MATH4570 Midterm 2

David Zhuang

22 November 2021

1 Problem 1

1.1 Part 1

$v_1 = u_1$
$v_2 = u_2 - \text{proj}_{v_1} u_2 = u_2 - \frac{\langle u_2, v_1 \rangle}{\langle v_1, v_1 \rangle} v_1$
$= u_2 - \frac{1(0)+2(1)+1(2)+0(1)+0(0)}{1^2+2^2+1^2+0^2+0^2} v_1$
$= u_2 - \frac{4}{6} v_1$

Orthogonal basis =

$$\left\{ u_1, \begin{bmatrix} -\frac{2}{3} \\ -\frac{1}{3} \\ \frac{4}{3} \\ 1 \\ 0 \end{bmatrix} \right\}$$

For orthonormal, divide each vector by its norm = $\sqrt{\langle b, b \rangle}$:

$$\left\{ \begin{bmatrix} \frac{\sqrt{6}}{6} \\ \frac{2\sqrt{6}}{6} \\ \frac{\sqrt{6}}{6} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -\frac{2\sqrt{30}}{30} \\ -\frac{\sqrt{30}}{30} \\ \frac{4\sqrt{30}}{30} \\ \frac{\sqrt{30}}{10} \\ 0 \end{bmatrix} \right\}$$

1.2 Part 2

In:

```
M = sympy.Matrix([[1,  0],
                  [2,  1],
                  [1,  2],
                  [0,  1],
                  [0,  0]]).T
M.rref()
```

Out:

```
(Matrix([
  [1, 0, -3, -2, 0],
  [0, 1,  2,  1, 0]]),
  (0, 1))
```

$a_1 = 3a_3 + 2a_4$
$a_2 = -2a_3 + -a_4$

$$\left\{\begin{bmatrix} 3 \\ -2 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right\}$$

1.3 Part 3

$||\vec{x} - \text{proj}_{u_1}\vec{x} - \text{proj}_{u_2}\vec{x}||$

2 Problem 2

2.1 Part 1

$v_1 = 1$
$v_2 = 2x + 3x^2 - \frac{\langle 2x+3x^2, 1\rangle}{\langle 1,1\rangle}$
$= 2x + 3x^2 - \frac{\pi^2+\pi^3}{\pi} = 2x + 3x^2 - \pi - \pi^2$

$$\{1, 2x + 3x^2 - \pi - \pi^2\}$$

2.2 Part 2

$\langle \sin x - a - bx - cx^2, 1\rangle = 0$
$\langle \sin x - a - bx - cx^2, x\rangle = 0$
$\langle \sin x - a - bx - cx^2, x^2\rangle = 0$
Integrating we get:
$2 - \pi a - \frac{\pi^2}{2}b - \frac{\pi^3}{3}c = 0$
$\pi - \frac{\pi^2}{2}a - \frac{\pi^3}{3}b - \frac{\pi^4}{4}c = 0$
$\pi^2 - 4 - \frac{\pi^3}{3}a - \frac{\pi^4}{4}b - \frac{\pi^5}{5}c = 0$
Solving we get:
$a = 12(\pi^2 - 10)/\pi^3$
$b = 60(-\pi^2 + 12)/\pi^4$
$c = 60(\pi^2 - 12)/\pi^5$

$$\frac{12(\pi^2 - 10)}{\pi^3} + \frac{60(-\pi^2 + 12)}{\pi^4}x + \frac{60(\pi^2 - 12)}{\pi^5}x^2$$

3 Problem 3

3.1 Part 1

$\nabla||X\vec{\theta} - \vec{b}||^2 = 2X^T X\vec{\theta} - 2X^T\vec{b}$
$\nabla J(\vec{\theta}) = \nabla\frac{1}{n}||X\vec{\theta} - \vec{b}||^2 = \frac{1}{n}(2X^T X\vec{\theta} - 2X^T\vec{b})$

3.2 Part 2

$H(||X\vec{\theta} - \vec{b}||^2) = 2X^T X$
$H(J(\vec{\theta})) = H(\frac{1}{n}||X\vec{\theta} - \vec{b}||^2) = 2(\frac{1}{n}X^T)(\frac{1}{n}X) = \frac{2}{n^2}X^T X$

3.3 Part 3

$$\vec{\theta}^{\text{next}} = \vec{\theta} - \alpha\nabla J(\vec{\theta})$$

3.4 Part 4

$$\vec{\theta}^{\text{next}} = \vec{\theta} - H^{-1}\nabla J(\vec{\theta})$$

where H is the Hessian matrix of $J(\vec{\theta})$

4 Problem 4

4.1 Part 1

In:

```
x= np.array([0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4] )
y= np.array([5.1, 6.4, 6.1, 8.2, 9.5, 8.6, 12, 14.8])
A = np.array([[z, 1] for z in x])
b = np.array([[z] for z in y])
ata = np.matmul(np.transpose(A), A)
atb = np.matmul(np.transpose(A), b)
np.matmul(np.linalg.inv(ata), atb)
```

Out:

```
array([[6.23214286],
       [4.475     ]])
```

$\theta_0 = 4.475$
$\theta_1 = 6.23214286$

4.2 Part 2

In:

```
x= np.array([0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4] )
y= np.array([5.1, 6.4, 6.1, 8.2, 9.5, 8.6, 12, 14.8])
A = np.array([[z*z, z, 1] for z in x])
b = np.array([[z] for z in y])
ata = np.matmul(np.transpose(A), A)
atb = np.matmul(np.transpose(A), b)
np.matmul(np.linalg.inv(ata), atb)
```

Out:

```
array([[3.73511905],
       [1.00297619],
       [5.52083333]])
```

$\theta_0 = 5.52083333$
$\theta_1 = 1.00297619$
$\theta_2 = 3.73511905$

4.3 Part 3

Linear:
In:

```
sum([(6.23214286*x[z] + 4.475 - y[z])**2 for z in range(len(x))])
```

Out:

```
8.608214285714295
```

Quadratic:
In:

```
sum([(x[z]**2*3.73511905 + 1.00297619*x[z] + 5.52083333 - y[z])**2
        for z in range(len(x))])
```

Out:

```
4.858154761904766
```

The quadratic fit has a smaller RSS.

5 Problem 5

5.1 Part 1

In:

```
import pandas as pd
import scipy
from sklearn.linear_model import LogisticRegression
X1 = np.array([0.2, 0.6, 2, 2.6, 3.1, 3.8])
X2 = np.array([3.4, 1.8, 2, 2.7, 3.5, 1.5])
Z1 = np.array([ -0.7, -2.1, -2.5, -3, -3.9])
Z2 = np.array([ -2.9, -2.8, -1.3, -2, -1.5])
data = {'X1': list(X1)+list(Z1),
        'X2': list(X2)+list(Z2),
        'label': [0]*len(X1) + [1]*len(Z1)}
data = pd.DataFrame(data = data)
X = data.drop(columns=['label'])
y = data['label']
#X = (X - X.mean()) / X.std()
lr = LogisticRegression()
lr.fit(X.values, y)
lr.coef_
```

Out:

```
array([[-0.68393719, -0.86658715]])
```

$$\theta = \begin{bmatrix} -0.68393719 \\ -0.86658715 \end{bmatrix}$$

5.2 Part 2

In:

```
m = -lr.coef_[0,0]/lr.coef_[0,1]
b = -lr.intercept_[0]/lr.coef_[0,1]
print('y = %fx + %f' % (m, b))
```

Out:

```
y = -0.789231x + -0.258191
```

5.3 Part 3

In:

```
lr.predict_proba([[0, 0]])
```

Out:

```
array([[0.55570396, 0.44429604]])
```

$P(y = 0|\vec{x}) = 0.55570396$
$P(y = 1|\vec{x}) = 0.44429604$

5.4 Part 4

The predicted label is 0