CSCE 633 :: Machine Learning :: Texas A&M University :: Spring 2022

Written Assignment 1 (W1)

Due on Tuesday, Feb 1, 11:59 PM. **Total points: 100**

Name: Jin Huang Fangqing Xia
UIN: 73000 9249 926001495

Instructions:

- This PDF has blank spaces left out after each question for you to fill in your solutions.
- You are free to either use LaTeX(use the provided *.tex* file) or handwritten them as long as they are legible.
- Refer to the submission instructions on the course webpage to upload your solutions to Canvas.
- **SHOW YOUR WORK.**

---

1 Linear Algebra and Probability Review

Question 1: Matrix Multiplication. **(10 points)**
NOTE: This is not a programming assignment, so you may NOT use programming tools to help solve this question. Show your work.

In this question you are required to perform matrix multiplication.

(1) (3 points)

$$\begin{bmatrix} 10\times0 & 10\times3 & 10\times0 & 10\times1 \\ -5\times0 & -5\times3 & -5\times0 & -5\times1 \\ 2\times0 & 2\times3 & 2\times0 & 2\times1 \\ 8\times0 & 8\times3 & 8\times0 & 8\times1 \end{bmatrix}$$

$$\begin{bmatrix} 10 \\ -5 \\ 2 \\ 8 \end{bmatrix} \begin{bmatrix} 0 & 3 & 0 & 1 \end{bmatrix} = ? \quad \begin{bmatrix} 0 & 30 & 0 & 10 \\ 0 & -15 & 0 & -5 \\ 0 & 6 & 0 & 2 \\ 0 & 24 & 0 & 8 \end{bmatrix}$$

(2) (3 points)

$$7\times(-3)+(-3)\times(-4)+1\times6+9\times0 = -21+12+6+0 = -3$$

$$\begin{bmatrix} 7 & -3 & 1 & 9 \end{bmatrix} \begin{bmatrix} -3 \\ -4 \\ 6 \\ 0 \end{bmatrix} = ? \quad -3$$

$$\begin{bmatrix} 1 & -1 & 6 & 7 \\ 9 & 0 & 8 & 1 \\ -8 & 1 & 2 & 3 \\ 10 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 0 \\ -3 \\ 3 \end{bmatrix} = \begin{bmatrix} 6-18+21 \\ 54-24+3 \\ -48-6+9 \\ 60+3 \end{bmatrix} = \begin{bmatrix} 9 \\ 33 \\ -45 \\ 63 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 & 6 & 7 \\ 9 & 0 & 8 & 1 \\ -8 & 1 & 2 & 3 \\ 10 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \\ 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 2+1+28 \\ 18+4 \\ -16-1+12 \\ 20-4+4 \end{bmatrix} = \begin{bmatrix} 31 \\ 22 \\ -5 \\ 20 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & 6 & 7 \\ 9 & 0 & 8 & 1 \\ -8 & 1 & 2 & 3 \\ 10 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 4 \\ 7 \end{bmatrix} = \begin{bmatrix} 72 \\ 39 \\ 30 \\ 11 \end{bmatrix}$$

(4 points)

$$\begin{bmatrix} 1 & -1 & 6 & 7 \\ 9 & 0 & 8 & 1 \\ -8 & 1 & 2 & 3 \\ 10 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & 2 & 0 \\ 0 & -1 & 1 \\ -3 & 0 & 4 \\ 3 & 4 & 7 \end{bmatrix} = ? \quad \begin{bmatrix} 9 & 31 & 72 \\ 33 & 22 & 39 \\ -45 & -5 & 30 \\ 63 & 20 & 11 \end{bmatrix}$$

Question 2: Vector Norms. **(8 points)**

NOTE: This is not a programming assignment, so you may NOT use programming tools to help solve this Question. Show your work.

Consider these two points in the 3-dimensional space:

$$\mathbf{a} = \begin{bmatrix} 7 \\ 0 \\ -1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 7 \\ 9 \\ -5 \end{bmatrix}$$

Calculate their distance using the following norms. $d = \begin{bmatrix} 7-7 \\ 0-9 \\ -1-(-5) \end{bmatrix} = \begin{bmatrix} 0 \\ -9 \\ 4 \end{bmatrix}$

(1) $\ell_0 = \#(i \mid d_i \neq 0) = 2$ (2 points)

(2) $\ell_1 = |0| + |-9| + |4| = 13$ (2 points)

(3) $\ell_2 = \sqrt{0^2 + (-9)^2 + 4^2} = \sqrt{97}$ (2 points)

(4) $\ell_\infty = \max_i |d_i| = 9$ (2 points)

Question 3: Probability Calculation. **(12 points)**
NOTE: This is not a programming assignment, so you may NOT use programming tools to help solve this Question. Show your work.

Consider a situation where we are rolling 2 dices where each dice has 6 faces numbered from 1 to 6. Answer the following questions:

(1) What is the size of the sample space? $\Omega = \{(i,j) \mid i=1,2,3,4,5,6; j=1,2,3,4,5,6\}$ (4 points)

size of Ω is $6 \times 6 = 36$

(2) If the event we are interested in is the sum being 11, what would be the probability of observing such an event? (4 points)

If the sum is 11, only 2 events, (5, 6) and (6, 5) are possible.

$\therefore$ probability $= 2/36 = 1/18$

(3) If the event we are interested in is the sum being 6, what would be the probability of observing such an event? (4 points)

5 possible events: (1, 5), (5, 1), (2, 4), (4, 2), (3, 3)

5/36

Question 4: Mean/Variance Calculation. **(10 points)**
NOTE: This is not a programming assignment, so you may NOT use programming tools to help solve this Question. Show your work.

Assume we have a random variable X with a Uniform probability density function. Uniform probability density is defined as:

$$f_X(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \le x \le b \\ 0 & otherwise \end{cases}$$

(1) What is the mean of X? (5 points)

$$E(X) = \int_a^b \frac{x}{b-a}\,dx = \frac{b^2-a^2}{2} \cdot \frac{1}{b-a} = \frac{a+b}{2}$$

(2) What is the standard deviation of X? (5 points)

$$E[(X-E[X])^2] = E[X^2 - 2X \cdot E[X] + E[X]^2] = E[X^2] - 2E[X] \cdot E[X] + E[X]^2$$
$$= E[X^2] - E[X]^2$$

$$E[X^2] = \int_a^b \frac{x^2}{b-a}\,dx = \frac{b^3-a^3}{3} \cdot \frac{1}{b-a} = \frac{b^2+ab+a^2}{3}$$

$$\therefore E[X^2] - E[X]^2 = \frac{b^2+ab+a^2}{3} - \frac{(a+b)^2}{4} = \frac{(b-a)^2}{12}$$

$\therefore$ standard deviation of $X = \sqrt{E[(X-E[X])^2]} = \sqrt{\frac{(b-a)^2}{12}} = \frac{b-a}{2\sqrt{3}}$

2 k-Nearest Neighbors

Question 5: Feature Scaling. **(15 points)**

(1) Why is feature scaling necessary in k-NN? (5 points)

K-NN is a distance based algorithm, which is affected by the scale of the variables.

In order to avoid bias towards variables with higher magnitudes, feature scalling is necessary.

(2) What is the potential issue with using categorical variables in k-NN? (5 points)

Categorical variable is a variable that can take on one of limited numbers of values, like gender, color. So categorical variables are usually non-ordinal, and even non-numerical.

Obviously, K-NN could not directly use categorical variables; otherwise, we would get nonsensical results.

(3) How would you pre-process categorical variables to make them suitable for use with k-NN? (5 points)

Categorical encoding, like apply One-Hot-Encoding for the non-ordinal categorical feature,

apply Label Encoding for the categorical feature.

Question 6: Curse of dimensionality. **(15 points)**

(1) Briefly explain why k-NN suffers when the data dimensionality is high? (15 points)

We consider a d-dimension feature vector $[0,1]^d$.

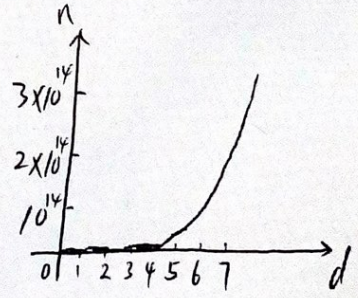
We assume accurate predictions require K neighbors within l distance, and there are n samples (uniform distribution).

Accordingly, a cube of size l^d covers $\frac{l^d}{1^d}$ of the state space and should include K/n of the uniform samples on expectancy.

$\therefore \quad n = K/l^d$

We assume $K=10$, and $l=0.1$, then we could see "Curse of dimensionality" when d increases.

For example, when $d>100$, we'll need far more data points than the number of electrons in the universe.



Question 7: Choosing the best "k". **(15 points)**

(1) Briefly explain a strategy you could use to choose the best "k" value for a given dataset for k-NN? (15 points)

(*Hint: cross-validation.*)

Let K iterate from 1 to n, and the size of the step is 1.
For each iteration, apply K-NN algorithm with cross-validation, and get the prediction error.
Then I will choose the K with the minimum error.

Question 8: Algorithmic Complexity. **(15 points)**

(1) Let n be the number of training examples in a dataset and m be the dimensionality of each data point. For simplicity, assuming $n \gg m$, the algorithmic complexity for classifying a query point using naive k-NN implementation is $\mathcal{O}(k \times n)$. Can you think of a way to make it more efficient? (15 points)

(*Hint:* $\mathcal{O}(n \log(k))$)

① Calculate all the distances between the target point and each point in the training examples. It takes $O(n)$
② Based on the distances computed in ①, build a ~~max~~ max heap of size K, it takes $O(k)$, notice we could use any k points, like the first k, and the corresponding distances.
③ Iterate over the rest of $(n-k)$ points in the training examples, and maintain the size K ~~max-heap~~ max-heap, it takes $O((n-k)\log(k))$.
④ With these K smallest distance points in the size K ~~max~~ max-heap, it takes $O(k)$ to classify the target point.

So the overall algorithmic complexity is $O(n) + O(k) + O((n-k)\log(k)) + O(k)$
$= O(n \log(k))$

Notice, since $n \gg m$, we could ignore the contribution of m to the algorithmic complexity.