3.1 Principal Component Analysis

The principal component analysis is used theoretically to extract spectrum and elemental maps by filtering noise present in SI or data cube. So, let us understand the statistical theory behind PCA step by step with examples [8]:

Statement : A statistical mathematical algebra technique that examines the interrelations among any data sets(EELS,EDX) of variables and identify the underlying structure of those variables from 'n'-dimensional to 2-dimensional by using diagonalization of the covariance matrix where eigen vectors represent spectra as a function of their associated eigen values and first few eigen vectors contains highest variance used for reconstruction by neglecting the eigen vectors with least variance contributing noise. The assumptions of PCA are [9] :

1. 'Linearity'-Assumes the data set to be linear combination of the variables
2. 'Mean &Covariance'-Not sure whether the directions of maximum variance exists with good features for discrimination.
3. 'Large variances have important dynamics'-Larger variance components exhibits greater dynamics while lower ones corresponds to noise.

Theory : Data cube or data matrix consists of several pixels as many as 100*100 pixels each pixel in hyperspectral image or SI contains spatial positions of x, y directions with STEM probe intensity in Z direction which gives spectrum in each pixel. As a result, there is high data to find interrelation between the variables. PCA technique can clearly suit for this data analysis and can significantly improve the signal to noise ratio. The mathematical expression for PCA is

$$A_{((x,y),E)} = UV^T \dots\dots\dots\dots\dots\dots.(15)$$

Where A is an original data set, U is m*n score matrix and V is an n*n loading matrix. Matrices T and V can be found instance from the Singular value decomposition algorithm of A: [10]

$$\boldsymbol{A} = \boldsymbol{U\,\Sigma V^T}...............(16)$$

In EELS, we unwrap two spatial dimensions to obtain data matrix A with dimensions $p{\times}N_E$ where p is the total amount of spatial positions and N_E is the amount of pixels in each spectrum and T=UΣ, Σ is the diagonal matrix which gives the total variance of 'A' which is known as 'principal component' and is most important for avoiding the noise in data set. U &Vis formed with the cross product of $A*A^T$ and $A^T* A$. Let's understand this whole SVD algorithm step by step with an example of 3d matrix.

Let A be 3*3 matrix with rank3, $A = \begin{bmatrix} 1 & 5 & 3 \\ 2 & 7 & 9 \\ 4 & 8 & 6 \end{bmatrix}_{3*3}$

Now apply equation 16 where we can decompose matrix into three different forms and extract highest variance.

Step-1: $U = \frac{1}{\sigma} A * v_i \Longrightarrow A * A^T = \begin{bmatrix} 35 & 64 & 62 \\ 64 & 134 & 118 \\ 62 & 118 & 116 \end{bmatrix}$

Step-2: calculate eigen vectors for $A*A^T$ with $|A * A^T - \lambda I| = 0 \dots\dots(17)$ where I is identity matrix.

$$\begin{bmatrix} (35-\lambda) & 64 & 62 \\ 64 & (134-\lambda) & 118 \\ 62 & 118 & (116-\lambda) \end{bmatrix} = 0$$

Therefore, we will get an equation from step 2 as ***(λ^3-285λ^2+2430λ-2916=0)---(18)*** and this equation requires 'Newton Raphson method' where we get iterations and repeated value as outcome. The eigen values of matrix A are;

λ = 1.44297, 7.31547, 276.24156 now compute this values in eq.17 to get eigen vectors(v) for each eigen value(λ) in descending order and we get,

For λ1=276.24156 $\Longrightarrow v_1 = \begin{bmatrix} 0.54175 \\ 1.07333 \\ 1 \end{bmatrix}; \lambda 2 = 7.31547 \Rightarrow v_2 = \begin{bmatrix} 0.51366 \\ -1.19094 \\ 1 \end{bmatrix}$ and

$$\lambda 3 = 1.44297 \Rightarrow v_3 = \begin{bmatrix} -1.89238 \\ 0.02348 \\ 1 \end{bmatrix}$$

calculate the length of each eigen vector using $L = \sqrt{(\Sigma x_n^2)}$ and normalize them by means of individual vector with

$$u_1 = \left(\frac{0.54175}{1.56382}, \frac{1.07333}{1.56382}, \frac{1}{1.56382}\right) = (0.34643, 0.68635, 0.63946)$$

and similarly, u_2 and u_3 which gives,

$$U = \begin{bmatrix} 0.34643 & 0.68635 & 0.63946 \\ 0.68635 & 0.72718 & 0.01098 \\ 0.66503 & 0.71733 & 0.20775 \end{bmatrix}_{3*3}$$

and alternatively, V is calculated in same method that has applied for calculating U.

$$V = \begin{bmatrix} 0.25733 & 0.48126 & 0.83795 \\ 0.70108 & 0.50381 & -0.50466 \\ 0.66503 & -0.71734 & 0.20774 \end{bmatrix}_{3*3}$$

and now we calculate 'Σ' the diagonal matrix or correlation matrix which gives the variance of A. 'Σ' can be calculated from U&V eigen values in descending order with their mean squared roots which, gives the same result compared from U and V:

$$\Sigma = \begin{bmatrix} \sqrt{\lambda_{pc1}} & 0 & 0 \\ 0 & \sqrt{\lambda_{pc2}} & 0 \\ 0 & 0 & \sqrt{\lambda_{pc3}} \end{bmatrix} where\ eigen\ values of\left(\sqrt{\lambda_{pc1}} > \sqrt{\lambda_{pc2}} > \sqrt{\lambda_{pc3}}\right)\ldots(19)$$

and each one of the vector in diagonal matrix is single principal component. From U, eigen values are computed in Σ

$$\Sigma = \begin{bmatrix} \sqrt{276.24156} & 0 & 0 \\ 0 & \sqrt{7.31547} & 0 \\ 0 & 0 & \sqrt{1.44297} \end{bmatrix} \Rightarrow \begin{bmatrix} 16.62052 & 0 & 0 \\ 0 & 2.70471 & 0 \\ 0 & 0 & 1.20124 \end{bmatrix}$$

From 'Σ' we have first two vectors in diagonal matrix correspond to most of the variance in the matrix A. Hence, we can neglect the remaining principal components and avoid noise which is the notion implemented in spectrum extraction. PC's 1&2 have the 95% of dynamics which can explain A. By using PC1 and PC2 we can reduce the 3dimensional data to 2d dimensionality maintaining all the dynamics of initial data. So, lets reconstruct by using PC1 and PC2.

Step-3: Reconstruction

$$U\Sigma V^T = \begin{bmatrix} 0.3464 & 0.3136 \\ 0.6864 & -0.7272 \end{bmatrix} * \begin{bmatrix} 16.62052 & 0 \\ 0 & 2.70471 \end{bmatrix} * \begin{bmatrix} 0.2573 & 0.7011 \\ 0.4813 & 0.5038 \end{bmatrix}$$

Therefore, we got $\begin{bmatrix} 1 & 5 \\ 2 & 7 \end{bmatrix}_{2*2}$ as reconstructed which is equal to two dimensional matrix of A. If we want the full original A add all the PC's and vectors of U&V. Hence, the above explained theory is applied for SI for filtering noise and getting clear spectrum. Extracting number of principal components depends upon the dimensionality of data sets. To check how much percentage of A data is accounted in principal components or estimate how much of A is accountable in principal component sets we use;

$$Trace\ of\ (A)\% = \left(\Sigma \frac{Selected\ sum\ of\ PC's}{Total\ sum\ of\ all\ PC's}\right) \times 100 \ldots\ldots (20)$$

From the obtained PC's of matrix, A, we apply eq.20 and check how first two PC's are consisting data% of A:

$Trace\ of\ A\% = \frac{16.62052+2.70471}{16.62052+2.7047+1.20124} \times 100 \Rightarrow 94.14\% \cong 95\%$ of data from A, by reducing 3d to 2d and these applies for multidimensional data sets too where the final consideration of highest variance of dynamics is found in first few principal components and the rest would be noise of the SI.

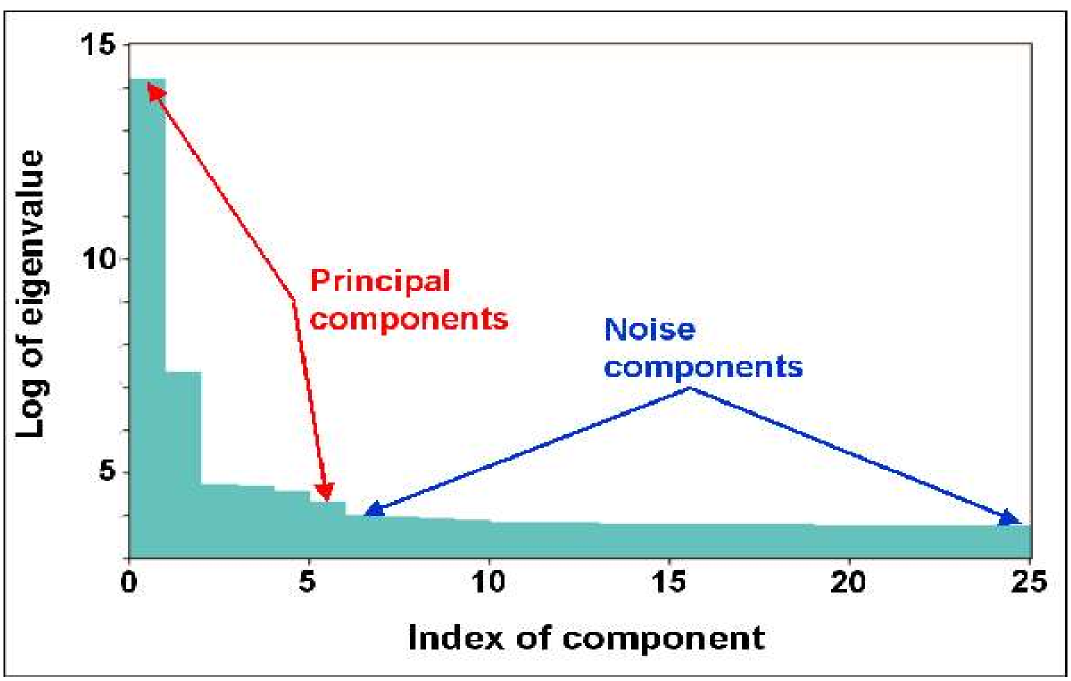

***Figure 19 An example of the scree plot generated after PCA decomposition in the PCA plugin** [11].*