Knowledge-Based Systems 178 (2019) 1–10

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Multi-label learning with kernel extreme learning machine autoencoder

Yusheng Cheng [a,b,*], Dawei Zhao [a], Yibin Wang [a,b], Gensheng Pei [a]

[a] School of Computer and Information, Anqing Normal University, Anhui Anqing 246011, China
[b] The University Key Laboratory of Intelligent Perception and Computing of Anhui Province, Anqing 246011, China

ARTICLE INFO

Article history:
Received 15 September 2018
Received in revised form 1 April 2019
Accepted 4 April 2019
Available online 8 May 2019

Keywords:
Multi-label learning
Extreme learning machine
Autoencoder
Non-equilibrium labels completion
Information entropy
Labels correlations

ABSTRACT

In multi-label learning, in order to improve the accuracy of classification, many scholars have considered the relationship between features and features, features and labels or labels and labels, but how to combine the correlation among them is rarely studied. Based on this, this paper proposes a multi-label learning algorithm with kernel extreme learning machine autoencoder. Firstly, the label space is reconstructed by using the non-equilibrium labels completion method in the label space. Then, the non-equilibrium labels space information is added to the input node of the kernel extreme learning machine autoencoder network, and the input features are output as the target. Finally, the kernel extreme learning machine is used for classification. Our method implements the information fusion between features and features, between labels and features, and between labels and labels. Compared with the traditional autoencoder network, the extreme learning machine autoencoder has no iterative process, which reduces the network training time and improves the classification accuracy. The experimental results of the proposed algorithm in the opening benchmark multi-label data sets show that the KELM-AE algorithm has some advantages over other comparative multi-label learning algorithms and the statistical hypothesis testing and stability analysis further illustrate the effectiveness of the proposed algorithm.

1. Introduction

In multi-label learning [1], a single instance is associated with multi-label, and a valid model is trained through the training set to effectively predict the set of labels belonging to unknown instances. Many scholars have proposed a lot of multi-label learning algorithms. For example, the BR [2] (binary relevance) algorithm, the LP (label power set) algorithm, etc., the methods solve the multi-label problem by increasing the number of classifiers or the types of the label but affect the efficiency of the classifier to some extent. Back-propagation for multi-label learning BP-MLL [3] (rank-propagation for multi-label learning) introduces the ranking loss factor and the MLKNN [4] algorithm used the maximized a posteriori probability (MAP) to solve the multi-label learning prediction problem, the performance of them increases the complexity of its calculation although the classification was improved.

The relationship between labels in the real world are often not independent of each other, and there is a certain correlation between them. For the correlation between labels, many scholars have proposed the correlation algorithm and achieved good results. For example, the RankSVM [5] uses the maximum interval criteria strategy to adapt to multi-label learning. During the modeling process, the SVM classifier is constructed for the sorting loss between relevant labels and irrelevant labels corresponding to samples. But the time consumption is relatively large because a large number of variables need to calculate.

At the same time, as an effective measure of uncertainty, information entropy [6] and other relevant information theory have been widely used in the research of label correlation. Based on this theory, Zhang [7] et al. proposed a multi-label classification algorithm based on correlation information entropy. On the basis of the RAkEL(random k-label sets) algorithm, it used the relevant information entropy to measure the correlation between the labels to improve the performance of multi-label classification. Lee [8] et al. proposed a new multi-label learning method based on the CC(classifier chains) algorithm. Using the directed acyclic graph to model the correlation of labels, and using conditional entropy to design a multi-label learning method to maximize the correlation between labels, these methods achieved good results. It has achieved better results by using the information entropy theory to measure the correlation between labels. However, these methods basically only calculated the mutual information

* Corresponding author at: School of Computer and Information, Anqing Normal University, Anhui Anqing 246011, China.
E-mail address: chengyshaq@163.com (Y. Cheng).

https://doi.org/10.1016/j.knosys.2019.04.002
0950-7051/

between two labeled labels and then measure the interaction between labeled labels by mutual information. It can be seen that using this kind of basic label confidence matrix to measure the relevant information between labels only consider the mutual influence between labeled objects but ignoring the influence of the labeled of unlabeled labels on the quality of label sets and the impact of known labels on unlabeled labels.

Besides, a method of reconstructing information of a feature space using label information was also widely used. The LIFT [9] method first used the K-means clustering algorithm to cluster the positive and negative examples of each labels and calculated the distance between the sample and the cluster center to generate the exclusive features of each label, thus obtained a new training set. Based on this training set, binary relevance classification learning was performed for each labels. However, the LIFT method did not consider labels correlation, some scholars have proposed a joint learning of label-specific features and label correlations. Zhang [10] et al. account the correlation among labels by constructing additional features. Huang et al. [11] proposed to learn the label-specific features and shared features by using pairwise label correlations to distinguish each category labels, and then constructed a multi-label classifier on the low-dimensional data representations composed of these learned features. Zhang [12] et al. proposed a multi-label learning with feature-induced labeling information enrichment(MLFE), which changed the structural information in the feature space by enriching the label information of the multi-label samples. Based on tailored multiple regression method, the classification effect of the algorithm can be improved with rich labels information from the training samples. In the multi-label learning data set, the number of labels in the label data set is generally large, but the average number of labels and the labels density are not high for each object. This phenomenon is also consistent with common sense: the labeled labels of an object should not be larger than the unlabeled labels, otherwise, the multi-label of the object will lose its meaning. It is undeniable that there may be a lot of valuable information in the unlabeled labels, just as it is about the abnormal research. For the purpose, a method of non-equilibrium labels completion is introduced to describe the relationship between labels.

It is not difficult to find that the classification performance of the algorithm is improved by the construction of the features with the labels and the labels-to-labels relationship. In recent years, a large number of unsupervised learning methods have been applied in the field of data mining. Based on graph basis system (GBS) [13] multi-view clustering method was proposed to tackle the limitations of the existing graph-based. Further, a clustering method based on the local linear embedding(LLE) and Laplace feature mapping(LEE) method(L3E-M2VC) [14] were proposed to deal with multi-task multi-view problems. The Autoencoder neural network [15] was an unsupervised learning paradigm that automatically learns features from unlabeled data. Autoencoder has been widely used in image classification [16]. The autoencoder neural network was a class of models which aim to map the input to a latent space and map it back to the original space, with low reconstruction error as its objective. But the current training process of autoencoder neural network involved lots of iterations. The ELM algorithm, which was proposed by Huang [17,18] as a simple and efficient single hidden layer feed-forward neural network learning algorithm, does not need any iterative adjustment to the network weight and bias in the training process. Compared with the traditional neural network algorithm, its training speed is fast. In this regard, L.L.C. Kasun [19] et al. put forward an ELM-AE classification algorithm which was a novel method of neural network. ELM-AE can reproduce the input signal as well as autoencoder. Based on this, this paper proposes a kernel extreme learning machine autoencoder for multi-label learning algorithm(KELM-AE). We use a two-layer KELM module as the base model and the first KELM as a autoencoder block and adds labels node information in the input layer, and the output layer outputs features that contain feature and labels relationships. The second KELM model serves as classification module, is used during the classification process while the labels space uses the non-equilibrium labels completion matrix algorithm. The experimental and statistical hypothesis testing of the algorithm on multiple published multi-label data sets proves that the algorithm has a certain validity, and it is also confirmed that the combination of feature space reconstruction and label correlation can improve the rationality of algorithm performance.

The rest of the paper is organized as follows. Section 2 gives some basic notions related to multi-label learning and the rough entropy. Section 3 introduces the modeling of the non-equilibrium matrix. Section 4 introduces the modeling of KELM. Our proposed method for the multi-label classification of KELM-AE is proposed in Section 5. In Section 6, experimental results of the KELM-AE in opening multi-label data sets shows that our algorithm is effective and statistical hypothesis tests further prove our method in Section 6 too. In the last section, we sum up what has been discussed and point out further research.

2. The multi-label learning and rough entropy

2.1. The multi-label learning and traditional entropy

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}} \in \mathbf{R}^{N\times d}$ be the d-dimensional input feature space, where N denotes the number of samples. $\mathbf{x}_i \in \mathbf{R}^d$ denotes the feature vector corresponding to the ith sample; $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^{\mathrm{T}} \in \mathbf{R}^{N\times k}$ denotes the label matrix corresponding to the sample, where k denotes the number of labels in the data; $\mathbf{y}_i = \{1, -1\}^k$ denotes the binary label indicator vector corresponding to the ith sample. Therefore, the multi-label training data set containing N samples is:

$$\mathbf{D} = \{\mathbf{x}_i, \mathbf{Y}_i \,|1 \le i \le N\} \subset \mathbf{R}^d \times \{+1, -1\}^k \tag{1}$$

Definition 1 (*[6,7,20]*)**.** Suppose the set $A = \{a_1, \ldots, a_m\}$, and $p(a_i)$ denotes the prior probability of the element a_i :

$$H(A) = -\sum_{i=1}^{n} p(a_i)\log_2 p(a_i) \tag{2}$$

Then $H(A)$ is the information entropy of the set A, and the larger value of it, the more uncertainty of the set.

Definition 2 (*[6,7,20]*)**.** Suppose the set $A = \{a_1, \ldots, a_n\}$ and the set $B = \{b_1, \ldots, b_n\}$, then the conditional entropy of the set B under the given constraints of the set A is:

$$H(B\,|A\,) = -\sum_{i=1}^{m}\sum_{j=1}^{n} \mathrm{H}(b_j\,|a_i\,) \tag{3}$$

where $H(b_j\,|a_i\,)$, the conditional information, is employed to describe the uncertainty of the elementb_j with the appearing element a_i. The larger the value, the more uncertainty between a_i and b_j, and vice versa:

$$H(b_j\,|a_i\,) = -p(a_i b_j)\log_2 p(b_j\,|a_i\,) \tag{4}$$

The conditional entropy is thus employed to describe the uncertainty of the set B with the appearing set A.

Meanwhile, the traditional entropy is often used in the multi-label learning algorithms and it has a high complexity of computation because it has no nature of complement. Therefore, a new definition about the rough entropy will be introduced in this paper.

2.2. A new definition about the rough entropy

An information system is usually denoted as triplet $S = (U, A, f)$, which is called a decision table, where U is the universe which consists of a finite set of objects, A is the set of attributes. With every attribute $a \in A$, set of its values V_a is associated. Each attribute a determines an information function $f : U \to V_a$ such that for any $a \in A$ and $x \in U$, $f(x) \in V_a$. Each non-empty subset $P \subseteq A$ determines an indiscernible relation:

$$R_p = \{(x, y) : \forall \alpha \in P, f_a(x) = f_a(y), x, y \in U\} \tag{5}$$

R_p is called a equivalence relation and partitions U into a family of a disjoint subsets U/R_p called a quotient set of U:

$$U/R_p = \{X_1, \ldots, X_n\} \tag{6}$$

In the traditional entropy definition, $\log_2 \frac{1}{p(X_i)}$ is used to measure the information quantity of the equivalence classes X_i. Similarly, we construct the definition of information quantity expressed by equivalence classes based on rough set theory as follows:

$$I(X_i) = 1 - \frac{|X_i|}{|U|} \tag{7}$$

$|\cdot|$ represents the cardinality of the set element and $0 \le I(X_i) \le 1 - \frac{1}{|U|}$.

Definition 3 ([20]). An information system $S = (U, A, f), p \subseteq A, U/R_p = \{X_1, \ldots, X_n\}$, the information entropy of attributes P is defined as follows:

$$E(p) = E(X) = \sum_{i=1}^{n} \frac{|X_i|}{|U|} I(X_i) = \sum_{i=1}^{n} \frac{|X_i|}{|U|}\left(1 - \frac{|X_i|}{|U|}\right) = \sum_{i=1}^{n} \frac{|X_i|}{|U|} \frac{|X_i|^{\mathrm{C}}}{|U|} \tag{8}$$

In which C represents the complement. It is easy for $E(X)$ to be a rough entropy and $0 \le E(X) \le 1 - \frac{1}{|U|}$.

3. The modeling of the non-equilibrium label completion matrix

The number of unlabeled items of a sample in the real world is much larger than that of annotated ones, as seen in the example that a picture with known labels including *greenmountains* and *clearwater* is more probable to contain unlabeled *forests*, rather than unlabeled *deserts* or *sea*. We have found in many cases that researchers calculate the conditional information between annotated and unlabeled elements in each label set of the sample by applying Eq. (4), to obtain the basic label confidence matrix. Suppose the matrix of training samples $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^{\mathrm{T}} \in \mathrm{R}^{N \times k}$ and $\mathbf{y}_i = \{1, -1\}^k$. According to Eq. (4) of traditional entropy, we have:

$$a_{ij} = \frac{1}{H(\bar{l}_j \,|\, l_i)}$$

where l_i and $\bar{l}_i$ denote that the value of $\mathbf{y}_i$ is 1, and -1; $i = 1, \ldots, k, j = 1, \ldots, k$ and $i \neq j$.

The information entropy is used to measure the relationship between the unknown labels and the known labels weight, but the traditional information entropy does not have the complementary property, and the calculation method is more complicated. The relationship between unknown labels and known labels is complementary, therefore the measurement of new rough information entropy is undoubtedly more accurate. According to Eq. (7) of new rough information entropy, the new basic label confidence matrix can be redefined as follows:

$$newa_{ij} = \frac{1}{I(l_j^{\mathrm{C}} \,|\, l_i)}, newb_{ij} = \frac{1}{I(l_j \,|\, l_i^{\mathrm{C}})}$$

Therefore, $newa_{ij}$ the new basic label confidence matrix focuses on the confidence of known labels to unknown ones, while $newb_{ij}$ the confidence of unknown labels to known ones, and it directly affects the quality of label sets. Since most multi-label data sets are currently artificially labeled, what can be confirmed is an unknown sample may directly affect the quality of multi-label data sets. The paper therefore introduces α, the non-equilibrium parameter and proposes the algorithm of the non-equilibrium labels confidence matrix (NeLCM) based on weighted calculation of decreasing the basic label confidence matrix (BCLM) of $newa_{ij}$ and increasing that of $newb_{ij}$:

$$Conf_{ij} = -\alpha \times newa_{ij} + (1 - \alpha) \times newb_{ij} \tag{9}$$

This construction method is a high-order strategy. We suggest the range of the non-equilibrium parameter $0 \le \alpha \le 0.5$.

Inspired by the idea of labels propagation dependency [21], the non-equilibrium label completion matrix is defined as follows:

$$\hat{\mathbf{Y}} = \mathbf{Conf} \times \mathbf{Y} \tag{10}$$

Introduced non-equilibrium parameters, the algorithm of non-equilibrium label confidence matrix is calculated as follows:

Algorithm 1 **N**on-**e**quilibrium **L**abels **C**onfidence **M**atrix (**NeLCM**)

Input: $\mathbf{Y}$, the matrix of training samples, and α, the unbalanced parameter
Output: $\hat{\mathbf{Y}}$, **NeLCM**

```
Y = {Y_i | i = 1, ..., k}. The label set of the training set.
for each l_i, l_j do
    if i ≠ j then
        Calculate newa_ij and newb_ij by employing Eq. (7).
    else i = j
        newa_ij = newb_ij = 0
    end if
    Normalize the matrix a, b by row and obtain the corresponding matrix a, b.
    if i = j then
        newa_ij = newb_ij = 1/*Set the diagonal element as 1.;
    end if
    Obtain the confidence matrix Conf by employing Eq. (9).
end for
non-equilibrium labels confidence matrix Ŷ = Conf × Y.
return Ŷ
```

4. The modeling of kernel extreme learning machine with autoencoder

4.1. The theory of kernel extreme learning machine

The ELM algorithm is an effective single-hidden layer feed forward neural networks learning algorithm. The learning parameters of the hidden layer in the ELM algorithm network structure are randomly selected that only necessary to set the number of hidden layer network neurons. Finally, the output weight of the hidden layer is obtained by the least squares method, and no iteration is required for the network weight and offset in the process. Therefore, compared with the traditional neural network algorithm, the ELM algorithm has the advantages of fast training speed and strong generalization ability.

Before analyzing the two phases of ELM, the following formal definitions need to be made: For N distinct sample $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$, $i = 1, \ldots, N$, $\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{in}]^{\mathrm{T}}$, $\mathbf{Y} = [\mathbf{y}_{i1}, \mathbf{y}_{i2}, \ldots, \mathbf{y}_{im}]^{\mathrm{T}}$. For a single hidden layer neural network with L hidden nodes can be expressed as Eq. (11):

$$f_L(\mathbf{X}_j) = \sum_{i=1}^{L} \beta_i g_i(\mathbf{X}_j) = \sum_{i=1}^{L} \beta_i g(\omega_i \cdot X_j + \mathbf{b}_i) = \mathbf{o}_j \tag{11}$$

In Eq. (11), $\beta = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^{\mathrm{T}}$ is the output weight, $g(x)$ is the activation function, $\omega_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{im}]^{\mathrm{T}}$ $\mathbf{b}_i$ is the input weight, $\mathbf{b}_i$ is expressed as the offset of the ith hidden neuron. For classification problems, you can use the sigmoid function to limit the range of output values to achieve classification.

The above is the first stage of the ELM, namely random feature mapping. For the linear parameter solution of the second stage, the weights β of the hidden layer and the output layer are solved by minimizing the approximation error of the square error. Hence, it can be expressed as follows:

$$\min_{\beta} \|\mathbf{H}\beta - \mathbf{Y}\|^2 \tag{12}$$

where $\mathbf{H} = \begin{bmatrix} h(\mathbf{x}_1) \\ \vdots \\ h(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & \cdots & h_L(\mathbf{x}_N) \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{y}_N^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{11} & \cdots & \mathbf{y}_{1m} \\ \vdots & \ddots & \vdots \\ \mathbf{y}_{N1} & \cdots & \mathbf{y}_{Nm} \end{bmatrix}$

The above N equations can be written compactly as Eq. (13):

$$\begin{aligned} &\mathbf{H}\beta = \mathbf{Y} \\ &\beta = \mathbf{H}^{\dagger}\mathbf{Y} \\ &\text{s.t.}\mathbf{H}^{\dagger} = \begin{cases} (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}} \\ \mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1} \end{cases} \end{aligned} \tag{13}$$

where $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ and $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ are a nonsingular matrix and $\mathbf{H}^{\dagger}$ is a Moore–Penrose generalized inverse matrix of $\mathbf{H}$. According to the ridge regression theory, in order to improve the stability and generalization ability of the algorithm, adding the regular term C, the minimum target of the Eq. (11) can be expressed as:

$$\begin{aligned} &\min L_f = \|\beta\|^2 + C\sum_{i=1}^{N} \|\xi_i\|^2 \\ &\text{s.t.}\xi_i = \mathbf{Y}_i - f_l(\mathbf{x}_i), i = 1, 2 \cdots, N \end{aligned} \tag{14}$$

According to Karush–Kuhn–Tucker (KKT) optimal conditions, the hidden output weight β can be obtained by Eq. (15):

$$\beta = \mathbf{H}^{\mathrm{T}}(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1}\mathbf{Y} \tag{15}$$

Then the multi-labels output function can be expressed as

$$f_l(\mathbf{x}) = \mathbf{H}\beta = \mathbf{H}\mathbf{H}^{\mathrm{T}}\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^{\mathrm{T}}\right)^{-1}\mathbf{Y} \tag{16}$$

In the traditional ELM algorithm, the calculation results are easily affected by random set values. This paper introduces a kernel matrix to solve this problem.

$$\begin{aligned} &\Omega_{\mathrm{ELM}} = \mathbf{H}\mathbf{H}^{\mathrm{T}} : \Omega_{\mathrm{ELM}(i,j)} = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \\ &\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|) \end{aligned} \tag{17}$$

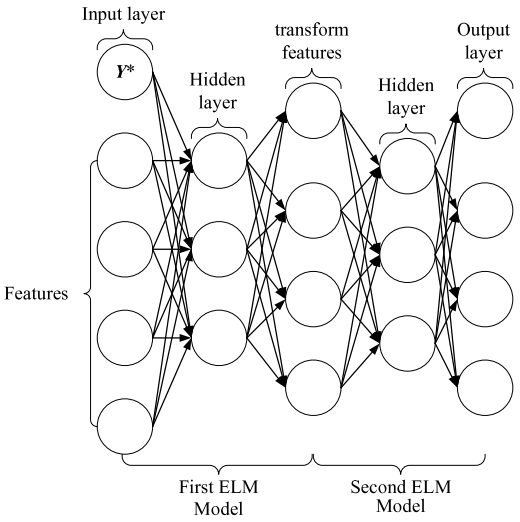


Fig. 1. KELM AutoEncoding algorithm network structure diagram.

According to Eq. (16), $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ can be rewritten as:

$$\mathbf{H}\mathbf{H}^{\mathrm{T}} = \begin{bmatrix} \mathbf{K}(\mathbf{x}, \mathbf{x}_1) \\ \mathbf{K}(\mathbf{x}, \mathbf{x}_2) \\ \vdots \\ \mathbf{K}(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}^{\mathrm{T}} \tag{18}$$

The output of KELM network $f_l(\mathbf{x})$ can be presented by Eq. (18):

$$\begin{aligned} f(\mathbf{x}) &= h(\mathbf{x})\mathbf{H}^{\mathrm{T}}(\tfrac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1}\mathbf{Y} \\ &= \begin{bmatrix} \mathbf{K}(\mathbf{x},\mathbf{x}_1) \\ \vdots \\ \mathbf{K}(\mathbf{x},\mathbf{x}_N) \end{bmatrix} (\tfrac{\mathbf{I}}{C} + \Omega_{\mathrm{ELM}})^{-1}\mathbf{Y} \end{aligned} \tag{19}$$

4.2. Kernel extreme learning machine autoencoder for multi-label

AutoEncoder is an important class of models for representation learning, and is one of the key ingredients of deep learning. An autoencoder has two basic functions: encoding and decoding. AutoEncoder can effectively extract the intrinsic link of features in the data. Learning strategy can be expressed as a minimum reconstruction error function [22]:

$$\sum_{i=1}^{d} \left\|\mathbf{x}_i - \hat{\mathbf{x}}_i\right\|^2$$

ELM is a simple three-layer network structure, namely input layer, hidden layer and output layer. Given a multi label training set $\mathbf{D} = \{(\mathbf{x}_1, \mathbf{Y}_1), \ldots (\mathbf{x}_N, \mathbf{Y}_N)\}$, where $\mathbf{x}_i \in \mathbf{X}$ is a single instance, $\mathbf{Y}_i \in \mathbf{y}$ is a set $\mathbf{x}_i$ of associated labels. When feature $\mathbf{x}_i$ is used as the input layer and the output layer of the ELM is equal to the input layer, the ELM at this time becomes the Autoencoder. The Autoencoder kernel extreme learning machine algorithm proposed in this paper is to connect two KELMs. The network structure of the algorithm is shown in Fig. 1.

The first KELM is AutoEncoder KELM, adding targets information $\mathbf{Y}^*$ to the input layer feature set. $\mathbf{Y}^* \in \mathbf{R}^{N\times 1}$ is the summation result of each labeled sample set value for the Non-equilibrium Labels Completion $\hat{\mathbf{Y}}$. This reduces the impact of excessive data size and reduced classification efficiency. The input feature $\mathbf{X}$ this time is represent as $\mathbf{X}_i = \{\mathbf{x}_1{}^{\mathrm{T}}, \ldots, \mathbf{x}_i{}^{\mathrm{T}}, \mathbf{y}^*\}$, $\mathbf{X}_i \in \mathbf{R}^{(d+1)\times N}$. Taking $\mathbf{X}_i$ as an input feature into Eqs. (12), (13) can be represent as:

$$\mathbf{x}_i = \mathbf{H}_1\beta_1 \tag{20}$$

The first KELM model can be represented as:

$$\begin{aligned} &\min L_f = \|\beta_1\|^2 + C\sum_{i=1}^{N}\|\xi_i\|^2 \\ &\text{s.t.}\xi_i = \mathbf{x}_i - f_l(\mathbf{x}), i = 1, 2, \ldots, N \end{aligned} \tag{21}$$

The output of the first KELM contains information about the feature and labels information $\hat{\mathbf{X}}_i = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_N\}$ and the transforming features as input to the second classification KELM neural network. The function $h(\hat{\mathbf{x}}_i)$ is used to map $\hat{\mathbf{x}}_i$ from the input space to the L-dimensional feature space, and the Non-equilibrium Labels Completion matrix is obtained by the algorithm 1 as a corresponding set of output labels. For a new object x to be classified, the Non-equilibrium Labels Completion matrix is added to the second KELM model to predict the labels set.

$$\begin{aligned} &\hat{\mathbf{Y}} = \mathbf{Conf} \times \mathbf{Y} = \mathbf{H}_2\beta_2 \\ &\beta_2 = \mathbf{H}_2{}^{\dagger}\hat{\mathbf{Y}} \end{aligned} \tag{22}$$

Then the output function $f_l(\mathbf{x})$ of the multi-label ELM is represented as Eq. (23):

$$f_l(\mathbf{x}) = \mathbf{H}_2\beta_2 = \begin{bmatrix} h(\hat{\mathbf{x}}_1) \\ \vdots \\ h(\hat{\mathbf{x}}_N) \end{bmatrix}_{N\times L} \begin{bmatrix} \beta_1^{\mathrm{T}} \\ \vdots \\ \beta_L^{\mathrm{T}} \end{bmatrix}_{L\times d} \tag{23}$$

Eq. (24) can be obtained from Eq. (22):

$$\begin{aligned} &\beta_2 = \mathbf{H}_2{}^{\dagger}\hat{\mathbf{Y}} \\ &\text{s.t.}\mathbf{H}_2{}^{\dagger} = \mathbf{H}_2{}^{\mathrm{T}}(\mathbf{H}_2\mathbf{H}_2{}^{\mathrm{T}})^{-1} \end{aligned} \tag{24}$$

The mathematical model of the second KELM network for multi-label classification learning can be expressed by Eq. (25):

$$\begin{aligned} &\min L_f = \|\beta_2\|^2 + C\sum_{i=1}^{N}\|\xi_i\|^2; \\ &\text{s.t.}\xi_i = \mathbf{Y} - f_l(\mathbf{x}), i = 1, 2, \ldots, N \end{aligned} \tag{25}$$

where, β_2 is output weight of hidden layer, and C is cost parameter (also called ridge regression parameter). ξ_i is the error between theoretical output $\mathbf{Y}_i$ and the actual output $f_l(\mathbf{x}_i)$. Then the Eq. (15) can be written compactly as Eq. (26):

$$\beta_2 = \mathbf{H}^{\mathrm{T}}(\frac{\mathbf{I}}{C} + \mathbf{H}_2\mathbf{H}_2{}^{\mathrm{T}})^{-1}\hat{\mathbf{Y}} \tag{26}$$

The multi-label input function added to the kernel matrix according to Eq. (19) is represented as Eq. (27):

$$\begin{aligned} f_l(\mathbf{x}) &= h(\mathbf{x})\mathbf{H}_2{}^{\mathrm{T}}(\tfrac{\mathbf{I}}{C} + \mathbf{H}_2\mathbf{H}_2{}^{\mathrm{T}})^{-1}\hat{\mathbf{Y}} \\ &= \begin{bmatrix} K(\mathbf{x},\mathbf{x}_1) \\ \vdots \\ K(\mathbf{x},\mathbf{x}_N) \end{bmatrix} (\tfrac{\mathbf{I}}{C} + \boldsymbol{\Omega}_{\mathrm{ELM}})^{-1}\hat{\mathbf{Y}} \end{aligned} \tag{27}$$

Minimize the objective function is represented as Eq. (28):

$$f_{\mathrm{SSE}} = \sum_{i}^{N}\left\|f_l(\mathbf{x}_i) - \hat{\mathbf{Y}}_i\right\|^2 \tag{28}$$

Therefore, the Multi-label learning with Kernel Extreme Learning Machine AutoEncoder algorithm is as follows:

Algorithm 2 **K**ernel **E**xtreme **L**earning **M**achine **A**uto**E**ncoder(**KELM-AE**)

Input: training set data $\mathbf{D}$, testing set $\mathbf{D}^* = \{\mathbf{x}_i, \mathbf{Y}_i \,|\, 1 \le i \le M\} \subset \mathbf{R}^d \times \{+1, -1\}^k$.
Output: $\mathbf{Y}^*$, the prediction label.

```
for training data set D do
    Compute Non-equilibrium Labels Completion matrix NeLCM;
    Compute the first KELM kernel matrix Ω_ELM ;
    Compute the first KELM network structure (I/C + Ω_ELM)^{-1}X;
    Compute the output X̂_i of the first KELM by Eqs. (19)–(21);
    Compute the second KELM kernel matrix Ω_ELM;
    Compute the second KELM network structure (I/C + Ω_ELM)^{-1} Ŷ;
    Compute the output weight β_2 of the second KELM by Eqs. (24)–(26);
end for
for testing set D* do
    Compute the outputs by Eq. (23);
    Predict labels Outputs(k, i)
      = { 1   f_l(x̂_i) ≥ 0 ;   i = 1, ···, M;   k = 1, ···, l;
        {-1   f_l(x̂_i) < 0
end for
return Y*
```

5. The KELM-AE experiment and results

5.1. The description of the experimental data sets

In order to illustrate the effectiveness of the algorithm KELM-AE, we choose 14 sets of data sets such as *Emotions*, *Natural scene* and *Yeast* 3 *Mulan datasets* and 11 sets of *Yahoo Web Pages*. The *Mulan datasets* is from http://mulan.sourceforge.net/datasets-mlc.html. The *Yahoo Web Pages datasets* is from http://www.kecl.ntt.co.jp/as/members/ueda/yahoo.tar. The specific description is shown in Table 1.

5.2. The experimental environment and evaluation indicators

The experiment is conducted on a computer equipped with Windows 7 Operation System, Intel Core(TM) i5-2380p, and 3.10 GHz CPU, and in Matlab 2016a for the operation of experimental codes. We choose 5 commonly-applied evaluation criterions, namely, Average Precision, Coverage, Hamming Loss, One-Error, and Ranking Loss [23] to evaluate the MLLA performance. The criterions are abbreviated as AP↑, CV↓, HL↓, OE↓, and RL↓ for convenience, where ↑ indicates the higher value, the better, and ↓ the lower, the better. Suppose $h(\cdot)$, the multi-label classifier; $f(\cdot,\cdot)$, the prediction function; $rank_f$, the ranking function; $\mathbf{D} = \{(x_i, Y_i \,|\, 1 \le i \le n)\}$, the MLD. The formal methods of these criterions are defined as follows:

(1) Average Precision (AP): Evaluating the average score of correct labels ranked in the specific label $y \in \mathbf{Y}_i$:

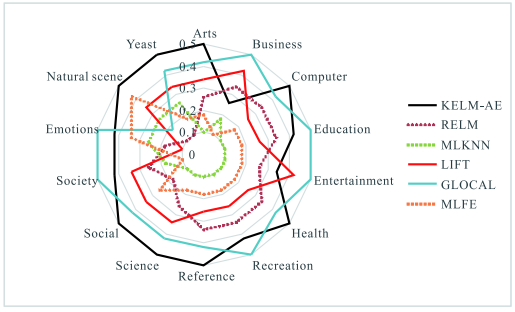
$$AP_{\mathbf{D}}(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\mathbf{Y}_i|}\sum_{y\in Y_i}\frac{\left|\{rank_f(x_i, y') \le rank_f(x_i, y), y' \in \mathbf{Y}_i\}\right|}{rank_f(x_i, y)}$$

(2) Coverage (CV): An indicator to measure the average step number for traversing all related labels of the given sample:

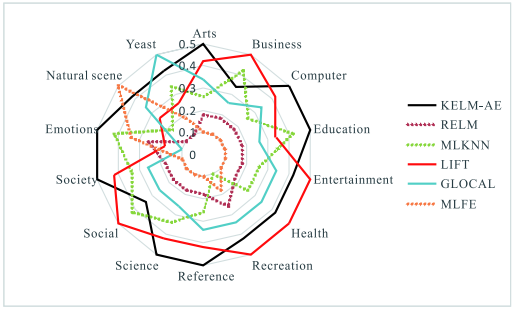
$$CV_{\mathbf{D}}(f) = \frac{1}{n}\sum_{i=1}^{n}\max_{y\in Y_i} rank_f(x_i, y) - 1$$

Table 1
Detailed descriptions of multi-label data sets.

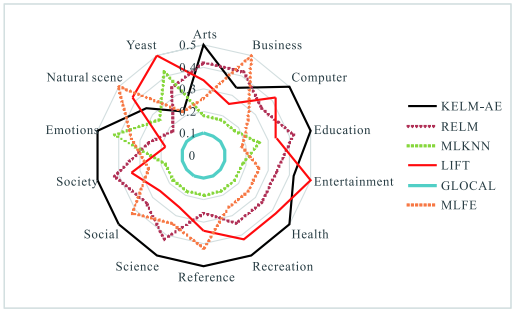
Data sets	Training sets	Test sets	No. of labels	No. of features	Average no. of labels	Label density	Fields
Arts	2000	3000	26	462	1.636	0.063	Text
Business	2000	3000	30	438	1.588	0.053	Text
Computers	2000	3000	33	681	1.508	0.046	Text
Education	2000	3000	33	550	1.461	0.044	Text
Entertainment	2000	3000	21	640	1.42	0.068	Text
Health	2000	3000	32	612	1.663	0.052	Text
Recreation	2000	3000	22	606	1.423	0.065	Text
Reference	2000	3000	33	793	1.169	0.035	Text
Science	2000	3000	40	743	1.451	0.036	Text
Social	2000	3000	39	1047	1.283	0.033	Text
Society	2000	3000	26	462	1.692	0.063	Text
Emotions	391	202	6	72	1.868	0.311	Music
Natural scene	1000	1000	5	294	1.236	0.247	Images
Yeast	1500	917	14	103	4.237	0.303	Biology



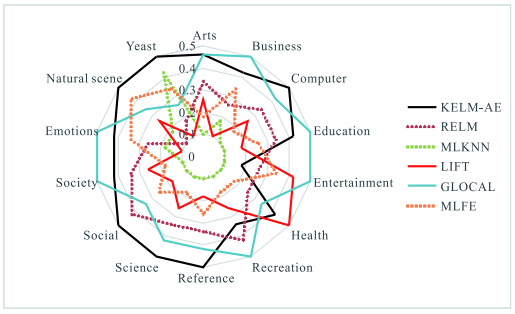
(a) AP.



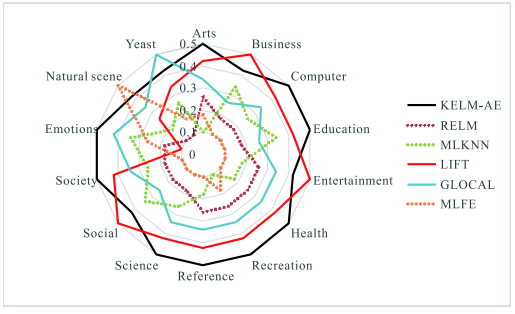
(b) CV.



(c) HL.



(d) OE.



(e) RL.

Fig. 2. The stability index values obtained on 14 benchmark multi-label data sets with different evaluation metrics.

(3) Hamming Loss (HL): An indicator to measure real labels in a single label and wrong matches of prediction labels of the given sample:

$$HL_{\mathbf{D}}(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y|}|h(x_i) \neq Y_i|$$

(4) One-Error (OE): Evaluating the occurrence number of labels when top-ranking labels are not correct:

$$OE_{\mathbf{D}}(f)\frac{1}{n}\sum_{i=1}^{n}[[\arg\max_{y\in Y} f(x_i, y)] \notin Y_i]$$

(5) Ranking Loss (RL): An indicator to evaluate the circumstances where the ranking of uncorrelated labels of a given sample is lower than that of correlated labels:

$$RL_{\mathbf{D}}(f) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|\left|\overline{Y}_i\right|} \times \left|\{(y_1, y_2)\left|f(x_i, y_1) \le f(x_i, y_2), (y_1, y_2) \in Y_i \times \overline{Y}_i\right.\}\right|$$

5.3. The choice of algorithms and the configuration of related parameters

In order to verify the performance of the proposed algorithm, the KELM-AE algorithm is compared with 5 multi-label mainstream classification algorithms, which are RELM [24], MLKNN, LIFT, GLOCAL [25] and MLFE, respectively. In the KELM-AE algorithm, the Non-equilibrium parameter is set to [0,0.5]. In the RELM algorithm, the regularization coefficient is set between [1,100], and the number of hidden layer neurons is set to 100. In the ML-KNN algorithm, the nearest neighbor K and the smoothing parameter s are set to 15 and 1 respectively. In the LIFT algorithm, the parameter $r = 0.1$ and the kernel function select Linear. In GLOCAL, the parameter λ is set to 1, and the other parameters, as well as those of the baseline methods, are selected via 5-fold cross-validation on the training set. In the MLFE algorithm, the kernel function selects RBF, and the kernel parameter β_1, β_2 and β_3 are selected from {1, 2, 10}, {1, 10, 15} and {1, 10}, which are cross validated on the training set respectively. Because the result of LIFT algorithm is unstable, in order to improve the accuracy, run LIFT 10 times, and the average (mean) and standard deviation (STD) are given in our experiments.

5.4. The experimental results

The experimental results of the KELM-AE and other 5 algorithms on 14 data sets are shown in Tables 2 to 6, where the ranking of the experimental results corresponding to each data set is shown in Tables 2 to 6 in the form of subscripts. The average ranking of each algorithm in all data sets is given in the last row, where the lower the average ranking, the better the algorithm. (Notes: No value indicates that the algorithm is stable with no change in the total 10 operations).

According to all the experimental results which are listed in Tables 2–6, we can draw conclusions as follows: (a)Comparing KELM-AE with MLkNN and RELM, the major deficiency of MLkNN and RELM is that it fails to consider the labels correlation. KELM-AE performs better than MLkNN and RELM at most cases since it considers the labels correlation. (b) Compared with MLFE, KELM-AE performs better on all data sets because MLFE only considers the relationship between features and tags and does not take advantage of the correlation among labels. (c) From the experimental results of LIFT, which train the multi-label classifier based on label-specific features, we can observe that KELM-AE achieves significantly better performance than LIFT at most cases. The reason is that the feature extraction in LIFT help it avoid dimensional disaster and feature redundancy. LIFT prompts us to consider feature dimension reduction to improve the accuracy of classification. (d) Compared with GLOCAL, which considers both global and local label correlations while the training of multi-label classifier, KELM-AE basically performs better overall on various evaluation criteria. It is worth mentioning that this also inspired us to consider local labels correlation information.

6. The stability analysis and statistical hypothesis test

In order to further illustrate the effectiveness of the proposed method, the stability analysis and hypothesis testing of the algorithm are carried out based on the experimental results.

6.1. The stability analysis

In order to verify the stability of different multi label learning algorithms, the spider net diagram is used to represent the stability analysis of algorithm [26]. Because the results of the prediction classification are very different in different data sets for different evaluation indicators, we standardize the results between [0.1,0.5] as a general standard. Finally, the stability index is represented through normalized values. Fig. 2 shows the stability of the algorithm under different data sets for each evaluation index.

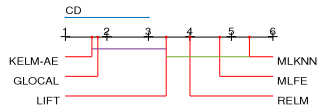
As shown in Fig. 2, we can observe: (1) For AP, KELM-AE obtains a fairly stable effect between the stable finger values of the 12 data sets in the [0.4,0.5]. (2) For CV, the stable value of KELM-AE on 12 data sets is between [0.4,0.5], and the solution is quite stable compared to the GLOCAL and LIFT algorithm. (3) For the HL, KELM-AE can get more stable results on 11 data sets, and are more stable than other algorithms. (4) For the OE, KELM-AE can provide a more stable solution on 12 data sets in the [0.4,0.5], and KELM-AE is also more stable than LIFT, GLOCAL and MLFE. (5) For RL, KELM-AE can achieve more stable solution on all data sets. Therefore, the results in Fig. 2 show that KELM-AE is more stable and has better prediction performance.

6.2. The statistical hypothesis test

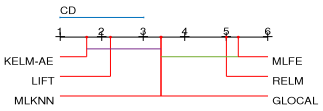
We statistically employ the Nemenyi Test [11,27] with significance of 5% to compare the experimental results of the KELM-AE and other algorithms in all 14 data sets. We also believe there is no significant difference between any two algorithms when their difference of the average ranking in all data sets are smaller or equal to the critical difference (CD), or there is significant difference. Every two algorithms are compared in terms of different evaluation indicators, as shown in Fig. 3, where the CD on the top line equals 2.0913, and the algorithms with no significant difference are connected by colorful lines. The algorithms are ranked in a decreasing order from left to right in each figure.

For each algorithm, there are 25 comparative results (5 comparative algorithms and 5 evaluation criterions). It is found in Fig. 3 that:

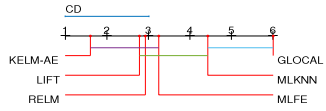
- For the KELM-AE algorithm, there is no statistically significant difference from the other algorithms about 52%. In terms of the AP, as shown in Fig. 3(a), there is no significant difference among KELM-AE, LIFT and GLOCAL algorithms. In terms of the CV, as shown in Fig. 3(b), there is no significant difference among KELM-AE. LIFT, GLOCAL and MLKNN algorithms. In terms of the HL, as shown in Fig. 3(c), KELM-AE, LIFT, MLFE and RELM algorithms do not have a significant difference. In terms of OE, as shown in Fig. 3(d), there is no significant difference among KELM-AE, RELM and GLOCAL algorithms. In terms of the RL, as shown in Fig. 3(e), there is no significant difference among KELM-AE, LIFT and GLOCAL algorithms. Therefore, the KELM-AE is superior to other algorithms in 48% cases.
- For the LIFT algorithm, there is no statistical difference between it and other algorithms in 80% of conditions, but in 24% cases, it is superior to other algorithms.
- For the GLOCAL algorithm, there is no statistical difference between it and other algorithms in 56% of conditions, but in 32% cases, it is superior to other algorithms.



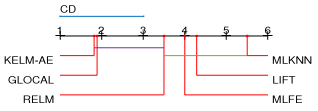
(a) AP.



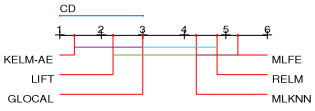
(b) CV.



(c) HL.



(d) OE.



(e) RL.

Fig. 3. The performance comparison of algorithms.

Table 2
The AP (↑) results of all 14 data sets.

Data	KELM-AE	RELM	MLKNN	LIFT	GLOCAL	MLFE
Arts	0.6257(1)	0.6071 ± 0.0011(4)	0.5093(6)	0.6072 ± 0.0039(3)	0.6234(2)	0.5912(5)
Business	0.8811(4)	0.8823 ± 0.0004(3)	0.8798(5)	0.8827 ± 0.0013(2)	0.8870(1)	0.8775(6)
Computer	0.7069(1)	0.6995 ± 0.0013(3)	0.6334(6)	0.6980 ± 0.0048(4)	0.7064(2)	0.6848(5)
Education	0.6459(2)	0.6331 ± 0.0014(3)	0.5995(6)	0.6328 ± 0.0024(4)	0.6467(1)	0.6190(5)
Entertainment	0.6791(3)	0.6778 ± 0.0015(4)	0.6012(6)	0.6852 ± 0.0032(2)	0.6913(1)	0.6745(5)
Health	0.7947(1)	0.7775 ± 0.0017(3)	0.6812(6)	0.7934 ± 0.0010(4)	0.7906(2)	0.7680(5)
Recreation	0.6311(2)	0.6269 ± 0.0015(3)	0.4544(6)	0.6213 ± 0.0032(4)	0.6321(1)	0.6104(5)
Reference	0.7215(1)	0.7050 ± 0.0015(3)	0.6193(6)	0.6991 ± 0.0021(4)	0.7173(2)	0.6977(5)
Science	0.6033(1)	0.5880 ± 0.0021(4)	0.5327(6)	0.5882 ± 0.0024(3)	0.6017(2)	0.5689(5)
Social	0.7719(1)	0.7664 ± 0.0012(5)	0.7490(6)	0.7686 ± 0.0028(3)	0.7715(2)	0.7565(4)
Society	0.6361(2)	0.6304 ± 0.0012(4)	0.6125(5)	0.6317 ± 0.0019(3)	0.6431(1)	0.6095(6)
Emotions	0.8015(2)	0.7640 ± 0.0162(5)	0.7808(4)	0.7485 ± 0.0106(6)	0.8031(1)	0.7822(3)
Natural scene	0.8196(1)	0.7531 ± 0.0065(6)	0.7615(4)	0.8059 ± 0.0023(3)	0.8112(5)	0.8166(2)
Yeast	0.7649(1)	0.7533 ± 0.0025(6)	0.7585(4)	0.7591 ± 0.0015(3)	0.7598(2)	0.7545(5)
Average ranking	1.6429	4	5.4286	3.4286	1.7857	4.7143

Table 3
The CV (↓) results of all 14 data sets.

Data	KELM-AE	RELM	MLKNN	LIFT	GLOCAL	MLFE
Arts	4.4583(1)	5.5812 ± 0.0369(5)	5.4453(4)	4.6974 ± 0.0733(2)	5.0237(3)	5.6857(6)
Business	2.2670(3)	2.5430 ± 0.0294(5)	2.1847(2)	2.1121 ± 0.0283(1)	2.3987(4)	2.7190(6)
Computer	3.6353(1)	4.6083 ± 0.0339(5)	4.4160(4)	3.8325 ± 0.0783(2)	4.0540(3)	4.7850(6)
Education	3.4767(1)	4.4921 ± 0.0332(5)	3.4953(2)	3.5057 ± 0.0244(3)	4.1257(4)	5.0973(6)
Entertainment	2.6650(2)	3.3208 ± 0.0233(5)	3.1477(4)	2.6568 ± 0.0289(1)	2.8983(3)	3.4060(6)
Health	2.6683(2)	3.5983 ± 0.0457(5)	3.3047(4)	2.6325 ± 0.0010(1)	3.2303(3)	3.9427(6)
Recreation	3.8303(2)	4.2339 ± 0.0246(4)	5.0973(6)	3.8044 ± 0.0348(1)	4.1420(3)	4.5443(5)
Reference	2.5733(1)	3.7601 ± 0.0472(5)	3.5420(4)	2.7317 ± 0.0353(2)	3.1087(3)	4.0130(6)
Science	5.5140(1)	6.9849 ± 0.0707(5)	6.0430(3)	5.5897 ± 0.0700(2)	6.0583(4)	7.3880(6)
Social	3.1713(3)	4.0361 ± 0.0422(5)	3.0313(2)	2.9629 ± 0.0908(1)	3.4410(4)	4.3287(6)
Society	5.3023(1)	6.2731 ± 0.0289(5)	5.3653(3)	5.3604 ± 0.0541(2)	5.6253(4)	6.3313(6)
Emotions	1.7871(1)	2.0965 ± 0.0733(4)	1.9158(2)	2.1574 ± 0.0604(5)	1.8614(6)	1.9703(3)
Natural_scene	0.8700(2)	1.0892 ± 0.0271(6)	1.0680(5)	0.8957 ± 0.0044(4)	0.8860(1)	0.8440(3)
Yeast	6.2999(2)	6.5706 ± 0.0277(6)	6.4144(3)	6.4689 ± 0.0254(4)	6.2756(1)	6.5027(5)
Average ranking	1.6429	5	3.4286	2.2143	3.4286	5.2857

- For the MLFE algorithm, there is no statistical difference between it and other algorithms in 60%, but in 4% cases, it is superior to other algorithms.

From the above analysis, the KELM-AE algorithm has the best performance. In 48% of cases, it is statistically superior to other algorithms, followed by the GLOCAL algorithm. In 32% of cases,

Table 4
The HL (↓) results of all 14 data sets.

Data	KELM-AE	RELM	MLKNN	LIFT	GLOCAL	MLFE
Arts	0.0539(1)	0.0545 ± 0.0001(2)	0.0612(5)	0.0546 ± 0.0002(3)	0.0632(6)	0.0576(4)
Business	0.0260(3)	0.0254 ± 0.0001(2)	0.0269(5)	0.0262 ± 0.0001(4)	0.0529(6)	0.0254(1)
Computer	0.0342(1)	0.0351 ± 0.0001(3)	0.0412(5)	0.0343 ± 0.0001(2)	0.0461(6)	0.0358(4)
Education	0.0371(1)	0.0377 ± 0.0001(2)	0.0387(4)	0.0381 ± 0.0002(3)	0.0442(6)	0.0389(5)
Entertainment	0.0524(2)	0.0525 ± 0.0001(3)	0.0603(5)	0.0521 ± 0.0002(1)	0.0675(6)	0.0540(4)
Health	0.0324(1)	0.0347 ± 0.0001(3)	0.0458(5)	0.0326 ± 0.0022(2)	0.0518(6)	0.0384(4)
Recreation	0.0547(1)	0.0565 ± 0.0001(3)	0.0618(5)	0.0548 ± 0.0002(2)	0.0650(6)	0.0571(4)
Reference	0.0253(1)	0.0257 ± 0.0001(4)	0.0314(5)	0.0256 ± 0.0002(3)	0.0357(6)	0.0256(2)
Science	0.0307(1)	0.0312 ± 0.0001(2)	0.0325(5)	0.0316 ± 0.0001(4)	0.0356(6)	0.0313(3)
Social	0.0201(1)	0.0206 ± 0.0001(3)	0.0218(5)	0.0207 ± 0.0001(4)	0.0331(6)	0.0202(2)
Society	0.0509(1)	0.0519 ± 0.0002(2)	0.0536(5)	0.0525 ± 0.0002(3)	0.0624(6)	0.0531(4)
Emotions	0.2104(1)	0.2381 ± 0.0069(4)	0.2137(2)	0.2368 ± 0.0034(5)	0.3292(6)	0.2459(3)
Natural scene	0.1736(3)	0.1991 ± 0.0046(5)	0.1836(4)	0.1654 ± 0.0017(2)	0.2470(6)	0.1624(1)
Yeast	0.2038(4.5)	0.2016 ± 0.0015(3)	0.1980(2)	0.1974 ± 0.0010(1)	0.3038(6)	0.2038(4.5)
Average ranking	1.6071	2.9286	4.4286	2.7857	6	3.25

Table 5
The OE (↓) results of all 14 data sets.

Data	KELM-AE	RELM	MLKNN	LIFT	GLOCAL	MLFE
Arts	0.4670(1.5)	0.4803 ± 0.0024(3)	0.6327(6)	0.4920 ± 0.0063(4)	0.4670(1.5)	0.4953(5)
Business	0.1137(2)	0.1168 ± 0.0009(4)	0.1213(5)	0.1222 ± 0.0023(6)	0.1130(1)	0.1147(3)
Computer	0.3497(1)	0.3610 ± 0.0032(3)	0.4367(6)	0.3614 ± 0.0069(4)	0.3520(2)	0.3770(5)
Education	0.4550(2)	0.4718 ± 0.0024(3)	0.5207(6)	0.4781 ± 0.0030(5)	0.4537(1)	0.4780(4)
Entertainment	0.4253(5)	0.4130 ± 0.0028(4)	0.5303(6)	0.4084 ± 0.0057(2)	0.3977(1)	0.4123(3)
Health	0.2497(2)	0.2716 ± 0.0034(4)	0.4207(6)	0.2494 ± 0.0019(1)	0.2543(3)	0.2743(5)
Recreation	0.4693(3)	0.4684 ± 0.0024(2)	0.7067(6)	0.4815 ± 0.0052(4)	0.4610(1)	0.4860(5)
Reference	0.3567(1)	0.3762 ± 0.0015(3)	0.4730(6)	0.3865 ± 0.0020(5)	0.3677(2)	0.3820(4)
Science	0.4920(1)	0.4992 ± 0.0033(3)	0.5803(6)	0.5103 ± 0.0040(4)	0.4930(2)	0.5187(5)
Social	0.2793(1)	0.2853 ± 0.0020(2)	0.3257(6)	0.2941 ± 0.0033(5)	0.2890(3)	0.2923(4)
Society	0.3987(2)	0.4007 ± 0.0017(3)	0.4370(6)	0.4059 ± 0.0019(4)	0.3933(1)	0.4273(45
Emotions	0.2921(2)	0.3287 ± 0.0308(4)	0.3317(5)	0.3569 ± 0.0145(6)	0.2822(1)	0.3069(3)
Natural scene	0.2770(1)	0.3852 ± 0.0107(6)	0.3670(5)	0.3002 ± 0.0059(4)	0.2950(3)	0.2910(2)
Yeast	0.2312(1)	0.2388 ± 0.0053(5)	0.2345(2)	0.2412 ± 0.0034(6)	0.2386(4)	0.2356(3)
Average ranking	1.8214	3.5	5.5	4.2857	1.8929	4

Table 6
The RL (↓) results of all 14 data sets.

Data	KELM-AE	RELM	MLKNN	LIFT	GLOCAL	MLFE
Arts	0.1135(1)	0.1434 ± 0.0011(4)	0.1520(6)	0.1208 ± 0.0018(2)	0.1259(3)	0.1489(5)
Business	0.0365(2)	0.0422 ± 0.0006(5)	0.0374(3)	0.0346 ± 0.0001(1)	0.0396(4)	0.0464(6)
Computer	0.0709(1)	0.0949 ± 0.0008(5)	0.0922(4)	0.0758 ± 0.0002(2)	0.0818(3)	0.1007(6)
Education	0.0744(1)	0.0936 ± 0.0006(5)	0.0800(3)	0.0773 ± 0.0018(2)	0.0858(4)	0.1097(6)
Entertainment	0.0927(2)	0.1139 ± 0.0008(4)	0.1151(5)	0.0916 ± 0.0012(1)	0.0976(3)	0.1184(6)
Health	0.0412(1)	0.0563 ± 0.0008(4)	0.0605(5)	0.0416 ± 0.0004(2)	0.0486(3)	0.0646(6)
Recreation	0.1296(1)	0.1436 ± 0.0007(4)	0.1912(6)	0.1310 ± 0.0015(2)	0.1398(3)	0.1568(5)
Reference	0.0609(1)	0.0889 ± 0.0011(4)	0.0919(5)	0.0661 ± 0.0010(2)	0.0729(3)	0.0970(6)
Science	0.1009(1)	0.1300 ± 0.0007(5)	0.1166(4)	0.1036 ± 0.1008(2)	0.1107(3)	0.1399(6)
Social	0.0554(2)	0.0709 ± 0.0007(5)	0.0561(3)	0.0539 ± 0.0013(1)	0.0601(4)	0.0779(6)
Society	0.1268(1)	0.1481 ± 0.0006(5)	0.1339(4)	0.1286 ± 0.0012(2)	0.1305(3)	0.1547(6)
Emotions	0.1529(1)	0.2020 ± 0.0143(5)	0.1729(3)	0.2226 ± 0.0125(6)	0.1608(2)	0.1803(4)
Natural scene	0.1490(2)	0.2057 ± 0.0062(6)	0.1982(5)	0.1551 ± 0.0031(4)	0.1543(3)	0.1452(1)
Yeast	0.1682(2)	0.1789 ± 0.0017(6)	0.1715(4)	0.1697 ± 0.0011(3)	0.1636(1)	0.1777(5)
Average ranking	1.3571	4.7857	4.2867	2.2857	3	5.2857

it is statistically superior to other algorithms. The third is the LIFT algorithm, which is superior to other algorithms in 24% of cases. The KELM-AE algorithm has the best performance, and the experiment further illustrates the effectiveness of the KELM-AE algorithm.

7. Conclusion

In multi-label classification learning, it is very important to study the correlation between feature information and labels in multi label learning. In this paper we propose a Kernel Extreme Learning Machine AutoEncoder algorithm, which use the kernel extreme learning machine to autoencoder the fuzzy associations between the features in the input space, and use the non-equilibrium labels completion algorithm to add the correlation between the labels in the labels space. Therefore, the relevant information contained in the feature space and the labels space can be fully investigated. Experimental results show that KELM-AE algorithm is better than some common multi-label learning algorithms.

Because the new generated features cannot theoretically guarantee a strong correlation with the labels. In the future, we will study the deep relationship between feature space and label space, as well as the method of feature selection and local label correlation. We will fully exploit the effective information contained in the output space, and combine these methods to construct a unified multi-label learning framework.

Acknowledgments

This research is supported by the Natural Science Foundation of Higher Education of Anhui Province, China (No. KJ2017A177),

and Program for Innovative Research Team of Anqing Normal University, China.

References

[1] Zhi-Hua Zhou, Min-Ling Zhang, Multi-label Learning, Springer, 2017, pp. 875–881.

[2] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, Christopher M. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.

[3] Min-Ling Zhang, Zhi-Hua Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351.

[4] Min-Ling Zhang, Zhi-Hua Zhou, Ml-knn: A lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048.

[5] André Elisseeff, Jason Weston, A kernel method for multi-labelled classification, in: Advances in Neural Information Processing Systems, 2002, pp. 681–687.

[6] Claude Elwood Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423.

[7] Zhenhai Zhang, Shining Li, Zhigang Li, A multi-label classification algorithm using correlation information entropy, J. Northwest. Polytech. Univ. 30 (6) (2012) 968–973, in Chinese.

[8] Jaedong Lee, Heera Kim, Noo-ri Kim, Jee-Hyong Lee, An approach for multi-label classification by directed acyclic graph with label correlation maximization, Inform. Sci. 351 (2016) 101–114.

[9] Min-Ling Zhang, Lei Wu, Lift: Multi-label learning with label-specific features, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2015) 107–120.

[10] Jia Zhang, Candong Li, Donglin Cao, Yaojin Lin, Songzhi Su, Liang Dai, Shaozi Li, Multi-label learning with label-specific features by resolving label correlations, Knowl.-Based Syst. 159 (2018) 148–157.

[11] Jun Huang, Guorong Li, Qingming Huang, Xindong Wu, Joint feature selection and classification for multilabel learning, IEEE Trans. Cybern. 48 (3) (2018) 876–889.

[12] Qian-Wen Zhang, Yun Zhong, Min-Ling Zhang, Feature-induced labeling information enrichment for multi-label learning, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, 2018, pp. 4446–4453.

[13] Hao Wang, Yan Yang, Bing Liu, Hamido Fujita, A study of graph-based system for multi-view clustering, Knowl.-Based Syst. 163 (2019) 1009–1019.

[14] Yiling Zhang, Yan Yang, Tianrui Li, Hamido Fujita, A multitask multiview clustering algorithm in heterogeneous situations based on lle and le, Knowl.-Based Syst. 163 (2019) 776–786.

[15] Chao-Hui Tang, Qing-Xin Zhu, Chao-Qun Hong, William Zhu, Multi-label feature selection with autoencoders and hypergraph learning, Acta Automat. Sinica 42 (7) (2016) 1014–1021, in Chinese.

[16] Yu-Dong Zhang, Yin Zhang, Xiao-Xia Hou, Hong Chen, Shui-Hua Wang, Seven-layer deep neural network based on sparse autoencoder for voxelwise detection of cerebral microbleed, Multimedia Tools Appl. (2018) 1–18.

[17] Gao Huang, Guang-Bin Huang, Shiji Song, Keyou You, Trends in extreme learning machines: A review, Neural Netw. 61 (2015) 32–48.

[18] Guang-Bin Huang, An insight into extreme learning machines: random neurons, random features and kernels, Cogn. Comput. 6 (3) (2014) 376–390.

[19] Liyanaarachchi Lekamalage Chamara Kasun, Hongming Zhou, Guang-Bin Huang, Chi Man Vong, Representational learning with extreme learning machine for big data, IEEE Intell. Syst. 28 (6) (2013) 31–34.

[20] J.Y. Liang, C.Y. Dang, K.S. Chin, C.M. Yam Richard, A new method for measuring of rough sets and rough relational databases, Inform. Sci. 31 (4) (2002) 331–342.

[21] Clara Pizzuti, A multi-objective genetic algorithm for community detection in networks, in: Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on, IEEE, 2009, pp. 379–386.

[22] Ji Feng, Zhi-Hua Zhou, Autoencoder by forest, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[23] Min-Ling Zhang, Zhi-Hua Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1819–1837.

[24] Wan-Yu Deng, Qing-Hua Zheng, Lin Chen, Xue-Bin Xu, Research on extreme learning of neural networks, Chinese J. Comput. 33 (2) (2010) 279–287.

[25] Yue Zhu, James T. Kwok, Zhi-Hua Zhou, Multi-label learning with global and local label correlation, IEEE Trans. Knowl. Data Eng. 30 (6) (2018) 1081–1094.

[26] Yaojin Lin, Yuwen Li, Chenxi Wang, Jinkun Chen, Attribute reduction for multi-label learning with fuzzy rough set, Knowl.-Based Syst. 152 (2018) 51–61.

[27] Janez Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.