Neural Networks 70 (2015) 1–8

Contents lists available at ScienceDirect

Neural Networks

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

Discriminative clustering via extreme learning machine

Gao Huang[a], Tianchi Liu[b,*], Yan Yang[c,d], Zhiping Lin[b], Shiji Song[a], Cheng Wu[a]

[a] *Department of Automation, Tsinghua University, Beijing 100084, China*
[b] *School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, 639798, Singapore*
[c] *Energy Research Institute @ NTU (ERI@N), Nanyang Technological University, Nanyang Avenue, 639798, Singapore*
[d] *State Key Laboratory of Millimeter Waves, School of Information Science and Engineering, Southeast University, Nanjing 210096, China*

ARTICLE INFO

Article history:
Received 17 December 2014
Received in revised form 27 April 2015
Accepted 2 June 2015
Available online 19 June 2015

Keywords:
Discriminative clustering
Extreme learning machine
k-means
Linear discriminant analysis

ABSTRACT

Discriminative clustering is an unsupervised learning framework which introduces the discriminative learning rule of supervised classification into clustering. The underlying assumption is that a good partition (clustering) of the data should yield high discrimination, namely, the partitioned data can be easily classified by some classification algorithms. In this paper, we propose three discriminative clustering approaches based on Extreme Learning Machine (ELM). The first algorithm iteratively trains weighted ELM (W-ELM) classifier to gradually maximize the data discrimination. The second and third methods are both built on Fisher's Linear Discriminant Analysis (LDA); but one approach adopts alternative optimization, while the other leverages kernel *k*-means. We show that the proposed algorithms can be easily implemented, and yield competitive clustering accuracy on real world data sets compared to state-of-the-art clustering methods.

1. Introduction

As one of the most fundamental unsupervised learning tasks in machine learning and computational intelligence, clustering has been widely studied and applied in various domains (Punj & Stewart, 1983; Steinbach, Karypis, & Kumar, 2000; Xu & Wunsch, 2005). The goal of clustering is to find a partition of the data, such that samples within the same cluster are similar, while samples from different clusters are distinct (Jain & Dubes, 1988). Many clustering algorithms have been proposed to fulfil this task, such as the *k*-means algorithm (Hartigan & Wong, 1979), graph theoretic clustering (Belkin & Niyogi, 2001; Ng, Jordan, & Weiss, 2001; Shi & Malik, 2000) and information theoretic clustering (Gokcay & Principe, 2002; Gomes, Krause, & Perona, 2010; Sugiyama, Yamada, Kimura, & Hachiya, 2011).

Discriminative Clustering (DC) is an important type of clustering approach, and is relatively new in the clustering research field (Ding & Li, 2007; Huang, Zhang, Song, & Zheng, 2015; Niu, Dai, Shang, & Sugiyama, 2013; Xu, Neufeld, Larson, & Schuurmans, 2004; Ye, Zhao, & Wu, 2007; Zhao, Wang, & Zhang, 2008). Generally, DC aims to separate the training data into clusters with high discrimination. In other words, if we take the clustering labels of a good clustering of the data as the targets, then we can easily learn a supervised classifier on this "labeled" data set with high accuracy. Intuitively, the goal of DC is compatible with that of classical clustering, since high discrimination between different clusters also implies that samples from different clusters are dissimilar, while samples within the same cluster have relatively high similarity. This assumption inspires many novel clustering algorithms.

As one of the representative DC approaches, Maximum Margin Clustering (MMC) (Xu et al., 2004) introduces the idea of margin maximization in supervised learning into clustering. MMC tries to find a partition of the data so that different clusters are separated by large margins, and thus large margin based classifiers, e.g., support vector machines (SVM), can classify the clusters with high accuracy. Though MMC has achieved encouraging results on many clustering tasks (Xu et al., 2004), it has two main drawbacks: (1) it is limited to binary clustering, and (2) it involves solving a Semi-Definite Programming (SDP) which is computationally expensive. Regarding the first problem, Xu and Schuurmans (2005) extended MMC to multi-class clustering. With regards to the second issue, Valizadegan and Jin (2007) proposed a generalized MMC which reduces the number of parameters in the SDP formulation from n^2 in Xu and Schuurmans (2005) to n, thus significantly improves the efficiency of MMC. In Zhang, Tsang, and Kwok (2007), an iterative

[*] Corresponding author.
E-mail addresses: huang-g09@mails.tsinghua.edu.cn (G. Huang), tcliu@ntu.edu.sg (T. Liu), y.yang@ntu.edu.sg (Y. Yang), ezplin@ntu.edu.sg (Z. Lin), shijis@mail.tsinghua.edu.cn (S. Song), wuc@tsinghua.edu.cn (C. Wu).

http://dx.doi.org/10.1016/j.neunet.2015.06.002
0893-6080/© 2015 Elsevier Ltd. All rights reserved.

support vector regression approach was introduced to scale MMC to data sets with thousands of samples. In Zhao et al. (2008), a linear time MMC algorithm was proposed based on cutting-plane optimization.

Another important type of DC is the Linear Discriminant Analysis (LDA)-based clustering. In De la Torre and Kanade (2006), a Discriminative Cluster Analysis (DCA) approach was proposed by jointly performing dimension reduction and clustering. Ding and Li (2007) combined LDA with k-means, yielding an efficient clustering algorithm which alternatively performs dimension reduction using supervised LDA and k-means clustering in the low dimensional space. Later, Ye et al. (2007) showed that the objective in Ding and Li (2007) can be optimized without alternative optimization, but solved by a single pass of kernel k-means.

There also exist many other types of DC algorithms, such as maximum volume clustering (Niu et al., 2013), information maximization-based clustering (Gomes et al., 2010), maximin separation probability clustering (Huang et al., 2015). However, few of these existing DC algorithms can simultaneously meet the following three basic requirements for clustering: (1) efficiently scales to large data sets; (2) naturally handles multi-cluster problem; and (3) capable of discovering nonlinear data structures.

Extreme Learning Machine (ELM) is a state-of-the-art supervised learning algorithm proposed by Huang, Zhu, and Siew (2004). ELM was originally proposed for classification and regression. It has several salient features: efficient, accurate and can be implemented easily (Butcher, Verstraeten, Schrauwen, Day, & Haycock, 2013; Huang, Huang, Song, & You, 2015; Huang, Zhou, Ding, & Zhang, 2012; Liu, Gao, & Li, 2012), and has been widely used in various applications (Cao, Chen, & Fan, 2014, 2015; Cao & Xiong, 2014; Shi, Cai, Zhu, Zhong, & Wang, 2013). Extending ELM for clustering has been addressed in several existing works. One straightforward approach is to perform clustering using any existing clustering algorithms, e.g., k-means, in the embedding space obtained by ELM (He, Jin, Du, Zhuang, & Shi, 2014). Though easy to be implemented, these approaches sacrifice the flexibility of ELM because the output weights of ELM are omitted, and it is not possible to perform regularization in them. This may degrade the robustness of clustering when training data are perturbed by noise. Huang, Song, Gupta, and Wu (2014) proposed to determine the output weight in unsupervised ELM (US-ELM) via manifold regularization, and perform clustering in the output space. The US-ELM algorithm can capture the manifold structure in the data, and is shown to perform well on data set with manifold property (Huang et al., 2014). Kasun, Liu, Yang, Lin, and Huang (2015) proposed to project the data along the output weight learned by ELM Auto Encoder, which is also an unsupervised learning process. It has been shown that the output weights learn the variance information of the data, and this embedding process reduces the within-cluster variance and preserve the between-cluster variance. Results suggest that this method works well on cluster-alike data sets. Different from embedding-based clustering, Zhang, Xia, Liu, and Lei (2013) introduced a clustering algorithm by iteratively training ELM classifier. Since the undesired imbalanced clustering problem often occurs in the iterative training procedure, some heuristics have to be introduced to avoid trivial solutions. Yang et al. (2014) proposed to find optimal data partitions using multiple ELMs. However, their work is designed for supervised learning and requires the ground truth of the data during training.

In this paper, we investigate the problem of extending ELM to discriminative clustering. The motivation is to take advantage of ELM, and to design clustering algorithms which inherit its salient advantages, such as high efficiency, easiness of implementation and capable of handling multi-class data set. The first proposed algorithm was an iterative weighted ELM (ELMC^{Iter}) approach similar to that proposed by Zhang et al. (2013). The difference is that we use the weighted ELM (W-ELM) (Zong, Huang, & Chen, 2013) to avoid imbalance clustering in a more principled way. The second and third methods are embedding-based methods, which take advantages of LDA. Different from Huang et al. (2014) and Kasun et al. (2015), the proposed methods learn the optimal embedding in a supervised manner, i.e., LDA, and therefore are expected to minimize the within-cluster distance and between-cluster distance at the same time. The second approach ELMC^{LDA} was inspired by Ding and Li (2007) which performs LDA and k-means alternatively. In their work (Ding & Li, 2007), LDA is performed in the original space. In contrast, we run LDA in the output space of ELM, which is a nonlinear mapping of the input space. In this way, our approach is able to discover nonlinear structure in training data. The third approach ELMC^{KM} has the same objective as our second approach, but it is solved via a kernel k-means with the kernel matrix calculated based on the centered hidden layer output matrix of ELM. ELMC^{KM} is built on the theoretical analysis given in Ye et al. (2007). However, the proposed method can efficiently deal with nonlinear clustering tasks, while the kernel-based clustering algorithm proposed in Ye et al. (2007) needs to solve a SDP which is computationally expensive. Compared to existing DC algorithms, the proposed methods simultaneously meet all the three basic requirements for clustering. We demonstrate the advantages of the proposed algorithms on a wide range of real world clustering tasks.

2. Extreme learning machine

Consider a supervised classification problem where we have a training set with N samples, $\{\boldsymbol{X}, \boldsymbol{Y}\} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$. Here $\boldsymbol{x}_i \in \mathbb{R}^d$, $\boldsymbol{y}_i = [\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{im}]^\top$ is a m-dimensional binary vector such that $\boldsymbol{y}_{ij} = 1$ if $\boldsymbol{x}_i \in \mathcal{C}_j$, and $\boldsymbol{y}_{ij} = 0$ otherwise. Here d and m are the dimensions of input and output respectively.

Traditional supervised ELM learns a nonlinear classifier from the training data set in two stages (Huang et al., 2015; Huang, Wang, & Lan, 2011; Huang et al., 2004). The first stage is to map the training data into a feature space using randomly generated nonlinear activation functions. Typical activation functions include the sigmoid function and Gaussian function, as given below.

(1) Sigmoid function

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

(2) Gaussian function

$$g(\boldsymbol{x}; \boldsymbol{\theta}) = \exp(-b\|\boldsymbol{x} - \boldsymbol{a}\|); \tag{2}$$

where $\boldsymbol{\theta} = \{\boldsymbol{a}, b\}$ are the parameters of the mapping function and $\|\cdot\|$ denotes the Euclidean norm.

A notable feature of ELMs is that the parameters of the hidden mapping functions can be randomly generated according to any continuous probability distribution, e.g., the uniform distribution on $(-1, 1)$. This makes ELMs distinct from the traditional feedforward neural networks and SVMs. The only free parameters that need to be optimized in the training process are the output weights between the hidden neurons and the output nodes. As a consequence, training ELMs is equivalent to solving a regularized least squares problem which is considerately more efficient than training SVMs or learning with backpropagation (BP) (Rumelhart, Hinton, & Williams, 1986).

In the first stage, a number of hidden neurons which map the data from the input space into a l-dimensional feature space (l is the number of hidden neurons) are randomly generated. We denote by $\boldsymbol{h}(\boldsymbol{x}_i) \in \mathbb{R}^{1\times l}$ the output vector of the hidden layer with respect to $\boldsymbol{x}_i$, and $\boldsymbol{\beta} \in \mathbb{R}^{l\times m}$ the output weights that connect the hidden layer with the output layer. Then, the outputs of the network are given by

$$\boldsymbol{f}(\boldsymbol{x}_i) = \boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta}, \quad i = 1, \ldots, N. \tag{3}$$

In the second stage, we solve the output weights by minimizing the sum of the squared losses of the prediction errors, which leads to the following formulation

$$\begin{aligned}&\min_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\|\boldsymbol{e}_i\|^2 \\ &\text{s.t.} \quad \boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta} = \boldsymbol{y}_i^T - \boldsymbol{e}_i^T, \quad i = 1, \ldots, N,\end{aligned} \tag{4}$$

where the first term in the objective function is a regularization term against over-fitting, $\boldsymbol{e}_i \in \mathbb{R}^m$ is the error vector with respect to the ith training pattern, and $C > 0$ is a penalty coefficient on the training errors.

By substituting the constraints into the objective function, we obtain the following equivalent unconstrained optimization problem:

$$\min_{\boldsymbol{\beta}} L_{ELM} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\|\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{\beta}\|^2 \tag{5}$$

where $\boldsymbol{H} = [\boldsymbol{h}(\boldsymbol{x}_1)^T, \ldots, \boldsymbol{h}(\boldsymbol{x}_N)^T]^T \in \mathbb{R}^{N\times l}$.

The above problem is widely known as the ridge regression or regularized least squares, for which we have the following closed form solution:

$$\boldsymbol{\beta}^* = \left(\boldsymbol{H}^T\boldsymbol{H} + \frac{\boldsymbol{I}_l}{C}\right)^{-1}\boldsymbol{H}^T\boldsymbol{Y}, \tag{6}$$

where $\boldsymbol{I}_l$ is an identity matrix of dimension l.

If the number of training patterns is smaller than the number of hidden neurons, i.e., $N < l$, we have the following equivalent solution which is however more efficient to solve:

$$\boldsymbol{\beta}^* = \boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{H}^T + \frac{\boldsymbol{I}_N}{C}\right)^{-1}\boldsymbol{Y} \tag{7}$$

where $\boldsymbol{I}_N$ is an identity matrix of dimension N.

Therefore, in the case where training patterns are plentiful compared to the hidden neurons, we use (6) to compute the output weights, otherwise we use (7).

3. Related discriminative clustering methods

Before introducing the proposed ELM-based discriminative clustering approaches, in this section we briefly review several related works which inspire our work.

3.1. Iterative ELM clustering

Zhang et al. (2013) proposed an iterative ELM algorithm (*IterELM*) for clustering, which is initiated by a traditional clustering method, and then repeatedly trains ELM on the clustered data. Similar to the iterative support vector regression method for MMC (Zhang et al., 2007), this process is able to minimize the discrimination between different classes, yielding reasonable clustering results. However, training ELM iteratively may lead to imbalanced results since the size of some clusters may shrink gradually while other clusters may grow overly large. Therefore, Zhang et al. (2013) proposed several heuristics to balance the clusters at each iteration. In our approach, we introduce a more principled method to avoid imbalanced clustering by adopting the recently proposed weighted ELM (Huang et al., 2014; Zong et al., 2013).

3.2. Discriminative clustering via LDA

Fisher's linear discriminant analysis (LDA) is an elegant supervised dimension reduction and classification method (Bishop et al., 2006). Recently, the connection between LDA and k-means has been investigated, and an unsupervised extension to LDA has been proposed by Ding and Li (2007). The discriminative clustering algorithm *DisCluster* proposed in Ding and Li (2007) alternates between performing k-means and subspace selection using LDA, and repeat the process until convergence. Essentially, the *DisCluster* is to optimize the following objective function:

$$\max_{\boldsymbol{U},\boldsymbol{Y}} \operatorname{Tr}\Big(\big(\boldsymbol{U}^\top\boldsymbol{\Sigma}_w(\boldsymbol{Y})\boldsymbol{U}\big)^{-1}\boldsymbol{U}^\top\boldsymbol{\Sigma}_b(\boldsymbol{Y})\boldsymbol{U}\Big), \tag{8}$$

where $\boldsymbol{U} \in \mathbb{R}^{d\times(m-1)}$ is a linear transformation matrix, and $\boldsymbol{Y} \in \mathbb{B}^{N\times m}$ is the label matrix. The between-class scatter matrix $\boldsymbol{\Sigma}_b$ and within-class scatter matrix $\boldsymbol{\Sigma}_w$ are respectively given by

$$\boldsymbol{\Sigma}_b = \sum_{k=1}^{m} N_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^\top, \tag{9}$$

$$\boldsymbol{\Sigma}_w = \sum_{k=1}^{m}\sum_{i\in\mathcal{I}_k}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top, \tag{10}$$

where $\mathcal{I}_k$ is the index set for cluster k, $\boldsymbol{\mu}_k$ is the center of cluster k, and $\boldsymbol{\mu}$ is the center of all sample points.

It can be observed that both $\boldsymbol{\Sigma}_b$ and $\boldsymbol{\Sigma}_w$ are functions of the label matrix $\boldsymbol{Y}$, since $\boldsymbol{Y}$ decides which cluster a sample is assigned to. For high dimensional data, a ridge term $\lambda\boldsymbol{I}_d$ ($\boldsymbol{I}_d$ is the identity matrix of dimension d) is added to the within-class scatter matrix to avoid numeric problems.

The *DisCluster* algorithm solves $\boldsymbol{Y}$ and $\boldsymbol{U}$ alternatively (Ding & Li, 2007). Specifically, it first fixes $\boldsymbol{U}$, and solves for $\boldsymbol{Y}$ via k-means; then with fixed $\boldsymbol{Y}$, optimizes $\boldsymbol{U}$ via LDA. The process is repeated until a local minimum is found. This simple algorithm works surprisingly well in practice and converges very fast. One limitation of *DisCluster* is that it is only efficient for linear problems, while its kernel extension for nonlinear clustering tasks has a relatively high computational complexity. In this paper, we introduce ELM, which is a highly efficient nonlinear model, to address this issue.

3.3. Discriminative k-means

The discriminative k-means (*DisKmeans*) proposed by Ye et al. (2007) has an identical objective as the *DisCluster* introduced above. However, *DisKmeans* searches for the optimal clustering in a different manner.

Define the *total scatter matrix* as

$$\boldsymbol{\Sigma}_t = \sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top. \tag{11}$$

It can be shown that $\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_b + \boldsymbol{\Sigma}_w$, thus the objective function in (8) is equivalent to

$$\max_{\boldsymbol{U},\boldsymbol{Y}} \operatorname{Tr}\Big(\big(\boldsymbol{U}^\top\boldsymbol{\Sigma}_t\boldsymbol{U}\big)^{-1}\boldsymbol{U}^\top\boldsymbol{\Sigma}_b(\boldsymbol{Y})\boldsymbol{U}\Big). \tag{12}$$

For simplicity, we assume that the training data are centered at the origin ($\sum_{i=1}^{N}\boldsymbol{x}_i = 0$). It can be verified that $\boldsymbol{\Sigma}_t = \boldsymbol{X}^\top\boldsymbol{X}$, and $\boldsymbol{\Sigma}_b = \boldsymbol{X}^\top\boldsymbol{L}\boldsymbol{L}^\top\boldsymbol{X}$, where $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ is the centered training data matrix, and $\boldsymbol{L} = \boldsymbol{Y}(\boldsymbol{Y}^\top\boldsymbol{Y})^{-\frac{1}{2}}$ is known as the *weighted cluster indicator matrix*. Thus the above formulation can be expressed as

$$\max_{\boldsymbol{U},\boldsymbol{L}} \operatorname{Tr}\Big(\big(\boldsymbol{U}^\top(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I}_d)\boldsymbol{U}\big)^{-1}\boldsymbol{U}^\top\boldsymbol{X}^\top\boldsymbol{L}\boldsymbol{L}^\top\boldsymbol{X}\boldsymbol{U}\Big), \tag{13}$$

where $\lambda\boldsymbol{I}_d$ is a regularization term added to the total scatter matrix.

Algorithm 1 The ELMC^{Iter} algorithm

```
Input: The training data $\boldsymbol{X}$;
Output: The label matrix $\boldsymbol{Y}$.
    Step 1: Run $k$-means on the training data $\boldsymbol{X}$ to get the initial label matrix $\boldsymbol{Y}$.
    Step 2: Initialize an ELM network with $n_h$ hidden neurons.
    Step 3: Calculate the output matrix using the W-ELM (17) based on the current label matrix $\boldsymbol{Y}$.
    Step 4: Update the label matrix $\boldsymbol{Y}$ based on the prediction of W-ELM.
    Step 5: Go to Step 3 until the label matrix $\boldsymbol{Y}$ converges or an maximum allowable number of iterations is reached.
    Return The label matrix $\boldsymbol{Y}$.
```

According the Representer Theorem (Schölkopf & Smola, 2001), the optimal transformations matrix $\boldsymbol{U}$ can be expressed as $\boldsymbol{U} = \boldsymbol{X}^\top \boldsymbol{V}$. Therefore, we can rewrite the objective in (13) as

$$\max_{\boldsymbol{V},\boldsymbol{L}} \mathrm{Tr}\Big(\big(\boldsymbol{V}^\top(\boldsymbol{G}\boldsymbol{G}+\lambda\boldsymbol{G})\boldsymbol{V}\big)^{-1}\boldsymbol{V}^\top\boldsymbol{G}\boldsymbol{L}\boldsymbol{L}^\top\boldsymbol{G}\boldsymbol{V}\Big), \tag{14}$$

where $\boldsymbol{G} = \boldsymbol{X}\boldsymbol{X}^\top$ is the so called Gram matrix which is symmetric and semi-positive definite.

With regards to the formulation (14), Ye et al. (2007) showed that the transformation matrix $\boldsymbol{V}$ can be factored out, thus simplifying the optimization problem as follows:

$$\max_{\boldsymbol{L}} \mathrm{Tr}\Big(\boldsymbol{L}^\top(\boldsymbol{I}_N-(\boldsymbol{I}_N+\boldsymbol{G}/\lambda)^{-1})\boldsymbol{L}\Big). \tag{15}$$

As shown in Dhillon, Guan, and Kulis (2004), the trace maximization problem with respect to $\boldsymbol{L}$ is equivalent to solving a kernel k-means (Schölkopf, Smola, & Müller, 1998) problem with the kernel Gram matrix given by

$$\boldsymbol{K} = \boldsymbol{I}_N - (\boldsymbol{I}_N + \boldsymbol{G}/\lambda)^{-1}. \tag{16}$$

Therefore, the clustering label matrix $\boldsymbol{Y}$ can be obtained via kernel k-means.

4. Discriminative clustering using ELM

In this section, we show how to extend ELM for discriminative clustering. Three novel clustering algorithms are presented: the iterative weighted ELM approach (ELMC^{Iter}), the LDA-based method using alternating optimization (ELMC^{LDA}) and the kernel k-mean-based method using non-alternating optimization (ELMC^{KM}).

4.1. Iterative weighted ELM for clustering

The iterative weighted ELM clustering method (ELMC^{Iter}) is inspired by the MMC algorithm via iterative support vector regression (Zhang et al., 2007) and the iterative ELM clustering algorithm (Zhang et al., 2013). However, different from the *IterELM*, we adopt weighted ELM (W-ELM) (Zong et al., 2013) to avoid imbalanced clustering. Specifically, we proposed to minimize the following objective function with respect to both the output weights $\boldsymbol{\beta}$ and the label matrix $\boldsymbol{Y}$,

$$\min_{\boldsymbol{\beta},\boldsymbol{Y}} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{\boldsymbol{C}(\boldsymbol{Y})}{2}\|\boldsymbol{Y}-\boldsymbol{H}\boldsymbol{\beta}\|^2, \tag{17}$$

where $\boldsymbol{C}$ is a diagonal matrix with the ith element on the diagonal being the weight associated with sample $\boldsymbol{x}_i$, i.e., $C_i = \lambda(\bar{N}/N_i)^p$. Here N_i is the size of the cluster that sample x_i belongs to, $\bar{N}$ is the average cluster size, $p \geq 0$ is a weight degree, λ is the penalty coefficient on the training error, and $\boldsymbol{C}$ is a function of the label matrix $\boldsymbol{Y}$.

Algorithm 2 The ELMC^{LDA} algorithm

```
Input: The training data: $\boldsymbol{X}$;
Output: The label vector of cluster index $\boldsymbol{y}$.
    Step 1: Initiate an ELM network of $n_h$ hidden neurons. Denote by $\overline{\boldsymbol{H}}$ the centered hidden layer output matrix.
    Step 2: Treat each row of $\overline{\boldsymbol{H}}$ as a sample, and run $k$-means on $\overline{\boldsymbol{H}}$ to get the initial predictions $\boldsymbol{y}$.
    Step 3: Run supervised LDA on the pseudo-labeled data set $\{\overline{\boldsymbol{H}}, \boldsymbol{y}\}$ to obtain the projection matrix $\boldsymbol{\beta}$.
    Step 4: Run $k$-means on the output matrix $\overline{\boldsymbol{H}}\boldsymbol{\beta}$ to obtain the cluster index vector $\boldsymbol{y}$.
    Step 5: Go to Step 4 until the cluster index vector $\boldsymbol{y}$ converges or an maximum allowable number of iterations is reached.
    Return The label vector of cluster index $\boldsymbol{y}$.
```

It is obvious that for samples from small clusters (whose sizes are under the average size), we have the corresponding coefficient $C_i > 1$, while for samples from large clusters (whose sizes are above the average size), we have that $C_i < 1$. Hence, the effect of majority clusters will be down-weighted while the minority clusters will be associated with larger weights. By doing so, ELMC^{Iter} is able to yield balanced and accurate clustering results, given that the weight degree p is chosen properly.

In summary, the proposed ELMC^{Iter} algorithm first 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 updating the class information based on the prediction of W-ELM until convergence. We summarize the ELMC^{Iter} algorithm as Algorithm 1.

4.2. ELM clustering based on LDA

Inspired by the *DisCluster* algorithm (Ding & Li, 2007), we extend ELM for discriminative clustering based on LDA. The idea is to perform LDA and k-means in the output space of ELM alternatively. Since the transformation matrix learned by LDA is a linear mapping, it can be absorbed by the output weight matrix of ELM, and we can directly learn the output weight $\boldsymbol{\beta}$ by performing LDA on the hidden layer output of ELM.

Basically, the hidden layer output matrix $\boldsymbol{H}$ can be viewed as the new data matrix, and its within-class and between-class scatter matrices can be computed similarly as that in standard LDA. However, the expressions in Section 3.3 are based on the assumption that the data matrix is centered with zero mean, while the hidden layer output matrix $\boldsymbol{H}$ does not necessarily meet this condition (even the original data are preprocessed), i.e., we usually have $\sum_{i=1}^{N}\boldsymbol{h}_i \neq 0$. In order to compactly write out the objective function of the proposed clustering algorithm, which is denoted by ELMC^{LDA}, we first introduce the following projection matrices:

$$\boldsymbol{P}_w = \boldsymbol{I}_N - \boldsymbol{L}\boldsymbol{L}^\top, \qquad \boldsymbol{P}_b = \boldsymbol{L}\boldsymbol{L}^\top - \boldsymbol{1}_N\boldsymbol{1}_N^\top/N, \tag{18}$$

where $\boldsymbol{1}_N$ is a vector of dimension N with all its elements equal to 1, and $\boldsymbol{L}$ is the weighted cluster indicator matrix introduce in Section 3.3.

Denote by $\overline{\boldsymbol{H}} = \boldsymbol{P}_t\boldsymbol{H}$ the *centered* hidden layer output matrix of ELM, where $\boldsymbol{P}_t = \boldsymbol{I}_N - \boldsymbol{1}_N\boldsymbol{1}_N^\top/N$ is a projection matrix. It is obvious that $\boldsymbol{P}_t = \boldsymbol{P}_w + \boldsymbol{P}_b$, and it can be verified that the ith row of $\overline{\boldsymbol{H}}$ is given by $\bar{\boldsymbol{h}}_i = \boldsymbol{h}_i - \sum_{j=1}^{N}\boldsymbol{h}_j/N$.

With the above expressions, the within-class and between-class scatter matrices are respectively given by $\boldsymbol{H}^\top\boldsymbol{P}_w\boldsymbol{H}$ and $\boldsymbol{H}^\top\boldsymbol{P}_b\boldsymbol{H}$. This leads to the following objective for our proposed ELMC^{LDA} algorithm:

$$\max_{\boldsymbol{\beta},\boldsymbol{L}} \mathrm{Tr}\Big(\big(\boldsymbol{\beta}^\top\boldsymbol{H}^\top\boldsymbol{P}_w(\boldsymbol{L})\boldsymbol{H}\boldsymbol{\beta}\big)^{-1}\boldsymbol{\beta}^\top\boldsymbol{H}^\top\boldsymbol{P}_b(\boldsymbol{L})\boldsymbol{H}\boldsymbol{\beta}\Big). \tag{19}$$

Algorithm 3 The ELMC^{KM} algorithm

```
Input: The training data $\boldsymbol{X}$;
Output: The label vector of cluster index $\boldsymbol{y}$.
    Step 1: Initiate an ELM network of $n_h$ hidden neurons.
    Step 2: Calculate the centered hidden layer output matrix $\overline{\boldsymbol{H}}$.
    Step 3: Compute the kernel gram matrix $\boldsymbol{K}$ based on the centered matrix $\overline{\boldsymbol{H}}$ using (20) or (21).
    Step 4: Run kernel $k$-means with kernel gram matrix $\boldsymbol{K}$ to obtain the cluster index $\boldsymbol{y}$.
    Return The label vector of cluster index $\boldsymbol{y}$.
```

To solve this optimization, we first train the output weights $\boldsymbol{\beta}$ of ELM with a fixed $\boldsymbol{L}$. This actually corresponds to solving a generalized eigenvalue problem (GEP) (Friedberg, Insel, & Spence, 1989). As the second step, we perform k-means in the ELM output space to learn $\boldsymbol{L}$ with fixed $\boldsymbol{\beta}$. This process is repeated until a local minimum is found. Details of ELMC^{KM} is summarized in Algorithm 2. Since LDA minimizes the within-class distortion, and maximizes between class discrimination, the algorithm is able to find cluster structure in the ELM feature space. Different from *DisCluster* which adopts linear subspace selection, the proposed ELMC^{LDA} nonlinearly transforms the data into new space (ELM output space) for clustering. Note that this approach is different from directly performing clustering in the ELM feature space, since ELMC^{LDA} needs to solve the output weights of ELM and regularization techniques can be introduced.

4.3. ELM clustering based on kernel k-means

The objective function in ELMC^{LDA} involves the optimization with the output weights $\boldsymbol{\beta}$ and the weighted cluster indicator matrix $\boldsymbol{L}$. Motivated by the *DisKmeans* algorithm (Ye et al., 2007), we can actually factor out $\boldsymbol{\beta}$ in (19), and greatly reduce the number of optimization variables.

Following the analysis in Section 3.3 and the derivation in Ye et al. (2007), the optimal $\boldsymbol{L}$ in (19) can be obtained via solving a kernel k-means with the following kernel matrix:

$$\boldsymbol{K} = \boldsymbol{I}_N - (\boldsymbol{I}_N + \overline{\boldsymbol{H}}\overline{\boldsymbol{H}}^\top/\lambda)^{-1}, \tag{20}$$

where $\overline{\boldsymbol{H}}$ is the centered hidden layer output matrix of ELM, whose ith row is given by $\bar{\boldsymbol{h}}_i = \boldsymbol{h}_i - \sum_{j=1}^N \boldsymbol{h}_j/N$.

Note that for large data sets, it will be inefficient to compute the kernel matrix since it requires to invert a N-by-N matrix. Fortunately, by the Woodbury identity, we can show that the kernel matrix in (20) is equivalent to:

$$\boldsymbol{K} = \overline{\boldsymbol{H}}(\overline{\boldsymbol{H}}^\top\overline{\boldsymbol{H}} + \lambda\boldsymbol{I}_l)^{-1}\overline{\boldsymbol{H}}, \tag{21}$$

which only needs to invert a l-by-l matrix.

Therefore, when $N \leq l$, we use (20) to compute the kernel matrix, otherwise we adopt (21). The ELMC^{KM} algorithm is summarized as Algorithm 3.

5. Experimental results

In this section, we empirically study the properties of the three proposed algorithms on real world data sets, and compare their performance with six related clustering algorithms.

5.1. Data sets

We evaluated the proposed algorithms using a wide range of data sets as summarized in Table 1. The number of classes in our experiment ranges from 2 to 10, the number of instances ranges up to 6435, and the number of features ranges up to 256. All of them are from real world applications, e.g., protein site identification, object recognition and image segmentation, and are therefore widely used as benchmarks for evaluating unsupervised learning algorithms.

Table 1
Description of datasets.

Data sets	# classes	# features	# instances
Iris	3	4	150
Wine	3	13	178
Glass	6	9	214
Ecoli	8	7	336
Diabetes	2	8	768
Vehicle	4	18	846
LetterA-B	2	16	1555
USPST	10	256	2007
Segment	7	19	2310
Satimage	6	36	6435

Nine data sets including Iris, Wine, Glass, Ecoli, Diabetes, Vehicle, LetterA–B, Segment and Satimage are from UCI data repository (Bache & Lichman, 2013). The LetterA–B data set is a subset (consisting of only the first two classes, i.e., A and B) of a letter recognition problem. The USPST data set is the test set of the well-known USPS data set, i.e., recognition of handwritten digits from envelopes by the US Postal Service. Each feature is normalized to [−1, 1] using min–max normalization.

5.2. Experimental setups

In our experiment, we removed the category label information and used all the ten data sets for clustering tasks. The criteria used for measuring the clustering result is the widely used clustering Accuracy (ACC) which is defined in terms of true category label information:

$$ACC = \frac{\sum_{i=1}^{N} \delta(y_i, map(c_i))}{N}, \tag{22}$$

where N is the number of instances, y_i and c_i are the true category label and the predicted category label of pattern x_i, respectively, $\delta(y_i, c)$ is a function that equals to 1 if $y = c$ or equals to 0 otherwise, and $map(\cdot)$ is an optimal permutation function that maps each cluster label to a category label by Hungarian algorithm, such that the resulting ACC is maximized.

We applied the three proposed algorithms to cluster all data sets. For comparison purposes, we also conducted experiments using six other clustering algorithms:

(1) k-means (Hartigan & Wong, 1979). The cluster centroids were initialized as k observations from the data at random.

(2) ELM k-means (He et al., 2014). The clustering results were obtained by conducting k-means clustering on the data in ELM feature space.

(3) Unsupervised Extreme Learning Machine (US-ELM) (Huang et al., 2014). The number of nearest neighbor and the regularization parameter were selected from candidate sets $\{1, 5, 10\}$ and $\{10^{-6}, 10^{-5}, \ldots, 10^{6}\}$ respectively. Other parameters were fixed at their default values for all data sets: A nearest neighbor graph on Euclidean distance with binary weight was used. The 1st degree graph was directly used without normalization. The final results were obtained by conducting a k-means clustering on the data in the three-dimensional embedding space.

Table 2
Performance comparison of the proposed ELMC^{Iter}, ELMC^{LDA}, ELMC^{KM} and related algorithms.

Data Set	k-means	ELM k-means	US-ELM	IterSVR	ELMC^{Iter}	DisCluster	ELMC^{LDA}	DisKmeans	ELMC^{KM}
Iris	83.03 ± 12.16	83.63 ± 11.52	86.93 ± 7.38	–	85.40 ± 3.59	**89.59 ± 15.26**	84.33 ± 12.05	82.28 ± 11.33	83.63 ± 10.11
Wine	95.27 ± 0.39	94.56 ± 1.04	**96.98 ± 0.49**	–	96.13 ± 0.64	96.65 ± 5.76	95.53 ± 5.37	94.91 ± 7.20	96.27 ± 0.93
Glass	42.76 ± 1.54	43.37 ± 2.48	**51.05 ± 3.86**	–	46.99 ± 1.47	45.47 ± 4.25	43.80 ± 3.74	45.09 ± 3.65	44.38 ± 2.68
Ecoli	53.83 ± 5.76	54.75 ± 5.65	63.31 ± 4.02	–	**77.64 ± 2.17**	54.01 ± 5.40	58.19 ± 4.26	65.49 ± 6.79	62.15 ± 4.95
Diabetes	67.29 ± 0.65	67.17 ± 0.33	67.39 ± 2.66	67.22 ± 0.26	**68.03 ± 0.54**	67.61 ± 0.49	67.80 ± 0.07	67.45 ± 0.00	67.24 ± 0.33
Vehicle	37.09 ± 0.73	37.50 ± 1.24	**43.41 ± 1.59**	–	41.82 ± 2.35	38.04 ± 1.58	39.83 ± 1.99	41.14 ± 6.36	40.05 ± 4.53
LetterA-B	93.70 ± 0.00	93.75 ± 0.18	94.18 ± 0.80	94.41 ± 0.00	**94.47 ± 0.01**	94.41 ± 0.00	94.41 ± 0.02	94.12 ± 0.03	93.93 ± 0.13
USPST	65.94 ± 2.42	64.95 ± 4.16	65.43 ± 4.90	–	**70.30 ± 3.93**	69.13 ± 4.01	66.76 ± 3.91	68.23 ± 3.64	68.47 ± 3.37
Segment	61.04 ± 5.42	60.65 ± 6.49	64.17 ± 7.33	–	**76.66 ± 4.30**	66.59 ± 6.81	66.70 ± 6.29	67.55 ± 5.53	69.74 ± 4.79
Satimage	66.81 ± 0.30	68.96 ± 0.84	**75.17 ± 6.64**	–	73.79 ± 0.22	67.99 ± 0.08	69.05 ± 0.87	68.56 ± 1.38	69.91 ± 3.03

(4) Iterative Support Vector Regression (IterSVR) (Zhang et al., 2007). The implementation was the same as (Zhang et al., 2007). More specifically, ϵ as in ϵ-insensitive loss was set to 0.05. The Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/\sigma^2)$ was used with the width σ selected from $\{10^{-6}, 10^{-5}, \ldots, 10^6\}$. The regularization parameter C was fixed at 500. The class balance parameter was set to $l = 0.03n$ for balanced data sets, and $l = 0.15n$ for imbalanced ones.

(5) Discriminative Clustering via LDA (DisCluster) (Ding & Li, 2007). The regularization parameter λ was selected from a candidate set $\{10^{-6}, 10^{-5}, \ldots, 10^6\}$ based on the clustering performance.

(6) Discriminative k-means (DisKmeans) (Ye et al., 2007). The regularization parameter λ was selected from a candidate set $\{10^{-6}, 10^{-5}, \ldots, 10^6\}$ based on the clustering performance.

For a fair comparison among the proposed and related algorithms, we standardized parameter settings and initialization and termination conditions for iterative methods as follows: The number of hidden neurons in all ELM related methods was set to 1000 for Iris and Wine data sets, and 2000 for the rest data sets. The regularization parameter λ in the proposed algorithms was selected from $\{10^{-6}, 10^{-5}, \ldots, 10^6\}$ based on averaged clustering accuracy over 50 independent runs. The weight degree p in ELMC^{Iter} was selected from $\{0.5, 1, 2\}$. For iterative methods, i.e., DisCluster, ELMC^{Iter} and ELMC^{LDA}, the class labels were initialized with outputs of k-mean procedure. All the iterative algorithms were terminated on the condition that there was no change in the predicted labels. A zero clustering accuracy was recorded for ELMC^{Iter}, when the algorithm failed to converge within 50 iterations. We reported the averaged clustering accuracy over 50 independent runs for all the algorithms.

In summary, we optimized no more than two parameters for each method. The second parameter was always selected from a smaller candidate set with only three elements. We anticipate that such setting would provide us with insights on the clustering performance without bias.

5.3. Results analysis

Comparison with related algorithms. Table 2 presents the averaged clustering results with standard deviation. Multi-class data sets that cannot be handled by IterSVR are marked with "–". ELMC^{Iter}, ELMC^{LDA} and ELMC^{KM} outperform the baseline methods, k-means and ELM k-means, on most data sets. ELMC^{LDA} and ELMC^{KM} achieve comparable results with their counterparts, i.e., DisCluster and DisKmeans, but lower results compared to US-ELM and ELMC^{Iter}. It is observed that the performance of ELMC^{LDA} is lower than DisCluster on some data sets. One explanation is that LDA is known to perform well in situations where the dimension of the data is much smaller than the sample size. ELMC^{LDA} projects the data to a high dimensional ELM feature space, which makes the within-class scatter matrix prone to be singular for data sets with small sample size, e.g., Iris and Wine. This may weaken the advantage brought by ELM nonlinear transformation.

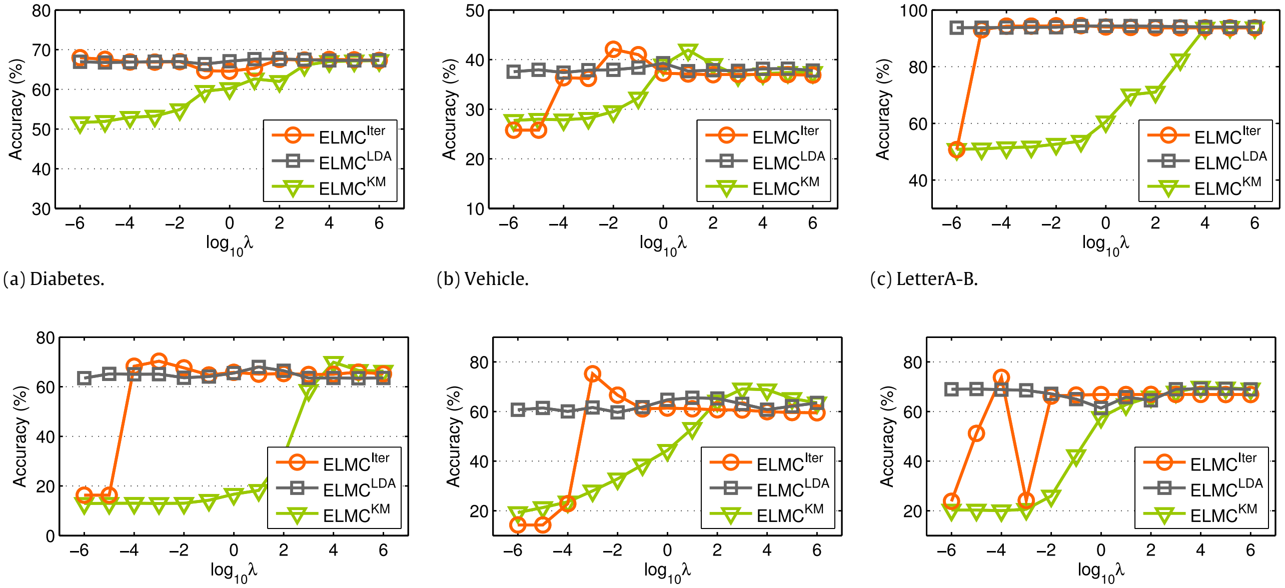
ELMC^{Iter} performs better than its counterpart IterSVR on both binary clustering tasks. The performance of US-ELM is generally high on image data sets, e.g., Satimage and Vehicle. Because US-ELM is developed under manifold regularization framework, and is expected to work well on data sets with manifold structure, e.g., image data. On data sets with large number of classes, such as Segment and Ecoli, ELMC^{Iter} yields significantly higher accuracy than all the other algorithms. The result also shows that ELMC^{Iter} yields overall smaller variance in the clustering accuracy, which demonstrates the robustness of ELMC^{Iter} with respect to class label initialization and random hidden neurons.

***Effect of the regularization parameter* $\boldsymbol{\lambda}$**. Fig. 1 shows the accuracy of the three proposed algorithms with different regularization parameter λ values on the six data sets with $>$500 samples. We can observe that λ has a significant impact on the performance of ELMC^{Iter} and ELMC^{KM}, while the performance of ELMC^{LDA} is relatively less sensitive to the value of λ. Though ELMC^{Iter} generally achieves high clustering accuracy under the optimal λ value, unusual local minimum points are observed on Diabetes and Satimage data sets. This may be caused by the zero clustering accuracy reported in the trials when ELMC^{Iter} failed to converge within 50 iterations. This implies that ELMC^{LDA} is preferred in problems where efficient model selection is not feasible, and ELMC^{Iter} should be applied only if λ is carefully selected.

Effect of the iterative process. Fig. 2 presents the clustering accuracy across different iterations in one trial produced by ELMC^{Iter} and ELMC^{LDA} on six larger data sets. We observe both algorithms are generally able to improve the clustering results over iterations and converge within a reasonable number of iterations under optimal parameter settings. ELMC^{LDA} converges faster than ELMC^{Iter} on most data sets.

6. Conclusion

In this paper, we have investigated the problem of extending ELM to discriminative clustering, and proposed three novel clustering methods, i.e., ELMC^{Iter}, ELMC^{LDA} and ELMC^{Iter}. Although ELM-based clustering is not new, the proposed algorithms have remarkable advantages over existing methods. Particularly, comparing with the methods which directly perform clustering in the ELM feature space, our methods have the flexibility of adopting regularization techniques in solving ELM output weights to improve clustering accuracy and robustness. Comparing with the recently proposed manifold learning based ELM clustering method in Huang et al. (2014), the clustering methods introduced in this paper have fewer parameters to tune. Empirically, we have demonstrated the advantages of the introduced methods on a variety of real world data sets. The proposed discriminative clustering algorithms also have the potential to be adapted for online learning, and we leave this for future research.



(a) Diabetes.

(b) Vehicle.

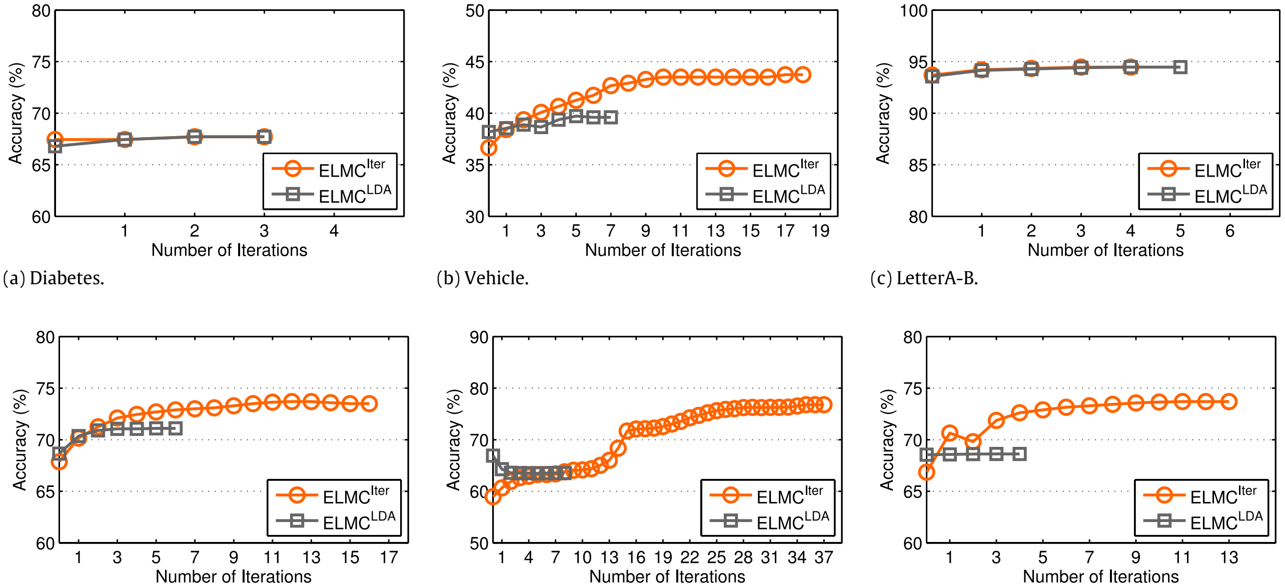
(c) LetterA-B.

(d) USPST.

(e) Segment.

(f) Satimage.

Fig. 1. Effect of the regularization parameter λ on ELMC^{Iter}, ELMC^{LDA}, ELMC^{KM}.



(a) Diabetes.

(b) Vehicle.

(c) LetterA-B.

(d) USPST.

(e) Segment.

(f) Satimage.

Fig. 2. Evolution of clustering accuracy in ELMC^{Iter}, ELMC^{LDA} as a function of iterations.

Acknowledgments

G. Huang, S. Song and C. Wu were partially supported by the National Natural Science Foundation of China under Grant Nos. 41427806 and 61273233, the Research Fund for the Doctoral Program of Higher Education under Grant Nos. 20120002110035 and 20130002130010, the Project of China Ocean Association under Grant DY125-25-02, and Tsinghua University Initiative Scientific Research Program under Grant No. 20131089300. T. Liu's research was supported by the Singapore Academic Research Fund (AcRF) Tier 1 under Project Grant No. RG 80/12 (M4011092). Y. Yang's research was partially supported by the National Science Foundation of China under Grant No. 61101216, and the 111 Project under Grant No. 111-2-05.

References

Bache, K., & Lichman, M. (2013). UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

Belkin, M., & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Neural information processing systems. Vol. 14* (pp. 585–591).

Bishop, C. M., et al. (2006). *Pattern recognition and machine learning. Vol. 1*. New York: Springer.

Butcher, J. B., Verstraeten, D., Schrauwen, B., Day, C. R., & Haycock, P. W. (2013). Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural Networks*, *38*, 76–89.

Cao, J., Chen, T., & Fan, J. (2014). Fast online learning algorithm for landmark recognition based on bow framework. In *2014 IEEE 9th conference on industrial electronics and applications*. June (pp. 1163–1168).

Cao, J., Chen, T., & Fan, J. (2015). Landmark recognition with compact BoW histogram and ensemble ELM. *Multimedia Tools and Applications*, 1–19. URL: http://dx.doi.org/10.1007/s11042-014-2424-1.

Cao, J., & Xiong, L. (2014). Protein sequence classification with improved extreme learning machine algorithms. *BioMed Research International*, *2014*.

De la Torre, F., & Kanade, T. (2006). Discriminative cluster analysis. In *International conference on machine learning (ICML)* (pp. 241–248). ACM.

Dhillon, I., Guan, Y., & Kulis, B. (2004). *A unified view of kernel k-means, spectral clustering and graph cuts. Tech. Rep.* Department of Computer Sciences, University of Texas at Austin.

Ding, C., & Li, T. (2007). Adaptive dimension reduction using discriminant analysis and *k*-means clustering. In: *International conference on machine learning* (pp. 521–528).

Friedberg, S. H., Insel, A. J., & Spence, L. E. (1989). *Linear algebra* (2nd ed.). Prentice Hall.

Gokcay, E., & Principe, J. C. (2002). Information theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(2), 158–171.

Gomes, R., Krause, A., & Perona, P. (2010). Discriminative clustering by regularized information maximization. In *Neural information processing systems* (pp. 775–783).

Hartigan, J. A., & Wong, M. A. (1979). A *k*-means clustering algorithm. *Applied Statistics*, 100–108.

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

Huang, G., Huang, G.-B., Song, S., & You, K. (2015). Trends in extreme learning machines: A review. *Neural Networks*, *61*, 32–48.

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

Huang, G.-B., Wang, D. H., & Lan, Y. (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, *2*(2), 107–122.

Huang, G., Zhang, J., Song, S., & Zheng, C. (2015b). Maximin separation probability clustering. In *Twenty-ninth AAAI conference on artificial intelligence*.

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *42*(2), 513–529.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In *International joint conference on neural networks. Vol. 2* (pp. 985–990). IEEE.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Kasun, L. L. C., Liu, T., Yang, Y., Lin, Z., & Huang, G.-B. (2015). Extreme learning machine for clustering. In J. Cao, K. Mao, E. Cambria, Z. Man, & K.-A. Toh (Eds.), *Proceedings in adaptation, learning and optimization*: *Vol. 3. Proceedings of ELM-2014 volume 1* (pp. 435–444). Springer International Publishing.

Liu, X. Y., Gao, C. H., & Li, P. (2012). A comparative analysis of support vector machines and extreme learning machines. *Neural Networks*, *33*, 58–66.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Neural information processing systems. Vol. 14* (pp. 849–856).

Niu, G., Dai, B., Shang, L., & Sugiyama, M. (2013). Maximum volume clustering: A new discriminative clustering approach. *The Journal of Machine Learning Research*, *14*(1), 2641–2687.

Punj, G., & Stewart, D. W. (1983). Cluster analysis in marketing research: review and suggestions for application. *Journal of Marketing Research*, 134–148.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagation errors. *Nature*, *323*, 533–536.

Schölkopf, B., & Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computing*, *10*(5), 1299–1319.

Shi, J., Cai, Y., Zhu, J., Zhong, J., & Wang, F. (2013). Semg-based hand motion recognition using cumulative residual entropy and extreme learning machine. *Medical & Biological Engineering & Computing*, *51*(4), 417–427. URL: http://dx.doi.org/10.1007/s11517-012-1010-9.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. In *KDD workshop on text mining*.

Sugiyama, M., Yamada, M., Kimura, M., & Hachiya, H. (2011). On information-maximization clustering: Tuning parameter selection and analytic solution. In *International conference on machine learning* (pp. 65–72).

Valizadegan, H., & Jin, R. (2007). Generalized maximum margin clustering and unsupervised kernel learning. In *Neural information processing systems. Vol. 19* (p. 1417).

Xu, L., Neufeld, J., Larson, B., & Schuurmans, D. (2004). Maximum margin clustering. In: *Neural information processing systems. Vol. 16* (p. 2).

Xu, L., & Schuurmans, D. (2005). Unsupervised and semi-supervised multi-class support vector machines. In *AAAI conference on artificial intelligence. Vol. 5*.

Xu, R., & Wunsch, D. I. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*(3), 645–678.

Yang, Y., Wu, Q., Wang, Y., Zeeshan, K., Lin, X., & Yuan, X. (2014). Data partition learning with multiple extreme learning machines.

Ye, J., Zhao, Z., & Wu, M. (2007). Discriminative *k*-means for clustering. In *Neural information processing systems. Vol. 7* (pp. 1649–1656).

Zhang, K., Tsang, I. W., & Kwok, J. T. (2007). Maximum margin clustering made practical. In *International conference on machine learning (ICML)* (pp. 1119–1126). ACM.

Zhang, C., Xia, S., Liu, B., & Lei, Z. (2013). Extreme maximum margin clustering. *IEICE Transactions on Information and Systems*, *96*(8), 1745–1753.

Zhao, B., Wang, F., & Zhang, C. (2008). Efficient multiclass maximum margin clustering. In *International conference on machine learning (ICML)* (pp. 1248–1255). ACM.

Zong, W., Huang, G.-B., & Chen, Y. (2013). Weighted extreme learning machine for imbalance learning. *Neurocomputing*, *101*, 229–242.