Topics for Nov 11 and 13

36-711 Fall 2002

Outline

1. Numerical integration and standard Bayesian calculations.
   (a) Numerical integration to compute posterior probabilities, e.g. CI's. `bayes.s`
2. Sketch of some standard univariate numerical integration methods.
3. Sampling to produce pictures of posterior densities
   (a) Sampling generalities.
   (b) Exploring posterior distributions in Bayesian inference, using Monte Carlo. `sampling.s`

[later we will talk briefly about multivariate numerical integration]

1 Numerical integration and standard Bayesian calculations

Rarely will the anti-derivatives composed of elementary functions like x^k, e^x, $\sin(x)$, etc., that we learned in Calculus, be sufficient to compute definite integrals in real life.

We have already seen in the E-M algorithm that the integral

$$Q(\theta, \theta^k) = \int \log f(X|\theta) f(X|Y, \theta^k) dX$$

is a basic ingredient in maximum likelihood problems with missing data. In "simple" problems this can be evaluated in closed form, but usually it cannot and must somehow be approximated.

Another motivation for thinking about numerical integration methods is is in calculations for Bayesian inference. Mathematically similar integration problems arise in statistical inference for mixture models.

Given a (possibly multidimensional) parameter θ with prior distribution $\pi(\theta)$ and likelihood $L(y|\theta) = \prod_i L_i(y_i|\theta)$, Bayesian inference is based on functionals (i.e. numerical functions) of the posterior distribution

$$p(\theta|y) = \frac{L(y|\theta)\pi(\theta)}{\displaystyle\int L(y|\theta)\pi(\theta)d\theta}$$

Common functionals are

- The posterior probability of an interval $[a, b]$, $\displaystyle\int 1_{\{\theta \in [a,b]\}} p(\theta|y) d\theta$;

- The posterior mean $E[\Theta|y] = \int \theta p(\theta|y) d\theta$;
- The posterior variance $\mathrm{Var}\,(\Theta|y) = \int (\theta - E[\Theta|y])^2 p(\theta|y) d\theta$;
- The "entire" posterior density $p(\theta|y)$.

A direct attack on the first three quantities requires at the very least two integrals:

$$E[g(\theta)|y] = \int g(\theta) p(\theta|y) d\theta = \frac{\int g(\theta) L(y|\theta) \pi(\theta) d\theta}{\int L(y|\theta) \pi(\theta) d\theta} \tag{1}$$

The first part of our discussion will treat some ways of accomplishing the direct attack.

Paradoxically, we can often approximate $p(\theta|y)$ quite well without evaluating any integrals, by doing Monte Carlo simulation: we generate iid random draws from $p(\theta|y)$, and then draw a histogram or a kernel density estimate. This also permits an end-run around the integral calculations for the first three problems; for, by the Law of Large Numbers

$$\int g(\theta) p(\theta|y) d\theta \approx \frac{1}{n} \sum_{1}^{n} g(\theta_i)$$

where $\theta_1, \ldots, \theta_n$ is a random sample from $p(\theta|y)$.

Often it is difficult to sample directly from $p(\theta|y)$. The second part of our discussion will focus on elementary ways to indirectly generate a random sample from $p(\theta|y)$, by first obtaining a sample from $\pi(\theta)$ and then adjusting it somehow.

2 Sketches of some standard numerical integration methods

In addition to Monahan's text, the text *Elements of Statistical Computing* by Thisted is a good reference here.

2.1 Newton-Cotes rules

The simple methods one learns about in Calculus can actually be quite useful. We begin with a "nice" function $f(x)$ on a fixed interval (a, b) and an increment $h = (b-a)/n$, so that $a = a + 0h$, …, $x_i = a + ih$, …, $b = a + nh$.

- *Rectangle rule.*

$$\int_a^b f(x) dx \approx h \sum_{i=1}^{n-1} f(x_i)$$

  - *Open vs Closed:* Are the endpoints included in the calculation (maybe$x_i = a + (n + \frac{1}{2})$?). Most of the rules below have "open" and "closed" versions. We will mainly discuss the "closed" ones, since they are a little simpler to describe.

– *Extended/composite rule:* Add the answers

$$hf(x_i),$$

which are exact as long as $f(x)$ is *constant* on each subinterval.

- *Trapezoid rule.*

$$\int_a^b f(x)dx \approx \frac{h}{2}f(a) + h\sum_{i=1}^{n-1} f(x_i) + \frac{h}{2}f(b)$$

  – Can show the approximation is exact up to $O(f''(x^*)/n^2)$, so doubling the number of points *quadruples* the accuracy of the integral approximation.

  – *Extended/composite rule:* Add the answers

$$\frac{h}{2}\left[f(x_i) + f(x_{i+1})\right]$$

which are exact as long as $f(x)$ is *piecewise linear* on the intervals defined by $x_0, \ldots, x_n$.

- *A reasonably efficient trapezoid rule.*

```
trap.int _ function(fcn, a, b, maxcalls, releps) {
#
# Trapezoid rule, doubling number of points in the interval until
# estimated integral stops changing values.  Re-uses old function
# calls, so that no redundant calls to the integrand (fcn) are made.
#
#
        k _ ceiling(log(maxcalls - 1)/log(2))
        h _ b-a
        n _ 1
        I0 _ h*(fcn(b) - fcn(a))/2
        I1 _ (I0+1)/(releps + 1)
        j _ 1
        while ((j<=k) && (abs(I1-I0)>releps*abs(I0))) {
                tmp _ a + h/2
                p _ 0
                for (i in (0:(n-1))) {
                        p _ p + fcn(tmp + i*h)
                }
                I0 _ I1
                I1 _ (I0+h*p)/2
                h _ h/2
                n _ 2*n
                j _ j + 1
        }
        return(I1,j,k)
}
```

- *Simpson's Rule.*

$$\int_a^b f(x)dx \approx \frac{h}{3}\left[f(a) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + f(b)\right]$$

  - Can show the approximation is exact up to $O(f^{(4)}(x^*)/n^4)$: doubling the number of points increases the the accuracy of the integral approximation *by a factor of 16*.
  - *Extended/composite rule:* Add the answers

$$\frac{h}{2}\left[f(x_i) + 4f(x_{i+1}) + f(x_{i+2})\right]$$

    which are exact as long as $f(x)$ is *piecewise linear* on the each pair of adjacent subintervals defined by $x_0, \ldots, x_n$.

- *General Newton-Cotes methods.*
  - They evaluate the function on a grid of equally-spaced points.
  - They give the exact answer when integrating certain polynomials.

  Clearly this is just a linear system to solve:

$$\left.\begin{array}{rcl} w_1 f^0(x_1) + w_2 f^0(x_2) + \cdots + w_k f^0(x_k) & = & I_0 \\ w_1 f^1(x_1) + w_2 f^1(x_2) + \cdots + w_k f^1(x_k) & = & I_1 \\ w_1 f^2(x_1) + w_2 f^2(x_2) + \cdots + w_k f^2(x_k) & = & I_2 \\ & \vdots & \\ w_1 f^{k-1}(x_1) + w_1 f^{k-1}(x_2) + \cdots + w_k f^{k-1}(x_k) & = & I_{k-1} \end{array}\right\} \qquad (*)$$

  where $f^m(x) = x^k$, and $I_m = \int_a^b f^m(x)dx$.

- *Romberg integration.* Simpson's rule turns out to be equivalent to a weighted sum of two applications of the Trapezoid Rule. Let I_j be the result of the trapezoid rule on $2^j + 1$ points, and S_j the result of Simpson's rule on the same grid. Then one can readily show that

$$S_{j+1} = \frac{4}{3} I_{j+1} - \frac{1}{3} I_j$$

  and a careful error analysis shows that error terms in the trapezoid rules partially cancel, leaving an error that depends only on $f^{(4)}(x)$—in other words, *even though Simpson's involves only a quadratic approximation to $f(x)$, it yields results that are exact up to cubics!* **Romberg integration** methods are general methods for taking weighted sums of trapezoid rules to obtain cancellation of higher order terms like this.

2.2 Gaussian quadrature

Newton-Cotes rules for integrating polynomials up to order $k-1$ exactly are based on solving the linear equations $(*)$ for weights $w_1, \ldots, w_k$ at k equally spaced grid points $x_1, \ldots, x_k$,

$$\int_a^b f(x)dx = \sum_{i=0}^{n} w_i f(x_i)$$

and then extending by adding the integral approximations over adjacent groups of k points.

Gaussian quadrature methods allow not only the weights $w_1, \ldots, w_k$ to to be freely chosen, but also the locations $x_1, \ldots, x_k$ to be freely chosen. *We can then obtain integration rules that are exact for polynomials up to order* $2k - 1$.

Gaussian quadrature formulae can also be obtained by solving the system $(*)$, but now $(*)$ is a *nonlinear* system. The nonlinear systems become ill conditioned quickly as k grows, so that either (a) very high-precision arithmetic must be used; or (b) more accurate methods employing stable recurrences relating othrogonal polynomials to one another are used.

Usually the problem is generalized to deal with integrals of the form

$$\int_a^b f(x)w(x)dx$$

for fixed weight function $w(x)$ (which is equivalent to the density of a random variable with all moments finite).

Polynomials are chosen for the functions $f^j(x)$ in $(*)$ that are *orthogonal* with respect to the w-inner product:

$$\langle f^{j_1}, f^{j_2} \rangle_w = \int_a^b f^{j_1}(x) f^{j_2}(x) w(x) dx = \begin{cases} 0, \ j_1 \neq j_2 \\ c^{j_1}, \ j_1 = j_2 \end{cases}$$

Sometimes $c^j \equiv 1 \ \forall j$ is desired (orthonormal polynomials) and sometimes c^j is allowed to be free and instead a standardization like $f^j(1) \equiv 1 \ \forall j$ is chosen.

The solution of $(*)$ jointly for w_i and x_i is clearly tied up with the mathematics of the Hilbert space L^2_w of all functions f with finite integral $\langle f, f \rangle_w$. The main result is as follows:

> *Let $w(x)$ be a density on (a, b) for a random variable X with all moments finite, and let $f^j(x)$, $j = 1, \ldots, k$ be a set of orthonormal polynomials with respect to $\langle \cdot, \cdot \rangle_w$, and let c_j be the leading coefficient of each $f^j(x)$. Then:*
>
> - *The positions $a < x_1 < \cdots < x_k < b$ given by the roots of $f^k(x)$, and the weights $w_j > 0$ given by*
>
> $$w_j = -\frac{c_{k+1}}{c_k} \frac{1}{f^{k+1}(x_j)[f^k]'(x_j)}$$
>
> *sastify*
>
> $$\int_a^b f(x)w(x)dx = \sum_{i=0}^{n} w_i f(x_i)$$
>
> *whenever $f(x)$ is a polynomial of degree $2k - 1$ or less; and*
>
> - *If $f(x)$ is any $2k$-times differentiable function,*
>
> $$\int_a^b f(x)w(x)dx - \sum_{i=0}^{n} w_i f(x_i) = \frac{f^{(2k)}(x^*)}{(2k)! c_k^2}, \ , , , x \in (a, b)$$

A very common case is when $(a, b) = (-1, 1)$ and $w(x) \propto 1$, and we take the standardization $f^k(1) \equiv 1$. In that case the orthogonal polynomials are the *Legendre polynomials*, and the quadrature method is known as *Gauss-Legendre* or just *Gaussian* quadrature.

Tables of x_j's and w_j's can be found in Abramowitz and Stegun (1964), but it is almost always better to use a "canned" routine rather than rederive these or program them by cutting and pasting from a table. It is almost never worthwhile, computationally or from an error-introducing point of view, to generate the x_j's and w_j's.

Automatic/extension forms of Gaussian quadrature rules are harder to come by than for Newton-Cotes methods, because the $x_1, \ldots, x_k$ do not form a subset of locations for some larger k, in general. The most commonly used approximate extensions are for Gauss-Legendre quadrature; they are due to Kronrod, and they are implemented in the Splus (and R) function `integrate()`.

2.3 Numerical integration to compute posterior probabilities

Suppose we have obtained an interval $[a, b]$ which we hope is "95% credible" for θ, i.e. $P\{\theta \in [a, b]|y\} \approx 0.95$. How can we verify this? The integrals in equation (1) are now of the form

$$p(\theta \in [a,b]|y) = \frac{\int_a^b L(y|\theta)\pi(\theta)d\theta}{\int_{-\infty}^{\infty} L(y|\theta)\pi(\theta)d\theta}$$

These integrals present two immediate computational problems:

1. Computers don't deal well with large numbers like $\pm\infty$;
2. As n grows, $L(y|\theta)\pi(\theta) = \prod_{i=1}^n L_i(y_i|\theta)\pi(\theta)$ is likely to become quite small; computers don't deal well with small numbers either.

For Problem 1. In a few problems, the *support*, or range of θ for which $L(y|\theta)\pi(\theta) > 0$, will actually be a finite interval, in which case 1. is not a problem. More likely the support will be $(0, \infty)$, $(1, \infty)$, or $(-\infty, \infty)$. If so, then we can try to use the change of variables formula

$$\int_A^B f(\theta)d\theta = \int_{u(A)}^{u(B)} \frac{f(u^{-1}(t))}{u'(u^{-1}(t))}dt$$

where $f(\theta) \equiv L(y|\theta)\pi(\theta)$, and A or B may be infinite. We should choose the transformation $u(\theta)$ with two goals in mind:

- Make $(u(A), u(B))$ a nice interval. For example,

Transformation	Some $u(\theta)$'s
$(0,\infty) \to (0,1)$:	$e^{-\theta}$; $\theta/(1+\theta)$
$(1,\infty) \to (0,1)$:	$1/x$
$(-\infty,\infty) \to (0,1)$:	$\frac{2}{\pi}\tan^{-1}(\theta)$; $e^{\theta}/(1 + e^{\theta})$; other CDF's.

- Make $g(t) = \dfrac{f(u^{-1}(t))}{u'(u^{-1}(t))}$ nearly equal to some easy-to-integrate function (constant, polynomial, etc.)

This is difficult; for example to ask for $g(t) = constant$ is basically to ask for $f/u' = constant$, i.e. $u(\theta) = \int f', d\theta$

The next best thing to do is go for computational convenience, or match tails, center, or spread of f with u'.

An alternative to the transformation method is just to graph (something proportional to) $f(\theta)$ choose A' and B' from the graph, so that the integral from A' to B' is a good approximation to the integral from $-\infty$ to ∞.

For Problem 2. If $L(y|\theta)\pi(\theta)$ is small, the basic strategy for fixing this is to work with $c \cdot L(y|\theta)\pi(\theta)$ instead. The idea is that c will cancel out of the numerator and denominator integrals in equation (1). Choosing c can be difficult. Some first steps include:

- Graph $c \cdot L(y|\theta)\pi(\theta)$ as a function of θ for various trial values of c and see what happens
- Divide by $L(y|\hat{\theta})\pi(\hat{\theta})$ where $\hat{\theta}$ is the MLE or posterior mode (probably you want to work with the difference of the logs and then exponentiate).

3 Sampling to produce pictures of posterior densities; Monte Carlo integration

This section gives the background for sketches two simple sampling methods to produce samples from the posterior distribution in a Bayesian inference problem. Some books to consider for further details about Monte Carlo simulation are:

- Devroye, L. (1986). <u>Nonuniform random variate generation.</u> New York: Springer-Verlag.
- Hammersley, I. M. and Handscomb, D. C. (1964). <u>Monte Carlo methods.</u> London: Metheun.
- Ripley, B. D. (1987). <u>Stochastic simulation.</u> New York: Wiley.
- Gelman, A., Carlin, J. Stern, H. and Rubin, D. (1996?) *Bayesian data analysis*. New York: Chapman-Hall.

The connections with Bayesian posterior sampling come from the 1992 article by Smith and Gelfand that I am also handing out this week.

3.1 Sampling Generalities

Monte Carlo sampling typically has either (or both) of two goals:

1. To generate a sample from a distribution so that it may be plotted as a histogram, density estimate, etc.
2. To evaluate an integral, such as the mean, variance, etc., of a distribution.

For either of these goals the approach is the same:

Produce a sample that is (approximately) an i.i.d. sample from the distribution of interest.

Most programming languages used in statistical calculations (including, of course, SPLUS) have built-in pseudo-random number generators that generate observations that appear to be i.i.d. $Unif[0,1]$. In this section we'll list some common ways to generate observations from other distributions using these.

As a matter of notation, let

$$\begin{aligned} f(x) &= \text{the density of the } \textit{target} \text{ distribution} \\ F(x) &= \int_{-\infty}^{x} f(t)dt \;=\; P[X \le x] \;=\; \text{the CDF for } f(x) \\ g(y) &= \text{the density of a } \textit{convenient} \text{ distribution} \\ G(y) &= \int_{-\infty}^{y} g(t)dt \;=\; P[Y \le x] \;=\; \text{the CDF for } g(y) \end{aligned}$$

Then a generic way to satisfy each of the goals above is:

1. Generate $x_1, \ldots, x_n$ as i.i.d. observations from $f(x)$.
2. With the data generated in 1., compute

$$\frac{1}{n}\sum_{i=1}^{n} h(x_i) \;\approx\; \int_{-\infty}^{\infty} h(x)f(x)dx \;\equiv\; E_f[h(X)]$$

Probability Integral Transform You can verify with calculus the following two identities.

- If X is distributed following $f(x)$, then $U = F(X)$ is $Unif[0,1]$.
- If U is $Unif[0,1]$, then $F^{-1}(U)$ is distributed following $f(x)$.

If $F(x)$ can be obtained and inverted in closed form, this provides a way to generate from X. Relatively few distributions are easy to sample from in this way, however.

3.1.1 Clever Transformations

One often learns about these in math stat courses as "applications" of the "change of variables" formula in integral calculus. They actually turn out to be useful and convenient in some cases, though most of the distributions for which there are easy transformations are already coded into standard random number function (e.g. in SPLUS).

The basic idea is that if $h(x)$ is a strictly increasing function, then

$$Y \sim g(y) \Rightarrow X = h^{-1}(Y) \sim g(h(x))h'(x)$$

This is easily seen from the calculation

$$\begin{aligned} \frac{d}{dx}P[X \le x] &= \frac{d}{dx}P[h^{-1}(Y) \le x] \\ &= \frac{d}{dx}P[Y \le h(x)] \;=\; \frac{d}{dx}G(h(x)) \\ &= g(h(x))h'(x) \end{aligned}$$

(there are many details, such as what to do if $h(x)$ is strictly *decreasing* or even *noninvertible* but this is the basic idea.)

If you want to sample from $f(x)$ then the goal is to find $h(x)$ so that $f(x) = g(h(x))h'(x)$. Sometimes one finds h by accident, sometimes by experience, and sometimes by explicitly solving this equation. There is a similar but more complicated formula when x is multivariate. Here are some typical examples of transformations that lead to familiar distributions, in both the univariate and multivariate cases:

- $Y \sim Unif[0,1]$, $h^{-1}(y) = -\log y \Rightarrow X = h^{-1}(Y) \sim Exp(1)$.
- $Y \sim Unif[0,1] \Rightarrow X = \tan[\pi(Y - 1/2)] \sim$ Cauchy (this is also an instance of the probability integral transform).
- $Y_1, Y_2 \sim$ independent $Unif[0,1]$'s $\Rightarrow X_1 = \cos(2\pi Y_1)\sqrt{-2\log(Y_2)}$ and $X_2 = \sin(2\pi Y_1)\sqrt{-2\log(Y_2)}$ are independent $N(0,1)$'s.
- $Y \sim N(0,1) \Rightarrow Y^2 \sim$ Chi-squared, *df* $= 1$.
- $Y_1, \ldots, Y_n \sim$ independent $N(0,1)$'s $\Rightarrow X = Y_1^2 + Y_2^2 + \cdots + Y_n^2 \sim$ Chi-squared, *df* $= n$.

3.1.2 Rejection Method

More frequently the cdf $F(x)$ will not be readily computable nor invertable, and there will be no obvious transformation to produce $X \sim F(x)$ from uniform or other simpler random variables. The rejection method is a useful, generic approach to sampling from such problematic $F(x)$'s.

> **The Rejection Method.** *Suppose $f(x)$ is <u>proportional to</u> the density we <u>want</u> to sample from; and $g(x)$ is a density we <u>can</u> sample from. Also, suppose*
>
> $$f(x) \leq a \cdot g(x) \quad \text{for all } x$$
>
> *Then the following procedure produces observations from $f(x)$:*
>
> - *Generate Y from $g(x)$;*
> - *Generate U from $Unif[0, a \cdot g(x)]$;*
> - *If $U < f(Y)$, accept Y as an observation X from $f(x)$;*
>   *otherwise reject Y and start over.*

Useful exercises:

- Use calculus in 2D, to show that $P[Y \leq x | Y \text{ is accepted}] = F(x)$, justifying the method.
- Show that you can generate $X \sim N(0,1)$ by applying the rejection method to Cauchy observations $Y = \tan(\pi(U - 1/2))$. What is (the ***<u>best</u>***) a?
- Show (analytically) that it does ***<u>not</u>*** work to generate $X \sim Cauchy$ by applying the rejection method to $N(0,1)$ observations Y. *What goes wrong???*

The fact that one does not need to know the "normalizing constant" for $f(x)$ is a big advantage, especially in Bayesian problems, as we shall see below.

3.1.3 Control Variates

The basic idea of *control variates* is that if you want to estimate an integral with an i.i.d. sum,

$$E_f[h(X)] = \int h(x)f(x)dx \approx \hat{E}_n \equiv \frac{1}{n}\sum_{i=1}^{n} f(X_i)$$

and you can arrange to get another i.i.d sum that estimates 0,

$$0 \approx \hat{Z}_n$$

<u>*and*</u> Cov $(\hat{E}_n, \hat{Z}_n) < 0$, then $\hat{E}_n + \hat{Z}_n \approx E_f[h(X)]$ still works, and

$$\text{Var}\,(\hat{E}_n + \hat{Z}_n) = \text{Var}\,(\hat{E}_n) + 2\text{Cov}\,(\hat{E}_n, \hat{Z}_n) + \text{Var}\,(\hat{Z}_n) < \text{Var}\,(\hat{E}_n)$$

as long as $2|\text{Cov}\,(\hat{E}_n, \hat{Z}_n)| > \text{Var}\,(\hat{Z}_n)$.

A similar trick works with ratios $\hat{E}_n/\hat{Z}_n$ where now $\hat{Z}_n \approx 1$ and Cov $(\hat{E}_n, \hat{Z}_n) > 0$; an example of this "ratio method" will be given in the next subsection.

3.1.4 Importance Sampling and Sampling-Importance-Resampling (SIR)

Importance sampling Importance sampling is natural when one has to calculate $E_f[h(X)]$ where $X \sim f(x)$ but it is easier to sample from $g(x)$. There are two versions of importance sampling.

<u>*Simple importance sampling.*</u> Since

$$E_f[h(X)] = \int h(x)f(x)dx = \int \frac{h(x)f(x)}{g(x)} \cdot g(x)dx = E_g[h(Y)f(Y)/g(Y)]$$

we can estimate $E_f[h(X)]$ as follows:

- Generate $Y_1, \ldots, Y_n$ i.i.d. from $g(y)$;
- Estimate $E_f[h(X)] \approx \frac{1}{n}\sum_1^n h(Y_i)f(Y_i)/g(Y_i)$.

<u>*Fancy importance sampling.*</u> Sometimes, especially in Bayesian problems, it is useful to be able to do importance sampling without computing the normalizing constant for $f(x)$. We noted above that

$$\tilde{I}_n = \frac{1}{n}\sum_1^n h(Y_i)f(Y_i)/g(Y_i) \to E_f[h(X)]$$

It similarly follows that

$$\tilde{O}_n = \frac{1}{n}\sum_1^n 1 \cdot f(Y_i)/g(Y_i) \to 1$$

so a fancier version of importance sampling that is sometimes used is

$$\hat{I}_n \equiv \frac{\tilde{I}_n}{\tilde{O}_n} = \frac{\sum_1^n h(Y_i)f(Y_i)/g(Y_i)}{\sum_1^n f(Y_i)/g(Y_i)} \to E_f[h(X)]$$

This version has two possible advantages.

- Since $f(x)$ appears in both the numerator and denominator, any normalizing constant would cancel, so it is only necessary to know $f(x)$ up to a(ny) multiplicative constant. Similarly, $g(x)$ need only be known up to a multiplicative constant.
- We can see that the variance of the simpler version is

$$\text{Var}\,(\tilde{I}_n) = \frac{1}{n}\text{Var}_g[h(Y)f(Y)/g(Y)]$$

and, applying the multivariate delta method to the function $g(x,y) = x/y$, where $x = \frac{1}{n}\tilde{I}_n$ and $y = \frac{1}{n}\tilde{O}_n$, the variance of the fancier version is approximately

$$\text{Var}\,(\hat{I}_n) = \text{Var}\,(\tilde{I}_n) - 2E_f[h(X)]\text{Cov}\,(\tilde{I}_n, \tilde{O}_n) + (E_f[h(X)])^2\text{Var}\,(\tilde{O}_n)$$

By making Cov $(\tilde{I}_n, \tilde{O}_n)$ large and Var $(\tilde{O}_n)$ small, we can possibly make the MC SE of the fancier version of importance sampling less than the MC SE of the simpler version.

Sampling Importance Resampling (SIR) This is a method of using importance sampling to generate a *sample* from a new distribution, not just estimate an integral. It also goes by the name "weighted bootstrap".

Note that, in the fancier version of importance sampling, if we take $h(x)$ to be

$$h(x) = 1_{\{x \le x_0\}}$$

then, since $h(Y_i) = 0$ if $Y_i > x_0$ and $= 1$ if $Y_i \le x_0$,

$$\hat{I}_n = \frac{\sum_1^n h(Y_i)f(Y_i)/g(Y_i)}{\sum_1^n f(Y_i)/g(Y_i)} = \sum_{i:\, Y_i \le x_0} q_i$$

where $q_i = [f(Y_i)/g(Y_i)]/\sum_1^n f(Y_j)/g(Y_j)$. But also,

$$\hat{I}_n = \sum_{i:\, Y_i \le x_0} q_i \approx E_f[1_{\{X \le x_0\}}] = P[X \le x_0] = F(x_0).$$

Now consider the *discrete* random variable T that takes on the value Y_i with probability q_i; the cdf of T is

$$P[T \le x_0] = \sum_{i:\, Y_i \le x_0} q_i = \hat{I}_n \approx P[X \le x_0]$$

This leads to the following method of modifying a random sample $Y_1, \ldots, Y_n$ from $g(y)$ so that it looks like a random sample $X_1, \ldots, X_n$ from $f(x)$.

> **SIR.** *Suppose $f(x)$ is* proportional to *the density we* want *to sample from; and $g(x)$ is a density we* can *sample from.*
>
> *Then the following procedure produces observations that are* approximately *distributed according to $f(x)$:*
>
> - *Generate $Y_1, \ldots, Y_M$ i.i.d. from $g(x)$;*
> - *Compute $q_1, \ldots, q_M$ as $q_i = [f(Y_i)/g(Y_i)]/\sum_1^n f(Y_j)/g(Y_j)$;*
> - *Sample $X_1 \ldots, X_n$ i.i.d. from the Y's with probabilities the q's.*

Once again, we really only need to know $f(x)$ up to a multiplicative constant.

A cautionary note: Usually you need $M \gg n$ (like, at least $M > 10n$) to make the distribution of the X's reasonably close to $f(x)$.

3.2 Exploring posterior distributions in Bayesian inference, using Monte Carlo

Given a (possibly multidimensional) parameter θ with prior distribution $\pi(\theta)$ and likelihood $L(y|\theta) = \prod_i L_i(y_i|\theta)$, Bayesian inference is based on functionals (i.e. numerical functions) of the posterior distribution

$$p(\theta|y) = \frac{L(y|\theta)\pi(\theta)}{\int L(y|\theta)\pi(\theta)d\theta}$$

Common functionals are

- The posterior probability of an interval $[a, b]$, $\int 1_{\{\theta\in[a,b]\}} p(\theta|y)d\theta$;
- The posterior mean $E[\Theta|y] = \int \theta p(\theta|y)d\theta$;
- The posterior variance $\mathrm{Var}\,(\Theta|y) = \int (\theta - E[\Theta|y])^2 p(\theta|y)d\theta$;
- The "entire" posterior density $p(\theta|y)$.

A direct attack on the first three quantities requires at the very least two integrals:

$$E[h(\theta)|y] = \int h(\theta)p(\theta|y)d\theta = \frac{\int h(\theta)L(y|\theta)\pi(\theta)d\theta}{\int L(y|\theta)\pi(\theta)d\theta} \tag{2}$$

In this section we show how rejection and importance sampling provide basic tools for estimtating the integral (1) and/or obtaining a sample from the full posterior distribution $p(\theta|y)$.

3.2.1 Rejection Method

To link up with the general rejection method, let

$$\begin{aligned} f(\theta) &= L(y|\theta)\pi(\theta), \text{ and} \\ g(\theta) &= \pi(\theta). \end{aligned}$$

In many Bayesian problem s $g(\theta)$ will be relatively simple to sample from, e.g. it may be a product of univariate densities like normals or gammas or uniforms, that SPLUS has built-in functions to generate samples from, so that each coordinate of θ may be sampled independently from its own SPLUS generator. $p(\theta|y)$ is usually hard to sample from, and we typically only know $f(\theta) \propto p(\theta|y)$.

For the rejection method, we need

$$\begin{aligned} f(\theta) &\leq a \cdot g(\theta), \text{ or} \\ L(y|\theta)\pi(\theta) &\leq a \cdot \pi(\theta) \end{aligned}$$

This is easily accomplished by taking $a = L(y|\hat{\theta})$ where $\hat{\theta}$ is the MLE for the problem. Thus the rejection method becomes:

Rejection for Bayes.

- *Generate θ from $\pi(\theta)$;*

- *Generate U from $Unif[0, L(y|\hat{\theta})\pi(\theta)]$;*
- *If $U < L(y|\theta)\pi(\theta)$, accept θ as an observation from $p(\theta|y)$;*
  otherwise reject θ and start over.

Note that $\pi(\theta)$ can be dropped from steps 2 and 3, if we generate $U \sim Unif[0,1]$ and check $U < L(y|\theta)/L(y|\hat{\theta})$. This is good.

Also note that if the prior $\pi(\theta)$ disagrees radically with the likelihood $L(y|\theta)$ about where θ lives (compare the modes, for example), then this is going to be a very slow way to generate posterior observations. This is bad.

3.2.2 Importance Sampling and SIR

A useful question to apply Importance Sampling and SIR to in Bayesian inference is, *"What if I had chosen a different prior distribution?"*

Assume that $\theta_1, \ldots, \theta_M$ are a random sample from $p(\theta|y) \propto L(y|\theta)\pi(\theta)$, and we want switch the prior from $\pi(\theta)$ to $\pi^*(\theta)$.

Now if we let

$$\begin{aligned} f(\theta) &= L(y|\theta)\pi^*(\theta), \text{ and} \\ g(\theta) &= L(y|\theta)\pi(\theta) \end{aligned}$$

we see that the "importance weights" q_i become

$$q_i = [f(\theta_i)/g(\theta_i)]/\sum_1^n f(\theta_j)/g(\theta_j) = [\pi^*(\theta_i)/\pi(\theta_i)]/\sum_1^n \pi^*(\theta_j)/\pi(\theta_j)$$

Thus, posterior expected values for the new prior π^* can be approximated using importance sampling as

$$E_{\pi^*}[h(\theta)|y] \approx \sum_1^M h(\theta_i)\, q_i, \text{ for i.i.d. } \theta_i \sim p(\theta|y) \propto L(y|\theta)\pi(\theta)$$

and an SIR sample (approximately) from the new posterior can be obtained by

Sample $\theta_1^*, \ldots, \theta_n^*$ as i.i.d. from $\theta_1, \ldots, \theta_M$, with probabilities $q_1, \ldots q_M$.