Unsupervised Learning

15-17 November 2021

UNSUPERVISED MACHINE LEARNING ALGORITHMS[1]

- Are applied when given data are ***neither classified nor labeled***.
- No desired output, rather find difference among data
- Draw inferences to describe hidden structures from unlabeled data.

Goal: model the underlying structure of or distribution in the data, group data according to similarities, represent data in a compressed format

- Algorithms are left to their own devising to discover and present the interesting structure in the input data.

UNSUPERVISED MACHINE LEARNING ALGORITHMS[2]

Clustering: discover the inherent groupings in the data

- e.g. clustering DNA sequences into functional groups.

Association: discover rules that describe large portions of your data

- e.g. association analysis-based techniques for pre-processing protein interaction networks for the task of protein function prediction.

Dimensionality reduction: reduce the variable space of high dimensionality before the subsequent analysis is carried out.

- e.g. in a gene-expression analysis, for finding a list of candidate genes with a more operable length ideally including all the relevant genes.

EXAMPLES OF UNSUPERVISED LEARNING ALGORITHMS

15-17 November 2021

PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA can be applied for dimensionality reduction.

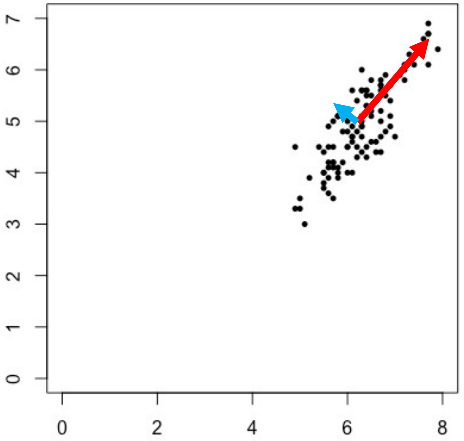
- Data with a wide range of features (e.g. omics), probably highly correlated between each other => overfitting models

Algorithm: find linear combinations of variables having maximum variances

- Standardization
- Covariance matrix computation
- Compute eigenvalues (amounts of variance) and eigenvectors (PCs) of the covariance matrix

Advantage:

- low-dimensional sample representation
- synchronized low-dimensional representation of the variables
- visually find variables that are characteristic of a group of samples.



HIERARCHICAL CLUSTERING

Group similar objects together into *clusters* (unknown number of clusters *a priori*): Bottom-up (Agglomerative) & Top-down (Divisive)

Agglomerative

- Algorithm:

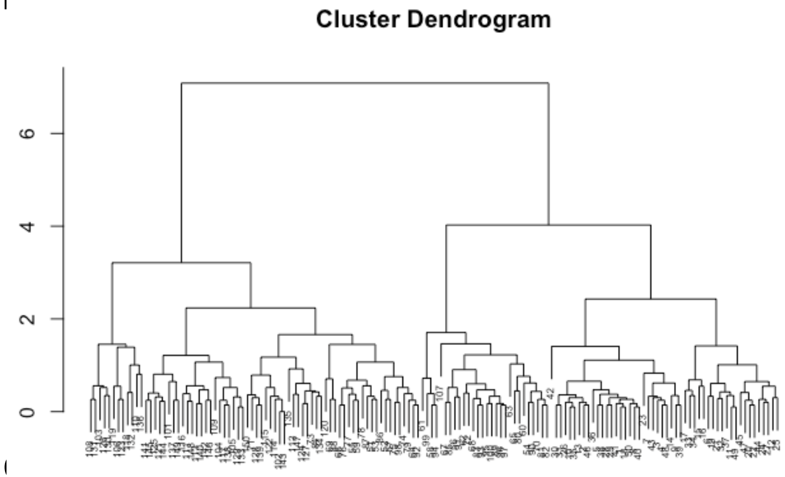
  It starts by treating each observation as a separate cluster.

  Then, it repeatedly executes the following two steps:

  1) identify the two clusters that are closest together, and
  2) merge the two most similar clusters.

  This iterative process continues until all the clusters are merged together.

- The main output of Hierarchical Clustering is a dendrogram, which shows the hierarchical relationship between the clusters

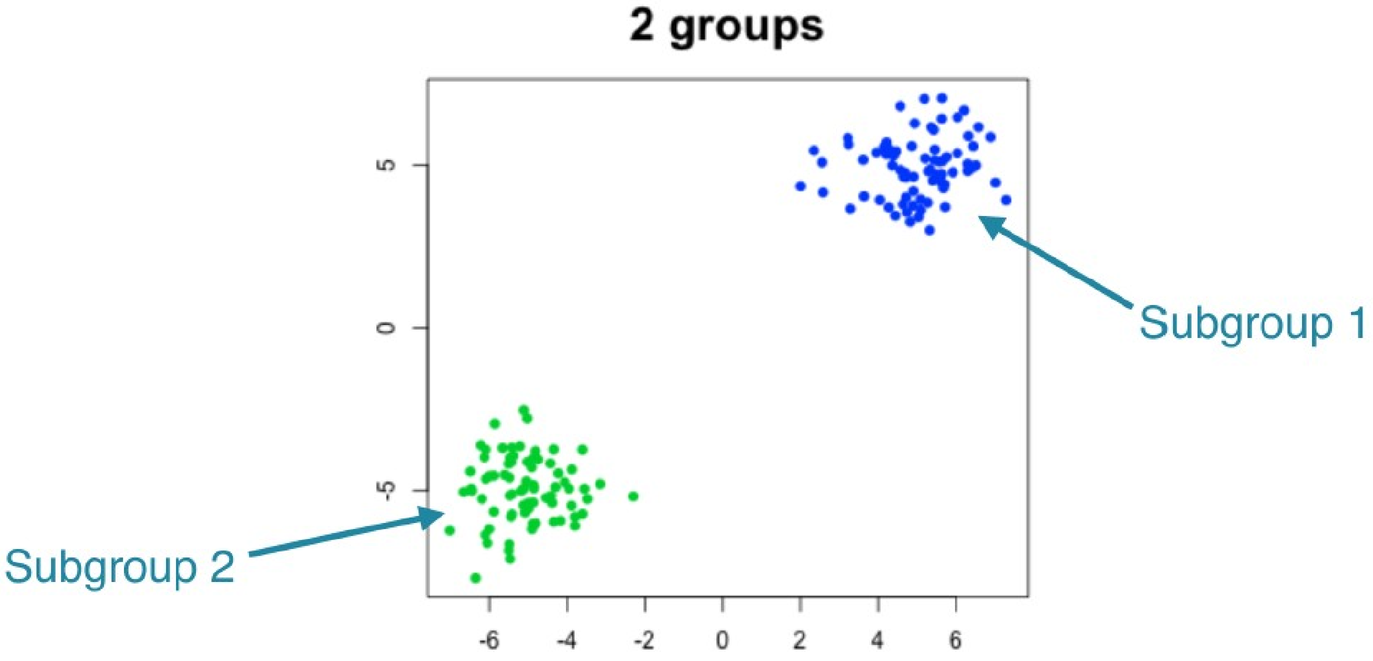

Cluster Dendrogram

E.g. gene expression data analysis - genes with similar expression patterns are grouped together and are connected by a series of branches.

K-MEANS[1]

Find homogeneous subgroups in a population

- Break observations into a pre-defined number of clusters



K-MEANS[2]

Algorithm

1. Divide the data into K clusters

   Initialize the centroids with the mean of the clusters

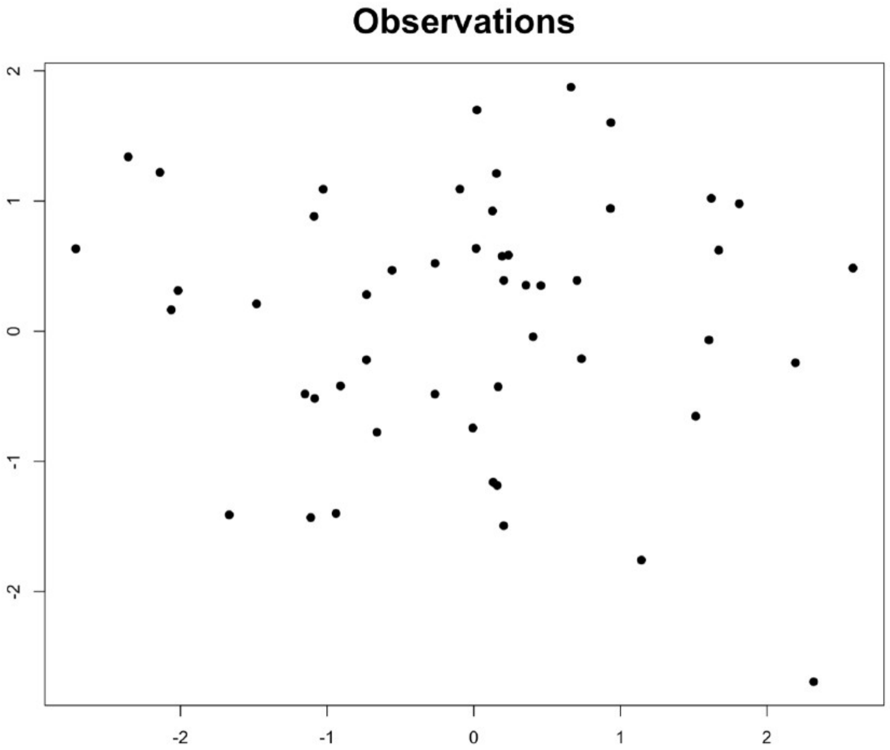
2. Assign each item to the cluster with closest centroid

3. When all objects have been assigned, recalculate the centroids (mean)

4. Repeat 2-3 until the centroids no longer move

K-MEANS[3]

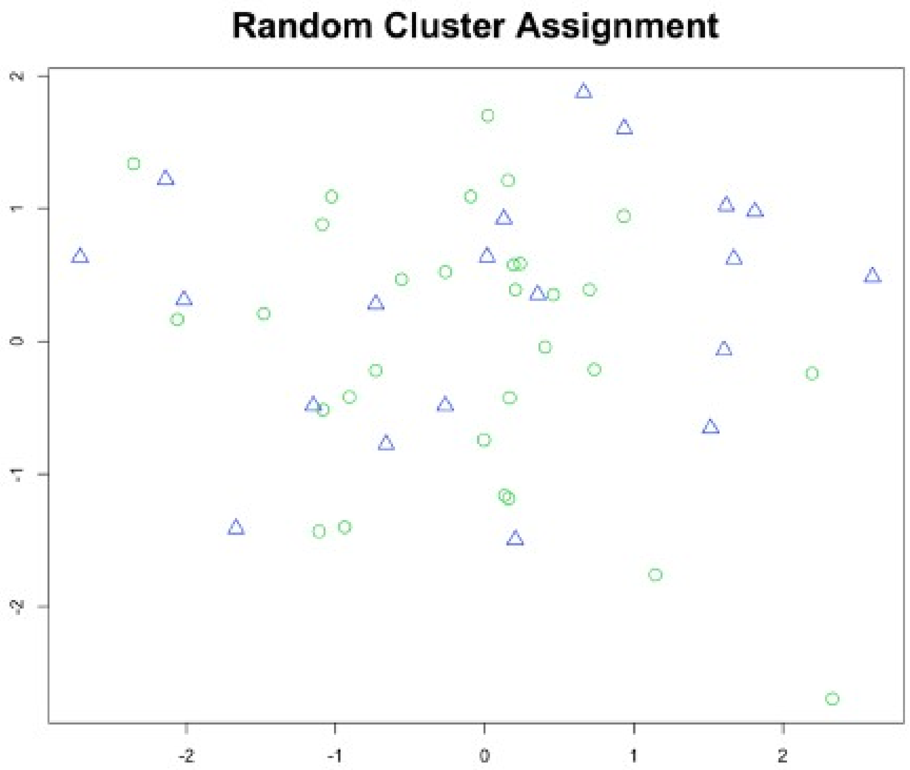
The Algorithm in action



15-17 November 2021

K-MEANS[4]

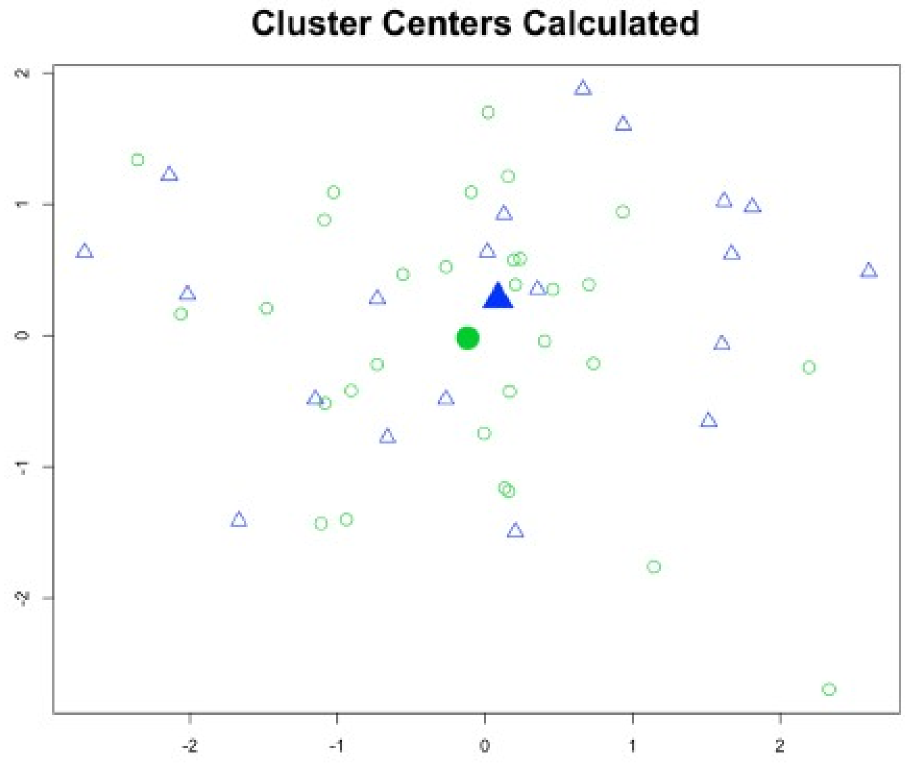
The Algorithm in action



15-17 November 2021

K-MEANS[5]

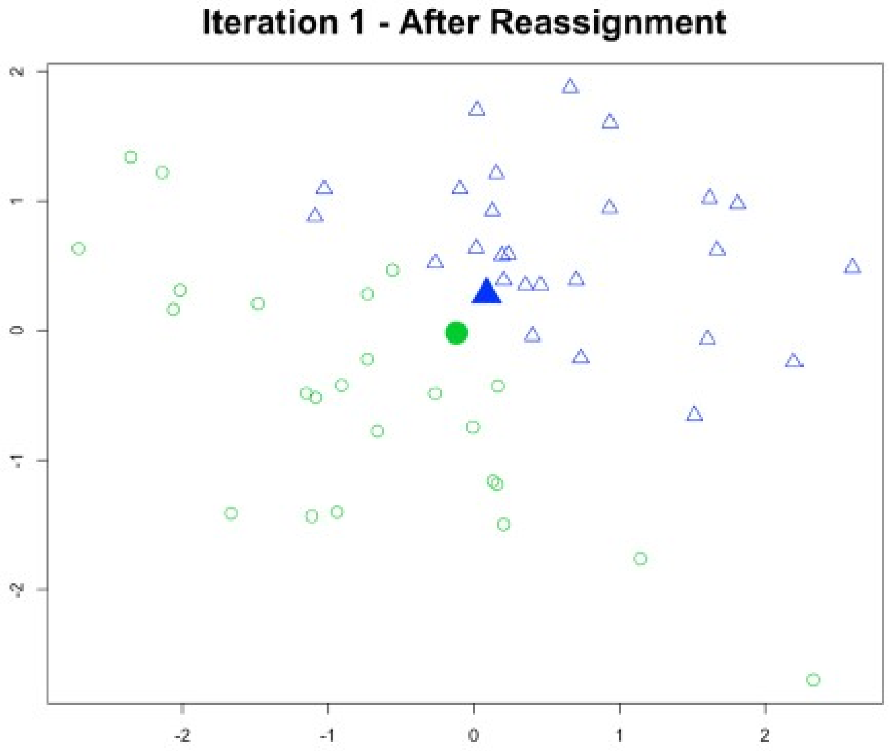
The Algorithm in action



15-17 November 2021

K-MEANS[6]

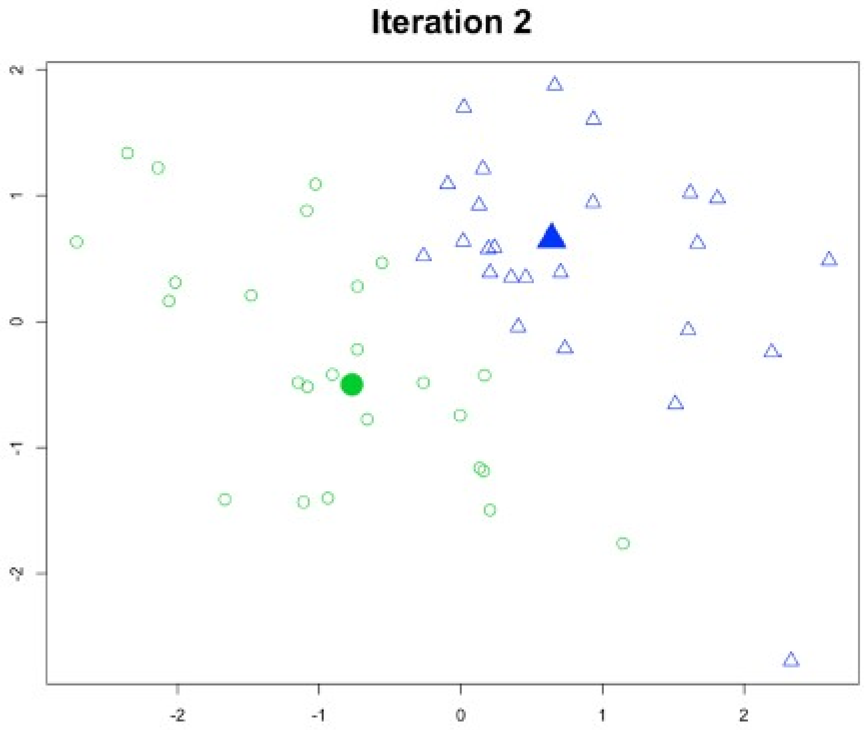
The Algorithm in action



15-17 November 2021

K-MEANS[7]

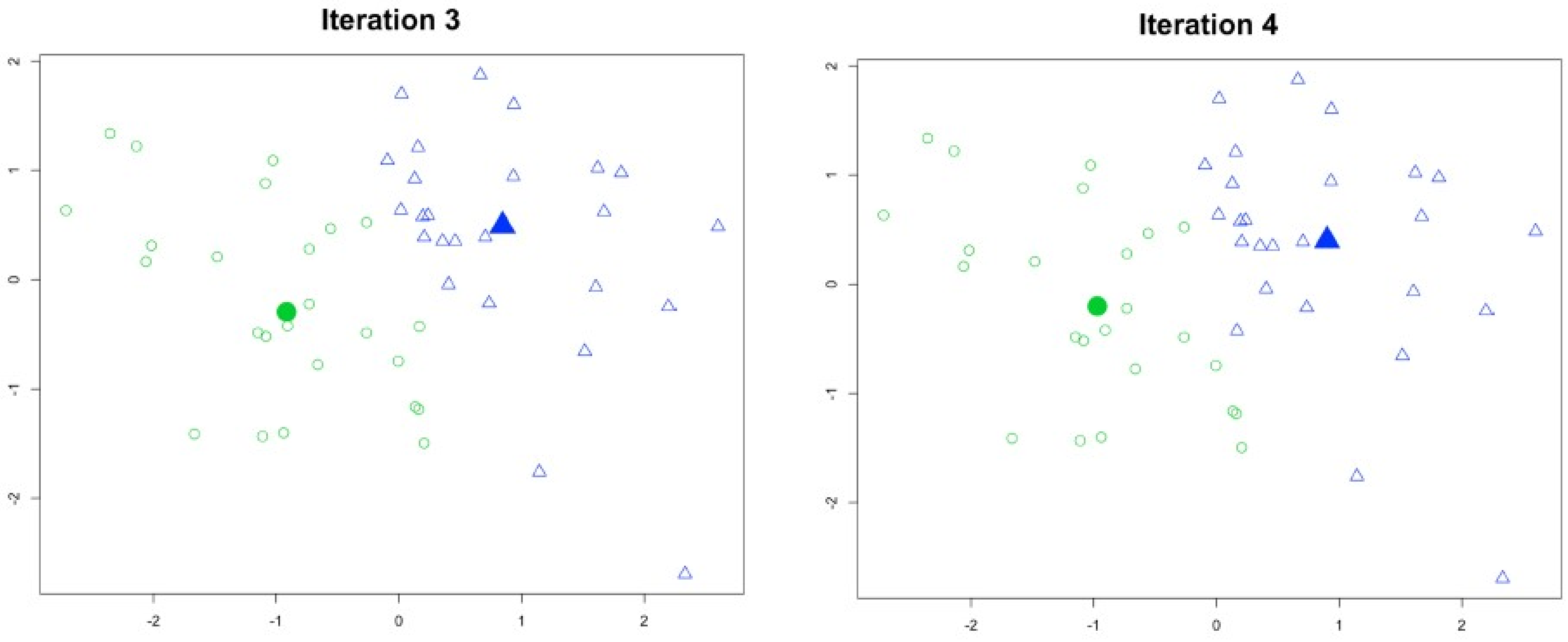
The Algorithm in action



15-17 November 2021

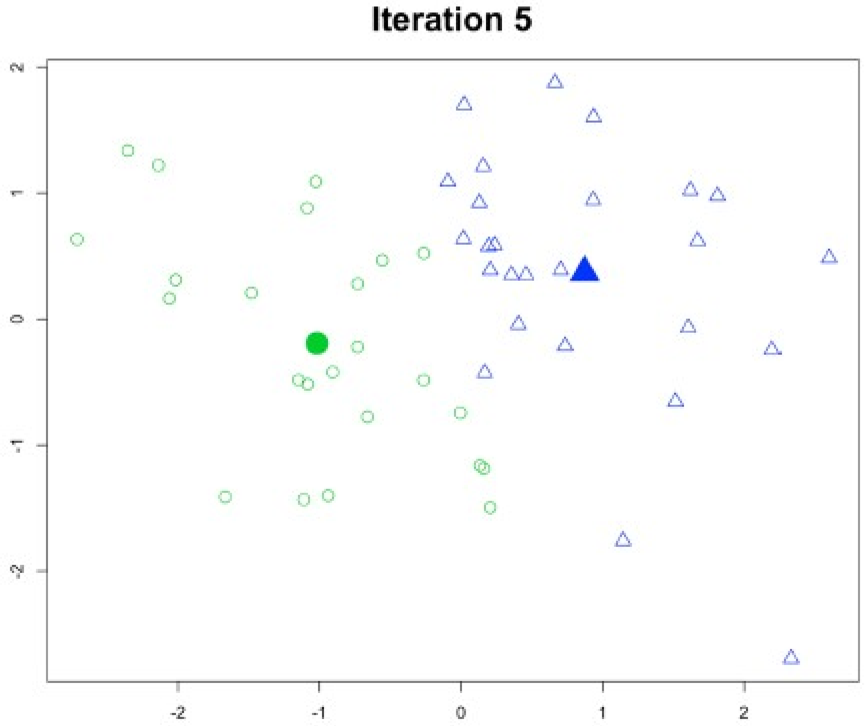
K-MEANS[8]

The Algorithm in action



K-MEANS[9]

The Algorithm in action



Remember k-means has a random component!

K-MEANS[10]

Goal: find groups in the data, with a pre-defined number of groups *K*.

The algorithm works iteratively to assign each data point to one of *K* groups based on the features that are provided. Data points are clustered based on feature similarity.

Advantage: Easy to implement and fast and efficient in terms of computational cost

Disadvantage include:

- Initial seeds have a strong impact on the final results
- The order of the data has an impact on the final results
- K-Means needs to know in advance how many clusters there will be in your data, so this may require a lot of trials to “guess” the best K number of clusters to define.

E.g. clustering COVID regions, virus sequences, lockdown measures

DETERMINING OPTIMAL CLUSTERS

- **Elbow method** : heuristic where you look at the metric optimised by your model (mean error, explained variance, …) and pick a point where the curve does an ‘elbow’ or ‘knee’: this is the point where complexifiying your model (eg. adding a cluster), has a diminishing return.

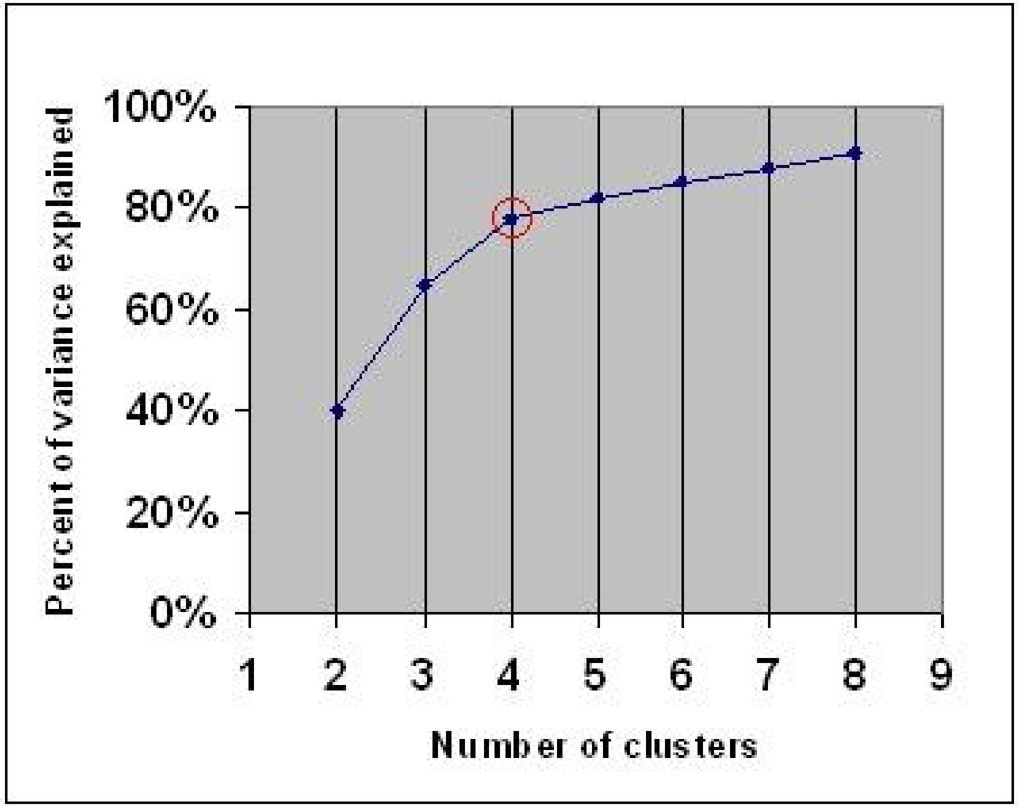


Image credit to wikipedia user Maksim, distributed under the CC BY-SA 3.0 license

DETERMINING OPTIMAL CLUSTERS

- Use a metric, for example the **silhouette coefficient**
  - -1 (worst) to 1 (best)
  - (a – b) / max(a,b)
  - a = average intra-cluster distance
  - b = average distance to closest other cluster

  - Computed for each point,

    then average

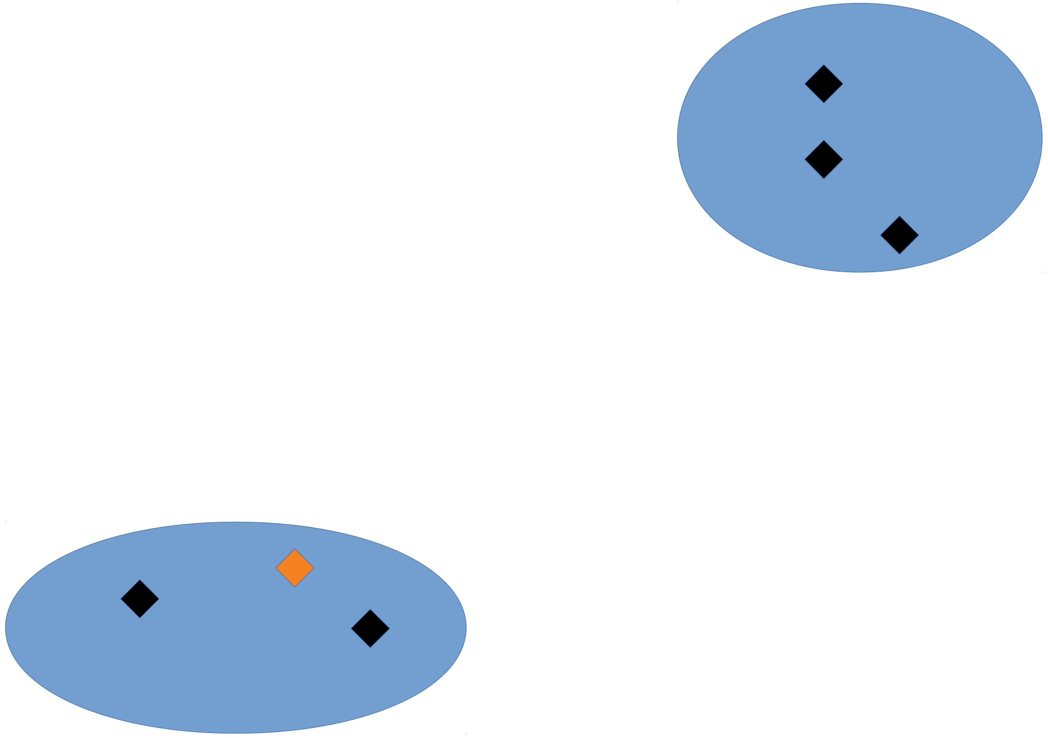


Image credit to wikipedia user Maksim, distributed under the CC BY-SA 3.0 license

DETERMINING OPTIMAL CLUSTERS

- Use a metric, for example the **silhouette coefficient**
  - -1 (worst) to 1 (best)
  - (a – b) / max(a,b)
  - a = average intra-cluster distance
  - b = average distance to closest other cluster
  - Computed for each point,

    then average

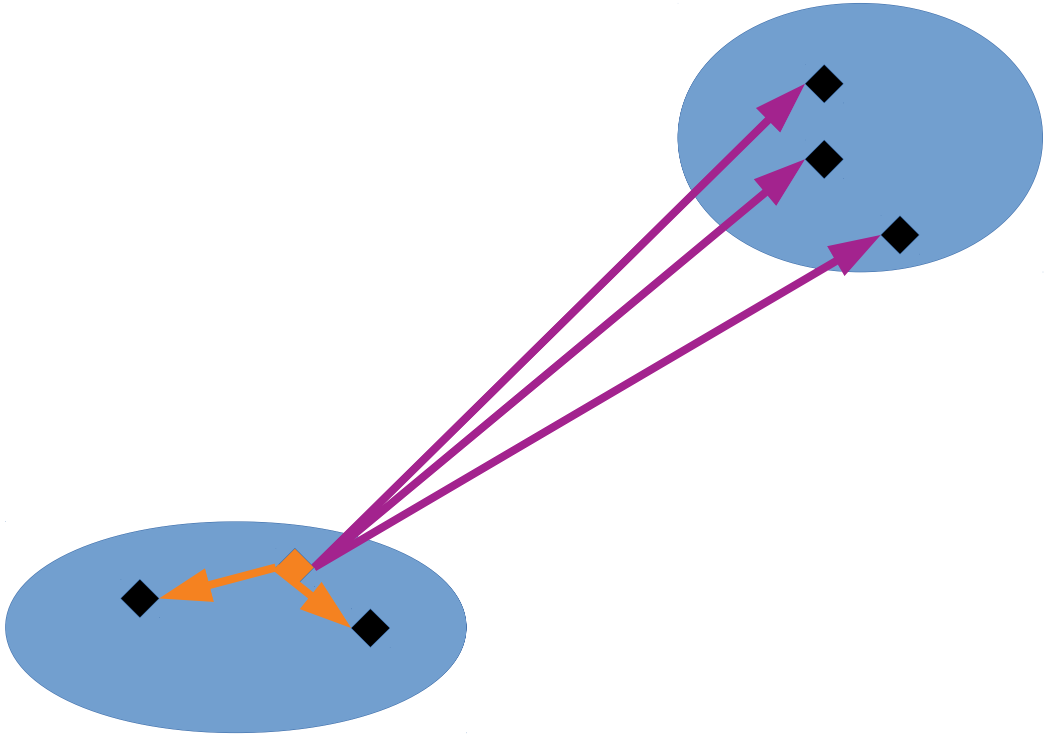


Image credit to wikipedia user Maksim, distributed under the CC BY-SA 3.0 license

15-17 November 2021