BiLabel-Specific Features for Multi-Label Classification

MIN-LING ZHANG, JUN-PENG FANG AND YI-BO WANG, Southeast University, China

In multi-label classification, the task is to induce predictive models which can assign *a set of* relevant labels for the unseen instance. The strategy of label-specific features has been widely employed in learning from multi-label examples, where the classification model for predicting the relevancy of each class label is induced based on its *tailored* features rather than the original features. Existing approaches work by generating a group of tailored features for each class label independently, where label correlations are not fully considered in the label-specific features generation process. In this paper, we extend existing strategy by proposing a simple yet effective approach based on BiLabel-specific features. Specifically, a group of tailored features are generated for a pair of class labels with heuristic prototype selection and embedding. Thereafter, predictions of classifiers induced by BiLabel-specific features are ensembled to determine the relevancy of each class label for unseen instance. To thoroughly evaluate the BiLabel-specific features strategy, extensive experiments are conducted over a total of thirty-five benchmark data sets. Comparative studies against state-of-the-art label-specific features techniques clearly validate the superiority of utilizing BiLabel-specific features to yield stronger generalization performance for multi-label classification.

CCS Concepts: • **Computing methodologies** → **Supervised learning**; **Machine learning algorithms**.

Additional Key Words and Phrases: Multi-label classification, label-specific features, label correlations, pairwise comparison

ACM Reference Format:
Min-Ling Zhang, Jun-Peng Fang and Yi-Bo Wang. 2020. BiLabel-Specific Features for Multi-Label Classification. *ACM Trans. Knowl. Discov. Data.* 1, 1 (March 2020), 21 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Multi-label classification aims to build predictive models capable of assigning *a set of* relevant labels for the unseen instance, which has been extensively applied to learn from real-world objects with rich semantics [14, 53, 55]. In recent years, the strategy of *label-specific features* has been widely utilized as an effective solution to learn from multi-label examples [10, 16–18, 24, 28, 41–43, 48, 52, 54]. Here, the classification model for predicting the relevancy of each class label is induced based on its *tailored* features rather than the original features. For instance, in text categorization, features corresponding to word terms *GDP*, *tax* and *housing price* are informative in discriminating economic and non-economic documents, while features corresponding to word terms *league*, *match* and *championship* are informative in discriminating sports and non-sports documents. In this way, label-specific features are expected to encode inherent and distinct characteristics of each class label so as to facilitate the model induction process.

Author's address: Min-Ling Zhang, Jun-Peng Fang and Yi-Bo Wang, School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, {zhangml,fangjp,wang_yb}@seu.edu.cn.

Generally, existing approaches based on label-specific features work in a label-by-label manner such that the group of features specific to each class label are generated independently. Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ denote the original feature space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ denote the label space consisting of q possible class labels. Accordingly, one feature mapping function $\varphi_u : \mathcal{X} \mapsto \mathcal{Z}_u$ ($\mathcal{Z}_u = \mathbb{R}^{d_u}$) is introduced where a group of d_u label-specific features $\varphi_u(\boldsymbol{x}) \in \mathbb{R}^{d_u}$ w.r.t. the u-th class label $y_u \in \mathcal{Y}$ ($1 \leq u \leq q$) can be generated for instance $\boldsymbol{x} \in \mathbb{R}^d$ in the original feature space. Nonetheless, this label-independence practice may lead to suboptimal performance due to the ignorance of label correlations exploitation. Therefore, in order to make use of label correlations, the strategy of label-specific features needs to be extended by generating tailored correlation-ware features for model induction. For instance, features related to word terms *spacecraft*, *exploration* and *interstellar* would be informative in inducing classifiers for documents with *science fiction* and *adventure* genres, while features related to word terms such as *conspiracy*, *redemption* and *Holmes* would be informative in inducing classifiers for documents with *crime* and *thriller* genres.

In light of these observations, we propose to extending the existing strategy for label-specific features from *uni-label* mode to *bi-label* mode. Accordingly, a novel approach named BiLAS, i.e. *BiLAble-Specific features for multi-label classification*, is proposed in this paper. For each pair of class labels, a group of BiLabel-specific features are generated via prototype embedding based on a heuristic prototype selection procedure over training examples with distinct bi-label assignment. Thereafter, an ensemble of binary classifiers are induced based on the BiLabel-specific features whose predictions are aggregated with empirical weighted voting. Extensive experiments over a total of thirty-five benchmark data sets show that the proposed approach based on BiLabel-specific features significantly outperforms state-of-the-art label-specific features techniques for multi-label classification.

The rest of this paper is organized as follows. Section 2 reviews existing works related to BiLAS. Section 3 gives technical details of the proposed BiLAS approach. Section 4 reports detailed experimental results. Finally, Section 5 concludes this paper.

2 RELATED WORKS

Multi-label classification is related to several well-established classification frameworks such as *multi-class classification*, *multi-instance classification*, *multi-instance multi-label learning*, and *multi-dimensional classification*. In traditional multi-class classification, each training example is assumed to be associated with a single class label which can be regarded as a degenerated version of multi-label classification by restricting the size of label set to be one. In multi-instance classification [1, 7, 25], a single class label is assigned to a bag of instances while for multi-label classification a number of class labels are assigned to a single instance. In multi-instance multi-label learning (MIML) [12, 29, 57], each training example is represented by a bag of instances and associated with a number of class labels which encompasses multi-label classification as one of its degenerated versions with single instance representation. In multi-dimensional classification [22, 23, 33], each training example is assumed to be associated with class labels from heterogenous class spaces while for multi-label classification all the class labels are assumed to belong to one homogeneous class space.

One major challenge for multi-label classification lies in the huge number of label sets to be predicted for the classification model, which is exponential to the number of possible class labels in the label space. Therefore, exploitation of label correlations serves as the key factor to enable effective multi-label classification which has been extensively studied in recent years [14, 55]. Generally, the type of label correlations considered by learning algorithms could be *first-order*, such as treating each class label in an independent manner [4, 53], *second-order* such as exploiting pairwise relationships between class labels [5, 13, 21, 26, 32, 58], or *high-order* such as exploiting interactions among a subset of or all class labels [20, 34, 39, 47].

Other than label correlations exploitation in the output space, another effective strategy to facilitate multi-label classification is to manipulate feature representation in the input space. The most straightforward strategy is to perform dimensionality reduction [36] or feature selection [30] over the original set of features. There are some other feature manipulation strategies such as generating meta-level features with strong discriminative information from the original feature representation [6, 49], learning distance metric from multi-label examples [15, 27, 38], making use of multi-view representation available for multi-label data [44, 45, 51], learning feature representation in an end-to-end manner [9, 40, 46, 50]. For these feature manipulation techniques, the common practice of utilizing the identical feature representation for the discrimination of all class labels has been adopted.

In recent years, the strategy of label-specific features has attracted significant attentions as an alternative feature manipulation solution to multi-label classification. Rather than relying on the identical feature representation, tailored feature representation for each class label is employed in its discrimination process. Earlier approach towards label-specific features works by generating tailored feature representation in transformed feature space [54]. Specifically, clustering analysis is performed over the set of positive and negative training instances w.r.t. each class label where the resulting clustering centers are utilized as the embedding basis for the transformed feature space. The generation process for label-specific features can be further enhanced by employing customized strategies, such as removing redundant information in generated label-specific features with attribute reduction [48], imposing structured sparsity regularization over the generation process of label-specific features [10, 18], expanding label-specific features by exploiting nearest neighbor rule to achieve stronger discrimination ability [43], applying linear discriminant analysis to excavate most informative label-specific features [16], performing label completion over label matrix with missing labeling information to facilitate label-specific features generation [41], and enriching feature representation by extracting label-specific features at various levels of granularity [28], etc.

In addition to generate label-specific features in a transformed feature space, it is also feasible to work by retaining specific subset of original features for different class labels. Specifically, the process of retaining specific subset of original features can be instantiated as by conducting feature selection for each class label, such as identifying features with strong discriminative ability by taking sparse [17] or non-sparse [42] assumption over the selected subset of features, retaining a subset of features for meta-labels by invoking LASSO and spectral clustering techniques [37], and learning to assign weight vector over the original features via linear regression [24] or regularized optimization [52] for each class label, etc.

3 THE PROPOSED APPROACH

Given the multi-label training set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq m\}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the d-dimensional feature vector and $Y_i \subseteq \mathcal{Y}$ is the set of relevant labels associated with $\boldsymbol{x}_i$, the task of multi-label classification is to induce a predictive model $h : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ from $\mathcal{D}$ which can assign a set of proper labels for unseen instance. For the proposed BiLAS approach, the learning task is fulfilled in two stages including *BiLabel-specific features generation* and *predictive model induction*.

3.1 BiLabel-Specific Features Generation

For BiLAS, a group of tailored features would be generated for each pair of class labels y_u and y_v ($1 \leq u < v \leq q$) based on prototype embedding. Specifically, a set of candidate prototypes $\mathcal{P}_{uv} \bigcup \mathcal{N}_{uv}$ are formed for label pair (y_u, y_v)

as follows:

$$\mathcal{P}_{uv} = \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, Y_i) \in \mathcal{D},\ y_u \in Y_i,\ y_v \notin Y_i\} \tag{1}$$
$$\mathcal{N}_{uv} = \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, Y_i) \in \mathcal{D},\ y_u \notin Y_i,\ y_v \in Y_i\}$$

In other words, $\mathcal{P}_{uv}$ and $\mathcal{N}_{uv}$ consist of training examples with distinct label assignments w.r.t. y_u and y_v. Conceptually, the candidate prototypes in $\mathcal{P}_{uv} \bigcup \mathcal{N}_{uv}$ are informative in characterizing specific label correlations between y_u and y_v. Rather than utilizing all the candidate prototypes for feature embedding, BiLAS chooses to select representative prototypes for subsequent BiLabel-specific features generation. For each candidate prototype $\boldsymbol{\mu}_k \in \mathcal{P}_{uv} \bigcup \mathcal{N}_{uv}$ ($1 \le k \le |\mathcal{P}_{uv}| + |\mathcal{N}_{uv}|$), the heuristic prototype selection procedure considers its *separability* as well as *dispersion* for the embedded feature. Accordingly, let $P_k \bigcup N_k$ be the set of values for the embedded feature w.r.t. $\boldsymbol{\mu}_k$:

$$P_k = \{d(\boldsymbol{x}, \boldsymbol{\mu}_k) \mid \boldsymbol{x} \in \mathcal{P}_{uv}\} \tag{2}$$
$$N_k = \{d(\boldsymbol{x}, \boldsymbol{\mu}_k) \mid \boldsymbol{x} \in \mathcal{N}_{uv}\}$$

Here, $d(\cdot, \cdot)$ corresponds to the distance between two instances which is set to be Euclidean distance in this paper.

Then, a heuristic ranking score s_k can be defined for $\boldsymbol{\mu}_k$ as follows:

$$s_k = t \cdot sep_k + (1-t) \cdot dis_k \quad \text{where} \tag{3}$$
$$sep_k = \frac{p_k + n_k}{pn_k}, \quad dis_k = 1/\sqrt{var(P_k \bigcup N_k)}$$
$$p_k = \frac{1}{|\mathcal{P}_{uv}|^2} \sum_{a \in P_k} \sum_{a' \in P_k} |a - a'|$$
$$n_k = \frac{1}{|\mathcal{N}_{uv}|^2} \sum_{b \in N_k} \sum_{b' \in N_k} |b - b'|$$
$$pn_k = \frac{1}{|\mathcal{P}_{uv}| \cdot |\mathcal{N}_{uv}|} \sum_{a \in P_k} \sum_{b \in N_k} |a - b|$$

Generally, we prefer candidate prototype whose embedded feature has good separability as measured by *high coherence* within P_k and N_k and *low coupling* across P_k and N_k. Furthermore, we prefer candidate prototype whose embedded feature has good dispersion as measured by the standard deviation of the feature values in $P_k \bigcup N_k$. Here, $t \in [0, 1]$ serves as the balancing parameter between separability and dispersion and the candidate prototype with smaller value of s_k is preferable.

Accordingly, BiLAS selects a set of m_{uv} candidate prototypes $\overline{\mathcal{P}}_{uv} \subseteq \mathcal{P}_{uv}$ from $\mathcal{P}_{uv}$ which has least ranking scores s_k ($1 \le k \le |\mathcal{P}_{uv}|$). Similarly, another set of m_{uv} candidate prototypes $\overline{\mathcal{N}}_{uv} \subseteq \mathcal{N}_{uv}$ with least ranking scores s_k ($|\mathcal{P}_{uv}| + 1 \le k \le |\mathcal{P}_{uv}| + |\mathcal{N}_{uv}|$) are selected from $\mathcal{N}_{uv}$. Here, the same number (i.e. m_{uv}) of candidate prototypes are selected from both $\mathcal{P}_{uv}$ and $\mathcal{N}_{uv}$ to account for their equal importance in feature embedding. Specifically, the value of m_{uv} is set as:

$$m_{uv} = \lceil r \cdot \min(|\mathcal{P}_{uv}|, |\mathcal{N}_{uv}|) \rceil \tag{4}$$

with ratio parameter $r \in (0, 1]$.

Without loss of generality, let $\overline{\mathcal{P}}_{uv} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_{m_{uv}}\}$ and $\overline{\mathcal{N}}_{uv} = \{\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \boldsymbol{n}_{m_{uv}}\}$ where each of them consists of m_{uv} selected prototypes. Then, the BiLabel-specific features for an instance $\boldsymbol{x} \in \mathcal{X}$ can be generated based on the following mapping function $\varphi_{uv} : \mathbb{R}^d \mapsto \mathbb{R}^{2m_{uv}}$:

$$\varphi_{uv}(\boldsymbol{x}) = [d(\boldsymbol{x}, \boldsymbol{p}_1), \ldots, d(\boldsymbol{x}, \boldsymbol{p}_{m_{uv}}), d(\boldsymbol{x}, \boldsymbol{n}_1), \ldots, d(\boldsymbol{x}, \boldsymbol{n}_{m_{uv}})] \tag{5}$$

3.2 Predictive Model Induction

Following the well-established strategy of pairwise comparison for multi-label classification [13, 14, 55], one binary training set $\overline{\mathcal{D}}_{uv}$ for label pair (y_u, y_v) is instantiated as follows:

$$\overline{\mathcal{D}}_{uv} = \{(\varphi_{uv}(\boldsymbol{x}_i), +1) \mid \boldsymbol{x}_i \in \mathcal{P}_{uv}\} \bigcup \{(\varphi_{uv}(\boldsymbol{x}_i), -1) \mid \boldsymbol{x}_i \in \mathcal{N}_{uv}\} \tag{6}$$

Thereafter, one binary classifier $g_{uv} : \mathbb{R}^{2m_{uv}} \mapsto \mathbb{R}$ can be induced by invoking some binary training algorithm $\mathcal{B}$ on $\overline{\mathcal{D}}_{uv}$: $g_{uv} \leftarrow\!\shortmid \mathcal{B}\left(\overline{\mathcal{D}}_{uv}\right)$.

In addition, one virtual label y_V is introduced to serve as a natural splitting point between relevant and irrelevant labels. Correspondingly, we can also instantiate one binary training set $\overline{\mathcal{D}}_{uV}$ $(1 \le u \le q)$ for label pair (y_u, y_V) by setting $\mathcal{P}_{uV}$ and $\mathcal{N}_{uV}$ as follows:

$$\begin{aligned} \mathcal{P}_{uV} &= \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, Y_i) \in \mathcal{D},\ y_u \in Y_i\} \\ \mathcal{N}_{uV} &= \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, Y_i) \in \mathcal{D},\ y_u \notin Y_i\} \end{aligned} \tag{7}$$

Similarly, one binary classifier g_{uV} can be induced by invoking $\mathcal{B}$ on $\overline{\mathcal{D}}_{uV}$: $g_{uV} \leftarrow\!\shortmid \mathcal{B}\left(\overline{\mathcal{D}}_{uV}\right)$.

Accordingly, let ϑ_{uv} (ϑ_{uV}) denote the *empirical accuracy* of g_{uv} (g_{uV}) in classifying examples in $\overline{\mathcal{D}}_{uv}$ $(\overline{\mathcal{D}}_{uV})$. Given the unseen instance $\boldsymbol{x}^*$, its modeling outputs $\Gamma(\boldsymbol{x}^*, y_u)$ on each class label y_u $(1 \le u \le q)$ are determined by aggregating predictions from q binary classifiers coupled with y_u via empirical weighted voting:

$$\Gamma(\boldsymbol{x}^*, y_u) = \sum_{l=1}^{u-1} \vartheta_{lu} \cdot [\![g_{lu}\left(\varphi_{lu}(\boldsymbol{x}^*)\right) \le 0]\!] + \sum_{l=u+1}^{q} \vartheta_{ul} \cdot [\![g_{ul}\left(\varphi_{ul}(\boldsymbol{x}^*)\right) > 0]\!] + \vartheta_{uV} \cdot [\![g_{uV}\left(\varphi_{uV}(\boldsymbol{x}^*)\right) > 0]\!] \tag{8}$$

Note that g_{lu} (g_{ul}) represents the binary classifier which treats y_u as the negative (positive) class. Furthermore, $[\![\pi]\!]$ represents the zero-one indicator operator which returns 1 if predicate π holds. Otherwise, $[\![\pi]\!]$ returns 0. Similarly, the modeling output $\Gamma(\boldsymbol{x}^*, y_V)$ on the virtual label y_V is determined as follows:

$$\Gamma(\boldsymbol{x}^*, y_V) = \sum_{u=1}^{q} \vartheta_{uV} \cdot [\![g_{uV}\left(\varphi_{uV}(\boldsymbol{x}^*)\right) \le 0]\!] \tag{9}$$

Thereafter, the set of relevant labels $h(\boldsymbol{x}^*) \subseteq \mathcal{Y}$ for the unseen instance $\boldsymbol{x}^*$ are identified by thresholding $\Gamma(\boldsymbol{x}^*, y_u)$ over $\Gamma(\boldsymbol{x}^*, y_V)$:

$$h(\boldsymbol{x}^*) = \{y_u \mid \Gamma(\boldsymbol{x}^*, y_u) > \Gamma(\boldsymbol{x}^*, y_V),\ 1 \le u \le q\} \tag{10}$$

Table 1 summarizes the complete procedure of BILAS.[1] To generate BiLabel-specific features for each pair of class labels, a heuristic prototype selection and embedding strategy is utilized by analyzing separability and dispersion properties of the features embedded by prototypes (Steps 1-8). Based on the binary training sets derived from BiLabel-specific features, a total of $\binom{q}{2} + q$ binary classifiers are induced for each pair of class labels (Steps 9-14) as well as the virtual label (Steps 15-19). Finally, the set of relevant labels for the unseen instance are obtained by aggregating predictions of all binary classifiers with empirical weighted voting (Steps 20-21).

Remarks As a first attempt towards BiLabel-specific features generation, it is worth noting that BILAS exhibits the following relationships and differences with related works on multi-label classification:

- As discussed in Section 2, second-order approaches towards multi-label classification work by considering pairwise relationships between class labels, such as the ranking between relevant label and irrelevant label [5, 13, 26],

[1] Code package of BILAS is publicly available at: http://palm.seu.edu.cn/zhangml/files/BILAS.zip

Table 1. The pseudo-code of BILAS.

Inputs:

$\mathcal{D}$: the multi-label training set $\{(\boldsymbol{x}_i, Y_i) \mid 1 \le i \le m\}$ ($\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{y_1, y_2, \ldots, y_q\}, \boldsymbol{x}_i \in \mathcal{X}, Y_i \subseteq \mathcal{Y}$)

t, r: the balancing parameter $t \in [0, 1]$ in Eq.(3) and the ratio paramter $r \in (0, 1]$ in Eq.(4)

$\mathcal{B}$: the binary training algorithm

$\boldsymbol{x}^*$: the unseen instance

Outputs:

Y^*: the predicted label set for $\boldsymbol{x}^*$

Process:

for $u = 1$ to $q - 1$ **do**
  for $v = u + 1$ to q **do**
    Identify the set of candidate prototypes $\mathcal{P}_{uv}$ and $\mathcal{N}_{uv}$ according to Eq.(1);
    Calculate the heuristic ranking score s_k for each candidate prototype $\boldsymbol{\mu}_k \in \mathcal{P}_{uv} \bigcup \mathcal{N}_{uv}$ according to Eq.(3);
    Form $\overline{\mathcal{P}}_{uv} \subseteq \mathcal{P}_{uv}$ by selecting m_{uv} prototypes from $\mathcal{P}_{uv}$ with least ranking scores s_k $(1 \le k \le |\mathcal{P}_{uv}|)$;
    Form $\overline{\mathcal{N}}_{uv} \subseteq \mathcal{N}_{uv}$ by selecting m_{uv} prototypes from $\mathcal{N}_{uv}$ with least ranking scores s_k $(|\mathcal{P}_{uv}| + 1 \le k \le |\mathcal{P}_{uv}| + |\mathcal{N}_{uv}|)$;
  end for
end for
for $u = 1$ to $q - 1$ **do**
  for $v = u + 1$ to q **do**
    Instantiate binary training set $\overline{\mathcal{D}}_{uv}$ according to Eqs.(5)-(6);
    Induce binary classifier $g_{uv} \leftarrow\!\shortmid \mathcal{B}\left(\overline{\mathcal{D}}_{uv}\right)$ and obtain the empirical accuracy ϑ_{uv} of g_{uv} in classifying examples in $\overline{\mathcal{D}}_{uv}$;
  end for
end for
for $u = 1$ to q **do**
  Identify the set of candidate prototypes $\mathcal{P}_{uV}$ and $\mathcal{N}_{uV}$ according to Eq.(7);
  Instantiate binary training set $\overline{\mathcal{D}}_{uV}$ according to Eqs.(2)-(6);
  Induce binary classifier $g_{uV} \leftarrow\!\shortmid \mathcal{B}\left(\overline{\mathcal{D}}_{uV}\right)$ and obtain the empirical accuracy ϑ_{uV} of g_{uV} in classifying examples in $\overline{\mathcal{D}}_{uV}$;
end for
Obtain the modeling output $\Gamma(\boldsymbol{x}^*, y_u)$ $(1 \le u \le q)$ and $\Gamma(\boldsymbol{x}^*, y_V)$ according to Eq.(8) and Eq.(9) respectively;
Return $Y^* = h(\boldsymbol{x}^*)$ according to Eq.(10).

interaction between a pair of labels [32, 58], etc. Nonetheless, existing second-order approaches employ identical feature representation while exploiting pairwise label correlations. For BILAS, a group of tailored features w.r.t. a pair of class labels instead of identical feature representation are utilized to induce the multi-label classification model.

- The strategy of *virtual label* has been widely used in multi-label classification to help bipartition the ranked class labels into relevant and irrelevant ones [13, 14, 51]. In view of the real-valued modeling output over each

class label (Eq.(8)), BILAS also introduces virtual label (Eq.(9)) to enable multi-label prediction by comparing the real-valued modeling output over each class label and the virtual label.

- By either generating label-specific features in transformed feature space or retaining specific feature subset in original feature space, existing approaches fulfill label-specific features generation in *uni-label* mode by considering tailored features for individual class label. On the other hand, BILAS fulfills the generation process in *bi-label* mode by considering tailored features for a pair of class labels.
- For existing approaches, the embedding prototypes used for feature transformation correspond to the clustering centers identified via clustering analysis [10, 16, 43, 48] or the projection vectors learned via linear regression [18, 28, 41]. However, the embedding prototypes used by BILAS correspond to training examples selected via heuristic criteria measured by separability (high coherence & low coupling) and dispersion (Eq.(3)).
- For existing approaches, the candidate features used for feature selection correspond to the raw ones without feature mapping. On the other hand, the BiLabel-specific features employed by BILAS are the transformed ones mapped from selected embedding prototypes.

4 EXPERIMENTS

4.1 Experimental Setup

To show the effectiveness of the proposed BiLabel-specific features strategy, a total of thirty-five benchmark multi-label data sets have been employed for experimental studies.[2] Given a multi-label data set $\mathcal{S}$, let $|\mathcal{S}|$, $dim(\mathcal{S})$, $L(\mathcal{S})$ and $LCard(\mathcal{S})$ denote the *number of examples*, *number of features*, *number of class labels* and *label cardinality* (i.e. average number of relevant labels per example). Correspondingly, Table 2 summarizes detailed characteristics of the thirty-five experimental data sets. As shown in Table 2, the employed data sets exhibit diversified properties in terms of the summarized statistics $|\mathcal{S}|$, $dim(\mathcal{S})$, $L(\mathcal{S})$ and $LCard(\mathcal{S})$.[3] To the best of our knowledge, the experimental data sets employed in this paper are most comprehensive for label-specific features studies [10, 16–18, 24, 28, 41–43, 48, 52, 54], which serve as a rather solid testbed for performance evaluation.

For comparative studies, the first series include state-of-the-art multi-label classification approaches which work by exploiting label correlations based on original feature representation: a) CLR: A representative second-order approach which learns from multi-label data based on pairwise comparison [13]; b) LPLC: A second-order approach which integrates pairwise label correlation exploitation with kNN-based posteriori probability maximization [19]; c) MLFE: A high-order approach which facilitates label correlation exploitation by utilizing manifold structure information encoded in feature space [56].

The second series include state-of-the-art multi-label classification approaches which work by employing the label-specific features strategy: d) LIFT: The seminal label-specific features approach for multi-label classification which generates tailored features for each class label based on clustering analysis [54]; e) LLSF: Another comparing approach based on label-specific features which generates tailored features by retaining different subset of features for each class label [17]; f) MLSF: The third comparing approach based on label-specific features which generates tailored features by performing sparse regression over partitioned label space [37].

[2]Data sets publicly available at http://mulan.sourceforge.net/datasets-mlc.html, http://www.kecl.ntt.co.jp/as/members/ueda/yahoo.tar.gz and https://rdrr.io/cran/mldr.datasets/

[3]To help enhance the *multi-labelledness* property, several benchmark data sets with low label cardinality (i.e. $L(\mathcal{S})$) are preprocessed by randomly excluding examples associated with only one relevant label. Thereafter, the label cardinality of most experimental data sets in Table 2 (30 out of 35) is greater than 2.0.

Table 2. Characteristics of the experimental data sets.

Data set	$\|\mathcal{S}\|$	$dim(\mathcal{S})$	$L(\mathcal{S})$	$LCard(\mathcal{S})$	Domain	Data set	$\|\mathcal{S}\|$	$dim(\mathcal{S})$	$L(\mathcal{S})$	$LCard(\mathcal{S})$	Domain
CAL500	502	68	174	26.044	audio	rcv1-s2	6,000	944	101	2.634	text
emotions	593	72	6	1.868	audio	rcv1-s3	6,000	944	101	2.614	text
water_quality	1,060	16	14	5.073	chemistry	rcv1-s4	6,000	944	101	2.266	text
stackex_chess	1,675	585	227	2.411	text	rcv1-s5	6,000	944	101	2.625	text
enron	1,702	1001	53	3.378	text	bibtex	7,395	1,835	159	2.402	text
image	2,000	294	5	1.236	image	stackex_cs	9,270	635	274	2.556	text
recreation	2,073	606	22	2.021	text	NUS-WIDE-c	10,000	128	81	2.403	image
education	2,296	550	33	2.003	text	NUS-WIDE-b	10,000	500	81	2.412	image
yeast	2,417	103	14	4.237	biology	imdb	10,000	570	28	1.986	movie
social	2,684	737	38	2.010	text	corel16k001	13,761	500	164	2.883	image
arts	3,137	462	26	2.014	text	corel16k002	13,766	500	153	2.858	image
entertainment	3,450	545	21	2.036	text	delicious	16,064	500	301	3.074	text
computer	3,464	880	33	2.692	text	erulex_ed	19,314	100	301	3.074	text
business	3,813	581	29	2.425	text	eurlex_sm	19,338	100	201	2.214	text
health	3,816	1483	32	2.381	text	eurlex_dc	19,340	100	410	1.293	text
society	4,396	478	27	2.685	text	tmc2007	28,596	981	22	1.674	text
corel5k	5,000	499	374	3.522	image	mediamill	42,177	120	101	4.555	video
rcv1-s1	6,000	944	101	2.880	text						

Parameter configurations suggested in respective literatures are used for the comparing approaches: a) CLR: ensemble size $\binom{q}{2}$; b) LPLC: $k = 10$, $\alpha = 0.1$; c) MLFE: $\beta_1 = 1$, $\beta_2 = 10$, $\beta_3 = 10$; d) LIFT: $r = 0.1$; e) LLSF: $\alpha = 0.5$, $\beta = 0.5$, $\gamma = 0.5$; f) MLSF: $\epsilon = 0.01$, $\alpha = 0.8$, $\gamma = 0.01$. For BILAS, the parameter configuration corresponds to $t = 0.1$ and $r = 0.5$. Furthermore, Libsvm [8] is utilized as the binary training algorithm to instantiate BILAS as well as CLR, LIFT, MLSF.

Let $\mathcal{T} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \le i \le T\}$ be the test set and $h : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ be the learned multi-label classification model. Generally, a real-valued function $f : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ is returned by the multi-label learning system as an intermediate model to induce h, i.e. $h(\boldsymbol{x}_i) = \{y_u \mid f(\boldsymbol{x}_i, u) \ge t(\boldsymbol{x}_i),\ 1 \le u \le q\}$. Here, $t : \mathcal{X} \mapsto \mathbb{R}$ corresponds to the threshold function for bipartitioning relevant and irrelevant labels.[4] Accordingly, six popular multi-label evaluation metrics *hamming loss*, *example-based precision*, *coverage*, *ranking loss*, *average precision*, *macro-averaging AUC* [14, 55] are utilized for performance evaluation in this paper.

- *Hamming loss:*

$$hloss(h) = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{q}|h(\boldsymbol{x}_i)\Delta Y_i|, \quad \text{where } h(\boldsymbol{x}_i)\Delta Y_i = (h(\boldsymbol{x}_i) \setminus Y_i) \cup (Y_i \setminus h(\boldsymbol{x}_i))$$

- *Example-based precision:*

$$Prec_{\text{exam}} = \frac{1}{T}\sum_{i=1}^{T}\frac{|Y_i \bigcap h(\boldsymbol{x}_i)|}{|h(\boldsymbol{x}_i)|}$$

- *Coverage:*

$$coverage(h) = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{q}\max_{y_u \in Y_i}\left(\sum\nolimits_{y \in \mathcal{Y}}[\![f(\boldsymbol{x}_i, y_u) \le f(\boldsymbol{x}_i, y)]\!] - 1\right)$$

[4]For BILAS, we have $f(\boldsymbol{x}_i, y_u) = \Gamma(\boldsymbol{x}_i, y_u)$ (Eq.(8)) and $t(\boldsymbol{x}_i) = \Gamma(\boldsymbol{x}_i, y_V)$ (Eq.(9)). For each comparing approach, in order to avoid the case of empty prediction, the class label with largest modeling output will be put into the predicted label set [4, 55].

Table 3. Experimental results of comparing approaches in terms of *hamming loss*, where the best performance on each data set is shown in boldface (the smaller the value the better the performance).

Data set	Comparing approach						
	BiLAS	Clr	Lplc	Mlfe	Lift	Llsf	Mlsf
CAL500	**0.136±0.006**	0.144±0.006	0.166±0.015	0.155±0.007	0.137±0.006	0.137±0.005	0.137±0.006
emotions	**0.177±0.018**	0.196±0.024	0.213±0.025	0.191±0.020	0.184±0.020	0.201±0.021	0.206±0.023
water-quality	**0.303±0.021**	0.316±0.028	0.337±0.027	0.317±0.026	0.305±0.027	0.324±0.026	0.322±0.031
stackex_chess	**0.010±0.001**	0.018±0.006	0.011±0.001	0.011±0.002	**0.010±0.001**	**0.010±0.001**	**0.010±0.001**
enron	**0.049±0.005**	0.060±0.012	0.063±0.008	0.052±0.005	**0.049±0.005**	0.050±0.004	0.054±0.006
image	0.152±0.009	0.170±0.013	0.198±0.018	**0.150±0.016**	0.155±0.014	0.182±0.010	0.188±0.018
recreation	0.086±0.018	0.084±0.017	0.168±0.008	0.084±0.018	**0.083±0.019**	**0.083±0.015**	0.085±0.018
education	0.054±0.011	0.056±0.011	0.064±0.008	0.053±0.011	0.053±0.011	0.054±0.009	**0.052±0.011**
yeast	**0.191±0.007**	0.201±0.008	0.227±0.013	0.195±0.007	**0.191±0.007**	0.199±0.007	0.207±0.009
social	**0.052±0.012**	0.053±0.012	0.062±0.012	0.057±0.013	0.055±0.016	0.052±0.012	0.054±0.012
arts	0.071±0.013	0.068±0.013	0.118±0.017	0.068±0.013	**0.066±0.013**	0.067±0.013	0.068±0.014
entertainment	0.094±0.023	0.095±0.024	0.115±0.008	0.096±0.020	0.097±0.026	**0.093±0.021**	0.096±0.025
computer	**0.066±0.002**	0.090±0.001	0.079±0.006	0.071±0.005	0.071±0.002	0.074±0.004	0.070±0.002
business	**0.052±0.001**	0.089±0.013	0.077±0.003	0.056±0.002	**0.052±0.001**	0.057±0.001	**0.052±0.001**
health	**0.060±0.001**	0.073±0.015	0.078±0.002	0.063±0.001	0.074±0.016	**0.060±0.001**	0.070±0.004
society	**0.095±0.001**	0.096±0.001	0.113±0.001	0.100±0.002	0.096±0.002	0.096±0.001	0.096±0.002
corel5k	**0.009±0.000**	0.010±0.002	0.011±0.005	0.010±0.000	**0.009±0.000**	0.012±0.000	**0.009±0.000**
rcv1-s1	**0.028±0.000**	**0.028±0.000**	0.031±0.005	**0.028±0.000**	0.029±0.000	0.029±0.000	0.034±0.001
rcv1-s2	0.025±0.001	0.026±0.003	0.032±0.004	**0.024±0.002**	0.026±0.001	0.028±0.001	0.025±0.003
rcv1-s3	0.025±0.001	0.026±0.003	0.032±0.003	**0.024±0.002**	0.025±0.001	0.028±0.001	0.025±0.002
rcv1-s4	**0.021±0.002**	0.023±0.005	0.028±0.004	**0.021±0.002**	0.022±0.002	0.023±0.001	0.022±0.003
rcv1-s5	0.025±0.001	0.026±0.002	0.032±0.002	**0.024±0.001**	0.025±0.001	0.028±0.000	0.025±0.002
bibtex	**0.010±0.000**	0.014±0.000	0.016±0.004	0.018±0.000	0.012±0.000	0.020±0.000	0.012±0.000
stackex_cs	0.009±0.001	0.011±0.000	0.009±0.000	0.009±0.000	0.009±0.000	**0.008±0.000**	0.009±0.001
NUS-WIDE-c	0.027±0.000	**0.026±0.000**	0.031±0.005	0.027±0.000	**0.026±0.000**	0.027±0.000	**0.026±0.000**
NUS-WIDE-b	**0.026±0.000**	0.027±0.000	0.033±0.005	0.027±0.000	0.027±0.000	0.032±0.000	0.027±0.000
imdb	0.071±0.001	**0.070±0.001**	0.084±0.011	0.071±0.002	**0.070±0.001**	0.072±0.001	0.073±0.001
corel6k001	**0.016±0.001**	0.018±0.001	0.024±0.008	0.018±0.001	0.018±0.001	0.020±0.001	0.018±0.001
corel6k002	**0.017±0.001**	**0.017±0.002**	0.024±0.006	0.018±0.000	**0.017±0.002**	0.018±0.002	**0.017±0.002**
delicious	**0.040±0.000**	0.046±0.000	0.060±0.012	0.048±0.000	0.046±0.000	0.049±0.000	0.048±0.000
eurlex_ed	**0.009±0.002**	**0.009±0.000**	0.010±0.002	**0.009±0.003**	0.022±0.003	0.010±0.000	0.010±0.006
eurlex_sm	**0.008±0.008**	0.011±0.002	**0.008±0.001**	0.010±0.002	0.017±0.006	**0.008±0.000**	0.012±0.002
eurlex_dc	0.003±0.002	0.003±0.000	**0.002±0.000**	0.008±0.001	0.008±0.004	**0.002±0.000**	**0.002±0.000**
tmc2007	**0.062±0.002**	**0.062±0.002**	0.079±0.009	0.075±0.003	0.065±0.006	0.068±0.004	0.079±0.008
mediamill	**0.027±0.000**	**0.027±0.000**	0.035±0.002	0.032±0.000	0.301±0.000	0.031±0.000	0.036±0.001

- *Ranking loss:*

$$rloss(h) = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{|Y_i||\bar{Y}_i|}\left|\{(y_u, y_v) \mid f(\boldsymbol{x}_i, y_u) \le f(\boldsymbol{x}_i, y_v),\ (y_u, y_v) \in Y_i \times \bar{Y}_i\}\right|, \quad \text{where } \bar{Y}_i = \mathcal{Y} \setminus Y_i$$

- *Average precision:*

$$avgprec(h) = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{Y_i}\sum_{y_u \in Y_i}\frac{\sum_{y_v \in Y_i}[\![f(\boldsymbol{x}_i, y_u) \le f(\boldsymbol{x}_i, y_v)]\!]}{\sum_{y \in \mathcal{Y}}[\![f(\boldsymbol{x}_i, y_u) \le f(\boldsymbol{x}_i, y)]\!]}$$

- *Macro-averaging AUC:*

$$AUC_{\text{macro}} = \frac{1}{q}\sum_{u=1}^{q}\frac{|\{(\boldsymbol{x}', \boldsymbol{x}'') \mid f(\boldsymbol{x}', y_u) \ge f(\boldsymbol{x}'', y_u),\ (\boldsymbol{x}', \boldsymbol{x}'') \in \mathcal{P}_u \times \mathcal{N}_u\}|}{|\mathcal{P}_u||\mathcal{N}_u|}$$

$$\text{where } \mathcal{P}_u = \{\boldsymbol{x}_i \mid y_u \in Y_i,\ 1 \le i \le T\},\ \mathcal{N}_u = \{\boldsymbol{x}_i \mid y_u \notin Y_i,\ 1 \le i \le T\}$$

Table 4. Experimental results of comparing approaches in terms of *ranking loss*, where the best performance on each data set is shown in boldface (the smaller the metric value, the better the performance).

Data set	Comparing approach						
	BiLAS	Clr	Lplc	Mlfe	Lift	Llsf	Mlsf
CAL500	**0.174±0.008**	0.178±0.009	0.231±0.030	0.210±0.017	0.183±0.007	0.196±0.016	0.207±0.004
emotions	**0.137±0.019**	0.155±0.021	0.182±0.042	0.144±0.027	0.147±0.026	0.158±0.022	0.162±0.027
water-quality	**0.262±0.026**	0.290±0.033	0.307±0.028	0.282±0.027	0.280±0.026	0.292±0.032	0.306±0.031
stackex_chess	0.136±0.013	**0.112±0.010**	0.265±0.013	0.140±0.007	0.125±0.017	0.124±0.003	0.127±0.015
enron	**0.082±0.012**	0.096±0.020	0.221±0.041	0.134±0.007	0.088±0.010	0.108±0.010	0.097±0.012
image	0.136±0.021	0.171±0.018	0.204±0.033	**0.128±0.018**	0.141±0.022	0.173±0.022	0.186±0.019
recreation	0.195±0.010	0.192±0.016	0.350±0.042	0.239±0.031	**0.182±0.019**	0.212±0.020	0.208±0.028
education	**0.085±0.010**	0.086±0.007	0.141±0.012	0.154±0.020	**0.085±0.011**	0.130±0.023	0.120±0.030
yeast	**0.159±0.011**	0.177±0.012	0.188±0.032	0.165±0.011	0.163±0.013	0.171±0.013	0.194±0.021
social	**0.134±0.044**	0.150±0.044	0.318±0.079	0.257±0.042	0.144±0.048	0.187±0.045	0.146±0.050
arts	0.139±0.021	0.146±0.019	0.407±0.061	0.191±0.026	**0.135±0.022**	0.169±0.030	0.151±0.036
entertainment	**0.151±0.010**	0.175±0.010	0.274±0.015	0.206±0.027	0.166±0.012	0.223±0.010	0.240±0.083
computer	**0.116±0.011**	0.176±0.011	0.236±0.016	0.132±0.007	0.140±0.012	0.230±0.013	0.134±0.010
business	**0.078±0.002**	0.121±0.009	0.183±0.003	0.099±0.013	0.086±0.002	0.124±0.006	0.113±0.005
health	**0.075±0.003**	0.115±0.015	0.198±0.010	0.119±0.028	0.104±0.013	0.139±0.007	0.105±0.018
society	**0.182±0.003**	0.197±0.005	0.320±0.008	0.375±0.026	0.196±0.002	0.222±0.007	0.190±0.002
corel5k	0.201±0.046	0.200±0.047	0.686±0.119	0.254±0.031	**0.199±0.043**	0.481±0.039	0.213±0.051
rcv1-s1	0.051±0.005	**0.038±0.003**	0.394±0.114	0.064±0.005	0.055±0.004	0.116±0.008	0.098±0.008
rcv1-s2	0.052±0.005	**0.043±0.005**	0.327±0.082	0.053±0.004	0.058±0.005	0.100±0.004	0.087±0.016
rcv1-s3	0.054±0.006	**0.044±0.007**	0.332±0.076	0.051±0.004	0.058±0.005	0.099±0.005	0.094±0.008
rcv1-s4	**0.043±0.008**	0.045±0.015	0.275±0.104	0.055±0.009	0.045±0.007	0.085±0.007	0.066±0.017
rcv1-s5	0.050±0.003	**0.040±0.004**	0.315±0.060	0.049±0.004	0.054±0.004	0.098±0.006	0.081±0.011
bibtex	0.072±0.003	**0.058±0.005**	0.537±0.084	0.078±0.005	0.074±0.005	0.088±0.005	0.088±0.006
stackex_cs	0.079±0.011	**0.067±0.008**	0.436±0.025	0.096±0.005	0.101±0.010	0.074±0.004	0.102±0.012
NUS-WIDE-c	**0.083±0.004**	**0.083±0.002**	0.287±0.102	0.106±0.065	0.108±0.003	0.134±0.007	0.157±0.055
NUS-WIDE-b	**0.088±0.005**	0.094±0.003	0.336±0.102	0.165±0.065	0.106±0.003	0.202±0.010	0.179±0.075
imdb	**0.170±0.002**	0.189±0.002	0.360±0.100	0.189±0.004	0.172±0.002	0.205±0.003	0.180±0.003
corel6k001	**0.171±0.027**	0.179±0.029	0.599±0.130	0.191±0.021	0.197±0.038	0.411±0.047	0.212±0.043
corel6k002	**0.180±0.021**	**0.180±0.017**	0.623±0.102	0.227±0.014	0.202±0.022	0.394±0.015	0.210±0.031
delicious	**0.129±0.001**	**0.129±0.001**	0.350±0.094	0.178±0.006	0.163±0.001	0.173±0.002	0.200±0.003
eurlex_ed	**0.060±0.005**	0.066±0.002	0.371±0.084	0.096±0.078	0.168±0.005	0.158±0.003	0.254±0.068
eurlex_sm	**0.028±0.006**	**0.028±0.003**	0.206±0.057	0.065±0.034	0.071±0.008	0.075±0.004	0.081±0.064
eurlex_dc	**0.030±0.002**	0.031±0.001	0.282±0.053	0.156±0.005	0.098±0.004	0.113±0.006	0.080±0.005
tmc2007	**0.068±0.011**	0.070±0.012	0.161±0.073	0.098±0.015	0.070±0.012	0.072±0.013	0.154±0.032
mediamill	**0.048±0.001**	0.049±0.002	0.136±0.047	0.085±0.003	0.049±0.002	**0.048±0.001**	0.121±0.009

Briefly, *hamming loss* evaluates the fraction of instance-label pairs which are misclassified, *example-based precision* evaluates the fraction of predicted labels which are truly relevant, *coverage* evaluates the average number steps needed to move down the ranked label list in order to cover all relevant labels, *ranking loss* evaluates the fraction of relevant-irrelevant label pairs which are reversely ordered, *average precision* evaluates the average fraction of relevant labels which rank higher than a particular relevant label, and *macro-averaging AUC* evaluates the average AUC values across all class labels. As shown in the above definitions, these metrics are capable of investigating the quality of the predicted label set $h(\boldsymbol{x}_i)$ (with $f(\boldsymbol{x}_i, \cdot)$ being the intermediate model) from various aspects. All metric values are normalized in [0,1] where for *hamming loss*, *coverage* and *ranking loss* the smaller the value the better the performance, and for the other three metrics the larger the value the better the performance.

Table 5. Experimental results of comparing approaches in terms of *macro-averaging AUC*, where the best performance on each data set is shown in boldface (the larger the metric value, the better the performance).

Data set	Comparing approach						
	BiLAS	Clr	Lplc	Mlfe	Lift	Llsf	Mlsf
CAL500	**0.582±0.055**	0.573±0.050	0.529±0.014	0.540±0.043	0.543±0.036	0.555±0.042	0.518±0.026
emotions	0.840±0.023	0.834±0.023	0.816±0.039	**0.851±0.027**	0.840±0.027	0.835±0.029	0.836±0.032
water-quality	0.655±0.061	0.646±0.057	0.638±0.039	**0.674±0.044**	0.665±0.042	0.649±0.052	0.654±0.033
stackex_chess	0.751±0.017	**0.808±0.012**	0.605±0.003	0.712±0.025	0.757±0.012	0.791±0.013	0.741±0.020
enron	0.679±0.013	**0.701±0.015**	0.573±0.028	0.647±0.053	0.682±0.019	0.697±0.021	0.645±0.015
image	0.845±0.022	0.827±0.018	0.818±0.030	**0.874±0.022**	0.860±0.026	0.828±0.022	0.822±0.024
recreation	0.665±0.032	0.694±0.029	0.538±0.022	0.685±0.029	**0.704±0.033**	0.702±0.028	0.690±0.039
education	0.689±0.035	0.716±0.028	0.669±0.034	0.693±0.036	**0.719±0.025**	0.721±0.035	0.702±0.033
yeast	0.652±0.016	0.648±0.014	0.691±0.025	**0.721±0.016**	0.699±0.016	0.686±0.017	0.647±0.013
social	0.573±0.027	0.595±0.017	0.548±0.028	0.582±0.026	0.545±0.031	**0.598±0.026**	0.567±0.035
arts	0.717±0.031	0.724±0.015	0.569±0.015	0.721±0.031	**0.729±0.023**	0.726±0.025	0.725±0.024
entertainment	0.589±0.033	0.588±0.022	0.573±0.013	**0.615±0.039**	0.584±0.037	0.613±0.023	0.587±0.032
computer	0.653±0.011	0.673±0.010	0.616±0.007	**0.706±0.024**	0.657±0.030	0.675±0.017	0.643±0.020
business	0.620±0.019	0.632±0.020	0.572±0.009	0.665±0.024	0.583±0.033	**0.683±0.009**	0.597±0.019
health	0.681±0.029	0.703±0.030	0.601±0.013	0.675±0.023	0.631±0.038	**0.712±0.018**	0.662±0.019
society	**0.599±0.017**	0.580±0.014	0.539±0.004	0.568±0.011	0.538±0.011	0.592±0.015	0.572±0.009
corel5k	0.603±0.042	**0.610±0.047**	0.516±0.014	0.562±0.038	0.608±0.045	0.532±0.015	0.561±0.033
rcv1-s1	**0.921±0.010**	0.911±0.012	0.616±0.029	0.872±0.016	0.898±0.013	0.628±0.009	0.843±0.010
rcv1-s2	**0.886±0.019**	0.873±0.017	0.653±0.027	0.779±0.018	0.863±0.020	0.792±0.020	0.810±0.018
rcv1-s3	**0.878±0.022**	0.868±0.022	0.648±0.030	0.776±0.015	0.861±0.022	0.793±0.039	0.804±0.021
rcv1-s4	**0.887±0.023**	0.879±0.022	0.667±0.037	0.824±0.024	0.871±0.019	0.778±0.012	0.826±0.028
rcv1-s5	0.880±0.024	**0.883±0.024**	0.644±0.024	0.787±0.025	0.876±0.023	0.799±0.015	0.820±0.029
bibtex	**0.924±0.004**	0.919±0.006	0.658±0.033	0.906±0.006	0.909±0.005	0.906±0.004	0.869±0.006
stackex_cs	0.868±0.007	0.882±0.007	0.641±0.005	0.802±0.017	0.825±0.007	**0.887±0.007**	0.773±0.015
NUS-WIDE-c	**0.796±0.008**	0.791±0.009	0.625±0.031	0.786±0.012	0.681±0.014	0.677±0.010	0.702±0.014
NUS-WIDE-b	**0.758±0.014**	0.755±0.009	0.589±0.025	0.736±0.014	0.704±0.016	0.605±0.012	0.682±0.016
imdb	0.561±0.014	0.573±0.012	0.508±0.006	0.560±0.014	0.575±0.020	**0.587±0.013**	0.545±0.016
corel6k001	**0.685±0.026**	0.653±0.033	0.535±0.017	0.611±0.035	0.610±0.030	0.570±0.010	0.595±0.024
corel6k002	**0.684±0.031**	0.655±0.041	0.532±0.014	0.679±0.047	0.612±0.036	0.576±0.017	0.593±0.029
delicious	**0.796±0.005**	0.788±0.003	0.675±0.026	0.790±0.002	0.745±0.004	0.777±0.002	0.718±0.001
eurlex_ed	**0.936±0.009**	0.917±0.002	0.781±0.034	0.905±0.005	0.816±0.007	0.831±0.002	0.845±0.025
eurlex_sm	**0.928±0.024**	0.918±0.006	0.758±0.029	0.908±0.026	0.788±0.014	0.798±0.005	0.886±0.045
eurlex_dc	**0.932±0.021**	0.911±0.008	0.723±0.025	0.905±0.045	0.760±0.011	0.725±0.009	0.808±0.016
tmc2007	0.887±0.025	0.882±0.007	0.781±0.043	**0.898±0.009**	0.875±0.036	0.885±0.008	0.858±0.023
mediamill	**0.889±0.006**	0.883±0.007	0.707±0.019	0.878±0.005	0.885±0.009	0.840±0.007	0.854±0.021

4.2 Experimental Results

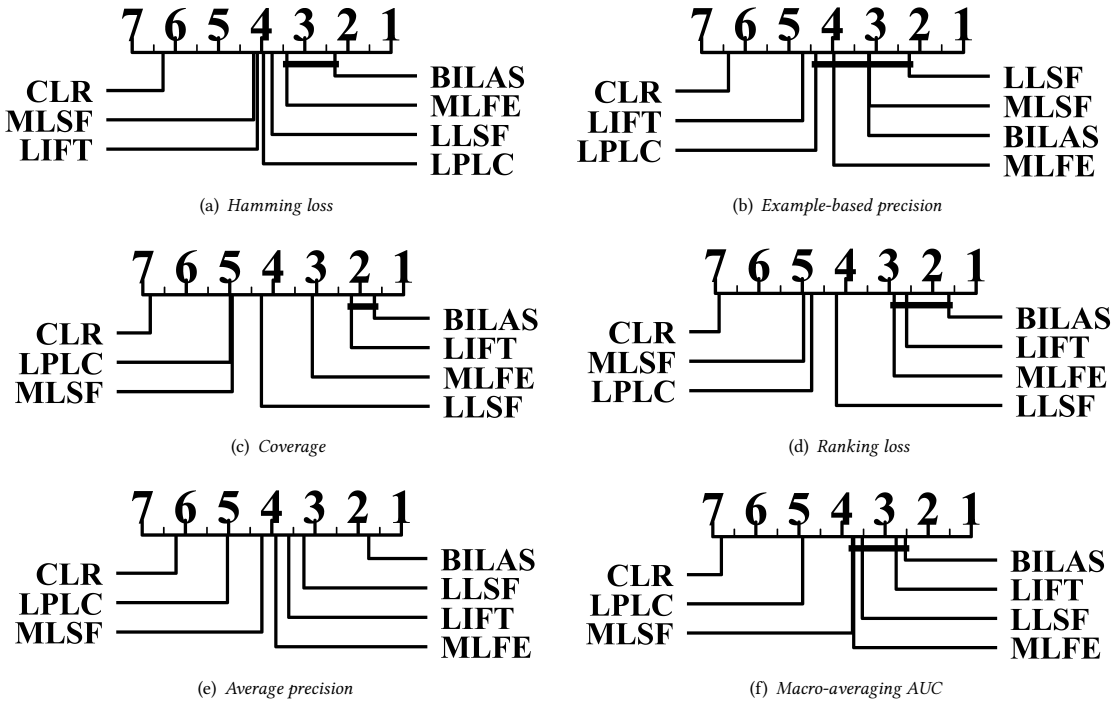
On each data set, ten-fold cross-validation is performed for each comparing approach where the mean metric value as well as standard deviation are recorded. Tables 3 to 5 report the detailed experimental results in terms of *hamming loss*, *ranking loss* and *macro-averaging AUC* respectively, where the best performance on each data set is shown in boldface.[5]

In addition, the widely-used *Friedman test* [11] is used in this paper for statistical comparison among multiple approaches over a number of data sets. Table 6 reports the Friedman statistics over all evaluation metrics as well as the critical value at 0.05 significance level (# comparing approaches $n = 7$; # data sets $N = 35$). As shown in Table 6, the F_F value is greater than the critical value 2.1432 in terms of all evaluation metrics. Therefore, the null hypothesis of "equal" performance among comparing approaches should be clearly rejected.

[5]For brevity, the detailed experimental results in terms of the other three evaluation metrics are reported in Appendix A.

Table 6. Friedman statistics F_F in terms of each evaluation metric as well as the critical value at 0.05 significance level (# comparing approaches $n = 7$, # data sets $N = 35$).

Evaluation metric	F_F	critical value
Hamming loss	16.8482	2.1432
Example-based precision	23.1480	
Coverage	78.0906	
Ranking loss	72.1566	
Average precision	26.3561	
Macro-averaging AUC	28.5878	



(a) *Hamming loss* (b) *Example-based precision*

(c) *Coverage* (d) *Ranking loss*

(e) *Average precision* (f) *Macro-averaging AUC*

Fig. 1. Comparison of BILAS (control approach) against six comparing approaches with the *Bonferroni-Dunn test*. Approaches not connected with BILAS in the CD diagram are considered to have significantly different performance from the control approach (CD=1.5228 at 0.05 significance level).

Therefore, *Bonferroni-Dunn test* [11] is employed as the post-hoc test to show relative performance among the comparing approaches. Here, the difference between the average ranks of control approach (i.e. BILAS) and one comparing approach is calibrated with the *critical difference* (CD). If the average ranks between BILAS and one comparing approach differ by at least one CD (CD=1.5228 with $n = 7$ and $N = 35$), their performance is deemed to be significantly different.

Figure 1 illustrates the CD diagrams [11] on each evaluation metric by treating BiLAS as the control approach. The average rank of each comparing approach is marked along the axis with lower ranks to the right, where any comparing approach whose average rank is within one CD to BiLAS is interconnected to each other with a thick line. Otherwise, it is considered to have significantly different performance against BiLAS.

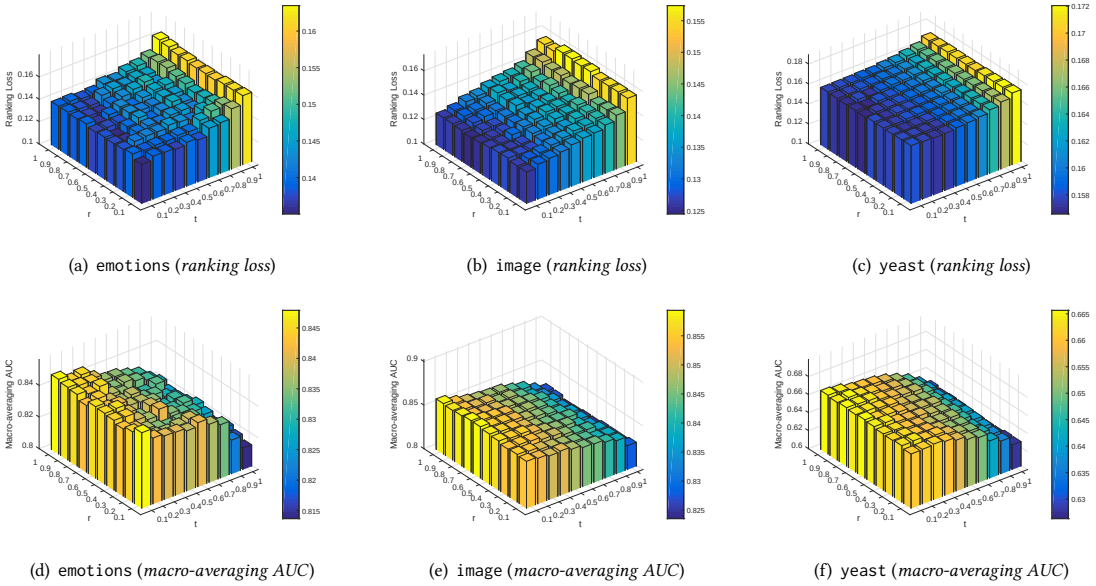
Based on the reported experimental results, the following observations can be made:

- As shown in Fig. 1, it is impressive that BiLAS achieves the lowest rank out of all the comparing approaches in terms of all evaluation metrics except example-based precision. Furthermore, most comparing approaches except Clr and Lift achieve statistically comparable performance in terms of example-based precision.
- Among the first series of comparing approaches without considering label-specific features, BiLAS significantly outperforms Clr in terms of all evaluation metrics. These results clearly indicate the effectiveness of generating BiLabel-specific features for multi-label label classification, where one major algorithmic difference between Clr and BiLAS lies in the utilization of identical features (for Clr) or BiLabel-specific features (for BiLAS). Furthermore, BiLAS is comparable to Mlfe in terms of *hamming loss*, *example-based precision*, *ranking loss*, *macro-averaging AUC*, comparable to Lplc in terms of *example-based precision*, and significantly outperforms both Mlfe and Lplc in the rest 7 cases.
- Among the second series of comparing approaches based on label-specific features, BiLAS significantly outperforms Llsf and Mlsf in terms of *hamming loss*, *coverage*, *ranking loss* and *average precision*. Furthermore, BiLAS is comparable to Lift in terms of *coverage*, *ranking loss*, *macro-averaging AUC*, and significantly outperforms Lift in the rest 3 cases. These results clearly indicate the strategy of *BiLabel-specific* features serves as a more effective way in achieving strong generalization performance than traditional strategy of *unilabel-specific* features across various multi-label evaluation metrics.
- As shown in Tables 3-5 and 8-10, the performance advantage of BiLAS over the comparing approaches is generally more pronounced on data sets with larger number of examples ($|\mathcal{S}| \geq 10,000$). One potential reason for this lies in that as the number of training examples increases, the number of selected prototypes for feature embedding also increases which could lead to robust embedding feature representation for subsequent classifier induction.
- On some data sets with challenging characteristics such as `CAL500` (least # examples and large # class labels), `health` and `bibtex` (large # features), `eurlex_dc` (largest # class labels), BiLAS achieves better performance than the comparing approaches in most cases. These results indicate that BiLabel-specific features could yield noticeable benefits for multi-label classification tasks under challenging circumstances.

4.3 Further Analysis

4.3.1 Parameter Sensitivity. As shown in Table 1, two parameters need to be instantiated to implement the BiLAS approach, including t (balancing parameter in Eq.(3)) and r (ratio parameter in Eq.(4)).

Figure 2 gives an illustrative example on how the performance of BiLAS changes with varying parameter configurations $t \in \{0.1, 0.2, \ldots, 1.0\}$ and $r \in \{0.1, 0.2, \ldots, 1.0\}$ on the `emotions`, `image` and `yeast` data sets (evaluation metrics: *ranking loss* and *macro-averaging AUC*). As shown in Figure 2, the performance of BiLAS is relatively stable as the value of r increases under fixed value of t. On the other hand, the performance of BiLAS gradually decreases as



(a) emotions (*ranking loss*) (b) image (*ranking loss*) (c) yeast (*ranking loss*)

(d) emotions (*macro-averaging AUC*) (e) image (*macro-averaging AUC*) (f) yeast (*macro-averaging AUC*)

Fig. 2. Performance of BILAS changes with varying parameter configurations $t \in \{0.1, 0.2, \ldots, 1.0\}$ and $r \in \{0.1, 0.2, \ldots, 1.0\}$ (Data sets: emotions, image, yeast; First row: *ranking loss*, the smaller the better; Second row: *macro-averaging AUC*, the larger the better).

the value of t increases under fixed value of r. As per these observations, the parameter configuration of BILAS given in Subsection 4.1 ($t = 0.1$, $r = 0.5$) is employed in this paper for experimental studies.

4.3.2 Ablation Study. In this subsection, ablation study on two variants of BILAS is further conducted to analyze the essential building blocks of BILAS. As shown in Table 1, for a pair of class labels y_u and y_v, the prototypes in $\overline{\mathcal{P}}_{uv}$ and $\overline{\mathcal{N}}_{uv}$ used for feature embedding are selected based on the heursitic ranking score given in Eq.(3) (Steps 1-8). To show the effectiveness of heuristic prototype selection, a variant named BILAS-R is investigated which works by choosing the embedding prototypes in a fully *random* manner while keeps other learning process of BILAS unchanged.

Secondly, for each induced binary classifier g_{uv} (g_{uV}), its empirical accuracy ϑ_{uv} (ϑ_{uV}) is utilized in yielding weighted voting for model prediction (Eqs.(8) and (9)). To show the effectiveness of empirical weighted voting, a variant named BILAS-U is investigated which works by taking *uniform* weighted voting while keeps other learning process of BILAS unchanged.

Table 7 reports the experimental results of BILAS and its two variants BILAS-R, BILAS-U on ten benchmark data sets, where the best performance on each data set is shown in boldface. Out of the 60 cases (10 data sets $\times$ 6 evaluation metrics), pairwise t-tests at 0.05 significance level show that BILAS achieves superior or at least comparable performance than BILAS-R and BILAS-U in 90.0% and 78.3% cases respectively. These results clearly validate the usefulness of both *heuristic prototype selection* and *empirical weighted voting* building blocks for BILAS to learn from multi-label examples with BiLabel-specific features.

4.3.3 Algorithmic Complexity. Let $\mathcal{F}_{\mathcal{B}}(a, b)$ be the training complexity of the binary learning algorithm $\mathcal{B}$ w.r.t. a training examples and b features, and $\mathcal{F}'_{\mathcal{B}}(b)$ be the testing complexity of $\mathcal{B}$ in classifying one unseen instance with

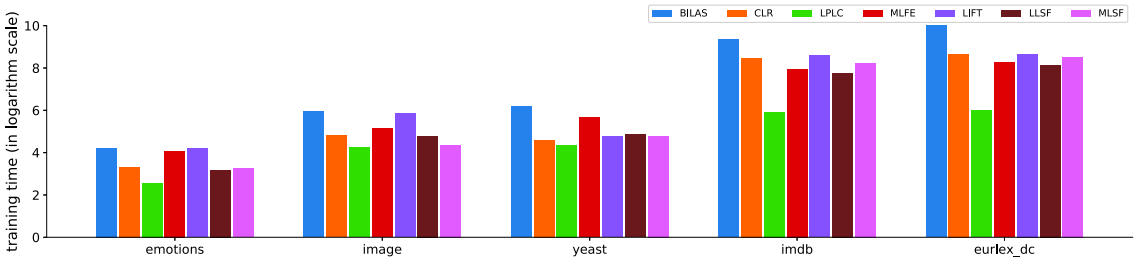
Table 7. Experimental results of BILAS and its two variants BILAS-R, BILAS-U on ten benchmark data sets, where the best performance on each data set is shown in boldface (↓: the smaller the metric value, the better the performance; ↑: the larger the metric value, the better the performance).

Data set	Comparing approach					
	hamming loss ↓			*example-based precision* ↑		
	BILAS	BILAS-R	BILAS-U	BILAS	BILAS-R	BILAS-U
CAL500	**0.136±0.006**	0.138±0.006	0.137±0.006	**0.617±0.034**	0.602±0.035	0.609±0.035
emotions	**0.177±0.018**	0.186±0.023	**0.177±0.018**	0.736±0.043	0.748±0.030	**0.770±0.042**
water-quality	**0.303±0.021**	0.304±0.021	0.305±0.022	0.614±0.066	0.607±0.062	**0.625±0.072**
stackex_chess	**0.010±0.001**	**0.010±0.001**	**0.010±0.001**	**0.820±0.086**	0.819±0.092	0.753±0.066
enron	0.049±0.005	**0.048±0.008**	**0.048±0.008**	0.729±0.037	0.731±0.055	**0.751±0.056**
image	**0.152±0.009**	**0.152±0.012**	**0.152±0.009**	0.803±0.022	0.789±0.037	**0.814±0.025**
yeast	**0.191±0.007**	**0.191±0.008**	**0.191±0.008**	**0.727±0.017**	**0.727±0.020**	**0.727±0.018**
arts	0.071±0.013	**0.067±0.013**	0.068±0.013	**0.826±0.073**	0.782±0.045	0.809±0.040
society	**0.095±0.001**	**0.095±0.002**	**0.095±0.002**	0.593±0.030	**0.604±0.035**	0.595±0.041
imdb	**0.071±0.001**	**0.071±0.001**	**0.071±0.002**	**0.495±0.225**	0.392±0.033	0.404±0.036
	coverage ↓			*ranking loss* ↓		
	BILAS	BILAS-R	BILAS-U	BILAS	BILAS-R	BILAS-U
CAL500	**0.740±0.017**	0.742±0.017	0.743±0.017	**0.174±0.008**	**0.174±0.009**	**0.174±0.009**
emotions	**0.276±0.030**	0.277±0.034	0.278±0.031	**0.137±0.019**	0.140±0.022	0.146±0.020
water-quality	**0.636±0.053**	0.637±0.058	**0.636±0.053**	**0.262±0.026**	0.264±0.024	0.276±0.026
stackex_chess	0.274±0.015	**0.246±0.043**	0.260±0.043	0.136±0.013	**0.119±0.019**	0.128±0.019
enron	0.234±0.032	0.230±0.050	**0.228±0.050**	**0.082±0.012**	0.083±0.018	0.083±0.018
image	**0.163±0.017**	**0.163±0.018**	**0.163±0.017**	0.136±0.021	**0.135±0.022**	0.146±0.019
yeast	**0.441±0.017**	0.443±0.017	0.443±0.017	**0.159±0.011**	0.165±0.011	0.165±0.011
arts	0.234±0.061	**0.230±0.061**	0.233±0.061	0.139±0.021	**0.135±0.022**	0.144±0.022
society	**0.348±0.005**	**0.348±0.008**	**0.348±0.007**	0.182±0.003	**0.181±0.005**	0.187±0.004
imdb	**0.266±0.004**	**0.266±0.006**	0.269±0.003	**0.170±0.002**	0.175±0.003	0.174±0.007
	average precision ↑			*macro-averaging AUC* ↑		
	BILAS	BILAS-R	BILAS-U	BILAS	BILAS-R	BILAS-U
CAL500	**0.510±0.017**	**0.510±0.021**	**0.510±0.021**	**0.582±0.055**	0.580±0.055	0.580±0.055
emotions	**0.827±0.022**	0.824±0.027	0.826±0.021	**0.840±0.023**	0.837±0.025	0.837±0.023
water-quality	**0.678±0.052**	0.677±0.047	0.677±0.052	**0.655±0.061**	**0.655±0.053**	0.654±0.059
stackex_chess	0.390±0.025	0.415±0.027	**0.434±0.026**	0.751±0.017	**0.802±0.018**	0.784±0.019
enron	0.670±0.036	**0.677±0.051**	0.671±0.051	0.679±0.013	0.702±0.024	**0.703±0.016**
image	**0.833±0.025**	0.831±0.023	0.832±0.023	**0.845±0.022**	0.842±0.023	0.843±0.022
yeast	**0.770±0.013**	**0.770±0.013**	**0.770±0.013**	0.652±0.017	0.652±0.017	**0.654±0.016**
arts	0.577±0.026	**0.587±0.027**	0.582±0.027	**0.717±0.031**	**0.717±0.032**	0.716±0.033
society	**0.514±0.008**	0.513±0.012	**0.514±0.012**	**0.599±0.017**	0.579±0.021	0.580±0.024
imdb	**0.477±0.008**	0.472±0.009	**0.477±0.006**	**0.561±0.014**	0.560±0.013	0.559±0.011

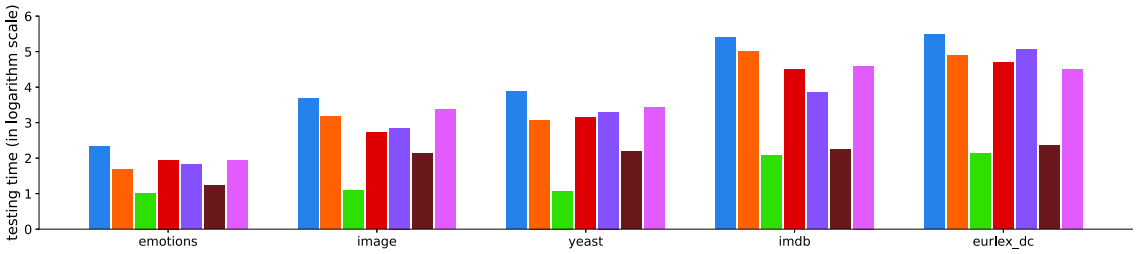
b features. According to the learning procedure given in Table 1, the (worst-case) training and testing complexity of BILAS correspond to $O\left(q^2\left(m^2(m+d)+\mathcal{F}_{\mathcal{B}}(m,\lceil r\cdot m\rceil)+m\cdot\mathcal{F}'_{\mathcal{B}}(\lceil r\cdot m\rceil)\right)\right)$ and $O\left(q^2\left(d\lceil r\cdot m\rceil+\mathcal{F}'_{\mathcal{B}}(\lceil r\cdot m\rceil)\right)\right)$ respectively.[6]

Furthermore, Figure 3 illustrates the training and testing time of each comparing approach on five benchmark data sets `emotions`, `image`, `yeast`, `imdb` and `eurlex_dc`. Generally, the empirical training time of BILAS is relatively comparable to the comparing approaches. Furthermore, the empirical testing time of BILAS is higher than LPLC and LLSF while relatively comparable to the other comparing approaches. However, it is worth noting that due to the

[6]During the training phase, for each candidate prototype $\boldsymbol{\mu}_k \in \mathcal{P}_{uv} \bigcup \mathcal{N}_{uv}$ $(1 \le u < v \le q,\ k \le m)$, the (worst-case) computational complexities of Eq.(2) and Eq.(3) correspond to $O(md)$ and $O(m^2)$ respectively. The number of training examples and feature dimensionality in $\overline{\mathcal{D}}_{uv}$ (Eq.(6)) are upper-bounded by m and $\lceil r\cdot m\rceil$, which leads to computational complexity of $\mathcal{F}_{\mathcal{B}}(m,\lceil r\cdot m\rceil)$ to train binary classifier g_{uv}. In addition, it takes up to $m\cdot\mathcal{F}'_{\mathcal{B}}(\lceil r\cdot m\rceil)$ calculations to estimate the empirical accuracy ϑ_{uv} of g_{uv}. During the test phase, for each pair of class labels, the computational complexities of feature embedding and model prediction over unseen instance $\boldsymbol{x}^*$ correspond to $O(d\lceil r\cdot m\rceil)$ and $\mathcal{F}'_{\mathcal{B}}(\lceil r\cdot m\rceil)$ respectively.



(a) training time



(b) testing time

Fig. 3. Running time (train/test) of each comparing approach on five benchmark data sets (for histogram illustration, the y-axis corresponds to the value of $\log_{10} t$ with t being the running time measured in seconds).

quadratic computational complexity of BILAS w.r.t. the number of class labels (i.e. q), the proposed approach cannot scale up to the case of extreme multi-label classification [2, 3, 31, 35] where thousands or even millions of class labels exist in the label space.

5 CONCLUSION

In this paper, the problem of multi-label classification based on label-specific features is investigated. Specifically, a novel label-specific features strategy is proposed which differs from existing label-wise strategy by generating tailored features for a pair of class labels. Accordingly, the BiLabel-specific features strategy is instantiated by generating tailored features based on heuristic prototype selection and embedding. After that, an ensemble of binary classifiers are induced based on the BiLabel-specific features whose modeling outputs are aggregated via empirical weighted voting. Comprehensive experimental studies over a total of thirty-five benchmark data sets show that the proposed BiLabel-specific features strategy is capable of achieving superior generalization performance than state-of-the-art label-specific features techniques for multi-label classification.

REFERENCES

[1] J. Amores. 2013. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence* 201 (2013), 81–105.
[2] R. Babbar and B. Schölkopf. 2019. Data scarcity, robustness and extreme multi-label classification. *Machine Learning* 108 (2019), 1329–1351.
[3] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. 2018. Sparse Local Embeddings for Extreme Multi-label Classification. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.). MIT Press, Cambridge, MA, 730ÍC738.
[4] M. Boutell, J. Luo, X. Shen, and C. M. Brown. 2004. Learning multi-label scene classification. *Pattern Recognition* 37, 9 (2004), 1757–1771.

[5] C. Brinker, E. Loza Mencía, and J. Fürnkranz. 2014. Graded multilabel classification by pairwise comparisons. In *Proceedings of the 14th IEEE International Conference on Data Mining*. Shenzhen, China, 731–736.

[6] S. Canuto, M. A. Gonçalves, and F. Benevenuto. 2016. Exploiting new sentiment-based meta-level features for effective sentiment analysis. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. San Francisco, CA, 53–62.

[7] M.-A. Carbonneaua, V. Cheplyginabc, E. Granger, and G. Gagnon. 2018. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition* 77 (2018), 329–353.

[8] C.-C. Chang and C.-J. Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 3 (2011), Article 27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[9] Z.-M. Chen, X.-S. Wei, P. Wang, and Y.-W. Guo. 2019. Multi-label image recognition with graph convolutional networks. In *Proceedings of the 32nd IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, CA, 5177–5186.

[10] Z.-S. Chen and M.-L. Zhang. 2019. Multi-label learning with regularization enriched label-specific features. In *Proceedings of the 11th Asian Conference on Machine Learning*. Nagoya, Japan, 411–424.

[11] J. Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, Jan (2006), 1–30.

[12] J. Feng and Z.-H. Zhou. 2017. Deep MIML network. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. San Francisco, CA, 1884–1890.

[13] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker. 2008. Multilabel classification via calibrated label ranking. *Machine Learning* 73, 2 (2008), 133–153.

[14] E. Gibaja and S. Ventura. 2015. A tutorial on multilabel learning. *ACM Computing Surveys* 47, 3 (2015), Article 52.

[15] H. Gouk, B. Pfahringer, and M. Cree. 2016. Learning Distance Metrics for Multi-Label Classification. In *Proceedings of the 8th Asian Conference on Machine Learning*. Hamilton, New Zealand, 318–333.

[16] Y. Guo, F. Chung, G. Li, J. Wang, and J. C. Gee. 2019. Leveraging label-specific discriminant mapping features for multi-label learning. *ACM Transactions on Knowledge Discovery from Data* 13, 2 (2019), Article 24.

[17] J. Huang, G. Li, Q. Huang, and X. Wu. 2016. Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering* 28, 12 (2016), 3309–3323.

[18] J. Huang, G. Li, Q. Huang, and X. Wu. 2018. Joint feature selection and classification for multilabel learning. *IEEE Transactions on Cybernetics* 48, 3 (2018), 876–889.

[19] J. Huang, G. Li, S. Wang, Z. Xue, and Q. Huang. 2017. Multi-label classification by exploiting local positive and negative pairwise label correlation. *Neurocomputing* 257 (2017), 164–174.

[20] M. Huang, F. Zhuang, X. Zhang, X. Ao, Z. Niu, M.-L. Zhang, and Q. He. 2019. Supervised representation learning for multi-label classification. *Machine Learning* 108, 5 (2019), 747–763.

[21] S.-J. Huang, G.-X. Li, W.-Y. Huang, and S.-Y. Li. 2020. Incremental multi-label learning with active queries. *Journal of Computer Science and Technology* 35, 2 (2020), 234–246.

[22] B.-B. Jia and M.-L. Zhang. 2020. Multi-dimensional classification via kNN feature augmentation. *Pattern Recognition* 106 (2020), Article 107423.

[23] B.-B. Jia and M.-L. Zhang. 2020. Multi-dimensional classification via stacked dependency exploitation. *Science China Information Sciences* 63, 12 (2020), Article 222102.

[24] X.-Y. Jia, S.-S. Zhu, and W.-W. Li. 2020. Joint label-specific features and correlation information for multi-label learning. *Journal of Computer Science and Technology* 35, 2 (2020), 247–258.

[25] X.-C. Li, D.-C. Zhan, J.-Q. Yang, and Y. Shi. 2021. Deep multiple instance selection. *Science China Information Sciences* 64, 3 (2021), Article 130102.

[26] Y. Li, Y. Song, and J. Luo. 2017. Improving pairwise ranking for multi-label image classification. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, HI, 3617–3625.

[27] W. Liu and I. W. Tsang. 2015. Large Margin Metric Learning for Multi-Label Prediction. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. Austin, TX, 2800–2806.

[28] J. Ma, H. Zhang, and T. W. S. Chow. 2021. Multilabel classification with label-specific features and classifiers: A coarse- and fine-tuned framework. *IEEE Transactions on Cybernetics* 15, 2 (2021), 1028–1042.

[29] Y. Ma, C. Cui, X. Nie, G. Yang, K. Shaheed, and Y. Yin. 2019. Pre-course student performance prediction with multi-instance multi-label learning. *Science China Information Sciences* 62, 2 (2019), Article 029101.

[30] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann. 2018. Categorizing feature selection methods for multi-label classification. *Artificial Intelligence Review* 49, 1 (2018), 57–78.

[31] Y. Prabhu and M. Varma. 2014. FastXML: a fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, 263–272.

[32] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang. 2007. Correlative multi-label video annotation. In *Proceedings of the 15th ACM International Conference on Multimedia*. Augsburg, Germany, 17–26.

[33] J. Read, C. Bielza, and P. Larrañaga. 2014. Multi-dimensional classification with super-classes. *IEEE Transactions on Knowledge and Data Engineering* 26, 7 (2014), 1720–1733.

[34] J. Read, B. Pfahringer, G. Holmes, and E. Frank. 2011. Classifier chains for multi-label classification. *Machine Learning* 85, 3 (2011), 333–359.

[35] W. Siblini, P. Kuntz, and F. Meyer. 2018. CRAFTML, an Efficient Clustering-based Random Forest for Extreme Multi-label Learning. In *Proceedings of the 35th International Conference on Machine Learning*. Vienna, Austria, 4664–4673.

[36] L. Sun, S. Ji, and J. Ye. 2013. *Multi-label Dimensionality Reduction*. Chapman and Hall/CRC, Boca Ration, FL.

[37] L. Sun, M. Kudo, and K. Kimura. 2016. Multi-label classification with meta-label-specific features. In *Proceedings of the 23rd International Conference on Pattern Recognition*. Cancun, Mexico, 1612–1617.

[38] Y.-P. Sun and M.-L. Zhang. 2021. Compositional Metric Learning for Multi-Label Classification. *Frontiers of Computer Science* 15, 5 (2021), Article 155320.

[39] G. Tsoumakas, I. Katakis, and I. Vlahavas. 2011. Random k-labelsets for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering* 23, 7 (2011), 1079–1089.

[40] L. Wang, Z. Ding, S. Han, J.-J. Han, C. Choi, and Y. Fu. 2019. Generative correlation discovery network for multi-label learning. In *Proceedings of the 19th IEEE International Conference on Data Mining*. Beijing, China, 588–597.

[41] Y. Wang, W. Zheng, Y. Cheng, and D. Zhao. 2020. Joint label completion and label-specific features for multi-label learning algorithm. *Soft Computing* 24 (2020), 6553–6569.

[42] W. Weng, Y.-N. Chen, C.-L. Chen, S.-X. Wu, and J.-H. Liu. 2020. Non-sparse label specific features selection for multi-label classification. *Neurocomputing* 377 (2020), 85–94.

[43] W. Weng, Y. Lin, S. Wu, Y. Li, and Y. Kang. 2018. Multi-label learning based on label-specific features and local pairwise label correlation. *Neurocomputing* 273 (2018), 385–394.

[44] X. Wu, Q.-G. Chen, Y. Hu, D. Wang, X. Chang, X. Wang, and M.-L. Zhang. 2019. Multi-view multi-label learning with view-specific information extraction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. Macau, China, 3884–3890.

[45] Y. Xing, G. Yu, D. Carlotta, J. Wang, and Z. Zhang. 2018. Multi-label co-training. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. Stockholm, Sweden, 2882–2888.

[46] J.-H. Xu, H.-D. Tian, Z.-Y. Wang, Y. Wang, F. Chen, and W.-X. Kang. in press. Joint input and output space learning for multi-label image classification. *IEEE Transactions on Multimedia* (in press).

[47] M. Xu and L.-Z. Guo. 2021. Learning from group supervision: The impact of supervision deficiency on multi-label learning. *Science China Information Sciences* 64, 3 (2021), Article 130101.

[48] S. Xu, X. Yang, H. Yu, D.-J. Yu, J. Yang, and E. C. C. Tsang. 2016. Multi-label learning with label-specific feature reduction. *Knowledge-Based Systems* 104 (2016), 52–61.

[49] Y. Yang and S. Gopal. 2012. Multilabel classification with meta-level features in a learning-to-rank framework. *Machine Learning* 88, 1-2 (2012), 47–68.

[50] C. K. Yeh, W. C. Wu, W. J. Ko, and Y. C. F. Wang. 2017. Learning deep latent spaces for multi-label classification. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. San Francisco, CA, 2838–2844.

[51] C. Zhang, Z. Yu, Q. Hu, P. Zhu, X. Liu, and X. Wang. 2018. Latent semantic aware multi-view multi-label classification. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. New Orleans, LA, 4414–4421.

[52] J. Zhang, C. Li, D. Cao, Y. Lin, S. Su, L. Dai, and S. Li. 2018. Multi-label learning with label-specific features by resolving label correlations. *Knowledge-Based Systems* 159 (2018), 148–157.

[53] M.-L. Zhang, Y.-K. Li, Y.-Y. Liu, and X. Geng. 2018. Binary relevance for multi-label learning: An overview. *Frontiers of Computer Science* 12, 2 (2018), 191–202.

[54] M.-L. Zhang and L. Wu. 2015. LIFT: Multi-label learning with label-specific features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 1 (2015), 107–120.

[55] M.-L. Zhang and Z.-H. Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2014), 1819–1837.

[56] Q.-W. Zhang and M.-L. Zhang. 2018. Feature-induced labeling information enrichment for multi-label learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. New Orleans, LA, 4446–4453.

[57] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li. 2012. Multi-instance multi-label learning. *Artificial Intelligence* 176, 1 (2012), 2291–2320.

[58] S. Zhu, X. Ji, W. Xu, and Y. Gong. 2005. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Salvador, Brazil, 274–281.

A DETAILED EXPERIMENTS IN TERMS OF THREE EVALUATION METRICS

Table 8. Experimental results of comparing approaches in terms of *example-based precision*, where the best performance on each data set is shown in boldface (the larger the value the better the performance).

Data set	Comparing approach						
	BiLAS	Clr	Lplc	Mlfe	Lift	Llsf	Mlsf
CAL500	0.617±0.034	0.597±0.046	0.505±0.026	0.562±0.046	0.613±0.037	0.620±0.033	**0.630±0.028**
emotions	0.736±0.043	0.750±0.046	0.633±0.051	**0.782±0.048**	0.751±0.045	0.689±0.041	0.651±0.039
water-quality	0.614±0.066	**0.628±0.114**	0.538±0.049	0.570±0.066	0.610±0.072	0.603±0.097	0.604±0.113
stackex_chess	**0.820±0.086**	0.259±0.058	0.490±0.054	0.805±0.043	0.727±0.053	0.719±0.045	0.659±0.052
enron	0.729±0.037	0.598±0.107	0.445±0.072	**0.763±0.017**	0.728±0.017	0.706±0.045	0.650±0.028
image	0.803±0.022	0.772±0.041	0.572±0.030	**0.846±0.040**	0.783±0.040	0.778±0.038	0.643±0.039
recreation	**0.858±0.055**	0.680±0.061	0.189±0.053	0.707±0.061	0.762±0.037	0.699±0.077	0.687±0.138
education	**0.771±0.076**	0.761±0.087	0.516±0.080	0.742±0.068	0.716±0.071	0.681±0.074	0.712±0.064
yeast	0.727±0.017	0.734±0.021	0.623±0.017	**0.737±0.016**	0.726±0.015	0.716±0.018	0.669±0.018
social	0.434±0.045	**0.556±0.049**	0.356±0.028	0.499±0.046	0.384±0.129	0.531±0.036	0.509±0.100
arts	**0.826±0.073**	0.348±0.072	0.460±0.073	0.787±0.079	0.751±0.033	0.733±0.057	0.631±0.067
entertainment	0.570±0.125	0.560±0.091	0.376±0.122	0.506±0.090	**0.803±0.154**	0.571±0.070	0.580±0.113
computer	0.715±0.041	0.477±0.015	0.562±0.036	**0.728±0.021**	0.657±0.031	0.638±0.053	0.668±0.027
business	0.743±0.012	0.489±0.247	0.574±0.016	0.757±0.006	0.728±0.014	**0.758±0.018**	0.730±0.010
health	0.681±0.013	0.578±0.177	0.487±0.017	0.687±0.020	0.568±0.171	**0.691±0.019**	0.559±0.055
society	**0.593±0.030**	0.567±0.023	0.437±0.007	0.585±0.018	0.547±0.020	**0.593±0.028**	0.553±0.021
corel5k	0.285±0.043	0.041±0.035	0.152±0.029	0.321±0.059	0.209±0.096	**0.327±0.064**	0.311±0.094
rcv1-s1	0.353±0.149	0.422±0.063	0.342±0.040	**0.466±0.121**	0.370±0.145	0.392±0.117	0.395±0.078
rcv1-s2	0.086±0.173	0.578±0.177	0.487±0.017	**0.687±0.020**	0.568±0.171	0.482±0.102	0.488±0.094
rcv1-s3	0.442±0.115	0.494±0.067	0.337±0.130	**0.595±0.151**	0.410±0.124	0.501±0.108	0.488±0.075
rcv1-s4	0.581±0.083	0.545±0.124	0.411±0.058	**0.709±0.109**	0.514±0.102	0.636±0.084	0.612±0.047
rcv1-s5	0.469±0.105	0.494±0.059	0.337±0.044	**0.596±0.131**	0.430±0.123	0.536±0.077	0.502±0.049
bibtex	0.875±0.013	0.597±0.008	0.653±0.035	**0.974±0.004**	0.821±0.014	0.862±0.011	0.780±0.011
stackex_cs	0.684±0.063	0.495±0.031	0.532±0.042	**0.717±0.033**	0.704±0.036	0.681±0.038	0.635±0.042
NUS-WIDE-c	0.719±0.015	0.707±0.004	0.515±0.009	**0.727±0.010**	0.683±0.008	0.686±0.006	0.678±0.024
NUS-WIDE-b	**0.722±0.024**	0.667±0.027	0.437±0.011	0.681±0.021	0.637±0.017	0.650±0.024	0.636±0.033
imdb	0.495±0.225	0.357±0.014	0.314±0.016	0.404±0.022	0.366±0.017	**0.500±0.095**	0.288±0.067
corel16k001	0.292±0.031	0.279±0.037	0.145±0.020	0.250±0.030	0.270±0.029	**0.318±0.035**	0.220±0.036
corel16k002	0.298±0.000	0.279±0.030	0.131±0.031	0.262±0.024	0.293±0.015	**0.329±0.013**	0.198±0.017
delicious	**0.711±0.009**	0.518±0.007	0.364±0.003	0.584±0.007	0.663±0.012	0.487±0.006	0.614±0.004
eurlex_ed	0.140±0.006	0.119±0.002	0.335±0.009	**0.445±0.011**	0.282±0.008	0.348±0.008	0.282±0.023
eurlex_sm	0.315±0.005	0.365±0.488	0.699±0.010	**0.903±0.004**	0.824±0.008	0.856±0.006	0.806±0.012
eurlex_dc	0.158±0.065	0.062±0.088	**0.462±0.005**	**0.462±0.005**	0.379±0.006	0.383±0.007	0.423±0.015
tmc2007	0.706±0.060	0.636±0.097	0.606±0.062	**0.800±0.058**	0.674±0.111	0.739±0.063	0.672±0.060
mediamill	0.779±0.008	0.672±0.030	0.595±0.003	**0.793±0.003**	0.779±0.011	0.786±0.004	0.789±0.007

Table 9. Experimental results of comparing approaches in terms of *coverage*, where the best performance on each data set is shown in boldface (the smaller the value the better the performance).

Data set	Comparing approach						
	BiLAS	Clr	Lplc	Mlfe	Lift	Llsf	Mlsf
CAL500	**0.174±0.008**	0.178±0.009	0.231±0.030	0.210±0.017	0.183±0.007	0.196±0.016	0.207±0.004
emotions	**0.137±0.019**	0.155±0.021	0.182±0.042	0.144±0.027	0.147±0.026	0.158±0.022	0.162±0.027
water-quality	**0.636±0.053**	0.642±0.055	0.678±0.051	0.638±0.059	0.653±0.059	0.660±0.056	0.670±0.053
stackex_chess	0.274±0.015	**0.221±0.012**	0.451±0.029	0.276±0.014	0.248±0.022	0.252±0.013	0.253±0.021
enron	**0.234±0.032**	0.237±0.036	0.470±0.078	0.337±0.031	0.248±0.031	0.287±0.030	0.259±0.036
image	0.136±0.021	0.171±0.018	0.204±0.033	**0.128±0.018**	0.141±0.022	0.173±0.022	0.186±0.019
recreation	0.303±0.057	**0.284±0.060**	0.461±0.034	0.365±0.088	0.290±0.068	0.335±0.073	0.341±0.087
education	0.152±0.044	**0.146±0.039**	0.234±0.056	0.260±0.067	0.154±0.045	0.230±0.068	0.233±0.081
yeast	**0.159±0.011**	0.177±0.012	0.188±0.032	0.165±0.011	0.163±0.013	0.171±0.013	0.194±0.021
social	**0.224±0.083**	0.225±0.081	0.356±0.105	0.398±0.085	0.235±0.087	0.275±0.083	0.243±0.093
arts	0.234±0.061	**0.228±0.056**	0.399±0.090	0.304±0.072	0.229±0.061	0.284±0.077	0.257±0.086
entertainment	**0.241±0.055**	0.255±0.051	0.356±0.057	0.309±0.059	0.258±0.051	0.313±0.062	0.395±0.188
computer	**0.244±0.014**	0.295±0.012	0.366±0.012	0.267±0.009	0.282±0.019	0.379±0.011	0.283±0.025
business	**0.170±0.005**	0.212±0.009	0.266±0.006	0.211±0.026	0.182±0.005	0.222±0.009	0.246±0.012
health	**0.160±0.005**	0.199±0.014	0.296±0.009	0.240±0.051	0.198±0.015	0.245±0.008	0.210±0.039
society	**0.348±0.005**	0.357±0.007	0.466±0.012	0.614±0.030	0.367±0.004	0.392±0.007	0.361±0.004
corel5k	0.201±0.046	0.200±0.047	0.686±0.119	0.254±0.031	**0.199±0.043**	0.481±0.039	0.213±0.051
rcv1-s1	0.051±0.005	**0.038±0.003**	0.394±0.114	0.064±0.005	0.055±0.004	0.116±0.008	0.098±0.008
rcv1-s2	0.052±0.005	**0.043±0.005**	0.327±0.082	0.053±0.004	0.058±0.005	0.100±0.004	0.087±0.016
rcv1-s3	0.054±0.006	**0.044±0.007**	0.332±0.076	0.051±0.004	0.058±0.005	0.099±0.005	0.094±0.008
rcv1-s4	**0.043±0.008**	0.045±0.015	0.275±0.104	0.055±0.009	0.045±0.007	0.085±0.007	0.066±0.017
rcv1-s5	0.050±0.003	**0.040±0.004**	0.315±0.060	0.049±0.004	0.054±0.004	0.098±0.006	0.081±0.011
bibtex	0.072±0.003	**0.058±0.005**	0.537±0.084	0.078±0.005	0.074±0.005	0.088±0.005	0.088±0.006
stackex_cs	0.162±0.018	**0.136±0.013**	0.433±0.022	0.207±0.009	0.206±0.014	0.159±0.007	0.204±0.017
NUS-WIDE-c	**0.083±0.004**	**0.083±0.002**	0.287±0.102	0.106±0.065	0.108±0.003	0.134±0.007	0.157±0.055
NUS-WIDE-b	**0.088±0.005**	0.094±0.003	0.336±0.102	0.165±0.065	0.106±0.003	0.202±0.010	0.179±0.075
imdb	**0.170±0.002**	0.189±0.002	0.360±0.100	0.189±0.004	0.172±0.002	0.205±0.003	0.180±0.003
corel6k001	**0.171±0.027**	0.179±0.029	0.599±0.130	0.191±0.021	0.197±0.038	0.411±0.047	0.212±0.043
corel6k002	**0.180±0.021**	**0.180±0.017**	0.623±0.102	0.227±0.014	0.202±0.022	0.394±0.015	0.210±0.031
delicious	**0.129±0.001**	**0.129±0.001**	0.350±0.094	0.178±0.006	0.163±0.001	0.173±0.002	0.200±0.003
eurlex_ed	**0.060±0.005**	0.066±0.002	0.371±0.084	0.096±0.078	0.168±0.005	0.158±0.003	0.254±0.068
eurlex_sm	**0.028±0.006**	**0.028±0.003**	0.206±0.057	0.065±0.034	0.071±0.008	0.075±0.004	0.081±0.064
eurlex_dc	**0.030±0.002**	0.031±0.001	0.282±0.053	0.156±0.005	0.098±0.004	0.113±0.006	0.080±0.005
tmc2007	**0.068±0.011**	0.070±0.012	0.161±0.073	0.098±0.015	0.070±0.012	0.072±0.013	0.154±0.032
mediamill	**0.048±0.001**	0.049±0.002	0.136±0.047	0.085±0.003	0.049±0.002	**0.048±0.001**	0.121±0.009

Table 10. Experimental results of comparing approaches in terms of *average precision*, where the best performance on each data set is shown in boldface (the larger the value the better the performance).

Data set	Comparing approach						
	BiLAS	Clr	Lplc	Mlfe	Lift	Llsf	Mlsf
CAL500	**0.510±0.017**	0.487±0.023	0.457±0.026	0.456±0.027	0.498±0.015	0.498±0.024	0.473±0.015
emotions	**0.827±0.022**	0.814±0.025	0.787±0.040	0.816±0.034	0.817±0.027	0.807±0.027	0.797±0.030
water-quality	**0.678±0.052**	0.663±0.056	0.644±0.045	0.651±0.046	0.664±0.046	0.647±0.060	0.638±0.068
stackex_chess	0.390±0.025	0.315±0.046	0.298±0.020	0.442±0.014	0.426±0.020	**0.497±0.018**	0.377±0.013
enron	**0.670±0.022**	0.608±0.083	0.462±0.048	0.629±0.023	**0.670±0.015**	0.663±0.026	0.623±0.037
image	0.833±0.025	0.807±0.021	0.768±0.031	**0.842±0.017**	0.826±0.027	0.790±0.022	0.776±0.022
recreation	0.512±0.019	0.535±0.014	0.314±0.055	0.506±0.011	**0.542±0.016**	**0.542±0.010**	0.512±0.027
education	0.635±0.017	**0.636±0.027**	0.578±0.029	0.594±0.016	0.635±0.010	0.619±0.013	0.619±0.018
yeast	**0.770±0.013**	0.759±0.013	0.757±0.017	0.766±0.013	**0.770±0.013**	0.761±0.011	0.732±0.022
social	**0.511±0.081**	0.485±0.065	0.406±0.049	0.429±0.056	0.484±0.095	0.497±0.071	0.501±0.078
arts	0.577±0.026	0.581±0.021	0.340±0.023	0.548±0.022	**0.588±0.024**	0.576±0.025	0.569±0.035
entertainment	**0.545±0.055**	0.521±0.060	0.483±0.069	0.510±0.054	0.501±0.082	0.527±0.031	0.467±0.046
computer	0.636±0.023	0.451±0.011	0.574±0.021	**0.642±0.010**	0.579±0.020	0.572±0.016	0.605±0.020
business	0.767±0.008	0.515±0.059	0.709±0.006	0.747±0.010	0.760±0.010	0.738±0.009	0.751±0.007
health	**0.675±0.009**	0.558±0.097	0.592±0.012	0.656±0.025	0.556±0.090	0.653±0.007	0.618±0.018
society	**0.514±0.008**	0.509±0.008	0.452±0.008	0.416±0.018	0.494±0.007	0.499±0.012	0.501±0.008
corel5k	**0.223±0.037**	0.129±0.047	0.106±0.041	0.201±0.042	0.210±0.042	0.150±0.038	0.205±0.030
rcv1-s1	**0.602±0.016**	0.598±0.014	0.339±0.026	0.512±0.014	0.511±0.018	0.420±0.020	0.509±0.014
rcv1-s2	**0.611±0.012**	0.606±0.033	0.370±0.058	0.596±0.019	0.531±0.014	0.536±0.014	0.531±0.054
rcv1-s3	0.565±0.011	**0.603±0.042**	0.360±0.073	0.592±0.017	0.533±0.013	0.532±0.013	0.532±0.042
rcv1-s4	**0.638±0.023**	0.629±0.103	0.437±0.009	0.609±0.042	0.606±0.032	0.622±0.032	0.614±0.061
rcv1-s5	**0.625±0.015**	0.619±0.026	0.393±0.033	0.602±0.018	0.533±0.014	0.542±0.014	0.537±0.027
bibtex	**0.594±0.004**	0.555±0.013	0.250±0.008	0.544±0.006	0.568±0.007	0.570±0.010	0.537±0.011
stackex_cs	0.487±0.027	0.472±0.018	0.332±0.021	0.493±0.020	0.464±0.024	**0.520±0.020**	0.391±0.025
NUS-WIDE-c	**0.584±0.016**	0.548±0.010	0.469±0.048	0.568±0.031	0.522±0.012	0.515±0.010	0.468±0.031
NUS-WIDE-b	0.514±0.012	0.513±0.010	0.421±0.036	0.504±0.065	**0.516±0.010**	0.458±0.013	0.446±0.054
imdb	0.477±0.008	0.467±0.008	0.409±0.035	0.480±0.017	**0.479±0.009**	0.463±0.008	0.461±0.013
society	**0.597±0.068**	0.585±0.008	0.523±0.035	0.587±0.008	0.546±0.074	0.555±0.010	0.572±0.009
computer	**0.668±0.054**	0.400±0.010	0.613±0.031	0.650±0.014	0.602±0.068	0.607±0.018	0.642±0.015
corel6k001	0.312±0.022	0.282±0.033	0.161±0.032	**0.313±0.036**	0.281±0.041	0.261±0.037	0.247±0.038
corel6k002	**0.295±0.019**	0.276±0.017	0.152±0.025	**0.295±0.016**	0.273±0.013	0.254±0.012	0.238±0.029
delicious	0.438±0.006	0.448±0.004	0.344±0.027	**0.452±0.005**	0.405±0.004	0.419±0.004	0.351±0.006
eurlex_ed	**0.632±0.120**	0.568±0.122	0.624±0.019	0.612±0.014	0.321±0.112	0.521±0.006	0.601±0.015
eurlex_sm	**0.645±0.112**	0.576±0.123	0.632±0.016	0.598±0.015	0.367±0.122	0.532±0.005	0.541±0.026
eurlex_dc	**0.581±0.101**	0.548±0.131	0.561±0.008	0.504±0.024	0.224±0.100	0.410±0.009	0.490±0.010
tmc2007	0.800±0.065	**0.812±0.064**	0.693±0.070	0.808±0.048	0.801±0.048	0.770±0.040	0.784±0.036
mediamill	0.728±0.008	0.728±0.006	0.711±0.015	**0.730±0.004**	0.723±0.004	0.723±0.003	0.715±0.002