Tailin Lo tl1720 N15116873 Homework 4

1 Section 2.4

The trick is how to speed up KMeans. One of expansive function of KMeans is the partition steps. The original partition step is to calculate the distance between a point and each cluster. This trick is basically from Elken algorithm [1]. The concept of this algorithm is to reduce the number of calculating the distance as many as possible. From triangle inequality, we can decide upper-bound and lower-bound of the distance between point and point. By updating the upper-bound and lower-bound of the distance in every iteration, it doesn't have to calculate exact the distance. Calculating exact distance between centroids and data point only when that point change from current cluster to the other cluster.

2 Section 3.3

The trick of feature extraction is representation of sparse matrix in Python code. Unluckily, there are no sparse matrix in my case. Thus, it's not efficient to represent my data to sparse representation. The only thing I can do is to normalize the features before projecting data points on the feature space. By doing this first, it doesn't have to recalculate normalization of each feature vector when doing projection.

3 Section 4

There are five parameters in the program, i.e. window size, class size, subsample size, training ratio, cluster number. The following is the definitions of those parameters.

1. Window Size : the size of filter of extract local feature

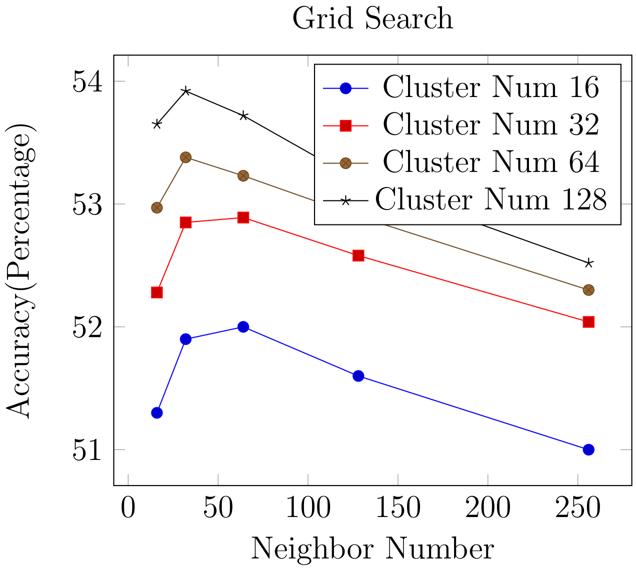
2. Class Size : There are 61 classes in my dataset, and use class size to choose the size of class.

3. Subsample Size : There are 92 images in each class, and use subsample size to choose the size of image.

4. Training Ratio : The ratio of size of training and testing

5. Cluster Number : The size of bag of words.

Because I ran the program in my local computer, I fixed window size to be 3. I sweeped cluster number in the sequence 16, 32, 64, 128. And for each cluster number, I sweep the neighbor number in the sequence 16, 32, 64, 128, 256.



From the above figure, the optimal neighbor number is 32. And the cluster number is 128 in this case. But the optimal for the cluster number is still not achieved. I can still find cluster number 256, 512, ... etc..

References

[1] Charles Elkan, *Using the Triangle Inequality to Accelerate K-Means*