Towards More Efficient Multi-label Classification Using Dependent and Independent Dual Space Reduction

Eakasit Pacharawongsakda and Thanaruk Theeramunkong

School of Information, Computer, and Communication Technology
Sirindhorn International Institute of Technology
Thammasat University, Thailand
{eakasit,thanaruk}@siit.tu.ac.th

Abstract. While multi-label classification can be widely applied for problems where multiple classes can be assigned to an object, its effectiveness may be sacrificed due to curse of dimensionality in the feature space and sparseness of dimensionality in the label space. Moreover, it suffers with high computational cost when there exist a high number of dimensions, as well as with lower accuracy when there are a number of noisy examples. As a solution, this paper presents two alternative methods, namely Dependent Dual Space Reduction and Independent Dual Space Reduction, to reduce dimensions in the dual spaces, i.e., the feature and label spaces, using Singular Value Decomposition (SVD). The first approach constructs the covariance matrix to represent dependency between the features and labels, project both of them into a single reduced space, and then perform prediction on the reduced space. On the other hand, the second approach handles the feature space and the label space separately by constructing a covariance matrix for each space to represent feature dependency and label dependency before performing SVD on dependency profile of each space to reduce dimension and for noise elimination and then predicting using their reduced dimensions. A number of experiments evidence that prediction on the reduced spaces for both dependent and independent reduction approaches can obtain better classification performance as well as faster computation, compared to the prediction using the original spaces. The dependent approach helps saving computational time while the independent approach tends to obtain better classification performance.

Keywords: multi-label classification, Singular Value Decomposition, SVD, dimensionality reduction, Problem Transformation

1 Introduction

In the past, most traditional classification techniques usually assumed a single category for each object to be classified by means of minimum distance. However, in some tasks it is natural to assign more than one categories to an object. For examples, some news articles can be categorized into both *politic*

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 383–394, 2012.

and *crime*, or some movies can be labeled as *action* and *comedy*, simultaneously. As a special type of task, multi-label classification was initially studied by Schapire and Singer (2000) [10] in text categorization. Later many techniques in multi-label classification have been proposed for various applications such as semantic scene classification [2], music emotion categorization [12] and automated tag recommendation [8]. However, these methods can be grouped into two main approaches: *Algorithm Adaptation* (AA) and *Problem Transformation* (PT) as suggested in [13]. The former approach modifies existing classification methods to handle multi-label data [4,10]. On the other hand, the latter approach transforms a multi-label classification task into several single classification tasks and then applies traditional classification method on each task [2,9,14].

Residing in these two main approaches, one major issue is curse of dimensionality, which causes a well-known overfitting problem. To solve this issue, many techniques have been proposed, e.g., sparse regularization [6], feature selection [17] and dimensionality reduction [15,16,18]. Among these methods, the dimensionality reduction which transforms data in a high-dimensional space to those in the lower-dimensional space, has been focused for multi-label classification problem. The dimensionality reduction in multi-label data was formerly studied by Yu et al. [16]. In their work, Multi-label Latent Semantic Indexing (MLSI) was proposed to project the original feature space into a reduced feature space. Motivated by MLSI, Multi-label Dimensionality Reduction via Dependence Maximization (MDDM) was introduced by Zhang and Zhou in [18]. In MDDM, Hilbert-Schmidt Independence Criterion (HSIC) was applied rather than LSI and its aim was to identify reduced feature space that maximizes dependency between the original feature space and the label space. Recently, Wang et al. [15] has proposed a method to extend Linear Discriminant Analysis (LDA), a well-known dimensionality reduction method, to handle multi-label data. Such methods mainly focused on how to project the original feature space into a smaller one, but still suffered with high dimensionality in the label space. By this reason, these methods usually have high time complexity in the classification task.

On the other hand, for the label space reduction, to improve the efficiency of multi-label classification, Hsu et al. [7] posed a sparseness problem that mostly occurred in the label space and then applied Compressive Sensing (CS) technique, widely used in the image processing field, to encode and decode the label space. While the encoding step of this CS method seems efficient but the decoding step does not. Toward this issue, Tai and Lin [11] proposed Principle Label Space Transformation (PLST) to transform the label space into a smaller linear label space using Singular Value Decomposition (SVD) [5] with a simple threshold setting (i.e., 0.5). More recently, Bi and Kwok (2011) [1] extended the PLST to handle the labels which are organized in the form of a tree or directed acyclic graph (DAG). Although the PLST-based methods are effective by reducing dimensions in the label space, it seems not handle neither the curse of dimensionality in the feature space nor the correlation (dependency) among labels in the label space.

Toward these issues, this paper presents an approach that considers both the curse of dimensionality problem in the feature space and the sparseness problem in the label space. Moreover, the dependency profile among features and labels, the dependency profile among features and the dependency profile among labels are also taken into account. Two alternative methods, namely Dependent Dual Space Reduction (DDSR) and Independent Dual Space Reduction (IDSR), are proposed to reduce dimensions in the dual spaces to eliminate redundancy as well as noise using Singular Value Decomposition (SVD) for better prediction and lower computational cost.

In the rest of this paper, Section 2 gives a formal description of the multi-label classification task and literature review to the SVD method. Section 3 presents two dual dimensionality reduction approaches, Dependent Dual Space Reduction (DDSR) and Independent Dual Space Reduction (IDSR). The multi-label benchmark datasets and experimental settings are described in Section 4. In Section 5, the experimental results using seven datasets are given and finally Section 6 provides conclusion of this work.

2 Preliminaries

2.1 Definition of Multi-label Classification Task

Let $\mathcal{X} = \mathbb{R}^M$ and $\mathcal{Y} = \{0,1\}^L$ be an M-dimensional feature space and L-dimensional binary label space, where M is the number of features and L is a number of possible labels, i.e. classes. Let $\mathcal{D} = \{\langle \mathbf{x}_1, \mathbf{y}_1 \rangle, \langle \mathbf{x}_2, \mathbf{y}_2 \rangle, ..., \langle \mathbf{x}_N, \mathbf{y}_N \rangle\}$ is a set of N objects (e.g., documents, images, etc.) in a training dataset, where $\mathbf{x}_i \in \mathcal{X}$ is a feature vector that represents an i-th object and $\mathbf{y}_i \in \mathcal{Y}$ is a label vector with the length of L, $[y_{i1}, y_{i2}, ..., y_{iL}]$. Here, y_{ij} indicates whether the i-th object belongs (1) or not (0) to the j-th class (j-th label or not).

In general, two main phases are exploited in a multi-label classification problem: (1) *model training* phase and (2) *classification* phase. The goal of the *model training* phase is to build a classification model that can predict the label vector $\mathbf{y}_t$ for a new object with the feature vector $\mathbf{x}_t$. This classification model is a mapping function $\mathcal{H} : \mathbb{R}^M \rightarrow \{0,1\}^L$ can predict a target value closest to its actual value in total. The *classification* phase uses this classification model to assign labels. For convenience, $\mathbf{X}_{N \times M} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$ denotes the *feature matrix* with N rows and M columns and $\mathbf{Y}_{N \times L} = [\mathbf{y}_1, ..., \mathbf{y}_N]^T$ represents the *label matrix* with N rows and L columns , where $[\cdot]^T$ denotes matrix transpose.

2.2 Singular Value Decomposition (SVD)

This subsection gives a brief introduction to SVD, which was developed as a method for dimensionality reduction using a least-squared technique [5]. The SVD transforms a feature matrix $\mathbf{X}$ to a lower-dimensional matrix $\mathbf{X}'$ such that the distance between the original matrix and a matrix in a lower-dimensional space (i.e., the 2-norm $\| \mathbf{X} - \mathbf{X}' \|_2$) are minimum.

Generally, a feature matrix $\mathbf{X}$ can be decomposed into the product of three matrices as shown in (1).

$$\mathbf{X}_{N\times M} = \mathbf{U}_{N\times M} \times \mathbf{\Sigma}_{M\times M} \times \mathbf{V}^T_{M\times M}, \tag{1}$$

where N is a number of objects, M is a number of features and $M < N$. The matrices $\mathbf{U}$ and $\mathbf{V}$ are two orthogonal matrices, where $\mathbf{U}^T \times \mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T \times \mathbf{V} = \mathbf{I}$. The columns in the matrix $\mathbf{U}$ are called the *left singular vectors* while as the columns in matrix $\mathbf{V}$ are called the *right singular vectors*. The matrix $\mathbf{\Sigma}$ is a diagonal matrix, where $\mathbf{\Sigma}_{i,j} = 0$ for $i \neq j$, and the diagonal elements of $\mathbf{\Sigma}$ are the singular values of matrix $\mathbf{X}$. The singular values in the matrix $\mathbf{\Sigma}$ are sorted by descending order such that $\mathbf{\Sigma}_{1,1} \geq \mathbf{\Sigma}_{2,2} \geq ... \geq \mathbf{\Sigma}_{M,M}$. To discard noise, it is possible to ignore singular values less than $\mathbf{\Sigma}_{K,K}$, where $K \ll M$. By this ignorance the three matrices are reduced to (2).

$$\mathbf{X}'_{N\times M} = \mathbf{U}'_{N\times K} \times \mathbf{\Sigma}'_{K\times K} \times \mathbf{V}'^T_{M\times K}, \tag{2}$$

where $\mathbf{X}'_{N\times M}$ is expected to be close $\mathbf{X}_{N\times M}$, i.e. $\| \mathbf{X} - \mathbf{X}' \|_2 < \delta$, $\mathbf{U}'_{N\times K}$ is a reduced matrix of $\mathbf{U}_{N\times M}$, $\mathbf{\Sigma}'_{K\times K}$ is the reduced version of $\mathbf{\Sigma}_{M\times M}$ from M to K dimensions and $\mathbf{V}'_{M\times K}$ is a reduced matrix of $\mathbf{V}_{M\times M}$.

In the next section, we show our two approaches that deploy the SVD technique to construct lower-dimensional space for both features and labels for multi-label classification.

3 Two Proposed Approaches

As mentioned earlier, most of previous approaches were presented to handle either the problem of a curse of dimensionality in the feature space or the sparseness problem in the label space.

This work presents two alternative approaches to deal with these aforementioned problems. In both approaches, the Singular Value Decomposition (SVD) is used to project both feature and label spaces into reduced spaces then a classification method can be applied. In the *classification* phase, on the other hand, SVD is used to reconstruct the original higher-dimensional label space from the prediction result in the constructed lower-dimensional label space.

In this work, we propose two alternative methods, called Dependent Dual Space Reduction (DDSR) and Independent Dual Space Reduction (IDSR). To promote the dependency among features and labels, DDSR computes dependency matrix between features and labels then applies SVD to eliminate the less correlated data. These lower-dimensional matrices computed from SVD are used to project both feature and label spaces into a common lower-dimensional space. While the DDSR approach retains only data with high correlated between features and labels, it neither considers dependency among features nor dependency among labels. As our second proposed method, the Independent Dual Space Reduction (IDSR) uses the feature dependency matrix built from the feature space and label dependency matrix computed from the label space as two projection

matrices. After that two independent SVDs are applied to these matrices and project feature space and label space to lower-dimensional spaces. The prediction can be done on lower-dimensional spaces before transforming back to the original space. The next subsections describe DDSR and IDSR in order.

3.1 Dependent Dual Space Reduction (DDSR)

As previously mentioned, it is possible to utilize the characteristic that the feature space and the label space may have some dependency with each other. While it is possible to characterize a dependency among feature and label spaces, for example, cosine similarity, entropy and symmetric uncertainty, with limited space, we considered only covariance matrix. In this work, both spaces can be simultaneously compressed by performing SVD on the feature-label covariance, viewed as a dependency profile between features and labels. Equation (3) shows construction of a covariance matrix $\mathbf{S}_{M\times L}$ to represent a dependency between feature and label spaces.

$$\mathbf{S}_{M\times L} = E[(\mathbf{X}_{N\times M} - E[\mathbf{X}_{N\times M}])^T(\mathbf{Y}_{N\times L} - E[\mathbf{Y}_{N\times L}])], \tag{3}$$

where $\mathbf{X}$ is the feature matrix, $\mathbf{Y}$ is the label matrix and $E[\cdot]$ is an expected value of the matrix.

Applying SVD, the covariance matrix $\mathbf{S}_{M\times L}$ is later decomposed to matrices $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}$. To retain only significant dimensions and reduce noise, the first K ($\leq min(M, L)$) dimensions from matrices $\mathbf{U}$ and $\mathbf{V}$ are selected as $\mathbf{U'}_{M\times K}$ and $\mathbf{V'}_{M\times K}$. Here, a lower-dimensional feature matrix $\mathbf{X'}$, can be created as $\mathbf{X'}_{N\times K} = \mathbf{X}_{N\times M} \times \mathbf{U'}_{M\times K}$. In the same way, a lower-dimensional label matrix $\mathbf{Y'}$ can be computed as $\mathbf{Y'}_{N\times K} = \mathbf{Y}_{N\times L} \times \mathbf{V'}_{L\times K}$. These tasks constitute the *pre-processing* phase.

In the next step, these two lower-dimensional matrices, $\mathbf{X'}$ and $\mathbf{Y'}$, are used for building a classification model. Among existing methods on multi-label classification, Binary Relevance (BR) is a simple approach and widely used. BR simply reduces the multi-label classification task to a set of binary classifications and then builds a classification model for each class. However, in this approach, the projected label matrix $\mathbf{Y'}$ contains numeric values rather than discrete classes. By this situation, a regression method can be applied to estimate these numeric values. While the projected label matrix $\mathbf{Y'}$ has K dimensions, it is possible to construct a regression model for each dimension. That is, K regression models are constructed for K lower-dimensions. Moreover, each model, later denoted by $r_k(\mathbf{X'})$ is a regression model built for predicting each column $\mathbf{Y'}[k]$ using the matrix $\mathbf{X'}$. While the regression model returns continuous values, we propose a method to find the optimal threshold for mapping a continuous value to binary decision (0 or 1) as described in Section 3.3.

In the *classification* phase, firstly a test feature vector $\hat{\mathbf{X}}$ is transformed to the lower-dimensional feature vector $\hat{\mathbf{X}}'$ using $\hat{\mathbf{X}}'_{1\times K} = \hat{\mathbf{X}}_{1\times M} \times \mathbf{U'}_{M\times K}$. Then this vector is fed to a series of regression models $r(\hat{\mathbf{X}}')$ to estimate the numeric value in each dimension of the predicted lower-dimensional label vector $\hat{\mathbf{Y}}'_{1\times K}[k]$.

After that, a matrix $\mathbf{V'}^T$, an orthogonal matrix of the matrix $\mathbf{V'}$, is multiplied to reconstruct the lower-dimensional label vector $\hat{\mathbf{Y}}_{1\times K}$ back to the higher-dimensional label vector $\hat{\mathbf{Y}}_{1\times L}$. Next the predicted values in the label vector $\hat{\mathbf{Y}}_{1\times L}$ are rounded to the value in {0,1} by the predefined threshold. The set of predicted multiple labels is the union of the dimensions which have the value of 1.

3.2 Independent Dual Space Reduction (IDSR)

As opposed to the former approach, the Independent Dual Space Reduction (IDSR) approach presents how to use two independent SVDs for transforming the feature space and label space into the two lower-dimensional spaces similar to DDSR even there are several possibilities of dependency calculation. In this work, to consider the dependency in the feature space, the covariance matrix $\mathbf{S}_{M\times M}$ is computed from the feature matrix $\mathbf{X}_{N\times M}$. On the other hand, the dependency among labels in the label space can be derived by calculating the covariance matrix $\mathbf{R}_{L\times L}$.

In the *pre-processing* phase of this approach, a feature dependency matrix $\mathbf{S}_{M\times M}$ is built from a feature matrix $\mathbf{X}_{N\times M}$ and then it is decomposed to three matrices $\mathbf{U_x}$, $\mathbf{\Sigma_x}$ and $\mathbf{V_x}$ and select the top D dimensions from the matrix $\mathbf{U_x}$. Then the lower-dimensional feature matrix $\mathbf{X'}$ can be constructed by $\mathbf{X'}_{N\times D} = \mathbf{X}_{N\times M} \times \mathbf{U'}_{\mathbf{x}M\times D}$. The label dependency matrix $\mathbf{R}_{L\times L}$ is constructed from the label matrix $\mathbf{Y}$. Likewise, this matrix is decomposed to three matrices $\mathbf{U_y}$, $\mathbf{\Sigma_y}$ and $\mathbf{V_y}$ and the top K dimensions are selected from the matrix $\mathbf{U_y}$. The lower-dimensional label matrix $\mathbf{Y'}$ can be formulated by $\mathbf{Y'}_{N\times K} = \mathbf{Y}_{N\times L} \times \mathbf{U'}_{\mathbf{y}L\times K}$. While the original label matrix $\mathbf{Y}_{N\times L}$ contains either 0 or 1 as its members, its lower-dimensional label matrix $\mathbf{Y'}_{N\times K}$ may include non-binary numeric values. Moreover, it is not necessary that the dimension of the lower-dimensional feature space D and that of the lower-dimensional label space K are identical. Note that this condition is not the same with the DDSR approach, where D always equals to K. After that, as the *model training* phase, we can construct K regression models to predict $\mathbf{Y'}_{N\times K}$ from $\mathbf{X'}_{N\times D}$. Note that each regression model is for each of K dimensions of $\mathbf{Y'}$. To transform a numeric value to a binary value a threshold is established. Section 3.3 describes our proposed method for searching the best threshold for each label. This step is done in the *model training* phase.

In the *classification* phase, the feature vector $\hat{\mathbf{X}}$ of an unseen object will be reduced to the lower-dimensional feature vector $\hat{\mathbf{X}}'$ using $\hat{\mathbf{X}}'_{1\times D} = \hat{\mathbf{X}}_{1\times M} \times \mathbf{U'}_{\mathbf{x}M\times D}$. Then the regression models estimate the numeric value in the lower-dimensional label vector $\hat{\mathbf{Y}}'$ based on the feature vector $\hat{\mathbf{X}}'$. Next the matrix $\mathbf{U'_y}^T$, an orthogonal matrix of the matrix $\mathbf{U'_y}$, is used to reconstruct the original higher-dimensional label vector $\hat{\mathbf{Y}}$ from the prediction result in the lower-dimensional label vector $\hat{\mathbf{Y}}'$ i.e., $\hat{\mathbf{Y}}_{N\times L} = \hat{\mathbf{Y}}_{N\times K} \times \mathbf{U'}^T_{\mathbf{y}L\times K}$. To assign labels to an unseen object, the prediction values in the label vector $\hat{\mathbf{Y}}$ need to be rounded to {0,1}. At this point, the threshold found in the *model training* phase can be applied. Finally, the assigned label set is the union set of the dimensions that have the value of 1.

3.3 Threshold Selection

As stated above, by the orthogonal property of SVD, it can be used to reconstruct an original label space from a lower-dimensional label space. As the result, the reconstructed label vector may include non-binary values. To interpret the values as binary decision, a threshold need to be set to map these values to either 0 or 1 for representing whether the object belongs to the class or not. As a naive approach, the fixed value of 0.5 is used to assign 0 if the value is less than 0.5, otherwise 1 [11]. As a more efficient method, it is possible to apply an adaptive threshold. In this work, we propose a method to determine an optimal threshold by selecting the value that maximizes classification accuracy in the training dataset that is similar to the mechanism in Han et al [6]. In other words, the threshold selection is done by first sorting prediction values in each label dimension in a descending order and examine performance (e.g., *macro F-measure*) for each rank position from the top to the bottom to find the point that maximize the performance. Then, the threshold for binary decision is set based on that point.

4 Datasets and Experimental Settings

To evaluate the performance of our two proposed approaches, the benchmark multi-label datasets are downloaded from MULAN[1]. Table 1 shows the characteristics of seven multi-label datasets. For each dataset, N, M and L denote the total number of objects, the number of features and the number of labels, respectively. L_C represents the *label cardinality*, the average number of labels per example and L_D stands for *label density*, the normalized value of *label cardinality* as introduced by Read et al [9].

Table 1. Characteristics of the datasets used in our experiments

Dataset	Domain	N	M		L	L_C	L_D
			Nominal	Numeric			
bibtex	text	7,395	1,836	-	159	2.402	0.015
corel5k	images	5,000	499	-	374	3.522	0.009
enron	text	1,702	1,001	-	53	3.378	0.064
medical	text	978	1,449	-	45	1.245	0.028
emotions	music	593	-	72	6	1.869	0.311
scene	image	2,407	-	294	6	1.074	0.179
yeast	biology	2,417	-	103	14	4.237	0.303

Since each object in the dataset can be associated with multiple labels simultaneously, the traditional evaluation metric of single-label classification could not be applied. The well-known multi-label evaluation metrics are of two types [9]. As the first type, a *label-based* metric evaluates each label separately such

[1] `http://mulan.sourceforge.net/datasets.html`

as *hamming loss* and *macro F-measure*. As the second type, a *label set-based* metric considers a set of labels simultaneously, i.e., *accuracy* and *0/1 loss*. In this work, *hamming loss*, *macro F-measure*, *accuracy* and *0/1 loss* are used to assess the effectiveness of the multi-label classification methods. Their detailed descriptions can be found in several literatures such as those in Read et al [9].

Table 2. Performance comparison (mean) between BR, BR+, CC, PLST, DDSR, and IDSR in terms of *hamming loss (HL), macro F-measure (F1), accuracy, 0/1 loss* on the seven datasets. (↓ indicates the smaller the better; ↑ indicates the larger the better; † denotes the nominal-feature dataset and ‡ denotes the numeric-feature dataset, the superscript (x,y) shows the percentage of dimensions reduced from the original ones for the feature space and that for the label space.)

Dataset	Metrics	BR	BR+	CC	PLST	DDSR	IDSR
bibtex†	HL ↓	0.0172	0.0163	0.0166	$0.0155^{[20]}$	$\mathbf{0.0137}^{(9,100)}$	$0.0140^{(40,100)}$
	F1 ↑	0.0047	0.0022	0.0038	$0.0075^{[100]}$	$0.3949^{(9,100)}$	$\mathbf{0.4025}^{(80,100)}$
	Accuracy ↑	0.0000	0.0001	0.0001	$0.0012^{[100]}$	$0.1543^{(3,40)}$	$\mathbf{0.3699}^{(80,100)}$
	0/1 Loss ↓	1.0000	1.0000	1.0000	$0.9999^{[40]}$	$\mathbf{0.8270}^{(9,100)}$	$0.8301^{(60,100)}$
corel5k†	HL ↓	0.0094	0.0195	0.0094	$\mathbf{0.0094}^{[20]}$	$0.0145^{(30,40)}$	$0.0150^{(60,100)}$
	F1 ↑	0.0284	0.0710	0.0354	$0.0286^{[40]}$	$\mathbf{0.1215}^{(15,20)}$	$0.1143^{(60,100)}$
	Accuracy ↑	0.0528	0.1432	0.0584	$0.0533^{[40]}$	$0.1479^{(60,80)}$	$\mathbf{0.1502}^{(60,100)}$
	0/1 Loss ↓	0.9948	0.9938	**0.9918**	$0.9944^{[20]}$	$0.9944^{(45,60)}$	$0.9948^{(60,100)}$
enron†	HL ↓	0.1019	0.0997	0.1014	$0.0721^{[20]}$	$\mathbf{0.0529}^{(3,60)}$	$0.0532^{(20,100)}$
	F1 ↑	0.1957	0.1882	0.1920	$0.2170^{[40]}$	$\mathbf{0.3118}^{(5,100)}$	$0.2926^{(20,100)}$
	Accuracy ↑	0.3328	0.3274	0.3317	$0.3475^{[20]}$	$\mathbf{0.4723}^{(5,100)}$	$0.4617^{(20,100)}$
	0/1 Loss ↓	0.9089	0.9083	0.9066	$0.9077^{[60]}$	$\mathbf{0.8713}^{(5,100)}$	$0.8766^{(60,100)}$
medical†	HL ↓	0.0996	0.0943	0.0974	$0.0556^{[20]}$	$0.0114^{(3,100)}$	$\mathbf{0.0107}^{(20,100)}$
	F1 ↑	0.3383	0.3370	0.3351	$0.3411^{[60]}$	$0.6485^{(3,100)}$	$\mathbf{0.6815}^{(40,100)}$
	Accuracy ↑	0.4017	0.3997	0.4060	$0.4074^{[40]}$	$0.7288^{(3,100)}$	$\mathbf{0.7603}^{(20,100)}$
	0/1 Loss ↓	0.7167	0.7085	0.7095	$0.7167^{[100]}$	$0.3731^{(3,100)}$	$\mathbf{0.3507}^{(20,100)}$
emotions‡	HL ↓	0.2068	0.2264	0.2211	$\mathbf{0.2037}^{[60]}$	$0.2728^{(8,100)}$	$0.2152^{(100,60)}$
	F1 ↑	0.6317	0.6673	0.6243	$0.6339^{[60]}$	$0.6455^{(8,100)}$	$\mathbf{0.6804}^{(100,20)}$
	Accuracy ↑	0.5091	**0.5537**	0.5176	$0.5091^{[100]}$	$0.5090^{(8,100)}$	$0.5534^{(60,40)}$
	0/1 Loss ↓	0.7467	**0.7267**	0.7451	$0.7300^{[60]}$	$0.8293^{(8,100)}$	$0.7417^{(60,60)}$
scene‡	HL ↓	**0.1105**	0.2583	0.1162	$\mathbf{0.1105}^{[100]}$	$0.1190^{(2,80)}$	$0.1113^{(60,80)}$
	F1 ↑	0.6480	0.5351	0.6903	$0.6480^{[100]}$	$0.7046^{(2,100)}$	$\mathbf{0.7201}^{(20,80)}$
	Accuracy ↑	0.5302	0.5744	**0.6579**	$0.5302^{[100]}$	$0.6109^{(2,80)}$	$0.6451^{(20,80)}$
	0/1 Loss ↓	0.5186	0.5148	**0.3831**	$0.5186^{[100]}$	$0.5106^{(2,80)}$	$0.4778^{(40,80)}$
yeast‡	HL ↓	**0.2008**	0.2229	0.2165	$0.2491^{[20]}$	$0.2807^{(14,100)}$	$0.2626^{(80,100)}$
	F1 ↑	0.4455	0.4053	0.4404	$0.2688^{[20]}$	$0.4926^{(3,20)}$	$\mathbf{0.4927}^{(40,100)}$
	Accuracy ↑	0.5019	0.4838	0.4796	$0.3425^{[20]}$	$0.4884^{(8,60)}$	$\mathbf{0.5021}^{(60,60)}$
	0/1 Loss ↓	0.8510	0.8564	**0.8105**	$0.9785^{[40]}$	$0.9259^{(14,100)}$	$0.9086^{(80,100)}$

In this work, DDSR and IDSR are compared with four multi-label classification techniques; BR, BR+ [3], CC [9] and PLST [11]. BR+ (Binary Relevance with label dependency consideration) and CC (Classifier Chains) are two well-known methods, which incorporate label dependency in multi-label classification. PLST (Principle Label Space Transformation) is an efficient algorithm that uses the reduction of label space dimension. Using ten-fold cross validation method, the results of the four evaluation metrics and the execution time are recorded and shown in Table 2 and 3, respectively. All multi-label classification methods used in this work is implemented in R environment version 2.11.1[2] and linear

[2] `http://www.R-project.org/`

regression is used for the regression model in the *model training* phase. For PLST and IDSR method, we experiments with K ranging from 20% to 100% of the dimension of the original label matrix, with 20% as interval. Likewise, the parameter D is also varied from 20% to 100% of the dimension of the original feature matrix, with 20% as interval. Though, the K parameter in DDSR approach is calculated from the minimum value between a number of features and labels, this parameter is also used the same criteria as PLST and IDSR method. To compare the computational time, all methods were performed on the AMD Opteron Quad Core 8356 1.1 GHz Processor with 512 KB of cache, 64GB RAM.

5 Experimental Results

To evaluate our proposed approaches, seven datasets are used to compare performance of BR, BR+, CC, PLST, DDSR and IDSR. Table 2 reports the best value for each evaluation metric computed from all datasets. The numbers in the superscript (x,y) represents the percentages of dimensions reduced in the feature space and that in the label space, respectively. In the DDSR method, the maximum number of reduced dimensions K cannot excess the minimum between the number of features (M) and the number of labels (L) since the reduced dimension has the same size of both feature and label space. The superscript [y] in the PLST approach means the percentage of reduce labels, compared to the original. Note that PLST does not reduce the feature space. In the Table 2, the best value for each row is emphasized by bold font.

From the table, we can make some observations as follows. First, we observe that both DDSR and IDSR give comparable performance in terms of *hamming loss*, compared to BR, BR+, CC and PLST. On the other hand, DDSR and IDSR approaches gain an average gap of 16% *macro F-measure* increment. Moreover, the DDSR approach shows an average gap of 11% *accuracy* improvement while the IDSR approach improves with an average gap of 15%. Likewise, the DDSR approach can reduce the *0/1 loss* with decrement of 5% while IDSR can reduce with 8% gap. Note that the *medical* dataset whose number of features are greater than the number of objects, gives the maximum improvement when both spaces are reduced.

As shown in Table 3, the execution time of the DDSR approach was reduced with a factor of 10, compared to PLST and approximately 100 times, compared to the traditional BR approach. Likewise, the IDSR approach with lower-dimensional features used less time than the PLST and the BR method. We can conclude that our two proposed methods, DDSR and IDSR, could transform the feature and label spaces into the reduced spaces with less computational time than the traditional BR, BR+, CC and PLST. As an additional observation, IDSR is better than DDSR in several datasets while DDSR can be executed faster than IDSR. The smaller K and D are, the faster we can compute.

In more details, Table 4 presents the complexity of learning process. However, the time used for the transformation process and covariance calculation is trivial. The N, M and L denote the number of objects, features and labels, respectively.

Table 3. Average execution time (in seconds) on the seven datasets. † denotes the nominal-feature dataset and ‡ denotes the numeric-feature dataset. Here, D is the reduced number of feature dimensions. K is the reduced number of label dimensions.

Dataset	Method	K		
		20%×L	60%×L	100%×L
bibtex†	BR	-	-	40442.05
	BR+	-	-	33537.76
	CC	-	-	19700.05
	PLST	7101.01	21961.42	29815.07
	DDSR	69.85	126.56	283.36
	IDSR (D=20%×M)	1338.02	2360.49	3855.54
	IDSR (D=60%×M)	4656.35	14636.93	18787.84
	IDSR (D=100%×M)	14800.45	27392.90	46642.77
corel5k†	BR	-	-	5721.81
	BR+	-	-	11971.31
	CC	-	-	7007.41
	PLST	1172.75	3174.11	5386.21
	DDSR	69.73	366.07	1265.54
	IDSR (D=20%×M)	364.17	560.04	818.53
	IDSR (D=60%×M)	817.59	2667.54	4572.47
	IDSR (D=100%×M)	1905.29	5273.13	6397.23
enron†	BR	-	-	656.94
	BR+	-	-	1904.89
	CC	-	-	799.58
	PLST	197.43	537.04	824.20
	DDSR	4.62	6.57	11.56
	IDSR (D=20%×M)	29.85	41.03	50.12
	IDSR (D=60%×M)	60.48	120.46	175.46
	IDSR (D=100%×M)	99.27	233.00	355.83
medical†	BR	-	-	1353.93
	BR+	-	-	2896.23
	CC	-	-	1414.56
	PLST	192.49	509.45	1240.37
	DDSR	1.35	2.57	4.76
	IDSR (D=20%×M)	238.56	256.15	234.38
	IDSR (D=60%×M)	170.31	324.86	383.25
	IDSR (D=100%×M)	449.11	735.24	982.16
emotions‡	BR	-	-	1.10
	BR+	-	-	4.09
	CC	-	-	0.87
	PLST	0.19	0.47	0.79
	DDSR	0.41	0.47	0.53
	IDSR (D=20%×M)	0.48	0.56	0.59
	IDSR (D=60%×M)	0.51	0.68	0.85
	IDSR (D=100%×M)	0.57	1.00	1.27
scene‡	BR	-	-	9.13
	BR+	-	-	57.33
	CC	-	-	22.26
	PLST	2.31	12.08	10.80
	DDSR	2.70	2.79	2.87
	IDSR (D=20%×M)	3.52	4.09	4.47
	IDSR (D=60%×M)	4.80	6.84	8.13
	IDSR (D=100%×M)	6.54	10.72	13.24
yeast‡	BR	-	-	16.23
	BR+	-	-	37.98
	CC	-	-	15.62
	PLST	3.93	9.23	10.15
	DDSR	10.52	10.82	11.37
	IDSR (D=20%×M)	11.68	11.99	12.04
	IDSR (D=60%×M)	11.82	12.85	14.26
	IDSR (D=100%×M)	12.33	14.15	16.20

For our two approaches, D and K are the reduced number of dimensions. The $f(X, Y)$ is the complexity of the model that depends on the number of objects (X) and the number of features (Y). When linear regression is applied for the *model training* phase, it requires $O(4XY^2 + X^3 + 2XY)$ and $O(Y)$ for the *classification* phase. We can observe that the BR+ method is recognized as the slowest algorithm since it appends labels to the feature space for incorporating label dependency and it requires two learning process, initial prediction step and final prediction step, to complete the classification process. On the other hand,

Table 4. The learning complexity of BR, BR+, CC, PLST, DDSR and IDSR. Note that the complexity is in the function $f(X, Y)$, where X is the number of objects and Y is the number of features.

Methods	Complexity (O)
BR	$O(L \times f(N, M))$
BR+	$O((L \times f(N, M)) + (L \times f(N, (M + L - 1))))$
CC	$O(L \times f(N, (M + L/2)))$
PLST	$O(K \times f(N, M))$
DDSR	$O(K \times f(N, K))$
IDSR	$O(K \times f(N, D))$

our DDSR approach is the fastest method because the feature and label spaces are transformed to the lower-dimensional space before classification technique is applied.

6 Conclusion

This paper presents two alternative approaches to handle the curse of dimensionality problem in the feature space as well as the sparseness problem in the label space. The Dependent Dual Dimensionality Reduction (DDSR) considers the dependency between feature and label spaces before transforming the feature and label spaces into a single reduced space. On the other hand, the Independent Dual Space Reduction (IDSR) approach transforms the feature space and label space into the two lower-dimensionality spaces. Experiments with a broad range of multi-label datasets show that our two proposed approaches achieve a better performance, compared to PLST and BR, as well as other recent methods such as Classifier Chains (CC) and BRplus (BR+). In addition, the DDSR approach helps saving computational time while the IDSR approach tends to obtain better classification performance As our future work, we will analyze three dependencies, feature-label, feature-feature, and label-label, in detail. The ensemble of these dependencies may help in improving the performance.

Acknowledgement. This work has been supported by the TRF Royal Golden Jubilee Ph.D. Program [PHD/0304/2551] and partially supported by the National Research University Project of Thailand Office of Higher Education Commission as well as the National Electronics and Computer Technology Center (NECTEC) under Project Number NT-B-22-KE-38-54-01.

References

1. Bi, W., Kwok, J.: Multi-label classification on tree- and dag-structured hierarchies. In: Getoor, L., Scheffer, T. (eds.) Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 17–24. ACM, New York (2011)
2. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004)

3. Cherman, E.A., Metz, J., Monard, M.C.: Incorporating label dependency into the binary relevance framework for multi-label classification. Expert Systems with Applications (2011)
4. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Proceedings of the Advances in Neural Information Processing Systems, vol. 14, pp. 681–687. MIT Press (2001)
5. Golub, G., Reinsch, C.: Singular value decomposition and least squares solutions. Numerische Mathematik 14, 403–420 (1970)
6. Han, Y., Wu, F., Jia, J., Zhuang, Y., Yu, B.: Multi-task sparse discriminant analysis (mtsda) with overlapping categories. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, pp. 469–474 (2010)
7. Hsu, D., Kakade, S., Langford, J., Zhang, T.: Multi-label prediction via compressed sensing. In: Proceedings of the Advances in Neural Information Processing Systems, vol. 22, pp. 772–780 (2009)
8. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: Proceedings of the the ECML/PKDD 2008 Discovery Challenge (2008)
9. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning, 1–27 (2011)
10. Schapire, R., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning 39(2/3), 135–168 (2000)
11. Tai, F., Lin, H.T.: Multi-label classification with principle label space transformation. In: Proceedings of the 2nd International Workshop on Learning from Multi-Label Data (MLD 2010), pp. 45–52 (2010)
12. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multi-label classification of music into emotions. In: Proceedings of the International Symposium/Conference on Music Information Retrieval, pp. 325–330 (2008)
13. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Data Mining and Knowledge Discovery Handbook, 2nd edn. Springer (2010)
14. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multilabel classification. IEEE Transactions on Knowledge and Data Engineering 23, 1079–1089 (2011)
15. Wang, H., Ding, C., Huang, H.: Multi-label Linear Discriminant Analysis. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 126–139. Springer, Heidelberg (2010)
16. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 258–265 (2005)
17. Zhang, M.L., Pea, J.M., Robles, V.: Feature selection for multi-label naive bayes classification. Information Science, 3218–3229 (2009)
18. Zhang, Y., Zhou, Z.H.: Multilabel dimensionality reduction via dependence maximization. ACM Transactions on Knowledge Discovery from Data (TKDD) 4(3), 1–21 (2010)