Postprint of Multi-label Label-Specific Feature Extraction Algorithm Based on Random Subspace

Authors: Zhang Jing, Li Yu, Li Peipei

Date: 2018-05-20T00:00:00+00:00

Abstract

Multi-label learning has been widely applied in numerous scenarios. In such learning problems, a sample can often be associated with multiple class labels. Since the unique attributes that class labels may carry (i.e., label-specific attributes) can be more beneficial for label classification, several multi-label learning algorithms based on label-specific attributes have been proposed. To address the issue of redundancy in the attribute space caused by the construction of label-specific attributes, this paper proposes a multi-label label-specific feature extraction algorithm LIFT_{RSM}. This method extracts effective feature information from the label-specific attribute space by comprehensively utilizing the random subspace model and pairwise constraint dimensionality reduction, aiming to improve classification performance. Experimental results on multiple datasets show that, compared with several classical multi-label algorithms, the proposed LIFT_{RSM} algorithm achieves superior classification performance.

Full Text

Preamble

Multi-label Label-Specific Feature Extraction Algorithm Based on Random Subspace

Zhang Jing, Li Yu†, Li Peipei

(School of Computer & Information, Hefei University of Technology, Hefei 230009, China)

Abstract: Multi-label learning has been widely used in many application scenarios. In this kind of learning problem, each instance is simultaneously assigned with more than one class label. Since different class labels might have their own unique characteristics (i.e., label-specific features) which would be

more useful for label classification, some multi-label learning approaches based on label-specific features have already been proposed. Aiming at the problem of redundant feature space caused by label-specific feature construction, this paper proposes a multi-label label-specific feature extraction algorithm named LIFT_{RSM}, which can improve classification performance by comprehensively using the random subspace method and the idea of pairwise constraint dimensionality reduction to extract effective feature information in the label-specific feature space. Experimental results on several datasets show that the proposed algorithm can achieve better classification results compared with several classical multi-label algorithms.

Key Words: multi-label learning; pairwise constraints; feature extraction; random subspace

0 Introduction

With the development of information technology, multi-label learning [1-3] has gradually become a research hotspot in data mining and has received extensive attention. Unlike traditional single-label data, each sample in multi-label data can simultaneously belong to multiple labels, making such data often lack unique semantics. Due to the multi-semantic nature of multi-label data, multi-label learning can be widely applied to many practical scenarios and has achieved good results in text classification, music emotion classification, semantic scene classification, bioinformatics, and other fields.

A multi-label learning problem can be formally described as follows: Given a d-dimensional sample space $\mathcal{X} = \mathbb{R}^d$ and a label set $\mathcal{L} = \{l_1, l_2, \ldots, l_q\}$ containing q labels, the main task is to learn a classification function $f : \mathcal{X} \to \mathcal{Y}$ from the training set $\mathcal{T} = \{(\mathbf{x}_i, Y_i) \mid 1 \le i \le p, \mathbf{x}_i \in \mathcal{X}, Y_i \subseteq \mathcal{L}\}$ that maps any unknown sample $\mathbf{x}$ to the corresponding label set Y. Since the relationships among labels in set $\mathcal{L}$ are not assumed to be mutually exclusive, the single-label learning framework is no longer applicable to such data.

Consequently, through continuous research by many scholars in recent years, a series of multi-label algorithms have been proposed. Summarizing existing algorithms, their main construction ideas can be roughly divided into three categories: problem transformation, algorithm adaptation, and ensemble methods. Problem transformation methods [4-6] transform the multi-label problem into several single-label problems by modifying the data, and then use mature single-label methods to handle the transformed problems. Although these methods are simple and not limited to specific algorithms, they ignore the correlation information among labels, which affects the learning performance to some extent. Algorithm adaptation methods [7-9] directly extend and improve traditional single-label learning algorithms to enhance their applicability and generalization ability, making them suitable for processing multi-label data. Ensemble methods [10-11] usually combine problem transformation and algorithm adaptation methods to handle multi-label learning problems in order to achieve better

learning performance.

When processing multi-label data, the above methods adopt the same strategy: using the same feature set to predict all class labels. Although this strategy has achieved good results in multi-label research, it is not the optimal choice. Since each label may have unique feature attributes (i.e., label-specific features) that are most relevant to the label and have stronger discriminative power for the corresponding label, Zhang et al. proposed the LIFT (multi-label learning with Label-specific FeaTures) algorithm [12] based on this viewpoint. Unlike existing strategies, the LIFT algorithm determines the label set of unknown samples with the help of label-specific features. However, during the construction of label-specific features, it fails to fully consider the correlation among samples, leading to increased dimensionality of label-specific features and redundant information in the label-specific feature space.

1.1 Pairwise Constraints

In many application domains, besides sample class labels, some other forms of background knowledge can also be used as supervision information, including pairwise constraints. Pairwise constraints refer to a relationship between two samples. Compared with class labels, pairwise constraints are more general and widely applicable, as they do not focus on the specific class of samples but only on whether two samples belong to the same class, making them easier to obtain. Moreover, equivalent pairwise constraint information can be relatively easily obtained from class label information, but not vice versa. Therefore, pairwise constraints are more universal than class labels.

Pairwise constraints can typically be divided into two types [13]: must-link (ML) and cannot-link (CL). Must-link constraints indicate that two samples belong to the same class, while cannot-link constraints require that two samples belong to different classes. Specifically, for a given sample set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, all must-link constraints can form a must-link set, formally expressed as $\mathcal{M} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same class}\}$. Correspondingly, the cannot-link set consists of all cannot-link constraints, denoted as $\mathcal{C} = \{(\mathbf{x}_p, \mathbf{x}_q) \mid \mathbf{x}_p \text{ and } \mathbf{x}_q \text{ belong to different classes}\}$.

1.2 Random Subspace

The random subspace method, proposed by Ho [14-15], is an effective ensemble learning method based on feature partitioning, originally used to overcome overfitting problems in decision tree classifiers. The basic idea is to randomly select different feature subsets from the original feature space to construct feature subspaces, then use each feature subspace to construct corresponding sub-classifiers, and finally integrate the classification results obtained from different sub-classifiers according to certain combination rules to obtain the final learning decision. During the random feature selection process, it can not only make fuller use of original feature information and reduce data redundancy, but also

effectively avoid the small sample problem. However, due to the randomness of feature selection, it cannot guarantee that all selected features contain effective discriminative information, making it difficult to ensure the accuracy of base classifiers.

2 Multi-Label Label-Specific Feature Extraction Algorithm Based on Random Subspace

2.1 Construction of Label-Specific Feature Space

The LIFT algorithm needs to examine the intrinsic properties of the attribute space under each label when constructing the label-specific feature space. Specifically, for any label $l_k \in \mathcal{L}$, the training set can be divided into two parts: a positive sample set $\mathcal{P}_k = \{\mathbf{x}_i \mid (\mathbf{x}_i, Y_i) \in \mathcal{T}, l_k \in Y_i\}$ and a negative sample set $\mathcal{N}_k = \{\mathbf{x}_i \mid (\mathbf{x}_i, Y_i) \in \mathcal{T}, l_k \notin Y_i\}$. Thus, $\mathcal{P}_k$ is the set of samples with label l_k, while $\mathcal{N}_k$ consists of samples not labeled with l_k.

In [12], the k-means algorithm is used to perform cluster analysis on the above two sets respectively. Here, set $\mathcal{P}_k$ can be partitioned into m_k^+ clusters with cluster centers denoted as $\{\mathbf{p}_{k1}, \mathbf{p}_{k2}, \ldots, \mathbf{p}_{km_k^+}\}$. Correspondingly, set $\mathcal{N}_k$ is partitioned into m_k^- clusters with cluster centers $\{\mathbf{n}_{k1}, \mathbf{n}_{k2}, \ldots, \mathbf{n}_{km_k^-}\}$. Reference [12] assigns the same weight to the clustering information of $\mathcal{P}_k$ and $\mathcal{N}_k$, thus setting the number of cluster centers to be equal, i.e., $m_k^+ = m_k^- = m_k$. Specifically, the number of clusters for sets $\mathcal{P}_k$ and $\mathcal{N}_k$ is obtained by the following formula:

$$m_k = \min\{\lfloor \gamma \cdot |\mathcal{P}_k| \rfloor, \lfloor \gamma \cdot |\mathcal{N}_k| \rfloor\}$$

where $|\cdot|$ denotes the cardinality of a set and $\gamma \in [0, 1]$ is a parameter controlling the number of clusters.

From the properties of clustering, the above two sets of cluster centers can respectively characterize the intrinsic structure of the corresponding sets. Therefore, based on this, label-specific features can be defined as follows:

$$\phi(\mathbf{x}) = [d(\mathbf{x}, \mathbf{p}_{k1}), \ldots, d(\mathbf{x}, \mathbf{p}_{km_k}), d(\mathbf{x}, \mathbf{n}_{k1}), \ldots, d(\mathbf{x}, \mathbf{n}_{km_k})]$$

where $d(\cdot, \cdot)$ returns the distance between two samples, and Euclidean distance is used in [12].

2.2 Feature Extraction Based on Random Subspace

2.2.1 Random Subspace Partitioning and Fusion Using the label-specific feature space constructed above, P features are randomly selected from the original D-dimensional space to build T different P-dimensional subspace sets, denoted as $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_T\}$. Here, $P < D$, and each subspace

$\mathcal{F}_t$ consists of P-dimensional samples, i.e., $\mathcal{F}_t = \{\mathbf{f}_{t1}, \mathbf{f}_{t2}, \ldots, \mathbf{f}_{tn}\}$. To clearly describe the neighbor relationships of samples in the subspace, the distance mean is used here to adaptively determine the number of neighbors for samples. Specifically, in any subspace $\mathcal{F}_t$, the neighbor relationship between samples is defined based on the mean distance M_t between sample $\mathbf{f}_{ti}$ and all samples, i.e., $M_t = \frac{1}{N}\sum_{j=1}^{N} d(\mathbf{f}_{ti}, \mathbf{f}_{tj})$. When the distance $d(\mathbf{f}_{ti}, \mathbf{f}_{tj})$ between samples $\mathbf{f}_{ti}$ and $\mathbf{f}_{tj}$ is less than M_t, $\mathbf{f}_{tj}$ is considered a neighbor of $\mathbf{f}_{ti}$; otherwise, there is no neighbor relationship between them. Thus, the number of neighbors for different samples is generally unequal.

For any subspace $\mathcal{F}_t$, construct the corresponding adaptive neighbor graph $\mathcal{G}_t^N$, non-neighbor graph $\mathcal{G}_t^F$, and between-class neighbor graph $\mathcal{G}_t^B$. Specifically, nodes in the graph represent specific samples, and edges reflect the neighbor relationships among samples. Based on the above graph relationships, define the weight matrix $\mathbf{S}_t^N$ for each sample and its corresponding neighbor samples, the weight matrix $\mathbf{S}_t^F$ for each sample and its corresponding non-neighbor samples, and the weight matrix $\mathbf{S}_t^B$ for each sample and its corresponding between-class neighbor samples. The weights of these matrices are defined as follows:

$$S_{t,ij}^N = \begin{cases} 1, & d_{t,ij} < M_t \text{ or } d_{t,ji} < M_t \\ 0, & \text{otherwise} \end{cases}$$

$$S_{t,ij}^F = \begin{cases} 1, & d_{t,ij} \geq M_t \text{ and } d_{t,ji} \geq M_t \\ 0, & \text{otherwise} \end{cases}$$

$$S_{t,ij}^B = \begin{cases} 1, & (\mathbf{f}_{ti}, \mathbf{f}_{tj}) \in \mathcal{C} \text{ and } d_{t,ij} < M_t \\ 0, & \text{otherwise} \end{cases}$$

where $d_{t,ij}$ is the Euclidean distance between samples, M_t is the mean distance between sample $\mathbf{f}_{ti}$ and all samples, and F_{ti} represents the distance between sample $\mathbf{f}_{ti}$ and its farthest sample of the same class.

To more effectively utilize subspace information to reflect the true distribution of data and reduce uncertainty caused by random feature selection, fuse the constructed T adaptive neighbor graphs, T non-neighbor graphs, and T between-class neighbor graphs to obtain the corresponding mixed neighbor graph $\mathcal{G}_{rsn}$, mixed non-neighbor graph $\mathcal{G}_{rsf}$, and mixed between-class neighbor graph $\mathcal{G}_{rsb}$. Then construct the corresponding weight matrices $\mathbf{S}_{rsn}$, $\mathbf{S}_{rsf}$, and $\mathbf{S}_{rsb}$ based on these mixed graph relationships. The weight matrices of these mixed graphs can be linearly reconstructed using the corresponding weight matrices from each subspace. Specifically, their relationships can be defined as follows:

$$\mathbf{S}_{rsn} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{S}_t^N, \quad \mathbf{S}_{rsf} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{S}_t^F, \quad \mathbf{S}_{rsb} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{S}_t^B$$

where $\mathbf{S}_{rsn}$, $\mathbf{S}_{rsf}$, and $\mathbf{S}_{rsb}$ are symmetric matrices, and $\mathbf{D}_{rsn}$, $\mathbf{D}_{rsf}$, and $\mathbf{D}_{rsb}$ are diagonal matrices whose diagonal elements are the column (or row) sums of the corresponding matrices in $\mathbf{S}_{rsn}$, $\mathbf{S}_{rsf}$, and $\mathbf{S}_{rsb}$, i.e., $D_{rsn,ii} = \sum_j S_{rsn,ij}$, $D_{rsf,ii} = \sum_j S_{rsf,ij}$, $D_{rsb,ii} = \sum_j S_{rsb,ij}$. $\mathbf{L}_{rsn} = \mathbf{D}_{rsn} - \mathbf{S}_{rsn}$, $\mathbf{L}_{rsf} = \mathbf{D}_{rsf} - \mathbf{S}_{rsf}$, and $\mathbf{L}_{rsb} = \mathbf{D}_{rsb} - \mathbf{S}_{rsb}$ are Laplacian matrices, which are symmetric positive semi-definite matrices.

2.2.2 Design of Objective Function For must-link constraints (ML), to effectively maintain the overall compactness within classes, this paper selects all samples of the same class corresponding to each sample to construct the weight matrix $\mathbf{S}_m$. Therefore, the within-class scatter matrix $\mathbf{Q}_m$ can be constructed based on the must-link set $\mathcal{M}$ to describe the compactness within classes, defined as follows:

$$\mathbf{Q}_m = \sum_{(\mathbf{x}i,\mathbf{x}_j)\in\mathcal{M}} \|\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j\|^2 = 2\mathbf{w}^T\mathbf{X}\mathbf{D}_m\mathbf{X}^T\mathbf{w} - 2\mathbf{w}^T\mathbf{X}\mathbf{S}_m\mathbf{X}^T\mathbf{w} = \mathbf{w}^T\mathbf{X}\mathbf{L}_m\mathbf{X}^T\mathbf{w}$$

where $\mathbf{S}_m$ is a symmetric matrix, $\mathbf{D}_m$ is a diagonal matrix, and $\mathbf{L}_m = \mathbf{D}_m - \mathbf{S}_m$.

For cannot-link constraints (CL), to fully reflect the differences between samples, this paper uses the mixed between-class neighbor graph $\mathcal{G}_{rsb}$ to adjust the original cannot-link set $\mathcal{C}$ and construct a new cannot-link set. Based on the corresponding weight matrix $\mathbf{S}_{rsb}$, the between-class mixed scatter matrix $\mathbf{Q}_{rsb}$ is constructed to characterize the degree of between-class dispersion, defined as:

$$\mathbf{Q}_{rsb} = \sum_{i,j} \|\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j\|^2 S_{rsb,ij} = \mathbf{w}^T\mathbf{X}(2\mathbf{D}_{rsb} - \mathbf{S}_{rsb} - \mathbf{S}_{rsb}^T)\mathbf{X}^T\mathbf{w} = \mathbf{w}^T\mathbf{X}\mathbf{L}_{rsb}\mathbf{X}^T\mathbf{w}$$

where $\mathbf{S}_{rsb}$ is a non-symmetric matrix, and $\mathbf{D}_{rsb}^{col}$ and $\mathbf{D}_{rsb}^{row}$ are diagonal matrices, i.e., $D_{rsb,ii}^{col} = \sum_j S_{rsb,ij}$ and $D_{rsb,ii}^{row} = \sum_j S_{rsb,ji}$.

So far, only information related to pairwise constraints has been considered, and the potential information contained in the sample set has not been involved. To make full use of the neighbor information among samples, the neighbor relationships among samples can be introduced into the dimensionality reduction process as local structural information based on the manifold assumption [16]. On the one hand, it is hoped that samples close to each other in the original space will also be close to each other in the low-dimensional space after projection. Therefore, based on the weight matrix $\mathbf{S}_{rsn}$ of the mixed neighbor graph $\mathcal{G}_{rsn}$, the mixed neighbor scatter matrix $\mathbf{Q}_{rsn}$ can be constructed to describe the closeness among neighbor points, defined as:

$$\mathbf{Q}_{rsn} = \sum_{i,j} \|\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j\|^2 S_{rsn,ij} = 2\mathbf{w}^T\mathbf{X}\mathbf{D}_{rsn}\mathbf{X}^T\mathbf{w} - 2\mathbf{w}^T\mathbf{X}\mathbf{S}_{rsn}\mathbf{X}^T\mathbf{w} = \mathbf{w}^T\mathbf{X}\mathbf{L}_{rsn}\mathbf{X}^T\mathbf{w}$$

On the other hand, for non-neighbor samples, it is expected that their projections in the low-dimensional space can be as scattered as possible. Based on this, using the weight matrix $\mathbf{S}_{rsf}$ of the mixed non-neighbor graph $\mathcal{G}_{rsf}$, the mixed non-neighbor scatter matrix $\mathbf{Q}_{rsf}$ is defined to measure the scattering degree among non-neighbor samples:

$$\mathbf{Q}_{rsf} = \sum_{i,j} \|\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j\|^2 S_{rsf,ij} = 2\mathbf{w}^T\mathbf{X}\mathbf{D}_{rsf}\mathbf{X}^T\mathbf{w} - 2\mathbf{w}^T\mathbf{X}\mathbf{S}_{rsf}\mathbf{X}^T\mathbf{w} = \mathbf{w}^T\mathbf{X}\mathbf{L}_{rsf}\mathbf{X}^T\mathbf{w}$$

Based on the above preparation, when designing the target transformation vector $\mathbf{w}$, pairwise constraint information should be used as guidance while fully utilizing the neighbor relationships among samples. Therefore, the final target transformation vector can be obtained by defining the following function:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T(\mathbf{Q}_{rsb} + \mathbf{Q}_{rsf})\mathbf{w}}{\mathbf{w}^T(\mathbf{Q}_m + \mathbf{Q}_{rsn})\mathbf{w}} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{X}(\mathbf{L}_{rsb} + \mathbf{L}_{rsf})\mathbf{X}^T\mathbf{w}}{\mathbf{w}^T\mathbf{X}(\mathbf{L}_m + \mathbf{L}_{rsn})\mathbf{X}^T\mathbf{w}}$$

where α and β are constant coefficients used to regulate the contributions of $\mathbf{Q}_{rsb}$ and $\mathbf{Q}_{rsn}$, respectively. If $\mathbf{X}(\mathbf{L}_m + \mathbf{L}_{rsn})\mathbf{X}^T$ is non-singular, the Lagrange method can be used to transform the above equation, converting the solution problem into solving for the eigenvector corresponding to the maximum generalized eigenvalue of the following equation:

$$\mathbf{X}(\mathbf{L}_{rsb} + \mathbf{L}_{rsf})\mathbf{X}^T\mathbf{w} = \lambda\mathbf{X}(\mathbf{L}_m + \mathbf{L}_{rsn})\mathbf{X}^T\mathbf{w}$$

The final dimension d is determined based on the threshold parameter thr, and the transformation matrix $\mathbf{W}$ is constructed by selecting the first d eigenvectors corresponding to the largest non-zero eigenvalues.

2.3 Algorithm Description

This section introduces the random subspace idea into the label-specific space, makes full use of pairwise constraint information and sample neighbor relationships, and proposes a multi-label label-specific feature extraction algorithm based on random subspace. The complete process from label-specific feature construction, subspace partitioning and fusion, feature extraction, classification model training to unknown sample prediction is shown below, and the detailed procedure can be summarized as follows:

Input: Training set X, clustering number control parameter γ, number of random subspaces T, feature subspace dimension P, contribution control parameters α and β, threshold parameter thr, unlabeled sample $\mathbf{x}'$.

Output: Predicted label set Y'.

The LIFT-RSM algorithm first constructs a label-specific feature space for each class label and centers it (steps 2-5). Then it uses the random subspace idea to partition the original label-specific space, fuses the neighbor relationships from each subspace, and uses pairwise constraint information to guide dimensionality reduction of the original label-specific space (steps 6-12). Next, it trains a binary classification model in the reduced label-specific feature space (steps 13-14). Finally, it predicts unknown samples (step 15).

3.1 Datasets

This paper uses five different publicly available multi-label datasets to experimentally validate the proposed multi-label label-specific feature extraction algorithm based on random subspace. The specific statistical information of these datasets is shown in . Since the selected datasets cover different application domains such as music, images, and text, and have different label properties, they have strong generalizability.

In , $|S|$ represents the number of samples, $\dim(S)$ represents the number of attributes, $L(S)$ represents the number of labels, $\mathrm{LCard}(S)$ represents label cardinality (the average number of relevant labels per sample), and $\mathrm{LDen}(S)$ represents label density (the label cardinality normalized by the number of labels).

3.2.1 Evaluation Metrics

In multi-label learning, since each sample can simultaneously belong to multiple labels, evaluating the effectiveness of multi-label algorithms is typically more complex than evaluating single-label algorithms. Traditional evaluation metrics widely used in single-label algorithms, such as accuracy, recall, and precision, are no longer suitable for multi-label problems. Therefore, specialized multi-label evaluation metrics are needed to verify algorithm effectiveness. Currently, multi-label evaluation metrics mainly measure algorithm performance from two perspectives: sample-based and label-based, and can be roughly divided into two categories [1]: sample-based metrics [17] and label-based metrics [18]. In this paper' s experiments, the following five evaluation metrics are selected to comprehensively assess the performance of the proposed algorithm, including one sample-based metric: Hamming Loss (HL), and four label ranking-based metrics: One-Error (OE), Ranking Loss (RL), Coverage (CV), and Average Precision (AP).

These five metrics evaluate algorithm performance from different perspectives, directly reflected in the numerical values of the metrics. Among them, larger Average Precision values indicate better algorithm performance, with optimal performance achieved when the value is 1. For the remaining four metrics, smaller values indicate better algorithm performance, with optimal performance achieved when the value is 0. Detailed introductions to these evaluation metrics can be found in [1] and are not repeated here.

3.2.2 Comparison Algorithms

This paper selects five classic multi-label learning methods as comparison algorithms to compare and analyze with the proposed LIFT-RSM algorithm. These five algorithms include: the k-nearest neighbor-based ML-kNN algorithm [7], the LIFT algorithm [12], the multi-label dimensionality reduction algorithm MDDM [19], MLNB [20], and MLSI [21]. In the experiments, for the LIFT and LIFT-RSM algorithms, parameter γ is adjusted in the interval [0,1] with a step size of 0.1, and finally set to $\gamma = 0.2$. For the MLNB and MLSI algorithms, the setting retains 98% of the original information. For the MDDM and ML-kNN algorithms, default parameter configurations are selected according to the recommendations in the corresponding literature.

Unless otherwise specified, in the LIFT_{RSM} algorithm, the contribution control parameters α and β are both set to 0.05, the number of random subspaces T is set to 10, and the feature subspace dimension P is set to 20. If the dimension of the original space is less than 20, then P is set to $\lfloor 0.3 \times d \rfloor$. In the dimensionality reduction process, 95% of the information from the original label-specific attributes is retained. Except for the MLNB algorithm, a linear kernel LIBSVM is used as the base classifier for all other algorithms. All experiments in this paper are completed on a host with 4GB memory and a 2.50GHz processor, with a 64-bit Windows 7 operating system, and MATLAB 2014a is selected as the development platform.

3.3 Results Analysis

For the Image, Scene, and Flags datasets, this paper randomly extracts 80% of the samples from each dataset as the training set and the remaining 20% as the test set. The sampling process is repeated 50 times and the mean of the 50 experiments is recorded. For the remaining datasets, the original training and test sets are used, and the experiments are repeated 50 times with the mean recorded.

- record the experimental results of each algorithm on the five datasets, with results expressed as mean values. For each evaluation metric, the symbol $\downarrow$ ($\uparrow$) indicates that smaller (larger) values represent better algorithm performance. Additionally, the optimal values of each evaluation metric are underlined.

Observing the results in -, it can be seen that the proposed multi-label label-specific feature extraction algorithm based on random subspace, LIFT_{RSM}, achieves good classification performance. For the Emotions and Image datasets, except for Coverage and Ranking Loss, LIFT_{RSM} outperforms other comparison algorithms on the remaining three evaluation metrics. For the Scene and Flags datasets, except for Coverage, LIFT_{RSM} outperforms comparison algorithms on the remaining four metrics. For the Slashdot dataset, LIFT_{RSM} achieves results of 0.0397, 0.4096, 2.4202, 0.0948, and 0.6871 on the HL, OE, CV, RL, and AP metrics, respectively, showing varying degrees of improvement com-

pared to the original LIFT algorithm, with even more significant improvements compared to other algorithms.

It is worth noting that the proposed algorithm performs slightly worse on some metrics in certain datasets. Upon analysis, the main reason is that the relevant datasets have larger label density, with many instances having multiple labels simultaneously, which increases the number of marginal samples and noisy samples in each actual class, affecting feature extraction performance and resulting in suboptimal classification effects.

Taking the Flags, Scene, Emotions, and Image datasets as examples, [Figure 1: see original paper]-[Figure 4: see original paper] respectively show the statistical comparison of the label-specific attribute dimensions obtained after learning with the LIFT and LIFT_{RSM} algorithms. It can be observed from [Figure 1: see original paper]-[Figure 4: see original paper] that for different datasets and across all class labels, the final label-specific attribute dimensions obtained by LIFT_{RSM} are all lower than those of the LIFT algorithm. Taking the Scene dataset as an example, among its 6 labels, the original label-specific attribute dimensions learned by the LIFT algorithm are 138, 118, 128, 139, 171, and 138, respectively, while the corresponding attribute dimensions decrease to 110, 83, 96, 107, 144, and 110 after learning with LIFT_{RSM}. This demonstrates that the label-specific attribute dimensions obtained by LIFT_{RSM} can indeed be reduced to a certain extent. Although LIFT_{RSM} performs slightly worse than comparison algorithms on some evaluation metrics, overall, LIFT_{RSM} can still achieve good learning and classification performance.

The LIFT_{RSM} algorithm can more accurately represent the correlations among samples by fusing neighbor relationships from each random subspace, thus effectively solving multi-label data classification problems. In summary, the proposed LIFT_{RSM} algorithm is generally superior to other comparison algorithms in comprehensive performance, improving classifier performance and achieving good results.

4 Conclusion

Unlike previous multi-label learning algorithms, the LIFT algorithm focuses on examining the impact of attribute space operations on multi-label learning performance. Based on the LIFT algorithm, this paper utilizes the random subspace model to partition the original label-specific space, fuses neighbor relationships from each subspace, and employs the idea of using pairwise constraint information to guide dimensionality reduction, proposing a multi-label label-specific feature extraction algorithm based on random subspace. A series of experimental results show that the proposed algorithm is generally superior to other classic algorithms, meeting expected goals and verifying the algorithm' s effectiveness. In future research, label correlations can be incorporated into label-specific feature extraction to further improve the learning performance of multi-label algorithms. Additionally, the current algorithm has relatively many

parameters, and finding effective adaptive methods to reduce the number of required parameters is also a direction for future work.

References

[1] Zhang M L, Zhou Z H. A Review on multi-label learning algorithms [J]. IEEE Trans on Knowledge & Data Engineering, 2014, 26 (8): 1819-1837.

[2] Tsoumakas G, Katakis I. Multi-label classification: an overview [J]. International Journal of Data Warehousing & Mining, 2009, 3 (3): 1-13.

[3] Zhou Z H, Zhang M L. Multi-label learning [M]// Encyclopedia of Machine Learning and Data Mining, Berlin: Springer, 2017, 875-881.

[4] Zhang M L, Zhang K. Multi-label learning by exploiting label dependency [C]// Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2010: 999-1008.

[5] Fürnkranz J, Hüllermeier E, Mencía E L, et al. Multilabel classification via calibrated label ranking [J]. Machine Learning, 2014, 73 (2): 133-153.

[6] Boutell M R, Luo J, Shen X, et al. Learning multi-label scene classification. [J]. Pattern Recognition, 2004, 37 (9): 1757-1771.

[7] Zhang M L, Zhou Z H. ML-KNN: A lazy learning approach to multi-label learning [J]. Pattern Recognition, 2007, 40 (7): 2038-2048.

[8] Zhang M L, Zhou Z H. Multilabel neural networks with applications to functional genomics and text categorization [J]. IEEE Trans on Knowledge & Data Engineering, 2006, 18 (10): 1338-1351.

[9] Schapire R E, Singer Y. BoosTexter: A Boosting-based System for Text Categorization [J]. Machine Learning, 2000, 39 (2-3): 135-168.

[10] Tsoumakas G, Vlahavas I. Random k-labelsets: an ensemble method for multilabel classification [C]// Proc of the 18th European Conference on Machine Learning. Berlin: Springer, 2007: 406-417.

[11] Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification [C]// Lecture Notes in Computer Science. 2004, : 22-30.

[12] Zhang M L, Wu L. Lift: Multi-label learning with label-specific features. [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2015, 37 (1): 107-120.

[13] Klein D, Kamvar S D, Manning C D. From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering [C]// Proc of the 19th International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers Inc. 2002: 307-314.

[14] Ho T K. Random decision forests [C]// Proc of International Conference on Document Analysis and Recognition. 2002: 278.

[15] Ho T K. The random subspace method for constructing decision forests [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 1998, 20 (8): 832-844.

[16] Liu Jianwei, Liu Yuan, Luo Xionglin. Semi-supervised learning methods [J]. Chinese Journal of Computers, 2015, 38 (8): 1592-1610.

[17] Godbole S, Sarawagi S. Discriminative methods for multi-labeled classification [J]. Machine Learning, 2011, 85 (3): 254-269.

[18] Yang Y. An evaluation of statistical approaches to text categorization [J]. Information Retrieval Journal, 1999, 1 (1): 69-90.

[19] Zhang Y, Zhou Z H. Multilabel dimensionality reduction via dependence maximization [J]. ACM Trans on Knowledge Discovery from Data, 2010, 4 (3): 1503-1505.

[20] Zhang M L, Peña J M, Robles V. Feature selection for multi-label naive Bayes classification [J]. Information Sciences, 2009, 179 (19): 3218-3229.

[21] Yu K, Yu S, Tresp V. Multi-label informed latent semantic indexing [C]// Proc of International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2005: 258-265.

Note: Figure translations are in progress. See original paper for figures.