CPLST formulation.

Claim 2. *The solution to (3) is equivalent to $Z^{CPLST} = W_{CPLST} H_{CPLST}^T$ which is the closed form solution for the CPLST scheme.*

See Appendix A for a proof. Note that (Chen & Lin, 2012) derive their method by relaxing a Hamming loss problem and dropping constraints in the canonical correlation analysis in a relatively ad-hoc manner. The above results, on the other hand, show that the same model can be derived in a more principled manner. This helps us in extending the method for several other problem settings in a principled manner and also helps in providing excess risk bounds:

- As shown empirically, CPLST tends to overfit significantly whenever d is large. However, we can handle this issue by setting λ appropriately.
- The closed form solution in (Chen & Lin, 2012) cannot directly handle missing labels as it requires SVD on fully observed Y. In contrast, our framework can itself handle missing labels without any modifications.
- The formulation in (Chen & Lin, 2012) is tied to the L_2 loss function. In contrast, we can easily handle other loss functions; although, the optimization problem might become more difficult to solve.

We note that such links between low rank solutions to multi-variate regression problems and PCA/SVD are well known in literature (Izenman, 1975; Breiman & Friedman, 1997). However, these results are mostly derived in the stochastic setting under various noise models whereas ours apply to the empirical setting. Moreover, these classical results put little emphasis on large scale implementation.

3. Algorithms

In this section, we apply the alternating minimization technique for optimizing (1) and (2). For a matrix Z with a known low rank k, it is inefficient to represent it using $d \times L$ entries, especially when d and L are large. Hence we consider a low-rank decomposition of the form $Z = WH^T$, where $W \in \mathbb{R}^{d\times k}$ and $H \in \mathbb{R}^{L\times k}$. We further assume that $r(Z)$ can be decomposed into $r_1(W) + r_2(H)$. In the following sections, we present results with the trace norm regularization, i.e., $r(Z) = \|Z\|_{\text{tr}}$, which can be decomposed as $\|Z\|_{\text{tr}} = \frac{1}{2}\left(\|W\|_F^2 + \|H\|_F^2\right)$. Thus, $\min_Z J_\Omega(Z)$ with the rank constraint is equivalent to minimizing over W, H:

$$J_\Omega(W,H) = \sum_{(i,j)\in\Omega} \ell(Y_{ij}, \boldsymbol{x}_i^T W \boldsymbol{h}_j) + \frac{\lambda}{2}\left(\|W\|_F^2 + \|H\|_F^2\right) \tag{4}$$

where $\boldsymbol{h}_j^T$ is the j-th row of H. Note that when either of W or H is fixed, $J_\Omega(W,H)$ becomes a convex function. This allows us to apply alternating minimization, a standard technique for optimizing functions with such a property, to (4). For a general loss function, after proper initialization, a sequence $\{(W^{(t)}, H^{(t)})\}$ is generated by

$$H^{(t)} \leftarrow \arg\min_H \quad J_\Omega(W^{(t-1)}, H),$$
$$W^{(t)} \leftarrow \arg\min_W \quad J_\Omega(W, H^{(t)}).$$

For a convex loss function, $(W^{(t)}, H^{(t)})$ is guaranteed to converge to a stationary point when the minimum for both $\min_H J_\Omega\left(W^{(t-1)}, H\right)$ and $\min_W J_\Omega\left(W, H^{(t)}\right)$ are uniquely defined (see Bertsekas, 1999, Proposition 2.7.1). In fact, when the squared loss is used and Y is fully observed, the case considered in Section 3.2, we can prove that $(W^{(t)}, H^{(t)})$ converges to the global minimum of (4) when either $\lambda = 0$ or X is orthogonal.

Once W is fixed, updating H is easy as each row $\boldsymbol{h}_j$ of H can be independently updated as follows:

$$\boldsymbol{h}_j \leftarrow \arg\min_{\boldsymbol{h}\in\mathbb{R}^k} \sum_{i:(i,j)\in\Omega} \ell(Y_{ij}, \boldsymbol{x}_i^T W \boldsymbol{h}) + \frac{1}{2}\lambda \cdot \|\boldsymbol{h}\|_2^2, \tag{5}$$

which is easy to solve as k is small in general. Based on the choice of the loss function, (5) is essentially a linear classification or regression problem over k variables with $|\{i : (i,j) \in \Omega\}|$ instances. If H is fixed, updating W is more involved as all variables are mixed up due to the pre-multiplication with X. Let $\tilde{\boldsymbol{x}}_{ij} = \boldsymbol{h}_j \otimes \boldsymbol{x}_i$ (where $\otimes$ denotes the outer product). It can be shown that updating W is equivalent to a regularized linear classification/regression problem with $|\Omega|$ data points $\{(Y_{ij}, \tilde{\boldsymbol{x}}_{ij}) : (i,j) \in \Omega\}$. Thus if $W^* = \arg\min_W \; J_\Omega(W,H)$ and we denote $\boldsymbol{w}^* := \boldsymbol{vec}\,(W^*)$, then $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^{dk}} g(\boldsymbol{w})$,

$$g(\boldsymbol{w}) \equiv \sum_{(i,j)\in\Omega} \ell\left(Y_{ij}, \boldsymbol{w}^T \tilde{\boldsymbol{x}}_{ij}\right) + \frac{1}{2}\lambda \cdot \|\boldsymbol{w}\|_2^2. \tag{6}$$

Taking the squared loss as an example, the above is equivalent to a regularized least squares problem with dk variables. When d is large, say 1M, the closed form solution, which requires inverting a $dk \times dk$ matrix, can hardly be regarded as feasible. As a result, updating W efficiently turns out to be the main challenge for alternating minimization.

In large-scale settings where both dk and $|\Omega|$ are large, iterative methods such as Conjugate Gradient (CG), which perform cheap updates and offer a good approximate solution within a few iterations, are more appropriate to solve (6). Several linear classification/regression packages such as LIBLINEAR (Fan et al., 2008) can handle such problems if $\{\tilde{\boldsymbol{x}}_{ij} : (i,j) \in \Omega\}$ are available. The main operation in such iterative methods is a gradient calculation ($\nabla g(\boldsymbol{w})$) or a multiplication of the Hessian matrix and a vector ($\nabla^2 g(\boldsymbol{w})\boldsymbol{s}$). Let $\tilde{X} = [\cdots \tilde{\boldsymbol{x}}_{ij} \cdots]^T_{(i,j)\in\Omega}$ and $\bar{d} = \sum_{i=1}^n \|\boldsymbol{x}\|_0/n$. Then these operations require at least $nnz(\tilde{X}) = O(|\Omega|\bar{d}k)$ time to compute in general.

However, as we show below, we can exploit the structure in $\tilde{X}$ to develop efficient techniques such that both the operations mentioned above can be done in $O\left((|\Omega| + nnz(X) + d + L) \times k\right)$ time. As a result, iterative methods, such as CG, can achieve $O(\bar{d})$ speedup. See Appendix B for a detailed CG procedure for (6) with the squared loss. Our techniques make the alternating minimization efficient enough to handle large-scale problems.

3.1. Fast Operations for General Loss Functions

We assume that the loss function is a general twice-differentiable function $\ell(a,b)$, where a and b are scalars. Let $\ell'(a,b) = \frac{\partial}{\partial b}\ell(a,b)$, and $\ell''(a,b) = \frac{\partial^2}{\partial b^2}\ell(a,b)$. The gradient and the Hessian matrix for $g(\boldsymbol{w})$ are:

$$\nabla g(\boldsymbol{w}) = \sum_{(i,j)\in\Omega} \ell'(Y_{ij}, \boldsymbol{w}^T\tilde{\boldsymbol{x}}_{ij})\tilde{\boldsymbol{x}}_{ij} + \lambda\boldsymbol{w}, \tag{7}$$

$$\nabla^2 g(\boldsymbol{w}) = \sum_{(i,j)\in\Omega} \ell''(Y_{ij}, \boldsymbol{w}^T\tilde{\boldsymbol{x}}_{ij})\tilde{\boldsymbol{x}}_{ij}\tilde{\boldsymbol{x}}_{ij}^T + \lambda I. \tag{8}$$

A direct computation of $\nabla g(\boldsymbol{w})$ and $\nabla^2 g(\boldsymbol{w})\boldsymbol{s}$ using (7) and (8) requires at least $O(|\Omega|\bar{d}k)$ time. Below we give faster procedures to perform both operations.

Gradient Calculation. Recall that $\tilde{\boldsymbol{x}}_{ij} = \boldsymbol{h}_j \otimes \boldsymbol{x}_i = \boldsymbol{vec}\left(\boldsymbol{x}_i\boldsymbol{h}_j^T\right)$. Therefore, we have $\sum_{(i,j)\in\Omega} \ell'(Y_{ij}, \boldsymbol{w}^T\tilde{\boldsymbol{x}}_{ij})\boldsymbol{x}_i\boldsymbol{h}_j^T = X^TDH$, where D is sparse with $D_{ij} = \ell'(Y_{ij}, \boldsymbol{w}^T\tilde{\boldsymbol{x}}_{ij})$, $\forall(i,j) \in \Omega$. Thus,

$$\nabla g(\boldsymbol{w}) = \boldsymbol{vec}\left(X^TDH\right) + \lambda\boldsymbol{w}. \tag{9}$$

Assuming that $\ell'(a,b)$ can be computed in constant time, which holds for most loss functions (e.g. squared-L_2 loss, logistic loss), the gradient computation can be done in $O\left((nnz(X) + |\Omega| + d) \times k\right)$ time. Algorithm 1 gives the details of computing $\nabla g(\boldsymbol{w})$ using (9).

Hessian-vector Multiplication. After substituting $\tilde{\boldsymbol{x}}_{ij} = \boldsymbol{h}_j \otimes \boldsymbol{x}_i$, we have

$$\nabla^2 g(\boldsymbol{w})\boldsymbol{s} = \sum_{(i,j)\in\Omega} \ell''_{ij} \cdot \left((\boldsymbol{h}_j\boldsymbol{h}_j^T) \otimes (\boldsymbol{x}_i\boldsymbol{x}_i^T)\right)\boldsymbol{s} + \lambda\boldsymbol{s},$$

Algorithm 1 General Loss with Missing Labels

```
To compute $\nabla g(\boldsymbol{w})$:
    $A \leftarrow XW$, where $\boldsymbol{vec}\,(W) = \boldsymbol{w}$.
    $D_{ij} \leftarrow \ell'(Y_{ij}, \boldsymbol{a}_i^T \boldsymbol{h}_j)$, $\forall (i,j) \in \Omega$.
    Return: $\boldsymbol{vec}\left(X^T(DH)\right) + \lambda \boldsymbol{w}$
To compute: $\nabla^2 g(\boldsymbol{w})\boldsymbol{s}$
    $A \leftarrow XW$, where $\boldsymbol{vec}\,(W) = \boldsymbol{w}$.
    $B \leftarrow XS$, where $\boldsymbol{vec}\,(S) = \boldsymbol{s}$.
    $U_{ij} \leftarrow \ell''(Y_{ij}, \boldsymbol{a}_i^T \boldsymbol{h}_j)\boldsymbol{b}_i^T \boldsymbol{h}_j$, $\forall (i,j) \in \Omega$.
    Return: $\boldsymbol{vec}\left(X^T(UH)\right) + \lambda \boldsymbol{s}$.
```

Algorithm 2 Squared Loss with Full Labels

```
To compute $\nabla g(\boldsymbol{w})$:
    $A \leftarrow XW$, where $\boldsymbol{vec}\,(W) = \boldsymbol{w}$.
    $B \leftarrow YH$.
    $M \leftarrow H^T H$.
    Return: $\boldsymbol{vec}\left(X^T(AM - B)\right) + \lambda \boldsymbol{w}$
To compute: $\nabla^2 g(\boldsymbol{w})\boldsymbol{s}$
    $A \leftarrow XS$, where $\boldsymbol{vec}\,(S) = \boldsymbol{s}$.
    $M \leftarrow H^T H$.
    Return: $\boldsymbol{vec}\left(X^T(AM)\right) + \lambda \boldsymbol{s}$
```

where $\ell''_{ij} = \ell''(Y_{ij}, \boldsymbol{w}^T \tilde{\boldsymbol{x}}_{ij})$. Let S be the $d \times k$ matrix such that $\boldsymbol{s} = \boldsymbol{vec}\,(S)$. Using the identity $(B^T \otimes A)\boldsymbol{vec}\,(X) = \boldsymbol{vec}\,(AXB)$, we have $\left((\boldsymbol{h}_j \boldsymbol{h}_j^T) \otimes (\boldsymbol{x}_i \boldsymbol{x}_i^T)\right)\boldsymbol{s} = \boldsymbol{vec}\left(\boldsymbol{x}_i \boldsymbol{x}_i^T S \boldsymbol{h}_j \boldsymbol{h}_j^T\right)$. Thus,

$$\begin{aligned} \sum_{ij} \ell''_{ij} \boldsymbol{x}_i \boldsymbol{x}_i^T S \boldsymbol{h}_j \boldsymbol{h}_j^T &= \sum_{i=1}^n \boldsymbol{x}_i \Big(\sum_{j:(i,j)\in\Omega} \ell''_{ij} \cdot (S^T \boldsymbol{x}_i)^T \boldsymbol{h}_j \boldsymbol{h}_j^T \Big) \\ &= \sum_{i=1}^n \boldsymbol{x}_i \Big(\sum_{j:(i,j)\in\Omega} U_{ij} \boldsymbol{h}_j^T \Big) = X^T U H, \end{aligned}$$

where U is sparse, and $U_{ij} = \ell''_{ij} \cdot (S^T \boldsymbol{x}_i)^T \boldsymbol{h}_j$, $\forall (i,j) \in \Omega$. Thus, we have

$$\nabla^2 g(\boldsymbol{w})\boldsymbol{s} = \boldsymbol{vec}\left(X^T U H\right) + \lambda \boldsymbol{s}. \qquad (10)$$

In Algorithm 1, we describe a detailed procedure for computing the Hessian-vector multiplication in $O\left((nnz(X) + |\Omega| + d) \times k\right)$ time using (10).

Loss Functions. See Appendix B.1 for expressions of $\ell'(a,b)$ and $\ell''(a,b)$ for three common loss functions: squared loss, logistic loss, and squared hinge loss. Thus, to solve (6), we can apply CG for squared loss and TRON (Lin et al., 2008) for the other two loss functions.

3.2. Fast Operations for Squared Loss with Full Labels

For the situation where labels are fully observed, solving (1) efficiently in the large-scale setting remains a challenge. The closed form solution from (3) is not ideal for two reasons: firstly since it involves the SVD of both X and $U_X^T Y$, the solution becomes infeasible when rank of X is large. Secondly, since it is an unregularized solution, it might overfit. Indeed CPLST has similar scalability and overfitting issues due to absence of regularization and requirement of pseudo inverse calculations for X. When Y is fully observed, Algorithm 1, which aims to handle missing labels with a general loss function, is also not scalable as $|\Omega| = nL$ imposing a $O\left(nLk + nnz(X)k\right)$ cost per operation which is prohibitive when n and L are large.

Although, for a general loss, an $O(nLk)$ cost seems to be inevitable, for the L_2 loss, we propose fast procedures such that the cost of each operation only depends on $nnz(Y)$ instead of $|\Omega|$. In most real-world multi-label problems, $nnz(Y) \ll nL = |\Omega|$. As a result, for the squared loss, our technique allows alternating minimization to be performed efficiently even when $|\Omega| = nL$.

If the squared loss is used, the matrix D in Eq. (9) is $D = XWH^T - Y$ when Y is fully observed, where W is the $d \times k$ matrix such that $\boldsymbol{vec}\,(W) = \boldsymbol{w}$. Then, we have

$$\nabla g(\boldsymbol{w}) = \boldsymbol{vec}\left(X^T X W H^T H - X^T Y H\right) + \lambda \boldsymbol{w}. \quad (11)$$

Similarly, U in Eq. (10) is $U = XSH^T$ which gives us

$$\nabla^2 g(\boldsymbol{w})\boldsymbol{s} = \boldsymbol{vec}\left(X^T X S H^T H\right) + \lambda \boldsymbol{s}. \qquad (12)$$

With a careful choice of the sequence of the matrix multiplications, we show detailed procedures in Algorithm 2, which use only $O(nk + k^2)$ extra space and $O\left((nnz(Y) + nnz(X))\,k + (n+L)k^2\right)$ time to compute both $\nabla g(\boldsymbol{w})$ and $\nabla^2 g(\boldsymbol{w})\boldsymbol{s}$ efficiently.

Remark on parallelization. As we can see, matrix multiplication acts as a crucial subroutine in both Algorithms 1 and 2. Thus, with a highly-optimized parallel BLAS library (such as ATLAS or Intel MKL), our algorithms can easily enjoy speedup brought by the parallel matrix operations provided in the library without any extra efforts. Figure 3 in Appendix E shows that both algorithms do indeed enjoy impressive speedups as the number of cores increases.

Remark on kernel extension. Given a kernel function $\mathcal{K}(\cdot,\cdot)$, let $f^j \in \mathcal{H}_{\mathcal{K}}$ be the minimizer of the empirical loss defined in Eq. (2). Then by the Representer Theorem (for example, Schölkopf et al., 2001), f^j admits a representation of the form: $f^j(\cdot; \boldsymbol{z}_j) = \sum_{t=1}^n z_{jt} \mathcal{K}(\cdot, \boldsymbol{x}_t)$, where $\boldsymbol{z}_j \in \mathbb{R}^n$. Let the vector function $\boldsymbol{k}(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^n$ for $\mathcal{K}$ be defined as $\boldsymbol{k}(\boldsymbol{x}) = [\cdots, \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_t), \cdots]^T$. Then $f(\boldsymbol{x}; Z)$ can be written as $f(\boldsymbol{x}; Z) = Z^T \boldsymbol{k}(\boldsymbol{x})$, where Z is an $n \times L$ matrix with $\boldsymbol{z}_j$ as the j-th column. Once again, we can impose the same trace norm regularization $r(Z)$ and the low rank constraint in Eq. (4). As a result, $Z = WH^T$ and $f^j(\boldsymbol{x}_i, \boldsymbol{z}_j) = \boldsymbol{k}^T(\boldsymbol{x}_i) W \boldsymbol{h}_j$. If K is the kernel Gram matrix for the training set $\{\boldsymbol{x}_i\}$ and K_i is its i^{th} column, then the loss in (4) can be replaced by $\ell(Y_{ij}, K_i^T W \boldsymbol{h}_j)$. Thus, the proposed alternating minimization can be applied to solve Equations (1) and (2) with the kernel extension as well.

4. Generalization Error Bounds

In this section we analyze excess risk bounds for our learning model with trace norm regularization. Our analysis demonstrates the superiority of our trace norm regularization-based technique over BR and Frobenius norm regularization. We require a more careful analysis for our setting since standard results do not apply because of the presence of missing labels.

Our multi-label learning model is characterized by a distribution $\mathcal{D}$ on the space of data points and labels $\mathcal{X} \times \{0,1\}^L$ where $\mathcal{X} \subseteq \mathbb{R}^d$ and a distribution that decides the pattern of missing labels. We receive n training points $(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)$ sampled i.i.d from the distribution $\mathcal{D}$, where $\boldsymbol{y}_i \in \{0,1\}^L$ are the *ground truth* label vectors. However we shall only be able to observe the ground truth label vectors $\boldsymbol{y}_i$ at s random locations. More specifically, for each i we only observe $\boldsymbol{y}_i$ at locations $l_i^1, \ldots, l_i^s \in [L]$ where the locations are chosen uniformly from the set $[L]$ and the choices are independent of $(\boldsymbol{x}_i, \boldsymbol{y}_i)$.

Given this training data, we learn a predictor $\hat{Z}$ by performing ERM over a constrained set of predictors as follows:

$$\hat{Z} = \underset{r(Z)\leq\lambda}{\arg\inf}\ \hat{\mathcal{L}}(Z) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{s} \ell(\boldsymbol{y}_i^{l_i^j}, f^{l_i^j}(\boldsymbol{x}_i; Z)),$$

where $\hat{\mathcal{L}}(Z)$ is the *empirical risk* of a predictor Z. Note that although the method in Equation 2 uses a regularized formulation that is rank-constrained, we analyze just the regularized version without the rank constrain for simplicity. As the class of rank-constrained matrices is smaller than the class of trace-norm constrained matrices, we can in fact expect better generalization performance than that indicated here, if the ERM problem can be solved exactly.

Our goal would be to show that $\hat{Z}$ has good generalization properties i.e. $\mathcal{L}(\hat{Z}) \leq \inf_{r(Z)\leq\lambda} \mathcal{L}(Z) + \epsilon$, where $\mathcal{L}(Z) := \underset{\boldsymbol{x},\boldsymbol{y},l}{\mathbb{E}} \left[\!\left[\ell(\boldsymbol{y}^l, f^l(\boldsymbol{x}; Z)) \right]\!\right]$ is the *population risk* of a predictor.

Theorem 3. *Suppose we learn a predictor using the formulation $\hat{Z} = \underset{\|Z\|_{\mathrm{tr}}\leq\lambda}{\arg\inf}\ \hat{\mathcal{L}}(Z)$ over a set of n training points. Then with probability at least $1-\delta$, we have*

$$\mathcal{L}(\hat{Z}) \leq \inf_{\|Z\|_{\mathrm{tr}}\leq\lambda} \mathcal{L}(Z) + \mathcal{O}\left(s\lambda\sqrt{\frac{1}{n}}\right) + \mathcal{O}\left(s\sqrt{\frac{\log\frac{1}{\delta}}{n}}\right),$$

where we assume (w.l.o.g.) that $\mathbb{E}\left[\!\left[\|\boldsymbol{x}\|_2^2\right]\!\right] \leq 1$.

We refer to Appendix C for the proof. Interestingly, we can show that our analysis, obtained via uniform convergence bounds, is tight and cannot be improved in general. We refer the reader to Appendix D.1 for the tightness argument. However, it turns out that Frobenius norm regularization is also able to offer the same excess risk bounds and thus, this result does not reveal any advantage for trace norm regularization. Nevertheless, we can still get improved bounds for a general class of distributions over $(\boldsymbol{x}, \boldsymbol{y})$:

Theorem 4. *Let the data distribution satisfy the following conditions: 1) The top singular value of the covariance matrix $X = \underset{\boldsymbol{x}\sim\mathcal{D}}{\mathbb{E}} \left[\!\left[\boldsymbol{x}\boldsymbol{x}^\top\right]\!\right]$ is $\|X\|_2 = \sigma_1$, 2) $tr(X) = \Sigma$ and 3) the distribution on $\mathcal{X}$ is sub-Gaussian i.e. for some $\eta > 0$, for all $\boldsymbol{v} \in \mathbb{R}^d$, $\mathbb{E}\left[\!\left[\exp\left(x^\top \boldsymbol{v}\right)\right]\!\right] \leq \exp\left(\|\boldsymbol{v}\|_2^2\,\eta^2/2\right)$, then with probability at least $1-\delta$, we have*

$$\mathcal{L}(\hat{Z}) \leq \inf_{\|Z\|_{\mathrm{tr}}\leq\lambda} \mathcal{L}(Z) + \mathcal{O}\left(s\lambda\sqrt{\frac{d(\eta^2+\sigma_1)}{nL\Sigma}} + s\sqrt{\frac{\log\frac{1}{\delta}}{n}}\right).$$

In particular, if the data points are generated from a unit normal distribution, then we have

$$\mathcal{L}(\hat{Z}) \leq \inf_{\|Z\|_{\mathrm{tr}}\leq\lambda} \mathcal{L}(Z) + \mathcal{O}\left(s\lambda\sqrt{\frac{1}{nL}}\right) + \mathcal{O}\left(s\sqrt{\frac{\log\frac{1}{\delta}}{n}}\right).$$

The proof of Theorem 4 can be found in Appendix C. Our proofs do not follow either from existing techniques for learning with matrix predictors (for instance (Kakade et al., 2012)) or from results on matrix completion with trace norm regularization (Shamir & Shalev-Shwartz, 2011) due to the complex interplay of feature vectors and missing labels that we encounter in our learning model. Instead, our results utilize a novel form of Rademacher averages, bounding which requires tools from random matrix theory. We note that our results can even handle non-uniform sampling of labels (see Theorem 6 in Appendix C for details).

We note that the assumptions on the data distribution are trivially satisfied with finite σ_1 and η by any distribution with support over a compact set. However, for certain distributions, this allows us to give superior bounds for trace norm regularization. We note that Frobenius norm regularization can give no better than a $\left(\frac{\lambda}{\sqrt{n}}\right)$ style excess error bound even for such distributions (see Appendix D.2 for a proof), whereas trace norm regularization allows us to get superior $\left(\frac{\lambda}{\sqrt{nL}}\right)$ style bounds. This is especially contrasting when, for instance, $\lambda = \mathcal{O}(\sqrt{L})$, in which case trace norm regularization gives $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ excess error whereas the excess error for Frobenius regularization deteriorates to $\mathcal{O}\left(\sqrt{\frac{L}{n}}\right)$. Thus, trace norm seems better suited to exploit situations where the data distribution is isotropic.

Intuitively, we expect such results due to the following reason: when labels are very sparsely observed, such as when $s = \mathcal{O}(1)$, we observe the value of each label on $\mathcal{O}(n/L)$ training points. In such a situation, Frobenius norm regularization with say $\lambda = \sqrt{L}$ essentially allows an independent

Table 1. Data statistics. d and L are the number of features and labels, respectively, and $\bar{d}$ and $\bar{L}$ are the average number of nonzero features and positive labels in an instance, respectively.

			Training set			Test set		
Dataset	d	L	n	$\bar{d}$	$\bar{L}$	n	$\bar{d}$	$\bar{L}$
bibtex	1,836	159	4,880	68.74	2.40	2,515	68.50	2.40
autofood	9,382	162	155	143.92	15.80	38	143.71	13.71
compphys	33,284	208	161	792.78	9.80	40	899.02	11.83
delicious	500	983	12,920	18.17	19.03	3,185	18.80	19.00
eurlex	5,000	3,993	17,413	236.69	5.30	1,935	240.96	5.32
nus-wide	1,134	1,000	161,789	862.70	5.78	107,859	862.94	5.79
wiki	366,932	213,707	881,805	146.78	7.06	10,000	147.78	7.08

Table 2. Comparison of LEML with various loss functions and WSABIE on smaller datasets. SQ denotes squared loss, LR denotes logistic regression loss, and SH denotes squared hinge loss

		Top-3 Accuracy				Average AUC			
		LEML				LEML			
	k/L	SQ	LR	SH	WSABIE	SQ	LR	SH	WSABIE
bibtex	20%	**34.16**	25.65	27.37	28.77	0.8910	0.8677	0.8541	**0.9055**
	40%	**36.53**	28.20	24.81	30.05	0.9015	0.8809	0.8467	**0.9092**
	60%	**38.00**	28.68	23.26	31.11	0.9040	0.8861	0.8505	**0.9089**
autofood	20%	**81.58**	80.70	**81.58**	66.67	0.9565	**0.9598**	0.9424	0.8779
	40%	76.32	**80.70**	78.95	70.18	0.9277	**0.9590**	0.9485	0.8806
	60%	70.18	80.70	**81.58**	60.53	0.8815	**0.9582**	0.9513	0.8518
compphys	20%	**80.00**	**80.00**	**80.00**	49.17	0.9163	0.9223	**0.9274**	0.8212
	40%	**80.00**	78.33	79.17	39.17	**0.9199**	0.9157	0.9191	0.8066
	60%	**80.00**	**80.00**	**80.00**	49.17	**0.9179**	0.9143	0.9098	0.8040
delicious	20%	**61.20**	53.68	57.27	42.87	0.8854	0.8588	**0.8894**	0.8561
	40%	**61.23**	49.13	52.95	42.05	0.8827	0.8534	**0.8868**	0.8553
	60%	**61.15**	46.76	49.58	42.22	0.8814	0.8517	**0.8852**	0.8523

predictor $z_l \in \mathbb{R}^d$ to be learned for each label $l \in [L]$. Since all these predictors are being trained on only $\mathcal{O}\left(n/L\right)$ training points, the performance accordingly suffers.

On the other hand, if we were to train a single predictor for all the labels i.e. $Z = z\mathbf{1}^\top$ for some $z \in \mathbb{R}^d$, such a predictor would be able to observe $O(n)$ points and consequently have much better generalization properties. Note that this predictor also satisfies $\|z\mathbf{1}^\top\|_{\text{tr}} \leq \sqrt{L}$. This seems to indicate that trace norm regularization can capture cross label dependencies, especially in the presence of missing labels, much better than Frobenius norm regularization.

Having said that, it is important to note that trace norm and Frobenius norm regularization induce different biases in the learning framework. It would be interesting to study the bias-variance trade-offs offered by these two regularization techniques. However, in presence of label correlations we expect both formulations to suffer similar biases.

5. Experimental Results

We now evaluate our proposed algorithms in terms of accuracy and stability. This discussion shall demonstrate the superiority of our method over other approaches.

Datasets. We considered a variety of benchmark datasets including four standard datasets (bibtex, delicious, eurlex, and nus-wide), two datasets with $d \gg L$ (autofood and compphys), and a very large scale Wikipedia based dataset, which contains about 1M wikipages and 200K labels. See Table 1 for more information about the datasets. We conducted all experiments on an Intel machine with 32 cores.

Competing Methods. A list containing details of the competing methods (including ours) is given below. Note that CS (Hsu et al., 2009) and PLST (Tai & Lin, 2012) are not included as they are shown to be suboptimal to CPLST and BCS in (Chen & Lin, 2012; Kapoor et al., 2012).

1. LEML (**L**ow rank **E**mpirical risk minimization for **M**ulti-Label **L**earning): our proposed method. We implemented CG with Algorithms 1 and 2 for squared loss, and TRON (Lin et al., 2008) with Algorithm 1 for logistic and squared hinge loss.
2. CPLST: the method proposed in (Chen & Lin, 2012). We used code provided by the authors.
3. BCS: the method proposed in (Kapoor et al., 2012). We used code provided by the authors.
4. BR: Binary Relevance with various loss functions.
5. WSABIE: Due to lack of publicly available code, we implemented this method and hand-tuned learning rates and the margins for each dataset as suggested by the authors of WSABIE (Weston, 2013).

Evaluation Criteria. We used three criteria to compare the methods: top-K accuracy (performance on a few top predictions), Hamming loss (overall classification performance), and average AUC (ranking performance). See Appendix E.1 for details.

5.1. Results with full labels

We divide datasets into two groups: *small datasets* (bibtex, autofood, compphys, and delicious) to which all methods are able to scale and *large datasets* (eurlex, nus-wide, and wiki) to which only LEML and WSABIE are able to scale.

Small datasets. We first compare dimension reduction based approaches to assess their performance with varying dimensionality reduction ratios. Figure 1 presents these results for LEML, CPLST and BCS on the squared L_2 loss with BR included for reference. Clearly LEML consistently outperforms other methods for all ratios. Next we compare LEML to WSABIE with three surrogates (squared, logistic, and L_2-hinge), which approximately optimize a weighted approximate ranking loss. Table 2 shows that although the best loss function for each dataset varies, LEML is always superior to or competitive with WSABIE. Based on Figure 1, Table 2, and further results in Appendix E.3, we make the following observations. 1) LEML can deliver accuracies competitive with BR even with a severe reduction in dimensionality, 2) On bibtex and compphys, LEML is even shown to outperform BR. This is a benefit brought forward by the design of LEML, wherein the relation between labels can be captured by a low rank Z. This enables LEML to better utilize label information than BR and yield better accuracies. 3) On autofood and compphys, CPLST seems to suffer from overfitting and demonstrates a significant dip in performance. In contrast, LEML, which brings regularization into the formulation performs well consistently on all datasets.

Table 3. Comparison of LEML and WSABIE on large datasets

		LEML				WSABIE			
dataset	k	time (s)	top-1	top-3	AUC	time (s)	top-1	top-3	AUC
eurlex	250	**175**	**51.99**	**39.79**	**0.9425**	373	33.13	25.01	0.8648
	500	**487**	**56.90**	**44.20**	**0.9456**	777	31.58	24.00	0.8651
nus-wide	50	**574**	**20.71**	**15.96**	**0.7741**	4,705	14.58	11.37	0.7658
	100	**1,097**	**20.76**	**16.00**	**0.7718**	6,880	12.46	10.21	0.7597
wiki	250	**9,932**	**19.56**	14.43	**0.9086**	79,086	18.91	**14.65**	0.9020
	500	**18,072**	**22.83**	**17.30**	**0.9374**	139,290	19.20	15.66	0.9058

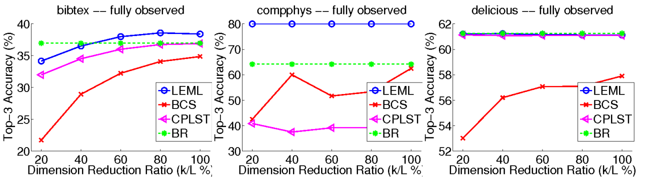
Table 4. Comparison between various dimensionality reduction approaches on Y with 20% observed entries, and $k = 0.4L$.

	Top-3 Accuracy			Hamming loss			Average AUC		
	LEML	BCS	BR	LEML	BCS	BR	LEML	BCS	BR
bibtex	**28.50**	23.84	25.78	**0.0136**	0.2496	0.0193	**0.8332**	0.7871	0.8087
autofood	**67.54**	35.09	62.28	**0.0671**	0.2445	0.0760	**0.8634**	0.6322	0.8178
compphys	**65.00**	35.83	31.67	**0.0518**	0.2569	0.0566	**0.7964**	0.6442	0.7459

Larger data. Table 3 shows results for LEML and WSABIE on the three larger datasets. We implemented LEML with the squared L_2 loss using Algorithm 2 for comparison in the full labels case. Note that Hamming loss is not used here as it is not clear how to convert the label ranking given by WSABIE to a 0/1 encoding. For LEML, we report the time and the accuracies obtained after five alternating iterations. For WSABIE, we ran the method on each dataset with the hand-tuned parameters for about two days, and reported the time and results for the epoch with the highest average AUC. On eurlex and nus-wide, LEML is clearly superior than WSABIE on all evaluation criteria. On wiki, although both methods share a similar performance for $k = 250$, on increasing k to 500, LEML again outperforms WSABIE. Also clearly noticeable is the stark difference in the running times of the two methods. Whereas LEML takes less than 6 hours to deliver 0.9374 AUC on wiki, WSABIE requires about 1.6 days to achieve 0.9058 AUC. More specifically, WSABIE takes about 7,000s for the first epoch, 16,000s for the second and 36,000s for the third epoch which result in it spending almost two days on just 5 epochs. Although this phenomenon is expected due to the sampling scheme in WSABIE (Weston et al., 2010), it becomes more serious as L increases. We leave the issue of designing a better sampling scheme with large L for future work. Figure 2a further illustrates this gap in training times for the nus-wide dataset. All in all, the results clearly demonstrate the scalability and efficiency of LEML.

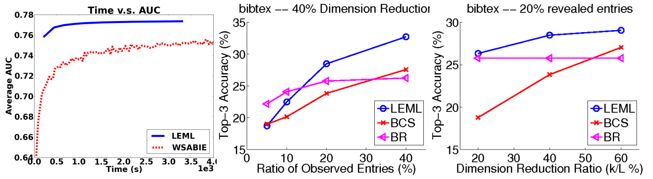
5.2. Results with missing labels

For experiments with missing labels, we compare LEML, BCS, and BR. We implemented BR with missing labels by learning an L_2-regularized binary classifier/regressor for each label on observed instances. Thus, the model derived from BR corresponds to the minimizer of (2) with Frobenius norm regularization. Table 4 shows the results when 20% entries were revealed (i.e. 80% missing rate) and squared loss function was used for training. We used $k = 0.4L$ for both LEML and BCS. The results clearly show that LEML outperforms BCS and LEML with respect to all three evaluation criteria. On bibtex, we further present results for various rates of observed labels in Figure 2b and results for various dimension reduction ratios in Figure 2c. LEML clearly shows superior performance over other approaches, which corroborates the theoretical results of Section 4 that indicate better generalization performance for low-rank promoting regularizations. More empirical results for other loss functions, various observed ratios and dimension reduction ratios can be found in Appendix E.4.



(a) bibtex (b) compphys (c) delicious

Figure 1. Comparison between different dimension reduction methods with fully observed Y by varying the reduction ratio.



(a) (b) (c)

Figure 2. Results for (a): running time on nus-wide. (b): various observed ratios on bibtex. (c): various reduction ratios on bibtex.

6. Conclusion

In this paper we studied the multi-label learning problem with missing labels in the standard ERM framework. We modeled our framework with rank constraints and regularizers to increase scalability and efficiency. To solve the obtained non-convex problem, we proposed an alternating minimization based method that critically exploits structure in the loss function to make our method scalable. We showed that our learning framework admits excess risk bounds that indicate better generalization performance for our methods than the existing methods like BR, something which our experiments also confirmed. Our experiments additionally demonstrated that our techniques are much more efficient than other large scale multi-label classifiers and give superior performance than the existing label compression based approaches. For future work, we would like to extend LEML to other (non decomposable) loss functions such as ranking losses and study conditions under which alternating minimization for our problem is guaranteed to converge to the global optimum. Another open question is if our risk bounds can be improved by avoiding the uniform convergence route that we use in the paper.

Acknowledgments

We thank a reviewer for pointing out classical results in the rank-reduced regression literature. This research was supported by NSF grant CCF-1117055.

References

Agrawal, Rahul, Gupta, Archit, Prabhu, Yashoteja, and Varma, Manik. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the International World Wide Web Conference*, 2013.

Bertsekas, Dimitri P. *Nonlinear Programming*. Athena Scientific, Belmont, MA 02178-9998, second edition, 1999.

Breiman, Leo and Friedman, Jerome H. Predicting Multivariate Responses in Multiple Linear Regression. *Journal of the Royal Stat. Soc.: Series B*, 59(1):3–54, 1997.

Bucak, Serhat Selcuk, Mallapragada, Pavan Kumar, Jin, Rong, and Jain, Anil K. Efficient multi-label ranking for multi-class learning: Application to object recognition. In *Proceedings of IEEE International Conference on Computer Vision*, 2009.

Bucak, Serhat Selcuk, Jin, Rong, and Jain, Anil K. Multi-label learning with incomplete class assignments. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

Carneiro, Gustavo, Chan, Antoni B., Moreno, Pedro J., and Vasconcelos, Nuno. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):394–410, 2007.

Chen, Yao-Nan and Lin, Hsuan-Tien. Feature-aware label space dimension reduction for multi-label classification. In Bartlett, P., Pereira, F.C.N., Burges, C.J.C., Bottou, L., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1538–1546, 2012.

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Hariharan, Bharath, Zelnik-Manor, Lihi, Vishwanathan, S. V. N., and Varma, Manik. Large scale max-margin multi-label classification with priors. In *Proceedings of the International Conference on Machine Learning*, June 2010.

Hsu, Daniel, Kakade, Sham, Langford, John, and Zhang, Tong. Multi-label prediction via compressed sensing. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22*, pp. 772–780, 2009.

Izenman, Alan Julian. Reduced-Rank Regression for the Multivariate Linear Model. *Journal of Multivariate Analysis*, 5:248–264, 1975.

Kakade, Sham M., Sridharan, Karthik, and Tewari, Ambuj. On the Complexity of Linear Prediction: Risk Bounds, Margin Bounds, and Regularization. In Koller, Daphne, Schuurmans, Dale, Bengio, Yoshua, and Bottou, Léon (eds.), *Advances in Neural Information Processing Systems 21*, 2008.

Kakade, Sham M., Shalev-Shwartz, Shai, and Tewari, Ambuj. Regularization Techniques for Learning with Matrices. *Journal of Machine Learning Research*, 13:1865–1890, 2012.

Kapoor, Ashish, Viswanathan, Raajay, and Jain, Prateek. Multilabel classification using bayesian compressed sensing. In Bartlett, P., Pereira, F.C.N., Burges, C.J.C., Bottou, L., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 2654–2662, 2012.

Ledoux, Michel and Talagrand, Michel. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, 2002.

Lin, Chih-Jen, Weng, Ruby C., and Keerthi, S. Sathiya. Trust region Newton method for large-scale logistic regression. *Journal of Machine Learning Research*, 9:627–650, 2008.

Schölkopf, Bernhard, Herbrich, Ralf, and Smola, Alex J. A Generalized Representer Theorem. In *14th Annual Conference on Computational Learning Theory*, pp. 416–426, 2001.

Shamir, Ohad and Shalev-Shwartz, Shai. Collaborative Filtering with the Trace Norm: Learning, Bounding, and Transducing. In *24th Annual Conference on Learning Theory*, 2011.

Sun, Liang, Ji, Shuiwang, and Ye, Jieping. Canonical correlation analysis for multi-label classification: A least squares formulation, extensions and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):194–200, 2011.

Tai, Farbound and Lin, Hsuan-Tien. Multi-label classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.

Vershynin, Roman. *Introduction to the non-asymptotic analysis of random matrices*, chapter 5 of Compressed Sensing, Theory and Applications, pp. 210–268. Cambridge University Press, 2012.

Wang, Changhu, Yan, Shuicheng, Zhang, Lei, and Zhang, Hong-Jiang. Multi-Label Sparse Coding for Automatic Image Annotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

Weston, Jason. Personal Communication, 2013.

Weston, Jason, Bengio, Samy, and Usunier, Nicolas. Large scale image annotation: learning to rank with joint word-image embeddings. *Mach. Learn.*, 81(1):21–35, October 2010.