Project 3

Jess Hansen, Osbaldo Matias, Noah Hoberg

5/5/2020

K- Means Algorithm Implementation

Average WSS/BSS measures for SigGenes file

	k	k*2	k*3
Euclidean: WSS	227613948075.74863	176252967886.81323	150353803392.58063
Euclidean: BSS	241575839071.1494	280312157383.6445	299827663132.667
Manhatten: WSS	227022795268.6914	170764624362.41177	142245958873.30527
Manhatten: BSS	240390296428.43073	284222687924.7884	304857725150.855

Average WSS/BSS measures for AllGene file

	k	k*2	k*3
Euclidean: WSS	181914218639206.84	157908127447018.03	145205371861500.62
Euclidean: BSS	20790751704849.79	43619523341237.08	55456852196072.16
Manhatten: WSS	180575328288321.1	152362480834826.0	140022180335354.92
Manhatten: BSS	22456669473106.695	48616826124367.26	61075452190213.8

SigGenes - WSS Averages for 10 Iterations

	2	4	6
Euclidean: WSS	227416897140.06293	179737986227.9125	144711438442.48746
Manhatten: WSS	227022795268.6914	171643646809.20795	139577301669.75018

SigGenes – Ratio of (average BSS/average WSS) for 10 Iterations

	2	4	6
Euclidean: Ratio	1.0605221565471599	1.5885406658008143	2.1332751211943735
Manhatten: Ratio	1.0588817574196383	1.6667249954590602	2.1932436383306047

AllGenes- WSS Averages for 10 Iterations

	2	4	6
Euclidean: WSS	182981351037851.8	160913074662062.53	143605017405827.66
Manhatten: WSS	181172812497139.72	153178685958066.78	138858271270524.92

AllGenes – Ratio of (average BSS/average WSS) for 10 Iterations

	2	4	6
Euclidean: Ratio	0.10953219749818172	0.25649159050466763	0.3918507383089917
Manhatten: Ratio	0.12090520799482693	0.3167215616010049	0.4388119964498451

(1) Do you notice the following validity measures (*WSS*, *BSS*/*WSS*, and *info gain*) agreeing for *AllGenes.csv*? Please elaborate.

Yes, although there is a relatively small difference between the Euclidean and Manhatten distance measures for all k's, the higher average WSS value for all the k's correlates with a lesser ratio value.

(2) What about for *SigGenes.csv*? Again, please elaborate.

Yes, it is very similar to what is occurring with the AllGenes data. The higher average WSS value for all the k's except for one in this particular execution, correlates with a lesser ratio value.

(3) What is the best distance measure/*K* combination according to *WSS* alone, *BSS*/*WSS*, and *information gain* for *AllGenes.csv*? Why?

For AllGenes.csv, the best distance measure and k combination according to WSS alone is using Manhatten distance measure with a k value of 6.

The best distance measure and k combination according to BSS/WSS alone is using Manhatten distance measure with a k value of 6.

(4) What about for *SigGenes.csv* and why?

For SigGenes.csv, the best distance measure and k combination according to WSS alone is using Manhatten distance measure with a k value of 6.

For AllGenes.csv, the best distance measure and k combination according to BSS/ WSS alone is using Manhatten distance measure with a k value of 6.

(5) Overall, which of the two datasets clusters is better and when?

Overall, the dataset clusters from SigGenes.csv are better when k=6.

Group Evaluation and percentage marks

Jess 40%

Osbaldo 30%

Noah 30%

In the beginning, we all started by creating our own initial programs to see how far each of us could get on our own. We gave ourselves a week to see how much we could each get done and we let each other know if we were having difficulties with any parts of the assignment. The next time we met, we compared programs and decided who had a better start to continue building the program with. That first week not much progress was made by any of the group members so we decided that moving forward, we would try to work together as much as possible. We then shared this program over Github. We used Jess's program which had the initial distance measures and WSS/BSS calculations. We then collaborated multiple times over the next couple weeks and received a lot of help from both Osbaldo and Noah debugging and getting the values correct. Working over zoom was not ideal but we made it work by constantly screen sharing and annotating. Jess was our primary coder and most of the initial setup was done by him. Noah was a little more knowledgeable in coding python and was a big help in setting up the loops and conditions we needed to iterate through the datasets. Osbaldo helped as much as possible by reviewing the code and pointing out any mistakes and constantly referred to the slides to

make sure the calculations were correct. Toward the end, we couldn't get tables to work on Jupyter and so we switched to Pycharm and Osbaldo coded the tables with assistance from Jess and Noah. We then all worked together to get the averages for all iterations and put them into a nicely formatted table. Finally, we all worked together to write out and organize this report. We all agree with this group evaluation and think that the percentage marks are fair.