**K**-**means Clusters**

Unlike supervised learning, clustering is considered an unsupervised learning method since we don’t have the ground truth to compare the output of the clustering algorithm to the true labels to evaluate its performance. We only want to try to investigate the structure of the data by grouping the data points into distinct subgroups.

**K-means Algorithm:**

K-means algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster’s centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

**The way kmeans algorithm works is as follows:**

1. Specify number of clusters K.
2. Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing.

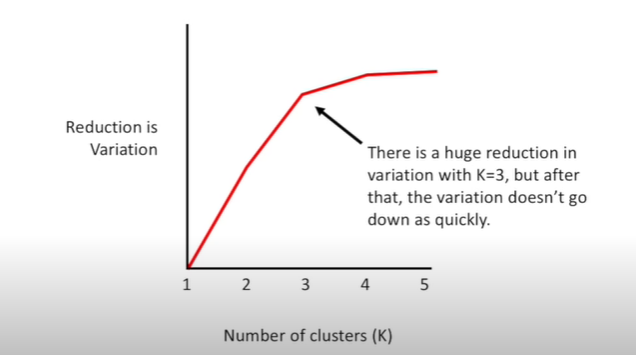
* Compute the sum of the squared distance between data points and all centroids.
* Assign each data point to the closest cluster (centroid).
* Compute the centroids for the clusters by taking the average of the all data points that belong to

each cluster.

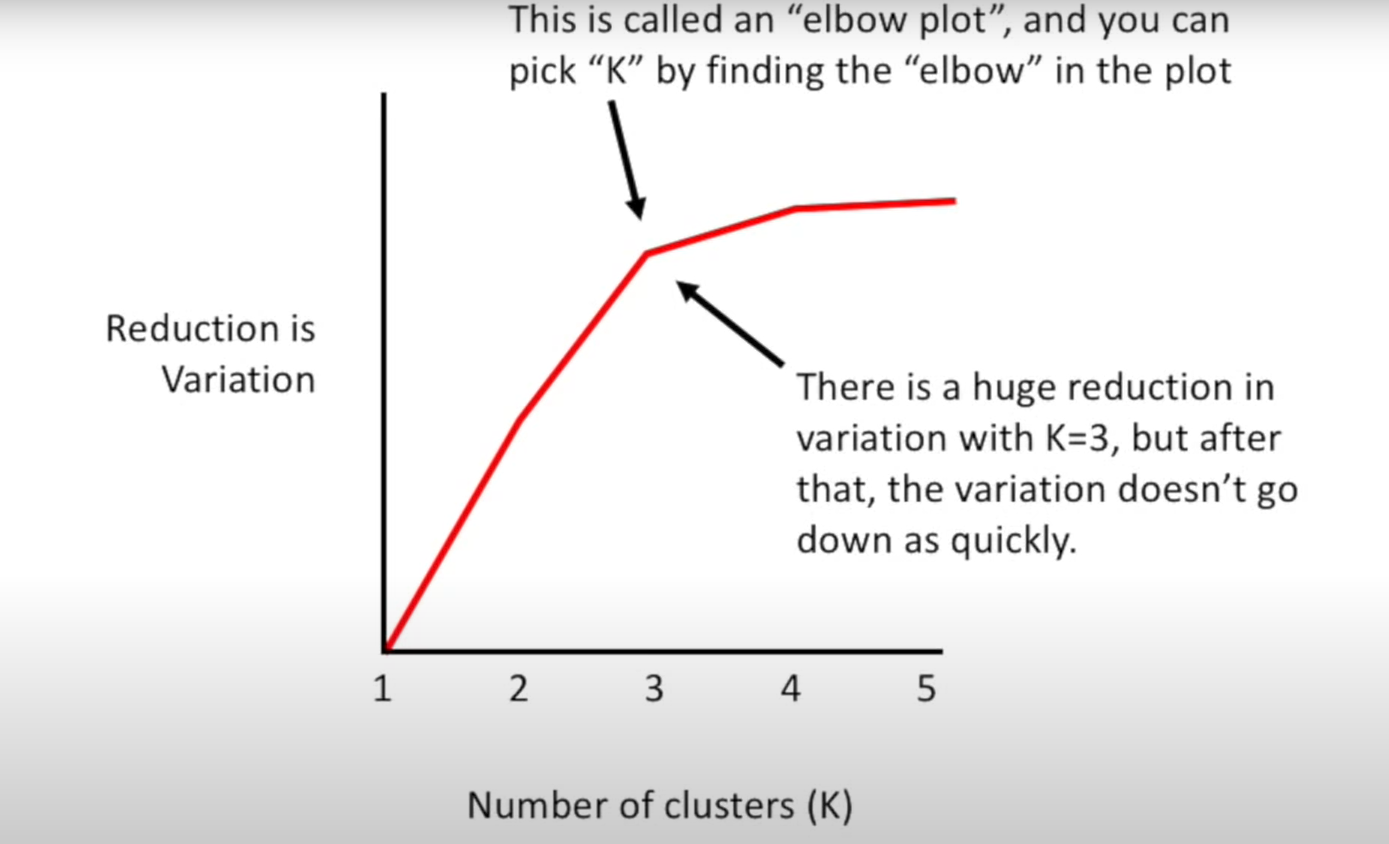
**How to choose the best K?**

After iterating over k multiple times, each time we add new cluster, total variation within each cluster is smaller than before, and when there is only one point per cluster, the variation = 0

So, if we plot the reduction in variance per value for K …



After the huge reduction in variation with k =3, the graph doesn’t go down quickly, and this point called the elbow plot, and we choose the K by finding the elbow in the plot



If our data is in 2D, we use the Euclidean distance between the points on the graph:

A screenshot of a math game

Description automatically generated

A diagram of a graph

Description automatically generated

Gaussian Mixture Models (GMM):

Gaussian Mixture Model (GMM) is a probabilistic model used for clustering and density estimation. It is a soft clustering algorithm, meaning that it assigns data points to clusters based on their probability of belonging to each cluster rather than assigning them to a single cluster. GMM assumes that the data points are generated from a mixture of multiple Gaussian distributions, each representing a cluster.

**Here's how GMM works:**

1. Initialization:
   * **Choose the number of clusters k** that you want to find in the data.
   * **Randomly initialize the parameters of the k Gaussian distributions**. These parameters include the mean, covariance matrix, and the mixture weights (weights that represent the probability of each cluster).
2. Expectation-Maximization (EM) Algorithm:

GMM uses the Expectation-Maximization (EM) algorithm to iteratively update the model parameters until convergence.

* **Expectation Step (E-step):** Calculate the probability (responsibility) of each data point belonging to each cluster (component) using the current parameters of the Gaussian distributions. This is done using the Bayes' theorem.
* **Maximization Step (M-step):** Update the parameters of the Gaussian distributions based on the calculated responsibilities. The updates involve calculating new means, covariance matrices, and mixture weights that maximize the likelihood of the data given the current model.

1. Convergence:

The E-step and M-step are performed iteratively until the model converges, or until a specified number of iterations is reached.

1. Cluster Assignment:

Once the model has converged, each data point is assigned to the cluster with the highest probability (responsibility) of belonging to that cluster. This assignment is probabilistic, meaning that data points might have partial memberships in multiple clusters.

GMM is useful when the underlying data distribution is complex and not well-separated, and when data points can belong to multiple clusters simultaneously. It is also effective for density estimation, as it models data as a combination of Gaussian distributions.

One of the main advantages of GMM is its flexibility in capturing clusters with different shapes and sizes. However, GMM can be sensitive to the initialization of parameters and may converge to local optima. Therefore, it is common to run the EM algorithm multiple times with different initializations and choose the model that gives the highest likelihood or lowest WCSS (within-cluster sum of squares).

**Mathematical side:**

The Gaussian Mixture Model (GMM) algorithm uses probability and statistics to model the data as a mixture of multiple Gaussian distributions. Let's break down the mathematical equations used in the GMM algorithm:

1. Probability Density Function (PDF) of a Gaussian Distribution:

The PDF of a univariate Gaussian distribution is given by:

Where:

* or width) of the Gaussian distribution.

For multivariate Gaussian distributions (when dealing with multiple features), the equation extends to:

A number of mathematical equations

Description automatically generated

Where:

* + **x** is the vector of input data points (a data point with multiple features).
  + **μ** is the vector of means of the Gaussian distribution (a vector of means for each feature).
  + **Σ** is the covariance matrix, representing the spread and correlations between different features.

1. Mixture Model:

In GMM, the data is modeled as a mixture of **k** Gaussian distributions. The probability of a data point **x** being generated by the entire mixture is the sum of probabilities from each individual Gaussian component:

Where:

* k is the number of Gaussian components (clusters).
* is the mixing coefficient for the i-th Gaussian component, representing the probability of the i-th cluster being selected.
* is the PDF of the i-th Gaussian component.

1. Expectation-Maximization (EM) Algorithm: The EM algorithm is used to iteratively estimate the model parameters that maximize the likelihood of the data given the mixture model.

* Expectation step (E-Step): Calculate the posterior probabilities (responsibilities) of each data point belonging to each cluster, given the current model parameters:

A mathematical equation with numbers and symbols

Description automatically generated

A screenshot of a computer code

Description automatically generated

* Maximization step (M-Step):

A math equation with black text

Description automatically generated

1. Convergence:

The E-step and M-step are performed iteratively until the model parameters converge, or until a maximum number of iterations is reached.

The process of Expectation-Maximization allows GMM to estimate the parameters of the mixture model that best fit the given data distribution, effectively identifying the underlying clusters in the data.

A diagram of different color points

Description automatically generated