STAT/BIOST 527
Homework 2
Out Thursday April 20, 2023
Due Monday May 3, 2023

©Marina Meilă
mmp@stat.washington.edu

Problem 1 – Mean Shift clustering
Implement the kernel density estimator (KDE) with Gaussian kernel for data in 1 dimension (was done in Homework 1).

a. Read the input data $\mathcal{D} = \{x_1, \dots x_n\}$ from the file `hw2-p1.dat` and plot the KDE on the interval $[-4, 10]$. For this, you will create a grid $\tilde{x}_1, \dots \tilde{x}_l$ on $[-4, 10]$ and calculate $f_h(\tilde{x}_j)$, $j = 1 : l$. Choose $h = 0.4$.

b. Implement the function `mean_shift()` that takes as input the data $\mathcal{D}$, the kernel width h, an initial point x and outputs the value x' where the Mean-Shift started at x converges. Convergence should be achieved at the first iteration when the mean shift step $m(x) - x$ is smaller than $tol = 10^{-3}h$. Limit the total number of iterations to a number such as $T = 100$. [Optionally: what should `mean_shift()` return when T is reached? What do you think should happen then?]

c. Using `mean_shift()`, implement the Mean-Shift clustering algorithm. Specifically, initialize `mean_shift()` with every $x_i \in \mathcal{D}$ and record x'_i the convergence point. Plot the values $(x'_i, f_h(x'_i))$ for all the data, superimposed on the KDE graph. Make sure the graph is legible.

d. Theoretically, all x'_i which are equal would be assigned to the same cluster, but in practice none of these values will be equal. Implement one of the following heuristics.

Rounding Choose a tolerance δ. Take $\delta = 5tol$. Then compute $y_i = \texttt{round}(\frac{x'_i}{\delta})$. Now assign all i's with the same y_i to the same cluster.

Nearest neighbors Choose a tolerance δ like above. Connect all the x'_i points that have distance $|x'_i - x'_j| < \delta$. The connected components of this graph are the clusters. *For the connected components algorithm, OK to use library software. However, in 1 dimension, there is a very simple method to find them, or directly the clusters.* **Extra credit** for implementing the algorithm yourself, give clear pseudocode and/or code snippets in the submitted homework.

Choose one of the methods and compute the cluster assignments $k(i)$ for the data $\mathcal{D}$. For this problem $k(i)$ should be an integer between 0 and $K - 1$ where K is the total number of clusters you found.

Plot the data as $k(i)$ vis x_i, and superimpose the KDE on this graph. The following is an example plot. The bar-like visualization is created by scatter plot with points $\{(x_i, k(i))\}_{i=1}^n$. You will need to scale the cluster assignments, $k(i) \in \{0, \dots, K-1\}$, accordingly to make the plot of the same scale as your KDE plot; any scaling that makes the plot readable is good; labels don't need to increase from left to right, either. Output the cluster assignements in the ASCII file `p1-d.out` having n integers, each in a new line as in `hw2-p1.dat`, representing $k(i)$ for $i = 1 : n$ and submit this file.

e. Now select a subset of $n' = 25$ data points at random, and re-run steps **c, d**. Plot the graph in **d** with both clusterings on the same graph.

Problem 2 – Dendrograms and distances between clusterings – Moved to Hw 3, do not submit anything

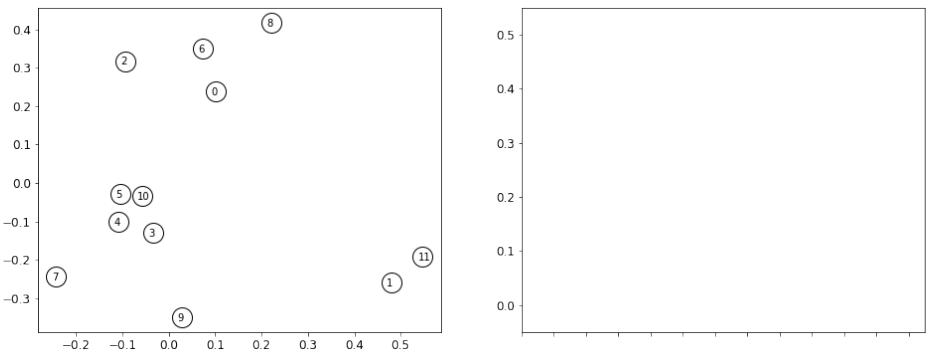

Figure 1: Plot of `hw2-toy-dendro.dat` from Problem 2.

a. Figure 1 below displays $n = 12$ data points found in file `hw2-toy-dendro.dat`. The first two columns contain the x and y coordinates and the third column contains the label. Compute all the distances between these points (you do not need to submit code or output). Using the calculated distance matrix, draw the dendrogram of this dataset obtained by the SINGLE LINKAGE ALGORITHM. Implemementation is not required. The algorithm can be "run" manually for these data and the dendrogram can be copied by hand.

Use the plot on the right to display this dendrogram; the height of a dendrogram node should be equal to the distance between the two clusters merged at this node. Recall that the distance between two clusters C_1 and C_2 in the single-linkage framework is given by

$$\text{distance}(C_1, C_2) = \min_{x \in C_1, y \in C_2} ||x - y||_2. \tag{1}$$

b. The dendrogram in Figure 2 displays the output of a different hierarchical clustering algorithm on the same data as above. On the plots below, draw the first 5 stages of this algorithm (from the top down). Stages are denoted by the number of clusters K; for example level $K = 2$ is the clustering with 2 clusters, resulting after the first split, level $K = 3$ results after the second split and has 3 clusters, etc.

c. Denote by Δ_1 the clustering at level $K_1 = 3$ in the dendrogram obtained in **a**, and denote by Δ_2 the clustering at level $K_2 = 4$ in the dendrogram in **b**. Compute the confusion matrix M of these clusterings. *Cluster labels are arbitrary, hence any permutation of rows or columns of a correct M is equally correct*

d. From the confusion matrix M obtained in **c**, calculate the Misclassification Error distance $d_{ME}(\Delta_1, \Delta_2)$.

e. From the confusion matrix M obtained in **c**, calculate N_{22}, N_{12}, N_{21} and the Jaccard index $J(\Delta_1, \Delta_2)$.

f. Calculate the $n \times K$ matrix representation $\tilde{X}_{1,2}$ for clusterings Δ_1, Δ_2 (see Lecture II, part 3). Verify that $M = \tilde{X}_1^T \tilde{X}_2$. Provide a code snippet that prints M, $\tilde{X}_1$, and $\tilde{X}_2$, and computes and prints

$$||M - \tilde{X}_1^T \tilde{X}_2||_F,$$

where $||\cdot||_F$ denotes the Frobenius norm of a matrix.

g. Prove that $M = \tilde{X}_1^T \tilde{X}_2$ for any two (arbitrary) clusterings represented by $\tilde{X}_1, \tilde{X}_2$.

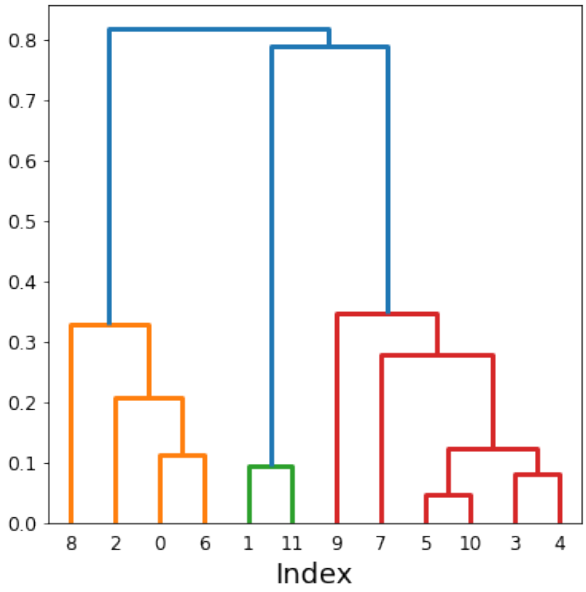

Figure 2: Dendrogram on `hw2-toy-dendro.dat`.

h. Denote $\tilde{Z}(\Delta) = \tilde{X}\tilde{X}^T$. Show that $Z \in \{0,1\}^{n\times n}$. Find a simple expression for the Jaccard index of clusterings $\Delta_{1,2}$ as a function of $\tilde{Z}_{1,2}$ and n. Both matrix and elementwise arithmetic or Boolean operations with $\tilde{Z}_{1,2}$ are allowed as long as they are on all elements. E.g. $\tilde{Z}_1 + 3\tilde{Z}_2$, $\max(\tilde{Z}_1, \tilde{Z}_2)$; $\tilde{Z}_1 + C$ (where C is a constant matrix).

Problem 3 – Mean-Shift is gradient ascent
One may wonder why not use gradient ascent to find the peaks of $f_h(x)$ and how would it compare with Mean-Shift. Here you will prove that Mean-Shift is actually a gradient ascent algorithm with automatically chosen step size.

Calculate the expression of $\nabla \ln f_h(x)$; then show that $\nabla \ln f_h(x) \propto m(x) - x$ the Mean-Shift step, with a proportionality constant independent of x.

Problem 4 – NP clustering example Give a concrete real world example, preferable from your own research experience, where collecting more data reveals more clusters. Extra credit for a more specific example.

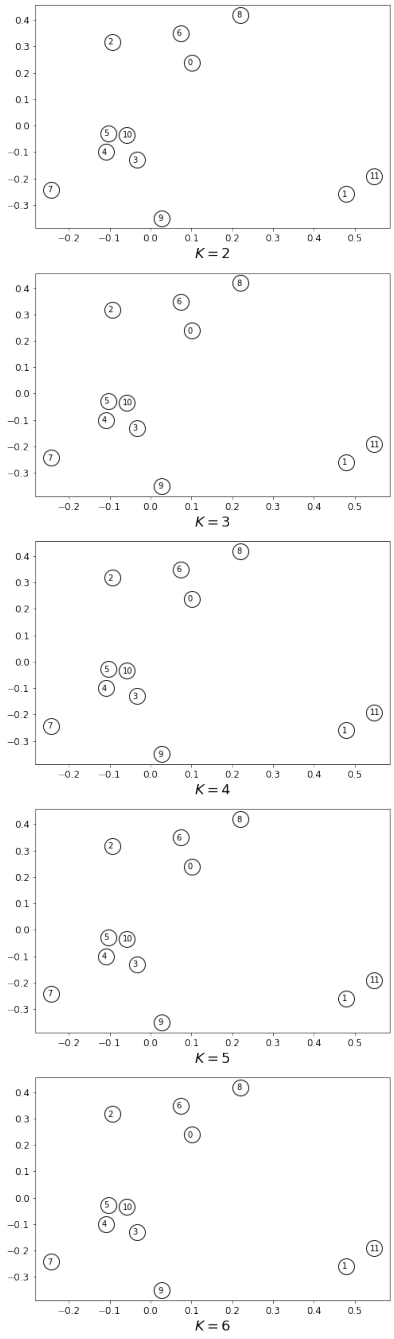

Figure 3: Plots on which to draw cluster assignments at each stage of the dendrogram from Figure 2.