ELEC-E7130 Assignment 7. Sampling

Markus Peuhkuri Esa Hyytiä Seyud Mortezaei
Tran Thien Thi Weixuan Jiang
César Iván Olvera Espinosa Yu Fu

Prerequisites

1. To complete this assignment, students are required to have prior knowledge about how to use R or Python, statistics, and how to leverage the libraries available to process, analyze and plot data.

   If you lack the relevant skills, you may want to

   a. Take a look through related slides
   b. Refer to earlier assignments where you can learn some R and Python knowledge.
   c. Read the supporting materials

Learning outcomes

At the end of this assignment, students should be able to:

1. Gain a more detailed understanding of the utility of sampling and sampling distributions.
2. Develop a good understanding of how to sample their data sets for various applications.
3. Estimate the true mean using the sample mean in different ways, both from stored data sets and real-time data.
4. Handle data appropriately before processing it and prepare the data set for real machine learning cases, continuing with the process of selecting the best model through respective training, evaluation, and predictions.

Introduction

This assignment contains three tasks to cover important topics related to different sampling applications such as *off-line estimation* (collecting samples through a data set stored), *on-line estimation* (collecting samples in real-time, i.e., streaming data), or even one of the types of sampling for Machine Learning

purposes called *stratified random sampling*. Please read all instructions before starting because it is helpful to identify common work.

- Task 1: Sampling and distributions (off-line sampling)
- Task 2: High variability (on-line sampling)
- Task 3: Data pre-processing for ML purposes

All data for tasks 1 and 2 can be found from the `sampling-data.zip` archive from the assignment page and `$RDATA/sampling-data.zip` (extracted to `$RDATA/sampling/` directory) at Aalto IT computers; while the file for task 3 is located in the `/work/courses/unix/T/ELEC/E7130/general/ml-data` directory or using `MLDATA` environment variable as a path.

Task 1: Sampling and distributions (off-line sampling)

The first task aims to familiarize oneself with sampling and sampling distributions, and the size of sampling for statistics from a data set (***off-line sampling***).

Download file `sampling.txt` which contains certain session inter-arrival times to study estimation of the mean inter-arrival time based on different sample sizes.

Complete the next action points:

1. Plot the histogram of the original data and compute the mean.
2. Select 5000 random samples from original data (i.e., you should have a vector with length 5000 values). Plot its histogram and compute the mean.
3. Select 10000 times n random elements from the data to compute the mean of these n values. As a result, you should have a vector of 10000 values, where each of them is the mean value of n random elements. In this case, we will consider 3 scenarios:
   - $n = 10$
   - $n = 100$
   - $n = 1000$ For each scenario of n, you should:
   - Plot the histogram of these 10000 values as well as Q-Q plot (or any of the goodness-of-fit plots) to study the values against normal distribution.
   - Compute the mean and standard deviation of these 10000 values.
   - Compute and analyze the *sampling error-single mean* and *variance*.

     Note: Sampling error-single mean refers to the discrepancy between a sample statistic ($\bar{x}$), which represents the average value of the sample, and its corresponding population parameter (μ), which denotes the true mean of the entire population.

Mathematically,
the sampling error is calculated as:

$$Sampling\ error\ = \bar{x} - \mu$$

Note: Each mean contained in the vectors represent different results you could get for your statistic in a random sample and can be seen as samples from the sampling distribution of the sample mean statistic for n samples.

Discuss the following points: - Explain *the effects of sample size* on the sampling distribution and the accuracy of the estimate based on both the results (mean and standard deviation) and plots obtained by the different values of n and 5000 random samples concerning the original data. - What observations can be made regarding the presence of sampling bias in each scenario?

Tips: - Useful Python libraries could be `pandas`, `matplotlib`, `numpy`, `seaborn`, `fitdist`, `scipy` and `statistics` (for `variance()`) library. Besides, `sm.qqplot()` can be used to plot Q-Q plot. - Useful R functions could be `hist()`, `fitdistr()`, `rnorm()`, `qqplot()`, `mean()`, `sd()`, and `var()`.

Report, task 1

- Histogram plots of each case.
- Mean values of each case.
- Q-Q plots, mean, standard deviations, sampling error-single mean and variance for the different cases of n.
- Discuss and draw observations of each case in terms of sample size, bias, etc.

Reminder: Add commands that generated the plots and how statistics are computed.

Task 2: High variability (on-line sampling)

This task attempts to demonstrate the effects of high variability in network measurements by estimating means with ***on-line sampling***, i.e., as "real-time" data; the previous task is focused on off-line estimation, which is obtained by a stored data set. On the other hand, high variability can, for example, make them unpredictable in the long term.

Download the file `flows.txt` which contains once more values of flow lengths in packets and in bytes captured from a network.

Complete the following action points:

1. *Original data*

- Compute the mean and median for both packets and bytes
- Plot the data set according to what you want to describe (there is no single correct plot)

2. *On-line measurement*

   - First of all, *develop a function called* ***`running_mean`*** *to calculate* $mean_n$, that is, the sample means of the first n flow lengths in bytes. Thereby, the function writes y-axis, as the sample mean values, and x-axis, as the number of flows passed.

   Hint: *For example, there are 6 flows (*flow1*,* flow2*,* flow3*, and so on), and if* n* is 3, i.e., to calculate the sample means of the first 3 flow lengths writting the axes.*

   - The first sample mean is obtained considering the first flow (*flow1*)
   - The second sample mean is obtained by the first 2 flows (*flow1*, *flow2*)
   - The third sample mean is obtained by the first 3 flows (*flow1*, *flow2*, *flow3*)

     Note: This mimics a kind of an on-line measurement; we assume that the flows depart one by one and our estimate of the mean flow size in bytes is updated each time

   - Using the `running_mean`, plot the mean estimate after each flow, i.e., *plot the mean statistic for first observations as a function of n.* Explain your observations concerning the original data and this scenario.

   - Suppose that the interesting statistic is the *median instead of the mean* as `running_median` in an online scenario where a measurement system provides you with a large number of samples every second. How would you proceed in the function to calculate median_n?

Draw your conclusion about the mean and median obtained and plots generated by both the online scenario and the original data set.

Report, task 2

- Plots and values from point 1.
- Expression for `running_mean`.
- Plot of the mean estimate. Explain your observations.
- Computing median instead of mean. Derive expression.
- Observations with the results obtained by both scenarios.

Reminder: Document operations and reason your answers.

Task 3: Data pre-processing for ML purposes

The purpose of the last task is to introduce the preparing data set before choosing a model or even training. During this stage, it is important to select the samples appropriately, one of the techniques is called ***stratified random sampling***, where the population data is divided into subgroups, known as strata, so that a specific number of samples are selected from those subgroups ensuring a balance of information for each subgroup based on the specific feature(s) (reducing selection bias and chances of sampling error as well as higher accuracy than *simple random sampling*).

> **Note:** Data pre-processing is the most important step in most machine learning procedures. Not having the data in suitable form would increase the learning time or it would simply be impossible to learn for the ML model.

Download the file `simple_flow_data.csv` which contains simplified NetMate output which only 6 columns: *source IP address*, *source port*, *destination IP address*, *destination port*, *protocol number*, and *duration of the flow (in micro-seconds)*.

> **Notes:** - The file can be found in the directory `/work/courses/unix/T/ELEC/E7130/general/ml-data` or using `MLDATA` environment variable as a path if you have sourced the `use.sh` file. - Important to consider source and destination IP addresses as *non-numerical* values, the rest are *numerical* values.

Perform a function to **prepare the whole data set** through the steps below. Furthermore, you can use skeleton code `skeleton_ml_0.py` to solve task.

1. Delete the instances that have empty values
2. Perform *stratified random sampling* where:
   - First, take 100 instances whose flow duration is less than 2000 microseconds.
   - Then, take other 100 instances whose flow duration is more than 2000 microseconds.
   - Finally, concatenate both to have 200 data samples in total.
3. Encode the *non-numerical* values, i.e., *srcip* and *dstip*.
4. Standardize the values
5. Normalize the values between 0 and 1
6. Return the new data set pre-processed.

   Note: At the end, the data set must contain 200 instances, and it would look something like the following (rows were shuffled here):

   ```
   srcip   srcport     dstip   dstport   proto  duration
   ```

```
109  0.500000  0.835867  0.242857  0.006227  0.3125  0.196619
115  0.500000  0.547144  0.628571  0.000431  0.3125  0.193576
181  0.142857  0.287867  0.157143  0.278581  1.0000  0.964189
87   0.500000  0.751349  0.171429  0.159641  0.3125  0.000003
163  0.500000  0.616890  0.542857  0.006227  0.3125  0.098573
```

Answer the following points:

1. Mention three types of probability sampling applied in ML apart from the one already mentioned.
2. What is the purpose of encoding the values in ML?
3. What are the differences between standardization and normalization in terms of *feature scaling* in ML?

   Tips: - Useful Python functions could be `fit_transform()`. - Search for the documentation of the functions `LabelEncoder()`, `StandardScaler()`, `MinMaxScaler()` to perform the steps above in case of the library `scikit-learn`.

Report, task 3

- Perform successfully the data pre-processing.
- Answer the questions appropriately.

Reminder: Document operations and code used.

Grading standard

To pass this course, you need to achieve at least 15 points in this assignment. And if you submit the assignment late, you can get a maximum of 15 points.

You can get up to 30 points for this assignment:

Task 1

- Draw a histogram and QQ plot for **each** sample, and calculate its mean, standard deviation and variance. (12p)
- Discussion based on the results obtained previously. (4p)

Task 2

- Plot and calculate the average and median as required (original data). (2p)
- Write the correct running mean expression. (1p)
- Plot the estimate of the mean and state your observations. (2p)
- Write the correct running median expression and plot. (2p)
- Conclusions about the results obtained of mean and median (1p)

Task 3 - Prepare data set for Machine Learning purposes (2p) - Answer the questions appropriately (4p)

The quality of the report (bonus 2p)

The instruction of assignment

For the assignment, your submission must contain (Please don't contain original data in your submission):

- A zip file that includes your codes and scripts.
- A PDF file as your report.

Regarding the report, your report must have:

- A cover page indicating your name, student ID and your e-mail address.
- The report should include a description of measurements, a summary of the results and conclusions based on the results.
- An explanation of each problem, explain how you solved it and why you did it.