Clustering – K means

Rohit Budhiraja

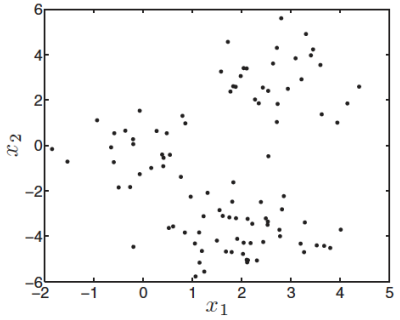
Machine Learning for Wireless Communications (EE798L)

March 2, 2024

Agenda of today's class

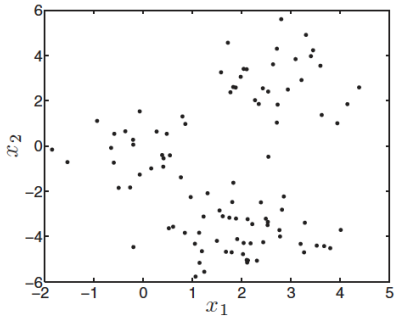
- We had training data with attributes $\mathbf{x}_n$ and labels t_n
  - Learning is called supervised
- What if do not have training data and we have to do a machine learning task
  - Learning is called unsupervised, and we will learn it using clustering example
- Discuss un-supervised K-mean clustering technique
- Will apply K-mean clustering to user scheduling in wireless systems
- Will discuss limitations of K-mean clustering and a technique to solves it
  - Ref: Chap 6 of FCML

K-Means Clustering (1)



- Above data consists of 100 objects, $\mathbf{x}_1, \ldots, \mathbf{x}_{100}$, each represented by two attributes, $\mathbf{x} = [x_1, x_2]^T$
- We have no class information all of the dots look the same i.e., no labels t_n
- If we have to partition these objects into groups by hand such that groups contained similar objects
  - We might conclude that there are three groups
- By clustering data in this manner, who do we mean?

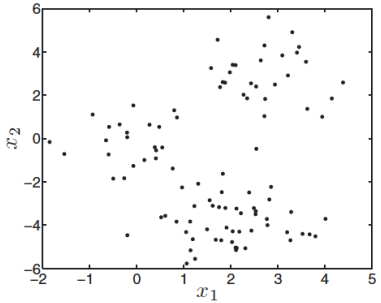
K-Means Clustering (2)



- We mean that similar objects are one that are close to one another in terms of squared distance
  - i and j are similar if $||\mathbf{x}_i - \mathbf{x}_j||^2 = (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)$ is low
- This is a reasonable measure of similarity
  - For data of other types (for example, text), different distance measures would be required

K-Means Clustering (3)

- To develop an algorithm for automatic grouping, we need to formally define a cluster

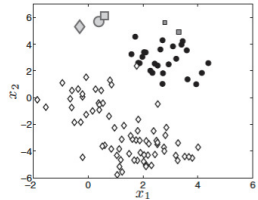


- K-means defines a cluster as a representative point, just like one of the data objects
  - Point is defined as the mean of objects that are assigned to the cluster (hence the name K-means)
- We will use $\boldsymbol{\mu}_k$ to define the mean point for the kth cluster
- Use z_{nk} as a binary indicator variable that is 1 if object n is assigned to cluster k and 0 otherwise
  - Each object has to be assigned to one, and only one cluster, i.e., $\sum_k z_{nk} = 1$
- This leads us to the following expression for mean $\boldsymbol{\mu}_k = \frac{\sum_n z_{nk}\mathbf{x}_n}{\sum_n z_{nk}}$
- $\mathbf{x}_i$ object is assigned to cluster k that gives the minimum value of $(\mathbf{x}_i - \boldsymbol{\mu}_k)^T(\mathbf{x}_i - \boldsymbol{\mu}_k)$

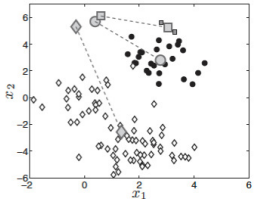
K-Means Clustering – iterative algorithm

- We recap the K means process
  - Clusters are defined as the centres of the points assigned to them and
  - Points are assigned to their closest clusters
- If we know the clusters, $\boldsymbol{\mu}_1 \ldots, \boldsymbol{\mu}_K$, we can compute the assignments
  - But without the assignments we cannot compute the clusters
- K-means clustering overcomes this problem with an iterative scheme as follows
- Starting with initial (random) values for the cluster means, $\boldsymbol{\mu}_1 \ldots, \boldsymbol{\mu}_K$
  1. For each data object, $\mathbf{x}_n$, find k that minimises $(\mathbf{x}_i - \boldsymbol{\mu}_k)^T(\mathbf{x}_i - \boldsymbol{\mu}_k)$ (i.e., find the closest cluster mean) and set $z_{nk} = 1$, and $z_{nj} = 0$ for all $j \neq k$
  2. If all of the assignments (z_{nk}) are unchanged from the previous iteration, stop
  3. Update each k with $\boldsymbol{\mu}_k = \frac{\sum_n z_{nk}\mathbf{x}_n}{\sum_n z_{nk}}$
  4. Return to 1

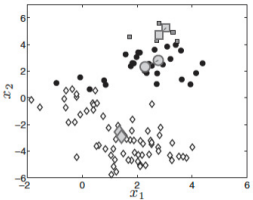
K-Means Clustering - different iterations



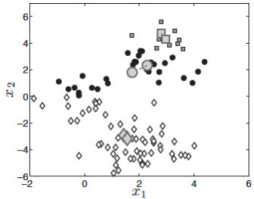
(a) Data and initial random means. Means are depicted by large symbols. Each data object is given the symbol of its closest mean.



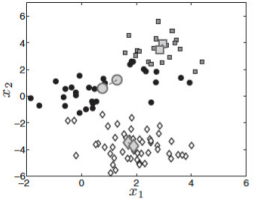
(b) Means updated according to assigned objects.



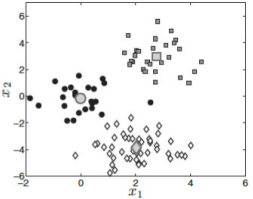
(c) Objects re-assigned to new means and means updated again.



(d) Means updated after three iterations.



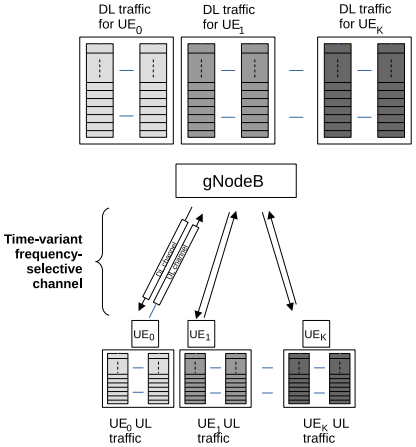
(e) Means updated afterfive iterations.



(f) Means updated after eight iterations. Algorithm has converged.

Application of clustering to user scheduling in wireless systems (1)

- Cellular systems have centralized architecture where BS makes all the decisions
- Base station has to serve multiple users



Application of clustering to user scheduling in wireless systems (2)

- We consider a BS with 2 antennas and two users each with single antenna
- Uplink signal received by the BS

$$\begin{aligned} y_1 &= h_{11}x_1 + h_{12}x_2 + n_1 \\ y_2 &= h_{21}x_1 + h_{22}x_2 + n_2 \\ \mathbf{y} &= \mathbf{H}\mathbf{x} + \mathbf{n} \end{aligned}$$

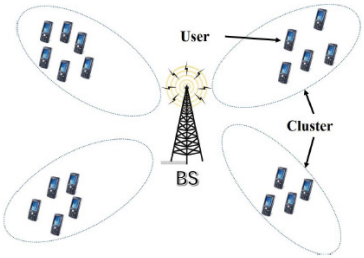
- Tx signal $\mathbf{x} = [x_1, x_2]^T$, rx signal $\mathbf{y} = [y_1, y_2]^T$, and noise $\mathbf{n} = [n_1, n_2]^T$
- Channel

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 \end{bmatrix}$$

- $\mathbf{h}_1$ and $\mathbf{h}_2$ are channels of two users
- Use a zero forcing receiver to detect $\mathbf{x}$ from $\mathbf{y}$ i.e., $(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ or $\mathbf{H}^{-1}$ for a square $\mathbf{H}$
- Channel setting - 1: $\mathbf{h}_1$ and $\mathbf{h}_2$ of two users are linearly dependent – ZF receiver will not work
- Channel setting - 2: $\mathbf{h}_1$ and $\mathbf{h}_2$ of two users are $[1\ 0]^T$ and $[0\ 1]^T$ then $\mathbf{y} = \mathbf{x} + \mathbf{n}$
- Second channel setting is always desirable – will not practically observe
- It is necessary that $\mathbf{h}_1$ and $\mathbf{h}_2$ are linearly independent

Application of clustering to user scheduling in wireless systems (2)

- Consider an example cellular setting:



- BS clusters these users based on the estimate channels – can use K mean clustering
- Schedules two users from different clusters – ensures that their are channels are linear independent

Clustering in 5G wireless systems - example

Learning-Assisted User Clustering in Cell-Free Massive MIMO-NOMA Networks

Quang Nhat Le, Van-Dinh Nguyen, Nam-Phong Nguyen, Symeon Chatzinotas, Octavia A. Dobre, and Ruiqin Zhao

K-Means Clustering -convergence

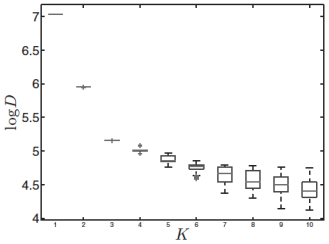
- This iterative scheme is guaranteed to converge to a local minimum of the following quantity

$$D = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)^T \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)$$

- Above can be interpreted as total distance between the objects and their respective cluster centres.
- However, it is not guaranteed to reach the lowest possible value (the global minimum)
- Solution – run algorithm from several random starting points and use solution that gives lowest value of total distance

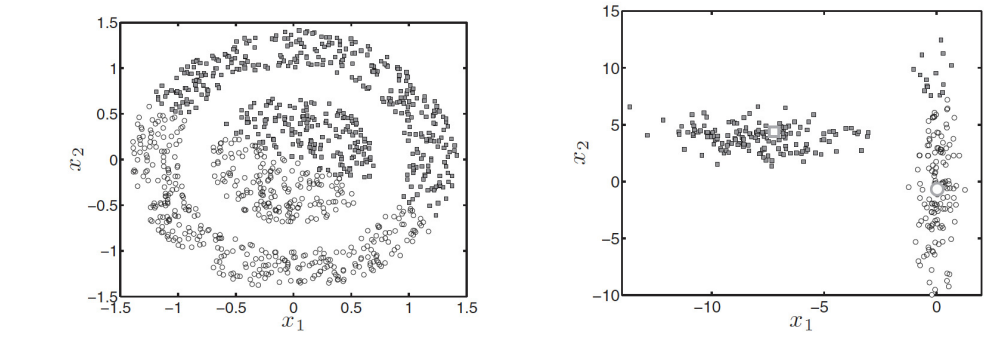
Choosing the number of clusters

- To use K-means, we need to choose the number of clusters K
- Recall that K-means produces a clustering that corresponds to a local minima of D



- Figure shows $\log D$ as K is increased. For each K value, algorithm is initialized randomly 50 times
- It is clear that $\log D$ (and hence D) decreases as K is increased
  - As K increases, large clusters will be broken down into smaller and smaller parts
  - Smaller each cluster, closer each point will get to its cluster mean, reducing its contribution to D
  - For extreme case of $K = N$, $D = 0$ when each cluster contains just one object and $k = \mathbf{x}_n$
- No simple anwers to this problem

Where K-means fails



- Figure shows two datasets where K-means has failed to extract what looks like true cluster structure
- In both cases, objects in true clusters do not conform to our current notion of similarity (distance)
- In first example, data exist in concentric circles
  - Standard K-means can never work in this setting, as, the means of both circles are in the same place
- In second example, clusters are stretched in such a way (check the scaling of the axes) that
  - Objects at the top of right hand cluster are closer to the mean of the left hand cluster
- Means in second plot are shown in this plot as large symbols

Kernelised K-means (1)

- We can extend K-means by using kernel substitution trick
- Key operation in K- means is computation of distance between nth object and kth mean

$$d_{nk} = (\mathbf{x}_n - \boldsymbol{\mu}_k)^T (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

- $\boldsymbol{\mu}_k = \frac{\sum_m z_{mk}\mathbf{x}_m}{\sum_m z_{mk}} = \frac{\sum_{m=1}^N z_{mk}\mathbf{x}_m}{N_k}$, where N_k is number of objects assigned to cluster k. Simplifying:

$$\begin{aligned} d_{nk} &= \left(\mathbf{x}_n - \frac{1}{N_k}\sum_{m=1}^N z_{mk}\mathbf{x}_m\right)^T \left(\mathbf{x}_n - \frac{1}{N_k}\sum_{r=1}^N z_{rk}\mathbf{x}_r\right) \\ &= \mathbf{x}_n^T\mathbf{x}_n - \frac{2}{N_k}\sum_{m=1}^N z_{mk}\mathbf{x}_n^T\mathbf{x}_m + \frac{1}{N_k^2}\sum_{m=1}^N\sum_{r=1}^N z_{mk}z_{rk}\mathbf{x}_m^T\mathbf{x}_r \end{aligned}$$

- Above expression results has data ($\mathbf{x}_n$) only appearing in product terms
- Replace inner products with kernel functions which compute inner products in transformed space
  - Transforming (inner product of) data into a space in which simple algorithm works

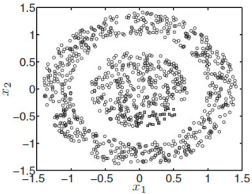
Kernelised K-means (2)

- All that remains is to replace the inner products with kernel functions to give a kernelised distance:

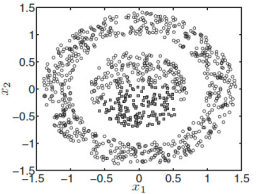
$$
\begin{aligned}
d_{nk} &= \mathbf{x}_n^T \mathbf{x}_n - \frac{2}{N_k} \sum_{m=1}^{N} z_{mk} \mathbf{x}_n^T \mathbf{x}_m + \frac{1}{N_k^2} \sum_{m=1}^{N} \sum_{r=1}^{N} z_{mk} z_{rk} \mathbf{x}_m^T \mathbf{x}_r \\
&= K(\mathbf{x}_n, \mathbf{x}_n) - \frac{2}{N_k} \sum_{m=1}^{N} z_{mk} K(\mathbf{x}_n, \mathbf{x}_m) + \frac{1}{N_k^2} \sum_{m=1}^{N} \sum_{r=1}^{N} z_{mk} z_{rk} K(\mathbf{x}_m, \mathbf{x}_r) \qquad (1)
\end{aligned}
$$

- This distance is purely a function of data and current assignments; cluster means do not appear
- Equation (1) suggests the following procedure for kernelised K-means:
  1. Randomly initialise z_{nk} for each n
  2. Compute $d_{n1}, \cdots, d_{nK}$ for each object using Eq. (1)
  3. Assign each object to the cluster with the lowest d_{nk}
  4. If assignments have changed, return to step 2, otherwise stop

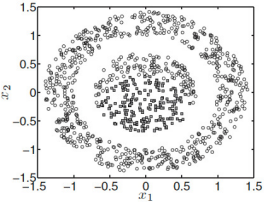
Example of kernelised K-means



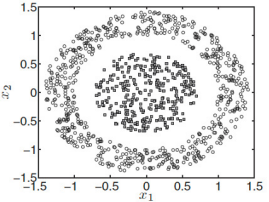
(a) Kernel K-means after one iteration.



(b) After five iterations.



(c) After ten iterations.



(d) At convergence (30 iterations).

- Figure shows result of applying the kernel K-means algorithm to earlier data
- Initialised by assigning all but one object to 'circle' cluster and remaining object to 'square' cluster
  - A Gaussian kernel ($k(\mathbf{x}_n, \mathbf{x}_m) = \exp\{-\gamma(\mathbf{x}_n - \mathbf{x}_m)^T(\mathbf{x}_n - \mathbf{x}_m)\}$) was used with $\gamma = 1$
- Figure shows the assignments 1, 5, 10 and 20 iterations after initialization
- We can cluster any type of data for which a kernel function exists
  - It is hard to find a data type for which there does not

Course feedback

- Total responses - 24 out of 71 students
- Pace of the course - Moderate 16, Fast 7, Slow 1
- Difficulty level so far - Just right 21, too difficult 2, too easy 1
- Learning probability, and less of machine learning - 1
- Research paper, where ML is used on wireless communication - 1
- Connect ML to beamforming , channel estimation and multi user Massive MIMO - 3
- Difficulty in understanding wireless part - 1
- More examples - 1
- Tutor assigned to a group of students - 1
- Attendance policy-1; No attendance policy - 1
- Lots of positive feedback - censored
- Little data, I need to combine data with my common sense - Bayesian learning!