Principal Component Analysis(PCA)

Introduction:

Principal Component Analysis is one of the methods of feature extraction. Principal component analysis (PCA) in many ways forms the basis for multivariate data analysis. PCA provides an approximation of a data table, a data matrix, X, in terms of the product of two small matrices T and P'. These matrices, T and P', capture the essential data patterns of X [1].

1) What is PCA?

PCA stands for Principal Component Analysis. It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (or sometimes, principal modes of variation). The number of principal components is less than or equal to the smaller of the number of original variables or the number of observations.

This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

PCA was invented by Karl Pearson in 1901 as an analogue of the principal axis theorem. PCA is mostly used as a tool to reduce the dimensions of data. PCA can be done by eigenvalue decomposition of data covariance matrix or singular value decomposition of data matrix usually after normalising.

PCA is the simplest of the true eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing 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.

2) Problem Definition:

The starting point in all multivariate data analysis is a data matrix (a data table) denoted by X. The N rows in the table are termed “objects”. These often correspond to chemical or geological samples. The K columns are termed “variables” and comprise the measurements made on the objects.

Many of the goals of PCA are concerned with finding relationships between objects. for example, in finding classes of similar objects. The class membership may be known in advance, but it may also be found by exploration of the available data. Associated with this is the detection of outliers, since outliers do not belong to known classes.

Another goal could be data reduction. This is useful when large amounts of data may be approximated by a moderately complex model structure.

In general, almost any data matrix can be simplified by PCA. A large table of numbers is one of the more difficult things for the human mind to comprehend. PCA can be used together with a well selected set of objects and variables to build a model of how a physical or chemical system behaves, and this model can be used for prediction when new data are measured for the same system. PCA has also been used for unmixing constant sum mixtures. This branch is usually called curve resolution

Here our main aim is to project the higher dimension data into lower dimension.

3) Intuition

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 commensurately small amount of information.

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.

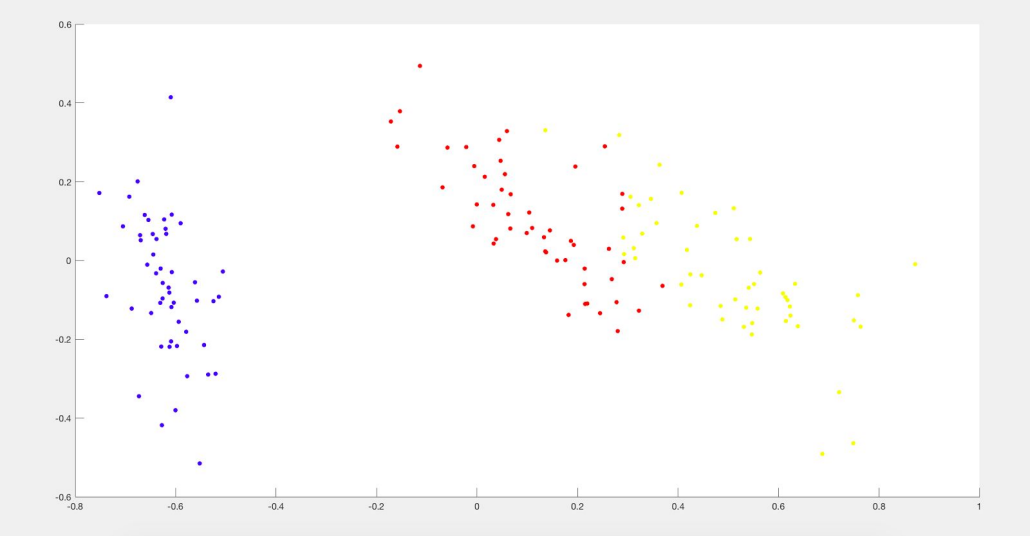
Then, we must orthogonalize the set of eigenvectors, and normalize each to become unit vectors. Once this is done, each of the mutually orthogonal, unit eigenvectors can be interpreted as an axis of the ellipsoid fitted to the data. The proportion of the variance that each eigenvector represents can be calculated by dividing the eigenvalue corresponding to that eigenvector by the sum of all eigenvalues.

This procedure is sensitive to the scaling of the data, and there is no consensus as to how to best scale the data to obtain optimal results.

4) Why PCA ?

a) Visualization :

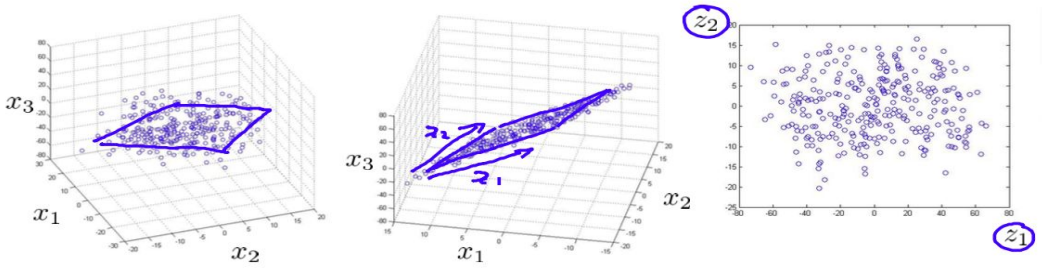
If we want to visualize a very high dimensional dataset, then we have to take the data and project into say a 2D plot or project into a 3D plot so you can render like a 3D display on a computer so we can better visualize data and look for structure.



(PCA 2D Visualization of Iris Dataset)

b) Compression :

In Machine Learning, sometimes we just give a really high dimensional input data and for computational reasons we don’t want to deal with such high dimensional computation data and such high potential data.



(Compression of data from 3D to 2D)

c) Learning :

One common use of PCA is to take a very high dimensional data and represent it with low dimensional subspace $y^{(i)}$ so we work with much lower dimensional data,this stuff just seems like a fact of life that when we give an extremely high dimensional data,of having all points lying on much lower subspaces.

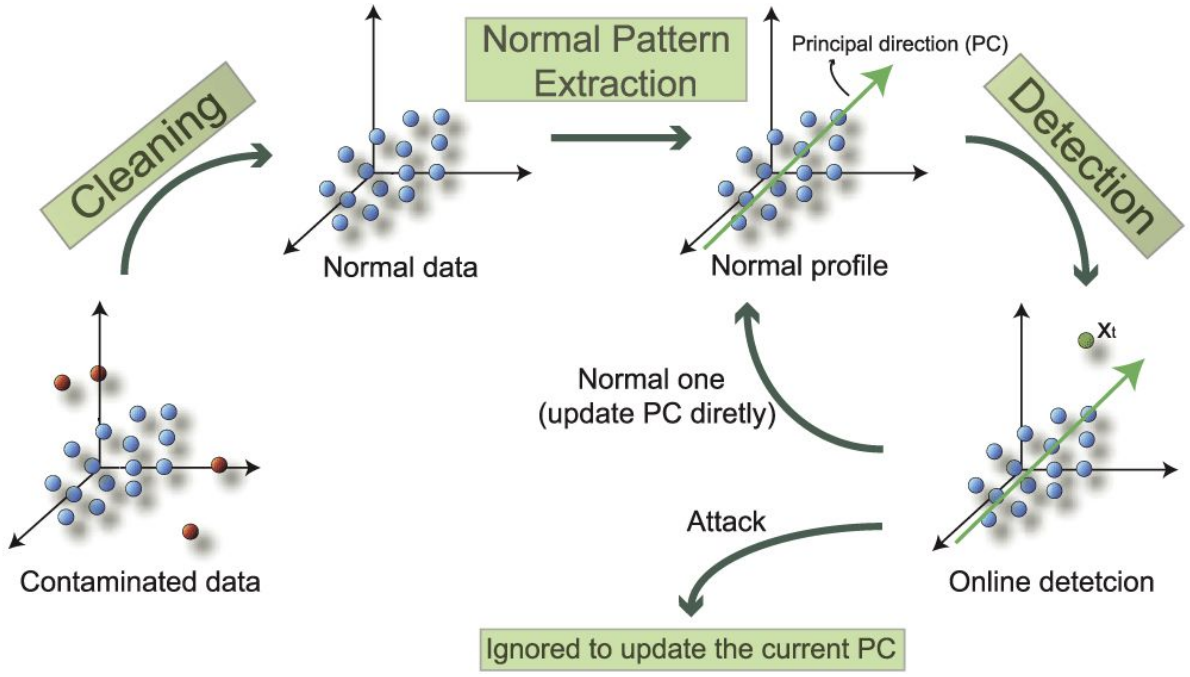
So very often we dramatically reduce dimension of the data and really not throwing too much of information away.

If the learning algorithms take long time to run very high dimensional data we can use PCA algorithm to reduce the data into a lower dimension so that learning algorithm can run much faster.

The more feature we have,more the complex the hypothesis cause.If say we are doing linear classification where if we have more features we have lots of features then we are more prone to overfitting.So we use PCA to reduce features and be less prone to overfitting

d) Anomaly Detection (Outlier Detection):

Suppose we have a dataset ,we run the PCA to find roughly the subspace on which the data lies.If we want to find anomalies in future examples then we have to look at future examples and see if they lie very far from your subspace.



(Anomaly Detection using PCA)

But this PCA is not the best anomaly detection algorithm. There are other algorithms that give better results for outlier detection than PCA.

e) PCA is also used in Matching / Distance calculation.

5) Algorithm :

Given input feature vector $X=\{X^{(1)}, X^{(2)}, X^{(3)}, \dots, X^{(n)}\}$ where $X \in R^n$

Step 1: Normalize the each feature vector to zero mean and unit variance

Step 2: Compute the covariance matrix Sigma

$$\texttt{Sigma} = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)})(x^{(i)})^T$$

where m is the total number of samples.

Step 3: Compute the eigenvectors of Sigma, the covariance matrix.

Step 4: Multiply the data matrix with the matrix of first k eigenvectors.

Singular Value Decomposition (SVD)

In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix. It is the generalization of the eigendecomposition of a positive semidefinite normal matrix (for example, a symmetric matrix with positive eigenvalues) to any (displaystyle m times n) m times n matrix via an extension of the polar decomposition. It has many useful applications in signal processing and statistics.

The singular value decomposition can be computed using the following observations:

- The left-singular vectors of $\mathbf{M}$ are a set of orthonormal eigenvectors of $\mathbf{M}\mathbf{M}^*$.
- The right-singular vectors of $\mathbf{M}$ are a set of orthonormal eigenvectors of $\mathbf{M}^*\mathbf{M}$.
- The non-zero singular values of $\mathbf{M}$ (found on the diagonal entries of $\mathbf{\Sigma}$) are the square roots of the non-zero eigenvalues of both $\mathbf{M}^*\mathbf{M}$ and $\mathbf{M}\mathbf{M}^*$.

The columns of U and V are orthonormal bases :

Since **U** and $\mathbf{V}^*$ are unitary, the columns of each of them form a set of orthonormal vectors, which can be regarded as basis vectors. The matrix **M** maps the basis vector $\mathbf{V}_i$ to the stretched unit vector $\sigma_i \mathbf{U}_i$. By the definition of a unitary matrix, the same is true for their conjugate transposes $\mathbf{U}^*$ and **V**, except the geometric interpretation of the singular values as stretches is lost. In short, the columns of **U**, $\mathbf{U}^*$, **V**, and $\mathbf{V}^*$ are orthonormal bases.

$$\underset{m \times n}{\overset{X}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$

Further

Matrix U's columns will be the eigenvectors of $\mathbf{AA}^{\mathbf{T}}$.

Matrix V's columns will be the eigenvectors of $\mathbf{A}^{\mathbf{T}}\mathbf{A}$.

The advantage about this is that we can use it to compute eigenvector in PCA very efficiently without computing the covariance matrix.

To see how efficient this is let us take an example. If our data if 10000 dimensional then our covariance matrix will be of 10000 X 10000 dimensional which is huge when computing eigenvectors. When we use SVD we're dealing with 10000 dimensions.

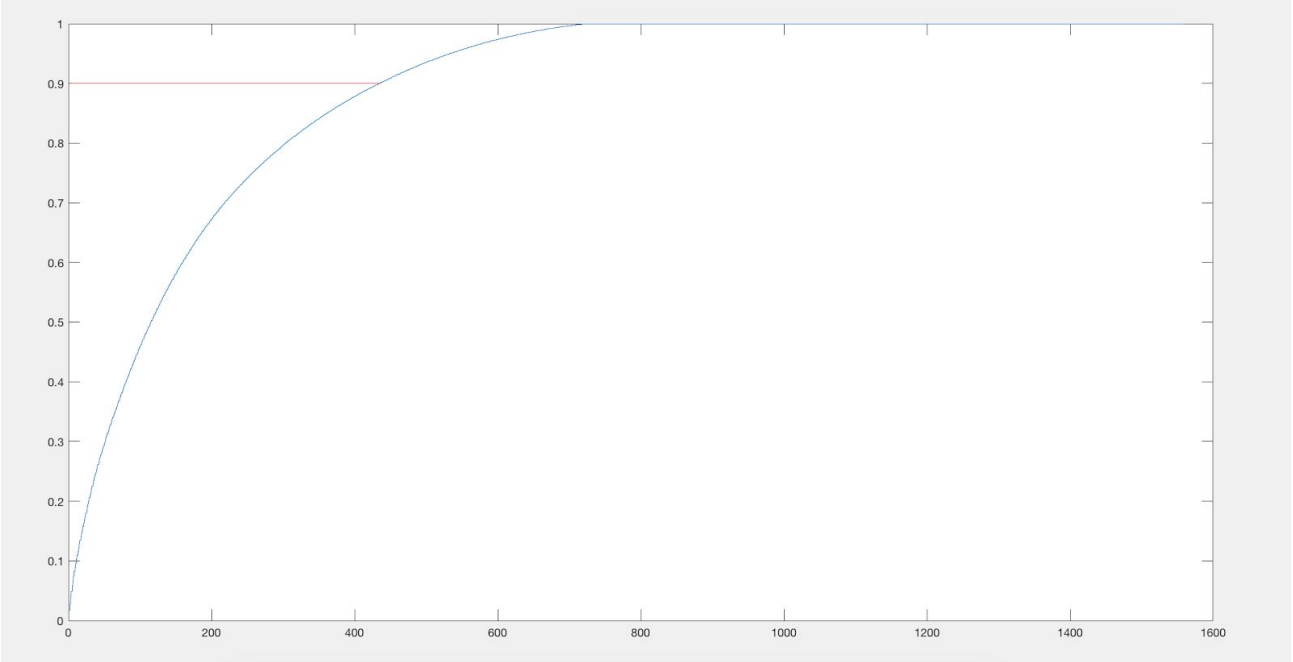
Applications of SVD :

Applications that employ the SVD include computing the

a. pseudo inverse
b. least squares fitting of data
c. multivariable control
d. matrix approximation
e. determining the rank, range and null space of a matrix.

How to choose the term k?

We now know that eigenvectors captures the variance of the data and more the eigenvalue, more the variance the corresponding eigenvector will capture. We now plot number of eigenvectors on the x-axis and the amount of variance captured on the y axis. Now, at what point in x-axis it crosses more than 0.9 in y-axis, i.e., for how many number of eigenvectors it captures more than 90% percent of variance. We'll take those many number of eigenvectors.

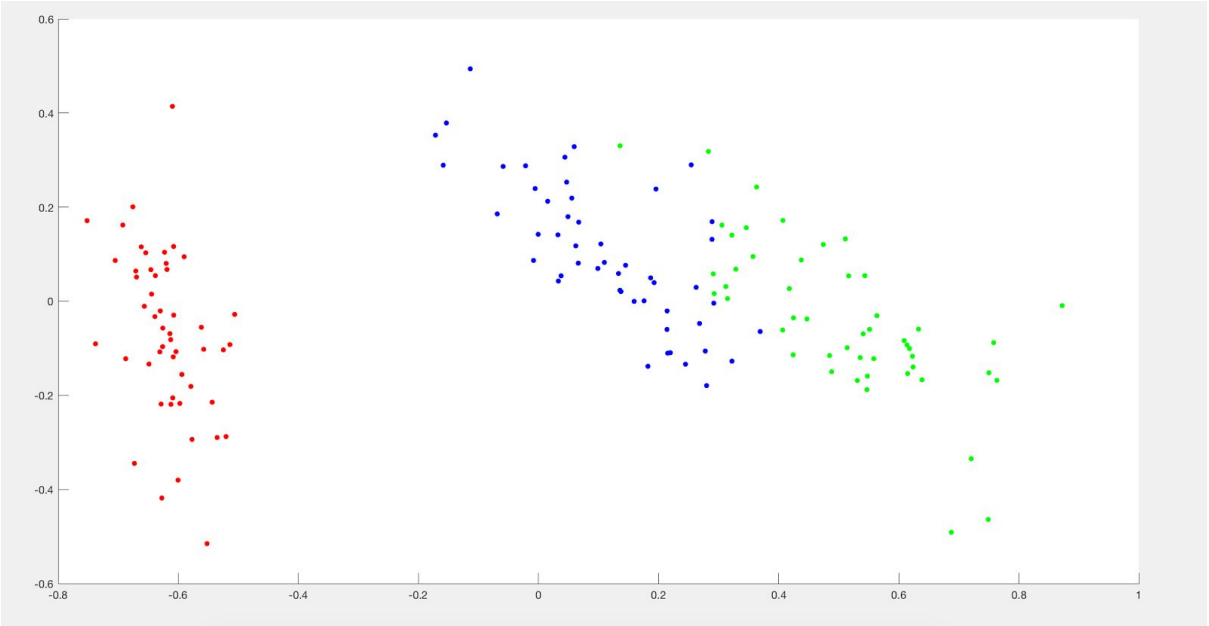


In the above figure it approximately corresponds to 435 eigenvectors.

Results:

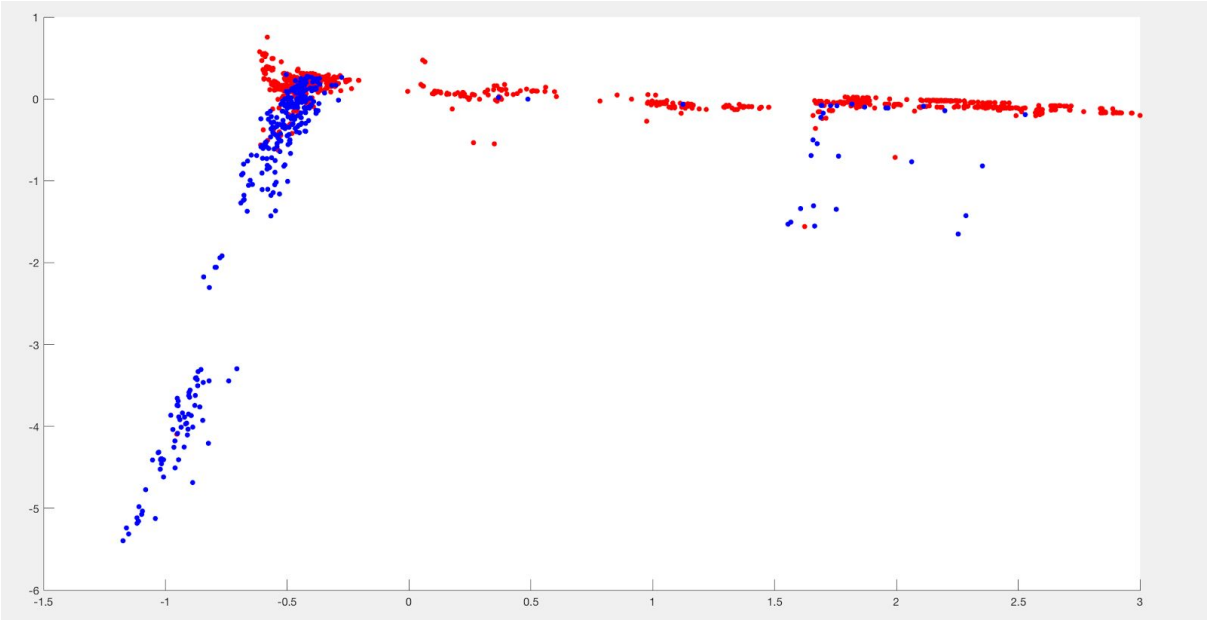
1) Visualisation:

a) Visualisation of iris dataset



Red dots => ‘setosta’
Blue dots => ‘versicolor’
Green dots => ‘virginica’

b) Visualisation of Internet Advertisement dataset



Blue dots => Ads
Red dots => Non-Ads

2) Accuracy when 90% of variance is captured

Dataset	Original dataset accuracy	Minimum number of features capturing 90% variance	Accuracy
Internet Advertisement	95%	435	96.12%
Arrhythmia	48%	139	53.76%

Remark: In both the datasets the percentage of accuracy has been increased when classified after applying PCA.

References

1. Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis." Chemometrics and intelligent laboratory systems 2.1-3 (1987): 37-52.