Written by Yarden Nativ, mail: nativyardenac@gmail.com

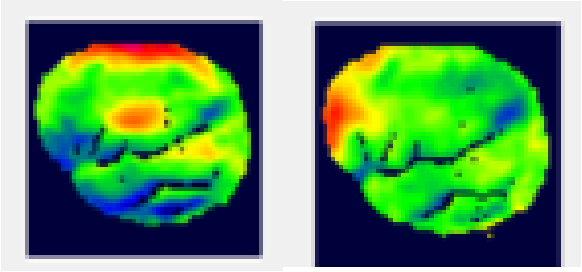
10th Recitation 01.06.23

Clustering

Class Discussion:

In the following images there are maps of neural recording using Voltage Sensitive Dye Imaging representing the membrane potential of neurons in the primary visual cortex in response to a small black dot stimulus. The map is color coded so that hot colors represent higher activity (higher membrane depolarization of the neurons population) and cold colors lower activity.

1. If one would like to computationally compare the activities in the ROIs of the stimulus, how can he use it?
2. How to determine if a pixel is outlier or not?
3. How to determine which of the possible ROIs is the right one?
4. Should the offered ROIs be the same sizes?



The above-mentioned example is only one of very common clustering problems in neuroscience. In this class, the clustering methods we will get acquainted with are very basic methods and commonly used in the field (and not always most efficient).

Definitions

Clustering is A way of grouping together data samples that are similar in some way - according to some criteria.

Clustering is a form of ***unsupervised learning*** – you generally don't have examples demonstrating how the data *should* be grouped together.

In order to cluster, we need to define 'similarity', but there is no single answer – it depends on what we want to find or emphasize in the data. The similarity measure is often more important than the clustering algorithm used – don't overlook this choice!

In the methods we will study, clusters will use the distances between samples in order to cluster them. Possible distance functions:

- Euclidean distance: $d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$
- Squared Euclidean distance: $d(x,y) = \sum_i (x_i - y_i)^2$
- City-block (Manhattan) distance: $d(x,y) = \sum_i |x_i - y_i|$
- Pearson linear correlation $d(x,y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}$

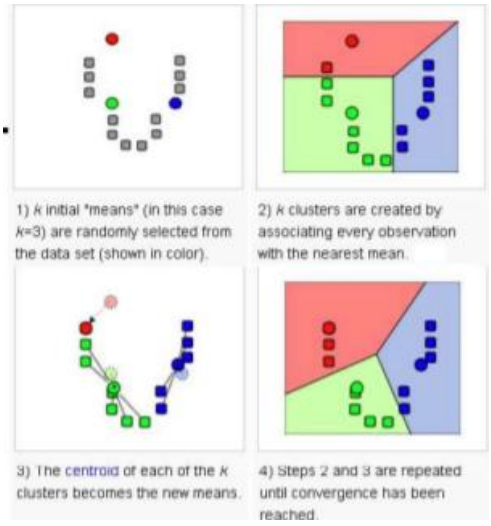
Class Discussion:

How would you use the above-mentioned distances if you need to cluster samples which have duration (for example bold signals, LFP signals, spike trains…)?

K-means

1) Choose the number of clusters k
2) Initialize cluster centers $\mu_1, \dots \mu_k$
    - Choose the centers from known distribution
    - Or randomly assign points to clusters and take means of cluster
3) For each data point, compute the cluster center it is closest to (using some distance measure) and assign the data point to this cluster:
   Sample x_i belongs to cluster I with the center m_i if $d(x_i, m_i) < d(x_i, m_{j \neq i})$
4) Re-compute cluster centers (mean of data points in the cluster you found)
5) Stop when there are no new re-assignments.
6) Re-run the analysis using randomly different initial points with “enough iterations”

Illustration of the method:



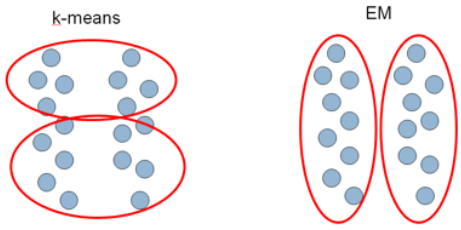
Some important notes:

- There are various methods to define distance
- Disadvantages of K-means:
  - Need to specify number of clusters (k) in advance. What if it is unknown?
  - The result may depend on the initial clusters.
  - The algorithm works best on data which contains spherical clusters; clusters with other geometry may not be found.
  - Can't handle outliers and very noisy data
  - Applicable only when mean is defined (e.g. not for categorical data).
- Advantages:
  - Low computing complexity.
  - Easy to implement.

GMM (MOG)

A method from the family of model based clustering in which the algorithm optimizes a probabilistic model criterion. The model based clustering is usually done by the Expectation Maximization (EM) algorithm. EM provides soft decision in contrast to the hard decision (p=1 or p=0) k-means – each point belongs with some probability to all clusters.

Illustration:



GMM/MOG Algorithm:

1) Define the expectation $p(x_n, g_n)$, the probability of sample x_n to belong to the gaussian g_k: $p(x_n, g_k) = \frac{g_k(x_n)}{\sum_{i=1}^{k} g_i(x_n)}$
2) Define the weights: $w(x_n, g_k) = \frac{p(x_n, g_k)}{\sum_{i=1}^{N} p(x_i, g_k)}$
3) Optimize the centers of g_k by the gaussian covariance: $m_k = \sum_{i=1}^{N} w(x_i, g_k) x_i$, $V_k = \sum_{i=1}^{N} w(x_i, g_k) \cdot (x_i - m_k) \cdot (x_i - m_k)^T$
4) If m_k or V_k changed, repeat expectation step. Else end.

Written by Yarden Nativ, mail: nativyardenac@gmail.com

Class exercise:

Exam 2007: One thousand elves, dwarves, ogres and goblins are assessed using 50 “almost” normal parameters (heights, ear shape, weight, decay of teeth, etc.). What should be done to identify one “typical” member of each of the species?

a. K-Means followed by PCA.
b. PCA followed by K-Means.
c. Mutual information followed by maximum likelihood estimator (MLE).
d. Maximum likelihood estimator (MLE) followed by mutual information.

Solution:

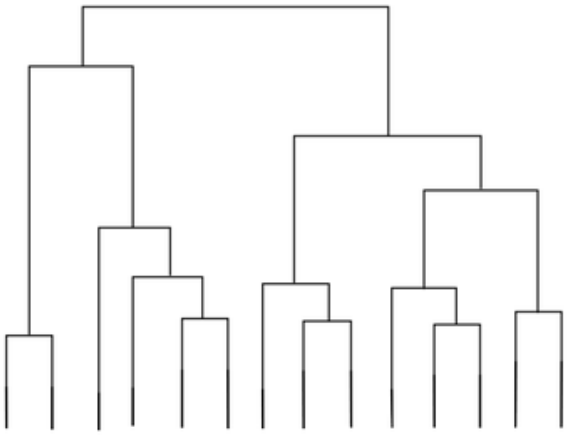
First, we should reduce the dimensionality, and then to divide into the four groups- so first PCA and then K-means (b).

Agglomerative clustering

Algorithm:

1. Start from each point as its own cluster
2. Merge the two closest clusters using the defined distance measure
3. Repeat until all data points are merged into a single cluster.

Therefore, we need to define in advance a distance measure and a similarity measure (to merge closest clusters), but no need of the number of clusters and an initial assignment of data.



Similarity measures are commonly choose by defining inter-cluster distance:

- Single-linkage: the least distance of the closest pair.
- Complete-linkage: the least distance of the furthest pair.
- Average-linkage: the least distance of the average of all pairs

Written by Yarden Nativ, mail: nativyardenac@gmail.com

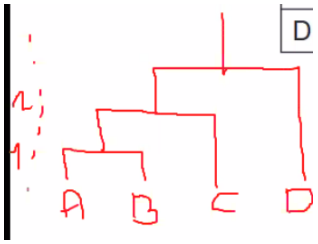
Class exercise:

Use single link and complete link agglomerative clustering to group the data escribed in this distance matrix. Show the dendrograms.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Solution:

We cluster by the smallest distances between the dots. For example, the smallest distance between each of the dots A,B,C and D is between C and D- 3.



In complete link, after we link A and B which are the closest, we need to check three possible clusters (A+B+C, A+B+D, C+D) and see what is the furthest possible pair between the points, and so we see that we prefer to cluster C and D together: