Multi-Dimensional Multi-Label Classification: Towards Encompassing Heterogeneous Label Spaces and Multi-Label Annotations

Bin-Bin Jia[a,b], Min-Ling Zhang[a,c,*]

[a] *School of Computer Science and Engineering, Southeast University, Nanjing 210096, China*
[b] *College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China*
[c] *Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China*

Abstract

In traditional classification framework, the semantics of each object is usually characterized by annotating a single class label from one homogeneous label space. Nonetheless, objects with rich semantics naturally arise in real-world applications whose properties need to be characterized in a more sophisticated manner. In this paper, a new classification framework named *Multi-Dimensional Multi-Label* (MDML) classification is investigated which models objects with rich semantics by encompassing heterogeneous label spaces and multi-label annotations. Specifically, MDML generalizes the traditional classification framework by assuming a number of *heterogeneous label spaces* to characterize semantics from different dimensions, where each object is further annotated with *multiple class labels* from each heterogeneous label space. To learn from MDML training examples, a first attempt named CLIM is proposed based on an augmented stacking strategy. Firstly, CLIM induces a base multi-label predictive model w.r.t. each label space by maximizing the likelihood of the observed multiple class labels. Secondly, the thresholding predictions from all base models are used to augment the original feature space to yield stacked multi-label predictive models. The two-level models are refined alternately via empirical threshold tun-

*Corresponding author
Email address: zhangml@seu.edu.cn (Min-Ling Zhang)

ing. Experiments on four real-world MDML data sets validate the effectiveness of CLIM in learning from training examples with heterogeneous label spaces and multi-label annotations.
Keywords: Machine learning, supervised learning, multi-dimensional classification, multi-label classification, multi-dimensional multi-label classification

1. Introduction

In traditional supervised learning, one common task is to learn from objects whose semantics are characterized by annotating a single class label from one homogeneous label space, e.g., multi-class classification. Although this learning setting has been successfully applied in many real-world applications [1, 2, 3], some recent studies show that this simplified assumption may not work well when the needs of modeling objects with rich semantics arise. On one hand, if the semantics of objects need to be characterized with multiple label spaces from different dimensions rather than a single label space, then multi-dimensional classification is more suitable [4, 5]. For example, given one vehicle, we are usually interested in its information from the type dimension (including possible classes car, SUV, bus, etc.) and from the brand dimension (including possible classes Audi, Benz, BMW, etc.) [6, 7]. On the other hand, if the semantics of objects need to be characterized with multiple class labels rather than a single class label, then multi-label classification is more suitable [8, 9]. For example, one natural scene image may contain several kinds of sceneries simultaneously (e.g., desert, mountains, sunset, etc.) [10, 11].

In this paper, we focus on a more sophisticated scenario which encompasses *heterogeneous label spaces* and *multi-label annotations* to characterize the rich semantics of objects. In fact, the needs of learning in such scenario naturally arise in many real-world applications. Take song categorization as an example, each song can be categorized by its emotion, genre, scenario, and each song might also contain many different emotions, belong to many kinds of genres, be suitable for

playing in many scenarios. We formalize this problem as a new learning framework called *Multi-Dimensional Multi-Label* (MDML) classification. Compared to multi-class/multi-label/multi-dimensional classification, each MDML example is also represented by a single instance while associated with multiple sets of labels. Here, each set of labels belong to one heterogeneous label space [12, 13] which characterizes the semantics of objects from one specific dimension. Similar to MDML, the traditional multi-class/multi-label/multi-dimensional classification can also be referred to as *Single-Dimensional Single-Label* (SDSL) classification, *Single-Dimensional Multi-Label* (SDML) classification, *Multi-Dimensional Single-Label* (MDSL) classification, respectively.

Obviously, the MDML problem cannot be solved by directly adopting existing multi-class/multi-dimensional classification methods due to the multi-label nature in each label space.[1] If the MDML problem is *decomposed* into multiple multi-label classification problems, one per dimension, or *transformed* into one multi-label classification problem by simply concatenating all label spaces into a single one, existing multi-label classification methods could be used to solve the MDML problem. However, the decomposition strategy deals with each dimension independently and thus cannot consider the label correlations across different dimensions, while the transformation strategy aligns labels from heterogeneous label spaces into a homogeneous one and thus cannot consider the multi-dimensional nature of heterogeneous label spaces in MDML. In other words, the derived MDML model should not only consider the two kinds of label correlations within individual dimension and across multiple dimensions, but also treat them in different ways.

To address these issues, an approach named CLIM, i.e., *Considering Label*

[1] Generally speaking, when designing multi-dimensional classification methods, the characteristics of single relevant label in each label space will be specially exploited which prevents the designed methods from being applied to solve MDML problem [14, 15]. Nonetheless, some ideas of dealing with multi-dimensional semantic spaces in multi-dimensional classification can still be borrowed to design MDML methods.

correlations within Individual dimension and across Multiple dimensions, is proposed to solve the MDML problem based on an augmented stacking strategy. Specifically, CLIM induces a predictive model for each dimension by maximizing the likelihood of relevant labels to consider the label correlations within individual dimension, where it utilizes the fact that modeling outputs of labels from the same label space are comparable. To consider the label correlations across multiple dimensions, CLIM augments the original feature space with binary predictions from predictive models w.r.t. all dimensions and then refines these predictive models based on the augmented feature space. The predictive models and augmented feature space will be updated alternately until convergence. To investigate the effectiveness of the proposed approach, comparative studies are conducted over four real-world MDML data sets, and the experimental results clearly show the effectiveness of the proposed CLIM approach for solving MDML problems.

The rest of this paper is organized as follows. Firstly, Section 2 discusses some related works on multi-dimensional multi-label classification. Then, Section 3 presents the technical details of two benchmark MDML approaches and the proposed CLIM approach. After that, Section 4 conducts experimental studies to investigate the effectiveness of the proposed approach. Finally, Section 5 concludes this paper.

2. Related Work

The multi-dimensional multi-label classification framework is closely related to multi-dimensional classification and multi-label classification. Figure 1 shows an intuitive comparison among different classification paradigms. Specifically, multi-class classification can be regarded as a special case of both multi-dimensional classification and multi-label classification, where multi-dimensional classification generalizes the single label space assumption in multi-class classification to multiple label spaces to characterize multi-dimensional semantics of objects, and multi-label classification generalizes the single relevant label assumption in

multi-class classification to multiple relevant labels to characterize ambiguous semantics of objects. Furthermore, both multi-dimensional classification and multi-label classification can be regarded as a special case of multi-dimensional multi-label classification, which can be regarded as either the generalization of multi-dimensional classification by no longer restricting single relevant label in each label space or the generalization of multi-label classification by assuming multiple label spaces in output space.

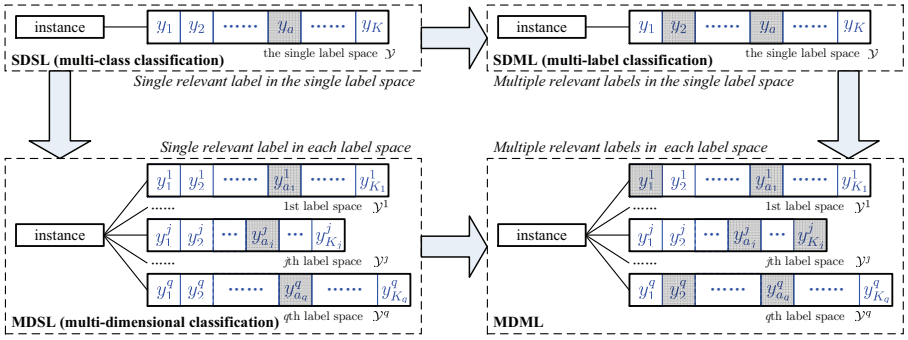


Figure 1: Relationships among SDSL (multi-class classification), SDML (multi-label classification), MDSL (multi-dimensional classification) and the proposed MDML framework. Here, relevant labels are shown in shaded style.

Here, we would like to further discuss what new challenges the multi-dimensional assumption in MDML brings for model induction. Conceptually, each label space characterizes the semantics of objects from one dimension and each label in one label space specifies the relevancy of one concept in this dimension. Generally, we can optimize the MDML model to make the modeling outputs of relevant labels larger than the modeling outputs of irrelevant labels in the same label space. However, we cannot require that the modeling outputs of relevant labels from one label space are larger than the modeling outputs of either relevant or irrelevant labels from another label space. In other words, the modeling outputs of labels from different label spaces are not directly comparable even though they are correlated to each other. Take song categorization as an example, we can only expect that, if the modeling output of label '*happy*'

from `emotion` dimension is larger, then the modeling outputs of labels '*wedding*' and '*memorial*' from `scenario` dimension are likely to be larger and smaller, respectively. We cannot require that the modeling output of label '*happy*' is larger than the modeling output of label '*memorial*' even if '*memorial*' is an irrelevant label.

Multi-label classification has been widely studied in the past two decades. The basic strategy for solving multi-label classification problem is binary relevance (BR) [16], which deals with each label independently. However, this strategy is usually criticized for being incapable of considering label correlations. An improved strategy is to deal with each label in a chaining structure, i.e., classifier chains (CC) [17, 18], where labeling information for training previous classifiers on the chain will be used as extra features when training subsequent classifiers. Another common strategy to consider label correlations is to optimize the label ranking results [19, 20] based on the single label space assumption, i.e., promoting the modeling outputs of relevant labels while depressing the modeling outputs of irrelevant labels. In addition to explicitly consider label correlations in label space, learning label-specific features [21, 22] has also been shown as a good alternative solution for multi-label classification, where a distinct feature representation for each label is extracted to help induce the multi-label predictive model [23, 24]. For more details of multi-label classification (e.g., problem formulation, evaluation metrics, categorization and description of existing algorithms, emerging trends and new challenges), the comprehensive surveys in [25, 26] are recommended.

Multi-dimensional classification has also attracted more and more attentions in recent years. The BR strategy in multi-label field can also be adapted to solve the multi-dimensional classification problem in a dimension-wise manner. To consider label correlations across different dimensions, one common strategy is to assume some specific structure over label spaces, such as directed acyclic graph [27, 28], chaining structure [29, 30], super-class partition [4], etc. Due to the heterogeneity of label spaces, it is very challenging to consider label correlations with only one specific structure. Recent studies show that it is helpful to progressively consider low-order and high-order label correlations in a

cascaded way [31, 32] or learn the multi-dimensional classification model in a transformed label space [33, 34]. In addition to these aforementioned approaches which aim at manipulating the output space, another recently proposed feature augmentation strategy [35, 36] which aims at manipulating feature space has also been shown as an effective solution for multi-dimensional classification, where an augmented feature vector is generated for each example by exploiting its labeling information and then predictive model is induced based on the concatenation of the original and augmented features.

3. MDML Approaches

3.1. Problem Formulation

Let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional input (feature) space and $\mathcal{Y} = \mathcal{Y}^1 \cup \mathcal{Y}^2 \cup \ldots \cup \mathcal{Y}^q$ be the output space. Here, $\mathcal{Y}$ corresponds to the union of q heterogeneous label spaces and there are K_j labels in the jth label space $\mathcal{Y}^j$, i.e., $\mathcal{Y}^j = \{y_1^j, y_2^j, \ldots, y_{K_j}^j\}$ $(1 \leq j \leq q)$. Given a set of MDML training examples $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{l}_i) \mid 1 \leq i \leq m\}$, for each example $(\boldsymbol{x}_i, \boldsymbol{l}_i) \in \mathcal{D}$, $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^\top \in \mathcal{X}$ is the d-dimensional instance vector and $\boldsymbol{l}_i = [\boldsymbol{l}_i^1; \boldsymbol{l}_i^2; \ldots; \boldsymbol{l}_i^q] \in \{0,1\}^K$ is the corresponding binary label vector associated with $\boldsymbol{x}_i$, where $\boldsymbol{l}_i^j = [l_{i1}^j, l_{i2}^j, \ldots, l_{iK_j}^j]^\top \in \{0,1\}^{K_j}$ and $K = \sum_{j=1}^q K_j$. The ath entry l_{ia}^j in $\boldsymbol{l}_i^j$ indicates whether the ath label (i.e., y_a^j) in the jth label space (i.e., $\mathcal{Y}^j$) is relevant to $\boldsymbol{x}_i$ or not (1-relevant, 0-irrelavant). Besides, we use comma/semicolon to represent row/column vector concatenation throughout this paper. The task of MDML is to learn a multi-dimensional multi-label classifier $f : \mathcal{X} \mapsto \{0,1\}^K$ from $\mathcal{D}$ which can properly assign a label vector $\hat{\boldsymbol{l}}_* = f(\boldsymbol{x}_*)$ for unseen instance $\boldsymbol{x}_*$.

Moreover, we use $h(\cdot)$ to denote the corresponding hypothesis whose outputs are real-valued vector instead of binary one, and the relationship between $f(\cdot)$ and $h(\cdot)$ is $f(\boldsymbol{x}) = \mathcal{T}(h(\boldsymbol{x}))$, where $\mathcal{T}(\cdot)$ denotes the threshold function that can divide label set into relevant part and irrelevant part according to the real-valued modeling outputs of $h(\cdot)$. We also use $f^j(\boldsymbol{x}) \in \{0,1\}^{K_j}$ and $h^j(\boldsymbol{x}) \in \mathbb{R}^{K_j}$ to

denote the corresponding predictions of $\boldsymbol{x}$ w.r.t. the jth label space, and use $f_a^j(\boldsymbol{x}) \in \{0,1\}$ and $h_a^j(\boldsymbol{x}) \in \mathbb{R}$ to denote the ath entry of $f^j(\boldsymbol{x})$ and $h^j(\boldsymbol{x})$, respectively. To facilitate understanding, Table 1 summarizes the notations used in this section.

Table 1: Notations used in the MDML formulation.

Notation	Descriptions
d	the number of features in input space
q	the number of label spaces (dimensions) in output space
K_j	the number of labels in the jth label space ($1 \leq j \leq q$)
K	the number of labels in the whole output space, i.e., $K = \sum_{j=1}^{q} K_j$
m	the number of training examples
$\mathcal{X}$	the d-dimensional input (feature) space, i.e., $\mathcal{X} = \mathbb{R}^d$
$\mathcal{Y}^j$	the jth label space where $\mathcal{Y}^j = \{y_1^j, y_2^j, \ldots, y_{K_j}^j\}$ ($1 \leq j \leq q$)
y_a^j	the ath label in $\mathcal{Y}^j$ ($1 \leq a \leq K_j$)
$\mathcal{Y}$	the output space where $\mathcal{Y} = \mathcal{Y}^1 \cup \mathcal{Y}^2 \cup \ldots \cup \mathcal{Y}^q$
$\mathcal{D}$	the MDML training set where $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{l}_i) \mid 1 \leq i \leq m\}$
$\boldsymbol{x}_i$	the ith feature vector where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^\top \in \mathcal{X}$
$\boldsymbol{l}_i$	the label vector associated with $\boldsymbol{x}_i$ where $\boldsymbol{l}_i = [\boldsymbol{l}_i^1; \boldsymbol{l}_i^2; \ldots; \boldsymbol{l}_i^q] \in \{0,1\}^K$
$\boldsymbol{l}_i^j$	the part of label vector in $\boldsymbol{l}_i$ w.r.t. $\mathcal{Y}^j$ where $\boldsymbol{l}_i^j = [l_{i1}^j, l_{i2}^j, \ldots, l_{iK_j}^j]^\top \in \{0,1\}^{K_j}$
l_{ia}^j	the ath entry in $\boldsymbol{l}_i^j$ where $l_{ia}^j = 1$ (or 0) denotes that y_a^j is relevant (or irrelevant) to $\boldsymbol{x}_i$
f (or h)	the MDML predictive model which returns binary (or real-valued) predictions
f^j (or h^j)	the part of MDML predictive model f (or h) w.r.t. the jth label space
f_a^j (or h_a^j)	the part of MDML predictive model f (or h) w.r.t. the ath label in the jth label space
$\mathcal{T}(\cdot)$	the threshold function where $f(\cdot) = \mathcal{T}(h(\cdot))$
$\boldsymbol{x}_*$	the unseen instance where $\boldsymbol{x}_* \in \mathcal{X}$
$\hat{\boldsymbol{l}}_*$	the predicted label vector for $\boldsymbol{x}_*$, i.e., $\hat{\boldsymbol{l}}_* = f(\boldsymbol{x}_*) \in \{0,1\}^K$

3.2. Benchmark Approaches

Motivated by the binary relevance strategy in multi-label classification and multi-dimensional classification, we can also solve the MDML problem via *dimension decomposition* strategy where a multi-label classifier is induced for each label space independently. Without loss of generality, a multi-label data set $\mathcal{D}^j = \{(\boldsymbol{x}_i, \boldsymbol{l}_i^j) \mid 1 \leq i \leq m\}$ can be constructed based on the MDML training set $\mathcal{D}$ for the jth label space $\mathcal{Y}^j$, and a multi-label classifier f^j can be trained over it, i.e., $f^j \leftarrow\!\shortmid \mathcal{M}(\mathcal{D}^j)$, where $\mathcal{M}$ is the employed multi-label classification algorithm. In testing phase, the predicted label vector $\hat{\boldsymbol{l}}_*$ of unseen instance $\boldsymbol{x}_*$ can be obtained by concatenating the predictions of $f^1, f^2, \ldots, f^q$,

i.e., $\hat{\boldsymbol{l}}_* = [f^1(\boldsymbol{x}_*); f^2(\boldsymbol{x}_*); \ldots; f^q(\boldsymbol{x}_*)]$. For brevity, this baseline is denoted as DiDe (i.e., **Di**mension **De**composition) in the following parts of this paper.

Moreover, motivated by the relationships between multi-label classification and MDML, we can also solve the MDML problem via *dimension concatenation* strategy where labels in different label spaces are no longer discriminated. In other words, this strategy simply aligns all labels in output space into a homogeneous label space and ignores the multi-dimensional nature in the MDML framework. Specifically, given a multi-label classification algorithm $\mathcal{M}$, the MDML model f is just learned over the MDML training set $\mathcal{D}$ with $\mathcal{M}$, i.e., $f \hookleftarrow \mathcal{M}(\mathcal{D})$. The predicted label vector $\hat{\boldsymbol{l}}_*$ of unseen instance $\boldsymbol{x}_*$ can be obtained via $\hat{\boldsymbol{l}}_* = f(\boldsymbol{x}_*)$. For brevity, this baseline is denoted as DiCo (i.e., **Di**mension **Co**ncatenation) in the following parts of this paper.

It is worth noting that DiDe deals with each label space in an independent way, which leads to the issue of ignoring label correlations across different dimensions. On the other hand, DiCo treats class labels for all label spaces in a homogeneous way, which leads to the issue of ignoring the heterogeneity nature of different dimensions.

3.3. The Proposed Approach

In this section, we present technical details of the proposed CLIM approach which works in an augmented stacking strategy. Based on the fact that modeling outputs of labels belonging to the same label space are comparable, CLIM chooses to consider the label correlations within individual dimension by maximizing the likelihood of relevant labels. Motivated by the feature augmentation strategy in multi-dimensional classification [36, 35, 7], CLIM chooses to consider the label correlations across multiple dimensions by manipulating feature space, which is good at dealing with examples with heterogeneous label spaces and does not need to compare modeling outputs of labels. Compared with DiDe, CLIM considers the label correlations not only within individual dimension, but also across multiple dimensions. Compared with DiCo, CLIM does not align labels from heterogeneous label spaces into a homogeneous one, but still considers the

label correlations across multiple dimensions via feature augmentation strategy. Comparative studies in Section 4 will show that the technical designs make CLIM achieve very superior performance against DiDe and DiCo.

Figure 2 shows the workflow of the proposed CLIM approach, where each kind of colors denotes one working phase. In the first phase (blue color), CLIM induces predictive model for each dimension over the original feature space to initialize the augmented features. In the second phase (red color), CLIM induces predictive model for each dimension again over the augmented feature space. In the third phase (green color), CLIM updates the predictions that are used to augment the feature space in the second phase if specific conditions are satisfied. CLIM works in the second phase and third phase alternately until convergence.

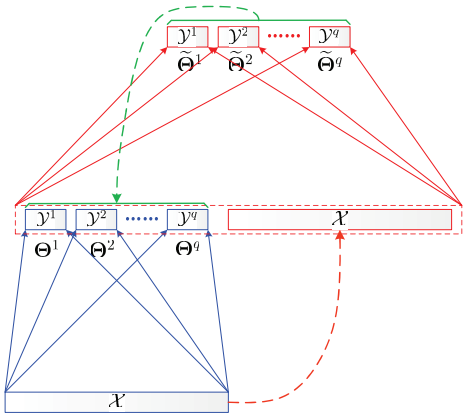


Figure 2: The workflow of the proposed CLIM approach.

3.3.1. Predictive Model Induction for Each Dimension

Without loss of generality, for the jth label space, let $\boldsymbol{\Theta}^j = [\boldsymbol{\theta}_1^j, \boldsymbol{\theta}_2^j, \ldots, \boldsymbol{\theta}_{K_j}^j] \in \mathbb{R}^{d \times K_j}$ be the model parameters of hypothesis $h^j(\cdot)$, which can return a K_j-dimensional probability estimation vector for instance $\boldsymbol{x}_i$ as follows:

$$h^j(\boldsymbol{x}_i) = \begin{bmatrix} P(y_1^j \mid \boldsymbol{x}_i, \boldsymbol{\Theta}^j) \\ P(y_2^j \mid \boldsymbol{x}_i, \boldsymbol{\Theta}^j) \\ \vdots \\ P(y_{K_j}^j \mid \boldsymbol{x}_i, \boldsymbol{\Theta}^j) \end{bmatrix} = \frac{1}{\sum_{s=1}^{K_j} e^{\langle \boldsymbol{\theta}_s^j, \boldsymbol{x}_i \rangle}} \begin{bmatrix} e^{\langle \boldsymbol{\theta}_1^j, \boldsymbol{x}_i \rangle} \\ e^{\langle \boldsymbol{\theta}_2^j, \boldsymbol{x}_i \rangle} \\ \vdots \\ e^{\langle \boldsymbol{\theta}_{K_j}^j, \boldsymbol{x}_i \rangle} \end{bmatrix} \tag{1}$$

where $\langle \cdot, \cdot \rangle$ computes the inner product of two vectors. We hope that $\boldsymbol{\Theta}^j$ can make the probabilities of relevant labels as large as possible. Therefore, we use the maximum likelihood estimation (MLE) method to determine the model parameters $\boldsymbol{\Theta}^j$. The likelihood function can be given as follows:

$$L(\boldsymbol{\Theta}^j) = \prod_{i=1}^{m} \prod_{a=1}^{K_j} \left[P(y_a^j \mid \boldsymbol{x}_i, \boldsymbol{\Theta}^j) \right]^{l_{ia}^j} \tag{2}$$

Note that $l_{ia}^j = 1$ if y_a^j is a relevant label of $\boldsymbol{x}_i$ and $l_{ia}^j = 0$ if y_a^j is an irrelevant label of $\boldsymbol{x}_i$. Therefore, only the probabilities w.r.t. relevant labels will function in Eq.(2). Instead of the likelihood function, we usually optimize the log-likelihood function which can be given as follows:

$$LL(\boldsymbol{\Theta}^j) = \ln L(\boldsymbol{\Theta}^j) = \sum_{i=1}^{m} \sum_{a=1}^{K_j} l_{ia}^j \cdot \ln P(y_a^j \mid \boldsymbol{x}_i, \boldsymbol{\Theta}^j) \tag{3}$$

According to MLE, the value of $\boldsymbol{\Theta}^j$ can be determined via $\max_{\boldsymbol{\Theta}^j} LL(\boldsymbol{\Theta}^j)$, which can be equivalently formulated as $\min_{\boldsymbol{\Theta}^j} -LL(\boldsymbol{\Theta}^j)$. Moreover, we penalize the complexity of $\boldsymbol{\Theta}^j$ by adding a regularization term in objective function to avoid overfitting and then obtain the following optimization problem:

$$\min_{\boldsymbol{\Theta}^j} \quad -\sum_{i=1}^{m} \sum_{a=1}^{K_j} l_{ia}^j \cdot \ln P(y_a^j \mid \boldsymbol{x}_i, \boldsymbol{\Theta}^j) + \frac{\lambda}{2} \left\| \boldsymbol{\Theta}^j \right\|_F^2 \tag{4}$$

where λ is the regularization parameter and $\|\cdot\|_F$ computes the Frobenius norm of matrices. Because there is not an analytical solution to problem (4), we use gradient descent to solve it in this paper. Specifically, let's denote the objective function by $J(\boldsymbol{\Theta}^j)$, the gradient w.r.t. $\boldsymbol{\theta}_t^j$ can be calculated as follows:

$$\frac{\partial J(\boldsymbol{\Theta}^j)}{\partial \boldsymbol{\theta}_t^j} = -\sum_{i=1}^{m} \sum_{a=1}^{K_j} l_{ia}^j \boldsymbol{x}_i \cdot \big(\mathbb{I}(t = a) - P(y_a^j \mid \boldsymbol{x}_i, \boldsymbol{\Theta}^j) \big) + \lambda \boldsymbol{\theta}_t^j, \quad (1 \leq t \leq K_j) \tag{5}$$

where $\mathbb{I}(\cdot)$ is the indicator function, i.e., $\mathbb{I}(\pi)$ returns 1 if predicate π holds and 0 otherwise.

It is worth noting that we normalize the distribution by $\sum_{s=1}^{K_j} e^{\langle \boldsymbol{\theta}_s^j, \boldsymbol{x}_i \rangle}$ in Eq.(1) to make the sum of the K_j probabilities be equal to one. This operation also makes the model parameters $\boldsymbol{\Theta}^j$ overparameterized. Therefore, we always

fix $\boldsymbol{\theta}_{K_j}^j$ as $\mathbf{0}$ and only optimize $\boldsymbol{\theta}_1^j, \boldsymbol{\theta}_2^j, \ldots, \boldsymbol{\theta}_{K_j-1}^j$ when implementing the above optimizing procedure.

With the obtained $\boldsymbol{\Theta}^j$ via optimizing problem (4), the modeling outputs of any instance can be determined via Eq.(1), where the corresponding modeling output of one label represents its probability of being relevant to the instance. The larger the modeling output is, the more likely the label is relevant to the instance. To further distinguish relevant and irrelevant labels according to their real-valued modeling outputs, CLIM needs to determine one threshold to decide whether one label is relevant or irrelevant. In this paper, the threshold is determined with maximum a posteriori (MAP) criterion in a dimension-wise manner. Specifically, CLIM searches in the candidate threshold set and selects the value which can minimize the metric value of *hamming loss* (cf. Section 4.1.2) over training set as the final threshold (denoted by T^j). Here, *hamming loss* is used as it is one of the simplest classification-based multi-label evaluation metrics and other classification-based metrics can also be investigated in the future. Algorithm 1 presents the complete procedure of MAP-based threshold determination for the jth dimension, where $\boldsymbol{\eta}(i)$, $vec\mathbf{Y}_s(i)$ denote the ith element of $\boldsymbol{\eta}$, $vec\mathbf{Y}_s$ and ϵ denotes any positive value. Specifically, we firstly initialize $\boldsymbol{\eta}(1)$ as the number of labels that are predicted correctly when the threshold is set so small that all labels are predicted to be relevant, i.e., the number of ones in $\mathbf{Y}$. Then, we gradually raise the threshold to predict the corresponding items in the sorted vector as irrelevant labels in turn. It is easy to know that we will make wrong (or correct) prediction if $vec\mathbf{Y}_s(i)$ is equal to 1 (or 0). Finally, we determine the threshold according to the index $\bar{i}$ of the largest value in $\boldsymbol{\eta}$.

With the threshold, we can obtain the multi-label classifier $f^j(\cdot)$ which returns binary predictions for any instance $\boldsymbol{x}$ based on the learned hypothesis $h^j(\cdot)$ for the jth dimension:

$$f_a^j(\boldsymbol{x}) = \mathbb{I}(h_a^j(\boldsymbol{x}) > T^j), \quad (1 \leq a \leq K_j) \tag{6}$$

Algorithm 1 MAP-based threshold determination for the jth dimension

Input: The ground-truth label matrix $\mathbf{Y} \in \{0,1\}^{m \times K_j}$, the real-valued modeling outputs matrix $\mathbf{Y}_r \in \mathbb{R}^{m \times K_j}$;

Output: The determined threshold T^j;

Vectorize $\mathbf{Y}$, $\mathbf{Y}_r$ as $vec\mathbf{Y}$, $vec\mathbf{Y}_r$ with length $(m \times K_j)$;
Sort the items of $vec\mathbf{Y}_r$ in ascending order;
Rearrange the items of $vec\mathbf{Y}$ according to the ascending order of $vec\mathbf{Y}_r$ and denote the sorted vector as $vec\mathbf{Y}_s$;
Initialize $\boldsymbol{\eta} = \mathbf{0}$ with length $(m \times K_j + 1)$;
Set $\boldsymbol{\eta}(1) = \sum_{i=1}^{m \times K_j} vec\mathbf{Y}_s(i)$;
for $i = 1$ to $(m \times K_j)$ **do**
  if $vec\mathbf{Y}_s(i)$ is equal to 1 **then**
    $\boldsymbol{\eta}(i+1) = \boldsymbol{\eta}(i) - 1$; %*wrong prediction*
  else
    $\boldsymbol{\eta}(i+1) = \boldsymbol{\eta}(i) + 1$; %*correct prediction*
  end if
end for
Determine the index $\bar{i}$ of the largest value in $\boldsymbol{\eta}$;
if $\bar{i}$ is equal to 1 **then**
  Return $T^j = vec\mathbf{Y}_s(\bar{i}) - \epsilon$;
else if $\bar{i}$ is equal to $(m \times K_j + 1)$ **then**
  Return $T^j = vec\mathbf{Y}_s(\bar{i}) + \epsilon$;
else
  Return $T^j = \frac{vec\mathbf{Y}_s(\bar{i}-1) + vec\mathbf{Y}_s(\bar{i})}{2}$;
end if

3.3.2. Feature Augmentation and Predictive Model Updating

After traversing all dimensions, we can obtain the multi-dimensional multi-label classifier $f(\cdot)$ which returns binary predicted vector for any instance $\boldsymbol{x}$. To consider the label correlations across different dimensions, CLIM further manipulates the feature space via feature augmentation mechanism [36, 35,

7]. Specifically, the MDML training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{l}_i) \mid 1 \leq i \leq m\}$ can be transformed into the following data set $\widetilde{\mathcal{D}}$:

$$\widetilde{\mathcal{D}} = \{(\widetilde{\boldsymbol{x}}_i, \boldsymbol{l}_i) \mid 1 \leq i \leq m\}, \text{ where } \widetilde{\boldsymbol{x}}_i = [\boldsymbol{x}_i; f(\boldsymbol{x}_i)] \tag{7}$$

Here, each instance $\widetilde{\boldsymbol{x}}_i$ belongs to the augmented feature space (denoted by $\widetilde{\mathcal{X}}$) which corresponds to the Cartesian product between $\mathcal{X}$ and a K-dimensional feature space.

Similar to Eq.(1) to Eq.(5), based on the transformed data set $\widetilde{\mathcal{D}}$, we can also determine the model parameters $\widetilde{\boldsymbol{\Theta}}^j = [\widetilde{\boldsymbol{\theta}}_1^j, \widetilde{\boldsymbol{\theta}}_2^j, \ldots, \widetilde{\boldsymbol{\theta}}_{K_j}^j] \in \mathbb{R}^{(d+K)\times K_j}$ of hypothesis $h^j(\cdot)$ for the jth label space, which can return a K_j-dimensional probability estimation vector for instance $\boldsymbol{x}_i$ as follows:[2]

$$h^j(\boldsymbol{x}_i) = \begin{bmatrix} P(y_1^j \mid \widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{\Theta}}^j) \\ P(y_2^j \mid \widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{\Theta}}^j) \\ \vdots \\ P(y_{K_j}^j \mid \widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{\Theta}}^j) \end{bmatrix} = \frac{1}{\sum_{s=1}^{K_j} e^{\langle \widetilde{\boldsymbol{\theta}}_s^j, \widetilde{\boldsymbol{x}}_i \rangle}} \begin{bmatrix} e^{\langle \widetilde{\boldsymbol{\theta}}_1^j, \widetilde{\boldsymbol{x}}_i \rangle} \\ e^{\langle \widetilde{\boldsymbol{\theta}}_2^j, \widetilde{\boldsymbol{x}}_i \rangle} \\ \vdots \\ e^{\langle \widetilde{\boldsymbol{\theta}}_{K_j}^j, \widetilde{\boldsymbol{x}}_i \rangle} \end{bmatrix} \tag{8}$$

With the newly obtained $h^j(\cdot)$, if it achieves better performance in terms of *ranking loss* (cf. Section 4.1.2) over training set, then the threshold T^j will be re-calculated and the multi-label classifier $f^j(\cdot)$ will be updated accordingly, which will be used to generate $\widetilde{\mathcal{D}}$. Otherwise, the part of augmented features w.r.t. the jth label space won't be updated when generating $\widetilde{\mathcal{D}}$. The model parameters $\widetilde{\boldsymbol{\Theta}}^j$ will be updated until $\widetilde{\mathcal{D}}$ keeps unchanged. Here, we use the ranking-based multi-label evaluation metric *ranking loss* (cf. Section 4.1.2) to test updating condition and other ranking-based metrics besides *ranking loss* can also be investigated in the future. The reason is that the proposed CLIM approach aims at optimizing the likelihood of relevant labels and ranking-based metrics is more sensitive to model parameters than classification-based metrics. Take the following example for an intuition, assume that the ground-truth label vector is [1, 0, 1, 1, 0, 0], two

[2]For convenience, we slightly abuse the notations in Eq.(8), where the hypothesis is also denoted by $h^j(\cdot)$ even though it is different from the one in Eq.(1).

sets of slightly different model parameters respectively return the corresponding modeling outputs [0.2014, 0.2172, 0.1775, 0.1571, 0.1637, 0.0831] and [0.2172, 0.2014, 0.1775, 0.1637, 0.1571, 0.0831], the threshold is fixed as 0.1667, it is easy to know that the two modeling outputs correspond to the same binary prediction [1, 1, 1, 0, 0, 0], which means that the two modeling outputs will result in the same value w.r.t. classification-based metrics (e.g., *hamming loss*). However, because the two modeling outputs correspond to different ranks for relevant labels and irrelevant labels, they will result in different values w.r.t. ranking-based metrics (e.g., *ranking loss*). In other words, ranking-based metrics is more sensitive to model parameters than classification-based metrics.

4. Experiments

To validate the effectiveness of the proposed CLIM approach, comparative studies are conducted in this section. Specifically, Subsection 4.1 firstly introduces the experimental setup, including benchmark data sets, evaluation metrics, and compared approaches. Then, Subsection 4.2 reports the detailed experimental results with corresponding analyses. Finally, Subsection 4.3 further investigates some algorithmic properties of CLIM.

4.1. Experimental Setup

4.1.1. Benchmark Data Sets

In this paper, we have collected four benchmark data sets from real-world MDML tasks to compare the performance of CLIM and some baselines. The detailed characteristics of each dimension w.r.t. these four benchmark data sets are summarized in Table 2 and Table 3 respectively, including *number of examples* (#Examples), *number of features* (#Features), *number of dimensions* (#Dim.), and *number of labels in each dimension* (#Labels). In addition, as each label space corresponds to a multi-label classification problem, the two tables also show the multi-label properties of each dimension characterized by several statistics, including *label cardinality* (LCard), *label density* (LDen), *distinct label*

sets (DL) and *proportion of distinct label sets* (PDL). The detailed definitions of these statistics can be easily found in multi-label literatures [17, 25], and thus we omit them in this paper.

The two `Song` data sets are collected from music domain and annotated by the authors' research group. In these two data sets, each example corresponds to one Chinese song and each dimension corresponds to one kind of semantics. Specifically, the three dimensions correspond to emotion, genre, and scenario, respectively. After listening to a song, we (eight annotators) assign a confidence value to each label for this song and the average of our assignments will be used as the final confidence value for each label. For `Song-v1`, the label will be regarded to be relevant to one instance when its confidence value is larger than the average value within this dimension. For `Song-v2`, assume that labels of one instance in the same dimension are sorted in descending order according to their confidence values, here we just focus on those labels whose confidence values are larger than the average value within this dimension, the label will be regarded as a relevant one if it is in the front of the position of the largest difference between two adjacent labels.

The two `Yeast` data sets are collected from biology domain. In these two data sets, each example corresponds to one yeast gene and each dimension corresponds to one biological experiment on the budding yeast *Saccharomyces cerevisiae* [37]. Specifically, the six dimensions correspond to alpha factor arrest & release, cdc15 arrest & release, elutriation, diauxic shift, heat shock, and sporulation, respectively.[3] In each dimension, each label corresponds to one discrete time point during the biological experiment. For `Yeast-v1`, the time point will be regarded as a relevant label when the current gene expression level (after normalization) is larger than the average level in the biological experiment. For `Yeast-v2`, assume that time points during the same biological experiment are sorted in descending order according to their gene expression levels, here we just focus on those time points whose gene expression levels are larger than the

[3]Supplementary material at `https://www.pnas.org/doi/10.1073/pnas.95.25.14863`

average value in the biological experiment, the time point will be regarded as a relevant label if it is in the front of the position of the largest difference between two adjacent time points.

Table 2: The data sets `Song-v1` and `Song-v2` collected from Chinese song categorization.

Dim.	#Examples	#Features	#Label	LCard(v1/v2)	LDen(v1/v2)	DL(v1/v2)	PDL(v1/v2)	Domain
Dim.1	785	98	11	3.524/1.738	0.320/0.158	204/105	0.260/0.134	music
Dim.2			10	2.270/1.333	0.227/0.133	89/46	0.113/0.059	
Dim.3			18	8.505/4.084	0.472/0.227	381/269	0.485/0.343	

Table 3: The data sets `Yeast-v1` and `Yeast-v2` collected from biological experiments on the budding yeast *Saccharomyces cerevisiae*.

Dim.	#Examples	#Features	#Label	LCard(v1/v2)	LDen(v1/v2)	DL(v1/v2)	PDL(v1/v2)	Domain
Dim.1	2465	24	18	8.904/2.977	0.495/0.165	2264/862	0.918/0.350	biology
Dim.2			15	7.601/3.205	0.507/0.214	1662/750	0.674/0.304	
Dim.3			14	7.167/3.476	0.512/0.248	1519/838	0.616/0.340	
Dim.4			7	3.527/2.572	0.504/0.367	95/91	0.039/0.037	
Dim.5			6	3.014/2.089	0.502/0.348	58/60	0.024/0.024	
Dim.6			6	2.867/2.103	0.478/0.350	58/59	0.024/0.024	

4.1.2. Evaluation Metrics

To measure the generalization performance of MDML approaches, we evaluate their performance w.r.t. each dimension as well as their average performance over all dimensions. Because each dimension in the MDML problem corresponds to a multi-label classification task, a total of six multi-label evaluation metrics are utilized in this paper, including four ranking-based metrics (*ranking loss*, *coverage*, *one-error*, *average precision*) and two classification-based metrics (*hamming loss*, *macro*-F1). Specifically, following the notations defined in previous sections, given the MDML test set $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{l}_i) \mid 1 \leq i \leq p\}$ with p examples, and the MDML predictive models $h(\cdot)$ and $f(\cdot)$ to be evaluated, for the jth dimension $(1 \leq j \leq q)$, these six evaluation metrics can be formulated as follows [21, 38, 39]:

- *Ranking loss*: $\frac{1}{p}\sum_{i=1}^{p}\frac{|\mathcal{Z}_i^j|}{r_i^j \times \bar{r}_i^j}$, where $\mathcal{Z}_i^j = \{(y_{a_1}^j, y_{a_2}^j) \mid 1 \leq a_1, a_2 \leq K_j,\ h_{a_1}^j(\boldsymbol{x}_i) \leq h_{a_2}^j(\boldsymbol{x}_i),\ l_{ia_1}^j = 1,\ l_{ia_2}^j = 0\}$, $r_i^j = \sum_{a=1}^{K_j} l_{ia}^j$ and $\bar{r}_i^j = \sum_{a=1}^{K_j}(1 - l_{ia}^j)$.

- *Coverage*: $\frac{1}{K_j}\left[\frac{1}{p}\sum_{i=1}^{p}\max_{k\in\mathcal{K}_i^j} rank(\boldsymbol{x}_i, y_k^j) - 1\right]$, where $rank(\boldsymbol{x}_i, y_k^j) = \sum_{a=1}^{K_j}\mathbb{I}(h_a^j(\boldsymbol{x}_i) \geq h_k^j(\boldsymbol{x}_i))$ and $\mathcal{K}_i^j = \{k \mid l_{ik}^j = 1,\ 1 \leq k \leq K_j\}$.

- *One-error*: $\frac{1}{p}\sum_{i=1}^{p}\mathbb{I}(l_{i\hat{a}}^j \neq 1)$, where $\hat{a} = \arg\max_{1\leq a\leq K_j} h_a^j(\boldsymbol{x}_i)$.

- *Average precision*: $\frac{1}{p}\sum_{i=1}^{p}\frac{1}{r_i^j}\sum_{k\in\mathcal{K}_i^j}\frac{|\mathcal{R}(\boldsymbol{x}_i, y_k^j)|}{rank(\boldsymbol{x}_i, y_k^j)}$, where $\mathcal{R}(\boldsymbol{x}_i, y_k^j) = \{y_a^j \mid h_a^j(\boldsymbol{x}_i) \geq h_k^j(\boldsymbol{x}_i), a \in \mathcal{K}_i^j\}$.

- *Hamming loss*: $\frac{1}{p}\sum_{i=1}^{p}\frac{1}{K_j}\sum_{a=1}^{K_j} l_{ia}^j \times f_a^j(\boldsymbol{x}_i)$.

- *Macro*-F1: $\frac{1}{K_j}\sum_{a=1}^{K_j}\frac{2\sum_{i=1}^{p} l_{ia}^j \times f_a^j(\boldsymbol{x}_i)}{\sum_{i=1}^{p} l_{ia}^j + \sum_{i=1}^{p} f_a^j(\boldsymbol{x}_i)}$.

Based on the above definitions, the average value over all dimensions for each evaluation metric corresponds to:

- *Average metric value*: $\frac{1}{q}\sum_{j=1}^{q} M(j)$, where $M(j)$ denotes the metric value w.r.t. the jth dimension (e.g., *average ranking loss*, *average coverage*).

It is easy to know that all these evaluation metrics take values in $[0, 1]$. For ranking loss, coverage, one-error, hamming loss, the *smaller* the metric value, the better the performance, while for average precision and macro-F1, the *larger* the metric value, the better the performance. In experiments, we conduct ten-fold cross validation over each data set and record both mean value and standard deviation in terms of each evaluation metric.

4.1.3. Compared Approaches

As a new learning framework, there aren't existing MDML approaches which can be used as baselines to make comparisons. In this paper, the performance of the proposed CLIM approach is compared with the two benchmark approaches (i.e., DiDe and DiCo) proposed in Subsection 3.2 as well as three multi-label classification baselines, including Binary Relevance (BR) [16], Classifier Chains (CC) [17, 18] and WRAPping multi-label classification with label-specific features generation (WRAP) [24].

Technical details of DiDe and DiCo can be found in Subsection 3.2. For fair comparison, the multi-label problems in DiDe and DiCo are also solved by

maximizing the likelihood of relevant labels similar to CLIM, which makes the two benchmark approaches act as degenerated versions of CLIM. For the three multi-label classification approaches, we simply regard the MDML problem as a vanilla multi-label classification problem by concatenating all label spaces as an entirety. Specifically, BR independently learns a binary classifier for each label and then all possible label correlations are ignored. CC learns a chain of binary classifiers, one per label, where the subsequent classifiers on the chain will use the labeling information for training previous classifiers as extra features. WRAP is a recently proposed multi-label approach which wraps the multi-label model induction and label-specific features generation in a unified formulation. For fair comparison, the binary classifiers in BR and CC are trained with logistic regression which is implemented by the popular the LIBLINEAR package [40]. For WRAP, its parameters are set according to the suggestions in the original literature [24], i.e., $\alpha = 0.9$, $\lambda_1 = 5$, $\lambda_2 = 5$, $\lambda_3 = 0.1$. For the proposed CLIM approach, the only parameter λ is set to 2^{-3}.

4.2. Experimental Results

The detailed experimental results in terms of each evaluation metric over the four data sets are reported in Tables 4-7, respectively. In this paper, both the experimental results of each dimension and the average experimental result of all dimensions are reported and the best performance is shown in bold face. Besides, the performance ranks of all compared approaches are also shown in parentheses for clearer comparison.

According to the reported experimental results, we can have the following observations:

- Compared with DiDe and DiCo, it is shown that the proposed CLIM approach achieves superior average performance in terms of each metric over all data sets. DiDe ignores the label correlations across dimensions while DiCo violates the multi-dimensional assumption in MDML. In other words, the comparison between CLIM and these two benchmark approaches

Table 4: Experimental results (mean±std. deviation) of CLIM and each compared approach over data set `Song-v1`. In addition, the performance ranks of all compared approaches are also shown in parentheses and the best performance is in bold. ↓ (↑) behind the name of each evaluation metric indicates that the smaller (larger) the value, the better the performance.

Dim. of Song-v1	Ranking loss (↓)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.216±0.015(2)	0.241±0.018(5)	0.240±0.018(4)	0.222±0.051(3)	**0.214±0.046(1)**	0.258±0.020(6)
Dim2	0.111±0.013(4)	0.110±0.012(2)	**0.109±0.012(1)**	0.281±0.024(5)	0.284±0.025(6)	0.110±0.014(2)
Dim3	**0.086±0.009(1)**	0.098±0.009(2)	0.098±0.009(2)	0.119±0.024(6)	0.101±0.021(4)	0.105±0.009(5)
Avg.	**0.138±0.008(1)**	0.150±0.010(3)	0.149±0.010(2)	0.207±0.025(6)	0.200±0.019(5)	0.157±0.011(4)
Dim. of Song-v1	Coverage (↓)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.504±0.014(3)	0.560±0.017(5)	0.557±0.016(4)	0.489±0.043(2)	**0.476±0.036(1)**	0.576±0.016(6)
Dim2	0.294±0.021(3)	0.295±0.020(4)	0.293±0.021(2)	0.464±0.028(5)	0.468±0.029(6)	**0.292±0.023(1)**
Dim3	0.605±0.016(2)	0.631±0.021(5)	0.631±0.021(5)	0.616±0.023(3)	**0.602±0.020(1)**	0.624±0.018(4)
Avg.	**0.468±0.011(1)**	0.495±0.014(3)	0.493±0.013(2)	0.523±0.021(6)	0.515±0.015(5)	0.497±0.014(4)
Dim. of Song-v1	One error (↓)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.237±0.049(3)	**0.234±0.040(1)**	**0.234±0.050(1)**	0.379±0.261(6)	0.367±0.233(5)	0.257±0.055(4)
Dim2	**0.020±0.012(1)**	**0.020±0.012(1)**	**0.020±0.012(1)**	0.464±0.098(5)	0.466±0.093(6)	**0.020±0.012(1)**
Dim3	**0.004±0.006(1)**	0.006±0.007(2)	0.006±0.007(2)	0.010±0.010(4)	0.011±0.013(5)	0.025±0.015(6)
Avg.	**0.087±0.016(1)**	**0.087±0.012(1)**	**0.087±0.016(1)**	0.284±0.101(6)	0.282±0.085(5)	0.101±0.018(4)
Dim. of Song-v1	Average precision (↑)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.729±0.024(1)**	0.711±0.023(2)	0.711±0.025(2)	0.699±0.085(5)	0.711±0.077(2)	0.692±0.026(6)
Dim2	0.853±0.015(3)	**0.855±0.013(1)**	0.854±0.013(2)	0.584±0.033(5)	0.582±0.032(6)	0.848±0.017(4)
Dim3	**0.920±0.008(1)**	0.912±0.008(2)	0.912±0.008(2)	0.879±0.026(6)	0.898±0.026(5)	0.899±0.009(4)
Avg.	**0.834±0.011(1)**	0.826±0.010(2)	0.826±0.011(2)	0.721±0.037(6)	0.730±0.026(5)	0.813±0.012(4)
Dim. of Song-v1	Hamming loss (↓)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.235±0.015(1)**	0.245±0.013(4)	0.239±0.014(3)	0.246±0.034(5)	0.237±0.031(2)	0.255±0.015(6)
Dim2	0.128±0.011(2)	0.129±0.010(3)	**0.127±0.011(1)**	0.166±0.013(5)	0.168±0.012(6)	0.131±0.008(4)
Dim3	0.164±0.010(3)	0.175±0.011(5)	0.174±0.011(4)	**0.157±0.012(1)**	0.158±0.013(2)	0.182±0.013(6)
Avg.	**0.176±0.007(1)**	0.183±0.007(3)	0.180±0.006(2)	0.190±0.014(6)	0.187±0.013(4)	0.189±0.006(5)
Dim. of Song-v1	Macro-F1 (↑)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.378±0.031(3)	0.320±0.027(5)	0.356±0.024(4)	0.471±0.031(2)	**0.520±0.028(1)**	0.249±0.015(6)
Dim2	0.214±0.031(4)	0.218±0.022(3)	0.213±0.020(5)	0.260±0.027(2)	**0.263±0.028(1)**	0.101±0.007(6)
Dim3	0.502±0.012(3)	0.443±0.014(5)	0.442±0.009(6)	0.605±0.023(2)	**0.613±0.017(1)**	0.444±0.012(4)
Avg.	0.365±0.013(3)	0.327±0.010(5)	0.337±0.009(4)	0.445±0.019(2)	**0.465±0.016(1)**	0.265±0.006(6)

actually act as ablation studies and the experimental results clearly validate that either ignoring label correlations among multiple dimensions or aligning labels from heterogeneous label spaces into a homogeneous one will lead to performance degeneration.

- Compared with BR and CC, it is shown that the proposed CLIM approach achieves superior average performance in terms of the four ranking-based metrics and *hamming loss* over all data sets. These experimental results show the superiority of CLIM against the two compared approaches. It is

Table 5: Experimental results (mean±std. deviation) of CLIM and each compared approach over data set `Yeast-v1`. In addition, the performance ranks of all compared approaches are also shown in parentheses and the best performance is in bold. ↓ (↑) behind the name of each evaluation metric indicates that the smaller (larger) the value, the better the performance.

Dim. of	Ranking loss (↓)					
Yeast-v1	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.389±0.016(1)**	0.392±0.015(2)	0.392±0.015(2)	0.396±0.023(4)	0.404±0.024(6)	0.399±0.015(5)
Dim2	**0.393±0.007(1)**	0.403±0.014(3)	0.400±0.013(2)	0.415±0.017(6)	0.412±0.016(5)	0.404±0.013(4)
Dim3	**0.378±0.012(1)**	0.387±0.010(2)	0.387±0.009(2)	0.387±0.022(2)	0.388±0.019(5)	0.396±0.009(6)
Dim4	0.278±0.011(3)	0.291±0.010(5)	0.283±0.009(4)	0.277±0.008(2)	**0.276±0.009(1)**	0.300±0.009(6)
Dim5	0.406±0.019(2)	0.421±0.021(5)	0.414±0.020(3)	**0.404±0.021(1)**	0.414±0.020(3)	0.447±0.018(6)
Dim6	**0.389±0.024(1)**	0.391±0.026(3)	0.399±0.025(6)	0.391±0.027(3)	0.396±0.024(5)	0.390±0.026(2)
Avg.	**0.372±0.009(1)**	0.381±0.009(4)	0.379±0.009(3)	0.378±0.011(2)	0.382±0.009(5)	0.389±0.008(6)
Dim. of	Coverage (↓)					
Yeast-v1	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.837±0.007(1)**	0.841±0.006(3)	0.840±0.006(2)	0.842±0.009(4)	0.844±0.011(5)	0.844±0.006(5)
Dim2	0.830±0.005(3)	0.837±0.008(5)	0.835±0.008(4)	**0.827±0.010(1)**	0.829±0.006(2)	0.841±0.007(6)
Dim3	**0.803±0.012(1)**	0.815±0.011(4)	0.815±0.011(4)	0.811±0.011(2)	0.812±0.010(3)	0.818±0.010(6)
Dim4	**0.611±0.009(1)**	0.621±0.008(5)	0.616±0.007(4)	0.612±0.008(2)	0.612±0.008(2)	0.625±0.008(6)
Dim5	0.650±0.012(3)	0.672±0.013(5)	0.665±0.015(4)	**0.643±0.015(1)**	0.649±0.015(2)	0.698±0.012(6)
Dim6	0.568±0.015(3)	0.567±0.016(2)	0.569±0.015(4)	0.574±0.016(5)	0.582±0.013(6)	**0.566±0.016(1)**
Avg.	**0.717±0.005(1)**	0.725±0.005(5)	0.724±0.005(4)	0.718±0.006(2)	0.721±0.005(3)	0.732±0.004(6)
Dim. of	One error (↓)					
Yeast-v1	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.318±0.028(1)**	0.321±0.035(2)	0.321±0.035(2)	0.371±0.133(5)	0.401±0.119(6)	0.340±0.041(4)
Dim2	**0.314±0.033(1)**	0.315±0.034(2)	0.315±0.034(2)	0.328±0.024(5)	0.335±0.030(6)	0.315±0.034(2)
Dim3	0.333±0.028(4)	**0.316±0.026(1)**	**0.316±0.026(1)**	0.334±0.038(5)	0.338±0.041(6)	**0.316±0.026(1)**
Dim4	**0.196±0.034(1)**	**0.196±0.031(1)**	**0.196±0.031(1)**	0.201±0.032(6)	0.199±0.034(5)	**0.196±0.031(1)**
Dim5	**0.376±0.036(1)**	0.391±0.033(3)	0.391±0.033(3)	0.382±0.024(2)	0.392±0.024(6)	0.391±0.033(3)
Dim6	0.387±0.031(3)	0.386±0.024(2)	0.441±0.027(6)	0.392±0.032(4)	0.398±0.044(5)	**0.381±0.021(1)**
Avg.	**0.321±0.013(1)**	**0.321±0.014(1)**	0.330±0.012(4)	0.335±0.018(5)	0.344±0.017(6)	0.323±0.014(3)
Dim. of	Average precision (↑)					
Yeast-v1	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.656±0.013(1)**	0.654±0.014(3)	0.655±0.014(2)	0.647±0.027(5)	0.639±0.027(6)	0.649±0.014(4)
Dim2	**0.675±0.008(1)**	0.669±0.010(3)	0.670±0.010(2)	0.658±0.014(6)	0.660±0.011(5)	0.668±0.010(4)
Dim3	**0.689±0.009(1)**	0.683±0.007(2)	0.683±0.007(2)	0.683±0.020(2)	0.681±0.017(5)	0.677±0.008(6)
Dim4	0.786±0.014(3)	0.781±0.013(5)	0.783±0.013(4)	0.788±0.010(2)	**0.789±0.012(1)**	0.772±0.013(6)
Dim5	0.707±0.015(2)	0.694±0.015(5)	0.698±0.015(4)	**0.711±0.013(1)**	0.704±0.013(3)	0.682±0.015(6)
Dim6	0.738±0.012(3)	0.740±0.013(2)	0.718±0.013(6)	0.731±0.013(4)	0.725±0.012(5)	**0.742±0.011(1)**
Avg.	**0.709±0.006(1)**	0.704±0.006(2)	0.701±0.006(4)	0.703±0.007(3)	0.700±0.006(5)	0.698±0.006(6)
Dim. of	Hamming loss (↓)					
Yeast-v1	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.426±0.014(1)**	**0.426±0.012(1)**	0.427±0.012(3)	0.429±0.014(4)	0.436±0.015(5)	0.437±0.012(6)
Dim2	**0.417±0.007(1)**	0.424±0.010(3)	0.425±0.010(4)	0.423±0.008(2)	0.427±0.006(6)	0.426±0.010(5)
Dim3	**0.409±0.009(1)**	0.419±0.006(4)	0.419±0.006(4)	0.413±0.010(2)	0.413±0.009(2)	0.428±0.010(6)
Dim4	0.336±0.009(3)	0.336±0.008(3)	0.337±0.007(5)	0.335±0.007(2)	**0.334±0.010(1)**	0.343±0.009(6)
Dim5	0.433±0.020(2)	0.446±0.017(5)	0.440±0.015(4)	**0.432±0.016(1)**	0.436±0.018(3)	0.474±0.010(6)
Dim6	**0.412±0.015(1)**	0.414±0.016(2)	0.414±0.017(2)	0.416±0.022(4)	0.420±0.021(5)	0.424±0.016(6)
Avg.	**0.406±0.007(1)**	0.411±0.006(4)	0.410±0.006(3)	0.408±0.008(2)	0.411±0.005(4)	0.422±0.005(6)
Dim. of	Macro-F1 (↑)					
Yeast-v1	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.468±0.011(3)	0.375±0.030(4)	0.374±0.029(5)	0.488±0.016(2)	**0.496±0.012(1)**	0.368±0.010(6)
Dim2	0.472±0.018(3)	0.444±0.025(4)	0.414±0.037(5)	0.495±0.012(2)	**0.500±0.012(1)**	0.388±0.004(6)
Dim3	0.514±0.017(3)	0.453±0.024(4)	0.451±0.023(5)	0.521±0.014(2)	**0.522±0.013(1)**	0.424±0.020(6)
Dim4	0.433±0.008(6)	0.438±0.006(4)	0.435±0.005(5)	0.474±0.010(2)	**0.477±0.013(1)**	0.444±0.006(3)
Dim5	0.537±0.028(4)	**0.592±0.070(1)**	0.530±0.035(5)	0.540±0.016(3)	0.543±0.020(2)	0.302±0.027(6)
Dim6	0.382±0.022(3)	0.349±0.003(4)	0.328±0.018(5)	0.389±0.019(2)	**0.419±0.018(1)**	0.292±0.010(6)
Avg.	0.467±0.007(3)	0.442±0.015(4)	0.422±0.016(5)	0.485±0.008(2)	**0.493±0.007(1)**	0.370±0.006(6)

Table 6: Experimental results (mean±std. deviation) of CLIM and each compared approach over data set `Song-v2`. In addition, the performance ranks of all compared approaches are also shown in parentheses and the best performance is in bold. ↓ (↑) behind the name of each evaluation metric indicates that the smaller (larger) the value, the better the performance.

Dim. of Song-v2	Ranking loss (↓)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.183±0.022(1)**	0.213±0.013(3)	0.213±0.016(3)	0.422±0.091(5)	0.445±0.082(6)	0.201±0.017(2)
Dim2	0.064±0.012(2)	0.071±0.014(3)	0.071±0.014(3)	0.633±0.043(6)	0.628±0.045(5)	**0.061±0.015(1)**
Dim3	**0.103±0.016(1)**	0.112±0.018(2)	0.112±0.018(2)	0.446±0.095(6)	0.435±0.084(5)	0.115±0.021(4)
Avg.	**0.117±0.010(1)**	0.132±0.008(3)	0.132±0.009(3)	0.500±0.065(5)	0.502±0.060(6)	0.126±0.011(2)
Dim. of Song-v2	Coverage (↓)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.267±0.029(1)**	0.305±0.027(4)	0.303±0.025(3)	0.453±0.079(5)	0.471±0.067(6)	0.287±0.026(2)
Dim2	0.109±0.024(2)	0.116±0.024(3)	0.116±0.024(3)	0.591±0.040(6)	0.586±0.041(5)	**0.105±0.025(1)**
Dim3	**0.334±0.039(1)**	0.345±0.037(3)	0.345±0.037(3)	0.553±0.073(6)	0.544±0.064(5)	0.341±0.037(2)
Avg.	**0.237±0.019(1)**	0.255±0.015(4)	0.254±0.015(3)	0.532±0.054(5)	0.534±0.048(6)	0.244±0.017(2)
Dim. of Song-v2	One error (↓)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.396±0.036(2)	0.396±0.030(2)	**0.394±0.032(1)**	0.977±0.022(5)	0.977±0.017(5)	0.409±0.037(4)
Dim2	**0.161±0.049(1)**	**0.161±0.049(1)**	**0.161±0.049(1)**	0.995±0.007(5)	0.995±0.007(5)	**0.161±0.049(1)**
Dim3	**0.213±0.043(1)**	0.239±0.041(2)	0.245±0.041(3)	0.985±0.013(5)	0.985±0.013(5)	0.294±0.024(4)
Avg.	**0.256±0.015(1)**	0.265±0.022(2)	0.266±0.022(3)	0.986±0.008(5)	0.986±0.006(5)	0.288±0.020(4)
Dim. of Song-v2	Average precision (↑)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.683±0.018(1)**	0.658±0.012(4)	0.660±0.013(2)	0.283±0.050(5)	0.273±0.051(6)	0.659±0.016(3)
Dim2	0.867±0.030(2)	0.857±0.031(3)	0.857±0.031(3)	0.181±0.012(6)	0.183±0.012(5)	**0.868±0.032(1)**
Dim3	**0.783±0.020(1)**	0.762±0.027(2)	0.760±0.028(3)	0.288±0.033(6)	0.292±0.031(5)	0.727±0.028(4)
Avg.	**0.778±0.009(1)**	0.759±0.012(2)	0.759±0.013(2)	0.251±0.026(5)	0.249±0.026(6)	0.751±0.015(4)
Dim. of Song-v2	Hamming loss (↓)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.136±0.012(1)**	0.138±0.013(3)	0.137±0.012(2)	0.219±0.011(5)	0.219±0.011(5)	0.204±0.009(4)
Dim2	0.067±0.011(3)	0.067±0.011(3)	0.066±0.011(2)	0.163±0.012(5)	0.163±0.012(5)	**0.065±0.012(1)**
Dim3	**0.175±0.012(1)**	0.179±0.013(2)	0.180±0.014(3)	0.230±0.015(6)	0.229±0.015(5)	0.205±0.019(4)
Avg.	**0.126±0.006(1)**	0.128±0.007(2)	0.128±0.007(2)	0.204±0.007(5)	0.204±0.008(5)	0.158±0.008(4)
Dim. of Song-v2	Macro-F1 (↑)					
	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.119±0.007(4)	0.119±0.010(4)	0.122±0.012(3)	**0.197±0.035(1)**	**0.197±0.035(1)**	0.100±0.004(6)
Dim2	0.106±0.012(3)	0.099±0.010(5)	0.105±0.013(4)	0.137±0.022(2)	**0.139±0.024(1)**	0.091±0.003(6)
Dim3	0.226±0.013(4)	0.231±0.007(3)	0.207±0.014(5)	0.249±0.019(2)	**0.250±0.021(1)**	0.174±0.009(6)
Avg.	0.150±0.004(3)	0.150±0.006(3)	0.145±0.007(5)	0.194±0.011(2)	**0.196±0.013(1)**	0.122±0.004(6)

also shown that CLIM achieves inferior performance in terms of macro-F1 over all data sets. Possible reason is that both BR and CC directly learn a binary classifier for each label while CLIM divides label set into relevant part and irrelevant part according the threshold determined by maximizing the metric value of *hamming loss* over training set.

- Compared with WRAP, it is shown that the proposed CLIM approach achieves superior average performance in terms of each metric over all data sets except average precision over `Yeast-v2`. WRAP is one of the state-of-

Table 7: Experimental results (mean±std. deviation) of CLIM and each compared approach over data set `Yeast-v2`. In addition, the performance ranks of all compared approaches are also shown in parentheses and the best performance is in bold. ↓ (↑) behind the name of each evaluation metric indicates that the smaller (larger) the value, the better the performance.

Dim. of	Ranking loss (↓)					
Yeast-v2	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.349±0.014(1)**	0.366±0.015(4)	0.353±0.015(2)	0.614±0.021(5)	0.614±0.021(5)	0.356±0.015(3)
Dim2	**0.323±0.010(1)**	0.325±0.011(4)	**0.323±0.012(1)**	0.525±0.056(5)	0.525±0.056(5)	**0.323±0.011(1)**
Dim3	0.405±0.012(2)	0.415±0.011(4)	**0.397±0.011(1)**	0.566±0.015(5)	0.567±0.015(6)	0.405±0.011(2)
Dim4	**0.263±0.006(1)**	0.270±0.009(3)	0.265±0.008(2)	0.423±0.043(5)	0.429±0.035(6)	0.275±0.009(4)
Dim5	**0.396±0.019(1)**	0.406±0.022(3)	0.405±0.021(2)	0.580±0.029(5)	0.580±0.026(5)	0.408±0.021(4)
Dim6	**0.333±0.021(1)**	0.334±0.022(2)	0.513±0.017(4)	0.650±0.040(6)	0.643±0.036(5)	0.337±0.022(3)
Avg.	**0.345±0.006(1)**	0.353±0.006(3)	0.376±0.005(4)	0.560±0.009(5)	0.560±0.009(5)	0.351±0.006(2)
Dim. of	Coverage (↓)					
Yeast-v2	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.487±0.021(1)**	0.503±0.020(4)	0.489±0.020(2)	0.708±0.025(5)	0.708±0.025(5)	0.492±0.019(3)
Dim2	0.488±0.018(3)	0.490±0.018(4)	**0.484±0.018(1)**	0.642±0.055(5)	0.642±0.054(5)	0.485±0.018(2)
Dim3	0.592±0.010(3)	0.601±0.009(4)	**0.581±0.011(1)**	0.677±0.020(5)	0.677±0.019(5)	0.587±0.009(2)
Dim4	0.453±0.011(3)	0.454±0.012(4)	**0.451±0.012(1)**	0.519±0.037(5)	0.530±0.031(6)	**0.451±0.013(1)**
Dim5	**0.491±0.025(1)**	0.495±0.027(2)	0.495±0.027(2)	0.630±0.024(6)	0.628±0.024(5)	0.495±0.027(2)
Dim6	**0.427±0.017(1)**	0.430±0.019(2)	0.595±0.018(4)	0.686±0.035(6)	0.681±0.031(5)	0.432±0.019(3)
Avg.	**0.490±0.006(1)**	0.496±0.006(3)	0.516±0.007(4)	0.644±0.008(5)	0.644±0.007(5)	**0.490±0.006(1)**
Dim. of	One error (↓)					
Yeast-v2	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.725±0.031(3)	0.732±0.031(4)	**0.703±0.027(1)**	0.948±0.015(5)	0.948±0.015(5)	**0.703±0.027(1)**
Dim2	**0.627±0.029(1)**	**0.627±0.029(1)**	**0.627±0.029(1)**	0.898±0.013(5)	0.898±0.013(5)	**0.627±0.029(1)**
Dim3	0.656±0.025(2)	0.660±0.027(3)	**0.655±0.026(1)**	0.845±0.023(5)	0.846±0.022(6)	0.662±0.026(4)
Dim4	**0.392±0.038(1)**	0.396±0.041(3)	0.394±0.042(2)	0.780±0.030(6)	0.776±0.027(5)	0.398±0.040(4)
Dim5	**0.536±0.045(1)**	0.558±0.034(3)	0.557±0.035(2)	0.729±0.027(6)	0.727±0.026(5)	0.558±0.034(3)
Dim6	**0.438±0.027(1)**	**0.438±0.027(1)**	0.652±0.026(4)	0.755±0.024(6)	0.753±0.026(5)	**0.438±0.027(1)**
Avg.	**0.562±0.016(1)**	0.568±0.015(3)	0.598±0.017(4)	0.826±0.010(6)	0.825±0.011(5)	0.564±0.017(2)
Dim. of	Average precision (↑)					
Yeast-v2	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.404±0.018(3)	0.380±0.019(4)	**0.453±0.019(1)**	0.214±0.015(5)	0.214±0.015(5)	0.452±0.018(2)
Dim2	**0.491±0.018(1)**	0.490±0.019(3)	**0.491±0.019(1)**	0.286±0.012(5)	0.286±0.012(5)	0.490±0.019(3)
Dim3	0.440±0.014(2)	0.430±0.017(4)	**0.447±0.015(1)**	0.327±0.012(5)	0.327±0.012(5)	0.440±0.019(2)
Dim4	**0.696±0.018(1)**	0.693±0.020(3)	0.694±0.020(2)	0.547±0.036(5)	0.544±0.033(6)	0.688±0.019(4)
Dim5	**0.620±0.022(1)**	0.607±0.017(3)	0.608±0.017(2)	0.475±0.024(6)	0.477±0.022(5)	0.606±0.017(4)
Dim6	**0.714±0.011(1)**	0.713±0.012(2)	0.515±0.012(4)	0.449±0.020(6)	0.452±0.020(5)	0.711±0.012(3)
Avg.	0.561±0.010(2)	0.552±0.009(3)	0.535±0.009(4)	0.383±0.007(5)	0.383±0.007(5)	**0.565±0.010(1)**
Dim. of	Hamming loss (↓)					
Yeast-v2	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	**0.165±0.013(1)**	**0.165±0.013(1)**	**0.165±0.013(1)**	0.289±0.025(5)	0.289±0.025(5)	0.188±0.012(4)
Dim2	**0.214±0.011(1)**	**0.214±0.011(1)**	**0.214±0.011(1)**	0.406±0.052(5)	0.406±0.052(5)	0.231±0.010(4)
Dim3	**0.248±0.010(1)**	**0.248±0.010(1)**	**0.248±0.010(1)**	0.574±0.018(6)	0.571±0.013(5)	0.271±0.007(4)
Dim4	**0.322±0.009(1)**	0.327±0.012(2)	0.328±0.011(3)	0.425±0.022(5)	0.430±0.020(6)	0.339±0.010(4)
Dim5	**0.346±0.015(1)**	0.348±0.015(3)	**0.346±0.018(1)**	0.585±0.024(5)	0.593±0.025(6)	0.368±0.017(4)
Dim6	0.331±0.013(3)	**0.330±0.013(1)**	0.351±0.006(4)	0.610±0.041(5)	0.628±0.041(6)	**0.330±0.014(1)**
Avg.	**0.271±0.004(1)**	0.272±0.005(2)	0.275±0.005(3)	0.482±0.011(5)	0.486±0.010(6)	0.288±0.004(4)
Dim. of	Macro-F1 (↑)					
Yeast-v2	CLIM	DiDe	DiCo	BR	CC	WRAP
Dim1	0.000±0.001(4)	0.000±0.001(4)	0.000±0.000(4)	**0.033±0.011(1)**	**0.033±0.011(1)**	0.025±0.002(3)
Dim2	0.001±0.002(5)	0.001±0.002(5)	0.002±0.001(4)	0.076±0.031(2)	**0.078±0.030(1)**	0.036±0.002(3)
Dim3	0.015±0.003(4)	0.014±0.004(5)	0.012±0.005(6)	**0.203±0.014(1)**	0.202±0.012(2)	0.036±0.002(3)
Dim4	0.286±0.017(3)	0.245±0.010(4)	0.232±0.014(5)	0.368±0.010(2)	**0.381±0.011(1)**	0.114±0.008(6)
Dim5	0.077±0.039(4)	0.005±0.003(6)	0.060±0.010(5)	0.434±0.033(2)	**0.454±0.023(1)**	0.102±0.006(3)
Dim6	0.120±0.007(3)	0.120±0.008(3)	0.009±0.006(6)	0.452±0.050(2)	**0.477±0.054(1)**	0.120±0.004(3)
Avg.	0.083±0.008(3)	0.064±0.003(5)	0.053±0.004(6)	0.261±0.014(2)	**0.271±0.010(1)**	0.072±0.002(4)

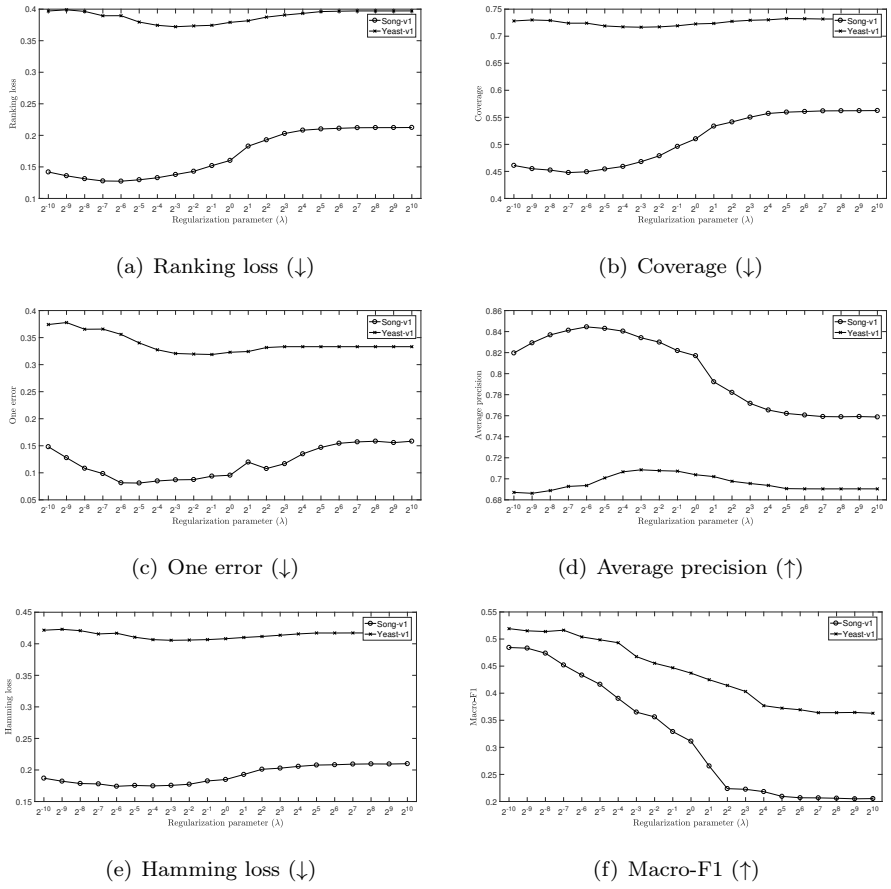
the-art multi-label classification baselines and achieves highly competitive performance in solving multi-label learning tasks [24]. These experimental results not only show the superiority of CLIM against WRAP, but also demonstrate that it is necessary to specifically design learning methods for MDML rather than solve the MDML problem by directly adopting some algorithms from related fields.

- As shown in Tables 2-3, the four benchmark data sets have diverse characteristics (e.g., the two Song data sets and the two Yeast data sets have three and six dimensions in output space, respectively), while as shown in Tables 4-7, there are not too many differences in their respective experimental results. These experimental results show that our CLIM approach can achieve stably superior performance over diverse MDML learning tasks which is a desirable property in real-world applications.

4.3. Further Analysis

4.3.1. Parameter Sensitivity Analysis

As shown in Section 3.3, the proposed CLIM approach only includes one parameter λ to be set, which is used to trade-off the empirical risk and structural risk (i.e., the regularization term). In this section, we investigate how the performance of CLIM fluctuates when the value of λ changes. Figure 3 illustrates the performance curves when λ increases from 2^{-10} to 2^{10} over data sets `Song-v1` and `Yeast-v1` in terms of each metric. Generally speaking, it is shown that the performance of CLIM is sensitive to the value of λ where either small or large λ will lead to performance degeneration. As stated in Section 3.3, the convergence condition in CLIM corresponds to that the *ranking loss* over *training set* cannot be further improved, which might be the cause of this phenomenon. In previous sections, we fix λ as the moderate value 2^{-3} which is usually a better choice over all data sets in terms of all metrics except for macro-F1. The exception might be due to that the dividing threshold is determined by maximizing the metric value of *hamming loss* over training set. Similar results also exist in the comparison of CLIM against BR&CP.



(a) Ranking loss (↓)

(b) Coverage (↓)

(c) One error (↓)

(d) Average precision (↑)

(e) Hamming loss (↓)

(f) Macro-F1 (↑)

Figure 3: Performance of CLIM changes as λ increase from 2^{-10} to 2^{10} in terms of each evaluation metric over data sets `Song-v1` and `Yeast-v1`.

4.3.2. Ablation Study

As we have stated before, DiDe and DiCo act as two degenerated versions of CLIM. Then the comparison between CLIM and these two benchmark approaches act as ablation studies to investigate the effectiveness of CLIM's design. The superiority of CLIM against DiDe and DiCo clearly show that either ignoring label correlations among multiple dimensions or aligning labels from heterogeneous label spaces into a homogeneous one will lead to performance degeneration.

In this subsection, we further investigate the effectiveness of the MAP-based threshold determination strategy. Specifically, CLIM searches in the candidate

Table 8: Experimental results (mean±std. deviation) of CLIM and its two degenerated versions in terms of *average hamming loss* and *average macro*-F1. In addition, the performance ranks of all compared approaches are also shown in parentheses and the best performance is in bold. ↓ (↑) behind the name of each evaluation metric indicates that the smaller (larger) the value, the better the performance.

Data Set	Hamming loss (↓)		
	CLIM	CLIM-THv1	CLIM-THv2
Song-v1	**0.176±0.007(1)**	0.219±0.008(3)	0.187±0.008(2)
Yeast-v1	**0.406±0.007(1)**	**0.406±0.008(1)**	0.436±0.007(3)
Song-v2	**0.126±0.006(1)**	0.180±0.005(3)	0.129±0.008(2)
Yeast-v2	**0.271±0.004(1)**	0.406±0.005(3)	0.303±0.006(2)
Data Set	Macro-F1 (↑)		
	CLIM	CLIM-THv1	CLIM-THv2
Song-v1	0.365±0.013(2)	**0.439±0.006(1)**	0.302±0.010(3)
Yeast-v1	**0.467±0.007(1)**	0.446±0.005(2)	0.254±0.016(3)
Song-v2	0.150±0.004(2)	**0.234±0.016(1)**	0.143±0.004(3)
Yeast-v2	0.083±0.008(3)	**0.298±0.008(1)**	0.166±0.014(2)

threshold set and selects the value which can minimize the metric value of *hamming loss* over training set as the final threshold. Based on the obtained threshold, the label set is divided into relevant part and irrelevant part according to the real-valued modeling outputs of one instance (c.f. Eq.(6)). We compare this strategy with another two general strategies, which are respectively named as CLIM-THv1 and CLIM-THv2.

For CLIM-THv1, the threshold is simply fixed as the average value of modeling outputs, which is similar to the relevant labels determination strategy of `Song-v1` and `Yeast-v1`. For CLIM-THv2, assume that labels in one dimension are sorted in descending order according to their modeling outputs, among the labels whose modeling outputs are larger than the average value, we compute all differences of modeling outputs between one label and its adjacent one with smaller modeling output, the label will be predicted as a relevant one if it is in the front of the position of the largest difference. In other words, this strategy is similar to the relevant labels determination strategy of `Song-v2` and `Yeast-v2`.

Table 8 shows the detailed experimental results in terms of the two classification-

based metrics *hamming loss* and *macro*-F1 which are directly related to threshold determination. It is shown that CLIM achieves superior performance against CLIM-THv1 and CLIM-THv2 in terms of *hamming loss*, but usually achieves inferior performance against CLIM-THv1 in terms of *macro*-F1. These results demonstrate that minimizing empirical *hamming loss* does improve the generalization performance for this metric, but not necessarily for other metrics. This means that we should design specific threshold determination strategy for our aiming classification-based metric. Besides, it is shown that CLIM-THv1 (or CLIM-THv2) does not necessarily achieve better performance over `Song-v1` and `Yeast-v1` (or `Song-v2` and `Yeast-v2`), which means that label generation strategy does not have necessary relationship with threshold determination strategy. Overall, CLIM achieves moderate performance against the two compared strategies and other subtler designs can be explored in future.

4.3.3. Complexity Analysis

The main computational complexity of CLIM is to optimize the problem (4), which is solved via gradient descent. For each iteration in gradient descent, the main complexity is to compute the objective function of problem (4) and its gradient in Eq.(5), where the complexity corresponds to $\mathcal{O}(m \cdot d \cdot K_{\max})$. Here, m, d, $K_{\max} = \max_{1 \le j \le q} K_j$ denotes the number of examples, the number of features and the largest number of labels per label space. For the threshold determination procedure in Algorithm 1, the main complexity corresponds to the ascending sort process (Step 2) which can be done with $\mathcal{O}(m \cdot K_{\max} \cdot \log(m \cdot K_{\max}))$. Therefore, the total computational complexity of CLIM corresponds to $\mathcal{O}(q \cdot T_{\max} \cdot m \cdot d \cdot K_{\max} + q \cdot (m \cdot K_{\max}) \cdot \log(m \cdot K_{\max}))$, where q denotes the number of dimensions and $T_{\max}$ denotes the number of largest iteration rounds of gradient descent. However, it is hard to theoretically analyze the value of $T_{\max}$ which depends on specific objective function. Table 9 shows the time costs (unit: s) of CLIM and each compared approach over each data set. It is shown that CLIM takes comparable time with WRAP and longer time than the rest of compared approaches.

Table 9: The time costs (unit: s) of CLIM and each compared approach over each data set.

Data Set	CLIM	DiDe	DiCo	BR	CC	WRAP
Song-v1	62	11	12	4	9	96
Yeast-v1	1030	18	46	7	29	418
Song-v2	41	8	10	4	8	145
Yeast-v2	232	13	41	6	20	672

5. Conclusion

The main contributions of this paper are three-fold: (1) We formalize a new learning framework named multi-dimensional multi-label classification to learn from objects whose semantics need to be characterized with multiple heterogeneous label spaces as well as multi-label annotations in each label space. (2) We specifically design an approach named CLIM to solve the MDML problem which can consider label correlations within individual dimension and across multiple dimensions. (3) We evaluate the performance of CLIM over four real-world MDML data sets and the experimental results clearly validate the effectiveness of the proposed approach.

This paper only makes the first attempt towards the new MDML classification framework. As the key issue for MDML model induction, it is desirable to further explore how to consider the two kinds of label correlations within individual dimension and across multiple dimensions more effectively. Besides, it is also desirable to collect more benchmark data sets from real-world MDML tasks of diverse domains and design some specific evaluation metrics to evaluate the performance of MDML classifiers more comprehensively in the future.

References

[1] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, John Wiley & Sons, New York City, NY, USA, 2001.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, F. Li, ImageNet large

scale visual recognition challenge, International Journal of Computer Vision 115 (3) (2015) 211–252. doi:10.1007/s11263-015-0816-y.

[3] X. Shu, J. Tang, G. Qi, Z. Li, Y. Jiang, S. Yan, Image classification with tailored fine-grained dictionaries, IEEE Transactions on Circuits and Systems for Video Technology 28 (2) (2018) 454–467. doi:10.1109/TCSVT.2016.2607345.

[4] J. Read, C. Bielza, P. Larrañaga, Multi-dimensional classification with super-classes, IEEE Transactions on Knowledge and Data Engineering 26 (7) (2014) 1720–1733. doi:10.1109/TKDE.2013.167.

[5] Z. Ma, S. Chen, Multi-dimensional classification via a metric approach, Neurocomputing 275 (2018) 1121–1131. doi:10.1016/j.neucom.2017.09.057.

[6] C. Liu, P. Zhao, S.-J. Huang, Y. Jiang, Z.-H. Zhou, Dual set multi-label learning, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI Press, New Orleans, LA, USA, 2018, pp. 3635–3642.

[7] B.-B. Jia, M.-L. Zhang, Multi-dimensional classification via selective feature augmentation, Machine Intelligence Research 19 (1) (2022) 38–51. doi:10.1007/s11633-022-1316-5.

[8] E. Gibaja, S. Ventura, A tutorial on multilabel learning, ACM Computing Surveys 47 (3) (2015) Article 52. doi:10.1145/2716262.

[9] Z.-H. Zhou, M.-L. Zhang, Multi-label learning, in: C. Sammut, G. I. Webb (Eds.), Encyclopedia of Machine Learning and Data Mining, Springer, 2017, pp. 875–881. doi:10.1007/978-1-4899-7687-1_910.

[10] M. R. Boutell, J. Luo, X. Shen, C. M. Brown, Learning multi-label scene classification, Pattern Recognition 37 (9) (2004) 1757–1771. doi:10.1016/j.patcog.2004.03.009.

[11] M.-L. Zhang, Z.-H. Zhou, ML-KNN: A lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038–2048. doi:10.1016/j.patcog.2006.12.019.

[12] X. Shu, G. Qi, J. Tang, J. Wang, Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation, in: Proceedings of the 23rd ACM International Conference on Multimedia, ACM, Brisbane, Australia, 2015, pp. 35–44. doi:10.1145/2733373.2806216.

[13] J. Tang, X. Shu, Z. Li, G. Qi, J. Wang, Generalized deep transfer networks for knowledge propagation in heterogeneous domains, ACM Transactions on Multimedia Computing, Communications, and Applications 12 (4s) (2016) Article 68. doi:10.1145/2998574.

[14] B.-B. Jia, M.-L. Zhang, Decomposition-based classifier chains for multi-dimensional classification, IEEE Transactions on Artificial Intelligence 3 (2) (2022) 176–191. doi:10.1109/TAI.2021.3110935.

[15] B.-B. Jia, M.-L. Zhang, Maximum margin multi-dimensional classification, IEEE Transactions on Neural Networks and Learning Systems 33 (12) (2022) 7185–7198. doi:10.1109/TNNLS.2021.3084373.

[16] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, X. Geng, Binary relevance for multi-label learning: An overview, Frontiers of Computer Science 12 (2) (2018) 191–202. doi:10.1007/s11704-017-7031-7.

[17] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Machine Learning 85 (3) (2011) 333–359. doi:10.1007/s10994-011-5256-5.

[18] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains: A review and perspectives, Journal of Artificial Intelligence Research 70 (2021) 683–718. doi:10.1613/jair.1.12376.

[19] M.-L. Zhang, Z.-H. Zhou, Multi-label neural networks with applications to functional genomics and text categorization, IEEE Transactions on

Knowledge and Data Engineering 18 (10) (2006) 1338–1351. doi:10.1109/TKDE.2006.162.

[20] M.-L. Zhang, Q.-W. Zhang, J.-P. Fang, Y.-K. Li, X. Geng, Leveraging implicit relative labeling-importance information for effective multi-label learning, IEEE Transactions on Knowledge and Data Engineering 33 (5) (2021) 2057–2070. doi:10.1109/TKDE.2019.2951561.

[21] M.-L. Zhang, L. Wu, LIFT: Multi-label learning with label-specific features, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (1) (2015) 107–120. doi:10.1109/TPAMI.2014.2339815.

[22] J. Huang, G. Li, Q. Huang, X. Wu, Learning label-specific features and class-dependent labels for multi-label classification, IEEE Transactions on Knowledge and Data Engineering 28 (12) (2016) 3309–3323. doi:10.1109/TKDE.2016.2608339.

[23] M.-L. Zhang, J.-P. Fang, Y.-B. Wang, BiLabel-specific features for multi-label classification, ACM Transactions on Knowledge Discovery from Data 16 (1) (2022) Article 18. doi:10.1145/3458283.

[24] Z.-B. Yu, M.-L. Zhang, Multi-label classification with label-specific feature generation: A wrapped approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (9) (2022) 5199–5210. doi:10.1109/TPAMI.2021.3070215.

[25] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE Transactions on Knowledge and Data Engineering 26 (8) (2014) 1819–1837. doi:10.1109/TKDE.2013.39.

[26] W. Liu, X. Shen, H. Wang, I. W. Tsang, The emerging trends of multi-label learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (11) (2022) 7955–7974. doi:10.1109/TPAMI.2021.3119334.

[27] C. Bielza, G. Li, P. Larrañaga, Multi-dimensional classification with Bayesian networks, International Journal of Approximate Reasoning 52 (6) (2011) 705–727. doi:10.1016/j.ijar.2011.01.007.

[28] S. Gil-Begue, C. Bielza, P. Larrañaga, Multi-dimensional Bayesian network classifiers: A survey, Artificial Intelligence Review 54 (1) (2021) 519–559. doi:10.1007/s10462-020-09858-x.

[29] J. H. Zaragoza, L. E. Sucar, E. F. Morales, C. Bielza, P. Larrañaga, Bayesian chain classifiers for multidimensional classification, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI/AAAI, Barcelona, Catalonia, Spain, 2011, pp. 2192–2197. doi:10.5591/978-1-57735-516-8/IJCAI11-365.

[30] J. Read, L. Martino, D. Luengo, Efficient monte carlo methods for multi-dimensional learning with classifier chains, Pattern Recognition 47 (3) (2014) 1535–1546. doi:10.1016/j.patcog.2013.10.006.

[31] B.-B. Jia, M.-L. Zhang, Multi-dimensional classification via stacked dependency exploitation, Science China Information Sciences 63 (12) (2020) Article 222102. doi:10.1007/s11432-019-2905-3.

[32] B.-B. Jia, M.-L. Zhang, MD-KNN: An instance-based approach for multi-dimensional classification, in: Proceedings of the 25th International Conference on Pattern Recognition, IEEE, Virtual Event / Milan, Italy, 2020, pp. 126–133. doi:10.1109/ICPR48806.2021.9412974.

[33] B.-B. Jia, M.-L. Zhang, Multi-dimensional classification via sparse label encoding, in: Proceedings of the 38th International Conference on Machine Learning, PMLR, Virtual Event, 2021, pp. 4917–4926.

[34] B.-B. Jia, M.-L. Zhang, Multi-dimensional classification via decomposed label encoding, IEEE Transactions on Knowledge and Data Engineering 35 (2) (2023) 1844–1856. doi:10.1109/TKDE.2021.3100436.

[35] H. Wang, C. Chen, W. Liu, K. Chen, T. Hu, G. Chen, Incorporating label embedding and feature augmentation for multi-dimensional classification, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI Press, New York City, NY, USA, 2020, pp. 6178–6185.

[36] B.-B. Jia, M.-L. Zhang, Multi-dimensional classification via kNN feature augmentation, Pattern Recognition 106 (2020) Article 107423. doi:10.1016/j.patcog.2020.107423.

[37] M. B. Eisen, P. T. Spellman, P. O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proceedings of the National Academy of Sciences 95 (25) (1998) 14863–14868.

[38] X.-Z. Wu, Z.-H. Zhou, A unified view of multi-label performance measures, in: Proceedings of the 34th International Conference on Machine Learning, PMLR, Sydney, Australia, 2017, pp. 3780–3788.

[39] J.-Y. Hang, M.-L. Zhang, Collaborative learning of label semantics and deep label-specific features for multi-label classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (12) (2022) 9860–9871. doi:10.1109/TPAMI.2021.3136592.

[40] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, LIBLINEAR: A library for large linear classification, Journal of Machine Learning Research 9 (2008) 1871–1874. doi:10.5555/1390681.1442794.