

Intoroducing Monte Carlo Methods with R (Part2 Monte Carlo integration)

Naoya Hieda

2017年05月26日

ParticleFilter に到達する前の重点サンプリングや PMCMC(Particle Marcoh chain monte carlo) のもとになっている MCMC の理論について

いろいろと怪しい部分があったのでまとめる。

この Markdown は主に三章のモンテカルロ積分関係について

参考文献

R によるモンテカルロ法入門

人工知能に関する断創録

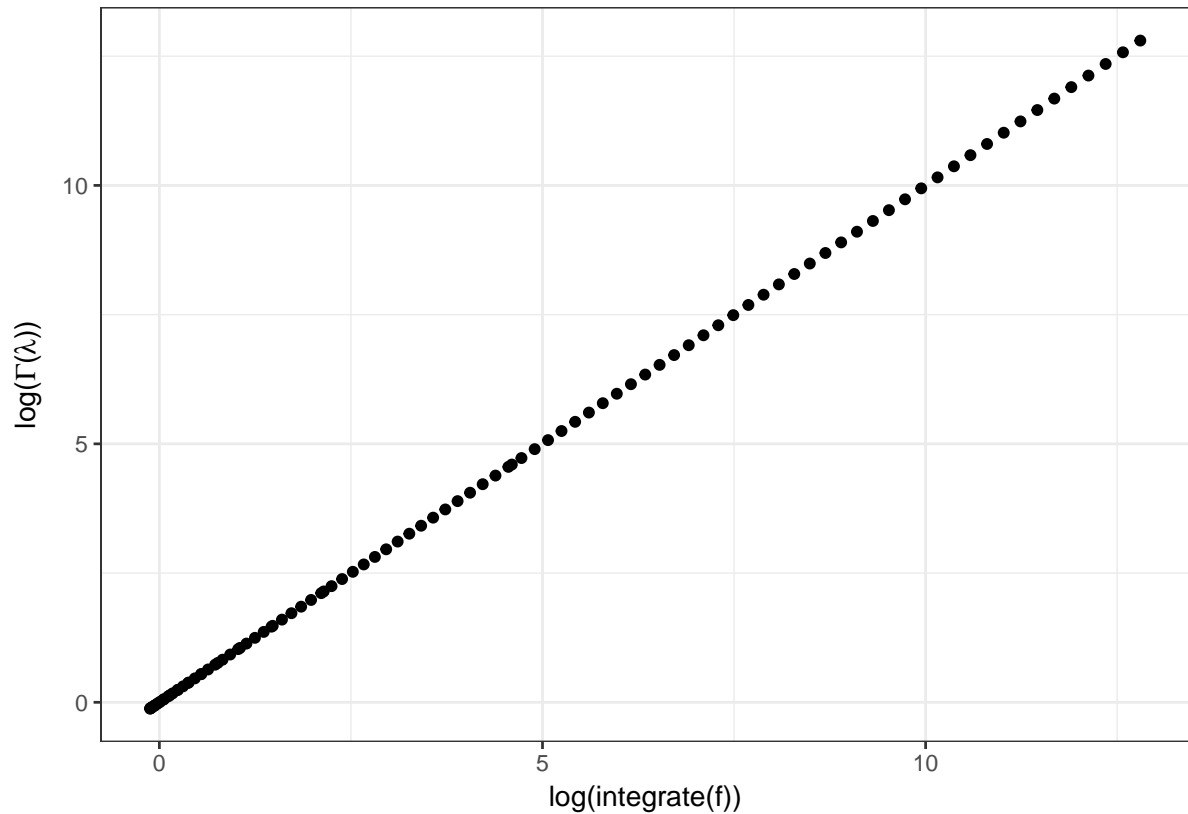
Wolfeyes Bioinformatics beta

R にもともと area 関数と integrate 関数が存在する。ただし、area() は積分で無限の範囲を扱えないし、integrate も安定性に乏しい

実験として以下の積分について、integrate 関数と実際の値を比較してみる (下記を積分すると Γ)

$$\int_0^{\infty} x^{\lambda-1} \exp(-x) dx$$

```
ch <- function(la){
  integrate(function(x) {x^{la-1}*exp(-x)}, 0, Inf)$val
}
plot_d <- data.frame(x = lgamma(seq(0.01,10,le=100)),y = log(apply(as.matrix(seq(0.01,10,le=100)), 1, ch)))
ggplot(plot_d,aes(x=x,y=y))+geom_point()+xlab("log(integrate(f))")+
  ylab(expression(paste("log(",Gamma,"(",lambda,")"))))
```



この場合は、結構きれい

integrate 関数のような数値積分方法で困難なのは、被積分関数において重要な範囲を見逃しやすいこと。これに対してシミュレーションでは、積分にかかわる確率密度の情報を活用することで、この範囲に絞った適用が可能。

位置パラメータ $\theta = 350$ とするコーシー分布の乱数 10 個をサンプルとして検討する。一様分布を事前分布に仮定するとサンプルの (疑似) 周辺分布が以下ようになる

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

しかし、integrate は誤った数値を返す

下記は誤差評価を確認した後、area 関数との積分結果の対数尤度比較

```
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)
}

print(integrate(lik,-Inf,Inf))
```

```
## 5.462081e-44 with absolute error < 1.1e-43
```

```
print(integrate(lik,200,400))
```

```
## 4.391768e-09 with absolute error < 8.1e-09
```

```
cac=rcauchy(10)
```

```
nin=function(a){integrate(lik,-a,a)$val}
```

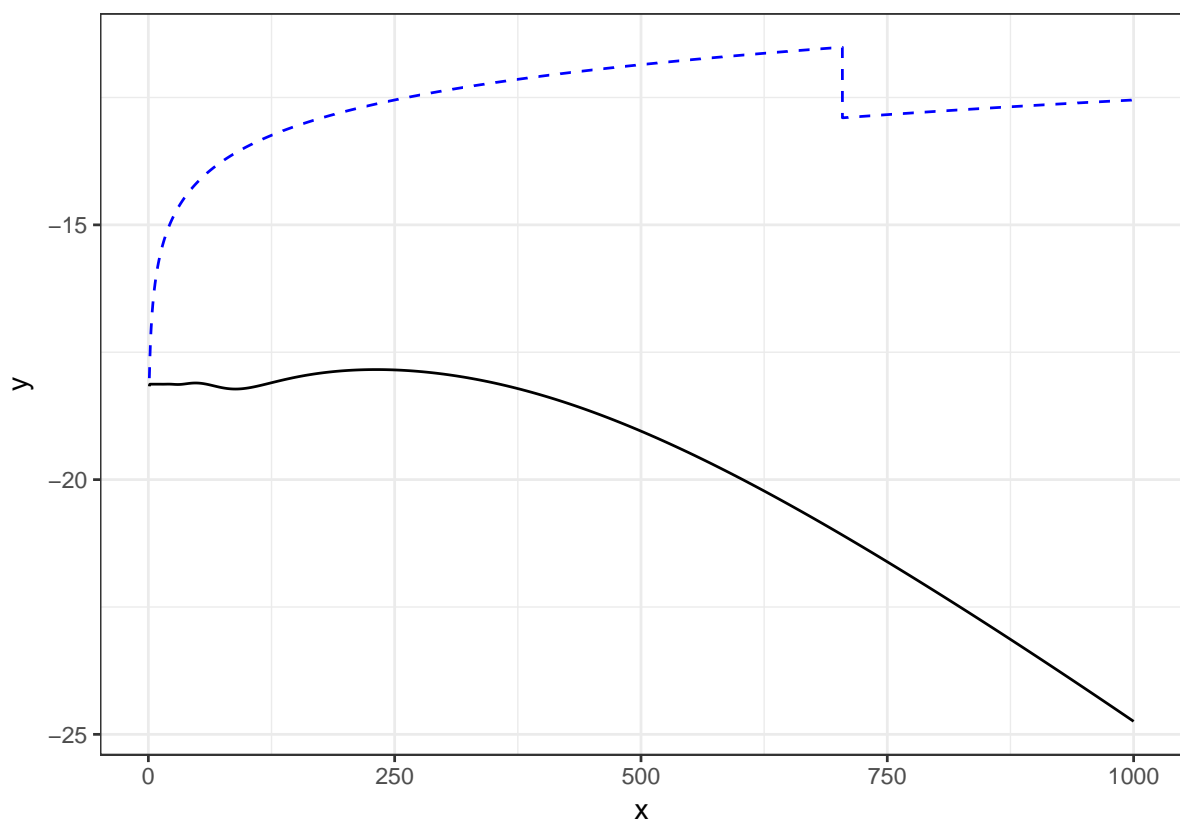
```
nan=function(a){area(lik,-a,a)}
```

```
x=seq(1,10^3,le=10^4)
```

```
y=log(apply(as.matrix(x),1,nin))
```

```
z=log(apply(as.matrix(x),1,nan))
```

```
ggplot()+geom_line(data = data.frame(x,y),mapping = aes(x=x,y=y))+geom_line(data = data.frame(x,z),mapping = aes(x=x,y=z),
```



古典的なモンテカルロ計算

シミュレーションを実際の問題に適用する前に、こうした応用が適正であることを再確認しておく。一般に以下の積分をどのように評価するのかということが問題になる。

$$E_f[h(X)] = \int_{\chi} h(x)f(x)dx \quad (3,1)$$

ここで χ は乱数 x の値の集合で、これは通常密度 f の台に等しくなる。

これを近似するモンテカルロ法の原理は、密度 f からサンプル (X_1, \dots, X_m) を生成して、近似として以下の

経験平均を揭示すること

$$\bar{h}_m = \frac{1}{m} \sum_{j=1}^m h(x_j)$$

\bar{h}_m は大数の強法則により、ほぼ間違いなく $E_f(h(X))$ に収束する。よって、R なら `mean` などを計算することで求まる。さらに、 $h^2(X)$ が f のもとで有限の期待値をもつ場合、 \bar{h}_m の収束時間を評価することが可能。収束時間は $O(\sqrt{m})$ となり、近似の漸近的な分散は以下となるため。

$$\text{var}(\bar{h}_m) = \frac{1}{m} \int_{\mathcal{X}} (h(x) - E_f[h(X)])^2 f(x) dx$$

これは、サンプル (X_1, \dots, X_m) から以下のように推定できる。

$$v_m = \frac{1}{m^2} \sum_{j=1}^m [h(x_j) - \bar{h}_m]^2$$

さらには中心極限定理によって m が十分大きいのであれば、以下が近似的に $N(0, 1)$ の正規分布をする変数となる。

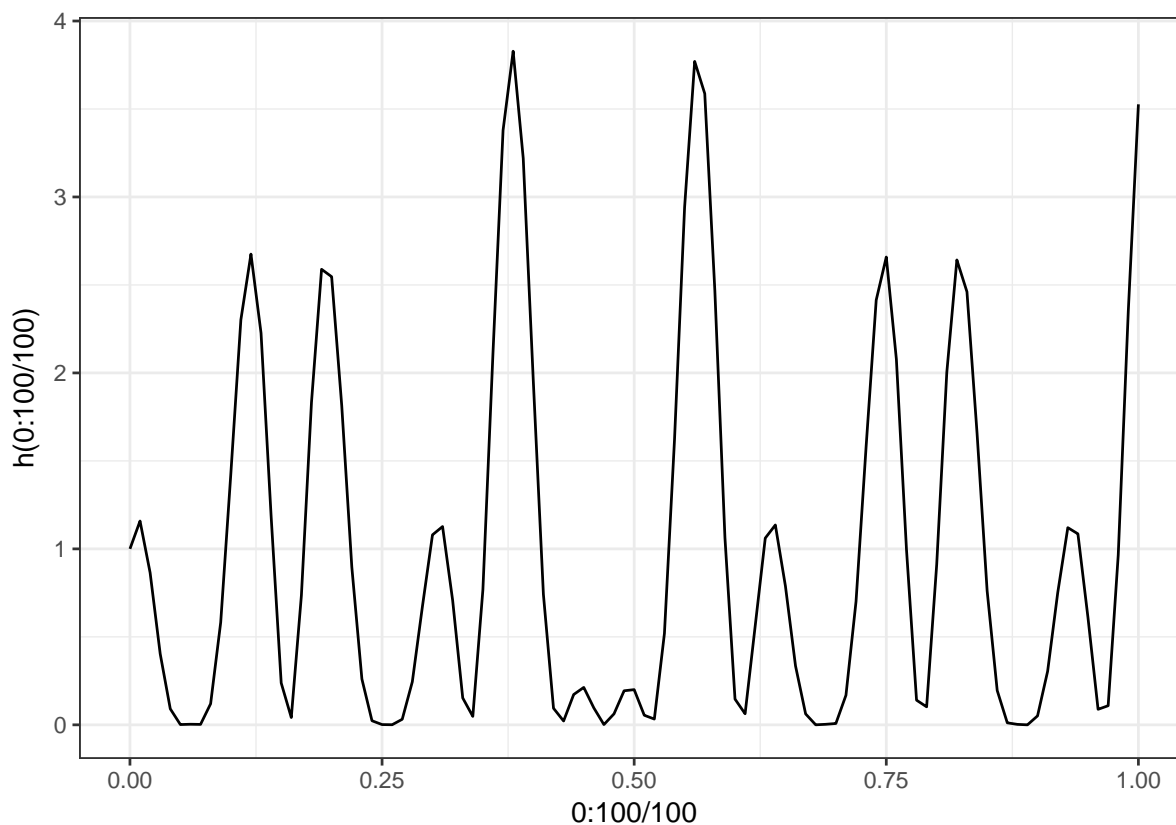
$$\frac{\bar{h}_m - E_f[h(X)]}{\sqrt{v_m}}$$

次の仮の関数の $[0, 1]$ でモンテカルロ積分を試してみる

$$h(x) = [\cos(50x) + \sin(20x)]^2$$

上が実際にこの曲線が描く概形で、下がモンテカルロ積分の収束をしめしている

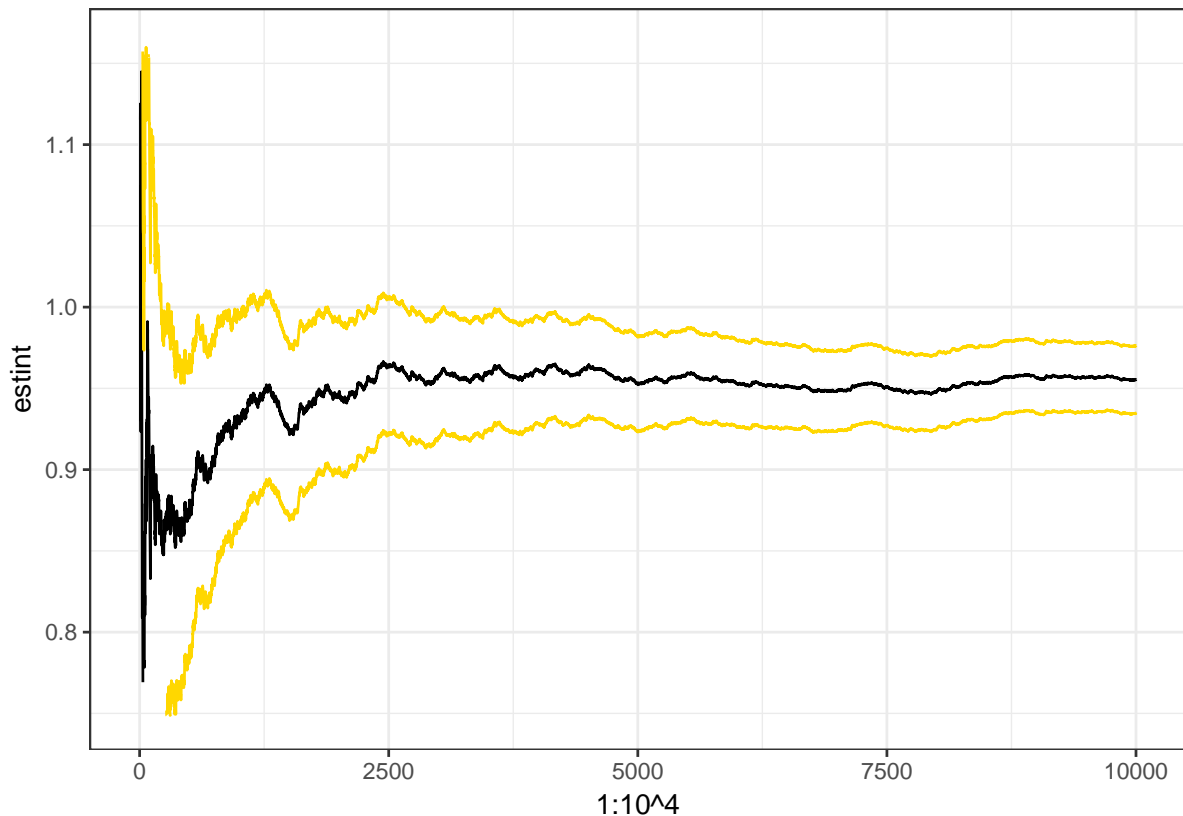
```
h <- function(x){(cos(50 * x) + sin(20 * x))^2}
ggplot()+geom_line(mapping=aes(x= 0:100/100,y =h(0:100/100)))
```



```
integrate(h,0,1)
```

```
## 0.9652009 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)
ggplot(data.frame(estint), aes(x=1:10^4,y=estint))+geom_line(colour=1)+ylim(c(mean(x)+20*c(-esterr[10^4],esterr[10^4]))) +
  geom_line(mapping = aes(x=1:10^4,y=estint+2*esterr),colour="gold")+geom_line(mapping=aes(x=1:10^4,y=estint-2*esterr),colour="gold")
```



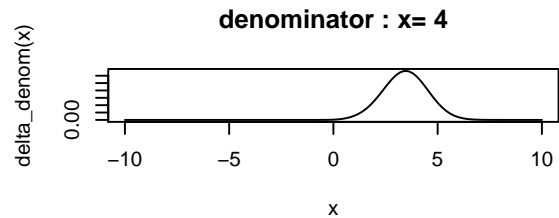
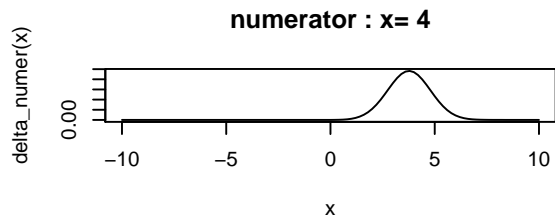
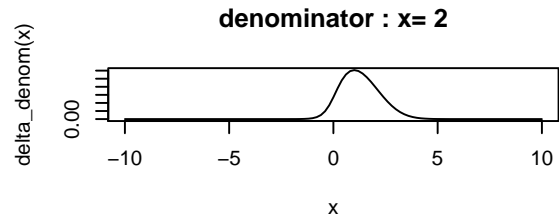
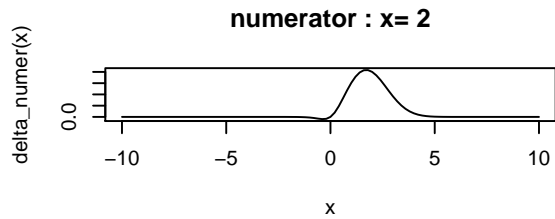
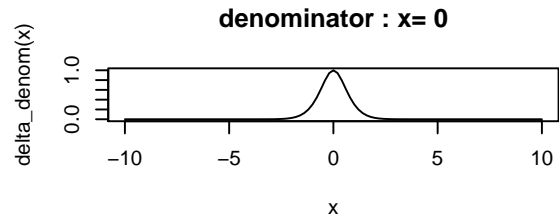
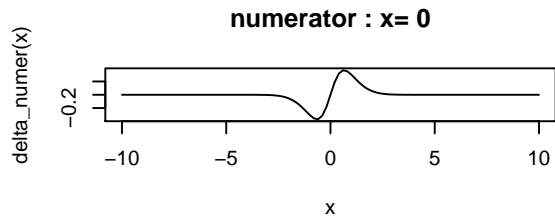
練習問題 3.1

正規・コーシーベイズ推定量

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

- 被積分関数をプロットし、コーシー・シミュレーションにもとづくモンテカルロ積分を計算する。

```
delta_numer <- function(theta){theta/(1+theta^2)*exp(-(x-theta)^2/2)}
delta_denom <- function(theta){1/(1+theta^2)*exp(-(x-theta)^2/2)}
par(mfrow=c(3,2))
for(x in c(0,2,4)){
  curve(delta_numer,from=-10,to=10,main=paste("numerator : x=",x))
  curve(delta_denom,from=-10,to=10,main=paste("denominator : x=",x))
}
```

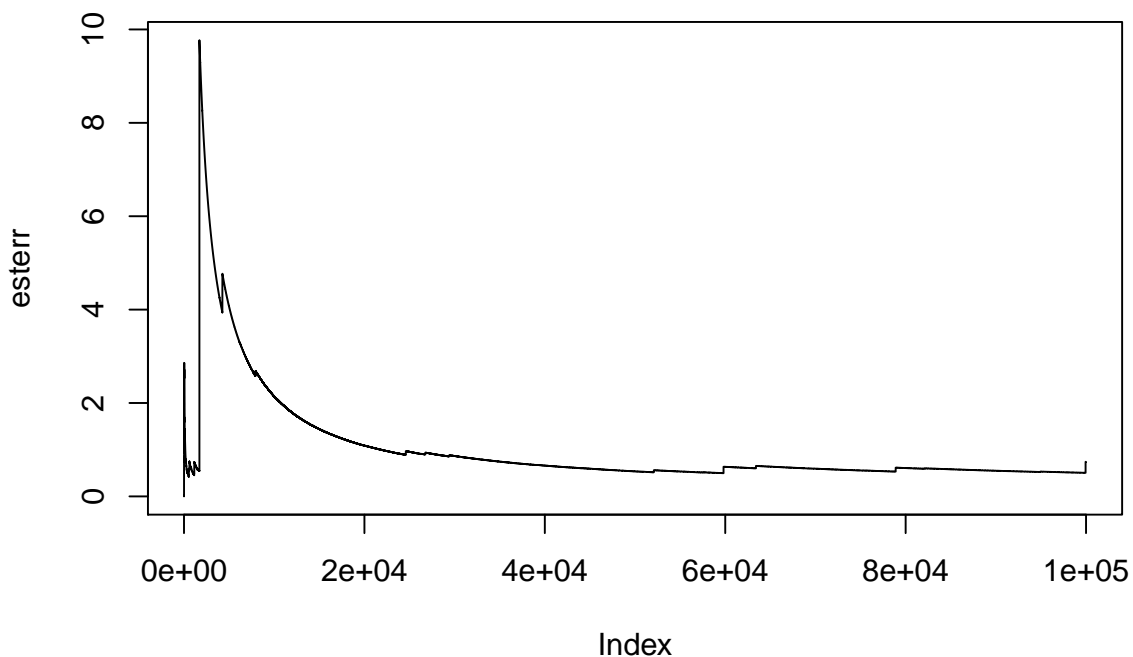


```
for(x in c(0,2,4)){
  #正規分布から乱数を発生させて、コーシーで評価
  N <- 10^5
  norm <- rnorm(N,x)
  cauchy <- dcauchy(norm)
  print(mean(norm*cauchy)/mean(cauchy))
  #その逆 結果はほぼ同じ
  cauchy <- rcauchy(N)
  norm <- dnorm(cauchy,x)
  print(mean(cauchy*norm)/mean(norm))
}
```

```
## [1] -0.001062424
## [1] -0.000571549
## [1] 1.277758
## [1] 1.28406
## [1] 3.429738
## [1] 3.425443
```

- 収束を推定値の標準誤差でモニタリングする。95%の信頼幅を小数点3位の精度で求める

```
estint <- cumsum(cauchy*norm)/cumsum(norm)
esterr <- sqrt(cumsum((cauchy-estint)^2)/c(1:length(cauchy))^2)
plot(esterr,type='l')
```



本題その1

重点サンプリング

上の近似の評価は、たいていの場合には最適ではない

3.3.1 参照量を任意に変更

重点サンプリング法は、期待値 $E_f[h(x)] = \int_{\mathcal{X}} h(x)f(x)dx$ の代替式に基づく。ある任意の密度 g で $h \times f$ がゼロとは異なり、真に正ならば、次のように書き換えることができる

$$E_f[h(X)] = \int_{\mathcal{X}} h(x) \frac{f(x)}{g(x)} g(x) dx = E_g\left[\frac{h(X)f(X)}{g(X)}\right]$$

これは密度 g の下での期待値となる。この重点サンプリング基本恒等式は、次の推定量の利用を保証する。

$$\frac{1}{n} \sum_{j=1}^m \frac{f(X_j)}{g(X_j)} h(X_j) \rightarrow E_f[h(X)] \quad (3, 4)$$

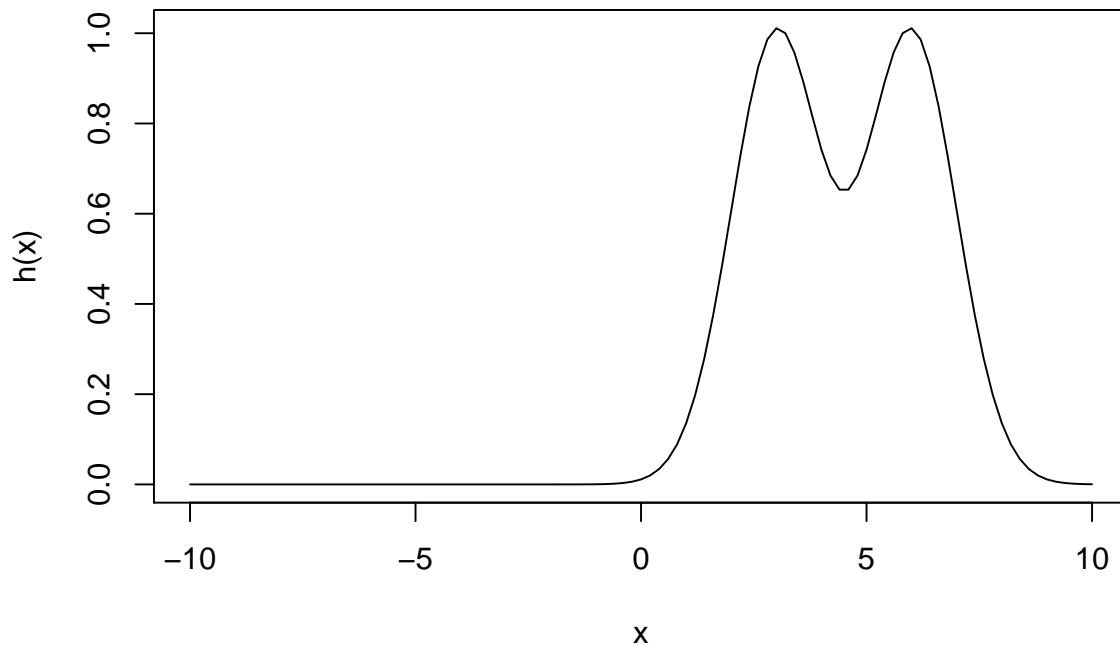
これは g から生成したサンプルに基づく。このように期待値を g の下での期待値として書くことができるので、分布 g の選択がなんであれ、通常のモンテカルロ推定量 \bar{h} が収束するのと同じように収束する。

練習問題 3.4

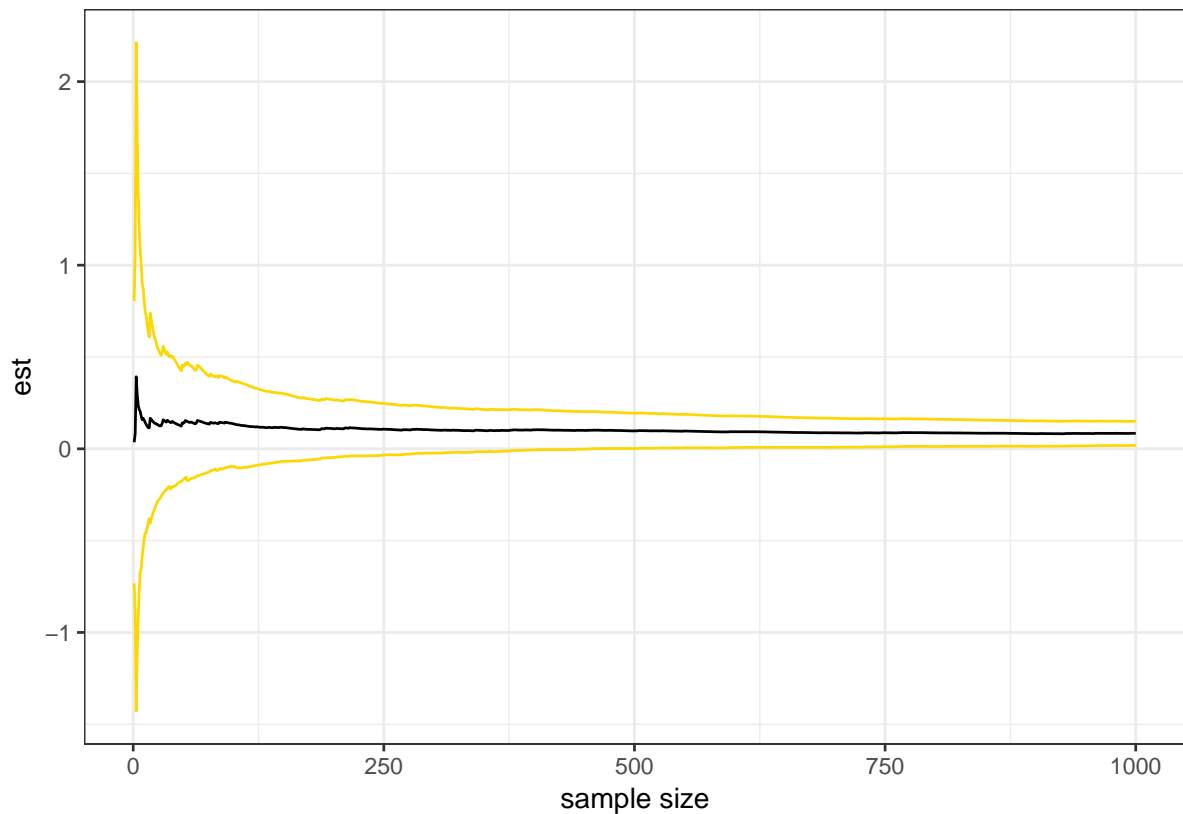
f を正規 pdf とし、 $h(x)$ を $\exp(-(x-3)^2) + \exp(-(x-6)^2/2)$ として期待値 $E[h(X)]$ の計算を検討する

- $E_f[h(X)]$ が閉じた式で計算できることを示し、その値を導く。
- 正規分布 $N(0,1)$ に基づくサンプルサイズ 10^3 の通常のモンテカルロ近似を構成し、誤差を評価する。

```
N <- 10^3
h <- function(x){exp(-(x-3)^2/2)+exp(-(x-6)^2/2)}
curve(h,-10,10)
```



```
sample <- rnorm(N)
n_mc_est <- cumsum(h(sample))/c(1:length(sample))
n_mc_err <- sqrt(cumsum((sample - n_mc_est)^2))/(1:length(sample))
n_mc <- data.frame(est = n_mc_est,error_max = n_mc_est + 2 * n_mc_err,error_min = n_mc_est - 2 * n_mc_err)
ggplot(n_mc,aes(x=1:length(sample),y=est))+geom_line()+
  geom_line(mapping = aes(x=1:length(sample),y=error_max),colour='gold')+
  geom_line(mapping = aes(x=1:length(sample),y=error_min),colour='gold')+
  xlab('sample size')
```

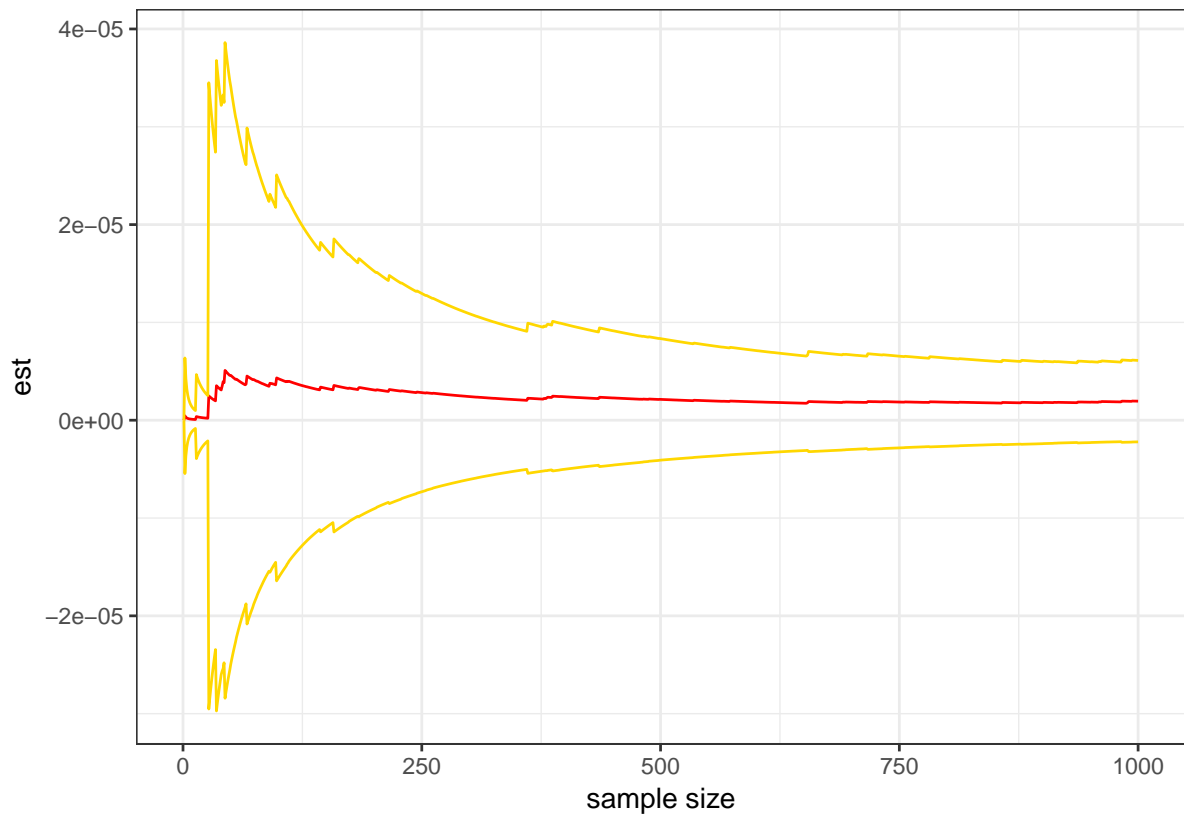



```
n_mc_est[N]
```

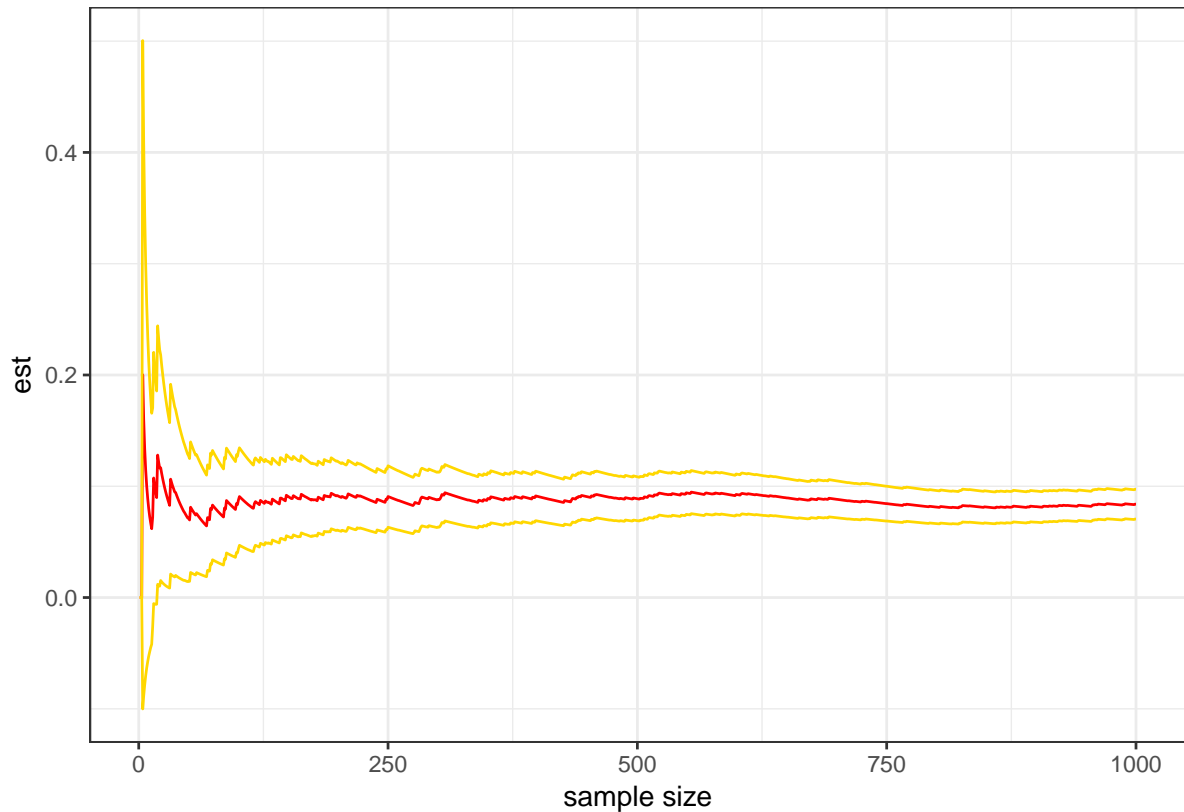
```
## [1] 0.08424968
```

- 一様分布 $u(-8,-1)$ に対応する重点関数 g にもとづくサンプルサイズ 10^3 の重点サンプリング近似と比較する。(これは収束しない 範囲を網羅してない)

```
sample_g <- runif(N,-8,-1)
g_mc_est <- cumsum(h(sample_g) * dnorm(sample_g)) / c(1:length(sample_g))
g_mc_err <- sqrt(cumsum(((h(sample_g) * dnorm(sample_g) / dunif(sample_g,-8,-1) - g_mc_est)^2)) / c(1:length(sample_g))
g_mc <- data.frame(est = g_mc_est,error_max = g_mc_est + 2 * g_mc_err,error_min = g_mc_est - 2 * g_mc_err)
ggplot(g_mc,aes(x=1:length(sample),y=est))+geom_line(colour='red')+
  geom_line(mapping = aes(x=1:length(sample),y=error_max),colour='gold')+
  geom_line(mapping = aes(x=1:length(sample),y=error_min),colour='gold')+
  xlab('sample size')
```



```
sample_g <- runif(N,-10,10)
g_mc_est <- cumsum(h(sample_g) * dnorm(sample_g) / dunif(sample_g,-10,10)) / c(1:length(sample_g))
g_mc_err <- sqrt(cumsum((h(sample_g) * dnorm(sample_g) / dunif(sample_g,-10,10) - g_mc_est)^2)) / c(1:length(sample_g))
g_mc <- data.frame(est = g_mc_est,error_max = g_mc_est + 2 * g_mc_err,error_min = g_mc_est - 2 * g_mc_err)
ggplot(g_mc,aes(x=1:length(sample),y=est))+geom_line(colour='red')+
  geom_line(mapping = aes(x=1:length(sample),y=error_max),colour='gold')+
  geom_line(mapping = aes(x=1:length(sample),y=error_min),colour='gold')+
  xlab('sample size')
```



```
g_mc_est[N]
```

```
## [1] 0.08388949
```

通常のモンテカルロ合計によって裾の確率を近似する方法は、裾の外れに近づくにつれ破綻する。たとえば、 $Z \sim N(0, 1)$ で $P(Z > 4.5)$ の確率を知りたいとする。

```
pnorm(-4.5, log=T)
```

```
## [1] -12.59242
```

```
pnorm(-4.5)
```

```
## [1] 3.397673e-06
```

約 300 万回の反復で 1 回しか起きない。

きわめて稀な事象の確率を知ろうとしたとき、 f からの単純なシミュレーションでは、安定した解をえるのに膨大なシミュレーションが必要になる。しかし、重点サンプリングのおかげで、精度を大幅に改善することができ、したがってシミュレーション回数を数桁分減らすことができる。

例えば台が $(4.5, \infty)$ に制約された分布を例にする。ここではモンテカルロ推定量の余分で不要な分散がゼロのシミュレーション (つまり $x \leq 4.5$ の範囲) により消えている。g を 4.5 で切り詰めた指数分布の密度とするのが自然

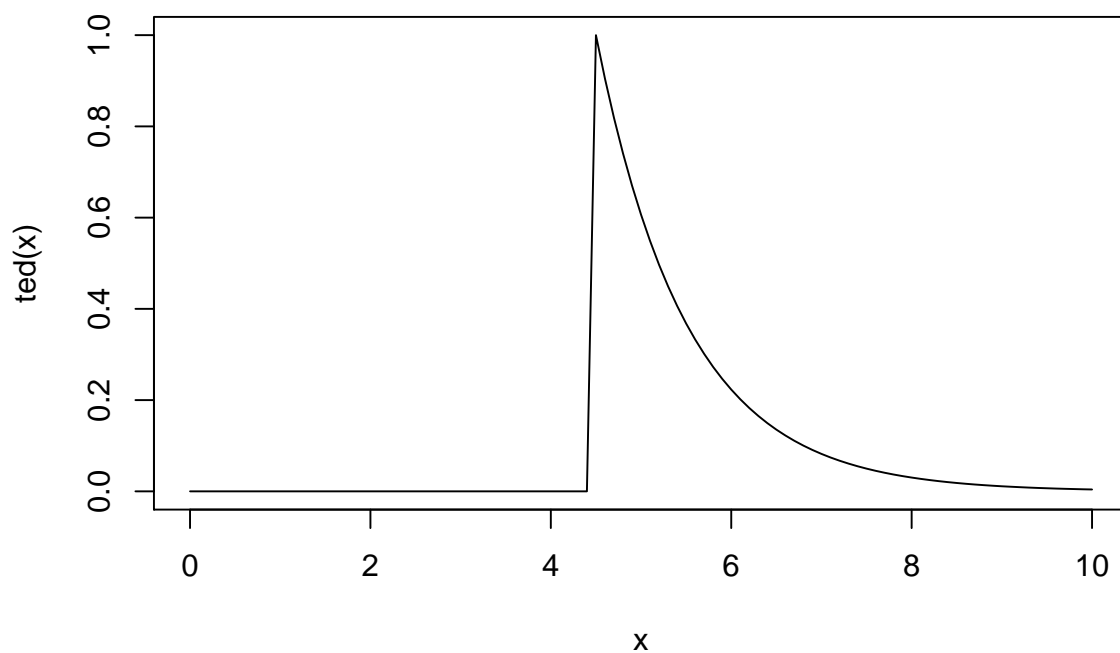
切断指数分布 (truncated exponential distribution) という分布、今は指数分布から 4.5 以上の区間を取り出してきて、その区間の面積が 1 になるように再調整した分布

$$g(y) = e^{-y} / \int_{4.5}^{\infty} e^{-x} dx = e^{-(y-4.5)}$$

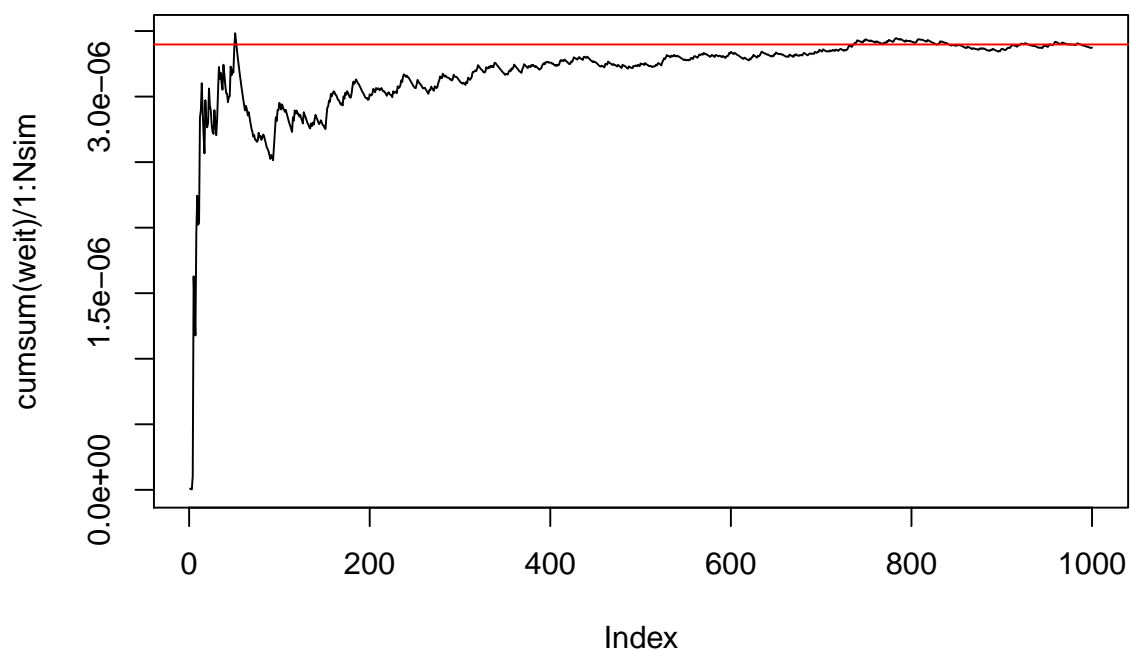
すると対応する重点サンプリングによる裾の確率の推定量が以下のように与えられる

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

```
ted <- function(y){dexp(y-4.5)}  
curve(ted,0,10)
```



```
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")
```

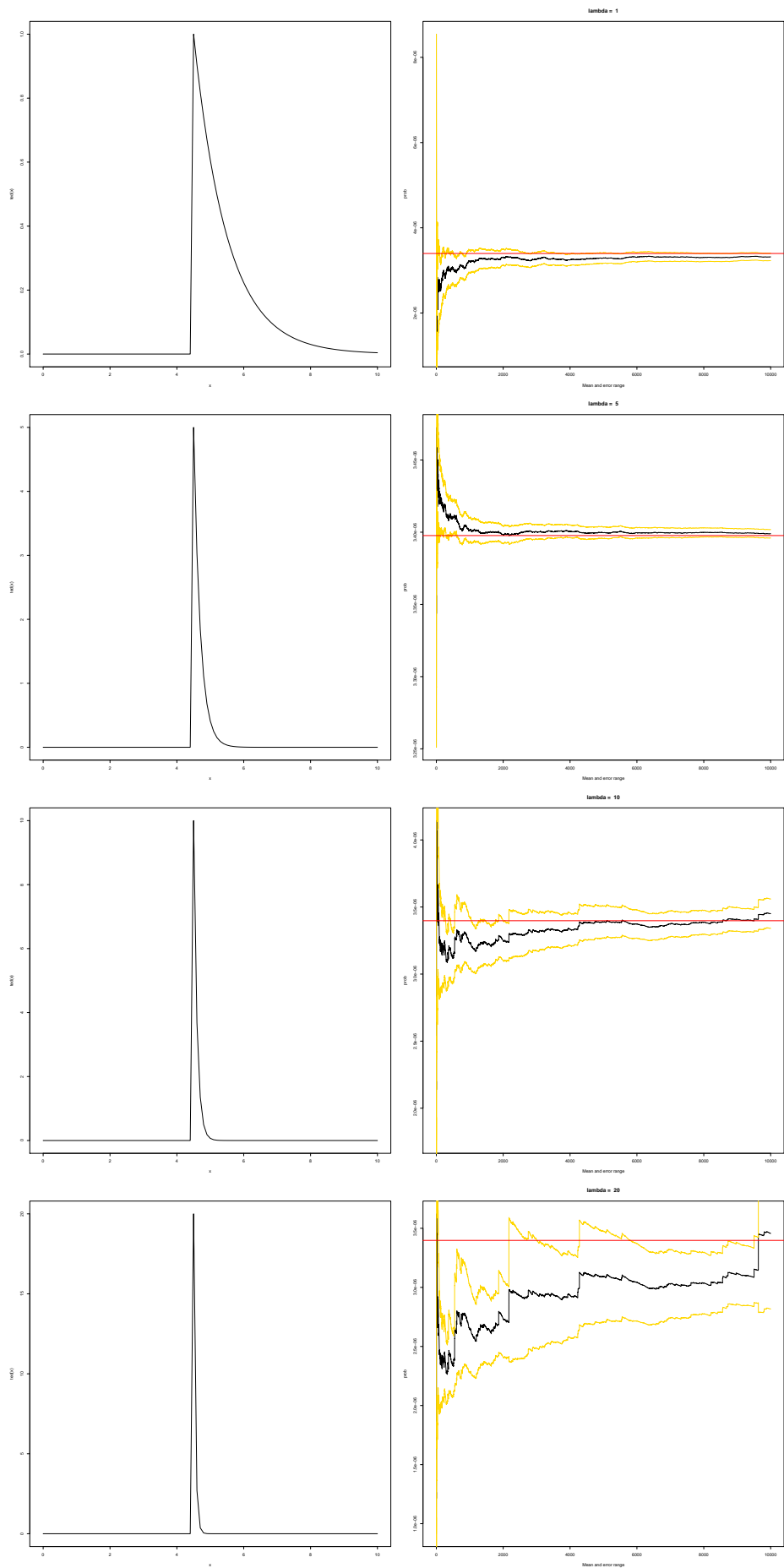


かなり早い段階で収束している。

練習問題 3.5

上の例で指数分布が切り詰められると、裾の確率近似の分散に影響することを調べる

```
par(mfrow=c(4,2))
Nsim <- 10^4
x <- rexp(Nsim)
for(lambda in c(1, 5, 10, 20)){
  ted <- function(y){dexp(y-4.5,lambda)}
  curve(ted,0,10)
  y <- x/lambda+4.5
  weit <- dnorm(y)/dexp(y-4.5, lambda)
  estint <- cumsum(weit)/1:Nsim
  esterr <- sqrt(cumsum((weit-estint)^2))/(1:Nsim)
  plot(estint, xlab="Mean and error range", ylab="prob", type="l", main=paste("lambda = ",lambda))
  lines(estint+2*esterr, col="gold", lwd=2)
  lines(estint-2*esterr, col="gold", lwd=2)
  abline(h=pnorm(-4.5), col="red")
}
```



たぶん、去年の研究の時も、VaRの計算はgの範囲を狭めてサンプリングによって求めるべきだった。

ベイズの枠組みにもとづいて、ベータ分布 $Be(\alpha, \beta)$ からの観測地 x を考える。

$$x \sim \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} II$$

(α, β) には以下の形式の共役事前分布族がある。

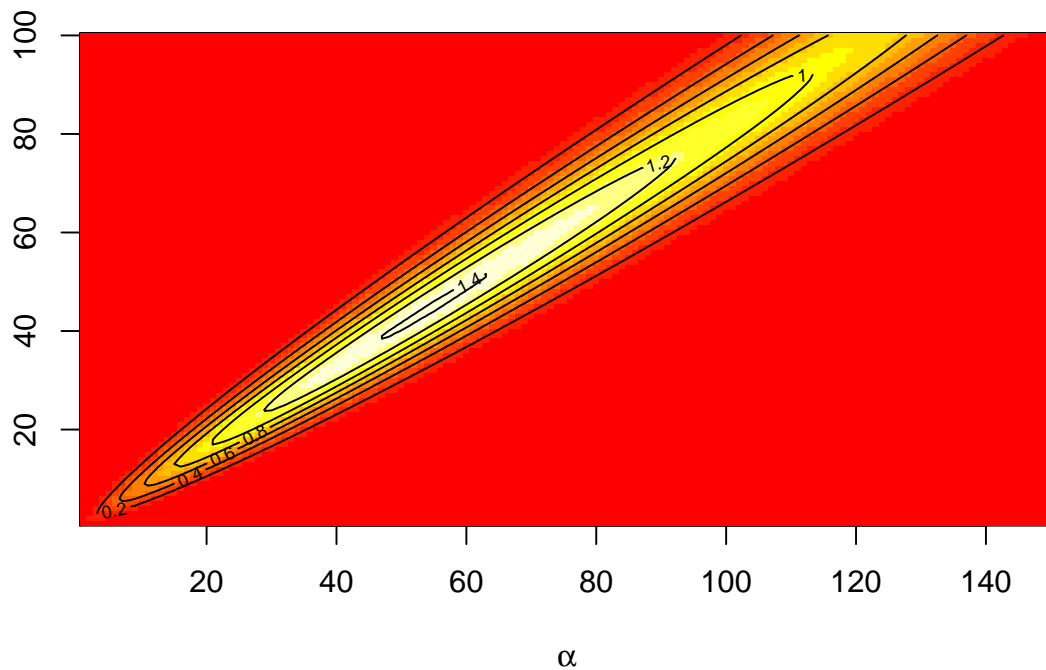
$$\pi(\alpha, \beta) \propto \left\{ \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \right\}^{\lambda} x_0^{\alpha} y_0^{\beta}$$

ここで λ, x_0, y_0 はハイパーパラメータ。この際、事後分布は以下に等しい

$$\pi(\alpha, \beta | x) \propto \left\{ \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \right\}^{\lambda+1} [xx_0]^{\alpha} [(1-x)y_0]^{\beta}$$

この分布は、ガンマが扱いにくいというだけでも制御しにくい。そのため、重点関数が欲しいので、予備調査として、 $\pi(\alpha, \beta | \pi)$ をグラフ表示してみる。

```
f <- function(a,b){  
  exp(2*(lgamma(a+b) - lgamma(a) - lgamma(b)) +  
    a*log(0.3) + b*log(0.2))}  
  
aa = 1:150  
bb = 1:100  
  
post = outer(aa,bb,f)  
image(aa,bb,post,xlab=expression(alpha),ylab="")  
contour(aa,bb,post,add=T)
```



対 (α, β) には正規分布かスチューデントの t 分布が適切そう

スチューデント $T(3, \mu, \Sigma)$ 分布で $\mu = (50, 45)$ とし

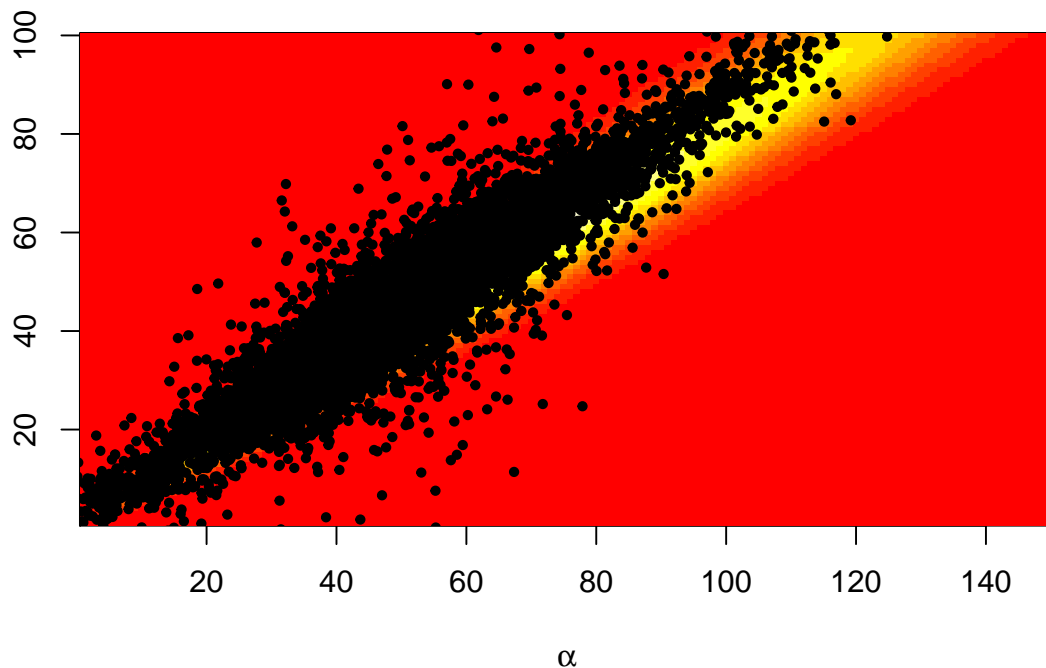
$$\Sigma = \begin{pmatrix} 220 & 190 \\ 190 & 180 \end{pmatrix}$$

の場合でシミュレーションした結果、次のように適切な当てはめが得られる。この共分散行列は著者が試行錯誤して得られたもの

元々独立で、分散 1 の 2 変数の乱数に対してコレスキー分解した分散共分散行列をかけると、

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix} \quad \text{col}(\Sigma) = \begin{pmatrix} \sigma_1 & 0 \\ \frac{\sigma_{12}}{\sigma_1} & \sqrt{\sigma_2^2 - \frac{\sigma_{12}^2}{\sigma_1^2}} \end{pmatrix}$$

```
x <- matrix(rt(2*10^4,3),ncol=2)
E <- matrix(c(220,190,190,180),ncol=2)
image(aa,bb,post,xlab=expression(alpha),ylab="")
y <- t(t(chol(E)) %*% t(x)+c(50,45))
points(y, cex=0.6,pch=19)
```

対象とする量が、ベイズにおけるモデル比較のように周辺尤度であるならば

$$m(x) = \int_{R_+^2} f(x|\alpha, \beta) \pi(\alpha, \beta) d\alpha d\beta = \frac{\int_{R_+^2} \left\{ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \right\}^{\lambda+1} [xx_0]^\alpha [(1-x)y_0]^\beta d\alpha d\beta}{x(1-x) \int_{R_+^2} \left\{ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \right\}^\lambda x_0^\alpha y_0^\beta d\alpha d\beta}$$

二つの積分を近似する必要があるが、どちらにも同じ t サンプルを利用することができる。この近似は以下の様になる。

$$\hat{m}(x) = \frac{\sum_{i=1}^n \left\{ \frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \right\}^{\lambda+1} [xx_0]^{\alpha_i} [(1-x)y_0]^{\beta_i} / g(\alpha_i, \beta_i)}{\sum_{i=1}^n \left\{ \frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \right\}^\lambda x_0^{\alpha_i} y_0^{\beta_i} / g(\alpha_i, \beta_i)}$$

ここで $(\alpha_i, \beta_i)_{i \leq n}$ は g からの n の独立同分布の実現値。

```
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(0.3) + a[,2]*log(0.2))
h <- function(a) exp(1*(lgamma(a[,1] + a[,2]) - lgamma(a[,1]) - lgamma(a[,2])) + a[,1]*log(0.5) + a[,2]*log(0.5))
den <- dt(normx, 3)
sim <- mean(f(y)/den)/mean(h(y)/den)
sim
```

```
## [1] 0.3282088
```

ここでのシミュレーションによる周辺尤度の値は 0.3282088 となった。同様に、パラメータ α, β の事後期待値も以下のように得られる。

```
mean(y[,1]*f(y)/den)/mean(f(y)/den)
```

```
## [1] 88.0124
```

```
mean(y[,2]*f(y)/den)/mean(f(y)/den)
```

```
## [1] 74.84048
```

3.3.2 サンプル重点サンプリング

複雑な分布をシミュレーションする代替手法

この手法では、 g からシミュレーションしたサンプルとその重点重み $f(X_i)/g(X_i)$ を生成する。このサンプルを多項リサンプリングによってリサイクルし、(ほぼ) f からのサンプルを得ることができる。

実際、重み付けされた母集団 $\{X_1, \dots, X_n\}$ から、確率 $f(X_i)/ng(X_i)$ で X_i を復元抽出できるとすれば、以下の分布に従う X^* をえられる。

$$Pr(X^* \in A) = \sum_{i=1}^n Pr(X^* \in A \text{ かつ } X^* = X_i) = \int_A \frac{f(x)}{g(x)} g(x) dx = \int_A f(x) dx$$

すると f からの正確なシミュレーションが生成される。残念ながら確率 $f(X_i)/ng(X_i)$ の和は 1 にならない。そのため以下のように再正規化する必要がある。

$$w_i = \frac{1}{n} \{f(X_i)/g(X_i)\} / \frac{1}{n} \sum_{j=1}^n \{f(X_j)/g(X_j)\}$$

ここで、分母はほぼ確実に 1 に収束するが、リサンプリングされた値の分布にバイアスが生じる。

しかしながら、サンプルサイズが大きい場合、このバイアスは無視できる程度であり、多項リサンプリングを使って f からのサンプルを近似できる。

練習問題 3.6

重点サンプル $(X_i, f(X_i)/g(X_i))$ を所与とすると、 w_i がポアソン分布 $w_i \sim P(f(X_i)/g(X_i))$ に従うのならば、推定量

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

にバイアスがないことを示す。またこのサンプリングの仕組みによるサンプルが、 f の周辺分布であることを導く。

とりあえず保留

重点サンプリング推定量で再正規化重みを使うと、自己正規化重点サンプリング推定量が生成される。

$$\sum_{i=1}^n h(X_i) f(X_i)/g(X_i) / \sum_{j=1}^n f(X_j)/g(X_j)$$

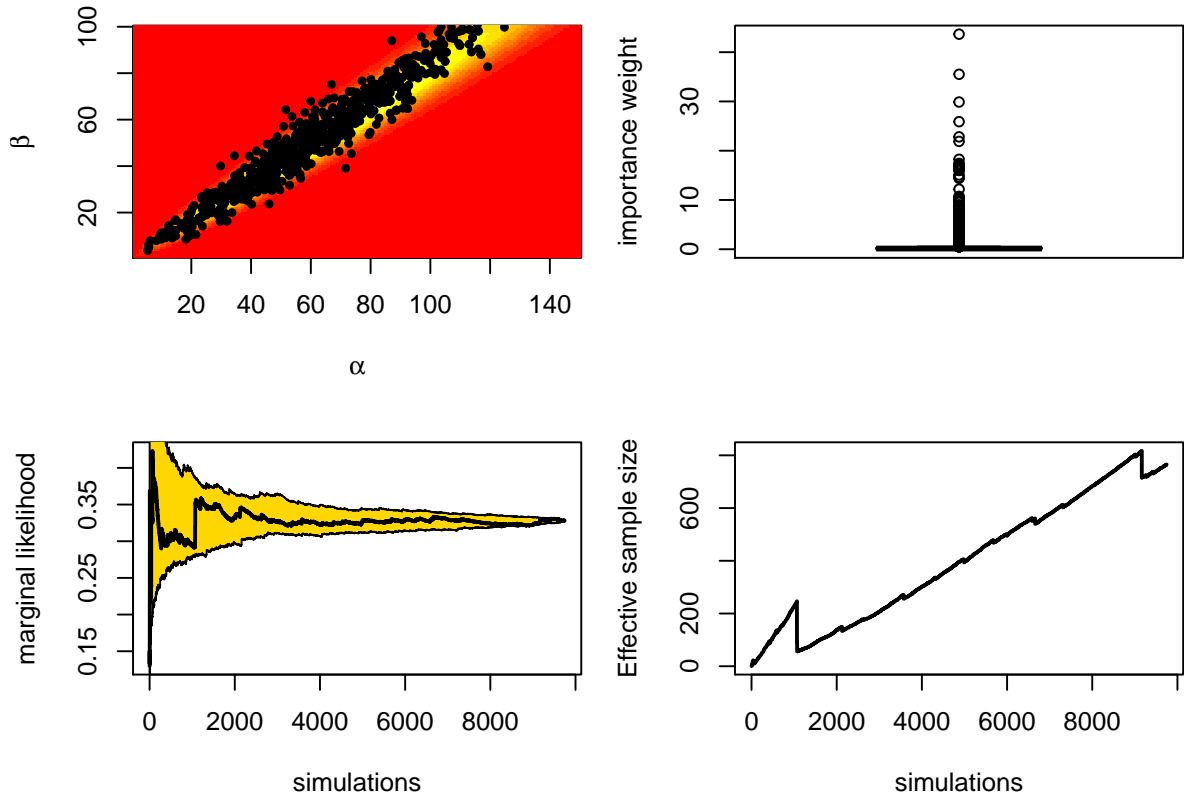
これは f か g に正規化定数がかけている場合にも利用できる。

注意

重点重みは、目標密度に対するシミュレーション・サンプルの妥当性を相対的に評価するにすぎない。つまり、生成した x のなかで、ほかの値よりはそれらしいというだけで、実際の分布からサンプリングできていることを絶対的に評価はしていない。そりゃそうだ

先ほどの、ベータ分布からの観測値についての周辺分布の近似の妥当性(重点サンプリングの回の収束性)については、以下のようにグラフを利用していく

```
par(mfrow=c(2,2),mar=c(4,4,2,1))
weit <- (f(y)/den/mean(h(y)/den))
image(aa, bb, post, xlab=expression(alpha),
      ylab = expression(beta))
points(y[sample(1:length(weit), 10^3, rep=T, pro=weit), ],
       cex = 0.6, 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, 0.95)
lowa <- apply(boot, 2, quantile, 0.05)
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)
```



左上の図は、重点重み $\pi(\alpha_i, \beta_i|x)/g(\alpha_i, \beta_i)$ で重み付けしたサンプルが、 $\pi(\alpha, \beta|x)$ からのサンプルをよく表現していることが分かる。リサンプリングした点では、いくつかの点で劣化が認められず、代わりに目標分布の正確なレンジを高い密度でカバーしている。右上の図は重点重みの広がりを表している。他と比べて重みが大きなシミュレーションもあるが、この方法が劣化を招いていることを示唆するほど極端ではない。左下の図は n の増加に対する推定量 $\hat{m}(x)$ の収束を表現している。ここで系列を囲むグレーの帯域は、推定量の変動をブーストラップによって表現している。右下のパネルの曲線は、重点サンプリングを使った場合の有効サンプルサイズによる有効性の低下を、以下で表現したもの。

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

これは (α_i, β_i) が事後分布から生成されていれば n に等しくなるはず。このプロットでは、生成されたサンプルの有効性は約 6% となる。

3.3.3 重点関数の選択

重点サンプリングは汎用性の高い技法ですが、その反面、重点関数 g の選択を誤ると期待した結果は得られない。最適な g は、理論的には求まるものの、実戦での有用性は低いので、該当の重点関数 g の妥当性を判断する場合は非常に重要となるのが、結果の推定量の分散を検討すること。

実際、(3,4) は (3,1) が存在すれば、ほぼ確実に (3,1) に収束するが、この推定量の分散は、以下の期待値が有限の場合にだけ有限となる。

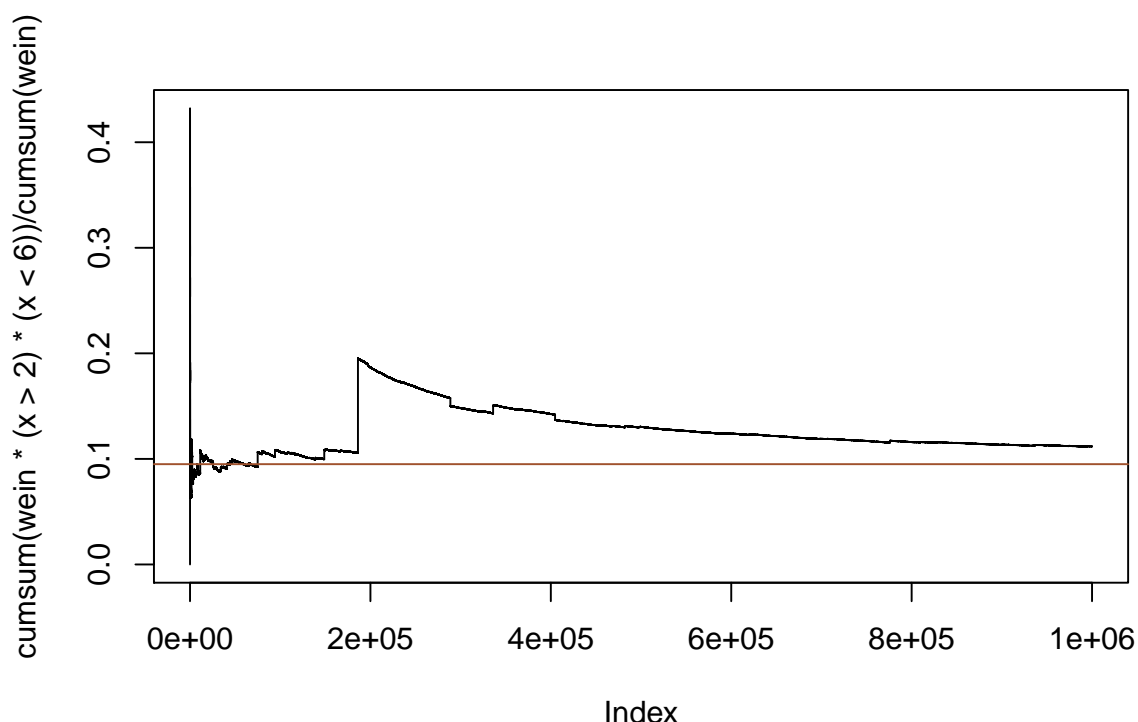
$$E_g \left[h^2(X) \frac{f^2(X)}{g^2(X)} \right] = E_f \left[h^2(X) \frac{f(X)}{g(X)} \right] = \int_{\mathcal{X}} h^2(x) \frac{f^2(x)}{g(x)} dx < \infty$$

比 f/g が有界でなくなる f よりも裾のカルチ重点関数を禁じているというよりは、この条件はこれらの関数で分散推定値が無限になりやすいことを強調するもの。

この問題をさらに詳細に論じる前に、簡単な事例で、分散推定量が無限になると、どのような望ましくない結果を得られるかを示しておく。

無限の分散が生じる単純な設定として、目標のコーシー分布 $C(0,1)$ に正規重点関数 $N(0,1)$ を使う場合を考える。このとき、比 $f(x)/g(x) \propto \exp(x^2/2)/(1+x^2)$ は、 x が少しばかり大きくなるだけで重点重みが非常に大きくなるため、激増する。

```
set.seed(71000)
x <- rnorm(10^6)
wein <- dcauchy(x)/dnorm(x)
plot(cumsum(wein*(x > 2)*(x < 6))/cumsum(wein), type='l')
abline(a=pcauchy(6) - pcauchy(2), b=0, col="sienna")
```



累積平均の推移で反復数が大きい場合も含めて大きなジャンプが何度も繰り返されている。ジャンプが生じているのは、シミュレーションで $\exp(x^2/2)/(1+x^2)$ が大きくなる値の時であり、これは x が大きい時ということ。

この現象の原因は、これらの値が正規重点分布では稀であるため、これを相殺するために重みを大きくなるということ。例えば図中で大きく跳ね上がっているところは、 $x=5.170653$ という値をとっており、正規重み 0.03896199 をとっている。これは 100 万個の点のうち、単独の点が、全体の 3% の重みをもつことになる。とうぜん、このシミュレーションの結果を信頼することはできない。サンプルサイズが不適切であるためである。

比 f/g が有界でない場合、重点重み $f(x_j)/g(x_j)$ は広範囲に散らばりがちであり、 x_j のほんの数個の値に過度の重みを置くことで、推定量の有効性を劣化させる。上の例のように、多数の反復を行った後でさえも、特定

のシミュレーションの結果、推定値が直前のものとは突然変わってしまうようなことがある。逆に、 f よりも裾の厚い重点分布 g の場合、比 f/g の挙動が、 $E_f[h^2(X)f(X)/g(X)]$ の発散する要因にはならないことが保証される。

裾の厚い重点サンプリング提案分布は、 $(3,1)$ は存在しても $E_f[h^2(X)]$ が存在しないような関数 h の近時を検討する際には、ほとんど必須。そのような場合、 $h(X^2)$ の経験平均に分散はないので、通常のモンテカルロ法は不可能。

練習問題 3.7

f が T_ν 分布の場合に、重点関数 g と日積分関数 $h(x) = \sqrt{x/(1-x)}$ に関連付けられた重点サンプリング推定量の分散が、 $g(1) < \infty$ であるすべての g について無限であることを示す。また g の分散が有限になる必要十分条件を検討する (こっちは難しいから保留)。

被積分関数 $h(x)$ が $x=1$ で inf ty に発散するためである

この段階で一般的に勧められるのは、 $|h|g/g$ がほぼ定数であるか、あるいは少なくとも裾の挙動を制御できるような分布 g を探すこと。

重点サンプリングが、主に f が検討しにくいようなケースで適用される場合は、 f の裾に対するこうした制約は、特に次元が大きい場合には実現しにくいもの。しかし、実は一般解もある。重点関数 g に厚い裾成分を人工的に取り込む。この解法は Hesterberg による防御サンプリングと呼ばれている。これは密度 g の代わりに混合密度を使う手法。

$$\rho g(x) + (1-\rho)l(x) \quad (1 < \rho < 1)$$

ここで ρ は 1 に近く、密度 l は、必ずしも目下の問題とは関連しない裾の重い分布 (コーシー分布やパレート分布) が選ばれる。

この設定で g が与えられているとすると、裾の思い関数 l の選択はややデリケート。ベイズ的推論で目標分布

$$f$$

が事後分布を選択するのが自然。確かに、この関数は構成的に f よりも裾が重く、シミュレーションの容易な通常の分布であるのが普通。これに対して事前分布を主な重点関数 g として利用するのは、データが報知的 (informative) であれば無駄が生じるので意味をなさない。しかしながら安定因子 (stabilizing factor) としてならば意味がある。

その構成において重点サンプリング推定量は、 g と l を決めるために使われる一様変数を積分消去する。この一様分布に条件付けるとすれば変動を大きくするだけでなく、重点重みにおいても一度 $g(x)$ で割ることにより、混合系を利用することの効能を台無しにしてしまうことにもなる。

実は g と f からのシミュレーションの乱数選択は、 g から正確に、 ρn 個の x_i を、また l から $(1-\rho)$ 個の y_i を生成するとバイアスのない推定量が作られるので結局のところ不要
すなわち重点サンプリング推定量

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

はグローバルには $E_f[h(X)]$ と等しいと期待される。それゆえ、各分布から一定の点をシミュレーションするのは妥当でもある。

例

以下の積分計算はデリケートである。

$$\int_1^\infty \sqrt{\frac{x}{x-1}} t_2(x) dx = \int_1^\infty \sqrt{\frac{x}{x-1}} \frac{\Gamma(3/2)/\sqrt{2\pi}}{(1+x^2/2)^{3/2}} dx$$

関数 $h(x) = \sqrt{1/(x-1)}$ が二乗可積分でなく、したがって T_2 分布からのシミュレーションを使うと、積分のモンテカルロ推定量で分散が無限になるため。この特徴は T_2 密度と、制御しやすい l の混合刑が必要になることがわかる。

すなわち $h^2(x)f(x)/l(x)$ を積分するには、 l が $x=1$ で発散し、また l は x が無限に近づくとき x^5 よりも速く減少する必要がある。この境界条件より

$$l(x) \propto \frac{1}{\sqrt{x-1}} \frac{1}{x^{3/2}} \prod_{x>1}$$

は受理可能な密度。この密度の特徴を知りたいければ、以下を確認すればよい。

$$\int_1^y \frac{dx}{\sqrt{x-1}x^{3/2}} = \int_0^{y-1} \frac{dw}{\sqrt{w}(w+1)^{3/2}} = \int_0^{\sqrt{y-1}} \frac{2dw}{(w^2+1)^{3/2}} = \int_0^{\sqrt{2(y-1)}} \frac{2dt}{(1+t^2/2)^{3/2}}$$

これは、 $T \sim T_3$ の場合に、 $l(x)$ が $(1 + T^2/2)$ の密度に対することを示す

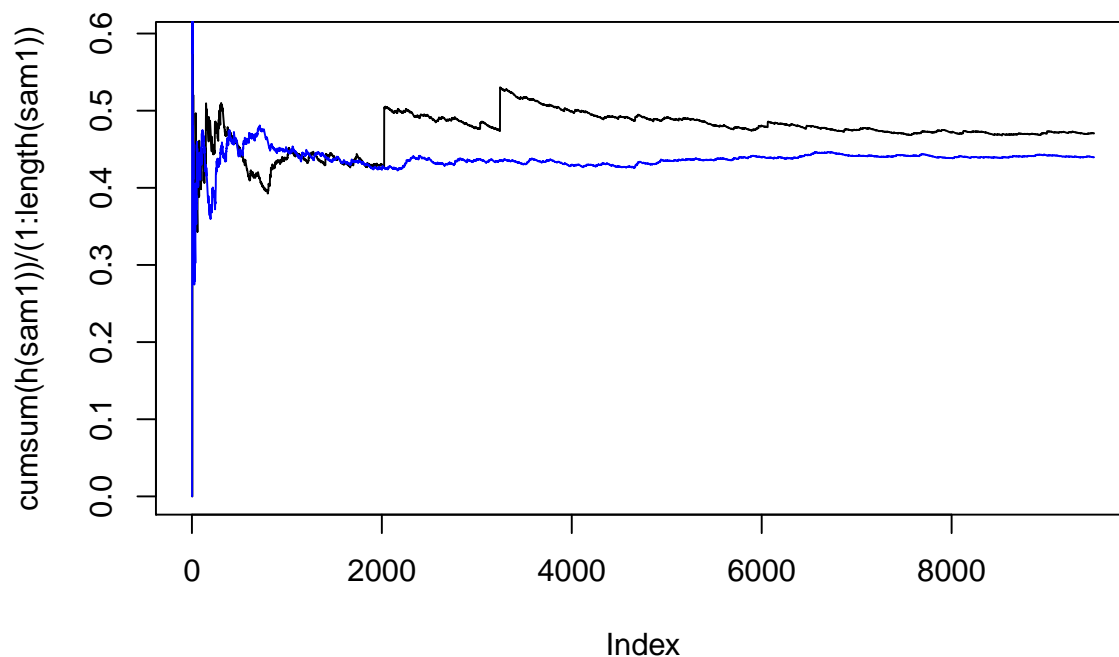
$$l(x) = \frac{\sqrt{2}\Gamma(3/2)/\sqrt{2\pi}}{\sqrt{x-1}x^{3/2}} II_{(1,\infty)}(x)$$

これが正しい正規化定数であることは `integrate()` 関数を実行すると確認できる。

防御サンプリングと最初の重点サンプラーの比較では、もとの $g=f$ からのサンプルに、 l からのサンプルを少し追加していく。

防御サンプリングが安定することを確認

```
set.seed(110)
sam1 <- rt(0.95*10^4,df=2)
sam2 <- 1 + 0.5*rt(0.05*10^4, df=2)^2
sam <- sample(c(sam1, sam2), 0.95*10^4)
weit <- dt(sam, df=2)/(0.95*dt(sam, df=2) + 0.05*(sam>0)*
            dt(sqrt(2*abs(sam-1)), df=2)*sqrt(2)/sqrt(abs(sam-1)))
h <- function(a) apply(a,function(x) if(x>1) sqrt(x/(x-1)) else 0)
plot(cumsum(h(sam1))/(1:length(sam1)), ty='l')
lines(cumsum(weit*h(sam))/1:length(sam1), col='blue')
```



続いて、ベイズ的観点からプロビット・モデルを検討する。プロビット・モデルは一般化線形モデルにおいて、観測値 y が二項変数で 0 か 1 のいずれかの値であり、共変量が以下のようなベクトル $x \in R^p$ の場合に該当する

$$Pr(y = 1|x) = 1 - Pr(y = 0|x) = \Phi(xT\beta) \quad (\beta \in R^p)$$

このモデルのデータは容易にシミュレーションできるが、ここでは MASS パッケージの Pima.tr データセットを利用する。200 名の女性について糖尿病疾患の有無が目的変数

```
model1 <- glm(type ~ bmi, data=Pima.tr,
              family=binomial(link="probit"))
summary(model1)

##
## Call:
## glm(formula = type ~ bmi, family = binomial(link = "probit"),
##      data = Pima.tr)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5788 -0.9242 -0.6442  1.2506  1.9661
##
## 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
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 239.55  on 198  degrees of freedom
## AIC: 243.55
##
## Number of Fisher Scoring iterations: 4
```

実行結果から、MNI 共変量が糖尿病の発症に有意に影響していることがわかる。(まあ交絡とか一旦無視で) ベイズ的な観点では、正規分布 $N(0, 100)$ のパラメータ $\beta = (\beta_1, \beta_2)$ に曖昧事前分布を導入する。すると β の事後分布は、本質的にフラットな事前分布と尤度の積となる。尤度の関数

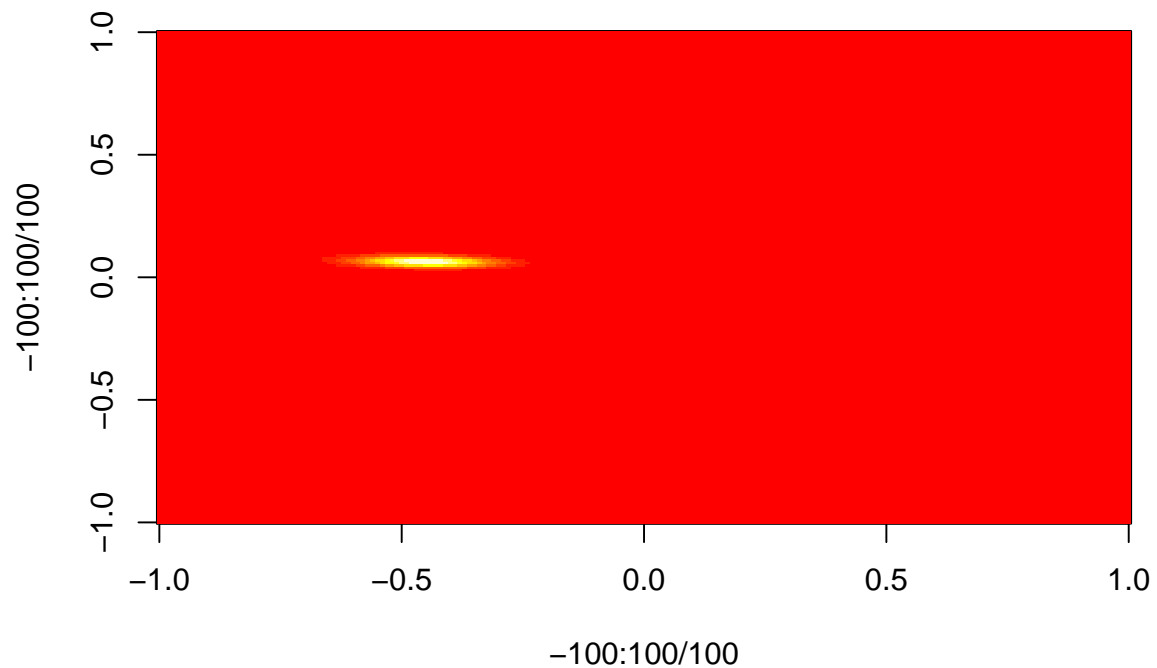
```
like <- function(beda){
  mia <- mean(Pima.tr$bmi)
  prod(pnorm(beda[1] + (Pima.tr$bmi[Pima.tr$type == 'Yes']
    - mia) * beda[2])) *
  prod(pnorm(-beda[1] - (Pima.tr$bmi[Pima.tr$type == 'No']
    - mia) * beda[2])) / exp(sum(beda^2) / 200)
}
```

最尤法による解確認

並列処理できるように関数が書かれていない

ちょっと時間かかるけどしょうがない

```
outer_ <- matrix(0, ncol=201, nrow=201)
Num <- -100:100/100
for(i in 1:201){for(j in 1:201){outer_[i,j] <- like(c(Num[i], Num[j]))}}
image(-100:100/100, -100:100/100, outer_)
```

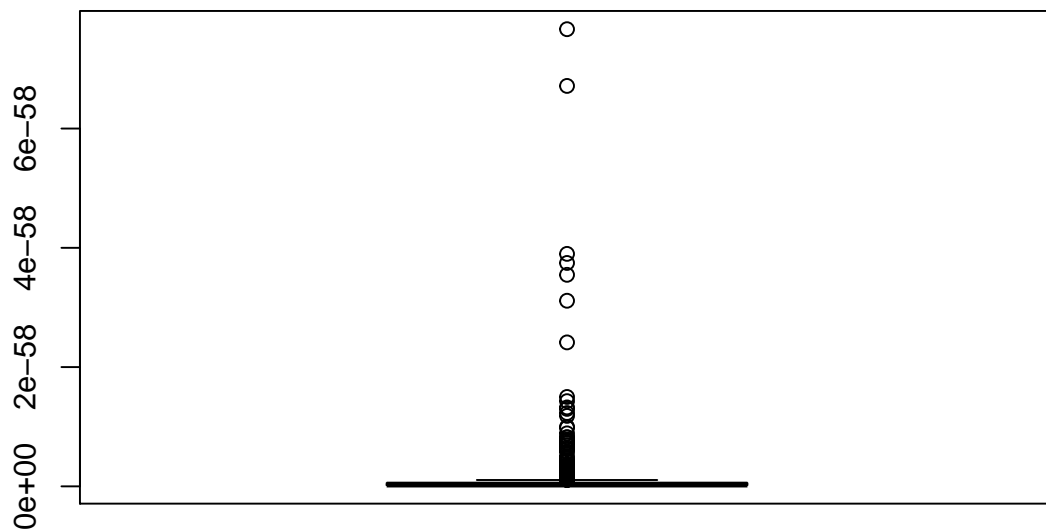


glm 関数による推定値に対応する共分散対角行列と最尤法を中心とする正規分布が、提案分布 g の選択になる。

ただし、この選択があらゆるケースにおいて有限な分散を保障するものではない。

この考えを以下の様に実装してみる。

```
post<-function(x){ dnorm(x[1],0,100) * dnorm(x[1],0,100) * like(x)}
sim <- cbind(rnorm(10^3, mean=-0.4, sd=0.04),
rnorm(10^3, mean=0.065, sd=0.005))
weit <- apply(sim, 1, post)/
(dnorm(sim[,1], mean=-0.4, sd=0.04) *
dnorm(sim[,2], mean=0.065, sd=0.005))
boxplot(weit)
```



すると、重点重みが一様ではないことが分かる。

防御サンプリングの実装すると

```
sim <- rbind(sim[1:(0.95)*10^3,],
cbind(rnorm(0.05*10^3, sd=10),rnorm(0.05*10^3, sd=10)))
weit <- apply(sim, 1, post)/
(0.95*dnorm(sim[,1],m=-0.4,sd=0.081)*
  dnorm(sim[,2], mean=0.065, sd=0.01) +
  0.05*dnorm(sim[,1],sd=10)*
  dnorm(sim[,2], sd=10))
boxplot(weit)
```

