CSE/ECE-471: Statistical Methods in AI **Spring 2020**

Principal Component Analysis (PCA)

Prepared by: Vijayraj Shanmugaraj(20171026), Amrit Sahai(2019201009), Anvesh Bakwad(2019201030)

1 Introduction

Principal Component is a statistical procedure used for feature selection. Say one has data in the form

$$(x_i, y_i),\ x_i \epsilon \mathbb{R}^D,\ y_i \epsilon \mathbb{R}^1$$

Where D is the number of dimensions of the original data. Instead of using all D dimensions, we would like to use a subset of the dimensions of the data for training and prediction purposes, since it gives both spatial and computational advantage, with maximum preservation of data during dimensionality reduction.

2 Notations used

$$\mu = \frac{1}{N}\sum_i x_i = E(X)$$

$$V = \frac{1}{K}\sum_i (x_i - \mu)^2 = Var(X)$$

Where

$$K = \begin{cases} N-1, & \text{for a sample} \\ N, & \text{for a population} \end{cases}$$

For representing the D-dimensional data, we introduce

$$X = \begin{bmatrix} X_1 & X_2 & X_3 & \dots & X_N \end{bmatrix}_{D\times N}$$

Where X_i is a $D \times 1$ column vector.

$$X = \begin{bmatrix} X_1^1 & X_2^1 & X_3^1 & \dots & X_N^1 \\ X_1^2 & X_2^2 & X_3^2 & \dots & X_N^2 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ X_1^D & X_2^D & X_3^D & \dots & X_N^D \end{bmatrix}$$

Covariance between two dimensions x and y is defined as:

$$Covariance = \frac{1}{N-1}\sum_i (x_i - \mu_x)(y_i - \mu_y)$$

Say

$$M = \begin{bmatrix} \mu & \mu & \mu & \dots & \mu \end{bmatrix}_{D\times N}$$

Where

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_D \end{bmatrix}_{D\times 1}$$

Where μ_i is the mean of the i^{th} dimension.

3 Motivation of PCA with an example

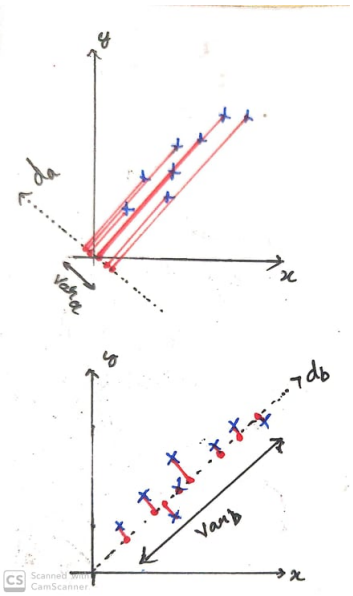

Figure 1: Projection of data and their variances of projection

Let us take a toy example of 2D data. We want to project these 2-Dimensional points onto a 1-Dimensional space. The 1-Dimensional Vector (which is the single principal component here) should be chosen in such a way that, it would have to account for the most variability possible, and hence, we search for the 'direction' that gives maximum variance of data, so that the chosen components (component in the 2D to 1D case) collects the most 'uniqueness' from the data set. We would project the data in direction d_b rather than d_a for maximum uniqueness/variance.

4 Mathematical derivation

Take

$$\tilde{X} = \left[X - M\right]$$

Now $\tilde{X}\tilde{X}^T$ is the **covariance matrix**. (Multiplying row i of $\tilde{X}$ with column j of $\tilde{X}^T$ will give us the covariance between the i$^{\text{th}}$ and j$^{\text{th}}$ dimension. This matrix will be a $D \times D$ symmetric matrix, since

$$(Cov_{(i,j)}) = (Cov_{(j,i)})$$

Say U is a 2D vector, which represents a particular direction. We want U such that

$$var(U_i^T X)$$

is maximized.

i.e. U is a dimension on which we want to project the data x_i so that we obtain a new representation z_i that has maximum variance.

It is easy to see that the mean of the original representation gets projected to the mean of the new representation.
(By $\bar{z}$, we imply E(z))

$$\bar{z} = \frac{1}{N}\sum_{i=1}^{N} z_i = \frac{1}{N}\sum_{i=1}^{N} U^T x_i = \frac{1}{N} U^T \sum_{i=1}^{N} x_i = U^T \mu$$

We are interested in finding a u that maximizes the variance after the projection. Hence, we want the following:

$$
\begin{aligned}
& argmax(\frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})^2) \\
=& argmax(\frac{1}{N}\sum_{i=1}^{N}(U^T x_i - U^T \mu)^2) \\
=& argmax(E[(U^T X - U^T \mu)(U^T X - U^T \mu)^T]) \\
=& argmax(U^T(E[(X-\mu)(X-\mu)^T]U) \\
=& argmax(U^T S U) \\
& \text{(Taking } E[(X-\mu)(X-\mu)^T] \text{ as S)}
\end{aligned}
$$

We need to maximise $U^T SU$. Since U is a direction vector,

$$norm(U) = 1$$

$$U^T U = 1$$

Using lagrangian transformation along with the above constraint, we get and try to maximise the expression:

$$U^T SU - \lambda(U^T U - 1)$$

Differentiating the above expression,

Note: $U^T SU$ is a scalar. We're differentiating this with respect to U_1 (a vector) to see how the scalar varies with respect to each component of the vector

$$2SU - 2\lambda(U) = 0$$

$$\boxed{SU = \lambda(U)}$$

Using the above result,

$$U^T SU = \lambda(U^T U) = \lambda$$

$$\text{(Since } U^T U = 1.\text{)}$$

Essentially, we are trying to maximise λ, which are clearly **eigenvectors of S, the covariance matrix**. Hence we can say that these eigenvalues are simply the coefficients attached to eigenvectors, which give the axes' magnitude. In this case, they are the measure of the data's covariance. By ranking the eigenvectors in order of their eigenvalues, highest to lowest, we get the principal components in order of significance for PCA. So, the dimension to project the data on, so as to maximize the variance is the eigenvector corresponding to the largest eigenvalue. The second best will be the second largest one and so on.

How many vectors to pick (Deciding k)?

We need to pick the vectors in the descending order of eigenvalues, and we pick k such that the first k eigenvalues satisfy the condition:

$$\frac{\sum_{i=1}^{k}\lambda_i}{\sum_i \lambda_i} \geq threshold$$

Often k is picked such that the threshold is set at 90 or 95%.

5 A brief on Singular Value Decomposition and its relation with PCA

Singular Value Decomposition (SVD), of a given Matrix, tells us how exactly can we decompose the Matrix, into its "Rotation" and "Scaling" Part. Applying SVD on the initial data matrix, we get:

$$\tilde{X} = U\Sigma\, V^T$$

Where U is the "final rotation vector", V^T is the "initial rotation vector", and Σ is the scaling factor.

and attempting to construct the covariance matrix from this decomposition gives

$$\tilde{X}\tilde{X}^T = (U\Sigma\, V^T)(U\Sigma\, V^T)^T = (U\Sigma\, V^T)(V\Sigma\, U^T) = U\Sigma\, (V^T V)\Sigma\, U^T$$

and since V is an orthogonal matrix, $V^T V = 1$

$$\tilde{X}\tilde{X}^T = U\Sigma^2 U^T$$

Which is equivalent to the PCA decomposition (the correspondence is easily visible).

In fact, using the SVD to perform PCA makes much better sense numerically than forming the covariance matrix to begin with, since the formation of $\tilde{X}\tilde{X}^T$ can apparently cause loss of precision. The SVD, is hence used to perform the eigenvector decomposition of the covariance matrix.

Note: The diagonals of the covariance matrix $\tilde{X}\tilde{X}^T$ correspond to the eigenvalues of the D eigenvectors we will get from the decomposition.

6 How is linear regression different from PCA?

Linear regression and PCA don't really have much in common, since PCA is a dimension reduction tool which yields principal components which are linear combinations of the variables, wheras linear regression is a method of assessing the linear relationship between a dependent variable and one or more independent vriables. It yields a formula describing that relationship and measures of its strength.

References:

Prof. Vineet Gandhi's slides for PCA
https://math.stackexchange.com/questions/3869/what-is-the-intuitive-relationship-between-svd-and-pca