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/

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

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