Two-Stage Sparse Representation for Robust Recognition on Large-Scale Database

Ran He[1], BaoGang Hu[2], Wei-Shi Zheng[3], and YanQing Guo[1]
[1]School of Electronic and Information Engineering, Dalian University of Technology, Dalian, P.R. China
[2]National Laboratory of Pattern Recognition, Institute of Automation Chinese Academy of Sciences, Beijing, P.R. China
[3]Department of Computer Science, Queen Mary University of London, London, UK
rhe@dlut.edu.cn, bghu@nlpr.ia.ac.cn, wszheng@ieee.org, and guoyq@dlut.edu.cn

Abstract

This paper proposes a novel robust sparse representation method, called the two-stage sparse representation (TSR), for robust recognition on a large-scale database. Based on the divide and conquer strategy, TSR divides the procedure of robust recognition into outlier detection stage and recognition stage. In the first stage, a weighted linear regression is used to learn a metric in which noise and outliers in image pixels are detected. In the second stage, based on the learnt metric, the large-scale dataset is firstly filtered into a small set according to the nearest neighbor criterion. Then a sparse representation is computed by the non-negative least squares technique. The sparse solution is unique and can be optimized efficiently. The extensive numerical experiments on several public databases demonstrate that the proposed TSR approach generally obtains better classification accuracy than the state-of-the-art Sparse Representation Classification (SRC). At the same time, by using the TSR, a significant reduction of computational cost is reached by over fifty times in comparison with the SRC, which enables the TSR to be deployed more suitably for large-scale dataset.

1 Introduction

Automatically classifying image-based object has wide applications in computer vision and machine learning. An image-based recognition system compares a query image with prototypical images recorded in a database and output the category label of the image. Two major concerns in designing a recognition system are that (1) the query images are subject to changes in illumination as well as occlusion (Sanja, Skocaj, and Leonardis 2006); and (2) the number of the prototypical images is often tens of thousands. For these concerns, we have to address two basic issues: (1) how to yield a robust representation of an object and (2) how to classify a query image as efficient as possible.

Recently, the sparse representation has proven to be robust and discriminative for machine learning problems (Wright et al. 2009a). Typically, the sparse technique is casted into an l^1 minimization problem, which is an equal approximation of the l^0 minimization problem under some conditions (Candes and Tao 2005; Donoho 2006). Along this line, a sparse representation classifier (SRC) for robust face recognition was proposed in (Wright et al. 2009b) and impressive results were reported against many well-known face recognition methods(Turk and Pentland 1991; He et al. 2005). However, the sparse assumption of noise in SRC makes it computationally expensive (Wright et al. 2009b).

In order to tackle the above two basic issues in a unified framework, we propose a two-stage sparse representation (TSR) framework based on the divide and conquer strategy. The procedure of robust recognition is divided into outlier detection stage and recognition stage. In the first stage, to deal with varying illumination as well as occlusion, a robust linear regression method is proposed to learn a metric in which noise and outliers in image pixels are detected. Different from SRC which assumes that the noisy item has a sparse representation, the robust metric in TSR is derived from a robust function which is robust to non-Gaussian noise and large outliers. In the second stage, to reduce the computational cost, we filter the large-scale dataset [1] into a small subset according to the nearest neighbor criterion based on the learnt metric. Then we harness the technique of non-negative least squares to compute a sparse representation on the filtered subset to further improve recognition rate. This non-negative sparse solution is unique and can be optimized efficiently. Extensive experiments on recognition tasks corroborate above claims about robustness and sparsity, and demonstrate that the proposed TSR framework significantly reduces the computational cost and meanwhile achieve better performance as compared to SRC.

The remainder of this paper is outlined as follows: in Section 2, we begin with a brief review of the SRC. Then we present the two-stage sparse representation (TSR) framework in Section 3. Comparison results between the proposed framework and the state-of-the-art are reported in Section 4. Finally, we draw the conclusions and discuss the future work in Section 5.

2 Sparse Representation by l^1 minimization

Let $X_c \doteq [x_1^c, x_2^c, \ldots, x_{n_c}^c] \in \mathbb{R}^{d\times n_c}$ be a matrix and each column of which is the training samples of the c-th class, where $c = 1, \cdots, k$, and $y \in \mathbb{R}^{d\times 1}$ be a new test sample.

[1]Here, we specify that large scale dataset contains nearly ten thousand images.

Let x_{ij}^c and y_j be the j-th entry of x_i^c and y respectively. By concatenating the training samples of all k classes, we get a new matrix X for the entire training set as :

$$X \doteq [X_1, X_2, ..., X_k] = [x_1^1, x_2^1, ..., x_{n_k}^k] \in \mathbb{R}^{d\times n} \quad (1)$$

where $n = \sum_{c=1}^{k} n_c$. A test sample y then can be expressed as a linear combination of all training samples: $y \approx X\beta$ where the coefficient vector $\beta \in \mathbb{R}^n$.

The Sparse representation classifier (SRC) (Wright et al. 2009a; 2009b) aims to seek the sparsest solution, i.e.,

$$\min ||\beta||_0 \quad s.t. \quad y = X\beta \quad (2)$$

where the l^0-norm $||.||_0$ counts the number of nonzero entries in a vector. Originally inspired by theoretical results (Candes and Tao 2005; Donoho 2006) that the solution of the l^0 minimization problem is equal to the solution of the l^1 minimization problem if the solution is *sparse enough*, SRC seeks an approximate solution of β by solving the following convex relaxation:

$$\min ||\beta||_1 \quad s.t. \quad y = X\beta \quad (3)$$

where $||\beta||_1 = \sum_{i=}^{n} |\beta_i|$. Here, we denote the algorithm that solves Eq. (3) by ***SRC0***.

Normally, the Lasso solution (Tibshirani 1996) can be defined as an unconstraint minimizer of Eq.(3):

$$\min ||y - X\beta||^2 + \lambda||\beta||_1 \quad (4)$$

where λ can be viewed as an inverse of the Lagrange multiplier in Eq.(3). To deal with occlusions and corruptions, Wright et al. further proposed a robust linear model as:

$$y = X\beta + e \quad (5)$$

where $e \in \mathbb{R}^d$ is a vector of errors. Assuming that the noisy vector e has a sparse representation, SRC seeks the sparsest solution to the robust system of Eq.(5):

$$\min ||\beta||_1 + ||e||_1 \quad s.t. \quad y = X\beta + e \quad (6)$$

We denote the algorithm that solves Eq.(6) by ***SRC1***.

Although SRC1 can effectively deal with the occlusion and corruption problems, its computational cost is very high. It will take nearly 100 seconds for SRC1 to process a test image stacked in a 700-D vector (Wright et al. 2009b).

3 Two-stage Sparse Representation

The key point of our two-stage Sparse Representation (TSR) framework is to learn a robust metric to detect the outliers and then to harness the non-negative sparse representation to perform classification.

3.1 Learn A Robust Metric

In real-world face recognition, facial images are often corrupted by noise or outliers, that is, some pixels that do not belong to the facial images are depicted. One expects to learn a metric M by which the outliers are efficiently detected and rejected so that any classifier can work on the uncorrupted subsets of pixels of facial images. Generally, the M is assumed to be a diagonal matrix (Xing et al. 2002).

To deal with the outliers, we define the metric M as a function of a test sample y, a subspace $U \in \mathbb{R}^{m\times d}$ that models variation of the dataset X [2], and a projection coefficient $\beta \in \mathbb{R}^{m\times 1}$, i.e.,

$$M \triangleq M(U, y, \beta) \quad (7)$$

where $M_{jj} \triangleq g(y_j - \sum_{i=1}^{m} U_{ij}\beta_i)$ and $g(x) = \exp(-\frac{||x||^2}{2\sigma^2})$ is a Gaussian kernel function. We wish to find a metric that has the maximum matrix norm,

$$M^* = \arg\max_{M(y,U,\beta)} ||M(y, U, \beta)||_1 \quad (8)$$

where $||M||_1 \triangleq \sum_{i=1}^{d}\sum_{j=1}^{d} |M_{ij}|$. Then we obtain the following optimization problem,

$$M^* = \arg\max_{M(y,U,\beta)} \sum_{j=1}^{d} g(y_j - \sum_{i=1}^{m} U_{ij}\beta_i) \quad (9)$$

The optimization problem in Eq.(9) is actually a maximum Correntropy problem (Liu, Pokharel, and Principe 2007) and the kernel function $g(x)$ is a robust M-estimator (Liu, Pokharel, and Principe 2007; Yuan and Hu 2009) that can efficiently deal with non-Gaussian noise and large outliers. The problem in Eq. (9) can be optimized in an alternative maximum way (Yuan and Hu 2009; Yang et al. 2009):

$$M_{jj}^t = g(y_j - \sum_{i=1}^{m} U_{ij}\beta_i^{t-1}) \quad (10)$$

$$\beta^t = \arg\max_{\beta} (y - U\beta)^T(-M^t)(y - U\beta) \quad (11)$$

The optimization problem in Eq. (11) is a weighted linear regression problem, and its analytical solution can be directly computed by

$$\beta^t = (U^T M^t U)^{-1} U^T M^t y \quad (12)$$

Algorithm 1: Learning a robust metric

Input: Subspace U, a test visual data $y \in \mathbb{R}^{m\times 1}$, and a small positive value ε
Output: M
1: **repeat**
2: Initialize *converged* = FALSE.
3: Update M according to Eq. (10).
4: Update β according to Eq. (11).
5: **if** the entropy delta [a] is smaller than ε **then**
6: *converged* = TRUE.
7: **end if**
8: **until** *converged*==TRUE

[a]The entropy delta is the difference of the entropy objective between two iterations. In practice, we actually set the maximum iteration to 20 instead of selection of an ϵ. We find that the algorithm always sufficiently converges for 20 iterations.

[2]In this paper, the subspace U is composed of the eigen vectors computed by principal component analysis (Sanja, Skocaj, and Leonardis 2006)

Algorithm 1 outlines the optimal procedure. According to the half-quadratic optimization framework (Yuan and Hu 2009), Algorithm 1 alternatively maximizes the objective by Eq. (10) and Eq. (11) until Algorithm 1 converges. Since the outliers are significantly far away from the uncorrupted pixels, the M-estimator will punish the outliers during alternative maximum procedure. When the algorithm converges, we obtain a robust metric in which the diagonal entries corresponding to the outliers would have small values (Yuan and Hu 2009).

3.2 Non-negative Sparse Representation

In face recognition, we find that if there is no occlusion or corruption, the sparse coefficients computed by SRC method are often non-negative. Inspired by this observation, we aim to achieve a non-negative sparse representation (Vo, Moran, and Challa 2009; Ji, Lin, and Zha 2009) for robust object recognition problem.

Firstly, we filter the database into a small subset according to the nearest neighbor criterion in the learnt robust metric. We denote the number of the nearest neighbor by n_{knn} [3] and let $n_{knn} < min(n, d)$. The motivation of this filtering step is three folds: 1) it ensures that the the optimization problem in Eq. (14) has a unique solution; 2) it plays as a similar role as λ in Lasso to remove some samples corresponding to small coefficients [4]; 3) it significantly reduces the computational cost so that TSR can deal with large-scale problem. The efficiency of this filtering step will be further corroborated in the experiment section 4.4 and 4.6 . Unless otherwise stated, we set n_{knn} to 300 throughout the paper.

Secondly, a non-negative representation (Ji, Lin, and Zha 2009) is computed on the subset by imposing a non-negative constraint on Eq. (4),

$$\min ||y - X\beta||^2 + \lambda||\beta||_1 \quad s.t. \quad \beta \geq 0 \tag{13}$$

Instead of using second-order cone programming (Boyd and Vandenberghe 2004) to solve the non-negative l^1-regularized least squares problem in Eq. (13), we simply set $\lambda = 0$ (Vo, Moran, and Challa 2009). Then we have

$$\min ||y - X\beta||^2 \quad s.t. \quad \beta \geq 0 \tag{14}$$

We denote the method to solve Eq. (14) by non-negative sparse representation (NSR). If the matrix X is of full column rank ($rank(X) = n$), the matrix X^TX is positive definite and thus the strictly convex programming in Eq. (14) has a unique solution for each vector y (Bjorck 1988). In our implementation, Eq. (14) is minimized using an active set algorithm for linear programming based on (Bjorck 1988; Portugal, Judice, and Vicente 1994)

Inspired by the sparse classifier proposed in (Wright et al. 2009b), we classify a test sample y as follows. For each class c, let $\delta_c : \mathbb{R}^n \rightarrow \mathbb{R}^{n_c}$ be a function which selects the coefficients belonging to class c, i.e. $\delta_c(\beta) \in \mathbb{R}^{n_c}$ is a vector whose entries are the entries in β corresponding to class c. Utilizing only the coefficients associated to class c, the given sample y is reconstructed as $\hat{y}_c = X_c\delta_c(\beta)$. Then y is classified based on these reconstructions by assigning it to the class that minimizes the residual between y and $\hat{y}_c$:

$$\min_c r_c(y) \doteq ||y - \hat{y}_c||_2 \tag{15}$$

3.3 Algorithm of TSR

Algorithm 2 summarizes the procedure of the two-stage sparse representation (TSR). In step 1, we make use of the Algorithm 1 to learn a robust metric M. And then we set $\hat{X} = \sqrt{M}X$ and $\hat{y} = \sqrt{M}y$ so that we can perform a more robust classification in the metric space induced by M in the following steps. In step 2, to reduce the computational cost, the large-scale dataset $\hat{X}$ is filtered into a small subset $\hat{X}^1$. In step 3, a sparse representation is computed by NSR on $\hat{X}^1$. In step 4, considering that each object class often has several instances, i.e., $n_c \geq 1$, we select all instances of the most competitive classes that correspond to the nonzero coefficients in step 4. In step 5, the query image y is classified according to the computed residuals.

Algorithm 2: Two-stage sparse representation(TSR)

Input: data matrix $X = [X_1, X_2, ..., X_k] \in \mathbb{R}^{m\times n}$ for k classes, a test sample $y \in \mathbb{R}^{m\times 1}$, the number of the nearest neighbor n_{knn}

Output: $identity(y)$

1: Compute a robust diagonal metric M according to Algorithm 1, and set $\hat{X} = \sqrt{M}X$ and $\hat{y} = \sqrt{M}y$.

2: Compute a nearest subset I^1 in $\hat{X}$ to $\hat{y}$ according to the nearest neighbor criterion, and set $\hat{X}^1 = \{x_i | i \in I^1\}$.

3: Solve the non-negative least squares problem:

$$\beta^* = \arg\min_\beta ||\hat{X}^1\beta - \hat{y}|| \ s.t. \ \beta \geq 0 \tag{16}$$

4: Set $I^2 = \{i | \beta_i > 0 \ and \ i \in I^1\}$ and $\hat{X}^2 = \{\hat{X}_c | \hat{x}_i \in \hat{X}_c \ and \ i \in I^2\}$, solve the non-negative least squares problem:

$$\beta^* = \arg\min_\beta ||\hat{X}^2\beta - \hat{y}|| \ s.t. \ \beta \geq 0 \tag{17}$$

5: Compute the residuals $r_c(\hat{y}) = ||\hat{y} - \hat{X}^2\delta_c(\beta^*)||_2$, for $c = 1, \ldots, k$

6: $identity(y) = \arg\min_c r_c(\hat{y})$

---

4 Experimental Verification

In this section, the proposed method is systematically compared with the state-of-the-art methods: reconstructive and discriminative subspace method (LDAonK) (Sanja, Skocaj, and Leonardis 2006), linear regression classification (LRC) (Naseem, Togneri, and Bennamoun 2009) and sparse representation based classification (SRC) (Wright et al. 2009b). All of the algorithms were implemented in MATLAB. The experiments were performed on an AMD Quad-Core 1.80GHz Windows XP machine with 2GB memory.

[3] The n_{knn} is the k in KNN.

[4] The active set algorithm (Black and Jepson 1996; Sanja, Skocaj, and Leonardis 2006) to solve Eq. (17) selects the sample that can significantly reduce the objective step by step. The lastly selected samples often correspond to smallest coefficients and are faraway from the query y.

4.1 Experimental setting and databases

Database. Three image databases are selected to evaluate different methods. All images are converted to grayscale and the facial images are aligned by fixing the locations of two eyes. The descriptions of three databases are following:

1) *AR Database* (Martinez and Benavente 1998): The AR database consists of over 4,000 facial images of 126 subjects (70 men and 56 women). For each subject, 26 facial images are taken in two separate sessions. We select a subset of the dataset consisting of 65 male subjects and 54 female subjects. The dimension of the cropped images is 112×92.

2) *Extended Yale B Database* (Georghiades, Belhumeur, and Kriegman 2001) : The Extended Yale B database consists of 2,414 frontal-face images of 38 subjects. The cropped 192×168 face images were captured under various laboratory-controlled lighting conditions and with different facial expressions. For each subject, half of the images are randomly selected for training (i.e., about 32 images per subject), and the left half for testing.

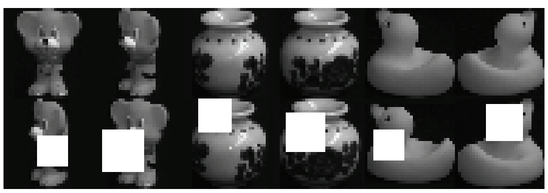


Figure 1: Some of the objects in COIL database.

3) *COIL Database (Nene, Nayar, and Murase 1996)*: Columbia Object Image Library (COIL-100) is a database consists of 7,200 color images of 100 objects (72 images per object). All images are resized to 32×32. Some of them are shown in the first row of Fig.1. For each object, 2 images are randomly selected for testing, and the left 70 images are used for training. Hence, there are 7000 images in training set.

Algorithm Setting. The details of compared techniques are:

1) *SRC*: we compare its two robust models, which are different in the aspects of robustness and computational strategy (Wright et al. 2009b).

   SRC1: the implementation minimizes the l^1-norm in Eq.(18) via a primal-dual algorithm for linear programming based on (Boyd and Vandenberghe 2004; Candes and Romberg 2005). [5]

$$\min ||\beta||_1 + ||e||_1 \quad s.t. \quad ||y - X\beta + e||_2 \leq \varepsilon \qquad (18)$$

   where ε is a given non-negative error tolerance. The algorithm's setting is the same as that in (Wright et al. 2009b).

   SRC2: the implementation minimizes the l^1-norm in Eq.(19) via an active set algorithm based on (Lee et al. 2006). [6]

$$\min ||y - X\beta + e||_2 + \lambda(||\beta||_1 + ||e||_1) \qquad (19)$$

   where λ is a given sparsity penalty. The λ is empirically set to 0.005 to achieve the best results.

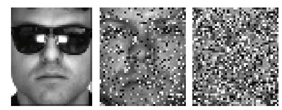
2) ***TSR***: The n_{knn} in step 2 is set to 300. The subspace U in Algorithm 1 is composed of the eigenvectors corresponding to the 10 largest eigenvalues as suggested in (Black and Jepson 1996; Sanja, Skocaj, and Leonardis 2006). The kernel size σ in Gaussian kernel is estimated by Silverman's rule (Silverman 1986):

$$(\sigma^t)^2 = 1.06 \times \min\{\sigma_E, R/1.34\} \times (d)^{-\frac{1}{5}} \qquad (20)$$

   as suggested in the Correntropy (Liu, Pokharel, and Principe 2007). The σ_E is the standard deviation and R is the interquartile range. The implementation minimizes the Eq. (14) via an active set algorithm based on (Bjorck 1988) (Portugal, Judice, and Vicente 1994). [7]

3) ***LDAonK***: The parameters of LDAonK are set by following the suggestion in (Sanja, Skocaj, and Leonardis 2006).

Note that since the computational cost of SRC1 is very high, we only give the experimental results in lower dimensional feature space.



(a) (b) (c)

Figure 2: Illustrative images used in the experiments. (a) a facial image with sunglasses in AR database. (b) 30% random pixel corruption. (c) 70% random pixel corruption.

4.2 Recognition under Disguise

For training, we use 952 images (about 8 for each subject) of un-occluded frontal views with varying facial expression. And for testing, we use images with sunglasses. Fig. 3 shows the recognition performance of different methods against different downsampled images of dimension, i.e. 161, 644 and 2576 (Wright et al. 2009b). Those numbers correspond to downsampling ratios of 1/8, 1/4 and 1/2, respectively.

We observe from the numerical simulations that the methods can be ordered in descending recognition accuracies as TSR, SRC1, SRC2, LDAonK, and LRC. Three sparse representation based methods outperform the rest two non-sparse ones. If occlusions happen, it is unlikely that the test image will be very close to the subspace of the same class, so that LRC performs poorly. In this case, TSR significantly performs better than the two SRC methods and achieves the highest recognition rates. This is because that the outlier detection stage of TSR can efficiently detect the sunglasses occlusion.

4.3 Recognition under Random Pixel Corruption

In some scenarios, the query image y could be partially corrupted. We testify the efficiency of TSR on the Extended Yale B Face Database. For each subject, half of the images are randomly selected for training, and the left half are for testing. The training and testing set contain 1205 and 1209

[5]The source code: http://www.acm.caltech.edu/l1magic/

[6]The source code: http://redwood.berkeley.edu/bruno/sparsenet/

[7]The MATLAB source code: http://www.mathworks.com/matlabcentral/ fileexchange/10908

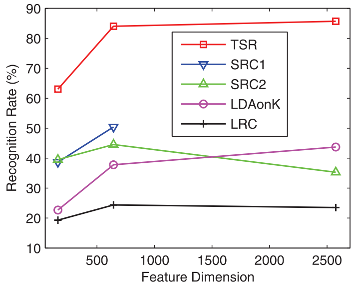


Figure 3: Recognition rates under occlusion of sunglasses.

images respectively. All images are downsampled to 48 $\times$ 42. Each test image is corrupted by replacing a percentage of randomly selected pixels with random pixel value which follows a uniform distribution over [0, 255]. The percentage of corruption is from 10% to 80%.

Fig. 4 shows the recognition accuracies of five methods under different levels of corruption. When the level of corruption trends to be high, three sparse methods begin to significantly outperform the two non-sparse ones. Recognition rates of the two SRC methods are very close. And the TSR performs slightly worse than the SRC methods. Although TSR only obtains similar recognition rates to the SRC methods, TSR can significantly reduce the computational cost.

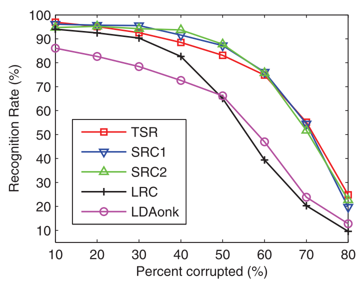


Figure 4: Recognition rates under Random Pixel Corruption.

4.4 Recognition on large-scale dataset

To evaluate the performance of TSR on large-scale database, we select the COIL dataset where there are 7000 images in the training set. We denote the TSR without the filtering steps (step 2 and step 3) by ***TSR1***, that is the TSR1 directly computes a non-negative sparse representation on the whole dataset $\hat{X}$. Since the computational costs of other three methods are very high, we only compare TSR with LRC on COIL. We evaluate the performance of different methods in two scenarios. In the first scenario, the images in testing set are the original ones in COIL database. In the second scenario, each image in the testing set is occluded by a white square whose size and location are randomly determined for each image and is unknown to the computer. Fig. 1 shows some examples of the occluded objects.

Table 1 shows the experimental results of the two scenarios. As expected, TSR still achieves the highest recognition rate. Compared with LRC, TSR can significantly improve the recognition accuracy when there is occlusion.

Table 1: Recognition rates (%) and cpu time (s) on COIL dataset.

	LRC	TSR	TSR1
original data	94.0	95.0	92.0
occluded data	23.0	84.0	80.0

We also observe that the recognition rate of TSR is higher than that of TSR1. This means that using part of the dataset can improve the recognition accuracy. We know that the classification of TSR is based on the coefficients computed by Eq.(17). Since discarding the faraway samples actually is an approximation of discarding some samples corresponding to small coefficients, the coefficients computed by TSR may be more informative and discriminant, so that TSR outperforms TSR1. Hence we consider this result as a coincidence with this phenomenon.

4.5 Computational cost

The computation complexity of algorithm 2 mainly depends on the knn filtering step. Since the computation complexity of quick sort is $O(nlog(n))$, the computation complexity of algorithm 2 is $O(nlog(n))$. Fig. 5 shows the overall computational time of using various number of features on AR database, with the same experiment setting as that in Section 4.2 . The feature dimension is 644-D. SRC1, SRC2, TSR1 and TSR take 56, 6.9, 0.43, and 0.13 seconds for each test image respectively. (In (Wright et al. 2009b), SRC1 requires about 75 seconds per test image on a PowerMac G5.) It is clear to see the computational advantage of TSR over SRC1 and SRC2. Significant reduction of computational cost is reached by TSR by over fifty times in comparison with the SRC2. Since TSR works on a subset of training data, TSR can further save more computational time as compared to TSR1.

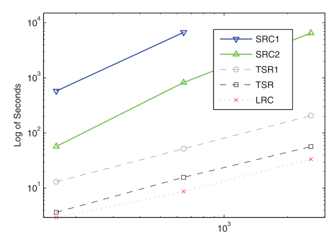


Figure 5: computational time on various feature spaces.

4.6 The Parameter n_{knn} and Sparsity

The number of the nearest neighbor n_{knn} is a key parameter to control computational cost, recognition rates and number of bits (l^0 norm) of coefficients. In this subsection, We study how the n_{knn} affects the performance of TSR. The experiment setting is the same as that in Section 4.2 and the feature dimension is 644-D.

Table 2 tabulates recognition rates and number of bits for various number of the nearest neighbor. The total number of coefficients of SRC1 and SRC2 is 40 and 25 respectively. We observe from the numerical simulations that TSR can yield a sparse representation, and both recognition rates and the total number of nonzero coefficients increase as the number of the nearest neighbor increases. It is also interesting to observe that the recognition rate at 500 is even higher that at 952. This is coincident with the results in section 4.4 . Only using part of the facial images to compute a sparse representation can yield a higher recognition rate. It also corroborates that the filtering large-scale database in step 2 is reasonable.

In addition, the experimental results demonstrate that even if a robust metric has been learned, a test image may not be very close to the same class in training set. The non-negative sparse representation is informative and discriminant for image based object recognition.

Table 2: Recognition rates (%) and the total number of nonzero coefficients (l^0 norm) for various number of n_{knn} of TSR.

	100	200	300	400	500	952
rate	84.03	84.03	86.55	87.39	88.24	86.55
l^0 norm	19	22	23	24	25	28

5 Conclusion and Future Work

Sparse representation is a powerful tool for robust object recognition. This paper divides the procedure of computing a robust sparse representation into two stages. In the first stage, a robust metric is derived from a robust function and is solved by a weighted linear regression method. In the second stage, the large-scale dataset is firstly filtered into a small subset in the learnt robust metric. Then a non-negative sparse representation method based on non-negative least squares technique is proposed to obtain a sparse representation for classification. Extensive experiments demonstrate that the proposed framework not only significantly reduces the computational cost but also can achieve better recognition performance as compared to the state-of-the-art SRC method.

Although extensive experimental observations show that without harnessing the l^1-norm technique the non-negative least squares technique can also learn a sparse representation for image-based object recognition, a theoretical investigation needs to be further done to support the sparse idea and discuss the relationship with the l^1 minimization technique in our future work.

Acknowledgments

This work was supported by DUT R & D Start-up costs and Major Program of National Natural Science Foundation of China (No. 70890080, NO.70890083)

References

Bjorck, A. 1988. A direct method for sparse least-squares problems with lower and upper bounds, numer. *Math* 54:19–32.

Black, M. J., and Jepson, A. D. 1996. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV* 329–342.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge University Press.

Candes, E., and Romberg, J. 2005. *l1-magic: recovery of sparse signals via convex programming*. http://www.acm.caltech.edu/l1magic/.

Candes, E., and Tao, T. 2005. Decoding by linear programming. *IEEE Transactions on Information Theory* 51(12).

Donoho, D. 2006. Compressed sensing. *IEEE Transactions on Information Theory* 52(4):1289–1306.

Georghiades, A.; Belhumeur, P.; and Kriegman, D. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE TPAMI* 23(6):643–660.

He, X.; Yan, S.; Hu, Y.; Niyogi, P.; and Zhang, H. 2005. Face recognition using laplacianfaces. *IEEE TPAMI* 27(3):328–340.

Ji, Y. F.; Lin, T.; and Zha, H. B. 2009. Mahalanobis distance based non-negative sparse representation for face recognition. In *ICMLA*.

Lee, H.; Battle, A.; Raina, R.; and Ng, A. Y. 2006. Efficient sparse coding algorithms. In *NIPS*.

Liu, W.; Pokharel, P. P.; and Principe, J. C. 2007. Correntropy: Properties and applications in non-gaussian signal processing. *IEEE Transactions on Signal Processing* 55(11):5286–5298.

Martinez, A., and Benavente, R. 1998. The ar face database. *CVC, Tech. Rep.*

Naseem, I.; Togneri, R.; and Bennamoun, M. 2009. Linear regression for face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence. Accepted.*

Nene, S.; Nayar, S.; and Murase, H. 1996. Columbia object image library (coil-20). *Technical Report CUCS-005-96.*

Portugal, L.; Judice, J.; and Vicente, L. 1994. A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables. *Mathematics of Computation* 63(208):625–643.

Sanja, F.; Skocaj, D.; and Leonardis, A. 2006. Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. *IEEE TPAMI* 28(3):337–350.

Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman and Hall.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B* 58(1):267–288.

Turk, M., and Pentland, A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1):71–86.

Vo, N.; Moran, B.; and Challa, S. 2009. Nonnegative-least-square classifier for face recognition. In *Advances in Neural Networks*.

Wright, J.; Ma, Y.; Mairal, J.; Sapiro, G.; Huang, T.; and Yan, S. 2009a. Sparse representation for computer vision and pattern recognition. *Proceedings of IEEE.*

Wright, J.; Yang, A.; Ganesh, A.; Sastry, S.; and Ma, Y. 2009b. Robust face recognition via sparse representation. *IEEE TPAMI.*

Xing, E. P.; Ng, A. Y.; Jordan, M. I.; and Russell, S. 2002. Distance metric learning with application to clustering with side-information. In *NIPS.*

Yang, S. H.; Zha, H. Y.; Zhou, S.; and Hu, B.-G. 2009. Variational graph embedding for globally and locally consistent feature extraction. In *ECML*, 538–553.

Yuan, X., and Hu, B. 2009. Robust feature extraction via information theoretic learning. In *ICML.*