21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Kernel based Sparse Representation for Face Recognition

Qi Zhu[1,2], Yong Xu[1,2], Jinghua Wang[3] and Zizhu Fan[1,2]
1 Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology
2 Key Laboratory of Network Oriented Intelligent Computation, Shen Zhen
3 Department of Computing, The Hong Kong Polytechnic University, Hong Kong
ksqiqi@sina.com; {laterfall2,zzfan3}@yahoo.com.cn; jinghuawng@gmail.com

Abstract

In this paper, we extend the idea of sparse representation into the high dimensional feature space induced by the kernel function, and propose a kernel based test sample sparse representation and classification algorithm (KTSRC) for the first time. The KTSRC is based on the assumption that the test sample can be linearly represented by a part of the training samples in the high dimensional feature space. Although the explicit form of the sample in the feature space is unknown, we can implement the KTSRC by the kernel trick. The experimental results show that the KTSRC achieves promising performance in face recognition, and outperforms the state-of-the-art methods.

1. Introduction

Sparse coding technique is widely used in image compressive sensing due to its good performance [1,2]. Under the assumption that the testing face image can be reconstructed by the training face images from the same class [3], sparse coding can also be used in face recognition. In 2009, Wright et al. first applied the sparse coding technique to face recognition and proposed the sparse representation based classification (SRC) [3], which attracted much attention in the past few years [4-8]. In SRC, the testing face image is sparsely coded over the training samples. In the other words, the test sample is represented as a sparse weighted combination of the training samples. Then the classification is performed by comparing which class yields the least representation error.

The sparsest representation model of the SRC is determined by solving the weights with L0 norm optimizer, while the solution with the L1 norm optimizer may not achieve the sparest model. But the latter is more widely used because of its relatively high computation efficiency. Yang et al. provided some reasonable supports for L1 norm optimizer based classification method [9]. Zhang believes that the collaborative representation is also appropriate for face recognition, and proposed the collaborative representation based classification (CRC) in [10]. Xu et al. proposed the two-phase test sample representation (TPTSR) that can be viewed as a local or sparse version of CRC, and achieved better performance [11]. Both CRC and TPTSR are based on the L2 norm optimizer.

In this paper, we propose a test sample sparse representation based method. As this method works in the feature space induced by the kernel function [12], we refer to it as kernel based test sample sparse representation and classification algorithm (KTSRC). As the kernel method, KTSRC is able to capture the nonlinear relationships of samples [13]. The KTSRC includes two stages. In the first stage of the KTSRC, we construct a collaborative representation for the test sample in the feature space. Basing on this representation model, we find some "nearest neighbors" of the test sample, which are some training samples having the most contributions in representing the test sample. In the second stage of the KTSRC, we represent the test sample as a new linear combination of the determined training samples in the feature space and use the representation error of each class for classification. Compared to pervious sample sparse representation based methods, our method has the explicit physical meaning in selecting the training sample for representing the test sample, which may be important for constructing more robust representation model. Besides, although the feature space is usually high dimensional, our method is very efficient in computation by using the kernel trick.

The rest of this paper is organized as follows: In section 2, we formally present the KTSRC. In Section 3, we analyze some characteristics of the KTSRC by comparing it with the other sample representation based methods. In Section 4, the experiments are carried on public face datasets to evaluate the performance of our method. Finally, we offer the conclusion in Section 5.

2. The kernel based SRC

The KTSRC aims at finding the sparse representation of the test sample in feature space, and it is implemented by the following two stages.

2.1 The first stage of the KTSRC

Suppose there are n training samples, $\{x_{1,1}, x_{1,2},..., x_{c,n_c}\} \in R^{d\times n}$ (d is the dimension of the samples), from c classes, and the i th class has n_i training samples $x_{i,1},..., x_{i,n_i}$ ($n = \sum n_i$). Let y be the test sample. It is assumed that a nonlinear function ϕ maps all the samples $y, x_{1,1}, x_{1,2},..., x_{c,n_c}$ into the feature space where they are represented by $\phi(y), \phi(x_{1,1}), \phi(x_{1,2}),..., \phi(x_{c,n_c})$, respectively. Suppose that y can be represented by all the training samples in the feature space by

$$\phi(y) = Bw \tag{1}$$

where $B = (\phi(x_{1,1}), \phi(x_{1,2}),..., \phi(x_{c,n_c}))$ and $w = (w_{1,1}, w_{1,2},..., w_{c,n_c})^T$, respectively.

The w in Eq.(1) is equivalent to that in the following equation

$$B^T\phi(y) = B^T Bw \tag{2}$$

By the kernel trick [13], we have

$$K = B^T B = \begin{bmatrix} k(x_{1,1},x_{1,1}) & k(x_{1,1},x_{1,2}) \dots k(x_{1,1},x_{c,n_c}) \\ k(x_{1,2},x_{1,1}) & k(x_{1,2},x_{1,2}) \dots k(x_{1,2},x_{c,n_c}) \\ \dots\dots\dots & \dots\dots\dots \\ k(x_{c,n_c},x_{1,1}) & k(x_{c,n_c},x_{1,2}) \dots k(x_{c,n_c},x_{c,n_c}) \end{bmatrix},$$

and $K_y = B^T\phi(y) = (k(y,x_{1,1}), k(y,x_{1,2}),..., k(y,x_{c,n_c}))^T$. w can be calculated by:

$$w = (K^T K + uI)^{-1} K^T K_y \tag{3}$$

where u is a small positive constant and I is the identity matrix. As we can see from Eq. (1), the different training samples have the different contributions in representing the test sample in the feature space. For the training sample $X_{i,j}$, we use the 'distance' $\| \phi(y) - w_{i,j}\phi(x_{i,j}) \|_2^2$ to evaluate the contribution of $X_{i,j}$ in representing $\phi(y)$ ($\| \cdot \|_2$ denotes the L2 norm of a vector). The 'distance' between y and $x_{i,j}$ is calculated using

$$\begin{aligned} &\| \phi(y) - w_{i,j}\phi(x_{i,j}) \|_2^2 \\ &= k(y,y) + {w_{i,j}}^2 k(x_{i,j},x_{i,j}) - 2w_{i,j}k(y,x_{i,j}) \end{aligned} \tag{4}$$

We identify M training samples that have the first M smallest 'distances', and denote them by $\phi(x_1'), \phi(x_2'),..., \phi(x_M')$. We also regard that these M training samples have the most contribution in representing the test sample.

2.2 The second stage of the KTSRC

In the second stage, the KTSRC represents the test sample as a linear combination of the training samples $\phi(x_1'), \phi(x_2'),..., \phi(x_M')$ in feature space that is

$$\phi(y) = w_1'\phi(x_1') + w_2'\phi(x_2') + ... + w_M'\phi(x_M') \tag{5}$$

Let $B' = (\phi(x_1'), \phi(x_2'),..., \phi(x_M'))$ and $w' = (w_1', w_2',..., w_M')^T$. Then Eq.(5) can be rewritten as:

$$\phi(y) = B'w' \tag{6}$$

Eq. (6) is equivalent to

$$B'^T\phi(y) = B'^T B'w' \tag{7}$$

Using the kernel trick, we can get

$$K' = B'^T B' = \begin{bmatrix} k(x_1',x_1') & k(x_1',x_2') \dots k(x_1',x_M') \\ k(x_2',x_1') & k(x_2',x_2') \dots k(x_2',x_M') \\ \dots\dots\dots & \dots\dots\dots \\ k(x_M',x_1') & k(x_M',x_2') \dots k(x_M',x_M') \end{bmatrix},$$

and $K'_y = B'^T\phi(y) = (k(y,x_1'), k(y,x_2') \dots k(y,x_M'))^T$

Then, w' can be calculated by

$$w' = (K'^T K' + u'I)^{-1} K'^T K'_y \tag{8}$$

where u' is a positive constant.

When we get the weight vector w', the residual of each class can be derived. For example, the residual of class i is:

$$\| \phi(y) - \sum w_j'\phi(x_j) \|_2^2, \ (\phi(x_j) \in class \ \ i) \tag{9}$$

By comparing the residuals in feature space, we assign y to the class of the training sample that has the minimum residual.

3. The characteristics of the KTSRC

The sample representation based methods use different training samples to represent the test sample. For simplicity, we do not consider the space difference between our method and the other sample representation based methods, and just focus on the sample selection criterions of all the methods. When the SRC with L0 or L1 norm optimizer selects the training samples, it tries to minimize either L0 or L1 norm of the weight vector, which consists of the weights of the training samples, and the representation error, which is the deviation between the test sample and its representation result. Besides considering the representation error, our method essentially simultaneously uses the weight and the training sample itself in the selection of the training samples. Such a "double checking" may make the classification more effective and robust. Using all the training samples for representing the test sample, the CRC may obtain more precise representation model for each test sample. But the CRC considers all the training samples equally in the representation model. The fact is that the test sample only belongs to one class, so it is probably that the test sample is mainly represented by only a part of training samples, which is also the basic assumption of SRC and our method.

For all the representation based classification methods, the main cost of the computational time is spent on computing the weight vector. Obviously, the computation time of SRC with L0 norm optimizer is the most, because it needs to solve a NP-hard problem. SRC with L1 norm optimizer can speed up the process of the solution by some iterative algorithms. Our method, CRC [10] and TPTSR [11] use L2 norm optimizer, and they can easily get the analytic solution by solving the linear equations. So our method, CRC and TPTSR are faster than both SRC with L0 norm optimizer and SRC with L1 norm optimizer. In detail, our method obtains the weight vector by solving the linear equations with n variables in Eq. (3), whose number of equations is equal to that of variables, while the CRC and the TPTSR method obtain the weight vector by solving the linear equations with n variables, whose number of equations is equal to that of the features of the sample in original space.

4. Experiments

The proposed method was applied to face recognition. We adopted the Gaussian kernel in the form of $k(x, y) = \exp(-\| x - y \|_2^2 / \sigma)$ in the experiments, where σ is set to 1. The first face dataset used in experiments is the AR dataset. It contains over 4000 face images of 126 people. The face portion of each image is manually cropped to the size of 50× 40. We used only the images of 120 subjects and 14 non-occluded face images of each subject to test different methods. We pick 7 images of each subject at random for training. Remaining 7 images of each subject are employed for testing. 10 randomly possible selections of training images per class are chosen in the experiments, and the experiments are repeated 10 times with these selections. Figure 1 and Figure 2 show the average classification error rate and time of our method and TPTSR with different "nearest neighbors". The lowest average error rate of our method and TPTSR were obtained by using 200 and 120 "nearest neighbors", respectively. Table 1 shows the average classification error rates and time of our method using 200 "nearest neighbors", TPTSR using 120 "nearest neighbors", CRC and SRC. Our method achieves the lowest classification average error rate and the highest classification efficiency among all the methods on this dataset.

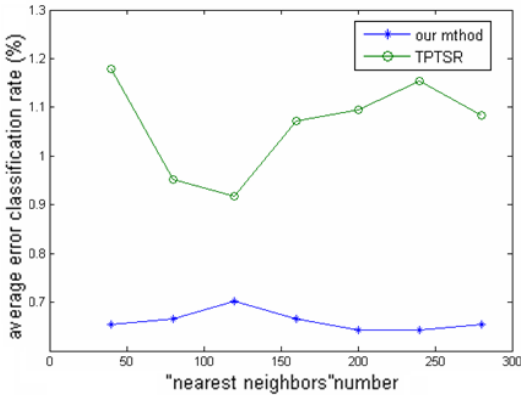

Figure 1. The average error classification rate (%) of our method and TPTSR on AR.

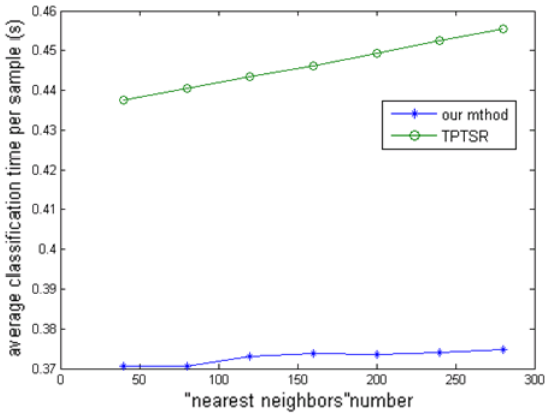

Figure 2. The average classification time (s) of our method and TPTSR on AR

Table 1. The performance of the methods on AR

Methods	KTSRC	TPTSR	CRC	SRC
Average classification error rate (%)	0.64	0.92	1.02	1.07
Average classification time per sample (s)	0.37	0.44	0.40	4.23

The Carnegie Mellon University Pose, Illumination and Expression (CMU PIE) dataset is also used in our experiments. The CMU PIE dataset consists of more than 40,000 facial images of 68 people. In the construction of this dataset, the images of each individual are captured under 43 different illumination conditions, across 13 different poses, and with 4 different expressions. All the images are resized to a resolution of 64×64 pixels. The following experiments were carried on pose09 subset. This subset includes 24 images for each individual. 8, 10, 12 training images per class are used for training, and the remaining samples are used for testing. We set the "nearest neighbors" number be 30, 50, and 100 in our method and TPTSR. CRC and SRC are also carried on this dataset. Table 2 presents the classification results of the four methods. Both our method and TPTSR outperforms CRC and SRC greatly in classification accuracy. Compared to TPTSR, our method achieves the lower classification error rate, when they use the same "nearest neighbors" number.

Table 2. The average error classification rate of the methods on PIE

Methods ("nearest neighbors" number)	Error rate (%)		
	8 training images per class	10 training images per class	12 training images per class
KTSRC(30)	3.86	4.20	5.15
TPTSR(30)	4.60	5.57	6.74
KTSRC (50)	3.40	3.99	5.02
TPTSR(50)	4.41	5.04	6.13
KTSRC (100)	3.21	3.67	4.90
TPTSR(100)	3.76	5.04	5.75
CRC	5.15	6.30	7.72
SRC	5.61	7.46	7.48

5. Conclusion

The previous sample representation based classification methods work in the original space, whereas the KTSRC proposed in this paper is implemented in a high dimensional feature space for the first time. As the kernel method, KTSRC is able to capture the nonlinear relationships of samples. By the kernel trick, we can implement our method using the computational time of the linear method. The experimental results show that KTSRC has better performance than SRC, CRC and TPTSR. In the further study, we will explore how to automatically set the "nearest neighbors" number for obtaining the best classification performance.

References

[1] E. Candes, M. Rudelson, T. Tao, and R. Vershynin, *Error Correction via Linear Programming*, in IEEE Symposium on FOCS:295–308, 2005

[2] S. Mallat, Z. Zhang, *Matching Pursuits with Time-Frequency Dictionaries*, IEEE Transactions on Signal Processing, 41 (12):3397–34, 1993

[3] J. Wright, A. Y. Yang, A. Ganesh, S. Shankar Sastry, Y. Ma, *Robust Face Recognition via Sparse Representation*, 31(2):1-17, 2009

[4] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, and Y. Ma, *Towards a Practical Face Recognition System: Robust Registration and Illumination via Sparse Representation*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[5] Z. Zhou, A. Wagner, H. Mobahi, J. Wright, and Y. Ma, *Face Recognition with Contiguous Occlusion using Markov Random Fields*, IEEE International Conference on Computer Vision (ICCV), 2009.

[6] R. Sala Llonch, E. Kokiopoulou, I. Tošić, P. Frossard, *3D Face Recognition with Sparse Spherical Representations*, Pattern Recognition, 43(3):824-834, 2010

[7] R. Zhi et al., *Graph-Preserving Sparse Nonnegative Matrix Factorization With Application to Facial Expression Recognition*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 41 (1):38-52, 2011

[8] I. Naseem, R. Togneri, and M. Bennamoun, *Linear Regression for Face Recognition*, IEEE PAMI, 32(11):2106-2112, 2010

[9] J. Yang, L. Zhang, Y. Xu, et al., *Beyond Sparsity: The Role of L1-Optimizer in Pattern Classification*, Pattern Recognition, 45:1104–1118, 2012

[10] L. Zhang, M. Yang and X. Feng, *Sparse Representation or Collaborative Representation: Which Helps Face Recognition?*, IEEE International Conference on Computer Vision ICCV, 2011

[11] Y. Xu, D. Zhang, J. Yang, J.-Y. Yang, *A Two-Phase Test Sample Sparse Representation Method for Use with Face Recognition*, IEEE Transactions on Circuits and Systems for Video Technology, 21(9):1255-1262, 2011

[12] C. Liu, *Gabor-based Kernel Pca with Fractional Power Polynomial Models for Face Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(5):572-581, 2004.

[13]Y. Xu, D. Zhang, Z. Jin, M. Li, et al., *A Fast Kernel-based Nonlinear Discriminant Analysis for Multi-Class Problems*, Pattern Recognition,39(6):1026-1033, 2006