# Principal Component Analysis

## Student Training Program on AI/ML at IIIT Hyderabad – May 2025

**Dr. Monalisa Patra**

Higher-dimensional data refers to data sets that have a large number of features or attributes, typically represented as columns in a table or matrix. This can include data from a variety of sources, such as images, audio, text, or sensor data, where each data point can have thousands or even millions of dimensions. Dealing with higher-dimensional data requires specialized techniques and algorithms that can handle the increased complexity and size of the data. Note that as the number of dimensions increases, the amount of data required to generalize accurately grows exponentially. This makes it harder to obtain a representative sample, thereby leading to the term curse of dimensionality. Moreover, it can be challenging to visualize data in more than three dimensions. This can make it harder to interpret and understand the data, and can also make it harder to identify patterns and correlations. Overall, higher-dimensional data can be more challenging to work with in machine learning than lower-dimensional data. However, with appropriate techniques such as dimensionality reduction, feature selection, and feature extraction, it is possible to overcome many of these challenges and build accurate and robust models. Dimensionality reduction is a process of reducing the number of features or attributes in the data while retaining as much information as possible. Dimensionality reduction can lead to improved model performance, faster training times, reduced storage requirements, and improved interpretability of the data.

Feature extraction is a process of transforming the original features or attributes of the data into a new set of features that capture the most relevant information for the problem at hand. This is done with the aim to reduce the number of features in the dataset, with the new reduced set of features summarizing most of the information contained in the original set of features. Considering we have 4 features in the dataset, $(x_1, x_2, x_3, x_4)$ feature extraction leads to the creation of two new features, capturing most of the information of the dataset, and can be summarized as:

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 1.2 & 1.0 & 0.5 & 0.7 \\ 2.1 & 0.3 & 1.7 & 0.2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

The transformation of the original features into a lower-dimensional space is done using methods such as principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), or autoencoders. These techniques can be divided into linear and nonlinear methods, depending on whether they preserve linear or nonlinear relationships between the variables. The principal component analysis is a popular technique for dimensionality reduction and data compression. It is used to transform a dataset of high-dimensional variables into a smaller set of uncorrelated variables called principal components, which capture most of the variance in the original dataset. Screenshots of slides:

# Summary

- Given a set of points, how do we know that they can be expressed like the example before?
  - We need to look at the correlation between points.
- How do we find the lines to keep / discard?
  - The tool we use is PCA (**Linear transformation techniques** by finding **orthogonal axes** that capture the most **variance.)**.
  - The axes are obtained by Eigen Analysis of the Covariance Matrix of the data.
  - Another approach is to do Singular Vector Decomposition of data matrix.
  - Either approach will give us the "best" directions to project the data to.
  - In general, the projection is to a lower dimensional sub-space.



# Covariance

- $\mathbf{cov(X, Y) = cov(Y, X)}$
- $n$- dimensional data will result in $n \times n$ covariance matrix.

$$M = \begin{bmatrix} 5 & 3 & 1 \\ 1 & 4 & 5 \\ 6 & 8 & 3 \end{bmatrix} \xrightarrow{\text{Covariance}} \begin{bmatrix} 4.67 & cov(1,2) & cov(1,3) \\ cov(2,1) & 4.67 & cov(2,3) \\ cov(3,1) & cov(3,2) & 2.67 \end{bmatrix} \rightarrow \begin{bmatrix} 4.67 & 2.33 & -2.67 \\ 2.33 & 4.67 & 0.67 \\ -2.67 & 0.67 & 2.67 \end{bmatrix}$$

$$cov(X,Y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N}$$

$$X = \begin{bmatrix} 5 \\ 1 \\ 6 \end{bmatrix}, \quad Y = \begin{bmatrix} 3 \\ 4 \\ 8 \end{bmatrix} \qquad \bar{x} = 4, \quad \bar{y} = 5$$

$$cov(X,Y) = \frac{(5-4)(3-5) + (1-4)(4-5) + (6-4)(8-5)}{3}$$

$$= 2.33$$

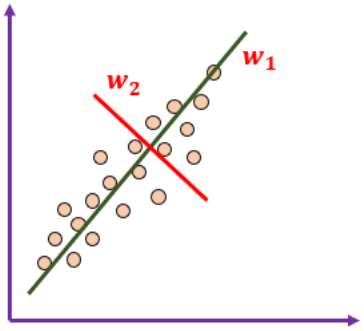
# Interpretation of Covariance calculations

- Say we have a 2-dimensional dataset.
  - x: number of hours studied for a subject, y: marks obtained in that subject
  - Covariance value is say: 108.45
- What does this value mean?
- Exact value is not as important as it's sign.
- A **positive value** of covariance indicates **both dimensions increase or decrease together** e.g. as the number of hours studied increases, the marks in that subject increase.
- A **negative value** indicates **while one increases the other decreases**, or vice-versa e.g. active social life at university vs performance in exams.
- If **covariance is zero** : the two dimensions are **independent** of each other e.g. heights of students vs the marks obtained in a subject.



# Mathematical Setup

- $\boldsymbol{X} \in \mathbb{R}^{N \times n}$: centered data matrix
- $\boldsymbol{Q} = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}$: the covariance matrix
- $\boldsymbol{w} \in \mathbb{R}^{n \times 1}$: a unit vector (i.e., $\|\boldsymbol{w}\| = 1$)



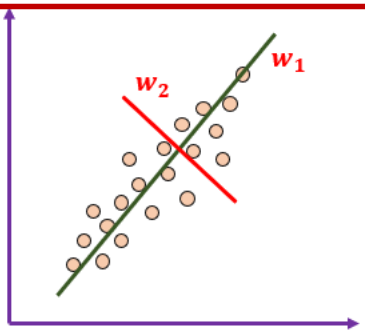

- We want to **project** the data onto $\boldsymbol{w}$:
  $z_i = x_i.\boldsymbol{w}$, for $i = 1,2,\cdots,N$
  ($z_i$ is **the scalar projection** of $\boldsymbol{x}_i$ onto $\boldsymbol{w}$)
- We get the 1D data vector $z = \boldsymbol{X}\,\boldsymbol{w} \in \mathbb{R}^{N \times 1}$
- What is the variance of these projected values?
- The mean of $z$ is $\bar{z} = \frac{1}{N}\sum_{i=1}^{N} z_i = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i.\boldsymbol{w} = \left(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i\right).\boldsymbol{w} = \boldsymbol{0}.\boldsymbol{w} = \boldsymbol{0}$
- Because $\boldsymbol{X}$ is centered, the sample mean is zero $\Rightarrow \bar{z} = 0$

# Mathematical Setup

$$\mathbf{Var}(z) = \frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})^2 = \frac{1}{N}\sum_{i=1}^{N} z_i^2 = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x_i}.\boldsymbol{w})^2$$

$$\mathbf{Var}(z) = \boldsymbol{w}^T\left(\frac{1}{N}\sum_{i=1}^{N} x_i^T\, x_i\right)\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{Q}\,\boldsymbol{w}$$

The eigenvectors of the covariance matrix give you the direction that maximizes the variance.




- This is a **quadratic form** and is maximized with respect to $\boldsymbol{w}$ under the constraint $\|w\| = 1$ (maximize the variance).
- We want to maximize: $L(\boldsymbol{w}, \lambda) = \boldsymbol{w}^T\boldsymbol{Q}\,\boldsymbol{w} - \lambda(\boldsymbol{w}^T\boldsymbol{w} - \mathbf{1})$
- Set derivatives to zero: $\nabla_w L = 2\,Q\,\boldsymbol{w} - 2\,\lambda\,\boldsymbol{w} = 0 \Rightarrow Q\,\boldsymbol{w} = \lambda\,\boldsymbol{w}$

**This is the eigenvalue equation!**

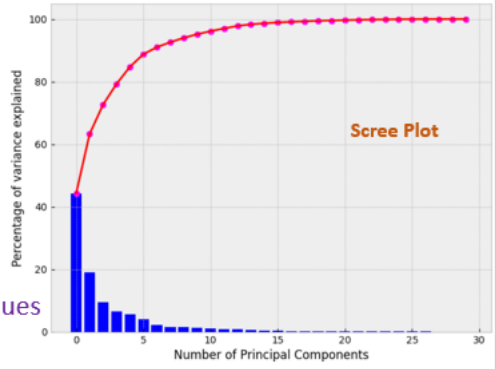


# How many PCs

- A dataset with **N samples** and **n features**, will give rise to a $\boldsymbol{n} \times \boldsymbol{n}$ covariance matrix.
- The $\boldsymbol{n} \times \boldsymbol{n}$ covariance matrix will have $\boldsymbol{n}$ eigenvectors, so $\boldsymbol{n}$ PCs.

**n features** → **n principal components**

- Where does dimensionality reduction come from?
- Can always ignore the components of lesser significance.




Note: You do lose some information, but if the eigenvalues are small, you don't lose much

## Singular Value Decomposition (SVD)

- SVD gives the decomposition for any arbitrary matrix, $M = U \Lambda V^T$

$$M_{N\times n} = U_{N\times r}\, \Lambda_{r\times r}\, V^T_{r\times n}$$

✓ $\Lambda$ is the diagonal matrix equal to the root of the positive eigenvalues of $M$

✓ U and V are the orthogonal matrices, $U^T U = 1, V^T V = 1$

✓ U consists of orthonormal eigenvectors of $M$

✓ V consists of orthonormal eigenvectors of $M^T$



## Singular Value Decomposition (SVD)

- The SVD of the centered data matrix, $\mathbf{X} = U \Lambda V^T$
- After standardization, the covariance matrix of the data matrix, $\sum = \frac{1}{N} X^T X$

$$\sum = \frac{1}{N} X^T X = \frac{1}{N}(U \Lambda V^T)^T (U \Lambda V^T) = \frac{1}{N}(V \Lambda^T U^T)(U \Lambda V^T)$$

$$= \frac{1}{N}(V \Lambda^T \Lambda V^T) = \frac{1}{N}(V\ (\Lambda)^2\ V^T)$$

- $(\Lambda)^2$ is a diagonal matrix whose entries are $\Lambda_{ii} = \lambda_i^2$, the squares of the eigenvalues of the SVD of $X$.
- Both $X$ and $X^T X$ share the same eigenvectors in their SVD.

➢ **We can run SVD on $X$ without ever instantiating the large $X^T X$ to obtain the necessary principal components more efficiently**



Important Links:

https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

https://dimensionless.in/principal-component-analysis-in-r/

https://dataknowsall.com/blog/imagepca.html