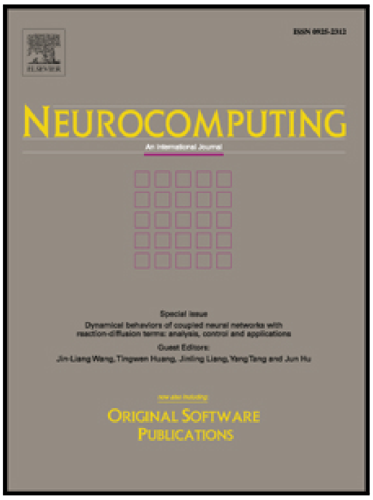
Communicated by A. Abraham


ISSN 0925-2312
ELSEVIER
NEUROCOMPUTING
An International Journal
Special issue
Dynamical behaviors of coupled neural networks with reaction-diffusion terms: analysis, control and applications
Guest Editors:
Jin-Liang Wang, Tingwen Huang, Jinling Liang, Yang Tang and Jun Hu
now also including
ORIGINAL SOFTWARE PUBLICATIONS

Local Kernel Alignment Based Multi-view Clustering Using Extreme Learning Machine

Qiang Wang, Yong Dou, Xinwang Liu, Fei Xia, Qi Lv, Ke Yang

PII: S0925-2312(17)31579-5
DOI: 10.1016/j.neucom.2017.09.060
Reference: NEUCOM 18942

To appear in: *Neurocomputing*

Received date: 1 July 2016
Revised date: 14 September 2017
Accepted date: 18 September 2017

Please cite this article as: Qiang Wang, Yong Dou, Xinwang Liu, Fei Xia, Qi Lv, Ke Yang, Local Kernel Alignment Based Multi-view Clustering Using Extreme Learning Machine, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2017.09.060

Local Kernel Alignment Based Multi-view Clustering Using Extreme Learning Machine

Qiang Wang[a,b,*], Yong Dou[a,b], Xinwang Liu[a,b], Fei Xia[c,*], Qi Lv[a,b], Ke Yang[a,b]

[a] *National Laboratory for Parallel and Distributed Processing,*
National University of Defense Technology, Changsha, China
[b] *College of Computer, National University of Defense Technology, Changsha, China*
[c] *Electronic Engineering College, Naval University of Engineering, Wuhan, China*

Abstract

A similarity or dissimilarity measure, such as the Euclidean distance, is crucial to discriminative clustering algorithms. These measures used to calculate pairwise similarities between samples rely on data representations in a feature space. However, discriminative clustering fails if the samples in a feature space are linearly inseparable. This problem can be solved by performing a nonlinear data transformation into a high dimensional feature space, which can increase the probability of the linear separability of the samples within the transformed feature space and simplify the associated data structure. Mercer kernels, which are constructed using such a nonlinear data transformation, have been widely used in clustering tasks. Extreme learning machine (ELM) is a new method that exhibits promising clustering performance owing to its universal approximation capability, easy parameter selection, explicit feature mapping process, and excellent feature representation capability. This study proposes an ELM based multi-view learning approach with different views generated by ELM random feature mapping with respect to different hidden-layer nodes, and exploits the properties of these views. Experiments show that better clustering results can be obtained by combining these views together compared with the correspond-

[*] Corresponding author
Email addresses: qiangwang@nudt.edu.cn (Qiang Wang), xcyphoenix@nudt.edu.cn (Fei Xia)

ing ELM-based single-view clustering methods and the traditional algorithms which are performed in the feature space of the original data. Moreover, local kernel alignment property is widespread in these views. This alignment helps the clustering algorithm focus on closer sample pairs. This study also proposes an ELM based multiple kernel clustering algorithm with local kernel alignment maximization. The proposed algorithm is experimentally demonstrated on 10 single-view benchmark datasets and yields superior clustering performance when compared with the state-of-the-art multi-view clustering methods in recent literatures. Thus, the effectiveness and superiority of maximizing local kernel alignment on those views constructed by the proposed method are verified.

Keywords: Multi-view clustering, Extreme learning machine, Local kernel alignment.

1. Introduction

Clustering [1] partitions a set of samples into groups (called clusters) based on their similarities. Samples that belong to the same cluster are similar, whereas samples from different clusters are distinct. Clustering problems can be categorized into generative and discriminative approaches. Generative approaches attempt to learn generative models from samples by maximizing the probability of the generation of samples. Each model represents one cluster. By contrast, discriminative approaches attempt to cluster samples by using their pairwise similarities, aiming at maximizing within-cluster similarities and minimizing between-cluster similarities. Discriminative approaches do not make restrict assumptions about the samples. Thus, these methods are more robust than the generative approaches. Moreover, obtaining a generative model in many application domains is difficult [2]. Therefore, discriminative clustering algorithms have been widely used in real applications, such as information retrieval [3], pattern recognition [4] and data mining [5]. Typical discriminative clustering algorithms include k-means clustering [6], spectral clustering [7] and kernel k-means clustering [8].

A similarity or dissimilarity measure [9], such as the Euclidean distance, is crucial to discriminative clustering algorithms. However, these measures used to calculate the pairwise similarities between samples rely on the data representations in a feature space. That is, the feature space of data is crucial for final clustering performance. Discriminative clustering algorithms are effective for data with ellipsoidal or hyper-spherical distribution. However, these algorithms fail if samples in a feature space are nonlinearly separable. This problem can be solved by performing clustering within the feature space generated by a nonlinear data transformation process. This process maps samples in the linearly inseparable feature space to a high-dimensional feature space, and can be implemented by a kernel matrix or the random feature mapping of ELM [10]. The eigenvectors of a kernel matrix, which defines the implicit mapping, provides a means to estimate the number of clusters inherent within the data, and a simple computational iterative procedure is presented for the subsequent feature space partitioning of the data [11]. Differently, the feature mapping of ELM is explicit. Moreovers, the feature mapping of ELM can satisfy the universal approximation condition. That is, ELM can approximate any continuous function using a linear function in the feature space of ELM.

ELM was originally proposed by Huang et al. [10] and has been widely used in regression and classification [12–14] owing to its universal approximation capability, easy parameter selection, explicit feature mapping process, and excellent feature representation capability. Recently, ELM has been extended for clustering. In [15, 16], the authors proposed to perform clustering in the embedding space obtained by ELM random feature mapping. Benefitting from ELM's universal approximation capability[17], the data structure becomes much simpler after the ELM feature transformation process. Thus, this approach is convenient for implementation and is efficient for computation. Moreover, the approach only needs an ELM feature mapping process, which is simpler than the kernel-based feature mapping methods. The unsupervised ELM (US-ELM) algorithm [18] proposed by Huang et al. can capture the manifold structure of the output weight of ELM, which performs well on datasets with manifold property.

Huang et al. [19] extended ELM to discriminative clustering and proposed three novel clustering methods based on weighted ELM (W-ELM), Fisher's linear discriminant analysis, and kernel k-means. All methods mentioned previously are conducted on the single-view dataset.

In many practical applications, data has multiple feature representations (i.e., views) from different data sources [20], which usually contain complementary and compatible information, that are helpful for clustering. For example, flowers in the clustering task of Oxford Flower17 [21], can be represented by different features, such as color, shape, and texture. Excellent clustering performance can be achieved by combining these views together, which is known as multi-view learning [22]. Efforts have been devoted on extending multi-view learning to clustering. These works can be divided into three categories based on their combination strategy, as follows: (1) Early integration directly incorporates multi-view features into a common representation before the clustering process by optimizing certain loss functions[23–27]. (2) Intermediate integration conducts clustering in the subspace generated by projecting a multi-view dataset onto a common low-dimensional feature space [28]. (3) Late integration combines the intermediate outputs on each view separately to obtain a final consensus clustering solution [29] [30].

In [16][31], clustering performance in the feature space of ELM is sensitive to the number of ELM hidden layer nodes. Thus, selecting the reasonable nodes that can yield the best clustering results for different applications is difficult. We take inspiration from multi-view clustering mentioned previously, where clear gains on datasets have resulted from integrating complementary and compatible information of different views. This study proposes to construct views using ELM random feature mapping with respect to different hidden-layer nodes. And these views are used for existing multi-view clustering algorithms to further improve the clustering performance. Moreover, motivated by the work of [32], we exploit the properties of kernel alignment on the constructed views. Local kernel alignment, which only requires that the similarity of a sample to its k-nearest neighbors be aligned with the ideal similarity matrix, helps improve clustering. The contributions of this study can be summarized as follows:

(a) This study proposes a general multi-view clustering approach which combines multiple views generated by ELM random feature mapping on original single-view dataset. Each view corresponds to a different value of hidden-layer nodes.

(b) This study exploits the properties of the local kernel alignment of constructed views and proposes an ELM-based multiple kernel clustering algorithm with local kernel alignment maximization.

(c) This study presents state-of-the-art clustering results on 10 datasets, including two datasets used in practical face recognition tasks.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of primary discriminative clustering methods and ELM. Section 3 presents the proposed multi-view clustering method. Section 4 shows the extensive experimental results. Finally, Section 5 concludes this paper.

2. Preliminaries

This section reviews several classical clustering algorithms which are the basis of multi-view clustering algorithms used in this study, and introduces the ELM theory which has inspired this work.

2.1. K-means clustering

Let $\{\mathbf{x}_i\}_{i=1}^n \in \mathcal{X}$ be a collection of n samples. The k-means [6][33][34] clustering algorithm partitions samples into k disjoint clusters. Typically $k \ll n$ and each sample $\mathbf{x}_i$ is a vector of dimension d ($\mathbf{x}_i \in \mathbf{R}^d$). And samples within a cluster are more similar than samples from different clusters. The intra-cluster variance can be measured by the squared Euclidean distance between the center of a cluster and the samples in the cluster. The mean value of cluster c_j can be formulated as follows:

$$\mu_k = \frac{\sum_{i=1}^n G_{ij}\mathbf{x}_i}{N_j} \quad s.t.\ G_{ij} = \begin{cases} 1, & if\ \mathbf{x}_i \in c_j \\ 0, & otherwise \end{cases} \tag{1}$$

where $\{G_{ik}\}_{i,j=1}^{n,k} \in \mathbf{G}$ is the cluster indicator and N_j is the number of samples in $j-$th $(1 \leq j \leq k)$ cluster.

K-means clustering aims to minimize the sum of the intra-cluster variances, which can be formulated as the following optimization problem:

$$J(C) = \sum_{j=1}^{k} J(c_j) = \sum_{j=1}^{k} \sum_{i=1}^{n} G_{ij} \|\mathbf{x}_i - \mu_j\|^2 \tag{2}$$

where $J(c_j) = \sum_{i=1}^{n} G_{ij} \|\mathbf{x}_i - \mu_j\|^2$ is the variance between μ_j and the samples in c_j. K-means clustering solves the objective function of Eq. (2) iteratively. In each iteration, it assigns samples to clusters whose mean value yields the least intra-cluster variances. The formula is expressed as follows:

$$c_i^t = \{\mathbf{x}_p : \|\mathbf{x}_p - \mu_i^t\| \leq \|\mathbf{x}_p - \mu_j^t\| \ \ \forall i, j, 1 \leq i, j \leq k\}, \tag{3}$$

where each sample $\mathbf{x}_p$ can be assigned to only one cluster c_i in the $t-$th iteration, even if it could be assigned to two or more clusters. That is, $\sum_{j=1}^{k} G_{ij} = 1$.

K-means clustering first initializes the centers randomly. Then, it assigns samples to their closest clusters to reduce intra-cluster variances. The cluster centers are later determined by recomputing the mean values of the samples for each cluster. Finally, k-means converges to a local minimum monotonically when no sample is reassigned from one cluster to another. The intra-cluster variances decrease dramatically given an increase in the number of clusters k (with $J(C) = 0$ when $k = n$). Thus, the objective function of Eq. (2) can be minimized for a fixed number of clusters (i.e., $k \ll n$). K-means clustering algorithm can be summarized in Algorithm 1.

2.2. Spectral clustering

Spectral clustering [35][7] partitions samples $\{\mathbf{x}_i\}_{i=1}^{n} \in \mathcal{X}$ into k different clusters by exploiting the properties of the Laplacian of the undirected weighted graph $Graph = \{\mathcal{X}, \mathbf{E}, \mathbf{K}\}$. $\mathcal{X}$ denotes the vertex set, $\mathbf{E}$ is the edge set that denotes the connected vertices, and $\mathbf{K} \in \mathbf{R}^{n \times n}$ is an edge affinity matrix that denotes the pairwise similarities between samples (vertices in graph). $\mathbf{K}$ is a

```
Algorithm 1 K-means Clustering Algorithm (KM)
Input: Given a dataset $\mathcal{X}$ with $n$ samples, $k$ clusters.
Output: Cluster indicator $\mathbf{G}$.
1: Initialize $k$ cluster centers.
2: Assign each sample to its closest cluster which satisfies Eq. (3).
3: Compute new cluster centers using Eq. (2).
4: Repeat steps 2 and 3 until convergence has been reached, and return the
   cluster indicator $\mathbf{G}$.
```

symmetric matrix and Gaussian kernel is used to construct the matrix, which can be formulated as follows:

$$K_{ij} = \begin{cases} exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}), & \mathbf{x}_i \; and \; \mathbf{x}_j \; are \; neighbors \\ 0, & otherwise \end{cases} \quad s.t. \;\; 1 \leq i, j \leq n \qquad (4)$$

where the standard deviation (parameter σ) controls the spread of neighbors in the $k-$nearest graph. An affinity matrix $\mathbf{K}$ is used to calculate the eigenvectors. The top k eigenvectors of the normalized graph Laplacian are the relaxations of the cluster indicator vector $\mathbf{G}$ that assign each node in the graph to one of the k clusters.

According to [36], the minimization of the normalized cut criterion can be formulated as follows:

$$\max_{\mathbf{Z}^\top \mathbf{D}\mathbf{Z} = \mathbf{I}} tr(\mathbf{Z}^\top \mathbf{K}\mathbf{Z}) \qquad (5)$$

where $\mathbf{Z} = \mathbf{G}(\mathbf{G}^\top \mathbf{D}\mathbf{G})^{-1/2}$ is a scaled partition matrix, $\mathbf{D}$ is a diagonal matrix with diagonal elements as $D_{ii} = \sum_j K_{ij}$.

Eq. (5) can be rewritten as follows:

$$\max_{\mathbf{F}^\top \mathbf{F} = \mathbf{I}} tr(\mathbf{F}^\top \mathbf{D}^{-1/2}\mathbf{K}\mathbf{D}^{-1/2}\mathbf{F}) \qquad (6)$$

where $\mathbf{F} = \mathbf{D}^{1/2}\mathbf{Z} = \mathbf{D}^{1/2}\mathbf{G}(\mathbf{G}^\top \mathbf{D}\mathbf{G})^{-1/2}$ is a scaled cluster assignment matrix. The elements of $\mathbf{F}$ are constrained to be discrete values. Thus, a solution, which relaxes the matrix $\mathbf{F}$ from the desired discrete values to the continuous values, is used to optimize the challenging problem of Eq. (6). The problem can be transformed into the following optimization problem:

$$\max_{\mathbf{F}^\top\mathbf{F}=\mathbf{I}} tr(\mathbf{F}^\top \mathcal{L}\mathbf{F}) \tag{7}$$

where $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{K}\mathbf{D}^{-1/2}$ is the normalized graph Laplacian. The optimization problem Eq. (7) can be solved by the eigenvalue decomposition of the matrix $\mathcal{L}$. The final discrete solution can be obtained using k-means or spectral rotation based on the relaxed continuous solution.

Spectral clustering exhibits excellent clustering performance on arbitrary shaped clusters, and can be theoretically derived. Various variants of the spectral approach have been proposed and the typical algorithm proposed by Ng et al. [37] can be summarized in Algorithm 2.

Algorithm 2 Spectral Clustering Algorithm (SC)

```
Input: Given a dataset $\mathcal{X}$ with $n$ samples, $k$ clusters.
Output: Cluster indicator $\mathbf{G}$.
1: Construct a $n \times n$ positive semi-definite similarity matrix $\mathbf{K}$ according to Eq. (4).
2: Define the diagonal matrix $\mathbf{D}$.
3: Compute the normalized graph Laplacian $\mathcal{L}$.
4: Calculate the eigenvectors of $\mathcal{L}$, and stack the top $k$ eigenvectors in column to form the $n \times k$ matrix $\mathbf{U}$ .
5: Normalize each row of $\mathbf{U}$ to unit length and form matrix $\mathbf{V}$, where $V_{ij} = U_{ij}/(\sum_{r=1}^{k} U_{ir}^2)^{1/2}$.
6: Treat each row of $\mathbf{V}$ as a point and partition them into $k$ clusters using k-means clustering.
7: Assign sample $\mathbf{x}_i$ to cluster $c_m$ only if the $i$-th row of matrix $\mathbf{V}$ is assigned to the $m-$th cluster, and return the cluster indicator $\mathbf{G}$ .
```

2.3. Kernel k-means clustering

Compared with the traditional k-means clustering, kernel k-means [8] maps the samples $\mathcal{X}$ onto a reproducing kernel Hilbert space $\mathcal{H}$ via a nonlinear transformation $\phi : \mathcal{X} \rightarrow \mathcal{H}$.

The optimization problem in Eq. (2) can be equivalently rewritten as the following matrix-vector form:

$$\text{minmize } J(C) = \text{minimize } tr((\Phi - \mathbf{M})^\top (\Phi - \mathbf{M})) \tag{8}$$

where $\Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_n)]$, $\mathbf{M} = \Phi\mathbf{G}\mathbf{N}\mathbf{G}^\top$ and $\mathbf{N} = diag(N_1^{-1}, N_2^{-1}, \cdots, N_k^{-1})$. Substituting $\mathbf{M}$ to Eq. (8), the optimization problem can be rewritten as follows:

$$\begin{aligned} tr((\Phi - \mathbf{M})^\top (\Phi - \mathbf{M})) &= tr((\Phi - \Phi\mathbf{G}\mathbf{N}\mathbf{G}^\top)^\top (\Phi - \Phi\mathbf{G}\mathbf{N}\mathbf{G}^\top)) \\ &= tr((\mathbf{I} - \mathbf{G}\mathbf{N}\mathbf{G}^\top)^\top \Phi^\top \Phi (\mathbf{I} - \mathbf{G}\mathbf{N}\mathbf{G}^\top)) \\ &= tr(\Phi(\mathbf{I} - \mathbf{G}\mathbf{N}\mathbf{G}^\top)(\mathbf{I} - \mathbf{G}\mathbf{N}\mathbf{G}^\top)^\top \Phi^\top) \\ &= tr(\Phi(\mathbf{I} - \mathbf{G}\mathbf{N}\mathbf{G}^\top)^2 \Phi^\top) \\ &= tr(\Phi(\mathbf{I} - \mathbf{G}\mathbf{N}\mathbf{G}^\top)\Phi^\top) \\ &= tr(\Phi\Phi^\top) - tr(\Phi\mathbf{G}\mathbf{N}\mathbf{G}^\top\Phi^\top) \\ &= tr(\Phi\Phi^\top) - tr(\Phi\mathbf{G}\mathbf{N}^{1/2}\mathbf{N}^{1/2}\mathbf{G}^\top\Phi^\top) \\ &= tr(\Phi\Phi^\top) - tr(\mathbf{N}^{1/2}\mathbf{G}^\top\Phi^\top\Phi\mathbf{G}\mathbf{N}^{1/2}) \\ &= tr(\mathbf{K}) - tr(\mathbf{N}^{1/2}\mathbf{G}^\top\mathbf{K}\mathbf{G}\mathbf{N}^{1/2}) \end{aligned} \tag{9}$$

where $tr(\mathbf{AB}) = tr(\mathbf{BA})$ and $(\mathbf{AB})^\top = \mathbf{B}^\top\mathbf{A}^\top$ are used in the deviation. $\mathbf{K}$ is a kernel matrix with $K_{ij} = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$, which can be constructed as in Eq. (4). It should be noted that $\mathbf{G}^\top\mathbf{G} = \mathbf{N}^{-1}$ with the diagonal elements indicating the size of each cluster. $\mathbf{N}^{1/2}$ is defined as taking the square root of the diagonal elements.

The optimization problem of Eq. (9) is very difficult to solve due to the binary cluster indicator matrix $\mathbf{G}$. However, this challenge problem can be approximately solved through a relaxation on this constraint. Specifically, by defining $\mathbf{H} = \mathbf{G}\mathbf{N}^{1/2}$ and letting $\mathbf{H}$ take arbitrary real values subject to orthogonality constraints, we obtain the following relaxation of the above problem:

$$\min_{\mathbf{H}\in\mathbf{R}^{n\times k}} tr\left(\mathbf{K}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right) \quad s.t. \;\; \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{10}$$

where $\mathbf{I}_k$ is an identity matrix with size $k \times k$. The optimal $\mathbf{H}$ for Eq. (10) can be obtained by taking the k eigenvectors that correspond to the k largest eigenvalues of $\mathbf{K}$ [38]. K-means clustering is then performed on the normalized matrix of $\mathbf{H}$ with each row of $\mathbf{H}$ being normalized to the unit length. The sketch of kernel k-means clustering is shown in Algorithm 3.

```
Algorithm 3 Kernel K-means Clustering Algorithm (KKM)
Input: Given a dataset $\mathcal{X}$ with $n$ samples, $k$ clusters.
Output: Cluster indicator $\mathbf{G}$.
1: Initialize $k$ cluster centers.
2: Construct a $n \times n$ positive semi-definite similarity matrix $\mathbf{K}$.
3: Calculate $\mathbf{H}$ by solving Eq. (10).
4: Normalize each row of $\mathbf{H}$ to unit length and form matrix $\tilde{\mathbf{H}}$.
5: Treat each row of $\tilde{\mathbf{H}}$ as a point and perform k-means clustering on it.
6: Assign sample $\mathbf{x}_i$ to cluster $c_m$ only if the $i$-th row of matrix $\tilde{\mathbf{H}}$ is assigned
   to the $m$-th cluster, and return the cluster indicator $\mathbf{G}$.
```

2.4. Extreme learning machine

Let $\{\mathbf{x}_i\}_{i=1}^{n} \in \mathcal{X}$ be a collection of n samples. Here $\mathbf{x}_i \in \mathbf{R}^d$ with d denoting the dimensions of input samples. In ELM theory, ELM projects samples $\mathcal{X}$ in the input space onto a $l-$dimensional feature space using a nonlinear data transformation (i.e., nonlinear activation functions) $h(\mathcal{X})$, where l denotes the number of ELM hidden nodes. The output function of ELM can be expressed as follows:

$$f(\mathcal{X}) = \sum_{i=1}^{l} h_i(\mathcal{X})\beta_i = h(\mathcal{X})\beta \tag{11}$$

where $\beta = [\beta_1, ..., \beta_l]^\top$ denotes the output weight vector that connects the hidden-layer nodes and output nodes, and $h(\mathcal{X}) = [h_1(\mathcal{X}), ..., h_l(\mathcal{X})]$ denotes the output vector of the hidden layer with respect to input $\mathcal{X}$. The activation function $h_i(\mathcal{X})$ in real applications can be formally described as follows:

$$h_i(\mathcal{X}) = G(\mathbf{a_i}, b_i, \mathcal{X}), \qquad \mathbf{a_i} \in \mathbf{R}^d, b_i \in \mathbf{R} \tag{12}$$

where $G(\mathbf{a}, \mathbf{b}, \mathcal{X})$ is a nonlinear piecewise continuous function that satisfies ELM universal approximation capability theorems [39]. Typical activation functions used in ELM are given as follows:

(1)Sigmoid function

$$G(\mathbf{a}, \mathbf{b}, \mathcal{X}) = \frac{1}{1 + exp(-(\mathbf{a}^\top \mathcal{X} + \mathbf{b}))} \tag{13}$$

(2)Gaussian function

$$G(\mathbf{a}, \mathbf{b}, \mathcal{X}) = exp(-\mathbf{b}\|\mathcal{X} - \mathbf{a}\|) \tag{14}$$

where $(\mathbf{a}, \mathbf{b})$ are parameters of the mapping function, and $\| \cdot \|$ denotes the Euclidean norm. Compared with the traditional BP neural networks and SVM, $(\mathbf{a}, \mathbf{b})$ can be randomly generated (independent of the training data) according to any continuous probability distribution instead of being explicitly trained. Thus, the result is remarkably efficiency. The generalization performance of ELM is relative to the hidden-layer nodes of the ELM. Generally speaking, a larger value of hidden nodes usually result in better generalization performance. Details on the selection of hidden nodes of ELM can be found in [39, 40].

3. Multi-view clustering using ELM

This section describes the proposed multi-view clustering method, which combines ELM random feature mapping and multi-view clustering algorithms. First, the main steps of the proposed method and the rationality of ELM feature mapping are given. Next, we exploit the local kernel alignment property on the constructed views generated by ELM and propose a multi-view clustering algorithm using this property.

3.1. ELM-based multi-view clustering approach

This section first describes the proposed ELM-based multi-view clustering approach. Then, the rationality of introducing ELM in the framework is given.

3.1.1. Framework of proposed approach

ELM has been widely used in clustering because it can approximate any target continuous functions. It has been proved that clustering in the feature space generated by ELM random feature mapping can obtain better clustering results than the corresponding Mercer kernel-based methods and traditional algorithms using the original data [16]. The work in [41] showed that performing multi-view clustering on the multi-view datasets with each view transformed to a high-dimensional feature space can further improve clustering performance. Motivated by these works, we propose to construct different views for the original single-view dataset using ELM random feature mapping, and then perform multi-view clustering on these views.

Let $\{\mathbf{x}_i\}_{i=1}^{n} \in \mathcal{X}$ be a collection of n samples, with $\mathbf{x}_i \in \mathbf{R}^d$ denoting the dimensions of input samples. Clustering is performed to partition the n input samples into k clusters. The proposed approach can be summarized in three steps. The details are given as follows.

Step1 : Data normalization. Each feature of original data is normalized to $[-1, 1]$ using min-max normalization to avoid being conditioned by features with a wide range of possible values. Moreover, this process can preserve the range and introduce the dispersion of the series. The original input $\mathcal{X}$ is normalized to $\tilde{\mathcal{X}}$. The normalized value of the $i-$th feature $\mathcal{X}_i (i = 1, \cdots, d)$ can be calculated as follows:

$$\tilde{\mathcal{X}}_i = (nmax - nmin) * \frac{\mathcal{X}_i - min(\mathcal{X}_i)}{max(\mathcal{X}_i) - min(\mathcal{X}_i)} + nmin \tag{15}$$

where $nmax$ and $nmin$ are equal to 1 and -1, respectively. And $\tilde{\mathcal{X}}_i$ denotes the i-th column of the normalized $n \times d$ matrix.

Step2 : Views construction. Views are constructed using ELM random feature mapping with different hidden-layer nodes l. Here, the output matrix $\mathbf{O}$ (i.e., a $n \times l$ matrix) of the hidden layer is regarded as a view. And $\mathbf{O} = [h_1(\tilde{\mathcal{X}}), h_2(\tilde{\mathcal{X}}), \cdots, h_l(\tilde{\mathcal{X}})]^\top$ with $h_i(\tilde{\mathcal{X}})$ denoting the output vector of the hidden-layer of ELM with respect to the i-th hidden node.

Step3 : Multi-view clustering. Multi-view clustering is performed on the constructed views.

3.1.2. The rationality of ELM feature mapping

ELM has many salient features owing to its explicit feature mapping, such as excellent feature representation capability, easy parameter selection and promising performance. Therefore, clustering performance can be further improved by utilizing ELM feature mapping. In ELM theory, ELM can approximate any target function [42] using a widespread type of feature mapping $h(x)$. That is, given any target continuous function $f(x)$, there exists a series of β_i, such that

$$\lim_{l\to+\infty} \|f_l(x) - f(x)\| = \lim_{l\to+\infty} \|\sum_{i=1}^{l} \beta_i h_i(x) - f(x)\| = 0 \tag{16}$$

Eq. (16) shows that ELM can approximate any continuous function using a linear function in the ELM feature space. However, the feature mapping $\phi(x_i)$ may be unknown in the kernel-based algorithms, such as SVM. That is, not every feature mapping used in SVM and its variants satisfy the approximation condition. Obviously, obtaining promising results is difficult for a learning machine with feature mapping, which does not satisfy the universal approximation condition. Huang et al. in [17] proved that ELM can separate any decision regions regardless of the shapes of these regions if $h(x)\beta$ can approximate any continuous functions. That is, data in the ELM transformed feature space are easier to be partitioned by clustering methods than in the original input feature space. Different views can be constructed with respect to different hidden-layer nodes by taking the hidden-layer matrix generated by ELM feature mapping as a data view. Clustering performance can then be improved by conducting existing multi-view clustering methods on these views compared with the methods that performed in the feature space of the original data.

Similar to ELM feature mapping, Mercer kernel can also be used to increase the probability of the linear separability of samples within a feature space generated by nonlinear data transformation. Such a transformation can be implemented by kernel functions, such as Gaussian kernel and polynomial kernel. Many Mercer kernel-based clustering algorithms [11] have been proposed. The feature mapping $\phi(x_i)$ of these Mercer kernel-based methods is implicit unlike the clustering methods in ELM feature mapping spaces [15][16][41]. And it is also not efficient for computation owing to the use of kernel functions, which needs to calculate pairwise similarities between samples. Without the explicit form of feature mapping, feature mapping used in clustering may not satisfy the universal approximation condition, and may adversely affect clustering performance. Thus, explicit feature mapping methods, such as ELM feature mapping, are more appropriate for the proposed framework.

3.2. Instantiation of proposed approach

This section first provides the motivation of our proposed algorithm. And then, a multi-view clustering algorithm is proposed using local kernel alignment, which is based on the proposed clustering approach.

3.2.1. Motivation of the proposed algorithm

As mentioned in Section 3.1, this study proposes a general multi-view clustering approach, which conducts clustering on the views generated by ELM random feature mapping. In our approach, the output hidden-layer matrix of ELM is regarded as a view. The views have similar data distribution because the views are generated by the same nonlinear transformation, such as sigmoid function. That is, these constructed views have compatible information. However, views are generated with respect to different hidden-layer nodes. Thus, they are not the same. It is acknowledged that multi-view clustering method combines complementary and compatible information of multiple views to improve clustering performance. Therefore, existing multi-view clustering algorithms can be conducted on constructed views to obtain satisfactory clustering results.

Existing multi-view clustering algorithms usually perform clustering with the help of kernels. For example, multiple kernel clustering optimizes the integration of a group of pre-specified kernels to improve clustering performance [25][24]. This integration can be implemented by forcing all sample pairs to be equally aligned with the same ideal similarities, which is known as global kernel alignment. And it has demonstrated promising performance for multi-view clustering [8]. However, global kernel alignment indiscriminately forces sample pairs to be equally aligned to the sample ideal similarities regardless of the distance between sample pairs and the intra-cluster variance of samples. Moreover, the alignment is inconsistent with a well-established concept that the similarity evaluated for two farther samples in a high-dimensional space is less reliable owing to the presence of underlying manifold structure. Therefore, maximizing global alignment could make these pre-specified kernels less effectively utilized, which in turn degrades clustering performance. Differently, local kernel alignment, which only requires that the similarity of a sample to its k-nearest neighbors be aligned with the ideal similarity matrix, has been proven beneficial for clustering [43][8]. And it helps the clustering process to focus on close sample pairs that shall stay together and avoids involving unreliable similarity evaluation for distant sample pairs. Thus, the local structure of data can be well utilized to produce better alignment for clustering. Therefore, clustering performance can be further improved by maximizing local kernel alignment on the constructed multiple views generated by the proposed approach.

3.2.2. Localized multi-view clustering with ELM

Li et al. [32] derived a new optimization problem to implement the idea of local kernel alignment, and designed a two-step algorithm to solve it efficiently. The algorithm partitions a multi-view dataset with n samples represented in m views into k disjoint clusters. The objective function of the algorithm can be formalized as follows:

$$\begin{aligned}\min_{\mathbf{H}\in\mathbf{R}^{n\times k},\mu\in\mathbf{R}_{+}^{m}} \sum_{i=1}^{n}[tr(\mathbf{K}_{\mu}(\mathbf{T}^{(i)}-\mathbf{T}^{(i)}\mathbf{H}\mathbf{H}^{\top}\mathbf{T}^{(i)}))+\frac{\lambda}{2}\mu^{\top}\mathbf{W}^{(i)}\mu] \\ s.t.\ \ \mathbf{H}^{\top}\mathbf{H}=\mathbf{I}_{k},\mu^{\top}\mathbf{1}_{m}=1\end{aligned} \tag{17}$$

where $\mathbf{W}$ is a positive semi-definite matrix with $W_{pq}=tr(\mathbf{K}_{p}^{\top}\mathbf{K}_{q})$ $(1\leq p,q\leq m)$, $\mu=[\mu_1,\mu_2,\cdots,\mu_m]^{\top}$ denotes the coefficients of each base kernel that should be optimized during learning, $\mathbf{H}\mathbf{H}^{\top}$ is the ideal kernel matrix and

λ is the regularization parameter. For each sample, $\mathbf{T}^{(i)} = \mathbf{S}^{(i)}\mathbf{S}^{(i)\top}$ with $\mathbf{S}^{(i)} \in \{0,1\}^{n\times\tau}$ indicating the τ-nearest neighbors of the i-th sample. $\mathbf{K}_\mu$ is a combined kernel, which can be calculated as follows:

$$k_\mu(\mathbf{x}_i, \mathbf{x}_j) = \phi_\mu(\mathbf{x}_i)^\top \phi_\mu(\mathbf{x}_j) = \sum_{p=1}^{m} \mu_p^2 k_p(\mathbf{x}_i, \mathbf{x}_j) \tag{18}$$

Inspired by this work, we propose a ELM based multiple kernel clustering algorithm with local kernel alignment maximization using the proposed multi-view clustering approach in Section 3.1. The proposed algorithm first extends a single-view dataset to a multi-view dataset by ELM feature mapping with each view corresponding to a different value of hidden-layer node. Gaussian kernel matrix is then calculated for each view. After that, these kernels are combined for clustering with the help of local kernel alignment maximization, and a matrix $\mathbf{H}$, which contains k-dimensional representations of samples on the unit sphere, is returned. Finally, k-means clustering is performed on $\mathbf{H}$ to obtain the clustering result. The proposed algorithm can be summarized in Algorithm 4.

It should be noted that $\mathbf{H}$ in Eq. (17) is very difficult to solve directly. A two-step alternating optimization strategy is used to solve this problem.

Step1 : H is optimized given μ. With the sample-specific kernel weights μ fixed, $\mathbf{H}$ can be obtained by solving the following optimization problem:

$$\min_{\mathbf{H}\in\mathbf{R}^{n\times k}} \sum_{i=1}^{n} [tr(-\mathbf{K}_\mu \mathbf{T}^{(i)} \mathbf{H}\mathbf{H}^\top \mathbf{T}^{(i)}) \quad s.t. \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k \tag{19}$$

The problem in (19) can be reformulated as follows:

$$\max_{\mathbf{H}\in\mathbf{R}^{n\times k}} tr(\mathbf{H}^\top \sum_{i=1}^{n} (\mathbf{T}^{(i)}\mathbf{K}_\mu\mathbf{T}^{(i)})\mathbf{H}) \; s.t. \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{20}$$

where $\mathbf{H}$ can be obtained by performing eigenvalue decomposition of matrix $\sum_{i=1}^{n}(\mathbf{T}^{(i)}\mathbf{K}_\mu\mathbf{T}^{(i)})$.

Step2 : μ is optimized given H. With the eigenvectors obtained from the first step, the optimization in (17) with respect to μ is equal to solve the following quadratic programming with linear constraints:

$$\min_{\mu \in \mathbf{R}_+^m} \frac{1}{2}\mu^\top (2\mathbf{P} + \lambda \mathbf{W})\mu \;\; s.t. \;\; \mu^\top \mathbf{1}_m = 1 \tag{21}$$

where $\mathbf{P} = diag([tr(\mathbf{K}_1\mathbf{Q}), \cdots, tr(\mathbf{K}_m\mathbf{Q})])$ with $\mathbf{Q}$ is defined by $\sum_{i=1}^n (\mathbf{T}^{(i)} - \mathbf{T}^{(i)}\mathbf{H}\mathbf{H}^\top\mathbf{T}^{(i)})$ and $W_{pq} = \sum_{i=1}^n tr(\mathbf{K}_p\mathbf{T}^{(i)}\mathbf{K}_q\mathbf{T}^{(i)})$.

Algorithm 4 Localized multi-view clustering with ELM

```
Input: Given a dataset $\mathcal{X}$ with $n$ samples, $k$ clusters and a set of hidden-layer nodes $\{\mathbf{L}_t\}_{t=1}^m$.
Output: Cluster indicator $\mathbf{G}$.
Initialize parameters $\lambda$, $\epsilon_0$ and $\sigma$.
Construct $m$ views $\{\mathbf{O}_s\}_{s=1}^m$ using ELM random feature mapping with different hidden-layer nodes $\mathbf{L}_t$.
Calculate Gaussian kernels $\{\mathbf{K}_p\}_{p=1}^m$ for constructed views.
Initialize $\mu^{(1)} = \mathbf{1}_m/m$ and $iter = 1$.
Generate $\mathbf{S}^{(i)}$ for $i-$th sample by $\mathbf{K}_{\mu^{(1)}}$.
repeat
    $\mathbf{K}_\mu^{(iter)} = \sum_{p=1}^m (\mu_p^{(iter)})^2 \mathbf{K}_p$.
    Update $\mathbf{H}^{(iter)}$ by solving (20) with given $\mathbf{K}_\mu^{(t)}$.
    Update $\mathbf{K}_\mu^{(t)}$ by solving (21) with given $\mathbf{H}^{(iter)}$.
    $iter = iter + 1$
until $\left(obj^{(iter-1)} - obj^{(iter)}\right)/obj^{(iter)} \le \epsilon_0$
Perform k-means clustering on $\mathbf{H}$ and return the cluster indicator $\mathbf{G}$.
```

3.3. Discussion and extension

This study proposes a general multi-view clustering approach based on ELM random feature mapping. In our approach, original single-view data are transformed to a nonlinear feature space by ELM activation functions. And the hidden-layer matrix is regarded as a view with respect to different hidden-layer nodes. These views are combined to improve clustering performance by existing multi-view clustering algorithms. The results shown in Fig. 1 demonstrate that the views generated by ELM random feature mapping have some compatible and complementary information, which is the key idea of multi-view learning. And the results shown in Tables 2-4 exhibit the effectiveness of the proposed ELM based multi-view learning approach. Therefore, the proposed approach can be generalized to various machine learning tasks, such as classification, clustering and regression. And, this multi-view learning approach can be used in multiple kernel learning, co-training and subspace learning. Besides, it can be used in various applications, such as object detection, information retrieval and pattern recognition.

It should be noted that this work is obviously different from the work in [41], which only performs multi-view clustering on multi-view datasets with each view recomputed by ELM random feature mapping. Moreover, the multi-view datasets used in [41] have complementary and compatible information, whereas the datasets used in the present work are represented by only one view.

Furthermore, we exploit the properties of multi-view clustering based on the proposed approach. And we find that existing multi-view clustering algorithms usually conduct clustering with the help of kernels. Inspired by the work in [32], we propose an ELM-based multi-view clustering algorithm with local kernel alignment maximization. Differently, our proposed algorithm is performed on the original single-view datasets, whereas the work in [32] performs clustering in the pre-specified kernels. And the effectiveness of the proposed algorithm relies on the views generated by our proposed approach in Section 3.1.1.

4. Experimental results

This section evaluates the proposed algorithm on 10 datasets and compares it with other algorithms, which includes single-view clustering method performed on the original data, ELM-based single-view clustering methods and ELM-based multi-view clustering methods using the proposed approach. All studies are conducted on a computer with a 3.6GHz Intel Xeon E5-1620 CPU and 48GB of memory with Matlab R2014a (64bit).

4.1. Datasets

A wide range of datasets used in the experiment are summarized in Table 1. The number of datasets classes ranges from 2 to 40, the sample number reaches up to 2,310. These datasets have only one feature representation and the number of attributes of each feature representation scales from 4 to 1,024. All datasets are from real-world applications (e.g., image segmentation and face recognition) and are widely used as benchmarks for verifying the effectiveness of clustering and classification.

Eight datasets, which include Iris, heart, Libras, Balance, Diabetes, Vehicle, CNAE and Segment are downloaded from the UCI Machine Learning Repository [1]. However, these datasets have a relatively small size and low dimensions. Yale and Orl [2], which are usually used in face recognition tasks, are selected in the experiments to fully verify the effectiveness of the clustering algorithms on high-dimensional datasets. The features of datasets are normalized to [-1,1] using min-max normalization.

4.2. Baseline clustering methods

The proposed ELM-based multi-view clustering algorithms are compared with several typical clustering algorithms, namely:

- **K-means (KM)[33]:** This algorithm minimizes the sum of intra-cluster variance and proceeds by alternating between assigning instances to their closest center and recomputing the centers, until a local minimum is monotonically reached.

- **Kernel k-means (KKM)[33]:** This algorithm is a generalization of the standard k-means algorithms where the samples from the input space are mapped to a high-dimensional feature space via nonlinear transformation.

[1] http://archive.ics.uci.edu/ml/datasets.html

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

Table 1: Description of datasets

	Datasets	Attributes	Classes	Samples
UCI datasets	Iris	4	3	150
	Heart	13	2	270
	Libras	90	15	360
	Balance	4	3	625
	Diabetes	8	2	768
	Vehicle	18	4	846
	CNAE	856	9	1080
	Segment	19	7	2310
Face Recognition	Yale	1024	15	165
	Orl	1024	40	400

- **Spectral clustering (SC)[33]:** This algorithm partitions samples into clusters using the eigenvectors of an affinity matrix (i.e., kernel matrix) derived from the original data.

- **Sparse subspace clustering (SSC)[44]:** This algorithm cluster samples that lie in a union of low-dimensional subspaces. It finds a sparse representation of each sample in the dictionary of the other samples, builds a similarity graph using the sparse coefficients, and obtains the segmentation of the data using spectral clustering.

- **ELM k-means [16]:** The clustering results are obtained by conducting k-means clustering on the data in ELM feature space.

- **Unsupervised Extreme Learning Machine (US-ELM) [18]:** This algorithm determines the output weight in unsupervised ELM via manifold regularization and conducts k-means clustering on the data in the three dimensional embedding space. This algorithm can capture the manifold structure in the data and performs well on a dataset with manifold property.

- **Iterative weighted ELM for clustering (ELMC$^{\mathbf{Iter}}$)[19]:** This algorithm initializes the class information with a simple clustering method (e.g., k-means). Then, it alternates between training W-ELM based on the current class labels and updates the class information based on the prediction of W-ELM until convergence.

- **ELM clustering based on LDA (ELMC$^{\mathbf{LDA}}$)[19]:** This algorithm initially trains the output weights β of ELM with a fixed L, which solves a generalized eigenvalue problem. Then, k-means clustering is performed in the ELM output space to learn L with fixed β. This process is repeated until a local minimum is obtained.

- **ELM clustering based on kernel k-means (ELMC$^{\mathbf{KM}}$)[19]:** This algorithm initially computes the kernel gram matrix based on the hidden-layer output matrix of ELM. Then, kernel k-means is performed with kernel gram matrix to obtain the cluster index vector.

- **Co-regularized spectral clustering (CRSC)[24]:** This method is for spectral clustering. The pairwise co-regularization scheme is used in the experiment.

- **Robust multi-view spectral clustering (RMSC)[25]:** This algorithm initially constructs a transition probability matrix from each single view. Then, a low-rank and sparse decomposition is conducted to recover a shared transition probability matrix. The recovered matrix is subsequently used as an input to the standard Markov chain for clustering.

- **Multi-modal spectral clustering (MMSC)[45]:** This algorithm considers each type of feature as one modal and learns a common shared graph Laplacian matrix by unifying different modals (features or views). A nonegative relaxation is also added to improve the robustness and efficiency of clustering.

- **Multiple kernel k-means (MKKM)[46]:** This algorithm alternately performs kernel k-means and updates kernel coefficients.

- **Robust multiple kernel k-means (RMKKM)[47]:** This algorithm improves the robustness of MKKM by replacing the sum-of-squared loss with an $l_{2,1}$-norm one.

The Matlab code of MMSC is available from `http://www.escience.cn/people/fpnie/papers.html`, whereas the codes of CRSC and RMSC can be downloaded from `http://ss.sysu.edu.cn/~py/`. The codes of KKM and MKKM are publicly available at `http://github.com/mehmetgoen/lmkkmeans`. The matlab implementations of other algorithms are obtained from the websites of the authors or by email.

4.3. Evaluation of clustering

Clustering partitions a set of samples $\{\mathbf{x}_i\}_{i=1}^n \in \mathcal{X}$ into groups (called clusters) based on their similarities. Samples belonging to the same cluster are similar, whereas samples from different clusters are distinct. Typical objective functions in clustering are optimized to maximize the inter-cluster variance and minimize the intra-cluster variance. This criterion for evaluating the quality of a cluster is known as internal criterion. However, good scores on the internal criterion are not equivalent to good effectiveness in an application. An alternative to the internal criterion is a direct evaluation in the application of interest. An external criterion that evaluates how well the clustering matches the gold standard class [43] can then be computed. Three external criteria for evaluating the clustering quality are given as follows:

(1) **Clustering Accuracy(ACC):** It has been widely used to measure the clustering result, which is defined in terms of true label information. And it can be computed as follows:

$$ACC = \frac{\sum\limits_{i=1}^{n} \delta(y_i, map(\hat{y_i}))}{n} \tag{22}$$

where n is number of samples to be clustered, $\hat{y_i}$ and y_i are the predicted label and the ground-truth label of pattern x_i, respectively. $\delta(x, y)$ is a function that equals to 1 if $x = y$; otherwise, 0. $map(\cdot)$ is an optimal mapping function that permutes clustering labels to match the ground-truth labels by the Hungarian algorithm [48]. A larger ACC indicates better performance.

(2) **Normalized Mutual Information(NMI):** It measures the trade-off between the quality of the clustering and the number of clusters. NMI is defined as follows:

$$NMI(S,C) = \frac{I(S,C)}{[H(S)+H(C)]/2} \tag{23}$$

where $S = \{s_1, s_2, ..., s_k\}$ and $C = \{c_1, c_2, ..., c_j\}$ are the set of clusters and the set of classes, respectively. I is the mutual information which is defined as follows:

$$\begin{aligned} I(S,C) &= \sum_k \sum_j P(s_k \cap c_j) log \frac{P(s_k \cap c_j)}{P(s_k)P(c_j)} \\ &= \sum_k \sum_j \frac{|s_k \cap c_j|}{n} log \frac{N|s_k \cap c_j|}{|s_k||c_j|} \end{aligned} \tag{24}$$

H is the entropy which is defined as follows:

$$H(S) = -\sum_k P(s_k) log P(s_k) = -\sum_k \frac{|s_k|}{n} log \frac{|s_k|}{n} \tag{25}$$

where $P(s_k), P(c_j)$, and $P(s_k \cap c_j)$ are the probabilities of a data sample in cluster s_k, class c_j, and in the intersection of s_k and c_j, respectively. Similarly, the value of NMI is between 0 and 1, with a larger value indicating a better clustering quality.

(3) **Purity:** As mentioned before, each cluster is assigned to the class that is most frequent in the cluster. Purity is a criterion that measures the accuracy of this assignment by counting the number of correctly assigned samples and dividing by n. Purity can be formalized as follows:

$$purity(S,C) = \frac{1}{N} \sum_k \max_j |s_k \cap c_j| \tag{26}$$

Similar to the other two criteria, a larger purity value indicates better clustering quality. High purity is easy to achieve when the number of clusters is large. Particularly, purity is 1 if each sample is viewed as a cluster.

4.4. Result analysis

The clustering algorithms are performed on the 10 datasets, as described in Section 4.1. As these algorithms are stochastic, every algorithm is run 50 times on each dataset and the average result is obtained. The evaluation measures used in this study are ACC, NMI, and Purity as described in Section 4.3. In order to validate the effectiveness of the proposed ELM based multi-view clustering approach in Section 3.1, we calculate the Euclidean distance distributions of different views on the Libras dataset, which are shown in Fig. 1. And then, we conduct experiments to test the affect of hidden-layer nodes on ELM based clustering. The results are depicted in 2, which validate the rationality of ELM based multi-view clustering. Tables 2-4 show the detailed clustering results of various clustering algorithms on the 10 datasets in terms of ACC, NMI, and Purity, respectively. Each cell of these tables depicts the clustering result with the top one denoting the mean value and the bottom one denoting the standard deviation. The best results are marked in bold. It can be seen that, on all these datasets, algorithms based on our proposed approach yield satisfactory results.

Burghouts et al. [49] mentioned that knowledge on the distribution of distances generated by similarity functions is crucial in deciding whether features are similar or not. Moreover, knowledge on the distribution of distances aids in the construction of good clustering algorithms. Thus, we calculate the Euclidean distance distribution of different views with respect to different hidden-layer nodes on the Libras dataset. The result is depicted in Fig. 1. The horizontal axis indicates the distance between two descriptors in increasing order from left to right. The distributions of distances between corresponding views are shown in different colors. The overlapping area between different distributions indicates their similarity. The histograms colored in red and blue show the distance distribution of the views generated by ELM with 1,000 and 2,000 hidden-layer nodes, respectively. The green histogram shows the distance distribution of the concatenated features of the two views. The probability density function is

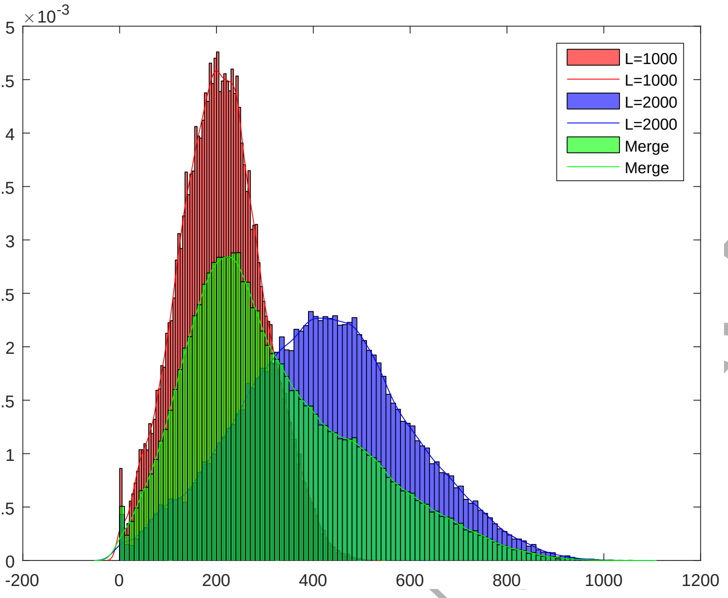
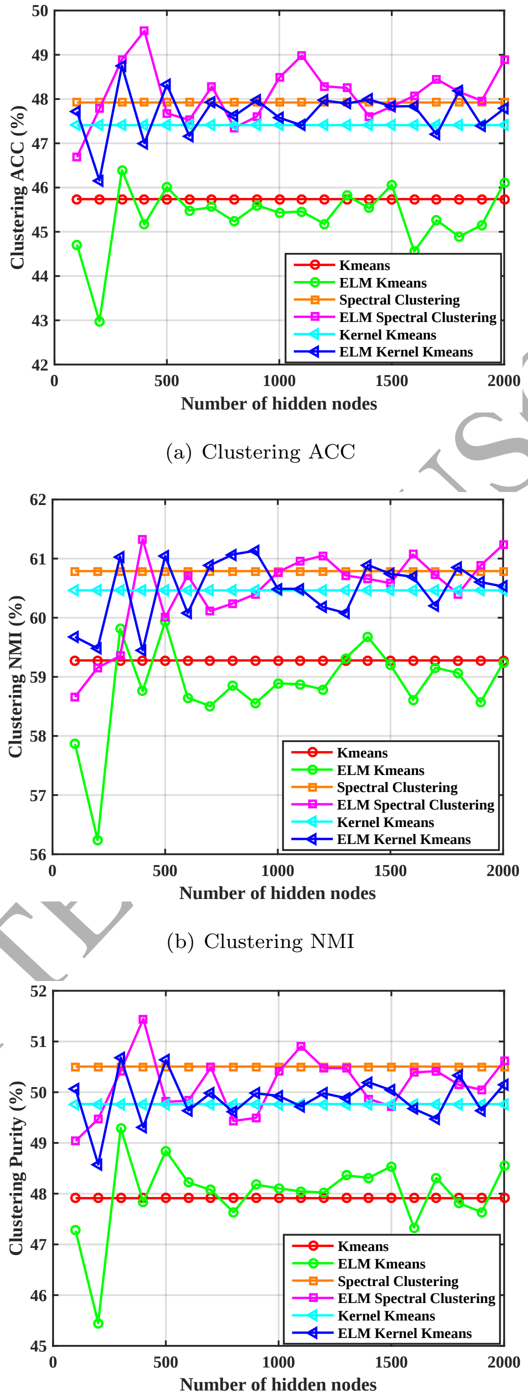


Figure 1: Distance distribution of different views on Libras

plotted by the kernel density estimation toolbox [3]. The figure shows that views generated by ELM with different hidden-layer nodes have the similar distribution. That is, they have complementary and compatible information. Therefore, these constructed views can be combined to improve clustering performance by multi-view clustering algorithms and validate the effectiveness of the proposed approach in Section 3.1.

Fig. 2 shows the clustering performance of several algorithms under different hidden nodes of ELM. It should be noted that k-means clustering, spectral clustering and kernel k-means clustering perform clustering in the original single-view datasets, so their clustering performances are independent of the hidden nodes of ELM. They are only used as a reference here. The other three algorithms conduct clustering in the feature space generated by ELM random feature mapping with different hidden nodes. This experiment uses Gaussian

[3] `http://www.ics.uci.edu/~ihler/code/kde.html`



(a) Clustering ACC

(b) Clustering NMI

(c) Clustering Purity

Figure 2: Clustering performance in the feature space generated by different hidden nodes of ELM on Libras

Table 2: Clustering ACC on datasets

Data Set	***Single-view methods***									***Multi-view methods***					
	KM	**KKM**	**SC**	**SSC**	**ELM KM**	**US-ELM**	**ELMC^{Iter}**	**ELMC^{LDA}**	**ELMC^{KM}**	**MKKM**	**RMKKM**	**MMSC**	**CRSC**	**RMSC**	**Proposed**
iris	88.67 0.00	92.67 0.00	89.33 0.00	90.67 0.00	90.00 0.00	87.64 1.02	90.67 0.00	85.68 9.47	86.16 8.63	93.33 0.00	92.93 1.26	90.67 0.00	91.70 1.00	96.00 0.00	**96.00** **0.00**
heart	79.02 0.41	79.63 0.00	79.63 0.00	79.26 0.00	80.00 0.00	71.10 1.91	80.33 0.89	75.79 7.34	74.44 8.65	79.26 0.00	80.00 0.00	77.78 0.00	79.63 0.00	80.00 0.00	**81.36** **0.17**
libras	45.74 1.55	50.07 1.52	49.27 1.77	44.43 2.78	45.30 1.77	50.03 1.66	48.83 2.02	45.32 2.61	45.83 2.76	48.94 1.86	46.72 1.75	46.69 2.19	48.83 2.14	50.18 1.70	**51.13** **1.59**
balance	52.40 1.18	61.12 0.83	60.56 1.70	54.08 0.44	52.11 1.55	59.98 2.60	52.80 1.09	52.45 3.14	52.84 3.10	61.32 0.31	55.92 2.82	60.38 1.90	61.50 2.55	61.48 0.39	**67.68** **0.00**
diabetes	67.45 0.00	66.54 0.00	66.80 0.00	65.63 0.00	67.71 0.00	68.10 0.00	68.01 0.57	67.67 0.51	67.16 0.38	65.49 0.00	68.22 0.04	66.67 0.00	68.23 0.00	68.23 0.00	**69.27** **0.00**
vehicle	36.96 0.09	39.48 0.00	39.19 0.09	37.35 0.00	40.79 1.64	41.77 0.26	43.18 0.56	38.99 2.39	41.09 5.09	38.18 0.02	43.43 2.92	42.01 1.88	38.49 0.49	44.09 0.00	**46.85** **2.67**
CNAE	42.25 4.83	59.39 1.88	58.01 2.71	25.09 0.44	43.69 4.54	51.85 1.60	56.75 6.56	39.32 5.51	45.62 3.75	32.07 0.33	45.73 5.96	55.45 3.90	50.12 2.78	54.52 0.06	**62.57** **0.52**
segment	66.11 1.48	74.11 0.04	73.28 0.16	63.89 0.19	66.43 2.41	69.11 0.67	74.01 2.48	65.67 7.42	67.57 6.89	72.95 0.36	67.94 0.02	58.99 6.08	73.35 2.87	74.60 0.32	**77.63** **0.74**
yale	42.75 3.27	54.39 2.63	48.47 3.83	36.45 2.72	42.39 3.26	45.22 3.54	47.33 3.50	40.72 3.57	41.21 3.84	51.89 2.48	46.67 4.36	49.20 2.19	53.52 1.92	52.04 2.47	**56.56** **2.64**
orl	52.99 3.26	66.69 3.41	66.95 3.10	51.95 2.02	51.82 2.69	53.30 2.22	60.29 3.00	50.11 2.91	50.84 3.35	58.13 2.42	53.75 2.85	63.58 2.36	**71.48** **2.77**	65.57 3.02	66.66 3.29

kernels to build the similarity matrix for each view. The standard deviation (parameter σ) is set to the median of the pairwise Euclidean distances between every pair of data points for the Libras dataset. It can be seen that on the Libras, the clustering performances of the ELM based methods are not stable and sensitive to the hidden nodes of ELM. Moreover, this observation exists on the rest datasets. This figure also indicates that perform clustering in the hidden-layer matrix by ELM can enhance clustering performance than perform clustering in the original data. For example, the clustering performance of ELM based spectral clustering achieves promising results when the number of hidden nodes is set to 400. However, choosing a reasonable value for the hidden-layer nodes of the ELM that can yield the best clustering performance is difficult. Therefore, we construct views by a set of hidden-layer nodes, and then perform clustering on these views.

Table 2 shows the clustering accuracy of different algorithms on all datasets. As can be seen, multi-view clustering algorithms based on our approach achieve superior performance than the single-view clustering algorithms. And the proposed algorithm using local kernel alignment maximization yields the best result on the datasets except for the ORL dataset. For example, it exceeds the second best one by 6.2%, 5.82%, 3.03%, and 3.04% on balance, CNAE, segment and

Table 3: Clustering NMI on datasets

Data Set	***Single-view methods***									***Multi-view methods***					
	KM	**KKM**	**SC**	**SSC**	**ELM KM**	**US-ELM**	**ELMC^{Iter}**	**ELMC^{LDA}**	**ELMC^{KM}**	**MKKM**	**RMKKM**	**MMSC**	**CRSC**	**RMSC**	**Proposed**
iris	74.19 0.00	79.00 0.00	79.07 0.00	78.57 0.00	74.76 0.00	74.04 0.92	80.57 0.00	74.40 7.54	72.49 4.08	80.38 0.00	79.92 1.44	80.57 0.00	77.82 1.21	86.42 0.00	**87.05** **0.00**
heart	26.23 0.72	27.12 0.00	27.12 0.00	26.32 0.00	27.95 0.00	13.37 1.30	28.31 1.77	22.25 9.00	20.81 10.43	27.30 0.00	27.79 0.00	24.74 0.00	0.00 0.00	27.79 0.00	**30.17** **0.36**
libras	59.28 1.28	62.80 0.78	62.02 1.05	58.16 1.45	58.89 1.48	63.02 0.84	62.61 1.31	59.17 1.96	59.37 1.69	61.53 1.36	60.20 1.81	**64.43** **1.77**	63.33 1.35	62.59 1.32	63.46 1.17
balance	11.65 1.75	26.50 1.79	28.49 3.20	11.93 0.87	11.81 1.88	26.50 4.05	12.32 1.57	12.60 4.67	12.87 4.56	29.16 0.39	16.41 4.52	24.18 1.27	28.47 3.66	31.66 0.87	**33.34** **0.80**
diabetes	5.62 0.00	7.02 0.00	6.48 0.00	5.26 0.00	6.28 0.00	6.96 2.71	6.09 0.08	5.58 1.09	5.57 0.45	8.30 0.00	7.01 0.06	1.88 0.62	6.80 0.00	13.27 0.00	**14.65** **0.00**
vehicle	10.11 0.03	12.23 0.00	13.17 3.81	7.53 0.02	11.93 0.70	19.47 0.21	13.41 0.98	12.80 2.19	14.46 4.87	10.53 0.00	19.99 2.39	18.46 2.42	13.42 0.42	19.30 0.14	**21.93** **2.52**
CNAE	42.80 4.58	51.36 1.41	50.70 1.75	16.23 0.36	41.11 3.27	52.38 1.54	49.10 5.85	37.05 4.90	41.19 2.59	17.49 0.24	42.76 4.14	54.07 3.36	43.50 1.50	46.48 0.33	**54.06** **0.14**
segment	61.19 0.25	66.86 0.08	68.61 0.05	56.43 0.09	61.35 0.53	66.53 0.61	70.02 1.36	65.66 4.00	66.39 5.88	67.90 0.13	62.25 1.72	65.27 2.03	67.62 0.46	68.22 0.26	**70.08** **0.88**
yale	50.58 2.34	58.23 2.01	54.00 2.53	42.61 2.05	48.92 2.80	51.92 1.36	52.78 2.93	47.97 3.40	47.65 3.25	55.05 1.92	52.63 2.85	56.42 1.86	58.38 1.76	57.14 2.15	**59.16** **1.90**
orl	75.19 1.41	82.13 1.59	82.38 1.42	74.01 1.12	73.49 1.43	78.56 0.93	78.82 1.44	72.76 1.77	73.06 1.89	77.75 1.20	75.29 1.34	81.30 0.62	**85.49** **1.55**	81.90 1.54	82.26 1.35

yale, respectively.

It should be noted that the performances of MKKM and RMKKM are unsatisfactory compared with single-view methods, particularly on the CNAE dataset. This observation indicates that algorithms maximizing global kernel alignment may make pre-specified kernels less effectively utilized, and in turn adversely affect clustering performance. Therefore, it verifies the effectiveness of the proposed algorithm.

The NMI of the clustering results are shown in Table 3. And that, the results of multi-view methods based on the proposed approach achieve superior performance than the ELM-based single-view clustering methods and the traditional algorithms using the original data. For the CNAE dataset, the proposed algorithm gains lower variance, although the mean NMI is slightly lower than that of the MMSC algorithm.

Table 4 summarizes the clustering purity of different algorithms. Obviously, our proposed algorithm yields superior performance than the baselines.

Overall, multi-view clustering algorithms, which combine the different views generated by the proposed method, can achieve better performance compared with the single-view clustering methods. Our proposed algorithm, which exploits the local kernel alignment maximization can further improve the cluster-

Table 4: Clustering Purity on datasets

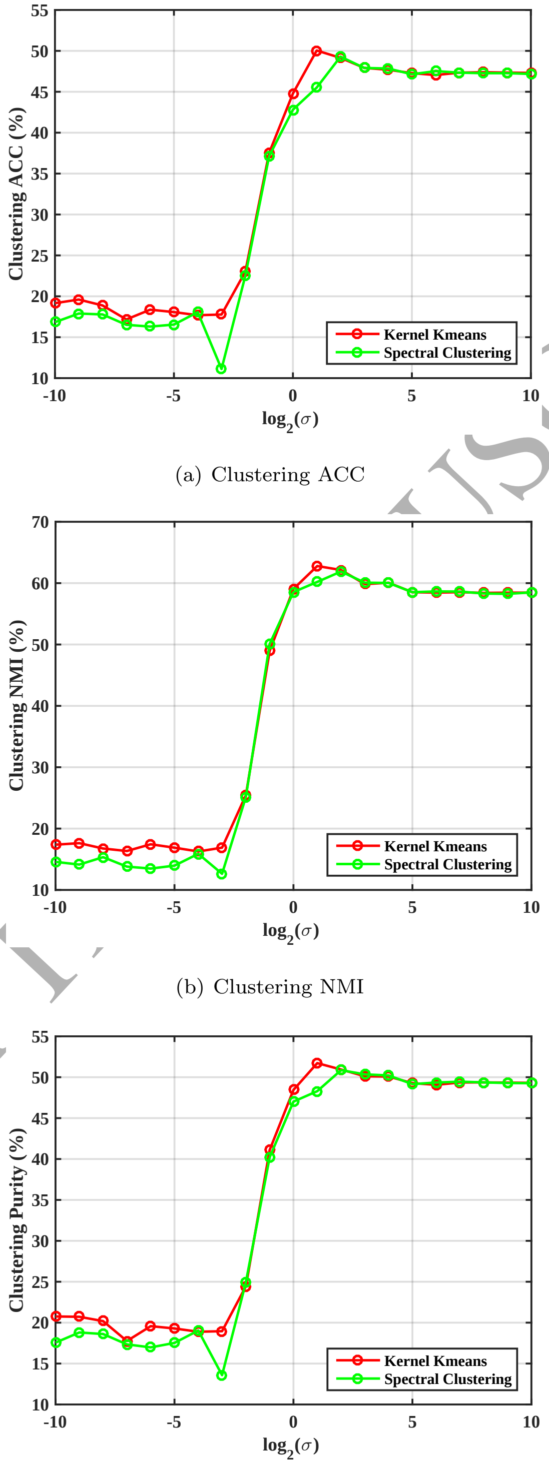
Data Set	*Single-view methods*									*Multi-view methods*					
	KM	**KKM**	**SC**	**SSC**	**ELM KM**	**US-ELM**	**ELMC^{Iter}**	**ELMC^{LDA}**	**ELMC^{KM}**	**MKKM**	**RMKKM**	**MMSC**	**CRSC**	**RMSC**	**Proposed**
iris	88.67 0.00	92.67 0.00	89.33 0.00	90.67 0.00	90.00 0.00	87.64 1.02	90.67 0.00	86.57 6.79	86.72 6.82	93.33 0.00	92.93 1.26	90.67 0.00	91.70 1.00	96.00 0.00	**96.00** **0.00**
heart	79.02 0.41	79.63 0.00	79.63 0.00	79.26 0.00	80.00 0.00	71.10 1.91	80.33 0.89	75.79 7.34	74.44 8.65	79.26 0.00	80.00 0.00	77.78 0.00	79.63 0.00	80.00 0.00	**81.36** **0.17**
libras	47.91 1.34	51.75 0.99	50.88 1.40	46.69 2.40	48.02 1.67	52.69 1.49	51.28 1.65	48.44 2.36	48.46 2.31	50.75 1.78	49.42 1.10	50.69 1.90	51.01 1.45	51.77 1.14	**52.74** **1.62**
balance	65.35 1.52	74.09 0.66	75.40 1.02	66.04 1.51	65.84 1.58	73.97 1.89	66.10 1.54	66.20 2.85	66.46 3.17	75.46 0.47	69.09 2.55	73.04 1.21	74.86 1.09	77.58 0.73	**79.04** **1.06**
diabetes	67.45 0.00	66.54 0.00	66.80 0.00	65.76 0.00	67.71 0.00	68.10 0.00	68.01 0.57	67.70 0.41	67.16 0.38	65.76 0.00	68.22 0.04	66.67 0.00	68.23 0.00	68.23 0.00	**69.27** **0.00**
vehicle	40.43 0.05	41.61 0.00	40.40 0.26	37.59 0.00	41.04 1.26	44.99 0.16	43.26 0.38	39.76 2.18	42.58 4.59	40.78 0.00	45.35 1.90	44.50 0.96	38.94 0.47	45.87 0.03	**48.88** **2.36**
CNAE	44.71 4.39	58.01 2.71	60.56 2.49	25.83 0.37	45.94 3.47	55.48 0.04	58.54 6.24	40.86 5.64	47.90 2.94	33.69 0.36	47.85 4.86	57.85 3.08	52.48 2.12	56.36 0.18	**64.96** **0.05**
segment	66.87 0.06	74.11 0.04	73.28 0.16	63.89 0.19	67.05 0.91	71.64 1.42	74.73 1.18	69.64 5.20	70.47 4.37	73.64 0.27	68.10 0.55	63.83 4.42	74.07 1.39	74.60 0.32	**77.63** **0.74**
yale	44.44 2.94	54.92 2.61	49.35 3.57	38.10 2.53	43.93 3.20	46.61 2.58	48.59 3.08	42.82 3.37	42.69 3.55	52.65 2.50	48.36 4.37	50.76 2.67	54.12 1.88	52.65 2.30	**56.95** **2.42**
orl	58.20 2.62	70.27 2.58	70.63 2.52	56.58 1.86	56.53 2.14	59.40 1.72	63.59 2.90	54.81 2.73	54.74 3.05	62.86 1.94	58.40 2.50	68.34 1.65	**74.60** **2.16**	69.31 2.60	70.13 2.51

ing performance. And multiple views generated by our approach can be used to improve clustering performance by various multi-view clustering algorithms. Therefore, the proposed approach exhibits good extensibility.

4.5. Parameter selection

For multi-view clustering methods in the ELM feature space, the sigmoid function is selected as the hidden-layer nodes activation function. An advantage of using the ELM mapping process is that its input weight matrix can be randomly generated according to any continuous probability distribution, and it does not need to be tuned. The only parameter that needs to be specified by the user is the number of hidden-layer nodes. And a set of hidden-layer nodes are used in the experiments to construct different views. λ is a regularization parameter for different algorithms in the experiment.

All experiments use the Gaussian kernels to build the similarity matrix for each view. Kernel k-means and spectral clustering are the basic methods of the clustering algorithms used in the experiments. The clustering performance of these methods with different σ values are shown in Fig. 3. The performance is relatively stable when σ is extremely too large or small. It can be seen that the clustering performance is highly sensitive to the σ values in a wide range, and this rule is widespread in different datasets. Different datasets require different



(a) Clustering ACC

(b) Clustering NMI

(c) Clustering Purity

Figure 3: Clustering performance under different σ on Libras

Table 5: Parameter selection

Methods	**L**	**P**	**NN**	**ρ**	**λ**	**σ**	**α**	**γ**
KM	-	-	-	-	-	-	-	-
KKM	-	-	-	-	-	2^[-10:1:10]	-	-
SC	-	-	-	-	-	2^[-10:1:10]	-	-
SSC	-	-	-	-	0.001	-	-	-
ELM KM	2000	-	-	-	-	-	-	-
US-ELM	2000	[0.5,1,2]	[1,5,10]	-	10^[-6:1:6]	-	-	-
ELMC$^{\text{Iter}}$	2000	[0.5,1,2]	-	-	10^[-6:1:6]	-	-	-
ELMC$^{\text{LDA}}$	2000	-	-	-	10^[-6:1:6]	-	-	-
ELMC$^{\text{KM}}$	2000	-	-	-	10^[-6:1:6]	-	-	-
MKKM	[100:100:2000]	-	-	-	-	2^[-10:1:10]	-	-
RMKKM	[100:100:2000]	-	-	-	-	2^[-10:1:10]	-	[0.1:0.1:0.9]
MMSC	[100:100:2000]	[0.5,1,2]	[1,5,10]	-	-	-	10^[-2:0.1:2]	-
CRSC	[100:100:2000]	-	-	-	0.01	2^[-10:1:10]	-	-
RMSC	[100:100:2000]	-	-	-	2^[-20:1:10]	2^[-10:1:10]	-	-
Proposed	[100:100:2000]	-	-	[0.05:0.05:0.95]	2^[-15:1:15]	2^[-10:1:10]	-	-

σ values to obtain a satisfactory performance. A set of σ values are used in the experiment to solve this problem. As shown in Table 5, the standard deviation $log_2\sigma$ ranged from -10 to 10 with incremental step 1.

Table 5 shows that the hidden-layer node is set to 2,000 for ELM based single-view clustering algorithms as mentioned in [16]. For multi-view clustering methods, a set of hidden-layer nodes, which range from 100 to 2,000 with incremental step 100 are used to construct different views. That is, 20 views are used for multi-view clustering in the experiments. The number of the nearest neighbor and the weight degree are denoted by NN and P, respectively. It should be noted that the parameter τ used in the proposed algorithm is selected from the set of $\rho * n$ by grid search, where n is the number of samples. The parameter γ is used to control the distribution of weights for different kernels in RMKKM. α is a penalty parameter for MMSC. The details of parameter selection are summarized in Table 5.

5. Conclusion

This study proposes a general multi-view clustering approach using ELM random feature mapping. First, hidden-layer matrix is calculated by ELM with different hidden-layer nodes. Taking each hidden-layer matrix as a view, multi-view clustering is performed on these views. Experiments show that combining these views together, we can get better clustering results than the corresponding Mercer kernel-based methods and the traditional algorithms performed in the feature space of the original single-view datasets. Moreover, a multi-view clustering algorithm with local kernel alignment maximization is proposed using our approach. As experimentally demonstrated on a total of 10 benchmark datasets, the clustering results of the proposed algorithm indicate obvious improvement over the corresponding baselines.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Nos. U1435219, 61403405, 61402507, 61303070.

References

[1] J. A. Hartigan, Clustering algorithms, John Wiley & Sons, Hoboken, New Jersey, 1975.

[2] V. Zografos, L. Ellis, R. Mester, Discriminative subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2107–2114.

[3] N. Jardine, C. J. van Rijsbergen, The use of hierarchic clustering in information retrieval, Information Storage and Retrieval 7 (5) (1971) 217–240.

[4] A. Baraldi, P. Blonda, A survey of fuzzy clustering algorithms for pattern recognition, IEEE Trans. Sys. Man Cybern. Part B(Cybern.) 29 (6) (1999) 778–785.

[5] P. Berkhin, A survey of clustering data mining techniques, in: Grouping Multidimensional Data, Springer, Berlin Heidelberg, 2006, pp. 25–71.

[6] S. P. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (2) (1982) 129–137.

[7] U. Von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.

[8] M. Gönen, A. A. Margolin, Localized data fusion for kernel k-means clustering with application to cancer biology, in: Advances in Neural Information Processing Systems, 2014, pp. 1305–1313.

[9] Z.-P. Liu, Quantifying gene regulatory relationships with association measures: A comparative study, Frontiers in genetics 8.

[10] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: IEEE International Joint Conference on Neural Networks, 2004, Proceedings, Vol. 2, IEEE, Piscataway, New Jersey, 2004, pp. 985–990.

[11] M. Girolami, Mercer kernel-based clustering in feature space, IEEE Transactions on Neural Networks 13 (3) (2002) 780–784.

[12] X. Liu, L. Wang, G.-B. Huang, J. Zhang, J. Yin, Multiple kernel extreme learning machine, Neurocomputing 149 (2015) 253–264.

[13] Y. Wang, Y. Dou, X. Liu, Y. Lei, PR-ELM: Parallel regularized extreme learning machine based on cluster, Neurocomputing 173 (2015) 1073–1081.

[14] Q. Lv, X. Niu, Y. Dou, J. Xu, Y. Lei, Classification of hyperspectral remote sensing image using hierarchical local-receptive-field-based extreme learning machine, IEEE Geoscience and Remote Sensing Letters 13 (3) (2016) 434–438.

[15] A. K. Alshamiri, B. R. Surampudi, A. Singh, A novel elm k-means algorithm for clustering, in: Swarm, Evolutionary, and Memetic Computing, Springer, Berlin Heidelberg, 2014, pp. 212–222.

[16] Q. He, X. Jin, C. Du, F. Zhuang, Z. Shi, Clustering in extreme learning machine feature space, Neurocomputing 128 (2014) 88–95.

[17] G. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Sys. Man Cybern. Part B(Cybern.) 42 (2012) 513–529.

[18] G. Huang, S. Song, J. N. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Transactions on Cybernetics 44 (12) (2014) 2405–2417.

[19] G. Huang, T. Liu, Y. Yang, Z. Lin, S. Song, C. Wu, Discriminative clustering via extreme learning machine, Neural Networks 70 (2015) 1–8.

[20] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, ACM, New York, 1998, pp. 92–100.

[21] P. Gehler, S. Nowozin, On feature combination for multiclass object classification, in: IEEE 12th International Conference on Computer Vision, 2009, IEEE, Piscataway, New Jersey, 2009, pp. 221–228.

[22] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, arXiv preprint arXiv:1304.5634.

[23] S. Bickel, T. Scheffer, Multi-view clustering, in: ICDM, Vol. 4, 2004, pp. 19–26.

[24] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: Advances in Neural Information Processing Systems, 2011, pp. 1413–1421.

[25] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, in: AAAI Conference on Artificial Intelligence, 2014, pp. 2149–2155.

[26] H. Wang, C. Weng, J. Yuan, Multi-feature spectral clustering with minimax optimization, in: IEEE Conference on Computer Vision and Pattern

Recognition (CVPR), 2014, IEEE, Piscataway, New Jersey, 2014, pp. 4106–4113.

[27] H. Wang, F. Nie, H. Huang, Multi-view clustering and feature learning via structured sparsity, in: Proceedings of the 30th International Conference on Machine Learning (ICML'13), 2013, pp. 352–360.

[28] K. Chaudhuri, S. M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, New York, 2009, pp. 129–136.

[29] D. Greene, P. Cunningham, A matrix factorization approach for integrating multiple data views, in: Machine Learning and Knowledge Discovery in Databases, Springer, Berlin Heidelberg, 2009, pp. 423–438.

[30] E. Bruno, S. Marchand-Maillet, Multiview clustering: A late fusion approach using latent models, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, 2009, pp. 736–737.

[31] C. Zhang, X. ShiXiong, L. Bing, L. Zhang, Extreme maximum margin clustering, IEICE Trans. on Information and Systems 96 (8) (2013) 1745–1753.

[32] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, Multiple kernel clustering with local kernel alignment maximization, in: IJCAI, 2016.

[33] G. Tzortzis, Clustering using similarity and kernel matrices, Ph.D. thesis, University of Ioannina (2014).

[34] A. K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognition Letters 31 (8) (2010) 651–666.

[35] F. Nie, D. Xu, I. W. Tsang, C. Zhang, Spectral embedded clustering, in: IJCAI, 2009, pp. 1181–1186.

[36] S. X. Yu, J. Shi, Multiclass spectral clustering, in: Proceedings. Ninth IEEE International Conference on Computer Vision, 2003, IEEE, Piscataway, New Jersey, 2003, pp. 313–319.

[37] A. Y. Ng, M. I. Jordan, Y. Weiss, et al., On spectral clustering: Analysis and an algorithm, Advances in Neural Information Processing Systems 2 (2002) 849–856.

[38] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319.

[39] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.

[40] Y. Lan, Y. C. Soh, G.-B. Huang, Constructive hidden nodes selection of extreme learning machine for regression, Neurocomputing 73 (16) (2010) 3191–3199.

[41] Q. Wang, Y. Dou, X. Liu, Q. Lv, S. Li, Multi-view clustering with extreme learning machine, Neurocomputing 214 (2016) 483–494.

[42] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16) (2007) 3056–3062.

[43] C. D. Manning, P. Raghavan, H. Schütze, et al., Introduction to information retrieval, Vol. 1, Cambridge University Press Cambridge, 2008.

[44] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE transactions on pattern analysis and machine intelligence 35 (11) (2013) 2765–2781.

[45] X. Cai, F. Nie, H. Huang, F. Kamangar, Heterogeneous image feature integration via multi-modal spectral clustering, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, IEEE, Piscataway, New Jersey, 2011, pp. 1977–1984.

[46] H.-C. Huang, Y.-Y. Chuang, C.-S. Chen, Multiple kernel fuzzy clustering, IEEE Transactions on Fuzzy Systems 20 (1) (2012) 120–134.

[47] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, S. Yi-Dong, Robust multiple kernel k-means clustering using ℓ_{21} norm, in: IJCAI, 2015, pp. 3476–3482.

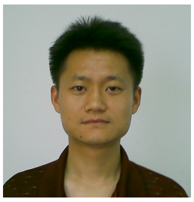
[48] J. Goldberger, T. Tassa, A hierarchical clustering algorithm based on the hungarian method, Pattern Recognition Letters 29 (11) (2008) 1632–1638.

[49] G. Burghouts, A. Smeulders, J.-M. Geusebroek, The distribution family of similarity distances, in: Advances in neural information processing systems, 2008, pp. 201–208.

Qiang Wang received his B.S. degree in Computer Science and Technology from Jilin University, PR China, in 2011, and received his M.S. degree in computer science and technology from National University of Defense Technology, PR China, in 2013. Now he is a Ph.D. candidate at National University of Defense Technology. His research interests include high performance computing, information security, and machine learning.

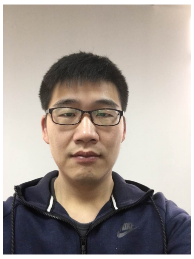
Yong Dou was born in 1966, professor, Ph.D. supervisor, senior membership of China Computer Federation (E200009248). He received his B.S., M.S., and Ph.D. degrees in computer science and technology from National University of Defense Technology in 1989, 1992 and 1995. His research interests include high performance computer architecture, high performance embedded microprocessor, reconfigurable computing, machine learning, and bioinformatics. He is a member of the IEEE and the ACM.

Xinwang Liu is a research assistant at National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, PR China. He received his B.S. degree in computer science and technology from Chongqing Technology and Business University, Chongqing, in 2006, and received M.S. and Ph.D. degrees in computer science and technology from National University of Defense Technology in 2008 and 2013. From October 2010, he spent one year in visiting the Engineering & Computer Science, the Australia National University, supported by the China Scholarship Council. From November 2011 to October 2012, he is a visiting student of the School of Computer Science and Software Engineering, University of Wollongong. His research interests focus on kernel learning and feature selection.



Fei Xia, born in 1980. Received his Ph.D. degree from National University of Defense Technology in 2011. His research interests include computer architecture, bioinformatics and signal processing.

Qi Lv is a Ph.D. candidate of National University of Defense Technology, PR China. He received his B.S. degree in computer science and technology from Tsinghua University, Beijing, in 2009, and received his M.S. degree in computer science and technology from National University of Defense Technology in 2011. His research interests include high performance computer architecture, machine learning, and remote sensing image processing.



Ke Yang received his M.S. degree in computer science and technology from National University of Defense Technology, PR China, in 2015. Now he is a Ph.D. candidate at National University of Defense Technology. His research interests include computer vision, pattern recognition, and machine learning.