Cmpsc 448 HW1

Theory Problems

Problem #1

$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

a. $A \times B = \begin{bmatrix} 1\cdot1+2\cdot3 & 1\cdot2+2\cdot4 \\ 2\cdot1+4\cdot3 & 2\cdot2+4\cdot4 \end{bmatrix} = \begin{bmatrix} 7 & 10 \\ 14 & 20 \end{bmatrix}$

b. $\underset{1\times2}{x^T} \underset{2\times2}{A} \underset{2\times1}{x} = [2\ 1] \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} =$

$[2\cdot1+1\cdot2 \quad 2\cdot2+1\cdot4] \begin{bmatrix} 2 \\ 1 \end{bmatrix} = [4\ 8] \begin{bmatrix} 2 \\ 1 \end{bmatrix} = [4\cdot2+8] = 16$

c. $x^T x = [2\ 1] \begin{bmatrix} 2 \\ 1 \end{bmatrix} = 2\cdot2+1\cdot1 = 5$

d. $x x^T = \begin{bmatrix} 2 \\ 1 \end{bmatrix} [2\ 1] = \begin{bmatrix} 2\cdot2 & 2\cdot1 \\ 1\cdot2 & 1\cdot1 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}$

e. $\text{Proj}_A x = A(A^TA)^{-1}A^Tx = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \left(\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

Since the subspace of A is the same as the span of A we use span instead $= \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \left(\begin{bmatrix} 5 & 10 \\ 10 & 20 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

$= \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} (\text{not invertible})$

$\text{Span}(A) = \{ \begin{bmatrix} 1 \\ 2 \end{bmatrix} \}$ so we project onto the first column (a_1) of A instead

$\text{proj}_{a_1} x = \frac{x \cdot a_1}{\|a_1\|^2} a_1 = \frac{2\cdot1+1\cdot2}{2^2+1^2} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \frac{4}{5} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4/5 \\ 8/5 \end{bmatrix}$

f. $f(z) = z^T A z = \underset{1\times2}{[z_1\ z_2]} \underset{2\times2}{\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}} \underset{2\times1}{\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}} = \begin{bmatrix} z_1+2z_2 \\ 2z_1+4z_2 \end{bmatrix}^T \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = z_1^2 + 2z_1z_2 + 2z_1z_2 + 4z_2^2$

$= z_1^2 + 4z_1z_2 + 4z_2^2 \quad \nabla f = (2z_1+4z_2,\ 4z_1+8z_2)$

$f(z) = \begin{bmatrix} 2z_1+4z_2 \\ 4z_1+8z_2 \end{bmatrix} = 2 \begin{bmatrix} z_1+2z_2 \\ 2z_1+4z_2 \end{bmatrix} = 2 \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} z$

g. Use 1st deriv vector from above $\nabla f(z) = \begin{bmatrix} 2z_1+4z_2 \\ 4z_1+8z_2 \end{bmatrix}$

Since the 1st deriv is $2Az$, the second deriv will just be $2A$

so $\nabla^2 f(z) = \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix}$

h. $\|z\|_2 = 1$ find eigenvalues/eigenvectors

$\text{Det}(A - \lambda I) = \text{Det}\left(\begin{bmatrix} 1-\lambda & 2 \\ 2 & 4-\lambda \end{bmatrix} \right) = 4 - 5\lambda + \lambda^2 - 4 = \lambda^2 - 5\lambda \quad \lambda = 0, 5$

$A - 5I = \begin{bmatrix} -4 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 = \begin{bmatrix} -4x_1+2x_2 \\ 2x_1-x_2 \end{bmatrix} \quad 2x_1 = x_2$ eigenvector $= \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

but doesn't fit constraint, so normalize $\frac{1}{\sqrt{2^2+1^2}} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}$

Problem #2

a. $X \sim N(1,2)$ $E(X) = 1$ $E(X^2) - E(X)^2 = Var(X) = 2$

b. $X_1 X_2 \ldots X_n$ are independent $X_i \sim Bern(p)$

$\sum X_i \sim$ Binomial distribution with parameters n and p

c. $X_i \sim Bern(p)$

$$P(X_i = x_i | p) = p^{x_i}(1-p)^{1-x_i}$$

$$L = \prod_{i=1}^{14} P(X_i = x_i | p) = \prod_{i=1}^{14} p^{x_i}(1-p)^{1-x_i}$$

$$L = p^{\sum x_i}(1-p)^{n-\sum x_i} \qquad \ell(\theta) = \sum x_i \log(p) + (n - \sum x_i)\log(1-p) = 0$$

$$\frac{\partial}{\partial p} \quad \sum x_i \log(p) + n\log(1-p) - \sum x_i \log(1-p) = 0$$

$$\frac{\sum x_i}{p} + \frac{-n}{1-p} + \frac{\sum x_i}{1-p} = 0 \qquad \sum x_i (1-p) = (n - \sum x_i)p$$

$$\sum x_i = np \qquad p = \frac{\sum x_i}{n}$$

$$p = \frac{8}{14} = \frac{4}{7}$$

Problem #3

$$\begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 3 \\ 1 & 1 & 2 \end{bmatrix}$$ reduce to row echelon form

$$R_2 \leftarrow R_2 - R_1 \begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 2 \\ 1 & 1 & 2 \end{bmatrix} \qquad R_3 \leftarrow R_3 - R_1 \begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 2 \\ 0 & -1 & 1 \end{bmatrix}$$

$$R_3 \leftarrow R_3 - \tfrac{1}{2}R_2 \begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

the rank of the matrix is 2 because it is the number of non-zero rows

Problem #5

$$f(x) = \ln(1+e^{-2x}) \qquad f'(x) = \frac{1}{1+e^{-2x}} \frac{\partial}{\partial x}(1+e^{-2x}) = \frac{1}{1+e^{-2x}}(e^{-2x})\frac{\partial}{\partial x}(-2x)$$

$$= \frac{1}{1+e^{-2x}}(e^{-2x})(-2) = \frac{-2e^{-2x}}{1+e^{-2x}} = -\frac{2}{e^{2x}+1}$$

Problem #6 A is symmetric matrix so it is equal to A^T

$\frac{\partial}{\partial x}(x^T A x) = 2Ax$ as seen in problem 1f so the $\frac{\partial}{\partial x}(\frac{1}{2}x^T A x) = Ax$ and since $b^T x$ is a linear term its gradient is just b in this scenario the product is the same as differentiating with respect to x because there is only one vector variable so $\nabla(\frac{1}{2}x^T A x + b^T x) = Ax + b$

Problem #7

a. To find the maximum set the derivative to 0

$g'(x) = \frac{3}{2}x^2 - x - 6 = 0$

$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ use quadratic formula

$x = \frac{1 \pm \sqrt{1-4(\frac{3}{2})(-6)}}{2(\frac{3}{2})} = \frac{1 \pm \sqrt{1+36}}{3} = \frac{1 \pm \sqrt{37}}{3}$

now plug values of x into $g(x)$

$g(\frac{1+\sqrt{37}}{3}) = 3.1273$ $g(\frac{1-\sqrt{37}}{3}) = 19.7986$

check endpoints:

$g(-4) = -\frac{5}{2}$ $g(4) = \frac{27}{2}$

the value $x = \frac{1-\sqrt{37}}{3}$ maximizes $g(x)$

b. $g(x) = \frac{1}{2}x^3 \frac{1}{2}x^2 - 6x + \frac{27}{2}$

$\int_0^1 g(x)dx = \int_0^1 \frac{1}{2}x^3 - \frac{1}{2}x^2 - 6x + \frac{27}{2}dx = \frac{1}{2}\int_0^1 x^3 - x^2 - 12x + 27\,dx$

$= \frac{1}{2}\left[\frac{1}{4}x^4 - \frac{1}{3}x^3 - 6x^2 + 27x\right|_0^1 = \frac{1}{2}\left(\left[\frac{1}{4} - \frac{1}{3} - 6 + 27\right] - [0]\right) = \frac{1}{2}\left(\frac{251}{12}\right) = \frac{251}{24}$

$\boxed{\frac{251}{24}}$

Problem #4

```
>>> import numpy as np
>>> x = np.array([
...               [3,1,1],
...               [2,4,2],
...               [-1,-1,1]])
>>> x
array([[ 3,  1,  1],
       [ 2,  4,  2],
       [-1, -1,  1]])
>>> eigenvalues, eigenvectors = np.linalg.eig(x)
>>> eigenvalues
array([4.+0.0000000e+00j, 2.+4.4408921e-16j, 2.-4.4408921e-16j])
>>> eigenvectors
array([[-0.40824829+0.j        ,  0.42295663-0.35410429j,
         0.42295663+0.35410429j],
       [-0.81649658+0.j        ,  0.27886944+0.35410429j,
         0.27886944-0.35410429j],
       [ 0.40824829+0.j        , -0.70182607+0.j        ,
        -0.70182607-0.j        ]])
>>>
```

1. How many men and women (sex feature) are represented in this dataset?

[44]:
```
# You answer (code + results)
gender_counts = data['sex'].value_counts()
print(gender_counts)
```

```
sex
Male      21790
Female    10771
Name: count, dtype: int64
```

2. What is the average age (age feature) of women?

[54]:
```
# You answer (code + results)
avg_woman_age = data[data["sex"].str.strip() == "Female"]["age"].mean()
print("average age: ",avg_woman_age)
```

```
average age:  36.85823043357163
```

3. What is the percentage of German citizens (native-country feature)?

[55]:
```
# You answer (code + results)
count_german = data[data["native-country"].str.strip() == "Germany"].shape[0]
total_count = data.shape[0]
print((count_german/total_count)*100, "% are Germans")
```

```
0.42074874850281013 % are Germans
```

4. What are the mean and standard deviation of age for those who earn more than 50K per year (salary feature) and those who earn less than 50K per year?

[61]:
```
# You answer (code + results)
greater_than = data[data["salary"].str.strip() == ">50K"]["age"]
less_than = data[data["salary"].str.strip() == "<=50K"]["age"]
printable = f"Earn more than 50K] Mean: {greater_than.mean()} Standard Deviation: {greater_than.std()}\nEarn less than 50K] Mean: {less_than.mean()} Stan

print(printable)
```

```
Earn more than 50K] Mean: 44.24984058155847 Standard Deviation: 10.519027719851826
Earn less than 50K] Mean: 36.78373786407767 Standard Deviation: 14.02008849082488
```

5. Is it true that people who earn more than 50K have at least high school education? (education – Bachelors, Prof-school, Assoc-acdm, Assoc-voc, Masters or Doctorate feature)

[75]:
```
# You answer (code + results)
# at least a highschool education implies that person is HS-Grad
education = ['Bachelors','HS-grad','Masters','Some-college','Assoc-acdm','Assoc-voc','Doctorate' 'Prof-school']
at_least_hs = data[(data["salary"].str.strip() == ">50K")]["education"].str.strip().isin(education)

print(at_least_hs.value_counts())
print("\nThis dataframe above shows that there are 973 people that make over 50K that have not completed Highschool, meaning that the statement is false"
```

```
education
True     6868
False     973
Name: count, dtype: int64

This dataframe above shows that there are 973 people that make over 50K that have not completed Highschool, meaning that the statement is false
```

6. Display age statistics for each race (race feature) and each gender (sex feature).

Hint: Use `groupby()` and `describe()` functions of DataFrame. Find the maximum age of men of Amer-Indian-Eskimo race.

[78]:
```
# You answer (code + results)
age_stats = data.groupby(['race', 'sex'])['age'].describe()
print(age_stats)

print("\nFrom the table above, we can see that the max age for Amer-Indian-Eskimo men is 82 years old")
```

```
                          count       mean        std   min   25%   50%  \
race               sex
Amer-Indian-Eskimo Female    119.0  37.117647  13.114991  17.0  27.0  36.0
                   Male      192.0  37.208333  12.049563  17.0  28.0  35.0
Asian-Pac-Islander Female    346.0  35.089595  12.300845  17.0  25.0  33.0
                   Male      693.0  39.073593  12.883944  18.0  29.0  37.0
Black              Female   1555.0  37.854019  12.637197  17.0  28.0  37.0
                   Male     1569.0  37.682600  12.882612  17.0  27.0  36.0
Other              Female    109.0  31.678899  11.631599  17.0  23.0  29.0
                   Male      162.0  34.654321  11.355531  17.0  26.0  32.0
White              Female   8642.0  36.811618  14.329093  17.0  25.0  35.0
                   Male    19174.0  39.652498  13.436029  17.0  29.0  38.0

                            75%   max
race               sex
Amer-Indian-Eskimo Female  46.00  80.0
                   Male    45.00  82.0
Asian-Pac-Islander Female  43.75  75.0
                   Male    46.00  90.0
Black              Female  46.00  90.0
                   Male    46.00  90.0
Other              Female  39.00  74.0
                   Male    42.00  77.0
White              Female  46.00  90.0
                   Male    49.00  90.0

From the table above, we can see that the max age for Amer-Indian-Eskimo men is 82 years old
```

7. What is the maximum number of hours a person works per week (hours-per-week feature)? How many people work such a number of hours, and what is the percentage of those who earn a lot (>50K) among them?

[88]:
```
# You answer (code + results)
max_hours = data['hours-per-week'].max()
print(f"The maximum number of hours a person works a week is {max_hours}")

max_workers = data[data['hours-per-week'] == max_hours]
num_max_workers = max_workers.shape[0]
print(f"There are {num_max_workers} workers that work {max_hours} hours a week")

num_greater = max_workers[max_workers["salary"].str.strip() == ">50K"].shape[0]
print(f"{(num_greater/num_max_workers)*100}% of workers that work {max_hours} hours a week make more than $50K a year")
```

```
The maximum number of hours a person works a week is 99
There are 85 workers that work 99 hours a week
29.411764705882355% of workers that work 99 hours a week make more than $50K a year
```

8. Count the average time of work (hours-per-week) for those who earn a little and a lot (salary) for each country (native-country). What will these be for Japan?

[95]:
```
# You answer (code + results)
data["native-country"] = data['native-country'].str.strip() #remove spaces around country names
average_hours = data.groupby(["native-country","salary"])["hours-per-week"].mean()

print(average_hours.loc['Japan'])
```

```
salary
<=50K    41.000000
>50K     47.958333
Name: hours-per-week, dtype: float64
```