Machine Learning

Homework 1

JARRY Guillaume

Part I

Problem 1

1 Q1

Because X_i is a fixed data point, we can infer: $Y_i|X_i, w, b = Y_i|w, b$ And then because

$$Y_i|w, b \sim \mathcal{N}(X_i^T w - b, \sigma^2$$

Thus:

$$P(Y_i|w, b) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{(Y_i - X_i^T w - b)^2}{2\sigma^2}\right)}$$

2 Q2

Because the variables $Y_i|w, b$ are all independant from one another, the log-likelyhood of the data will take the form of the logarithl of a product:

$$log(P(Y|\beta)) = log\left(\prod_{i=1}^{n} P(Y_i|w, b)\right) = log\left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{(Y_i - X_i^T w - b)^2}{2\sigma^2}\right)}\right)$$

3 Q3

Now that we have the expression of the log-likelihood, we can search for its maximum by minimizing its opposite. First let us simplify the expression:

$$argmin - log(P(Y|\beta)) = \sum_{i=1}^{n}\left(log(\sigma) + \frac{(Y_i - X_i^T w - b)^2}{2\sigma^2}\right)$$

Then, because σ is a constant, it does not intervene in the minimization problem and thus we can multiply the expression by $2\sigma^2$ and neglect $log(\sigma)$. This then gives the optimization problem:

$$-log(P(Y|\beta)) = \sum_{i=1}^{n}(Y_i - X_i^T w - b)^2) = \sum_{i=1}^{n}(Y_i - X_i'^T \beta)^2)$$

In which we can clearly recognize the expression of a norm and a least squarre problem

$$-log(P(Y|\beta)) = \|Y - X'\beta\|^2$$

And thus we have shown that solving the maximum of likelihood problem 1 is the same as solving the least square problem (1).

4 Q4

Because we have established the equivalency between the maximum likelihood estimation and the least square problem, we can find the maximum of likelihood by inputing the analytical formula of the solution of the least square problem. First, let us check if this problem is convex by calculating the Hessian of the function $\beta \to \|Y - X'\beta\|^2$. (found in the nyu paper attached to the homework, line 5g, after computing the gradient below, line 5e):

$$\nabla^2(\|Y - X'\beta\|^2) = {X'}^T X'$$

Here, because X' is of full rank on the column space, the hessian will be invertible, which guaranties that the problem is convex. Then, because the problem is convex, the solution will be β which nulifies the gradient, (found in the nyu paper attached to the homework, line 5g), ie :

$$\nabla_\beta(\|Y - X'\beta\|^2) = 2{X'}^T Y - {X'}^T X\beta = 0$$

Here, X' is of full rank, thus ${X'}^T X'$ is invertible, and the formula will be :

$$\beta = ({X'}^T X')^{-1} {X'}^T Y$$

Thus we have computed β that maximizes the log-likelihood.

Part II

Problem 2

5 Q1

Using the central limit theorem, knowing that $E(U_i) = 1/2$ and $V(U_i) = \frac{1}{12}$, because they are all independant and identically distributed, we find the formula:

$$\sqrt{N}\left(\frac{(\frac{1}{N}\sum_{i=1}^{N} U_i) - 1/2}{\sqrt{\frac{1}{12}}}\right) \sim \mathcal{N}(0, 1)$$

Thus, to generate our approximation of a standard normal distribution, we will draw from a number of uniform variables U_i , and to the result of the draw we apply the formula $\sqrt{N}\left(\frac{(\frac{1}{N}\sum_{i=1}^{N} U_i) - 1/2}{\sqrt{\frac{1}{12}}}\right)$. The number given will be our approximation of a draw from the standard normal distribution.

Then, because we are not allowed to use division of multiplication, we must choose N carefully. Let us remark that for $N = 12$, all the division and multiplication disappear and we get:

$$\sum_{i=1}^{12} U_i - 6 \sim \mathcal{N}(0, 1)$$

This equation gives us the pseudo code right below: S = 0 repeat twelve times: draw from a uniform variable add the result to S return S - 6

Machine Learning

Homework 1
JARRY Guillaume

6 Q2

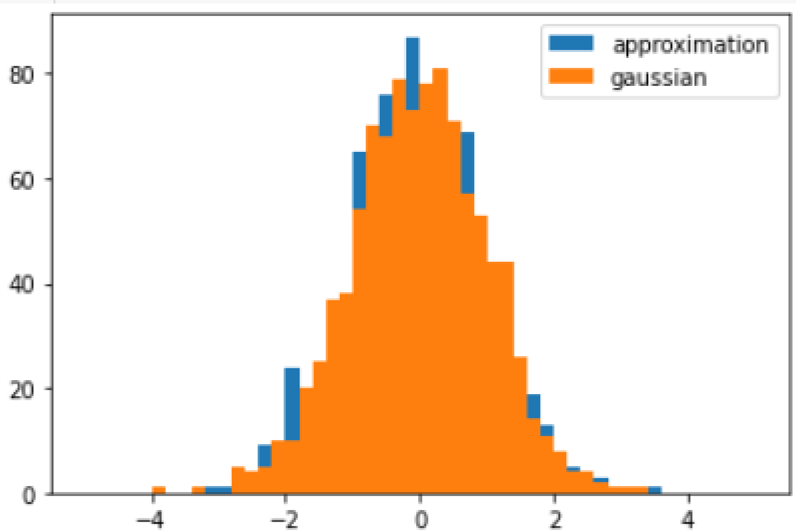
The code below generates the graph below:

```python
import numpy as np
import matplotlib.pyplot as plt
```

```python
def gaussian_estimation():
    S = 0
    for i in range(12):
        S += np.random.uniform()
    return S - 6
```

```python
approximate = [gaussian_estimation() for i in range(1000)]
normal = np.random.randn(1000)
```

```python
plt.hist(approximate, bins=50, range=(-5, 5), label='approximation')
plt.hist(normal, bins=50, range=(-5, 5), label='gaussian')
plt.legend()
plt.show()
```



We can see that the two plot overlap really well, The empirical distribution are both centered and seem to have similar variance. Their shape is also similar. All of this tends to show the validity of our procedure.