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 $\mathcal{Y}$ often scatter 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-
dicted labels by using a binary classifier.

However, the size of the search space in these approaches is much larger than \( O(k) \), 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 can then 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 a 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 result correlations are naturally preserved in the subspace ensemble. Given a new instance, its labels are predicted by estimating its sparse representation in the subspace ensemble, wherein the non-zero 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 \( X \), with \( X \) 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 \( l \) and a sparse residual \( s \)

\[
x = \sum_{i:y_i=1} l_i + s.
\]

(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 \( \beta_{G_i} \) is the representation coefficients corresponding to \( C_i \). Thus the model (1) can be equivalently written as

\[
x = \sum_{i:y_i=1} \beta_{G_i} C_i + s, \quad \forall i \in \{ i:y_i = 0 \}, \beta_{G_i} = \mathbf{0}.
\]

(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 \( \beta_{G_i} \), corresponding to the labels that \( x \) does not belong to are zeros, so \( \beta \) 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 $L_{\Omega_i}^i$, wherein $\Omega_i$ is the index set of training samples with label $i$. Thus the row space of $L_{\Omega_i}^i$ 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 $\beta$ on the subspace ensemble $C$, and then a simple thresholding is used to test which groups that $\beta$ 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 $\Omega_i$ and $\Omega_j$ will have many shared elements. Thus the row spaces of $L_{\Omega_i}^i$ and $L_{\Omega_j}^j$, 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 $X = \sum_{i=1}^{k} L_i^i + S$. For the matrix $L_i^i$, 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 $\Omega_i$ to denote the index set of samples with label $i$ in the matrix $X$ and $L_i^i$, and then the matrix composed of the nonzero rows in $L_i^i$ is represented by $L_{\Omega_i}^i$. In the decomposition, the rank of $L_{\Omega_i}^i$ 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 $L_{\Omega_i}^i$ 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:

$$\min_{L_i^i, S} \left\| X - \sum_{i=1}^{k} L_i^i - S \right\|_F^2, \quad \text{s.t.} \quad \text{rank} (L_{\Omega_i}^i) \leq r^i, L_{\Omega_i}^i \cdot 0 = 0, \forall i = 1, \ldots, k$$

(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 $L_{\Omega_i}^i$.

3.1 Alternating minimization

Although the rank constraint to $L_{\Omega_i}^i$ 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:

$$L_{\Omega_i}^i = \arg \min_{\text{rank}(L_{\Omega_i}^i) \leq r^i} \left\| X - \sum_{j=1, j \neq i}^{k} L_j^j - S \right\|_F^2, \quad \forall i = 1, \ldots, k,$$

$$S = \arg \min_{\text{card}(S) \leq K} \left\| X - \sum_{j=1}^{k} L_j^j - S \right\|_F^2.$$

(4)

The solutions of $L_{\Omega_i}^i$ 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, $L_{\Omega_i}^i$ is built from the first $r^i$ largest singular values and the corresponding singular vectors of $X - \sum_{j=1, j \neq i}^{k} L_j^j - S$, while $S$ is built from the $K$ entries with the largest absolute value in $X - \sum_{j=1}^{k} L_j^j$, i.e.,

$$L_{\Omega_i}^i = \sum_{q=1}^{r^i} \lambda_q U_q V_q^T, i = 1, \ldots, k,$$

$$\text{svd} \left( X - \sum_{j=1, j \neq i}^{k} L_j^j - S \right)_{\Omega_i} = U \Lambda V^T;$$

$$S = \mathcal{P}_\Phi \left( X - \sum_{j=1}^{k} L_j^j \right) \Phi : \left( X - \sum_{j=1}^{k} L_j^j \right)_{r,s \in \Phi} \neq 0$$

and

$$\left( X - \sum_{j=1}^{k} L_j^j \right)_{r,s \in \Phi^c} \leq K.$$  

(5)

The projection $S = \mathcal{P}_\Phi (R)$ represents that the matrix $S$ has the same entries as $R$ on the index set $\Phi$, 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 $L_{\Omega_i}^i$ and $S$ as

$$L_{\Omega_i}^i := Z_{\Omega_i}, i = 1, \ldots, k,$$

$$Z = D^{-1} X, D = \text{diag} (Y 1);$$

(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 $\| X - \sum_{i=1}^{k} L_i^i - S \|_F^2$ that converges to a local minimum.
Proof. Let the objective value (decomposition error) \( \| X - \sum_{i=1}^{k+1} L_i^i - S \|_F^2 \) 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

\[
E_{1}^{(t)} = \left\| X - S_{(t-1)} - L_{1}^{(t)} - \sum_{i=3}^{k} L_{i}^{(t-1)} - L_{2}^{(t-1)} \right\|_F^2, \\
E_{2}^{(t)} = \left\| X - S_{(t-1)} - L_{1}^{(t)} - \sum_{i=3}^{k} L_{i}^{(t-1)} - L_{2}^{(t)} \right\|_F^2, \\
\vdots \\
E_{k}^{(t)} = \left\| X - \sum_{i=1}^{k} L_{i}^{(t)} - S_{(t-1)} \right\|_F^2, \\
E_{k+1}^{(t)} = \left\| X - \sum_{i=1}^{k} L_{i}^{(t)} - S_{(t)} \right\|_F^2,
\]

(7) \( \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quad
The unitary invariance of the spectral norm leads to
\[
\|X - L\| = \left\| U^T \left[ I - X A_1 (A_2^T X A_1)^{-1} A_2^T \right] X \right\|
\]
\[
= \left\| \Lambda \left[ I - V^T A_1 (A_2^T X A_1)^{-1} A_2^T U \Lambda \right] \right\|.
\]
(17)

In low-rank approximation, the left random projection matrix \(A_2\) is built from the left random projection \(Y_1 = X A_1\), and then the right random projection matrix \(A_1\) is built from the left random projection \(Y_2 = X^T A_2\). Thus \(A_2 = Y_1 = X A_1 = U A V^T A_1\) and \(A_1 = Y_2 = X^T A_2 = X^T X A_1 = V A^2 V^T A_1\). Hence the approximation error given in (17) has the following form
\[
\left\| \Lambda \left[ I - \Lambda^2 V^T A_1 (A_2^T V A^2 V^T A_1)^{-1} A_2^T V \Lambda^2 \right] \right\|.
\]
(18)

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

**Theorem 2. (Deterministic error bound)** Given an \(n \times n\) \((m \geq n)\) real matrix \(X\) with singular value decomposition \(X = U A V^T\), 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
\[
\|X - L\|^2 \leq \left\| \Lambda^2 (V_2^T A_1) (V_1^T A_1)^{-1} \Lambda^{-1} \right\|^2 + \| A_2 \|^2.
\]

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
\[
\mathbb{E}[\|X - L\|] \leq \left( \frac{1}{p - 1} \sum_{i=1}^{r} \frac{\lambda_i^2}{\lambda_i^2 + 1} \right)^\frac{1}{2} |\lambda_{r+1}|^2
\]
\[
+ \frac{e^{\sqrt{r + p}}}{p} \sqrt{\frac{1}{\sum_{i=r+1}^{n} \lambda_i^2}}.
\]

The average error bound will approach to the SVD approximation error \(|\lambda_{r+1}|\) if \(|\lambda_{r+1}| \ll |\lambda_{i=1, \ldots, r}|\) and \(|\lambda_i| \gg |\lambda_{i=r+1, \ldots, n}|\).

The deviation bound for the spectral norm of the approximation error can be obtained by applying the deviation bound of \(\| A^2 (V_2^T A_1) (V_1^T A_1)^{-1} \Lambda^{-1} \|\) 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
\[
\|X - L\| \leq \left( 1 + t \sqrt{\frac{12r}{p}} \sum_{i=1}^{r} \frac{\lambda_i^{-1}}{2} \right) + \frac{e^{\sqrt{r + p}}}{p} \left( \sum_{i=r+1}^{n} \lambda_i^2 \right)^\frac{1}{2}.
\]

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

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:** \(X, \Omega_i, r^i, i = 1, \ldots, k, K, \epsilon\)

**Output:** \(C^i, i = 1, \ldots, k\)

Initialize \(L^i\) and \(S\) according to (6), \(t := 0;\)

while \(\|X - \sum_{j=1}^{k} L^j - S\|^2_F > \epsilon\) do

\[ t := t + 1; \]

for \(i \leftarrow 1 \text{ to } k\) do

\[ N := \left( X - \sum_{j=1, j \neq i}^{k} L^j - S \right) \Omega_i; \]

Generate standard Gaussian matrix \(A_1 \in \mathbb{R}^{p \times r^i};\)

\(Y_1 := N A_1, A_2 := Y_1;\)

\(Y_2 := N^T Y_1, Y_1 := N Y_2;\)

\(L^i_{\Omega_i} := Y_1 (A_2^T Y_1)^{-1} Y_2^T, L^i_{\Omega_i^\perp} := 0;\)

end

\[ N := \left| X - \sum_{j=1}^{k} L^j \right|; \]

\(S := \mathcal{P}_k (N), \Phi\) is the index set of the first \(K\) largest entries of \(|N|\); end

QR decomposition \((L^i_{\Omega_i})^T = Q^i R^i\) for \(i = 1, \ldots, k, C^i := (Q^i)^T;\)


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 \(L^i_{\Omega_i}\) 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 \(L^i_{\Omega_i}\), 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 \(\beta \in \mathbb{R}^r\).
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 \( \beta \) concentrate on

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

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

Algorithm 2 summarizes the prediction stage of MSE.

**Algorithm 2: MSE Prediction**

**Input:** \( x, C^i, i = 1, \ldots, k, \lambda, \delta \)

**Output:** \( y \)

Solve group lasso in (19) by using an existing group lasso algorithm [8];

Predict \( y \) via thresholding in (20);

\[\text{card} \quad \text{card} \quad \text{card} \quad \text{card} \]

\[
\sum_{i=1}^{n} \frac{\text{card}(Y_{1i} \cap Y_{2i})}{\text{card}(Y_{2i})}, \tag{21}
\]

\[
\sum_{i=1}^{n} \frac{\text{card}(Y_{1i} \cap Y_{2i})}{\text{card}(Y_{1i})}, \tag{22}
\]

\[
\sum_{i=1}^{n} \frac{2\text{card}(Y_{1i} \cap Y_{2i})}{\text{card}(Y_{1i}) + \text{card}(Y_{2i})}, \tag{23}
\]

\[
\sum_{i=1}^{n} \frac{\text{card}(Y_{1i} \cup Y_{2i})}{\text{card}(Y_{1i}) + \text{card}(Y_{2i})}, \tag{24}
\]

These four metrics have been broadly applied on general binary data. However, their importance is 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 importance can be roughly given by \( F1, Acc > \{ Prec, Rec \} \).

\[\text{Card} \quad \text{Card} \quad \text{Card} \quad \text{Card} \]

\[\text{http://mulan.sourceforge.net/datasets.html} \quad \text{http://mekao.sourceforge.net/}\]

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 \( Y_1, Y_2 \in \{0, 1\}^{n \times k} \), wherein \( Y_1 \) is the real one and \( Y_2 \) is the prediction one, precision, recall, F1 score and accuracy and are defined as:

\[
\text{Prec} = \frac{\sum_{i=1}^{n} \text{card}(Y_{1i} \cap Y_{2i})}{\text{card}(Y_{2i})}, \tag{21}
\]

\[
\text{Rec} = \frac{\sum_{i=1}^{n} \text{card}(Y_{1i} \cap Y_{2i})}{\text{card}(Y_{1i})}, \tag{22}
\]

\[
F1 = \frac{2 \sum_{i=1}^{n} \text{card}(Y_{1i} \cap Y_{2i})}{\sum_{i=1}^{n} \text{card}(Y_{1i}) + \sum_{i=1}^{n} \text{card}(Y_{2i})}, \tag{23}
\]

\[
\text{Acc} = \frac{\sum_{i=1}^{n} \text{card}(Y_{1i} \cup Y_{2i})}{\sum_{i=1}^{n} \text{card}(Y_{1i}) + \sum_{i=1}^{n} \text{card}(Y_{2i})}, \tag{24}
\]

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 Mula’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 \( \text{Card} \) of different datasets.

---

\(^1\)http://mulan.sourceforge.net/datasets.html

\(^2\)http://mekao.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.

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

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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Prec</th>
<th>Rec</th>
<th>F1</th>
<th>Acc</th>
<th>CPU sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR</td>
<td>76</td>
<td>25</td>
<td>26</td>
<td>24</td>
<td>46.8</td>
</tr>
<tr>
<td>ML-knn</td>
<td>62</td>
<td>7</td>
<td>25</td>
<td>6</td>
<td>77.6</td>
</tr>
<tr>
<td>MDDM</td>
<td>68</td>
<td>6</td>
<td>21</td>
<td>5</td>
<td>37.4</td>
</tr>
<tr>
<td>MSE</td>
<td>35</td>
<td>40</td>
<td>31</td>
<td>26</td>
<td>11.7</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Prec</th>
<th>Rec</th>
<th>F1</th>
<th>Acc</th>
<th>CPU sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR</td>
<td>69</td>
<td>35</td>
<td>43</td>
<td>33</td>
<td>120141</td>
</tr>
<tr>
<td>ML-knn</td>
<td>41</td>
<td>6</td>
<td>54</td>
<td>5</td>
<td>5713</td>
</tr>
<tr>
<td>MDDM</td>
<td>36</td>
<td>5</td>
<td>53</td>
<td>4</td>
<td>48237</td>
</tr>
<tr>
<td>MSE</td>
<td>58</td>
<td>78</td>
<td>53</td>
<td>37</td>
<td>1155</td>
</tr>
</tbody>
</table>

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.

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 $r^3$ as an integer in $[1, 6]$, $K \in [10^{-6}, 10^{-3}]$, $\lambda \in [0.2, 0.45]$ and $\delta \in [10^{-4}, 10^{-2}]$. We roughly selected 4 groups of parameters in the ranges for each dataset and chose the one with

3http://www.csie.ntu.edu.tw/~cjlin/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 seen 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 verify 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 \( \mathbf{L}_i^T \), 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 \( L_i^T \) 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


