CSE6250: Big Data Analytics in Healthcare
Homework 3
Richard Albright
Deadline: Feb 28, 2021, 11:55 PM AoE

1

2

2.1

2.2

2.3 K-Means Clustering [8 points]

b. Compare clustering for the $k = 3$ case with the ground truth phenotypes that you computed for the rule-based PheKB algorithms. Specifically, for each of *case*, *control* and *unknown*, report the percentage distribution in the three clusters for the two feature construction strategies. Report the numbers in the format shown in Table ?? and Table ??. [3 points]

Percentage Cluster	Case	Control	Unknown
Cluster 1	76.95%	8.86%	69.90%
Cluster 2	8.71%	47.47%	13.38%
Cluster 3	14.34%	43.67%	16.72%
	100%	**100%**	**100%**

Table 1: Clustering with 3 centers using all features

Percentage Cluster	Case	Control	Unknown
Cluster 1	95.08%	100.00%	32.00%
Cluster 2	4.92%	0.00%	0.62%
Cluster 3	0.00%	0.00%	67.38%
	100%	**100%**	**100%**

Table 2: Clustering with 3 centers using filtered features

2.4 Clustering with Gaussian Mixture Model (GMM) [8 points]

b. Compare clustering for the $k = 3$ case with the ground truth phenotypes that you computed for the rule-based PheKB algorithms. Specifically, for each of *case*, *control* and *unknown*, report the percentage distribution in the three clusters for the two feature construction strategies. Report the numbers in the format shown in Table ?? and Table ??. [3 points]

Percentage Cluster	Case	Control	Unknown
Cluster 1	66.70%	34.28%	46.71%
Cluster 2	5.02%	7.28%	15.82%
Cluster 3	28.28%	58.44%	37.47%
	100%	**100%**	**100%**

Table 3: Clustering with 3 centers using all features

Percentage Cluster	Case	Control	Unknown
Cluster 1	64.65%	0.00%	28.00%
Cluster 2	0.10%	0.00%	70.36%
Cluster 3	35.25%	100.00%	1.64%
	100%	**100%**	**100%**

Table 4: Clustering with 3 centers using filtered features

2.5 Clustering with Streaming K-Means [11 points]

When data arrive in a stream, we may want to estimate clusters dynamically and update them as new data arrives. Spark's MLLib provides support for the streaming k-means clustering algorithm that uses a generalization of the mini-batch k-means algorithm with **forgetfulness**.

a. Show why we can use streaming K-Means by deriving its update rule and then describe how it works, the pros and cons of the algorithm, and how the forgetfulness value balances the relative importance of new data versus past history. [3 points]

Streaming K means is a generalization of the mini-batch K means algorithm

1. Initialize K centers as a set C
2. Update the centers t times
   a. Sample b data points as a batch M
   b. Assign the points in M to the closest center in set C
   c. Update the set C based on the assigments in batch M
      $c_{t+1} = \frac{\alpha c_t n_t + x_t m_t}{\alpha n_t + m_t}$ and $n_{t+1} = n_t + mt$
    where
      c_t = the current cluster center
      n_t = the number of points currently assigned to c_t

x_t = the current batch cluster center
m_t = the number of points assigned to x_t
α = the decay factor (forgetfulness value)
$\alpha = 1$ retains all memory from the beginning
$\alpha = 0$ retains only the most recent batch

Setting a decay value < 1 is similar to using an exponential weighted moving average, where the most recent batches are weighted more than the older batches.

The pros of using streaming Kmeans is that it can be used on real time data, and is not dependent on having the complete set of data available to perform the computation.

The cons of using streaming Kmeans is that the means are dependent on the order in which the data arrives, and its forgetfullness setting, which may not reflect the actual Kmeans cluster taken over the complete data set. It is also a more expensive computation vs the regular Kmeans algorithm.

c. Compare clustering for the $k = 3$ case with the ground truth phenotypes that you computed for the rule-based PheKB algorithms. Specifically, for each of *case*, *control* and *unknown*, report the percentage distribution in the three clusters for the two feature construction strategies. Report the numbers in the format shown in Table **??** and Table **??**. [3 points]

Percentage Cluster	Case	Control	Unknown
Cluster 1	76.95%	8.86%	69.90%
Cluster 2	8.71%	47.47%	13.38%
Cluster 3	14.34%	43.67%	16.72%
	100%	**100%**	**100%**

Table 5: Clustering with 3 centers using all features

Percentage Cluster	Case	Control	Unknown
Cluster 1	0.00%	0.00%	8.92%
Cluster 2	100.00%	100.00%	36.82%
Cluster 3	0.00%	0.00%	54.26%
	100%	**100%**	**100%**

Table 6: Clustering with 3 centers using filtered features

2.6 Discussion on K-means and GMM [8 points]

We'll now summarize what we've observed in the preceeding sections:

a. Briefly discuss and compare what you observed in 2.3b using the k-means algorithm and 2.4b using the GMM algorithm. [3 points]

K-Means and GMM on all features both performed worse than on the filtered features. For k=3 on all features, the purity for K-Means @ 0.5686 was better than GMM @ 0.4783 for all. While the purity for filtered features was better for GMM @ 0.78055 vs K-Means @ 0.56871. However purity is not an indication of accuracy, the accuracy for K-Means on filtered features was 0.5486, while GMM was only 0.3745.

b. Re-run k-means and GMM from the previous two sections for different k (you may run it each time with different k). Report the purity values for all features and the filtered features for each k by filling in Table **??**. Discuss any patterns you observed, if any. [5 points]

NOTE: Please change k back to 3 in your final code deliverable!

	K-Means	K-Means	GMM	GMM
k	All features	Filtered features	All Features	Filtered features
2	0.52522	0.35202	0.47831	0.56698
5	0.60114	0.87020	0.50949	0.89373
10	0.68140	0.87574	0.58785	0.89477
15	0.69523	0.89685	0.57267	0.89997

Table 7: Purity values for different number of clusters