T1_soln

December 20, 2023

[1]:
```python
import numpy as np
import pandas as pd
from pandas import DataFrame
# Group Members: Dhvaniben Jasoliya Leutrim Uka Nicola Horst Tauqeer Rumaney␣
↪Yuvraj Dhepe

def read_data(path_x: str, path_y: str):
    x: DataFrame = pd.read_csv(path_x, sep=',', header=None, names=['x1', 'x2',␣
↪'x3'])
    y: DataFrame = pd.read_csv(path_y, sep=' ',  header=None, names=['y'])

    # transform the df into numpy arrays
    _x = np.array(x.values, 'float')
    _y = np.array(y.values, 'float')

    # add ones in x[0: ] for beta 0
    _x = np.column_stack((np.ones(len(_x)), _x))
    return _x, _y


def get_beta_hat_estimate(x: np.array, y: np.array):
    """
    We know that minimizing the LS estimator is equivalent to maximising the␣
↪(log) likelihood estimate ML. Therfore
    to obtain the ML w.r.t beta, we compute the optimum for the LS estimator by␣
↪using the derived formula
    (X.T @ X)^-1 @ X.T @ y
    """
    beta_hat_estimate = np.linalg.inv(x.T @ x) @ x.T @ y
    return beta_hat_estimate


def get_sigma_hat_estimate(x: np.array, y: np.array, beta_hat: np.array):
    """
    Using the ML Estimator for beta (beta_hat) we can compute the estimate for␣
↪the squared variance
```

```python
    Therefore we compute the sum of squared residuals with is equivalent to eps.T @ eps since eps = (y - X @ beta_hat)
    finally we return the sse divided by the number of samples to obtain our ML estimate for sigma^2 (sigma^2_hat)
    """
    sse: np.array = np.sum((y - x @ beta_hat)**2)
    n = len(x)
    return sse / n


def get_adjusted_sigma_hat_estimate(x: np.array, y: np.array, beta_hat: np.array):
    """
    Using the ML Estimator for beta (beta_hat) we can compute the adjusted estimator for the squared variance.
    This is almost the same computation as the non-adjusted version but we divide not by the number of samples but
    n-p-1 where n - 1 the total amount of freedom coming from our samples.
    p = is the amount of freedom we get form our prediction variables beta
    """
    sse: np.array = np.sum((y - x @ beta_hat) ** 2)
    n = len(x)
    p = len(beta_hat)

    return sse / (n - p - 1)



path_x: str = "./X.txt"
path_y: str = "./Y.txt"

# load the data
x, y = read_data(path_x, path_y)

# compute the estimator
beta_hat_estimator = get_beta_hat_estimate(x, y)
sigma_hat_estimator = get_sigma_hat_estimate(x, y, beta_hat_estimator)
ad_sigma_hat_sse = get_adjusted_sigma_hat_estimate(x, y, beta_hat_estimator)

print(f"ML Estimator   hat: \n{beta_hat_estimator}")
print(f"ML Estimator of  ^2: {sigma_hat_estimator}")
print(f"ML Estimator of adjusted  ^2 sse: {ad_sigma_hat_sse}")
```

```
ML Estimator   hat:
[[-0.00800698]
 [ 0.88161162]
 [-2.45938171]
```

```
 [-0.97715699]]
ML Estimator of  ^2: 0.9548405627555108
ML Estimator of adjusted  ^2 sse: 0.9791987403768247
```

2. **6 points**
Let Z be a multivariate random variable with expectation $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$. Let $A \in \mathbb{R}^{n\times n}$ be a matrix.

(a) Show that $\mathbb{E}[Z^T AZ] = \mu^T A\mu + \mathrm{tr}(A\Sigma)$.

(b) Show that $\mathrm{Cov}(AZ) = A^T \Sigma A$.

(c) Let $\hat{\beta} = (X^T X)^{-1} X^T y$ be the LS-estimator from the linear regression lecture, i.e. for $y = X\beta + \varepsilon$ and $\mathrm{Cov}(y) = \sigma^2 I_{n\times n}$. Show that $\mathrm{Cov}(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$.

Group Members:
Dhvaniben Jasoliya
Leutrim Uka
Nicola Horst
Tauqeer Rumaney
Yuvraj Dhepe

Since Z is multivariate, $Z = (Z_1, Z_2, \dots, Z_n)$ & $\mathbb{E}[Z] = \mu = (\mathbb{E}[Z_1], \dots, \mathbb{E}[Z_n])$
and $\mathrm{Var}(Z) = \underbrace{\Sigma}_{n\times n} = \mathbb{E}[\underbrace{(Z-\mu)}_{n\times 1}\underbrace{(Z-\mu)^T}_{1\times n}] = \mathbb{E}[Z \cdot Z^T] - \mu\mu^T$

with $\Sigma = \begin{pmatrix} \mathrm{Var}(Z_1) & \cdots & \mathrm{Cov}(Z_1, Z_n) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}(Z_n, Z_1) & \cdots & \mathrm{Var}(Z_n, Z_n) \end{pmatrix}$

constant not I.I.D.

a) $\mathbb{E}[Z^T A Z] = \mathbb{E}\left[\sum_{i=1}^n \sum_{j=1}^n a_{ij} \cdot X_i X_j\right] = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \cdot \mathbb{E}[X_i \cdot X_j] = \sum_{i=1}^n \sum_{j=1}^n a_{ij}\left(\mu_i \mu_j + \mathrm{Cov}(Z_i, Z_j)\right)$

$\mathbb{E}[X_i X_j] \Rightarrow \mathrm{Cov}(X_i, X_j) = \mathbb{E}[X_i X_j] - \mathbb{E}[X_i] \cdot \mathbb{E}[X_j]$
$\Rightarrow \mathbb{E}(X_i, X_j) = \mathbb{E}(X_i) \cdot \mathbb{E}(X_j) + \mathrm{Cov}(X_i, X_j)$

$= \sum_{i=1}^n \sum_{j=1}^n a_{ij} \mu_i \mu_j + \sum_{i=1}^n \sum_{j=1}^n a_{ij} \cdot \mathrm{Cov}(Z_i, Z_j) = \mu^T A \mu + \sum_{i=1}^n (A\Sigma)_{ii} = \mu^T A \mu + \mathrm{tr}(A\Sigma)$

b) $\mathrm{Cov}(AZ) = \mathbb{E}[(AZ)(AZ)^T] = \mathbb{E}[AZZ^T A^T] = A \cdot \mathbb{E}[Z \cdot Z^T] \cdot A^T = A \cdot \Sigma \cdot A^T$

c) $\hat{\beta} = (X^T X)^{-1} X^T y$ From the definition, we have $y = X\beta + \varepsilon$
$= (X^T X)^{-1} X^T (X\beta + \varepsilon)$
$= \cancel{(X^T X)^{-1} X^T X}\beta + (X^T X)^{-1} X^T \varepsilon$ $(X^T X)^{-1} \cdot X^T X$ is the identity matrix
$= \beta + (X^T X)^{-1} X^T \varepsilon \quad / \cdot \mathrm{Cov}$
$= \mathrm{Cov}(\beta + (X^T X)^{-1} X^T \varepsilon)$
$= \mathrm{Cov}((X^T X)^{-1} \cdot X^T \cdot \varepsilon)$
$= (X^T X)^{-1} \cdot X^T \cdot \mathrm{Cov}(\varepsilon) \cdot X \cdot (X^T X)^{-1}$
$= (X^T X)^{-1} \cdot X^T \cdot \sigma^2 \cdot I \cdot X (X^T X)^{-1}$
$= \sigma^2 (X^T X)^{-1} X^T X (X^T X)^{-1}$
$= \sigma^2 (X^T X)^{-1}$

Q 3)

As per the given Q: Likelihood is

$$L(p) = [(1-p)]^{2x} \cdot [2p(1-p)]^{y} [p^2]^{z}$$

$$\log(L(p)) = 2x \log(1-p) + y \log 2p + y \log(1-p) + 2z \log p$$

$\frac{\partial \log(L(p))}{\partial p} = 0$ to find MLE, we get

$$-\frac{2x}{1-p} + \frac{y}{p} - \frac{y}{1-p} + \frac{2z}{p} = 0$$

Multiplying with $p(1-p)$ on both sides

$$y(1-p) + 2z(1-p) = 2px + yp$$

$$y - py + 2z - 2zp = 2px + yp$$

$$y + 2z = 2px + 2py + 2zp$$

$$\therefore p = \frac{y+2z}{2x+2y+2z}$$

Page No

Date

$$\therefore \frac{\partial^2 \log(L(p))}{\partial p^2} = -\frac{2x}{(1-p)^2} - \frac{y}{p^2} - \frac{y}{(1-p)^2} - \frac{2z}{p^2} < 0$$

as x, y & z are counts of people & deno's are square $\therefore$ the critical pt value of p denotes the maximum.

$\therefore$ MLE is $\dfrac{y+2z}{2x+2y+2z}$.