MA2823: Foundations of Machine Learning
Chapter 12: Clustering

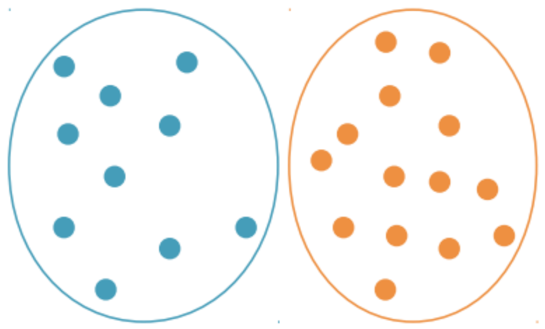
Lecturer: Benoît Playe ; Scribes: Soufiane Hadji, Seung Eun Yi, John Pougué, François Tirvaudey

In this section, we will see:
- what clustering algorithms can be used for;
- how to implement 3 different ways to evaluate clustering algorithms;
- how to implement hierarchical clustering, discuss its various flavors;
- how to implement k-means clustering, and how to discuss its advantages and drawbacks.

1 Introduction to clustering

The goal of *clustering* is to group objects that are similar into *clusters*: classes that are unknown beforehand.

We can see below an illustration of this goal :



Clustering is widely used, for instance:

- group genes that are similarly affected by a disease
- group the people who share the same interest (marketing segmentation)
- group pixels in an image that belong to the same object (image segmentation)

Clustering allows us to *understand general characteristics of the data*, to *visualize it*, and to *infer some properties of a data point* based on how it relates to other data points. Therefore, we find its applications in various industries:

- find subtypes of diseases;
- visualize protein families;
- find categories among images;
- find patterns in financial transactions;

- detect communities in social networks.

In each cluster, we can define a **centroid** and a **medoid**. The centroid is the mean of the points in the cluster, while the medoid is the point in the cluster which is the closest to the centroid.

2 Distance and similarities

First, let's define what is the *kernel* of two objects. We consider the mapping :

$$\phi : X \mapsto H \tag{1}$$

X is the space of objects and H is a Hilbert space. The *kernel* between 2 random objects x and x' of X is the inner product of their images in the feature space.

$$\forall x, x' \in X, K(x, x') =< \Phi(x), \Phi(x') >_H \tag{2}$$

Then, we introduce the concept of **distance**. It allows us to *assess how close/far* :

- data points are from each other;
- a data point is from a cluster;
- two clusters are from each other.

Distances verifie some properties :

$$d : X \mapsto \Re \tag{3}$$

$$d(x, x) = 0 \tag{4}$$

Symmetry :

$$d(x^1, x^2) = d(x^2, x^1) \tag{5}$$

Triangle inequality :

$$d(x^1, x^2) \leq d(x^1, x^3) + d(x^3, x^2) \tag{6}$$

Distance and similarities are heavily linked. For instance, we can transform distances into similarities :

$$sim(x, x') = \frac{1}{1 + d(x, x')} \tag{7}$$

$$sim(x, x') = e^{-\gamma * d(x,x')^2} \tag{8}$$

Finally, we'll be using L-q norm and Pearson's correlation:

$$d(x^1, x^2) = ||x^1 - x^2||_q = (\sum_{j=1}^{p} |x^1_j - x^2_j|^q)^{1/q} \tag{9}$$

Pearson's correlation measures the *linear correlation between 2 variables* :

$$\rho(x,z) = \frac{\sum_{j=1}^{p}(x_j-\bar{x})(z_j-\bar{z})}{\sqrt{\sum_{j=1}^{p}(x_j-\bar{x})^2}\sqrt{\sum_{j=1}^{p}(z_j-\bar{z})^2}} \tag{10}$$

where

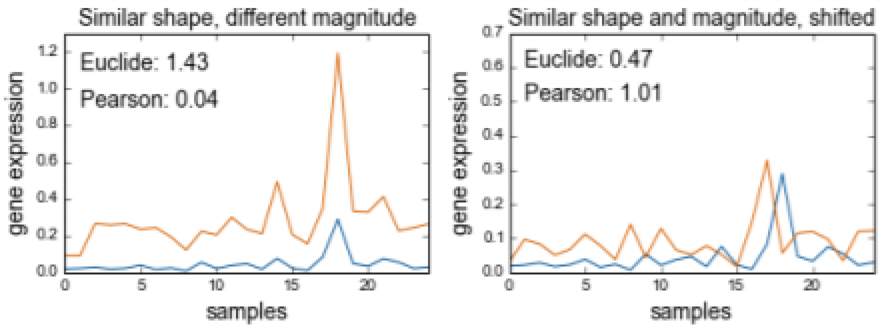
$$\bar{x} = \frac{1}{p}\sum_{j=1}^{p} x_j \tag{11}$$

$$\bar{z} = \frac{1}{p}\sum_{j=1}^{p} z_j \tag{12}$$

If the two variables are *centered*, the correlation becomes much simpler :

$$\rho(x,z) = \frac{\sum_{j=1}^{p} x_j z_j}{\sqrt{\sum_{j=1}^{p} x_j^2}\sqrt{\sum_{j=1}^{p} z_j^2}} = \frac{<x,z>}{||x||||z||} = \cos(\theta) \tag{13}$$

It is interesting then to compare Pearson's coefficient with Euclidean distance. The value of Pearson's coefficient essentially depends on profiles of *shapes*, while Euclidean distance takes into account both *shapes* and **magnitude**.



3 Evaluating clusters

As its definition indicates, clustering is **unsupervised** - we can find patterns in the data but we cannot conclude directly that the discovered patterns are true. We can evaluate the quality of a clustering algorithm by three ways :
- based on the *shape* of the clusters;
- based on the *stability* of the clusters;
- based on *domain knowledge* (the clusters should "make sense").

3.1 Clusters shape

We can define several indicators:

- cluster **tightness** (**homogeneity**):

$$\sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{x \in C_k} d(x, \mu_k) = \sum_{k=1}^{K} T_k \tag{14}$$

- cluster **separation** :

$$\sum_{k=1}^{K} \sum_{l=k+1}^{K} d(\mu_k, \mu_l) = \sum_{k=1}^{K} \sum_{l=k+1}^{K} S_{kl} \tag{15}$$

- **Davies-Bouldin index** :

$$DB = \frac{1}{K} \sum_{k=1}^{K} D_k \tag{16}$$

  where

$$D_k = \max_{l:l \neq k} \frac{T_k + T_k}{S_{kl}} \tag{17}$$

- **silhouette coefficient** :

$$s(x) = \frac{b(x) - a(x)}{\max\left(a(x), b(x)\right)} \tag{18}$$

  where a(x) is the **average similarity** of x with the other points in the same cluster, and b(x) the **lowest average dissimilarity** of x with any other point. If x is very close to other points in the same cluster, and very different from points in other cluster, s(x) would be *close to 1.* If x is assigned to the wrong cluster, s(x) would be *close to -1.* A s(x) *near zero* means that x is on the border of two neighboring clusters.

3.2 Cluster stability

Given a set of data, several algorithms or runs of the same algorithm might give us different answers for the number and the form of clusters. To measure cluster stability :

- We generate perturbed versions of the original dataset (by sub-sampling or adding noise for example).
- We cluster the data set with the desired algorithm into k clusters.
- Then we can measure instability :

$$\widehat{Instab}(k) = \frac{1}{b_{max}^2} \sum_{b,b'=1}^{b_{max}} d(C_b, C_{b'}) \tag{19}$$

We can therefore choose the number of cluster k which minimizes the instability mesure.

3.3 Domain knowledge

To make sure that the clusters formed match natural categories, we should check them with human expertise.An example of this enrichment analysis is **ontology**, build by human experts, which argues that entities may be grouped, related within a hierarchy, and subdivided according to similarities and differences. It is used in many domains such as the genetics.

For example, to answer the question of whether there are more data points from ontology category G in cluster C than expected by chance :
- We assume data points sampled from a hypergeometric distribution.
- We calculate the *probability for the intersection of G and C to contain more than t points* :

$$Pr(|G \cap C| \geq t) = 1 - \sum_{i=1}^{t} \frac{\binom{|G|}{i}\binom{n-|G|}{|C|-i}}{\binom{n}{|C|}} \tag{20}$$

4 Hierarchical clustering

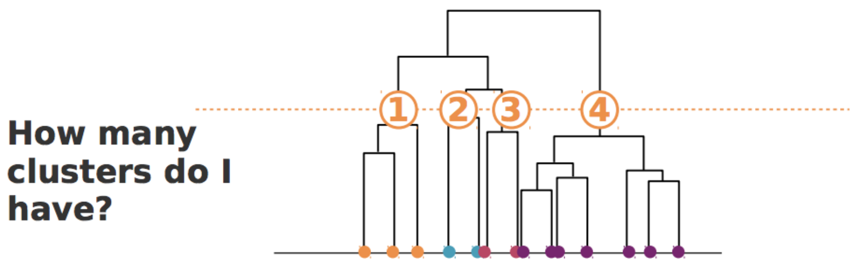
It consists in groupping data over a variety of possible scales, in a multi-level hierarchy.

4.1 Construction

There are two approaches: the agglomerative and the divisive one:
- in the **agglomerative** approach (bottom-up), we start with each element in its own cluster and iteratively join neighboring clusters.
- in the **divisive** approach (top-down), we start with all elements in the same cluster and iteratively separate into smaller clusters.

4.2 Dendogram

The results of a hierarchical clustering algorithm are presented in a dendogram, where the branch length is equal to the cluster distance.



4.3 Linkage

It is thus natural to wonder how we should decide how to connect or split two clusters. Various linkages exist:

- single linkage:
$$d(C_1, C_2) = min_{x \in C_1, z \in C_2} d(x, z) \tag{21}$$

- complete linkage:
$$d(C_1, C_2) = max_{x \in C_1, z \in C_2} d(x, z) \tag{22}$$

- Average linkage or UPGMA (i.e. Unweighted Paired Group Method with Arithmetic mean):
$$d(C_1, C_2) = \sum_{x \in C_1, z \in C_2} \frac{d(x, z)}{\mid C_1 \mid\mid C_2 \mid} \tag{23}$$

- Centroid linkage or UPGMC (i.e. Unweighted Paired Group Method using Centroids):
$$d(C_1, C_2) = d(\sum_{x \in C_1} \frac{x}{\mid C_1 \mid}, \sum_{z \in C_2} \frac{z}{\mid C_2 \mid}) \tag{24}$$

- Ward: we join clusters as to minimize within-cluster variance
$$Var_{in}(C) = \frac{1}{\mid C \mid} \sum_{x \in C} \|x - \mu_C\|^2 \tag{25}$$

4.4 Advantages & Drawbacks

Advantages:

- no need to pre-define the number of clusters
- interpretability
- two clusters are from each other.

Drawbacks:

- computational complexity: E.g. for single/complete linkage (naive) at least $O(pn^2)$ to compute all pairwise distances
- we must decide at which level of the hierarchy to split
- lack of robustness (unstable)

5 K-means clustering

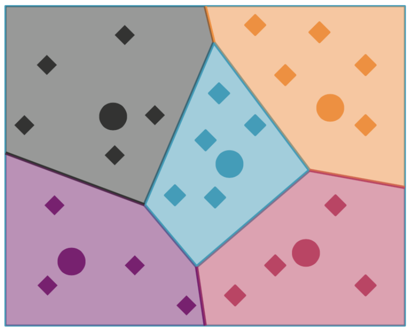
5.1 Explanation

Here, the goal is to minimize the intra-cluster variance defined by :

$$Var_{in}(C_k) = \frac{1}{|C_k|} \sum_{C_k} \|x - \mu_{C_k}\|^2 \tag{26}$$

$$V = \sum_{k=1}^{K} Var(C_k) \tag{27}$$

The K-means clustering partition of space looks like **(Voronoi tesselation)** :



5.2 Advantages & Drawbacks

Advantages:

- Computational time is linear;
- Easily implementable;

Drawbacks:

- K-means cannot be easily optimized : we adopt a greedy strategy;
- Need to set up K ahead of time;
- Sensitive to noise and outliers;
- Stochastic (different solutions which each iteration)
- The clusters are forced to have spherical shapes

5.3 K-means variants

- **K-means ++** : deterministic variant - Seeding algorithm to initialize clusters with centroids "spread-out" throughout the data.
- **K-medoids**
- **Kernel k-means** : find clusters in feature space thanks to use of adapted kernel(s).

6 More clustering approaches

- **Soft clustering**: each point gets a probability of belonging to each cluster
- **Disjunctive clustering**: each point can belong to multiple clusters.
- **Density-based clustering**: look for dense regions of the space (E.g. DBSCAN)

Summary

- **Clustering** : unsupervised approach to group similar data points together.
- **Evaluate** clustering algorithms based on the **shape** of the cluster, the **stability** of the results, the consistency with **domain knowledge**.
- Stochastic (different solutions which each iteration)
- **Hierarchical clustering** using top-down / bottom-up approaches, various **linkage** functions
- **K-means clustering**