Stochastic Simulation

Jan-Pieter Dorsman & Michel Mandjes

Korteweg-de Vries Institute for Mathematics, University of Amsterdam

University of Amsterdam,
Fall, 2018

This course

- There will be 12 classes.

This course

- There will be 12 classes.
- The book that will be used is *Stochastic Simulation*, by S. Asmussen and P. Glynn (Springer, 2007). In addition, we have lecture notes on discrete-event simulation, based on a manuscript by H. Tijms.

This course

- There will be 12 classes.
- The book that will be used is *Stochastic Simulation*, by S. Asmussen and P. Glynn (Springer, 2007). In addition, we have lecture notes on discrete-event simulation, based on a manuscript by H. Tijms.
- Homework (40%): see `https://staff.fnwi.uva.nl/j.l.dorsman/stochSim1819.html`

This course

- There will be 12 classes.
- The book that will be used is *Stochastic Simulation*, by S. Asmussen and P. Glynn (Springer, 2007). In addition, we have lecture notes on discrete-event simulation, based on a manuscript by H. Tijms.
- Homework (40%): see `https://staff.fnwi.uva.nl/j.l.dorsman/stochSim1819.html`
- Exam (60%): written or oral, depending on number of students.

Chapter I
WHAT THIS COURSE IS ABOUT

What is this course about?

- Complex stochastic dynamical systems

What is this course about?

- Complex stochastic dynamical systems of the *discrete event type*.

What is this course about?

- Complex stochastic dynamical systems of the *discrete event type*.
- Focus on systems in which no explicit results for performance measure are available.

What is this course about?

- Complex stochastic dynamical systems of the *discrete event type*.
- Focus on systems in which no explicit results for performance measure are available.
- System is *simulated*. After repeating this experiment (to obtain the output of multiple *runs*), statistical claims can be made about value of performance measure.

What is this course about?

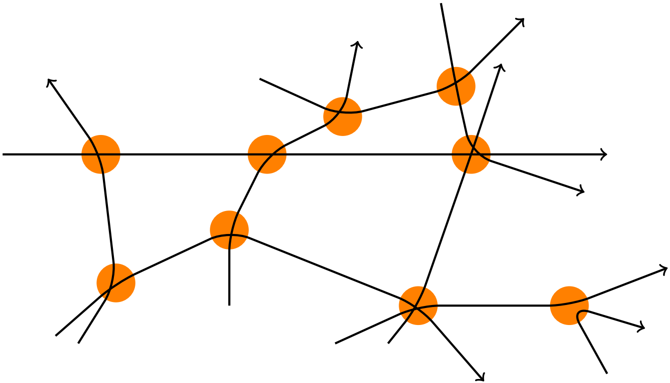
- Complex stochastic dynamical systems of the *discrete event type*.
- Focus on systems in which no explicit results for performance measure are available.
- System is *simulated*. After repeating this experiment (to obtain the output of multiple *runs*), statistical claims can be made about value of performance measure.
- I start with two motivating examples.

What is this course about?

Two motivating examples:

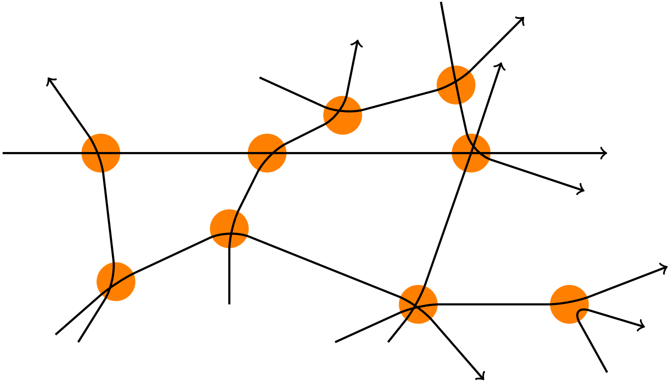
- (Networks of) queueing systems.
- (Multivariate) ruin models.

(Networks of) queueing systems



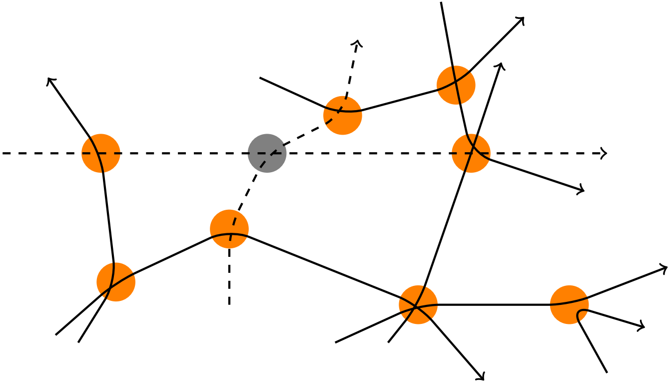
(Networks of) queueing systems

Complication: *link failures*



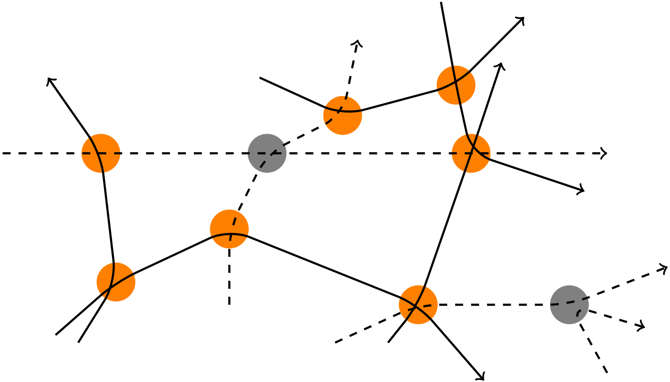
(Networks of) queueing systems

Complication: *link failures*



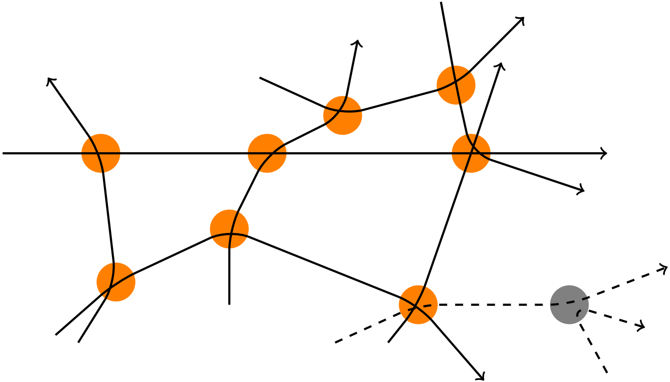
(Networks of) queueing systems

Complication: *link failures*



(Networks of) queueing systems

Complication: *link failures*



(Networks of) queueing systems

- particles ('customers' in queueing lingo) move through a network;
- arrival rates and service rates are affected by an external process ('background process') — for instance to model link failures, or other 'irregularities';
- queues are 'coupled' because they fact to *common* background process.
- resulting processes applicable to communication networks, road traffic networks, chemistry, economics.
- in full generality, network way too complex to allow explicit analysis — simulation comes in handy!

(Multivariate) ruin models

Classical ruin model: reserve of insurance company given by

$$X_t := X_0 + ct - \sum_{n=0}^{N_t} B_n,$$

with c premium rate, N_t a Poisson process (rate λ), B_n i.i.d. claim sizes.

(Multivariate) ruin models

Classical ruin model: reserve of insurance company given by

$$X_t := X_0 + ct - \sum_{n=0}^{N_t} B_n,$$

with c premium rate, N_t a Poisson process (rate λ), B_n i.i.d. claim sizes.

Lots is known about this model. Goal:

$$\mathbb{P}\left(\exists t \leqslant T : X_t < 0\right).$$

(Multivariate) ruin models

Less is known if there are multiple insurance companies with correlated claims:

$$X_t := X_0 + c_X t - \sum_{n=0}^{N_t^{(X)}} B_n^{(X)},$$

$$Y_t := Y_0 + c_Y t - \sum_{n=0}^{N_t^{(Y)}} B_n^{(Y)}.$$

(Multivariate) ruin models

Less is known if there are multiple insurance companies with correlated claims:

$$X_t := X_0 + c_X t - \sum_{n=0}^{N_t^{(X)}} B_n^{(X)},$$

$$Y_t := Y_0 + c_Y t - \sum_{n=0}^{N_t^{(Y)}} B_n^{(Y)}.$$

Here:

- correlated claim size sequences $B_n^{(X)}$ and $B_n^{(Y)}$ (can be achieved by e.g., letting the claims depend on a common background process),
- $N_t^{(X)}$: Poisson process with rate $\lambda^{(X)}$, and $N_t^{(Y)}$ Poisson process with rate $\lambda^{(Y)}$.

(Multivariate) ruin models

Goal is to compute (both go bankrupt before time T):

$$\mathbb{P}\left(\exists s, t \leqslant T : \{X_s < 0\} \cap \{Y_t < 0\}\right).$$

(Multivariate) ruin models

Goal is to compute (both go bankrupt before time T):

$$\mathbb{P}\left(\exists s, t \leqslant T : \{X_s < 0\} \cap \{Y_t < 0\}\right).$$

Or (one of them goes bankrupt before time T):

$$\mathbb{P}\left(\exists s, t \leqslant T : \{X_s < 0\} \cup \{Y_t < 0\}\right).$$

(Multivariate) ruin models

Goal is to compute (both go bankrupt before time T):

$$\mathbb{P}\left(\exists s, t \leqslant T : \{X_s < 0\} \cap \{Y_t < 0\}\right).$$

Or (one of them goes bankrupt before time T):

$$\mathbb{P}\left(\exists s, t \leqslant T : \{X_s < 0\} \cup \{Y_t < 0\}\right).$$

Not known how to analyze this. Again: simulation comes in handy!

When is simulation a viable approach?

- In situations in which neither explicit results are known, nor alternative numerical approaches are viable. 'Monte Carlo' ($\approx$ stochastic simulation) is often used in financial industry and in engineering.
- In research, to validate conjectures of exact or asymptotic results, or to assess the accuracy of approximations.

Issues to be dealt with

[§I.4 of A & G]

To set up a simulation in a concrete context, various (conceptual, practical) issues have to be dealt with.

I elaborate on the issues mentioned by A & G, and I'll add a few.

Issue 1: proper generation of random objects

How do we generate the needed input random variables?
Optimally: we generate an *exact sample* of the random object (variable, process) under study.

Issue 2: number of runs needed

One replicates a simulation experiment N times, and based on the output the performance metric under consideration is estimated. How large should N be to obtain an estimate with a given precision?

Issue 3: estimation of stationary performance measures

This is about estimation of stationary performance measures. Typically one starts the process at an initial state, and after a while the process tends to equilibrium.
But how do we know when we 'are' in equilibrium? Or are there other (clever) ways to estimate stationary performance measures?

Issue 4: exploitation of problem structure

In many situations one could follow the naïve approach of simulating the stochastic process at hand at a fine time grid, but often it suffices to consider a certain embedded process.

Issue 5: rare-event probabilities

In many applications extremely small probabilities are relevant. For instance: probability of ruin of an insurance firm; probability of an excessively high number of customers in a queue. Direct simulation is time consuming (as one does not 'see' rare event under consideration). Specific rare-event-oriented techniques are needed.

Issue 6: parameter sensitivity

A standard simulation experiment provides an estimate of a performance measure for a given set of parameters. Often one is interested in the sensitivity of the performance with respect to changes of those parameters. Can simulation methods be designed that are capable of this?

Issue 7: simulation-based optimization

A standard simulation experiment provides an estimate of a performance measure for a given set of parameters. Can simulation be used (and if yes, how) to optimize a given objective function?

Issue 8: continuous-time systems

This course is primarily on the simulation of dynamic stochastic systems that allow a discrete-event setup. What can be done if processes in continuous-time are to be simulated?

Issue 9: programming environment

Various packages are available, some really intended to do simulations with, others have a more general purpose. Which packages are good for which purposes?

Issue 10: data structure

For a given simulation experiment, what is the best data structure to store the system's key quantities?

Chapter II
GENERATING RANDOM OBJECTS

Generating random objects

Generating random objects

- Generating uniformly distributed random variables

Generating random objects

- Generating uniformly distributed random variables
- Generating generally distributed random variables

Generating random objects

- Generating uniformly distributed random variables
- Generating generally distributed random variables
- Generating random vectors

Generating random objects

- Generating uniformly distributed random variables
- Generating generally distributed random variables
- Generating random vectors
- Generating elementary stochastic processes

Generating uniform numbers

[§II.1 of A & G]

We want to develop machine to generate i.i.d. numbers $U_1, U_2, \dots$ that are uniformly distributed on $[0, 1]$.

Generating uniform numbers

[§II.1 of A & G]

We want to develop machine to generate i.i.d. numbers $U_1, U_2, \ldots$ that are uniformly distributed on $[0, 1]$.

As we will see: building block for nearly any simulation experiment.

Generating uniform numbers

Physical devices:

Generating uniform numbers

Physical devices:

Classical approach: time between emission of particles by radioactive source is essentially exponential. Say the mean of this exponential time T is λ^{-1}, then $e^{-\lambda T}$ has a Uniform distribution on the interval $[0, 1]$.

Generating uniform numbers

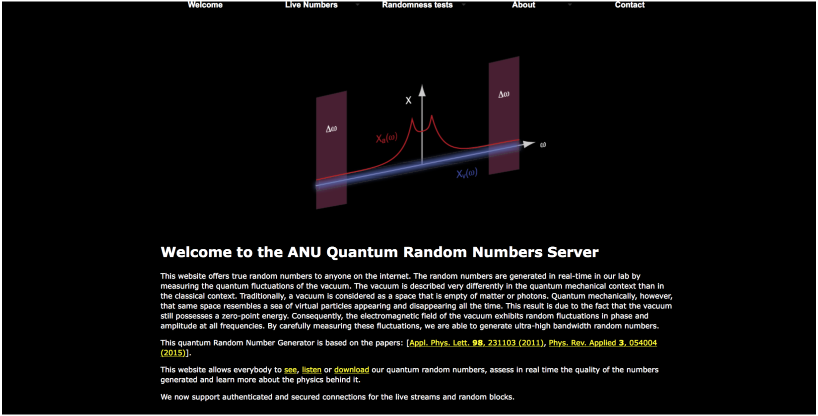
Physical devices:

Classical approach: time between emission of particles by radioactive source is essentially exponential. Say the mean of this exponential time T is λ^{-1}, then $e^{-\lambda T}$ has a Uniform distribution on the interval $[0, 1]$.

[Check!]

Generating uniform numbers

Physical devices:



Generating uniform numbers

Alternative: deterministic recursive algorithms (also known as: *pseudo-random number generators*).

Generating uniform numbers

Alternative: deterministic recursive algorithms (also known as: *pseudo-random number generators*).

Typical form: with $A, C, M \in \mathbb{N}$,

$$u_n := \frac{s_n}{M}, \quad \text{where } \; s_{n+1} := (As_n + C) \mod M.$$

Generating uniform numbers

Alternative: deterministic recursive algorithms (also known as: *pseudo-random number generators*).

Typical form: with $A, C, M \in \mathbb{N}$,

$$u_n := \frac{s_n}{M}, \quad \text{where} \;\; s_{n+1} := (As_n + C) \bmod M.$$

It is the art to make the period of the recursion as long as possible. Early random number generators work with $M = 2^{31} - 1$, $A = 7^5$, and $C = 0$.

Generating uniform numbers

Alternative: deterministic recursive algorithms (also known as: *pseudo-random number generators*).

Typical form: with $A, C, M \in \mathbb{N}$,

$$u_n := \frac{s_n}{M}, \quad \text{where} \;\; s_{n+1} := (As_n + C) \bmod M.$$

It is the art to make the period of the recursion as long as possible. Early random number generators work with $M = 2^{31} - 1$, $A = 7^5$, and $C = 0$.

By now, way more sophisticated algorithms have been implemented. In all standard software packages (Matlab, Mathematica, R) implementations are used that yield nearly i.i.d. uniform numbers.

Generating uniform numbers

All kinds of tests have been proposed to test a random number generator's performance.

Clearly not sufficient to show that the samples stem from a uniform distribution on $[0, 1]$,

Generating uniform numbers

All kinds of tests have been proposed to test a random number generator's performance.

Clearly not sufficient to show that the samples stem from a uniform distribution on $[0, 1]$, as also the *independence* is important.

Test of uniformity can be done by bunch of standard tests: χ^2, Kolmogorov-Smirnov, etc. Test on independence by *run test*.

Generating generally distributed numbers

[§II.2 of A & G]

What is the game?

Generating generally distributed numbers

[§II.2 of A & G]

What is the game? Suppose I provide you with a machine that can spit out i.i.d. uniform numbers (on $[0, 1]$, that is).

Generating generally distributed numbers

[§II.2 of A & G]

What is the game? Suppose I provide you with a machine that can spit out i.i.d. uniform numbers (on $[0, 1]$, that is). Can you give me a sample from a general one-dimensional distribution?

Generating generally distributed numbers

Discrete random variables	Continuous random variables
Binomial	Normal, Lognormal
Geometric	Exponential
Neg. binomial	Gamma (and Erlang)
Poisson	Weibull, Pareto

Generating generally distributed numbers

Two simple examples:

Generating generally distributed numbers

Two simple examples:

- Let X be binomial with parameters n and p. Let $X_i := 1$ if $U_i < p$ and 0 else. Then $\sum_{i=1}^{n} X_i$ has desired distribution.

Generating generally distributed numbers

Two simple examples:

- Let X be binomial with parameters n and p. Let $X_i := 1$ if $U_i < p$ and 0 else. Then $\sum_{i=1}^{n} X_i$ has desired distribution.
- Let X be geometric with parameter p. Sample U_i until $U_i < p$; let us say this happens at the N-th attempt. Then N has desired distribution.

Generating generally distributed numbers

Two simple examples:

- Let X be binomial with parameters n and p. Let $X_i := 1$ if $U_i < p$ and 0 else. Then $\sum_{i=1}^{n} X_i$ has desired distribution.
- Let X be geometric with parameter p. Sample U_i until $U_i < p$; let us say this happens at the N-th attempt. Then N has desired distribution.

The negative binomial distributed distribution can be sampled similarly. [How?]

Generating generally distributed numbers

Discrete random variables	Continuous random variables
Binomial ✓	Normal, Lognormal
Geometric ✓	Exponential
Neg. binomial ✓	Gamma (and Erlang)
Poisson	Weibull, Pareto

Generating generally distributed numbers

Easiest case: random variables with finite support (say X).

Generating generally distributed numbers

Easiest case: random variables with finite support (say X). Say, it lives on $\{i_1, \ldots, i_N\}$ (ordered), with respective probabilities $p_1, \ldots, p_N$ that sum to 1.

Generating generally distributed numbers

Easiest case: random variables with finite support (say X). Say, it lives on $\{i_1, \ldots, i_N\}$ (ordered), with respective probabilities $p_1, \ldots, p_N$ that sum to 1.

Define $\bar{p}_i := \sum_{j=1}^{i} p_j$ (with $\bar{p}_0 := 0$); observe that $\bar{p}_N = 1$.
Let U be uniform on $[0, 1]$. Then

$$X := i_j \ \text{ if } \ U \in [\bar{p}_{j-1}, \bar{p}_j).$$

Proof:

$$\mathbb{P}(X = i_j) = \bar{p}_j - \bar{p}_{j-1} = p_j,$$

as desired.

Generating generally distributed numbers

Easiest case: random variables with finite support (say X). Say, it lives on $\{i_1, \ldots, i_N\}$ (ordered), with respective probabilities $p_1, \ldots, p_N$ that sum to 1.

Define $\bar{p}_i := \sum_{j=1}^{i} p_j$ (with $\bar{p}_0 := 0$); observe that $\bar{p}_N = 1$. Let U be uniform on $[0, 1]$. Then

$$X := i_j \;\text{ if }\; U \in [\bar{p}_{j-1}, \bar{p}_j).$$

Proof:

$$\mathbb{P}(X = i_j) = \bar{p}_j - \bar{p}_{j-1} = p_j,$$

as desired. Procedure easily extended to case of countable support.

Generating generally distributed numbers

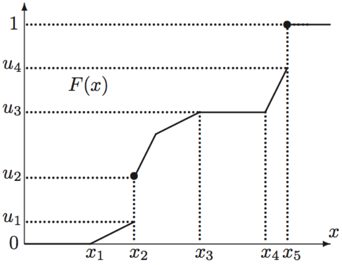
This algorithm is a simple form of *inversion*. Relies on the left-continuous version of the inverse distribution function: with $F(\cdot)$ the distribution function of the random variable X,

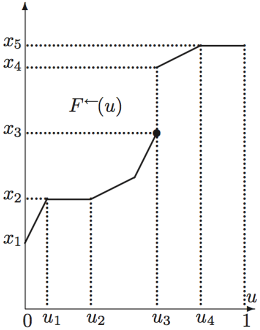
$$F^{\leftarrow}(u) := \min\{x : F(x) \geqslant u\}.$$

Generating generally distributed numbers

This algorithm is a simple form of *inversion*. Relies on the left-continuous version of the inverse distribution function: with $F(\cdot)$ the distribution function of the random variable X,

$$F^{\leftarrow}(u) := \min\{x : F(x) \geqslant u\}.$$

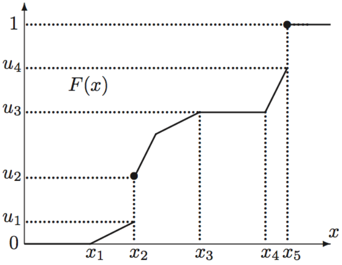


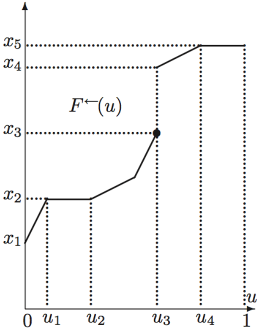


Generating generally distributed numbers

This algorithm is a simple form of *inversion*. Relies on the left-continuous version of the inverse distribution function: with $F(\cdot)$ the distribution function of the random variable X,

$$F^{\leftarrow}(u) := \min\{x : F(x) \geqslant u\}.$$





$F^{\leftarrow}(u)$ known as quantile function.

Generating generally distributed numbers

Proposition Let U be uniform on $[0, 1]$.
(a) $u \leqslant F(x) \;\Leftrightarrow\; F^{\leftarrow}(u) \leqslant x$.

Generating generally distributed numbers

Proposition Let U be uniform on $[0,1]$.
(a) $u \leqslant F(x) \Leftrightarrow F^{\leftarrow}(u) \leqslant x$.
(b) $F^{\leftarrow}(U)$ has CDF F.

Generating generally distributed numbers

Proposition Let U be uniform on $[0,1]$.
(a) $u \leqslant F(x) \Leftrightarrow F^{\leftarrow}(u) \leqslant x$.
(b) $F^{\leftarrow}(U)$ has CDF F.
(c) if F is continuous, then $F(X) \sim U$.

Generating generally distributed numbers

Proposition Let U be uniform on $[0,1]$.
(a) $u \leqslant F(x) \;\Leftrightarrow\; F^{\leftarrow}(u) \leqslant x$.
(b) $F^{\leftarrow}(U)$ has CDF F.
(c) if F is continuous, then $F(X) \sim U$.

Proof:

Generating generally distributed numbers

Proposition Let U be uniform on $[0,1]$.
(a) $u \leqslant F(x) \;\Leftrightarrow\; F^{\leftarrow}(u) \leqslant x$.
(b) $F^{\leftarrow}(U)$ has CDF F.
(c) if F is continuous, then $F(X) \sim U$.

Proof: Part (a) follows from definitions.

Generating generally distributed numbers

Proposition Let U be uniform on $[0, 1]$.
(a) $u \leqslant F(x) \Leftrightarrow F^{\leftarrow}(u) \leqslant x$.
(b) $F^{\leftarrow}(U)$ has CDF F.
(c) if F is continuous, then $F(X) \sim U$.

Proof: Part (a) follows from definitions. Part (b) follows from (a):

$$\mathbb{P}(F^{\leftarrow}(U) \leqslant x) = \mathbb{P}(U \leqslant F(x)) = F(x).$$

Part (c) follows from (a) and continuity of F:

$$\mathbb{P}(F(X) \geqslant u) = \mathbb{P}(X \geqslant F^{\leftarrow}(u)) = \mathbb{P}(X > F^{\leftarrow}(u)) = 1 - F(F^{\leftarrow}(u)),$$

which equals (as desired) $1 - u$. This is because $F(F^{\leftarrow}(u)) \geqslant u$ from (a); also, by considering sequence y_n such that $y_n \uparrow F^{\leftarrow}(u)$, such that $F(y_n) < u$, and by continuity $F(F^{\leftarrow}(u)) \leqslant u$.

Generating generally distributed numbers

Example: how to generate exponentially distributed numbers (with mean λ^{-1}) from U?

Generating generally distributed numbers

Example: how to generate exponentially distributed numbers (with mean λ^{-1}) from U?

$$F(x) = 1 - e^{-\lambda x} \iff F^{\leftarrow}(u) = -\frac{\log(1-u)}{\lambda}.$$

Generating generally distributed numbers

Example: how to generate exponentially distributed numbers (with mean λ^{-1}) from U?

$$F(x) = 1 - e^{-\lambda x} \;\;\Leftrightarrow\;\; F^{\leftarrow}(u) = -\frac{\log(1-u)}{\lambda}.$$

Proposition yields: $-\log(1-U)/\lambda$ has an $\exp(\lambda)$ distribution.

Generating generally distributed numbers

Example: how to generate exponentially distributed numbers (with mean λ^{-1}) from U?

$$F(x) = 1 - e^{-\lambda x} \ \Leftrightarrow \ F^{\leftarrow}(u) = -\frac{\log(1-u)}{\lambda}.$$

Proposition yields: $-\log(1-U)/\lambda$ has an $\exp(\lambda)$ distribution.

We may as well use $-\log U/\lambda$. [Check!]

Generating generally distributed numbers

Example: how to generate exponentially distributed numbers (with mean λ^{-1}) from U?

$$F(x) = 1 - e^{-\lambda x} \iff F^{\leftarrow}(u) = -\frac{\log(1-u)}{\lambda}.$$

Proposition yields: $-\log(1-U)/\lambda$ has an $\exp(\lambda)$ distribution.

We may as well use $-\log U/\lambda$. [Check!]

Erlang(k, λ) now follows directly as well. [How?]

Generating generally distributed numbers

Discrete random variables	Continuous random variables
Binomial ✓	Normal, Lognormal
Geometric ✓	Exponential ✓
Neg. binomial ✓	Gamma (and Erlang ✓)
Poisson	Weibull, Pareto

Generating generally distributed numbers

Now we also have a fast way to sample a geometric random variable:

$$X := \left\lceil \frac{\log U}{\log p} \right\rceil .$$

Generating generally distributed numbers

Now we also have a fast way to sample a geometric random variable:

$$X := \left\lceil \frac{\log U}{\log p} \right\rceil.$$

Justification:

$$\mathbb{P}\left(\left\lceil \frac{\log U}{\log p} \right\rceil \geqslant k\right) = \mathbb{P}\left(\frac{\log U}{\log p} \geqslant k - 1\right)$$

Generating generally distributed numbers

Now we also have a fast way to sample a geometric random variable:

$$X := \left\lceil \frac{\log U}{\log p} \right\rceil.$$

Justification:

$$\mathbb{P}\left(\left\lceil \frac{\log U}{\log p} \right\rceil \geqslant k\right) = \mathbb{P}\left(\frac{\log U}{\log p} \geqslant k - 1\right),$$

which equals

$$\mathbb{P}(\log U \leqslant (k-1)\log p) = \mathbb{P}(U \leqslant p^{k-1}) = p^{k-1}.$$

Generating generally distributed numbers

Now we also have a fast way to sample a geometric random variable:

$$X := \left\lceil \frac{\log U}{\log p} \right\rceil.$$

Justification:

$$\mathbb{P}\left(\left\lceil \frac{\log U}{\log p} \right\rceil \geqslant k\right) = \mathbb{P}\left(\frac{\log U}{\log p} \geqslant k-1\right),$$

which equals

$$\mathbb{P}(\log U \leqslant (k-1)\log p) = \mathbb{P}(U \leqslant p^{k-1}) = p^{k-1}.$$

Advantage: requires just one random number. Disadvantage: log is slow operator.

Generating generally distributed numbers

Now that we can sample exponential random variable, we can also sample from Poisson distribution.

Generating generally distributed numbers

Now that we can sample exponential random variable, we can also sample from Poisson distribution.
Let $S_0 := 0$ and X_i i.i.d. exponentials.

$$S_n := \sum_{i=1}^{n} X_i.$$

Generating generally distributed numbers

Now that we can sample exponential random variable, we can also sample from Poisson distribution.
Let $S_0 := 0$ and X_i i.i.d. exponentials.

$$S_n := \sum_{i=1}^{n} X_i.$$

Define:

$$N := \max\{n \in \mathbb{N} : S_n \leqslant 1\}.$$

Observe: S_n is the number of arrivals of a Poisson process at time 1, which has a Poisson distribution with mean $\lambda \cdot 1 = \lambda$.

Generating generally distributed numbers

Now that we can sample exponential random variable, we can also sample from Poisson distribution.
Let $S_0 := 0$ and X_i i.i.d. exponentials.

$$S_n := \sum_{i=1}^{n} X_i.$$

Define:

$$N := \max\{n \in \mathbb{N} : S_n \leqslant 1\}.$$

Observe: S_n is the number of arrivals of a Poisson process at time 1, which has a Poisson distribution with mean $\lambda \cdot 1 = \lambda$.

To avoid the log operation, we can also use

$$N := \max\left\{n \in \mathbb{N} : \prod_{i=1}^{n} U_i \geqslant e^{-\lambda}\right\}.$$

Generating generally distributed numbers

Discrete random variables	Continuous random variables
Binomial ✓	Normal, Lognormal
Geometric ✓	Exponential ✓
Neg. binomial ✓	Gamma (and Erlang ✓)
Poisson ✓	Weibull, Pareto

Generating generally distributed numbers

Weibull distribution: for some $\alpha > 0$ and $\lambda > 0$,

$$F(x) = \mathbb{P}(X \leqslant x) = 1 - e^{-\lambda x^{\alpha}}.$$

Generating generally distributed numbers

Weibull distribution: for some $\alpha > 0$ and $\lambda > 0$,

$$F(x) = \mathbb{P}(X \leqslant x) = 1 - e^{-\lambda x^{\alpha}}.$$

Pareto distribution: for some $\alpha > 0$,

$$F(x) = \mathbb{P}(X \leqslant x) = 1 - (x+1)^{-\alpha}.$$

Generating generally distributed numbers

Weibull distribution: for some $\alpha > 0$ and $\lambda > 0$,

$$F(x) = \mathbb{P}(X \leqslant x) = 1 - e^{-\lambda x^{\alpha}}.$$

Pareto distribution: for some $\alpha > 0$,

$$F(x) = \mathbb{P}(X \leqslant x) = 1 - (x + 1)^{-\alpha}.$$

Both can be done with inverse-CDF technique.

Generating generally distributed numbers

Discrete random variables	Continuous random variables
Binomial ✓	Normal, Lognormal
Geometric ✓	Exponential ✓
Neg. binomial ✓	Gamma (and Erlang ✓)
Poisson ✓	Weibull, Pareto ✓

Generating generally distributed numbers

Acceptance-rejection: to be used when (i) no explicit expression for CDF is available, or

Generating generally distributed numbers

Acceptance-rejection: to be used when (i) no explicit expression for CDF is available, or (ii) the CDF does not allow explicit inversion.

Generating generally distributed numbers

Acceptance-rejection: to be used when (i) no explicit expression for CDF is available, or (ii) the CDF does not allow explicit inversion.

Setup:

- X has PDF $f(x)$, and Y has PDF $g(x)$ such that, for some finite constant C and all x,

$$f(x) \leqslant Cg(x).$$

Generating generally distributed numbers

Acceptance-rejection: to be used when (i) no explicit expression for CDF is available, or (ii) the CDF does not allow explicit inversion.

Setup:

- X has PDF $f(x)$, and Y has PDF $g(x)$ such that, for some finite constant C and all x,

$$f(x) \leqslant Cg(x).$$

- We know how to sample Y (but we don't know how to sample X).

Generating generally distributed numbers

Algorithm:

Generating generally distributed numbers

Algorithm:

- Sample Y, and in addition U that is uniform on $[0, 1]$.

Generating generally distributed numbers

Algorithm:

- Sample Y, and in addition U that is uniform on $[0,1]$.
- If
$$U \leqslant \frac{f(Y)}{Cg(Y)},$$
  then $X := Y$, and otherwise you try again.

Generating generally distributed numbers

Algorithm:

- Sample Y, and in addition U that is uniform on $[0,1]$.
- If
$$U \leqslant \frac{f(Y)}{Cg(Y)},$$
then $X := Y$, and otherwise you try again.

Define the event of acceptance by

$$A := \left\{ U \leqslant \frac{f(Y)}{Cg(Y)} \right\}.$$

Generating generally distributed numbers

Proof:

$$\mathbb{P}(X \leqslant x) = \mathbb{P}(Y \leqslant x \,|\, A) = \frac{\mathbb{P}(Y \leqslant x,\, A)}{\mathbb{P}(A)}$$

Generating generally distributed numbers

Proof:

$$\mathbb{P}(X \leqslant x) = \mathbb{P}(Y \leqslant x \,|\, A) = \frac{\mathbb{P}(Y \leqslant x,\, A)}{\mathbb{P}(A)}$$

Denominator:

$$\begin{aligned}\mathbb{P}(A) &= \mathbb{P}\left(U \leqslant \frac{f(Y)}{Cg(Y)}\right) = \mathbb{E}\left(\frac{f(Y)}{Cg(Y)}\right)\\ &= \int_{-\infty}^{\infty} \frac{f(y)}{Cg(y)} g(y)\mathrm{d}y = \int_{-\infty}^{\infty} \frac{f(y)}{C}\mathrm{d}y = \frac{1}{C}.\end{aligned}$$

Here we use that

$$\mathbb{P}(U \leqslant X) = \int \mathbb{P}(U \leqslant x)\mathbb{P}(X \in \mathrm{d}x) = \int x\mathbb{P}(X \in \mathrm{d}x) = \mathbb{E}X.$$

Generating generally distributed numbers

Proof, continued: Numerator can be dealt with similarly.

$$\begin{aligned}\mathbb{P}(Y \leqslant x,\, A) &= \mathbb{P}\left(U \leqslant \frac{f(Y)}{Cg(Y)};\, Y \leqslant x\right) \\ &= \mathbb{E}\left(\frac{f(Y)}{Cg(Y)}\, 1\left\{Y \leqslant x\right\}\right) \\ &= \int_{-\infty}^{x} \frac{f(y)}{Cg(y)} g(y)\mathrm{d}y = \int_{-\infty}^{x} \frac{f(y)}{C}\mathrm{d}y = \frac{F(x)}{C}.\end{aligned}$$

Generating generally distributed numbers

Proof, continued: Numerator can be dealt with similarly.

$$\begin{aligned}\mathbb{P}(Y \leqslant x,\, A) &= \mathbb{P}\left(U \leqslant \frac{f(Y)}{Cg(Y)};\, Y \leqslant x\right)\\ &= \mathbb{E}\left(\frac{f(Y)}{Cg(Y)}\,1\{Y \leqslant x\}\right)\\ &= \int_{-\infty}^{x} \frac{f(y)}{Cg(y)} g(y)\mathrm{d}y = \int_{-\infty}^{x} \frac{f(y)}{C}\mathrm{d}y = \frac{F(x)}{C}.\end{aligned}$$

Ratio: $F(x)$, as desired.

Generating generally distributed numbers

Can be used to sample from Normal distribution. Without loss of generality [Why?] we show how to sample from Normal distribution conditioned on being positive.

Generating generally distributed numbers

Can be used to sample from Normal distribution. Without loss of generality [Why?] we show how to sample from Normal distribution conditioned on being positive.

Has density

$$f(x) = \sqrt{\frac{2}{\pi}} e^{-x^2/2}.$$

We now find a C such that, with $g(x) = e^{-x}$ (exp. distr. with mean 1!),

$$f(x) \leqslant Cg(x).$$

Generating generally distributed numbers

Can be used to sample from Normal distribution. Without loss of generality [Why?] we show how to sample from Normal distribution conditioned on being positive.

Has density

$$f(x) = \sqrt{\frac{2}{\pi}} e^{-x^2/2}.$$

We now find a C such that, with $g(x) = e^{-x}$ (exp. distr. with mean 1!),

$$f(x) \leqslant Cg(x).$$

Observe that we may pick $C := \sqrt{2e/\pi}$, because

$$\frac{f(x)}{g(x)} = \sqrt{\frac{2}{\pi}} e^{-x^2/2+x} = \sqrt{\frac{2}{\pi}} e^{-(x-1)^2/2} e^{1/2} \leqslant \sqrt{\frac{2e}{\pi}}.$$

Generating generally distributed numbers

Discrete random variables	Continuous random variables
Binomial ✓	Normal, Lognormal ✓
Geometric ✓	Exponential ✓
Neg. binomial ✓	Gamma (and Erlang ✓)
Poisson ✓	Weibull, Pareto ✓

Generating generally distributed numbers

On the efficiency of acceptance-rejection: probability of acceptance is $1/C$. More precisely: the number of attempts before a sample can be accepted is geometrically distributed with success probability $1/C$.

Generating generally distributed numbers

On the efficiency of acceptance-rejection: probability of acceptance is $1/C$. More precisely: the number of attempts before a sample can be accepted is geometrically distributed with success probability $1/C$.

As a consequence you want to pick C as small as possible.

Generating generally distributed numbers

Gamma distribution can be done with acceptance rejection;
Example 2.8 in A & G.
'Dominating density' (i.e., the density of Y) is 'empirically found'.

Generating generally distributed numbers

Discrete random variables	Continuous random variables
Binomial ✓	Normal, Lognormal ✓
Geometric ✓	Exponential ✓
Neg. binomial ✓	Gamma ✓ (and Erlang ✓)
Poisson ✓	Weibull, Pareto ✓

Generating generally distributed numbers

A cool specific algorithm for the Normal distribution (Box & Muller):
With U_1 and U_2 independent uniforms on $[0, 1]$,

$$Y_1 := \sqrt{-2 \log U_1} \sin(2\pi U_2), \quad Y_2 := \sqrt{-2 \log U_1} \cos(2\pi U_2)$$

yields two independent standard Normal random variables.

Generating generally distributed numbers

Proof: first observe

$$U_1 \equiv U_1(Y_1, Y_2) = e^{-(Y_1^2+Y_2^2)/2},$$

$$U_2 \equiv U_2(Y_1, Y_2) = \frac{1}{2\pi} \arctan\left(\frac{Y_2}{Y_1}\right).$$

Generating generally distributed numbers

Proof: first observe

$$U_1 \equiv U_1(Y_1, Y_2) = e^{-(Y_1^2+Y_2^2)/2},$$
$$U_2 \equiv U_2(Y_1, Y_2) = \frac{1}{2\pi} \arctan\left(\frac{Y_2}{Y_1}\right).$$

Hence, with $J(y_1, y_2)$ the determinant of the Jacobian,

$$\begin{aligned} f_{Y_1,Y_2}(y_1, y_2) &= J(y_1, y_2) \cdot f_{U_1,U_2}(u_1(y_1, y_2), u_2(y_1, y_2)) \\ &= J(y_1, y_2). \end{aligned}$$

Generating generally distributed numbers

$$\frac{\mathrm{d}u_1}{\mathrm{d}y_1} = -y_1 u_1, \quad \frac{\mathrm{d}u_2}{\mathrm{d}y_1} = \frac{1}{2\pi} \cdot \frac{1}{1 + y_2^2/y_1^2} \cdot -\frac{y_2}{y_1^2} \quad = \frac{1}{2\pi} \cdot \frac{-y_2}{y_1^2 + y_2^2}$$

$$\frac{\mathrm{d}u_1}{\mathrm{d}y_2} = -y_2 u_1, \quad \frac{\mathrm{d}u_2}{\mathrm{d}y_2} = \frac{1}{2\pi} \cdot \frac{1}{1 + y_2^2/y_1^2} \cdot \frac{1}{y_1} \quad = \frac{1}{2\pi} \cdot \frac{y_1}{y_1^2 + y_2^2}.$$

Generating generally distributed numbers

$$\frac{\mathrm{d}u_1}{\mathrm{d}y_1} = -y_1 u_1, \quad \frac{\mathrm{d}u_2}{\mathrm{d}y_1} = \frac{1}{2\pi} \cdot \frac{1}{1+y_2^2/y_1^2} \cdot -\frac{y_2}{y_1^2} \quad = \frac{1}{2\pi} \cdot \frac{-y_2}{y_1^2+y_2^2}$$

$$\frac{\mathrm{d}u_1}{\mathrm{d}y_2} = -y_2 u_1, \quad \frac{\mathrm{d}u_2}{\mathrm{d}y_2} = \frac{1}{2\pi} \cdot \frac{1}{1+y_2^2/y_1^2} \cdot \frac{1}{y_1} \quad = \frac{1}{2\pi} \cdot \frac{y_1}{y_1^2+y_2^2}.$$

Determinant is $(2\pi)^{-1} u_1 = (2\pi)^{-1}\, e^{-(y_1^2+y_2^2)/2}$.

Generating generally distributed numbers

$$\frac{\mathrm{d}u_1}{\mathrm{d}y_1} = -y_1 u_1, \quad \frac{\mathrm{d}u_2}{\mathrm{d}y_1} = \frac{1}{2\pi} \cdot \frac{1}{1 + y_2^2/y_1^2} \cdot -\frac{y_2}{y_1^2} \quad = \frac{1}{2\pi} \cdot \frac{-y_2}{y_1^2 + y_2^2}$$

$$\frac{\mathrm{d}u_1}{\mathrm{d}y_2} = -y_2 u_1, \quad \frac{\mathrm{d}u_2}{\mathrm{d}y_2} = \frac{1}{2\pi} \cdot \frac{1}{1 + y_2^2/y_1^2} \cdot \frac{1}{y_1} \quad = \frac{1}{2\pi} \cdot \frac{y_1}{y_1^2 + y_2^2}.$$

Determinant is $(2\pi)^{-1} u_1 = (2\pi)^{-1} e^{-(y_1^2+y_2^2)/2}$.
Hence, as desired,

$$f_{Y_1,Y_2}(y_1, y_2) = \frac{1}{2\pi} e^{-(y_1^2+y_2^2)/2}.$$

Generating generally distributed numbers

Apart from inversion-CDF (uses CDF) and acceptance-rejection (uses PDF) there are quite a few alternative techniques.

In some cases, CDF or PDF is not available, but the transform

$$\hat{F}[s] := \int e^{sx}\mathbb{P}(X \in \mathrm{d}x)$$

is. Then algorithms of following type may work.

Other techniques

Inversion formula, with $\psi(s) := \hat{F}[\mathrm{i}s]$:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \psi(s) e^{-\mathrm{i}sx} \mathrm{d}s.$$

Other techniques

Inversion formula, with $\psi(s) := \hat{F}[\mathrm{i}s]$:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \psi(s) e^{-\mathrm{i}sx} \mathrm{d}s.$$

Algorithm:

- Fix n. Evaluate $f(x_i)$ for $x_1, \ldots, x_n$ using inversion formula.

Other techniques

Inversion formula, with $\psi(s) := \hat{F}[\mathrm{i}s]$:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \psi(s) e^{-\mathrm{i}sx} \mathrm{d}s.$$

Algorithm:

- Fix n. Evaluate $f(x_i)$ for $x_1, \ldots, x_n$ using inversion formula.
- Construct approximate distribution with probability mass

$$p_i := \frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)}$$

  in x_i.

Other techniques

Inversion formula, with $\psi(s) := \hat{F}[\mathrm{i}s]$:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \psi(s) e^{-\mathrm{i}sx} \mathrm{d}s.$$

Algorithm:

- Fix n. Evaluate $f(x_i)$ for $x_1, \ldots, x_n$ using inversion formula.
- Construct approximate distribution with probability mass

  $$p_i := \frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)}$$

  in x_i.
- Sample from this approximate distribution function.

Generating random vectors

[§II.3 of A & G]

The most relevant example is: how to draw a sample from a multivariate Normal distribution? Make us of Cholesky decomposition.

Generating random vectors

[§II.3 of A & G]

The most relevant example is: how to draw a sample from a multivariate Normal distribution? Make us of Cholesky decomposition.

Multivariate Normal distribution characterized through mean vector $\boldsymbol{\mu}$ (of length p) and covariance matrix Σ (of dimension $p \times p$). Without loss of generality: $\boldsymbol{\mu} = \mathbf{0}$ [Why?]

Generating random vectors

[§II.3 of A & G]

The most relevant example is: how to draw a sample from a multivariate Normal distribution? Make us of Cholesky decomposition.

Multivariate Normal distribution characterized through mean vector $\boldsymbol{\mu}$ (of length p) and covariance matrix Σ (of dimension $p \times p$). Without loss of generality: $\boldsymbol{\mu} = \mathbf{0}$ [Why?]

Σ is positive definite, hence can be written as CC^{T}, for a lower triangular matrix C.

Generating random vectors

Now the X_i (for $i = 1, \ldots, p$) can be sampled as follows.
Let $\boldsymbol{Y}$ a p-dimensional Normal vector with independent standard Normal components (i.e., with covariance matrix I). Let $\boldsymbol{X} := C\boldsymbol{Y}$.

Generating random vectors

Now the X_i (for $i = 1, \dots, p$) can be sampled as follows.
Let $\boldsymbol{Y}$ a p-dimensional Normal vector with independent standard Normal components (i.e., with covariance matrix I). Let
$\boldsymbol{X} := C\boldsymbol{Y}$.

Then $\boldsymbol{X}$ has the right covariance matrix $CC^{\mathsf{T}} = \Sigma$ (use standard rules for multivariate Normal distributions).

Generating elementary stochastic processes

[§II.4 of A & G]

I. Discrete-time Markov chains. Characterized by transition matrix $P = (p_{ij})_{i,j \in E}$, with E the state space.

For ease we take $E = \{1, 2, \ldots\}$. Trivially simulated, relying procedure that uses $\bar{p}_{ij} := \sum_{k=1}^{j} p_{ik}$.

Generating elementary stochastic processes

[§II.4 of A & G]

I. Discrete-time Markov chains. Characterized by transition matrix $P = (p_{ij})_{i,j \in E}$, with E the state space.

For ease we take $E = \{1, 2, \dots\}$. Trivially simulated, relying procedure that uses $\bar{p}_{ij} := \sum_{k=1}^{j} p_{ik}$.

II. Continuous-time Markov chains. Characterized by transition rate matrix $Q = (q_{ij})_{i,j \in E}$, with E the state space. Sample times spent in each of states from exponential distribution, and then use procedure above to sample next state.

Generating elementary stochastic processes

III. Poisson process (with rate λ). Interarrival times are exponential (with mean λ^{-1}).

IV. Inhomogeneous Poisson process (with rate $\lambda(t)$ at time t). Assume $\lambda(t) \leqslant \lambda$ across all values of t. U_n: i.i.d. uniform numbers on $[0, 1]$.

Generating elementary stochastic processes

III. Poisson process (with rate λ). Interarrival times are exponential (with mean λ^{-1}).

IV. Inhomogeneous Poisson process (with rate $\lambda(t)$ at time t). Assume $\lambda(t) \leqslant \lambda$ across all values of t. U_n: i.i.d. uniform numbers on $[0, 1]$.

Algorithm: (where n corresponds to a homogeneous Poisson process with rate β, and $n^\star$ to the inhomogeneous Poisson process)

- Step 1: $n := 0$, $n^\star := 0$, $\sigma := 0$.
- Step 2: $n := n + 1$, $T_n \sim \exp(\lambda)$ (i.e., $T_n := -\log U_n/\lambda$), $\sigma := \sigma + T_n$.
- Step 3: If $U_n' \leqslant \lambda(\sigma)/\lambda$, then $n^\star := n^\star + 1$.
- Step 4: Go to Step 2.

Intermezzo
DISCRETE-EVENT SIMULATION

Chapter III
OUTPUT ANALYSIS

Simulation as a computational tool

[§III.1 of A & G]

Idea: we want to estimate the performance measure $z := \mathbb{E}Z$.
Perform independent samples $Z_1, \ldots, Z_R$ of Z.
Estimate:

$$\hat{z}_R := \frac{1}{R} \sum_{r=1}^{R} Z_r.$$

Simulation as a computational tool

[§III.1 of A & G]

Idea: we want to estimate the performance measure $z := \mathbb{E}Z$.
Perform independent samples $Z_1, \ldots, Z_R$ of Z.
Estimate:

$$\hat{z}_R := \frac{1}{R} \sum_{r=1}^{R} Z_r.$$

Question: what is the performance of this estimator?

Simulation as a computational tool

[§III.1 of A & G]

Idea: we want to estimate the performance measure $z := \mathbb{E}Z$.
Perform independent samples $Z_1, \ldots, Z_R$ of Z.
Estimate:

$$\hat{z}_R := \frac{1}{R} \sum_{r=1}^{R} Z_r.$$

Question: what is the performance of this estimator?

Two issues play a role: *bias* (is expectation equal to parameter of interest?) and *accuracy* (what is variance of estimator?).
When unbiased, we would like to have a minimal variance. Often the objective is to approximate its distribution (e.g. asymptotic Normality).

Simulation as a computational tool

Standard approach: if $\sigma^2 := \mathbb{V}\mathrm{ar} Z < \infty$, then the central limit theorem provided us with: as $R \to \infty$,

$$\sqrt{R}\big(\hat{z}_R - z\big) \to \mathscr{N}(0, \sigma^2).$$

Simulation as a computational tool

Standard approach: if $\sigma^2 := \mathbb{V}\mathrm{ar} Z < \infty$, then the central limit theorem provided us with: as $R \to \infty$,

$$\sqrt{R}\big(\hat{z}_R - z\big) \to \mathscr{N}(0, \sigma^2).$$

This suggests for finite R the approximation, with $V \sim \mathscr{N}(0, 1)$,

$$\hat{z}_R \overset{\mathsf{d}}{\approx} z + \frac{\sigma V}{\sqrt{R}}.$$

Simulation as a computational tool

Based on this convergence in distribution, one could use confidence intervals if the type

$$\left(\hat{z}_R - q_\alpha \frac{\sigma}{\sqrt{R}}, \hat{z}_R + q_\alpha \frac{\sigma}{\sqrt{R}}\right),$$

with q_α reflecting the $\alpha/2$-quantile of $\mathcal{N}(0,1)$, in the sense that

$$1 - \Phi(q_\alpha) = \mathbb{P}(\mathcal{N}(0,1) \geqslant q_\alpha) = \frac{1-\alpha}{2}$$

(for example confidence level α equalling 0.95 leads to $q_\alpha = 1.96$).

Simulation as a computational tool

Based on this convergence in distribution, one could use confidence intervals if the type

$$\left(\hat{z}_R - q_\alpha \frac{\sigma}{\sqrt{R}}, \hat{z}_R + q_\alpha \frac{\sigma}{\sqrt{R}}\right),$$

with q_α reflecting the $\alpha/2$-quantile of $\mathscr{N}(0,1)$, in the sense that

$$1 - \Phi(q_\alpha) = \mathbb{P}(\mathscr{N}(0,1) \geqslant q_\alpha) = \frac{1-\alpha}{2}$$

(for example confidence level α equalling 0.95 leads to $q_\alpha = 1.96$).

Are we done now?

Simulation as a computational tool

Based on this convergence in distribution, one could use confidence intervals if the type

$$\left(\hat{z}_R - q_\alpha \frac{\sigma}{\sqrt{R}}, \hat{z}_R + q_\alpha \frac{\sigma}{\sqrt{R}}\right),$$

with q_α reflecting the $\alpha/2$-quantile of $\mathscr{N}(0,1)$, in the sense that

$$1 - \Phi(q_\alpha) = \mathbb{P}(\mathscr{N}(0,1) \geqslant q_\alpha) = \frac{1-\alpha}{2}$$

(for example confidence level α equalling 0.95 leads to $q_\alpha = 1.96$).

Are we done now? NO! We don't know σ^2.

Simulation as a computational tool

Idea: *estimate* σ^2.

Simulation as a computational tool

Idea: *estimate* σ^2.

Traditional estimator:

$$s^2 := \frac{1}{R-1}\sum_{r=1}^{R}(Z_r - \hat{z}_R)^2 = \frac{1}{R-1}\left(\sum_{r=1}^{R} Z_r^2 - R\hat{z}_R^2\right).$$

This is an unbiased estimator. [Check!]

Computing smooth functions of expectations

[§III.3 of A & G]

Example: estimation of standard deviation of random variable W.

Computing smooth functions of expectations

[§III.3 of A & G]

Example: estimation of standard deviation of random variable W.

Then: $\boldsymbol{z} = (z_1, z_2)^{\mathsf{T}}$, with $z_i = \mathbb{E}Z(i)$, where

$$Z(1) = W^2, \quad Z(2) = W.$$

Computing smooth functions of expectations

[§III.3 of A & G]

Example: estimation of standard deviation of random variable W.

Then: $\boldsymbol{z} = (z_1, z_2)^\mathsf{T}$, with $z_i = \mathbb{E}Z(i)$, where

$$Z(1) = W^2, \quad Z(2) = W.$$

Standard deviation $\sigma = f(\boldsymbol{z})$, with

$$f(\boldsymbol{z}) = \sqrt{z_1 - z_2^2}.$$

Here our aim to estimate a smooth function of the expectations $\mathbb{E}W^2$ and $\mathbb{E}W$.

Computing smooth functions of expectations

Another example: Let A_n and B_n be independent sequences of i.i.d. random variables.
Perpetuity is:

$$Y = \sum_{n=0}^{\infty} B_n \prod_{m=1}^{n} A_m$$

(think of B_n as amounts put on an account in slot n, and A_n the interest in slot n).

Computing smooth functions of expectations

Another example: Let A_n and B_n be independent sequences of i.i.d. random variables.
Perpetuity is:

$$Y = \sum_{n=0}^{\infty} B_n \prod_{m=1}^{n} A_m$$

(think of B_n as amounts put on an account in slot n, and A_n the interest in slot n). Then

$$\mathbb{E}Y = \frac{\mathbb{E}B}{1 - \mathbb{E}A}.$$

Computing smooth functions of expectations

We wish to estimate

$$\varrho := \mathbb{E}Y = \frac{\mathbb{E}B}{1 - \mathbb{E}A}.$$

Then: $\boldsymbol{z} = (z_1, z_2)^{\mathsf{T}}$, with $z_i = \mathbb{E}Z(i)$, where

$$Z(1) = B, \quad Z(2) = A.$$

Computing smooth functions of expectations

We wish to estimate

$$\varrho := \mathbb{E}Y = \frac{\mathbb{E}B}{1 - \mathbb{E}A}.$$

Then: $\boldsymbol{z} = (z_1, z_2)^\mathsf{T}$, with $z_i = \mathbb{E}Z(i)$, where

$$Z(1) = B, \quad Z(2) = A.$$

Goal is to estimate $\varrho = f(\boldsymbol{z})$, with

$$f(\boldsymbol{z}) = \frac{z_1}{1 - z_2}.$$

Again, our aim to estimate a smooth function of expectations, in this case $\mathbb{E}B$ and $\mathbb{E}A$.

Computing smooth functions of expectations

How to to estimate a smooth function of expectations?

Naïve idea: estimate $f(\boldsymbol{z})$ by $f(\hat{\boldsymbol{z}}_R)$, with

$$\hat{\boldsymbol{z}}_R := \frac{1}{R}\sum_{r=1}^{R} \boldsymbol{Z}_r .$$

Computing smooth functions of expectations

First example: recall $Z_r(1) = W_r^2$ and $Z_r(2) = W_r$, and

$$f(\boldsymbol{z}) = \sqrt{z_1 - z_2^2}.$$

So estimator is

$$\hat{\sigma} = \sqrt{\frac{1}{R}\sum_{r=1}^{R} Z_r(1) - \left(\frac{1}{R}\sum_{r=1}^{R} Z_r(2)\right)^2}.$$

Computing smooth functions of expectations

Second example: recall $Z_r(1) = B_r$ and $Z_r(2) = A_r$, and

$$f(\boldsymbol{z}) = \frac{z_1}{1 - z_2}.$$

So estimator is

$$\hat{\varrho} = \left(\frac{1}{R} \sum_{r=1}^{R} Z_r(1) \right) \Bigg/ \left(1 - \frac{1}{R} \sum_{r=1}^{R} Z_r(2) \right).$$

Computing smooth functions of expectations

Question: is this procedure any good?

Computing smooth functions of expectations

Question: is this procedure any good?

Evident: consistent as $R \to \infty$

Computing smooth functions of expectations

Question: is this procedure any good?

Evident: consistent as $R \to \infty$ — at least, in case $f(\boldsymbol{z})$ is continuous at $\boldsymbol{z}$.

Computing smooth functions of expectations

Question: is this procedure any good?

Evident: consistent as $R \to \infty$ — at least, in case $f(\boldsymbol{z})$ is continuous at $\boldsymbol{z}$.

Can we again quantify the rate of convergence? Is there approximate Normality?

Computing smooth functions of expectations

Question: is this procedure any good?

Evident: consistent as $R \to \infty$ — at least, in case $f(\boldsymbol{z})$ is continuous at $\boldsymbol{z}$.

Can we again quantify the rate of convergence? Is there approximate Normality? This can be assessed by applying the so-called delta-method.

Computing smooth functions of expectations

Using the usual Taylor arguments, with $\nabla f(\boldsymbol{z})$ the row vector of partial derivatives,

$$
\begin{aligned}
f(\hat{\boldsymbol{z}}_R) - f(\boldsymbol{z}) &= \nabla f(\boldsymbol{z}) \cdot (\hat{\boldsymbol{z}}_R - \boldsymbol{z}) + o(\|\hat{\boldsymbol{z}}_R - \boldsymbol{z}\|) \\
&= \frac{1}{R} \sum_{r=1}^{R} V_r + o(\|\hat{\boldsymbol{z}}_R - \boldsymbol{z}\|),
\end{aligned}
$$

where V_r denotes $\nabla f(\boldsymbol{z}) \cdot (\boldsymbol{Z}_r - \boldsymbol{z})$. Is there again a central limit theorem?

Computing smooth functions of expectations

Yes!

$$\sqrt{R}\big(f(\hat{\boldsymbol{z}}_R) - f(\boldsymbol{z})\big) \to \mathcal{N}(0, \sigma^2),$$

with $\sigma^2 := \mathbb{V}\mathrm{ar}\, V_1$.

Computing smooth functions of expectations

Yes!

$$\sqrt{R}\big(f(\hat{\boldsymbol{z}}_R) - f(\boldsymbol{z})\big) \to \mathcal{N}(0, \sigma^2),$$

with $\sigma^2 := \mathbb{V}\mathrm{ar}\, V_1$.

How to evaluate σ^2? Define $\Sigma_{ij} := \mathbb{C}\mathrm{ov}(Z_i, Z_j)$. Then

$$\sigma^2 = \nabla f(\boldsymbol{z}) \cdot \Sigma \cdot \nabla f(\boldsymbol{z})^{\mathsf{T}}.$$

Computing smooth functions of expectations

Yes!

$$\sqrt{R}\big(f(\hat{\boldsymbol{z}}_R) - f(\boldsymbol{z})\big) \to \mathcal{N}(0, \sigma^2),$$

with $\sigma^2 := \mathbb{V}\mathrm{ar}\, V_1$.

How to evaluate σ^2? Define $\Sigma_{ij} := \mathbb{C}\mathrm{ov}(Z_i, Z_j)$. Then

$$\sigma^2 = \nabla f(\boldsymbol{z}) \cdot \Sigma \cdot \nabla f(\boldsymbol{z})^{\mathsf{T}}.$$

(Later we'll see an application of this technique: regenerative method to compute steady-state quantities.)

Computing smooth functions of expectations

Yes!

$$\sqrt{R}\big(f(\hat{\boldsymbol{z}}_R) - f(\boldsymbol{z})\big) \to \mathcal{N}(0, \sigma^2),$$

with $\sigma^2 := \mathbb{V}\mathrm{ar}\, V_1$.

How to evaluate σ^2? Define $\Sigma_{ij} := \mathbb{C}\mathrm{ov}(Z_i, Z_j)$. Then

$$\sigma^2 = \nabla f(\boldsymbol{z}) \cdot \Sigma \cdot \nabla f(\boldsymbol{z})^\mathsf{T}.$$

(Later we'll see an application of this technique: regenerative method to compute steady-state quantities.)

For $d = 1$ (where d is dimension of vector $\boldsymbol{Z}_r$) , we just get $\sigma^2 = (f'(z))^2\, \mathbb{V}\mathrm{ar} Z$; for $f(z) = z$ this gives us back our earlier procedure. But there are two differences as well.

Computing smooth functions of expectations

Difference 1: estimator $f(\hat{\boldsymbol{z}}_R)$ is generally biased.

Computing smooth functions of expectations

Difference 1: estimator $f(\hat{\boldsymbol{z}}_R)$ is generally biased. Effect can be quantified (include one more term in expansion): with $H(\boldsymbol{z})$ the Hessian in $\boldsymbol{z}$,

$$\begin{aligned} f(\hat{\boldsymbol{z}}_R) - f(\boldsymbol{z}) &= \nabla f(\boldsymbol{z}) \cdot (\hat{\boldsymbol{z}}_R - \boldsymbol{z}) \\ &\qquad + \frac{1}{2}(\hat{\boldsymbol{z}}_R - \boldsymbol{z})^{\mathsf{T}} H(\boldsymbol{z})(\hat{\boldsymbol{z}}_R - \boldsymbol{z}) + o(\|\hat{\boldsymbol{z}}_R - \boldsymbol{z}\|^2). \end{aligned}$$

Now take means, to obtain the bias, as $R \to \infty$,

$$\mathbb{E}\big(f(\hat{\boldsymbol{z}}_R) - f(\boldsymbol{z})\big) = \frac{1}{2R} \sum_{i,j} \Sigma_{ij} H_{ij}(\boldsymbol{z}) + o(R^{-1}),$$

with as before $\Sigma_{ij} := \mathbb{C}\mathrm{ov}(Z_i, Z_j)$.

Computing smooth functions of expectations

Difference 1: estimator $f(\hat{\boldsymbol{z}}_R)$ is generally biased. Effect can be quantified (include one more term in expansion): with $H(\boldsymbol{z})$ the Hessian in $\boldsymbol{z}$,

$$\begin{aligned} f(\hat{\boldsymbol{z}}_R) - f(\boldsymbol{z}) &= \nabla f(\boldsymbol{z}) \cdot (\hat{\boldsymbol{z}}_R - \boldsymbol{z}) \\ &\qquad + \frac{1}{2}(\hat{\boldsymbol{z}}_R - \boldsymbol{z})^{\mathsf{T}} H(\boldsymbol{z})(\hat{\boldsymbol{z}}_R - \boldsymbol{z}) + o(\|\hat{\boldsymbol{z}}_R - \boldsymbol{z}\|^2). \end{aligned}$$

Now take means, to obtain the bias, as $R \to \infty$,

$$\mathbb{E}\big(f(\hat{\boldsymbol{z}}_R) - f(\boldsymbol{z})\big) = \frac{1}{2R} \sum_{i,j} \Sigma_{ij} H_{ij}(\boldsymbol{z}) + o(R^{-1}),$$

with as before $\Sigma_{ij} := \mathbb{C}\mathrm{ov}(Z_i, Z_j)$.
Remedy: adapted estimator (with $\hat{\Sigma}_{ij}$ the obvious estimator of Σ_{ij})

$$f(\hat{\boldsymbol{z}}_R) - \frac{1}{2R} \sum_{i,j} \hat{\Sigma}_{ij} H_{ij}(\hat{\boldsymbol{z}}_R).$$

Computing smooth functions of expectations

Difference 2: σ^2 harder to estimate.

Computing smooth functions of expectations

Difference 2: σ^2 harder to estimate. Evident candidate:

$$\hat{\sigma}^2 := \frac{1}{R-1} \sum_{r=1}^{R} \big(\nabla f(\hat{\boldsymbol{z}}_R) (\boldsymbol{Z}_r - \hat{\boldsymbol{z}}_R) \big)^2.$$

This requires computation of the gradient of f (if this is explicitly available, then there is obviously no problem).

Computing roots of equations defined by expectations

[§III.4 of A & G]

Let $f : \mathbb{R}^{d+1} \mapsto \mathbb{R}$ be known, $\boldsymbol{z} := \mathbb{E}\boldsymbol{Z} \in \mathbb{R}^d$. Then we want to find the (a?) θ such that

$$f(\boldsymbol{z}, \theta) = 0;$$

call the root $\theta^\star$.

Computing roots of equations defined by expectations

[§III.4 of A & G]

Let $f : \mathbb{R}^{d+1} \mapsto \mathbb{R}$ be known, $\boldsymbol{z} := \mathbb{E}\boldsymbol{Z} \in \mathbb{R}^d$. Then we want to find the (a?) θ such that

$$f(\boldsymbol{z}, \theta) = 0;$$

call the root $\theta^\star$.

If an explicit $\zeta : \mathbb{R}^d \mapsto \mathbb{R}$ is known such that $\theta^\star = \zeta(\boldsymbol{z})$, then we're in the framework of the previous section. So focus on case ζ is not known.

Computing roots of equations defined by expectations

Procedure: (i) estimate $\boldsymbol{z}$ by $\hat{\boldsymbol{z}}_R$, and (ii) find $\hat{\theta}_R$ by solving

$$f(\hat{\boldsymbol{z}}_R, \hat{\theta}_R) = 0.$$

How to get confidence intervals?

Computing roots of equations defined by expectations

Procedure: (i) estimate $\boldsymbol{z}$ by $\hat{\boldsymbol{z}}_R$, and (ii) find $\hat{\theta}_R$ by solving

$$f(\hat{\boldsymbol{z}}_R, \hat{\theta}_R) = 0.$$

How to get confidence intervals?

Trivial:

$$\begin{aligned} 0 &= f(\hat{\boldsymbol{z}}_R, \hat{\theta}_R) - f(\boldsymbol{z}, \theta^\star) \\ &= f(\hat{\boldsymbol{z}}_R, \hat{\theta}_R) - f(\boldsymbol{z}, \hat{\theta}_R) + f(\boldsymbol{z}, \hat{\theta}_R) - f(\boldsymbol{z}, \theta^\star). \end{aligned}$$

Apply delta-method again: in usual notation,

$$\nabla f_{\boldsymbol{z}}(\boldsymbol{z}, \theta^\star)\big(\hat{\boldsymbol{z}}_R - \boldsymbol{z}\big) + f_\theta(\boldsymbol{z}, \theta^\star)(\hat{\theta}_R - \theta^\star) + O(\|\hat{\boldsymbol{z}}_R - \boldsymbol{z}\|^2) = 0.$$

Computing roots of equations defined by expectations

From

$$\nabla f_{\boldsymbol{z}}(\boldsymbol{z},\theta^\star)\big(\hat{\boldsymbol{z}}_R-\boldsymbol{z}\big)+f_\theta(\boldsymbol{z},\theta^\star)(\hat\theta_R-\theta^\star)+O(\|\hat{\boldsymbol{z}}_R-\boldsymbol{z}\|^2)=0$$

we obtain

$$\sqrt{R}\big(\hat\theta_R-\theta^\star\big)\to\mathcal{N}(0,\sigma^2),$$

with (recalling $\nabla f_{\boldsymbol{z}}(\boldsymbol{z},\theta^\star)$ is d-dimensional row vector)

$$\sigma^2=\frac{\mathbb{V}\mathrm{ar}\big(\nabla f_{\boldsymbol{z}}(\boldsymbol{z},\theta^\star)\cdot\boldsymbol{Z}\big)}{\big(f_\theta(\boldsymbol{z},\theta^\star)\big)^2}.$$

As before σ^2 can be estimated by

$$\hat\sigma^2=\frac{\dfrac{1}{R-1}\displaystyle\sum_{r=1}^{R}\big(\nabla f_{\boldsymbol{z}}(\hat{\boldsymbol{z}}_R,\hat\theta_R)\cdot(\boldsymbol{Z}_r-\boldsymbol{z})\big)^2}{\big(f_\theta(\hat{\boldsymbol{z}}_R,\hat\theta_R)\big)^2}.$$

Chapter IV
STEADY-STATE SIMULATION

Complications arising when estimating steady-state quantities

[§IV.1 of A & G]

We now focus on ergodic stochastic processes $Y(\cdot)$. This means that a limiting time-average limit Y exists — think a stable queue.

Complications arising when estimating steady-state quantities

[§IV.1 of A & G]

We now focus on ergodic stochastic processes $Y(\cdot)$. This means that a limiting time-average limit Y exists — think a stable queue.

Our objective is to estimate

$$z := \lim_{t\to\infty} \frac{1}{t}\int_0^t Y(s)\mathrm{d}s.$$

How to do this?

Complications arising when estimating steady-state quantities

Naïve approach: let t grow large, and simulate process $Y(\cdot)$ for long time:

$$\hat{z}_T := \frac{1}{T}\int_0^T Y(s)\mathrm{d}s,$$

for T 'large'.

Complications arising when estimating steady-state quantities

Naïve approach: let t grow large, and simulate process $Y(\cdot)$ for long time:

$$\hat{z}_T := \frac{1}{T}\int_0^T Y(s)\mathrm{d}s,$$

for T 'large'.

Inherent problems:

Complications arising when estimating steady-state quantities

Naïve approach: let t grow large, and simulate process $Y(\cdot)$ for long time:

$$\hat{z}_T := \frac{1}{T}\int_0^T Y(s)\mathrm{d}s,$$

for T 'large'.

Inherent problems:

- how large should T be? (Depends on speed of convergence.)

Complications arising when estimating steady-state quantities

Naïve approach: let t grow large, and simulate process $Y(\cdot)$ for long time:

$$\hat{z}_T := \frac{1}{T}\int_0^T Y(s)\mathrm{d}s,$$

for T 'large'.

Inherent problems:

- how large should T be? (Depends on speed of convergence.)
- how to construct confidence intervals? (Observe that there are now no i.i.d. observations.)

Complications arising when estimating steady-state quantities

Under 'rather general conditions', as T grows large,

$$\sqrt{T}\big(\hat{z}_T - z\big) \to \mathscr{N}(0, \sigma^2),$$

for a σ^2 characterized below.

Complications arising when estimating steady-state quantities

Under 'rather general conditions', as T grows large,

$$\sqrt{T}\big(\hat{z}_T - z\big) \to \mathscr{N}(0, \sigma^2),$$

for a σ^2 characterized below.

This means that we can approximate, with $V \sim \mathscr{N}(0, 1)$,

$$\hat{z}_T \stackrel{\mathsf{d}}{=} z + \frac{\sigma V}{\sqrt{T}}.$$

Complications arising when estimating steady-state quantities

Essentially for the CLT to hold, we should have that

$$\mathbb{V}\mathrm{ar}\left(\int_0^T Y(s)\mathrm{d}s\right)$$

scales linearly in T as T grows large, with σ^2 being the proportionality constant.

Complications arising when estimating steady-state quantities

Essentially for the CLT to hold, we should have that

$$\mathbb{V}\mathrm{ar}\left(\int_0^T Y(s)\mathrm{d}s\right)$$

scales linearly in T as T grows large, with σ^2 being the proportionality constant.
Let's compute this constant

$$\sigma^2 := \lim_{T\to\infty} \frac{1}{T}\mathbb{V}\mathrm{ar}\left(\int_0^T Y(s)\mathrm{d}s\right).$$

Complications arising when estimating steady-state quantities

Essentially for the CLT to hold, we should have that

$$\mathbb{V}\mathrm{ar}\left(\int_0^T Y(s)\mathrm{d}s\right)$$

scales linearly in T as T grows large, with σ^2 being the proportionality constant.
Let's compute this constant

$$\sigma^2 := \lim_{T\to\infty} \frac{1}{T}\mathbb{V}\mathrm{ar}\left(\int_0^T Y(s)\mathrm{d}s\right).$$

Define

$$c(s) := \mathbb{C}\mathrm{ov}_\pi(Y(0), Y(s)),$$

with subscript π denoting the system was in stationarity at time 0.
In addition, [Why?]

$$c(s,v) := \mathbb{C}\mathrm{ov}_\pi(Y(s), Y(v)) = c(|s-v|).$$

Complications arising when estimating steady-state quantities

Using in the first step a standard rule for variance of integrals,

$$
\begin{aligned}
\frac{1}{T}\mathbb{V}\mathrm{ar}_\pi\left(\int_0^T Y(s)\mathrm{d}s\right) &= \frac{1}{T}\int_0^T\int_0^T \mathbb{C}\mathrm{ov}_\pi(Y(s),Y(v))\mathrm{d}s\,\mathrm{d}v \\
&= \frac{2}{T}\int_0^T\int_0^s \mathbb{C}\mathrm{ov}_\pi(Y(s),Y(v))\mathrm{d}v\,\mathrm{d}s \\
&= \frac{2}{T}\int_0^T\int_0^s c(s-v)\mathrm{d}v\,\mathrm{d}s \\
&= \frac{2}{T}\int_0^T\int_0^s c(v)\mathrm{d}v\,\mathrm{d}s = \frac{2}{T}\int_0^T\int_v^T c(v)\mathrm{d}s\,\mathrm{d}v \\
&= 2\int_0^T\left(1-\frac{v}{T}\right)c(v)\mathrm{d}v \to 2\int_0^\infty c(v)\mathrm{d}v,
\end{aligned}
$$

as $T\to\infty$. (Last step: dominated convergence.)

Complications arising when estimating steady-state quantities

[Notice an inconsistency in A & G: proof is for initial distribution π, whereas claim is stated for general initial distribution.]

Complications arising when estimating steady-state quantities

[Notice an inconsistency in A & G: proof is for initial distribution π, whereas claim is stated for general initial distribution.]

Result:

$$\sigma^2 = 2\int_0^\infty c(v)\mathrm{d}v.$$

Complications arising when estimating steady-state quantities

[Notice an inconsistency in A & G: proof is for initial distribution π, whereas claim is stated for general initial distribution.]

Result:

$$\sigma^2 = 2\int_0^\infty c(v)\mathrm{d}v.$$

Apparently, for the claim to hold, one should have that the covariances of $Y(0)$ and $Y(t)$ (and hence the autocorrelation function) has a finite integral. We sometimes call this: the process $Y(\cdot)$ is *short-range dependent*.

Finite-state continuous-time Markov chain

[§IV.2 of A & G]

$(X(t))_{t\geqslant 0}$ irreducible Markov chain on $\{1,\ldots,d\}$, with transition rate matrix Q.
Define $Y(t)=f(X(t))$. To be computed:

$$\sigma^2 = 2\int_0^\infty \mathbb{C}\mathrm{ov}_\pi(Y(0),Y(s))\mathrm{d}s.$$

First rewrite expression to, with $p_{ij}(t) := \mathbb{P}(X(t)=j\,|\,X(0)=i)$,

$$2\int_0^\infty \sum_{i=1}^d \sum_{j=1}^d \left(\pi_i p_{ij}(s) - \pi_i\pi_j\right) f(i)f(j)\mathrm{d}s.$$

Finite-state continuous-time Markov chain

[§IV.2 of A & G]

$(X(t))_{t\geqslant 0}$ irreducible Markov chain on $\{1,\ldots,d\}$, with transition rate matrix Q.
Define $Y(t)=f(X(t))$. To be computed:

$$\sigma^2 = 2\int_0^\infty \mathbb{C}\mathrm{ov}_\pi(Y(0),Y(s))\mathrm{d}s.$$

First rewrite expression to, with $p_{ij}(t) := \mathbb{P}(X(t)=j\,|\,X(0)=i)$,

$$2\int_0^\infty \sum_{i=1}^d \sum_{j=1}^d \left(\pi_i p_{ij}(s) - \pi_i\pi_j\right) f(i)f(j)\mathrm{d}s.$$

Call

$$D_{ij} := \int_0^\infty \left(p_{ij}(s)-\pi_j\right)\mathrm{d}s.$$

Finite-state continuous-time Markov chain

We obtain that, with '$\bullet$' denoting the componentwise product,

$$\sigma^2 = 2\,(\boldsymbol{f} \bullet \boldsymbol{\pi})^{\mathsf{T}} D \boldsymbol{f}.$$

Finite-state continuous-time Markov chain

We obtain that, with '$\bullet$' denoting the componentwise product,

$$\sigma^2 = 2\,(\boldsymbol{f} \bullet \boldsymbol{\pi})^{\mathsf{T}} D \boldsymbol{f}.$$

The matrix D is referred to as the *deviation matrix*, and essentially measures the speed of convergence to the invariant distribution.

Finite-state continuous-time Markov chain

We obtain that, with '$\bullet$' denoting the componentwise product,

$$\sigma^2 = 2\,(\boldsymbol{f} \bullet \boldsymbol{\pi})^{\mathsf{T}} D \boldsymbol{f}.$$

The matrix D is referred to as the *deviation matrix*, and essentially measures the speed of convergence to the invariant distribution.

D can be alternatively evaluated as $F - \Pi$, with $\Pi := \mathbf{1}\boldsymbol{\pi}^{\mathsf{T}}$ (rank-one matrix), and

$$F := (\Pi - Q)^{-1}.$$

Regenerative method

[§IV.4 of A & G]

Recall: our objective is to estimate

$$z := \lim_{t\to\infty} \frac{1}{t}\int_0^t Y(s)\mathrm{d}s.$$

Let T be a regenerative point. Then (regeneration ratio formula):

$$z = \frac{\mathbb{E}I(T)}{\mathbb{E}T}, \qquad I(T) := \int_0^T Y(s)\mathrm{d}s.$$

Regenerative method

For discrete-state-space irreducible ergodic Markov process $X(t)$, one could define a 'return state' $i^\star$. Suppose $X(0) = i^\star$. Then the r-th return time is defined recursively: $\tau_0 := 0$ and

$$\tau_r := \inf\left\{t \geqslant 0 : X(t + \tau_{r-1}) = i^\star\right\},$$

and

$$I_r := \int_{\tau_{r-1}}^{\tau_r} Y(s)\mathrm{d}s.$$

Regenerative method

Consequence for simulation: we need to estimate $\mathbb{E}I(\tau)$ and $\mathbb{E}\tau$ (and find confidence intervals for the resulting estimator).

Regenerative method

Consequence for simulation: we need to estimate $\mathbb{E}I(\tau)$ and $\mathbb{E}\tau$ (and find confidence intervals for the resulting estimator).

Idea: simulate R regenerative cycles, providing observations $I_1, \ldots, I_R$ and $\tau_1, \ldots, \tau_R$.

Estimator: estimating numerator and denominator separately,

$$\hat{z}_R = \frac{1}{R}\sum_{r=1}^{R} I_r \Big/ \frac{1}{R}\sum_{r=1}^{R} \tau_r = \sum_{r=1}^{R} I_r \Big/ \sum_{r=1}^{R} \tau_r.$$

Confidence intervals?

[Many typos in A & G: τ_r and $\tilde{Y}_r$ need to be swapped. I chose slightly different, and more transparent, notation.]

Regenerative method

Define $Z_r := I_r - z\tau_r$, and $Z \overset{\text{d}}{=} Z_r$, $I \overset{\text{d}}{=} I_r$, $\tau \overset{\text{d}}{=} \tau_r$,

$$\eta^2 := \frac{\mathbb{E}Z^2}{(\mathbb{E}\tau)^2}.$$

Then, as $R \to \infty$,

$$\sqrt{R}\big(\hat{z}_R - z\big) \to \mathscr{N}(0, \eta^2).$$

Confidence intervals can be constructed as before, after estimating η^2 by

$$\hat{\eta}_R^2 := \left(\frac{1}{R-1}\sum_{r=1}^{R}\big(I_r - \hat{z}_R\tau_r\big)^2\right) \Bigg/ \left(\frac{1}{R}\sum_{r=1}^{R}\tau_r\right)^2.$$

Regenerative method

Proof: an application of the delta-method. Let $z_1 := \mathbb{E}I$ and $z_2 := \mathbb{E}\tau$. Define $f(z_1, z_2) = z_1/z_2$.
As we have seen, variance of estimator:

$$\sigma^2 = \nabla f(\boldsymbol{z}) \cdot \Sigma \cdot \nabla f(\boldsymbol{z})^\mathsf{T}.$$

Can be rewritten to

$$\left(\frac{1}{\mathbb{E}\tau}, -\frac{\mathbb{E}I}{(\mathbb{E}\tau)^2}\right) \begin{pmatrix} \mathbb{V}\mathrm{ar} I & \mathbb{C}\mathrm{ov}(I,\tau) \\ \mathbb{C}\mathrm{ov}(I,\tau) & \mathbb{V}\mathrm{ar}\tau \end{pmatrix} \left(\frac{1}{\mathbb{E}\tau}, -\frac{\mathbb{E}I}{(\mathbb{E}\tau)^2}\right)^\mathsf{T},$$

or

$$\left(\frac{\mathbb{E}\tau}{(\mathbb{E}\tau)^2}, -\frac{\mathbb{E}I}{(\mathbb{E}\tau)^2}\right) \begin{pmatrix} \mathbb{V}\mathrm{ar} I & \mathbb{C}\mathrm{ov}(I,\tau) \\ \mathbb{C}\mathrm{ov}(I,\tau) & \mathbb{V}\mathrm{ar}\tau \end{pmatrix} \left(\frac{\mathbb{E}\tau}{(\mathbb{E}\tau)^2}, -\frac{\mathbb{E}I}{(\mathbb{E}\tau)^2}\right)^\mathsf{T}.$$

Regenerative method

Going through the computations, we obtain

$$\frac{1}{(\mathbb{E}\tau)^4} \cdot \left((\mathbb{E}\tau)^2 \mathbb{V}\mathrm{ar}\, I - 2\mathbb{E}I\, \mathbb{E}\tau\, \mathbb{C}\mathrm{ov}(I,\tau) + (\mathbb{E}I)^2 \mathbb{V}\mathrm{ar}\,\tau\right),$$

or equivalently,

$$\begin{aligned} \frac{1}{(\mathbb{E}\tau)^2} \cdot \left(\mathbb{V}\mathrm{ar}\, I - 2z\, \mathbb{C}\mathrm{ov}(I,\tau) + z^2 \mathbb{V}\mathrm{ar}\,\tau\right) &= \frac{\mathbb{V}\mathrm{ar} Z}{(\mathbb{E}\tau)^2} \\ &= \frac{\mathbb{E}Z^2}{(\mathbb{E}\tau)^2}. \end{aligned}$$

Regenerative method

Going through the computations, we obtain

$$\frac{1}{(\mathbb{E}\tau)^4} \cdot \left((\mathbb{E}\tau)^2 \mathbb{V}\mathrm{ar}\, I - 2\mathbb{E}I\,\mathbb{E}\tau\,\mathbb{C}\mathrm{ov}(I,\tau) + (\mathbb{E}I)^2\mathbb{V}\mathrm{ar}\,\tau\right),$$

or equivalently,

$$\begin{aligned}\frac{1}{(\mathbb{E}\tau)^2} \cdot \left(\mathbb{V}\mathrm{ar}\, I - 2z\,\mathbb{C}\mathrm{ov}(I,\tau) + z^2\mathbb{V}\mathrm{ar}\,\tau\right) &= \frac{\mathbb{V}\mathrm{ar} Z}{(\mathbb{E}\tau)^2} \\ &= \frac{\mathbb{E}Z^2}{(\mathbb{E}\tau)^2}.\end{aligned}$$

Second claim (estimator for η^2) is obvious.

Regenerative method

Actually, a CLT for the related estimator

$$\hat{z}_T := \frac{1}{T} \int_0^T Y(s) \mathrm{d}s,$$

for T *deterministic*, can be found along the same lines.

Batch-means method

[§IV.5 of A & G]

Recall: our objective is to estimate

$$z := \lim_{t\to\infty} \frac{1}{t}\int_0^t Y(s)\mathrm{d}s.$$

Prerequisite is weak convergence to BM:

$$\left(\sqrt{nt}\cdot\left(\frac{1}{nt}\int_0^{nt} Y(s)\mathrm{d}s - z\right)\right)_{t\geqslant 0} \to (\sigma\, B(t))_{t\geqslant 0}.$$

[First factor is *not* $\sqrt{n}\,t$, as in A & G!]

Batch-means method

[§IV.5 of A & G]

Recall: our objective is to estimate

$$z := \lim_{t\to\infty} \frac{1}{t} \int_0^t Y(s)\mathrm{d}s.$$

Prerequisite is weak convergence to BM:

$$\left(\sqrt{nt} \cdot \left(\frac{1}{nt} \int_0^{nt} Y(s)\mathrm{d}s - z\right)\right)_{t\geqslant 0} \to (\sigma\, B(t))_{t\geqslant 0}.$$

[First factor is *not* $\sqrt{n}\, t$, as in A & G!] Such a *functional central limit theorem* typically holds when there is weak dependence in the $Y(\cdot)$ process.

Batch-means method

Define contributions of intervals of length t/R:

$$\bar{Y}_r(t) = \frac{1}{t/R} \int_{(r-1)t/R}^{rt/R} Y(s) \mathrm{d}s;$$

there are R of these.

Batch-means method

Define contributions of intervals of length t/R:

$$\bar{Y}_r(t) = \frac{1}{t/R} \int_{(r-1)t/R}^{rt/R} Y(s) \mathrm{d}s;$$

there are R of these.

Estimator:

$$\hat{z}_R := \frac{1}{R} \sum_{r=1}^{R} \bar{Y}_r(t).$$

Batch-means method

How to construct confidence intervals?
For any R,

$$\sqrt{R}\left(\frac{1}{t}\int_0^t Y(s)\mathrm{d}s - z\right)\Big/ s_R(t) \to T_{R-1}$$

as $t \to \infty$. Here T_R is a Student-t distribution with R degrees of freedom, and

$$S_R(t) := \frac{1}{R-1}\sum_{r=1}^{R}\left(\frac{1}{t/R}\int_{(r-1)t/R}^{rt/R} Y(s)\mathrm{d}s - \frac{1}{t}\int_0^t Y(s)\mathrm{d}s\right)^2.$$

Batch-means method

How to construct confidence intervals?
For any R,

$$\sqrt{R}\left(\frac{1}{t}\int_0^t Y(s)\mathrm{d}s - z\right)\Big/ s_R(t) \to T_{R-1}$$

as $t \to \infty$. Here T_R is a Student-t distribution with R degrees of freedom, and

$$S_R(t) := \frac{1}{R-1}\sum_{r=1}^{R}\left(\frac{1}{t/R}\int_{(r-1)t/R}^{rt/R} Y(s)\mathrm{d}s - \frac{1}{t}\int_0^t Y(s)\mathrm{d}s\right)^2.$$

(And when R is large as well, this behaves as $\mathscr{N}(0,1)$.)

Chapter V
VARIANCE REDUCTION