CS548 Homework 4

Cate Dunham

April 18, 2025

Exercise 7.1

Consider a data set consisting of 2^{20} data vectors, where each vector has 32 components and each component is a 4-byte value. Suppose that vector quantization is used for compression and that 2^{16} prototype vectors are used. How many bytes of storage does that data set take before and after compression and what is the compression ratio?

Bytes before compression $= 2^{20} \times 32 \times 4 = 1,048,576 \times 128 = 134,217,728$ bytes

$$\text{Prototype vector bytes} = 2^{16} \times 32 \times 4 = 65,536 \times 128 = 8,388,608$$

We need 2 bytes per index (since the indices need to be long enough for the 2^{16} prototype vectors).

$$\text{Bytes for indices} = 2^{20} \times 2 \text{ bytes} = 2,097,152$$

$$\text{Total bytes after compression} = 8,388,608 + 2,097,152 = 10,485,760$$

$$\text{Compression ratio} = \frac{134,217,728}{10,485,760} = 12.8$$

Exercise 7.4

Given K equally sized clusters, the probability that a randomly chosen initial centroid will come from any given cluster is 1/K, but the probability that each cluster will have exactly one initial centroid is much lower. (It should be clear that having one initial centroid in each cluster is a good starting situation for K-means.) In general, if there are K clusters and each cluster has n points, then the probability, p, of selecting in a sample of size K one initial centroid from each cluster is given by Equation 7.20. (This assumes sampling with replacement.) From this formula we can calculate, for example, that the chance of having one initial centroid from each of four clusters is $4!/44 = 0.0938$.

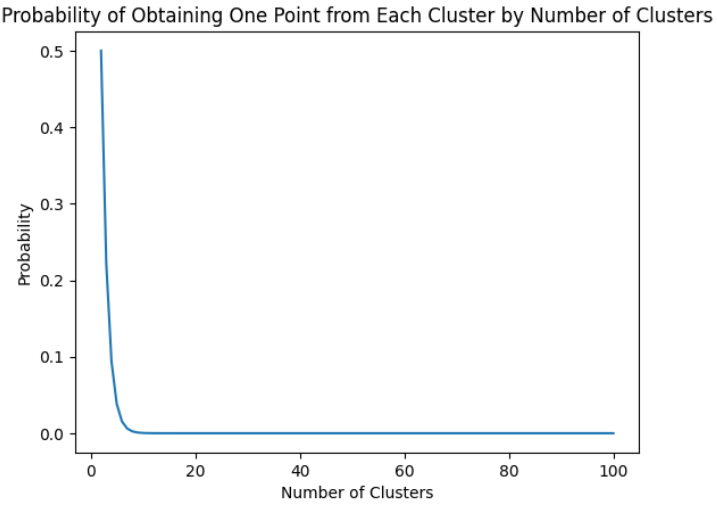

Figure 1

$$p = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K! n^K}{(Kn)^K} = \frac{K!}{K^K}$$

a. Plot the probability of obtaining one point from each cluster in a sample of size K for values of K between 2 and 100.

   See Figure 1.

b. For K clusters, K = 10, 100, and 1000, find the probability that a sample of size 2K contains at least one point from each cluster. You can use either mathematical methods or statistical simulation to determine the answer.

   We know that the chances a point comes from a specific cluster are $\frac{1}{K}$, and the chances the point does *not* come from a specific cluster are $1 - \frac{1}{K}$. For $2K$ points we can say the probability that *none* of them come from a specific cluster is $(1 - \frac{1}{K})^{2K}$ and that at least one of them *does* come from a specific cluster is $(1 - (1 - \frac{1}{K})^{2K})^K$.

$$K = 10 = (1 - (1 - \frac{1}{10})^{20})^{10} = .274 \tag{1}$$

$$= 100 = (1 - (1 - \frac{1}{100})^{200})^{100} = 5.66\text{e} - 07 \tag{2}$$

$$= 1000 = (1 - (1 - \frac{1}{1000})^{2000})^{1000} = 8.236\text{e} - 64 \tag{3}$$

Exercise 7

Suppose that for a data set

- there are m points and K clusters,
- half the points and clusters are in "more dense" regions,
- half the points and clusters are in "less dense" regions, and
- the two regions are well-separated from each other.

For the given data set, which of the following should occur in order to minimize the squared error when finding K clusters:

a. Centroids should be equally distributed between more dense and less dense regions.

b. More centroids should be allocated to the less dense region.

c. More centroids should be allocated to the denser region.

Note: Do not get distracted by special cases or bring in factors other than density. However, if you feel the true answer is different from any given above, justify your response.

Answer b. would result in the lowest squared error since more spread out points in the less dense region will result in a higher error with fewer centroids. The resulting clusters won't be accurate in this scenario, but the error would be the lowest.

Exercise 13

The Voronoi diagram for a set of K points in the plane is a partition of all the points of the plane into K regions, such that every point (of the plane) is assigned to the closest point among the K specified points. (See Figure 2.) What is the relationship between Voronoi diagrams and K-means clusters? What do Voronoi diagrams tell us about the possible shapes of K-means clusters?

Voronoi diagrams and K-means clusters are similar ideas. Both partition data into K regions/clusters based on proximity to a specified point/centroid. A Voronoi diagram can be used as a visual representation of the data and how it falls around the K centroids found by the K-means algorithm. Voronoi diagrams illustrate the linear boundaries between clusters that form as a result of distance-based clustering, and highlight K-means inability to capture non-linear boundaries.

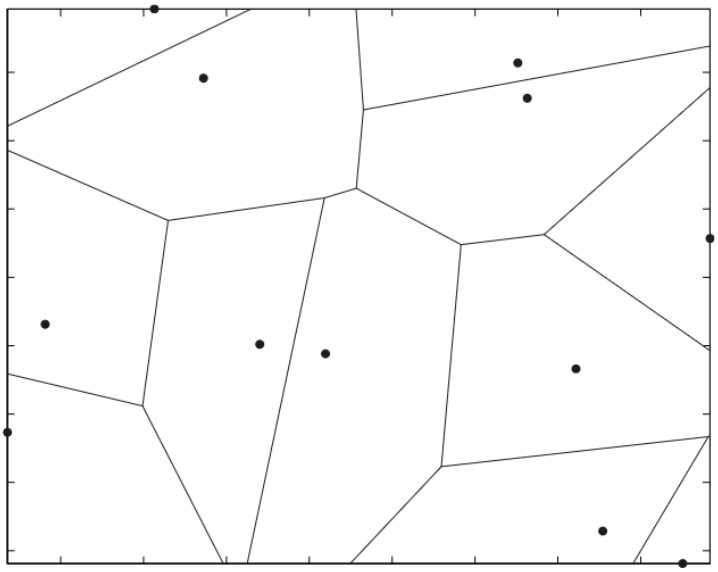

Figure 2: Voronoi diagram for Exercise 13

Exercise 14

You are given a data set with 100 records and are asked to cluster the data. You use K-means to cluster the data, but for all values of K,$1 \leq K \leq 100$, the K-means algorithm returns only one non-empty cluster. You then apply an incremental version of K-means, but obtain exactly the same result. How is this possible? How would single link or DBSCAN handle such data?

This scenario implies that the 100 records are nearly identical or are duplicate objects. Single link might still make groups if the records are nearly identical, but a look at the proximity matrix would show that the proximities were all very similar. DBSCAN would make all of the records core points and would also create one cluster.

Exercise 15

Traditional agglomerative hierarchical clustering routines merge two clusters at each step. Does it seem likely that such an approach accurately captures the (nested) cluster structure of a set of data points? If not, explain how you might postprocess the data to obtain a more accurate view of the cluster structure.

It does not seem likely that this approach will always capture the structure of the data, especially in the presence of complex relationships or different cluster shapes. During postprocessing we could split clusters with high SSE and/or merge clusters with low SSE to better reflect the true cluster structure.