# *Homework 3 Answers*
# *Kelly Scott Sims*

## *Q2.3b*

| **Percentage Cluster** | **Case** | **Control** | **Unknown** | **Total** |
|---|---|---|---|---|
| **Cluster 1** | 85 **(8.701 %)** | 468 **(49.367 %)** | 271 **(15.362 %)** | **824** |
| **Cluster 2** | 794 **(81.352 %)** | 410 **(43.2489 %)** | 1282 **(72.675 %)** | **2486** |
| **Cluster 3** | 97 **(9.938%)** | 70 **(7.384 %)** | 211 **(11.9614 %)** | **378** |
|  | **100%** | **100%** | **100%** | **3688** |

Table 1: Clustering with 3 centers using all features

| **Percentage Cluster** | **Case** | **Control** | **Unknown** | **Total** |
|---|---|---|---|---|
| **Cluster 1** | 936 **(95.901 %)** | 0 **(0 %)** | 280 **(28.717 %)** | **1216** |
| **Cluster 2** | 21 **(2.151%)** | 938 **(100 %)** | 689 **(70.666%)** | **1648** |
| **Cluster 3** | 19 **(1.9467%)** | 0 **(0 %)** | 6 **(0.6154 %)** | **25** |
|  | **100%** | **100%** | **100%** | **2889** |

Table 2: Clustering with 3 centers using filtered features

## *Q2.4b*

| Percentage Cluster | Case | Control | Unknown | Total |
|---|---|---|---|---|
| **Cluster 1** | 90 **(9.2213%)** | 468 **(49.367 %)** | 430 **(24.3764 %)** | **988** |
| **Cluster 2** | 146 **(14.959 %)** | 371 **(39.135 %)** | 510 **(28.9115 %)** | **1027** |
| **Cluster 3** | 740 **(75.8196 %)** | 109 **(11.49789 %)** | 824 **(46.712%)** | **1673** |
| | **100%** | **100%** | **100%** | **3688** |

Table 3: GMM with 3 centers using all features

| Percentage Cluster | Case | Control | Unknown | Total |
|---|---|---|---|---|
| **Cluster 1** | 594 **(60.8606%)** | 938 **(100 %)** | 135 **(13.8461%)** | **1667** |
| **Cluster 2** | 145 **(14.8565 %)** | 0 **(0 %)** | 770 **(78.97435 %)** | **915** |
| **Cluster 3** | 237 **(24.2827%)** | 0 **(0 %)** | 70 **(7.17948%)** | **307** |
| | **100%** | **100%** | **100%** | **2889** |

Table 4: GMM with 3 centers using filtered features

## *Q2.5a*

As noted in the Spark Documentation "Streaming k-means" retrieved from https://spark.apache.org/docs/latest/mllib-clustering.html#streaming-k-means, the streaming k-means update rule is a generalization of the mini batch k-means update rule.

**1.)** $C_{t+1} = \frac{C_t n_t \alpha + x_t m_t}{n_t \alpha + m_t}$

**2.)** $n_{(t+1)} = n_t + m_t$

Where $C_t$ is the previous center for the cluster, $n_t$ is the number of points assigned to the cluster so far, $x_t$ is new cluster center and mt is number of new points. Alpha is the "forgetfulness" factor and is in range of [0,1]. When alpha is 1, all previous data is factored into the weight. As alpha approaches 0, the newer data is weighted heavier.

Streaming k-means works in the same fashion that SGD works. Data arrives in batches with an arbitrary amount in each batch. Each observation in the batch is assigned to a cluster and the centroids locations are updated by computing the average of points within the cluster. This process is repeated over and over. If alpha is relatively low, the centroids will follow more closely with the positions of the incoming data. This is beneficial in instances that the patterns in the data is dynamic and changes greatly over time. This allows the centroid to converge quickly with respect to the new data. If alpha is high, the previous centroid location will bear more weight in the placement of the updated location.

The pros of this approach is that an entire dataset doesn't have to be loaded in memory to perform the updates. As seen in equation (1) above, in order to preserve information about the previous centroid location, instead of needing the entire dataset, only the number of points in the centroid and previous centroid location is needed. This can potentially reduce the computation time. Also, as mentioned earlier, this approach allows the model to realign itself with dynamic data patterns.

The cons of this approach is that it is allowing a trade off in reducing computational time for a reduction in overall cluster quality.

## *Q2.5c*

| Percentage Cluster | Case | Control | Unknown | Total |
|---|---|---|---|---|
| **Cluster 1** | 202 **(20.6967%)** | 476 **(50.2109 %)** | 375 **(21.258 %)** | **1053** |
| **Cluster 2** | 705 **(72.233 %)** | 467 **(49.261 %)** | 949 **(53.798 %)** | **2121** |
| **Cluster 3** | 69 **(7.069 %)** | 5 **(0.527 %)** | 440 **(24.9433%)** | **514** |
| | **100%** | **100%** | **100%** | **3688** |

Table 5: Streaming k-means with 3 centers using all features

| Percentage Cluster | Case | Control | Unknown | Total |
|---|---|---|---|---|
| **Cluster 1** | 14 **(1.434%)** | 0 **(0 %)** | 21 **(2.1538 %)** | **35** |
| **Cluster 2** | 959 **(98.258 %)** | 938 **(100 %)** | 337 **(34.564 %)** | **2234** |
| **Cluster 3** | 3 **(0.307 %)** | 0 **(0 %)** | 617 **(63.282%)** | **620** |
| | **100%** | **100%** | **100%** | **2889** |

Table 6: Streaming k-means with 3 centers using filtered features

## *Q2.6a*

*The first thing to note with the k-means clustering algorithm with all features is that the majority of the data points have mostly ended up in Cluster 2. This, in a first guess analysis, would lead one to believe that the algorithm may need more iterations to drive convergence. This is compounded with the fact that only ~10% of the data resides in Cluster 3. There is also a probability of the model suffering from too much noise from all features.*

*When the model is trained on filtered data, there is a little better grouping between the classes. Cluster 1 is predominantly predicting Case patients and Cluster 2 is comprised of the other two classes mostly. This is an improvement. Yet it still appears that maybe convergence isn't being achieved, as the amount of points in Cluster 3 has dropped off to 25 observation from 378.*

*When examining GMM, the same analysis applies. Either convergence or too much noise is present in the data to create clear seperators between the clusters. When the filtered features are applied, there's a better separation between the groups between clusters.*

*If stopping criteria is being achieved within the 20 iterations, then a better initial centroid initialization algorithm might be worth trying.*

## *Q2.6b*

| K | K-Means All features | K-Means Filtered features | GMM All features | GMM Filtered features |
|---|---|---|---|---|
| **2** | **0.4783** | **0.6549** | **0.47831** | **0.52440** |
| **5** | **0.4731** | **0.87020** | **0.5235** | **0.76393** |
| **10** | **070309** | **0.8993** | **0.5437** | **0.8985** |
| **15** | **0.6613** | **0.89235** | **0.62852** | **0.89927** |

Table 5: Purity values for different number of clusters

When number of clusters are increased, purity increases. This is to be expected as with increasing clusters, each point will more or less start become its own cluster with significant amount of initialized centroids. This consequently removes any insightful/meaningful pattern potentially seen in the data.