Stat4DS | Homework 02

Pierpaolo Brutti

Due Friday, January 12 (on Moodle)

General Instructions

I expect you to upload your solutions (only 1 per team) on Moodle as a **single running** `R Markdown` file (`.rmd`) + its `html` output, **named with your surnames**. Alternatively, a `zip`-file with all the material inside will be fine too.

You will give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Your responses must be supported by both textual explanations and code.

R Markdown Test

To be sure that everything is working fine, start `RStudio` and create an empty project called `HW1`. Now open a new `R Markdown` file (`File > New File > R Markdown...`); set the output to `HTML mode`, press `OK` and then click on `Knit HTML`. This should produce a web page with the knitting procedure executing the default code blocks. You can now start editing this file to produce your homework submission.

Please Notice

- For more info on `R Markdown`, check the support webpage that explains the main steps and ingredients: R Markdown from RStudio. For more info on how to write math formulas in LaTex: Wikibooks.
- Remember our **policy on collaboration**: *collaboration on homework assignments with fellow students is* ***encouraged****. However, such collaboration should be clearly acknowledged, by listing the names of the students with whom you have had* ***discussions*** *(no more) concerning your solution. You may* ***not****, however, share written work or code after discussing a problem with others. The solutions should be written by* ***you and your group*** *only.*

---

1. Background: Be Safe!

It is now common practice for organizations with online presence or for web creators like youtubers and similia to run large-scale randomized experiments (often called *A/B tests*), to improve product performance or user experience or video-related metrics such as *Watch Time*, *Impressions*, *Click-Through Rate (CTR)*, etc.

Visiti la pag web che ha 2 versioni diverse ma tu non lo sai e vedono se l'utente lo apprezza o meno

Such experiments are inherently **sequential**: visitors arrive in a stream and outcomes are typically observed quickly relative to the duration of the test.
Despite this, results are often monitored *continuously* using inferential methods that assume a *fixed* sample size: a very bad idea as hightlighted in the statistical literature since the early '70s (Armitage et al., 1969; Berman et al., 2018).
Furthermore, most of these A/B tests are run with little formal planning compared to clinical trials or industrial quality control, the traditional applications of sequential analysis.

In order to describe a satisfying solution, to fix the ideas, consider the problem of estimating a population mean $\mu = \mathbb{E}(X)$ from a sequence of IID data $\{Y_t\}_{t=1}^{\infty} = \{Y_1, Y_2, \ldots\}$ that, differently from usual, are observed **sequentially** over time, not in one single shot/batch.

From our notes we know that, by definition, a **non-asymptotic** $(1-\alpha)$ level confidence interval (CI) for μ is a random set $\dot{C}_n = \dot{C}(Y_1, \ldots, Y_n)$[1] such that:

$$\text{For any sample size } n,\ \Pr\left(\mu \in \dot{C}_n\right) \geqslant 1-\alpha \quad \text{or} \quad \text{For any sample size } n,\ \Pr\left(\mu \notin \dot{C}_n\right) \leqslant \alpha. \tag{1}$$

It's important to remark that the coverage guarantee in Equation 1 of a CI is only valid at *some* **prespecified** sample size n, which **must** be decided **before** we see any data. Peeking at the data in order to determine the sample size is a well known form of *data snooping*: to be avoided at all costs!

[1] We use overhead dots $\dot{C}_n$ to denote fixed-time (pointwise) confidence **intervals** and overhead bars $\overline{C}_t$ to denote time-uniform confidence **sequences**.

However, as already highlighted, in many applied contexts, it is restrictive to fix n beforehand, and even if clever sample size calculations are carried out based on prior knowledge, it is impossible to know *a priori* whether the selected sample size n will be large enough to detect some signal of interest: after collecting the data, one may regret collecting too little data or collecting much more than would have been required.

Confidence Sequences (CS) provide the flexibility to choose sample sizes data-adaptively while still controlling for the coverage level (or the Type-I error rate in their hypothesis testing version).

Formally, a CS is a sequence of CIs $\{\overline{C}_t\}_{t=1}^{\infty}$ such that

$$\Pr\left(\mu \in \overline{C}_t \text{ for any sample size } t\right) \geqslant 1-\alpha \quad \text{or} \quad \Pr\left(\text{There is a sample size } t \text{ such that } \mu \notin \overline{C}_t\right) \leqslant \alpha. \tag{2}$$

The statements in Equation 1 and Equation 2 look very similar but are markedly different from the data analyst's or experimenter's perspective. In particular, employing a CS has the following implications:

1. The CS can be (optionally) updated whenever new data become available;
2. Experiments can be continuously monitored, adaptively stopped, or continued;
3. The coverage levels and/or the Type-I error rate are controlled at all stopping times, including *data-dependent* times.

2. An Basic Example

In this homework in particular, and this class more in general, there is no time (thx god!) to introduce the probabilistic machinery needed to construct CSs from scratch (i.e. martingales theory).

Consequently here we will just consider a basic example of CS for the population mean $\mu = \mathbb{E}(X)$ that we can then easily contrast and compare with the usual Hoeffding/Chebyshev/Gaussian solutions we explored in class.

Hence, given a sequence of IID observations $\{Y_t\}_{t=1}^{\infty} = \{Y_1, Y_2, \ldots\}$ from a 1-sub-Gaussian distribution[2] with mean $\mu = \mathbb{E}(X)$, the following is a $(1-\alpha)$ confidence sequence for μ

$$\frac{\sum_{i=1}^{t} X_i}{t} \pm 1.7 \cdot \sqrt{\frac{\log\log(2t) + 0.72\log(10.4/\alpha)}{t}}. \tag{3}$$

If we are willing to relax the "anytime" validity of the CS, then the following can be considered (in a very technical sense) an *asymptotic* $(1-\alpha)$ CS for μ (assuming only a population with finite variance):

$$\frac{\sum_{i=1}^{t} X_i}{t} \pm \widehat{\sigma}_t \cdot 1.7 \cdot \sqrt{\frac{\log\log(2t) + 0.72\log(10.4/\alpha)}{t}}, \tag{4}$$

where $\widehat{\sigma}_t$ denotes the sample variance based on the first t observations.

⇝ Your job ⇜

1. In a nutshell, design a *simulation study* to compare the performance of the two CSs in Equation 3 and Equation 4 with our classics, i.e. Hoeffding/Chebyshev/Gaussian CIs.
   You've to pick a suitable population to sample from (possibly two), the simulation size, the level α you're targeting, the min and max sample sizes $t_{\min}$ and $t_{\max}$ you'll consider and, of course, some relevant metrics.
   Two important examples are: the "*Cumulative Miscoverage Probability*" (i.e. how frequently the intervals $\dot{C}_t$ and $\overline{C}_t$ do not capture the true mean μ for different t), and the "*Running Interval Lenght*" (i.e. how large the intervals are, on average, at different t).
   Suitable comments and relevant visualizations of the metrics and the CSs/CSs themselfes will be very welcome.

2. Now imagine your goal is to design a sequential "self-experiment" entitle **Environmental Sound Levels and Mood** to investigate the relationship between, well, environmental sound levels and **your** mood over an extended period of time (say 2-3 weeks).
   Describe in details how you would like to organize the data collection, what kind of tools/phone app/etc you may use, what kind of *intervention* you may intentionally try to check the effect on *you* and, more importantly, what population parameter you'd like to monitor/estimate over time to draw relevant conclusions (. . . and how CSs may help in this!)
   Notice: you don't have to actually run the experiment (for now), just describe it in *realistic* details.

[2] Essentially, sub-Gaussian random variables have a distribution whose tails decay to zero at least as fast as the tails of a Gaussian. A notable example are random variables with bounded support like those handled by Hoeffding inequality.