Multiple Kernel Sparse Representation Based Classification

Hao Zheng[1,2], Fan Liu[1], and Zhong Jin[1]

[1] School of Computer Science and Technology,
Nanjing University of Science and Technology,
210094 Nanjing, P.R. China
[2] School of Mathematics and Information Technology,
Nanjing XiaoZhuang University,
210017 Nanjing, P.R. China
zhh710@163.com

Abstract. Sparse representation based classification (SRC) has been very successful in many pattern recognition problems. Recently, some extended kernel methods have been proposed through mapping the samples from original feature space into a high dimensional feature space, and then performing the SRC in the high dimensional feature space. However they are all simple kernel methods whose kernel is not most suitable one. For addressing this question, we proposed a novel method named multiple kernel sparse representation based classification (MKSRC), which combine several possible kernels and make full of kernel information. More importantly kernel weights of MKSRC can be automatically selected. The experimental results of face databases indicated recognition performance of new method is superior to other state-of-the-art methods.

Keywords: sparse representation based classification (SRC), multiple kernel, face recognition, kernel weight.

1 Introduction

The nearest-neighbor classifier(NN) is extremely simple and it is accurate and applicable to various problems [1]. The simplest 1-nn algorithm assigns an input sample to the category of its nearest neighbor from the labeled training set. Instead of looking at the closest reference sample, Hart [2] and Wilson [3] proposed k-nn classfier which looks at the k samples in the reference set that are closest to the unknown sample and carries out a vote to make a decision. The support vector machine classifier is also another classifier which is solidly based on the theory of structural risk minimization in statistical learning. It is well known that the SVM maps the inputs to a high-dimensional feature space and then finds a large margin hyperplane between the two classes which can be solved through the quadratic programming algorithm.

C.-L. Liu, C. Zhang, and L. Wang (Eds.): CCPR 2012, CCIS 321, pp. 48–55, 2012.

Recently the kernel approach [4] has attracted great attention. It offers an alternative solution to increase the computational power of linear learning machines by mapping the data into a high dimensional feature space. The approach has been studied and extended some kernel based algorithms such as kernel principal component analysis (KPCA) [5] and kernel fisher discriminant analysis(KFD)[6,7]. As the extension of conventional nearest-neighbor algorithm, the kernel optimization algorithm [8-14] was proposed which can be realized by substitution of a kernel distance metric for the original one in Hilbert space. By choosing an appropriate kernel function, the results of kernel nearest-neighbor algorithm are better than those of conventional nearest-neighbor algorithm. Similarity, the single-kernel SVM classifier was proposed, and various remedies were introduced, such as the reduced set method[15],[16],bottom-up method[17], building of a sparse large margin classifier[18],[19], and the incremental building of a reduced-complexity classifier[9].

But these methods have some disadvantages. NN predicts the category of the image to be tested by only using its nearest neighbor in the training data, so it can easily be affected by noise. The shortcoming of the SVM is that it is often not as compact as the other classifiers such as neural networks. Fortunately Wright et al. proposed a sparse representation based classification for face recognition[20](SRC) which first codes a testing sample as a sparse linear combination of all the training samples, and then classifies the testing sample by evaluating which class leads to the minimum representation error. SRC is much more effective than state-of-art methods in dealing with face occlusion, corruption, lighting and expression changes, etc. It is well known that if an appropriate kernel function is utilized for a test sample, more neighbors probably have the same class label in the high dimensional feature space. So sparse representation in the high dimensional can improve the performance of recognition and discriminating ability, and Some methods were proposed such as kernel representation based classification algorithm (KSRC) [21, 22, 23], etc. However it is often unclear what the most suitable kernel for the task at hand is, and hence the user may wish to combine several possible kernels. One problem with simply adding kernels is that using uniform weights is possibly not optimal. To overcome it, we proposed a novel method names multiple sparse representation based classfication (MKSRC) which can optimize the kernel weights while training the KSRC.

2 Multiple Kernel Sparse Representention Based Classification (MKSRC)

Suppose there are p classes in all, and the set of the training samples is $A=[A_1,A_2,\ldots,A_p]=[x_{1,1},x_{1,2},\ldots,x_{p,n_p}]\in\Re^{d\times N}$, where $N=\sum_{i=1}^{p}n_i$ and $y\in\Re^{d\times 1}$ is the test sample. The traditional sparse coding model is equivalent to the so-called LASSO problem [24]:

$$\min \| y-A\beta \|_2^2 \ \ s.t. \ \ \| \beta \|_1 < \sigma \qquad \text{where } \sigma > 0 \text{ is a constant.}$$

Suppose there exists a feature mapping function $\phi:\Re^d \to \Re^k (d<K)$. It maps the feature and basis to the high dimensional feature space: $y \to \phi(y), A=[x_{1,1},x_{1,2},\ldots,x_{p,n_p}] \to U=[\phi(x_{1,1}),\phi(x_{1,2}),\cdots,\phi(x_{p,n_p})]$ .there exits one problem that one kernel is not most suitable kernel, so we wish to combine several possible kernels. The mode of Multiple Kernel by Lanckriet[25] et al. is $k(x_i,x_j)=\sum_{k=1}^{m}\alpha_k k_k(x_i,x_j)$, and we constraint the kernel weights by $\sum_{i=1}^{m}\alpha_K{}^2=1,\ \alpha_K \ge 0$, then substitute the mapped features and basis to the formulation of sparse coding, obtain the objective function as follow:

$$\min \|\phi(y)-Uv\|_2^2 \quad s.t. \|v\|_1 < \sigma \ \sum_{\mathrm{i=1}}^{\mathrm{m}}\alpha_K{}^2 = 1 \tag{1}$$

The Lagrangian function for Eq.(1) is

$$\begin{aligned}
J &= \|\phi(y)-Uv\|_2^2 + \lambda\|v\|_1 + \gamma(\boldsymbol{\alpha}^T\boldsymbol{\alpha}-1)\\
&= \phi(y)^T\phi(y) - 2v^TU^T\phi(y) + v^TU^TUv + \lambda\|v\|_1 + \gamma(\boldsymbol{\alpha}^T\boldsymbol{\alpha}-1)\\
&= \sum_{k=1}^{m}\alpha_k k_k(y,y) - 2v^T[\phi(x_{1,1}),\phi(x_{1,2}),\cdots,\phi(x_{p,n_p})]^T\phi(y)\\
&\quad + v^T[\phi(x_{1,1}),\phi(x_{1,2}),\cdots,\phi(x_{p,n_p})]^T[\phi(x_{1,1}),\phi(x_{1,2}),\cdots,\phi(x_{p,n_p})]v + \lambda\|v\|_1 + \gamma(\boldsymbol{\alpha}^T\boldsymbol{\alpha}-1)
\end{aligned} \tag{2}$$

For sample x and y , we have

$$\phi(x_i)^T\phi(y_j) = k(x_i,y_j) \quad k(x_i,x_j)=\sum_{k=1}^{m}\alpha_k k_k(x_i,x_j)$$

Therefore

$$\begin{aligned}
J = \sum_{k=1}^{m}\alpha_k k_k(y,y) - 2v^T
\begin{bmatrix}
\sum_{k=1}^{m}\alpha_k k_k(x_{1,1},y)\\
\sum_{k=1}^{m}\alpha_k k_k(x_{1,2},y)\\
\vdots\\
\sum_{k=1}^{m}\alpha_k k_k(x_{p,n_p},y)
\end{bmatrix}
+ v^T
\begin{bmatrix}
\sum_{k=1}^{m}\alpha_k k_k(x_{1,1},x_{1,1}) & \sum_{k=1}^{m}\alpha_k k_k(x_{1,1},x_{1,2}) & \cdots\sum_{k=1}^{m}\alpha_k k_k(x_{1,1},x_{p,n_p})\\
\sum_{k=1}^{m}\alpha_k k_k(x_{1,2},x_{1,1}) & \sum_{k=1}^{m}\alpha_k k_k(x_{1,2},x_{1,2}) & \cdots\sum_{k=1}^{m}\alpha_k k_k(x_{1,2},x_{p,n_p})\\
\vdots & \vdots & \ddots\\
\sum_{k=1}^{m}\alpha_k k_k(x_{p,n_p},x_{1,1}) & \sum_{k=1}^{m}\alpha_k k_k(x_{p,n_p},x_{1,2}) & \cdots\sum_{k=1}^{m}\alpha_k k_k(x_{p,n_p},x_{p,n_p})
\end{bmatrix} v \\
+ \lambda\|v\|_1 + \gamma(\sum_{k=1}^{m}\alpha_k{}^2 - 1)
\end{aligned} \tag{3}$$

Setting the derivative of J w.r.t. the primal variable α_i to zero,

$$\frac{\partial J}{\partial \alpha_K} = k_i(y,y) - 2v^T \begin{bmatrix} k_k(x_{1,1},y) \\ k_k(x_{1,1},y) \\ \vdots \\ k_k(x_{p,n_p},y) \end{bmatrix} + v^T \begin{bmatrix} k_k(x_{1,1},x_{1,1}) & k_k(x_{1,1},x_{1,2}) & \cdots & k_k(x_{1,1},x_{p,n_p}) \\ k_k(x_{1,2},x_{1,1}) & k_k(x_{1,2},x_{1,2}) & \cdots & k_k(x_{1,2},x_{p,n_p}) \\ \vdots & \vdots & \ddots & \\ k_k(x_{p,n_p},x_{1,1}) & k_k(x_{p,n_p},x_{1,2}) & \cdots & k_k(x_{p,n_p},x_{p,n_p}) \end{bmatrix} v + 2\gamma\alpha_K = 0 \quad (4)$$

Finally we obtain

$$\alpha_K = -\frac{1}{2\gamma}(k_i(y,y) - 2v^T \begin{bmatrix} k_k(x_{1,1},y) \\ k_k(x_{1,1},y) \\ \vdots \\ k_k(x_{p,n_p},y) \end{bmatrix} + v^T \begin{bmatrix} k_k(x_{1,1},x_{1,1}) & k_k(x_{1,1},x_{1,2}) & \cdots & k_k(x_{1,1},x_{p,n_p}) \\ k_k(x_{1,2},x_{1,1}) & k_k(x_{1,2},x_{1,2}) & \cdots & k_k(x_{1,2},x_{p,n_p}) \\ \vdots & \vdots & \ddots & \\ k_k(x_{p,n_p},x_{1,1}) & k_k(x_{p,n_p},x_{1,2}) & \cdots & k_k(x_{p,n_p},x_{p,n_p}) \end{bmatrix} v) \quad (5)$$

Because $\phi(y)$ and U are unknown, Eq.(1) cannot be solved directly. But according to [21], Eq.(1) can be transformed to

$$\hat{v} = \arg\min\{\| U^T\phi(y) - U^TUv \|_2{}^2 + \lambda \| v \|_1\} \quad (6)$$

where $$U^T\phi(y) = \begin{bmatrix} \sum_{k=1}^{m} \alpha_k k_k(x_{1,1},y) \\ \sum_{k=1}^{m} \alpha_k k_k(x_{1,2},y) \\ \vdots \\ \sum_{k=1}^{m} \alpha_k k_k(x_{p,n_p},y) \end{bmatrix}$$

and

$$U^TU = \begin{bmatrix} \sum_{k=1}^{m} \alpha_k k_k(x_{1,1},x_{1,1}) & \sum_{k=1}^{m} \alpha_k k_k(x_{1,1},x_{1,2}) & \cdots \sum_{k=1}^{m} \alpha_k k_k(x_{1,1},x_{p,n_p}) \\ \sum_{k=1}^{m} \alpha_k k_k(x_{1,2},x_{1,1}) & \sum_{k=1}^{m} \alpha_k k_k(x_{1,2},x_{1,2}) & \cdots \sum_{k=1}^{m} \alpha_k k_k(x_{1,2},x_{p,n_p}) \\ \vdots & \vdots & \ddots \\ \sum_{k=1}^{m} \alpha_k k_k(x_{p,n_p},x_{1,1}) & \sum_{k=1}^{m} \alpha_k k_k(x_{p,n_p},x_{1,2}) & \cdots \sum_{k=1}^{m} \alpha_k k_k(x_{p,n_p},x_{p,n_p}) \end{bmatrix}$$

Since initial weights are an estimator which is not optimal, the implementation of MKSRC is an iterative process. When the difference of weights α_i is small enough, the convergence is stopped. It can be formulated as follow:

$\| \alpha^{t+1} - \alpha^t \| \leq tol$ From above all, the MKSRC algorithmic procedures can be summarized as Algorithm 1:

Algorithm 1. Multiple Kernel Sparse Representation based Classification algorithm

Step 1: Input training samples $A \in \Re^{d \times \sum_{i=1}^{p} n_i}$ partitioned into p classes, and a test sample y . The number of kernel function is m .

Step 2: Compute initial weights $\alpha_K = \frac{1}{\sqrt{m}}$ and $k(x_i, x_j) = \sum_{k=1}^{m} \alpha_k k_k (x_i, x_j)$

Step 3: Compute the coefficient $\hat{v} = \arg\min\{\| U^T \phi(y) - U^T U v \|_2^{\,2} + \lambda \| v \|_1\}$

Step 4: Compute the weights α_K by Eq. (5)

Step 5: Go back to step 3 until the condition of convergence is met.

Step 6: Compute $r_j(y) = \| U^T \phi(y) - U^T U \hat{v} \|_2^{\,2}$

Step 7: Output that $\mathrm{identity}(y) = \mathrm{argmin}\ r_j(y)$.

3 Experiments and Discussions

In this section, we perform experiments on face databases to demonstrate the efficiency of MKSRC. To evaluate more comprehensively the performance of MKSRC, in section 3.1 we first test face recognition without occlusion, and then in section 3.2 we demonstrate the robustness and high efficiency of the proposed method to random block occlusion. In the experiments, three single kernel function were tried with linear, polynomial, and Gaussian kernels and the kernel parameters were tuned using cross validation. We performed 10 trials , and report the average test results.

3.1 Face Recognition without Occlusion

The ORL face dataset consists of 400 frontal face images of 40 subjects. They are captured under various lighting conditions and cropped and normalized to 112 × 92 pixels. The face images were captured under various illumination conditions. We randomly split the database into two halved. One half (5 images per person) was used for training, and the other half for testing. The images are reduced to 30, 60, 110 dimensions, respectively. Table 1 and Fig.1 illustrate the face recognition rates under different methods. We can see that the recognition rates are different with the various dimensions. Our MKSRC method achieves a recognition rate between 89% and 97.8%, much better than the other methods.

Table 1. Accuracy on ORL face database

	SVM	SRC	KSRC(Polynomial)	KSRC(Gaussian)	MKSRC
Dimensions(d=30)					
Accuracy	93%	93%	95.5%	95.5%	97.1%
Parameter			d=13	t=2	d=13 t=2
Dimensions(d=60)					
Accuracy	94%	93.5%	96.5%	95.5%	97.8%
Parameter			d=13	t=2	d=13 t=2
Dimensions(d=110)					
Accuracy	94%	94%	96.5%	94%	97.2%
Parameter			d=13	t=2	d=13 t=2

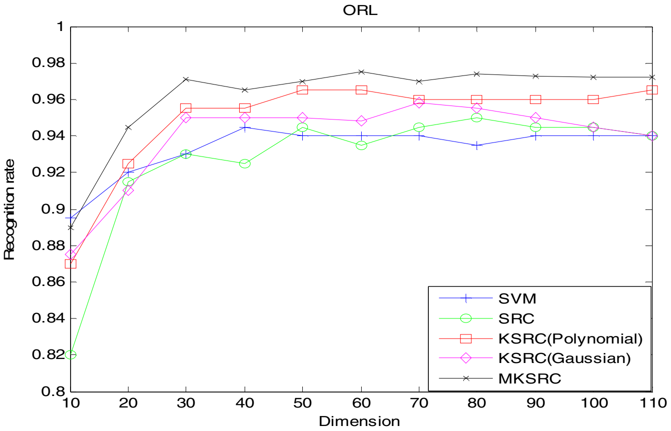

Fig. 1. The average recognition rates of SVM, SRC, KSRC (Polynomial), KSRC (Gaussian) and MKSRC versus the dimensions on ORL face database

Table 2. Accuracy on ORL face database with block occlusion

	SRC	KSRC(Polynomial)	KSRC(Gaussian)	MKSRC
Occlusion(10%)				
Accuracy	89%	91%	90.4%	93.3%
Parameter		d=2	t=3	d=2 t=3
Occlusion (20%)				
Accuracy	80.5%	83.5%	81%	84%
Parameter		d=2	t=3	d=2 t=3
Occlusion (30%)				
Accuracy	71%	73.6%	71%	74.8%
Parameter		d=2	t=3	d=2 t=3

3.2 Face Recognition with Block Occlusion

The next one is more challenging, we test the efficiency of MKSRC to the block occlusion using the ORL face dataset. We randomly take the first half for training and the rest for testing. We simulate various levels of contiguous occlusion, from 10% to 30%, by replacing a randomly located square block of each test image with an unrelated image, Again, the location of occlusion are randomly chosen for each image and are unknown to the computer. A test example of ORL with 30% occluded block is shown as Figure 2. Here, for computational convenience, the size of image is cropped to 32 × 32, and the images are reduced to 60 dimensions. In Table2, the accuracy rate of all the methods decline with the occlusion levels increasing, which indicates that loss of feature affects the face recognition performance. But MKSRC preserves good performance of 74.8% when the occlusion percentage is 30%.

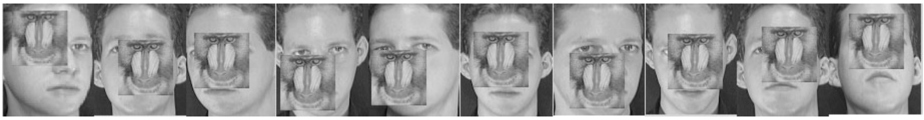

Fig. 2. An test example of ORL face database with 30% occluded block

4 Conclusion

This paper proposed a multiple kernel sparse representation based classification. On the high-dimensional data such as face images, KSRC could get better performance than SRC. But KSRC does not make full of kernel information. MKSR can solve this problem by combining several possible kernels e.g. RBF kernel, while selecting the suitable weights of kernel function. On face database containing varying pose, MKSRC achieves the best performance. Because kernel parameter is important for the recognition performance, we will focus on the estimating the kernel parameter in the future.

Acknowledgments. This work is partially supported by the National Science Foundation of China under grant No. 60973098.

References

1. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley, New York (1973)
2. Hart, P.E.: The condensed nearest neighbor rule. IEEE Trans. Inf. Theory 16, 515–516 (1968)
3. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybern. SMC-2, 408–421 (1972)

4. Aizerman, M.A., Braverman, E.M., Rozonoer, L.I.: T heoretical foundation of potential function method in pattern recognition learning. Automat. Remote Contr. 25, 821–837 (1964)
5. Scholkopf, B., Smola, A., Muller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10, 1299–1319 (1998)
6. Mike, S., Ratsch, G., Weston, J., Scholkopf, B., Muller, K.R.: Fisher discriminant analysis with kernels. In: Proceedings of the 1999 IEEE Signal Processing Society Workshop Neural Networks for Signal Processing, vol. IX, pp. 41–48 (1999)
7. Mike, S., Ratsch, G., Scholkopf, B., Smola, A., Weston, J., Muller, K.R.: Invariant feature extraction and classification in kernel spaces. In: Proceedings of the 13th Annual Neural Information Processing Systems Conference, pp. 526–532 (1999)
8. Argyriou, A., Hauser, R., Micchelli, C.A., Pontil, M.: A DC algorithm for kernel selection. In: Proc. 23rd Int. Conf. Mach., Pittsburgh, PA, pp. 41–49 (2006)
9. Argyriou, A., Micchelli, C.A., Pontil, M.: Learning convex combinations of continuously parameterized basic kernels. In: Proc. 18th Annu. Conf. Learn. Theory, Bertinoro, Italy, pp. 338–352 (2005)
10. Ong, C.S., Smola, A.J., Williamson, R.C.: Learning the kernel with hyperkernels. J. Mach. Learn. Res. 6, 1043–1071 (2005)
11. Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y.: More efficiency in multiple kernel learning. In: Proc. 24th Int. Conf. Mach. Learn., Corvallis, OR, pp. 775–782 (2007)
12. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: Grandvalet: Simple MKL. J. Mach. Learn. Res. 9, 2491–2521 (2008)
13. Sonnenburg, S., Ratsch, G., Schafer, C., Scholkopf, B.: Large scale multiple kernel earning. J. Mach. Learn. Res. 7, 1531–1565 (2006)
14. Zien, A., Ong, C.S.: Multiclass multiple kernel learning. In: Proc. 24th Int. Conf. Mach. Learn., Corvallis, OR, pp. 1191–1198 (2007)
15. Burges, C.J.C.: Simplified support vector decision rules. In: Proc.13th Int. Conf. Mach. Learn., San Mateo, CA, pp. 71–77 (1996)
16. Scholkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge (2002)
17. Nguyen, D., Ho, T.: An efficient method for simplifying support vector machines. In: Proc. 22nd Int. Conf. Mach. Learn., Bonn, Germany, pp. 617–624 (2005)
18. Wu, M., Scholkopf, B., Bakir, B.: A direct method for building sparse kernel learning algorithms. J. Mach. Learn. Res. 7, 603–624 (2006)
19. Wu, M., Scholkopf, B., Bakir, G.: Building sparse large margin classifiers. In: Proc. 22nd Int. Conf. Mach. Learn., Bonn, Germany, pp. 996–1003 (2005)
20. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. TPAMI 31(2), 210–227 (2009)
21. Yin, J., Jin, Z.: Kernel sparse representation based classification. Neurocomputing 77(1), 120–128 (2012)
22. Gao, S., Tsang, I.W.-H., Chia, L.-T.: Kernel Sparse Representation for Image Classification and Face Recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 1–14. Springer, Heidelberg (2010)
23. Zhang, L., Zhou, W.-D.: Kernel sparse representation-based classifier. IEEE Transactions on Signal Processing 60(4), 1684–1695 (2012)
24. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society B 58(1), 267–288 (1996)
25. Lanckriet, G.R.G., et al.: Learning the Kernel Matrix with Semidefinite Programming. J. Machine Learning Research 5, 27–72 (2004)