AIML Lecture 6 : Reading Material

1 K-Means Clustering

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable 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. The results of the K-means clustering algorithm are the centroids of the K clusters, which can be used to label new data.

Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically. Each centroid of a cluster is a collection of feature values which define the resulting groups. Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.

The K-means clustering algorithm uses iterative refinement to produce a final result. The algorithm inputs are the number of clusters K and the data set. The data set is a collection of features for each data point. The algorithms starts with initial estimates for the K centroids, which can either be randomly generated or randomly selected from the data set. The algorithm then iterates between two steps:

1. Data assignment step: Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance.

2. Centroid update step: In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

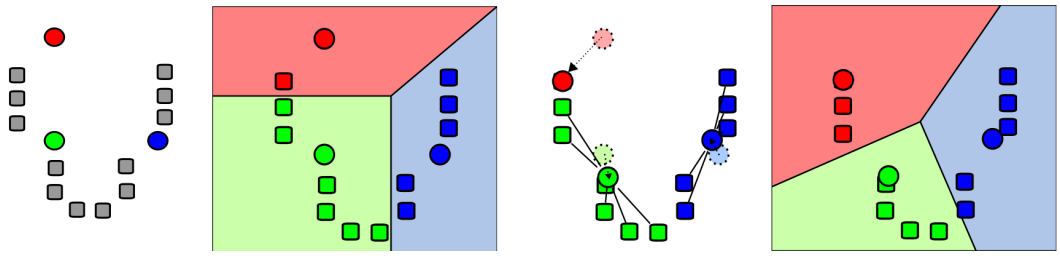


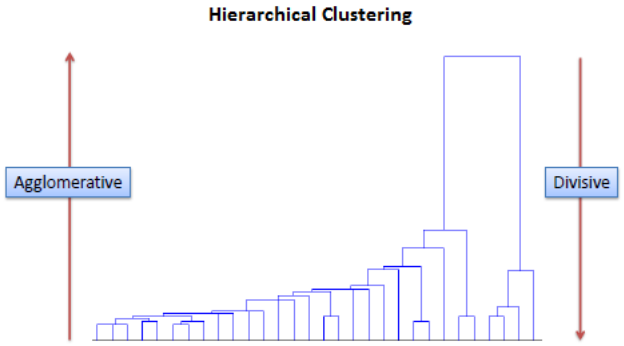
Figure 1: The input vectors are the squares. Circles denote the current cluster centers. 1.) Cluster centers are randomly initialized. 2.) Vectors are mapped to the closest cluster centers. 3.) Cluster center is recomputed to be the mean of all vectors mapped to the same cluster center in the previous step. 4.) Vectors are remapped again to the newly compute cluster centers, according to distance.

2 Hierarchical Clustering

Hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

1. Agglomerative: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

2. Divisive: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.



In order to decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, this is achieved by use of an appropriate metric (a measure of distance between pairs of observations), and a linkage criterion which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets.

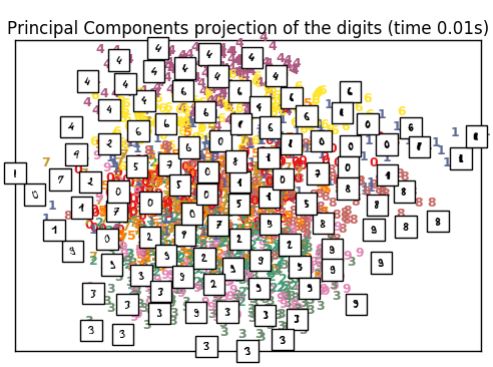
In single linkage hierarchical clustering, the distance between two clusters is defined as the shortest distance between two points in each cluster. In complete linkage hierarchical clustering, the distance between two clusters is defined as the longest distance between two points in each cluster.

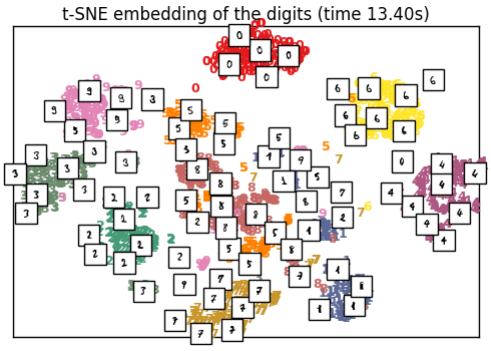
3 Non-linear Dimensionality Reduction

High-dimensional data, meaning data that requires more than two or three dimensions to represent, can be difficult to interpret. One approach to simplification is to map it to a space of low enough dimension, the data can be visualised in the low-dimensional space. PCA (principal component analysis) is one way of doing this which was discussed previously. However such methods involve multiplication by a matrix which is a linear function. Most data in real life require more sophisticated non-linear data-driven approaches for giving a proper visualization.

3.1 t-distributed Stochastic Neighbor Embedding (t-SNE)

t-distributed stochastic neighbor embedding (t-SNE) is a machine learning algorithm for dimensionality reduction developed by Geoffrey Hinton and Laurens van der Maaten. It is a nonlinear dimensionality reduction technique that is particularly well-suited for embedding high-dimensional data into a space of two or three dimensions, which can then be visualized in a scatter plot. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points.





4 References

1. K-Means Clustering
   https://en.wikipedia.org/wiki/K-means_clustering
   https://www.datascience.com/blog/k-means-clustering
   https://www.geeksforgeeks.org/k-means-clustering-introduction/

2. Hierarchical Clustering
   https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction

   https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html

3. Non-Linear Dimensionality Reduction
   https://en.wikipedia.org/wiki/Hierarchical_clustering
   https://nlp.stanford.edu/IR-book/html/htmledition/hierarchical-clustering-1.html
   http://www.analytictech.com/networks/hiclus.htm
   https://www.cs.utah.edu/~piyush/teaching/25-10-slides.pdf

4. t-SNE
   https://lvdmaaten.github.io/tsne/ https://colah.github.io/posts/2015-01-Visualizing-Re
   https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/
   https://www.youtube.com/watch?v=RJVL80Gg3lA