CS 529: Homework 3

Andrea Zapata - PUID : 0031827996

Problem 1

1. The training error for nearest neighbors when k=1 is zero because each input would chose itself as the nearest neighbor as all training samples are kept for testing, then the model would not make any mistakes.

2. The decision surface is different, K-means clustering will produce clusters with circular shapes where the radius is the furthest point to the center of the cluster. Gaussian Mixture Models (GMM) Clustering is able to find more complex decision surfaces (Ex: elliptical clusters) as it would model the normal probability distribution of each region.

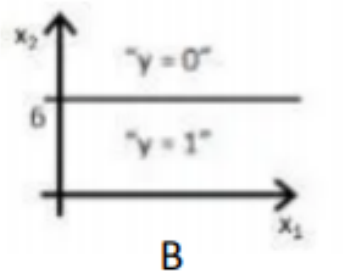
3. Euclidean distance will add the individual distances for each feature. If some features are correlated, then they might be adding more weight to the distance. Mahalanobis distance will measure how many standard deviations each of the points is from the mean for each feature and add these values. As this distance takes into consideration the covariance matrix, if two features are correlated, then the distance will be scaled appropriately.

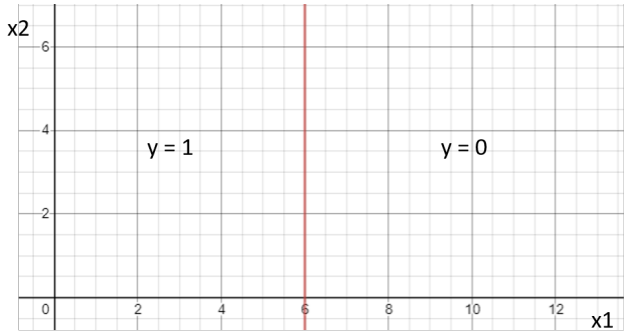
4. It is true that the objective function optimized by the EM algorithm can also be optimized by a gradient descent algorithm, but the whole statement is FALSE. EM is sometimes the preferred over Gradient Descent, because taking the derivatives of some functions might be too complex, but it is not always true that EM finds its solution more quickly. Also, there area cases when gradient descent algorithm will not find the global optimal solution (not convex functions), but a locally optimal solution.

5. The main assumption of Naive Bayes is that features are independent to each other, so that $P(X, Y, Z|C_1) = P(X|C_1)P(Y|C_1)P(Z|C_1)$, then we can express $P(C_1|X, Y, Z) \propto P(X|C_1)P(Y|C_1)P(Z|C_1)P(C_1)$

6. Our decision function will be $y = \begin{cases} 1, & x_2 \leq 6 \\ 0, & x_2 > 6 \end{cases}$, then the corresponding plot would be B.

   If we change the coefficients for $x1, x_2$, then our decision function will be $y = \begin{cases} 1, & x_1 \leq 6 \\ 0, & x_1 > 6 \end{cases}$





Problem 2

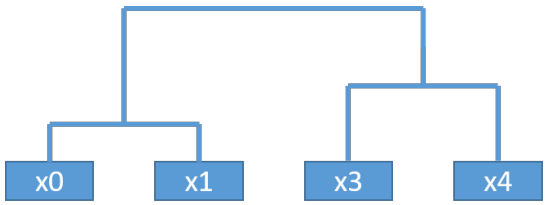
$$x_0 = (1, 1) \qquad x_1 = (2, 1) \qquad x_2 = (4, 3) \qquad x_3 = (5, 4)$$

Let's find the distance between each pair of points, I will only show the upper diagonal matrix, as the distance matrix is symmetric. We will form a cluster with (x_0, x_1) and recalculate our distances. Then we will form a cluster with (x_2, x_3).

	x1	x2	x3
x0	1	5	7
x1		4	6
x2			2

	x2	x3
x0, x1	4	6
x2		2

Then for our two clusters the centers will be $c_1 = (1.5, 1)$ and $c_2 = (4.5, 3.5)$. The dendogram of the clustering is the following:



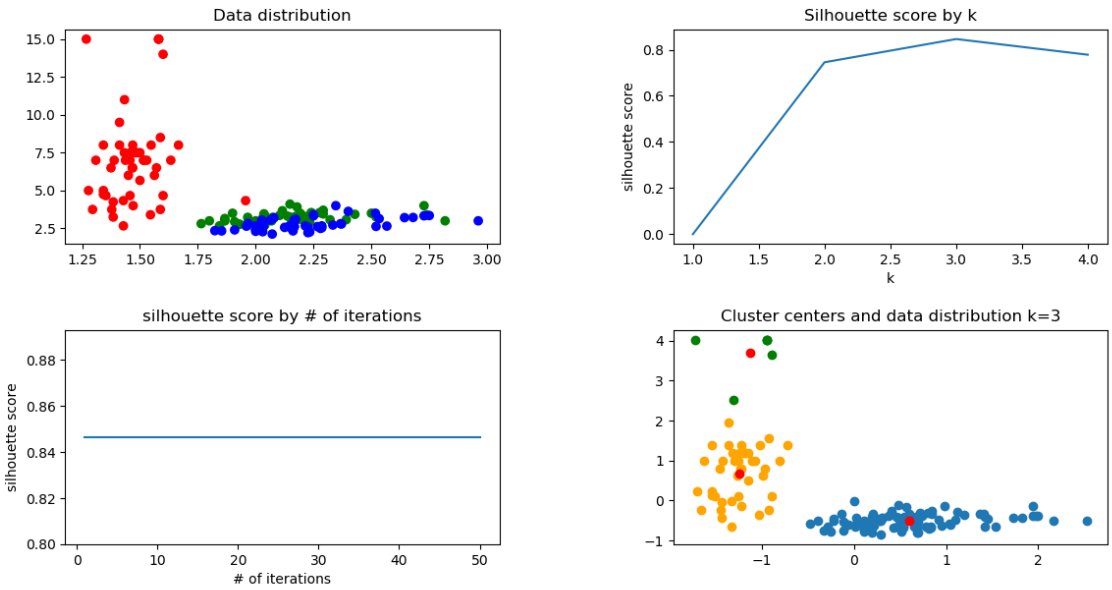
Problem 3

For this problem, I first normalized the data to ensure that all features are given the same importance for calculating the distance. Then for implementing kmeans++, I selected a random element from X, using the random seed 23 and then selected the two farthest possible cluster centers. For assigning each element to a cluster, I find the squared euclidean distance to each of the centers, and then use a helper function map_cluster to create the binary dictionary. Finally, once I have the data_map dictionary , I multiply the data map column corresponding to each center, to get the points that belong on that cluster. With these points, I can find the new centers for each cluster using the mean and repeat the process until the number of iterations is satisfied. For measuring the accuracy of a found cluster, I used Silhouette Index. This index compares the intra-cluster similarity with the inter-cluster similarity. According to the definition, for each element of a cluster I found the Silhouette Index with:

$$S(i) = \frac{b(i) - a(i)}{max\,[a(i), b(i)]}$$

Where, a(i) is the average dissimilarity of ith object to all other objects in the same cluster and b(i) is the average dissimilarity of ith object with all objects in the closest cluster. Then, I took the mean from all elements of a cluster to have the Silhouette Index of each cluster, and then took the mean of all these values to find the accuracy for a given k. In the plot, we can see that k=3 gives us the best results, then that was the selected value to the next steps.

For k = 3, I observed that my algorithm found the optimal cluster centers on the second iteration, then the number of iterations will not improve the accuracy, as all of them converge on the same cluster centers. In the plot we can see, the selected cluster centers and the clusters for our input. We can conclude that even when our silhouette score was high, the selected k will not help us to classify the dataset. We could have noticed that from the beginning because our real clusters do not follow a spherical shape, then knn would not be able find a decision surface at least for the green and blue cluster.



Problem 4

From definition we got :

$$P(x, w_1) \sim N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} a & c \\ c & b \end{bmatrix}\right) \qquad |\Sigma_1| = 1 \qquad \Sigma_1^{-1} = \begin{bmatrix} b & -c \\ -c & a \end{bmatrix} \qquad P(w_1) = \frac{1}{2}$$

$$P(x, w_2) \sim N_2\left(\begin{bmatrix} d \\ e \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \qquad |\Sigma_2| = 1 \qquad \Sigma_2^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad P(w_2) = \frac{1}{2}$$

Then $g1$ is :

$$\begin{aligned} g_1(x) &= \ln \frac{1}{2\pi|\Sigma_1|^{1/2}} - \frac{1}{2}(\mathbf{x} - \mu_\mathbf{1})^T \mathbf{\Sigma_1}(\mathbf{x} - \mu_\mathbf{1}) \\ &= -\ln 2\pi - \frac{1}{2}\ln|\Sigma_1| - \ln 2 - \frac{1}{2}(\mathbf{x} - \mu_\mathbf{1})^T \mathbf{\Sigma_1}(\mathbf{x} - \mu_\mathbf{1}) \end{aligned}$$

As we know μ_1 is a zero-matrix then $(x - \mu_1) = x$, so we got :

$$g_1(x) = -\ln 2\pi - \ln 2 - \frac{1}{2}\mathbf{x}^T\mathbf{\Sigma_1}\mathbf{x}$$

Then, we can multiply all the matrices to get a scalar value :

$$\begin{bmatrix} \text{x}_1 & \text{x}_2 \end{bmatrix} \begin{bmatrix} \text{b} & \text{-c} \\ \text{-c} & \text{a} \end{bmatrix} \begin{bmatrix} \text{x}_1 \\ \text{x}_2 \end{bmatrix} = \begin{bmatrix} \text{b x}_1 - cx_2 & \text{- c x}_1 + ax_2 \end{bmatrix} \begin{bmatrix} \text{x}_1 \\ \text{x}_2 \end{bmatrix} = {bx_1}^2 - cx_2x_1 - cx_2x_1 + a{x_2}^2$$
$$= {bx_1}^2 - 2cx_2x_1 + a{x_2}^2$$

Then, if we replace that value in g_1, we get

$$g_1(x) = -\ln 2\pi - \ln 2 - \frac{1}{2}({bx_1}^2 - 2cx_2x_1 + a{x_2}^2)$$

Then, for $g2$, we got :

$$g_2(x) = \ln \frac{1}{2\pi|\Sigma_2|^{1/2}} - \frac{1}{2}(\mathbf{x} - \mathbf{\mu_2})^T\mathbf{\Sigma_2}(\mathbf{x} - \mathbf{\mu_2})$$
$$= -\ln 2\pi - \frac{1}{2}\ln|\Sigma_2| - \ln 2 - \frac{1}{2}(\mathbf{x} - \mathbf{\mu_2})^T\mathbf{\Sigma_2}(\mathbf{x} - \mathbf{\mu_2})$$

As Σ_2 is an identity matrix, then its inverse is also an identity matrix, so $(\mathbf{x} - \mathbf{\mu_2})^T\mathbf{\Sigma_2}(\mathbf{x} - \mathbf{\mu_2}) = (\mathbf{x} - \mathbf{\mu_2})^T(\mathbf{x} - \mathbf{\mu_2})$

$$g_2(x) = -\ln 2\pi - \ln 2 - \frac{1}{2}(\mathbf{x} - \mathbf{\mu_2})^T(\mathbf{x} - \mathbf{\mu_2})$$

Then, we can multiply all the matrices to get a scalar value :

$$\begin{bmatrix} \text{x}_1 - d & \text{x}_2 - e \end{bmatrix} \begin{bmatrix} \text{x}_1 - d \\ \text{x}_2 - e \end{bmatrix} = (x_1 - d)^2 + (x_2 - e)^2$$

Then, if we replace that value in g_2, we get:

$$g_2(x) = -\ln 2\pi - \ln 2 - \frac{1}{2}((x_1 - d)^2 + (x_2 - e))$$

If we set $g_1(x) = g_2(x)$, then :

$${bx_1}^2 - 2cx_2x_1 + a{x_2}^2 = (x_1 - d)^2 + (x_2 - e)^2$$
$${bx_1}^2 - 2cx_2x_1 + a{x_2}^2 = x_1^2 - 2dx_1 + d^2 + x_2^2 - 2ex_2 + e^2$$
$$x_1^2(b-1) + x_2^2(a-1) - 2cx_1x_2 + 2dx_1 + 2ex_2 - d^2 - e^2 = 0$$

$$choose \begin{cases} w_1, & x_1^2(b-1) + x_2^2(a-1) - 2cx_1x_2 + 2dx_1 + 2ex_2 - d^2 - e^2 \leq 0 \\ w_2, & \text{otherwise} \end{cases}$$

If we set $a = 1$, $b = 1$, $c = 0$, this values also fit our constraint $ab - c^2 = 1$, then our function would be linear for any value of d and e:

$$2dx_1 + 2ex_2 - d^2 - e^2 = 0$$
$$x_1 = \frac{d^2 + e^2 - 2ex_2}{2d}$$

Then if we choose $d = 1$, $e = 1$, we get

$$x_1 = \frac{2 - 2x_2}{2} = 1 - x_2$$

$$choose \begin{cases} w_1, & x_1 \leq 1 - x_2 \\ w_2, & \text{otherwise} \end{cases}$$