UDC 004.032.26

ADAPTIVE MULTISTEP SELF-LEARNING PROCEDURE FOR SOLVING PRINCIPAL COMPONENT ANALYSIS TASK

Ye. Bodyanskiy , A. Deineko , M. Shalamov
Kharkiv National University of Radio Electronics

Abstract. *In many tasks associated with the large data sets processing, often arises a problem of compression with minimal loss of information in order to select the most essential features that define the nature of the phenomenon under investigation, data visualization, their transmitting over channels with limited bandwidth, etc. For solving such tasks principal component analysis (PCA) is widely used. This task is, also known as algebraic eigenvalue problem, or Karhunen-Loeve transformation. In situation when data sequentially are fed to processing in on-line mode, their volume is not known beforehand, and the system that generates the data is non-stationary, the traditional algorithms that implement the method of principal components loses its effectiveness and adaptive procedures based on neural network technology have to be used. In this regards, multistep self-learning rule for adaptive linear associator designed for finding eigenvalues and eigenvectors of the correlation matrix data that sequentially are fed to processing in on-line mode had proposed. This rule is a generalization of D. Hebb and E. Oja algorithms, used for neural networks training, implementing the method of principal components.*

Key words: *Data Mining, Text Mining, Web Mining, principal component analysis (PCA), principal components space, data compression, eigenvector.*

Introduction

In many tasks associated with Data Mining, Text Mining, Web Mining the problem of large arrays compression arises. For solving such tasks principal component analysis (PCA) is widely used, consisting in the orthogonal projection of data vectors $\{x(1), x(2), ..., x(k), ..., x(N)\} \in R^n$ from the original $n-$dimensional space to the reduced $m-$dimensional space $(m<n)$, called principal components space. Mathematically this method reduces to the search of system w_1, $w_2, \ldots, w_m$ of $n-$dimensional orthogonal eigenvectors of the correlation matrix that are previously centered relatively to the mean of initial data, wherein the vector $w_1 = (w_{11}, w_{12}, ..., w_{1n})^T$ corresponds to the first eigenvalue λ_1 of the correlation matrix, $w_2 -$second largest (dominant) eigenvalue etc. From a formal viewpoint, this problem reduces to finding solutions of the matrix equation

$$\left(R(N) - \lambda_j I_{nn}\right) w_j = \vec{0} \qquad (1)$$

such as $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_m \geq 0$ where

$$\|w_j\| = 1.$$

Here

$$R(N) = \frac{1}{N}\sum_{k=1}^{N}\left(x(k) - \bar{x}(N)\right)\left(x(k) - \bar{x}(N)\right)^T = \frac{1}{N}\sum_{k=1}^{N}\tilde{x}(k)\tilde{x}^T(k) \qquad (2)$$

$(n \times n)-$ data correlation matrix,

$$\bar{x}(N) = \frac{1}{N}\sum_{k=1}^{N} x(k). \qquad (3)$$

This task is, also known as algebraic eigenvalue problem, or Karhunen-Loeve transformation, well studied and solved assuming that input dataset of observations $x(1)$, $x(2), \ldots, x(N)$, had been compressed, is a priory given, and its characteristics doesn't change with time. In situation when data sequentially are fed to processing in on-line mode, their volume is not known beforehand, and the system that generates the data is non-stationary, the traditional algorithms that implement the method of principal components loses its effectiveness and adaptive procedures based on neural network technology have to be used [1–3].

In this regard, it is necessary to synthesize fast on-line learning algorithms, that can be used for the principal components estimation data streams, that are fed in the on-line mode.

1. Problem statement

In order to find first principal component and the corresponding eigenvector w_1 E. Oja [4, 5] using D. Hebb normalized learning rule [2] proposed self-learning algorithm for adaptive linear associator (Adaline)

$$y_1(k) = w_1^T(k)\tilde{x}(k) \tag{4}$$

that minimizes local (one-step) learning criterion (energy function, Lyapunov function)

$$E(k) = \frac{1}{2}\left\|\tilde{x}(k) - \hat{x}(k)\right\|^2, \tag{5}$$

where $w_1(k)-$ the current estimate of Adaline's synaptic weights, that coincides with first eigenvector w_1;

$y_1(k)-$ current estimate of the first principal component;

$\hat{x}(k) = w(k)\hat{y}_1(k)-$ restored ("uncompressed") estimate of the centered input signal $\tilde{x}(k)$.

Let's notice too, that in the present time for the Adalines learning, set of algorithms are developed, out of which the most popular is one-step procedure proposed by Widrow–Hoff, but all these algorithms are based on the supervised learning paradigm. The known algorithms are based on the recurrent least squares methods and used for stationary signals processing. In respect of the self-learning mode, the most popular procedures are based on the one-step gradient Oja's algorithm.

Considering that the gradient of criterion (5) respectively to synaptic weights has the form

$$\nabla_w E(k) = -\left(\tilde{x}(k) - \hat{x}(k)\right)y_1(k) = \\ = -\left(\tilde{x}(k) - w(k)y_1(k)\right)y_1(k),$$

Oja's rule for the first eigenvector can be written in the recurrent form

$$w_1(k+1) = w_1(k) + \eta(k)\times \\ \times y_1(k)\left(\tilde{x}(k) - w_1(k)y_1(k)\right), \tag{6}$$

and considering (4) –

$$w_1(k+1) = w_1(k) + \eta(k)\times \\ \times\left(\tilde{x}(k)\tilde{x}^T(k)w_1(k) - w_1^T(k)\tilde{x}(k)\tilde{x}^T(k)w_1(k)w_1(k)\right) = \\ = w_1(k) + \eta(k)\times \\ \times\left(\tilde{x}(k)\tilde{x}^T(k) - w_1^T(k)\tilde{x}(k)\tilde{x}^T(k)w_1(k)I_{nn}\right)w_1(k) \tag{7}$$

where $\eta(k)-$ learning rate parameter, chosen according with the A. Dvoretsky's conditions [6].

In this case, instead of (2), (3) recurrent estimates can be used

$$\bar{x}(k+1) = \bar{x}(k) + \frac{1}{k+1}\left(x(k+1) - \bar{x}(k)\right), \tag{8}$$

$$R(k+1) = R(k) + \\ + \frac{1}{k+1}\left(\tilde{x}(k+1)\tilde{x}^T(k+1) - R(k)\right). \tag{9}$$

Thus, relations (6) and (7) essentially are stochastic approximation procedures, characterized by a low speed of convergence and unsuitable for using in non-stationary conditions.

In this regard in [7–9] Oja's modified rule was introduced in the form

$$\begin{cases} w_1(k+1) = w_1(k) + \\ + r^{-1}(k)y_1(k)\left(\tilde{x}(k) - w_1(k)y_1(k)\right), \\ r(k) = \alpha r(k-1) + y_1^2(k), \\ 0 \le \alpha \le 1, \end{cases} \tag{10}$$

having the increased speed and extra smoothing properties, which are provided by a variable smoothing parameter α.

Thus instead of (8), (9) it's naturally to use relations

$$\bar{x}(k+1) = \alpha\bar{x}(k) + (1-\alpha)x(k+1),$$

$$R(k+1) = \alpha R(k) + (1-\alpha)\tilde{x}(k+1)\tilde{x}^T(k+1).$$

In this case, the algorithm (6) and rule (10) are generated by the one-step criterion (5), that is, on each iteration accounted only one current observation $x(k+1)$ that limits the ability of these procedures.

2. Hebb–Oja multistep rule

In [10] it was shown that the problem of finding the dominant eigenvalue and eigenvector can be solved by optimization of global criterion

$$E^N = w^T R(N) w = \frac{1}{N}\sum_{k=1}^{N}\left(w^T \tilde{x}(k)\right)^2$$

with constraints

$$w^T w = 1.$$

Introducing the Lagrange function

$$L(w,\lambda) = w^T R(N) w - \lambda\left(w^T w - 1\right) \qquad (11)$$

and solving the system of Karush – Kuhn – Tucker equations

$$\begin{cases} \nabla_w L(w,\lambda) = R(N) w - \lambda w = \vec{0}, \\ \dfrac{\partial L(w,\lambda)}{\partial \lambda} = w^T w - 1 = 0, \end{cases}$$

we obtain the expressions

$$\begin{cases} \lambda = w^T R(N) w, \\ \left(R(N) - \lambda I_{nn}\right) w = \vec{0}, \end{cases} \qquad (12)$$

thus comparing the second equation in (12) with (1), it can be concluded that the Lagrange's undetermined multiplier λ in (11) has a since of eigenvalue λ_1 in (1).

Since the equation

$$\left(R(N) - w^T R(N) w I_{nn}\right) w = \vec{0}$$

that follows from (12), obviously doesn't have analytical solution so for finding the saddle point of the Lagrangian (11) we can use the Arrow – Hurwitz procedure

$$\begin{cases} w_1(\tau+1) = w_1(\tau) + \eta_w(\tau) \times \\ \times\left(R(N) - \lambda_1(\tau) I_{nn}\right) w_1(\tau), \\ \lambda_1(\tau+1) = \lambda_1(\tau) - \eta_\lambda(\tau) \times \\ \times\left(w_1^T(\tau+1) w_1(\tau+1) - 1\right) \end{cases} \qquad (13)$$

where $\tau = 0,1,2...$ – machine time iterations. Taking into account the first equation (12) instead of (13) we can introduce Hebb – Oja multistep rule in the form

$$w_1(\tau+1) = w_1(\tau) + \eta_w(\tau) \times$$
$$\times\left(R(N) - w_1^T(\tau) R(N) w_1(\tau) I_{nn}\right) w_1(\tau). \qquad (14)$$

Comparing (14) with (7) we can note that (13) is a generalization of (7) and coincides with it when

$$R(N) = \tilde{x}(N)\tilde{x}^T(N).$$

Self-learning rule (14) can be easily adapted to the situation when data $\tilde{x}(k)$ are fed sequentially to the processing in on-line mode. This assumes that between the two real-time moments k and $(k+1)$ it can be realized $T(\tau = 0,1,2,...,T)$ computer time cycles.

Wherein

$$w_1(k+1,\tau+1) = w_1(k+1,\tau) + \eta_w(\tau) \times$$
$$\times\left(R(k+1) - w_1^T(k+1,\tau) R(k+1) w_1(k+1,\tau) I_{nn}\right) \times$$
$$\times w_1(k+1,\tau),$$
$$w_1(k+1,0) = w_1(k,T),$$

and for computing $R(k+1)$ expressions (8), (9) are used.

In nonstationary situations it's easy to implement the computational procedure at the sliding window s, wherein

$$\bar{x}(k+1,s) = \frac{1}{s}\sum_{l=k-s+2}^{k+1} x(l) =$$
$$= \bar{x}(k,s) + \frac{1}{s}\left(x(k+1) - x(k-s+1)\right),$$
$$R(k+1,s) = \frac{1}{s}\sum_{l=k-s+2}^{k+1} \tilde{x}(l)\tilde{x}^T(l) = R(k,s) +$$
$$+\frac{1}{s}\left(\tilde{x}(k+1)\tilde{x}^T(k+1) - \tilde{x}(k-s+1)\tilde{x}^T(k-s+1)\right),$$
$$w_1(k+1,\tau+1,s) = w_1(k+1,\tau,s) +$$
$$+\eta_w(\tau)\left(R(k+1,s) - w_1^T(k+1,\tau,s) \times\right. \qquad (15)$$
$$\left.\times R(k+1,s) w_1(k+1,\tau,s) I_{nn}\right) w_1(k+1,\tau,s).$$

When $s = 1$ the rule (15) takes the form

$$w_1(k+1,\tau+1,1) = w_1(k+1,\tau,1) +$$
$$+\eta_w(\tau)\left(\tilde{x}(k+1)\tilde{x}^T(k+1) - w_1^T(k+1,\tau,1) \times\right.$$
$$\left.\times\tilde{x}(k+1)\tilde{x}^T(k+1) w_1(k+1,\tau,1) I_{nn}\right) w_1(k+1,\tau,1)$$

and practically coincides with (7), however has derived from entirely different assumptions.

Thus, by solving the optimization problem (11) Hebb–Oja multistep self-learning rule can be obtained.

3. Experimental modeling

Efficiency of proposed multistep self-learning rule for adaptive linear associator is demonstrated in table 1, and has been investigated on data sets from the UCI repository and compared with the principal component analysis (PCA), and adaptive linear associator "bottle neck". For the first experimental series, three marked data sets of observations were taken. Data set "Iris" consists of 150 observations; each of all observations has 4 features. Data set "Wine" consists of 178 observations; each of all observations has 13 features. Data set "Wisconsin Diagnostic Breast Cancer" (WDBC) consists of 569 observations; each of all observations has 30 features. The working results of the principal component analysis (PCA), and adaptive linear associator "bottle neck" were compared with the multistep self-learning rule for adaptive linear associator. The average, minimum and maximum results for a series of 50 experiments are shown in table 1 and time required is shown in table 2. As estimates of the results mean absolute percentage error was used MAPE.

Efficiency of the adaptive linear associator based on multistep self-learning rule higher and more stable than PCA and adaptive linear associator "bottle neck" is shown by experimental series.

Although the proposed algorithm is more bulky in computational point of view comparatively whit the standard one-step Oja's rule, it's provides high operating speed and also can be used for processing of nonstationary short data sets in on-line mode.

Table 1

Comparison of different data sets compression accuracy.

Methods for data compression		Iris			Wine			Cancer	
	min	avg	max	min	avg	max	min	avg	max
PCA	60	72	79	68	71	73	55	67	73
Adaptive linear associator "bottle neck"	54	65	69	49	60	70	57	64	75
Adaptive linear associator based on multistep self-learning rule	58	69	73	59	66	69	66	69	76

Table 2

Time required comparison of investigated methods for data compression.

Methods for data compression		Time(s)	
	Iris	Wine	Cancer
PCA	0.02	0.58	2.527
Adaptive linear associator "bottle neck"	<0.01	0.13	0.28
Adaptive linear associator based on multistep self-learning rule	<0.01	<0.01	0.01

Conclusions

Multistep self-learning rule for adaptive linear associator designed for finding eigenvalues and eigenvectors of the correlation matrix data that sequentially are fed to processing in on-line mode had suggested. This rule is a generalization of D. Hebb and E. Oja algorithms, used for neural networks training, implementing the method of principal components.

The proposed system provides high operating speed and is designed for solving Data Stream Mining and Dynamic Data Mining tasks in conditions of nonstationary input data processing that are sequentially fed to processing in the on-line mode.

References:

1. Cichocki, A., Unbehauen, R. (1993) Neural Networks for Optimization and Signal Processing. Stuttgart: Teubner, 526 p.

2. Haykin, S. (1999) Neural Networks. A Comprehensive Foundation. Upper Saddle River, N. J.: Prentice Hall, Inc. p. 842.

3. Ham, F. M., Kostanic, I. (2001) Principles of Neurocomputing for Science and Engineering. N.Y.:Mc Graw-Hill, Inc., 642 p.

4. Oja, E. (1982) A simplified neuron model as a principal component analyzer. J. of Math. Biology. 15. pp. 267–273.

5. Oja, E. (1992) Principal component, minor components, and linear neural networks. Neural Networks. 5. pp. 927–935.

6. Vazan, M (1972) Stochastic approcsimation. M.: Myr, 289 p. (In Russian)

7. Bodyanskiy, Ye.V., Pliss, I. P., Teslenko, N. O. (2006) Optimal learning algorithm of Oja's neuron. Decision making theory. :Proc. of the Inter. conf. Uzhgorod. pp.10-11. (in Ukrainian)

8. Bodyanskiy, Ye.V., Pliss, I. P., Teslenko, N. O. (2006) Modification of Oja's neuron for nonstationary data analysis. Automation: problems, ideas and solutions.: Proc. of the Inter. conf. Sebastopol. pp.17–21. (in Russian)

9. Bodyanskiy, Ye.V., Deineko, A.O.,. Teslenko, N. O., Shalamov, M. O. (2011) Evolving cascade neural network for sequential principal component analysis and its learning. System tecnologies. Dnepropetrovsk. № 1(72), V. 2. pp. 140–147. (in Russian)

10. Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Moor, B. D., Vandewalle, J. (2002) Least Squares Support Vector Machines. Singapore: World Scientific. p. 294.

АДАПТИВНАЯ МНОГОШАГОВАЯ МОДИФИКАЦИЯ ПРАВИЛА САМООБУЧЕНИЯ В ЗАДАЧЕ АНАЛИЗА ГЛАВНЫХ КОМПОНЕНТ

Е. В. Бодянский, А. А. Дейнеко, М. А. Шаламов
Харьковский национальный университет радиоэлектроники

Анотация. *В статье предложено многошаговое правило самообучения адаптивного линейного ассоциатора, предназначенное для нахождения в on-line-режиме собственных значений и векторов корреляционных матриц данных, последовательно поступающих на обработку.*

Ключевые слова: *интеллектуальный анализ данных, анализ главных компонент, компрессия данных, собственный вектор, нейронная сеть.*

АДАПТИВНА БАГАТОКРОКОВА МОДИФІКАЦІЯ ПРАВИЛА САМОНАВЧАННЯ В ЗАДАЧІ АНАЛІЗУ ГОЛОВНИХ КОМПОНЕНТ

Є. В. Бодянський, А. О. Дейнеко, М. О. Шаламов
Харківський національний університет радіоелектроніки

Анотація. *У статті запропоновано багатокрокове правило самонавчання адаптивного лінійного асоціатора для знаходження в послідовному режимі власних значень та власних векторів кореляційних матриць даних, які послідовно надходять на опрацювання.*

Ключові слова: *інтелектуальний аналіз даних, аналіз головних компонент, компресування даних, власні вектори, нейронна мережа.*

Received 14.08.2016

Бодянский Евгений Владимирович, руководитель ПНИЛ АСУ, д. т. н., проф., профессор кафедры искусственного интеллекта Харьковского Национального университета радиоэлектроники, Харьков, пр. Ленина, 14, ауд. 511, E-mail: yevgeniy.bodyanskiy@nure.ua

Bodyanskiy Yevgeniy Volodymyrovych, The Scientific Head of Control Systems Research Laboratory, Professor at Artificial Intelligence Department, Dr. Sci., Professor, Kharkiv National University of Radio Electronics, 14 Nauka Ave, Kharkiv, 61166, Ukraine

ORCID ID: 0000-0001-5418-2143

Дейнеко Анастасия Александровна, старший научный сотрудник, ПНИЛ АСУ, к. т. н., Харьковский национальный университет радиоэлектроники, Харьков, пр. Ленина, 14, ауд. 511, E-mail: Anastasiya.deineko@gmail.com, Тел. (050)753-60-50

Deineko Anastasiia Oleksandrivna, Senior Researcher at Control Systems Research Laboratory, Ph. D., Kharkiv National University of Radio Electronics, 14 Nauka Ave, Kharkiv, 61166, Ukraine

ORCID ID: 0000-0002-3279-3135

Шаламов Максим Алексеевич, аспирант кафедры искусственного интеллекта, Харьковский национальный университет радиоэлектроники,Харьков, пр. Ленина, 14., E-mail: mshalamov@ukr.net

Shalamov Maksym Oleksiyovich, Ph. D. student,Kharkiv National University of Radio Electronics, 14 Nauka Ave, Kharkiv, 61166, Ukraine

ORCID ID: 0000-0003-4417-6924