Incremental Nonlinear PCA for Classification*

Byung Joo Kim[1] and Il Kon Kim[2]

[1] Youngsan University School of Network and Information Engineering, Korea
bjkim@ysu.ac.kr
[2] Kyungpook National University Department of Computer Science, Korea
ikkim@knu.ac.kr

Abstract. The purpose of this study is to propose a new online and nonlinear PCA(OL-NPCA) method for feature extraction from the incremental data. Kernel PCA(KPCA) is widely used for nonlinear feature extraction, however, it has been pointed out that KPCA has the following problems. First, applying KPCA to patterns requires storing and finding the eigenvectors of a kernel matrix, which is infeasible for a large number of data N. Second problem is that in order to update the eigenvectors with an another data, the whole eigenspace should be recomputed. OL-NPCA overcomes these problems by incremental eigenspace update method with a feature mapping function. According to the experimental results, which comes from applying OL-NPCA to a toy and a large data problem, OL-NPCA shows following advantages. First, OL-NPCA is more efficient in memory requirement than KPCA. Second advantage is that OL-NPCA is comparable in performance to KPCA. Furthermore, performance of OL-NPCA can be easily improved by relearning the data. For classification extracted features are used as input for least squares support vector machine. In our experiments we show that proposed feature extraction method is comparable in performance to a Kernel PCA and proposed classification system shows a high classification performance on UCI benchmarking data and NIST handwritten data set.

Keywords: Incremental nonlinear PCA, Kernel PCA, Feature mapping function, LS-SVM

1 Introduction

In many pattern recognition problem it relies critically on efficient data representation. It is therefore desirable to extract measurements that are invariant or insensitive to the variations within each class. The process of extracting such measurements is called *feature extraction*. Principal Component Analysis(PCA)[1] is a powerful technique for extracting features from possibly high-dimensional data sets. For reviews of the existing literature is described in [2][3][4]. Traditional PCA, however, has several problems. First PCA requires a batch computation

* This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea (02-PJ1-PG6-HI03-0004)

J.-F. Boulicaut et al. (Eds.): PKDD 2004, LNAI 3202, pp. 291–300, 2004.

step and it causes a serious problem when the data set is large i.e., the PCA computation becomes very expensive. Second problem is that, in order to update the subspace of eigenvectors with another data, we have to recompute the whole eigenspace. Finial problem is that PCA only defines a linear projection of the data, the scope of its application is necessarily somewhat limited. It has been shown that most of the data in the real world are inherently non-symmetric and therefore contain higher-order correlation information that could be useful[5]. PCA is incapable of representing such data. For such cases, nonlinear transforms is necessary. Recently kernel trick has been applied to PCA and is based on a formulation of PCA in terms of the dot product matrix instead of the covariance matrix[8]. Kernel PCA(KPCA), however, requires storing and finding the eigenvectors of a $N \times N$ kernel matrix where N is a number of patterns. It is infeasible method when N is large. This fact has motivated the development of incremental way of KPCA method which does not store the kernel matrix. It is hoped that the distribution of the extracted features in the feature space has a simple distribution so that a classifier could do a proper task. But it is point out that extracted features by KPCA are global features for all input data and thus may not be optimal for discriminating one class from others[6]. This has naturally motivated to combine the feature extraction method with classifier for classification purpose. In this paper we propose a new classifier for on-line and nonlinear data. Proposed classifier is composed of two parts. First part is used for feature extraction. To extract nonlinear features, we propose a new feature extraction method which overcomes the problem of memory requirement of KPCA by incremental eigenspace update method incorporating with an adaptation of feature mapping function. Second part is used for classification. Extracted features are used as input for classification. We take Least Squares Support Vector Machines(LS-SVM)[7] as a classifier. LS-SVM is reformulations to the standard Support Vector Machines(SVM)[8]. SVM typically solving problems by quadratic programming(QP). Solving QP problem requires complicated computational effort and needs more memory requirement. LS-SVM overcomes this problem by solving a set of linear equations in the problem formulation. Paper is composed of as follows. In Section 2 we will briefly explain the incremental eigenspace update method. In Section 3 nonlinear PCA is introduced and to make nonlinear PCA incrementally feature mapping function is explained. Proposed classifier combining LS-SVM with proposed feature extraction method is described in Section 4. Experimental results to evaluate the performance of proposed classifier is shown in Section 5. Discussion of proposed classifier and future work is described in Section 6.

2 Incremental Eigenspace Update Method

In this section, we will give a brief introduction to the method of incremental PCA algorithm which overcomes the computational complexity and memory requirement of standard PCA. Before continuing, a note on notation is in order. Vectors are columns, and the size of a vector, or matrix, where it is important, is

denoted with subscripts. Particular column vectors within a matrix are denoted with a superscript, while a superscript on a vector denotes a particular observation from a set of observations, so we treat observations as column vectors of a matrix. As an example, A^i_{mn} is the ith column vector in an $m \times n$ matrix. We denote a column extension to a matrix using square brackets. Thus $[A_{mn}b]$ is an $(m \times (n+1))$ matrix, with vector b appended to A_{mn} as a last column.
To explain the incremental PCA, we assume that we have already built a set of eigenvectors $U = [u_j], j = 1, \cdots, k$ after having trained the input data $\mathbf{x}_i, i = 1, \cdots, N$. The corresponding eigenvalues are Λ and $\bar{\mathbf{x}}$ is the mean of input vector. Incremental building of eigenspace requires to update these eigenspace to take into account of a new input data. Here we give a brief summarization of the method which is described in [9]. First, we update the mean:

$$\overline{x}' = \frac{1}{N+1}(N\overline{x} + x_{N+1}) \tag{1}$$

We then update the set of Eigenvectors to reflect the new input vector and to apply a rotational transformation to U. For doing this, it is necessary to compute the orthogonal residual vector $\hat{h} = (Ua_{N+1} + \overline{x}) - x_{N+1}$ and normalize it to obtain $h_{N+1} = \frac{h_{N+1}}{\|h_{N+1}\|_2}$ for $\| h_{N+1} \|_2 > 0$ and $h_{N+1} = 0$ otherwise. We obtain the new matrix of Eigenvectors U' by appending h_{N+1} to the eigenvectors U and rotating them :

$$U' = [U, h_{N+1}]R \tag{2}$$

where $R \in \mathbf{R}_{(\mathbf{k+1})\times(\mathbf{k+1})}$ is a rotation matrix. R is the solution of the eigenproblem of the following form:

$$DR = R\Lambda' \tag{3}$$

where Λ' is a diagonal matrix of new Eigenvalues. We compose $D \in \mathbf{R}_{(\mathbf{k+1})\times(\mathbf{k+1})}$ as:

$$D = \frac{N}{N+1}\begin{bmatrix} \Lambda & 0 \\ 0^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix} aa^T & \gamma a \\ \gamma a^T & \gamma^2 \end{bmatrix} \tag{4}$$

where $\gamma = h^T_{N+1}(x_{N+1} - \bar{x})$ and $a = U^T(x_{N+1} - \bar{x})$. Though there are other ways to construct matrix D[10,11], the only method ,however, described in [9] allows for the updating of mean.

2.1 Eigenspace Updating Criterion

The incremental PCA represents the input data with principal components $a_{i(N)}$ and it can be approximated as follows:

$$\widehat{x}_{i(N)} = Ua_{i(N)} + \bar{x} \tag{5}$$

To update the principal components $a_{i(N)}$ for a new input x_{N+1} , computing an auxiliary vector η is necessary. η is calculated as follows:

$$\eta = \left[U\widehat{h}_{N+1}\right]^T (\overline{x} - \overline{x}') \tag{6}$$

then the computation of all principal components is

$$a_{i(N+1)} = (R')^T \begin{bmatrix} a_{i(N)} \\ 0 \end{bmatrix} + \eta, \qquad i = 1, \cdots, N+1 \tag{7}$$

The above transformation produces a representation with $k+1$ dimensions. Due to the increase of the dimensionality by one, however, more storage is required to represent the data. If we try to keep a k-dimensional eigenspace, we lose a certain amount of information. It is needed for us to set the criterion on retaining the number of eigenvectors. There is no explicit guideline for retaining a number of eigenvectors. Here we introduce some general criteria to deal with the model's dimensionality:

- Adding a new vector whenever the size of the residual vector exceeds an absolute threshold;
- Adding a new vector when the percentage of energy carried by the last Eigenvalue in the total energy of the system exceeds an absolute threshold, or equivalently, defining a percentage of the total energy of the system that will be kept in each update;
- Discarding Eigenvectors whose Eigenvalues are smaller than a percentage of the first Eigenvalue;
- Keeping the dimensionality constant.

In this paper we take a rule described in b). We set our criterion on adding an Eigenvector as $\lambda_{k+1}^{'} > 0.7\bar{\lambda}$ where $\bar{\lambda}$ is a mean of the λ. Based on this rule, we decide whether adding $u_{k+1}^{'}$ or not.

3 Online and Nonlinear PCA

A prerequisite of the incremental eigenspace update method is that it has to be applied on the data set. Furthermore incremental PCA builds the subspace of eigenvectors incrementally, it is restricted to apply the linear data. But in the case of KPCA this data set $\Phi(x^N)$ is high dimensional and can most of the time not even be calculated explicitly. For the case of nonlinear data set, applying feature mapping function method to incremental PCA may be one of the solutions. This is performed by so-called *kernel-trick*, which means an implicit embedding to an infinite dimensional Hilbert space[8](i.e. feature space) F.

$$K(x, y) = \Phi(x) \cdot \Phi(y) \tag{8}$$

Where K is a given kernel function in an input space. When K is semi positive definite, the existence of Φ is proven[8]. Most of the case, however, the mapping Φ is high-dimensional and cannot be obtained explicitly except polynomial kernel function. We can easily derive polynomial feature mapping function as following procedure. Let d =2, $x = (x_1, x_2), y = (y_1, y_2))$ then $(x \cdot y)^2 = (x_1^2, \sqrt{2}x_1x_2, x_2^2)(y_1^2, \sqrt{2}y_1y_2, y_2^2)^T = (\phi(x) \cdot \phi(y))$. Now it is then easy to see that

$(\phi(x)) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$. In case of polynomial feature mapping function there is no difference in performance according to degree d[8]. By this result, we only need to apply the polynomial feature mapping function to one data point at a time and do not need to store the $N \times N$ kernel matrix.

4 Proposed Classification System

In earlier Section 3 we proposed an incremental nonlinear PCA method for nonlinear feature extraction. Feature extraction by incremental nonlinear PCA effectively acts a nonlinear mapping from the input space to an implicit high dimensional feature space. It is hoped that the distribution of the mapped data in the feature space has a simple distribution so that a classifier can classify them properly. But it is point out that extracted features by nonlinear PCA are global features for all input data and thus may not be optimal for discriminating one class from others. For classification purpose, after global features are extracted using they must be used as input data for classification. There are many famous classifier in machine learning field. Among them neural network is popular method for classification and prediction purpose. Traditional neural network approaches, however have suffered difficulties with generalization, producing models that can overfit the data. To overcome the problem of classical neural network technique, support vector machines(SVM) have been introduced. The foundations of SVM have been developed by Vapnik and it is a powerful methodology for solving problems in nonlinear classification. Originally, it has been introduced within the context of statistical learning theory and structural risk minimization. In the methods one solves convex optimization problems, typically by quadratic programming(QP). Solving QP problem requires complicated computational effort and need more memory requirement. LS-SVM overcomes this problem by solving a set of linear equations in the problem formulation. LS-SVM method is computationally attractive and easier to extend than SVM.

5 Experiment

To evaluate the performance of proposed classification system, experiment is performed by following step. First we evaluate the feature extraction ability of online and nonlinear PCA(OL-NPCA). The disadvantage of incremental method is their accuracy compared to batch method even though it has the advantage of memory efficiency. So we shall apply proposed method to a simple toy data and image data set which will show the accuracy and memory efficiency of incremental nonlinear PCA compared to APEX model proposed by Kung[15] and batch KPCA. Next we will evaluate the training and generalization ability of proposed classifier on UCI benchmarking data and NIST handwritten data set. To do this, extracted features by OL-NPCA will be used as input for LS-SVM.

5.1 Toy Data

To evaluate the feature extraction accuracy and memory efficiency of OL-NPCA compared to APEX and KPCA we take nonlinear data used by Scholkoff[5]. Totally 41 training data set is generated by:

$$y = x^2 + 0.2\varepsilon : \ \varepsilon \ from \ N(0,1), x = [-1,1] \tag{9}$$

First we compare feature extraction ability of OL-NPCA to APEX model. APEX model is famous principal component extractor based on Hebbian learning rule. Applying toy data to OL-NPCA we finally obtain 2 eigenvectors. To evaluate the performance of two methods on same condition, we set 2 output nodes to standard APEX model.

Table 1. Performance evaluation of OL-NPCA and APEX

Method	Iteration	Learning Rate	$\| w_1 \|$	$\| w_2 \|$	$\cos\theta_1$	$\cos\theta_2$	MSE
APEX	50	0.01	0.6827	1.4346	0.9993	0.7084	14.8589
APEX	50	0.05				do not converge	
APEX	500	0.01	1.0068	1.0014	0.9995	0.9970	4.4403
APEX	500	0.05	1.0152	1.0470	0.9861	0.9432	4.6340
APEX	1000	0.01	1.0068	1.0014	0.9995	0.9970	4.4403
APEX	1000	0.05	1.0152	1.0470	0.9861	0.9432	4.6340
OL-NPCA	100		1	1	1	1	0.0223

In table 1 we experimented APEX method on various conditions. Generally neural network based learning model has difficulty in determining the parameters; for example learning rate, initial weight value and optimal hidden layer node. This makes us to conduct experiments on various conditions. $\| w \|$ is norm of weight vector in APEX and $\| w \| = 1$ means that it converges stable minimum. $cos\theta$ is angle between Eigenvector of KPCA and APEX, OL-NPCA respectively. $cos\theta$ of Eigenvector can be a factor of evaluating accuracy how much OL-NPCA and APEX is close to accuracy of KPCA. Table 1 nicely shows the two advantages of OL-NPCA compared to APEX: first, performance of OL-NPCA is better than APEX; second, the performance of OL-NPCA is easily improved by re-learning. Another factor of evaluating accuracy is reconstruction error. Reconstruction error is defined as the squared distance between the image of x_N and reconstruction when projected onto the first i principal components.

$$\delta = |\Psi(x_N) - P_l\Psi(x_N)|^2 \tag{10}$$

In here P_l is the first i principal component. The MSE(Mean Square Error) value of reconstruction error in APEX is 4.4403 whereas OL-NPCA is 0.0223. This means that the accuracy of OL-NPCA is superior to standard APEX and similar to that of batch KPCA. Above results of simple toy problem indicate that OL-NPCA is comparable to the batch way KPCA and superior in terms of accuracy.

Next we will compare the memory efficiency of OL-NPCA compared to KPCA. To extract nonlinear features, OL-NPCA only needs D matrix and R matrix whereas KPCA needs kernel matrix. Table 2 shows the memory requirement of each method. Memory requirement of standard KPCA is 93 times more than OL-NPCA. We can see that OL-NPCA is more efficient in memory requirement than KPCA and has similar ability in extracting nonlinear features. By this simple toy problem we can show that OL-NPCA has similar ability in extracting nonlinear features compare to KPCA and more efficient in memory requirement than KPCA.

Table 2. Memory efficiency of OL-NPCA compared to KPCA on toy data

	KPCA	OL-NPCA
Kernel matrix	41 X 41	none
R matrix	none	3 X 3
D matrix	none	3 X 3
Efficiency ratio	93.3889	1

5.2 Reconstruction Ability

To compare the reconstruction ability of incremental eigenspace update method proposed by Hall to APEX model we conducted experiment on US National Institute of Standards and Technology(NIST) handwritten data set. Data has been size-normalized and 16 X 16 images with their values scaled to the interval [0,1]. Applying this data to incremental eigenspace update method we finally obtain 6 Eigenvectors. As earlier experiment we set 6 output nodes to standard APEX method. Figure 1 shows the original data and their reconstructed images by incremental eigenspace update method and APEX respectively. We can see that reconstructed features by incremental eigenspace update method is more clear and similar to original image compared to APEX method.

5.3 UCI Machine Learning Repository

To test the performance of proposed classifier for real world data, we enlarge our experiment to the Cleveland heart disease data and wine data obtained from the UCI Machine Learning Repository. Detailed description of data is available from web site(http://www.ics.uci.edu/ mlearn/MLSummary.html). In this problem we randomly split training data as 80% and remaining as test data. A RBF kernel has been taken with and obtained by 10-fold cross-validation procedure to select the optimal hyperparameter. Table 3 shows the learning and generalization ability by proposed classifier.

By this result we can see that proposed classification system classifies well on specific data.

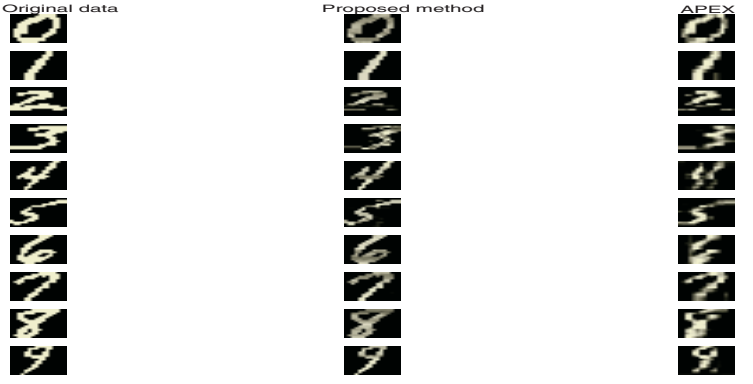

Fig. 1. Reconstructed image by OL-NPCA and APEX

Table 3. Training and generalization result by proposed classifier on UCI Machine Learning Repository

	Training	Generalization	Eigenvalue update criterion
Cleveland heart-disease	100%	97.35%	$\lambda^{'} > 0.7\overline{\lambda}$
Wine data	100%	98.04%	$\lambda^{'} > 0.7\overline{\lambda}$

5.4 NIST Handwritten Data Set

To validate the above results on a widely used pattern recognition benchmark database, we conducted classification experiment on the NIST data set. This database originally contains 15,025 digit images. For computational reasons, we decided to use a subset of 2000 data set, 1000 for training and 1000 for testing. In this problem we use multiclass LS-SVM classifier proposed by Suykens[16]. An important issue for SVM is model selection. In [17] it is shown that the use of 10-fold cross-validation for hyperparameter selection of LS-SVMs consistently leads to very good results. In this problem RBF kernel has been taken and hyperparameter $\gamma_1 = 1.5198,\ \gamma_2 = 179.731,\ \gamma_3 = 10.51,\ \gamma_4 = 12.81$ and $\sigma_1 = 67.416,\ \sigma_2 = 656.351,\ \sigma_3 = 54.349,\ \sigma_4 = 57.909$ are obtained by 10-fold cross-validation technique. The results on the NIST data are given in Table 4 and 5. For this widely used pattern recognition problem, we can see that proposed classification system classifies well on given data.

6 Conclusion and Remarks

This paper is devoted to the exposition of a new technique on extracting nonlinear features and classification system from the incremental data. To develop this technique, we apply an incremental eigenspace update method to KPCA with an polynomial feature mapping function approach. Proposed OL-NPCA has following advantages. Firstly, OL-NPCA has similar feature extracting performance

Table 4. Training and generalization result on NIST handwritten data

	Training	Generalization	Eigenvalue update criterion
Proposed Classifier	100%	98.7%	$\lambda^{'} > 0.7\overline{\lambda}$

Table 5. Misclassification frequency by proposed classification system on test data

Pattern	0	1	2	3	4	5	6	7	8	9	Total
Frequency	0	0	0	3	4	0	0	6	0	0	13

for incremental and nonlinear data comparable to batch KPCA. Secondly, OL-NPCA is more efficient in memory requirement than batch KPCA. In batch KPCA the $N \times N$ kernel matrix has to be stored, while for OL-NPCA requirements are $O((k+1)^2)$. Here $k(1 \leq k \leq N)$ is the number of eigenvectors stored in each eigenspace updating step, which usually takes a number much smaller than N. Thirdly, OL-NPCA allows for complete incremental learning using the eigenspace approach, whereas batch KPCA recomputes whole decomposition for updating the subspace of eigenvectors with another data. Finally, experimental results show that extracted features from OL-NPCA lead to good performance when used as a pre-preprocess data for a LS-SVM.

References

1. Tipping, M.E. and Bishop, C.M. :Mixtures of probabilistic principal component analysers. Neural Computation 11(2), (1998) 443-482
2. Kramer, M.A.:Nonlinear principal component analysis using autoassociative neural networks. AICHE Journal 37(2),(1991) 233-243
3. Diamantaras, K.I. and Kung, S.Y.:Principal Component Neural Networks: Theory and Applications. New York John Wiley & Sons, Inc.(1996)
4. Kim, Byung Joo. Shim, Joo Yong. Hwang, Chang Ha. Kim, Il Kon, "Incremental Feature Extraction Based on Emperical Feature Map," Foundations of Intelligent Systems, volume 2871 of Lecture Notes in Artificial Intelligence, pp 440-444, 2003
5. Softky, W.S and Kammen, D.M, "Correlation in high dimensional or asymmetric data set: Hebbian neuronal processing," Neural Networks vol. 4, pp.337-348, Nov. 1991.
6. Gupta, H., Agrawal, A.K., Pruthi, T., Shekhar, C., and Chellappa., R., "An Experimental Evaluation of Linear and Kernel-Based Methods for Face Recognition," accessible at http://citeseer.nj.nec.com.
7. Suykens, J.A.K. and Vandewalle, J.:Least squares support vector machine classifiers. Neural Processing Letters, vol.9, (1999) 293-300
8. Vapnik, V. N.:Statistical learning theory. John Wiley & Sons, New York (1998)
9. Hall, P. Marshall, D. and Martin, R.: Incremental eigenalysis for classification. In British Machine Vision Conference, volume 1, September (1998)286-295
10. Winkeler, J. Manjunath, B.S. and Chandrasekaran, S.:Subset selection for active object recognition. In CVPR, volume 2, IEEE Computer Society Press, June (1999) 511-516

11. Murakami, H. Kumar.,B.V.K.V.:Efficient calculation of primary images from a set of images. IEEE PAMI, 4(5), (1982) 511-515
12. Scholkopf, B. Smola, A. and Muller, K.R.:Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10(5), (1998) 1299-1319
13. Tsuda, K., "Support vector classifier based on asymmetric kernel function," Proc. ESANN, 1999.
14. Mika, S.:Kernel algorithms for nonlinear signal processing in feature spaces. Master's thesis, Technical University of Berlin, November (1998)
15. Diamantaras, K.I. and Kung, S.Y, Principal Component Neural Networks: Theory and Applications, New York John Wiley&Sons, Inc. 1996.
16. Suykens, J.A.K. and Vandewalle, J.: Multiclass Least Squares Support Vector Machines, In: Proc. International Joint Conference on Neural Networks (IJCNN'99), Washington DC (1999)
17. Gestel, V. Suykens, T. J.A.K. Lanckriet, G. Lambrechts, De Moor, A. B. and Vandewalle, J., "A Bayesian Framework for Least Squares Support Vector Machine Classifiers," Internal Report 00-65, ESAT-SISTA, K.U. Leuven.