Taylor Kim
taylor97@csu.fullerton.edu
@893438416
CPSC479
12/08/2023

Project #2: K-Means Clustering in CUDA (with multidimensional datasets)

Table of Content

README.md

Project #2 - K-Means clustering in CUDA

Description:

Implemented K-Means clustering in CUDA with multidimensional datasets.

There are two datasets used, obtained from UCI Machine Learning Repository.

1. Iris dataset from https://archive.ics.uci.edu/dataset/53/iris

   contains N=150 datapoints, F=4 features, K=3 expected clusters

2. Wine Quality dataset from https://archive.ics.uci.edu/dataset/186/wine+quality

   contains N=1599 datapoints, F=11 features, K=6 expected clusters

For loading these datasets for clustering, last column for target (categorical) values has been removed, but used at the end of the program to check for the clustering's correctness (if implemented clustering outputs expected clustering).

NOTE:

1. Current implementation does not check for convergence in the k-means clustering algorithm, rather uses MAX_ITER to repeat two steps of the alorithm (assignment and update).
2. Expected K cluster values are assumed by looking at the dataset's target (categorical/classification) values. Therefore, fine-tuning for choosing the K value has not been implemented in the code.

Group member:

Taylor Kim taylor97@csu.fullerton.edu

Pseudocode for K-Means Clustering in CUDA

```
// load data
// initialize datapoints to host
Allocate memory for host
Allocate memory for device using cudaMalloc
Initialize k centroids randomly (or systematically)
cudaMemcpy(d_, h_, cudaMemcpyHostToDevice) // for datapoints and centroids
While not converged  // until converges
        // assign datapoints to minimum distance cluster using Euclidean Distance
        // blocks = (n datapoints+threads-1)/threads , threads = 1024
        assignment<<<blocks, threads>>>(d_datapoints, d_clusters_assign, d_centroids)
        // update centroids
        update<<<1, k clusters>>>(d_datapoints, d_clusters_assign, d_centroids)
        Check if converged and break out of loop
// for datapoints assigned to cluster
cudaMemcpy(h_clusters, d_clusters, cudaMemcpyDeviceToHost)
Output datapoints assigned to K clusters
Free all host memory and device memory cudaFree()
```

How to run the code (2 separate codes for each dataset.)

1. Connect to CSUF VPN GlobalProtect
2. Connect to ssh -l USERNAME aries.ecs.fullerton.edu

For Iris dataset:

3. Comple: nvcc kMeans_cuda_iris.cu
4. Run: ./a.out

For WineQuality dataset:

3. Comple: nvcc kMeans_cuda_wine.cu
4. Run: ./a.out

NOTE: resulting clusters and its datapoints are saved to result_iris.csv or result_wineQuality.csv for better readability.

Two snapshots of code executing for some two distinct values of N.

Iris dataset (N = 150 datapoint, F=4 features, K=3 expected clusters)

```
taylor97@prudence:~/cpsc479/project2_kMeans_CUDA$ nvcc kMeans_cuda_iris.cu
taylor97@prudence:~/cpsc479/project2_kMeans_CUDA$ ./a.out
Final Clusters:
Datapoint Row 0 assigned to Cluster 2
Datapoint Row 1 assigned to Cluster 2
Datapoint Row 2 assigned to Cluster 2
Datapoint Row 3 assigned to Cluster 2
Datapoint Row 4 assigned to Cluster 2
Datapoint Row 5 assigned to Cluster 2
Datapoint Row 6 assigned to Cluster 2
Datapoint Row 7 assigned to Cluster 2
Datapoint Row 8 assigned to Cluster 2
Datapoint Row 9 assigned to Cluster 2
Datapoint Row 10 assigned to Cluster 2
Datapoint Row 11 assigned to Cluster 2
Datapoint Row 12 assigned to Cluster 2
Datapoint Row 13 assigned to Cluster 2
Datapoint Row 14 assigned to Cluster 2
Datapoint Row 15 assigned to Cluster 2
Datapoint Row 16 assigned to Cluster 2
```

…Continues down to 149

WineQuality (red wine only) dataset

(N = 1599 datapoint, F=11 features, K=6 expected clusters)

```
taylor97@prudence:~/cpsc479/project2_kMeans_CUDA$ nvcc kMeans_cuda_wine.cu
taylor97@prudence:~/cpsc479/project2_kMeans_CUDA$ ./a.out
Final Clusters:
Datapoint Row 0 assigned to Cluster 4
Datapoint Row 1 assigned to Cluster 1
Datapoint Row 2 assigned to Cluster 2
Datapoint Row 3 assigned to Cluster 1
Datapoint Row 4 assigned to Cluster 4
Datapoint Row 5 assigned to Cluster 2
Datapoint Row 6 assigned to Cluster 1
Datapoint Row 7 assigned to Cluster 4
Datapoint Row 8 assigned to Cluster 5
Datapoint Row 9 assigned to Cluster 0
Datapoint Row 10 assigned to Cluster 1
Datapoint Row 11 assigned to Cluster 0
Datapoint Row 12 assigned to Cluster 1
Datapoint Row 13 assigned to Cluster 4
Datapoint Row 14 assigned to Cluster 3
Datapoint Row 15 assigned to Cluster 3
```

…Continues down to 1598

Conclusion

K-Means clustering is an unsupervised learning algorithm using unlabeled data to find similarity in one cluster or dissimilarity between other clusters. The problem with this algorithm is that by itself, it is *incapable of handling large datasets* and finding the minimum of k-means cost function is *NP-hard problem* when d>1 and k>1. Other drawbacks of k-means algorithm are choosing an incorrect number of k clusters can result in poor clustering performance and data containing outliers and clusters with different shapes and sizes may result centroids to be misplaced.

Using the Iris dataset, the code outputs expected clusters which is described under “Iris K-Means Clustering in CUDA” section. However, wine quality dataset resulted in poor clustering performance due to its poor data quality, described in detail under “WineQuality (Red) K-Means Clustering In CUDA” section below.

Iris K-Means Clustering in CUDA

First classification for Iris-setosa were correctly clustered as this classification is linearly separable from the other two classes (Iris-versicolor and Iris-virginica) according to UCI Machine Learning Repository.

Datapoint misplacement in cluster:
2 datapoints (Iris-versicolor) clustered to Iris-virginica
14 datapoints (Iris-virginica) clustered to Iris-versicolor

These datapoints did not clustered to their expected classification due to Iris-versicolor and Iris-virginica clusters/classification having correlations on their features. By observing the original datasets graph, these two clusters are located very closely which makes K-means clustering difficult to correctly cluster their datapoints.

WineQuality (Red) K-Means Clustering in CUDA

According to UCI Machine Learning Repository, the dataset contains mostly normal wines without including excellent or poor ones meaning that the quality rating of the wine ranges only between 3 to 8. Also, as the K-Means algorithm works best with the hard clusters, it was hard to determine the correct clustering done with this dataset. The reason why resulting clusters were incorrect and hard to determine comes from the poor quality of the dataset.

Reference

https://archive.ics.uci.edu/dataset/53/iris
https://archive.ics.uci.edu/dataset/186/wine+quality