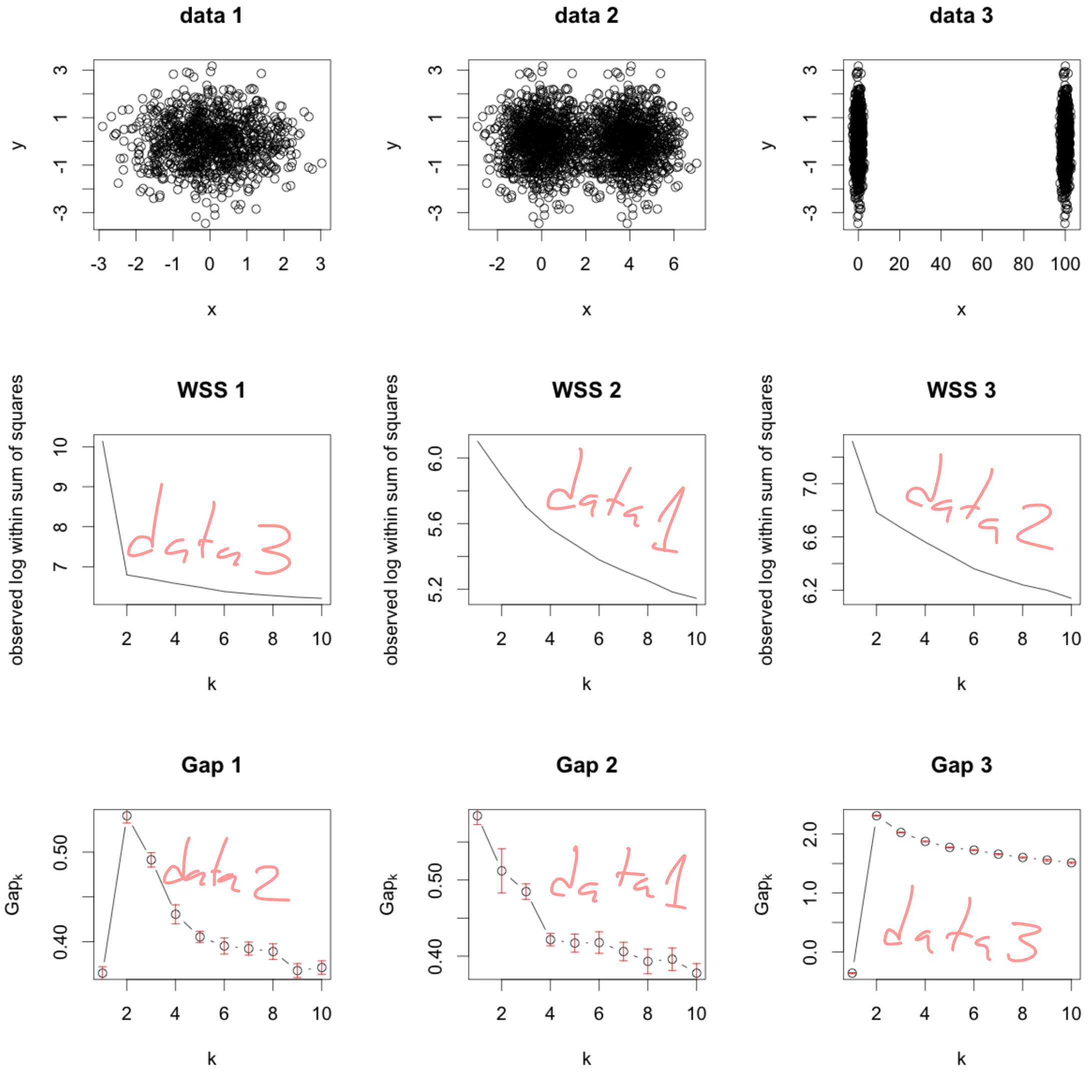
Problem 1

Below are three data sets and they are corresponding plots of: (1) the data, (2) the log total within the sum of squares, and (3) the Gap statistic.

Match the data with the corresponding entry in the second row and the entry in the third row. Explain your logic for the matching carefully.



Explanation:

data 1 → WSS 2 → Gap 2

↑ does not have defined elbow

↑ has a peak at 1

data 2 → WSS3 → Gap 1

↑ has elbow @ 2 but not very sharp

↑ because data2 has 2 clusters that are close as k increases there there is still information gain

data3 → WSS 4 → Gap3

↑ sharp elbow

↑ diminishing returns of information after k = 2

Problem 2

The standard k-means loss function for a fixed value of K is:

$$L_{1K}(\boldsymbol{\mu}, \boldsymbol{\alpha}) = \sum_{k=1}^{K} \sum_{i=1}^{n} \sum_{p=1}^{P} (x_{ip} - \mu_{kp})^2 \mathbb{1}\{\alpha_i = k\}$$

where

- x_{ip} is p-th dimension of the i-th data point,
- n is the total number of data points,
- μ_{kp} is the center of cluster k in dimension p,
- $\mathbb{1}\{\cdot\}$ is an indicator function. That is, it is equal to one if the test is true, zero otherwise,
- α_i is the label of the i-th data point.

Now, let's say we change the loss function to:

$$L_{2K}(\boldsymbol{\mu}, \boldsymbol{\alpha}) = \sum_{k=1}^{K} \sum_{i=1}^{n} \sum_{p=1}^{P} \left((x_{ip} - \mu_{kp})^2 + \log(K) \right) \mathbb{1}\{\alpha_i = k\},$$

(a) Given a fixed K and fixed μ, would the two loss functions result in different assignments? Explain your logic.

(b) Given a fixed assignment, what is the optimal cluster center update? Make sure to justify and check that it is indeed an optimal update.

(c) How does the total within sum of squares behave as a function of K? A plot of both loss functions could help, but is not required.

(d) As the L_{2K} is written, how does it behave differently for different numbers of dimensions? Stated differently: given random data in dimension P, if you increase P, how does the plot of the within sum of squares change?

2a) No! if K is fixed then we would just have a constant difference of log(K) for each evaluation of the loss function. This constant will shift every evaluation equally so we would se no change in assignments

2b)

$$L_{2K}(\boldsymbol{\mu}, \boldsymbol{\alpha}) = \sum_{k=1}^{K} \sum_{i=1}^{n} \sum_{p=1}^{P} \left((x_{ip} - \mu_{kp})^2 + \log(K)\right) \mathbb{1}\{\alpha_i = k\},$$

$$\frac{\partial}{\partial \mu} L_{2K}(\mu, \alpha) = -\sum_i 2(X_{ip} - \mu_{kp}) = 0$$

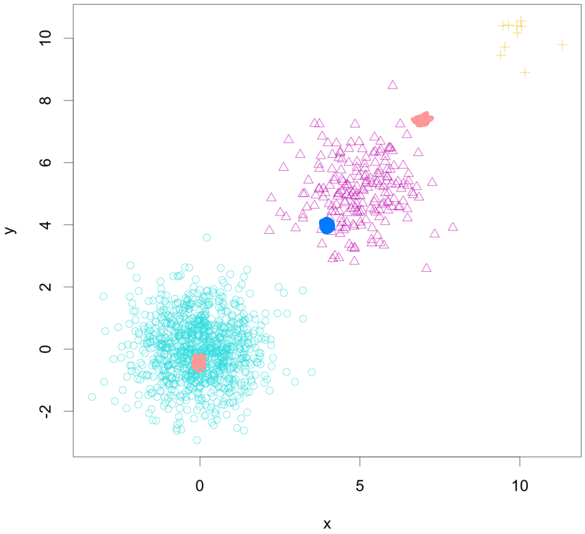
$$\mu_{kp} = \frac{1}{|i|} \sum_i X_{ip}$$

2c) The total within sum of squares is a function of K because it will increase by a factor of $P \cdot \log(K)$. This occurs because it is summed P times.

2d) The plot will increase by $(\Delta P) \log(K)$ over the course of the plot.

Problem 3

Consider the following data:



It is generated by random Gaussian noise centered around (0, 0), (5, 5), and (10, 10). In the first cluster there are 1,000 points. In the second cluster there are 200 points. In the last cluster there are 10 points.

(a) If I were to run k-means with $K = 1$, where would the center be? No need to be precise, but justify.

(b) If I were to run k-means with $K = 2$, where would the cluster centers be? No need to be precise again, just justify.

(c) If I were to run k-means with $K = 3$, I would likely get unreliable behavior with standard k-means. Can you give two examples of the results you might see and justify?

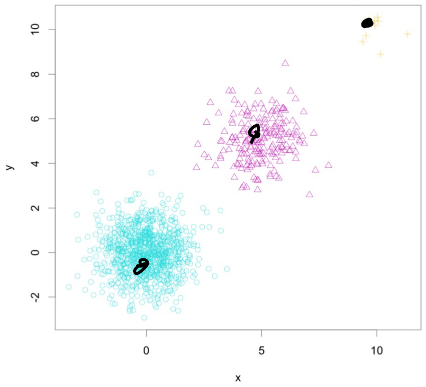
(d) What are the factors about the algorithm and the data resulting in this behavior? Looking at the data (because you can in two-dimensions), is there any way to guarantee k-means will give the behavior you might expect to with $K = 3$?

(e) Would the problems in (c) be fixed if you increased K moderately, say, up to 5?

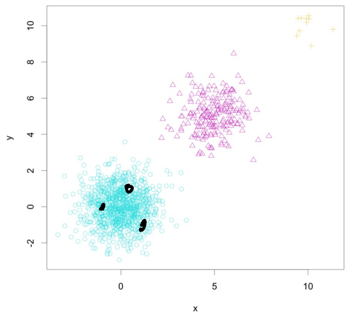
3a) around 4,4 because it would just get the center of all points

3b) One in the center of blue (0,0) ish and one at the top right of purple because the yellow would get grouped with the purple.

3c) The behavior would be unreliable because depending on the start position the clusters might form differently.



If this was the start then it would converge to correctly identifying the 3 clusters



If this was the start then it might not be able to converse the same way as above

3d) the only way to guarantee or attempt to guarantee is that you run $k=3$ kmeans for many times then soft classify each point into a cluster then at the end you classify each point based on the highest percent relation to each cluster.

3e) No we would still have this problem because there would be more room for missclassification

Problem 4

Here is some data:

i	x_{i1}	x_{i2}
1	1	1
2	3	2
3	3	3

and the following centers:

k	μ_{k1}	μ_{k2}
1	2	2
2	5	5

We define the following distance function:

$$d(x, y) = |x_1 - y_1| + |x_2 - y_2|$$

To keep things simple, let's say:

$$HiddenMatrix_{i,k} = \frac{\frac{1}{d(x_i,\mu_k)}}{Z_i}$$

where Z_i is a normalization factor that you must compute.

(a) Compute the "E"-step. Scare quotes are there to make Rose happy. The numbers mostly play nice if you keep them in fractions, so I might do that.

(b) Compute the "M"-step. The numbers don't play as nicely here so might be easier to just use a calculator/python/R, etc.

4a)

i	x_{i1}	x_{i2}
1	1	1
2	3	2
3	3	3

$$d(x,y) = |x_1 - y_1| + |x_2 - y_2|$$

$$HiddenMatrix_{i,k} = \frac{\frac{1}{d(x_i,\mu_k)}}{Z_i}$$

k	μ_{k1}	μ_{k2}
1	2	2
2	5	5

point to cluster

$i=1$

$k=1 \quad \frac{1}{|1-2|+|1-2|} = \frac{1}{2}$ $\qquad \frac{1}{2}+\frac{1}{8} = 5/8$

$k=2 \quad \frac{1}{8}$

$i=2$

$k=1 \quad 1$ $\qquad 1+\frac{1}{5} = \frac{6}{5}$

$k=2 \quad 1/5$

$i=3$

$k=1 \quad 1/2$ $\qquad 1/2 + 1/4 = 3/4$

$k=2 \quad 1/4$

hidden matrix =

$$\begin{array}{cc|ccc} & & & i & \\ & & 1 & 2 & 3 \\ \hline k & 1 & 4/5 & 5/6 & 2/3 \\ & 2 & 1/5 & 1/6 & 1/3 \end{array}$$

4b)

$$\frac{(4/5 \cdot 1) + (5/6 \cdot 3) + (2/3 \cdot 3)}{4/5 + 5/6 + 2/3 = 2.3} = \mu_{11} = 2.30$$

$$\frac{\left(\frac{4}{5} \cdot 1\right) + \left(\frac{5}{6} \cdot 2\right) + \left(\frac{2}{3} \cdot 3\right)}{2.3} = \mu_{21} = 1.94$$

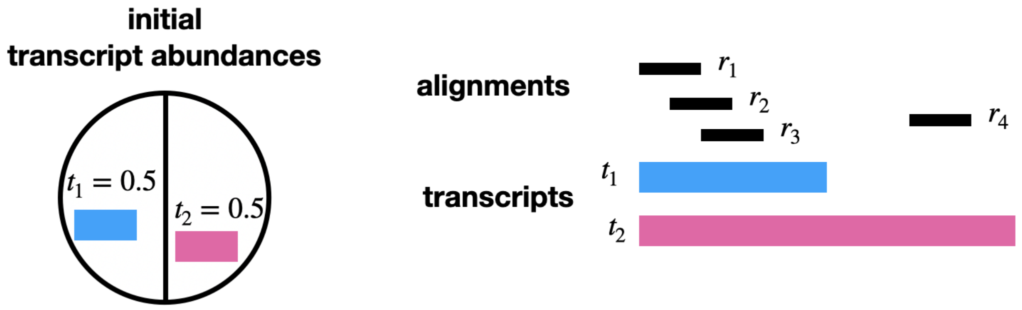
$$\frac{\left(\frac{1}{5} \cdot 1\right) + \left(\frac{1}{6} \cdot 3\right) + \left(\frac{1}{3} \cdot 3\right)}{1/5 + 1/6 + 1/3 = 0.7} = \mu_{21} = 2.43$$

$$\frac{\left(\frac{1}{5} \cdot 1\right) + \left(\frac{1}{6} \cdot 2\right) + \left(\frac{1}{3} \cdot 3\right)}{0.7} = \mu_{22} = 2.19$$

Problem 5

[extra credit. no questions, please.]

Consider the following toy RNA-seq example:



We initialize the (relative) abundances to be equal since we don't know anything. You can assume $\tilde{l}_1 = l$, $\tilde{l}_2 = 2 \cdot l$. Additionally, assume that the fragment lengths are the same and the fragment length distribution is a point mass.

(a) Compute the E-step.

(b) Compute the M-step.

(c) Why did the EM behave the way it did given this data?

Problem 6

[1 point]

Don't be so serious. Draw something silly.