List 1 Report

Statistics and Linear Models

Dawid Dieu

8 November 2022

Task 1

In the first task, we are asked to generate n observations from a $N(\theta, \sigma^2)$ distribution. Then we had to calculate a value of an estimator of the parameter θ of four forms.

(i) $\hat{\theta}_1 = \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$

(ii) $\hat{\theta}_2 = Me\{X_1, \ldots, X_n\}$

(iii) $\hat{\theta}_3 = \sum_{i=1}^{n} w_i X_i, \sum_{i=1}^{n} w_i = 1, 0 \leq w_i \leq 1, i = 1,\ldots,n$

with an arbitrary weights selection

(iv) $\hat{\theta}_4 = \sum_{i=1}^{n} w_i X_{i:n}$, where $X_{1:n} \leq \ldots \leq X_{n:n}$ are the order statistics from the sample $X_1, \ldots, X_n$, $w_i = \phi(\Phi^{-1}(\frac{i-1}{n})) - \phi(\Phi^{-1}(\frac{i}{n}))$, while ϕ is the density and Φ is the cumulative distribution function of the standard normal $N(0,1)$ distribution.

Then we had to run those calculations 10000 times. Here are the results:

a) $n = 50, \theta = 1, \sigma = 1$

	(i) $\hat{\theta}_1$	**(ii) $\hat{\theta}_2$**	**(iii) $\hat{\theta}_3$**	**(iv) $\hat{\theta}_4$**
variance	0.020236	0.030750	0.026426	0.020583
MSE	0.020240	0.030762	0.026429	1.013974
Bias	-0.001809	-0.003445	-0.001576	-0.996690

The variance of all estimators was low because we were running the experiment 10000 times. The first three estimators had quite close values. However, we can see that θ_1 had a smaller variance, MSE and bias because it is the true estimator. One can see that if we subtract a very large number from the smallest observation or add a very large number to the largest observation, θ_2 would not change but θ_1 would be heavily influenced. In the case of θ_3 , it was dependent on weights selection. When I was drawing from the uniform distribution, the estimator was behaving similarly to θ_1 and θ_2. But when drawn from the standard normal distribution, variance and MSE were growing to values near 1.4 (they were growing because more weight was in the middle close to 0, when the true mean $\theta = 1$). The last estimator had the biggest bias and quite big MSE. It's understandable, as we were drawing from $N(0,1)$ (the same argument as for θ_3 with standard normal weights).

b) $n = 50, \theta = 4, \sigma = 1$

	(i) $\hat{\theta}_1$	(ii) $\hat{\theta}_2$	(iii) $\hat{\theta}_3$	(iv) $\hat{\theta}_4$
variance	0.020141	0.030749	0.026806	0.019733
MSE	0.020141	0.030750	0.026806	16.013985
Bias	0.000318	-0.000352	0.000531	-3.999282

The number of experiments was still 10000, so the variance did not change and remained low. θ_1 remained the best estimator for the same reasons as in a). We can see that MSE for θ_4 increased a lot due to the larger true mean. The fourth estimator was close to 0, that's why the bias is almost -4 and MSE 16.

c) $n = 50, \theta = 1, \sigma = 2$

	(i) $\hat{\theta}_1$	(ii) $\hat{\theta}_2$	(iii) $\hat{\theta}_3$	(iv) $\hat{\theta}_4$
variance	0.080680	0.122366	0.106017	0.081172
MSE	0.080701	0.122376	0.106031	1.083821
Bias	-0.004582	-0.003050	-0.003701	-1.001324

This case is similar to a), but we have a bigger standard deviation. Therefore we see bigger variances. Estimators (1-3) were close to the real value as the biases are small, but due to the larger σ we can observe a higher MSE.

Task 5

In the fifth task, we were asked to generate n observations from a logistic distribution $L(\theta, \sigma)$ with the shift parameter θ and the scale parameter σ. Then we had to calculate the variance, MSE and bias of the estimator by running an experiment 10000 times. As we know, we cannot find the closed-form solution of the MLE estimator for the logistic regression. That's why we have to choose some optimization algorithm. I chose the *scipy.optimize.minimize* function that can minimize a function of one or more variables. I will present different methods like BFGS and L-BFGS-B. In the tables shown below, the starting point was the population mean $\bar{X}$.

a) $n = 50, \theta = 1, \sigma = 1$

	BFGS	L-BFGS-B
variance	0.06	0.06
MSE	0.06	0.06
bias	0.0	0.0
number of steps	2.33	2.39

b) $n = 50, \theta = 4, \sigma = 1$

	BFGS	L-BFGS-B
variance	0.06	0.06
MSE	0.06	0.06
bias	0.0	0.0
number of steps	2.32	2.4

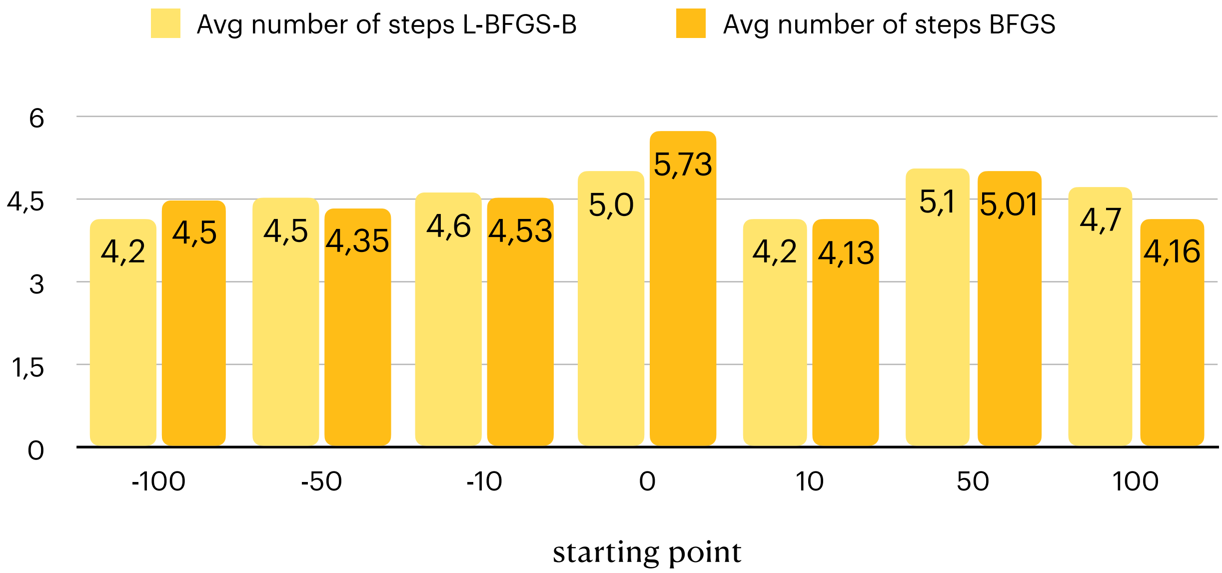
c) $n = 50, \theta = 1, \sigma = 2$

	BFGS	L-BFGS-B
variance	0.24	0.24
MSE	0.24	0.24
bias	0	0
number of steps	1.89	2.1

In all cases, the algorithm worked perfectly and always found the true θ in less than three steps!

We will also look at the influence of the starting point x_{st} for the second case, where $n = 50, \theta = 4, \sigma = 1$ with L-BFGS-B and BFGS method.

In all of the scenarios, the algorithm was able to find the true θ or the very close value (like 3.99891…). The average number of steps for both methods was also very low.



Task 6

In this task, we repeat exactly the same experiments from Task 5, but for the Cauchy $C(\theta, \sigma)$ distribution.

a) $n = 50, \theta = 1, \sigma = 1$

	BFGS	L-BFGS-B
variance	0.09	0.08
MSE	0.09	0.08
bias	0.0	0.0
number of steps	3.27	3.29

b) $n = 50, \theta = 4, \sigma = 1$

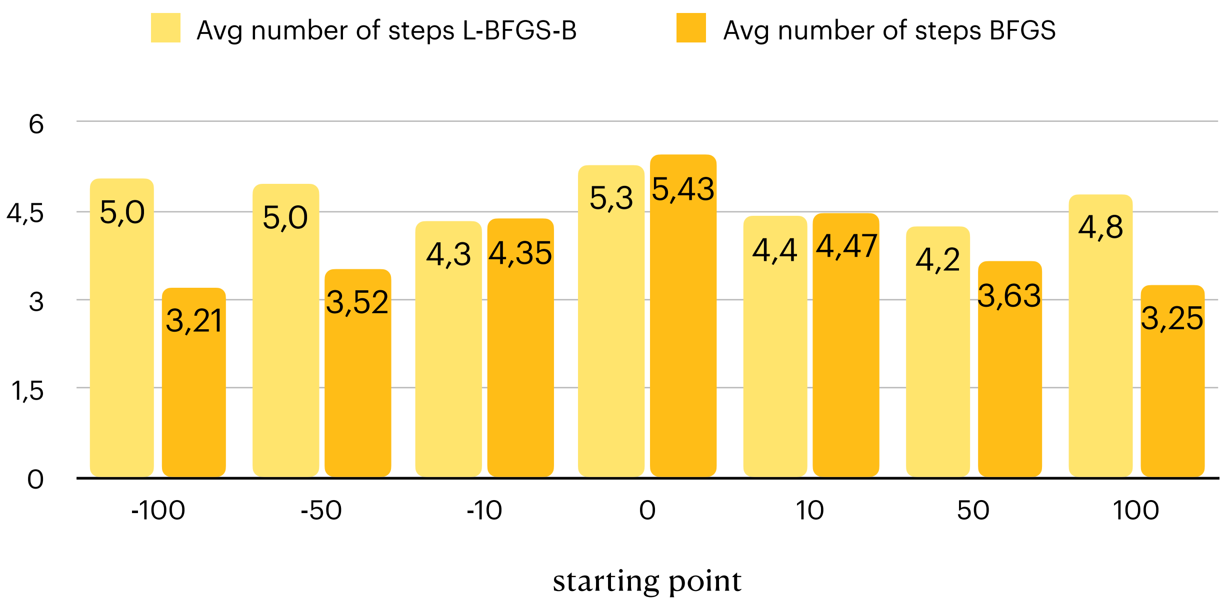
	BFGS	L-BFGS-B
variance	0.08	0.08
MSE	0.08	0.08
bias	0.0	-0.01
number of steps	3.26	3.28

c) $n = 50, \theta = 1, \sigma = 2$

	BFGS	L-BFGS-B
variance	0.32	0.33
MSE	0.32	0.33
bias	0.01	0.0
number of steps	3.06	3.13

Here, the algorithm also worked perfectly and found the true θ. The number of steps increased by one, and errors (MSE) were a tiny bit bigger.

Looking at the average number of steps for the second case, where $n = 50$, $\theta = 4, \sigma = 1$ with L-BFGS-B and BFGS method. The found estimators were also equal to θ. In the case of BFGS, we can observe that the closer the starting value was to 0, the greater number of steps to find the solution was.



Task 7

In this task, we are asked to repeat the experiments from tasks 1, 5 and 6 for $n = 20$ and $n = 100$.

In my opinion, the results weren't very surprising. Here are a few observations. In the case of a smaller $n = 20$ the variance of estimators was higher, for $n = 100$ it was much lower. The bigger the sample size is, the better it can describe the underlying distribution, that's why we would expect lower MSE. The optimizers from tasks 5 and 6 also need a lower number of steps to reach the optimal value.

Let's generate samples from $N(4,1)$. As one can see on this chart, for $n = 100$ the distribution looks the closest to the normal one.