I.J. Image, Graphics and Signal Processing, **2010, 2, 1-10**
Published Online December 2010 in MECS (http://www.mecs-press.org/)

Matrix-based Kernel Method for Large-scale Data Set

Weiya Shi
School of Information Science and Engineering
Henan University of Technology, Zhengzhou, China
Email: wyshi@fudan.edu.cn

Abstract— **In the computation process of many kernel methods, one of the important step is the formation of the kernel matrix. But the size of kernel matrix scales with the number of data set, it is infeasible to store and compute the kernel matrix when faced with the large-scale data set. To overcome computational and storage problem for large-scale data set, a new framework, matrix-based kernel method, is proposed. By initially dividing the large scale data set into small subsets, we could treat the *autocorrelation matrix* of each subset as the special computational unit. A novel polynomial-matrix kernel function is then adopted to compute the similarity between the data matrices in place of vectors. The proposed method can greatly reduce the size of kernel matrix, which makes its computation possible. The effectiveness is demonstrated by the experimental results on the artificial and real data set.**

Index Terms—**matrix, kernel, autocorrelation, computation**

I. Introduction

In the field of machine learning, there are many successful methods to be used for feature extraction and dimension reduction. These methods are generally divided into two categories: linear methods and nonlinear methods. The commonly used linear methods include Principal component analysis (PCA) [1][2], Linear discriminate analysis (LDA) [3] and so on. Support Vector Machine (SVM) [4] and kernel principal component analysis (KPCA) [5] is the frequently used nonlinear methods. These methods have been used in many complex applications, such as face recognition, image compression, etc.

Principal component analysis (PCA) is a classical method for feature extraction and dimension reduction [6]. It uses the dimensions with larger variances and neglects the less important components. Linear discriminate analysis (LDA) is a classical technique for feature extraction and dimension reduction. It finds the subspace which maximizes the between-class and minimized the intra-class scatter matrices. Although these linear methods have been successfully used in many applications, it does not work well in nonlinear data distribution. Therefore, it is necessary to generalize the linear methods for the nonlinear structure.

Vapnik et al. [4] firstly introduced kernel method [7] into Support Vector Machine. After that, it has successfully generalized to kernel principal component analysis, Generalized discriminant analysis [8] and other algorithms. Its main idea is to map the data set from the input space into high-dimensional feature space. Thus, the nonlinear components can be extracted using the traditional linear algorithm in feature space. The kernel trick is used to calculate the inner product between data set without knowing the explicit mapping function. These nonlinear methods have been used in classification, regression and supervising learning [9][10][11], etc.

In the computation process, many kernel methods need to compute the kernel matrix for all samples. For example, SVM uses all the training samples to learn hyperplane to maximize the separating margin. It needs to solve the quadratic programming (QP) problems, which time and space complexity is $O(m^3)$ and $O(m^2)$ respectively, where m is the number of data samples. The standard KPCA generally needs to eigen-decompose the kernel matrix (called Gram matrix) , which is acquired using the kernel function. It must firstly store the kernel matrix of all data, which takes the space complexity of $o(m^2)$. In addition, it needs the time complexity of $o(m^3)$ to extract the kernel principal components. But traditional kernel function is based on the inner product of data vectors, the size of kernel matrix scales with the number of data set. When faced with the large-scale data set, it is infeasible to store and compute the kernel matrix because of the limited storage capacity. However, the ever-growing large-scale data set needs to be processed in many applications, such as data mining, network data detection and video retrieval. Consequently, some approaches must be adopted to account for the inconvenience.

In order to solve the problem of the large-scale data set, some methods have been proposed to reduce the time and space complexities. In general, these approaches are classified into two categories: the sampling and non-sampling based approaches.

For the sampling based approaches, Zheng [12] proposed to partition the data set into several small-scale data sets and handle them, respectively. Some representative data [13] are chosen to approximate the original data set. Some approximation algorithms [14][15][16][17][18] are also proposed to extract the nonlinear components. The major difference between these methods lies in the sampling way. One disadvantage is that these methods will lose some information in the sampling process. Aside from that, it is time-consuming to search for the representative data.

For the non-sampling based approaches, an iterative procedure is proposed to estimate the kernel principal components by kernelizing the generalize Hebbian algorithm [19]. But the convergence is slow and cannot be guaranteed. Cauwenberghs & Poggio [20] gave an incremental algorithm for SVM. Wu et al. [21] also proposed the Kronecker factorization approach to approximate the kernel matrix by the Kronecker product of two smaller matrices, and then extracted the nonlinear components. Osuna et al. [22] used decomposition methods to break down large QP problems into many smaller QP sub-problems.

In this paper, we propose a new framework, matrix-based kernel methods, which can effectively solve the problem of large-scale data set. We have given elementary application on KPCA [23] and SVM [24]. In this paper, we will give further demonstration and discussion. The core idea of matrix-based kernel methods is firstly to divide the large scale data set into small subsets, each of which can produce the 1-order and 2-order statistical quantity (mean and autocorrelation matrix). For the 1-order statistical quantity, the traditional kernel method can be used to compute the kernel matrix. Because the 2-order statistical quantity is a matrix, we proposed a novel polynomial-matrix kernel function to compute the similarity between 2-order statistical quantities. Because the number of subsets is less than the number of samples, the size of kernel matrix can be greatly reduced. The small size of kernel matrix makes the computation and storage of large-scale data set possible. The effectiveness of the proposed method is demonstrated by the experimental results on the artificial and real data set.

The rest of this paper is organized as follows: section II gives reviews of kernel methods. Section III describes the proposed algorithm in detail. The experimental evaluation of the proposed method is given in the section IV. Finally we conclude with a discussion.

.

II. REVIEW OF KERNEL METHODS

Let $X=(x_1,x_2,...,x_m)$ be the data matrix in input space, where $x_i, i=1,2,...,m$, is a n-dimensional vector and m is the number of data samples. There exists a mapping function ϕ, which projects the data into high-dimensional (even infinite dimensional) Reproducing Kernel Hibert Space (RKHS).

$$\begin{aligned}\phi: \quad & \Re^n \to \mathrm{F} \\ & x_i \mapsto \phi(x_i)\end{aligned} \tag{1}$$

Using mapping function ϕ, we can get the projected data set $\Phi(X)=(\phi(x_1),\phi(x_2),...,\phi(x_m))$ in feature space. In practice, the mapping function ϕ needs not to be known explicitly but performed implicitly via kernel trick. A positive definite kernel function $\kappa(.,.)$ is used to calculate the dot product between mapped sample vectors, where $\kappa(.,.)$ is given by $\kappa(.,.)=\kappa(x_i,x_j)=\phi(x_i)^T\phi(x_j)$. The polynomial and Gaussian kernel are two widely used kernel function, given by:

$$\begin{cases}\kappa(.,.)=\phi(x_i)^T\phi(x_j)=(x_i^T x_j)^d \\ \kappa(.,.)=\phi(x_i)^T\phi(x_j)=\exp(-\dfrac{\|x_i-x_j\|^2}{2\sigma^2})\end{cases} \tag{2}$$

Where d is the degree of the polynomial kernel function and σ is the width parameter of the Gaussian kernel function.

Many algorithms based on kernel method were proposed to treat with the nonlinear datum. We only give some derivation of KPCA and other algorithms are similar.

In mapping feature space, the covariance matrix is given as follows:

$$C=\frac{1}{m}\sum_{i=1}^{m}\phi(x_i)^T\phi(x_i) \tag{3}$$

It accords with the following equation:

$$Cv=\lambda v \tag{4}$$

Where v and λ are corresponding eigenvector and eigenvalue of covariance matrix. It is an intractable task to solve the eigen-equation directly, which can be overcome via kernel trick. The solution v can be expanded using all the mapping sample vectors in the data set $\Phi(X)=(\phi(x_1),\phi(x_2),...,\phi(x_m))$ as:

$$v=\sum_{i=1}^{m}\alpha_i\phi(x_i) \tag{5}$$

By substituting Eq.3, Eq.5 into Eq.4, we can get the following formula:

$$K\alpha=m\lambda\alpha \tag{6}$$

Where α is span coefficient, K is Gram matrix denoted as $K=\Phi(X)^T\Phi(X)=(k_{ij})_{1\le i,j\le m}$. The entry of Gram matrix is $k_{ij}=k(x_i,x_j)$. It is proven [25] that the Gram matrix is positive semi-definite. To compute the kernel principal components, the traditional method is to diagonalizable Gram matrix K using eigen-decomposition technique.

Once the eigenvector α has been achieved, we can achieve the kernel principal components V using Eq.5. For a test sample x, the nonlinear feature is given:

$$(v,\phi(x))=\sum_{i=1}^{m}\alpha_i(\phi(x_i)\cdot\phi(x))$$
$$=\sum_{i=1}^{m}\alpha_i k(x_i,x) \quad (7)$$

In the process of whole derivation, it is assumed that the data have zero mean, if it is not, we can get the centering matrix $\tilde{K}=K-1_mK-K1_m+1_mK1_m$, where $1_m=(\frac{1}{m})_{m\times m}$.

III. THE PROPOSED ALGORITHM

The data set $X=(x_1,x_2,...,x_m)$ is firstly divided into M subsets $X_i(i=1,...,M)$, each of which consists of about $k=m/M$. Without loss of generality, it is denoted:

$$X_1=(x_1,...x_k),X_2=(x_{k+1},...x_{2k}),...,$$
$$X_M=(x_{(M-1)k+1},...x_m) \quad (8)$$

Accordingly, $X=\bigcup_{i=1}^{M}X_i=(X_1,X_2,...,X_M)$.

A. Computing 1-order statistical quantity

First, we compute 1-order statistical quantity (mean) for each subset. It is given as follows:

$$X_1^{center}=\frac{1}{k}\sum_{i=1}^{k}x_i,\cdots,X_M^{center}$$
$$=\frac{1}{m-(M-1)k}\sum_{i=(M-1)k+1}^{m}x_i \quad (9)$$

Having computing mean of each subset, the 1-order statistical quantity is still a vector. The standard kernel method can be used to map it to high dimensional space.

B. Computing the autocorrelation matrix of subset

Similarly, 2-order statistical quantity (autocorrelation matrix) for each subset can be computed. The autocorrelation matrix can be defined as follows:

$$\Sigma_1=X_1X_1{}^T,\Sigma_2=X_2X_2{}^T,...,\Sigma_M=X_MX_M{}^T \quad (10)$$

The data set $X=(X_1,X_2,...,X_M)$ is then transformed into $\Sigma=(\Sigma_1,\Sigma_2,...,\Sigma_M)$, where Σ_i is $n\times n$ autocorrelation matrix. Because the traditional kernel method is based on vector, the 2-order statistical quantity is a matrix. We must find some way of approaching the problem.

In this circumstance, we can treat them as the special computational units in input space. The data autocorrelation matrix can also be considered as generalized form of data vector. It is shown [6] that the autocorrelation matrix contains the statistical information between samples.

Thus, the special computational unit can be projected into high-dimensional (even infinite dimensional) Reproducing Kernel Hibert Space (RKHS) using a mapping function ϕ.

$$\phi: \qquad \Re^{n\times n}\rightarrow \mathrm{F}$$
$$\Sigma_i\mapsto\phi(\Sigma_i) \quad (11)$$

After having projected into feature space, the data set can be represented as $\Phi(\Sigma)=(\phi(\Sigma_1),\phi(\Sigma_2),...,\phi(\Sigma_M))$.

C. A novel polynomial-matrix kernel function

In this situation, we also need not know the mapping function explicitly, but perform it via kernel trick. According to the Eq.2, traditional kernel function is based on the inner product of data vector. Because the special computational unit is now based on matrix, the kernel function cannot be used. In order to compute the similarity between the mapping computational units in feature space, a positive definite kernel function needs to be denoted.

Considering to characteristic of polynomial kernel function, a novel polynomial-matrix kernel function is denoted as:

$$k(.,.)=k(\Sigma_i,\Sigma_j)$$
$$=(\phi(\Sigma_i)\cdot\phi(\Sigma_i))=\|\Sigma_i.*\Sigma_j\|_B^D \quad (12)$$

Where $\|.\|_B=\|(.).^{1/2}\|_F$ ($\|.\|_F$ is the Frobenius norm of matrix), $.*$ denotes the component-wise multiplication of matrices, $.^{1/2}$ means the component-wise square roots of matrices, and D is the degree of the polynomial-matrix kernel function. Now, we will investigate if the polynomial-matrix kernel function has some relationship with the polynomial kernel one.

Theorem: When each subset has one sample, the polynomial kernel function based on the data vector equals to twice of the polynomial-matrix one based on the autocorrelation matrix. That means the degree d is the twice of the degree D.

Proof:

When each subset contains one sample, the autocorrelation matrix $\sum_i = X_i X_i^T$. Using the polynomial-matrix kernel function, it follows:

$$k(\sum_i, \sum_j) = \| \sum_i .* \sum_j \|_B^D$$

$$= \| x_i x_i^T .* x_j x_j^T \|_B^D$$

$$= \left\| \begin{pmatrix} x_{i1}^2 & x_{i1}x_{i2}\cdots & x_{i1}x_{in} \\ \vdots & \ddots & \vdots \\ x_{in}x_{i1} & x_{in}x_{i2}\cdots & x_{in}^2 \end{pmatrix} .* \begin{pmatrix} x_{j1}^2 & x_{j1}x_{j2}\cdots & x_{j1}x_{jn} \\ \vdots & \ddots & \vdots \\ x_{jn}x_{j1} & x_{jn}x_{j2}\cdots & x_{jn}^2 \end{pmatrix} \right\|_B^D$$

$$= ((x_{i1}^2 x_{j1}^2 + x_{i1}x_{i2}x_{j1}x_{j2} +, ..., + x_{in}^2 x_{jn}^2))^D$$

$$= ((\sum_{k=1}^n x_{ik}x_{jk})^2)^D$$

$$= (x_i^T x_j)^{2D} = k(x_i, x_j)^{2D} = k(x_i, x_j)^d$$

, and the theorem is derived.

In other words, the polynomial kernel function is the extreme case of the polynomial-matrix one, when each subset comprises only one sample. The theorem also shows the kernel matrix computed using two different kernel functions is same at this time, which can give the same classified result when used in the kernel method.

We have given the matrix-based kernel method. Now, we will give specific application for some kernel-based algorithms ---support vector machine and kernel principal component analysis. Other kernel algorithms can be derived similarly.

D. Support Matrix Machine (SMM)

Let $X^+ = (x_1, x_2, ..., x_{m^+})$ and $X^- = (x_{m^++1}, x_{m^++2}, ..., x_{m^++m^-})$ be the positive and negative samples, respectively, where $x_i, i = 1, 2, \cdots, m^+ + m^-$, is a n-dimensional vector and m^+, m^- is the number of positive and negative samples separately. The label y_i corresponding to sample x_i lies in {1,-1}. For the proposed method, positive and negative samples are firstly clustered using Kmeans Algorithm, respectively. The N^+ positive and N^- negative training clusters can be obtained.

Because the data set is divided into many subsets in positive and negative samples, the number of subsets is less than the number of original data set. As a result, the large-scale data set is compressed by down-sampling the data using the matrix of each subset. In this situation, the size of kernel matrix can be greatly reduced from $(m^+ + m^-) \times (m^+ + m^-)$ to $(N^+ + N^-) \times (N^+ + N^-)$ by the novel polynomial-matrix kernel function. The small size of kernel matrix makes the computation and storage possible. Thus, the general SVM method can be used except that the vector was substituted for matrix.

In the case of training case, all the training clusters meet the following constraints:

$$y(\omega^T \phi(X_i) + b) \geq 1 - \xi_i, i = 1, 2, \cdots, (N^+ + N^-) \quad (13)$$

where mapping function ϕ projects the matrix subset into high-dimensional (even infinite dimensional) Reproducing Kernel Hibert Space (RKHS).

Similarly to the SVM, the SMM also learns hyper-plane to maximize the separating margin between the positive and negative subset. By introducing the regularization term, the cost function can be transformed into minimize the VC bound：

$$\min_{\omega} \frac{1}{2}\omega^T \omega + C \sum_{i=1}^{(N^+ + N^-)} \xi_i \quad (14)$$

The dual problem can be transformed like the SVM, which can be given as:

$$\max_{\omega} \sum_{i=1}^{(N^+ + N^-)} \alpha_i - \frac{1}{2} \sum_{i=1}^{(N^+ + N^-)} \sum_{i=1}^{(N^+ + N^-)} y_i y_j \alpha_i \alpha_j \phi(X_i)^T \phi(X_i)$$

$$s.t. \quad 1 \leq \alpha_i \leq C, i = 1, \cdots, k$$

$$\sum_{i=1}^{(N^+ + N^-)} \alpha_i y_i = 0$$

Using the quadratic programming (QP) solver, the parameter $\alpha_i, i = 1, \cdots, k$ can be easily obtained.

In the case of testing case, for a test sample x, its autocorrelation matrix $\Sigma_x = xx^T$ and following decision function can be used to predict it:

$$f(x) = \text{sgn}(\sum_{i=1}^{(N^+ + N^-)} \alpha_i y_i \phi(X_i)^T \phi(X)) \quad (15)$$

From above derivation, it shows that the proposed SMM has the same framework as the SVM except that the vector is replaced with matrix.

E. Kernel Principal Component Analysis based on 2-order statistical quantity

Because the data set is divided into many subsets, the number of subsets is less than the number of original data set. As a result, the large-scale data set is compressed by down-sampling the data using the matrix of each subset. In this situation, the size of kernel matrix can be greatly reduced from $m \times m$ to $M \times M$ by the novel polynomial-matrix kernel function. Thus, the small size of kernel matrix makes the computation and storage possible.

Currently, the mapped data set is $\Phi(\Sigma) = (\phi(\Sigma_1), \phi(\Sigma_2), ..., \phi(\Sigma_M))$ in feature space. The similar Equation for nonlinear component analysis can be derived. The covariance matrix is given as follows:

$$C = \frac{1}{M}\sum_{i=1}^{M}\phi(\Sigma_i)^T\phi(\Sigma_i) \quad (16)$$

It also accords with the eigen-equation:

$$Cv = \lambda v \quad (17)$$

But the eigenvector is now expanded using the linear sum of projected matrix in $\Phi(\Sigma) = (\phi(\Sigma_1), \phi(\Sigma_2), ..., \phi(\Sigma_M))$ as:

$$v = \sum_{i=1}^{M}\alpha_i\phi(\Sigma_i) \quad (18)$$

By substituting Eq.14, Eq.16 into Eq.15, we can get the following formula:

$$K\alpha = m\lambda\alpha \quad (17)$$

Where α is span coefficient, K is Gram matrix denoted as $K = \Phi(\Sigma)^T\Phi(\Sigma) = (k_{ij})_{1\le i,j\le M}$. The entry of Gram matrix is $k_{ij} = k(\Sigma_i, \Sigma_j)$. After having got the eigenvector α, the kernel principal components v can be achieved using Eq.16.

For a test sample x, its autocorrelation matrix is $\Sigma_x = xx^T$. The nonlinear feature is then given:

$$(v, \phi(\Sigma_x)) = \sum_{i=1}^{M}\alpha_i(\phi(\Sigma_i)\cdot\phi(\Sigma_x)) = \sum_{i=1}^{M}\alpha_i k(\Sigma_i, \Sigma_x)$$

In the process of whole derivation, it is assumed that the special computational unit, autocorrelation matrix of data subset, has zero mean; otherwise, it is easy to derive the centering kernel matrix:

$$\begin{aligned}\tilde{k}(\Sigma_i, \Sigma_j) &= \|(\Sigma_i - \frac{1}{M}\Sigma).*(\Sigma_j - \frac{1}{M}\Sigma)\|_C^D \\ &= \|(\Sigma_i .* \Sigma_i - \frac{1}{M}\Sigma_i .* \Sigma - \frac{1}{M}\Sigma .* \Sigma_i \\ &+ \frac{1}{M^2}\Sigma * \Sigma)\|_C^D \\ &= (K - 1_M K - K1_M + 1_M K1_M)_{ij}\end{aligned}$$

Therefore, the centering kernel matrix is $\tilde{K} = K - 1_M K - K1_M + 1_M K1_M$, where

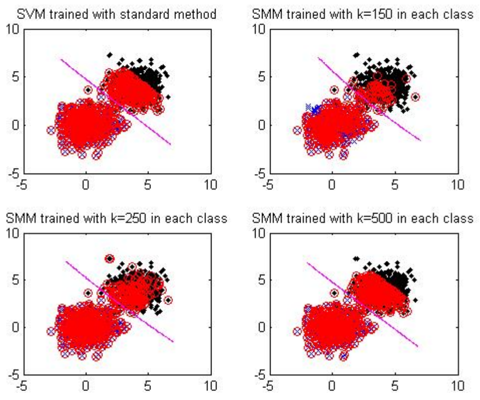

Figure 1. Learning result on the 2-D toy problem obtained from the standard SVM (upper left) and SMM (upper right, lower left, lower right)

$$1_M = (\frac{1}{M})_{M\times M}.$$

The computation process of proposed Kernel Principal Component Analysis based on 2-order statistical quantity is shown in Algorithm 1, Kernel Principal Component Analysis based on 1-order statistical quantity is identical to the traditional KPCA.

IV. EXPERIMENT

To demonstrate the effectiveness of the proposed kernel methods, we do some experiments using support matrix machine and kernel principal component analysis for large scale data set.

A. *Experiment using support matrix machine*

Firstly, we perform the experiments on two-dimensional toy problem using the standard SVM and the proposed SMM method, respectively. In addition, we use USPS data set to validate the feasibility of SMM. The polynomial kernel $k(x, y) = (x^T y)^d$ (where d is the degree) is used in standard SVM. The polynomial-matrix kernel function $k(\Sigma_i, \Sigma_j) = \|\Sigma_i .* \Sigma_j\|_C^D$ (where D is the degree) is used in SMM. The sample of cluster in the Kmeans algorithm is adjusted according to the training samples.

Toy examples:

We firstly use 2-dimensional toy problem to demonstrate the effectiveness of SMM. 500 positive and 500 negative samples are generated, where follows a Gaussian distributions. In the SMM, the data set of each class is divided into 150, 250,500 subsets, respectively. The degree d and D of two kernel function equal to 2 and 1, respectively.

The experiment results are given in Fig. I. The support vectors were labeled in red bold. From the result, SMM can get almost similar performance with the standard SVM. The result shows the effectiveness of SMM, which can successfully learn hyper plane to maximize the separating margin.

USPS examples:

We also test the proposed method on real-world data. The US postal Service (USPS) data set {Available at http://www.kernel-machines.org.} is 256-dimensional handwritten digits '0'--'9'. It consists of 7291 training samples and 2007 testing samples. It is known the recognition rate for the dataset is only 2.5% for human [26]. In order to test the proposed method, a preprocessing step was firstly used with a Gaussian kernel with width 0.75 to smooth the image. For each method, we use the average result of 45 one-against-one classifiers. The degree d and D of the kernel function in two methods equal to 4 and 2, respectively.

TABLE I.
ERROR RATE OF TESTING SAMPLE ON THE USPS DATA SET USING THE STANDARD SVM AND THE PROPOSED SMM WITH THE NUMBER OF EACH CLASS K= 400, 500 AND THE SIZE OF THE SAMPLES IN EACH CLASS, RESPECTIVELY.TABLE TYPE STYLES

	SVM	SMM with different k		
		the size of samples in each class	*K=400*	*K=500*
Error Rate(%)	4.68	4.68	4.73	4.78

Table I gives the average testing errors obtained by the standard SVM and the proposed method. In the proposed SMM, the number of cluster in each class was set to 400, 500 and the size of the samples in each class, respectively. When the number of cluster in each class was equal to the size of the samples in each class, which means the class is still original dataset. Although the dataset is not clustered at this time, matrix is still used to replace vector in the computation process. According to the above theorem, the two methods can get the same result, which can be found from the table. In addition, it also shows that the performance based on matrix of cluster is comparable to standard SVM.

B. *Experiment using Kernel Principal Component Analysis*

To demonstrate the effectiveness of the proposed methods, we perform the experiments on two-dimensional toy problem using the standard KPCA (using the original Matlab source code [5] available at http://www.kernel-machines.org) and the proposed methods, respectively. In addition, we use USPS data set to validate the feasibility of proposed methods when standard eigen-decomposition technique cannot be applied. In order to differentiate from the standard KPCA, we abbreviate the method 1order-KPCA, which means Kernel Principal Component Analysis based on 1-order statistical quantity and shorten the method 2order-KPCA, which means Kernel Principal Component Analysis based on 2-order statistical quantity. The polynomial kernel $k(x, y) = (x^T y)^d$ (where d is the degree) is used in standard KPCA and 1order-KPCA. The polynomial-matrix kernel function

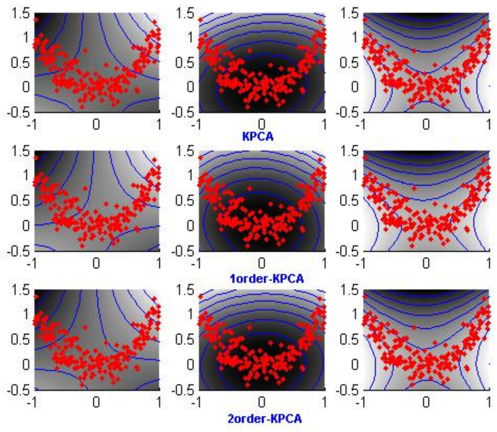

Figure 2. Contour image of first 3 principal components obtained from standard KPCA (the first row), 1order-KPCA (the second row) and 2order-KPCA(the third row).

$k(\Sigma_i, \Sigma_j) = \| \Sigma_i .* \Sigma_j \|_C^D$ (where D is the degree) is used in 2order-KPCA

Toy examples:

We firstly use 2-dimensional toy problem to demonstrate the effectiveness of proposed methods. The 200 2-dimensional data samples are generated, where x-values are uniformly distributed in [-1,1] and y-values are given by $y = x^2 + \eta$ (η is the normal noise with standard deviation 0.2). In the 1order-KPCA and 2order-KPCA, the data set is divided into 100 subsets, each of which contains 2 samples. The degree d and D for two kernel functions equal to 2 and 1, respectively.

The 1-order statistical quantity and 2-order statistical quantity was firstly computed. The experiment results are given in Fig.2. It gives contour lines of constant value of the first 3 principal components, where the gray values represent the feature value. From the result, the 1order-KPCA and 2order-KPCA can get almost similar performance with the standard KPCA. The result shows the effectiveness of proposed methods, which can successfully extract the nonlinear components.

USPS examples:

Firstly, we randomly select 3000 training samples to extract the nonlinear feature. In the proposed methods, we divide 3000 training samples into some subsets with different size ($1 \le k \le 5$) in each subset. For each k, 10 independent runs are performed, where the data samples are randomly reordered. The classified results of 2007 testing samples are averaged.

TABLE II.
ERROR RATE OF 2007 TESTING SAMPLE USING PROPOSED METHODS (HAVING DEGREE D =1 AND DIFFERENT SAMPLE NUMBERS K IN EACH SUBSET) AND THE STANDARD KPCA (HAVING DEGREE d =2) WITH 3000 TRAINING SAMPLES.

Number of components	KPCA	1order-KPCA with d=2				
		K=1	K=2	K=3	K=4	K=5
32	7.17	7.17	6.88	7.13	7.08	7.22
64	6.98	6.98	6.98	7.08	7.13	7.13
128	7.17	7.17	7.47	7.22	7.47	7.52
256	7.22	7.22	6.93	7.13	7.22	7.13
512	7.42	7.42	7.32	7.13	7.21	7.03

(a). Result of 1order-KPCA

Number of components	KPCA	2order-KPCA with D=1				
		K=1	K=2	K=3	K=4	K=5
32	7.17	7.17	7.27	7.03	7.17	7.42
64	6.98	6.98	6.98	6.93	6.78	6.78
128	7.17	7.17	7.37	6.88	7.08	7.03
256	7.22	7.22	7.32	7.03	7.22	6.93
512	7.42	7.42	7.27	7.47	7.32	7.42

(b). Result of 2order-KPCA

Table II ~~Table IV gives the error rate of testing sample using standard KPCA and proposed methods with different number samples in each subset. Table II, Table III, and Table IV are the results of the polynomial-matrix kernel function under degree $D = 1, 1.5\ and\ 2$ for 2order-KPCA, respectively. It also gives the corresponding result of the polynomial kernel function under degree $d = 2, 3\ and\ 4$ for standard KPCA and 1order-KPCA, respectively. According to the aforementioned theorem, when each subset has one sample, the polynomial kernel function based on the data vector equals to twice of the polynomial-matrix one based on the matrix. In this situation, the result of 1order-KPCA and 2order-KPCA should equal to the result of the standard KPCA (one sample in each subset), which can be found in each Table. It also shows 1order-KPCA and 2order-KPCA with different number samples in each subset could generally achieve competitively classified result than the standard KPCA. The reason may be that the mean and autocorrelation matrix contains the statistical information between samples in each subset. In addition, the computation instability for the small size kernel matrix can be greatly reduced when performing its eigen-decomposition. To visualize the result more clear, we plot the recognize rate under different number of kernel principal components in Fig.3~~Fig5.

In addition, we also use all the training samples to extract the nonlinear feature. Because the size of Gram matrix is 7291×7291, it is impossible for standard KPCA algorithm to run in the normal hardware. Using the proposed methods, we firstly divide 7291 training samples into 1216 subsets, each of which consists of 6 samples (The last subset contains only 1 sample). Table 5 is the result of proposed method with 6 samples in each subset trained with all training samples. Here, the size of kernel matrix drops from 7291×7291 to 1216×1216, which can be easily stored and computed. As shown in Table Ⅴ, it can also be seen that 1order-KPCA and 2order-KPCA can achieve the right classified performance even the eigen-decomposition technique cannot work out when faced with large-scale data set. The result shows that the proposed method is more effective and efficient than standard KPCA.

IV. CONCLUSION

In this paper, an efficient matrix-based kernel method for large-scale data set is proposed. The method firstly divides the large scale data set into small subsets, each of which can produce autocorrelation matrix. Then the autocorrelation matrix can be treated as special computational units. The similarity between matrices can be computed using a novel polynomial-matrix kernel function. It can greatly reduce the size of kernel matrix, which can effectively solve the large scale problem.

Compared to other related methods, the proposed method is different in the following aspects: (a). the proposed method uses the matrix, while other methods use data vector, as the computational unit in data space. (b). the autocorrelation matrix contains the correlation information of sample subset, which can help the improvement of performance. (c). the proposed method can be easily implemented; while many other methods are more complicated.

We only consider the polynomial-matrix kernel function in the whole process. We will investigate other kernel function based on matrix. In addition, the optimal sample number in each subset deserves further study in our future work

ACKNOWLEDGMENT

This work was supported in part by Doctor Fund of Henan University of Technology under contract 2009BS013, and Innovation Scientists and Technicians Troop Construction Projects of Henan Province under contract 094200510009.

REFERENCES

[1] Y. Kirby and L. Sirovich, “Application of the Karhunen-loeve Procedure for the Characterization of Human Faces,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.12,no.1,pp.103--108, 1990

[2] M. Turk, A. Pentland, “Eigenfaces for Recognition,” *J. Cogn. Neurosci.*, pp.71--86, 1991

[3] P. Belhumeur, J. Hespanha, and K. D.J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence.*, 19:711–720, 1997

[4] C. Cortes, V. Vapnik, “Support Vector Networks,” *Machine. Learning*. Vol.20,pp.273-297, 1995.

[5] B. Scholkopf, A. J. Smola, and K.-R. Muller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” *Neural Comput.*, Vol.10,no.5, pp.1299--1319, 1998.

[6] K. Fukunaga, “Introduction to Statistical Pattern Recognition,” *Academic Press*, 1990.

[7] M. Aizerman, E. Braverman, L. Rozonoer, “Theoretical foundations of the potential function method in pattern recognition learning,” *Automat Remote Control*. vol. 25,pp.821-837,1964.

[8] G. Bandat and F. Anouar. Generalized discriminant analysis using a kernel method. *Neural comput.*, 12:2385–2404, 2000

[9] B. Scholkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A.J. SmolaI. S. Jacobs and C. P. Bean, “Input space versus feature space in kernel-based methods,” *IEEE Trans. Neural Network*.vol.10,pp.1000-1017,1999.

[10] M.-H. Yang, “Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition using Kernel Methods,” in *Proc. 5th IEEE Conf. Auto. Face Gesture Recognit*.,pp.215-220, 2002

[11] S. Mika, B. Scholkopf, A. Smola, K.R. Muller, M. Scholz, and G. Ratsch, “Kernel PCA and de-noising in Feature Spaces,” In *Adv. Neural Inf. Process. Syst.*, 1998

[12] W.M. Zheng, C.R. Zou, and L. Zhao “An Improved Algorithm for Kernel Principal Components Analysis,” *Neural Processing Letters*. Vol.22, pp.49--56, 2005

[13] V. France, and V. Hlavac, “Greedy Algorithm for a Training Set Reduction in the Kernel Methods,” In *IEEE Int. Conf. Comp. Anal. of Imags and Patterns*, pp.426--433. 2003.

[14] D. Achlioptas, M. McSherry, and B. Scholkopf, “Sampling techniques for kernel methods,” In *Adv. Neural Inf. Process. Syst.*, 2002

[15] C. Williams, and M. Seeger, “Using the Nystrom Method to Speed up Kernel Machine,” In *Adv. Neural Inf. Process. Syst.*., 2001

[16] R.Rosipal, and M. Girolami, “An Expectative-maximization Approach to Nonlinear Component Analysis,” *Neural Comput*, vol. 13, pp.505--510, 2001

[17] A. Smola and B. Scholkopf, “Sparse greedy matrix approximation for machine learning” *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 911–918). Standord, CA, USA., 2000.

[18] I. W. Tsang, J. T. Kwok and P. Cheung,” Core Vector Machines: Fast SVM training on very large data sets” *J. of Machine Learning Research*, vol.6,pp.:363– 392,2005

[19] K.I. Kim, M.O. Franz, and B. Scholkopf, “Iterative Kernel Principal Component Analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.27,no.9, pp1351--1366, 2005

[20] G. Cauwenberghs, and T. Poggio, “Incremental and decremental support vector machine learning,” In *Advanced Neural Information Processing Systems*. Cambridge, MA: MIT Press,2000

[21] G.. Wu, Z. Zhang, and E. Chang, “Kronecker Factorization for Speeding up Kernel Machines,” *SIAM Int. Conf. Data Mining(SDM)*, 2005

[22] E. Osuna, R. Freund, and F. Girosi, “An improved training algorithm for support vector machines,” Proc. of *the 1997 IEEE Workshop on Neural Networks for Signal Processing*,pp. 276–285,1997

[23] Shi Weiya, Guo Yue-Fei, and Xue Xiangyang. “Matrix-based Kernel Principal Component Analysis for Large-scale Data Set,” *International Joint Conference on Neural Network*, 2908-2913，2009 , USA

[24] Weiya. Shi and Dexian zhang, “SupportMatrix Machine for Large-scale data set”, *International Conference on Information Engineering and Computer Science,* ICIECS2009

[25] Scholkopf B and Smola A. *Learning with kernel: Support Vector Machines,Regularization,Optimization and Beyond,* MIT Press,2002

[26] J. Bromley and E. Sackinger,” Neural-network and k-nearest-neighbor classifiers.” *Technical Report* 11359-910819-16TM, AT&T.

Weiya Shi was born at Zhoukou, Henan,China in April 3rd,1973. He received his B.S. degree in Physics from ZhengZhou University, ZhengZhou, China, in 1998. He received his Ph.D. degree in Department of Computer Science and Engineering, Fudan University, Shanghai, China, in 2009. His current research interest includes pattern recognition, neural network and image processing.

He is now a teacher at the Henan University of Technology, Zhengzhou, China. He is associate professor in the field of computer science. His previous publications consist of “Matrix-based Kernel Principal Component Analysis for Large-scale Data Se”(IJCNN 2009), “SupportMatrix Machine for Large-scale data set” (ICIECS2009) and so on.

Dr. Shi is the Fellow of INNS. His hobbies include music, football and dance.

TABLE III.
ERROR RATE OF 2007 TESTING SAMPLE USING PROPOSED METHODS (HAVING DEGREE D=1.5 AND DIFFERENT SAMPLE NUMBERS K IN EACH SUBSET) AND THE STANDARD KPCA (HAVING DEGREE d =3) WITH 3000 TRAINING SAMPLES..

Number of components	KPCA	1order-KPCA with d=3				
		K=1	K=2	K=3	K=4	K=5
32	8.42	8.42	8.02	7.97	7.72	8.12
64	7.13	7.13	7.32	6.98	7.87	7.72
128	7.62	7.62	7.72	7.52	8.17	7.87
256	8.07	8.07	7.87	7.97	8.22	7.97
512	8.17	8.17	8.07	8.07	8.07	8.07

(a). Result of 1order-KPCA

Number of components	KPCA	2order-KPCA with D=1.5				
		K=1	K=2	K=3	K=4	K=5
32	8.42	8.42	7.87	8.07	8.02	8.17
64	7.13	7.13	7.13	7.17	7.13	7.42
128	7.62	7.62	7.57	7.27	7.17	7.62
256	8.07	8.07	7.77	7.82	7.97	7.72
512	8.17	8.17	8.37	8.07	8.17	8.27

(b). Result of 2order-KPCA

TABLE IV.
ERROR RATE OF 2007 TESTING SAMPLE USING PROPOSED METHODS (HAVING DEGREE D=2 AND DIFFERENT SAMPLE NUMBERS K IN EACH SUBSET) AND THE STANDARD KPCA (HAVING DEGREE d =4) WITH 3000 TRAINING SAMPLES.

Number of components	KPCA	1order-KPCA with d=4				
		K=1	K=2	K=3	K=4	K=5
32	7.17	7.17	6.88	7.13	7.08	7.22
64	6.98	6.98	6.98	7.08	7.13	7.13
128	7.17	7.17	7.47	7.22	7.47	7.52
256	7.22	7.22	6.93	7.13	7.22	7.13
512	7.42	7.42	7.32	7.13	7.21	7.03

(a). Result of 1order-KPCA

Number of components	KPCA	2order-KPCA with D=2				
		K=1	K=2	K=3	K=4	K=5
32	7.17	7.17	7.27	7.03	7.17	7.42
64	6.98	6.98	6.98	6.93	6.78	6.78
128	7.17	7.17	7.37	6.88	7.08	7.03
256	7.22	7.22	7.32	7.03	7.22	6.93
512	7.42	7.42	7.27	7.47	7.32	7.42

(b). Result of 2order-KPCA

TABLE V.
ERROR RATE OF 2007 TESTING SAMPLE USING 1ORDER-KPCA AND 2ORDER-KPCA (HAVING DIFFERENT DEGREE D) WITH ALL TRAINING SAMPLES. IN THE PROPOSED METHODS, THE DATA SET IS DIVIDED INTO 1216 SUBSETS, EACH OF WHICH CONSISTS OF 6 SAMPLES. THE STANDARD KPCA CANNOT PRODUCE THE RESULT BECAUSE OF STORAGE PROBLEM..

Number of components	D=2	D=3	D=4	D=5
32	5.93	6.48	7.42	7.82
64	5.63	5.93	6.73	7.52
128	5.73	5.98	6.33	7.13
256	5.93	5.63	6.44	6.58
512	5.83	5.73	6.18	6.38

(a). Result of 1order-KPCA

Number of components	D=2	D=3	D=4	D=5
32	6.08	6.88	7.87	9.42
64	5.78	6.13	7.37	8.17
128	5.73	6.18	6.93	7.47
256	5.48	6.13	6.48	7.37
512	5.73	6.03	6.38	6.68

(b). Result of 2order-KPCA

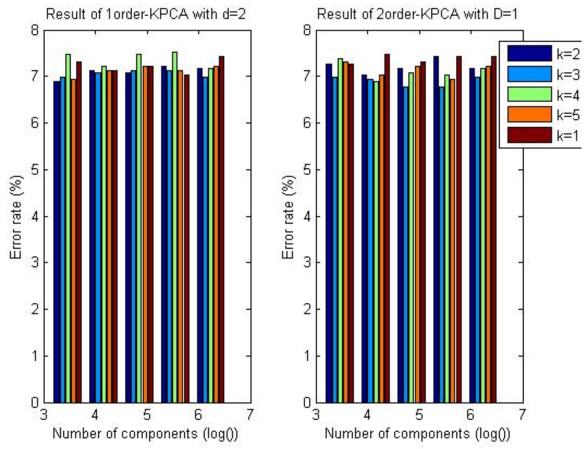

Figure 3. Performance of proposed methods using different number samples (k) in each subset under varying number of kernel principal components (using log scale) corresponding to Table Ⅱ..

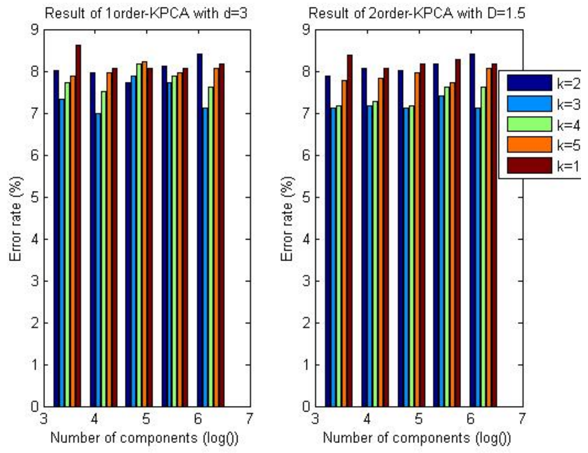

Figure 4. Performance of proposed methods using different number samples (k) in each subset under varying number of kernel principal components (using log scale) corresponding to Table Ⅲ.

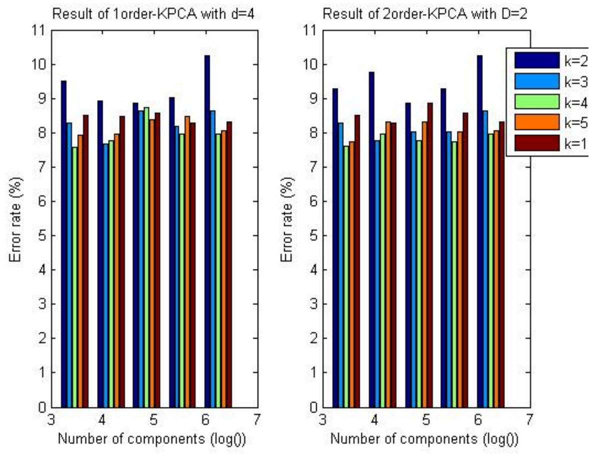

Figure 5. Performance of proposed methods using different number samples (k) in each subset under varying number of kernel principal components (using log scale) corresponding to Table Ⅳ.