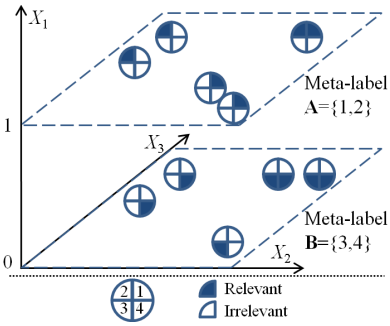
2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

# Multi-Label Classification with Meta-Label-Specific Features

Lu Sun, Mineichi Kudo and Keigo Kimura
Graduate School of Information Science and Technology
Hokkaido University, Sapporo 060-0814, JAPAN
Email: {sunlu,mine,kkimura}@main.ist.hokudai.ac.jp

***Abstract*—Multi-label classification has attracted many attentions in various fields, such as text categorization and semantic image annotation. Aiming to classify an instance into multiple labels, various multi-label classification methods have been proposed. However, the existing methods typically build models in the identical feature (sub)space for all labels, possibly inconsistent with real-world problems. In this paper, we develop a novel method based on the assumption that meta-labels with specific features exist in the scenario of multi-label classification. The proposed method consists of meta-label learning and specific feature selection. Experiments on twelve benchmark multi-label datasets show the efficiency of the proposed method compared with several state-of-the-art methods.**




| No. | Features | | | Labels | | | |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ |
| 1 | 1 | 1.0 | 0.1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 0.9 | 0.2 | 1 | 0 | 0 | 0 |
| 3 | 1 | 0.8 | 0.8 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0.2 | 0.8 | 0 | 1 | 0 | 0 |
| 5 | 1 | 0.1 | 0.7 | 0 | 1 | 0 | 0 |
| 6 | 0 | 0.9 | 0.1 | 0 | 0 | 1 | 0 |
| 7 | 0 | 0.7 | 0.8 | 0 | 0 | 1 | 1 |
| 8 | 0 | 0.9 | 0.8 | 0 | 0 | 1 | 1 |
| 9 | 0 | 0.2 | 0.8 | 0 | 0 | 0 | 1 |
| 10 | 0 | 0.1 | 0.7 | 0 | 0 | 0 | 1 |

Fig. 1. Meta-labels with specific feature subsets.

## I. Introduction

Multi-Label Classification (MLC) associates an unseen instance with a relevant label subset. MLC is ubiquitous in real-world problems. For example, a single image probably belongs to several semantic concepts, like "sky", "building", "field", etc; a news article is possibly relevant to a set of topics, like "economic", "politics", "sports", etc. Obviously, this special characteristic imposes significant challenges to traditional classification algorithms.

To handle the MLC problems, various methods have been proposed. The previous efforts on MLC concentrates mainly on two aspects: modeling label correlations and reducing the dimensionality. A number of researches [1], [2] have shown that modeling label correlations is very crucial for performing accurate classification. On the other hand, a variety of dimension reduction approaches [3], [4] have been developed in order to reduce the execution time and storage space, and improve the classification performance.

Although these methods have achieved much success, some limitations are still remained. Label correlations are typically modeled on the basis of the label space, without the information from the instance space, although similar instances possibly belong to similar labels. In addition, Dimension Reduction (DR) is usually performed globally, transforming the original feature space into a single subspace in common to all labels. Such a global DR could harm the discriminative ability with respect to a specific subset of labels. Some papers have been proposed to address the limitations. For instance, [5], [6], [7] and [8] proposed to model conditional label dependency or apply local (label-specific) DR. However, further improvement in terms of complexity and accuracy is recently required.

In this study, we propose an MLC method based on two assumptions: (a) meta-labels, strong and reasonable label combinations, exist implicitly in the label space; (b) only a fraction of features is relevant to a meta-label, and different meta-labels relate with different feature subsets. Such assumptions hold in several real-world observations. For example, in image annotation, the tags "ocean" and "sky" can be viewed as forming a meta-label, since they highly correlates and shares similar color features; in text categorization, the topics "science" and "technology" have strong correlation with specific features, like "research", "laboratory", "institute", etc. Fig. 1 shows a toy multi-label example satisfying the assumptions. In Fig. 1, two meta-labels $\mathbf{A} = \{1, 2\}$ and $\mathbf{B} = \{3, 4\}$ can be found by preserving the strong label dependency within each meta-label. In addition, mining meta-label-specific features is also interesting, since feature $X_1$ is useful for separating meta-labels, but is useless for classification inside a meta-label.

In order to justify the assumptions, we propose a novel MLC method using Meta-Label-Specific Features (MLSF). MLSF consists of meta-label learning and specific features mining. In meta-label learning, highly correlated labels are grouped together using the information from both the label and instance spaces. We discuss the usage of the Spectral Clustering [9] technique. In specific feature selection, we use the LASSO [14] with an efficient optimization approach, Alternating Direction Method of Multipliers (ADMM) [10]. To capture label correlations in each meta-label, Classifier Chains (CC) [1] is built on the meta-label-specific features. To evaluate the performance of MLSF, extensive experiments are conducted on twelve multi-label datasets.

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

## II. Related works

In order to overcome the curse of dimensionality and make the algorithms tractable for large-scale multi-label datasets, a number of recent research works focus on Feature Space Dimension Reduction (FS-DR).

In general the unsupervised DR approaches like PCA cannot utilize the information from labels, so supervised dimension reduction is preferred. To the best of the authors' knowledge, the first MLC method with the help of supervised FS-DR is MLSI [3]. In MLSI, a supervised Latent Semantic Indexing (LSI) approach was developed to map the input features into a subspace by preserving the label information. By maximizing the feature-label dependence under the Hilbert-Schmidt independence criterion, MDDM [11] derived a closed-form solution to efficiently find the projection into the feature subspace. In addition, several traditional supervised DR techniques, such as Canonical Correlation Analysis (CCA), Linear Discriminant Analysis (LDA), and Hypergraph Spectral Learning (HSL), are specifically extended for MLC [4], [12], [13].

All the above FS-DR methods take a global strategy, finding an identical feature subspace globally for all the labels. However, it is more reasonable to think that each label holds a specific supporting feature subset. Moreover, global FS-DR typically seeks a linear feature projection for efficiency, saving both relevant and irrelevant features. To overcome the limitations, local FS-DR methods [7], [8] have been proposed to find label-specific features. In LIFT [7], label-specific features are extracted by cluster analysis on the positive and negative instances of each label. LLSF [8] selects label-specific features by optimizing the least squares problem with constraints of label correlations and feature sparsity. Although the local FS-DR methods have gained much success, they still have some limitations. They are difficult to model label correlations, since LIFT simply ignores label correlations and LLSF models them in an implicit manner. In addition, as shall be shown in Sec. IV, they typically consumes more processing time compared with MLC methods of linear complexity.

## III. The MLSF method

In order to overcome the limitations that previous FS-DR methods have, in this study we propose to use Meta-Label-Specific Features (MLSF). Under the assumption on the existence of meta-labels with specific feature subsets, in MLSF, label correlations extracted from both the label and instance space are modeled in the form of meta-labels, and then meta-label specific features are selected following the local FS-DR strategy.

### A. Preliminary

A multi-label instance is represented by a pair $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$, $\mathbf{y} \in \mathcal{Y} \subseteq \{0,1\}^L$ denote the feature and label vectors, respectively. For the $i$th instance $(\mathbf{x}_i, \mathbf{y}_i)$, the $j$th element $y_{ij}$ of $\mathbf{y}_i$ becomes one when the $j$th label is relevant to $\mathbf{x}_i$, and zero otherwise. Given a dataset of $N$ instances $\mathcal{T} = [\mathbf{X}, \mathbf{Y}]$, we use $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^\intercal \in \mathbb{R}^{N \times D}$ and $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^\intercal \in \{0,1\}^{N \times L}$ to represent the feature and label matrices, respectively.

**Algorithm 1** Meta-label learning

**Input:** $\mathbf{X}$: data matrix, $\mathbf{Y}$: label matrix, $\alpha$, $\epsilon$, $K$: parameters
**Output:** $\mathbf{m}$: meta-label membership vector, $\mathbf{m} \in \{1, ..., K\}^L$

1: Compute $\mathbf{A}$ by (3) according to $\alpha$ and $\epsilon$;
2: $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \text{diag}(\sum_k A_{jk})$;
3: Solve $\mathbf{Lu} = \lambda \mathbf{Du}$ by $K$ smallest eigenvalues;
4: $\mathbf{m} \leftarrow k$-means$(\mathbf{U}, K)$, where $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_K]$.

### B. Meta-label learning

In this section, we embed label correlations into meta-labels in such a way that the member labels in a meta-label share strong dependency with each other but have weak dependency with the other non-member labels. To this end, we construct a graph $\mathbf{G} =< \mathbf{V}, \mathbf{E} >$ in the label space, where $\mathbf{V}$ denotes the vertex/label set, and $\mathbf{E}$ is the edge set containing edges between each label pair. Given an appropriate affinity matrix $\mathbf{A}$ on $\mathbf{E}$, meta-label learning can be considered as a graph cut problem: cutting the graph $\mathbf{G}$ into a set of sub-graphs.

For constructing affinity matrix $\mathbf{A}$, we use two different sources: the label space $\mathcal{Y}$ and the instance space $\mathcal{X}$. To model the affinity obtained from the label space, Jaccard index, a metricated variant of mutual information, is used:

$$A_{jk}^{(L)} \coloneqq \frac{\sum_{i=1}^N y_{ij} \cdot y_{ik}}{\sum_{i=1}^N (y_{ij} + y_{ik} - y_{ij} \cdot y_{ik})}. \tag{1}$$

Next, focusing on the instance space, we have

$$A_{jk}^{(I)} \coloneqq \mathrm{e}^{-\|\boldsymbol{\mu}_j - \boldsymbol{\mu}_k\|_2^2}, \quad \text{where } \boldsymbol{\mu}_j = \frac{\sum_{i=1}^N \mathbf{x}_i \cdot y_{ij}}{\sum_{i=1}^N y_{ij}}. \tag{2}$$

Last, by $\epsilon$-neighborhood, we combine these two matrices into one the affinity matrix $\mathbf{A} = \{A_{jk}\}_{j,k=1}^L$ as follows,

$$A_{jk} \coloneqq \begin{cases} \alpha A_{jk}^{(L)} + (1-\alpha) A_{jk}^{(I)} & (A_{jk} > \epsilon) \\ 0 & (A_{jk} \leq \epsilon), \end{cases} \tag{3}$$

where $\alpha \in [0,1]$ is a balance factor.

By regarding $\mathbf{A}$ as the edge weight matrix on $\mathbf{E}$, $\mathbf{G}$ becomes a graph representation of the label space. Then, to cut $\mathbf{G}$ into $K$ sub-graphs, i.e., $K$ meta-labels, is equivalent to perform $k$-means on $K$ smallest eigenvectors $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_K]$ of the generalized eigenvalue problem:

$$\mathbf{Lu} = \lambda \mathbf{Du}, \tag{4}$$

where $\mathbf{D}$ is the diagonal degree matrix, $\mathbf{D} = (D_{jj}) = (\sum_k A_{jk})$, and $\mathbf{L}$ denotes the Laplacian matrix, $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Applying $k$-means on $\mathbf{U}$, we have the meta-label membership vector $\mathbf{m} = \{m_j\}_{j=1}^L \in \{1, ..., K\}^L$, where $m_j = k$ indicates the $j$th label belongs to the $k$th meta-label. The pesudo code of meta-label learning is depicted in Algorithm 1.

**Algorithm 2** Specific feature selection

**Input:** $\mathbf{X}$: data matrix, $\mathbf{Y}$: label matrix, $\mathbf{m}$: meta-label membership, $\gamma$, $\rho$: parameters
**Output:** $\mathbf{V}$: regression parameter matrix

1: **for** $k \in \{1, ..., K\}$ **do**
2: $\quad \mathbf{Z}(:, k) \leftarrow \text{bi2de}(\mathbf{Y}(:, \mathbf{m}==k))$;
3: $\mathbf{V} := \mathbf{0}$, $\mathbf{\Lambda} := \mathbf{0}$;
4: **repeat**
5: $\quad \mathbf{W} \leftarrow (\mathbf{X}^\intercal\mathbf{X} + \rho\mathbf{I})^{-1}(\mathbf{X}^\intercal\mathbf{Z} + \rho\mathbf{V} - \mathbf{\Lambda})$;
6: $\quad \mathbf{V} \leftarrow \mathbf{W} + \mathbf{\Lambda}/\rho$;
7: $\quad V_j = \text{sign}(V_j) \cdot \max(0, |V_j| - \gamma/\rho)$, $\forall j$;
8: $\quad \mathbf{\Lambda} \leftarrow \mathbf{\Lambda} + \rho(\mathbf{W} - \mathbf{V})$;
9: **until** Convergence

### C. Specific feature selection

Next, we find meta-label-specific feature subsets. For this end, we transform $\mathbf{Y} \in \{0, 1\}^{N\times L}$ into the meta-label matrix $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_K] \in \mathbb{Z}^{N\times K}$. Here $\mathbf{Y}$ is firstly partitioned into $K$ parts by $\mathbf{m}$, then each part is encoded into a meta-label vector $\mathbf{z}$ by converting binary to decimal. Hence, we can use multivariate linear regression with $\ell_1$ loss. That is, LASSO [14], whose objective function is given by

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{XW} - \mathbf{Z}\|_2^2 + \gamma\|\mathbf{W}\|_1, \tag{5}$$

where $\gamma$ controls the sparsity of parameter matrix $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_K] \in \mathbb{R}^{D\times K}$. Here we treat nonzero elements of $\mathbf{w}_k$ as the meta-label specific features for the $k$th meta-label.

Lasso regression is a convex optimization problem, so it looks easy to solve. However, it is not trivial to efficiently optimize the objective function due to the $\ell_1$ penalty. In this study, we employ the Alternating Direction Method of Multiplier (ADMM) [10] to separate (5) into two sub-problems, which could be efficiently addressed. By employing a dummy variable matrix $\mathbf{V} \in \mathbb{R}^{D\times K}$ into (5), it can be rewritten in the augmented Lagragian form $L(\mathbf{W}, \mathbf{V}, \mathbf{\Lambda})$:

$$\frac{1}{2}\|\mathbf{XW}-\mathbf{Z}\|_2^2+\gamma\|\mathbf{V}\|_1+\frac{\rho}{2}\|\mathbf{W}-\mathbf{V}\|_2^2+\text{vec}(\mathbf{\Lambda})^\intercal\text{vec}(\mathbf{W}-\mathbf{V}), \tag{6}$$

where $\mathbf{\Lambda} \in \mathbb{R}^{D\times K}$ is the Lagrange multiplier matrix. ADMM performs the following iterations to optimize (6):

$$\mathbf{W}^{t+1} = \arg\min_{\mathbf{W}} L(\mathbf{W}, \mathbf{V}^t, \mathbf{\Lambda}^t), \tag{7}$$

$$\mathbf{V}^{t+1} = \arg\min_{\mathbf{V}} L(\mathbf{W}^{t+1}, \mathbf{V}, \mathbf{\Lambda}^t), \tag{8}$$

$$\mathbf{\Lambda}^{t+1} = \mathbf{\Lambda}^t + \rho(\mathbf{W}^{t+1} - \mathbf{V}^{t+1}). \tag{9}$$

In this way, ADMM separates the original optimization problem (5) into two sub-problems (7) and (8). Specifically, (7) is simply addressed by ridge regression, while (8) can be solved by the soft thresholding technique. After the convergence, $\mathbf{V}$ instead of $\mathbf{W}$ will be used as the sparse matrix indicating meta-label-specific features. The pesudo code of specific feature selection is given in Algorithm 2.

**Algorithm 3** MLSF

**Input:** $\mathbf{X}$: data matrix, $\mathbf{Y}$: label matrix, $\hat{\mathbf{x}}$: test instance, $\alpha$, $\epsilon$, $K$, $\gamma$, $\rho$: parameters
**Output:** $\hat{\mathbf{y}}$: prediction label set

**Training**:
1: $\mathbf{m} \leftarrow$ <Algorithm 1>$(\mathbf{X}, \mathbf{Y}, \alpha, \epsilon, K)$;
2: $\mathbf{V} \leftarrow$ <Algorithm 2>$(\mathbf{X}, \mathbf{Y}, \mathbf{m}, \gamma, \rho)$;
3: **for** $k \in \{1, ..., K\}$ **do**
4: $\quad \mathbf{h}_k : \mathbf{X}(:, \mathbf{v}_k\neq 0) \mapsto \mathbf{Y}(:, \mathbf{m}==k)$;

**Testing**:
5: **for** $k \in \{1, ..., K\}$ **do**
6: $\quad \hat{\mathbf{y}}(\mathbf{m}==k) \leftarrow \mathbf{h}_k(\hat{\mathbf{x}}(\mathbf{v}_k\neq 0))$;

### D. Discussion

After meta-label learning and specific features mining, in order to capture the correlations preserved in meta-labels, we employ a multi-label classifier, such as PairWise (PW) [15], Label Powerset (LP) [2] and Classifier Chains (CC) [1]. In this study CC is constructed for each meta-label, because CC can efficiently captures label dependency by randomly building a fully-connected Bayesian network in the label space.

Algorithm 3 illustrates the complete procedure of MLSF. In the training phase (Steps 1 to 4), MLSF firstly finds meta-label membership $\mathbf{m}$ to know which label belongs to which meta-label and regression matrix $\mathbf{V}$ by Steps 1 and 2; Then a CC classifier $\mathbf{h}_k$ is built on the data matrix with specific features for the $k$th meta-label, $k \in \{1, ..., K\}$, by Steps 3 and 4. In the testing phase (Steps 5 and 6), a test instance with meta-label-specific features is feed to each $\mathbf{h}$ for the prediction on corresponding meta-label.

In time complexity of MLSF, Step 1 has $O(KL^2 + i_1K^2L)$ and Step 2 has $O(i_2KD^2)$ in Algorithm 3, where $i_1$ and $i_2$ are the number of iterations in $k$-means and ADMM, respectively. The complexity in Steps 3 and 4 is $O(KT(dNl))$, where $T(\cdot)$ denotes the complexity of the classifier $\mathbf{h}$, while $d$ ($d < D$) and $l$ ($l < L$) is the number of specific features and labels of each meta-label, respectively.

## IV. Experiments

### A. Datasets and evaluation metrics

We conducted experiments on a variety of real-world multi-label datasets [16], which are summarized in Table I. In Table I, "Card." and "Den." denote the label cardinality and label density, respectively. Based on the problem size, the datasets are separated into regular-scale sets (the first six sets) and large-scale sets (the rest six sets).

Given a test dataset $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_t}$, we use four evaluation metrics to report the experimental results.

- **Exact-Match** $:= \frac{1}{N_t}\sum_{i=1}^{N_t} \mathbb{1}_{\hat{\mathbf{y}}_i=\mathbf{y}_i}$,
- **Hamming-Score** $:= \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}_{\hat{y}_{i\ell}=y_{i\ell}}$,
- **Macro-F1** $:= \frac{1}{L}\sum_{j=1}^{L}\frac{2\sum_{i=1}^{N_t}\hat{y}_{ij}\cdot y_{ij}}{\sum_{i=1}^{N_t}\hat{y}_{ij}+\sum_{i=1}^{N_t}y_{ij}}$,
- **Micro-F1** $:= \frac{2\sum_{j=1}^{L}\sum_{i=1}^{N_t}\hat{y}_{ij}\cdot y_{ij}}{\sum_{j=1}^{L}\sum_{i=1}^{N_t}\hat{y}_{ij}+\sum_{j=1}^{L}\sum_{i=1}^{N_t}y_{ij}}$.

TABLE I
MULTI-LABEL DATASETS USED IN EXPERIMENTS. "CARD.", "DEN." AND "TYPE" DENOTE THE LABEL CARDINALITY, THE LABEL DENSITY AND THE TYPE OF FEATURES, RESPECTIVELY.

| Dataset | $N$ | $D$ | $L$ | Card. | Den. | Type | Domain |
|---|---|---|---|---|---|---|---|
| Emotions | 593 | 72 | 6 | 1.869 | 0.311 | numeric | music |
| Scene | 2407 | 294 | 6 | 1.074 | 0.179 | numeric | image |
| Yeast | 2417 | 103 | 14 | 4.237 | 0.303 | numeric | biology |
| Genbase | 662 | 1186 | 27 | 1.252 | 0.046 | nominal | biology |
| Medical | 978 | 1449 | 45 | 1.245 | 0.028 | nominal | text |
| Enron | 1702 | 1001 | 53 | 3.378 | 0.064 | nominal | text |
| Rcv1s1 | 6000 | 944 | 101 | 2.880 | 0.029 | numeric | text |
| Rcv1s2 | 6000 | 944 | 101 | 2.634 | 0.026 | numeric | text |
| Mediamill | 43907 | 120 | 101 | 4.376 | 0.043 | numeric | video |
| Bibtex | 7395 | 1836 | 159 | 2.402 | 0.015 | nominal | text |
| Corel16k1 | 13766 | 500 | 153 | 2.859 | 0.019 | nominal | image |
| Corel16k2 | 13761 | 500 | 164 | 2.867 | 0.018 | nominal | image |

Exact-Match measures how well label correlations are modeled, Hamming-Score emphasizes on the accuracy on label-instance pairs, while Macro-F1 and Micro-F1 take the partial match of label sets into account.

### B. Compared methods and configuration

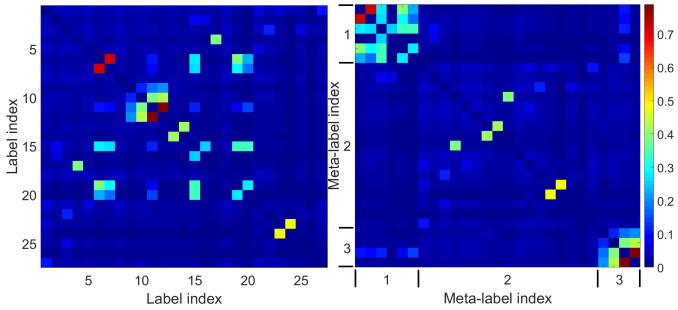
The proposed **MLSF** was compared with four popular MLC methods:

- **BR** [17]: a baseline method. A multi-label problem is transformed to $L$ single-label problems.
- **MDDM** [11]: a global FS-DR method. Features are projected into the low-dimensional subspace by maximizing the feature-label dependence.
- **LIFT** [7]: a local FS-DR method. Label-specific features are selected by cluster analysis.
- **LLSF** [8]: a local FS-DR method. Label-specific features are selected by preserving label correlations.

BR was introduced as a baseline MLC method with linear time complexity in the problem size. MDDM was chosen as a representative of global FS-DR methods, which outperformed several global FS-DR methods, like PCA, LPP [18], MLSI [3], as reported in [11]. As local FS-DR methods, LIFT and LLSF were chosen. They have been shown their performance advantages in comparison with several state-of-the-art MLC methods in [7], [8].

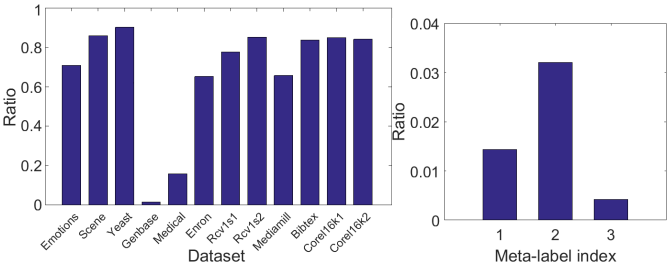
In the experiments, *5-fold cross validation* was performed to evaluate the classification performance. For fair comparison, BR with a linear SVM of LIBSVM [19] was used as the multi-label classifier for MDDM, LIFT, LLSF and MLSF. In parameter setting, we set the parameters of LIFT and LLSF as suggested by the authors, and set the dimensionality of the feature subspace to 30 for its optimal average performance. For MLSF, to balance the classification accuracy and processing time, we set the five parameters $K$, $\epsilon$, $\alpha$, $\gamma$ and $\rho$ as $\lceil L/10 \rceil$, 0.01, 0.8, 0.01 and 1, respectively. In addition, to make MDDM executable for all datasets, we randomly sampled 8000 instances for Mediamill, Corel16k1 and Corel16k2. We implemented the MATLAB codes of BR[1] and MLSF[1], and

[1]https://github.com/futuresun912/MLSF.git




(a) Affinity matrix $\mathbf{A}$ (b) $\mathbf{A}$ reindexed by $\mathbf{m}$ ($K = 3$)

Fig. 2. Label affinity matrices of Genbase.




(a) Feature ratio of all datasets (b) Feature ratio of Genbase

Fig. 3. The average ratio of meta-label-specific features.

obtained the MATLAB codes of MDDM[2], LIFT[3] and LLSF[4] from the authors. Experiments were performed in a computer configured with a Intel Quad-Core i7-4770 CPU at 3.4GHz with 4GB RAM.

### C. Experimental results

First we evaluated the degree to which labels are correlated to each other, and thus, how well our meta-label modeling succeeded to choose specific feature subsets. Fig. 2 shows the affinity matrices of Genbase, where the warmer the color, the stronger the label correlation. The axes indicate the (meta-)label index. On Genbase, 27 labels were grouped into 3 meta-labels according to the membership indicator $\mathbf{m}$, which was produced by Algorithm 1. Fig. 2a shows the affinity matrix $\mathbf{A}$ calculated by (3), while Fig. 2b shows the affinity matrix of Fig. 2a reindexed by $\mathbf{m}$. In Fig. 2b, most of strong label correlations have been saved within meta-labels. For example, the 3rd meta-label in Fig. 2b was formed by the 9-12th labels of Fig. 2a. In addition, from Fig. 3a, we know how many, possibly redundant, features have been removed. On Genbase and Medical, the feature size was dramatically reduced. In Fig. 3b, we see the reduced ratio of meta-label-specific features to the original feature size on Genbase.

Next we compared the classification accuracies of five MLC methods. The experimental results are shown in Table II where the averaged performance order over all datasets is shown in the last row of each metric. Among the five comparing

[2]http://lamda.nju.edu.cn/files/mddm.rar
[3]http://cse.seu.edu.cn/PersonalPage/zhangml/files/LIFT.rar
[4]https://github.com/JunHuangUCAS/LLSF.git

TABLE II
EXPERIMENTAL RESULTS (MEAN±STD.) ON TWELVE MULTI-LABEL DATASETS IN FOUR EVALUATION METRICS.

| Method | Emotions | Scene | Yeast | Genbase | Medical | Enron | Rcv1s1 | Rcv1s2 | Mediamill | Bibtex | Corel16k1 | Corel16k2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Exact-Match** | | | | | | | | | | | | |
| BR | .285±.015 | .533±.013 | .148±.009 | **.982±.005** | .665±.008 | .111±.009 | .071±.004 | .172±.003 | .066±.004 | .143±.004 | .006±.001 | .004±.001 |
| MDDM | .263±.013 | .529±.007 | .137±.009 | .980±.005 | .609±.014 | .121±.006 | .065±.002 | .174±.006 | .068±.004 | .143±.003 | .000±.000 | .001±.000 |
| LIFT | .184±.014 | **.637±.010** | .154±.008 | .953±.012 | .574±.010 | .119±.004 | .059±.003 | .145±.003 | .069±.004 | .139±.003 | .000±.000 | .000±.000 |
| LLSF | .285±.015 | .531±.014 | .148±.009 | **.982±.005** | .662±.008 | .111±.009 | .072±.004 | .172±.003 | .066±.004 | **.144±.004** | .006±.001 | .004±.000 |
| MLSF | **.315±.016** | **.637±.007** | **.212±.012** | .982±.005 | **.689±.009** | **.122±.009** | **.128±.004** | **.214±.006** | **.070±.003** | .143±.002 | **.008±.001** | **.007±.003** |
| Rank | MLSF ≻ LLSF ≻ BR ≻ MDDM ≻ LIFT | | | | | | | | | | | |
| **Hamming-Score** | | | | | | | | | | | | |
| BR | **.805±.007** | .895±.003 | .801±.004 | **.999±.000** | **.990±.000** | .940±.001 | .965±.000 | .969±.000 | .968±.000 | .984±.000 | .980±.000 | .981±.000 |
| MDDM | .788±.004 | .899±.001 | .798±.004 | **.999±.000** | .988±.000 | .953±.001 | .973±.000 | .974±.000 | **.969±.000** | **.988±.000** | **.981±.000** | **.983±.000** |
| LIFT | .755±.005 | **.919±.002** | **.804±.003** | .998±.000 | .987±.000 | **.955±.000** | **.974±.000** | **.977±.000** | **.969±.000** | **.988±.000** | **.981±.000** | **.983±.000** |
| LLSF | **.805±.007** | .895±.003 | .801±.004 | **.999±.000** | **.990±.000** | .940±.001 | .965±.000 | .969±.000 | .968±.000 | .984±.000 | .980±.000 | .981±.000 |
| MLSF | .793±.016 | .891±.002 | .789±.004 | **.999±.000** | **.990±.000** | .940±.001 | .966±.000 | .970±.000 | .968±.000 | .984±.000 | .980±.000 | .981±.003 |
| Rank | LIFT ≻ MDDM ≻ BR ≽ LLSF ≻ MLSF | | | | | | | | | | | |
| **Macro-F1** | | | | | | | | | | | | |
| BR | .633±.013 | .694±.008 | .322±.005 | .761±.020 | .366±.011 | **.222±.005** | .250±.008 | .235±.004 | .028±.000 | .328±.003 | **.047±.001** | **.051±.004** |
| MDDM | .583±.017 | .684±.005 | .318±.005 | .754±.017 | .323±.008 | .201±.009 | .156±.005 | .217±.005 | **.035±.001** | .159±.001 | .008±.001 | .012±.001 |
| LIFT | .496±.010 | **.759±.005** | .319±.005 | .704±.020 | .240±.009 | .136±.005 | .134±.005 | .096±.002 | **.035±.000** | .145±.001 | .003±.001 | .004±.000 |
| LLSF | .633±.013 | .693±.009 | .322±.005 | **.769±.016** | .370±.016 | **.222±.005** | .246±.007 | .236±.004 | .028±.000 | **.329±.001** | .045±.001 | .049±.003 |
| MLSF | **.657±.016** | .699±.007 | **.346±.009** | **.769±.016** | **.387±.012** | .221±.005 | **.255±.007** | **.240±.005** | .029±.003 | .328±.002 | **.047±.002** | **.051±.003** |
| Rank | MLSF ≻ BR ≽ LLSF ≻ MDDM ≻ LIFT | | | | | | | | | | | |
| **Micro-F1** | | | | | | | | | | | | |
| BR | .661±.014 | .688±.009 | .631±.008 | **.993±.002** | .810±.004 | .515±.005 | .399±.004 | .413±.005 | .510±.002 | .422±.002 | **.072±.002** | **.079±.003** |
| MDDM | .627±.010 | .682±.005 | .627±.009 | .992±.002 | .780±.007 | **.579±.006** | .356±.005 | .395±.003 | **.528±.002** | .364±.004 | .007±.001 | .016±.001 |
| LIFT | .557±.011 | **.755±.007** | .632±.007 | .980±.005 | .679±.005 | .570±.003 | .345±.004 | .327±.007 | .519±.002 | .338±.002 | .005±.001 | .012±.001 |
| LLSF | .661±.014 | .686±.010 | .631±.008 | **.993±.002** | .804±.005 | .515±.005 | .396±.003 | .412±.005 | .510±.002 | **.423±.001** | .069±.002 | **.079±.002** |
| MLSF | **.665±.016** | .692±.005 | **.639±.008** | **.993±.002** | **.815±.004** | .515±.004 | **.407±.004** | **.419±.004** | .491±.011 | **.423±.001** | .070±.002 | .076±.003 |
| Rank | MLSF ≻ BR ≻ LLSF ≻ MDDM ≻ LIFT | | | | | | | | | | | |

methods, the best performance is highlighted in boldface. MLSF outperformed the other methods on the average in three metrics of Exact-Match, Macro/Micro-F1. It demonstrates the effectiveness of MLSF and verifies the existence of meta-labels with specific features. Nevertheless MLSF worked the worst in term of Hamming-Score. It is probably because MLSF tends to optimize Exact-Match by learning meta-labels, which would harm the performance in Hamming-Score [20]. LIFT ranked at the first in Hamming-Score, and was even better than the Hamming-Score optimizer, BR, showing the success of local FS-DR strategy. However, LIFT worked the worst in other three metrics. The unsatisfactory performance of LIFT possibly shows that the extracting way of label-specific features is not enough for handling MLC problems. By taking label correlations into account for local FS-DR, LLSF ranked at the second in Exact-Match and worked better than LIFT, except in Hamming-Score, showing the importance of modeling label correlations. As the global FS-DR method, MDDM performed worse than the baseline BR, except in Hamming-Score. It seems that projecting features into the identical subspace possibly weakens the discriminative ability for some labels. On the other hand, as the simplest MLC method, BR provided competitive classification performance in comparison with FS-DR methods, especially in large-scale datasets. It is probably because large-scale datasets typically need a sufficient number of instances for training, while FS-DR tends to remove too many features.

TABLE III
EXECUTION TIME ($10^2$ SEC) ON LARGE-SCALE DATASETS.

| | Rcv1s1 | Rcv1s2 | Mediamill | Bibtex | Corel16k1 | Corel16k2 |
|---|---|---|---|---|---|---|
| BR | 1.531 | 1.441 | **0.624** | 4.796 | 1.324 | 1.367 |
| MDDM | **0.701** | 1.783 | 1.698 | **1.893** | **0.478** | **0.467** |
| LIFT | 7.279 | 7.053 | 2.345 | 21.258 | 10.214 | 10.209 |
| LLSF | 1.568 | 1.481 | 0.644 | 4.970 | 1.343 | 1.384 |
| MLSF | 1.388 | **1.371** | 0.914 | 4.705 | 1.219 | 1.243 |
| Rank | MDDM ≻ MLSF ≻ BR ≻ LLSF ≻ LIFT | | | | | |

Table III reports the mean execution time on six large-scale datasets. The least time cost is highlighted in boldface. Among all five methods, MDDM needed the least time on the average and was remarkably faster on the three largest datasets. Such an advantage benefits from the global FS-DR strategy, leading to significant reduction of the training cost. MLSF consumed less time than the linear method BR, except on Mediamill, showing efficiency of mining meta-label with specific features. The other local FS-DR methods, LLSF and LIFT, cost consistently more time than BR. It seems that mining label-specific features wastes too much time. In summary, MLSF is one of the best choices in the balance of performance and execution time.

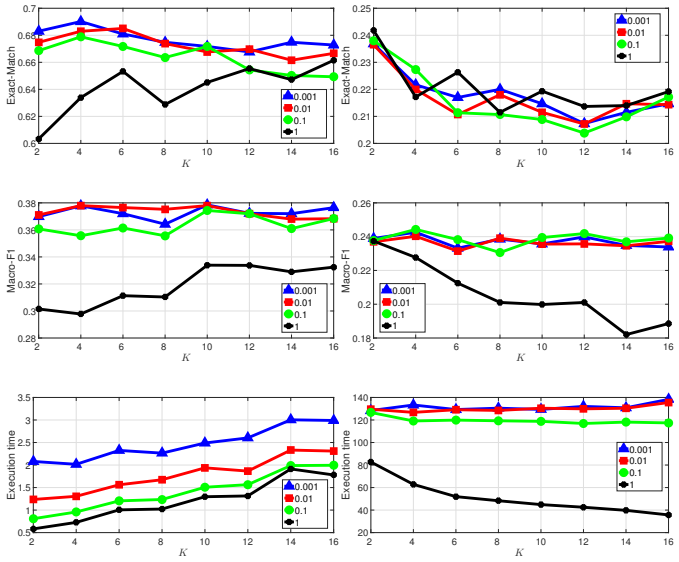



Fig. 4. Parameter sensitivity analysis over the number $K$ of meta-labels and the sparsity factor $\gamma$ in the datasets Medical (the left column) and Rcv1s2 (the right column). The value of $K$ was varied from 2 to 16 by step 2 and $\gamma \in \{0.001, 0.01, 0.1, 1\}$.

A parameter sensitivity analysis on the number $K$ of meta-labels and the sparsity factor $\gamma$ was performed on Medical and Rcv1s2. The results are shown in Fig. 4, where the value of $K$ was varied from 2 to 16 by step 2 and $\gamma \in \{0.001, 0.01, 0.1, 1\}$. The values of Exact-match (upper), Macro-F1 (middle) and Execution time (bottom) were averaged by *5-fold cross validation*. We observe that as the value of $\gamma$ increased, the classification performance degraded, although its execution time was consistently reduced. In contrast, as the value of $K$ increased, the performance degraded in Exact-Match, and kept nearly constant in Macro-F1, while the time increased. Hence, by considering the balance between accuracy and time, in the experiments we set the values of $K$ and $\gamma$ to $\lceil L/10 \rceil$ and $0.01$ in Sec. IV-B. Note that Exact-Match is the most sensitive metric to the existence of meta-labels. If label correlations are well modeled as meta-labels by MLSF, the larger value of $K$ would degrade the metric. In this sense, the fast performance degradation in Exact-Match indicates the existence of meta-labels in Rcv1s2.

## V. Conclusion

In this paper, a novel MLC method with Meta-Label-Specific Features (MLSF) has been proposed, on expecting the existence of meta-labels with specific features. MLSF consists of meta-label learning and specific features mining, and captures label correlations by meta-labels and selects discriminative features in each meta-label. MLSF has a reasonable possibility to overcome the limitations of previous global/local FS-DR methods. Experiments performed on twelve multi-label datasets demonstrated the advantage of MLSF over the state-of-the-art methods.

## Acknowledgment

The authors thank the reviewers for their helpful comments. This work was partially supported by JSPS KAKENHI Grant Number 15H02719 and China Scholarship Council.

## References

[1] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, pp. 333–359, 2011.

[2] G. Tsoumakas, I. Katakis, and L. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.

[3] K. Yu, S. Yu, and V. Tresp, "Multi-label informed latent semantic indexing," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '05, 2005, pp. 258–265.

[4] L. Sun, S. Ji, and J. Ye, "Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 194–200, 2011.

[5] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10. ACM, 2010, pp. 999–1008.

[6] L. Sun and M. Kudo, "Polytree-augmented classifier chains for multi-label classification," in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 3834–3840.

[7] M. Zhang and L. Wu, "Lift: multi-label learning with label-specific features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 107–120, 2015.

[8] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label specific features for multi-label classification," in *2015 IEEE International Conference on Data Mining, 2015*, 2015, pp. 181–190.

[9] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[11] Y. Zhang and Z. Zhou, "Multi-label dimensionality reduction via dependence maximization," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 2008, pp. 1503–1505.

[12] H. Wang, C. Ding, and H. Huang, "Multi-label linear discriminant analysis," in *Proceedings of the 11th European Conference on Computer Vision*, 2010, vol. 6316, pp. 126–139.

[13] L. Sun, S. Ji, and J. Ye, "Hypergraph spectral learning for multi-label classification," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 668–676.

[14] R. Tibshirani, "Regression shrinkage and selection via the lasso: a retrospective," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 3, pp. 273–282, 2011.

[15] J. Fürnkranz, E. Hüllermeier, E. Mencia, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[16] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A java library for multi-label learning," *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

[17] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[18] X. He and P. Niyogi, "Locality preserving projections," in *In Advances in Neural Information Processing Systems 16*. MIT Press, 2003.

[19] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[20] K. Dembczyski, W. Waegeman, W. Cheng, and E. Hllermeier, "On label dependence and loss minimization in multi-label classification," *Machine Learning*, vol. 88, no. 1-2, pp. 5–45, 2012.