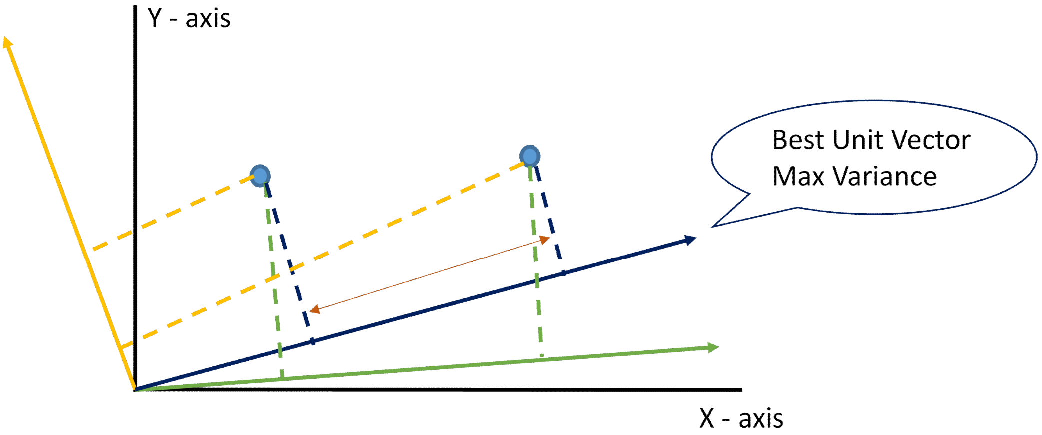
Principal Component Analysis

19 November 2023 11:20

Step to find the best component in PCA

1) Consider any point in plane and vector of that point on plane - a'
2) Consider a unit vector in same plane - u'
3) Try to project that point vector on unit vector
   a. Formula will became $\frac{\bar{a}\cdot\bar{u}}{|\bar{u}|}$
   b. Since this is unit vector the magnitude will be 1
   c. Therefor projection formula will became $\bar{a}\cdot\bar{u}$
   d. Finally we can write it as $u^T \cdot a$
   e. So this projection will be a scaler value means a number.

4) Now unit vector can be anywhere on plane but we will choose one that has maximum **variance**
5) We can take all the projections on unit vector and then can find the variance of all projections
   a. Consider projections -> $[u^T \cdot a_1]$, $[u^T \cdot a_2]$, $[u^T \cdot a_3]$ … … $[u^T \cdot a_\mathrm{n}]$
   b. We can calculate variance of these point as given below
   c. $\frac{1}{n}\sum_{i=1}^{n}(u^T \cdot a_i - u^T \cdot \bar{a})^2$
   d. So here we are going to find the **u**, where above given vector will be high.
   e.



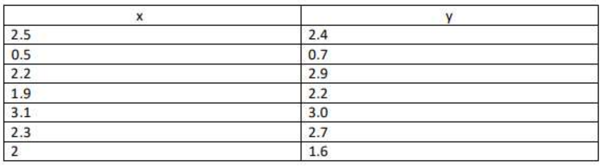
6) To find the best unit vector we need to know Covariance and Covariance Matrix
   a. Covariance is the measure of variance of two points on same plane
   b. Covariance matrix is used to describe the variance of data as well as the relation of two data (pos+ or neg-)
   c. Consider two vectors x1 and x2, the covariance matrix of these two will be
   d. $\begin{bmatrix} cov(x1,x1) & cov(x2,x1) \\ cov(x2,x1) & cov(x2,x2) \end{bmatrix}$
   e. As per variance the cov(x1,x1) = var(x1) and cov(x2,x1) = cov(x1,x2) so the matrix will be
   f. $\begin{bmatrix} var(x1) & cov(x2,x1) \\ cov(x1,x2) & var(x2) \end{bmatrix}$

7) Eigen Decomposition of covariance matrix
8) IMP ---- > Matrices are used to transform the coordinate system - Linear Transformation -> Rotate, Expand, Squeeze
9) Eigen vectors are the vectors that remain constant for direction, magnitude can be change
10) Eigen value - It is the stretch of value from in Eigen Vector / Difference of magnitude
11) For Eigen Vectors below equation will be always true

$$A\vec{v} = \lambda\vec{v}$$
$$where$$
$$A \rightarrow any\ vector$$
$$\vec{v} \rightarrow Eigen\ vector$$
$$\lambda \rightarrow Scalar\ Eigen\ Value$$

12) Eigenvalues and Eigenvectors: PCA uses Eigen decomposition of the covariance matrix in order to determine the principal components. Eigenvalues and Eigenvectors exist in pairs, i.e., every Eigenvector has a corresponding eigenvalue. For an n*n covariance matrix, we will have n Eigenvectors. Eigenvectors are used to understand variance (spread) in our dataset, i.e., in which variable we have more variance and the Eigenvector will be equal to the magnitude of that direction. If we sort the Eigenvalues in descending order, the Eigenvector associated with the first Eigenvalue gives us the first principal component (PC1), the second Eigenvector associated with the second Eigenvalue gives us the second principal component (PC2), and so on.
13) Some properties of these eigenvectors:
   a. Eigenvectors of the covariance matrix are always orthogonal (perpendicular) to each other, and the data is expressed in terms of these orthogonal Eigenvectors.
   b. When a linear transformation (multiply them with another vector) is performed on them, their direction does not change.
   c. We are only concerned with the direction of the vector and not the length. Hence the length of the Eigenvectors is set to 1 so that all eigenvectors will have the same length.

14) Methodology of Principal Component Analysis Algorithm:
   We will understand PCA by analysing a own designed dataset.



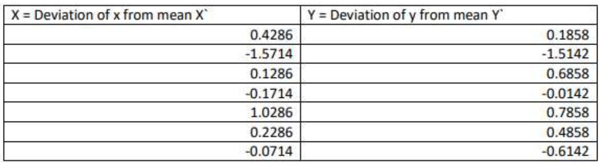
x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6

15) Data Scaling

Here, N = 7 Now we find the mean of both the features and subtract the mean from each of the data dimension

X` = (2.5 + 0.5 + 2.2 + 1.9 + 3.1 + 2.3 + 2)/7
= 14.5/7
= 2.0714

Y` = (2.4 + 0.7 + 2.9 + 2.2 + 3.0 + 2.7 + 1.6)/7
= 15.5/7
= 2.2142



X = Deviation of x from mean X`	Y = Deviation of y from mean Y`
0.4286	0.1858
-1.5714	-1.5142
0.1286	0.6858
-0.1714	-0.0142
1.0286	0.7858
0.2286	0.4858
-0.0714	-0.6142

16) Computing the covariance matrix

Now we will calculate the covariance matrix of the scaled data.

Cov= [0.63571429, 0.58547619
0.58547619, 0.67142857]

Since the data is 2 dimensional, the size of covariance matrix will be 2×2. Also,
both the features increase together as the non-diagonal elements in this covariance matrix are positive.

17) Calculating Eigenvalues and Eigenvectors

Since the covariance matrix is square, the eigenvectors and eigenvalues for the matrix can be calculated.

Eigenvectors of the covariance matrix are:

[[-0.7178043, -0.69624492],
[0.69624492, -0.7178043]]

Eigenvalues associated with the above Eigenvalues of the covariance matrix are:

[0.06782298, 1.23931988]

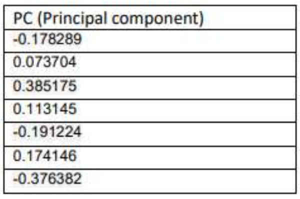
18) Computing the principal components

The eigenvector with the highest eigenvalue is the principle component of the data set.
1.23931988 > 0.06782298.
Hence Eigenvector [0.69624492, -0.7178043] which is associated with the Eigenvalue 1.23931988 will give us the First Principal Component.
We will explain this with the concept of Explained Variance. The explained variance ratio is the percentage of variance that is attributed by each of the selected components.
The explained variance of the two vectors is:
[0.948113568624776, 0.05188643137522321]
The sum of explained variances is always equal to 1.
The first two principal components account for around 94% of the variance in the dataset.

19) Deriving the new dataset

Now we derive our new dataset from the chosen components (Eigenvectors).
Final Data = Row Feature Vector × Row Data Adjust

Where,
Row Feature vector = matrix with the eigenvectors in the rows
Row Data Adjust = matrix with data items are in each column, with each row holding a separate dimension.



PC (Principal component)
-0.178289
0.073704
0.385175
0.113145
-0.191224
0.174146
-0.376382

As expected, it only has a single dimension. Therefore, we have successfully reduced our two dimensional dataset to one dimension.