Principle Component Analysis Single Value Decomposition (SVD)

Lecture # 14

Principle Component Analysis (PCA): Introduction

Principle Component Analysis (PCA) is a technique to study the internal structure of the data in a way that best explains the variance in the data. If a multivariate dataset is visualised as a set of coordinates in a high dimensional data space (1 axis per variable), PCA can supply the user with a lower-dimensional picture, a projection of this object when viewed from its most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.

PCA can be thought of as fitting an n-dimensional ellipsoid to the data, where each axis of the ellipsoid represents a principal component. If some axis of the ellipsoid is small, then the variance along that axis is also small, and by omitting that axis and its corresponding principal component from our representation of the dataset, we lose only a small amount of information (Figure 1).

Example of the principle component in the direction of maximum variance and the second component orthogonal to it (Fig 1).

(from *Machine Learning in Action" by Peter Harrington)*

Fig 1: Finding the component with maximum variance

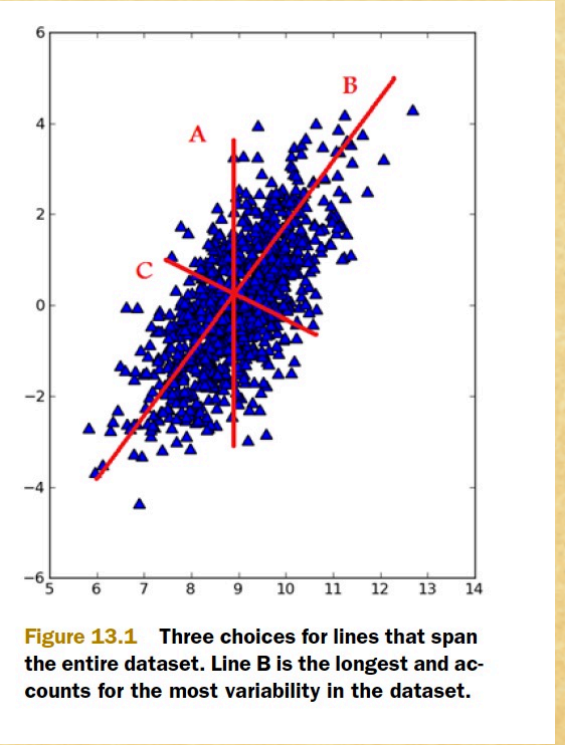


Figure 13.1 Three choices for lines that span the entire dataset. Line B is the longest and accounts for the most variability in the dataset.

Example of results after applying the PCA and reducing dimensionality (Fig 2)

(from *Machine Learning in Action" by Peter Harrington)*

Fig 2: Reduction from 2 to one dimension

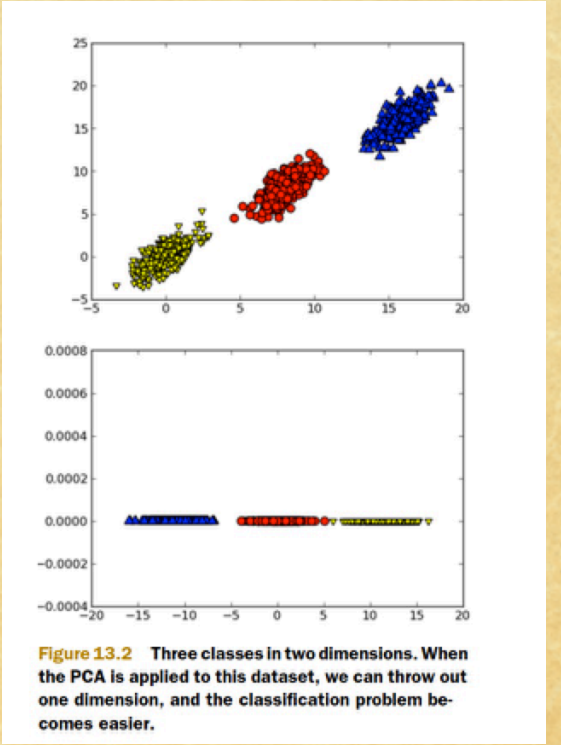


Figure 13.2 Three classes in two dimensions. When the PCA is applied to this dataset, we can throw out one dimension, and the classification problem becomes easier.

To find the axes of the ellipsoid, we must first subtract the mean of each variable from the dataset to center the data around the origin. Then, we compute the covariance matrix of the data, and calculate the eigenvalues and corresponding eigenvectors of this covariance matrix. Then we must normalize each of the orthogonal eigenvectors to become unit vectors. Once this is done, each of the mutually orthogonal, unit eigenvectors can be interpreted as an axis of the ellipsoid fitted to the data. This choice of basis will transform our covariance matrix into a diagonal form with the diagonal elements representing the variance of each axis. The proportion of the variance that each eigenvector represents can be calculated by dividing the eigenvalue corresponding to that eigenvector by the sum of all eigenvalues (Figure 2).

PCA is defined as an orthogonal linear transformation that transforms data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called first principle component), the second greatest variance on the second coordinate and so on. (Figure 1).

Refresher from Statistics Lecture

Some Definitions

Standard Deviation: The standard deviation of a dataset is the measure of the deviation of dataset from their mean value. For a set of data $X = \{x_1, x_2, \dots, x_n\}$, the mean is defined as $\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$. Here $\bar{x}$ is the mean.

The standard deviation is defined as

$$s = \left[\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1} \right]^{1/2}$$

Variance

Variance is another expression of spread of data in a dataset. It is very similar to the standard deviation. The variance is defined as

$$s^2 = \left[\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1} \right]$$

The variance will be used in the next section where we define covariance matrix.

Covariance

The last two measures we have looked at are purely 1-dimensional. this could be: heights of all the people in the room, marks for the last exam etc. However many data sets have more than one dimension, and the aim of the statistical analysis of these data sets is usually to see if there is any relationship between the dimensions. For example, we might have as our data set both the height of all the students in a class, and the mark they received for that paper. We could then perform statistical analysis to see if the height of a student has any effect on their mark.

Standard deviation and variance only operate on 1 dimension, so that you could only calculate the standard deviation for each dimension of the data set *independently* of the other dimensions. However, it is useful to have a similar measure to find out how much the dimensions vary from the mean *with respect to each other*.

Covariance is such a measure. Covariance is always measured between 2 dimensions. If you calculate the covariance between one dimension and itself, you get the variance. So, if you had a 3-dimensional data set (x,y,z), then you could estimate the variance between x and y dimensions, x and z dimensions and y and z dimensions. Measuring the covariance between (x and x) or (y and y) or (z and z) gives the variance for x, y and z respectively. The formula for covariance is similar to that for the variance. The variance is expressed as

$$var = \left[\sum_{i=1}^{n} \frac{(x_i - \bar{x})(x_i - \bar{x})}{n-1}\right]$$

where the square term is expanded. Based on this, the formula for covariance is

$$cov\,(x,y) = \left[\sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{n-1}\right]$$

How does this work? Lets use some example data. Imagine we have collected some 2-dimensional data about the hours in total that students spent studying and the mark that they received. So we have two dimensions, the first is the dimension, the hours studied, and the second is the dimension, the mark received. We can calculate the covariance between the Hours of study done and the Mark received. If the covariance is positive, it means that the number of hours studied and the grades received increase together. A negative covariance means the opposite- the grades decrease with an increase in the number of hours studied. If the covariance is zero, it means that the two dimensions are independent from one another. Obviously, cov x,y=cov y,x.

Covariance Matrix

As we discussed, covariance could always be measured between two dimensions. Therefore, if we have a dataset in more than two dimensions, there is more than one covariance measurement that can be calculated. From 3 dimensional data set (x,y,z) one could calculate cov (x,y), cov(x,z) and cov (y,z).

We could get all covariance values between all different dimensions in a matrix. The covariance matrix with n dimensions is defined as a matrix with elements being the covariance of two separate dimensions. For example, the covariance matrix for an imaginary 3 dimensional dataset using dimensions x, y and z has 3 rows and 3 columns as:

Covariance matrix:

$$c = \begin{pmatrix} cov\ (x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

As is clear, the diagonal elements in the matrix are the covariance of one of the dimensions with itself. These are the variances for that dimension. Also, since *cov a,b=cov b,a,* the matrix is symmetrical about the main diagonal.

Formulation of PCA

The Principle Component Analysis (PCA) is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data (Fig 3a). Once you find these patterns in the data, you compress the data (by reducing the number of dimensions without much lose of information)- (Figure 3b).

(from *Machine Learning in Action" by Peter Harrington)*

Figure 3b: The rate of change of variance as a function of principle components. As the principle components increase, the variance decreases.

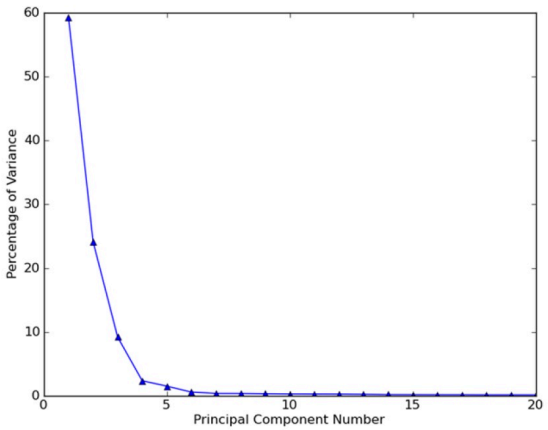


Figure 13.4 Percentage of total variance contained in the first 20 principal components. From this plot, you can see that most of the variance is contained in the first few principal components, and little information would be lost by dropping the higher ones. If we kept only the first six principal components, we'd reduce our dataset from 590 features to 6 features, almost a 100:1 compression.

Figure 3a: Dimensionality reduction after applying the PCA. All the points lie on the same line (1 dimensional)

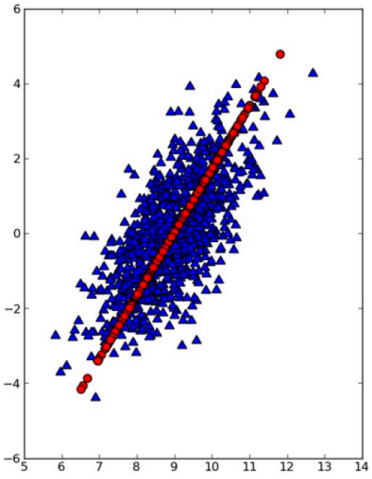


Figure 13.3 The original dataset (triangles) plotted with the first principal component (circles)

To formulate PCA, consider a data matrix, X, where the sample mean of each column is shifted to zero and each of the n rows represent a different repetition of the experiment and each of the p columns gives a particular kind of features. The transformation is defined as a set of p-dimensional vectors $w_k = (w_1, w_2, \ldots, w_p)_k$ that map each row vector x_i of X to a new vector of principle component scores $t_i = (t_1, t_2, \ldots, t_l)_l$

$t_{ki} = x_i \, , \, w_k$ for $i = 1, \ldots, k$ and $k = 1, \ldots, l$

This is done in a way that the individual variables $t_1, \ldots, t_l$ of $\boldsymbol{t}$ considered over the dataset successively inherit the maximum possible variance from $\boldsymbol{x}$, with each weight vector $\boldsymbol{w}$ constrained to be a unit vector.

In order to maximize the variance, the first vector $\boldsymbol{w_1}$ has to satisfy

$$w_1 = \arg\max\{\sum_i (t_1)_i^2\} = \arg\max\{\sum_i (x_i . w)^2\}$$

for $\|w\| = 1$. Alternatively, writing this in matrix form gives:

$$w_1 = \arg\max\{\|Xw\|^2\} = \arg\max\{W^T X^T XW\}$$

Since w_1 has been defined to be a unit vector, it equivalently also satisfies

$$W_l = \arg\max\left\{\frac{W^T X^T XW}{W^T W}\right\}$$

With W_1 found, the first principle component of a data vector x_l can then be given as $t_{1(i)} = x_l . w_1$ in the transformed coordinates or as the corresponding vector in the original variables $\{X_l . w_1\}w_1$

Further Components

The *k-th* component can be found by subtracting the first principle components from $\boldsymbol{X}$

$$\hat{X}_k = X - \sum_{s=1}^{k} X\, W_s W_s^T$$

We now need to find the weight vector (w) that extracts the maximum variance from this new data matrix

$$w_k = \arg\max\left\{\left\|\hat{X}_k\, w\right\|^2\right\} = \arg\max\left\{\frac{w^T \hat{X}_k^T\, \hat{X}_k w}{w^T w}\right\}$$

This gives the remaining eigenvectors of $\boldsymbol{X^T X}$ with the maximum values of the quantity in brackets given by their corresponding eigenvalues (Figure 3b). Therefore, the weight vectors are eigenvectors of $\boldsymbol{X^T X}$. The k-th principle component of a data vector X_l can therefore be given as $t_k = x_l . w_k$ in the transformed coordinates or as the corresponding vector in the space of the original variables $\{x_l . w_k\}\, w_k$ where w_k is the k-th eigenvector of $X^T X$. The full principle component decomposition of X can therefore be given as

$$\boldsymbol{T = X\, W}$$

where W is a $p\ x\ p$ matrix whose columns are the eigenvectors of $\boldsymbol{X^T X}$

Demonstration of PCA Technique

We now follow the PCA technique step-by-step on a dataset to show how the technique works.

Step 1- Get the data: For simplicity, we take 2 dimensional data. This allows simple visualization of the data (Fig 4)

Step 2- Subtraction of the mean: We take the mean for each dimension separately and subtract the mean from each of the data dimensions. Therefore, all the x values have a mean <x> (the mean of x values of all the data points) and all the y values have mean <y> subtracted from them. This produces a data set whose mean is zero.

Step 3- Calculate the covariance matrix: The covariance matrix is calculated the same way as we discussed in the last section. Since the data is 2 dimensional, the covariance matrix also has two dimensions

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

Since the non-diagonal matrix elements here are positive, we should expect that both the x and y variables increase together.

Step 4- Calculating the eigenvalues and eigenvectors: Since the covariance matrix is square, we can calculate the eigenvalue and eigenvector for the matrix. This is an important step in extracting information from our data. These are estimates as described in the background section. For the covariance matrix we have

$$eigenvalues = \begin{pmatrix} 0.049 \\ 1.284 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -0.735 & -0.677 \\ 0.677 & -0.735 \end{pmatrix}$$

Both the eigenvectors here are unit vectors- with length unity. The data are shown in Figure 3. This shows, as expected from the covariance matrix, both variables increase together (covariance matrix has positive diagonal elements). The eigenvectors are shown as diagonal lines on the plot. They are perpendicular to one another. The eigenvectors also provide information about the pattern in the data- one of the eigenvectors goes through the data like a best-fit line, showing how these datasets are correlated along the line. The second eigenvector gives information about the less important pattern in the data. The point to take away here is that by calculating the eigenvectors of the covariance matrix we have been able to extract lines that characterize the data (Figure 5).

The Original Data

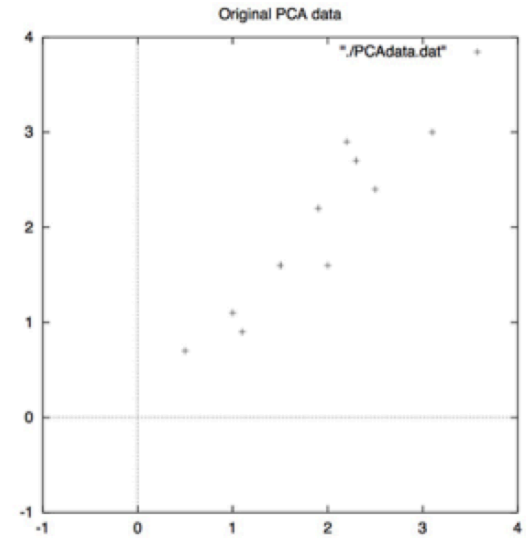
Figure 4. PCA example data. Original data on the left, data with the means subtracted on the right. A 2-D plot of the data is also shown (taken from Machine Learning in action by Peter Harrington)

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

DataAdjust =

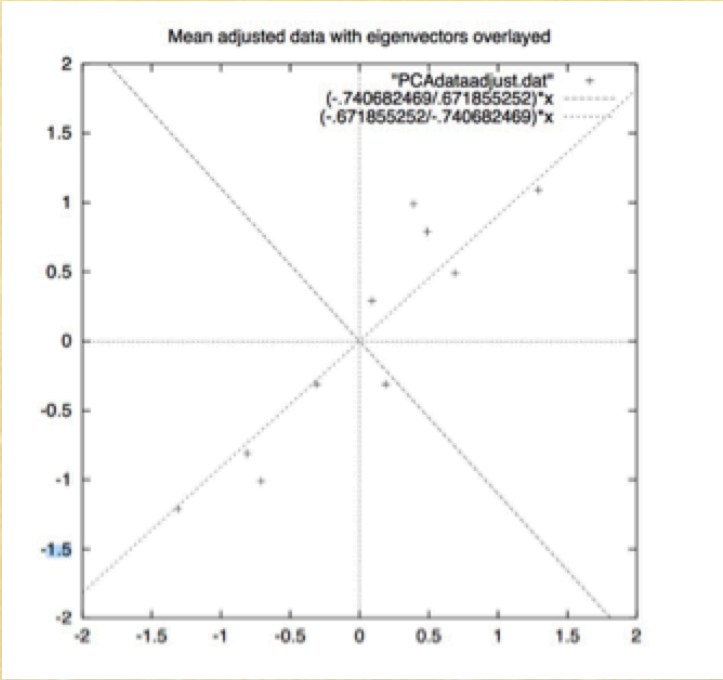
x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01



Means subtracted data with the eigenvalues

Figure 5. A plot of the normalized data (means subtracted) with the eigenvectors of the covariance matrix overlaid on top. The axes are rotated by 90 deg. These are the 1st and 2nd PCA components

(*from Machine Learning in Action by Peter Harrington*)



Step 5: Choosing components and forming a feature vector: Here is where dimensionality reduction (data compression) takes place. It is clear that the eigenvalues are different (from the last section). The eigenvector with the highest eigenvalue is the principle component. In the above example, the eigenvector with the largest eigenvalue was the one that went through the middle of the data. That is the most significant relationship between the data dimensions. When the eigenvectors are estimated for covariance matrices, the next step is to order them by the eigenvalues- from the highest to the lowest. This gives the components in the order of significance. One could always ignore the components of lesser significance. In this case, one loses some information, but if the eigenvalues are small, the loss is not much. Therefore, if you leave some components out, the final data set will have less dimensions than the original. For example, if you originally have n dimensions in the data, and you calculate n eigenvectors and eigenvalues, and choose only the first p eigenvectors, then the final data set has only p dimensions.

We could form a **feature vector** now that is constructed by taking the eigenvectors that you want to keep from the list of the eigenvectors and forming a column matrix with these eigenvectors. Therefore, given the fact that we only have 2 eigenvectors, we have two choices. We can either form a feature vector with both the eigenvectors:

$$\begin{pmatrix} -0.677 & -0.735 \\ -0.735 & 0.677 \end{pmatrix}$$

Or can choose to leave out the smaller, less significant component and only have a single column.

$$\begin{pmatrix} -0.677 \\ -0.735 \end{pmatrix}$$

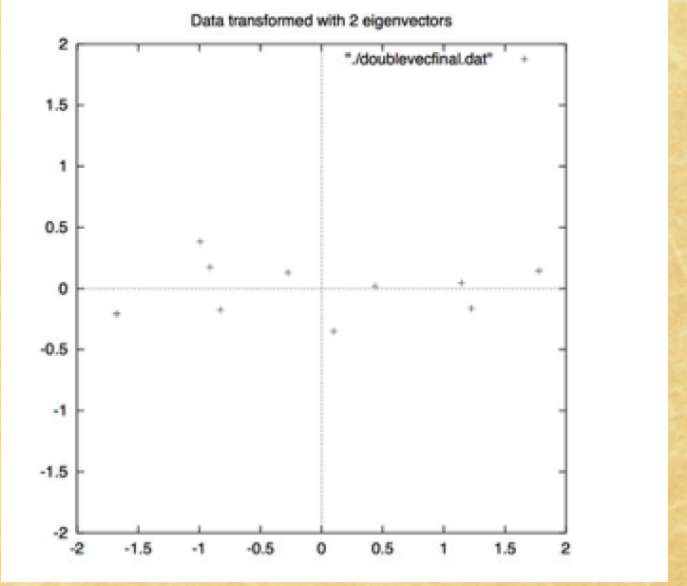
Step 6- Driving the new dataset: Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we take the transpose of the vector and multiply it on the left of the original dataset. We use the transpose and hence, the eigenvectors are now in the rows with each row holding a separate dimension. This will give us the original data terms of vectors we chose. The original data were in terms of x and y axes. We could display them in terms of any axes we wish to. For example, we could express the data in terms of eigenvector axes. When the dataset has reduced dimensionality (when we left some eigenvectors out), the new data are only in terms of the vectors we decided to keep. Here we do this for our data and for each of the feature vectors. We take the transpose of the result in each case to reproduce the data. Of course, if we use all the dimensions (all eigenvectors), we reproduce the original data (Figure 5).

In the case of keeping both eigenvectors for the transformation, we get the data and the plot found in Figure 6. This plot is basically the original data, rotated so that the eigenvectors are the axes. This is understandable since we have lost no information in this decomposition.

Figure 6. The table of data by applying PCA using both eigenvectors and a plot of the new data points. The pattern of the data is observed around the eigenvector (from Machine Learning in Action By Peter Harrington)

Transformed Data=

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287



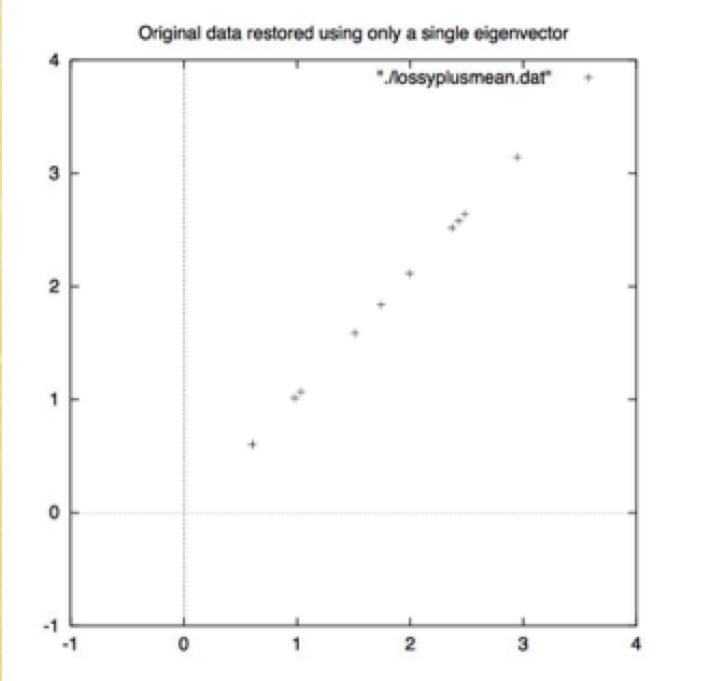
The other transformation we can make is by taking only the eigenvector with the largest eigenvalue. The result is listed in the following table. As expected, it only has a single dimension. If you compare this data set with the one resulting from using both eigenvectors, you will notice that this data set is exactly the first column of the other. So, if you were to plot this data, it would be 1 dimensional, and would be points on a line in exactly the positions of the points in the plot in Figure 7. We have effectively thrown away the whole other axis, which is the other eigenvector. Following table is the data after transforming using only the most significant eigenvector.

Transformed Data (Single eigenvector)

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

So what have we done here? Basically we have transformed our data so that is expressed in terms of the patterns between them, where the patterns are the lines that most closely describe the relationships between the data. This is helpful because we have now classified our data point as a combination of the contributions from each of those lines. Initially we had the simple and axes. (From Machine Learning in Action by Peter Harrington)

Fig 7. Reconstruction from the data that was derived using a single eigenvector. Dimension is reduced from two to one



Source used for this lecture

Machine Learning in Action by Peter Harrington

Single Value Decomposition (SVD)

Introduction

Often, a few pieces of data in our dataset can contain most of the information in our dataset. The other information in the matrix is noise or irrelevant. In linear algebra, there are many techniques for decomposing matrices. The decomposition is done to put the original matrix in a new form that's easier to work with. The new form is a product of two or more matrices. This decomposition can be thought of like factoring in algebra. How can we factor 12 into the product of two numbers? (1,12), (2,6), and (3,4) are all valid answers.

The various matrix factorization techniques have different properties that are more suited for one application or another. One of the most common factorizations is the Single Value Decomposition (SVD) . The SVD takes an original data set matrix called A, and decomposes it into three matrices called U, Σ and V^T.

The SVD is used to represent our original data set with a much smaller data set. In doing so, we remove noise and redundant information. In other words, SVD will extract information from a set of noisy data.

The SVD is a kind of matrix factorization which will break down our data matrix into separate parts.

The singular value decomposition states that every n x p matrix can be written as the product of three matrices: $A = U \Sigma V^T$ where

- U is an orthogonal n x n matrix
- Σ is a diagonal n x p matrix. In practice, the diagonal elements are ordered so that $\Sigma ii \geq \Sigma jj$ for all $i < j$.
- V is an orthogonal p x p matrix and V^T represents a matrix transpose.

The SVD represents the essential geometry of a linear transformation. It tells us that every linear transformation is a composition of three fundamental actions. Reading the equation from right to left: The

The matrix V represents a rotation or reflection of vectors in the *p*-dimensional domain.

- The matrix Σ represents a linear dilation or contraction along each of the *p* coordinate directions. If $n \neq p$, this step also canonically embeds (or projects) the *p*-dimensional domain into the *n*-dimensional range.
- The matrix U represents a rotation or reflection of vectors in the *n*-dimensional range.
- Thus the SVD specifies that every linear transformation is fundamentally a rotation or reflection, followed by a scaling, followed by another rotation or reflection.

If the original data set is size mxn , then U will be mxm, Σ will be mxn, and V^T will be nxn. Let's write this out on one line to be clear (the subscript is the matrix dimensions):

$$A = U_{mm} \Sigma_{mn} V^T_{nn}$$

U is a rotation, Σ is stretch and V^T is rotation (Figure **).

The decomposition creates the Σ which will have only diagonal elements with all the other elements of this matrix being zero. Another convention is that the diagonal elements of Σ are sorted from largest to smallest. These diagonal elements are called singular values and they correspond to the singular values of our original data set, A. on principal component analysis, we found the eigenvalues of a matrix. These eigenvalues tell us what features were most important in our data set. The same thing is true about the singular values in Σ.

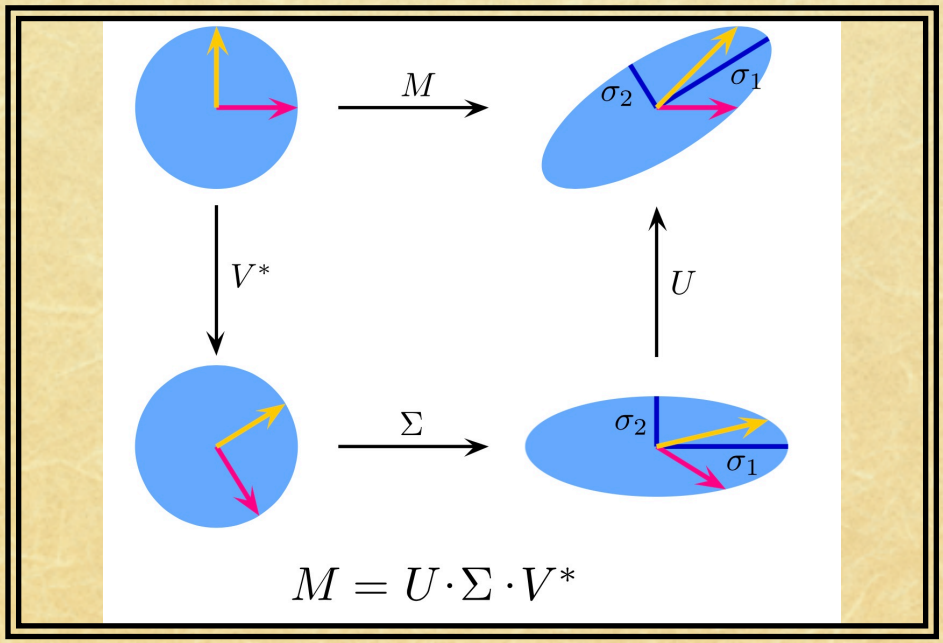


Fig 1.The matrixes defined lead to rotation (V^T), stretching (Σ) and rotation (U). Therefore the matrix $M = U \Sigma V^T$ when applied on a vector (unit vector) causes rotation, stretching and rotation

The single values and eigenvalues are related. Our singular values are the square root of the eigenvalues of AA^T

$A^TA = (U \Sigma V^T)^T (U \Sigma V^T) = (V \Sigma U^T) (U \Sigma V^T) = V \Sigma^2 V^T$

$A^T A V = V \Sigma^2 V^T V = V \Sigma^2$

This is an eigenvalue equation showing that Σ^2 is the eigenvalue of the matrix A. Similarly

$A^TA U = (U \Sigma V^T)^T (U \Sigma V^T) U = \Sigma V^T V \Sigma U^T U = U \Sigma^2 U^T U =$
$= U \Sigma^2$

Meaning that Σ^2 is the eigenvalue of U.

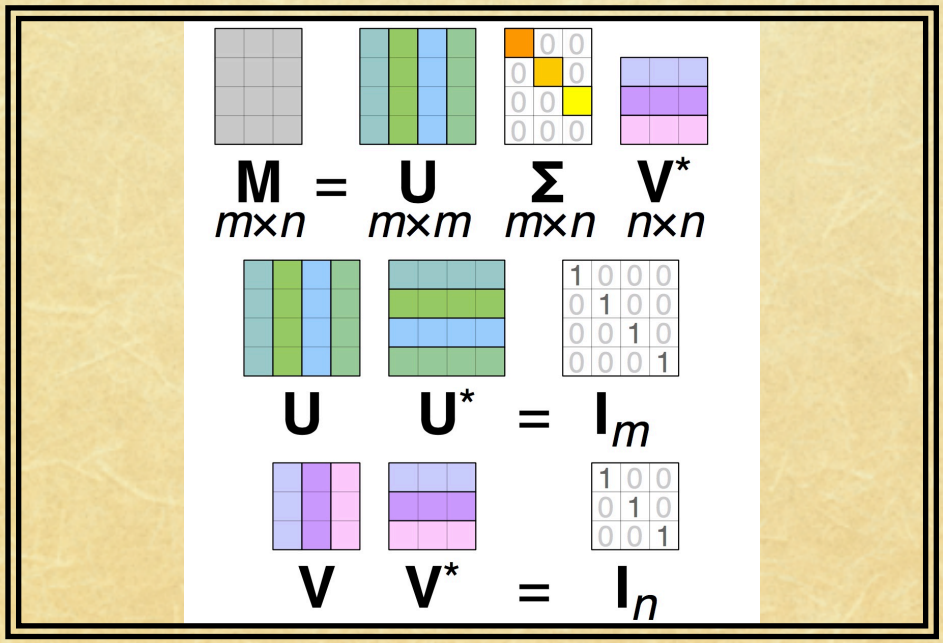


Fig 2: Shows the matrix combination in SVD

Often after a certain number of singular values (call this r) of a data set, the other values will drop to zero. This means that the data set has only r important features, and the rest of the features are noise or repeats.

The Singular-Value Decomposition, or SVD for short, is a matrix decomposition method for reducing a matrix to its constituent parts in order to make certain subsequent matrix calculations simpler (Figure 2).

Calculating the SVD consists of finding the eigenvalues and eigenvectors of AA^T and A^TA. The eigenvectors of A^TA make up the columns of V , the eigenvectors of AA^T make up the columns of U. Also, the singular values in Σ are square roots of eigenvalues from AA^T or A^TA. The singular values are the diagonal entries of the Σ matrix and are arranged in descending order. The singular values are always real numbers. If the matrix A is a real matrix, then U and V are also real.

Example for SVD (Taken from Dr. Edel Garcia)

https://fenix.tecnico.ulisboa.pt/downloadFile/3779576344458/singular-value-decomposition-fast-track-tutorial.pdf

Problem: Compute the full SVD for the following matrix:

$$A = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix}$$

Step 1. Compute its transpose A^T and A^TA.

Since $A^T = \begin{bmatrix} 4 & 3 \\ 0 & -5 \end{bmatrix}$ then, $A^TA = \begin{bmatrix} 4 & 3 \\ 0 & -5 \end{bmatrix}\begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix}$

$$A^TA = \begin{bmatrix} 25 & -15 \\ -15 & 25 \end{bmatrix}$$

Step 2. Determine the eigenvalues of $\mathbf{A}^T\mathbf{A}$ and sort these in descending order, in the absolute sense. Square roots these to obtain the singular values of **A**.

$$\mathbf{A}^T\mathbf{A} - c\mathbf{I} = \begin{bmatrix} 25 - c & -15 \\ -15 & 25 - c \end{bmatrix}$$

$$| \mathbf{A}^T\mathbf{A} - c\mathbf{I} | = (25 - c)(25 - c) - (-15)(-15) = 0$$

characteristic equation $\longrightarrow$ $c^2 - 50c + 400 = 0$

The quadratic equation gives two values.
In decreasing order, these are $\longrightarrow$ $|40| > |10|$

eigenvalues $\longrightarrow$ $c_1 = 40 \quad c_2 = 10$

singular values $\longrightarrow$ $s_1 = \sqrt{40} = 6.3245 > s_2 = \sqrt{10} = 3.1622$

Step 3. Construct diagonal matrix **S** by placing singular values in descending order along its diagonal. Compute its inverse, $\mathbf{S}^{-1}$.

$$\mathbf{S} = \begin{bmatrix} 6.3245 & 0 \\ 0 & 3.1622 \end{bmatrix} \qquad \mathbf{S}^{-1} = \begin{bmatrix} 0.1581 & 0 \\ 0 & 0.3162 \end{bmatrix}$$

Step 4. Use the ordered eigenvalues from step 2 and compute the eigenvectors of $\mathbf{A^TA}$. Place these eigenvectors along the columns of $\mathbf{V}$ and compute its transpose, $\mathbf{V^T}$.

for $c_1 = 40$

$$A^TA - cI = \begin{bmatrix} 25 - 40 & -15 \\ -15 & 25 - 40 \end{bmatrix} = \begin{bmatrix} -15 & -15 \\ -15 & -15 \end{bmatrix}$$

$$(A^TA - cI)\, x_1 = 0$$

$$\begin{bmatrix} -15 & -15 \\ -15 & -15 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$-15x_1 + -15x_2 = 0$$
$$-15x_1 + -15x_2 = 0$$

Solving for x_2 for either equation: $x_2 = -x_1$

$$x_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ -x_1 \end{bmatrix}$$

for $c_2 = 10$

$$A^TA - cI = \begin{bmatrix} 25 - 10 & -15 \\ -15 & 25 - 10 \end{bmatrix} = \begin{bmatrix} 15 & -15 \\ -15 & 15 \end{bmatrix}$$

$$(A^TA - cI)\, x_2 = 0$$

$$\begin{bmatrix} 15 & -15 \\ -15 & 15 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$15x_1 + -15x_2 = 0$$
$$-15x_1 + 15x_2 = 0$$

Solving for x_2 for either equation: $x_2 = x_1$

$$x_2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_1 \end{bmatrix}$$

Dividing by its length,

$$L = \sqrt{x_1^2 + x_2^2} = x_1\sqrt{2}$$

$$x_1 = \begin{bmatrix} x_1 / L \\ -x_1 / L \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix}$$

Dividing by its length,

$$L = \sqrt{x_1^2 + x_2^2} = x_1\sqrt{2}$$

$$x_2 = \begin{bmatrix} x_1 / L \\ x_1 / L \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix}$$

$$V = \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$V^T = \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}$$

Step 5. Compute **U** as $\mathbf{U = AVS^{-1}}$. To complete the proof, compute the full SVD using $\mathbf{A = USV^T}$.

$$U = AVS^{-1} = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix} \begin{bmatrix} 0.1581 & 0 \\ 0 & 0.3162 \end{bmatrix}$$

$$U = AVS^{-1} = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} \begin{bmatrix} 0.1118 & 0.2236 \\ -0.1118 & 0.2236 \end{bmatrix}$$

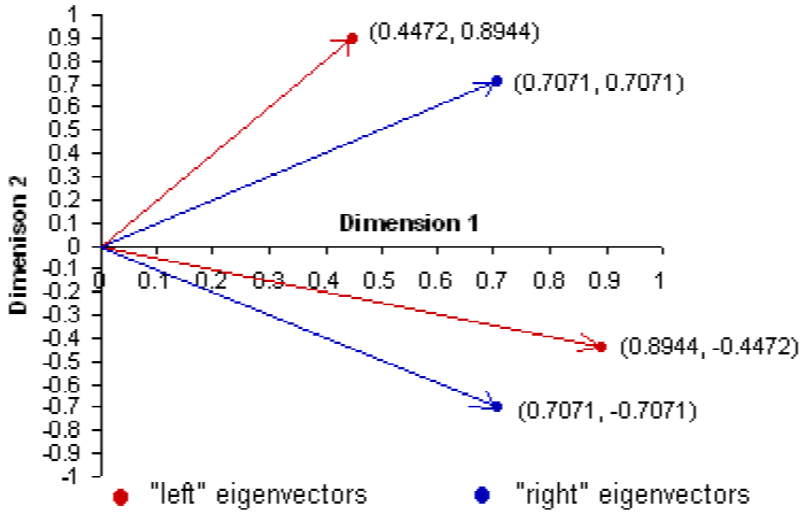
$$U = AVS^{-1} = \begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{bmatrix}$$

$$A = USV^T = \begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{bmatrix} \begin{bmatrix} 6.3245 & 0 \\ 0 & 3.1622 \end{bmatrix} \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}$$

$$A = USV^T = \begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{bmatrix} \begin{bmatrix} 4.4721 & -4.4721 \\ 2.2360 & 2.2360 \end{bmatrix}$$

$$A = USV^T = \begin{bmatrix} 3.9998 & 0 \\ 2.9999 & -4.9997 \end{bmatrix} \approx \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix}$$

The orthogonal nature of the **V** and **U** matrices is evident by inspecting their eigenvectors. This can be demonstrated by computing dot products between column vectors. All dot products are equal to zero. Alternatively, we can plot these and see they are all orthogonal.



Source of the material in this lecture

The material in this lecture are taken from the book Machine Learning in Action by Peter Harrington (Chapter 14)

The example on Single Value Decomposition (SVD) is taken from Dr. Edel Garcia's notes
https://fenix.tecnico.ulisboa.pt/downloadFile/377957634445 8/singular-value-decomposition-fast-track-tutorial.pdf