Bi-Directional Representation Learning for Multi-label Classification

Xin Li and Yuhong Guo

Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122, USA
{xinli,yuhong}@temple.edu

Abstract. Multi-label classification is a central problem in many application domains. In this paper, we present a novel supervised bi-directional model that learns a low-dimensional mid-level representation for multi-label classification. Unlike traditional multi-label learning methods which identify intermediate representations from either the input space or the output space but not both, the mid-level representation in our model has two complementary parts that capture intrinsic information of the input data and the output labels respectively under the autoencoder principle while augmenting each other for the target output label prediction. The resulting optimization problem can be solved efficiently using an iterative procedure with alternating steps, while closed-form solutions exist for one major step. Our experiments conducted on a variety of multi-label data sets demonstrate the efficacy of the proposed bi-directional representation learning model for multi-label classification.

1 Introduction

Multi-label classification is a central problem in many areas of data analysis, where each data instance can simultaneously have multiple class labels. For example, in image labelling [3, 13], an image can contain multiple objects of interest and thus have multiple annotation tags; in text categorization [20], a webpage can be assigned into multiple related topic categories; similarly, in gene or protein function prediction [4], a gene or protein can exhibit multiple functions. Moreover, in these multi-label classification problems, strong label co-occurrences and label dependencies usually exist. For example, an object "*computer*" often appears together with the object "*desk*", but is rarely seen together with the object "*cooking pan*". Hence different from the standard multi-class problems where each instance is mapped to a single class label, multi-label classification needs to map each instance to typically a few interdependent class labels in a relatively large output space.

One straightforward approach for multi-label classification is to decompose the multi-label learning problem into a set of independent binary classification problems [16], which however has the obvious drawback of ignoring the interdependencies among multiple binary prediction tasks. Exploiting label prediction

dependence is critical for multi-label learning, especially when the label information is sparse. A group of methods in the literature explore label prediction dependencies or correlations by identifying mid-level low-dimensional representations shared across labels from the input data [8, 21, 26, 22, 14, 23]. Many other methods exploit the label dependency information directly in the output space [4, 6, 12, 24, 25]. Moreover, a number of recent works perform label space reduction to produce a low-dimensional intermediate label representation to facilitate multi-label classification with many labels [1, 11, 19, 25]. They demonstrate that even simple label space reduction can capture intrinsic information in the output space for multi-label classification tasks while reducing the computational cost.

In this paper, we propose a novel bi-directional model for multi-label classification, which introduces a compact mid-level representation layer between the input features and the output labels to capture the common prediction representations shared across multiple labels. The mid-level representation layer is constructed from both input and output spaces, and it has two complementary parts, one of which captures the *predictive* low-dimensional semantic representation of the input features and the other captures the *predictable* low-dimensional intrinsic representation of the output labels. These two parts augment each other to integrate information encoded in both the feature and label spaces and enhance the overall multi-label classification.

This bi-directional model exploits the autoencoder principle [9] to generate the mid-level representation from two directions, while extending this principle by promoting the discriminability of the mid-level representation for predicting the target output labels. We formalize this model as a joint optimization problem over all the encoding/decoding/prediction parameters. We show that this optimization problem can be solved using an iterative optimization algorithm with alternating steps, in which one major step has efficient closed-form solution. We conduct experiments on a variety of multi-label classification data sets. The results show the proposed model outperforms its one-directional component models, and a number of multi-label classification methods.

The remainder of the paper is organized as follows. We review related works in Section 2 and present the proposed model in Section 3. The experiments are reported in Section 4. We finally conclude the paper in Section 5.

2 Related Work

Multi-label classification has received significant attention from machine learning community in recent years. There is a rich body of work on multi-label learning in the literature. In this section, we present a brief review over existing methods that are closely related to the proposed work.

One direction of multi-label learning research exploits bottom-up learning schemes that induce intermediate layers from the inputs to bridge the gap between the input features and output labels. For example, [10] trained multiple base models on randomly selected subsets of the input features and then combined the multiple models. This method however ignores label dependencies.

To amend this drawback, [26] proposed a multi-label dimensionality reduction method, which induces a low-dimensional feature space by maximizing the dependence between the original feature description and the class labels. [5] first augmented the original input features with the output of base binary classifiers and then performed multi-label learning on the augmented representation. A few other methods perform feature selection for multi-label learning [23, 15]. [23] used a combination of PCA and genetic algorithms to search for the best feature subset. [15] proposed a feature selection algorithm that takes feature-label correlation into account based on a symmetrical uncertainty measure. A more advanced method [8] performs sparse feature learning with sparsity inducing norms to capture common predictive model structures across labels. The methods in [21, 22, 14] explore common subspaces shared among multiple labels.

Another set of methods exploit top-down learning schemes and induce alternative label representations from the original label space to facilitate multi-label learning. [24] applied canonical correlation analysis to extract error-correcting code words as intermediate prediction outputs between the input features and the original output labels. The work in [25] further enhanced this output coding learning scheme with a maximum margin output coding (MMOC) method, which formulates the output coding problem in a max-margin form to capture both discriminability and predictability of the output codes. [7] extended the kernel techniques widely used in the input feature space into the output label space to induce kernelized outputs in a large margin multi-label learning framework.

Moreover, a number of works pursue label space dimensionality reduction to produce intermediate low-dimensional label representations. An early work in [11] employs random label projection to address multi-label classification with a large number of labels. It first projects the high-dimensional label vectors to a low-dimensional space using a random transformation matrix, and then learns a multi-dimension regression model with the transformed labels. For a test instance, the estimated label vector from the regression models is then projected from the low-dimensional space back to the original high-dimensional label space. Following this work, a number of improvements have been proposed. [19] proposed a principal label space transformation method, which employs the principal component analysis to reduce the original label matrix to a low-dimensional space. Unlike random projections, the PCA dimensionality reduction minimizes the encoding error between the projected label matrix and the original label matrix. Subsequently, [1] proposed a conditional principal label space transformation method. It is a feature-aware method, which simultaneously minimizes both the label encoding error and the least squares linear regression error in the reduced label space. [27] proposed a Gaussian random projection method for label space transformation. Though these label space transformation methods have demonstrated that the intrinsic information of output labels can be captured in a low-dimensional output space, they mainly focus on reducing the computational cost without much loss of the multi-label classification performance.

Different from these existing methods, the proposed bi-directional model in this paper integrates the strengths of both bottom-up and top-down interme-

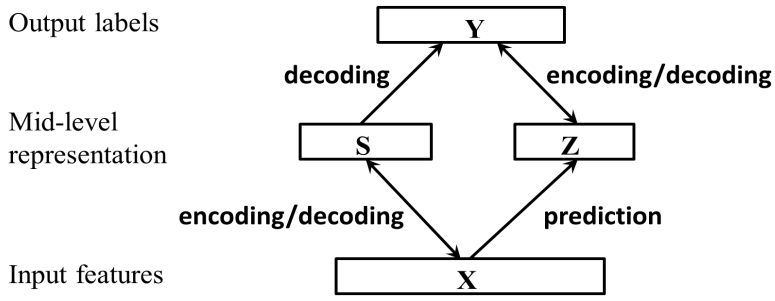

Fig. 1. The proposed bi-directional model. **X** denotes the input features and **Y** denotes the output labels. The mid-level latent layer has two parts, **S** and **Z**. **S** is the low-dimensional representation of the input features, and **Z** is the low-dimensional representation of the output labels. Both **S** and **Z** contribute to the decoding of the target output labels.

diate representation learning schemes in a complementary structure, aiming to improve multi-label classification performance. Our learning framework is not about producing any ad-hoc latent representations from the inputs and outputs, but aims to augment each other from two directions. Our model extends the generative autoencoder principle [9] in a discriminative way, sharing some similarity with the multi-class zero-shot learning approach in [18]. The work in [18] nevertheless is still a one-directional method that learns low-dimensional semantic representations from the inputs.

3 A Bi-Directional Model for Multi-label Classification

Traditional multi-label models typically learn a mapping function from the input space to the output space directly. Recent studies show that an intermediate representation can be very useful for bridging the original inputs and outputs, as we discussed in Section 2. Nevertheless, all these previous works have focused on one-directional representation learning using either bottom-up or top-down schemes. In this section, we present a novel bi-directional representation learning model for multi-label classification, which has a hybrid mid-level representation layer that captures both feature-sourced and label-sourced intrinsic information of the data in a complementary way, aiming to boost the learning performance.

Figure 1 shows the proposed bi-directional model. In this model, **X** and **Y** denote the input features and the output labels respectively. The latent mid-level layer has two parts, **S** and **Z**, which encode information from two directions in a low-dimensional space. The low-dimensional representation code **S** is constructed from the input **X** using the autoencoder principle such that **S** can be produced from **X** with an encoding function and **X** can be reconstructed from **S** with a decoding function. This mechanism ensures that **S** captures the intrinsic

information stored in the input features. The latent representation code $\mathbf{Z}$ is produced from the output $\mathbf{Y}$ with an encoding function, while its predictability from the input $\mathbf{X}$ is simultaneously promoted with a prediction function. To ensure the informativeness of the latent layer for the target output prediction, both $\mathbf{S}$ and $\mathbf{Z}$ are used to predict the output $\mathbf{Y}$ with a joint decoding function. With such a learning structure, we expect $\mathbf{S}$ and $\mathbf{Z}$ can contribute complementary information for accurate identification of $\mathbf{Y}$. Moreover, in the test phase, multi-label classification can be naturally achieved by first following the double line information flow from $\mathbf{X}$ to $\mathbf{S}$ and $\mathbf{Z}$, and then decoding $\mathbf{Y}$ from the concatenation of $\mathbf{S}$ and $\mathbf{Z}$.

Below we will introduce the components of the proposed model and the training algorithm in detail. The following notations will be used. We assume a set of t training instances, (X, Y), is given, where $X \in \mathbb{R}^{t\times d}$ is the input data matrix and $Y \in \{0,1\}^{t\times k}$ is the label indicator matrix. The low-dimensional representation matrix of X is denoted as $S \in \mathbb{R}^{t\times m}$ for $m < d$, and the low-dimensional representation matrix of Y is denoted as $Z \in \mathbb{R}^{t\times n}$ for $n < k$. We use $\mathbf{1}$ to denote any column vector with all 1 values, assuming its length can be determined from the context; use I_t to denote an identity matrix with size t; and use "$\circ$" to denote the Hadamard product operator between two matrices.

3.1 Framework: Encoding, Prediction and Decoding

The learning framework on the proposed model involves three major components: encoding, prediction and decoding, which follows but extends the standard autoencoder models. We will introduce each of them below.

Encoding. We propose to use typical sigmoid based functions to perform nonlinear encoding over the input data matrix X and the output label matrix Y, which map X to the low-dimensional latent representation matrix S and map Y to the low-dimensional latent representation matrix Z respectively. The two encoder functions are compositions of the standard sigmoid function and linear functions:

$$S = \sigma(XW_x + \mathbf{1}\mathbf{b}_x^\top), \tag{1}$$

$$Z = \sigma(YW_y + \mathbf{1}\mathbf{b}_y^\top) \tag{2}$$

where $(W_x \in \mathbb{R}^{d\times m}, \mathbf{b}_x \in \mathbb{R}^m)$ and $(W_y \in \mathbb{R}^{k\times n}, \mathbf{b}_y \in \mathbb{R}^n)$ are the linear model parameters for the two encoder functions respectively; $\sigma(x) = 1/(1+\exp(-x))$ is a sigmoid function that encodes entry-wise nonlinear transformation of the input. Moreover the sigmoid functions in the two encoder functions also put the values of S and Z in a comparable range.

A linear version of the encoder functions can be obtained by simply dropping the outer sigmoid functions.

Prediction. The encoder function over Y produces a low-dimensional mid-level prediction target Z for the input data. To ensure the information flow from the

input data to the output labels, we consider a prediction function, $f : \mathcal{X} \to \mathcal{Z}$, that maps X to the low-dimensional representation code Z. In particular, we consider the following linear regression function:

$$\hat{Z} = f(X) = X\Omega + \mathbf{1}\mathbf{q}^\top \tag{3}$$

where $\Omega \in \mathbb{R}^{d\times n}$ and $\mathbf{q} \in \mathbb{R}^n$ are the prediction model parameters, and $\hat{Z}$ denotes the prediction value matrix. This prediction function will be learned by minimizing a prediction loss $\ell_p(Z, \hat{Z})$ between the low-dimensional matrix Z and the prediction value matrix $\hat{Z}$ produced by the predictor over the input data. It thus enforces the predictability of Z from the inputs. This component is also necessary for exploiting the latent representation part $\mathbf{Z}$ in the test phase. In the test phase, where Y is unknown, one can produce Z from the input data using this prediction function and then use Z and S to reconstruct Y.

Decoding. There are two decoder functions, $g(\cdot)$ and $h(\cdot)$, in our model to reconstruct the observed data X and Y from the latent low-dimensional code matrices S and Z. The decoder function $g : \mathcal{S} \to \mathcal{X}$ reconstructs the input data X in its original space from its low-dimensional representation S. We consider a linear decoder function:

$$\hat{X} = g(S) = SU + \mathbf{1}\mathbf{d}^\top \tag{4}$$

where $U \in \mathbb{R}^{m\times d}$ and $\mathbf{d} \in \mathbb{R}^d$ are decoding parameters, and $\hat{X}$ is the reconstructed data matrix in the original input space. This decoder function and the encoder function from X to S in Eq. (1) together form an *autoencoder* model. An *autoencoder* in general is a two-layered construction, in which a first layer encodes the input data into the latent low-dimensional representation and a second layer decodes this representation back into the original data space. The *autoencoder* is trained to minimize the reconstruction error in the original data space. In our model, the two-layered constructions of the autoencoder correspond to the encoding from X to S and the decoding from S to $\hat{X}$ respectively. We will minimize a decoding loss $\ell_d(X, \hat{X})$ between the original input data X and the reconstructed data $\hat{X}$ in the training process.

The reconstruction of Y however is not a standard decoding problem in an autoencoder, since it is the key step for the overall multi-label classification and information from the input data should be taken into account. Hence instead of reconstructing it from its low-dimensional representation Z, we consider a decoder function $h : \mathcal{S} \times \mathcal{Z} \to \mathcal{Y}$ that reconstructs Y from a concatenated low-dimensional space of the latent representation codes S and Z. Specifically, we use the following linear decoding function:

$$\hat{Y} = h(S, Z) = SA_s + ZA_z + \mathbf{1}\mathbf{a}^\top \tag{5}$$

where $A_s \in \mathbb{R}^{m\times k}$, $A_z \in \mathbb{R}^{n\times k}$ and $\mathbf{a} \in \mathbb{R}^k$ are the decoder parameters, and $\hat{Y}$ is the reconstructed data matrix in the original output label space. We will minimize a decoding loss $\ell_d(Y, \hat{Y})$ between the original Y and the reconstructed

$\hat{Y}$ in the training process. Since the latent representation matrices, S and Z, will be simultaneously induced in the training process, we expect the latent S and Z identified will complement each other in this final decoder function to accurately reconstruct the output label matrix Y. The encoding and decoding process between the output layer and the mid-level representation layer can be viewed as an augmented autoencoder.

3.2 Optimization Formulation

Given the framework introduced above, we formulate the bi-directional model training as a joint optimization problem over all the model parameters that minimizes a regularized linear combination of the prediction loss and the two decoding losses:

$$\min_{W_x,\mathbf{b}_x,W_y,\mathbf{b}_y,A_s,A_z,\mathbf{a},U,\mathbf{d},\Omega,\mathbf{q}} \quad \mathcal{L}(X,Y) \tag{6}$$

such that

$$\begin{aligned}\mathcal{L}(X,Y) = \ell_p(Z,\hat{Z}) + \eta\ell_d(Y,\hat{Y}) + \rho\ell_d(X,\hat{X}) + \\ \alpha_x\mathcal{R}(W_x) + \alpha_y\mathcal{R}(W_y) + \alpha_u\mathcal{R}(U) + \alpha_s\mathcal{R}(A_s) + \alpha_z\mathcal{R}(A_z) + \alpha_o\mathcal{R}(\Omega)\end{aligned} \tag{7}$$

where the trade-off parameters η and ρ are used to adjust the relative degrees of focus on the three different loss terms; all the other α_* trade-off parameters adjust the degrees of regularization over the model parameter matrices; $\mathcal{R}(\cdot)$ denotes the regularization function. Note with the encoding parameters, $(W_x, \mathbf{b}_x)$ and $(W_y, \mathbf{b}_y)$, the latent representation matrices S and Z are directly available through the nonlinear functions in Eq. (1) and Eq. (2) respectively. This objective function expresses the following properties we expect from the latent representations: 1) S and Z should be low-dimensional (enforced by the encoding model parameters); 2) S should preserve as much information as possible from X (enforced by $\ell_d(X,\hat{X})$); 3) Z should preserve information from Y that is complementary to S for the reconstruction of Y (enforced by $\ell_d(Y,\hat{Y})$), while being predictable from X (enforced by $\ell_p(Z,\hat{Z})$); and 4) the concatenation of S and Z should be discriminative for the target output label matrix Y (enforced by $\ell_d(Y,\hat{Y})$).

To produce a concrete training problem, we use least squares loss functions for both the prediction loss and the two decoding losses, such that

$$\ell_p(Z,\hat{Z}) = \|Z-\hat{Z}\|_F^2 = \|Z-f(X)\|_F^2 \tag{8}$$

$$\ell_d(X,\hat{X}) = \|X-\hat{X}\|_F^2 = \|X-g(S)\|_F^2 \tag{9}$$

$$\ell_d(Y,\hat{Y}) = \|Y-\hat{Y}\|_F^2 = \|Y-h(S,Z)\|_F^2 \tag{10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. We use the square of Frobenius norm as the regularization function $\mathcal{R}$ over the parameter matrices, such that $\mathcal{R}(\cdot) = \|\cdot\|_F^2$.

Moreover, among the three loss terms in the objective function (7), the decoding loss $\ell_d(X,\hat{X})$ in (9) is used to ensure the input data can be reconstructed

from its low-dimensional representation S with a small error by using the decoder function $g(\cdot)$. The decoder function $g(\cdot)$ is not directly involved in the overall target label prediction process, since S can be produced from the original input matrix X with the encoding function. The necessity of having this decoding component in the framework, hence the decoding loss $\ell_d(X, \hat{X})$ in the optimization objective, can be questioned. In our empirical study later, we investigated this issue by dropping $g(\cdot)$, hence $\ell_d(X, \hat{X})$ and $\mathcal{R}(U)$, from the optimization problem. Our results suggest the decoder component $g(\cdot)$ is useful and the decoding loss $\ell_d(X, \hat{X})$ should be included in the learning process.

3.3 Optimization Algorithm

The minimization problem in (6) involves two sets of parameters, the encoder parameters, $\{W_x, \mathbf{b}_x, W_y, \mathbf{b}_y\}$, and the decoder and prediction model parameters, $\{A_s, A_z, \mathbf{a}, U, \mathbf{d}, \Omega, \mathbf{q}\}$. We develop an iterative optimization algorithm that conducts optimization over these two groups of model parameters in an alternating way in each iteration.

We first randomly initialize the model parameters. Then in each iteration, we perform the following two steps.

Step I. In this step, given the current encoder parameters, $\{W_x, \mathbf{b}_x, W_y, \mathbf{b}_y\}$, to be fixed, we optimize the decoder and prediction model parameters to minimize the objective function in (7). We first compute the latent matrices S and Z according to Eq. (1) and Eq. (2) respectively. Given S and Z, the joint minimization problem in (6) can be decomposed into the following sub-optimization problems over the decoder and prediction model parameters:

$$\min_{A_s, A_z, \mathbf{a}} \ \eta \|Y - h(S, Z)\|_F^2 + \alpha_s \|A_s\|_F^2 + \alpha_z \|A_z\|_F^2 \tag{11}$$

$$\min_{\Omega, \mathbf{q}} \ \|Z - f(X)\|_F^2 + \alpha_o \|\Omega\|_F^2 \tag{12}$$

$$\min_{U, \mathbf{d}} \ \rho \|X - g(S)\|_F^2 + \alpha_u \|U\|_F^2 \tag{13}$$

which have the following three sets of closed-form solutions respectively:

$$\left.\begin{array}{l} A_s = (S^\top H S + \frac{\alpha_s}{\eta} I_m)^{-1} S^\top H (Y - Z A_z) \\ A_z = (Z^\top H Z + \frac{\alpha_z}{\eta} I_n)^{-1} Z^\top H (Y - S A_s) \\ \mathbf{a} \ = \frac{1}{t} (Y - S A_s - Z A_z)^\top \mathbf{1} \end{array}\right\} \tag{14}$$

$$\left.\begin{array}{l} \Omega = (X^\top H X + \alpha_o I_d)^{-1} X^\top H Z \\ \mathbf{q} \ = \frac{1}{t} (Z - X\Omega)^\top \mathbf{1} \end{array}\right\} \tag{15}$$

$$\left.\begin{array}{l} U = (S^\top H S + \frac{\alpha_u}{\rho} I_m)^{-1} S^\top H X \\ \mathbf{d} \ = \frac{1}{t} (X - S U)^\top \mathbf{1} \end{array}\right\} \tag{16}$$

where $H = I_t - \frac{1}{t} \mathbf{1}\mathbf{1}^\top$ is a centering matrix of size t.

Algorithm 1 The Bi-Directional Learning Algorithm

1: **Initialize** all the model parameters.
2: **repeat**
3: **Step I**: given current encoder parameters $\{W_x, \mathbf{b}_x, W_y, \mathbf{b}_y\}$, update the decoder and prediction model parameters, $\{A_s, A_z, \mathbf{a}, U, \mathbf{d}, \Omega, \mathbf{q}\}$, with closed-form solutions in Eq. (14)–(16).
4: **Step II**: given current $\{A_s, A_z, \mathbf{a}, U, \mathbf{d}, \Omega, \mathbf{q}\}$, perform optimization over encoder parameters $\{W_x, \mathbf{b}_x, W_y, \mathbf{b}_y\}$ to minimize the objective $\mathcal{J}$ in (18) using gradient descent with line search.
5: **until** Convergence or maximum number of iterations is reached

Step II. In this step, given the current decoder and prediction model parameters, $\{A_s, A_z, \mathbf{a}, U, \mathbf{d}, \Omega, \mathbf{q}\}$, to be fixed, we optimize the encoder parameters, $\{W_x, \mathbf{b}_x, W_y, \mathbf{b}_y\}$, to minimize the objective function in (7). This leads to the following minimization problem:

$$\min_{W_x, \mathbf{b}_x, W_y, \mathbf{b}_y} \quad \ell_p(Z, \hat{Z}) + \eta \ell_d(Y, h(S, Z)) + \rho \ell_d(X, g(S)) + \alpha_x \mathcal{R}(W_x) + \alpha_y \mathcal{R}(W_y) \tag{17}$$

where $\hat{Z}$ is pre-computed with the fixed parameters via Eq. (3). By expressing S and Z in terms of the encoder functions in Eq. (1) and Eq. (2), the objective function in (17) can be written as

$$\begin{aligned} \mathcal{J} = & \|\hat{Z} - \sigma(YW_y + \mathbf{1}\mathbf{b}_y^\top)\|_F^2 \\ & + \eta \|\sigma(XW_x + \mathbf{1}\mathbf{b}_x^\top)A_s + \sigma(YW_y + \mathbf{1}\mathbf{b}_y^\top)A_z + \tilde{Y}\|_F^2 \\ & + \rho \|\sigma(XW_x + \mathbf{1}\mathbf{b}_x^\top)U + \tilde{X}\|_F^2 + \alpha_x \|W_x\|_F^2 + \alpha_y \|W_y\|_F^2 \end{aligned} \tag{18}$$

where $\tilde{Y}$ and $\tilde{X}$ are defined as

$$\tilde{Y} = \mathbf{1}\mathbf{a}^\top - Y, \qquad \tilde{X} = \mathbf{1}\mathbf{d}^\top - X. \tag{19}$$

We use a gradient descent algorithm with line search [17] to solve the minimization problem in (17), which requires computing the gradients of the objective function $\mathcal{J}$ regarding the decoder parameters $\{W_x, \mathbf{b}_x, W_y, \mathbf{b}_y\}$.

The overall optimization algorithm for training the bi-directional model is summarized in Algorithm 1.

Test Phase: In the test phase, given a new instance $\mathbf{x}$, we first produce the latent representations, $\mathbf{s} = \sigma(\mathbf{x}W_x + \mathbf{1}\mathbf{b}_x^\top)$ and $\mathbf{z} = f(\mathbf{x})$. Then the final output can be computed by $\mathbf{y} = h(\mathbf{s}, \mathbf{z})$. The labels of $\mathbf{x}$ can be determined by simply rounding the entries of $\mathbf{y}$ to 0s or 1s.

4 Experimental Results

Data Sets. To evaluate the proposed bi-directional model for multi-label classification, we conducted experiments on 5 different types of real-world data sets:

Table 1. The statistic information of the data sets used in the experiments.

Data set	*corel5k*	*delicious*	*yeast*	*genbase*	*mirflickr*
Num. of Instances	4999	5000	2417	662	5000
Num. of Features	512	500	103	1186	512
Num. of Labels	209	918	14	15	38
Label Cardinality	3.32	18.72	4.24	1.16	4.71

two image data sets (*corel5k* [3] and *mirflickr* [13]), one text set (*delicious* [20]), and two biology data sets (*yeast* [4] and *genbase* [2]). For each big data set with more than $10k$ instances, we randomly sampled a 5000-instance subset to use. We conducted experiments with 5-fold cross-validation and dropped the labels that do not appear in at least one of the five fold partitions. The statistic information of all 5 data sets used is summarized in Table 1, where label cardinality denotes the average number of labels assigned to each instance.

Methods. We compared the proposed bi-directional multi-label learning method with the following multi-label learning methods:

- *Binary relevance* (BR). This baseline method decomposes multi-label classification into a set of independent binary classification problems via the one-vs-all scheme. We used linear SVM classifiers for the binary problems.
- *Multi-label Output Codes using CCA* (MOC-CCA) [24]. This method performs error-correcting coding for the labels based on canonical correlation analysis (CCA).
- *Multi-Label Learning using Local Correlation* (ML-LOC) [12]. Instead of assuming global label correlations, this method separates instances into different groups and allows label correlations to be exploited locally.
- *Calibrated Separation Ranking Loss* (CSRL) [6]. This method performs large margin multi-label learning based on a novel loss function.

Experimental Setting. In each iteration of the 5-fold cross-validation, the training set is further randomly divided into two parts for parameter selection: 80% for model training and 20% for parameter evaluation. For the proposed method, there are a number of parameters need to be determined. We fixed the regularization parameters as relatively small values, such as $\alpha_x = \alpha_y = \alpha_o = 0.05$, $\alpha_s = \alpha_z = 0.05\eta$, and $\alpha_u = 0.05\rho$. The trade-off parameters, ρ and η, and the dimensions of latent representations, m and n, are automatically selected in the learning phase. The values of ρ and η are both selected from $\{0.1, 1, 10, 100\}$. The candidate values for m and n however vary across data sets since the feature and label dimensions are different for different data sets. We used $m \in \{20, 50, 100\}$ and $n \in \{20, 60, 80\}$ on *corel5k*; used m, $n \in \{20, 50, 100\}$ on *delicious*; used $m \in \{20, 50\}$ and $n \in \{5, 10\}$ on *yeast*; used $m \in \{20, 50, 100\}$ and $n \in \{5, 10\}$ on *genbase*; and used $m \in \{20, 50, 100\}$ and $n \in \{5, 15, 25\}$ on *mirflickr*. For the comparison methods, we performed parameter selection using the same scheme. For *BR*, the trade-off parameter C is selected from

Table 2. The average and standard deviation results in terms of Hamming Loss(%). Lower values indicate better performance.

Data set	BR	MOC-CCA	ML-LOC	CSRL	Proposed
corel5k	0.9 ± 0.03	0.6 ± 0.02	0.6 ± 0.01	**0.4 ± 0.02**	**0.4 ± 0.02**
delicious	7.2 ± 0.04	10.3 ± 0.06	17.5 ± 0.07	4.5 ± 0.03	**3.9 ± 0.03**
yeast	8.6 ± 0.03	4.1 ± 0.03	14.2 ± 0.05	6.7 ± 0.04	**4.1 ± 0.04**
genbase	0.6 ± 0.01	0.5 ± 0.02	1.2 ± 0.01	0.5 ± 0.03	**0.3 ± 0.01**
mirflickr	3.2 ± 0.02	3.1 ± 0.04	8.1 ± 0.05	**2.3 ± 0.03**	2.5 ± 0.02

Table 3. The average and standard deviation results in terms of Macro-F1 measure(%). Larger values indicate better performance.

Data set	BR	MOC-CCA	ML-LOC	CSRL	Proposed
corel5k	2.1 ± 0.02	5.6 ± 0.01	3.9 ± 0.02	5.3 ± 0.05	**8.0 ± 0.02**
delicious	6.5 ± 0.04	13.9 ± 0.04	10.0 ± 0.09	14.1 ± 0.03	**15.3 ± 0.02**
yeast	30.1 ± 0.14	38.1 ± 0.17	39.7 ± 0.18	39.3 ± 0.12	**40.6 ± 0.15**
genbase	46.4 ± 0.08	61.3 ± 0.05	55.1 ± 0.07	61.5 ± 0.12	**63.8 ± 0.05**
mirflickr	18.1 ± 0.14	22.5 ± 0.19	21.2 ± 0.16	24.9 ± 0.20	**25.6 ± 0.13**

$\{0.1, 1, 10, 50, 100\}$; for *MOC-CCA*, the number of canonical components d is selected from $[1, \min(\#\text{features}, \#\text{labels})]$ and the trade-off parameter λ is selected from $\{0.25, 1, 4\}$; for *ML-LOC*, the parameters are selected as $\lambda_1 \in \{0.1, 1, 10\}$, $\lambda_2 \in \{1, 10, 100\}$, and $m \in \{10, 15, 20\}$; for CSRL, the trade-off parameter C is selected from $\{0.1, 1, 10\}$.

4.1 Multi-label Classification Results

We evaluated the performance of each comparison method with three criteria: *Hamming Loss*, *Macro-F1*, and *Micro-F1*. These three criteria measure the multi-label classification performance from different aspects. The 5-fold cross validation results for all comparison methods over the five data sets are reported in Table 2 – Table 4 in terms of the three evaluation criteria respectively. Both the average result values and their standard deviations are reported.

From the results in Table 2, we can see that the multi-label comparison method *MOC-CCA* does not show consistent advantages over the baseline *BR* in terms of hamming loss, *ML-LOC* even has inferior performance on most data sets comparing to *BR*, while *CSRL* and the proposed approach outperform *BR* across all data sets. Moreover, the proposed approach produces the best results among all the comparison methods on four out of the total five data sets. Hamming loss however may prefer extreme prediction results without balancing the prediction recall and precision. Table 3 presents the comparison results in terms of Macro-F1 score which takes both prediction recall and precision into account. In terms of Macro-F1, the proposed approach consistently outperforms all the other methods across all the five data sets. In particular, on *corel5k*, the proposed

Table 4. The average and standard deviation results in terms of Micro-F1 measure(%). Larger values indicate better performance.

Data set	BR	MOC-CCA	ML-LOC	CSRL	Proposed
corel5k	7.6 ± 0.09	**11.1 ± 0.10**	9.3 ± 0.14	10.6 ± 0.10	**11.1 ± 0.08**
delicious	16.5 ± 0.07	**23.8 ± 0.07**	16.1 ± 0.08	22.9 ± 0.03	23.2 ± 0.03
yeast	52.4 ± 0.22	64.8 ± 0.20	67.7 ± 0.25	60.1 ± 0.16	**68.2 ± 0.17**
genbase	54.8 ± 0.12	**71.4 ± 0.17**	71.1 ± 0.21	65.3 ± 0.17	70.9 ± 0.18
mirflickr	24.1 ± 0.45	32.1 ± 0.49	36.7 ± 0.69	31.5 ± 0.32	**36.8 ± 0.25**

method produces incredible improvement over the other comparison methods. It improves the performance of *MOC-CCA* by more than 40%, improves the performance of *CSRL* by more than 50%, and improves the performance of *ML-LOC* by more than 100%. Moreover, all the four multi-label learning methods greatly outperform the baseline *BR* across all the data sets in terms of Macro-F1. *Table 4* presents the comparison results in terms of Micro-F1 score. We can see that the four multi-label learning methods again outperform the baseline *BR* across all the data sets, except on *delicious* where *ML-LOC* has slightly inferior result. Among the multi-label learning methods, the proposed approach produces the best results on three data sets, while presenting results very close to the best ones on the remaining two data sets. It demonstrates more consistent good performance across different types of data sets. All these results suggest our proposed bi-directional model is effective for multi-label classification by capturing complementary information from both inputs and outputs.

4.2 Study of the Bi-Directional Model

To gain a deeper understanding over the novel bi-directional learning scheme, we have also conducted experiments to investigate the influence of different components in the proposed bi-directional model.

First, we investigated the capacity of the bi-directional model by comparing the full model to its two essential one-directional components, the bottom-up component and the top-down component. The mid-level representation of the bi-directional full model has two complementary parts, S and Z. The bottom-up component and the top-down component consider solely the feature-sourced mid-level representation S and the label-sourced mid-level representation Z respectively, by deactivating the other component. We denote the full model with $S + Z$ and denote the two component models with S and Z respectively. Parameter selection for the two component models is conducted using the same procedure introduced before. The comparison results for these three models are reported in Table 5. We can see that the two one-directional component models have strengths on different data sets. The label-sourced one-directional model Z performs better than the feature-sourced one-directional model S on *corel5k* and *delicious* where the label space is large, while model S performs better on the other three data sets. Nevertheless, the proposed bi-directional model greatly

Table 5. Comparison results over the bi-directional model and its two one-directional components. $S+Z$ denotes the proposed bi-directional model, S denotes the bottom-up one-directional model and Z denotes the top-down one-directional model.

Measure	Method	*corel5k*	*delicious*	*yeast*	*genbase*	*mirflickr*
Hamming Loss	$S+Z$	**0.4 ± 0.02**	**3.9 ± 0.03**	**4.1 ± 0.04**	**0.3 ± 0.01**	**2.5 ± 0.02**
	S	0.7 ± 0.03	5.0 ± 0.05	5.8 ± 0.03	0.5 ± 0.04	2.9 ± 0.05
	Z	0.6 ± 0.05	4.4 ± 0.02	7.2 ± 0.02	2.4 ± 0.05	2.6 ± 0.05
Macro-F1	$S+Z$	**8.0 ± 0.02**	**15.3 ± 0.02**	**40.6 ± 0.15**	**63.8 ± 0.05**	**25.6 ± 0.13**
	S	4.1 ± 0.05	8.8 ± 0.04	34.2 ± 0.12	48.4 ± 0.17	18.2 ± 0.18
	Z	6.2 ± 0.05	11.8 ± 0.03	26.6 ± 0.12	44.1 ± 0.10	19.1 ± 0.13
Micro-F1	$S+Z$	**11.1 ± 0.08**	**23.2 ± 0.03**	**68.2 ± 0.17**	**70.9 ± 0.18**	**36.8 ± 0.25**
	S	8.5 ± 0.12	17.1 ± 0.05	60.2 ± 0.11	58.3 ± 0.14	28.7 ± 0.21
	Z	9.8 ± 0.10	21.8 ± 0.06	55.6 ± 0.14	50.5 ± 0.17	28.4 ± 0.44

outperforms both one-directional component models across all the five data sets in terms of all the three measures. This suggests that the proposed bi-directional model can successfully integrate the strengths of its one-directional components in a complementary way and has capacity of capturing useful information from both the input and output spaces.

Next, we have also compared the proposed approach with two of its alternative versions: one drops the decoder component $g(\cdot)$ and is denoted as "*Proposed w/o g*"; the other removes the nonlinear $\sigma(\cdot)$ function from encoding and uses linear encoder functions, which is denoted as "*Proposed w/o* σ". We conducted the comparison experiment using varying latent dimension sizes, m and n, on the *corel5k* data set. We first set $n = 20$ and studied the performance of the three methods by varying the m value within the set $\{20, 50, 100, 300\}$. Then we set $m = 50$ and vary the n value within the set $\{20, 60, 100, 140\}$. The experimental results are presented in Figure 2, in terms of the logarithm of the three evaluation criteria. We can see that the proposed full model clearly outperforms the other two variants across all learning scenarios, which suggests that both the decoder component $g(\cdot)$ and the nonlinear encoders are important in our bi-directional model. From Figure 2 (a)-(c), we can see that with fixed n value, the performance of all the three methods becomes stable when the m value reaches 50. This suggests that with even a reasonably small m value such as $m = 50$, our model can already capture the intrinsic information from the input data. On the other hand, from Figure 2 (d)-(f), we can see that the performance of all the three methods deteriorates when the n value becomes larger than 60. This suggests that if the latent representation size of the output labels is too large, noise may be introduced in augmenting the latent component produced from the input data and hence harm the performance.

In summary, all these experimental results demonstrated the compactness and effectiveness of the novel bi-directional model for multi-label classification.

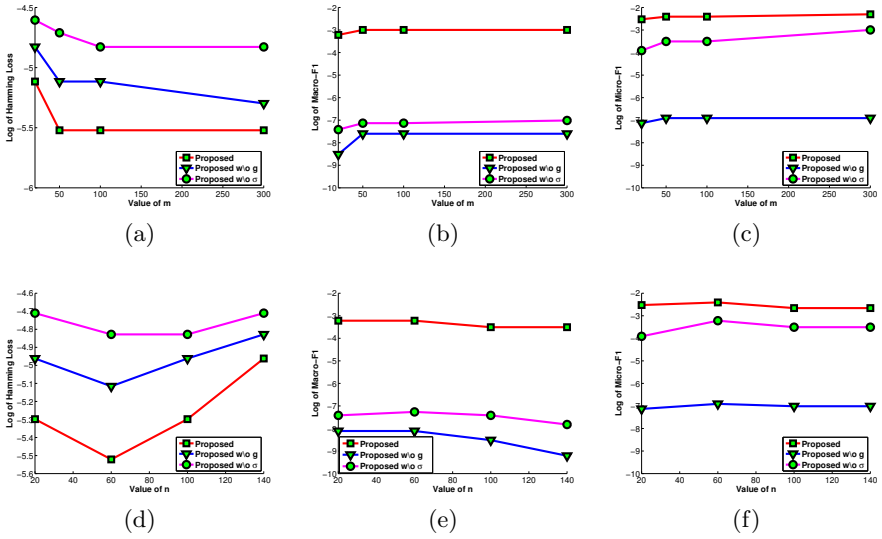

Fig. 2. Impact of the latent dimension sizes m and n over the model performance on *corel5k*. In the top row, n is fixed to 20, and m varies from 20 to 300. In the bottom row, m is fixed to 50, and n varies from 20 to 140. The first column shows the results in terms of log hamming loss, the middle column shows the results in terms of log Macro-F1, and the right column shows the results in terms of log Micro-F1.

5 Conclusion

In this paper, we proposed a novel bi-directional representation learning model for multi-label classification, which has a two-part latent representation layer constructed from both the input data and the output labels. The two latent parts augment each other by integrating information from both the feature and label spaces to enhance the overall multi-label classification. We formulated multi-label learning over this model as a joint minimization problem over all parameters of the component functions, and developed an iterative optimization algorithm with alternating steps to perform optimization. We conducted experiments on five real world multi-label data sets by comparing the proposed method to a number of previously developed multi-label classification methods and a few variant models. Our experimental results suggest that our proposed model is compact and showed that it outperformed all the other comparison methods.

References

1. Chen, Y., Lin, H.: Feature-aware label space dimension reduction for multi-label classification. In: Proceedings of NIPS (2012)
2. Diplaris, S., Tsoumakas, G., Mitkas, P., Vlahavas, I.: Protein classification with multiple algorithms. In: Proceedings of Advances in Informatics (2005)

3. Duygulu, P., Barnard, K., Freitas, J., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Proceedings of ECCV (2002)
4. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Proceedings of NIPS (2001)
5. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Advances in Knowledge Discovery and Data Mining (2004)
6. Guo, Y., Schuurmans, D.: Adaptive large margin training for multilabel classification. In: Proceedings of AAAI (2011)
7. Guo, Y., Schuurmans, D.: Multi-label classification with output kernels. In: Proceedings of ECML (2013)
8. Guo, Y., Xue, W.: Probablistic mult-label classification with sparse feature learning. In: Proceedings of IJCAI (2013)
9. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science 313, 504–507 (2006)
10. Ho, T.: The random subspace method for constructing decision forests. IEEE TPAMI 20(8) (Aug 1998)
11. Hsu, D., Kakade, S., Langford, J., Zhang, T.: Multi-label prediction via compressed sensing. In: Proceedings of NIPS (2009)
12. Huang, S., Zhou, Z.: Multi-label learning by exploiting label correlations locally. In: Proceedings of AAAI (2012)
13. Huiskes, M., Lew, M.: The MIR Flickr retrieval evaluation. In: Proceedings of ACM MIR (2008)
14. Ji, S., Tang, L., Yu, S., Ye, J.: Extracting shared subspace for multi-label classification. In: Proceedings of KDD (2008)
15. Lastra, G., Luaces, O., Quevedo, J., Bahamonde, A.: Graphical feature selection for multilabel classification tasks. In: Proceedings of IDA (2011)
16. Lewis, D., Yang, Y., Rose, T., Li, F.: RCV1: A new benchmark collection for text categorization research. JMLR 5, 361–397 (2004)
17. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (2006)
18. Sharmanska, V., Quadrianto, N., Lampert, C.: Augmented attribute representations. In: Proceedings of ECCV (2012)
19. Tai, F., Lin, H.: Multilabel classification with principal label space transformation. In: Proceedings of Inter. Workshop on Learning from Multi-Label Data (2010)
20. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels. In: ECML/PKDD Workshop on Mining Multidimensional Data (2008)
21. Yan, R., Tesic, J., Smith, J.: Model-shared subspace boosting for multi-label classification. In: Proceedings of KDD (2007)
22. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: Proceedings of the Annual ACM SIGIR Conference (2005)
23. Zhang, M., Peña, J., Robles, V.: Feature selection for multi-label naive bayes classification. Inf. Sci. 179(19) (Sep 2009)
24. Zhang, Y., Schneider, J.: Multi-label output codes using canonical correlation analysis. In: Proceedings of AISTATS (2011)
25. Zhang, Y., Schneider, J.: Maximum margin output coding. In: Proceedings of ICML (2012)
26. Zhang, Y., Zhou, Z.: Multilabel dimensionality reduction via dependence maximization. In: Proceedings of AAAI (2008)
27. Zhou, T., Tao, D., Wu, X.: Compressed labeling on distilled lablsets for multi-label learning. Machine Learning 88, 69–126 (2012)