Computing for Data Sciences

Lecture 8

Principal Component Analysis

In machine learning or general theory of statistical learning, the problems are classified broadly into two types:-

1) Prediction problems: In this type of problems we take a training data set and try to predict its outcome at any given value. The output value of this type of data set should be continuous. These are solved by regression in general.
2) Classification problems: Classification problems have discrete output or Boolean variable. They are defined in sets. And we predict a set the given data will fall into. For example predicting whether real estate value of an asset will be more than or less than 50 lacs. This classification problem is generally solved with the help of classifiers. The classification problem may be easier if data has high variance i.e. variance within the group should be low and between the groups should be high.

Principal Component Analysis

Principal component analysis (PCA) is a multivariate technique that analyzes a data table in which observations are described by several inter-correlated quantitative dependent variables. Its goal is to extract the important information from the table, to represent it as a set of new orthogonal variables called principal components and to display the pattern of similarity of the observations and of the variables as points in maps.

A Toy Example

For understanding how the principal component analysis works, we will take an example of simple spring mass system from physics **(fig 1)** and will try to understand the motion of ball in x-direction. But as we are not physicist we don't understand the system. We as layman are trying to understand a system which has clouded and unclear data using some measuring quantities. We are so unfamiliar with the system that we do not see the underlying system that maybe quite simple. As we don't know the system, we try to gain the knowledge about the system by some experiment. This is a standard problem in physics in which the motion along the x direction is solved by an explicit function of time. In other words, the underlying dynamics can be expressed as a function of a single variable x.

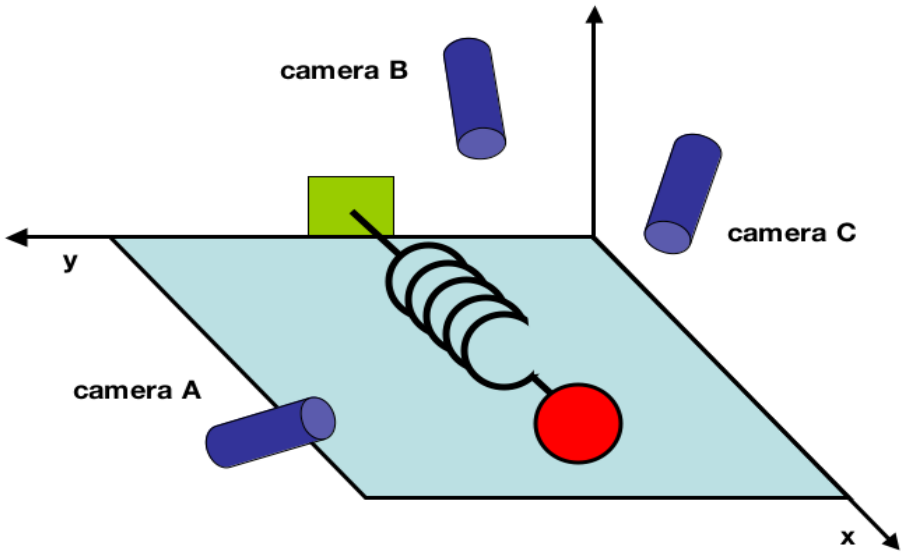

Fig. 1

Thus, we decide to measure the ball's position in a three-dimensional space and most simple experiment we can think for predicting the motion of mass is to put light source on the ball and see its projection or shadows, thereby reducing the 3-dimension motion of ball into 2-dimension. This is the basic notion that every experiment reduces the dimension of the actual data. No variable defines the complete data. That means the projection is not in right direction as we do not even know which x, y and z axes to place the light source which is arbitrarily placed and might not even be right angles. In this example we know the maximum number of light source that will suffice. As it is 3-dimensional space that maximum of three light sources, if it is not collinear, will be able to define the motion of the ball. But in practice we don't even know the given data set belong to which dimension. In general we don't have any control over no. of variable so we need to optimize over the variables to define the system and predict about it. Also they come with a lot of noise. PCA is used to achieve all this. This toy example is a practical one which we as layman are facing when doing an experiment or research.

The Goal:

Principal component analysis computes the most meaningful basis to re-express a noisy, garbled data set. The hope is that this new basis will filter out the noise and reveal hidden dynamics. In the example of the spring, the explicit goal of PCA is to determine: "the dynamics are along the x-axis." In other words, the goal of PCA is to determine that $\hat{x}$ - the unit basis vector along the x-axis - is the important dimension.

Solving PCA:

Even in the classification problem we would like a light source which makes the data as separate as possible, so that we have better chance of classifying. We require some structure in data. So we have to bring the lights in proper direction such that we could see proper projection of data and that must be as orthonormal as possible. The **fig 2** shows the types of different classifiers.

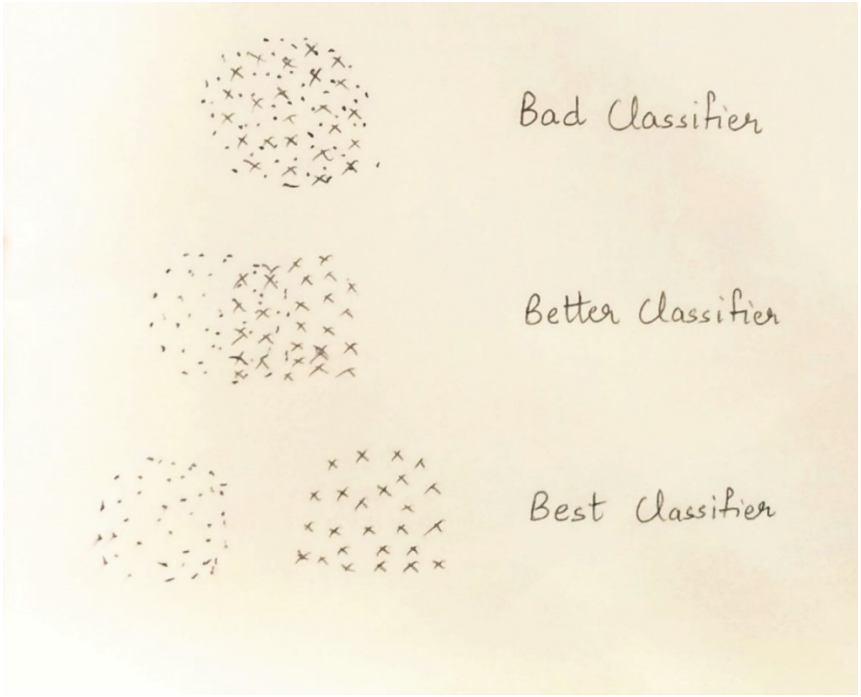

Fig. 2

Data just represents the system, it's not the system. We have no control over how the data is collected. So as the variables overlap we have to compute the covariance matrix.

Variance: $\mathbf{Var = \sum (x-\mu)^2}$

$= \vec{x}.\vec{x}$ **or** $\vec{x}^{\,T}.\vec{x}$

Covariance: $\mathbf{Cov} = \vec{x}^{\,T}.\vec{y} = \sum \vec{x}.\vec{y}$

Now we can define a new **m × n** matrix **X** which has m rows and n no. of variables.

$$\mathbf{X} = \begin{bmatrix} x_1^1 & \cdots & x_n^1 \\ \vdots & \ddots & \vdots \\ x_1^m & \cdots & x_n^m \end{bmatrix}$$

One interpretation of X is the following. Each row of X corresponds to all measurements of a particular type (x_i). Each column of X corresponds to a set of measurements from one particular trial.

$$\mathbf{Cov = X X^T}$$

The covariance matrix is m x m matrix and shows the interdependence between different explanatory variables. In covariance matrix the diagonal elements show variance. Best way to explain the output is to have zero covariance. In order to achieve this, one of the method is Eigen Vector Decomposition of covariance matrix.

In PCA we have to find some orthonormal matrix **P** where

$$\mathbf{Y = PX} \qquad \text{such that}$$

$$\mathbf{S_y\ (covariance)} = \frac{1}{n+1} \mathbf{*\ Y\ Y^T}$$

is diagonalized. The Diagonal values of this matrix will be equal to the rows of **P** and will be the principal components.

In the **fig 3** we can see the representation of co-variance matrix. Here the diagonal elements represent the variances of the variable.

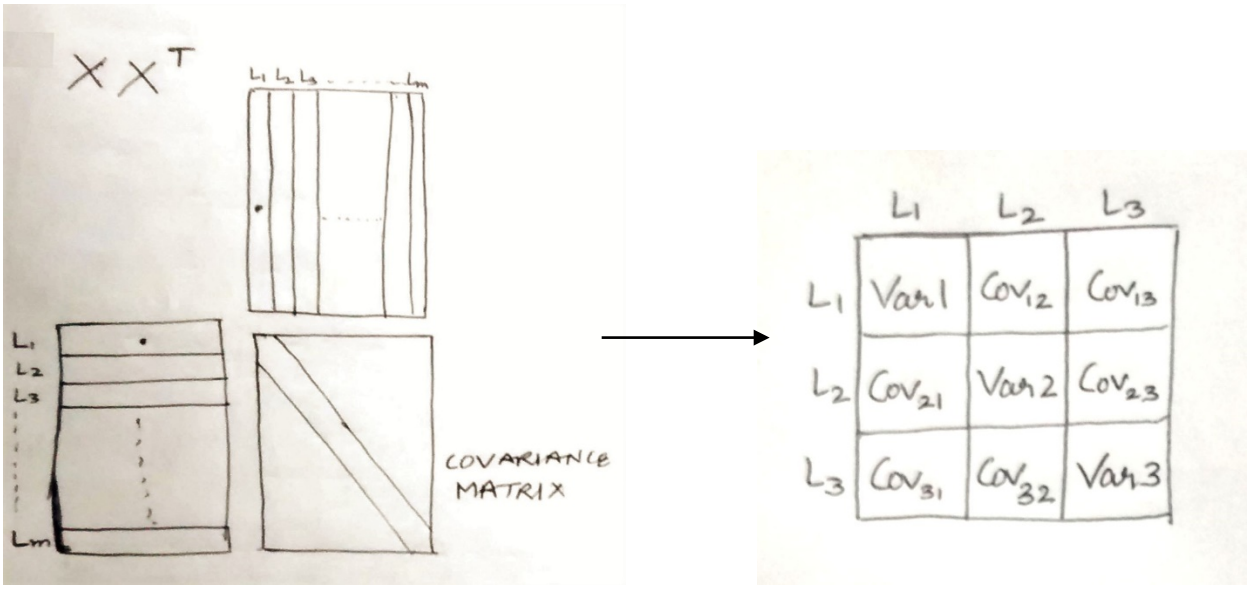

Fig. 3

We start by writing the $\mathbf{S_y}$ in terms of **P**

$$\mathbf{S_y} = \frac{1}{n+1} \mathbf{*\ Y.\ Y^T}$$

$$= \frac{1}{n+1} * (\mathbf{PX}).(\mathbf{PX})^T$$

$$= \frac{1}{n+1} * \mathbf{P}(\mathbf{X}\,\mathbf{X}^T)\,\mathbf{P}^T$$

$$= \frac{1}{n+1} * \mathbf{PAP}^T \text{ where } \mathbf{A} = \mathbf{XX}^T$$

Now, **A** is a symmetric matrix thus **A** can be arranged into $\mathbf{A} = \mathbf{EDE}^T$, where **D** is the diagonal matrix and **E** is Eigen vectors of **A** arranged as column. The matrix **A** has r x m orthonormal eigenvectors where r is the rank of the matrix. The rank of **A** is less than m when A is degenerate or all data occupy a subspace of dimension r x m. Maintaining the constraint of orthogonality, we can remedy this situation by selecting (m − r) additional orthonormal vectors to "fill up" the matrix **E**. These additional vectors do not affect the final solution because the variances associated with these directions are zero.

Now we select a P such that each row p_i is the Eigen vector of $\mathbf{XX}^T$ i.e. $\mathbf{P} = \mathbf{E}^T$. So

$$\mathbf{S_y} = \frac{1}{n+1} * \mathbf{PAP}^T$$

$$= \frac{1}{n+1} * \mathbf{P}(\mathbf{P}^T\mathbf{DP})\mathbf{P}^T$$

As **P** is a orthonormal matrix so

$$\mathbf{P}^T = \mathbf{P}^{-1}$$

$$= \frac{1}{n+1} * \mathbf{PP}^{-1}\mathbf{DPP}^{-1}$$

So,

$$\mathbf{S_y} = \frac{1}{n+1} * \mathbf{D}$$

It is evident that the choice of **P** diagonalizes $\mathbf{S_Y}$. This was the goal for PCA. We can summarize the results of PCA in the matrices **P** and $\mathbf{S_Y}$.

- The principal components of **X** are the eigenvectors of $\mathbf{XX}^T$; or the rows of **P**.
- The $\mathbf{i_{th}}$ diagonal value of $\mathbf{S_Y}$ is the variance of **X** along p_i.