3

Monte Carlo Integration

"Every time I think I know what's going on, suddenly there's another layer of complications. I just want this damn thing solved."

John Scalzi
The Last Colony

Reader's guide

While Chapter 2 focused on the simulation techniques useful to produce random variables by computer, this chapter introduces the major concepts of Monte Carlo methods; that is, taking advantage of the availability of computer-generated random variables to approximate univariate and multidimensional integrals. In Section 3.2, we introduce the basic notion of Monte Carlo approximations as a by-product of the Law of Large Numbers, while Section 3.3 highlights the universality of the approach by stressing the versatility of the representation of an integral as an expectation. Chapter 5 will similarly deal with the resolution of optimization problems by simulation techniques.

C.P. Robert, G. Casella, *Introducing Monte Carlo Methods with R*, Use R,
DOI 10.1007/978-1-4419-1576-4_3, © Springer Science+Business Media, LLC 2010

3.1 Introduction

Two major classes of numerical problems that arise in statistical inference are *optimization* problems and *integration* problems. Indeed, numerous examples (see Rubinstein, 1981, Gentle, 2002, or Robert, 2001) show that it is not always possible to analytically compute the estimators associated with a given paradigm (maximum likelihood, Bayes, method of moments, etc.).

Thus, whatever the type of statistical inference, we are often led to consider numerical solutions. The previous chapter introduced a number of methods for the computer generation of random variables with any given distribution and hence provides a basis for the construction of solutions to our statistical problems. A general solution is indeed to use simulation, of either the true or some substitute distributions, to calculate the quantities of interest. In the setup of decision theory, whether it is classical or Bayesian, this solution is natural since risks and Bayes estimators involve integrals with respect to probability distributions.

Note that the possibility of producing an almost infinite number of random variables distributed according to a given distribution gives us access to the use of *frequentist* and *asymptotic* results much more easily than in the usual inferential settings, where the sample size is most often fixed. One can therefore apply probabilistic results such as the Law of Large Numbers or the Central Limit Theorem, since they allow assessment of the convergence of simulation methods (which is equivalent to the deterministic bounds used by numerical approaches).

Before embarking upon the description of Monte Carlo techniques, note that an apparently obvious alternative to the use of simulation methods for approximating integrals of the form

$$\int_{\mathcal{X}} h(x)\, f(x)\, \mathrm{d}x,$$

where f is a probability density, would be to rely on numerical methods such as Simpson's and the trapezium rules. For instance, `R` offers two related functions that run unidimensional integration, `area` (in the `MASS` library) and `integrate`. However, `area` cannot deal with infinite bounds in the integral and therefore requires some prior knowledge of the region of integration. The other function, `integrate`, accepts infinite bounds but is unfortunately very fragile and can produce untrustworthy output.

Example 3.1. As a test, we compare the use of `integrate` on the integral

$$\int_0^\infty x^{\lambda-1} \exp(-x)\, \mathrm{d}x$$

with the computation of $\Gamma(\lambda)$ via the `gamma` function. Implementing this comparison as

```
> ch=function(la){
+   integrate(function(x){x^(la-1)*exp(-x)},0,Inf)$val}
> plot(lgamma(seq(.01,10,le=100)),log(apply(as.matrix(
+ seq(.01,10,le=100)),1,ch)),xlab="log(integrate(f))",
+ ylab=expression(log(Gamma(lambda))),pch=19,cex=.6)
```

we obtain the sequence represented in Figure 3.1, which does not show any discrepancy even for very small values of λ. ◄

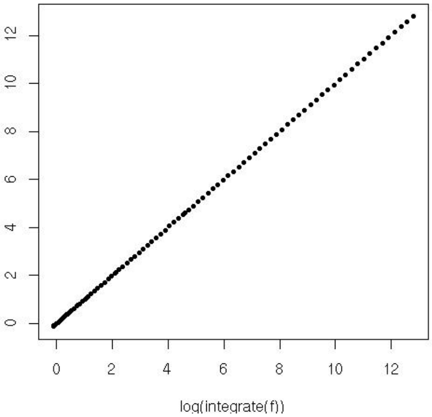

Fig. 3.1. Comparison of the integrate evaluation of the $\Gamma(\lambda)$ integral with its true value.

A main difficulty with numerical integration methods such as `integrate` is that they often fail to spot the region of importance for the function to be integrated. In contrast, simulation methods naturally target this region by exploiting the information provided by the probability density associated with the integrals.

Example 3.2. Consider a sample of ten Cauchy rv's x_i $(1 \le i \le 10)$ with location parameter $\theta = 350$. The (pseudo-) marginal of the sample under a flat prior is then

$$m(\mathbf{x}) = \int_{-\infty}^{\infty} \prod_{i=1}^{10} \frac{1}{\pi} \frac{1}{1+(x_i-\theta)^2} \, \mathrm{d}\theta \,.$$

However, `integrate` returns a wrong numerical value

```
> cac=rcauchy(10)+350
> lik=function(the){
+   u=dcauchy(cac[1]-the)
+   for (i in 2:10)
+       u=u*dcauchy(cac[i]-the)
+    return(u)}
>  integrate(lik,-Inf,Inf)
7.38034e-46 with absolute error < 1.5e-45
> integrate(lik,200,400)
4.83155e-13 with absolute error < 9e-13
```

and fails to signal the difficulty since the error evaluation is absurdly small. Furthermore, the result is not comparable to area:

```
> cac=rcauchy(10)
> nin=function(a){integrate(lik,-a,a)$val}
> nan=function(a){area(lik,-a,a)}
> x=seq(1,10^3,le=10^4)
> y=log(apply(as.matrix(x),1,nin))
> z=log(apply(as.matrix(x),1,nan))
> plot(x,y,type="l",ylim=range(cbind(y,z)),lwd=2)
> lines(x,z,lty=2,col="sienna",lwd=2)
```

Using area in that case produces a more reliable evaluation, as shown in Figure 3.2, since area(lik,-a,a) flattens out as a increases, but this obviously requires some prior knowledge about the location of the mode of the integrand. ◀

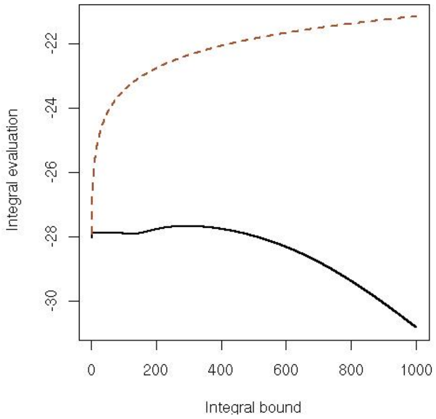

Fig. 3.2. Comparison of integrate and area on the integral of a Cauchy likelihood in log scale (the outcome of area corresponds to the dashed curve above).

Lastly, numerical integration tools cannot easily face the highly (or even moderately) multidimensional integrals that are the rule in statistical problems. Devising specific integration tools for those problems would be too costly, especially because we can take advantage of the probabilistic nature of those integrals.

3.2 Classical Monte Carlo integration

Before applying our simulation techniques to practical problems, let us recall the properties that justify their use, referring to Robert and Casella (2004) for (many) more details. The generic problem is about evaluating the integral

$$\mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x)\, f(x)\, \mathrm{d}x, \tag{3.1}$$

where $\mathcal{X}$ denotes the set where the random variable X takes its values, which is usually equal to the support of the density f. The principle of the Monte Carlo method for approximating (3.1) is to generate a sample $(X_1, \ldots, X_n)$ from the density f and propose as an approximation the empirical average

$$\overline{h}_n = \frac{1}{n} \sum_{j=1}^{n} h(x_j)\ ,$$

computed by `mean(h(x))` in `R`, since $\overline{h}_n$ converges almost surely (i.e. for almost every generated sequence) to $\mathbb{E}_f[h(X)]$ by the Strong Law of Large Numbers. Moreover, when $h^2(X)$ has a finite expectation under f, the speed of convergence of $\overline{h}_n$ can be assessed since the convergence takes place at a speed $\mathrm{O}(\sqrt{n})$ and the asymptotic variance of the approximation is

$$\mathrm{var}(\overline{h}_n) = \frac{1}{n} \int_{\mathcal{X}} \left(h(x) - \mathbb{E}_f[h(X)]\right)^2 f(x)\mathrm{d}x,$$

which can also be estimated from the sample $(X_1, \ldots, X_n)$ through

$$v_n = \frac{1}{n^2} \sum_{j=1}^{n} [h(x_j) - \overline{h}_n]^2\ .$$

More specifically, due to the Central Limit Theorem, for large n,

$$\frac{\overline{h}_n - \mathbb{E}_f[h(X)]}{\sqrt{v_n}}$$

is approximately distributed as a $\mathcal{N}(0,1)$ variable, and this leads to the construction of a convergence test and confidence bounds on the approximation of $\mathbb{E}_f[h(X)]$.

Example 3.3. For the toy function[1]

$$h(x) = [\cos(50x) + \sin(20x)]^2\ , \tag{3.2}$$

represented in the upper panel of Figure 3.3, consider evaluating its integral over $[0,1]$. It can be seen as a uniform expectation, and therefore we generate $U_1, U_2, \ldots, U_n$ iid $\mathcal{U}(0,1)$ random variables and approximate $\int h(x)\mathrm{d}x$ with

[1] This function can be integrated analytically.

$\sum h(U_i)/n$. The lower panel in Figure 3.3 shows the running means and the bounds derived from the estimated standard errors against the number n of simulations. The R implementation is as follows:

```
> h=function(x){(cos(50*x)+sin(20*x))^2}
> par(mar=c(2,2,2,1),mfrow=c(2,1))
> curve(h,xlab="Function",ylab="",lwd=2)
> integrate(h,0,1)
0.965201 with absolute error < 1.9e-10
> x=h(runif(10^4))
> estint=cumsum(x)/(1:10^4)
> esterr=sqrt(cumsum((x-estint)^2))/(1:10^4)
> plot(estint, xlab="Mean and error range",type="l",lwd=
+ 2,ylim=mean(x)+20*c(-esterr[10^4],esterr[10^4]),ylab="")
> lines(estint+2*esterr,col="gold",lwd=2)
> lines(estint-2*esterr,col="gold",lwd=2)
```

Note that the confidence band produced in this figure is not a 95% confidence band in the classical sense (i.e., it does not correspond to a confidence band on the graph of estimates, but rather to the confidence assessment that you can produce for every number of iterations, were you to stop at this number of iterations). ◀

ϟ While the bonus brought by the simultaneous evaluation of the error of the Monte Carlo estimate cannot be disputed, you must be aware that it is only trustworthy as far as v_n is a proper estimate of the variance of $\overline{h}_n$. In critical situations where v_n does not converge at all or does not even converge fast enough for a CLT to apply, this estimate and the confidence region associated with it cannot be trusted.

When monitoring Monte Carlo convergence, an issue that will be fully addressed in the next chapter, the R command `cumsum` is quite handy in that it computes all the partial sums of a sequence at once and thus allows the immediate representation of the sequence of estimators.

Exercise 3.1 For the normal-Cauchy Bayes estimator

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

solve the following questions when $x = 0, 2, 4$.

a. Plot the integrands, and use Monte Carlo integration based on a Cauchy simulation to calculate the integrals.
b. Monitor the convergence with the standard error of the estimate. Obtain three digits of accuracy with probability $.95$.
c. Repeat the experiment with a Monte Carlo integration based on a normal simulation and compare both approaches.

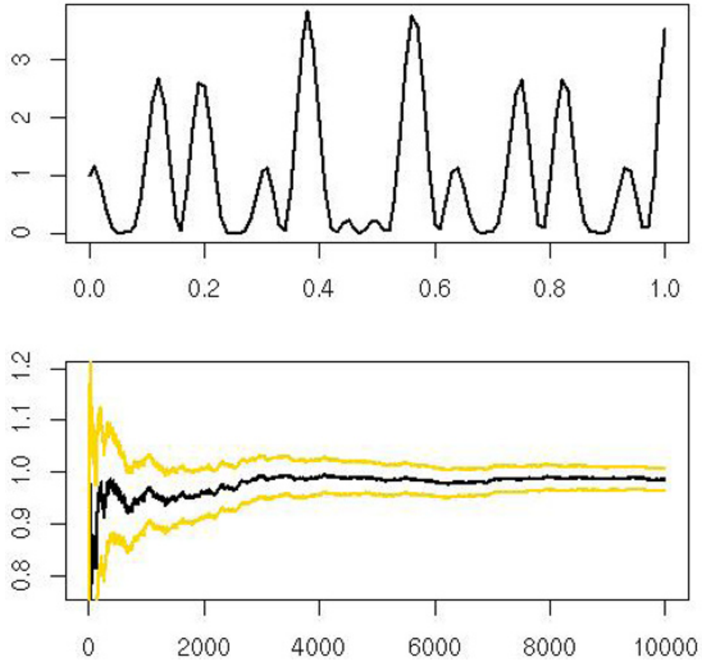

Fig. 3.3. Approximation of the integral of the function (3.2): *(upper)* function (3.2), and *(lower)* mean ± two standard errors against iterations for a single sequence of simulations.

The Monte Carlo methodology illustrated by the example above can be successfully implemented in a wide range of cases where the distributions involved in the model can be simulated. For instance, we could use Monte Carlo sums to calculate a normal cumulative distribution function (even though the normal cdf can now be found in all software and many pocket calculators).

Example 3.4. Given a normal $\mathcal{N}(0,1)$ sample of size n, $(x_1, \ldots, x_n)$, the approximation of

$$\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \mathrm{d}y$$

by the Monte Carlo method is

$$\hat{\Phi}(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{x_i \le t},$$

Table 3.1. Evaluation of some normal probabilities $\Pr(X \leq t)$ by a regular Monte Carlo experiment based on n replications of a normal generation. The last line achieves the exact values.

n/t	0.0	0.67	0.84	1.28	1.65	2.32	2.58	3.09	3.72
10^2	0.485	0.74	0.77	0.9	0.945	0.985	0.995	1	1
10^3	0.4925	0.7455	0.801	0.902	0.9425	0.9885	0.9955	0.9985	1
10^4	0.4962	0.7425	0.7941	0.9	0.9498	0.9896	0.995	0.999	0.9999
10^5	0.4995	0.7489	0.7993	0.9003	0.9498	0.9898	0.995	0.9989	0.9999
10^6	0.5001	0.7497	0.8	0.9002	0.9502	0.99	0.995	0.999	0.9999
10^7	0.5002	0.7499	0.8	0.9001	0.9501	0.99	0.995	0.999	0.9999
10^8	0.5	0.75	0.8	0.9	0.95	0.99	0.995	0.999	0.9999

with (exact) variance $\Phi(t)[1-\Phi(t)]/n$ (since the variables $\mathbb{I}_{x_i \leq t}$ are independent Bernoulli with success probability $\Phi(t)$). The R implementation that led to Table 3.1 is

```
> x=rnorm(10^8)                    #whole sample
> bound=qnorm(c(.5,.75,.8,.9,.95,.99,.999,.9999))
> res=matrix(0,ncol=8,nrow=7)
> for (i in 2:8)                   #lengthy loop!!
+ for (j in 1:8)
+   res[i-1,j]=mean(x[1:10^i]<bound[j])
> matrix(as.numeric(format(res,digi=4)),ncol=8)
```

For values of t around $t=0$, the variance is thus approximately $1/4n$, and to achieve a precision of four decimals, we need $2 \times \sqrt{1/4n} \leq 10^{-4}$ simulations, i.e., about $n = (10^4)^2 = 10^8$ simulations. Table 3.1 gives the evolution of this approximation for several values of t and shows an accurate evaluation for 100 million iterations. Note that greater (absolute) accuracy is achieved in the tails and that (much) more efficient simulation methods could be used. ◀

As you have presumably noticed, the outputs in R are represented with all the available digits, as in

```
> rnorm(1)
[1] -0.08581098
```

While this is logical from an informatic point of view, it is not recommended to produce all those digits in statistical and simulation environments because most of them are not significant and also because it impairs the readability of the output. The `format` function is then quite handy to cut down the number of represented digits, as shown in the last line of the R program above.

The Monte Carlo approximation of a probability distribution function illustrated by Example 3.4 has nontrivial applications since it can be used in assessing the distribution of a test statistic such as a likelihood ratio test under a null hypothesis, as illustrated in Robert and Casella (2004), as well as its power under alternatives.

It may thus seem at this stage that the Monte Carlo methodology introduced in this section is sufficient to approximate integrals like (3.1) in a controlled way. However, while the straightforward Monte Carlo method indeed provides good approximations of (3.1) in most regular cases, there exist more efficient alternatives that not only avoid a direct simulation from f but also can be used repeatedly for several integrals of the form (3.1). The repeated use can be for either a family of functions h or a family of densities f. In addition, problems of tail simulation as in Example 3.4 can be processed much more efficiently than simulating from f since simulating events with a very small probability requires a very large number of simulations under f to achieve a given (relative) precision.

Exercise 3.2 Given that $\mathbb{I}_{X_i \le t}$ is a Bernoulli random variable equal to 1 with probability $\Phi(t)$, show that the variance of the normalized estimator $\mathbb{I}_{X_i \le t}/\Phi(t)$ goes to infinity when t decreases to $-\infty$. Deduce the number of simulations (as a function of t) that are necessary to achieve a variance less than 10^{-8}.

Exercise 3.3 If we are interested in the tail probability $\Pr(X > 20)$ when $X \sim \mathcal{N}(0,1)$, simulating from a $\mathcal{N}(0,1)$ distribution does not work. Express the probability as an integral and use an obvious change of variable to rewrite this integral as an expectation under a $\mathcal{U}(0,1/20)$ distribution. Deduce a Monte Carlo approximation to $\Pr(X > 20)$ along with an error assessment.

3.3 Importance sampling

The method we now study is called *importance sampling* because it relies on so-called *importance functions*, which are instrumental distributions , in lieu of the original distributions. In fact, an evaluation of (3.1) based on simulations from f is almost never optimal in the sense that using alternative distributions can improve the variance of the resulting estimator of (3.1).

3.3.1 An arbitrary change of reference measure

The importance sampling method is based on an alternative representation of (3.1). Given an arbitrary density g that is strictly positive when $h \times f$ is different from zero, we can indeed rewrite (3.1) as

$$(3.3) \qquad \mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x)\,\frac{f(x)}{g(x)}\,g(x)\,\mathrm{d}x = \mathbb{E}_g\left[\frac{h(X)f(X)}{g(X)}\right]\;;$$

that is, as an expectation under the density g. (Note that $\mathcal{X}$ is again the set where X takes its value and that it may therefore be *smaller* than the support of the density g.) This *importance sampling fundamental identity* justifies the use of the estimator

$$\frac{1}{n}\sum_{j=1}^{n}\frac{f(X_j)}{g(X_j)}\,h(X_j)\rightarrow\mathbb{E}_f[h(X)] \tag{3.4}$$

based on a sample $X_1,\ldots,X_n$ generated from g (not from f!). Indeed, because (3.1) can thus be written as an expectation under g, (3.4) does converge to (3.1) for the same reason the regular Monte Carlo estimator $\overline{h}_n$ converges, whatever the choice of the distribution g (as long as $\text{supp}(g)\supset\text{supp}(h\times f)$). This ubiquitous property relates to the fact that (3.1) can be represented in an infinite number of ways by pairs (h,f) and thus that a given integral is not intrinsically associated with a given distribution. On the contrary, there is almost absolute freedom in its representation as an expectation.

↯ The constraint on the support of g, $\text{supp}(g)\supset\text{supp}(h\times f)$, is absolute in that using a smaller support truncates the integral (3.3) and thus produces a biased result. This means, in particular, that when considering nonparametric solutions for g, the support of the kernel must be unrestricted.

Exercise 3.4 For the computation of the expectation $\mathbb{E}_f[h(X)]$ when f is the normal pdf and $h(x)=\exp(-(x-3)^2/2)+\exp(-(x-6)^2/2)$:

a. Show that $\mathbb{E}_f[h(X)]$ can be computed in closed form and derive its value.
b. Construct a regular Monte Carlo approximation based on a normal $\mathcal{N}(0,1)$ sample of size `Nsim=10^3` and produce an error evaluation.
c. Compare the above with an importance sampling approximation based on an importance function g corresponding to the $\mathcal{U}(-8,-1)$ distribution and a sample of size `Nsim=10^3`. (Warning: This choice of g does not provide a converging approximation of $\mathbb{E}_f[h(X)]$!)

Example 3.5. As mentioned at the end of Example 3.4, approximating tail probabilities using standard Monte Carlo sums breaks down once one goes far enough into the tails. For example, if $Z\sim\mathcal{N}(0,1)$ and we are interested in the probability $P(Z>4.5)$, which is very small,

```
> pnorm(-4.5,log=T)
[1] -12.59242
```

simulating $Z^{(i)}\sim\mathcal{N}(0,1)$ only produces a hit once in about 3 million iterations!

Of course, the problem is that we are now interested in the probability of a very rare event and thus naïve simulation from f will require a huge number of simulations to get a stable answer. However, thanks to importance sampling, we can greatly improve our accuracy and thus bring down the number of simulations by several orders of magnitude.

For instance, if we consider a distribution with support restricted to $(4.5, \infty)$, the additional and unnecessary variation of the Monte Carlo estimator due to simulating zeros (i.e., when $x < 4.5$) disappears. A natural choice is to take g as the density of the exponential distribution $\mathcal{E}xp(1)$ truncated at 4.5,

$$g(y) = e^{-y} \Big/ \int_{4.5}^{\infty} e^{-x} \mathrm{d}x = e^{-(y-4.5)} \,,$$

and the corresponding importance sampling estimator of the tail probability is

$$\frac{1}{n} \sum_{i=1}^{n} \frac{f(Y^{(i)})}{g(Y^{(i)})} = \frac{1}{n} \sum_{i=1}^{n} \frac{e^{-Y_i^2/2+Y_i-4.5}}{\sqrt{2\pi}} \,,$$

where the Y_i's are iid generations from g. The corresponding code is

```
> Nsim=10^3
> y=rexp(Nsim)+4.5
> weit=dnorm(y)/dexp(y-4.5)
> plot(cumsum(weit)/1:Nsim,type="l")
> abline(a=pnorm(-4.5),b=0,col="red")
```

The final value is then $3.312\,10^{-6}$, to be compared with the true value of 3.398×10^{-6}. As shown in Figure 3.4, the accuracy of the approximation is remarkable, especially when compared with the original size requirements imposed by a normal simulation. ◀

Exercise 3.5 In the exercise above, examine the impact of using a truncated exponential distribution $\mathcal{E}xp(\lambda)$ on the variance of the approximation of the tail probability.

Importance sampling is therefore of considerable interest since it puts very little restriction on the choice of the instrumental distribution g, which can be chosen from distributions that are either easy to simulate or efficient in the approximation of the integral. Moreover, the same sample (generated from g) can be used repeatedly, not only for different functions h but also for different densities f.

Example 3.6. This example stems from a Bayesian setting: When considering an observation x from a beta $\mathcal{B}(\alpha, \beta)$ distribution,

$$x \sim \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \, x^{\alpha-1}(1-x)^{\beta-1} \, \mathbb{I}_{[0,1]}(x),$$

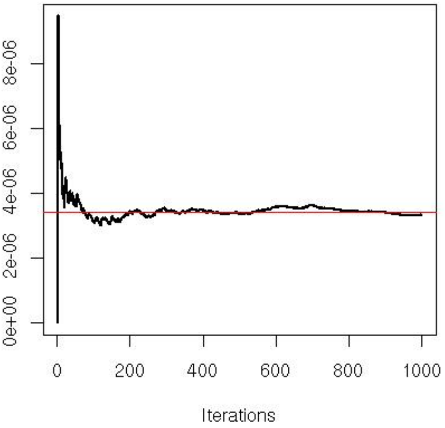

Fig. 3.4. Convergence of the importance sampling approximation of the normal tail probability $P(Z \geq 4.5)$, based on a sequence simulated from a translated exponential distribution. The straight line corresponds to the true value of the integral.

there exists a family of conjugate priors on (α, β) of the form

$$\pi(\alpha,\beta) \propto \left\{ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \right\}^{\lambda} x_0^{\alpha} y_0^{\beta} \,,$$

where λ, x_0, y_0 are hyperparameters, since the posterior is then equal to

$$\pi(\alpha,\beta|x) \propto \left\{ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \right\}^{\lambda+1} [x x_0]^{\alpha} [(1-x) y_0]^{\beta} \,.$$

This family of distributions is intractable if only because of the difficulty of dealing with gamma functions. Simulating directly from $\pi(\alpha, \beta|x)$ is therefore impossible. We thus need to use a substitute distribution $g(\alpha, \beta)$, and we can get a preliminary idea by looking at an image representation of $\pi(\alpha, \beta|x)$. If we take $\lambda = 1$, $x_0 = y_0 = .5$, and $x = .6$, the R code is

```
> f=function(a,b){
+   exp(2*(lgamma(a+b)-lgamma(a)-lgamma(b))+
+       a*log(.3)+b*log(.2))}
> aa=1:150      #alpha grid for image
> bb=1:100      #beta grid for image
> post=outer(aa,bb,f)
> image(aa,bb,post,xlab=expression(alpha),ylab=" ")
> contour(aa,bb,post,add=T)
```

The outer command is a handy abbreviation to compute a matrix A=outer(a,b,f) of dimension c(dim(a),dim(b)) whose A[i,j] element is equal to f(a[i],b[j]). While it is much faster than the basic double allocation loop,

```
> system.time(outer(aa,bb,f))
  user  system elapsed
 0.028   0.000   0.029
> system.time(for (j in 1:100){for (i in 1:150)
+ post[i,j]=f(a=aa[i],b=bb[j])})
  user  system elapsed
 0.360   0.004   0.367
```

it compares speedwise with a single allocation loop

```
> system.time(outer(aa,bb,f))
  user  system elapsed
 0.028   0.000   0.028
> system.time(for (j in 1:100){post[,j]=f(a=aa,b=bb[j])})
  user  system elapsed
 0.028   0.000   0.027
> system.time(for (i in 1:150){post[i,]=f(a=aa[i],b=bb)})
  user  system elapsed
 0.032   0.000   0.031
```

and thus does not offer a superefficient way to allocate values to a matrix.

The examination of Figure 3.5 *(left)* shows that a normal or a Student's t distribution on the pair (α,β) could be appropriate. Choosing a Student's $\mathcal{T}(3,\mu,\Sigma)$ distribution with $\mu=(50,45)$ and

$$\Sigma=\begin{pmatrix}220 & 190\\ 190 & 180\end{pmatrix}$$

does produce a reasonable fit, as shown on Figure 3.5 *(right)* using the superposition of simulation from this $\mathcal{T}(3,\mu,\Sigma)$ distribution with the surface of the posterior distribution. The covariance matrix above was obtained by trial-and-error, modifying the entries until the sample in Figure 3.5 *(right)* fit well enough:

```
> x=matrix(rt(2*10^4,3),ncol=2)         #T sample
> E=matrix(c(220,190,190,180),ncol=2) #Scale matrix
> image(aa,bb,post,xlab=expression(alpha),ylab=" ")
> y=t(t(chol(E))%*%t(x)+c(50,45))
> points(y,cex=.6,pch=19)
```

Note the use of t(chol(E)) to ensure that the covariance matrix is E (up to a factor of 3 due to the use of the Student's t_3 distribution).

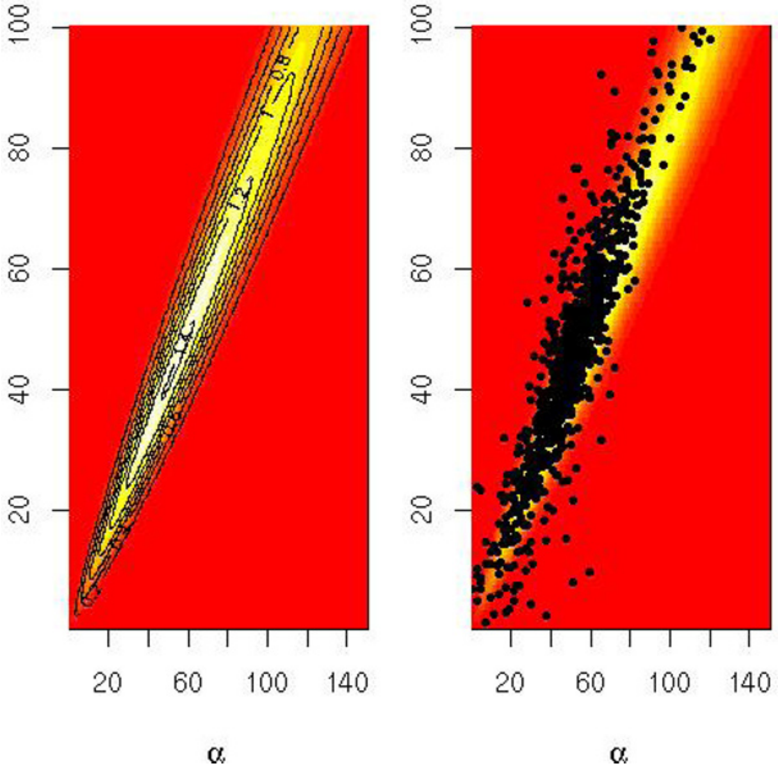

Fig. 3.5. *(left)* Representation of the posterior distribution $\pi(\alpha, \beta|x)$ on the parameters of a $\mathcal{B}(\alpha, \beta)$ distribution for $x = 0.6$. *(right)* Superposition of a sample of 10^3 points from a Student's t $\mathcal{T}(3, \mu, \Sigma)$ distribution used as an importance function.

If the quantity of interest is the marginal likelihood, as in Bayesian model comparison (Robert, 2001),

$$\begin{aligned} m(x) &= \int_{\mathbb{R}^2_+} f(x|\alpha, \beta)\, \pi(\alpha, \beta)\, \mathrm{d}\alpha \mathrm{d}\beta \\ &= \frac{\int_{\mathbb{R}^2_+} \left\{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right\}^{\lambda+1} [x x_0]^{\alpha} [(1-x) y_0]^{\beta}\, \mathrm{d}\alpha \mathrm{d}\beta}{x(1-x) \int_{\mathbb{R}^2_+} \left\{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right\}^{\lambda} x_0^{\alpha} y_0^{\beta}\, \mathrm{d}\alpha \mathrm{d}\beta}, \end{aligned}$$

we need to approximate both integrals and the same t sample can be used for both since the fit is equally reasonable on the prior surface. This approximation

$$\hat{m}(x) = \sum_{i=1}^{n} \left\{\frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)}\right\}^{\lambda+1} [x x_0]^{\alpha_i} [(1-x) y_0]^{\beta_i} / g(\alpha_i, \beta_i) \Big/$$

$$\sum_{i=1}^{n} \left\{ \frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \right\}^{\lambda} x_0^{\alpha_i} y_0^{\beta_i} / g(\alpha_i, \beta_i) \,, \tag{3.5}$$

where $(\alpha_i, \beta_i)_{1 \leq i \leq n}$ are n iid realizations from g, is straightforward to implement in R:

```
> ine=apply(y,1,min)
> y=y[ine>0,]
> x=x[ine>0,]
> normx=sqrt(x[,1]^2+x[,2]^2)
> f=function(a) exp(2*(lgamma(a[,1]+a[,2])-lgamma(a[,1])
+    -lgamma(a[,2]))+a[,1]*log(.3)+a[,2]*log(.2))
> h=function(a) exp(1*(lgamma(a[,1]+a[,2])-lgamma(a[,1])
+    -lgamma(a[,2]))+a[,1]*log(.5)+a[,2]*log(.5))
> den=dt(normx,3)
> mean(f(y)/den)/mean(h(y)/den)
[1] 0.1361185
```

Our approximation of the marginal likelihood, based on those simulations is thus 0.1361. Similarly, the posterior expectations of the parameters α and β are obtained by

```
> mean(y[,1]*apply(y,1,f)/den)/mean(apply(y,1,h)/den)
[1] 19.33745
> mean(y[,2]*apply(y,1,f)/den)/mean(apply(y,1,h)/den)
[1] 16.54468
```

i.e., are approximately equal to 19.34 and 16.54, respectively. ◀

3.3.2 Sampling importance resampling

The importance sampling technique does more than approximate integrals, though, since it provides an alternative way to simulate from complex distributions. Recall that the method produces a sample $X_1, \ldots, X_n$ simulated from g along with its importance weights $f(X_i)/g(X_i)$. This sample can then be recycled by multinomial resampling into a sample that is (almost) from f.

Indeed, if we could sample with replacement from the weighted population $\{X_1, \ldots, X_n\}$, picking X_i with probability $f(X_i)/ng(X_i)$, we would get an outcome X^* distributed as

$$\begin{aligned} \Pr(X^* \in A) &= \sum_{i=1}^{n} \Pr(X^* \in A \text{ and } X^* = X_i) \\ &= \int_A \frac{f(x)}{g(x)}\, g(x)\, \mathrm{d}x = \int_A f(x)\, \mathrm{d}x, \end{aligned}$$

and the method would then produce an exact simulation from f! Unfortunately, the probabilities $f(X_i)/ng(X_i)$ do not sum up to 1 (worse, some may even be larger than 1) and need to be renormalized into $(i = 1, \ldots, n)$

$$\omega_i = \frac{1}{n}\left\{f(X_i)/g(X_i)\right\} \Big/ \frac{1}{n}\sum_{j=1}^{n}\left\{f(X_j)/g(X_j)\right\}. \tag{3.6}$$

While the denominator is converging almost surely to one, the renormalization induces a bias in the distribution of the resampled values. Nonetheless, for large sample sizes, this bias is negligible, and we will thus use multinomial resampling (or an improved version; see Exercises 3.6 and 3.12) to approximate samples generated from f.

Exercise 3.6 Given an importance sample $(X_i, f(X_i)/g(X_i))$, show that if ω_i has a Poisson distribution $\omega_i \sim \mathcal{P}(f(X_i)/g(X_i))$, the estimator

$$\frac{1}{n}\sum_{i=1}^{n}\omega_i h(x_i)$$

is unbiased. Deduce that the sample derived by this sampling mechanism is marginally distributed from f.

The sole difficulty with the solution proposed in Exercise 3.6 is that the samples thus produced have a random size due to the random replications of each value in the weighted sample, ranging from 0 to ∞. While the setting where either f or g is missing a normalizing constant can be handled as well by replacing f/g by $\alpha f/g$, the impact on the final sample size is even harder to fathom (see Exercises 3.10 and 3.12).

The use of the renormalized weights in the importance sampling estimator produces the *self-normalized importance sampling estimator*

$$\sum_{i=1}^{n} h(X_i)\, f(X_i)/g(X_i) \Big/ \sum_{j=1}^{n}\left\{f(X_j)/g(X_j)\right\}, \tag{3.7}$$

which can also be used in situations when either f or g are missing a normalizing constant. The denominator of (3.6) is then estimating the missing constant(s) as well. (This is for instance the case in Example 3.6: The missing normalizing constant of the prior is estimated by `mean(apply(y,1,h)/den)` in the code above.)

↯ The importance weights only provide a *relative* assessment of the adequacy of the simulated sample to the target density in that they indicate how much more likely X_i is to be simulated from f compared with X_j, but they should not be overinterpreted. For instance, if X_i has a self-normalized weight that is close to 1, it does not mean that this value is very likely to be generated from f but simply that it is much more likely than the other simulated values! Even when the fit between f and g is very poor, this occurrence is bound to happen. Therefore, more trustworthy indicators must be used to judge the adequacy of g against f.

Example 3.7. (Continuation of Example 3.6) The validity of the approximation (3.5) of the marginal likelihood, (i.e., the convergence of the importance sampling solution) can be assessed by graphical means as follows:

```
> par(mfrow=c(2,2),mar=c(4,4,2,1))
> weit=(apply(y,1,f)/den)/mean(apply(y,1,h)/den)
> image(aa,bb,post,xlab=expression(alpha),
+       ylab=expression(beta))
> points(y[sample(1:length(weit),10^3,rep=T,pro=weit),],
+ cex=.6,pch=19)
> boxplot(weit,ylab="importance weight")
> plot(cumsum(weit)/(1:length(weit)),type="l",
+      xlab="simulations", ylab="marginal likelihood")
> boot=matrix(0,ncol=length(weit),nrow=100)
> for (t in 1:100)
+    boot[t,]=cumsum(sample(weit))/(1:length(weit))
> uppa=apply(boot,2,quantile,.95)
> lowa=apply(boot,2,quantile,.05)
> polygon(c(1:length(weit),length(weit):1),c(uppa,rev(lowa)),
+        col="gold")
> lines(cumsum(weit)/(1:length(weit)),lwd=2)
> plot(cumsum(weit)^2/cumsum(weit^2),type="l",
+     xlab="simulations", ylab="Effective sample size",lwd=2)
```

We will not discuss in detail all those indicators, as some are explained in the next chapter. The upper left graph in Figure 3.6 shows that the sample weighted using the importance weight $\pi(\alpha_i, \beta_i|x)/g(\alpha_i, \beta_i)$ produces a fair rendering of a sample from $\pi(\alpha, \beta|x)$. The resampled points do not degenerate in a few points but instead cover, with high density, the correct range for the target distribution (compare it with the right-hand side of Figure 3.5). The upper right graph gives a representation of the spread of the importance weights. While there exist simulations with much higher weights than the others, the spread of the weight is not so extreme as to signify a degeneracy of the method. For instance, the highest reweighted point only represents 1% of the whole sample. The lower left graph represents the convergence of the estimator $\hat{m}(x)$ as n increases. The colored band surrounding the sequence is a bootstrap rendering (Section 1.5) of the variability of this estimator that mimics the confidence band represented in Figure 3.3 at a low cost. The lower right curve is representing the efficiency loss in using importance sampling by the effective sampling size (see Section 4.4),

$$\left\{\sum_{i=1}^{n} \pi(\alpha_i, \beta_i|x)/g(\alpha_i, \beta_i)\right\}^2 \Bigg/ \sum_{i=1}^{n} \left\{\pi(\alpha_i, \beta_i|x)/g(\alpha_i, \beta_i)\right\}^2 ,$$

which should be equal to n, if the (α_i, β_i)'s are generated from the posterior. The current graph shows that the sample produced has an efficiency of about 6%. We will further consider this indicator in Section 4.2. ◀

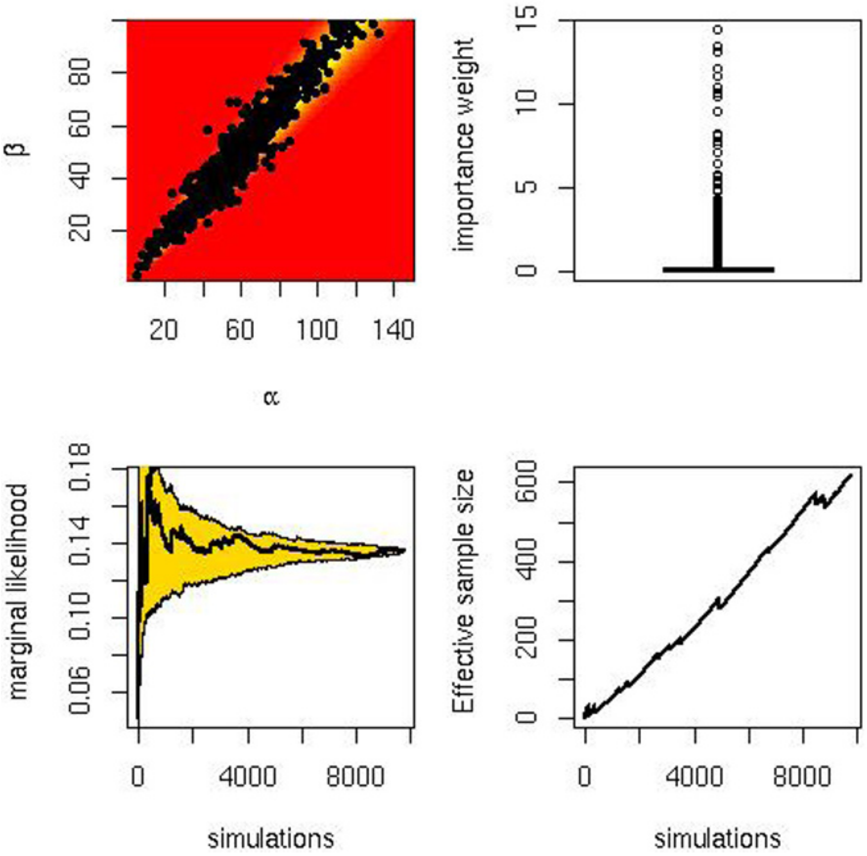

Fig. 3.6. *(upper left)* Superposition of 10^3 resampled points over the posterior distribution $\pi(\alpha, \beta|x)$ on the parameters of a $\mathcal{B}(\alpha, \beta)$ distribution for $x = 0.6$. *(upper right)* Boxplot of the importance weights. *(upper right)* Convergence of the approximation $\hat{m}(x)$ and bootstrap rendering of its variability. *(upper right)* Evolution of the effective sample size.

3.3.3 Selection of the importance function

The versatility of the importance sampling technique is high, but the downside of this versatility is that a poor choice of the importance function g may produce very poor outcomes. While the optimal choice of g is more of a theoretical exercise (see Rubinstein, 1981, or Robert and Casella, 2004, Theorem 3.12) than anything useful, an issue of primary relevance is to consider the variance of the resulting estimator (3.3) when judging the adequacy of the corresponding importance function g.

Indeed, while (3.4) does converge almost surely to (3.1), given that the expectation (3.1) exists, the variance of this estimator is finite only when the expectation

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

is finite. While not exactly prohibiting importance functions with tails lighter than those of f that lead to unbounded ratios f/g, this condition stresses that those functions are much more likely to lead to infinite variance estimators.

Before discussing this issue in more detail, let us consider a simple example to illustrate the disastrous impact of an infinite variance estimator.

Example 3.8. A simple setting where infinite variance occurs is when using a $\mathcal{N}(0,1)$ normal importance function aimed at a Cauchy $\mathcal{C}(0,1)$ target. The ratio $f(x)/g(x) \propto \exp(x^2/2)/(1+x^2)$ is then explosive in that even moderately high values of x get very large importance weights. If you run the code

```
> x=rnorm(10^6)
> wein=dcauchy(x)/dnorm(x)
> boxplot(wein/sum(wein))
> plot(cumsum(wein*(x>2)*(x<6))/cumsum(wein),type="l")
> abline(a=pcauchy(6)-pcauchy(2),b=0,col="sienna")
```

a few times, you should see graphs like the one in Figure 3.7 occurring, namely patterns with huge jumps in the cumulated average, even with a large number of terms in the average. The jumps happen at values of the simulations for which $\exp(x^2/2)/(1+x^2)$ is large, which means when x is large. The reason for this phenomenon is that since those values are rare under the normal importance distribution (meaning rarer than under the Cauchy target), they need to compensate for their rarity by taking high weights. For instance, in Figure 3.7, the major jump is due to a value of $x = 5.49$ associated with a normalized weight of $\omega_i = 0.094$. It means that this single point has a weight of about 10% in a sample of a million points! Obviously, it is impossible to trust the outcome of this simulation since the sample size is then irrelevant (i.e., most simulated values have a negligible weight). ◀

When the ratio f/g is unbounded, the importance weights $f(x_j)/g(x_j)$ often vary widely, giving too much importance to a few values x_j and thus degrading the efficiency of the estimator (3.4). As in the example above, it may happen that the estimate abruptly changes from one iteration to the next one, even after many iterations, due to a single simulation. Conversely, importance distributions g with thicker tails than f ensure that the behavior of the ratio f/g is not the cause of the divergence of $\mathbb{E}_f[h^2(X)f(X)/g(X)]$.

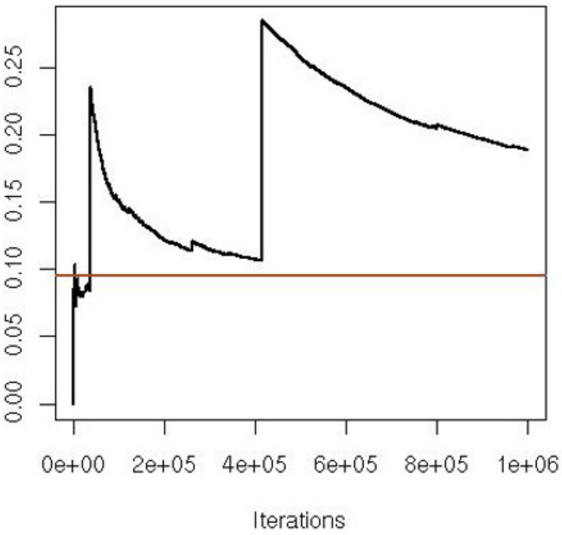

Fig. 3.7. Evolution of the importance sampling estimator of the probability $P(2 \leq Z \leq 6)$ against iteration indices, when Z is distributed from a Cauchy distribution and the importance function is normal. The straight line is the exact value, 0.095.

Using thicker-tailed importance sampling proposal distributions is almost a "must" when considering the approximation of functions h such that (3.1) exists but $\mathbb{E}_f[h^2(X)]$ does not. In such cases, using regular Monte Carlo is not possible, since the empirical average of the $h(X_i)$'s then has no variance.

Exercise 3.7 When f is a $\mathcal{T}_\nu$ distribution, show that the variance of the importance sampling estimator associated with an importance function g and the integrand $h(x) = \sqrt{x/(1-x)}$ is infinite for all g's such that $g(1) < \infty$. Discuss a sufficient condition on g for the variance to be finite. (*Hint:* See Example 3.9.)

↯ The self-normalized estimator (3.7) requires the same condition as in the nonnormalized case for the variance to be finite. But, as detailed in Chapter 4, the expression of the variance is not available in closed form and needs to be approximated by Monte Carlo methods.

As a generic recommendation, at this stage we thus suggest looking for distributions g for which $|h|f/g$ is almost constant or at least enjoys a controlled tail behavior, since this is more likely to produce estimators with a finite variance.

A basic requirement for functions h with restricted supports as in Example 3.5 is that g adopt the same support as h unless this is prevented by the complexity of h. Obviously, this requires fitting a new importance function for each integrand h to be considered, but this is the price to pay to achieve (much) more efficiency, as shown by Example 3.5.

Given that importance sampling primarily applies in settings where f is not easy to study, this constraint on the tails of f is often not easy to implement, especially when the dimensionality is high. A generic solution nonetheless exists based on the artificial incorporation of a fat tail component in the importance function g. This solution is called *defensive sampling* by Hesterberg (1995) and can be achieved by substituting a mixture density for the density g,

$$\rho g(x) + (1-\rho)\ell(x), \qquad 0 < \rho < 1\,, \tag{3.8}$$

where ρ is close to 1 and the density ℓ is chosen for its heavy tails (for instance, a Cauchy or a Pareto distribution), not necessarily in conjunction with the problem at hand.

Assuming g is provided by the setting, choosing the heavy-tailed function ℓ is potentially delicate. In the special case of Bayesian inference, when the target distribution f is the posterior distribution, it is, however, natural to choose the prior if proper. Indeed, this function has heavier tails than f by construction and is usually a standard distribution that is easy to simulate. Using the prior as the main importance function g would not make sense because of the waste induced (assuming the data are informative). But using it as a stabilizing factor does make sense.

Exercise 3.8 (Smith and Gelfand, 1992) Show that when evaluating an integral based on a posterior distribution

$$\pi(\theta|x) \propto \pi(\theta)\ell(\theta|x),$$

where π is the prior distribution and ℓ the likelihood function, the prior distribution can always be used as an instrumental distribution.

a. Show that the variance of the weight is finite when the likelihood is bounded.
b. Compare the previous choice with choosing $\ell(\theta|x)$ as the instrumental distribution when the likelihood is proportional to a density. (*Hint*: Consider the case of exponential families.)
c. Discuss the drawbacks of this (these) choice(s) in specific settings.
d. Show that a mixture between both instrumental distributions can ease some of the drawbacks.

From an operational point of view, generating from (3.8) means that the observations are generated with probability ρ from g and with probability $1-\rho$ from ℓ, using a code like

```
> mix=function(n=1,p=0.5){
+   m=rbinom(1,size,pro=p)
+   c(simg(m),siml(n-m))}
```

if `simg` and `siml` denote generators from g and ℓ, respectively. We stress that the fact that some points are generated from g and others from ℓ does not impact the importance weight in that it is equal to $f(x)/\{\rho g(x)+(1-\rho)\ell(x)\}$ for *all* generated values.

By construction, the importance sampling estimator integrates out the uniform variable used to decide between g and ℓ. Conditioning on this uniform variable would both induce more variability and destroy the purpose of using a mixture by dividing once again by $g(x)$ in the importance weight. (We discuss in detail such a marginalization perspective in the next chapter, in Section 4.6, where uniform variables involved in the simulation are integrated out in the estimator.)

Note that the selection of a random number of simulations from g and ℓ is *in fine* unnecessary, however, since generating exactly ρn x_i's from g and $(1-\rho)n$ y_i's from ℓ produces an unbiased estimator (under the assumption that ρn is an integer) in the sense that the importance sampling estimator

$$\frac{1}{n}\sum_{i=1}^{\rho n} h(x_i)\frac{f(x_i)}{\rho g(x_i)+(1-\rho)\ell(x_i)}+\frac{1}{n}\sum_{i=1}^{(1-\rho)n} h(y_i)\frac{f(y_i)}{\rho g(y_i)+(1-\rho)\ell(y_i)}$$

has a global (if not termwise) expectation equal to $\mathbb{E}_f[h(X)]$ (see Owen and Zhou, 2000, for more details). Thus, simulating a fixed number of points from each distribution is both valid and interesting in that it completely eliminates the variability due to the binomial sampling above.

Example 3.9. As indicated in Exercise 3.7, the computation of the integral

$$\int_1^\infty \sqrt{\frac{x}{x-1}}\,t_2(x)\,\mathrm{d}x=\int_1^\infty \sqrt{\frac{x}{x-1}}\,\frac{\Gamma(3/2)/\sqrt{2\pi}}{(1+x^2/2)^{3/2}}\,\mathrm{d}x \tag{3.9}$$

is delicate because the function $h(x)=\sqrt{1/(x-1)}$ is not square-integrable and therefore using simulations from the $\mathcal{T}_2$ distribution will produce an infinite variance for the Monte Carlo estimator of the integral.

This feature means that a mixture of the $\mathcal{T}_2$ density with a well-behaved ℓ is required. To achieve integrability of $h^2(x)f(x)/\ell(x)$ calls for ℓ to be divergent in $x=1$ and for ℓ to decrease faster than x^5 when x goes to infinity. Those boundary conditions suggest that

$$\ell(x) \propto \frac{1}{\sqrt{x-1}}\,\frac{1}{x^{3/2}}\,\mathbb{I}_{x>1}$$

(which is defined up to a constant) is an acceptable density. To characterize this density, you can check that

$$\begin{aligned}\int_1^y \frac{\mathrm{d}x}{\sqrt{x-1}x^{3/2}} &= \int_0^{y-1} \frac{\mathrm{d}w}{\sqrt{w}(w+1)^{3/2}}\\ &= \int_0^{\sqrt{y-1}} \frac{2\mathrm{d}\omega}{(\omega^2+1)^{3/2}}\\ &= \int_0^{\sqrt{2(y-1)}} \frac{2\mathrm{d}t}{(1+t^2/2)^{3/2}}\,.\end{aligned}$$

This implies that $\ell(x)$ corresponds to the density of $(1+T^2/2)$ when $T \sim \mathcal{T}_3$, namely

$$\ell(x) = \frac{\sqrt{2}\,\Gamma(3/2)/\sqrt{2\pi}}{\sqrt{x-1}x^{3/2}}\,\mathbb{I}_{(1,\infty)}(x)\,.$$

(You can verify that this is the correct normalizing constant by running `integrate`.)

The comparison of defensive sampling with the original importance sampler thus consists in adding a small sample from ℓ to the original sample from $g = f$:

```
> sam1=rt(.95*10^4,df=2)
> sam2=1+.5*rt(.05*10^4,df=2)^2
> sam=sample(c(sam1,sam2),.95*10^4)
> weit=dt(sam,df=2)/(0.95*dt(sam,df=2)+.05*(sam>0)*
+ dt(sqrt(2*abs(sam-1)),df=2)*sqrt(2)/sqrt(abs(sam-1)))
> plot(cumsum(h(sam1))/(1:length(sam1)),ty="l")
> lines(cumsum(weit*h(sam))/1:length(sam1),col="blue")
```

Note that simulations that are smaller than 1 get a weight equal to $1/.95$ in the defensive version since $\ell(x) = 0$ for $x \leq 1$. As in Example 3.8 and Figure 3.7, the original sample may exhibit important jumps in the cumulated average that are warnings of infinite variance problems. The defensive sampling solution produces a much more stable evaluation of the integral. In alternative simulations, both convergence graphs may also be quite similar if no simulation is close enough to 1 to induce a large value of $1/\sqrt{x-1}$. In Figure 3.8, we selected one occurrence of discrepancy between both samples where defensive sampling brings a clear element of stabilization. ◀

The example above clearly illustrates the impact of defensive sampling when the heavy-tailed component of the mixture is somehow related to the problem at hand. Generic choices of ℓ often lead to less efficient solutions, even when they ensure a finite variance for the Monte Carlo estimator.

Example 3.10. This example considers a probit model from a Bayesian point of view. We recall that the probit model is a particular case of a generalized linear

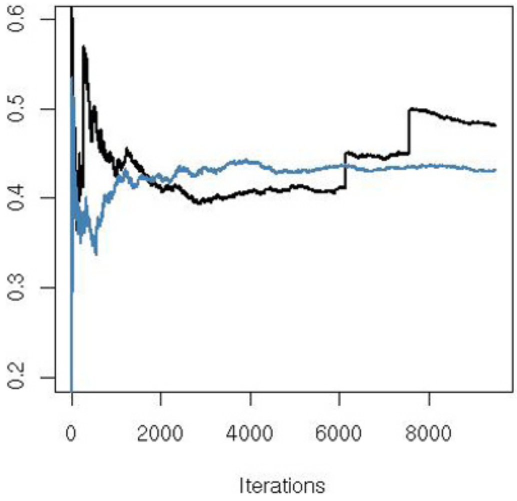

Fig. 3.8. Convergence of two estimators of the integral (3.9) of Example 3.9 based on a sample from $\mathcal{T}_2$ (dark line) and a defensive version (grey line).

model where the observables y are binary variables, taking values 0 and 1, and the covariates are vectors $x \in \mathbb{R}^p$ such that

$$\mathsf{Pr}(y=1|x) = 1 - \mathsf{Pr}(y=0|x) = \Phi(x^\mathsf{T}\beta)\,, \quad \beta \in \mathbb{R}^p\,.$$

Data for this model can easily be simulated, but we use instead an `R` dataset called `Pima.tr` that is available in the library `MASS`. This dataset surveys 200 Pima Indian women in terms of presence or absence of diabetes, `Pima.tr$type` (this is the binary variable y to explain) and various physiological covariates. For illustration purposes, we only consider the body mass index covariate, `Pima.tr$bmi`, with an intercept.

A standard GLM estimation of the model is provided by

```
> glm(type~bmi,data=Pima.tr,family=binomial(link="probit"))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.54303    0.54211  -4.691 2.72e-06 ***
bmi          0.06479    0.01615   4.011 6.05e-05 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

which indicates that the body mass index covariate has a significant impact on the possible presence of diabetes.

From a Bayesian perspective, we introduce a vague prior on $\beta = (\beta_1, \beta_2)$ that is a normal $\mathcal{N}(0, 100)$ distribution. The posterior distribution on β is then the product of this essentially flat prior by the likelihood, which can be defined as

```
like=function(beda){
  mia=mean(Pima.tr$bmi)
  prod(pnorm(beda[1]+(Pima.tr$bm[Pima.tr$t=="Yes"]-
      mia)*beda[2]))*
  prod(pnorm(-beda[1]-(Pima.tr$bm[Pima.tr$t=="No"]
     -mia)*beda[2]))/exp(sum(beda^2)/200)
  }
```

Experimenting with the image function and this likelihood indicates that the central part of the likelihood is located near the maximum likelihood estimator (MLE) with a range of $-.6/-.3$ for the intercept β_1 and a range of $0.04/0.09$ for β_2. Using a normal proposal centered at the MLE with a diagonal covariance matrix corresponding to the estimate provided by glm is a natural choice for g, even though this does not guarantee a finite variance for all purposes. However, implementing this idea with

```
> sim=cbind(rnorm(10^3,mean=-.4,sd=.04),
+           rnorm(10^3,mean=.065,sd=.005))
> weit=apply(sim,1,post)/(dnorm(sim[,1],mean=-.4,sd=.04)*
+                          dnorm(sim[,2],mean=.065,sd=.005))
```

shows that the importance weights are rather uneven, if not degenerate (you can check using boxplot(weit) for instance). A representation of 10^4 resampled points based on those weights in Figure 3.9 confirms this pattern.

In order to evaluate the (low) impact of a defensive sampling implementation, we also create an importance sample that includes simulations from the prior with probability $.05$ by modifying the above into

```
> sim=rbind(sim[1:(.95*10^3),],cbind(rnorm(.05*10^3,sd=10),
+           rnorm(.05*10^3,sd=10)))
> weit=apply(sim,1,post)/(.95*dnorm(sim[,1],m=-.4,sd=.081)*
+   dnorm(sim[,2],m=0.065,sd=.01)+.05*dnorm(sim[,1],sd=10)*
+   dnorm(sim[,2],sd=10))
```

The difference in efficiency is not visible, though. When use the effective sample size criterion (defined in Section 4.4), the difference on 10^3 simulations is an effective sample size of 302 for the normal sample versus an effective sample size of 283 for the defensive one. The estimates of β produced by both methods are $(-0.452, .0653)$ and $(-0.452, .0652)$, respectively. (Note the proximity with the MLE if we incorporate the mean of Pima.tr$bmi.) The reason for this strong similarity is that the additional term in the denominator due to the inclusion of the prior density is mostly zero. ◀

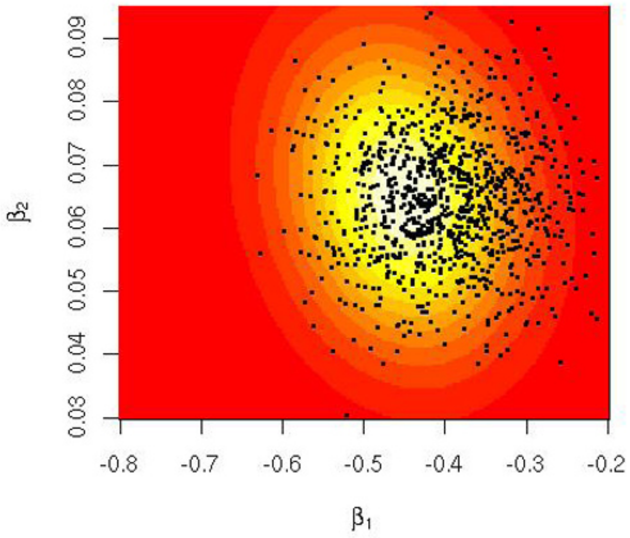

Fig. 3.9. Posterior distribution of the parameter (β_1, β_2) for the regression of diabetes on body mass index in the Pima.tr dataset with resampled values from a normal proposal superimposed.

3.4 Additional exercises

Exercise 3.9 For the same estimator $\delta(x)$ as in Exercise 3.1:

a. Build an Accept–Reject algorithm based on a Cauchy candidate to generate a sample from the posterior distribution and then deduce the estimator.
b. Design a computer experiment to compare Monte Carlo errors when using (i) the same random variables θ_i in the numerator and denominator or (ii) different random variables.

Exercise 3.10 Consider the same question as in Exercise 3.6 when

$$\omega_i = \lfloor f(X_i)/g(X_i) \rfloor + \delta_i, \text{ with } \delta_i \sim \mathcal{B}\text{in}\{1, f(X_i)/g(X_i) - \lfloor f(X_i)/g(X_i) \rfloor\}$$

and $\lfloor x \rfloor$ denoting the integer part of x. Show that there also exists an unbiased estimator based on the replacement of $f(X_i)/g(X_i)$ with $\alpha\, f(X_i)/g(X_i)$ for any $\alpha > 0$.

Exercise 3.11 Referring to Example 3.5:

a. Show that to simulate $Y \sim \mathcal{E}xp^+(a, 1)$, an exponential distribution left truncated at a, we can simulate $X \sim \mathcal{E}xp(1)$ and take $Y = a + X$.
b. Use this method to calculate the probability that a χ^2_3 random variable is greater than 25 and that a t_5 random variable is greater than 50.

c. Explore the gain in efficiency from this method. Take $a = 4.5$ in part (a) and run an experiment to determine how many random variables would be needed to calculate $P(Z > 4.5)$ to the same accuracy obtained from using 100 random variables in an importance sampler.

Exercise 3.12 Show that if

$$\omega_i \sim \begin{cases} \mathcal{B}\text{in}(1, f(X_i)/g(X_i)) & \text{if } f(X_i)/g(X_i) < 1\,, \\ \mathcal{G}eo(g(X_i)/f(X_i)) & \text{otherwise}\,, \end{cases}$$

the estimator $\frac{1}{n}\sum_{i=1}^{n} \omega_i h(x_i)$ is also unbiased.

Exercise 3.13 (Ó Ruanaidh and Fitzgerald, 1996) For simulating random variables from the density $f(x) \propto \exp\{-x^2\sqrt{x}\}[\sin(x)]^2$, $0 < x < \infty$, compare the following choices of instrumental densities on $\mathbb{R}$:

$$g_1(x) = \frac{1}{2}e^{-|x|}, \quad g_2(x) = \frac{1}{2\pi}\frac{1}{1+x^2/4}, \quad g_3(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}\,.$$

For each of the instrumental densities, estimate the number M of simulations needed to obtain three digits of accuracy in estimating $\mathbb{E}_f[X]$. Deduce from the acceptance rate an estimator of the normalizing constant of f for each of the instrumental densities.

Exercise 3.14 When a cdf $F(x)$ has a tail power of α (i.e., when $1 - F(x) \propto x^{-\alpha}$ for x large enough):

a. Show that $\mathbb{E}[X|X > K] = K\alpha/(\alpha - 1)$ for K large enough. Discuss the existence of this expectation as a function of α.
b. Derive an estimate of $\mathbb{E}[X|X > K]$ based on a sample from F.
c. Evaluate the stability of this estimate as a function of K when F is a Pareto $\mathcal{P}(2)$, $\mathcal{P}(3)$, $\mathcal{P}(4)$ distribution (see Exercise 2.13).

Exercise 3.15 (Gelfand and Dey, 1994) Consider a density function $f(x|\theta)$ and a prior distribution $\pi(\theta)$ such that the marginal $m(x) = \int_\Theta f(x|\theta)\pi(\theta)\mathrm{d}\theta$ is finite a.e. The marginal density is of use in the comparison of models since it appears in the Bayes factor (see Robert, 2001).

a. Give the general shape of an importance sampling approximation of m.
b. Detail this approximation when the importance function is the posterior distribution and when the normalizing constant is unknown.
c. Show that, for a proper density τ,

$$m(x)^{-1} = \int_\Theta \frac{\tau(\theta)}{f(x|\theta)\pi(\theta)}\pi(\theta|x)\mathrm{d}\theta\,,$$

and deduce that when the θ_i^*'s are generated from the posterior,

$$\hat{m}(x) = \left\{\frac{1}{T}\sum_{t=1}^{T}\tau(\theta_i^*)\Big/ f(x|\theta_i^*)\pi(\theta_i^*)\right\}^{-1}$$

is another importance sampling estimator of $m(x)$.

Exercise 3.16 Given a real importance sample $X_1, \ldots, X_n$ with importance function g and target density f:

a. Show that the sum of the weights $\omega_i = f(X_i)/g(X_i)$ is only equal to n in expectation and deduce that the weights need to be renormalized even when both densities have known normalizing constants.
b. Assuming that the weights ω_i have been renormalized to sum to one, we sample, with replacement, n points $\tilde{X}_j$ from the X_i's using those weights. Show that the $\tilde{X}_j$'s satisfy

$$\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^{n} h(\tilde{X}_j)\right] = \mathbb{E}\left[\sum_{i=1}^{n} \omega_i h(X_i)\right] .$$

c. Deduce that if the formula above is satisfied for $\omega_i = f(X_i)/g(X_i)$ instead, the empirical distribution associated with the $\tilde{X}_j$'s is unbiased.

Exercise 3.17 Monte Carlo marginalization is a technique for calculating a marginal density when simulating from a joint density. Let $(X_i, Y_i) \sim f_{XY}(x, y)$, independent, and the corresponding marginal distribution $f_X(x) = \int f_{XY}(x, y)\mathrm{d}y$.

a. Let $w(x)$ be an arbitrary density. Show that

$$\lim_n \frac{1}{n}\sum_{i=1}^{n} \frac{f_{XY}(x^*, y_i)w(x_i)}{f_{XY}(x_i, y_i)} = \int\int \frac{f_{XY}(x^*, y)w(x)}{f_{XY}(x, y)} f_{XY}(x, y)\mathrm{d}x\mathrm{d}y = f_X(x^*) ,$$

which provides a Monte Carlo estimate of f_X, the marginal distribution of X, when the joint distribution is only known up to a constant.
b. Let $X|Y = y \sim \mathcal{G}(y, 1)$ and $Y \sim \mathcal{E}xp(1)$. Use the technique above to plot the marginal density of X. Compare this with the exact marginal.
c. Show that choosing $w(x) = f_X(x)$ works to produce the marginal distribution and that it is optimal in the sense of the variance of the resulting estimator.

Exercise 3.18 Given the *Gumbel distribution*, with density $f(x) = \exp\{x - \exp(x)\}$ over the real line, we are interested in comparing the variability of regular importance sampling based on a normal importance function with the variability of the corresponding self-normalized version of (3.7).

a. Show that the expectation of $\exp(X)$ is well-defined for the Gumbel distribution.
b. Create a matrix x of normal simulations with 100 columns using rnorm(100*10^4) and deduce the importance weights we.
c. Deduce the regular and the self-normalized sequences of estimators of $\mathbb{E}[\exp(X)]$ by

```
> nore=apply(we*exp(x),2,cumsum)/(1:10^4)
> reno=apply(we*exp(x),2,cumsum)/apply(we,2,cumsum)
```

and plot the ranges of both sequences of estimates using polygon.

Exercise 3.19 (Berger et al., 1998) For a $p \times p$ positive-definite symmetric matrix Σ, consider the distribution

$$\pi(\theta) \propto \exp\left(-(\theta - \mu)^t \Sigma^{-1}(\theta - \mu)/2\right) \Big/ ||\theta||^{p-1} .$$

a. Show that the distribution is well-defined; that is, $\int_{\mathbb{R}^p} \pi(\theta)\mathrm{d}\theta < \infty$.
b. Show that an importance sampling implementation based on the normal instrumental distribution $\mathcal{N}_p(\mu, \Sigma)$ is not satisfactory from both theoretical and practical points of view.
c. Examine the alternative based on a gamma distribution $\mathcal{G}(\alpha, \beta)$ on $\eta = ||\theta||^2$ and a uniform distribution on the angles.