Information Sciences 678 (2024) 120954

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Exploiting asymmetric influence between instances for label enhancement

Heng-Ru Zhang [a,*], Peng-Cheng Li [a], Yuan-Yuan Xu [a], Fan Min [a,b,c]

[a] School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu 610500, China
[b] Lab of Machine Learning, Southwest Petroleum University, Chengdu 610500, China
[c] Institute for Artificial Intelligence, Southwest Petroleum University, Chengdu 610500, China

ARTICLE INFO

Keywords:
Asymmetric influence
Bipartite network
Label enhancement
Multi-label learning

ABSTRACT

In multi-label learning, label enhancement (LE) aims to recover label distributions from logical labels, thereby reinforcing supervision information in the training set. Existing LE algorithms mainly leverage pairwise similarities between instances to recover label distributions. However, symmetric similarities, which are widely used, are incapable of reflecting individual differences between instances. In this paper, we propose an asymmetric influence between instances to fully explore the information contained in logical labels. First, we build a bipartite network and employ the mass diffusion algorithm on it to calculate the asymmetric influence between instances. Such asymmetry distinguishes the influence of an instance on others from the influence it receives from others. Second, to cope with the new influence, we construct a graph with self-connections. In this way, we allow instances to be influenced by themselves. Extensive experiments on thirteen benchmark datasets demonstrate the superiority of the proposed method over six state-of-the-art LE algorithms.

1. Introduction

Label ambiguity [7] refers to the uncertainty among the ground-truth labels. Multi-label learning (MLL) [26,28,36] addresses the label ambiguity problem of whether an instance has certain labels. For data representation, if the label is related to an instance, the corresponding value will be set to 1, otherwise it will be 0. Label distribution learning (LDL) [9,12,19] addresses the label ambiguity problem of how well labels can describe an instance. In LDL, each instance is assigned a label distribution vector. Each element of the vector ranges from 0 to 1, representing the degree to which the label describes the instance. LDL is more general than MLL because quantitative descriptions of labels are required in some complex semantic tasks, such as age estimation [8] and sentiment classification [24]. However, since the high cost of the process of quantifying the description degree [23], it is difficult to obtain the label distribution directly.

To reinforce supervision information in the existing multi-label datasets, label enhancement (LE) [22] is proposed to recover label distributions from logical labels. Existing LE algorithms [5,10,33] mainly leverage the topological information of the feature space and the correlation among the labels. They primarily compute pairwise similarity in feature space before transferring it to label space. Symmetric similarity means that a pair of instances have the same influence on each other, ignoring the fact that there are actually individual differences in the real world.

* Corresponding author.
E-mail address: zhanghr@swpu.edu.cn (H.-R. Zhang).

https://doi.org/10.1016/j.ins.2024.120954
Received 15 February 2023; Received in revised form 11 December 2023; Accepted 5 June 2024

Available online 10 June 2024

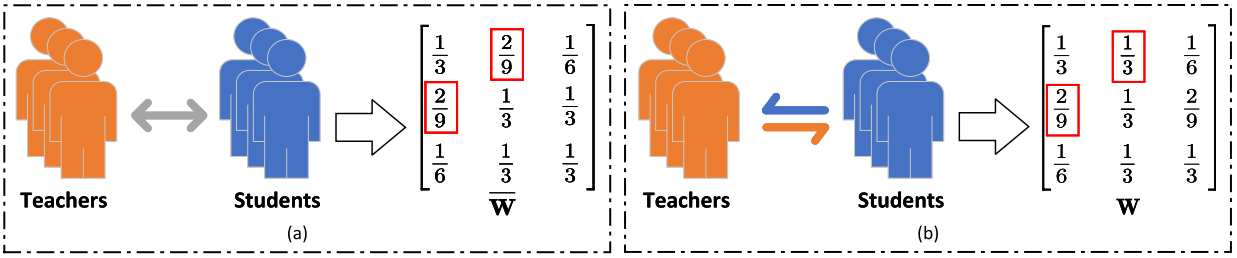

Fig. 1. Examples of symmetrical and asymmetrical influences between teachers and students. (a) The gray double-headed arrow represents peer-to-peer influence between teachers and students, which has been quantified as a symmetric matrix $\overline{\mathbf{W}}$. (b) Two single colored arrows indicate unequal influence between teacher and student, which has been quantified as an asymmetric matrix $\mathbf{W}$.

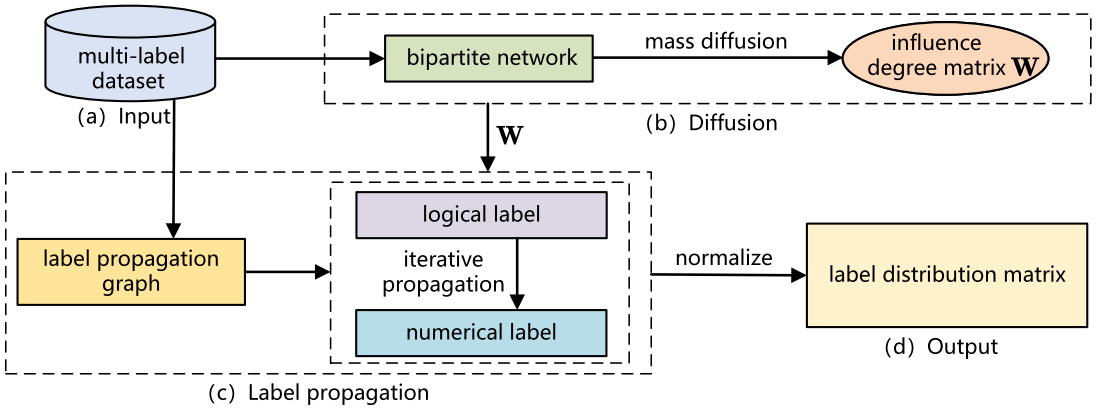

Fig. 2. Flowchart for the AILE algorithm. (a) Input: A multi-label dataset is used as input for the diffusion and label propagation stages. (b) Diffusion: The mass diffusion algorithm is employed on the bipartite network to obtain the influence degree matrix $\mathbf{W}$. (c) Label propagation: The matrix $\mathbf{W}$ is used to generate an edge-weighted graph with self-connections. Following iterative propagation in the graph, logical labels are transformed into numerical ones. (d) Output: The label distribution matrix is obtained by performing a softmax normalization on the numerical labels.

In this paper, we exploit asymmetric influence between instances for label enhancement (AILE). Since no label distribution information participates in the training process, our proposed AILE runs only in the unsupervised learning framework [33]. Semi-supervised learning effectively utilizes labeled and unlabeled samples to improve model generalization, such as MixMatch [1]. In contrast, our method is self-contained, extracting patterns and relationships solely from unlabeled data without the assistance of labeled instances.

Fig. 1 depicts the main idea. Let $\overline{\mathbf{W}} = [\overline{w}_{ij}]_{n\times n}$ and $\mathbf{W} = [w_{ij}]_{n\times n}$ be symmetric and asymmetric matrices, respectively. In the case of a symmetric relation, the expression $\overline{w}_{ij} = \overline{w}_{ji}$ represents that instance i has the same influence on instance j as vice versa. In the case of an asymmetric relation, the influence of instance i on instance j is not necessarily the same as the influence of i on j, i.e., in most cases $w_{ij} \neq w_{ji}$. In Fig. 1(a), $\overline{w}_{12} = \overline{w}_{21}$, indicating that the 1st teacher and 2nd student have the same mutual influence. In Fig. 1(b), $w_{12} \neq w_{21}$, indicating that the 1st teacher and 2nd student have different influences on each other. In the real world, the mutual influence of teachers and students is usually not the same, so an asymmetric matrix $\mathbf{W}$ is more appropriate for characterizing this influence.

Fig. 2 depicts the learning process of $\mathbf{W}$ and its application in label enhancement. Fig. 2(a) takes a multi-label dataset as input. Fig. 2(b) shows the asymmetric matrix $\mathbf{W}$ of influence degree between instances obtained by employing the mass diffusion (MD) algorithm [14,32] on a bipartite network [2,15]. The bipartite network employs instances and labels as nodes, with the instance nodes receiving initial resources. MD first evenly distributes the initial resources among label nodes linked to instance nodes. The label node then returns the newly acquired resource to the instance node to which it is connected in the same way. The asymmetric matrix $\mathbf{W}$ is formed by the ratio of final resources to initial resources on instance nodes. Fig. 2(c) constructs a self-connected graph with $\mathbf{W}$ as edge weights. The label information of all instances is iteratively propagated on the graph. Iterative label propagation technique [11,18] is employed in here. In each iteration, the label information of each node is distributed simultaneously to other nodes based on its weights. Fig. 2(d) depicts the output of the final label distribution when the label information becomes stable after multiple iterations.

To validate the proposed AILE algorithm, we conducted extensive experiments on thirteen publicly available datasets. The experimental results show that the asymmetric influence is more reasonable than the symmetric one. AILE is stable and performing admirably, just as we expected.

In summary, the major contributions of this paper are:

Table 1
Notations.

Notation	Meaning
$\mathcal{X}$	The instance space
$\mathcal{Y}$	The label space
$\mathbf{x}_i$	The i-th instance
y_j	The j-th label
n	The number of instances
m	The number of labels
$\mathbf{l}_i$	The logical label vector of $\mathbf{x}_i$
$\mathbf{d}_i$	The label distribution vector of $\mathbf{x}_i$
$\mathbf{W}$	The influence degree matrix
$\mathbf{T}$	The initial label information matrix

- We propose AILE, a new LE algorithm that can more fully utilize the logical label information of the instance to recover label distributions.
- For the purpose of defining individual differences, we construct the asymmetric influences between instances.
- We introduce the MD algorithm to obtain asymmetric influences. To the best of our knowledge, this is the first attempt at applying MD to the field of LE.

The rest of this paper is organized as follows. Section 2 presents the formulation of LE and briefly reviews related works on the LE method. Section 3 introduces the details of our method. Section 4 reports the results of comparative experiments. Finally, Section 5 summarizes the conclusions of this paper. The source code of AILE is available at https://github.com/zhanghrswpu/AILE.

2. Related work

In this section, we present the formulation of label enhancement and briefly review the existing LE algorithms. Table 1 lists the notations used in this paper.

2.1. Formulation of label enhancement

Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ denote the instance space, where $\mathbf{x}_i = \left(x_{i1}, x_{i2}, \ldots, x_{ic}\right)^{\top}$ denotes the i-th instance. $\mathcal{Y} = \{y_1, y_2, \ldots, y_m\}$ denotes the label space, where y_j denotes the j-th label. For a specific instance $\mathbf{x}_i \in \mathcal{X}$, the logical label vector of $\mathbf{x}_i$ is denoted by $\mathbf{l}_i = \left(l_{\mathbf{x}_i}^{y_1}, l_{\mathbf{x}_i}^{y_2}, \ldots, l_{\mathbf{x}_i}^{y_m}\right)^{\top}$ and each element $l_{\mathbf{x}_i}^{y_j} \in \{0, 1\}$. Correspondingly, $\mathbf{d}_i = \left(d_{\mathbf{x}_i}^{y_1}, d_{\mathbf{x}_i}^{y_2}, \ldots, d_{\mathbf{x}_i}^{y_m}\right)^{\top}$ denotes the label distribution of $\mathbf{x}_i$ where $d_{\mathbf{x}_i}^{y_j} \in [0, 1]$ and $\sum_{j=1}^{m} d_{\mathbf{x}_i}^{y_j} = 1$.

Given a multi-label dataset $S = \{(\mathbf{x}_i, \mathbf{l}_i) \mid 1 \le i \le n\}$, where $\mathbf{x}_i \in \mathcal{X}$ and $\mathbf{l}_i \in \{0, 1\}^m$. The goal of LE is to recover the label distribution $\mathbf{d}_i$ of $\mathbf{x}_i$ from the logical label $\mathbf{l}_i$, thereby transforming the multi-label dataset into a label distribution dataset $S = \{(\mathbf{x}_i, \mathbf{d}_i) \mid 1 \le i \le n\}$.

2.2. Existing LE algorithm

The existing LE algorithms are mainly divided into two categories: fuzzy-based label enhancement and graph-based label enhancement.

Fuzzy-based LE algorithms mainly include the fuzzy clustering-based LE algorithm (FCM) [5] and the kernel-based LE algorithm (KM) [13]. FCM employs fuzzy C-means clustering [17] to achieve the LE process. The membership degree of an instance in a cluster is obtained from the Euclidean distance. Then, the membership degree vector of each instance to labels is obtained by fuzzy composition. Softmax normalization is employed to get label distribution. KM utilizes the fuzzy membership function in a fuzzy SVM to achieve the LE process. The instances are projected into a high-dimensional space by a kernel function. Then, instances are divided into two classes according to the logical labels. The label distribution is obtained by calculating the distance between the instance and the center of the class.

Graph-based LE algorithms mainly include the manifold learning-based LE algorithm (ML^2) [10], the graph laplacian-based LE algorithm (GLLE) [22], and the label propagation-based LE algorithm (RELIAB-LP) [27]. ML^2 utilizes the smoothness assumption [34, 35] to achieve the LE. The feature space and the label space share the same topological structures, which are obtained by the linear combination of the neighbors. GLLE leverages the graph information in the feature space to achieve LE. The local topological structures are obtained by calculating the distance from the neighbors. A constructed graph is combined to learn a mapping model to get the label distribution. RELIAB-LP employs the label propagation technique [30] to achieve LE. The weights are obtained by calculating the distance between all instances. Then the label information is propagated iteratively according to the weights to get the label distribution.

To sum up, fuzzy-based LE algorithms lack exploration of sample correlations. Graph-based LE algorithms mainly exploit the associations between instances in the feature space and subsequently transfer them to the label space. Meanwhile, the calculation of the edge weight of the graph based on the variance of the Gaussian function can significantly affect performance. Its symmetric weights are also unreasonable for real-world learning problems.

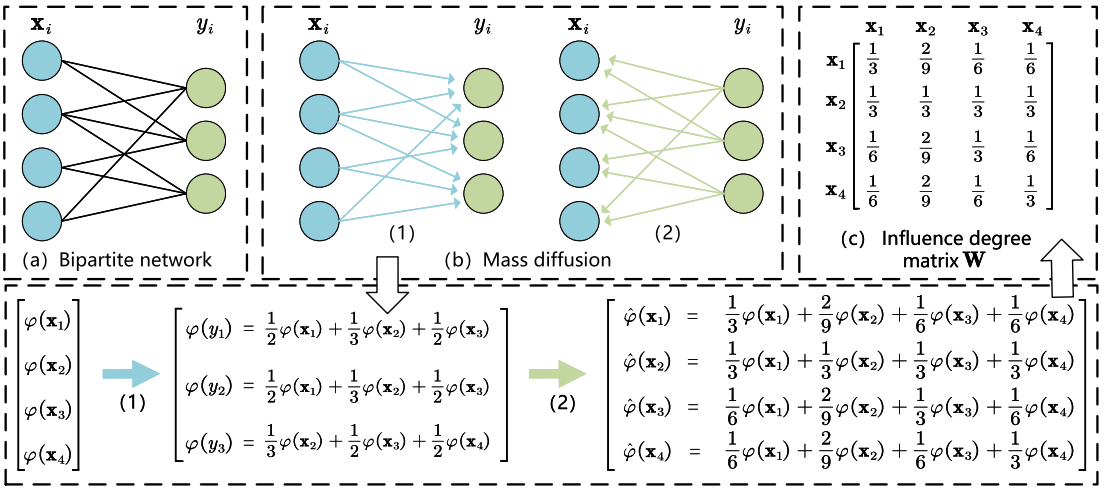

Fig. 3. Diffusion. (a) Bipartite network: $\mathbf{x}_i$ denotes the i-th instance node and y_j denotes the j-th label node. (b) Mass diffusion: $\varphi(\mathbf{x}_i)$ refers to the resource on the node $\mathbf{x}_i$, while $\varphi(y_j)$ refers to the resource on the node y_j. The arrow (1) indicates that resources are evenly distributed from node $\mathbf{x}$ to its connecting node y, whereas the arrow (2) indicates that resources are evenly distributed from node y to its connecting node $\mathbf{x}$. (c) Influence degree matrix $\mathbf{W}$: It shows the asymmetric influence between instances.

3. The proposed method

This section introduces AILE, which considers the asymmetric influence between instances. In a training set $S = \{(\mathbf{x}_i, \mathbf{l}_i) \mid 1 \le i \le n\}$, $\mathbf{l}_i = \left(l_{\mathbf{x}_i}^{y_1}, l_{\mathbf{x}_i}^{y_2}, \ldots, l_{\mathbf{x}_i}^{y_m}\right)^{\mathsf{T}}$ is the logic label vector of instance $\mathbf{x}_i$ and each element $l_{\mathbf{x}_i}^{y_j} \in \{0, 1\}$. $l_{\mathbf{x}_i}^{y_j} = 1$ represents that the label y_j is associated with instance $\mathbf{x}_i$.

3.1. Calculation of asymmetric influences

According to the logic labels $l_{\mathbf{x}_i}^{y_j}$, we construct a bipartite network $N = (X, Y, E)$ where $X \cap Y = \emptyset$. $X = \{X_1, X_2, \ldots, X_n\}$ is the set of X-nodes, where X_i denotes the instance $\mathbf{x}_i$. $Y = \{Y_1, Y_2, \ldots, Y_m\}$ is the set of Y-nodes, where Y_j denotes the label y_j. $E \subseteq X \times Y$ is the set of edges between instances and labels. If $l_{\mathbf{x}_i}^{y_j} = 1$, there is an edge (X_i, Y_j) between instance $\mathbf{x}_i$ and label y_j.

We let resources diffuse in the bipartite network. The whole propagation process called mass diffusion (MD) is divided into two steps. The resource first flows from the X-node to the Y-node, then from the Y-node back to the X-node. Since the network is unweighted, we do not need to consider the influence of edge weights. The resources of each node should be evenly distributed to its connecting nodes. $\mathbf{A}$ is an $n \times m$ adjacent matrix composed by

$$\forall_{i=1}^{n} \forall_{j=1}^{m} : \quad a_{ij} = \begin{cases} 1, & \text{if } (X_i, Y_j) \in E; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Let $\varphi(X_i) \ge 0$ be the initial resource of X_i, and the diffuse process from X to Y is represented by

$$\varphi(Y_j) = \sum_{i=1}^{n} \frac{a_{ij}\varphi(X_i)}{h(X_i)}, \tag{2}$$

where $\varphi(Y_j)$ is the resource received by Y_j, and $h(X_i)$ is the degree of X_i. The next step (from Y to X) is represented by

$$\hat{\varphi}(X_i) = \sum_{j=1}^{m} \frac{a_{ij}\varphi(Y_j)}{h(Y_j)} = \sum_{j=1}^{m} \frac{a_{ij}}{h(Y_j)} \sum_{c=1}^{n} \frac{a_{cj}\varphi(X_c)}{h(X_c)}. \tag{3}$$

According to the MD, we can construct an influence degree matrix $\mathbf{W} = [w_{ij}]_{n \times n}$ where

$$\forall_{i=1}^{n} \forall_{j=1}^{n} : \quad w_{ij} = \sum_{c=1}^{m} \frac{a_{ic}a_{jc}}{h(X_j)h(Y_c)}. \tag{4}$$

$\mathbf{W}$ is an asymmetric matrix that quantifies the degree of mutual influence between each instance. w_{ij} denotes the degree of influence of the i-th instance on the j-th instance. Fig. 3 shows an example of an asymmetric influence created by mass diffusion. Fig. 3(a) is the bipartite network. Fig. 3(b) shows the resource changes at different nodes during the process of mass diffusion. The resources on each node will be allocated evenly to its connected nodes in each diffusion. Fig. 3(c) is the influence degree matrix $\mathbf{W}$, which shows the asymmetric influence between instances.

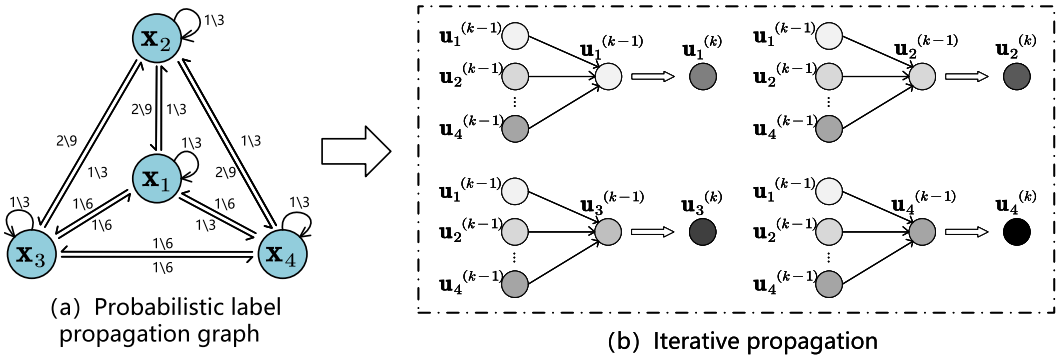

Fig. 4. Label propagation. (a) Probabilistic label propagation graph: It is a weighted directed complete graph that allows self-connection. (b) Iterative propagation: $\mathbf{u}_i^{(\kappa-1)}$ refers to the label information of instance $\mathbf{x}_i$ after the $(\kappa-1)$-th iteration, while $\mathbf{u}_i^{(\kappa)}$ refers to the new label information after the (κ)-th iteration. The propagation ratio of $\mathbf{u}_i^{(\kappa-1)}$ to $\mathbf{u}_i^{(\kappa)}$ is determined by the weight on the edge in (a).

3.2. Iterative propagation

Based on logical labels, an initial label information matrix $\mathbf{T}$ is constructed by

$$\forall_{i=1}^{n}\forall_{j=1}^{m}: \quad t_{ij}=\begin{cases}1, & \text{if } l_{\mathbf{x}_i}^{y_j}=1;\\ 0, & \text{otherwise.}\end{cases} \tag{5}$$

Let $\mathbf{U}=[u_{ij}]_{n\times m}$ be a non-negative matrix, and $\mathbf{U}^{(0)}=\mathbf{T}$. We construct a label propagation graph $G=(V,E,P)$ where $V=X$, $E=\{\langle v_i,v_j\rangle | v_i,v_j\in V\}$ is the edge set, and P is the set of weights on edges. G is a weighted directed complete graph that allows self-connection. The weight p_{ij} of each edge $\langle v_i,v_j\rangle$ in the graph G is determined by the matrix $\mathbf{W}=[w_{ij}]_{n\times n}$:

$$\forall_{i=1}^{n}\forall_{j=1}^{n}: \quad p_{ij}=w_{ji}. \tag{6}$$

It is worth mentioning that since the diagonal elements of the matrix $\mathbf{W}$ are non-zero, each instance can obtain information from itself, which ensures comprehensive access to information and avoids huge errors due to inaccurate labels in other instances. The label information is propagated iteratively in the graph, and the matrix $\mathbf{U}$ is updated by

$$\mathbf{U}^{(\kappa)}=\alpha\mathbf{W}^{\top}\mathbf{U}^{(\kappa-1)}+(1-\alpha)\mathbf{T}, \tag{7}$$

where κ is the number of iterations, and $\alpha\in(0,1)$ is a balance parameter to control the influence of label propagation ($\mathbf{W}^{\top}\mathbf{U}^{(\kappa-1)}$) and initial labels ($\mathbf{T}$) on new label information. During each iteration, each instance receives information from itself and other instances while retaining its initial information. It is worth mentioning that since the diagonal elements of the influence degree matrix W are not set to zero, each instance is affected not only by other instances but also by itself. Moreover, since W is an asymmetric matrix, the label information is propagated asymmetrically between instances. The condition to end the iteration is

$$\sum_{i=1}^{n}\left|u_i^{(\kappa)}-u_i^{(\kappa-1)}\right|\le 10^{-10}. \tag{8}$$

When the label propagation process ends, $\mathbf{U}^{(\kappa)}$ will converge to $\mathbf{U}^{*}$. Fig. 4 shows an example of the probabilistic label propagation graph and the iterative propagation process. Fig. 4(a) is the probabilistic label propagation graph. The nodes indicate the instances, and the weight of each edge is determined by the influence degree matrix $\mathbf{W}$. Fig. 4(b) shows the κ-th iterative propagation process. Each node indicates the label information for a specific instance. The descriptive values of each label for each instance are determined by normalizing each row of $\mathbf{U}^{*}$:

$$\forall_{i=1}^{n}\forall_{j=1}^{m}: \quad d_{\mathbf{x}_i}^{y_j}=\frac{\exp(u_{ij}^{*})}{\sum_{c=1}^{m}\exp(u_{ic}^{*})}. \tag{9}$$

In other words, $\mathbf{d}_i=\left(d_{\mathbf{x}_i}^{y_1},d_{\mathbf{x}_i}^{y_2},\ldots,d_{\mathbf{x}_i}^{y_m}\right)^{\top}$ is the label distribution of instance $\mathbf{x}_i$.

3.3. Time complexity analysis

We elaborate on the time complexity of Algorithm 1 as follows.

Proposition 1. *The time complexity of the proposed algorithm is* $\mathcal{O}(Kn^2m)$.

Algorithm 1 The AILE algorithm.

Input: $S = \{(\mathbf{x}_i, \mathbf{l}_i) \mid 1 \le i \le n\}$ ← The multi-label data set.
Parameter: α. ← The balance parameter.
Output: $D = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n]$. ← The recovered label distribution matrix.

1: Construct $\mathbf{W}$ base on Eq. (4); ← The influence degree matrix.
2: Initialize $\mathbf{T}$ and $\mathbf{U}^{(0)}$ according to Eq. (5); ← The initial label information matrix.
3: $\kappa = 0$; ← The number of iterations.
4: **repeat**
5: $\quad \mathbf{U}^{(\kappa)} = \alpha \mathbf{W}^{\mathrm{T}} \mathbf{U}^{(\kappa-1)} + (1-\alpha)\mathbf{T}$;
6: $\quad \kappa = \kappa + 1$;
7: **until** The iteration converges to U^*
8: Calculate $\{d_{\mathbf{x}_i}^{y_j} \mid 1 \le i \le n, 1 \le j \le m\}$ according to Eq. (9);
9: Obtain $D = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n]$. ← The recovered label distribution matrix.

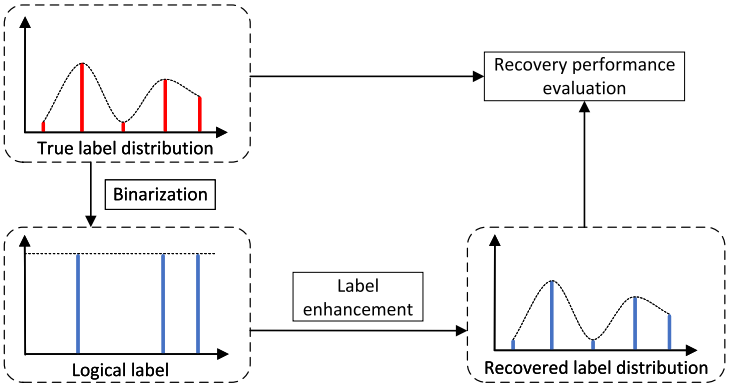

Fig. 5. The flowchart of label enhancement experiments. Specifically, the ground-true label distributions are first binarized to obtain the logical labels, and then the LE method is used to recover the label distributions. Finally, we compare the recovered label distributions with the ground-truth distributions to evaluate the performance of the recovery.

Proof 1. Suppose our algorithm iterates K times. Line 1 constructs $\mathbf{W}$ in $\mathcal{O}(n^2m)$. Line 2 initializes $\mathbf{T}$ and $\mathbf{U}^{(0)}$ in $\mathcal{O}(nm)$. Line 3 through 7 update $\mathbf{U}^{(\kappa)}$ in $\mathcal{O}(Kn^2m)$. Line 8 calculates $\{d_{\mathbf{x}_i}^{y_j} \mid 1 \le i \le n, 1 \le j \le m\}$ in $\mathcal{O}(nm^2)$. Line 9 obtains $D = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_n]$ in $\mathcal{O}(n)$.

Because m is much smaller than n, the total time complexity is $\mathcal{O}(n^2m) + \mathcal{O}(nm) + \mathcal{O}(Kn^2m) + \mathcal{O}(nm^2) + \mathcal{O}(n) = \mathcal{O}(Kn^2m)$.

4. Experiments

In this section, we conduct extensive experiments and present the experimental results to verify the effectiveness and superiority of AILE. Experiments are undertaken on thirteen benchmark datasets with three distance measures and two similarity measures. The flowchart of experiments is shown in Fig. 5.

4.1. Datasets

There are one artificial dataset and twelve real-world datasets used in our experiments. Some basic statistics about these thirteen datasets are given in Table 2.

The artificial dataset [22] is created as follows. Each instance $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})$. x_{i1} and x_{i2} are located on a grid in the interval of 0.04 in the range $[-1, 1]$ and x_{i3} is calculated by

$$x_{i3} = \sin(x_{i1} + x_{i2})\,\pi. \tag{10}$$

The label distribution $\mathbf{d}_i = \left(d_{\mathbf{x}_i}^{y_1}, d_{\mathbf{x}_i}^{y_2}, d_{\mathbf{x}_i}^{y_3}\right)$ is created by following equations.

$$r_j = kx_{ij} + px_{ij}^2 + qx_{ij}^3 + z, j = 1, 2, 3, \tag{11}$$

$$\psi_1 = \left(\mathbf{t}_1^{\mathrm{T}}\mathbf{r}\right)^2, \psi_2 = \left(\mathbf{t}_2^{\mathrm{T}}\mathbf{r} + \gamma_1\psi_1\right)^2, \psi_3 = \left(\mathbf{t}_3^{\mathrm{T}}\mathbf{r} + \gamma_2\psi_2\right)^2, \tag{12}$$

$$d_{\mathbf{x}_i}^{y_j} = \frac{\psi_i}{\psi_1 + \psi_2 + \psi_3}, j = 1, 2, 3, \tag{13}$$

where $\mathbf{r} = (r_1, r_2, r_3)^{\mathrm{T}}, x_{ij} \in [-1, 1], k = 1, p = 0.5, q = 0.2, z = 1, \mathbf{t}_1^{\mathrm{T}} = (4, 2, 1), \mathbf{t}_2^{\mathrm{T}} = (1, 2, 4), \mathbf{t}_3^{\mathrm{T}} = (1, 4, 2)$, and $\gamma_1 = \gamma_2 = 0.01$. SJAFFE [16] and SBU_3DFE [25] are datasets of facial expression images. The yeast datasets [4] from Yeast-spo5 to Yeast-alpha are datasets of biological experiments collected on yeast genes. Movie [9] is a dataset collected from movies.

Table 2
The detailed statistics of the thirteen datasets.

No.	Dataset	#Examples	#Features	#Labels
1	Artificial	2,601	3	3
2	SJAFFE	213	243	6
3	SBU_3DFE	2,500	243	6
4	Yeast-spo5	2,465	24	3
5	Yeast-dtt	2,465	24	4
6	Yeast-cold	2,465	24	4
7	Yeast-heat	2,465	24	6
8	Yeast-spo	2,465	24	6
9	Yeast-diau	2,465	24	7
10	Yeast-elu	2,465	24	14
11	Yeast-cdc	2,465	24	15
12	Yeast-alpha	2,465	24	18
13	Movie	7,755	1,869	5

Table 3
Introduction to evaluation metrics.

	Name	Formula
Distance	Cheb↓	$Dis_1(\mathbf{d},\hat{\mathbf{d}}) = max_j \left\| d^{y_j} - \hat{d}^{y_j} \right\|$
	Clark↓	$Dis_2(\mathbf{d},\hat{\mathbf{d}}) = \sqrt{\sum_{j=1}^{m} \frac{(d^{y_j} - \hat{d}^{y_j})^2}{(d^{y_j} + \hat{d}^{y_j})^2}}$
	Canber↓	$Dis_3(\mathbf{d},\hat{\mathbf{d}}) = \sum_{j=1}^{m} \frac{\lvert d^{y_j} - \hat{d}^{y_j} \rvert}{d^{y_j} + \hat{d}^{y_j}}$
Similarity	Cosine↑	$Sim_1(\mathbf{d},\hat{\mathbf{d}}) = \frac{\sum_{j=1}^{m} d^{y_j} \hat{d}^{y_j}}{\sqrt{\sum_{j=1}^{m} (d^{y_j})^2} \sqrt{\sum_{j=1}^{m} (\hat{d}^{y_j})^2}}$
	Intersec↑	$Sim_2(\mathbf{d},\hat{\mathbf{d}}) = \sum_{j=1}^{m} min(d^{y_j}, \hat{d}^{y_j})$

It is worth noting that due to the lack of datasets with both logical labels and label distributions, the logical labels must be obtained by binarizing the ground-truth label distribution in the original dataset. To ensure consistency of evaluation, we employ the same binarization method as in GLLE [22]. Specifically, for each instance $\mathbf{x}_i$, we find the largest descriptive degree $d_{\mathbf{x}_i}^{y_j}$ in its label distribution vector $\mathbf{d}_i = \left(d_{\mathbf{x}_i}^{y_1}, d_{\mathbf{x}_i}^{y_2}, \ldots, d_{\mathbf{x}_i}^{y_m}\right)^{\top}$ and consider label y_j as the relevant label of $\mathbf{x}_i$, i.e., $l_{\mathbf{x}_i}^{y_j} = 1$. Then we continue to repeat the above operation on the remaining vector values until the sum of the found descriptive degrees is greater than 0.5.

4.2. Evaluation metrics

To comprehensively evaluate the performance of the LE algorithm, we compare the recovered label distribution with the ground-truth one using the five evaluation metrics that calculate the average distance or similarity between them. The five metrics are Chebyshev distance (Cheb), Clark distance (Clark), Canberra metric (Canber), Cosine coefficient (Cosine) and Intersection similarity (Intersec). The former three (i.e., Cheb, Clark, and Canber) are distance measures, and the smaller the value, the better the performance of the LE algorithm. The last two (i.e., Cosine and Intersec) are similarity measures, and the larger the value, the better the performance of the LE algorithm. Table 3 lists the formulas for calculating these metrics. In Table 3, $\mathbf{d}$ denotes the predicted label distribution and $\hat{\mathbf{d}}$ denotes the ground-truth label distribution.

4.3. Influence of parameters

Our proposed method involves a trade-off hyperparameter α. In this section, we conduct experiments on different datasets with five evaluation metrics. The influence of α is analyzed by choosing a value from $\{0.1, 0.2, \ldots, 1\}$. Fig. 6 shows the influence of α on five metrics for six datasets. The abscissa indicates the value of α, which ranges from 0 to 1, and the ordinate represents the outcome of evaluation metric. Although only the cases of six datasets are illustrated here, the same observations can be obtained in other datasets.

Figs. 6(a) – 6(c) show that the experimental results on Chebyshev, Clark, and Canberra are first inversely proportional to the value of α, and then proportional to the value of α. As α increases, the experimental results decrease numerically and the performance of LE improves. When α exceeds a certain value, the experimental results increase numerically instead, and the LE performance declines. Fig. 6(d) and 6(e) show that the experimental results on Cosine and Intersection are proportional to the value of α. As α increases, the experimental results increase numerically and the performance of LE improves. When α exceeds a certain value, the experimental results decrease numerically instead, and the LE performance declines. We can see that the performance of LE can reach its best when α is set to a certain value. As a result, the value of α in this paper is chosen among $\{0.3, 0.5, 0.7, 0.8, 0.9\}$.

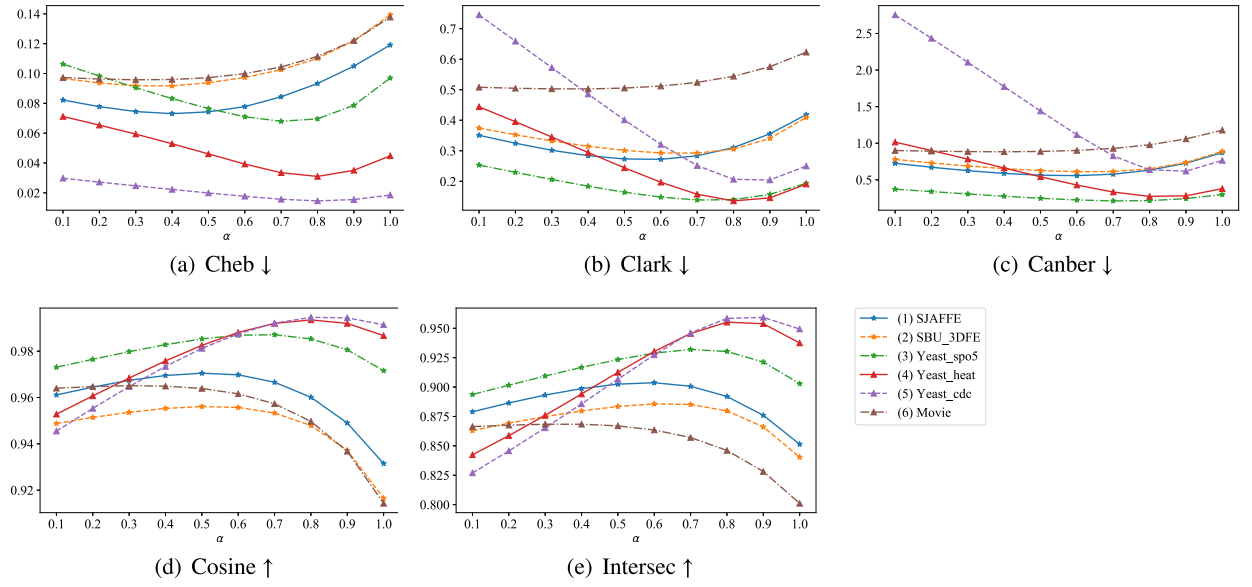

Fig. 6. The influence of α on five evaluation metrics for the six datasets. (a) Cheb ↓, (b) Clark ↓, and (c) Canber ↓ are charts of distance measures, and the smaller the value, the better the performance of LE. (d) Cosine ↑ and (e) Intersec ↑ are charts of similarity measures, and the larger the value, the better the performance of LE.

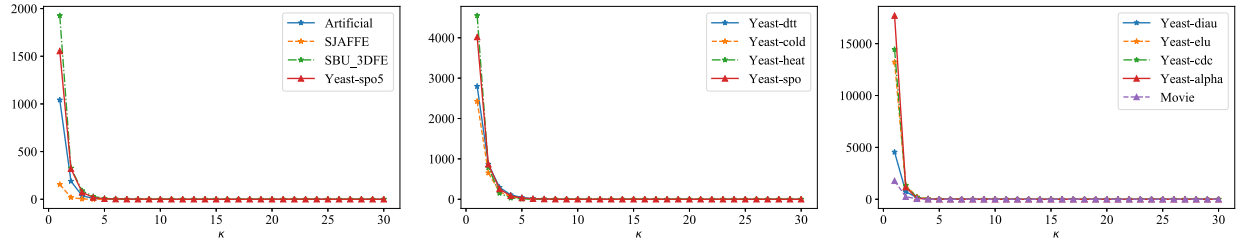

Fig. 7. The influence of the number of iterations κ on the difference in label information before and after each iteration for thirteen datasets.

4.4. Convergence of iterative propagation

Our proposed method employs the iterative label propagation technique to obtain the final label distribution. We experimentally demonstrate the convergence of our method on all datasets.

Fig. 7 shows the influence of the iteration number on thirteen datasets. Although only the cases of thirteen datasets are illustrated here, the same observations can be obtained in other datasets. The abscissa indicates the number of iterations, which ranges from 0 to 30, and the ordinate represents the difference in label information before and after each iteration. As shown in Fig. 7, the experimental results are inversely proportional to the value of the number of iterations. As the number of iterations increases, the experimental results decrease numerically. We can see that our method tends to converge after about five iterations on all datasets. As a result, we consider our method to be convergent.

4.5. Experimental setting

The proposed AILE algorithm is compared with six advanced LE algorithms, including FCM [5], KM [13], ML^2 [10], RELIAB-LP [27], GLLE [22], PLEML [33].

The parameter settings of the comparison algorithms are consistent with the original text of the corresponding literature. Specifically, the parameter β in FCM is set to 2. The kernel function in KM is a Gaussian kernel. The number of neighbors K in ML^2 is set to $c+1$. The parameter α in RELIAB-LP is set to 0.5. The parameter λ in GLLE is chosen among $\{10^{-2}, 10^{-1}, \ldots, 10^{2}\}$ and the number of neighbors K is set to $c+1$. The kernel function in GLLE is a Gaussian kernel. The parameters λ_1 and λ_2 in PLEML are chosen among $\{2^{-4}, 2^{-3}, \ldots, 2^{8}\}$, and $\gamma = 0.1$, $C = 0.1$. For AILE, the parameter α is chosen among $\{0.3, 0.5, 0.7, 0.8, 0.9\}$. (See Table 4.)

4.6. Results

In order to better discuss the performance of AILE, we calculate the quantitative results of the seven LE algorithms on five evaluation metrics. Tables 5–9 show the experimental results of Cheb ↓, Canber ↓, Clark ↓, Cosine ↑ and Intersec ↑, respectively. The

Table 4
The parameter settings.

Algorithms	Parameter settings (are consistent with the original text)
RELIAB-LP	$\alpha = 0.5$
PLEML	$\lambda_1, \lambda_2 = \{2^{-4}, 2^{-3}, \ldots, 2^{8}\}$, $\gamma = 0.1$, $C = 0.1$
GLLE	$\lambda = \{10^{-2}, 10^{-1}, \ldots, 10^{2}\}$, $K = c + 1$, kernel = rbf
FCM	$\beta = 2$
ML^2	$K = c + 1$
KM	kernel = rbf
AILE	$\alpha = \{0.3, 0.5, 0.7, 0.8, 0.9\}$

Table 5
Recovery results (value (rank)) measured by Cheb ↓.

Datasets	KM	ML^2	FCM	GLLE	PLEML	RELIAB-LP	AILE
Articifial	0.2602(7)	0.2271(6)	0.1885(5)	0.1083(3)	0.0842(2)	0.1306(4)	**0.0755(1)**
SJAFFE	0.2170(7)	0.2120(6)	0.1341(5)	0.0906(2)	0.0972(3)	0.1071(4)	**0.0743(1)**
SBU_3DEF	0.2368(7)	0.2341(6)	0.1352(5)	0.1315(4)	0.1209(2)	0.1228(3)	**0.0938(1)**
Yeas-spo5	0.2764(7)	0.2746(6)	0.1639(5)	0.1009(3)	0.0921(2)	0.1146(4)	**0.0680(1)**
Yeas-dtt	0.2490(7)	0.2448(6)	0.0942(4)	0.0407(3)	0.0373(2)	0.1286(5)	**0.0278(1)**
Yeast-cold	0.2470(7)	0.2432(6)	0.1403(5)	0.0583(3)	0.0540(2)	0.1371(4)	**0.0465(1)**
Yeast-heat	0.1705(7)	0.1654(6)	0.1566(5)	0.0449(3)	0.0435(2)	0.0863(4)	**0.0310(1)**
Yeast-spo	0.1747(7)	0.1715(6)	0.1252(5)	0.0590(2)	0.0603(3)	0.0902(4)	**0.0423(1)**
Yeast-diau	0.1536(7)	0.1483(6)	0.0865(4)	0.0503(3)	**0.0416(1)**	0.0989(5)	0.0454(2)
Yeast-elu	0.0758(7)	0.0721(6)	0.0531(5)	0.0191(3)	0.0165(2)	0.0437(4)	**0.0159(1)**
Yeast-cdc	0.0758(7)	0.0713(6)	0.0309(4)	0.0185(3)	0.0166(2)	0.0415(5)	**0.0155(1)**
Yeast-alpha	0.0616(7)	0.0568(6)	0.0353(4)	0.0168(3)	**0.0137(1)**	0.0400(5)	0.0147(2)
Movie	0.2346(7)	0.1647(4)	0.2304(6)	0.1224(2)	0.1681(5)	0.1616(3)	**0.0958(1)**
Avg.Rank	7	5.85	4.77	2.84	2.23	4.15	1.15

Table 6
Recovery results (value (rank)) measured by Clark ↓.

Datasets	KM	ML^2	FCM	GLLE	PLEML	RELIAB-LP	AILE
Articifial	1.2516(7)	1.0419(6)	0.5613(5)	0.4526(3)	0.1795(2)	0.4872(4)	**0.1638(1)**
SJAFFE	1.8740(7)	1.8444(6)	0.5121(5)	0.3081(2)	0.4312(3)	0.5050(4)	**0.2731(1)**
SBU_3DEF	1.9062(7)	1.8761(6)	0.4246(4)	0.3933(3)	0.3640(2)	0.5810(5)	**0.3012(1)**
Yeas-spo5	1.0590(7)	1.0469(6)	0.3559(5)	0.1991(3)	0.1855(2)	0.2741(4)	**0.1388(1)**
Yeas-dtt	1.4763(7)	1.4603(6)	0.3053(5)	0.1112(3)	0.1014(2)	0.1286(4)	**0.0755(1)**
Yeast-cold	1.4714(7)	1.4546(6)	0.4092(4)	0.1552(3)	0.1465(2)	0.5011(5)	**0.1255(1)**
Yeast-heat	1.8021(7)	1.7820(6)	0.5097(5)	0.1949(3)	0.1880(2)	0.2864(4)	**0.1358(1)**
Yeast-spo	1.8110(7)	1.7882(6)	0.4429(4)	0.2543(2)	0.2559(3)	0.5585(5)	**0.1792(1)**
Yeast-diau	1.8856(7)	1.8636(6)	0.6660(4)	0.2816(3)	**0.2224(1)**	0.7879(5)	0.2434(2)
Yeast-elu	2.7673(7)	2.7377(6)	0.6177(4)	0.2381(3)	0.2043(2)	0.9735(5)	**0.1963(1)**
Yeast-cdc	2.8849(7)	2.8531(6)	0.6342(4)	0.2540(3)	0.2191(2)	1.0143(5)	**0.2043(1)**
Yeast-alpha	3.1521(7)	3.1175(6)	0.8742(4)	0.2808(3)	**0.2124(1)**	1.1864(5)	0.2358(2)
Movie	1.7663(7)	1.1402(6)	0.8593(4)	0.5694(2)	0.7079(3)	0.9132(5)	**0.5028(1)**
Avg.Rank	7	6	4.38	2.77	2.08	4.62	1.15

best results from each dataset are highlighted in boldface. Cheb ↓, Canber ↓, and Clark ↓ are distance measures, and ↓ indicates the smaller the better. Cosine ↑ and Intersec ↑ are similarity measures, and ↑ indicates the bigger the better. Due to the page limitation, we only discuss the performance on the two evaluation metrics of Cheb and Cosine in Table 5 and Table 8, respectively. The results of other evaluation metrics are similar. To further analyze the superiority of AILE, the *Friedman test* [6] and the *Bonferroni-Dunn test* [3] are employed on all LE algorithms.

For *Friedman test*, the statistically significant difference between the comparison algorithms is verified by the value of F_F and the critical value. In this paper, the number of comparison algorithms is 7 ($k = 7$) and the number of datasets is 13 ($N = 13$). If the value of an evaluation metric's F_F is greater than the critical value, AILE is considered to outperform the comparison algorithms in this evaluation metric. Table 10 shows the values of the Friedman statistics F_F on each evaluation metric and the critical value at the significance level $\alpha = 0.05$.

For *Bonferroni-Dunn test*, the average rank difference between the AILE and the comparison algorithm is calibrated with the *critical difference* (CD). In this paper, the CD = 2.2352 at the significance level of 0.05. AILE is considered to have significantly different performance from the comparison algorithm if their average ranks differ by at least one CD. Fig. 8 shows the CD diagrams for all evaluation metrics, where the average ranks of different LE algorithms are marked along the axis from high to low.

Table 7
Recovery results (value (rank)) measured by Canber ↓.

Datasets	KM	ML^2	FCM	GLLE	PLEML	RELIAB-LP	AILE
Articifial	1.7795(7)	1.4136(6)	0.7974(5)	0.6172(3)	0.2707(2)	0.6687(4)	**0.2473(1)**
SJAFFE	4.0083(7)	0.7839(3)	1.0706(5)	0.6267(2)	0.8937(4)	1.0708(6)	**0.5633(1)**
SBU_3DEF	4.1209(7)	4.0593(6)	0.9051(4)	0.8409(3)	0.7801(2)	1.2463(5)	**0.6262(1)**
Yeas-spo5	1.3820(7)	1.3680(6)	0.5312(5)	0.3100(3)	0.2849(2)	0.4013(4)	**0.2122(1)**
Yeas-dtt	2.5961(7)	2.5714(6)	0.5226(4)	0.1919(3)	0.1741(2)	0.9434(5)	**0.1286(1)**
Yeast-cold	2.5674(7)	2.5415(6)	0.7010(5)	0.2679(3)	0.2528(2)	0.5297(4)	**0.2173(1)**
Yeast-heat	3.8514(7)	3.8141(6)	1.0603(4)	0.3920(3)	0.3741(2)	1.2939(5)	**0.2741(1)**
Yeast-spo	3.8548(7)	3.8116(6)	0.9043(4)	0.5269(2)	0.5284(3)	1.2341(5)	**0.3562(1)**
Yeast-diau	4.2576(7)	4.2192(6)	1.4674(4)	0.6411(3)	**0.4778(1)**	1.7490(5)	0.5343(2)
Yeast-elu	9.1129(7)	9.0292(6)	1.9762(4)	0.7171(3)	0.6014(2)	3.3835(5)	**0.5782(1)**
Yeast-cdc	9.8760(7)	9.7836(6)	2.2488(4)	0.7818(3)	0.6546(2)	3.6460(5)	**0.6187(1)**
Yeast-alpha	11.8116(7)	11.7027(6)	2.6927(4)	0.9202(3)	**0.6879(1)**	4.5494(5)	0.7376(2)
Movie	3.4442(7)	1.9337(6)	1.6642(4)	1.0446(2)	1.3263(3)	1.7204(5)	**0.8859(1)**
Avg.Rank	7	5.77	4.31	2.77	2.15	4.85	1.15

Table 8
Recovery results (value (rank)) measured by Cosine ↑.

Datasets	KM	ML^2	FCM	GLLE	PLEML	RELIAB-LP	AILE
Articifial	0.9188(7)	0.9256(6)	0.9335(5)	0.9803(3)	0.9815(2)	0.9741(4)	**0.9856(1)**
SJAFFE	0.8261(7)	0.8282(6)	0.9022(5)	0.9613(2)	0.9477(3)	0.9410(4)	**0.9704(1)**
SBU_3DEF	0.8130(7)	0.8139(6)	0.9148(5)	0.9212(4)	0.9361(2)	0.9220(3)	**0.9561(1)**
Yeas-spo5	0.8816(7)	0.8828(6)	0.9193(5)	0.9698(3)	0.9736(2)	0.9686(4)	**0.9871(1)**
Yeas-dtt	0.7600(7)	0.7619(6)	0.9569(4)	0.9926(3)	0.9937(2)	0.7855(5)	**0.9965(1)**
Yeast-cold	0.7803(7)	0.7822(6)	0.9234(5)	0.9859(3)	0.9873(2)	0.9660(4)	**0.9915(1)**
Yeast-heat	0.7801(7)	0.7819(6)	0.8893(5)	0.9863(3)	0.9872(2)	0.9323(4)	**0.9935(1)**
Yeast-spo	0.7995(7)	0.8015(6)	0.9132(5)	0.9758(2)	0.9747(3)	0.9386(4)	**0.9886(1)**
Yeast-diau	0.7982(7)	0.8018(6)	0.9225(4)	0.9775(3)	**0.9853(1)**	0.9146(5)	0.9841(2)
Yeast-elu	0.7590(7)	0.7613(6)	0.9329(4)	0.9914(3)	0.9938(2)	0.9176(5)	**0.9943(1)**
Yeast-cdc	0.7543(7)	0.7570(6)	0.9410(4)	0.9912(3)	0.9930(2)	0.9158(5)	**0.9944(1)**
Yeast-alpha	0.7518(7)	0.7546(6)	0.9507(4)	0.9912(3)	**0.9945(1)**	0.9108(5)	0.9940(2)
Movie	0.8802(6)	0.9195(4)	0.7733(7)	0.9362(2)	0.8808(5)	0.9286(3)	**0.9651(1)**
Avg.Rank	6.92	5.85	4.77	2.85	2.23	4.23	1.15

Table 9
Recovery results (value (rank)) measured by Intersec ↑.

Datasets	KM	ML^2	FCM	GLLE	PLEML	RELIAB-LP	AILE
Articifial	0.7401(7)	0.7735(6)	0.8126(5)	0.8927(3)	0.9157(2)	0.8702(4)	**0.9244(1)**
SJAFFE	0.5927(7)	0.5961(6)	0.8161(5)	0.8726(2)	0.8581(3)	0.8361(4)	**0.9025(1)**
SBU_3DEF	0.5793(7)	0.5832(6)	0.8361(4)	0.8472(3)	0.8587(2)	0.8096(5)	**0.8834(1)**
Yeas-spo5	0.7236(7)	0.7254(6)	0.8361(5)	0.8991(3)	0.9079(2)	0.8855(4)	**0.9319(1)**
Yeas-dtt	0.5411(7)	0.5437(6)	0.8802(5)	0.9527(3)	0.9570(2)	0.9210(4)	**0.9684(1)**
Yeast-cold	0.5595(7)	0.5622(6)	0.8355(5)	0.9332(3)	**0.9736(1)**	0.8756(4)	0.9464(2)
Yeast-heat	0.5591(7)	0.5617(6)	0.8185(4)	0.9356(3)	0.9385(2)	0.8048(5)	**0.9552(1)**
Yeast-spo	0.5749(7)	0.5776(6)	0.8437(4)	0.9133(2)	0.9129(3)	0.8184(5)	**0.9424(1)**
Yeast-diau	0.5883(7)	0.5902(6)	0.8223(4)	0.9099(3)	**0.9330(1)**	0.7875(5)	0.9259(2)
Yeast-elu	0.5395(7)	0.5421(6)	0.8590(4)	0.9489(3)	0.9576(2)	0.7814(5)	**0.9590(1)**
Yeast-cdc	0.5327(7)	0.5534(6)	0.8600(4)	0.9484(3)	0.9569(2)	0.7791(5)	**0.9593(1)**
Yeast-alpha	0.5323(7)	0.5350(6)	0.8752(4)	0.9496(3)	**0.9620(1)**	0.7733(5)	0.9596(2)
Movie	0.6487(7)	0.7792(3)	0.6774(6)	0.8313(2)	0.7678(5)	0.7782(4)	**0.8684(1)**
Avg.Rank	7	5.77	4.54	2.77	2.15	4.54	1.23

4.6.1. Comparison with symmetric LE algorithm

RELIAB-LP recovers label distributions using a symmetric similarity matrix as well as the iterative label propagation technique, so the experimental results are discussed separately here. According to Tables 5–9, the AILE significantly outperforms RELIAB-LP on all datasets across all evaluation metrics. The average rank for AILE across thirteen datasets is 1.15 according to the Cheb metric, whereas RELIAB-LP has a value of 4.15. For the Cosine metric, the average ranks of AILE and RELIAB-LP are 1.15 and 4.23, respectively. Compared to RELIAB-LP, AILE performs better and more consistently.

Moreover, as shown in Table 10, the values of F_F on all evaluation metrics are greater than the critical value. The results demonstrate that the null hypothesis of indistinguishable performance between comparison algorithms is rejected. Fig. 8 shows that AILE differs from RELIAB-LP by more than one *critical difference* (CD), which further illustrates that AILE is significantly different

Table 10
Results of the Friedman statistics F_F and the Critical value on various evaluation metrics ($k = 7, N = 13$).

Evaluation metrics	F_F	Critical value ($\alpha = 0.05$)
Cheb	125.6428	2.227
Clark	280.3264	
Canber	156.7933	
Cosine	106.8370	
Intersec	113.5793	

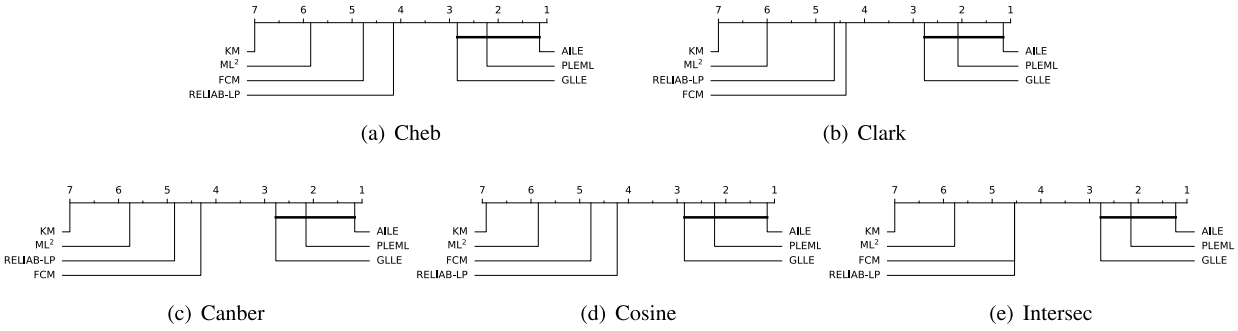

Fig. 8. CD diagrams of different LE algorithms on five evaluation metrics. Algorithms that are not connected to AILE are considered to have significantly different performances from AILE (CD = 2.2352, significance level 0.05).

from RELIAB-LP. Meanwhile, for all five evaluation metrics, AILE significantly outperforms RELIAB-LP. Actually, as introduced in Section 2, the variance of the Gaussian function used to compute the edge weights of the graph can have a significant impact on performance. Its symmetric weights are also unreasonable in the LE process. The experimental results validate that asymmetric influence can indeed improve the performance of LE.

4.6.2. Comparison with other LE algorithms

According to Tables 5–9, AILE significantly outperforms FCM, KM, ML^2, GLLE, and PLEML on most datasets across all evaluation metrics. Given that PLEML performs generally better than other comparison algorithms, we analyze the experimental results of PLEML and AILE. For the Cheb or Cosine metrics, AILE and PLEML have average ranks of 1.15 and 2.23, respectively. AILE outperforms PLEML on ten out of the thirteen datasets, with marginally worse performance only on the Yeast-diau and Yeast-alpha datasets.

In terms of statistics, AILE ranks first in 83.1% of cases and ranks second in 16.9% of cases across all datasets and evaluation metrics. Compared with existing graph-based LE algorithms (i.e., ML^2, GLLE, and RELIAB-LP), AILE ranks first in 100% of cases. Moreover, Table 10 demonstrates that AILE has a significant performance advantage over all comparison algorithms. Fig. 8 shows that AILE differs from KM, ML^2 and FCM by more than one *critical difference* (CD), which illustrates that AILE is significantly different from KM, ML^2 and FCM. Meanwhile, for all five evaluation metrics, AILE outperforms all other comparison algorithms. The above experimental results substantiate the superiority of AILE.

5. Conclusion and future works

This paper proposes a novel LE method AILE, which exploits the asymmetric influence between instances. The mass diffusion algorithm is first introduced to the LE problem to compute the influence between instances. The extensive experimental results on one artificial dataset and twelve real-world datasets demonstrate the effectiveness of the proposed AILE. However, a limitation of our method is that the feature information of instances is not effectively explored. A better strategy that can fully exploit the feature and label information of instances may further improve the performance of our algorithm.

There are still some topics that require further investigation.

1) Other physical models to portray this asymmetric influence. Asymmetric relationships in the physical world can be explained by models like acceleration [20] and heat spreading [31]. It is possible to migrate these models to describe the relationship between the instances.
2) Relationships between instances within and between clusters. The introduction of clustering algorithms separates the instances into various clusters. The instances within the cluster have similarities, which can be described by symmetric relations. The instances between clusters differ greatly, necessitating the use of asymmetric relations.
3) Extreme multi-label learning (XML). XML [21,29] presents a more challenging problem because of its extremely high-dimensional feature and label spaces, as well as its sparse and missing labels. We need to design new learning models to exploit the asymmetric influence between instances.

CRediT authorship contribution statement

Heng-Ru Zhang: Formal analysis, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing, Data curation, Writing – original draft, Conceptualization, Methodology. **Peng-Cheng Li:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Yuan-Yuan Xu:** Formal analysis, Project administration, Resources, Supervision, Visualization, Writing – original draft. **Fan Min:** Conceptualization, Formal analysis, Funding acquisition, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Heng-Ru Zhang reports financial support was provided by National Natural Science Foundation of China (61902328). Heng-Ru Zhang reports financial support was provided by Applied Basic Research Project of Science and Technology Bureau of Nanchong City (SXHZ040, SXHZ051). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (61902328), and the Applied Basic Research Project of Science and Technology Bureau of Nanchong City (SXHZ040, SXHZ051).

References

[1] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C.A. Raffel, Mixmatch: a holistic approach to semi-supervised learning, Adv. Neural Inf. Process. Syst. 32 (2019).
[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.U. Hwang, Complex networks: structure and dynamics, Phys. Rep. 424 (2006) 175–308.
[3] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[4] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. 95 (1998) 14863–14868.
[5] N. El Gayar, F. Schwenker, G. Palm, A study of the robustness of knn classifiers trained using soft labels, in: ANNPR, 2006, pp. 67–80.
[6] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92.
[7] B.B. Gao, C. Xing, C.W. Xie, J. Wu, X. Geng, Deep label distribution learning with label ambiguity, IEEE Trans. Image Process. 26 (2017) 2825–2838.
[8] B.B. Gao, H.Y. Zhou, J. Wu, X. Geng, Age estimation using expectation of label distribution learning, in: IJCAI, 2018, pp. 712–718.
[9] X. Geng, Label distribution learning, IEEE Trans. Knowl. Data Eng. 28 (2016) 1734–1748.
[10] P. Hou, X. Geng, M.L. Zhang, Multi-label manifold learning, in: AAAI, 2016.
[11] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: CVPR, 2019, pp. 1–10.
[12] X. Jia, W. Li, J. Liu, Y. Zhang, Label distribution learning by exploiting label correlations, in: AAAI, 2018.
[13] X. Jiang, Z. Yi, J.C. Lv, Fuzzy SVM with a new fuzzy membership function, Neural Comput. Appl. 15 (2006) 268–276.
[14] R. Lambiotte, J.C. Delvenne, M. Barahona, Random walks, Markov processes and the multiscale modular organization of complex networks, IEEE Trans. Netw. Sci. Eng. 1 (2014) 76–90.
[15] D.B. Larremore, A. Clauset, A.Z. Jacobs, Efficiently inferring community structure in bipartite networks, Phys. Rev. E 90 (2014) 012805.
[16] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with Gabor wavelets, in: AFGR, 1998, pp. 200–205.
[17] P. Melin, O. Castillo, Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing: an Evolutionary Approach for Neural Networks and Fuzzy Systems, Springer Science & Business Media, 2005.
[18] F. Wang, C. Zhang, Label propagation through linear neighborhoods, IEEE Trans. Knowl. Data Eng. 20 (2008) 55–67.
[19] J. Wang, X. Geng, Label distribution learning machine, in: ICML, 2021, pp. 10749–10759.
[20] D. Williams, Mechanics (including force, mass, and acceleration), Anaesth. Int. Care Med. 18 (2017) 364–367.
[21] C. Xu, D. Tao, C. Xu, Robust extreme multi-label learning, in: SIGKDD, 2016, pp. 1275–1284.
[22] N. Xu, Y.P. Liu, X. Geng, Label enhancement for label distribution learning, IEEE Trans. Knowl. Data Eng. 33 (2019) 1632–1643.
[23] N. Xu, J. Lv, X. Geng, Partial label learning via label enhancement, in: AAAI, 2019, pp. 5557–5564.
[24] J. Yang, M. Sun, X. Sun, Learning visual sentiment distributions via augmented conditional probability neural network, in: AAAI, 2017.
[25] L. Yin, X. Wei, Y. Sun, J. Wang, M.J. Rosato, A 3d facial expression database for facial behavior research, in: AFGR, 2006, pp. 211–216.
[26] H.F. Yu, P. Jain, P. Kar, I. Dhillon, Large-scale multi-label learning with missing labels, in: ICML, 2014, pp. 593–601.
[27] M.L. Zhang, Q.W. Zhang, J.P. Fang, Y.K. Li, X. Geng, Leveraging implicit relative labeling-importance information for effective multi-label learning, IEEE Trans. Knowl. Data Eng. 33 (2019) 2057–2070.
[28] M.L. Zhang, Z.H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (2013) 1819–1837.
[29] W. Zhang, J. Yan, X. Wang, H. Zha, Deep extreme multi-label learning, in: ICMR, 2018, pp. 100–107.
[30] D. Zhou, O. Bousquet, T. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, Adv. Neural Inf. Process. Syst. 16 (2003).
[31] T. Zhou, Z. Kuscsik, J.G. Liu, M. Medo, J.R. Wakeling, Y.C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. 107 (2010) 4511–4515.
[32] T. Zhou, J. Ren, M. Medo, Y.C. Zhang, Bipartite network projection and personal recommendation, Phys. Rev. E 76 (2007) 046115.
[33] W. Zhu, X. Jia, W. Li, Privileged label enhancement with multi-label learning, in: IJCAI, 2021, pp. 2376–2382.
[34] X. Zhu, Semi-Supervised Learning with Graphs, Carnegie Mellon University, 2005.
[35] X. Zhu, A.B. Goldberg, Introduction to semi-supervised learning, Synth. Lect. Artif. Intell. Mach. Learn. 3 (2009) 1–130.
[36] Y. Zhu, J.T. Kwok, Z.H. Zhou, Multi-label learning with global and local label correlation, IEEE Trans. Knowl. Data Eng. 30 (2017) 1081–1094.