Multimed Tools Appl (2017) 76:8859–8880
DOI 10.1007/s11042-016-3510-3

Robust structured sparse representation via half-quadratic optimization for face recognition

Yong Peng [1,2] · Bao-Liang Lu [3]

Received: 26 November 2015 / Revised: 27 March 2016 / Accepted: 1 April 2016 /
Published online: 13 April 2016

Abstract By representing a test sample with a linear combination of training samples, sparse representation-based classification (SRC) has shown promising performance in many applications such as computer vision and signal processing. However, there are several shortcomings in SRC such as 1) the l_2-norm employed by SRC to measure the reconstruction fidelity is noise sensitive and 2) the l_1-norm induced sparsity does not consider the correlation among the training samples. Furthermore, in real applications, face images with similar variations, such as illumination or expression, often have higher correlation than those from the same subject. Therefore, we correspondingly propose to improve the performance of SRC from two aspects by: 1) replacing the noise-sensitive l_2-norm with an M-estimator to enhance its robustness and 2) emphasizing the sparsity in terms of the number of classes instead of the number of training samples, which leads to the structured sparsity. The formulated robust structured sparse representation (RGSR) model can be efficiently optimized via alternating minimization method under the half-quadratic (HQ) optimization framework. Extensive experiments on representative face data sets show that RGSR can achieve competitive performance in face recognition and outperforms several state-of-the-art methods in dealing with various types of noise such as corruption, occlusion and disguise.

Keywords Sparse representation · Structured sparsity · Robustness · Half-quadratic optimization · Face recognition

✉ Yong Peng
stany.peng@gmail.com

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

[2] Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou 310018, China

[3] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

1 Introduction

Sparse representation [6, 27] is an efficient statistical signal modeling tool which has become a promising model in many machine learning and computer vision applications [8, 19, 31, 35]. When applied to image clustering or classification, sparse representation codes an image using a small number of atoms parsimoniously chosen out of an over-complete dictionary. The original definition of sparsity is based on the l_0-norm, which directly counts the number of non-zero elements in a vector. As the closest convex surrogate, the l_1-norm is widely used as an alternative to measure the sparsity of the representation coefficient, which makes the optimization much more efficient. Yang et al. [31, 35] reviewed several fast approaches to the optimization of these l_1-norm minimization based sparse representation models.

Recently, many studies [1, 37] have shown that the l_1-norm induced sparse models perform well in low-correlation settings. However, if samples from the same class or manifold are highly correlated, the l_1-norm minimization will encounter the stability problems [12]. Generally, it tends to randomly select a single representative data point and ignore other correlated points. This leads to a sparse solution but misses the correlated information in data, which often causes sub-optimal performance. Specifically, for face recognition task in uncontrolled environment, the variation information such as illumination and expression may be more significant than the identity [18]. In this case, it is possible that face images from different subjects with similar variations could have higher correlation than those from the same subject but with different variations. Therefore, it is of great necessity to consider the label information of training samples and emphasize the sparsity of the number of classes instead of the number of training samples, which leads the structured sparsity.

Moreover, for most real-world data sets, they are usually noisy or grossly corrupted. The original sparse representation and most of its variants use the sum of squared error or the l_2-norm error function to measure the quality of signal reconstruction, which implicitly assumes that the noise follows the Gaussian distribution. However, it is not the case for real world problems which do not conform to the assumptions made by the model. The least-squares error is sensitive to outliers, which will greatly degrade the quality of approximation if there is a single corrupted point. Therefore, it is necessary to replace the quadratic form of residuals by lowering down the weight of noisy or corrupted region of samples. Instead of minimizing the non-quadratic and possibly non-convex loss function, we propose to use the M-estimator technique [17], which can be optimized by HQ minimization. The HQ optimization [25] is a unified framework for both error correction and detection [16]. By utilizing robust M-estimators under HQ, the robustness of certain models can be greatly improved. The maximum correntropy criterion [21], which is essentially the Welsch M-estimator, has been widely used for enhancing the robustness of sparse representation [14, 15], low-rank matrix recovery [14, 15], NMF [7] and least square [22]. Other M-estimators, such as l_1-l_2 [14, 15], Huber [17] and "Fair" [16], can also be used in such settings.

By conducting extensive experiments on representative face data sets, the results show that RGSR achieves competitive performance in face recognition. RGSR outperforms several state-of-the-art methods in dealing with various types of noise such as corruption, occlusion and disguise.

The remainder of this paper is organized as follows. In Section 2, we give a brief overview of SRC, M-estimator and the HQ minimization. The proposed RGSR model will be presented in Section 3. In Section 4, we conduct experiments to evaluate the effectiveness of RGSR. In section 5, we give a brief discussion on the connection between the proposed method and He's work [16]. Section 6 concludes the paper.

2 Related work

We summarize the notations and definition of norms used in this paper. Matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. Given a matrix $\mathbf{M}$, m_{ij} is its element in the i-th row and j-th column. The l_1-, l_2-norm of a vector $\mathbf{v}$ are defined as $\|\mathbf{v}\|_1 = \sum_i |v_i|$ and $\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}^T\mathbf{v}}$ respectively. $\|\mathbf{M}\|_1 = \sum_{ij} |m_{ij}|$, $\|\mathbf{M}\|_2^2 = \sum_{ij} m_{ij}^2$.

2.1 Sparse representation-based classification

SRC method was proposed in [30] for application on face recognition. Generally, the dictionary matrix $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_c]$ is formed by stacking the training samples together, where $\mathbf{A}_i$ is the subset of training samples from class i and c is the number of classes. For each test sample $\mathbf{y}$, the sparse representation coefficient $\boldsymbol{\alpha}$ can be obtained via optimizing the following l_1-norm regularized minimization problem

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1, \tag{1}$$

where λ is a trade-off parameter to control the sparsity of $\boldsymbol{\alpha}$; then the classification is made by

$$\text{identity}(\mathbf{y}) = \arg\min_i \{error_i\}, \tag{2}$$

where $error_i = \|\mathbf{y} - \mathbf{A}_i \hat{\boldsymbol{\alpha}}_i\|_2$, $\hat{\boldsymbol{\alpha}} = [\hat{\boldsymbol{\alpha}}_1; \hat{\boldsymbol{\alpha}}_2; \cdots; \hat{\boldsymbol{\alpha}}_c]$ and $\hat{\boldsymbol{\alpha}}_i$ $(i = 1, 2, \cdots, c)$ is the coefficient vector associated with the i-th class. It was claimed in [30] that the success of SRC is mainly caused by the l_1-norm sparsity imposed on the representation efficient. However, this l_1-norm induced sparsity treats each element in $\boldsymbol{\alpha}$ equally, which does not consider the correlation of columns in dictionary $\mathbf{A}$. Therefore, it performs well only when $\mathbf{A}$ is under low-correlation settings. In this paper, we use training data $\mathbf{X} \in \mathbb{R}^{d \times n}$ (d, n respectively denote the dimensionality and number of training samples) as dictionary other than some existing studies which use learned dictionary [26, 33].

It is usually to introduce an identity matrix $\mathbf{I}$ as a dictionary to code the outlier pixels (e.g., corrupted or occluded pixels), which lead to the following unconstrained Lagrangian function.

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \|\mathbf{y} - [\mathbf{A}, \mathbf{I}] \cdot [\boldsymbol{\alpha}; \boldsymbol{\beta}]\|_2^2 + \lambda \|[\boldsymbol{\alpha}, \boldsymbol{\beta}]\|_1. \tag{3}$$

It assumes that the error $\boldsymbol{\beta}$ has a sparse representation. Such type of SRC method shows high robustness to face occlusion and corruption.

2.2 M-estimators

M-estimator [17] is a class of popular robust learning technique in statistics, which are generalized maximum likelihood estimation to the minimization of the sums of functions of the data. M-estimators have been widely used in machine learning and computer vision fields for improving the models' robustness. In robust regression, iteratively reweighted least squares (IRLS) is often used to solve M-estimators. Another common used technique is the half-quadratic optimization [25]. By the multiplicative or additive half-quadratic reformulation of M-estimator, the original problem can be solved by alternately minimizing an augmented objective function.

There are some popular M-estimators such as l_1-l_2 function, Fair function, log-cosh function, Welsch function, Huber function and $l_1$function. Figure 1 shows these loss functions and their corresponding weight functions in multiplicative half-quadratic form, which provides a better understanding to their properties.

Generally, M-estimators have the following properties: 1) all loss functions are less increasing and give less punishment to large fitting errors and all weight functions (except l_1) are upper bounded by 1 for small error and lower bound by 0 for large error; 2) the l_1 norm is often used to pursue robustness, but the corresponding weight $1/|e|$ is not upper bounded, so the objective function would be dominated by the data points with near-zero fitting errors which leads to the singularity problem [5]; 3) Welsch function, behaves like the l_2 norm on small errors, like the l_1 norm on relative larger errors, and approaching l_0 norm with further increasing of errors; 4) Huber function behaves like the l_2 norm on small errors and like l_1 norm on large errors, controlled by cutoff parameters.

2.3 The half-quadratic minimization

This section briefly reviews the background of half-quadratic modeling based on conjugate function theory [9, 10] for convex or non-convex minimization. The detailed introduction can be found in [16].

Given a differentiable function $f(\mathbf{v})$: $S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, the conjugate function $f^*(\mathbf{p})$: $\mathbb{R}^n \rightarrow \mathbb{R}$ of the function $f(\cdot)$ is defined as [3]

$$f^*(\mathbf{p}) = \inf_{\mathbf{v}\in S} \mathbf{p}^T\mathbf{v} - f(\mathbf{v}). \tag{4}$$

The domain of $f^*(\mathbf{p})$ is bounded above on S [3]. $f^*(\mathbf{p})$ is the pointwise supremum of a family of convex functions of $\mathbf{p}$, which is also a convex function. Based on conjugate function theory, a loss function in image restoration and signal recovery can be defined as [2, 4, 25]

$$f(\mathbf{v}) = \min_{\mathbf{p}} \{\psi(\mathbf{v},\mathbf{p}) + \varphi(\mathbf{p})\}, \tag{5}$$

where $f(\cdot)$ is a potential loss function such as a certain M-estimator, $\mathbf{v}$ is a set of adjustable parameters of a linear system, $\mathbf{p}$ is an auxiliary variable in HQ optimization, $\psi(\mathbf{v}, \mathbf{p})$ is a quadratic function, and $\varphi(\cdot)$ is the dual potential function of $f(\cdot)$.

For face recognition application, we use the multiplicative form quadratic function of $\psi(\mathbf{v}, \mathbf{p})$ as $\psi(\mathbf{v},\mathbf{p}) \doteq \sum_i p_i v_i^2$, where v_i is the coding residual for each pixel and p_i is the learned weight

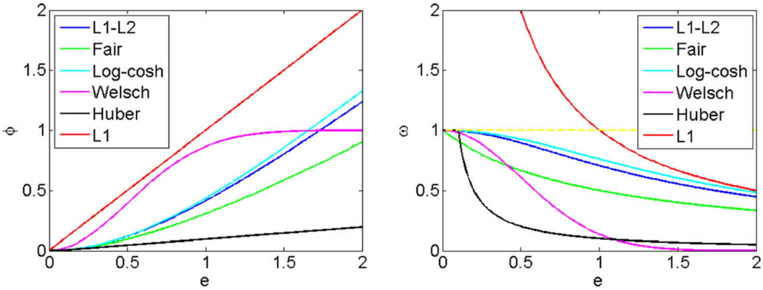


Fig. 1 Some popular loss functions (left) and the corresponding weight functions (right)

for such pixel. p_i will be a small value to alleviate its influence if this pixel is corrupted. Therefore, the learned $\mathbf{p}$ can adjust the influence of each pixel according to its corruption level.

3 Robust structured sparse representation

3.1 Robustness improvement

Specifically, using structured sparse representation with respect to a test sample $\mathbf{y}$, we have

$$\phi(\mathbf{e}) = \min_{\mathbf{w}} \{\psi(\mathbf{e}, \mathbf{w}) + \varphi(\mathbf{w})\}, \tag{6}$$

where $\mathbf{e} \triangleq \mathbf{X}\boldsymbol{\alpha} - \mathbf{y} \in \mathbb{R}^d$ is the coding residual and $\mathbf{w} \in \mathbb{R}^d$ is the corresponding pixel-level weight for face image. In this paper we only consider to use the multiplicative form of ψ as $\psi(\mathbf{e}, \mathbf{w}) = \sum_{i=1}^{d} w_i e_i^2$, which plays the role as error detection [16].

The first term in (1) which uses the l_2-norm to measure the coding residual can be easily dominated by a few outliers with large errors. This can be illustrated by Fig. 2, where the l_2-norm induces more penalty for large fitting errors than the l_1-norm and Logistic loss function (one type of M-estimator we will use in this paper). Accordingly, the l_2-norm loss function uses a constant weight for both small and large errors, that is, it actually does not consider whether the pixel is corrupted or not. However, M-estimator can learn the weight $\mathbf{w}$ to adapt the corruption level, which can greatly alleviate the influence of outliers. In general, M-estimator uses small weight w_i for large e_i to make learning models robust to outliers.

Therefore, by replacing the l_2-norm with M-estimator $\phi(\cdot)$, we can obtain the following objective

$$\min_{\boldsymbol{\alpha}} \phi(\mathbf{X}\boldsymbol{\alpha} - \mathbf{y}) + \lambda \mathfrak{R}(\boldsymbol{\alpha}), \tag{7}$$

where $\mathfrak{R}(\boldsymbol{\alpha})$ is the structured sparsity regularizer to be explained in the following subsection. Using the multiplicative form of ψ as

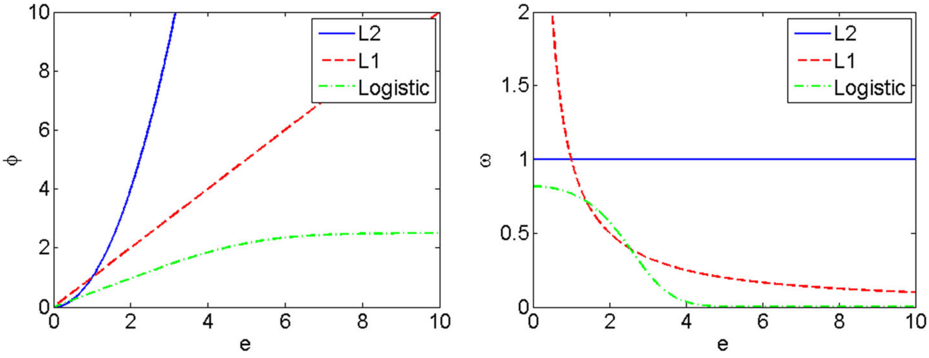


Fig. 2 Potential loss functions (*left*) and the corresponding weight functions (*right*) of l_2-norm, l_1-norm and Logistic function

$$\psi(\mathbf{X}\boldsymbol{\alpha}-\mathbf{y},\mathbf{w})=\sum_{i=1}^{d}w_i\left(y_i-\sum_{j=1}^{n}x_{ij}\alpha_j\right)^2, \tag{8}$$

we have the following minimization of the augmented objective.

$$\min_{\boldsymbol{\alpha},\mathbf{w}}\sum_i\left(w_i\left(y_i-\sum_j x_{ij}\alpha_j\right)^2+\varphi(w_i)\right)+\lambda\mathfrak{R}(\boldsymbol{\alpha}). \tag{9}$$

We will use $J(\boldsymbol{\alpha},\mathbf{w})$ to denote the above objective function. Following the HQ optimization framework [16, 25], a local minimizer $(\boldsymbol{\alpha},\mathbf{w})$ to $J(\boldsymbol{\alpha},\mathbf{w})$ can be alternately calculated by executing the following rules.

$$w_i^{t+1}=\omega\left(y_i-\sum_j x_{ij}\alpha_j^t\right), \tag{10}$$

$$\boldsymbol{\alpha}^{t+1}=\arg\min_{\boldsymbol{\alpha}}\left\|\mathbf{W}^{1/2}(\mathbf{y}-\mathbf{X}\boldsymbol{\alpha}^t)\right\|_2^2+\lambda\mathfrak{R}(\boldsymbol{\alpha}^t), \tag{11}$$

Where $\boldsymbol{\alpha}^t$ is an estimated coefficient vector in the t-th iteration, $\omega(\cdot)$ is the weight function derived from the conjugate of $\phi(\cdot)$. $\omega(\cdot)$ satisfies that

$$\psi(e_i,\omega(e_i))+\varphi(\omega(e_i))\le\psi(e_i,w_i)+\varphi(w_i). \tag{12}$$

Here, $\mathbf{W}$ is a diagonal matrix with each entry on the diagonal as $(\mathbf{W})_{ii}=w_i^{t+1}$. The optimization of $\boldsymbol{\alpha}^{t+1}$ can be rewritten as the following regularized quadratic problem

$$\boldsymbol{\alpha}^{t+1}=\arg\min_{\boldsymbol{\alpha}}\left\|\hat{\mathbf{X}}\boldsymbol{\alpha}^t-\hat{\mathbf{y}}\right\|_2^2+\lambda\mathfrak{R}(\boldsymbol{\alpha}^t), \tag{13}$$

where $\hat{\mathbf{X}}=\sqrt{\mathbf{W}}\mathbf{X}$ and $\hat{\mathbf{y}}=\sqrt{\mathbf{W}}\mathbf{y}$. The robust improvement of structured sparse representation is given in Algorithm 1.

Based on the HQ framework [16, 25], we use the Logistic weight function to determine $\mathbf{w}$ for fair comparison with the robust sparse coding (RSC) [34], whose loss function $\phi(\cdot)$ and weight function $\omega(\cdot)$ as shown in Fig. 3 are respectively defined as

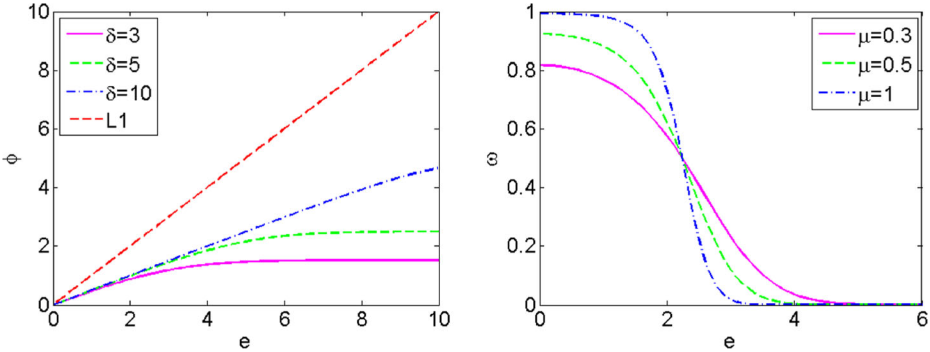


Fig. 3 The Logistic weight function ω relevant to the multiplicative form of HQ for corresponding potential loss function ϕ

$$\phi(e_i) = \frac{-1}{2\mu}\ln\frac{1+\exp\left(\mu\delta - \mu e_i^2\right)}{1+\exp(\mu\delta)}, \tag{14}$$

$$\omega(e_i) = \frac{\exp\left(\mu\delta - \mu e_i^2\right)}{1+\exp\left(\mu\delta - \mu e_i^2\right)} \tag{15}$$

The logistic M-estimator shown is parameterized by (μ, δ). μ controls the decreasing rate of weight and δ is the demarcation point. When δ is small, this function behaves like l_1-norm on small errors and l_0-norm on large errors, which give less punishment on large outliers than l_1 M-estimator. The weight is bounded by 0 and 1.

Algorithm 1 Robust Improvement Based on HQ

```
Input: Training data $\mathbf{X}$, test sample $\mathbf{y}$ and regularization parameter $\lambda$, initial guess $\boldsymbol{\alpha}^0$;
Output: The representation coefficient $\boldsymbol{\alpha}$ and weight vector $\mathbf{w}$.
$t = 0$;
While not converged do
    For $i = 1, 2, \ldots, d$ do
        $e_i = y_i - \sum_{j=1}^{n} x_{ij}\alpha_j^t$; // compute the coding residual for $i$-th pixel
        $w_i^{t+1} = \omega(e_i)$; // compute the weight for $i$-th pixel based on the weight function
    End for
    $(\mathbf{W})_{ii} = w_i^{t+1}$; // $\mathbf{W}$ is a diagonal matrix with $i$-th diagonal entry as $w_i$
    $\mathbf{X} = \sqrt{\mathbf{W}}\mathbf{X}$ and $\hat{\mathbf{y}} = \sqrt{\mathbf{W}}\mathbf{y}$; // re-arrange the variable
    $\boldsymbol{\alpha}^{t+1} = \arg\min_{\boldsymbol{\alpha}} \left\|\mathbf{X}\boldsymbol{\alpha}^t - \hat{\mathbf{y}}\right\|_2^2 + \lambda\Re\left(\boldsymbol{\alpha}^t\right)$; // solve the representation coefficient
    $t = t + 1$;
End while
```

3.2 Structured sparsity

Considering $\boldsymbol{\alpha} = \left[\alpha_1^1, \cdots, \alpha_{|S_1|}^1, \cdots, \alpha_1^{|S_c|}, \cdots, \alpha_{|S_c|}^{|S_c|}\right]$, where$\{S_k\}$, $k = 1, 2, \cdots, c$ is the partition of training samples from different classes and $|S_k|$ is the number of samples in the k-th class, the RGSR model can be reformulated as

$$\min_{\boldsymbol{\alpha},\mathbf{w}} \sum_i \left(w_i \left(y_i - \sum_j x_{ij}\alpha_j \right)^2 + \varphi(w_i) \right) + \lambda \sum_{S_k} \|\boldsymbol{\alpha}_{S_k}\|_2. \tag{16}$$

Obviously, (13) has the following specific form

$$\boldsymbol{\alpha}^{t+1} = \arg\min_{\boldsymbol{\alpha}} \left\| \hat{\mathbf{X}}\boldsymbol{\alpha} - \hat{\mathbf{y}} \right\|_2^2 + \lambda \sum_{S_k} \| \boldsymbol{\alpha}_{S_k} \|_2. \tag{17}$$

Set its derivative with respect to α to zero and we can obtain a simple method to update $\boldsymbol{\alpha}^{t+1}$ as

$$\hat{\boldsymbol{\alpha}} = \left(\hat{\mathbf{X}}^T \hat{\mathbf{X}} + \lambda \mathbf{L} \right)^{-1} \hat{\mathbf{X}}^T \hat{\mathbf{y}}, \tag{18}$$

where **L** is defined as

$$\mathbf{L} = \begin{bmatrix} \frac{1}{2\|\boldsymbol{\alpha}_{S_1}\|_2} \mathbf{I}_{|S_1|} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \frac{1}{2\|\boldsymbol{\alpha}_{S_c}\|_2} \mathbf{I}_{|S_c|} \end{bmatrix}. \tag{19}$$

The whole procedure of optimizing the RGSR model is summarized in Algorithm 2. The stop criteria for the outer loop and inner loop are respectively defined as

$$\left\| \mathbf{w}^{t+1} - \mathbf{w}^t \right\|_2 / \| \mathbf{w}^t \|_2 \leq \varepsilon_1, \tag{20}$$

$$\left\| \mathrm{obj}^{k+1} - \mathrm{obj}^k \right\|_2 / \left\| \mathrm{obj}^k \right\|_2 \leq \varepsilon_2, \tag{21}$$

where obj is the objective value of (17), ε_1 and ε_2 are small positive values (0.05 and 0.001 in following experiments). *k* here is the index of iteration in the inner loop. The

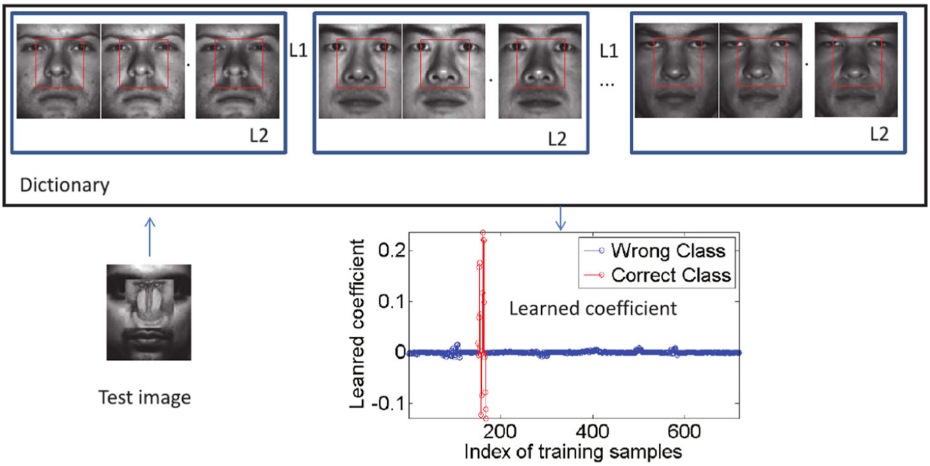


Fig. 4 The diagram of RGSR model. For an occluded test sample, RGSR learns the weight map which can mask the corresponding area of training samples shown in red rectangle. The learned coefficients for intra-class samples are measured by l_2-norm, while that for inter-class samples are measured by l_1-norm. This results in the structured sparsity

diagram of RGSR is shown in Fig. 4. The convergence analysis of Algorithm 2 will be given below.

Algorithm 2. Robust Structured Sparse Representation Model

```
Input: Training data $\mathbf{X}$, test sample $\mathbf{y}$, regularization parameter $\lambda$ and initial guess $\boldsymbol{\alpha}^0$;
Output: The representation coefficient $\boldsymbol{\alpha}$ and feature weight $\mathbf{w}$.
$t=0$;
// Outer loop for optimizing the feature weight vector $\mathbf{w}$
While not converged do
    // the following loop is to update the feature weight vector $\mathbf{w}$
    For $i=1,2,\ldots,d$ do
        $e_i=y_i-\sum_{j=1}^{n}x_{ij}\alpha_j^t$; //compute the coding residual for each pixel
        $w_i^{t+1}=\exp\left(\mu\delta-\mu e_i^2\right)/\left(1+\exp\left(\mu\delta-\mu e_i^2\right)\right)$; //compute the weight based on (15);
    End for
    $(\mathbf{W})_{ii}=w_i^{t+1}$; // $\mathbf{W}$ is a diagonal matrix with $i$-th diagonal entry as $w_i$
    $\mathbf{X}=\sqrt{\mathbf{W}}\mathbf{X}$ and $\hat{\mathbf{y}}=\sqrt{\mathbf{W}}\mathbf{y}$; //re-arrange the variable
    // the following loop for optimizing $\boldsymbol{\alpha}^{t+1}$ based on (17)
    $k=0$ and initialize $\boldsymbol{\alpha}^k$;
    While not converged do
        Compute $\mathbf{L}^k$ based on (19);
        $\boldsymbol{\alpha}^{k+1}=\left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{L}^k\right)^{-1}\mathbf{X}^T\hat{\mathbf{y}}$;
        $k=k+1$;
    End while
    Compute the reconstruction $\mathbf{y}_{rec}^{t+1}=\mathbf{X}\boldsymbol{\alpha}^{t+1}$;
    $t=t+1$;
End while
```

Lemma 1 [24] *For arbitrary two non-zero vectors* $\mathbf{u}$ *and* $\mathbf{v}$*, the following inequality holds*

$$\|\mathbf{u}\|_2-\frac{\|\mathbf{u}\|_2^2}{2\|\mathbf{v}\|_2}\leq\|\mathbf{v}\|_2-\frac{\|\mathbf{v}\|_2^2}{2\|\mathbf{v}\|}.$$

Theorem 2 *The alternating optimization of objective* $J(\boldsymbol{\alpha},\mathbf{w})$ *in* (16) *by Algorithm 2 converges.*

Proof First we show that the inner loop for optimizing α decreases the objective of (17). In the $t+1$-th iteration, once the feature weight vector $\mathbf{w}^{t+1}$, we need to optimize $\boldsymbol{\alpha}^{t+1}$ by solving the following objective

$$\boldsymbol{\alpha}^{t+1} = \arg\min_{\boldsymbol{\alpha}} \left\| \hat{\mathbf{X}}\boldsymbol{\alpha}^t - \hat{\mathbf{y}} \right\|_2^2 + \lambda(\boldsymbol{\alpha}^t)^T \mathbf{L}\boldsymbol{\alpha}^t. \tag{22}$$

Obviously, we have

$$\left\| \hat{\mathbf{X}}\boldsymbol{\alpha}^{t+1} - \hat{\mathbf{y}} \right\|_2^2 + \lambda\left(\boldsymbol{\alpha}^{t+1}\right)^T \mathbf{L}\boldsymbol{\alpha}^{t+1} \le \left\| \hat{\mathbf{X}}\boldsymbol{\alpha}^t - \hat{\mathbf{y}} \right\|_2^2 + \lambda(\boldsymbol{\alpha}^t)^T \mathbf{L}\boldsymbol{\alpha}^t. \tag{23}$$

Based on Lemma 1, we have

$$\lambda\sum_{S_k} \|\boldsymbol{\alpha}_{S_k}^{t+1}\|_2 - \lambda\sum_{S_k} \frac{\|\boldsymbol{\alpha}_{S_k}^{t+1}\|_2^2}{2\|\boldsymbol{\alpha}_{S_k}^t\|_2} \le \sum_{S_k} \|\boldsymbol{\alpha}_{S_k}^t\|_2 - \lambda\sum_{S_k} \frac{\|\boldsymbol{\alpha}_{S_k}^t\|_2^2}{2\|\boldsymbol{\alpha}_{S_k}^t\|_2},$$

which is equivalent to

$$\lambda\sum_{S_k} \|\boldsymbol{\alpha}_{S_k}^{t+1}\|_2 - \lambda\left(\boldsymbol{\alpha}^{t+1}\right)^T \mathbf{L}\boldsymbol{\alpha}^{t+1} \le \lambda\sum_{S_k} \|\boldsymbol{\alpha}_{S_k}^t\|_2 - \lambda(\boldsymbol{\alpha}^t)^T \mathbf{L}\boldsymbol{\alpha}^t. \tag{24}$$

Add both sides of inequalities (23) and (24) together and we can obtain

$$\left\| \hat{\mathbf{X}}\boldsymbol{\alpha}^{t+1} - \hat{\mathbf{y}} \right\|_2^2 + \lambda\sum_{S_k} \|\boldsymbol{\alpha}_{S_k}^{t+1}\|_2 \le \left\| \hat{\mathbf{X}}\boldsymbol{\alpha}^t - \hat{\mathbf{y}} \right\|_2^2 + \lambda\sum_{S_k} \|\boldsymbol{\alpha}_{S_k}^t\|_2, \tag{25}$$

which means that the solution to optimize α satisfies $J(\boldsymbol{\alpha}^{t+1}, \mathbf{w}^{t+1}) \le J(\boldsymbol{\alpha}^t, \mathbf{w}^{t+1})$.

According to the property of weight function $\omega(\cdot)$ shown in inequality (12), for a fixed $\boldsymbol{\alpha}^{t+1}$, we have $J(\boldsymbol{\alpha}^t, \mathbf{w}^{t+1}) \le J(\boldsymbol{\alpha}^t, \mathbf{w}^t)$. Combine with the above conclusion $J(\boldsymbol{\alpha}^{t+1}, \mathbf{w}^{t+1}) \le J(\boldsymbol{\alpha}^t, \mathbf{w}^{t+1})$, and we can get

$$J\left(\boldsymbol{\alpha}^{t+1}, \mathbf{w}^{t+1}\right) \le J\left(\boldsymbol{\alpha}^t, \mathbf{w}^{t+1}\right) \le J(\boldsymbol{\alpha}^t, \mathbf{w}^t). \tag{26}$$

Thus, the objective value series $\{\cdots, J(\boldsymbol{\alpha}^t, \mathbf{w}^t), J(\boldsymbol{\alpha}^t, \mathbf{w}^{t+1}), J(\boldsymbol{\alpha}^{t+1}, \mathbf{w}^{t+1}), \cdots\}$ generated by Algorithm 2 converges as $t \to \infty$.

4 Experimental studies

In this section, we conduct experiments to show the effectiveness of the proposed RGSR model.

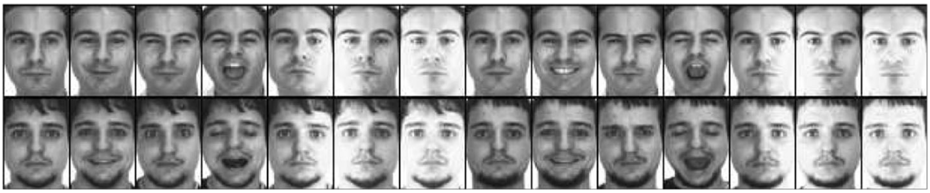


Fig. 5 Sample images of two subjects in AR data set

Table 1 Face recognition rates on AR data set

AR	#30	#54	#120	#300
NN	62.5 %	68.0 %	70.1 %	71.3 %
NS	66.1 %	70.1 %	75.4 %	76.0 %
SVM	66.1 %	69.4 %	74.5 %	76.0 %
SRC [30]	73.5 %	83.3 %	90.1 %	93.3 %
CRC-RLS [36]	64.4 %	80.5 %	90.0 %	93.4 %
RSC [34]	71.4 %	86.8 %	94.0 %	96.0 %
RGSR	73.7 %	87.7 %	94.4 %	96.7 %

4.1 Experimental settings

We conduct experiments under two settings: (1) face recognition without occlusion but with variations such as illumination and expression changes and (2) face recognition with three types of occlusions: random pixel corruption, random block occlusion and real disguise.

There are three parameters involved in RGSR model: the regularization parameter λ and the Logistic weight function related parameters (μ, δ). In this paper, λ is set as 0.001 by default. According to the properties of (μ, δ), smaller δ (larger μ) is encouraged if the image is grossly corrupted, which can group more pixels into outliers. For a corrupted image, the squared error vector is $\boldsymbol{\pi}=[e_1^2, e_2^2, \cdots, e_d^2]$ (e_i is the coding residual with respect to the i-th pixel) and its ascending sorted version is $\boldsymbol{\pi}_a$. We set δ as $\boldsymbol{\pi}_a$ ($\lfloor \tau d \rfloor$) and$\mu=c/\delta$. Thus, two new parameters (c, τ) are introduced to build the tight connection to the corruption level instead of using (μ, δ) directly [34]. In our experiments, the corruption level for the second setting is higher than the first one and smaller τ is preferred; thus we set (c, τ) respectively as (8,0.8) and (8,0.6) for both settings.

4.2 Face recognition without occlusion

In this section, we compare RGSR with state-of-the-art methods such as nearest neighbor (NN), nearest subspace (NS), linear support vector machine, SRC [30], collaborative representation based classification (CRC) [36] and RSC [34].

Similar to general face recognition methods, we perform experiments in PCA subspace in which the Eigenface [28] features are used as input. By applying PCA to the training data, (17) will become $\|\mathbf{P}\left(\hat{\mathbf{X}}\boldsymbol{\alpha}-\hat{\mathbf{y}}\right)\|_2^2 + \lambda\sum_{S_k}\|\boldsymbol{\alpha}_{S_k}\|_2$, where $\mathbf{P}$ is the projection matrix.

Three benchmark face data sets: AR [23], Extended Yale B [11, 20] and CMU Multi-PIE [13] are used in the following experiments.

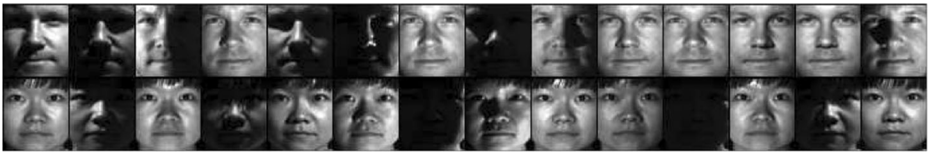


Fig. 6 Sample images of two subjects in Extended Yale B data set

Table 2 Face recognition rates on Extended Yale B data set

Extended Yale B	#30	#84	#150	#300
NN	66.3 %	85.8 %	90.0 %	91.6 %
NS	63.6 %	94.5 %	95.1 %	96.0 %
SVM	92.4 %	94.9 %	96.4 %	97.0 %
SRC [30]	89.1 %	95.1 %	96.8 %	97.9 %
CRC-RLS [36]	74.0 %	92.9 %	96.5 %	98.0 %
RSC [34]	91.3 %	98.1 %	98.4 %	99.4 %
RGSR	88.2 %	96.4 %	98.6 %	99.6 %

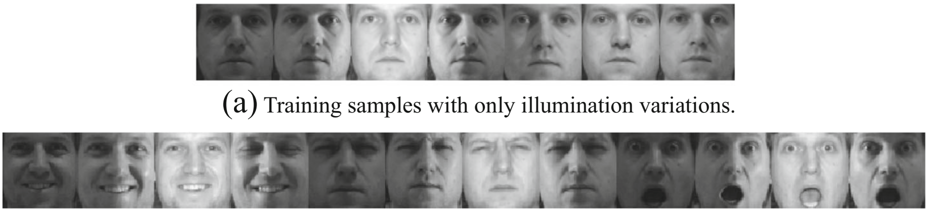
1) *AR*: As in [30], a subset with only illumination and expression changes which contains 50 males and 50 females was chosen from the AR data set. In our experiments, for each subject, the seven images from Session 1 were used for training, and the other seven images from Session 2 for testing. The image size is cropped to 60 × 43 pixels. Figure 5 shows some sample images of two subjects in AR data set.

The comparison of RGSR and its competing methods is given in Table 1. RGSR achieves the best results among all methods in all dimensions. RGSR consistently performs better than RSC because the structured sparsity is encouraged than the l_1-norm induced flat sparsity.

2) *Extended Yale B*: The Extended Yale B data set contains 16,128 face images of 38 human subjects under 9 pose and 64 illumination conditions. A subset contains about 2414 frontal face images from 38 individuals is selected. We used the cropped and normalized 54 × 48 images, which were taken under varying illuminations. We randomly split the database into two halves. One half (about 32 images per subject) was used as training samples, and the other half for testing. Figure 6 shows some sample images of two subjects in Extended Yale B data set.

Table 2 shows the recognition rates versus feature dimension by the competing methods. RSGR has much performance improvement in higher dimensions. In this experiment, the training samples from each class are sufficient (about 32) and they are more uncorrelated in lower dimensional subspace when comparing with AR data set; thus the l_1-norm is more appropriate to regularize the representation of samples with big variations. RGSR has limited improvement over RSC in higher dimensional subspace.

3) *Multi-PIE*: The CMU Multi-PIE data set contains face images of 337 subjects taken in four sessions with simultaneous variations in pose, expression, and illumination. Among these



(a) Training samples with only illumination variations.

(b) Test samples with smile (1-4), squint (5-8) and surprise (9-12) expressions and illumination

Fig. 7 Sample images of one subject in Multi-PIE data set. (**a**) Training samples with only illumination variations. (**b**) Test samples with smile (1–4), squint (5–8) and surprise (9–12) expressions and illumination variations

Table 3 Face recognition rates on Multi-PIE data set

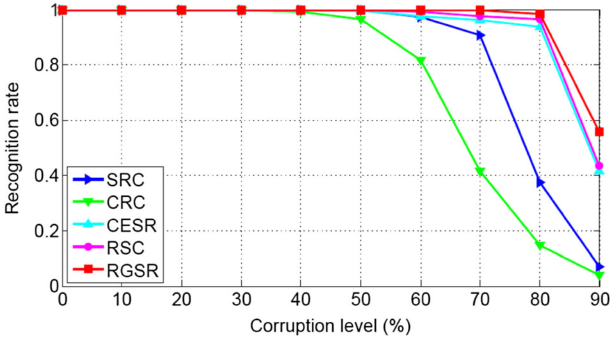
Multi-PIE	Smile-S1	Smile-S3	Surperise-S2	Squint-S2
NN	88.7 %	47.3 %	40.1 %	49.6 %
NS	89.6 %	48.8 %	39.6 %	51.2 %
SVM	88.9 %	46.3 %	25.6 %	47.7 %
SRC [30]	93.7 %	60.3 %	51.4 %	58.1 %
CRC_RLS [36]	92.1 %	58.7 %	52.3 %	57.9 %
RSC [34]	97.8 %	75.0 %	68.8 %	64.6 %
RGSR	98.3 %	77.1 %	70.4 %	66.7 %

'Smile-S1' means that the test samples with smile expressions are from Session 1; 'Surperise-S2' means that the test samples with surprise expressions are from Session 2; 'Squint-S2' means that the test samples with squint expressions are from Session 2

337 subjects, all the 249 subjects in Session 1 are used as training set. Four subsets with both illumination and expression variations in Session 1, 2 and 3 are used for testing. We select the seven frontal images with extreme illuminations {0,1,7,13,14,16,18} as in [29, 34] and neutral expression to form the training set. Four typical frontal images with illuminations {0,2,7,13} and different expressions (smile in Session 1 and 3, squint and surprise in Session 2) are used to form the testing set. Figure 7 shows the training and testing samples of one subject in Multi-PIE data set. All face images are cropped into 64x64 pixels. We use the eigenface with dimensionality of 300 as the face feature.

Table 3 shows the recognition rates in four testing sets by the competing methods. From Table 3, we can see that RGSR achieves the best performance in all tests, and RSC holds the second place. All the methods obtain their best results when Smile-S1 is used for testing because the training samples are also from Session 1. The highest recognition rate of RGSR on Smile-S1 is 98.3 %. From testing set Smile-S1 to Smile-S3, the recognition rate of RGSR drops by 21.2 % because of the longer data acquisition time interval. For testing sets Surprise-S2 and Squint-S2, the recognition rates of RGSR are respectively 1.6 and 2.1 % higher than those of RSC. This reflects the insights from two aspects: 1) the robustness improvement is effective for removing the influence of illuminations and expression variations. Both RGSR and RSC outperform the remaining methods. 2) The label information in sparse coding stage is important. The flat sparsity

Fig. 8 Recognition rates versus different percentages of pixel corruption



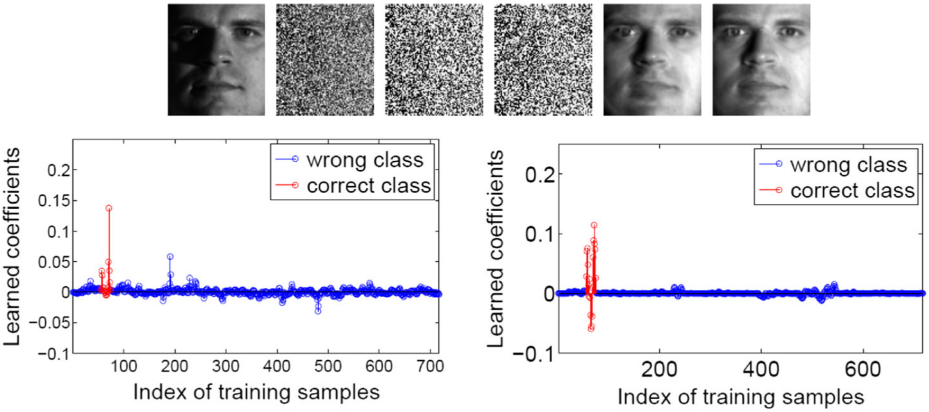


Fig. 9 An example of face recognition with random pixel corruption (80 % level). **First-row**: the original image, corrupted image, weight maps obtained via RSC and RGSR, reconstructed images via RSC and RGSR; **Second-row**: learned coefficients via RSC and RGSR

does not consider label information of training samples in coding stage, which leads to the slightly weak performance of RSC in comparison with RGSR.

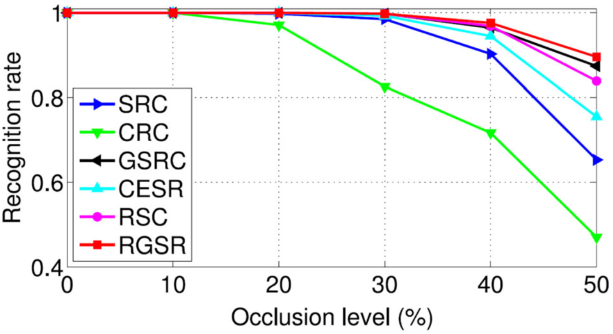
4.3 Face recognition with occlusion

In this section, we test the robustness of RGSR to different types of occlusions including random pixel corruption, random block occlusion and real disguise.

1) *Face Recognition with Random Pixel Corruption*: Identical to the experimental settings in [30], we used Subsets 1 and 2 (717 images, normal-to-moderate lighting conditions) of Extended Yale B for training, and used Subset 3 (453 images, more extreme lighting conditions) for testing. The face images are resized to 96 × 84 pixels. For each test image, we replaced a certain percentage of its pixels by uniformly distributed random values within [0,255]. The corrupted pixels were randomly chosen from test image and the locations are unknown.

We compare RGSR with SRC, CRC, correntropy-based sparse representation (CESR) [14, 15] and RSC. Figure 8 shows the results of different models under the corruption level from 0 to 90 %. All the models except CRC perform well when the corruption level is lower than 60 %. However, when the percentage is more than 60 %, the performance of SRC was greatly

Fig. 10 Recognition rates versus different percentages of block occlusion



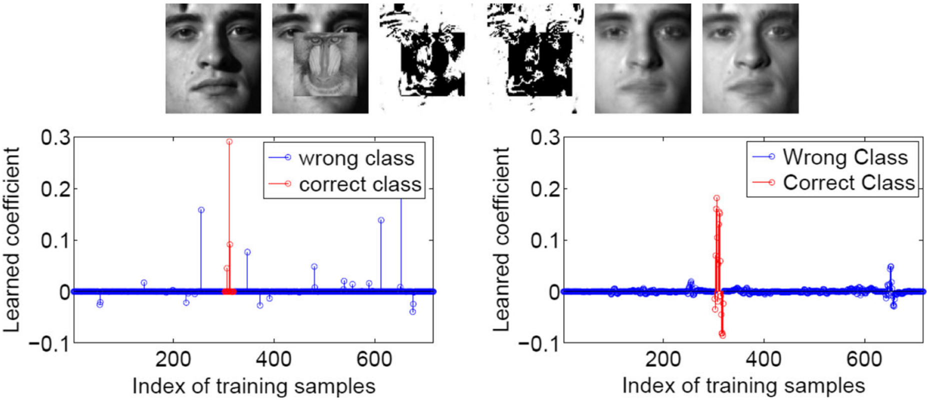


Fig. 11 An example of face recognition with block occlusion (40 % level). **First-row**: the original image, occluded image, weight maps obtained via RSC and RGSR, reconstructed images via RSC and RGSR; **Second-row**: learned coefficients via RSC and RGSR

reduced. Even with 90 % pixels corrupted, RGSR still obtains an acceptable accuracy (55.85 %). A representative example of RSC and RGSR with 80 % random pixel corruption is shown in Fig. 9. The corrupted face image is difficult to recognize even for human; however, both RSC and RGSR can accurately estimate the weight map and recover the clean image. Both the corrupted pixels and shadow region are reflected in the learned weight maps. The reconstructed images are faithful to the original image but with better visual quality. From the learned coefficients, we find only one sample from the correct class plays a main role in reconstruction for RSC; while for RGSR, all the samples from the correct class have large coefficients. Therefore, the reconstructed face image by RGSR is cleaner than that by RSC especially for the right half face (lower illumination). The coefficients obtained by RGSR have obvious grouping effect and are smoother than those of RSC.

2) *Face Recognition with Block Occlusion*: In this part, we test the robustness of RGSR model to block occlusion. We also used the same experimental settings as in [30], i.e., Subsets 1 and 2 of Extended Yale B for training and Subset 3 for testing. The images were resized to 96 × 84 pixels. We compare RGSR with SRC, CRC, Gabor-SRC (use Gabor features to construct the occlusion dictionary) [32], CESR and RSC. Figure 10 shows the change trend of different models under the level of the occluded area from 0 to 50 %. Obviously, RGSR gets promising results even if the occlusion level is high. Figure 11 gives a representative example under 40 % random block occlusion.

Table 4 Recognition rates on AR with disguise occlusion

Algorithms	Sun-glasses	Scarves
SRC [30]	87.0 %	59.5 %
CRC-RLS [36]	68.5 %	90.5 %
GSRC [32]	93.0 %	79.0 %
CESR [14, 15]	99.0 %	42.0 %
RSC [34]	98.5 %	96.5 %
RGSR	100 %	97.5 %

Table 5 Recognition rates on AR with sunglasses or scarves in Session 1 and Session 2

Algorithms	Sg-s1	Sc-s1	Sg-s2	Sc-s2
SRC [30]	89.3 %	32.3 %	57.3 %	12.7 %
CRC-RLS [36]	43.7 %	30.7 %	17.7 %	13.7 %
GSRC [32]	87.3 %	85.0 %	45.0 %	66.0 %
CESR [14, 15]	95.3 %	38.0 %	79.0 %	20.7 %
RSC [34]	94.7 %	91.0 %	80.3 %	72.7 %
RGSR	99.0 %	93.0 %	86.7 %	76.7 %

From the coefficients learned by RSC, we can find that many training samples from the wrong classes contribute to the reconstruction, which blurs the area around the lip in the reconstructed image. There are only three non-zero values w.r.t. the samples from correct class, which means that the l_1-norm sparsity encourages selecting representative samples when they are highly correlated.

For RGSR, the reconstruction is mainly achieved by the training samples from the correct class because they have similar non-zero values and samples from wrong classes have near-zero values.

3) *Face Recognition with Real Disguise*: A subset from AR data set is used in this experiment, which consists of 2,599 face images from 100 subjects (about 26 samples per

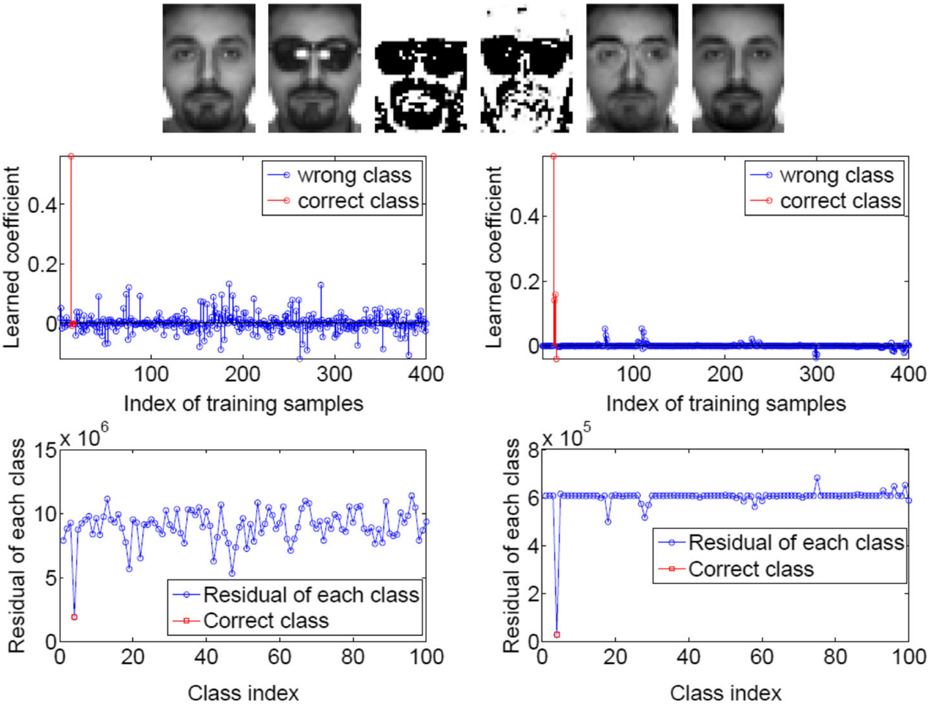


Fig. 12 An example of face recognition with disguise. **First-row**: the face image without disguise, sunglass disguised test image, weight maps obtained via RSC and RGSR, reconstructed images via RSC and RGSR; **Mid-row**: learned coefficients associated with each training sample via RSC and RGSR; **Third-row**: residuals of each class via RSC and RGSR

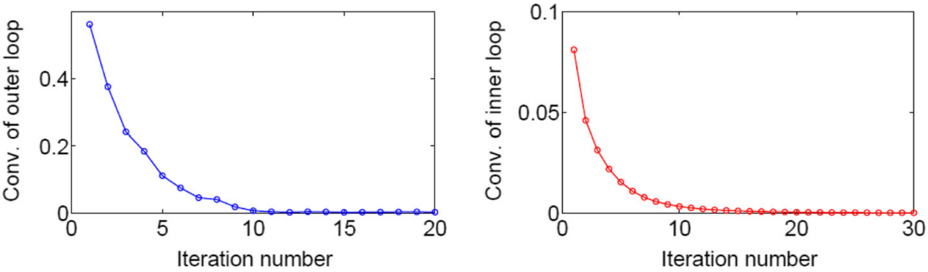


Fig. 13 The convergence of the outer loop for optimizing the feature weight vector (*left*) and inner loop for optimizing the representation coefficient (*right*) of Algorithm 2

subject), 50 males and 50 females. We conduct two tests: one follows the experimental settings in [30], while the other follows [34] and is more challenging. The images are resized to 42 × 30 pixels.

In the first test, 800 images (8 samples per subject) of non-occluded frontal views with various facial expressions in Session 1 and 2 were used for training, while two separate subsets (with sunglasses and scarves) of 200 images (1 sample per subject per Session, with neutral expression) for testing. The recognition rates of different models are listed in Table 4. RGSR achieves 100 % recognition rate under the sunglass disguise and 97.5 % under the scarf disguise, which are respectively 13 and 38 % improvements w.r.t. SRC. Though RSC performs well on both disguises, RGSR still has respectively 1.5 and 1 % improvement over it.

In the second test, we use more complex disguises (disguise with variations of illumination and longer data acquisition interval). 400 images (4 neutral images with different illuminations per subject) of non-occluded frontal views in Session 1 were used for training, while the disguise images (3 images with various illuminations and sunglasses or scarves per subject per Session) in Session 1 and 2 for testing. Table 5 shows the results of different competing models. RGSR obtains much improvement w.r.t. RSC, about 4.3 % (Session 1) and 6.4 % (Session 2) for the sunglass disguise; for the scarf disguise, the improvements are respectively 2 % (Session 1) and 4 % (Session 2). Figure 12 illustrates the classification process of RGSR on a representative example. Compared to RSC, the reconstructed image by RGSR has better visual quality around the eye corner for the disguised test image, which can easily remove the sunglass disguise. The coefficients learned by RGSR have obvious grouping effect, which enforces training samples from the same class have similar coefficients. And there are samples from only a few wrong classes which have large values. But for RSC, the coefficients have large values across each class and correspondingly the coding residual for each class has similar variation tendency.

For validating the convergence of RGSR, Fig. 13 shows the convergence curves w.r.t. the outer loop for optimizing the feature weight vector and inner loop for optimizing the representation coefficient in Algorithm 2, which reflects that even under gross corruption, RGSR can converge in a few iterations (about 10). Under moderate corruption, usually 5 iterations are sufficient.

5 Discussions

In this section, we give discuss on the connection as well as difference between reference [16] and our work.

In [16], the authors compare three types of sparse representation: 1) standard sparse representation models which are not robust to outliers, 2) robust sparse representation models which are robust to outliers and 3) using the M-estimator to measure the coding residual of sparse representation. The experimental results show that the M-estimator in multiplicative form can greatly enhance the robustness of sparse representation model. Therefore, we can easily find the connections as well as differences between [16] and our work. Both methods use M-estimator via half-quadratic optimization to enhance the model's robustness. However, He et al. uses the M-estimators such as Welsch and Huber while the Logistic loss function is employed in our work. He's work considers more sparse representation models (Eqs. (41)-(49) in [16]) while we consider one representative non-robust sparse representation model (Eq. (1) in our work, which is equivalent to (42) in [16]) and one representative robust sparse representation model (Eq. (3) in our work, which is equivalent to (48) in [16]).

Moreover, taking Welsch and Huber M-estimators as an example, we conduct experiments to show the performance differences for both methods. The potential function of Welsch and Huber estimators are respectively shown as follows:

$$\phi_W(e_i) = 1 - \exp\left(-\frac{e_i^2}{\sigma^2}\right) \tag{27}$$

$$\phi_H(e_i) = \begin{cases} e_i^2/2, & |e_i| \le \lambda \\ \lambda|e_i| - \dfrac{\lambda^2}{2}, & |e_i| > \lambda \end{cases} \tag{28}$$

Accordingly, the corresponding weight functions in multiplicative form are respectively defined as

$$\omega_W(e_i) = \exp\left(-\frac{e_i^2}{\sigma^2}\right) \tag{29}$$

$$\omega_H(e_i) = \begin{cases} 1, & |e_i| \le \lambda \\ \dfrac{\lambda}{|e_i|}, & |e_i| > \lambda \end{cases} \tag{30}$$

The involved parameter kernel size σ in Welsch estimator is determined by $\sigma^2 = 0.5 \times median_i\left(\left(y_i - \sum_{j=1}^{n} x_{ij}\alpha_j^t\right)\ ^2\right)$ and the threshold parameter λ in Huber estimator is determined by $\lambda = 0.8 \times median_i\left(\left|y_i - \sum_{j=1}^{n} x_{ij}\alpha_j^t\right|\right)$ [16]. We conduct pairwise comparison between RSGR and He's method by using the above two M-estimators in two experimental settings: face recognition with random pixel corruption and block occlusion. The data sets are the same as those described in Section 4.3. Here we consider the Welsch and Huber M-estimators and their corresponding weight functions in multiplicative form.

1) *Face recognition with random pixel corruption*. In this experimental setting, each test image was corrupted by replacing a set of randomly selected pixels with a random pixel value which follows a uniform distribution over [0,255]. We vary the percentage of image pixels that suffer corruptions from 10 to 80 %. Figure 14 shows the recognition accuracy of both methods, as a function of the level of corruption. From this figure, we have two findings: 1) the RGSR-based methods are basically better than He's work [16]. The reason accounting for this may be caused by the incorporation of structured sparsity which directly considers the label information in sparse coding phase. 2) The performance of Welsch M-estimator-based methods is better than that of Huber M-estimator-based methods, which is consistent with the results in [16].

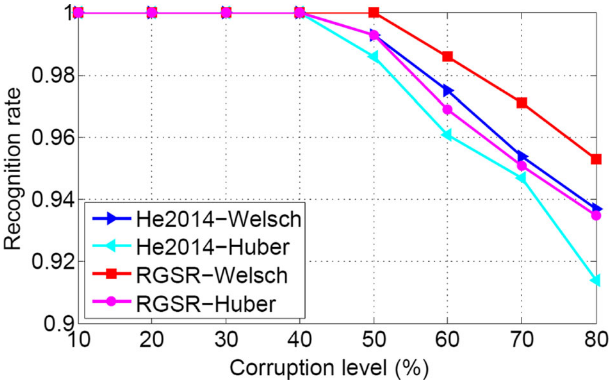


Fig. 14 Recognition rates in terms of different percentage of random pixel corruption: RGSR vs. He's work

2) *Face recognition with block occlusion*. Figure 15 shows the recognition rates of both methods, as a function of the level of occlusion. We can see that both methods show excellent performance across different percentages of occlusion. The results via Welsch M-estimator are slightly better than those obtained via Huber M-estimator. The proposed RGSR on Welsch M-estimator achieves the best recognition rate, which is nearly 88 % even when the occlusion level is 50 %. It is about 3 % higher than that of He's work [16]. This reflects that both the robustness improvement and the structured sparsity are beneficial for learning more effective sparse codes.

6 Conclusion

This paper proposed the robust group sparse representation-based classifier by improving SRC from two aspects: using robust M-estimator to measure the representation fidelity and the group sparsity constraint on the coefficients. The optimization method to proposed RGSR model is efficient and we provide its convergence analysis. The RGSR model was evaluated under different conditions, including variations of illuminations, expressions, occlusion and combined corruption. Our experimental results demonstrated that RGSR performs well especially under high-dimensional cases and outperforms many state-of-the-art methods including robust sparse coding.

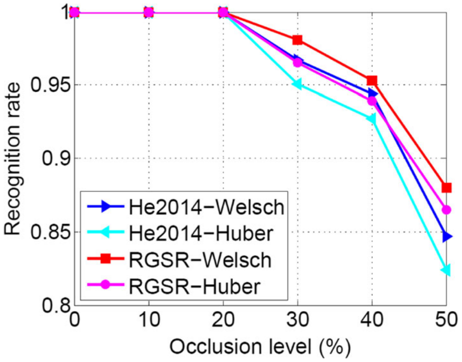


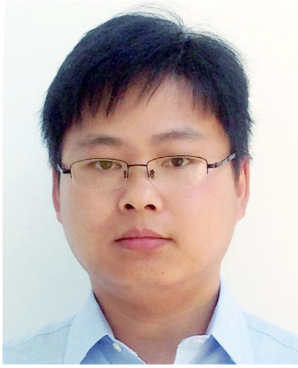
Fig. 15 Recognition rates in terms of different percentage of block occlusion: RGSR vs. He's work

Acknowledgments The authors would like to thank the anonymous reviewers for their helpful comments on this paper. This work was partially supported by the National Natural Science Foundation of China (No.61272248), the National Basic Research Program of China (No.2013CB329401), the Science and Technology Commission of Shanghai Municipality (No.13511500200) and the funding of Hangzhou Dianzi University (KYS055616025).

References

1. Bickel PJ, Ritov Y, Tsybakov AB (2009) Simultaneous analysis of Lasso and Dantzig selector. Ann Stat 37(4):1705–1732
2. Bioucas-Dias JM, Figueiredo MA (2007) A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration. IEEE Trans Image Process 16(12):2992–3004
3. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press
4. Combettes PL, Wajs VR (2005) Signal recovery by proximal forward-backward splitting. Multiscale Model Simul 4(4):1168–1200
5. Ding C, Zhou D, He X, Zha H (2006) R1-PCA: rotational invariant L1-norm principal component for robust subspace factorization. Proc Int Conf Mach Learn 281–288
6. Donoho DL, Elad M (2003) Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization. Proc Natl Acad Sci 100(5):2197–2202
7. Du L, Li X, Shen Y-D (2012) Robust nonnegative matrix factorization via half-quadratic minimization. Proc IEEE Int Conf Data Min 201–210
8. Elhamifar E, Vidal R (2009) Sparse subspace clustering. Proc IEEE Conf Comput Vis Pattern Recognit 2790–2797
9. Geman D, Reynolds G (1992) Constrained restoration and the recovery of discontinuities. IEEE Trans Pattern Anal Mach Intell 14(3):367–383
10. Geman D, Yang C (1995) Nonlinear image recovery with half-quadratic regularization. IEEE Trans Image Process 4(7):932–946
11. Georghiades AS, Belhumeur PN, Kriegman D (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660
12. Grave E, Obozinski GR, Bach FR (2011) Trace Lasso: a trace norm regularization for correlated designs. Proc Adv Neural Inf Proces Syst 2187–2195
13. Gross R, Matthews I, Cohn J, Kanade T, Baker S (2010) Multi-PIE. Image Vis Comput 28:807–813
14. He R, Sun Z, Tan T, Zheng W-S (2011) Recovery of corrupted low-rank matrices via half-quadratic based nonconvex minimization. Proc IEEE Conf Comput Vis Pattern Recognit 2889–2896
15. He R, Zheng W-S, Hu B-G (2011) Maximum correntropy criterion for robust face recognition. IEEE Trans Pattern Anal Mach Intell 33(8):1561–1576
16. He R, Zheng W-S, Tan T, Sun Z (2014) Half-quadratic-based iterative minimization for robust sparse representation. IEEE Trans Pattern Anal Mach Intell 36(2):261–275
17. Huber PJ (2011) Robust statistics. Springer
18. Lai J, Jiang X (2014) Supervised trace lasso for robust face recognition. Proc IEEE Int Conf. Multimedia Expo 1–6
19. Lee HY, Hoo WL, Chan CS (2015) Color video denoising using epitome and sparse coding. Expert Syst Appl 42(2):751–759
20. Lee K-C, Ho J, Kriegman D (2005) Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans Pattern Anal Mach Intell 27(5):684–698
21. Liu W, Pokharel PP, Principe JC (2007) Correntropy: properties and applications in non-gaussian signal processing. IEEE Trans Signal Process 55(11):5286–5298
22. Lu C, Tang J, Lin M, Lin L, Yan S, Lin Z (2013) Correntropy induced *l*2 graph for robust subspace clustering. Proc IEEE Int Conf Comput Vis 1801–1808
23. Martinez AM (1998) The AR face database, CVC Technical Report
24. Nie F, Huang H, Cai X, Ding CH (2010) Efficient and robust feature selection via joint *l*2, 1-norms minimization. Proc Adv Neural Inf Proces Syst 1813–1821
25. Nikolova M, Ng MK (2005) Analysis of half-quadratic minimization methods for signal and image recovery. SIAM J Sci Comput 27(3):937–966
26. Rosas-Romero R, Tagare HD (2014) Segmentation of endocardium in ultrasound images based on sparse representation over learned redundant dictionaries. Eng Appl Artif Intell 29:201–210
27. Tibshirani R (1996) Regression shrinkage and selection via the Lasso. J R Statist Soc B 58(1):267–288
28. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):71–86
29. Wagner A, Wright J, Ganesh A, Zhou Z, Ma Y (2009) Towards a practical face recognition system: robust registration and illumination by sparse representation. Proc IEEE Conf Comput Vis Pattern Recognit 597–604

30. Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y (2009) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 31(2):210–227
31. Yang AY, Zhou Z, Balasubramanian, Sastry SS, Ma Y (2013) Fast *l*1-minimization algorithms for robust face recognition. IEEE Trans Image Process 22(8):3234–3246
32. Yang M, Zhang L (2010) Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary. Proc Eur Conf Comput Vis 448–461
33. Yang M, Zhang L, Yang J, Zhang D (2010) Metaface learning for sparse representation based face recognition. Proc Int Conf Image Proces 1601–1604
34. Yang M, Zhang L, Yang J, Zhang D (2011) Robust sparse coding for face recognition. Proc IEEE Conf Comput Vis Pattern Recognit 625–632
35. Yang S, Lv Y, Ren Y, Jiao L (2013) Superpixel-wise semi-supervised structural sparse coding classifier for image segmentation. Eng Appl Artif Intell 26(10):2608–2612
36. Zhang L, Yang M, Feng X (2011) Sparse representation or collaborative representation: Which helps face recognition?. Proc IEEE Int Conf Comput Vis 471–478
37. Zhao P, Yu B (2006) On model selection consistency of Lasso. J Mach Learn Res 7:2541–2563



Yong Peng received the B.S. degree from Hefei New Star Research Institute of Applied Technology, the M.S. degree from Graduate University of Chinese Academy of Sciences, and the PhD degree from Shanghai Jiao Tong University, all in computer science, in 2006, 2010 and 2015, respectively. From September 2012 to August 2014, he was a visiting PhD student in the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor. He joined School of Computer Science and Technology, Hangzhou Dianzi University as an Assistant Professor in June 2015 where he is currently serving as a Research Associate Professor. He was awarded by the Presidential Scholarship, Chinese Academy of Sciences in 2009 and National Scholarship for Graduate Students, Ministry of Education in 2012. His research interests are machine learning, pattern recognition and evolutionary computation. He has published several papers in peer reviewed journals such as *Information Sciences*, *Neural Networks*, *Applied Soft Computin*g, *Neurocomputing* and *Neural Processing Letters*.

Bao-Liang Lu received his B.S. degree from Qingdao University of Science and Technology in 1982, the M.S. degree from Northwestern Polytechnical University in 1989 and the Ph.D. degree from Kyoto University in 1994. From 1982 to 1986, he was with the Qingdao University of Science and Technology. From April 1994 to March 1999, he was a Frontier Researcher at the Bio-Mimetic Control Research Center, the Institute of Physical and Chemical Research (RIKEN), Japan. From April 1999 to August 2002, he was a Research Scientist at the RIKEN Brain Science Institute. Since August 2002, he has been a full Professor at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include brain-like computing, neural networks, machine learning, pattern recognition, and brain–computer interface. He was the past President of the Asia Pacific Neural Network Assembly (APNNA) and the general Chair of ICONIP2011. He serves as Associate Editors of *IEEE Transactions on Cognitive and Developmental Systems* and *Neural Networks* (Elsevier). He is a governing board member of APNNA and a senior member of IEEE.