Optimization and Simulation

Statistical analysis and bootstrapping

Michel Bierlaire

Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne

EPFL

Introduction

- The outputs of the simulator are random variables.
- Running the simulator provides one realization of these r.v.
- We have no access to the pdf or CDF of these r.v.
- Well... this is actually why we rely on simulation.
- How to derive statistics about a r.v. when only instances are known?
- How to measure the quality of this statistic?

Sample mean and variance

- Consider $X_1, \ldots, X_n$ i.i.d. r.v.
- $\mathrm{E}[X_i] = \mu$, $\mathrm{Var}(X_i) = \sigma^2$.

The sample mean

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

is an unbiased estimate of the population mean μ, as $\mathrm{E}[\bar{X}] = \mu$.

The sample variance

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2$$

is an unbiased estimator of the population variance σ^2, as $\mathrm{E}[S^2] = \sigma^2$. (see proof: Ross, chapter 7)

Sample mean and variance

Recursive computation (Welford's algorithm)

1. Initialize: $\bar{X}_1 = X_1,\ M_2 = 0.$
2. For $k = 2, \ldots, n$:

$$\delta = X_k - \bar{X}_{k-1}, \qquad \bar{X}_k = \bar{X}_{k-1} + \frac{\delta}{k},$$

$$M_2 = M_2 + \delta(X_k - \bar{X}_k).$$

3. Sample variance:

$$S^2 = \frac{M_2}{n-1}.$$

Mean Square Error

- Consider $X_1, \ldots, X_n$ i.i.d. r.v. with CDF F.
- Consider a parameter $\theta(F)$ of the distribution (mean, quantile, mode, etc.)
- Consider $\widehat{\theta}(X_1, \ldots, X_n)$ an estimator of $\theta(F)$.
- The Mean Square Error of the estimator is defined as

$$\mathrm{MSE}(F) = \mathrm{E}_F \left[\left(\widehat{\theta}(X_1, \ldots, X_n) - \theta(F) \right)^2 \right],$$

  where E_F emphasizes that the expectation is taken under the assumption that the r.v. all have distribution F.
- If F is unknown, it is not immediate to find an estimator of MSE.

How many draws must be used?

- Let X a r.v. with mean θ and variance σ^2.
- We want to estimate the mean θ of the simulated distribution.
- The estimator used is the sample mean: $\bar{X}$.
- The mean square error is

$$E[(\bar{X} - \theta)^2] = \frac{\sigma^2}{n}$$

- The sample mean $\bar{X}$ is normally distributed with mean θ and variance σ^2/n.
- So we can stop generating data when $\sigma/\sqrt{n}$ is small.
- σ^2 is approximated by the sample variance S.
- Law of large numbers: at least 100 draws (say) should be used.
- See Ross p. 121 for details.

How many draws are enough? — When theory applies

Analytical stopping rule

For the sample mean $\bar{X}$:

$$\mathrm{MSE}(\bar{X}) = \frac{\sigma^2}{n}$$

so precision improves at rate $1/\sqrt{n}$.

Required assumptions

- Indicator is the sample mean
- Draws are i.i.d.
- Finite variance σ^2

Takeaway

This gives a principled stopping rule — but only in this narrow setting.

Beyond the mean: when theory breaks down

Typical simulation indicators

- Quantiles (e.g. 95% travel time)
- Maxima or tail probabilities
- Nonlinear performance metrics

Problem

- Closed-form MSE is unavailable
- Asymptotic approximations may be unreliable
- No universal rule for choosing n

Solution

Use simulation itself to assess estimator quality: bootstrapping.

Empirical distribution function

- Consider $X_1, \ldots, X_n$ i.i.d. r.v. with CDF F.
- Consider a realization $x_1, \ldots, x_n$ of these r.v.
- The empirical distribution function is defined as

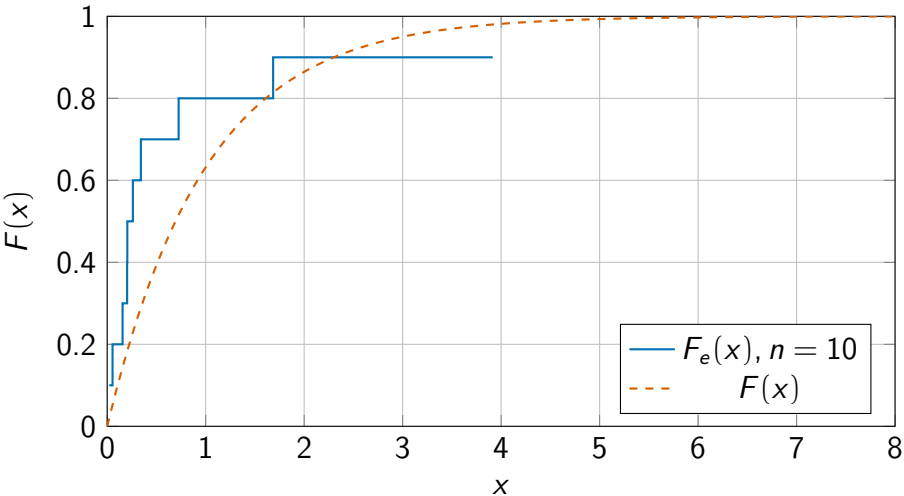
$$F_e(x) = \frac{1}{n} \sum_{i=1}^{n} I\{x_i \leqslant x\},$$

where

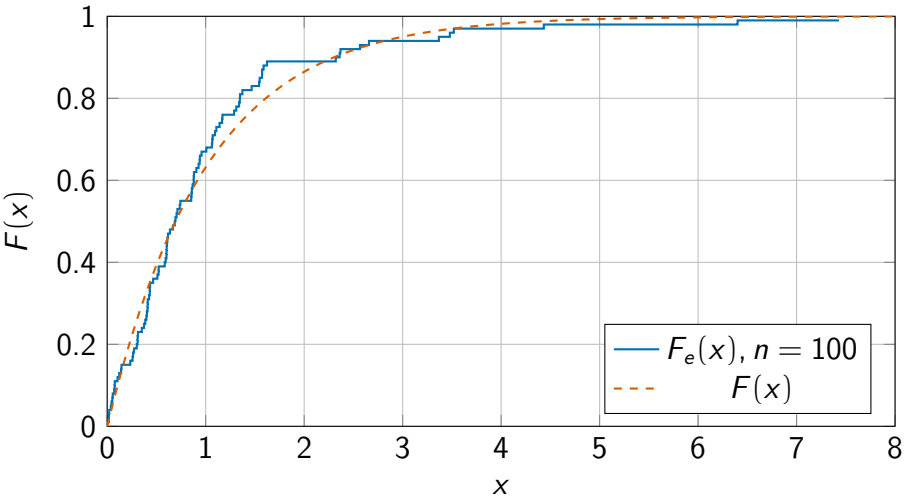
$$I\{x_i \leqslant x\} = \begin{cases} 1 & \text{if } x_i \leqslant x, \\ 0 & \text{otherwise.} \end{cases}$$

- CDF of a r.v. that can take any x_i with equal probability.

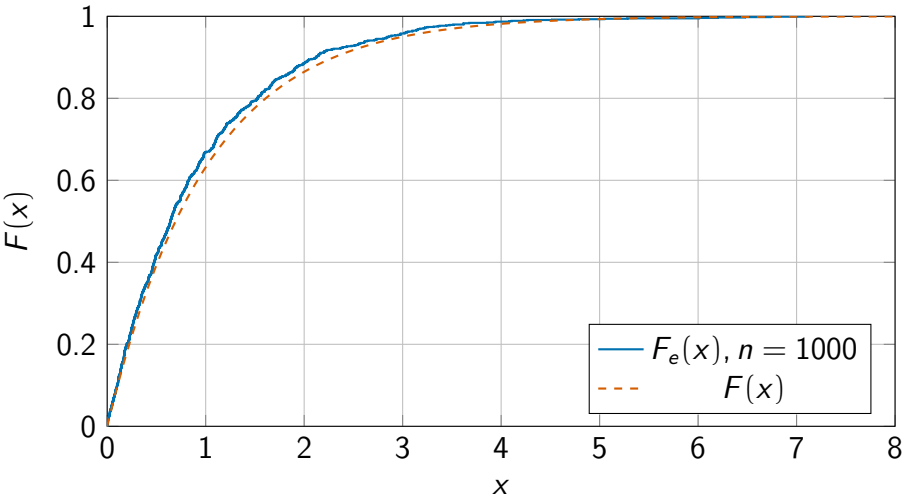
Empirical CDF



Empirical CDF



Empirical CDF



From reality to data

	Reality	Data
Random variable	X	X_e
CDF	F	F_e
True parameter	$\theta(F)$	$\theta(F_e)$
Sample	$X_1, \ldots, X_n \sim F$	$X_1^e, \ldots, X_n^e \sim F_e$
Estimate	$\widehat{\theta}(X_1, \ldots, X_n)$	$\widehat{\theta}(X_1^e, \ldots, X_n^e)$

Mean Square Error

- We use the empirical distribution function F_e
- We can approximate

$$\mathrm{MSE}(F) = \mathrm{E}_F\left[\left(\widehat{\theta}(X_1, \ldots, X_n) - \theta(F)\right)^2\right],$$

by

$$\mathrm{MSE}(F_e) = \mathrm{E}_{F_e}\left[\left(\widehat{\theta}(X_1^e, \ldots, X_n^e) - \theta(F_e)\right)^2\right],$$

- $\theta(F_e)$ can be computed directly from the data (mean, variance, etc.)

Mean Square Error

- We want to compute

$$\mathrm{MSE}(F_e) = \mathrm{E}_{F_e}\left[\left(\widehat{\theta}(X_1^e, \ldots, X_n^e) - \theta(F_e)\right)^2\right],$$

- X_i^e are r.v. that can take any x_i with equal probability.
- Therefore,

$$\mathrm{MSE}(F_e) = \frac{1}{n^n}\sum_{i_1=1}^{n}\cdots\sum_{i_n=1}^{n}\left[\left(\widehat{\theta}(x_{i_1}, \ldots, x_{i_n}) - \theta(F_e)\right)^2\right],$$

- Clearly impossible to compute when n is large.
- Solution: simulation.

Bootstrapping

- For $r = 1, \ldots, R$
- Draw $x_1^r, \ldots, x_n^r$ from F_e, that is draw from the data:
  1. Let s be a draw from $U[0, 1]$
  2. Set $j = \text{floor}(ns)$.
  3. Return x_j.
- Compute

$$M_r = \left(\widehat{\theta}(x_1^r, \ldots, x_n^r) - \theta(F_e)\right)^2,$$

- Estimate of $\text{MSE}(F_e)$ and, therefore, of $\text{MSE}(F)$:

$$\frac{1}{R} \sum_{r=1}^{R} M_r$$

- Typical value for R: 100.

Bootstrap: simple example

- Data: 0.636, -0.643, 0.183, -1.67, 0.462
- Mean= -0.206
- MSE= $\mathrm{E}[(\bar{X} - \theta)^2] = S^2/n = 0.1817$

r						$\hat{\theta}$	$\theta(F_e)$	MSE
1	-0.643	-0.643	-0.643	0.462	0.462	-0.201	-0.206	2.544e-05
2	-0.643	0.183	0.636	0.636	0.636	0.2896	-0.206	0.2456
3	-1.67	-1.67	0.183	0.462	0.636	-0.411	-0.206	0.04204
4	-1.67	-0.643	0.183	0.183	0.636	-0.2617	-0.206	0.003105
5	-0.643	0.462	0.462	0.636	0.636	0.3105	-0.206	0.2667
6	-1.67	-1.67	0.183	0.183	0.183	-0.5573	-0.206	0.1234
7	-0.643	0.183	0.183	0.462	0.636	0.1642	-0.206	0.137
8	-1.67	-1.67	-0.643	0.183	0.183	-0.7225	-0.206	0.2667
9	0.183	0.462	0.462	0.636	0.636	0.4756	-0.206	0.4646
10	-0.643	0.183	0.183	0.462	0.636	0.1642	-0.206	0.137
								0.1686

Python code

```
def calculate_quantile(data: np.ndarray, q: float) -> float:
    return float(np.quantile(data, q))

def bootstrap_sample(data):
    return np.random.choice(data, size=len(data), replace=True)
```

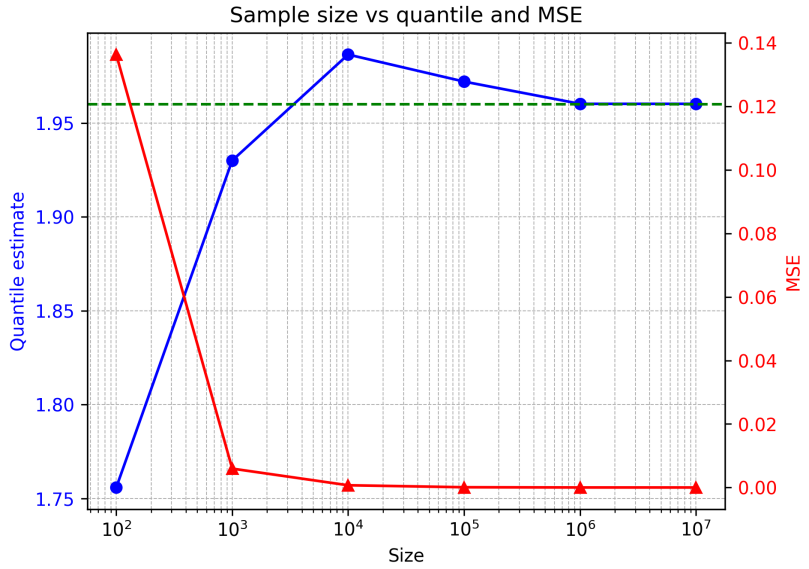
MSE for the percentile: Python code

```
def mean_squared_error_quantile(
  data: np.ndarray,
  q: float,
  num_bootstrap: int = 100,
) -> tuple[float, float]:
  true_quantile = calculate_quantile(data, q)
  bootstrap_estimates = np.array(
    [
      calculate_quantile(bootstrap_sample(data), q)
      for _ in range(num_bootstrap)
    ]
  )
  mse = float(np.mean((bootstrap_estimates - true_quantile) ** 2))
  return true_quantile, mse
```

Experiments

```
quantile_threshold = 0.975
MAX_SAMPLE_SIZE = 7
results = [
  (10**power, mean_squared_error_quantile(
    data[0 : 10**power],
    q=quantile_threshold)
  )
  for power in range(2, MAX_SAMPLE_SIZE + 1)
]
```

Results


Sample size vs quantile and MSE
Quantile estimate
MSE
Size
1.75
1.80
1.85
1.90
1.95
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
10^2
10^3
10^4
10^5
10^6
10^7

Bootstrap confidence intervals

Idea

Bootstrapping approximates the sampling distribution of an estimator $\widehat{\theta}$.

Percentile method

- Generate bootstrap estimates $\widehat{\theta}^1, \ldots, \widehat{\theta}^R$.
- A 95% confidence interval is given by the 2.5% and 97.5% quantiles of this empirical distribution.

Remark

No parametric assumption on the distribution of $\widehat{\theta}$ is required.

Summary

- The number of draws is determined by the required precision.
- In some cases, the precision is derived from theoretical results.
- If not, rely on bootstrapping.
- Idea: use simulation to estimate the Mean Square Error.

Appendix: MSE for the mean

- Consider $X_1, \ldots, X_n$ i.i.d. r.v.
- Denote $\theta = \mathrm{E}[X_i]$ and $\sigma^2 = \mathrm{Var}(X_i)$.
- Consider $\bar{X} = \sum_{i=1}^{n} X_i/n$.
- $\mathrm{E}[\bar{X}] = \sum_{i=1}^{n} \mathrm{E}[X_i]/n = \theta$.
- MSE:

$$\begin{aligned} \mathrm{E}[(\bar{X} - \theta)^2] &= \mathrm{Var}\,\bar{X} \\ &= \mathrm{Var}\left(\sum_{i=1}^{n} X_i/n\right) \\ &= \sum_{i=1}^{n} \mathrm{Var}(X_i)/n^2 \\ &= \sigma^2/n. \end{aligned}$$

Appendix: why Welford's update is equivalent

Invariant

Define

$$M_{2,k} = \sum_{i=1}^{k} (X_i - \bar{X}_k)^2 \quad \Rightarrow \quad S_k^2 = \frac{M_{2,k}}{k-1}.$$

Appendix: why Welford's update is equivalent

Let $\delta_k = X_k - \bar{X}_{k-1}$ and $\bar{X}_k = \bar{X}_{k-1} + \delta_k/k$. Then

$$M_{2,k} = \sum_{i=1}^{k-1} (X_i - \bar{X}_k)^2 + (X_k - \bar{X}_k)^2.$$

For $i \leqslant k-1$, write $X_i - \bar{X}_k = (X_i - \bar{X}_{k-1}) - (\bar{X}_k - \bar{X}_{k-1})$ and use $\sum_{i=1}^{k-1}(X_i - \bar{X}_{k-1}) = 0$ to obtain

$$\sum_{i=1}^{k-1} (X_i - \bar{X}_k)^2 = M_{2,k-1} + (k-1)(\bar{X}_k - \bar{X}_{k-1})^2.$$

Also, $X_k - \bar{X}_k = \delta_k - (\bar{X}_k - \bar{X}_{k-1})$ and $\bar{X}_k - \bar{X}_{k-1} = \delta_k/k$, hence

$$(k-1)\Big(\frac{\delta_k}{k}\Big)^2 + \Big(\delta_k - \frac{\delta_k}{k}\Big)^2 = \delta_k^2\Big(1 - \frac{1}{k}\Big) = \delta_k(X_k - \bar{X}_k).$$

Therefore,

$$M_{2,k} = M_{2,k-1} + \delta_k(X_k - \bar{X}_k),$$

which is exactly Welford's update.