Learning a Nonnegative Sparse Graph for Linear Regression

Xiaozhao Fang, *Student Member, IEEE*, Yong Xu, *Senior Member, IEEE*, Xuelong Li, *Fellow, IEEE*, Zhihui Lai, and Wai Keung Wong

***Abstract*— Previous graph-based semisupervised learning (G-SSL) methods have the following drawbacks: 1) they usually predefine the graph structure and then use it to perform label prediction, which cannot guarantee an overall optimum and 2) they only focus on the label prediction or the graph structure construction but are not competent in handling new samples. To this end, a novel nonnegative sparse graph (NNSG) learning method was first proposed. Then, both the label prediction and projection learning were integrated into linear regression. Finally, the linear regression and graph structure learning were unified within the same framework to overcome these two drawbacks. Therefore, a novel method, named learning a NNSG for linear regression was presented, in which the linear regression and graph learning were simultaneously performed to guarantee an overall optimum. In the learning process, the label information can be accurately propagated via the graph structure so that the linear regression can learn a discriminative projection to better fit sample labels and accurately classify new samples. An effective algorithm was designed to solve the corresponding optimization problem with fast convergence. Furthermore, NNSG provides a unified perceptiveness for a number of graph-based learning methods and linear regression methods. The experimental results showed that NNSG can obtain very high classification accuracy and greatly outperforms conventional G-SSL methods, especially some conventional graph construction methods.**

***Index Terms*— Graph learning, linear regression, label propagation, semi-supervised classification.**

Manuscript received June 18, 2014; revised November 30, 2014; accepted April 16, 2015. Date of publication April 22, 2015; date of current version May 22, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61370163, Grant 61233011, and Grant 61332011, in part by the Shenzhen Municipal Science and Technology Innovation Council under Grant JCYJ20130329151843309, Grant JCYJ20130329151843309, Grant JCYJ20140417172417174, and Grant CXZZ20140904154910774, in part by the China Post-Doctoral Science Foundation under Grant 2014M560264, and in part by the Shaanxi Key Innovation Team of Science and Technology under Grant 2012KCT-04. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Rebecca Willett. *(Corresponding author: Yong Xu.)*

X. Fang is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: xzhfang168@126.com).

Y. Xu is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China, and also with the Key Laboratory of Network Oriented Intelligent Computation, Shenzhen 518055, China (e-mail: laterfall2@yahoo.com.cn).

X. Li is with the State Key Laboratory of Transient Optics and Photonics, Center for Optical Imagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn).

Z. Lai is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China, and also with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518055, China (e-mail: lai_zhi_hui@163.com).

W. K. Wong is with the Institute of Textiles and Clothing, The Hong Kong Polytechnic University, Hong Kong, and also with The Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen 518057, China (e-mail: calvin.wong@polyu.edu.hk).

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/TIP.2015.2425545

I. Introduction

In THE past decades, a lot of dimension reduction techniques have been proposed [1]–[5]. Principal component analysis (PCA) is an unsupervised method which can project high dimensional data into a lower dimensional space by seeking the direction of maximum variance for optimal data reconstruction [6], [7]. Linear discriminant analysis (LDA) is a supervised dimension reduction method which pursues a linear projection that simultaneously maximizes the distances among the means of the classes and minimizes the distances among the data points sharing the same label using the Fisher's criterion [8]–[11]. Neighborhood component analysis (NCA) was developed to directly optimize the expected leave-one-out classification accuracy on the training set and the resulting lower subspace seems to be more discriminant than those obtained by PCA and LDA [12], [13]. Some nonlinear dimension reduction techniques such as locally linear embedding (LLE) [14] and Laplacian eigenmap [LE] [15] were recently proposed to discover the intrinsic manifold structure of the data. A drawback of LE is that it cannot deal with new data points that are not included in the training set. He et al. proposed the locality preserved projection (LPP) to solve this problem, in which the learned linear projection is used for handling the new data points [16]. In addition, local learning projection (LLP) was proposed to address this problem [17]. He et al. also proposed the neighborhood preserving embedding (NPE) method for preserving the local neighborhood structure on the data manifold [18]. Zhang et al. [19] demonstrated that many dimension reduction methods can be reformulated into a unified patch alignment framework. Yan et al. further reformulated some dimension reduction methods (e.g., PCA, LDA, ISOMAP, LLE, LE) into a unified graph embedding framework, in which the desired geometric structure of data are encoded as graph relationships [20].

In general, the labeled training samples are always insufficient because the labeling data requires expensive

labor [21]–[23]. The performance of some supervised algorithms degenerates in the case of insufficient labeled training samples. Semi-supervised learning is a method that can improve the performance of algorithms in this case by using the unlabeled samples obtained in much easier ways. Some semi-supervised learning methods such as transductive support vector machine (TSVM) [24], co-training [25], and semi-supervised discriminant analysis (SDA) [26] were developed and demonstrated promising results. Recently, G-SSL methods were proposed and aroused great interest among researchers [27]–[29]. These methods model the geometric relationship between all data points by the form of an affinity graph and then propagate the label information from the labeled data points to the unlabeled data points via the affinity graph [22]. Thus the quality of the constructed graph is of great importance to label propagation. The graph-based regularization term is commonly used to improve the performance of semi-supervised learning methods [30], [31]. By adding a graph-based regularization term, Laplacian support vector machines (LapSVM) and Laplacian regularized least squares (LapRLS) [27], [32] achieved good classification results. A flexible manifold embedding (FME) framework was proposed for semi-supervised and unsupervised dimension reduction, in which the low dimension representation of data is obtained by linear regression [33]. Moreover, FME used the learned projection to map new data points. The common assumption of G-SSL methods is label consistency, i.e., nearby points are prone to have the same labels. In most G-SSL methods, graph construction is based on various techniques, among which the popular ones include the k nearest neighbor graph, local linear reconstruction graph proposed in LLE [14], and ℓ_1 graph [34]. The performance of semi-supervised classification relies heavily on the graph construction process. According to Wright et al. [35], an informative graph should have three characteristics: suitable sparsity, high discriminant power, and adaptive neighborhood. Conventional methods (such as the k nearest neighbor graph and LLE graph) use the specified nearest neighbor parameter and the setting may be variational for different data sets and thus, the adaptive neighborhood is lost in these methods. Although the ℓ_1 graph is sparse, datum-adaptive and robust to data noise, it is only to find the sparse representation for data reconstruction. However, the best data reconstruction does not represent the best discriminate power [36]. In this way, the ℓ_1 graph may lose the high discriminant power. What's more, in most of these G-SSL methods, their graph structures are pre-defined. As a consequence, the graph construction and semi-supervised learning algorithm are often independent steps, and hence the overall optimum of the algorithm cannot be guaranteed. For example, in FME, the graph structure was constructed in advance by the k nearest neighbor graph technique which may result in that the label prediction and projection learning may be not optimal. Since label propagation relies heavily on the graph construction process, a simultaneous approach which integrates both label propagation and graph learning is demanded. Wang et al. proposed a face annotation method by using weak label regularized local coordinate coding (LCC) [37], in which the sparse features and the graph-based weak label regularization were simultaneously employed to enhance the weak labels of similar facial images [38]. A similar strategy was also used for joint learning of labels and distance metric, in which the label prediction and distance metric were optimized in a unified scheme [39]. In LCC, each data point can be locally approximated by a linear combination of its nearby points. In other words, the coding scheme in LCC can ensure that nearby points are prone to have similar coding coefficients, which is very similar to the assumption of the label consistency. This observation motivated us to construct the graph structure for label propagation by using LCC. Prior works (e.g., LDA, LLP, NPE and FME) used the learned projection to deal with new samples. We thus considered using the projection to classify new samples. In order to guarantee an overall optimum, the projection learning, label propagation and graph structure learning should be completed in one step.

Based on the above observations, we proposed a novel method, termed learning a non-negative sparse graph (NNSG) for linear regression which is based on two previous works LCC [37] and FME [33]. The whole learning process is driven by the philosophy that the linear regression and graph structure learning should be simultaneously performed to find an overall optimum. With this simultaneous learning scheme, the label information, in the learning process, can be accurately propagated via the graph structure so that the linear regression can learn a discriminative projection to better fit sample labels and accurately classify new samples. An iterative procedure is presented for solving the corresponding optimization problem and it converges fast. The most important contributions of the proposed method are as follows.

(1) Unlike previous G-SSL methods, in which the graph structure and the designed algorithm are often independent steps, NNSG integrates these two tasks into one single optimization step to guarantee an overall optimum.

(2) NNSG not only solves two essential tasks in G-SSL, i.e., label propagation and graph learning, but also can effectively deal with new samples by the learned projection. In other words, NNSG unifies the graph learning, label propagation and projection learning within the same framework.

(3) An efficient optimized strategy is proposed to solve the optimization problem. Some theoretical and experimental analysis are presented to show the convergence behavior of NNSG. Analysis is presented to compare NNSG and existing graph-based learning methods and linear regression methods, which is very useful to provide some insight for explaining these methods.

This paper is organized as follows: Section II introduces some related works. NNSG and the corresponding solution are described in Section III. In Section IV, discussions are provided. Extensive experiments are conducted in Section V. Finally, some remarks are given in Section VI.

II. Related Works

Since our work in this paper is based on FME [33] and LCC [37], we briefly review their formulations for the sake of completeness. Sample set is denoted as $X = [x_1, x_2, ..., x_u, x_{u+1}, ..., x_n] \in \Re^{m \times n}$, where $x_i|_{i=1}^{u}$ and

$x_j|_{j=u+1}^{n}$ are labeled and unlabeled samples, respectively. The labels of labeled samples are denoted as $y_i \in \{1, 2, ..., c\}$, where c is the total number of classes. The label indicator binary matrix Y is defined as follows: for each training sample $x_i (i = 1, ..., n)$, $y_i \in \Re^c$ is its label vector, if x_i is from the kth class ($k = 1, ..., c$), then only the kth entry of y_j is one and all the other entries are zero.

A. Flexible Manifold Embedding (FME)

In this section, FME [33] is briefly reviewed. In LapRLS/L [32], the predicted label matrix F is constrained to lie within the space spanned by all training samples X. In other words, $F = X^T W + \mathbf{1}b^T$ is approximately satisfied, in which $\mathbf{1} \in \Re^{n\times 1}$ is defined as a vector with all elements as 1. $W \in \Re^{m\times c}$ is the projection. $F = X^T W + \mathbf{1}b^T$ may be too strict to fit the samples sampled from a nonlinear manifold because it is linear. Therefore, it is relaxed by modeling a regression residue in FME. That is, $F = X^T W + \mathbf{1}b^T + F_0$ is assumed to be satisfied, where F_0 is the regression residue modeling the mismatch between F and $X^T W + \mathbf{1}b^T$. The advantage of this relaxation is that it is more flexible to cope with the samples which reside on the nonlinear manifold. FME aims to simultaneously predict the sample label matrix F and minimize the regression residual F_0, namely

$$(F^*, W^*, F_0^*) = arg \min_{F,W,F_0} tr(F - Y)^T U(F - Y) + tr(F^T HF) + \mu(\| W \|^2 + \gamma \| F_0 \|^2) \quad (1)$$

where μ and γ are parameters to balance different terms, $H \in \Re^{n\times n}$ is the Laplacian matrix and $U \in \Re^{n\times n}$ is the diagonal matrix with the first μ and the rest of $n - \mu$ are diagonal elements 1 and 0, respectively. $tr(\cdot)$ is the trace of a square matrix. The first term in (1) represents the label fitness. The second term in (1) is the graph embedding which is used to propagate labels from labeled samples to unlabeled samples. The last term in (1) controls the norm of projection matrix and regression residue F_0. Replacing F_0 with $X^T W + \mathbf{1}b^T - F$, the objective function of FME can be reformulated as

$$(F^*, W^*, b^*) = arg \min_{F,W,b} tr(F - Y)^T U(F - Y) + tr(F^T HF) + \mu(\| W \|^2 + \gamma \| X^T W + \mathbf{1}b^T - F \|^2) \quad (2)$$

B. Local Coordinate Coding (LCC)

Sparse representation [40] has been successfully applied in machine learning and computer vision. Wang et al. pointed out that locality is more essential than sparsity, as locality must lead to sparsity but not necessarily vice versa [35], [41], [42]. The goal of LCC [37] is to seek a new representation of the input samples in which the samples are encoded by a linear combination of several nearby samples among a given dictionary. The input samples can be transformed into more discriminative codes through the coding process.

Given a dictionary $D = [d_1, d_2, ..., d_K]$ consisting of K atoms with dimension of m, LCC computes a reconstruction coefficient vector $s_i \in \Re^K$ to represent the input sample $x_i \in \Re^m$ by minimizing the following objective function

$$\min_{s_i} \| x_i - Ds_i \|_2^2 + \lambda \sum_{k=1}^{K} \left| s_i^k \right| \| D_{*k} - x_i \|_2^2 \quad (3)$$

where D_{*k} is the kth column of dictionary D and s_i^k denotes the kth element of the coding coefficient vector s_i. The regularization term $\sum_{k=1}^{K} \left| s_i^k \right| \| D_{*k} - x_i \|_2^2$ ensures that each input sample can be locally approximated by a linear combination of its nearby atoms in the dictionary.

III. NNSG

In this section, the non-negative sparse graph (NNSG) learning for linear regression is introduced. The task of NNSG is to perform the linear regression and graph learning simultaneously. The linear regression is used to learn a projection $W \in \Re^{m\times c}$ for fitting sample labels $F \in \Re^{n\times c}$ and classifying new samples. Thus the linear regression can be formulated as

$$F = X^T W + F_0 \quad (4)$$

where F_0 is the regression residual [33]. Before introducing the objective function of NNSG, we firstly present how to learn a reasonable graph for label propagation.

A. The Graph Learning

For label propagation, it usually has the following assumption: a sample and its nearest neighbors usually belong to the same class. In the process of label propagation, these nearest neighbors make great contribution in determining the label of this sample. Similar samples should have similar neighbors which is very helpful to propagate similar labels for these similar samples. Therefore, an ideal graph should capture such neighbor and similarity structure by allocating larger weights to these nodes associated with this sample and its nearest neighbors. The regularization term in (3) can automatically determine the sample neighbors by using a measurement of the distance between the sample and the atoms in the dictionary and simultaneously assign larger weights to these sample neighbors. However, the obtained reconstruction coefficient in (3) cannot be directly used as the indication of graph weight since it may be negative. Thus, we propose the following objective function for the graph learning

$$\min_{S} \| X - XS \|_F^2 + \lambda tr(\Xi(S \odot M)) \quad \text{s.t. } S \geq 0, \quad S_{ii} = 0 \forall i \quad (5)$$

where $M \in \Re^{n\times n}$, $M_{ij} = \| X_{*i} - X_{*j} \|_2^2$, $\Xi \in \Re^{n\times n}$ is a matrix with all elements as 1. $\odot$ is a Hadamard product operator of matrices. The reconstruction coefficient matrix S in (5) essentially reflects a close relation between the sample pairs. $S_{ii} = 0$ is used to avoid the trivial solution of (5). The first term in (5) is to minimize the linear reconstruction error. It should be noted that the LLE [14] framework also tries to minimize the linear reconstruction error. However, the minimization of linear reconstruction error in LLE is only processed within the sample neighbors defined by the k nearest

neighbor or the ϵ-neighbor graph methods. Thus the structure of graph adjacency has been determined by the previous step and LLE only generates the corresponding graph weights. In this way, the graph deduced by LLE is not optimal. The structure of graph adjacency and weights are simultaneously determined in (5) and thus it is natural to utilize (5) for the graph learning.

The regularization term $tr(\Xi(S \odot M))$ has three advantages: 1) It can ensure that each sample is more accurately represented by multiple nearby samples and larger weights can be assigned to these nearby samples simultaneously, which is useful for label propagation. 2) It allows nearby samples to have nearly similar coding coefficients. Thus this coding captures the similarity among samples by sharing similar reconstruction coefficients. 3) It allows the graph construction process to have sparsity, adaptive neighborhood and nonnegative weights. Thus, it is reasonable that S is used as the graph weight matrix.

B. Learning a Non-Negative Sparse Graph for Linear Regression

In NNSG, the graph learning and linear regression are simultaneously completed within one step to guarantee an overall optimum. The graph embedding (manifold smoothness [33]) is used to link these two tasks. Based on FME and the graph learning proposed in (5), we propose the following objective function for NNSG.

$$\begin{aligned}&\Theta(F, W, S)\\ &= arg \min_{F,W,S} \gamma_\infty \sum_{i=1}^{u} \parallel F_{i*} - Y_{i*} \parallel_2^2\\ &\quad + \sum_{i}^{n}\sum_{j}^{n} \parallel F_{i*} - F_{j*} \parallel_2^2 S_{ij} + \alpha \parallel X^T W - F \parallel_F^2\\ &\quad + \lambda tr(\Xi(S \odot M)) + \beta \parallel X - XS \parallel_F^2\\ &\quad \text{s.t. } S \geq 0, \quad S_{ii} = 0 \ \forall i \end{aligned} \tag{6}$$

where γ_∞ is a very large number such that $F_{i*} = Y_{i*}$ $(i = 1, 2, ..., u)$ can be approximately satisfied, and $F \in \Re^{n \times c}$ is the predicted labels of both labeled and unlabeled samples. The other variables follow the same definitions as in (4) and (5). The first term evaluates the label fitness (i.e., F should be close to the labels of the labeled samples). The second term is the graph embedding (i.e., F should vary smoothly along the geodesics on the whole graph of both labeled and unlabeled samples) which is used to link the graph learning and linear regression and propagate labels from the labeled samples to the unlabeled samples. For the sample x_i, the larger the weight S_{ij}, the more contribution the label F_{j*} of the sample x_j offers to the prediction of the label F_{i*} for the sample x_i. The purpose of the third term is to minimize the regression residual. The linear regression is used to learn the projection for fitting samples labels and classifying new samples. The last two terms aim to learn the graph structure. Three parameters α, λ, and β are given to balance the importance of the corresponding three terms. (6) can be further formulated as follows

$$\begin{aligned}\Theta(F, W, S) = arg \min_{F,W,S} &tr((F - Y)^T U (F - Y))\\ &+ tr(F^T L F) + \alpha \parallel X^T W - F \parallel_F^2\\ &+ \lambda tr(\Xi(S \odot M)) + \beta \parallel X - XS \parallel_F^2 \end{aligned} \tag{7}$$

where $L = D - S$ is graph Laplacian, in which D is a diagonal matrix with $D_{ii} = \frac{\sum S_{i*} + \sum S_{*i}}{2}$. $U \in \Re^{n \times n}$ is a diagonal matrix with the first u and the rest of $n - u$ diagonal elements as γ_∞ and 0, respectively.

The optimization problem of (7) can be solved by updating F and S iteratively and calculating W independently.

Firstly, W is solved when F and S are fixed. The optimization problem defined in (7) is written as

$$\Theta(W) = arg \min_{W} \parallel X^T W - F \parallel_2^2 \tag{8}$$

It is an unconstrained optimization problem. If the derivative of (8) with respect to W is set equal to zero, we will have

$$\frac{\partial \Theta(W)}{\partial W} = XX^T W - XF = 0 \Rightarrow W = AF \tag{9}$$

where $A = (XX^T)^{-1}X$, if XX^T is a singular square matrix, $A = (XX^T + \tau I)^{-1}X$, where τ is a small positive constant and I is the identity matrix.

Secondly, F is solved when W and S are fixed. The optimization problem defined in (7) is written as

$$\begin{aligned}\Theta(F) = arg \min_{F} &tr((F - Y)^T U (F - Y))\\ &+ tr(F^T L F) + \alpha \parallel X^T W - F \parallel_F^2 \end{aligned} \tag{10}$$

It is an unconstrained optimization problem. If $W = AF$ is integrated and the derivative of (10) with respect to F is set equal to zero, we have

$$\begin{aligned}&\frac{\partial \Theta(F)}{\partial F} = UF - UY + LF + \alpha ZF = 0\\ &\Rightarrow F = (U + L + \alpha Z)^{-1} UY \end{aligned} \tag{11}$$

where $Z = ((X^T A - I)^T (X^T A - I))$.

Finally, S is solved when W and F are fixed. The optimization problem defined in (7) is written as

$$\begin{aligned}\Theta(S) = arg \min_{S} &tr(F^T L F) + \lambda tr(\Xi(S \odot M))\\ &+ \beta \parallel X - XS \parallel_F^2\\ &\text{s.t. } S \geq 0, \quad S_{ii} = 0 \forall i \end{aligned} \tag{12}$$

It is also a constrained optimization problem, (12) can be rewritten as

$$\begin{aligned}\Theta(S) = arg \min_{S} &tr(\Xi(S \odot R)) + \beta \parallel X - XS \parallel_F^2\\ &\text{s.t. } S \geq 0, \quad S_{ii} = 0 \forall i \end{aligned} \tag{13}$$

where $R = \lambda M + V$. $V_{ij} = \parallel F_{i*} - F_{j*} \parallel_2^2$, F_{i*} is the ith row of F. The optimization problem in (11) can be decomposed into n independent sub-problems for each coding coefficient S_{*i} corresponding to the sample X_{*i} and each sub-problem is a weighted nonnegative sparse coding problem.

$$\begin{aligned}&\min_{S_{*i}} \sum_{k=1}^{n} R_{*i}^k S_{*i}^k + \beta \parallel X_{*i} - XS_{*i} \parallel_F^2\\ &\text{s.t. } S \geq 0, \quad S_{ii} = 0 \forall i \end{aligned} \tag{14}$$

Algorithm 1 NNSG

Input: Training samples set X; Label matrix Y; Matrix U; Parameter α, λ and β;
Output: The projection matrix W; The predicted label matrix F
Initialization: $S = \mathbf{1}_{n\times n}$;
Set $t = 0$;
repeat
1. Update F by solving (11).
2. Update S by solving (12).
3. Update $t = t + 1$.

until *Convergence*
Calculate $W = AF$

where S_{*i}^k is the kth element of the coding coefficient vector S_{*i} and R_{*i}^k is the kth element of the ith column of matrix R. Many algorithms can be used to solve (14), such as fast iterative shrinkage and thresholding (FISTA) [36] and basis pursuit (SP) [40]. In this paper, the alternating direction method (ADM) [43], [44] is used to solve (14). It is a popular and efficient method for solving the weight non-negative sparse coding problem. This method can be regarded as a first-order primal-dual algorithm since both primal and dual variables are updated at each iteration.

From the above deduction procedure of solving problem (7), we can see that variables S and F are closely dependent. However, variable W is only related to variable F. Thus, we only need to iteratively update S and F. After obtaining the optimal solution of F, we can calculate W by $W = AF$.

The overall algorithm of NNSG is described in Algorithm 1.

IV. Algorithm Analysis

In this section, related works are discussed, and then the algorithmic convergence property is analyzed. It should be noted that, to our knowledge, NNSG is the first work which integrates the graph learning, label prediction, label propagation and projection learning into one step. For simplicity of discussion, in the following analysis, the graph Laplacian matrix L is assumed to be given in advance in NNSG.

A. Connection to Graph-Based Methods

Local and global consistency (LGC) [28] and Gaussian fields and harmonic function (GFHF) [29], two prominent G-SSL methods can be seen as special cases of NNSG and they share the same formulation [33]

$$\min_F tr(F^T LF) + tr((F-Y)^T U(F-Y)) \tag{15}$$

where L is the (normalized) graph Laplacian matrix. If $\alpha = \lambda = \beta = 0$ and the graph structure is pre-defined, then the objective function of NNSG will degrade into (15) which is a general formulation for LGC and GFHF. Thus LGC and GFHF are the special cases of NNSG. LGC and GFHF have to encounter an out-of-sample problem [33] because they did not deal with new samples. However, NNSG can map new samples that are not included in the training set by using the learned projection.

LapRLS/L [32] is a G-SSL method which is equivalent to the special case of NNSG. If we set $\lambda = \beta = 0$ and $\alpha \to \infty$, then $F = X^T W$. Replacing F to (7), (7) can be rewritten as

$$\min_F tr((X^T W - Y)^T U(X^T W - Y)) + tr(W^T XLX^T W) + \mu \parallel W \parallel_F^2 \tag{16}$$

The regularization term $\parallel W \parallel_F^2$ is added to control the norm of W. If the bias term b is absorbed, the objective function of LapRLS/L becomes

$$\min_W \frac{1}{\lambda_I} \parallel X^T W - Y \parallel_2^2 + tr(W^T XLX^T W) + \frac{\lambda_A}{\lambda_I} \parallel W \parallel_F^2 \tag{17}$$

If it is further assumed that L in (16) is the graph Laplacian matrix, $\mu = \frac{\lambda_A}{\lambda_I}$ and the first u and the rest of $n-u$ diagonal elements of U in (16) are $\frac{1}{\lambda_I}$ and 0, respectively, then (16) is equal to (17). Thus, LapRLS/L is also equivalent to a special case of NNSG.

FME and NNSG have some similar components. If $\lambda = \beta = 0$ is set and L is pre-defined as the graph Laplacian matrix in order to prevent the trivial solution of S, the objective function of NNSG can be represented as

$$\min_{F,W} tr((F-Y)^T U(F-Y)) + tr(F^T LF) + \alpha \parallel X^T W - F \parallel_F^2 \tag{18}$$

It can be found that the first two terms in FME and NNSG have the same purposes.

B. Connection to Linear Regression Methods

Hou et al. [45] proposed a feature selection method via embedding learning and sparse regression (JELSR). The objection function of JELSR is as follows

$$\min_{YY^T = I} tr(YLY^T) + \beta(\parallel W^T X - Y \parallel_F^2 + \alpha \parallel W \parallel_{2,1}) \tag{19}$$

where Y is the lower dimension representation of samples and $\parallel W \parallel_{2,1}$ is used to select the related features. JELSR uses a linear regression to relax the hard constraint in LapRLS/L. If $U \to 0$ and $\lambda = \beta = 0$ are set, then a new formulation for NNSG is obtained

$$\min_{F,W} tr(F^T LF) + \alpha \parallel X^T W - F \parallel_F^2 \tag{20}$$

The first two terms in (19) are the same as (20). In this way, JELSR and NNSG share the same goals in terms of the linear regression and graph embedding.

Similarly, Ma et al. [46] proposed a semi-supervised feature analyzing framework for multimedia data understanding and named structural feature selection with sparisity (SFSS). The objective function of SFSS is as follows

$$\min_{F,W} tr((F-Y)^T U(F-Y)) + tr(F^T LF) + \mu \parallel X^T W - F \parallel_F^2 + \gamma \parallel W \parallel_{2,1} \tag{21}$$

where the bias term b is also absorbed. If $\lambda = \beta = 0$ is set, then (7) will degraded into (18) which is the same as (21) besides there is no the regularization term $\| W \|_F^2$ in NNSG in this case.

Other works including the spectral embedding clustering (SEC) framework [47] and the spectral regression (SR) [48] have some similar components as NNSG. The related analyses are similar to that in [33].

C. Convergence Analysis

The optimization problem in (7) is non-convex due to the non-convexity of objective function, and thus the globally optimal solution cannot be guaranteed. Here, we can prove that the iterative procedure will converge to a local optimum and objective function of NNSG has a lower bound (at least bigger than a constant $\delta > 0$). From the inner loop of the iteration by solving the subproblems (11) and (12), we know that for each iteration the variables solution makes the objective function achieve a local minimum. Therefore, in the outer loop, we have $\Theta(F^t, S^t) \geq \Theta(F^{t+1}, S^t) \geq \Theta(F^{t+1}, S^{t+1})) \geq \delta > 0$.

Therefore, the objective function will converge to a local optimum.

D. Computation Complexity of NNSG

This section analyzes the computation complexity of NNSG described in Algorithm 1. The most computationally demanding steps of Algorithm 1 are step 1 and 2. It is easy to justify that calculating A will be $\mathcal{O}(2m^2n + m^3)$, and calculating Z will be $\mathcal{O}(n^2m + n^3)$. Thus, in total, the computation complexity of step 1 will be up to $\mathcal{O}(n^3 + max\{n^3, n^2c\})$. The computation complexity of step 2 will be up to $\mathcal{O}(\imath n^3)$, where $\imath$ is the number of iteration of calculating S. Another computation complexity of calculating W will be $\mathcal{O}(nmc)$. The computation complexity of algorithm 1 will be $\mathcal{O}(\tau(n^3 + max\{n^3, n^2c\} + \imath n^3) + (nmc) + (2m^2n + m^3 + n^2m + n^3))$, where τ is the number of outside loop.

E. Discussion

By using the obtained projection W, the novel samples can be accurately classified. From Algorithm 1, it can be see that W is independent to the iterative process. After obtaining the optimal solution of F, we can calculate W by $W = AF$. As clearly shown in (11) (i.e., $F = (U + L + \alpha Z)^{-1}UY)$), the discriminant information contained in the learned graph (i.e., graph Laplacian L) is delivered into W by $W = AF$ so that the obtained projection matrix W can accurately classify the novel samples. In other words, although W do not involve the iterative process in Algorithm 1, W can obtain its optimal solution due to the optimal F. The subsequent experimental results show that the obtained W is competent to classify the novel samples. The classification performance of W relies on the quality of the learned graph S. If the obtained graph S is optimal, W can classify the novel samples well. However, in some real-world applications, data set may be corrupted by the noise such as illumination and expression which can lead to a low-quality graph. Thus, the classification performance of W may slightly degrade on such data set. For example, since large variations occur on the YaleB data set, the classification accuracy of W slightly degrades on the data set (see TABLE VI).

V. Experiments

In our experiments, seven data sets were used, including four face data sets ORL [49], YALEB [33], GEORGIA TECH (GT) [42], AR [42], [49], an object data set COIL-20 [37], a handwritten digit data set USPS [50], and a spoken letter recognition data set Isolet5 [50]. The MATLAB code of NNSG is publicly available at http://www.yongxu.org/lunwen.html.

A. Data Sets Descriptions

1) Face Data Sets: The images in all face data sets were cropped and then resized to 32×32 pixels. The ORL data set consists of 400 face images of 40 peoples. The images were taken at different times, with varying lighting, facial expressions (open/closed eyes, smiling/not smiling), and facial details (glasses/no glasses). For YALEB data set, 38 persons are used in this paper, with each person having around 64 near frontal images under different illuminations. The GT face data set contains of 50 peoples with 15 images per person. The images were taken with several variations such as pose, expression, cluttered background, and illumination. For AR data set, 3,120 gray images from 120 peoples were used with each people providing 26 images. These images were generated from frontal view faces with different facial expression, conditions of illumination, and occlusion (sun glasses and scarf).

2) Object Data Set: The COIL-20 data set consists of images of 20 objects, and each object has 72 images captured from varying angles at intervals of five degrees. All images were cropped and then resized to 32×32 pixels.

3) Handwritten Digit Data Set: The USPS has training set of 7291 samples and test set of 2007 samples. In the experiments, 7000 images from 10 projects were randomly selected with each project providing around 700 images. Each image was of size 16×16 pixels.

4) Spoken Letter Recognition Data Set: The Isolet spoken letter recognition data set contains 150 subjects who spoke the name of each letter of the alphabet twice. The speakers are grouped into sets of 30 speakers each, and are referred to as Isolet1 through isolet5. In this work, Isolet5 data set consisting of 1559 images from 26 peoples was used with each subject providing about 60 images. This data set can be downloaded at http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html.

For the sake of computational efficiency, all data in these seven data sets were eventually reduced to 60D vectors by PCA.

B. Visualization of Graph Weight Matrix by Learned From NNSG and Some Conventional Methods

In this section, the visual property of the non-negative sparse graph weight matrix S learned by NNSG is demonstrated. Specifically, the graphs used in the experiments for

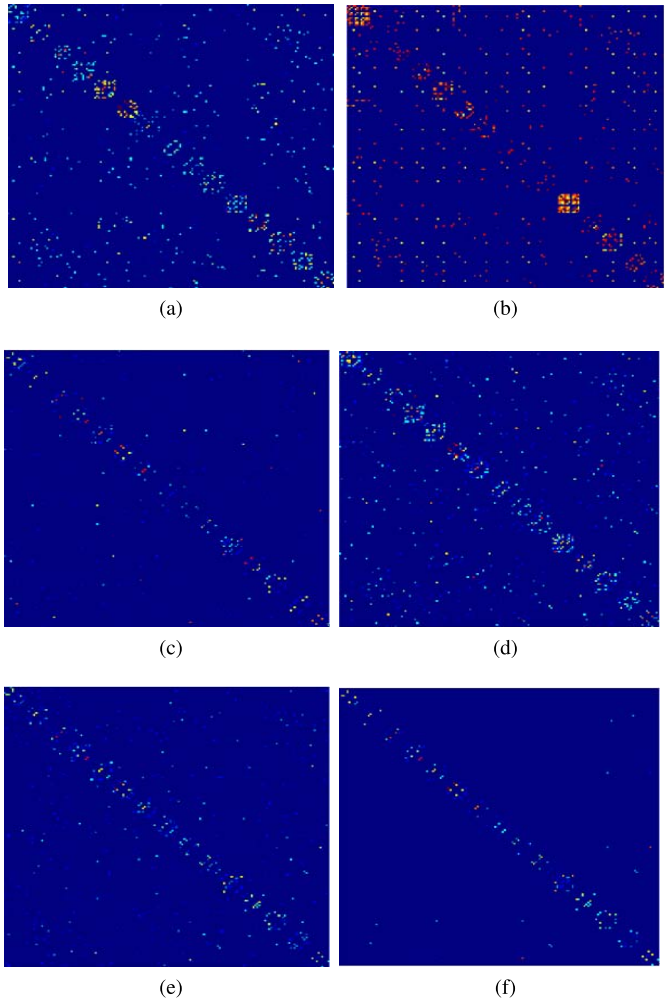


Fig. 1. Visualization of the graph weight matrix of the (a) k nearest neighbor graph, (b) ϵ neighbor graph, (c) ℓ_1 graph, (d) LLE graph, (e) SPG, and (f) NNSG graph.

comparison include: the k nearest neighbor graph (k-NN), ϵ neighbor graph, LLE graph [51], ℓ_1 graph [34], and sparse probability graph (SPG) [52] (SPG in essence is a sparse coding problem with the non-negativity constraint). For the LLE and ℓ_1 graph, the weight matrix S may be negative and asymmetric and they were symmetrized by $S = \frac{(|S|+|S^T|)}{2}$. We construct these different graphs on the YALE [51] face data set. The introduction of this data set is shown in [51] and all images were finally also reduced to 60D vectors by PCA. The first five images per subject were selected as labeled samples and the rest of the images were selected as unlabeled samples. For all graphs, model (7) was used as the baseline to perform semi-supervised classification (the Laplacian matrix L is pre-determined in conventional methods by using the corresponding methods). For all compared graphs, model parameters were carefully chosen to obtain the best semi-supervised classification accuracies, and then the graphs were displayed. The graph weight matrices are shown in Fig. 1. From this figure, two observations can be made: 1) The edges in NNSG are sparser than others. 2) There is much less inter-subject adjacency structure in the NNSG graph than in the others, which means the NNSG graph delivers strong discriminant information and thus is more effective for label propagation and classification than the other graphs.

C. Semi-Supervised Classification With Different Graphs

In this subsection, the semi-supervised classification was done on the ORL, AR, YALEB, GT, and Isolet5 data sets using the above mentioned different graphs. For fair comparison, NNSG was also used as the baseline to perform the semi-supervised classification. For the k-NN graph, ϵ neighbor graph, LLE graph, ℓ_1 graph and SPG graph, the graph Laplacian matrix L was also pre-defined by the corresponding graph methods. The classification method proposed in [28] was used for classification and the parameters were carefully chosen to obtain the best classification results. For NNSG, only the effectiveness of semi-supervised classification was tested. The classification performance of the learned projection W is tested in the next subsection. The value of parameters α, λ, and β of NNSG were selected from the range $(10^{-4}, 10^{0})$, respectively. For each data set, different number of samples per subject were randomly selected as the labeled samples and the remaining were used as unlabeled samples and this process was repeated 20 times and then the mean classification accuracy and standard deviation are reported in Table I, from which it can be observed that:

1) In most cases, NNSG consistently achieves the highest classification accuracy in comparison with the other graphs, even with very small amount of labeled samples. In many cases, the improvements are very significant. This means that the graph learned from NNSG is more effective for label propagation and discriminant analysis.

2) The performance of the ℓ_1 graph and SPG graph is similar in many cases since they all solve the sparse coding problem besides a non-negative constraint in the SPG graph. Although both the SPG graph and NNSG need to solve non-negative sparse coding problem for constructing the graph structure, NNSG integrates the graph learning and linear regression into one step which can guarantee an overall optimum. In other words, the graph structure learned by NNSG can effectively capture the sample adjacent structure to accurately propagate label information, thus NNSG outperforms the SPG graph in most cases.

3) The performance of the LLE graph is consistently better than that of the k-NN graph and ϵ neighbor graph. In this experiment, it was found that it is extremely difficult to find a proper ϵ for ϵ neighbor. On the contrary, the k-nearest neighbor graph is generally more robust than the ϵ neighbor graph. This observation is consistent with the conclusion in [51].

D. Semi-Supervised Classification and New Samples Classification

In this subsection, NNSG is compared with FME [33], GFHF [29], Transductive component analysis (TCA) [53], SDA [26], LapRLS/L [27], and MFA [20] for semi-supervised classification and new samples classification by the learned projection. For FME, SDA, LapRLS/L, MFA, TCA, the nearest neighbor classification was used for classification.

TABLE I

CLASSIFICATION PERFORMANCE (MEAN CLASSIFICATION ACCURACY ± STANDARD DEVIATION%) OF DIFFERENT GRAPHS AND NNSG ON FIVE DATA SETS. NOTE THAT THE BOLD NUMBERS ARE THE BEST ACCURACIES FOR EACH CONFIGURATION AND THE NUMBERS IN PARENTHESIS ARE THE NUMBER OF THE LABELED SAMPLES

Data sets (label #)	k-NN	ϵ neighbor	LLE	ℓ_1 graph	SPG	NNSG
ORL (5)	83.70±1.42	68.12±2.08	88.97±2.04	91.13±2.33	89.67±2.26	**94.00±2.26**
ORL (6)	85.25±2.57	69.22±3.15	91.29±2.13	92.74±1.94	92.52±2.12	**96.19±1.46**
ORL (7)	86.42±3.14	70.42±2.72	93.19±1.96	94.52±2.13	94.03±2.19	**96.63±2.36**
GT (5)	65.71±1.65	48.91±2.18	72.33±1.28	67.58±4.73	63.10±2.15	**73.09±2.14**
GT (8)	69.09±2.15	52.02±2.06	**78.19±1.94**	72.75±4.14	72.49±2.06	**78.19±1.73**
GT (11)	72.99±2.59	53.27±2.51	80.35±2.45	80.37±2.45	79.10±2.10	**82.08±1.89**
AR (7)	55.13±1.03	42.29±0.86	63.30±0.95	66.84±3.80	78.98±2.33	**83.87±0.49**
AR (14)	62.78±0.96	49.03±1.21	76.14±1.15	86.94±1.01	86.43±2.11	**94.13±0.69**
AR (21)	67.42±1.54	52.80±1.52	83.92±1.44	90.54±1.05	90.01±1.01	**97.62±0.83**
YALEB (5)	46.16±1.37	30.08±1.64	49.17±1.52	44.37±1.41	72.20±2.03	**75.73±1.45**
YALEB (10)	52.45±0.99	36.40±1.26	56.23±0.97	51.52±3.54	82.13±2.05	**84.75±0.91**
YALEB (15)	55.38±0.89	39.52±1.00	60.15±0.92	53.64±3.02	85.39±2.42	**89.37±0.77**
Isolet5 (10)	71.39±1.57	61.87±2.14	72.19±1.32	73.26±1.52	71.00±1.43	**81.23±1.51**
Isolet5 (20)	76.91±1.01	69.05±1.60	78.64±0.83	80.59±1.48	79.95±1.22	**86.10±1.42**
Isolet5 (30)	79.67±1.12	72.85±1.67	82.14±1.62	84.64±1.47	83.67±1.21	**88.07±0.98**

TABLE II

CLASSIFICATION RESULTS FOR UNLABEL AND TEST USING THE COIL-20 DATA SET

Method	$p=1$		$p=2$		$p=3$	
	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
GFHF	79.46±2.79	-	82.14±1.83	-	84.34±1.73	-
MFA	-	-	71.37±2.77	71.61±2.66	77.49±2.54	77.44±2.49
SDA	66.27±2.10	66.40±2.25	73.85±2.99	74.10±3.03	80.76±2.94	80.13±3.02
TCA	74.00±2.24	73.08±2.67	78.92±2.99	77.76±2.34	81.05±2.27	80.11±2.30
LapRLS	71.08±2.95	71.19±2.38	77.28±3.00	77.82±3.10	81.96±2.54	80.16±2.23
FME	79.91±2.52	71.37±2.27	83.25±2.25	77.42±1.65	85.61±1.85	80.13±1.58
NNSG	**82.08±2.71**	**75.00±2.50**	**85.90±2.46**	**79.26±2.62**	**87.64±2.83**	**80.22±1.53**

TABLE III

CLASSIFICATION RESULTS FOR UNLABEL AND TEST USING THE USPS DATA SET

Method	$p=1$		$p=2$		$p=3$	
	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
GFHF	74.44±3.88	-	81.60±3.76	-	85.47±3.24	-
MFA	-	-	68.83±3.95	68.47±4.62	74.83±4.80	72.68±3.61
SDA	58.83±3.95	56.61±3.78	69.09±3.50	69.31±2.78	74.55±3.00	71.56±2.66
TCA	73.33±3.27	66.90±3.36	79.78±3.26	74.00±3.82	83.40±3.91	75.77±2.34
LapRLS	58.39±4.69	60.56±4.64	70.00±4.20	70.53±2.86	77.89±3.39	75.75±3.66
FME	77.35±6.77	67.66±4.98	82.83±3.72	74.70±3.45	85.33±2.40	76.00±2.23
NNSG	**79.89±4.00**	**68.92±3.85**	**84.51±2.47**	**76.32±2.75**	**87.90±2.33**	**78.90±2.71**

TABLE IV

CLASSIFICATION RESULTS FOR UNLABEL AND TEST USING THE ISOLET5 DATA SET

Method	$p=1$		$p=2$		$p=3$	
	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
GFHF	50.72±2.51	-	57.30±2.57	-	61.93±2.09	-
MFA	-	-	63.03±2.13	63.47±3.07	68.00±2.19	68.87±2.52
SDA	54.50±2.54	53.11±2.24	63.32±2.93	63.75±2.69	**69.03±2.15**	69.00±2.07
TCA	50.57±2.78	50.33±3.15	61.46±2.92	61.02±2.61	66.50±2.24	67.01±2.16
LapRLS	53.53±3.10	52.91±3.22	63.52±2.75	63.63±3.14	67.33±1.97	67.40±1.96
FME	50.26±2.91	50.41±2.73	61.50±2.75	61.15±2.82	67.79±1.80	68.12±2.08
NNSG	**55.26±2.65**	**53.83±2.52**	**64.09±2.65**	**64.01±2.63**	**69.01±2.25**	**70.21±2.07**

For GFHF and NSGG, the classification method proposed in [28] was used for classification. For FME, SDA, LapRLS/L, MFA, and TCA, the graph Laplacian matrix L is needed to be determined beforehand. The graph weight matrix is calculated as $S_{ij} = e^{\frac{-\|x_i - x_j\|^2}{\sigma}}$, if x_i is among the k nearest neighbors of x_j or x_j is among the k nearest neighbors of x_i , $S_{ij} = 0$, otherwise, where k and σ are nearest neighbors and the heat kernel parameter which was selected from the sets $\{3, 4, 5, 6, 7, 8, 9, 10\}$ and $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$, respectively. For FME, SDA, LapRLS/L, MFA, TCA, the final dimension was set to c (the number of classes) for classification and their corresponding two parameters (e.g., μ and γ in FME) were respectively selected from the set $\{10^{-9}, 10^{-8}, ..., 10^{8}, 10^{9}\}$. The parameters α, λ and β of NNSG were selected from the range $(10^{-4}, 10^{0})$, respectively.

TABLE V
CLASSIFICATION RESULTS FOR UNLABEL AND TEST USING THE ORL DATA SET

Method	$p=1$		$p=2$		$p=3$	
	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
GFHF	54.53±4.97	-	65.38±4.03	-	70.75±3.96	-
MFA	-	-	80.83±4.33	81.25±4.35	87.30±4.25	86.50±2.87
SDA	67.19±2.71	67.30±3.06	77.83±3.58	79.28±3.67	84.06±2.58	84.95±1.80
TCA	66.40±2.45	66.03±2.87	79.00±3.01	80.03±2.56	86.45±3.88	86.16±3.10
LapRLS	63.34±3.23	61.92±3.27	80.83±3.76	79.63±2.78	87.00±3.75	87.10±1.86
FME	70.53±2.61	68.03±3.45	82.46±3.52	82.38±2.96	87.25±3.95	86.97±2.31
NNSG	**73.34±3.26**	**69.90±4.13**	**84.17±2.73**	**84.90±2.04**	**88.56±3.98**	**87.33±3.05**

TABLE VI
CLASSIFICATION RESULTS FOR UNLABEL AND TEST USING THE YALEB DATA SET

Method	$p=5$		$p=10$		$p=15$	
	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
GFHF	29.35±1.80	-	36.63±2.15	-	42.12±1.96	-
MFA	-	-	71.92±3.05	72.63±3.23	75.45±3.45	75.66±3.93
SDA	53.27±2.09	53.79±1.69	68.64±1.53	69.61±1.48	76.38±1.71	75.60±1.42
TCA	53.36±2.12	54.06±2.00	67.56±1.78	68.35±2.14	75.56±1.98	75.35±2.21
LapRLS	62.62±2.36	61.96±2.55	76.66±1.50	75.81±1.44	80.87±2.55	**79.89±2.41**
FME	65.72±2.16	65.42±1.92	78.07±2.30	**75.97±1.48**	82.46±2.04	79.39±2.09
NNSG	**74.14±1.95**	**67.36±1.76**	**83.03±1.65**	**75.49±1.70**	**85.43±1.63**	**79.45±1.30**

The experiments are performed on these data sets: COIL-20, USPS, Isolet5, ORL and YALEB. For each data set, 50% of samples per subject were randomly selected as the training samples and the remaining samples were used as the test samples. Among the training samples, p samples per subject were randomly selected as the labeled samples and remaining samples were used as unlabeled samples. The unlabeled samples were used for semi-supervised classification and the test samples were used as the new samples to test the classification performance of the learned projection. For data sets COIL-20, USPS, Isolet5 and ORL, we set $p = 1, 2$ and 3, respectively. For the YALEB data set, we set $p = 5$, 10 and 15, respectively. All training samples were used for learning the projection, except that the labeled samples were used for subspace learning in MFA. The mean classification accuracy and standard deviation (%) over 30 random splits on the unlabeled samples and the test samples, which are respectively referred to as Unlabel and Test, are shown in Tables II-VI. The results with the boldface are better than the others. From these tables, the following conclusions can be drawn:

1) Semi-supervised classification methods TCA, SDA, and LapRLS/L outperform the supervised classification method MFA in terms of classification performance. This indicates that unlabeled samples can indeed improve semi-supervised classification performance.

2) The semi-supervised classification accuracy of GFHF is better than TCA, SDA and LapRLS/L on some data sets. However, when the images in the data sets have strong variations (e.g., strong illuminations and expressions et al in the YALEB face data set), the label may not be correctly propagated in this case, which degrades the performance of GFHF. The phenomenon is more evident on the YALEB data set.

3) NNSG significantly outperforms all compared methods on unlabeled samples, which demonstrates that the graph structure learned by NNSG encodes more discriminant information and can more effectively propagate labels for unlabeled samples. This also demonstrates that it is necessary and effective to simultaneously perform graph learning and label propagation. The classification performance on the test samples using the learned projection of SNGG is not consistently the best on all data sets (FME obtains the best classification result for one case on the YALEB data set), possibly because large variations of images occurred in the data set. This may make the learned projection subject to these variations of images, resulting in slight degradation on classification performance.

E. Parametric Sensitivity and Algorithmic Convergence

NNSG requires three parameters α, λ, and β to be set in advance. In this subsection, their sensitivity is discussed. The classification accuracies variations with different parameters are plotted in Fig. 2. It can be seen that the performance changes are different with respect to different data sets. However, the best classification performances for unlabeled samples and test samples were always achieved with large α and λ when the value of parameter β is fixed. Through tuning the parameters α and λ, it can be observed that the best results were achieved on the given data sets when α and λ were close to one. When they were too small (such as they close to 10^{-3}), the performance obtained was very bad. This demonstrates that the terms in (6) corresponding to α and λ are more significant to learn a discriminative projection. Specifically, the third term (linear regression) in (6) is significant to learn a discriminative projection for fitting labels and classifying new samples, while the fourth term in (6) can guarantee that the learned graph can propagate the correct

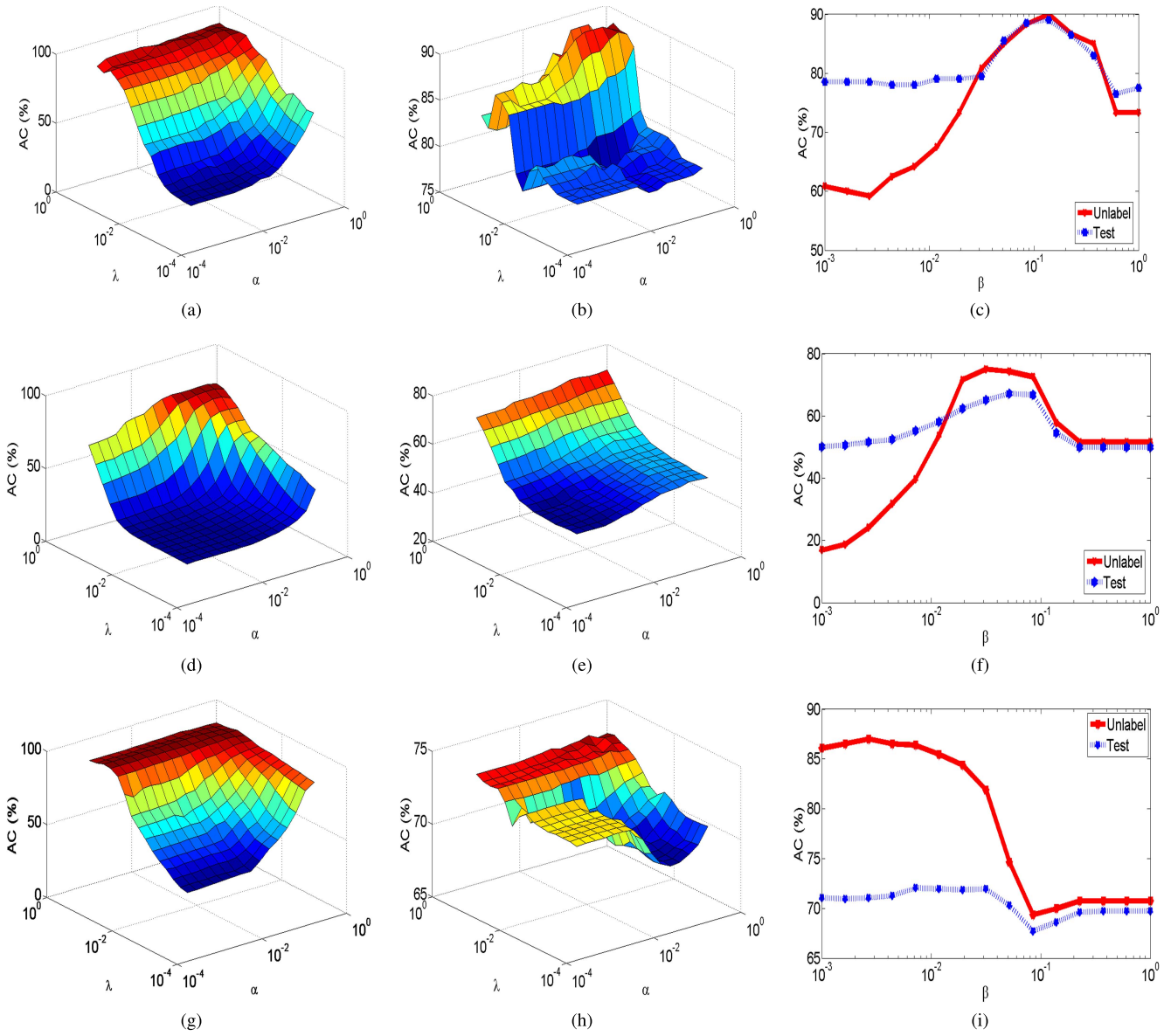
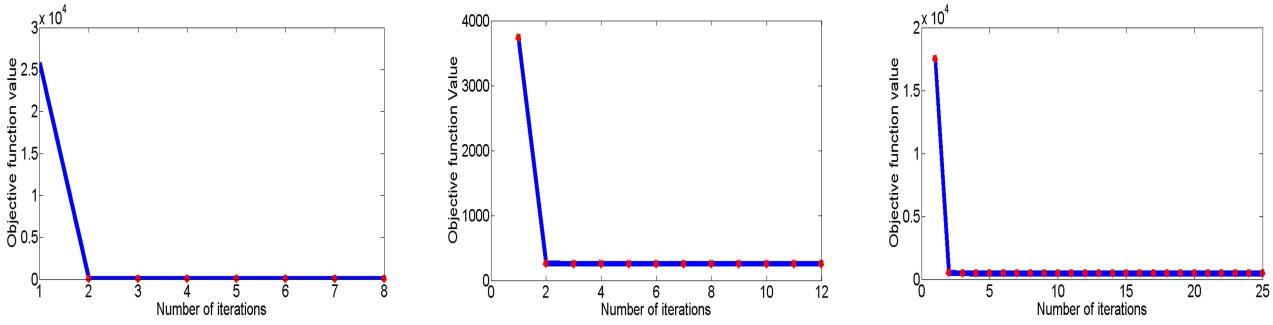


Fig. 2. Classification accuracies (AC) variation different values of parameters α, λ and β on the ORL (The first row), USPS (The second row) and YALEB (The third row) data sets, respectively. 50% of samples per subject were randomly selected as training samples and remaining samples were used as test samples and among training samples two samples per subject were randomly labeled on the ORL and USPS data sets and fifteen samples per subject were labeled on the YALEB data set. The three sub-figures in the first and second columns are the classification accuracy variations with different α and λ on the unlabeled and test samples, respectively, when the value of parameter β was fixed. The last three sub-figures in the third column were the classification accuracy variations with different values of parameter β on the unlabeled and test samples when the values of parameters α and λ were fixed (a) Unlabel (b) Test (c) Unlabel/Test (d) Unlabel (e) Test (d) Unlabel/Test (e) Unlabel (f) Test (g) Unlabel/Test.

label information. Finally, the sensitivity of the parameter β to unlabeled samples and test samples classification was evaluated, when the value of parameters α and λ were fixed. It was observed that NNSG is relatively sensitive to β. From the results in Fig.2, it can be seen that the best classification results were achieved when β was in the middle interval of the tuned range. When it was not too small or large, the performance was good. How to identify the optimal values of these parameters is data set dependent and still an open problem, which will be studied in our future work. In the experiments, β was firstly fixed in advance and an attempt was made to find a candidate interval where the optimal parameters α and λ may exist. Then, by fixing the value of α and λ in the candidate interval, the candidate interval of β was determined. Finally, the optimal parameters in the 3D candidate space of (α, λ, and β) with a fixed step length were searched.

To solve the resulting optimization problem of NNSG, an iterative update rule was developed. The convergence of the update rule was proven in the section IV. The convergence process using the proposed update rule on the ORL, USPS, and YALEB data sets was experimentally shown in Fig.3. From Fig.3, we can see that the proposed update rule converges fast. Such fast convergence is mainly attributed to the process of solving the sparse coding problem (12). During solving



(a) (b) (c)

Fig. 3. Convergence process for NNSG on (a) ORL, (b) USPS and (c) YALEB data sets.

problem (12), some samples make very small contribution in the reconstruction task and label fitness (i.e., the value of R_{ij} is very small) and these samples corresponding weighted value in S are approximately set to zero, which is useful to enhance the sparsity of S. As a result, the obtained S has to be able to effectively capture the adjacency structure of data, thus it can accurately propagate label information so that F can approximately obtain its optimal solution. Moreover, the experiments also showed that NNSG converges fast, usually within 4 iterations for these three data sets, which demonstrates that the proposed update rule is effective.

VI. Conclusion

In this paper, a novel non-negative sparse graph learning method for linear regression is proposed, which simultaneously performs the graph learning and linear regression to seek an overall optimum. The scheme of simultaneous learning makes sure that the graph, in the process of learning, can propagate accurate label information from the labeled samples to unlabeled samples via the graph structure. Thus the linear regression can learn an optimal projection to fit labels and classify new samples. This paper provides an iterative update rule to optimize the corresponding optimization problem and a series of theoretical analysis of the convergence behavior and the comparison of the related methods. Comprehensive experiments on seven different data sets clearly show that the proposed NNSG outperforms existing G-SSL methods. In our future work, we will apply the idea of NNSG to some other machine learning methods.

Acknowledgment

Thanks to Dr. Edward C. Mignot, Shandong University, for linguistic advice.

References

[1] S.-J. Wang, S. Yan, J. Yang, C.-G. Zhou, and X. Fu, "A general exponential framework for dimensionality reduction," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 920–930, Feb. 2014.

[2] T. Zhou and D. Tao, "Double shrinking sparse dimension reduction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 244–257, Jan. 2013.

[3] F. Nie, D. Xu, X. Li, and S. Xiang, "Semisupervised dimensionality reduction and classification through virtual label regression," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 3, pp. 675–685, Jun. 2011.

[4] D. Xu, S. Yan, S. Lin, T. S. Huang, and S.-F. Chang, "Enhancing bilinear subspace learning by element rearrangement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1913–1920, Oct. 2009.

[5] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 342–352, Apr. 2008.

[6] Z. Fan *et al.*, "Modified principal component analysis: An integration of multiple similarity subspace models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1538–1552, Aug. 2014.

[7] X. Li, Y. Pang, and Y. Yuan, "L1-norm-based 2DPCA," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 4, pp. 1170–1175, Aug. 2010.

[8] Z. Fan, Y. Xu, and D. Zhang, "Local linear discriminant analysis framework using sample neighbors," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1119–1132, Jul. 2011.

[9] D. Tao, L. Jin, Y. Wang, and X. Li, "Rank preserving discriminant analysis for human behavior recognition on wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 813–823, Feb. 2014.

[10] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2761–2773, Oct. 2010.

[11] F. Nie, S. Xiang, Y. Jia, and C. Zhang, "Semi-supervised orthogonal discriminant analysis via label propagation," *Pattern Recognit.*, vol. 42, no. 11, pp. 2615–2627, Nov. 2009.

[12] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1178–1188, Jul. 2008.

[13] B. Li, S. Yan, and A. Kassim, "Learning a propagable graph for semisupervised learning: Classification and regression," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 1, pp. 114–126, Jan. 2012.

[14] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[15] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. NIPS*, 2001, pp. 585–591.

[16] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[17] M. Wu, K. Yu, S. Yu, and B. Schölkopf, "Local learning projections," in *Proc. ICML*, 2007, pp. 1039–1046.

[18] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. ICCV*, Oct. 2005, pp. 1208–1213.

[19] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1299–1313, Sep. 2009.

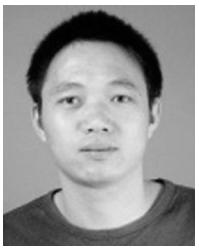
[20] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[21] Y. Luo, D. Tao, B. Geng, C. Xu, and S. Maybank, "Shared feature extraction for semi-supervised image classification," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 1165–1168.

[22] Y. Zhang and D.-Y. Yeung, "Semisupervised generalized discriminant analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 8, pp. 1207–1217, Aug. 2011.

[23] X. Zhu, *Semi-Supervised Learning Literature Survey*. Madison, WI, USA: Univ. Wisconsin-Madison, 2007.

[24] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. ICML*, 1999, pp. 200–209.
[25] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.
[26] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. ICCV*, Oct. 2007, pp. 1–7.
[27] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.
[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. NIPS*, 2003, pp. 1–8.
[29] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. ICML*, 2003, pp. 912–919.
[30] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. KDD*, 2014, pp. 977–986.
[31] X. Fang, Y. Xu, X. Li, Z. Fan, H. Liu, and Y. Chen, "Locality and similarity preserving embedding for feature selection," *Neurocomputing*, vol. 128, pp. 304–315, Mar. 2014.
[32] V. Sindhwani, P. Niyogi, M. Belkin, and S. Keerthi, "Linear manifold regularization for large scale semi-supervised learning," in *Proc. ICML Workshop Learn. Partially Classified Training Data*, 2005, pp. 1–4.
[33] F. Nie, D. Xu, I. W. 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.
[34] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with ℓ_1-graph for image analysis," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 858–866, Apr. 2010.
[35] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.
[36] L. Zhang *et al.*, "Kernel sparse representation-based classifier," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1684–1695, Apr. 2012.
[37] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Proc. NIPS*, 2009, pp. 1–9.
[38] D. Wang, S. C. H. Hoi, Y. He, J. Zhu, T. Mei, and J. Luo, "Retrieval-based face annotation by weak label regularized local coordinate coding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 550–563, Mar. 2014.
[39] B. Liu, M. Wang, R. Hong, Z. Zha, and X.-S. Hua, "Joint learning of labels and distance metric," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 973–978, Jun. 2010.
[40] J. Yang, D. Chu, L. Zhang, Y. Xu, and J. Yang, "Sparse representation classifier steered discriminative projection with applications to face recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1023–1035, Jul. 2013.
[41] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. CVPR*, Jun. 2010, pp. 3360–3367.
[42] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1255–1262, Sep. 2011.
[43] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-problems in compressive sensing," *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, 2011.
[44] Y. Zhang, J. Yang, and W. Yin. (2011). *YALL1: Your Algorithms for L1*. [Online]. Available: http://yall1.blogs.rice.edu/
[45] C. Hou, F. Nie, D. Yi, and Y. Wu, "Feature selection via joint embedding learning and sparse regression," in *Proc. IJCAI*, 2011, pp. 1–6.
[46] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, N. Sebe, and A. G. Hauptmann, "Discriminating joint feature analysis for multimedia data understanding," *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1662–1672, Dec. 2012.
[47] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.
[48] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *Proc. ICCV*, Oct. 2007, pp. 1–8.
[49] Y. Xu *et al.*, "Data uncertainty in face recognition," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1950–1961, Oct. 2014.
[50] [Online]. Available: http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html
[51] S. Yan and H. Wang, "Semi-supervised learning by sparse representation," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2009, pp. 792–801.
[52] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Nonnegative sparse coding for discriminative semi-supervised learning," in *Proc. CVPR*, Jun. 2011, pp. 2849–2856.
[53] W. Liu, D. Tao, and J. Liu, "Transductive component analysis," in *Proc. ICDM*, Dec. 2008, pp. 433–442.



Xiaozhao Fang received the M.S. degree in computer science from the Guangdong University of Technology, Guangzhou, China, in 2008. He is currently pursuing the Ph.D. degree in computer science and technology with the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. He has authored over 15 journal papers. His current research interests include pattern recognition and machine learning.

Yong Xu was born in Sichuan, China, in 1972. He received the B.S. and M.S. degrees in 1994 and 1997, respectively, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, China, in 2005. He is currently with the Shenzhen Graduate School, Harbin Institute of Technology. His current interests include pattern recognition, biometrics, machine learning, and video analysis.

Xuelong Li is currently a Full Professor with the State Key Laboratory of Transient Optics and Photonics, Center for Optical Imagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.

Zhihui Lai received the B.S. degree in mathematics from South China Normal University, Guangdong, China, in 2002, the M.S. degree from Jinan University, Guangdong, in 2007, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2011.

He was a Research Associate and a Post-Doctoral Fellow with the Hong Kong Polytechnic University, Hong Kong, from 2010 to 2013. He is currently a Post-Doctoral Fellow with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Harbin, China. He has authored over 30 scientific papers in pattern recognition and computer vision. His current research interests include face recognition, image processing and content-based image retrieval, pattern recognition, and compressive sense.

Wai Keung Wong received the Ph.D. degree from The Hong Kong Polytechnic University. He is currently with the Institute of Textiles and Clothing, The Hong Kong Polytechnic University, Hong Kong, and The Hong Kong Polytechnic University Shenzhen Research Institute. He has authored over 50 scientific articles in refereed journals, including the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition*, the *International Journal of Production Economics*, the *European Journal of Operational Research*, the *International Journal of Production Research*, *Computers in Industry*, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. His recent research interests include artificial intelligence, pattern recognition, and optimization of manufacturing scheduling, planning and control.