

Nonlinear Nearest Subspace Classifier

Li Zhang¹, Wei-Da Zhou², and Bing Liu³

¹ Research Center of Machine Learning and Data Analysis, School of Computer Science and Technology, Soochow University, Suzhou 215006, Jiangsu, China

² AI Speech Ltd., Suzhou 215123, Jiangsu, China

³ Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China

Abstract. As an effective nonparametric classifier, nearest subspace (NS) classifier exhibits its good performance on high-dimensionality data. However, NS could not well classify the data with the same direction distribution. To deal with this problem, this paper proposes a nonlinear extension of NS, or nonlinear nearest subspace classifier. Firstly, the data in the original sample space are mapped into a kernel empirical mapping space by using a kernel empirical mapping function. In this kernel empirical mapping space, NS is then performed on these mapped data. Experimental results on the toy and face data show this nonlinear nearest subspace classifier is a promising nonparametric classifier.

Keywords: Nearest subspace classifier, Kernel empirical mapping, Least square method, Machine learning.

1 Introduction

At present, subspace learning has attracted substantial attention in machine learning and computer vision. The idea of nearest subspace (NS) classification is very popular [9],[10],[3],[8]. Here, subspaces mean not only the (reduced) low-dimensional space but also the space spanned by samples per class. In [10], the nearest feature line (NFL) algorithm is presented to classify a test sample by finding the minimum distance between the test sample and any pair of training samples belonging to the same class. The nearest feature plane (NFP) and nearest feature space (NFS) classifiers are proposed as extensions of NFL in [3]. In these two methods, the distance is computed between a test sample and any plane or space spanned by the training samples per class, respectively. In addition, NFS uses at least four feature training samples to span subspaces in each class. The subspaces are typically spanned by five to nine training samples in [8], and spanned by all the samples per class in [9]. Here, we consider the NS classifier used in [9],[12],[18],[4], i.e., the subspace spanned by all training samples per class. The NS classifier is to classify a test sample based on the best linear representation in terms of all the training samples in each class [12]. There is no training procedure, so NS is a nonparametric learning method, like nearest neighbor (NN) [6] and sparse representation-based classifier (SRC) [12]. The test sample is classified based on the best representation in terms of a single training sample in NN, and based on the best sparse representation according to all training samples of all classes in SRC. NS has been widely applied to face recognition [9],[3],[12],[8], microarray cancer datasets [4],

and credit risk evaluation [18]. However, NS loses its classification ability even for the linearly separable tasks in which the data from different classes have the same direction.

This paper deals with this problem occurred in NS by introducing the kernel empirical mapping, and proposes a nonlinear extension of NS classifier, or nonlinear NS (NNS) classifier. In this method, we firstly map data in the original sample space into a kernel empirical mapping space by using some kernel empirical mapping function. Next, the NS classifier is adopted to classify these mapped samples.

2 Nonlinear Nearest Subspace Classifier

This section proposes a nonlinear extension of NS classifier, NNS classifier. Firstly, the kernel trick and kernel empirical mapping is briefly reviewed in this section. Then we discuss the construction of NNS classifier.

2.1 Kernel Empirical Mapping

The kernel trick is a very popular technique in machine learning and pattern recognition. Typically, the role of the kernel trick is to generalize a linear algorithm to a nonlinear one. It has been successfully applied to SVM, KPCA and KFDA. In these kernel methods, only Mercer kernels, kernels satisfying Mercer's condition, are used. In other words, a Mercer kernel is continuous, symmetric, positive semi-definite kernel function. Usually, a Mercer kernel can be expressed as

$$k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^T \Phi(\mathbf{x}') \quad (1)$$

where $\mathbf{x}$ and $\mathbf{x}'$ are any two points in $\mathcal{X}$, Φ is a nonlinear mapping and $\Phi(\mathbf{x})$ is the image of $\mathbf{x}$ in the feature space. Some common used nonlinear kernels include polynomial kernels, Gaussian radial basis function (RBF) kernels, and wavelet kernels [15],[16]. The polynomial kernels have the form $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + 1)^d$ and RBF kernels can be expressed as $k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|_2^2)$, where $d \in \mathbb{N}$ and $\gamma > 0$ are the kernel parameters.

In kernel methods, we don't know what Φ is and just adopt the kernel function (1). Thus, we can not get the mapped feature space, which makes the operation on the images of samples difficult. Zhang et al. construct a family of empirical mapping with kernel functions, and use them in SVM [14], PCA [17], and FDA [13]. The kernel empirical mapping can be explicitly computed in an empirical mapping space (feature space). Typically, the kernel empirical mapping on the data set $X = \{\mathbf{x}\}_{i=1}^n$ is:

$$\Phi(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \dots, k(\mathbf{x}_n, \mathbf{x})]^T \quad (2)$$

where $\mathbf{x} \in \mathbb{R}^m$ is an arbitrary sample in the input space, and n is the number of samples. From (2), we can see that the dimensionality of the kernel empirical mapping feature space is n . If $m > n$, then a dimensionality reduction is performed by (2). Importantly, it doesn't require that kernel functions used in (2) must satisfy Mercer's condition [14].

2.2 Nonlinear Nearest Subspace Classifier

Assume that there are c training subsets $\{X_1, X_2, \dots, X_c\}$, where the j th class training subset $X_j = \{\mathbf{x}_{j,i}, y_{j,i} = j\}_{i=1}^{n_j}$ is a subspace, $\mathbf{x}_{j,i} \in \mathcal{X} \subset \mathbb{R}^m$, $y_{j,i} \in \{1, \dots, c\} \subset \mathbb{N}$ are labels corresponding to $\mathbf{x}_{j,i}$, n_j is the number of the j th class training samples, $\mathcal{X}$ is the input space, m is the dimensionality of the sample space $\mathcal{X}$ and c is the number of classes. Given a test sample $\mathbf{x} \in \mathcal{X}$, the goal is to assign the label y of $\mathbf{x}$ according to the reconstruction errors between $\mathbf{x}$ and its approximations.

By introducing kernel empirical mapping, we map the samples in the input subspaces into a feature subspace as follows.

$$\Phi : \mathbf{x}_{j,i} \in \mathcal{X}_j \rightarrow \Phi(\mathbf{x}_{j,i}) = [k(\mathbf{x}_{1,1}, \mathbf{x}_{j,i}), \dots, k(\mathbf{x}_{1,n_1}, \mathbf{x}_{j,i}), \dots, k(\mathbf{x}_{c,n_c}, \mathbf{x}_{j,i})]^T \in \mathcal{F}_j \quad (3)$$

where $\Phi(\mathbf{x}_{j,i}) \in \mathbb{R}^n$ is the image of $\mathbf{x}_{j,i}$ in the feature subspace $\mathcal{F}_j$, and $n = \sum_{j=1}^c n_j$. Note that the mapping is performed on the whole training samples instead of the j th class samples. Actually, the kernel function $k(\mathbf{x}, \mathbf{x}')$ can measure a relationship between $\mathbf{x}$ and $\mathbf{x}'$, e.g. similarity relationship. Thus, the image $\Phi(\mathbf{x}_{j,i})$ reflects the relationship between $\mathbf{x}_{j,i}$ and whole training samples, and can be taken as the globally nonlinear features of $\mathbf{x}_{j,i}$ to a certain extent.

Now we construct the j th sample matrix from the j th training samples in $\mathcal{F}_j$. Namely,

$$\mathbf{K}_j = [\Phi(\mathbf{x}_{j,1}), \Phi(\mathbf{x}_{j,2}), \dots, \Phi(\mathbf{x}_{j,n_j})] \in \mathbb{R}^{n \times n_j}, j = 1, \dots, c \quad (4)$$

where each column denotes an image corresponding to a training sample. Obviously, $\mathbf{K}_j$ is full rank in column since $n > n_j$. Now, we represent the test sample linearly by all images in a feature subspace, and then have

$$\Phi(\mathbf{x}) = \mathbf{K}_j \boldsymbol{\alpha}_j, j = 1, \dots, c \quad (5)$$

where $\boldsymbol{\alpha}_j \in \mathbb{R}^{n_j}$ is the weight vector of the j th feature subspace, and

$$\Phi(\mathbf{x}) = [k(\mathbf{x}_{1,1}, \mathbf{x}), \dots, k(\mathbf{x}_{1,n_1}, \mathbf{x}), k(\mathbf{x}_{2,1}, \mathbf{x}), \dots, k(\mathbf{x}_{c,n_c}, \mathbf{x})]^T \quad (6)$$

As mentioned above, $\mathbf{K}_j$ is not a square matrix, so we can not also directly solve the linear equation set (5). By using the least square method, we construct the objective function

$$\min_{\boldsymbol{\alpha}_j} \|\Phi(\mathbf{x}) - \mathbf{K}_j \boldsymbol{\alpha}_j\|_2^2, j = 1, \dots, c \quad (7)$$

Likewise, the solution to (7) can be computed as

$$\boldsymbol{\alpha}_j = (\mathbf{K}_j^T \mathbf{K}_j)^\dagger \mathbf{K}_j^T \Phi(\mathbf{x}) \quad (8)$$

or

$$\boldsymbol{\alpha}_j = (\mathbf{K}_j^T \mathbf{K}_j + \sigma \mathbf{I})^{-1} \mathbf{K}_j^T \Phi(\mathbf{x}) \quad (9)$$

where $(\cdot)^{-1}$ denotes the inverse of a matrix, $\sigma \geq 0$ is a very small constant, say 10^{-8} , and $\mathbf{I}$ is the identity matrix. The approximations or reconstructions of $\mathbf{x}$ in feature subspaces

are $\mathbf{K}_j \boldsymbol{\alpha}_j$, $j = 1, \dots, c$. Among them, the best reconstruction with minimal reconstruction error is selected. The reconstruction error (residual) is defined as the Euclidean distance from the image to its reconstruction. Namely,

$$\delta_j = \|\Phi(\mathbf{x}) - \mathbf{K}_j \boldsymbol{\alpha}_j\|_2, j = 1, \dots, c \quad (10)$$

By using which, we assign the label of the nearest feature subspace to $\mathbf{x}$, or

$$\hat{y} = \arg \min_{j=1, \dots, c} \delta_j \quad (11)$$

The complete classification procedure of NNS is shown in Algorithm 1. The step 5 in Algorithm 1 makes all projected samples lie on a unit hypersphere. Note that the step 4 in Algorithm 1 is an optional one. If we don't choose a projection method, then $\mathbf{P}_j$ is assigned the identity matrix. Clearly, the feature space $\mathcal{F}$ has the dimensionality of n . As stated previously, the dimensionality of the samples is already reduced for small sample size problems in which $m > n$ when applying kernel empirical mapping. Of course, n is definitely larger than n_j . If n is too large, we can perform dimensionality reduction by using any possible projection methods, such as principle component analysis (PCA) [6], Fisher discriminant analysis (FDA) [6], [1], and even random projection (RP) [12].

Algorithm 1. Nonlinear nearest subspace method

1. **Input:** A set of training sample subsets $\{X_j\}_{j=1}^c$, where the subset $X_j = \{\mathbf{x}_{ji}, y_{ji} = j\}_{i=1}^{n_j}, \mathbf{x}_{ji} \in \mathbb{R}^m$, a test sample $\mathbf{x} \in \mathbb{R}^m$, and let $\sigma > 0$.
2. Select a Mercer kernel $k(\cdot, \cdot)$ and its parameters.
3. Map samples in the input space into a feature space. we get the images of the training samples $\mathbf{x}_{ji}$:

$$\Phi(\mathbf{x}_{ji}) = \left[k(\mathbf{x}_{1,1}, \mathbf{x}_{ji}), \dots, k(\mathbf{x}_{2,1}, \mathbf{x}_{ji}), \dots, k(\mathbf{x}_{c,n_c}, \mathbf{x}_{ji}) \right]^T, i = 1, \dots, n_j, j = 1, \dots, c,$$

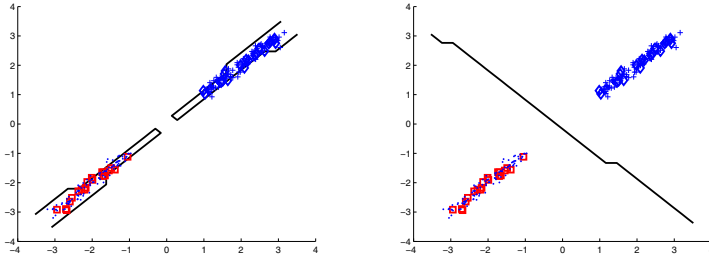
the sample matrices $\mathbf{K}_j$, $j = 1, \dots, c$, and the image of the test sample $\mathbf{x}$

$$\Phi(\mathbf{x}) = \left[k(\mathbf{x}_{1,1}, \mathbf{x}), \dots, k(\mathbf{x}_{1,n_1}, \mathbf{x}), k(\mathbf{x}_{2,1}, \mathbf{x}_{ji}), \dots, k(\mathbf{x}_{c,n_c}, \mathbf{x}) \right]^T.$$

4. [Optional] Select a projection method and get the corresponding projection matrix $\mathbf{P}_j$.
 5. Normalize the columns of $\mathbf{P}_j^T \mathbf{K}_j$, $j = 1, \dots, c$ and $\mathbf{P}_j^T \Phi(\mathbf{x})$ to have unit ℓ_2 -norm.
 6. Solve the least square problem (7) to get the weight vectors $\boldsymbol{\alpha}_j$, $j = 1, \dots, c$ according to (9) or (8).
 7. Compute the c reconstruction errors $\delta_j = \|\mathbf{P}_j^T \Phi(\mathbf{x}) - \mathbf{P}_j^T \mathbf{K}_j \boldsymbol{\alpha}_j\|_2$, $j = 1, \dots, c$.
 8. **Output:** The estimated label $\hat{y}$ for $\mathbf{x}$ according to (11).
-

3 Numerical Experiments

To validate the proposed NNS classifier, we perform numerical experiments on toy and face data sets, and compare NNS with other nonparametrical methods. All numerical experiments are performed on the personal computer with a 1.8GHz Pentium III and



(a) Decision boundary obtained by NS (b) Decision boundary obtained by NNS

Fig. 1. Comparison of NNS with NS on the synthetic data set. The bolded lines are decision boundaries. Training data are denoted by "□" and "◇", respectively; Test data are denoted by "·" and "+", respectively.

3.3 Face Data Sets

It is well known that NS have good performance on face data, so we perform experiment on two face databases, including ORL face database [11], and UMIST face database [7]. Here, NNS is compared with nonparametric learning methods (NN, SRC, and NS). The original features of each face image are obtained by stacking its columns. Then for a $m_1 \times m_2$ gray-scale face image, we get $m_1 m_2$ features which are normalized by 255. Namely features take values from the interval $[0, 1]$. In each database, we randomly select half of images in each subject as the training samples, and the rest as the test samples. This procedure is repeated ten times. The polynomial kernel with $d = 3$ is adopted for NNS.

Since the feature number is very large, it is difficult to directly perform on the original data for some methods, such as SRC. The random projection (RP) method is used for reducing the dimensionality of original data. RP shows its effective in face recognition [12]. Eight dimensionality are adopted here, or 10, 20, 30, 40, 50, 70, 100, and 150. For each dimensionality, random projection is performed 10 runs.

ORL Database: There are 10 different images for each subject in the ORL face database composed of 40 distinct subjects. All the subjects are in up-right, frontal position (with tolerance for some side movement). The size of each face image is 112×92 , and the resulting standardized input vectors are of dimensionality 10,304. The number of images for both training and test is 200. The classification errors on test set are reported in Figure 2(a). We can see that NNS is the best algorithm from 10D to 150D on the ORL face data, where D denotes dimensionality. In addition, NNS slightly decreases its error when the dimensionality is larger than about 40. Whilst NN and NS improve their performance as increasing dimensionality.

UMIST Database: The UMIST face database is a multi-view database which consists of 574 cropped gray-scale images of 20 subjects, each covering a wide range of poses from profile to frontal views as well as race, gender and appearance. Each image in the

database is resized into 112×92 . The total number of the training samples is 290, and that of the test samples is 284. The final results on the UMIST database are shown in Figure 2(b). We observe that NNS is always better than NS from 10D to 150D. Only on 10D and 20D, NNS is worse than NN and SRC.

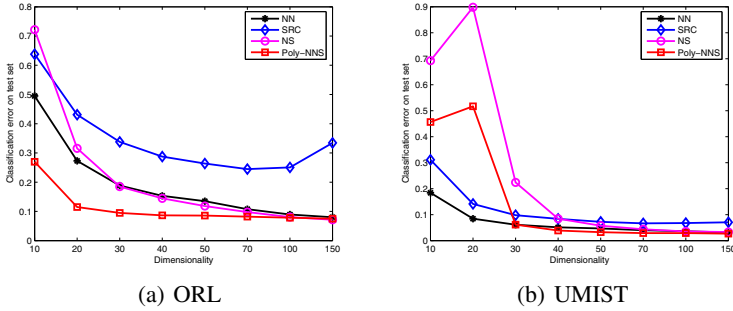


Fig. 2. Classification errors of test set on two face databases (a) ORL, and (b) UMIST

According to the results on two face databases, we have the following conclusions. NNS is always better than NS when the dimensionality is relatively lower, such as less than 50D. In addition, the performance NNS is improved slightly when increasing dimensionality to some degree. On two face data sets, NNS is better than SRC.

4 Conclusion

This paper proposes a nonlinear nearest subspace classifier. NS loses its classification ability when applying it to a general classification task in which samples belonging to different classes have the same orientation. This problem can be solved by mapping the samples into a (RBF) kernel empirical mapping space and then performing NS in this space, which is supported by the experiment on toy data set. Experiments on public face data sets are performed. The performance comparison of NNS with NN, NS, and SRC are taken into account. In the case of high-dimensional face data, NNS with polynomial kernel is better than NS when the dimensionality is lower than 50D.

Although NNS is a promising nonparametric classifier, NNS also has its drawback when processing high-dimensional data. NNS first greatly decreases its test error as increasing dimensionality, and then slightly decreases as increasing dimensionality. In [5], a method for optimizing sensing matrix and sparsifying dictionary is proposed. The sample matrix in NS and NNS can be regarded as the sparsifying dictionary. In the further, the optimization of sample matrix will be taken into account to expectantly improve performance of NNS. In addition, we will take into account the selection of kernel parameters for NNS, which would further improve the performance of NNS.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 60970067, 61033013, 60872135 and 60803098.

References

1. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces versus Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Special Issue on Face Recognition 19, 711–720 (1997)
2. Candès, E., Romberg, J.: ℓ_1 -magic: Recovery of sparse signals via convex programming (October 2005), <http://www.acm.caltech.edu/l1magic/>
3. Chien, J.T., Wu, C.C.: Discriminant waveletfaces and nearest feature classifiers for face recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 24(12), 1644–1649 (2002)
4. Cohen, M.C., Paliwal, K.K.: Classifying microarray cancer datasets using nearest subspace classification. In: Chetty, M., Ngom, A., Ahmad, S. (eds.) *Third IAPR International Conference on Pattern Recognition in Bioinformatics*. LNCS, vol. 5265, Springer, Heidelberg (2008)
5. Duarte-Carvajalino, J.M., Sapiro, G.: Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. *IEEE Transactions on Image Processing* 18(7), 1395–1408 (2009)
6. Duda, R., Hart, P., Stork, D.: *Pattern Classification*, 2nd edn. John Wiley & Sons (2000)
7. Graham, D.B., Allinson, N.M.: Characterizing virtual Eigensignatures for general purpose face recognition. In: *Face Recognition: From Theory to Applications*. NATO ASI Series F, Computer and Systems Sciences, vol. 163, pp. 446–456 (1998)
8. Lee, K.C., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(5), 684–698 (2005)
9. Li, S.Z.: Face recognition based on nearest linear combinations. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 839–844. IEEE Computer Society, Washington, DC, USA (1998)
10. Li, S.Z., Lu, J.: Face recognition using the nearest feature line method. *IEEE Transactions on Neural Networks* 10(2), 439–443 (1999)
11. Samaria, F.S., Harter, A.C.: Parameterisation of a stochastic model for human face identification. In: *Proceedings of the 2nd IEEE International Workshop on Applications of Computer Vision*, Sarasota, Florida, pp. 138–142 (December 1994)
12. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(2), 210–226 (2009)
13. Zhang, L., Zhou, W.D., Chang, P.C.: Generalized nonlinear discriminant analysis and its small sample size problems. *Neurocomputing* 74, 568–574 (2011)
14. Zhang, L., Zhou, W.D., Jiao, L.C.: Hidden space support vector machines. *IEEE Transactions on Neural Networks* 16(6), 1424–1434 (2004)
15. Zhang, L., Zhou, W.D., Jiao, L.C.: Wavelet support vector machine. *IEEE Transactions on Systems, Man, and Cybernetics - Part B* 34(1), 34–39 (2004)
16. Zhang, L., Zhou, W.D., Jiao, L.C.: Support vector machines based on the orthogonal projection kernel of father wavelet. *International Journal of Computational Intelligence and Applications* 5(3), 283–303 (2005)
17. Zhou, W., Zhang, L., Jiao, L.: Hidden Space Principal Component Analysis. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) *PAKDD 2006*. LNCS (LNAI), vol. 3918, pp. 801–805. Springer, Heidelberg (2006), <http://www.springerlink.com/content/2102835260t2m2x6/fulltext.pdf>
18. Zhou, X.F., Jiang, W.H., Shi, Y.: Credit risk evaluation by using nearest subspace method. *Procedia Computer Science* 1, 2443–2449 (2010)