International Journal of Security and Its Applications
Vol.8, No.5 (2014), pp.243-252
http://dx.doi.org/10.14257/ijsia.2014.8.5.22

Incremental Eigenspace Model Applied to Monitoring System[1]

Byung Joo Kim

Department of Computer Engineering Youngsan University, Korea
bjkim@ysu.ac.kr

Abstract

This paper describes a real time feature extraction for real-time surveillance system. We use incremental KPCA method in order to represent images in a low-dimensional subspace for real-time surveillance in the traditional approach to calculate these eigen space models, known as batch PCA method, model must capture all the images needed to build the internal representation. Updating of the existing eigen space is only possible when all the images must be kept in order to update the eigen space, requiring a lot of storage capability. Proposed method allows discarding the acquired images immediately after the update. By experimental results we can show that incremental KPCA has similar accuracy compare to KPCA and more efficient in memory requirement than KPCA. This makes pro-posed model is suitable for real time surveillance system. We will extend our research to real time face recognition based on this research.

Keywords: *Incremental KPCA, Eigen space, Monitoring System*

1. Introduction

In recent years, there has been an increase in video surveillance systems in public and private environments due to a heightened sense of security. The next generation of surveillance systems will be able to annotate video and locally coordinate the tracking of objects while multiplexing hundreds of video streams in real-time. Unsupervised surveillance gadgets aided by hi-tech visual information retrieval and indexing systems use computerized face recognition techniques that can recognizes faces from an image. There are two main approaches for face recognition [1]. The first approach is the feature based matching approach using the relationship between facial features [2]. The second approach is the template matching approach using the holistic features of the face images [2]. Template based techniques often follow the subspace method called eigen face originated by Turkand Pentland [3].

This technique is based on the Karhunen-Loeve transformation, which is also referred as PCA. It has gained great success and become a de facto standard and a common performance benchmark in face recognition. One of the attractive characteristics of PCA is that a high dimension in-formation t vector can be represented by a small number of orthogonal basis vectors. The conventional methods of PCA such as singular value decomposition (SVD) and eigen-decomposition perform in batch-mode with a computational complexity of O (m^3) when is the minimum value between the data dimension and the number of training examples. Undoubtedly these methods are computationally expensive when dealing with large scale problems where both the dimension and the number of training examples are large. To address this problem, many researchers have been working on incremental algorithms. Among them Chandrasekaran, *et al.*, presented an incremental eigen space update method using SVD[4]. Hall, *et al.*, derived aneigen-decomposition based incremental algorithm and later extended their workto merge and split eigenspace models[5]. Another problem of PCA is that it 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 hat could be useful [6]. 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 [7]. 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 for when N is large. This fact has motivated the development of incremental way of KPCA method which does not store the kernel matrix. In this paper we propose a method that allows for incremental eigen space update method by incremental kernel PCA for vision learning and recognition. Paper is organized as follows. In Section 2 we will briefly explain the incremental PCA method. In Section 3 KPCA is introduced and to make KPCA incrementally, empirical kernel map method is explained. An experimental result to evaluate the performance of proposed method is shown in Section 4. Discussion of proposed method and future work is described in Section 5.

2. Incremental PCA

In this section, we will give a brief introduction to the method of incremental PCA algorithm which overcomes the computational complexity 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 $[A_{mn} b]$ isan(m × (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 images $x_i, i =, \cdots, N$. The corresponding eigen values are Λ and $\overline{\mathrm{X}}$ is the mean of input image. Incremental building of eigen space requires to update these eigen space to take into account of a new input image. Here we give a brief summarization of the method which is described in [5]. First, we update the mean:

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

We then update the set of eigenvectors to reflect the new input image and to apply a rotational transformation to U. For doing this, it is necessary to compute the orthogonal residual vector $\hat{h} = (Ua_{N+1} + \bar{x}) - x_{N+1}$ where

a_{N+1} is principal component and normalize it to obtain $h_{N+1} = \frac{h_{N+1}}{\|h_{N+1}\|_2}$ for $\|h_{N+1}\|_2 \rangle 0$ and $h_{N+1} = 0$ otherwise. We obtain the new matrix of eigenvectors $U^{'}$ by appending h_{N+1} to t he eigenvectors U and rotating them:

$$U^{'} = [U, h_{N+1}]R \quad (2)$$

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

$$DR = R\Lambda^{'} \quad (3)$$

where $\Lambda^{'}$ is a diagonal matrix of new eigen values. We compose D $\in \Re_{(k+1)\times(k+1)}$ as:

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

where $\gamma = h_{N+1}^T(x_{N+1} - \overline{x})$ and $a = U^T(x_{N+1} - \overline{x})$. Though there are other ways to constru ct matrix D [4][5], the only method ,however, described in [6] allows for the updatin g of mean.

2.1. Updating Image Representations

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

$$x_{i(N)} = \hat{U}a_{i(N)} + \overline{x} \quad (5)$$

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

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

then the computation of all principal components is

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

The transformations described above yield a model that represents the input images with the same accuracy as the previous one; therefore we can now discard the old subspace and the coefficients that represent the image in it. x_{N+1} Is represented accurately as well, so we can safely discard it. The representation of all N + 1 image are possible because the subspace is spanned by k +1eigenvector. Due to the increase of the dimensionality by one, however, more storage is required to represent t he data. If we try to keep ak-dimensional eigen space, we lose a certain amount of information. In order to balance the storage requirements with the level of accuracy, 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.

In this paper we set our criterion on adding an eigenvector as $\lambda'_{K+1} \rangle 0.7\bar{\lambda}$ where $\bar{\lambda}$ is a mean of the λ. Based on this rule, we decide whether adding u'_{K+1} or not.

3. Incremental KPCA

A prerequisite of the incremental eigen space 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 most of the time cannot even be calculated explicit ly. 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 im-plicit embedding to an infinite dimensional Hilbert space [9] (*i.e.*, feature space) F.

$$K(x, y) = \Phi(x) \cdot \Phi(y) \quad (8)$$

Where K is a given kernel function in an input space. When K is semi positive definite, the existence of Φ is proven [7]. Most of the case, however, the mapping Φ is high-dimensional and cannot be obtained explicitly. The vector in the feature space is not observable and only the inner product between vectors cans be observed via a kernel function. However, for a given data set, it is possible to approximate Φ by empirical kernel map proposed by Scholkopf [10] and Tsuda [11]which is defined as $\Psi_N : \Re^d \rightarrow \Re^N$

$$\Psi_N(x) = [\Phi(x_1) \cdot \Phi(x), \cdots, \Phi(x_N) \cdot \Phi(x)]^T$$
$$= [K(x_1, x), \cdots, K(x_N, x)]^T \quad (9)$$

A performance evaluation of empirical kernel map was shown by Tsuda. He shows that support vector machine with an empirical kernel map is identical with the conventional kernel map [12]. The empirical kernel map $\Psi_N(x_N)$,however, do not form an orthonormal basis in $\Re^N$, the dot product in this space is not the ordinary dot product. In the case of KPCA, however, we can be ignored as the following argument. The idea is that we have to perform linear PCA on the $\Psi_N(x_N)$ from the empirical kernel map and thus diagonalize its covariance matrix. Let the N × N matrix $\Psi = [\Psi_N(x_1), \Psi_N(x_2), \cdots, \Psi_N(x_N)]$, then from equation (9) and definition of the kernel matrix we can construct Ψ = NK. The covariance matrix of the empirically mapped data is:

$$C_\Psi = \frac{1}{N}\Psi\Psi^T = NKK^T = NK^2 \quad (10)$$

In case of empirical kernel map, we diagonalize NK^2 instead of K as in KPCA. Mika shows that the two matrices have the same eigenvectors $\{u_k\}$ [12]. The eigen values $\{\lambda_k\}$ of K are related to the eigen values $\{K_k\}$ of NK^2 by

$$\lambda_k = \sqrt{\frac{K_k}{N}} \quad (11)$$

and as before we can normalize the eigenvectors $\{v_k\}$ for the covariance matrix C_Ψ of the data by dividing each $\{u_k\}$ by $\sqrt{\lambda_k N}$. Instead of actually diagonalize the covariance matrix C_Ψ, the incremental KPCA is applied directly on the mapped data Ψ = NK. This makes it easy for us to adapt the incremental eigen space update method to KPCA such that it is also correctly takes into account the centering of the mapped data in an incremental way. By this result, we only need to apply the empirical map to one data point at a time and do not Need to store the N × N kernel matrix.

4. Experiment

To evaluate the performance of accuracy on eiegn space update for incremental data we take nonlinear data. 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 which will show the accuracy and memory efficiency of incremental KPCA compared to APEX model proposed by Kung [13] and batch KPCA. Next we will use images from the Columbia Object Image Library (COIL-20). The set is consisted of images of 20 objects rotated about their vertical axis, resulting in 72 images per objects. We used these images for testing the performance of incremental KPCA.

4.1. Toy Data

To evaluate the eigen space update accuracy and memory efficiency of incremental KPCA compared to APEX and KPCA we take nonlinear data used by Scholkoff [8]. Totally 41 training data set is generated by:

$$y = x^2 + 0.2\varepsilon : \varepsilon \;\; from \;\; N(0,1), x = [-1,1] \quad (12$$

Table 1. Performance Evaluation of Incremental KPCA(IKPCA) and APEX

Method	Iteration	Learning Rate	$\|\omega_1\|$	$\|\omega_2\|$	$\cos\theta_1$	$\cos\theta_2$	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
IKPCA	100		1	1	1	1	0.0223

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

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. $\|\omega\|$ is norm of weight vector in APEX and $\|\omega\|$=1 means that it converges stable minimum cosθ is angle between eigenvector of KPCA and APEX, incremental.

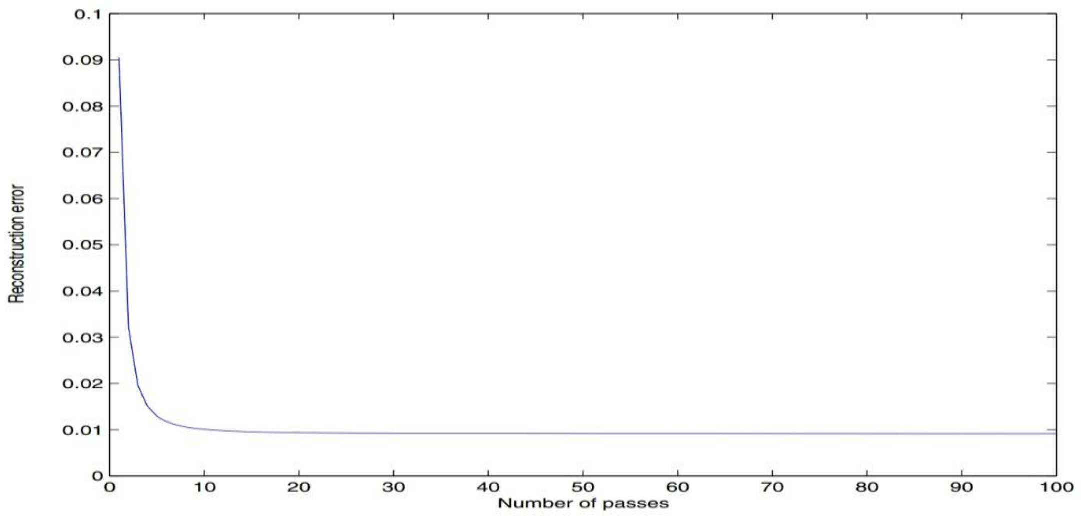

Figure 1. Reconstruction Error Change by Re-learning in Incremental KPCA

KPCA respectively. cosθ of eigenvector can be a factor of evaluating accuracy how much incremental KPCA and APEX is close to accuracy of KPCA. Table 1nicely shows the two advantages of incremental KPCA compared to APEX: first, performance of incremental KPCA is better than APEX; second, the performance of incremental KPCA is easily improved by re-learning. Another factor of evaluating accuracy is reconstruction error. Reconstruction error is defined as the squared distance between the Ψ image of x_N and reconstruction when projected onto the first principal components.

$$\delta = \left|\Psi(x_N) - P_l\Psi(x_N)\right|^2 \quad (13)$$

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 incremental KPCA is 0.0223. This means that the accuracy of incremental KPCA is superior to standard APEX and similar to that of batch KPCA. Figure 1 shows the MSE value change for reconstruction error by re-learning in incremental KPCA. Re-learning is similar meaning of epoch in neural network learning. We can see that the performance of incremental KPCA is easily improved by relearning. Above results of simple toy problem indicate that incremental KPCA is comparable to the batch way KPCA and superior in terms of accuracy.

Next we will compare the memory efficiency of incremental KPCA compared to KPCA. In these experiments, incremental KPCA only needs D matrix and R matrix whereas KPCA needs kernel matrix.

Table 2. Memory Efficiency of Incremental KPCA Compared to KPCA on Toy Data

	KPCA	IKPCA
Kernel matrix	14 X 41	None
R matrix	None	3 X 3
D matrix	None	3 X 3
Efficiency ratio	93.3889	1

Table 2 shows the memory requirement of each method. Memory requirement of standard KPCA is 93 times more than in-cremental KPCA. We can see that incremental KPCA is more efficient in memory requirement than KPCA and has similar ability of eigen space update accuracy. By this simple toy problem we can show that incremental KPCA has similar accuracy compare to KPCA and more efficient i n memory requirement than KPCA.

4.2. Reconstruction Ability

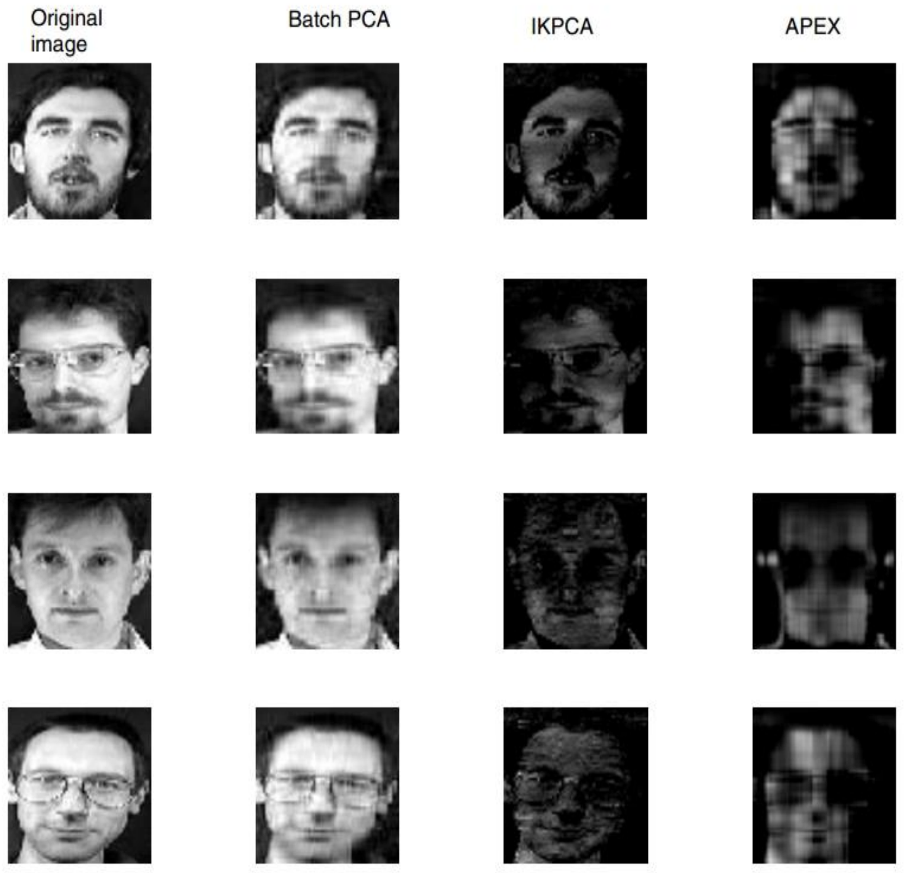

Figure 2. Shows the Original Data and their Reconstructed Images by Incremental KPCA Method, Batch PCA and APEX Respectively

To compare the reconstruction ability of incremental eigen space update method proposed by Hall to APEX model we conducted experiment on face data. Applying this data to incremental eigen space update method we finally obtain30 Eigenvectors. As earlier experiment we set 30 output nodes to standard method.

The MSE (Mean Square Error) value of reconstruction error in APEX is 10.159 whereas incremental KPCA is 0.26941 and KPCA is 0.15762. This means that the accuracy of incremental KPCA is superior to standard APEX and similar to that of batch KPCA. We can see that reconstructed images by incremental KPCA update method is similar to original image and more clearly compared to APEX method.

5. Conclusion and Remarks

This paper was devoted to the real time feature extraction for real-time surveillance system. To develop this technique, we made use of empirical kernel mapping with incremental learning by eigen space approach. Proposed IKPCA has following advantages.

Firstly, IKPCA has similar feature extracting performance for incremental and nonlinear data comparable to batch KPCA.

Secondly, IKPCA is more efficient in memory requirement than batch KPCA. In batch KPCA the $\mathrm{N} \ \mathrm{X} \ \mathrm{N}$ kernel matrix has to be stored, while for IKPCA requirements are$O((k+1))^2$. Here $(1 \leq k \leq N)$ is the number of Eigenvectors stored in each eigenspace updating step, which usually takes a number much smaller than$N$.

Thirdly, IKPCA allows for complete incremental learning using the eigen space approach, whereas batch KPCA re computes whole decomposition for update the subspace of Eigenvectors with another data.

Fourthly, IKPCA can easily be improved by re-learning the data.

Fifthly IKPCA do not require nonlinear optimization techniques.

Lastly, experimental results show that extracted features from IKPCA lead to good performance when used as preprocess data for a linear classifier.

Future works is developing proper classifier with proposed method so as to perform roboust intelligent monitoring system.

References

[1] R. Chellappa, C. L. Wilson and S. Sirohey, “Human and machine recognition off aces”, a survey Proceedings of IEEE, **(1995)** May.
[2] R. Brunelli and T. Poggio, “Face recognition: feature versus templates”, IEEE Trans. PAMI, vol. 15, no. 10, **(1993).**
[3] M. Turkand and A. Pentland, “Face recognition using eigen faces”, Proceedings of CVPR, **(1991).**
[4] J. Winkeler, B. S. Manjunath and S. Chandrasekaran, “Subset selection for active ob-ject recognition”, In CVPR, vol. 2, **(1999).**
[5] P. Hall, D. Marshall and R. Martin, “On-line eigenalysis for classification”, In British Machine Vision Conference, **(1998)** September.
[6] W. S. Softky and D. M. Kammen, “Correlation in high dimensional or asymmetric data set”, Hebbian neuronal processing, Neural Networks, vol. 4, **(1991)** November.
[7] H. Gupta, A. K. Agrawal, T. Pruthi, C. Shekhar and R. Chellappa, “An Experimental Evaluation of Linear and Kernel-Based Methods for Face Recognition”, accessible at http://citeseer.nj.nec.com.
[8] H. Murakami and B. V. K. V. Kumar, “Efficient calculation of primary images from a set of images”, IEEE PAM, vol. 4, no. 5, **(1982).**
[9] V. N. Vapnik, “Statistical learning theory”, John Wiley & Sons, New York, **(1998).**
[10] B. Scholkopf, A. Smola and K. R. Muller, Nonlinear component analysis as a kernel eigen value problem, Neural Computation, vol. 10, no. 5, **(1998).**
[11] K. Tsuda, “Support vector classifier based on asymmetric kernel function”, Proceedings of ESANN, **(1999).**

[12] S. Mika, "Kernel algorithms for nonlinear signal processing in feature spaces", Master thesis, Technical University of Berlin, **(1998)** November.
[13] K. I. Diamantaras and S. Y. Kung, "Principal Component Neural Networks: Theory and Applications", New York John Wiley & Sons, Inc., **(1996).**

Author

ByungJoo Kim, PhD in Computer Science, Kyungpook National University. Master and BSc in computer and statistics from Pusan National University. His research area is machine learning, Brain scienc. Now he is professor department of computer engineering at Youngsan University.