Chapter 6: Rejection Sampling and Importance Sampling

ST4234: Bayesian Statistics

Semester 2, AY 2019/2020

Department of Statistics and Applied Probability

National University of Singapore

LI Cheng

stalic@nus.edu.sg

Introduction

- This chapter corresponds to Sections 5.8-5.10 in Jim Albert's book, and part of Chapter 3 in Martin A. Tanner's book.
- We continue to investigate how to approximate the Bayesian posterior distribution. This time, we focus on Monte Carlo methods.
- Two methods will be introduced in this chapter: rejection sampling and importance sampling.
- The output of rejection sampling is a sample from the target distribution.
- The output of importance sampling is an estimate of a target integral. An induced method, called sampling importance resampling (SIR), can be used to generate a sample from the target distribution.

Outline

Rejection sampling

Importance sampling

Sampling importance resampling

6.1 Rejection sampling

- Rejection sampling is a general-purpose algorithm for simulating random draws from a given probability distribution. In this method, instead of trying to approximate the posterior, we try to "blanket" it.

- Suppose we wish to generate independent samples from a posterior density

$$p(\theta|\boldsymbol{y}) = \frac{f(\theta)}{C}$$

  For notational convenience, we suppress dependence of f on data y. C is the unknown normalizing constant.

- The first step in rejection sampling is to find another probability density $g(\theta)$, called the envelope function satisfying these conditions:
  - It is easy to simulate draws from g.
  - The density g resembles $p(\theta|\boldsymbol{y})$ in terms of location and spread.
  - $\exists$ a finite constant $M > 0$ such that $f(\theta) \leq M\, g(\theta)$ for all $\theta \in \Theta$.

6.1 Rejection sampling

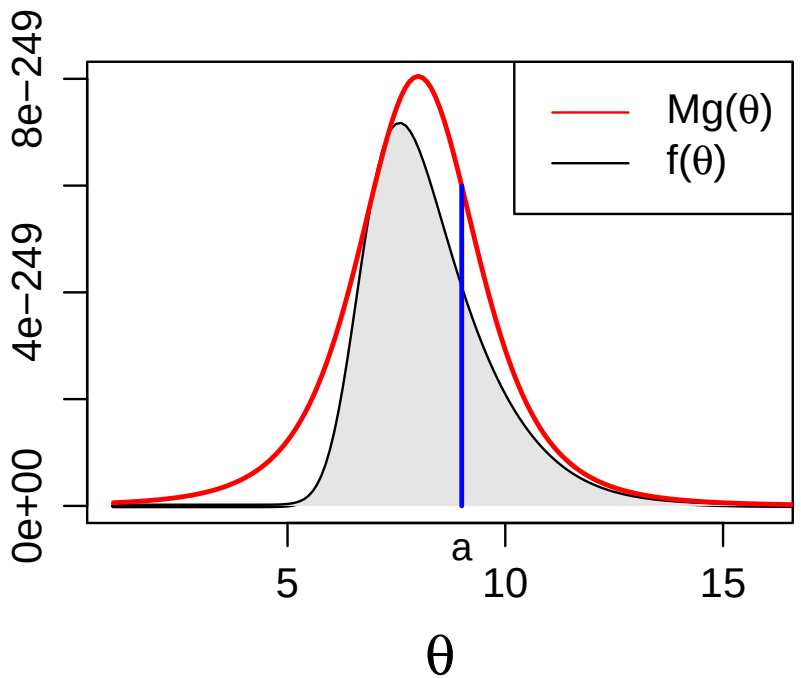
- Suppose we are able to find a density g with these properties. Then we can obtain draws from $p(\theta|\boldsymbol{y})$ using the following algorithm:

Rejection sampling algorithm

1. Generate $\theta^{(s)} \sim g(\theta)$.
2. Generate $U \sim$ Uniform[0,1].
3. If $U < \dfrac{f(\theta^{(s)})}{M\, g(\theta^{(s)})}$, accept $\theta^{(s)}$; otherwise reject $\theta^{(s)}$.
4. Repeat steps 1–3 until the desired sample $\{\theta^{(s)} : \ s = 1, \ldots, S\}$ is obtained. The members of this sample will be random variables from $p(\theta|\boldsymbol{y})$.

6.1 Rejection sampling

- The figure below is an illustration of rejection sampling. The top red curve is an approximation $Mg(\theta)$ and the bottom black curve is the target $f(\theta)$. As required $f(\theta) \leq Mg(\theta)$ for all θ.
- The vertical blue line indicates a single random draw "a" from $g(\theta)$. The probability that "a" is accepted is the ratio of the height of the lower curve to the height of the higher curve at "a".



6.1 Rejection sampling

Why does reject sampling work?

- Consider a large sample of points generated from $g(\theta)$. An appropriately scaled histogram of these points would have roughly the same shape as the curve labeled $Mg(\theta)$ in the previous figure.
- Now consider the histogram bar centered at "a". The rejection step has the effect of slicing off the top portion of the bar (i.e. the portion between the two curves), since only those points having values below the lower curve are retained.
- This is true for every potential value of "a" along the horizontal axis, so a histogram of the accepted values would mimic the shape of the lower curve, which would be proportional to the posterior distribution $p(\theta|\boldsymbol{y})$.

6.1 Rejection sampling

Acceptance probability

- Recall that $p(\theta|\boldsymbol{y}) = f(\theta)/C$.
- The unconditional acceptance probability (proportion of proposed samples which are accepted) is given by

$$\begin{aligned} P(\theta \text{ is accepted}) &= \int P(\theta \text{ is accepted}|\theta) g(\theta)\, \mathrm{d}\theta \\ &= \int P\left(U < \frac{f(\theta)}{Mg(\theta)} \;\middle|\; \theta\right) g(\theta)\, \mathrm{d}\theta \\ &= \int \frac{f(\theta)}{Mg(\theta)} g(\theta)\, \mathrm{d}\theta \\ &= \frac{1}{M} \int Cp(\theta|\boldsymbol{y})\, \mathrm{d}\theta = \frac{C}{M}. \end{aligned}$$

Therefore M should be chosen as small as possible, so as not to unnecessarily waste samples.

6.1 Rejection sampling

- It follows that

$$\begin{aligned} p(\theta|\theta \text{ is accepted}) &= \frac{P(\theta \text{ is accepted}|\theta)g(\theta)}{P(\theta \text{ is accepted})} \\ &= \frac{\frac{f(\theta)}{Mg(\theta)}g(\theta)}{C/M} \\ &= \frac{f(\theta)}{C} = p(\theta|\boldsymbol{y}). \end{aligned}$$

Therefore the distribution of a sample θ generated from $g(\theta)$, conditional on it being accepted is indeed $p(\theta|\boldsymbol{y})$.

6.1 Rejection sampling

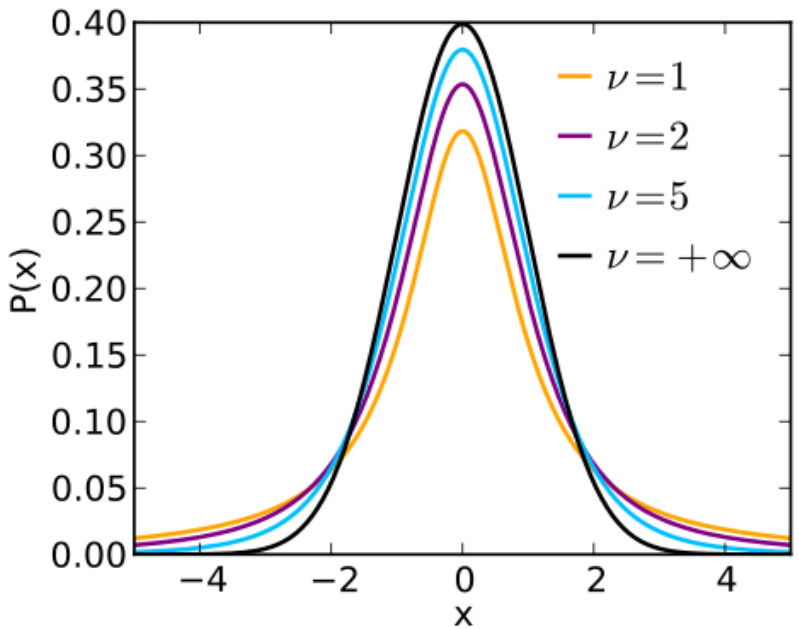
- Let us use rejection sampling to simulate draws of θ in the beta-binomial example.
- We wish to find a proposal density g of a simple functional form that, when multiplied by an appropriate constant M, covers the posterior density of interest.
- The probability of accepting a candidate draw is given by $\frac{f(\theta)}{Mg(\theta)}$. We can monitor the algorithm by computing the proportion of draws of g that are accepted; an efficient rejection sampling algorithm has a high acceptance rate.
- One choice for g would be the bivariate normal approximation to the posterior. Although this density resembles the posterior, the normal density has light tails and the ratio $\frac{f(\theta)}{g(\theta)}$ would likely not be bounded.

6.1 Rejection sampling

Proposal density

- A better choice for a covering density is the multivariate t distribution. The mean and scale matrix of the multivariate t can be chosen to match the posterior density. A small number of degrees of freedom can be used to give the density heavy tails so that one is more likely to find bounds for the ratio $\frac{f(\theta)}{g(\theta)}$.

Univariate t distribution, t_ν, for different values of ν.



- A random vector $\mathbf{X} = (X_1, \ldots, X_p)^\top$ is said to follow a multivariate t distribution with location $\boldsymbol{\mu}$, scale matrix $\boldsymbol{\Sigma}$ and degrees of freedom ν, denoted $\mathbf{X} \sim t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if the pdf of $\mathbf{X}$ is

$$p(\mathbf{x}) = \frac{\Gamma\left(\frac{\nu+p}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)(\pi\nu)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}}\left[1 + \frac{(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}{\nu}\right]^{-\frac{\nu+p}{2}}.$$

The median and mode of X are μ, $\mathrm{E}(\mathbf{X}) = \mu$ if $\nu > 1$ and $\mathrm{Var}(\mathbf{X}) = \frac{\nu}{\nu-2}\boldsymbol{\Sigma}$ if $\nu > 2$.

6.1 Rejection sampling

- At the end of Chapter 5, we found a normal approximation to the posterior with mean $\hat{\theta}$ and covariance matrix $-[h''(\hat{\theta})]^{-1}$, where $\hat{\theta}$ is the posterior mode.
- Suppose we decide to use a multivariate t density with location $\hat{\theta}$, scale matrix $-2[h''(\hat{\theta})]^{-1}$ and 4 degrees of freedom.
- These choices are made to mimic the posterior density and ensure that the ratio $f(\theta)/g(\theta)$ is bounded from above.

6.1 Rejection sampling

Find bounding constant

- To set up the rejection algorithm, we need to find the bounding constant M such that

$$f(\theta) \leq M\, g(\theta)$$

  for all $\theta \in \Theta$.

- Equivalently, we want to find the constant $M' = \log M$ such that

$$\log f(\theta) - \log g(\theta) \leq M'$$

  for all $\theta \in \Theta$.

- We can find the maximum value of $\log f(\theta) - \log g(\theta)$ over all $\theta \in \Theta$ by using `optim`.

6.1 Rejection sampling

- We write a function named `diff` to compute the difference $\log f(\theta) - \log g(\theta)$. For the beta-binomial model, $\log f(\theta)$ is given by the function `h` written previously.
- The function `dmvt` in the `R` package `mvtnorm` computes the logarithm of the density of a multivariate t distribution, with location parameter `delta`, scale matrix `sigma`, and degrees of freedom `df`. Set `log=FALSE` to obtain the density (without taking log).

6.1 Rejection sampling

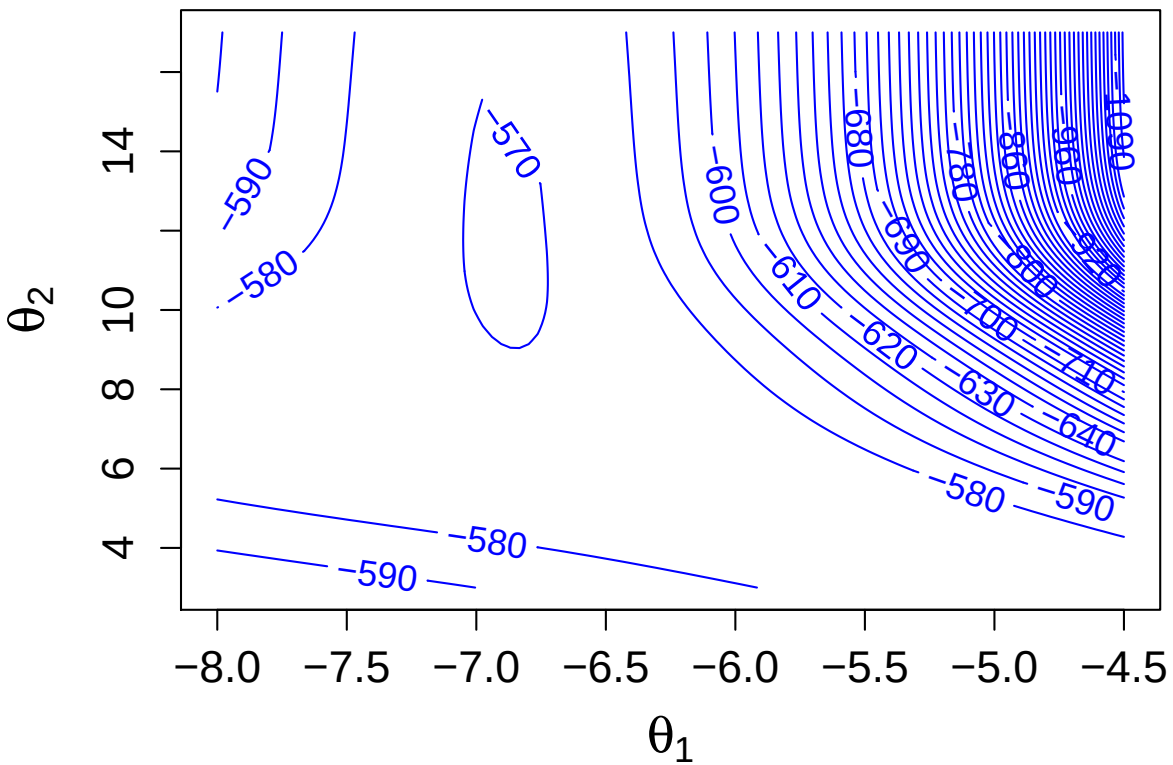
- The arguments of `diff` are `theta` and `data`, where `data` is a list containing `y`, `n`, the posterior mode `post.mode`, the covariance matrix of the normal approximation `post.cov` and the degrees of freedom `df`.

```
data <- list(y=y,n=n,post.mode=post.mode,post.cov=post.cov,df=4)
diff <- function(theta, data){
    y <- data$y
    n <- data$n
    post.mode <- data$post.mode
    post.cov <- data$post.cov
    df <- data$df
    return(logpost(theta,y,n) -
       dmvt(theta,delta=post.mode,sigma=2*post.cov,df=df))
}
```

- The `logpost` function, as well as the objects of `y, n, post.mode, post.cov`, are all the same as defined in Chapter 5.

6.1 Rejection sampling

- The figure below shows a contour plot of the function `diff`. From the plot, the maximum is located close to $\theta = (-6.8, 12)$. We will use this as our starting value in the optimization.



6.1 Rejection sampling

- Using `optim`, we find that the maximum value of `diff` is around -569.2781 and it occurs at $\theta = (-6.89, 12.43)$.

```
> (out1 <- optim(par=c(-6.8,12),fn=diff,
+    control=list(fnscale=-1),data=data))
$par
[1] -6.889241 12.425375
$value
[1] -569.2781
$convergence
[1] 0
```

- In the rejection sampling algorithm, we take M' to be `out1$value`.

```
logM <- out1$value
data$logM <- logM
```

6.1 Rejection sampling

```
rej_sampling <- function(S, data){
    post.mode <- data$post.mode
    df <- data$df
    post.cov <- data$post.cov
    logM <- data$logM
    theta.samples <- matrix(0,S,length(post.mode))
    s <- 0              # no. of samples collected
    T <- 0              # no. of iterations performed
    while (s < S){
        T <- T+1
        theta <- rmvt(1,delta=post.mode,sigma=2*post.cov,df=df)
        U <- runif(1)
        if (log(U) < diff(theta,data) - logM){
            s <- s+1
            theta.samples[s,] <- theta
        }
    }
    accept.rate <- S/T
    list(accept.rate=accept.rate, theta.samples=theta.samples)
}
```

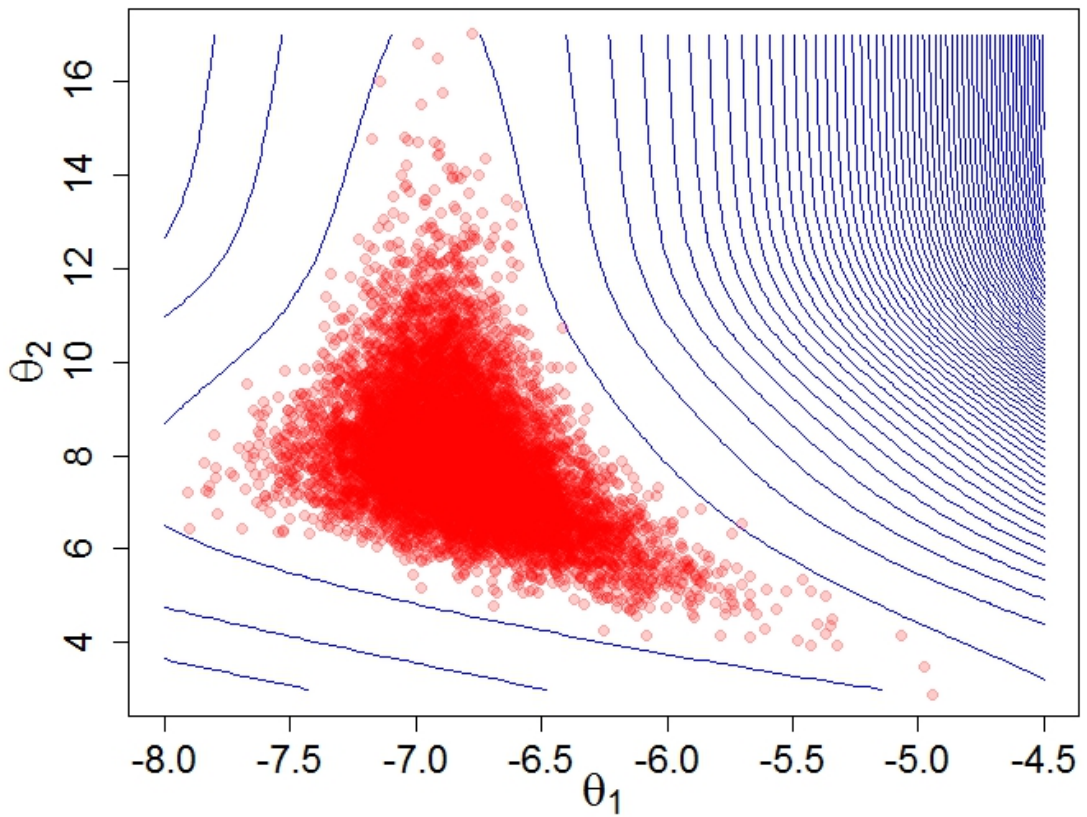
6.1 Rejection sampling

- We use the function `rej_sampling` to generate 10,000 samples from the posterior distribution. This function will run until S independent samples are obtained.
- The acceptance rate of this algorithm is about 0.24. This is a relatively inefficient algorithm since the acceptance rate is low. However, the proposal density (bivariate t_4) was found without too much effort.

```
> set.seed(1)
> S <- 10000
> output <- rej_sampling(S, data)
> output$accept.rate
[1] 0.2389258
```

6.1 Rejection sampling

- The figure below plots the draws from rejection sampling on the contour plot of the log posterior density. As expected, most of the draws fall within the inner contours of the exact density.



Outline

Rejection sampling

Importance sampling

Sampling importance resampling

6.2 Importance sampling

- Let us return to the problem of approximating a posterior expectation when it is not possible to sample directly from the posterior.
- In many situations, the normalizing constant of the posterior density $p(\theta|\boldsymbol{y})$ will be unknown, so let us suppose

$$p(\theta|\boldsymbol{y}) \propto f(\theta).$$

Then the posterior mean of a function $\varphi(\theta)$ is given by

$$\mathrm{E}[\varphi(\theta)|\boldsymbol{y}] = \int \varphi(\theta) p(\theta|\boldsymbol{y}) \, \mathrm{d}\theta = \frac{\int \varphi(\theta) f(\theta) \, \mathrm{d}\theta}{\int f(\theta) \, \mathrm{d}\theta}$$

- Suppose that we can approximate $f(\theta)$ with some density $g(\theta)$ from which we can easily sample.

6.2 Importance sampling

- Then we can rewrite the posterior mean as

$$\mathrm{E}[\varphi(\theta)|\boldsymbol{y}] = \frac{\int \varphi(\theta)\frac{f(\theta)}{g(\theta)}g(\theta)\,\mathrm{d}\theta}{\int \frac{f(\theta)}{g(\theta)}g(\theta)\,\mathrm{d}\theta} = \frac{\int \varphi(\theta)w(\theta)g(\theta)\,\mathrm{d}\theta}{\int w(\theta)g(\theta)\,\mathrm{d}\theta},$$

  where $w(\theta) = \dfrac{f(\theta)}{g(\theta)}$ is the **weight function**.

- If $\{\theta^{(1)}, \ldots, \theta^{(S)}\} \overset{\text{iid}}{\sim} g(\theta)$, then

$$\mathrm{E}[\varphi(\theta)|\boldsymbol{y}] \approx \frac{\frac{1}{S}\sum_{s=1}^{S}\varphi(\theta^{(s)})w(\theta^{(s)})}{\frac{1}{S}\sum_{s'=1}^{S}w(\theta^{(s')})} = \sum_{s=1}^{S} W_s\varphi(\theta^{(s)}) = \widehat{\varphi}_{IS}, \quad (1)$$

  where $W_s = \dfrac{w(\theta^{(s)})}{\sum_{s'=1}^{S}w(\theta^{(s')})}$ are called **normalized weights** since they sum to 1 over all $s = 1, \ldots, S$.

6.2 Importance sampling

- We refer to (1) as an **importance sampling estimate** and $g(\theta)$ as an **importance function**.

- The simulation standard error of an importance sampling estimator $\widehat{\varphi}_{IS}$ is estimated by

$$\sqrt{\sum_{s=1}^{S} \left\{ W_s \left[\varphi(\theta^{(s)}) - \widehat{\varphi}_{IS} \right] \right\}^2}.$$

- How good the estimate in (1) is depends on how closely $g(\theta)$ resembles $f(\theta)$. If $g(\theta)$ is a good approximation, the weights will all be roughly equal, which in turn will minimize the variance of the numerator and denominator.

- On the other hand, if $g(\theta)$ is a poor approximation, many of the weights will be close to zero, and thus a few $\theta^{(s)}$ will dominate the sum, producing an inaccurate approximation.

6.2 Importance sampling

- Preferably, $g(\theta)$ should mimic the posterior density $p(\theta|y)$ and have relatively flat tails so that the weight function $w(\theta)$ is bounded from above.

- One can monitor the choice of g by inspecting the values of the simulated weights $w(\theta^{(s)})$. If there are no unusually large weights, then it is likely that the weight function is bounded and the importance sampler is providing a suitable estimate.

6.2 Importance sampling

Beta-binomial model

- To illustrate the use of different proposal densities in importance sampling, let us consider the problem of estimating the posterior mean of θ_2 conditional on $\theta_1 = -6.82$ (its value in the posterior mode).
- The posterior density of θ_2 conditional on θ_1 is

$$\begin{aligned} p(\theta_2|\theta_1, \boldsymbol{y}) &\propto p(\theta|\boldsymbol{y}) \\ &\propto \frac{K}{(1+K)^2} \prod_{j=1}^{20} \frac{\mathrm{B}(K\eta + y_j, K(1-\eta) + n_j - y_j)}{\mathrm{B}(K\eta, K(1-\eta))} \\ &= f(\theta_2), \end{aligned}$$

where $\eta = \dfrac{\mathrm{e}^{\theta_1}}{1+\mathrm{e}^{\theta_1}}$ and $K = \mathrm{e}^{\theta_2}$.

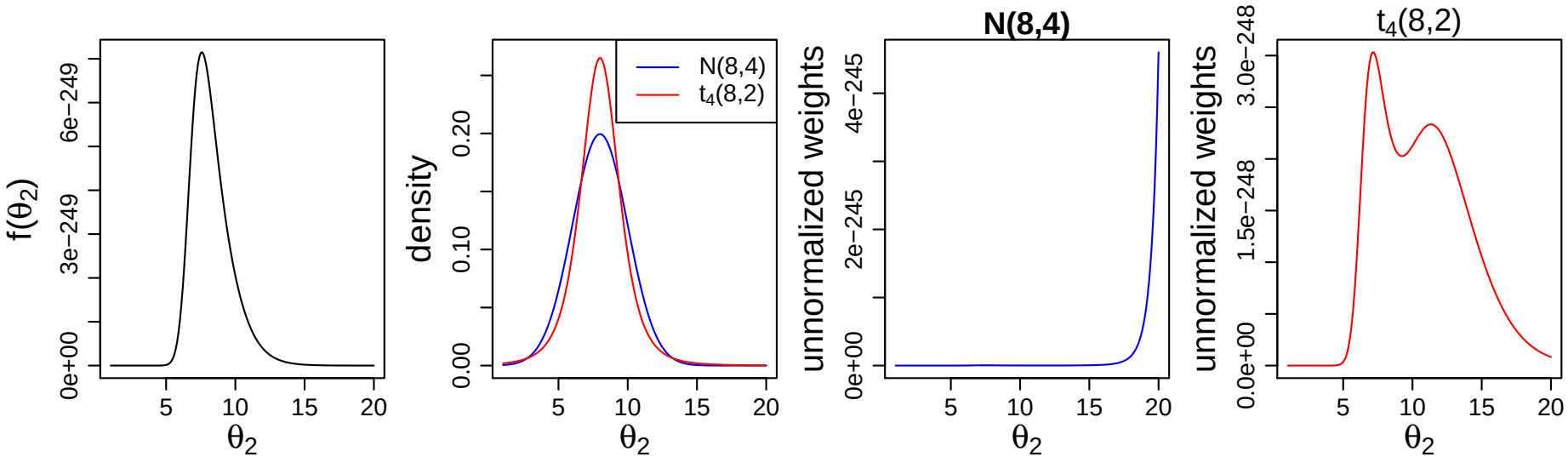
6.2 Importance sampling

- The R-code below writes a function `logf` for computing $\log f(\theta_2)$. This function allows the input of a vector of values for θ_2.

```
logf <- function(theta2,theta1,y,n){
    J <- length(y)   # no. of cities
    eta <- 1/(1+exp(-theta1))
    K <- exp(theta2)
    L <- rep(0, length(theta2))
    for (j in 1:J){
        L <- (L + lbeta(K*eta+y[j],K*(1-eta)+n[j]-y[j])
                - lbeta(K*eta,K*(1-eta)) )
    }
    L <- L + theta2 - 2*log(1+K)
    return(L)
}
```

6.2 Importance sampling

- Let us consider the proposal densities $\mathrm{N}(8, 2^2)$ and $t_4(8, 2)$.
- In the figure below, the leftmost graph shows $f(\theta_2)$, which is proportional to $p(\theta_2|\theta_1, \boldsymbol{y})$, the second graph shows the two proposal densities and the last two graphs show the unnormalized weights computed as $w = f(\theta_2)/g(\theta_2)$.



6.2 Importance sampling

- Although the normal proposal density resembles the posterior density with respect to location and spread, the posterior density has a heavier right tail than the normal proposal density and the weight function is unbounded for large θ_2.

- The t proposal density has heavier tails than the posterior density and the weight function is bounded. Hence the t density is a better proposal for importance sampling.

6.2 Importance sampling

- Using $t_4(8, 2)$ as the proposal density, the R-code below implements importance sampling; simulating $S = 10000$ samples from the proposal density and computing the normalized weights.

```
set.seed(1)
S <- 10000
df <- 4
theta1 <- post.mode[1]
theta2.samples <- rmvt(S,delta=8,sigma=matrix(2,1,1),df=df)
logw <- logf(theta2.samples,theta1,y,n) -
     dmvt(theta2.samples,delta=8,sigma=matrix(2,1,1),df=df)
w <- exp(logw - max(logw))
W <- w/sum(w)
```

6.2 Importance sampling

- An estimate of the posterior mean of θ_2 conditional on $\theta_1 = -6.82$ is then computed together with the simulation standard error.

```
> (est <- sum(W*theta2.samples))
[1] 8.263786
> (se <- sqrt(sum((W*(theta2.samples-est))^2)))
[1] 0.01429726
```

6.2 Importance sampling

- If we are interested instead in the posterior marginal mean of θ_2, we can use importance sampling to obtain samples of $\theta = (\theta_1, \theta_2)$ from the joint posterior distribution $p(\theta|y)$. The proposal is a multivariate t distribution with 4 degrees of freedom, where the location is the posterior mode and scale matrix is twice the covariance matrix of the normal approximation.

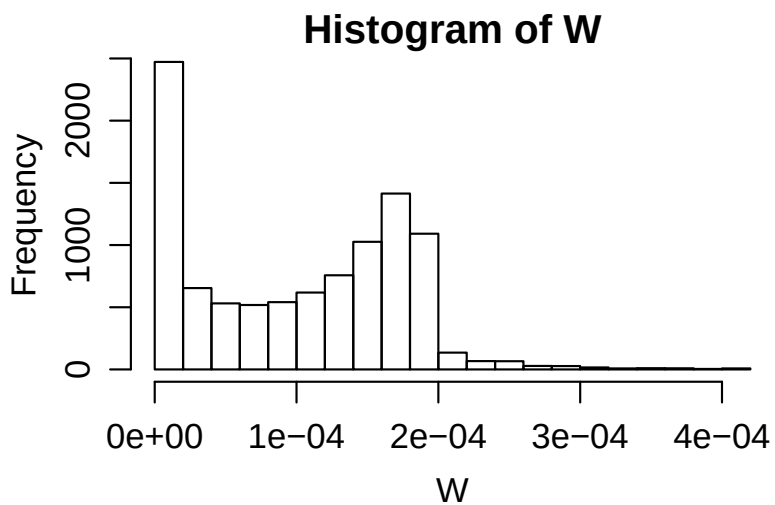
```
set.seed(1)
S <- 10000
theta.samples <- rmvt(S,delta=post.mode,sigma=2*post.cov,df=df)
lf <- rep(0,S)
for (s in 1:S){ lf[s] <- logpost(theta.samples[s,],y,n) }
logw <- lf -
  dmvt(theta.samples,delta=post.mode,sigma=2*post.cov,df=df)
w <- exp(logw - max(logw))
W <- w/sum(w)
```

6.2 Importance sampling

- From the output, the importance sampling estimate of the marginal posterior mean of θ_2 is 7.94 with an associated standard error of 0.019.

```
> (est <- sum(W*theta.samples[,2]))
[1] 7.944095
> (se <- sqrt(sum((W*(theta.samples[,2]-est))^2)))
[1] 0.01867257
```

- The figure below shows a histogram of the normalized weights. We note that there are no extreme weights.



Outline

Rejection sampling

Importance sampling

Sampling importance resampling

6.3 Sampling importance resampling

- Suppose we have obtained a sample $\{\theta^{(1)}, \ldots, \theta^{(S)}\}$ with corresponding normalized weights $\{W_1, \ldots, W_S\}$ using importance sampling.

- We can obtain a new sample $\{\theta^{*(1)}, \ldots, \theta^{*(S)}\}$ by sampling from the discrete distribution over $\{\theta^{(1)}, \ldots, \theta^{(S)}\}$ with respective probabilities $\{W_1, \ldots, W_S\}$.

- Then the $\{\theta^{*(s)}\}$ will be approximately distributed according to the posterior distribution $p(\theta|\boldsymbol{y})$. This method, called **sampling importance resampling (SIR)**, is a weighted bootstrap procedure where we sample with replacement from $\{\theta^{(1)}, \ldots, \theta^{(S)}\}$ with unequal sampling probabilities.

- The SIR algorithm is straightforward to implement in R using the `sample` command.

6.3 Sampling importance resampling

- Suppose the samples obtained using importance sampling are stored in the $S \times p$ matrix `theta.samples`, where p denotes the length of θ, and the weights are stored in the vector `W`.

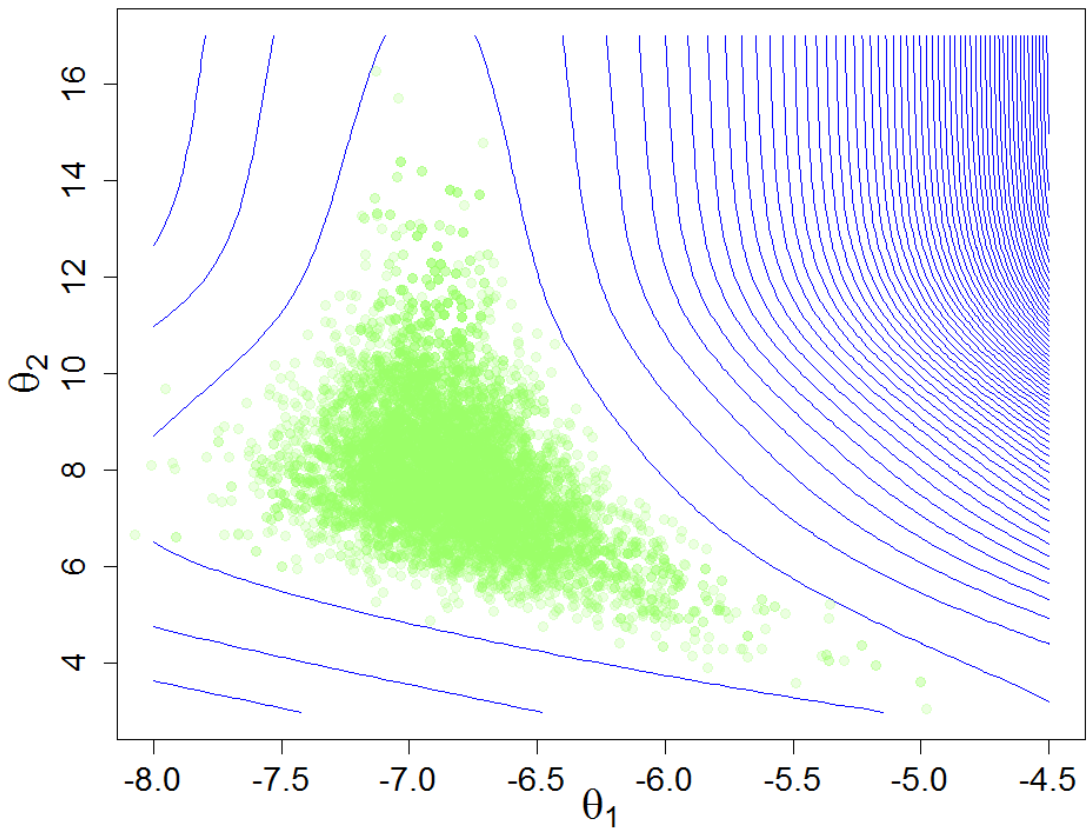
- To obtain a simulated sample of size N using the SIR algorithm, we use `sample` to take a sample with replacement from the indices $1, \ldots, S$, where the sampling probabilities are contained in `W` and the simulated indices are stored in the vector `indices`.

- We then use `indices` to select the rows of `theta.samples` and assign them to the matrix `theta.SIR`. The matrix `theta.SIR` contains the simulated draws from the posterior.

```
indices <- sample(1:S, size=N, prob=W, replace=TRUE)
theta.SIR <- theta.samples[indices,]
```

6.3 Sampling importance resampling

- The figure below plots the draws from SIR on the contour plot of the log posterior density. As expected, most of the draws fall within the inner contours of the exact density. The quality of SIR is comparable to that of rejection sampling.



6.3 Sampling importance resampling

- The SIR algorithm can also be used to convert simulated draws from one probability density to another probability density.

- Suppose we wish to perform a Bayesian sensitivity analysis with respect to the individual observations in the `cancermortality` dataset. Let us focus on posterior inference about θ_2 and question how the inference would change if we removed individual observations from the likelihood.

- Let $p(\theta|\boldsymbol{y})$ denote the posterior density from the full dataset as before and $p(\theta|\boldsymbol{y}_{-i})$ denote the posterior density with the ith observation removed. Let $\{\theta^{(s)}\}$ represent a simulated sample from the full dataset.

6.3 Sampling importance resampling

- We can obtain a simulated sample from $p(\theta|\boldsymbol{y}_{-i})$ by resampling from $\{\theta^{(s)}\}$, where the sampling probabilities are proportional to the weights

$$
\begin{aligned}
w(\theta^{(s)}) &= \frac{p(\theta^{(s)}|\boldsymbol{y}_{-i})}{p(\theta^{(s)}|\boldsymbol{y})} \\
&= \frac{p(\boldsymbol{y}_{-i}|\theta^{(s)})\cancel{p(\theta^{(s)})}/p(\boldsymbol{y}_{-i})}{p(\boldsymbol{y}|\theta^{(s)})\cancel{p(\theta^{(s)})}/p(\boldsymbol{y})} \\
&\propto \frac{p(\boldsymbol{y}_{-i}|\theta^{(s)})}{p(\boldsymbol{y}|\theta^{(s)})} \\
&= \frac{1}{p(y_i|\theta^{(s)})} = \frac{\mathrm{B}(K^{(s)}\eta^{(s)}, K^{(s)}(1-\eta^{(s)}))}{\mathrm{B}(K^{(s)}\eta^{(s)}+y_i, K^{(s)}(1-\eta^{(s)})+n_i-y_i)}.
\end{aligned}
$$

where $\eta^{(s)} = \dfrac{\exp(\theta_1^{(s)})}{1+\exp(\theta_1^{(s)})}$ and $K^{(s)} = \exp(\theta_2^{(s)})$.

6.3 Sampling importance resampling

- Suppose we are interested in 90% confidence intervals for θ_1 and θ_2. To look at how posterior inference would change if observation i were left out, we proceed as follows.

- Let us assume that we have already simulated samples from the complete data posterior $p(\theta|\boldsymbol{y})$ and the draws are stored in the matrix `theta.samples`.

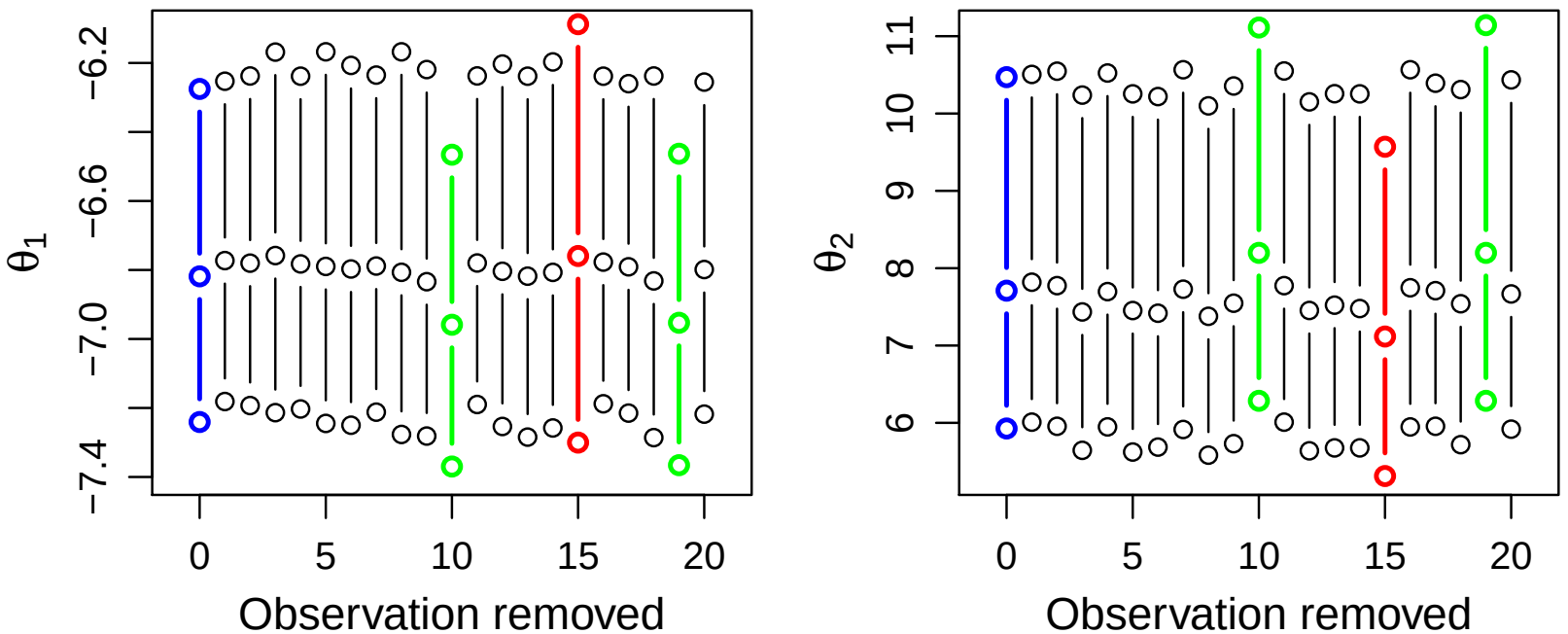
- First we compute the sampling weights and normalize them to obtain the sampling probabilities. Then the sample command is used to do the resampling from `theta.samples` and the simulated draws from the leave one out posterior are stored in `theta.SIR`. We summarize the simulated values of θ_1 and θ_2 by the 5th, 50th, and 95th quantiles.

6.3 Sampling importance resampling

```
S <- nrow(theta.samples)
summary.full1 <- quantile(theta.samples[,1],c(0.05,0.5,0.95))
summary.full2 <- quantile(theta.samples[,2],c(0.05,0.5,0.95))
summary.obs1 <- matrix(0, 20, 3)
summary.obs2 <- matrix(0, 20, 3)
for (i in 1:20){
    eta <- 1/(1+exp(-theta.samples[,1]))
    K <- exp(theta.samples[,2])
    logw <- lbeta(K*eta,K*(1-eta)) -
            lbeta(K*eta+y[i],K*(1-eta)+n[i]-y[i])
    w <- exp(logw - max(logw))
    W <- w/sum(w)
    indices <- sample(1:S, size=N, prob=W, replace=TRUE)
    theta.SIR <- theta.samples[indices,]
    summary.obs1[i,] <- quantile(theta.SIR[,1],c(0.05,0.5,0.95))
    summary.obs2[i,] <- quantile(theta.SIR[,2],c(0.05,0.5,0.95))
}
```

6.3 Sampling importance resampling

- The figure below displays graphically the sensitivity of the posterior inference about θ_1 and θ_2 with respect to the individual observations.
- The bold blue lines shows the posterior median and 90% confidence interval for the complete dataset, and the remaining lines show the inference with each possible observation removed.



6.3 Sampling importance resampling

- The parameters η and K in the prior Beta$(K\eta, K(1-\eta))$ represent the mean and provide a measure of the precision respectively. The transformed parameters θ_1 and θ_2 are increasing functions of η and K.

- If observation 15 is removed, where $(y_{15}, n_{15}) = (54, 53637)$, then the location of θ_1 is shifted towards larger values (higher average mortality) while θ_2 is shifted toward smaller values (less precision).

- Also, if either observations 10 or 19 are removed, θ_1 is shifted towards smaller values (lower average mortality) while θ_2 is shifted toward larger values (more precision). These two observations are notable since each city experienced three deaths and had relatively high mortality rates.

6.3 Sampling importance resampling

- The plot can be obtained using the R-code below.

```
> par(mfrow=c(1,2))
> plot(c(0,0,0), summary.full1, xlim=c(-1,21),
+   ylim=c(-7.4,-6.1), type="b", col="blue", lwd=2)
> for (i in 1:20){lines(c(i,i,i),summary.obs1[i,],type="b")}
> plot(c(0,0,0), summary.full2, xlim=c(-1,21),
+   ylim=c(5.3,11.1), type="b", col="blue", lwd=2)
> for (i in 1:20){lines(c(i,i,i),summary.obs2[i,],type="b")}
> c(y[15],n[15])
[1]    54 53637
> c(y[10],n[10])
[1]   3 582
> c(y[19],n[19])
[1]   3 588
```