Due: October 29, 2021, 11:59 PM

Problem 1 - k-means

- Consider a dataset of 9 points: $p_1 = (2, 10), p_2 = (2, 5), p_3 = (8, 4), p_4 = (5, 8), p_5 = (7, 2), p_6 = (6, 4), p_7 = (1, 2), p_8 = (4, 9), p_9 = (7, 5)$. Preform k-means on these points with $k = 3$, initial centroids: p_1, p_4 and p_7. Use ℓ_2-norm to calculate distances.
- If one uses ℓ_1-norm instead of ℓ_2-norm to calculate distances in k-means, what will the clusters look like? Why?
- As shown in Figure 1[1], first column shows the performance of k-means on different datasets. Please explain why k-means fails in some of these datasets.

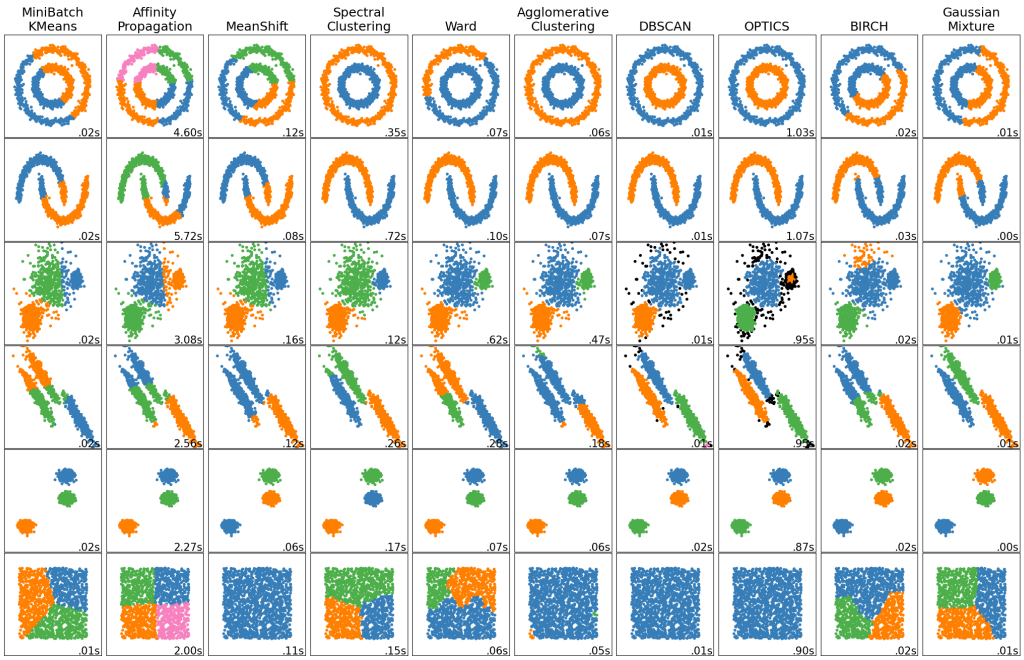

Figure 1: Problem 1 plot

Problem 2 - Evaluation

- Discuss why SSE is the most suitable objective function for k-means.
- Prove that $\sum_{i=1}^{K} \sum_{x \in C_i} (x - c_i)(c - c_i) = 0$, where K is the number of clusters, c is the center of the dataset, and c_i is the center of cluster C_i.

Problem 3 - Problem 3 (Programming) - Hierarchical Clustering

In this question, you will implement a variation of the Agglomerative Hierarchical Clustering method.

[1] https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

- Download dataset http://cs.joensuu.fi/sipu/datasets/dim032.txt.
- Using k-means (you can directly use packages from sklearn, etc.) obtain clusters (try different k). Plot SSE changes with respect to k to obtain an optimal k using this plot (there is no universal answer and the answer might change with randomization).
- Using these optimal clusters, in each iteration, merge the two closest clusters based on their MIN/MAX/average distances. This can be implemented as a function that takes parameters (see formula in the slides).
- Track the merging history and plot the hierarchical clustering dendrograms (three dendrograms for min/max/average).
- Please submit the code, and plots including SSE with respect to k, 3 dendrograms for MIN, MAX, and average distances.

Problem 4 - Text Clustering (Takes Time! - Start Early)

Download the fine foods dataset from:
http://snap.stanford.edu/data/web-FineFoods.html

Perform the following:

- Identify all the unique words that appear in the "review/text" field of the reviews. Denote the set of such words as L.
- Remove from L all stopwords in "Long Stopword List" from http://www.ranks.nl/stopwords. Denote the cleaned set as W.
- Count the number of times each word in W appears among all reviews ("review/text" field) and identify the top 500 words.
- Vectorize **all** reviews ("review/text" field) using these 500 words (see an example of vectorization here: https://bit.ly/3CcY9i4).
- Cluster the vectorized reviews into 10 clusters using k-means. You are allowed to use any program or code for k-means. This will give you 10 centroid vectors.
- From each centroid, select the top 5 words that represent the centroid (i.e., the words with the highest feature values)
- Submit the following:
  1. Top 500 words + counts for these words
  2. The top 5 words representing each cluster and their feature values (50 words + 50 values).
  3. **IMPORTANT:** submit your code and a step-by-step readme to help reproduce your results. We should be able to get the same results by running your code and by following your readme for this problem to be graded.