Project 3

ESE 545, Data Mining: Learning from Massive Datasets

November 13, 2017

Due at 11:59PM on **November 22, 2017**

This question consists of several parts. You are required to solve the problems using Python, Matlab or C and turn in your code. You are allowed to work in groups of at most two members. You should submit your code with a brief report containing responses to each part. Turn in one report and script per group. Upload a zipped file containing the report and the code on Canvas.

Question 1.1. For the first part of the project, you are required to download the Yahoo user click log dataset, which can be found at the following link: `https://webscope.sandbox.yahoo.com/catalog.php?datatype=r&did=49`). The dataset contains over 45 million user visits to the Yahoo news articles, and clicks are binary features indicating whether a user has clicked on a given article. The details of the data format are described in the Readme. The dataset is compressed into a 1.1 GB zip file, which contains ten more zip files. For the first part of the project, you are only required to test using any one of the ten zip files, since the dataset is too large otherwise. Unzip the first part of the data, *ydata-fp-td-clicks-v1_0.20090501.gz*, and read it in to a matrix. It may be helpful to use Python's gzip library, which allows you to read data from zipped files. **20 pts**

Question 1.2. In the next two parts of the project, you should implement an online version of the k-means clustering algorithm. As noted in class, k-means is solving a non-convex optimization problem, for which there are many local minima. This means that the initialization of the cluster centroids can greatly affect the final result. In this part, implement **both** randomized centroid initialization, as well as the k-means++ initialization algorithm discussed in class. **30 pts**

Question 1.3. Once the cluster centroids are chosen, the next step is to iteratively improve the solution until a local optimum is achieved. As discussed in the lectures, Lloyd's heuristic is one approach, but this requires taking a pass over the entire dataset each time, which is not feasible for the dataset here. Instead, implement Mini-batch k-means, which will adjust the centroids by taking a subset of the data each time. Generate plots showing the mean, minimum, and maximum distance to the cluster centroids for the dataset, and run this for several values of k, ranging from k = 5,..., 500. Which k has the lowest error? Using plots, compare the performance of the kmeans++ initialization and randomized centroids. **50 pts**