Assignment 1 - Machine Learning

Emil Ramsbæk - 400550580

September 2023

Question 1

(a) $Z = X^T X + 0.1 I_{d \times d}$

Proof: If $X^T X$ is invertible it also implies it is a positive definite matrix, and when $0.1 I_{d \times d}$ matrix is added, which is also positive definite the sum of these will also remain positive definite.

Given any non-zero vector $u \in \mathbb{R}^d$:

$$u^T (X^T X) u > 0$$
$$u^T (0.1 I_{d \times d}) u = 0.1 \|u\|^2 > 0$$

Combining both of these, and i get:

$$u^T (X^T X + 0.1 I_{d \times d}) u > 0$$

Thus i can conclude that, $X^T X + 0.1 I_{d \times d}$ is positive definite, and invertible.

(b) $Z = [X|u]$

The additon og u gives the matrix $Z^T Z$ the block from:

$$Z^T Z = \begin{bmatrix} X^T X & X^T u \\ u^T X & u^T u \end{bmatrix}$$

In order for this matrix to not be invertible, it must mean the determinant is 0.

Counter example: Let $X = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $u = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. The matrix $Z^T Z$ is:

$$Z^T Z = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Therefore $Z^T Z$ is not invertible and its determinant is zero.

(c) $Z = [X^T|v]^T$

Given the definition of Z, I can derive XX^T, which is the matrix product of a $n \times d$ matrix. Even though the invertibility of $X^T X$, that is $d \times d$ is guaranteed it does not guarantee that the matrix derived the same X from a $n \times n$ is also invertible.

Counter example: Let $X = \begin{bmatrix} 1 & 0 \end{bmatrix}$ (a 1x2 matrix). The matrix $X^T X$ is:

$$X^T X = \begin{bmatrix} 1 \end{bmatrix}$$

which is invertible. However,

$$XX^T = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

is not invertible.

Question 2

(a)

Given the OLS solution:

$$W_{LS} = (X^T X)^{-1} X^T Y$$

Given Rank$(X) = n = d$, $X^T X$ is invertible.
Inserting X^T into the minimization objective for the sum of squared errors:

$$\begin{aligned} X^T(Y - XW_{LS}) &= X^T(Y - X(X^T X)^{-1} X^T Y) \\ &= X^T Y - X^T X(X^T X)^{-1} X^T Y \\ &= X^T Y - X^T Y \\ &= 0 \end{aligned}$$

Thus, I get:

$$X^T(Y - XW_{LS}) = 0$$

(b)

Given that X has more features than observations, this suggests that there isn't a unique best-fit line or hyperplane but multiple solutions W could minimize the residuals.

However, regardless of the rank of X, a foundational characteristic of the least squares problem is that the residuals, $Y - XW$, are orthogonal to the columns of X.

This implies that the residuals the model makes are uncorrelated to the predictor X.

Therefore, for any potential solution W to the least squares problem, the equation $X^T(Y - XW) = 0$ will always be valid.

(c)

Assume Rank$(X) = n = d$. I want to prove that $\|XW_{LS} - Y\|_2^2 = 0$.

Using the result from (a), where the solution implied:

$$X^T(Y - XW_{LS}) = 0$$

This equation shows that the residuals $Y - XW_{LS}$ are orthogonal to all columns of X. Given this orthogonality, the residuals are minimized for the solution W_{LS}.

Since Rank$(X) = n = d$ and $X^T X$ is invertible, the least squares solution W_{LS} results in a perfect fit to the data. This implies that the residuals $Y - XW_{LS}$ are zero for every data point.

$$\|Y - XW_{LS}\|_2^2 = \|0\|_2^2 = 0$$

This confirms that the squared norm of the residuals is zero, showing a perfect fit of the model to the data points when using W_{LS}.

(d)

Assume $\text{Rank}(X) = n < d$. I want to determine if $\min_W \|XW - Y\|_2^2 = 0$.

Given that the rank of X is less than d, the system is underdetermined. This means there are infinitely many solutions for W.

1. Since $\text{Rank}(X) = n < d$, there exists a non-zero vector z such that $Xz = 0$.
2. I let W_0 be a solution to $XW = Y$.
3. For any scalar c, $W = W_0 + cz$ is also a solution because:

$$X(W_0 + cz) = XW_0 + cXz = Y$$

4. Given the above, there are infinitely many W for which $XW = Y$ is true.
5. Therefore, it can be concluded that $\min_W \|XW - Y\|_2^2 = \|XW_0 - Y\|_2^2 = 0$.

This drives me to the given conditions, that the residuals can always be minimized to zero.

Question 3

TO find the line that best fit $\hat{y} = ax + b$, I will be using the least squares method, where i minimize the error between the function values and the predicted values. The error is given by:

$$E(a, b) = \int_0^1 (f(x) - (ax + b))^2 \, dx$$

Where $f(x) = 2x - 5x^3 + 1$.

To minimize the error I will take the partial derivatives for a and b and set them equal to zero. This will give me two equations which i can solve to find the values of a and b.

$$\frac{\partial E}{\partial a} = \int_0^1 -2x(f(x) - (ax + b)) \, dx = 0$$
$$\frac{\partial E}{\partial b} = \int_0^1 -2(f(x) - (ax + b)) \, dx = 0$$

From the integrals, I get the equations:

$$-\frac{1}{3} + \frac{2a}{3} + b = 0$$
$$-\frac{3}{2} + a + 2b = 0$$

Solving this system of equations, I get:

$$a = -\frac{5}{2}$$
$$b = 2$$

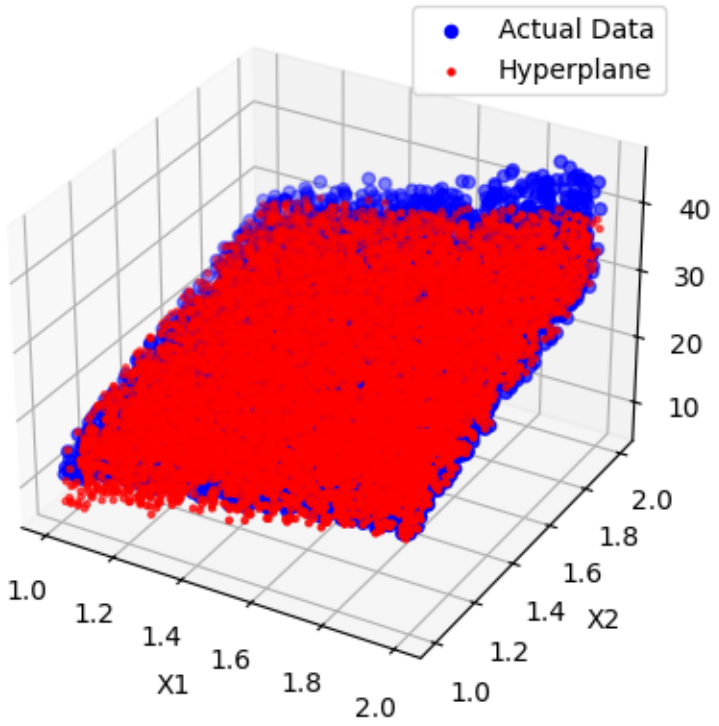
Thus, the best fit line for the function $f(x) = 2x - 5x^3 + 1$ when $n \to \infty$ is:

$$\hat{y} = -\frac{5}{2}x + 2$$

Question 4

Using a set seed, which allows me to obtain consistant results, my results were:

$$a \approx 11.34,\ b \approx 12.91,\ c \approx 9.53,\ d \approx 27.71$$



The code is attached at the end of the document.

Question 5

Given the objective function:

$$J(a) = \int_1^2 \left[\log\left(\frac{ax}{x^2}\right)\right]^2 dx$$

I get:

$$J(a) = \log^2(a) + (2 - 4\log(2))\log(a) + 2(1 + \log^2(2) - \log(4))$$

To find the optimal value of a, I will differentiate $J(a)$ with respect to a and set the result equal to zero.

The solution to this is:

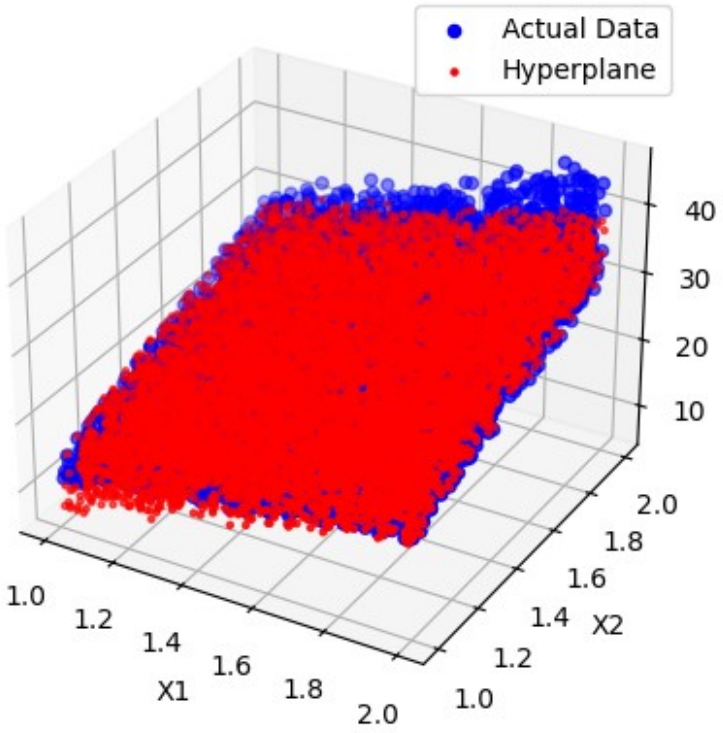
$$a = \frac{4}{e}$$

where e is the base of the natural logarithm, approximately equal to ≈ 2.71828.

```python
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

np.random.seed(1337)

# Generate sample data uniformly distributed in [1, 2]^3
num_samples = 10000
x1 = np.random.uniform(1, 2, num_samples)
x2 = np.random.uniform(1, 2, num_samples)
x3 = np.random.uniform(1, 2, num_samples)

#Target function
def f(x1, x2, x3):
    return x1 + 3*x2 + 4*x3 + 5*x1*x2 - 5*x2*x3 + x2**2*x3**2 + (x1 +
x2)**x3

y = f(x1, x2, x3)

#Input matrix for OLS
X = np.column_stack((x1, x2, x3, np.ones(num_samples)))

# Fit the model
model = LinearRegression(fit_intercept=False)
model.fit(X, y)

# Extract coefficients
a, b, c, d = model.coef_

print(f"a: {a}, b: {b}, c: {c}, d: {d}")
```

```
a: 11.34123614620876, b: 12.913533517281461, c: 9.531161458231729, d:
-27.715931843378396
```

```python
y_pred = model.predict(X)

# Visualization
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x1, x2, y, color='b', label='Actual Data')
ax.scatter(x1, x2, y_pred, color='r', marker='.', label='Hyperplane')
ax.set_xlabel('X1')
ax.set_ylabel('X2')
ax.set_zlabel('Y')
ax.legend()
plt.show()
```


Actual Data
Hyperplane
40
30
20
10
2.0
1.8
1.6
1.4
X2
1.2
1.0
1.0
1.2
1.4
1.6
1.8
2.0
X1