Homework 1

Sloan Atkins
CSC 546
Fall '25

Part 1

1. There are two vectors, x_1 and x_2

$$x_1 = \begin{bmatrix}1\\2\end{bmatrix} \text{ and } x_2 = \begin{bmatrix}10\\18\end{bmatrix}$$

What is the distance between x_1 and x_2 ? (Please show the steps of the calculations)

(1) if the distance measure is based on L2 norm (a.k.a Euclidean norm)

(2) if the distance measure is based on L1 norm

(3) if the distance measure is based on L∞ norm (a.k.a infinity norm)

In class, we studied the customer segmentation example and tried to find the most valuable customers who have good income but low spend. There are two feature components $x = \begin{bmatrix}income\\spend\end{bmatrix}$ in this application. Assume that we use a clustering method similar to k-mean, and this method could use any type of vector norms as distance measure. Then, does the L∞ norm-based distance measure make sense for this application?

1. Based on L2 norm:

$$= \sqrt{(10-1)^2 + (18-2)^2}$$
$$= \sqrt{9^2 + 16^2}$$
$$= \sqrt{81 + 256}$$
$$= \sqrt{337}$$

2. Based on L1 norm:

$$= |10-1| + |18-2|$$
$$= |9| + |16|$$
$$= 25$$

3. Based on $L\infty$ norm:

$$= \max(|10-1|, |18-2|)$$
$$= \max(|9|, |16|)$$
$$= \max(9, 16)$$
$$= 16$$

No, L_∞ only looks at the single largest coordinate difference, ignoring the others. For customer segmentation where both income and spend matter, L_∞ can treat very different customers as equal distance if their max difference is the same. So, L_2 or L_1 would be more useful.

2. We define a scalar valued function of a vector variable

$$f(x) = x^T A x$$

Here, x is a column vector, x^T is the transpose of x, and A is a symmetric matrix

To simplify this question, let's assume x has only two elements $x = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, and $A = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$

The derivative of f with respect to x is a vector defined by $\frac{df}{dx} = \begin{bmatrix} \frac{df}{d\alpha} \\ \frac{df}{d\beta} \end{bmatrix}$

Show that $\frac{df}{dx} = 2Ax$

Hint: calculate $f(x)$, $2Ax$, $\frac{df}{d\alpha}$ and $\frac{df}{d\beta}$

$$f(x) = \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} a & c \\ c & b \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

$$f(x) = a\alpha^2 + 2c\alpha\beta + b\beta^2$$

$$\frac{df}{dx} = \frac{d}{dx}(a\alpha^2 + 2cx\beta + b\beta^2)$$

$$\frac{df}{dx} = \begin{bmatrix} \frac{d}{d\alpha}(a\alpha^2 + 2c\alpha\beta + b\beta^2) \\ \frac{d}{d\beta}(a\alpha^2 + 2c\alpha\beta + b\beta^2) \end{bmatrix}$$

$$\frac{df}{dx} = \begin{bmatrix} 2a\alpha + 2c\beta + 0 \\ 0 + 2c\alpha + 2b\beta \end{bmatrix}$$

$$\frac{df}{dx} = \begin{bmatrix} 2a\alpha + 2c\beta \\ 2c\alpha + 2b\beta \end{bmatrix}$$

$$\frac{df}{dx} = 2\begin{bmatrix} a\alpha + c\beta \\ c\alpha + b\beta \end{bmatrix}$$

$$\frac{df}{dx} = 2\begin{bmatrix} a & c \\ c & b \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

$$\frac{df}{dx} = 2Ax$$

3. Briefly describe the two key steps in one iteration of the k-means algorithm. (1 point)

Step 1: Assign each data point x_n to its nearest cluster centroid based on distance metric

$$\alpha(n) = \arg \min_{k \in (1,\dots,K)} \|x_n - c_k\|^2$$

Step 2: Move center c_k to the average location of the data points in cluster-K

$$c_k = \frac{1}{N_k} \sum_{n:\alpha(n)=k} x_n$$

4. What is the distance measure used in k-means (implemented in sk-learn)? (1 point)

l_2 norm (Euclidean) is used in K-means algorithm to measure distance

5. The k-means algorithm can always converge in a finite number of iterations. Why? (1 point)

- The loss is defined by $L = \frac{1}{N} \sum_{n=1}^{N} \|x_n - c_{\alpha}(n)\|^2$
- There are only K^n possible labelings, so L can take only finitely many values.
- Each iteration never increases L:

  Assignment: $\alpha(n) \leftarrow \arg \min_k \|x_n - c_k\|^2$

  Update: $c_k \leftarrow \arg \min_c \sum_{n:\alpha(n)=k} \|x_n - c\|^2$

- Because L will not increase (stays the same) and has finitely many values, K-means must converge in finitely many steps to a fixed labeling/centers

6. The clustering result of k-means could be random. Why? (1 point)

Because the algorithm will randomly initialize the clusters centers/centroids.

7. The minimum value of the loss function is zero for any dataset. What is the clustering result when the loss function is zero? – assuming that the dataset has millions of different samples. (1 point)

Since $L = \frac{1}{N} \sum_n \|x_n - c_{\alpha(n)}\|^2$, $L=0$ means every term is 0, so $x_n = c_{\alpha(n)}$ $\forall n$. This only happens when $K=N$: each point forms a one-point cluster and its centroid equals the point.

8. find the optimal centers by following the steps below when cluster labels are given. (5 points)

The loss function is $L = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K}\alpha_{(n,i)}||x_n - c_i||^2$ as defined in the lecture notes.

First, we calculate $\frac{\partial L}{\partial c_k}$, where k could be 1, 2, 3, …, K.

$$\frac{\partial L}{\partial c_k} = \frac{\partial\left[\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K}\alpha_{(n,i)}||x_n - c_i||^2\right]}{\partial c_k} = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial\left[\sum_{i=1}^{K}\alpha_{(n,i)}||x_n - c_i||^2\right]}{\partial c_k} = \frac{1}{N}\sum_{n=1}^{N}\boxed{\text{(A)}}\frac{\partial\left[||x_n - c_k||^2\right]}{\partial c_k}$$

$$||x_n - c_k||^2 = (x_n - c_k)^T(x_n - c_k) = x_n^T x_n + c_n^T c_n - \boxed{\text{(B)}}$$

$$\frac{\partial\left[||x_n - c_k||^2\right]}{\partial c_k} = \boxed{\text{(C)}}$$

$$\text{Thus, } \frac{\partial L}{\partial c_k} = \frac{1}{N}\sum_{n=1}^{N}\boxed{\text{(D)}}$$

Then, we set $\frac{\partial L}{\partial c_k} = 0$, and we obtain the optimal center $c_k = \frac{1}{N_k}\sum_{n=1}^{N}\alpha_{(n,k)}x_n$, where $N_k = \sum_{n=1}^{N}\boxed{\text{(E)}}$

What are (A), (B), (C), (D), and (E) in the above equations ?

Note: (E) is a variable, not a word or sentence

A: $\alpha_{(n,k)}$
B: $2x_n^T c_k$
C: $2(c_k - x_n)$
D: $2\alpha_{(n,k)}(c_k - x_n)$
E: $\alpha_{(n,k)}$