

# Computational statistics

## Chapter 4: Classical simulation of probability distributions

Thierry Denœux  
Université de technologie de Compiègne

January-March 2024



# Overview

## 1 Introduction

## 2 Exact simulation

- Generating from standard parametric families
- Probability integral transform
- Rejection Sampling

## 3 Sampling Importance Resampling



# Purpose of this chapter

- This chapter addresses the simulation of **random draws**  $X_1, \dots, X_n$  from a **target distribution**  $f$ .
- The most frequent use of such draws is to estimate the **expectation** of a function of a random variable, say  $\mathbb{E}\{h(X)\}$ . For instance:  $\mathbb{E}\{X^k\}$ ,  $\mathbb{P}(X \in A) = \mathbb{E}\{I(X \in A)\}$ , etc.
- Example of applications:
  - ▶ E-step in the EM algorithm (“Monte-carlo EM”)
  - ▶ Calculation of some likelihood functions (“simulated likelihood”)
  - ▶ In Bayesian analyses, approximation of posterior moments, posterior probabilities, credible intervals, etc.
  - ▶ Estimation of risk, power of tests, etc.
  - ▶ etc.



# Monte Carlo integration

- Let  $f$  denote the density of  $X$ , and  $\mu$  denote the expectation of  $h(X)$  with respect to  $f$ .
- When an i.i.d. random sample  $X_1, \dots, X_n$  is obtained from  $f$ , we can approximate  $\mu$  by a sample average:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n h(X_i) \rightarrow \int h(x)f(x)dx = \mu$$

as  $n \rightarrow \infty$ , by the strong law of large numbers.



# Error estimation

- Further, let  $\sigma^2 = \mathbb{E}\{(h(X) - \mu)^2\}$  be the variance of  $h(X)$ , assuming that this quantity exists.
- The Monte Carlo approach can be used to estimate  $\sigma^2$  by

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n [h(X_i) - \hat{\mu}]^2 \quad (1)$$

- The Monte Carlo or simulation standard error (sse) of  $\hat{\mu}$  is  $\sigma/\sqrt{n}$ . It can be estimated by  $\hat{\sigma}/\sqrt{n}$ .
- When  $\sigma^2$  exists, the central limit theorem implies that  $\hat{\mu}$  has an approximate normal distribution for large  $n$ , so we get the following approximate confidence bounds for  $\mu$  with confidence level  $1 - \alpha$ :

$$\hat{\mu} \pm u_{1-\alpha/2} \frac{\hat{\sigma}}{\sqrt{n}}$$



# Non standard distributions

- Problem: how to generate draws from a **target distribution  $f$** ?
- When the target distribution comes from a standard parametric family, abundant software exists to easily generate random deviates.
- We focus on what should be done when **the target density is not one easily sampled using the software**.
- For example, nearly all Bayesian posterior distributions are not members of standard parametric families. Posteriors obtained when using conjugate priors in exponential families are exceptions.



# Difficulties

- There can be additional difficulties beyond the absence of an obvious method to sample  $f$ . In many cases – especially in Bayesian analyses – the target density may be **known only up to a multiplicative proportionality constant**. In such cases,  $f$  cannot be sampled and can only be evaluated up to that constant. Fortunately, there are a variety of simulation approaches that still work in this setting.
- Finally, it may be possible to evaluate  $f$ , but **computationally expensive**. If each computation of  $f(x)$  requires an optimization, an integration, or other time-consuming computations, we may seek simulation strategies that avoid direct evaluation of  $f$  as much as possible.
- Simulation methods can be categorized by whether they are **exact** or **approximate**.



# Overview

## 1 Introduction

## 2 Exact simulation

- Generating from standard parametric families
- Probability integral transform
- Rejection Sampling

## 3 Sampling Importance Resampling





# Overview

## 1 Introduction

## 2 Exact simulation

- Generating from standard parametric families
- Probability integral transform
- Rejection Sampling

## 3 Sampling Importance Resampling



# Standard uniform distribution

- At some level, all of code for simulation relies on the generation of **pseudorandom number generators (PRNGs)**, which are algorithms that can automatically create long runs of numbers that are statistically indistinguishable from **independent standard uniform variates**.
- The series of values generated by such algorithms is generally determined by a fixed number called a **seed**  $X_0$ . One of the most common PRNG is the **linear congruential generator**, which uses the recurrence

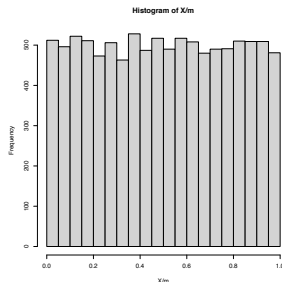
$$X_{n+1} = (aX_n + b) \bmod m$$

to generate numbers, where  $0 < a < m$ ,  $0 \leq b < m$  and  $m > 0$  are large integers, and mod is the remainder of the integer division. The maximum number of numbers the formula can produce is the modulus,  $m$ .



# Example in R

```
m<-2^32  
a<-1664525  
b<-1013904223  
N<-10000  
X<-rep(2^20,N)  
for(i in 2:N) X[i]<-(a*X[i-1]+b)%%m  
hist(X/m)
```



# Familiar distributions

Methods to draw from some standard parametric distributions. The methods may be special cases of a general method, or may be specific to the particular parametric family (ex: Student, Chi-square, etc.)

Distribution	Method
Uniform	See [195, 227, 383, 538, 539, 557]. For $X \sim \text{Unif}(a, b)$ ; draw $U \sim \text{Unif}(0, 1)$ ; then let $X = a + (b - a)U$ .
Normal( $\mu, \sigma^2$ ) and Lognormal( $\mu, \sigma^2$ )	Draw $U_1, U_2 \sim \text{i.i.d. Unif}(0, 1)$ ; then $X_1 = \mu + \sigma \sqrt{-2 \log U_1} \cos\{2\pi U_2\}$ and $X_2 = \mu + \sigma \sqrt{-2 \log U_1} \sin\{2\pi U_2\}$ are independent $N(\mu, \sigma^2)$ . If $X \sim N(\mu, \sigma^2)$ then $\exp(X) \sim \text{Lognormal}(\mu, \sigma^2)$ .
Multivariate $N(\mu, \Sigma)$	Generate standard multivariate normal vector, $\mathbf{Y}$ , coordinatewise; then $\mathbf{X} = \Sigma^{-1/2} \mathbf{Y} + \mu$ .
Cauchy( $\alpha, \beta$ )	Draw $U \sim \text{Unif}(0, 1)$ ; then $X = \alpha + \beta \tan\{\pi(U - \frac{1}{2})\}$ .
Exponential( $\lambda$ )	Draw $U \sim \text{Unif}(0, 1)$ ; then $X = -(\log U)/\lambda$ .
Poisson( $\lambda$ )	Draw $U_1, U_2, \dots \sim \text{i.i.d. Unif}(0, 1)$ ; then $X = j - 1$ , where $j$ is the lowest index for which $\prod_{i=1}^j U_i < e^{-\lambda}$ .
Gamma( $r, \lambda$ )	See Example 6.1, references, or for integer $r$ , $X = -(1/\lambda) \sum_{i=1}^r \log U_i$ for $U_1, \dots, U_r \sim \text{i.i.d. Unif}(0, 1)$ .
Chi-square ( $\text{df} = k$ )	Draw $Y_1, \dots, Y_k \sim \text{i.i.d. } N(0, 1)$ , then $X = \sum_{i=1}^k Y_i^2$ ; or draw $X \sim \text{Gamma}(k/2, \frac{1}{2})$ .
Student's $t$ ( $\text{df} = k$ ) and $F_{k,m}$ distribution	Draw $Y \sim N(0, 1)$ , $Z \sim \chi_k^2$ , $W \sim \chi_m^2$ independently, then $X = Y/\sqrt{Z/k}$ has the $t$ distribution and $F = (Z/k)/(W/m)$ has the $F$ distribution.
Beta( $a, b$ )	Draw $Y \sim \text{Gamma}(a, 1)$ and $Z \sim \text{Gamma}(b, 1)$ independently; then $X = Y/(Y + Z)$ .
Bernoulli( $p$ ) and Binomial( $n, p$ )	Draw $U \sim \text{Unif}(0, 1)$ ; then $X = 1_{\{U < p\}}$ is Bernoulli( $p$ ). The sum of $n$ independent Bernoulli( $p$ ) draws has a Binomial( $n, p$ ) distribution.
Negative Binomial( $r, p$ )	Draw $U_1, \dots, U_r \sim \text{i.i.d. Unif}(0, 1)$ ; then $X = \sum_{i=1}^r \lfloor (\log U_i) / \log(1 - p) \rfloor$ , and $\lfloor \cdot \rfloor$ means greatest integer.
Multinomial( $1, (p_1, \dots, p_k)$ )	Partition $[0, 1]$ into $k$ segments so the $i$ th segment has length $p_i$ . Draw $U \sim \text{Unif}(0, 1)$ ; then let $X$ equal the index of the segment into which $U$ falls. Tally such draws for Multinomial( $n, (p_1, \dots, p_k)$ ).
Dirichlet( $\alpha_1, \dots, \alpha_k$ )	Draw independent $Y_i \sim \text{Gamma}(\alpha_i, 1)$ for $i = 1, \dots, k$ ; then $\mathbf{X}^T = \left( Y_1 / \sum_{i=1}^k Y_i, \dots, Y_k / \sum_{i=1}^k Y_i \right)$ .

# Overview

## 1 Introduction

## 2 Exact simulation

- Generating from standard parametric families
- Probability integral transform
- Rejection Sampling

## 3 Sampling Importance Resampling



# Principle

- The methods for the Cauchy and exponential distributions in the previous table are justified by the **inverse cumulative distribution function** or **probability integral transform** approach, based on the following proposition:

## Proposition

*For any continuous **univariate** distribution function  $F$ , if  $U \sim \text{Unif}(0,1)$ , then  $X = F^{-1}(U)$  has a cumulative distribution function equal to  $F$ .*

Proof:  $\mathbb{P}(X \leq x) = \mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x)$ .

- If  $F^{-1}$  is available for the target density, then this strategy is probably the simplest option.



# Approximation

- If  $F^{-1}$  is not available but  $F$  is either available or easily approximated, then a crude approach can be built upon **linear interpolation**.
- Using a grid of  $x_1, \dots, x_m$  spanning the region of support of  $f$ , calculate or approximate  $u_i = F(x_i)$  at each grid point. Then, draw  $U \sim \text{Unif}(0, 1)$  and **linearly interpolate** between the two nearest grid points for which  $u_i \leq U \leq u_j$  according to

$$X = \frac{u_j - U}{u_j - u_i} x_i + \frac{U - u_i}{u_j - u_i} x_j.$$



# Discussion

- This approach is not exact, but its the degree of approximation is deterministic and can be reduced to any desired level by increasing  $m$  sufficiently.
- Compared to the alternatives, this simulation method is not appealing because
  - ▶ It requires a complete approximation to  $F$  regardless of the desired sample size
  - ▶ It does not generalize to multidimensional distributions
  - ▶ It is less efficient than other approaches.





# Overview

## 1 Introduction

## 2 Exact simulation

- Generating from standard parametric families
- Probability integral transform
- Rejection Sampling

## 3 Sampling Importance Resampling



# Basic idea

- If  $f(x)$  can be calculated, at least up to a proportionality constant, then we can use **rejection sampling** to obtain a random draw from exactly the target distribution.
- This strategy relies on sampling candidates from an **easier distribution** and then correcting the sampling probability through **random rejection** of some candidates.

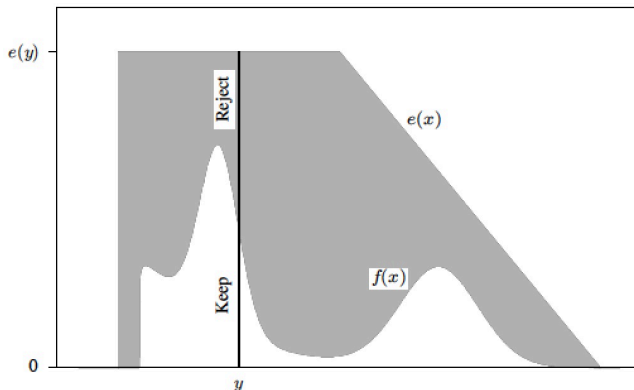


# Algorithm

- Let  $g$  denote another density from which we know how to sample and for which we can easily calculate  $g(x)$ .
- Let  $e(\cdot)$  denote an **envelope**, having the property  $e(x) = g(x)/\alpha \geq f(x)$  for all  $x$  for which  $f(x) > 0$ , for a given constant  $\alpha \leq 1$ .
- Rejection sampling proceeds as follows:
  - 1 Sample  $Y \sim g$ .
  - 2 Sample  $U \sim \text{Unif}(0, 1)$ .
  - 3 If  $U > f(Y)/e(Y)$ , reject  $Y$ ; in this case, do not record the value of  $Y$  as an element in the target random sample. Instead, return to step 1.
  - 4 Otherwise, keep the value of  $Y$ ; set  $X = Y$ , and consider  $X$  to be an element of the target random sample. Return to step 1 until you have accumulated a sample of the desired size.



# Rejection sampling



The shaded region above  $f$  and below  $e$  indicates the waste. The draw  $Y = y$  is very likely to be rejected when  $e(y)$  is far larger than  $f(y)$ . Envelopes that exceed  $f$  everywhere by at most a slim margin produce fewer wasted (i.e., rejected) draws and correspond to  $\alpha$  values near 1.



# Property

## Proposition

*The draws kept using this algorithm constitute an i.i.d. sample from the target density  $f$ ; there is no approximation involved.*



## Proof

$$\begin{aligned}
P[X \leq y] &= P\left[Y \leq y \mid U \leq \frac{f(Y)}{e(Y)}\right] \\
&= P\left[Y \leq y \text{ and } U \leq \frac{f(Y)}{e(Y)}\right] / P\left[U \leq \frac{f(Y)}{e(Y)}\right] \\
&= \int_{-\infty}^y \int_0^{f(z)/e(z)} du g(z) dz / \int_{-\infty}^{+\infty} \int_0^{f(z)/e(z)} du g(z) dz \\
&= \int_{-\infty}^y \frac{f(z)}{e(z)} g(z) dz / \int_{-\infty}^{+\infty} \frac{f(z)}{e(z)} g(z) dz \quad (2) \\
&= \frac{\int_{-\infty}^y \alpha f(z) dz}{\alpha} = \int_{-\infty}^y f(z) dz.
\end{aligned}$$



# Efficiency of the algorithm

- We have shown that

$$P \left[ U \leq \frac{f(Y)}{e(Y)} \right] = \alpha.$$

Consequently,  $\alpha$  can be interpreted as the expected proportion of candidates that are accepted.

- Hence  $\alpha$  is a **measure of the efficiency** of the algorithm.
- We may continue the rejection sampling procedure until it yields exactly the desired number of sampled points, but this requires a **random total number of iterations** that will depend on the proportion of rejections.



## Case where $f$ is known up to a proportionality constant

- Suppose now that the target distribution  $f$  is only known **up to a proportionality constant  $c$** . That is, suppose we are only able to compute easily  $q(x) = f(x)/c$ , where  $c$  is unknown.
- Such densities arise, for example, in **Bayesian inference** when  $f$  is a **posterior distribution** known to equal the product of the prior and the likelihood scaled by some normalizing constant.
- Fortunately, rejection sampling can be applied in such cases. We find an envelope  $e$  such that  $e(x) \geq q(x)$  for all  $x$  for which  $q(x) > 0$ .
- A draw  $Y = y$  is rejected when  $U > q(y)/e(y)$ . The sampling probability remains correct because the unknown constant  $c$  cancels out in the numerator and denominator of (2) when  $f$  is replaced by  $q$ . The proportion of kept draws is  $\alpha/c$ .





# Good rejection sampling envelopes

Good rejection sampling envelopes have three properties:

- 1 They are easily constructed to exceed the target everywhere
- 2 They are easy to sample
- 3 They generate few rejected draws.



# Sampling from a Bayesian posterior

- Suppose we want to sample from

$$f(\theta | x) \propto f(x | \theta)f(\theta) = L(\theta | x)f(\theta)$$

- Let  $q(\theta | x) = L(\theta | x)f(\theta)$ . We have

$$q(\theta | x) \leq L(\hat{\theta} | x)f(\theta) = e(\theta)$$

where  $\hat{\theta}$  is the MLE of  $\theta$ .

- The rejection sampling algorithm becomes:

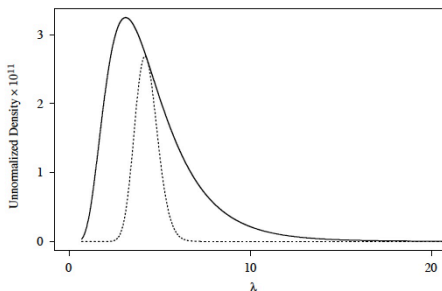
- 1 Sample  $\theta_i \sim f(\theta)$  (the prior)
- 2 Sample  $U_i \sim \text{Unif}(0, 1)$
- 3 Keep  $\theta_i$  if

$$U_i < \frac{q(\theta_i | x)}{e(\theta_i)} = \frac{L(\theta_i | x)}{L(\hat{\theta} | x)}$$



## Example

- Suppose 10 independent observations (8, 3, 4, 3, 1, 7, 2, 6, 2, 7) are collected from the model  $X_i | \lambda \sim \mathcal{P}(\lambda)$ . A lognormal prior distribution for  $\lambda$  is assumed:  $\log \lambda \sim \mathcal{N}(\log 4, 0.52)$ . We have  $\hat{\lambda} = \bar{x} = 4.3$ .
- Unnormalized target  $q(\lambda | x)$  (dotted) and envelope  $e(\lambda)$  (solid):



- Although not efficient – only about 30% of candidate draws are kept – this approach is easy and exact.

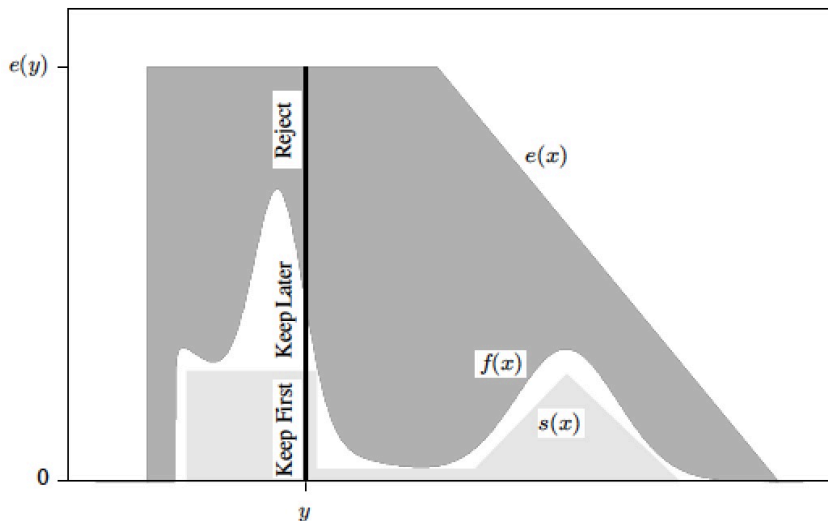


# Squeezed rejection sampling

- When evaluating  $f$  is computationally expensive, we can use sometimes a nonnegative **squeezing function**  $s$  that is easier to compute and such that  $s(x) \leq f(x)$  for all  $x$  such that  $f(x) > 0$ .
- The algorithm becomes
  - 1 Sample  $Y \sim g$ .
  - 2 Sample  $U \sim \text{Unif}(0, 1)$ .
  - 3 If  $U \leq s(Y)/e(Y)$ , keep  $Y$  and set  $X = Y$ .
  - 4 Else if  $U \leq f(Y)/e(Y)$ , keep  $Y$  and set  $X = Y$ .
  - 5 Otherwise, reject  $Y$  and return to step 1.



# Squeezed rejection sampling



# Overview

## 1 Introduction

## 2 Exact simulation

- Generating from standard parametric families
- Probability integral transform
- Rejection Sampling

## 3 Sampling Importance Resampling



# Need for approximations

- Although the methods described above have the appealing feature that they are exact, there are many cases when an **approximate method** is easier or perhaps the only feasible choice.
- Approximation is not a critical flaw as long as the degree of approximation can be controlled by user-specified parameters in the algorithms.
- Many approximate simulation methods are based to some extent on the **Sampling Importance Resampling (SIR)** principle.



# Basic idea

- The SIR algorithm simulates realizations approximately from some target distribution.
- SIR is based upon the notion of **importance sampling**.
- Briefly, importance sampling proceeds by drawing a sample from an **importance sampling function**,  $g$ . Informally, we will call  $g$  an **envelope**.
- Each point in the sample is weighted to correct the sampling probabilities so that the weighted sample can be related to a target density  $f$ .





# SIR algorithm

- Let  $X$  denote a random variable or vector with density  $f(x)$ , and let  $g$  denote the density corresponding to an envelope for the target density  $f$ , such that the support of  $g$  includes the entire support of  $f$  ( $\forall x, g(x) = 0 \Rightarrow f(x) = 0$ ).
- SIR algorithm:

- 1 Sample candidates  $Y_1, \dots, Y_m$  i.i.d. from  $g$ .
- 2 Calculate the **standardized importance weights**,  $w(Y_1), \dots, w(Y_m)$ , with

$$w(Y_i) = \frac{f(Y_i)/g(Y_i)}{\sum_{j=1}^m f(Y_j)/g(Y_j)} \quad (3)$$

- 3 Resample  $X_1, \dots, X_n$  from  $Y_1, \dots, Y_m$  with replacement with probabilities  $w(Y_1), \dots, w(Y_m)$ .
- Remark: when  $f = cq$  for some unknown proportionality constant  $c$ , the unknown  $c$  cancels in the numerator and denominator of (3).



# Property

## Proposition

*A random variable  $X$  drawn with the SIR algorithm has distribution that converges to  $f$  as  $m \rightarrow \infty$ .*



# Sketch of proof

- Let  $X$  be a r.v. drawn with the SIR algorithm. Define  $w^*(y) = f(y)/g(y)$ , let  $Y_1, \dots, Y_m$  i.i.d. from  $g$  and consider an event  $A$ .

$$\begin{aligned}\mathbb{P}(X \in A \mid Y_1, \dots, Y_m) &= \sum_{\{i: Y_i \in A\}} w(Y_i) \\ &= \sum_{i=1}^m I(Y_i \in A) w^*(Y_i) / \sum_{i=1}^m w^*(Y_i)\end{aligned}$$

- From the strong law of large numbers,

$$\frac{1}{m} \sum_{i=1}^m I(Y_i \in A) w^*(Y_i) \rightarrow \mathbb{E} \{I(Y \in A) w^*(Y)\} = \int_A w^*(y) g(y) dy$$

and

$$\frac{1}{m} \sum_{i=1}^m w^*(Y_i) \rightarrow \mathbb{E} \{w^*(Y)\} = \int w^*(y) g(y) dy = 1$$



# Sketch of proof (continued)

- Consequently,

$$\mathbb{P}(X \in A \mid Y_1, \dots, Y_m) \rightarrow \int_A w^*(y)g(y)dy = \int_A f(y)dy$$

- Finally, we have

$$\mathbb{P}(X \in A) = \mathbb{E} \{ \mathbb{P}(X \in A \mid Y_1, \dots, Y_m) \} \rightarrow \int_A f(y)dy$$

(by Lebesgue's dominated convergence theorem)



# Sample sizes

- When conducting SIR, it is important to consider the relative sizes of the initial sample ( $m$ ) and the resample ( $n$ ).
- In principle, we require  $n/m \rightarrow 0$  for distributional convergence of the sample. In the context of asymptotic analysis of Monte Carlo estimates based on SIR, where  $n \rightarrow \infty$ , this condition means that  $m \rightarrow \infty$  even faster than  $n \rightarrow \infty$ .
- Rule of thumb: ensure  $m/n \geq 10$  so long as the resulting resample does not contain too many replicates of any initial draw.



# Envelope

- The SIR algorithm can be sensitive to the choice of  $g$ .
- First, the support of  $g$  must include the entire support.
- Further,  $g$  should have **heavier tails** than  $f$ , or more generally  $g$  should be chosen to ensure that  $f(x)/g(x)$  never grows too large.
- If  $g(x)$  is nearly zero anywhere where  $f(x)$  is positive, then a draw from this region will happen only extremely rarely, but when it does it will receive a huge weight. When this problem arises, one or a few standardized importance weights are enormous compared to the other weights, and the secondary sample consists nearly entirely of replicated values of one or a few initial draws.
- When the distribution of weights is found to be highly skewed, it is probably wiser to switch to a different envelope or a different sampling strategy altogether.



# Application to Bayesian inference

- Suppose that we seek a sample from the posterior distribution from a Bayesian analysis.
- Let  $f(\theta)$  denote the prior, and  $L(\theta | x)$  the likelihood, so the posterior is  $f(\theta | x) = c \cdot f(\theta)L(\theta | x)$  for some constant  $c$  that may be difficult to determine.
- If the prior does not seriously restrict the parameter region favored by the data via the likelihood function, then the prior can serve as a useful importance sampling function.
- Sample  $\theta_1, \dots, \theta_m$  i.i.d. from  $f(\theta)$ . Since the target density is the posterior, the  $i$ -th unstandardized weight equals  $c \cdot L(\theta_i | x)$ . Thus the SIR algorithm has a very simple form: Sample from the prior, weight by the likelihood, and resample.
- Remark: we do not need to know  $\hat{\theta}$ , in contrast with the rejection sampling method.

