Sheet 7.

Q.1 Let $X_1, \ldots X_n$ be independent r.v's with densities

$$f_{X_i}(x|\theta) = \begin{cases} \exp(i\theta - x) & x \geq i\theta \\ 0 & x < i\theta \end{cases} \quad -(1)$$

Proove that $T((X_1, \ldots X_n)) = \min_i (X_i / i)$ is a suff. statistic for θ.

i) Computing the joint pdf : $f(x_1, \ldots x_n | \theta)$.

$$\text{Joint pdf} = 0 \quad \text{if } x < i\theta$$
$$= \prod_{i=1}^{n} \exp(i\theta - x) \quad \text{if } x \geq i\theta$$
$$= \exp\left(\theta \sum_{i=1}^{n} i - \sum_{i=1}^{n} x_i\right) \quad \because x_i = x$$
$$f(x_1, \ldots x_n | \theta) = \exp\left(\theta \cdot \frac{n(n+1)}{2} - \sum_{i=1}^{n} x_i\right)$$

ii) CDF $P(T(X_1, \ldots X_n) \leq t) = 1 - P(T(X_1, \ldots X_n) > t)$

$$= 1 - P\left(\min_i \left(\frac{X_i}{i}\right) > t\right)$$

$$= 1 - P(X_i > it) \quad \forall i \in \{1, \ldots n\}$$

$\because$ given.

By using independence & (1)

$$= 1 - \prod_{i=1}^{n} \exp(it - x_i)$$

$$= 1 - \exp\left(t \cdot \frac{(n)(n+1)}{2} - \sum_{i=1}^{n} x_i\right)$$

deriving CDF wrt t

$$\text{PDF} = 0 - \frac{n(n+1)}{2} \exp\left(t \cdot \frac{n(n+1)}{2} - \sum_{i=1}^{n} x_i\right)$$

$$f_T(T(x_1, \ldots x_n) | \theta) = \text{-11-}$$

Taking the ratio:

$$f(x_1, \dots x_n | \theta) / f_T(T(x_1, \dots x_n) | \theta)$$

$$= \frac{\exp\left(\theta \frac{(n)(n+1)}{2} - \sum_{i=1}^{n} x_i\right)}{-\frac{n(n+1)}{2} \exp\left(t \frac{(n)(n+1)}{2} - \sum_{i=1}^{n} x_i\right)}$$

As the pdf $f_T(T(x_1, \dots x_n) | \theta)$ is wrt θ
i.e. $t = \theta$.

The ratio / fraction $= \frac{2}{-n(n+1)}$ which is independent of θ for $x \geq i\theta$

& also for $x < i\theta$, since the ratio is 0 ~~if $x < i\theta$~~

Ex 2: $X_1 \dots X_n$ are i.i.d distributed according to:

$$f(x|\theta) = \begin{cases} \exp(\theta - x) & \text{if } x \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

Likelihood function.

i) For $x \geq \theta$

$$\mathcal{L}(\theta | x_1, \dots x_n) = \prod_{i=1}^{n} \exp(\theta - x)$$

$$= \exp\left(n\theta - \sum_{i=1}^{n} x\right)$$

$\because \bar{x}_n = \frac{1}{n} \sum_{i=1}^{n} x$

$\therefore \mathcal{L}(\theta | x_1, \dots x_n) = \exp(n(\theta - \bar{x}_n))$

ii) For $x < \theta$

$$\mathcal{L}(\theta | x_1, \dots x_n) = 0$$

Page No.
Date

$\because f(x|\theta) \neq 0$ only if $x \geq \theta$ i.e. $\theta \leq \min(x_1, \dots x_n)$

$\therefore$ we can write

$$\ell(\theta|x) = \exp(n(\theta - \bar{x}_n)) \qquad \theta \leq \min(x_1, \dots x_n)$$
$$= 0 \qquad \theta > \min(x_1, \dots x_n)$$

Since $\ell(\theta|x)$ is increasing func^n on the interval when its non 0

$$\sup_{-\infty < \theta < \infty} \ell(\theta|x) = \exp(n(\min(x_i) - \bar{x}_n))$$
i.e $\theta \in \mathbb{R}$.

whereas

$$\sup_{\theta \in \Theta_0} \ell(\theta|x) = \exp(n(\min(x_i) - \bar{x}_n)) \text{ if } \theta_0 \geq \min(x_i)$$

$$\sup_{\theta \in \Theta_0} \ell(\theta|x) = \exp(n(\theta_0 - \bar{x}_n)) \text{ if } \theta_0 < \min(x_i)$$

Thus

$$\lambda(x) = \frac{\sup_{\Theta_0} \mathcal{L}(\theta|x)}{\sup_{\Theta} \mathcal{L}(\theta|x)} = 1 \text{ if } \theta_0 \geq \min(x_1 \dots x_n)$$

$$= \exp(n(\theta_0 - \cancel{\bar{x}_n} - \min(x_i) + \cancel{\bar{x}_n}))$$
$$= \exp(n(\theta_0 - \min(x_i))) \text{ if } \theta_0 < \min(x_i)$$

Sheet_7_Task_2_Code

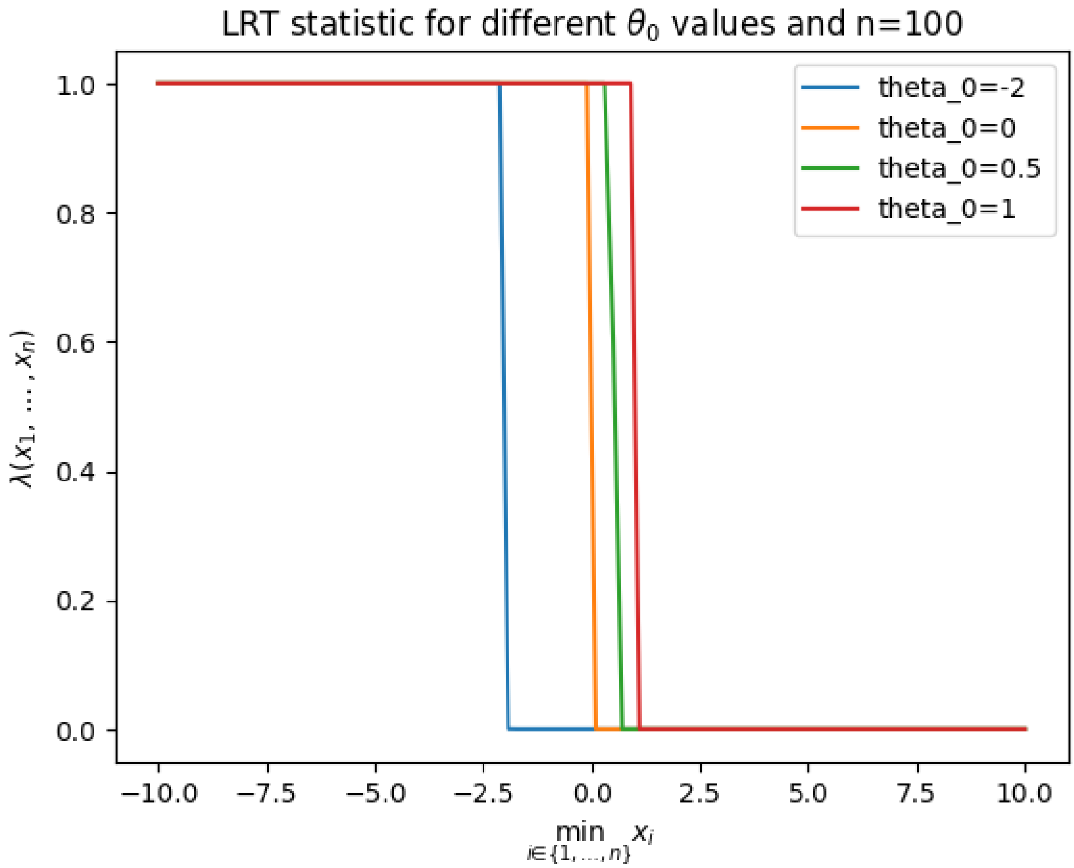
December 15, 2023

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
```

[2]:
```
def lrt(t0, min_x, n):
    return np.exp(n * (t0 - min_x) )if t0 <= min_x else 1

def test_t0(t0, minima, n):
    res = np.zeros(len(minima))

    for i, min_x in enumerate(minima):
        res[i] = lrt(t0, min_x, n)

    return  res
```

[3]:
```
n = 100
minima_x = np.linspace(-10,10,n)
theta_0_vals = [-2, 0, 0.5, 1]
```

[4]:
```
for t0 in theta_0_vals:
    res=test_t0(t0, minima_x, n)
    plt.plot(minima_x, res, label=f'theta_{0}={t0}')
    plt.xlabel('$\min_{i\in\{1,\dots,n\}}{x_i}$')
    plt.ylabel('$\lambda(x_1,\dots,x_n)$')
    plt.legend(loc='best')
    plt.title(f'LRT statistic for different $\\theta_{0}$ values and n={n}')

plt.show()
```


LRT statistic for different θ_0 values and n=100
theta_0=-2
theta_0=0
theta_0=0.5
theta_0=1
$\lambda(x_1, \dots, x_n)$
1.0
0.8
0.6
0.4
0.2
0.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
$\min_{i \in \{1,\dots,n\}} x_i$

Task 3

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [82]:
```
def read_data():
  X = np.array(pd.read_csv('X.txt', header=None))
  Y = np.array(pd.read_csv('Y.txt', header=None))
  ones = np.ones((X.shape[0], 1))
  X = np.c_[ones, X]
  return X, Y
```

a) n=4

In [98]:
```
def task_a(X: np.ndarray, Y: np.ndarray, n: int=4) -> None:
  X = X[:n]
  Y = Y[:n]
  xtx = np.linalg.inv(np.matmul(X.T, X))
  ls = np.matmul(xtx, np.matmul(X.T, Y))
  return (xtx, ls.flatten())

X, Y = read_data()
xtx, betas = task_a(X, Y, n=4)
print(f'XTX^(-1): \n{xtx}')
print(f'Least-Squares for n=4:\nBeta 0: {betas[0]}\nBeta 1: {betas[1]}\nBeta 2:
```

```
XTX^(-1):
[[1.86905258e+09 1.73429460e+08 7.85725234e+07 3.79496696e+08]
 [1.73429460e+08 1.60950983e+07 7.28635452e+06 3.52145345e+07]
 [7.85725234e+07 7.28635452e+06 3.31066733e+06 1.59517851e+07]
 [3.79496696e+08 3.52145345e+07 1.59517851e+07 7.70542826e+07]]
Least-Squares for n=4:
Beta 0: 6294.731056213379
Beta 1: 612.530839920044
Beta 2: 218.07334327697754
Beta 3: 1287.5460586547852
```

b) Update step X^T X (n+1)

In [89]:
```
# Read the data
X, Y = read_data()

# Create a sub-matrix with n=4
X_4 = X[:4, ]
Y_4 = Y[:4]

# Calculate the new design matrix X^T X (n+1)
XTX = np.linalg.inv(X_4 @ X_4.T)
u = XTX @ X[4].T
v = 1 / (1 + X[4] @ u)
XTX = XTX - v * (u @ u.T)
XTX
```

Out[89]:
```
array([[-1.86739490e+09, -2.25935268e+09, -1.67065732e+09,
        -2.06428681e+09],
       [-2.25935268e+09, -1.08404739e+09, -2.84929596e+09,
        -1.66895707e+09],
       [-1.67065732e+09, -2.84929596e+09, -1.07902509e+09,
        -2.26273272e+09],
       [-2.06428681e+09, -1.66895707e+09, -2.26273272e+09,
        -1.86569571e+09]])
```

c) Sequential LS estimator for n > 4 using Sherman-Morrison formula

In [93]:
```
X, Y = read_data()

X_4 = X[:4]
Y_4 = Y[:4]

# Calculate A = (X^TX)^-1 for n = 4
A = np.linalg.inv(X_4.T @ X_4)

# Calculate the LS estimator for n = 201 using the batch formula
beta_201 = np.linalg.inv(X.T @ X) @ X.T @ Y

# List for ||beta_i - beta_201||
beta_diff = []

# Sequentially update (X^TX)^-1 and compute the LS estimator for n > 4
for i in range(5, X.shape[0] + 1):
    # Add a new data point to X_4 and the corresponding value to Y_4
    new_X = X[i-1:i, :]
    new_Y = Y[i-1:i]
    X_4 = np.vstack([X_4, new_X])
    Y_4 = np.append(Y_4, new_Y)

    # Sherman-Morrison formula
    u = A @ new_X.T
    v = 1 / (1 + new_X @ u)
    A = A - v * (u @ u.T)

    # LS estimator for specific n
    beta_i = A @ X_4.T @ Y_4

    # ||beta_n - beta_201||
    norm_diff = np.linalg.norm(beta_i - beta_201)

    beta_diff.append(norm_diff)
```

d) Plotting the LS difference

In [97]:
```
# Plotting
plt.plot(range(4, 201), beta_diff, label=r'$||\hat{\beta}_n - \hat{\beta}_{201}
plt.xlabel('n')
plt.ylabel(r'$||\hat{\beta}_n - \hat{\beta}_{201}||$')
plt.title('Task 3 (d)')
plt.legend()
plt.show()
```