Asmt 3: Clustering

Turn in a hard copy at the start of class:
Wednesday, March 06
20 points

Overview

In this assignment you will explore clustering: hierarchical and point-assignment. You will also experiment with high dimensional data.

You will use two data sets for this assignment:

- `http://www.cs.utah.edu/~jeffp/teaching/cs5955/A3/C1.txt`
- `http://www.cs.utah.edu/~jeffp/teaching/cs5955/A3/C2.txt`

These data sets each describe the location of 26 points, each on one line of the file. The first character is a label (from the lower case letters a,b,c,d,e,...). Then separated by white space are two numbers, the x and the y coordinate. We'll use L_2 distance to say which are close

$$\mathbf{d}(a,b) = \sqrt{(a.x - b.x)^2 + (a.y - b.y)^2}.$$

The data sets are small enough that it may be possible to run the algorithms below *by hand* if you have less programming experience. However, it may also be useful (or faster) to implement the algorithms.

You are also asked to prove things in this assignment (even outside the BONUS question). The questions should be simple, and by playing with algebra you should be able to solve them. I suggest you work in groups to discuss the problems, but recall, that you must write up your solutions by yourself. And remember to make your proofs clear, they need to be verifiable to be graded as correct.

As usual, it is highly recommended that you use LaTeX for this assignment. If you do not, you may lose points if your assignment is difficult to read or hard to follow. Find a sample form in this directory: `http://www.cs.utah.edu/~jeffp/teaching/latex/`

1 Hierarchical Clustering

There are several variants of hierarchical clustering; here we explore 3. The key difference is how you measure the distance $d(S_1, S_2)$ between two clusters S_1 and S_2.

Single-Link: measures the shortest link $d(S_1, S_2) = \min_{(s_1,s_2) \in S_1 \times S_2} \|s_1 - s_2\|_2$.

Complete-Link: measures the longest link $d(S_1, S_2) = \max_{(s_1,s_2) \in S_1 \times S_2} \|s_1 - s_2\|_2$.

Mean-Link: measures the distances to the means. First compute $a_1 = \frac{1}{|S_1|} \sum_{s \in S_1} s$ and $a_2 = \frac{1}{|S_2|} \sum_{s \in S_2} s$ then $d(S_1, S_2) = \|a_1 - a_2\|_2$.

A (4 points): Run all hierarchical clustering variants on data set `C1.txt` until there are $k = 3$ clusters, and report the results as sets.

Which variant did the best job, and which was the easiest to compute (think if the data was much larger)? Explain your answers.

2 Point Assignment Clustering

Point assignment clustering works by assigning every point $x \in X$ to the closest cluster centers C. Let $\phi_C : X \to C$ be this assignment map so that $\phi_C(x) = \arg\min_{c \in C} \mathbf{d}(x, c)$. All points that map to the same cluster center are in the same cluster.

Two good heuristics for these types of cluster are the Gonzalez (Algorithm 9.4.1) and k-Means++ (Algorithm 10.1.2) algorithms.

A: (4 points) Run Gonzalez and k-Means++ on data set `C2.txt` for $k = 3$. To avoid too much variation in the results, choose c_1 as the point a.

Report the centers and the subsets for Gonzalez. Report:

- the 3-center cost $\max_{x \in X} \mathbf{d}(x, \phi_C(x))$ and
- the 3-means cost $\sum_{x \in X} (\mathbf{d}(x, \phi_C(x)))^2$

For k-Means++, the algorithm is randomized, so you will need to report the variation in this algorithm. Run it several trials (at least 20) and plot the *cumulative density function* of the 3-means cost. Also report what fraction of the time the subsets are the same as the result from Gonzalez.

B: (4 points) Recall that Lloyd's algorithm for k-means clustering starts with a set of k centers C and runs as described in Algorithm 10.1.1.

- Run Lloyds Algorithm with C initially as $\{$a,b,c$\}$. Report the final subset and the 3-means cost.
- Run Lloyds Algorithm with C initially as the output of Gonzalez above. Report the final subset and the 3-means cost.
- Run Lloyds Algorithm with C initially as the output of each run of k-Means++ above. Plot a *cumulative density function* of the 3-means cost. Also report the fraction of the trials that the subsets are the same as the input.

C: (4 points) Consider a set of points $S \subset \mathbb{R}^d$ and $\mathbf{d}$ the Euclidean distance. Prove that

$$\arg\min_{p \in \mathbb{R}^d} \sum_{x \in S} (\mathbf{d}(x, p))^2 = \frac{1}{|S|} \sum_{x \in S} x.$$

Here are some steps to follow towards the proof:

1. *First prove the same results for $S \in \mathbb{R}^1$.*
2. *Expand each term $(\mathbf{d}(x, p))^2 = (x - p)^2 = x^2 + p^2 - 2xp$.*
3. *Add the above terms together and take the first derivative.*
4. *Show the results for each dimension can be solved independently (use properties of edge lengths in a right triangle).*

3 Distances in High Dimensions

We will explore a couple potentially unintuitive properties of high dimensional data.

A: (2 points) We will explore what happens to the distribution of a Gaussian distributions in high-dimensions. A d-dimensional uniform Gaussian distribution is defined:

$$G(x) = \frac{1}{(2\pi)^{d/2}} e^{-\|x\|_2^2/2}.$$

If we have two uniform random numbers $u_1, u_2 \in [0,1]$ then we can generate two independent 1-dimensional Gaussian random variables as (using the Box-Muller transform):

$$\begin{aligned} y_1 &= \sqrt{-2\ln(u_1)}\cos(2\pi u_2) \\ y_2 &= \sqrt{-2\ln(u_1)}\sin(2\pi u_2). \end{aligned}$$

A uniform Gaussian has the (*amazing!*) property that all coordinates (in any orthogonal basis) are independent of each other. Thus to generated a point $x \in \mathbb{R}^d$ from a d-dimensional Gaussian, for each coordinate i we assign it the value of an independent 1-dimensional Gaussian random variable.

Your task is to generated d-dimensional Gaussian random variables, and plot their L_2 and L_1 norms. For each $d = \{1, 2, 3, 5, 10, 50\}$ generate $t = 100$ Gaussian random variables $\{g_1, \ldots, g_t\}$, and report their average L_2 norm

$$\frac{1}{t}\sum_{i=1}^{t} \|g_i\| = \frac{1}{t}\sum_{i=1}^{t} \sqrt{\sum_{j=1}^{d} g_{i,j}^2}$$

and average L_1 norm

$$\frac{1}{t}\sum_{i=1}^{t} \|g_i\|_1 = \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{d} |g_{i,j}|.$$

B: (2 points) We will again explore the difference between different L_p distances in high dimensions. Generate uniform random variables y in $[-1,1]^d$ (each coordinate is an independent random variable in the range $[-1,1]$. (i.e. a random $u \in [0,1]$ is transformed to $x = 2u - 1$.)

For dimensions $d = \{1, 2, 3, 5, 10, 50, 100\}$ estimate the probability that the random variable y has an L_2 norm less than 1. For instance compute how many points x out of $t = 100$ random points in $[-1,1]^d$ have $\|x\|_2 < 1$.

4 BONUS

A: (2 points) Recall that the k-center problem on set X is to find a set of k centers C to minimize

$$E(X, C) = \max_{x \in X} \mathbf{d}(x, \phi_C(x)).$$

Let C^* be the optimal choice of k centers for the k-center problem, and let $E^* = E(X, C^*)$.

Prove that the Gonzalez algorithm (Algorithm 9.4.1) always finds a set of k centers C such that

$$E(X, C) \leq 2E^*.$$

B: (1 point) Find an example point set X and initial point c_1 (for $k > 1$) such that after running the Gonzalez algorithm to get center C, the cost $E(X, C) = 2E^*$.