CS446 HW1

Shuzhen Zhang
Net ID: shuzhen2

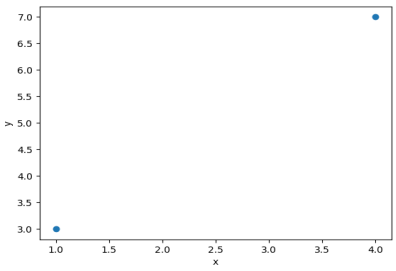
1 PCA:

(a)

- False: By definition of PCA, PCA is looking for finding the axis which minimizes the sum of squared distances from points to their orthogonal projections on that axis rather than the sum of all the vertical distances.
- True
- True
- False: By definition, as we mentioned before, the PCA is looking for two axes which are orthogonal projections with each other.

(b)

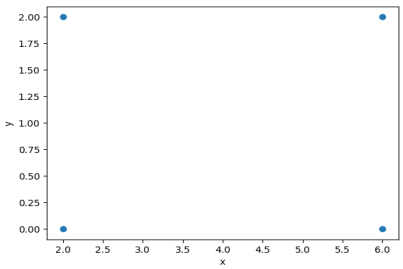
- First, we need to calculate the variance matrix of points$(1,3),(4,7)$
- graph here:



- let X donate our data set such that $\begin{bmatrix}1 & 3\\4 & 7\end{bmatrix}$
- let u donate the mean of X is $\begin{bmatrix}2.5 & 5\end{bmatrix}$
- lets donate X' to be the centered data set that $\begin{bmatrix}-1.5 & -2\\1.5 & 2\end{bmatrix}$
- $\Sigma = X'^{T}@X'/(number(X)-1) = \begin{bmatrix}4.5 & 6\\6 & 8\end{bmatrix}$
- $eigenvalue(\Sigma) = \begin{bmatrix}0 & 12.5\end{bmatrix}$
- corresponding $eigenvector(\Sigma) = \begin{bmatrix}-0.8 & -0.6\\0.6 & -0.8\end{bmatrix}$
- since norm(w)=1, which means we are looking for the largest eigenvalue and corresponding eigenvector, w will be $\begin{bmatrix}-0.6\\-0.8\end{bmatrix}$

(c)

- data: $(2,0),(2,2),(6,0),(6,2)$
- graph here:



- we can re-do the process that we did in part (b)
- donate $X = \begin{bmatrix} 2 & 0 \\ 2 & 2 \\ 6 & 0 \\ 6 & 2 \end{bmatrix}$
- $X' = X - mean(X) = \begin{bmatrix} -2 & -1 \\ -2 & 1 \\ 2 & -1 \\ 2 & 1 \end{bmatrix}$
- then $\Sigma = X'^T @ X/3 = \begin{bmatrix} 5.333 & 0 \\ 0 & 1.333 \end{bmatrix}$
- Therefore, Σ has dimension 2

(d)

- $\Sigma = \begin{bmatrix} 12 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$
- Since Σ is a linearly independent matrix, the eigenvalue of Σ is the number in diagonal, and the optimal value is the biggest eigenvalue of Σ which is 20, then the optimal w will be $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$

2 K-mean

(a)The k-Means method is an unsupervised method. The K-means method is looking to classify data with a given number of clusters. To achieve this goal we have 3 steps :

- initialize the original centers of data
- assign data to different clusters
- calculated new center by Euclidean distance

We repeat the 2 and 3 steps until nothing changes with the assignment of data. In this process, we don't need to train our machine with data or any further modification to our algorithm. Therefore K-Means method is an unsupervised method.

(b)In the equation, we are looking for the minimal distance between x_i and $\mu_1, ...\mu_K$. We will assign x_i to the most recent cluster. However, some exceptions may happen like x_i has an equal distance with $\mu_i, \mu_j : i, j \in K$, then we can be assigned x_i to the first encountered cluster. Therefore, we can make sure the hard assignment is.

(c) If we are asking to provide a soft assignment method, I would do the following.

- arrange all $\{\mu_i, \mu_j...., \mu_k\} \rightarrow \{\mu_1, \mu_2...., \mu_n\} : i, j, k \in K$, and n donate the number of centers which has a same square distance with x_i
- setting a random number generator with range 1$\rightarrow$ n and obtain number l.
- assign x_i to the μ_l

(d) In this problem, I believe 5 would be an optimal number of clusters for two reasons.

- When the number of clusters is smaller than 5 the squared distance between points and clusters is too large to present the real property of data.
- when a number of clusters are larger than 5 we can see the squared distance only has a tiny change. which means 6,7,8,9,10 could be the number of clusters but 5 is better because less cluster means letter computation by the formula $O(KNd)$.

(e) I believe K-Means would not be an efficient algorithm for the graph of data because the blue cluster in inside of the orange cluster. Two situations may happen when we apply K-Means method:

- If orange cluster was assigned entirely to the cluster μ_k, then the blue cluster will also in μ_k
- If there are two or more center inside of orange cluster, The orange cluster and blue cluster will be splits to pieces.

2.1 K-Means 2

(a) Considering our data set D is collected from $\mathbb{R}^2$. The domain of μ_k will also in $\mathbb{R}^2$ because μ_k is the center of cluster in D will have same domain with D.

(b) r_x is a column vector for assigning data x to some cluster μ_k. By the definition of the K-Means method, any data $x \in D$ will belong to one cluster in a single iteration.r_x will looks like $\begin{bmatrix} \mathbf{r}_{x,1} \\ \mathbf{r}_{x,2} \\ \dots \\ \mathbf{r}_{x,k} \end{bmatrix}$, and $\forall k \in K r_{x,k}$ means sending x to μ_k cluster. Therefore, $r_{x,k}$ will have two choices that:

- $r_{x,k} = 1$, if $k = argmin_{k \in \{1,\dots,k\}} \|x - \mu_k\|_2^2$
- $r_{x,k} = 1$, otherwise

That will be the optimal $r_{x,k}$.

(c) When we have a fixed $r_{x,k}$ we can find the optimal μ_k by calculating the first derivative and considering the situation of the first derivative equal to 0:

$$\sum_{x \in D, k \in \{1,\dots,k\}} \frac{1}{2} r_{x,k} \|x - \mu_k\|_2^2 = \sum_{x \in D} r_{x,k}(x - \mu_k) = 0 \tag{1}$$

$$\rightarrow \mu_k = \frac{\sum_{x \in D} r_{x,k} * x}{\sum_{x \in D} r_{x,k}} \tag{2}$$

(d) Lets define

- Data in 2D: $X = \{x^{(1)}, ..., x^{(n)}\}$, n is size of X
- Clusters with initial center: $K = \{\mu_1, \mu_2, ..., \mu_m\}$, m is size of K
- $x^{(i)}$ and μ_k is vector consist axes $\begin{bmatrix} x & y \end{bmatrix} \in \mathbb{R}^2$ and $k \leq m, i \leq n$

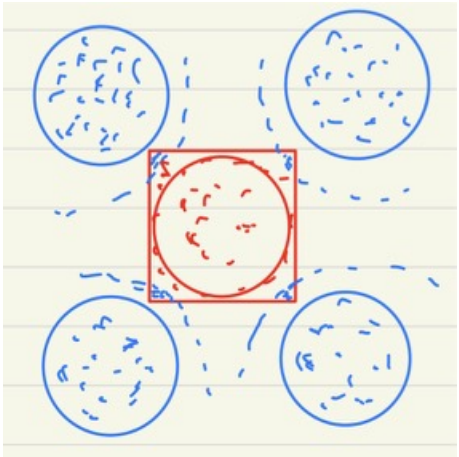
As we discussed previously in class, the two steps are:

- Assigning data X to clusters K, such that $x^{(i)} \rightarrow \mu_k$
- to calculate the average distance inside of cluster and define a new center of the cluster

pseudo-code here:

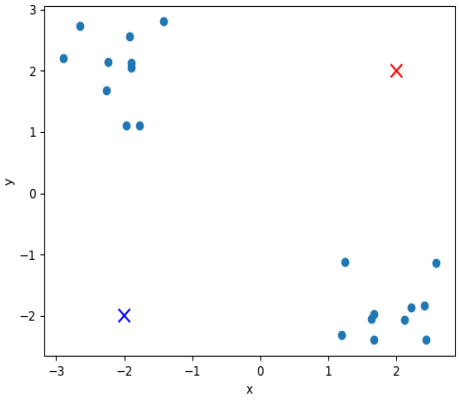
```
# pseudo-code:
# def K-mean():
#     2D vector: clusters (store points)
#     while loop: loop continue if step2(X,K)(new clusters) not equal to clusters
#         renew clusters: clusters=step2(X,K)
#         renew K: K=step3(clusters)
#     return K

# def step3(clusters):
#     mean_K(1d vector with size m)= mean of clusters
#     return mean_K

# def step2(X,K):
#     for i in size of X:
#         minimal_distance=Maximun_distance
#         index= m+1
#         for k in size of K:
#             if distance(x(i),u(k))< minimal_distance{
#                 minimal_distance=distance(x(i),u(k))
#                 index=k
#             }
#         put x(i) in clusters at index
#     return clusters
```

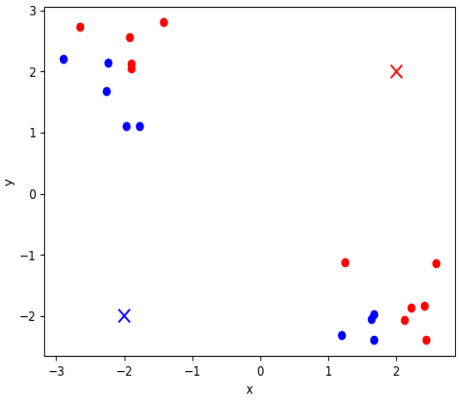
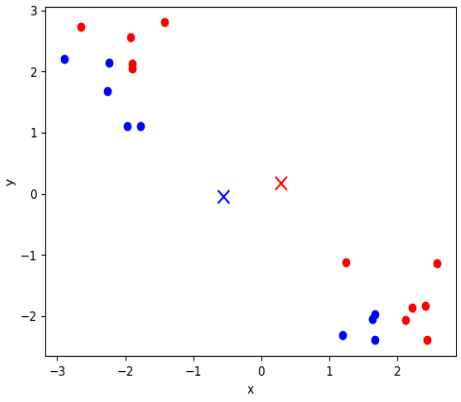
- This algorithm is not guaranteed to converge, if the initial point is just at the center of the local optimal, the algorithm will stop. To solve this problem we can try multiple times with initialization to avoid exceptions. In that case, the K-Means will guaranteed to converge.
- Global optimum also not guaranteed example here:



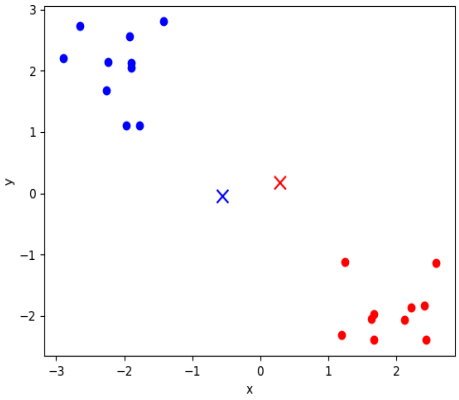
- Suppose there are 5 clusters with 1 square and 4 circles, when we apply the K-Means method, the angle of the square which is outside of the red circle will be split which will cause global optimal failure,

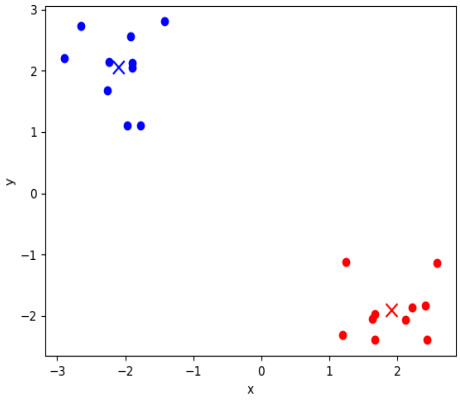
(e) In the question, we have two different clusters and we only need to repeat 2,3 steps twice. The graphs here:

-  initial state: 'x' is center of cluster with $\begin{bmatrix} c1 & c2 \\ 2 & -2 \\ 2 & -2 \end{bmatrix}$

-  

- after first converge of steps 2 and 3: 'x'= $\begin{bmatrix} c1 & c2 \\ 0.2911 & -0.5545 \\ 0.1694 & -0.0518 \end{bmatrix}$

- 



- after second converge of steps 2 and 3: 'x'= $\begin{bmatrix} c1 & c2 \\ 1.9163 & -2.0952 \\ -1.9143 & 2.0540 \end{bmatrix}$, which is our final answer of centers, and my function cost is 243.039505.

3 Gaussian Mixture Models

(a)

- In the class lecture, we defined the GMM that:

$$P(x^{(i)}|\pi,\mu,\sigma) = \sum_{k=1}^{K} \pi_k N(x^i|\mu_k,\sigma_k) \tag{3}$$

- after we log-likelihood this equal:
-

$$\log \Pi_{i\in D} P(x^{(i)}|\pi,\mu,\sigma) = \sum_{i\in D} \log \sum_{k=1}^{K} \pi_k N(x^i|\mu_k,\sigma_k) \tag{4}$$

(b) As we defined and proved in class:

- $\mu_k = \frac{1}{N_k}\sum_{i\in D} r_{ik}x^{(i)}$, N_k is the number of element in cluster k^{th}
- $\sigma_k^2 = \frac{1}{N}\sum_{i\in D} r_{ik}(x^{(i)}-\mu_k)^2$
- $\pi_k = \frac{N_k}{N}$

when $K = 1$, which means we only have one cluster in our data set. Then μ_1, σ_1, π_1 will be:

- $\mu_1 = \frac{1}{N}\sum_{i\in D} x^{(i)}$, $N_k = N$ because all data in the same cluster, and we delete r_{ik} for the same reason.
- $\sigma_1^2 = \frac{1}{N}\sum_{i\in D}(x^{(i)}-\mu_1)^2$
- $\pi_1 = \frac{N_k}{N} = 1$, since $N_k = N$

(c) Let define the $P(z_{i1},...,z_{ik})$ first:

- since $P(z_{ik}=1) = \pi_k$
- then $P(z_{i1},...,z_{ik}) = P(z_i) = \Pi_{k=1}^{K}\pi_k^{z_{ik}}$
- Then the probability distribution underlying Gaussian mixture models will be:
-

$$P(x^i|\pi,\mu,\sigma) = \sum_{z_i} P(x^i|z_i)p(z_i)) = \sum_{zi} \Pi_{k=1}^{K}\pi_k^{z_{ik}} N(x^i|\mu_k,\sigma_k)^{z_{ik}} \tag{5}$$

(d) In the class lecture, we defined the posterior $r_{ik} = P(Z_{ik}|x^{(i)})$

$$P(Z_{ik}|x^{(i)}) = \frac{P(Z_{ik}=1)P(x^{(i)}|Z_{ik}=1)}{\sum_{\hat{k}=1}^{K} P(Z_{i\hat{k}}=1)P(x^{(i)}|Z_{i\hat{k}}=1)} \tag{6}$$

$$= \frac{\pi_k N(x^i|\mu_k, \sigma_k)}{\sum_{\hat{k}=1}^{K} \pi_{\hat{k}} N(x^i|\mu_{\hat{k}}, \sigma_{\hat{k}})} \tag{7}$$

we can see that we can use $N(x^i|\mu_k, \sigma_k)$ wherever possible and Gaussian distribution over $x_i \in \mathbb{R}$ having mean μ_k and σ_2^k

(e) The K-Means and GMM is related in 3 different ways:

- In K-Means method $\sum r_{ik} = 1$, which is a hard assignment but in GMM the sum could be bigger than 1 by the latent variable z_{ik} that allows oval cluster to exist is a soft assignment.
- The variance in the K-Means methods are equal in all clusters, but, in the GMM, the variance of different clusters is different
- π_k is the mixing coefficients that only GMM have which used to present the proportion of each component in GMM.

(f) proof:

$$\lim_{\epsilon \to 0} -\epsilon \log \sum_{k=1}^{K} \exp \frac{-F_k}{\epsilon} = min_k F_k, \epsilon \in \mathbb{R}^+ \tag{8}$$

- let define a p such that $\lim_{\epsilon \to 0} p = \frac{1}{\epsilon} : p \to \inf$
- Then

$$\lim_{\epsilon \to 0} -\epsilon \log \sum_{k=1}^{K} \exp \frac{-F_k}{\epsilon} \tag{9}$$

$$\Rightarrow \lim_{p \to \inf} -\frac{1}{p} \log \sum_{k=1}^{K} \exp -F_k * p \tag{10}$$

$$\Rightarrow -log \lim_{p \to \inf} (\sum_{k=1}^{K} (\exp -F_k)^p)^{\frac{1}{p}} \tag{11}$$

- we can see $(\sum_{k=1}^{K} (\exp -F_k)^p)^{\frac{1}{p}}$ is a norm function with infinity norm:

$$-log \lim_{p \to \inf} (\sum_{k=1}^{K} (\exp -F_k)^p)^{\frac{1}{p}} = -log \lim_{p \to \inf} (\| \exp -F_k \|_p) \tag{12}$$

- we know the infinity norm is looking for the maximum value of the equation.
- then $\lim_{p\to\inf}(\| \exp -F_k \|_p) = \exp - \min_k F_k$
- $-log \lim_{p\to\inf}(\| \exp -F_k \|_p) = \min_k F_k$
- the equation is valid, prof complete.

(g) The 0-temperature limit of GMM will be:

$$\lim_{\epsilon\to 0} \min_u \sum_{x^i\in D} -\epsilon \log \sum_{k=1}^{K} \exp \frac{-(x_i-\mu_k)^2}{\epsilon} \tag{13}$$

- since we proved $\lim_{\epsilon\to 0} -\epsilon \log \sum_{k=1}^{K} \exp \frac{-F_k}{\epsilon} = min_k F_k, \epsilon \in \mathbb{R}^+$ in (f)
- Let $F_k = (x-\mu_k)^2$ we can do a simple replacement that:
-

$$\lim_{\epsilon\to 0} \min_u \sum_{x^i\in D} -\epsilon \log \sum_{k=1}^{K} \exp \frac{-(x_i-\mu_k)^2}{\epsilon} = \min_u \sum_{x^i\in D} \lim_{\epsilon\to 0} -\epsilon \log \sum_{k=1}^{K} \exp \frac{-(x_i-\mu_k)^2}{\epsilon} \tag{14}$$

$$= \min_k \lim_{\epsilon\to 0} -\epsilon \log \sum_{k=1}^{K} \exp \frac{-F_k}{\epsilon} = \min_k F_k = \min_{u,k} \sum_{x^i\in D} (x_i-\mu_k)^2 \tag{15}$$

- We can see this is the form of K-Means