IE 529: Stats of Big Data and Clustering
Computational assignment 2
Due: Monday, December 18, 2017

This assignment will be worth 75 points, with each part worth 25 points. You must complete the assignment alone.

There will be 2 data sets posted for you to test your algorithms on. You should code all routines yourself, "from scratch", but you are welcome and encouraged to compare your codes to existing functions in Matlab or Python (or whatever coding environment you are using).

You should submit a report in pdf format, documenting and discussing the results you found for **all** of your algorithms, on **both** of the data sets. You should compare the results of the different algorithms to each other as well. Color coded scatter plots indicating the cluster assignments would be appreciated for illustration of results and discussion. You should attach your personal codes as an appendix to your report.

I. Write a basic implementation of Lloyd's algorithm for a large set of data in $\mathbf{R}^d$ (i.e., to find a Voronoi partition and a set of K centroids). Your algorithm should attempt to solve the classic *K-means* problem, for any user-selected positive integer value K.

- Assume the input data is given to you in a matrix $X \in \mathbf{R}^{N\times d}$, where each row in X corresponds to an observation of a d-dimensional point. That is, your inputs will be a user-provided matrix X and the number of clusters K.
- Your outputs should be (i) a matrix $Y \in \mathbf{R}^{K\times d}$, where row j contains the centroid of the j^{th} partition; (ii) a cluster index vector $C \in \{1, 2, \dots K\}^N$, where $C(i) = j$ indicates that the i^{th} row of X (or the i^{th} observation x_i) belongs to cluster j; and (iii) the final objective function value, i.e., the best distortion, or averaged distance value, D obtained.
- Convergence may be based on a norm-based comparison of the iterates of Y, i.e., $\|Y_{p+1} - Y_p\| < tol$, OR on a norm-based comparison of the distortion achieved $\|D_{p+1} - D_p\| < tol$. Choose tol to be (1) 1×10^{-5}, and (2) a different value of your choice, with your reasoning provided.

II.1 Write a basic implementation of the "GreedyKCenters" algorithm (described in the reading by S. Har-Peled, and discussed in class). Your algorithm should attempt to solve the classic *K-centers* problem, for any user-selected positive integer value K. The underlying distance function used in your algorithm should be the Euclidean distance, and your objective should be to *minimize* the *maximum* distance between any observation $x_i \in X$ and it's closest center $c_j \in Q$, i.e., to find Q giving

$$\min_{Q\subset X,\ |Q|=K} \left(\max_{x_i\in X} (\min_{c_j\in Q} \|x_i - c_j\|_2) \right) \tag{1}$$

- You can again assume the input data is given to you as a matrix $X \in \mathbf{R}^{N\times d}$, and a positive integer K, as in **I.**
- Your output should be a matrix $Q \in \mathbf{R}^{K\times d}$ containing the final K d-dimensional centers, and the objective function value, i.e., the final $\max_{x_i\in X}(\min_{c_j\in Q} \|x_i - c_j\|_2)$ obtained.

- You do not need a convergence criteria for this algorithm.

II.2 Write a basic implementation of the single-swap heuristic for which you try to improve the solution to the *K-centers* problem in **II.1** by a implementing a series of "swaps". If Q is your current set of centers, and you make a single swap, giving $Q_{new} = Q - \{c_j\} \cup \{o\}$, then you should replace Q with Q_{new} whenever the new objective value, that is the computed value for (1), is reduced by a factor of $(1 - \tau)$. When there is no swap that improves the solution by this factor, the local search stops. Let $\tau = 0.05$.

III. Write an implementation of the Spectral Clustering algorithm, using either basic unnormalized clustering or normalized clustering (refer to the reading by Luxborg for details). Assume you are given a matrix of data $X \in \mathbf{R}^{N \times d}$, and you would like to identify some user-selected number of clusters, K. Your outputs should be:

- a weighted adjacency matrix, W, using the **Gaussian similarity function** based on the Euclidean distance (with parameter value σ of your choice but clearly stated) and a **k-nearest neighborhood structure** (where k is also your choice and clearly stated);
- a matrix U containing the first K eigenvectors of the Laplacian L (or generalized eigenvectors for the normalized case).;
- a cluster index vector $C \in \{1, 2, \ldots, K\}^N$, where $C(i) = j$ indicates that the ith row of U belongs to cluster j.

Please let us know if you have any questions!