Introduction to Bayesian computation (cont.)

Dr. Jarad Niemi

STAT 544 - Iowa State University

March 21, 2024

Outline

Bayesian computation

- Adaptive rejection sampling
- Importance sampling

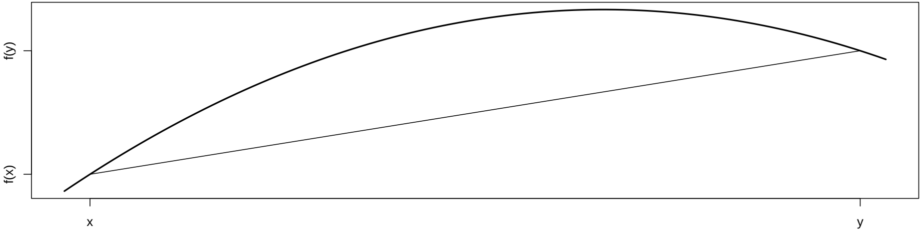
Adaptive rejection sampling

Definition

A function is concave if

$$f((1-t)x + t\,y) \geq (1-t)f(x) + t\,f(y)$$

for any $0 \leq t \leq 1$.



Log-concavity

Definition

A function $f(x)$ is log-concave if $\log\, f(x)$ is concave.

Lemma

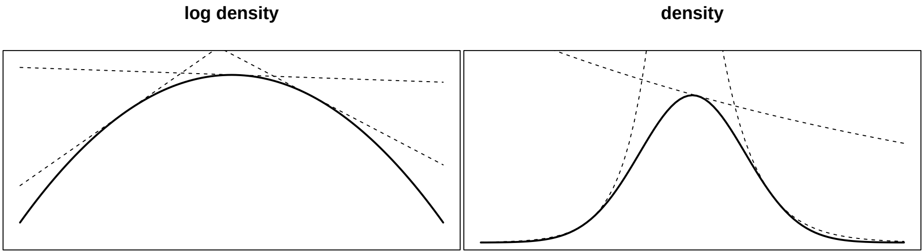
A function is log-concave if and only if $(\log\, f(x))'' \le 0\,\forall\, x$.

For example, $X \sim N(0,1)$ has log-concave density since

$$\frac{d^2}{dx^2}\log\, e^{-x^2/2} = \frac{d^2}{dx^2}\frac{-x^2}{2} = \frac{d}{dx} - x = -1.$$

Adaptive rejection sampling

Adaptive rejection sampling can be used for distributions with log-concave densities. It builds a piecewise linear envelope to the log density by evaluating the log function and its derivative at a set of locations and constructing tangent lines, e.g.



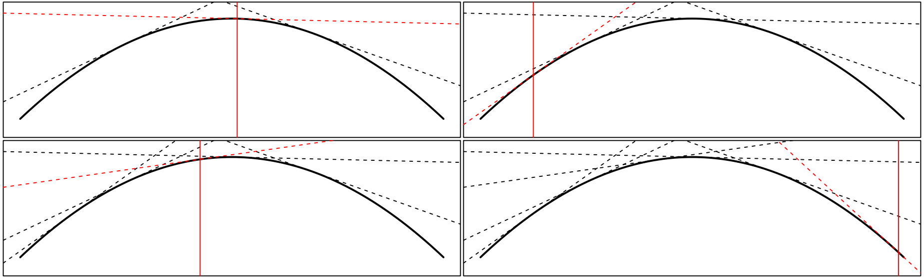
Adaptive rejection sampling

Pseudo-algorithm for adaptive rejection sampling:

1. Choose starting locations θ, call the set Θ
2. Construct piece-wise linear envelope $\log g(\theta)$ to the log-density
   a. Calculate $\log q(\theta|y)$ and $(\log q(\theta|y))'$.
   b. Find line intersections
3. Sample a proposed value θ^* from the envelope $g(\theta)$
   a. Sample an interval
   b. Sample a truncated (and possibly negative of an) exponential r.v.
4. Perform rejection sampling
   a. Sample $u \sim Unif(0, 1)$
   b. Accept if $u \leq q(\theta^*|y)/g(\theta^*)$.
5. If rejected, add θ^* to Θ and return to 2.

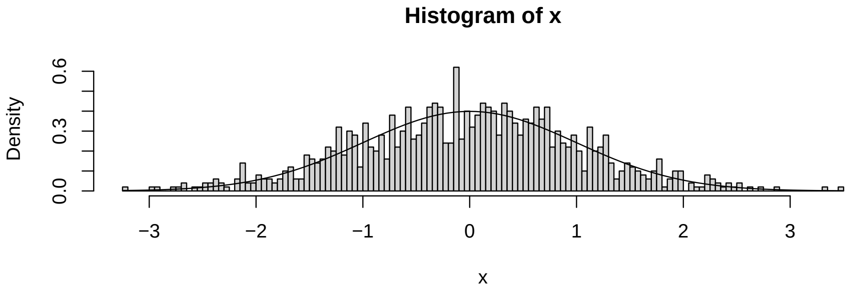
Updating the envelope

As values are proposed and rejected, the envelope gets updated:



Adaptive rejection sampling in R

```
library(ars)
x = ars(n=1000, function(x) -x^2/2, function(x) -x)
hist(x, prob=T, 100)
curve(dnorm, type='l', add=T)
```



Adaptive rejection sampling summary

- Can be used with log-concave densities
- Makes rejection sampling efficient by updating the envelope

There is a vast literature on adaptive rejection sampling. To improve upon the basic idea presented here you can

- include a lower bound
- avoid calculating derivatives
- incorporate a Metropolis step to deal with non-log-concave densities

Importance sampling

Notice that

$$E[h(\theta)|y] = \int h(\theta)p(\theta|y)d\theta = \int h(\theta)\frac{p(\theta|y)}{g(\theta)}g(\theta)d\theta$$

where $g(\theta)$ is a proposal distribution, so that we approximate the expectation via

$$E[h(\theta)|y] \approx \frac{1}{S}\sum_{s=1}^{S} w\left(\theta^{(s)}\right) h\left(\theta^{(s)}\right)$$

where $\theta^{(s)} \stackrel{iid}{\sim} g(\theta)$ and

$$w\left(\theta^{(s)}\right) = \frac{p\left(\theta^{(s)}\middle| y\right)}{g(\theta^{(s)})}$$

is known as the importance weight.

Importance sampling

If the target distribution is known only up to a proportionality constant, then

$$E[h(\theta)|y] = \frac{\int h(\theta)q(\theta|y)d\theta}{\int q(\theta|y)d\theta} = \frac{\int h(\theta)\frac{q(\theta|y)}{g(\theta)}g(\theta)d\theta}{\int \frac{q(\theta|y)}{g(\theta)}g(\theta)d\theta}$$

where $g(\theta)$ is a proposal distribution, so that we approximate the expectation via

$$E[h(\theta)|y] \approx \frac{\frac{1}{S}\sum_{s=1}^{S} w\left(\theta^{(s)}\right) h\left(\theta^{(s)}\right)}{\frac{1}{S}\sum_{s=1}^{S} w\left(\theta^{(s)}\right)} = \sum_{s=1}^{S} \tilde{w}\left(\theta^{(s)}\right) h\left(\theta^{(s)}\right)$$

where $\theta^{(s)} \stackrel{iid}{\sim} g(\theta)$ and

$$\tilde{w}\left(\theta^{(s)}\right) = \frac{w\left(\theta^{(s)}\right)}{\sum_{j=1}^{S} w\left(\theta^{(j)}\right)}$$

is the normalized importance weight.

Example: Normal-Cauchy model

If $Y \sim N(\theta, 1)$ and $\theta \sim Ca(0, 1)$, then

$$p(\theta|y) \propto e^{-(y-\theta)^2/2} \frac{1}{(1+\theta^2)}$$

for all θ.

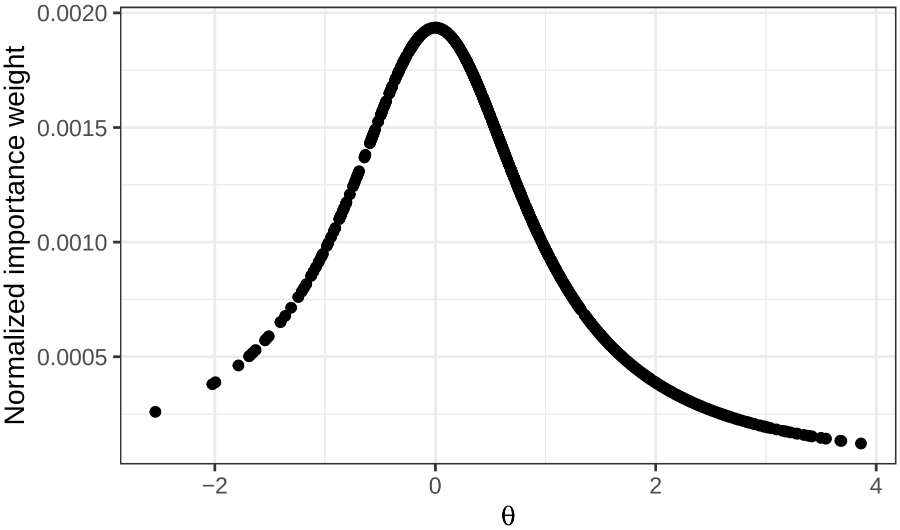
If we choose a $N(y, 1)$ proposal, we have

$$g(\theta) = \frac{1}{\sqrt{2\pi}} e^{-(\theta-y)^2/2}$$

with

$$w(\theta) = \frac{q(\theta|y)}{g(\theta)} = \frac{\sqrt{2\pi}}{(1+\theta^2)}$$

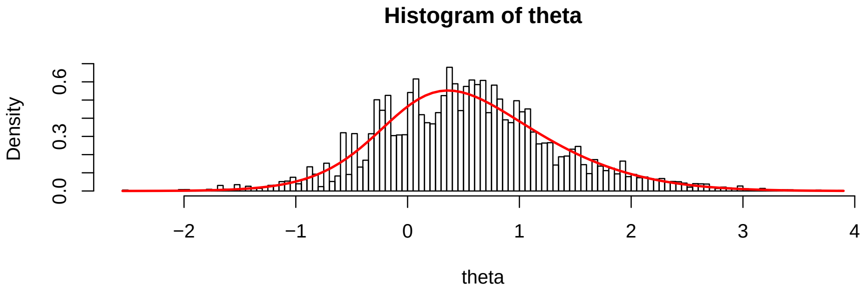
Normalized importance weights



```r
library(weights)
theta  <- d$theta; weight <- d$weight
sum(weight*theta/sum(weight)) # Estimate mean
```

```
[1] 0.5504221
```

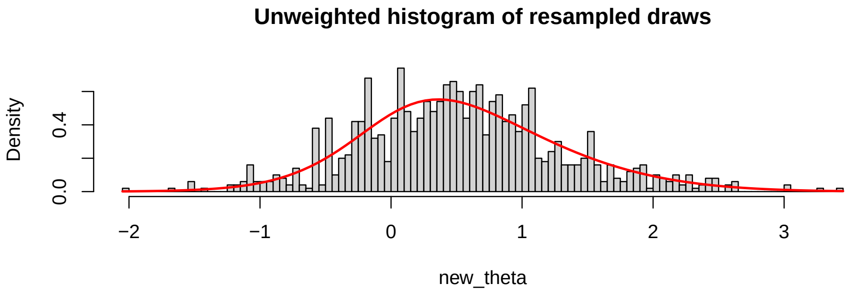
```r
wtd.hist(theta, 100, prob=TRUE, weight=weight)
curve(q(x,y)/py(y), add=TRUE, col="red", lwd=2)
```



Resampling

If an unweighted sample is desired, sample $\theta^{(s)}$ with replacement with probability equal to the normalized weights, $\tilde{w}\left(\theta^{(s)}\right)$.

```
# resampling
new_theta <- sample(theta, replace=TRUE, prob = weight) # internally normalized
hist(new_theta, 100, prob = TRUE, main = "Unweighted histogram of resampled draws"); curve(q(x,y)/py(y), add = TRUE, col="red", lwd=2)
```



Heavy-tailed proposals

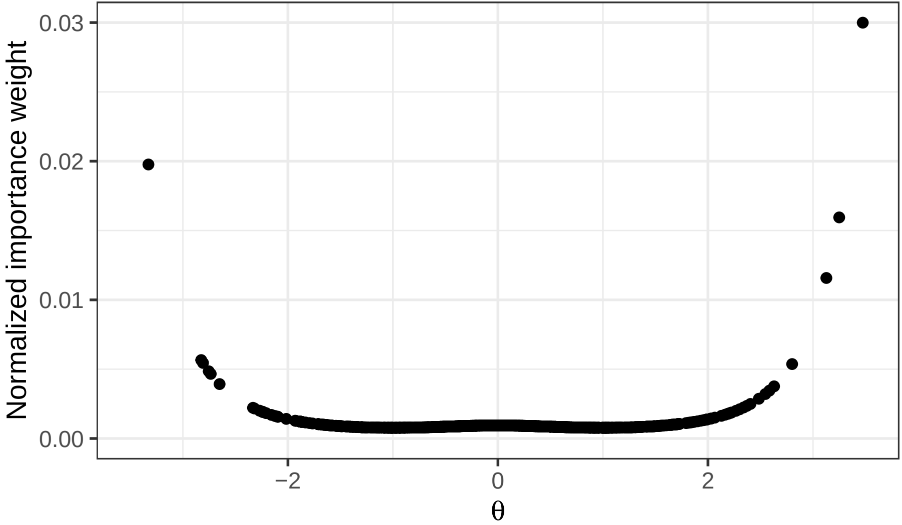
Although any proposal can be used for importance sampling, proposals with tails as heavy as the target will be efficient and have a CLT.

For example, suppose our target is a standard Cauchy and our proposal is a standard normal, the weights are

$$w\left(\theta^{(s)}\right) = \frac{p\left(\theta^{(s)}\middle| y\right)}{g(\theta^{(s)})} = \frac{\frac{1}{\pi(1+\theta^2)}}{\frac{1}{\sqrt{2\pi}}e^{-\theta^2/2}}$$

For $\theta^{(s)} \stackrel{iid}{\sim} N(0,1)$, the weights for the largest $|\theta^{(s)}|$ will dominate the others.

Importance weights for proposal with thin tails



Effective sample size

We can get a measure of how efficient the sample is by computing the effective sample size (ESS), i.e. how many independent unweighted draws do we effectively have:

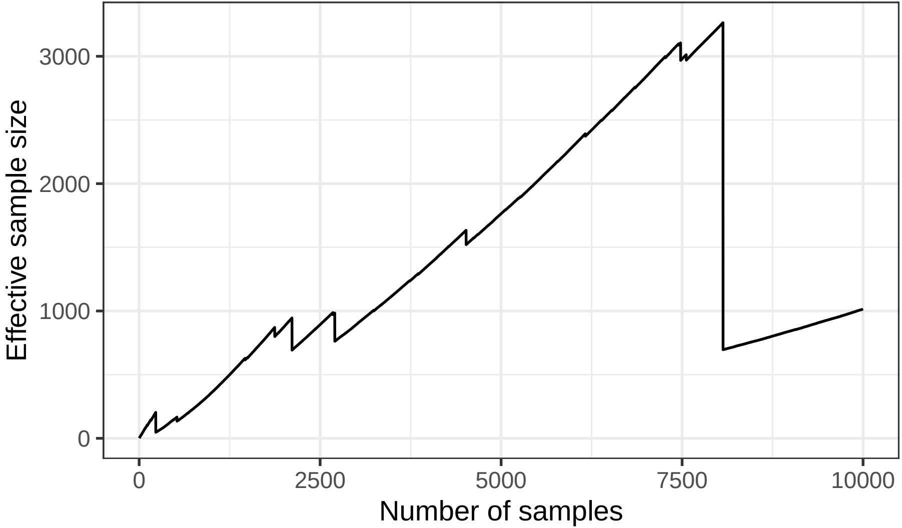
$$ESS = \frac{1}{\sum_{s=1}^{S} \left(\tilde{w}\left(\theta^{(s)}\right)\right)^2}$$

```
weight <- d$unweight          # Unnormalized weight
(n    <- length(d$weight))    # Number of samples

[1] 1000

(ess <- 1/sum(d$weight^2))    # Effective sample size

[1] 371.432

ess/n                         # Effective sample proportion

[1] 0.371432
```

Effective sample size

```
set.seed(5)
theta           <- rnorm(1e4)
lweight         <- dcauchy(theta,log=TRUE) - dnorm(theta,log=TRUE)
cumulative_ess <- length(lweight)

for (i in 1:length(lweight)) {
  lw = lweight[1:i]
  w = exp(lw-max(lw))
  w = w/sum(w)
  cumulative_ess[i] = 1/sum(w^2)
}
```

ESS - Light tail proposal



Practical Monte Carlo

As a practical matter, we typically obtain a single collection of samples, say $\theta^{(s)}$ for $s = 1, \ldots, S$ and we want to address many scientific questions.

For example,

- $E[\theta|y]$
- Equal-tail 95% CI for θ
- other functions of θ

So how large should S be? Large enough so the Monte Carlo error on the worst estimated quantity is sufficiently small.

Practical Monte Carlo - Monte Carlo error

Calculate the Monte Carlo error.

```
# Normal distribution
theta <- rnorm(1e3)

mcmcse::mcse(theta)         # expectation

$est
[1] -0.02732462

$se
[1] 0.03144569

mcmcse::mcse.q(theta, .025) # quantile

$est
[1] -1.877009

$se
[1] 0.06260711

$nsim
[1] 1000

# mcmcse::mcse.q(theta, .975)
```

Monte Carlo Error as a Function of Sample Size