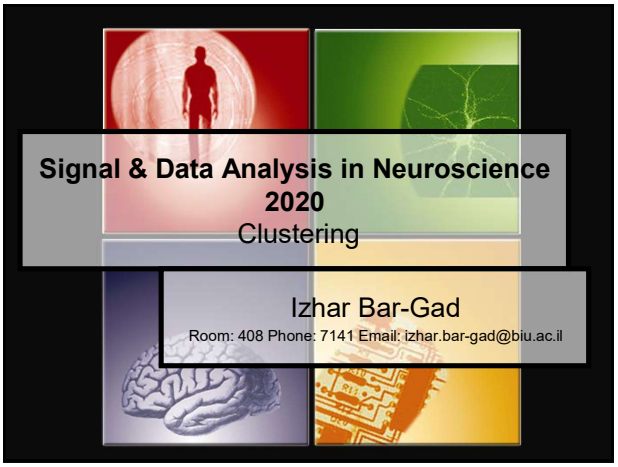
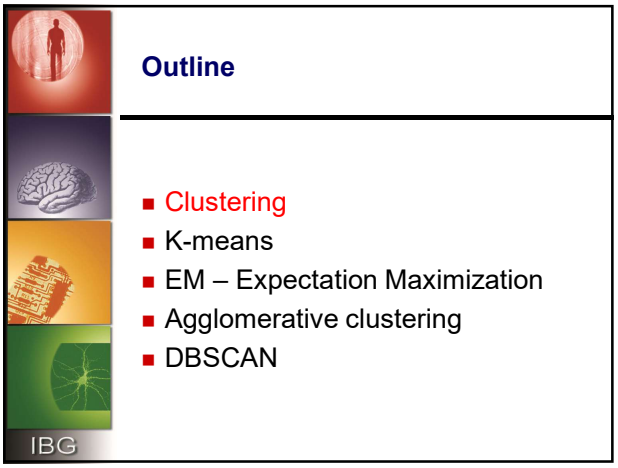
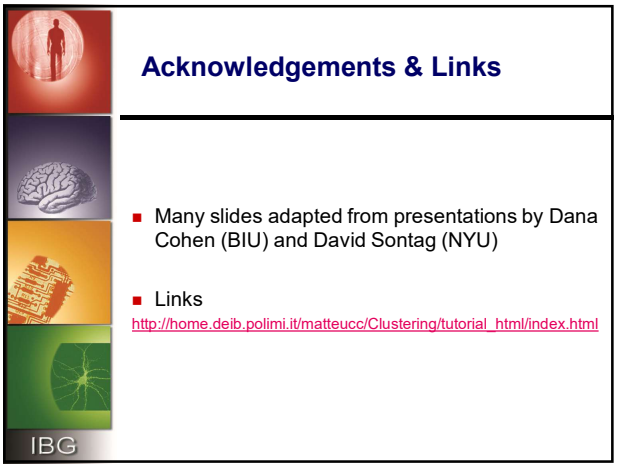

Signal & Data Analysis in Neuroscience
2020
Clustering
Izhar Bar-Gad
Room: 408 Phone: 7141 Email: izhar.bar-gad@biu.ac.il

1


Outline
Clustering
K-means
EM – Expectation Maximization
Agglomerative clustering
DBSCAN
IBG

2


Acknowledgements & Links
Many slides adapted from presentations by Dana Cohen (BIU) and David Sontag (NYU)
Links
http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/index.html
IBG

3

Clustering

Clustering of **data** is a method by which large sets of data is **grouped** into clusters of smaller sets of similar data.

Example:

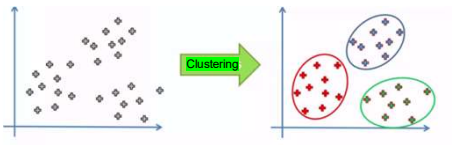
The balls of same color are clustered into a group as shown below :

Thus, clustering means grouping of data or dividing a large data set into smaller data sets of some similarity.

IBG

4

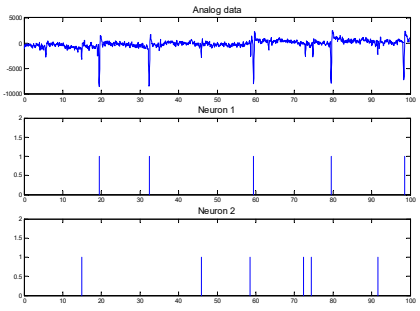
Clustering



IBG

5

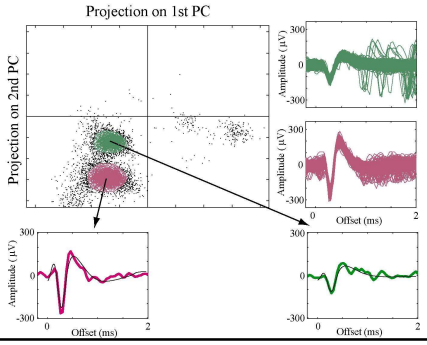
Example: Spike sorting I



IBG

6

Example: Spike sorting II



IBG

7

Classification vs. Clustering

- Classification
  - Supervised learning
  - Partition examples into groups according to pre-defined categories/classes
  - Requires labeled data for training
- Clustering
  - Unsupervised learning
  - Partition examples into groups when no pre-defined categories/classes are available
  - Only instances required, but no labels

IBG

8

A “good” clustering algorithm

- Internal criterion:
  - The intra-cluster similarity is high
  - The inter-cluster similarity is low

depends on representation & similarity measure

- External criterion:

The quality of a clustering is also measured by its ability to discover hidden patterns or latent classes or in comparison to “gold standard”

IBG

9

How hard is clustering?

- Brute force: consider all possible clusters, and pick the one that has best inter and intra cluster distance properties.
- Given ***n*** points tested in in ***k*** clusters, the number of different configurations is:

$$\frac{k^n}{k!}$$

- Enumerating using brute force is too hard leading to iterative optimization algorithms

IBG

10

Clustering methods

- **Hierarchical**
  - Agglomerative (bottom-up)
  - Divisive (top-down)
- **Partitioning**
  - K-means
  - Mixture of Gaussians

IBG

11

Outline

- Clustering
- K-means
- EM – Expectation Maximization
- Agglomerative clustering
- DBSCAN

IBG

12

K-means

An iterative clustering algorithm:

- **Initialize**: Pick *K* random cluster centers
- **Alternate:**

1. Assign data points to closest cluster center
2. Change the cluster center to the average of its assigned points

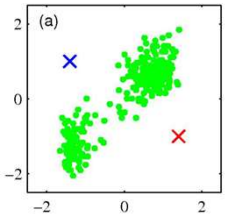
- **Stop:** No assignments change occurs

IBG

13

K-means – example 1

- Initialize: Pick 2 random cluster centers

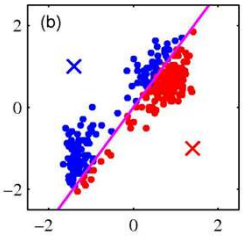


IBG

14

K-means – example 2

- Alternate 1: Assign all data points to closest center

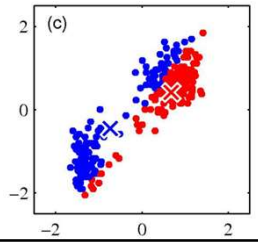


IBG

15

K-means – example 3

- Alternate 2: Change cluster center to the average of the assigned points

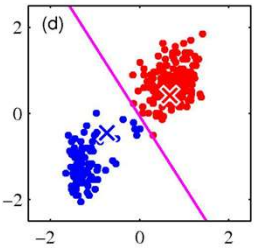


IBG

16

K-means – example 4

- Repeat alternate stages.



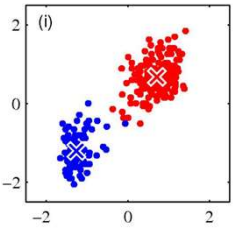
IBG

17

K-means – example 5

- Finish: Convergence.

K means is guaranteed to converge !



IBG

18

K-mean – formal description

- Start with random guess of where the K cluster centers $\mathbf{m}_1 \cdots \mathbf{m}_K$ are
- Repeat the following until cluster centers stop changing:
  - assign each data point to the nearest cluster:

$$p(n,k)=1 \text{ if data point } \mathbf{x}^{(n)} \text{ is closer to } \mathbf{m}_k \text{ than to any other } \mathbf{m}_{j\neq k}$$

  - move each cluster center to the mean of all data points assigned to it:

$$\mathbf{m}_k = \frac{\sum_n p(n,k)\mathbf{x}^{(n)}}{\sum_j p(j,k)}$$

← Vector sum of all data points assigned to cluster k

← Count of all data points assigned to cluster k

$$\mathbf{m}_k = \sum_n w(n,k)\mathbf{x}^{(n)} \quad \text{where} \quad w(n,k) \triangleq \frac{p(n,k)}{\sum_j p(j,k)}$$

IBG

19

K-means converges

Objective

$$\min_{\mu}\min_{C} \sum_{i=1}^{k} \sum_{x\in C_i} |x-\mu_i|^2$$

1. Fix μ, optimize C:

$$\min_{C} \sum_{i=1}^{k} \sum_{x\in C_i} |x-\mu_i|^2 = \min_{c} \sum_{i}^{n} |x_i - \mu_{x_i}|^2$$

Step 1 of kmeans

2. Fix C, optimize μ:

$$\min_{\mu} \sum_{i=1}^{k} \sum_{x\in C_i} |x-\mu_i|^2$$

– Take partial derivative of μ_i and set to zero, we have

$$\mu_i = \frac{1}{|C_i|} \sum_{x\in C_i} x$$

Step 2 of kmeans

Kmeans takes an alternating optimization approach, each step is guaranteed to decrease the objective – thus guaranteed to converge

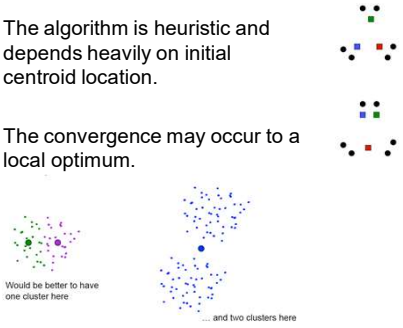
[Slide from Alan Fern]

IBG

20

K-mean – heuristic algorithm

- The algorithm is heuristic and depends heavily on initial centroid location.
- The convergence may occur to a local optimum.



IBG

21

Distances

The results of the clustering depend on the distance

- Euclidean distance (L_2 norm):

$$L_2(\vec{x},\vec{x}') = \sum_{i=1}^{m}(x_i - x_i')^2$$

- L1 norm distance:

$$L_1(\vec{x},\vec{x}') = \sum_{i=1}^{m}|x_i - x_i'|$$

IBG

22

K-means

Strength of the K-means:

• *Efficient: $O(tkn)$, **n** objects, **k** clusters, and **t** iterations. $k, t << n$.*

Weakness of the k-means:

• *Requires mean, what about categorical data?*

• *Need to specify **k**, in advance.*

• *Unable to handle noisy data and outliers*

•*Not suitable for clusters of non-convex shapes.*

In an Euclidian world a convex shape is a set of points containing all line segments between each pair of its points

IBG

23

Soft (fuzzy) K-mean

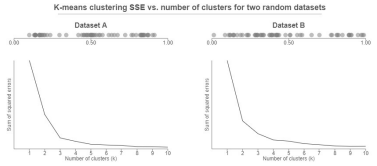
- Clustering typically assumes that each instance is given a “hard” assignment to exactly one cluster.
- Does not allow uncertainty in class membership or for an instance to belong to more than one cluster.
- *Soft clustering* gives probabilities that an instance belongs to each of a set of clusters.
- Each instance is assigned a probability distribution across a set of discovered categories (probabilities of all categories must sum to 1).

IBG

24

Number of clusters

- The number of clusters (K) is unknown and typically cannot be set unambiguously.
- A simple method is the “elbow” method which locates the point of change in the additional reduction of “within cluster sum of squares”.



IBG

25

Number of clusters

The Silhouette method - how similar a point is to its own cluster (cohesion) compared to other clusters (separation)

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

and

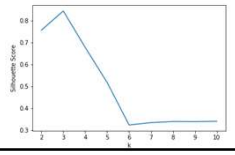
$$s(i) = 0, \text{ if } |C_i| = 1$$

For each data point $i \in C_i$ (data point i in the cluster C_i), let

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

For each data point $i \in C_i$, we now define

$$b(i) = \min_{i \neq j} \frac{1}{|C_j|} \sum_{j \in C_j} d(i, j)$$



(wikipedia.org)

IBG

26

Outline

- Clustering
- K-means
- EM – Expectation Maximization
- Agglomerative clustering
- DBSCAN

IBG

27

A better algorithm: Mixture-of-Gaussians clustering

When the data vectors are clustered, it is more appropriate to fit a distribution with multiple peaks. Consider the mixture-of-Gaussians distribution:

$$p(\mathbf{x}; \mathbf{m}_{1\ldots K}, V_{1\ldots K}) = \frac{1}{K}\sum_k g_k(\mathbf{x}; \mathbf{m}_k, V_k)$$

mixture distribution

Gaussian distributions, with means and covariances $\mathbf{m}_k, V_k$

How do we fit such a distribution to a set of data vectors $\mathbf{x}^{(1)}\ldots\mathbf{x}^{(N)}$? If we knew which Gaussian is "responsible" for each data vector, we could compute the mean and covariance separately for each Gaussian – from the vectors it is responsible for. This suggests the following iterative algorithm (the EM algorithm):

Iterate the following two steps until convergence:

Expectation (E-step): Compute P(x_i | *E(g)*) for each example given the current model, and probabilistically re-label the examples based on these posterior probability estimates.

Maximization (M-step): Re-estimate the model parameters from the probabilistically re-labeled data.

IBG

28

Expectation Maximization

1. Compute the probability p(n,k) that data point n came from Gaussian k, and the normalized weights w(n,k) which sum to 1 for each Gaussian:

$$p(n,k) = g_k(\mathbf{x}^{(n)}) \Big/ \sum_j g_j(\mathbf{x}^{(n)}) \qquad w(n,k) = p(n,k) \Big/ \sum_j p(j,k)$$

2. Re-compute the mean and covariance of all data points that Gaussian k is responsible for, using w(n,k) as weights:

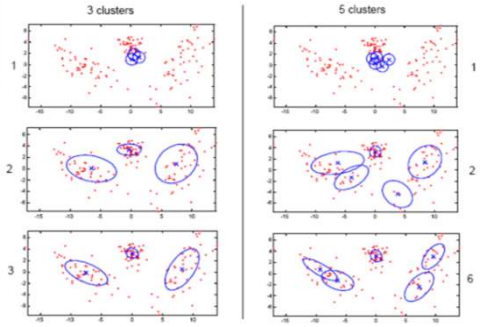
$$\mathbf{m}_k = \sum_n w(n,k)\mathbf{x}^{(n)} \qquad V_k = \sum_n w(n,k)\left(\mathbf{x}^{(n)} - \mathbf{m}_k\right)\left(\mathbf{x}^{(n)} - \mathbf{m}_k\right)^T$$

3. Repeat until $\mathbf{m}$, V no longer change.

IBG

29

Mixture of gaussian - example



IBG

30

Comparison of the two algorithms

In both cases, we compute a quantity p(n,k) that tells us how well data point n fits in cluster k. Then we compute the normalized weights

$$w(n,k) = p(n,k) \Big/ \sum_j p(j,k)$$

and re-compute the cluster centers according to weighted center-of-mass calculation

$$m_k = \sum_n w(n,k)\mathbf{x}^{(n)}$$

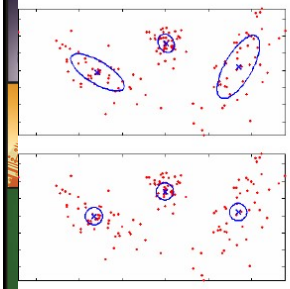
There are two differences:

1. In K-means the "fit" p is either 1 or 0, depending on which is the nearest cluster; In MOG, the values of p vary continuously between 0 and 1, and correspond to probabilities
2. In MOG we also re-compute the covariance matrix, which in turn affects how we determine the fit of data points to clusters; In K-means, the fit is always computed in the same way, corresponding to the assumption of circular clusters

31

Mixture-of-Gaussians vs. K-means clustering

The results are similar when the clusters are well-separated and roughly circular

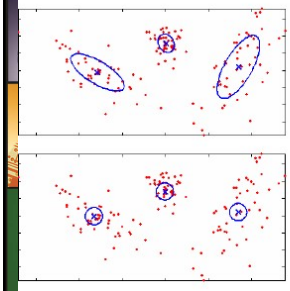


IBG

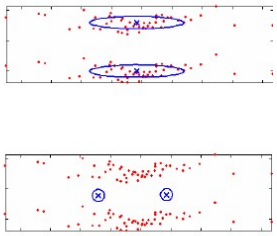
32

Mixture-of-Gaussians vs. K-means clustering

The results are similar when the clusters are well-separated and roughly circular



But for more complex problems K-means can be fooled more easily



IBG

33

Outline

- Clustering
- K-means
- EM – Expectation Maximization
- Agglomerative clustering
- DBSCAN

IBG

34

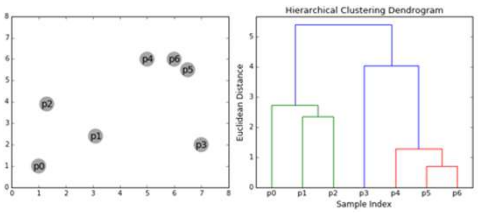
Agglomerative clustering

- Bottom-up approach to hierarchical clustering
- Neighboring points generate a common cluster iteratively building larger clusters.
- Initialization: Each instance in its own cluster.
- Repeat:
  - Pick the two closest clusters
  - Merge them into a new cluster
- Stop: there is only one cluster
- Produces a family of clusterings represented by a dendrogram

IBG

35

Agglomerative clustering Example

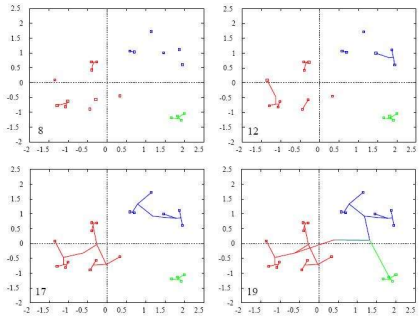


(kdnuggets.com)

IBG

36

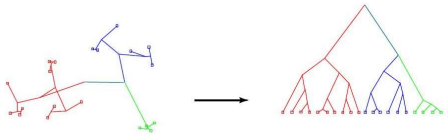
Agglomerative clustering Example



IBG

37

Agglomerative clustering Example



Data with clustering order and distances

Dendrogram representation

IBG

38

Agglomerative clustering Distances

- How should we define “closest” for clusters with multiple elements?
  - Closest pair (single-link clustering)
  - Furthest pair (complete-link clustering)
  - Average of all pairs
- Different choices create different clustering behaviors

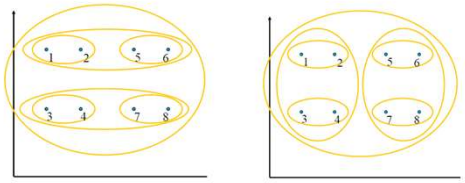
IBG

39

Agglomerative clustering Distances

Closest pair
(single-link clustering)

Farthest pair
(complete-link clustering)

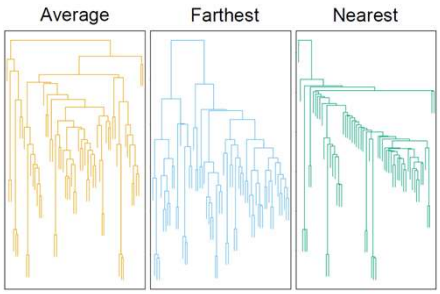


[Pictures from Thorsten Joachims]

IBG

40

Agglomerative clustering Distances



Mouse tumor data from [Hastie *et al.*]

IBG

41

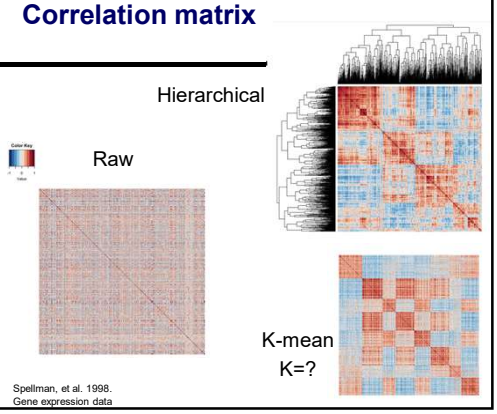
Agglomerative clustering Questions

- Will agglomerative clustering converge?
  - To a global optimum?
- Will it always find the true patterns in the data?
- How many clusters to pick?

IBG

42

Correlation matrix



43

Outline

- Clustering
- K-means
- EM – Expectation Maximization
- Agglomerative clustering
- DBSCAN

44

DBSCAN

- Density-based spatial clustering of applications with noise (DBSCAN) – Ester et al. 1996
- Discovers clusters of arbitrary shape
  - Group together points in high-density
  - Mark as outliers the points that lie alone in low-density regions
- Two main parameters:
  - **ε (epsilon)** - radius of the *neighborhood region*
  - ***minPoints*** - the minimum number of points that should be contained within that neighborhood.

45

- Epsilon neighborhood (N_ε) : set of all points within a distance 'ε'.
- Core point : A point that has at least 'minPoint' (including itself) points within it's N_ε .
- Direct Density Reachable (DDR) : A point *q* is directly density reachable from a point *p* if *p* is core point and *q* ∈ *N*ε .
- Density Reachable (DR) : Two points are DR if there is a chain of DDR points that link these two points.
- Border Point: Point that are DDR but not a core point.
- Noise : Points that do not belong to any point's N_ε .

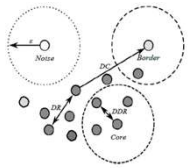


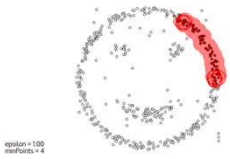
Figure 1: *DBSCAN* clustering with *minPoints* = 4

IBG

46

DBSCAN - algorithm

- Arbitrarily pick a point in the dataset which is not marked.
  - If there are at least **minPoint** points within a radius of **ε** to the point then we consider all these points to be part of the same cluster.
  - Expand cluster by breadth first repeating the neighborhood calculation for each neighboring point

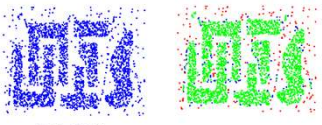


(kdnuggets.com)

IBG

47

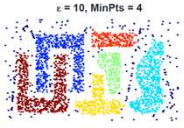
DBSCAN



Original Points

Point types: core, border and outliers

ε = 10, MinPts = 4



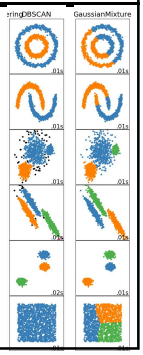
Clusters

IBG

48

DBSCAN vs. Gaussian mixtures

- Advantages
  - Arbitrary distributions
  - Deals well with noise and outliers
- Disadvantages
  - Varying densities
  - Parameter dependence



(towardsdatascience.com)

IBG

49