https://doi.org/10.1007/s11704-025-41432-y

RESEARCH ARTICLE

Similarity-based multi-dimensional multi-label classification

Zi-Zhan GU[1,3], Bin-Bin JIA[2], Min-Ling ZHANG[1,3]✉

1. School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
2. College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China
3. Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China

Received December 29, 2024; accepted March 25, 2025

E-mail: zhangml@seu.edu.cn

© The Author(s) 2025. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract

In multi-dimensional multi-label classification (MDML), a number of heterogeneous label spaces are assumed to characterize the rich semantics of one object from different dimensions and a set of proper labels can be assigned to the object from each heterogeneous label space. In recent years, similarity-based framework has achieved a promising performance in classification tasks (e.g., multi-class/multi-label classification), while its effectiveness has not been investigated in solving the MDML problems. Moreover, existing similarity-based approaches only utilize either instance-based or label-based information which limits their generalization ability. In this paper, we propose a novel similarity-based MDML approach, naming SIDLE which attempts to utilize both instance-based and label-based information. To extract similarity information, SIDLE first identifies k nearest neighbors in instance space and enhanced label space, respectively. Then, with these identified samples, SIDLE calculates the simple counting statistics based on their labels as well as a bias based on distance between the sample and these identified samples. Finally, the instance space is enriched with extracted similarity information to update instance space and enhanced label space. These three steps are iteratively conducted until convergence. Experiments validate the effectiveness of the proposed SIDLE approach.

Keywords

machine learning; multi-dimensional multi-label classification; similarity-based learning; k nearest neighbor; feature augmentation

1 Introduction

In supervised learning, the most commonly-used assumption is that each object is annotated with a unique class label from a single label space, e.g., multi-class classification. However, the rich semantics of objects in some applications need to be characterized by encompassing heterogeneous label spaces and multi-label annotations. For example, songs can be categorized based on the emotions they contain, the genres they belong to, and the scenarios in which they are suitable to play. Moreover, each song might contain different kinds of emotions (e.g., happy, sad, relax), belong to different kinds of genres (e.g., classical, rock, popular), and be suitable to play in different kinds of scenarios (e.g., bedtime, wedding, walking). These problems can be naturally formalized under the multi-dimensional multi-label classification (MDML) situation [1]. Here, the multi-dimensional semantics of one MDML object is characterized with a number of heterogeneous label spaces and multiple proper labels from each heterogeneous label space can be assigned to the object to characterize its ambiguous semantics along this dimension. Specifically, MDML problems can be found in different kinds of applications [2–5].

To learn from MDML samples, the most intuitive way is to decompose the original MDML problem into q multi-label classification problems, one per label space, or concatenate q label spaces to form the single multi-label classification problem. However, the former completely ignores potential label correlations across different label spaces, and the latter ignores the nature of multi-dimensional semantics for MDML objects. Then the pioneering work CLIM aims to consider label correlations within individual dimension and across multiple dimensions in different ways [1]. To consider label correlations within individual dimension, CLIM chooses to maximize the likelihood of relevant labels by regarding the fact that predictive confidences of labels from the same label space are comparable. To consider label correlations across multiple dimensions, CLIM chooses to manipulate the feature space via augmenting the feature space with predictions for each label space.

Similarity-based learning technique has shown its promising performance in different kinds of classification tasks [6–13], while its effectiveness has not been investigated in solving MDML problems. Moreover, existing similarity-based approaches only utilize either

instance-based or label-based information which limits their generalization ability. In this paper, we make a first attempt to adapt similarity-based learning technique for solving MDML problem, and propose a novel approach named SImilarity-based multi-Dimensional multi-LabEl classification (SIDLE). SIDLE proposes to extract both instance-based and label-based similarity information by identifying k nearest neighbors. For instance-based information, the identification is initiated in the original instance space, while for label-based information, the identification is conducted in an enhanced label space, which is initiated via optimizing cross entropy-like loss. With these identified samples, similarity information is represented by the simple counting statistics based on their labels as well as a bias based on distance between the sample and these identified samples. SIDLE combines instance-based and label-based similarity information with an element-wise addition and enriches the instance space with extracted similarity information. As the instance space is enriched, the enhanced label space will also be updated. These steps will be iteratively conducted until convergence. Comparative studies clearly demonstrate the superiority of the proposed SIDLE approach against state-of-the-art baselines.

The rest of this paper is organized as follows. In Section 2, related works on MDML are briefly discussed. In Section 3, technical details of the proposed SIDLE approach are introduced. In Section 4, experimental results of comparative studies are reported. Finally, we conclude this paper in Section 5.

2 Related work

The most related learning frameworks to MDML include multi-class classification (MCC), multi-label classification (MLC) [14–18], and multi-dimensional classification (MDC) [19–22]. As shown in Table 1, MDML generalizes multi-label classification from a single homogeneous label space to multiple heterogeneous label spaces and generalizes multi-dimensional classification from unique relevant label assumption in each label space to multiple relevant labels.

The study of MDML is started by [1], where the MDML problem is formalized and a specially designed MDML approach named CLIM is proposed. On one hand, based on the assumption that predictive confidences of labels from the same label space are comparable, CLIM considers the label correlations within each label space by maximizing the likelihood of relevant labels. On the other hand, motivated by [8,23], CLIM augments the instance space with binary predictions of all label spaces to update the predictive model.

Similarity-based classification refers to a category of classifiers that determine their judgment based on the computed similarity (e.g., Euclidean distance) between the target instance and the set of training instances [10]. Existing works can be roughly categorized into instance-based and label-based approach based on the type of used similarity information, whose representative algorithms are k nearest neighbors (kNN) classifier [24] and nearest class mean (NCM) classifier [25] respectively. kNN determines the prediction via majority voting according to the labels of k nearest neighbors identified in instance space. NCM calculates the mean instance vector for each label and assigns unseen instance to the class label with the closest mean. Similarity-based learning techniques have been utilized to solve many learning problems and applied in real-world applications, including MCC [10,25,26], MLC [7,27], MDC [8,9,28], multiple-instance classification [6], drug-target interactions prediction [29], etc.

Table 1 Relationships among multi-class classification (MCC), multi-label classification (MLC), multi-dimensional classification (MDC), and MDML

Paradigms	#Label spaces	#Relevant label(s)
MCC	One	One
MLC	One	More than one
MDC	More than one	One
MDML	More than one	More than one

However, similarity-based techniques have not been utilized to deal with MDML problem. Moreover, existing works utilize either instance-based or label-based similarity information, which limits their generalization ability. In the next section, we will present the SIDLE approach which solves the MDML problem with both instance-based and label-based similarity information.

3 The SIDLE approach

3.1 Problem formulation

Formally, let $\mathcal{X}=\mathbb{R}^d$ denote the d-dimensional input space and $\mathcal{Y}=\bigcup_{j=1}^{q}\mathcal{Y}^j$ be the output space which is the union of q heterogeneous label spaces. Here, each label space $\mathcal{Y}^j=\{y_1^j,y_2^j,\ldots,y_{K_j}^j\}$ $(1\leqslant j\leqslant q)$ comprises K_j labels and then the total number of labels across all q label spaces is given by $K=\sum_{j=1}^{q}K_j$. Let $\mathcal{D}=\{(\boldsymbol{x}_i,\boldsymbol{l}_i)\mid 1\leqslant i\leqslant m\}$ represent the MDML training set consisting of m training samples, the task of MDML is to learn a mapping function $f:\mathcal{X}\mapsto\mathcal{Y}$ from $\mathcal{D}$, enabling it to assign an appropriate set of labels to any unseen instance $\boldsymbol{x}_*$, denoted as $\hat{\boldsymbol{l}}_*=f(\boldsymbol{x}_*)$. For the ith sample $(\boldsymbol{x}_i,\boldsymbol{l}_i)\in\mathcal{D}$, where $\boldsymbol{x}_i=[x_{i1},x_{i2},\ldots,x_{id}]^\top\in\mathcal{X}$ is the length-d feature vector, its associated label vector is usually represented as a length-K binary label vector $\boldsymbol{l}_i=[\boldsymbol{l}_i^1;\boldsymbol{l}_i^2;\ldots;\boldsymbol{l}_i^q]\in\{0,1\}^K$ for notation convenience. Here, $\boldsymbol{l}_i^j=[l_{i1}^j,l_{i2}^j,\ldots,l_{iK_j}^j]^\top\in\{0,1\}^{K_j}$ signifies the relevance of labels within the jth label space. Specifically, the ath element $l_{ia}^j=1$ indicates that the ath label (i.e., y_a^j) in the jth label space (i.e., $\mathcal{Y}^j$) is relevant to $\boldsymbol{x}_i$, and $l_{ia}^j=0$ suggests otherwise. To facilitate the understanding, we summarize these notations in Table 2.

The technical details of SIDLE can be divided into three parts, including k nearest neighbors identification, similarity information extraction, and predictive model updating. Following the notations in Table 2, we will present these technical details in the rest of this section.

3.2 K Nearest neighbors identification

Given one instance $\boldsymbol{x}\in\mathbb{R}^d$, to identify its k nearest neighbors in the MDML training set $\mathcal{D}=\{(\boldsymbol{x}_i,\boldsymbol{l}_i)\mid 1\leqslant i\leqslant m\}$, in this paper, we

Table 2 Notations used in the MDML formulation

Notation	Description
d	The number of features in input space
q	The number of label spaces in output space
K_j	The number of labels in the jth label space
K	The total number of labels $K = \sum_{j=1}^{q} K_j$
m	The number of training examples
$\mathcal{X}$	The d dimensional input (feature) space
$\mathcal{Y}^j$	The jth label space $\mathcal{Y}^j = \{y_1^j, y_2^j, \ldots, y_{K_j}^j\}$
y_a^j	The ath label in $\mathcal{Y}^j$ $(1 \leqslant a \leqslant K_j)$
$\mathcal{Y}$	The output space $\mathcal{Y} = \bigcup_{j=1}^{q} \mathcal{Y}^j$
$\mathcal{D}$	The MDML training set
$\boldsymbol{x}_i$	The ith feature vector
$\boldsymbol{l}_i$	The label vector associated with $\boldsymbol{x}_i$
$\boldsymbol{l}_i^j$	The part of label vector in $\boldsymbol{l}_i$ w.r.t. $\mathcal{Y}^j$
l_{ia}^j	The ath element in $\boldsymbol{l}_i^j$
$\boldsymbol{x}_*$	The unseen instance
$\hat{\boldsymbol{l}}_*$	The predicted label vector for $\boldsymbol{x}_*$

simply calculate the Euclidean distance between $\boldsymbol{x}$ and each training sample $\boldsymbol{x}_i$ in $\mathcal{D}$:

$$d(\boldsymbol{x}, \boldsymbol{x}_i) = \|\boldsymbol{x} - \boldsymbol{x}_i\|_2, \ (1 \leqslant i \leqslant m). \tag{1}$$

Then we sort the obtained m distances and identify the k smallest ones among them. For the identified k nearest training samples of $\boldsymbol{x}$, their indices are stored as follows:

$$\mathcal{N}_k^I(\boldsymbol{x}) = \{i_r \mid 1 \leqslant r \leqslant k\}. \tag{2}$$

For convenience, the distance between $\boldsymbol{x}$ and $\boldsymbol{x}_{i_r}$ is arranged in ascending order. In other words, for the identified k nearest training samples, $\boldsymbol{x}_{i_1}$ is the closest one to $\boldsymbol{x}$ and $\boldsymbol{x}_{i_k}$ is the farthest one to $\boldsymbol{x}$. Note that Eq. (1) regards the similarity among samples in their instance space, thus we call the identified samples as instance-based k nearest neighbors. In Eq. (2), we use the superscript ^{I} (i.e., the first letter of *Instance*) to indicate this meaning.

However, for an unseen instance, as its labeling information is unknown, we cannot directly identify its k nearest neighbors in label space. Motivated by the idea of label enhancement [30], we choose to learn an enhanced label space via inducing a predictive model. Without loss of generality, let $\boldsymbol{\Theta}^j = [\boldsymbol{\theta}_1^j, \ldots, \boldsymbol{\theta}_{K_j}^j] \in \mathbb{R}^{d \times K_j}$ denote the model parameters for the jth label space $(1 \leqslant j \leqslant q)$, we determine these parameters via optimizing the following cross-entropy-like loss on the MDML training set $\mathcal{D}$:

$$\min_{\boldsymbol{\Theta}^j} -\sum_{i=1}^{m}\sum_{a=1}^{K_j} l_{ia}^j \cdot \ln \frac{e^{\langle \boldsymbol{\theta}_a^j, \boldsymbol{x}_i \rangle}}{\sum_{s=1}^{K_j} e^{\langle \boldsymbol{\theta}_s^j, \boldsymbol{x}_i \rangle}} + \frac{\lambda}{2}\left\|\boldsymbol{\Theta}^j\right\|_F^2, \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors, λ is the trade-off parameter to balance the importance of two terms. It is easy to see that we cannot derive a closed-form solution to Eq. (3), then we can optimize it via gradient descent.

With the determined model parameters $\boldsymbol{\Theta}^j$ $(1 \leqslant j \leqslant q)$, for each training sample $(\boldsymbol{x}_i, \boldsymbol{l}_i) \in \mathcal{D}$, we can transform its length-K binary label vector $\boldsymbol{l}_i = [\boldsymbol{l}_i^1; \ldots; \boldsymbol{l}_i^j; \ldots; \boldsymbol{l}_i^q] \in \{0,1\}^K$ to a length-K real-valued label vector $\boldsymbol{p}_i = [\boldsymbol{p}_i^1; \ldots; \boldsymbol{p}_i^j; \ldots; \boldsymbol{p}_i^q] \in \mathbb{R}^K$, where $\boldsymbol{p}_i^j$ is defined as follows:

$$\boldsymbol{p}_i^j = (\boldsymbol{\Theta}^j)^\top \boldsymbol{x}_i, \ (1 \leqslant j \leqslant q). \tag{4}$$

Given one instance $\boldsymbol{x}$ without label, we can also determine its length-K real-valued label vector $\boldsymbol{p}$ in a similar way. Then we can calculate the Euclidean distance between $\boldsymbol{p}$ and each transformed label vector $\boldsymbol{p}_i$:

$$d(\boldsymbol{p}, \boldsymbol{p}_i) = \|\boldsymbol{p} - \boldsymbol{p}_i\|_2, \ (1 \leqslant i \leqslant m). \tag{5}$$

By identifying the k smallest distances, we can identify the label-based k nearest training samples of $\boldsymbol{x}$ and store their indices as follows:

$$\mathcal{N}_k^L(\boldsymbol{x}) = \{i_r \mid 1 \leqslant r \leqslant k\}. \tag{6}$$

Here, we use the superscript ^{L} (i.e., the first letter of *Label*) to indicate that $\mathcal{N}_k^L(\boldsymbol{x})$ stores the set of indices for $\boldsymbol{x}$'s k nearest neighbors identified in label space.

3.3 Similarity information extraction

Motivated by the rationale of KNN classifier, we extract similarity information for one instance with the simple counting statistics based on the labels of its k nearest neighbors. For convenience, we temporarily use $\mathcal{N}_k(\boldsymbol{x}) = \{i_r \mid 1 \leqslant r \leqslant k\}$ to denote the set of indices for $\boldsymbol{x}$'s k nearest neighbors and no longer distinguish whether it is instance-based or label-based k nearest neighbors in this subsection.

According to the notations used in previous sections, for the rth nearest neighbor of $\boldsymbol{x}$, $l_{i_r a}^j = 1$ indicates that the ath label (i.e., y_a^j) in the jth label space (i.e., $\mathcal{Y}^j$) is relevant to $\boldsymbol{x}_{i_r}$, and $l_{i_r a}^j = 0$ suggests otherwise. Here, $i_r \in \mathcal{N}_k(\boldsymbol{x})$ and $1 \leqslant r \leqslant k$. Then we can define the following length-k indicating vector w.r.t. y_a^j in $\boldsymbol{x}$'s k nearest neighbors:

$$\boldsymbol{l}_{ja}^{\boldsymbol{x}} = [l_{i_1 a}^j, l_{i_2 a}^j, \ldots, l_{i_k a}^j]^\top \in \{0,1\}^k. \tag{7}$$

Let $\mathbf{1}_k$ be a length-k column vector of all ones, we can calculate the following statistics:

$$\delta_{ja}^{\boldsymbol{x}} = \langle \mathbf{1}_k, \boldsymbol{l}_{ja}^{\boldsymbol{x}} \rangle, \ (1 \leqslant j \leqslant q, 1 \leqslant a \leqslant K_j). \tag{8}$$

It is easy to know that $\delta_{ja}^{\boldsymbol{x}}$ records the number of samples which are associated with y_a^j in $\boldsymbol{x}$'s k nearest neighbors and $0 \leqslant \delta_{ja}^{\boldsymbol{x}} \leqslant k$ always holds.

Note that in the process of calculating the above $\delta_{ja}^{\boldsymbol{x}}$, the k nearest neighbors contribute equally. Generally, closer neighbors are of greater importance than farther neighbors in the determination of the final judgment. Thus, it might be better to assign a nonuniform weight to k nearest neighbors based on their distance from the target

sample. In this paper, we simply set $\boldsymbol{w} = [1, 1/\sqrt{2}, \ldots, 1/\sqrt{k}]^\top$ as the weight vector. Our motivation is that the weight for the closest neighbor is set to 1 and the weights will gradually decrease. Then we define a corresponding bias $\epsilon_{ja}^{\boldsymbol{x}}$ for $\delta_{ja}^{\boldsymbol{x}}$ as follows:

$$\epsilon_{ja}^{\boldsymbol{x}} = \frac{\langle \boldsymbol{w}, \boldsymbol{l}_{ja}^{\boldsymbol{x}} \rangle - min(\boldsymbol{l}_{ja}^{\boldsymbol{x}})}{max(\boldsymbol{l}_{ja}^{\boldsymbol{x}}) - min(\boldsymbol{l}_{ja}^{\boldsymbol{x}})}(\epsilon_{max} - \epsilon_{min}) + \epsilon_{min}. \tag{9}$$

Here, $max(\boldsymbol{l}_{ja}^{\boldsymbol{x}})$ and $min(\boldsymbol{l}_{ja}^{\boldsymbol{x}})$ represent the possible maximum and minimum of $\langle \boldsymbol{w}, \boldsymbol{l}_{ja}^{\boldsymbol{x}} \rangle$ respectively. It is not difficult to know that $max(\boldsymbol{l}_{ja}^{\boldsymbol{x}})$ corresponds to the sum of the first $\delta_{ja}^{\boldsymbol{x}}$ elements in $\boldsymbol{w}$ and $min(\boldsymbol{l}_{ja}^{\boldsymbol{x}})$ corresponds to the sum of the last $\delta_{ja}^{\boldsymbol{x}}$ elements in $\boldsymbol{w}$. In other words, their definitions can be respectively given as follows:

$$max(\boldsymbol{l}_{ja}^{\boldsymbol{x}}) = \sum\nolimits_{r=1}^{\delta_{ja}^{\boldsymbol{x}}} w(r),$$
$$min(\boldsymbol{l}_{ja}^{\boldsymbol{x}}) = \sum\nolimits_{r=k-\delta_{ja}^{\boldsymbol{x}}+1}^{k} w(r),$$

where $w(r)$ denotes the r-th element of weight vector $\boldsymbol{w}$. ϵ_{max} and ϵ_{min} are two hyper-parameters to control the range of $\epsilon_{ja}^{\boldsymbol{x}}$. In this paper, we set ϵ_{max} to 0.5 and ϵ_{min} to 0.

In fact, the function of Eq. (9) is like a normalization process, aiming to normalize the distance value $\langle \boldsymbol{w}, \boldsymbol{l}_{ja}^{\boldsymbol{x}} \rangle$ to the range $[\epsilon_{min}, \epsilon_{max}]$, which is similar to min-max normalization. Therefore, it can fine-tune Eq. (8) which serves as the primary component for augmenting features.

Thereafter, we combine $\delta_{ja}^{\boldsymbol{x}}$ and $\epsilon_{ja}^{\boldsymbol{x}}$ together:

$$\zeta_{ja}^{\boldsymbol{x}} = \delta_{ja}^{\boldsymbol{x}} + \epsilon_{ja}^{\boldsymbol{x}}. \tag{10}$$

After traversing all labels, we can obtain the following length-K vector containing similarity information:

$$\boldsymbol{z}_{\boldsymbol{x}} = [\underbrace{\zeta_{11}^{\boldsymbol{x}}, \zeta_{12}^{\boldsymbol{x}}, \ldots, \zeta_{1K_1}^{\boldsymbol{x}}}_{\text{The 1st dim.}}, \underbrace{\zeta_{21}^{\boldsymbol{x}}, \zeta_{22}^{\boldsymbol{x}}, \ldots, \zeta_{2K_2}^{\boldsymbol{x}}}_{\text{The 2nd dim.}}, \ldots, \underbrace{\zeta_{q1}^{\boldsymbol{x}}, \zeta_{q2}^{\boldsymbol{x}}, \ldots, \zeta_{qK_q}^{\boldsymbol{x}}}_{\text{The } q\text{th dim}}.]^\top. \tag{11}$$

3.4 Predictive model updating

For the predictive model via optimizing problem Eq. (3), to help improve its generalization performance with the extracted similarity information, we choose to manipulate the feature space. Specifically, for each instance $\boldsymbol{x}_i$, an augmented feature vector can generated as follows:

$$\boldsymbol{\xi}_i = \boldsymbol{z}_{\boldsymbol{x}_i}^I + \boldsymbol{z}_{\boldsymbol{x}_i}^L + \boldsymbol{p}_i. \tag{12}$$

Here, $\boldsymbol{z}_{\boldsymbol{x}_i}^I$ and $\boldsymbol{z}_{\boldsymbol{x}_i}^L$ are the instance-based and label-based versions of similarity vector for $\boldsymbol{x}_i$ in Eq. (11). $\boldsymbol{p}_i$ is the real-valued label vector defined in Eq.(4). It is worth noting that, since $\boldsymbol{p}_i$ is a probability vector while $\boldsymbol{z}_{\boldsymbol{x}_i}^I$ and $\boldsymbol{z}_{\boldsymbol{x}_i}^L$ are counting statistics, to let them share the same importance in augmented feature, both $\boldsymbol{z}_{\boldsymbol{x}_i}^I$ and $\boldsymbol{z}_{\boldsymbol{x}_i}^L$ need to be normalized in the range $[0, 1]$ before adding up. Then the MDML dataset can be transformed into the new version:

$$\widetilde{\mathcal{D}} = \{(\widetilde{\boldsymbol{x}}_i, \boldsymbol{l}_i) \mid 1 \leqslant i \leqslant m\}, \text{ where } \widetilde{\boldsymbol{x}}_i = [\boldsymbol{x}_i; \boldsymbol{\xi}_i], \tag{13}$$

where $[\boldsymbol{x}_i; \boldsymbol{\xi}_i]$ means concatenation. With the transformed dataset, the optimization problem Eq. (3) will be updated into ($1 \leqslant j \leqslant q$):

$$\min_{\boldsymbol{\Theta}^j} -\sum_{i=1}^{m}\sum_{a=1}^{K_j} l_{ia}^j \cdot \ln \frac{e^{\langle \boldsymbol{\theta}_a^j, \widetilde{\boldsymbol{x}}_i \rangle}}{\sum_{s=1}^{K_j} e^{\langle \boldsymbol{\theta}_s^j, \widetilde{\boldsymbol{x}}_i \rangle}} + \frac{\lambda}{2}\left\|\boldsymbol{\Theta}^j\right\|_F^2. \tag{14}$$

Here, we slightly abuse the notation because model parameter $\boldsymbol{\theta}_a^j$ is updated from length-d vector to length-$(d+K)$ one due to the changing of feature space, which can match the dimension update in Eq. (13).

With the transformed dataset $\widetilde{\mathcal{D}}$ and updated predictive model, the instance-based and label-based k nearest neighbors for any instance $\boldsymbol{x}$ will also be updated. In this paper, we iteratively conduct the three steps (including k nearest neighbors identification, similarity information extraction and predictive model updating) until convergence.

The Algorithm 1 shows the pseudo code of the working procedure of the SIDLE approach. Specifically, steps 1−10 initialize the predictive model over original dataset $\mathcal{D}$. Then steps 11−23 repeatedly update the predictive model over the transformed dataset $\widetilde{\mathcal{D}}$. Finally, the predicted label vector $\hat{\boldsymbol{l}}_*$ for unseen instance $\boldsymbol{x}_*$ is determined based on its augmented version $\widetilde{\boldsymbol{x}}_*$ and the converged predictive model.

Algorithm 1 The SIDLE approach

Input: The MDML training set $\mathcal{D}$, the number of nearest neighbors k, the unseen instance $\boldsymbol{x}_*$;
Output: The predicted label vector $\hat{\boldsymbol{l}}_*$ for $\boldsymbol{x}_*$;

1: **for** j = 1 to q **do**
2: Estimate $\boldsymbol{\Theta}^j$ with Eq.(3);
3: **end for**
4: **for** i = 1 to m **do**
5: Identify $\mathcal{N}_k^I(\boldsymbol{x}_i)$ with Eq.(2);
6: Obtain $\boldsymbol{p}_i$ with Eq.(4);
7: Identify $\mathcal{N}_k^L(\boldsymbol{x}_i)$ with Eq.(6);
8: Obtain $\boldsymbol{z}_{\boldsymbol{x}_i}^I$ and $\boldsymbol{z}_{\boldsymbol{x}_i}^L$ with Eq.(11);
9: Obtain $\boldsymbol{\xi}_i$ with Eq.(12);
10: **end for**
11: **repeat**
12: Form the transformed dataset $\widetilde{\mathcal{D}}$ with Eq.(13);
13: **for** j = 1 to q **do**
14: Update $\boldsymbol{\Theta}^j$ with Eq.(14) over $\widetilde{\mathcal{D}}$;
15: **end for**
16: **for** i = 1 to m **do**
17: Update $\mathcal{N}_k^I(\boldsymbol{x}_i)$ with Eq.(2) over $\widetilde{\mathcal{D}}$;
18: Update $\boldsymbol{p}_i$ with Eq.(4) over $\widetilde{\mathcal{D}}$;
19: Update $\mathcal{N}_k^L(\boldsymbol{x}_i)$ with Eq.(6);
20: Update $\boldsymbol{z}_{\boldsymbol{x}_i}^I$ and $\boldsymbol{z}_{\boldsymbol{x}_i}^L$ with Eq.(11);
21: Update $\boldsymbol{\xi}_i$ with Eq.(12);
22: **end for**
23: **until** Convergence
24: Obtain the augmented version $\widetilde{\boldsymbol{x}}_*$;
25: Return $\hat{\boldsymbol{l}}_*$ based on $\widetilde{\boldsymbol{x}}_*$ and the learned model.

4 Experiments

4.1 Experimental setup

In this paper, we collect five benchmark MDML datasets to conduct comparative study. Table 3 summarizes their detailed characteristics, including number of dimensions (#Dimension), number of labels per dimension (#Label/Dim.), number of features (#Feature), number of samples (#Sample), and their application domain.

Song data sets are the data from Chinese songs, each dimension corresponds to one kind of semantics, i.e., emotion, genre, and scenario [1]. The label organization strategy for Song-v1 is that the label be regarded to be relevant to one instance is the one has the largest confidence value within dimension. For Song-v2, assume that the labels in the same dimension are sorted in descending order according to confidence values, and therefore the relevant labels are those that appear before the largest difference between two adjacent labels.

Yeast data sets are collected about budding yeast *Saccharomyces cerevisiae* [31]. Dimensions correspond to alpha factor arrest & release, cdc15 arrest & release, elutriation, diauxic shift, heat shock, and sporulation. For Yeast-v1, if the current gene expression level is larger than the average level in the biological experiment, it will be considered as relevant. For Yeast-v2, like mentioned in Song-v2, the label will be treated as relevant if it is in the front of the position of the largest different between two adjacent time points (ordered).

The Flickr dataset is a set of pictures in mirflickr25k [32] categorized by different dimensions. Each example corresponds to one picture and total of three dimensions correspond to circumstance, item, and light, respectively. The circumstance includes sky condition, indoor or outdoor, and whether the scene is far or close. The item includes whether there are people in the scene and their genders, and other decorative items like plants and others. The light dimension includes the light condition of the picture, e.g., daytime, night, or cannot decide. The label will be regarded as a relevant one if the scenario actually appears in the picture.

To evaluate the performance of different MDML approaches, five popular multi-label evaluation metrics are utilized in experiments, including Hamming loss, ranking loss, coverage, one error, and average precision. Their definitions can be easily found in many literatures [14,33], so we omit them in this paper. We compute the value of these metrics dimension by dimension, and then report the average value of all dimensions.

The performance is compared with five competing approaches, including CLIM [1], Multi-Label k-Nearest Neighbors (MLKNN) [34], Binary Relevance (BR) [35], Classifier Chains (CC) [36], and wrapping multi-label classification with label-specific features generation (WRAP) [37]. Specifically, CLIM is an MDML approach which considers label correlations within individual dimension and across multiple dimensions in different ways. All the remaining four competing approaches are proposed for multi-label classification. They can be used to solve MDML problem by transforming a MDML problem to a multi-label one via simply concatenating all label spaces as an entirety. Here, we include them because there are not enough MDML approaches currently. MLKNN is based on k-nearest neighbors techniques. BR learns a binary classifier for each label independently while CC learns a chain of binary classifier in a cascaded manner. WRAP learns multi-label predictive model based on label-specific features generation technique.

Table 3 Detailed information of the employed benchmark MDML datasets

Dataset	#Dimension	#Label/Dim.	#Feature	#Sample	Domain
Song-v1/v2	3	11/10/18	98	785	Music
Yeast-v1/v2	6	18/15/14/7/6/6	24	2465	Biology
Flickr	3	8/7/3	1536	12198	Image

The parameters for each approach are set as suggested in their original literature. Specifically, $\lambda = 2^{-3}$ for CLIM, $k = 10$ for MLKNN, $\alpha = 0.9$, $\lambda_1 = 5$, $\lambda_2 = 5$, $\lambda_3 = 0.1$ for WRAP. For BR and CC that necessitate a base binary classifier, we use cross entropy-based logistic regression for fair comparison, which is implemented by the popular LIBLINEAR software [38]. For our proposed SIDLE approach, $\lambda = 2^{-3}$ and $k = 10$ which are consistent to CLIM and MLKNN respectively. Ten-fold cross-validation is conducted over each dataset and we report the mean metric value as well as standard deviation.

All experiments were conducted on a Windows 11 operating system using MATLAB R2023b. The hardware used in the system includes an Intel i7-12700 CPU and 32GB of memory.

4.2 Experimental results

The detailed results can be found in Table 4. To facilitate comparison, the performance ranks are also shown in subscript style and the best performance is shown in boldface.

According to these experiment results, the following observations can be made:

- Across all 25 configurations (5 metrics × 5 datasets), the proposed SIDLE approach achieves the best performance in 24 cases and the only rest one ranks the second.
- Compared with the up-to-date approach CLIM, the proposed SIDLE approach achieves superior performance in terms of all the five evaluation metrics. While CLIM enhances feature representations by augmenting the feature space with predictions from each label space to capture label correlations, it relies solely on label-based information. In contrast, SIDLE integrates both instance-based and label-based similarity information. It identifies k-nearest neighbors in the original instance space and in an enhanced label space, extracting similarity features via counting statistics with distance-based biasing. This dual approach enriches the feature space, capturing nuanced inter-instance and inter-label similarities for superior classification performance. Empirical results across metrics and datasets consistently show that SIDLE outperforms CLIM, underscoring the benefit of complementary similarity cues.
- Compared with the popular MLKNN approach, the proposed SIDLE approach achieves overwhelming superior performance

Table 4 Experimental results (mean±std) of the proposed SIDLE approach and the five compared approaches. In addition, the performance ranks of all compared approaches are also shown in subscript style and the best performance is shown in boldface. The arrow before each metric name indicates the trend for better performance where ↑ (↓) means the higher (lower) the better

Dataset	SIDLE	CLIM	MLKNN	BR	CC	WRAP
			(↓) Hamming loss			
Song-v1	**0.174±0.007**$_1$	0.176±0.007$_2$	0.179±0.007$_3$	0.190±0.014$_6$	0.187±0.013$_4$	0.189±0.006$_5$
Song-v2	**0.126±0.006**$_1$	**0.126±0.006**$_1$	0.129±0.007$_3$	0.204±0.007$_5$	0.204±0.008$_5$	0.158±0.008$_4$
Yeast-v1	**0.403±0.005**$_1$	0.406±0.007$_2$	0.407±0.006$_3$	0.408±0.008$_4$	0.411±0.005$_5$	0.422±0.005$_6$
Yeast-v2	0.271±0.005$_2$	0.271±0.004$_2$	**0.270±0.004**$_1$	0.482±0.011$_5$	0.486±0.010$_6$	0.288±0.004$_4$
Flickr	**0.133±0.003**$_1$	0.137±0.003$_2$	0.138±0.005$_3$	0.236±0.006$_5$	0.235±0.008$_6$	0.177±0.004$_4$
			(↓) Ranking loss			
Song-v1	**0.133±0.009**$_1$	0.138±0.008$_2$	0.138±0.008$_2$	0.207±0.025$_6$	0.200±0.019$_5$	0.157±0.011$_4$
Song-v2	**0.111±0.009**$_1$	0.117±0.010$_2$	0.117±0.011$_2$	0.500±0.065$_5$	0.502±0.060$_6$	0.126±0.011$_4$
Yeast-v1	**0.368±0.009**$_1$	0.372±0.009$_2$	0.373±0.007$_3$	0.378±0.011$_4$	0.382±0.009$_5$	0.389±0.008$_6$
Yeast-v2	**0.336±0.006**$_1$	0.345±0.006$_3$	0.342±0.008$_2$	0.560±0.009$_5$	0.560±0.009$_5$	0.351±0.006$_4$
Flickr	**0.100±0.003**$_1$	0.109±0.002$_2$	0.110±0.012$_3$	0.479±0.042$_5$	0.481±0.055$_6$	0.117±0.009$_4$
			(↓) Coverage			
Song-v1	**0.457±0.011**$_1$	0.468±0.011$_3$	0.464±0.013$_2$	0.523±0.021$_6$	0.515±0.015$_5$	0.497±0.014$_4$
Song-v2	**0.230±0.017**$_1$	0.237±0.019$_2$	0.239±0.018$_3$	0.532±0.054$_5$	0.534±0.048$_6$	0.244±0.017$_4$
Yeast-v1	**0.715±0.005**$_1$	0.717±0.005$_2$	0.718±0.003$_3$	0.718±0.006$_3$	0.721±0.005$_5$	0.732±0.004$_6$
Yeast-v2	**0.480±0.007**$_1$	0.490±0.006$_3$	0.487±0.008$_2$	0.644±0.008$_5$	0.644±0.007$_5$	0.490±0.006$_3$
Flickr	**0.244±0.003**$_1$	0.254±0.002$_2$	0.259±0.014$_3$	0.567±0.055$_5$	0.567±0.044$_5$	0.266±0.003$_4$
			(↓) One error			
Song-v1	**0.084±0.019**$_1$	0.087±0.016$_2$	0.093±0.018$_3$	0.284±0.101$_6$	0.282±0.085$_5$	0.101±0.018$_4$
Song-v2	**0.256±0.018**$_1$	**0.256±0.015**$_1$	0.261±0.021$_3$	0.986±0.008$_5$	0.986±0.006$_5$	0.288±0.020$_4$
Yeast-v1	**0.318±0.017**$_1$	0.321±0.013$_2$	0.327±0.015$_4$	0.335±0.018$_5$	0.344±0.017$_6$	0.323±0.014$_3$
Yeast-v2	**0.552±0.012**$_1$	0.562±0.016$_3$	0.555±0.016$_2$	0.826±0.010$_6$	0.825±0.011$_5$	0.564±0.017$_4$
Flickr	**0.120±0.006**$_1$	0.134±0.002$_2$	0.135±0.008$_3$	0.244±0.017$_5$	0.247±0.011$_6$	0.141±0.006$_4$
			(↑) Average precision			
Song-v1	**0.841±0.009**$_1$	0.834±0.011$_2$	0.832±0.008$_3$	0.721±0.037$_6$	0.730±0.026$_5$	0.813±0.012$_4$
Song-v2	**0.782±0.012**$_1$	0.778±0.009$_2$	0.775±0.012$_3$	0.251±0.026$_5$	0.249±0.026$_6$	0.751±0.015$_4$
Yeast-v1	**0.711±0.005**$_1$	0.709±0.006$_2$	0.707±0.005$_3$	0.703±0.007$_4$	0.700±0.006$_5$	0.698±0.006$_6$
Yeast-v2	**0.577±0.008**$_1$	0.561±0.010$_4$	0.574±0.010$_2$	0.383±0.007$_5$	0.383±0.007$_5$	0.565±0.010$_3$
Flickr	**0.891±0.003**$_1$	0.882±0.002$_2$	0.879±0.004$_3$	0.875±0.007$_4$	0.875±0.012$_4$	0.870±0.012$_6$

in terms of each metric. MLKNN works based on the similarity information in instance space, SIDLE's superiority shows the necessity of leveraging label-based similarity information.

- Both BR and CC works based on learning a binary classifier for each label and do not utilize any similarity information. It can be seen that, by introducing similarity information into the process of predictive model induction, the proposed SIDLE approach achieves superior performance against them in terms of all the five evaluation metrics.
- Compared with the WRAP approach, even it is the state-of-the-art baseline on multi-label classification problems, it still achieves inferior performance to the proposed SIDLE approach. The possible reason is that, the WRAP approach ignores the heterogeneous nature in MDML problems which

degenerates its performance.

- Both Song and Yeast datasets are collected form real-world information, and two version of them have different organized label space. The result shows that both MDML approaches perform better than traditional multi-label approaches, which means considering dimension correlations will help improve the performance for MDML missions. Flickr dataset is also real-world dataset but with relatively bigger size of features and instance numbers. The result shows that for large data, MDML approaches can still have better performance than traditional multi-label approaches, showing the high robustness of proposed approach. From the analysis of these datasets, it can be found that take the new nature of MDML framework into account will significantly help improving the final performance for classification approach.
- CLIM has already demonstrated that label space can enhance prediction performance, and SIDLE extends this by showing that both label space and instance space play similar roles in the MDML learning process. The performance improvements in SIDLE validate that combining label-based and instance-based similarity information can enhance the feature representations from different instances and then improve classification performance.

4.3 Further analysis

4.3.1 Ablation study

To validate the effectiveness of the technical design for SIDLE, ablation study is conducted in this section. Specifically, SIDLE is compared with its four variants regarding the way of constructing the augmented features in Eq. (12). They are denoted by SIDLE$_I$, SIDLE$_L$, SIDLE$_C$, SIDLE$_H$, respectively. SIDLE$_I$ and SIDLE$_L$ construct the augmented features with only instance-based similarity information $\boldsymbol{z}^I_{\boldsymbol{x}_i}$ and label-based similarity information. SIDLE$_C$ constructs the augmented features in a concatenated manner, i.e., $\boldsymbol{\xi}_i = [\boldsymbol{z}^I_{\boldsymbol{x}_i}; \boldsymbol{z}^L_{\boldsymbol{x}_i}; \boldsymbol{p}_i]$, while SIDLE$_H$ constructs the augmented features in a hybrid manner (addition+concatenation), i.e., $\boldsymbol{\xi}_i = [\boldsymbol{z}^I_{\boldsymbol{x}_i} + \boldsymbol{z}^L_{\boldsymbol{x}_i}; \boldsymbol{p}_i]$.

The detailed results can be found in Table 5. To facilitate comparison, the best performance is shown in boldface. It can be observed that, across all 25 configurations (5 metrics × 5 datasets), the proposed SIDLE approach achieves the best performance in 13 cases. Although SIDLE may not achieve the top score for every metric, it consistently performs well, with only minor gaps compared to the best-performing versions on specific metrics. When averaging these metrics and ranking them, SIDLE emerges as the best overall choice.

It is worth noting that SIDLE$_H$ outperforms SIDLE$_C$. The reason this phenomenon happens is complicated, and some possible factors are listed as follows. 1) Since both instance-based and label-based similarity information is derived from neighbor information, they may have redundant or overlapping features. Therefore, simply concatenating them together may lead to feature redundancy. In contrast, adding them together inherently merges shared neighborhood information while suppressing redundant components, where dimensionality reduction mitigates the risk of overfitting and improves computational efficiency. 2) The noises in either instance-based or label-based similarity information will be amplified if concatenation applied. The addition of instance-based and label-based similarity information also acts as a linear smoothing mechanism which can cancel out erratic variations while reinforcing stable similarity information.

4.3.2 Parameter sensitivity analysis

In our proposed SIDLE approach, the similarity information is extracted in the k nearest neighbors. To investigate how does this parameter affect the performance of SIDLE, we repeat the experiments of SIDLE with k increasing from 6 to 14 over all datasets.

Figure 1 illustrates the performance fluctuation of SIDLE with varying value of k. It can be observed that the proposed SIDLE approach achieves relatively stable performance when changing the parameter value of k. The reason why k value does not affect the final performance a lot is about the application of weighted kNN strategy, which can minimize the effect of value selection of k because a limited number of close neighbors is enough to make the final judgment. In previous experiments, the value of k is simply fixed as a moderate value of 10 which can be used as a default setting.

Another parameter sensitivity experiment is about the threshold ϵ_{max} and ϵ_{min} in Eq. (9). From Fig. 2, it can be found that when the threshold $(\epsilon_{max} - \epsilon_{min})$ is bigger, the metrics will be slightly increased (performs better). Since the threshold is greater than 0.5, the performance remains stable. Therefore in previous experiments the value of ϵ_{max} and ϵ_{min} are fixed as 0.5 and 0. In addition, the selection of the position of the range of $[\epsilon_{min}, \epsilon_{max}]$ in $[0,1]$ will not heavily affect the performance as well (i.e., $[0,0.5]$ and $[0.5,1]$ will return similar results in metrics), which leads to fixing ϵ_{min} at 0 because it simplifies Eq. (9) to a single term. In fact, the function of Eq. (9) is similar to min-max normalization, and $\epsilon_{max} - \epsilon_{min}$ act as thresholds for the normalization.

4.3.3 Complexity analysis

The majority of the complexity of SIDLE is from optimizing the Eq. (3), which applies gradient descent. Since the number of iterations that gradient descent used to converge is hard to theoretically analyze because it depends on the objective function. Here we measure the time costs of SIDLE and the comparable approach CLIM under the same hardware and software conditions. All approaches can be done on a computer with at least 16 GB RAM size, and the experiment is performed in a 32 GB RAM hardware environment. The result can be seen in Table 6.

It can be shown that SIDLE is more efficient than the compared approach CLIM, which enjoys a comparable time with the traditional approach WRAP and longer than other multi-label approaches.

5 Conclusion

The main contribution of this paper can be summarized from the following two aspects. 1) From a similarity-based learning perspective, we propose that it can help induce better similarity-based predictive models by simultaneously leveraging instance-based

Table 5 Ablation study result (mean±std) of the proposed SIDLE approach and its degenerated versions. In addition, the performance ranks of all compared approaches are also shown in subscript style and the best performance is shown in boldface. The arrow before each metric name indicates the trend for better performance where ↑ (↓) means the higher (lower) the better

Dataset	SIDLE	SIDLE$_I$	SIDLE$_L$	SIDLE$_C$	SIDLE$_H$
			(↓) Hamming loss		
Song-v1	**0.174±0.007**	0.175±0.006	0.179±0.006	0.176±0.007	0.176±0.007
Song-v2	0.126±0.006	0.127±0.007	0.129±0.005	0.126±0.007	**0.125±0.007**
Yeast-v1	0.403±0.005	**0.401±0.005**	0.418±0.006	0.402±0.005	0.404±0.006
Yeast-v2	**0.271±0.005**	0.272±0.004	0.275±0.004	**0.271±0.004**	**0.271±0.004**
Flickr	**0.133±0.003**	0.134±0.003	**0.133±0.004**	0.136±0.007	0.134±0.003
			(↓) Ranking loss		
Song-v1	**0.133±0.009**	0.134±0.009	0.140±0.011	0.132±0.008	0.133±0.010
Song-v2	**0.111±0.009**	0.118±0.009	0.124±0.006	0.115±0.009	**0.111±0.009**
Yeast-v1	0.368±0.009	**0.367±0.008**	0.384±0.010	0.369±0.007	0.369±0.007
Yeast-v2	**0.336±0.006**	0.340±0.005	0.349±0.006	0.339±0.006	**0.336±0.006**
Flickr	**0.100±0.003**	0.106±0.009	0.110±0.010	0.108±0.007	0.108±0.006
			(↓) Coverage		
Song-v1	**0.457±0.011**	0.465±0.012	0.477±0.015	0.462±0.010	0.461±0.012
Song-v2	**0.230±0.017**	0.239±0.017	0.246±0.014	0.235±0.017	0.231±0.016
Yeast-v1	0.715±0.005	**0.714±0.005**	0.725±0.005	0.715±0.004	0.715±0.003
Yeast-v2	**0.480±0.007**	0.485±0.007	0.490±0.006	0.482±0.008	0.481±0.006
Flickr	**0.244±0.003**	0.257±0.004	0.249±0.006	0.246±0.008	0.250±0.004
			(↓) One error		
Song-v1	**0.084±0.019**	0.086±0.016	**0.084±0.012**	**0.084±0.016**	**0.084±0.021**
Song-v2	0.256±0.018	0.255±0.017	0.266±0.013	0.254±0.019	**0.253±0.022**
Yeast-v1	0.318±0.017	**0.315±0.013**	0.321±0.013	0.317±0.014	0.318±0.017
Yeast-v2	**0.552±0.012**	0.561±0.012	0.568±0.014	0.558±0.012	0.554±0.011
Flickr	0.120±0.006	0.122±0.014	0.121±0.010	**0.119±0.009**	0.121±0.018
			(↑) Average precision		
Song-v1	**0.841±0.009**	0.837±0.010	0.832±0.011	0.839±0.009	0.838±0.010
Song-v2	**0.782±0.012**	0.776±0.012	0.765±0.009	0.778±0.012	**0.782±0.012**
Yeast-v1	0.711±0.005	**0.713±0.005**	0.702±0.006	0.711±0.004	0.711±0.005
Yeast-v2	**0.577±0.008**	0.566±0.008	0.558±0.009	0.569±0.008	0.574±0.008
Flickr	**0.891±0.003**	0.889±0.007	0.890±0.004	0.886±0.009	0.886±0.008

and label-based similarity information. Moreover, different from nearest class mean, we introduce another effective strategy to leverage label-based similarity information. 2) From the MDML perspective, we propose a novel approach named SIDLE that utilizes both instance-based and label-based similarity information. The experimental results clearly demonstrate its superiority over the competing baselines.

Looking ahead, one promising direction for SIDLE is to refine similarity information extraction by exploring adaptive or learned distance metrics (metric learning) and sophisticated weight schemes for kNN statistics. These enhancements could better capture the nuances of label and instance relationships and boost classification performance. Moreover, similarity information strategies can extend to complex classification tasks and other machine learning schemes,

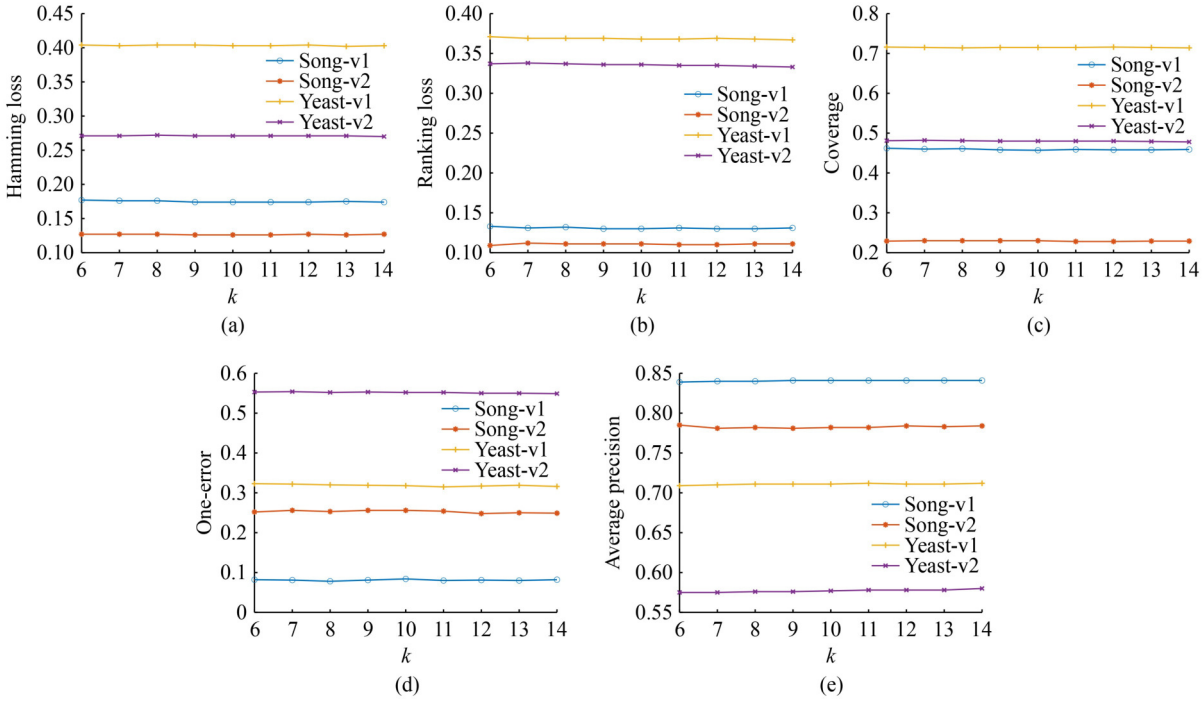

Fig. 1 Performance of SIDLE changes as the value of k increases from 6 to 14. (a) Hamming loss; (b) ranking loss; (c) coverage; (d) one error; (e) average precision

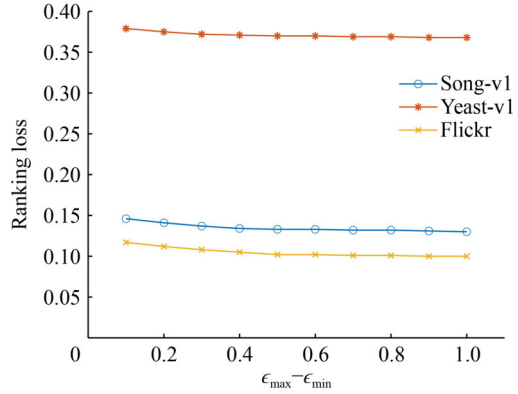

Fig. 2 Performance of SIDLE changes as the value $\epsilon_{max}-\epsilon_{min}$ increases from 0 to 1 (take three datasets, and ranking loss as the test metric)

Table 6 The time consumption (s) of SIDLE and compared approaches

Dataset	SIDLE	CLIM	WRAP	BR	CC
Song-v1	23	13	34	2	6
Song-v2	15	9	48	2	6
Yeast-v1	187	280	106	4	17
Yeast-v2	130	53	126	5	14
Flickr	3074	2290	4416	95	144

such as semi-supervised, unsupervised, or deep feature augmentation learning. Broadening this approach to future works could improve SIDLE's performance and underscore the MDML paradigm's versatility in multi-dimensional and multi-label challenges.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62225602, 62306131) and the Big Data Computing Center of Southeast University, China.

Competing interests

Min-Ling ZHANG is an Action Editor of the journal and a co-author of this article. To minimize bias, he was excluded from all editorial decision-making related to the acceptance of this article for publication. The remaining authors declare no conflict of interest.

Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

[1] Jia B B, Zhang M L. Multi-dimensional multi-label classification: Towards encompassing heterogeneous label spaces and multi-label

annotations. Pattern Recognition, 2023, 138: 109357
[2] Song L, Liu J, Qian B, Sun M, Yang K, Sun M, Abbas S. A deep multi-modal CNN for multi-instance multi-label image classification. IEEE Transactions on Image Processing, 2018, 27(12): 6025−6038
[3] Yin T, Chen H, Wang Z, Li T. Missing labels feature selection based on multilabel multi-scale fusion fuzzy rough sets. In: Proceedings of the 3rd International Conference on Digital Society and Intelligent Systems. 2023, 348−352
[4] He X, Ma P, Chen Y, Liu Y. MOD-YOLO: improved YOLOv5 based on multi-softmax and omni-dimensional dynamic convolution for multi-label bridge defect detection. In: Proceedings of the 20th International Conference on Advanced Intelligent Computing Technology and Applications. 2024, 44−55
[5] Yin T, Chen H, Wang Z, Liu K, Yuan Z, Horng S J, Li T. Feature selection for multilabel classification with missing labels via multi-scale fusion fuzzy uncertainty measures. Pattern Recognition, 2024, 154: 110580
[6] Xiao Y, Liu B, Hao Z, Cao L. A similarity-based classification framework for multiple-instance learning. IEEE Transactions on Cybernetics, 2014, 44(4): 500−515
[7] Rossi R A, Ahmed N K, Eldardiry H, Zhou R. Similarity-based multi-label learning. In: Proceedings of 2018 International Joint Conference on Neural Networks. 2018, 1−8
[8] Jia B B, Zhang M L. Multi-dimensional classification via *k*NN feature augmentation. Pattern Recognition, 2020, 106: 107423
[9] Jia B B, Zhang M L. MD-KNN: An instance-based approach for multi-dimensional classification. In: Proceedings of the 25th International Conference on Pattern Recognition. 2021, 126−133
[10] Ma Z, Chen S. A similarity-based framework for classification task. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(5): 5438−5443
[11] Qin Y, Yang J, Zhou J, Pu H, Mao Y. A new supervised multi-head self-attention autoencoder for health indicator construction and similarity-based machinery RUL prediction. Advanced Engineering Informatics, 2023, 56: 101973
[12] Foumani N M, Tan C W, Webb G I, Rezatofighi H, Salehi M. Series2vec: similarity-based self-supervised representation learning for time series classification. Data Mining and Knowledge Discovery, 2024, 38(4): 2520−2544
[13] Gou Q, Dong Y, Wu Y, Ke Q. Semantic similarity-based program retrieval: a multi-relational graph perspective. Frontiers of Computer Science, 2024, 18(3): 183209
[14] Zhang M L, Zhou Z H. A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(8): 1819−1837
[15] Liu W, Wang H, Shen X, Tsang I W. The emerging trends of multi-label learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(11): 7955−7974
[16] Si C, Jia Y, Wang R, Zhang M L, Feng Y, Qu C. Multi-label classification with high-rank and high-order label correlations. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(8): 4076−4088
[17] Wang Y B, Hang J Y, Zhang M L. Multi-label open set recognition. In: Proceedings of the 38th Conference on Neural Information Processing Systems. 2024
[18] Tang W, Zhang W, Zhang M L. Multi-instance partial-label learning: towards exploiting dual inexact supervision. Science China Information Sciences, 2024, 67(3): 132103
[19] Gil-Begue S, Bielza C, Larrañaga P. Multi-dimensional Bayesian network classifiers: a survey. Artificial Intelligence Review, 2021, 54(1): 519−559
[20] Jia B B, Zhang M L. Multi-dimensional classification: paradigm, algorithms and beyond. Vicinagearth, 2024, 1(1): 3
[21] Jia B B, Zhang M L. Multi-dimensional classification via decomposed label encoding. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(2): 1844−1856
[22] Huang T, Jia B B, Zhang M L. Deep multi-dimensional classification with pairwise dimension-specific features. In: Proceedings of the 33rd International Joint Conference on Artificial Intelligence. 2024, 4183−4191
[23] Wang H, Chen C, Liu W, Chen K, Hu T, Chen G. Incorporating label embedding and feature augmentation for multi-dimensional classification. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence. 2020, 6178−6185
[24] Cover T M, Hart P E. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 1967, 13(1): 21−27
[25] Datta P, Kibler D F. Symbolic nearest mean classifiers. In: Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference. 1997, 82−87
[26] Guerriero S, Caputo B, Mensink T. DeepNCM: Deep nearest class mean classifiers. In: Proceedings of the 6th International Conference on Learning Representations. 2018
[27] Zhang Z, Zou Q, Lin Y, Chen L, Wang S. Improved deep hashing with soft pairwise similarity for multi-label image retrieval. IEEE Transactions on Multimedia, 2020, 22(2): 540−553
[28] Shi Y, Ye H, Man D, Han X, Zhan D, Jiang Y. Revisiting multi-dimensional classification from a dimension-wise perspective. Frontiers of Computer Science, 2025, 19(1): 191304
[29] Ding H, Takigawa I, Mamitsuka H, Zhu S. Similarity-based machine learning methods for predicting drug-target interactions: a brief review. Briefings in Bioinformatics, 2014, 15(5): 734−747
[30] Xu N, Liu Y P, Geng X. Label enhancement for label distribution learning. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(4): 1632−1643
[31] Eisen M B, Spellman P T, Brown P O, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences, 1998, 95(25): 14863−14868
[32] Huiskes M J, Lew M S. The MIR flickr retrieval evaluation. In: Proceedings of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval. 2008, 39−43
[33] Wu X Z, Zhou Z H. A unified view of multi-label performance measures. In: Proceedings of the 34th International Conference on Machine Learning. 2017, 3780−3788
[34] Zhang M L, Zhou Z H. ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition, 2007, 40(7): 2038−2048
[35] Zhang M L, Li Y K, Liu X Y, Geng X. Binary relevance for multi-label learning: an overview. Frontiers of Computer Science, 2018, 12(2): 191−202
[36] Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. Machine Learning, 2011, 85(3): 333−359
[37] Yu Z B, Zhang M L. Multi-label classification with label-specific feature generation: a wrapped approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(9): 5199−5210
[38] Fan R E, Chang K W, Hsieh C J, Wang X R, Lin C J.

LIBLINEAR: a library for large linear classification. The Journal of Machine Learning Research, 2008, 9: 1871−1874

Zi-Zhan GU received his BSc degree in aeronautics and astronautics engineering from Purdue University, USA in 2020 and M.Sc. degree in computer control engineering from University of Technology Sydney, Australia in 2022. Currently, he is a PhD student in software engineering at the School of Computer Science and Engineering, Southeast University, China. His main research interests are machine learning, especially in multi-label and multi-dimension classification.

Bin-Bin JIA received his bachelor's degree in electronic information science and technology from North China Electric Power University, China in 2010, his master's degree in information and communication engineering from Beihang University, China in 2013, and his PhD degree in software engineering from Southeast University, China in 2022. He joined the College of Electrical and Information Engineering, Lanzhou University of Technology, China in 2013 and is an associate professor currently. In recent years, Dr. Jia has served as the PC member of ICML, NeurIPS, ICLR, AAAI, IJCAI, KDD, etc. His main research interests include machine learning and data mining.

Min-Ling ZHANG received the BSc, MSc, and PhD degrees in computer science from Nanjing University, China in 2001, 2004, and 2007, respectively. Currently, he is a professor at the School of Computer Science and Engineering, Southeast University, China. His main research interests include machine learning and data mining. In recent years, Dr. Zhang has served as the General Co-Chairs of ACML'18, Program Co-Chairs of PAKDD'19, CCF-ICAI'19, ACML'17, CCFAI'17, PRICAI'16, Senior PC member or Area Chair of AAAI 2022−2025, IJCAI 2017−2025, KDD 2021−2025, ICML 2024−2025, etc. He is also on the editorial board of IEEE Transactions on Pattern Analysis and Machine Intelligence, ACM Transactions on Information Systems, ACM Transactions on Intelligent Systems and Technology, Science China Information Sciences, Frontiers of Computer Science, Machine Intelligence Research, etc. Dr. Zhang is the Steering Committee Member of ACML and PAKDD, Vice Chair of the CAAI Machine Learning Society. He is a Distinguished Member of CCF, CAAI, and Senior Member of AAAI, ACM, IEEE.