International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-023-01858-x

ORIGINAL ARTICLE

Label-dependent feature exploration for label distribution learning

Run-Ting Bai[1] · Heng-Ru Zhang[1] · Fan Min[1,2]

Received: 18 September 2022 / Accepted: 11 May 2023

Abstract
Label distribution learning (LDL) explicitly models label ambiguity by assigning a real-valued vector with label description degrees to each sample. Most LDL methods only build models on the same feature (sub)space shared by all labels. However, they ignore that each label has its own specific features, and there are some common features among labels. In this paper, we propose a novel LDL (LDL-LDF) algorithm that aims to exploit both label-dependent and common features. First, label-dependent feature reconstruction utilizes thresholding for relevant sample subset identification, density peaks clustering for representative sample selection, and Euclidean distance for feature value calculation. Second, common feature reconstruction follows a similar approach, however, on the whole dataset. Finally, the prediction neural network is composed of several components that serve each label with label-dependent features, one component that serves all labels with common features, and the fusion component. The effectiveness and competitiveness of our algorithm are verified through various experiments comparing seven algorithms on fourteen real-world datasets.

Keywords Common feature · Label-dependent feature · Label distribution learning · Representative sample selection.

1 Introduction

In multi-label learning (MLL) [1], the correspondence between samples and labels is modeled by 0/1. However, there is often uncertainty about the relationship between samples and labels, a phenomenon known as "label ambiguity" [2]. To better model this relationship, a novel learning paradigm known as label distribution learning (LDL) is proposed. LDL assigns each sample a label description vector, where elements indicate how well the label describes the sample. It has been successfully applied in a wide range of practical scenarios, including age estimation [3, 4], emotion recognition [5, 6], head-pose estimation [7], crowd counting [8], beauty perception [9], sentiment analysis [10], and so on.

Most existing LDL models are built on the same feature (sub)space shared by all labels. They ignore that different labels have their specific features. It is obvious that building models only in the same feature space is not conducive to the discrimination of different labels. Figure 1 shows an example of a scene image. In Fig. 1**a**, the feature "sea buckthorn" plays an important role in distinguishing the labels "desert" and "non-desert". In Fig. 1**b**, the feature "water" plays an important role in distinguishing the labels "beach" and "non-beach". At the same time, these labels also share some features, like "sand". Therefore, mining the latent specific features of each label and the common features of all labels is helpful for label learning.

In this paper, we propose a label distribution learning (LDL-LDF) algorithm with label-dependent and common features. Inspired by the multi-label algorithm LIFT [11], we construct label-dependent features through cluster analysis. Due to the particularity of label distribution, the distinction between relevant and irrelevant labels is relative [12]. In many cases, there is no clear boundary between relevant and irrelevant labels, and their distinction usually depends on the choice of threshold. Therefore, different from clustering in the positive and negative samples in

These authors contributed equally to this work.

✉ Heng-Ru Zhang
zhanghrswpu@163.com

Run-Ting Bai
bairunting@gmail.com

Fan Min
minfan@swpu.edu.cn

[1] School of Computer Science, Southwest Petroleum University, ChengDu 610500, China

[2] Institute for Artificial Intelligence, Southwest Petroleum University, ChengDu 610500, China

Published online: 12 June 2023

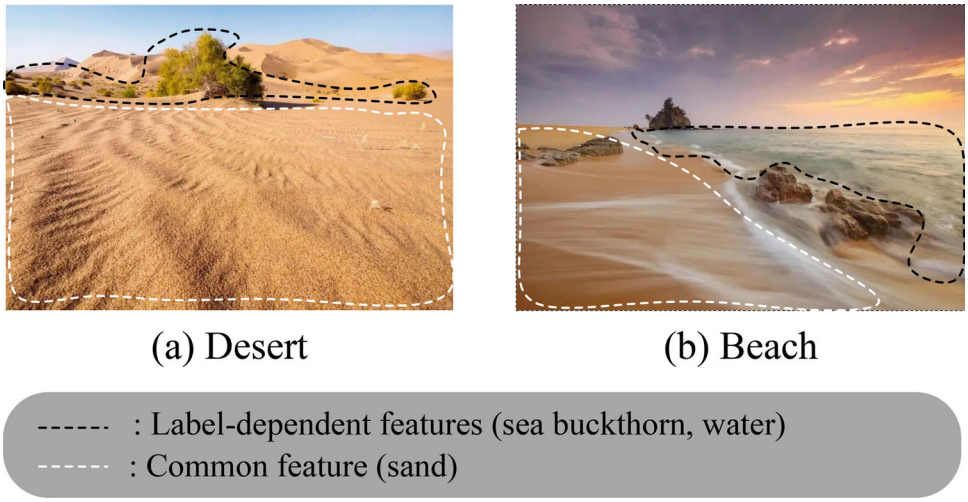


Fig. 1 An illustrative example for the label-dependent and common features. **a** the feature "sea buckthorn" plays an important role in distinguishing the labels "desert" and "non-desert". **b** the feature "water" plays an important role in distinguishing the labels "beach" and "non-beach". For such features, we consider them as label-dependent features. Meanwhile, the feature "sand" shared by the labels are considered to be a common feature

LIFT, we only select samples that have a certain correlation with the label for cluster analysis. In this way, the computational burden and adverse effects of irrelevant samples can be mitigated.

Specifically, we mine label-dependent features by clustering samples related to corresponding labels. The commonality of these samples reflects the hidden features of the corresponding labels. First, we binarize the label distribution [13] through a threshold to select the sample set related to each label. Second, cluster analysis is performed on local samples, and representative samples of each label are selected. Finally, we generate label-dependent feature vectors by computing the distance of each sample to representative samples. In addition, we also consider common features that are discriminative for all labels, which LIFT does not consider. Likewise, we obtain the representative samples by clustering the global samples. The distance between each sample and representative samples is then calculated to generate a common feature vector.

For prediction, we design different networks for two kinds of features. Specifically, parallel networks make separate predictions on labels, where the input to each network is the label-dependent features of the corresponding label. We then concatenate and normalize the predictions of each label to obtain the predicted label distribution. Furthermore, a single network jointly predicts all labels using common features to directly outputting a label distribution. Finally, we use an attention mechanism to fuse the predicted label distributions obtained using different features. By considering both label-dependent features and common features, a more efficient label distribution learning method is proposed. Moreover, we also consider label correlation to improve performance.

The main contributions of this paper are summarized as follows:

- We mine label-dependent features and common features and design different networks to leverage both for more accurate predictions.
- We perform clustering on the set of relevant samples for each label, and select representative samples of each label to reconstruct label-dependent features to deeply explore potential specific information of labels.
- We perform clustering among all samples and select representative samples to reconstruct common features to deeply mine the commonality of labels.

The rest of the paper is organized as follows: Section 2 briefly reviews of some related works. Section 3 describes the proposed algorithm. Section 4 reports the experiments on 14 real-world data sets. Finally, Section 5 concludes this paper.

2 Related works

LDL is dedicated to learning the importance of labels to samples, which is more in line with the relationship between labels and samples in real scenes than MLL. Label distribution data in the real world is difficult to obtain directly. Therefore, many label enhancement algorithms [14, 15] have been proposed aiming at recovering label distribution from logical labels. This enables the label distribution to be better applied to various domains. Therefore, the development of label distribution learning is more vigorous. In recent years, more and more studies on LDL have emerged. Existing LDL methods can be roughly divided into three categories according to the following different strategies.

Problem Transformation (PT) methods transform the LDL problem into single label learning (SLL) or MLL problem to apply the existing SLL or MLL methods. For example, PT-Bayes and PT-SVM [16] convert the training

samples with label distributions into weighted single-label samples, and then adopt Bayes classifier and SVM to deal with single-label problems. Algorithm Adaptation (AA) methods extend existing traditional algorithms to accommodate label distribution learning. For example, LDLogitBoost [17] learns a general model family, which can be explained to a combination of the boosting method and the logistic regression. LCR-LDL [16] modifies kNN to build a local dictionary where each unlabeled sample are treated as a collaborative representation of the dictionary. LDLFs [18] builds forests based on differentiable decision trees and define a distribution-based loss function that enables all trees to learn jointly. Special Algorithm (SA) methods directly match the LDL problem, learning the relative importance of each label to a single sample. For instance, the maximum entropy model [19–22] and the linear model [23] are used as the output model, which can directly output the label distribution.

However, the above algorithms are all based on traditional feature representations, whose features are considered to be discriminative for all labels. This general strategy of using a feature set shared by all labels may not be optimal since different labels have their specific features. Therefore, in multi-label learning, the algorithm LIFT [11] first proposes to explore the specific features of each label. LIFT performs cluster analysis on the positive and negative samples, and constructs specific features of each label by querying the clustering results. Subsequently, various multi-label methods for learning label-specific features have been proposed. For example, LLSF [24] selects features corresponding to non-zero items in the weight parameter vector as label-specific features, and its weight vector is sparsed by l_1-regularization. MLFC [25] designs an optimization framework to learn a feature weight assignment scheme to select label-specific features.

Although, in multi-label learning, various methods have been proposed to make better label predictions by exploiting specific features. Only a few studies have focused on the specific features of LDL. To the best of our knowledge, similar to LLSF, LDLSF [23] considers l_1-regularization to sparse weight vectors to select label-specific features. Furthermore, HFSLDL [26] applies $l_{2,1}$-norm to make feature sparsity to learn specific features of each label. However, these methods constrain the parameter matrix by improving the loss function, which complicates the corresponding optimization strategy and may lead to overfitting. In addition, label characteristic information hidden in samples of the same label is ignored, which is important for learning specific features. Therefore, similar to LIFT, we employ clustering to analyze the latent features of each label. However, unlike multi-label data, the relevant and irrelevant of labels to samples is relative in LDL. A threshold is usually employed to determine the boundary between relevant and irrelevant. Consequently, in order to avoid redundant calculations and the influence of irrelevant samples, we only perform clustering analysis on samples that have a certain correlation with the label. At the same time, we consider the features shared between labels and label correlation to enhance the model.

3 The proposed algorithm

3.1 Preliminaries

Let $\mathcal{X} = \mathbb{R}^q$ denote the q-dimensional feature space, and $\mathcal{Y} = \{y_1, y_2, \cdots, y_l\}$ denote the label space. Given a set of n label distribution training samples $\mathcal{S}$, we construct the feature matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$, and the label distribution matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_n]$. $\mathbf{x}_i \in \mathcal{X}$ is a q-dimensional feature vector and $\mathbf{d}_i \in \mathcal{Y}$ is the corresponding l-dimensional label distribution vector. And its element $d_{ij} \in [0, 1]$ represents description degree of the j-th label to the i-th sample. It is generally considered that all labels in the label set can completely describe the sample, that is, $\sum_{j=1}^{l} d_{ij} = 1$. Then, the task of LDL is to learn the mapping $\phi(\mathbf{X}) = \mathbf{D}$ to predict the label distribution of an unlabeled sample.

As discussed in Sect. 1, our proposed algorithm LDL-LDF considers both label-dependent features and common features, and exploits label correlation by adding loss function constraints. Therefore, we sequentially introduce the label-dependent feature construction, common feature construction, output model and loss function construction of LDL-LDF. Table 1 summarizes the notations in the paper.

Table 1 Notations

Notation	Meaning
$\mathbf{X}$	The feature matrix
$\mathbf{D}$	The label distribution matrix
$\mathbf{P}$	The predicted label distribution matrix
$\mathbf{R}^k$	The relevant samples set of the k-th label
$\mathbf{x}_i$	The feature vector of i-th sample
$\mathbf{d}_i$	The label distribution of i-th sample
$\mathbf{p}_i$	The predicted label distribution of i-th sample
$\mathbf{h}_i$	The common feature vector of the i-th sample
$\mathbf{g}_i^k$	The label-dependent feature vector of the i-th sample corresponding to the k-th label

3.2 The construction of label-dependent features

To capture the intrinsic characteristics of each label, we perform cluster analysis [11] on the samples associated with each label. We first binarize [13] the label distribution of the training set to select samples related to each label. Specifically, for the sample $\mathbf{x}_i$, its label distribution is $\mathbf{d}_i = \{d_{i1}, \cdots, d_{il}\}$. Assuming that d_{ij} is the largest descriptive degree among them, the sample $\mathbf{x}_i$ is assigned to the relevant sample set of label y_j, i.e. $\mathbf{l}_{ij} = 1$, where $\mathbf{l}_i$ is the logical label vector for $\mathbf{x}_i$. We then compute the sum of the label description degrees corresponding to the labels associated with sample $\mathbf{x}_i$, i.e. $o = \sum_{\mathbf{l}_{ij}=1} d_{ij}$. Comparing o with a threshold τ, if $o < \tau$, we find the highest distribution value in the remaining label distribution, and then divide the sample $\mathbf{x}_i$ into related sample set of the corresponding label. The above operations are repeated until $o > \tau$. Finally, we divide the remaining labels into irrelevant labels, that is, let their corresponding logical labels be 0. In this way, we get the relevant sample set $\mathbf{R}^k = \{\mathbf{x}_i | (\mathbf{x}_i, \mathbf{d}_i) \in \mathcal{S}, \mathbf{l}_{ik} = 1\}$ for each label y_k. Notably, the binarization is only used to select samples related to each label and will not participate in subsequent work.

With the $\mathbf{R}^k$ associated with label y_k, we employ the density peak (DP) [27] clustering to investigate the underlying properties of the associated samples for each label. Unlike LIFT, we employ DP clustering instead of k-means. The clustering results of k-means are affected by the selection of initial values, while the clustering results of DP are relatively stable. The cluster center of DP needs to meet two conditions: (1) a higher local density; (2) a larger distance from other samples with high local density. For each element R_i^k in the set $\mathbf{R}^k$, its local density ρ_i^k is calculated using the Gaussian kernel:

$$\rho_i^k = \sum_{j \neq i} e^{-\left(\frac{dis(R_i^k, R_j^k)}{d^k}\right)}, \tag{1}$$

where $dis()$ is the Euclidean distance and d^k is the cutoff distance. We use the maximum distance of any pair of elements in the set $\mathbf{R}^k$ multiplied by μ as the cutoff distance: We use a multiple of the maximum distance between each element-pair in the set $\mathbf{R}^k$ as the cutoff distance:

$$d^k = \mu \cdot \max_{i,j} \{dis(R_i^k, R_j^k)\}, \tag{2}$$

where μ is a trade-off parameter that is set to 0.2 by experience. Moreover, we also need to calculate the minimum value δ_i^k of the distance between R_i^k and other samples with higher local density than it:

$$\delta_i^k = \begin{cases} \max\limits_{m,n} \{dis(R_m^k, R_n^k)\}, \\ \min\limits_{j} \{dis(R_i^k, R_j^k)\}, \end{cases} \tag{3}$$

When $\mathbf{R}_j^k$ has the largest local density, its δ_i^k takes the maximum distance between any pair of samples in the $\mathbf{R}^k$, otherwise δ_i^k takes its minimum distance between $\mathbf{R}_i^k$ with higher local density. The v_i^k that quantifies the degree to which $\mathbf{R}_i^k$ satisfies the two conditions mentioned above is expressed as:

$$v_i^k = \rho_i^k \cdot \delta_i^k. \tag{4}$$

Then, we get top c_k cluster centers with higher value of v in $\mathbf{R}^k$, which are denoted as $\{\mathbf{r}_1^k, \ldots, \mathbf{r}_{c_k}^k\}$. Moreover, the number c_k of cluster centers corresponding to the label y_k is determined by its related sample number $|\mathbf{R}_k|$ and ratio r_{dep}, i.e., $c_k = \lfloor r_{dep} \cdot |\mathbf{R}_k| \rfloor$. As the representative samples of the corresponding label, the cluster centers characterize the underlying sample structure with the same label. Therefore, the representative samples are used to construct specific features as appropriate building blocks.

Then, the original q-dimensional features are mapped into a c_k-dimensional label-dependent feature space through $\psi_k(\mathbf{x})$,

$$\psi_k(\mathbf{x}) = [dis(\mathbf{x}, \mathbf{r}_1^k), dis(\mathbf{x}, \mathbf{r}_2^k), \cdots, dis(\mathbf{x}, \mathbf{r}_{c_k}^k)], \tag{5}$$

where $dis()$ is the Euclidean distance.

3.3 The construction of common features

The common features are built in a similar way to the label-dependent features. Since these features are shared by all labels, we perform cluster analysis on the global samples. DP clustering are used to efficiently analyze properties common to all labels. Similarly, $t = \lfloor r_{com} \cdot n \rfloor$ cluster centers are obtained, denoted as $\{\eta_1, \ldots, \eta_t\}$. We map the original features to the t-dimensional common features space through a mapping $\psi_{com}(\mathbf{x})$,

$$\psi_{com}(\mathbf{x}) = [dis(\mathbf{x}, \eta_1), dis(\mathbf{x}, \eta_2), \cdots, (\mathbf{x}, \eta_t)]. \tag{6}$$

3.4 Prediction model

We design l parallel neural networks $\{f_1(\cdot), \ldots, f_l(\cdot)\}$ that utilize label-dependent features to predict the label

distribution value for each label separately. For the label y_k, a new feature matrix $\mathbf{G}^k$ is constructed, whose i-row $\mathbf{g}_i^k$ is obtained by mapping the original features vector $\mathbf{x}_i$ through ψ_k,

$$\mathbf{g}_i^k = \psi_k(\mathbf{x}_i). \tag{7}$$

Input $\mathbf{g}_i^k$ into the network $f_k(\cdot)$ corresponding to the label y_k, and get the predicted label distribution value p_{ik}^{dep} as follows:

$$p_{ik}^{dep} = \sigma(\mathbf{W}^k \mathbf{g}_i^k), \tag{8}$$

where σ is the *relu* activation function, $\mathbf{W}^k \in \mathbb{R}^{(c_k \times 1)}$ is the parameter matrix of the neural network $f_k(\cdot)$. We concatenate the individual predicted label distribution values of l parallel networks into a vector and normalize to get $\mathbf{p}_i^{dep} = softmax([p_{i1}^{dep}, p_{i2}^{dep}, \cdots, p_{il}^{dep}])$. It represents the predicted label distribution of $\mathbf{x}_i$ learned using label-dependent features.

For common features, we design a neural network $s(\cdot)$ to directly output the label distribution. With the mapping $\psi_{com}(\mathbf{x})$, we obtain the common feature matrix $\mathbf{H}$, and the i-th row $\mathbf{h}_i$ corresponds to $\mathbf{x}_i$ is calculated as follows:

$$\mathbf{h}_i = \psi_{com}(\mathbf{x}_i). \tag{9}$$

Input $\mathbf{h}_i$ into $s(\cdot)$, and its corresponding label distribution $\mathbf{p}_i^{com}$ is obtained as follows:

$$\mathbf{p}_i^{com} = \sigma(\mathbf{Z}\mathbf{h}_i), \tag{10}$$

where $\mathbf{Z} \in \mathbb{R}^{(t \times l)}$ is the parameter matrix of the neural network $s(\cdot)$.

To improve the performance of the model, we utilize a attention mechanism $Q(\cdot)$ to fuse these two label distributions obtained using different features. The attention mechanism can automatically learn the importance of two label distributions. Specifically, $Q(\cdot)$ is a single-layer feed-forward neural network parameterized by a weight vector $\mathbf{a}$ ($\mathbf{a} \in \mathbb{R}^{(l \times 1)}$):

$$\begin{aligned} \alpha_i^{dep} &= \sigma(\mathbf{a}\mathbf{p}_i^{dep}), \\ \alpha_i^{com} &= \sigma(\mathbf{a}\mathbf{p}_i^{com}). \end{aligned} \tag{11}$$

In general, normalization makes it easy to compare the difference between α_i^{dep} and α_i^{com}:

$$\begin{aligned} \alpha_i^{dep} &= \frac{\alpha_i^{dep}}{\alpha_i^{dep} + \alpha_i^{com}}, \\ \alpha_i^{com} &= \frac{\alpha_i^{com}}{\alpha_i^{dep} + \alpha_i^{com}}. \end{aligned} \tag{12}$$

Furthermore, the final prediction vector $\mathbf{p}_i$ of the sample $\mathbf{x}_i$ is obtained as:

$$\mathbf{p}_i = \alpha_i^{dep} \cdot \mathbf{p}_i^{dep} + \alpha_i^{com} \cdot \mathbf{p}_i^{com}. \tag{13}$$

And the predicted label distribution matrix $\mathbf{P}$ is constructed as $[\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n]$. The overall structure of the network is shown in Fig. 2.

We design a loss function using MSE to guide parameter optimization as follows:

$$Loss = \sum_{i=1}^{n} \|\mathbf{p}_i - \mathbf{d}_i\|^2. \tag{14}$$

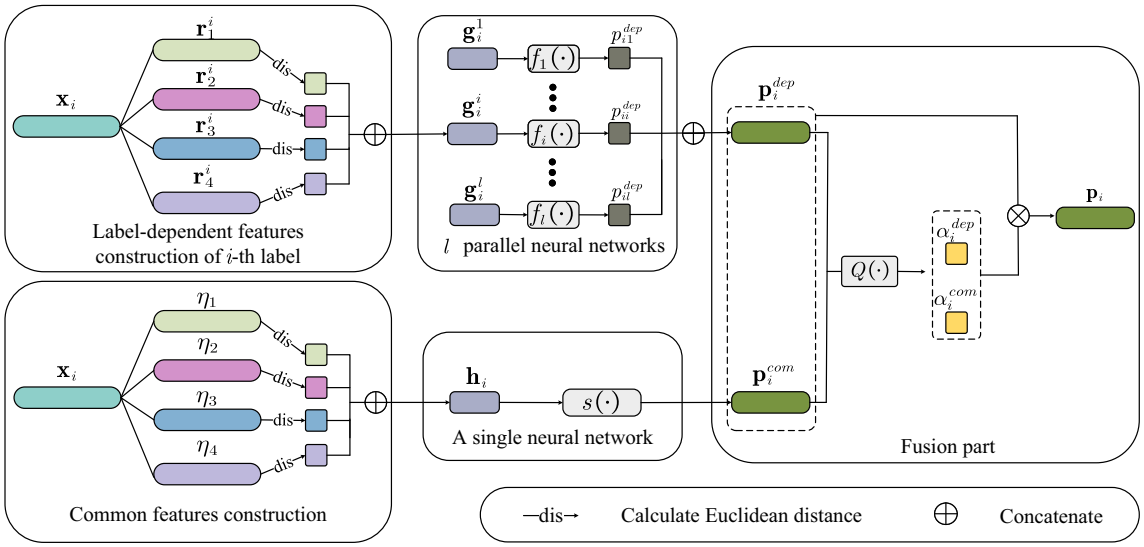


Fig. 2 The overall framework of LDL-LDF. First, label-dependent features and common features are constructed. Second, l parallel networks serving label-dependent features and a single network serving common features are designed. Finally the prediction results are fused through the attention network

Furthermore, we exploit label correlation to improve the performance. Inspired by [23], we constrain the output of the label distribution to exploit label correlations. Therefore, the Eq. 14 is rewritten as:

$$Loss = \sum_{i=1}^{n} \|\mathbf{p}_i - \mathbf{d}_i\|^2 + \frac{\lambda}{2} \sum_{m,o}^{l} \mathbf{Q}_{mo} \|\mathbf{P}_{.m} - \mathbf{P}_{.o}\|^2, \quad (15)$$

where λ is the balance factor, and $\mathbf{Q}_{mo}$ is the correlation coefficient between the m-th and o-th labels calculated by Pearson's correlation coefficient [5]. $\mathbf{P}_{.m}$ and $\mathbf{P}_{.o}$ represents the m-th and o-th columns of the predicted label distribution matrix $\mathbf{P}$. The pseudo codes of LDL-LDF are given in the algorithm 1, which consists of two phases, training and testing.

4 Experiments

4.1 Datasets

Our experimental studies are undertaken on 14 real-world datasets derived from yeast gene experiments, natural scene recognition, emotion recognition, human genetic and disease research and movie rating. The first nine datasets [28] contain 2,456 yeast genes, each characterized by a phylogenetic profile of length 24. The labels are discrete time points in different biological experiments, ranging in number from 3 to 18. Nature_Scene is collected from 2000 natural scene images, each image has 294-dimensional features and nine different labels (i.e., plant, sky, cloud,

Algorithm 1 Label-dependent feature exploration for label distribution learning

Input: the feature matrix $\mathbf{X}$, the label distribution matrix $\mathbf{D}$, parameters r_{dep}, r_{com}, λ, unseen sample $\mathbf{x}_m$;
Output: label distribution $\mathbf{d}_m$ of $\mathbf{x}_m$;

Train:
Initialize the l parallel prediction networks (serving for each label) and an individual prediction network (serving for all labels);
for $k = 1$ to l **do**
    Select the sample set $\mathbf{R}^k$ for label y_k;
    Perform DP clustering on $\mathbf{R}^k$ to obtain $c_k = \lfloor r_{dep} \cdot |\mathbf{R}_k| \rfloor$ clusters;
    Create the mapping $\psi_k(\mathbf{x})$ with Eq. 5;
    Construct $\mathbf{G}^k$ using $\psi_k(\mathbf{x})$ with Eq. 7;
end for
Perform DP clustering on $\mathbf{X}$ to obtain $t = \lfloor r_{com} \cdot n \rfloor$ clusters;
Create the mapping $\psi_{com}(\mathbf{x})$ with Eq. 6;
Construct $\mathbf{H}$ with Eq. (9);
Train the l parallel prediction networks and the individual prediction network using $\mathbf{G}, \mathbf{H}, \mathbf{D}$;
Test:
for $k = 1$ to l **do**
    Compute $\mathbf{g}_m$ with Eq. (7)
end for
Compute $\mathbf{h}_m$ with Eq. (9);
Return the label distribution $\mathbf{d}_m$ with Eq. (13);

Table 2 Statistics of the fourteen datasets

Index	Dataset	Sample	Feature	Label
1	Yeast-alpha	2,465	24	18
2	Yeast-cdc	2,465	24	15
3	Yeast-elu	2,465	24	14
4	Yeast-cold	2,465	24	4
5	Yeast-dtt	2,465	24	4
6	Yeast-diau	2,465	24	7
7	Yeast-heat	2,465	24	6
8	Yeast-spo	2,465	24	6
9	Yeast-spo5	2,465	24	3
10	Natrue_Scene	2,000	294	9
11	Emotion6	1,980	168	7
12	Flickr_LDL	11,150	168	8
13	Human_Gene	30,542	36	68
14	Movie	7,755	1,869	5

Table 3 Metrics of LDL algorithms

	Name	Formula
Distance	KL↓	$Dis_1(\mathbf{p}_i, \mathbf{d}_i) = \sum_{j=1}^{l} \mathbf{p}_{ij} \ln \frac{\mathbf{p}_{ij}}{\mathbf{d}_{ij}}$
	Euclidean↓	$Dis_2(\mathbf{p}_i, \mathbf{d}_i) = \sqrt{\sum_{j=1}^{l} (\mathbf{p}_{ij} - \mathbf{d}_{ij})}$
	Sϕrensen↓	$Dis_3(\mathbf{p}_i, \mathbf{d}_i) = \frac{\sum_{j=1}^{l} \lvert \mathbf{p}_{ij} - \mathbf{d}_{ij} \rvert}{\sum_{j=1}^{l} \lvert \mathbf{p}_{ij} + \mathbf{d}_{ij} \rvert}$
Similarity	Intersection↑	$Sim_1(\mathbf{p}_i, \mathbf{d}_i) = \sum_{j=1}^{l} min(\mathbf{p}_{ij}, \mathbf{d}_{ij})$
	Fidelity↑	$Sim_2(\mathbf{p}_i, \mathbf{d}_i) = \sum_{j=1}^{l} \sqrt{\mathbf{p}_{ij} \mathbf{d}_{ij}}$

snow, building, desert, mountain, water and sun). Emotion6 [29] has a total of 1,980 images, each containing 168-dimensional features and seven basic emotion labels (i.e., anger, disgust, joy, fear, sadness, surprise, and neutral). Flickr_LDL [23] contains 11,150 images with 168-dimensional features. Each image is typically labeled with eight emotions (i.e., anger, amusement, awe, contentment, disgust, excitement, fear and sadness). Human_Gene [30] contains 30,542 samples whose 36-dimensional features represent different gene sequences and 68 labels represent different diseases. Movie contains 7,755 movies from Netflix, and the label distribution for each movie consists of a percentage of the rating class with a 5-level scale. Each movie has 1869-dimensional features extracted from metadata about director, actor, country etc. Table 2 tabulates characteristics of these experimental data sets.

4.2 Evaluation metrics

To comprehensively evaluate the performance of the comparison algorithms, we select metrics based on two different perspectives, distance and similarity. Five evaluation metrics commonly found in the LDL community [31–33] are used in our experiments, three of which are distance-based: Kullback–Leibler (KL)↓, Euclidean↓ and Sϕrensen↓, two of which are similarity-based: Intersection↑, Fidelity↑. ↓ means smaller values are better and vice versa. Table 3 lists the formulas for each metrics, where $\mathbf{p}_i$ and $\mathbf{d}_i$ are the predicted label distribution and the ground-truth one.

4.3 Parameter analysis

LDL-LDF has four parameters, the threshold τ, which determines the number of selected samples associated with each label, the ratio r_{dep}, which determines the number of clusters in the sample set for each label, the ratio r_{com}, which determines the number of clusters in training samples, and the balance factor λ, which is empirically set to 0.001. To learn the effects of the first three parameters, we conduct extensive experiments running LDL-LDF with τ enumerated from $\{0.1, 0.2, \cdots, 0.9\}$, r_{dep} and r_{com} enumerated from $\{0.1, 0.3, \cdots, 0.9\}$. Figure 3 shows the experimental results of different τ on the dataset Yeast-alpha. When τ increases from 0.1, the performance effect gradually improves. The effect is best when τ is set to 0.3, and then gradually stabilizes.

Figure 4 shows the experimental results of r_{dep} and r_{com} on the dataset Yeast-alpha. As can be seen from the figure 4, when r_{dep} and r_{com} increase, the performance of LDL-LDF gradually deteriorates. This indicates that when r_{dep} and r_{com} is too large, some underrepresented samples will be selected as representative samples, thus affecting the mining of label characteristics. For the dataset Yeast_alpha, LDL-LDF performs better when r_{dep} and r_{com} take 0.1.

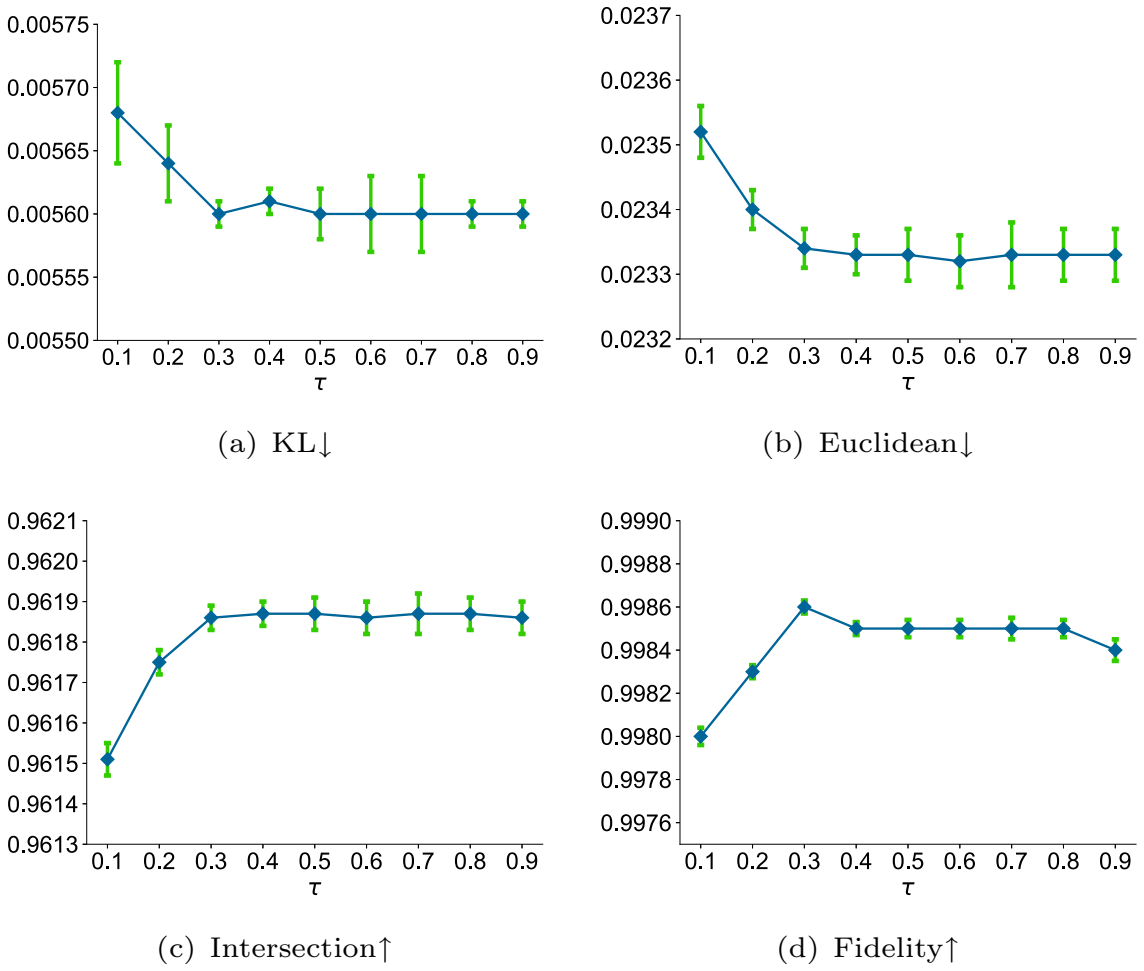


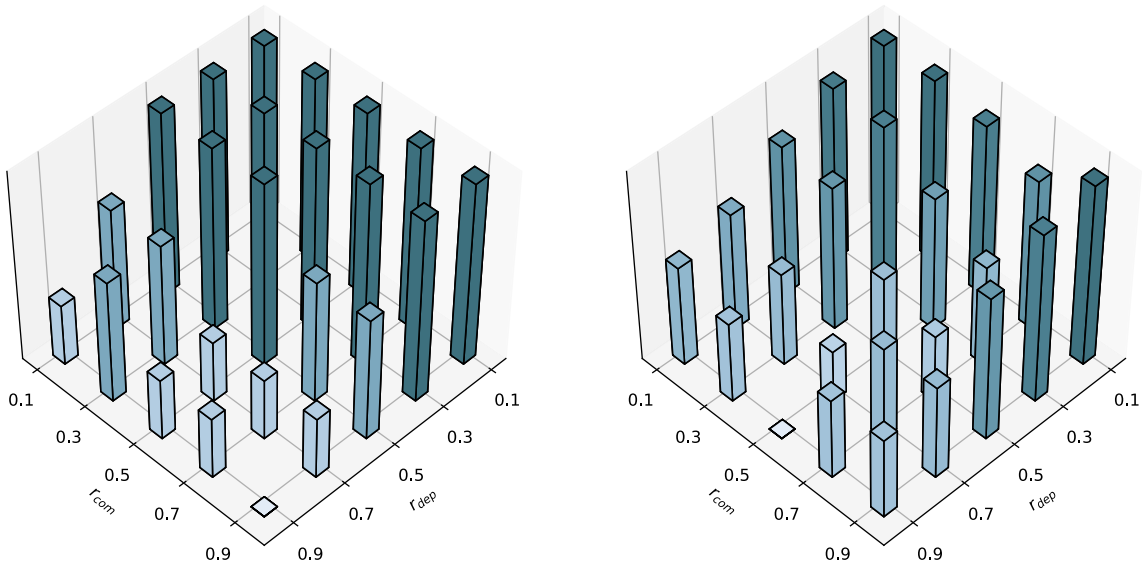
Fig. 3 Sensitivity of τ on Yeast-alpha with KL↓, Euclidean↓, Intersection↑ and Fidelity↑

4.4 Comparing algorithms

We compare LDL-LDF with three baseline algorithms, AA-*k*NN [3], IIS-LLD [3], EDL [5], four state-of-the-art algorithms LDLSF [23], Adam-LDL-SCL [33], LDL-LDM [34], LDL-HR [35]. For AA-*k*NN, its neighborhood size k is enumerated from $\{5, 10, \cdots, 20\}$. For IIS-LLD and EDL, we use the optimal parameters from their paper. Moreover, for LDLSF, ρ is set to 10^{-3}, λ_1 is set to 10^{-4}, λ_2 is set to 10^{-2} and λ_3 is set to 10^{-3}. For Adam-LDL-SCL, λ_1, λ_2 and λ_3 are set to 10^{-2}, and the number of clusters m is simply set to 6. For LDL-LDM, λ_1 is set to 10^{-2}, λ_2 and λ_3 are enumerated from $\{10^{-3}, \cdots, 10^{3}\}$, and g is set to 10. For LDL-HR, the margin ρ between the highest label description degree and those of other labels is set to 10^{-2}, λ_1 is set to 10^{-4}, λ_2 and λ_3 is set to 1. For LDL-LDF, the threshold τ is set to 0.3, which determines the samples to be selected related to each label, the ratio r_{dep} is set to 0.1, which determines the number of clusters in the sample set for each label, the ratio r_{com} is set to 0.1, which determines the number of clusters in the global samples, and the balance factor λ is set to 0.001. Table 4 lists the parameter settings for these comparing algorithms.

4.5 Ablation study

We analyze the effectiveness of label-dependent and common features through ablation studies. Table 5 shows the experimental results with KL↓. Specifically, the results of LDL_{ori} are obtained by inputting original features into a



(a) Intersection↑ (b) Fidelity↑

Fig. 4 Sensitivity of r_{dep} and r_{com} on Yeast-alpha with Intersection↑ and Fidelity↑

Table 4 Parameter settings of LDL algorithms

Algorithms	Parameter settings
AA-kNN	$k \in \{5, 10, \cdots, 20\}$
IIS-LLD	-
EDL	-
LDLSF	$\rho = 10^{-3}, \lambda_1 = 10^{-4}, \lambda_2 = 10^{-2}$ and $\lambda_3 = 10^{-3}$
Adam-LDL-SCL	$\lambda_1 = \lambda_2 = \lambda_3 = 10^{-2}, m = 6$
LDL-LDM	$\lambda_1 = 10^{-2}, \lambda_2$ and $\lambda_3 \in \{10^{-3}, \cdots, 10^{3}\}, g = 10$
LDL-HR	$\rho = 10^{-2}, \lambda_1 = 10^{-4}, \lambda_2 = \lambda_3 = 1$
LDL-LDF	$\tau = 0.3, r_{dep} = r_{com} = 0.1, \lambda = 0.001$

Table 6 Friedman statistics F_F on five metrics and the critical value when significance level $\alpha = 0.05$ (Algorithms k = 8, Datasets N = 14)

Metrics	F_F	Critical value
KL↓	23.6197	2.1119
Euclidean↓	16.9506	
Sϕrensen↓	34.3547	
Intersection↑	14.0029	
Fidelity↑	7.5417	

multi-layer neural network that is similar to the one serving for common features. The corresponding results of LDL_{dep} are obtained by using label-dependent features alone, with the parameters $\tau = 0.3$ and $r_{dep} = 0.1$. The corresponding results of LDL_{com} are obtained using the common features with parameter $r_{com} = 0.1$. The results of LDL-LDF are obtained by considering label-dependent and common

Table 5 Results (mean±std) of ablation experiments on KL↓. The best results are in bolded

Dataset	LDL_{ori}	LDL_{dep}	LDL_{com}	LDL-LDF
Yeast-alpha	0.0069 ± 0.0000	0.0057 ± 0.0000	0.0057 ± 0.0001	$\boldsymbol{0.0056 \pm 0.0000}$
Yeast-cold	0.0133 ± 0.0010	0.0132 ± 0.0010	0.0132 ± 0.0008	$\boldsymbol{0.0122 \pm 0.0001}$
Yeast-cdc	0.0083 ± 0.0003	0.0072 ± 0.0003	0.0073 ± 0.0003	$\boldsymbol{0.0070 \pm 0.0000}$
Yeast-elu	0.0076 ± 0.0002	0.0064 ± 0.0002	0.0064 ± 0.0000	$\boldsymbol{0.0062 \pm 0.0000}$
Yeast-dtt	0.0070 ± 0.0005	0.0065 ± 0.0005	0.0066 ± 0.0000	$\boldsymbol{0.0061 \pm 0.0000}$
Yeast-diau	0.0149 ± 0.0006	0.0140 ± 0.0006	0.0143 ± 0.0006	$\boldsymbol{0.0128 \pm 0.0000}$
Yeast-heat	0.0140 ± 0.0003	0.0133 ± 0.0003	0.0133 ± 0.0003	$\boldsymbol{.0126 \pm 0.0001}$
Yeast-spo	0.0269 ± 0.0015	0.0257 ± 0.0015	0.0269 ± 0.0015	$\boldsymbol{.0244 \pm 0.0001}$

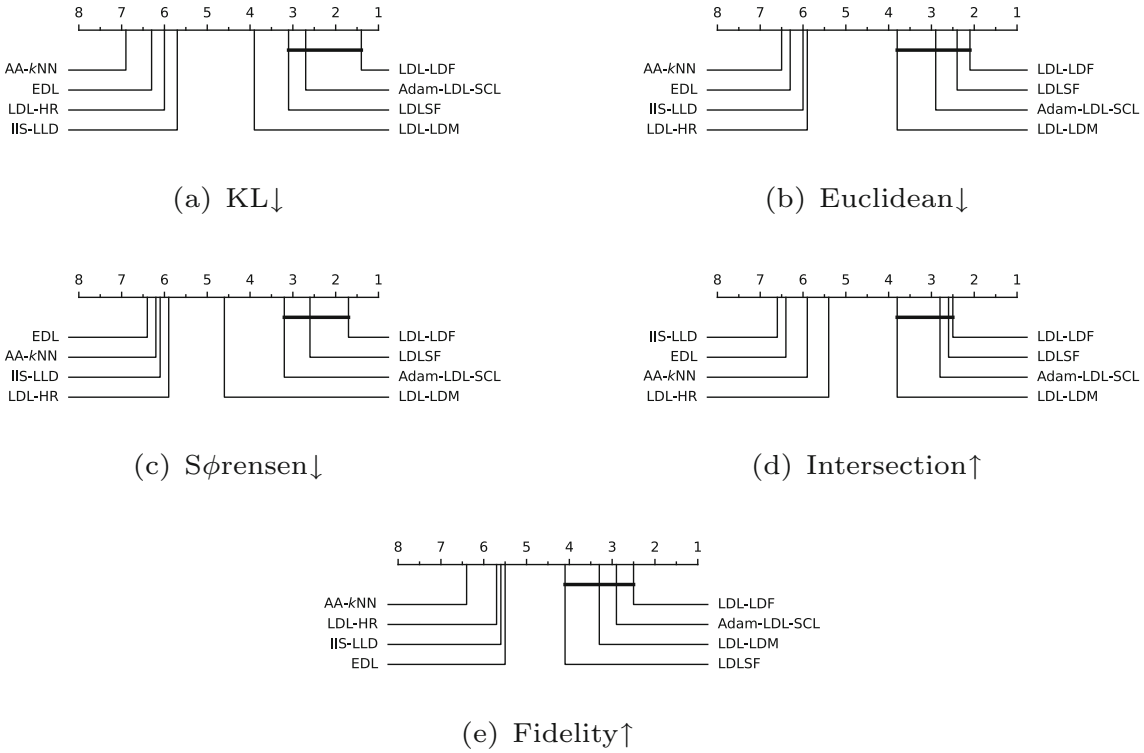


Fig. 5 CD diagrams on each metrics (CD = 2.49 at 0.05 significance level)

features with parameters $\tau = 0.3$, $r_{dep} = 0.1$ and $r_{com} = 0.1$.

It can be seen from the table 5 that the performance of LDL_{dep} and LDL_{com} is better than that of LDL_{ori} on most datasets. This illustrates the effectiveness of label-dependent features and common features. In addition, in LDL-LDF, these two features are considered simultaneously, and the effect is more significantly improved. Such results are reasonable, since considering both features together can build a more comprehensive model of the relationship between features and labels.

4.6 Results

We compare LDL-LDF with seven algorithms on fourteen datasets and employ 10-fold cross-validation (10CV) to obtain stable results. Tables 7, 8, 9, 10, 11 show experimental results in different metrics, where the best results in each table are marked in bold, and Avg.Rank is summarized in the last row. It can be seen that LDL-LDF is significantly better than other comparison algorithms, and ranks first on average in five metrics. Furthermore, LDL-LDF ranks first in 71% of the 70 (metrics 5 × datasets 14) results. This is because LDL-LDF considers both common features and label-dependent features, which makes it more reasonable than other algorithms that only utilize the shared feature strategy.

Moreover, we conducted *Friedman* test [36] to further analyze the performance of the comparing algorithms. This test method is widely used to verify whether multiple algorithms are statistically different when compared on multiple data sets. In the comparison of k algorithms and N datasets, the ranking of j-th algorithms on i-th datasets is denoted as r_i^j. Then the average ranking R_j of the j-th algorithm is calculated as $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j$. With $k-1$ numerator degrees of freedom and $(k-1)(N-1)$ denominator degrees of freedom, the Friedmann statistic F_F following the F distribution is calculated as $F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}$, where $\chi_F^2 = \frac{12N}{k(k+1)}\sum_{j=1}^{k}\left[R_j^2 - \frac{k(k+1)^2}{4}\right]$. Table 6 lists the Friedman statistic F_F and critical value on the five metrics. It can be seen that at the significance level $\alpha = 0.05$, F_F is much larger than the critical value on each metrics. Therefore, the null hypothesis of performance equality between the compared algorithms is clearly rejected.

we further performed the *Bonferroni-Dunn* test [37] to analyze the difference in competitiveness among compared

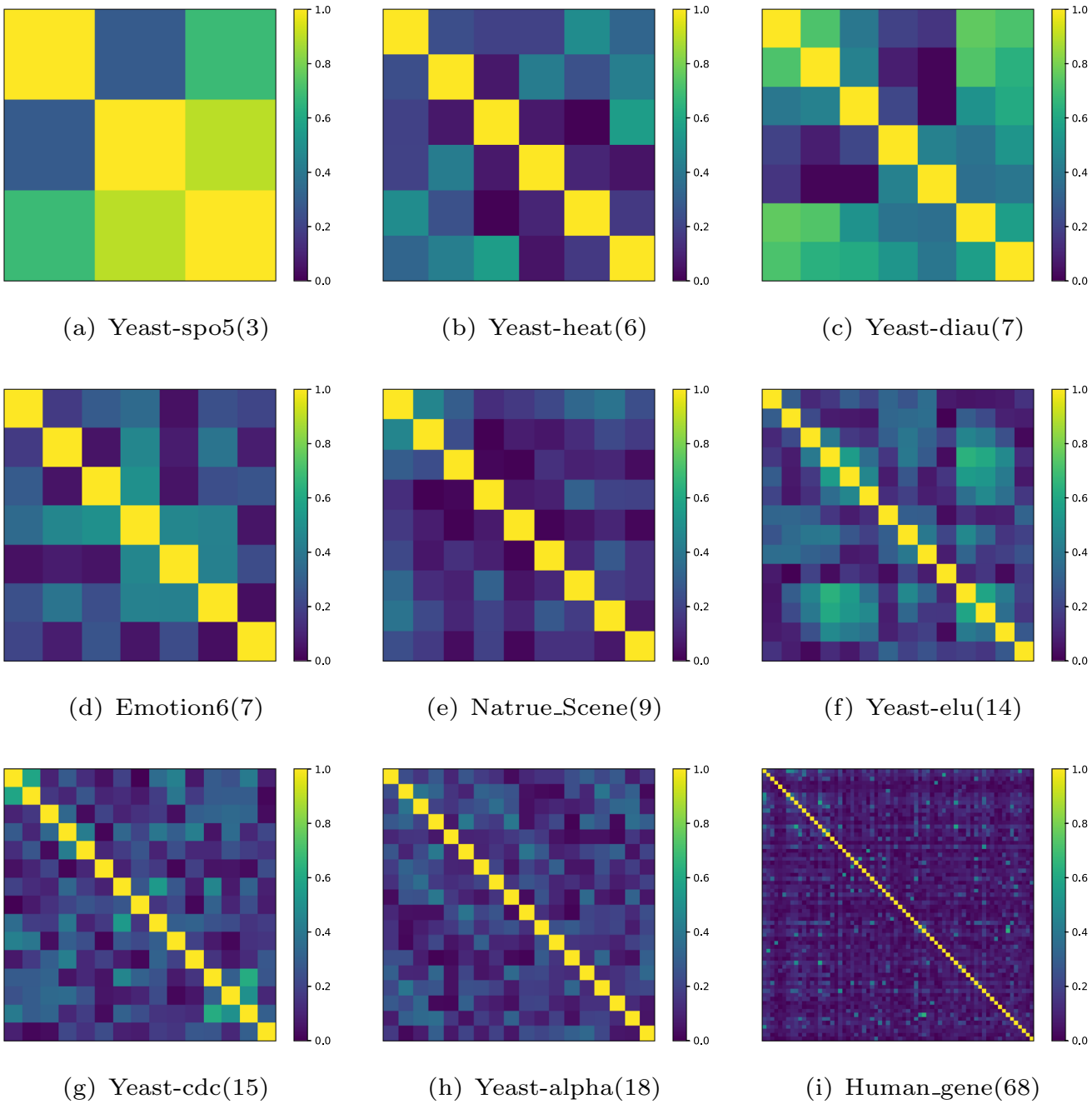


Fig. 6 Illustration of label correlations between different datasets. Blue indicates negative correlation and yellow indicates positive correlation. The number of labels for each dataset is indicated in parentheses after the dataset, e.g. Yeast-spo5(3)

algorithms. In order to visually compare the performance of the algorithms, Fig. 5 plots CD diagrams on different metrics with LDL-LDF as the control algorithm. If the comparison algorithms are not connected to the LDL-LDF by a line, then their mean rank differs from the LDL-LDF by more than one critical difference (CD) value. In other words, there are significant differences between these algorithms and control algorithms. As shown in Fig. 5, LDL-LDF significantly outperforms AA-kNN, IIS-LLD, EDL, and LDL-HR on all metrics. In terms of KL$\downarrow$ and Sϕrensen, LDL-LDF is significantly better than LDL-LDM. Furthermore, LDL-LDF achieves superior performance than LDLSF and Adam-LDL-SCL on all metrics.

Table 7 Comparison results of different LDL algorithms on eleven datasets under KL↓ (mean± std). ↓ indicates that the smaller the value, the better the effect. The best result is marked in bold, and Avg.Rank is summarized in the last row

Dataset	KL↓							
	AA-*k*NN	IIS-LLD	EDL	LDLSF	Adam-LDL-SCL	LDL-LDM	LDL-HR	LDL-LDF
Yeast-alpha	0.0070 ± 0.0004	0.0069 ± 0.0004	0.0068 ± 0.0006	0.0057 ± 0.0001	0.0056 ± 0.0005	0.0057 ± 0.0001	0.0076 ± 0.0004	**0.0055 ± 0.0001**
Yeast-cdc	0.0075 ± 0.0004	0.0072 ± 0.0005	0.0072 ± 0.0004	0.0071 ± 0.0001	0.0071 ± 0.0004	0.0072 ± 0.0001	0.0082 ± 0.0002	**0.0070 ± 0.0001**
Yeast-elu	0.0071 ± 0.0006	0.0071 ± 0.0004	0.0067 ± 0.0003	.0062 ± 0.0000	.0062 ± 0.0003	.0064 ± 0.0001	.0072 ± 0.0001	**0.0062 ± 0.0001**
Yeast-cold	0.0136 ± 0.0011	0.0155 ± 0.0015	0.0153 ± 0.0009	0.0122 ± 0.0008	0.0125 ± 0.0006	0.0127 ± 0.0002	0.0126 ± 0.0001	**0.0122 ± 0.0001**
Yeast-dtt	0.0070 ± 0.0007	0.0068 ± 0.0005	0.0068 ± 0.0008	0.0063 ± 0.0001	0.0065 ± 0.0005	0.0065 ± 0.0001	0.0064 ± 0.0001	**0.0061 ± 0.0001**
Yeast-diau	0.0145 ± .0010	0.0177 ± 0.0013	0.0155 ± 0.0005	0.0134 ± 0.0001	0.0131 ± 0.0005	0.0135 ± 0.0003	0.0148 ± 0.0001	**0.0128 ± 0.0001**
Yeast-heat	0.0141 ± 0.0010	0.0182 ± 0.0016	0.0143 ± 0.0008	0.0127 ± 0.0001	0.0126 ± 0.0009	0.0129 ± 0.0001	0.0132 ± 0.0001	**0.0126 ± 0.0001**
Yeast-spo	.0286 ± 0.0002	.0289 ± 0.0022	.0269 ± 0.0016	.0247 ± 0.0001	.0246 ± 0.0010	.0249 ± 0.0003	.0257 ± 0.0001	**0.0244 ± 0.0001**
Yeast-spo5	0.0334 ± 0.0005	0.0321 ± 0.0000	0.0321 ± 0.0000	0.0290 ± 0.0000	0.0303 ± 0.0001	0.0291 ± 0.0005	0.0296 ± 0.0001	**0.0286 ± 0.0001**
Nature_Scene	1.2103 ± 0.0161	0.9170 ± 0.0431	1.1649 ± 0.0242	1.0096 ± 0.0102	0.8945 ± 0.0342	0.8430 ± 0.0117	1.0303 ± 0.0086	**.7796 ± 0.0009**
Emotion6	0.8776 ± 0.0134	0.6149 ± 0.0004	0.7083 ± 0.0082	0.5998 ± 0.0082	0.6136 ± 0.0025	0.6017 ± 0.0065	0.6955 ± 0.0072	**0.5955 ± 0.0006**
Flickr_LDL	1.1459 ± 0.0000	.7120 ± 0.0000	1.0609 ± 0.0000	.6950 ± 0.0000	**0.6508 ± 0.0001**	0.7132 ± 0.0032	1.0367 ± 0.0171	.6902 ± 0.0015
Human_Gene	.2894 ± 0.0176	.2414 ± 0.0310	.2466 ± 0.0052	.2363 ± 0.0001	**.2347 ± 0.0042**	.2459 ± 0.0026	.2647 ± 0.0002	.2384 ± 0.0001
Movie	0.1259 ± 0.0173	0.1399 ± 0.0102	0.1287 ± 0.0014	0.1997 ± 0.0000	**0.1034 ± 0.0130**	.1036 ± 0.0022	.1641 ± 0.0151	.1052 ± 0.0009
Avg.Rank	6.9	5.7	6.3	3.1	2.7	3.9	6.0	1.4

Table 8 Comparison results of different LDL algorithms on eleven datasets under Euclidean↓ (mean±std). ↓ indicates that the smaller the value, the better the effect. The best result is marked in bold, and Avg.Rank is summarized in the last row

Dataset	Euclidean↓							
	AA-*k*NN	IIS-LLD	EDL	LDLSF	Adam-LDL-SCL	LDL-LDM	LDL-HR	LDL-LDF
Yeast-alpha	0.0278 ± 9.0006	0.0269 ± .0004	0.0260 ± 0.0011	0.0234 ± 0.0001	0.0234 ± 0.0003	0.0235 ± 0.0001	0.0277 ± .0003	**0.0230 ± 0.0001**
Yeast-cdc	0.0301 ± 0.0009	0.0290 ± 0.0010	0.0283 ± 0.0006	0.0280 ± 0.0003	0.0281 ± 0.0008	0.0284 ± 0.0002	0.0320 ± 0.0002	**0.0280 ± 0.0001**
Yeast-elu	0.0297 ± 0.0010	0.0307 ± 0.0009	0.0289 ± 0.0005	0.0279 ± 0.0005	.0279 ± 0.0005	.0282 ± 0.0001	.0285 ± 0.0001	**0.0276 ± 0.0001**
Yeast-cold	.0724 ± .0027	.0767 ± .0004	.0771 ± .0018	.0683 ± .0001	.0696 ± .0003	.0693 ± .0005	.0694 ± .0001	**.0680 ± .0006**
Yeast-dtt	0.0512 ± 0.0019	0.0535 ± 0.0023	0.0508 ± 0.0022	0.0480 ± 0.0001	0.0493 ± 0.0017	0.0489 ± 0.0002	0.0486 ± 0.0001	**0.0477 ± 0.0001**
Yeast-diau	0.0567 ± 0.0019	0.0652 ± 0.0031	0.0597 ± 0.0010	0.0543 ± 0.0001	.0551 ± 0.0008	.0551 ± 0.0004	0.0575 ± 0.0001	**0.0537 ± 0.0003**
Yeast-heat	0.0624 ± 0.0020	0.0703 ± 0.0036	0.0629 ± 0.0016	0.0593 ± 0.0001	0.0596 ± 0.0016	0.0599 ± 0.0002	0.0609 ± 0.0001	**0.0591 ± 0.0003**
Yeast-spo	0.0879 ± 0.0030	0.0863 ± 0.0041	0.0843 ± 0.0029	0.0822 ± 0.0019	0.0820 ± 0.0006	0.0830 ± 0.0001	0.0830 ± 0.0001	**0.0815 ± 0.0003**
Yeast-spo5	0.1231 ± 0.0007	0.1233 ± 0.0000	0.1191 ± 0.0001	0.1173 ± 0.0003	0.1193 ± 0.0002	0.1165 ± 0.0011	0.1176 ± 0.0001	**0.1156 ± 0.0002**
Nature_Scene	0.4269 ± 0.0194	0.4808 ± 0.0062	0.5223 ± 0.0062	0.4325 ± 0.0020	0.4308 ± 0.0018	0.4471 ± 0.0049	0.5439 ± .00095	**0.4185 ± 0.0017**
Emotion6	0.4551 ± 0.0134	0.4122 ± 0.0004	0.4502 ± 0.0029	**0.4048 ± 0.0007**	0.4111 ± 0.0018	0.4349 ± 0.0030	0.4178 ± 0.0072	0.4117 ± 0.0001
Flickr_LDL	0.4445 ± 0.0000	0.4280 ± 0.0000	0.5494 ± 0.0000	**0.3932 ± 0.0000**	0.3938 ± 0.0001	0.4369 ± 0.0016	0.4584 ± 0.0024	0.4206 ± 0.0003
Human_Gene	.1036 ± 0.0044	0.0877 ± 0.0078	0.0887 ± 0.0021	0.0864 ± 0.0001	**0.0858 ± 0.0016**	0.0861 ± 0.0002	0.0943 ± 0.0001	0.0867 ± 0.0000
Movie	0.1814 ± 0.0063	0.2121 ± 0.0058	0.1765 ± 0.0115	0.1906 ± 0.0000	**0.1720 ± 0.0028**	0.1751 ± 0.0017	0.2254 ± 0.0033	0.1744 ± 0.0011
Avg.Rank	6.5	6.0	6.3	2.4	2.9	3.8	5.9	2.1

Table 9 Comparison results of different LDL algorithms on eleven datasets under Sϕrensen↓ (mean±std). ↓ indicates that the smaller the value, the better the effect. The best result is marked in bold, and Avg.Rank is summarized in the last row

Dataset	Sϕrensen↓							
	AA-*k*NN	IIS-LLD	EDL	LDLSF	Adam-LDL-SCL	LDL-LDM	LDL-HR	LDL-LDF
Yeast-alpha	0.0449 ± 0.0012	0.0429 ± 0.0012	0.0429 ± 0.0022	0.0377 ± 0.0000	.0380 ± 0.0005	.0385 ± 0.0002	0.0447 ± 0.0001	**0.0376 ± 0.0001**
Yeast-cdc	0.0462 ± 0.0013	0.0445 ± .0015	0.0429 ± 0.0008	0.0426 ± 0.0000	0.0428 ± 0.0012	0.0433 ± 0.0003	0.0477 ± 0.0001	**0.0425 ± 0.0001**
Yeast-elu	0.0443 ± 0.0014	0.0472 ± 0.0014	0.0431 ± 0.0008	0.0412 ± 0.0000	0.0415 ± 0.0005	0.0420 ± 0.0001	0.0452 ± 0.0001	**0.0411 ± 0.0002**
Yeast-cold	0.0630 ± 0.0024	0.0653 ± 0.0034	0.0668 ± 0.0016	0.0592 ± 0.0000	0.0604 ± 0.0013	0.0602 ± 0.0005	0.0598 ± 0.0001	**0.0590 ± 0.0001**
Yeast-dtt	0.0443 ± 0.0017	0.0480 ± 0.0023	0.0440 ± 0.0018	.0417 ± 0.0001	0.0427 ± 0.0015	0.0424 ± 0.0002	0.0418 ± 0.0001	**0.0414 ± 0.0001**
Yeast-diau	0.0622 ± 0.0022	0.0593 ± 0.0032	0.0653 ± 0.0010	0.0597 ± 0.0000	0.0606 ± 0.0011	0.0605 ± 0.0006	0.0608 ± 0.0001	**0.0590 ± 0.0001**
Yeast-heat	0.0632 ± 0.0018	0.0692 ± 0.0033	0.0633 ± 0.0017	0.0599 ± 0.0001	0.0602 ± 0.0014	0.0604 ± 0.0003	0.0605 ± 0.0001	**0.0597 ± 0.0007**
Yeast-spo	0.0899 ± 0.0024	0.0861 ± 0.0036	0.0872 ± 0.0029	0.0848 ± 0.0001	0.0847 ± 0.0018	0.0846 ± 0.0006	0.0851 ± 0.0001	**0.0840 ± 0.0001**
Yeast-spo5	0.0960 ± 0.0005	0.0962 ± 0.0001	0.0930 ± 0.0001	0.0916 ± 0.0002	0.0932 ± 0.0001	0.0909 ± 0.0008	0.0917 ± 0.0001	**0.0903 ± 0.0001**
Nature_Scene	0.4369 ± 0.0188	0.5400 ± 0.0082	0.6344 ± 0.0074	0.4592 ± 0.0013	**0.4561 ± 0.0077**	0.5054 ± 0.0049	0.6109 ± 0.0070	0.4847 ± 0.0017
Emotion6	0.4506 ± 0.0008	0.4213 ± 0.0003	0.4748 ± 0.0032	**0.4143 ± 0.0008**	0.4164 ± 0.0018	0.4271 ± 0.0021	0.7558 ± 0.0032	0.4261 ± 0.0004
Flickr_LDL	0.4468 ± 0.0000	0.4477 ± 0.0001	0.5910 ± 0.0000	0.4061 ± 0.0001	**0.4025 ± 0.0001**	0.4595 ± 0.0013	0.4426 ± 0.0015	0.4420 ± 0.0002
Human_Gene	0.2548 ± 0.0036	0.2187 ± 0.0054	.2191 ± 0.0018	0.2156 ± 0.0000	**0.2152 ± 0.0014**	0.2162 ± 0.0005	0.2445 ± 0.0001	0.2160 ± 0.0001
Movie	0.1781 ± 0.0055	0.2026 ± 0.0048	0.1766 ± 0.0099	0.1876 ± 0.0000	0.1763 ± 0.0019	0.1727 ± 0.0017	0.3029 ± 0.0028	**0.1716 ± 0.0011**
Avg.Rank	6.2	6.1	6.4	2.6	3.2	4.6	5.9	1.7

Table 10 Comparison results of different LDL algorithms on eleven datasets under Intersection↑ (mean±std). ↑ indicates that the larger the value, the better the effect. The best result is marked in bold, and Avg.Rank is summarized in the last row

Dataset	Intersection↑							
	AA-*k*NN	IIS-LLD	EDL	LDLSF	Adam-LDL-SCL	LDL-LDM	LDL-HR	LDL-LDF
Yeast-alpha	0.9561 ± 0.0012	0.9571 ± 0.0012	0.9570 ± 0.0022	0.9619 ± 0.0005	0.9622 ± 0.0005	0.9615 ± 0.0002	0.9552 ± 0.0004	**0.9623 ± 0.0003**
Yeast-cdc	0.9538 ± 0.0013	0.9556 ± 0.0015	0.9571 ± 0.0008	0.9574 ± 0.0005	0.9572 ± 0.0013	0.9567 ± 0.0003	0.9544 ± 0.0003	**0.9574 ± 0.0001**
Yeast-elu	0.9557 ± 0.0014	0.9528 ± 0.0015	0.9569 ± 0.0007	0.9588 ± 0.0001	0.9585 ± 0.0005	0.9580 ± 0.0001	0.9455 ± 0.0001	**0.9588 ± 0.0000**
Yeast-cold	0.9370 ± 0.0024	0.9347 ± 0.0034	0.9332 ± 0.0016	0.9408 ± 0.0001	0.9396 ± 0.0013	0.9398 ± 0.0005	0.9397 ± 0.0001	**0.9409 ± 0.0005**
Yeast-dtt	0.9557 ± 0.0017	0.9520 ± 0.0023	0.9560 ± 0.0018	0.9580 ± 0.0001	0.9573 ± 0.0015	0.9576 ± 0.0002	0.9581 ± 0.0001	**0.9586 ± 0.0006**
Yeast-diau	0.9378 ± 0.0022	0.9282 ± 0.0033	0.9347 ± .0010	0.9403 ± 0.0001	0.9394 ± 0.0011	0.9394 ± 0.0006	0.9365 ± 0.0001	**0.9409 ± 0.0004**
Yeast-heat	0.9368 ± 0.0018	0.9309 ± 0.0033	0.9366 ± 0.0017	0.9401 ± 0.0001	0.9402 ± 0.0014	0.9396 ± 0.0003	0.9384 ± 0.0001	**0.9402 ± 0.0003**
Yeast-spo	0.9096 ± 0.0034	0.9139 ± 0.0036	0.9128 ± 0.0028	0.9154 ± 0.0001	0.9154 ± 0.0019	0.9154 ± 0.0006	0.9144 ± 0.0001	**0.9159 ± 0.0001**
Yeast-spo5	0.9040 ± 0.0005	0.9038 ± 0.0000	0.9070 ± 0.0001	0.9084 ± 0.0002	0.9068 ± 0.0001	0.9089 ± 0.0008	0.9081 ± 0.0001	**0.9096 ± 0.0001**
Nature_Scene	**0.5630 ± 0.0188**	0.4600 ± 0.0082	0.3662 ± 0.0074	0.5424 ± 0.0020	0.5523 ± .0014	0.4946 ± 0.0049	0.5050 ± 0.0070	0.5152 ± 0.0017
Emotion6	.5494 ± .0008	.5730 ± .0004	.5252 ± .0032	**.5857 ± .0008**	.5836 ± .0018	.5427 ± .0030	.5642 ± .0072	.5737 ± .0008
Flickr_LDL	0.5532 ± 0.0000	0.5523 ± 0.0000	0.4090 ± 0.0000	0.5939 ± 0.0000	0.5975 ± 0.0001	0.5405 ± 0.0013	0.5530 ± 0.0015	0.5557 ± 0.0004
Human_Gene	0.7451 ± 0.0036	0.7813 ± 0.0054	0.7810 ± 0.0018	0.7844 ± 0.0005	**0.7854 ± 0.0013**	0.7848 ± 0.0005	0.7755 ± 0.0001	.7831 ± 0.0001
Movie	0.8219 ± 0.0055	0.7974 ± 0.0048	0.8228 ± 0.0094	0.8124 ± 0.0000	0.8230 ± 0.0040	0.8273 ± 0.0017	0.7906 ± 0.0028	**0.8284 ± 0.0011**
Avg.Rank	5.9	6.6	6.4	2.6	2.8	3.8	5.4	2.5

Table 11 Comparison results of different LDL algorithms on eleven datasets under Fidelity↑ (mean±std). ↑ indicates that the larger the value, the better the effect. The best result is marked in bold, and Avg.Rank is summarized in the last row

Dataset	Fidelity↑							
	AA-*k*NN	IIS-LLD	EDL	LDLSF	Adam-LDL-SCL	LDL-LDM	LDL-HR	LDL-LDF
Yeast-alpha	0.9980 ± 0.0001	0.9983 ± 0.0011	0.9985 ± 0.0011	0.9985 ± 0.0001	0.9985 ± 0.0001	0.9986 ± 0.0002	0.9981 ± 0.0004	**0.9986 ± 0.0001**
Yeast-cdc	0.9980 ± 0.0001	0.9982 ± 0.0012	0.9981 ± 0.0011	0.9982 ± 0.0005	0.9982 ± 0.0003	0.9982 ± 0.0003	0.9962 ± 0.0001	**0.9983 ± 0.0001**
Yeast-elu	0.9982 ± 0.0002	0.9982 ± 0.0035	0.9979 ± 0.0009	0.9983 ± 0.0001	0.9983 ± 0.0001	0.9984 ± 0.0004	0.9974 ± 0.0001	**0.9984 ± 0.0000**
Yeast-cold	0.9966 ± 0.0003	0.9960 ± 0.0039	0.9968 ± 0.0036	0.9967 ± 0.0023	0.9968 ± 0.0001	0.9968 ± 0.0005	0.9968 ± 0.0001	**0.9969 ± 0.0000**
Yeast-dtt	0.9982 ± 0.0002	0.9983 ± 0.0013	0.9982 ± 0.0010	0.9984 ± 0.0002	0.9983 ± 0.0002	0.9983 ± 0.0002	0.9984 ± 0.0001	**0.9984 ± 0.0000**
Yeast-diau	0.9963 ± 0.0003	0.9964 ± 0.0036	0.9960 ± 0.0031	0.9960 ± 0.0002	0.9962 ± 0.0002	0.9966 ± 0.0001	0.9961 ± 0.0001	**0.9967 ± .0000**
Yeast-heat	0.9964 ± 0.0003	0.9954 ± 0.0042	0.9961 ± 0.0048	0.9966 ± 0.0003	0.9967 ± 0.0003	0.9967 ± 0.0003	0.9961 ± 0.0001	**0.9968 ± 0.0001**
Yeast-spo	0.9927 ± 0.0005	0.9937 ± 0.0005	0.9932 ± 0.0007	0.9937 ± 0.0003	0.9937 ± 0.0002	0.9936 ± 0.0006	0.9935 ± 0.0001	**0.9938 ± 0.0002**
Yeast-spo5	0.9915 ± 0.0001	0.9917 ± 0.0000	0.9922 ± 0.0000	0.9924 ± 0.0000	0.9923 ± 0.0001	0.9925 ± 0.0008	0.9923 ± 0.0001	**0.9926 ± 0.0002**
Nature_Scene	0.7301 ± 0.0194	0.6766 ± 0.0080	0.6087 ± 0.0062	0.7314 ± 0.0018	**0.7503 ± 0.0007**	0.7057 ± 0.0049	0.6815 ± 0.0090	0.7160 ± 0.0012
Emotion6	0.7726 ± .0008	0.7965 ± .0002	0.7593 ± .0022	**0.8006 ± .0007**	0.8074 ± 0.0013	0.7729 ± 0.0030	0.7935 ± 0.0031	0.7954 ± 0.0050
Flickr_LDL	0.7354 ± 0.0000	0.7436 ± 0.0000	.6295 ± 0.0000	.7779 ± 0.0000	**.7872 ± 0.0001**	.7370 ± 0.0014	.7663 ± 0.0010	.7751 ± 0.0004
Human_Gene	0.9324 ± 0.0034	0.9454 ± 0.0049	0.9462 ± 0.0052	0.9267 ± 0.0014	**0.9469 ± 0.0007**	0.9467 ± 0.0005	0.9381 ± 0.0001	0.9458 ± 0.0009
Movie	.9691 ± 0.0040	.9642 ± 0.0033	.9686 ± 0.0059	.9621 ± 0.0000	**0.9730 ± .00019**	0.9725 ± 0.0017	0.9612 ± 0.0019	0.9725 ± 0.0003
Avg.Rank	6.4	5.6	5.5	4.1	2.9	3.3	5.7	2.5

Table 12 Formulas of used distances

Name		Formula
Euclidean	$Distance_1$	$\sqrt{\sum_{k=1}^{q}(\mathbf{x}_{ik}-\mathbf{x}_{jk})^2}$
City	$Distance_2$	$\sum_{k=1}^{q}\lvert\mathbf{x}_{ik}-\mathbf{x}_{jk}\rvert$
Cosine	$Distance_3$	$1-\frac{\sum_{k=1}^{q}(\mathbf{x}_{ik}\mathbf{x}_{jk})}{\sqrt{\sum_{k=1}^{q}\mathbf{x}_{ik}^2}\sqrt{\sum_{k=1}^{q}\mathbf{x}_{jk}^2}}$
Intersection	$Distance_4$	$\sum_{k=1}^{q} min(\mathbf{x}_{ik},\mathbf{x}_{jk})$
Innerproduct	$Distance_5$	$\sum_{k=1}^{q}\mathbf{x}_{ik}\mathbf{x}_{jk}$

4.7 Label-correlation exploration

As mentioned earlier, our method exploits label correlation. In this section, we visualize the label correlation in different datasets to verify the rationality of the proposed method. The correlation metric learned by Pearson Correlation Coefficient [5] is shown in Fig. 6, where the original correlation coefficient is scaled to [0,1]. In detail, blue indicates negative correlation and yellow indicates positive correlation. From the results shown in the Fig. 6, we can intuitively see that label correlations are prevalent in different datasets. Even in the dataset Yeast-spo5 with only three labels, there are still two labels that are positively correlated. Therefore, it is reasonable and desirable to exploit label correlation in label distribution learning.

4.8 Distance measurement exploration

As mentioned above, we construct label-dependent and common features by computing distances to representative samples. But it is worth noting that no distance measurement can maintain its advantages and effectiveness in all scenarios. Therefore, we select five commonly used distances: Euclidean, City, Cosine, Intersection, and Innerproduct to construct features respectively. Their formulas are given in the table 12. We make predictions using features constructed with different distances. The table 13 shows the results of the datasets under different application scenarios. For the yeast data, we randomly selected 3 of the 9 datasets (Yeast-alpha, Yeast-cold and Yeast-spo). As indicated in Table 13, Euclidean and City distance guarantee that the algorithm can achieve satisfactory results on the yeast data. Furthermore, for scene data (Nature_Scene), City distance achieves best performance. In the recognition tasks of human facial expression (Emotion6 and Flicker_LDL) and human gene sequences (Human_Gene), Intersection and Innerproduct distance yield the best results for most metrics. In movie rating task (Movie), Euclidean and City distance are obviously the most suitable distance measures.

5 Conclusion and future works

We have proposed an LDL-LDF algorithm that aims to exploit label-dependent and common features. Compared with previous algorithms based on the same feature (sub)space, the proposed algorithm is more suitable for handling the relationship between labels and features in real-world scenarios. Specifically, we construct label-dependent features and common features by mining the characteristics and commonalities of labels through cluster analysis in local and global samples. For prediction, we design different networks for the two types of features. Parallel neural networks serve features with different labels, while shared neural networks serve shared features. We also exploit label correlation by limiting the output of labels. Finally, experiments show that LDL-LDF achieves performance improvements compared to state-of-the-art algorithms.

The following aspects of our algorithm can be further studied.

1) Multi-granularity feature extraction. Guided by multi-granularity [38, 39], we will explore new methods to construct label-dependent and common features for mining the characteristics and commonality of labels from different granularities.
2) Performance improvements for predictive models. We will employ new convolutional neural networks or graph attention networks to learn the mapping of label-dependent and common features to label distributions.
3) Application to label enhancement (LE) [12, 40]. LE aims to reinforce the supervision information in the training set by recovering the label distributions from the logical labels. Label-dependent and common

Table 13 Comparison results (mean ± std) using different distances on different datasets. ↓ indicates the smaller the better. ↑ indicates the bigger the better. The best results are shown in bold

dataset	distance	KL↓	Euclidean↓	Sϕrensen↓	Intersection↑	Fidelity↑
Yeast-alpha	$Distance_1$	**0.00550 ± 0.00002**	0.02316 ± 0.00007	0.03784 ± 0.00001	0.96216 ± 0.00000	**0.99860 ± 0.00000**
	$Distance_2$	0.00551 ± 0.00001	0.02308 ± 0.00008	**0.03766 ± 0.00001**	**0.96234 ± 0.00000**	0.99861 ± 0.00000
	$Distance_3$	0.00551 ± 0.00001	**0.02307 ± 0.00002**	0.03771 ± 0.00001	0.96229 ± 0.00010	0.99861 ± 0.00000
	$Distance_4$	0.00551 ± 0.00002	0.02308 ± 0.00001	0.03770 ± 0.00000	0.96230 ± 0.00010	0.99861 ± 0.00000
	$Distance_5$	0.00554 ± 0.00001	0.02313 ± 0.00006	0.03777 ± 0.00000	0.96223 ± 0.00000	0.99860 ± 0.00000
Yeast-cold	$Distance_1$	**0.01197 ± .00000**	**0.06761 ± .00000**	**0.05860 ± .00000**	0.94140 ± 0.00000	0.99694 ± 0.00000
	$Distance_2$	0.01201 ± 0.00000	0.06772 ± 0.00000	0.05873 ± 0.00000	0.94127 ± 0.00000	0.99694 ± 0.00000
	$Distance_3$	0.01207 ± 0.00000	0.06790 ± 0.00000	0.05889 ± 0.00000	**0.94111 ± 0.00000**	**0.99692 ± 0.00000**
	$Distance_4$	0.01207 ± 0.00000	0.06777 ± 0.00000	0.05876 ± 0.00000	0.94124 ± 0.00000	0.99692 ± 0.00001
	$Distance_5$	0.01205 ± 0.00000	0.06786 ± 0.00000	0.05890 ± 0.00000	0.94111 ± 0.00000	0.99692 ± 0.00001
Yeast-spo	$Distance_1$	0.02441 ± 0.00001	**0.08154 ± 0.00003**	0.08408 ± 0.00002	**0.91592 ± 0.00001**	**0.99370 ± 0.00005**
	$Distance_2$	0.02441 ± 0.00005	0.08157 ± 0.00002	0.08408 ± 0.00001	0.91592 ± 0.00002	0.99373 ± 0.00005
	$Distance_3$	0.02442 ± 0.00007	0.08160 ± 0.00004	0.08411 ± 0.00002	0.91589 ± 0.00006	0.99373 ± 0.00004
	$Distance_4$	0.02440 ± 0.00008	0.08156 ± 0.00005	**0.08396 ± 0.00006**	0.91604 ± 0.00004	0.99373 ± 0.00006
	$Distance_5$	**0.02438 ± 0.00002**	0.08167 ± 0.00007	0.08418 ± 0.00004	0.91582 ± 0.00008	0.99373 ± 0.00003
Nature_Scene	$Distance_1$	0.77965 ± 0.00001	0.42500 ± 0.00007	0.49625 ± 0.00001	0.50375 ± 0.0000	0.71568 ± 0.00001
	$Distance_2$	**0.74786 ± 0.00001**	**0.40624 ± 0.00008**	**0.47180 ± 0.00002**	**0.52820 ± 0.0000**	**0.73293 ± 0.00002**
	$Distance_3$	0.82101 ± 0.00002	0.43398 ± 0.00010	0.50493 ± 0.00003	0.49507 ± 0.0000	0.70914 ± 0.00003
	$Distance_4$	0.75480 ± 0.00005	0.40841 ± 0.00011	0.47667 ± 0.00004	0.52333 ± 0.0000	0.72984 ± 0.00001
	$Distance_5$	0.83865 ± 0.00003	0.43931 ± 0.00009	0.51283 ± 0.00005	0.48717 ± 0.0000	0.70333 ± 0.00002
Emotion6	$Distance_1$	**0.59451 ± 0.0000**	0.41204 ± 0.00001	0.42743 ± 0.00008	0.57257 ± 0.00005	0.79493 ± 0.00005
	$Distance_2$	0.59520 ± .0000	0.41173 ± .00002	0.42696 ± 0.00006	0.57304 ± 0.00005	0.79550 ± 0.00005
	$Distance_3$	0.61678 ± 0.0000	0.41965 ± 0.00003	0.43298 ± 0.00005	0.56702 ± 0.00003	0.79044 ± 0.00007
	$Distance_4$	0.59539 ± 0.0000	0.41220 ± 0.00002	0.42784 ± 0.00008	0.57216 ± 0.00002	0.79468 ± 0.00008
	$Distance_5$	0.59630 ± 0.0000	**0.41151 ± 0.00002**	0.42552 ± .00000	**0.57448 ± 0.00003**	**0.79621 ± 0.00003**
Flickr_LDL	$Distance_1$	0.69028 ± 0.00002	0.42072 ± 0.00002	0.44397 ± 0.00003	0.55603 ± 0.0000	0.74637 ± 0.0000
	$Distance_2$	**0.68988 ± .00002**	0.42059 ± 0.00003	0.44375 ± .00003	0.55625 ± .0000	0.74734 ± 0.0000
	$Distance_3$	0.71704 ± 0.00005	0.42684 ± 0.00004	0.44958 ± 0.00002	0.55042 ± 0.0000	0.74458 ± 0.0000
	$Distance_4$	0.68991 ± 0.00002	**0.41904 ± 0.00001**	0.44219 ± 0.00004	0.55781 ± 0.0000	0.74785 ± 0.0000
	$Distance_5$	0.68665 ± .00003	0.41646 ± .00002	**0.43984 ± .00005**	0.56016 ± 0.0000	**0.75089 ± 0.0000**
Human_Gene	$Distance_1$	0.23823 ± 0.00001	0.008661 ± 0.00003	0.21649 ± 0.00005	0.78351 ± 0.00006	0.94597 ± 0.00000
	$Distance_2$	0.23820 ± 0.00004	0.08660 ± 0.00006	0.21649 ± 0.00002	0.78351 ± 0.00001	0.94597 ± 0.00006
	$Distance_3$	0.23826 ± 0.00002	0.08662 ± 0.00001	0.21652 ± 0.00000	0.78348 ± 0.00001	0.94596 ± 0.00002
	$Distance_4$	**0.23818 ± 0.00006**	**0.08660 ± 0.00001**	**0.21648 ± 0.00001**	0.78352 ± 0.00004	0.94598 ± 0.00008
	$Distance_5$	0.23836 ± 0.00002	0.08664 ± 0.00002	0.21661 ± .00003	**0.78339 ± .00001**	**0.94594 ± .00001**
Movie	$Distance_1$	**0.17267 ± 0.00001**	**0.24555 ± 0.00001**	**0.24219 ± 0.00006**	0.75781 ± 0.00003	0.95243 ± 0.00003
	$Distance_2$	0.17432 ± 0.00005	0.24713 ± 0.00001	0.24368 ± 0.00001	**0.75632 ± 0.00008**	**0.95199 ± 0.00004**
	$Distance_3$	0.17358 ± 0.00008	0.24653 ± 0.00002	0.24309 ± 0.00005	0.75691 ± 0.00007	0.95218 ± 0.00001
	$Distance_4$	0.17451 ± 0.00008	0.24631 ± 0.00006	0.24321 ± 0.00008	0.75640 ± 0.00007	0.95230 ± 0.00002
	$Distance_5$	0.17379 ± 0.00003	0.24651 ± 0.00003	0.24295 ± 0.00000	0.75658 ± 0.00002	0.95227 ± 0.00001

features can be introduced to LE for performance improvement.

Acknowledgements This work is supported by the National Natural Science Foundation of China (61902328), the Applied Basic Research Project of Science and Technology Bureau of Nanchong City (SXHZ040), and Central Government Funds of Guiding Local Scientific and Technological Development (2021ZYD0003).

Declarations

Conflict of Interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work. There is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

References

1. Zhang M-L, Zhou Z-H (2013) A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng 26(8):1819–1837
2. Gao B-B, Xing C, Xie C-W, Wu J-X, Geng X (2017) Deep label distribution learning with label ambiguity. IEEE Trans Imag Process 26(6):2825–2838
3. Geng X, Smith-Miles K, Zhou Z-H (2010) Facial age estimation by learning from label distributions. In: AAAI, pp. 451– 456
4. Wen X, Li B, Guo H, Liu Z, Hu G, Tang M, Wang J (2020) Adaptive variance based label distribution learning for facial age estimation. In: ECCV, pp. 379– 395
5. Zhou Y, Xue H, Geng X (2015) Emotion distribution recognition from facial expressions. In: ACM MM, pp. 1247– 1250
6. Li S, Deng W (2019) Blended emotion in-the-wild: multi-label facial expression recognition using crowdsourced annotations and deep locality feature learning. Inter J Comput Vision 127(6):884–906
7. Geng X, Xia Y (2014) Head pose estimation based on multivariate label distribution. In: CVPR, pp. 1837– 1842
8. Zhang Z, Wang M, Geng X (2015) Crowd counting in public video surveillance by label distribution learning. Neurocomputing 166:151–163
9. Liang L, Lin L, Jin L, Xie D, Li M (2018) Scut-fbp5500: a diverse benchmark dataset for multi-paradigm facial beauty prediction. In: ICPR, pp. 1598– 1603
10. Yang J, She D, Sun M (2017) Joint image emotion classification and distribution learning via deep nonvolutional neural network. In: IJCAI, pp. 3266– 3272
11. Zhang M-L, Wu L (2014) Lift: Multi-label learning with label-specific features. IEEE Trans Pattern Anal Machine Intell 37(1):107–120
12. Xu N, Liu Y-P, Geng X (2019) Label enhancement for label distribution learning. IEEE Trans Knowl Data Eng 33(4):1632–1643
13. Xu N, Lv J, Geng X (2019) Partial label learning via label enhancement. AAAI 33:5557–5564
14. Gao Y, Wang K, Geng X (2022) Sequential label enhancement. IEEE Transactions on Neural Networks and Learning Systems
15. Du G, Zhang J, Jiang M, Long J, Lin Y, Li S, Tan KC (2021) Graph-based class-imbalance learning with label enhancement. IEEE Transactions on Neural Networks and Learning Systems
16. Geng X (2016) Label distribution learning. IEEE Trans Knowl Data Eng 28(7):1734–1748
17. Xing C, Geng X, Xue H (2016) Logistic boosting regression for label distribution learning. In: CVPR, pp. 4489– 4497
18. Shen W, Zhao K, Guo Y, Yuille AL (2017) Label distribution learning forests. Advances in Neural Information Processing Systems **30**
19. Jia X, Zheng X, Li W, Zhang C, Li Z (2019) Facial emotion distribution learning by exploiting low-rank label correlations locally. In: CVPR, pp. 9841– 9850
20. Xu S, Shang L, Shen F (2019) Latent semantics encoding for label distribution learning. In: IJCAI, pp. 3982– 3988
21. Zheng X, Jia X, Li W (2018) Label distribution learning by exploiting sample correlations locally. In: AAAI, pp. 4556– 4563
22. Jia X, Li W, Liu J-Y, Zhang Y (2018) Label distribution learning by exploiting label correlations. In: AAAI, pp. 3310– 3317
23. Ren T, Jia X, Li W, Chen L, Li Z (2019) Label distribution learning with label-specific features. In: IJCAI, pp. 3318– 3324
24. Huang J, Li G, Huang Q, Wu X (2015) Learning label specific features for multi-label classification. In: ICDM, pp. 181– 190
25. Zhang J, Li C, Cao D, Lin Y, Su S, Dai L, Li S (2018) Multi-label learning with label-specific features by resolving label correlations. Knowl-Based Syst 159:148–157
26. Lin Y, Liu H, Zhao H, Hu Q, Zhu X, Wu X (2022) Hierarchical feature selection based on label distribution learning. IEEE Transactions on Knowledge and Data Engineering
27. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344(6191):1492–1496
28. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proceed Nat Acad Sci 95(25):14863–14868
29. Peng K-C, Chen T, Sadovnik A, Gallagher AC (2015) A mixed bag of emotions: model, predict, and transfer emotion distributions. In: CVPR, pp. 860– 868
30. Yu J-F, Jiang D-K, Xiao K, Jin Y, Wang J-H, Sun X (2012) Discriminate the falsely predicted protein-coding genes in aeropyrum pernix k1 genome based on graphical representation. Match-Commun Mathe Comput Chem 67(3):845–866
31. Zhang H-R, Huang Y-T, Xu Y-Y, Min F (2020) COS-LDL: Label distribution learning by cosine-based distance-mapping correlation. IEEE Access 8:63961–63970
32. Jia X, Ren T, Chen L, Wang J, Zhu J, Long X (2019) Weakly supervised label distribution learning based on transductive matrix completion with sample correlations. Pattern Recognit Lett 125:453–462
33. Jia X, Li Z, Zheng X, Li W, Huang S-J (2021) Label distribution learning with label correlations on local samples. IEEE Trans Knowl Data Eng 33(04):1619–1631
34. Wang J, Geng X (2021) Label distribution learning by exploiting label distribution manifold. IEEE Trans Neural Netw Learn Syst 01:1–14
35. Wang J , Geng X (2021) Learn the highest label and rest label description degrees. In: IJCAI, pp. 3097– 3103
36. Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. Annals Math Stat 11(1):86–92
37. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Machine Learn Res 7:1–30
38. Li J, Huang C, Qi J, Qian Y, Liu W (2017) Three-way cognitive concept learning via multi-granularity. Inform Sci 378:244–263
39. Qian Y, Liang J, Wu W-Z, Dang C (2010) Information granularity in fuzzy binary grc model. IEEE Trans Fuzzy Syst 19(2):253–264
40. Jia X, Lu Y, Zhang F (2021) Label enhancement by maintaining positive and negative label relation. IEEE Trans Knowledge Data Eng 1(1):1–1