Olga VESELSKA[1], Oleksandr PETROV[2], Anton PETROV[3], Gulmira SHANGYTBAYEVA[4], Ruslana ZIUBINA[5]

PRZETWARZANIE INFORMACJI CYFROWYCH Z WYKORZYSTANIEM ANALIZY GŁÓWNYCH SKŁADOWYCH

Streszczenie: Artykuł dotyczy Analizy Głównych Składowych (ang. PCA - Principle Component Analysis). Głownym celem jest omówienie oraz przedstawienie przykładowego zastosowania algorytmu obliczeniowego metody PCA. Tak więc: opisano iteracyjny algorytm obliczania PCA oraz przedstawiono przykład zastosowania w grafice komputerowej. Główne obszary cyfrowego przetwarzania informacji to: przetwarzanie obrazów, kompresja, synteza, rozpoznawanie, identyfikacja. Przedstawiono przykład przejścia z modelu RGB na model trójskładnikowy. W artykule przedstawiono szczegółową implementację metody PCA do przetwarzania obrazów w Pythonie.

Słowa kluczowe: Podstawowa analiza składników, algorytm iteracyjny, przetwarzanie obrazu

PROCESSING OF DIGITAL INFORMATION USING THE PRINCIPLE COMPONENT ANALYSIS

Summary: The article is devoted to the Principle Component Analysis (PCA). The purpose of the article is to give the reader some information on the computational algorithm of the PCA method and its application for a concrete example. An iterative algorithm for calculating the PCA is described; an example of the use of the Principle Component Analysis in the field of computer graphics is given. Traditionally, the main areas of digital information processing are image-processing (compression, synthesis, recognition, identification). An example of the optimal transition from the RGB model to the optimal three-component model is given. The article gives a detailed implementation of the PCA method for image processing in Python.

Keywords: Principle Component Analysis, iterative algorithm, image processing

[1]University of Bielsko-Biala, Department of Computer Science and Automatics, oveselska@ath.edu.pl

[2] AGH University of Science and Technology, Krakow, Poland, asp1951@gmail.com

[3] Federal State Budgetary Educational Institution of Higher Education “Kuban State Agrarian University named after I.T. Trubilin”, Krasnodar, Russia, anton.a.petrov@gmail.com

[4] K.Zhubanov Aktobe Regional State University, Aktobe, Kazakhstan, gul_janet@mail.ru

[5] University of Bielsko-Biala, Department of Computer Science and Automatics, rziubina@ath.edu.pl

1. Introduction

The principal component analysis is the one of the main way to reduce data dimension with loss of the minimum quantity information, developed by Karl Pearson in 1901. Basically, it is applied in many areas, such as: pattern recognition, data visualization, image and video compression, image noise reduction, video indexing, bioinformatics and even social sciences [1-5]. Sometimes the PCA method calls the Hotelling transform (*the Karhunen-Loeve Transform KLT)* (see, for example, [1-4]).

Recently, approaches that work with images as an array of features, each of which reflects the value of one pixel of the image (for example, the brightness of a single pixel), have become widespread. Since the vector of each image has a sufficiently high dimension, the task of processing a large number of images is not trivial. Most image analysis systems are based on methods that reduce the dimensionality of images. The problem of reducing the dimension is also important because the complexity of most algorithms increases exponentially with increasing dimension of the problem. Here one of the most common image reduction techniques is PCA.

The idea of using the PCA for solving the image processing (recognition, compression etc) problem is that the image can be represented as a linear combination of the eigenvectors of the covariance matrix. The eigenvectors have the same dimension as the image itself, but are linearly independent, which allows you to accurately reconstruct the image. Eigenvalues determine the degree of contribution of each eigenvector to image reconstruction.

The main components are called eigenvectors with the largest eigenvalues. Each image can be described by a weighted combination of principal components. Thus, for each image, it is enough to store only a set of weighting factors (main factors), but each factor already reflects not a single pixel, but a group of pixels, which can be represented as a picture.

Usually, the PCA method is applied for the solution of the image recognition problem. It includes the following steps: a) normalization of the original image, b) calculation of the principal components; c) classification of images in a reduced space (clustering, neural network etc.). The recognition process consists in comparing the main factors of the desired image with the main factors of all other images. It is assumed that images of faces corresponding to one person are grouped into clusters in the image space. The result of the search are those images that have the smallest distance to the desired image.

Here we analyze another problem of image processing. Namely, image compression which is based on the PCA method. For this purpose we develop a modification of the method and apply it for the RGB image model.

2. The main idea of PCA

Let's consider the basic steps of the PCA method. At the beginning we will find the constant μ, which in the best way describing input data

$$\varepsilon(\mu) = \sum_{i=1}^{n} \quad (x_i - \mu)^2 \to \min_{\mu}.$$

For finding of the minimum we will equate the derivative to zero and find value μ delivering the minimum

$$\frac{d}{d\mu}\varepsilon(\mu) = -2\sum_{i=1}^{n} \quad (x_i - \mu) = 0 \Rightarrow \sum_{i=1}^{n} \quad \mu = \sum_{i=1}^{n} \quad x_i \Rightarrow \mu n = \sum_{i=1}^{n} \quad x_i \Rightarrow \mu = \frac{1}{n}\sum_{i=1}^{n} \quad x_i.$$

Furthermore, we will carry out mean-centering, that is, we can redefine original values as follows $x_{new} = x_{old} - \mu$, i.e from each value xi the mean μ is subtracted. It is clear, that new data have the average equal to zero.

$$E\big(X - E(X)\big) = E(X) - E(X) = 0.$$

As a matter of fact, we made parallel translation in the existing coordinate system. In what follows we assume that input data is centered. Not to mention that, we are going to find the most faithful representation of data $D = \{x_1, \dots, x_n\}$ in some W subspace which has dimension k <n.

Let $\{e_1, \dots, e_k\}$ be orthonormal basis of W. Any vector from W can be written in the form of linear combination of basis vectors, therefore for x_1 it is possible to determine vector in the following form $\sum_{i=1}^{k} \quad \alpha_{1,i}e_i$. The error between them is calculated as follows

$$\varepsilon_1 = \|x_1 - \sum_{i=1}^{k} \quad \alpha_{1,i}e_i\|_2^2 = \langle x_1 - \sum_{i=1}^{k} \quad \alpha_{1,i}e_i, x_1 - \sum_{i=1}^{k} \quad \alpha_{1,i}e_i\rangle.$$

To find the integral error, we need to sum the quantities errors over all x_j, so the total error is

$$\varepsilon\big(e_1, \dots, e_k, \alpha_{1,1}, \dots, \alpha_{n,k}\big)_{\underbrace{}_{unknowns}} = \sum_{j=1}^{n} \quad \varepsilon_j^2 = \sum_{j=1}^{n} \quad \|x_j - \sum_{i=1}^{k} \quad \alpha_{j,i}e_i\|_2^2. \tag{1}$$

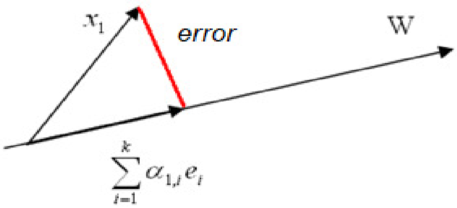

Figure. 1. Vector error recovery illustration

To minimize the error, it is necessary to calculate the partial derivative of an above mention function with respect to the coefficients and consider restrictions for orthogonality $\{e_1, \dots, e_k\}$. At the beginning we will simplify the expression (equation) (1)

$$\varepsilon\left(e_1,\dots,e_k,\alpha_{1,1},\dots,\alpha_{n,k}\right)=\sum_{j=1}^{n}\quad \|x_j-\sum_{i=1}^{k}\quad \alpha_{j,i}e_i\|_2^2$$

$$=\sum_{j=1}^{n}\quad \|x_j\|_2^2-2\sum_{j=1}^{n}\quad x_j^T\sum_{i=1}^{k}\quad \alpha_{j,i}e_i+\sum_{j=1}^{n}\quad\sum_{i=1}^{k}\quad \alpha_{j,i}^2=$$

$$\textstyle\sum_{j=1}^{n}\quad \|x_j\|_2^2-2\sum_{j=1}^{n}\quad \sum_{i=1}^{k}\quad \alpha_{j,i}x_j^Te_i+\sum_{j=1}^{n}\quad \sum_{i=1}^{k}\quad \alpha_{j,i}^2.$$

Then

$$\frac{\partial}{\partial\alpha_{m,l}}\varepsilon\left(e_1,\dots,e_k,\alpha_{1,1},\dots,\alpha_{n,k}\right)=-2x_m^Te_l+2\alpha_{m,l}.$$

The necessary and sufficient condition for the extremum will take the following form

$$-2x_m^Te_l+2\alpha_{m,l}=0\Rightarrow\alpha_{m,l}=x_m^Te_l.$$

Thus, the error (1) will be described in the form

$$\varepsilon(e_1,\dots,e_k)=\textstyle\sum_{j=1}^{n}\quad \|x_j\|_2^2-2\sum_{j=1}^{n}\quad \sum_{i=1}^{k}\quad \left(x_j^Te_i\right)x_j^Te_i+$$
$$\textstyle\sum_{j=1}^{n}\quad \sum_{i=1}^{k}\quad \left(x_j^Te_i\right)^2.$$

After simplifying that equation, we receive

$$\varepsilon(e_1,\dots,e_k)=\textstyle\sum_{j=1}^{n}\quad \|x_j\|_2^2-\sum_{j=1}^{n}\quad \sum_{i=1}^{k}\quad \left(x_j^Te_i\right)^2 .\text{(2)}$$

Taking into account that $\langle a,b\rangle=a^Tb$ and $\langle b,a\rangle=\langle a,b\rangle$,we obtain

$$(a^Tb)^2=(a^Tb)(a^Tb)=(b^Ta)(a^Tb)=b^T(aa^T)b,$$

therefore,

$$\varepsilon(e_1,\dots,e_k)=\textstyle\sum_{j=1}^{n}\quad \|x_j\|_2^2-\sum_{i=1}^{k}\quad e_i^T\left(\sum_{j=1}^{n}\quad \left(x_jx_j^T\right)\right)e_i=\sum_{j=1}^{n}\quad \|x_j\|_2^2-$$
$$\textstyle\sum_{i=1}^{k}\quad e_i^TSe_i,$$

where $S=\sum_{j=1}^{n}\quad \left(x_jx_j^T\right)$. is the covariation matrix.
Next will be the minimization $\varepsilon(e_1,\dots,e_k)=\sum_{j=1}^{n}\quad \|x_j\|_2^2-\sum_{i=1}^{k}\quad e_i^TSe_i$ under the condition $e_i^Te_i=1$ for all *i*. Using the method of indefinite Lagrange multipliers, we enter multipliers $\lambda_1,\dots,\lambda_k$and, noticing that $\sum_{j=1}^{n}\quad \|x_j\|_2^2\equiv Const$ Then we can describe the objective function

$$l(e_1,\dots,e_k)=\sum_{i=1}^{k}\quad e_i^TSe_i-\sum_{i=1}^{k}\quad \lambda_i(e_i^Te_i-1).$$

It is worth pointing out that $\frac{d}{dX}(X^TX)=\frac{d}{dX}\langle X,X\rangle=2X$ and if A is a symmetric matrix, then $\frac{d}{dX}(X^TAX)=2AX$, as the result we get,

$$\frac{\partial}{\partial e_m}l(e_1,\dots,e_k)=2Se_m-2\lambda_me_m=0,$$

that is, $Se_m = \lambda_m e_m$. Thus, it is necessary to find the solution of the equation $(S - \lambda I)e = 0$ (here I- an identity matrix) that determine λ_m as eigenvalues and e_m as eigenvectors of the covariation matrix of S.
In this case, the error takes the following form

$$\varepsilon(e_1,\dots,e_k) = \sum_{j=1}^{n} \|x_j\|_2^2 - \sum_{i=1}^{k} \lambda_i \|e_i\|_2^2 = \sum_{j=1}^{n} \|x_j\|_2^2 - \sum_{i=1}^{k} \lambda_i. \quad (3)$$

Minimization (3) consists in the choice k of the greatest eigenvalues and its corresponding eigenvectors of covariance matrix S. The bigger eigenvalue of the matrix S gives the big variation in the direction to the corresponding eigenvector. This result can be reformulated as follows – the projection X on the k - dimension subspace provides the greatest variation. Thus, the PCA can be treated as follows - we take orthogonal basis and rotate it on one of the directions as long as we do not receive the maximum variation. We fix this direction and we rotate the others, until we find the second direction yet and so on[5-6].
Let $\{e_1,\dots,e_n\}$ all eigenvectors of S matrix be ordered in respect to the corresponding eigenvalue, then for any

$$x_i = \sum_{j=1}^{n} \alpha_{i,j} e_j = \alpha_{i,1}e_1 + \dots + \alpha_{i,k}\underbrace{e_k}_{approximation} + \alpha_{i,k+1}e_{k+1} + \dots + \alpha_{i,n}\overbrace{e_n}^{error}$$

The coefficient $\alpha_{m,l} = x_m^T e_l$ are the coordinates of the main components, for the greater value k gives the best approximation. At the same time, the main components are ordered according to the importance, more important at the beginning and less important at the end.

Let's give the algorithm of PCA.

Let input (original) data $D = \{x_1^0,\dots,x_n^0\}$, where each of these vectors x_i^0 has dimension of N

- Let's find the average $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i^0$.
- Let's subtract the average from each vector $x_i = x_i^0 - \mu$.
- Let's find the covariation matrix $S = \sum_{j=1}^{n} x_j x_j^T$.
- Let's calculate eigenvectors $\{e_1,\dots,e_k\}$, corresponding k to the greatest eigenvalues of S.
- Let $\{e_1,\dots,e_k\}$ form the matrix $E = [e_1 \dots e_k]$.
- Then the closest approximation to x is $z = E^T x$.

Let's review the example.

The set of data $D^0\big((x_1^0, y_1^0),\dots,(x_8^0, y_8^0)\big)$ is determined by the table:

Table 1. Input data

x	1	2	3	4	5	6	7	8
y	2	3	2	4	4	7	6	7

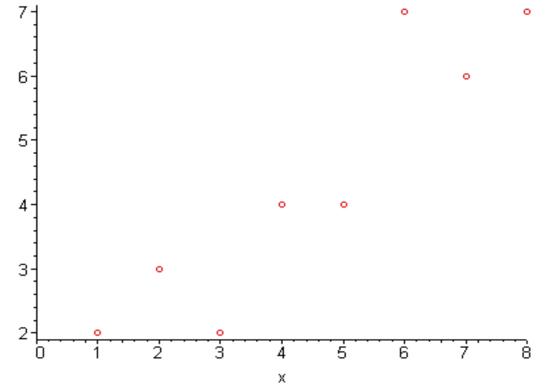

Figure 2. Input data

Let's find mean value $\mu = (4.5, 4.375)$, then after centering data Dwill take the form

Table 2. The centered data

x	- 3.5	- 2.5	- 1.5	- 0.5	0.5	1.5	2.5	3.5
y	- 2.38	- 1.38	- 2.38	- 0.38	- 0.38	2.63	1.63	2.63

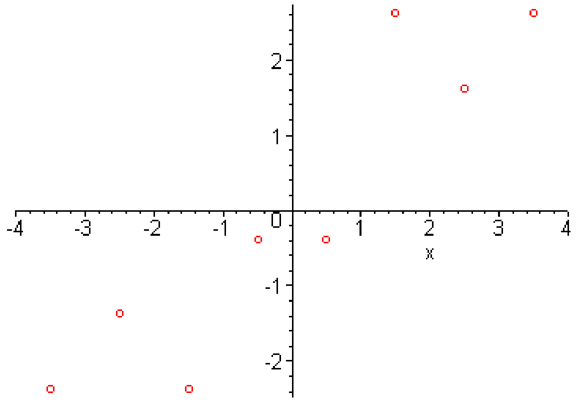

Figure 3. Parallel shift, combining the origin of coordinates with the mathematical expectation of input data

Then

$$s_{1,1} = \langle x, x\rangle = \sum_{i=1}^{8} x_i x_i = 42,$$

$$s_{2,1} = s_{1,2} = \langle x, y \rangle = \sum_{i=1}^{8} \quad x_i y_i = 32{,}5,$$

$$s_{2,2} = \langle y, y \rangle = \sum_{i=1}^{8} \quad y_i y_i = 29{,}875,$$

and the covariation matrix can be written as follows

$$S = \left(s_{1,1}\ s_{1,2}\ s_{2,1}\ s_{2,2}\ \right) = (42\ 32{,}5\ 32{,}5\ 29{,}875\ ).$$

Solving the equation

$$|42 - \lambda\ 32{,}5\ 32{,}5\ 29{,}875 - \lambda\ | = 0 \Leftrightarrow (42 - \lambda)(29{,}875 - \lambda) - (32{,}5)^2 = 0$$

we receive eigenvalues $\lambda_1 = 68{,}99810959, \lambda_2 = 2{,}876890413.$

For determining the eigenvectors $e_1 = \left(e_{1,1}, e_{1,2}\right)^T$ and $e_2 = \left(e_{2,1}, e_{2,2}\right)^T$ let's find any uncommon solution of the following system

$$\{\left(s_{1,1} - \lambda_1\right)e_{1,1} + s_{1,2}e_{1,2} = 0, s_{1,2}e_{1,1} + \left(s_{2,2} - \lambda_1\right)e_{1,2} = 0,$$

and, respectively, for second eigenvalues

$$\{\left(s_{1,1} - \lambda_2\right)e_{2,1} + s_{1,2}e_{2,2} = 0, s_{1,2}e_{2,1} + \left(s_{2,2} - \lambda_2\right)e_{2,2} = 0.$$

Under the eigenvalue λ in the both system of equations the main determinant is equal zero. So in both cases the equations are linearly dependent. For finding the solutions it is necessary taking any nonzero values for first unknown and the second unknown determines from the one of equation respectively to each system of equations. Therefore for the example $e_{1,1} = 1, e_{1,2} = 0{,}8307110643$ and $e_{2,,1} = 1, e_{2,2} = -1{,}203787987.$

Thus, the vector $e_1 =$ corresponds to the eigenvalue $\lambda_1 = 68.99810959$, and the value $\lambda_2 = 2.876890413$ corresponds the vector $e_2 =$ Bigger eigenvalue corresponds greater principal direction. After normalizing eigenvectors to unit length, we get $e_1 =$ and $e_2 =$
It is necessary to determine the first main component $z_1 = e_1^T D$ (table 3).

Table 3. The first main component

z_1	-4,2099	-2,8016	-2,67142	-0,6243	0,14498	2,83117	2,96139	4,3696

Respectively, the second main component $z_2 = e_2^T D$ will take the form as included within table 4

Table 4. The second main component

z_2	-0,41	-0,5398	0,8684	-0,031	0,608	-1,06	0,348	0,2173

Let's outline that for receiving a result for input data (uncentered) it is necessary to add the corresponding mean value.

The recovery of data referring to first main component (that is projections of input data to principal direction) will take the form $x_i = e_{1,1} z_{1,i} + \mu_1, y_i = e_{1.2} z_{1,i} + \mu_2$ (table 3)

Table 5. The input data recovered on the first main the component

$\mathbf{0,769 \times z_1 + 4,5}$	1,2617	2,345	2,445	4,02	4,612	6,678	6,778	7,861
$\mathbf{0,639 \times z_1 + 4,37}$	1,6849	2,585	2,668	3,976	4,468	6,184	6,267	7,167

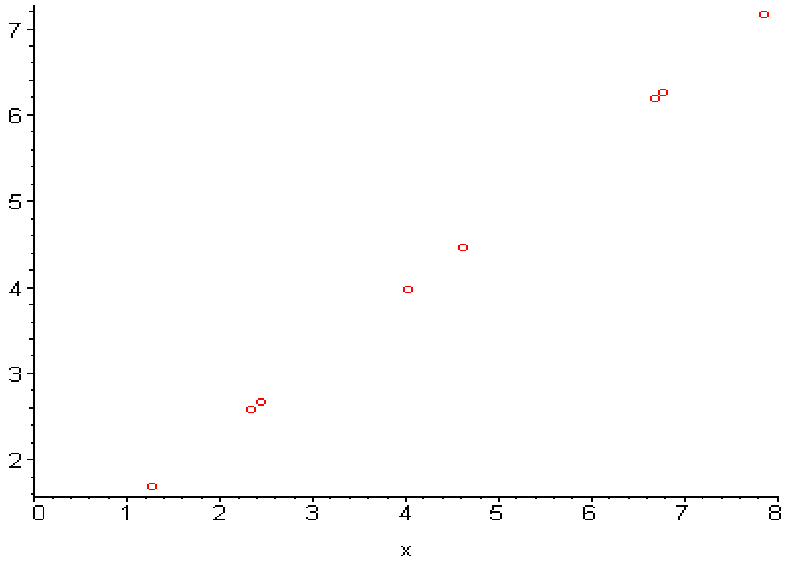

Figure 4. The data presentation of the first main component

3. An iteration algorithm for computing the principal components

If The described above method of determining the principal component is rather resource-intensive and unstable, especially if eigenvalues of the matrix are close to zero.

Basically, more effective is use of the iterative method of principal component. To achieve this aim, we can consider the task (1) from a different point of view[7-8].

For i=1 case, the task (1) comes down to definition referring to the first component e_1, which recovers all input data $\{x_1, \dots, x_n\}$ the most efficiently.

$$\varepsilon(e_1, \alpha_{1,1}, \dots, \alpha_{n,1}) = \sum_{j=1}^{n} \quad \|x_j - \alpha_{j,1} e_1\|_2^2 \to min \tag{4}$$

on all e_1 and $\{\alpha_{i,1}\}_{i=1}^{n}$under the condition $\sum_{i=1}^{n} \quad \alpha_{i,1}^2 = 1$.

If $\{\alpha\sim_{i,1}\}_{i=1}^{n}$ and $e\sim_1$ there is the solution of this task and $\Delta x_j = x_j - \alpha\sim_{j,1}e\sim_1$ - the error of data recovery based on the first main component, solving the following problem

$$\sum_{j=1}^{n} \quad \|\Delta x_j - \alpha_{j,2}e_2\|_2^2 \rightarrow min$$

on all e_2 and $\{\alpha_{i,2}\}_{i=1}^{n}$ under the condition $\sum_{i=1}^{n} \quad \alpha_{i,1}^2 = 1$, we receive the second main component $e\sim_2$ and corresponding vector $\{\alpha\sim_{i,2}\}_{i=1}^{n}$ etc[9].

At fixed $\{\alpha_{i,1}\}_{i=1}^{n}$ the problem (4) can be solved by the least-squares method. Knowing that the objective function represents the quadratic functional, the necessary and sufficient condition of the extremum are identical. Thus, the solution of the task comes down to solving to an equation

$$\frac{\partial}{\partial e_1}\varepsilon(e_1,\alpha_{1,1},\ldots,\alpha_{n,1}) = -2\sum_{j=1}^{n} \quad (x_j - \alpha_{j,1}e_1)\alpha_{j,1} = -2(\sum_{j=1}^{n} \quad x_j\alpha_{j,1} - \sum_{j=1}^{n} \quad \alpha_{j,1}^2 e_1).$$

From here we receive

$$e_1 = \frac{\sum_{j=1}^{n} \quad x_j\alpha_{j,1}}{\sum_{j=1}^{n} \quad \alpha_{j,1}^2},$$

considering the rationing condition unit, that is $\sum_{i=1}^{n} \quad \alpha_{i,1}^2 = 1$, we get

$$e_1 = \sum_{j=1}^{n} \quad x_j\alpha_{j,1}.$$

We can take the following step proceeding on the assumption that in the task (4) we know the component e_1 . Also, it is required to find the extremum on $\{\alpha_{i,1}\}_{i=1}^{n}$

$$\frac{\partial}{\partial \alpha_{v,1}}\varepsilon(e_1,\alpha_{1,1},\ldots,\alpha_{n,1}) = -2(x_v - \alpha_{v,1}e_1)e_1 = -2(\langle x_v,e_1\rangle - \alpha_{v,1}\langle e_1,e_1\rangle) = 0,$$

that is

$$\alpha_{v,1} = \frac{\langle x_v,e_1\rangle}{\langle e_1,e_1\rangle},$$

where, as usual $\langle x,y\rangle$is scalar product of vectors *x* and *y*.

Further, including known $\{\alpha_{i,1}\}_{i=1}^{n}$, we repeat all process. Moreover, there will be no stabilization of the error yet. After getting e_1 let's consider the first main component $e\sim_1$. Then $\Delta x_j = x_j - \alpha\sim_{j,1}e\sim_1$ - error recovery of data of the first main component. Applying this algorithm to the error recovery Δx_j, we find the second main component e_2 together with coefficients $\alpha_{j,2}$, etc[10-12].

Let's give the algorithm of this method[13, 14].

At the beginning we center data, subtracting mean value from input data and further we consider that data are averagely equal to zero.

1. Let's put number of iteration $v = 1$.

2. We choose starting values $\{\alpha_{i,1}^{\nu}\}_{i=1}^{n}$, for example, let all of them be equal among themselves, that is $\alpha_{i,1}^{\nu} = \frac{1}{\sqrt{n}}, i = 1,2,\dots,n$.
3. We calculate $e_1^{\nu} = \sum_{j=1}^{n} \quad x_j \alpha_{j,1}^{\nu}$.
4. Further we find $\beta_i = \frac{\langle x_i, e_1^{\nu}\rangle}{\langle e_1^{\nu}, e_1^{\nu}\rangle}$, and, normalizing it to unit length, we receive $\alpha_{i,1}^{\nu+1} = \frac{\beta_i}{\sqrt{\sum_{j=1}^{n} \quad \beta_j^2}}$.
5. Next, we increase $\nu = \nu + 1$.
6. We perform the inspection of stop criterion i.e., as stabilization of coefficients $\{\alpha_{i,1}^{\nu}\}_{i=1}^{n}$, stabilization main components e_1^{ν}, or in advance check the set fixed number of iterations. If the condition of the repetitive process is not satisfied, then we pass to point 3.

Let's illustrate the iteration scheme of search the principal components on the same example which is stated above.
For already centered data (see table 2) we will give several iterations. So, let in the beginning $\nu = 1$ and $\alpha_{i,1}^{1} = \frac{1}{\sqrt{2}}, i = 1,2$. Calculating $e_{1,j}^{1} = \alpha_{1,1}^{1} x_j + \alpha_{2,1}^{1} y_j$, we receive (table 6)

Table 6. First approximation the main component

e_1^1	-4.1542	-2,740	-2,740	-0,619	0,088	2,917	2,917	4,33

Further we will calculate $\beta_i = \frac{\langle x_i, e_1^1\rangle}{\langle e_1^1, e_1^1\rangle} = (0.7697, 0.64447)$ and after the normalization we receive

$$\alpha_{i,1}^{2} = \frac{\beta_i}{\sqrt{\beta_1^2 + \beta_2^2}} = (0.7667, 0.64343).$$

Thus, after the first iteration approximate values of input data will be equal $x\sim_i = \alpha_{1,1}^{2} e_{1,i}^{1} + \mu_1$, $y\sim_i = \alpha_{2,1}^{2} e_{1,i}^{1} + \mu_2$ (compare results with table 5):

Table 7. The input data recovered on first approximation main components

$x\sim$	1,315	2,399	2,399	4,026	4,568	6,736	6,736	7,82
$y\sim$	1,702	2,612	2,612	3,977	4,432	6,252	6,252	7,172

After **ten iterations, we receive** $\boldsymbol{x\sim_i = \alpha_{1,1}^{11} e_{1,i}^{10} + \mu_1}$, $\boldsymbol{y\sim_i = \alpha_{2,1}^{11} e_{1,i}^{10} + \mu_2}$**(compare results with table** 5):

Table 8. The input data recovered on the tenth iteration of approximating main components

x	1,262	2,345	2,445	4,02	4,612	6,678	6,778	7,861
y	1,685	2,585	2,668	3,976	4,468	6,184	6,267	7,167

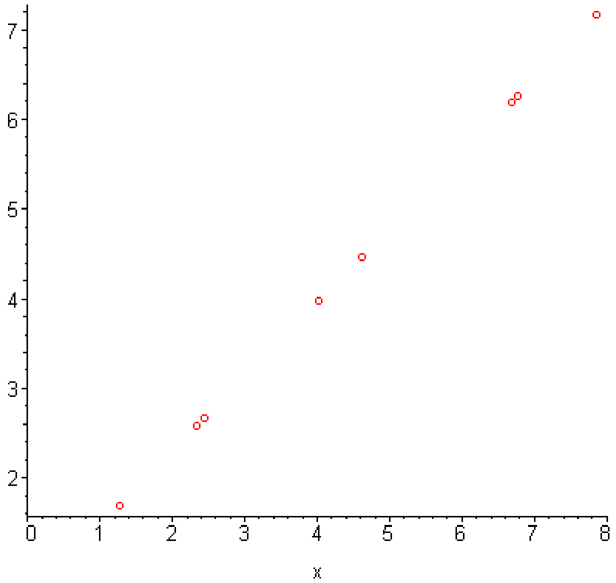

Figure 5. Data presentation recovered on the tenth iteration of approximating main components

4. Optimum transition from the RGB model to optimum three-component model

Let's give one example of use of the PCA in such area as computer graphics. Here outlining the transfer of the image from space of equal color characteristics in space of unequal is vital. Any images are visualized with use of mixing equal color component - red, green and blue making - the RGB model. As unequal color the component is used, as a rule, respectively, too three by components – value of illumination (the luminescent component), the characteristic of warm colors and the characteristic of cold tones. The avail of unequal color component is used for compression of images and video flows, applying to each of unequal the component the method of compression (see, for example, [14]).

It is possible to approach the problem of unequal color space cration from a different perspective, proceeding with the maximum informational content of everyone components. Let's apply the method of principal component to receiving unequal three-component model of the root-mean-square error of recovery of the initial image, optimum from the point of view of minimization. That is, the first of their received color the component will carry the most information about the image among all received color components, and the second will contain the most great information among the remaining.

Thus, in our terminology, the task (2.1) will take the form

$$||R - \sum_{i=1}^{3} \quad \alpha_{r,i}e_i||_2^2 + ||G - \sum_{i=1}^{3} \quad \alpha_{g,i}e_i||_2^2 + ||B - \sum_{i=1}^{3} \quad \alpha_{b,i}e_i||_2^2 \to min,$$

where the minimum undertakes on all $\alpha_{r,i}, \alpha_{g,i}, \alpha_{b,i}$ and e_i i=1,2,3.

As data we can consider the test image Lena.

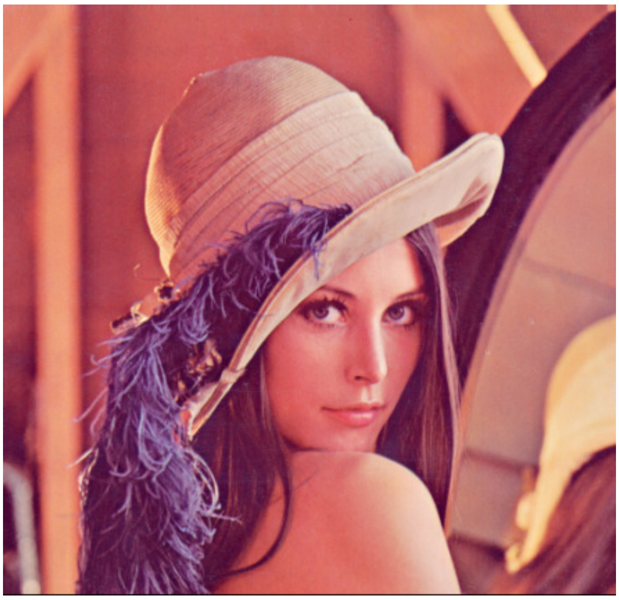

Figure 6. Test image Lena

Applying the principal components method, we receive

$$(R\ G\ B\ ) = \\ (\ 0.767785 \quad 0.45439 \quad 0.4517034)(-0.6164395 \quad 0.716085 \quad 0.3274513)\ \ .$$

Here, in the first column of the matrix, there are coefficients $\alpha_{r,1}, \alpha_{r,2}, \alpha_{r,3}$, in the second - $\alpha_{g,1}, \alpha_{g,2}, \alpha_{g,3}$ and in the third -$\alpha_{b,1}, \alpha_{b,2}, \alpha_{b,3}$. Then recovery of the test image after one component can be written down as follows

Ri, j = 0.767785 Yi, j,

Gi, j = 0.45439 Yi, j,

Bi, j = 0.4517034 Yi, j,

where Yi, j – values of the first main components e_1, corresponding to pixel with coordinates (i, j).

Also, we can compress an image using PCA without a significant loss of its variance. The earlier in this paragraph we have demonstrated using PCA to compress high dimensional data to lower dimensional data. We wanted to briefly mention that PCA can also take the compressed representation of the data (lower dimensional data) back to an approximation of the original high dimensional data. If you are interested in the code that produces the image in fig. 7, check out example code in python below.

```
# Image reconstruction from compressed representation
from sklearn.decomposition import PCA
from pylab import *
from skimage import data, io, color
import matplotlib.pyplot as plt
from matplotlib import gridspec
file = "Lenna.png"
lenna = io.imread(file, as_grey=True)
gs = gridspec.GridSpec(2, 2, width_ratios=[1, 1])
fig = plt.figure(figsize=(8, 8))
fig.subplots_adjust(hspace=0.4, wspace=0.4)
plt.subplot(gs[0])
io.imshow(lenna)
xlabel('Original Image')
for i in range(1, 4):
    n_comp = 5 ** i
    pca = PCA(n_components=n_comp)
    pca.fit(lenna)
    lenna_pca = pca.fit_transform(lenna)
    lenna_restored = pca.inverse_transform(lenna_pca)
    plt.subplot(gs[i])
    io.imshow(lenna_restored)
    xlabel('Restored image n_components = %s' % n_comp)
    print('Variance retained %s %%' % (
       (1        -        sum(pca.explained_variance_ratio_)
        / size(pca.explained_variance_ratio_)) * 100))
    print('Compression Ratio %s %%' % (float(size(lenna_pca))
/ size(lenna) * 100))
show()
```

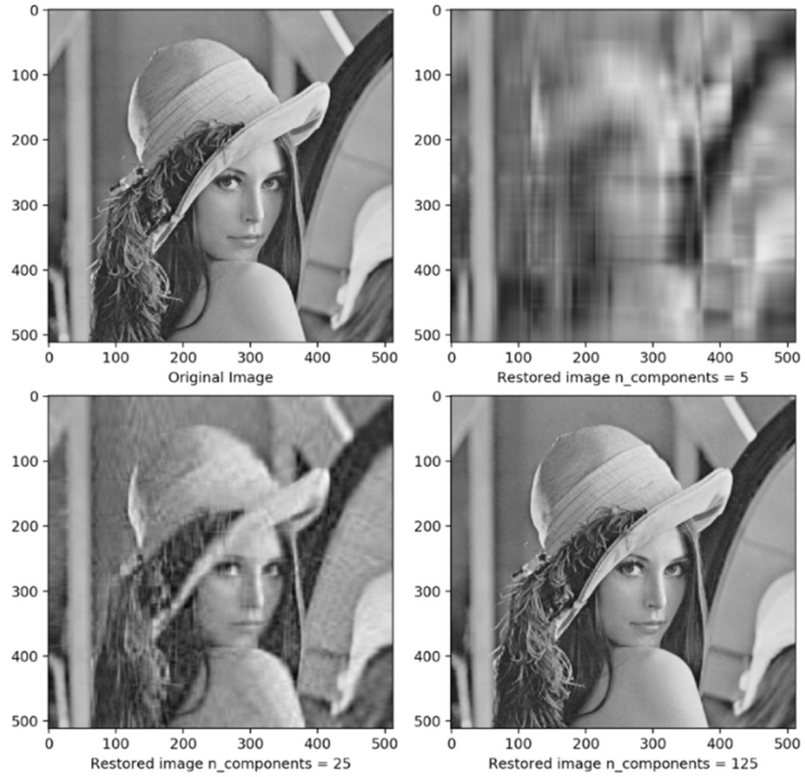
0
100
200
300
400
500
Original Image
Restored image n_components = 5
Restored image n_components = 25
Restored image n_components = 125

Figure ***7****. The image Lena in original state and restored on the 5, 25, 125 components*

5. Conclusion

The Method of Main Components is proposed as one of the methods of digital signal processing in a wide variety of fields, primarily communication and control systems, radio engineering and electronics, acoustics, television, measuring technology. The main areas of digital signal processing are sound and image processing (compression, synthesis, recognition, identification). So, in this article we touches upon the issue of of signal processing enhancement, oriented to the construction of high-speed algorithms and gives PCA algorithm in details in Python. As noted above, the results of PCA depend on the scaling of the variables. A scale-invariant form of PCA has been developed. The applicability of PCA is limited by certain assumptions made in its derivation. The other limitation is the mean-removal process before constructing the covariance matrix for PCA.

REFERENCES

1. ARCHAMBEA C., DELANNAY N., VERLEYSEN M.: Mixtures of robust probabilistic principal component analyzers, Elsevier, Neurocomputing, 71(2008)7-9, 1274-1282.
2. PEARSON K.: On Lines and Planes of Closest Fit to Systems of Points in Space (PDF). Philosophical Magazine. 2(1901)11, 559–572. doi:10.1080/14786440109462720.
3. HOTELLING H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, p. 24, pp. 417–441, and pp. 498–520, 1933.
4. HOTELLING H.: Relations between two sets of variates. Biometrika. 28: 321–377, 1936. doi:10.2307/2333955.
5. JOLLIFFE I.T.: Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4.
6. BOBER M., KUCHARSKI K., SKARBEK W.: Face Recognition by Fisher and Scatter Linear Discriminant Analysis, In: Petkov N., Westenberg M.A. (Eds.) Computer Analysis of Images and Patterns. CAIP 2003. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol 2756, pp. 638-645, 2003.
7. IATAN I.F.: The Fisher's Linear Discriminant. In: Borgelt C. et al. (Eds.) Combining Soft Computing and Statistical Methods in Data Analysis. Advances in Intelligent and Soft Computing, Springer, Berlin, Heidelberg, 2010,vol. 77, pp. 345-352.
8. JOLLIFFE I.T.: Principal Component Analysis, Springer Series in Statistics, Springer-Verlag New York, 2002, p. 488, ISBN:978-0-387-95442-4, [DOI 10.1007/b98835].
9. LI Y.Z., WANG Y.H.: PCA and algorithms analysis, Journal of Suzhou University, 1(2005), pp.32–36.
10. PETROV A., KHOROSHKO V., SCHERBAK L., PETROV A., ALEKSANDER M. University of Science and Technology Press, (2016) Reliability basics of information systems, ed. by Alexander Petrov; 246p., ISBN: 978-83-7464-859-2.
11. SCHOLKOPF B., BURGES J.C., SMOLA A.J., MULLER K.R.: Kernel Principal Component Analysis, In: Scholkopf B., Burges J.C., Smola A.J., Muller K.R., (Eds.), Advances in Kernel Methods – Support Vector Learning, MIT Press, Cambridge, MA, Chapter 20, pp. 327–352, 1998.
12. TANWAR S., RAMANI T., TYAGI S.: Dimensionality Reduction Using PCA and SVD in Big Data: A Comparative Case Study, Chapter in: Future Internet Technologies and Trends, Springer International Publishing, 2018, pp 116-125, ISBN:978-3-319-73711-9. [Online]. Available: https://doi.org/10.1007/978-3-319-73712-6_12; date accessed: 20.02.2018.
13. TIPPING M.E., BISHOP C.M.: Probabilistic Principal Component Analysis, Journal of the Royal Statistical Society. Series B (Statistical Methodology), 61(1999)3, 611-622.
14. YANG D., XU T., YANG R., LIU W.: Face Image Enhancement via Principal Component Analysis. In: Nicholson A., Li X. (Eds.) AI 2009: Advances in

Artificial Intelligence. AI 2009. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, (2009)5866, 190-198.

15. ZHANG Y., MA H.: The Electronic Countermeasures Optimization Based on PCA and Multiple Regression Analysis”. In: Tan T., Ruan Q., Chen X., Ma H., Wang L. (Eds.) Advances in Image and Graphics Technologies. IGTA 2013. Communications in Computer and Information Science, vol. 363, Springer, Berlin, Heidelberg, 2013.