Ex2 - Lior Ziv

Thoretical part - Soft K-Means and EM

1. **Propose a method for calculating the weights $w_{i,c}$ for this soft K-Means version. Provide the full equation(s) and explain your answer.**

   I will propose the next calculation $w_{i,c} = \frac{e^{||x_j - \mu_i||^2}}{\sum_{I=1}^{k} e^{||x_j - \mu_i||^2}}$, in this case for each point i and cluster c we assign a weight $w_{i,c} \in [0,1]$ were the total sum of all the weights per point is 1 $\sum_{c_i \in \text{clusters}} w_{i,c_i} = 1$.

   This formula fits to calculate the weights since it increase when the distance between x_j, μ_i decrease→the numerator grows.

2. **Explain how the soft K-Means problem can be solved using the EM algorithm. Formalize a Likelihood function, and show that it is equivalent to the soft K- Means objective function.**

   In order to answer this question I will assume that:

   - Each centroid is sampled$\mu_c \in N(\mu_c, 1)$.
   - Each point x_i is sampled with $P(x|\mu_c)$

   With this specific assumptions the probability to a point in the set will be written as

   - $P(x_i) = \sum_{c=1}^{k} P(x_i|\mu_c) \cdot p(\mu_c)$

     Now under our assumptions we can translate it parts into:
   - $P(x_i|\mu_c) = N(x|\mu_c) = \gamma_c e^{-\frac{1}{2}||x-\mu_c||^2}$ according to multivariate Gaussian distribution, s.t $\gamma_c = \frac{1}{\sqrt{2\pi}}$
   - $P(\mu_c) = \pi_c$

     Those combined we get - $P(x_i) = \sum_{c=1}^{k} N(x|\mu_c) \cdot \pi_c$

     Now in order to estimate our set of points we need to calculate the maximum likelihood for $\theta = \{\pi_1, \mu_1, ...\pi_k, \mu_k)$, $L(\theta|X) = \prod_{i=1}^{n} p(x_i) = \prod_{i=1}^{n} \sum_{c=1}^{k} N(x_i|\mu_c) \cdot \pi_c$

     Usually in order to find the maximum for this equation I will apply log and derive it, but in this case it is too complicated to in order to solve the problem I will associate it to the weights formula from the previews question and add a parameter to the likelihood.

Let us therefore assume a set Z=$\{z_{11}, z_{12}, \ldots, z_{nk}\}$ such that each z is a weight which represents the probability of point $x_i$to be assign to centroid $\mu_i \rightarrow L(\theta|X,Z) = \prod_{i=1}^{n} \sum_{c=1}^{k} z_{ic} N(x_i|\mu_c) \cdot \pi_c$

To solve the equation we will assume that the optimal solution gives only one $z_{i,c} = 1$ ($x_i \in \mu_c$) and all the rest $z_{i,c'} = 0 (x_i \notin \mu_{c'})$

- $\sum_{i=1}^{n} \sum_{c=1}^{k} z_{ic} N(x_i|\mu_c) \cdot \pi_c = \prod_{i=1}^{n} \prod_{c=1}^{k} [N(x_i|\mu_c) \cdot \pi_c]^{z_{ic}}$

  $LL(\theta|X,Z) = \sum_{i=1}^{n} \sum_{c=1}^{k} log([N(x_i|\mu_c) \cdot \pi_c]^{z_{ic}}) = \sum_{i=1}^{n} \sum_{c=1}^{k} z_{i,c}(log(N(x_i|\mu_c)) + log(\pi_c)) = \sum_{i=1}^{n} \sum_{c=1}^{k} z_{i,c}(log(\gamma_c e^{-\frac{1}{2}||x-\mu_c||^2}) + log(\pi_c)) = \sum_{i=1}^{n} \sum_{c=1}^{k} z_{i,c}(log(\frac{1}{\sqrt{2\pi}}) + (-\frac{1}{2}||x-\mu_c||^2) + log(\pi_c))$

- $[\sum_{i=1}^{n} \sum_{c=1}^{k} z_{i,c} log(\pi_c)] + [\sum_{i=1}^{n} \sum_{c=1}^{k} z_{i,c} \log(\frac{1}{\sqrt{2\pi}})] - [\sum_{i=1}^{n} \sum_{c=1}^{k} z_{i,c}(\frac{1}{2}||x-\mu_c||^2)]$

The second brackets are constant therefore we can ignore it, the first and third are independent. In order to maximize the LL we would like to minimize the third brackets, pay attention that we can treat $z_{i,c}$as $w_{i,c}$.

* Denote c_i*as the closest centroid to x_i

** $\sum_{c_i \in \text{clusters}} w_{i,c_i} = 1$

- $\sum_{i=1}^{n} \sum_{c=1}^{k} z_{i,c}(\frac{1}{2}||x-\mu_c||^2) = \sum_{i=1}^{n} \sum_{c=1}^{k} w_{i,c}(\frac{1}{2}||x-\mu_c||^2) \leq \sum_{i=1}^{n} \sum_{c=1}^{k} w_{i,c}(\frac{1}{2}||x_i-\mu_{c_i^*}||^2)$
  $= \sum_{i=1}^{n} (\frac{1}{2}||x-\mu_{c_i^*}||^2) \sum_{c=1}^{k} w_{i,c} =^{**} \sum_{i=1}^{n} (\frac{1}{2}||x-\mu_{c_i^*}||^2)$

Therfore we can see that maximazing the LL is like minimizing the objective function.

3. We can see the maximizing the last brackets equals to minimizing $\sum_{i=1}^{n} \sum_{c=1}^{k} z_{i,c}(\frac{1}{2}||x-\mu_c||^2)$

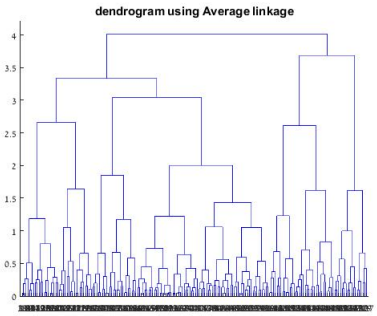
   - I will take the set $\Rightarrow (0,1),(10,1),(10,0),(0,0)$ with initial centroids of (5,1),(5,0)
     first we get the clusters {(10,0),(0,0)} $\in (5,0)$, {(10,1),(0,1)} $\in (5,1)$
     Update the centroids $\frac{10+0}{2} = 5$ $\frac{0+0}{2} = 0 \rightarrow (5,0)$ | $\frac{10}{2} = 5$ $\frac{2}{2} = 1 \rightarrow 5,1$
     We got the same centroids which means the algorithm converged(since next we assign our points to the new centroids which are the same - nothing changes) , those the score we get , $\sum_{i=1}^{2} \sum_{c=1}^{4} 1(||x_i-\mu_c||^2) = 25 + 25 + 25 + 25 = 100$
   - But if we check a better minimal score would be centroids $\rightarrow$(10,0.5),(0,0.5).
     with a score of - $\sum_{i=1}^{2} \sum_{c=1}^{4} 1(||x_i-\mu_c||^2) = 0.25 + 0.25 + 0.25 + 0.25 = 1$

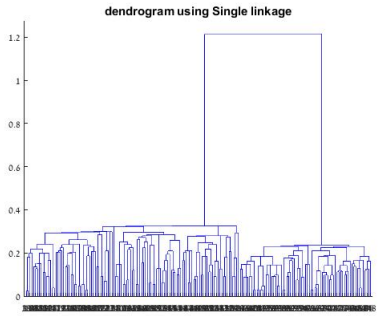
- A way to avoid it is to run the algorithm few times with new initialized centroids and take the centroids which give us the best score over all the iterations, which is what we do in the restartNum task.

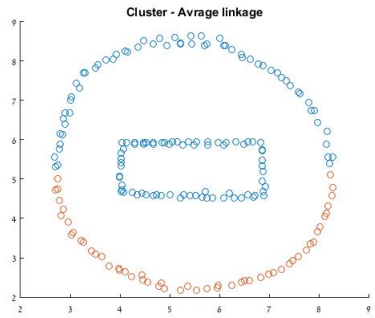
4. In the case we have an optimal solution then we have a global minimal value and since the optimal solution for the soft k-means in that way is the same as the hard we get that the same exaple from the previews question can be applied. Not really sure what is meant to be here..

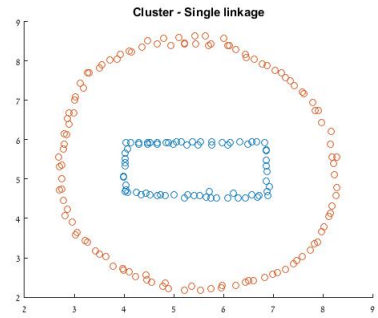
Programming part 1 - Agglomerative clustering

1. Submitted
2. Plots -

- 



- 



- We can see that both the dendogram and the cluster plot are different, since we used different linkage types.

  Average linkage - it is divided different then our eye would divide it, since the algorithm looked for the minimal sum of cluster groups avreage distance(between points inside the cluster).

  Also in the dendogram we can see the seperation into two groups were the elements are more far from each other than in the single linkage.

Single linkage - here we see that since each time only the minimal distance from a point in a cluster sets the tree and not the sum we get that all the outer group is tagged together and the inner group tagged togther.

Also in the dendogram we see that the points are more close togther.

Programming part 2 - K-Means

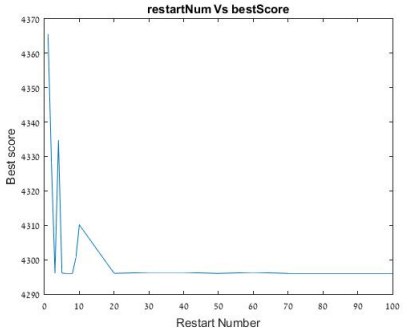
Random restarts

1. Submitted.

2. In the figure we can see that first the algorithm has a larger score (worst in our case - we look for the minimal score), and after restartNum ~>=20 the score converges.

   I can explain it by the fact that restartNum is controlling the number of times we start k-means with the same data but new centroids.

   As we know k-means might get stuck on a local minimal value so if we give it more restarts (new centroids) it will have a better probability to reach a better score→ as restartNum grows the score converges.



Subsampling

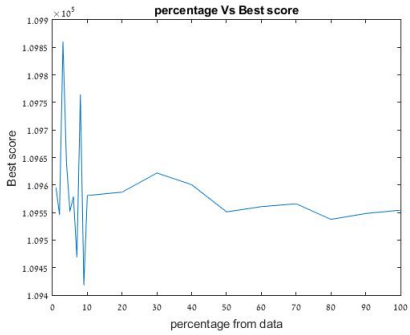
1. submitted.

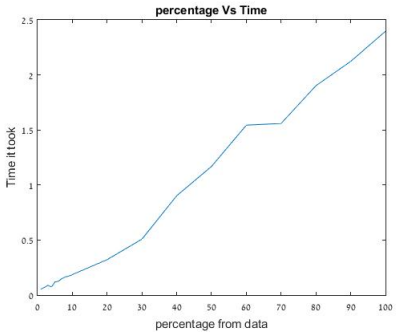
2. In the figure we can see that first the algorithm has a larger score (worst in our case - we look for the minimal score) , and as the percentage increase the score converges.

   I can explain it by the fact that the lower the percentage is the smaller the amount of data which is sent to K- means which causes to the range to choose centeroids from to be biased towards the sampled data.

   But from 50% above we see that the best score stays almost the same, which means that for this set of points ~50% is enough to estimate the centrodis that will give us the best score.

And as a result of sending only % from the data we get better running time.





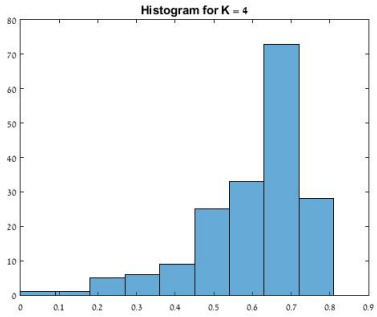
Choosing K

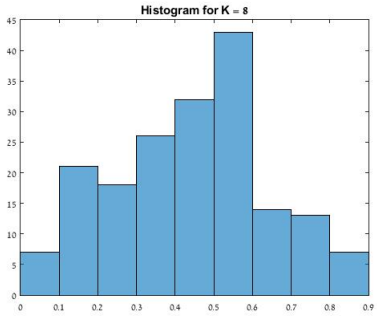
1. submitted.

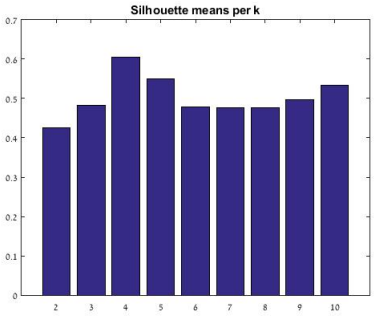
2. **If s ˜ <= 0** it means that $b_i <= a_i$, which means or that the distance of the point from the second cluster is closer ($b_i <$ a_i - bad results) either it is almost the same($a_i \tilde{=} b_i$).

   If s > 0 it means that $b_i >$ $a_i$and as s grows $\rightarrow b_i >> a_i$.

   Therefore I would choose k=4 ,we can see that the S value at 4 is maximal, which means that the distance between the each point and the cluster we choose a_i has the most difference to the next closer cluster group b_i.






Silhouette means per k
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2
3
4
5
6
7
8
9
10