Neurocomputing 237 (2017) 185–192

Contents lists available at ScienceDirect

Neurocomputing

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

Hypergraph regularized sparse feature learning

Mingxia Liu[a], Jun Zhang[a,*], Xiaochun Guo[b], Liujuan Cao[c]

[a] *School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China*
[b] *School of Life Science and Technology, Xidian University, Xian, Shaanxi, China*
[c] *Department of Computer Science, School of Information Science and Engineering, Xiamen University, Xiamen, China*

ARTICLE INFO

Communicated by Ke Lu

Keywords:
Feature selection
Hypergraph
Sparse
Classification

ABSTRACT

As an important pre-processing stage in many machine learning and pattern recognition domains, feature selection deems to identify the most discriminate features for a compact data representation. As typical feature selection methods, Lasso and its variants using the l_1-norm based regularization have received much attention in recent years. However, most of existing l_1-norm based sparse feature selection methods ignore the structure information of data or only consider the pairwise relationships among samples. In this paper, we propose a hypergraph regularized sparse feature learning method, where the high-order relationships among samples are modeled and incorporated into the learning process. Specifically, we first construct a hypergraph with multiple hyperedges to capture the high-order relationships among samples, followed by the computation of a hypergraph Laplacian matrix. Then, we propose a hypergraph regularization term, and a hypergraph regularized Lasso model. We conduct a series of experiments on a number of data sets from UCI machine learning repository, and two real-world neuroimaging based classification tasks. Experimental results demonstrate that the proposed method achieves promising classification results, compared with several well known feature selection approaches.

1. Introduction

During the past decades, the rapid advances in data collection and storage capabilities have led to an information overload in many machine learning and pattern recognition domains. Researchers working in domains, such as computer vision, neuroimaging analysis, biology and remote sensing, are facing larger and larger observations and simulations [1–4]. Usually, the feature dimension is even much higher than the number of features, which is called 'small-sample-size' problem [1,5]. The high dimensional features will consume more computation and storage resources, and even may degrade the performances of learning algorithms, which is typically referred as 'the curse of dimensionality' [1]. For addressing the high-dimensional feature problem, various dimension reduction approaches are proposed, including feature extraction methods and feature selection methods [5]. Feature extraction approaches aim to find a lower dimensional representation to capture the content in the original data, according to some criterion [6]. In contrast, feature selection methods deem to find the most informative features from the original feature sets to find a more compact representation of the original data, which is the very focus of this study.

Generally, existing feature selection approaches can be roughly categorized into two classes, i.e., filter-type methods and wrapper-type methods [7]. Specifically, wrapper-type methods require a pre-defined learning algorithm to evaluate the performance on each candidate feature subset, and then determine the optimal feature subset according to the learning performance [8]. In contrast to wrapper-type methods, filter-type feature selection approaches directly select features according to some criterion (e.g., mutual information, correlation [9–12]), and involve no learning algorithm. Among a huge literature on feature selection methods, Laplacian Score (LS) [13], Fisher Score (FS) [14] and Constraint Score (CS) [15] are typical examples.

On the other hand, sparse feature learning methods (e.g., Lasso and its variants) have received increasing attention in feature selection domain, where the l_1-norm based regularization is adopted to encourage sparsity among feature weights [1,16–20]. In recent years, the l_1-norm based sparse feature selection methods have been widely used in various machine learning and pattern recognition domains, such as dimension reduction [21–24,11], imaging annotation [25], and objective categorization [26,27]. However, most of existing sparse feature learning methods seldom take advantage of the high-order relationships among samples that is a kind of important prior information. Intuitively, modeling the high-order relationship information can further boost the performance of feature learning models.

[*] Corresponding authors.
E-mail addresses: xdzhangjun@gmail.com (J. Zhang), caoliujuan@xmu.edu.cn (L. Cao).

http://dx.doi.org/10.1016/j.neucom.2016.10.031
Received 17 September 2015; Received in revised form 12 October 2016; Accepted 24 October 2016
Available online 27 October 2016

For addressing that problem, in this paper, we propose a hypergraph regularized sparse feature learning method, where a hypergraph Laplacian regularization is developed to explicitly model the high-order relationship information of data. Specifically, we first construct a hypergraph, by constructing multiple hyperedges that reflect the high-order relationships among samples. Then, we propose a hypergraph Laplacian regularization term, as well as a hypergraph regularized Lasso model. We conduct a series of experiments on a number of data sets from University of California-Irvine (UCI) machine learning repository [28], and two real-world neuroimaging based classification tasks on the baseline Alzheimer's Disease Neuroimaging Initiative (ADNI) database [29]. Experimental results demonstrate that the proposed method achieves promising classification results, compared with several established feature selection methods.

The rest of this paper is organized as follows. Section 2 introduces the related work on sparse feature selection and hypergraph learning. In Section 3, we present the proposed hypergraph regularized sparse feature learning method in details. We then introduce the experiments and related discussions in Section 4 and Section 5, respectively. Finally, we conclude this paper in Section 6.

2. Related work

2.1. Sparse feature selection

In the last two decades, sparse learning attracts much attention in pattern recognition and machine learning domains. As a typical sparse learning methods, Lasso [16] is a shrinkage and feature selection method for linear regression, and has been proven to be very popular and well studied for high-dimensional data [30–32,1]. Some reasons for the popularity might include 1) the entire regularization path of the Lasso can be computed efficiently, and 2) Lasso is able to handle more predictor variables than samples by producing sparse models [1,33].

Generally, Lasso is a penalized least squares method, which minimizes the usual sum of squared errors with the l_1-norm penalty on the weight vector (i.e., $\mathbf{w} \in \mathcal{R}^D$). Mathematically, the objective function of Lasso is defined as follows [16]:

$$\arg\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda_1 \|\mathbf{w}\|_1, \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D]$ is the $N \times D$ matrix whose rows consist of the D-dimensional fixed predictor variables $\mathbf{x}_n (n = 1, 2, \ldots, N)$, and $\mathbf{w} \in \mathcal{R}^D$ is the estimated weighted vector. The vector $\mathbf{y}$ contains the N-dimensional set of real valued observations of the response variable. Due to the sparsity nature of l_1-norm, the Lasso method can perform feature selection and regression/classification simultaneously.

Recently, several extensions of Lasso are proposed, following the work in [16]. Some typical approaches include elastic net [19], group Lasso [18], and fused Lasso [17]. It is worth noting that most existing l_1-norm based feature learning methods seldom consider the structure information of data. Intuitively, such structure information is one type of prior information that can benefit the subsequent learning problem at hand. Recently, researchers in [2] develop a manifold Laplacian regularized Lasso (called LapLasso in this study) method, where the Laplacian matrix based on the manifold assumption for data is adopted to guide the sparse feature learning method. Specifically, the LapLasso model is defined in the following [2]:

$$\arg\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \mathbf{w}^T \mathbf{X}^T \mathbf{L}^m \mathbf{X}\mathbf{w}, \tag{2}$$

where $\mathbf{L}^m$ is the manifold Laplacian matrix, which is defined as $\mathbf{L}^m = \mathbf{D}^m - \mathbf{S}^m$. Note that $\mathbf{S}^m$ is the similarity matrix with the element $S_{ij}^m = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma}}$, and $\mathbf{D}^m$ is a diagonal matrix with the element $D_{ii}^m = \sum_{j=1}^N S_{ij}^m$. Note that the last term in Eq. (2) is to preserve the pairwise relationship information of data during the process of mapping data in the original feature space into the label space. Although the LapLasso model considers the relationship information of data, it only focuses on the pairwise relationships, while the relationships among three or more samples (i.e., the high-order relationships) are not modeled.

2.2. Hypergraph learning

In the literature, graph learning [34,35] and hypergraph learning [36–38] have achieved promising performance in many applications. In graph learning, one sample is represented by a vertex in a graph, and one edge connects a pair of vertices based on some distance measure metric (e.g., Euclidean distance). Thus, only the pairwise relationships among samples can be captured in graph learning methods.

Different from conventional graph learning approaches, a hyperedge in a hypergraph can connect more than two vertices according to some criterion [36]. In this way, hypergraph learning can automatically model the complex (e.g., high-order) relationships of data, which is a very appealing property. Recently, hypergraph learning has been widely used in various applications. For instance, Gao et al. [37] develop a hypergraph-based 3D object retrieval and recognition method, and achieved state-of-the-art results. In this method, the relationship among different objects are formulated in the hypergraph structure. Based on different sample clustering results, multiple hypergraphs are constructed and the learning on multiple hypergraphs is jointly conducted to estimate the relevance among these objects. Hypergraph has been also investigated in hyperspectral image classification. In [39], both the spectral and the spatial correlations among samples are formulated in a hypergraph structure, and the learning on the hypergraph is conducted to classify different samples. To further extend this method, a bi-layer graph learning strategy is proposed in [40], where a simple graph layer is generated to learn the connection-based feature among samples, and a hypergraph layer is employed to further modeling these samples based on the output of the first layer.

The main contribution of this study is that we propose to incorporate the hypergraph regularizer into a sparse feature learning model, where the high-order relationships among samples can be modeled explicitly. To the best of our knowledge, this is the first work to consider the high-order relationship among samples in sparse feature learning. We evaluate the proposed method on both UCI data sets and a real-world data set, with results demonstrating the efficacy of our method.

3. Hypergraph regularized sparse feature learning

In this section, we first present the proposed hypergraph Laplacian regularization in Section 3.1, and then flesh out the proposed hypergraph regularized Lasso (HLasso) model in Section 3.2.

3.1. Hypergraph laplacian regularization

Fig. 1 shows the flowchart of our proposed hypergraph regularized sparse feature learning method. As can be seen from Fig. 1, we first construct a hypergraph using the input data matrix, and then compute the hypergraph Laplacian matrix. Next, we develop a hypergraph regularized sparse feature selection approach. Finally, we perform classification using the selected features. Throughout the paper, we denote boldface upper-case letters, boldface lower-case letters and normal italic letters as matrices, vectors and scalars, respectively. Table 1 summarizes important notation and their corresponding definitions used in the rest of this paper.

Given a vertex set $\mathcal{V}$ where each vertex represents a sample, a hyperedge set $\mathcal{E}$ with each one connecting two or more vertices, and a weight vector $\mathbf{a} = (a_i) \in \mathcal{R}^{N_e}$ for N_e hyperedges, a hypergraph is represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{a})$. It is worth noting, different from the edge in simple graph that connects only two vertices, a hyperedge in a

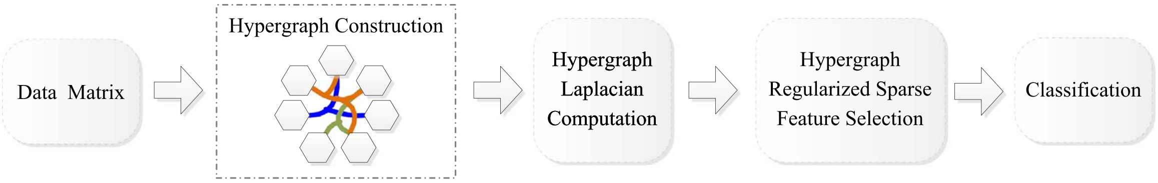

Data Matrix
Hypergraph Construction
Hypergraph Laplacian Computation
Hypergraph Regularized Sparse Feature Selection
Classification

Fig. 1. Flowchart of the proposed hypergraph regularized sparse feature learning method. There are four main steps, including (1) hypergraph construction, (2) hypergraph Laplacian computation and (3) Hypergraph regularized sparse feature selection, and (4) classification.

Table 1
List of important notation used in this paper.

Notation	Definition
$\mathbf{X}$	The data matrix $\mathbf{X} \in \mathcal{R}^{N\times D}$, where N is the sample size and D is the feature dimension.
$\mathbf{y}$	The class label vector for N subjects, i.e., $\mathbf{y} \in \mathcal{R}^{N}$.
$\mathbf{w}$	The weighting vector that mapping the data in original feature space into the label space, and $\mathbf{w} \in \mathcal{R}^{D}$.
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{a})$	$\mathcal{G}$ denotes a hypergraph, where $\mathcal{V}$, $\mathcal{E}$ and $\mathbf{a}$ represent the set of vertices, the set of hyperedges, and the weights of hyperedges, respectively.
N_e	The number of hyperedges, i.e., $N_e = \lvert\mathcal{E}\rvert$.
$\mathbf{A}$	The diagonal matrix of the hyperedge weights $\mathbf{A} \in \mathcal{R}^{N_e\times N_e}$, and $A_{ii} = a_i$.
$d(v)$	The degree of the vertex v.
$\delta(e)$	The degree of the hyperedge e.
$\mathbf{D}_v$	The diagonal matrix of the vertex degrees, and $\mathbf{D}_v \in \mathcal{R}^{N\times N}$.
$\mathbf{D}_e$	The diagonal matrix of the hyperedge degrees, and $\mathbf{D}_e \in \mathcal{R}^{N_e\times N_e}$.
$\mathbf{H}$	The incidence matrix for the hypergraph, and $\mathbf{H} \in \mathcal{R}^{N\times N_e}$.
$\mathbf{L}^m$	The manifold Laplacian matrix $\mathbf{L}^m \in \mathcal{R}^{N\times N}$.
$\mathbf{L}^h$	The hypergraph Laplacian matrix $\mathbf{L}^h \in \mathcal{R}^{N\times N}$.

hypergraph can connect more than two vertices, through which high-order relationships can be modeled explicitly [34,41]. Here, each hyperedge e_i $(i = 1,\ldots,N_e)$ is assigned a weighting value $a(e_i)$. For the hypergraph $\mathcal{G}$, its incidence matrix $\mathbf{H} \in \mathcal{R}^{N\times N_e}$ is generated to represent the relationships among different vertices. Specifically, the entry $h(v, e)$ of the incidence matrix $\mathbf{H}$ denotes whether the vertex v is connected with other vertices via the hyperedge e, which is defined as follows:

$$h(v, e) = \begin{cases} 1, & if \;\; v \in e \\ 0, & if \;\; v \notin e \end{cases}. \tag{3}$$

Based on the incidence matrix $\mathbf{H}$, the vertex degree of each vertex $v \in \mathcal{V}$ and hyperedge degree of the hyperedge $e \in \mathcal{E}$ are defined, respectively, as

$$d(v) = \sum_{e\in\mathcal{E}} w(e)h(v, e), \tag{4}$$

and

$$\delta(e) = \sum_{v\in\mathcal{V}} h(v, e). \tag{5}$$

In addition, we denote $\mathbf{D}_v$ and $\mathbf{D}_e$ as diagonal matrices of vertex degrees and hyperedge degrees, respectively, with elements defined as follows

$$D_v(i, j) = \begin{cases} d(i), & if \;\; i{=}{=}j \\ 0, & otherwise \end{cases}, \tag{6}$$

and

$$D_e(i, j) = \begin{cases} \delta(i), & if \;\; i{=}{=}j \\ 0, & otherwise \end{cases}. \tag{7}$$

Let $\mathbf{A}$ represent the diagonal matrix of hyperedge weights, with the diagonal element $A_{ii} = a_i$. Currently, there are various methods for computing the hypergraph Laplacian, which can be divided into two categories [36]. The first category aims to construct a simple graph from the original hypergraph, e.g., star expansion [42], clique expansion [42] and Rodriquez's Laplacian [43]. In the second category, a hypergraph Laplacian is defined by using the analogies from the simple graph Laplacian, e.g., normalized Laplacian [36] and Bolla's Laplacian [44]. It is reported that the aforementioned two categories are close to each other [45]. In this study, we adopt the method proposed in [36] to compute the hypergraph Laplacian. That is the positive semi-definite matrix $\mathbf{L}^h = \mathbf{I} - \boldsymbol{\Theta}$ is called the hypergraph Laplacian, where $\boldsymbol{\Theta} = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{A}\mathbf{D}_e^{-1/2}\mathbf{H}^T\mathbf{D}_v^{-1/2}$.

It is worth noting that, hyperedge construction is an essential process in hypergraph learning [41,44]. Following [36,37], we adopt a KNN technique to construct hyperedges. To be specific, we treat each vertex as a center, and compute its K nearest neighbors via the Euclidean distance between the center vertex and the other vertices. Then, a hyperedge is constructed by connecting that center vertex and its K-nearest neighbors. Given N samples, we can then construct N hyperedges. Following [37], we empirically assign equal weighting values to all hyperedges, i.e., $a_{e_i} = 1$ for the hyperedge e_i $(i = 1, 2,\ldots,N_e)$.

3.2. Hyergraph regularized lasso model

Based on the hypergraph Laplacian matrix, we now present our hypergraph regularization term, which is defined as follows

$$\Omega = \mathbf{w}^T\mathbf{X}^T\mathbf{L}^h\mathbf{X}\mathbf{w}. \tag{8}$$

The intuition of Eq. (8) is that we want to preserve the structure information of data in the original feature space, while such structure is high-order reflected by hypergraph Laplacian matrix $\mathbf{L}^h$. With the proposed hypergraph regularization term Ω, we now present our hypergraph regularized Lasso (HLasso for short) model as follows:

$$\arg\min_{\mathbf{w}} \frac{1}{2}\,\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda_1\,\|\mathbf{w}\|_1 + \lambda_2\mathbf{w}^T\mathbf{X}^T\mathbf{L}^h\mathbf{X}\mathbf{w}, \tag{9}$$

where $\mathbf{L}^h$ is the hypergraph Laplacian matrix. The first term in Eq. (9) is the empirical loss on the training data, the second term is the l_1-norm regularization, and the last term is the hypergraph Laplacian regularization.

With the proposed HLasso model, we can not only perform sparse feature selection, but also utilize the high-order structure information conveyed by the hypergraph Laplacian matrix $\mathbf{L}^h$. It is worth noting that our proposed HLasso model is different from conventional manifold Laplacian regularized Lasso model (i.e., LapLasso in Eq. (2)). That is, HLasso can explicitly model the high-order relationships among samples, while LapLasso can only capture the pairwise relationships.

Now, we introduce an efficient optimization algorithm for solving the objective function of HLasso defined in Eq. (9). It is straightforward to verify that the proposed objective function is convex but non-smooth because of the non-smooth l_1-norm regularization term. The basic idea to solve the problem is to use a smooth function to approximate the original non-smooth objective function, and then solve the former by utilizing some off-the-shelf fast algorithms. In this paper, we resort to the widely used accelerated proximal gradient (APG) method [46] to solve the proposed problem. For a fixed Q (i.e., the maximum iteration), the APG algorithm for the problem in Eq. (9) has $O(1/Q^2)$ asymptotic convergence rate. Then, we list the process of our hypergraph regularized sparse feature learning method in Algorithm 1.

Algorithm 1. The hypergraph regularized sparse feature learning method.

```
Input: Data matrix $\mathbf{X} \in \mathcal{R}^{N \times D}$; Label vector $\mathbf{y} \in \mathcal{R}^{N}$.
Initialization: Parameters $\lambda_1$ and $\lambda_2$.
Step 1. Construct a hypergraph $\mathcal{G}$, and compute the hypergraph
Laplacian matrix $\mathbf{L}^h$ according to the data matrix $\mathbf{X}$.
Step 2. Perform sparse feature selection using Eq.(9), by selecting
    features with non-zero coefficients in the learned weight vector
$\mathbf{w}$.
Step 3. Perform classification based on the data with only those
    selected features.
Output: Classification results.
```

4. Experiment

4.1. Data sets

First, we evaluate the efficacy of the proposed method on eight data sets from the UCI machine learning repository [28]. These data sets have small or middle size of feature numbers, with class numbers ranging from 2 to 6. The detailed information of those UCI data sets used in this study are shown in Table 2.

Then, we evaluate our method on two neuroimaging based classification tasks using the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (http://adni.loni.usc.edu) [29]. This database have included 202 subjects with the magnetic resonance imaging (MRI). There are three categories of subjects, including 51 AD patients, 99 mild cognitive impairment (MCI) patients and 52 normal controls (NCs). Two classification tasks are performed, including AD vs. NC classification and MCI vs. NC classification. For each subject, we adopt the voxel-based morphometry (VBM) to extract voxel-based gray matter (GM) density features from its MR imaging. The whole image processing step follows the standard VBM protocol. Particularly, we adopt the t-test to select the informative features.

4.2. Experimental setting

We compare the proposed method with five popular feature selection approaches, which are described as follows.

- Laplacian Score (LS) [13]. Laplacian Score is a simple unsupervised feature selection method, with a key assumption that the data points from the same class should be close to each other. LS prefers features with larger variances as well as stronger locality preserving ability.
- Fisher Score (FS) [14]. As a supervised feature selection method, Fisher Score needs full class labels of samples. This method seeks features that can maximize the distance of data points between different classes and minimize the distance of data points within the same class simultaneously.
- Constraint Score (CS) [15]. This method is semi-supervised, where pairwise constraints are used to guide the feature selection process. Specifically, Constraint Score performs feature selection according to the constraint preserving ability of features. It utilizes $\mathbf{M} = \{(\mathbf{x}_i, \mathbf{x}_j)|\mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same class}\}$ containing pairwise must-link constraints and $\mathbf{C} = \{(\mathbf{x}_i, \mathbf{x}_j)|\mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to different classes}\}$ containing pairwise cannot-link constraints as the supervision information.
- Lasso [16]. As a typical sparse feature selection method, Lasso performs feature selection through the l_1-norm, where features corresponding to zero coefficients in the weight vector will be discarded [16].
- Manifold Laplacian regularized Lasso (LapLasso). LapLasso performs feature selection via the l_1-norm regularization, which is similar to Lasso. In particular, LapLasso adopts a manifold Laplacian regularization to preserve the structure information of data in the original feature space [2].

Table 2
Statistics of UCI data sets used in the experiments.

Name	Feature Dimension	Class Number	Sample Number
Dermatology	34	6	361
Diabetes	8	2	768
Hepatitis	19	2	155
Ionosphere	33	2	351
Parkinson	22	2	195
Vehicle	18	4	846
Wdbc	14	2	569
Wine	13	3	178

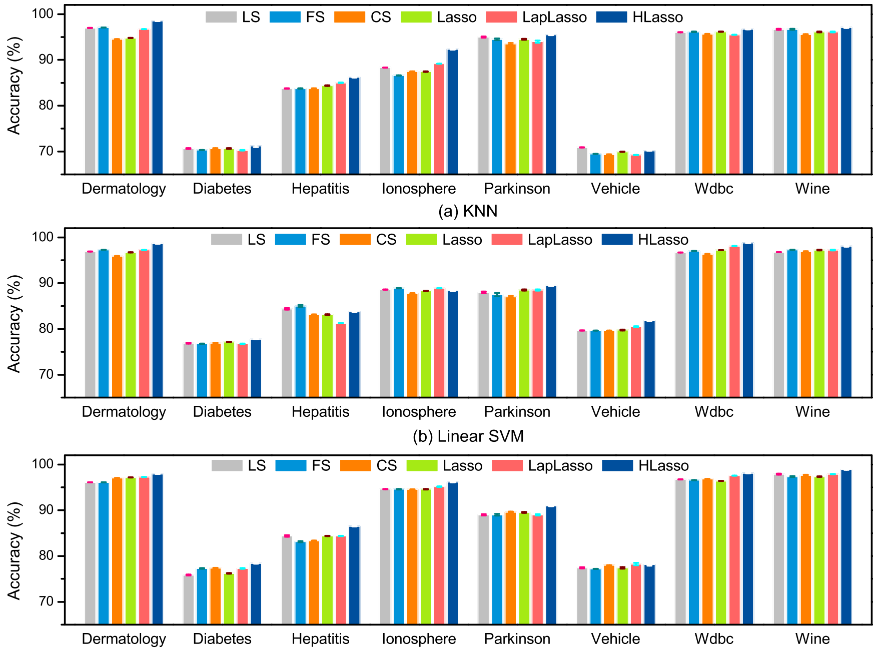
Generally, we adopt a 5-fold cross validation strategy to compute the classification accuracy. To be specific, we first partition the whole data set into 5 subsets (each subset with roughly equal size), and each time one of these subsets is utilized as the test set while the other 4 subsets are combined together to be the training set. In addition, to determine the number of optimal feature dimensions in each fold, we further perform feature selection via an inner cross-validation on training data (i.e., another 5-fold cross-validation is performed on training data). Note that the test data is not used to determine the selected features, and not used for parameter selection. Then, the mean and the standard deviation of classification accuracies on the test set using such optimal feature subset are reported.

In the experiments, three classifiers are used to perform classification tasks based on the selected features achieved by different feature selection methods. The first one is the K-nearest neighborhood (KNN) classifier with Euclidean distance and K=1, the second one is a linear support vector machines (Linear SVM) with the default regularization parameters value (i.e., C=1) [47], and the third one is the SVM classifier with RBF kernel (RBF SVM) with a heat kernel. The bandwidth parameter of RBF kernel in RBF SVM is selected from $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ via inner cross-validation on the training data.

For feature ranking methods (i.e., LS, FS and CS), we first select the first d features from the ranking list of features generated by corresponding algorithms, where d is the desired number of selected features specified as $d = 1, 2, \ldots, D$ in the experiments. Then, we report the highest classification accuracy as well the number of selected features for LS, FS, and CS. For sparse feature selection methods (i.e., Lasso, LapLasso and the proposed HLasso method), the optimal feature subset is determined on the training data through corresponding algorithms, and the classification results on the test data are reported using such fixed feature subsets. Following [15,23], the parameter λ for CS is set to be 0.1 empirically. The regularization parameters λ for Lasso are chosen from $\{10^{-6}, 10^{-5}, \ldots, 10^{0}\}$ through inner 5-fold cross validation on the training data. Similarly, the parameters λ_1 and λ_2 in LapLasso and our HLasso method are selected from the same range by inner cross validation on the training data. As mentioned in Section 2, we adopt the KNN technique for constructing hyperedges, where the neighbor size K is selected from the range $\{3, 5, 7, 11, 15, 25, 35, 50\}$ through inner cross validation on the training data.

4.3. Results on UCI data sets

In the first group of experiments, we evaluate the performance of our proposed method and those compared methods (including LS, FS, CS, Lasso and LapLasso) on eight data sets from the UCI machine learning repository. In Fig. 2, we report the mean classification



(a) KNN

(b) Linear SVM

(c) RBF SVM

Fig. 2. Classification results achieved by the proposed method and the compared methods on UCI data sets, using (a) KNN classifier, (b) Linear SVM classifier, and (c) RBF SVM classifier.

accuracies as well as corresponding variances, by using KNN, Linear SVM and RBF SVM classifiers.

From Fig. 2, one can observe two main points. First, regardless of the using of different classifiers, our proposed HLasso method usually achieves the overall better performance than the compared methods. For instance, on the Dermatology data set, the accuracy achieved by HLasso is 98.52%, while the best accuracy of the compared methods is only 97.08% (achieved by FS) using KNN classifier. Second, among three l_1-norm based sparse feature selection algorithms, HLasso performs better than Lasso and LapLasso in most cases. In particular, HLasso usually outperforms LapLasso in eight UCI data sets with three classifiers. It indicates that the higher-order relationship information can further boost the performance of conventional sparse feature learning models that only model the pairwise relationships among samples.

In addition, we list the average number of selected features achieved by different feature selection methods among those 5-fold cross validation, with results shown in Table 3. From Table 3, one can see that the number of selected features achieved by our HLasso method is similar to the compared methods. Recall the classification results shown in Fig. 2, we can find that using similar number of selected features, the proposed HLasso method can achieve more accurate classification accuracy, compared with the other five methods (i.e., LS, FS, CS, Lasso and LapLasso).

Table 3
Number of selected features on UCI data sets.

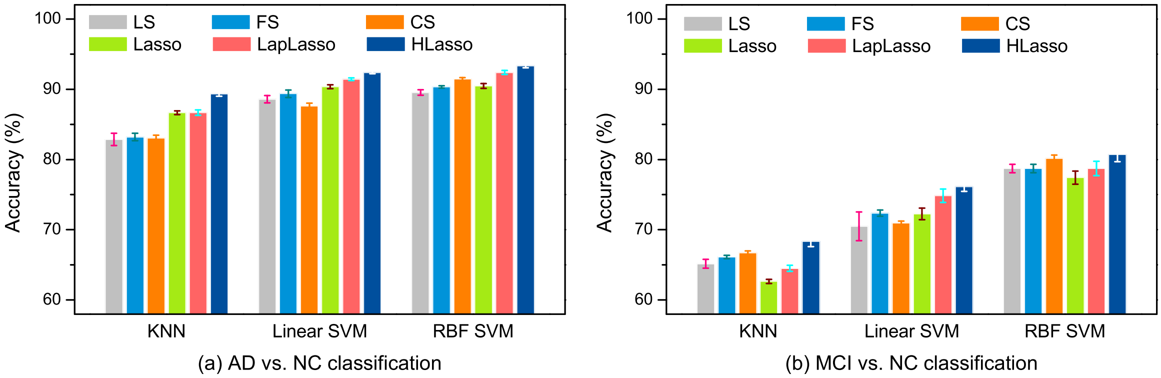
		LS	FS	CS	Lasso	LapLasso	HLasso
KNN	Dermatology	25	22	28	34	33	32
	Diabetes	8	6	8	8	6	5
	Hepatitis	19	19	19	19	18	18
	Ionosphere	32	29	29	33	28	15
	Parkinson	15	21	23	21	19	21
	Vehicle	17	10	18	18	17	17
	Wdbc	18	20	18	29	27	26
	Wine	11	12	10	12	12	13
Linear SVM	Dermatology	25	22	26	33	32	33
	Diabetes	8	6	8	8	6	7
	Hepatitis	1	15	12	19	17	18
	Ionosphere	32	30	35	32	31	32
	Parkinson	12	18	20	21	21	21
	Vehicle	17	18	17	18	17	17
	Wdbc	26	25	26	29	27	26
	Wine	8	5	8	13	13	12
RBF SVM	Dermatology	25	25	27	34	33	33
	Diabetes	8	6	6	8	6	6
	Hepatitis	1	18	13	18	18	16
	Ionosphere	31	31	26	33	30	26
	Parkinson	11	21	15	19	20	18
	Vehicle	16	18	15	18	17	17
	Wdbc	21	27	29	29	28	26
	Wine	12	9	9	13	12	12

4.4. Results on real-world database

In the second group of experiments, we evaluate our proposed method on the ADNI database. Specifically, two classification tasks are performed, including AD vs. NC classification and MCI vs. NC classification, with experimental results shown in Fig. 3. From Fig. 3, we can observe that in AD vs. NC classification, our HLasso method consistently outperforms the compared methods, using KNN, Linear SVM and RBF SVM classifiers. For instance, with KNN classifier, HLasso achieves an accuracy of 89.33%, while the best results achieved by the compared methods is only 86.67% achieved by Lasso and LapLasso. Similarly, the proposed HLasso method perform better than the compared methods in MCI vs. NC classification. These results further demonstrate the superiority of our HLasso method, where the high-order information is incorporated into the sparse feature selection process.

5. Discussion

In this section, we first investigate the influence of neighbor size (for constructing the hyperedge) on the performance of the proposed HLasso method (Section 5.1), and then study the influence of two parameters on the performance of HLasso (Section 5.2).



(a) AD vs. NC classification

(b) MCI vs. NC classification

Fig. 3. Classification results achieved by the proposed method and the compared methods on the ADNI data set for (a) AD vs. NC classification and (b) MCI vs. NC classification.

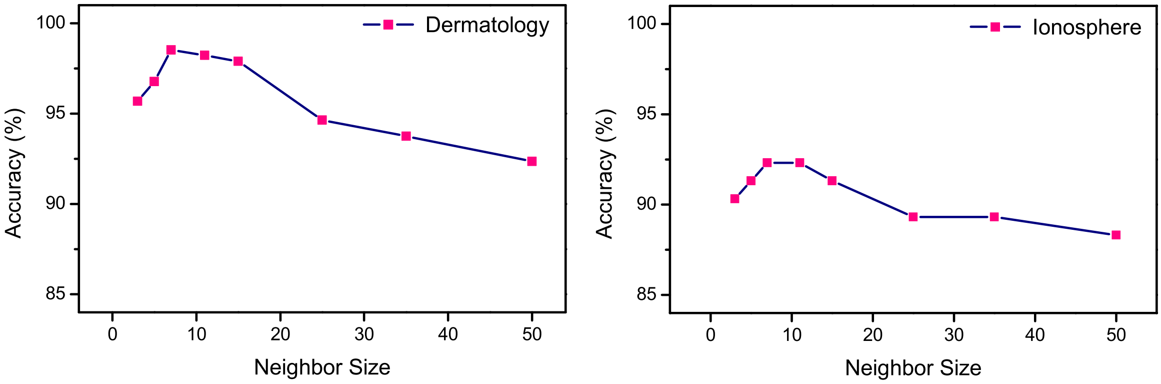


Fig. 4. Classification results achieved by the proposed method using different neighbor sizes for hyperedge construction on the Dermatology (left) and the Ionosphere (right) data sets.

5.1. Influence of neighbor size

As mentioned in Section 3.1, we construct a hypergraph by using a KNN technique for hyperedge generation. Now we investigate the influence of neighbor size in KNN on the performance of our proposed method. Fig. 4 reports the results achieved by our HLasso method on Dermatology and Ionosphere data sets, where the neighbor size varies in the range {3, 5, 7, 11, 15, 25, 35, 50}. As can be seen from Fig. 4, on the Dermatology data set, HLasso achieves the best performance using the neighbor size 7. On the Ionosphere data set, HLasso adopts the neighbor size 7 (or 11) to obtain the best results. When the neighbor size is large than 15, the performance degrade rapidly. The underlying reason could be that, with a large neighbor size, only the global structure information (other than the local structure information) can be modeled, which is not sufficient to reflect the true data structure.

5.2. Influence of parameters

Furthermore, we investigate the influence of two parameters (i.e., λ_1 and λ_2) on the proposed HLasso method. In Fig. 5, we show the results of HLasso on the Dermatology data set using KNN classifier, where λ_1 and λ_2 vary in the range $\{10^{-6}, 10^{-5}, \ldots, 10^{0}\}$. As can be seen from Fig. 5, the performance of the proposed HLasso method has some fluctuations by using different values for two parameters. When λ_1 and λ_2 lie in the range $\{10^{-4}, 10^{-3}, 10^{-2}\}$, HLasso achieves the best result, and such performance is stable. These results imply that the selection of parameters is important for our proposed method, which is also an open problem in sparse learning. For choosing the optimal parameters, one can adopt the cross-validation strategy, as suggested in [1].

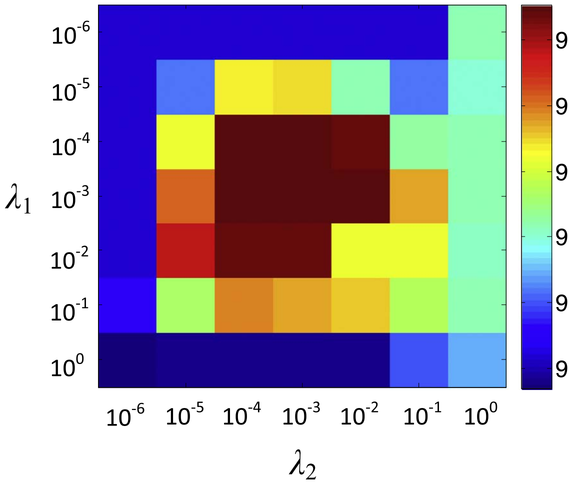


Fig. 5. Classification results achieved by the proposed method using different parameters (i.e., λ_1 and λ_2) on the Dermatology data set using KNN classifier.

6. Conclusion

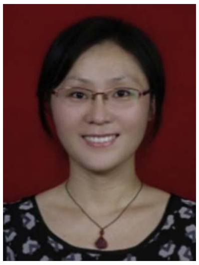
In this paper, we propose a hypergraph regularized sparse feature learning method, where the higher-order relationships among samples can be modeled explicitly and incorporated into the sparse feature selection process. Specifically, we first construct a hypergraph based on the input data matrix. Then, we develop a hypergraph Laplacian regularization term, and a hypergraph regularized Lasso model. We evaluate our method on a series of data sets, with experimental results demonstrating the efficacy of the proposed method. In the currrent work, we simply assign equal weights to each hyperedge in the constructed hypergraph. Since different hyperedges may contain different information of data structure, it is interesting to learn different weights for different hyperedges, which will be our future work. In addition, besides class labels, there are many other kinds of weak supervision information (such as pairwise constraints). In the future work, it is also interesting to investigate how to integrate the pairwise constraints into our proposed hypergraph regularized sparse feature learning approach.

Acknowledgment

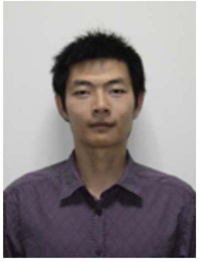
This study was supported by the National Natural Science Foundation of China (61402320).

References

[1] T. Hastie, R. Tibshirani, M. Wainwright, Statistical learning with sparsity: the lasso and generalizations, CRC Press, 2015.
[2] C. Li, H. Li, Network-constrained regularization and variable selection for analysis of genomic data, Bioinformatics 24 (9) (2008) 1175–1182.
[3] J. Zhang, H. Zhao, J. Liang, Continuous rotation invariant local descriptors for texton dictionary-based texture classification, Comput. Vis. Image Underst. 117 (1) (2013) 56–75.
[4] J. Zhang, J. Liang, H. Zhao, Local energy pattern for texture classification using self-adaptive quantization thresholds, IEEE Trans. Image Process. 22 (1) (2013) 31–42.
[5] A.R. Webb, Statistical Pattern Recognition, John Wiley & Sons, NJ, USA, 2003.
[6] B. Scholkopft, K.-R. Mullert, Fisher discriminant analysis with kernels, Neural Netw. Signal Process. IX 1 (1) (1999) 1.
[7] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[8] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[9] N. Kwak, C.-H. Choi, Input feature selection by mutual information based on Parzen window, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1667–1671.
[10] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: Proceedings of International Conference on Machine Learning, Vol. 3, 2003, pp. 856–863.
[11] M. Liu, D. Zhang, Sparsity score: a novel graph-preserving feature selection method, Int. J. Pattern Recognit. Artif. Intell. 28 (04) (2014) 1450009.
[12] M. Liu, L. Miao, D. Zhang, Two-stage cost-sensitive learning for software defect prediction, IEEE Trans. Reliab. 63 (2) (2014) 676–686.
[13] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems, 2005, pp. 507–514.
[14] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
[15] D. Zhang, S. Chen, Z.-H. Zhou, Constraint score: a new filter method for feature selection with pairwise constraints, Pattern Recognit. 41 (5) (2008) 1440–1451.
[16] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (Methodol.) (1996) 267–288.
[17] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, K. Knight, Sparsity and smoothness via the fused lasso, J. R. Stat. Soc.: Ser. B (Stat. Methodol.) 67 (1) (2005) 91–108.
[18] L. Meier, S. Van De Geer, P. Bühlmann, The group lasso for logistic regression, J. R. Stat. Soc.: Ser. B (Stat. Methodol.) 70 (1) (2008) 53–71.
[19] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc.: Ser. B (Stat. Methodol.) 67 (2) (2005) 301–320.
[20] L. An, X. Chen, S. Yang, B. Bhanu, Sparse representation matching for person re-identification, Inf. Sci. 355–356 (2016) 74–89.
[21] W. Liu, H. Zhang, D. Tao, Y. Wang, K. Lu, Large-scale paralleled sparse principal component analysis, Multimed. Tools Appl. (2014) 1–13.
[22] N. Kwak, Principal component analysis based on L1-norm maximization, IEEE Trans. Pattern Anal. Mach. Intell. 30 (9) (2008) 1672–1680.
[23] M. Liu, D. Zhang, Pairwise constraint-guided sparse learning for feature selection, IEEE Trans. Cybern. 46 (1) (2016) 298–310.
[24] X. Shi, Z. Guo, Z. Lai, Y. Yang, Z. Bao, D. Zhang, A framework of joint graph embedding and sparse regression for dimensionality reduction, IEEE Trans. Image Process. 24 (4) (2015) 1341–1355.
[25] W. Liu, D. Tao, J. Cheng, Y. Tang, Multiview hessian discriminative sparse coding for image annotation, Comput. Vis. Image Underst. 118 (2014) 50–60.
[26] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[27] U. Srinivas, Y. Suo, M. Dao, V. Monga, T.D. Tran, Structured sparse priors for image classification, IEEE Trans. Image Process. 24 (6) (2015) 1763–1776.
[28] A. Asuncion, D. Newman, UCI machine learning repository (2007).
[29] C.R. Jack, M.A. Bernstein, N.C. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P.J. Britson, J.L. Whitwell, C. Ward, et al., The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods, J. Magn. Reson. Imaging 27 (4) (2008) 685–691.
[30] K. Knight, W. Fu, Asymptotics for Lasso-type estimators, Ann. Stat. (2000) 1356–1378.
[31] M.J. Wainwright, Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (Lasso), IEEE Trans. Inf. Theory 55 (5) (2009) 2183–2202.
[32] P. Zhao, B. Yu, On model selection consistency of Lasso, J. Mach. Learn. Res. 7 (2006) 2541–2563.
[33] M.R. Osborne, B. Presnell, B.A. Turlach, On the Lasso and its dual, J. Comput. Graph. Stat. 9 (2) (2000) 319–337.
[34] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, Adv. Neural Inf. Process. Syst. 16 (16) (2004) 321–328.
[35] F. Wang, C. Zhang, Label propagation through linear neighborhoods, IEEE Trans. Knowl. Data Eng. 20 (1) (2008) 55–67.
[36] D. Zhou, J. Huang, B. Schölkopf, Learning with hypergraphs: Clustering, classification, and embedding, in: Advances in Neural Information Processing Systems, 2006, pp. 1601–1608.
[37] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3-D object retrieval and recognition with hypergraph analysis, IEEE Trans. Image Process. 21 (9) (2012) 4290–4303.
[38] T. Jin, J. Yu, J. You, K. Zeng, C. Li, Z. Yu, Low-rank matrix factorization with multiple hypergraph regularizer, Pattern Recognit. 48 (3) (2015) 1011–1022.
[39] R. Ji, Y. Gao, R. Hong, Q. Liu, D. Tao, X. Li, Spectral-spatial constraint hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 52 (3) (2014) 1811–1824.
[40] Y. Gao, R. Ji, P. Cui, Q. Dai, G. Hua, Hyperspectral image classification through bilayer graph-based learning, IEEE Trans. Image Process. 23 (7) (2014) 2769–2778.
[41] C. Berge, E. Minieka, Graphs and Hypergraphs, Vol. 7, North-Holland Publishing Company Amsterdam, 1973.
[42] J.Y. Zien, M.D. Schlag, P.K. Chan, Multilevel spectral hypergraph partitioning with arbitrary vertex sizes, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 18 (9) (1999) 1389–1399.
[43] J. Rodríguez, On the laplacian spectrum and walk-regular hypergraphs, Linear Multilinear Algebra 51 (3) (2003) 285–297.
[44] M. Bolla, Spectra, euclidean representations and clusterings of hypergraphs, Discret. Math. 117 (1) (1993) 19–39.
[45] S. Agarwal, K. Branson, S.Belongie, Higher order learning with graphs, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 17–24.
[46] Y. Nesterov, Introductory Lectures on Convex Optimization 87, Springer Science & Business Media, NY, USA, 2004.
[47] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27.



Mingxia Liu, received the B.S. degree and the M.S. degree from Shandong Normal University, Shandong, China, in 2003 and 2006, respectively, and the Ph.D. degree in Computer Science from Nanjing University of Aeronautics and Astronautics, China, in 2015. Her research interests include machine learning, pattern recognition, computer vision, and neuroimaging analysis. She served as the TPC member/reviewer for several outstanding journals and conferences, including IEEE Transactions on Cybernetics (TCYB), IEEE Transactions on Knowledge and Data Engineering (TKDE), PLOS ONE, Neurocomputing, AAAI 2017, and International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2015, MICCAI 2016) etc. She is also a guest editor of both Neurocomputing and Multimedia Tools and Applications.



Jun Zhang, was born in Shaanxi province, China. He received the B.S. in 2009 and Ph.D. in 2014 from Xidian University, Xi'an, China. He is a postdoc research associate with the School of Medicine in University of North Carolina at Chapel Hill. His research interests include pattern recognition, optical imaging, and medical image analysis. He is also a guest editor of Multimedia Tools and Applications.

Xiaochun Guo, received the M.S. degree from Shandong University of Science and Technology, Shandong, China, in 2007. She is currently a senior laboratory associate in Taishan University. Her research interests include machine learning, pattern recognition, and data mining.

Liujuan Cao, received the B.S. degree, the M.S. degree and the Ph.D. degree from the School of Computer Science and Technology, Harbin Engineering University. She was a visiting researcher at Columbia University from 2012 to 2013. She joined the School of Information Science and Engineering, Xiamen University at 2014, and she is currently an assistant professor. Her research interest covers computer vision, pattern recognition, remote sensing, and artificial intelligence. She has published over 40 papers in top and major tired journals and conferences, including AAAI, CVPR, IEEE Transactions on GRS, Information Sciences, Neurocomputing, Signal Processing, and Digital Signal Processing etc. She is the financial chair of IEEE MMSP 2015, workshop chair of ACM ICIMCS 2016, local chair of VALSE (visual and learning seminar) 2017.