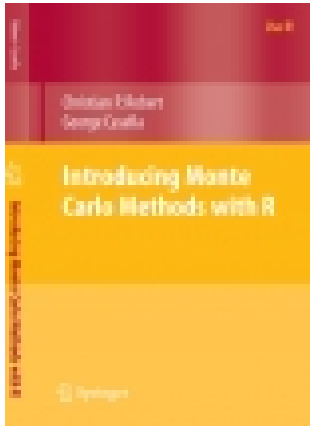
Monte Carlo Methods

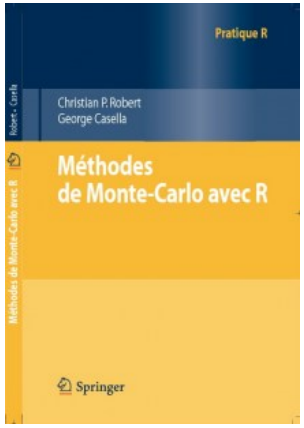
Christian P. Robert

Université Paris-Dauphine & University of Warwick
http://www.ceremade.dauphine.fr/~xian

September 8, 2024

textbook: *Introducing Monte Carlo Methods with R* by Christian. P. Robert and George Casella [trad. française 2010; japonaise 2011]





Outline

Motivations, Random Variable Generation Chapters 1 & 2

Monte Carlo Integration Chapter 3

Monte Carlo Optimization Chapter 4

Motivations

Monte Carlo

About

Monte Carlo is an official administrative area of Monaco, specifically the ward of Monte Carlo/Spélugues, where the Monte Carlo Casino is located.

[Wikipedia]

Monte Carlo

About

Monte Carlo methods, or Monte Carlo experiments, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems that might be deterministic in principle.

[Wikipedia]

[Stanislas Ulam]

Monte Carlo

About

Monte Carlo methods, or Monte Carlo experiments, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems that might be deterministic in principle. The name comes from the Monte Carlo Casino in Monaco, where the primary developer of the method, physicist Stanislaw Ulam, was inspired by his uncle's gambling habits.

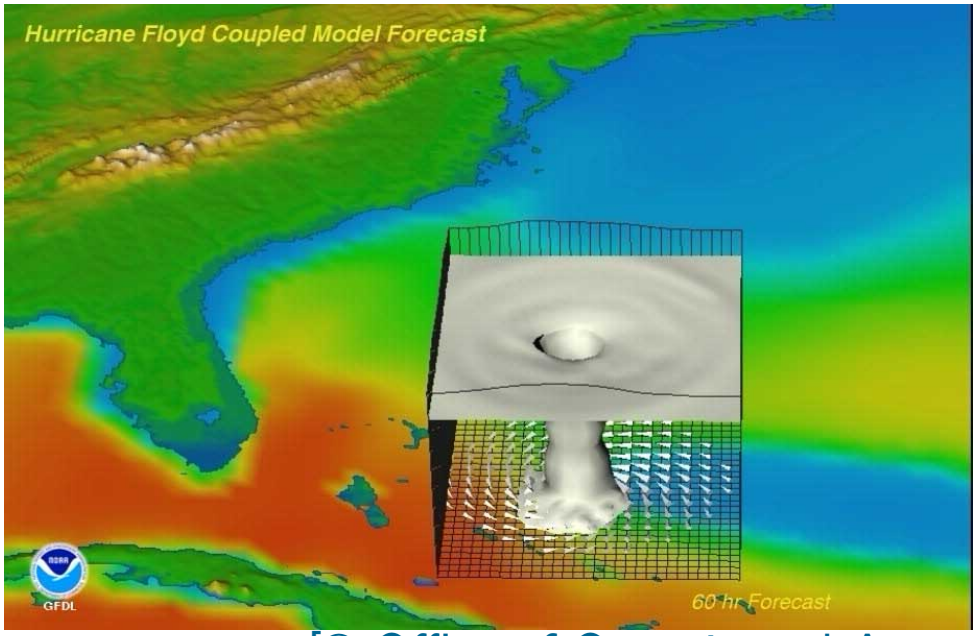
[Wikipedia]

[Cimetière du Montparnasse]

Illustrations

Necessity to “(re)produce chance” on a computer

- Evaluation of the behaviour of a complex system (network, computer program, queue, particle system, atmosphere, epidemics, economic actions, &tc)



[© Office of Oceanic and Atmospheric Research]

Illustrations

Necessity to “(re)produce chance” on a computer

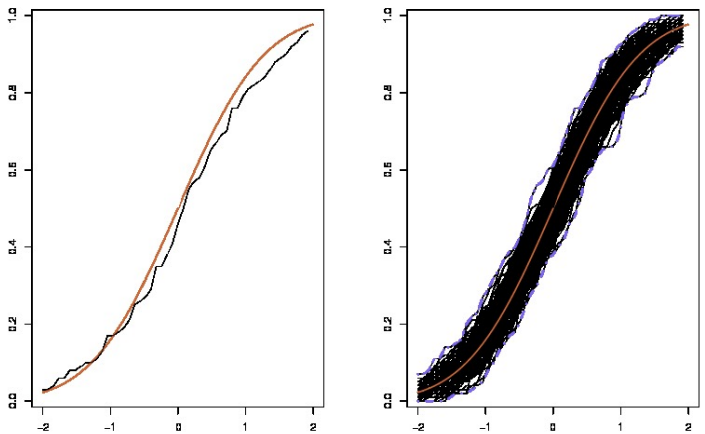
- Production of changing landscapes, characters, behaviours in computer games and flight simulators

[© guides.ign.com]

Illustrations

Necessity to "(re)produce chance" on a computer

- Determine probabilistic properties of a new statistical procedure or under an unknown distribution [bootstrap]



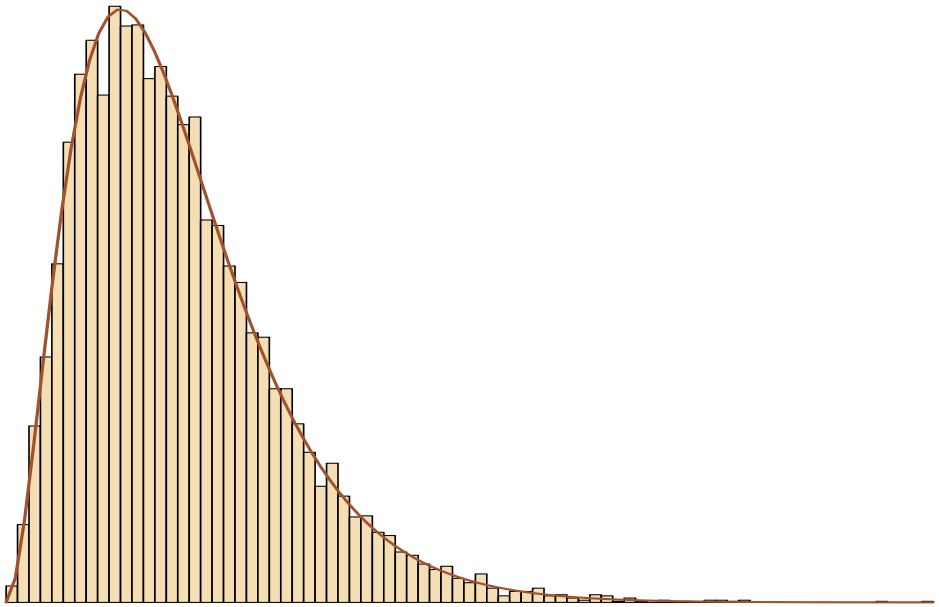
(left) Estimation of the cdf F from a normal sample of 100 points;

(right) variation of this estimation over 200 normal samples

Illustrations

Necessity to "(re)produce chance" on a computer

- Validation of a probabilistic model



Histogram of 10^3 variates from a distribution and fit by this distribution density

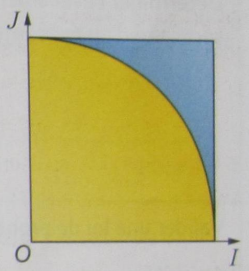
Illustrations

Necessity to "(re)produce chance" on a computer

- Approximation of a integral

14 Atteindre une cible

Sur une planche carrée de côté 1 m, on colorie en jaune le quart de disque comme sur la figure ci-contre :



Cette planche devient la cible sur laquelle Mehdi envoie des fléchettes au hasard et de façon aléatoire (on suppose qu'il ne rate jamais la planche).

On souhaite estimer la probabilité que Mehdi atteigne la zone colorée en jaune.

1. On se place dans le repère orthonormé donné sur la figure.

On choisit de simuler un lancer par le choix au hasard de deux réels x et y dans $[0\,;1]$.
Le point d'impact de la fléchette sera repéré par les coordonnées $(x\,;y)$.

a. Donner un critère sur x et y permettant de savoir si la zone colorée en jaune a été atteinte ou non.

b. Simuler plusieurs lancers à l'aide d'une calculatrice, d'un tableur ou d'un logiciel de programmation, puis estimer la probabilité que Mehdi atteigne la zone colorée en jaune.

2. En écrivant la probabilité cherchée comme le quotient de deux aires, calculer la valeur exacte de la probabilité pour que Mehdi atteigne la zone colorée en jaune.

[© my daughter's math book]

Illustrations

Necessity to "(re)produce chance" on a computer

- Maximisation of a weakly regular function/likelihood

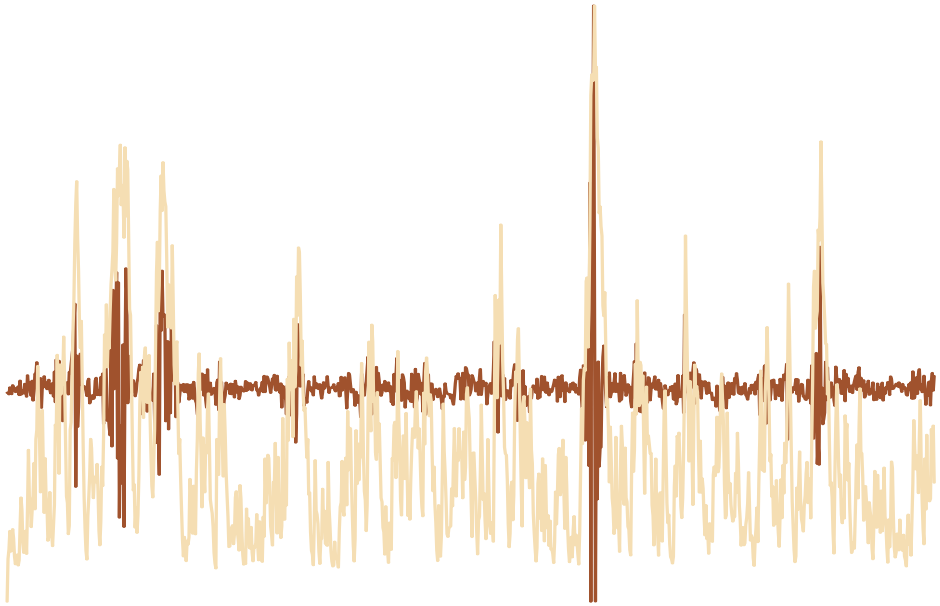
8			4		6			7
						4		
	1					6	5	
5		9		3		7	8	
				7				
	4	8		2		1		3
	5	2					9	
		1						
3			9		2			5

[© Dan Rice Sudoku blog]

Illustrations

Necessity to "(re)produce chance" on a computer

- Pricing of a complex financial product (exotic options)

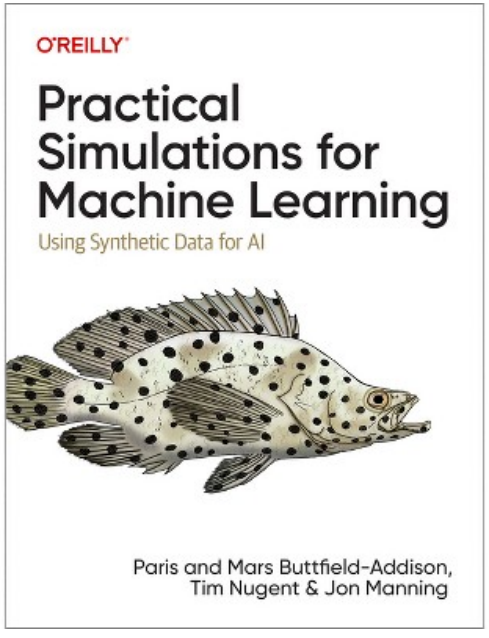


Simulation of a Garch(1,1) process and of its volatility (10^3 time units)

Illustrations

Necessity to “(re)produce chance” on a computer

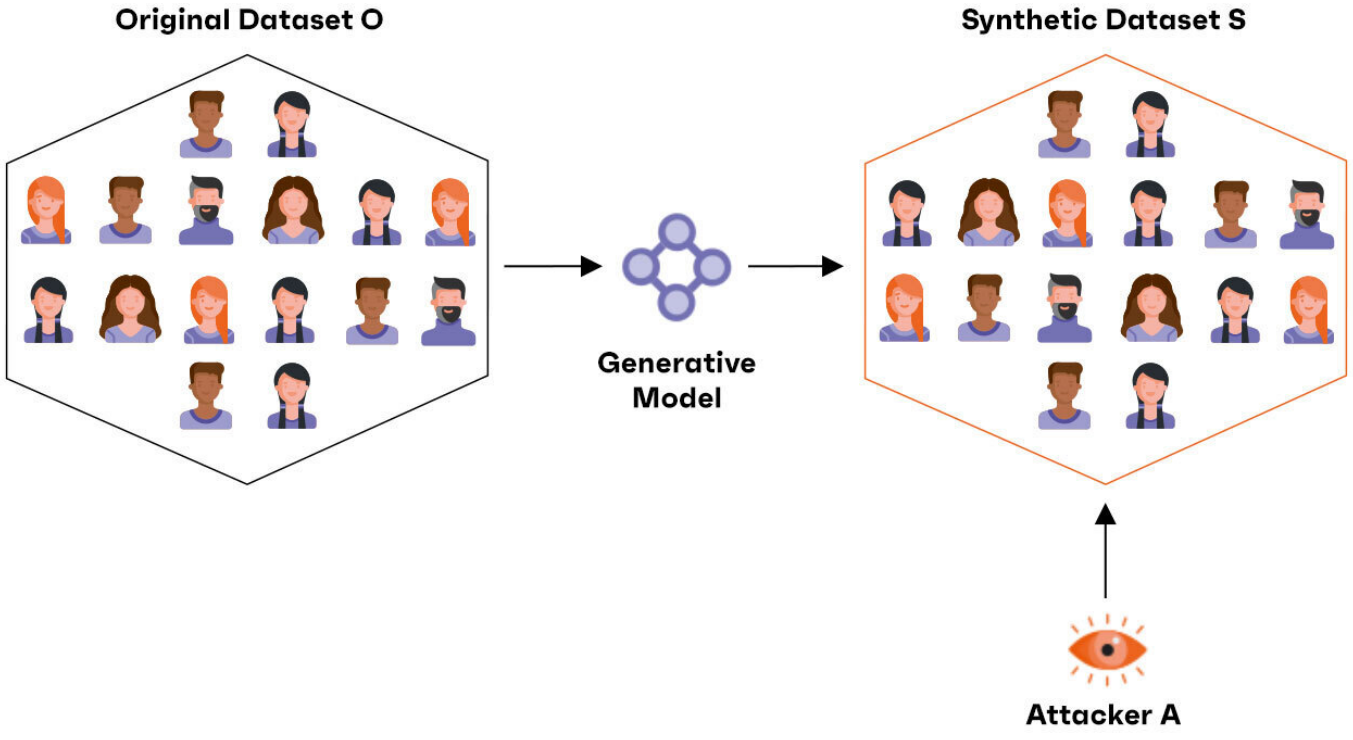
- Training neural networks with simulated data, as e.g. in deep leaning



Illustrations

Necessity to "(re)produce chance" on a computer

- Replacing true data with synthetic data, to combine privacy protection and learning



[©Clearbox AI]

Illustrations

Necessity to "(re)produce chance" on a computer

- Handling complex statistical problems by approximate Bayesian computation (ABC)

core principle

- Simulate a parameter value (at random) and pseudo-data from the likelihood until the pseudo-data is "close enough" to the observed data, then
- keep the corresponding parameter value

[Tavaré & al., 1999; Beaumont, Sisson & Tan, 2019]

Illustrations

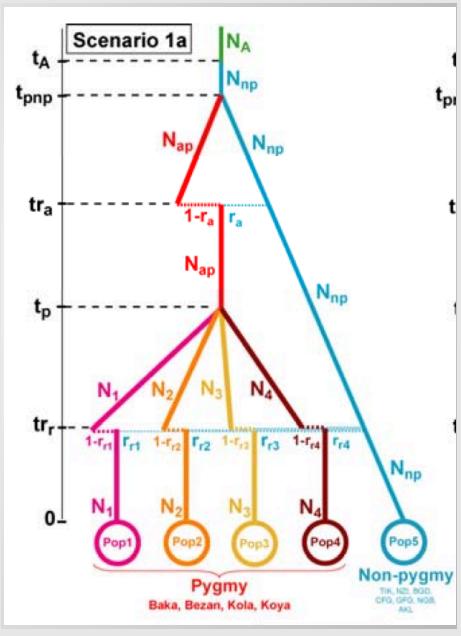
Necessity to "(re)produce chance" on a computer

- Handling complex statistical problems by approximate Bayesian computation (ABC)

demo-genetic inference

Genetic model of evolution from a common ancestor (MRCA) characterized by a set of parameters that cover historical, demographic, and genetic factors

Dataset of polymorphism (DNA sample) observed at the present time



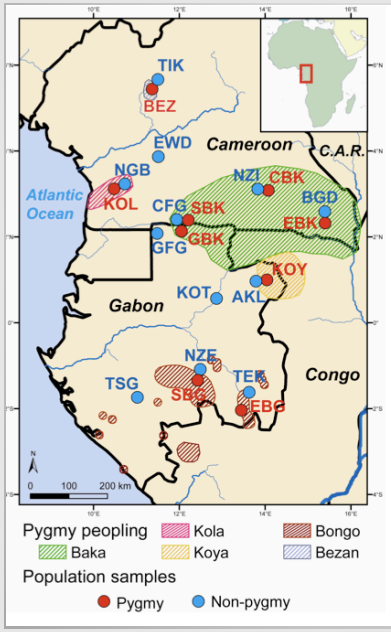
Illustrations

Necessity to "(re)produce chance" on a computer

- Handling complex statistical problems by approximate Bayesian computation (ABC)

Pygmies population demo-genetics

Pygmies populations: *do they have a common origin? when and how did they split from non-pygmies populations? were there more recent interactions between pygmies and non-pygmies populations?*



Crédit : Serge Bahuchet

604 individus, 12 populations non-pygmées, 9 populations pygmées, 28 marqueurs microsatellites

Verdu *et al.* (2009) *Current Biology* **19**: 312-318

Interlude # 1: counting socks

Interlude # 1: A pedestrian example

paired and orphan socks

A drawer contains an unknown number of socks, some of which can be paired and some of which are orphans (single). One takes at random 11 socks without replacement from this drawer: no pair can be found among those. What can we infer about the total number of socks in the drawer?

- sounds like an impossible task
- one observation $x = 11$ and two unknowns, n_{socks} and n_{pairs}
- writing the likelihood is a challenge [exercise]

Interlude # 1: A pedestrian example

paired and orphan socks

A drawer contains an unknown number of socks, some of which can be paired and some of which are orphans (single). One takes at random 11 socks without replacement from this drawer: no pair can be found among those. What can we infer about the total number of socks in the drawer?

- sounds like an impossible task
- one observation $x = 11$ and two unknowns, n_{socks} and n_{pairs}
- writing the likelihood is a challenge [exercise]

A priori on socks

Given parameters n_{socks} and n_{pairs}, set of socks

$$\mathcal{S} = \left\{ s_1, s_1, \ldots, s_{n_{\text{pairs}}}, s_{n_{\text{pairs}}}, s_{n_{\text{pairs}}+1}, \ldots, s_{n_{\text{socks}}} \right\}$$

and 11 socks picked at random from $\mathcal{S}$ give X unique socks.

Rassmus' reasoning

If you are a family of 3-4 persons then a guesstimate would be that you have something like 15 pairs of socks in store. It is also possible that you have much more than 30 socks. So as *a prior* for n_{socks} I'm going to use a negative binomial with mean 30 and standard deviation 15.
On $n_{\text{pairs}}/2n_{\text{socks}}$ I'm going to put a Beta *prior* distribution that puts most of the probability over the range 0.75 to 1.0,

[Rassmus Bååth's Research Blog, Oct 20th, 2014]

A priori on socks

Given parameters n_{socks} and n_{pairs}, set of socks

$$\mathcal{S} = \left\{ s_1, s_1, \ldots, s_{n_{\text{pairs}}}, s_{n_{\text{pairs}}}, s_{n_{\text{pairs}}+1}, \ldots, s_{n_{\text{socks}}} \right\}$$

and 11 socks picked at random from $\mathcal{S}$ give X unique socks.

Rassmus' reasoning

If you are a family of 3-4 persons then a guesstimate would be that you have something like 15 pairs of socks in store. It is also possible that you have much more than 30 socks. So as *a prior* for n_{socks} I'm going to use a negative binomial with mean 30 and standard deviation 15.
On $n_{\text{pairs}}/2n_{\text{socks}}$ I'm going to put a Beta *prior* distribution that puts most of the probability over the range 0.75 to 1.0,

[Rassmus Bååth's Research Blog, Oct 20th, 2014]

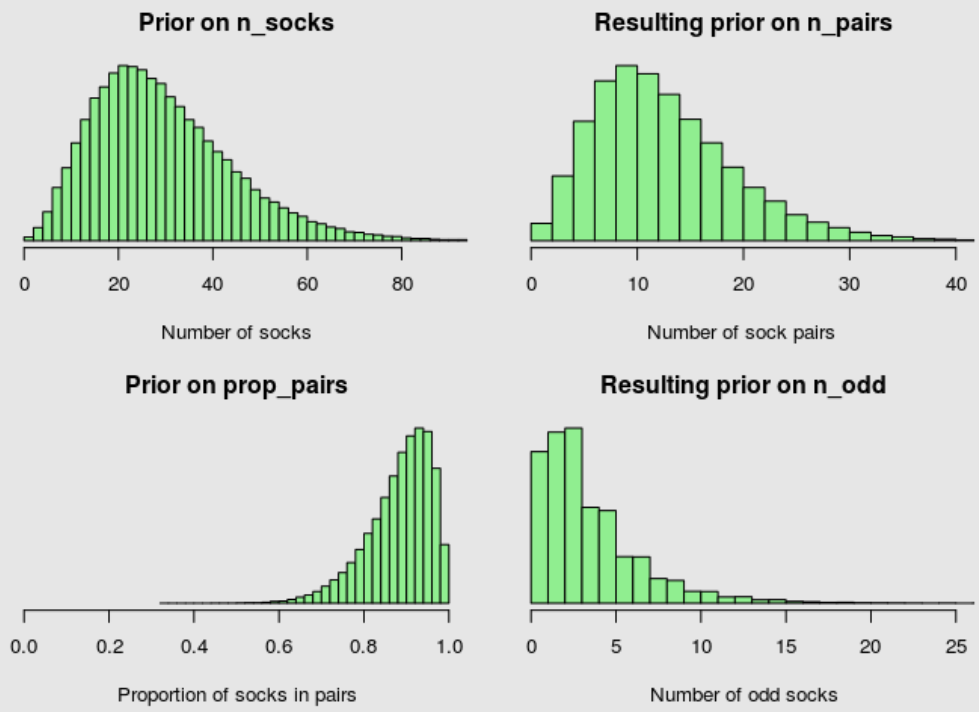
Simulating the experiment

Given a *prior* distribution on n_{socks} and n_{pairs},

$$n_{\text{socks}} \sim \mathcal{N}eg(30, 15) \quad n_{\text{pairs}} | n_{\text{socks}} \sim {}^{n_{\text{socks}}/2}\mathcal{B}e(15, 2)$$

possible to

1. generate new values of n_{socks} and n_{pairs},
2. generate a new observation of X, number of unique socks out of 11.
3. accept the pair $(n_{\text{socks}}, n_{\text{pairs}})$ if the realisation of X is equal to 11



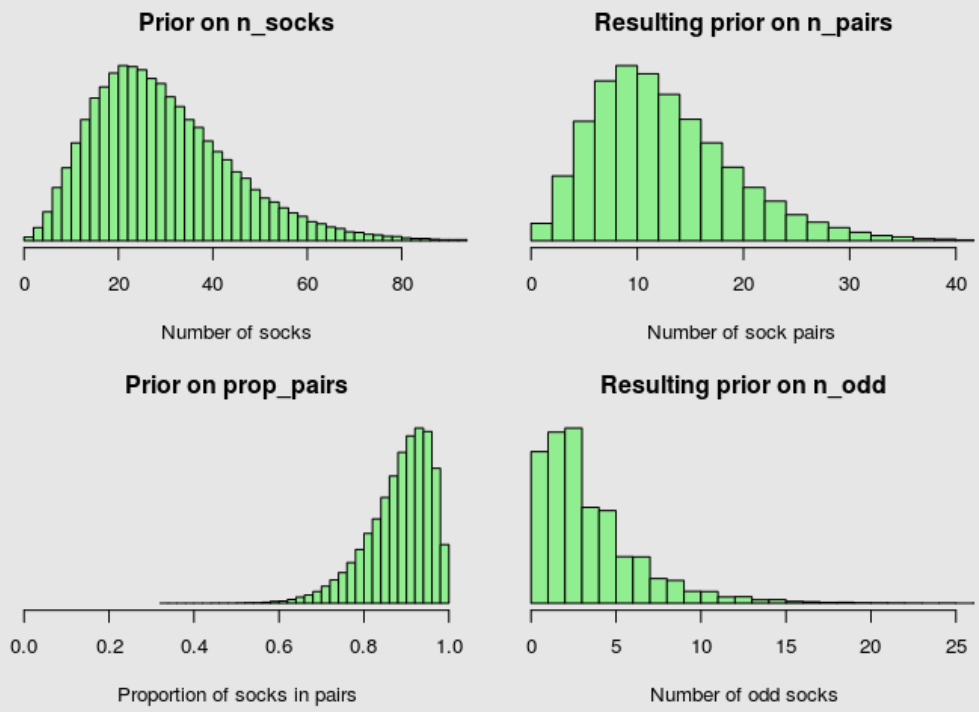
Simulating the experiment

Given a *prior* distribution on n_{socks} and n_{pairs},

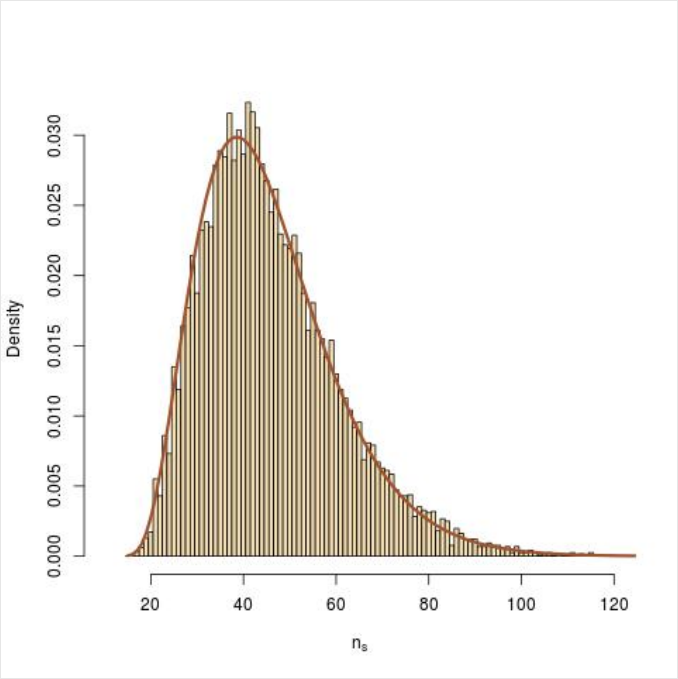
$$n_{\text{socks}} \sim \mathcal{N}eg(30, 15) \quad n_{\text{pairs}} | n_{\text{socks}} \sim {}^{n_{\text{socks}}/2}\mathcal{B}e(15, 2)$$

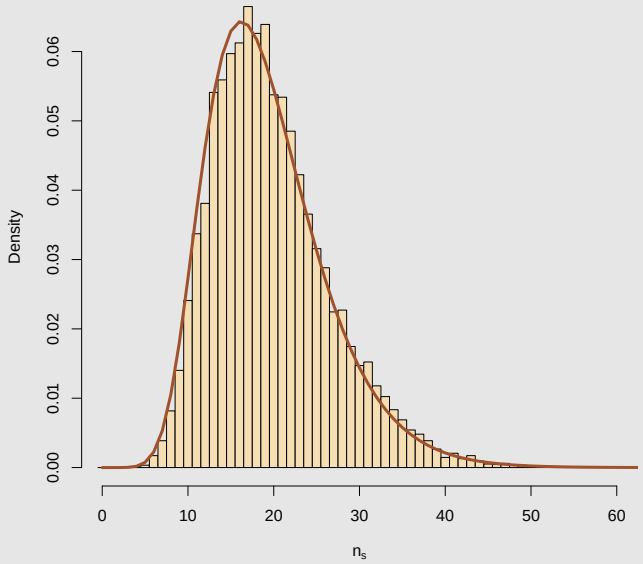
possible to

1. generate new values of n_{socks} and n_{pairs},
2. generate a new observation of X, number of unique socks out of 11.
3. accept the pair $(n_{\text{socks}}, n_{\text{pairs}})$ if the realisation of X is equal to 11



Meaning



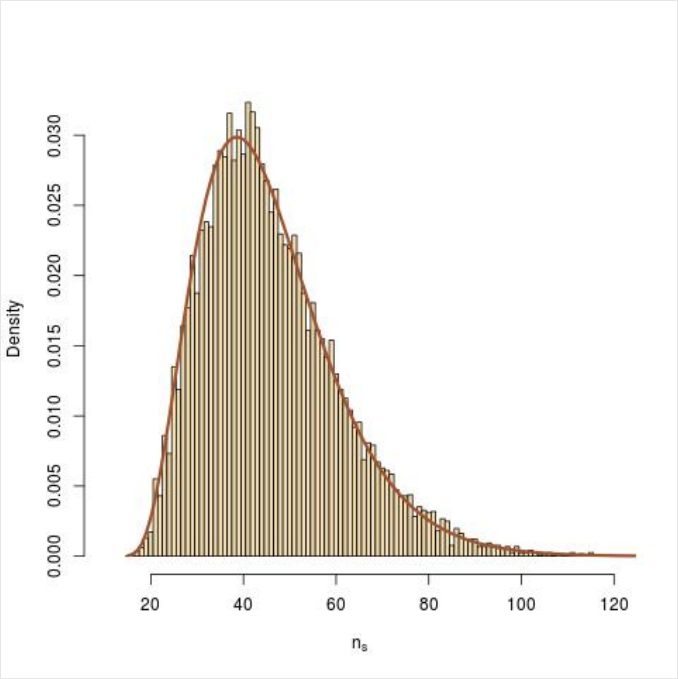


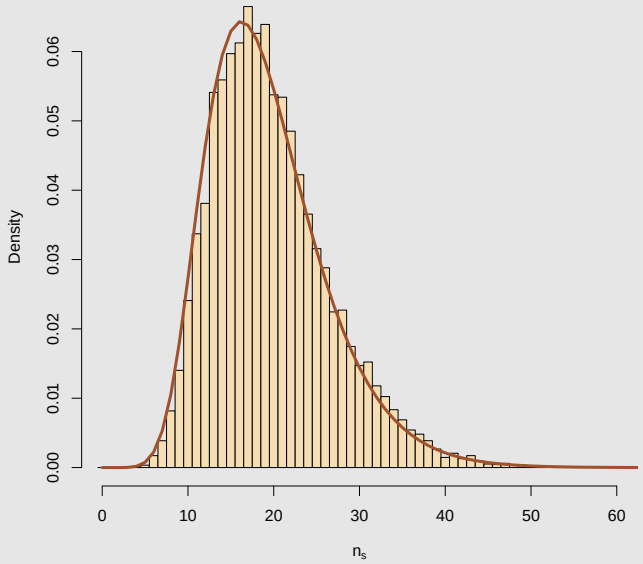
The outcome of this simulation method returns a distribution on the pair $(n_{\text{socks}}, n_{\text{pairs}})$ that is the conditional distribution of the pair given the observation $X = 11$

Proof: Generations from $\pi(n_{\text{socks}}, n_{\text{pairs}})$ are accepted with probability

$$\mathbb{P}\{X = 11 | (n_{\text{socks}}, n_{\text{pairs}})\}$$

Meaning





The outcome of this simulation method returns a distribution on the pair $(n_{\text{socks}}, n_{\text{pairs}})$ that is the conditional distribution of the pair given the observation $X = 11$

Proof: Hence accepted values distributed from

$$\pi(n_{\text{socks}}, n_{\text{pairs}}) \times \mathbb{P}\{X = 11 | (n_{\text{socks}}, n_{\text{pairs}})\} = \pi(n_{\text{socks}}, n_{\text{pairs}} | X = 11)$$

The Bayesian Perspective

In the Bayesian paradigm, the information brought by the data x, realization of

$$X \sim f(x|\theta),$$

is combined with **prior information** specified by *prior distribution* with density

$$\pi(\theta)$$

The Bayesian Perspective

In the Bayesian paradigm, the information brought by the data x, realization of

$$X \sim f(x|\theta),$$

is combined with **prior information** specified by *prior distribution* with density

$$\pi(\theta)$$

Posterior distribution

Information summary contained in a probability distribution, $\pi(\theta|x)$, called the **posterior distribution**

Derived from the *joint* distribution $f(x|\theta)\pi(\theta)$, according to

$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta},$$

[Bayes Theorem]

where

$$Z(x) = \int f(x|\theta)\pi(\theta)d\theta$$

is the *marginal density* of X also called the (Bayesian) evidence

[Gelman & al., 2020]

Posterior distribution

Information summary contained in a probability distribution, $\pi(\theta|x)$, called the **posterior distribution**
Derived from the *joint* distribution $f(x|\theta)\pi(\theta)$, according to

$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta},$$

[Bayes Theorem]

where

$$Z(x) = \int f(x|\theta)\pi(\theta)d\theta$$

is the *marginal density* of X also called the (Bayesian) evidence

[Gelman & al., 2020]

Posterior distribution

Information summary contained in a probability distribution, $\pi(\theta|x)$, called the **posterior distribution**
Derived from the *joint* distribution $f(x|\theta)\pi(\theta)$, according to

$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta},$$

[Bayes Theorem]

where

$$Z(x) = \int f(x|\theta)\pi(\theta)d\theta$$

is the *marginal density* of X also called the (Bayesian) evidence

[Gelman & al., 2020]

A typology of Bayes computational problems

(i). missing variable models

$$f(x^{obs}|\theta) = \int f^\star(x^{obs}, x^\star|\theta)\, dx^\star$$

(ii). use of complex parameter spaces, as for instance in constrained parameter sets like those resulting from imposing stationarity constraints in dynamic models;

(iii). use of a complex sampling model with an intractable likelihood, as for instance in some graphical models;

(iv). use of a huge dataset;

(v). use of a complex prior distribution (which may be the posterior distribution associated with an earlier sample);

(vi). use of a complex inferential procedure as for instance, **Bayes factors**

$$B_{01}^{\pi}(x) = \frac{P(\theta \in \Theta_0 \mid x)}{P(\theta \in \Theta_1 \mid x)} \Big/ \frac{\pi(\theta \in \Theta_0)}{\pi(\theta \in \Theta_1)}.$$

[Robert, 2001]

A typology of Bayes computational problems

(i). missing variable models

$$f(x^{obs}|\theta) = \int f^\star(x^{obs}, x^\star|\theta)\, dx^\star$$

(ii). use of complex parameter spaces, as for instance in constrained parameter sets like those resulting from imposing stationarity constraints in dynamic models;

(iii). use of a complex sampling model with an intractable likelihood, as for instance in some graphical models;

(iv). use of a huge dataset;

(v). use of a complex prior distribution (which may be the posterior distribution associated with an earlier sample);

(vi). use of a complex inferential procedure as for instance, **Bayes factors**

$$B^\pi_{01}(x) = \frac{P(\theta \in \Theta_0 \mid x)}{P(\theta \in \Theta_1 \mid x)} \Big/ \frac{\pi(\theta \in \Theta_0)}{\pi(\theta \in \Theta_1)}.$$

[Robert, 2001]

A typology of Bayes computational problems

(i). missing variable models

$$f(x^{obs}|\theta) = \int f^\star(x^{obs}, x^\star|\theta)\, dx^\star$$

(ii). use of complex parameter spaces, as for instance in constrained parameter sets like those resulting from imposing stationarity constraints in dynamic models;

(iii). use of a complex sampling model with an intractable likelihood, as for instance in some graphical models;

(iv). use of a huge dataset;

(v). use of a complex prior distribution (which may be the posterior distribution associated with an earlier sample);

(vi). use of a complex inferential procedure as for instance, **Bayes factors**

$$B^\pi_{01}(x) = \frac{P(\theta \in \Theta_0 \mid x)}{P(\theta \in \Theta_1 \mid x)} \Big/ \frac{\pi(\theta \in \Theta_0)}{\pi(\theta \in \Theta_1)}.$$

[Robert, 2001]

A typology of Bayes computational problems

(i). missing variable models

$$f(x^{obs}|\theta) = \int f^\star(x^{obs}, x^\star|\theta)\, dx^\star$$

(ii). use of complex parameter spaces, as for instance in constrained parameter sets like those resulting from imposing stationarity constraints in dynamic models;

(iii). use of a complex sampling model with an intractable likelihood, as for instance in some graphical models;

(iv). use of a huge dataset;

(v). use of a complex prior distribution (which may be the posterior distribution associated with an earlier sample);

(vi). use of a complex inferential procedure as for instance, **Bayes factors**

$$B_{01}^{\pi}(x) = \frac{P(\theta \in \Theta_0 \mid x)}{P(\theta \in \Theta_1 \mid x)} \Big/ \frac{\pi(\theta \in \Theta_0)}{\pi(\theta \in \Theta_1)}.$$

[Robert, 2001]

A typology of Bayes computational problems

(i). missing variable models

$$f(x^{obs}|\theta) = \int f^\star(x^{obs}, x^\star|\theta)\, dx^\star$$

(ii). use of complex parameter spaces, as for instance in constrained parameter sets like those resulting from imposing stationarity constraints in dynamic models;

(iii). use of a complex sampling model with an intractable likelihood, as for instance in some graphical models;

(iv). use of a huge dataset;

(v). use of a complex prior distribution (which may be the posterior distribution associated with an earlier sample);

(vi). use of a complex inferential procedure as for instance, **Bayes factors**

$$B_{01}^{\pi}(x) = \frac{P(\theta \in \Theta_0 \mid x)}{P(\theta \in \Theta_1 \mid x)} \Big/ \frac{\pi(\theta \in \Theta_0)}{\pi(\theta \in \Theta_1)}.$$

[Robert, 2001]

Random variable generation

1 Motivations

2 Random variable generation
- Uniform generators
- Interlude #2: Fibonacci generators
- Beyond Uniform distributions
- Transformation methods
- Accept-Reject Methods
- Interlude #3: Log-concave densities
- Ratio of Uniforms

3 Monte Carlo integration

4 Monte Carlo Optimization

Random variable generation

- Rely on the possibility of producing (computer-wise) an endless flow of random variables (usually iid) from well-known distributions
- Given a uniform random number generator, illustration of methods that produce random variables from both standard and nonstandard distributions

Random variable generation

- Rely on the possibility of producing (computer-wise) an endless flow of random variables (usually iid) from well-known distributions
- Given a uniform random number generator, illustration of methods that produce random variables from both standard and nonstandard distributions

Pseudo-random generator

Pivotal element/building block of simulation: always requires availability of uniform $\mathscr{U}(0,1)$ random variables

[© MMP World]

Pseudo-random generator

Pivotal element/building block of simulation: always requires availability of uniform $\mathscr{U}(0,1)$ random variables

0.1333139
0.3026299
0.4342966
0.2395357
0.3223723
0.8531162
0.3921457
0.7625259
0.1701947
0.2816627
⋮

[R `runif(10)`]

Pseudo-random generator

Pivotal element/building block of simulation: always requires availability of uniform $\mathscr{U}(0,1)$ random variables

Definition (Pseudo-random generator)

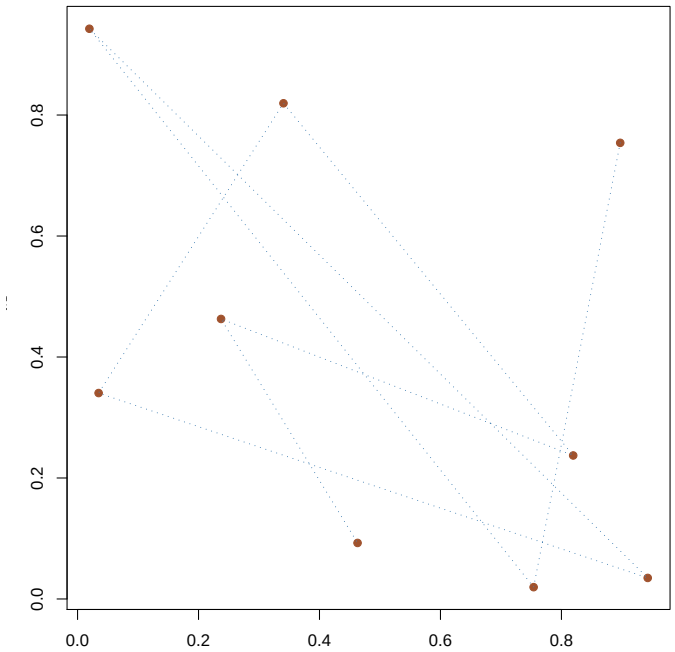
A *pseudo-random generator* is a deterministic function $\mathbf{f}$ from $]0,1[$ to $]0,1[$ such that, for any starting value u_0 and any n, the sequence

$$\{u_0, \mathbf{f}(u_0), \mathbf{f}(\mathbf{f}(u_0)), \ldots, \mathbf{f}^n(u_0)\}$$

behaves (statistically) like an iid $\mathscr{U}(0,1)$ sequence

Pseudo-random generator

Pivotal element/building block of simulation: always requires availability of uniform $\mathcal{U}(0,1)$ random variables



10 steps (u_t, u_{t+1}) of a uniform generator

Philosophical foray

¡Paradox!

While avoiding randomness, the deterministic sequence

$$(u_0, u_1 = f(u_0), \ldots, u_n = f(u_{n-1}))$$

must resemble a random sequence!

Debate on whether or not true randomness does exist (Laplace's demon versus Schroedinger's cat), in which case pseudo random generators are *not* random *(von Neuman's state of sin)*

Philosophical foray

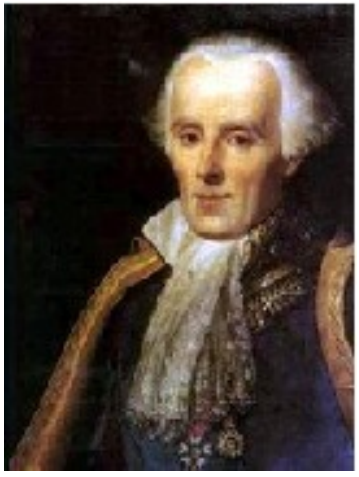
¡Paradox!

While avoiding randomness, the deterministic sequence

$$(u_0, u_1 = f(u_0), \ldots, u_n = f(u_{n-1}))$$

must resemble a random sequence!

Debate on whether or not true randomness does exist (Laplace's demon versus Schroedinger's cat), in which case pseudo random generators are *not* random *(von Neuman's state of sin)*



Philosophical foray

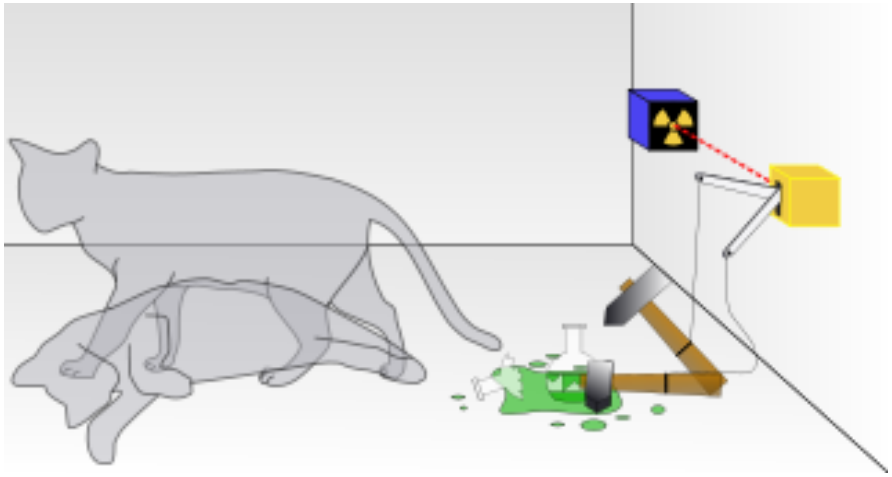
¡Paradox!

While avoiding randomness, the deterministic sequence

$$(u_0, u_1 = f(u_0), \ldots, u_n = f(u_{n-1}))$$

must resemble a random sequence!

Debate on whether or not true randomness does exist (Laplace's demon versus Schroedinger's cat), in which case pseudo random generators are *not* random *(von Neuman's state of sin)*



Philosophical foray

¡Paradox!

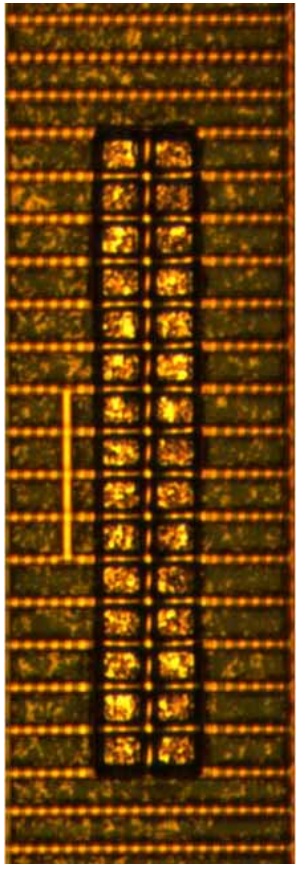
While avoiding randomness, the deterministic sequence

$$(u_0, u_1 = f(u_0), \ldots, u_n = f(u_{n-1}))$$

must resemble a random sequence!

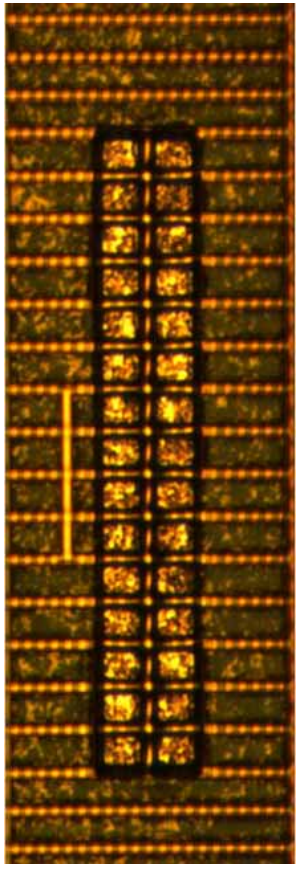
Debate on whether or not true randomness does exist (Laplace's demon versus Schroedinger's cat), in which case pseudo random generators are *not* random *(von Neuman's state of sin)*

True random generators



Intel circuit producing "truly random" numbers:
There is no reason physical generators should be "more" random than congruential (deterministic) pseudo-random generators, as those are valid generators, i.e. their distribution is exactly known (e.g., uniform) and, in the case of parallel generations, completely independent

True random generators



Intel generator satisfies all benchmarks of "randomness" maintained by NIST:
Skepticism about physical devices, when compared with mathematical functions, because of (a) non-reproducibility and (b) instability of the device, which means that proven uniformity at time t does not induce uniformity at time $t + 1$

Desiderata and limitations

- Production of a *deterministic* sequence of values in $[0, 1]$ which imitates a sequence of *iid* uniform random variables $\mathcal{U}_{[0,1]}$.
- Can't use the physical imitation of a "random draw" `[no guarantee of uniformity, no reproducibility]`
- *Random* sequence in the sense: Having generated $(X_1, \cdots, X_n)$, knowledge of X_n [or of $(X_1, \cdots, X_n)$] imparts no discernible knowledge of the value of X_{n+1}.
- Deterministic: Given the initial value X_0, sample $(X_1, \cdots, X_n)$ always the same
- Validity of a random number generator based on a single sample $X_1, \cdots, X_n$ when n tends to $+\infty$, **not** on replications

$$(X_{11}, \cdots, X_{1n}), (X_{21}, \cdots, X_{2n}), \ldots (X_{k1}, \cdots, X_{kn})$$

where n fixed and k tends to infinity.

Desiderata and limitations

- Production of a *deterministic* sequence of values in $[0, 1]$ which imitates a sequence of *iid* uniform random variables $\mathcal{U}_{[0,1]}$.
- Can't use the physical imitation of a "random draw" `[no guarantee of uniformity, no reproducibility]`
- *Random* sequence in the sense: Having generated $(X_1, \cdots, X_n)$, knowledge of X_n [or of $(X_1, \cdots, X_n)$] imparts no discernible knowledge of the value of X_{n+1}.
- Deterministic: Given the initial value X_0, sample $(X_1, \cdots, X_n)$ always the same
- Validity of a random number generator based on a single sample $X_1, \cdots, X_n$ when n tends to $+\infty$, **not** on replications

$$(X_{11}, \cdots, X_{1n}), (X_{21}, \cdots, X_{2n}), \ldots (X_{k1}, \cdots, X_{kn})$$

where n fixed and k tends to infinity.

Desiderata and limitations

- Production of a *deterministic* sequence of values in $[0, 1]$ which imitates a sequence of *iid* uniform random variables $\mathcal{U}_{[0,1]}$.
- Can't use the physical imitation of a "random draw" `[no guarantee of uniformity, no reproducibility]`
- *Random* sequence in the sense: Having generated $(X_1, \cdots, X_n)$, knowledge of X_n [or of $(X_1, \cdots, X_n)$] imparts no discernible knowledge of the value of X_{n+1}.
- Deterministic: Given the initial value X_0, sample $(X_1, \cdots, X_n)$ always the same
- Validity of a random number generator based on a single sample $X_1, \cdots, X_n$ when n tends to $+\infty$, **not** on replications

$$(X_{11}, \cdots, X_{1n}), (X_{21}, \cdots, X_{2n}), \ldots (X_{k1}, \cdots, X_{kn})$$

where n fixed and k tends to infinity.

Desiderata and limitations

- Production of a *deterministic* sequence of values in $[0, 1]$ which imitates a sequence of *iid* uniform random variables $\mathcal{U}_{[0,1]}$.
- Can't use the physical imitation of a "random draw" `[no guarantee of uniformity, no reproducibility]`
- *Random* sequence in the sense: Having generated $(X_1, \cdots, X_n)$, knowledge of X_n [or of $(X_1, \cdots, X_n)$] imparts no discernible knowledge of the value of X_{n+1}.
- Deterministic: Given the initial value X_0, sample $(X_1, \cdots, X_n)$ always the same
- Validity of a random number generator based on a single sample $X_1, \cdots, X_n$ when n tends to $+\infty$, **not** on replications

$$(X_{11}, \cdots, X_{1n}), (X_{21}, \cdots, X_{2n}), \ldots (X_{k1}, \cdots, X_{kn})$$

where n fixed and k tends to infinity.

Desiderata and limitations

- Production of a *deterministic* sequence of values in $[0, 1]$ which imitates a sequence of *iid* uniform random variables $\mathcal{U}_{[0,1]}$.
- Can't use the physical imitation of a "random draw" `[no guarantee of uniformity, no reproducibility]`
- *Random* sequence in the sense: Having generated $(X_1, \cdots, X_n)$, knowledge of X_n [or of $(X_1, \cdots, X_n)$] imparts no discernible knowledge of the value of X_{n+1}.
- Deterministic: Given the initial value X_0, sample $(X_1, \cdots, X_n)$ always the same
- Validity of a random number generator based on a single sample $X_1, \cdots, X_n$ when n tends to $+\infty$, **not** on replications

$$(X_{11}, \cdots, X_{1n}), (X_{21}, \cdots, X_{2n}), \ldots (X_{k1}, \cdots, X_{kn})$$

where n fixed and k tends to infinity.

Uniform pseudo-random generator

Algorithm starting from an initial value $0 \leq u_0 \leq 1$ and a transformation D, which produces a sequence

$$(u_i) = (D^i(u_0))$$

in $[0, 1]$.
For all n,

$$(u_1, \cdots, u_n)$$

reproduces the behavior of an iid $\mathcal{U}_{[0,1]}$ sample $(V_1, \cdots, V_n)$ when compared through usual statistical tests (e.g., Kolmogorov)

Uniform pseudo-random generator

Algorithm starting from an initial value $0 \leq u_0 \leq 1$ and a transformation D, which produces a sequence

$$(u_i) = (D^i(u_0))$$

in $[0, 1]$.
For all n,

$$(u_1, \cdots, u_n)$$

reproduces the behavior of an iid $\mathcal{U}_{[0,1]}$ sample $(V_1, \cdots, V_n)$ when compared through usual statistical tests (e.g., Kolmogorov)

Uniform pseudo-random generator (2)

- Validity means the sequence $U_1, \cdots, U_n$ leads to accept the hypothesis

$$H : U_1, \cdots, U_n \quad \text{are iid} \quad \mathcal{U}_{[0,1]}.$$

- The set of tests used is generally of some consequence
  - Kolmogorov–Smirnov and other nonparametric tests
  - Time series methods, for correlation between U_i and $(U_{i-1}, \cdots, U_{i-k})$
  - Marsaglia's battery of tests called *Die Hard* (!)

[Diehard, Marsaglia, 1995, 2006]

Uniform pseudo-random generator (2)

- Validity means the sequence $\mathrm{U}_1, \cdots, \mathrm{U}_n$ leads to accept the hypothesis

$$\mathrm{H} : \mathrm{U}_1, \cdots, \mathrm{U}_n \quad \text{are iid} \quad \mathscr{U}_{[0,1]}.$$

- The set of tests used is generally of some consequence
  - Kolmogorov–Smirnov and other nonparametric tests
  - Time series methods, for correlation between U_i and $(\mathrm{U}_{i-1}, \cdots, \mathrm{U}_{i-k})$
  - Marsaglia's battery of tests called *Die Hard* (!)

[Diehard, Marsaglia, 1995, 2006]

Usual generators

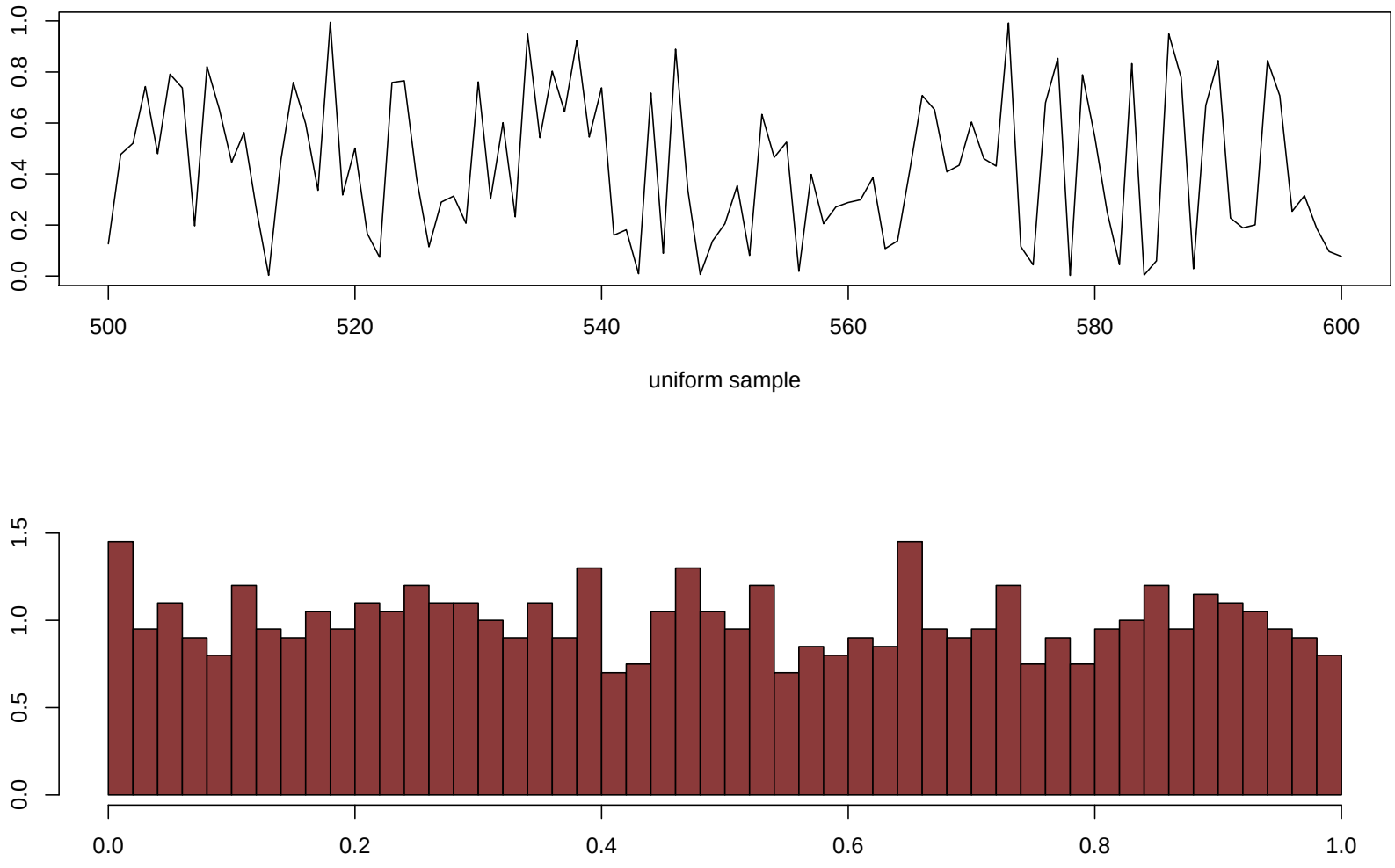
In R and S-plus, procedure `runif()`

```
The Uniform Distribution

Description:
‘runif’ generates random deviates.

Example:
     u <- runif(20)

‘.Random.seed’ is an integer vector, containing
the random number generator state for random
number generation in R. It can be saved and
restored, but should not be altered by users.
```


0.0
0.2
0.4
0.6
0.8
1.0
500
520
540
560
580
600
uniform sample
0.0
0.5
1.0
1.5
0.0
0.2
0.4
0.6
0.8
1.0

Usual generators (2)

In C, procedure `rand()` or `random()`

```
SYNOPSIS
       #include <stdlib.h>
       long int random(void);
DESCRIPTION
The random() function uses a non-linear additive
feedback random number generator employing a
default table of size 31 long integers to return
successive pseudo-random numbers in the range
from 0 to RAND_MAX. The period of this random
generator is very large, approximately
16*((2**31)-1).
RETURN VALUE
random() returns a value between 0 and RAND_MAX.
```

Usual generators (3)

In Matlab and Octave, procedure `rand()`

```
RAND Uniformly distributed pseudorandom numbers.
   R = RAND(M,N) returns an M-by-N matrix containing
   pseudorandom values drawn from the standard uniform
   distribution on the open interval(0,1).

   The sequence of numbers produced by RAND is
   determined by the internal state of the uniform
   pseudorandom number generator that underlies RAND,
   RANDI, and RANDN.
```

Usual generators (4)

In python, procedure `random.uniform()`

```
random.uniform(a, b)

    Return a random floating-point number N
    such that  a <= N <= b  for  a <= b and
    b <= N <= a for b < a.

    The end-point value b may or may not be
    included  in  the  range  depending  on
    floating-point rounding in the expression
    a + (b-a) * random().
```

The R generator options

Options for R `runif()`

```
Details
The currently available RNG kinds are given below. kind is partially ma

"Wichmann-Hill"
The seed, .Random.seed[-1] == r[1:3] is an integer vector of length 3,

"Marsaglia-Multicarry":
A multiply-with-carry RNG is used, as recommended by George Marsaglia i

It exhibits 40 clear failures in L'Ecuyer's TestU01 Crush suite. Combin

The seed is two integers (all values allowed).
```

The R generator options

Options for R `runif()`

```
"Super-Duper":
Marsaglia's famous Super-Duper from the 70's. This is the original vers

We use the implementation by Reeds et al (1982{84).

The two seeds are the Tausworthe and congruence long integers, respecti

It exhibits 25 clear failures in the TestU01 Crush suite (L'Ecuyer, 200

"Mersenne-Twister":
From Matsumoto and Nishimura (1998); code updated in 2002. A twisted GF

R uses its own initialization method due to B. D. Ripley and is not aff

It exhibits 2 clear failures in each of the TestU01 Crush and the BigCr

"Knuth-TAOCP-2002":
A 32-bit integer GFSR using lagged Fibonacci sequences with subtraction

"Knuth-TAOCP":
```

The R generator options

Options for R `runif()`

```
normal.kind can be "Kinderman-Ramage", "Buggy Kinderman-Ramage" (not fo

sample.kind can be "Rounding" or "Rejection", or partial matches to the

set.seed uses a single integer argument to set as many seeds as are req

The use of kind = NULL, normal.kind = NULL or sample.kind = NULL in RNG
```

A simple uniform generator

The congruencial generator on $\{1, 2, \ldots, M\}$

$$f(x) = (ax + b) \bmod (M)$$

has a period equal to M for proper choices of (a, b) and becomes a generator on $]0, 1[$ when dividing by $M + 1$

A simple uniform generator

The congruencial generator on $\{1, 2, \ldots, M\}$

$$f(x) = (ax + b) \bmod (M)$$

has a period equal to M for proper choices of (a, b) and becomes a generator on $]0, 1[$ when dividing by $M + 1$

Example

Take

$$f(x) = (69069069x + 12345) \bmod (2^{32})$$

and produce

... 518974515 2498053016 1113825472 1109377984 ...

i.e.

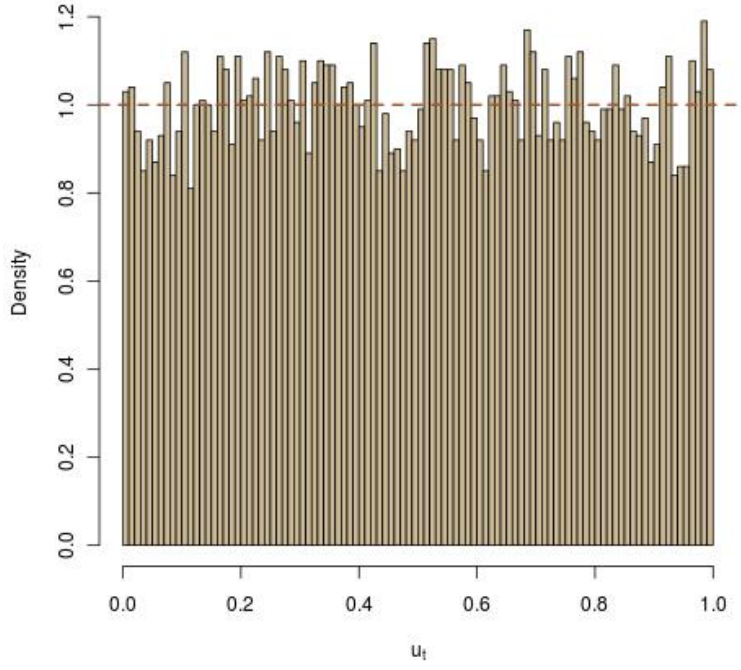
... 0.1208332 0.5816233 0.2593327 0.2582972 ...

A simple uniform generator

The congruencial generator on $\{1, 2, \ldots, M\}$

$$f(x) = (ax + b) \bmod (M)$$

has a period equal to M for proper choices of (a, b) and becomes a generator on $]0, 1[$ when dividing by $M + 1$

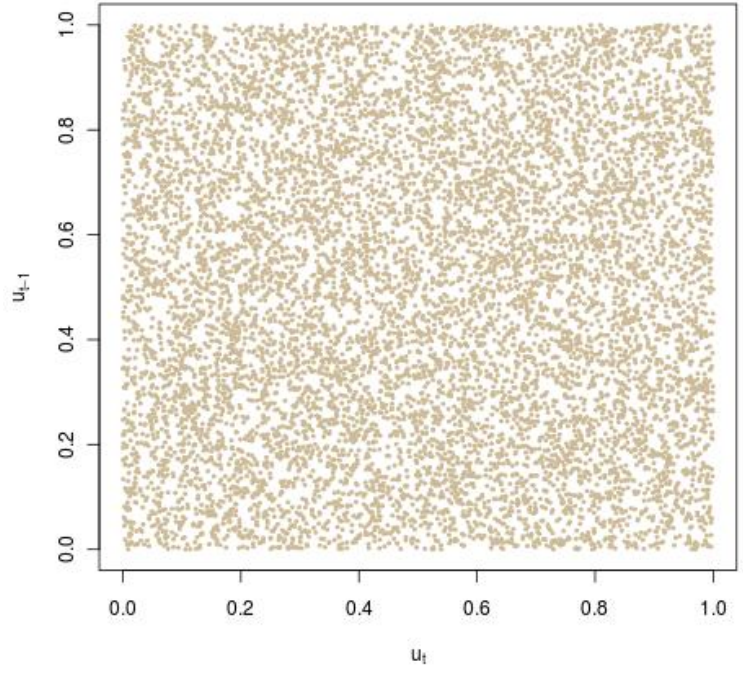


A simple uniform generator

The congruencial generator on $\{1, 2, \ldots, M\}$

$$f(x) = (ax + b) \bmod (M)$$

has a period equal to M for proper choices of (a, b) and becomes a generator on $]0, 1[$ when dividing by $M + 1$



Approximating π

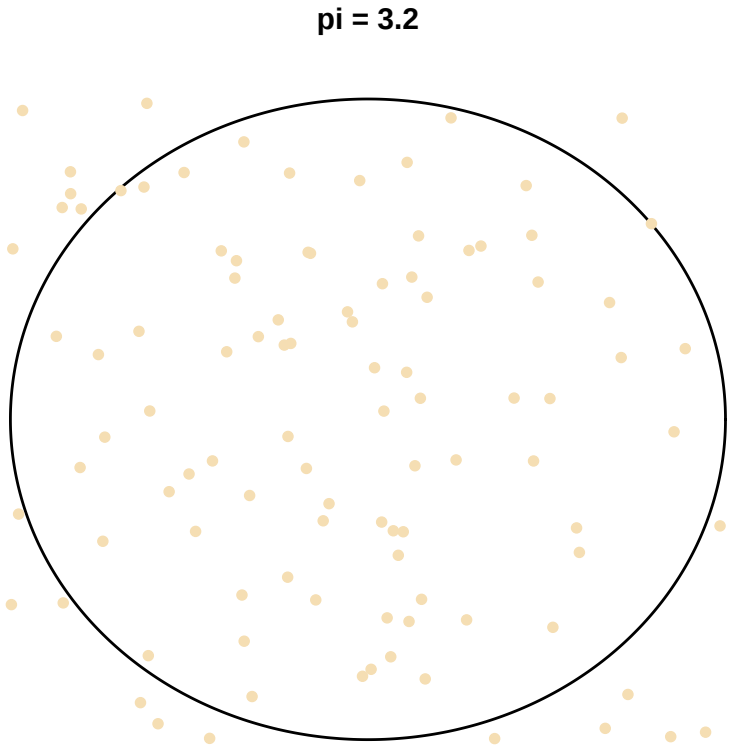
My daughter's pseudo-code:

N=1000
$\hat{\pi} = 0$
for I=1,N **do**
  X=RDN(1), Y=RDN(1)
  if $X^2 + Y^2 < 1$ **then**
    $\hat{\pi} = \hat{\pi} + 1$
  end if
end for
return 4*$\hat{\pi}$/N

Approximating π

My daughter's pseudo-code:

N=1000
$\hat{\pi} = 0$
for I=1,N **do**
  X=RDN(1), Y=RDN(1)
  if $X^2 + Y^2 < 1$ **then**
    $\hat{\pi} = \hat{\pi} + 1$
  end if
end for
return 4*$\hat{\pi}$/N

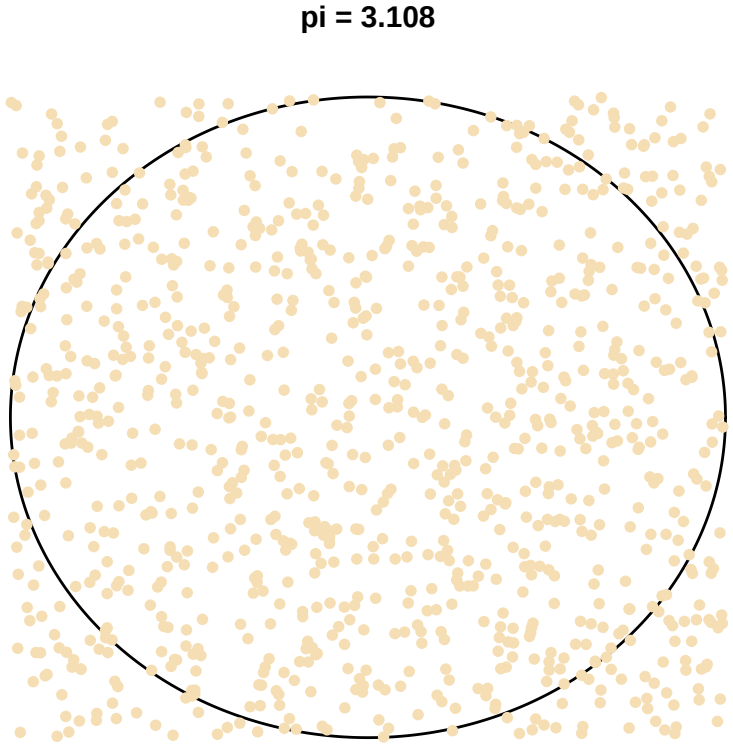


100 simulations

Approximating π

My daughter's pseudo-code:

N=1000
$\hat{\pi} = 0$
for I=1,N **do**
  X=RDN(1), Y=RDN(1)
  if $X^2 + Y^2 < 1$ **then**
    $\hat{\pi} = \hat{\pi} + 1$
  end if
end for
return 4*$\hat{\pi}$/N

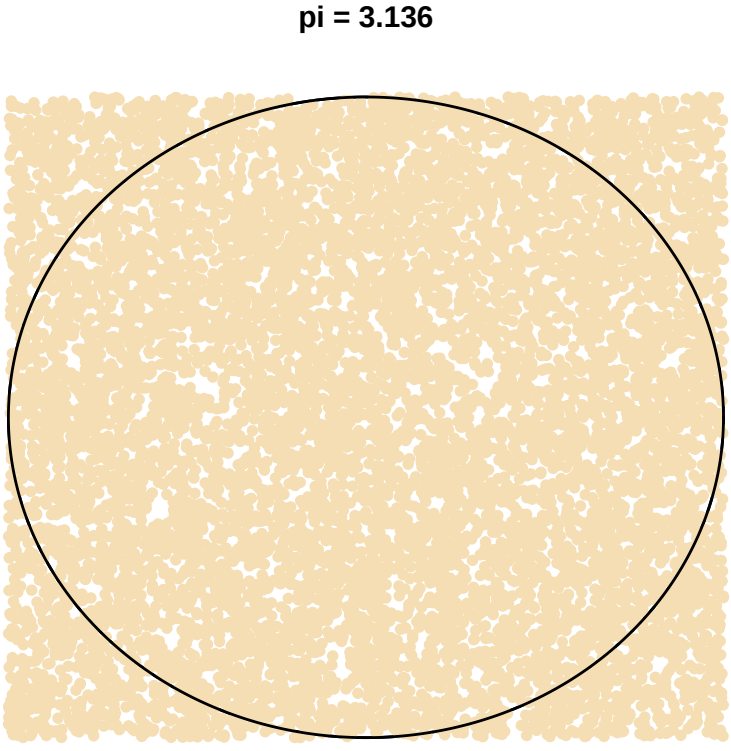


1000 simulations

Approximating π

My daughter's pseudo-code:

N=1000
$\hat{\pi} = 0$
for I=1,N **do**
  X=RDN(1), Y=RDN(1)
  if $X^2 + Y^2 < 1$ **then**
    $\hat{\pi} = \hat{\pi} + 1$
  end if
end for
return 4*$\hat{\pi}$/N

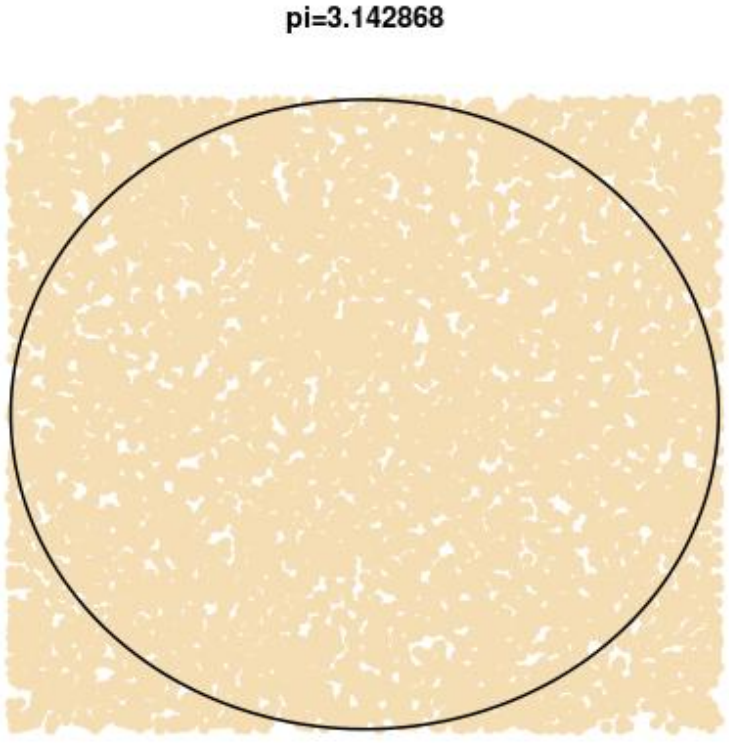


10,000 simulations

Approximating π

My daughter's pseudo-code:

N=1000
$\hat{\pi} = 0$
for I=1,N **do**
  X=RDN(1), Y=RDN(1)
  if $X^2 + Y^2 < 1$ **then**
    $\hat{\pi} = \hat{\pi} + 1$
  end if
end for
return 4*$\hat{\pi}$/N



10^6 simulations

Interlude #2: Fibonacci generators

Interlude #2: Fibonacci generators

Recall that Fibonacci sequence defined by recurrence

$$S_n = S_{n-1} + S_{n-2}$$

that can be generalised into

$$S_n \equiv S_{n-j} \star S_{n-k} \pmod{m}, 0 < j < k$$

where

- m usually a power of 2 ($m = 2^M$),
- $\star$ denotes a general binary operation (addition, subtraction, multiplication, XOR).

Interlude #2: Fibonacci generators

Recall that Fibonacci sequence defined by recurrence

$$S_n = S_{n-1} + S_{n-2}$$

that can be generalised into

$$S_n \equiv S_{n-j} \star S_{n-k} \pmod{m}, 0 < j < k$$

where

- m usually a power of 2 ($m = 2^M$),
- $\star$ denotes a general binary operation (addition, subtraction, multiplication, XOR).

Interlude #2: Fibonacci generators

Maximum period of Fibonacci generators depends on choice of $\star$.

- For addition or subtraction, $\max = (2k - 1) \times 2^{M-1}$
- For multiplication, $\max = (2k - 1) \times 2^{M-3}$
- For bitwise XOR, $\max = 2^{k-1}$

Examples of valid* (j, k)'s:

$$(24, 55), (38, 89), (37, 100), (30, 127), (83, 258), (107, 378), (273, 607)$$

$$(576, 3217), (4187, 9689), (7083, 19937), (9739, 23209)$$

Example of the (default) Mersenne twister with period $2^{19937} - 1$

[Matsumoto & Nishimura, 1997]

*Polynomial must be primitive over the integers mod 2.

Interlude #2: Fibonacci generators

Maximum period of Fibonacci generators depends on choice of $\star$.

- For addition or subtraction, $\max = (2k - 1) \times 2^{M-1}$
- For multiplication, $\max = (2k - 1) \times 2^{M-3}$
- For bitwise XOR, $\max = 2^{k-1}$

Examples of valid* (j, k)'s:

$$(24, 55), (38, 89), (37, 100), (30, 127), (83, 258), (107, 378), (273, 607)$$

$$(576, 3217), (4187, 9689), (7083, 19937), (9739, 23209)$$

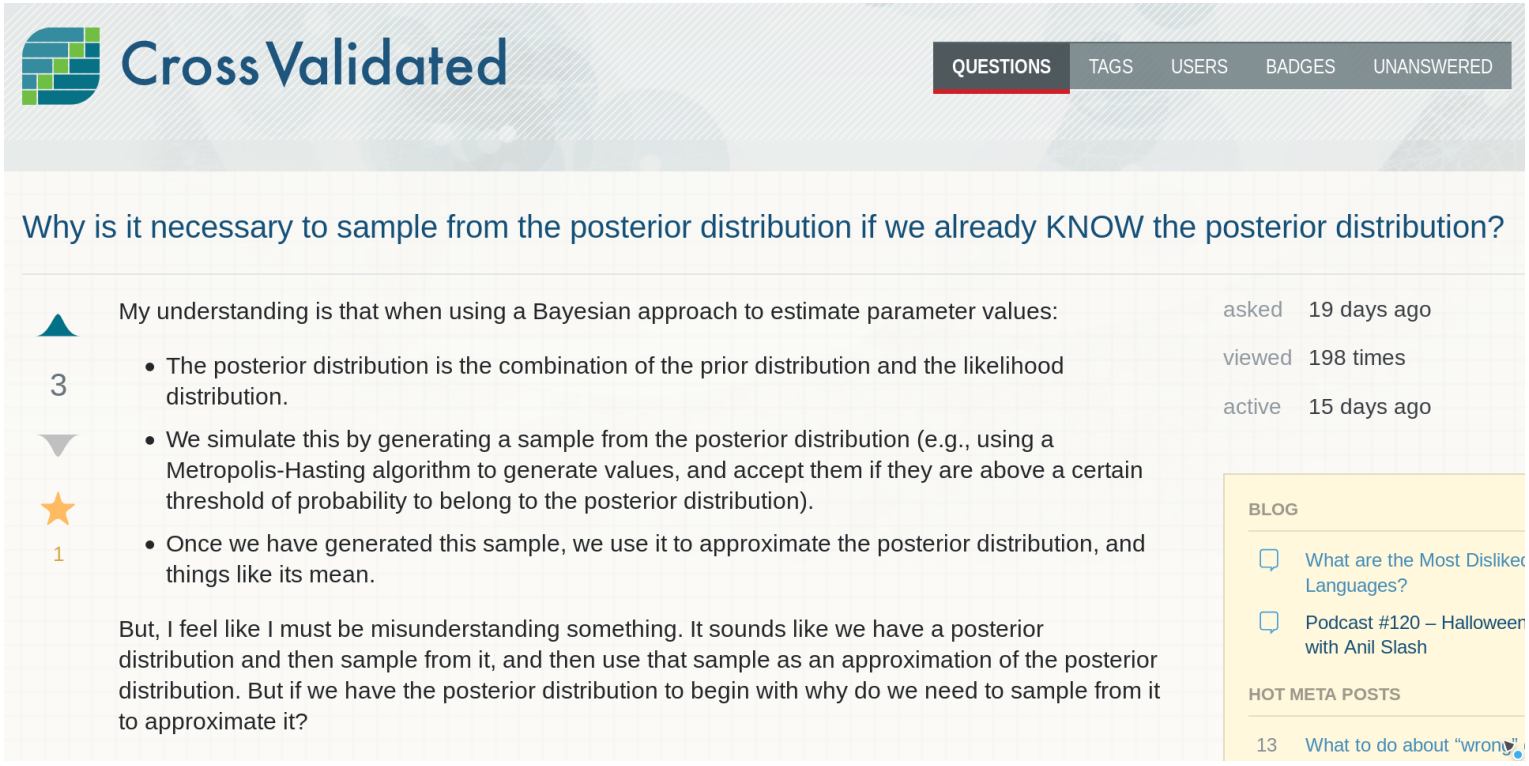
Example of the (default) Mersenne twister with period $2^{19937} - 1$

[Matsumoto & Nishimura, 1997]

*Polynomial must be primitive over the integers mod 2.

Beyond Uniform distributions

Why is it complicated to sample from the posterior distribution if we already KNOW it?



CrossValidated

QUESTIONS TAGS USERS BADGES UNANSWERED

Why is it necessary to sample from the posterior distribution if we already KNOW the posterior distribution?

3

1

My understanding is that when using a Bayesian approach to estimate parameter values:

- The posterior distribution is the combination of the prior distribution and the likelihood distribution.
- We simulate this by generating a sample from the posterior distribution (e.g., using a Metropolis-Hasting algorithm to generate values, and accept them if they are above a certain threshold of probability to belong to the posterior distribution).
- Once we have generated this sample, we use it to approximate the posterior distribution, and things like its mean.

But, I feel like I must be misunderstanding something. It sounds like we have a posterior distribution and then sample from it, and then use that sample as an approximation of the posterior distribution. But if we have the posterior distribution to begin with why do we need to sample from it to approximate it?

asked 19 days ago

viewed 198 times

active 15 days ago

BLOG

What are the Most Disliked Languages?

Podcast #120 – Halloween with Anil Slash

HOT META POSTS

13 What to do about "wrong"

[Cross Validated, Stack Exchange]

Beyond Uniform generators

- Generation of any sequence of random variables can be formally implemented through a uniform generator
  - Distributions with explicit F^{-} (for instance, exponential, and Weibull distributions), use the probability integral transform here
  - Case specific methods rely on unique properties of the distribution (e.g., Normal distribution, Poisson distribution)
  - Generic methods (for instance, accept-reject here and ratio-of-uniform here)
- Simulation of standard distributions solved quite efficiently by many numerical and statistical programming packages.

Beyond Uniform generators

- Generation of any sequence of random variables can be formally implemented through a uniform generator
  - Distributions with explicit F^{-} (for instance, exponential, and Weibull distributions), use the probability integral transform here
  - Case specific methods rely on unique properties of the distribution (e.g., Normal distribution, Poisson distribution)
  - Generic methods (for instance, accept-reject here and ratio-of-uniform here)
- Simulation of standard distributions solved quite efficiently by many numerical and statistical programming packages.

Beyond Uniform generators

- Generation of any sequence of random variables can be formally implemented through a uniform generator
  - Distributions with explicit F^{-} (for instance, exponential, and Weibull distributions), use the probability integral transform here
  - Case specific methods rely on unique properties of the distribution (e.g., Normal distribution, Poisson distribution)
  - Generic methods (for instance, accept-reject here and ratio-of-uniform here)
- Simulation of standard distributions solved quite efficiently by many numerical and statistical programming packages.

Beyond Uniform generators

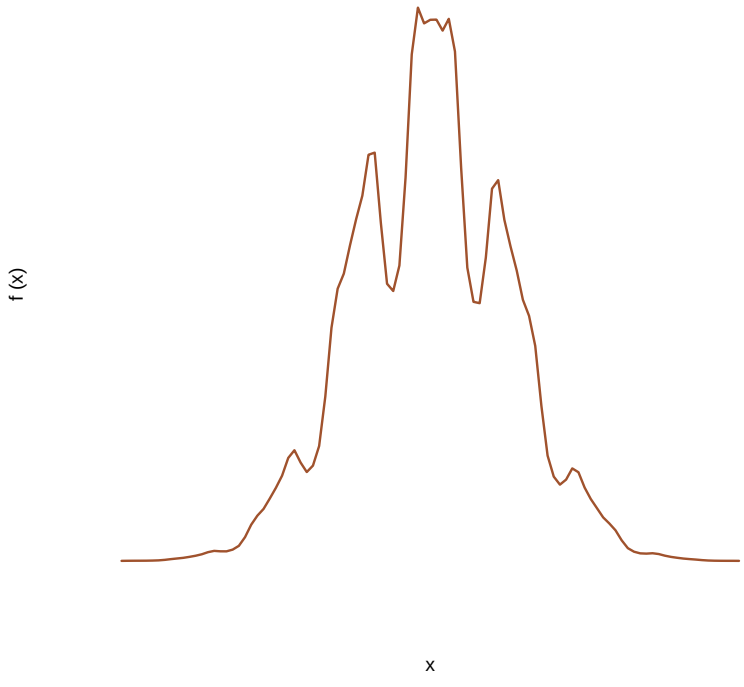
- Generation of any sequence of random variables can be formally implemented through a uniform generator
  - Distributions with explicit F^{-} (for instance, exponential, and Weibull distributions), use the probability integral transform here
  - Case specific methods rely on unique properties of the distribution (e.g., Normal distribution, Poisson distribution)
  - Generic methods (for instance, accept-reject here and ratio-of-uniform here)
- Simulation of standard distributions solved quite efficiently by many numerical and statistical programming packages.

Distributions that differ from uniform distributions

Problem

Given probability distribution with density f, how can we produce randomness according to f?!

- implemented algorithms in a resident software only available for common distributions
- new distributions may require fast resolution
- no approximation allowed



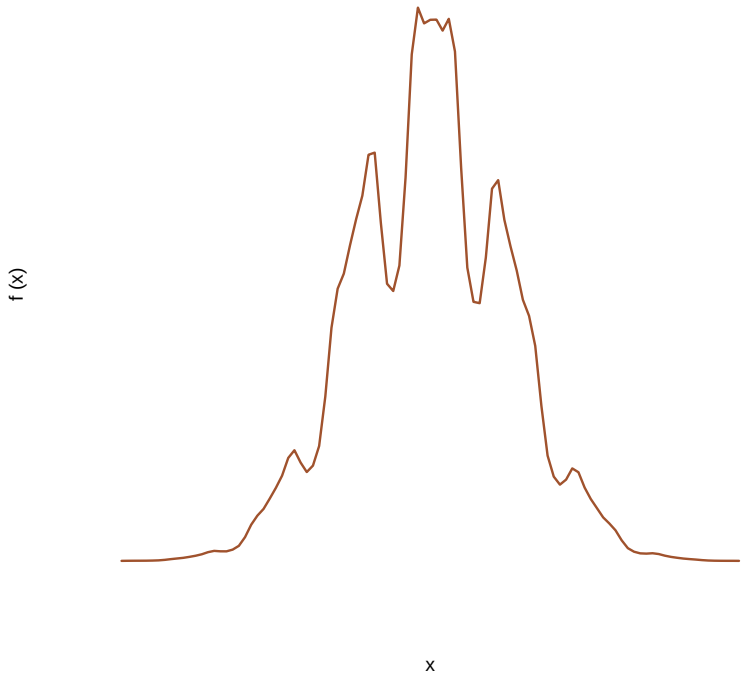
Example of an arbitrary density

Distributions that differ from uniform distributions

Problem

Given probability distribution with density f, how can we produce randomness according to f?!

- implemented algorithms in a resident software only available for common distributions
- new distributions may require fast resolution
- no approximation allowed



Example of an arbitrary density

Simulation 101: The inverse transform method

For a function F on $\mathbb{R}$, the *generalized inverse* of F, F^-, is defined by

$$F^-(u) = \inf\{x;\ F(x) \geq u\}\,.$$

Definition (Probability Integral Transform)

If $U \sim \mathcal{U}_{[0,1]}$, then the random variable $F^-(U)$ is distributed from F.

Simulation 101: The inverse transform method

For a function F on $\mathbb{R}$, the *generalized inverse* of F, F^-, is defined by

$$F^-(u) = \inf\{x;\ F(x) \geq u\}\,.$$

Definition (Probability Integral Transform)

If $U \sim \mathcal{U}_{[0,1]}$, then the random variable $F^-(U)$ is distributed from F.

The inverse transform method (2)

To generate a random variable $X \sim F$, simply generate

$$U \sim \mathcal{U}_{[0,1]}$$

and then make the transform

$$x = F^{-}(u)$$

The inverse transform method (2)

To generate a random variable $X \sim F$, simply generate

$$U \sim \mathcal{U}_{[0,1]}$$

and then make the transform

$$x = F^{-}(u)$$

Transformation methods

Transformation methods

Case where a distribution F is linked in a simple way to another distribution easy to simulate/already available

Example (Exponential variables)

If $U \sim \mathcal{U}_{[0,1]}$, the random variable

$$X = -\log U/\lambda$$

has distribution

$$\begin{aligned} P(X \leq x) &= P(-\log U \leq \lambda x) \\ &= P(U \geq e^{-\lambda x}) = 1 - e^{-\lambda x}, \end{aligned}$$

Exponential distribution $\mathcal{E}xp(\lambda)$.

Further standard distributions

Other random variables that can be generated starting from an exponential include

$$Y = -2 \sum_{j=1}^{\nu} \log(U_j) \sim \chi^2_{2\nu} \qquad \text{(chi-square)}$$

$$Y = -\frac{1}{\beta} \sum_{j=1}^{a} \log(U_j) \sim \mathcal{G}a(a, \beta) \qquad \text{(Gamma)}$$

$$Y = \frac{\sum_{j=1}^{a} \log(U_j)}{\sum_{j=1}^{a+b} \log(U_j)} \sim \mathcal{B}e(a, b) \qquad \text{(Beta)}$$

Points to note

- Transformation must be immediate/free to use
- There are more efficient algorithms for Gamma and Beta random variables
- Cannot generate Gamma random variables with a non-integer shape parameter
- For instance, cannot get a χ^2_1 variable, which would get us a $\mathcal{N}(0, 1)$ variable.

Box-Muller Normal Generator

Example (Normal variables)

If r, θ polar coordinates of (X_1, X_2), then,

$$r^2 = X_1^2 + X_2^2 \sim \chi_2^2 = \mathcal{E}(1/2) \quad \text{and} \quad \theta \sim \mathcal{U}[0, 2\pi]$$

Consequence: If U_1, U_2 iid $\mathcal{U}_{[0,1]}$,

$$\begin{aligned} X_1 &= \sqrt{-2\log(U_1)}\,\cos(2\pi U_2) \\ X_2 &= \sqrt{-2\log(U_1)}\,\sin(2\pi U_2) \end{aligned}$$

iid $\mathcal{N}(0, 1)$.

Box-Muller Normal Generator

Example (Normal variables)

If r, θ polar coordinates of (X_1, X_2), then,

$$r^2 = X_1^2 + X_2^2 \sim \chi_2^2 = \mathcal{E}(1/2) \quad \text{and} \quad \theta \sim \mathcal{U}[0, 2\pi]$$

Consequence: If U_1, U_2 iid $\mathcal{U}_{[0,1]}$,

$$\begin{aligned} X_1 &= \sqrt{-2\log(U_1)}\,\cos(2\pi U_2) \\ X_2 &= \sqrt{-2\log(U_1)}\,\sin(2\pi U_2) \end{aligned}$$

iid $\mathcal{N}(0, 1)$.

Box-Muller Algorithm (2)

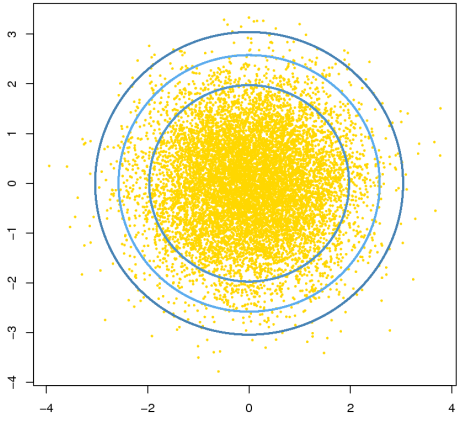
1. Generate U_1, U_2 iid $\mathcal{U}_{[0,1]}$;
2. Define

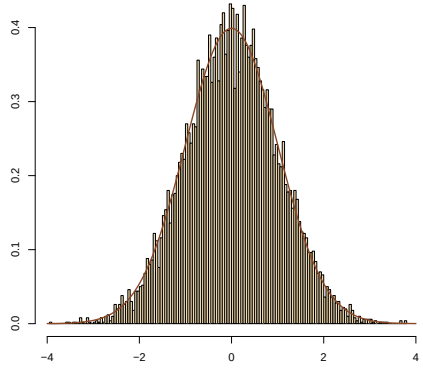
$$\begin{aligned} x_1 &= \sqrt{-2\log(u_1)}\cos(2\pi u_2)\ , \\ x_2 &= \sqrt{-2\log(u_1)}\sin(2\pi u_2)\ ; \end{aligned}$$

3. Take x_1 and x_2 as two independent draws from $\mathcal{N}(0,1)$.

Box-Muller Algorithm (3)

- Unlike algorithms based on the CLT, this algorithm is exact
- Get two normals for the budget of two uniforms
- Drawback (in speed) in calculating $\log$, $\cos$ and $\sin$.





More transforms

▸ Reject

Example (Poisson generation)

Poisson–exponential connection:
If $N \sim \mathcal{P}(\lambda)$ and $X_i \sim \mathcal{E}xp(\lambda)$, $i \in \mathbb{N}^*$,

$$P_\lambda(N = k) = P_\lambda(X_1 + \cdots + X_k \leq 1 < X_1 + \cdots + X_{k+1}) \,.$$

[Poisson process]

More Poisson

Skip Poisson

- A Poisson can be simulated by generating $\mathscr{E}\mathrm{xp}(1)$ till their sum exceeds 1.
- This method is simple, but only practical for smal values of λ as...
- ...on average, the number of exponential variables required is λ.
- Other approaches are more suitable for large λ's.

Atkinson's Poisson (1979)

To generate $N \sim \mathcal{P}(\lambda)$:

1. Define

$$\beta = \pi/\sqrt{3\lambda}, \quad \alpha = \lambda\beta \quad \text{and} \quad k = \log c - \lambda - \log \beta;$$

2. Generate $U_1 \sim \mathcal{U}_{[0,1]}$ and calculate

$$x = \{\alpha - \log\{(1-u_1)/u_1\}\}/\beta$$

   until $x > -0.5$;

3. Define $N = \lfloor x + 0.5 \rfloor$ and generate $U_2 \sim \mathcal{U}_{[0,1]}$;

4. Accept N if

$$\alpha - \beta x + \log\left(u_2/\{1+\exp(\alpha-\beta x)\}^2\right) \leq k + N\log\lambda - \log N!$$

Negative extension

- A generator of Poisson random variables can produce Negative Binomial random variables since,

$$Y \sim \mathcal{G}a(n, (1-p)/p) \quad X|y \sim \mathcal{P}(y)$$

implies

$$X \sim \mathcal{N}eg(n, p)$$

Mixture representation

- The representation of the Negative Binomial is a particular case of a *mixture distribution*
- The principle of a mixture representation is to represent a density f as the marginal of another distribution, for example

$$f(x) = \sum_{i \in \mathscr{Y}} p_i \, f_i(x) \, ,$$

- If the component distributions $f_i(x)$ can be easily generated, X can be obtained by first choosing f_i with probability p_i and then generating an observation from f_i.

Partitioned sampling

Special case of mixture sampling when

$$f_i(x) = f(x)\, \mathbb{I}_{A_i}(x) \Big/ \int_{A_i} f(x)\, dx$$

and

$$p_i = \Pr(X \in A_i)$$

for a partition $(A_i)_i$ and available p_i's

Accept-Reject Methods

Accept-Reject algorithm

- Many distributions from which it is difficult, or even impossible, to **directly** simulate.
- Another class of methods that only require us to know the functional form of the density f of interest **only** up to a multiplicative constant.
- The key to this method is to use a simpler (simulation-wise) density g, the *instrumental density*, from which the simulation from the *target density* f is actually done.

Fundamental theorem of simulation

Lemma

Simulating

$$X \sim f(x)$$

equivalent to simulating

$$(X, U) \sim \mathcal{U}\{(x, u) : 0 < u < f(x)\}$$

The Accept-Reject algorithm

Given a density of interest f, find a density g and a constant M such that

$$f(x) \leq Mg(x)$$

on the support of f.

Accept-Reject Algorithm

1. Generate $X \sim g$, $U \sim \mathcal{U}_{[0,1]}$;
2. Accept $Y = X$ if $U \leq f(X)/Mg(X)$;
3. Return to 1. otherwise.

The Accept-Reject algorithm

Given a density of interest f, find a density g and a constant M such that

$$f(x) \leq Mg(x)$$

on the support of f.

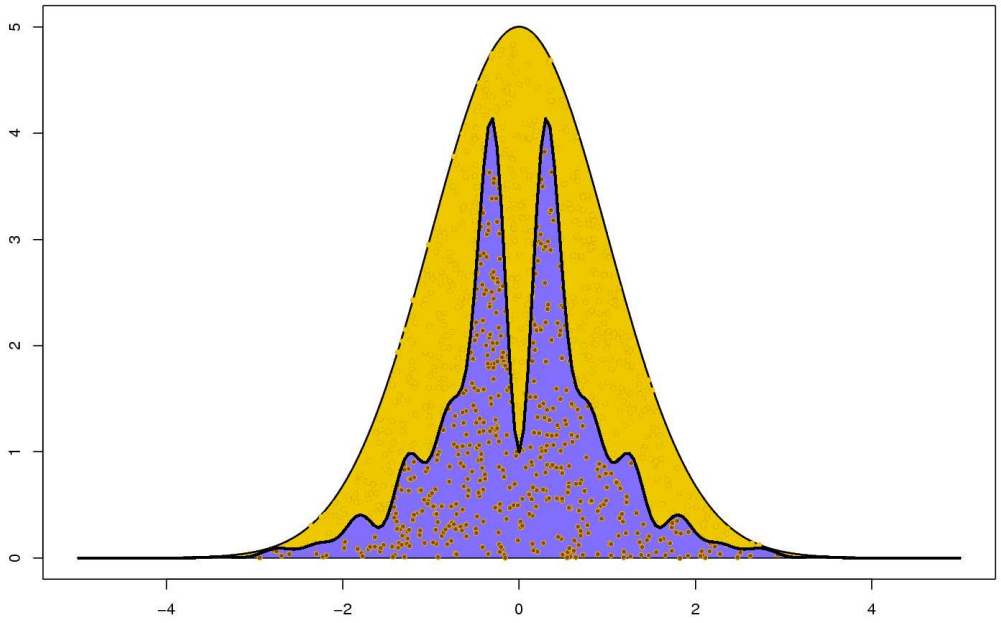
Accept-Reject Algorithm

1. Generate $X \sim g$, $U \sim \mathcal{U}_{[0,1]}$;
2. Accept $Y = X$ if $U \leq f(X)/Mg(X)$;
3. Return to 1. otherwise.

Validation of the Accept-Reject method

Warranty:

This algorithm produces a variable Y distributed according to f



Two interesting properties

- First, it provides a generic method to simulate from any density f that is known *up to a multiplicative factor* Property particularly important in Bayesian calculations where the posterior distribution

$$\pi(\theta|x) \propto \pi(\theta)\, \mathrm{f}(x|\theta)\ .$$

  is specified up to a normalizing constant

- Second, the probability of acceptance in the algorithm is $1/M$, e.g., expected number of trials until a variable is accepted is M (including normalizing constants)

Two interesting properties

- First, it provides a generic method to simulate from any density f that is known *up to a multiplicative factor* Property particularly important in Bayesian calculations where the posterior distribution

$$\pi(\theta|x) \propto \pi(\theta)\, f(x|\theta)\ .$$

  is specified up to a normalizing constant
- Second, the probability of acceptance in the algorithm is $1/M$, e.g., expected number of trials until a variable is accepted is M (including normalizing constants)

More interesting properties

- In cases f and g both probability densities, the constant M is necessarily larger that 1.
- The size of M, and thus the efficiency of the algorithm, are functions of how closely g can imitate f, especially in the tails
- For f/g to remain bounded, necessary for g to have tails thicker than those of f.
  It is e.g. impossible to use the A-R algorithm to simulate a Cauchy distribution f using a Normal distribution g, however the reverse works quite well.

More interesting properties

- In cases f and g both probability densities, the constant M is necessarily larger that 1.
- The size of M, and thus the efficiency of the algorithm, are functions of how closely g can imitate f, especially in the tails
- For f/g to remain bounded, necessary for g to have tails thicker than those of f.
  It is e.g. impossible to use the A-R algorithm to simulate a Cauchy distribution f using a Normal distribution g, however the reverse works quite well.

More interesting properties

- In cases f and g both probability densities, the constant M is necessarily larger that 1.
- The size of M, and thus the efficiency of the algorithm, are functions of how closely g can imitate f, especially in the tails
- For f/g to remain bounded, necessary for g to have tails thicker than those of f.
  It is e.g. impossible to use the A-R algorithm to simulate a Cauchy distribution f using a Normal distribution g, however the reverse works quite well.

Illustration (1)

No Cauchy!

Example (Normal from a Cauchy)

Take

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$$

and

$$g(x) = \frac{1}{\pi} \frac{1}{1+x^2},$$

densities of the Normal and Cauchy distributions.

Then

$$\frac{f(x)}{g(x)} = \sqrt{\frac{\pi}{2}}(1+x^2)\, e^{-x^2/2} \leq \sqrt{\frac{2\pi}{e}} = 1.52$$

attained at $x = \pm 1$.

Illustration (1)

No Cauchy!

Example (Normal from a Cauchy)

Take

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$$

and

$$g(x) = \frac{1}{\pi} \frac{1}{1+x^2},$$

densities of the Normal and Cauchy distributions.
Then

$$\frac{f(x)}{g(x)} = \sqrt{\frac{\pi}{2}}(1+x^2)\, e^{-x^2/2} \leq \sqrt{\frac{2\pi}{e}} = 1.52$$

attained at $x = \pm 1$.

Illustration (1)

Example (Normal from a Cauchy (2))

So probability of acceptance

$$1/1.52 = 0.66,$$

and, on the average, one out of every three simulated Cauchy variables is rejected.

Illustration (2)

No Double!

Example (Normal/Double Exponential)

Generate a $\mathcal{N}(0,1)$ by using a double-exponential distribution with density

$$g(x|\alpha) = (\alpha/2)\exp(-\alpha|x|)$$

Then

$$\frac{f(x)}{g(x|\alpha)} \leq \sqrt{\frac{2}{\pi}}\alpha^{-1}e^{-\alpha^2/2}$$

and minimum of this bound (in α) attained for

$$\alpha^\star = 1$$

Illustration (2)

Example (Normal/Double Exponential (2))

Probability of acceptance

$$\sqrt{\pi/2e} = .76$$

To produce one Normal random variable requires on the average $1/.76 \approx 1.3$ uniform variables.

Illustration (3)

▸ truncate

Example (Gamma generation)

Illustrates a real advantage of the Accept-Reject algorithm
The Gamma distribution $\mathcal{G}a(\alpha, \beta)$ represented as the sum of α exponential random variables, only if α is an integer

Illustration (3)

Example (Gamma generation (2))

Can use the Accept-Reject algorithm with instrumental distribution

$$\mathcal{G}a(a, b), \text{ with } a = [\alpha], \quad \alpha \geq 0.$$

(Without loss of generality, $\beta = 1$.)
Up to a normalizing constant,

$$f/g_b = b^{-a} x^{\alpha-a} \exp\{-(1-b)x\} \leq b^{-a} \left(\frac{\alpha - a}{(1-b)e}\right)^{\alpha-a}$$

for $b \leq 1$.
The maximum is attained at $b = a/\alpha$.

Illustration (3)

Example (Gamma generation (2))

Can use the Accept-Reject algorithm with instrumental distribution

$$\mathcal{G}a(a, b), \text{ with } a = [\alpha], \quad \alpha \geq 0.$$

(Without loss of generality, $\beta = 1$.)
Up to a normalizing constant,

$$f/g_b = b^{-a} x^{\alpha - a} \exp\{-(1-b)x\} \leq b^{-a} \left(\frac{\alpha - a}{(1-b)e}\right)^{\alpha - a}$$

for $b \leq 1$.
The maximum is attained at $b = a/\alpha$.

Cheng and Feast's Gamma generator

Gamma $\mathcal{G}a(\alpha, 1)$, $\alpha > 1$ distribution

1. Define $c_1 = \alpha - 1$, $c_2 = (\alpha - (1/6\alpha))/c_1$, $c_3 = 2/c_1$, $c_4 = 1 + c_3$, and $c_5 = 1/\sqrt{\alpha}$.
2. Repeat
   generate U_1, U_2
   take $U_1 = U_2 + c_5(1 - 1.86U_1)$ if $\alpha > 2.5$
   until $0 < U_1 < 1$.
3. Set $W = c_2 U_2 / U_1$.
4. If $c_3 U_1 + W + W^{-1} \leq c_4$ or $c_3 \log U_1 - \log W + W \leq 1$,
   take $c_1 W$;
   otherwise, repeat.

Truncated Normal simulation

Example (Truncated Normal distributions)

Constraint $x \geq \underline{\mu}$ produces density proportional to

$$e^{-(x-\mu)^2/2\sigma^2}\, \mathbb{I}_{x \geq \underline{\mu}}$$

for a bound $\underline{\mu}$ large compared with μ

There exists alternatives far superior to the naïve method of generating a $\mathcal{N}(\mu, \sigma^2)$ until exceeding $\underline{\mu}$, which requires an average number of

$$1/\Phi((\mu - \underline{\mu})/\sigma)$$

simulations from $\mathcal{N}(\mu, \sigma^2)$ for a single acceptance.

Truncated Normal simulation

Example (Truncated Normal distributions)

Constraint $x \geq \underline{\mu}$ produces density proportional to

$$e^{-(x-\mu)^2/2\sigma^2}\, \mathbb{I}_{x \geq \underline{\mu}}$$

for a bound $\underline{\mu}$ large compared with μ
There exists alternatives far superior to the naïve method of generating a $\mathcal{N}(\mu, \sigma^2)$ until exceeding $\underline{\mu}$, which requires an average number of

$$1/\Phi((\mu - \underline{\mu})/\sigma)$$

simulations from $\mathcal{N}(\mu, \sigma^2)$ for a single acceptance.

Truncated Normal simulation

Example (Truncated Normal distributions (2))

Instrumental distribution: translated exponential distribution, $\mathcal{E}(\alpha, \underline{\mu})$, with density

$$g_\alpha(z) = \alpha e^{-\alpha(z-\underline{\mu})}\, \mathbb{I}_{z \geq \underline{\mu}}\,.$$

The ratio f/g_α is bounded by

$$f/g_\alpha \leq \begin{cases} 1/\alpha\, \exp(\alpha^2/2 - \alpha\underline{\mu}) & \text{if } \alpha > \underline{\mu}\,, \\ 1/\alpha\, \exp(-\underline{\mu}^2/2) & \text{otherwise.} \end{cases}$$

Truncated Normal simulation

Example (Truncated Normal distributions (2))

Instrumental distribution: translated exponential distribution, $\mathscr{E}(\alpha, \underline{\mu})$, with density

$$g_\alpha(z) = \alpha e^{-\alpha(z-\underline{\mu})}\, \mathbb{I}_{z \geq \underline{\mu}}\,.$$

The ratio f/g_α is bounded by

$$f/g_\alpha \leq \begin{cases} 1/\alpha\ \exp(\alpha^2/2 - \alpha\underline{\mu}) & \text{if } \alpha > \underline{\mu}\,, \\ 1/\alpha\ \exp(-\underline{\mu}^2/2) & \text{otherwise.} \end{cases}$$

Interlude #3: Log-concave densities

Interlude #3: Log-concave densities

Densities f whose logarithm is concave, for instance Bayesian posterior distributions such that

$$\log\ \pi(\theta|x) = \log\ \pi(\theta) + \log\ f(x|\theta) + c$$

concave

Interlude #3: Log-concave densities

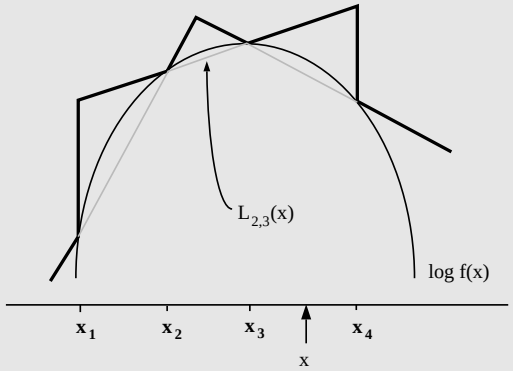
Take

$$\mathfrak{S}_n = \{x_i, i = 0, 1, \ldots, n+1\} \subset \text{supp}(f)$$

such that $h(x_i) = \log f(x_i)$ known up to the same constant.

By concavity of h, line $L_{i,i+1}$ through $(x_i, h(x_i))$ and $(x_{i+1}, h(x_{i+1}))$

- below h in $[x_i, x_{i+1}]$ and
- above this graph outside this interval



Interlude #3: Log-concave densities

For $x \in [x_i, x_{i+1}]$, if

$$\overline{h}_n(x) = \min\{L_{i-1,i}(x), L_{i+1,i+2}(x)\} \quad \text{and} \quad \underline{h}_n(x) = L_{i,i+1}(x)\,,$$

the envelopes are

$$\underline{h}_n(x) \leq h(x) \leq \overline{h}_n(x)$$

uniformly on the support of f, with

$$\underline{h}_n(x) = -\infty \quad \text{and} \quad \overline{h}_n(x) = \min(L_{0,1}(x), L_{n,n+1}(x))$$

on $[x_0, x_{n+1}]^c$.

Interlude #3: Log-concave densities

Therefore, if

$$\underline{f}_n(x) = \exp \underline{h}_n(x) \text{ and } \overline{f}_n(x) = \exp \overline{h}_n(x)$$

then

$$\underline{f}_n(x) \leq f(x) \leq \overline{f}_n(x) = \varpi_n \, g_n(x) \, ,$$

where ϖ_n normalizing constant of f_n

Interlude #3: ARS Algorithm

1. Initialize n and $\mathfrak{S}_n$.
2. Generate $X \sim g_n(x)$, $U \sim \mathcal{U}_{[0,1]}$.
3. If $U \leq \underline{f}_n(X)/\varpi_n \; g_n(X)$, accept X;
   otherwise, if $U \leq f(X)/\varpi_n \; g_n(X)$, accept X

Interlude #3: Log-concave densities

▸ kill ducks

Example (Northern Pintail ducks)

Ducks captured at time i with both probability p_i and size N of the population unknown.
Dataset

$$(n_1, \ldots, n_{11}) = (32, 20, 8, 5, 1, 2, 0, 2, 1, 1, 0)$$

Number of recoveries over the years 1957–1968 of 1612 Northern Pintail ducks banded in 1956

Interlude #3: Log-concave densities

Example (Northern Pintail ducks (2))

Corresponding conditional likelihood

$$L(n_1, \ldots, n_I | N, p_1, \ldots, p_I) \propto \prod_{i=1}^{I} p_i^{n_i} (1 - p_i)^{N - n_i},$$

where I number of captures, n_i number of captured animals during the ith capture, and r is the total number of different captured animals.

Interlude #3: Log-concave densities

Example (Northern Pintail ducks (3))

Prior selection

If

$$N \sim \mathscr{P}(\lambda)$$

and

$$\alpha_i = \log\left(\frac{p_i}{1-p_i}\right) \sim \mathcal{N}(\mu_i, \sigma^2),$$

[Normal logistic]

Interlude #3: Log-concave densities

Example (Northern Pintail ducks (4))

Posterior distribution

$$\pi(\alpha, N|, n_1, \ldots, n_I) \quad \propto \quad \frac{N!}{(N-r)!}\frac{\lambda^N}{N!}\prod_{i=1}^{I}(1+e^{\alpha_i})^{-N}$$
$$\prod_{i=1}^{I}\exp\left\{\alpha_i n_i - \frac{1}{2\sigma^2}(\alpha_i-\mu_i)^2\right\}$$

Interlude #3: Log-concave densities

Example (Northern Pintail ducks (5))

For the conditional posterior distribution

$$\pi(\alpha_i|N, n_1, \ldots, n_I) \propto \exp\left\{\alpha_i n_i - \frac{1}{2\sigma^2}(\alpha_i - \mu_i)^2\right\} \Big/ (1+e^{\alpha_i})^N ,$$

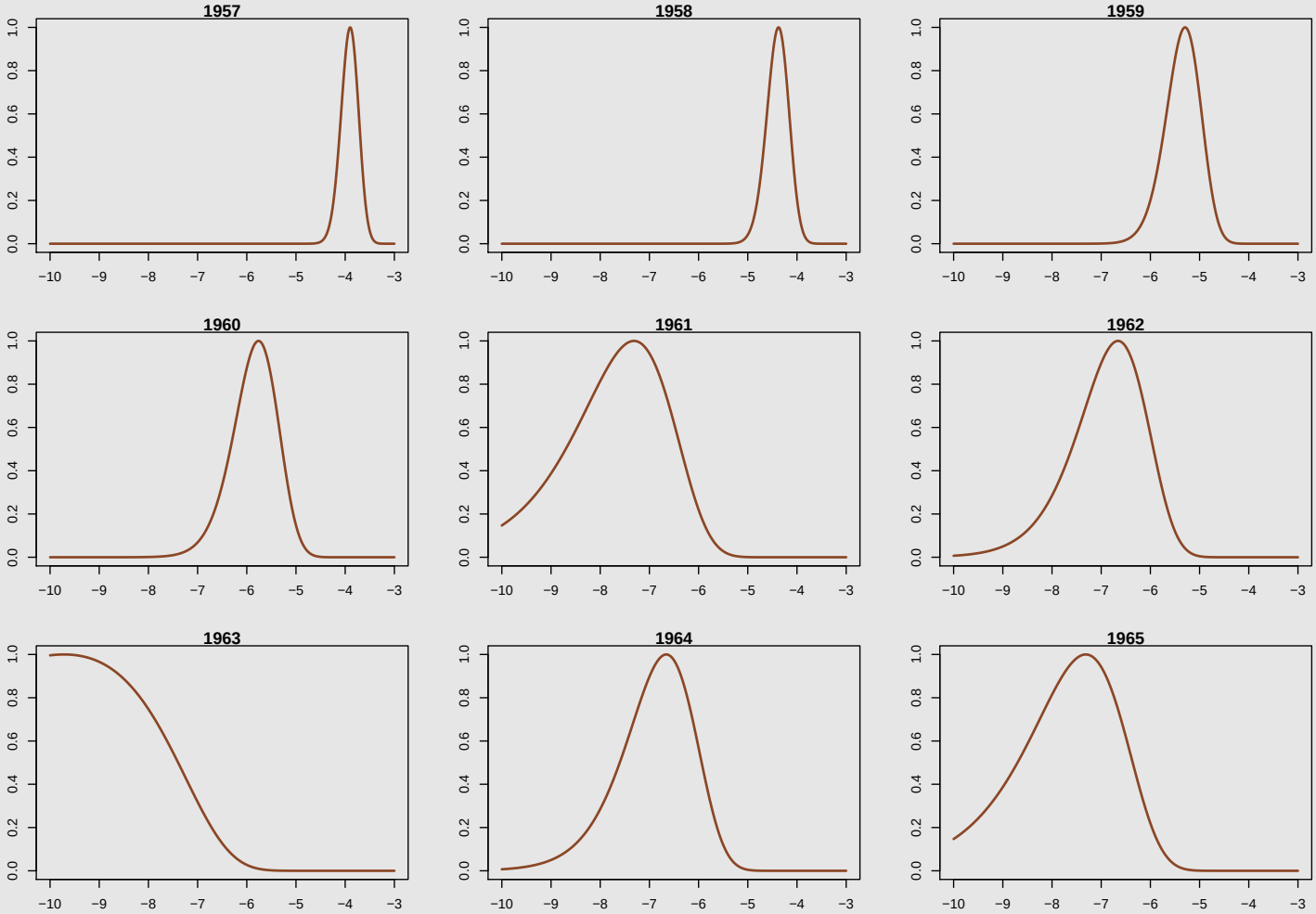
the ARS algorithm can be implemented since

$$\alpha_i n_i - \frac{1}{2\sigma^2}(\alpha_i - \mu_i)^2 - N \log(1 + e^{\alpha_i})$$

is concave in α_i.

Interlude #3: Log-concave densities

Posterior distributions of capture log-odds ratios for the years 1957–1965.



Interlude #3: Log-concave densities

True distribution versus histogram of simulated sample

Slice sampler

There exist other ways of exploiting the fundamental lemma (of simulation over the density subgraph)

[Damien & al., 1999; Neal, 2003]

Slice sampler

There exist other ways of exploiting the fundamental lemma (of simulation over the density subgraph)

If direct uniform simulation on

$$\mathcal{S}_f = \{(u, x);\, 0 \le u \le f(x)\}$$

is too complex *[because of unavailable hat/instrumental distribution]* use instead a **random walk** on $\mathcal{S}_f$

[Damien & al., 1999; Neal, 2003]

Slice sampler

There exist other ways of exploiting the fundamental lemma (of simulation over the density subgraph)

can be achieved by making random jumps in vertical then horizontal directions, accounting for the boundaries

- $0 \leq u \leq f(x)$, i.e. $\mathcal{U}(0,1)$
- $f(x) \geq u$, i.e. $x \sim \mathscr{U}_{\mathcal{S}(u)}$

Justification by Markov chain theory: **ergodic chain with Uniform stationary and limiting distribution**

[Damien & al., 1999; Neal, 2003]

Slice sampler

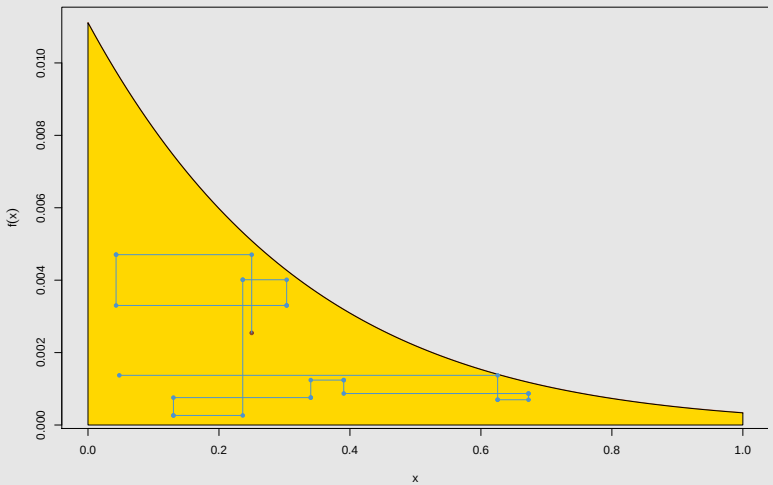
Slice sampler algorithm

For $t = 1, \ldots, T$

when at $(x^{(t)}, \omega^{(t)})$ simulate

1. $\omega^{(t+1)} \sim \mathcal{U}_{[0, f(x^{(t)})]}$
2. $x^{(t+1)} \sim \mathcal{U}_{\mathcal{S}^{(t+1)}}$, where

$$\mathcal{S}^{(t+1)} = \{y;\ f(y) \geq \omega^{(t+1)}\}.$$



Ratio of Uniforms

Ratio of uniforms principle

Consider the set A of (u, v)'s in $\mathbb{R}^+ \times \mathcal{X}$ such that

$$0 \leq u^2 \leq f(v/u)$$

Then a uniform distribution on A induces the distribution with density proportional to f on V/U.

[Kinderman and Monahan's (1977)]

Ratio of uniforms validation

Consider the change of variables from (u, v) to $(u, w = v/u)$ with Jacobian u, then (u, w) has the density

$$u \, \mathbb{I}_{(0, f(w)^{1/2})}(u)$$

Integrating out u leads to

$$\int_0^{f(w)^{1/2}} u \, du = f(w)^{1/2 \times 2} = f(w)$$

as proportional to the density of V/U

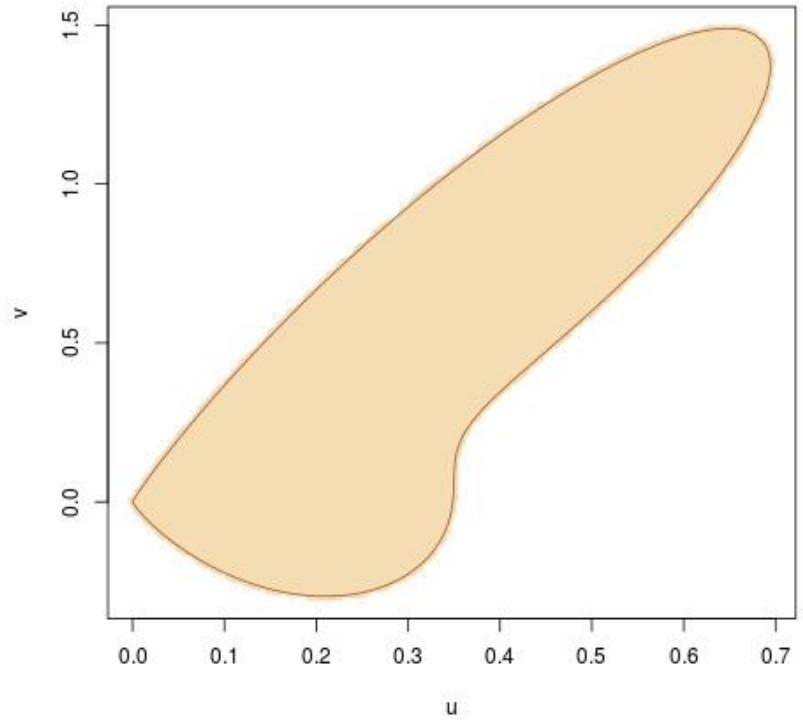
Ratio of uniforms implementation

Simulating a uniform distribution on $\mathcal{A}$ means identifying the region within a simple box $\mathfrak{B}$

Ratio of uniforms implementation

Simulating a uniform distribution on $\mathcal{A}$ means identifying the region within a simple box $\mathfrak{B}$
Boundaries of $\mathcal{A}$ given by

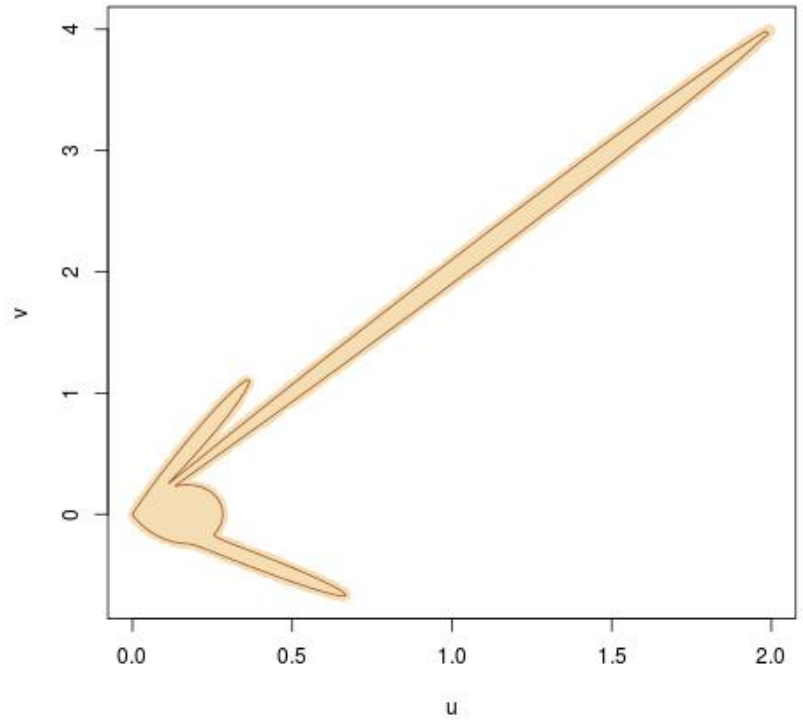
$$\mathcal{A}^b = \{(u(x) = f(x)^{1/2}, v(x) = xf(x)^{1/2});\ x \in \mathcal{X}\}$$



Ratio of uniforms implementation

Simulating a uniform distribution on $\mathcal{A}$ means identifying the region within a simple box $\mathfrak{B}$
Boundaries of $\mathcal{A}$ given by

$$\mathcal{A}^b = \{(u(x) = f(x)^{1/2}, v(x) = xf(x)^{1/2});\ x \in \mathcal{X}\}$$



Ratio of uniforms implementation

There exists a compact box $\mathfrak{B}$ containing $\mathfrak{A}$ iff

$$0 \leq f(x) \leq \bar{f} \qquad 0 \leq xf(x)^{1/2} \leq \tilde{f}$$

Applications to standard distributions like Student's t

[Devroye, 1986, Section 3.7]

Example: Chen and Feast (1979) gamma generator (R `rgamma`) is a ratio of uniforms algorithm

Ratio of uniforms implementation

There exists a compact box $\mathfrak{B}$ containing $\mathfrak{A}$ iff

$$0 \le f(x) \le \bar{f} \qquad 0 \le x f(x)^{1/2} \le \tilde{f}$$

Applications to standard distributions like Student's t

[Devroye, 1986, Section 3.7]

Example: Chen and Feast (1979) gamma generator (R `rgamma`) is a ratio of uniforms algorithm

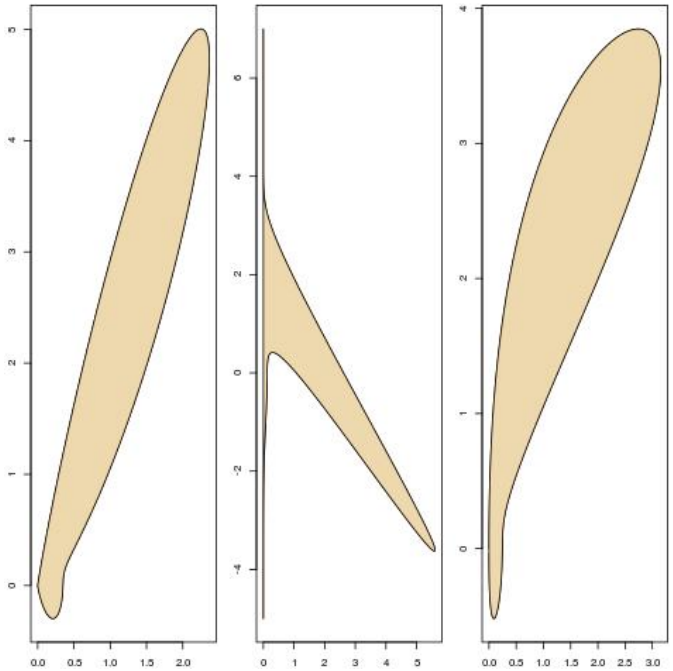
Ratio of uniforms generalisation

Principle that can be generalised to a monotone transform of f, $h(f)$, and the set

$$\mathfrak{H} = \{(u, v);\ 0 \leq u \leq h(f(v/g(u)))\}$$

which still produces a distribution with density proportional to f when

$$g(x) = {}^{dG}/_{dx}(x) \qquad G(x) = h^{-1}(x)$$



Ratio of uniforms generalisation

- choice of transform f most adequate for a given f
- slice sampler deduced from this construct
- case of an unbounded density f

Ratio of uniforms generalisation (2)

In dimension d, when generating a Uniform random variable over

$$C(r) = \left\{ (u, v_1, \ldots, v_d) : 0 < u \leq \left[f\left(\frac{v_1}{u^r}, \ldots, \frac{v_d}{u^r} \right) \right]^{1/(rd+1)} \right\} \quad r > 0$$

then

$$(v_1/u^r, \ldots, v_d/u^r) \sim f(x) / \int f(y(\, dy$$

[Wakefield & al., 1991; Northop & al., 2016]

Example: multivariate Normal distribution

Standard d-dimensional Normal distribution

$$f(x) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{d} x_i^2\right)$$

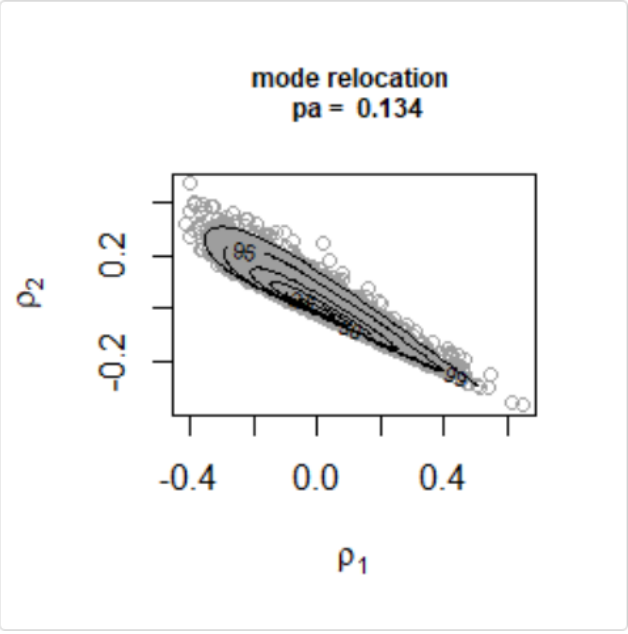
maximal probability of acceptance occurs when $r = 1/2$

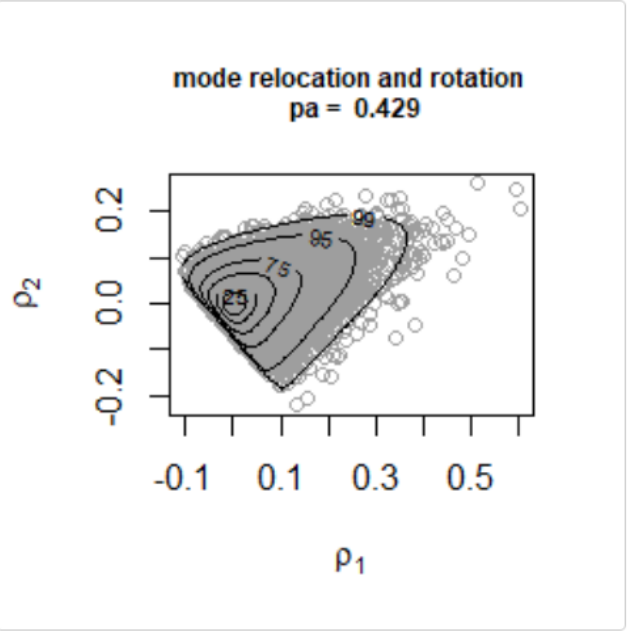
$$p_a(d, 1/2) = \frac{(\pi e)^{d/2}}{2^d (1 + d/2)^{1+d/2}}$$

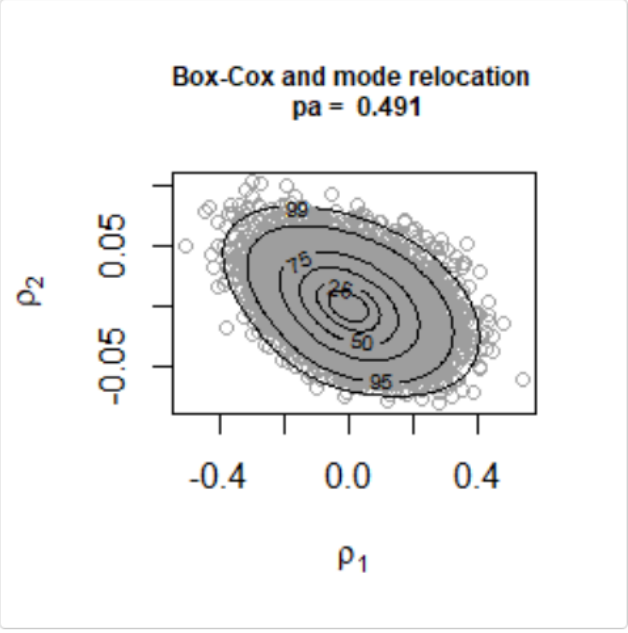
which quickly decreases in d

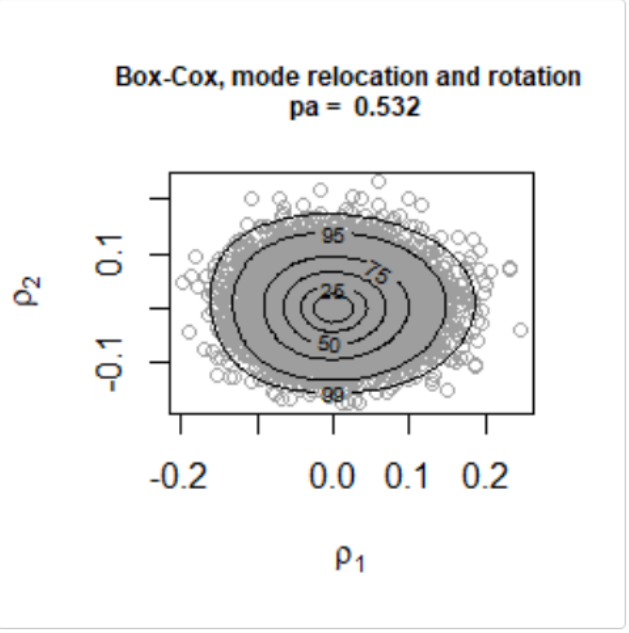
[rust R package, P. Northop, 2024]

Ratio of uniforms generalisation (2)









Monte Carlo integration

Quick reminder

Two major classes of numerical problems that arise in statistical inference

- **Optimization** - generally associated with the likelihood approach
- **Integration**- generally associated with the Bayesian approach

Quick reminder

Two major classes of numerical problems that arise in statistical inference

- **Optimization** - generally associated with the likelihood approach
- **Integration**- generally associated with the Bayesian approach

Illustration

skip Example!

Example (Bayesian decision theory)

Bayes estimators are not always posterior expectations, but rather solutions of the minimization problem

$$\min_{\delta} \int_{\Theta} \mathrm{L}(\theta, \delta)\, \pi(\theta)\, \mathrm{f}(\mathrm{x}|\theta)\, \mathrm{d}\theta\ .$$

Proper loss:
For $\mathrm{L}(\theta, \delta) = (\theta - \delta)^2$, the Bayes estimator is the **posterior mean**

Absolute error loss:
For $\mathrm{L}(\theta, \delta) = |\theta - \delta|$, the Bayes estimator is the **posterior median**

With no loss function
use the maximum a posteriori (MAP) estimator

$$\arg\max_{\theta} \ell(\theta|\mathrm{x})\pi(\theta)$$

Illustration

skip Example!

Example (Bayesian decision theory)

Bayes estimators are not always posterior expectations, but rather solutions of the minimization problem

$$\min_{\delta} \int_{\Theta} \mathrm{L}(\theta, \delta)\, \pi(\theta)\, \mathrm{f}(\mathrm{x}|\theta)\, \mathrm{d}\theta\ .$$

Proper loss:
For $\mathrm{L}(\theta, \delta) = (\theta - \delta)^2$, the Bayes estimator is the **posterior mean**

Absolute error loss:
For $\mathrm{L}(\theta, \delta) = |\theta - \delta|$, the Bayes estimator is the **posterior median**

With no loss function
use the maximum a posteriori (MAP) estimator

$$\arg\max_{\theta} \ell(\theta|\mathrm{x})\pi(\theta)$$

Illustration

skip Example!

Example (Bayesian decision theory)

Bayes estimators are not always posterior expectations, but rather solutions of the minimization problem

$$\min_{\delta} \int_{\Theta} \mathrm{L}(\theta, \delta)\, \pi(\theta)\, \mathrm{f}(\mathrm{x}|\theta)\, \mathrm{d}\theta\ .$$

Proper loss:
For $\mathrm{L}(\theta, \delta) = (\theta - \delta)^2$, the Bayes estimator is the **posterior mean**

Absolute error loss:
For $\mathrm{L}(\theta, \delta) = |\theta - \delta|$, the Bayes estimator is the **posterior median**

With no loss function
use the maximum a posteriori (MAP) estimator

$$\arg\max_{\theta} \ell(\theta|\mathrm{x})\pi(\theta)$$

Monte Carlo integration

Monte Carlo integration

Theme:

Generic problem of evaluating the integral

$$\mathfrak{I} = \mathbb{E}_{\mathrm{f}}[\mathrm{h}(\mathrm{X})] = \int_{\mathscr{X}} \mathrm{h}(\mathrm{x})\ \mathrm{f}(\mathrm{x})\ \mathrm{dx}$$

where $\mathscr{X}$ is uni- or multidimensional, f is a closed form, partly closed form, or implicit density, and h is a function

Monte Carlo integration (2)

Monte Carlo solution

First use a sample $(X_1, \ldots, X_m)$ from the density f to approximate the integral $\mathfrak{I}$ by the empirical average

$$\overline{h}_m = \frac{1}{m} \sum_{j=1}^{m} h(x_j)$$

which converges

$$\overline{h}_m \longrightarrow \mathbb{E}_f[h(X)]$$

by the **Strong Law of Large Numbers**

Monte Carlo integration (2)

Monte Carlo solution

First use a sample $(X_1, \ldots, X_m)$ from the density f to approximate the integral $\mathfrak{I}$ by the empirical average

$$\overline{h}_m = \frac{1}{m} \sum_{j=1}^{m} h(x_j)$$

which converges

$$\overline{h}_m \longrightarrow \mathbb{E}_f[h(X)]$$

by the **Strong Law of Large Numbers**

Monte Carlo precision

Estimate the variance with

$$v_m = \frac{1}{m-1} \sum_{j=1}^{m} [h(x_j) - \overline{h}_m]^2,$$

and for m large,

$$\frac{\overline{h}_m - \mathbb{E}_f[h(X)]}{\sqrt{v_m}} \sim \mathcal{N}(0,1).$$

Note: This can lead to the construction of a convergence test and of confidence bounds on the approximation of $\mathbb{E}_f[h(X)]$.

Illustration

Example (Cauchy prior/normal sample)

For estimating a normal mean, a *robust* prior is a Cauchy prior

$$X \sim \mathcal{N}(\theta, 1), \quad \theta \sim \mathcal{C}(0, 1).$$

Under squared error loss, posterior mean

$$\delta^{\pi}(x) = \frac{\displaystyle\int_{-\infty}^{\infty} \frac{\theta}{1+\theta^2} e^{-(x-\theta)^2/2} d\theta}{\displaystyle\int_{-\infty}^{\infty} \frac{1}{1+\theta^2} e^{-(x-\theta)^2/2} d\theta}$$

Illustration

Example (Cauchy prior/normal sample (2))

Form of δ^π suggests simulating iid variables

$$\theta_1, \cdots, \theta_m \sim \mathcal{N}(x, 1)$$

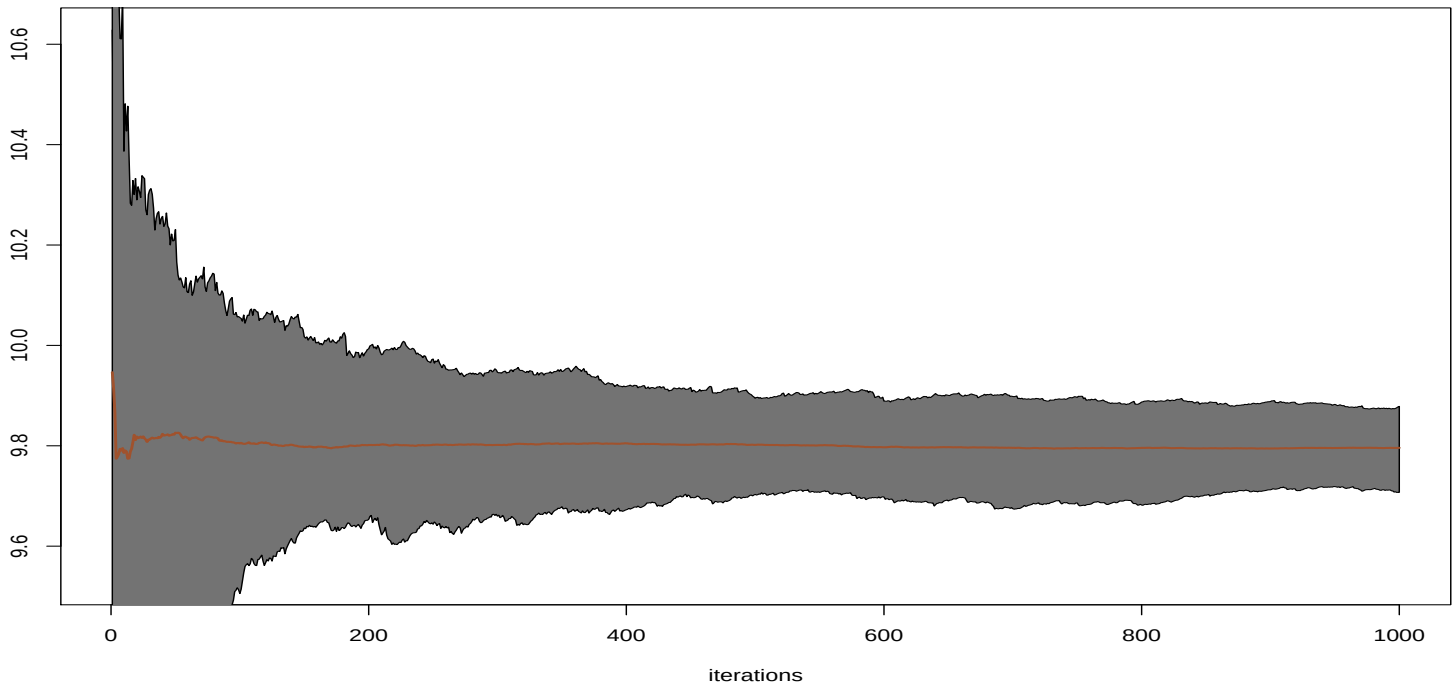
and calculating

$$\hat{\delta}^\pi_m(x) = \sum_{i=1}^m \frac{\theta_i}{1+\theta_i^2} \Big/ \sum_{i=1}^m \frac{1}{1+\theta_i^2} \, .$$

The Law of Large Numbers implies

$$\hat{\delta}^\pi_m(x) \longrightarrow \delta^\pi(x) \text{ as } m \longrightarrow \infty.$$

Illustration



Range of estimators δ_m^π for 100 runs and $x = 10$

Importance Sampling

Importance sampling

Paradox

Simulation from f (the true density) is not necessarily **optimal**

Alternative to direct sampling from f is **importance sampling**, based on the alternative representation

$$\mathbb{E}_f[h(X)] = \int_{\mathcal{X}} \left[h(x) \, \frac{f(x)}{g(x)} \right] \, g(x) \, dx \, .$$

which allows us to use **other** distributions than f

Importance sampling

Paradox

Simulation from f (the true density) is not necessarily **optimal**

Alternative to direct sampling from f is **importance sampling**, based on the alternative representation

$$\mathbb{E}_f[h(X)] = \int_{\mathcal{X}} \left[h(x) \, \frac{f(x)}{g(x)} \right] \, g(x) \, dx \, .$$

which allows us to use **other** distributions than f

Importance sampling algorithm

Evaluation of

$$\mathbb{E}_f[h(X)] = \int_{\mathscr{X}} h(x)\, f(x)\, dx$$

by

1. Generate a sample $X_1, \ldots, X_n$ from a distribution g
2. Use the approximation

$$\frac{1}{m} \sum_{j=1}^{m} \frac{f(X_j)}{g(X_j)}\, h(X_j)$$

Same thing as before!!!

Convergence of the estimator

$$\frac{1}{m} \sum_{j=1}^{m} \frac{f(X_j)}{g(X_j)} h(X_j) \longrightarrow \int_{\mathscr{X}} h(x)\, f(x)\, dx$$

converges for any choice of the distribution g
[as long as $\mathrm{supp}(g) \supset \mathrm{supp}(f)$**]**

Same thing as before!!!

Convergence of the estimator

$$\frac{1}{m}\sum_{j=1}^{m}\frac{f(X_j)}{g(X_j)}\,h(X_j) \longrightarrow \int_{\mathscr{X}} h(x)\,f(x)\,dx$$

converges for any choice of the distribution g
[as long as $\mathrm{supp}(g) \supset \mathrm{supp}(f)$**]**

Important details

- Instrumental distribution g chosen from distributions easy to simulate
- The same sample (generated from g) can be used repeatedly, not only for different functions h, but also for different densities f
- Even dependent proposals can be used, as seen later

▸ PMC chapter

Important choice

Although g can be any density, some choices are better than others:

- Finite variance only when

$$\mathbb{E}_f\left[h^2(X)\frac{f(X)}{g(X)}\right] = \int_{\mathcal{X}} h^2(x)\,\frac{f^2(X)}{g(X)}\,dx < \infty\,.$$

- Instrumental distributions with tails lighter than those of f (that is, with $\sup f/g = \infty$) not appropriate.
- If $\sup f/g = \infty$, the weights $f(x_j)/g(x_j)$ vary widely, giving too much importance to a few values x_j.
- If $\sup f/g = M < \infty$, the accept-reject algorithm can be used as well to simulate f directly.

Important choice

Although g can be any density, some choices are better than others:

- Finite variance only when

$$\mathbb{E}_f\left[h^2(X)\frac{f(X)}{g(X)}\right] = \int_{\mathcal{X}} h^2(x)\,\frac{f^2(X)}{g(X)}\,dx < \infty\,.$$

- Instrumental distributions with tails lighter than those of f (that is, with $\sup f/g = \infty$) not appropriate.
- If $\sup f/g = \infty$, the weights $f(x_j)/g(x_j)$ vary widely, giving too much importance to a few values x_j.
- If $\sup f/g = M < \infty$, the accept-reject algorithm can be used as well to simulate f directly.

Important choice

Although g can be any density, some choices are better than others:

- Finite variance only when

$$\mathbb{E}_f\left[h^2(X)\frac{f(X)}{g(X)}\right] = \int_{\mathcal{X}} h^2(x)\,\frac{f^2(X)}{g(X)}\,dx < \infty\,.$$

- Instrumental distributions with tails lighter than those of f (that is, with $\sup f/g = \infty$) not appropriate.
- If $\sup f/g = \infty$, the weights $f(x_j)/g(x_j)$ vary widely, giving too much importance to a few values x_j.
- If $\sup f/g = M < \infty$, the accept-reject algorithm can be used as well to simulate f directly.

Illustration

Example (Cauchy target)

Case of Cauchy distribution $C(0,1)$ when importance function is Gaussian $\mathcal{N}(0,1)$.
Ratio of the densities

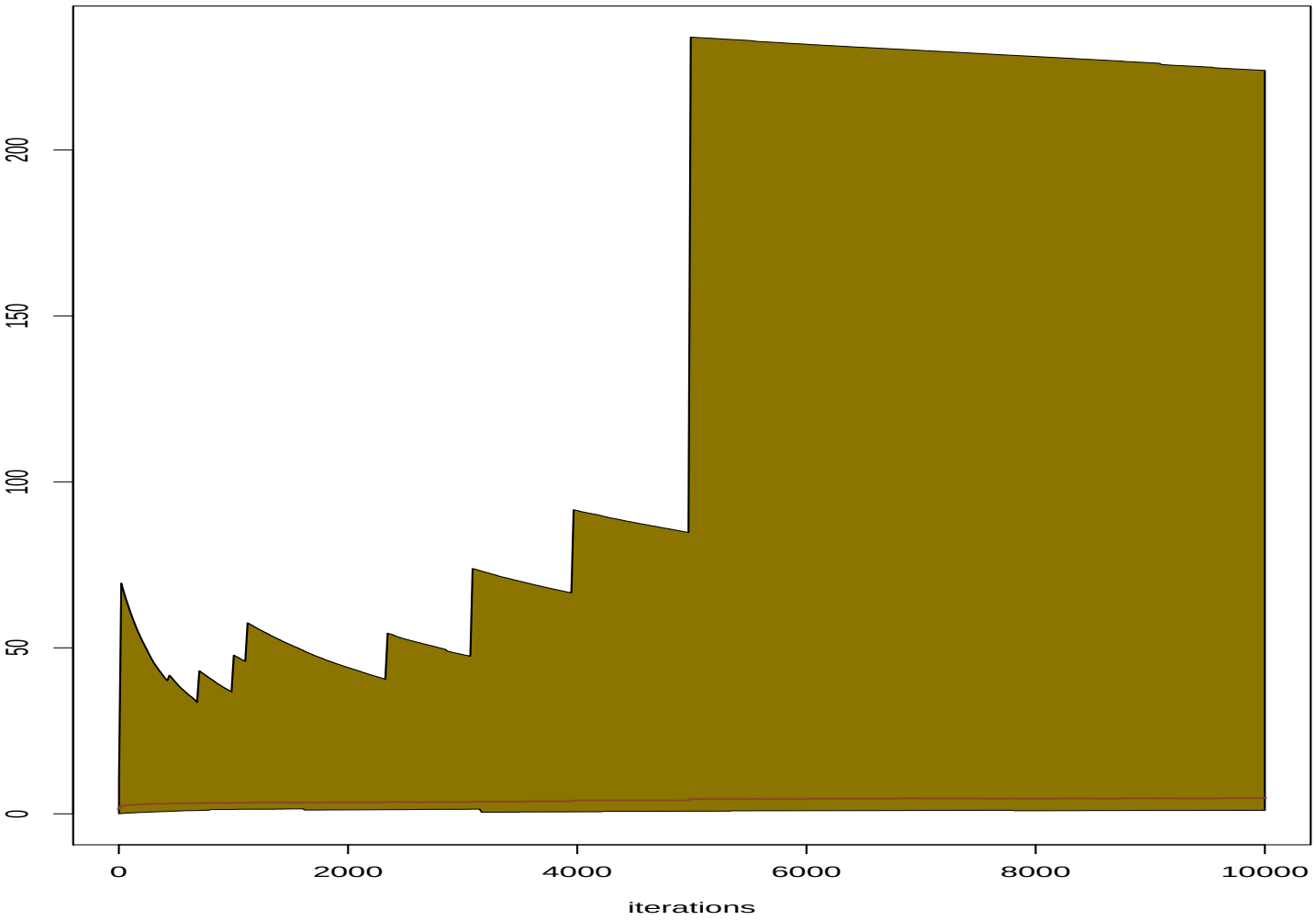
$$\rho(x) = \frac{p^\star(x)}{p_0(x)} = \sqrt{2\pi}\,\frac{\exp x^2/2}{\pi\,(1+x^2)}$$

very badly behaved: e.g.,

$$\int_{-\infty}^{\infty} \rho(x)^2 p_0(x)\,dx = \infty\,.$$

Poor performances of the associated importance sampling estimator

Illustration



Range and average of 500 replications of IS estimate of $\mathbb{E}[\exp -X]$ over $10,000$ iterations.

Interlude #4: Harmonic mean estimator

Interlude #4: Harmonic mean estimator

Estimating

$$\mathfrak{Z}(x) = \int \pi(\theta) L(\theta) L(\theta|x) \, d\theta$$

via [harmonic mean] identity

$$\mathbb{E}^{\pi}\left[\frac{\varphi(\theta)}{\pi_\theta) L(\theta|x)} \middle| x \right] = \int \frac{\varphi(\theta)}{\pi(\theta) L(\theta|x)} \frac{\overbrace{\pi(\theta) L(\theta|x)}^{\pi(\theta|x)}}{\mathfrak{Z}(x)} \, d\theta = \frac{1}{\mathfrak{Z}(x)}$$

no matter what the proposal $\varphi(\cdot)$ is.

[Gelfand & Dey, 1994; Bartolucci et al., 2006]

Direct exploitation of the posterior simulation output

Interlude #4: Harmonic mean estimator

Estimating

$$\mathfrak{Z}(x) = \int \pi(\theta) L(\theta) L(\theta|x) \, d\theta$$

via [harmonic mean] identity

$$\mathbb{E}^{\pi}\left[\frac{\varphi(\theta)}{\pi_\theta) L(\theta|x)} \middle| x \right] = \int \frac{\varphi(\theta)}{\pi(\theta) L(\theta|x)} \frac{\overbrace{\pi(\theta) L(\theta|x)}^{\pi(\theta|x)}}{\mathfrak{Z}(x)} \, d\theta = \frac{1}{\mathfrak{Z}(x)}$$

no matter what the proposal $\varphi(\cdot)$ is.

[Gelfand & Dey, 1994; Bartolucci et al., 2006]

Direct exploitation of the posterior simulation output

Interlude #4: Harmonic mean estimator

Original version with $\varphi(\cdot) = \pi(\cdot)$

$$\widehat{\mathfrak{Z}(x)} = 1 \Big/ \frac{1}{T} \sum_{t=1}^{T} \frac{\cancel{\varphi(\theta^{(t)})}}{\cancel{\pi(\theta^{(t)})} L(\theta^{(t)}|x)} \qquad \theta^{(t)} \sim \pi(\theta|x)$$

[Newton & Raftery, 1994]

"The bad news is that the number of points required for this estimator to get close to the right answer will often be greater than the number of atoms in the observable universe. The even worse news is that it's easy for people to not realize this, and to naïvely accept estimates that are nowhere close to the correct value of the marginal likelihood." R. Neal's blog, 17/08/2008

Interlude #4: Harmonic mean estimator

Original version with $\varphi(\cdot) = \pi(\cdot)$

$$\widehat{\mathfrak{Z}(x)} = 1 \Big/ \frac{1}{T} \sum_{t=1}^{T} \frac{\cancel{\varphi(\theta^{(t)})}}{\cancel{\pi(\theta^{(t)})} L(\theta^{(t)}|x)} \qquad \theta^{(t)} \sim \pi(\theta|x)$$

[Newton & Raftery, 1994]

"The bad news is that the number of points required for this estimator to get close to the right answer will often be greater than the number of atoms in the observable universe. The even worse news is that it's easy for people to not realize this, and to naïvely accept estimates that are nowhere close to the correct value of the marginal likelihood." R. Neal's blog, 17/08/2008

Illustration: Normal mean (Neal, 2008)

Take $X|\theta \sim \mathcal{N}(\theta, \sigma_1^2)$ and $\theta \sim \mathcal{N}(0, \sigma_0^2)$
Define

```
harmonic.mean.marg.lik <- function (x, s0, s1, n)
   { post.prec <- 1/s0 + 1/s1
     t <- rnorm(n,(x/s1)/post.prec,sqrt(1/post.prec))
     lik <- dnorm(x,t,s1)
     1/mean(1/lik)
   }
```

Illustration: Normal mean (Neal, 2008)

Take $X|\theta \sim \mathcal{N}(\theta, \sigma_1^2)$ and $\theta \sim \mathcal{N}(0, \sigma_0^2)$

```
> for (i in 1:5)
   +   print(harmonic.mean.marg.lik(2,10,1,1e7))
    [1] 0.08439447
    [1] 0.0989342
    [1] 0.0973829
    [1] 0.08654892
    [1] 0.09364961
> true.marg.lik(2,10,1)
    [1] 0.03891791
```

Worst Monte Carlo Method Ever

"My characterization of the harmonic mean of the likelihood as the Worst Monte Carlo Method Ever is based not just on its abysmal performance in most real problem, nor just on the fact that users of the method generally do not realize its poor performance, but also on the continued use of this method despite these flaws, due partly to wishful thinking on the part of its users, but also due to the connivance or negligence of many in the statistical community who ought to know better." R. Neal's blog, 17/08/2008

Optimal importance function

The choice of g that minimizes the variance of the importance sampling estimator is

$$g^*(x) = \frac{|h(x)|\, f(x)}{\int_{\mathcal{Z}} |h(z)|\, f(z)\, dz}\,.$$

Rather formal optimality result since optimal choice of $g^*(x)$ requires the knowledge of $\mathfrak{I}$, the integral of interest!

Optimal importance function

The choice of g that minimizes the variance of the importance sampling estimator is

$$g^*(x) = \frac{|h(x)|\, f(x)}{\int_{\mathcal{Z}} |h(z)|\, f(z)\, dz}\,.$$

Rather formal optimality result since optimal choice of $g^*(x)$ requires the knowledge of $\mathfrak{I}$, the integral of interest!

Practical impact

$$\frac{\sum_{j=1}^{m} h(X_j)\, f(X_j)/g(X_j)}{\sum_{j=1}^{m} f(X_j)/g(X_j)},$$

where f and g are known up to constants.

- Also converges to $\mathfrak{I}$ by the Strong Law of Large Numbers.
- Biased, but the bias is quite small
- In some settings beats the unbiased estimator in squared error loss.
- Using the 'optimal' solution does not always work:

$$\frac{\sum_{j=1}^{m} h(x_j)\, f(x_j)/|h(x_j)|\, f(x_j)}{\sum_{j=1}^{m} f(x_j)/|h(x_j)|\, f(x_j)} = \frac{\#\text{positive } h - \#\text{negative } h}{\sum_{j=1}^{m} 1/|h(x_j)|}$$

Practical impact

$$\frac{\sum_{j=1}^{m} h(X_j)\, f(X_j)/g(X_j)}{\sum_{j=1}^{m} f(X_j)/g(X_j)},$$

where f and g are known up to constants.

- Also converges to $\mathfrak{I}$ by the Strong Law of Large Numbers.
- Biased, but the bias is quite small
- In some settings beats the unbiased estimator in squared error loss.
- Using the 'optimal' solution does not always work:

$$\frac{\sum_{j=1}^{m} h(x_j)\, f(x_j)/|h(x_j)|\, f(x_j)}{\sum_{j=1}^{m} f(x_j)/|h(x_j)|\, f(x_j)} = \frac{\#\text{positive } h - \#\text{negative } h}{\sum_{j=1}^{m} 1/|h(x_j)|}$$

Selfnormalised importance sampling

For ratio estimator

$$\delta_h^n = \sum_{i=1}^{n} \omega_i \, h(x_i) \Big/ \sum_{i=1}^{n} \omega_i$$

with $X_i \sim g(y)$ and W_i such that

$$\mathbb{E}[W_i | X_i = x] = \kappa f(x)/g(x)$$

Selfnormalised variance

then

$$\mathrm{var}(\delta_h^n) \approx \frac{1}{n^2\kappa^2}\left(\mathrm{var}(S_h^n) - 2\mathbb{E}^\pi[h]\,\mathrm{cov}(S_h^n, S_1^n) + \mathbb{E}^\pi[h]^2\,\mathrm{var}(S_1^n)\right) .$$

for

$$S_h^n = \sum_{i=1}^n W_i h(X_i)\,, \quad S_1^n = \sum_{i=1}^n W_i$$

Rough approximation

$$\mathrm{var}\delta_h^n \approx \frac{1}{n}\,\mathrm{var}^\pi(h(X))\{1 + \mathrm{var}_g(W)\}$$

Example (Student's t distribution)

$X \sim \mathcal{T}(\nu, \theta, \sigma^2)$, with density

$$f_\nu(x) = \frac{\Gamma((\nu+1)/2)}{\sigma\sqrt{\nu\pi}\,\Gamma(\nu/2)} \left(1 + \frac{(x-\theta)^2}{\nu\sigma^2}\right)^{-(\nu+1)/2}.$$

Without loss of generality, take $\theta = 0$, $\sigma = 1$.

Problem: Calculate the integral

$$\int_{2.1}^{\infty} \left(\frac{\sin(x)}{x}\right)^n f_\nu(x)\,dx.$$

Example (Student's t distribution (2))

- Simulation possibilities
  - Directly from f_ν, since $f_\nu = \frac{\mathcal{N}(0,1)}{\sqrt{\chi^2_\nu}}$
  - Importance sampling using Cauchy $\mathcal{C}(0,1)$
  - Importance sampling using a normal $\mathcal{N}(0,1)$ (expected to be nonoptimal)
  - Importance sampling using a $\mathcal{U}([0,1/2.1])$ change of variables

Example (Student's t distribution (2))

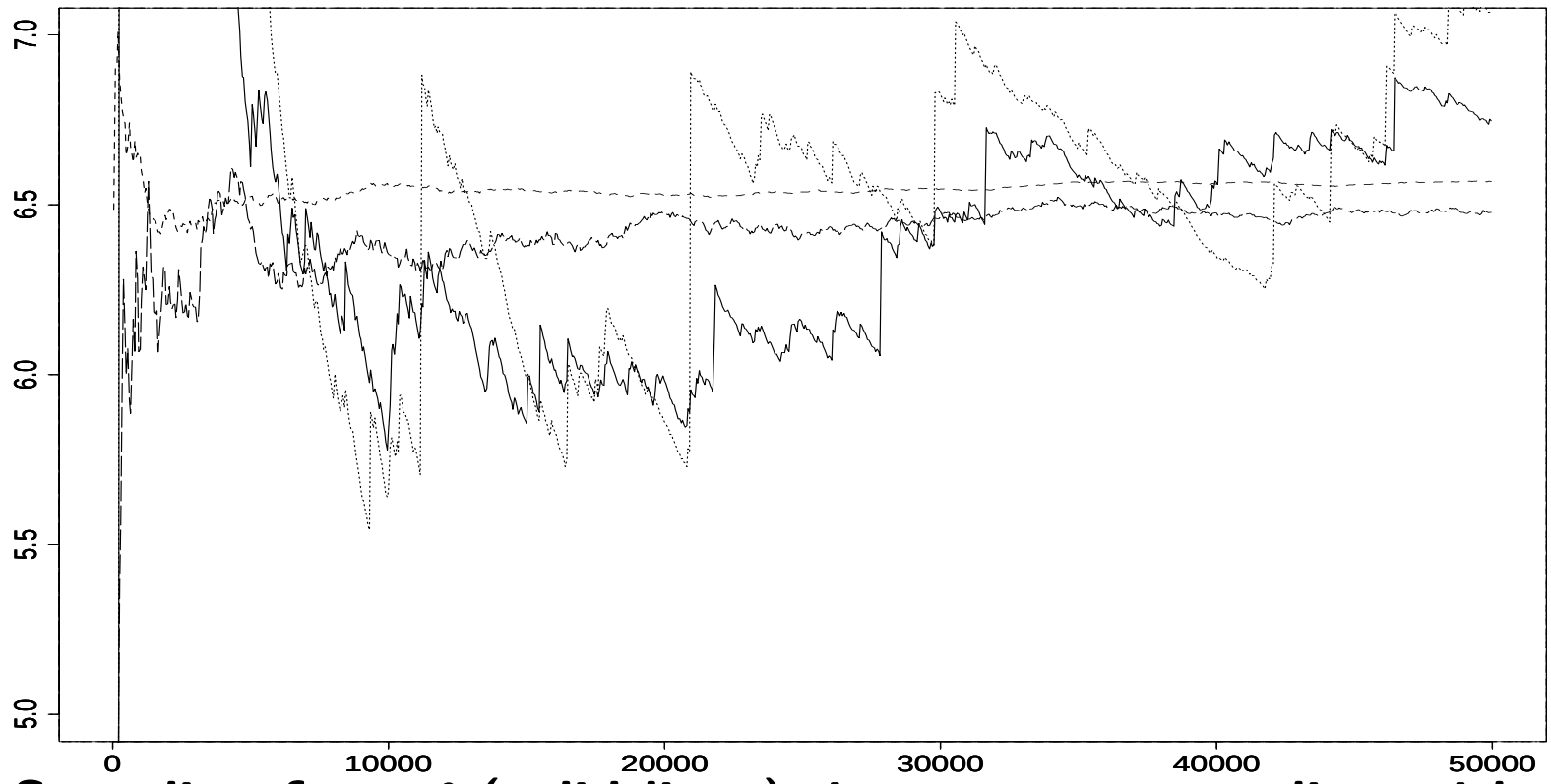
- Simulation possibilities
  - Directly from f_ν, since $f_\nu = \frac{\mathcal{N}(0,1)}{\sqrt{\chi^2_\nu}}$
  - Importance sampling using Cauchy $\mathcal{C}(0,1)$
  - Importance sampling using a normal $\mathcal{N}(0,1)$ (expected to be nonoptimal)
  - Importance sampling using a $\mathcal{U}([0,1/2.1])$ change of variables

Example (Student's t distribution (2))

- Simulation possibilities
  - Directly from f_ν, since $f_\nu = \frac{\mathcal{N}(0,1)}{\sqrt{\chi^2_\nu}}$
  - Importance sampling using Cauchy $\mathcal{C}(0,1)$
  - Importance sampling using a normal $\mathcal{N}(0,1)$ (expected to be nonoptimal)
  - Importance sampling using a $\mathcal{U}([0,1/2.1])$ change of variables

Example (Student's t distribution (2))

- Simulation possibilities
  - Directly from f_ν, since $f_\nu = \frac{\mathcal{N}(0,1)}{\sqrt{\chi^2_\nu}}$
  - Importance sampling using Cauchy $\mathcal{C}(0,1)$
  - Importance sampling using a normal $\mathcal{N}(0,1)$ (expected to be nonoptimal)
  - Importance sampling using a $\mathcal{U}([0,1/2.1])$ change of variables



Sampling from f (solid lines), importance sampling with Cauchy instrumental (short dashes), $\mathscr{U}([0, 1/2.1])$ instrumental (long dashes) and normal instrumental (dots).

Acceleration methods

Interlude #5: IS suffers from curse of dimensionality

As dimension increases, discrepancy between importance and target worsens

▸ skip explanation

Explanation:

Take target distribution μ and instrumental distribution ν

Simulation of a sample of iid samples of size n $x_{1:n}$ from $\mu_n = \mu^{\bigotimes n}$

Importance sampling estimator for $\mu_n(f_n) = \int f_n(x_{1:n})\mu_n(dx_{1:n})$

$$\widehat{\mu_n(f_n)} = \frac{\sum_{i=1}^N f_n(\xi_{1:n}^i)\prod_{j=1}^N W_j^i}{\sum_{j=1}^N \prod_{j=1}^N W_j},$$

where $W_k^i = \frac{d\mu}{d\nu}(\xi_k^i)$, and ξ_j^i are iid with distribution ν.

For $\{V_k\}_{k\geq 0}$, sequence of iid nonnegative random variables and for $n \geq 1$, $\mathcal{F}_n = \sigma(V_k; k \leq n)$, set

$$U_n = \prod_{k=1}^n V_k$$

Interlude #5: IS suffers from curse of dimensionality

As dimension increases, discrepancy between importance and target worsens

skip explanation

Explanation:

Take target distribution μ and instrumental distribution ν

Simulation of a sample of iid samples of size n $x_{1:n}$ from $\mu_n = \mu^{\bigotimes n}$

Importance sampling estimator for $\mu_n(f_n) = \int f_n(x_{1:n})\mu_n(dx_{1:n})$

$$\widehat{\mu_n(f_n)} = \frac{\sum_{i=1}^N f_n(\xi_{1:n}^i) \prod_{j=1}^N W_j^i}{\sum_{j=1}^N \prod_{j=1}^N W_j},$$

where $W_k^i = \frac{d\mu}{d\nu}(\xi_k^i)$, and ξ_j^i are iid with distribution ν.

For $\{V_k\}_{k\geq 0}$, sequence of iid nonnegative random variables and for $n \geq 1$, $\mathcal{F}_n = \sigma(V_k; k \leq n)$, set

$$U_n = \prod_{k=1}^n V_k$$

Interlude #5: IS suffers (2)

Since $\mathbb{E}[V_{n+1}] = 1$ and V_{n+1} independent from $\mathcal{F}_n$,

$$\mathbb{E}(U_{n+1} \mid \mathcal{F}_n) = U_n \mathbb{E}(V_{n+1} \mid \mathcal{F}_n) = U_n,$$

and thus $\{U_n\}_{n \geq 0}$ **martingale**
Since $x \mapsto \sqrt{x}$ concave, by Jensen's inequality,

$$\mathbb{E}(\sqrt{U_{n+1}} \mid \mathcal{F}_n) \leq \sqrt{\mathbb{E}(U_{n+1} \mid \mathcal{F}_n)} \leq \sqrt{U_n}$$

and thus $\{\sqrt{U_n}\}_{n \geq 0}$ **supermartingale**
Assume $\mathbb{E}(\sqrt{V_{n+1}}) < 1$. Then

$$\mathbb{E}(\sqrt{U_n}) = \prod_{k=1}^{n} \mathbb{E}(\sqrt{V_k}) \to 0, \quad n \to \infty.$$

Interlude #5: IS suffers (3)

But $\{\sqrt{U_n}\}_{n\geq 0}$ is a nonnegative supermartingale and thus $\sqrt{U_n}$ converges a.s. to a random variable $Z \geq 0$. By **Fatou's lemma**,

$$\mathbb{E}(Z) = \mathbb{E}\left(\lim_{n\to\infty} \sqrt{U_n}\right) \leq \liminf_{n\to\infty} \mathbb{E}(\sqrt{U_n}) = 0.$$

Hence, $Z = 0$ and $U_n \to 0$ a.s., which implies that the martingale $\{U_n\}_{n\geq 0}$ is not regular.
Apply these results to $V_k = \frac{d\mu}{d\nu}(\xi_k^i)$, $i \in \{1, \ldots, N\}$:

$$\mathbb{E}\left[\sqrt{\frac{d\mu}{d\nu}(\xi_k^i)}\right] \leq \mathbb{E}\left[\frac{d\mu}{d\nu}(\xi_k^i)\right] = 1.$$

with equality iff $\frac{d\mu}{d\nu} = 1$, ν-a.e., i.e. $\mu = \nu$.

Thus all importance weights converge to 0

▸ too volatile!

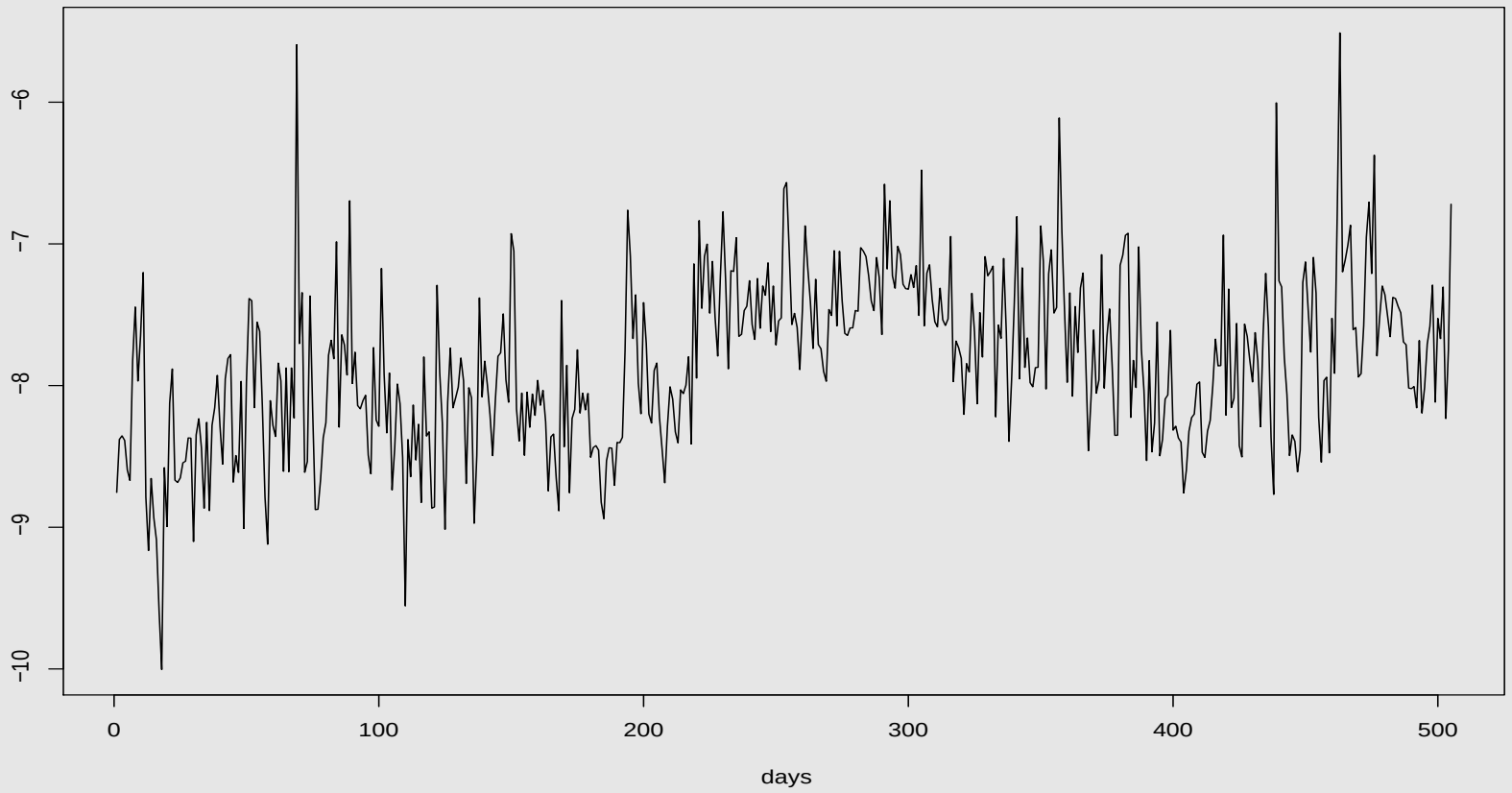
Example (Stochastic volatility model)

$$y_t = \beta \exp(x_t/2)\, \epsilon_t\,, \qquad \epsilon_t \sim \mathcal{N}(0,1)$$

with AR(1) log-variance process (or *volatility*)

$$x_{t+1} = \varphi x_t + \sigma u_t\,, \quad u_t \sim \mathcal{N}(0,1)$$

Evolution of IBM stocks (corrected from trend and log-ratio-ed)



Example (Stochastic volatility model (2))

Observed likelihood unavailable in closed from.
Joint posterior (or conditional) distribution of the hidden state sequence $\{X_k\}_{1\leq k\leq K}$ can be evaluated explicitly

$$\prod_{k=2}^{K} \exp -\left\{\sigma^{-2}(x_k - \phi x_{k-1})^2 + \beta^{-2}\exp(-x_k)y_k^2 + x_k\right\}/2\,, \quad (1)$$

up to a normalizing constant.

Computational problems

Example (Stochastic volatility model (3))

Direct simulation from this distribution impossible because of

(a) dependence among the X_k's,

(b) dimension of the sequence $\{X_k\}_{1 \leq k \leq K}$, and

(c) exponential term $\exp(-x_k)y_k^2$ within (1).

Importance sampling

Example (Stochastic volatility model (4))

Natural candidate: replace the exponential term with a quadratic approximation to preserve Gaussianity.
E.g., expand $\exp(-x_k)$ around its conditional expectation ϕx_{k-1} as

$$\exp(-x_k) \approx \exp(-\phi x_{k-1}) \left\{ 1 - (x_k - \phi x_{k-1}) + \frac{1}{2}(x_k - \phi x_{k-1})^2 \right\}$$

Example (Stochastic volatility model (5))

Corresponding Gaussian importance distribution with mean

$$\mu_k = \frac{\phi x_{k-1}\{\sigma^{-2} + y_k^2 \exp(-\phi x_{k-1})/2\} - \{1 - y_k^2 \exp(-\phi x_{k-1})\}/2}{\sigma^{-2} + y_k^2 \exp(-\phi x_{k-1})/2}$$

and variance

$$\tau_k^2 = (\sigma^{-2} + y_k^2 \exp(-\phi x_{k-1})/2)^{-1}$$

Prior proposal on X_1,

$$X_1 \sim \mathcal{N}(0, \sigma^2)$$

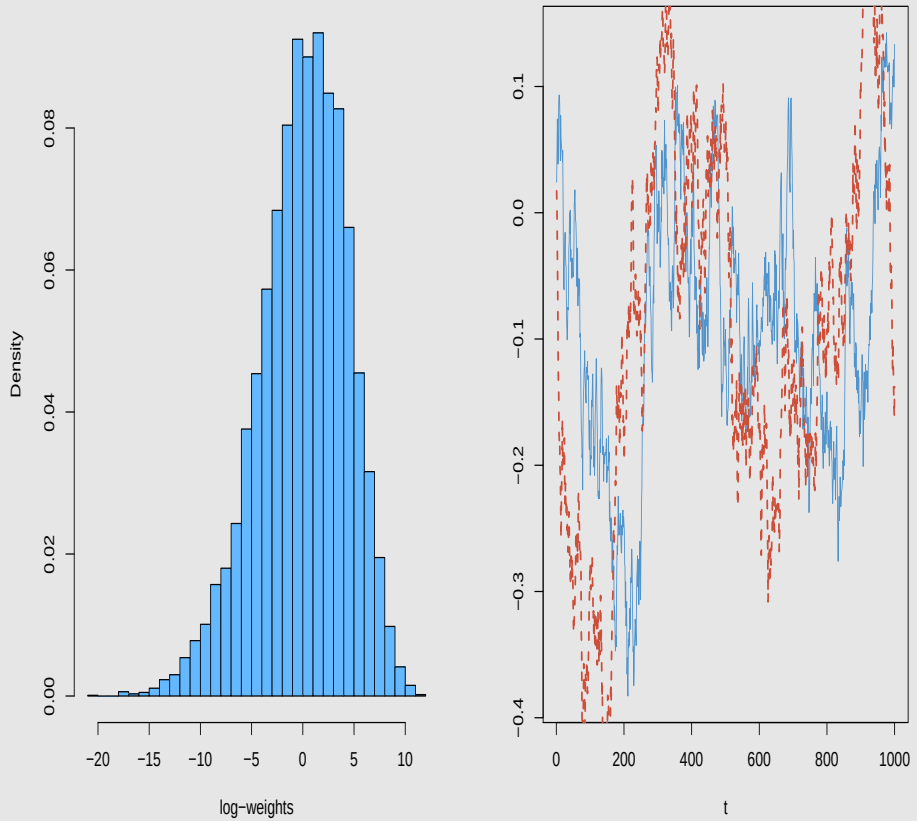
Example (Stochastic volatility model (6))

Simulation starts with X_1 and proceeds forward to X_n, each X_k being generated conditional on Y_k and the previously generated X_{k-1}.

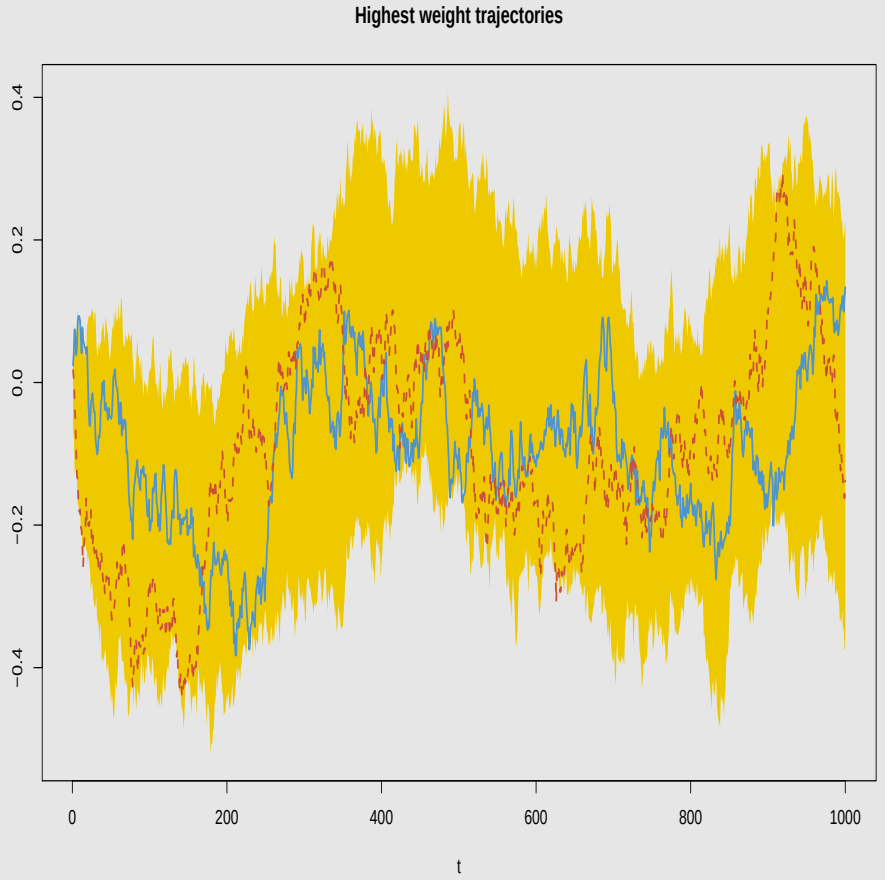
Importance weight computed sequentially as the product of

$$\frac{\exp -\left\{\sigma^{-2}(x_k - \phi x_{k-1})^2 + \exp(-x_k) y_k^2 + x_k\right\}/2}{\exp -\left\{\tau_k^{-2}(x_k - \mu_k)^2\right\}\tau_k^{-1}}.$$

$(1 \leq k \leq K)$



Histogram of the logarithms of the importance weights (left) and comparison between the true volatility and the best fit, based on $10,000$ simulated importance samples.



Corresponding range of the simulated $\{X_k\}_{1 \leq k \leq 100}$, compared with the true value.

Acceleration methods

Correlated simulations

Negative correlation reduces variance

Special technique — but efficient when it applies

Two samples $(X_1, \ldots, X_m)$ and $(Y_1, \ldots, Y_m)$ from f to estimate

$$\mathfrak{I} = \int_{\mathbb{R}} h(x) f(x) dx$$

by

$$\widehat{\mathfrak{I}}_1 = \frac{1}{m} \sum_{i=1}^{m} h(X_i) \quad \text{and} \quad \widehat{\mathfrak{I}}_2 = \frac{1}{m} \sum_{i=1}^{m} h(Y_i)$$

with mean $\mathfrak{I}$ and variance σ^2

Variance reduction

Variance of the average

$$\operatorname{var}\left(\frac{\widehat{\mathfrak{I}}_1 + \widehat{\mathfrak{I}}_2}{2}\right) = \frac{\sigma^2}{2} + \frac{1}{2}\operatorname{cov}(\widehat{\mathfrak{I}}_1, \widehat{\mathfrak{I}}_2).$$

If the two samples are **negatively correlated**,

$$\operatorname{cov}(\widehat{\mathfrak{I}}_1, \widehat{\mathfrak{I}}_2) \leq 0\,,$$

they improve on two independent samples of same size

Antithetic variables

- If f symmetric about μ, take $Y_i = 2\mu - X_i$
- If $X_i = F^{-1}(U_i)$, take $Y_i = F^{-1}(1 - U_i)$
- If $(A_i)_i$ partition of $\mathcal{X}$, **partitioned sampling** by sampling X_j's in each A_i (requires to know $\Pr(A_i)$)

Control variates

▸ out of control!

For

$$\mathfrak{I} = \int h(x)f(x)dx$$

unknown and

$$\mathfrak{I}_0 = \int h_0(x)f(x)dx$$

known,

$\mathfrak{I}_0$ estimated by $\widehat{\mathfrak{I}}_0$ and

$\mathfrak{I}$ estimated by $\widehat{\mathfrak{I}}$

Control variates (2)

Combined estimator

$$\widehat{\mathfrak{I}}^* = \widehat{\mathfrak{I}} + \beta(\widehat{\mathfrak{I}}_0 - \mathrm{I}_0)$$

$\widehat{\mathfrak{I}}^*$ **is unbiased for** $\mathfrak{I}$ **and**

$$\mathrm{var}(\widehat{\mathfrak{I}}^*) = \mathrm{var}(\widehat{\mathfrak{I}}) + \beta^2 \mathrm{var}(\widehat{\mathfrak{I}}) + 2\beta \mathrm{cov}(\widehat{\mathfrak{I}}, \widehat{\mathfrak{I}}_0)$$

Optimal control

Optimal choice of β

$$\beta^\star = -\frac{\mathrm{cov}(\widehat{\mathfrak{I}}, \widehat{\mathfrak{I}}_0)}{\mathrm{var}(\widehat{\mathfrak{I}}_0)} \ ,$$

with

$$\mathrm{var}(\widehat{\mathfrak{I}}^\star) = (1 - \rho^2)\ \mathrm{var}(\widehat{\mathfrak{I}}) \ ,$$

where ρ correlation between $\widehat{\mathfrak{I}}$ and $\widehat{\mathfrak{I}}_0$
Usual solution: **regression coefficient of** $h(x_i)$ **over** $h_0(x_i)$

Illustration

Example (Quantile Approximation)

Evaluate

$$\rho = \Pr(X > a) = \int_a^{\infty} f(x)dx$$

by

$$\widehat{\rho} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i > a),$$

with X_i iid f.
If $\Pr(X > \mu) = \frac{1}{2}$ known

Illustration

Example (Quantile Approximation (2))

Control variate

$$\tilde{\rho} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i > a) + \beta\left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i > \mu) - \Pr(X > \mu)\right)$$

improves upon $\widehat{\rho}$ if

$$\beta < 0 \quad \text{and} \quad |\beta| < 2\frac{\operatorname{cov}(\widehat{\rho}, \widehat{\rho}_0)}{\operatorname{var}(\widehat{\rho}_0)} 2\frac{\Pr(X > a)}{\Pr(X > \mu)}.$$

Illustration

Example (Quantile Approximation (2))

Control variate

$$\tilde{\rho} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i > a) + \beta\left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i > \mu) - \Pr(X > \mu)\right)$$

improves upon $\widehat{\rho}$ if

$$\beta < 0 \quad \text{and} \quad |\beta| < 2\frac{\operatorname{cov}(\widehat{\rho}, \widehat{\rho}_0)}{\operatorname{var}(\widehat{\rho}_0)} 2\frac{\Pr(X > a)}{\Pr(X > \mu)}.$$

Integration by conditioning

Use **Rao-Blackwell Theorem**

$$\boldsymbol{var}(\mathbb{E}[\delta(\mathbf{X})|\mathbf{Y}]) \leq \boldsymbol{var}(\delta(\mathbf{X}))$$

Consequence

If $\widehat{\mathfrak{I}}$ unbiased estimator of $\mathfrak{I} = \mathbb{E}_f[h(X)]$, with X simulated from a joint density $\tilde{f}(x, y)$, where

$$\int \tilde{f}(x, y)dy = f(x),$$

the estimator

$$\widehat{\mathfrak{I}}^* = \mathbb{E}_{\tilde{f}}[\widehat{\mathfrak{I}}|Y_1, \ldots, Y_n]$$

dominate $\widehat{\mathfrak{I}}(X_1, \ldots, X_n)$ variance-wise (and is unbiased)

Illustration

skip expectation

Example (Student's t expectation)

For

$$\mathbb{E}[h(x)] = \mathbb{E}[\exp(-x^2)] \quad \text{with} \quad X \sim \mathcal{T}(\nu, 0, \sigma^2)$$

a Student's t distribution can be simulated as

$$X|y \sim \mathcal{N}(\mu, \sigma^2 y) \qquad \text{and} \qquad Y^{-1} \sim \chi^2_\nu.$$

Illustration

Example (Student's t expectation (2))

Empirical distribution

$$\frac{1}{m}\sum_{j=1}^{m} \exp(-X_j^2)\,,$$

can be improved from the joint sample

$$((X_1, Y_1), \ldots, (X_m, Y_m))$$

since

$$\frac{1}{m}\sum_{j=1}^{m} \mathbb{E}[\exp(-X^2)|Y_j] = \frac{1}{m}\sum_{j=1}^{m} \frac{1}{\sqrt{2\sigma^2 Y_j + 1}}$$

is the conditional expectation.
In this example, precision **ten times** better

Illustration

Example (Student's t expectation (2))

Empirical distribution

$$\frac{1}{m}\sum_{j=1}^{m} \exp(-X_j^2)\,,$$

can be improved from the joint sample

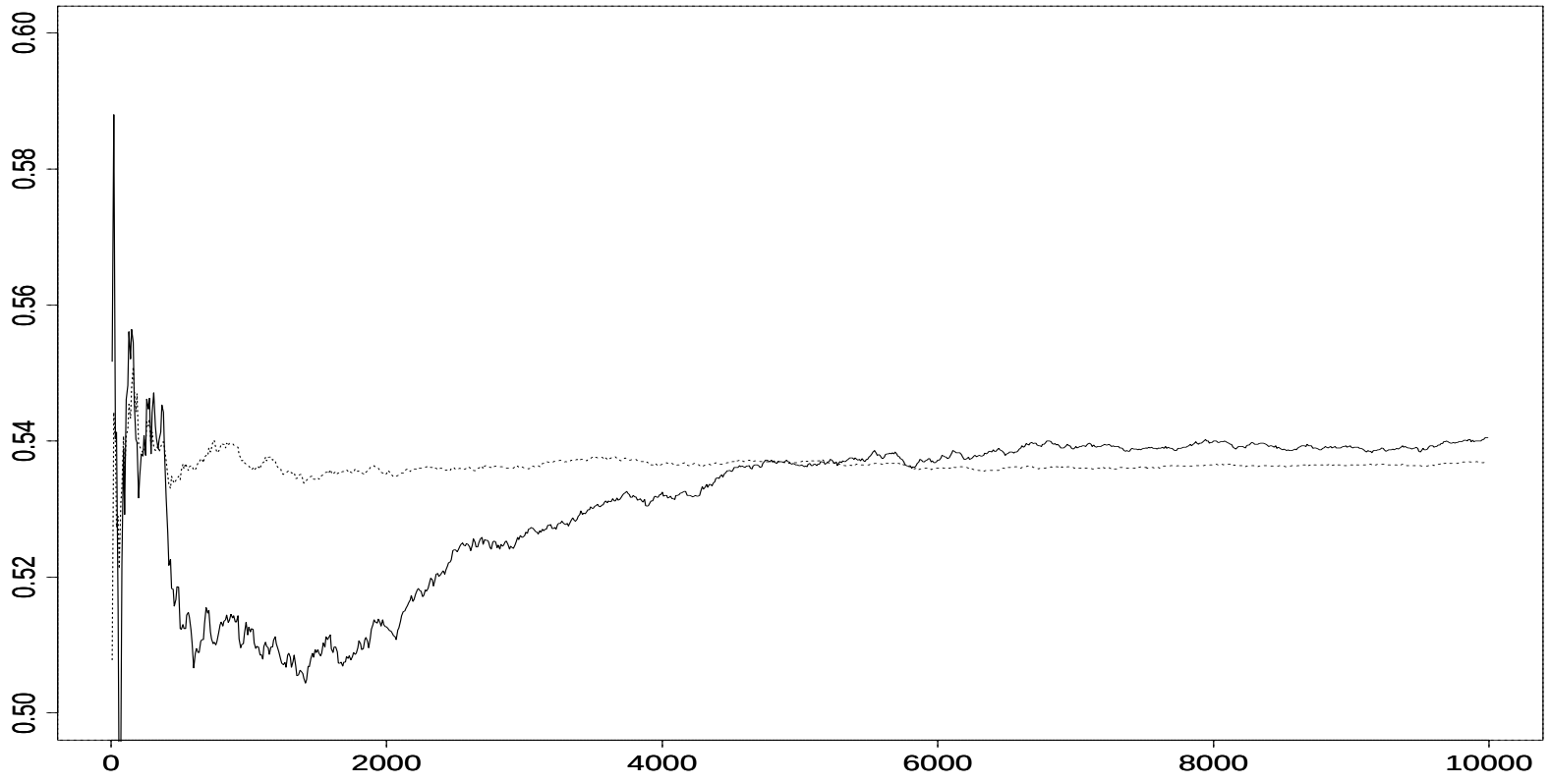
$$((X_1, Y_1), \ldots, (X_m, Y_m))$$

since

$$\frac{1}{m}\sum_{j=1}^{m} \mathbb{E}[\exp(-X^2)|Y_j] = \frac{1}{m}\sum_{j=1}^{m} \frac{1}{\sqrt{2\sigma^2 Y_j + 1}}$$

is the conditional expectation.
In this example, precision **ten times** better

Illustration



Estimators of $\mathbb{E}[\exp(-X^2)]$: empirical average (full) and conditional expectation (dotted) for $(\nu, \mu, \sigma) = (4.6, 0, 1)$.

Interlude #6: Rao-Blackwellisation

Interlude #5: Accept–Reject method

Given a density $f(\cdot)$ to simulate take $g(\cdot)$ density such that

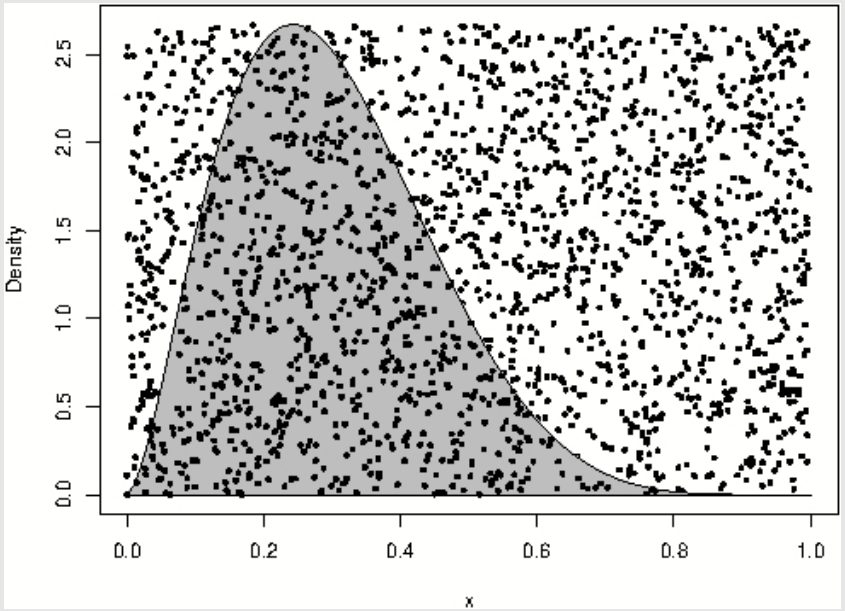
$$f(x) \leq Mg(x)$$

for $M \geq 1$
To simulate $X \sim f$, it is sufficient to generate

$$Y \sim g \; U|Y = y \sim \mathcal{U}(0, Mg(y))$$

until

$$0 < u < f(y)$$



Interlude #5: Demarginalisation

Raw outcome: id sequences $Y_1, Y_2, \ldots, Y_t \sim g$ and $U_1, U_2, \ldots, U_t \sim \mathcal{U}(0,1)$
Random number of accepted Y_i's

$$\mathbb{P}(N = n) = \binom{n-1}{t-1} (1/M)^t (1 - 1/M)^{n-t} ,$$

Interlude #5: Demarginalisation

Raw outcome: id sequences $Y_1, Y_2, \ldots, Y_t \sim g$ and $U_1, U_2, \ldots, U_t \sim \mathcal{U}(0,1)$
Joint density of $(N, \mathbf{Y}, \mathbf{U})$

$$\begin{aligned}
&\mathbb{P}(N = n, Y_1 \leq y_1, \ldots, Y_n \leq y_n, U_1 \leq u_1, \ldots, U_n \leq u_n) \\
&= \int_{-\infty}^{y_n} g(t_n)(u_n \wedge w_n)\,dt_n \int_{-\infty}^{y_1} \ldots \int_{-\infty}^{y_{n-1}} g(t_1)\ldots g(t_{n-1}) \\
&\quad \times \sum_{(i_1,\cdots,i_{t-1})} \prod_{j=1}^{t-1} (w_{i_j} \wedge u_{i_j}) \prod_{j=t}^{n-1} (u_{i_j} - w_{i_j})^+ dt_1 \cdots dt_{n-1},
\end{aligned}$$

where $w_i = f(y_i)/Mg(y_i)$ and sum over all subsets of $\{1, \ldots, n-1\}$ of size $t-1$

Interlude #5: Demarginalisation

Raw outcome: id sequences $Y_1, Y_2, \ldots, Y_t \sim g$ and $U_1, U_2, \ldots, U_t \sim \mathcal{U}(0,1)$
Marginal joint density of $(Y_i, U_i)|N = n,\ i < n$

$$\begin{aligned}
&\mathbb{P}(N = n, Y_1 \leq y, U_1 \leq u_1) \\
&\quad = \binom{n-1}{t-1} \left(\frac{1}{M}\right)^{t-1} \left(1 - \frac{1}{M}\right)^{n-t-1} \\
&\quad \times \left[\frac{t-1}{n-1}(w_1 \wedge u_1)\left(1 - \frac{1}{M}\right) + \frac{n-t}{n-1}(u_1 - w_1)^+ \left(\frac{1}{M}\right)\right] \int_{-\infty}^{y} g(t_1)\,dt_1
\end{aligned}$$

and marginal distribution of Y_i

$$m(y) = {}^{t-1}\!/_{n-1} f(y) + {}^{n-t}\!/_{n-1} \frac{g(y) - \rho f(y)}{1 - \rho}$$

$$\mathbb{P}(U_1 \leq w(y)|Y_1 = y, N = n) = \frac{g(y) w(y) M^{t-1}\!/_{n-1}}{m(y)}$$

Interlude #6: Demarginalisation

Accept-reject sample $(X_1, \ldots, X_m)$ associated with $(U_1, \ldots, U_N)$ and $(Y_1, \ldots, Y_N)$
N is stopping time for acceptance of m variables among Y_j's
Rewrite estimator of $\mathbb{E}[h]$ as

$$\frac{1}{m} \sum_{i=1}^{m} h(X_i) = \frac{1}{m} \sum_{j=1}^{N} h(Y_j)\, \mathbb{I}_{U_j \leq w_j}\,,$$

with $w_j = {}^{f(Y_j)}/_{Mg(Y_j)}$

[Robert & Casella, 1996]

Interlude #6: Demarginalisation

Rao-Blackwellisation: smaller variance produced by integrating out the U_i's,

$$\frac{1}{m}\sum_{j=1}^{N}\mathbb{E}[\mathbb{I}_{U_j\leq w_j}|N,Y_1,\ldots,Y_N]\,h(Y_j)=\frac{1}{m}\sum_{i=1}^{N}\rho_i h(Y_i),$$

where $(i<n)$

$$\begin{aligned}\rho_i &= \mathbb{P}(U_i\leq w_i|N=n,Y_1,\ldots,Y_n)\\ &= w_i\frac{\sum_{(i_1,\ldots,i_{m-2})}\prod_{j=1}^{m-2}w_{i_j}\prod_{j=m-1}^{n-2}(1-w_{i_j})}{\sum_{(i_1,\ldots,i_{m-1})}\prod_{j=1}^{m-1}w_{i_j}\prod_{j=m}^{n-1}(1-w_{i_j})},\end{aligned}$$

and $\rho_n=1$.
Numerator sum over all subsets of $\{1,\ldots,i-1,i+1,\ldots,n-1\}$ of size $m-2$, and denominator sum over all subsets of size $m-1$

[Robert & Casella, 1996]

Monte Carlo Optimization

Simulation for optimisation

- Two uses of computer-generated random variables to solve optimization problems.
- first use is to produce stochastic search techniques
  - To reach the maximum (or minimum) of a function
  - Avoid being trapped in local maxima (or minima)
  - Are sufficiently attracted by the global maximum (or minimum).
- The second use of simulation is to approximate the function to be optimized.

Introduction

- Optimization problems can mostly be seen as one of two kinds:
  - Find the extrema of a function $h(\theta)$ over a domain Θ
  - Find the solution(s) to an implicit equation $g(\theta) = 0$ over a domain Θ.

- The problems are exchangeable
  - The second one is a minimization problem for a function like $h(\theta) = g^2(\theta)$
  - while the first one is equivalent to solving $\partial h(\theta)/\partial\theta = 0$
- We only focus on the maximization problem

Deterministic or Stochastic

- Similar to integration, optimization can be deterministic or stochastic
- Deterministic: performance dependent on properties of the function
  - such as convexity, boundedness, and smoothness
- Stochastic (simulation)
  - Properties of $\mathfrak{h}$ play a lesser role in simulation-based approaches.
- Therefore, if $\mathfrak{h}$ is complex or Θ is irregular, chose the stochastic approach.

Numerical Optimization

- R has several embedded functions to solve optimization problems
  - The simplest one is optimize (one dimensional)

Example

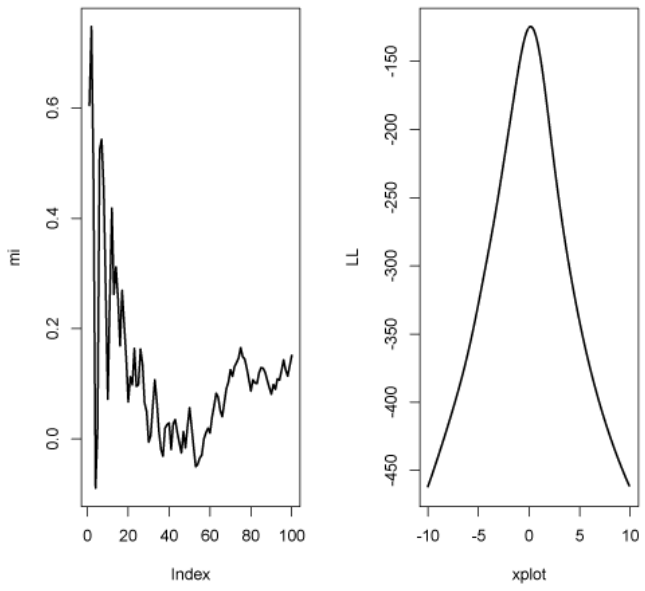
Maximizing a Cauchy likelihood $\mathcal{C}(\theta, 1)$

- When maximizing the likelihood of a Cauchy $\mathcal{C}(\theta, 1)$ sample,

$$\ell(\theta|x_1, \ldots, x_n) = \prod_{i=1}^{n} \frac{1}{1 + (x_i - \theta)^2},$$

- The sequence of maxima (MLEs) $\to \theta^* = 0$ when $n \to \infty$.
- But the journey is not a smooth one...

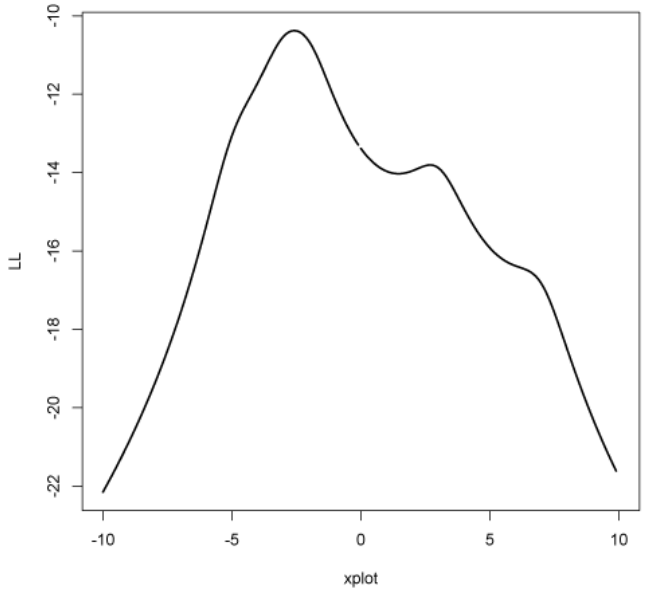
Cauchy Likelihood (2)



- MLEs *(left)* at each sample size, $n = 1,500$, and plot of final likelihood *(right)*.
  - Why are the MLEs so wiggly?
  - The likelihood is not as well-behaved as it seems

Cauchy Likelihood (3)

- The likelihood $\ell(\theta|x_1, \ldots, x_n) = \prod_{i=1}^{n} \frac{1}{1+(x_i-\theta)^2}$
  - Is like a polynomial of degree $2n$
  - The derivative has $2n$ zeros
  - Hard to see if $n = 500$
  - Here is $n = 5$



Newton-Raphson

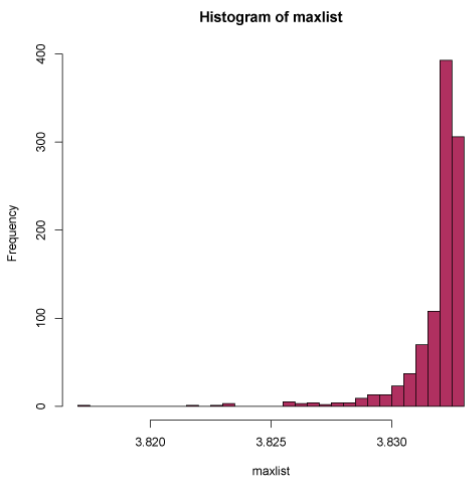
- Similarly, nlm is a generic R function uses the Newton–Raphson method
- Based on the recurrence relation

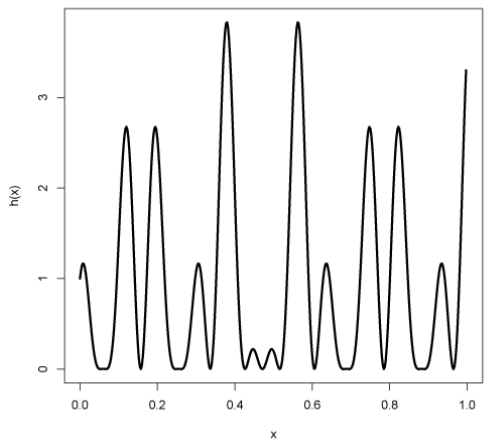
$$\theta_{i+1} = \theta_i - \left[\frac{\partial^2 h}{\partial\theta\partial\theta^T}(\theta_i)\right]^{-1} \frac{\partial h}{\partial\theta}(\theta_i)$$

- where the matrix of the second derivatives is the *Hessian*
  - This method is perfect when h is quadratic
    - But may also deteriorate when h is highly nonlinear
  - It also obviously depends on the starting point θ_0 when h has several minima.

A Basic Solution

- A natural if rudimentary way of using simulation to find $\max_\theta h(\theta)$
  - Simulate points over Θ according to an arbitrary distribution f positive on Θ
  - Until a high value of $h(\theta)$ is observed





- Recall $h(x) = [\cos(50x) + \sin(20x)]^2$
- Max=3.8325
- Histogram of 10^3 runs

Stochastic Gradient Methods

- Generating direct simulations from the target can be difficult.
- Different stochastic approach to maximization
  - Explore the surface in a local manner.
  - Can use $\theta_{j+1} = \theta_j + \epsilon_j$
  - A Markov Chain
  - The random component ϵ_j can be arbitrary
- Can also use features of the function: Newton-Raphson Variation

$$\theta_{j+1} = \theta_j + \alpha_j \nabla h(\theta_j)\,, \qquad \alpha_j > 0\,,$$

  - Where $\nabla h(\theta_j)$ is the gradient
  - α_j the step size

Stochastic Gradient Methods (2)

- In difficult problems
  - The gradient sequence will most likely get stuck in a local extremum of h.
- Stochastic Variation

$$\nabla h(\theta_j) \approx \frac{h(\theta_j + \beta_j \zeta_j) - h(\theta_j + \beta_j \zeta_j)}{2\beta_j} \zeta_j = \frac{\Delta h(\theta_j, \beta_j \zeta_j)}{2\beta_j} \zeta_j \,,$$

  - (β_j) is a second decreasing sequence
  - ζ_j is uniform on the unit sphere $\|\zeta\| = 1$.
- We then use

$$\theta_{j+1} = \theta_j + \frac{\alpha_j}{2\beta_j} \Delta h(\theta_j, \beta_j \zeta_j) \, \zeta_j$$

Likelihood optimisation

Practical optimisation of the likelihood function

$$\theta^\star = \arg\max_\theta L(\theta|\mathbf{x}) = \prod_{i=1}^{n} g(X_i|\theta).$$

assuming $\mathbf{X} = (X_1, \ldots, X_n) \text{iid} g(x|\theta)$

- analytical resolution feasible for exponential families

$$\nabla T(\theta) \sum_{i=1}^{n} S(x_i) = n \nabla \tau(\theta)$$

- use of standard numerical techniques like Newton-Raphson

$$\theta^{(t+1)} = \theta^{(t)} + I^{obs}(X, \theta^{(t)})^{-1} \nabla \ell(\theta^{(t)})$$

with $\ell(.)$ log-likelihood and I^{obs} observed information matrix

Likelihood optimisation

Practical optimisation of the likelihood function

$$\theta^\star = \arg\max_\theta L(\theta|\mathbf{x}) = \prod_{i=1}^{n} g(X_i|\theta).$$

assuming $\mathbf{X} = (X_1, \ldots, X_n) \text{iid} g(x|\theta)$

- analytical resolution feasible for exponential families

$$\nabla T(\theta) \sum_{i=1}^{n} S(x_i) = n \nabla \tau(\theta)$$

- use of standard numerical techniques like Newton-Raphson

$$\theta^{(t+1)} = \theta^{(t)} + I^{\text{obs}}(\mathbf{X}, \theta^{(t)})^{-1} \nabla \ell(\theta^{(t)})$$

with $\ell(.)$ log-likelihood and I^{obs} observed information matrix

Likelihood optimisation

Practical optimisation of the likelihood function

$$\theta^\star = \arg\max_\theta L(\theta|\mathbf{x}) = \prod_{i=1}^{n} g(X_i|\theta).$$

assuming $\mathbf{X} = (X_1, \ldots, X_n) \text{iid} g(x|\theta)$

- analytical resolution feasible for exponential families

$$\nabla T(\theta) \sum_{i=1}^{n} S(x_i) = n \nabla \tau(\theta)$$

- use of standard numerical techniques like Newton-Raphson

$$\theta^{(t+1)} = \theta^{(t)} + I^{obs}(\mathbf{X}, \theta^{(t)})^{-1} \nabla \ell(\theta^{(t)})$$

with $\ell(.)$ log-likelihood and I^{obs} observed information matrix

EM algorithm

Cases where g is too complex for the above to work

Special case when g is a marginal

$$g(x|\theta) = \int_{\mathcal{Z}} f(x, z|\theta)\, dz$$

Z called latent or missing variable

Illustrations

- censored data

$$X = \min(X^*, a) \qquad X^* \sim \mathcal{N}(\theta, 1)$$

- mixture model

$$X \sim .3\,\mathcal{N}_1(\mu_0, 1) + .7\,\mathcal{N}_1(\mu_1, 1),$$

- desequilibrium model

$$X = \min(X^*, Y^*) \qquad X^* \sim f_1(x|\theta) \quad Y^* \sim f_2(x|\theta)$$

Completion

EM algorithm based on completing data $\mathbf{x}$ with $\mathbf{z}$, such as

$$(\mathbf{X}, \mathbf{Z}) \sim \mathrm{f}(\mathbf{x}, \mathbf{z}|\theta)$$

$\mathbf{Z}$ missing data vector and pair $(\mathbf{X}, \mathbf{Z})$ complete data vector

Conditional density of $\mathbf{Z}$ given $\mathbf{x}$:

$$\mathrm{k}(\mathbf{z}|\theta, \mathbf{x}) = \frac{\mathrm{f}(\mathbf{x}, \mathbf{z}|\theta)}{\mathrm{g}(\mathbf{x}|\theta)}$$

Completion

EM algorithm based on completing data $\mathbf{x}$ with $\mathbf{z}$, such as

$$(\mathbf{X}, \mathbf{Z}) \sim \mathrm{f}(\mathbf{x}, \mathbf{z}|\theta)$$

$\mathbf{Z}$ missing data vector and pair $(\mathbf{X}, \mathbf{Z})$ complete data vector

Conditional density of $\mathbf{Z}$ given $\mathbf{x}$:

$$\mathrm{k}(\mathbf{z}|\theta, \mathbf{x}) = \frac{\mathrm{f}(\mathbf{x}, \mathbf{z}|\theta)}{\mathrm{g}(\mathbf{x}|\theta)}$$

Likelihood decomposition

Likelihood associated with complete data $(\mathbf{x}, \mathbf{z})$

$$\mathrm{L}^{\mathrm{c}}(\theta|\mathbf{x}, \mathbf{z}) = \mathrm{f}(\mathbf{x}, \mathbf{z}|\theta)$$

and likelihood for observed data

$$\mathrm{L}(\theta|\mathbf{x})$$

such that

$$\log \mathrm{L}(\theta|\mathbf{x}) = \mathbb{E}[\log \mathrm{L}^{\mathrm{c}}(\theta|\mathbf{x}, \mathbf{Z})|\theta_0, \mathbf{x}] - \mathbb{E}[\log \mathrm{k}(\mathbf{Z}|\theta, \mathbf{x})|\theta_0, \mathbf{x}] \quad (2)$$

for any θ_0, with integration operated against conditionnal distribution of $\mathbf{Z}$ given observables (and parameters), $\mathrm{k}(\mathbf{z}|\theta_0, \mathbf{x})$

[A tale of] two θ's

There are "two θ's" ! : in (2), θ_0 is a fixed (and arbitrary) value driving integration, while θ both free (and variable)

Maximising observed likelihood

$$\mathrm{L}(\theta|\boldsymbol{x})$$

equivalent to maximise r.h.s. term in (2)

$$\mathbb{E}[\log \mathrm{L}^{\mathrm{c}}(\theta|\boldsymbol{x}, \boldsymbol{Z})|\theta_0, \boldsymbol{x}] - \mathbb{E}[\log \mathrm{k}(\boldsymbol{Z}|\theta, \boldsymbol{x})|\theta_0, \boldsymbol{x}]$$

[A tale of] two θ's

There are "two θ's" ! : in (2), θ_0 is a fixed (and arbitrary) value driving integration, while θ both free (and variable)

Maximising observed likelihood

$$\mathrm{L}(\theta|\mathbf{x})$$

equivalent to maximise r.h.s. term in (2)

$$\mathbb{E}[\log \mathrm{L}^{\mathrm{c}}(\theta|\mathbf{x}, \mathbf{Z})|\theta_0, \mathbf{x}] - \mathbb{E}[\log \mathrm{k}(\mathbf{Z}|\theta, \mathbf{x})|\theta_0, \mathbf{x}]$$

Intuition for EM

Instead of maximising wrt θ r.h.s. term in (2), maximise only

$$\mathbb{E}[\log \mathrm{L}^{\mathrm{c}}(\theta|\mathbf{x}, \mathbf{Z})|\theta_0, \mathbf{x}]$$

Maximisation of complete log-likelihood impossible since z unknown, hence substitute by maximisation of expected complete log-likelihood, with expectation depending on term θ_0

Intuition for EM

Instead of maximising wrt θ r.h.s. term in (2), maximise only

$$\mathbb{E}[\log \mathrm{L}^{\mathrm{c}}(\theta|\mathbf{x}, \mathbf{Z})|\theta_0, \mathbf{x}]$$

Maximisation of complete log-likelihood impossible since $\boldsymbol{z}$ unknown, hence substitute by maximisation of expected complete log-likelihood, with expectation depending on term θ_0

Expectation–Maximisation

Expectation of complete log-likelihood denoted

$$Q(\theta|\theta_0, \mathbf{x}) = \mathbb{E}[\log L^c(\theta|\mathbf{x}, \mathbf{Z})|\theta_0, \mathbf{x}]$$

to stress dependence on θ_0 and sample $\mathbf{x}$

Principle

EM derives sequence of estimators $\hat{\theta}_{(j)}$, $j = 1, 2, \ldots$, through iteration of **E**xpectation and **M**aximisation steps:

$$Q(\hat{\theta}_{(j)}|\hat{\theta}_{(j-1)}, \mathbf{x}) = \max_{\theta} Q(\theta|\hat{\theta}_{(j-1)}, \mathbf{x}).$$

Expectation–Maximisation

Expectation of complete log-likelihood denoted

$$Q(\theta|\theta_0, \mathbf{x}) = \mathbb{E}[\log L^c(\theta|\mathbf{x}, \mathbf{Z})|\theta_0, \mathbf{x}]$$

to stress dependence on θ_0 and sample $\mathbf{x}$

Principle

EM derives sequence of estimators $\hat{\theta}_{(j)}$, $j = 1, 2, \ldots$, through iteration of **E**xpectation and **M**aximisation steps:

$$Q(\hat{\theta}_{(j)}|\hat{\theta}_{(j-1)}, \mathbf{x}) = \max_{\theta} Q(\theta|\hat{\theta}_{(j-1)}, \mathbf{x}).$$

EM Algorithm

Iterate (in m)

1. (*step E*) Compute

$$Q(\theta|\hat{\theta}_{(m)}, \mathbf{x}) = \mathbb{E}[\log L^c(\theta|\mathbf{x}, \mathbf{Z})|\hat{\theta}_{(m)}, \mathbf{x}] \,,$$

2. (*step M*) Maximise $Q(\theta|\hat{\theta}_{(m)}, \mathbf{x})$ in θ and set

$$\hat{\theta}_{(m+1)} = \arg\max_{\theta} \; Q(\theta|\hat{\theta}_{(m)}, \mathbf{x}).$$

until a fixed point [of Q] is found

[Dempster, Laird, & Rubin, 1978]

Justification

Observed likelihood

$$L(\theta|\mathbf{x})$$

increases at every EM step

$$L(\hat{\theta}_{(m+1)}|\mathbf{x}) \geq L(\hat{\theta}_{(m)}|\mathbf{x})$$

[Exercice: use Jensen and (2)]

Censored data

Normal $\mathcal{N}(\theta, 1)$ sample right-censored

$$L(\theta|\mathbf{x}) = \frac{1}{(2\pi)^{m/2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{m}(x_i-\theta)^2\right\}[1-\Phi(a-\theta)]^{n-m}$$

Associated complete log-likelihood:

$$\log L^c(\theta|\mathbf{x},\mathbf{z}) \propto -\frac{1}{2}\sum_{i=1}^{m}(x_i-\theta)^2 - \frac{1}{2}\sum_{i=m+1}^{n}(z_i-\theta)^2\,,$$

where z_i's are censored observations, with density

$$k(z|\theta,\mathbf{x}) = \frac{\exp\{-\frac{1}{2}(z-\theta)^2\}}{\sqrt{2\pi}[1-\Phi(a-\theta)]} = \frac{\varphi(z-\theta)}{1-\Phi(a-\theta)}, \qquad a < z.$$

Censored data

Normal $\mathcal{N}(\theta, 1)$ sample right-censored

$$L(\theta|\mathbf{x}) = \frac{1}{(2\pi)^{m/2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{m}(x_i - \theta)^2\right\}[1 - \Phi(a - \theta)]^{n-m}$$

Associated complete log-likelihood:

$$\log L^c(\theta|\mathbf{x}, \mathbf{z}) \propto -\frac{1}{2}\sum_{i=1}^{m}(x_i - \theta)^2 - \frac{1}{2}\sum_{i=m+1}^{n}(z_i - \theta)^2 \,,$$

where z_i's are censored observations, with density

$$k(z|\theta, \mathbf{x}) = \frac{\exp\{-\frac{1}{2}(z-\theta)^2\}}{\sqrt{2\pi}[1 - \Phi(a - \theta)]} = \frac{\varphi(z - \theta)}{1 - \Phi(a - \theta)}, \qquad a < z.$$

Censored data (2)

At j-th EM iteration

$$
\begin{aligned}
Q(\theta|\hat{\theta}_{(j)}, \mathbf{x}) &\propto -\frac{1}{2}\sum_{i=1}^{m}(x_i-\theta)^2 - \frac{1}{2}\mathbb{E}\left[\sum_{i=m+1}^{n}(Z_i-\theta)^2 \,\middle|\, \hat{\theta}_{(j)}, \mathbf{x}\right] \\
&\propto -\frac{1}{2}\sum_{i=1}^{m}(x_i-\theta)^2 \\
&\quad -\frac{1}{2}\sum_{i=m+1}^{n}\int_a^{\infty}(z_i-\theta)^2 k(z|\hat{\theta}_{(j)}, \mathbf{x})\, dz_i
\end{aligned}
$$

Censored data (3)

Differenciating in θ,

$$n\,\hat{\theta}_{(j+1)} = m\bar{x} + (n-m)\mathbb{E}[Z|\hat{\theta}_{(j)}]\ ,$$

with

$$\mathbb{E}[Z|\hat{\theta}_{(j)}] = \int_a^\infty z k(z|\hat{\theta}_{(j)}, \mathbf{x})\,dz = \hat{\theta}_{(j)} + \frac{\varphi(a-\hat{\theta}_{(j)})}{1-\Phi(a-\hat{\theta}_{(j)})}.$$

Hence, EM sequence provided by

$$\hat{\theta}_{(j+1)} = \frac{m}{n}\bar{x} + \frac{n-m}{n}\left[\hat{\theta}_{(j)} + \frac{\varphi(a-\hat{\theta}_{(j)})}{1-\Phi(a-\hat{\theta}_{(j)})}\right],$$

which converges to likelihood maximum $\hat{\theta}$

Censored data (3)

Differenciating in θ,

$$n\,\hat{\theta}_{(j+1)} = m\bar{x} + (n-m)\mathbb{E}[Z|\hat{\theta}_{(j)}]\,,$$

with

$$\mathbb{E}[Z|\hat{\theta}_{(j)}] = \int_a^\infty z k(z|\hat{\theta}_{(j)}, \mathbf{x})\,dz = \hat{\theta}_{(j)} + \frac{\varphi(a-\hat{\theta}_{(j)})}{1-\Phi(a-\hat{\theta}_{(j)})}.$$

Hence, EM sequence provided by

$$\hat{\theta}_{(j+1)} = \frac{m}{n}\bar{x} + \frac{n-m}{n}\left[\hat{\theta}_{(j)} + \frac{\varphi(a-\hat{\theta}_{(j)})}{1-\Phi(a-\hat{\theta}_{(j)})}\right],$$

which converges to likelihood maximum $\hat{\theta}$

Mixtures

Mixture of two normal distributions with unknown means

$$.3\,\mathcal{N}_1(\mu_0, 1) + .7\,\mathcal{N}_1(\mu_1, 1),$$

sample $X_1, \ldots, X_n$ and parameter $\theta = (\mu_0, \mu_1)$

Missing data: $Z_i \in \{0, 1\}$, indicator of component associated with X_i ,

$$X_i | z_i \sim \mathcal{N}(\mu_{z_i}, 1) \qquad Z_i \sim \mathcal{B}(.7)$$

Complete likelihood

$$\begin{aligned} \log L^c(\theta | \mathbf{x}, \mathbf{z}) &\propto -\frac{1}{2}\sum_{i=1}^{n} z_i (x_i - \mu_1)^2 - \frac{1}{2}\sum_{i=1}^{n} (1 - z_i)(x_i - \mu_0)^2 \\ &= -\frac{1}{2} n_1 (\hat{\mu}_1 - \mu_1)^2 - \frac{1}{2}(n - n_1)(\hat{\mu}_0 - \mu_0)^2 \end{aligned}$$

with

$$n_1 = \sum_{i=1}^{n} z_i\,, \quad n_1 \hat{\mu}_1 = \sum_{i=1}^{n} z_i x_i\,, \quad (n - n_1)\hat{\mu}_0 = \sum_{i=1}^{n} (1 - z_i) x_i$$

Mixtures

Mixture of two normal distributions with unknown means

$$.3\,\mathcal{N}_1(\mu_0, 1) + .7\,\mathcal{N}_1(\mu_1, 1),$$

sample $X_1, \ldots, X_n$ and parameter $\theta = (\mu_0, \mu_1)$

Missing data: $Z_i \in \{0, 1\}$, indicator of component associated with X_i ,

$$X_i | z_i \sim \mathcal{N}(\mu_{z_i}, 1) \qquad Z_i \sim \mathcal{B}(.7)$$

Complete likelihood

$$\begin{aligned} \log L^c(\theta | \mathbf{x}, \mathbf{z}) &\propto -\frac{1}{2}\sum_{i=1}^{n} z_i (x_i - \mu_1)^2 - \frac{1}{2}\sum_{i=1}^{n} (1 - z_i)(x_i - \mu_0)^2 \\ &= -\frac{1}{2} n_1 (\hat{\mu}_1 - \mu_1)^2 - \frac{1}{2}(n - n_1)(\hat{\mu}_0 - \mu_0)^2 \end{aligned}$$

with

$$n_1 = \sum_{i=1}^{n} z_i\,, \quad n_1 \hat{\mu}_1 = \sum_{i=1}^{n} z_i x_i\,, \quad (n - n_1)\hat{\mu}_0 = \sum_{i=1}^{n} (1 - z_i) x_i$$

Mixtures (2)

At j-th EM iteration

$$Q(\theta|\hat{\theta}_{(j)}, \mathbf{x}) = \frac{1}{2}\mathbb{E}\left[n_1(\hat{\mu}_1 - \mu_1)^2 + (n - n_1)(\hat{\mu}_0 - \mu_0)^2|\hat{\theta}_{(j)}, \mathbf{x}\right]$$

Differenciating in θ

$$\hat{\theta}_{(j+1)} = \begin{pmatrix} \mathbb{E}\left[n_1\hat{\mu}_1 \middle| \hat{\theta}_{(j)}, \mathbf{x}\right] \Big/ \mathbb{E}\left[n_1|\hat{\theta}_{(j)}, \mathbf{x}\right] \\ \\ \mathbb{E}\left[(n - n_1)\hat{\mu}_0 \middle| \hat{\theta}_{(j)}, \mathbf{x}\right] \Big/ \mathbb{E}\left[(n - n_1)|\hat{\theta}_{(j)}, \mathbf{x}\right] \end{pmatrix}$$

Mixtures (3)

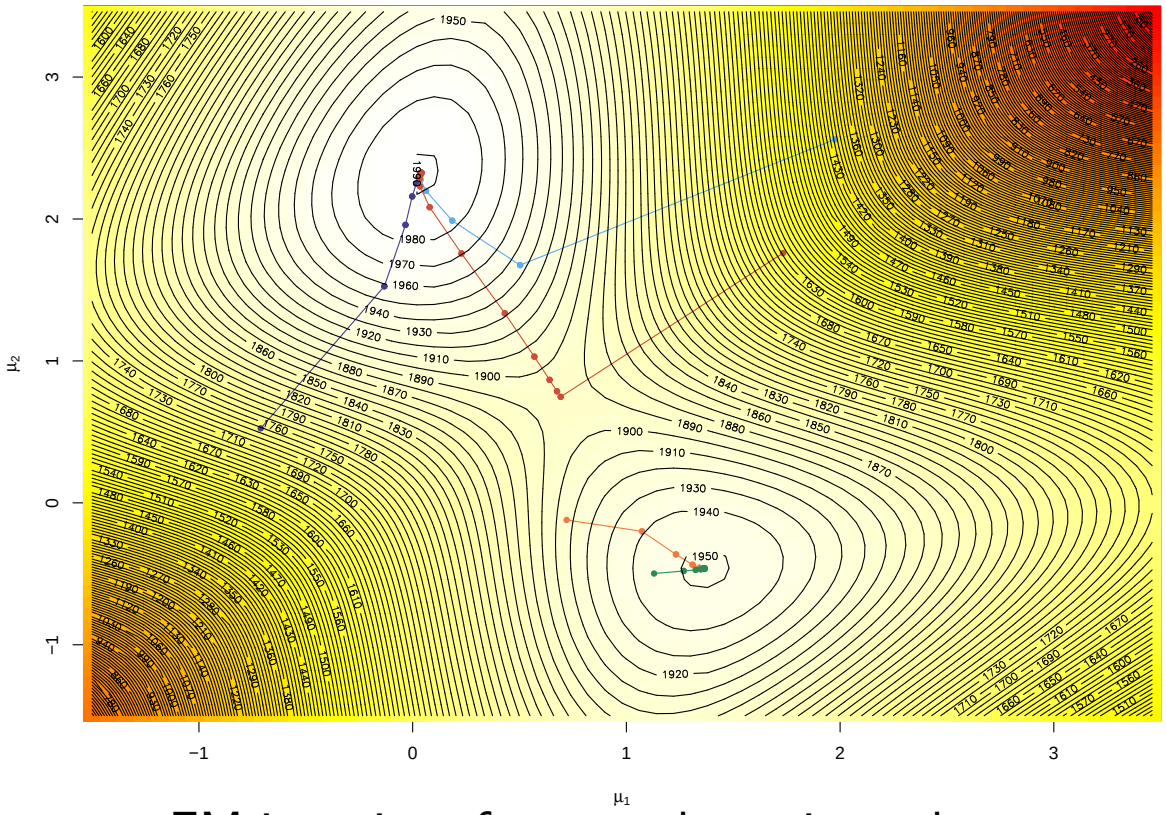
Hence $\hat{\theta}_{(j+1)}$ given by

$$\begin{pmatrix} \sum_{i=1}^{n} \mathbb{E}\left[Z_i \left| \hat{\theta}_{(j)}, x_i \right.\right] x_i \Big/ \sum_{i=1}^{n} \mathbb{E}\left[Z_i | \hat{\theta}_{(j)}, x_i\right] \\ \\ \sum_{i=1}^{n} \mathbb{E}\left[(1-Z_i) \left| \hat{\theta}_{(j)}, x_i \right.\right] x_i \Big/ \sum_{i=1}^{n} \mathbb{E}\left[(1-Z_i) | \hat{\theta}_{(j)}, x_i\right] \end{pmatrix}$$

Conclusion

Step *(E)* in EM replaces missing data Z_i with their conditional expectation, given $\mathbf{x}$ (expectation that depend on $\hat{\theta}_{(m)}$).

Mixtures (3)



EM iterations for several starting values

Properties

EM algorithm such that

- it converges to local maximum or saddle-point
- it depends on the initial condition $\theta_{(0)}$
- it really really depends on the initial condition $\theta_{(0)}$
- it hence requires several initial values when likelihood multimodal

Simulated Annealing: Introduction

- This name is borrowed from Metallurgy:
- A metal manufactured by a slow decrease of temperature (*annealing*)
  - Is stronger than a metal manufactured by a fast decrease of temperature.
- The fundamental idea of simulated annealing methods
  - A change of scale, or red temperature
  - Allows for faster moves on the surface of the function $\mathfrak{h}$ to maximize.
  - Rescaling partially avoids the trapping attraction of local maxima.
- As T decreases toward 0, the values simulated from this distribution become concentrated in a narrower and narrower neighborhood of the local maxima of $\mathfrak{h}$

simulated annealing algorithm

- Simulation method proposed by Metropolis *et al.* (1953)
- Starting from θ_0, ζ is generated from

$$\zeta \sim \text{Uniform in a neighborhood of } \theta_0.$$

- The new value of θ is generated as

$$\theta_1 = \begin{cases} \zeta & \text{with probability } \rho = \exp(\Delta h/T) \wedge 1 \\ \theta_0 & \text{with probability } 1 - \rho, \end{cases}$$

  - $\Delta h = h(\zeta) - h(\theta_0)$
  - If $h(\zeta) \geq h(\theta_0)$, ζ is accepted
  - If $h(\zeta) < h(\theta_0)$, ζ may still be accepted
  - This allows escape from local maxima

Metropolis Algorithm - Comments

- Simulated annealing typically modifies the temperature T at each iteration
- It has the form

Remark

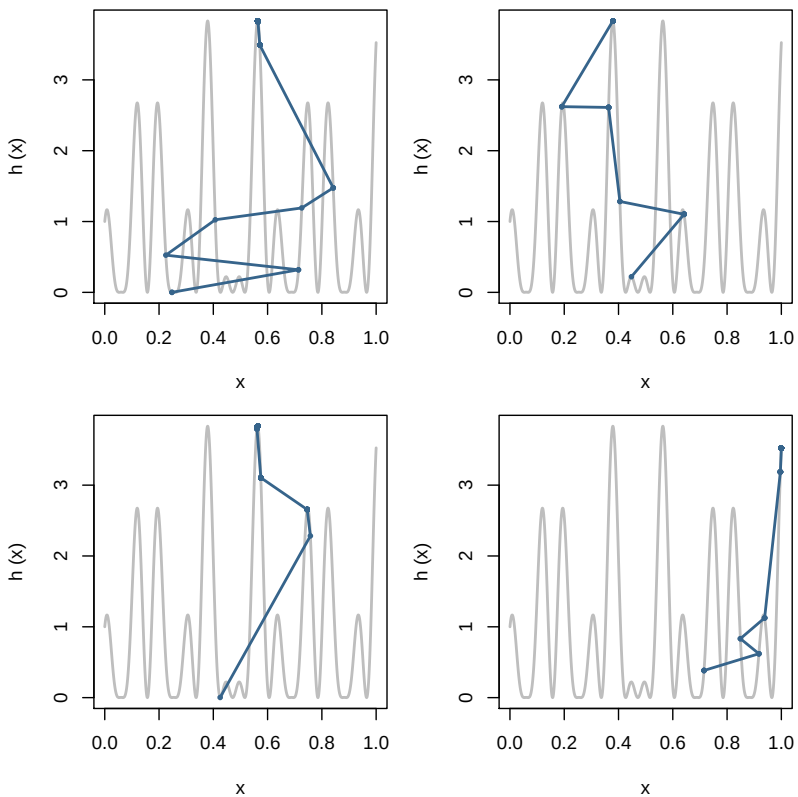
1. Simulate ζ from an instrumental distribution with density $g(|\zeta - \theta_i|)$;
2. Accept $\theta_{i+1} = \zeta$ with probability

$$\rho_i = \exp\{\Delta h_i / T_i\} \wedge 1;$$

   take $\theta_{i+1} = \theta_i$ otherwise.
3. Update T_i to T_{i+1}.

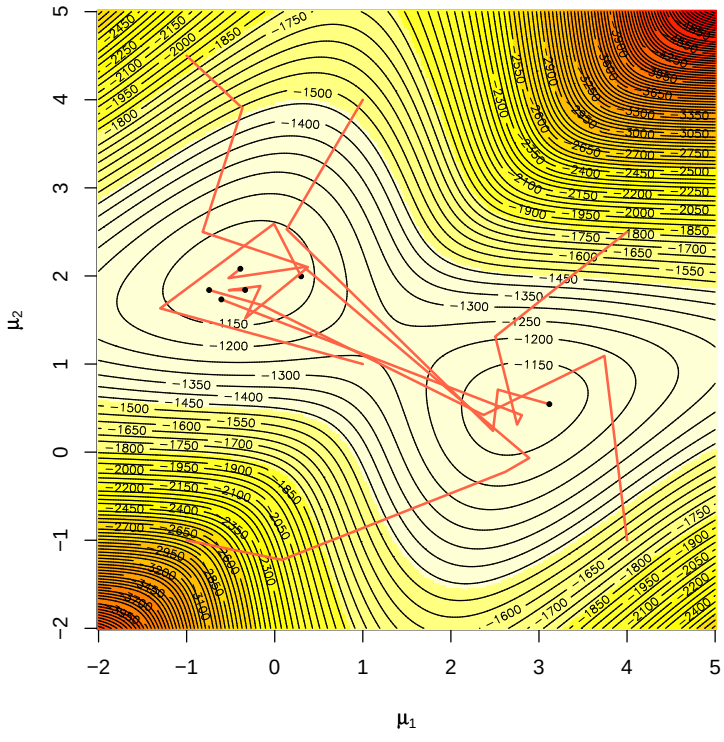
- All positive moves accepted
- As $T \downarrow 0$
  - Harder to accept downward moves
  - No big downward moves

Simple Example



- Trajectory: $T_i = \frac{1}{(1+i)^2}$
- Log trajectory also works
- Can Guarantee Finding Global Max
- R code

Normal Mixture



- Previous normal mixture
- Most sequences find max
- They visit both modes

Stochastic Approximation

- We now consider methods that work with the objective function $\mathfrak{h}$
  - Rather than being concerned with fast exploration of the domain Θ.
- Unfortunately, the use of those methods results in an additional level of error
  - Due to this approximation of $\mathfrak{h}$.
- But, the objective function in many statistical problems can be expressed as
  - $\mathfrak{h}(x) = \mathbb{E}[H(x, Z)]$
  - This is the setting of so-called missing-data models

optimizing Monte Carlo approximations

- If $h(x) = \mathbb{E}[H(x, Z)]$, a Monte Carlo approximation is

$$\hat{h}(x) = \frac{1}{m} \sum_{i=1}^{m} H(x, z_i),$$

  - Z_i's are generated from the conditional distribution $f(z|x)$.

- This approximation yields a convergent estimator of $h(x)$ for every value of x

  - This is a *pointwise convergent estimator*
  - Its use in optimization setups is not recommended
  - Changing sample of Z_i's $\Rightarrow$ unstable sequence of evaluations
  - And a rather noisy approximation to $\arg\max h(x)$

Bayesian Probit

Example: Bayesian analysis of a simple probit model

- $Y \in \{0, 1\}$ has a distribution depending on a covariate X:

$$P_\theta(Y = 1|X = x) = 1 - P_\theta(Y = 0|X = x) = \Phi(\theta_0 + \theta_1 x)\,,$$

  - Illustrate with Pima.tr dataset, Y= diabetes indicator, X=BMI

- Typically infer from the marginal posterior

$$\arg\max_{\theta_0} \int \prod_{i=1} \Phi(\theta_0+\theta_1 x_n)^{y_i} \Phi(-\theta_0-\theta_1 x_n)^{1-y_i}\, d\theta_1 = \arg\max_{\theta_0} h(\theta_0)$$

  - For a flat prior on θ and a sample $(x_1, \ldots, x_n)$.

Bayesian Probit – Importance Sampling

- No analytic expression for h
- The conditional distribution of θ_1 given θ_0 is also nonstandard
  - Use importance sampling with a t distribution with 5 df
  - Take $\mu = 0.1$ and $\sigma = 0.03$ (MLEs)
- Importance Sampling Approximation

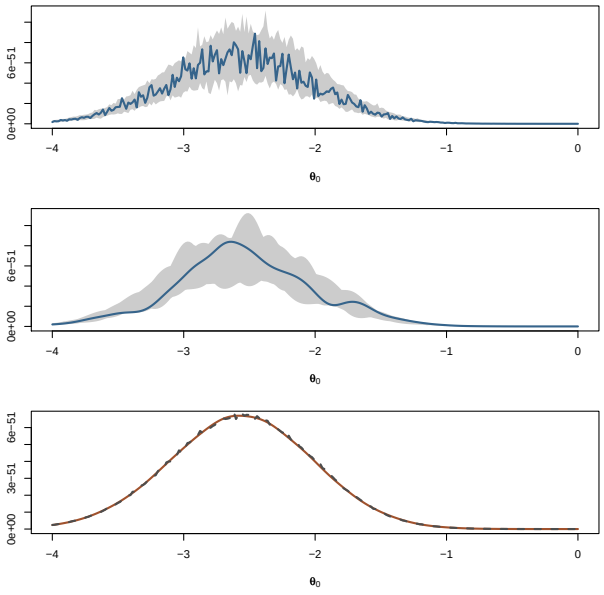
$$\widehat{h}_0(\theta_0) = \frac{1}{M} \sum_{m=1}^{M} \prod_{i=1} \Phi(\theta_0 + \theta_1^m x_n)^{y_i} \Phi(-\theta_0 - \theta_1^m x_n)^{1-y_i} t_5(\theta_1^m; \mu, \sigma)^{-}$$

Importance Sampling Evaluation

- Plotting this approximation of h with t samples simulated for each value of θ_0
  - The maximization of the represented $\widehat{h}$ function is not to be trusted as an approximation to the maximization of h.
- But, if we use the same t sample for all values of θ_0
  - We obtain a much smoother function
- We use importance sampling based on a *single* sample of Z_i's
  - Simulated from an importance function $g(z)$ for *all* values of x
  - Estimate h with

$$\widehat{h}_m(x) = \frac{1}{m} \sum_{i=1}^{m} \frac{f(z_i|x)}{g(z_i)} H(x, z_i).$$

Importance Sampling Likelihood Representation



- Top: 100 runs, different samples
- Middle: 100 runs, same sample
- Bottom: averages over 100 runs

- The averages over 100 runs are the same - but we will not do 100 runs
- R code: Run pimax(25) from mcsm

Comments

- This approach is not absolutely fool-proof
  - The precision of $\widehat{h}_m(x)$ has no reason to be independent of x
  - The number m of simulations has to reflect the most varying case.

- As in every importance sampling experiment
  - The choice of the candidate g is influential
  - In obtaining a good (or a disastrous) approximation of $h(x)$.

- Checking for the finite variance of the ratio $f(z_i|x)H(x,z_i)/g(z_i)$
  - Is a minimal requirement in the choice of g

Missing-Data models and demarginalization

- Missing data models are special cases of the representation $h(x) = \mathbb{E}[H(x, Z)]$
- These are models where the density of the observations can be expressed as

$$g(x|\theta) = \int_{\mathcal{Z}} f(x, z|\theta)\, dz\,.$$

- This representation occurs in many statistical settings
  - Censoring models and mixtures
  - Latent variable models (tobit, probit, arch, stochastic volatility, etc.)
  - Genetics: Missing SNP calls

Mixture Model

Example: Normal mixture model as a missing-data model

- Start with a sample $(x_1, \ldots, x_n)$
- Introduce a vector $(z_1, \ldots, z_n) \in \{1, 2\}^n$ such that

$$P_\theta(Z_i = 1) = 1 - P_\theta(Z_i = 2) = 1/4\,, \quad X_i | Z_i = z \sim \mathcal{N}(\mu_z, 1)\,,$$

- The (observed) likelihood is then obtained as $\mathbb{E}[H(\mathbf{x}, \mathbf{Z})]$ for

$$H(\mathbf{x}, \mathbf{z}) \propto \prod_{i;\, z_i = 1} \frac{1}{4} \exp\left\{-(x_i - \mu_1)^2/2\right\} \prod_{i;\, z_i = 2} \frac{3}{4} \exp\left\{-(x_i - \mu_2)^2/2\right\}$$

- We recover the mixture model

$$\frac{1}{4}\mathcal{N}(\mu_1, 1) + \frac{3}{4}\mathcal{N}(\mu_2, 1)$$

  - As the marginal distribution of X_i.

Censored–Data Likelihood

Example: Censored–data likelihood

- Censored data may come from experiments
  - Where some potential observations are replaced with a lower bound
  - Because they take too long to observe.

- Suppose that we observe $Y_1, \ldots, Y_m$, iid, from $f(y - \theta)$
  - And the $(n - m)$ remaining $(Y_{m+1}, \ldots, Y_n)$ are censored at the threshold a.
- The corresponding likelihood function is

$$L(\theta|\mathbf{y}) = [1 - F(a - \theta)]^{n-m} \prod_{i=1}^{m} f(y_i - \theta),$$

  - F is the cdf associated with f

Recovering the observed data likelihood

- If we had observed the last $n - m$ values
  - Say $\boldsymbol{z} = (z_{m+1}, \ldots, z_n)$, with $z_i \geq a$ $(i = m+1, \ldots, n)$,
  - We could have constructed the (complete data) likelihood

$$L^c(\theta|\boldsymbol{y}, \boldsymbol{z}) = \prod_{i=1}^{m} f(y_i - \theta) \prod_{i=m+1}^{n} f(z_i - \theta)\,.$$

- Note that

$$L(\theta|\boldsymbol{y}) = \mathbb{E}[L^c(\theta|\boldsymbol{y}, \boldsymbol{Z})] = \int_{\mathcal{Z}} L^c(\theta|\boldsymbol{y}, \boldsymbol{z}) f(\boldsymbol{z}|\boldsymbol{y}, \theta)\, d\boldsymbol{z},$$

  - Where $f(\boldsymbol{z}|\boldsymbol{y}, \theta)$ is the density of the missing data
  - Conditional on the observed data
  - The product of the $f(z_i - \theta)/[1 - F(a - \theta)]$'s
  - $f(z - \theta)$ restricted to $(a, +\infty)$.

Comments

- When we have the relationship

$$g(x|\theta) = \int_{\mathcal{Z}} f(x, z|\theta)\, dz\,.$$

  - Z merely serves to simplify calculations
  - it does not necessarily have a specific meaning
- We have the complete-data likelihood $L^c(\theta|\mathbf{x}, \mathbf{z})) = f(\mathbf{x}, \mathbf{z}|\theta)$
  - The likelihood we would obtain
  - Were we to observe $(\mathbf{x}, \mathbf{z})$,the complete data

- REMEMBER:

$$g(x|\theta) = \int_{\mathcal{Z}} f(x, z|\theta)\, dz\,.$$