Midterm 1 ECE 490/590 Spring 2024

Date: Feb 29, 2024

Instructor: Vikas Dhiman (vikas.dhiman@maine.edu)

1. Total marks are 75.
2. Total time allowed is 75 min.
3. One page cheatsheet is allowed.
4. Calculators are allowed but not needed.

1. Write your name here:

2. Write your email here:

Q1: What is the output of the following code (5 marks)

In [1]:
```
languages = ['Java', 'Python', 'JavaScript']
versions = [14, 3, 6]

result = list(zip(languages, versions))
print(result)
```

```
[('Java', 14), ('Python', 3), ('JavaScript', 6)]
```

[('Java', 14), ('Python', 3), ('JavaScript', 6)]

Q2: What is the output of the following code (5 marks)

In [2]:
```
nums = [0, 1, 2, 3, 4, 5, 6, 7]
even_squares = [x ** 2 for x in nums if (x % 2 == 0 and x <= 4)]
print(even_squares)
```

```
[0, 4, 16]
```

[0, 4, 16]

Q3: What is the output of the following code (5 marks)

In [3]:
```
X = 99
global
def f1(): local ft
    def f2(): global X
        X = 88
    f2() nonlocal X
```

```
  print(X)

f1()
```

```
99
```

99

Q4: What is the output of the following code (5 marks)

In [4]:
```
class MyNumbers:
  def __iter__(self):
    self.a = 1
    return self

  def __next__(self):
    if self.a <= 5:
      x = self.a
      self.a += 3
      return x
    else:
      raise StopIteration

myclass = MyNumbers()
myiter = iter(myclass)

for x in myiter:
  print(x)
```

[1, 4]

```
1
4
```

1

4

Q5: What is the output of the following code and why (5 marks)

In [5]:
```
import numpy as np
A = np.array([[2, 3],
              [3, 5]
              ])
B = np.array([[3, 5],
              [7, 2]
              ])
print((A * B).sum(axis=0))
```

$\left(\begin{bmatrix} 6 & 15 \\ 21 & 10 \end{bmatrix}\right).sum(axis=0)$

(2, 2)

axis=0
axis=-2

axis=1
axis=-1

(2)

```
[27 25]
```

`[27, 25]`

Because $[27, 25] = [2 \times 3 + 3 \times 7, 3 \times 5 + 5 \times 2]$

Q6: What is the output of the following code and why (5 marks)

Right to left

3 copies

In [6]:
```python
import numpy as np
row_vector = np.array([1, 3, 6]) # .shape = (3,)
column_vector = np.array([[1],
                          [-1],
                          [0]])  # .shape = (3,1)

row_vector + column_vector
```

3

Out[6]:
```
array([[2, 4, 7],
       [0, 2, 5],
       [1, 3, 6]])
```

$$\begin{bmatrix}[1, 3, 6] \\ [1, 3, 6] \\ [1, 3, 6]\end{bmatrix} + \begin{bmatrix}[1 & 1 & 1] \\ [-1 & -1 & -1] \\ [0 & 0 & 0]\end{bmatrix}$$

```
[[2, 4, 7],
[0, 2, 5],
[1, 3, 6]]
```

Because broadcasting

Q7: Convert the following scalar equation into vector form (20 marks)

$$e(a,b,c) = (z_1 - (x_1 a + y_1 b + c))^2 + (z_2 - (x_2 a + y_2 b + c))^2 + \dots \\ + (z_n - (x_n a + y_n b + c))^2$$

Your end result should contain

$$\mathbf{m} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

.

You can define other vectors and matrices as needed, included a vector of all ones like 1_n.

A7

Recall that the magnitude of a vector $\mathbf{v}$ is $\sqrt{v_1^2 + v_2^2 + \dots + v_n^n}$ has a similar form to the error function. This suggests that we can define an error vector with the signed error for each data point as it's elements

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} - \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}_{n\times 3} \begin{bmatrix} a \\ b \\ c \end{bmatrix}_{3\times 1}$$

$$\mathbf{e} = \begin{bmatrix} z_1 - (ax_1 + by_1 + c) \\ z_2 - (ax_2 + by_2 + c) \\ \vdots \\ z_n - (ax_n + by_n + c) \end{bmatrix} = \underline{Z} - \begin{bmatrix} a & b & c \end{bmatrix}_{1\times 3} \begin{pmatrix} \underline{x}_{n\times 1} \\ \underline{y}_{n\times 1} \\ \mathbb{1}_{n\times 1} \end{pmatrix}_{3n\times 1}$$

The total error is same as minimizing the square of error vector magnitude which is further same as vector product with itself.

$$e(m, c, (x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)) = \|\mathbf{e}\|^2 = \mathbf{e}^\top \mathbf{e}$$

Let us define $\mathbf{x} = [x_1; \ldots; x_n]$ to denote the vector of all x coordinates of the dataset and $\mathbf{y} = [y_1; \ldots; y_n]$ to denote y coordinates. Then the error vector is:

$$\mathbf{e} = \mathbf{z} - (\mathbf{x}a + \mathbf{y}b + \mathbf{1}_n c)$$

where $\mathbf{1}_n$ is a n-D vector of all ones. Finally, we vectorize parameters of the line $\mathbf{m} = [a; b; c]$. We will also need to horizontally concatenate $\mathbf{x}$ and $\mathbf{1}_n$. Let's call the result $\mathbf{X} = [\mathbf{x}, \mathbf{y}, \mathbf{1}_n] \in \mathbb{R}^{n\times 3}$. Now, the error vector looks like this:

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{m}$$

Expanding the error magnitude:

$$\|\mathbf{e}\|^2 = (\mathbf{y} - \mathbf{X}\mathbf{m})^\top (\mathbf{y} - \mathbf{X}\mathbf{m})$$
$$= \mathbf{y}^\top \mathbf{y} + \mathbf{m}^\top \mathbf{X}^\top \mathbf{X}\mathbf{m} - 2\mathbf{y}^\top \mathbf{X}\mathbf{m}$$

$$= (\underline{y} - X\underline{m})^T (\underline{y} - X\underline{m})$$

$$= (\underline{y}^T - \underline{m}^T X^T)(\underline{y} - X\underline{m})$$

Q8: Minimize the following function using vector derivatives (10 marks)

$$e(\mathbf{q}) = (\mathbf{y} - \mathbf{X}\mathbf{q} + \mathbf{q})^\top(\mathbf{y} - \mathbf{X}\mathbf{q})$$

Find the minimum point of the function $e(\mathbf{q})$.

Assume $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{q} \in \mathbb{R}^n$ are independent vectors and $\mathbf{X} \in \mathbb{R}^{n\times n}$ is a square matrix independent of $\mathbf{q}$. You can assume that $2\mathbf{X}^\top\mathbf{X} - \mathbf{X}^\top - \mathbf{X}$ is invertible and positive definite.

A8

$$e(\mathbf{q}) = (\mathbf{y} - \mathbf{X}\mathbf{q} + \mathbf{q})^\top(\mathbf{y} - \mathbf{X}\mathbf{q})$$

$$e(\mathbf{q}) = \mathbf{y}^\top\mathbf{y} - \mathbf{q}^\top\mathbf{X}^\top\mathbf{y} + \mathbf{q}^\top\mathbf{y} - \mathbf{y}^\top\mathbf{X}\mathbf{q} + \mathbf{q}^\top\mathbf{X}^\top\mathbf{X}\mathbf{q} - \mathbf{q}^\top\mathbf{X}\mathbf{q}$$

At the minimum point,

$$\frac{\partial e(\mathbf{q})}{\partial \mathbf{q}} = \mathbf{0}^\top$$

$$\implies \mathbf{0}^\top - \mathbf{y}^\top\mathbf{X} + \mathbf{y}^\top - \mathbf{y}^\top\mathbf{X} + 2\mathbf{q}^\top\mathbf{X}^\top\mathbf{X} - \mathbf{q}^\top(\mathbf{X}^\top + \mathbf{X}) = \mathbf{0}^\top$$

$$\implies (\mathbf{y}^\top - 2\mathbf{y}^\top\mathbf{X}) + \mathbf{q}^\top(2\mathbf{X}^\top\mathbf{X} - \mathbf{X} - \mathbf{X}^\top) = \mathbf{0}^\top$$

$$\implies (\mathbf{y} - 2\mathbf{X}^\top\mathbf{y}) + (2\mathbf{X}^\top\mathbf{X} - \mathbf{X}^\top - \mathbf{X})\mathbf{q} = \mathbf{0}$$

$$\implies \mathbf{q} = (2\mathbf{X}^\top\mathbf{X} - \mathbf{X}^\top - \mathbf{X})^{-1}(2\mathbf{X}^\top\mathbf{y} - \mathbf{y})$$

$$e(\underline{q}) = (\underline{y} - X\underline{q} + \underline{q})^T (\underline{y} - X\underline{q})$$

$$= (\underline{y}^T - \underline{q}^T X^T + \underline{q}^T)(\underline{y} - X\underline{q})$$

$$= \underbrace{\underline{y}^T\underline{y} - \underline{q}^T X^T \underline{y} - \underline{y}^T X \underline{q} + \underline{q}^T X^T X \underline{q}}_{\text{usual}} + \underbrace{\underline{q}^T \underline{y} - \underline{q}^T X \underline{q}}_{\text{new}}$$

$$\underbrace{e(\underline{q})}_{\in\mathbb{R}} = \underline{y}^T\underline{y} - \underbrace{\overbrace{2\underline{y}^T X}^{\underline{b}^T}_{1\times n} \, \underline{q}}_{} + \underbrace{\underline{q}^T X^T X \underline{q}}_{} + \underline{q}^T\underline{y} - \underline{q}^T X \underline{q}$$

$n\times n$

$1\times n$

$$\boxed{\frac{\partial}{\partial \underline{q}} e(\underline{q}) = [\quad\quad] \quad \text{Row vector}}$$

$$(\underbrace{\underline{q}^T_{1\times n} X^T \underline{y}_{n\times 1}}_{n\times n})^T$$

1×1

$$= \underline{y}^T X^{T^T} \underline{q}$$

$$\frac{\partial}{\partial \underline{q}} e(\underline{q}) = [\quad] \text{ column vector}$$

$$\nabla_{\underline{q}}^T e(\underline{q}) = \frac{\partial}{\partial \underline{q}} e(\underline{q}) = \text{row vector}$$

$$\frac{\partial}{\partial \underline{q}} e(\underline{q}) =$$

$$\frac{\partial}{\partial \underline{q}} e(\underline{q}) = \left[\frac{\partial e(\underline{q})}{\partial q_1}, \ldots, \frac{\partial e(\underline{q})}{\partial q_n}\right]$$

$$\frac{\partial}{\partial \underline{q}} e(\underline{q}) = \underline{0}^T - 2\underline{y}^T X + 2\underline{q}^T X^T X + \underline{y}^T - \underline{q}^T (X + X^T) = \underline{0}^T$$

$$\frac{\partial}{\partial \underline{x}}(\underline{b}^T \underline{x}) = \underline{b}^T$$

$$\frac{\partial}{\partial \underline{x}} \underline{x}^T A \underline{x} = \underline{x}^T (A + A^T)$$

$$\frac{\partial}{\partial \underline{x}} A\underline{x} = A$$

$$\frac{\partial}{\partial \underline{q}} \underline{q}^T \underbrace{X^T X}\underline{q}$$

$$\left.\frac{\partial}{\partial \underline{q}} e(\underline{q})\right|_{\underline{q}^*} = \underline{0}^T$$

$$(-2\underline{y}^T X + \underline{y}^T) + 2\underline{q}^T X^T X - \underline{q}^T (X + X^T) = \underline{0}^T$$

$$\underbrace{\underline{q}^T}_{1\times n} \underbrace{\overbrace{(2X^T X - X - X^T)}^{M}}_{n\times n} = \underbrace{\overbrace{(2\underline{y}^T X - \underline{y}^T)}^{b^T}}_{1\times n}$$

Multiply $(2X^T X - X - X^T)^{-1}$ on the right side

$$\boxed{\underline{q}^T_{1\times n} = (2\underline{y}^T X - \underline{y}^T)(2X^T X - X - X^T)^{-1}}$$

$$\boxed{\underline{q}_{n\times 1} = (2X^T X - X - X^T)^{-T}(2X^T \underline{y} - \underline{y})}$$

$$q^T M = \underline{b}^T$$

$$q^T \underbrace{M M^{-1}}_{I} = b^T M^{-1}$$

$$\underbrace{M^{-1}}_{n\times 1} \underbrace{q^T M}_{1\times n} = M^{-1} \underline{b}^T$$

Q9 Find the derivative (10 marks)

Let the dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \{-1, +1\}$ is the binary class label.

We encode the perceptron prediction model as

$$\hat{y}_i = f(\mathbf{x}_i; \mathbf{w}) = \mathbf{w}^\top \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix},$$

where $\mathbf{w} \in \mathbb{R}^{d+1}$.

We say that the prediction is of class -1, if $\hat{y}_i < 0$ and $+1$ if $\hat{y}_i > 0$.

The Hinge loss function is defined as

$$l(y_i, \hat{y}_i; \mathbf{w}) = \begin{cases} 0 & \text{if } y_i\hat{y}_i > 0 \\ -y_i \mathbf{w}^\top \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} & \text{if } y_i\hat{y}_i \leq 0 \end{cases}$$

The total loss over the entire dataset is defined as

$$L(\mathcal{D}, \mathbf{w}) = \frac{1}{n} \sum_{(x_i, y_i) \in \mathcal{D}} l(y_i, \hat{y}_i; \mathbf{w}) = \frac{1}{n} \sum_{(x_i, y_i) \in \mathcal{D}} \begin{cases} 0 & \text{if } y_i\hat{y}_i > 0 \\ -y_i \mathbf{w}^\top \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} & \text{if } y_i\hat{y}_i \leq 0 \end{cases}$$

Find the derivative (gradient) of the function $L(\mathcal{D}, \mathbf{w})$ with respect to $\mathbf{w}$

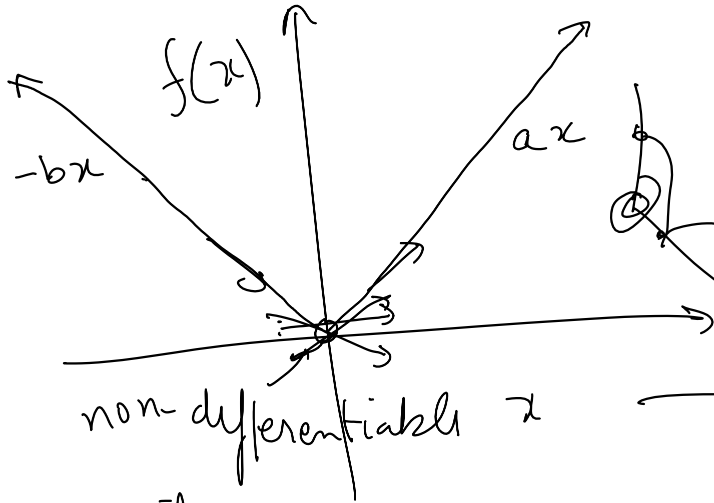
A9

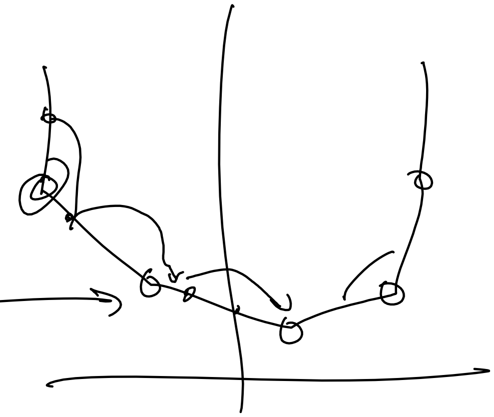
$$\nabla_{\mathbf{w}} L(\mathcal{D}, \mathbf{w}) = \nabla_{\mathbf{w}} \frac{1}{n} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} l(y_i, \hat{y}_i; \mathbf{w}) = \frac{1}{n} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} \nabla_{\mathbf{w}} l(y_i, \hat{y}_i; \mathbf{w})$$

$$\boxed{\nabla_{\mathbf{w}} L(\mathcal{D}, \mathbf{w}) = \frac{1}{n} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} \begin{cases} 0 & \text{if } y_i\hat{y}_i > 0 \\ -y_i \underbrace{\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}}_{\text{column vector}} & \text{if } y_i\hat{y}_i \leq 0 \end{cases}}$$

$$\nabla_{\underline{w}}^{T} = \frac{\partial}{\partial \underline{w}} \underline{w}^T \begin{bmatrix} \underline{x}_i \\ 1 \end{bmatrix} = \begin{bmatrix} \underline{x}_i^T & 1 \end{bmatrix}$$

$$\nabla_{w} = \begin{bmatrix} \underline{x}_i \\ 1 \end{bmatrix}$$





$$f(x) = \begin{cases} ax & \text{if } x > 0 \\ -bx & \text{if } x \leq 0 \end{cases}$$

$$\frac{d}{dx} f(x) = ? = \begin{cases} a & \text{if } x > 0 \\ -b & \text{if } x \leq 0 \end{cases}$$

sub gradient

Q10: Relationship between Hessian matrix and minimum; maximum and saddle points (5 marks)

Suppose you found an extreme point $\mathbf{x}^*$ of a function $f(\mathbf{x})$, where the gradient is zero

$$\nabla_{\mathbf{x}} f(\mathbf{x})|_{\mathbf{x}^*} = \mathbf{0}$$

You are given the Hessian matrix $\mathcal{H} f(\mathbf{x})|_{\mathbf{x}^*}$ at the extreme point. How would you find out if the extrement point $\mathbf{x}^*$ is a minimum, maximum or a saddle point?

A10:

1. If all the eigen values of the Hessian matrix are positive, then $\mathbf{x}^*$ is a miniumum.
2. If all the eigen values of the Hessian matrix are negative, then $\mathbf{x}^*$ is a maximum.
3. If some of the the eigen values of the Hessian matrix are positive and others are negative, then $\mathbf{x}^*$ is a saddle point.