Pairwise Constraint-Guided Sparse Learning for Feature Selection

Mingxia Liu and Daoqiang Zhang

***Abstract*—Feature selection aims to identify the most informative features for a compact and accurate data representation. As typical supervised feature selection methods, Lasso and its variants using L_1-norm-based regularization terms have received much attention in recent studies, most of which use class labels as supervised information. Besides class labels, there are other types of supervised information, e.g., pairwise constraints that specify whether a pair of data samples belong to the same class (must-link constraint) or different classes (cannot-link constraint). However, most of existing L_1-norm-based sparse learning methods do not take advantage of the pairwise constraints that provide us weak and more general supervised information. For addressing that problem, we propose a pairwise constraint-guided sparse (CGS) learning method for feature selection, where the must-link and the cannot-link constraints are used as discriminative regularization terms that directly concentrate on the local discriminative structure of data. Furthermore, we develop two variants of CGS, including: 1) semi-supervised CGS that utilizes labeled data, pairwise constraints, and unlabeled data and 2) ensemble CGS that uses the ensemble of pairwise constraint sets. We conduct a series of experiments on a number of data sets from University of California-Irvine machine learning repository, a gene expression data set, two real-world neuroimaging-based classification tasks, and two large-scale attribute classification tasks. Experimental results demonstrate the efficacy of our proposed methods, compared with several established feature selection methods.**

***Index Terms*—Feature selection, L_1-norm, pairwise constraint, sparse learning.**

Manuscript received August 31, 2014; revised December 2, 2014 and February 3, 2015; accepted February 6, 2015. Date of publication July 6, 2015; date of current version December 14, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61422204 and Grant 61473149, in part by the Jiangsu Natural Science Foundation for Distinguished Young Scholar of China under Grant BK20130034, in part by the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20123218110009, in part by the Nanjing University of Aeronautics and Astronautics Fundamental Research Funds under Grant NE2013105, and in part by the Jiangsu Natural Science Foundation of China under Grant BK20130813. This paper was recommended by Associate Editor D. Tao. *(Corresponding author: Daoqiang Zhang.)*

M. Liu is with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China, and also with the School of Information Science and Technology, Taishan University, Taian 271021, China.

D. Zhang is with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: dqzhang@nuaa.edu.cn).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. This includes a PDF file, which contains additional results of classification experiments as well as additional results using different parameters and constraint numbers. This material is 629 KB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2401733

I. Introduction

In MANY machine learning and data mining applications, numerous features can be extracted, and sometimes the feature dimension is even higher than the number of data points [1], [2]. Such high dimensional features not only consume more computation and storage resources, but also may degrade the performances of learning algorithms, which is typically referred as "the curse of dimensionality" [3]. Feature selection addresses this issue by selecting the most informative features for a compact and accurate data representation, and has been proven effective in reducing feature dimensionality, improving learning performances and facilitating data understanding [4]–[7].

According to the different use of supervised information, existing feature selection methods can be categorized into three groups, i.e., supervised, unsupervised, and semi-supervised ones [3], [7]–[11]. In the literature, most of supervised and semi-supervised feature selection methods use class labels as supervised information, while class labels are usually limited or expensive to be obtained. For example, in computer vision applications, it seems to be impossible to obtain enough labeled data to cover all object classes, especially when there is tens of thousands of categories [12], [13]. Actually, besides class labels, there are other types of supervised information, e.g., pairwise constraints that specify whether a pair of samples belong to the same class (must-link constraints) or different classes (cannot-link constraints) [14]. Compared with class labels that require to have detailed information about the category of sample, pairwise constraints simply mention for some pairs of samples that they are similar or dissimilar [15]–[17]. Currently, such kinds of constraints have been widely used in several fields of machine learning, such as distance metric learning [15]–[18], semi-supervised clustering [19]–[22], dimension reduction [23], [24], and feature selection [25]–[28]. Especially, several pairwise constraints-based feature selection studies have shown that using only a small amount of pairwise constraints can achieve comparable performance to those given by supervised feature selection methods using full class labels [25], [27].

On the other hand, Lasso and its variants have received increasing attention in feature selection domain, by using the L_1-norm-based regularization terms to encourage sparsity among feature weights [29]–[31]. In the literature, L_1-norm-based sparse learning methods have been used in various real-world applications, such as neuroimaging classification [32], object categorization [33], and dictionary learning [34]. However, to the best of our knowledge, most

TABLE I
LIST OF IMPORTANT NOTATION

Notation	Description	Notation	Description	Notation	Description
N	Number of samples	y_i	Label of $\boldsymbol{x}_i$	$\boldsymbol{S}^{\mathrm{M}}$	Similarity matrix defined on M
N_p	Number of samples in class p	$\boldsymbol{Y}$	Labels for the training data	$\boldsymbol{S}^{\mathrm{C}}$	Similarity matrix defined on C
N_E	Number of constraint sets	f_r	r-th feature vector	$\boldsymbol{D}$	Diagonal matrix
N_{M}	Number of must-link constraints	f_r^p	r-th feature vector of class p	$\boldsymbol{D}^{\mathrm{M}}$	Diagonal matrix defined on M
N_{C}	Number of cannot-link constraints	μ_r	Mean of the r-th feature	$\boldsymbol{D}^{\mathrm{C}}$	Diagonal matrix defined on C
d	Number of features	μ_r^p	Mean of the r-th feature in class p	$\boldsymbol{L}$	Laplacian matrix with $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$
P	Number of classes	M	Pairwise must-link constraint set	$\boldsymbol{L}^{\mathrm{M}}$	Laplacian matrix with $\boldsymbol{L}^{\mathrm{M}} = \boldsymbol{D}^{\mathrm{M}} - \boldsymbol{S}^{\mathrm{M}}$
$\boldsymbol{x}_i$	i-th sample	C	Pairwise cannot-link constraint set	$\boldsymbol{L}^{\mathrm{C}}$	Laplacian matrix with $\boldsymbol{L}^{\mathrm{C}} = \boldsymbol{D}^{\mathrm{C}} - \boldsymbol{S}^{\mathrm{C}}$
$\boldsymbol{X}$	Training data $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$	$\boldsymbol{S}$	Similarity matrix	$\boldsymbol{w}$	Weight vector

existing L_1-norm-based sparse learning methods use only class labels as supervised information, and do not take advantage of pairwise constraints that provide us weak and more general supervised information.

For addressing this problem, we propose a pairwise constraint-guided sparse (CGS) learning method for feature selection, where the must-link and the cannot-link constraints are used as discriminative regularizers that directly concentrated on the local discriminative structure of data. Furthermore, we develop two variants to the proposed CGS method. The first one is the semi-supervised CGS (SCGS) method that uses labeled data, pairwise constraints, and unlabeled data. The second one is the ensemble CGS (ECGS) method that uses the ensemble of pairwise constraint sets other than a single constraint set. We conduct a series of experiments on a number of data sets from University of California-Irvine (UCI) machine learning repository, a gene expression data set, and two real-world neuroimaging-based classification tasks. Experimental results validate the efficacy of our proposed methods.

The major contributions of this paper are threefold. First, we propose a novel pairwise CGS feature selection method, which exploits the pairwise constraint-based regularizers to reflect the local discriminative structure of data. Second, we develop a SCGS model by using labeled data, pairwise constraints, and unlabeled data, and an ECGS method by using an ensemble of multiple pairwise constraint sets. Third, we develop an efficient optimization algorithm for solving the proposed problem.

The remainder of this paper is organized as follows. Section II briefly reviews related background knowledge on feature selection. In Section III, we present the proposed CGS feature selection method, an efficient optimization algorithm, a semi-supervised variant of CGS, and an ensemble variant of CGS. Section IV provides the experimental results and analysis on a number of data sets, by comparing the proposed methods with several established feature selection methods. In Section V, we first discuss the influences of the parameters and the constraint number on the performances of CGS, and then compare our proposed methods with several sparse classifiers. Section VI concludes this paper.

II. BACKGROUND

In this section, we first briefly introduce several well-known supervised and unsupervised feature selection methods, including variance [35], Laplacian score (LS) [36], and Fisher score (FS) [35]. Then, we introduce recent work on pairwise constraint-based feature selection. Finally, we present related work on L_1-norm-based sparse feature selection.

A. Supervised and Unsupervised Feature Selection

Denote the training data set as $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ ($\boldsymbol{x}_i \in R^d$), where N is the number of data points and d is the feature dimension. Let f_{ri} denote the rth feature of sample $\boldsymbol{x}_i$. Define $\boldsymbol{\mu}_r = 1/N \sum_{i=1}^N f_{ri}$ as the mean of the rth feature $\boldsymbol{f}_r$. For supervised learning problems, class labels are given in $\boldsymbol{Y} = \{y_i\}_{i=1}^N$, $y_i \in \{1, \ldots, P\}$, where P is the class number and N_p denotes the number of data points belonging to the pth class. For convenience, we list important notation used in this paper in Table I.

As a simple unsupervised evaluation of features, variance utilizes the variance along a feature dimension to reflect the feature's representative power for the original data. The variance of the rth feature denoted as Var_r, which should be maximized, is computed as follows [35]:

$$\mathrm{Var}_r = \frac{1}{N} \sum_{i=1}^{N} (f_{ri} - \boldsymbol{\mu}_r)^2. \tag{1}$$

As another unsupervised method, LS prefers features with larger variances as well as stronger locality preserving ability. It can be regarded as the extension of the Laplacian eigenmaps [37] for feature selection. The key assumption in LS is that the data points from the same class should be close to each other. The LS of the rth feature denoted as LS_r, which should be minimized, is computed as follows [36]:

$$\mathrm{LS}_r = \frac{\sum_{i,j} \left(f_{ri} - f_{rj}\right)^2 S_{ij}}{\sum_i (f_{ri} - \mu_r)^2 D_{ii}}. \tag{2}$$

Here, $\boldsymbol{D}$ is a diagonal matrix with element $D_{ii} = \sum_{j=1}^N S_{ij}$, and S_{ij} represents the neighborhood relationship between samples x_i and x_j defined as follows:

$$S_{ij} = \begin{cases} e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2}}, & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are } k \text{ nearest neighbors} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where σ is a width parameter to be set.

FS is a supervised method using full class labels. It 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. Let $\boldsymbol{\mu}_r^p$ and $\boldsymbol{f}_r^p$ be the mean and the feature vector of class p corresponding to the rth feature. The FS of the rth feature (i.e., FS_r) is computed as follows [35]:

$$\mathrm{FS}_r = \frac{\sum_{p=1}^{P} N_p \left(\boldsymbol{\mu}_r^p - \boldsymbol{\mu}_r\right)^2}{\sum_{p=1}^{P} \sum_{i=1}^{N_p} \left(f_{ri}^p - \boldsymbol{\mu}_r^p\right)^2}. \tag{4}$$

B. Pairwise Constraints-Based Feature Selection

In practice, obtaining class labels is usually expensive, and, thus, the amount of labeled training data is sometimes very limited. Therefore, we are often faced with the so called "small labeled-sample problem" in many real-world applications [3]. For addressing that problem, pairwise constraints (also called side information) arise naturally in many tasks [15], [19], [22], [38]. For example, in the domain of image retrieval [39], considering pairwise constraints is more practical than trying to obtain class labels, because true class labels may be unknown but it is easier for users to specify whether some pairs of data samples belong to the same class (must-link constraint) or different classes (cannot-link constraint), i.e., similar or dissimilar. On the other hand, pairwise constraints can be derived from labeled data but not vice versa. In addition, pairwise constraints can be given in advance or generated from class labels. Given N labeled samples, we can generate approximately N^2 pairwise constraints. For these reasons, pairwise constraints have been widely used in machine learning fields.

Recently, researchers developed various feature selection methods by using pairwise constraints. For example, Zhang *et al.* [25] proposed the so called constraint score (CS) to evaluate the goodness of features by using pairwise constraints. To be specific, CS utilizes $\mathrm{M} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) |\ \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong to the same class}\}$ containing pairwise must-link constraints and $\mathrm{C} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) |\ \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong to different classes}\}$ containing pairwise cannot-link constraints as the supervised information. The CSs of the rth feature (denoted as CS_r) are computed in the following form [25]:

$$\mathrm{CS}_r = \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathrm{M}} \left(f_{ri} - f_{rj}\right)^2 - \lambda \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathrm{C}} \left(f_{ri} - f_{rj}\right)^2 \tag{5}$$

where λ is a parameter to balance the two terms in (5). It has been shown that CS, with only a small amount of pairwise constraints, can achieve comparable performance to fully supervised feature selection methods [25].

In addition, Kalakech *et al.* [27] developed a semi-supervised CS by using both pairwise constraints and local properties of the unlabeled data. To alleviate the bias introduced by the selection of appropriate pairwise constraint sets, Sun and Zhang [26] proposed a bagging CS method to boost the performance of original CS.

C. L_1-Norm-Based Sparse Feature Selection

In the literature, sparse learning attracts much attention in pattern recognition and machine learning domains, among which Lasso is one of the most widely used ones [29]. Generally, Lasso is a penalized least squares method with the L_1-norm penalty on the weight vector (i.e., $\boldsymbol{w} \in \mathbb{R}^d$), and its objective function is defined as follows:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \sum_{i=1}^{N} \left(y_i - \boldsymbol{w}^T \boldsymbol{x}_i\right)^2 \\ \text{s.t.} \quad & \|\boldsymbol{w}\|_1 \le t \end{aligned} \tag{6}$$

where $t \ge 0$ is a tuning parameter to control the amount of shrinkage applied to the estimate $\boldsymbol{w}$. Due to the sparsity nature of L_1-norm, the Lasso method can perform feature selection and regression/classification simultaneously.

Following [29], researchers have developed various L_1-norm-based sparse learning methods (e.g., elastic net [40], group Lasso [31], and fused Lasso [30]). These methods have been widely used in dimension reduction [13], [41], imaging annotation [42], [43], and face recognition [33]. It is worth noting that most existing L_1-norm-based feature selection methods only use class labels as supervised information, while class labels may be limited or expensive to be obtained in practice. To the best of our knowledge, no previous L_1-norm-based sparse learning research has tried to perform feature selection by exploiting pairwise constraints as supervised information.

III. Pairwise CGS Learning

In this section, we first propose a pairwise CGS learning method for feature selection, where pairwise constraints are used as discriminative regularization terms. Then, we present an efficient optimization algorithm for solving the proposed problem. In addition, we further extend our proposed CGS method into a semi-supervised variant called SCGS, and an ensemble variant denoted as ECGS.

A. Proposed CGS Method

Denote N as the size of data points, $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ as data set with class labels $\boldsymbol{Y} = \{y_i\}_{i=1}^N$. Let $\boldsymbol{w} \in \mathbb{R}^d$ denote the weight vector, while M and C represent the must-link constraints set and the cannot-link constraints set, respectively. Intuitively, in the mapping space, data points from the must-link constraint set should be close to each other, and those from the cannot-link constraint set should be as far as possible. Accordingly, the proposed sparse learning model using both pairwise constraints and class labels is defined as follows:

$$\min_{\boldsymbol{w}} \quad \frac{1}{2} \sum_{i=1}^{N} \mathrm{loss}(\boldsymbol{x}_i, y_i, F) + \frac{\lambda_1}{2}(U - \alpha V) + \lambda_2 \|\boldsymbol{w}\|_1 \tag{7}$$

where $U = \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathrm{M}} \left(\boldsymbol{w}^{\mathbf{T}} \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j\right)^2$, $V = \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathrm{C}} \left(\boldsymbol{w}^{\mathbf{T}} \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j\right)^2$, and $\mathrm{loss}(\boldsymbol{x}_i, y_i, F)$ is a general loss function. One can use various loss functions, such as least squares loss and logistic loss. Here, we adopt the least square loss in this paper, i.e., $F(\boldsymbol{x}_i) = y_i - \boldsymbol{w}^{\mathbf{T}} \boldsymbol{x}_i$. The second term is used to force the distance between samples involved in the must-link set to be small (i.e., intraclass compactness), and the distance between samples involved in the cannot-link set to be large (i.e., interclass separability). Actually, the second term is a discriminative regularization term that directly

concentrates on the local discriminative structure of data by using the underlying supervised information conveyed by the must-link and the cannot-link constraints [44], [45]. The parameter α is the regularization parameter that regulates the relative significance of the intraclass compactness and the interclass separability. The last term is the L_1-norm-based regularization term used to generate sparse coefficients for different features.

Let $\boldsymbol{S}^{\mathrm{M}}$ and $\boldsymbol{S}^{\mathrm{C}}$ denote similarity matrices defined on the pairwise cannot-link constraint set and the cannot-link constraint set, respectively. Here, $\boldsymbol{S}^{\mathrm{M}}$ and $\boldsymbol{S}^{\mathrm{C}}$ are defined as follows:

$$S_{ij}^{\mathrm{M}} = \begin{cases} 1, & \text{if } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathrm{M} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$S_{ij}^{\mathrm{C}} = \begin{cases} 1, & \text{if } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathrm{C} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Let $\boldsymbol{D}^{\mathrm{M}}$ and $\boldsymbol{D}^{\mathrm{C}}$ denote two diagonal matrices, where $D_{ii}^{\mathrm{M}} = \sum_{j=1}^{N} S_{ij}^{\mathrm{M}}$ and $D_{ii}^{\mathrm{C}} = \sum_{j=1}^{N} S_{ij}^{\mathrm{C}}$. Then, we can compute the must-link Laplacian matrix $\boldsymbol{L}^{\mathrm{M}} = \boldsymbol{D}^{\mathrm{M}} - \boldsymbol{S}^{\mathrm{M}}$, and the cannot-link Laplacian matrix $\boldsymbol{L}^{\mathrm{C}} = \boldsymbol{D}^{\mathrm{C}} - \boldsymbol{S}^{\mathrm{C}}$ [46]. Accordingly, the regularization terms defined on the must-link constraints and the cannot-link constraints in (8) can be rewritten as

$$\begin{aligned} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathrm{M}} \left(\boldsymbol{w}^{\mathbf{T}} \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j \right)^2 &= \sum_{i,j} \left(\boldsymbol{w}^T \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j \right)^2 S_{ij}^{\mathrm{M}} \\ &= 2 \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{L}^{\mathrm{M}} \boldsymbol{X} \boldsymbol{w} \end{aligned} \tag{10}$$

and

$$\begin{aligned} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathrm{C}} \left(\boldsymbol{w}^{\mathbf{T}} \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j \right)^2 &= \sum_{i,j} \left(\boldsymbol{w}^T \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j \right)^2 S_{ij}^{\mathrm{C}} \\ &= 2 \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{L}^{\mathrm{C}} \boldsymbol{X} \boldsymbol{w}. \end{aligned} \tag{11}$$

By employing the least square loss as well as (10) and (11), we can reformulate the proposed model defined in (7) as follows:

$$\min_{\boldsymbol{w}} \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda_1 \boldsymbol{w}^T \boldsymbol{X}^{\mathbf{T}} \left(\boldsymbol{L}^M - \alpha \boldsymbol{L}^C \right) \boldsymbol{X} \boldsymbol{w} + \lambda_2 \|\boldsymbol{w}\|_1. \tag{12}$$

In addition, to further preserve the structure information of original data, we introduce a manifold regularization term, which is as follows:

$$\sum_{i,j} \left(\boldsymbol{w}^T \boldsymbol{x}_i - \boldsymbol{w}^T \boldsymbol{x}_j \right)^2 S_{ij} = 2 \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{L} \boldsymbol{X} \boldsymbol{w} \tag{13}$$

where $\boldsymbol{L}$ denotes the Laplacian matrix on original data, and $\boldsymbol{D}$ is a diagonal matrix with element $D_{ii} = \sum_{j=1}^{N} S_{ij}$, while the similarity matrix $\boldsymbol{S}$ is defined in (3). By introducing the manifold regularization term into (12), we have the following objective function:

$$\begin{aligned} \min_{\boldsymbol{w}} \ \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda_1 \boldsymbol{w}^T \boldsymbol{X}^T \left(\boldsymbol{L}^M - \alpha \boldsymbol{L}^C \right) \boldsymbol{X} \boldsymbol{w} \\ + \lambda_2 \|\boldsymbol{w}\|_1 + \lambda_3 \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{L} \boldsymbol{X} \boldsymbol{w} \end{aligned} \tag{14}$$

where the first term is the empirical loss on the training data, and the second term is a discriminative regularization term to preserve the local discriminative structure of data conveyed by pairwise constraints. The last two terms are the L_1-norm-based regularizer and the manifold regularization terms, respectively.

Algorithm 1 Learning Algorithm Based on CGS

Input: The training data $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$; The class labels $\boldsymbol{Y} = \{y_i\}_{i=1}^N$; The number of must-link constraints N_M; The number of cannot-link constraints N_C; The learner Φ.
Initialize: The parameters for CGS, i.e., λ_1, λ_2, λ_3 and α.
1: Generate the must-link constraints set M and the cannot-link constraint set C;
2: Calculate the similarity matrices $\boldsymbol{S}^{\mathrm{M}}$, $\boldsymbol{S}^{\mathrm{C}}$ and $\boldsymbol{S}$ using Eqs. (8), (9) and (3), respectively; Compute the corresponding Laplacian matrices $\boldsymbol{L}^{\mathrm{M}}$, $\boldsymbol{L}^{\mathrm{C}}$ and $\boldsymbol{L}$;
3: Compute the optimal solution $\boldsymbol{w}$ of (14);
4: Construct the feature subset G by selecting features with nonzero coefficients of $\boldsymbol{w}$;
5: Call the learner Φ, providing it with the training data set $\{\boldsymbol{x}_i^G, y_i\}_{i=1}^N$ where $\boldsymbol{x}_i^G$ denotes the sample $\boldsymbol{x}_i$ with features in G.
Output: The hypothesis h: $\varphi^T \boldsymbol{x}^G \rightarrow \boldsymbol{Y}$

The model defined in (14) is called pairwise CGS feature selection method in this paper. Without the manifold regularization term (i.e., $\lambda_3 = 0$) in (14), the proposed model is called CGS without manifold regularization term (CGSwm). Let $\lambda_1 = 0$ and $\lambda_3 = 0$, and we find that the proposed CGS method is equivalent to the Lasso formulation defined in (6). Finally, with $\lambda_1 = 0$, the proposed CGS model is equivalent to the Laplacian Lasso (LapLasso) model [47]. The learning algorithm based on our proposed CGS approach is given in Algorithm 1.

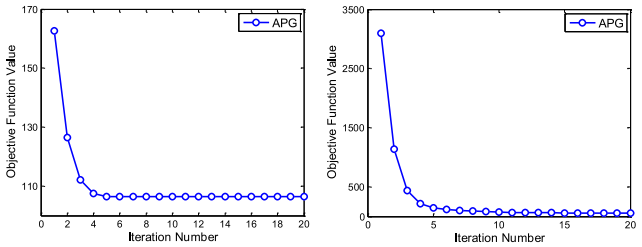
Now, we analyze the computational complexity of Algorithm 1. There are three main steps in Algorithm 1.

1) *Step 1:* Generates the must-link and the cannot-link pairwise constraints, requiring $O(N_{\mathrm{M}} + N_{\mathrm{C}})$ operations.
2) *Step 2:* Calculates three similarity matrices requiring $O(N^2 + N_{\mathrm{M}}^2 + N_{\mathrm{C}}^2)$ operations.
3) *Step 3:* Computes $\boldsymbol{w}$ using Algorithm 2 that needs $O(1/Q^2)$ operations given the iteration number Q.

Hence, the overall computational complexity of Algorithm 1 is $O(N^2 + N_{\mathrm{M}}^2 + N_{\mathrm{C}}^2)$.

B. Optimization Algorithm

Now, we introduce an efficient optimization algorithm for solving the objective function of CGS defined in (14). It is straightforward to verify that the proposed objective function is convex but nonsmooth because of the nonsmooth L_1-norm regularization term. The basic idea to solve the problem is to use a smooth function to approximate the original nonsmooth 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 [48], [49] to solve the proposed objective function in (14). To be specific, we first separate the objective function



(a) (b)

Fig. 1. Convergence of the objective function value on (a) Haberman and (b) ionosphere data sets.

in (14) to the smooth part

$$f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda_1 \boldsymbol{w}^T \boldsymbol{X}^T \left(\boldsymbol{L}^M - \alpha \boldsymbol{L}^C\right)\boldsymbol{X}\boldsymbol{w} + \lambda_3 \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{L}\boldsymbol{X}\boldsymbol{w} \quad (15)$$

and the nonsmooth part

$$h(\boldsymbol{w}) = \lambda_2 \|\boldsymbol{w}\|_1. \quad (16)$$

To approximate the composite function $f(\boldsymbol{w}) + h(\boldsymbol{w})$, we further construct the following function:

$$\Omega_{T,\boldsymbol{w}_j}(\boldsymbol{w}) = f(\boldsymbol{w}_j) + \langle \boldsymbol{w} - \boldsymbol{w}_j, \nabla f(\boldsymbol{w}_j)\rangle + \frac{T}{2}\|\boldsymbol{w} - \boldsymbol{w}_j\|_2^2 + h(\boldsymbol{w}) \quad (17)$$

where $\nabla f(\boldsymbol{w}_j)$ denotes the gradient of $f(\boldsymbol{w})$ at the point $\boldsymbol{w}_j$, and T is the step size that can be determined by line search, e.g., the Armijo–Goldstein rule [50]. Finally, the update step of AGP algorithm is defined as follows:

$$\boldsymbol{w}_{j+1} = \min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{v}_j\|_2^2 + \frac{1}{T}h(\boldsymbol{w}) \quad (18)$$

where the term $\boldsymbol{v}_j = \boldsymbol{w}_j - 1/T\nabla f(\boldsymbol{w}_j)$.

The key of AGP algorithm is deciding how to solve the update step efficiently. Liu and Ye [51] showed that this problem can be decomposed into several separate sub-problems. Thus, we can obtain the analytical solutions of these sub-problems easily. In addition, to achieve faster convergence rate, the accelerated gradient method is based on two sequences $\{\boldsymbol{w}_j\}$ and $\{\boldsymbol{g}_j\}$, where $\boldsymbol{w}_j$ is the sequence of approximate solutions and $\{\boldsymbol{g}_j\}$ is the sequence of search points [48]. The search point $\boldsymbol{g}_j$ is the affine combination of $\boldsymbol{w}_{j-1}$ and $\boldsymbol{w}_j$ as

$$\boldsymbol{g}_j = \boldsymbol{w}_j + \gamma_j(\boldsymbol{w}_j - \boldsymbol{w}_{j-1}) \quad (19)$$

where γ_j is a properly chosen coefficient. The detailed processes are given in Algorithm 2.

For a fixed Q (i.e., the maximum iteration), the APG algorithm for the problem in (14) has $O(1/Q^2)$ asymptotical convergence rate. In Fig. 1, we plot the change of the objective function values versus iteration number on the Haberman and the ionosphere data sets from UCI machine learning repository [52]. From Fig. 1, one can see that the values of the objective function value decreases rapidly within ten iterations, illustrating the fast convergence of Algorithm 2.

Algorithm 2 Optimization Algorithm for CGS

Input: The training data $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$; The class labels $\boldsymbol{Y} = \{y_i\}_{i=1}^N$; The Laplacian matrices $\boldsymbol{L}^{\mathrm{M}}$, $\boldsymbol{L}^{\mathrm{C}}$ and $\boldsymbol{L}$;
Initialize: The maximum iteration number Q; The step size T_0; The parameters for CGS, i.e., λ_1, λ_2 and λ_3.
1: Let $\boldsymbol{w}_0 = \boldsymbol{w}_1 = 0$, $\beta_0 = 1$, and $T = T_0$;
2: **for** $j = 1$ to Q **do**
3: Set $\gamma_j = \frac{(1-\beta_{j-1})\beta_j}{\beta_{j-1}}$, and compute $\boldsymbol{g}_j$ according to (19);
4: Find the smallest $T = T_{j-1}, 2T_{j-1}, \ldots$ such that

$$f(\boldsymbol{w}_{j+1}) + h(\boldsymbol{w}_{j+1}) \le \Omega_{T,\boldsymbol{g}_j}(\boldsymbol{w})$$

where $\boldsymbol{w}_{j+1}$ is computed using (18);
5: Set $T_j = T$ and $\beta_{j+1} = \frac{1+\sqrt{1+4\beta_j^2}}{2}$.
6: **end for**
Output: $\boldsymbol{w}_j$

C. Proposed SCGS Method

The proposed CGS method is supervised, which requires full class labels. However, in many real-world applications, labeled data is usually hard or expensive to be obtained, while unlabeled data and pairwise constraints may be easier to be obtained [20]. In such cases, semi-supervised learning methods are shown helpful to promote the performances of a learning model [10], [21], [22]. In this section, we extend our proposed CGS model to a semi-supervised variant.

Define a diagonal matrix $\boldsymbol{A} \in \mathrm{R}^{N\times N}$ to indicate the labeled data, i.e., $A_{ii} = 0$ if the class label of sample $\boldsymbol{x}_i$ is unknown and $A_{ii} = 1$ otherwise. The objective function of our proposed SCGS model (denoted as SCGS) is as follows:

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{A}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w})\|_2^2 + \lambda_1 \boldsymbol{w}^T \boldsymbol{X}^T \left(\boldsymbol{L}^M - \alpha \boldsymbol{L}^C\right)\boldsymbol{X}\boldsymbol{w} + \lambda_2\|\boldsymbol{w}\|_1 + \lambda_3 \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{L}\boldsymbol{X}\boldsymbol{w} \quad (20)$$

where $\boldsymbol{L}^{\mathrm{M}}$, $\boldsymbol{L}^{\mathrm{C}}$, and $\boldsymbol{L}$ denote Laplacian matrices defined on the must-link constraint set, the cannot-link constraint set, and the whole training data, respectively. In (20), the first term is the empirical loss on labeled data, the second term is a discriminant regularization term focusing on the local structure of data reflected by pairwise constraints, and the last term is an unsupervised estimation on intrinsic geometry distribution of the original data. Similarly to CGS and CGSwm, the SCGS method without the last Laplacian regularization term in (20) is called SCGSwm in this paper.

D. ECGS Method

In [25]–[27], it has been shown that the selection of pairwise constraints has a significant impact on the performances of pairwise constraint-based methods. That is, different sets of pairwise constraints often result in highly unstable results on the same data set. However, deciding how to select the appropriate pairwise constraints set for specific tasks is an open problem [53]. Recently, ensemble-based methods have been proposed for addressing that problem [26], [54], [55]. Inspired by those methods, instead of making efforts on finding a single proper pairwise constraint set, we extend our proposed CGS

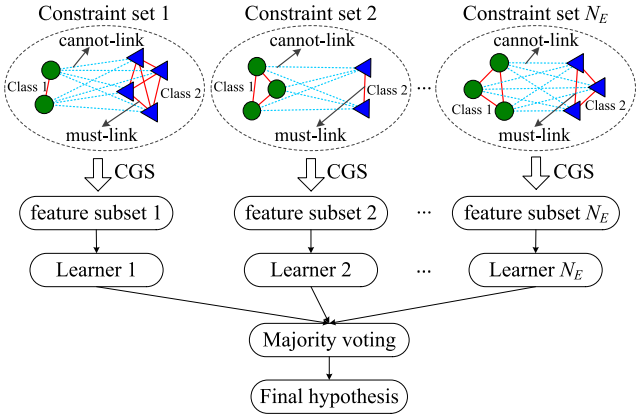


Fig. 2. Flowchart of the proposed ECGS method that uses an ensemble of multiple pairwise constraint sets. Note N_E is the number of pairwise constraint sets.

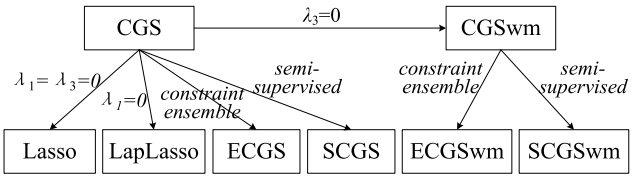


Fig. 3. Relationship of the proposed CGS method and other related methods.

approach by using the ensemble of multiple pairwise constraint sets. The flowchart of the proposed ECGS method is shown in Fig. 2.

As shown in Fig. 2, we first generate multiple pairwise constraint sets. Specifically, we randomly select pairs of samples from the training data, and then generate the must-link or cannot-link constraints are created depending on whether the underlying classes of two samples are the same or different. Then, we execute the proposed CGS algorithm on those individual constraint sets, through which multiple feature subsets can be determined. Afterwards, different individual learners are constructed based on those feature subsets. Finally, we adopt the majority voting strategy for the construction of classifier ensemble, because it is a very simple as well as widely used method for the fusion of multiple classifiers [26]. The ensemble versions of our proposed CGS and CGSwm methods are denoted as ECGS and ECGSwm, respectively. It is worth noting that the ensemble method we used here is to employ multiple learners and combine their prediction [54], which is different from multiview learning that mainly focuses on learning from multiview features [42], [56].

From the above analysis, we can see that our proposed CGS method can be regarded as a unified framework for various feature selection methods. The relationship between CGS method and the other related methods is shown in Fig. 3.

IV. Experiments

In this section, we first introduce the data sets used in our experiments, and then present the experiment design and the experimental results.

A. Data Sets

First, we evaluate our proposed methods (i.e., CGS, CGSwm, ECGS, and ECGSwm), on ten data sets from UCI machine learning repository [52], and a high-dimensional gene expression data set (i.e., colon cancer [57]).

We also perform experiments on attribute classification tasks on the aYahoo [58] and the ImageNet [59] data sets. The aYahoo data set consists of 1151 images of 2688-D features and 64 binary attributes [58], and the ImageNet data set consists of 9600 images with 1550-D features with 25 attributes [59]. These attributes describe properties of objects in the images, such as color, shape, texture, atomy, and parts. In each attribute classification task, images represented by low-level features are used as input data and the attributes for all images are regarded as labels. For the aYahoo data set, we use 15 attributes (with relatively balanced positive and negative samples) to perform attribute classification.

In addition, we evaluate our methods on two neuroimaging-based classification tasks using an Alzheimer's disease (AD) data set, obtained from the AD neuroimaging initiative database (http://adni.loni.usc.edu). This data set have including 202 subjects with the magnetic resonance imaging (MRI), positron emission tomography (PET), and cerebrospinal fluid (CSF) baseline data. There are three categories of subjects, including 51 AD patients, 99 mild cognitive impairment (MCI) patients, and 52 normal control (NC). In the experiments, we perform two classification tasks, including "AD versus NC" classification and "MCI versus NC" classification. Similar to [32], we first perform preprocessing for all MR and PET images. Then, for each of the 93 region of interest (ROI) regions in the labeled MR images, we compute the volume of gray matter tissue in that ROI region as a feature. For each subject, we obtain 93 features from the MR images. For PET images, we use a rigid transformation to align them onto its respective MR images of the same subject, and then compute the average intensity of each ROI in the PET image as a feature. For each subject, we obtain 93 features from the MRI image, another 93 features from the PET image, and three features from the CSF biomarkers. Then, we concatenate these features to form the 189-D representation for a subject. The statistics of data sets used in our experiments are summarized in Table II.

B. Experiment Design

In the experiments, we compare our proposed methods with several well-known feature selection methods, including LS [36], FS [35], CS [25], Lasso [29], and LapLasso [47], [60]. The performance of a specific feature selection method is measured by the classification accuracy based on the selected features on the training data. For LS, FS, and CS methods, we first select the first m features from the ranking list of features generated by corresponding algorithms, where m is the desired number of selected features specified as $m = \{1, 2, \ldots, d\}$ in the experiments. Then, we report the highest classification accuracy as well the number of selected features for LS, FS, and CS. For Lasso,

TABLE II
STATISTICS OF DATA SETS USED IN OUR EXPERIMENTS

Data Set	Dimension #	Class #	Sample #
Diabetes	8	2	768
Wine	13	3	178
Haberman	14	2	306
Wdbc	14	2	569
Vehicle	18	4	846
Hepatitis	19	2	155
Parkinson	22	2	195
Ionosphere	33	2	351
Dermatology	34	6	361
Sonar	60	2	208
Colon Cancer	2000	2	62
AD vs. NC	193	2	103
MCI vs. NC	193	2	151
aYahoo	2688	2	1151
ImageNet	1550	2	9600

TABLE III
TOP 12 ROIS IDENTIFIED BY OUR PROPOSED CGS METHOD IN AD VERSUS NC CLASSIFICATION

Selected ROIs	*p*-value
hippocampal formation left	$p<0.0001$
hippocampal formation right	$p<0.0001$
amygdala right	$p<0.0001$
lateral occipitotemporal gyrus right	$p<0.0001$
precuneus right	$p=0.0002$
superior frontal gyrus left	$p=0.0001$
thalamus left	$p<0.0001$
thalamus right	$p<0.0001$
temporal pole right	$p<0.0001$
temporal pole left	$p<0.0001$
precuneus left	$p=0.0007$
superior parietal lobule left	$p=0.0003$

LapLasso and the proposed methods, the optimal feature subset is determined through corresponding algorithms, and the classification results are reported using such fixed feature subsets. Two classifiers are used to perform classification tasks. The first one is the K-nearest neighborhood (K-NN) classifier with Euclidean distance and $K = 1$, and the second one is a linear support vector machines (SVMs) with the default regularization parameters value (i.e., $C = 1$) [61].

In the supervised classification experiments, we adopt a fivefold cross-validation strategy to compute the mean and the variance of classification accuracy. To be specific, the original data set is partitioned into five subsets (each subset with roughly equal size), and each time samples within one subset are successively selected as the testing data while all the remaining samples in the other four subsets are combined together as the training data to perform feature selection and to learn corresponding classifiers. The process is repeated for ten times independently to avoid any bias introduced by the random partitioning of original data in the cross-validation process. Similarly, in the semi-supervised classification experiments, we adopt a fivefold cross-validation strategy. Specifically, we first partition the original data into roughly equal five folds, and each time we select one of five subsets as the test data, and the others are used for training. For those training data, we randomly select 40% samples as labeled data, and select 40% samples to generate pairwise constraints, while the rest ones are used as unlabeled data. To avoid any bias induced by the random partitioning of original data, the above process is repeated for ten times.

The generation of pairwise constraints is simulated in the following way. First, we randomly select pairs of samples from the training data. Then, the must-link constraints and the cannot-link constraints are created depending on whether the underlying classes of two samples are the same or different. To alleviate the bias introduced by the selection of pairwise constraint sets, following [25], the results achieved by CS and the proposed CGS as well as CGSwm are averaged over 20 runs with different generation of pairwise constraints. For the proposed ECGS and ECGSwm methods, we first generate N_E different must-link and cannot-link constraint sets, and then create multiple sets of selected features by using our proposed CGS and CGSwm approaches, respectively. Based on these feature subsets, we can train multiple learners through corresponding learning algorithms (e.g., SVM and K-NN). Finally, we adopt the majority voting strategy to get a final decision for a specific testing sample. In addition, as shown in [26], one can obtain the best classification results with the ensemble size $N_E = 20$, and using more than 20 components in the ensemble will not further improve the classification results but bring much computational burden. So in our experiments, the ensemble size N_E (i.e., number of constraint sets) is set as 20 empirically.

It has been shown in [25] and [26] that using equal numbers of must-link and cannot-link constraints achieves better performance than using imbalanced constraints in pairwise constraint-based methods. Accordingly, for four constraint-based feature selection approaches (including CS, CGS, CGSwm, ECGS, and ECGSwm), we use equal numbers of must-link and cannot-link constraints in the experiments. To be specific, the numbers of must-link and cannot-link constraints are set as 20% of the sample size for a specific data set. For fair comparison, CS and our proposed methods (i.e., CGS, CGSwm, ECGS, and ECGSwm) share the same pool of pairwise constraints.

Following [25], the parameter λ for CS as well as the parameter α for the proposed CGS method are set to be 0.1 empirically. The regularization parameters λ_1, λ_2, and λ_3 for our proposed CGS method are chosen from $\{10^{-6}, 10^{-5}, \ldots, 10^{0}\}$ through fivefold cross-validation on the training data. Similarly, the parameters λ_1 in Lasso, as well as λ_1 and λ_2 in our proposed CGSwm method, are selected from the same range by cross-validation on the training data. The influence of different parameter values on the performance of our proposed method will be further discussed in Section V.

TABLE IV
SUPERVISED CLASSIFICATION RESULTS USING K-NN CLASSIFIER (%)

Data Set	LS	FS	CS	Lasso	LapLasso	CGSwm	CGS
Diabetes	69.61±0.13(8)	70.13±0.14(4)	69.61±0.06(8)	69.74±0.15(7)	69.74±0.15(7)	71.49±0.12(5)	**72.14±0.64(5)***
Wine	95.83±0.40(12)	96.11±0.10(8)	96.38±0.15(10)	95.83±0.39(13)	96.38±0.12(13)	96.38±0.39(13)	**96.94±0.26(10)**
Haberman	66.32±0.21(14)	69.28±0.18(8)	67.00±0.32(11)	66.32±0.21(14)	66.32±0.21(14)	70.58±0.37(3)	**71.34±0.38(4)***
Wdbc	96.52±0.01(25)	96.70±0.02(24)	95.78±0.03(27)	95.47±0.04(29)	95.82±0.06(27)	97.12±0.04(9)	**97.54±0.02(8)**
Vehicle	71.46±0.22(14)	71.23±0.21(16)	71.22±0.04(18)	70.99±0.32(17)	70.52±0.33(17)	72.09±0.03(17)	**72.15±0.03(17)**
Hepatitis	81.25±0.70(18)	83.13±0.24(17)	84.03±0.36(18)	84.88±0.26(13)	85.50±0.24(14)	85.06±0.37(14)	**86.23±0.36(13)***
Parkinson	**96.25±1.06(7)**	95.50±0.70(2)	95.75±1.76(15)	94.50±0.70(20)	95.75±0.16(21)	**96.25±0.24(19)**	**96.25±0.77(19)**
Ionosphere	88.32±0.26(18)	89.18±0.13(8)	89.59±0.02(8)	87.46±0.05(14)	87.74±0.06(14)	90.18±0.07(13)	**91.32±0.07(13)***
Dermatology	95.47±0.10(2)	95.47±0.56(16)	95.20±0.58(27)	94.38±0.48(34)	95.61±0.39(34)	95.61±0.19(34)	**96.16±0.55(28)**
Sonar	92.85±1.01(51)	92.85±0.33(34)	91.56±1.68(41)	90.47±1.30(46)	91.67±1.68(44)	90.47±1.35(41)	**94.28±0.68(43)***
Colon Cancer	79.00±0.21(42)	82.86±0.43(8)	80.90±0.32(42)	76.92±0.20(50)	81.53±0.43(15)	82.30±0.10(37)	**83.84±0.32(2)**
AD vs. NC	84.21±0.35(48)	84.21±0.32(68)	82.85±0.22(48)	82.86±0.33(40)	81.91±0.45(58)	83.80±0.12(32)	**84.90±0.22(30)***
MCI vs. NC	68.26±0.10(41)	**70.97±0.55**(39)	67.09±0.10(49)	67.09±0.25(56)	69.67±0.13(17)	70.32±0.12(90)	70.69±0.25(17)
aYahoo	69.79±0.57 (170)	69.20±0.63(163)	69.88±0.66(168)	70.32±0.57(160)	71.65±0.39(154)	72.31±0.37(157)	**73.57±0.42(151)**
ImageNet	64.41±0.95(83)	64.68±0.91(79)	64.53±0.93 (85)	65.16±0.88(83)	66.50±0.92(81)	68.16±0.90(78)	**68.68±0.79(73)**

C. Features Selected by the Proposed Method

To investigate whether the proposed CGS method can select the most informative features, we perform the AD versus NC classification on the AD data set to show the features selected by our proposed CGS method. Since the selected features (i.e., ROIs) are different in each cross-validation fold, we choose those features with the highest selection frequency in all folds as the most informative features. For each selected feature, a paired t-test is performed to evaluate its discriminative power for identifying AD patients from NCs, through which the p-value between each specific selected feature and class labels among all training samples can be computed. In Table III, we list the top 12 ROIs selected by our proposed CGS method from all 189 features, as well as corresponding p-values.

From Table III, we can see that the top 12 regions include hippocampal, amygdala, and temporal pole, which are reported to be very relevant to AD disease in [32] and [55]. On the other hand, from Table III, we can see that most of selected features have small p-values indicating their strong discriminative power for distinguishing patients from NCs. It implies that our proposed CGS method can effectively find the most informative features.

D. Results of Supervised Classification

We first validate the efficacy of proposed CGS and CGSwm methods in a supervised problem setting, in comparison to LS, FS, CS, Lasso, and LapLasso. In Table IV, we report the classification results using K-NN classifier, while the results using SVM are shown in Table SI in the online supplementary materials. The meaning of the symbols in the term "$a \pm b(c)$" is as follows: "a" and "b" denote the mean and the variance of classification accuracies among fivefold cross validation, respectively, while "c" represents the number of selected features. Note that the best results are shown in boldface.

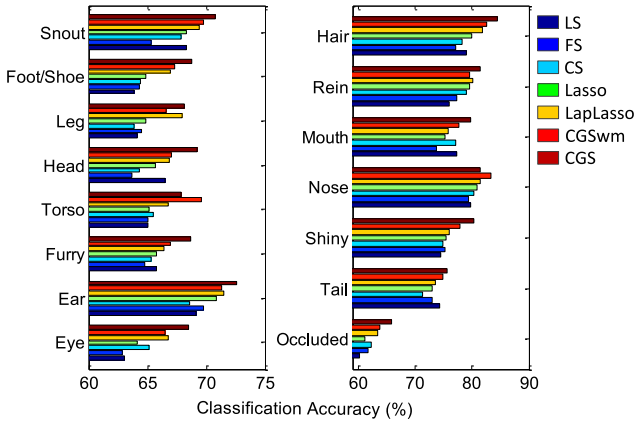


Fig. 4. Classification results of different supervised feature selection methods on the aYahoo data set using K-NN classifier.

We also perform paired t-test between the accuracy achieved by our proposed CGS method and the accuracy achieved by a compared method. The values in Table IV further marked by "*" indicate that our proposed CGS method achieves significant improvement than the other methods by paired t-test (with the confidence interval at 95%). In addition, we report the mean results of attribute classification using K-NN classifier on the aYahoo data set and the ImageNet data set in Table IV.

From Table IV, we can observe four main points. First, our proposed CGS and CGSwm methods usually achieve the overall better performances than the other methods. For example, on the sonar data set, the accuracy achieved by the proposed CGS approach is 94.28%, while the best accuracy of the other methods is only 92.85% (achieved by LS and FS). Second, in most cases, the numbers of features selected by CGS and CGSwm are less than that of other five methods. In particular, on colon cancer data set, CGS achieves the highest accuracy

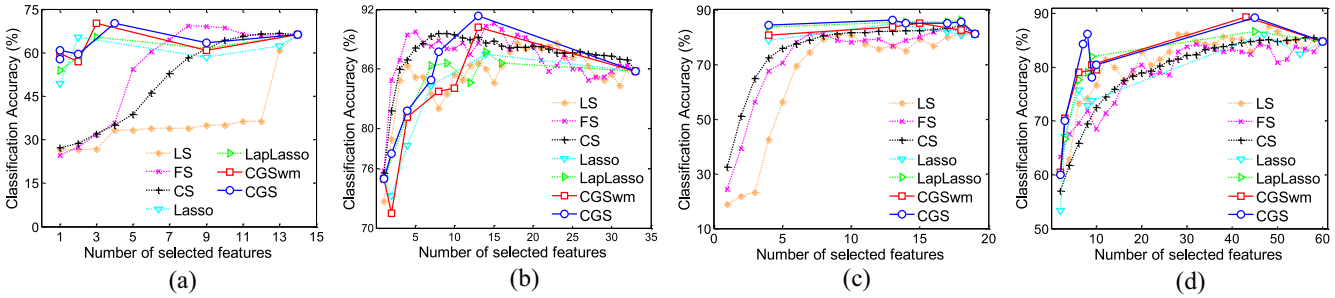


Fig. 5. Classification accuracies versus different numbers of selected features achieved by seven feature selection methods on (a) Haberman, (b) ionosphere, (c) hepatitis, and (d) sonar data sets.

using only two features, while the best result is achieved by FS with eight features. Third, for pairwise constraint-based methods, our proposed CGS method consistently performs better than CS and CGSwm usually outperforms CS. Finally, CGS that considers both the pairwise constraints and the manifold information is usually superior to CGSwm using only pairwise constraints. Similarly, LapLasso that considers the manifold information achieves better results than Lasso in most cases. It indicates that the structure information is important in guiding the process of feature selection.

In Fig. 4, we show the results of 15 attribute classification on aYahoo data set with K-NN classifier, while results with SVM classifier are given in Fig. S1 in the online supplementary materials. At the same time, we also report the attribute classification results on the ImageNet data set using both K-NN and SVM classifiers in Figs. S2 and S3, respectively, in the online supplementary materials. From Figs. 4 and S1–S3, one can observe that our proposed methods (i.e., CGS and CGSwm) usually outperform the other compared methods in multiple attribute classification tasks. Especially, as shown in Fig. 4, in the "Eye" and "Head" classification tasks on the aYahoo data set, CGS achieves much higher accuracy than the compared methods.

Furthermore, we investigate the influence of the number of selected features on the classification results. For LS, FS, and CS, the number of selected features vary from 1 to d according to feature ranking lists obtained by a specific algorithm. For L_1-norm-based methods (i.e., Lasso, LapLasso, CGSwm, and CGS), the parameter for the L_1-norm regularization term that controls the sparsity of feature coefficients (corresponding to optimal number of selected features) is chosen from $\{10^{-6}, 10^{-5}, \ldots, 10^{0}\}$ through cross-validation. The other parameters for CGS (i.e., λ_1 and λ_3), the parameter λ_1 for CGSwm, and the parameter λ_2 for LapLasso are chosen from the same range through cross-validation on the training data. In Fig. 5, we plot the curves of classification accuracy versus different numbers of selected features on four UCI data sets using K-NN classifier.

From Fig. 5, we can see that our proposed CGS and CGSwm methods achieve the overall best performances using smaller number of selected features, in comparison to LS, FS, and CS. Particularly on the Haberman and the ionosphere data sets, CGS, and CGSwm need less than half of features to achieve the best performances. Second, using similar number of selected features, CGS and CGSwm achieve higher classification accuracies than Lasso and LapLasso in most cases. This conclusion is consistent with the results in Table IV.

E. Results of Semi-Supervised Classification

In this section, we evaluate our proposed SCGS and SCGSwm methods in a semi-supervised problem setting on UCI data sets, in comparison to FS, Lasso, and LapLasso. Note that LapLasso, SCGSwm, and SCGS methods use both labeled data and unlabeled data, while FS and Lasso use only the labeled data. We report the classification results achieved by different feature selection methods using K-NN in Table V, while the results using SVM classifier are given in Table SII in the online supplementary materials.

From Table V, we can find that, in most cases, SCGS performs better than FS, Lasso, and LapLasso, especially on the hepatitis and the sonar data sets. These results imply that the features selected by SCGS have better discriminative ability compared with those selected by other methods. Recall the results in Table IV achieved by the proposed supervised CGS method, and we can find that although a small number of labeled data are used in the proposed SCGS method, the classification performances achieved by SCGS only decrease slightly compared with the results in a full supervised manner. For example, from Tables IV and V, we can clearly see that the performances of SCGS are similar to those of CGS on diabetes, wine, and Wisconsin diagnostic breast cancer. These results demonstrate that our proposed CGS method can effectively identify discriminative features in semi-supervised problem settings. The underlying reason may be that the disadvantage of lacking enough labeled data can be compensated by the information conveyed by pairwise constraints in the proposed SCGS method.

F. Results of ECGS

We then compare our proposed CGS and CGSwm methods with their ensemble counterparts, i.e., ECGS and ECGSwm, respectively. The classification results using the K-NN classifier are given in Fig. 6, and the results using the SVM classifier are shown in Fig. S4 in the online supplementary materials. From Fig. 6, we can see that, in most cases, the proposed ECGS method using the ensemble of pairwise constraint sets outperforms its traditional counterpart

TABLE V
SEMI-SUPERVISED CLASSIFICATION RESULTS
USING K-NN CLASSIFIER (%)

Data Set	FS	Lasso	LapLasso	SCGSwm	SCGS
Diabetes	67.51±0.18(4)	67.62±0.16(8)	68.75±0.28(3)	69.20±0.11(8)	**69.92±0.10(3)**
Wine	92.22±1.17(6)	93.33±0.12(11)	94.67±0.12(8)	93.56±0.39(9)	**95.00±0.05(8)***
Haberman	60.56±0.17(13)	60.44±0.20(14)	60.44±0.08(14)	60.77±0.10(12)	**61.09±0.13(12)**
Wdbc	95.13±0.03(22)	94.61±0.03(30)	95.65±0.01(21)	94.61±0.03(30)	**95.82±0.02(18)**
Vehicle	63.15±0.08(18)	63.15±0.08(18)	**64.21±0.09(13)**	63.15±0.08(18)	63.15±0.08(18)
Hepatitis	80.00±0.18(18)	82.50±0.32(19)	83.13±0.62(16)	85.15±0.21(16)	**86.88±0.21(15)***
Parkinson	87.00±0.42(7)	87.00±0.11(20)	88.00±0.08(21)	89.50±0.04(16)	**90.50±0.04(18)***
Ionosphere	87.11±0.13(8)	83.48±0.54(29)	85.47±0.45(23)	83.63±0.14(28)	**87.75±0.67(5)***
Dermatology	**93.69±0.12(23)**	89.32±0.11(32)	91.05±0.19(31)	90.41±0.11(32)	92.05±0.09(30)
Sonar	79.04±0.61(40)	77.14±0.57(50)	79.52±0.18(47)	78.57±0.58(45)	**82.38±0.47(41)***

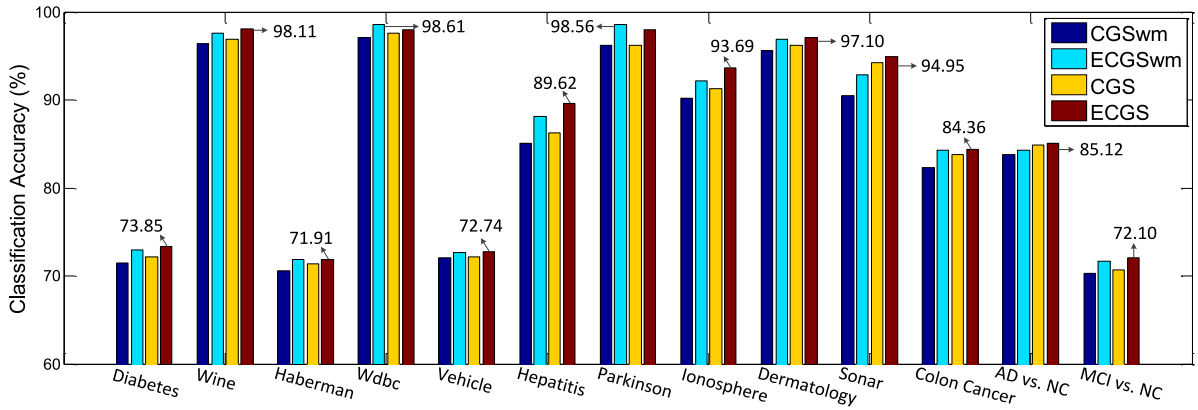


Fig. 6. Classification results of our proposed methods with and without using the ensemble of pairwise constraint sets with K-NN classifier, while the results using SVM classifier is reported in the supplementary materials.

(i.e., CGS) that utilizes only one single pairwise constraint set. Similarly, results achieved by the proposed ECGSwm methods are usually better than those of CGSwm. These observations suggest that using the ensemble of pairwise constraint sets can further promote the performances of our proposed pairwise CGS feature selection methods.

V. DISCUSSION

In this section, we first discuss the influences of different parameters and constraint numbers on the performance of our proposed CGS method. Then, we directly employ our proposed CGS method with logistic loss as a sparse classifier to perform classification tasks compared with other sparse learners.

A. Influence of Parameters

In our proposed CGS method, there are three parameters (i.e., λ_1, λ_2, and λ_3) to be tuned. In this section, we evaluate how CGS performs with different values of these parameters on two UCI data sets. In Fig. 7, we show the classification accuracies as a function of two of these three parameters on the Haberman data set, while the results on the ionosphere data set are shown in Fig. S5 in the online supplementary materials.

From Fig. 7, we can clearly see that the influence of parameters on our proposed CGS method on the Haberman and the ionosphere data sets. More specifically, on the Haberman data set, Fig. 7(a) shows that the performance of the proposed CGS method slightly fluctuate in a small rage with the increase of parameter λ_1 and λ_2, while Fig. 7(b) and (c) indicates that CGS is stable with respect to the other two pairs of parameters, i.e., λ_2 and λ_3, and λ_1 and λ_3. On the ionosphere data set, Fig. S5 reveals that that the performance of the proposed CGS method has some fluctuations by using different values for three parameters. These results imply that the selection of parameters is critical for our proposed methods, which is a common problem in sparse feature selection domain. As suggested in [3], cross-validation provides a good way to find the optimal parameters (as we do in this paper).

B. Influence of Constraint Number

Then, we discuss the influence of different constraint numbers on the performance of our proposed CGS method. In the experiments, we gradually increase the number of pairwise constraints, and record corresponding classification results achieved by different methods (including CS, CGS, and CGSwm). The classification results versus

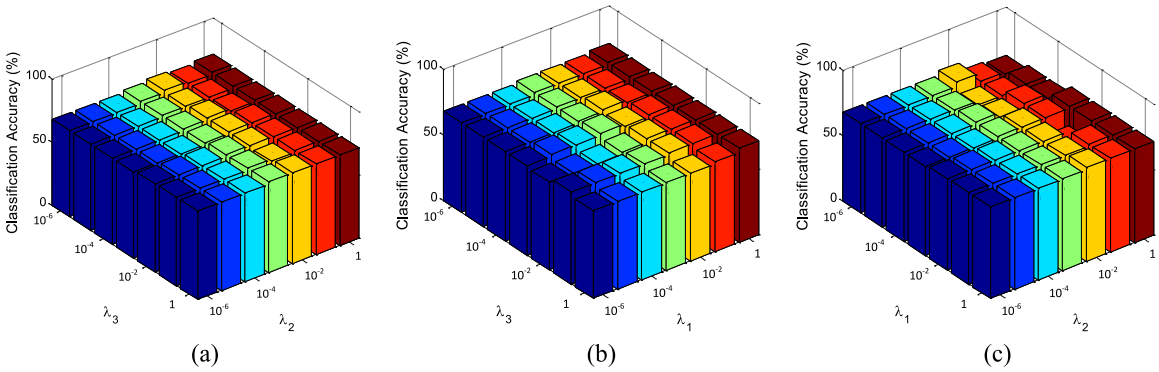


Fig. 7. Classification accuracies versus parameters λ_1, λ_2, and λ_3 on the Haberman data set. (a) Fixed λ_1. (b) Fixed λ_2. (c) Fixed λ_3. Note that in (a)–(c), when two parameters vary, another is fixed as 0.001.

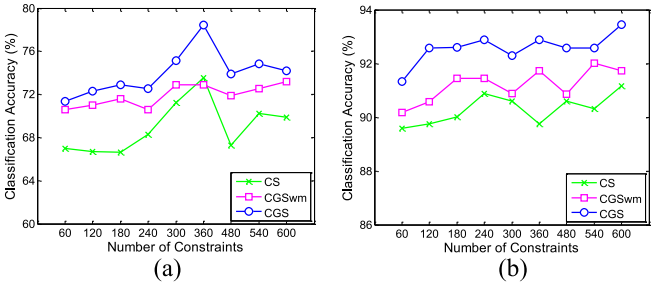


Fig. 8. Classification results versus the number of pairwise constraints achieved by different methods on (a) Haberman and (b) ionosphere data sets.

TABLE VI
RESULTS USING DIFFERENT SPARSE CLASSIFIERS (%)

Data Set	L_1_log	LapL_1_log	CGSwm_log	CGS_log
Diabetes	76.53±0.09(7)	77.79±0.04(8)	77.53±0.08(8)	**78.44±0.07(8)**
Haberman	75.17±0.08(9)	75.53±0.02(14)	76.14±0.06(14)	**77.05±0.05(13)***
Wdbc	97.91±0.01(24)	98.09±0.02(29)	**98.61±0.01(29)**	98.43±0.01(30)
Hepatitis	86.00±0.13(18)	86.25±0.13(19)	88.88±0.13(19)	**90.00±0.12(19)***
Parkinson	88.00±0.17(16)	89.50±0.23(22)	91.00±0.27(22)	**92.10±0.23(22)***
Ionosphere	90.02±0.05(33)	91.16±0.03(27)	92.02±0.09(33)	**92.87±0.06(33)***
Sonar	81.43±0.12(34)	82.76±0.21(57)	81.90±0.30(54)	**84.76±0.16(52)***

number of pairwise constraints using K-NN classifier on Haberman and ionosphere data sets are given in Fig. 8, and the results on the other four UCI data sets (i.e., hepatitis, sonar, wine, and Parkinson) are shown in Fig. S6 in the online supplementary materials.

As shown in Fig. 8(a), on the Haberman data set, the overall performances of the proposed CGS and CGSwm methods gradually become better with the increase of constraint number in a large scale. When the number of constraint is larger than 360, the performance of CGS has some fluctuations. On the ionosphere data set, from Fig. 8(b), we can see that on the whole the performances of CGS gradually increase when the constraint number increases. These results imply that adding more side information (i.e., pairwise constraints) help improve the classification results. On the other hand, despite of different pairwise constraint numbers, CGS consistently outperforms CS, while in most cases CGSwm performs better than CS.

C. Comparison With Sparse Classifiers

As shown in (14), the proposed CGS model can be used as a sparse classifier by using a logistic loss function. In this section, we employ CGS with logistic loss as a sparse classifier, and compare it with L_1 logistic regression model (L_1_log) and Laplacian regularized L_1 logistic regression model (LapL_1_log). We denote our proposed method with logistic loss function as CGS_log and CGSwm_log, respectively. In Table VI, we report the classification results on seven two-class UCI data sets achieved by four sparse learners.

From Table VI, we can see that our proposed CGS_log method significantly outperforms L_1_log and LapL_1_log, while the proposed CGSwm_log performs better than L_1_log and LapL_1_log on those seven UCI data sets. On the other hand, one can find another interesting observation from Tables IV, VI, and SI. That is, our proposed CGS_log method generally outperforms CGS with both K-NN and SVM classifiers, and the proposed CGSwm_log usually performs better than CGSwm with both K-NN and SVM classifiers. The underlying reason could be that the selected features in CGS_log and CGSwm_log are very suitable for their corresponding classifiers, because both selected features and corresponding classifiers are learned from a same objective function. It indicates that our proposed model can not only be used for performing feature selection, but also be effective to perform classification tasks as sparse learners.

VI. CONCLUSION

In this paper, we propose a pairwise CGS learning method for feature selection, where pairwise constraints are used as discriminative regularization terms that concentrate on the local discriminative structure of data by using the underlying supervised information conveyed by the must-link and the cannot-link constraints. Furthermore, we extend our proposed CGS method into a SCGS method and an ECGS method. Experimental results on a number of data sets validate the efficacy of our proposed methods.

In this paper, the number of selected features mainly depends on the parameter of L_1-norm regularization, which is

very limited. To select an optimal feature subset with a fixed size based on the needs of specific tasks seems to be more appealing, which is one of our future works. In addition, we plan to adapt our proposed pairwise CGS learning methods to multimodality learning problems.

ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous reviewers for their constructive comments and contributions for the improvement of this paper.

REFERENCES

[1] N. Kwak and C. H. Choi, "Input feature selection by mutual information based on Parzen window," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1667–1671, Dec. 2002.

[2] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[3] T. Hastie, R. Tibshirani, and J. J. H. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2001.

[4] A. R. Webb, *Statistical Pattern Recognition*. London, U.K.: Arnold, 1999.

[5] S. Li and D. Wei, "Extremely high-dimensional feature selection via feature generating samplings," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 737–747, Jun. 2014.

[6] H. Richang *et al.*, "Image annotation by multiple-instance learning with discriminative feature mapping and selection," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 669–680, May 2014.

[7] D. Ren, C. Fei, P. Taoxin, N. Snooke, and S. Qiang, "Feature selection inspired classifier ensemble reduction," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1259–1268, Aug. 2014.

[8] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[9] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, "Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1553–1566, Dec. 2004.

[10] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning,* vol. 2. Cambridge, MA, USA: MIT Press, 2006.

[11] H. Chenping, N. Feiping, L. Xuelong, Y. Dongyun, and W. Yi, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[12] J. Yu, R. Rui, and B. Chen, "Exploiting click constraints and multi-view features for image re-ranking," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 159–168, Jan. 2014.

[13] W. Liu, H. Zhang, D. Tao, Y. Wang, and K. Lu, "Large-scale paralleled sparse principal component analysis," *Multimedia Tools Appl.*, pp. 1–13, 2013.

[14] J. Yu, D. Liu, D. Tao, and H. S. Seah, "Complex object correspondence construction in two-dimensional animation," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3257–3269, Nov. 2011.

[15] M. S. Baghshah and S. B. Shouraki, "Semi-supervised metric learning using pairwise constraints," in *Proc. Int. Joint Conf. Artif. Intell.*, Pasadena, CA, USA, 2009, pp. 1217–1222.

[16] W. Liu, S. Ma, D. Tao, J. Liu, and P. Liu, "Semi-supervised sparse metric learning using alternating linearization optimization," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Washington, DC, USA, 2010, pp. 1139–1148.

[17] F. Wang, "Semisupervised metric learning by maximizing constraint margin," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 4, pp. 931–939, Aug. 2011.

[18] A. Mignon and F. Jurie, "PCCA: A new approach for distance learning from sparse pairwise constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 2666–2672.

[19] I. Davidson and S. Basu, "A survey of clustering with instance level constraints," *ACM Trans. Knowl. Disc. Data*, vol. w, no. x, pp. 1–41, 2007.

[20] H. Zeng and Y.-M. Cheung, "Semi-supervised maximum margin clustering with pairwise constraints," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 926–939, May 2012.

[21] W. Zhao, Q. He, H. Ma, and Z. Shi, "Effective semi-supervised document clustering via active learning with instance-level constraints," *Knowl. Inf. Syst.*, vol. 30, no. 3, pp. 569–587, 2012.

[22] S. Ding, H. Jia, L. Zhang, and F. Jin, "Research of semi-supervised spectral clustering algorithm based on pairwise constraints," *Neural Comput. Appl.,* vol. 24, no. 1, pp. 211–219, 2014.

[23] F. Nie, D. Xu, I. W.-H. Tsang, and C. Zhang, "Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1921–1932, Jul. 2010.

[24] C. Xu, D. Tao, C. Xu, and Y. Rui, "Large-margin weakly supervised dimensionality reduction," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 865–873.

[25] D. Zhang, S. Chen, and Z.-H. Zhou, "Constraint score: A new filter method for feature selection with pairwise constraints," *Pattern Recognit.*, vol. 41, no. 5, pp. 1440–1451, 2008.

[26] D. Sun and D. Zhang, "Bagging constraint score for feature selection with pairwise constraints," *Pattern Recognit.*, vol. 43, no. 6, pp. 2106–2118, 2010.

[27] M. Kalakech, P. Biela, L. Macaire, and D. Hamad, "Constraint scores for semi-supervised feature selection: A comparative study," *Pattern Recognit. Lett.*, vol. 32, no. 5, pp. 656–665, 2011.

[28] M. Liu and D. Zhang, "Sparsity score: A novel graph-preserving feature selection method," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 28, no. 4, 2014, Art. ID 1450009.

[29] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B, Met.*, vol. 58, no. 1, pp. 267–288, 1996.

[30] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. Roy. Statist. Soc. B, Stat. Met.*, vol. 67, no. 1, pp. 91–108, 2005.

[31] L. Meier, S. Van De Geer, and P. Bühlmann, "The group lasso for logistic regression," *J. Roy. Statist. Soc. B, Stat. Met.*, vol. 70, no. 1, pp. 53–71, 2008.

[32] D. Zhang, Y. Wang, L. Zhou, H. Yuan, and D. Shen, "Multimodal classification of Alzheimer's disease and mild cognitive impairment," *Neuroimage*, vol. 55, no. 3, pp. 856–867, 2011.

[33] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[34] Y. Xie *et al.*, "Discriminative object tracking via sparse representation and online dictionary learning," *IEEE Trans. Cybern.*, vol. 44, no. 4, pp. 539–553, Apr. 2014.

[35] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[36] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, Whistler, BC, Canada, 2005, pp. 507–514.

[37] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[38] P. K. Mallapragada, R. Jin, and A. K. Jain, "Online visual vocabulary pruning using pairwise constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 3073–3080.

[39] H. Li, X. Wang, J. Tang, and C. Zhao, "Combining global and local matching of multiple features for precise item image retrieval," *Multimedia Syst.*, vol. 19, no. 1, pp. 37–49, 2013.

[40] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc. B, Stat. Met.*, vol. 67, no. 2, pp. 301–320, 2005.

[41] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1672–1680, Sep. 2008.

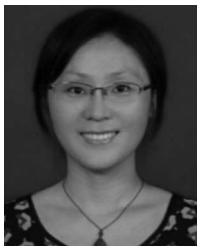
[42] W. Liu and D. Tao, "Multiview Hessian regularization for image annotation," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2676–2687, Jul. 2013.

[43] W. Liu, D. Tao, J. Cheng, and Y. Tang, "Multiview hessian discriminative sparse coding for image annotation," *Comput. Vis. Image Understand.*, vol. 118, pp. 50–60, Jan. 2014.

[44] H. Xue and S. Chen, "Discriminality-driven regularization framework for indefinite kernel machine," *Neurocomputing*, vol. 133, pp. 209–221, Jun. 2014.

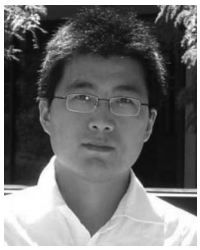
[45] H. Xue, S. Chen, and Q. Yang, "Discriminatively regularized least-squares classification," *Pattern Recognit.*, vol. 42, no. 1, pp. 93–104, 2009.

[46] F. R. Chung, *Spectral Graph Theory*. Providence, RI, USA: Amer. Math. Soc., 1997.
[47] C. Li and H. Li, "Network-constrained regularization and variable selection for analysis of genomic data," *Bioinformatics*, vol. 24, no. 9, pp. 1175–1182, 2008.
[48] X. Chen, W. Pan, J. T. Kwok, and J. G. Carbonell, "Accelerated gradient method for multitask sparse learning problem," in *Proc. IEEE Int. Conf. Data Min.*, Miami, FL, USA, 2009, pp. 746–751.
[49] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[50] J. J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia, PA, USA: SIAM, 1983.
[51] J. Liu and J. Ye, "Efficient L1/Lq norm regularization," Dept. Comput. Sci. Eng., Arizona State Univ., Tucson, AZ, USA, Tech. Rep., 2009.
[52] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2007.
[53] K. L. Wagstaff, *Value, Cost, and Sharing: Open Issues in Constrained Clustering*. New York, NY, USA: Springer, 2007.
[54] T. G. Dietterichl, "Ensemble learning," in *The Handbook of Brain Theory and Neural Networks.* Cambridge, MA, USA: MIT Press, 2002, pp. 405–408.
[55] M. Liu, D. Zhang, and D. Shen, *View-Centralized Multi-Atlas Classification for Alzheimer's Disease Diagnosis*. New York, NY, USA: Human Brain Map., 2015.
[56] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *Neural Comput. Appl.,* vol. 23, no. 7–8, pp. 2031–2038, 2013.
[57] U. Alon *et al.,* "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Nat. Acad. Sci. USA*, vol. 96, pp. 6745–6750, Jun. 1999.
[58] A. Farhadi, I. Endres, and D. Hoiem, "Attribute-centric recognition for cross-category generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 2352–2359.
[59] O. Russakovsky and L. Fei-Fei, "Attribute learning in large-scale datasets," in *Proc. Workshop Parts Attributes Eur. Conf. Comput. Vis.*, Crete, Greece, 2010, pp. 10–11.
[60] H. Fei, B. Quanz, and J. Huan, "Regularization and feature selection for networked features," in *Proc. Int. Conf. Inf. Knowl. Manage.*, Toronto, ON, Canada, 2010, pp. 1893–1896.
[61] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.



Mingxia Liu received the B.S. and M.S. degrees from Shandong Normal University, Shandong, China, in 2003 and 2006, respectively, and the Ph.D. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015.

Her current research interests include neuroimaging analysis, machine learning, pattern recognition, and data mining.



Daoqiang Zhang received the B.S. and Ph.D. degrees in computer science from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1999 and 2004, respectively.

In 2004, he joined the Department of Computer Science and Engineering, NUAA, as a Lecturer, where he is currently a Professor. His current research interests include machine learning, pattern recognition, data mining, and medical image analysis. He has published over 100 scientific articles in refereed international journals such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Neuroimage, Human Brain Mapping*, and conference proceedings such as *International Joint Conferences on Artificial Intelligence*, *IEEE International Conference on Data Mining*, and *International Conference on Medical Image Computing and Computer Assisted Interventions*.

Dr. Zhang is a member of the Machine Learning Society of the Chinese Association of Artificial Intelligence and the Artificial Intelligence and Pattern Recognition Society of the China Computer Federation.