Multi-label Subspace Ensemble

Tianyi Zhou
Centre for Quantum Computation & Intelligent Systems
University of Technology Sydney, Australia
tianyi.zhou@student.uts.edu.au

Dacheng Tao
Centre for Quantum Computation & Intelligent Systems
University of Technology Sydney, Australia
dacheng.tao@uts.edu.au

Abstract

A challenging problem of multi-label learning is that both the label space and the model complexity will grow rapidly with the increase in the number of labels, and thus makes the available training samples insufficient for training a proper model. In this paper, we eliminate this problem by learning a mapping of each label in the feature space as a robust subspace, and formulating the prediction as finding the group sparse representation of a given instance on the subspace ensemble. We term this approach as "multi-label subspace ensemble (MSE)". In the training stage, the data matrix is decomposed as the sum of several low-rank matrices and a sparse residual via a randomized optimization, where each low-rank part defines a subspace mapped by a label. In the prediction stage, the group sparse representation on the subspace ensemble is estimated by group *lasso*. Experiments on several benchmark datasets demonstrate the appealing performance of MSE.

1 Introduction

Multi-label learning [15][9][5] (ML) predicts multiple labels that characterize an instance from a set of possible labels. Conventional multi-label learning algorithms aim to find a mapping from the feature space $\mathcal{X} \subseteq \mathbb{R}^p$ to the label space $\mathcal{Y} \subseteq \{0,1\}^k$, wherein k is the number of labels and $y_i = 1, y \in \mathcal{Y}$ means the sample belongs to label i. Binary relevance (BR) [11] and label powerset (LP) [11] are two natural approaches. BR relaxes ML to k independent binary classifications on the k labels respectively, while LP frames ML as a multi-class classification problem, where each class denotes a unique k-dimensional label vector. Both BR and LP do not duly explore the characteristics of ML, because BR ignores the label correlations, and LP makes the training samples of each class far less than the prerequisite.

A central problem limiting ML is that the label space $\mathcal{Y}$ will exponentially grow with the increase in the number of labels, i.e., k labels lead to a search in a label space $\mathcal{Y}$ of size 2^k in the prediction. BR independently predicts each dimension of the k-dimensional label vector y in isolation and thus does not encounter this problem with the price of ignoring the label correlations. LP attempts to distinguish each element in $\mathcal{Y}$ from the other $2^k - 1$ ones. Thus the size of the training set, which is large enough for binary classification, will be insufficient for multi-label prediction. This problem also leads to a rapid growth of the model complexity, which increases the training costs. By viewing the problem from the perspective of probabilistic approaches, the exponential growth of $\mathcal{Y}$ drastically enlarges the parameter space for modeling $P(y|x), x \in \mathcal{X}$, which makes ML intractable in computation. Another important problem is that, for a given training set $\{X, Y\}$, the instances in Y often scatters sparsely in the ambient space $\mathcal{Y}$. It is therefore difficult to study the structure of $\mathcal{Y}$ and reduce its dimensionality.

Most recent multi-label learning approaches [20][19] investigate the label correlations (or dependencies) to build a structured classification model. They partially solve the first problem by reducing the size of the search space $\mathcal{Y}$. For example, the random k-labelsets (RAkEL) method [14] randomly selects an ensemble of subsets from the original labelsets (the set of labels one instance belongs to), and then LP is applied to each subset. The final prediction is obtained by ranking and thresholding of the results on the subsets. Hierarchical binary relevance (HBR) [1] builds a general-to-specific tree structure of labels, where a sample with a label must be associated with its parent labels. A binary classifier is trained on each non-root label. Hierarchy of multi-label classifiers (HOMER) [12] recursively partitions the labels into several subsets and builds a tree-shaped hierarchy. A binary classifier is trained on each non-root label subset. The classifier chain (CC) [10] adopts a greedy method to predict unknown label from features and pre-

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands, Spain. Volume 11 of JMLR: W&CP 11.

dicted labels by using a binary classifier.

However, the size of the search space in these approaches is much larger than $\mathcal{O}(k)$, and their model complexities are too high to be applied in practice. Also, the instances in Y are always insufficient to generate a reliable estimation of the label space structure, and thus weaken the effectiveness of the structured classification models used in these approaches.

Some existing learning methods except binary classification are reformulated and then can be extended to multi-label prediction problem. For example, the C&W procedure [2] separates multi-label prediction into two stages, i.e., BR and correction of the BR results by using the label dependence. Regularized multi-task learning [3] and shared-subspace learning [6] formulate the problem as regularized regression or classification problem. Multi-label dimensionality reduction via dependence maximization (MDDM) [21] maximizes the dependence between feature space and label space, and provides a data preprocessing for other multi-label learning methods. A linear dimensionality reduction method for multi-label data is proposed in [7]. In [5], multi-label prediction is formulated as a sparse signal recovery problem.

However, these methods cannot provide an explicit modeling of the label correlations (or dependence) and thus their performance improvements due to exploring label structure are limited. Moreover, they bring extra time costs to the training process, so the efficiency is weakened.

In this paper, we consider the ML problem in a novel manner: we study the mapping of each label as a feature subspace. In other words, we assume each instance x exists in the ensemble of subspaces defined by the labels that x belongs to. However, it is not always guaranteed that each instance can be completely explained by the labels we consider, so a method should be developed to separate the parts that can be explained by the considered labels and the part that cannot. The label correlations are naturally preserved in the subspace ensemble. Given a new instance, its labels are predicted by estimating its group sparse representation in the subspace ensemble, where the nonzero entries are associated with the predicted labels. There are only k subspaces that are demanded, so the model complexity is small. The prediction is accomplished by searching in the subspace ensemble, and thus avoids the estimation of the label space structure.

We therefore develop "multi-label subspace ensemble (MSE)" to solve the above problem. In the training stage, we develop a randomized decomposition of the training data X, where X is factorized to the sum of k low-rank parts and a sparse residual. Each of the low-rank part defines the subspace mapped by a particular label, while the sparse residual stores the part that cannot be explained by the considered labels. The decomposition is fast due to an application of the bilateral random projection (BRP) based low-rank approximation [22]. Its convergence to local optimum is proved. In the prediction stage of MSE, group *lasso* estimates the group sparse representation of a given instance in the subspace ensemble, and the nonzero entries indicates the predicted labels. The experiments on several benchmark datasets for ML imply the competitive effectiveness and efficiency of MSE.

The rest of the paper is organized as follows. Section 2 introduces the MSE model, which explains the mapping of the label in the feature space and includes the assumption of MSE to the multi-label data. Section 3 presents the training algorithm of MSE via randomized matrix decomposition, which produces the ensemble of the subspaces. Section 4 presents the prediction algorithm in MSE by exploring the group sparse representation of a multi-label sample on the subspace ensemble. Section 5 shows the experimental results of MSE on 13 benchmark datasets. Section 6 gives a discussion and concludes the paper.

2 MSE model

Given a sample $x \in \mathbb{R}^p$ and its label vector $y \in \{0,1\}^k$, we assume that x can be decomposed as the sum of several components l^i and a sparse residual s

$$x = \sum_{i:y_i=1} l^i + s. \tag{1}$$

The component l^i is caused by the label i that x belongs to. Thus l^i can be explained as the mapping from the label i in x to the feature space. The residual s is the component that all the labels in y cannot explain. The model in (1) reveals the general relationship between the feature space and the label space.

For all the samples with label i, we assume their components explained by label i lie in a linear subspace $C^i \in \mathbb{R}^{r^i \times p}$, i.e., $l^i = \beta_{G_i} C^i$, wherein β_{G_i} is the representation coefficients corresponding to C^i. Thus the model (1) can be equivalently written as

$$\begin{aligned} x \quad &= \sum_{i=1}^{k} \beta_{G_i} C^i + s, \\ &\forall i \in \{i : y_i = 0\}, \beta_{G_i} = \mathbf{0}. \end{aligned} \tag{2}$$

If we build the subspace ensemble $C = [C^1; \ldots; C^k]$ characterized by the k labels as a dictionary for x, the corresponding representation coefficient vector for x is $\beta = [\beta_{G_1}, \ldots, \beta_{G_k}]$. The coefficients β_{G_i} corresponding to the labels that x does not belong to are zeros, so β is group sparse, wherein the groups are $G_i, i = 1, \ldots, k$.

In the training stage of MSE, we learn the subspace ensemble $C^i, i = 1, \ldots, k$ from the training data X via a randomized decomposition of X, in which the components explained by label i from all the samples consists a low-rank

matrix $L^i_{\Omega_i}$, wherein Ω_i is the index set of training samples with label i. Thus the row space of $L^i_{\Omega_i}$ is the subspace C^i. In the prediction stage of MSE, given a new instance x, we use group *lasso* to find the group sparse representation β on the subspace ensemble C, and then a simple thresholding is used to test which groups that β concentrates on. The labels that these groups correspond to are the predicted labels for the instance x.

In the training stage of MSE, the label correlations is naturally preserved in the subspace ensemble C, because all the subspaces are jointly learned. Specifically, if two labels i and j simultaneously appear for many times in the training samples, then Ω_i and Ω_j will have many shared elements. Thus the row spaces of $L^i_{\Omega_i}$ and $L^i_{\Omega_j}$, i.e., C^i and C^j, will be close to each other. In the prediction stage, both discriminative and structured information encoded in the subspace ensemble are considered via group *lasso*. Since only k subspaces are learned in the training stage, MSE explores label correlations without increasing the model complexity.

3 MSE training: randomized decomposition

In this section, we introduce the training stage of MSE, which approximately decomposes the training data matrix $X \in \mathbb{R}^{n\times p}$ into $X = \sum_{i=1}^k L^i + S$. For the matrix L^i, the rows corresponding to the samples with label i are nonzero, while the other rows are all-zero vectors. The nonzero rows denote the components explained by label i in the feature space. We use Ω_i to denote the index set of samples with label i in the matrix X and L^i, and then the matrix composed of the nonzero rows in L^i is represented by $L^i_{\Omega_i}$. In the decomposition, the rank of $L^i_{\Omega_i}$ is upper bounded, which indicates that all the components explained by label i nearly lies in a linear subspace. The matrix S is the residual of the samples that cannot be explained by the given labels. In the decomposition, the cardinality of S is upper bounded, which makes S sparse.

If the label matrix of X is $Y \in \{0,1\}^{n\times k}$, the rank of $L^i_{\Omega_i}$ is upper bounded by r^i and the cardinality of S is upper bounded by K, the decomposition can be written as solving the following constrained minimization problem:

$$\begin{array}{ll}\min\limits_{L^i,S} & \left\|X-\sum_{i=1}^k L^i - S\right\|_F^2 \\ s.t. & \mathrm{rank}\left(L^i_{\Omega_i}\right) \le r^i, L^i_{\overline{\Omega}_i} = \mathbf{0}, \forall i = 1,\ldots,k \\ & \mathrm{card}\left(S\right) \le K.\end{array} \tag{3}$$

Therefore, each training sample in X is decomposed as the sum of several components, which respectively correspond to multiple labels that the sample belongs to. MSE separates these components from the original sample by building the mapping from the labels to the feature space. For label i, we obtain its mapping in the feature space as the row space of $L^i_{\Omega_i}$.

3.1 Alternating minimization

Although the rank constraint to $L^i_{\Omega_i}$ and cardinality constraint to S are not convex, the optimization in (3) can be solved by alternating minimization that decomposes it as the following $k+1$ subproblems, each of which has the global solution:

$$\begin{cases} L^i_{\Omega_i} = \arg\min\limits_{\mathrm{rank}\left(L^i_{\Omega_i}\right)\le r^i} \left\|X - \sum\limits_{j=1,j\neq i}^k L^j - S - L^i\right\|_F^2, \\ \quad \forall i = 1,\ldots,k. \\ S = \arg\min\limits_{\mathrm{card}(S)\le K} \left\|X - \sum\limits_{j=1}^k L^j - S\right\|_F^2. \end{cases} \tag{4}$$

The solutions of $L^i_{\Omega_i}$ and S in the above subproblems can be obtained via hard thresholding of singular values and the matrix entries, respectively. Note that both SVD and matrix entry-wise hard thresholding have global solutions. In particular, $L^i_{\Omega_i}$ is built from the first r^i largest singular values and the corresponding singular vectors of $\left(X - \sum_{j=1,j\neq i}^k L^j - S\right)_{\Omega_i}$, while S is built from the K entries with the largest absolute value in $X - \sum_{j=1}^k L^j$, i.e.,

$$\begin{cases} L^i_{\Omega_i} = \sum\limits_{q=1}^{r^i} \lambda_q U_q V_q^T, i = 1,\ldots,k, \\ \mathrm{svd}\left[\left(X - \sum_{j=1,j\neq i}^k L^j - S\right)_{\Omega_i}\right] = U\Lambda V^T; \\ S = \mathcal{P}_\Phi\left(X - \sum\limits_{j=1}^k L^j\right), \Phi : \left|\left(X - \sum\limits_{j=1}^k L^j\right)_{r,s\in\Phi}\right| \neq 0 \\ \quad \text{and } \ge \left|\left(X - \sum\limits_{j=1}^k L^j\right)_{r,s\in\overline{\Phi}}\right|, |\Phi| \le K. \end{cases} \tag{5}$$

The projection $S = \mathcal{P}_\Phi(R)$ represents that the matrix S has the same entries as R on the index set Φ, while the other entries are all zeros.

The decomposition is then obtained by iteratively solving these $k+1$ subproblems in (4) according to (5). In this paper, we initialize $L^i_{\Omega_i}$ and S as

$$\begin{cases} L^i_{\Omega_i} := Z_{\Omega_i}, i = 1,\ldots,k, \\ Z = D^{-1}X, D = \mathrm{diag}\left(Y\mathbf{1}\right); \\ S := \mathbf{0}. \end{cases} \tag{6}$$

In each subproblem, only one variable is optimized with the other variables fixed. The convergence of this alternating minimization can be proved in Theorem 1 by demonstrating that the approximation error keeps monotonically decreasing throughout the algorithm.

Theorem 1. *The alternating minimization of subproblems (4) produces a sequence of $\|X - \sum_{i=1}^k L^i - S\|_F^2$ that converges to a local minimum.*

Proof. Let the objective value (decomposition error) $\|X - \sum_{i=1}^{k} L^i - S\|_F^2$ after solving the $k+1$ subproblems in (4) be $E^1_{(t)}, \ldots, E^{k+1}_{(t)}$ respectively for the t^{th} iteration round. We use subscript (t) to signify the variable that is updated in the t^{th} iteration round. Then $E^1_{(t)}, \ldots, E^{k+1}_{(t)}$ are

$$E^1_{(t)} = \left\| X - S_{(t-1)} - L^1_{(t)} - \sum_{i=3}^{k} L^i_{(t-1)} - L^2_{(t-1)} \right\|_F^2, \tag{7}$$

$$E^2_{(t)} = \left\| X - S_{(t-1)} - L^1_{(t)} - \sum_{i=3}^{k} L^i_{(t-1)} - L^2_{(t)} \right\|_F^2, \tag{8}$$

$$\vdots$$

$$E^k_{(t)} = \left\| X - \sum_{i=1}^{k} L^i_{(t)} - S_{(t-1)} \right\|_F^2, \tag{9}$$

$$E^{k+1}_{(t)} = \left\| X - \sum_{i=1}^{k} L^i_{(t)} - S_{(t)} \right\|_F^2, \tag{10}$$

The global optimality of $L^i_{(t)}$ yields $E^1_{(t)} \geq E^2_{(t)} \geq \cdots \geq E^k_{(t)}$. The global optimality of $S_{(t)}$ yields $E^k_{(t)} \geq E^{k+1}_{(t)}$. In addition, we have

$$E^{k+1}_{(t)} = \left\| X - \sum_{i=2}^{k} L^i_{(t)} - S_{(t)} - L^1_{(t)} \right\|_F^2, \tag{11}$$

$$E^1_{(t+1)} = \left\| X - \sum_{i=2}^{k} L^i_{(t)} - S_{(t)} - L^1_{(t+1)} \right\|_F^2. \tag{12}$$

The global optimality of $L^1_{(t+1)}$ yields $E^{k+1}_{(t)} \geq E^1_{(t+1)}$. Therefore, the objective value (or the decomposition error) $\|X - \sum_{i=1}^{k} L^i - S\|_F^2$ keeps decreasing throughout the iteration rounds of (5), i.e.,

$$E^1_{(1)} \geq E^{k+1}_{(1)} \geq \cdots \geq E^1_{(t)} \geq E^{k+1}_{(t)} \geq \cdots \tag{13}$$

Since the objective value of (3) is monotonically decreasing and the constraints are satisfied all the time, iteratively solving (4) produces a sequence of objective values that converge to a local minimum. This completes the proof. □

After obtaining the decomposition by solving (3), each training sample is represented by the sum of several components in L^i characterized by the labels it belongs to and the residual in S. Therefore, the mapping of label i in feature subspace is defined as the row space $C^i \in \mathbb{R}^{r^i \times p}$ of the matrix $L^i_{\Omega_i}$, which can be obtained via the QR decomposition of $\left(L^i_{\Omega_i}\right)^T$.

3.2 Fast MSE training via bilateral random projections

The main computation in (5) is the k times of SVD in obtaining $L^i_{\Omega_i} (i = 1, \ldots, k)$. SVD requires $\min\left(mn^2, m^2 n\right)$ flops for an $m \times n$ matrix, and thus it is impractical when X is of large size. Random projection is effective in accelerating the matrix multiplication and decomposition [4]. In this paper, we introduce "bilateral random projections (BRP)", which is a direct extension of random projection, to accelerate the optimization of $L^i_{\Omega_i} (i = 1, \ldots, k)$.

For clear representation, we use letters independent to the ones we use in other parts of this paper to illustrate BRP. In particular, given r bilateral random projections (BRP) of an $m \times n$ dense matrix X (w.l.o.g, $m \geq n$), i.e., $Y_1 = XA_1$ and $Y_2 = X^T A_2$, wherein $A_1 \in \mathbb{R}^{n \times r}$ and $A_2 \in \mathbb{R}^{m \times r}$ are random matrices,

$$L = Y_1 \left(A_2^T Y_1\right)^{-1} Y_2^T \tag{14}$$

is a fast rank-r approximation of X. The computation of L includes an inverse of an $r \times r$ matrix and three matrix multiplications. Thus, for a dense X, $2mnr$ floating-point operations (flops) are required to obtain BRP, $r^2(2n + r) + mnr$ flops are required to obtain L. The computational cost is much less than that of the SVD based approximation, while its approximation error approaches to that of SVD based method.

We build the random matrices A_1 and A_2 in an adaptive way. Initially, both A_1 and A_2 are set to standard Gaussian matrices whose entries are independent variables following the standard normal distribution. We firstly compute $Y_1 = XA_1$, update $A_2 := Y_1$ and calculate the left random projection as $Y_2 = X^T A_2$ by using the new A_2, and then we update $A_1 := Y_2$ and calculate the right random projection $Y_1 = XA_1$ by using the new A_1. This adaptive updating of random matrices requires additional flops of mnr.

We analyze the error bounds of the BRP based low-rank approximation (14).

The SVD of an $m \times n$ (w.l.o.g, $m \geq n$) matrix X takes the form

$$X = U\Lambda V^T = U_1 \Lambda_1 V_1^T + U_2 \Lambda_2 V_2^T, \tag{15}$$

where Λ_1 is an $r \times r$ diagonal matrix which diagonal elements are the first largest r singular values, U_1 and V_1 are the corresponding singular vectors, Λ_2, U_2 and V_2 forms the rest part of SVD. Assume that r is the target rank, A_1 and A_2 have $r + p$ columns for oversampling. We consider the spectral norm of the approximation error E for (14)

$$\begin{aligned} \|X - L\| &= \left\| X - Y_1 \left(A_2^T Y_1\right)^{-1} Y_2^T \right\| \\ &= \left\| \left[I - XA_1 \left(A_2^T XA_1\right)^{-1} A_2^T \right] X \right\|. \end{aligned} \tag{16}$$

The unitary invariance of the spectral norm leads to

$$\begin{aligned}\|X-L\| &= \left\|U^T\left[I - XA_1\left(A_2^TXA_1\right)^{-1}A_2^T\right]X\right\| \\ &= \left\|\Lambda\left[I - V^TA_1\left(A_2^TXA_1\right)^{-1}A_2^TU\Lambda\right]\right\|.\end{aligned} \tag{17}$$

In low-rank approximation, the left random projection matrix A_2 is build from the left random projection $Y_1 = XA_1$, and then the right random projection matrix A_1 is build from the left random projection $Y_2 = X^TA_2$. Thus $A_2 = Y_1 = XA_1 = U\Lambda V^TA_1$ and $A_1 = Y_2 = X^TA_2 = X^TXA_1 = V\Lambda^2V^TA_1$. Hence the approximation error given in (17) has the following form

$$\left\|\Lambda\left[I - \Lambda^2V^TA_1\left(A_1^TV\Lambda^4V^TA_1\right)^{-1}A_1^TV\Lambda^2\right]\right\|. \tag{18}$$

The following theorem gives the bound for the spectral norm of the deterministic error $\|X-L\|$.

Theorem 2. (Deterministic error bound) *Given an $m\times n$ $(m\geq n)$ real matrix X with singular value decomposition $X = U\Lambda V^T = U_1\Lambda_1V_1^T + U_2\Lambda_2V_2^T$, and chosen a target rank $r\leq n-1$ and an $n\times(r+p)$ $(p\geq 2)$ standard Gaussian matrix A_1, the BRP based low-rank approximation (14) approximates X with the error*

$$\|X-L\|^2 \leq \left\|\Lambda_2^2\left(V_2^TA_1\right)\left(V_1^TA_1\right)^\dagger\Lambda_1^{-1}\right\|^2 + \|\Lambda_2\|^2.$$

If the singular values of X decay fast, the first term in the deterministic error bound will be very small. The last term is the rank-r SVD approximation error. Therefore, the BRP based low-rank approximation (14) is nearly optimal.

The average error bound of BRP based low-rank approximation is obtained by analyzing the statistical properties of the random matrices that appear in the deterministic error bound in Theorem 2.

Theorem 3. (Average error bound) *Frame the hypotheses of Theorem 2, we have*

$$\begin{aligned}\mathbb{E}\|X-L\| \leq &\left(\sqrt{\frac{1}{p-1}\sum_{i=1}^{r}\frac{\lambda_{r+1}^2}{\lambda_i^2}} + 1\right)|\lambda_{r+1}| \\ &+ \frac{\mathrm{e}\sqrt{r+p}}{p}\sqrt{\sum_{i=r+1}^{n}\frac{\lambda_i^2}{\lambda_r^2}}.\end{aligned}$$

The average error bound will approach to the SVD approximation error $|\lambda_{r+1}|$ if $|\lambda_{r+1}| \ll |\lambda_{i:i=1,\cdots,r}|$ and $|\lambda_r| \gg |\lambda_{i:i=r+1,\cdots,n}|$.

The deviation bound for the spectral norm of the approximation error can be obtained by analyzing the deviation bound of $\left\|\Lambda_2^2\left(V_2^TA_1\right)\left(V_1^TA_1\right)^\dagger\Lambda_1^{-1}\right\|$ in the deterministic error bound and by applying the concentration inequality for Lipschitz functions of a Gaussian matrix.

Theorem 4. (**Deviation bound**) *Frame the hypotheses of Theorem 2. Assume that $p\geq 4$. For all $u,t\geq 1$, we have*

$$\begin{aligned}\|X-L\| \leq &\left(1 + t\sqrt{\frac{12r}{p}}\left(\sum_{i=1}^{r}\lambda_i^{-1}\right)^{\frac{1}{2}} + \frac{\mathrm{e}\sqrt{r+p}}{p+1}\cdot\right. \\ &\left. tu\lambda_r^{-1}\right)\lambda_{r+1}^2 + \frac{\mathrm{e}\sqrt{r+p}}{p+1}\cdot t\lambda_r^{-1}\left(\sum_{i=r+1}^{n}\lambda_i^2\right)^{\frac{1}{2}}.\end{aligned}$$

except with probability $\mathrm{e}^{-u^2/2} + 4t^{-p} + t^{-(p+1)}$.

See supplemental material for the proof of the above theorems.

Algorithm 1 summarizes the training stage of MSE with BRP based acceleration.

Algorithm 1: MSE Training

Input: X, Ω_i, $r^i, i=1,\ldots,k$, K, ϵ
Output: $C^i, i=1,\ldots,k$
Initialize L^i and S according to (6), $t:=0$;
while $\left\|X-\sum_{j=1}^{k}L^j-S\right\|_F^2 > \epsilon$ **do**
  $t:=t+1$;
  for $i\leftarrow 1$ **to** k **do**
    $N := \left(X-\sum_{j=1,j\neq i}^{k}L^j-S\right)_{\Omega_i}$;
    Generate standard Gaussian matrix $A_1\in\mathbb{R}^{p\times r^i}$;
    $Y_1 := NA_1$, $A_2 := Y_1$;
    $Y_2 := N^TY_1$, $Y_1 := NY_2$;
    $L_{\Omega_i}^i := Y_1\left(A_2^TY_1\right)^{-1}Y_2^T$, $L_{\bar{\Omega}_i}^i := \mathbf{0}$;
  end
  $N := \left|X-\sum_{j=1}^{k}L^j\right|$;
  $S := \mathcal{P}_\Phi(N)$, Φ is the index set of the first K largest entries of $|N|$;
end
QR decomposition $\left(L_{\Omega_i}^i\right)^T = Q^iR^i$ for $i=1,\ldots,k$,
$C^i := \left(Q^i\right)^T$;

4 MSE prediction: group sparsity

In this section, we introduce the prediction stage of MSE by estimating the group sparse representation of a given sample on the obtained subspace ensemble C. Note that in the training stage, we decompose the training data into the sum of low-rank components $L_{\Omega_i}^i$ characterized by the labels and a sparse residual S. The mapping of label i in the feature space is defined as the row space C^i of $L_{\Omega_i}^i$, and the components of the training samples characterized by label i lies in the linear subspace C^i.

In the prediction stage of MSE, we use group *lasso* [17] to estimate the group sparse representation $\beta\in\mathbb{R}^{\sum r^i}$

of a test sample $x \in \mathbb{R}^p$ on the subspace ensemble $C = [C^1; \ldots; C^k]$, wherein the k groups are defined as index sets of the coefficients corresponding to $C^1, \ldots, C^k$. Since group *lasso* selects nonzero coefficients group-wisely, nonzero coefficients in the group sparse representation will concentrate on the groups corresponding to the labels that the sample belongs to.

According to the above analysis, we solve the following group *lasso* problem in the prediction stage of MSE

$$\min_{\beta} \frac{1}{2} \|x - \beta C\|_F^2 + \lambda \sum_{i=1}^{k} \|\beta_{G_i}\|_2 , \tag{19}$$

where the index set G_i includes all the integers between $1 + \sum_{j=1}^{i-1} r^j$ and $\sum_{j=1}^{i} r^j$ (including these two).

To obtain the final prediction of the label vector $y \in \{0,1\}^k$ for a test sample x, we use a simple thresholding of the magnitude sum of coefficients in each group to test which groups that the sparse coefficients in β concentrate on

$$y_\Psi = \mathbf{1}, y_{\overline{\Psi}} = \mathbf{0}, \Psi = \{i : \|\beta_{G_i}\|_1 \geq \delta\} . \tag{20}$$

Although y can also be obtained via selecting the groups with nonzero coefficients when λ in (19) is chosen properly, we set the threshold δ as a small positive value to guarantee the robustness to λ.

Algorithm 2 summarizes the prediction stage of MSE.

Algorithm 2: MSE Prediction

Input: x, $C^i, i = 1, \ldots, k$, λ, δ
Output: y
Solve group *lasso* in (19) by using an existing group *lasso* algorithm [8];
Predict y via thresholding in (20);

5 Experiments

In this section, we evaluate MSE on several benchmark datasets of text classification, image annotation, scene classification, music categorization, genomics and web page classification. We compare MSE with BR [11], ML-KNN [18] and MDDM [21] on four evaluation metrics for evaluating the effectiveness, as well as the CPU seconds for evaluating the efficiency. All the experiment are run in MatLab on a server with dual quad-core 3.33 GHz Intel Xeon processors and 32 GB RAM. In the experiments of multi-label prediction, four metrics, which are precision, recall, F1 score and accuracy, are used to measure the prediction performance. The detailed definitions of these metrics are given in Section 7.1.1 of [13].

5.1 Evaluation metrics

In the experiments of multi-label prediction, four metrics, which are precision, recall, F1 score and accuracy, are used to measure the prediction performance.

Given two label matrices $Y1, Y2 \in \{0,1\}^{n \times k}$, wherein $Y1$ is the real one and $Y2$ is the prediction one, precision, recall, F1 score and accuracy and are defined as:

$$Prec = \frac{1}{n} \sum_{i=1}^{n} \frac{\text{card}\,(Y1_i \cap Y2_i)}{\text{card}\,(Y2_i)}, \tag{21}$$

$$Rec = \frac{1}{n} \sum_{i=1}^{n} \frac{\text{card}\,(Y1_i \cap Y2_i)}{\text{card}\,(Y1_i)}, \tag{22}$$

$$F1 = \frac{1}{n} \sum_{i=1}^{n} \frac{2\text{card}\,(Y1_i \cap Y2_i)}{\text{card}\,(Y1_i) + \text{card}\,(Y2_i)}, \tag{23}$$

$$Acc = \frac{1}{n} \sum_{i=1}^{n} \frac{\text{card}\,(Y1_i \cap Y2_i)}{\text{card}\,(Y1_i \cup Y2_i)}. \tag{24}$$

These four metrics have been broadly applied on general binary data. However, their importances are different to each other in evaluating the performance of multi-label prediction, because there are much more 1s than 0s in the label matrix. Precision and recall should be considered together, because high precision always accompanies low recall when most positive samples are falsely predicted as positive. F1-score and accuracy are less sensitive to the imbalance of label matrix. Therefore, a fair evaluation of prediction performance should include integrative consideration of all the four metrics, whose importances can be roughly given by $F1, Acc > \{Prec, Rec\}$.

5.2 Datasets

We evaluate the prediction performance and time cost of MSE on 13 datasets from different domains and of different scales, including Corel5k (image), Scene (image), Mediamill (video), Enron (text), Genbase (genomics), Medical (text), Emotions (music), Slashdot (text) and 5 sub datasets selected in Yahoo dataset (web data). These datasets were obtained from Mulan's website [1] and MEKA's website [2]. They were collected from different practical problems. Table 1 shows the number of samples n (training samples+test samples), number of features p, number of labels k, and the average cardinality of all label vectors $Card$ of different datasets.

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

Table 1: Information of datasets that are used in experiments of MSE. In the table, n (training samples+test samples) is the number of samples, p is the number of features, k is the number of labels, "Card" is the average cardinality of all label vectors.

Datasets	n	p	k	Card
Corel5k	$4500 + 500$	499	374	3.522
Mediamill	$30993 + 12914$	120	101	4.376
Enron	$1123 + 579$	1001	53	3.378
Genbase	$463 + 199$	1186	27	1.252
Medical	$333 + 645$	1449	45	1.245
Emotions	$391 + 202$	72	6	1.869
Scene	$1211 + 1196$	294	6	1.074
Slashdot	$2338 + 1444$	1079	22	1.181
Arts	$2000 + 3000$	462	26	1.636
Business	$2000 + 3000$	438	30	1.587
Education	$2000 + 3000$	550	33	1.461
Recreation	$2000 + 3000$	606	22	1.423
Science	$2000 + 3000$	743	40	1.451

5.3 Performance comparison

Table 2: Prediction performances (%) and CPU seconds of BR [11], ML-KNN [18], MDDM [21] and MSE on Yahoo. Prec-precision, Rec-recall, F1-F1 score, Acc-accuracy

	Methods	Prec	Rec	F1	Acc	CPU sec.
Arts	BR	76	25	26	24	46.8
	ML-knn	62	7	25	6	77.6
	MDDM	68	6	21	5	37.4
	MSE	35	40	31	28	11.7
Education	BR	69	27	28	26	50.1
	ML-knn	58	6	31	5	99.8
	MDDM	59	5	26	5	45.2
	MSE	41	35	32	29	12.6
Recreation	BR	84	23	23	22	53.2
	ML-knn	70	9	23	8	112
	MDDM	66	7	18	6	41.9
	MSE	41	49	36	30	19.1
Science	BR	79	19	19	19	84.9
	ML-knn	59	4	20	4	139
	MDDM	66	4	19	4	53.0
	MSE	31	39	29	26	20.1
Business	BR	87	74	76	71	28.9
	ML-knn	68	9	70	8	93.2
	MDDM	66	7	69	7	42.7
	MSE	84	82	78	78	13.5

We show the prediction performance and time cost in CPU seconds of BR, ML-KNN, MDDM and MSE in Table 3 and Table 2. In BR, we use the MatLab interface of LIBSVM 3.0 [3] to train the classic linear SVM classifiers for each label. The parameter $C \in \{10^{-3}, 10^{-2}, 0.1, 1, 10, 10^2, 10^3\}$ with the best performance on the training set was used. In ML-KNN, the number of neighbors was 30 for all the datasets.

Table 3: Prediction performances (%) and CPU seconds of BR [11], ML-KNN [18], MDDM [21] and MSE on 8 datasets. Prec-precision, Rec-recall, F1-F1 score, Acc-accuracy

	Methods	Prec	Rec	F1	Acc	CPU sec.
Mediamill	BR	69	35	43	33	120141
	ML-knn	41	6	54	5	5713
	MDDM	36	5	53	4	48237
	MSE	58	78	53	37	1155
Enron	BR	51	28	35	24	77.1
	ML-knn	51	7	46	5	527
	MDDM	50	8	49	7	29
	MSE	44	50	40	28	271
Medical	BR	2	26	5	2	4.88
	ML-knn	75	7	48	6	22.8
	MDDM	74	3	30	2	32.3
	MSE	36	90	45	26	7.5
Slashdot	BR	11	22	14	10	140
	ML-knn	71	10	31	8	708
	MDDM	39	1	4	1	114
	MSE	38	61	37	27	175
Scene	BR	55	67	66	63	4.19
	ML-knn	78	62	69	54	14.3
	MDDM	75	64	69	53	7.59
	MSE	61	85	70	68	3.62
Emotions	BR	55	53	51	42	0.68
	ML-knn	68	28	41	22	0.66
	MDDM	54	28	41	22	0.66
	MSE	40	100	52	37	0.01
Genbase	BR	5	39	9	5	1.99
	ML-knn	100	50	92	50	9.38
	MDDM	98	51	92	51	6.09
	MSE	83	96	86	70	8.62
Corel5k	BR	2	20	4	2	2240
	ML-knn	62	1	3	0.9	2106
	MDDM	62	1	7	1	458
	MSE	9	11	8	5	1054

In MDDM, the regularization parameter for uncorrelated subspace dimensionality reduction was selected as 0.12 and the dimension of the subspace was set as 20% of the dimension of the original data. In MSE, we selected r^i as an integer in $[1, 6]$, $K \in \left[10^{-6}, 10^{-3}\right]$, $\lambda \in [0.2, 0.45]$ and $\delta \in \left[10^{-4}, 10^{-2}\right]$. We roughly selected 4 groups of parameters in the ranges for each dataset and chose the one with

[3] http://www.csie.ntu.edu.tw/čjlin/libsvm/

the best performance on the training data. Group *lasso* in MSE is solved by SLEP [8] in our experiments.

The experimental results show that MSE is competitive on both speed and prediction performance, because it explores label correlations and structure without increasing the problem size. In addition, the bilateral random projections further accelerate the computation. In particular, its training time increases much more slowly than other methods, so it is more efficient when applied to large scale datasets such as Mediamill, Arts and Education. MDDM is faster than MSE on a few datasets because MDDM invokes ML-knn on the data after dimension reduction, while MSE is directly applicable to the original high dimensional data.

In the comparison of performance via the four metrics, the F1 score and accuracy of MSE outperform those of other methods on most datasets. Moreover, MSE has smaller gaps between precision and recall on different tasks than other methods, and this implies it is robust to the imbalance between positive and negative samples. Note in multi-label prediction, only large values of all four metrics are sufficient to indicate the success of the prediction, while the combination of some large valued metrics and some small valued ones are always caused by the imbalance of the samples. Therefore, MSE provides better prediction performance than other methods on most datasets.

6 Conclusion

6.1 Discussion

In the training stage of MSE, the data matrix is decomposed as the sum of several low-rank parts and a sparse residual via a randomized alternating minimization algorithm. This can be viewed as an extension of GoDec algorithm [22], which aims at decomposing a data matrix as the sum of a low-rank part and a sparse part in the noisy case. A similar alternating minimization algorithm accelerated by random projection is invoked in GoDec. The essential difference between the training stage and GoDec lies on the problem formulation and decomposition model. In particular, the training algorithm of MSE is specifically designed for solving multi-label learning problem, while GoDec is developed for low-rank and sparse matrix decomposition in visual analysis and matrix completion. MSE seeks for k low-rank parts, each of which corresponds to a label, while GoDec seeks for one low-rank part. Moreover, the theoretical analysis of MSE includes the error bounds of the bilateral random projection based approximation, while the analysis in GoDec concentrates on the linear convergence of the low-rank part and the sparse part.

In the prediction stage of MSE, the group sparse representation of a multi-label instance on the subspace ensemble is obtained, and then the prediction can be derived from the nonzero groups of coefficients in the group sparse representation. This can be seem as a non-trivial extension for multi-label learning of sparse representation [16] for multi-class classification , where the sparse representation of an instance on the training samples indicates the class that the instance belongs to. They both verifies the success of sparse representation in discriminative learning. However, the prediction of MSE is different from the method of sparse representation because 1) MSE explores the group sparse representation and 2) the bases for the group sparse representation is composed of subspace ensemble rather than the training samples. These differences guarantee that MSE can be applied to multi-label learning problems, and its prediction time cost will not substantially increase with the increasing of the training samples.

6.2 Conclusion

In this paper, we develop a novel multi-label learning method "multi-label subspace ensemble (MSE)" by considering the mapping of each label in the feature space as a subspace and formulating the prediction as the estimation of sparse representation on the obtained subspace ensemble. Its training stage decomposes the training data as the sum of k low-rank components $L^i_{\Omega_i}$ explained by the k labels and a sparse residual S that cannot be explained by the given labels. This structured decomposition is accomplished by the bilateral random projections based alternating minimization with low time cost, and it converges to a local minimum. The row space C^i of $L^i_{\Omega_i}$ defines the mapping of label i in the feature space. The prediction stage estimates the group sparse representation of a new sample on the subspace ensemble composed of $C^i (i = 1, \cdots, k)$ via group *lasso*. MSE predicts the labels by selecting the groups that the nonzero representation coefficients concentrate on.

The main significance of MSE is that it eliminates the rapid growth of model complexity caused by the increase in the number of labels, which is a common problem of existing multi-label approaches. Meanwhile the label correlations are well preserved in the subspace ensemble via structured decomposition and fully explored in the prediction by group sparsity. Another compelling advantage of MSE is that the search of correct labels in prediction is conducted in the subspace ensemble rather in the ambient label space. Hence MSE needs not to estimate the structure of the label space. Furthermore, the subspaces obtained by MSE provide explicit interpretations of the labels in the feature space.

Acknowledgements

The authors would like to thank the anonymous reviewers who have provided constructive comments on improving this paper. This work is supported by the Australian ARC discovery project (ARC DP-120103730).

References

[1] N. C. Bianchi, C. Gentile, and L. Zaniboni. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, 7:31–54, 2006.

[2] L. Breiman and J. H. Friedman. Predicting multivariate responses in multiple linear regression (with discussion). *The Journal of the Royal Statistical Society Series B*, 54:5–54, 1997.

[3] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. In *SIGKDD*, 2010.

[4] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. *arXiv: 0909.4061*, 2009.

[5] D. Hsu, S. M. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, 2009.

[6] S. Ji, L. Tang, S. Yu, and J. Ye. A shared-subspace learning framework for multi-label classification. *ACM Trans on Knowledge Discovery from Data*, 2(1), 2010.

[7] S. Ji and J. Ye. Linear dimensionality reduction for multi-label classification. In *IJCAI*, 2009.

[8] J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009.

[9] J. Petterson and T. Caetano. Reverse multi-label learning. In *NIPS*, 2010.

[10] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, 2009.

[11] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[12] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *ECML/PKDD Workshop on Mining Multidimensional Data*, 2008.

[13] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook*, 2010.

[14] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *ECML*, pages 406–417, 2007.

[15] G. Tsoumakas, M.-L. Zhang, and Z.-H. Zhou. Learning from multi-label data. In *ECML/PKDD*, 2009.

[16] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[17] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[18] M. L. Zhang and Z. H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[19] X. Zhang, T. Graepel, and R. Herbrich. Bayesian online learning for multi-label and multi-variate performance measures. In *AISTATS*, 2010.

[20] Y. Zhang and J. Schneider. Multi-label output codes using canonical correlation analysis. In *AISTATS*, 2011.

[21] Y. Zhang and Z. H. Zhou. Multi-label dimensionality reduction via dependence maximization. In *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*, pages 1503–1505, 2008.

[22] T. Zhou and D. Tao. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *ICML*, 2011.