Machine Learning – 04

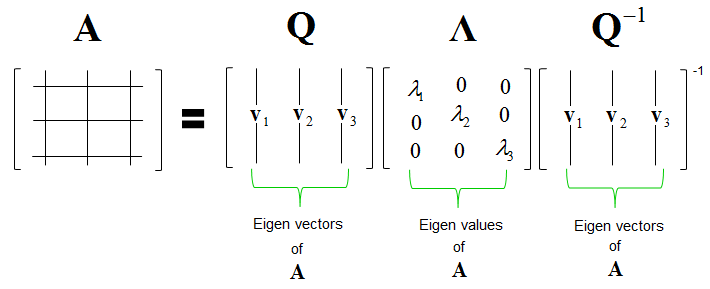
**Eigendecomposition of a matrix**

The Eigendecomposition of a square matrix is a factorization s.t the matrix is represented by its **eigenvalues and eigenvectors** (this concept is often called diagonalization of a matrix)

Let **A** be a square *n* × *n* matrix with *n* linearly independent eigenvectors *qi* (where *i* = 1, ..., *n*). Then **A** can be [factorized](https://en.wikipedia.org/wiki/Matrix_decomposition) as:

**A = Q Λ Q^-1**

�=���−1where **Q** is the square *n* × *n* matrix whose *i*th column is the **eigenvector** *qi* of **A**, and **Λ** is the [diagonal matrix](https://en.wikipedia.org/wiki/Diagonal_matrix) whose diagonal elements are the corresponding **eigenvalues**, *Λii* = *λi*

Note that only [diagonalizable matrices](https://en.wikipedia.org/wiki/Diagonalizable_matrix) can be factorized in this way, trivially

**A picture containing text

Description automatically generated**

**Remark:**

* The eigenvalues of a matrix can be found by **det(A - Iλ) = 0**
  + Where I is the n x n identity matrix
* The eigenvectors of a matrix can be found by substituting for each eigenvalue in the spectrum and computing **Ker(A-Iλ)**

Graphical user interface, text, application

Description automatically generated

Link : https://juanitorduz.github.io/the-spectral-theorem-for-matrices/

**Principal component analysis (PCA)**

PCA is used to reduce the dimensionality of a dataset (see overfitting – lack of generalization), using a transformation that preserves the most variance in the data using **the least amount of dimensions**

It in fact is assumed that the information is carried in the **variance** of the features, meaning that:

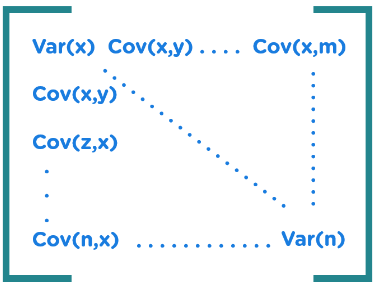
**Higher variation ⇒ more information**

We aim to follow these steps:

1. Construct the **covariance matrix**
2. Compute its eigenvalues
3. Use the eigenvectors to reconstruct data

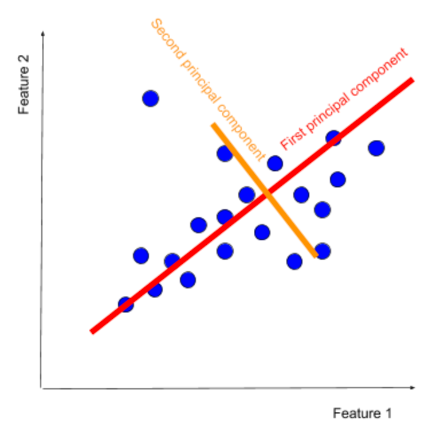
This is useful in **unsupervised learning**

**Remark: The covariance matrix**

The [covariance matrix](https://www.simplilearn.com/covariance-vs-correlation-article), a square matrix that displays the pairwise correlations between all pairs of variables in the dataset, is calculated in the setting of PCA using correlation

The covariance matrix's diagonal elements stand **for each variable's variance**, while the off-diagonal elements indicate the **covariances between different pairs of variables**

The strength and direction of the linear connection between two variables can be determined using the correlation coefficient, a standardized measure of correlation with a range of -1 to 1 (see first document)

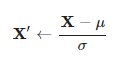
The principal components are vectors, but they are not chosen at random

The **first principal component** is computed so that it explains the greatest amount of variance in the original features

The **second component** is orthogonal to the first, and it explains the greatest amount of variance left after the first principal component

PCA allows the representation of data as linear combinations of principal components

**Calculating principal components** (https://www.keboola.com/blog/pca-machine-learning)

1. **Standardize the data** (using Z-score)

Where **µ** is the mean and **σ** is the std deviation

1. **Build the covariance matrix**
2. **Calculate its Eigendecomposition**
3. **Sort the eigenvectors from the highest eigenvalue to its lowest**

The eigenvector with the highest eigenvalue is the first principal component

Higher eigenvalues correspond to greater amounts of shared variance explained

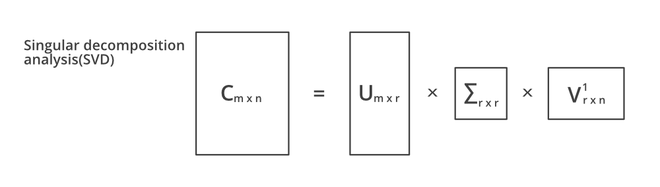
1. Text, letter

   Description automatically generated**Select the numbers of principal components** (this is dependent on the number of dimensions, for a plane it’s at most two)

**Singular value decomposition (SVD)**

SVD is a factorization of rectangular m×n matrix A as **A = U Σ V.T** where:

* U is an m×m **orthogonal** matrix with [Eigenvectors](http://mlwiki.org/index.php/Eigenvectors) of A A.T
* Σ is an **diagonal**m×n matrix with [Eigenvalues](http://mlwiki.org/index.php/Eigenvalues) of both A.T A and A A.T
* V.T is an n×n **orthogonal** matrix with [Eigenvalues](http://mlwiki.org/index.php/Eigenvalues) of A.T A

(<https://jonathan-hui.medium.com/machine-learning-singular-value-decomposition-svd-principal-component-analysis-pca-1d45e885e491>)

Chart

Description automatically generated