



# An innovative multi-label learning based algorithm for city data computing

Mengqing Mei<sup>1</sup> · Yongjian Zhong<sup>1</sup> · Fazhi He<sup>1</sup> · Chang Xu<sup>2</sup>

Received: 12 March 2019 / Revised: 2 September 2019 / Accepted: 10 October 2019 /

Published online: 6 January 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Investigating correlation between example features and example labels is essential to the solving of classification problems. However, identification and calculation of the correlation between features and labels can be rather difficult in case involving high-dimensional multi-label data. Both feature embedding and label embedding have been developed to tackle this challenge, and a shared subspace for both labels and features is usually learned by applying existing embedding methods to simultaneously reduce the dimension of features and labels. By contrast, this paper suggests learning separate subspaces for features and labels by maximizing the independence between the components in each subspace, as well as maximizing the correlation between these two subspaces. The learned independent label components indicate the fundamental combinations of labels in multi-label datasets, which thus helps to reveal the correlation between labels. Furthermore, the learned independent feature components lead to a compact representation of example features. The connections between the proposed algorithm and existing embedding methods are discussed in detail. Experimental results on real-world multi-label datasets demonstrate that it is necessary for us to explore independent components from multi-label data, and further prove the effectiveness of the proposed algorithm.

**Keywords** Multi-label · Independent components analysis · Embedding · Canonical correlation

---

Mengqing Mei and Yongjian Zhong contributed equally.

✉ Fazhi He  
fzhe@whu.edu.cn

Mengqing Mei  
727849936@qq.com

Yongjian Zhong  
yjzhong@whu.edu.cn

Chang Xu  
c.xu@sydney.edu.au

<sup>1</sup> School of Computer Science, Wuhan University, Wuhan, China

<sup>2</sup> UBTech Sydney AI Centre, School of Computer Science, Faculty of Engineering and IT, University of Sydney, Sydney, Australia

## 1 Introduction

In real-world applications such as text categorization and medical diagnosis, an instance often has multiple labels, which can lead to a multi-label classification problem arising. For example, a given image may be labeled as ‘sky’, ‘tree’ and ‘mountain’. In recent years, this multi-label classification problem has drawn increasing attention from researchers and has been widely studied in many fields, e.g. multimedia annotation [9, 27, 36], web mining [28, 28, 33], and tag recommendation [1, 19].

The multi-label classification problem can be treated as a generalization of traditional multi-class classification. The major difference between the two is that labels are mutually exclusive in multi-class classification; that is to say, an instance can only belong to one class while multi-label classification focuses on instances that could simultaneously have more than one label. Independently handling each label through a classical single-label classification would theoretically be straightforward; in practice, however, connections between labels often exist. For example, the labels ‘tree’ and ‘mountain’ usually appear simultaneously in an image. Investigating the correlation between labels has been demonstrated to be effective for improving the performance of multi-label classification, and many methods have been devised to capture label correlation. For example, pruning the label set to distill the most important label relationship [25, 29], including predicted labels as auxiliary features to build up classifier chains [26], and applying the maximum margin strategy to deal with multi-label data (Rank-SVM) [10].

Due to the rapid development of the internet and social media, a large number of labels are often associated with a single instance. For example, in image tagging, the number of tags can easily go beyond tens or even hundreds of thousands. There are millions of categories on Amazon, but any new product has to be assigned to only a small number of relevant categories before it goes on the website. The complexity of traditional methods is usually increased as the number of labels in the dataset also increases, which makes the handling of a large number of labels infeasible under most circumstances. Moreover, in a large-scale multi-label dataset, the dimensions of example features also tend to be large. Many approaches have been developed to tackle large-scale multi-label classification problems; these approaches aim to capture label correlation and tackle multi-label classification from different points of view. The major idea behind embedding methods concerns learning a low-dimensional subspace of the labels or features. Compared with one-vs-all methods, these approaches can achieve significant speed-up. Embedding methods can be roughly grouped into two subcategories; namely, FSDR (feature space dimension reduction) and LSDR (label space dimension reduction).

FSDR involves learning the subspace of example features, e.g. using locality-preserving projections to reduce the number of feature dimensions [13, 37, 42], while LSDR acts on example labels, e.g. PLST [31], which discovers the label subspace using Principal Component Analysis (PCA). Instead of separately investigating the subspaces of features and labels in classical FSDR and LSDR, some works have integrated both feature and label information, e.g. label-aware FSDR and feature-aware LSDR. MDDM [42] aims to find a low-dimensional feature subspace with a maximal dependence on the labels. As a least square extension of Canonical Correlation Analysis (CCA), LS-CCA [30] learns a feature subspace under the supervision of labels. Moreover, Chen et al. *et al.* [8] proposed the conditional PLST, which is a feature-aware extension of PLST.

Label-aware FSDR and feature-aware LSDR can be used effectively to integrate feature and label information. However existing embedding methods generally rely on a single subspace, which is discovered from feature space or label space, or shared by the features

and labels. In practice, while redundancy exists in both features and labels, investigating feature and label information in a shared subspace does not always yield accurate results; moreover, although an instance can have a number of labels, the correlation between these labels would naturally lead to limiting the possible types of label combinations. However, independent component of labels has not been fully investigated.

In this paper, we propose to combine the advantages of FSDR and LSDR through the learning of separated subspaces for features and labels. By maximizing the independence between components in the label subspace, we discover label correlation represented by independent label components. Furthermore, independent feature components are extracted from example features so that independent coefficients can provide a compact representation of examples. We further maximize the correlation between label subspace and feature subspace via a regression problem. As several existing methods can be regarded as special cases of the proposed algorithm, their connections are thoroughly analysed in order to reveal the advantages of the proposed framework. In addition, we conduct comprehensive experiments on real-world datasets. The experimental results demonstrate that the proposed algorithm can effectively discover independent components from multi-label data, and thus bring about classification performance improvement. Furthermore, our method can be extended to handle non-linear independent components of features and labels; we employ a powerful non-linear neural network to achieve this extension.

## 2 Related work

Embedding methods are popular approaches to tackling the multi-label classification problem due to their simplicity, ease of implementation, and ability to handle label correlation. In this section, we will first introduce some classic embedding-based approaches, after which we will review some works that have successfully improved these conventional embedding approaches.

The most basic multi-label classification method is binary relevance [40], which independently trains a classifier for each label. Its primary advantage in multi-label learning is efficiency. However, since multi-label learning techniques have diverse applications, pure BR cannot achieve good performance in many specific applications, especially when features and labels are high dimensional. Accordingly, some FSDR and LSDR methods were proposed to tackle this problem. Wang et al. proposed ML-LDA [34], which is a classical Linear Discriminant Analysis (LDA) method for multi-label classification. These authors redefined the scatter matrices to enable their approach to adapt to both single-label and multi-label situations. Zhang et al. proposed MDDM [42] which aimed to identify a low-dimensional feature subspace and maximize the dependence between feature subspace and label space using the Hilbert-Schmidt independent Criterion as their measurement of dependence. Yu et al. introduced a supervised Latent Semantic Indexing (LSI) approach to multi-label classification [39]. This method mapped the original feature into a low-dimensional subspace that retains the feature information while also capturing the label dependency. Jian et al. proposed a feature selection method known as MIFS [17], which first mapped the label information into a low-dimensional subspace that was later used to guide the feature selection phase. Hsu [14] introduced the LSDR paradigm, which finds the latent subspace of labels using random projection. Moreover, Tai and Lin [31] replaced the random transformation with Principal Component Analysis (PCA), and thereby proposed the principal label space transformation (PLST) for multi-label classification.

Conventional LSDR and FSDR methods focus on the latent space of features or labels. It is widely accepted that using either label information or feature information alone precludes a full investigation of the multi-label data. Consequently, many feature-aware LSDR and label-aware FSDR methods, retain both feature and label information, have been devised. The boundary between LSDR and FSDR has thus become ambiguous; accordingly, we categorize both of these approaches as embedding-based methods in the remainder of our paper.

Zhang & Schneider [41] applied Canonical Correlation Analysis to embed features and labels into a lower-dimensional vector. A codeword was subsequently generated by concatenating the original label vector with the lower-dimensional vector; for a given data point, its new codeword was predicted using Random Forest, although one can choose other regressors in practice. Bayesian Inference was then used to map the codeword to the label distribution. Chen and Lin proposed an enhanced version of PLST [31], known as conditional principal component transformation (CPLST). This approach combines the concepts of PLST and Canonical Correlation Analysis (CCA), thereby improving PLST through taking feature information into consideration.

Many embedding methods factorize the label matrix to a low-dimensional matrix as a low-rank approximation. In real-world applications, however, the label matrix is usually not low-rank due to the existence of tail labels. Rather than striving for global projection, Bhatia developed a method named SLEEC [5], which aimed to preserve the pairwise distances between the closest label vectors with local embedding. SLEEC proposed a novel objective function for preserving the local information of labels while also ensuring recoverability. During prediction, SLEEC used a  $k$ -nearest neighbor ( $k$ NN) classifier. Xu et al. [38] proposed a robust extreme multi-label classification approach by treating tail labels as additive noise on true label distribution. Moreover, while all previous embedding methods found a continuous subspace, Zhou et al. [43] were the first to introduce an embedding method with binary subspace. These authors found a lower-dimensional embedding that minimizes the residual error, but forced their low-dimensional vectors to be binary. Due to the binary embedding constraints, classification is applicable in the learning part rather than regression, which significantly accelerates their method.

Neural networks have long been known to be powerful non-linear representation learning tools. For BP-MALL, the first method to utilize neural network architectures to tackle the multi-label classification problem, a novel loss function was devised to exploit the dependency across labels. Subsequently, some more recent works utilizing deep neural network techniques have been proposed. For example, CNN-RNN is a method that learns a label embedding space while capturing label co-occurrence information via a recurrent neural network. To further utilize the correlation between feature and label, Yeh et al. proposed C2AE, which utilizes deep canonical correlation analysis and an autoencoder to learn a latent feature-aware subspace for multi-label classification.

The purpose of our method is to learn an independent representation to better tackle the multi-label classification problem. The benefits of independent representation have been noted in many previous papers. Le et al. proposed RICA [21] to efficiently discover the independent overcomplete representation for computer vision problems. Dinh et al. proposed a method called non-linear independent components estimator (NICE), which adopts a neural network architecture that allows the determinant of the Jacobian matrix to be efficiently computed, and also reveals that independent representation can improve performance in many applications. There are also some methods that have tried to generate independent representations [4, 6]. RICA is a linear model; while NICE is based on a neural network. However, NICE requires special architecture which restrains its capacity. By contrast, our

proposed DICE adds a regularization term that can be efficiently optimized. In fact, there is an approach named disentangled representation learning that assumes the data is generated from independent factors of variation [7, 32]. Recently, Kageback et al. proposed a method that contains only the  $\ell_1$  norm of the sample covariance matrix, which encourages the decorrelation [18]. Meanwhile, the combination of canonical correlation and multiple autoencoders is not rare in the field of multi-view learning, since learning a correlative subspace is essential for multi-view learning. Thus, Wang et al. proposed DCCAE [35], which consists of multiple autoencoders for each view and correlation loss for the representations of two views.

### 3 The proposed method

In this section, we describe our proposed approach by firstly introducing our motivation and an overview of our approach, after which technical details are provided.

#### 3.1 Preliminaries

Let  $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$  be the training data, where  $x_i \in \mathbb{R}^d$  is the feature vector for the  $i$ th multi-label example and  $y_i \in \{0, 1\}^l$  is its corresponding label. We denote  $X \in \mathbb{R}^{N \times d}$  as the training sample matrix, and  $Y \in \{0, 1\}^{N \times l}$  is the label matrix, where  $x_i$  is the  $i$ -th row of  $X$  and  $y_j$  is the  $j$ -th row of  $Y$ .

Taking FSDR as an example, the general approach is to find a low-dimensional vector in latent subspace  $\Psi : Y \rightarrow Z$ , where  $Z \in \mathbb{R}^{N \times k}$  ( $k \ll l$ ), and then learn a mapping  $\Phi : X \rightarrow Z$ . A new instance will be firstly transformed into a  $k$ -dimensional vector using  $\Phi$ , which is then transformed into original label space with the inverse mapping of  $\Psi$ .

#### 3.2 Motivation

FSDR and LSDR are efficient paradigms for multi-label classification; they reduce computational cost and improve performance by removing redundant information from either the feature or label perspective. To inherit the advantages of both approaches, we propose to discover two low-dimensional subspaces for labels and features, respectively. In the context of multi-label learning methods, it is widely accepted that capturing the label correlations of data is essential for performance improvement. Label correlations are common in practical multi-label datasets. By exploring and exploiting correlations among labels, we are able to better tackle the multi-label learning problem.

Although the number of labels is very large, given label correlations, the combination in which labels can be combined will be limited in practice. We therefore aim to discover a subset of label components so as to reconstruct the whole labels. Mathematically speaking, we assume the existence of some base labels  $\{w_1, w_2, \dots, w_k\}$ , and the label  $y_i$  of a data point can be decomposed as  $y_i = \sum_{j=1}^k a_{ij} w_j$ , and  $\{a_{ij}\}_{j=1}^k$  is a new low-dimensional representation of the original label. Furthermore, we want to reconstruct the whole labels using the minimal number of label components. To achieve this, we assume that the combination coefficients of components are mutually independent.

Since sparsity has a close connection with non-Gaussianity, sparse coding can be formulated as a special case of independent component analysis (ICA). ICA would further reduce statistical dependencies and produce a sparse and independent representation that will be useful for the subsequent learning procedure; accordingly, we also employ ICA to find a

compact independent representation for the features. It would be unreasonable to discover feature subspace and label subspace separately; there are not only correlations among labels, but also correlations between features and labels. For example, the features extracted from an image and the labels of this image are representations of the image content from different points of view. Particularly when the number of labels is large enough, we can naturally treat labels as textual features of the examples. Thus, we not only want to find two independent subspaces for features and labels, but also hope that these two subspaces will have a strong correlation with each other. Many studies have shown that the correlations between feature and label have a substantial impact on the predictability of the latent spaces [8, 22, 42].

We perform decomposition on both features and labels,  $Y = AW_y, X = BW_x$ , where  $W_x$  and  $W_y$  are the bases of features and labels, and  $A$  and  $B$  are the bases of features and labels, while  $A$  and  $B$  are projections on the bases. We assume the bases are orthonormal to avoid redundant information, and further hold that  $YW_y^T = A, XW_x^T = B$ . We aim to maximize the non-Gaussianity of every dimension of  $A$  and  $B$ , which is a common approach to pursuing independence. Finally, we maximize the correlation of  $A$  and  $B$  so that there can be a simple mapping between  $A$  and  $B$ . Based on this idea, we can write our primary objective function as follows:

$$\begin{aligned} \max_{W_x, W_y} & g(X, W) + g(Y, W_y) + \gamma h(XW_x^T, YW_y^T) \\ \text{s.t.} & W_x W_x^T = I, W_y W_y^T = I \end{aligned} \tag{1}$$

where  $\gamma$  is a trade-off parameter,  $g(\cdot)$  is the measurement about non-Gaussianity,  $h(\cdot)$  is the correlation of two latent spaces.

### 3.3 Non-Gaussianity prior

Discovering independent components from random variables is a complicated endeavour. This kind of problem can be well formulated as an Independent Components Analysis (ICA) problem [16], in which the observed matrix is decomposed with a assumption of independence. Maximizing independence is equivalent to maximizing the non-Gaussianity in ICA models. The standard ICA can be defined as the following optimization problem:

$$\begin{aligned} \min_W & \sum_{i=1}^N \sum_{j=1}^k g(w_j x_i) \\ \text{s.t.} & w_i w_j^T = \delta_{ij} \end{aligned} \tag{2}$$

where  $g$  is a non-linear convex function to pursue non-Gaussian components [15].  $\delta_{ij}$  is Kronecker delta function, which equals to 1 if and only if  $i$  equals to  $j$ , otherwise is zero. Then we can rewrite formula (1) as

$$\begin{aligned} \min_{W_x, W_y} & \alpha \sum_{i=1}^N \sum_{j=1}^{k_x} g(W_x^j x_i) + \beta \sum_{i=1}^N \sum_{j=1}^{k_y} g(W_y^j y_i) \\ & - \gamma * h(XW_x^T, YW_y^T) \\ \text{s.t.} & W_x W_x^T = I, W_y W_y^T = I \end{aligned} \tag{3}$$

where  $W_x^j$  is  $j$ -th row of  $W_x$ , and  $\alpha_j$  and  $\beta_j$  are trade-off parameters. The presence of orthonormal constraints will lead to a difficult optimization. One way is orthogonalizing  $W_x$  and  $W_y$  during every update, this will result in higher computational costs. Accordingly,

to get rid of these orthogonal constraints, we aim to create an unconstrained function by replacing the orthonormal constraints with soft reconstruction cost [21] to achieve efficient optimization,

$$\begin{aligned} \min_{W_x, W_y} & \|X - XW_x^T W_x\|_F^2 + \alpha \sum_{i,j} \log(\cosh(XW_x^T))_{ij} \\ & + \|Y - YW_y^T W_y\|_F^2 + \beta \sum_{i,j} \log(\cosh(YW_y^T))_{ij} \\ & - \gamma h(XW_x^T, YW_y^T), \end{aligned} \tag{4}$$

where we have adopted  $\log(\cosh(\cdot))$  as the function  $g(\cdot)$ .

### 3.4 Correlation

As noted in our motivation section, we aim to maximize the correlation between the latent subspaces of features and labels. However, these features and labels might have different numbers of independent components, which requires us to calculate the correlation between variables of different dimensions. Canonical correlation seems to be a suitable choice of algorithm to tackle our problem. This is a correlation defined on two sources of input data with different dimensions.

$$\rho = \frac{u^T \Sigma_{xy} v}{\sqrt{u^T \Sigma_{xx} u} \sqrt{v^T \Sigma_{yy} v}} \tag{5}$$

where  $\Sigma_{xx} = \frac{1}{N} \hat{X}^T \hat{X}$ ,  $\Sigma_{yy} = \frac{1}{N} \hat{Y}^T \hat{Y}$ , and  $\Sigma_{xy} = \frac{1}{N} \hat{X}^T \hat{Y}$ .  $\hat{X}$  and  $\hat{Y}$  satisfy  $\sum_i \hat{x}_i = 0$  and  $\sum_j \hat{y}_j = 0$ ; this can be achieved by subtracting the empirical mean from the sample. There are several ways to calculate the correlation; however what we expect to calculate is the canonical correlation coefficient, rather than the values of  $u$  and  $v$ . We therefore follow the solution proposed in [23], which was also adopted by Deep CCA [2], because it is differentiable.

The total correlation of all components of  $\hat{X}$  and  $\hat{Y}$  can be computed as the sum of all singular values of  $T \triangleq \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2}$ . The sum of all singular values indicates the trace norm of  $T$ . But due to the non-smooth nature of the trace norm, we choose the  $\ell$ -2 norm of singular values which is the Frobenius norm of  $T$ ,

$$h(B, A) = \|T_{BA}\|_F = \text{tr}(T_{BA}^T T_{BA})^{1/2}. \tag{6}$$

In conclusion, we decompose both feature and label to discover independent components by maximizing the non-Gaussianity of the subspaces. Meanwhile, we also maximize the correlation of the projections on the bases. Overall, therefore, we can rewrite Equation (1) as follows:

$$\begin{aligned} \min_{W_x, W_y} J(W_x, W_y) &= \|X - XW_x^T W_x\|_F^2 + \|Y - YW_y^T W_y\|_F^2 \\ &+ \alpha \sum_{i,j} \log(\cosh(XW_x^T))_{ij} + \beta \sum_{i,j} \log(\cosh(YW_y^T))_{ij} \\ &- \gamma * \|T(XW_x^T, YW_y^T)\|_F \end{aligned} \tag{7}$$

### 3.5 Connections with other methods

Canonical Correlation Analysis is a proven technique that has been widely used in multi-label learning [30, 41]. Many variants of CCA have been proposed. Based on two lemmas in [21], we show that our proposed approach can be expressed as a sparse CCA.

**Lemma 1** *When the data is whiten, the reconstruction error of whiten data  $\|X - XW^T W\|_F^2$  is equivalent to the orthonormality cost  $\|I - W^T W\|_F^2$ .*

*Proof* For whiten data, we have  $X^T X = I$ , then

$$\begin{aligned} \|X - XW^T W\|_F^2 &= \text{tr}(X - XW^T W)^T (X - XW^T W) \\ &= \text{tr}(X^T X - 2X^T XW^T W + W^T W X^T X W^T W) \\ &= \text{tr}(I - 2W^T W + W^T W W^T W) = \|I - W^T W\|_F^2 \end{aligned}$$

□

**Lemma 2** *The column orthonormality cost  $\|I_c - W^T W\|_F^2$  is equivalent to the row orthonormality cost  $\|I_r - W W^T\|_F^2$  up to an additive constant.*

*Proof* Note that  $c$  and  $r$  are numbers of columns and rows.

$$\begin{aligned} \|I_c - W^T W\|_F^2 &= \text{tr}(I_c - 2W^T W + W^T W W^T W) \\ &= \text{tr}(I_r - 2W W^T + W W^T W W^T) + c - r \\ &= \|I - W W^T\|_F^2 + c - r. \end{aligned}$$

□

According to Lemmas 1 and 2, we could rewrite Eq. 7 as following

$$\begin{aligned} \min_{W_x, W_y} \|I - W_x W_x^T\|_F^2 + \|I - W_y W_y^T\|_F^2 - \gamma \|T(B, A)\|_F \\ + \alpha \sum_{i,j} g(B)_{ij} + \beta \sum_{i,j} g(A)_{ij}, \end{aligned} \tag{8}$$

where  $T(B, A)$  is a correlation term to measure the correlation between  $B$  and  $A$ . Hence for whiten data, our approach is similar to the Lagrangian function of CCA with sparse constraints.

Our approach consists of two parts: namely, reconstruction and correlation. Some existing methods have also applied this idea. We will here demonstrate the connection between CPLST and our proposed approach. CPLST merges the feature-aware information into its model. The objective function of CPLST can be written as follows:

$$\begin{aligned} \min_{U, V} \|XU - YV\|_F^2 + \|YV V^T - Y\|_F^2 \\ \text{s.t. } V^T V = I. \end{aligned} \tag{9}$$

Obviously the feature-aware term  $\min \|XU - YV\|_F^2$  of CPLST is a variant of CCA, the right term is the reconstruction error of label.

FaIE [22] is another multi-label algorithm that has also been developed to utilize reconstruction and correlation terms. One of its major advancements is that it directly learns a lower-dimensional representation without making any assumption regarding the encoding process. Its objective function is written as follows:

$$\begin{aligned} \min_C -\text{tr}[C^T X(X^T X)^{-1} X^T C] + \|Y - C C^T Y\|_F^2 \\ \text{s.t. } C^T C = I \end{aligned} \tag{10}$$

where the left term is the correlation term between feature  $X$  and lower-dimensional representation  $C$ , while the right term is also the reconstruction error of the label. Obviously,



CPLST and FaIE are alike. If we ignore the reconstruction term of the feature and set  $\alpha = 0, \beta = 0, \gamma = 1$ , we might get a similar result to that obtained by CPLST and FaIE.

From the formulations, it is clear that our proposed approach has no notable connections to CPLST, sparse CCA and FaIE. Although they have totally different motivations, they all have reconstruction and correlation terms, which play an important role in embedding-based approaches for multi-label classification. In contrast with CPLST and FaIE, moreover, our proposed approach intends to encode features and labels into two different subspaces. In optimization, the original CPLST and FaIE also involve orthonormal constraints, but they solve this problem by turning it into an eigendecomposition problem. By contrast, our approach optimizes an unconstrained function and allows for an unconstrained optimizer (e.g., L-BFGS, S-GD). Given  $n$  examples, both CPLST and FaIE have to compute a  $n \times n$  matrix which is infeasible for larger  $n$ . So CPLST and FaIE used to cluster data at first. But clustering will slow down prediction speed. We use mini-batch update, which allows us to tackle large datasets without clustering (Table 1).

### 3.6 Mapping between subspaces

After learning subspaces for features and labels, we need to learn a mapping between these two subspaces, as this will allow us to transform from feature to label in order to accomplish multi-label learning. Since the two subspaces are low dimensional, we can easily adopt some efficient multi-dimensional regressors to assist us. We train the regressor based on following function:

$$\min_f \|f(XW_x^T)W_y - Y\|_F^2 \tag{11}$$

where  $f$  is an arbitrary regressor e.g. ridge regressor and neural network. Multi-label examples are supposed to share the independent components. For test sample, we map their features into the subspace of features. With the learned regressor, we can get the coefficients of test sample in label subspace, i.e.  $Y_{pred} = f(X_t W_x^T)W_y$  where  $X_t$  is test data and  $Y_{pred}$  is the prediction.

### 3.7 Neural network extension

To extend our method to a non-linear one, we leverage the power of the neural network. We can describe our method as a latent representation learning method. The first characteristic of the representation is that it is independent. The independence requirement is like an advanced concept of decorrelation: decorrelation aims to minimize the co-variance of representation, while the independence requirement is an attempt to minimize the high-order statistic of representation. Some recent works have suggested that independent representations are more effective. The next characteristic is that it is reconstructable: this means we can reconstruct the original label with the representation. The final characteristic is that it is

**Table 1** A summary of different multi-label learning methods

Algorithm	Correlation	Reconstruction	Regularization
CPLST	Yes	Only $Y$	Orthonormality
FaIE	Yes	Only $Y$	Orthonormality
SCCA	Yes	None	Orthonormality
IFLC	Yes	Both $X, Y$	Non-Gaussianity

correlative, as many works have shown that feature-aware label subspace is much more useful. Operating under these three guidelines, we present the neural network architecture of DeepIFLC in Fig 1. The architecture consists of two independent component autoencoders and one deep canonical correlation analysis (DCCA) module. The objective function can be written as follows:

$$J(\phi_x, \phi_y, \psi_x, \psi_y) = R(\phi_x) + R(\phi_y) + \alpha T(\phi_x, \phi_y) + L(\phi_x, \psi_x) + L(\phi_y, \psi_y) \tag{12}$$

where  $R$  is the regularizer of autoencoder,  $T$  is the canonical correlation term and  $L$  is the reconstruction loss for autoencoder.

It is well known that a non-linear ICA problem is an ill-posed problem that can only be solved with additional assumptions. Thus, rather than searching for a truly non-linear independent component estimator, we instead propose a method called Deep Independent Component autoEncoder (DICE), which satisfies only some essential properties of the ICA solution. This means that the representation should be uncorrelated, while the independent representation should minimize some high-order statistic, such as kurtosis. Therefore, our regularization is quite straightforward. Note that the output of encoder is  $Z$ , which has zero means: thus, the regularizer can be written as follows:

$$R(\phi) = \beta \|C_{zz}\|_1 - \gamma |n(1_n^T Z^4)/(1_n^T Z^2)^2 - 3|1_k \tag{13}$$

where  $C_{zz}$  is the sample covariance matrix of representation  $Z$ ,  $\|C_{zz}\|_1 = \sum_{i,j} |C_{zz}^{ij}|$ . And  $1_n$  is a column vector with all one elements,  $\gamma$  is trade-off parameter. Since this regularier is differentiable, so it can be applied to the autoencoder directly.

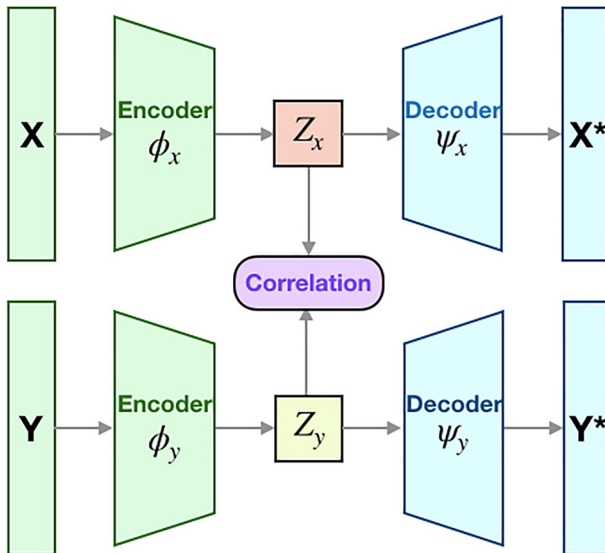


Fig. 1 The architecture of DeepIFLC

### 3.8 Stochastic independent loss

To apply on neural network, we need to estimate the true covariance and kurtosis of whole data with mini-batch. Therefore, we propose a stochastic version of Eq. 13. We denote mini-batch  $Z \in \mathbb{R}^{m \times k}$ , where  $m$  is the batch size,  $k$  is the number of independent components.

The mini-batch covariance matrix and kurtosis vector for  $t$ -step are given as:

$$C_{mini}^t = \frac{1}{m-1} Z^T Z \tag{14}$$

$$K_{mini}^t = n(1_n^T Z^4) / (1_n^T Z^2)^2 - 3 \tag{15}$$

Then we approximate the true covariance and kurtosis by accumulating the history

$$C_{accu}^t = \delta C_{accu}^{t-1} + C_{mini}^t \tag{16}$$

$$K_{accu}^t = \delta K_{accu}^{t-1} + K_{mini}^t \tag{17}$$

where  $\delta \in [0, 1)$ . A normalising factor is also computed as  $c^t = \delta c^{t-1} + 1$ . Then the approximate covariance and kurtosis are given as:

$$C_{appr}^t = \frac{C_{accu}^t}{c_t} \tag{18}$$

$$K_{appr}^t = \frac{K_{accu}^t}{c_t} \tag{19}$$

Therefore, Eq. 13 can be rewritten as follow:

$$R(\phi) = \beta \|C_{appr}\|_1 - \gamma \|K_{appr}\|_1 \tag{20}$$

For reconstruction loss, we use L2-loss for feature and BP-MLL loss for label.

$$L(\phi_x, \psi_x) = \|X - \psi_x(\phi_x(X))\|_F^2 \tag{21}$$

$$L(\phi_y, \psi_y) = \sum_i E_i$$

$$E_i = \frac{1}{|y_i^0| |y_i^1|} \sum_{(p,q) \in y_i^1 \times y_i^0} \exp(\psi_y(\phi_y(y_i))^q - \psi_y(\phi_y(y_i))^p) \tag{22}$$

As for correlation term, with the existence of covariance minimization, the L2-loss could be a good approximation of canonical correlation.

$$T(\phi_x, \phi_y) = \|Z_x - Z_y\|_F^2 \tag{23}$$

---

**Algorithm 1** Training stage of the proposed approach.

---

**Input:** training sample matrix  $X$ , training label matrix  $Y$ , number of independent components  $k$ , trade-off parameters  $\alpha$ ,  $\beta$  and  $\gamma$ ;

**Output:**  $W_x$ ,  $W_y$ , and  $f(\cdot)$ ;

- 1:  $W_x := 0, W_y := 0$
  - 2: **repeat**
  - 3: compute gradient matrix  $\nabla W_x$  and  $\nabla W_y$ ;
  - 4: update  $W_x, W_y$  via gradient decent;
  - 5: **until**  $J$  is convergent
  - 6:  $f(\cdot) \leftarrow$  train regression with Eq. 11;
-

**Algorithm 2** Test stage of the proposed approach.

**Input:** testing sample matrix  $X_t$ , independent component matrices  $W_x$  and  $W_y$ , and regressor  $f(\cdot)$ ;

**Output:** Predict label matrix  $Y_{pred}$ ;

- 1:  $B_t \leftarrow X_t W_x^T$ ;
- 2:  $Y_{pred} \leftarrow f(B_t) W_y$ ;

### 4 Optimization

Due to the existence of correlation term in formula (7), our objective function is not a convex function for both  $W_x$  and  $W_y$ . But formula (7) is a convex function for variables  $W_x$  or  $W_y$  separately. We can hardly find a global optimal solution with gradient descent method. So we decide to adopt a greedy strategy to tackle this problem approximately by alternatively solving the subproblems of formula (7).

#### 4.1 Updating $W_x$ and $W_y$

The subproblem of solving  $W_x$  can be written as following

$$W_x = \arg \min_{W_x} \|X - X W_x^T W_x\|_F^2 + \alpha \sum_{i,j} \log(\cosh(X W_x^T))_{ij} - \gamma \|T(X W_x^T, Y W_y^T)\|_F \tag{24}$$




where  $X$  is the feature matrix. It is hard to write a closed form of formula (24), but the subproblem is differentiable. So we can use some unconstrained optimizers (e.g., L-BFGS,SGD) to minimize this loss function. The derivative of  $W_x$  consists of two parts: reconstruction term and correlation term. The derivative of reconstruction term can be computed as

$$\nabla_{W_x} J(W_x, W_y) = 2W_x W_x^T W_x X^T X + 2W_x X^T X W_x W_x^T - 4W_x X^T X + \alpha * \tanh(W_x X^T) X \tag{25}$$

To compute the gradient of correlation term, we could use the chain rule. Given  $B = X W_x^T$  and  $A = Y W_y^T$ , the centered matrices are  $\hat{B} = B - \frac{1}{N} \mathbf{1} B$  and  $\hat{A} = A - \frac{1}{N} \mathbf{1} A$ , where  $\mathbf{1} \in \mathbb{R}^{N \times N}$  is an all-1s matrix. Assume the singular value decomposition of  $T(B, A)$  is  $T(B, A) = U D V^T$ . We have

$$\begin{aligned} (\nabla_{W_x} h(B, A))_{ij} &= \sum_{a,b} \frac{\partial h(B, A)}{\partial (B)_{ab}} \frac{\partial (B)_{ab}}{\partial (W_x)_{ij}} \\ &= \sum_{a,b} \frac{1}{N-1} (2\nabla_{xx} \hat{B} + \nabla_{xy} \hat{A})_{ab} X_{aj} \delta_{bi} \\ &= \frac{1}{N-1} ((2\nabla_{xx} \hat{B} + \nabla_{xy} \hat{A})^T X)_{ij} \end{aligned} \tag{26}$$

**Table 2** Examples of test images from *iapr tc12* dataset

			
Ground Truth	Court dress grandstand People player tennis	Bell front spectator Surfer wall	Mountain road sky Grey dirt forest
FaIE	Sky player court <b>Man building tree</b>	Wall spectator <b>house</b> Front <b>tree</b>	<b>tree slope</b> mountain Sky <b>short man</b>
SLEEC	Court player <b>stadium</b> Tennis grandstand <b>man</b>	Wall front <b>man</b> <b>Sky people</b>	<b>Slope rock</b> mountain <b>Tree man bush</b>
IFLC	Player court tennis Grandstand <b>man</b> people	Spectator wall front <b>People man</b>	<b>Tree slope</b> road mountain sky forest

For each image, we show the ground truth annotation and the most relevant labels predicted by FaIE, SLEEC and ours. The labels in black are those that match with ground truth. The labels in blue are related to image but not include in ground truth. The labels in red are irrelevant annotation

where

$$\begin{aligned} \nabla_{xy} &= \Sigma_{xx}^{-1/2} U V^T \Sigma_{yy}^{-1/2} \\ \nabla_{xx} &= -\frac{1}{2} \Sigma_{xx}^{-1/2} U D V^T \Sigma_{xx}^{-1/2} \end{aligned}$$

Then the total gradient of  $S_x$  is as following

$$\begin{aligned} \nabla_{W_x} J(W_x, W_y) &= 2W_x W_x^T W_x X^T X + 2W_x X^T X W_x W_x^T \\ &\quad - 4W_x X^T X + \alpha * \tanh(W_x X^T) X \\ &\quad - \frac{\gamma}{N-1} ((2\nabla_{xx} \hat{B} + \nabla_{xy} \hat{A})^T X), \end{aligned} \tag{27}$$

and the gradient w.r.t.  $W_y$  has a similar expression (Table 2).

## 5 Experiments

### 5.1 Configuration

In order to validate the proposed algorithm, we perform experiments on six benchmark datasets from Mulan.<sup>1</sup> There are two relatively small datasets: *medical* [24] and *enron* [20], and two other larger datasets: *Corel16k* [3] and *iapr tc12* [11]. For dataset *corel16k*, it contains over 16,000 different images. It is organized into 10 different samples, we choose three

<sup>1</sup><http://mulan.sourceforge.net/datasets-mlc.html>

**Table 3** Statistics of the datasets in experiments

	Domain	Train	Test	Features	Labels
<i>medical</i>	text	333	645	1449	45
<i>enron</i>	text	1123	579	1001	53
<i>corel16k1</i>	image	5188	1744	500	153
<i>corel16k2</i>	image	5241	1783	500	164
<i>corel16k7</i>	image	5266	1747	500	174
<i>iapr tc12</i>	image	17665	1962	1000	291

samples of them for computational cost. All these datasets have already been pre-separated into training set and testing set. And for *iapr tc12*<sup>2</sup> dataset, it is a collection of 19,627 nature images taken from locations around the world. This includes pictures of different sports and actions, photographs of people, animals, cities, landscapes and many other aspects of contemporary life. We adopt SIFT-based representation as [12]. Some statistics of data are given in Table 3.

In experiment, we compared our method with some competitive methods to validate its predictive performance. Their brief introductions are given as below.

- **CPLST**: The label space is encoded by a feature-aware principal label space transformation. It reduces label while considering feature information.
- **FaIE**: This approach aims to find a latent space that maximizes its recoverability and predictability. Linear-FaIE is adopted in experiments.
- **LEML**: This is one of the state-of-the-art approaches to tackle multi-label problem in a generic low rank empirical risk minimization framework.
- **SLEEC**: This is one of the state-of-the-art approaches for learning sparse local embeddings in multi-label classification. It finds a latent space that preserves the pairwise distances between the closest label vectors.

The implementations of all comparison methods were accomplished by using codes provided by authors. For the hyper parameters, we used the recommendation by authors, if there is. Otherwise, we tune their hyper parameters to achieve on different datasets. For SLEEC, it needs to cluster before embedding. We set the number of clusters as  $N_{clusters} = \lceil N/6000 \rceil$  as recommendation, where  $N$  is the number of training data. For our method, the hyper parameters  $\alpha$ ,  $\beta$  and  $\gamma$  are determined by 5-fold cross validation on training set. They are chosen from {0.001, 0.01, 0.1, 1, 10}

One key parameter for all methods is the label compression rate  $k/l$ , where  $l$  and  $k$  are the dimension of latent label space and the original label space, respectively. We compared all methods under four different rates, 20%, 40%, 60%, and 80%. We also investigate the sensitivity of parameters on the *Corel16k*, *medical* and *iapr tc12* datasets.

Following previous works, we use ridge regression as our base regressor for a fair comparison. Except for SLEEC, it used kNN to predict. We used two widely-used ranking-based evaluation metrics to validate all methods, *i.e.* precision of top-k prediction (P@k) that counts the fraction of correct prediction in the top-k scoring label, and normalized Discounted Cumulative Gain (nDCG@k). These metrics have been commonly used in many

<sup>2</sup><http://lear.inrialpes.fr/people/guillaumin/data.php>

multi-label learning experiments. Denote  $y \in \{0, 1\}^l$  as the ground-truth label,  $\hat{y} \in \{0, 1\}^l$  as the predicted label.  $P@k$  and  $nDCG@k$  are expressed as

$$P@k := \sum_{i \in \text{rank}_k(\hat{y})} y_i,$$

and

$$nDCG@k := \frac{1}{k} \sum_{i \in \text{rank}_k(\hat{y})} \frac{y_i}{\log(i + 1)},$$

where  $\text{rank}_k(\hat{y})$  returns the  $k$  largest indices of  $\hat{y}$  ranked in descending order.

### 5.2 Experiment results

To validate the proposed method, we compare our method with several popular methods, and then some variants of our proposed method are included to illustrate the reasonability of our configuration. Tables 4 and 5 show the general result. We evaluate experiment results in term of Precision@1,3,5 and nDCG@1,3,5. Generally, our approach obtains the best result. From the experiment results, we observe that: (1) The performance of our proposed approach improves as the label compression rate increases on almost every dataset; (2) Compared with other methods, our approach has a significant better performance in top-1 and top-3 evaluation metrics, and our approach achieves the best top-1 precision on all datasets; (3) SLEEC did not work very well in enron. We think the reason might be this dataset has small label size, so that the distance of two label vectors cannot precisely reflect the label structure; (4) Our approach has a better ranking in nDCG compared to that in terms of Precision. It is because that nDCG is a cumulative quantity, and the top-1 gain has the biggest weight. As shown in these two tables, our approach has advantages in terms of top-1 and top-3 metrics. Therefore, our approach can have a better performance in nDCG; (5) As an extension of CPLST and FaIE, our approach generally outperforms CPLST and FaIE on these datasets. Generally, we achieve 1% to 2% average improvement over CPLST and FaIE.

### 5.3 Feature independent components

The number of feature-independent components (FIC) is one of the key hyperparameters in our approach. We analyze the influence of the number of FIC across the three datasets; in so doing, we fix the other hyperparameters, and adjust only the number of FIC. We also choose FIC rates (FIC over the feature dimension) between 0.01 and 1.4. Results are presented in Figure 2. From the Figure, we can observe the following: (1) The performance on all datasets remain stable as the FIC rate increases, the converges to its best performance. We can therefore choose a high FIC rate to ensure high performance. Even for lower FIC rates such as 0.1, however, a good enough result can be expected. This phenomenon suggests that the example feature is often redundant, which is consistent with the motivation of FSDR methods. (2) We observe a decrease in performance when the FIC rate is below some threshold (i.e. the red line). Note that the decrease is usually on the left side of the red line, indication that the dimension of feature subspace is smaller than that of the label subspace. Under these circumstances, a multivariate regression that maps a low-dimensional space to a high dimensional space can be quite inaccurate. The performance of conventional embedding methods that discover single subspace should around the red line. As the FIC rate increases, moreover, the performance of our method becomes slightly better that red line, especially for the top-1 metric. This explains the superiority of our method in terms of the top-1 metric. (3) The behaviors of the FIC rate and label compression rate are quite different.

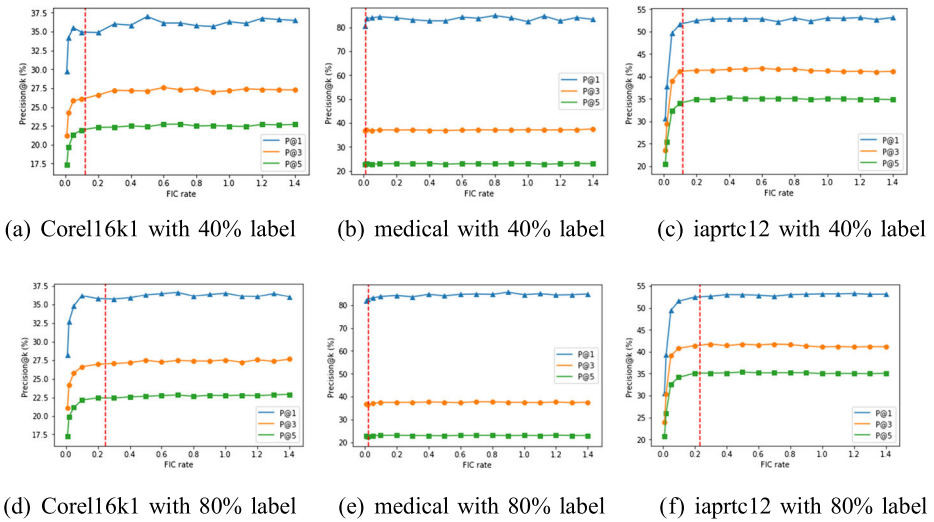
**Table 4** Performance comparison of multi-label learning methods with precision@k (%)

Dataset	Medical			Enron			Corell6k1			Corell6k2			Corell6k7			iapr tc12			
	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5	
CPLST	20%	76.12	33.17	21.24	77.72	59.70	45.94	35.03	26.39	22.06	37.63	28.80	23.55	36.17	27.18	22.18	51.78	40.71	34.71
	40%	82.17	37.36	23.19	77.89	59.75	46.66	34.34	26.28	22.22	37.35	28.60	23.67	35.54	27.34	22.25	52.29	40.89	34.90
	60%	84.65	38.08	23.44	77.72	59.81	46.63	34.46	26.37	22.13	37.18	28.71	23.70	35.60	27.34	22.23	52.24	41.05	34.92
FaIE	20%	84.65	37.98	23.50	77.72	59.75	46.66	34.34	26.35	22.07	37.12	28.77	23.74	35.77	27.45	22.26	52.40	40.93	34.96
	40%	73.33	31.36	19.75	77.72	59.81	46.18	34.86	26.07	21.54	37.35	28.43	23.44	36.12	26.57	21.72	51.89	40.01	34.04
	60%	81.86	36.74	22.82	78.23	60.67	47.42	34.40	26.50	22.14	37.12	28.52	23.76	35.54	26.63	21.55	52.34	41.03	34.91
LEMML	20%	84.49	37.89	23.50	78.58	60.62	47.39	34.51	26.54	22.32	37.07	28.77	23.80	35.60	26.69	21.37	52.75	41.23	35.17
	40%	84.65	38.03	23.56	78.58	60.73	47.46	35.03	26.49	22.32	37.12	28.79	23.91	35.66	26.59	21.51	52.80	41.23	35.24
	60%	71.16	31.58	20.12	76.17	59.93	45.94	34.86	26.44	22.26	37.41	28.77	23.77	36.12	27.11	22.13	52.14	40.25	34.15
SLEEC	20%	81.86	36.33	22.70	76.34	60.16	46.36	34.98	26.61	22.28	37.58	28.72	23.81	36.00	27.19	22.23	52.50	40.79	34.73
	40%	83.88	37.52	23.41	76.34	60.45	46.36	35.09	26.64	22.29	37.46	28.70	23.81	36.00	27.21	22.26	52.45	41.18	34.79
	60%	83.88	37.52	23.50	76.17	60.22	46.39	35.09	26.64	22.29	37.46	28.68	23.81	36.00	27.21	22.27	52.55	40.96	35.02
IFLC	20%	76.74	33.48	20.71	72.19	52.67	39.75	33.31	26.49	21.27	37.29	28.56	23.67	35.60	26.52	21.55	52.85	41.52	34.56
	40%	81.71	34.88	21.30	70.46	53.59	39.86	33.77	27.17	22.21	38.02	28.71	23.91	35.08	26.69	21.78	52.29	41.54	34.76
	60%	84.81	35.29	21.58	71.15	54.74	40.93	34.86	27.71	22.50	37.91	28.56	23.92	35.48	26.59	21.84	51.99	41.47	34.64
IFLC	20%	85.12	35.55	21.58	69.77	53.77	40.17	34.97	27.21	22.20	37.74	28.92	23.94	35.71	27.17	21.87	52.29	41.23	34.67
	40%	79.37	34.26	21.73	77.72	60.67	45.56	35.61	26.51	21.73	38.69	28.51	23.65	36.40	27.22	22.21	53.01	41.11	34.66
	60%	83.56	36.84	22.85	78.41	60.79	47.28	36.07	27.22	22.19	38.81	28.84	23.92	36.66	27.26	22.35	53.11	41.86	35.32
IFLC	20%	84.96	37.88	23.44	78.92	60.91	47.28	36.12	27.29	22.47	38.75	28.56	24.00	36.52	27.40	22.34	53.15	42.37	35.35
	40%	85.89	37.57	23.22	79.10	60.91	47.01	35.60	27.45	22.67	38.58	28.94	23.92	36.46	27.34	22.30	53.67	42.01	35.29
	60%	85.89	37.57	23.22	79.10	60.91	47.01	35.60	27.45	22.67	38.58	28.94	23.92	36.46	27.34	22.30	53.67	42.01	35.29



**Table 5** Performance comparison of multi-label learning methods with ndeg@k (%)

Dataset	Medical			Enron			Corell16k1			Corell16k2			Corell16k7			ipr tc12		
	n@1	n@3	n@5	n@1	n@3	n@5	n@1	n@3	n@5	n@1	n@3	n@5	n@1	n@3	n@5	n@1	n@3	n@5
CPLST	20%	76.12	77.89	80.07	77.72	70.72	71.57	35.03	30.34	34.03	37.63	32.46	35.73	36.17	31.51	51.78	43.99	41.38
	40%	82.17	87.19	88.51	77.89	70.46	72.48	34.34	30.06	33.96	37.35	32.28	35.78	35.54	31.50	52.59	44.28	41.66
	60%	84.65	89.36	90.30	77.72	70.48	72.48	34.46	30.15	33.91	37.18	32.33	35.80	35.60	31.62	52.24	44.41	41.71
FaLE	20%	84.65	89.24	90.40	77.72	70.46	72.53	34.34	30.15	33.84	37.12	32.32	35.80	35.77	31.69	52.40	44.35	41.71
	40%	81.86	85.99	87.27	78.23	71.35	73.50	34.40	30.30	34.05	37.12	32.17	35.80	35.54	30.83	52.34	44.45	41.74
	60%	84.49	89.19	90.33	78.58	71.39	73.53	34.51	30.37	34.22	37.07	32.36	35.90	35.60	30.90	52.75	44.68	42.05
LEMML	20%	84.65	89.32	90.48	78.58	71.43	73.57	35.03	30.37	34.26	37.12	32.37	36.01	35.66	30.85	52.80	44.69	42.13
	40%	81.16	85.48	87.02	76.17	69.98	71.26	34.86	30.71	34.29	37.41	32.38	35.81	36.12	31.27	52.14	43.77	40.98
	60%	83.88	88.28	89.76	76.34	70.26	71.82	34.98	30.61	34.31	37.58	32.39	35.89	36.00	31.31	52.50	44.18	41.55
SLEEC	20%	83.88	88.30	89.89	76.17	70.24	71.77	35.09	30.67	34.35	37.46	32.35	35.87	36.00	31.32	52.45	44.55	41.66
	40%	81.71	84.55	85.06	70.46	64.13	63.98	33.31	29.56	32.52	37.29	32.17	35.63	35.60	30.83	52.55	44.43	41.83
	60%	84.81	86.19	86.84	71.15	64.94	65.42	34.86	31.34	34.59	37.91	32.19	35.87	35.08	30.70	52.29	44.69	41.54
IFLC	20%	85.12	86.65	87.06	69.77	64.15	64.48	34.97	30.99	34.29	37.74	32.60	36.27	35.71	31.31	51.99	44.56	41.40
	40%	79.37	81.87	83.85	77.72	71.06	71.44	35.61	30.55	33.83	38.69	32.48	35.99	36.40	31.53	52.29	44.42	41.45
	60%	83.56	87.37	88.51	78.41	71.46	73.52	36.07	31.21	34.98	38.81	32.74	36.31	36.66	31.65	53.01	44.61	41.66
80%	84.96	89.52	90.17	78.92	71.51	73.62	36.12	31.39	34.97	38.75	32.42	36.33	36.52	31.71	53.15	45.19	42.37	42.50
	85.89	89.33	90.40	79.10	71.91	73.58	35.60	31.40	34.98	38.58	32.69	36.21	36.46	31.70	53.67	45.55	42.54	42.54



**Fig. 2** Precision under different FIC rates. Where blue line is Precision@1, orange line is Precision@3, and green line is Precision@5. The red line means the moment that the dimensions of two subspaces are the same

The FIC rate has a small threshold, beyond which performance will barely rely on FIC rate. Usually, however the higher the label compression rate is, the better performance we will obtain. As shown in Tables 4 & 5, there is a significant improvement on the medical dataset under a 20% label compression rate. Note that a 20% label compression rate on the medical dataset implies that there are nine label-independent components; in other words, there are nine feature-independent components for the traditional embedding method, while the FIC rate is around 0.006, which is close to the threshold on the medical dataset. Hence, comparison methods encounter a performance reduction given that the FIC rate is so low. This is likely why our proposed approach achieves such a significant improvement over comparison methods on the medical dataset. It is difficult to determine how many FIC do we exactly need in practice; however, the experimental results suggest using an FIC rate larger than 0.1, and the FIC should also be larger than the number of label-independent components.

**5.4 Comparison with variants algorithm**

To validate the effect of each part in our model, we consider the following variants: (1) L-IFLC considering only ICA terms of label and correlation term. We achieve that by fixing  $W_x$  to an identity matrix; (2) F-IFLC considers only ICA terms of feature and correlation term; (3) C-IFLC excludes the correlation term. In this experiment, we perform all methods with 40% label compression rate. For F-IFLC we use 40% FIC rate. Results are given in Table 6.

From the experimental results, we can draw the following observations. (1) C-IFLC becomes the worst method. There is a huge gap between C-IFLC and other methods. Therefore correlation term is very important for our method. (2) In Corel16k1, F-IFLC outperforms L-IFLC. But in iaprtc12, L-IFLC outperforms F-IFLC. As CPLST suggested, there are two types of label correlation: feature-unaware correlation and feature-aware correlation. L-IFLC might overemphasize the feature-unaware correlation, and F-IFLC goes to

**Table 6** Performance comparison of variant methods with precision@k

Dataset	P@k	L-IFLC	F-IFLC	C-IFLC	IFLC
Corel16k1	P@1	33.94	35.14	20.12	36.07
	P@3	25.72	26.98	19.26	27.22
	P@5	21.39	22.08	13.21	22.19
iaprtc12	P@1	52.03	51.12	47.50	53.11
	P@3	40.57	40.48	35.03	41.86
	P@5	34.59	34.46	27.69	35.32

the other way. Our method makes a reasonable balance of those two by introducing both feature and label subspaces, and achieves even better results.

### 5.5 Results on iapr tc12 dataset

The iapr tc12 dataset is much larger with various label categories. Our approach achieves the best result over all comparison methods under almost every label compression rates in terms of different evaluation metrics. Table 2 presents example annotations on the iapr tc12 dataset produced by FaIE, SLEEC and our proposed approach. The top-k predicted labels from each method were taken as the annotation labels, where k is the number of true labels. As shown in the table, the mismatched labels of our approach are still quite related to the image content.

### 5.6 Comparisons with DNN methods

In this section, we validate our deep learning version method (DICE) with some state-of-the-art works:

- BP-MLL: One of the baseline neural network methods for multi-label classification.
- C2AE: Being able to learn a feature-aware label subspace, while it only contains reconstruction of label.
- CNN-RNN: An unified framework that uses RNN to explore the label co-occurrence, then combining the recurrent representation and CNN feature to improve the performance of classification.

We conduct our experiments on these datasets: *iapr tc12*, *tmc2007*, *espgame* and *NUS-WIDE*. These four datasets are all image datasets. For the first three datasets, 1000-dimensional SIFT features are extracted. For *NUS-WIDE*, we extract 4096-dimensional features with pre-trained Alexnet. For fair comparison purpose, we also use the same pre-trained Alexnet structure for CNN-based method CNN-RNN, in its CNN part.

For neural network architecture, our method, C2AE and BP-MLL use the same architecture. We use two hidden layers to encode(decode) features, and one hidden layer to encode(decode) labels. For each hidden layer, a total of 512 neurons are deployed. For output layer, we use Sigmoid function as our activation function. And use leaky ReLU function for other layers. The batch size is fixed as 500 for C2AE and BP-MLL, DICE's batch size is fixed as 100. To select parameters for DICE, we randomly leave out 1/6 of our training data

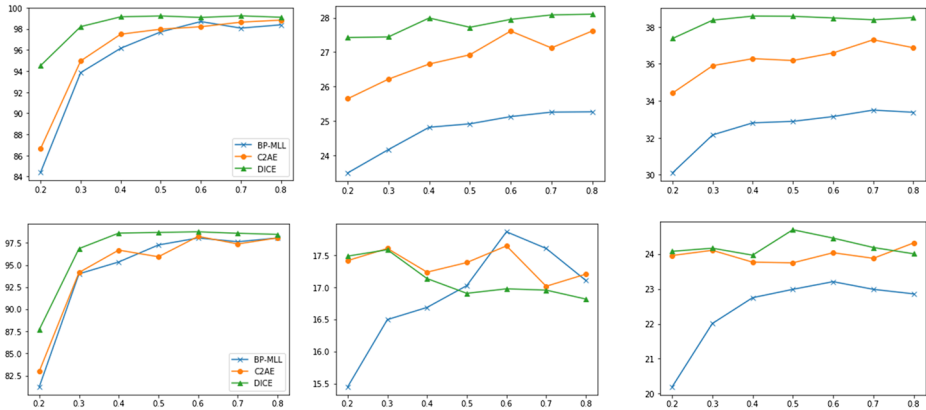


Fig. 3 Micro-F1 and macro-F1 of different methods

for validation. We adopt the same strategy for other methods. As for evaluation metrics, we consider micro-F1, macro-F1, top-k precision and top-k nDCG. But for CNN-RNN we only validate on micro-F1 and macro-F1.

Figure 3 and Table 7 illustrate and compare the performance of DICE, C2AE and BP-MLL. From Table 7, we can see that our DICE achieves superior performance against others.  $DICE > C2AE > BP-MLL$ . Both DICE and C2AE contain BP-MLL loss, so the improvement over BP-MLL are quite obvious. And compare to C2AE, DICE introduces the stochastic independent loss and the reconstruction of feature, which allow it attains better result. And with the help of stochastic independent loss, we are able to use much smaller batch size which can save many computing resources. From Fig. 3, we notice that BP-MLL will suffer more serious performance degradation when the label compression rate is small. We think that is because of the existence of feature-aware information.

### 5.7 Results on NUS-WIDE

NUS-WIDE dataset is a web image dataset, which consists of 269,648 images and 5018 tags from Flickr. After some simplifications, only 81 concepts are remained. They are more accurate and less noisy. Table 8 lists and compares the classification results of all four methods. CNN-RNN use RNN to exploit label co-occurrence information. According to the experiment results, label co-occurrence information seems like a weak information for classification. Since CNN-RNN can not even outperform BP-MLL. C2AE introduces feature-aware information and achieves better results. But our method DICE still attain promising results among all methods. This supports the benefits of independent representation and feature reconstruction.

For computation time, due to the learning of RNN, CNN-RNN takes relatively long time to train. Although DICE takes more time than BP-MLL and C2AE for one epoch, DICE actually converges faster. Overall, DICE might faster than C2AE, meanwhile achieves better results.

**Table 7** performance comparison of DNN methods

Dataset	tmc2007			esgame			iapr tc12			tmc2007			esgame			iapr tc12			
	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5	n@1	n@3	n@5	n@1	n@3	n@5	n@1	n@3	n@5	
BP-MLL	20%	96.26	67.37	43.06	34.16	27.76	22.58	47.68	38.53	32.33	96.26	97.46	97.86	34.16	29.35	29.57	47.67	41.47	38.68
	30%	98.06	68.64	43.61	34.51	27.69	22.84	49.08	39.62	33.35	98.06	99.04	99.11	34.51	28.99	28.48	49.09	42.65	39.93
	40%	98.05	68.70	43.65	34.32	27.59	22.97	48.72	39.08	33.36	98.05	99.07	99.14	34.32	30.07	29.06	48.72	42.06	39.69
	50%	97.75	68.80	43.71	34.07	28.15	23.08	49.96	38.97	33.10	97.75	99.05	99.11	34.07	30.56	29.22	49.96	42.47	39.92
	60%	98.47	68.87	43.74	34.75	27.91	23.29	49.24	40.03	33.82	98.47	99.35	99.39	34.75	30.44	29.34	49.24	42.75	40.08
	70%	97.56	68.78	43.73	34.78	27.45	23.33	49.67	39.59	33.68	97.56	98.96	99.05	34.78	30.51	29.46	49.67	42.72	40.14
C2AE	20%	96.38	67.05	42.96	35.66	28.93	26.49	51.53	41.06	35.20	96.38	97.50	97.28	35.66	31.41	30.65	51.53	44.30	41.94
	30%	98.47	68.10	43.41	34.94	28.08	24.28	52.04	42.90	36.36	98.47	98.65	98.91	34.94	30.62	30.29	52.04	45.91	43.23
	40%	99.01	68.56	43.64	35.03	29.25	24.67	52.19	42.42	36.28	99.01	99.23	99.38	35.03	31.72	30.97	52.19	45.51	43.15
	50%	99.05	68.48	43.72	36.71	29.68	24.91	52.80	42.32	36.06	99.05	99.17	99.43	36.71	32.36	31.46	52.80	45.56	42.90
	60%	99.18	68.61	43.70	35.61	29.92	25.41	53.41	43.36	36.21	99.18	99.31	99.48	35.61	32.27	31.61	53.41	46.55	43.34
	70%	98.78	68.46	43.69	35.18	29.51	25.16	53.57	43.95	36.55	98.78	99.04	99.29	35.18	31.72	31.15	53.57	46.99	43.68
DICE	20%	98.59	68.40	43.68	36.38	30.27	25.35	53.41	43.46	36.54	98.59	98.85	99.14	36.38	32.85	31.81	53.41	46.65	43.62
	30%	96.72	67.67	43.06	36.73	30.08	25.36	54.31	44.06	37.03	96.72	97.91	98.13	36.73	32.67	31.78	54.31	47.32	44.16
	40%	98.98	68.65	43.63	37.61	30.51	25.50	55.40	44.90	38.35	98.98	99.39	99.48	37.61	33.23	32.21	55.40	48.22	45.56
	50%	99.48	68.89	43.74	37.71	30.56	25.99	54.87	45.45	38.72	99.48	99.71	99.74	37.71	33.29	32.58	54.87	48.55	45.77
	60%	99.83	68.90	43.76	37.64	30.31	25.59	55.76	45.50	38.67	99.83	99.84	99.86	37.64	33.04	32.15	55.76	48.74	45.88
	70%	99.28	68.87	43.77	37.53	30.69	25.95	55.68	45.56	38.64	99.28	99.63	99.69	37.54	33.34	32.51	55.68	48.81	45.91
80%	99.87	68.88	43.77	37.69	30.82	26.04	55.47	45.59	38.81	99.87	99.83	99.89	37.69	33.42	32.57	55.47	48.80	45.99	
80%	99.87	68.89	43.75	38.01	31.12	26.18	55.77	45.91	38.96	99.87	99.85	99.87	38.01	33.68	32.86	55.77	49.10	46.26	

**Table 8** performance comparison on NUS-WIDE

Method	C-P	C-R	C-F1	O-P	O-R	O-F1
CNN-RNN	40.5	30.4	34.7	49.9	61.7	55.2
BP-MLL	44.5	39.8	38.3	57.3	68.9	62.5
C2AE	55.8	45.3	48.6	66.2	69.1	67.6
DICE	56.9	46.4	51.1	68.1	70.1	69.1

## 6 Conclusion

In this paper we proposed a method that learns separated subspaces for features and labels by maximizing the independence between components in each subspace and maximizing the correlation between two subspaces. To solve the obtained non-convex problem, we used an alternating optimization. We also study the connection between our model with some existing methods. We also shown the principles that we adopted in our method were widely-used in multi-label classification methods. Experiments on real-world multi-label datasets showed superior performance of our method and the necessity of exploring independence components from multi-label data. Further, we propose a stochastic independent loss, and build up a neural network version of IFLC. Which also attains superior performance.

**Acknowledgements** This work was supported in part by the Australian Research Council under Project DE180101438.

## References

1. Agrawal R, Gupta A, Prabhu Y, Varma M (2013) Multi-label learning with millions of labels: recommending advertiser bid phrases for web pages. In: Proceedings of the 22nd international conference on World Wide Web. ACM, pp 13–24
2. Andrew G, Arora R, Bilmes J, Livescu K (2013) Deep canonical correlation analysis. In: International conference on machine learning, pp 1247–1255
3. Barnard K, Duygulu P, Forsyth D, Freitas Nd, Blei DM, Jordan MI (2003) Matching words and pictures. *J Mach Learn Res* 3:1107–1135
4. Belghazi I, Rajeswar S, Baratin A, Hjelm RD, Courville A (2018) Mine., mutual information neural estimation. arXiv:1801.04062
5. Bhatia K, Jain H, Kar P, Varma M, Jain P (2015) Sparse local embeddings for extreme multi-label classification. In: Advances in neural information processing systems, pp 730–738
6. Brakel P, Bengio Y (2017) Learning independent features with adversarial nets for non-linear ica. arXiv:1710.05050
7. Chen X, Duan Y, Houthoof R, Schulman J, Sutskever I, Abbeel P (2016) Infogan: interpretable representation learning by information maximizing generative adversarial nets. In: Advances in neural information processing systems, pp 2172–2180
8. Chen YN, Lin HT (2012) Feature-aware label space dimension reduction for multi-label classification. In: Advances in neural information processing systems, pp 1529–1537
9. Du B, Wang Z, Zhang L, Zhang L, Tao D (2017) Robust and discriminative labeling for multi-label active learning based on maximum correntropy criterion. *IEEE Trans Image Process* 26(4):1694–1707
10. Elisseeff A, Weston J (2001) A kernel method for multi-labelled classification. In: International conference on neural information processing systems: natural and synthetic, pp 681–687
11. Escalante HJ, Hernández CA, Gonzalez JA, López-López A, Montes M, Morales EF, Sucar LE, Villaseñor L, Grubinger M (2010) The segmented and annotated iapr tc-12 benchmark. *Comput Vis Image Underst* 114(4):419–428

12. Guillaumin M, Mensink T, Verbeek J, Schmid C (2009) Tagprop: discriminative metric learning in nearest neighbor models for image auto-annotation. In: 2009 IEEE 12th international conference on computer vision. IEEE, pp 309–316
13. He X (2004) Locality preserving projections. *Adv Neural Informa Process Syst* 16(1):186–197
14. Hsu DJ, Kakade SM, Langford J, Zhang T (2009) Multi-label prediction via compressed sensing. In: *Advances in neural information processing systems*, pp 772–780
15. Hyvarinen A (1999) Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans Neural Netw* 10(3):626–634
16. Hyvärinen A, Karhunen J, Oja E (2004) *Independent component analysis*, vol 46. Wiley
17. Jian L, Li J, Shu K, Liu H (2016) Multi-label informed feature selection. In: *International joint conference on artificial intelligence*, pp 1627–1633
18. Kågebäck M, Mogren O (2018) Disentangled activations in deep networks. <http://mogren.one/phd/kageback2018disentanglement.pdf>
19. Katakis I, Tsoumakas G, Vlahavas I (2008) Multilabel text classification for automated tag suggestion. In: *Proceedings of the ECML/PKDD*, vol 18
20. Klimt B, Yang Y (2004) The enron corpus: a new dataset for email classification research. In: *European conference on machine learning*. Springer, pp 217–226
21. Le QV, Karpenko A, Ngiam J, Ng AY (2011) Ica with reconstruction cost for efficient overcomplete feature learning. In: *Advances in neural information processing systems*, pp 1017–1025
22. Lin Z, Ding G, Hu M, Wang J (2014) Multi-label classification via feature-aware implicit label space encoding. In: *International conference on machine learning*, pp 325–333
23. Martin N, Maes H (1979) *Multivariate analysis*. Academic Press
24. Pestian JP, Brew C, Matykiewicz P, Hovermale DJ, Johnson N, Cohen KB, Duch W (2007) A shared task involving multi-label classification of clinical free text. In: *Proceedings of the workshop on BioNLP 2007: biological, translational, and clinical language processing*. Association for Computational Linguistics, pp 97–104
25. Read J, Pfahringer B, Holmes G (2009) Multi-label classification using ensembles of pruned sets. In: *Eighth IEEE international conference on data mining*, pp 995–1000
26. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. *Mach Learn* 85(3):333–359
27. Shang S, Chen L, Wei Z, Jensen CS, Wen JR, Kalnis P (2015) Collective travel planning in spatial networks. *IEEE Trans Knowl Data Eng* 28(5):1132–1146
28. Shang S, Chen L, Zheng K, Jensen CS, Wei Z, Kalnis P (2018) Parallel trajectory-to-location join. *IEEE Trans Knowl Data Eng* 31(6):1194–1207
29. Shang S, Ding R, Zheng K, Jensen CS, Kalnis P, Zhou X (2014) Personalized trajectory matching in spatial networks. *VLDB J Int J Very Large Data Bases* 23(3):449–468
30. Sun L, Ji S, Ye J (2011) Canonical correlation analysis for multilabel classification: a least-squares formulation, extensions, and analysis. *IEEE Trans Pattern Anal Mach Intell* 33(1):194–200
31. Tai F, Lin HT (2012) Multilabel classification with principal label space transformation. *Neural Comput* 24(9):2508–2542
32. Tschannen M, Bachem O, Lucic M (2018) Recent advances in autoencoder-based representation learning. [arXiv:1812.05069](https://arxiv.org/abs/1812.05069)
33. Tsoumakas G, Katakis I, Vlahavas I (2009) Mining multi-label data. In: *Data mining and knowledge discovery handbook*. Springer, pp 667–685
34. Wang H, Ding C, Huang H (2010) Multi-label linear discriminant analysis. In: *European conference on computer vision*, pp 126–139
35. Wang W, Arora R, Livescu K, Bilmes J (2015) On deep multi-view representation learning. In: *International conference on machine learning*, pp 1083–1092
36. Wang Z, Du B, Zhang L, Zhang L, Fang M, Tao D (2016) Multi-label active learning based on maximum coreentropy criterion: towards robust and discriminative labeling. In: *European conference on computer vision*. Springer, pp 453–468
37. Xu C, Liu T, Tao D, Xu C (2016) Local rademacher complexity for multi-label learning. *IEEE Trans Image Process* 25(3):1495–1507
38. Xu C, Tao D, Xu C (2016) Robust extreme multi-label learning. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, pp 1275–1284
39. Yu K, Yu S, Tresp V (2005) Multi-label informed latent semantic indexing. In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp 258–265
40. Zhang ML, Zhou ZH (2014) A review on multi-label learning algorithms. *IEEE Trans Knowl Data Eng* 26(8):1819–1837

41. Zhang Y, Schneider J (2011) Multi-label output codes using canonical correlation analysis. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 873–882
42. Zhang Y, Zhou ZH (2008) Multi-label dimensionality reduction via dependence maximization. In: National conference on artificial intelligence, pp 1503–1505
43. Zhou WJ, Yu Y, Zhang ML (2017) Binary linear compression for multi-label classification. In: Proceedings of the 26th international joint conference on artificial intelligence. AAAI Press, pp 3546–3552

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Mengqing Mei** is now a PHD student in School of Computer Science, Wuhan University. Her research interests include machine learning and its application.



**Yongjian Zhong** is now a master student in School of Computer Science, Wuhan University. His research interests include machine learning and signal processing.





**Fazhi He** is now a professor in School of Computer, Wuhan University. His research interests include: parallel cooperative security intelligent computing principle of data processing, hardware accelerated AI computing technology, hardware and software system co-design method. He has published more than 100 academic papers in high quality journals, such as CAD, IEEE CSVT and IEEE TIP.



**Chang Xu** is Lecturer in Machine Learning and Computer Vision at the School of Information Technologies, The University of Sydney. He obtained a Bachelor of Engineering from Tianjin University, China, and a Ph.D. degree from Peking University, China. While pursuing his PhD degree, Chang received fellowships from IBM and Baidu. His research interests lie in machine learning, data mining algorithms and related applications in artificial intelligence and computer vision, including multi-view learning, multi-label learning, visual search and face recognition. His research outcomes have been widely published in prestigious journals and top-tier conferences.