Martin Mundt, Dr. Iuliia Pliushch, Prof. Dr. Visvanathan Ramesh

Pattern Analysis & Machine Intelligence Praktikum: MLPR-WS19

Week 8: K-Means and PCA

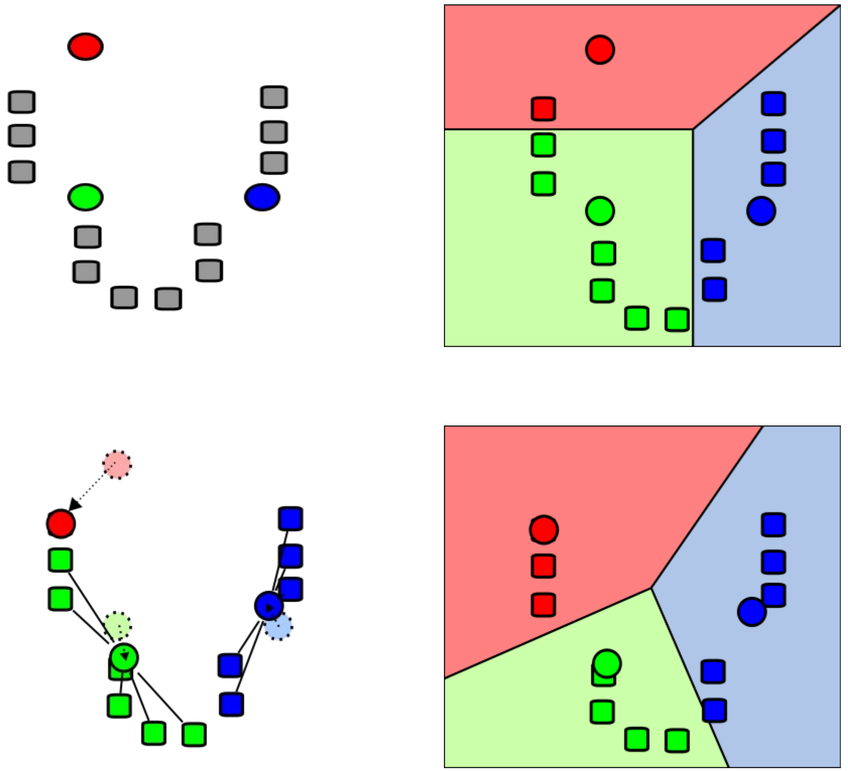
Unsupervised learning: K-Means and PCA

- **K-means**: clustering algorithm which operates on the distances between points and the supposed cluster centers

- **PCA**: helps in analyzing the data variance and can be used for **data compression**

K-Means clustering (Lloyd algorithm)

- Input: d-dimensional data points
- Randomly initialize k cluster means
- Assign points to its closest cluster mean
- Update the cluster means and repeat the two previous steps until the means converge



https://de.coursera.org/lecture/genomic-data/the-lloyd-algorithm-for-k-means-clustering-3O9eh

https://de.wikipedia.org/wiki/K-Means-Algorithmus

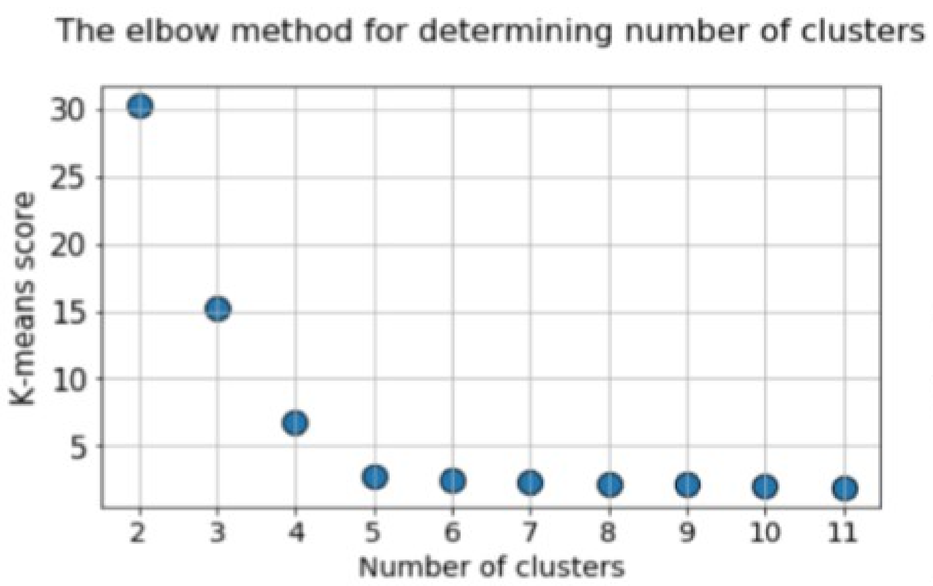
K-Means: Difficulties

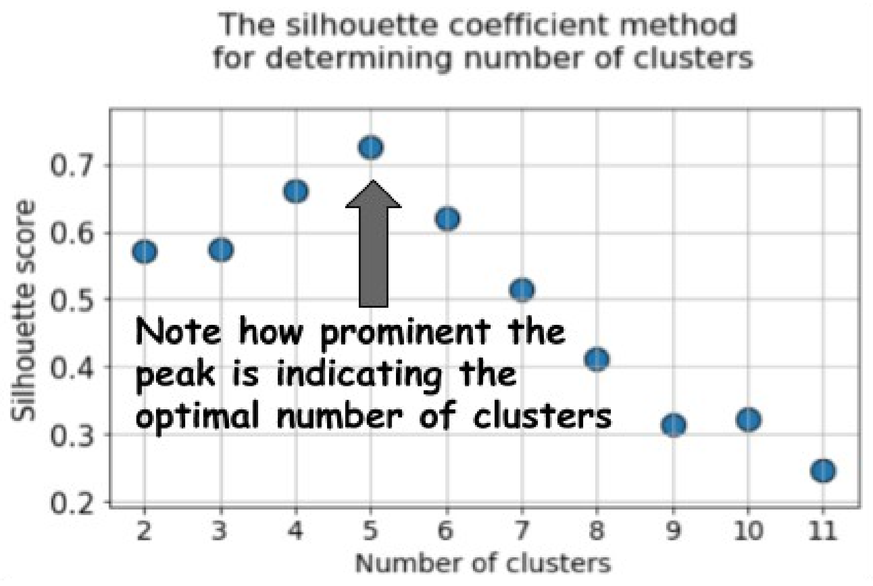
- How do we determine **k –** the number of clusters to split the data into?

Elbow	Silhouette
- run k-means for different values of k - calculate **WCSS**: within cluster sum of squares - plot WCSS for growing k - take the k where ‘the elbow bends’	- run k-means for different values of k - calculate the **average silhouette** - plot the measure for growing k - take the k at the peak

K-Means: Difficulties (1)

- How do we determine **k –** the number of clusters to split the data into?

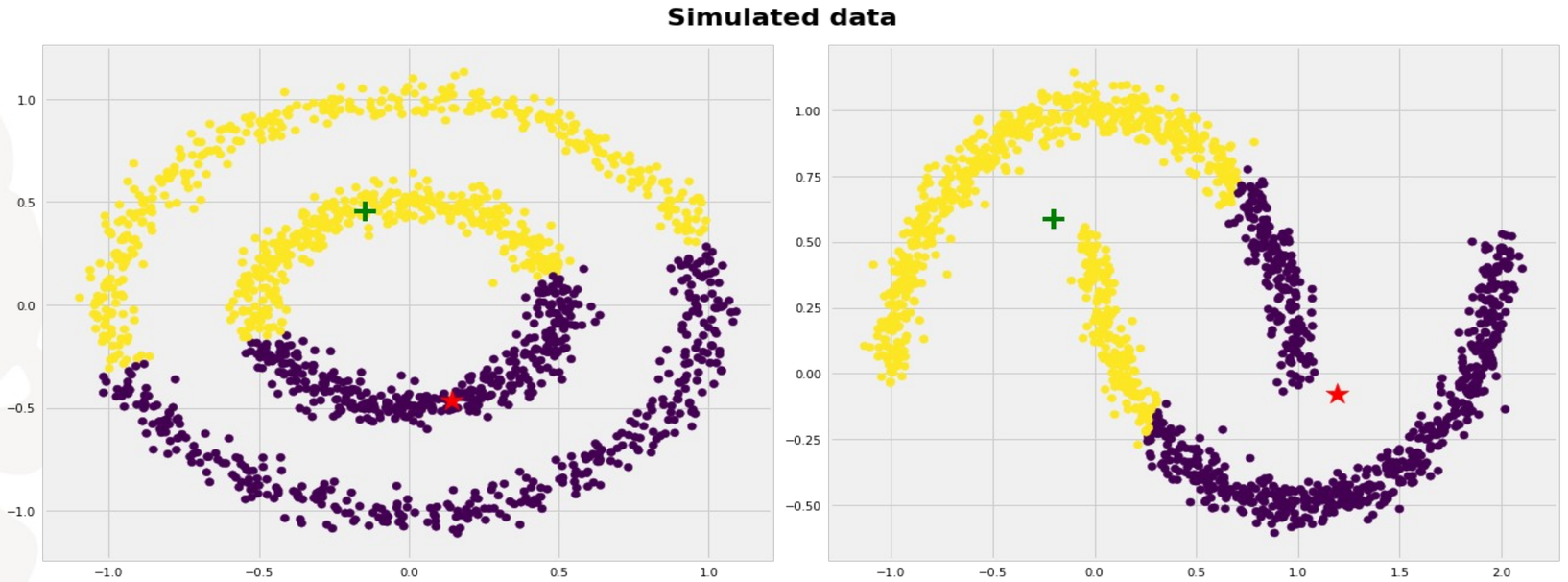




https://towardsdatascience.com/clustering-metrics-better-than-the-elbow-method-6926e1f723a6

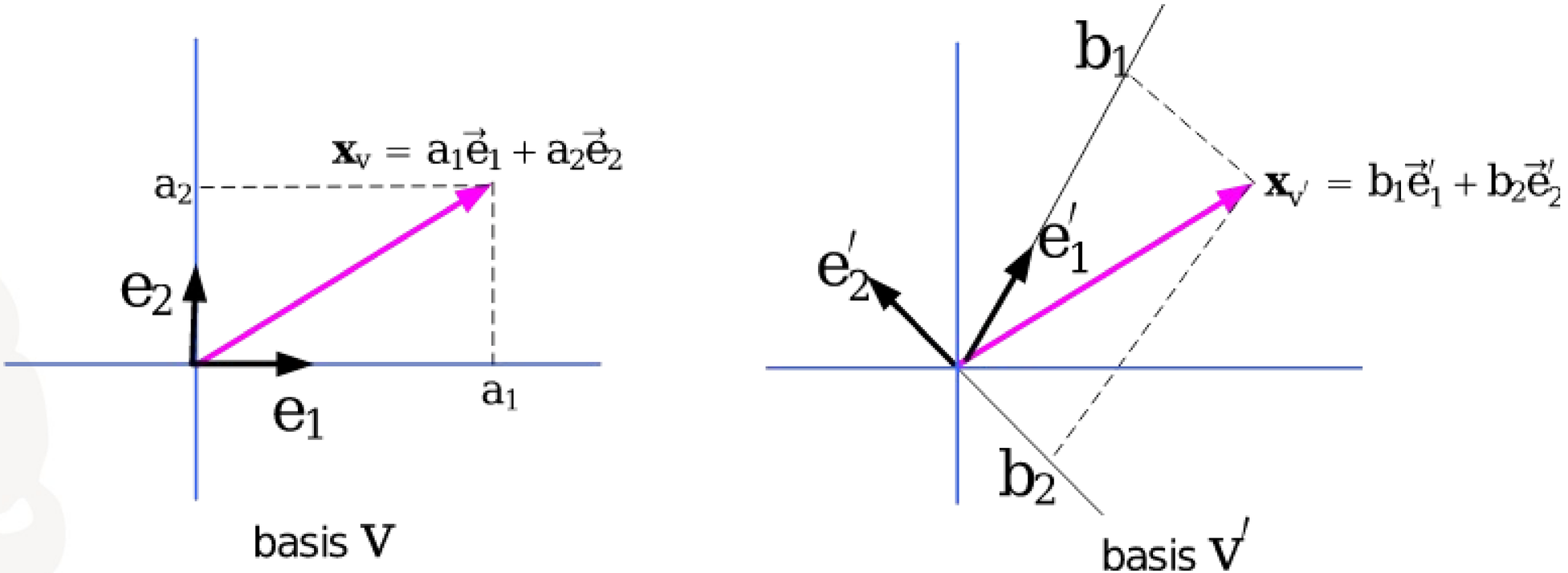
K-Means: Difficulties (2)

- What if the cluster are not of a spherical shape?



https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

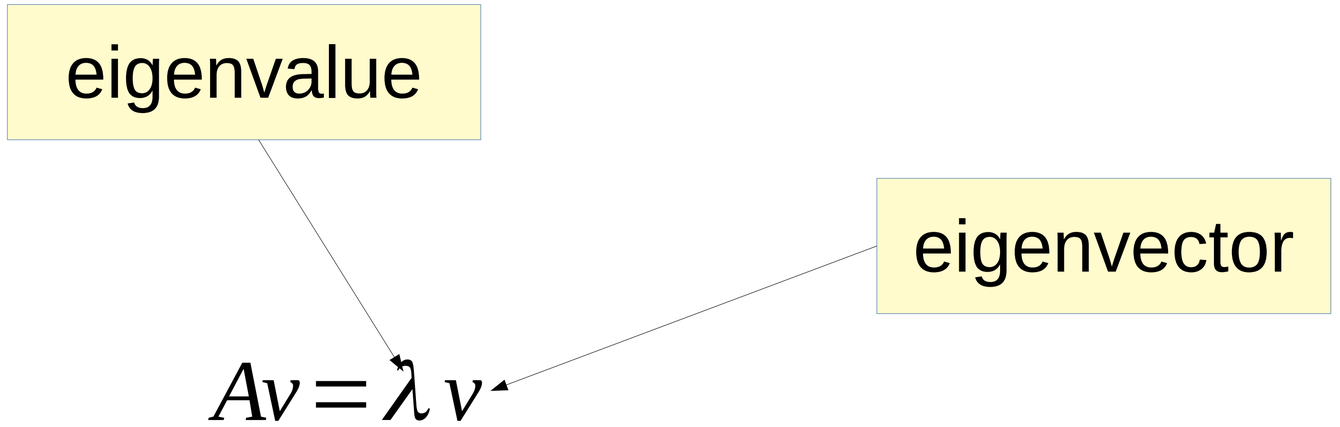
PCA: Basis transformation



The same vector having different representation depending on basis used

https://www.12000.org/my_notes/similarity_transformation_and_SVD/index.htm

PCA: Eigenvalues & eigenvectors

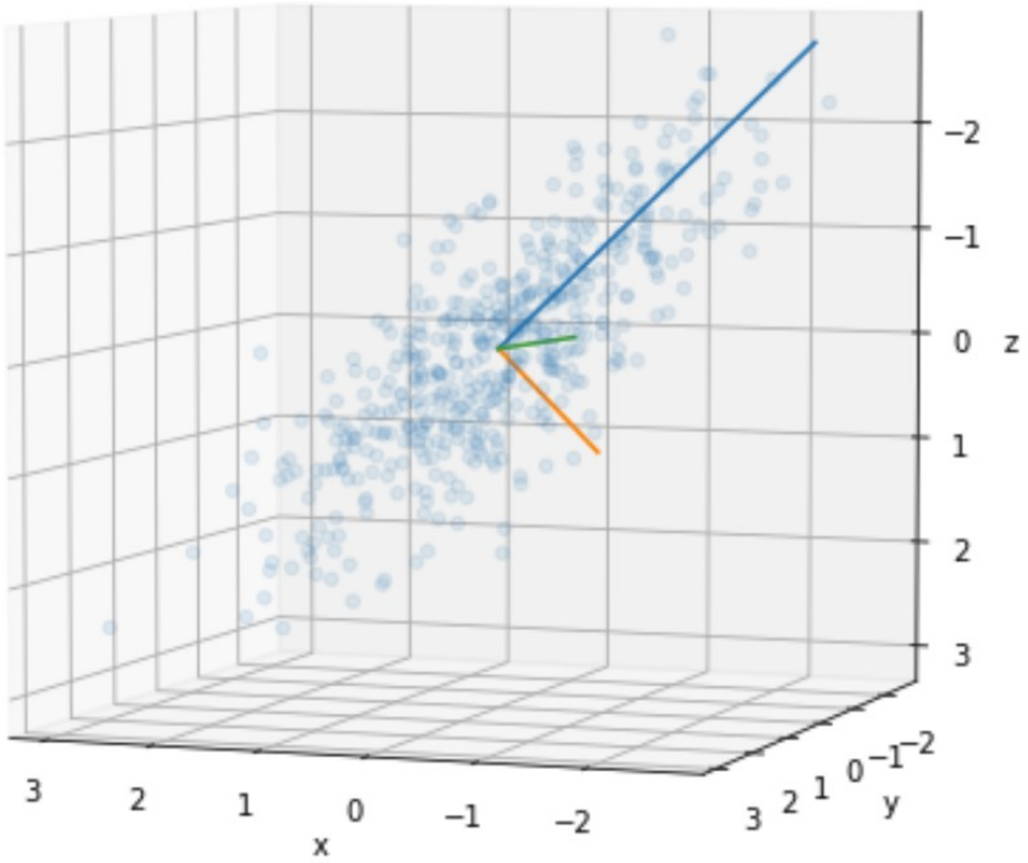


Interpretation: the eigenvector v does not change (its direction) when multiplied by A, it is only scaled.

PCA – Principal Component Analysis

- Input: d-dimensional data

- Subtract the mean from your data
- Compute the covariance matrix for your zero-mean data
- Compute the eigenvalues and eigenvectors of the **covariance matrix**
- Sort the **eigenvectors** (=principal components) in descending order according to the eigenvalues
- Pick a subset of them and transform your data



http://www.iro.umontreal.ca/~pift6080/H09/documents/papers/pca_tutorial.pdf

Covariance matrix

- **Variance:**

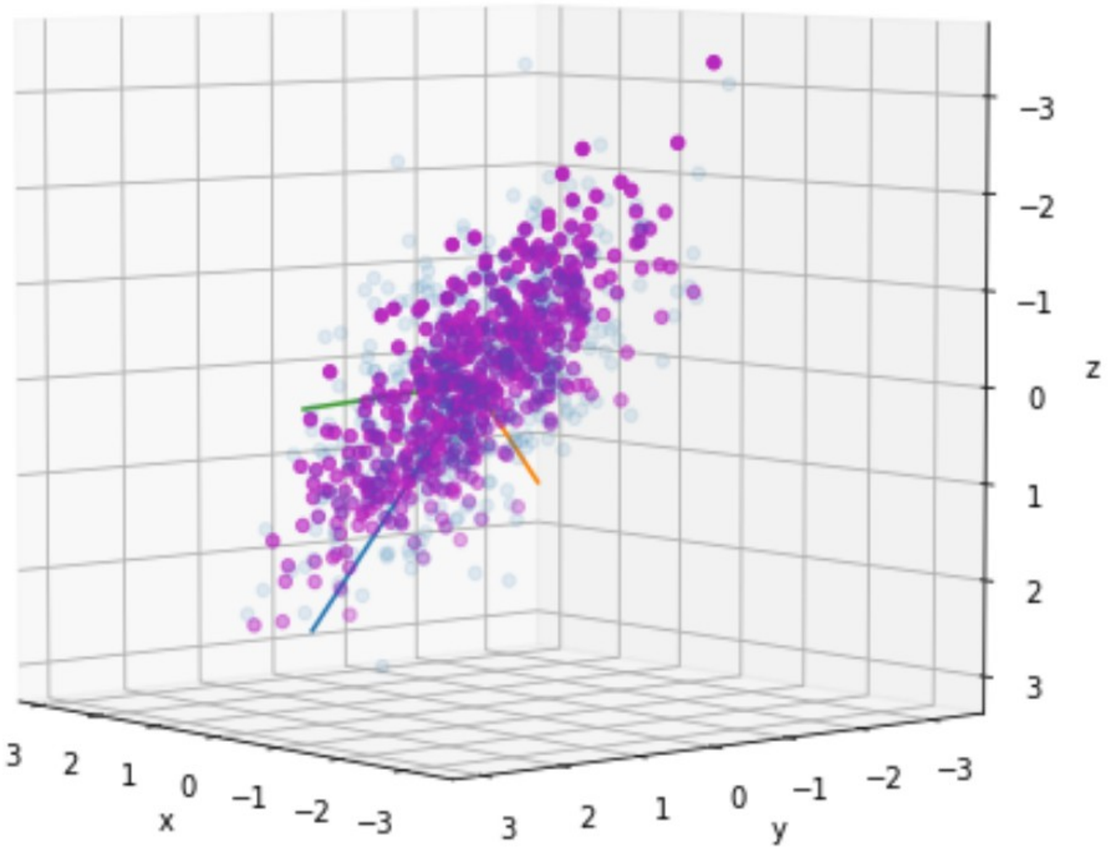
$$var(X) = \frac{\sum_{i \in N} (x_i - \mu_x)^2}{N-1} = \frac{\sum_{i \in N} (x_i - \mu_x) * (x_i - \mu_x)}{N-1}$$

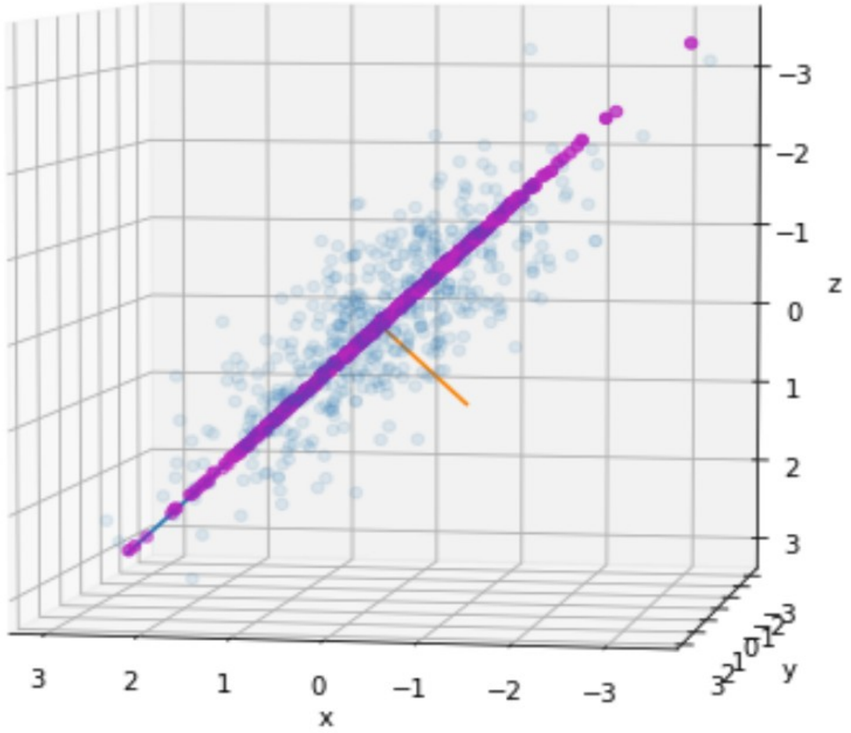
- **Covariance:**

$$covar(X, Y) = \frac{\sum_{i \in N} (x_i - \mu_x) * (y_i - \mu_y)}{N-1}$$

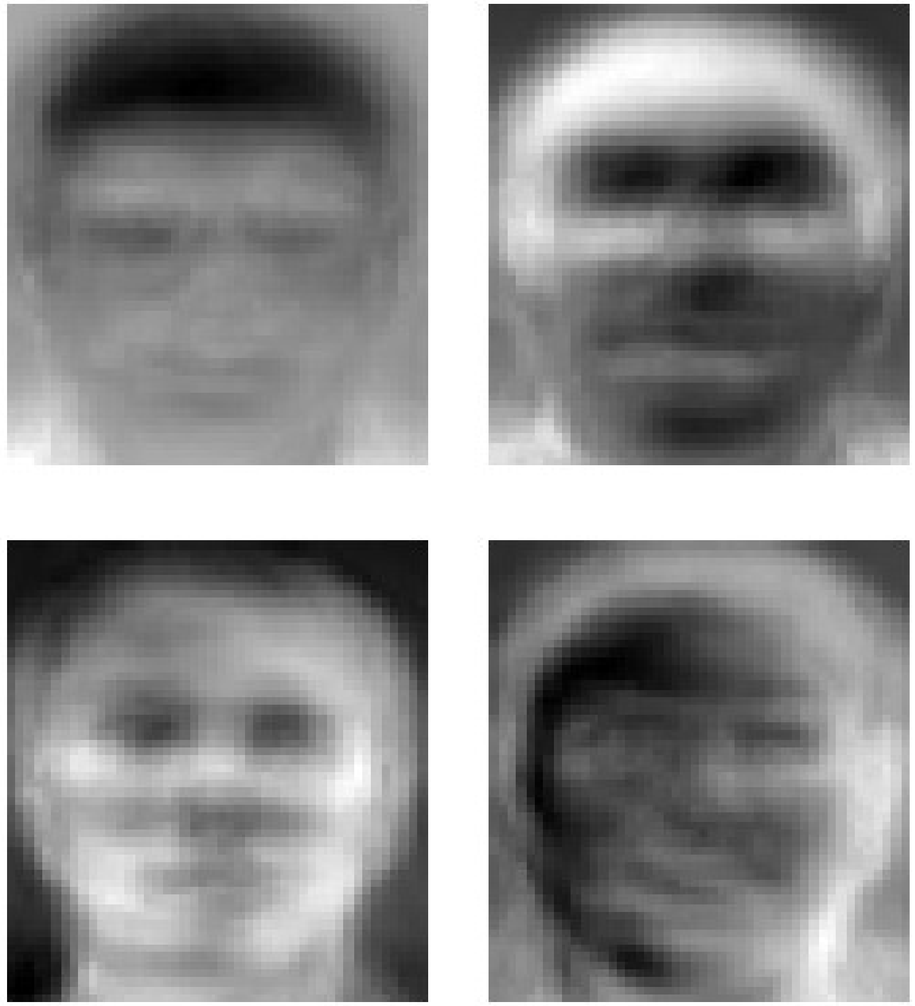
http://www.iro.umontreal.ca/~pift6080/H09/documents/papers/pca_tutorial.pdf

Dimensionality reduction: 3D - 2D





Facial recognition: Eigenfaces



https://en.wikipedia.org/wiki/Eigenface