International Journal of Statistics and Applied Mathematics 2020; 5(1): 39-50

International Journal of Statistics and Applied Mathematics

ISSN: 2456-1452
Maths 2020; 5(1): 39-50

www.mathsjournal.com
Received: 14-11-2019
Accepted: 18-12-2019

Orumie
Department of Mathematics/Statistics, University of Port Harcourt, Nigeria

Ukamaka Cynthia
Department of Mathematics/Statistics, University of Port Harcourt, Nigeria

Corresponding Author:
Orumie
Department of Mathematics/Statistics, University of Port Harcourt, Nigeria

Singular value decomposition and principal component analysis: Misconceptions and disparities

Orumie and Ukamaka Cynthia

Abstract
The application of singular value decomposition to perform principal component analysis is becoming increasingly evident in certain areas such as machine learning. The objective of this study is to differentiate these methods by clearly elucidating the disparities that exist theoretically and empirically. The study discovered that the PCA method is a product of the Eigen decomposition applied to the covariance/correlation matrix while SVD is achieved via a direct application to the dataset matrix after normalization. This was verified by carrying out a comparative analysis of both methods on a dataset consisting of standing heights and physical attributes (one dependent variable and seven independent variables) of 33 females applying for police officer positions in Akwa ibom state. The analysis was done using a MATLAB software program. The final results from the outputs obtained for both methods were the same. The study concluded that both methods are products of matrix decompositions and can be used to achieve the purpose of data reduction but the difference lies in how they are being applied.

Keywords: principal component analysis, singular value decomposition, data reduction, normalization, covariance/correlation matrix

1. Introduction

Mathematical methods tend to evolve on a daily basis. Due to the high demand of mathematical applications to technological innovations and virtually all aspect of human endeavor, the frontiers of mathematics have been stretched beyond expectations. The aforesaid is largely evident in linear algebra where applications from this area has been quite erroneous. Such ground-breaking outcomes of linear algebra are the singular value decomposition and principal component analysis.

This research presents misconceptions and disparities on singular value decomposition and principal component analysis, and it is organized as follows:

Introduction, development of singular value decomposition and principal component analysis are presented in Section Two. Comparative analysis of both singular value decomposition and principal component analysis with their in-depth structures are presented in Section Three, whereas results and discussions are presented in Section Four. Section five presents the concluding remark.

2. Development of singular value decomposition and principal component analysis

Singular value decomposition can be traced back to 1873 where it was first captured in a publication on bilinear forms by a mathematician called Eugenio Beltrami. A year after Beltrami's publication, Camille Jordan made another related publication on SVD but his accentuated the concept and made it more appealing to students for use. These two authors are considered co-discoverers of the singular value decomposition method.

According to Gibiansky (2013) [5], one of the best results of linear algebra is the decomposition method called singular value decomposition method. It is considered the highlight of linear algebra as it amplifies the inner structure of a matrix and produces the orthonormal bases for the four fundamental subspaces (column, left null, row and null spaces). It can be simply defined as the factorization of a matrix into the product of three matrices. That is, the SVD of a matrix B (m×n) is given as,

$$B = U\Sigma V$$

Where U is an m×m unitary matrix?
V Is an n×n unitary matrix
Σ Is a diagonal matrix with non-negative real numbers on the diagonal.

The SVD generalizes the unitary transformation of a vector space which diagonalizes a linear transformation to the case of non-square matrices. The unitary transformations are special, because they preserve the l2-norm between the differences of any two vectors in a space they act upon. Unitary transformations preserve the l2-norm 'length' of any vector in the space. Unitary transforms preserve the eigenvalues of a linear transformation and preserve the kernel and the co-domain of the linear transformations they act upon.

Over time, the SVD method evolved into a robust technique with a wide-range of applications. From providing solutions to least square problems; numerical determination of the rank of a matrix, solutions of over determined system to principal component analysis in Statistics, the applications of SVD has become inexhaustible. However, the focus of interest of this paper is on the principal component analysis and how it distinguishes from SVD.

The principal component analysis is a multivariate analysis tool used to reduce a high multidimensional dataset to a low multidimensional dataset without much loss of information. PCA was first applied by Karl Pearson in 1901 though it was further developed by Harold Hotelling in 1930s.

Geometrically, PCA is achieved by a rotation of the axes of the original coordinate system to new orthogonal axes, called principal axes, such that the new axes coincide with directions of the maximum variation of the original observations. The first component, which would lie in the direction of the maximum direction, accounts for the most variation while the second component, which lies in an orthogonal direction to the first, accounts for the second most variation of the data set and so on. With this kind of re-arrangement of the dataset, a new set of variables can be obtained which makes it easier to select fewer set of variables for the data analysis without much loss of information.

Basically, principal component analysis is performed by the Eigen decomposition of a data covariance or correlation matrix. It is at this point that the connection between these methods can be established given the fact both methods employ some form of matrix decomposition.

In an effort to simply lay the foundation for the extensions and generalization of SVD with respect to PCA, Cadima & Jollife, (2009) [2] described PCA as a singular value decomposition of a column-centered data matrix. Some applications do not want its data to be processed beforehand as the un-centered data is subjected to SVD. This is termed un-centered PCA. The paper considers the existing relationship between the results from SVD and PCA.

Additionally, classical PCA is hinged on covariance/correlation decomposition matrix (Geladi & Kowalski, 1986) [4] by eigen-value (spectral) decomposition of real data matrices using SVD (Wall, Dyck, & Brettin, 2001) [8].

Although compared with EVD, SVD is a more precise, robust and reliable methodology with no need to compute the input correlation/covariance matrix according to Will (1999) [9]. Classical PCA is not useful when the figure of variables is bigger than the observed number on each variable. In microarray data analysis, Lim, (2013) encountered this situation commonly when genes are accepted as the variables. In such instances, it is pertinent to use PCA using SVD (Deshunk & Purohit, 2007) [3]. For applying PCA using SVD, the DNA microarray data was in use for the small round blue cell tumors (SRBCT) of childhood by (Khan, *et al*., 2001) [7].

However, PCA and SVD have discovered wide-ranging tools. Here we define several studies that may infer potential strategies to employ for applications of these methods.

A study by Aremu, Adebayo, Ariyo, & Adewale (2007) [1] jointly applied PCA, single linkage cluster analysis and canonical analysis in identifying characters responsible for variation in Cowpea. These methods were effective in showing most distant accessions as having widest variation and possible choice of parent stocks in hybridization.

Anand, *et al*., (2018) reviewed the singular value decomposition (SVD) framework and its use in quantifying and discerning vertical information in green house from column integrated absorption measurements. Bayesian optimal estimation (OE) assumes a prior distribution in order to regularize the inversion problem, while the SVD approach identifies principal components that can be retrieved from the measurement without explicitly specifying a prior mean and prior covariance.

Anand *et al*, (2018) made a vital connection between the SVD method and the pseudo-inverse which intuitively makes it easy to understand.

Zheng, Nie, & Ding (2018) [10] presented a stretched insight to the application of SVD by regularization. This regularized SVD, which is simply abbreviated as RSVD, is considered computationally efficient as it is non-convex and has a closed form global solution. It was applied to a recommender system and the result showed that RSVD was more efficient compared to SVD.

3. Comprehensive analysis

A comparative analysis was conducted on a dataset consisting of standing heights and physical attributes of 33 females applying for police officer positions. Seven (7) independent variables and 1 (one) dependent variable were collected and they are given as follows:

$Y \equiv$ Standing height
$X_1 \equiv$ Sitting Height (cm)
$X_2 \equiv$ Upper Arm Length (cm)
$X_3 \equiv$ Forearm Length (cm)
$X_4 \equiv$ Hand Length (cm)
$X_5 \equiv$ Upper Leg Length (cm)

$X_6 \equiv$ Lower Leg Length (cm)
$X_7 \equiv$ Foot Length (inches)

Table 1: Standing heights and Physical Stature Attributes among 33 Female Police Applicants in Akwa Ibom State

ID	Y	X1	X2	X3	X4	X5	X6	X7	X8	X9
1.	165.8	88.7	31.8	28.1	187	40.3	38.9	6.7	88.4	96.5
2.	169.8	90.0	32.4	29.1	183	43.3	42.7	6.4	89.8	98.6
3.	170.7	87.7	33.6	29.5	20.7	43.7	41.1	7.2	87.8	94.1
4.	170.9	87.1	31.0	28.2	18.6	43.7	40.6	6.7	91.0	92.9
5.	157.5	81.3	32.1	27.3	17.5	38.1	39.6	6.6	85.0	103.9
6.	165.9	88.2	31.8	29.0	18.6	42.0	40.6	6.5	91.2	96.7
7.	158.7	86.1	30.6	27.8	18.4	40.0	37.0	5.9	90.8	92.5
8.	166.0	88.7	30.2	26.9	17.5	41.6	39.0	5.9	89.1	93.8
9.	158.7	83.7	31.1	27.1	18.1	38.9	37.5	6.1	87.1	96.4
10.	161.5	81.2	32.3	27.8	19.1	42.8	40.1	6.2	86.1	93.7
11.	167.3	88.6	34.8	27.3	18.3	43.1	41.8	7.3	78.4	97.0
12.	167.4	83.2	34.3	30.1	19.2	43.4	42.2	6.8	87.8	97.2
13.	159.2	81.5	31.0	27.3	17.5	39.8	39.6	4.9	88.1	99.5
14.	170.0	87.9	34.2	30.9	19.4	43.1	43.7	6.3	90.4	101.4
15.	166.3	88.3	30.6	28.8	183	41.8	41.0	5.9	94.1	98.1
16.	169.0	85.6	32.6	28.8	19.1	42.7	42.0	6.0	88.3	98.4
17.	156.2	81.6	31.0	25.6	17.0	44.2	39.0	5.1	82.6	88.2
18.	159.6	86.6	32.7	25.4	17.7	42.0	37.5	5.0	77.7	89.3
19.	155.0	82.0	30.3	26.6	17.3	37.9	36.1	5.2	87.8	95.3
20.	161.1	84.1	29.5	26.6	17.8	38.6	38.2	5.9	90.2	99.0
21.	170.3	88.1	34.0	29.3	18.2	43.2	41.4	5.9	86.2	95.8
22.	167.8	83.9	32.5	28.6	20.2	43.3	42.9	7.2	88.0	99.1
23.	163.1	88.1	31.7	26.9	18.1	40.1	39.0	5.9	84.9	97.3
24.	165.8	87.0	33.2	26.3	19.5	43.2	40.7	5.9	79.2	94.2
25.	175.4	89.6	35.2	30.1	19.1	45.1	44.5	6.3	85.5	98.7
26.	159.8	85.6	31.5	27.1	19.2	42.3	39.0	5.7	86.0	92.2
27.	166.0	84.9	30.5	28.1	17.8	41.2	43.0	6.1	92.1	104.4
28.	161.2	84.1	32.8	29.2	18.4	42.6	41.1	5.9	89.0	96.5
29.	160.4	84.3	30.5	27.8	16.8	41.0	39.8	6.0	91.1	97.1

In clearly delineating the disparities that exist between the PCA and SVD methods, it is imperative to go over the diagonalization of matrices and how the deficiencies in the eigen decomposition of matrices is being augmented by SVD.
A matrix A is diagonalizable if it can be written as a product such as

$$A = EDE^{-1} \qquad \dots (1)$$

Where
E is a matrix that can be inverted and D is a diagonal matrix. Seemingly, since E is invertible then it must be square. It follows that a non-square matrix cannot be diagonalized as this idea seems unreasonable in this context.
If equation (1) holds then $AE = ED$, then E is defined through its columns b_i and D via its diagonal entries;
$E = (a_1, a_2 \dots a_n)$

$$D = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

Due to the way matrix multiplication works, each non-zero element in D simply picks out a different column in E and scales it. Hence,

$$ED = (\lambda_1 a_1, \lambda_2 a_2 \dots \lambda_n a_n) \qquad \dots (2)$$

What we can draw from this is that since $AE = ED$, and we can consider the columns of E separately from each other, the columns of Emust be the eigenvectors of A and the values on the diagonal must be eigenvalues of A. (Since $E$is invertible, the space spanned by the eigenvectors must be the entire vector space, so the eigenvectors actually form a basis. Not all matrices are diagonalizable, since clearly not all matrices have eigenvectors that form a basis!)
Given an $m \times n$ matrix X for the dataset where the samples are the n columns (e.g. observations) and the m rows are variables, the objective is to transform the matrix X linearly to another matrix Y of similar dimension $m \times n$. Hence, for some $m \times m$ matrix P, the PCA is given as

$$Y = PX \qquad \dots (3)$$

The matrix P is a transformation matrix (the Eigen vectors of the data matrix) which is expected to change the basis from X to Y. Geometrically, P is a rotation and a stretch which moves X to Y.

In some sense, the singular value decomposition is essentially diagonalization in a more general sense. The singular value decomposition plays a similar role to diagonalization, but it fixes the flaws we just talked about; namely, the SVD applies to matrices of any shape. Not only that, but the SVD applies to all matrices, which makes it much more generally applicable and useful than diagonalization!

For the SVD we begin with an arbitrary real $m \times n$ matrix A. As we shall see, there are orthogonal matrices U and V and a diagonal matrix, this time denoted Σ, such that $A = U\Sigma V^T$. In this case, U is $m \times m$ and V is $n \times n$, so that Σ is rectangular with the same dimensions as A. The diagonal entries of Σ, that is the $\Sigma_{ii} = \sigma_i$, can be arranged to be nonnegative and in order of decreasing magnitude. The positive ones are called the singular values of A. The columns of U and V are called left and right singular vectors, for A.

The analogy between the EVD for a symmetric matrix and SVD for an arbitrary matrix can be extended a little by thinking of matrices as linear transformations. For a symmetric matrix A, the transformation takes $\mathcal{R}^n$ to itself and the columns of V define an especially nice basis. When vectors are expressed relative to this basis, we see that the transformation simply dilates some components and contracts others, according to the magnitudes of the eigenvalues (with a reflection through the origin tossed in for negative eigenvalues). Moreover, the basis is orthonormal, which is the best kind of basis to have.

Now let's look at the SVD for an $m \times n$ matrix A. Here the transformation takes $\mathcal{R}^n$ to a different space, $\mathcal{R}^m$, so it is reasonable to ask for a natural basis for each of domain and range. The columns of V and U provide these bases. When they are used to represent vectors in the domain and range of the transformation, the nature of the transformation again becomes transparent: it simply dilates some components and contracts others, according to the magnitudes of the singular values, and possibly discards components or appends zeros as needed to account for a change in dimension. From this perspective, the SVD tells us how to choose orthonormal bases so that the transformation is represented by a matrix with the simplest possible form, that is, diagonal.

How do we choose the bases $\{v_1, v_2,\cdots,v_n\}$ and $\{u_1,u_2,\cdots,u_m\}$ for the domain and range? There is no difficulty in obtaining a diagonal representation. For that, we need only $Av_i = \sigma_i\, u_i$, which is easily arranged. Select an orthonormal basis A^TA for $\mathcal{R}^n$ so that the first k elements span the row space of A and the remaining n-k elements span the null space of A, where k is the rank of A. Then for $1 \leq i \leq k$ define u_i to be a unit vector parallel to Av_i, and extend this to a basis for $\mathcal{R}^m$. Relative to these bases, A will have a diagonal representation. But in general, although the v's are orthogonal, there is no reason to expect the u's to be. The possibility of choosing the v-basis so that its orthogonality is preserved under A is the key point. We show next that the EVD of the n×n symmetric matrix A^TA provides just such a basis, namely, the eigenvectors of A^TA.

Let $A^TA = VDV^T$, with the diagonal entries λ_i of D arranged in non-increasing order, and let the columns of V (which are eigenvectors of A^TA) be the orthonormal basis $\{v_1,v_2,\cdots,v_n\}$.

Then

$$Av_i.Av_j = (Av_i)^T(Av_j) = v_i^T A^T A v_j = v_i^T(\lambda_j v_j) = \lambda_j v_i.v_j \quad \ldots (4)$$

So the image set $\{Av_1, Av_2,\cdots,Av_n\}$ is orthogonal, and the nonzero vectors in this set form a basis for the range of A. Thus, the eigenvectors of A^TA and their images under A provide orthogonal bases allowing A to be expressed in a diagonal form.

To complete the construction, we normalize the vectors Av_i. The eigenvalues of A^TA again appear in this step. Taking $i = j$ in the calculation above gives $|Av_i|^2 = \lambda_i$, which means $\lambda_i \geq 0$. Since these eigenvalues were assumed to be arranged in non-increasing order, we conclude that $\lambda_1 \geq \lambda_2 \geq\cdots\geq\lambda_k \geq 0$ and, since the rank of A equals k, $\lambda_i = 0$ for $i > k$. The orthonormal basis for the range is therefore defined by

$$u_i = \frac{Av_i}{|Av_i|} = \frac{1}{\sqrt{\lambda_i}} Av_i \; 1 \leq i \leq k \quad \ldots (5)$$

If $k < m$, we extend this to an orthonormal basis for $\mathcal{R}^m$.

This completes the construction of the desired orthonormal bases for $\mathcal{R}^n$ and $\mathcal{R}^m$. Setting $\sigma_i = \sqrt{\lambda_i}$ we have $Av_i = \sigma_i u_i$ for all $i \leq k$. Assembling the v_i as the columns of a matrix V and the u_i to form U, this shows that $AV = U\Sigma$, where Σ has the same dimensions as A, has the entries σi along the main diagonal, and has all other entries equal to zero. Hence, $A = U\Sigma V^T$, which is the singular value decomposition of A.

In summary, an $m \times n$ real matrix A can be expressed as the product $U\Sigma V^T$, where V and U are orthogonal matrices and Σ is a diagonal matrix, as follows. The matrix V is obtained from the diagonal factorization $A^TA = VDV^T$, in which the diagonal entries of D appear in non-increasing order; the columns of U come from normalizing the non-vanishing images under A of the columns of V, and extending (if necessary) to an orthonormal basis for $\mathcal{R}^m$; the nonzero entries of Σ are the respective square roots of corresponding diagonal entries of D.

The preceding construction demonstrates that the SVD exists, and gives some idea of what it tells about a matrix. There are a number of additional algebraic and geometric insights about the SVD that are derived with equal ease.

The SVD encapsulates the most appropriate bases for the domain and range of the linear transformation defined by the matrix A. There is a beautiful relationship between these bases and the four fundamental subspaces associated with A: the range and null space, and their orthogonal complements

To relate SVD to PCA, there exists a theorem from linear algebra which stipulates that the non-zero singular variables of **A** are the nonzero eigenvalues square roots of AA^T or A^TA. The first assertion for the A^TA case is proven in the succeeding way:

$$A^TA = (U\Sigma V^T)^T(U\Sigma V^T) = (V\Sigma^T U^T)(U\Sigma V^T) = V(\Sigma^T\Sigma)V^T \quad \ldots (6)$$

We note that A^TA is same as $\Sigma^T\Sigma$, and hence it has similar eigenvalues. Since $\Sigma^T\Sigma$ is a $(m \times m)$ square, diagonal matrix, the eigenvalues are hence the entries done diagonally, which are the singular values squares. Note that the non-zero eigenvalues of individual covariance matrices, AA^T and A^TA are very identical.

It is imperative to note that an eigenvalue matrix decomposition, A^TA has been carried out. Certainly, since A^TA is symmetric which is an orthogonal diagonalisation and hence the eigenvectors of A^TA are the V columns. This is pertinent in making the practical linking between the PCA and SVD of the X matrix, which comes up next.
Recalling the initial $m \times n$ data matrix, X, let us describe a new $n \times m$ matrix, Z:

$$Z = \frac{1}{\sqrt{n-1}} X^T \quad \dots (7)$$

Recalling that since the rows of X on m contained the samples of n data, we removed the average of the row from individual entries to ensure no mean existed across the rows. Hence, the new Z matrix, has columns without mean. Consider establishing the $m \times m$ matrix, Z^TZ:

$$Z^TZ = \left(\frac{1}{\sqrt{n-1}} X^T\right)^T \left(\frac{1}{\sqrt{n-1}} X^T\right)$$

$$= \frac{1}{n-1} XX^T \quad \dots (8)$$

$$i.e. Z^TZ = \Sigma_x$$

We observe that describing Z in this pattern ensures that Z^TZ is similar to the X, Σ_x is the covariance matrix. From the foregoing, the main components of X (subject to be identified) are the Σ_x eigenvectors. Thus, if we carry out an individual value matrix decomposition of Z^TZ, the main components will be the orthogonal matrix, V columns.
The final step is to link the SVD of Z^TZ back to the change of basis signified by equation (3):
Y = PX

We like to show the original data onto the principal component described direction. Having the relative $V = P^T$, this is basically:
$Y = V^TX$

If recovering the original data is what we wish to do, we basically compute (employing orthogonality of V):

$$X = VY \quad \dots (9)$$

4. Results and Discussion

A comparative analysis was carried out on data presented in Table 1. The MATLAB software program was used to run the following analysis. The outputs are presented as follows,

```
X= xlsread ('PROJECT.xlsx','Sheet 2', 'A1:G33')
% Principal component analysis of variables X1, X2, X3, X4, X5, X6, X7
% let the deviation of the data be given as X
X =
3.1333   -0.5667 0.0758 0.1303 -2.0152 -1.7030 0.6727
4.4333   0.0333 1.0758 -0.2697 0.9848 2.0970 0.3727
2.1333   1.2333 1.4758 2.1303 1.3848 0.4970 1.1727
1.5333   -1.3667 0.1758 0.0303 1.3848 -0.0030 0.6727
-4.2667  -0.2667 -0.7242 -1.0697 -4.2152 -1.0030 0.5727
2.6333   -0.5667 0.9758 0.0303 -0.3152 -0.0030 0.4727
0.5333   -1.7667 -0.2242 -0.1697 -2.3152 -3.6030 -0.1273
3.1333   -2.1667 -1.1242 -1.0697 -0.7152 -1.6030 -0.1273
-1.8667 -1.2667 -0.9242 -0.4697 -3.4152 -3.1030 0.0727
-4.3667 -0.0667 -0.2242 0.5303 0.4848 -0.5030 0.1727
3.0333 2.4333 -0.7242 -0.2697 0.7848 1.1970 1.2727
-2.3667 1.9333 2.0758 0.6303 1.0848 1.5970 0.7727
-4.0667 -1.3667 -0.7242 -1.0697 -2.5152 -1.0030 -1.1273
2.3333 1.8333 2.8758 0.8303 0.7848 3.0970 0.2727
2.7333 -1.7667 0.7758 -0.2697 -0.5152 0.3970 -0.1273
0.0333 0.2333 0.7758 0.5303 0.3848 1.3970 -0.0273
-3.9667 -1.3667 -2.4242 -1.5697 1.8848 -1.6030 -0.9273
1.0333 0.3333 -2.6242 -0.8697 -0.3152 -3.1030 -1.0273
-3.5667 -2.0667 -1.4242 -1.2697 -4.4152 -4.5030 -0.8273
-1.4667 -2.8667 -1.4242 -0.7697 -3.7152 -2.4030 -0.1273
2.5333 1.6333 1.2758 -0.3697 0.8848 0.7970 -0.1273
-1.6667 0.1333 0.5758 1.6303 0.9848 2.2970 1.1727
2.5333 -0.6667 -1.1242 -0.4697 -2.2152 -1.6030 -0.1273
1.4333 0.8333 -1.7242 0.9303 0.8848 0.0970 -0.1273
4.0333 2.8333 2.0758 0.5303 2.7848 3.8970 0.2727
0.0333 -0.8667 -0.9242 0.6303 -0.0152 -1.6030 -0.3273
-0.6667 -1.8667 0.0758 -0.7697 -1.1152 2.3970 0.0727
-1.4667 0.4333 1.1758 -0.1697 0.2848 0.4970 -0.1273
-1.2667 -1.8667 -0.2242 -1.7697 -1.3152 -0.8030 -0.0273
-0.5667 2.6333 -0.2242 0.4303 4.8848 1.7970 -1.0273
-2.9667 3.8333 0.5758 1.6303 2.6848 1.6970 -0.4273
-0.5667 1.2333 -0.9242 1.2303 3.6848 0.9970 -0.4273
```

-2.1667 1.1333 1.6758 0.8303 2.8848 3.3970 -0.8273

% transposing the data matrix of X to have the rows representing the variables and the columns representing the subjects
X'
Ans =
Columns 1 through 9
 3.1333 4.4333 2.1333 1.5333 -4.2667 2.6333 0.5333 3.1333 -1.8667
 -0.5667 0.0333 1.2333 -1.3667 -0.2667 -0.5667 -1.7667 -2.1667 -1.2667
 0.0758 1.0758 1.4758 0.1758 -0.7242 0.9758 -0.2242 -1.1242 -0.9242
 0.1303 -0.2697 2.1303 0.0303 -1.0697 0.0303 -0.1697 -1.0697 -0.4697
 -2.0152 0.9848 1.3848 1.3848 -4.2152 -0.3152 -2.3152 -0.7152 -3.4152
 -1.7030 2.0970 0.4970 -0.0030 -1.0030 -0.0030 -3.6030 -1.6030 -3.1030
 0.6727 0.3727 1.1727 0.6727 0.5727 0.4727 -0.1273 -0.1273 0.0727

Columns 10 through 18
 -4.3667 3.0333 -2.3667 -4.0667 2.3333 2.7333 0.0333 -3.9667 1.0333
 -0.0667 2.4333 1.9333 -1.3667 1.8333 -1.7667 0.2333 -1.3667 0.3333
 -0.2242 -0.7242 2.0758 -0.7242 2.8758 0.7758 0.7758 -2.4242 -2.6242
 0.5303 -0.2697 0.6303 -1.0697 0.8303 -0.2697 0.5303 -1.5697 -0.8697
 0.4848 0.7848 1.0848 -2.5152 0.7848 -0.5152 0.3848 1.8848 -0.3152
 -0.5030 1.1970 1.5970 -1.0030 3.0970 0.3970 1.3970 -1.6030 -3.1030
 0.1727 1.2727 0.7727 -1.1273 0.2727 -0.1273 -0.0273 -0.9273 -1.0273

Columns 19 through 27
 -3.5667 -1.4667 2.5333 -1.6667 2.5333 1.4333 4.0333 0.0333 -0.6667
 -2.0667 -2.8667 1.6333 0.1333 -0.6667 0.8333 2.8333 -0.8667 -1.8667
 -1.4242 -1.4242 1.2758 0.5758 -1.1242 -1.7242 2.0758 -0.9242 0.0758
 -1.2697 -0.7697 -0.3697 1.6303 -0.4697 0.9303 0.5303 0.6303 -0.7697
 -4.4152 -3.7152 0.8848 0.9848 -2.2152 0.8848 2.7848 -0.0152 -1.1152
 -4.5030 -2.4030 0.7970 2.2970 -1.6030 0.0970 3.8970 -1.6030 2.3970
 -0.8273 -0.1273 -0.1273 1.1727 -0.1273 -0.1273 0.2727 -0.3273 0.0727

Columns 28 through 33
 -1.4667 -1.2667 -0.5667 -2.9667 -0.5667 -2.1667
 0.4333 -1.8667 2.6333 3.8333 1.2333 1.1333
 1.1758 -0.2242 -0.2242 0.5758 -0.9242 1.6758
 -0.1697 -1.7697 0.4303 1.6303 1.2303 0.8303
 0.2848 -1.3152 4.8848 2.6848 3.6848 2.8848
 0.4970 -0.8030 1.7970 1.6970 0.9970 3.3970
 -0.1273 -0.0273 -1.0273 -0.4273 -0.4273 -0.8273
[M, N] = size(X)
M = 33
N = 7
[M, N] = size(X')
M = 7
N = 33

% Therefore, X can be made to represent the data matrix of X'
 X=X'
X =

Columns 1 through 9
3.1333 4.4333 2.1333 1.5333 -4.2667 2.6333 0.5333 3.1333 -1.8667
 -0.5667 0.0333 1.2333 -1.3667 -0.2667 -0.5667 -1.7667 -2.1667 -1.2667
 0.0758 1.0758 1.4758 0.1758 -0.7242 0.9758 -0.2242 -1.1242 -0.9242
 0.1303 -0.2697 2.1303 0.0303 -1.0697 0.0303 -0.1697 -1.0697 -0.4697
 -2.0152 0.9848 1.3848 1.3848 -4.2152 -0.3152 -2.3152 -0.7152 -3.4152
 -1.7030 2.0970 0.4970 -0.0030 -1.0030 -0.0030 -3.6030 -1.6030 -3.1030
 0.6727 0.3727 1.1727 0.6727 0.5727 0.4727 -0.1273 -0.1273 0.0727

Columns 10 through 18
 -4.3667 3.0333 -2.3667 -4.0667 2.3333 2.7333 0.0333 -3.9667 1.0333
 -0.0667 2.4333 1.9333 -1.3667 1.8333 -1.7667 0.2333 -1.3667 0.3333
 -0.2242 -0.7242 2.0758 -0.7242 2.8758 0.7758 0.7758 -2.4242 -2.6242
 0.5303 -0.2697 0.6303 -1.0697 0.8303 -0.2697 0.5303 -1.5697 -0.8697

0.4848 0.7848 1.0848 -2.5152 0.7848 -0.5152 0.3848 1.8848 -0.3152
-0.5030 1.1970 1.5970 -1.0030 3.0970 0.3970 1.3970 -1.6030 -3.1030
0.1727 1.2727 0.7727 -1.1273 0.2727 -0.1273 -0.0273 -0.9273 -1.0273

Columns 19 through 27
-3.5667 -1.4667 2.5333 -1.6667 2.5333 1.4333 4.0333 0.0333 -0.6667
-2.0667 -2.8667 1.6333 0.1333 -0.6667 0.8333 2.8333 -0.8667 -1.8667
-1.4242 -1.4242 1.2758 0.5758 -1.1242 -1.7242 2.0758 -0.9242 0.0758
-1.2697 -0.7697 -0.3697 1.6303 -0.4697 0.9303 0.5303 0.6303 -0.7697
-4.4152 -3.7152 0.8848 0.9848 -2.2152 0.8848 2.7848 -0.0152 -1.1152
-4.5030 -2.4030 0.7970 2.2970 -1.6030 0.0970 3.8970 -1.6030 2.3970
-0.8273 -0.1273 -0.1273 1.1727 -0.1273 -0.1273 0.2727 -0.3273 0.0727

Columns 28 through 33
-1.4667 -1.2667 -0.5667 -2.9667 -0.5667 -2.1667
0.4333 -1.8667 2.6333 3.8333 1.2333 1.1333
1.1758 -0.2242 -0.2242 0.5758 -0.9242 1.6758
-0.1697 -1.7697 0.4303 1.6303 1.2303 0.8303
0.2848 -1.3152 4.8848 2.6848 3.6848 2.8848
0.4970 -0.8030 1.7970 1.6970 0.9970 3.3970
-0.1273 -0.0273 -1.0273 -0.4273 -0.4273 -0.8273
% The covariance matrix for X can be obtained as follows
Covariance = (1/(N-1))*X*X'
Covariance =
6.9492 0.6379 0.9696 0.3715 1.1046 1.2501 0.6175
0.6379 2.8204 1.0418 1.0296 2.7040 2.3279 0.1569
0.9696 1.0418 1.7363 0.6323 1.0840 2.0109 0.3587
0.3715 1.0296 0.6323 0.9028 1.2836 1.0948 0.2099
1.1046 2.7040 1.0840 1.2836 5.0570 3.3684 -0.0426
1.2501 2.3279 2.0109 1.0948 3.3684 4.3891 0.3762
0.6175 0.1569 0.3587 0.2099 -0.0426 0.3762 0.4052
% find the eigenvectors and eigenvalues [PC, V] = eig(covariance)
[PC, V] = eig(Covariance)
PC =
-0.0700 -0.0039 -0.0472 0.0124 -0.1158 -0.9114 -0.3854
0.0422 -0.0906 -0.3891 0.7960 -0.1191 0.2061 -0.3850
0.0955 -0.6474 0.3860 0.0718 0.5929 0.0098 -0.2570
-0.4830 0.4439 0.6656 0.2931 0.0125 0.0811 -0.1845
0.2295 -0.1703 0.2519 -0.3391 -0.5925 0.2770 -0.5587
-0.1793 0.3559 -0.3834 -0.3758 0.4768 0.1934 -0.5368
0.8161 0.4689 0.2097 0.1377 0.2056 -0.0773 -0.0531
V =
0.1835 0 0 0 0 0 0
0 0.3742 0 0 0 0 0
0 0 0.5624 0 0 0 0
0 0 0 1.0798 0 0 0
0 0 0 0 2.0091 0 0
0 0 0 0 0 6.2129 0
0 0 0 0 0 0 11.8379

% Extract diagonal of matrix as vector V = diag(V)
V = diag(V)
V =
0.1835
0.3742
0.5624
1.0798
2.0091
6.2129
11.8379
% Sort the variances in decreasing order
[junk, rindices] = sort(-1*V)
junk =
-11.8379
-6.2129

```
 -2.0091
 -1.0798
 -0.5624
 -0.3742
 -0.1835
rindices =
 1
 2
 3
 4
 5
 6
 7
V = V (rindices)
V =
 11.8379
 6.2129
 2.0091
 1.0798
 0.5624
 0.3742
 0.1835
PC = PC (:rindices)
PC =
 -0.3854 -0.9114 -0.1158 0.0124 -0.0472 -0.0039 -0.0700
 -0.3850 0.2061 -0.1191 0.7960 -0.3891 -0.0906 0.0422
 -0.2570 0.0098 0.5929 0.0718 0.3860 -0.6474 0.0955
 -0.1845 0.0811 0.0125 0.2931 0.6656 0.4439 -0.4830
 -0.5587 0.2770 -0.5925 -0.3391 0.2519 -0.1703 0.2295
 -0.5368 0.1934 0.4768 -0.3758 -0.3834 0.3559 -0.1793
 -0.0531 -0.0773 0.2056 0.1377 0.2097 0.4689 0.8161

% Singular Value Decomposition of the data matrix(Deviation matrix)
% construct the matrix
Y = X / sqrt(N-1)
Y =

Columns 1 through 9
 0.5539 0.7837 0.3771 0.2711 -0.7542 0.4655 0.0943 0.5539 -0.3300
 -0.1002 0.0059 0.2180 -0.2416 -0.0471 -0.1002 -0.3123 -0.3830 -0.2239
 0.0134 0.1902 0.2609 0.0311 -0.1280 0.1725 -0.0396 -0.1987 -0.1634
 0.0230 -0.0477 0.3766 0.0054 -0.1891 0.0054 -0.0300 -0.1891 -0.0830
 -0.3562 0.1741 0.2448 0.2448 -0.7451 -0.0557 -0.4093 -0.1264 -0.6037
 -0.3011 0.3707 0.0879 -0.0005 -0.1773 -0.0005 -0.6369 -0.2834 -0.5485
 0.1189 0.0659 0.2073 0.1189 0.1012 0.0836 -0.0225 -0.0225 0.0129

Columns 10 through 18
 -0.7719 0.5362 -0.4184 -0.7189 0.4125 0.4832 0.0059 -0.7012 0.1827
 -0.0118 0.4302 0.3418 -0.2416 0.3241 -0.3123 0.0412 -0.2416 0.0589
 -0.0396 -0.1280 0.3669 -0.1280 0.5084 0.1371 0.1371 -0.4285 -0.4639
 0.0937 -0.0477 0.1114 -0.1891 0.1468 -0.0477 0.0937 -0.2775 -0.1537
 0.0857 0.1387 0.1918 -0.4446 0.1387 -0.0911 0.0680 0.3332 -0.0557
 -0.0889 0.2116 0.2823 -0.1773 0.5475 0.0702 0.2470 -0.2834 -0.5485
 0.0305 0.2250 0.1366 -0.1993 0.0482 -0.0225 -0.0048 -0.1639 -0.1816

Columns 19 through 27
 -0.6305 -0.2593 0.4478 -0.2946 0.4478 0.2534 0.7130 0.0059 -0.1179
 -0.3653 -0.5068 0.2887 0.0236 -0.1179 0.1473 0.5009 -0.1532 -0.3300
 -0.2518 -0.2518 0.2255 0.1018 -0.1987 -0.3048 0.3669 -0.1634 0.0134
 -0.2245 -0.1361 -0.0654 0.2882 -0.0830 0.1645 0.0937 0.1114 -0.1361
 -0.7805 -0.6568 0.1564 0.1741 -0.3916 0.1564 0.4923 -0.0027 -0.1971
 -0.7960 -0.4248 0.1409 0.4061 -0.2834 0.0171 0.6889 -0.2834 0.4237
 -0.1462 -0.0225 -0.0225 0.2073 -0.0225 -0.0225 0.0482 -0.0579 0.0129
```

Columns 28 through 33
-0.2593 -0.2239 -0.1002 -0.5244 -0.1002 -0.3830
0.0766 -0.3300 0.4655 0.6776 0.2180 0.2003
0.2078 -0.0396 -0.0396 0.1018 -0.1634 0.2962
-0.0300 -0.3128 0.0761 0.2882 0.2175 0.1468
0.0504 -0.2325 0.8635 0.4746 0.6514 0.5100
0.0879 -0.1420 0.3177 0.3000 0.1762 0.6005
-0.0225 -0.0048 -0.1816 -0.0755 -0.0755 -0.1462
% SVD is obtained as follows
[U, S, PC] = svd (Y)
u =
-0.3854 0.9114 0.1158 -0.0124 0.0472 0.0039 0.0700
-0.3850 -0.2061 0.1191 -0.7960 0.3891 0.0906 -0.0422
-0.2570 -0.0098 -0.5929 -0.0718 -0.3860 0.6474 -0.0955
-0.1845 -0.0811 -0.0125 -0.2931 -0.6656 -0.4439 0.4830
-0.5587 -0.2770 0.5925 0.3391 -0.2519 0.1703 -0.2295
-0.5368 -0.1934 -0.4768 0.3758 0.3834 -0.3559 0.1793
-0.0531 0.0773 -0.2056 -0.1377 -0.2097 -0.4689 -0.8161
S =

Columns 1 through 9
3.4406 0 0 0 0 0 0 0 0
0 2.4926 0 0 0 0 0 0 0
0 0 1.4174 0 0 0 0 0 0
0 0 0 1.0392 0 0 0 0 0
0 0 0 0 0.7500 0 0 0 0
0 0 0 0 0 0.6117 0 0 0
0 0 0 0 0 0 0.4283 0 0

Columns 10 through 18
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

Columns 19 through 27
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

Columns 28 through 33
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
PC =

Columns 1 through 9
0.0499 0.2767 -0.0339 -0.1782 -0.1120 -0.0290 -0.0384 -0.1359 0.0320
-0.1872 0.2408 -0.0761 0.1685 0.1095 0.0240 -0.0324 -0.2631 0.0081
-0.1630 0.0790 -0.0206 -0.2116 -0.4269 -0.1043 -0.0828 0.1414 0.0134
-0.0474 0.0953 0.0741 0.2421 -0.2448 -0.0278 -0.2908 -0.0791 -0.0561
0.2566 -0.1656 -0.2769 -0.2134 0.2930 -0.1920 -0.1712 0.1604 -0.2721
-0.0463 0.1865 -0.0778 0.0283 -0.1211 0.0875 -0.0761 -0.0882 -0.1678
0.1951 0.1556 0.0447 -0.1116 -0.2910 0.2081 0.0167 -0.0060 -0.1334

0.0709 0.2765 0.1436 0.2131 0.0101 0.0206 -0.0487 0.8878 0.0253
0.2621 0.0112 -0.0464 -0.1869 -0.0605 -0.0067 -0.0197 -0.0053 0.8974
0.0852 -0.2859 0.0130 -0.0137 -0.2003 -0.0645 -0.1517 0.0249 -0.0337
-0.1551 0.1377 0.0881 -0.2217 0.3638 -0.2907 -0.3944 -0.0002 0.0519
-0.1021 -0.2253 -0.1945 -0.1670 -0.0950 0.1399 -0.3031 0.0849 -0.0012
0.2302 -0.1793 -0.1211 0.0731 0.1775 0.0935 0.2653 0.0343 -0.0226
-0.2370 0.0609 -0.2862 -0.0929 0.0221 0.1653 0.1506 0.0835 0.0316
-0.0227 0.2075 -0.1021 0.2361 -0.0871 0.0876 0.1464 -0.0473 0.0344
-0.0700 -0.0317 -0.1082 0.0446 -0.0272 -0.0378 0.1481 0.0312 0.0167
0.1451 -0.2458 0.3625 0.3293 0.0864 0.0909 -0.2929 -0.0845 0.0589
0.1133 0.1119 0.4029 -0.1644 0.2063 0.0734 0.1004 -0.0551 -0.0508
0.3955 -0.0481 -0.0122 -0.1550 -0.0044 0.1962 0.0996 -0.0095 -0.1261
0.2851 0.0578 -0.0857 0.0821 -0.0192 -0.1629 0.1272 -0.0399 -0.0525
-0.1429 0.1121 -0.0116 -0.1187 0.1457 0.3106 -0.0612 0.0199 0.0241
-0.0875 -0.1639 -0.1611 0.0733 -0.2233 -0.4465 -0.0665 0.0377 0.0200
0.0905 0.2418 0.0454 -0.1052 0.1360 -0.0916 0.1695 -0.0331 -0.0385
-0.0587 0.0569 0.2220 -0.0810 0.0658 -0.3676 0.2466 -0.0097 -0.0125
-0.3565 0.1081 -0.0869 -0.0406 0.2061 0.0984 0.0238 0.0524 0.0939
0.0683 0.0323 0.1576 0.0015 -0.2217 -0.0679 0.1712 -0.0356 -0.0551
0.0221 -0.0220 -0.2686 0.3788 0.2145 -0.2480 0.1153 -0.0009 0.1036
-0.0150 -0.1141 -0.1067 -0.0103 -0.0226 0.2316 -0.0774 0.0361 0.0050
0.1417 -0.0076 -0.0754 0.2199 0.1196 0.1563 -0.2737 -0.0468 0.0454
-0.2290 -0.2037 0.3273 0.0466 0.1113 0.1660 0.0488 0.0208 0.0661
-0.1628 -0.3359 0.0775 -0.3278 0.0255 0.0111 0.1651 0.1274 -0.0353
-0.1447 -0.1495 0.3006 0.0704 -0.1097 -0.1623 0.1126 -0.0062 0.0238
-0.1838 -0.2704 -0.1073 0.1922 -0.0262 0.1389 0.2739 0.0773 0.0686

Columns 10 through 18
0.2691 -0.3629 0.1364 0.2900 -0.1199 -0.0598 0.0220 0.3628 -0.1467
0.1398 0.1018 0.1651 -0.0760 0.0133 -0.2740 -0.0085 -0.0440 0.1439
-0.1017 0.1372 -0.2970 0.1239 -0.2475 -0.0758 -0.0795 0.2841 0.4183
-0.1948 0.0229 0.0868 0.2332 0.2681 0.0189 0.0226 -0.0721 0.0530
-0.0871 -0.1523 -0.0983 -0.1372 0.0367 0.1633 0.0162 0.2262 0.0458
-0.1151 -0.0162 -0.2290 0.1964 0.1251 0.0794 0.1782 -0.1990 -0.1222
0.0776 0.5066 0.2068 -0.2149 -0.0817 -0.1465 -0.0656 0.2431 -0.1349
0.0711 -0.0403 0.0752 0.0300 0.0283 -0.0643 0.0220 -0.0521 -0.0392
-0.0641 0.0109 -0.0263 -0.0223 0.0845 0.0375 0.0376 -0.0424 -0.0124
0.8638 0.0235 -0.0918 0.0169 0.1007 0.0599 0.0308 -0.1514 0.0528
0.0540 0.6147 -0.0310 0.1242 0.0273 0.1196 0.0503 -0.0349 -0.0954
-0.0941 -0.0640 0.7810 0.0562 -0.0370 0.0779 0.0274 -0.0568 0.1010
0.0069 0.2119 0.0471 0.8161 -0.0473 -0.0601 -0.0514 0.0326 0.0406
0.0650 0.0564 -0.0402 -0.0174 0.8119 -0.0413 -0.0644 0.2313 0.0755
0.0747 0.1027 0.0631 -0.0240 -0.0818 0.8821 -0.0431 0.0971 0.0645
0.0148 0.0757 0.0310 -0.0343 -0.0561 -0.0357 0.9536 0.0954 0.0464
-0.0513 -0.0207 -0.0397 -0.0040 0.1239 0.0156 0.0663 0.6460 -0.0411
0.0430 -0.0610 0.1021 -0.0035 0.1002 0.0561 0.0580 -0.0715 0.7638
-0.0599 0.1304 -0.0198 -0.1022 0.0721 0.0092 0.0366 -0.0686 -0.0308
-0.0122 0.0883 0.0853 -0.0808 0.0582 -0.0512 -0.0163 0.0028 0.0375
0.0697 -0.0560 -0.0714 0.0253 -0.1083 0.0080 0.0135 0.0558 -0.0412
-0.0880 -0.0289 0.0022 0.0443 0.0536 0.0191 -0.0361 0.0131 0.1262
0.0721 -0.0287 0.1188 -0.0236 0.0117 -0.0146 -0.0050 0.0983 -0.0903
0.0293 -0.0449 0.1878 0.0077 0.0841 0.0319 -0.0200 0.0748 -0.1366
0.1170 -0.0818 -0.0107 0.0326 -0.1745 -0.0142 -0.0442 0.1723 -0.0065
-0.0380 0.0751 0.0823 0.0120 0.0957 -0.0025 0.0104 -0.0136 -0.0583
0.0820 0.0868 0.0978 -0.1217 -0.1115 -0.1289 -0.1026 0.0961 0.1571
-0.0251 0.0495 -0.1085 -0.0165 -0.0585 -0.0007 0.0034 -0.0232 0.0596
0.0109 -0.0216 -0.0896 -0.0252 -0.0260 -0.0428 0.0229 -0.1405 0.0853
0.0175 0.0083 0.0087 0.0119 -0.0111 0.0392 0.0077 -0.0742 -0.1190
-0.0688 0.0360 -0.0324 -0.0172 0.0029 0.1119 -0.0101 0.0501 -0.0807
-0.0330 0.0132 0.0856 0.0381 0.0876 0.0349 -0.0003 -0.0580 -0.0901
0.0103 0.2230 0.0099 -0.0972 -0.1317 -0.0701 -0.0869 0.0957 0.0875

Columns 19 through 27
0.1829 0.0727 -0.1433 0.0704 -0.2385 -0.1984 -0.2358 -0.0076 0.1268
-0.0195 -0.1969 0.0323 0.0444 -0.0663 0.1604 0.0449 0.0558 -0.3399

0.1150 0.0726 -0.0172 -0.2329 0.1722 0.2288 -0.0737 0.0508 0.0003
0.0945 -0.1152 0.3075 -0.3645 0.0245 -0.2125 0.2807 -0.2398 0.0776
-0.2508 -0.2058 0.1133 -0.1303 -0.1034 -0.0189 0.1731 0.0521 -0.0417
-0.2188 0.1090 -0.1695 0.2566 0.0874 0.3125 0.1454 -0.0084 0.4858
-0.3070 -0.1511 0.0139 0.1950 -0.1451 -0.0572 0.1344 -0.2175 0.1302
0.0241 -0.0143 -0.0273 0.0662 -0.0399 0.0119 -0.0192 0.0111 -0.0395
-0.1197 -0.0714 0.0453 -0.0038 -0.0044 0.0425 0.1269 -0.0173 0.0447
-0.0292 -0.0055 0.0662 -0.0783 0.0973 0.0663 0.1147 -0.0088 0.0358
0.1398 0.0998 -0.0691 0.0010 -0.0389 -0.0887 -0.1351 0.1034 0.0160
0.0177 0.1032 -0.0666 -0.0186 0.1273 0.1631 -0.0313 0.0902 0.0817
-0.1082 -0.0851 0.0315 0.0305 -0.0061 0.0338 0.0431 -0.0123 -0.1126
0.0386 0.0474 -0.0699 0.0021 -0.0138 0.0066 -0.1299 0.0229 -0.0072
0.0269 -0.0327 -0.0097 0.0089 -0.0285 0.0174 -0.0442 -0.0087 -0.0920
0.0204 -0.0223 0.0301 -0.0509 -0.0122 -0.0438 -0.0175 -0.0257 -0.0548
0.0063 0.0596 -0.0403 0.1031 0.1074 0.1368 0.0294 0.0620 -0.0068
-0.0975 -0.0042 -0.0336 0.1547 -0.0939 -0.0918 0.0419 -0.0522 0.1423
0.7900 -0.0904 0.0192 0.0836 0.0007 0.1069 0.1553 -0.0226 0.0573
-0.0676 0.8636 0.1061 -0.0620 -0.0381 -0.0241 0.1117 -0.0400 -0.1211
0.0026 0.1064 0.8455 0.1454 -0.0016 0.0802 -0.1303 0.0691 0.0968
0.1096 -0.0529 0.1586 0.7463 0.0381 -0.1021 0.0607 -0.0387 -0.1174
-0.0394 -0.0672 0.0215 0.0260 0.8855 -0.1141 0.0114 -0.0426 -0.0115
0.0248 -0.0738 0.1197 -0.0855 -0.1235 0.7200 0.0500 -0.1232 -0.0054
0.1197 0.1134 -0.1205 0.0505 -0.0330 -0.0456 0.7862 0.0559 0.0014
-0.0574 -0.0707 0.0894 -0.0371 -0.0367 -0.0840 0.1224 0.8919 0.0567
0.1117 -0.0720 0.0598 -0.1072 -0.0142 -0.0409 -0.0891 0.0419 0.6744
-0.0243 0.0532 -0.0679 0.0508 0.0702 0.1353 -0.0301 0.0531 0.0329
0.0150 0.0392 -0.0842 0.0792 0.0808 0.2012 -0.0530 0.1252 -0.0897
0.0309 0.1110 -0.0744 0.1048 0.0194 -0.0395 -0.0721 -0.0159 0.1126
-0.0477 0.0406 0.0212 -0.0216 0.0055 -0.1027 0.0282 -0.0677 0.1499
0.0401 0.0045 0.0749 -0.0534 -0.0131 -0.1513 0.0478 -0.0977 0.0548
0.0274 0.0196 0.0016 -0.0230 0.0416 -0.0021 -0.0601 -0.0282 -0.0691

Columns 28 through 33
0.1442 0.1699 0.0952 0.0925 0.0452 0.2715
0.0168 -0.2628 0.2673 0.4905 0.2242 0.0662
-0.1243 0.0134 0.1849 -0.0002 0.1224 -0.1108
0.1624 0.1353 0.1150 0.0874 -0.2393 0.1684
0.0232 0.0132 0.3410 -0.0542 0.2132 0.2145
-0.1363 -0.1725 0.0202 0.1029 0.1403 0.2889
0.0036 0.1832 0.0600 -0.0641 0.0078 -0.1120
0.0232 -0.0654 0.0148 0.1369 0.0301 0.0583
-0.0054 -0.0472 0.0933 0.0437 0.0540 0.0993
-0.0313 -0.0631 -0.0007 0.0002 -0.0278 0.0170
0.0487 -0.0046 -0.0720 -0.0423 -0.0158 0.1319
-0.1004 -0.0802 -0.0235 -0.0554 0.0566 -0.0060
-0.0315 -0.0111 0.0300 -0.0106 0.0396 -0.1331
-0.0314 0.0928 0.0120 -0.0933 0.0402 -0.1057
0.0034 0.0010 0.0554 0.0872 0.0400 -0.0580
0.0133 0.0633 0.0164 -0.0337 -0.0213 -0.0735
-0.0611 -0.1986 -0.1407 0.0928 -0.0173 -0.0117
0.0454 0.0357 -0.0761 -0.0353 -0.0428 0.1090
-0.0456 -0.0821 0.0845 0.0506 0.0909 0.0555
0.0386 -0.0251 0.1214 0.1013 0.0279 0.0179
-0.0609 -0.0131 -0.0634 -0.0494 0.0677 -0.0076
0.0635 0.0557 0.0599 -0.0052 -0.0927 -0.0166
0.0691 0.0694 0.0527 0.0126 0.0024 0.0644
0.1434 0.1839 -0.0170 -0.0798 -0.1552 0.0432
-0.0074 0.0920 -0.0860 -0.1095 -0.0036 -0.0852
0.0552 0.0605 0.0321 0.0141 -0.0677 0.0545
0.0259 -0.0053 0.0485 0.0676 0.0131 -0.1809
0.9173 -0.0646 -0.0196 -0.0137 0.0611 -0.0510
-0.0887 0.8001 -0.0033 0.1305 0.1188 -0.0263
-0.0168 0.0546 0.7997 -0.1337 -0.1050 -0.0781
0.0048 0.1529 -0.0982 0.7627 -0.1113 -0.0476
0.0675 0.1054 -0.0976 -0.0790 0.8365 -0.0115

-0.0375 0.0799 -0.0685 -0.1052 -0.0519 0.7708
% Calculate the variances
S = diag(S)
S =
3.4406
2.4926
1.4174
1.0392
0.7500
0.6117
0.4283
V = S.* S
V =
11.8379
6.2129
2.0091
1.0798
0.5624
0.3742
0.1835

The outputs above were used to compare the PCA and SVD methods. First of all, a PCA was carried out by first normalizing the dataset. After normalization the covariance matrix was obtained and an eigen decomposition was done on the covariance matrix. Usually, the eigenvalue and eigenvector outputs are not sorted for MATLAB R2007b. Thus, a sorting command was used to re-order the eigenvalues and their corresponding eigenvectors in decreasing order. This plays a huge part during the data reduction process as can be done here. For example, using a cut-off of 5.00 and above for the eigenvalues, the first and second eigenvalues will be selected since their eigenvalues are 11.8379 and 6.2129. Hence, their corresponding eigenvectors will be chosen to obtain a new set of 2 variables to represent an initial set of 7 variables.

Similarly, a PCA was achieved using the SVD. Although, the process is rather different as SVD employs a direct computation of the data matrix, PCA applies an eigen decomposition on a covariance matrix. In the same vein, the SVD is done firstly by normalizing the data matrix and named it Y. Subsequently, the [u, S, PC] = svd (Y) is implemented to obtain the product of three matrices. The PC matrix can be discarded as it is redundant for principal component analysis. However, the U and S represent the eigenvectors and singular value matrix sorted in decreasing order. Just as captured in equation …, the eigenvalues obtained in PCA can be derived from the vector of singular values (diagonal elements of the singular value matrix) by multiplying the vector of singular values by itself to give a vector of eigenvalues V. Hence, the results for SVD can be equated to that of PCA from the outputs of both methods.

5. Conclusion

Singular value decomposition and principal component analysis are two eigenvalue methods used to reduce a high dimensional dataset into fewer dimensions while retaining important information. The disparity clearly lies in how these methods are being applied. PCA is done via eigen decomposition of the covariance matrix. This is possible because the covariance matrix is square symmetric. While SVD is carried out on the matrix itself after normalization. This pontification was verified by the results of the PCA and SVD analysis.

6. References

1. Aremu CO, Adebayo MA, Ariyo OJ, Adewale BB. Classification of genetic diversity and choice of parents for hybridization in cowpea *Vigna unguiculata* (L.) Walp for humid savanna ecology. African Journal of Biotechnology. 2007; 6(20):2.
2. Cadima J, Jollife IT. Variable selection and the interpretation of principal subspaces. J Agri. Biol. Environ. Statist. 2009; 6:62-79.
3. Deshunk SR, Purohit SG. Microarray Data, Statistical Analysis Usng R. Alpha Science International Ltd., 2007.
4. Geladi P, Kowalski B. Partial Least Squares Regression. A Tutorial. Analytica Chimica Acta. 1986; 185:17.
5. Gibiansky A. (May 29). Cool linear algebra. Retrieved from Andrew gibiansky, 2013. http://andrew.gibiansky.com/blog/mathematics/cool-linear-algebra-singular-value-decomposition/
6. K AR, Hai MN, Xiaoli S, Jianping M, James BA, Jonathan MH *et al*. A Singular Value Decomposition framework for retrievals with vertical distribution information from greenhouse gas column absorption spectrosco, 2018.
7. Khan J, Ringner M, Saal LH, Ladanyi M, Westermann F, Berthold F *et al*. Classification and diagnostic prediction of cancers using gene expression profiling and artific, 2001.
8. Wall ME, Dyck PA, Brettin TS. SVDMAN-singular value decomposition analysis of microarray data. Bioinformatics. 2001; 17:566-68.
9. Will T. Introduction to the singular value decomposition, 1999.
10. Zheng S, Nie F, Ding C. Regularized Singular Value Decomposition and Application to Recommender system, 2018. Retrieved from https://arxiv.org/pdf/1804.0509.