IX1501 HT24 Project 2

Bootstrapping for an estimation of probability

1. Implementation and Assessment

The course includes three mandatory project tasks on a total of 3.5 hp. They will be given a summary grade in Pass/Fail (G/U). In project 1 and 2, you will work in a group of two and solve computer-based tasks, write a report (English), and prepare a short oral presentation on your solution to the tasks (English). Carefully read the following instructions so that you know which rules apply and what is expected from you.

1.1. Report

The report should be written in English and contain the following:

- Title and authors of the report
- KTH emails of the authors
- Summary of the results and findings (maximum 250 words)
- Separate sections for each part containing e.g.,
    - Description of the methodology
    - Mathematical formulas and equations (if relevant)
    - Numerical results (including figures and/or tables)
    - Analysis and discussion
- Separate code section (do not mix the code and the main text in the report)
    - Code must contain comments detailed enough for an easy understanding.

The report must be uploaded before the deadline (see Section 1.4 below). The file type for the report is limited to pdf. Note that the **report is a group task**.

1.2. Oral presentation

You should prepare an oral presentation of your solution to the task. You should record your presentation, create a video file, and upload the file before the deadline. Instructions on how to create a recorded presentation can be found on a Canvas page under Projektuppgifter module. It is important that your face (video, not a still image) must be overlaid to the presentation material in your video. The video ***must not exceed five minutes***. Also, you should make sure that the uploaded file has adequate audio-visual quality. Carefully consider what is important, in what order and how it will be illustrated. English shall be used for the presentation. Note that the **video presentation is an individual task**.

1.3. Rules

Although it is a team project, **you, as an individual, must have full knowledge of all the material you you submit and present**. To be approved, you must have solved the task and be able to explain the entire task

and solution. The project report will be prepared and uploaded per group of two students. On the other hand, the presentation material and video should be prepared and uploaded individually.

To account for the task that you do not have solved is considered cheating. It is also cheating to copy the whole or a part of a solution from others. If two solutions are deemed as (partial) copies, both will be rejected. If the solution contains parts that you do not have produced, e.g., background material, you must clearly indicate this and specify the source. Suspicion of cheating or misbehaving can be reported to the Disciplinary Board.

1.4. Examination

The project report and video should be uploaded before the deadline via Canvas.

If your report and/or video is not satisfactory, you will be requested to participate in an additional Zoom Q&A session on the presentation (redovisning) date. In addition, you can be randomly selected for the Q&A regardless of the quality of your work. Therefore, all students should be available for a Zoom meeting on the presentation (redovisning) date. The exact time will be announced a few hours before the session if you are selected. You may be requested to demonstrate and execute the code during the session.

Notice that the report and video should be uploaded in time even if you cannot attend the Zoom Q&A session. If you are not available on the redovisning date, you must inform the teacher, **Zhenyu Li** (zhenyuli@kth.se) with cc-ing Ki Won Sung (sungkw@kth.se), by email in advance.

If you do not fulfill any of the requirements without contacting the teachers in advance (in writing), you will have to wait for the next course for a new project exam.

2. Project Task

📢 Read Section 3 Background Knowledge carefully before delving into the task!

📢 As for the programming language, you are allowed to use Python or Mathematica for this project.

We have a sample consisting of 10 observations as follows:

56, 101, 78, 67, 93, 87, 64, 72, 80, 69

The sample is considered to be a realization of 10 independent and identically distributed (i.i.d.) random variables from an unknown probability distribution with unknown mean μ.

For given constraints $a = -4$ and $b = 6$, we are interested in estimating the following probability:

$$p = P\left(a < \frac{\sum_{i=i}^{n} X_i}{n} - \mu < b \right)$$

Your task is as follows:

[**Task 1**] Explain how bootstrapping method can be used to estimate p. Include a pseudo code in your explanation.

[**Task 2**] Estimate p with bootstrapping method.

(Hint: what is the most reasonable estimator of μ? Use it in the place of μ)

3. Background Knowledge

Bootstrapping is a statistical method of estimating parameters from a small sample by means of resampling, i.e., creating many virtual (or simulated) samples out of the original sample. This method is based on the principle of random **sampling with replacement** (*dragning med återläggning*).

The term bootstrapping comes from the phrase "pull oneself up by one's bootstraps," which means improving one's position by one's own efforts without help from anyone else. As the name suggests, the method of bootstrapping is useful particularly when we have a limited dataset and a limited knowledge of the population distribution. This method allows estimation of almost any statistical parameters.

Assume that we have a sample of size 5: [1, 2, 3, 4, 5]. With the sampling with replacement, we can create a new sample that is also of size 5, for example, [3, 1, 2, 4, 4]. This is called a **bootstrap sample**. We can create many other bootstrap samples such as [2, 5, 3, 1, 1], [1, 4, 4, 3, 2], [2, 2, 3, 4, 5], and so on. With sufficiently large resampling, we can estimate the statistical parameters of the interest.

Programming languages have built-in functions for the sampling with replacement. For example,

- Python has *random.choice*. See this link for details.
- Mathematica has *RandomChoice*. See this link for details.

Let us consider an example of how bootstrapping method can be used for an estimation of the confidence interval of mean value. Assume that we have a sample of size 10 from normal distribution with $\mu = 5$ and $\sigma = 10$, i.e., $X_i \in N(5,10),\ i = 1,\ldots,10$ as below and we are interested in 95% confidence interval of μ.

{14.6, -6.3, 9.1, 13.2, 14.5, 11.9, -4.2, 18.7, -11.6, 3.5}

Note that $\bar{x} = 6.34$ and $s = 9.87$ for the above sample.

If we know that the population is normal distributed and also know σ, the 95% confidence interval is given by

$$\mathrm{CI}_\mu = \bar{x} \pm q^N_{0.975} \frac{\sigma}{\sqrt{n}} = 6.34 \pm 1.96 \frac{10}{\sqrt{10}} = \left[0.14,\ 12.54\right].$$

For the case of unknown σ, the confidence interval is

$$\mathrm{CI}_\mu = \bar{x} \pm q^t_{0.975}(n-1) \frac{s}{\sqrt{n}} = 6.34 \pm 2.26 \frac{9.87}{\sqrt{10}} = \left[-0.71,\ 13.39\right],$$

which is broader than the former.

Now, assume that we do not have any information about the sample. Then, the above formula cannot be used because the sample size is not large enough to apply the central limit theorem. Yet, we can still use the bootstrapping method to estimate the confidence interval. The procedure is as follows:

- Generate m independent resamples (bootstrap samples) of size 10 out of the original sample by the sampling with replacement.
- For each bootstrap sample, compute its sample mean (i.e., a bootstrap estimate)
- Sort the sample means (bootstrap estimates) in ascending order
- Take the 97.5th percentile and 2.5th percentile from the list.

Below is a Python code for this example.

```
from numpy import random
import numpy as np

# A sample of size 10 from N(5,10)
X=[14.6, -6.3, 9.1, 13.2, 14.5, 11.9, -4.2, 18.7, -11.6, 3.5]

# Set the number of resampling (m)
NumberOfResamples = 1000
# This is the list storing bootstrap estimate
RecordResult = []

# Collecting bootstrap estimate of sample mean
for j in range (NumberOfResamples):
    Temp = random.choice(X, 10)
    RecordResult.append(np.mean(Temp))

# Sorting the result to obtain confidence interval
RecordResult = np.sort(RecordResult)

# Pick the lower and upper 2.5% percentile values
Lower = int(np.floor (NumberOfResamples*0.025))
Upper = int(np.floor (NumberOfResamples*0.975))
ConfIntBootstrap = [RecordResult[Lower], RecordResult[Upper]]

print('Confidence interval with bootstrap:', ConfIntBootstrap)
```

```
Confidence interval with bootstrap: [-0.14999999999999983, 12.11]
```

With 1000 resampling, bootstrapping gives a confidence interval of $[-0.15,\ 12.11]$, which is close to that with known σ.