Multimedia Systems (2017) 23:725–738
DOI 10.1007/s00530-016-0521-x

SPECIAL ISSUE PAPER

Semi-supervised image classification via nonnegative least-squares regression

Wei-Ya Ren[1] · Min Tang[2] · Yang Peng[2] · Guo-Hui Li[2]

Published online: 21 June 2016

Abstract Semi-supervised image classification is widely applied in various pattern recognition tasks. Label propagation, which is a graph-based semi-supervised learning method, is very popular in solving the semi-supervised image classification problem. The most important step in label propagation is graph construction. To improve the quality of the graph, we consider the nonnegative constraint and the noise estimation, which is based on the least-squares regression (LSR). A novel graph construction method named as nonnegative least-squares regression (NLSR) is proposed in this paper. The nonnegative constraint is considered to eliminate subtractive combinations of coefficients and improve the sparsity of the graph. We consider both small Gaussian noise and sparse corrupted noise to improve the robustness of the NLSR. The experimental result shows that the nonnegative constraint is very significant in the NLSR. Weighted version of NLSR (WNLSR) is proposed to further eliminate 'bridge' edges. Local and global consistency (LGC) is considered as the semi-supervised image classification method. The label propagation error rate is regarded as the evaluation criterion. Experiments on image datasets show encouraging results of the proposed algorithm in comparison to the state-of-the-art algorithms in semi-supervised image classification, especially in improving LSR method significantly.

Keywords Graph construction · Semi-supervised learning · Label propagation · Least-squares regression · Non-negative constraint

✉ Wei-Ya Ren
weiyren.phd@gmail.com

Min Tang
tangmin@nudt.edu.cn

Yang Peng
pengyang@nudt.edu.cn

Guo-Hui Li
gli2010a@163.com

[1] Department of Management Science and Engineering, Officers College of Chinese Armed Police Force, Chengdu 610213, Sichuan, China

[2] College of Information System and Management, National University of Defense Technology, Changsha 410073, Hunan, China

1 Introduction

Mobile phones are more widely used than ever in our daily lives. Pictures are playing an increasingly important role in mobile communications. With a lot of pictures extensively used in mobile phones, it is a meaningful idea to mark new unlabeled pictures with labels. In this way, we can mark a new picture with some predefined labels. For example, we hope to recognize and tag the acquaintances in new pictures based on their faces automatically in our Facebook account. In fact, this involves the semi-supervised image classification technique. In image classification, labeled images are often scarce and difficult to obtain compared with abundance unlabeled images in the real world. Actually the above image classification technique is the label propagation [4, 5] problem, which utilizes both labeled and unlabeled data. Label propagation is a part of the field of semi-supervised learning [1, 2], and can be treated as one of the graph-based semi-supervised learning methods [3–5].

Many researchers [7–12] find out that the construction of the graph $G = (V, E)$ is the key to label propagation. The vertex set V represents the data points, while the

weight W of the edge set E represents the similarity among data points. The task of graph construction is to explore the weighted edge connection strategy. After the graph being constructed, we can propagate the label information to all unlabeled samples by the graph-based semi-supervised learning method.

Famous graph study methods include: (1) neighbor based methods, e.g., k neighbor (KNN) [6], epsilon ball neighbor ε(-ll) [7]. (2) Local structure based methods, e.g., locally linear embedding (LLE) [8]. (3) Self-representation based methods, e.g., sparse subspace clustering (SSC) [9], low-rank representation (LRR) [10–12], least-squares regression (LSR) [13], smooth representation clustering (SMR) [14].

Assume the dataset is denoted as $X = [x_1, x_2, \ldots, x_n] \in R^{d\times n}$ the objective function of the self-representation-based graph construction method is

$$X = XZ, \quad \text{s.t.}\, Z \neq I, \tag{1}$$

where $Z = [z_1, z_2, \ldots, z_n] \in R^{n\times n}$ is a square matrix, and I is a unit matrix. Z_{ij} denotes the similarity between x_i and x_j. Self-representation-based graph construction assumes that each sample can be represented by a linear combination of all samples, then, the similarity between any two samples can be measured by the linear coefficients. Besides, a symmetry procedure is defined as [7–12]

$$W = (|Z| + |Z^T|)/2.$$

From (1), many researchers consider about various regularization terms of Z. Sparse subspace clustering (SSC) [9] aims to represent a sample by using the least other samples. The mathematic problem of SSC graph is

$$\underset{Z}{\text{argmin}}\, ||Z||_1. \quad \text{s.t.}\, X = XZ, \quad \text{diag}(Z) = 0, \tag{2}$$

where $||Z||_1$ denotes the l_1 norm of Z and $||Z||_1 = \sum_{i=1}^{n}\sum_{j}^{n}|Z_{ij}|$.

Low-rank representation (LRR) [10–12] requires the representation matrix has a low rank structure. The mathematical expression of LLR graph is

$$\underset{Z}{\text{argmin}}\, ||Z||_*, \quad \text{s.t.}\, X = XZ, \tag{3}$$

where $||Z||_*$ denotes the nuclear norm of Z that is, the sum of all eigenvalues of Z.

Least-squares regression (LSR) [13] graph solve the following mathematical problem

$$\underset{Z}{\text{argmin}}\, ||Z||_{\text{F}}. \quad \text{s.t.}\, X = XZ, \quad \text{diag}(Z) = 0, \tag{4}$$

where $||Z||_{\text{F}}$ denotes the Frobenius norm of Z and $||Z||_{\text{F}} = (\sum_{i=1}^{n}\sum_{j=1}^{n} Z_{ij}^2)^{\frac{1}{2}}$.

If the data is noisy, SSC, LRR and LSR graph construction methods usually adopt the following strategy: extend the constraint $X = XZ$ to $X = XZ + \mathcal{S}$, where $\mathcal{S} \in R^{d\times n}$ is the noise matrix. Three regularization terms ($||\mathcal{S}||_1, ||\mathcal{S}||_{2,1}, ||\mathcal{S}||_{\text{F}}$) are utilized for $\mathcal{S}$ in SSC, LRR, LSR, respectively. $||\mathcal{S}||_{2,1}$ is the $l_{2,1}$ norm of $\mathcal{S}$ that is, $||\mathcal{S}||_{2,1} = \sum_{j=1}^{n}\sqrt{\sum_{i=1}^{n}\mathcal{S}_{ij}^2}$. Different regularization terms for $\mathcal{S}$ mean different estimations of noise distribution, e.g., $||\mathcal{S}||_{\text{F}}$ assumes the noise distribution which is approximated to Gaussian noise (others can be seen in references [9, 11, 12] in detail). For example, LSR considers the norm $||\mathcal{S}||_{\text{F}}$ and its mathematical expression is

$$\underset{Z}{\text{argmin}}\, ||X - XZ||_{\text{F}}^2 + \lambda_1||Z||_{\text{F}}^2, \quad \text{s.t.}\, \text{diag}(Z) = 0, \tag{5}$$

where $\lambda_1 > 0$ is the tuned parameter.

Smooth representation clustering (SMR) [14] extends the LSR graph, which requires the smoothness of the representation, i.e., if $x_i \to x_j$ then $z_i \to z_j$, Its mathematical expression can be described as

$$\underset{Z}{\text{argmin}}\, ||X - XZ||_{\text{F}}^2 + \lambda_1 \text{trace}(\text{ZLZ}^T), \tag{6}$$

where L the Laplace matrix of the graph W, $L = D - W$, W the KNN graph, D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{n} W_{ij}$. Notice that SMR graph relies on the KNN graph W. Thus, the quality of the SMR graph will decrease if the quality of the KNN graph W is bad.

The construction of the graph is the key issue in graph-based semi-supervised learning. Thus, the main object is to explore to construct a graph that can reflect the true structure of the data. Many researches have reported that elimination of edges between different categories is the greatest challenge in constructing graphs. An ideal graph should have no connected edges between different categories. Those undesired wrongly connected edges among different categories are often called 'bridge' edges. The graph will improve many properties such as sparsity and block property without the bridge edges. Note that SSC graph and LRR graph try to eliminate 'bridge' edges by leading the graph to be sparse and low rank, respectively. In this paper, our main goal is to propose a novel method to eliminate 'bridge' edges more effectively.

Among all these self-representation-based graph construction methods, we observe that the LSR graph is the most effective method to achieve the self-representation

idea because LSR has the weakest regularization on Z, while the main disadvantage is its poor ability to eliminate the bridge edges, as seen in Fig. 6a. Inspired by this observation, we adopt the objective function of LSR to achieve the self-representation idea and combine the attempt to enhance the ability of eliminating 'bridge' edges. Based on LSR, we propose the nonnegative least-squares regression (NLSR) by adding nonnegative constraint on Z and removing the constraint $\text{diag}(Z) = 0$. The nonnegative constraint on Z can avoid the offset of positive and negative coefficients. And the nonnegative constraint can also improve the sparsity of Z to some extent. The constraint $\text{diag}(Z) = 0$ s removed to further increase the sparsity of Z In addition, the NLSR also considers the sparse corrupted noise E to handle noisy data and improve the robustness of the model. Furthermore, we propose the weighted version of NLSR method to further eliminate 'bridge' edges.

The rest of the paper is organized as follows: In Sect. 2, we introduce the proposed nonnegative least-squares regression (NLSR) method. In Sect. 3, we introduce the weighted NLSR method. Experimental results are presented in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Nonnegative least-squares regression

Assume that $X = [x_1, x_2, \ldots, x_n] \in R^{d\times n}$ denotes a dataset with n sample points, and each sample point has d dimensions. Considering the nonnegative constraint and neglecting the diagonal constraint based on the LSR graph, then (5) can be rewritten as

$$\begin{aligned} &\arg_Z \min ||X - XZ||_{\text{F}}^2 + \lambda_1 ||Z||_{\text{F}}^2, \\ &\qquad\qquad \text{s.t.}\, Z \geq 0, \end{aligned} \tag{7}$$

where $Z = [z_1, z_2, \ldots, z_n] \in R^{n\times n}$ is the representation matrix, and $\lambda_1 > 0$ is the tuned parameter.

Assume that the noise estimation is $\mathcal{S}$ then $X = XZ + \mathcal{S}$. Thus, the noise regularization term in (7) is actually $||\mathcal{S}||_{\text{F}}$. Moreover, we assume that there is sparse corrupted noise E n the data, then $X = XZ + \mathcal{S} + E$. Nonnegative least-squares regression (NLSR) considers the sparse corrupted noise E and the normal Gaussian noise $\mathcal{S}$, and its mathematical expression is

$$\begin{aligned} &\underset{Z}{\text{argmin}}\, ||\mathcal{S}|\,|_{\text{F}}^2 + \lambda_1 ||Z||_{\text{F}}^2 + \lambda_2 ||E||_1, \\ &\quad \text{s.t.}\, X = XZ + \mathcal{S} + E, \quad Z \geq 0, \end{aligned} \tag{8}$$

where $\lambda_1, \lambda_2 > 0$ are the tuned parameters.

The mathematical expression (8) can be written in further

$$\begin{aligned} &\underset{Z}{\text{argmin}}\, \frac{1}{2} ||X - XZ - E||_{\text{F}}^2 + \frac{\lambda_1}{2} ||Z||_{\text{F}}^2 + \lambda_2 ||E||_1. \\ &\text{s.t.}\, Z \geq 0. \end{aligned} \tag{9}$$

We can solve problem (9) by optimizing variables separately, which means that we can optimize a certain parameter by fixing other parameters. Using Inexact ALM [15] method, we can separate variables of the objective function by an auxiliary variable C, then problem (9) turns to

$$\begin{aligned} &\underset{Z}{\text{argmin}}\, \frac{1}{2} ||X - XZ - E||_{\text{F}}^2 + \frac{\lambda_1}{2} ||Z||_{\text{F}}^2 + \lambda_2 ||E||_1. \\ &\text{s.t.}\, C = Z, \quad C \geq 0, \end{aligned} \tag{10}$$

The Lagrange function of problem (10) is

$$\begin{aligned} \mathcal{L} = &\frac{1}{2} ||X - XZ - E||_{\text{F}}^2 + \frac{\lambda_1}{2} ||Z||_{\text{F}}^2 + \lambda_2 ||E||_1 \\ &+ < \Gamma, C - Z > + \frac{\mu}{2} (C - Z)^2, \end{aligned} \tag{11}$$

where $\Gamma \in R^{n\times n}$ is the Lagrange multiplier, and $\mu \geq 0$ is punishment parameter.

Fix other variables and solve Z by

$$\frac{\partial \mathcal{L}}{\partial Z} = -X^T (X - XZ - E) + \lambda_1 Z - \Gamma + \mu (Z - C) = 0 \tag{12}$$

then

$$Z = \left(\frac{(X^T X + \lambda_1 I)}{\mu} + I \right)^{-1} \left[\frac{X^T X - X^T E + \Gamma}{\mu} + C \right]. \tag{13}$$

Fix other variables and solve C, and we have

$$\frac{\partial \mathcal{L}}{\partial C} = \Gamma + \mu (C - Z) = 0, \quad C \geq 0, \tag{14}$$

$$C = \max \left(0, Z + \frac{\Gamma}{\mu} \right). \tag{15}$$

Fix other variables to solve E

$$E = \arg\min_E \lambda_2 ||E||_1 + \frac{1}{2} ||E - (X - XZ)||_{\text{F}}^2 = \Theta_{\lambda_2} (X - XZ), \tag{16}$$

where $\Theta_\beta(x) = \text{sign}(x)\max(|x| - \beta, 0)$ is the soft-threshold operator [16], and

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & \text{other.} \end{cases} \tag{17}$$

The whole algorithm is described in algorithm 1.

Algorithm.1 Solve problem (10) by Inexact ALM[15]

Input: Dataset X, parameters $\lambda_1 > 0,\ \lambda_2 > 0$.
Initialization:$Z = C = \Gamma = 0,\ \mu = 10^{-6}, \mu_{max} = 10^{10}, \rho = 1.1,\ \varepsilon = 10^{-6}$.
while do not converge

1. Fix other variables, update E by

$$E = argmin_E\ \lambda_2||E||_1 + \frac{1}{2}||E - (X - XZ)||_F^2 = \Theta_{\lambda_2}(X - XZ).$$

2. Fix other variables, update Z by

$$Z = (\frac{(X^TX + \lambda_1 I)}{\mu} + I)^{-1}\left[\frac{X^TX - X^TE + \Gamma}{\mu} + C\right].$$

3. Fix other variables, update C by

$$C = max\left(0, Z + \frac{\Gamma}{\mu}\right).$$

4.Update Lagrange multiplier

$$\Gamma = \Gamma + \mu(C - Z).$$

5.Update μ

$$\mu = max(\mu_{max}, \rho\mu).$$

6.Check converge condition

$$||X - XZ - E||_\infty < \varepsilon, \text{ and } ||C - Z||_\infty < \varepsilon.$$

end
Output:(Z, E, C).
Graph construction:$W = (|Z| + |Z|^T)/2$.

3 Weighted NLSR

Different categories of data should not exist in connected edges in a good graph, and wrongly connected edges are always named as 'bridge' edges. In order to further eliminate 'bridge' edges in the graph, we propose weighted NLSR by considering a weight multiplier. Weighted NLSR (WNLSR) is the problem

$$\begin{aligned} \arg_Z \min \frac{1}{2}||X - XZ - E||_{\mathrm{F}}^2 + \frac{\lambda_1'}{2}||K \cdot Z||_{\mathrm{F}}^2 + \lambda||E||_1, \\ \text{s.t.}\, Z \geq 0, \quad ||K||_{\mathrm{F}}^2 = \text{constant}, \quad K_{ii} \neq 0, \end{aligned} \tag{18}$$

where $||K||_{\mathrm{F}}^2 = \text{constant} > 0$ is to avoid the trivial solution $K = 0$. If $K_{ii} = 0$, Z will be a unit matrix which is also a trivial solution.

One can define the K by some other graphs such as k neighbor (KNN) [6] graph. In fact, the most important issue in graph constructing is eliminating 'bridge' edges. However, if the graph contains 'bridge' edges, K will also help $Z$to emerge 'bridge' edges.

K can also be regarded as a variable and can be solved automatically. Notice that constraints for K should be added to avoid the trivial solution $K = 0$. Consider the instance that the dataset is denoted as $X = [x_1, x_2, x_3, x_4] \in R^{d\times 4}$. There are two classes and $k = 2$. x_1, x_2 belong to the first class, while x_3, x_4 belong to the second class.

The ideal Z could be

$$Z = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Each element denotes an edge between a pair of points. The weight matrix K is designed to help wipe off wrong connections in Z. Concretely, K should punish two points, which do not belong to the same class, and the corresponding value in Z should be 0. Thus, the ideal K (set $K_{ii} = 1$ to avoid trivial solution) could be

$$K = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

Pivotal elements in K are displayed in green. These green elements can avoid emerging 'bridge' edges, which are the wrongly connected edges among different categories. However, the value of K does not depend on X but depends on Z. In the process of solving this problem, Z always has wrong connections. For example, Z could be

$$Z = \begin{bmatrix} 1 & 1 & 0 & \mathbf{1} \\ 1 & 1 & 0 & 0 \\ \mathbf{1} & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Wrong connections in Z are displayed in purple. When minimizing $||K \cdot Z||_{\mathrm{F}}^2$, Z will lead K be

Fig. 1 Sample images from YaleB dataset. Each *row* belongs to the same person

$$K = \begin{bmatrix} 1 & 0 & 1 & \mathbf{0} \\ 0 & 1 & 1 & 1 \\ \mathbf{0} & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

On the other hand, purple elements in K have no constraints on the wrong connections (displayed in purple) in Z. Wrong connections in Z cannot be eliminated by minimizing $||K \cdot Z||_F^2$. Thus, update K automatically cannot help improve the quality of Z.

For this problem, we start from the original problem (9). Formula (9) is a special case of (18) when $K = 1^{n\times n}$, where $1^{n\times n}$ is the $n \times n$ matrix in which all elements equal one. In this instance, i.e.,

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Next, we define K by giving the variable $p \geq 0$

$$K = \max[I, (A \geq p)], \tag{19}$$

where I is the unit matrix, A is a random matrix containing values drawn from the standard uniform distribution on the open interval (0,1). $(A \geq p)$ is a binary matrix that $(A \geq p)_{ij} = 1$ if $A_{ij} \geq p$ and $(A \geq p)_{ij} = 0$ otherwise. Notice that $K_{ii} = 1$ is required to avoid the trivial solution $Z = I$.

Edge connections among samples can be divided into right connections and wrong connections. Our goal is to construct the ideal K, which only has constraints on the wrong connections, i.e., elements in K equal to 1 when the corresponding connections are wrong connections, while elements in K equal to 0 when the corresponding connections are right connections. If $p = 0$, then $K = 1^{n\times n}$. Constraints on the right connections may reduce the ability of self-representation, while constraints on the wrong connections are always concealed due to the coefficients of the right connections are always nonzero.

If p becomes a little larger, then a few elements in K will equal zero. The constraints on some right connections are removed, and the self-representation ability can be improved. Though constraints on some wrong connections are also removed, we still have constraints on the other wrong connections, which gain more attention due to constraints on the right connections become less. Thus, a proper p will not only improve the self-representation ability of the model, but also improve the constraint ability on a certain number of wrong connections.

Define K by (19), we can solve problem (18) by optimizing variables separately. Using Inexact ALM [15] method, we can separate variables of the objective function by an auxiliary variable C, then problem (18) turns to

$$\begin{aligned} &\underset{Z}{\operatorname{argmin}} \frac{1}{2}||X - XZ - E||_{\mathrm{F}}^2 + \frac{\lambda_1'}{2}||K \cdot C||_{\mathrm{F}}^2 + \lambda_2||E_1, \\ &\quad \text{s.t. } C = Z, \quad C \geq 0. \end{aligned} \tag{20}$$

The Lagrange function of problem (20) is

$$\begin{aligned} \mathcal{L} = &\frac{1}{2}||X - XZ - E||_{\mathrm{F}}^2 + \frac{\lambda_1'}{2}||K \cdot C||_{\mathrm{F}}^2 \\ &+ \lambda_2||E||_1 + \langle \Gamma, C - Z\rangle + \frac{\mu}{2}(C - Z)^2, \end{aligned} \tag{21}$$

where $\Gamma \in R^{n\times n}$ is the Lagrange multiplier, and $\mu \geq 0$ is punishment parameter.

Fix other variables and solve Z by

$$\frac{\partial \mathcal{L}}{\partial Z} = -X^T(X - XZ - E) - \Gamma + \mu(Z - C) = 0, \tag{22}$$

then

$$Z = \left(\frac{X^TX}{\mu} + I\right)^{-1}\left[\frac{X^TX - X^TE + \Gamma}{\mu} + C\right]. \tag{23}$$

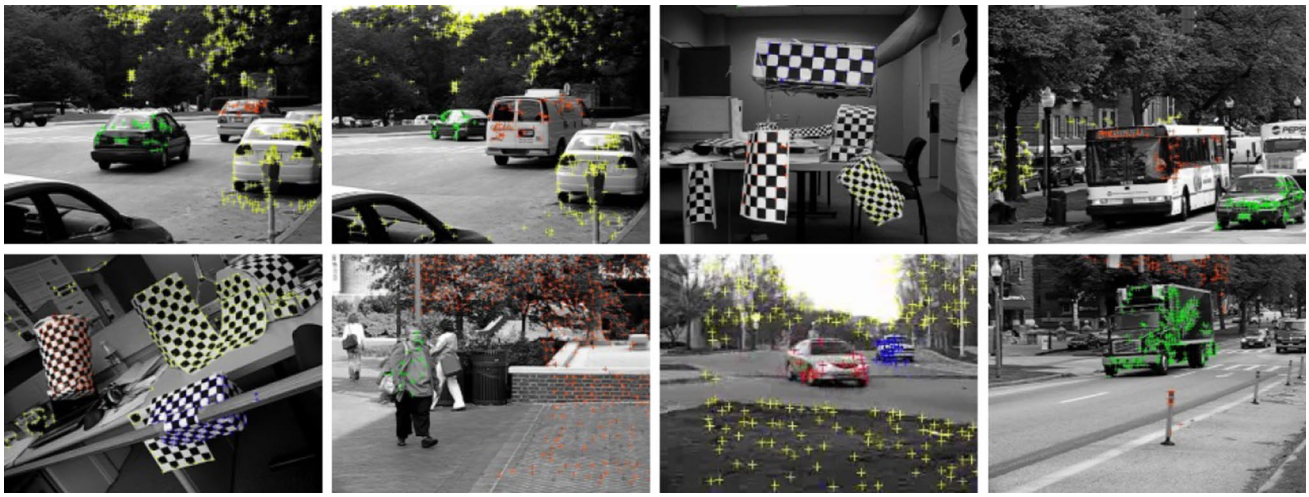


Fig. 2 Sample images from Hopkins155 datasets. Different *colors* indicate different motions

Fix other variables and solve C, and we have

$$\frac{\partial \mathcal{L}}{\partial C_{ij}} = \lambda_1' K_{ij}^2 \cdot C_{ij} + \Gamma_{ij} + \mu(C - Z)_{ij} = 0, C_{ij} \geq 0. \tag{24}$$

Then

$$C_{ij} = \frac{Z_{ij} - \frac{\Gamma_{ij}}{\mu}}{\left(\frac{\lambda_1'}{\mu} K_{ij}^2 + 1\right)}. \tag{25}$$

We can update C by

$$C = \left[1./\left(\frac{\lambda_1'}{\mu} K \cdot K + 1\right)\right] \cdot \left(Z - \frac{\Gamma}{\mu}\right), \tag{26}$$

where ./ and· are element-wise division and multiplication, respectively.

$$C = \max(0, C). \tag{27}$$

Fix other variables to solve E

$$E = \underset{\mathrm{E}}{\operatorname{argmin}} \lambda_2 ||E||_1 + \frac{1}{2}||E - (X - XZ)||_F^2 = \Theta_{\lambda_2}(X - XZ), \tag{28}$$

where $\Theta_\beta(x) = \text{sign}(x)\max(|x| - \beta, 0)$ is the soft-threshold operator [16], and

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & \text{other.} \end{cases} \tag{29}$$

The whole algorithm is described in algorithm 2.

Algorithm.2 Solve problem (20) by Inexact ALM[15]

Input: Dataset X, weight matrix K, parameters $\lambda_1' > 0$, $\lambda_2 > 0$.
Initialization: $Z = C = \Gamma = 0$, $\mu = 10^{-6}$, $\mu_{max} = 10^{10}$, $\rho = 1.1$, $\varepsilon = 10^{-6}$.
while do not converge

1. Fix other variables, update E by

$$E = argmin_E \ \lambda_2 ||E||_1 + \tfrac{1}{2}||E - (X - XZ)||_F^2 = \Theta_{\lambda_2}(X - XZ).$$

2. Fix other variables, update Z by

$$Z = \left(\frac{X^T X}{\mu} + I\right)^{-1} \left[\frac{X^T X - X^T E + \Gamma}{\mu} + C\right].$$

3. Fix other variables, update C by

$$C = \left[1./\left(\tfrac{\lambda_1'}{\mu} K \cdot K + 1\right)\right] \cdot \left(Z - \tfrac{\Gamma}{\mu}\right), \ C = max(0, C).$$

4.Update Lagrange multiplier

$$\Gamma = \Gamma + \mu(C - Z).$$

5.Update μ

$$\mu = max(\mu_{max}, \rho\mu).$$

6.Check converge condition

$$||X - XZ - E||_\infty < \varepsilon, \text{ and } ||C - Z||_\infty < \varepsilon.$$

end
Output:(Z, E, C).
Graph construction:$W = (|Z| + |Z|^T)/2$.

Table 1 Average classification error rates of different graph construction methods on ORL dataset

NL	LSR	SMR	SCC	Knn	NLSR	$WNLSR_2$ ($p = 0.75$)	$WNLSR_2$ ($p = 0.3$)	$WNLSR_1$ ($p = 0.75$)	$WNLSR_1$ ($p = 0.3$)
3	68.2	51.0	17.2	53.1	11.0	**9.85**	*9.92*	9.98	11.0
5	63.5	48.7	10.7	49.1	7.40	**6.44**	*6.63*	7.05	7.37
7	50.1	41.3	6.66	47.6	4.00	**3.71**	*3.75*	4.02	3.83
9	42.5	42.5	3.50	46.5	3.00	**2.52**	3.30	3.12	*2.90*

Bold means best performance, while italics means second best performance

Table 2 Average classification error rates of different graph construction methods on Yale dataset

NL	LSR	SMR	SCC	Knn	NLSR	$WNLSR_2$ ($p = 0.75$)	$WNLSR_2$ ($p = 0.3$)	$WNLSR_1$ ($p = 0.75$)	$WNLSR_1$ ($p = 0.3$)
3	85.7	84.5	68.7	69.6	**38.9**	39.7	39.2	39.3	*39.0*
5	86.1	86.2	72.0	65.9	**30.8**	31.8	*31.1*	*31.1*	**30.8**
7	84.3	86.5	72.0	67.6	25.6	**24.7**	*24.9*	26.0	25.9
8	84.4	86.5	71.6	66.3	**23.3**	**23.3**	*23.6*	23.9	23.7
10	87.6	88.0	73.6	73.3	23.3	**22.9**	23.5	23.9	*23.1*

Values in bold indicate the best performance

Values in italic indicate the second best performance

Table 3 Average classification error rates of different graph construction methods on Extend YaleB dataset

NL	LSR	SMR	SCC	Knn	NLSR	$WNLSR_2$ ($p = 0.75$)	$WNLSR_2$ ($p = 0.3$)	$WNLSR_1$ ($p = 0.75$)	$WNLSR_1$ ($p = 0.3$)
3	65.5	63.6	12.8	28.2	10.4	**6.26**	7.41	*7.32*	7.97
5	66.7	58.7	10.1	25.5	7.76	**4.87**	5.84	*5.59*	6.22
10	51.4	43.7	6.00	21.8	5.29	**3.83**	*4.25*	4.33	4.48
15	43.5	38.0	4.51	19.9	3.89	**2.81**	*3.20*	3.26	3.37
20	48.8	37.8	4.45	19.0	3.65	**2.85**	*3.15*	3.21	3.35
30	37.6	28.2	2.67	16.8	2.38	**1.95**	2.13	*2.00*	2.24
50	30.2	24.5	2.07	15.5	1.57	**1.28**	*1.41*	1.47	1.69

Values in bold indicate the best performance

Values in italic indicate the second best performance

Table 4 Average classification error rates of different graph construction methods on Hopkins155 datasets (100 datasets)

NL	LSR	SMR	SCC	Knn	NLSR	$WNLSR_2$ ($p = 0.75$)	$WNLSR_2$ ($p = 0.3$)	$WNLSR_1$ ($p = 0.75$)	$WNLSR_1$ ($p = 0.3$)
3	52.0	53.6	31.6	8.57	4.29	4.35	**4.16**	4.45	*4.19*
5	52.4	54.3	27.9	6.32	3.37	3.51	*3.04*	3.58	**3.01**
8	52.0	54.6	24.0	4.73	2.78	**2.74**	2.77	*2.76*	2.77
10	52.3	53.2	22.2	4.44	2.48	2.53	*2.44*	2.55	**2.43**
12	53.2	56.1	21.7	4.21	*2.55*	*2.55*	**2.46**	2.57	**2.46**
14	53.6	56.1	20.6	3.79	*2.29*	*2.29*	**2.26**	*2.29*	**2.26**

Values in bold indicate the best performance

Values in italic indicate the second best performance

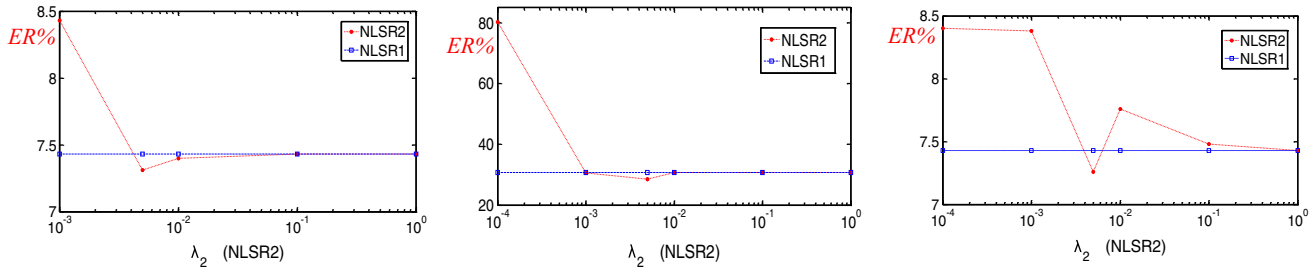
Table 5 A finite grid of parameter values

Parameter	Value
λ_1	0.0001, 0.001, 0.005, **0.01**, 0.05, 0.1, 1, 10
λ_2	0.0001, 0.001, 0.005, **0.01**, 0.05, 0.1, 1, 10

We find the best performance of our algorithm on this finite grid. The bold fonts indicate the adopted parameter values

4 Experiments

In this section, the semi-supervised learning task is used to evaluate the performance of the proposed NLSR method. We measure different graph construction algorithms by the classification error of the unlabeled data. Datasets include face dataset and motion dataset (as seen in Figs. 1, 2).



1:ORL dataset; 2:Yale dataset; 3: Extended YaleB dataset.

Fig. 3 Analysis of parameter λ_2 in NLSR. Experiments are the classification error rates (ER %) on *1* ORL; *2* Yale and *3* extend YaleB datasets when NL = 5. NLSR2, fixing $\lambda_1 = 0.01$ and varying λ_2. NLSR1: $\lambda_1 = 0.01, \lambda_2 = +\infty$

4.1 Datasets

ORL dataset The ORL dataset[1] consists of 10 different images for each of 40 distinct subjects, which are taken at different times, under different lighting condition, with different facial expression and with/without glasses. Each image is 32 × 32 pixels with 256 gray levels per pixel. The first ten people are selected for constructing the data matrix for experiment.

Yale dataset The Yale dataset[2] contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration. Each image is also represented by a 1024-dimensional vector in image space.

Extended YaleB dataset [17] This database is challenging due to large noise and corruptions. It contains 2414 frontal face images of 38 subjects. Each subject has 64 face images. We choose the cropped images of first five individuals, and resize them to 32 × 32 pixels. The data are projected into a 10 × 6-dimensional subspace by PCA.

Hopkins155 motion dataset [18] This dataset is a motion segmentation dataset, consisting of 156 video sequences with extracted feature points and their tracks across frames. It contains board sequences, traffic sequences and pedestrian movement sequences. The first 100 two-motion video sequences are selected for constructing the data matrix for experiment. We use PCA to project the data into a 12-dimensional subspace.

4.2 Semi-supervised image classification

To demonstrate how the classification performance can be improved by our method, we compare the proposed algorithm with four algorithms: KNN graph [6], SSC graph [9], LSR graph [13] and SMR graph [14]. The parameters are set according to corresponding references and the best parameters are determined by the finite grid [19]. We do not consider the LRR graph as the compared algorithm because it always performs poorly in the experiments.

The parameter λ_1' in WNLSR is always set as $\lambda_1' = \lambda_1$, which is denoted as $WNLSR_1$. In addition, we can set $\lambda_1' = \lambda_1(1 - p + p/n)$ by considering the number of elements in $K \cdot Z$, which is denoted as $WNLSR_2$. Besides, λ_1 is set as 0.01 throughout the whole paper.

We choose the famous local and global consistency (LGC) [5] method as the semi-supervised learning method to compare the performance of different graph construction methods. Assume that dataset $X = [x_1,x_2,\ldots,x_n] \in R^{d\times n}$, and the first l samples $x_1,x_2,\ldots, x_l$ are labeled. The label L comes from k categories, $L = [1,2,\ldots,k]$. $Y = [Y_l Y_u]^T \in R^{n\times k}$ is the label matrix. If sample x_i is labeled by $j, j \in \{1, 2 \ldots, k\}$, then $Y_{ij} = 1$, otherwise $Y_{ij} = 0$. The optimization function of LGC is

$$\underset{F}{\operatorname{argmin}} \operatorname{tr}\{ F^T \tilde{L}_W F + \beta(F - Y)^T (F - Y)\} , \tag{30}$$

where $\beta \in [0, +\infty)$ balances the local adaptation term and the overall smooth term of the objective function. Generally, we set $\beta = 0.99$. $F = [F_l F_u]^T \in R^{n\times k}$ is the desired classification function while $\tilde{L}_W$ is the standard Laplacian graph of W and $\tilde{L}_W = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$.

Datasets include three face datasets (ORL, Yale, and YaleB) and 100 small datasets in Hopkins155. The face datasets need to be normalized first ($x_i = x_i/\|x_i\|_2$, $i = 1, 2, \ldots n$). There are 103 datasets used in the semi-supervised learning experiments in total. For each data set, the evaluations are conducted with different labeled samples *NL*. For the fixed labeled samples *NL*, we run the experiments as follows:

1. Construct graphs by different methods.
2. Randomly choose *NL* points as labeled data from the data set as the collection for experiment.
3. Apply the LGC method to learn the label propagation.
4. Calculate the classification error on all unlabeled data.
5. Repeat the above process for 20 times.

[1] http://www.uk.research.att.com/facedatabase.html.

[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

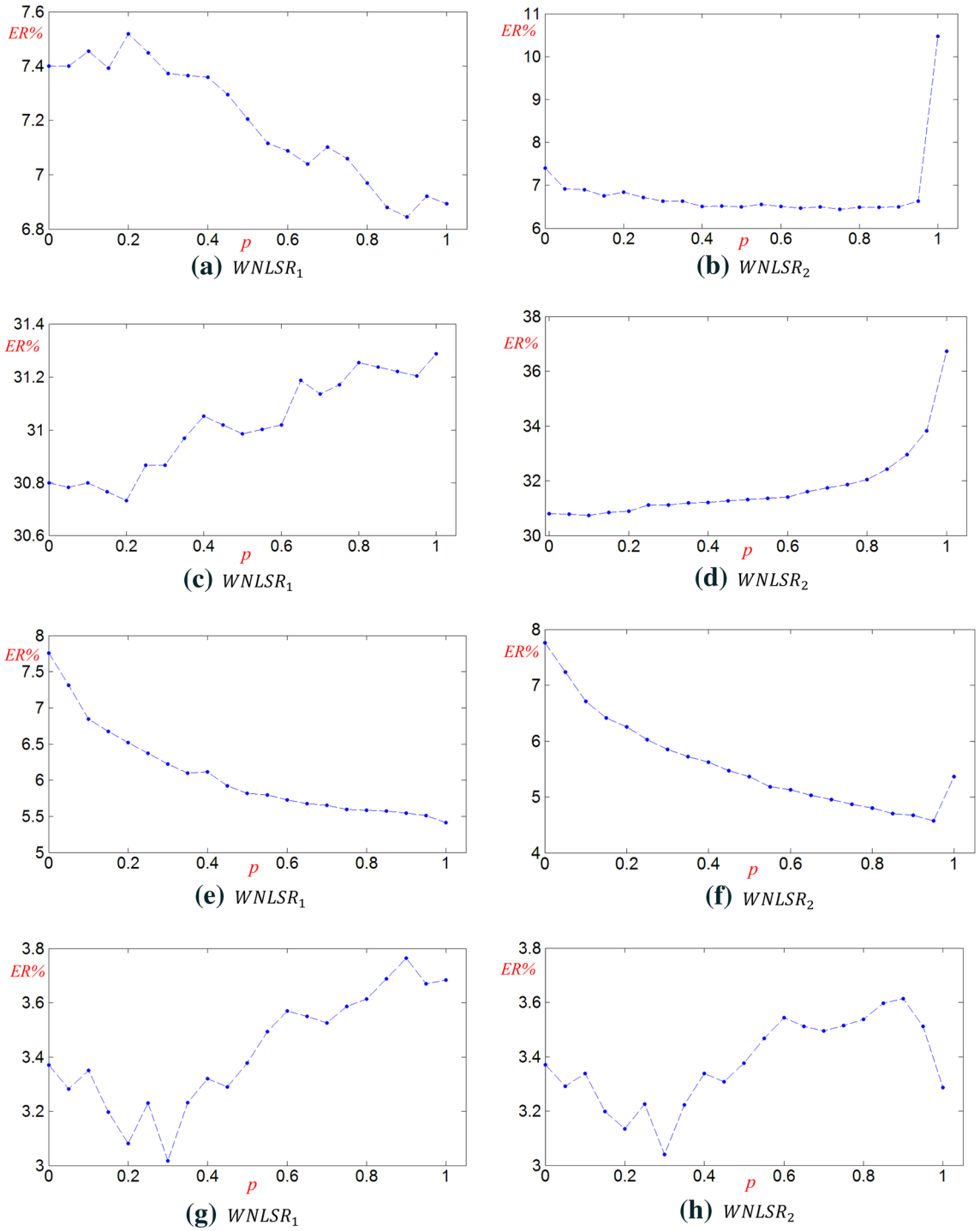


Fig. 4 Error rates vs p (WNLSR) on different datasets with NL = 5. y-axis denotes the error rate (ER %) of the method, x-axis denotes the variable p. **a**, **b** ORL dataset. **c**, **d** Yale dataset. **e**, **f** Extend YaleB dataset. **g**, **h** Hopkins155 datasets

4.3 Experimental results

Experimental results are shown in Tables 1, 2, 3, and 4. From these results, we can observe that: (1) in most cases, NLSR consistently achieves good performance compared with LSR while WNLSR performs good. (2) SSC also performs well on ORL, Yale and extended YaleB face datasets. (3) Though LSR and SMR always have good performance on unsupervised learning experiments [13, 14], they perform poorly on 100 small datasets of Hopkins155.

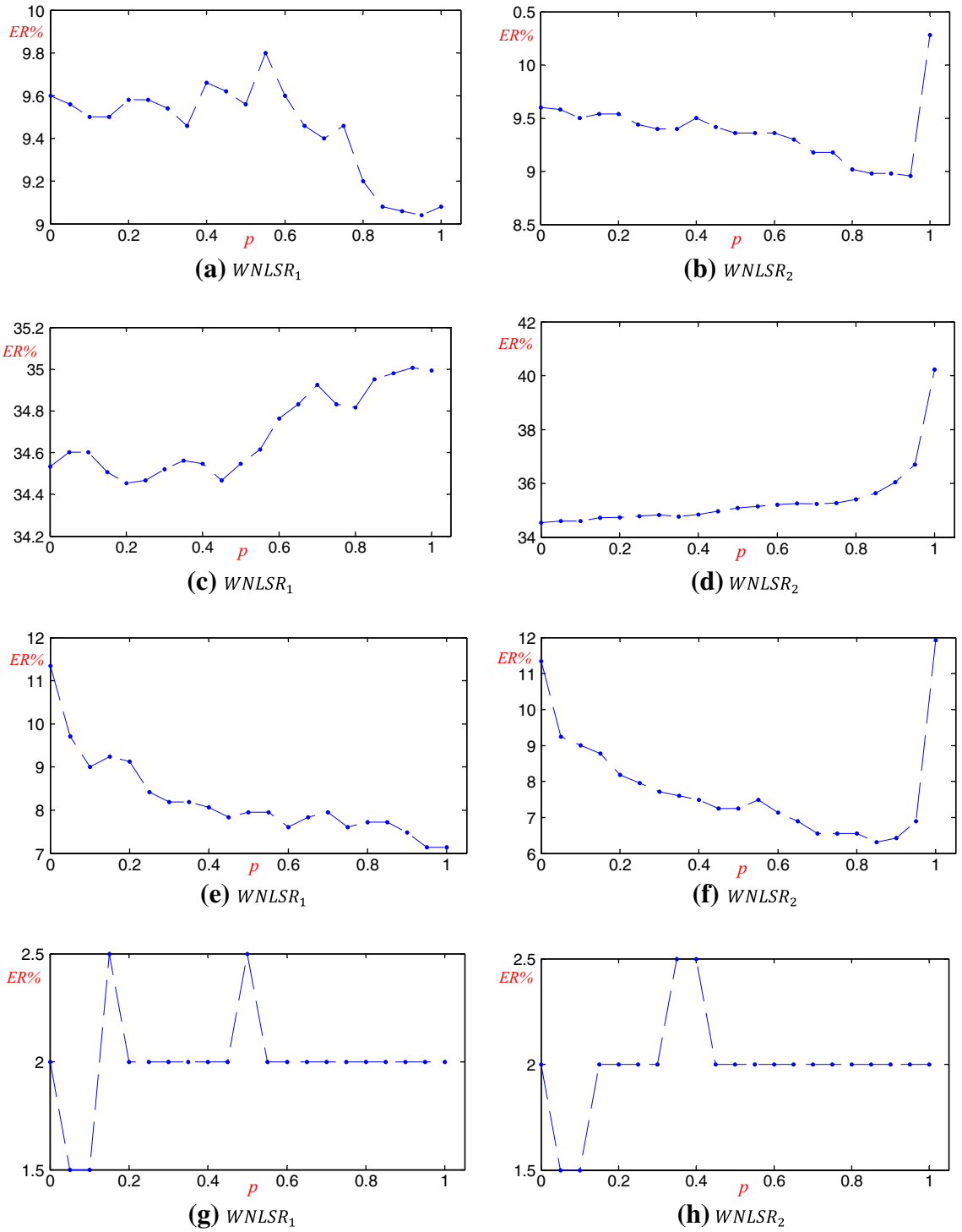


Fig. 5 Average predicted error rates of testing samples vs p (WNLSR) on different datasets with NL = 5. y-axis denotes the error rate (ER %) of testing samples, x-axis denotes the variable p. **a**, **b** ORL dataset. **c**, **d** Yale dataset. **e**, **f** Extend YaleB dataset. **g**, **h** Hopkins155 datasets

Meanwhile, KNN and NLSR graph perform well on these datasets. The results show that NLSR achieved good performance based on the nonnegative constraint and the design of the error estimation.

4.4 Algorithm analysis

Parameter selection is important for algorithms. We use the finite grid [19] method to select parameters for

Table 6 The mean sparseness of the representation coefficient Z obtained by different methods on different datasets

	LSR	SMR	SSC	Knn	NLSR
Hopkins155	0.30	0.31	*0.85*	**0.89**	0.81
Yale	0.29	0.51	0.82	*0.85*	**0.98**
YaleB	0.42	0.45	**0.91**	*0.90*	0.89
ORL	0.38	0.50	*0.90*	0.82	**0.92**

The sparsity value tends to one when Z becomes more sparse. Z of Knn is the original Knn graph before symmetrization

Values in bold indicate the best performance

Values in italic indicate the second best performance

NLSR, as seen in Table 5. Notice that parameters λ_1, λ_2 control the nonnegative constraint term and the error design term in NLSR problem [as seen in (8) and (7)]. When $\lambda_2 = +\infty$, we set $E = 0$, then formula (8) turns to formula (7), i.e., only the nonnegative constraint is considered.

Figure 3 shows the influence of parameter λ_2 when λ_1 is fixed. We can find that a suitable λ_2 will help reduce the classification error rate, but the reduction extent is limited. Notice that when λ_2 is small, the performance of the model will be affected. It suggests that the key role in NLSR model is the nonnegative constraint, and it also shows that the error estimation is more difficult.

In our experiments, we simply set $p = 0.3$ and $p = 0.75$ for WNLSR. Figure 4 shows the average performance (100 running times) of various p on different datasets when NL = 5. For simplicity, we set $\lambda_1 = 0.01$ and $\lambda_2 = +\infty$

Since the proper p is different in different datasets, it is important to find a way to estimate it. We can estimate p as follows:

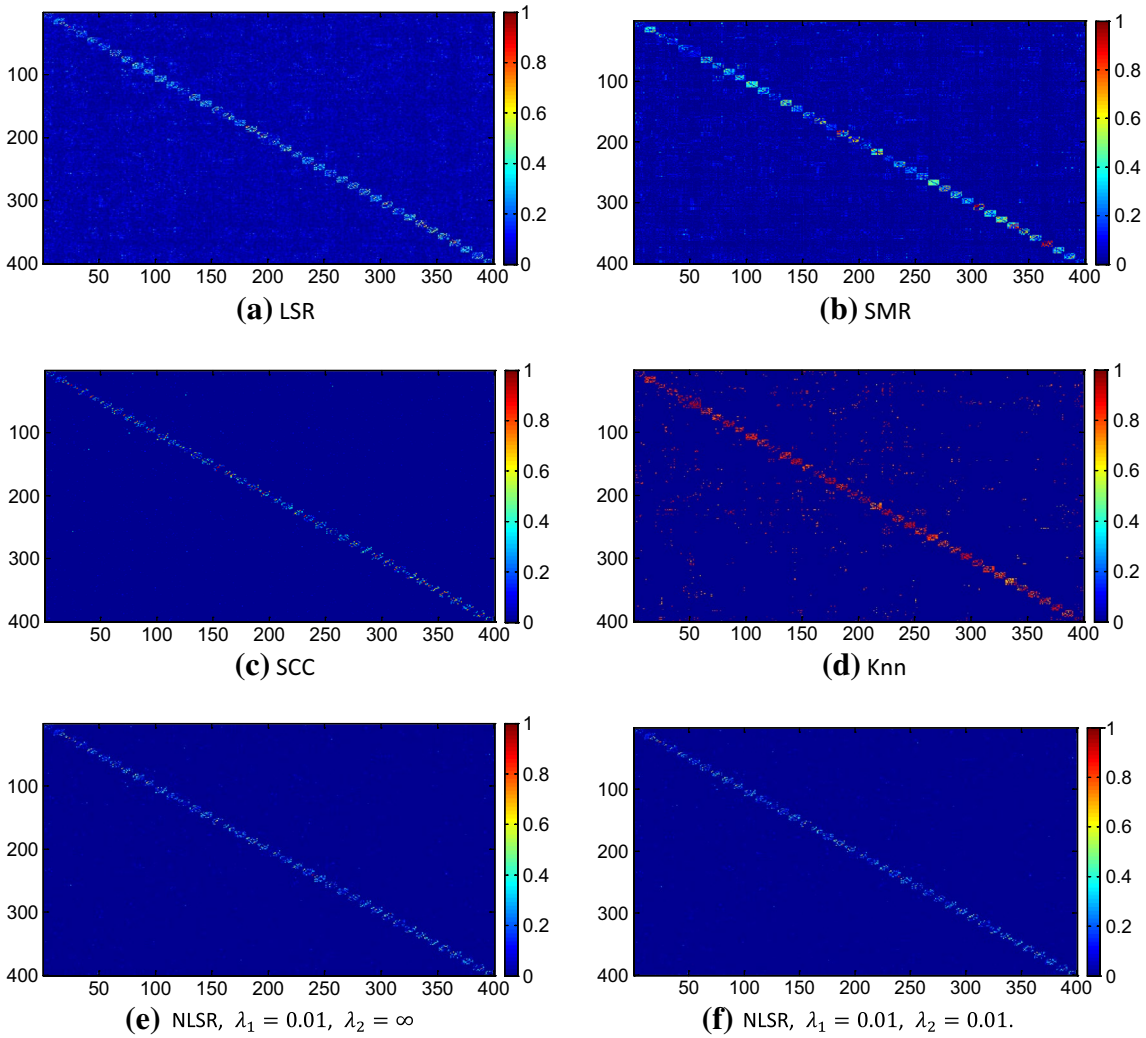


Fig. 6 Graphs by different graph construction methods on ORL dataset. Values of graph are normalized and they varies from [0,1], the diagonal of the graph is also set to 0 to display

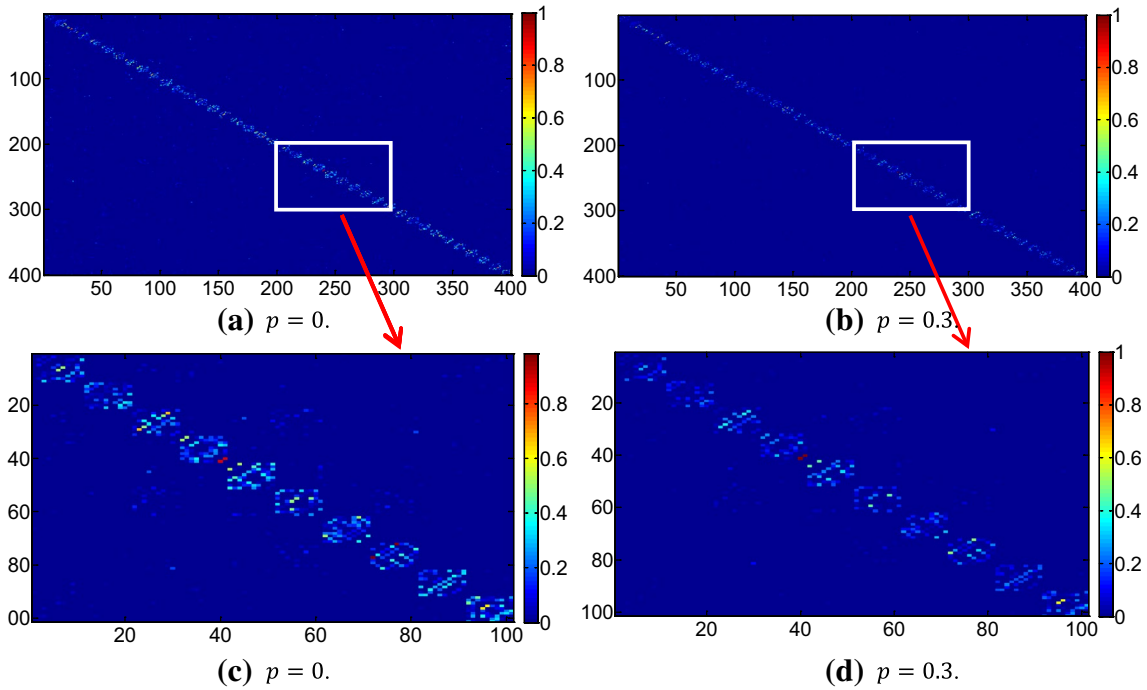


Fig. 7 Graphs by $WNLSR_1$ with different p on the ORL dataset. If $p = 0$, $WNLSR_1$ becomes NLSR method. Values of graph are normalized and they varies from [0,1], the diagonal of the graph is also set to 0 to display

Table 7 The average running time of different graph construction methods

	ORL (s)	Yale (s)	YaleB (s)	Hopkins155 (s)
LSR	7.0×10^{-4}	2.3×10^{-3}	8.6×10^{-3}	8.7×10^{-3}
SMR	9.8×10^{-3}	3.4×10^{-2}	1.2×10^{-1}	1.1×10^{-1}
Knn	2.0×10^{-3}	5.5×10^{-3}	3.9×10^{-3}	3.8×10^{-3}
SCC	8.7×10^{-1}	2.7	1.4	3.9
NLSR	3.7×10^{-1}	1.1	6.2	15
Weight NLSR	3.8×10^{-1}	1.2	6.3	30

1. Give a semi-supervised problem.
2. Randomly select one labeled sample in each class as the testing samples, and regard them as unlabeled samples in graph learning.
3. Give a random matrix and compute the predict error of testing samples with different p.
4. Repeat the whole procedure 100 times for the ORL dataset, the Yale dataset and the YaleB dataset, while compute the average predict error of 100 Hopkins155 datasets.

Figure 5 shows the average predicted error rates (100 running times) of testing samples with various p on different datasets when NL = 5. We can observe that the estimated proper p in Fig. 5 is roughly consistent with the proper p in Fig. 4 on different datasets.

Now we analyze the sparsity of the representation coefficient matrix Z. The sparsity of a matrix Z can be defined as follows:

$$\text{sparsity}(Z) = \frac{\sum_{i=1}^{n} \text{sparsity}(Z_i)}{n}. \tag{31}$$

The sparsity(Z_i) of the vector Z_i can be calculated as [20]

$$\text{sparsity}(Z_i) = \frac{\sqrt{n} - \sum |Z_{ij}| \Big/ \sqrt{\sum Z_{ij}^2}}{\sqrt{n} - 1}, \tag{32}$$

where Z_{ij} is the jth element of Z_i.

The sparsest possible vector will have a sparseness of one, whereas a vector with all elements equal should have a sparseness of zero. As seen in Table 6, the representation coefficient matrices Z of KNN and SSC have high sparsity, while the sparsity of representation coefficient matrices of LSR and SMR have low sparsity. NLSR can obtain more sparse representation coefficient matrix than LSR.

The sparsity of the representation coefficient matrix Z will directly lead to the sparsity of the corresponding weight matrix W. By observing experimental results and the sparsity of graphs, we can find that the sparsity of the graph has an important influence on semi-supervised learning. In fact, the most important factor of the good performance of NLSR is that the nonnegative constraint

can help improve the sparsity of the LSR graph. The sparsity, of course, is not the only factor in determining the classification performance. For example, KNN and SCC graphs have the highest sparsity on the Hopkins155 and the YaleB datasets, but their classification performances are not so good as NLSR. Figure 6 shows graphs obtained by different graph construction methods on the ORL dataset (40 categories). We can find that NSLR is more "clean" than LSR graph, which means that NLSR has less incorrect edges between points than LSR. Notice that SCC graph also has not too many incorrect edges between points. However, the aggregation degree of each category of SCC graph is lower than LSR graph or NLSR graph. Figure 7 shows the difference between NLSR ($p = 0$) and WNLSR ($p = 0.3$). We magnify partial area of the graph obtained by WNLSR. We can find out that WNLSR can eliminate more 'bridge' edges than NLSR.

In Table 7, we report the average running time of different graph construction methods. The LSR graph and the Knn graph usually have short running time. The SCC graph, the NLSR graph and the WNLSR graph, which always perform well, usually have long running time. The NLSR graph and the WNLSR graph run faster than the SCC graph on ORL, Yale and YaleB datasets, and run slower than the SCC graph on Hopkins155 datasets. The computer configuration is Intel(R) Core(TM) i5-3470 CPU @ 3.20 GHz, 4G of memory, Microsoft Windows7 system, and Matlab 2010b software.

The major computational burden of NLSR and WNLSR (both are iterative algorithms) lie in the computation of the inverse of matrix (formula 13), with a computational complexity of $O(n^3)$.

5 Conclusion and future work

Inspired by the idea of the self-representation of data, we propose a novel graph construction method named as NLSR graph based on the LSR graph. The biggest advantage of LSR is that it can directly adopt the self-representation idea. However, its poor ability to eliminate the wrong edges among samples limits its application. Based on LSR, we emphasize the nonnegative constraint on Z without increasing other regularizations on Z. In this way, NLSR not only maintains the inherent advantage of LSR, but also improves its ability to eliminate the bridge edges. In fact, the nonnegative constraint can avoid the offset of positive and negative coefficients, and improve the sparsity of the constructed graph. Weighted version of NLSR (WNLSR) is also proposed to further eliminate 'bridge' edges by constructing the proper weighted multiplier. At last, we redesign the noise estimation by taking both the small Gaussian noise and the sparse corrupted noise into consideration at the same time. Image classification experiments have showed encouraging results of the NLSR algorithm when compared with the state-of-the-art algorithms in semi-supervised learning, especially in improving LSR method.

The construction of the graph is the key issue in graph-based semi-supervised learning. Thus, constructing a good graph is a meaningful task. In constructing a graph, we can focus on the following issue: a pair of samples are close in distance space but belong to different classes. The edges between those sample pairs are usually 'bridge' edges. We can study how to pull those samples far away from each other by considering Mahalanobis space [21–23]. It is also a meaningful task to develop more efficient algorithms for NLSR and WNLSR in future work.

References

1. Zhu, X.: Semi-supervised learning literature survey[J]. Technical report 1530, Department of Computer Science, University of Wisconsin-Madison, Madison (2005)
2. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised learning[M]. MIT Press, Cambridge (2006)
3. Belkin, M., Niyogi, P., Sindhwani V.: On manifold regularization[C]. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005), pp. 17–24 (2005)
4. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions[C]. ICML **3**, 912–919 (2003)
5. Zhou, D., Bousquet, O., Lal, T.N., et al.: Learning with local and global consistency[J]. Adv. Neural Inf. Process. Sys. **16**(16), 321–328 (2004)
6. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction[J]. Science **290**(5500), 2319–2323 (2000)
7. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation[J]. Neural Comput. **15**(6), 1373–1396 (2003)
8. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding[J]. Science **290**(5500), 2323–2326 (2000)
9. Elhamifar, E., Vidal, R.: Sparse subspace clustering[C]. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009 (CVPR 2009), pp. 2790–2797 (2009)
10. Lin, Z., Liu, R., Su, Z.: Linearized alternating direction method with adaptive penalty for low-rank representation[C]. Adv. Neural Inf. Process. Sys. 612–620 (2011)
11. Liu, G., Lin, Z., Yan, S., et al.: Robust recovery of subspace structures by low-rank representation[J]. IEEE Trans. Pattern. Anal. Mach. Intell. **35**(1), 171–184 (2013)
12. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation[C]. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 663–670 (2010)
13. Lu, C.Y., Min, H., Zhao, Z.Q., et al.: Robust and efficient subspace segmentation via least squares regression[M]. In: Computer Vision–ECCV 2012, pp. 347–360. Springer, Berlin (2012)
14. Hu, H., Lin, Z., Feng, J., et al.: Smooth representation clustering[C]. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3834–3841 (2014)

15. Lin, Z., Chen, M., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices[J]. arXiv preprint arXiv:1009.5055 (2010)
16. Candès, E.J., Li, X., Ma, Y., et al.: Robust principal component analysis[J]. J. ACM (JACM) **58**(3), 11 (2011)
17. Lee, K.C., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting[J]. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 684–698 (2005)
18. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms[C]. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007 (CVPR'07), pp. 1–8 (2007)
19. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation[C]. In: Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics, pp. 57–64 (2005)
20. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints[J]. J. Mach. Learn. Res. **5**, 1457–1469 (2004)
21. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning[C]. In: Proceedings of IEEE International Conference on Machine Learning (2007)
22. Davis, J.V., Kulis, B., Jain, P., et al.: Information-theoretic metric learning[C]. In: Proceedings of the 24th International Conference on Machine Learning. ACM, pp. 209–216 (2007)
23. Kostinger, M., Hirzer, M., Wohlhart, P., Roth, P.M., Bischof, H.: Large scale metric learning from equivalence constraints[C]. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2288–2295, June 2012