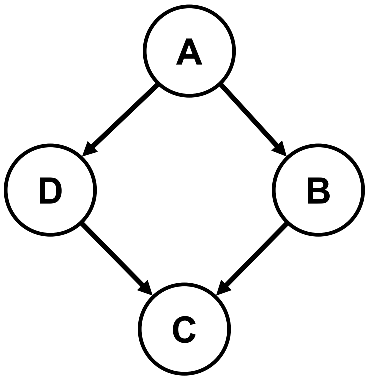
CS559-B: Machine Learning Name: Due: Nov. 21, 2018

Assignment 4

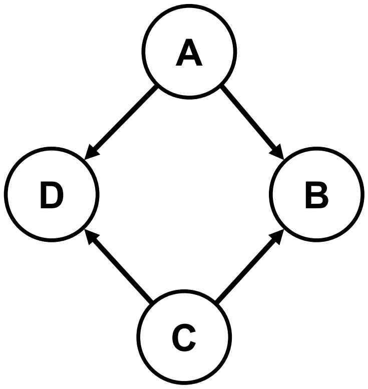
Homework assignments will be done individually: each student must hand in their own answers. Use of partial or entire solutions obtained from others or online is strictly prohibited. Electronic submission on Canvas is mandatory.

1. **Clustering** (10 points) Suppose we clustered a set of N data points using two different clustering algorithms: k-means and Gaussian mixtures. In both cases we obtained 5 clusters and in both cases the centers of the clusters are exactly the same. Can a few (say 3) points that are assigned to different clusters in the kmeans solution be assigned to the same cluster in the Gaussian mixture solution? If no, explain. If so, sketch an example or explain in 1-2 sentences.

   Yes, it is possible that the points assinged in k-means and gaussain can be different. K-means is a hard clustering algorithm i.e it is not based on probability, its based on distance.
   But Gaussian mixture is a soft probabilistic clustering algorithm so it is possible that when we fit a gaussian there might be points in the gaussian center that are clusteres together and the once in the tail may be places in different clusters.



(a)



(b)

2. **Bayesian Networks** (10 points) Do the following statements hold in each of the above networks ? Please explain your reasoning

   - $A \perp C | B, D$
   - $B \perp D | A, C$

   For the first graph:

3. $A \perp C | B, D$:
   True, the path is not active between A and C.

4. $B \perp D | A, C$
   False.
   Given the v-structure on C, once C is observed B,D are not independent.

   For Second:

5. $A \perp C | B, D$:
   False

   Again given the v-structure on B,D. A and C are not independent once B,D are observed.

6. $B \perp D | A, C$:
   True. Since there are no active paths between A and C, this dependence holds.

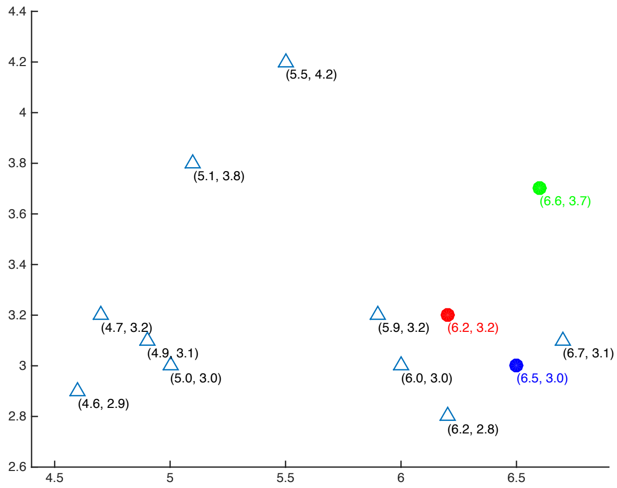


Figure 1: Scatter plot of datasets and the initialized centers of 3 clusters

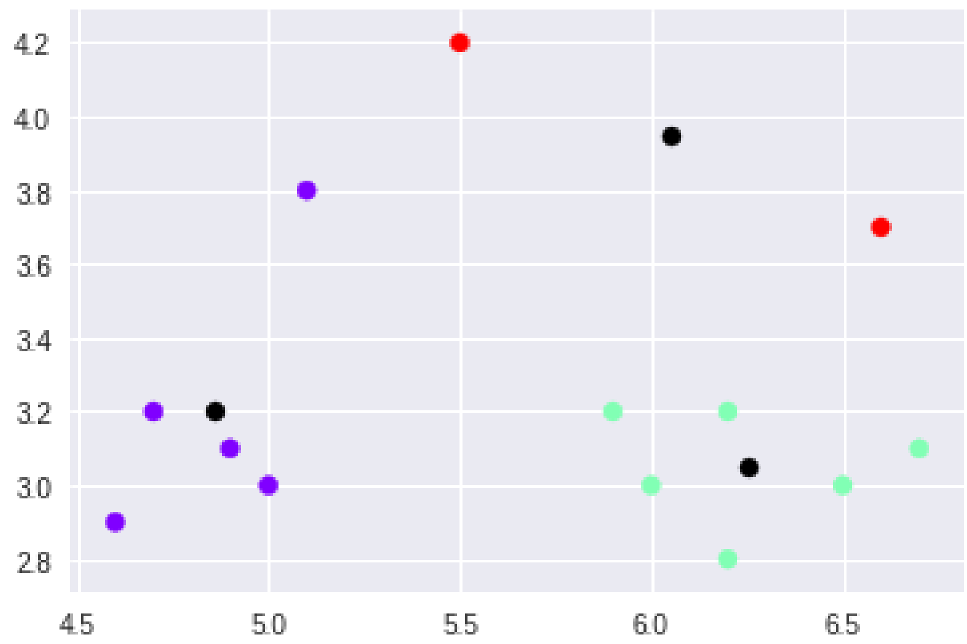
7. **K-means** (30 points) Given the matrix X whose rows represent different data points, you are asked to perform a k-means clustering on this dataset using the Euclidean distance as the distance function. Here k is chosen as 3. The Euclidean distance d between a vector x and a vector y both in $\mathcal{R}^d$ is defined as $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$. All data in X were plotted in Figure 1. The centers of 3 clusters were initialized as $\mu_1 = (6.2, 3.2)$(red), $\mu_2 = (6.6, 3.7)$(green), $\mu_3 = (6.5, 3.0)$(blue).

   (a) Whats the center of the first cluster (red) after one iteration? (Answer in the format of $[x_1, x_2]$, round your results to three decimal places) [5.4 , 3.13]

   (b) Whats the center of the second cluster (green) after two iteration?
   [6.05,3.95]

   (c) Whats the center of the third cluster (blue) when the clustering converges?
   [6.25,3.05]

   (d) How many iterations are required for the clusters to converge?
   3 iterations


4.2
4.0
3.8
3.6
3.4
3.2
3.0
2.8
4.5
5.0
5.5
6.0
6.5

8. **Expectation Maximization (EM)** (50 points) In this question you will implement the EM algorithm for Gaussian Mixture Models. A good read on gaussian mixture EM can be found at this link. A sample dataset for this problem can be downloaded in canvas files. For this problem:

   - n is the number of training points
   - f is the number of features
   - k is the number of gaussians
   - X is an $n \times f$ matrix of training data
   - w is an $n \times k$ matrix of membership weights. $w(i, j)$ is the probability that x_i was generated by gaussian j
   - π is a $k \times 1$ vector of mixture weights (gaussian prior probabilities). π_i is the prior probability that any point belongs to cluster i
   - μ is a $k \times f$ matrix containing the means of each gaussian
   - Σ is an $f \times f \times k$ tensor of covariance matrices. $\Sigma(:, :, i)$ is the covariance of gaussian i

   (a) **Expectation**: Complete the function $[w] = \text{Expectation}(X, k, \pi, \mu, \Sigma)$. This function takes in a set of parameters of a gaussian mixture model, and outputs the membership weights of each data point

   (b) **Maximization of Means**: Complete the function $[\mu] = \text{MaximizeMean}(X, k, w)$. This function takes in the training data along with the membership weights, and calculates the new maximum likelihood mean for each gaussian.

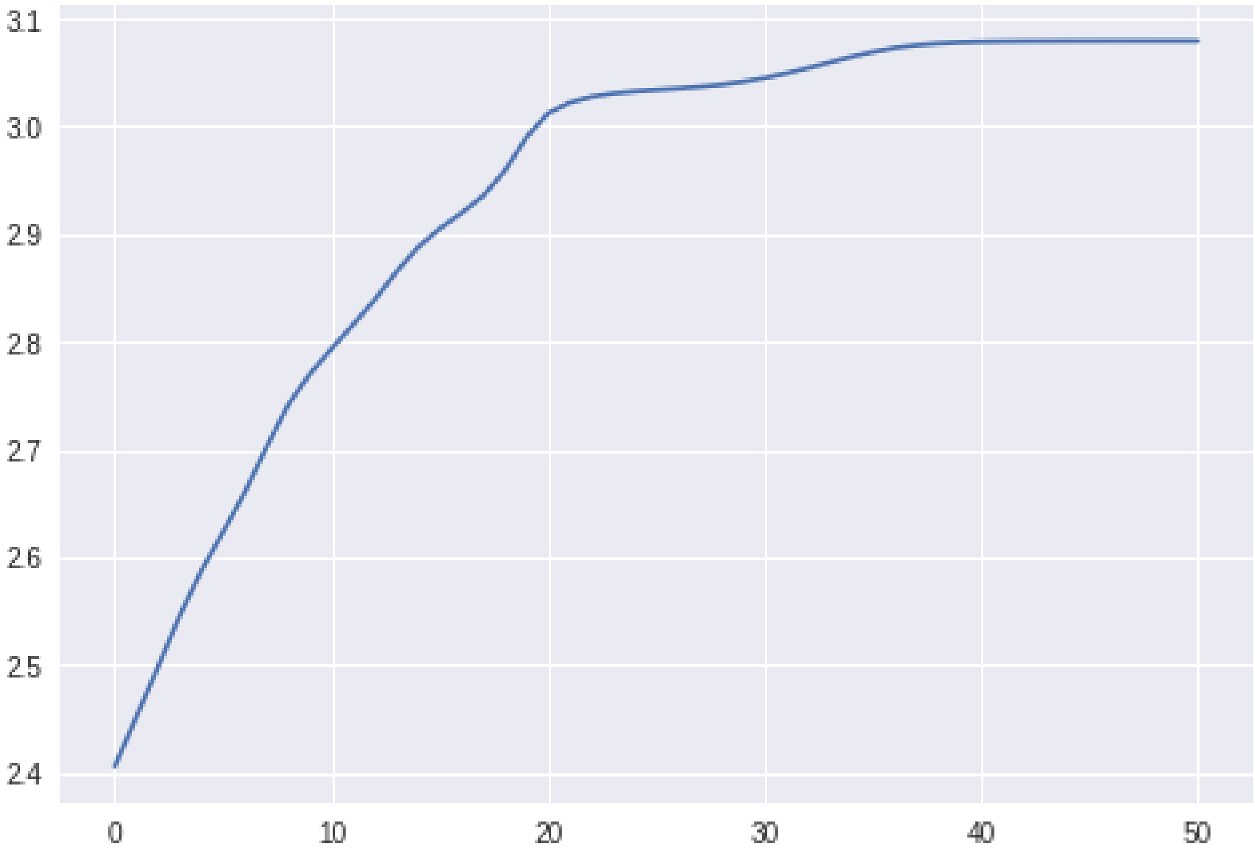
   (c) **Maximization of Covariances**: Complete the function $[\Sigma] = \text{MaximizeCovariance}(X, k, w, \mu)$. This function takes in the training data along with membership weights and means for each gaussian, and calculates the new maximum likelihood covariance for each gaussian

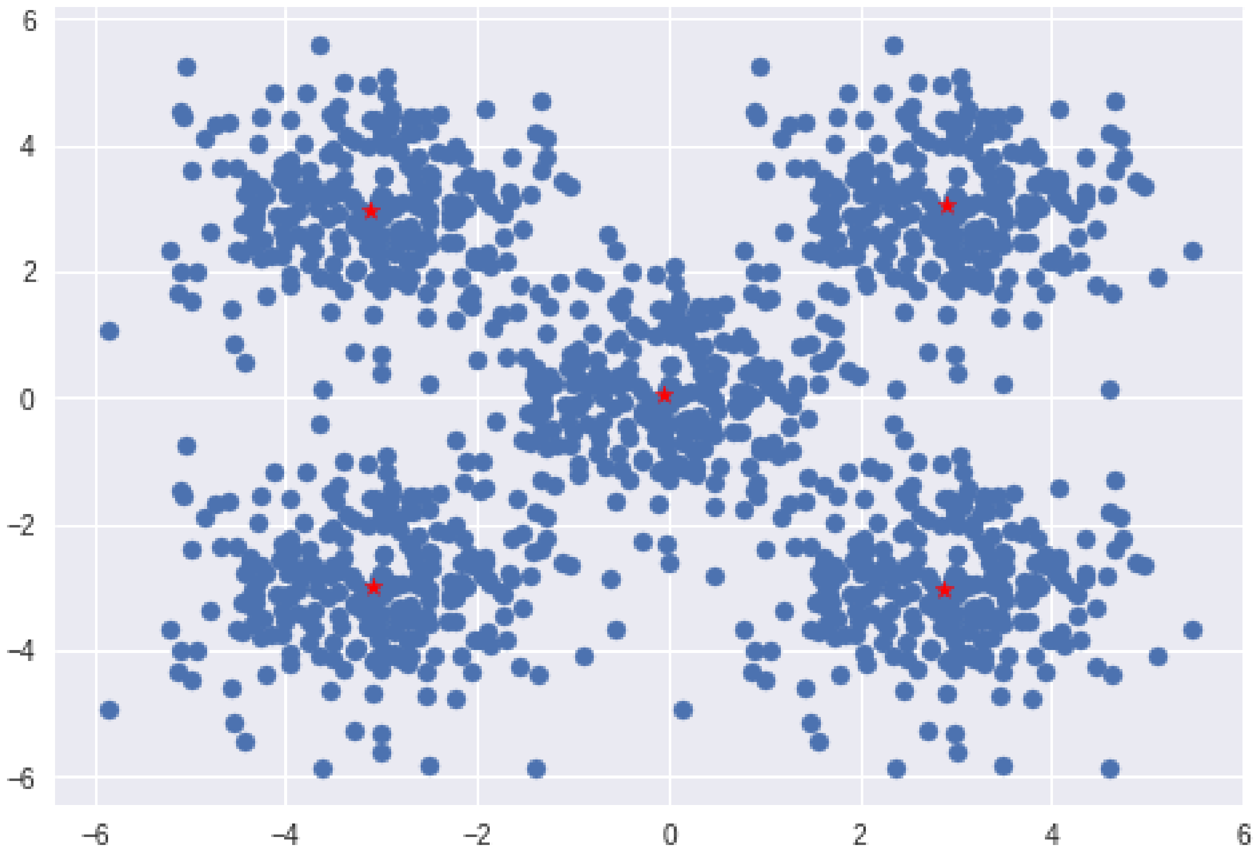
   (d) **Maximization of Mixture Weights** : Complete the function $[\pi] = \text{MaximizeMixtures}(k, w)$. This function takes in the membership weights, and calculates the new maximum likelihood mixture weight for each gaussian.

   (e) **EM**: Put everything together and implement the function $[\pi, \mu, \Sigma] = \text{EM}(X, k, \pi_0, \mu_0, \Sigma_0, \text{ nIter})$. This function runs the EM algorithm for nIter steps and returns the parameters of the underlying GMM. Note: Since this code will call your other functions, make sure that they are correct first. A good way to test your EM function offline is to check that the log likelihood, $\log P(X|\pi, \mu, \Sigma)$ is increasing for each iteration of EM.

The expectation minimization algorithm is implemented from scratch and we set the iterations are set at 50 based on the log likelihood graph below.

The best parameters are listed below:
'covars$'_1 : array([[8.86203613e-01, 4.27366400e-04], [4.27366400e-04, 8.78356071e-01]])$,
$'covars'_2 : array([[0.91342991, -0.0084315], [-0.0084315, 0.93509137]])$,
$'covars'_3 : array([[0.92230994, 0.02279331], [0.02279331, 0.99115499]])$,
$'covars'_4 : array([[0.84372863, -0.0651485], [-0.0651485, 0.87830482]])$,
$'covars'_5 : array([[0.96301923, 0.04237815], [0.04237815, 1.04382033]])$,
$'means'_1 : array([-0.04123075, 0.06042953])$,
$'means'_2 : array([-3.12818809, 2.98890143])$,
$'means'_3 : array([2.86602887, -3.01542802])$,
$'means'_4 : array([2.90588945, 3.04066146])$,
$'means'_5 : array([-3.0931381, -2.97107322])$,
$'mixing_coeff_pi'_1 : 0.19874285694843885,' mixing_coeff_pi'_2 : 0.20045454794034887$,
$'mixing_coeff_pi'_3 : 0.2006363977465461,' mixing_coeff_pi'_4 : 0.19372856962256896$,
$'mixing_coeff_pi'_5 : 0.20643762774209704$


3.1
3.0
2.9
2.8
2.7
2.6
2.5
2.4
0
10
20
30
40
50


6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6