Principal Components Analysis as enhancement Operator and Compression factor

KhaIiI Ibrahim Kadhim
College of Basic Education
University of AI- mustansiryah

Abstract

Principal components analysis (PCA) is effective at compressing information in multivariate data sets by computing orthogonal projections that maximize the amount of data variance. Unfortunately, information content in hyper spectral images does not always coincide with such projections. We propose an application of projection pursuit (pp), which seeks to find a set of projections that are "interesting" in the sense that they deviate from the Gaussian distribution assumption.

Once these projections are obtained, they can be used for image compression, segmentation, or enhancement for visual analysis. To find these projections, a two –step iterative process is followed where we first search for a projection that maximizes a projection index based on the information divergence of the projections estimated probability distribution from the Gaussian distribution and then reduce the rank by projections the data on to the subspace orthogonal to the previous projection . To calculate each projections, we use a simplified approach to maximizing the projection index, which does not require optimization algorithm. It searches for a solution by obtaining a set of candidate projections from the data and choosing the one with the highest projection index. The effectiveness of the method is demonstrated through simulated examples as well as data from the hyper spectral digital imagery collection experiment and the spatially enhanced broadband and array spectrograph system.

Introduction

A principle component can be defined as a linear combination of optimally-weighted observed variables. In order to understand the meaning of this definition it is necessary to first describe how subject scores on a principle component are computed. In the course of performing a principle component analysis, it is possible to calculate a score for each subject on a given principle component **[1]**. For example, in the preceding study, each subject would have score on two components. One score on the satisfaction with supervision component, and one score on the satisfaction with pay component **[2]**. The subject's actual score on the seven questionnaire items would be optimally weighted and then summed their scores on a given component.

Digital image

Digital images have two basic components: pixels, and print size. These components can be changed, either individually or in tandem, to achieve different affects. The basic digital component of an image is the pixel. This is the smallest piece of digital information in the image. Pixels are used to create the dots of color that make up the image. This concept should be immediately familiar from analog-word examples. When viewed on computer monitor, pixels always measure 1/72 of an inch, in both height and width (in other words, there are 72 pixels in an inch). The pixels can be stretched out to fit a specific print third component of digital images. The print size may be smaller or larger than its pixels height and width as viewed on a monitor **[3]**. When this happens, the output device stretches or compresses the dots of color to fit in the print size. The number of pixels in an inch of the print size determines the overall quality of the image, referred to as "resolution" these two factors normally have an inverse relationship to the another. Therefore, if the print size increases and the number of pixels stay the same, the pixels per inch (ppi) decreases meaning that resolution is lowered and vice versa. When this type of change is made, the number of pixels and therefore the file size is unchanged **[4]**. Dots are often used interchangeably with pixels in the expression of the resolution of a digital image (as dots per inch or "dpi") although this technically refers to resolution of a printed of the image. Scanners and bitmap image editors allow manipulation of these properties independently of one another-resolution can decrease or increase with the print size for example. This process is generally known as "resampling" a process which deletes or creates pixels from an image to achieve a desired effect. In some cases resampling is desirable, particularly in the case of decreasing resolution and file size. On the order hand, increasing is almost never desirable-When as increase in resolution occurs, the computer has to interpolate the new pixels based on the surrounding pixels properties should be, the resulting image appears fuzzy or slightly out of focus. For example, resampling a 300 ppi JPEG down to 72 ppi while keeping the file size the same would result in an image that will look just as sharp on the monitor screen as it would in print, but will have a much smaller file size for faster loading. On the other hand, if an image exists at 72 ppi the software Would have to use interpolation to create an image at a higher resolution resulting in poor image quality. When an image is not resampled when it is resized, the variables of resolution and print size have an inverse relationship to each other while pixel dimensions remain the same. So if a slide that measures only 1×1.5 inches is at very high resolution then choosing not to resample when the image is resized will result in an image of the same file size with a lower ppi but a larger print size. To avoid this, digital images that need to be enlarging for a specific purpose should be captured at a very high resolution to minimize the later decrease in resolution.

When resizing is done, the print size and resolution should change inversely to each other, keeping the size the same. In order word, file size is the key factor to consider when enlarging an image. When image are scanned without a known need for resizing, they should be scanned to meet the minimum requirements for resolution.

(Mathematical Description)

Matrix Algebra

I will be looking at eigenvectors and eigenvalues of a given matrix and assume a basic knowledge **[5],[6].**

Figure (1): Example of one non-eigenvector and one eigenvector

$$2 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

Figure (2):Example of how a scaled eigenvector is still and eigenvector

Eigenvectors:

In the first example, the resulting vector is not an integer multiple of the original vector, whereas in the second example, the example is exactly 4 times the vector we began with. The vector in 2dimensional spaces. The vector $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ represents an arrow pointing from the origin, (0, 0), to the point (3. 2). The other matrix, the square one, can be thought of as a transformation matrix. If we multiply this matrix on the left of a vector, the answer is another vector that is transformed from its original position **[7]**. It is the nature of the transformation that the eigenvectors arise from. Imagine a transformation matrix that, when multiplied on the left, reflected vector in the line y = x. Then we can see that if there were a vector that lay on the line y = x, its reflection it itself. This vector would be an eigenvector of that transformation matrix. Eigenvectors can only be found for square matrices and not every square matrix has eigenvectors. And, given an $n \times n$ matrix that does have eigenvectors, there are n of then. Given a 3×3 matrix, there are 3 eigenvectors. Another property of eigenvectors is that event if I scale the vector by some amount before I multiply it, I still get the same multiple of it as a result, as in figure 2. This is because if you are doing making it longer, not changing its direction. Lastly, all the eigenvectors of a matrix are perpendicular, i.e.at right angles to each other, no matter how many dimensions you have. By the way, another word for perpendicular, in maths talk, is orthogonal. This is important because it means that you can express the data in terms of these of these perpendicular eigenvectors, instead of expressing them in term of the x and y axes **[8]**.Another important thing to know is that when mathematicians find eigenvectors, they like to find the eigenvectors whose length is exactly one, because the length of a vector doesn't affect whether it's an eigenvector or not, whereas the direction does. So in order to keep eigenvector standard, whenever we find an eigenvector we usually scale it to make it have a length of l, so that all eigenvector have the same length. For example $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ is an eigenvector, and the length of that vector is $\sqrt{(3^2 + 2^2)} = \sqrt{13^2}$ So we divided the original vector by this much to make it have a length of l.

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix} \div \sqrt{13} = \begin{pmatrix} \frac{3}{\sqrt{13}} \\ \frac{2}{\sqrt{13}} \end{pmatrix}$$

The usual way to find the eigenvector is by some complicated iterative method which is beyond the scope of this tutorial. If we ever need to find the eigenvector of a matrix in a program, just find a maths library that does it all for us.

Eigenvalues:

Eigenvalues are closely related to eigenvectors, in fact, we saw an eigenvalue in figure1. In both those examples, amount by which the original vector was scaled after multiplication by the square matrix was the same. In that example the value was 4.4 is the eigenvalue associated with that eigenvector **[9]**. No matter what multiple of the eigenvector we took before we multiplied it by the square matrix, we would always get 4 times the scaled vector as our result as in figure 2. So we can see that eigenvector and eigenvector always come in pairs. When we get a fancy programming library to calculate our eigenvector for us, we usually get the eigenvector as well.

Exercises:

For the following square matrix:

$$\begin{pmatrix} 3 & 0 & 1 \\ -4 & 1 & 2 \\ -6 & 0 & -2 \end{pmatrix}$$

Decide which, if any of the following vectors are eigenvector of that matrix and give the corresponding eigenvalue.

$$\begin{pmatrix} 2 \\ 2 \\ -1 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \\ 3 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$$

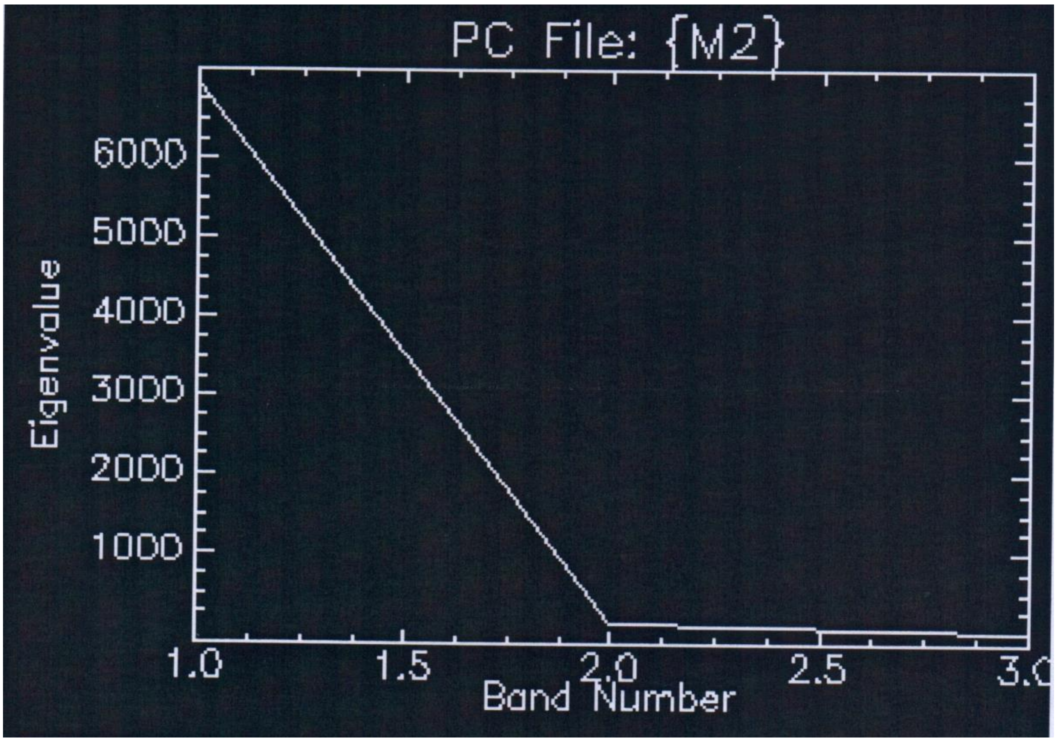

Figure (3): Eigen value

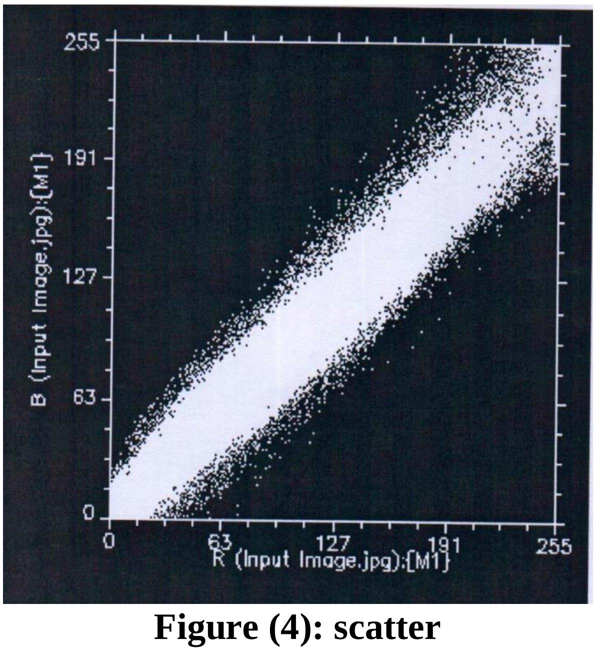

Figure (4): scatter

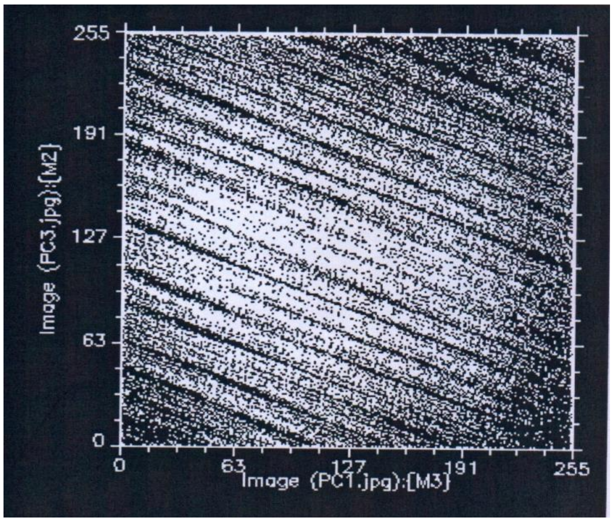

Figure (5): uncorelleted

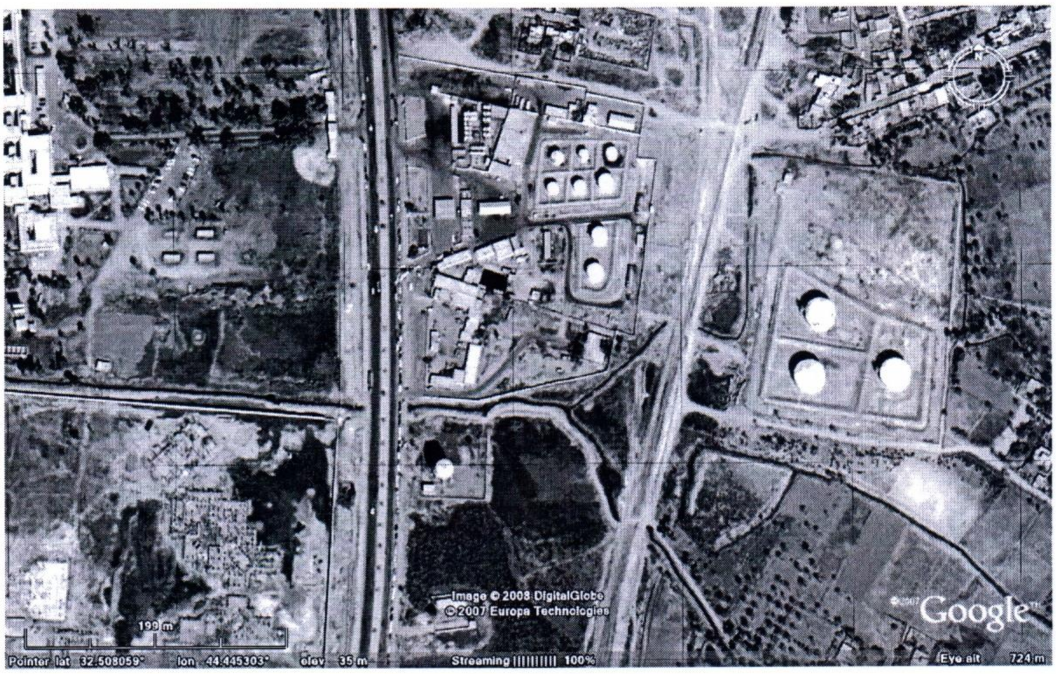

Figure (6): PC1

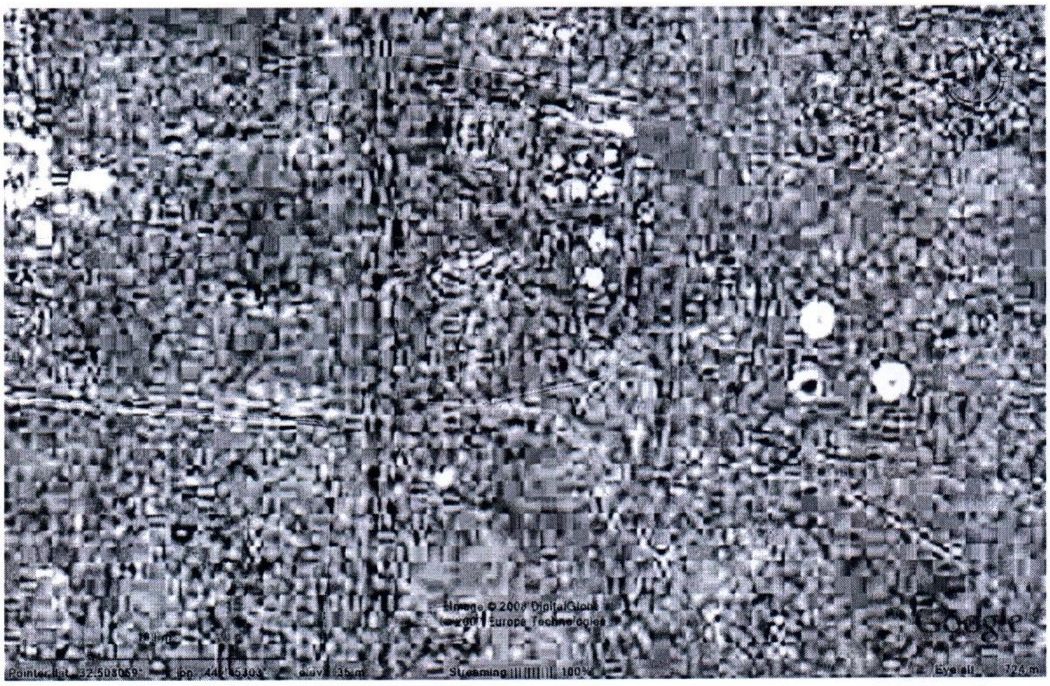

Figure (7): PC3

Figure (8): PC2

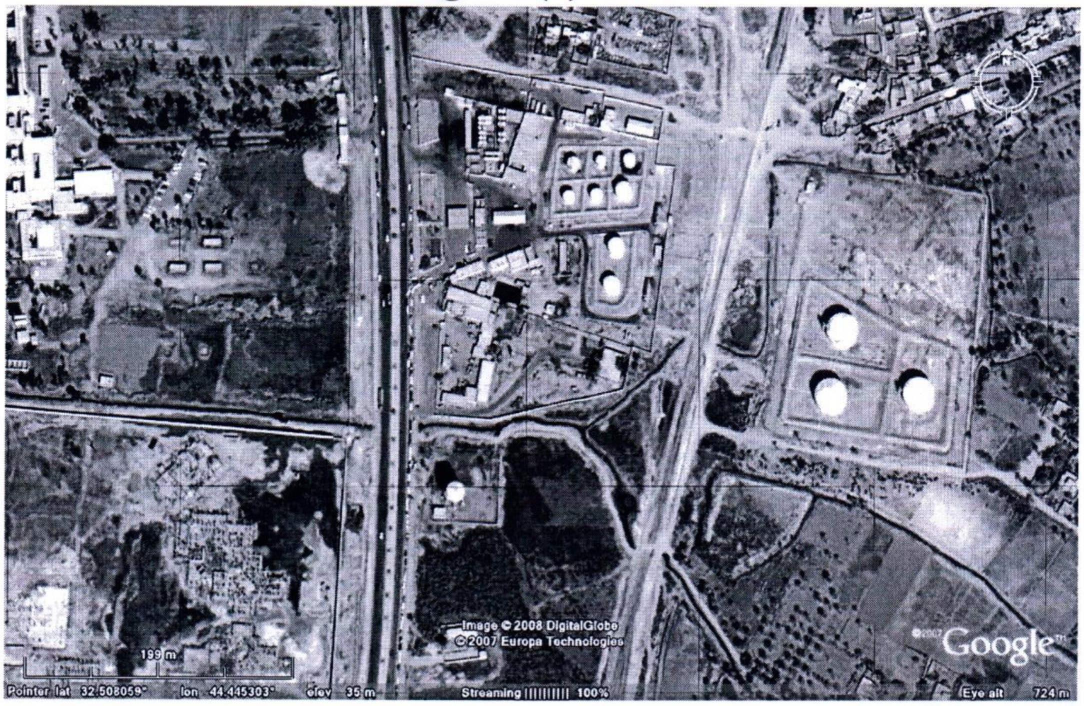

Figure (9): Enhance Image

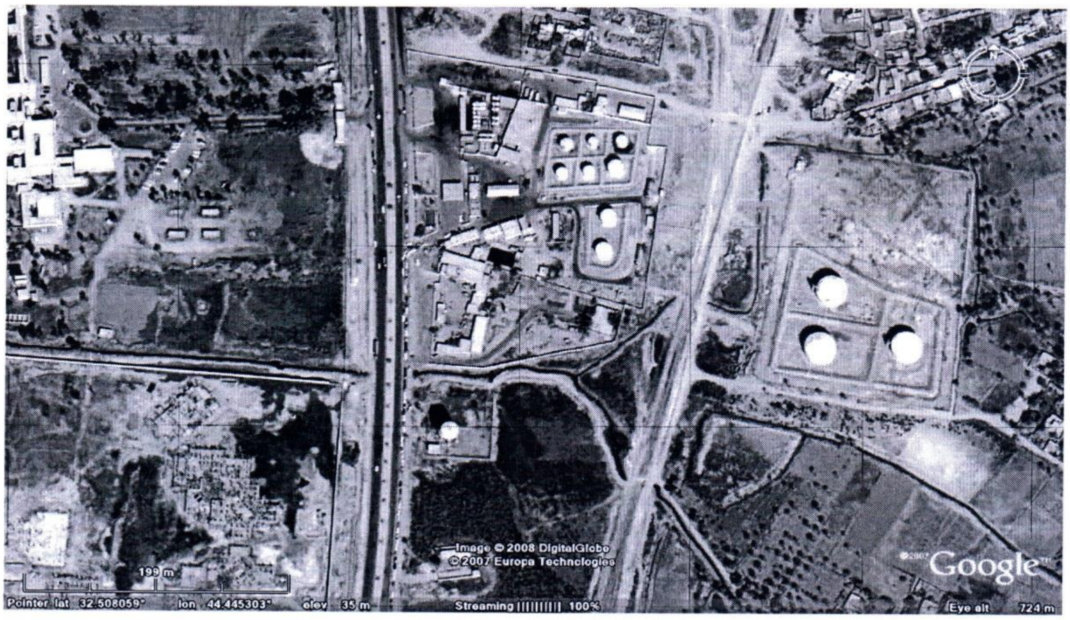

Figure (10): Input Image

Results Analysis:

a. PCA as enhancement operator
b. PCA as compression factor

Size of origin image = $1024 \times 659 \times 3 = 2024448$ bit

New size after compression $=1024 \times 659 + 16 = 674832$

Compression factor =2024448/ 674832= 2.99

Conclusion:

Principal component analysis is a powerful tool reducing a number of observed variables into a smaller number of artificial variables that account for most of the variance in the data set. It is particularly useful when you need a data reduction procedure that makes no assumption concerning an underlying causal structure that is responsible for covariation in the data. When it is possible to postulate the existence of such an under lying casual structure, it may be more appropriate to analysis the using exploratory factor analysis. Both principal component analysis and factor analysis are that used to construct multiple-item scales from the items that constitute questionnaires.

References:

(1) C-l chang and Q Du, "interference and noise- adjusted principal components analysis" IEEE Trans. Geosci Remote sensing, vol. 37,p. 2387, sept 1999.

(2) C-l chang. T-l. e .sun, and M.L.G Althouse, "unsupervised interference rejection approach to target detection and classification for hyper spectral imagery," opt. Eng., vol. 37, no.3, pp. 735-743

(3) Gonzalez, Rafael, c, woods. Richard E. Digital image processing, 3 rd Edition. Pearson prentice Hall. Pp. 577. 15BN013168728. (2008).

(4) Anderson, Richard. The universal photographic Digital imaging Guidelines. Ed. Michael Stewart 2006 vers. 2.0. Up DIG. Universal photographic Digital imaging Guide lines. P.8- Accessed on October 12, 2007.

(5) Y. e, J Generalized low rank approximations of matrices. Machine learning, 61, 167-191. (2005).

(6) El K A R oul, N. Tacy- widow limit for the largest eigenvalue of a large class of complex sample covariance matrices, Ann. Probab. 35663-714. MR 2308592 2007.

(7) JOHNSTONE, I. M. on the distribution of the largest eigen value in principal component analysis. Ann. Statist. 29295-327. MR 1863961. (2001).

(8) BALK, J. and silver STELN, J.W. Eigen values of large sample covariance matrices of spiked population models. J. Multivariate Aual. 97 1382-1408

(9) Y. Zhang and J. weng, "convergence analysis of complementary candid incremental principal component analysis" Technical Report MSU- CSE-01-23, Dept. of computer science and Eng. Michigan state univ, East LANSING. Aug. 2001.