Weekly Journal

Leila Uy

May 31, 2021

1 Work Update

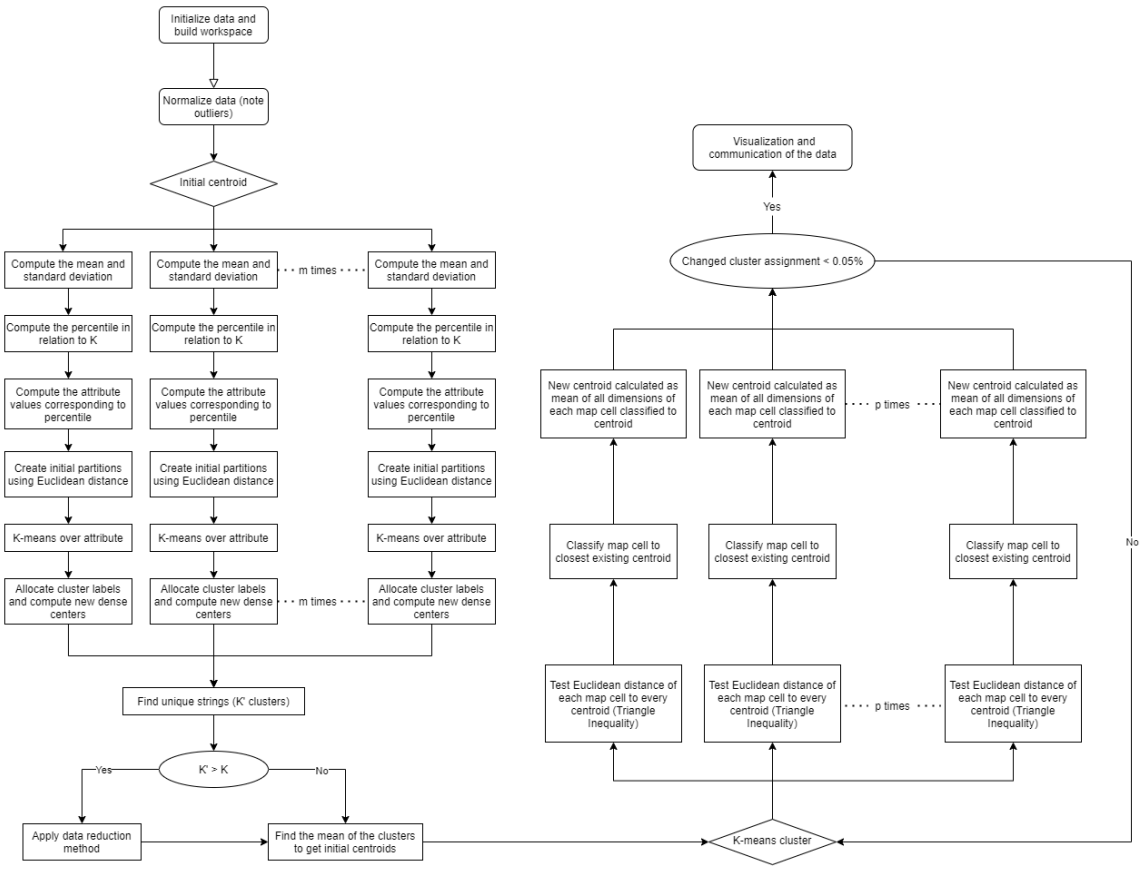
For this past week, I focused on creating a structure for my parallel K-means clustering algorithm. When reviewing the base code, Jishnu and I saw that you used `kmeans` which is the part we are most focused on adding upon. I am thinking that we wil not be using that function because we want to experiment with different forms of optimization.

1.1 Speed Testing

Last week, I said that I wanted to test the code for this week but I had a realization that different processors will result in different times. My laptop has an Intel Core i7-7560U CPU with clock speed of 2.40GHz which is different than the processors offered by AWS which include Intel Xeon, AMD EPYC, and AWS Graviton. Therefore, for more consistent timing, I should do speed testing later.

1.2 Flowchart

Figure 1: Flowchart of the parallel K-means clustering algorithm that I want to create



2 Literature Review

2.1 Initial centroid

The performance of iterative clustering algorithms like k-means clustering depend on initial cluster centroids. The more clusters there are, the more possibility that the random initial cluster centroids fail to find all the clusters correctly or lead to poor optimization.

Several studies in the past have determined several algorithms in creating more accurate initial centorids. Jishnu referred me to a paper that based their algorithm on the observation that patterns can be very similar to each other and attributes will provide information about initial cluster centers [2]. I incorporated this algorithm into my flowchart but I am open to other algorithms to finding initial centroids.

A 2020 research paper did a comparison between three hybrid methods of k-means clustering (genetic, MST, and hierarchical) [4]. They found that there was no significant improvement between the hybrid methods and the ordinary k-means algorithm and in some cases had poorer performance. Previous studies reported better performance, so the article encouraged more simulation, but it is good to note that my algorithm could worsen our performance and not show a significant difference in clusters. Therefore, it would be good to experiment with different algorithms.

In the beginning, I will initially use random selection for my initial centroids and focus on the optimization of the k-means clustering algorithm through parallelization and triangular inequality similar to the article we read first week [3]. Once I complete the rest of my K-means clustering in parallel, I want to go back and change the code to use more accurate intial centroids.

2.2 K-means clustering methods

Although, I am a big fan of Kumar et al.'s approach to parallel programming, Jishnu provided an interesting article which shows a table summarizing other parallel clustering methods on page 2 [1]. The article also provides its own algorithm that sepeartes the data into equal sepearte batches to perform parallel k-means clustering and then converges the data to do k-means clustering on the entire dataset.

References

[1] Rasim Alguliyev, Ramiz Aliguliyev, and Lyudmila Sukhostat. Parallel batch k-means for big data clustering. *Computers & Industrial Engineering*, 152, 2021.

[2] Shehroz Khan and Amir Ahmad. Cluster center initialization algorithm for k-means clustering. *Pattern Recognition Letters*, 25(11):1293–1302, 2004.

[3] Jitendra Kumara, Richard Millsa, Forrest Hoffmana, and William Hargrove. Parallel k-means clustering for quantitative ecoregion delineation using large data sets. *Procedia Computer Science*, 4:1602—-1611, 2011.

[4] Saeedeh Pourahmad, Atefeh Basirat, Amir Rahimi, Marziyeh Doostfatemeh, and Rafik Karaman. Does determination of initial cluster centroids improve the performance of k-means clustering algorithm? comparison of three hybrid methods by genetic algorithm, minimum spanning tree, and hierarchical clustering in an applied study. *Computational and mathematical methods in medicine*, 2020:7636857–11, 2020.