Collaboration Based Multi-Label Learning

Lei Feng,[1,2] **Bo An,**[1] **Shuo He**[3]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]Alibaba-NTU Singapore Joint Research Institute, Singapore
[3]College of Computer and Information Science, Southwest University, Chongqing, China
{feng0093, boan}@ntu.edu.sg, hs8207083890@email.swu.edu.cn

Abstract

It is well-known that exploiting label correlations is crucially important to multi-label learning. Most of the existing approaches take label correlations as prior knowledge, which may not correctly characterize the real relationships among labels. Besides, label correlations are normally used to regularize the hypothesis space, while the final predictions are not explicitly correlated. In this paper, we suggest that *for each individual label, the final prediction involves the collaboration between its own prediction and the predictions of other labels*. Based on this assumption, we first propose a novel method to learn the label correlations via sparse reconstruction in the label space. Then, by seamlessly integrating the learned label correlations into model training, we propose a novel multi-label learning approach that aims to explicitly account for the correlated predictions of labels while training the desired model simultaneously. Extensive experimental results show that our approach outperforms the state-of-the-art counterparts.

Introduction

Multi-label learning deals with the problem where an instance can be associated with multiple labels simultaneously. Formally speaking, let $\mathcal{X} \in \mathbb{R}^d$ be d-dimensional feature space and $\mathcal{Y} = \{y_1, y_2, \cdots, y_q\}$ be the label space with q labels. Given the multi-label training set $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathcal{X}$ is a feature vector and $\mathbf{y}_i \in \{-1, 1\}^q$ is the label vector, the goal of multi-label learning is to learn a model $f : \mathbb{R}^d \to \{-1, 1\}^q$, which maps from the space of feature vectors to the space of label vectors. As a learning framework that handles objects with multiple semantics, multi-label learning has been widely applied in many real-world applications, such as image annotation (Yang et al. 2016), document categorization (Li, Ouyang, and Zhou 2015), bioinformatics (Zhang and Zhou 2006), and information retrieval (Gopal and Yang 2010).

The most straightforward multi-label learning approach (Boutell et al. 2004) is to decompose the problem into a set of independent binary classification tasks, one for each label. Although this strategy is easy to implement, it may result in degraded performance, due to the ignorance of correlations among labels. To compensate for this deficiency, the exploitation of label correlations has been widely accepted as a key component of effective multi-label learning approaches (Gibaja and Ventura 2015; Zhang and Zhou 2014).

So far, many methods have been developed to improve the performance of multi-label learning by exploring various types of label correlations (Tsoumakas et al. 2009; Cesa-Bianchi, Gentile, and Zaniboni 2006; Petterson and Caetano 2011; Huang, Zhou, and Zhou 2012; Huang, Yu, and Zhou 2012; Zhu, Kwok, and Zhou 2018). There has been increasing interest in exploiting the label correlations by taking the label correlation matrix as prior knowledge (Hariharan et al. 2010; Cai et al. 2013; Huang et al. 2016; 2018). Concretely, these methods directly calculate the label correlation matrix by the similarity between label vectors using common similarity measures, and then incorporate the label correlation matrix into model training for further enhancing the predictions of multiple label assignments. However, the label correlations are simply obtained by common similarity measures, which may not be able to reflect complex relationships among labels. Besides, these methods exploit label correlations by manipulating the hypothesis space, while the final predictions are not explicitly correlated.

To address the above limitations, we make a key assumption that *for each individual label, the final prediction involves the collaboration between its own prediction and the predictions of other labels*. Based on this assumption, a novel multi-label learning approach named CAMEL, i.e., CollAboration based Multi-labEl Learning, is proposed. Different from most of the existing approaches that calculate the label correlation matrix simply by common similarity measures, CAMEL presents a novel method to learn such matrix and show that it is equivalent to sparse reconstruction in the label space. The learned label correlation matrix is capable of reflecting the collaborative relationships among labels regarding the final predictions. Subsequently, CAMEL seamlessly incorporates the learned label correlations into the desired multi-label predictive model. Specifically, label-independent embedding is introduced, which aims to fit the final predictions with the learned label correlations while guiding the estimation of the model parameters simultaneously. The effectiveness of CAMEL is clearly demonstrated

by experimental results on a number of datasets.

Related Work

In recent years, many algorithms have been proposed to deal with multi-label learning tasks. In terms of the *order of label correlations* being considered, these approaches can be roughly categorized into three strategies (Zhang and Zhou 2014; Gibaja and Ventura 2015).

For the first-order strategy, the multi-label learning problem is tackled in a label-by-label manner where label correlations are ignored. Intuitively, one can easily decompose the multi-label learning problem into a series of independent binary classification problems (one for each label) (Boutell et al. 2004). The second-order strategy takes into consideration pairwise relationships between labels, such as the ranking between relevant labels and irrelevant labels (Elisseeff and Weston 2002) or the interaction of paired labels (Zhu et al. 2005). For the third-order strategy, high-order relationships among labels are considered. Following this strategy, numerous multi-label algorithms are proposed. For example, by modeling all other labels' influences on each label, a shared subspace (Ji et al. 2008) is extracted for model training. By addressing connections among random subsets of labels, a chain of binary classifiers (Read et al. 2011) are sequentially trained.

Recently, there has been increasing interest in second-order approaches (Hariharan et al. 2010; Cai et al. 2013; Huang et al. 2016; 2018) that take the label correlation matrix as prior knowledge for model training. These approaches normally directly calculate the label correlation matrix by the similarity between label vectors using common similarity measures, and then incorporate the label correlation matrix into model training for further enhancing the predictions of multiple label assignments. For instance, cosine similarity is widely used to calculate the label correlation matrix (Cai et al. 2013; Huang et al. 2016; 2018). Such label correlation matrix is further incorporated into a structured sparsity-inducing norm regularization (Cai et al. 2013) for regularizing the learning hypotheses, or performing joint label-specific feature selection and model training (Huang et al. 2016; 2018). In addition, there are also some high-order approaches that exploit label correlations on the hypothesis space, while they do not rely on the label correlation matrix. For example, a boosting approach (Huang, Yu, and Zhou 2012) is proposed to exploit label correlations with a hypothesis reuse mechanism.

Note that most of the existing approaches using label correlation matrix are second-order and focus on the hypothesis space. Such simple label correlations exploited in the hypothesis space may not correctly depict the real relationships among labels, and final predictions are not explicitly correlated. In the next section, a novel high-order approach with crafted label correlation matrix that focus on the label space will be introduced.

The CAMEL Approach

Following the notations used in Introduction, the training set can be alternatively represented by $\mathcal{D} = \{(\mathbf{X}, \mathbf{Y})\}$ where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times d}$ denotes the instance matrix, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]^\top \in \mathbb{R}^{n \times q}$ denotes the label matrix. In addition, we denote by $\mathbf{Y}_j \in \mathbb{R}^n$ the j-th column vector of the matrix $\mathbf{Y}$ (versus $\mathbf{y}_j \in \mathbb{R}^q$ for the j-th row vector of $\mathbf{Y}$), and $\mathbf{Y}_{-j} = [\mathbf{Y}_1, \cdots, \mathbf{Y}_{j-1}, \mathbf{Y}_{j+1}, \cdots, \mathbf{Y}_q] \in \mathbb{R}^{n \times (q-1)}$ represents the matrix that excludes the j-th column vector of $\mathbf{Y}$.

Label Correlation Learning

To characterize the collaborative relationships among labels regarding the final predictions, CAMEL works by learning a label correlation matrix $\mathbf{S} = [s_{ij}]_{q \times q}$ where s_{ij} reflects the contribution of the i-label to the j-label. Guided by the assumption that *for each individual label, the final prediction involves the collaboration between its own prediction and the predictions of other labels*, we thus take the given label matrix as the final prediction, and propose to learn the label correlation matrix $\mathbf{S}$ in the following way:

$$\min_{s_{ij}} \left\| ((1-\alpha)\mathbf{Y}_j + \alpha \sum_{i \neq j, i \in [q]} s_{ij} \mathbf{Y}_i) - \mathbf{Y}_j \right\|_2^2 \quad (1)$$

where α is the tradeoff parameter that controls the collaboration degree. In other words, α is used to balance the j-th label's own prediction and the predictions of other labels. Since each label is normally correlated with only a few labels, the collaborative relationships between one label and other labels could be sparse. With a slight abuse of notation, we denote by $\mathbf{S}_j = [s_{1j}, \cdots, s_{j-1,j}, s_{j+1,j}, \cdots, s_{qj}]^\top \in \mathbb{R}^{(q-1)}$ the j-th column vector of $\mathbf{S}$ excluding s_{jj} ($s_{jj} = 0$). Under canonical sparse representation, the coefficient vector $\mathbf{S}_j$ is learned by solving the following optimization problem:

$$\min_{\mathbf{S}_j} \|(1-\alpha)\mathbf{Y}_j + \alpha \mathbf{Y}_{-j}\mathbf{S}_j - \mathbf{Y}_j\|_2^2 + \hat{\lambda} \|\mathbf{S}_j\|_1 \quad (2)$$

where $\hat{\lambda}$ controls the sparsity of the coefficient vector $\mathbf{S}_j$. By properly rewriting the above problem and setting $\lambda = \hat{\lambda}/\alpha$, it is easy to derive the following equivalent optimization problem:

$$\min_{\mathbf{S}_j} \|\mathbf{Y}_{-j}\mathbf{S}_j - \mathbf{Y}_j\|_2^2 + \lambda \|\mathbf{S}_j\|_1 \quad (3)$$

Here, this problem aims to estimate the collaborative relationships between the j-th label and the other labels via sparse reconstruction. The first term corresponds to the linear reconstruction error via ℓ_2 norm, and the second term controls the sparsity of the reconstruction coefficients by using ℓ_1 norm. The relative importance of each term is balanced by the tradeoff parameter λ, which is empirically set to $\frac{1}{100} \left\| \mathbf{Y}_j^\top \mathbf{Y}_{-j} \right\|_\infty$ in the experiments. To solve problem (3), the popular Alternating Direction Method of Multiplier (ADMM) (Boyd et al. 2011) is employed, and detailed information is given in Appendix A. After solving problem (3) for each label, the weight matrix $\mathbf{S}$ can be accordingly constructed with all diagonal elements set to 0. Note that for most of the existing second-order approaches using label correlation matrix (Hariharan et al. 2010; Cai et

al. 2013; Huang et al. 2016; 2018), only pairwise relationships are considered, and the relationships between one label and the other labels are separated. While for CAMEL, since the final prediction of each label is determined by all the predictions of other labels and itself, the relationships among all labels are exploited in a collaborative manner. Which means, the relationships between one label and the other labels are coordinated (influenced by each other). Therefore, CAMEL is a high-order approach.

Multi-Label Classifier Training

In this section, we propose a novel multi-label learning approach by seamlessly integrating the learned label correlations into the desired predictive model. Suppose the ordinary prediction matrix of $\mathbf{X}$ is denoted by $f(\mathbf{X}) = [f_1(\mathbf{X}), f_2(\mathbf{X}), \cdots, f_q(\mathbf{X})] \in \mathbb{R}^{n\times q}$ where $f_1(\cdot), f_2(\cdot), \cdots, f_q(\cdot)$ denotes the individual label predictors respectively. In the ordinary setting, each label predictor is only in charge of a single label, while label correlations are fully lost. To absorb the learned label correlations into predictions, we reuse the assumption that *for each individual label, the final prediction involves the collaboration between its own prediction and the predictions of other labels*, and propose to compute the final prediction of the j-th label as follows:

$$(1-\alpha)f_j(\mathbf{X}) + \alpha \sum_{i\neq j, i\in[q]} s_{ij} f_i(\mathbf{X}) \tag{4}$$

where α is consistent with problem (1), which controls the collaboration degree of label predictions. By considering all the q label predictions simultaneously, we thus obtain the following compact representation:

$$(1-\alpha)f(\mathbf{X}) + \alpha f(\mathbf{X})\mathbf{S} = f(\mathbf{X})((1-\alpha)\mathbf{I} + \alpha\mathbf{S}) \tag{5}$$

Here, the whole multi-label learning problem could be considered as two parallel subproblems, i.e., training the ordinary model and fitting the final predictions by the modeling outputs with label correlations. Thus, we propose to learn label-independent embedding denoted by $\mathbf{Z} \in \mathbb{R}^{n\times q}$, which works as a bridge between model training and prediction fitting. This brings several advantages: First, the two subproblems can be solved via alternation, which encourages the mutual adaption of model training and prediction fitting; Second, the relative importance of the two subproblems can be controlled by a tradeoff parameter; Third, closed-form solutions and kernel extension can be easily derived. Let $\mathbf{G} = (1-\alpha)\mathbf{I} + \alpha\mathbf{S}$, the proposed formulation is given as follows:

$$\min_{\mathbf{Z}, f} \frac{1}{2}\|f(\mathbf{X}) - \mathbf{Z}\|_F^2 + \frac{\lambda_1}{2}\|\mathbf{ZG} - \mathbf{Y}\|_F^2 + \frac{\lambda_2}{2}\Omega(f) \tag{6}$$

where $\Omega(f)$ controls the complexity of the model f, λ_1 and λ_2 are the tradeoff parameters determining the relative importance of the above three terms. To instantiate the above formulation, we choose to train the widely-used model $f(\mathbf{X}) = \phi(\mathbf{X})\mathbf{W} + \mathbf{1}\mathbf{b}^\top$ where $\mathbf{W}$ and $\mathbf{b}$ are the model parameters, $\mathbf{1} = [1, \cdots, 1]^\top$ denotes the column vector with all elements equal to 1, and $\phi(\cdot)$ is a feature mapping that maps the feature space to some higher (maybe infinite) dimensional Hilbert space. For the regularization term to control the model complexity, we adopt the widely-used squared Frobenius norm, i.e., $\|\mathbf{W}\|_F^2$. To further facilitate a kernel extension for the general nonlinear case, we finally present the formulation as a constrained optimization problem:

$$\min_{\mathbf{W},\mathbf{Z},\mathbf{E},\mathbf{b}} \frac{1}{2}\|\mathbf{E}\|_F^2 + \frac{\lambda_1}{2}\|\mathbf{ZG} - \mathbf{Y}\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 \tag{7}$$
$$\text{s.t.} \quad \mathbf{Z} - \phi(\mathbf{X})\mathbf{W} - \mathbf{1}\mathbf{b}^\top = \mathbf{E}$$

Algorithm 1 The CAMEL Algorithm

Inputs:
$\mathcal{D}$: the multi-label training set $\mathcal{D} = \{(\mathbf{X}, \mathbf{Y})\}$
$\alpha, \lambda_1, \lambda_2$: the hyperparameters
$\mathbf{x}$: the unseen test instance
Output:
$\mathbf{y}$: the predicted label for the test instance $\mathbf{x}$

learn the label correlation matrix $\mathbf{S}$ by solving problem (3) for each label via ADMM procedure;
set $\mathbf{G} = (1-\alpha)\mathbf{I} + \alpha\mathbf{S}$;
initialize $\mathbf{Z} = \mathbf{Y}$;
construct the kernel matrix $\mathbf{K} = [\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)]_{n\times n}$ by Gaussian kernel function;
repeat
  update $\mathbf{b}$ and $\mathbf{A}$ according to (9);
  update $\mathbf{T} = \frac{1}{\lambda_2}\mathbf{KA} + \mathbf{1}\mathbf{b}^\top$;
  update $\mathbf{Z}$ in terms of (11);
until convergence.
return the predicted label vector $\mathbf{y}$ according to (12).

Optimization

Problem (7) is convex with respect to $\mathbf{W}$ and $\mathbf{b}$ with $\mathbf{Z}$ fixed, and also convex with respect to $\mathbf{Z}$ with $\mathbf{W}$ and $\mathbf{b}$ fixed. Therefore, it is a biconvex problem (Gorski, Pfeuffer, and Klamroth 2007), and can be solved by an alternating approach.

Updating $\mathbf{W}$ and $\mathbf{b}$ with $\mathbf{Z}$ fixed With $\mathbf{Z}$ fixed, problem (7) reduces to

$$\min_{\mathbf{E},\mathbf{W},\mathbf{b}} \frac{1}{2}\|\mathbf{E}\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 \tag{8}$$
$$\text{s.t.} \quad \mathbf{Z} - \phi(\mathbf{X})\mathbf{W} - \mathbf{1}\mathbf{b}^\top = \mathbf{E}$$

By deriving the Lagrangian of the above constrained problem and setting the gradient with respect to $\mathbf{W}$ to 0, it is easy to show $\mathbf{W} = \frac{1}{\lambda_2}\phi(\mathbf{X})^\top\mathbf{A}$ where $\mathbf{A} = [a_{ij}]_{n\times q}$ is the matrix that stores the Lagrangian multipliers. Let $\mathbf{K} = \phi(\mathbf{X})\phi(\mathbf{X})^\top$ be the kernel matrix with its element $k_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$, where $\mathcal{K}(\cdot,\cdot)$ represents the kernel function. For CAMEL, Gaussian kernel function $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/(2\sigma^2))$ is employed with σ set to the average Euclidean distance of all pairs of training instances. In this way, we choose to optimize with respect to

$\mathbf{A}$ and $\mathbf{b}$ instead, and the close-form solutions are reported as follows:

$$\mathbf{b}^\top = \frac{\mathbf{1}^\top \mathbf{H}^{-1}\mathbf{Z}}{\mathbf{1}^\top \mathbf{H}^{-1}\mathbf{1}} \qquad (9)$$
$$\mathbf{A} = \mathbf{H}^{-1}(\mathbf{Z} - \mathbf{1}\mathbf{b}^\top)$$

where $\mathbf{H} = \frac{1}{\lambda_2}\mathbf{K} + \mathbf{I}$. The detailed information is provided in Appendix B.

Updating Z with W and b fixed When $\mathbf{W}$ and $\mathbf{b}$ are fixed, the modeling output matrix $\mathbf{T} \in \mathbb{R}^{n\times q}$ is calculated by $\mathbf{T} = \phi(\mathbf{X})\mathbf{W} + \mathbf{1}\mathbf{b}^\top = \frac{1}{\lambda_2}\mathbf{K}\mathbf{A} + \mathbf{1}\mathbf{b}^\top$. By inserting $\mathbf{E} = \mathbf{Z} - \mathbf{T}$, problem (7) reduces to:

$$\min_{\mathbf{Z}} \frac{1}{2}\|\mathbf{Z} - \mathbf{T}\|_F^2 + \frac{\lambda_1}{2}\|\mathbf{Z}\mathbf{G} - \mathbf{Y}\|_F^2 \qquad (10)$$

Setting the gradient with respect to $\mathbf{Z}$ to 0, we can obtain the following closed-form solution:

$$\mathbf{Z} = (\mathbf{T} + \lambda_1\mathbf{Y}\mathbf{G}^\top)(\mathbf{I} + \lambda_1\mathbf{G}\mathbf{G}^\top)^{-1} \qquad (11)$$

Once the iterative process converges, the predicted label vector $\mathbf{y} \in \{-1, 1\}^l$ of the test instance $\mathbf{x}$ is given as:

$$\mathbf{y} = \text{sign}(\mathbf{G}^\top(\sum_{i=1}^{m}\mathbf{a}_i\mathcal{K}(\mathbf{x}, \mathbf{x}_i) + \mathbf{b})) \qquad (12)$$

The pseudo code of CAMEL is presented in Algorithm 1. Since the proposed formulation is biconvex, this alternating optimization process is guaranteed to converge (Gorski, Pfeuffer, and Klamroth 2007).

Experiments

In this section, we conduct extensive experiments on various datasets to validate the effectiveness of CAMEL.

Experimental Setup

Datasets For comprehensive performance evaluation, we collect sixteen benchmark multi-label datasets. For each dataset $\mathcal{S}$, we denote by $|\mathcal{S}|$, $dim(\mathcal{S})$, $L(\mathcal{S})$, $LCard(\mathcal{S})$, and $F(\mathcal{S})$ the number of examples, the number of features (dimensions), the number of distinct class labels, the average number of labels associated with each example, and feature type, respectively. Table 1 summarizes the detailed characteristics of these datasets, which are organized in ascending order of $|\mathcal{S}|$. According to $|\mathcal{S}|$, we further roughly divide these datasets into regular-size datasets ($|\mathcal{S}| < 5000$) and large-size datasets ($|\mathcal{S}| \geq 5000$). For performance evaluation, 10-fold cross-validation is conducted on these datasets, where mean metric values with standard deviations are recorded.

Evaluation Metrics For performance evaluation, we use seven widely-used evaluation metrics, including *One-error*, *Hamming loss*, *Coverage*, *Ranking loss*, *Average precision*, *Macro-averaging F1*, and *Micro-averaging F1*. Note that for all the employed multi-label evaluation metrics, their values vary within the interval [0,1]. In addition, for the last three metrics, the larger values indicate the better performance, and we use the symbol $\uparrow$ to present such positive logic. While for the first five metrics, the smaller values indicate the better performance, which is represented by $\downarrow$. More detailed information about these evaluation metrics can be found in (Zhang and Zhou 2014).

Comparing Approaches CAMEL is compared with three well-established and two state-of-the-art multi-label learning algorithms, including the first-order approach BR (Boutell et al. 2004), the second-order approaches LLSF (Huang et al. 2016) and JFSC (Huang et al. 2018), and the high-order approaches ECC (Read et al. 2011), and RAKEL (Tsoumakas, Katakis, and Vlahavas 2011). Here, LLSF and JFSC are the state-of-the-art counterparts using label correlation matrix.

BR, ECC, and RAKEL are implemented under the MULAN multi-label learning package (Tsoumakas et al. 2011) by using the logistic regression model as the base classifier. Furthermore, parameters suggested in the corresponding literatures are used, i.e., ECC: ensemble size 30; RAKEL: ensemble size $2q$ with $k = 3$. For LLSF, parameters α, β are chosen from $\{2^{-10}, 2^{-9}, \cdots, 2^{10}\}$, and ρ chosen from $\{0.1, 1, 10\}$. For JFSC, parameters α, β, and γ are chosen from $\{4^{-5}, 4^{-4}, \cdots, 4^5\}$, and η is chosen from $\{0.1, 1, 10\}$. For the proposed approach CAMEL, λ_1 is empirically set to 1, λ_2 is chosen from $\{10^{-3}, 2\times10^{-3}, 10^{-2}, 2\times10^{-2}, \cdots, 10^0\}$, and α is chosen from $\{0, 0.1, \cdots, 1\}$. All of these parameters are decided by conducting 5-fold cross-validation on training set.

Table 1: Characteristics of the benchmark multi-label datasets.

Dataset	$\|\mathcal{S}\|$	$dim(\mathcal{S})$	$L(\mathcal{S})$	$LCard(\mathcal{S})$	$F(\mathcal{S})$
cal500	502	68	174	26.04	numeric
emotions	593	72	6	1.87	numeric
genbase	662	1185	27	1.25	nominal
medical	978	1449	45	1.25	nominal
enron	1702	1001	53	3.38	nominal
image	2000	294	5	1.24	numeric
scene	2407	294	5	1.07	numeric
yeast	2417	103	14	4.24	numeric
science	5000	743	40	1.45	numeric
arts	5000	462	26	1.64	numeric
business	5000	438	30	1.59	numeric
rcv1-s1	6000	944	101	2.88	nominal
rcv1-s2	6000	944	101	2.63	nominal
rcv1-s3	6000	944	101	2.61	nominal
rcv1-s4	6000	944	101	2.48	nominal
rcv1-s5	6000	944	101	2.64	nominal

Experimental Results

Table 2 and 3 report the detailed experimental results on the regular-scale and large-scale datasets respectively, where the best performance among all the algorithms is shown in boldface. From the two result tables, we can see that CAMEL outperforms other comparing algorithms in most cases. Specifically, on the regular-size datasets (Table 2), across all the evaluation metrics, CAMEL ranks first in 80.4% (45/56)

Table 2: Predictive performance of each algorithm (mean±std. deviation) on the regular-scale datasets.

Comparing algorithms	One-error↓							
	cal500	emotions	genbase	medical	enron	image	scene	yeast
CAMEL	0.129±0.053	0.292±0.052	**0.001±0.001**	**0.110±0.021**	**0.207±0.038**	**0.242±0.033**	**0.175±0.027**	**0.218±0.027**
BR	0.893±0.038	**0.284±0.077**	0.017±0.016	0.322±0.055	0.646±0.023	0.387±0.027	0.361±0.036	0.244±0.028
ECC	0.295±0.036	0.296±0.074	0.010±0.013	0.156±0.037	0.421±0.034	0.406±0.023	0.306±0.020	0.238±0.030
RAKEL	0.634±0.039	0.300±0.070	0.009±0.007	0.243±0.055	0.532±0.007	0.402±0.024	0.280±0.031	0.244±0.027
LLSF	0.138±0.050	0.412±0.051	0.002±0.005	0.120±0.020	0.250±0.042	0.327±0.030	0.259±0.020	0.394±0.029
JFSC	**0.116±0.051**	0.438±0.086	0.002±0.005	0.128±0.024	0.278±0.041	0.346±0.023	0.266±0.022	0.242±0.021
Comparing algorithms	Hamming loss↓							
	cal500	emotions	genbase	medical	enron	image	scene	yeast
CAMEL	**0.136±0.005**	**0.203±0.021**	**0.001±0.001**	0.011±0.001	**0.045±0.003**	**0.144±0.012**	**0.072±0.009**	**0.190±0.005**
BR	0.189±0.005	0.216±0.028	0.002±0.002	0.026±0.003	0.111±0.006	0.210±0.014	0.139±0.009	0.205±0.007
ECC	0.154±0.005	0.214±0.027	0.009±0.004	0.011±0.002	0.067±0.002	0.210±0.016	0.112±0.006	0.204±0.010
RAKEL	0.195±0.004	0.238±0.025	0.002±0.001	0.020±0.002	0.092±0.004	0.223±0.013	0.139±0.008	0.224±0.009
LLSF	0.138±0.006	0.267±0.022	**0.001±0.001**	**0.010±0.002**	0.048±0.002	0.180±0.010	0.109±0.003	0.278±0.009
JFSC	0.191±0.004	0.295±0.019	**0.001±0.001**	**0.010±0.001**	0.051±0.003	0.188±0.012	0.110±0.007	0.206±0.006
Comparing algorithms	Coverage↓							
	cal500	emotions	genbase	medical	enron	image	scene	yeast
CAMEL	0.752±0.019	0.312±0.031	**0.012±0.005**	**0.028±0.012**	**0.239±0.028**	**0.156±0.016**	**0.062±0.006**	**0.446±0.010**
BR	0.786±0.015	0.319±0.026	0.014±0.006	0.113±0.030	0.580±0.023	0.216±0.018	0.168±0.015	0.463±0.011
ECC	0.796±0.019	**0.310±0.029**	0.013±0.003	0.034±0.012	0.291±0.020	0.233±0.022	0.135±0.010	0.460±0.010
RAKEL	0.962±0.016	0.362±0.027	0.014±0.005	0.095±0.018	0.513±0.019	0.253±0.017	0.169±0.013	0.544±0.013
LLSF	0.778±0.025	0.362±0.032	0.021±0.006	0.031±0.014	0.283±0.023	0.192±0.007	0.092±0.006	0.601±0.020
JFSC	**0.730±0.026**	0.392±0.046	0.014±0.007	0.030±0.012	0.314±0.024	0.200±0.009	0.102±0.007	0.455±0.011
Comparing algorithms	Ranking loss↓							
	cal500	emotions	genbase	medical	enron	image	scene	yeast
CAMEL	**0.177±0.009**	0.180±0.032	**0.001±0.001**	**0.016±0.008**	**0.079 ±0.028**	**0.128±0.013**	**0.058±0.005**	**0.162±0.007**
BR	0.233±0.007	0.182±0.030	0.003±0.004	0.091±0.027	0.304±0.014	0.204±0.017	0.151±0.015	0.176±0.008
ECC	0.219±0.007	**0.172±0.031**	0.002±0.002	0.022±0.010	0.118±0.008	0.225±0.023	0.117±0.010	0.179±0.009
RAKEL	0.366±0.008	0.225±0.029	0.002±0.001	0.073±0.018	0.244±0.017	0.221±0.018	0.131±0.014	0.240±0.009
LLSF	0.184±0.012	0.245±0.033	0.002±0.003	0.019±0.010	0.107±0.009	0.174±0.006	0.093±0.005	0.346±0.017
JFSC	0.188±0.010	0.271±0.041	0.003±0.003	0.017±0.008	0.118±0.013	0.183±0.007	0.105±0.007	0.179±0.009
Comparing algorithms	Average precision↑							
	cal500	emotions	genbase	medical	enron	image	scene	yeast
CAMEL	**0.515±0.018**	0.788±0.035	**0.997±0.003**	**0.917±0.017**	**0.718±0.025**	**0.843±0.018**	**0.897±0.012**	**0.775±0.013**
BR	0.345±0.018	0.783±0.040	0.988±0.008	0.750±0.036	0.388±0.016	0.753±0.016	0.771±0.021	0.754±0.013
ECC	0.442±0.014	**0.789±0.036**	0.991±0.008	0.884±0.023	0.557±0.015	0.738±0.020	0.811±0.012	0.756±0.014
RAKEL	0.329±0.016	0.763±0.039	0.993±0.006	0.800±0.032	0.456±0.019	0.735±0.017	0.804±0.022	0.720±0.014
LLSF	0.507±0.021	0.716±0.035	**0.997±0.005**	0.912±0.015	0.682±0.028	0.790±0.014	0.843±0.008	0.601±0.015
JFSC	0.492±0.020	0.691±0.040	**0.997±0.004**	0.908±0.016	0.655±0.025	0.779±0.011	0.835±0.010	0.746±0.012
Comparing algorithms	Macro-averaging F1↑							
	cal500	emotions	genbase	medical	enron	image	scene	yeast
CAMEL	0.180±0.032	**0.625±0.052**	**0.971±0.030**	**0.779±0.043**	0.325±0.044	**0.660±0.030**	**0.787±0.023**	**0.411±0.018**
BR	0.167±0.019	0.620±0.044	0.951±0.029	0.640±0.060	0.236±0.016	0.553±0.027	0.623±0.026	0.391±0.021
ECC	0.236±0.027	0.622±0.043	0.928±0.037	0.755±0.054	0.303±0.030	0.540±0.030	0.662±0.026	0.395±0.015
RAKEL	0.187±0.020	0.614±0.044	0.958±0.030	0.689±0.051	0.256±0.017	0.540±0.028	0.644±0.024	0.381±0.020
LLSF	0.180±0.031	0.615±0.056	**0.971±0.031**	0.769±0.057	0.292±0.043	0.554±0.031	0.615±0.007	0.235±0.016
JFSC	**0.239±0.031**	0.345±0.023	**0.971±0.031**	0.772±0.043	**0.339±0.048**	0.559±0.035	0.705±0.019	0.300±0.007
Comparing algorithms	Micro-averaging F1↑							
	cal500	emotions	genbase	medical	enron	image	scene	yeast
CAMEL	0.337±0.017	**0.649±0.041**	0.988±0.012	**0.835±0.019**	**0.580±0.023**	**0.659±0.031**	**0.780±0.026**	**0.655±0.010**
BR	0.339±0.016	0.639±0.050	0.978±0.014	0.611±0.032	0.359±0.015	0.558±0.028	0.619±0.023	0.633±0.013
ECC	0.364±0.015	0.642±0.046	0.907±0.035	0.796±0.023	0.452±0.015	0.541±0.030	0.653±0.023	0.643±0.017
RAKEL	0.351±0.018	0.629±0.049	0.983±0.011	0.678±0.042	0.392±0.014	0.541±0.031	0.629±0.026	0.632±0.016
LLSF	0.325±0.015	0.637±0.049	0.992±0.003	0.823±0.027	0.534±0.025	0.557±0.032	0.618±0.008	0.280±0.018
JFSC	**0.473±0.013**	0.406±0.022	**0.995±0.006**	0.818±0.018	0.555±0.026	0.565±0.033	0.695±0.022	0.609±0.012

cases, and on the large-scale datasets (Table 3), across all the evaluation metrics, CAMEL ranks first in 69.6% (39/56) cases. Compared with the three well-established algorithms BR, ECC, and RAKEL, CAMEL introduces a new type of label correlations, i.e., collaborative relationships among labels, and achieves superior performance in 93.8% (315/336) cases. Compared with the two state-of-the-art algorithms LLSF and JFSC, instead of employing simple similarity measures to regularize the hypothesis space, CAMEL introduces a novel method to learn label correlations for explicitly correlating the final predictions, and achieves superior performance in 80.4% (180/224) cases. These comparative results clearly demonstrate the effectiveness of the collaboration based multi-label learning approach.

Sensitivity Analysis In this section, we first investigate the sensitivity of CAMEL with respect to the two tradeoff parameters λ_1 and λ_2, and the parameter α that controls the degree of collaboration, then illustrate the convergence of CAMEL. Due to page limit, we only report the experimental results on the enron dataset using the *Coverage* (↓) metric. Concretely, we study the performance of CAMEL when we vary one parameter while keeping other parameters fixed at

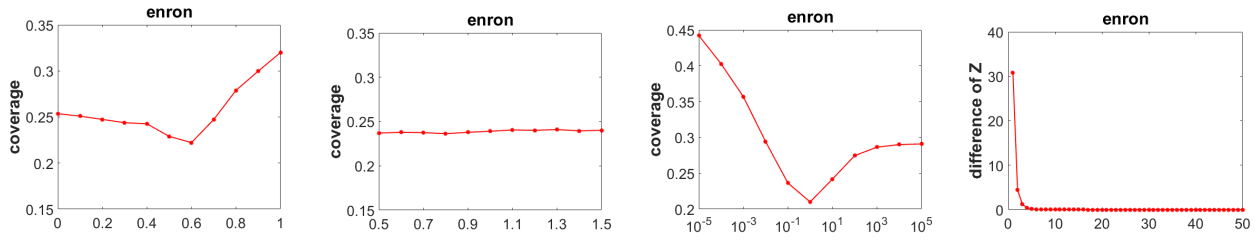
Table 3: Predictive performance of each algorithm (mean±std. deviation) on the large-scale datasets.

Comparing algorithms	One-error↓							
	science	arts	rcv1-s1	rcv1-s2	rcv1-s3	rcv1-s4	rcv1-s5	business
CAMEL	**0.457±0.021**	0.462±0.024	**0.404±0.019**	**0.403±0.018**	**0.413±0.019**	**0.331±0.016**	0.404±0.010	**0.101±0.009**
BR	0.760±0.015	0.642±0.022	0.742±0.019	0.723±0.024	0.718±0.021	0.662±0.021	0.715±0.015	0.417±0.016
ECC	0.574±0.022	0.526±0.023	0.471±0.020	0.441±0.021	0.448±0.021	0.378±0.019	0.425±0.016	0.153±0.008
RAKEL	0.623±0.014	0.543±0.024	0.613±0.019	0.592±0.022	0.578±0.020	0.552±0.020	0.575±0.014	0.201±0.009
LLSF	0.486±0.013	0.454±0.027	0.409±0.015	0.406±0.016	0.415±0.021	0.333±0.016	**0.399±0.018**	0.122±0.016
JFSC	0.489±0.027	**0.447±0.027**	0.418±0.016	0.407±0.014	0.418±0.025	0.337±0.015	0.407±0.023	0.122±0.019
Comparing algorithms	Hamming loss↓							
	science	arts	rcv1-s1	rcv1-s2	rcv1-s3	rcv1-s4	rcv1-s5	business
CAMEL	**0.030±0.001**	0.055±0.002	**0.026±0.008**	**0.023±0.001**	**0.023±0.001**	**0.018±0.001**	**0.022±0.001**	**0.024±0.001**
BR	0.072±0.002	0.079±0.003	0.056±0.001	0.053±0.001	0.053±0.001	0.041±0.001	0.051±0.002	0.049±0.001
ECC	0.036±0.002	0.063±0.002	0.028±0.001	0.024±0.001	0.024±0.001	0.019±0.001	0.024±0.001	0.030±0.001
RAKEL	0.042±0.002	0.075±0.002	0.046±0.001	0.039±0.001	0.035±0.001	0.035±0.001	0.036±0.003	0.035±0.002
LLSF	0.036±0.002	**0.054±0.002**	0.027±0.001	0.025±0.001	0.025±0.001	0.019±0.001	0.023±0.001	0.048±0.007
JFSC	0.035±0.002	0.057±0.002	0.029±0.001	0.025±0.001	0.025±0.001	0.019±0.001	0.025±0.001	0.027±0.002
Comparing algorithms	Coverage↓							
	science	arts	rcv1-s1	rcv1-s2	rcv1-s3	rcv1-s4	rcv1-s5	business
CAMEL	**0.189±0.010**	0.205±0.009	0.151±0.008	**0.142±0.012**	**0.131±0.006**	0.143±0.003	**0.132±0.005**	**0.082±0.006**
BR	0.303±0.011	0.204±0.009	0.393±0.011	0.341±0.013	0.351±0.018	0.294±0.015	0.336±0.013	0.141±0.002
ECC	0.196±0.009	0.229±0.009	0.166±0.011	0.154±0.007	0.154±0.008	0.108±0.003	0.145±0.001	0.104±0.001
RAKEL	0.209±0.012	0.214±0.008	0.273±0.011	0.329±0.012	0.293±0.017	0.273±0.012	0.246±0.012	0.107±0.003
LLSF	0.197±0.014	**0.195±0.011**	0.141±0.009	0.146±0.008	0.133±0.008	0.109±0.006	0.133±0.006	0.086±0.013
JFSC	0.196±0.011	0.233±0.018	**0.140±0.006**	0.143±0.009	0.136±0.010	**0.106±0.005**	0.139±0.006	0.086±0.011
Comparing algorithms	Ranking loss↓							
	science	arts	rcv1-s1	rcv1-s2	rcv1-s3	rcv1-s4	rcv1-s5	business
CAMEL	**0.139±0.007**	**0.135±0.008**	**0.058±0.003**	0.077±0.005	**0.047±0.003**	0.057±0.002	0.073±0.002	**0.040±0.004**
BR	0.245±0.009	0.145±0.006	0.197±0.006	0.190±0.008	0.198±0.010	0.173±0.009	0.181±0.006	0.088±0.006
ECC	0.151±0.006	0.164±0.007	0.074±0.005	0.069±0.003	0.070±0.002	0.047±0.004	0.063±0.003	0.055±0.002
RAKEL	0.195±0.007	0.156±0.008	0.183±0.006	0.153±0.008	0.178±0.010	0.112±0.009	0.123±0.006	0.067±0.005
LLSF	0.149±0.009	0.141±0.009	0.060±0.003	**0.060±0.004**	0.048±0.003	**0.034±0.003**	**0.045±0.003**	0.045±0.009
JFSC	0.147±0.008	0.159±0.009	0.061±0.003	0.062±0.006	0.061±0.004	0.047±0.003	0.060±0.003	0.045±0.008
Comparing algorithms	Average precision↑							
	science	arts	rcv1-s1	rcv1-s2	rcv1-s3	rcv1-s4	rcv1-s5	business
CAMEL	**0.624±0.016**	0.607±0.018	0.615±0.009	**0.644±0.012**	**0.635±0.010**	**0.717±0.008**	**0.626±0.009**	**0.891±0.009**
BR	0.383±0.011	0.514±0.013	0.353±0.011	0.382±0.015	0.382±0.015	0.443±0.013	0.390±0.009	0.709±0.008
ECC	0.516±0.020	0.553±0.018	0.545±0.016	0.587±0.015	0.585±0.016	0.677±0.017	0.600±0.009	0.844±0.005
RAKEL	0.487±0.012	0.526±0.015	0.424±0.012	0.489±0.016	0.459±0.014	0.479±0.012	0.432±0.009	0.858±0.007
LLSF	0.594±0.021	**0.631±0.016**	**0.627±0.009**	0.637±0.008	0.632±0.013	0.714±0.010	0.625±0.013	0.867±0.013
JFSC	0.595±0.020	0.621±0.020	0.606±0.008	0.630±0.009	0.624±0.014	0.700±0.012	0.624±0.013	0.874±0.018
Comparing algorithms	Macro-averaging F1↑							
	science	arts	rcv1-s1	rcv1-s2	rcv1-s3	rcv1-s4	rcv1-s5	business
CAMEL	**0.310±0.038**	**0.312±0.029**	0.250±0.023	**0.258±0.022**	0.247±0.025	**0.340±0.031**	0.253±0.016	**0.326±0.046**
BR	0.215±0.048	0.257±0.020	0.232±0.018	0.210±0.017	0.221±0.019	0.313±0.016	0.236±0.019	0.249±0.017
ECC	0.285±0.024	0.282±0.021	0.271±0.023	0.257±0.022	0.266±0.012	0.334±0.018	**0.285±0.014**	**0.326±0.032**
RAKEL	0.267±0.028	0.275±0.019	0.266±0.019	0.237±0.023	0.243±0.017	0.322±0.017	0.255±0.018	0.307±0.024
LLSF	0.312±0.038	0.219±0.032	0.261±0.022	0.257±0.025	**0.270±0.027**	0.334±0.031	0.217±0.018	0.325±0.028
JFSC	0.308±0.039	0.305±0.032	**0.308±0.026**	0.249±0.019	0.258±0.024	0.337±0.032	0.254±0.019	0.318±0.036
Comparing algorithms	Micro-averaging F1↑							
	science	arts	rcv1-s1	rcv1-s2	rcv1-s3	rcv1-s4	rcv1-s5	business
CAMEL	0.428±0.018	0.415±0.015	0.401±0.015	**0.437±0.017**	**0.431±0.025**	**0.491±0.017**	**0.441±0.015**	**0.746±0.011**
BR	0.277±0.013	0.349±0.018	0.301±0.009	0.310±0.009	0.307±0.013	0.356±0.009	0.321±0.009	0.595±0.003
ECC	0.343±0.028	0.377±0.018	0.385±0.016	0.410±0.022	0.414±0.013	0.482±0.024	0.440±0.011	0.690±0.007
RAKEL	0.337±0.014	0.368±0.017	0.341±0.010	0.337±0.008	0.335±0.014	0.369±0.008	0.350±0.008	0.701±0.014
LLSF	0.446±0.025	0.368±0.018	**0.463±0.016**	0.432±0.018	0.428±0.023	0.478±0.017	0.438±0.019	0.693±0.035
JFSC	**0.449±0.026**	**0.442±0.017**	0.456±0.008	0.422±0.011	0.424±0.012	0.482±0.013	0.438±0.011	0.712±0.021

their best setting. Figure 1(a), 1(b), and 1(c) show the sensitivity curve of CAMEL with respect to α, λ_1, and λ_2 respectively. It can be seen that α and λ_2 have an important influence on the final performance, because α and λ_2 control the collaboration degree and the model complexity. Figure 1(d) illustrates the convergence of CAMEL by using the difference of the optimization variable $\mathbf{Z}$ between two successive iterations, i.e., $\Delta\mathbf{Z} = \left\|\mathbf{Z}^{(t)} - \mathbf{Z}^{(t-1)}\right\|_F$. From Figure 1(d), we can observe that $\Delta\mathbf{Z}$ quickly decreases to 0 within a few number of iterations. Hence the convergence of CAMEL is demonstrated.

Conclusion

In this paper, we make a key assumption for multi-label learning that *for each individual label, the final prediction involves the collaboration between its own prediction and the predictions of other labels.* Guided by this assumption, we propose a novel method to learn the high-order label correlations via sparse reconstruction in the label space. Besides, by seamlessly integrating the learned label correlations into model training, we propose a novel multi-label learning approach that aims to explicitly account for the correlated predictions of labels while training the desired model simul-



(a) α (b) λ_1 (c) λ_2 (d) Convergence Curve

Figure 1: Parameter sensitivity and convergence analysis of CAMEL on the enron dataset.

taneously. Extensive experimental results show that our approach outperforms the state-of-the-art counterparts.

Despite the demonstrated effectiveness of CAMEL, it only considers the global collaborative relationships between labels, by assuming that such collaborative relationships are shared by all the instances. However, as different instances have different characteristics, such collaborative relationships may be shared by only a subset of instances rather than all the instances. Therefore, our further work is to explore different collaborative relationships between labels for different subsets of instances.

Acknowledgements

This work was supported by MOE, NRF, and NTU.

Appendix A. The ADMM Procedure

To solve problem (3) by ADMM, we first reformulate problem (3) into the following equivalent form:

$$\min_{\mathbf{S}_j,\mathbf{z}} \frac{1}{2}\left\|\mathbf{Y}_{-j}\mathbf{S}_i-\mathbf{Y}_j\right\|_2^2+\lambda\left\|\mathbf{z}\right\|_1 \quad (13)$$
$$\text{s.t.} \quad \mathbf{S}_j-\mathbf{z}=0$$

Following the ADMM procedure, the above constrained optimization problem (13) can be solved as a series of unconstrained minimization problems using augmented Lagrangian function, which is presented as:

$$\mathcal{L}(\mathbf{S}_j,\mathbf{z},\boldsymbol{\mu})=\frac{1}{2}\left\|\mathbf{Y}_{-j}\mathbf{S}_j-\mathbf{Y}_j\right\|_2^2+\lambda\left\|\mathbf{z}\right\|_1+ \quad (14)$$
$$\mathbf{v}^\top(\mathbf{S}_j-\mathbf{z})+\frac{\rho}{2}\left\|\mathbf{S}_j-\mathbf{z}\right\|_2^2$$

Here, ρ is the penalty parameter and $\mathbf{v}$ is the Lagrange multiplier. By introducing the scaled dual variable $\boldsymbol{\mu}=\frac{1}{\rho}\mathbf{v}$, a sequential minimization of the scaled ADMM iterations can be conducted by updating the three variables $\mathbf{S}_j$, $\mathbf{z}$ and $\boldsymbol{\mu}$ sequentially:

$$\begin{aligned}\mathbf{S}_j^{(k+1)}&=(\mathbf{Y}_{-j}^\top\mathbf{Y}_{-j}+\rho\mathbf{I})^{-1}(\mathbf{Y}_{-j}^\top\mathbf{Y}_j+\rho(\mathbf{z}^{(k)}-\boldsymbol{\mu}^{(k)}))\\ \mathbf{z}^{(k+1)}&=S_{\lambda/\rho}(\mathbf{S}_j^{(k+1)}+\boldsymbol{\mu}^{(k)})\\ \boldsymbol{\mu}^{(k+1)}&=\boldsymbol{\mu}^{(k)}+\mathbf{S}^{(k+1)}-\mathbf{z}^{(k+1)}\end{aligned} \quad (15)$$

where S is the proximity operator of the ℓ_1 norm, which is defined as $S_\omega(a)=(a-\omega)_+-(-a-\omega)_+$.

Appendix B. Model Parameter Optimization

The Lagrangian of problem (8) is expressed as:

$$\mathcal{L}(\mathbf{W},\mathbf{E},\mathbf{A},\mathbf{b})=\mathrm{tr}(\mathbf{E}^\top\mathbf{E})+\lambda_2\,\mathrm{tr}(\mathbf{W}^\top\mathbf{W})+ \quad (16)$$
$$\mathrm{tr}\left(\mathbf{A}^\top(\mathbf{Z}-\phi(\mathbf{X})\mathbf{W}-\mathbf{1}\mathbf{b}^\top-\mathbf{E})\right)$$

where tr is the trace operator, and $\mathbf{A}=[\mathbf{a}_1,\mathbf{a}_2,\cdots,\mathbf{a}_n]^\top\in\mathbb{R}^{n\times q}$ is the introduced matrix that stores the Lagrangian multipliers. Besides, we have used the property of trace operator that $\mathrm{tr}(\mathbf{W}^\top\mathbf{W})=\|\mathbf{W}\|_F^2$. By Setting the gradient w.r.t. $\mathbf{E},\mathbf{A},\mathbf{W},\mathbf{b}$ to 0 respectively, the following equations will be induced:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{E}}&=0\Rightarrow\mathbf{A}=\mathbf{E}\\ \frac{\partial\mathcal{L}}{\partial\mathbf{A}}&=0\Rightarrow\mathbf{Z}-\phi(\mathbf{X})\mathbf{W}-\mathbf{1}\mathbf{b}^\top=\mathbf{E}\\ \frac{\partial\mathcal{L}}{\partial\mathbf{W}}&=0\Rightarrow\mathbf{W}=\frac{1}{\lambda_2}\phi(\mathbf{X})^\top\mathbf{A}\\ \frac{\partial\mathcal{L}}{\partial\mathbf{b}}&=0\Rightarrow\mathbf{A}^\top\mathbf{1}=\mathbf{0}\end{aligned} \quad (17)$$

The above linear equations can be simplified by the following steps:

$$\begin{aligned}\mathbf{Z}&=\phi(\mathbf{X})\mathbf{W}+\mathbf{1}\mathbf{b}^\top+\mathbf{E}\\ \mathbf{Z}&=\frac{1}{\lambda_2}\phi(\mathbf{X})\phi(\mathbf{X})^\top\mathbf{A}+\mathbf{1}\mathbf{b}^\top+\mathbf{A}\\ \mathbf{Z}&=\frac{1}{\lambda_2}\mathbf{K}\mathbf{A}+\mathbf{1}\mathbf{b}^\top+\mathbf{A}\end{aligned} \quad (18)$$

Here, we define $\mathbf{H}=\frac{1}{\lambda_2}\mathbf{K}+\mathbf{I}$, then we can obtain:

$$\begin{aligned}\mathbf{H}\mathbf{A}+\mathbf{1}\mathbf{b}^\top&=\mathbf{Z}\\ \mathbf{A}+\mathbf{H}^{-1}\mathbf{1}\mathbf{b}^\top&=\mathbf{H}^{-1}\mathbf{Z}\\ \mathbf{1}^\top\mathbf{H}^{-1}\mathbf{1}\mathbf{b}^\top&=\mathbf{1}^\top\mathbf{H}^{-1}\mathbf{Z}\\ \mathbf{b}^\top&=\frac{\mathbf{1}\mathbf{H}^{-1}\mathbf{Z}}{\mathbf{1}^\top\mathbf{H}^{-1}\mathbf{1}}\end{aligned} \quad (19)$$

In this way, $\mathbf{A}$ can be calculated by $\mathbf{A}=\mathbf{H}^{-1}(\mathbf{Z}-\mathbf{1}\mathbf{b}^\top)$.

References

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

Cai, X.; Nie, F.; Cai, W.; and Huang, H. 2013. New graph structured sparsity model for multi-label image annotations. In *Proceedings of the IEEE International Conference on Computer Vision*, 801–808.

Cesa-Bianchi, N.; Gentile, C.; and Zaniboni, L. 2006. Hierarchical classification: combining bayes with svm. In *Proceedings of the 23rd International Conference on Machine learning*, 177–184.

Elisseeff, A., and Weston, J. 2002. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems*, 681–687.

Gibaja, E., and Ventura, S. 2015. A tutorial on multilabel learning. *ACM Computing Surveys* 47(3):52.

Gopal, S., and Yang, Y. 2010. Multilabel classification with meta-level features. In *Proceedings of the 33rd international ACM SIGIR Conference on Research and Development in Information Retrieval*, 315–322.

Gorski, J.; Pfeuffer, F.; and Klamroth, K. 2007. Biconvex sets and optimization with biconvex functions: A survey and extensions. *Mathematical Methods of Operations Research* 66(3):373–407.

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

Huang, J.; Li, G.; Huang, Q.; and Wu, X. 2016. Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering* 28(12):3309–3323.

Huang, J.; Li, G.; Huang, Q.; and Wu, X. 2018. Joint feature selection and classification for multilabel learning. *IEEE Transactions on Cybernetics* 48(3):876–889.

Huang, S.-J.; Yu, Y.; and Zhou, Z.-H. 2012. Multi-label hypothesis reuse. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 525–533.

Huang, S.-J.; Zhou, Z.-H.; and Zhou, Z. 2012. Multi-label learning by exploiting label correlations locally. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 949–955.

Ji, S.; Tang, L.; Yu, S.; and Ye, J. 2008. Extracting shared subspace for multi-label classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 381–389.

Li, X.; Ouyang, J.; and Zhou, X. 2015. Supervised topic models for multi-label classification. *Neurocomputing* 149:811–819.

Petterson, J., and Caetano, T. S. 2011. Submodular multi-label learning. In *Advances in Neural Information Processing Systems*, 1512–1520.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine Learning* 85(3):333.

Tsoumakas, G.; Dimou, A.; Spyromitros, E.; Mezaris, V.; Kompatsiaris, I.; and Vlahavas, I. 2009. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proceedings of the 1st International Workshop on Learning from Multi-Label Data*, 101–116.

Tsoumakas, G.; Spyromitros-Xioufis, E.; Vilcek, J.; and Vlahavas, I. 2011. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research* 12(7):2411–2414.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2011. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering* 23(7):1079–1089.

Yang, H.; Tianyi Zhou, J.; Zhang, Y.; Gao, B.-B.; Wu, J.; and Cai, J. 2016. Exploit bounding box annotations for multi-label object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 280–288.

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

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

Zhu, S.; Ji, X.; Xu, W.; and Gong, Y. 2005. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 274–281.

Zhu, Y.; Kwok, J. T.; and Zhou, Z.-H. 2018. Multi-label learning with global and local label correlation. *IEEE Transactions on Knowledge and Data Engineering* 30(6):1081–1094.