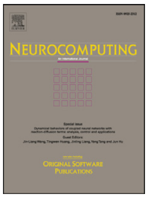
Neurocomputing 277 (2018) 78–88

Contents lists available at ScienceDirect

Neurocomputing

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



Extreme Learning Machine for Joint Embedding and Clustering

Tianchi Liu, Chamara Kasun Liyanaarachchi Lekamalage, Guang-Bin Huang, Zhiping Lin*

School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore

ARTICLE INFO

Article history:
Received 6 September 2016
Revised 24 November 2016
Accepted 7 January 2017
Available online 24 August 2017

Keywords:
Feature learning
Embedding
Clustering
k-means
Manifold regularization
Extreme learning machine

ABSTRACT

Clustering generic data, i.e., data not specific to a particular field, is a challenging problem due to their diverse complex structures in the original feature space. Traditional approaches address this problem by complementing clustering with feature learning methods, which either capture the intrinsic structure of the data or represent the data such that clusters are better revealed. In this paper, we propose an approach referred to as Extreme Learning Machine for Joint Embedding and Clustering (ELM-JEC), which incorporates desirable properties of both types of feature learning methods at the same time, specifically by (1) preserving the manifold structure of the data in the original space; (2) maximizing the class separability of the data in the embedded space. Since either type of method has improved clustering performance in some cases, our motivation is to integrate the two desirable properties to further improve the accuracy and robustness of clustering. Additional notable features of ELM-JEC are that it provides nonlinear feature mappings and achieves feature learning and clustering in the same formulation. The proposed approach can be implemented using alternating optimization, and its clustering performance compares favorably with several state-of-the-art methods on the real-world benchmark datasets.

1. Introduction

Clustering is a task of finding a partition of data in an unsupervised manner such that the data in the same group are more similar to each other than to those in other groups [1]. Many real world problems in a vast variety of fields, such as market research [2], document classification [3], and bioinformatics [4], can be formulated as the fundamental task of clustering. Clustering methods [5] reveal the intrinsic grouping of data and thus have a wide range of applications, such as image segmentation [6], speech separation [7], protein sequence classification [8], etc. Prominent traditional clustering methods include *k*-means, which is a centroid-based method [9], and Gaussian mixture model, which is a distribution-based clustering method solved using expectation maximization algorithm [10]. Many efficient optimization techniques [11,12] have been proposed to overcome the problems of slow convergence and local minima. Over the past decade, graph theories have been intensively studied and have inspired many graph-based clustering methods [13,14,6].

Feature learning (or embedding) methods aim to obtain a new representation of the data that is more suitable for the subsequent machine learning tasks, such as clustering. Feature learning is commonly referred as dimensionality reduction [15] if the embedded space has lower dimension than the original space. Traditional approaches [10] aim to find a subspace where data are presented with maximum variance, as in Principal Component Analysis (PCA), or maximum separability, as in Linear Discriminant Analysis (LDA). Motivated by the complex structures of data, researchers proposed many nonlinear feature learning approaches, such as locality linear embedding [16], Laplacian eigenmap [13], and isometric mapping [17]. These methods learn features by preserving important local or global structures of the data, while the redundant or noisy information is discarded during the process of feature learning.

With the advancement of sensor and internet technology, we are often confronted with generic data with diverse structures generated from many emerging fields. Since the cluster structures of such generic data may not be prominent in the original feature space, feature learning and clustering are usually conducted in sequence to achieve better clustering results, for instance, using PCA for dimensionality reduction followed by *k*-means for clustering. Alternatively, in the framework of joint feature learning and clustering, the two processes are conducted simultaneously. The advantage of this framework is that, with the clustering solution available as label information, supervised methods, such as LDA, can also be used for feature learning. Intuitively, better clustering performance is expected from the joint feature learning and clustering methods because the requirement of clustering is taken into account during the feature learning step. As a well-known example, Ding and Li [18] proposed to conduct LDA and *k*-means

* Corresponding author.
E-mail addresses: tcliu@ntu.edu.sg (T. Liu), chamarak001@e.ntu.edu.sg (C.K. Liyanaarachchi Lekamalage), egbhuang@ntu.edu.sg (G.-B. Huang), ezplin@ntu.edu.sg, ezplin@gmail.com (Z. Lin).

http://dx.doi.org/10.1016/j.neucom.2017.01.115
0925-2312/© 2017 Published by Elsevier B.V.

alternately, which yields significantly better clustering performance compared with the counterpart of conducting PCA and k-means in sequence. This method is later proved by Ye et al. [19] to be equivalent to kernel k-means with a specific kernel function. Recently, Hou et al. [20] proposed a framework that unifies several joint feature learning and clustering techniques. However, in these methods, embedding is achieved by linear transformation, which have limited capability in discovering the nonlinear structures of the data. Hence, it is desirable to look for methods which can perform nonlinear transformation for embedding.

Recently, Extreme Learning Machine (ELM) [21] was first proposed as a training algorithm for Single-Layer Feedforward Networks (SLFNs), which can approximate nonlinear feature mapping efficiently using the nonlinear hidden neurons. The notable advantage of ELM is that parameters of hidden layer neurons need not be tuned to achieve universal approximation capability but can be randomly generated and fixed. Consequently, only the output weights are free parameters, which can be adjusted to optimize various learning criteria and allow different types of regularization. For example, by minimizing the empirical training error, ELM [22] was first proposed for supervised classification and regression tasks; Later by adding manifold regularization to the formulation, Semi-Supervised ELM (SS-ELM) [23] extends ELM to handle semi-supervised classification and regression tasks. ELM has become a state-of-the-art learning framework and has been widely used in computer vision [24,25], bioinformatics [26], market research [27], engineering [28], etc.

Thanks to ELM’s universal approximation capability [29], the SLFNs trained using ELM can be used as effective nonlinear transformation tools. If more clear and accurate cluster structures can be observed afterward, such transformation is considered beneficial for clustering. Motivated by this fact, several research studies have been carried out on finding suitable criteria for ELM learning to achieve the optimal nonlinear transformation. He et al. [30] proposed to conduct Clustering in ELM Feature Space (C-ELM-FS) to achieve nonlinear embedding. Unlike C-ELM-FS, which omits the output weights and the output layer and thus sacrifices the flexibility of ELM, subsequent studies make use of the complete SLFN architecture and can be categorized into two groups. The first group aims to capture the intrinsic structure of the data in the original space. Huang et al. [23] extended ELM to embedding and clustering under the manifold regularization framework and called the method Un-Supervised ELM (US-ELM). The main idea is to learn a nonlinear data embedding which preserves the intrinsic manifold structure and subsequently conduct clustering in the embedded space. In [31], both local manifold structure and global discriminative information in the original data space are preserved by Unsupervised Discriminative ELM (UD-ELM). However, UD-ELM requires the dimension of embedded space to be equal to the number of classes, which sacrifices the flexibility of the data representation.

The second group uses ELM to represent the data in a way such that clusters are better revealed in the embedded space. In [32], clustering is conducted in the embedded space whose projection matrix is solved by ELM Auto-Encoder. The study shows that such embedding can preserve the between-class variance and reduce the within-class variance in the embedded space, which effectively increases the class separability. Huang et al. [33] proposed three ELM-based Discriminative Clustering methods (ELM-DC). The motivation is to solve ELM embedding and cluster indicators iteratively such that the embedding could lead to good classification results as measured by the objective functions of supervised methods, i.e., LDA or ELM. A better classification result implies higher class separability of the data in the embedded space.

While both groups achieve the desirable property of nonlinearity efficiently, they have some drawbacks. The first group conducts embedding and clustering separately, and hence the embedding fails to take into account the requirement of clustering. This could result in overlapping cluster structures or even no cluster structures in the embedded space, which may affect the clustering performance. The second group tries to obtain an embedded space with clearer cluster structures but ignores the intrinsic structure of the data in the original space. The embedding obtained in this way may alter the local structure of the data and result in artificial clusters in the embedded space, which could lead to wrong clustering results. To the best of our knowledge, no existing joint nonlinear embedding and clustering methods have a single mechanism that simultaneously preserves the intrinsic structure of the data and represents data in a way such that clusters are better revealed.

In this paper, to improve the clustering performance, we propose a method, named Extreme Learning Machine for Joint Embedding and Clustering (ELM-JEC), to satisfy the desirable properties of both groups, specifically by (1) preserving the manifold structure of the data in the original space and (2) maximizing the class separability of the data in the embedded space. In brief, the nonlinear embedding is approximated by an ELM SLFN, whose the output weights are solved together with cluster indicator variables. The first property is formulated based on the manifold regularization [34], and the second requirement based on Discriminative Embedded Clustering (DEC) [20]. We implement the proposed approach using alternating optimization and compare it with several state-of-the-art methods on the real-world benchmark datasets.

The paper is organized as follows: Section 2 provides background on the related concepts and two methods that inspire our work. In Section 3, we summarize some desirable properties for embedding and clustering. In Section 4, we propose the formulation and details of the ELM-JEC method. Experimental results are given in Sections 5, and 6 concludes the paper. We give a summary of notations used in Table 1.

2. Related works

In this section, we review the manifold regularization framework, DEC and ELM, which inspire our work.

2.1. Manifold regularization

Many applications, such as text document clustering, face recognition, etc., involve high-dimensional data. The idea that naturally generated high-dimensional data reside on a lower dimensional manifold has motivated many graph-based methods. Particularly, manifold regularization [34] is a framework proposed to capture the manifold structure based on the *smoothness assumption* in machine learning, i.e., if two data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are close to each other, the conditional probabilities $\mathcal{P}(y|\boldsymbol{x}_i)$ and $\mathcal{P}(y|\boldsymbol{x}_j)$ should be similar as well. In supervised learning, the conditional probability is approximated by the predicted class indicator $\hat{\boldsymbol{y}}_i$ and $\hat{\boldsymbol{y}}_j$ with respect to the i-th and j-th data points. Therefore, we have the following cost function:

$$\begin{aligned}\hat{L}_m &= \frac{1}{2}\sum_{i,j} w_{ij}\|\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_j\|^2 \\ &= \mathrm{Tr}\left(\hat{\boldsymbol{Y}}^\top \boldsymbol{L}\hat{\boldsymbol{Y}}\right),\end{aligned} \quad (1)$$

where $\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \ldots, \hat{\boldsymbol{y}}_N]^\top$ is the predicted class indicator matrix and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ is the *graph Laplacian*. Here $\boldsymbol{W} = [w_{ij}]$ is the *data adjacency graph* and can be computed in various ways according to the application (for example, a k nearest neighbor graph with binary edge weights) and $\boldsymbol{D}$ is a diagonal matrix with its diagonal elements $D_{ii} = \sum_{j=1}^{N} w_{ij}$, where w_{ij} is the weight between the ith

Table 1
Notations.

n_i	Dimension of input space
n_h	Number of hidden neurons in ELM
n_o	Dimension of output space
c	Number of clusters/class
N	Number of data points
$\boldsymbol{x}_i \in \mathbb{R}^{n_i}$	The *i*th data point in the input space
$\boldsymbol{y}_i \in \mathbb{R}^{c}$	The target class indicator of the *i*th data point
$\boldsymbol{h}(\boldsymbol{x}_i) \in \mathbb{R}^{1\times n_h}$	The output of hidden layer with respect to the *i*th data point
$\boldsymbol{f}_i \in \mathbb{R}^{c}$	The cluster indicator of the *i*th data point
$\boldsymbol{g}_j \in \mathbb{R}^{n_o}$	The *j*th cluster centroid
$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N\times n_i}$	Data matrix in the input space
$\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]^\top \in \mathbb{R}^{N\times c}$	Target class indicator matrix
$\boldsymbol{H} = [\boldsymbol{h}(\boldsymbol{x}_1)^\top, \boldsymbol{h}(\boldsymbol{x}_2)^\top, \ldots, \boldsymbol{h}(\boldsymbol{x}_N)^\top]^\top \in \mathbb{R}^{N\times n_h}$	Data matrix in hidden layer output space
$\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N]^\top \in \mathbb{R}^{N\times c}$	Cluster indicator matrix
$\boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_c] \in \mathbb{R}^{n_o\times c}$	Cluster centroid matrix
$\boldsymbol{\beta} \in \mathbb{R}^{n_h\times n_o}$	Output weights between hidden layer and output layer
$\boldsymbol{E} \in \mathbb{R}^{N\times n_o}$	Data matrix in the output space
$\boldsymbol{Q} \in \mathbb{R}^{n_i\times n_o}$	Linear transformation matrix
$\boldsymbol{L} \in \mathbb{R}^{N\times N}$	The *graph Laplacian* matrix

and *j*th data point and N is the total number of data points. We denote by $\mathrm{Tr}(\cdot)$ the operation of computing the trace of a matrix. By minimizing the cost function, we obtain the predicted class labels that are smooth with respect to the underlying manifold structure of the data and thereby have preserved the manifold structure.

2.2. Discriminative embedded clustering

DEC [20] is a recently proposed framework which unifies several existing methods for joint dimensionality reduction and clustering. It explicitly conducts the two tasks in the same formulation and uses an iterative algorithm to find the local optimal solution. From the perspective of dimensionality reduction, DEC essentially finds a linear transformation matrix that maximizes the class separability [35]. Specifically, given the desired number of clusters c and the unlabeled training data set $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N\times n_i}$, the formulation of DEC is as follows:

$$\begin{aligned} \max_{\boldsymbol{Q}, \boldsymbol{G}, \boldsymbol{F}} \quad & \mathrm{Tr}(\boldsymbol{Q}^\top \boldsymbol{S}_t \boldsymbol{Q}) - \lambda \|\boldsymbol{X}\boldsymbol{Q} - \boldsymbol{F}\boldsymbol{G}^\top\|_F^2 \\ s.t. \quad & \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}_{n_o} \end{aligned} \tag{2}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n_i\times n_o}$ is the linear transformation matrix, $\boldsymbol{S}_t = \sum_{i=1}^{N} \left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)^\top$ is the total scatter matrix with $\bar{\boldsymbol{x}}$ being the centroid of all data points, $\boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_c] \in \mathbb{R}^{n_o\times c}$ is the cluster centroid matrix with $\boldsymbol{g}_i \in \mathbb{R}^{n_o}$ being the centroid of *i*th cluster, and $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N]^\top \in \mathbb{R}^{N\times c}$ is the cluster indicator matrix with $\boldsymbol{f}_i = [f_{i1}, f_{i2}, \ldots, f_{ic}]^\top \in \mathbb{R}^c$, $f_{ij} = 1$ if and only if $\boldsymbol{x}_i$ is assigned to the *j*th cluster and $f_{ij} = 0$ otherwise. We denote by $\boldsymbol{I}_{n_o}$ the identity matrix of dimension n_o, and c is the predefined number of clusters. $\|\cdot\|_F$ represents the Frobenius norm of a matrix.

As we can see from the above formulation, the first term in the objective function aims to maximize the total variance of data points in the embedded space, similar to the objective of PCA. The second term is derived from the objective of *k*-means, i.e., minimizing the squared sum of distances of data points to their respective cluster centroids. It has been proved [20] that when $\lambda > 1$, the formulation is equivalent to maximizing the between-class separation and minimizing the within-class variance; when $\lambda = 1$, the formulation is equivalent to only maximizing the between-class separation; when $0 < \lambda < 1$, both the between-class separation and within-class variances are maximized, but DEC puts a larger weight on the between-class separation.

Hou et al. [20] proposed to solve this problem iteratively since it is not a jointly convex problem: (1) fixing $\boldsymbol{Q}$, $\boldsymbol{G}$ and optimizing $\boldsymbol{F}$, which is similar to cluster assignment process in *k*-mean given fixed cluster centroids; (2) fixing $\boldsymbol{F}$ and optimizing $\boldsymbol{Q}$, $\boldsymbol{G}$. The two steps are iteratively performed.

2.3. Extreme learning machine

ELM was proposed by Huang et al. [21] as a class of learning algorithms for Single-Layer Feedforward Networks (SLFNs). Given a data point $\boldsymbol{x}$, the sigmoid function can be used as the activation function of hidden neurons:

$$g(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{1 + \exp(-(\boldsymbol{a}^\top \boldsymbol{x} + b))}, \tag{3}$$

where $\boldsymbol{\theta} = \{\boldsymbol{a}, b\}$ are the parameters of the mapping function. We denote the output of the hidden layer by $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^{1\times n_h}$ and the output weights between the hidden layer and the output layer by $\boldsymbol{\beta} \in \mathbb{R}^{n_h\times n_o}$.

Without sacrificing the universal approximation capability of SLFNs, ELM theory [21,36] states that parameters of the hidden layer need not be tuned but can be generated randomly independent of training data. Consequently, training an SLFN using ELM is essentially just to solve the output weights. Different solutions of the output weights have been proposed based on the information available in the training set and the objective of the task.

Here we review the basic ELM for supervised classification problems. We have a training data set of N samples $\{\boldsymbol{X}, \boldsymbol{Y}\}$, where $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N\times n_i}$ is the data matrix, and $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]^\top \in \mathbb{R}^{N\times c}$ is the class indicator matrix with $\boldsymbol{y}_k = [y_{k1}, y_{k2}, \ldots y_{kc}]^\top$ and $y_{ki} = 1$ if and only if $\boldsymbol{x}_k$ belongs to the *i*th class or else $y_{ki} = 0$. In supervised learning, ELM can be used to approximate a mapping function between a input data point and its target class indicator. Therefore, the number of output nodes is set to the number of class, i.e., $n_o = c$. An optimal classifier can be achieved by minimizing the empirical training error which leads to the following formulation:

$$\min_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\|\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{\beta}\|^2, \tag{4}$$

where $\boldsymbol{H} = [\boldsymbol{h}(\boldsymbol{x}_1)^\top, \ldots, \boldsymbol{h}(\boldsymbol{x}_N)^\top]^\top \in \mathbb{R}^{N\times n_h}$ is the data matrix in the ELM hidden layer output space, and $C > 0$ is a trade-off parameter on the training errors. Here the first term in the objective function is a regularization term against over-fitting, and the second term is the error of all training data points.

This formulation is a case of the widely-known optimization problem of ridge regression or regularized least squares. The free parameters can be effectively solved in closed form as follows:

$$\boldsymbol{\beta}^* = \left(\boldsymbol{H}^\top \boldsymbol{H} + \frac{\boldsymbol{I}_{n_h}}{C}\right)^{-1} \boldsymbol{H}^\top \boldsymbol{Y} \tag{5}$$

where $\boldsymbol{I}_{n_h}$ is the identity matrix of dimension n_h.

Table 2
Desirable properties for embedding and clustering and the related methods.

	Property	C-ELM-FS	US-ELM	UD-ELM	C-ELM-AE	LDA-KM	ELM-DC	DEC	ELM-JEC
Embedding	Nonlinearity	√	√	√	√	✗	√	✗	√
	Structure preserving	✗	√	√	✗	✗	✗	✗	√
	Separability maximizing	✗	✗	✗	√	√	√	√	√
Clustering	Simultaneous embedding and clustering	✗	✗	✗	✗	√	√	√	√

A "√" symbol indicates the method (on the same column as the symbol) has the property (on the same row as the symbol), while "✗" indicates the absence of the property.

3. Desirable properties for embedding and clustering

Methods using embedding to enhance the performance of clustering can be viewed from the embedding and clustering aspects. In this section, we briefly compare some related methods and summarize the desirable properties as follows.

Nonlinearity. If data in the embedded space cannot be written as a linear combination of the data in the original space, we regard the embedding as nonlinear. Data from real-world applications usually form clusters that are not linearly separable. Therefore, nonlinearity is a highly desired property of embedding methods. For example, Clustering in ELM Feature Space (C-ELM-FS) [30], Un-Supervised ELM (US-ELM) [23], Unsupervised Discriminative ELM (UD-ELM) [31], Clustering with ELM Auto-Encoder (C-ELM-AE) [32], and ELM-based Discriminative Clustering (ELM-DC) [33] are all nonlinear in nature as they adopt ELM to approximate the embedding.

Preserving the intrinsic geometric structure of the data in the original space (Structure preserving). High-dimensional data usually reside on low-dimensional manifolds, which are essential for representing the data effectively. Therefore, the desirable embedded data should have the same intrinsic geometric structure as data in the original space [15]. For example, US-ELM [23] adopts the manifold regularization to achieve maximum smoothness of the embedding with respect to the manifold structure of the data. In UD-ELM [31], both local manifold structure and the global discriminant structure are captured by the embedding, while the redundant or noisy information is discarded.

Maximizing the class separability of the data in the embedded space (Separability maximizing). Since the purpose of embedding is to improve the performance of clustering, data in the embedded space should have more prominent clusters structure. In other words, the embedding should maximize the class separability [35] of the data in the embedded space. For example, C-ELM-AE achieves this by utilizing the embedding function solved by ELM-AE, which preserves the between-class variance and reduces the within-class variance. Adaptive LDA-guided k-means clustering (LDA-KM) [18] directly adopts LDA and alternates between LDA and k-means. ELM-DC achieves maximum class separability by using supervised embedding methods, i.e., LDA and ELM. The objective function of ELM can be interpreted as minimizing the within-class variance, where the class centroids are fixed as the unit vectors in the c-dimensional embedded space (c is the number of clusters). Discriminative Embedded Clustering (DEC) [20] maximizes the class separability of the data in the embedded space by combining the objective functions of PCA and k-means.

Simultaneous embedding and clustering. Simultaneous embedding and clustering are achieved if both the embedding and the clustering results can be obtained using the same approach. The main disadvantage of conducting embedding and clustering in sequence is that other off-the-shelf clustering methods may introduce additional uncertainty and the embedding may not be optimized for a specific clustering method. Only LDA-KM, ELM-DC, and DEC can jointly conduct embedding and clustering, whereas the other methods in Table 2 rely on separate off-the-shelf clustering methods. However, LDA-KM and DEC use a linear transformation for embedding. Although ELM-DC can approximate nonlinear feature mappings, it focuses only on the cluster structure in the embedded space, and ignore the intrinsic structure of the data in the original space.

We summarize the properties and the related methods in Table 2. To the best of our knowledge, no existing methods simultaneously satisfy all the four properties mentioned above. In this study, we derive an algorithm that is optimized for *structure preserving* and *separability maximizing* while enforcing simultaneous *nonlinear* embedding and clustering.

4. ELM for joint embedding and clustering

In a clustering task, given a desired number of clusters c and a data set of N data points $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N\times n_i}$, the goal is to find the cluster indicator matrix $\boldsymbol{F}$ and the cluster centroid matrix $\boldsymbol{G}$. Here $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N]^\top \in \mathbb{R}^{N\times c}$ and $\boldsymbol{f}_i = [f_{i1}, f_{i2}, \ldots f_{ic}]^\top$, where $f_{ij} = 1$ if and only if $\boldsymbol{x}_i$ is assigned to the jth cluster and $f_{ij} = 0$ otherwise; $\boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_c] \in \mathbb{R}^{n_o\times c}$, where $\boldsymbol{g}_i \in \mathbb{R}^{n_o}$ is the centroid of the ith cluster.

The main idea of ELM for Joint Embedding and Clustering (ELM-JEC) is to jointly approximate a nonlinear embedding using ELM and output clusters based on the data in the embedded space. According to ELM theory, the parameters of hidden layers need not be tuned so the desired embedding can be obtained by adjusting only the output weights between hidden layer and output layer. In order to satisfy the properties of both *structure preserving* and *separability maximizing*, we maximize the following objective function with respect to the output weight $\boldsymbol{\beta} \in \mathbb{R}^{n_h\times n_o}$, the cluster centroid matrix $\boldsymbol{F}$, and the cluster indicator matrix $\boldsymbol{G}$:

$$\begin{aligned} \max_{\boldsymbol{\beta}, \boldsymbol{G}, \boldsymbol{F}} \quad & \mathrm{Tr}(\boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \bar{\boldsymbol{H}} \boldsymbol{\beta}) - \lambda \|\bar{\boldsymbol{H}}\boldsymbol{\beta} - \boldsymbol{F}\boldsymbol{G}^\top\|_F^2 - \gamma\, \mathrm{Tr}(\boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \boldsymbol{L} \bar{\boldsymbol{H}} \boldsymbol{\beta}) \\ s.t. \quad & \boldsymbol{\beta}^\top \boldsymbol{\beta} = \boldsymbol{I}_{n_o} \end{aligned} \tag{6}$$

where $\boldsymbol{L}$ is the *graph Laplacian* as defined in Section 2.1, $\boldsymbol{I}_{n_o}$ is the identity matrix of dimension n_o, and $\lambda > 0$ and $\gamma > 0$ are the user-defined trade-off parameters controlling impacts of minimizing within-class variance and preserving manifold structure, respectively. The hidden layer output matrix are transformed to have zero mean, and we denote the centered output matrix from the hidden layer by $\bar{\boldsymbol{H}} = (\boldsymbol{I}_N - \frac{1}{N}\boldsymbol{1}_N\boldsymbol{1}_N^\top)\boldsymbol{H}$, where $\boldsymbol{I}_N$ is the identity matrix of dimension N, and $\boldsymbol{1}_N$ is a vector of dimension N with all its entries equal to one. Therefore, the output of the neural network, given as $\bar{\boldsymbol{H}}\boldsymbol{\beta}$, is the embedding of the data $\boldsymbol{E}$ in the output space.

The first term in the objective function maximizes the total variance of data in the ELM output space. The second term minimizes the squared sum of distances of data points to their respective cluster centroids in the ELM output space. Together, the first and second terms aim to maximize class separability of the data in the embedded space, similar to DEC. The third term is introduced under the manifold regularization. As explained in Section 2.1, the conditional probabilities are approximated by the predicted class in a supervised classification task. In an embedding task, we approximate the conditional probabilities using the data points in the ELM output space. Following the smoothness assumption, the embedding that preserves the manifold structure of the data should be favored.

Formulation (6) is not jointly convex with respect to all the variables. We optimize this objective function alternately between two groups of variables:

1) Fixing $\boldsymbol{\beta}$ and $\boldsymbol{G}$, and optimizing $\boldsymbol{F}$. Since $\boldsymbol{F}$ is constrained to be an indicator, the optimal $\boldsymbol{F}$ is

$$\boldsymbol{F}_{ij} = \begin{cases} 1, & j = \operatorname{argmin}_k \ \|\boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta} - \boldsymbol{g}_k^\top\|_F^2 \\ 0, & \text{otherwise.} \end{cases}, \quad \text{for } i = 1, \ldots, N. \tag{7}$$

Using this updating formula, the solution goes quickly to the local optimum. Therefore, we follow the update rule in DEC to alleviate the local optimum problem: (1) run k-means on the embedded data $\bar{\boldsymbol{H}}\boldsymbol{\beta}$ with the initial cluster centroid equal to $\boldsymbol{G}$. The resulting cluster indicator $\boldsymbol{F}^{(b)}$ and the value of k-means objective function $o^{(b)}$ are saved as the baselines; (2) run k-means on the embedded data $\bar{\boldsymbol{H}}\boldsymbol{\beta}$ for 20 times with randomly generated initialization. If the ith obtained objective function reaches a lower value than the baseline, i.e., $o^{(i)} < o^{(b)}$, we update the cluster indicator matrix using the corresponding results $\boldsymbol{F}^{(i)}$; otherwise, the baseline solution $\boldsymbol{F}^{(b)}$ is retained.

2) Fixing $\boldsymbol{F}$, and optimizing $\boldsymbol{\beta}$ and $\boldsymbol{G}$. The objective function becomes

$$L(\boldsymbol{\beta}, \boldsymbol{G}) = \mathrm{Tr}(\boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \bar{\boldsymbol{H}}\boldsymbol{\beta}) - \lambda \|\bar{\boldsymbol{H}}\boldsymbol{\beta} - \boldsymbol{F}\boldsymbol{G}^\top\|_F^2 - \gamma\, \mathrm{Tr}(\boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \boldsymbol{L}\bar{\boldsymbol{H}}\boldsymbol{\beta}). \tag{8}$$

Taking the derivative with respect to $\boldsymbol{G}$ first and set it to zero,

$$\frac{\partial L}{\partial \boldsymbol{G}} = -2\lambda(\boldsymbol{G}\boldsymbol{F}^\top\boldsymbol{F} - \boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \boldsymbol{F}) = 0, \tag{9}$$

we have

$$\boldsymbol{G} = \boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \boldsymbol{F}(\boldsymbol{F}^\top\boldsymbol{F})^{-1}. \tag{10}$$

Substituting $\boldsymbol{G}$ in Eq. (10) into Eq. (8), $L(\boldsymbol{\beta}, \boldsymbol{G})$ reduces to $L(\boldsymbol{\beta})$ as follows:

$$\begin{aligned} L(\boldsymbol{\beta}) &= \mathrm{Tr}(\boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \bar{\boldsymbol{H}}\boldsymbol{\beta}) - \lambda \|\bar{\boldsymbol{H}}\boldsymbol{\beta} - \boldsymbol{F}(\boldsymbol{F}^\top\boldsymbol{F})^{-1}\boldsymbol{F}^\top \bar{\boldsymbol{H}}\boldsymbol{\beta}\|_F^2 \\ &\quad - \gamma\, \mathrm{Tr}(\boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \boldsymbol{L}\bar{\boldsymbol{H}}\boldsymbol{\beta}) \\ &= \mathrm{Tr}(\boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \bar{\boldsymbol{H}}\boldsymbol{\beta}) - \lambda\, \mathrm{Tr}(\bar{\boldsymbol{H}}\boldsymbol{\beta} - \boldsymbol{F}(\boldsymbol{F}^\top\boldsymbol{F})^{-1}\boldsymbol{F}^\top \bar{\boldsymbol{H}}\boldsymbol{\beta})^\top \\ &\quad \cdot (\bar{\boldsymbol{H}}\boldsymbol{\beta} - \boldsymbol{F}(\boldsymbol{F}^\top\boldsymbol{F})^{-1}\boldsymbol{F}^\top \bar{\boldsymbol{H}}\boldsymbol{\beta}) - \gamma\, \mathrm{Tr}(\boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \boldsymbol{L}\bar{\boldsymbol{H}}\boldsymbol{\beta}) \\ &= Tr\left[\boldsymbol{\beta}^\top (\bar{\boldsymbol{H}}^\top \bar{\boldsymbol{H}} - \lambda \bar{\boldsymbol{H}}^\top \bar{\boldsymbol{H}} + \lambda \bar{\boldsymbol{H}}^\top \boldsymbol{F}(\boldsymbol{F}^\top\boldsymbol{F})^{-1}\boldsymbol{F}^\top \bar{\boldsymbol{H}} - \gamma \bar{\boldsymbol{H}}^\top \boldsymbol{L}\bar{\boldsymbol{H}})\boldsymbol{\beta}\right] \\ &= \mathrm{Tr}(\boldsymbol{\beta}^\top \bar{\boldsymbol{H}}^\top \boldsymbol{M}\bar{\boldsymbol{H}}\boldsymbol{\beta}), \end{aligned} \tag{11}$$

where $\boldsymbol{M} = (1-\lambda)\boldsymbol{I} + \lambda\boldsymbol{F}(\boldsymbol{F}^\top\boldsymbol{F})^{-1}\boldsymbol{F}^\top - \gamma\boldsymbol{L}$. So the optimization problem with respect to $\boldsymbol{\beta}$ becomes

$$\begin{aligned} &\max_{\boldsymbol{\beta}} \quad L(\boldsymbol{\beta}) \\ &s.t. \quad \boldsymbol{\beta}^\top \boldsymbol{\beta} = \boldsymbol{I}_{n_o}. \end{aligned} \tag{12}$$

Notice that the maximum value of the objective function $L(\boldsymbol{\beta})$ is infinite if β can take an arbitrarily large value. Therefore, we need the constraint on the norm of $\boldsymbol{\beta}$. The global optimal solution to (12) can be derived by selecting the n_o normalized eigenvectors corresponding to the n_o largest eigenvalues of the eigenvalue problem:

$$\bar{\boldsymbol{H}}^\top \boldsymbol{M}\bar{\boldsymbol{H}}\boldsymbol{v} = \mu\boldsymbol{v}. \tag{13}$$

Let $\mu_1, \mu_2, \ldots, \mu_{n_o}$ be the n_o largest eigenvalues of (13) and $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{n_o}$ be the corresponding normalized eigenvectors. Then, the solution to the output weights is obtained by

$$\boldsymbol{\beta}^* = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{n_o}]. \tag{14}$$

Regarding the initialization of the iterative method, we use the data in the original space and a simple clustering method, k-means, to get the initial values for cluster indicator $\boldsymbol{F}$. We then initialize the output weights $\boldsymbol{\beta}$ using Eqs. (13) and (14) and initialize the cluster centroid matrix $\boldsymbol{G}$ using Eq. (10). We summarize the proposed ELM-JEC in Algorithm 1.

Algorithm 1 The ELM-JEC algorithm.

```
Input: The training data: $\boldsymbol{X}$; the number of clusters: $c$; the trade-off parameters: $\lambda$, $\gamma$.
Output: The embedding in a $n_o$-dimensional space: $\boldsymbol{E} \in \mathbb{R}^{N \times n_o}$;
        The cluster indicator matrix: $\boldsymbol{F}$.
  Initialization:
  Step 1: Initialize an ELM network with $n_h$ hidden neurons and calculate the centered output matrix of hidden layers $\bar{\boldsymbol{H}}$.
  Step 2: Run $k$-means on the training data $\boldsymbol{X}$ to get the initial cluster indicator matrix $\boldsymbol{F}$.
  Step 3: Initialize the output weights $\boldsymbol{\beta}$ using Eqs. (13) and (14).
  Step 4: Initialize the cluster centroid matrix $\boldsymbol{G}$ using Eq. (10).
  Update:
  Step 5: Run $k$-means with current cluster centroid matrix $\boldsymbol{G}$ and record the baseline cluster indicator matrix $\boldsymbol{F}^{(b)}$ and the baseline value $o^{(b)}$ of the $k$-means objective function. Run $k$-means with 20 randomly generated initialization. If the $i$th obtained objective function value is smaller than the baseline value, i.e., $o^{(i)} < o^{(b)}$, update the cluster indicator using $\boldsymbol{F}^{(i)}$; otherwise, update using $\boldsymbol{F}^{(b)}$.
  Step 6: Update output weights $\boldsymbol{\beta}$ using Eqs. (13) and (14)
  Step 7: Update the cluster centroid matrix $\boldsymbol{G}$ using Eq. (10).
  Step 8: Go to Step 5 until a maximum allowable number of iterations is reached or the label matrix $\boldsymbol{F}$ is the same as that in the previous iteration, which is the early stopping criterion.
  Output:
  Step 9: Calculate the embedding matrix $\boldsymbol{E} = \bar{\boldsymbol{H}}\boldsymbol{\beta}$.
Return: $\boldsymbol{E}$ as the embedding of the data; $\boldsymbol{F}$ as the cluster indication matrix.
```

Table 3
Specification of real-world datasets.

Data sets	# clusters	# features	# observations
Iris	3	4	150
Ecoli	8	7	336
Diabetes	2	8	768
Segment	7	19	2310
Isolet	26	617	7797
Pollen	7	625	630
UMIST	20	644	575
COIL20	20	1024	1440
ORL	40	1024	400

Table 4
Clustering accuracy of the proposed ELM-JEC and the related algorithms.

Data set	k-means	LDA-KM	ELMC^{LDA}	US-ELM	DEC	ELM-JEC
Iris	76.13 ± 16.18	88.80 ± 16.58	81.60 ± 12.70	91.07 ± 8.38	95.33 ± 0	**97.20 ± 0.28**
Ecoli	53.93 ± 4.75	54.88 ± 6.71	58.48 ± 6.92	61.34 ± 5.98	**62.80 ± 0**	62.23 ± 1.11
Diabetes	66.88 ± 1.25	67.46 ± 0.78	67.71 ± 0.09	65.89 ± 0.06	67.71 ± 0	**74.26 ± 1.11**
Segment	60.67 ± 7.83	**69.66 ± 7.90**	69.64 ± 4.53	66.10 ± 6.22	66.90 ± 0.07	68.95 ± 1.41
Isolet	53.21 ± 3.71	57.22 ± 4.45	54.36 ± 4.23	54.15 ± 4.09	56.30 ± 2.16	**57.63 ± 1.24**
Pollen	46.95 ± 2.16	45.98 ± 4.83	46.97 ± 3.72	48.35 ± 2.81	49.70 ± 0.05	**50.25 ± 1.35**
UMIST	42.31 ± 1.56	42.17 ± 3.52	42.78 ± 3.57	65.11 ± 6.09	44.38 ± 2.02	**68.02 ± 3.01**
COIL20	61.67 ± 4.33	60.15 ± 3.91	59.67 ± 4.21	78.69 ± 4.66	67.48 ± 2.54	**80.79 ± 2.56**
ORL	56.18 ± 3.21	50.38 ± 1.89	48.25 ± 3.11	**59.63 ± 1.96**	57.45 ± 2.66	58.38 ± 2.08

5. Experimental results and discussions

In this section, we empirically compare the performance of the proposed ELM-JEC algorithm with the related embedding and clustering methods on a wide range of real-world datasets including both image and non-image data. We also study the impact of the parameters of the proposed ELM-JEC.

5.1. Datasets and experimental setup

We summarize the datasets in Table 3. The first four are low-dimensional datasets taken from UCI Repository [37]. Isolet[1] is voice data set collected from 26 individuals. Pollen,[2] UMIST,[3] COIL20,[4] and ORL[5] are image datasets [20,23], where the raw pixel values are directly used as input features. Pollen contains pollen grain images, UMIST and ORL are face images datasets, and COIL20 contains 20 types of object images. We use min-max normalization to normalize each feature to the range of [−1, 1].

We used clustering accuracy to measure the performance of each method:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \delta(y_i, map(\hat{y}_i))}{N}. \tag{15}$$

where N is the number of instances, y_i and $\hat{y}_i$ are the true cluster label and the predicted cluster label of pattern x_i, respectively, $\delta(y_i, \hat{y}_i)$ equals to 1 if $y = \hat{y}_i$ and equals to 0 otherwise, $map(\cdot)$ is an optimal permutation function that maps each predicted cluster label to a true cluster label by Hungarian algorithm [38], such that the resulting accuracy is maximized.

We compared the proposed ELM-JEC algorithm with several related algorithms:

(1) k-means [9]. k observations from the data were selected randomly as the initialization of the cluster centroids.
(2) Adaptive LDA-guided k-means clustering (LDA-KM) [18]. This is the representative joint embedding and clustering method.
(3) ELM clustering based on LDA (ELMC^{LDA}) [33]. This is a nonlinear extension of LDA-KM based on the random feature mapping of ELM.
(4) Un-Supervised Extreme Learning Machine (US-ELM) [23].
(5) Discriminative Embedded Clustering (DEC) [20].

For a fair comparison, we standardized the parameter settings, the initialization conditions, and the termination conditions of all methods as follows. The number of hidden neurons in the ELM-based methods was set to 1000 for Iris and 2000 for the other datasets. The sigmoid nonlinear activation function was used. The trade-off parameters in LDA-KM, ELMC^{LDA}, US-ELM, DEC, and ELM-JEC were selected from the same set, i.e., $[2^{-4}, 2^{-3}, \ldots, 2^{4}]$. The graph Laplacian used by US-ELM and ELM-JEC was constructed using the 5-nearest-neighbor graph with binary weights. The dimensionalities of embedded space used in US-ELM, DEC, and ELM-JEC were selected from a set of [2, 4, 8, 16, 32]. Iterative methods, i.e., LDA-KM and ELM-JEC, were initialized with outputs of k-mean and were terminated when there was no change in the predicted labels, or a maximum number of iteration (20 in our experiment) was reached. The updating rules for cluster indicator matrix in ELM-JEC and DEC are the same. We run all algorithms for ten times independently.

5.2. Comparison with the related methods

We reported the averaged clustering accuracy for all the algorithms in Table 4. The results obtained in this study are slightly different from those reported in the respective papers, which is due to the different selection ranges used for user-defined parameters. For a fair comparison, we have standardized the ranges for all the methods in comparison.

LDA-KM and ELMC^{LDA} outperform the baseline method k-means on most datasets; but on image datasets, conducting k-means in the original feature space achieves higher or comparable accuracy. US-ELM achieves better results than DEC on human-face image datasets and object image datasets, showing the nonlinear embedding that preserves manifold structure is desired in complex image clustering tasks. DEC produces better results, compared with US-ELM on most non-image datasets and simple image dataset, i.e., Pollen (compared with the complex shapes of human faces and objects of different categories, pollen grains have relatively simple shapes).

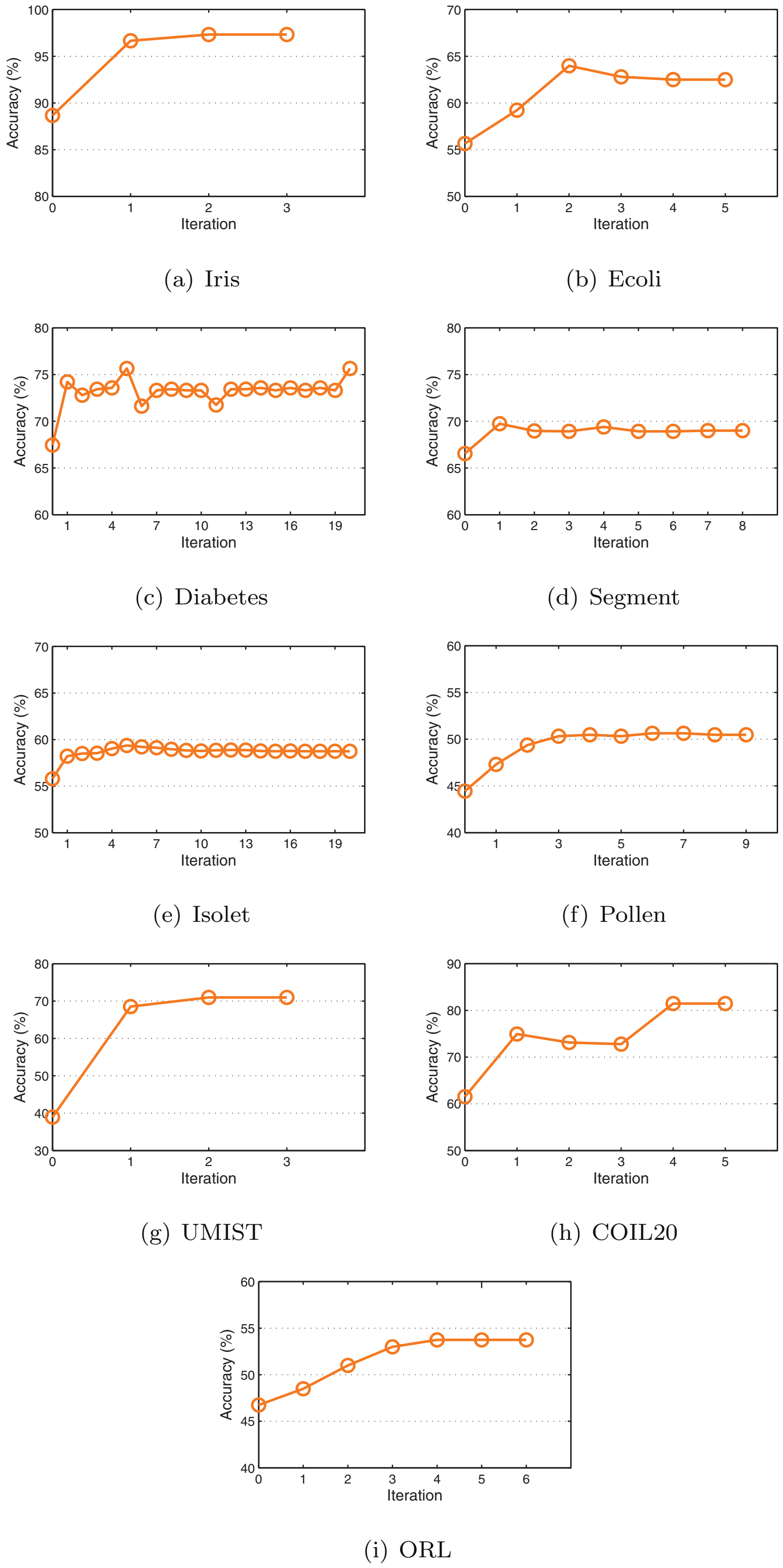
Compared with other methods, ELM-JEC achieves favorable clustering accuracy on most of the datasets. This observation shows the effectiveness of considering both requirements, i.e., preserving the manifold structure in the original space and maximizing the class separability in the output space, during feature learning simultaneously. Moreover, the results of ELM-JEC show smaller standard deviation compared with other ELM-based methods.

5.3. Study on the properties of ELM-JEC

In this section, we further study the properties of ELM-JEC, i.e., the iterative process and the effect of embedding dimension and trade-off parameters on the clustering performance. We hope to provide guidelines for readers to use ELM-JEC.

Iterative process. In this experiment, we study how clustering accuracy changes over the iterations in ELM-JEC. As can be seen

[1] http://archive.ics.uci.edu/ml/datasets/isolet.
[2] http://ome.grc.nia.nih.gov/iicbu2008/pollen/index.html.
[3] http://images.ee.umist.ac.uk/danny/database.html.
[4] http://www1.cs.columbia.edu/cave/research/softlib/coil-20.html.
[5] http://www.uk.research.att.com/facedatabase.html.



(a) Iris

(b) Ecoli

(c) Diabetes

(d) Segment

(e) Isolet

(f) Pollen

(g) UMIST

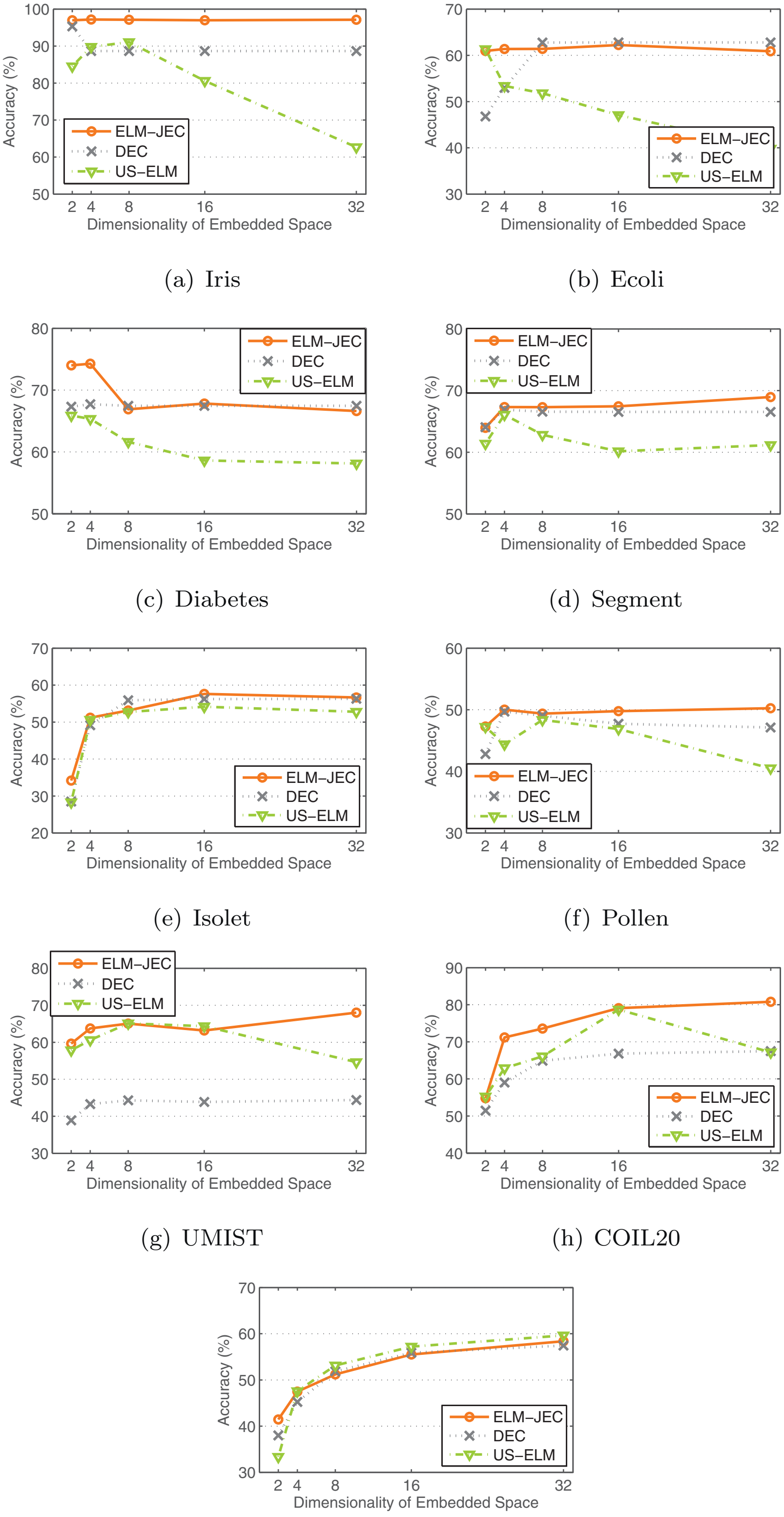
(h) COIL20

(i) ORL

Fig. 1. Evolution of clustering accuracy in ELM-JEC as a function of iterations.

from Fig. 1, ELM-JEC is able to improve the clustering accuracy over iterations in general. For most of the datasets, ELM-JEC stops within a small number of iterations (less than 10). On Diabetes and Isolet, the accuracy fluctuates within a small range after significant improvement is made at the initial iteration. In fact, a significant improvement is usually made within the first one or two iterations.

Embedding dimensions. In this experiment, we study the influence of embedded space dimensions in three embedding meth-



(a) Iris

(b) Ecoli

(c) Diabetes

(d) Segment

(e) Isolet

(f) Pollen

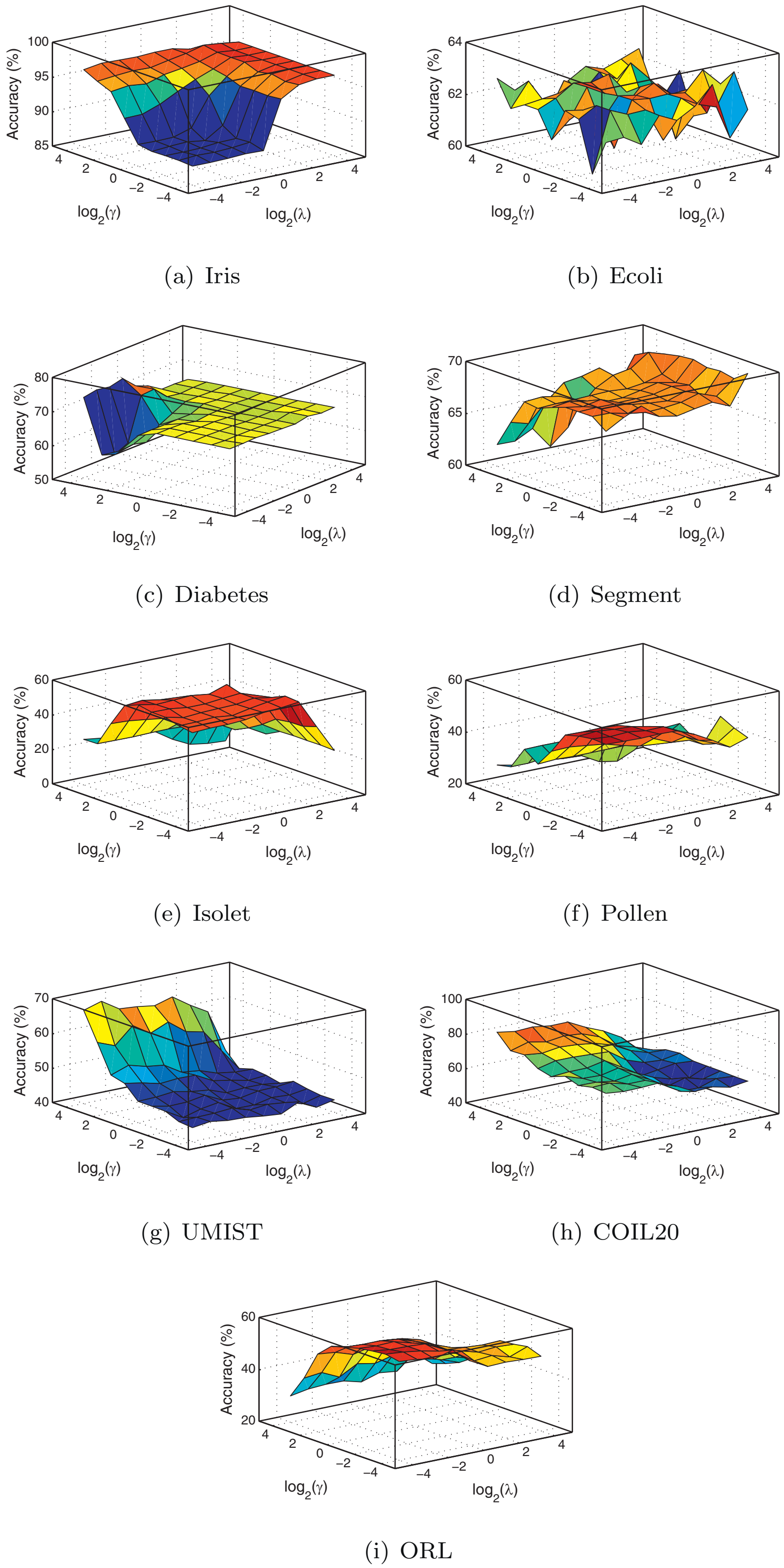
(g) UMIST

(h) COIL20

(i) ORL

Fig. 2. Influence of the embedded space dimension in ELM-JEC, DEC and US-ELM.

ods, i.e., US-ELM, DEC, ELM-JEC. The clustering accuracies using different embedding dimensions of the three methods are reported in Fig. 2. As observed, ELM-JEC and DEC produce satisfactory results using a larger range of dimensions compared with US-ELM. Moreover, our method ELM-JEC produces comparable or even better results than DEC on most datasets. This observation demonstrates that ELM-JEC is stable and will provide competitive performance even when the embedding dimension is not set to the optimal value, which makes it very suitable for applications where specific embedding dimension is required.



(a) Iris

(b) Ecoli

(c) Diabetes

(d) Segment

(e) Isolet

(f) Pollen

(g) UMIST

(h) COIL20

(i) ORL

Fig. 3. Influence of the trade-off parameters γ and λ in ELM-JEC.

Trade-off parameters. We continue to investigate the influence of trade-off parameters used in ELM-JEC. Fig. 3 shows the clustering accuracy of ELM-JEC with different combinations of trade-off parameters values, which are under the respective optimal dimensionality of embedded space. On most of the datasets, high accuracy is observed over a large range of values. However, on Diabetes, Segment, and UMIST, favorable parameters are limited to a small range. On Ecoli, clustering accuracy fluctuates within a small range. Though ELM-JEC produces stable performance over a large range of embedding dimensions, it may require the careful selection of trade-off parameters.

6. Conclusion

In this paper, we investigated the problem of clustering generic data. We first provided insights into the desirable properties for effective embedding and clustering: (1) nonlinearity, (2) structure preserving, (3) separability maximizing, and (4) simultaneous embedding and clustering. We found out that no existing joint feature learning and clustering methods have simultaneously satisfied all the properties. To address this limitation, we proposed an ELM-based methods, i.e., ELM-JEC. Compared with existing joint feature learning and clustering methods, ELM-JEC can model complex and nonlinear embedding functions to achieve better clustering accuracy. Compared with other ELM-based clustering methods, ELM-JEC has the advantage of adopting regularization techniques to improve accuracy and robustness, performing clustering and embedding in the same formulation to achieve clustering-oriented embedding results, and flexibility in selecting the dimension of the embedded space. The experimental results have demonstrated the advantages of ELM-JEC on several real-world datasets, i.e., it generally improves clustering accuracy over iterations, has stable clustering performance over a relatively large range of embedding dimensions, and achieves favorable performance compared with the related methods. As in many alternating optimization based algorithms, how to avoid local optimum and how to develop effective stopping criterion might be interesting subjects of future research.

Acknowledgments

T. Liu's research was supported by Ministry of Education (MOE), Singapore under Academic Research Fund (AcFR) Tier 1 Project RG 80/12 (M4011092).

References

[1] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[2] G. Punj, D.W. Stewart, Cluster analysis in marketing research: review and suggestions for application, J. Mark. Res. 20 (2) (1983) 134–148, doi:10.2307/3151680.

[3] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, in: Proceedings of the KDD Workshop on Text Mining, 2000.

[4] D. Jiang, C. Tang, A. Zhang, Cluster analysis for gene expression data: a survey, IEEE Trans. Knowl. Data Eng. 16 (11) (2004) 1370–1386, doi:10.1109/TKDE.2004.68.

[5] R. Xu, I. Wunsch D., Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678, doi:10.1109/TNN.2005.845141.

[6] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell 22 (8) (2000) 888–905, doi:10.1109/34.868688.

[7] F. R. Bach, M. I. Jordan, Spectral Clustering for Speech Separation, John Wiley & Sons, Ltd, pp. 221–250. doi:10.1002/9780470742044.ch13.

[8] A. Paccanaro, J.A. Casbon, M.A. Saqi, Spectral clustering of protein sequences, Nucl. Acids Res. 34 (5) (2006) 1571–1580, doi:10.1093/nar/gkj515.

[9] J.A. Hartigan, M.A. Wong, A k-means clustering algorithm, J. R. Stat. Soc. Ser. C Appl. Stat. (1979) 100–108.

[10] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[11] S. Wang, X. Zhou, G. Zhang, G. Ji, J. Yang, Z. Zhang, Z. Lu, Y. Zhang, Cluster analysis by variance ratio criterion and quantum-behaved PSO, in: Z. Huang, X. Sun, J. Luo, J. Wang (Eds.), Cloud Computing and Security, Springer International Publishing, Cham, 2015, pp. 285–293, doi:10.1007/978-3-319-27051-7_24.

[12] B. Jarboui, M. Cheikh, P. Siarry, A. Rebai, Combinatorial particle swarm optimization (CPSO) for partitional clustering problem, Appl. Math. Comput. 192 (2) (2007) 337–345, doi:10.1016/j.amc.2007.03.010.

[13] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Proceedings of the Advances in Neural Information Processing Systems, 14, 2001, pp. 585–591.

[14] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Proceedings of the Advances in Neural Information Processing Systems, 14, 2001, pp. 849–856.

[15] L.J. van der Maaten, E.O. Postma, H.J. van den Herik, Dimensionality reduction: a comparative review, J. Mach. Learn. Res. 10 (1–41) (2009) 66–71.

[16] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326, doi:10.1126/science.290.5500.2323.

[17] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323, doi:10.1126/science.290.5500.2319.

[18] C. Ding, T. Li, Adaptive dimension reduction using discriminant analysis and *k*-means clustering, in: Proceedings of the International Conference on Machine Learning (ICML), 2007, pp. 521–528.

[19] J. Ye, Z. Zhao, M. Wu, Discriminative k-means for clustering., in: Proceedings of the Advances in Neural Information Processing Systems, 7, 2007, pp. 1649–1656.

[20] C. Hou, F. Nie, D. Yi, D. Tao, Discriminative embedded clustering: a framework for grouping high-dimensional data, IEEE Trans. Neural Netw. Learn. Syst. 26 (6) (2015) 1287–1299, doi:10.1109/TNNLS.2014.2337335.

[21] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (13) (2006) 489–501, doi:10.1016/j.neucom.2005.12.126.

[22] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man, Cybern. B, Cybern. 42 (2) (2012) 513–529, doi:10.1109/TSMCB.2011.2168604.

[23] G. Huang, S. Song, J.N.D. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (12) (2014) 2405–2417, doi:10.1109/TCYB.2014.2307349.

[24] J. Cao, K. Zhang, M. Luo, C. Yin, X. Lai, Extreme learning machine and adaptive sparse representation for image classification, Neural Netw. 81 (2016) 91–102, doi:10.1016/j.neunet.2016.06.001.

[25] J. Cao, J. Hao, X. Lai, C.-M. Vong, M. Luo, Ensemble extreme learning machine and sparse representation classification, J. Frankl. Inst. 353 (17) (2016b) 4526–4541, doi:10.1016/j.jfranklin.2016.08.024.

[26] S. Lu, Z. Lu, J. Yang, M. Yang, S. Wang, A pathological brain detection system based on kernel based ELM, Multimed. Tools Appl. (2016) 1–14, doi:10.1007/s11042-016-3559-z.

[27] Z.-L. Sun, T.-M. Choi, K.-F. Au, Y. Yu, Sales forecasting using extreme learning machine with applications in fashion retailing, Decis. Support Syst. 46 (1) (2008) 411–419, doi:10.1016/j.dss.2008.07.009.

[28] J. Cao, W. Wang, J. Wang, R. Wang, Excavation equipment recognition based on novel acoustic statistical features, IEEE Trans. Cybern. PP (99) (2016) 1–13, doi:10.1109/TCYB.2016.2609999.

[29] G.-B. Huang, L. Chen, C.K. Siew, et al., Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.

[30] Q. He, X. Jin, C. Du, F. Zhuang, Z. Shi, Clustering in extreme learning machine feature space, Neurocomputing 128 (2014) 88–95, doi:10.1016/j.neucom.2012.12.063.

[31] Y. Peng, W.-L. Zheng, B.-L. Lu, An unsupervised discriminative extreme learning machine and its applications to data clustering, Neurocomputing 174, Part A (2016) 250–264, doi:10.1016/j.neucom.2014.11.097.

[32] C.K. Liyanaarachchi Lekamalage, T. Liu, Y. Yang, Z. Lin, G.-B. Huang, Extreme Learning Machine for Clustering, Springer International Publishing, Cham, pp. 435–444. doi:10.1007/978-3-319-14063-6_36.

[33] G. Huang, T. Liu, Y. Yang, Z. Lin, S. Song, C. Wu, Discriminative clustering via extreme learning machine, Neural Netw. 70 (2015) 1–8, doi:10.1016/j.neunet.2015.06.002.

[34] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.

[35] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, 2012.

[36] G. Huang, G.-B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Netw. 61 (2015) 32–48, doi:10.1016/j.neunet.2014.10.001.

[37] K. Bache, M. Lichman, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2013 http://archive.ics.uci.edu/ml.

[38] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Courier Corporation, 1998.

Tianchi Liu received the B.Eng. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2013, where she is currently working toward the Ph.D. degree. Her current research interests include extreme learning machine, semi-supervised learning, and unsupervised learning.

Liyanaarachchi Lekamalage Chamara Kasun received B.Sc degree from Sri Lanka Institute of Information Technology, Sri Lanka, and M.Sc degree from Nanyang Technological University, Singapore, in 2008 and 2010, respectively. He is currently working toward the Ph.D. degree in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include extreme learning machines, multi layer neural networks, and dimension reduction.

Guang-Bin Huang (M'98-SM'04) received the B.Sc degree in applied mathematics and M.Eng. degree in computer engineering from Northeastern University, PR China, in 1991 and 1994, respectively, and Ph.D degree in electrical engineering from Nanyang Technological University, Singapore in 1999. During undergraduate period, he also concurrently studied in Applied Mathematics department and Wireless Communication department of Northeastern University, PR China. He is now Professor at Nanyang Technological University, Singapore. He serves as an Associate Editor of Neurocomputing, Neural Networks, Cognitive Computation, and IEEE Transactions on Cybernetics. He is a senior member of IEEE. His current research interests include machine learning, computational intelligence, and extreme learning machines.

Zhiping Lin received the B.Eng. degree in control engineering from South China Institute of Technology, Canton, China in 1982 and the Ph.D. degree in information engineering from the University of Cambridge, England in 1987. Since February 1999, he has been an Associate Professor at Nanyang Technological University, Singapore. Dr. Lin served as the Editor-in-Chief of Multidimensional Systems and Signal Processing for 2011–2015; as an Associate Editor of Circuits, Systems and Signal Processing for 2000–2007 and IEEE Transactions on Circuits and Systems, Part II, for 2010–2011, and also a reviewer for Mathematical Reviews for 2011–2013. Currently he is serving as Subject Editor and Guest Editor of the Journal of Franklin Institute. His research interests include multidimensional systems and signal processing, statistical and biomedical signal processing, and machine learning. He is a co-author of the 2007 Young Author Best Paper Award from the IEEE Signal Processing Society, Distinguished Lecturer of the IEEE Circuits and Systems Society for 2007–2008, and the Chair of the IEEE Circuits and Systems Singapore Chapter for 2007–2008.