Pattern Recognition 45 (2012) 1104–1118

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Beyond sparsity: The role of L_1-optimizer in pattern classification

Jian Yang [a,b,*], Lei Zhang [c], Yong Xu [d], Jing-yu Yang [a]

[a] Department of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, PR China
[b] Computation and Neural Systems, California Institute of Technology, Pasadena, CA 91125, USA
[c] Biometric Research Centre, Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong
[d] Bio-Computing Research Centre, Shenzhen Graduate School of Harbin Institute of Technology, Shenzhen, China

ARTICLE INFO

Article history:
Received 28 September 2010
Received in revised form
23 July 2011
Accepted 22 August 2011
Available online 30 August 2011

Keywords:
Sparse representation
Pattern classification
Classifier
Feature extraction

ABSTRACT

The newly-emerging sparse representation-based classifier (SRC) shows great potential for pattern classification but lacks theoretical justification. This paper gives an insight into SRC and seeks reasonable supports for its effectiveness. SRC uses L_1-optimizer instead of L_0-optimizer on account of computational convenience and efficiency. We re-examine the role of L_1-optimizer and find that for pattern recognition tasks, L_1-optimizer provides more classification meaningful information than L_0-optimizer does. L_0-optimizer can achieve sparsity only, whereas L_1-optimizer can achieve closeness as well as sparsity. Sparsity determines a small number of nonzero representation coefficients, while closeness makes the nonzero representation coefficients concentrate on the training samples with the same class label as the given test sample. Thus, it is closeness that guarantees the effectiveness of the L_1-optimizer based SRC. Based on the closeness prior, we further propose two kinds of class L_1-optimizer classifiers (CL_1C), the closeness rule based CL_1C (C-CL_1C) and its improved version: the Lasso rule based CL_1C (L-CL_1C). The proposed classifiers are evaluated on five databases and the experimental results demonstrate advantages of the proposed classifiers over SRC in classification performance and computational efficiency for large sample size problems.

1. Introduction

"Sparse (or sparsity)" becomes a popular term in neuroscience, information theory and signal processing and related areas in the past decade [1–10]. Vinje and Gallant's studies suggested that primary visual cortex (area V1) uses a sparse code to efficiently represent natural scenes. The receptive fields function forms a sparse representation of the visual world during natural vision [1]. Olshausen and Field [2] and Serre [3] revealed that the firing of the neurons with respect to a given input image is typically highly sparse if these neurons are viewed as an overcomplete dictionary of base signal elements at each visual stage. All of these findings form a physiological basis for sparse coding and sparse representation.

Sparse coding and sparse representation has recently aroused intensive interest pattern recognition and computer vision area. Labusch et al. [11] presented a simple sparse-coding strategy for digit recognition and achieved state-of-the-art results on the MNIST benchmark. Zhou et al. [12] presented a sparse principal component analysis (SPCA), which uses the Lasso (elastic net) to produce modified principal components with sparse loadings and yields encouraging results for regular multivariate data and gene expression arrays. Subsequently, different formulations of SPCA and sparse linear discriminant analysis have been developed [13–15]. Cai et al. [16] suggested a sparse projection over graph and showed its power for document classification. Qiao et al. [17] put forward a sparse preserving projection technique and demonstrated its effectiveness for face recognition. Actually, Qiao et al.'s sparse preserving projection can be viewed as a special case of L_1-graph under a general dimensionality reduction framework [18–20]. Recently, Wright et al. presented a sparse representation based classification method and successfully applied it to recognize human faces with varying lighting condition, occlusion and disguise [21]. In addition, Wright et al. [20] reviewed other sparse representation methods that were applied to different vision tasks such as image super-resolution [22], image denoising and inpainting [23], signal and image classification [24–27], etc. In most of these applications, using sparsity as a prior leads to state-of-the-art results.

This paper focuses on sparse representation based classification. The basic idea of Wright et al.'s sparse representation based classification (SRC) method is to represent a given test sample as a sparse linear combination of all training samples; the sparse nonzero representation coefficients are supposed to concentrate on the training samples with the same class label as the test sample. The sparsest solution can be sought by solving the L_0-optimization problem. However, solving L_0-optimization problem is NP hard

* Corresponding author at: Department of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, PR China.
E-mail addresses: csjyang@mail.njust.edu.cn, jianyang@caltech.edu (J. Yang), cslzhang@comp.polyu.edu.hk (L. Zhang), laterfall2@yahoo.com.cn (Y. Xu), yangjy@mail.njust.edu.cn (J.-y. Yang).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.08.022

and even difficult to approximate [28]. Recent development in the emerging theory of sparse representation and compressed sensing [29,30,5] reveals that finding the solution of the L_0 optimization problem is equivalent to finding the solution of the L_1 optimization problem for certain dictionaries. The L_1-optimizer is therefore used instead of the L_0-optimizer in SRC.

Regarding SRC, a fundamental problem is: when one uses all classes of training samples to represent a given test sample, why does the small number of nonzero representation coefficients concentrate on the homo-class training samples? Wright and Ma [31] and Wright et al. [20] addressed the extended L_1-minimization model based error correction problem and interpreted why accurate recovery of sparse signals is possible even if the corruption error is almost dense. But the fundamental problem mentioned remains open, just as said in [20] "—the striking discriminative power of the sparse representation still lacks rigorous mathematical justification". In this paper, our intention is to seek some reasonable supports for SRC.

We begin with an example of the two-class handwritten numerical recognition problem in which the L_0-solution fails while the L_1-Solution succeeds for classification. This fact indicates that the sparest representation gained by the L_0-optimizer is not sufficient for classification. Conversely, the L_1-optimizer may not achieve the sparest solution, but achieves the meaningful solution for correct classification. We then introduce the closeness theory to reveal the connection of the L_1-solution to classification. The L_1-norm of nonzero weights can provide a metric to measure the degree of closeness between the testing sample and its support training samples, while the L_0-norm cannot. The effectiveness of SRC is due to the closeness prior: the homo-class representation leads to the minimal L_1-norm of nonzero weights. The physical meaning of minimizing L_1-norm of weights becomes clearer if a weight-sum-to-one constraint is imposed onto the L_1-optimizer, i.e., searching for the support training samples such that their centroid is closest to the given test sample in the sense of L_1-norm.

We further introduce the theory of (global) neighborliness and local neighborliness of quotient polytope associated with a dictionary, and use it to in-depth analyze the role of L_1-optimizer in pattern recognition. In global neighborliness cases where the quotient polytope associated with the dictionary formed by all training samples is t-neighborly, L_1-optimizer achieves both sparsity and closeness globally. In such cases, L_1-solution equals to L_0-solution, i.e., the globally sparsest solution. This sparsest solution determines the set of support training samples that is closest to the given testing sample. In local neighborliness cases where the quotient polytope associated with the dictionary formed by class training samples is t-neighborly, L_1-optimizer achieves sparsity locally and closeness globally. In such cases, L_1-solution is a locally sparse solution, possibly not the globally sparsest solution, but it is the solution which is most meaningful for classification. Beyond neighborliness, the degree of sparsity of L_1-solution cannot be guaranteed, but its effectiveness for classification can still be guaranteed, i.e., the L_1-solution determines the set of support training samples that is closest to the given testing sample.

Based on the closeness analysis, we present two class L_1-optimizer classifiers (CL_1C). To this end, we first provide theoretical, geometrical and computational justifications for supporting the class training samples based representation. We then present the closeness rule based CL_1C (C-CL_1C), which uses the *closeness* (i.e., the L_1-norm of the representation coefficients) as a criterion to make a decision. A normalized version of C-CL_1C is obtained based on geometrical meaning of the solution of the constrained L_1-optimizer. To overcome the limitation of C-CL_1C, which restricts the testing sample to lie on faces of the class polytopes and only suits for large sample size problems, we further present the Lasso rule based CL_1C (L-CL_1C) and its normalized version. To test the proposed classifiers, we finally use four databases which involve different recognition tasks: the AR database for gender recognition, the CENPARMI database for handwritten numeral Recognition, the NUST603 database for handwritten Chinese character recognition, the Extended Yale B database for face recognition. The experimental results demonstrate the effectiveness of the proposed classifiers.

2. Outline of sparse representation-based classifier

Suppose there are c known pattern classes. Let $\mathbf{A}_i$ be the matrix formed by the training samples of Class i, i.e., $\mathbf{A}_i=[\mathbf{y}_{i1},\mathbf{y}_{i2},\ldots,\mathbf{y}_{iM_i}]\in R^{N\times M}{}_i$, where M_i is the number of training samples of Class i. Let us define a matrix $\mathbf{A}=[\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_c]\in R^{N\times M}$, where $M=\sum_{i=1}^{c}M_i$. The matrix $\mathbf{A}$ is obviously composed of entire training samples.

Given a test sample $\mathbf{y}$, we represent $\mathbf{y}$ in a overcomplete dictionary whose basis vectors are training sample themselves, i.e., $\mathbf{y}=\mathbf{A}\mathbf{w}$. This system of linear equation is underdetermined if $N<M$. The idea of sparse representation based classification is motivated by the following observation: a valid test sample $\mathbf{y}$ can be sufficiently represented using only the training samples from the same class. The representation is naturally sparse if training sample size is large enough. The sparser the recovered representation coefficient vector $\mathbf{w}$ is, the easier it will be to accurately determine the identity of the test sample $\mathbf{y}$ [21].

The sparsest solution to $\mathbf{y}=\mathbf{A}\mathbf{w}$ can be sought by solving the following optimization problem:

$$(L_0)\ \hat{\mathbf{w}}_0=\arg\min\|\mathbf{w}\|_0,\ \text{subject to}\ \mathbf{A}\mathbf{w}=\mathbf{y}, \tag{1}$$

where $\|\cdot\|_0$ denotes the L_0-norm, which counts the number of nonzero entries in a vector.

Solving L_0 optimization problem in Eq. (1), however, is NP hard and extremely time-consuming. Fortunately, recent research efforts reveal that for certain dictionaries, if the solution $\hat{\mathbf{w}}_0$ is spare enough, finding the solution of the L_0 optimization problem is equivalent to finding the solution to the following L_1 optimization problem [5,29,30]:

$$(L_1)\ \hat{\mathbf{w}}_1=\arg\min\|\mathbf{w}\|_1,\ \text{subject to}\ \mathbf{A}\mathbf{w}=\mathbf{y}. \tag{2}$$

This problem can be solved in polynomial time by standard linear programming algorithms [33]. A more efficient algorithm, e.g., the homotopy algorithm which has a computational complexity that is linear to the size of the training set, is available recently [34].

After obtaining the sparsest solution $\hat{\mathbf{w}}_1$, we can design a sparse representation based classifier (SRC) in terms of the class reconstruction residual. Specifically, for Class i, let $\delta_i: R^N\to R^N$ be the characteristic function that selects the coefficients associated with the ith class. For $\mathbf{w}\in R^N$, $\delta_i(\mathbf{w})$ is a vector whose only nonzero entries are the entries in $\mathbf{w}$ that are associated with Class i. Using only the coefficients associated with the ith class, one can reconstruct a given test sample $\mathbf{y}$ as $\hat{\mathbf{y}}_i=\mathbf{A}\delta_i(\hat{\mathbf{w}}_1)$. The corresponding class reconstruction residual is defined by

$$r_i(\mathbf{y})=\|\mathbf{y}-\hat{\mathbf{y}}_i\|_2=\|\mathbf{y}-\mathbf{A}\delta_i(\hat{\mathbf{w}}_1)\|_2. \tag{3}$$

The SRC decision rule is: if $r_l(\mathbf{y})=\min_i r_i(\mathbf{y})$, $\mathbf{y}$ is assigned to Class l.

For convenience, the training samples (or basis vectors) associated with nonzero representation coefficients are called the *support training samples* (or support basis vectors) in the remainder of the paper, which is in spirit consistent with the concept of support vectors in support vector machine (SVM) literature [32].

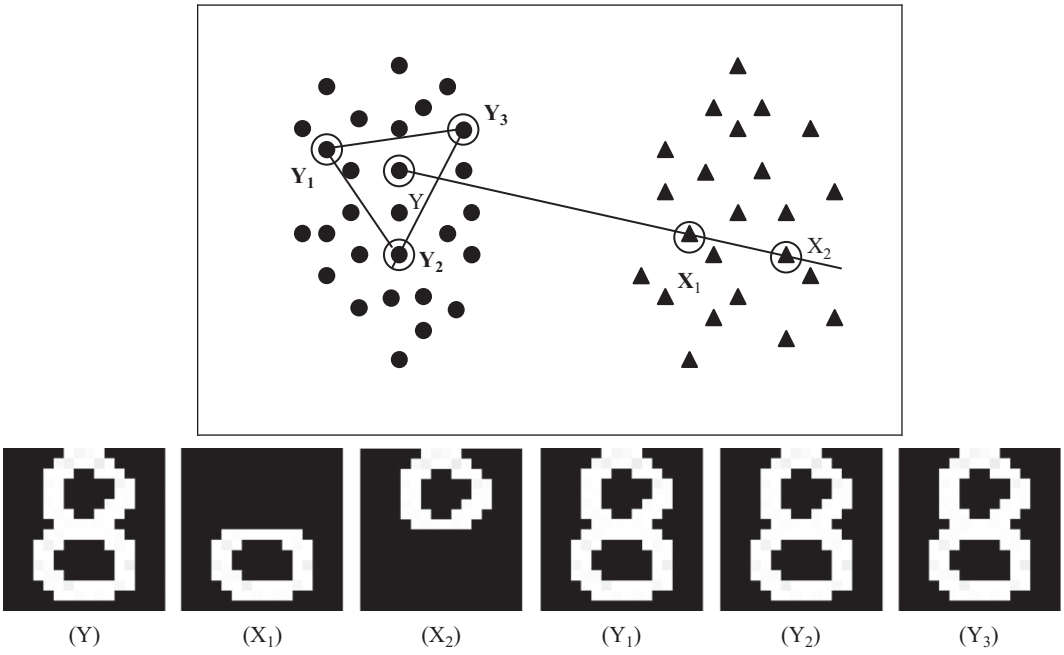

Fig. 1. Illustration of the two-class handwritten numerical recognition problem, where the dots represent samples of "8", while the triangles represent samples of "0".

Finally, it should be stressed that SRC relies on the following assumption to guarantee the sparsity of the representation of a test sample:

Assumption 1. (Large Sample Size Assumption): there are sufficient number of training samples for each class, such that any test sample can be sufficiently represented using only the training samples from the same class.

In all of our analysis in reminder of the paper, we always assume that the above assumption holds.

3. A classification example: L_0 solution fails while L_1 solution succeeds

The idea of sparse representation based classification implies that the identity of a test sample can be accurately determine, as long as the solution is sufficient sparse. However, this is not always the case. Sometimes, a test sample is misclassified even if the solution is extremely sparse. This problem becomes prominent when there exists a class formed by parts of objects among many classes of objects.

For example, in handwritten (or printed) numerical recognition problems, "0" can be viewed as a part of "8". For simplicity, let us consider a two-class problem which contains the samples of "0" and "8". As shown in Fig. 1, a sample of "8", Y, can be represented extremely sparsely by samples of same class, Y_1, Y_2 and Y_3. At the same time, Y can be represented extremely sparsely as well by samples of the other class ("0" class), X_1, X_2. Specifically, Y can be represented in the following ways:

$$\text{(Homo-class representation) } \mathbf{Y}=\tfrac{1}{3}\mathbf{Y}_1+\tfrac{1}{3}\mathbf{Y}_2+\tfrac{1}{3}\mathbf{Y}_3 +0\mathbf{Y}_4+\cdots+0\mathbf{Y}_{M_1}=\mathbf{A}\hat{\mathbf{w}}_1, \quad (4)$$

$$\text{(Hetero-class representation) } \mathbf{Y}=1\mathbf{X}_1+1\mathbf{X}_2+0\mathbf{X}_3 +0\mathbf{X}_4+\cdots+0\mathbf{X}_{M_2}=\mathbf{A}\hat{\mathbf{w}}_0. \quad (5)$$

Let $\mathbf{A}=[\mathbf{Y}_1,\ldots,\mathbf{Y}_{M_1},\mathbf{X}_1,\ldots,\mathbf{X}_{M_2}]$. If one uses L_0-optimizer in Eq. (1) to seek for the optimal solution, the sparsest representation coefficient vector is

$$\hat{\mathbf{w}}_0=\left[\underbrace{0,\ldots,0}_{M_1},\underbrace{1,1,0,\ldots,0}_{M_2}\right]^T.$$

here, the L_0-norm of $\hat{\mathbf{w}}_0$ is 2. In this case, the SRC makes a wrong decision and assign a sample of "8", Y, to the class of "0".

This example shows us that *natural sparsity itself cannot guarantee correct classification*. More specifically, for a given sample, one uses the L_0-optimizer to obtain the optimal solution. Even if the optimal solution is sufficiently sparse, the resulting representation still cannot guarantee the correctness of classification.

In order to achieve correct classification results, based on the SRC decision rule, it is necessary to make the sparse nonzero representation coefficients of a sample concentrate on the homo-class samples. The L_0-optimizer is not qualified for this, although it suffices to recover the sparse representation of a sample.

For the same two-class numerical classification problem mentioned above, if we use L_1-optimizer in Eq. (2) instead of L_0-optimizer to seek for the optimal solution, the sparsest representation coefficient vector is

$$\hat{\mathbf{w}}_1=\left[\underbrace{1/3,1/3,1/3,0,\ldots,0}_{M_1},\underbrace{0,0,\ldots,0}_{M_2}\right]^T,$$

because the $\hat{\mathbf{w}}_1$ results in the minimal the L_1-norm. It is obvious that $\|\hat{\mathbf{w}}_1\|_1=1<\|\hat{\mathbf{w}}_0\|_1=2$. In terms of the SRC decision rule, $\hat{\mathbf{w}}_1$ produces the correct classification result.[1] It appears that the L_1-optimizer recovered sparse representation is more meaningful for classification, although it is a bit denser than L_0-optimizer recovered one.

Compared to L_0-optimizer, it seems that L_1-optimizer has an extra power to concentrate the sparse nonzero representation

[1] Note that here for simplicity and being easily understood, we use the original samples to directly represent Y, as shown in Eqs. (4) and (5). If we normalize Y_1, Y_2, Y_3, X_1, X_2 to be unit vectors before the representation, we still have the same classification result.

coefficients of a sample on the homo-class samples. It is this power that makes L_1 solution more effective than L_0 solution for pattern recognition. In the following sections, we provide theoretical analysis on L_0-optimizer and L_1-optimizer, reveal the role of L_1-optimizer in pattern recognition and further show why L_1-optimizer based classifier is effective for pattern classification.

4. Why L_1 solution is more effective than L_0 solution for classification?

In this section, we will provide an intuitive interpretation for why L_1-optimizer can recover classification meaningful representation. Here, we do not address the problem whether L_1 solution is sparse or not (actually this is another problem we will address in the next section). Rather, we focus on the function of L_1-optimizer, i.e., minimizing the L_1-norm of nonzero representation coefficients (weights) and reveal its connections to pattern classification.

For a given test sample $\mathbf{y}$, the objective function of L_1-optimizer is $\|\mathbf{w}\|_1$, which provides a metric to measure the magnitude of the nonzero reconstruction weights under the constraint of $\mathbf{Aw}=\mathbf{y}$. The minimization of $\|\mathbf{w}\|_1$ is apt to select the set of support training samples associated with the smallest nonzero reconstruction weights in the sense of the L_1-norm, among all candidate sets of samples which can produce the representation $\mathbf{y}=\mathbf{Aw}$. Whereas, the objective function of L_0-optimizer, minimizing $\|\mathbf{w}\|_0$, does not provide this weight-selecting mechanism, except for determining the degree of sparsity, i.e., the minimum number of support training samples for representing $\mathbf{y}$. For instance, if two set of support vectors can represent the test sample with the same degree of sparsity, the L_1-optimizer has the ability to choose the set with minimal L_1-norm of nonzero weights, whereas the L_0-optimizer does not have this ability. In other words, L_1-optimizer is more informative than L_0-optimizer, since its objective function provides a mechanism for support vector selection, i.e., selecting the support training samples to represent a given test sample with the minimal "representation cost".

In the following, we aim to reveal the intuitive connection between the minimal L_1-norm of nonzero representation weights and classification. To this end, we first give the following assumption:

> **L_1-Prior (Closeness Prior)** For a given testing sample, using only the homo-class support training samples to represent it can give rise to the minimal representation weights (coefficients) in the sense of the L_1-norm.

The reason that calls L_1-Prior the Closeness Prior lies in two aspects. First, each sample should be naturally represented by the homo-class support training samples, thus a sample is closed in the homo-class sample set. Second, the magnitude of representation weights determines the degree of closeness between a testing sample and the set of support training samples used to represent it. The minimal representation weights imply that a testing sample is closest to the set of support training samples.

For the two-class numerical classification example mentioned above, as shown in Fig. 1, the test sample Y is very close to the sample set $\{Y_1, Y_2, Y_3\}$ of Class "8", but far away from the sample set $\{X_1, X_2\}$ of Class "0", noticing that the representation weights of the former, as a whole, is much smaller than that of the later in sense of L_1-norm. This closeness provides very important information for classification. Although the test sample Y can be represented most sparsely by X_1 and X_2, it is far away from these two samples, so Y is not likely to belong to the class of X_1 and X_2.

Actually, the physical meaning of minimal L_1-norm of nonzero representation weights (i.e., $\|\mathbf{w}\|_1$) becomes clearer if we put a weight-sum-to-one constraint on the L_1-optimizer. The constraint can eliminate the effect of rotation and rescaling. By adding this constraint, the L_1-optimizer becomes

$$\text{(Constrained } L_1)\ \hat{\mathbf{w}}_1=\arg\min\|\mathbf{w}\|_1,\ \text{subject to } \mathbf{Aw}=\mathbf{y} \text{ and } \mathbf{1}^T\mathbf{w}=1, \tag{6}$$

where $\mathbf{1}$ is an M-dimensional column vector in which every element is 1. Since $\mathbf{1}^T\mathbf{w}=1$ is a linear equation, it easy to integrate it with $\mathbf{Aw}=\mathbf{y}$, forming a new system of linear equations. Thus, Eq. (6) is equivalent to the following augmented L_1-optimizer:

$$\text{(Constrained } L_1)\ \hat{\mathbf{w}}_1=\arg\min\|\mathbf{w}\|_1,\ \text{subject to } \overline{\mathbf{A}}\mathbf{w}=\overline{\mathbf{y}}, \tag{7}$$

where

$$\overline{\mathbf{A}}=\begin{bmatrix}\mathbf{A}\\ \mathbf{1}^T\end{bmatrix},\ \overline{\mathbf{y}}=\begin{bmatrix}\mathbf{y}\\ 1\end{bmatrix}.$$

In this way, the algorithm for L_1-optimizer can be directly used to resolve the constrained L_1-optimizer. Assuming $\hat{\mathbf{w}}_1$ is the obtained optimal solution, $\mathbf{y}=\mathbf{A}\hat{\mathbf{w}}_1$ is actually a weighted mean of the support training samples since $\mathbf{1}^T\hat{\mathbf{w}}_1=1$.

Now, looking back at the two constraints of the constrained L_1-optimizer in Eq. (6) from a new viewpoint, we can see that the given sample $\mathbf{y}$ lies on the following hyperplane:

$$F=\left\{\mathbf{z}=\sum_{j=1}^{M}w_j\mathbf{x}_j\,\middle|\,\sum_{j=1}^{M}w_j=1\right\}, \tag{8}$$

here we rewrite all training samples in $\mathbf{A}$ as $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_M$. If the solution of the constrained L_1-optimizer is sparse, without loss of generality, let the first K training samples are support basis vectors. Then, the given sample $\mathbf{y}$ actually lies on a $(K-1)$-dimensional hyperplane

$$F=\left\{\mathbf{z}=\sum_{j=1}^{K}w_j\mathbf{x}_j\,\middle|\,\sum_{j=1}^{K}w_j=1\right\}. \tag{9}$$

Assuming that $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_K$ belong to the same class, the above hyperplane can be viewed as a local $(K-1)$-dimensional patch (local face) on the class manifold.

If we choose a reference point on the hyperplane as the origin, for example, using the centroid (the mean of the support training samples) $\overline{\mathbf{x}}=1/K\sum_{j=1}^{K}\mathbf{x}_j$ as the origin, the constraint $\sum_{j=1}^{K}w_j=1$ can be removed and the hyperplane in Eq. (9) is equivalently expressed as [35]

$$F=\left\{\mathbf{z}-\overline{\mathbf{x}}=\sum_{j=1}^{K}w_j\vec{\mathbf{x}}_j\right\},\ \text{where } \vec{\mathbf{x}}_j=\mathbf{x}_j-\overline{\mathbf{x}}. \tag{10}$$

Now, let us consider the problem in the new coordinate system whose axes are $\vec{\mathbf{x}}_1,\ldots,\vec{\mathbf{x}}_K$ and origin is $\overline{\mathbf{x}}$. If $\vec{\mathbf{x}}_1,\ldots,\vec{\mathbf{x}}_K$ are unitary vectors, for the test sample $\mathbf{y}$ on the hyperplane F, the L_1-norm of its corresponding weights, $\|\mathbf{w}\|_1$, is actually the L_1-distance from $\mathbf{y}$ to the origin $\overline{\mathbf{x}}$, i.e., the mean (centroid) of the support training samples, as shown in Fig. 2 where $K=3$. From this point, we know that the implication of $\|\mathbf{w}\|_1$: suggesting a metric to measure the distance between the testing sample and the set of support training samples. Minimizing $\|\mathbf{w}\|_1$, therefore,

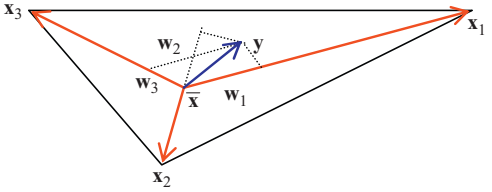

Fig. 2. Illustration of the geometric meaning of $\|\mathbf{w}\|_1$: the L_1-distance from $\mathbf{y}$ to the origin $\overline{\mathbf{x}}$ in the coordinate system formed by $\vec{\mathbf{x}}_1,\ldots,\vec{\mathbf{x}}_K$, here $K=3$.

implies to search for the support training samples such that their centroid is closest to the given test sample.

In a word, if a given test sample complies with the Closeness Prior, L_1-optimizer can concentrate the nonzero sparse representation coefficients of a test sample onto its homo-class support samples such that their centroid is closest to the test sample in the sense of L_1-norm. This provides an underlying justification for the effectiveness of the L_1-optimizer based sparse representation classifier.

In addition, from the example in Section 3, we can conclude that sparsity itself pays more attention to the local reconstruction, while closeness focuses on the global similarity. This global similarity is critical for pattern classification.

5. Analysis of the role of L_1-optimizer in pattern recognition

In this section, we first outline the existing theoretical results on the equivalence between the L_0 and L_1 problems which was developed by Donoho [36,37], then use the theory of neighborliness to address the uniqueness question and finally present the sufficient and necessary condition for the L_0–L_1 equivalence in Section 5.1. In Section 5.2, we apply the L_0-L_1 equivalence to analyze the role of L_1-optimizer in pattern recognition.

5.1. Fundamentals

For a given, general dictionary $\mathbf{A}$ and signal $\mathbf{y}$, the equivalence between the L_0 problem and the L_1 problem involves the following two questions [36]:

(1) *Uniqueness*: having the solution of the L_1 problem, under which conditions can we guarantee that this is also the solution of the L_0 problem?
(2) *Equivalence*: knowing the solution of the L_0 problem, what are the conditions under which L_1 is guaranteed to lead to the exact same solution? This question is called L_1/L_0 *equivalence*.

The Uniqueness question has been answered by the following theorem [36,37]:

Theorem 1. *(Uniqueness Theorem): given a general dictionary* $\mathbf{A}$*, given its corresponding Spark value* σ*, and given a signal* $\mathbf{y}$*, the solution* $\mathbf{w}$ *of the* L_1 *problem is also the solution of the* L_0 *problem if* $\|\mathbf{w}\|_0 \le \sigma/2$.

Theorem 1 involves a concept of the Spark value. Given a matrix $\mathbf{A}$, $\sigma = \mathrm{Spark}(A)$ is defined as the largest possible number such that every sub-set of σ columns from A are linearly independent, and at least one sub-set of $\sigma+1$ columns of A are linearly dependent.

The equivalence question has been addressed by Donoho [37] based on the ideas from the theory of convex polytopes. The related concept of quotient polytope corresponding to a dictionary $\mathbf{A}$ and its *neighborliness* are given below:

Definition 1. Let $\mathbf{a}_i$ denote the ith column of a $d \times n$ matrix $\mathbf{A}$. A quotient polytope P associated to $\mathbf{A}$ is defined as the convex hull of the $2n$ points $(\pm\mathbf{a}_i,\ i=1,\ldots,n)$ in R^d. The $2n$ points $\pm\mathbf{a}_i$ are called vertices of P. P is centrosymmetric and is called (centrally) k-neighborly if every subset of $k+1$ points not including an antipodal pair spans a face of P.

For any point on a k-dimensional face of a closed, convex polytope P, its representation is unique, and vice versa. Formally, the following lemma holds [37]:

Lemma 1. *(Unique Representation) Consider a k-face* $F_k(P)$ *and suppose that* F *is a k-simplex. Let* $\mathbf{x} \in F$*. Then (a)* $\mathbf{x}$ *has a unique representation as a convex combination of vertices of* P*; (b) this representation places nonzero weights only on vertices of* F*. Conversely, suppose that* F *is a k-dimensional closed convex subset of* P *with properties (a) and (b) for every* $\mathbf{x} \in F$*. Then* F *is a k-simplex and a k-face of* P.

Actually, the concept of neighborliness can be understood from the polytope map point of view. Let $C \subset R^n$ be the n-dimensional cross-polytope, characterized as the convex hull of the signed unit basis vectors $\pm\mathbf{e}_i$ with $i=1,\ldots,n$ and as the L_1 ball in R^n, i.e.,

$$\|\mathbf{w}\|_1 \le 1. \tag{11}$$

After being transformed by the matrix A, the cross-polytope C is mapped into a convex polytope $P=AC$. Neighborliness means that the l-faces of P are simply images under A of the l-faces of C. Specifically, we have the following Lemma

Lemma 2. *(Alternate Form of Neighborliness) [37] Suppose that the centrosymmetric polytope* $P=\mathbf{A}C$ *has 2n vertices and is k-neighborly. Then*

$$\forall l = 0,\ldots,k-1,\ \forall F \in F_l(C),\ \mathbf{A}F \in F_l(P). \tag{12}$$

Conversely, suppose that Eq. (12) holds; then $P=AC$ has $2n$ vertices and is k-neighborly.

Combining Lemmas 1 and 2, we know that if P is k-neighborly, there exists a one-to-one mapping from l-faces of the cross-polytope C to l-faces of the quotient polytope P, where $l<k$. For each point $\mathbf{w}$ on l-faces of the cross-polytope C, there is a unique, corresponding point $\mathbf{y}$ on l-faces of the quotient polytope P such that $\mathbf{y}=\mathbf{A}\mathbf{w}$, and vice versa. Fig. 3(a) gives an example of k-neighborliness ($k=3$), where each point on l-faces of C is mapped into a unique point on l-faces of P ($l<3$). Fig. 3(b) shows an example of non-k-neighborliness ($k=3$), where there exist two points on l-faces of C that are mapped into a point $\mathbf{y}$. The point $\mathbf{y}$ must be an interior point of P from Lemma 1, since its representation is not unique. The image of the point $\mathbf{w}_1$ or $\mathbf{w}_2$ on l-faces of C is not on l-faces of P, so P is not k-neighborly ($k=3$) from Lemma 2.

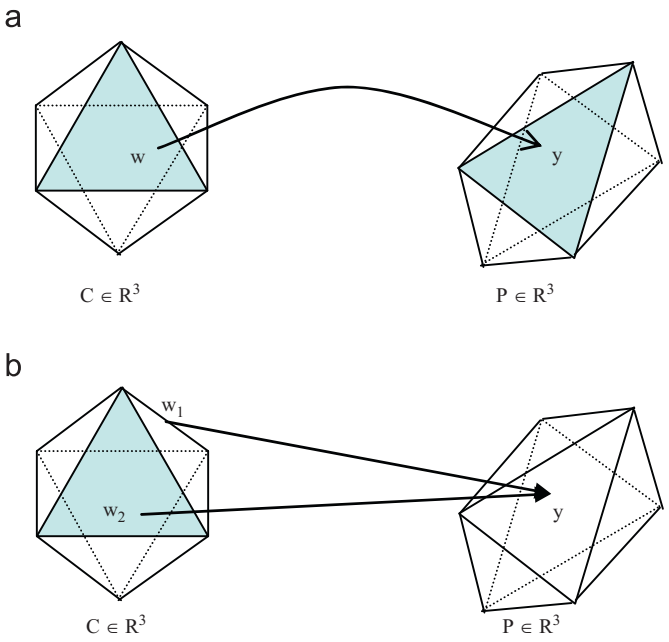

Fig. 3. Illustration of k-neighborliness and non-k-neighborliness of $P=AC$ from the mapping point of view. (a) P is k-neighborly ($k=3$) and (b) P is not k-neighborly ($k=3$).

Donoho [37] has connected the neighborliness to the question of L_1/L_0 equivalence:

Theorem 2. *(Equivalence from Neighborliness) Let $\mathbf{A}$ be a $d\times n$ matrix, $d<n$. The quotient polytope P has 2n vertices and is k-neighborly if and only if $\mathbf{w}_0$ is the unique optimal solution of the L_1 problem whenever $\mathbf{y}=\mathbf{A}\mathbf{w}_0$ has a solution $\mathbf{w}_0$ with at most k nonzeros.*

From the above results on neighborliness, an upper bound on the sparsity level at which L_1- optimizer can solve L_0 problem has been obtained [38,37]

Corollary 1. *Let P be a centrosymmetric d-polytope with $d\geq 2$ and $n\geq d+2$. If P is k-neighborly, we have*

$$k\leq[(d+1)/3], \tag{13}$$

here, we will go one step further and connect the neighborliness to the question of Uniqueness. To this end, let us first present the following lemma:

Lemma 3. *Let P be a quotient polytope associated to a $d\times n$ dictionary $\mathbf{A}$. If P is k-neighborly, then $\sigma=Spark(\mathbf{A})\geq 2k$.*

The Proof of Lemma 3 is given in appendix. From Theorem 1 and Lemma 3, we have

Theorem 3. *(Uniqueness from Neighborliness): given a general dictionary $\mathbf{A}$ and a signal $\mathbf{y}$, if the associated quotient polytope P is k-neighborly, any solution of the L_1 problem with at most k nonzeros is also the solution of the L_0 problem.*

Combining Theorems 2 and 3, we obtain the following theorem:

Theorem 4. *(L_0–L_1 Equivalence) Let P be a centrosymmetric polytope associated to a $d\times n$ dictionary $\mathbf{A}$. P is k-neighborly if and only if the L_0 problem is equivalent to the L_1 problem, that is, for every $\mathbf{w}_0$ with at most k nonzeros, if it is the solution of the L_0 problem, it must be the unique solution of the L_1 problem, and vice versa.*

5.2. The role of L_1-optimizer in pattern recognition

5.2.1. Achieving both sparsity and closeness globally in global neighborliness cases

Now, we discuss about the L_0–L_1 Equivalence for a special dictionary $\mathbf{A}$ formed by all training samples in SRC. The dictionary associated quotient polytope P is the convex hull of the $2M$ vertices ($\pm\mathbf{x}_{ij}$) corresponding to M training samples. From the Theorem 4, we know that if P is k-neighborly, for every solution of the L_1 problem with at most k nonzeros, it must be the unique solution of the L_0 problem. Conversely, for every solution of the L_0 problem with at most k nonzeros, solving the L_1 problem can exactly recover this sparest solution.

From the analysis in Section 4, we know that for a given test sample $\mathbf{y}$, the objective function of L_1-problem is to select the set of support training samples associated with the smallest nonzero reconstruction weights in the sense of the L_1-norm from all candidate sets of samples which can produce the representation $\mathbf{y}=\mathbf{A}\mathbf{w}$. As a result, the degree of closeness between the testing sample and the set of support training samples is minimal. If the number of support training samples is no more than k, this set of support training samples can also provide the sparsest representation of $\mathbf{y}$ provided that $\mathbf{A}$-associated quotient polytope P is k-neighborly. In summary, L_1-optimizer can achieve both *sparsity* and *closeness* globally if P is k-neighborly.

5.2.2. Achieving sparsity locally and closeness globally in local neighborliness cases

For the dictionary $\mathbf{A}$ formed by all training samples from different classes in SRC, however, the associated quotient polytope P is not necessarily (globally) k-neighborly. Although Donoho [29] has shown that for most large underdetermined systems of linear equations, the miminal L_1-norm solution is also the sparsest solution, it should be noted that this conclusion was drawn based on the assumption that columns of a dictionary are sampled independent and identically-distributed (iid) from the uniform distribution on the unit sphere S^{d-1}(see Theorem 4.2 in [37] for details). However, for SRC, the columns of $\mathbf{A}$ are generally not iid random vectors because they are sampled from different classes with different distributions. So, this conclusion is not suitable for the dictionary used in SRC. In other words, the dictionary formed all training samples from different classes may have a small probability to be k-neighborly.

If $\mathbf{A}$ is not k-neighborly, a test sample might be the interior point of the associated quotient polytope P. An interior point of the quotient polytope P has two or more different original images on faces of the cross-polytope C, as shown in Fig. 3(b). To avoid this many-to-one mapping, a possible way is to split the polytope P into a number of small polytopes such that the interior points exist on faces of the generated small polytopes. That is, when $\mathbf{A}$ is not k-neighborly, we would rather look at the associated quotient polytope P locally than globally. This idea connects to the concept of local neighborliness [37]:

Definition 2. Given a $d\times n$ matrix $\mathbf{A}$ and its associated quotient polytope P, let I denotes the subset of m columns of $\mathbf{A}$ and $\mathbf{A}_I$ denotes the matrix formed by this subset of m columns. If $\mathbf{A}_I$-associated quotient polytope P_I is k-neighborly, we call P is locally k-neighborly.

There are two justifications for supporting local neighborliness in SRC:

First, local neighborliness implies that the support vectors (corresponding to nonzeros) are collected from a subset of columns of the dictionary $\mathbf{A}$. For classification purposes, we would like a test sample to be represented by the samples of the same class. Therefore, it is reasonable to choose the subset I as the set of the training samples of the same class.

Second, the local quotient polytope P_I is more likely to be k-neighborly if the subset I is composed of training samples of the same class. This is because columns of $\mathbf{A}_I$ are independent and identically-distributed random vectors since they are sampled from one class. From Theorem 4.2 in [37], we know P_I have a large probability to be k-neighborly.

Based on the above analysis, for the dictionary $\mathbf{A}=[\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_c]$ composed of all training samples of c classes in SRC, we can split its associated quotient polytope P into c small local ones, $P_1,P_2,\ldots,P_c$, which are, respectively, associated with $\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_c$, where $\mathbf{A}_i$ is the matrix composed of the training samples of Class i, $i=1,\ldots,c$. If every P_i is k-neighborly, for a given testing sample $\mathbf{y}$, the L_1-optimizer can recover the local sparsest solution $\mathbf{w}^i$ from the L_0–L_1 Equivalence. That is, L_1-optimizer can achieve the *sparsity* locally. Based on the set of c local sparsest solutions $\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^c$, we can get the global L_1-optimal solution $\mathbf{w}_1=\arg\min_i\|\mathbf{w}^i\|_1$, which is not necessarily the global sparest solution $\mathbf{w}_0=\arg\min_i\|\mathbf{w}^i\|_0$, but the sparse one which yields the set of support training samples closest to the given test sample. That is, L_1-optimizer can achieve the closeness globally.

Looking back at the numerical classification example in Section 3, the quotient polytope P associated with the dictionary $\mathbf{A}=[\mathbf{Y}_1,\ldots,\mathbf{Y}_{M_1},\mathbf{X}_1,\ldots,\mathbf{X}_{M_2}]$ is not k-neighborly ($k=2$ here). The reason is that the representation of the given sample point of "8", Y, is not unique. Thus, Y must be interior point of the quotient

polytope P from Lemma 1. Note that $\mathbf{Y}=\mathbf{A}\hat{\mathbf{w}}_1$ from Eq. (4), where the point $\hat{\mathbf{w}}_1$ is on a face of the cross-polytope C. However, the image of $\hat{\mathbf{w}}_1$, Y, is not on a face of the quotient polytope P. From Lemma 2, we know that $\mathbf{A}$ is not k-neighborly ($k=2$). Let us divide P into two parts, P_1 and P_2, which are quotient polytopes corresponding to $\mathbf{A}_1=[\mathbf{Y}_1,\ldots,\mathbf{Y}_{M_1}]$ and $\mathbf{A}_2=[\mathbf{X}_1,\ldots,\mathbf{X}_{M_2}]$, respectively. If P_1 and P_2 are both k-neighborly, we can use L_1-optimizer to recover the local sparsest solutions $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_0$ for $\mathbf{A}_1$ and $\mathbf{A}_2$. Then, the global L_1-solution is $\hat{\mathbf{w}}_1$, which is the solution yielding the closest support training samples to Y, but not the global sparsest solution $\hat{\mathbf{w}}_0$.

In summary, the phenomenon of local neighborliness may commonly occur in pattern recognition problems. In such a case, locally, L_1-optimizer achieves the same solution with L_0-optimizer, but globally, the solution of L_1-optimizer might not be that of L_0-optimizer. L_1-optimizer achieves sparsity locally and closeness globally. The local sparsity implies that the global L_1-solution is a local sparse solution but not necessarily the globally sparsest. The global closeness means that the global L_1-solution is a solution most meaningful for classification.

5.2.3. Achieving closeness still beyond neighborliness

In real world pattern recognition problems, however, we cannot even guarantee the local neighborliness. Specifically, we cannot ensure each local quotient polytope P associated with class training sample matrix $\mathbf{A}_i$ to be k-neighborly. Further, even if local neighborliness can be guaranteed, the sparsity level k is strictly limited. Corollary 1 shows the upper bound of k is $[(d+1)/3]$ for neighborliness. This means that for a solution of L_0 problem with more than $[(d+1)/3]$ nonzeros, L_1-optimizer may fail to recover this solution.

This above fact is somewhat disappointing, from the viewpoint of sparsity recovery. However, from the viewpoint of classification, this limitation of L_1-optimizer is insignificant. The L_1 solution is classification meaningful, even if it is not as sparse as expected. This classification meaningfulness is due to the closeness, an inherent characteristic of the solution, i.e., searching for supporting training samples which are closest to the given test sample in the sense of L_1-norm.

6. Class L_1-optimizer classifier

6.1. Justifications for local class based classification

Wright et al.'s SRC method represents a testing sample across all training samples, thus can be called Global L_1-optimizer classifier. Here, we will present two local class L_1-optimizer classifiers, which represent a testing sample by training samples belonging to every class. Three justifications for supporting the class training samples based representation are given below.

First, based on the analysis in Section 5.2, we know that the training sample matrix $\mathbf{A}$ associated quotient polytope P is more likely to be locally k-neighborly. Specifically, the *class training sample matrix* $\mathbf{A}_l$ associated *class quotient polytope* P_l is more likely to be k-neighborly because its columns are sampled from one class thus they are apt to be independent and identically-distributed random vectors. The local neighborliness supports class training samples based representation from the sparsity point of view.

Second, the geometric meaning becomes clearer if the class training samples based representation is adopted. Obviously, the training samples of a class lie on the associated *class quotient polytope* P_l. For a given testing sample $\mathbf{y}$, finding the support training samples of the class to represent it is geometrically equivalent to finding a face of P_l such that using its all vertices to represent $\mathbf{y}$ leads to the minimal representation coefficients in the sense of L_1-norm (i.e., $\|\mathbf{w}\|_1$). If P_l is k-neighborly, this representation is the sparest. Based on the analysis in Section 4, we know that $\|\mathbf{w}\|_1$ is actually the L_1-distance from $\mathbf{y}$ to the centroid of all vertices of the face. Therefore $\|\mathbf{w}\|_1$ determines a geometric distance from $\mathbf{y}$ to the class. Based on the sample-to-class distances, we can classify the sample to the closest class (i.e., the class with minimal distance). The above geometric interpretation supports class training samples based representation from the *closeness* point of view.

Third, when the training sample size is very large, the Global L_1-optimizer classifier (GL_1C) encounters a large-scale L_1 optimization problem. The Class L_1-optimizer classifier (CL_1C) means that we can solve the problem instead by dividing the large-scale problem into c (the number of classes) relative small-scale problems. Therefore, CL_1C has the advantage of dealing with large-scale problems over GL_1C from the computational point of view.

6.2. Class L_1-optimizer classifier with the closeness rule

Assume there are enough training samples per class (i.e., Assumption 1 holds) and $\mathbf{A}_i$ is the matrix formed by the training samples of Class i. For a given testing sample $\mathbf{y}$, we use the training samples of Class i to represent it and obtain the representation coefficients by solving the following problem:

$$(L_1)\ \mathbf{w}^i=\arg\min\|\mathbf{w}\|_1,\ \text{subject to}\ \mathbf{A}_i\mathbf{w}=\mathbf{y}. \tag{14}$$

After getting all representation coefficient vectors $\mathbf{w}^1,\ldots,\mathbf{w}^c$ corresponding to all classes, we use the *closeness* (i.e., the L_1-norm of the representation coefficients) as a criterion to yield a decision rule: if $\mathbf{w}^l=\min_i\|\mathbf{w}^i\|_1$, then $\mathbf{y}$ belongs to Class l. This is the original version of the closeness rule based Class L_1-optimizer classifier (C-CL_1C).

We now use the geometric interpretation given in Section 4 to further refine the original C-CL_1C. To this end, we enforce the sum-to-one constraint $\mathbf{1}^T\mathbf{w}=1$ to the L_1-optimizer in Eq. (14) and have the constrained L_1-optimizer

$$(\text{Constrained}\,L_1)\ \mathbf{w}^i=\arg\min\|\mathbf{w}\|_1,\ \text{subject to}\ \overline{\mathbf{A}}_i\mathbf{w}=\overline{\mathbf{y}}, \tag{15}$$

where

$$\overline{\mathbf{A}}=\begin{bmatrix}\mathbf{A}\\ \mathbf{1}^T\end{bmatrix},\ \overline{\mathbf{y}}=\begin{bmatrix}\mathbf{y}\\ 1\end{bmatrix},$$

by solving the problem, we obtain the representation coefficient vector $\mathbf{w}^i$ and the corresponding support training samples of Class i. Without loss of generality, assume that $\mathbf{x}_{i1},\mathbf{x}_{i2},\ldots,\mathbf{x}_{iK}$ are support training samples. We use the mean of these support training samples, $\overline{\mathbf{x}}_i=1/K\sum_{j=1}^{K}\mathbf{x}_{ij}$, as the origin to center the data locally and then use $\mathbf{x}_{i1}-\overline{\mathbf{x}}_i,\ldots,\mathbf{x}_{iK}-\overline{\mathbf{x}}_i$ as axes to form the local coordinate system. To make the coordinate of a point meaningful, we need to normalize the axes to be unitary vectors. Note that the representation coefficient vector $\mathbf{w}^i$ is calculated based on the original axes $\mathbf{x}_{i1}-\overline{\mathbf{x}}_i,\ldots,\mathbf{x}_{iK}-\overline{\mathbf{x}}_i$. So, the normalized representation coefficient vector $\overline{\mathbf{w}}^i$ based on the normalized axes is

$$\overline{\mathbf{w}}^i=[\mathbf{w}_1^i\|\mathbf{x}_{i1}-\overline{\mathbf{x}}_i\|,\ldots,\mathbf{w}_K^i\|(\mathbf{x}_{iK}-\overline{\mathbf{x}}_i)\|,\ 0,\ldots,0]^T, \tag{16}$$

$\|\overline{\mathbf{w}}^i\|_1$ is geometrically the L_1-distance from $\mathbf{y}$ to the origin in the local coordinate system of Class i. This distance leads to a classification rule: if $\mathbf{w}^l=\min_i\|\overline{\mathbf{w}}^i\|_1$, then $\mathbf{y}$ belongs to Class l. We call this the *normalized closeness rule* based class L_1-optimizer classifier (NC-CL_1C).

We finally provide the geometric interpretation for the decision rule of NC-CL_1C. For every class quotient polytope, we seek a face of it on which the test sample y may lie, noticing that the vertices of the face are determined by the solution of Eq. (15).

Then, to determine which polytope the test sample belongs to, we compare the L_1 distances from the test sample to the centroids of the faces of class polytopes, i.e., the magnitude of $\|\overline{\mathbf{w}}^i\|_1$. We know that a face of a class polytope is a convex hull of its vertices. The smaller $\|\overline{\mathbf{w}}^i\|_1$ is, the larger possibility the test sample belongs to the convex hull (face) of the class polytope.

6.3. Class L_1-optimizer classifier with the Lasso rule

From the geometric interpretation, we know C-CL_1C (or NC-CL_1C) restrict the testing sample to lie on faces of the class polytopes, as shown in Fig. 2. This restriction is generally too strict and even infeasible when there are not enough training sample per class. Here we remove this restriction and allow the testing sample point not on faces of the class polytopes, as shown in Fig. 4. We seek the face of a class polytope that is nearest to the given testing sample. This nearness between a testing sample and a face can be measured by two criterions. One is the *residual criterion*, which characterizes the distance between the test sample point and its image (reconstruction point) on the face, and the other is the *closeness criterion* (L_1-norm of the reconstruction coefficients), which characterizes the distance between the image of the test sample point and the centroid of the face. These two criterions can be integrated into the Lasso criterion [12] as follows:

$$\text{(Lasso) } \mathbf{w}^i = \arg\min L(\mathbf{w}) = \|\mathbf{A}_i\mathbf{w}-\mathbf{y}\|_2^2+\lambda\|\mathbf{w}\|_1, \tag{17}$$

where λ is a non-negative parameter. Obviously, if we restrict the testing sample to lie on faces of the class polytopes, i.e., $\mathbf{A}_i\mathbf{w}=\mathbf{y}$, the Lasso criterion becomes the closeness criterion.

Solving the Lasso and obtaining the representation coefficient vector $\mathbf{w}^i$ corresponding to Class i, $i=1,\ldots,c$, we use the Lasso function $L(\mathbf{w})$ as a measure to yield the decision rule: if $L(\mathbf{w}^l)=\min_i L(\mathbf{w}^i)$, then $\mathbf{y}$ belongs to Class l. This forms the original version of the Lasso rule based Class L_1-optimizer classifier (L-CL_1C).

Now, we consider how to refine the Lasso rule and embed the sum-to-one constraint $\mathbf{1}^T\mathbf{w}=1$ to it. Let

$$\overline{\mathbf{A}}=\begin{bmatrix}\mathbf{A}\\ \alpha\mathbf{1}^T\end{bmatrix},\ \overline{\mathbf{y}}=\begin{bmatrix}\mathbf{y}\\ \alpha\end{bmatrix},$$

where $\alpha>0$. The constrained Lasso criterion is defined by

$$\text{(Constrained Lasso) } \mathbf{w}^i = \arg\min\|\overline{\mathbf{A}}_i\mathbf{w}-\overline{\mathbf{y}}\|_2^2+\lambda\|\mathbf{w}\|_1. \tag{18}$$

Since $\|\overline{\mathbf{A}}_i\mathbf{w}-\overline{\mathbf{y}}\|_2^2=\|\mathbf{A}_i\mathbf{w}-\mathbf{y}\|_2^2+\alpha^2\|\mathbf{1}^T\mathbf{w}-1\|_2^2$, if we set α large enough, the solution of the constrained Lasso naturally satisfies $\mathbf{1}^T\mathbf{w}=1$. Based on the solution of Eq. (18) and the determined support training samples, we further obtain the normalized representation coefficient vector $\overline{\mathbf{w}}^i$, as shown in Eq. (16). Then, the normalized Lasso distance is defined by

$$\overline{L}(\mathbf{w}^i)=\|\mathbf{A}_i\mathbf{w}^i-\mathbf{y}\|_2^2+\lambda\|\overline{\mathbf{w}}^i\|_1. \tag{19}$$

The geometric meaning of the normalized Lasso distance is shown in Fig. 4.

The normalized Lasso distance leads to a decision rule: if $\overline{L}(\mathbf{w}^l)=\min_i \overline{L}(\mathbf{w}^i)$, then $\mathbf{y}$ belongs to Class l. We call this the *normalized* Lasso rule based class L_1-optimizer classifier (NL-CL_1C).

Finally, we would like to explain why we use the Lasso criterion in our classification model from the regularization point of view. The Lasso criterion is the sum of two terms: the first is *the square reconstruction residual term*, and the second term is a L_1 *regularization term* which is introduced to avoid overfitting. If the regularization term is neglected, to minimize the Lasso criterion leads to a standard least-square regression problem. Its solution is geometrically the distance from the point $\mathbf{y}$ to the hyperplane spanned by the training samples [39]. However, sometimes this distance is unreliable and leads to misclassification. For example, in the two-class case where there are two training sample points per class, as shown in Fig. 5, the two points span a line provided that the sum-to-one constraint is enforced. A test sample x, which belongs to Class 1, is misclassified because it is closer to the line spanned by training samples of Class 2. In such a case, we notice that the reconstruction weights are very large because the image of x, x', is far away from the centroid of y_1 and y_2. Adding the regularization term and minimizing the regularized distance can pull the image of x closer to the centroid of y_1 and y_2, and therefore rectify the distance between the testing sample x and its image. The rectified distance gives rise to a correct classification.

Here, the role of L_1 regularization is twofold. First, it results in a sparse solution which produces a local characterization for a given testing sample. This solution determines a small number of support training samples, which forms a local "patch" of the class manifold. Particularly when the class training sample matrix $\mathbf{A}_i$ associated quotient polytope P_i is t-neighborly, the local patch forms a face of the polytope P_i. Second, it helps rectify the distance between the testing sample and the face spanned by the support training samples. Actually, the regularization term itself also provides a meaningful distance between the image of the testing sample and the centroid of the face. The two distances are integrated into the Lasso distance, which provides a robust measure between the testing sample and the class manifold.

It should be mentioned that L_2 regularization can also help rectify the distance between the testing sample and the hyperplane spanned by the support training samples [35]. But, it cannot give rise to a local characterization. Specifically, if we use the L_2 regularization term instead in Eq. (17) or (18), the solution of the model is dense. To obtain a local measure, one may appeal to the K nearest neighbor searching, which results in a series of local classification methods such as the nearest neighbor line ($K=2$) [40], the nearest neighbor plane ($K=3$) [41] and the K-local hyperplane [35]. However, how to choose the proper parameter K for these kinds of methods is a difficult problem. Generally we choose a common K for every class. This is not a good strategy since

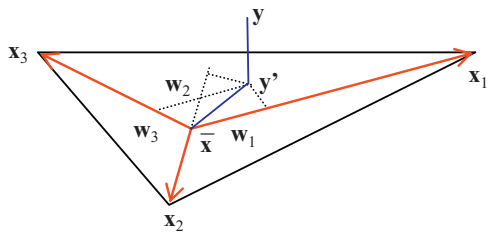

Fig. 4. The geometric meaning of the normalized Lasso distance is the weighted combination of the two distances: the L_2 distance between the test sample point $\mathbf{y}$ and its image $\mathbf{y}'$ on the face of the class polytopes, and the L_1 distance between the image $\mathbf{y}'$ and the centroid of the face.

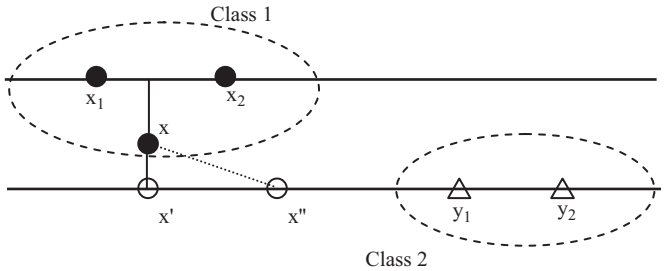

Fig. 5. An example where the minimum least-square distance leads to misclassification.

K represents the local dimension of class manifold which might be different for different local faces of different class manifold. In contrast, L_1 regularization can automatically determine the local dimension by counting nonzeros in the solution of Lasso.

7. Experiments

7.1. Experiment on the AR database for gender recognition

The AR face [42] contains over 4000 color face images of 126 people, including frontal views of faces with different facial expressions, lighting conditions and occlusions. The images of 110 persons including 55 males and 55 females are selected and used in our experiment. The pictures of each person were taken in two sessions (separated by two weeks) and each section contains 7 color images without occlusions. The face portion of each image is manually cropped and then normalized to 50×45 pixels. The sample images of one male and female are shown in Fig. 6.

In our experiment, images of the first 25 males and 25 females were used for training, and images of the remaining 30 males and 30 females for testing. Since there are 14 images per person, the total number of training samples is 700 (each class with 350 samples). We use PCA to reduce the dimension of each image to be D, where D varies from 10 to 100 with an interval of 10. In the D-dimensional PCA-transformed space, SRC [21] and the proposed class L_1-optimizer classifiers (CL_1C) including the closeness rule based CL_1C (C-CL_1C), the normalized closeness rule based CL_1C (NC-CL_1C), the Lasso rule based CL_1C (L-CL_1C) and the normalized Lasso rule based CL_1C (NL-CL_1C) are employed for classification. The nearest neighbor classifier is also used to provide a baseline. Note that here in SRC, C-CL_1C and NC-CL_1C, the matlab function "l1eq_pd" from the l_1-magic [43] is used to calculate the sparse representation coefficients. In L-CL_1C and NL-CL_1C, the matlab function "l1_ls" provided by Kim et al. [44] is used. The parameter λ in Lasso is chosen as 0.05 in L-CL_1C and 0.01 in NL-CL_1C. The recognition rate curve of each classifier versus the variation of dimensions is shown in Fig. 7. The maximal recognition rate of each classifier and the corresponding dimension are listed in Table 1.

From Fig. 7 and Table 1, we can see that the proposed class L_1-optimizer classifiers, C-CL_1C and L-CL_1C, improve the performance of the global SRC. The normalized class L_1-optimizer classifiers NC-CL_1C and NL-CL_1C can further improve the performance. The two classifiers consistently outperform the NN classifier and SRC, irrespective of the variation of dimensions. SRC does not perform well on this database, even worse than the NN classifier. In addition, we notice that the Lasso rule based class L_1-optimizer classifier L-CL_1C improve the performance of the closeness rule based class L_1-optimizer classifier C-CL_1C. However, their performance difference becomes insignificant after normalization: NC-CL_1C achieve comparable results with NL-CL_1C in this experiment.

7.2. Experiment on the CENPARMI database for handwritten numeral recognition

The experiment was done on Concordia University CENPARMI handwritten numeral database. The database contains 6000 samples of 10 numeral classes (each class has 600 samples). Some samples of "0" from the CENPARMI database are shown in Fig. 8.

In our first experiment, we choose the first 200 samples of each class for training, the remaining 400 samples for testing. Thus, the total number of training samples is 2000 while the total number of testing samples is 4000. PCA is used to transform the original 121-dimensional Legendre moment features [45] into D-dimensional features, where D varies from 10 to 80 with an interval of 10. Based on the PCA-transformed features, the nearest neighbor classifier, SRC, C-CL_1C, NC-CL_1C, L-CL_1C and NL-CL_1C are employed for classification. The parameter λ is chosen as 0.01 in L-CL_1C and NL-CL_1C. The recognition rate each classifier corresponding to the variation of dimensions is shown in Fig. 9.

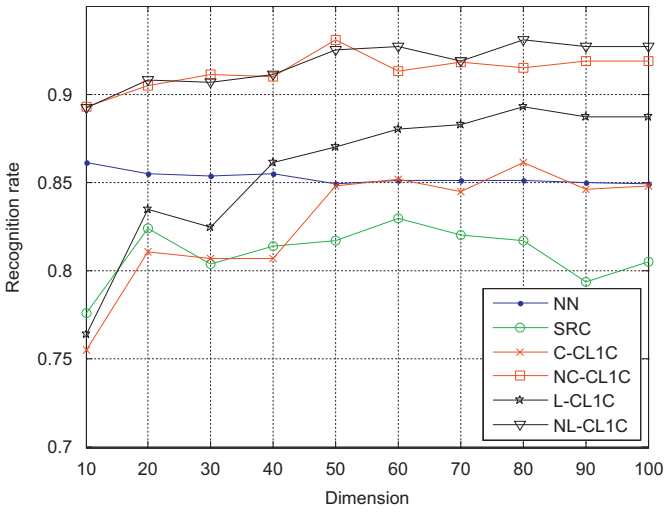

Fig. 7. The recognition rate of each classifier for gender recognition on the AR database versus the variation of dimensions.

Table 1
The maximal recognition rates (%) of each classifier for gender recognition on the AR database and the corresponding dimensions.

Classifier	NN	SRC	C-CL_1C	NC-CL_1C	L-CL_1C	NL-CL_1C
Recognition rate	86.1	83.0	86.1	93.1	89.3	93.1
Dimension	10	60	80	50	80	80

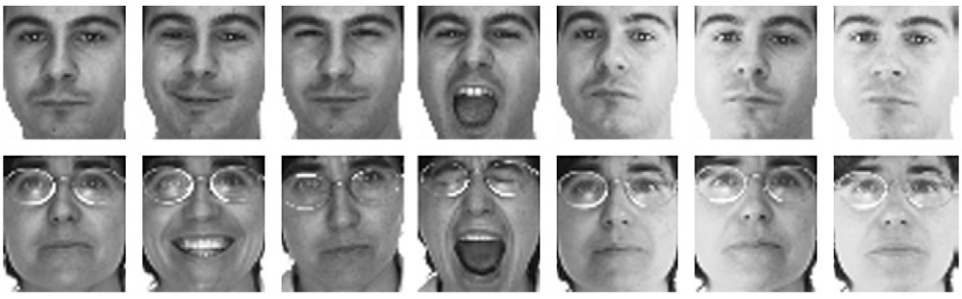

Fig. 6. Samples images of one male and female in the AR database.

Fig. 9 shows that the Lasso rule based class L_1-optimizer classifiers (L-CL_1C and NL-CL_1C) consistently outperform the closeness rule based class L_1-optimizer classifiers (C-CL_1C and NC-CL_1C) and SRC, irrespective of the variation of dimensions. This means that removing the restriction of the testing sample point on faces of the class manifold helps improve the classification performance. All of the five classifiers achieve (or nearly achieve) their maximal performance when the dimension reaches 50. As the dimension becomes larger, the performance of C-CL_1C, NC-CL_1C and SRC begins to decline, while the performance of L-CL_1C and NL-CL_1C keeps invariant or slightly increasing. This implies that the Lasso rule based class L_1-optimizer classifiers are more robust to dimensional variations than the closeness rule based class L_1-optimizer classifiers and SRC.

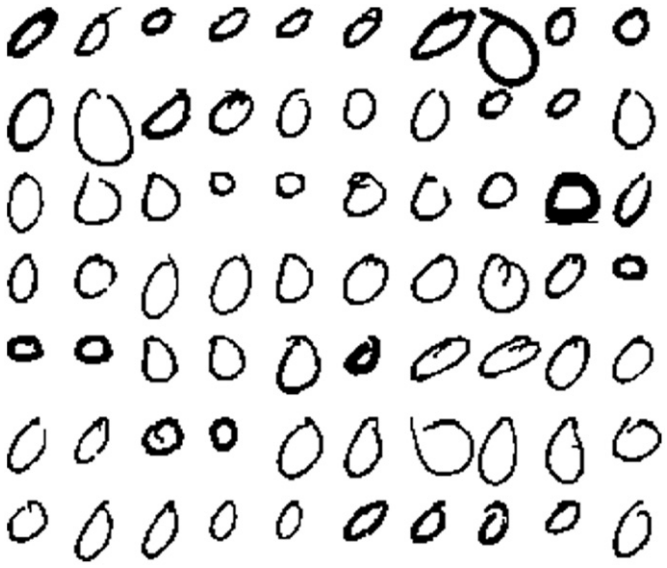

Fig. 8. Some samples in CENPARMI database.

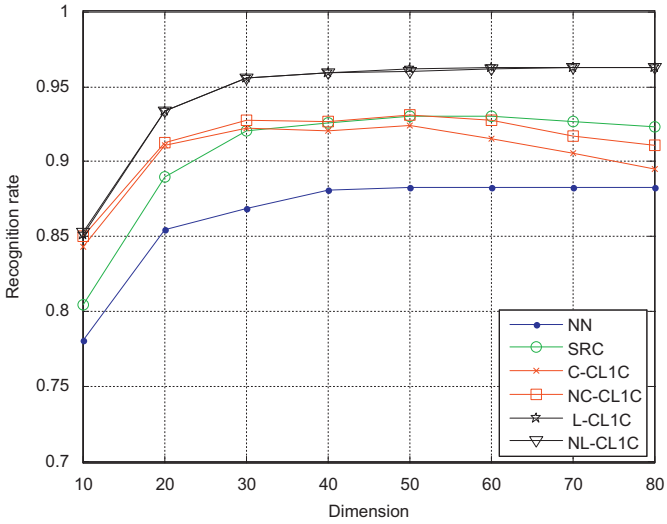

Fig. 9. The recognition rate of each classifier for handwritten numeral recognition on the CENPARMI database versus the variation of dimensions.

The recognition rate of each classifier as the dimension is 50 and the corresponding total CPU time (CPU: 2.33 GHz, RAM: 2.48 GB) are listed in Table 2. Table 2 shows that NC-CL_1C achieves comparable results with SRC, but the former is much faster than the later. The total CPU time of NC-CL_1C is only 1/7 of that of SRC, noticing that both classifiers use the same Matlab function (i.e., "l1eq_pd" from the l_1-magic [43]) to calculate the sparse representation coefficients.

To provide more insights into the Lasso rule based class L_1-optimizer classifiers, we would like to observe the representation coefficient vector of each testing sample with respect to each class. We calculate the number of nonzeros in the representation coefficient vector. Note that here the representation coefficient bigger than 10^{-3} is thought of as nonzero. The number of nonzeros represents the local dimension of the class manifold. The mean and standard deviation of the number of nonzeros corresponding to all testing samples across all classes are calculated and listed in Table 3. In addition, the means and standard deviations of the number of nonzeros corresponding to all testing samples via homo-class sample representation and hetero-class sample representation are respectively calculated and listed in Table 3. Table 3 shows us that the nonzero representation coefficients are quite different for different testing samples and different classes. In general, the homo-class representation of a testing sample yields much less nonzero representation coefficients than the hetero-class representation in the average sense. From classification point of view, for a given testing sample and a class, we find a local face of the class manifold that is closest to the sample. The dimension of the local face is generally different for different testing samples. The Lasso criterion, due to its L_1 regularization term, provides a mechanism to evaluate the local dimension adaptively.

The K-local hyperplane classifier [35] uses the L_2 regularization rather than the L_1 regularization. It does not have the ability to evaluate the local dimension K of the class manifold automatically. To address this problem, we generally assume local dimension is identical and determine a proper K by experiments. This K is obviously not theoretically optimal. Fig. 10 shows the recognition rate curse of the K-local hyperplane classifier with the variation of the parameter K. The maximal recognition rate is 95.1%. This result indicates that the K-local hyperplane classifier is effective, but not as good as the Lasso rule based classifiers L-CL_1C and NL-CL_1C. This is understandable since a common K for all testing samples and all classes is suboptimal. In addition, we find that when $K=31$, the K-local hyperplane classifier achieves a recognition rate of 94.9%, which is very close to the maximal recognition rate. Notice that 31 is the approximate local dimension estimated by the average number of nonzeros in the solution of Lasso (as shown in Table 3).

Table 3
The mean and standard deviation (std) of the number of nonzero representation coefficients.

Homo-class	Hetero-class	All
19.7568 ± 5.5557	32.2919 ± 8.2891	31.0384 ± 8.8919

Table 2
The recognition rates (%) of each classifier for handwritten numeral recognition on the CENPARMI database and the corresponding total CPU time.

Classifier	NN	SRC	C-CL_1C	NC-CL_1C	L-CL_1C	NL-CL_1C
Recognition rate	88.3	93.0	92.4	93.1	96.2	96.0
CPU time (s)	8.92×10^1	7.38×10^3	1.05×10^3	1.03×10^3	4.29×10^3	5.60×10^3

Finally, we let the number of training samples per class vary from 100 to 500 with an interval of 100, and use the remaining samples for test in the experiment. The recognition rate and the average CPU time (s) consumed for one test sample of each classifier is illustrated in Fig. 11. Fig. 11(a) shows that the Lasso rule based class L_1-optimizer classifiers (L-CL_1C or NL-CL_1C) achieve best recognition rate among all methods, irrespective of the variation of training sample size. NC-CL_1C consistently outperforms C-CL_1C, which implies that normalization does help improve the performance of the closeness rule based classifier. NC-CL_1C achieves very close results to those of SRC when the number of training samples per class is over 200. However, when the number of training samples per class is not enough, NC-CL_1C and C-CL_1C do not perform well. For instance, when the class training sample size is 100, as shown in Fig. 11, both methods achieve lower recognition rate than SRC.

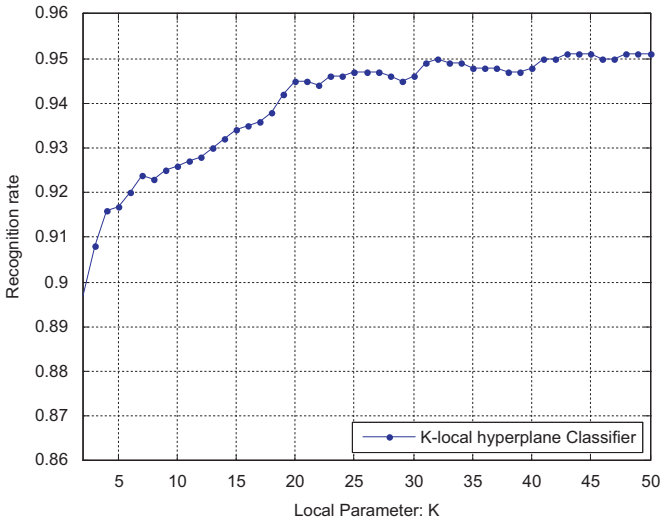

Fig. 10. The recognition rate curve of the *K*-local hyperplane classifier versus the variation of *K*-neighbor parameter *K*.

Fig. 11(b) shows that the closeness rule based classifiers (C-CL_1C and NC-CL_1C) are the fastest among all classifiers. The CPU time difference between SRC and all CL_1C classifiers (including C-CL_1C, NC-CL_1C, L-CL_1C and NL-CL_1C) become more and more significant with the increase of training sample size. SRC and C-CL_1C (or NC-CL_1C) both use the same matlab function "l1eq_pd" from the l_1-magic [43] to calculate the sparse representation coefficients. This means that the local class L_1-optimizer classifiers have computational advantage over the global L_1-optimizer classifier SRC.

7.3. Experiment on the NUST603 database for handwritten Chinese character recognition

The experiment was performed on the NUST603 handwritten Chinese character database which was built in Nanjing University of Science and Technology. The database contains 19 groups of Chinese characters that are collected from bank checks, each group with 400 samples. Some images from the NUST603HW database are shown in Fig. 12.

In our experiment, we let the number of training samples per class vary from 100 to 300 with an interval of 50, and use the remaining samples for test. Similar to the experimental methodology adopted in Section 7.2, PCA is used to transform the original 128-dimensionl peripheral feature vectors [46] into 50-dimensional features. Based on the PCA-transformed features, the nearest neighbor classifier, SRC, C-CL_1C, NC-CL_1C, L-CL_1C and NL-CL_1C are employed for classification. The parameter λ is chosen as 0.05 in L-CL_1C and NL-CL_1C. The recognition rate and the average CPU time curves of each classifier are illustrated in Fig. 13.

It is evident that here we achieve consistent results with the last experiment in Section 7.2. The Lasso rule, which combines the residual criterion and the closeness criterion, demonstrates its advantage again. L-CL_1C and NL-CL_1C consistently outperforms other classifiers, irrespective of the variation of training sample size. NC-CL_1C achieves similar (or better) results as SRC when the number of training samples per class is over 150, but as the number of training samples per class is 100, SRC performs better. This result shows again that NC-CL_1C need enough training samples to guarantee its performance. The local class L_1-optimizer classifiers, including

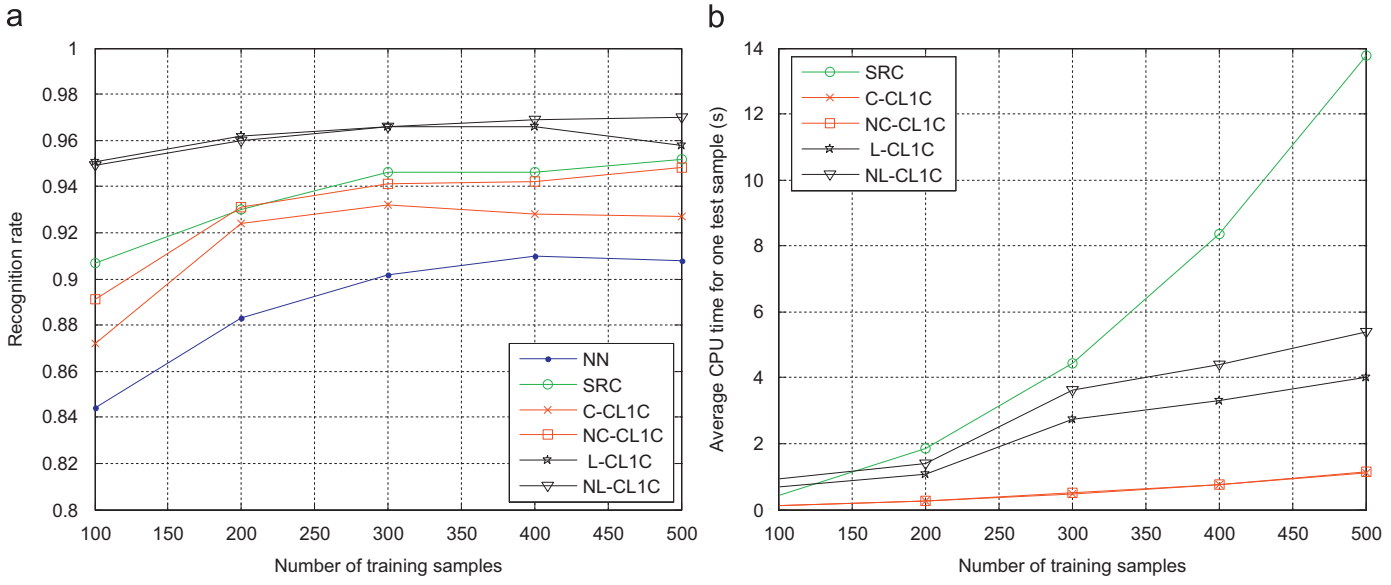

Fig. 11. Performance and speed comparison on the CENPARMI database. (a) The recognition rate of each method corresponds to the number of class training samples that varies from 100 to 500 with an interval of 100; (b) the average CPU time consumed for one test sample (s) corresponds to the number of class training samples that varies from 100 to 500 with an interval of 100.

Fig. 12. Some samples in NUST603HW database.

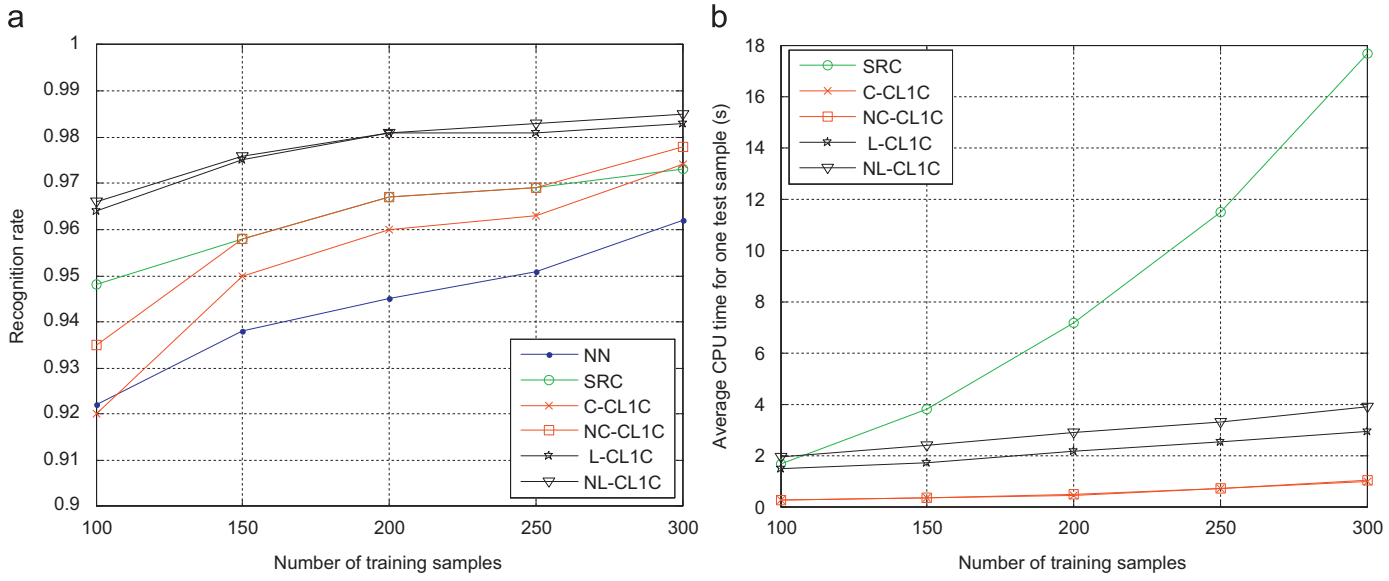

Fig. 13. Performance and speed comparison on the NUST603 database. (a) The recognition rate of each method corresponds to the number of class training samples that varies from 100 to 300 with an interval of 50; (b) the average CPU time consumed for one test sample (s) corresponds to the number of class training samples that varies from 100 to 300 with an interval of 50.

Fig. 14. Samples of a person under different illuminations in the extended Yale B face database.

C-CL_1C, NC-CL_1C, L-CL_1C and NL-CL_1C, still have their speed advantage over the global L_1-optimizer classifier SRC. This advantage becomes more and more evident with the increase of the training sample size. C-CL_1C (or NC-CL_1C) is always faster than L-CL_1C (or NL-CL_1C), because the algorithm used by the former [43] is computationally more efficient than the one used by the latter [44].

7.4. Experiment on the Extended Yale B database for face recognition

The Extended Yale B face database [47,48] contains 38 human subjects under 9 poses and 64 illumination conditions. The 64 images of a subject in a particular pose are acquired at camera frame rate of 30 frames/s, so there is only small change in head pose and facial expression for those 64 images. All frontal-face images marked with P00 are used in our experiment, and each is resized to 42×48 pixels. Some sample images of one person are shown in Fig. 14.

In the first experiment, we use histogram equalization as a preprocessing step to alleviate the effect of illuminations on images. The first 32 images of each subject are used for training, and the remaining for test. PCA (Eigenfaces [49]), LDA (Fisherfaces [50]) and LPP (Laplacianfaces [51]) are used to extract 100-dimensional features. To avoid overfitting, we perform LDA and LPP in the 200-dimensional PCA-transformed space. Finally, the nearest

Table 4
The recognition rates (%) of three classifiers with respect to three feature extraction methods for face recognition on the Extended Yale B database with histogram equalization.

Classifier	NN	SRC	L-CL_1C	NL-CL_1C
PCA	95.9	98.4	98.4	98.5
LDA	98.9	98.8	99.0	99.0
LPP	96.4	98.7	98.4	98.5

Table 5
The recognition rates (%) of three classifiers with respect to three feature extraction methods for face recognition on the Extended Yale B database without histogram equalization.

Classifier	NN	SRC	L-CL_1C	NL-CL_1C
PCA	85.8	94.7	95.1	95.1
LDA	95.9	94.3	95.0	94.8
LPP	89.8	94.2	94.1	94.1

neighbor classifier, SRC, L-CL_1C and NL-CL_1C are employed for classification. The parameter λ is chosen as 0.01 in L-CL_1C and NL-CL_1C. Note that the closeness rule based class L_1-optimizer classifiers C-CL_1C and NC-CL_1C are inapplicable here because the number of training samples per class is too small, in contrast to the dimension of feature vectors. The recognition rate of each classifier for three feature extraction methods are listed in Table 4. Table 4 shows that the proposed classifiers L-CL_1C and NL-CL_1C achieve similar recognition results with SRC for face recognition. But, the former ones are more than 5 times faster than the latter.

In the second experiment, we remove the histogram equalization step and just normalize image vectors to be unit vectors in preprocessing. Obviously, in this case, the face recognition problem becomes more challenging. We use the same experimental procedure as above to test the four classifiers for three feature extraction methods. The results are shown in Table 5. We can see that the performance of the NN classifier highly depends on what feature extraction method is used. It performs much worse than the other classifiers with respect to unsupervised feature extraction methods, such as PCA and LPP. Conversely, all L_1-optimizer classifiers, L-CL_1C, NL-CL_1C and SRC, are insensitive to feature extraction methods adopted.

7.5. Experiment using the PIE database for face recognition

The CMU PIE face database contains 68 subjects with over 40,000 face images [52]. Images of each person were taken across 13 different poses, under 43 different illumination conditions, and with 4 different expressions. Here we use a subset containing images of pose C05 (a nearly frontal pose) of 68 persons, each with 49 images. All images are manually aligned, cropped and resized to be 64×64 pixels [53] in our experiment.

Here, we only preprocess each image by normalizing image vectors to be unit vectors. The first 25 images of each subject are used for training, and the remaining for test. We use PCA, LDA and LPP for feature extraction and obtain 150 features for face representation. To avoid overfitting, we perform LDA and LPP in the 200-dimensional PCA-transformed space. The nearest neighbor classifier, SRC, L-CL_1C and NL-CL_1C are employed for classification. The parameter λ in L-CL_1C and NL-CL_1C is chosen as 0.05. The recognition rate of four classifiers corresponding to three feature extraction methods are listed in Table 6. The results in Table 6 are basically consistent with those in Table 5. We can see that the proposed classifiers L-CL_1C and NL-CL_1C achieve comparable results with SRC. The performance of the NN classifier highly depends on what feature extraction method is used. Its recognition rate is almost 20% lower than those of the other classifiers with respect to LPP. In contrast, the performance of all L_1-optimizer Classifiers is much more robust to the change of feature extraction methods.

Table 6
The recognition rates (%) of three classifiers with respect to three feature extraction methods for face recognition on the PIE database.

Classifier	NN	SRC	L-CL_1C	NL-CL_1C
PCA	83.6	96.1	98.0	97.9
LDA	98.3	99.3	99.1	99.1
LPP	77.9	97.4	97.3	97.4

8. Conclusions and discussions

We provide an insight into SRC and re-recognize the role of L_1-optimizer: using L_1-optimizer instead of L_0-optimizer is central for pattern classification. L_1-optimizer kills two birds with one stone: achieving sparsity[2] and closeness simultaneously in (global or local) neighborliness cases. Sparsity determines a small number support training samples to represent a given test sample, while closeness makes the nonzero representation coefficients concentrate on the homo-class training samples. Sparsity benefits for local reconstruction, while closeness helps for global similarity. By combining sparsity and closeness together, the solution of L_1-optimizer yields a geometrically meaningful measure for classification.

We propose two kinds of class L_1-optimizer classifiers (CL_1C), the closeness rule based ones (C-CL_1C and NC-CL_1C) and the Lasso rule based ones (L-CL_1C and NL-CL_1C). The former can be viewed as a special case of the latter. If the number of training sample size per class is large enough, NC-CL_1C achieve similar (or even better) performance as SRC but with much lower computational cost. So, in this case, if one cares more about the classification speed, we recommend using NC-CL_1C since it is the fastest L_1-optimizer classifier. However, in most real world pattern recognition problems, there is only limited number of training samples available, as shown in our experiments. In such general cases, we recommend using NL-CL_1C due to its robust performance, as demonstrated across all of our experiments. Besides, its speed is also acceptable; it is significantly faster than SRC when the number of training samples is relatively large.

Finally, we would like to specify some distinctions and connections between our work and Wright and Ma's work [31]. Our work focuses on the basic SRC with the standard L_1-optimizer model as shown in Eq. (2), while Wright's work [31] focuses on the general SRC with the extended L_1-optimizer model as follows: $[\hat{\mathbf{w}},\hat{\mathbf{e}}] = \arg\min \|\mathbf{w}\|_1 + \|\mathbf{e}\|_1$, subject to $\mathbf{Aw}+\mathbf{e}=\mathbf{y}$. Our work is to provide reasonable supports for L_1-optimizer based classifier: its discriminative power can be guaranteed, even if the L_1-solution could be denser than L_0-solution, whereas Wright's work is to provide theoretical justifications for error correction ability of the extended L_1-optimizer: it can work well, even if the error vector $\mathbf{e}$ is nearly dense. There is one thing in common in both works: sparsity is not a necessary condition any more: both weight vector and error vector could be dense to some degree.

[2] Note that the solution of L_1-optimizer is sparse but not necessarily the sparest solution of L_0-optimizer in local neighborliness case.

Acknowledgments

The authors would like to thank the anonymous reviewers for their critical and constructive comments and suggestions. This work was partially supported by the Program for New Century Excellent Talents in University of China, the NUST Outstanding Scholar Supporting Program, the National Science Foundation of China under Grant nos. 60973098 and 90820306, National Science Fund for Distinguished Young Scholars, and the Hong Kong RGC General Research Fund.

Appendix. Proof of Lemma 3

Proof. If $\mathrm{Spark}(\mathbf{A}) \le 2k-1$, there exists a subset of $2k$ columns from $\mathbf{A}$ which is linearly dependent. Without loss of generality, suppose that $\mathbf{a}_1,\mathbf{a}_2,\ldots,\mathbf{a}_{2k}$ are linearly dependent. Then, there exists a set of scalars $t_1,t_2,\ldots,t_{2k}$, not all zero, such that

$$t_1\mathbf{a}_1+t_2\mathbf{a}_2+\cdots+t_k\mathbf{a}_k+t_{k+1}\mathbf{a}_{k+1}+\cdots+t_{2k}\mathbf{a}_{2k}=0$$

Let $\mathbf{y}=t_1\mathbf{a}_1+t_2\mathbf{a}_2+\cdots+t_k\mathbf{a}_k=\mathbf{A}\mathbf{w}_1$. It is obvious that $\mathbf{y}$ can be expressed alternatively by $\mathbf{y}=(-t_{k+1})\mathbf{a}_{k+1}+\cdots+(-t_d)\mathbf{a}_{2k}=\mathbf{A}\mathbf{w}_2$.

Without loss of generality, we assume that the problem is scaled so that $\|\mathbf{w}_1\|_1=1$ and $\|\mathbf{w}_2\|_1=1$. Letting $\|\mathbf{w}_1\|_0=l_1$ and $\|\mathbf{w}_2\|_0=l_2$, it is evident that $l_1\le k$ and $l_2\le k$. Thus, $\mathbf{w}_1$ and $\mathbf{w}_2$ are on faces of the cross-polytope C. However, $\mathbf{y}=\mathbf{A}\mathbf{w}_1=\mathbf{A}\mathbf{w}_2$ are not on faces of the quotient polytope P from Lemma 1, since the representation is not unique. Actually, $\mathbf{y}$ is an interior point of P. From Lemma 2, it derives that P is not k-neighborly.

Therefore, if P is k-neighborly, $\mathrm{Spark}(\mathbf{A})\ge 2k$ must hold. □

References

[1] W.E. Vinje, J.L. Gallant, Sparse coding and decorrelation in primary visual cortex during natural vision, Science 287 (5456) (2000) 1273–1276.
[2] B.A. Olshausen, D.J. Field, Sparse coding of sensory inputs, Current Opinion in Neurobiology 14 (4) (2004) 481–487.
[3] T. Serre, Learning a Dictionary of Shape-Components in Visual Cortex: Comparison with Neurons, Humans and Machines, Ph.D. Dissertation, MIT, 2006.
[4] E. Candès, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Transactions on Information Theory 52 (2) (2006) 489–509 (February).
[5] E. Candès, T. Tao, Near optimal signal recovery from random projections: universal encoding strategies? IEEE Transactions on Information Theory 52 (12) (2006) 5406–5425 (December).
[6] D. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (4) (2006) 1289–1306 (April).
[7] E.J. Candes, M.B. Wakin, An introduction to compressive sampling, IEEE Signal Processing Magazine 25 (2) (2008) 21–30 (21 March).
[8] S. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Transactions on Signal Processing 41 (12) (1993) 3397–3415.
[9] M. Aharon, M. Elad, A. Bruckstein, The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation,, IEEE Transactions on Signal Processing 54 (11) (2006) 4311–4322.
[10] J. Bobin, J.-L. Starck, J. Fadili, Y. Moudden, D.L. Donoho, Morphological component analysis: an adaptative thresholding strategy, IEEE Transactions on Image Processing 16 (11) (2007) 2675–2681.
[11] K. Labusch, E. Barth, T. Martinetz, Simple method for high-performance digit recognition based on sparse coding, IEEE Transactions on Neural Networks 19 (11) (2008).
[12] H. Zhou, T. Hastie, R. Tibshirani, Sparse Principle Component Analysis, Technical Report, Statistics Department, Stanford University, 2004.
[13] A. d'Aspremont, L. El Ghaoui, M.I. Jordan, G.R.G. Lanckriet, A direct formulation for sparse PCA using semidefinite programming, Advances in Neural Information Processing Systems (NIPS), , 2004 (December).
[14] B. Moghaddam, Y. Weiss, S. Avidan, Spectral bounds for sparse PCA: exact and greedy algorithms, Advances in Neural Information Processing Systems 18 (2005).
[15] B. Moghaddam, Y. Weiss, S., Avidan, Generalized spectral bounds for sparse LDA, in: ICML '06: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 641–648.
[16] D. Cai, X. He, J. Han, Sparse projections over graph, in: Proceedings of AAAI Conference on Artificial Intelligence (AAAI-08), Chicago, Illinois, July 2008.
[17] L.S. Qiao, S.C. Chen, X.Y. Tan, Sparsity preserving projections with applications to face recognition, Pattern Recognition 43 (1) (2010) 331–341.
[18] S. Yan, H. Wang, Semi-supervised learning by sparse representation, in: Proceedings of SIAM International Conference on Data Mining (SDM 2009) Sparks, Nevada, USA, 2009.
[19] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extension: a general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.
[20] J. Wright, Y. Ma, J. Mairal, et al., Sparse representation for computer vision and pattern recognition, Proceedings of IEEE (2009) (March).
[21] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.
[22] J. Yang, J. Wright, T. Huang, Y. Ma, Image superresolution as sparse representation of raw patches, IEEE Conference on Computer Vision and Pattern Recognition (2008).
[23] J. Mairal, G. Sapiro, M. Elad., Learning multiscale sparse representations for image and video restoration, SIAM MMS 7 (1) (2008) 214–241 (April).
[24] K. Huang, S. Aviyente, Sparse representation for signal classification, Neural Information Processing Systems (2006).
[25] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Learning discriminative dictionaries for local image analysis, in: Proceedings of IEEE CVPR, 2008.
[26] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Supervised dictionary learning, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in NIPS, vol. 21, 2009.
[27] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009).
[28] E. Amaldi, V. Kann, On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems, Theoretical Computer Science 209 (1998) 237–260.
[29] D. Donoho, For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution, Communications on Pure and Applied Mathematics 59 (6) (2006) 797–829.
[30] E. Candès, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements,, Communications on Pure and Applied Mathematics 59 (8) (2006) 1207–1223.
[31] J. Wright, Y. Ma., Dense error correction via l1-minimization, IEEE Transactions on Information Theory 56 (7) (2010) 3540–3560.
[32] V. Vapnik., The Nature of Statistical Learning Theory, Springer, N.Y., 1995.
[33] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM Review 43 (1) (2001) 129–159.
[34] D. Donoho, Y. Tsaig, Fast Solution of L1 Norm Minimization Problems when the Solution May Be Sparse. 〈http://www-stat.stanford.edu/~donoho/reports.html〉, 2006.
[35] P. Vincent, Y. Bengio, K-local hyperplane and convex distance nearest neighbor algorithms, Advances in Neural Information Processing Systems (NIPS2002) (2002).
[36] D. Donoho, M. Elad, Maximal Sparsity Representation Via l1 Minimization. 〈http://www-stat.stanford.edu/~donoho/reports.html〉.
[37] D. Donoho, Neighborly Polytopes and Sparse Solution of Underdetermined Linear Equations, Technical Report 2005-4, Department. of Statistics, Stanford University, 2005.
[38] P. McMullen, G.C. Shephard, Diagrams for centrally symmetric polytopes, Mathematika 15 (1968) 123–138.
[39] J. Yang, Z. Lou, Z. Jin, J.-y. Yang, Minimal local reconstruction error measure based discriminant feature extraction and classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), Alaska, June, 2008.
[40] S.Z. Li, Juwei Lu, Face recognition using the nearest feature line method, IEEE Transactions on Neural Networks 10 (2) (1999) 439–443.
[41] J.-T. Chien, C.-C. Wu, Discriminant waveletfaces and nearest feature classifiers for face recognition, IEEE Trans. Pattern Anal, Machine Intelligence 24 (12) (2002) 1644–1649.
[42] A.M. Martinez, R. Benavente, The AR Face Database, CVC Technical Report #24, June 1998.
[43] E. Candès, J. Romberg, l1-Magic: Recovery of Sparse Signals Via Convex Programming, 〈http://www.acm.caltech.edu/l1magic/〉.
[44] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, A method for large-scale l1-regularized least squares, IEEE Journal on Selected Topics in Signal Processing 1 (4) (2007) 606–617.
[45] S.X. Liao, M. Pawlak, On image analysis by moments, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (3) (1996) 254–266.
[46] Y.H. Tseng, C.C. Kuo, H.J. Lee, Speeding up Chinese character recognition in an automatic document reading system, Pattern Recognition 31 (11) (1998) 1601–1612.
[47] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.
[48] K.C. Lee, J. Ho, D. Driegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) (2005) 684–698.

[49] M. Turk, A. Pentland, Face recognition using eigenfaces,, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (1991) 586–591.
[50] P.N. Belhumeur, J.P. Hespanha, D.J. Kriengman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.
[51] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using Laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.
[52] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression database, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (12) (2003) 1615–1618.
[53] D. Cai, X. He, J. Han, Spectral regression for efficient regularized subspace learning, in: Proceedings of IEEE 11th International Conference on Computer Vision (ICCV 2007).

Jian Yang received the B.S. degree in mathematics from the Xuzhou Normal University in 1995, the M.S. degree in applied mathematics from the Changsha Railway University in 1998 and the Ph.D. degree in computer science from the Nanjing University of Science and Technology (NUST) in 2002. He was a postdoctoral researcher at the University of Zaragoza in 2003. From 2004 to 2006, he was a Postdoctoral Fellow in Department of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow in Department of Computer Science of New Jersey Institute of Technology. From 2007, he has been a professor in the School of Computer Science and Technology of NUST. Now, he is a Visiting Associate at California Institute of Technology. He is the author of more than 50 scientific papers in pattern recognition, computer vision and the related areas. His journal papers have been cited more than 1100 times in the ISI Web of Science, and 2400 times in the Web of Scholar Google. Currently, he is an associate editor of Pattern Recognition Letters and IEEE Transactions on Neural Networks.

Lei Zhang received the B.S. degree in 1995 from Shenyang Institute of Aeronautical Engineering, Shenyang, PR China, the M.S. and Ph.D. degrees in Automatic Control Theory and Engineering from Northwestern Polytechnical University, Xi'an, PR China, respectively, in 1998 and 2001. From 2001 to 2002, he was a research associate in the Department of Computing, The Hong Kong Polytechnic University. From January 2003 to January 2006 he worked as a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, McMaster University, Canada. Since January 2006, he has been an Assistant Professor in the Department of Computing, The Hong Kong Polytechnic University. His research interests include Image and Video Processing, Biometrics, Pattern Recognition, Multisensor Data Fusion and Optimal Estimation Theory, etc.

Yong Xu received his B.S. and M.S. degrees at Air Force Institute of Meteorology (China) in 1994 and 1997, respectively. He then received his Ph.D. degree in pattern recognition and intelligence system at the Nanjing University of Science and Technology (NUST) in 2005. From May 2005 to April 2007, he worked at Shenzhen graduate school, Harbin Institute of Technology (HIT) as a postdoctoral research fellow. Now he is an associate professor at Shenzhen graduate school, HIT. He also acts as a research assistant researcher at the Hong Kong Polytechnic University from August 2007 to June 2008. His current interests include pattern recognition, biometrics and machine learning. He has published more than 40 scientific papers.

Jing-yu Yang received the B.S. Degree in Computer Science from Nanjing University of Science and Technology (NUST), Nanjing, China. From 1982 to 1984 he was a visiting scientist at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. From 1993 to 1994 he was a visiting professor at the Department of Computer Science, Missuria University. And in 1998, he acted as a visiting professor at Concordia University in Canada. He is currently a professor and Chairman in the department of Computer Science at NUST. He is the author of over 300 scientific papers in computer vision, pattern recognition and artificial intelligence. He has won more than 20 provincial awards and national awards. His current research interests are in the areas of pattern recognition, robot vision, image processing, data fusion and artificial intelligence.