Constrained Submodular Minimization for Missing Labels and Class Imbalance in Multi-label Learning

Constrained Submodular Minimization for Missing Labels and Class Imbalance in Multi-label Learning

Baoyuan Wu
KAUST, Saudi Arabia

Siwei Lyu
SUNY-Albany, NY USA

Bernard Ghanem
KAUST, Saudi Arabia

Abstract

In multi-label learning, there are two main challenges: missing labels and class imbalance (CIB). The former assumes that only a partial set of labels are provided for each training instance while other labels are missing. CIB is observed from two perspectives: first, the number of negative labels of each instance is much larger than its positive labels; second, the rate of positive instances (i.e. the number of positive instances divided by the total number of instances) of different classes are significantly different. Both missing labels and CIB lead to significant performance degradation. In this work, we propose a new method to handle these two challenges simultaneously. We formulate the problem as a constrained submodular minimization that is composed of a submodular objective function that encourages label consistency and smoothness, as well as, class cardinality bound constraints to handle class imbalance. We further present a convex approximation based on the Lovasz extension of submodular functions, leading to a linear program, which can be efficiently solved by the alternative direction method of multipliers (ADMM). Experimental results on several benchmark datasets demonstrate the improved performance of our method over several state-of-the-art methods.

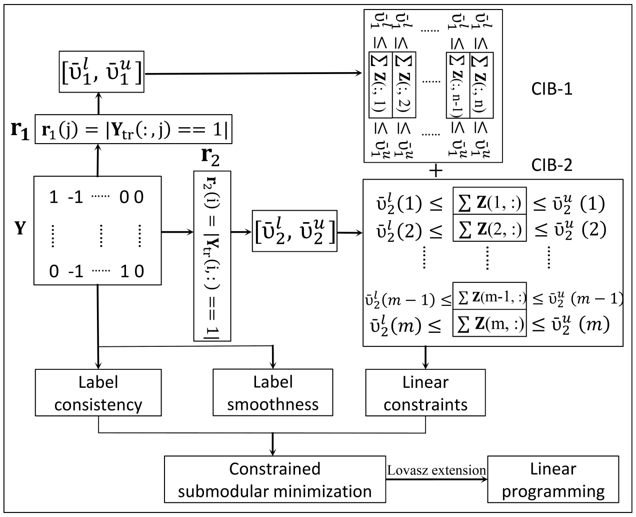
1 Introduction

Multi-label learning (ML) assumes that one instance can be assigned to multiple classes simultaneously. For example, one image can be annotated with several tags, and one document can be associated with multiple topics. Although many multi-label learning methods have been proposed in recent years, a main challenge remains for this problem, i.e., the lack of completely labeled training instances. This is important because in many real life applications, most training instances are only partially labeled, while other labels are not provided or missing. One such example is image annotation, a human labeler can only feasibly annotates each training image with a subset of tags, especially when the number of classes/tags is large. Learning from such partially labeled instances is referred to as the *multi-label learning with missing labels* (MLML) problem (Wu et al. 2014; Yu et al. 2014).

Several previous works have tried to handle the MLML problem, such as (Goldberg et al. 2010; Cabral et al. 2011; Kapoor, Viswanathan, and Jain 2012; Xu, Jin, and Zhou 2013; Wu et al. 2014; Yu et al. 2014; Wu et al. 2015; Chen et al. 2015). However, most of them disregard another important challenge in multi-label learning, i.e., class imbalance (CIB), which has two different phenomena. Firstly, each instance is assigned with only a few positive labels, while most other labels are negative. We refer to this type of class imbalance as CIB-1 in this work. Secondly, the proportions of positive instances of different classes may be significantly different. We refer to this type of CIB as CIB-2 in this work. CIB-1 often occurs in binary classification problems, while CIB-2 is more widely encountered in multi-label classification problems. It has been observed (Sahare and Gupta 2012; Zhang and Hu 2014) that both CIB-1 and CIB-2 are likely to lead to the performance degradation of many popular models, such as SVM and neural networks.

The more challenging case is missing labels and two types of class imbalances co-exist in a multi-label learning problem. This is becuase the bias between the positive and negative labels becomes larger than where there is no missing labels. Thus the missing labels tend to be negative labels with the larger degree, as the amount of missing labels increases. To date, there only exist a few works (Zhang, Li, and Liu 2015; Petterson and Caetano 2010) specifically consider CIB-1, but no existing work can handle CIB-1 and CIB-2 simultaneously, in particular in the context of MLML.

The goal of this work is to develop a unified model to handle missing labels and class imbalance jointly. We formulate the problem as a transductive learning problem that include five components that are label consistency, instance-level and class-level label smoothness, and two types of class cardinality (lower and upper) bounds. The first three components are used to propagate the label information from the provided labels to missing labels, and the latter two components are included to handle two types of the class imbalance problem. We first formulate a unified model that combines these components as a constrained submodular minimization problem (CSM). However, due to the class cardinality constraint, it is a NP-hard problem. Utilizing the Lovasz extension of submodular function, we approximate CSM by a continuous linear programming (LP) with linear constraints. The LP problem is efficiently solved by alternative direction method of multipliers (ADMM). As the final output, the signs of the continuous labels are adopted as the discrete



$[\bar{\upsilon}_1^l, \bar{\upsilon}_1^u]$
$\mathbf{r_1}$
$\mathbf{r}_1(j) = |\mathbf{Y}_{tr}(:, j) == 1|$
$\mathbf{r_2}$
$\mathbf{r}_2(i) = |\mathbf{Y}_{tr}(i, :) == 1|$
Y
1 -1 ...... 0 0
0 -1 ...... 1 0
$[\bar{\upsilon}_2^l, \bar{\upsilon}_2^u]$
$\bar{\upsilon}_1^l \leq \sum \mathbf{Z}(:, 1) \leq \bar{\upsilon}_1^u$
$\bar{\upsilon}_1^l \leq \sum \mathbf{Z}(:, 2) \leq \bar{\upsilon}_1^u$
$\bar{\upsilon}_1^l \leq \sum \mathbf{Z}(:, n\text{-}1) \leq \bar{\upsilon}_1^u$
$\bar{\upsilon}_1^l \leq \sum \mathbf{Z}(:, n) \leq \bar{\upsilon}_1^u$
CIB-1
+
CIB-2
$\bar{\upsilon}_2^l(1) \leq \sum \mathbf{Z}(1, :) \leq \bar{\upsilon}_2^u(1)$
$\bar{\upsilon}_2^l(2) \leq \sum \mathbf{Z}(2, :) \leq \bar{\upsilon}_2^u(2)$
$\bar{\upsilon}_2^l(m-1) \leq \sum \mathbf{Z}(m\text{-}1, :) \leq \bar{\upsilon}_2^u(m-1)$
$\bar{\upsilon}_2^l(m) \leq \sum \mathbf{Z}(m, :) \leq \bar{\upsilon}_2^u(m)$
Label consistency
Label smoothness
Linear constraints
Constrained submodular minimization
Lovasz extension
Linear programming

Figure 1: The overall framework of the proposed model.

class labels. A graphical illustration of the overall framework is presented in Figure 1.

The main contributions of this work include: (1) we propose a unified model to jointly handle missing labels and class imbalance in multi-label learning by incorporating several types of prior knowledge; (2) the unified framework is formulated as a constrained submodular minimization problem, of which a convex approximation is also provided based on the Lovasz extension of the submodular function; (3) experiments on several benchmark multi-label datasets verify the efficacy of the proposed method, on handling both missing labels and CIB, as well as its improvements over the state-of-the-art methods.

2 Related Work

We briefly review multi-label learning methods that handle missing labels and handle class imbalance.

Multi-label learning methods handling missing labels can be generally partitioned into four categories. **(i)** Missing labels can be treated as negative labels, which will bring in undesired label bias, such as (Chen et al. 2008; Sun, Zhang, and Zhou 2010; Bucak, Jin, and Jain 2011; Chen, Zheng, and Weinberger 2013; Wang et al. 2014; Wang, Si, and Zhang 2014; Chen et al. 2015). When massive missing labels exist, many ground-truth positive labels will be incorrectly initialized as negative labels, leading to significant performance degradation. **(ii)** Missing labels can be augmented into the label set as a special type of labels. Wu et al. (Wu et al. 2014; 2015) propose to use three different labels, including positive labels $+1$, negative labels -1, and missing labels 0 to model the learning problem. A similar setting of using $\{1, 0, \frac{1}{2}\}$ is also used (Wu, Lyu, and Ghanem 2015). The label bias is avoided in these three works. **(iii)** Techniques in matrix completion (MC) (Johnson 1990) is borrowed to handle missing labels, as in (Goldberg et al. 2010; Cabral et al. 2011; Xu, Jin, and Zhou 2013; Yu et al. 2014). In (Goldberg et al. 2010; Cabral et al. 2011; Xu, Jin, and Zhou 2013), a recent work known as LEML (Yu et al. 2014) proposes an empirical risk minimization (ERM) framework to handle missing labels. Both MC-based methods and LEML exploit the mask matrix to avoid the label bias and utilize the low rank assumption to explicitly embed the label dependencies. **(iv)** Missing labels can be treated as latent variables in probabilistic models, including Bayesian networks (Kapoor, Viswanathan, and Jain 2012; Vasisht et al. 2014; Bi and Kwok 2014) and conditional RBMs (Li, Zhao, and Guo 2015). However, all the aforementioned work assumes balanced positive and negative training data labels, though some of them simply set different costs to positive and negative labels in the loss function. Class imbalance has not been well handled in these models.

On the other hand, class imbalance (CIB) has not been extensively studied in the multi-label learning context. There are two main categories of methods here. One category is to directly maximize the imbalance-specific metric in multi-label learning, such as the $F_{1-macro}$ in (Petterson and Caetano 2010; Dembczynski et al. 2013). Another category is recently proposed in (Zhang, Li, and Liu 2015), where the binary-class imbalance classifier for the current class and the multi-class imbalance classifier for other classes are aggregated to make final predictions. However, it only considers CIB-1 in supervised multi-label learning scenarios, while CIB-2 and missing labels are ignored. In contrast, we handle CIB by embedding linear constraints, which are independent of any specific metric or base-learner. Moreover, to the best of our knowledge, no previous work has been developed to handle missing labels and the two CIB challenges jointly.

3 Proposed Model

Given a multi-dimensional dataset $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n) \in \mathrm{R}^{d \times n}$, each instance $\mathbf{x}_i$ corresponds to a multi-dimensional data vector and is associated with m different classes $\{c_1, \ldots, c_m\}$. An incomplete label matrix $\mathbf{Y} \in \{-1, 0, +1\}^{m \times n}$ is also provided, where $\mathbf{Y}_{ji} = +1$ means $\mathbf{x}_i$ is associated with c_j (i.e. the positive label), $\mathbf{Y}_{ji} = -1$ means c_j does not exist in $\mathbf{x}_i$ (i.e. the negative label), and $\mathbf{Y}_{ji} = 0$ denotes the missing label. The missing label proportion in the training label matrix is denoted as $\varepsilon = \frac{|Y_{tr}==0|}{n_{tr} \times m}$, with $|Y_{tr} == 0|$ indicating the number of zero entries in $\mathbf{Y}_{tr}$. The training label matrix $\mathbf{Y}_{tr}$ corresponds to the instances with at least 1 provided label (-1 or $+1$) and n_{tr} denotes the number of training instances. The positive label rate in the whole label matrix is denoted as $\eta = \frac{|\mathbf{Y}==1|}{n \times m}$. Our goal is to obtain a complete label matrix $\mathbf{Z} \in \{-1, +1\}^{m \times n}$, based on $\mathbf{X}$ and $\mathbf{Y}$. To this end, we use label dependencies among the instances and the classes to propagate the label information from the provided labels to the missing labels, as well as, adopt cardinality constraints to avoid degenerate results due to CIB. We adopt three types of information, including *label consistency, label smoothness*, and *class cardinality bounds*, of which the definitions will be presented in the following sections.

Label Consistency

Label consistency serves as the loss function. For $\mathbf{Y}_P = \mathbf{P} \circ \mathbf{Y}$ ('$\circ$' denotes Hadamard product), we have

$$\ell(\mathbf{Z}, \mathbf{Y}) = \sum_{i,j}^{m,n} \mathbf{P}_{ij} \mathbf{Y}_{ij} (\mathbf{Y}_{ij} - \mathbf{Z}_{ij}) = \text{tr}(\mathbf{Y}_P^\top (\mathbf{Y} - \mathbf{Z})). \quad (1)$$

When $\mathbf{Y}_{ij} = \pm 1$, if $\mathbf{Z}_{ij} \neq \mathbf{Y}_{ij}$, then $\ell(\mathbf{Z}_{ij}, \mathbf{Y}_{ij}) = 2\mathbf{P}_{ij}$, while $\ell(\mathbf{Z}_{ij}, \mathbf{Y}_{ij}) = 0$ if $\mathbf{Z}_{ij} = \mathbf{Y}_{ij}$. Besides, $\ell(\mathbf{Z}_{ij}, \mathbf{Y}_{ij} = 0) = 0$ indicates the missing labels do not contribute to the loss function. The label cost matrix $\mathbf{P}$ is defined as follows: if $\mathbf{Y}_{ij} = +1$, $\mathbf{P}_{ij} = \tau_+^i$; if $\mathbf{Y}_{ij} = -1$, $\mathbf{P}_{ij} = \tau_-^i$; if $\mathbf{Y}_{ij} = 0$, $\mathbf{P}_{ij} = 0$. τ_+^i / τ_-^i is the ratio between the number of negative and positive instances in class c_i.

Label Smoothness

Instance-level label smoothness (Wu et al. 2014) is

$$\text{tr}(\mathbf{Z}\mathbf{L}_X\mathbf{Z}^\top) = \sum_{k,i,j}^{m,n,n} \frac{\mathbf{W}_X(i,j)}{2} \Big(\frac{\mathbf{Z}_{ki}}{\sqrt{\mathbf{d}_X(i)}} - \frac{\mathbf{Z}_{kj}}{\sqrt{\mathbf{d}_X(j)}} \Big)^2,$$

where the instance similarity is $\mathbf{W}_X(i,j) = \mathbf{W}_X(j,i) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma_i \sigma_j}), i \neq j$, and $\mathbf{W}_X(i,i) = 0$. The kernel size σ_i and σ_j are determined by following the setting in (Wu et al. 2014). The normalized Laplacian matrix is $\mathbf{L}_X = \mathbf{I} - \mathbf{D}_X^{-\frac{1}{2}} \mathbf{W}_X \mathbf{D}_X^{-\frac{1}{2}}$ with $\mathbf{D}_X = \text{diag}(\mathbf{d}_X(1), \cdots, \mathbf{d}_X(n))$. $\mathbf{W}_X$ is viewed as a weighted graph $\mathcal{G}_X = \{\mathcal{V}_X, \mathcal{E}_X\}$ among instances, with $\mathcal{V}_X = \{1, \ldots, n\}$ and $\mathcal{E}_X = \{1, \ldots, n_{e_X}\}$. n_{e_X} indicates the number of edges in $\mathcal{G}_X$, which is half the number of non-zero entries in $\mathbf{W}_X$. The edge between nodes e_1 and e_2 is indexed as $e \in \{1, \ldots, n_{e_X}\}$, with its weight being $\mathbf{W}_X(e_1, e_2)$.

Class-level label smoothness (Wu et al. 2014) is

$$\text{tr}(\mathbf{Z}^\top \mathbf{L}_C \mathbf{Z}) = \sum_{k,i,j}^{n,m,m} \frac{\mathbf{W}_C(i,j)}{2} \Big(\frac{\mathbf{Z}_{ik}}{\sqrt{\mathbf{d}_C(i)}} - \frac{\mathbf{Z}_{jk}}{\sqrt{\mathbf{d}_C(j)}} \Big)^2,$$

where the class co-occurrence is $\mathbf{W}_C(i,j) = \frac{\langle \overline{\mathbf{Y}}_{i\cdot}, \overline{\mathbf{Y}}_{j\cdot} \rangle}{\|\overline{\mathbf{Y}}_{i\cdot}\| \cdot \|\overline{\mathbf{Y}}_{j\cdot}\|}$ when $i \neq j$ and $\mathbf{W}_C(i,i) = 0$. $\overline{\mathbf{Y}}_{i\cdot} = (\mathbf{Y}_{i\cdot} == 1) \in \{0,1\}^{1 \times n}$, $\mathbf{d}_C(i) = \sum_j^m \mathbf{W}_C(i,j)$. $\mathbf{L}_C = \mathbf{I} - \mathbf{D}_C^{-\frac{1}{2}} \mathbf{W}_C \mathbf{D}_C^{-\frac{1}{2}}$ with $\mathbf{D}_C = \text{diag}(\mathbf{d}_C(1), \cdots, \mathbf{d}_C(m))$. Similar to $\mathbf{W}_X$, $\mathbf{W}_C$ constitutes a weighted graph $\mathcal{G}_C$ among classes, where $\mathcal{G}_C = \{\mathcal{V}_C, \mathcal{E}_C\}$ with $\mathcal{V}_C = \{1, \ldots, m\}$ and $\mathcal{E}_X = \{1, \ldots, n_{e_C}\}$. n_{e_C} equals to the half number of the non-zero entries in $\mathbf{W}_C$, and the weight of edge e is $\mathbf{W}_C(e_1, e_2)$.

Note that, to the best of our knowledge, above two smoothness assumptions were firstly proposed in (Chen et al. 2008), and they have been borrowed and cited in some more recent works, such as (Wu et al. 2014; 2015) etc. Due to the space limit, we cannot cover all work that also use the similar smoothness assumptions.

Class Cardinality Bounds

We introduce two types of class cardinality bounds to handle class imbalance. The first type constrains the number of positive labels of each instance in range $[v_1^l, v_1^u] \subset [1, m]$. The second type requires the number of positive instances of class $c_i \in [\boldsymbol{v}_2^l(i), \boldsymbol{v}_2^u(i)] \subset [1, n]$. We formulate them as linear constraints:

$$\text{CIB-1:} \quad \overline{v}_1^l \mathbf{1}_n^\top \leq \mathbf{1}_m^\top \mathbf{Z} \leq \overline{v}_1^u \mathbf{1}_n^\top, \quad (2)$$

$$\text{CIB-2:} \quad \overline{\boldsymbol{v}}_2^l \leq \mathbf{Z}\mathbf{1}_n \leq \overline{\boldsymbol{v}}_2^u, \quad (3)$$

where $\overline{v}_1^l = 2v_1^l - m, \overline{v}_1^u = 2v_1^u - m$ and $\overline{\boldsymbol{v}}_2^l = 2\boldsymbol{v}_2^l - n\mathbf{1}_m, \overline{\boldsymbol{v}}_2^u = 2\boldsymbol{v}_2^u - n\mathbf{1}_m$.

Determine $\{v_1^l, v_1^u\}$ of CIB-1. As shown in Figure 1, we firstly calculate the vector $\mathbf{r}_1$ with $\mathbf{r}_1(j) = |\mathbf{Y}_{tr}(:, j) == 1|$. Then a histogram $\mathbf{h}$ is plotted, with x-coordinate being the number of positive labels and y-coordinate being the number of corresponding instances. Denoting the 5^{th} and 95^{th} percentiles of $\mathbf{h}$ as $\mathbf{h}_{0.05}$ and $\mathbf{h}_{0.95}$ respectively, we define the lower and upper bounds as: $v_1^l = \max\{1, \mathbf{h}_{0.05} \times \min\{1 + \varepsilon/3, 1.2\}\}$ and $v_1^u = \min\{0.8m, \mathbf{h}_{0.95} \times \min\{1 + \varepsilon, 1.5\}\}$. When $\varepsilon = 0$, we enforce that the number of positive labels of each instance to be in $[\mathbf{h}_{0.05}, \mathbf{h}_{0.95}]$. Although there is a range bias for about 10% of the instances, the rest will have satisfactory prediction. When $\varepsilon > 0$, both $\mathbf{h}_{0.05}$ and $\mathbf{h}_{0.95}$ will be smaller than their ground-truth values (i.e., in the case of $\varepsilon = 0$). Thus, we amplify them with a reasonable rate according to ε. In experiments, we observe that $\mathbf{h}_{0.05}$ varies in a very small range as ε increases. The reason is $\mathbf{h}_{0.05}^{\varepsilon=0}$ is always a small value in most experiments, and the range $[1, \mathbf{h}_{0.05}^{\varepsilon=0}]$ to which $\mathbf{h}_{0.05}^{\varepsilon>0}$ belongs is very narrow. Thus we increase $\mathbf{h}_{0.05}^{\varepsilon>0}$ with a small rate, by multiplying $\min\{1 + \varepsilon/3, 1.2\}$. Compared to $\mathbf{h}_{0.05}$, the variation range of $\mathbf{h}_{0.95}$ becomes larger. But $\mathbf{h}_{0.95}$ still varies smoothly w.r.t. ε. Thus we adjust it by multiplying it by $\min\{1 + \varepsilon, 1.5\}$.

Determine $\{\boldsymbol{v}_2^l, \boldsymbol{v}_2^u\}$ of CIB-2. As shown in Figure 1, we calculate $\mathbf{r}_2$ with $\mathbf{r}_2(i) = |\mathbf{Y}_{tr}(i, :) == 1|$, which is the number of positive instances of class c_i. The corresponding number in the complete label matrix is estimated as $\hat{\mathbf{r}}_2(i) = \mathbf{r}_2(i) \times \frac{n}{n_{tr}} \times \frac{1}{1-\varepsilon}$. Then, the bounds are: $\boldsymbol{v}_2^l(i) = \max\{1, \hat{\mathbf{r}}_2(i) \times \zeta_1\}$ and $\boldsymbol{v}_2^u(i) = \min\{0.8n, \hat{\mathbf{r}}_2(i) \times \zeta_2\}$, where we choose ζ_1 and ζ_2 from the sets $\{0.8, 1.2, 1.5\}$ and $\{1.5, 2, 3\}$ respectively.

Objective Function

Combining the previous terms, we formulate MLML as a discrete optimization problem with linear constraints,

$$\begin{aligned} &\min_{\mathbf{Z}} \ \text{tr}(\mathbf{Y}_P^\top (\mathbf{Y} - \mathbf{Z})) + \beta \text{tr}(\mathbf{Z}\mathbf{L}_X\mathbf{Z}^\top) + \gamma \text{tr}(\mathbf{Z}^\top \mathbf{L}_C \mathbf{Z}), \\ &\text{s.t. } \mathbf{Z} \in \{-1, 1\}, \overline{v}_1^l \mathbf{1}_n^\top \leq \mathbf{1}_m^\top \mathbf{Z} \leq \overline{v}_1^u \mathbf{1}_n^\top, \overline{\boldsymbol{v}}_2^l \leq \mathbf{Z}\mathbf{1}_n \leq \overline{\boldsymbol{v}}_2^u. \end{aligned} \quad (4)$$

β and γ control the trade-off between label consistency and label smoothness. They can be tuned by cross validation. For clarity, we denote the objective function of (4) as $F(\mathbf{Z})$, and define the constraint space as $\Omega_0 = \{\mathbf{Z} | \overline{v}_1^l \mathbf{1}_n^\top \leq \mathbf{1}_m^\top \mathbf{Z} \leq \overline{v}_1^u \mathbf{1}_n^\top, \overline{\boldsymbol{v}}_2^l \leq \mathbf{Z}\mathbf{1}_n \leq \overline{\boldsymbol{v}}_2^u\}, \Omega_1 = \{-1, 1\} \cap \Omega_0$. Problem (4) is simplified as $\min_{\mathbf{Z} \in \Omega_1} F(\mathbf{Z})$.

Proposition 1 *Problem (4) is a constrained submodular minimization (CSM) problem, and it is NP-hard*[1].

Proposition 2 *The Lovasz extension of objective function (4) is formulated as follows:*

$$f(\mathbf{Z}) = -\operatorname{tr}(\mathbf{Y}_P^\top \mathbf{Z}) + \frac{1}{2}\sum_{k}^{m}\sum_{i,j}^{n} \widehat{\mathbf{W}}_X(i,j)|\mathbf{Z}_{ki} - \mathbf{Z}_{kj}|$$
$$+ \frac{1}{2}\sum_{k}^{n}\sum_{i,j}^{m} \widehat{\mathbf{W}}_C(i,j)|\mathbf{Z}_{ik} - \mathbf{Z}_{jk}| + \mathit{const}, \tag{5}$$

where $\widehat{\mathbf{W}}_X(i,j) = 2\beta \mathbf{W}_X(i,j)\sqrt{\mathbf{d}_X(i)\mathbf{d}_X(j)}$ *and* $\widehat{\mathbf{W}}_C(i,j) = 2\gamma \mathbf{W}_C(i,j)\sqrt{\mathbf{d}_C(i)\mathbf{d}_X(j)}$. *const* $= \operatorname{tr}(\mathbf{Y}_P^\top \mathbf{Y}) + \frac{\beta}{2}\sum_k^m \sum_{i,j}^n [\mathbf{d}_X^{-\frac{1}{2}}(i) - \mathbf{d}_X^{-\frac{1}{2}}(j)]^2 + +\frac{\gamma}{2}\sum_k^n \sum_{i,j}^m [\mathbf{d}_C^{-\frac{1}{2}}(i) - \mathbf{d}_C^{-\frac{1}{2}}(j)]^2$.

Proposition 3 *$F(\mathbf{Z})$ and $f(\mathbf{Z})$ satisfy the conditions:*

$$\min_{\mathbf{Z}\in\Omega_1} f(\mathbf{Z}) \geq \min_{\mathbf{Z}\in\Omega_1} F(\mathbf{Z}) \geq \min_{\mathbf{Z}\in[-1,1]} f(\mathbf{Z}) = \min_{\mathbf{Z}\in\{-1,1\}} F(\mathbf{Z}),$$
$$\min_{\mathbf{Z}\in\Omega_1} f(\mathbf{Z}) \geq \min_{\mathbf{Z}\in\Omega_2} f(\mathbf{Z}) \geq \min_{\mathbf{Z}\in[-1,1]} f(\mathbf{Z}) = \min_{\mathbf{Z}\in\{-1,1\}} F(\mathbf{Z}).$$

A Convex Approximation to (4)

We will focus on three different approximation methods to solve (4). The first one is to simply drop the cardinality constraints, thus, becoming a submodular optimization problem $\min_{\mathbf{Z}\in\{-1,1\}} F(\mathbf{Z})$, which can be exactly solved using the st-cut algorithm (Boykov and Kolmogorov 2004). The second approximation is a LP relaxation of the first model based on Lovasz extension of submodular functions, as $\min_{\mathbf{Z}\in[-1,1]} f(\mathbf{Z})$ (see Proposition 2). The third method adds the cardinality constraints to the second one. For subsequent description, the corresponding algorithms of the second and third methods are referred to as MMIB-0 and MMIB, respectively.

Utilizing Proposition 2, we relax (4) to a continuous one:

$$\min_{\mathbf{Z}} \ f(\mathbf{Z}), \ \text{ s.t. } \mathbf{Z} \in \Omega_2 = [-1,1]^{m\times n} \cap \Omega_0. \tag{6}$$

As shown in Proposition 3, the original discrete problem (4) and its approximated continuous problem (6) share the same lower and upper bounds.

4 ADMM Solution to (6)

To handle the ℓ_1 terms in Problem (6), we introduce two auxiliary variables $\mathbf{U}_X$ and $\mathbf{U}_C$, as follows:

$$\min_{\mathbf{Z}\in\Omega_2, \mathbf{U}_X\geq 0, \mathbf{U}_C \geq 0} \operatorname{tr}(\mathbf{U}_X \overline{\mathbf{W}}_X^\top) + \operatorname{tr}(\mathbf{U}_C \overline{\mathbf{W}}_C^\top) - \operatorname{tr}(\mathbf{Y}_P^\top \mathbf{Z}),$$
$$\text{s.t.} \mathbf{Z}\mathbf{A} \leq \mathbf{U}_X, -\mathbf{Z}\mathbf{A} \leq \mathbf{U}_X, \mathbf{B}\mathbf{Z} \leq \mathbf{U}_C, -\mathbf{B}\mathbf{Z} \leq \mathbf{U}_C, \tag{7}$$

where $\mathbf{U}_X \in \mathrm{R}^{m\times n_{e_X}}$ with $\mathbf{U}_X(k,e) = |\mathbf{Z}_{ke_1} - \mathbf{Z}_{ke_2}|$. Two nodes e_1 and e_2 is connected by the edge e. $\overline{\mathbf{W}}_X = (\mathbf{w}_X, \ldots, \mathbf{w}_X)^\top \in \mathrm{R}^{m\times n_{e_X}}$ with $\mathbf{w}_X(e) = \widehat{\mathbf{W}}_X(e_1, e_2)$, $\forall e \in \mathcal{E}_X$. $\mathbf{U}_C \in \mathbb{R}^{n_{e_C}\times n}$ with $\mathbf{U}_C(e,k) = |\mathbf{Z}_{e_1 k} - \mathbf{Z}_{e_2 k}|$. $\overline{\mathbf{W}}_C = (\mathbf{w}_C, \ldots, \mathbf{w}_C) \in \mathbb{R}^{n_{e_C}\times n}$ with $\mathbf{w}_C(e) = \widehat{\mathbf{W}}_C(e_1, e_2)$, $\forall e \in \mathcal{E}_C$. $\mathbf{A} \in \{-1,1,0\}^{n\times n_{e_X}}$ with $\mathbf{A}(e_1, e) = 1, \mathbf{A}(e_2, e) = -1$ for $e \in \mathcal{E}_X$, while other entries being 0. $\mathbf{B} \in \{-1,1,0\}^{n_{e_C}\times m}$ with $\mathbf{B}(e, e_1) = 1, \mathbf{B}(e, e_2) = -1$ for $e \in \mathcal{E}_C$, while other entries being 0.

The LP problem (7) can be solved by many standard algorithms using off-the-shelf solvers. However, most existing solvers are designed for vector variables. Although we can vectorize (7), that will lead to a large LP problem that is inefficient to solve with standard solvers. Instead, we adopt the alternating direction method of multipliers (ADMM) (Boyd et al. 2011), which is known to have good convergence properties and can take advantage of special structural properties of our problem.

Following the procedure of the conventional ADMM, we firstly present the augmented Lagrange function of Problem (7), by introducing two slack variables $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$,

$$\mathcal{L}_{\rho_1,\rho_2}(\mathbf{Z}, \mathbf{U}_X, \mathbf{U}_C, \mathbf{\Phi}_1, \mathbf{\Phi}_2, \mathbf{\Lambda}_1, \mathbf{\Lambda}_2) = -\operatorname{tr}(\mathbf{Y}_P^\top \mathbf{Z}) + \tag{8}$$
$$\operatorname{tr}(\overline{\mathbf{W}}_X^\top \mathbf{U}_X) + \operatorname{tr}(\mathbf{U}_C \overline{\mathbf{W}}_C^\top) + \operatorname{tr}[\mathbf{\Lambda}_1^\top(\mathbf{Z}\overline{\mathbf{A}} - \mathbf{U}_X \overline{\mathbf{C}}$$
$$- \overline{\mathbf{G}}_1 + \mathbf{\Phi}_1)] + \operatorname{tr}[\mathbf{\Lambda}_2^\top(\overline{\mathbf{B}}\mathbf{Z} - \overline{\mathbf{D}}\mathbf{U}_C - \overline{\mathbf{G}}_2 + \mathbf{\Phi}_2)] +$$
$$\frac{\rho_1}{2}||\mathbf{Z}\overline{\mathbf{A}} - \mathbf{U}_X\overline{\mathbf{C}} - \overline{\mathbf{G}}_1 + \mathbf{\Phi}_1||_F^2 + \frac{\rho_2}{2}||\overline{\mathbf{B}}\mathbf{Z} - \overline{\mathbf{D}}\mathbf{U}_C$$
$$- \overline{\mathbf{G}}_2 + \mathbf{\Phi}_2||_F^2,$$

where $\overline{\mathbf{A}} = [\mathbf{A}, -\mathbf{A}, -\mathbf{1}_n, \mathbf{1}_n] \in \{-1,+1,0\}^{n\times 2(n_{e_X}+1)}$, $\overline{\mathbf{C}} = [\mathbf{I}, \mathbf{I}, \mathbf{0}_n, \mathbf{0}_n] \in \{0,1\}^{n_{e_X}\times 2(n_{e_X}+1)}$, $\overline{\mathbf{G}}_1 = [\mathbf{0}_{m\times 2n_{e_X}}, -\boldsymbol{\overline{v}}_2^l, \boldsymbol{\overline{v}}_2^u] \in \mathbb{R}^{m\times 2(n_{e_X}+1)}$, and $\mathbf{\Phi}_1 \in \mathbb{R}_+^{m\times 2(n_{e_X}+1)}$. $\overline{\mathbf{B}} = [\mathbf{B}; -\mathbf{B}; -\mathbf{1}_m^\top; \mathbf{1}_m^\top] \in \mathbb{R}^{(2n_{e_C}+2)\times m}$, $\overline{\mathbf{D}} = [\mathbf{I}; \mathbf{I}; \mathbf{0}_{2\times n_{e_c}}] \in \mathbb{R}^{(2n_{e_C}+2)\times n_{e_C}}$, $\overline{\mathbf{G}}_2 = [\mathbf{0}_{2n_{e_C}\times n}; -\overline{v}_1^l \mathbf{1}_n^\top; \overline{v}_1^u \mathbf{1}_n^\top] \in \mathbb{R}^{(2n_{e_C}+2)\times n}$, and $\mathbf{\Phi}_2 \in \mathbb{R}_+^{(2n_{e_C}+2)\times n}$. $\mathbf{\Lambda}_1 \in \mathbb{R}^{m\times 2(n_{e_X}+1)}$ and $\mathbf{\Lambda}_2 \in \mathbb{R}^{(2n_{e_C}+2)\times n}$ denote two Lagrangian parameter matrices, while $\rho_1, \rho_2 > 0$ indicate the trade-off parameters of two augmented terms. Based on (8), Problem (7) could be solved by following iterative updates:

$$\mathbf{Z}^{t+1} = \arg\min_{\mathbf{Z}\in[-1,1]} \operatorname{tr}(\mathbf{M}_0^\top \mathbf{Z}) + \frac{\rho_1}{2}\operatorname{tr}(\mathbf{Z}\mathbf{M}_1\mathbf{Z}^\top) \tag{9}$$
$$+ \frac{\rho_2}{2}\operatorname{tr}(\mathbf{Z}^\top \mathbf{M}_2 \mathbf{Z}),$$

$$\mathbf{U}_X^{t+1} = \max(\mathbf{0}, -\frac{1}{2\rho_1}\mathbf{M}_X), \tag{10}$$

$$\mathbf{U}_C^{t+1} = \max(\mathbf{0}, -\frac{1}{2\rho_2}\mathbf{M}_C), \tag{11}$$

$$\mathbf{\Phi}_1^{t+1} = \max(\mathbf{0}, \mathbf{U}_X^{t+1}\overline{\mathbf{C}} + \overline{\mathbf{G}}_1 - \frac{1}{\rho_1}\mathbf{\Lambda}_1^t - \mathbf{Z}^{t+1}\overline{\mathbf{A}}), \tag{12}$$

$$\mathbf{\Phi}_2^{t+1} = \max(\mathbf{0}, \overline{\mathbf{D}}\mathbf{U}_C^{t+1} + \overline{\mathbf{G}}_2 - \frac{1}{\rho_2}\mathbf{\Lambda}_2^t - \overline{\mathbf{B}}\mathbf{Z}^{t+1}), \tag{13}$$

$$\mathbf{\Lambda}_1^{t+1} = \mathbf{\Lambda}_1^t + \rho_1[\mathbf{Z}^{t+1}\overline{\mathbf{A}} - \mathbf{U}_X^{t+1}\overline{\mathbf{C}} - \overline{\mathbf{G}}_1 + \mathbf{\Phi}_1^{t+1}], \tag{14}$$

$$\mathbf{\Lambda}_2^{t+1} = \mathbf{\Lambda}_2^t + \rho_2[\overline{\mathbf{B}}\mathbf{Z}^{t+1} - \overline{\mathbf{D}}\mathbf{U}_C^{t+1} - \overline{\mathbf{G}}_2 + \mathbf{\Phi}_2^{t+1}], \tag{15}$$

[1] We refer the readers about the proofs of all propositions to *https://sites.google.com/site/baoyuanwu2015/Publications.*

Algorithm 1 ADMM algorithm for Problem (7)

```
Input: ADMM parameters and $\{\mathbf{Z}^0, \mathbf{U}_X^0, \mathbf{U}_C^0, \mathbf{\Phi}_1^0, \mathbf{\Phi}_2^0, \mathbf{\Lambda}_1^0, \mathbf{\Lambda}_2^0, \mathbf{y}_2^0\}$
Output: $\mathbf{Z}^*$
while not converged do
    update $\mathbf{Z}^{t+1}$ by solving Eq (9)
    update $(\mathbf{U}_X^{t+1}, \mathbf{U}_C^{t+1})$ using Eq (10) and (11)
    update $(\mathbf{\Phi}_1^{t+1}, \mathbf{\Phi}_2^{t+1})$ according to Eq (12) and (13)
    update $(\mathbf{\Lambda}_1^{t+1}, \mathbf{\Lambda}_2^{t+1})$ according to Eq (14) and (15)
end while
$\mathbf{Z}^* = sign(\mathbf{Z}^{t+1})$.
```

where $\mathbf{M}_0 = -\mathbf{Y}_P + (\mathbf{\Lambda}_1^t - \rho_1(\mathbf{U}_X^t \overline{\mathbf{C}} + \overline{\mathbf{G}}_1 - \mathbf{\Phi}_1^t))\overline{\mathbf{A}}^\top + \overline{\mathbf{B}}^\top[\mathbf{\Lambda}_2^t - \rho_2(\overline{\mathbf{D}}\mathbf{U}_C^t + \overline{\mathbf{G}}_2 - \mathbf{\Phi}_2^t)], \mathbf{M}_1 = \overline{\mathbf{A}\mathbf{A}}^\top, \mathbf{M}_2 = \overline{\mathbf{B}}^\top\overline{\mathbf{B}}$. $\mathbf{M}_X = \overline{\mathbf{W}}_X - [\mathbf{\Lambda}_1^t + \rho_1(\mathbf{Z}^{t+1}\overline{\mathbf{A}} - \overline{\mathbf{G}}_1 + \mathbf{\Phi}_1^t)]\overline{\mathbf{C}}^\top \in \mathbb{R}^{m \times n_{e_X}}$ and $\mathbf{M}_C = \overline{\mathbf{W}}_C - \mathbf{D}^\top[\mathbf{\Lambda}_2^t + \rho_2(\overline{\mathbf{B}}\mathbf{Z}^{t+1} - \overline{\mathbf{G}}_2 + \mathbf{\Phi}_2^t)] \in \mathbb{R}^{n_{e_C} \times n}$. In above updates, only the update of $\mathbf{Z}^{t+1}$ is efficiently solved by projected gradient descent algorithm (PGD) (Boyd and Vandenberghe 2004) with exact line search method, while others have closed-form solution. As only matrix multiplication operations involved in ADMM, its computational complexity is about $\mathcal{O}(T(amn^2 + bm^2n + cmn))$, where T is the iterations ($T < 100$ in our experiments), and a, b, c are small scalars. The global convergence of ADMM for convex problem has been proved in (Boyd et al. 2011; Ghadimi et al. 2013; Raghunathan and Di Cairano 2014). Algorithm 1 briefly summarizes above steps.

5 Experiments

Experimental Setup

Datasets and missing labels. Five benchmark multi-label datasets are used, namely Emotions (Trohidis et al. 2008), Scene (Boutell et al. 2004) Yeast (Elisseeff and Weston 2001), CAL500 (Turnbull et al. 2008) and Corel5k (Duygulu et al. 2002). The first four are downloaded from the 'Mulan' website[2], while Corel5k is downloaded from (Guillaumin et al. 2009)[3]. The first three datasets are provided with fixed training and testing partitions, and the testing results are reported, while CAL500 and Corel5k are evaluated using 5-fold cross validation. In Corel5k, some classes correspond to very few positive instances and some instances are assigned to very few positive classes. To avoid null classes (no positive instances exist) and null instances (no positive classes exist) when missing labels exist, we delete these rare classes and instances. The data details are summarized in Table 1, where $stats(\mathbf{r}_1) = [\min(\mathbf{r}_1), \text{median}(\mathbf{r}_1), \max(\mathbf{r}_1), \text{std}(\mathbf{r}_1)]$ denotes the basic statistics of $\mathbf{r}_1$, and $stats(\mathbf{r}_2) = [\min(\mathbf{r}_2), \text{median}(\mathbf{r}_2), \max(\mathbf{r}_2), \text{std}(\mathbf{r}_2)]$ denotes the basic statistics of $\mathbf{r}_2$. Obviously both CIB-1 and CIB-2 exist in all datasets and the more scattered distribution indicates the larger impact of CIB. Moreover, we create $\mathbf{Y}_{tr}$ by varying ε from 0% to 80%. In each case, the missing labels are randomly chosen and this process is conducted 5 times to obtain different missing labels.

Table 1: *Data statistics (symbols defined in the text)*

dataset	n	m	d	$\eta\%$	$stats(\mathbf{r}_1)$	$stats(\mathbf{r}_2)$
Emotions	593	6	72	31.16	[1, 2, 3, 0.67]	[148, 170, 264, 41]
Scene	2407	6	294	17.9	[1, 1, 3, 0.26]	[364, 429, 533, 56.81]
Yeast	2417	14	103	30.21	[13, 26, 48, 5.7]	[34, 660, 1816, 546.5]
CAL500	502	174	68	15	[1, 4, 11, 1.58]	[5, 39, 444, 81.17]
Corel5k	4211	62	1000	4.54	[2, 3, 5, 0.73]	[55, 117, 1059, 203.5]

Compared methods. As mentioned earlier, our proposed model can be approximated by three optimization problems, which are solved by st-cut, MMIB-0 and MMIB. We compare these methods in our experiments. We obtain st-cut from a publicly available MATLAB toolbox BK_matlab[4] and we implement MMIB using MATLAB. Several state-of-the-art mutli-label methods that can handle missing labels are compared, including MLR-GL (Bucak, Jin, and Jain 2011), MC-Pos (Cabral et al. 2011), FastTag (Chen, Zheng, and Weinberger 2013), MLML-exact (Wu et al. 2014) and LEML (Yu et al. 2014). We also compare with COCOA (Zhang, Li, and Liu 2015), which is the latest class-imbalance-aware multi-label method. Note that we use the MATLAB code made available for all the aforementioned methods. The binary weighted SVM (W-SVM) is also trained using the LIBSVM toolbox (Chang and Lin 2011), as a baseline classifier, which only trains on labeled instances of each class separately. The weight of each class is set to be the ratio between the number of negative and positive instances in this class. The predicted continuous labels of above methods are finally rounded to discrete labels by setting the threshold as their middle values (0 or $\frac{1}{2}$), while COCOA sets the threshold by maximizing F_1 score.

Evaluation metrics. Three widely used metrics, example-based F_1, $F_{1-macro}$ and $F_{1-micro}$, are adopted to evaluate the quality of the predicted label matrix from different perspectives. Their formal definitions can be found in (Sorower 2010). In our experiments, we observe that no single metric is enough to reflect prediction quality. Thus, we also define a new metric: $F_{1-mean} = \frac{1}{3}(F_1 + F_{1-macro} + F_{1-micro})$.

Comparison on Handling Missing Labels

Figure 2 presents the results of all compared methods on handling missing labels.

MMIB-0 v.s. **st-cut**. According to Proposition 3, their objective functions will be equal at the global optimum. However, their solutions tend to be very different. When $\varepsilon = 0\%$, st-cut gives good results on the first three datasets, where the CIB degrees are not very high. On CAL500 and Corel, due to the high CIB, st-cut shows poor performance. This demonstrates the performance of st-cut is significantly influenced by CIB, and it is likely to give a poor solution when the CIB degree is large. Moreover, when $\varepsilon > 0$, the performance of st-cut degrades sharply, even leading to degenerate

[2] http://mulan.sourceforge.net/datasets-mlc.html

[3] http://lear.inrialpes.fr/people/guillaumin/data.php

[4] http://vision.csd.uwo.ca/code/

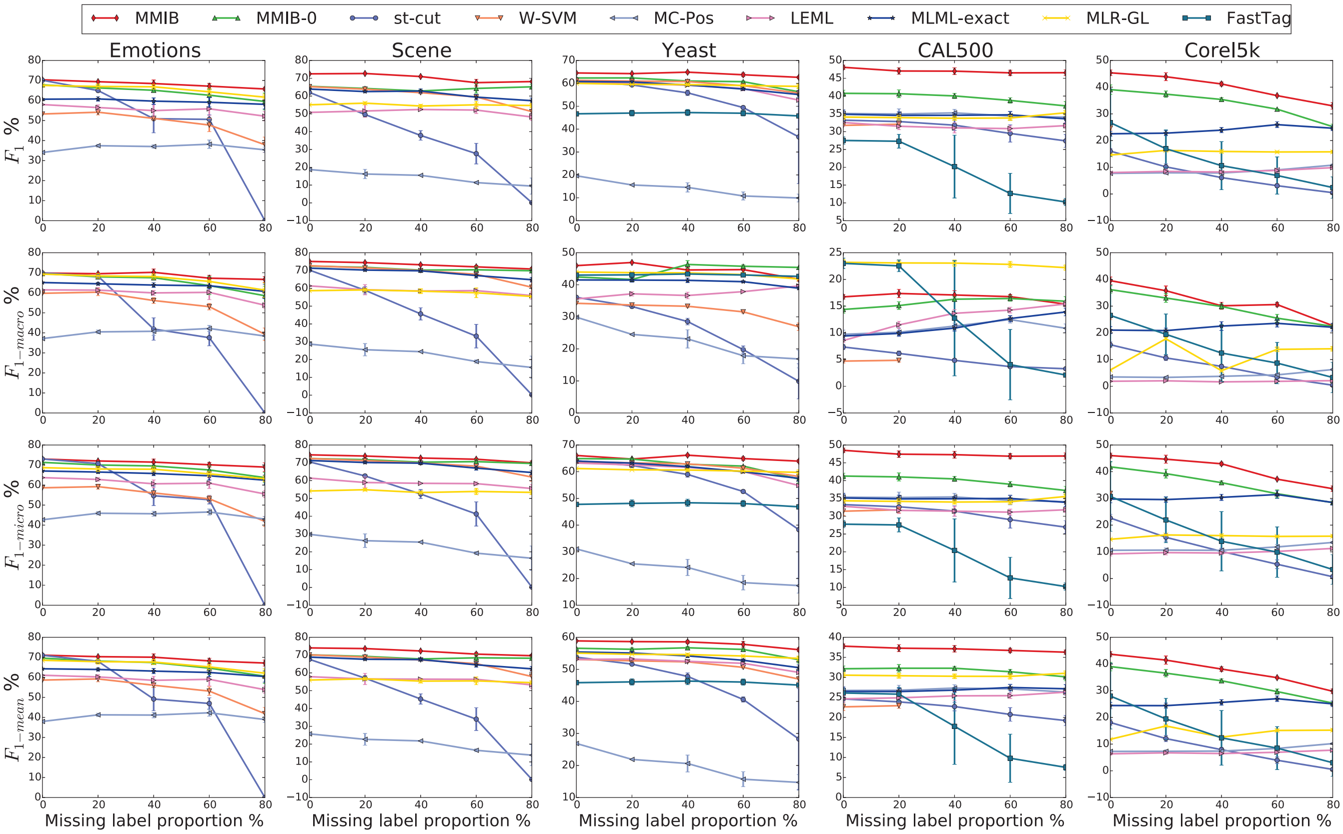


Figure 2: Results on all datasets with different missing label proportions.

results when ε is large. This reveals that when missing labels exist, the impact of CIB increased even more, as described in Introduction. Although MMIB-0 is also influenced by missing labels and CIB, MMIB-0 presents significant improvements over st-cut in most cases, especially when ε is large. The main reason is MMIB-0 can search for the solution in the continuous convex space, rather than in the discrete space as done by st-cut, thus, it is unlikely to get stuck at solutions corresponding to very poor performance. This verifies the efficacy of the proposed approximation.

MMIB v.s. **MMIB-0**. The main difference between them is the class cardinality bounds. Comparing them will evaluate the contribution of the linear constraints that enforce these bounds. MMIB gives better results than MMIB-0 in all cases. Specifically, following the sequence $\varepsilon = [0, 20, 40, 60, 80]\%$, the improvements of F_{1-mean} values are: Emotions [1.49, 2.2, 2.65, 3.6, 6.54]%; Scene [3.90, 4.36, 4.50, 2.08, 1.41]%; Yeast [2.30, 2.40, 1.85, 1.59, 3.12]%; CAL500 [5.64, 5.01, 4.86, 5.32, 6.15]%; Corel5k [4.62, 4.89, 4.42, 5.21, 4.32]%. These results show that the bounds (embedded as linear constraints) are beneficial to the model performance.

MMIB v.s. **Others**. MMIB-0 gives consistent improvements over most compared methods in most datasets, while MMIB shows further improvements over MMIB-0. This is due to two main reasons. **(1)** From the model perspective, both class-level and instance-level label smoothness are used to propagate the label information among different classes and instances. Also, there is no label bias since missing labels are treated as 0. In contrast, other methods except MLML-exact ignore the correlations among different instances, and label bias exists in MLR-GL and FastTag. Note that MLML-exact adopts a similar model. However, there are significant differences between MMIB-0 and MLML-exact. The objective function of MMIB-0 is based on Lovasz extension, which leads to the same objective value as the original discrete objective function, while MLML-exact directly relaxes $\{-1, 1\}$ to $[-1, 1]$. The specific label weights are assigned to different classes and different instances in MMIB-0 through $\mathbf{Y}_P$, while uniform weights are adopted in MLML-exact. **(2)** The linear constraints enforcing class cardinality bounds play an important role in MMIB to alleviate the negative impact of CIB. In contrast, other methods ignore this unavoidable challenge, and when $\varepsilon > 0$, the impact of CIB is further observed. Since MLR-GL and FastTag treat missing labels as negative, the imbalance between positive and negative labels will be further amplified. Thus, it is not strange that some methods show very poor (even degenerate) results when $\varepsilon > 0$. Note that there is an ex-

Table 2: *Comparison of multi-label learning methods on handling CIB when* no *labels are missing. Metric 1, 2, 3, 4 indicate* $F_1, F_{1-macro}, F_{1-micro}, F_{1-mean}$ *respectively.*

dataset	Emotions				Scene				Yeast				CAL500			
metric	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
W-SVM	53.27	59.75	62.88	58.63	65.29	72.70	72.41	70.13	61.31	34.22	63.71	53.08	31.77 ± 1.15	4.71 ± 0.37	31.41 ± 1.21	22.63 ± 0.91
COCOA	62.05	65.38	67.37	64.93	66.89	72.89	72.19	70.66	61.83	42.61	64.08	56.17	33.63 ± 1.75	**17.35 ± 0.98**	37.56 ± 1.52	29.51 ± 1.42
MMIB	**70.35**	**69.85**	**73.00**	**71.07**	**72.48**	**75.12**	**74.48**	**74.03**	**64.54**	**45.98**	**66.08**	**58.87**	**48.08 ± 0.60**	16.74 ± 0.43	**48.51 ± 0.61**	**37.78 ± 0.55**

Table 3: *Sensitivity test of class bounds* (ζ_1, ζ_2) *on CAL500 data, evaluated by* F_{1-mean} *values(%). The satisfied results are highlighted in bold. Please see text for details.*

$\varepsilon \rightarrow$	0%					20%					40%					60%					80%				
$\zeta_1 \downarrow, \zeta_2 \rightarrow$	∞	3	2.5	2	1.5	∞	3	2.5	2	1.5	∞	3	2.5	2	1.5	∞	3	2.5	2	1.5	∞	3	2.5	2	1.5
0	32.02	31.75	31.41	30.92	30.09	32.46	32.06	31.87	31.56	30.57	30.48	30.48	30.48	30.48	30.48	31.13	32.48	32.26	31.97	29.76	30.24	**34.76**	**34.72**	**34.09**	30.35
0.8	32.02	31.75	31.41	30.92	30.09	32.46	32.06	31.87	31.57	30.57	31.97	32.14	32.03	31.76	30.49	31.17	32.48	32.31	31.99	29.74	30.40	**34.80**	**34.75**	**34.10**	30.31
1.2	**37.44**	**37.35**	**37.28**	**37.25**	**37.28**	**35.91**	**35.72**	**35.60**	**35.49**	**35.11**	31.97	**34.75**	**34.65**	**34.47**	33.54	33.82	**34.72**	**34.76**	**34.39**	32.41	31.95	**35.34**	**35.41**	**34.90**	31.30
1.5	**36.87**	**36.83**	**36.82**	**36.78**	N/A	**37.55**	**37.61**	**37.57**	**37.51**	N/A	**34.78**	**36.99**	**36.96**	**36.81**	N/A	**35.19**	**36.22**	**36.33**	**36.04**	N/A	32.16	**35.66**	**35.75**	**35.34**	N/A
2	34.79	34.73	34.70	N/A	N/A	**36.28**	**36.17**	**36.13**	N/A	N/A	**36.86**	**36.84**	**36.81**	N/A	N/A	**36.69**	**36.80**	**36.82**	N/A	N/A	**35.22**	**36.02**	**36.25**	N/A	N/A

ception in the CAL500 dataset, where the $F_{1-macro}$ values of MLR-GL and FastTag are higher than those of MMIB, but the values of the other metrics are much lower. On this dataset, we see that MLR-GL and FastTag actually give degenerate results (i.e. classes with relatively high rates of positive instances in training have predictions that are always positive in the test). Thus the F_1 scores for these classes are very high, leading to a relatively high $F_{1-macro}$ value. This demonstrates that separate F_1 metrics are not enough to reflect prediction quality properly and that the proposed F_{1-mean} is a more comprehensive metric.

Comparison on Handling Class Imbalance

The results on handling CIB are shown in Table 2. As COCOA cannot handle missing labels, we only compare them in the case of $\varepsilon = 0$. The data format of Corel5k does not satisfy the requirement of COCOA, thus it is not tested. COCOA gives better results than W-SVM, while MMIB shows significant improvements over COCOA, up to $[6.1, 3.4, 2.7, 8.3]\%$ in F_{1-mean} on the four datasets, respectively. We believe such an improvement is due to two main reasons. First, COCOA is built on base classifiers (C4.5 decision tree with undersampling), whose performance dictates that of COCOA. In contrast, the class cardinality bounds adopted in our model are independent of any classifier. Second, COCOA only considers CIB-1, while MMIB enforces both CIB-1 and CIB-2.

Sensitivity Analysis of Cardinality Bounds

The estimated class bounds $\{v_1^l, v_1^u, \boldsymbol{v}_2^l, \boldsymbol{v}_2^u\}$ based on $\mathbf{Y}_{tr}$ may not be exactly the ground-truth values. However, the proposed model is robust to large variations in these bounds. To verify this, we test bound sensitivity on the challenging dataset CAL500. When ε changes, v_1^l, v_1^u do not vary much, while $\boldsymbol{v}_2^l, \boldsymbol{v}_2^u$ may vary in a range. Thus, we focus on the sensitivity test of $\boldsymbol{v}_2^l, \boldsymbol{v}_2^u$. For simplicity, we fix $\boldsymbol{v}_2^l = 0, \boldsymbol{v}_2^u = m$, i.e., CIB-1 is not embedded. We choose the rates ζ_1 and ζ_2 from $\{0, 0.8, 1.2, 1.5, 2\}$ and $\{\infty, 3, 2.5, 2, 1.5\}$ respectively, with $\zeta_1 < \zeta_2$. Using each pair (ζ_1, ζ_2), we run our model several times to report the average F_{1-mean} value. The test results are summarized in Table 3. In every ε, our model can always give satisfactory results in a relatively wide and stable range, e.g., $\zeta_1 \in \{1.2, 1.5, 2\}$ and $\zeta_2 \in \{3, 2.5, 2\}$. Note that the pair $(\zeta_1 = 0, \zeta_2 = \infty)$ means there is no CIB-2 constraint. Compared to other pairs (ζ_1, ζ_2), embedding CIB-2 constraints leads to significant improvements in most cases.

6 Conclusion

In this work, we propose a unified method to jointly handle missing labels and class imbalance in multi-label learning. To handle missing labels, our method propagates label information from the labeled instances to the unlabeled ones using label consistency and smoothness. The class imbalance problem is solved by the introduction of cardinality bounds over each instance and each class. We provide efficient numerical algorithms that demonstrate the improved performance over state-of-the-art methods on benchmark datasets.

Moreover, more useful prior knowledge, such as semantic hierarchy and mutual exclusion, can be naturally incorporated into the proposed linear programming framework as linear constraints, without any changes of the current algorithm. This will be explored in our future work.

7 Acknowledgments

This work is supported supported by competitive research funding from King Abdullah University of Science and Technology (KAUST). The participation of Siwei Lyu in this work is partly supported by US National Science Foundation Research Grant (CCF-1319800) and National Science Foundation Early Faculty Career Development (CAREER) Award (IIS-0953373). We thank reviewers for their constructive comments.

References

Bi, W., and Kwok, J. T. 2014. Multilabel classification with label correlations and missing labels. In *AAAI*.

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

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3(1):1–122.

Boykov, Y., and Kolmogorov, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(9):1124–1137.

Bucak, S. S.; Jin, R.; and Jain, A. K. 2011. Multi-label learning with incomplete class assignments. In *CVPR*, 2801–2808. IEEE.

Cabral, R. S.; De la Torre, F.; Costeira, J. P.; and Bernardino, A. 2011. Matrix completion for multi-label image classification. In *NIPS*, 190–198.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):27.

Chen, G.; Song, Y.; Wang, F.; and Zhang, C. 2008. Semi-supervised multi-label learning by solving a sylvester equation. In *SIAM international conference on data mining*, 410–419.

Chen, Z.; Chen, M.; Weinberger, K. Q.; and Zhang, W. 2015. Marginalized denoising for link prediction and multi-label learning. In *AAAI*.

Chen, M.; Zheng, A.; and Weinberger, K. 2013. Fast image tagging. In *ICML*, 1274–1282.

Dembczynski, K.; Jachnik, A.; Kotlowski, W.; Waegeman, W.; and Hüllermeier, E. 2013. Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, 1130–1138.

Duygulu, P.; Barnard, K.; de Freitas, J. F.; and Forsyth, D. A. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*. Springer. 97–112.

Elisseeff, A., and Weston, J. 2001. A kernel method for multi-labelled classification. *NIPS* 14:681–687.

Ghadimi, E.; Teixeira, A.; Shames, I.; and Johansson, M. 2013. Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems.

Goldberg, A. B.; Zhu, X.; Recht, B.; Xu, J.-M.; and Nowak, R. D. 2010. Transduction with matrix completion: Three birds with one stone. In *NIPS*, 757–765.

Guillaumin, M.; Mensink, T.; Verbeek, J.; and Schmid, C. 2009. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 309–316.

Johnson, C. R. 1990. Matrix completion problems: a survey. In *Proceedings of Symposia in Applied Mathematics*, volume 40, 171–198.

Kapoor, A.; Viswanathan, R.; and Jain, P. 2012. Multilabel classification using bayesian compressed sensing. In *NIPS*, 2654–2662.

Li, X.; Zhao, F.; and Guo, Y. 2015. Conditional restricted boltzmann machines for multi-label learning with incomplete labels. In *AISTATS*, 635–643.

Petterson, J., and Caetano, T. S. 2010. Reverse multi-label learning. In *NIPS*, 1912–1920.

Raghunathan, A. U., and Di Cairano, S. 2014. Optimal step-size selection in alternating direction method of multipliers for convex quadratic programs and model predictive control,. In *Proceedings of Symposium on Mathematical Theory of Networks and Systems*, 807–814.

Sahare, M., and Gupta, H. 2012. A review of multi-class classification for imbalanced data. *International Journal of Advanced Computer Research* 2(3):160–164.

Sorower, M. S. 2010. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*.

Sun, Y.; Zhang, Y.; and Zhou, Z.-H. 2010. Multi-label learning with weak label. In *AAAI*, 593–598.

Trohidis, K.; Tsoumakas, G.; Kalliris, G.; and Vlahavas, I. P. 2008. Multi-label classification of music into emotions. In *ISMIR*, volume 8, 325–330.

Turnbull, D.; Barrington, L.; Torres, D.; and Lanckriet, G. 2008. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing* 16(2):467–476.

Vasisht, D.; Damianou, A.; Varma, M.; and Kapoor, A. 2014. Active learning for sparse bayesian multilabel classification. In *SIGKDD*, 472–481. ACM.

Wang, Q.; Shen, B.; Wang, S.; Li, L.; and Si, L. 2014. Binary codes embedding for fast image tagging with incomplete labels. In *ECCV*. Springer. 425–439.

Wang, Q.; Si, L.; and Zhang, D. 2014. Learning to hash with partial tags: Exploring correlation between tags and hashing bits for large scale image retrieval. In *ECCV*. Springer. 378–392.

Wu, B.; Liu, Z.; Wang, S.; Hu, B.-G.; and Ji, Q. 2014. Multi-label learning with missing labels. In *ICPR*.

Wu, B.; Lyu, S.; Hu, B.-G.; and Ji, Q. 2015. Multi-label learning with missing labels for image annotation and facial action unit recognition. *Pattern Recognition* 48(7):2279–2289.

Wu, B.; Lyu, S.; and Ghanem, B. 2015. ML-MG: Multi-label learning with missing labels using a mixed graph. In *ICCV*. IEEE.

Xu, M.; Jin, R.; and Zhou, Z.-H. 2013. Speedup matrix completion with side information: Application to multi-label learning. In *NIPS*, 2301–2309.

Yu, H.-F.; Jain, P.; Kar, P.; and Dhillon, I. S. 2014. Large-scale multi-label learning with missing labels. In *ICML*.

Zhang, X., and Hu, B.-G. 2014. A new strategy of cost-free learning in the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering* 26(12):2872–2885.

Zhang, M.-L.; Li, Y.-K.; and Liu, X.-Y. 2015. Towards class-imbalance aware multi-label learning. In *IJCAI*.

Constrained Submodular Minimization Towards Missing Labels and Class Imbalance in Multi-label Learning (Supplementary Material)

Baoyuan Wu
KAUST, Saudi Arabia

Siwei Lyu
SUNY-Albany, NY USA

Bernard Ghanem
KAUST, Saudi Arabia

Here we provide the detailed proofs for the Propositions in the main manuscript. For clarity, here we rewrite the original objective function:

$$\min_{\mathbf{Z}\in\{-1,1\}} \quad \mathrm{tr}(\mathbf{Y}_P^\top(\mathbf{Y}-\mathbf{Z})) + \beta\mathrm{tr}(\mathbf{Z}\mathbf{L}_X\mathbf{Z}^\top) + \gamma\mathrm{tr}(\mathbf{Z}^\top\mathbf{L}_C\mathbf{Z}),$$
$$\text{s.t.} \quad \overline{v}_1^l\mathbf{1}_n^\top \leq \mathbf{1}_m^\top\mathbf{Z} \leq \overline{v}_1^u\mathbf{1}_n^\top, \overline{\boldsymbol{v}}_2^l \leq \mathbf{Z}\mathbf{1}_n \leq \overline{\boldsymbol{v}}_2^u. \tag{1}$$

The objective function is denoted as $F(\mathbf{Z})$, and the constraint space is defined as $\Omega_0 = \{\mathbf{Z}|\overline{v}_1^l\mathbf{1}_n^\top \leq \mathbf{1}_m^\top\mathbf{Z} \leq \overline{v}_1^u\mathbf{1}_n^\top, \overline{\boldsymbol{v}}_2^l \leq \mathbf{Z}\mathbf{1}_n \leq \overline{\boldsymbol{v}}_2^u\}$, and $\Omega_1 = \{-1,1\}\cap\Omega_0$.

1 Proof to Proposition 1

Before presenting the proof of Proposition 1, we firstly introduce some notations and Lemmas. The matrix variables can be transformed as follows:

$$\mathbf{z} = \mathrm{vec}(\mathbf{Z}) = [\mathbf{Z}_{11}, \ldots, \mathbf{Z}_{m1}, \ldots, \mathbf{Z}_{mn}]^\top \in \{-1,+1\}^{mn},$$
$$\mathbf{y} = \mathrm{vec}(\mathbf{Y}) = [\mathbf{Y}_{11}, \ldots, \mathbf{Y}_{m1}, \ldots, \mathbf{Y}_{mn}]^\top \in \{-1,0,+1\}^{mn},$$
$$\mathbf{y}_p = \mathrm{vec}(\mathbf{Y}_P) = [\mathbf{P}_{11}\mathbf{Y}_{11}, \ldots, \mathbf{P}_{mn}\mathbf{Y}_{mn}]^\top \in \mathrm{R}^{mn\times 1},$$
$$\mathbf{W} = \beta\cdot\mathbf{W}_X^\top\otimes\mathbf{I}_m + \gamma\cdot\mathbf{I}_n\otimes\mathbf{W}_C \in \mathrm{R}^{mn\times mn},$$
$$\mathbf{L} = \beta\cdot\mathbf{L}_X^\top\otimes\mathbf{I}_m + \gamma\cdot\mathbf{I}_n\otimes\mathbf{L}_C \in \mathrm{R}^{mn\times mn}, \tag{2}$$
$$\mathbf{H}_1 = [\mathbf{1}_m^\top, \mathbf{0}_{(n-1)m}^\top; \mathbf{0}_m^\top, \mathbf{1}_m^\top, \mathbf{0}_{(n-2)m}^\top; \cdots; \mathbf{0}_{(n-1)m}^\top, \mathbf{1}_m^\top]$$
$$\in \{0,1\}^{n\times mn},$$
$$\mathbf{H}_2 = [\mathbf{I}_m, \ldots, \mathbf{I}_m] \in \{0,1\}^{m\times mn},$$

where $\otimes$ denotes the Kronecker product (Zehfuss 1858). Then Problem (1) can be reformulated as follows:

$$\min_{\mathbf{z}\in\{-1,1\}^{mn}} \sum_i^{mn} \ell(z_i, y_i) + \frac{1}{2}\sum_{i,j}^{mn}\mathbf{W}(i,j)\Big[\frac{z_i}{\sqrt{\mathbf{d}_i}} - \frac{z_j}{\sqrt{\mathbf{d}_j}}\Big]^2$$
$$\equiv \min_{\mathbf{z}\in\{-1,1\}^{mn}} F(\mathbf{z}) = \mathbf{y}_p^\top(\mathbf{y}-\mathbf{z}) + \mathbf{z}^\top\mathbf{L}\mathbf{z}, \tag{3}$$
$$\text{s.t.} \quad [\overline{v}_1^l\mathbf{1}_n; \overline{\boldsymbol{v}}_2^l] \leq [\mathbf{H}_1; \mathbf{H}_2]\mathbf{z} \leq [\overline{v}_1^u\mathbf{1}_n; \overline{\boldsymbol{v}}_2^u].$$

Note that $\mathbf{W}$ can be understood as the similarity matrix of a large graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with mn nodes and $mn_{e_X} + nm_{e_C}$ edges. $\mathcal{G}$ consists of $\mathcal{G}_X$ and $\mathcal{G}_C$: first, treating each entry in $\mathbf{Z}$ as a node, then there are mn nodes; Second, we copy the edges among instances $\mathcal{E}_X$ for each class (building connections between the entries within the same row, for every row of $\mathbf{Z}$); last, we copy the edges among classes $\mathcal{E}_C$ for each instance (building connections between the entries within the same column, for every column of $\mathbf{Z}$). $\mathbf{d}_i \neq \sum_j^{mn}\mathbf{W}(i,j)$: if $(i,j)\in\mathcal{E}_X$, then $\mathbf{d}_i = \mathbf{d}_X(\hat{i})$, with $\hat{i}$ being the instance index (i.e., the column index of $\mathbf{Z}$) corresponding to the node i in $\mathcal{G}$; similarly, if $(i,j)\in\mathcal{E}_C$, then $\mathbf{d}_i = \mathbf{d}_C(\hat{i})$. That means we normalize $\mathbf{W}(i,j)$ by the sum of instance-level neighbors (in the same column) or class-level neighbors (in the same row), rather than the sum of all neighbors. As a result, this problem is a partially normalized graph-cut problem. Interestingly, the formulation in (3) is exactly the same as that of the standard GSSL problem (Zhu 2006). The only difference is that $\mathbf{L}$ is not a normalized Laplacian matrix in (3). Please see Lemma 1.

Lemma 1. $\mathbf{L}$ *matrix satisfies the following conditions:*

1. $\mathbf{L}$ *is not a normalized graph Laplacian matrix;*
2. *The off-diagonal entries of* $\mathbf{L}$ *are non-positive, i.e.,* $\forall i \neq j$, $\mathbf{L}(i,j) \leq 0$*;*
3. $\mathbf{L}$ *is positive semi-definite (PSD).*

Proof. 1. It is easy to know the diagonal entries of $\mathbf{L}$ are $\beta\mathbf{L}_X(i,i) + \gamma\mathbf{L}_C(j,j), i = 1,\ldots,n, j = 1,\ldots,m$. As both $\mathbf{L}_X$ and $\mathbf{L}_C$ are normalized Laplacian matrix, then $\mathbf{L}_X(i,i) = \mathbf{L}_C(j,j) = 1$, such that $\mathbf{L}(r,r) = \beta+\gamma, r = 1,\ldots,mn$. Since β and γ are two user-defined parameters, their summation is not always equivalent to 1. Thus $\mathbf{L}$ is not a normalized graph Laplacian matrix.

2. As both $\mathbf{L}_X$ and $\mathbf{L}_C$ are normalized Laplacian matrix, all of their off-diagonal entries are non-positive. According to the definition of Kronecker product, we know the off-diagonal entries of both $\beta\cdot\mathbf{L}_X^\top\otimes\mathbf{I}_m$ and $\gamma\cdot\mathbf{I}_n\otimes\mathbf{L}_C$ are non-positive. Thus $\forall i \neq j$, $\mathbf{L}(i,j) \leq 0$ holds.

3. Given two square matrix $\mathbf{S}_1 \in \mathrm{R}^{m\times m}$ and $\mathbf{S}_2 \in \mathrm{R}^{n\times n}$, their eigenvalues are denoted as $\lambda_1,\ldots,\lambda_m$ and $\mu_1,\ldots,\mu_n$. According to the property of Kronecker product, the eigenvalues of $\mathbf{S}_1\otimes\mathbf{S}_2$ are $\lambda_i\mu_j, i = 1,\ldots,m; j = 1,\ldots,n$. In Equation (2), for the first term, $\mathbf{L}_X^\top$ is PSD and I_m is positive definite (PD). Obviously all eigenvalues of $\mathbf{L}_X^\top\otimes\mathbf{I}_m$ are non-zero values, such that $\mathbf{L}_X^\top\otimes\mathbf{I}_m$ is a PSD matrix. Similarly we can obtain that

$\mathbf{I}_n \otimes \mathbf{L}_C$ is also PSD. Finally, as $\mathbf{L}$ is the positive weighted linear combination of two PSD matrices, it is easy to conclude that $\mathbf{L}$ is a PSD matrix.

□

Lemma 2. *Let $Q \in \mathrm{R}^{p\times p}$ and $q \in \mathrm{R}^p$, then the quadratic set function $F(A) = q^\top \mathbf{1}_A + \frac{1}{2}\mathbf{1}_A^\top Q \mathbf{1}_A$ is submodular if and only if the off-diagonal entries of Q are non-positive. $\mathbf{1}_A \in \{0,1\}^p$ denotes the indicator vector of the subset A: if $i \in A$, then $\mathbf{1}_A(i) = 1$, otherwise $\mathbf{1}_A(i) = 0$. (See Proposition 6.3 in (Bach 2013)).*

Lemma 3. *The objective function (3) is equivalent to a submodular set function $F : 2^V \to \mathrm{R}$, with $V = \{1, \dots, mn\}$.*

Proof. Given a subset $A \subset V$, it can be represented by the indicator vector $\mathbf{1}_A \in \{0,1\}^{mn}$. Obviously we know $\mathbf{z} = 2\mathbf{1}_A - 1$. Substitute it into (3), we obtain

$$\mathbf{y}_p^\top(\mathbf{y} - \mathbf{z}) + \mathbf{z}^\top \mathbf{L}\mathbf{z}$$
$$= (-2\mathbf{y}_p - 4\mathbf{L}\mathbf{1}_{mn})^\top \mathbf{1}_A + \frac{1}{2}\mathbf{1}_A^\top(8\mathbf{L})\mathbf{1}_A + \text{const} = F(A).$$

From Lemma 1, we know that the off-diagonal entries of $8\mathbf{L}$ are non-positive. Then according to Lemma 2, we conclude that the objective function (3) is equivalent to a submodular set function. □

Proposition 1. *Problem (1) is equivalent to a constrained submodular minimization (CSM) problem, and it is NP-hard*[1].

Proof. Lemma has demonstrated the objective function (3) is equivalent to a submodular function. And its constraint $[\overline{v}_1^l \mathbf{1}_n; \overline{\boldsymbol{v}}_2^l] \leq [\mathbf{H}_1; \mathbf{H}_2]\mathbf{z} \leq [\overline{v}_1^u \mathbf{1}_n; \overline{\boldsymbol{v}}_2^u]$ can be seen as the cardinality bounds on local parts of $\mathbf{z}$. Obviously (3) is also a cut function. As demonstrated in (Queyranne and Visitor 2002), if the objective function is a cut function and submodular, then its minimization with cardinality constraint is NP-hard. The local-part cardinality constraint in Problem (3) is tighter than the cardinality constraint of the whole vector $\mathbf{z}$. Thus we conclude that Problem (3) is NP-hard. As Problem (1) is equivalent to Problem (3), it is also NP-hard. □

2 Proof to Proposition 2

Proposition 2. *The Lovasz extension of objective function (1) is formulated as follows:*

$$f(\mathbf{Z}) = -\mathrm{tr}(\mathbf{Y}_P^\top \mathbf{Z}) + \frac{1}{2}\sum_k^m \sum_{i,j}^n \widehat{\mathbf{W}}_X(i,j)|\mathbf{Z}_{ki} - \mathbf{Z}_{kj}|$$
$$+ \frac{1}{2}\sum_k^n \sum_{i,j}^m \widehat{\mathbf{W}}_C(i,j)|\mathbf{Z}_{ik} - \mathbf{Z}_{jk}| + \mathit{const}, \tag{4}$$

where $\widehat{\mathbf{W}}_X(i,j) = 2\beta \mathbf{W}_X(i,j)(\mathbf{d}_X(i)\mathbf{d}_X(j))^{-\frac{1}{2}}$ *and* $\widehat{\mathbf{W}}_C(i,j) = 2\gamma \mathbf{W}_C(i,j)(\mathbf{d}_C(i)\mathbf{d}_X(j))^{-\frac{1}{2}}$. *const* $= \mathrm{tr}(\mathbf{Y}_P^\top \mathbf{Y}) + \frac{\beta}{2}\sum_k^m \sum_{i,j}^n [\mathbf{d}_X^{-\frac{1}{2}}(i) - \mathbf{d}_X^{-\frac{1}{2}}(j)]^2 + \frac{\gamma}{2}\sum_k^n \sum_{i,j}^m [\mathbf{d}_C^{-\frac{1}{2}}(i) - \mathbf{d}_C^{-\frac{1}{2}}(j)]^2$. *Note that here $\mathbf{Z}$ indicate continuous variables.*

Proof. We firstly prove that the Lovasz extension of the quadratic term $\mathbf{z}^\top \mathbf{L}\mathbf{z}$ of (3) is

$$\hat{f}(\mathbf{z}) = \frac{1}{2}\sum_{i,j}^{mn} \mathbf{W}_{ij}\big[2(\mathbf{d}_i\mathbf{d}_j)^{-\frac{1}{2}}|z_i - z_j| + (\mathbf{d}_i^{-\frac{1}{2}} - \mathbf{d}_j^{-\frac{1}{2}})^2\big]. \tag{5}$$

Define a set function $F : 2^V \to \mathrm{R}$ corresponding to the quadratic term in (3), we have

$$\hat{F}(A) = \frac{1}{2}\sum_{i,j}^{mn} \mathbf{W}_{ij}\big[\frac{z_i}{\sqrt{\mathbf{d}_i}} - \frac{z_j}{\sqrt{\mathbf{d}_j}}\big]^2, \tag{6}$$

where $A \subset V$, and $\mathbf{z}$ can be seen as the sign vector of A: when $i \in A$, then $z_i = 1$, otherwise $z_i = -1$. Note that the value of the null set is

$$\hat{F}(\emptyset) = \frac{1}{2}\sum_{i,j}^{mn} \mathbf{W}_{ij}\big[\frac{1}{\sqrt{\mathbf{d}_i}} - \frac{1}{\sqrt{\mathbf{d}_j}}\big]^2, \tag{7}$$

which is a constant of $\mathbf{z}$. To facilitate the following proof, we define a modified set function as follows:

$$\overline{F}(A) = \hat{F}(A) - \hat{F}(\emptyset) = \frac{1}{2}\sum_{i,j}^{mn} \overline{\mathbf{W}}_{ij}(z_i - z_j)^2 = \mathbf{z}^\top \overline{\mathbf{L}}\mathbf{z}, \tag{8}$$

where $\overline{\mathbf{W}}_{ij} = \frac{\mathbf{W}_{ij}}{\sqrt{\mathbf{d}_i\mathbf{d}_j}}$, and $\overline{\mathbf{L}} = \overline{\mathbf{D}} - \overline{\mathbf{W}}$ denotes the corresponding unnormalized Laplacian matrix. We have $\overline{F}(\emptyset) = \overline{F}(V) = 0$. Obviously it is a standard cut function. Thus, as demonstrated in Section 6.2 of (Bach 2013), the Lovasz extension of $\overline{F}$ is

$$\overline{f}(\mathbf{z}) = \sum_{i,j}^{mn} \overline{\mathbf{W}}_{ij}|z_i - z_j|. \tag{9}$$

Then adding the constant term $F(\emptyset)$, we obtain $\hat{f}(\mathbf{z}) = \overline{f}(\mathbf{z}) + F(\emptyset)$, i.e., Equation (5). Besides, the Lovasz extension of the linear term $\mathbf{y}_p^\top(\mathbf{y} - \mathbf{z})$ is still the same form. Thus the Lovasz extension of the objective function (3) is

$$f(\mathbf{z}) = \mathbf{y}_p^\top(\mathbf{y} - \mathbf{z}) + \hat{f}(\mathbf{z}). \tag{10}$$

Finally, utilizing the transformations between variables, it is easy to obtain (4). □

3 Proof to Proposition 3

Utilizing Proposition 2, we obtain the following optimization problem:

$$\min_{\mathbf{Z}\in\Omega_2} f(\mathbf{Z}), \tag{11}$$

where $\Omega_2 = [-1,1]^{m\times n} \cap \Omega_0$.

Lemma 4. *Let F be a submodular function and f its Lovasz extension; then we have*

$$\min_{A\subset V} F(A) = \min_{\mathbf{a}\in\{0,1\}^p} f(\mathbf{a}) = \min_{\mathbf{a}\in[0,1]^p} f(\mathbf{a}).$$

(See Proposition 3.7 in (Bach 2013)).

[1] Due to the space limit, the proofs of all propositions in this manuscript will be presented in **supplementary material**.

Proposition 3. *$F(\mathbf{Z})$ and $f(\mathbf{Z})$ satisfy the following conditions:*

$$\min_{\mathbf{Z}\in\Omega_1} f(\mathbf{Z}) \geq \min_{\mathbf{Z}\in\Omega_1} F(\mathbf{Z}) \geq \min_{\mathbf{Z}\in[-1,1]} f(\mathbf{Z}) = \min_{\mathbf{Z}\in\{-1,1\}} F(\mathbf{Z}),$$
$$\min_{\mathbf{Z}\in\Omega_1} f(\mathbf{Z}) \geq \min_{\mathbf{Z}\in\Omega_2} f(\mathbf{Z}) \geq \min_{\mathbf{Z}\in[-1,1]} f(\mathbf{Z}) = \min_{\mathbf{Z}\in\{-1,1\}} F(\mathbf{Z}).$$

Proof. According to Proposition 2 and Lemma 4, as well as a simple transformation from $[-1,1]$ to $[0,1]$, it is easy to obtain

$$\min_{\mathbf{Z}\in[-1,1]} f(\mathbf{Z}) = \min_{\mathbf{Z}\in\{-1,1\}} F(\mathbf{Z}). \tag{12}$$

□

As $\Omega_1 \subset \{-1,1\}$, we have

$$\min_{\mathbf{Z}\in\Omega_1} F(\mathbf{Z}) \geq \min_{\mathbf{Z}\in\{-1,1\}} F(\mathbf{Z}) = \min_{\mathbf{Z}\in[-1,1]} f(\mathbf{Z}).$$

As $\Omega_2 \subset [-1,1]$, we have

$$\min_{\mathbf{Z}\in\Omega_2} f(\mathbf{Z}) \geq \min_{\mathbf{Z}\in[-1,1]} f(\mathbf{Z}) = \min_{\mathbf{Z}\in\{-1,1\}} F(\mathbf{Z}),$$

As $\Omega_1 \subset \Omega_2$, we have

$$\min_{\mathbf{Z}\in\Omega_1} f(\mathbf{Z}) \geq \min_{\mathbf{Z}\in\Omega_2} f(\mathbf{Z}).$$

Then the only remaining proof is

$$\min_{\mathbf{Z}\in\Omega_1} f(\mathbf{Z}) \geq \min_{\mathbf{Z}\in\Omega_1} F(\mathbf{Z}). \tag{13}$$

We firstly prove the following inequality $f(\mathbf{Z}) = f(\mathbf{z}) \geq F(\mathbf{Z}) = F(\mathbf{z})$ holds for all $\mathbf{z} \in [-1,1]$, i.e.,

$$\begin{aligned} & f(\mathbf{z}) - F(\mathbf{z}) = \hat{f}(\mathbf{z}) - \mathbf{z}^\top L\mathbf{z} \qquad (14) \\ \equiv & \frac{1}{2}\sum_{i,j}^{mn} \mathbf{W}_{ij}\left[2t_it_j|z_i - z_j| + (t_i - t_j)^2 - (t_iz_i - t_jz_j)^2\right] \geq 0. \end{aligned}$$

We use $t_i = \mathbf{d}_i^{-\frac{1}{2}} > 0$ and $t_j = \mathbf{d}_j^{-\frac{1}{2}} > 0$ for clarity. When $z_i > z_j$, we have

$$\begin{aligned} & 2t_it_j|z_i - z_j| + (t_i - t_j)^2 - (t_iz_i - t_jz_j)^2 \qquad (15) \\ = & 2t_it_j(z_i - z_j) + (t_i - t_j)^2 - (t_iz_i - t_jz_j)^2 \\ = & 2t_it_j\left[\sqrt{(1 - z_i^2)(1 - z_j^2)} - (1 - z_i)(1 + z_j)\right] \\ & + \left[t_i\sqrt{1 - z_i^2} - t_j\sqrt{1 - z_j^2}\right]^2 \\ \geq & 2t_it_j\left[\sqrt{(1 - z_i^2)(1 - z_j^2)} - (1 - z_i)(1 + z_j)\right] \\ \geq & 2t_it_j\left[\sqrt{(1 - z_i)^2(1 + z_j)^2} - (1 - z_i)(1 + z_j)\right] \geq 0. \end{aligned}$$

It is easy to prove that when $z_i \leq z_j$, the inequality (15) still holds. Thus, considering $\mathbf{W}_{ij} \geq 0$, it is easy to know the inequality (14) holds $\forall \mathbf{z} \in [-1,+1]$. Furthermore, as $\Omega_1 \subset [-1,1]$, the last inequality (13) is proved. Thus all proofs are finished.

References

Bach, F. 2013. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning* 228.

Queyranne, M., and Visitor, I. 2002. An introduction to submodular functions and optimization.

Zehfuss, G. 1858. Über eine gewisse determinante. *Zeitschrift für Mathematik und Physik* 3:298–301.

Zhu, X. 2006. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison* 2:3.