INTRODUCTION

Briefly, for large-scale clustering task, I use kmeans++ first and then the coreset construction method to initialize the centers and finally pass the hard baseline. With kmeans++, we can sample the initial points not in random way but using importance sampling. After that, we can use these initial points to construct coresets by a new sampling function, which is used to summarize from large dataset and stored as a point with weight representing the number of the points belonging to it. Therefore, in the final step in reducer, we can do weighted-kmeans clustering on initial centers, which are initialized using kmeans++.

IMPLEMENTATION

For details, in the mapper, I first need to initial k=200 centers. I have implemented it as a function kmeanspp because we need to use it in both mapper and reducer. As mentioned above, kmeans++ first choose a center randomly and then choose next centers based on those previous chosen centers, so that it is not uniformly sampled. Then, based on kmeans++ chosen centers and original data points, we sample new sets of centers from q(x) with weights written as formula in slides, where Bi means the points in that cluster B the same as x, and the number of coresets for each mapper is 2500. Finally, after getting 2500\*9 coresets , we just reweight the initial chosen center from coresets iteratively, using the other coresets belonging to it and weighted mean to compute the updated one, with totally 15 iterations.