Project Proposal

Yige Hu and Zhiting Zhu

1 Problem Description

K-means clustering is a unsupervised learning algorithm for solving clustering problem. Formally, the problem states as follows [3]: Given a set of data points $\{x_i | i = 1..n\} \subseteq \mathbb{R}^d$, k-means clustering aims to partition the n data points in to k($\leq n$) sets S = $\{S_1, S_2, ..., S_k\}$ so as to minimize the within-cluster sum of squared errors,

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \parallel x - \mu(S_i) \parallel$$

where $\mu(S_i)$ is the mean of points in S_i.

2 Standard Sequential Algorithm

- Choose the number of clusters, k.
- Randomly generate k points as cluster centers.
- Assign each point to the nearest cluster center.
- Recompute the new cluster centers.
- Repeat the previous two steps until some convergence criterion is met.

3 Plan

We plan to use GPU and CUDA to implement the parallel version of k-means algorithm. For GPU, we will use NVIDIA GPU Tesla K20c to test and bench mark our implementation.

We will compare our implementation with two kinds of baselines:

- A sequential k-means implementation on CPU.
- Some existing GPU k-means implementation we find on the Internet[1, 2].

References

[1] A cuda implementation of the k-means clustering algorithm. `https://github.com/serban/kmeans`. Accessed: 03-05-2015.

[2] gpuminer. https://code.google.com/p/gpuminer/. Accessed: 03-05-2015.

[3] k-means clustering. http://en.wikipedia.org/wiki/K-means_clustering. Accessed: 03-05-2015.