An Improvement Algorithm of Principal Component Analysis

Yu Chuanqiang Guo Xiaosong Zhang An Pan Xingjie

(Xi'an Research Inst. of Hi-Tech Hongqing Town, Xian 710025, China)

Abstract: The conventional method of principal component analysis (PCA) is reducing data dimensions directly from m to k (k<m) by one step. The lost information of PCA is holistically determined by the k. To reduce the lost information in the case of k is determined, we decrease the dimensions of the data from m to k by n(1≤n≤(m-k))steps. This new PCA method is called multi-step PCA (MPCA). The algorithm of MPCA is shown in the article. Two linear Neural Networks based on the PCA or MPCA is analyzed. Compared the PCA with MPCA and compared the numeric algorithm with Neural Networks, we find that the correct classification capability of MPCA is some better than the PCA and the correct classification capability o f Neural Networks is some better than the numeric algorithm.

Keywords: Principal Component Analysis (PCA) Multi-step PCA Neural Networks Pattern Recognition

1 Introduction

Principal Component Analysis (PCA) has been widely used in feature extraction from complex and high dimensional data in many fields, such as signal processing, image processing and pattern recognition. It reduces the dimensionality of the feature space by creating new features that are linear combinations of the original features [1].

2 Principal Component Analysis (PCA) Algorithm

PCA s a linear dimensionality reduction technology. The features extracted from the sensor signals, such as vibration, temperature, current or voltage, are firstly analyzed by the PCA method. Suppose that n observations for m features form an $n \times m$ matrix, signed as $X = (x_{ij}), i = 1,2,\cdots n;$ $j = 1,2,\cdots m$. The PCA is processed as follows [2]:

(1) Standardize the data.

$$x_{ij} = \frac{x_{ij} - x_{j,mean}}{\sigma(x_j)} \qquad (1)$$

Where $x_{j,mean}$ and $\sigma(x_j)$ are the mean and the standard deviation of the j-th feature, respectively.

(2) Calculate $m \times m$ correlation matrix *C*, which is symmetrical and positive definite.

$$C = X^T X \qquad (2)$$

(3) The eigenvalue λ_j and the eigenvector p_j of *C* are computed in decreasing order of magnitude ($\lambda_1 > \lambda_2 > \ldots\ldots > \lambda_m$). The original data then can be expressed in terms of the eigenvalues and eigenvectors, which define the principal component directions:

$$X = t_1 p_1^T + t_2 p_2^T + \ldots + t_k p_k^T + E \qquad (3)$$

where $E = t_{k+1} p_{k+1}^T + t_{k+2} p_{k+2}^T + \ldots + t_m p_m^T$.

Eq.(3) can be rewritten in the matrix form:

$$X = TP^T + E \qquad (4)$$

where $T = [t_1, t_2, \ldots t_k]$ with k<m is called the

1-4244-1135-1/07/$25.00 ©2007 IEEE.

principal component scores, P = [p_1 , p_2 ,… p_k] is called the principal component loadings. E is the residual, and $k<m$ is the number of the scores. Then condition of the optimizing Eq.(4) is that the Euclidean norm of the residual matrix E must be minimized. To satisfy this criterion, it has been demonstrated that P = [p_1 , p_2 ,… p_k] is the eigenvector of the covariance matrix of X.

An approximate model, comprising of the first k terms of Eq.(3) , will capture most of the observed variance in X if the data are correlated. The percentage that the information in X can be expressed as the first k terms of principals is

$$Q = \frac{\lambda_1 + \lambda_{2+} ... + \lambda_k}{\sum_{j=1}^{m} \lambda_j} \qquad (5)$$

From expression (3) and (4) the scores can be obtained:

$$T = XP \qquad (6)$$

It can be seen that PCA is a linear mapping of the original observed data. The load vector P is the coefficients for linear transformation. After getting P and T, the reconstructed data can be written as follows:

$$X^{'} = TP^{T} + E \qquad (7)$$

where $X^{'}$=[$x_1, x_2,...x_k$]^T is the reconstruction of observed data and the dimensions of the data is reduced form m to k ($k<m$). The principle of PCA is shown in Fig. 1. The method of determining the number (k) of the principal components is the key in application of the PCA. There are several methods to do that, but no one is dominion.

3 Multi-Step Principal Component Analysis (MPCA) Algorithm

To reduce the dimensions of the data from m to k, the conventional PCA method is one step. The percentage that the lost information in X is expressed as:

$$Q_e = \frac{\lambda_{k+1} + \lambda_{k+2+} ... + \lambda_m}{\sum_{j=1}^{m} \lambda_j} = 1 - Q \qquad (8)$$

From the Eq. (8) we can think that if we reduce the dimensions of the data from m to k by n(1≤n≤(m-k))steps, the percentage of the residual or the lost information may be lower than the one step. This is the main idea of the multi-step PCA(MPCA). The principle of MPCA is shown as Fig. 2 We can give the algorithm of the MPCA as follow:

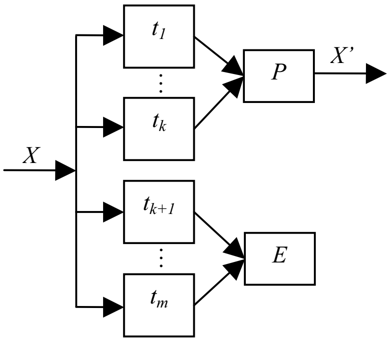

Fig. 0. The principle of PCA

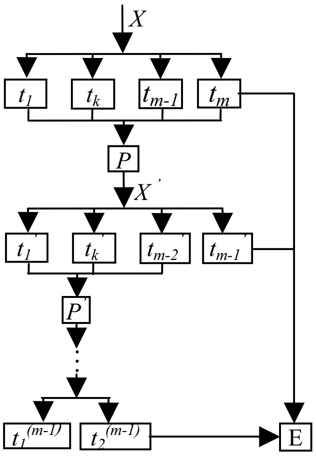

Fig. 0. The principle of MPCA

(1) Standardize the data (reference to Eq. (1)).

(2) Calculate $m \times m$ correlation matrix C, which is symmetrical and positive definite (reference to Eq.(2)).

(3) Calculate the eigenvector p,j=1,2…m) of matrix C and set P = [p_1,p_2…p_{m-1}]

(4) Calculate $X’=XP$ and dimensions of $X’$. If

dimensions of *X*’ is what we intend， end the process and the result is *X*’. Otherwise go to step (1).

Notice:

(1) If we reduce the dimensions of the data from *m* to *k* by (*m-k*) steps, whether the lost information will be the lowest or the classification capability is more powerful. Though the mathematical demonstration is not obtained, the following examples are shown that the case is thus.

(2)The step that the dimensions of the data reduced from the *m* to *k* is determined by the application. The number of the steps is more, the consumed time is more and the result is more satisfied. So we should select the number of the steps by the application.

4 Neural Networks of PCA and MPCA

There are close relationships between the act of the self-organized Neural Networks and the method of the PCA. Oja has demonstrated that a single neuron that its weight updates by the self-adaptive Hebbian learning rule can construct a filter of the first principal component from the input data [3]. And a linear Neural Networks with *k* neurons can obtain *k* principal components [4]. The architecture of this linear Neural Networks is shown as Fig. 3

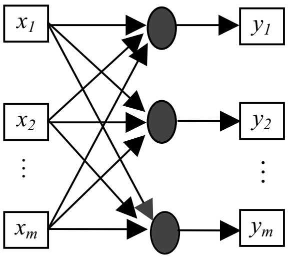

Fig.3 The architecture of Neural Networks that can obtain *k* principal components

From the principle of MPCA we can also construct the multilayer linear Neural Networks to obtain *k* principal components by *n*(1≤*n*≤(*m-k*)) steps. The number of the layer is *n.* The architecture of this multilayer linear Neural Networks is shown as Fig. 4

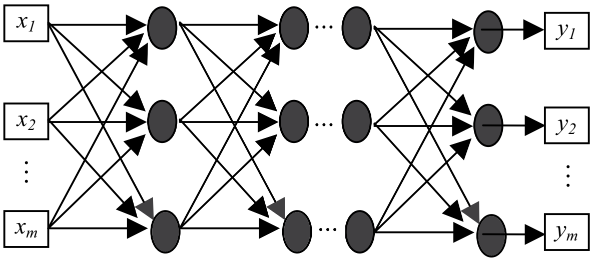

Fig. 4 The architecture of multilayer linear neural networks that can obtain principal components

5 Verifying MPCA

To verify MPCA, we use the typical Fisher data [5]. There are 3 types in the data which is shown as Tab 1, and one type includes *m*=4 features, *n*=50 observations. The figure of the original data is shown as Fig. 5

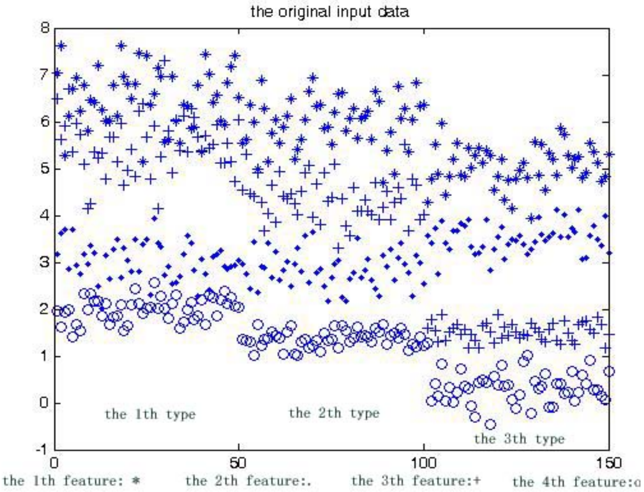

Fig. 5 The original input data

5.1 Numeric Algorithm of PCA and MPCA

Realization of PCA.

According to the algorithm of PCA, we calculate eigenvalue λ and the eigenvector V by the 3th type:

$$V=\begin{pmatrix} -0.9291 & 0.1025 & -0.2322 & 0.2691 \\ -0.3448 & -0.0866 & 0.3269 & -0.8756 \\ -0.0621 & -0.9904 & -0.0827 & 0.0916 \\ -0.1185 & -0.0326 & 0.9124 & 0.3905 \end{pmatrix} \quad (9)$$

$$\lambda = \begin{bmatrix} 1.6244 & 0 & 0 & 0 \\ 0 & 1.3391 & 0 & 0 \\ 0 & 0 & 0.7820 & 0 \\ 0 & 0 & 0 & 0.5536 \end{bmatrix} \qquad (10)$$

We chose $P = [\, v_1 , v_2 \,]$ as the principal component loading and calculate the output (T) of the 1th to the 3th type data according Eq.(6). The data after PCA is shown as Fig. 6

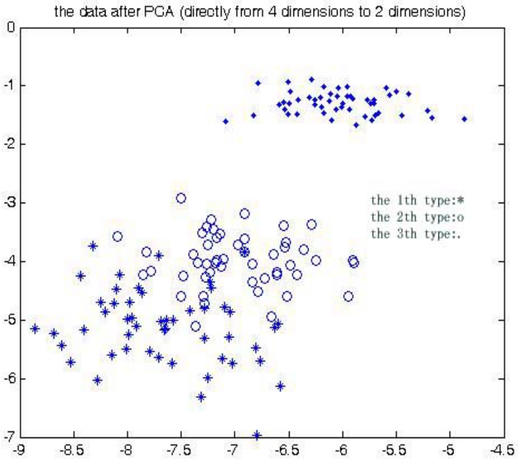

Fig.6 The data after PCA
(directly from 4 dimensions to 2 dimensions)

Table 1. The statistical features of the Fisher data

The 1th type	mean	standard deviation	scope
The 1th feature	6.59	0.64	4.9-7.9
The 2th feature	2.98	0.32	2.2-3.8
The 3th feature	5.55	0.55	4.5-6.9
The 4th feature	2.03	0.27	1.4-2.5
The 2th type	mean	standard deviation	scope
The 1th feature	5.94	0.52	4.9-7.0
The 2th feature	2.77	0.31	2.0-3.4
The 3th feature	4.29	0.47	3.0-5.1
The 4th feature	1.33	0.20	1.0-1.8
The 3th type	mean	standard deviation	scope
The 1th feature	5.01	0.35	4.3-5.8
The 2th feature	3.43	0.38	2.3-4.4
The 3th feature	1.46	0.17	1.0-1.9
The 4th feature	0.30	0.40	0.1-3.0

Realization of MPCA

According to the algorithm of MPCA, we calculate the first step’s eigenvalue $\lambda^{'}$ and eigenvector $V^{'}$ by the 3th type. They are the same as PCA (reference to Eq.(9) and Eq.(10). We chose $P = [\, v_1 , v_2 , v_3 \,]$ as the first principal component loading and calculate the output ($X^{'}$) of the 3th type data according Eq.(6). Then we calculate eigenvalue $\lambda^{''}$ and the eigenvector $V^{''}$ by the $X^{'}$ after $X^{'}$ is processed according to Eq.(1). At the end the principal component loading $P^{''}$is:

$$P^{''}= \begin{bmatrix} 0.5958 & -0.5472 \\ -0.7360 & -0.0789 \\ 0.3215 & 0.8333 \end{bmatrix} \qquad (11)$$

and the output (T) of the 1th to the 3th type data is shown as Fig. 7after 4-3-2 MPCA.

Conclusions.

Compared Fig.6 with Fig.7, we can find:

(1) The boundary between the 1th type data and the 2th type data processed by the MPCA method is more clear than the PCA method.

(2) The relative position of the 1th type data and the 3th type data is reverse.

(3) The boundary between the 3th type data and the 2th type data isn’t clearer by the MPCA method than the PCA method. But it can distinguish those two type data.

(4) The holistically effect of correctly classing the data by the MPCA method is more efficient than the PCA method.

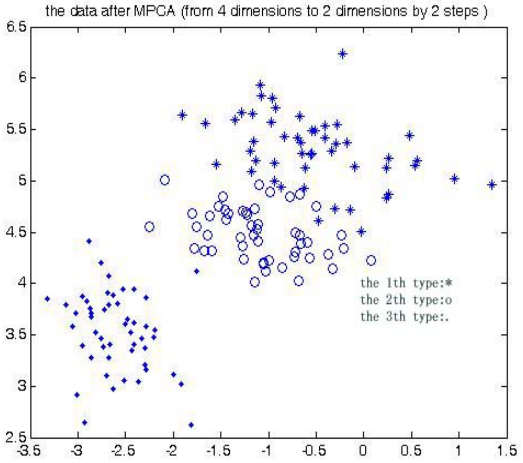

Fig. 7 The data after MPCA
(from 4 dimensions to 2 dimensions by 2 steps)

5.2 Neural Networks of PCA and MPCA

PCA Neural Networks.

Now we apply the idea of PCA to construct a linear single layer Neural Networks. The architecture of this linear Neural Networks is shown as Fig 3. where the number of input elements is 4, the number of the neurons is 2, the learning rule is generalized Hebbian algorithm and the weight initialization function is Matlab function: rands. Then we take the 3th type data as the sample to train the net. After 20000 epochs, the output of the networks is shown as Fig. 8. In the process of training, we can discover:

(1) The epochs of training heavily depend on the initial input weight matrix.

(2) Though the difference of the terminal weight matrix is obvious, the correct classification capability of the net is near.

(3) The correct classification capability of the net is some better than PCA realized by numeric algorithm.

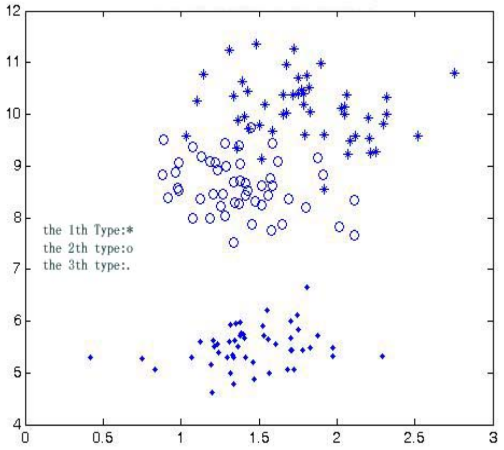

Fig. 8. The data after PCA Neural Networks (directly from 4 dimensions to 2 dimensions)

MPCA Neural Networks

Now we apply the idea of MPCA to construct a linear two layers Neural Networks. The architecture of this linear Neural Networks is shown as Fig 3. Where the number of input elements is 4, the number of the neurons in the 1th layer is 3 and in the 2th layer is 2, the two layers learning rule are all generalized Hebbian algorithm and the weight initialization function is Matlab function: rands. Then we take the 3th type data as the sample to train the net. After 13000 epochs, the output of the networks is shown as Fig. 9. In the process of training, we can discover:

(1) The epochs of training heavily depend on the initial weight matrix.

(2) Though the difference of the terminal weight matrix is obvious, the correct classification capability of the net is near.

(3) The correct classification capability of the net is some better than the MPCA realized by numeric algorithm.

(4) The correct classification capability of this net is some better than the single layer net (PCA).

(5) With the increasing of the training epochs, every type data is more clustering .

(6) The training epochs of MPCA net are less than the PCA net.

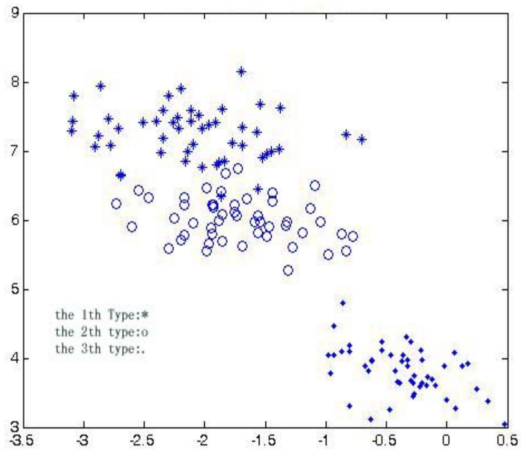

Fig. 9. The data after MPCA Neural Networks (from 4 dimensions to 2 dimensions by 2 steps)

6 Conclusions

In the process of application we find that the holistically correct classification capability of MPCA is better than the PCA. This capability is obtained by averaging the distance between all types of data. The clear boundary is some vaguer and the vague boundary is some clearer. So we can combine MPCA and PCA in some application to obtain better effect. In addition the mathematical demonstration of MPCA is not obtained. We can demonstrate it from information theory or matrix decomposition.

References

[1]Manabu K, Shinij H,Lori H, et. A New Multivariate Statistical Process Monitoring Method Using Pprincipal Ccomponent Aanalysis. Computers and Chemical Engineering. Vol. 25:103-1113, 2001.

[2]Lewin D R. Predictive Maintenance Using PCA . Control Engineering Practice. Vol. 3: 431-421, 1995.

[3]Oja.E. Principal Component Analysis, Minor Components, and Llinear Neural Networks. Neural Networks. Vol. 5:27-936, 1992.

[4]Sanger T.D. An Optimality Principle for Unsupervised Llearning. Advances in Neural Information Processing Systems. Vol. 1: 11-19, 1989.

[5]M. Chow. Methodologies of Using Neural Network and fuzzy logic technologies for motor incipient fault detection. World Scientific. Singapore,1997.

Author Biographies:

Yu Chuanqiang(1975): male, born in QiXia, ShanDong province. Doctor graduate student, interests: fault diagnosis of electric control systen,the apply of automatic checkout in the weapon system.