2.3 K-Means Clustering

Clustering with 3 centers using all features (purity = 0.47831)			
Percentage Cluster	Case	Control	Unknown
Cluster 1	9.94%	7.38%	11.96%
Cluster 2	90.06%	92.09%	84.69%
Cluster 3	0%	0.53%	3.34%
	100%	100%	100%

Clustering with 3 centers using filtered features (purity = 0.66126)			
Percentage Cluster	Case	Control	Unknown
Cluster 1	99.28%	0.00%	29.64%
Cluster 2	0.72%	97.78%	70.26%
Cluster 3	0.00%	2.22%	0.10%
	100%	100%	100%

2.4 Clustering with Gaussian Mixture Model

Clustering with 3 centers using all features (purity = 0.47831)			
Percentage Cluster	Case	Control	Unknown
Cluster 1	0.10%	0.84%	1.93%
Cluster 2	3.28%	3.80%	8.11%
Cluster 3	96.62%	95.36%	89.97%
	100%	100%	100%

Clustering with 3 centers using filtered features (purity = 0.68161)			
Percentage Cluster	Case	Control	Unknown
Cluster 1	77.36%	0.00%	26.97%
Cluster 2	3.28%	97.78%	42.88%
Cluster 3	19.36%	2.22%	30.15%
	100%	100%	100%

2.5 Clustering with Streaming K-Means

a. In streaming K-means, data arrive in batches. For every batch of data, we firstly allocate each to its nearest center, and then calculate the new centers and update the clusters.

1) Update rule for every cluster:

$$C_{t+1} = \frac{\alpha C_t N_t + B_t N_{Bt}}{\alpha N_t + N_{Bt}}$$

$$N_{t+1} = N_t + N_{Bt}$$

C_t : the last cluster center,

N_t: the number of points in the cluster

B_t: center of the current batch

N_{Bt}: the number of points in the batch

α: weight constant

2) How it works:

(1) Allocate new data points to its nearest cluster

(2) Update cluster weight by time unit

(3) Update each cluster

(4) If a cluster disappears, then divide the largest cluster to two new clusters.

3) Pros and Cons:

Pros: Able to use live streaming data, can reflect the changes of data source over time without depending on previous data. For example: analyzing real-time data.

Cons: Due to the uncertainty about incoming data, the algorithm has higher chances encountering error than regular K-Means.

4) Forgetfulness:

Forgetfulness allows us to assigns different weight for the data batches coming in different times. For example, If the data sources change over time, it's better to let the recent data batches have more importance. Or if the data sources do not change, then every batch of data can have the same weight, in the end the result would be similar to a K-Means clustering.

c.

Clustering with 3 centers using all features (purity = 0.50331)			
Percentage Cluster	Case	Control	Unknown
Cluster 1	13.93%	43.78%	17.74%
Cluster 2	70.70%	2.11%	50.11%
Cluster 3	15.37%	54.11%	32.14%
	100%	100%	100%

Clustering with 3 centers using filtered features (purity = 0.86512)			
Percentage Cluster	Case	Control	Unknown
Cluster 1	91.39%	0.00%	29.85%
Cluster 2	2.15%	100.00%	1.64%
Cluster 3	6.45%	0.00%	68.51%
	100%	100%	100%

2.6 Discussion on K-means and GMM

a. I found that for all-features, case, control and unknown patients all seem to focus on one cluster for K-Means and GMM. But for filtered-features, the purity has a huge improvement, and the clusters seem to work well based on the distribution except for unknown patients.

b.

k	K-Means All features	K-Means Filtered features	GMM All Features	GMM Filtered features
2	0.52494	0.66126	0.47831	0.43532
5	0.61008	0.40531	0.47831	0.84201
10	0.60602	0.41187	0.66269	0.87892
15	0.71366	0.89169	0.65998	0.89306

For all features, as the K increases, both purity of K-Means and GMM increase. For filtered features, GMM purity increases as K increases, but K-Means doesn't show a particular pattern.

3. Advanced phenotyping with NMF

b.

k	NMF All features	NMF Filtered features
2	0.4783	0.52061
3	0.5504	0.65022
4	0.4783	0.63035
5	0.4783	0.62461

c.

Clustering with 3 centers using all features			
Percentage Cluster	Case	Control	Unknown
Cluster 1	45.59%	47.78%	54.37%
Cluster 2	50.92%	20.36%	44.56%
Cluster 3	3.48%	31.86%	1.08%
	100%	100%	100%

Clustering with 3 centers using filtered features			
Percentage Cluster	Case	Control	Unknown
Cluster 1	68.43%	6.20%	23.74%
Cluster 2	5.26%	80.78%	38.65%
Cluster 3	26.31%	10.80%	37.61%
	100%	100%	100%

d. Show why we can use MU update rule by deriving the equation for it. [10 points bonus]

Given a feature matrix $\boldsymbol{V}$, the objective of NMF is to minimize the Euclidean distance between the original non-negative matrix $\boldsymbol{V}$ and its non-negative decomposition $\boldsymbol{W} \times \boldsymbol{H}$ which can be formulated as

$$\underset{\boldsymbol{W}\geq 0, \boldsymbol{H}\geq 0}{\operatorname{argmin}} \quad \frac{1}{2}||\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}||_2^2 \tag{1}$$

where $\boldsymbol{V} \in \mathbb{R}_{\geq 0}^{n\times m}$, $\boldsymbol{W} \in \mathbb{R}_{\geq 0}^{n\times r}$ and $\boldsymbol{H} \in \mathbb{R}_{\geq 0}^{r\times m}$. $\boldsymbol{V}$ can be considered as a dataset comprised of n number of m-dimensional data vectors, and r is generally smaller than n.

We can use gradient descent for minimization of $\boldsymbol{W}$ and $\boldsymbol{H}$. Let Tr be the trace operator.

$$\nabla_W \tfrac{1}{2}||\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}||_2^2 = \tfrac{1}{2}\nabla_W Tr[(\boldsymbol{V}^T - \boldsymbol{H}^T)(\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H})]$$

$$= \tfrac{1}{2}\nabla_W Tr[-\boldsymbol{V}^T\boldsymbol{W}\boldsymbol{H} - \boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{V} + \boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}]$$

$$= -\boldsymbol{V}\boldsymbol{H}^T + \boldsymbol{W}\boldsymbol{H}^T\boldsymbol{H}$$

$$\nabla_H \tfrac{1}{2}||\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}||_2^2 = \tfrac{1}{2}\nabla_H Tr[(\boldsymbol{V}^T - \boldsymbol{H}^T)(\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H})]$$

$$= \tfrac{1}{2}\nabla_H Tr[-\boldsymbol{V}^T\boldsymbol{W}\boldsymbol{H} - \boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{V} + \boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}]$$

$$= -\boldsymbol{W}^T\boldsymbol{V} + \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}$$

We can now update

$$\boldsymbol{W}_{ij}^{t+1} = \boldsymbol{W}_{ij}^t - \alpha_{ij}(-\boldsymbol{V}\boldsymbol{H}^T + \boldsymbol{W}\boldsymbol{H}^T\boldsymbol{H})$$
$$\boldsymbol{H}_{ij}^{t+1} = \boldsymbol{H}_{ij}^t - \beta_{ij}(-\boldsymbol{W}^T\boldsymbol{V} + \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H})$$

Choose α_{ij}, β_{ij} as below:

$$\alpha_{ij} = \frac{\boldsymbol{W}_{ij}^t}{(\boldsymbol{W}^t\boldsymbol{H}\boldsymbol{H}^\top)_{ij}}$$
$$\beta_{ij} = \frac{\boldsymbol{H}_{ij}^t}{(\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{H}^t)_{ij}}$$

Plug α_{ij}, β_{ij} back in, then we have the update rules.

$$\boldsymbol{W}_{ij}^{t+1} = \boldsymbol{W}_{ij}^t \frac{(\boldsymbol{V}\boldsymbol{H}^\top)_{ij}}{(\boldsymbol{W}^t\boldsymbol{H}\boldsymbol{H}^\top)_{ij}}$$
$$\boldsymbol{H}_{ij}^{t+1} = \boldsymbol{H}_{ij}^t \frac{(\boldsymbol{W}^\top\boldsymbol{V})_{ij}}{(\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{H}^t)_{ij}}$$