CMSC25400/STAT 27725 **Homework 1**

University of Chicago, Winter 2017 due Thursday, October 5 at 11:59pm

1. **(30 points)** As we saw in class, k-means clustering minimizes the average square distance distortion

$$J_{\text{avg}^2} = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} d(\mathbf{x}, \boldsymbol{m}_j)^2, \tag{1}$$

where $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|$, and C_j is the set of points belonging to cluster j. Another distortion function that we mentioned is the intra-cluster sum of squared distances,

$$J_{\text{IC}} = \sum_{j=1}^{k} \frac{1}{|C_j|} \sum_{\mathbf{x} \in C_j} \sum_{\mathbf{x}' \in C_j} d(\mathbf{x}, \mathbf{x}')^2.$$

(a) Given that in k-means, $\boldsymbol{m}_j = \frac{1}{|C_j|} \sum_{\mathbf{x} \in C_j} \mathbf{x}$, show that $J_{\text{IC}} = 2\, J_{\text{avg}^2}$.

(b) Let $\gamma_i \in \{1, 2, \ldots, k\}$ be the cluster that the i'th datapoint is assigned to, and assume that there are n points in total, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$. Then (1) can be written as

$$J_{\text{avg}^2}(\gamma_1, \ldots, \gamma_n, \boldsymbol{m}_1, \ldots, \boldsymbol{m}_k) = \sum_{i=1}^{n} d(\mathbf{x}_i, \boldsymbol{m}_{\gamma_i})^2. \tag{2}$$

Recall that k-means clustering alternates the following two steps:

1. Update the cluster assignments:

$$\gamma_i \leftarrow \underset{j \in \{1,2,\ldots,k\}}{\operatorname{argmin}} \; d(\mathbf{x}_i, \boldsymbol{m}_j) \qquad i = 1, 2, \ldots, n.$$

2. Update the centroids:

$$\boldsymbol{m}_j \leftarrow \frac{1}{|C_j|} \sum_{i \,:\, \gamma_i = j} \mathbf{x}_i \qquad j = 1, 2, \ldots, k.$$

Show that the first of these steps minimizes (2) as a function of $\gamma_1, \ldots, \gamma_n$, while holding $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$ constant, while the second step minimizes it as a function of $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$, while holding $\gamma_1, \ldots, \gamma_n$ constant. The notation "$i \,:\, \gamma_i = j$" should be read as "all i for which $\gamma_i = j$".

(c) Prove that as k-means progresses, the distortion decreases monotonically iteration by iteration.

(d) Give an upper bound on the maximum number of iterations required for full convergence of the algorithm, i.e., the point where neither the centroids, nor the cluster assignments change anymore.

2. **(30 points)** Implement the k-means algorithm in a language of your choice (MATLAB, Python, or R are recommended), initializing the cluster centers randomly, as explained in the slides. The algorithm should terminate when the cluster assignments (and hence, the centroids) don't change anymore.

   Note: if you use a relatively high level language like MATLAB or Python, your code can use linear algebra primitives, such as matrix/vector multiplication, eigendecomposition, etc. directly. However, you *are* expected to write you own k–means and k–means++ functions from scratch. Please don't submit code consisting of a single call to some pre-defined "`kmeans`" function.

   (a) The toy dataset `toydata.txt` contains 500 points in $\mathbb{R}^2$ coming from 3 well separated clusters. Test your code on this data and plot the final result as a 2D plot in which each point's color or symbol indicates its cluster assignment. Note that because of the random initialization, different runs may produce different results, and in some cases the algorithm might not correctly identify the three clusters. Plot the value of the distortion function as a function of iteration number for 20 sepearate runs of the algorithm on the same plot. Comment on the plot.

   (b) Now implement the k–means++ algorithm discussed in class and repeat part (a) using its result as intialization (except for the 2D plot). Comment on the convergence behavior of k–means++ vs. the original algorithm.

3. **(up to 20 points extra credit)** You likely found that on the "toydata" dataset, most of the time even vanilla k–means clustering produces acceptable solutions. Create a dataset of your own for which (on average over many runs) k–means++ improves the performance of k–means by at least a factor of 10 in terms of the distortion function value of the final clustering. Submit the code that you used to generate the dataset.