1 Kmeans

1.1 When k is large

When the number of clusters is large, the amount of distortions gets reduced. This is because the algorithm will be able to find a centroid very nearer to the sample points. Ultimately, when the number of clusters equals the number of samples, the distortion value will be zero.
The code (.py) file can be found at
`https://github.com/rajegannathan/computational-machine-learning/blob/master/hw2/kmeans.py` The png files can be found in the repository named as 'distortion_plot.png', 'centroid_plot.png'

1.2 When k is small

When the value of k is reduced (minimum = 1) the amount of distortion increases. This also corresponds to the possition where we will not be able to learn anything from the clustering as all the points are essentially grouped together.

1.3 Is it easy to draw conclusions

Since the centroid selection is done at random, the value of distortion can vary arbitrarily. This might make the algorithm difficult for drawing inferences.

1.4 What is problematic with the current algorithm

Again, since the centroid selection is done at random, the algorithm might converge for a centroid that is not optimal. This might result in very poor clustering performance.

2 Kmeans-multi

Code can be found at
`https://github.com/rajegannathan/computational-machine-learning/blob/master/hw2/multikmeans.py` The plots can be found as 'distortion_plot_multi.png' and 'centroid_plot_multi.png' in the same folder.

2.1 Observation

The modified kmeans algorithm runs by selecting random centroids for max_iterations times and then runs the convergence algorithm similar to the original k-means algorithm. Since we select multiple centroids randomly, there is a good chance of finding the optimum centroid for a given dataset through the modified k-means algorithm. The only drawback is that the run time is max_iterations times the run time of the original k-means algorithm.

2.2 How is new version preferable to old version

Though the new version performs better than the older version generally, we lose performance for the sake of accuracy. When we have big datasets: say, n_samples = 10M and k = 1000, the modified algorithm takes huge performance penality.

3 Kmeans++

`https://github.com/rajegannathan/computational-machine-learning/blob/master/hw2/kmeans_plus_plus.py` The plots can be found at 'plus_distortion_plot.png' and 'plus_centroid_plot.png' in the same folder.

3.1 Observation

The Kmeans++ algorithm is an elegant and beautiful algorithm. It uses the square of distance heuristic to pick up the next cluster. It works by finding the shortest distance between a point and any of the already selected clusters and converting all the measures as probabilistic values.

This makes sure that consequtive centroids are selected far apart from the already selected centroids.

The alogirhtm is similar to the original k-means in terms of the convergence logic.

3.2 Performance

This algorithm is very fast and almost as accurate as the multi-kmeans algorithm.

4 Project

As of now we are able to implement only the feature preprocessing of EEG data. The steps are as explained in the ProjectProposa.pdf.

5 Feedback

5.1 How long it took to complete the assignment

Implementing the k-means and related algorithms was trivial. I was able to implement it in a span of 5 hours. Figuring out how to get the preprocessing done for EEG data took a lot of time. I had to spend almost four days for going through the various tutorials, papers and baseline implementation just for understading the preprocessing pipeline.

5.2 Unclear concepts

Conceptually, I am clear with all areas of the assignment. Though I didn't get time to implement the Bag of feature implementation, I went through the tutorial and was able to understand the concept clearly.

5.3 Resubmission

Since I didn't get time to implement the Bag of feature implementation, will it be okay if I implement it in a day or two and submit the assignment? Also, my matplotlib's backend got messed up and I was unable to call show function on a plot. This is one of the main reason why I am using savefig. I will try to fix this issue and update my codebase. (This was working fine in my previous VM but somehow is not working properly in my new virtual machine).