MATH 598: TOPICS IN STATISTICS

THE DIRICHLET PROCESS AND THE BAYESIAN BOOTSTRAP

The Dirichlet Process (DP) is a probability distribution on the space of discrete distributions on $\mathbb{R}$. In light of a random sample of data $y_1, \ldots, y_n$, we have that under a $DP(\alpha, G_Y)$ prior, the posterior distribution can be shown to be a Dirichlet process, $DP(\alpha^\star, G_Y^\star)$, where

$$\alpha^\star = \alpha + n \qquad G_Y^\star(y) = \frac{\alpha}{\alpha + n} G_Y(y) + \frac{1}{\alpha + n} \sum_{j=1}^{n} \delta_{\{y_j\}}(y)$$

In the limit as $\alpha \longrightarrow 0$, $G_Y^\star(y)$ becomes the empirical distribution

$$G_Y^\star(y) = \frac{1}{n} \sum_{j=1}^{n} \delta_{\{y_j\}}(y)$$

and therefore a sampled value of the random distribution $\widetilde{f}(y)$ is obtained by sampling weights $W = (W_1, \ldots, W_n)$, with $W \sim Dirichlet(n, 1, 1, \ldots, 1)$, and attaching these weights to $y_1, \ldots, y_n$. Conditional on a realization w of W, we have a representation of the predictive distribution $p_n(.)$ derived from a Dirichlet process model as

$$p_n(y) = \sum_{i=1}^{n} w_i \delta_{\{y_i\}}(y).$$

For maximum likelihood estimation in a potentially mis-specified model $f(y; \theta)$, we identify the true value of θ, θ_0 as

$$\theta_0 = \arg\min_\theta KL(f_0, f(.; \theta)) = \arg\min_\theta \int \log \left\{ \frac{f_0(y)}{f(y; \theta)} \right\} f_0(y)\, dy.$$

An estimator is obtained when we replace the integral by a 'Monte Carlo' version based on an i.i.d. sample

$$\widehat{\theta} = \arg\max_\theta \sum_{i=1}^{n} \log f(y_i; \theta) = \arg\max_\theta \ell_n(\theta)$$

In alternative form, $\widehat{\theta}$ is the solution to the estimating equation

$$\dot{\ell}_n(\theta) = \sum_{i=1}^{n} \frac{\partial \log f(y_i; \theta)}{\partial \theta} = 0.$$

A Bayesian version of the calculation replaces the original sample by a sample from the predictive distribution $p_n(y_{(n+1):(n+m)})$. However, we are not restricted to use the 'score' function as the basis of an estimation procedure: we may use any loss function $L(y, \theta)$ say, and define the Bayesian estimator as

$$\arg\min_\theta \int L(y, \theta) p_n(y)\, dy = \arg\min_\theta \mathbb{E}_{p_n}[L(Y, \theta)].$$

This is a valid fully Bayesian estimator as it minimizes an expected posterior loss; via this route, we may achieve fully Bayesian inference in a semi-parametric fashion. This leads to the calculation

$$\mathbb{E}_{p_n}[L(Y, \theta)] = \sum_{i=1}^{n} w_i L(y_i, \theta) = -\sum_{i=1}^{n} w_i \ell(y_i; \theta)$$

in the specific case of the log-density loss. Hence, we must perform the calculation of

$$\theta_{\text{OPT}} = \arg\max_\theta \sum_{i=1}^{n} w_i \ell(y_i; \theta)$$

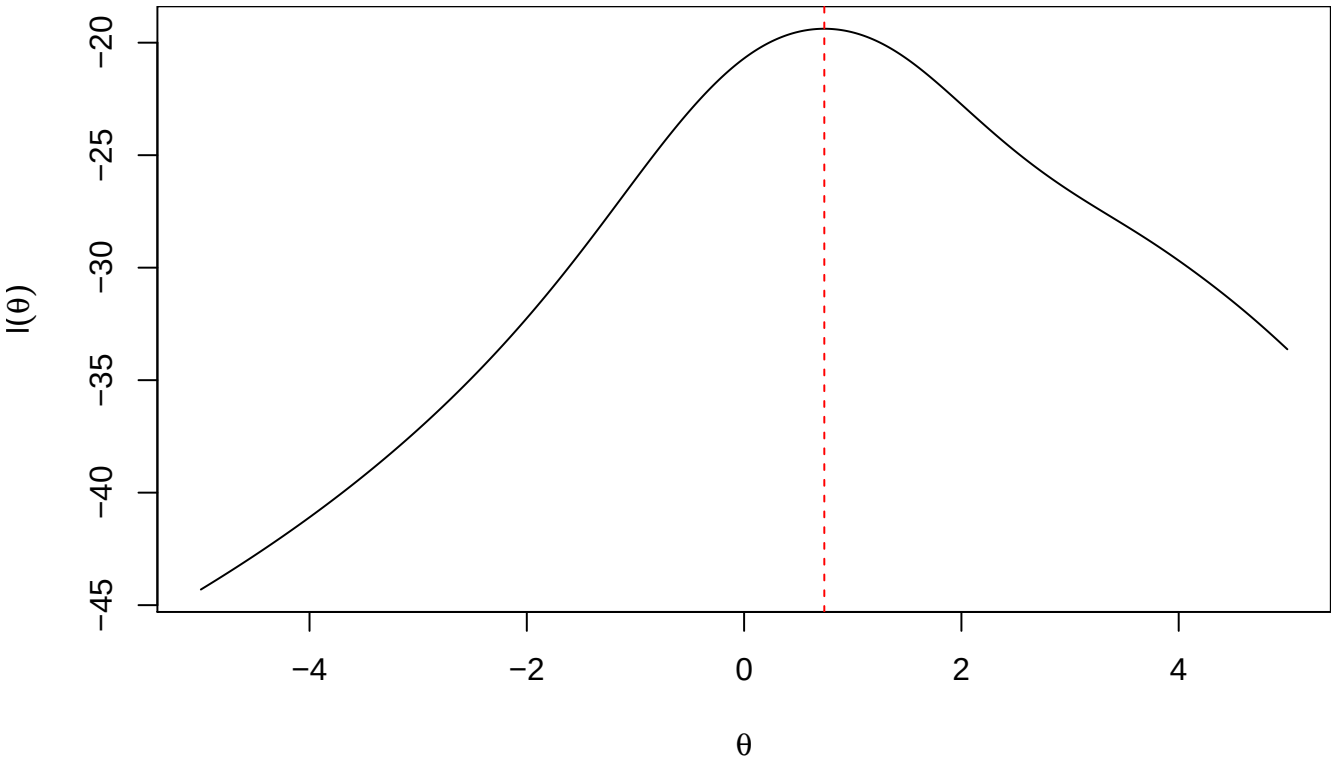
to minimize the loss. The quantity θ_{OPT} is a thus functional of the Dirichlet process posterior, and so we may build up a posterior distribution for it by repeatedly sampling the Dirichlet weights, and recomputing θ_{OPT} for each sample.

EXAMPLE: Suppose that $Y_1, \ldots, Y_n$ are a random sample from the Cauchy location family

$$f_Y(y;\theta) = \frac{1}{\pi}\frac{1}{1+(y-\theta)^2} \quad y \in \mathbb{R}$$

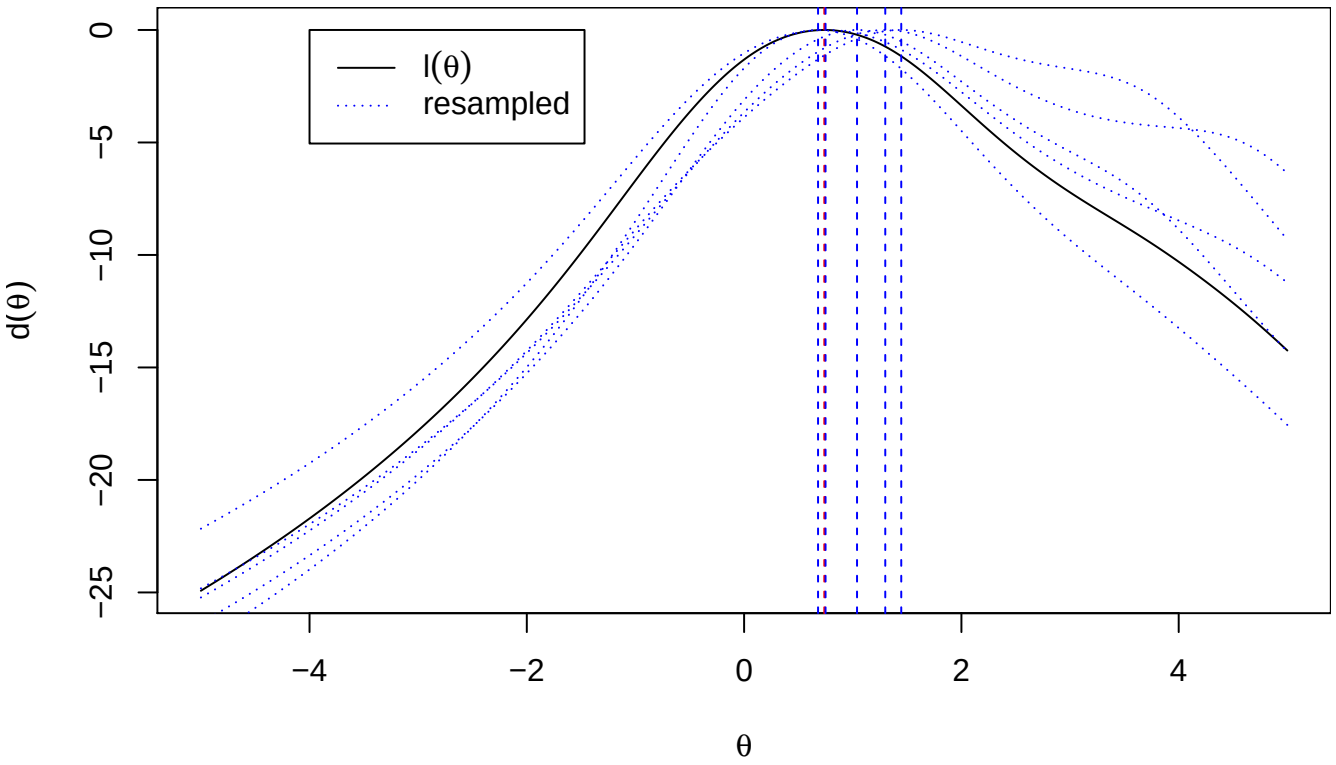
The log-likelihood from a sample of size $n = 10$ is plotted below, along with the maximizing value.

```
set.seed(7838)
n<-10
y<-rcauchy(n)
wlike<-function(th,yv,wv){
    logl<--wv*log(1+(yv-th)^2)
    return(sum(logl))
}
w<-rep(1,n)
opt0<-optimize(f=wlike,interval=c(-100,100),yv=y,wv=w,maximum=TRUE)
thvec<-seq(-5,5,by=0.001);llvec<-thvec*0
for(i in 1:length(thvec)){llvec[i]<-wlike(thvec[i],y,w)}
par(mar=c(4,4,0,0))
plot(thvec,llvec,type='l',xlab=expression(theta),ylab=expression(l(theta)))
abline(v=opt0$maximum,lty=2,col='red')
```



Using the Bayesian bootstrap procedure, we can simulate $B = 5$ resampled versions of the likelihood.

```
B<-5
par(mar=c(4,4,0,0))
plot(thvec,llvec-opt0$objective,type='l',xlab=expression(theta),ylab=expression(d(theta)))
abline(v=opt0$maximum,lty=2,col='red')
for(b in 1:B){
    w<-rgamma(n,1);w<-w/sum(w)
    opt1<-optimize(f=wlike,interval=c(-100,100),yv=y,wv=w,maximum=TRUE)
    for(i in 1:length(thvec)){llvec[i]<-wlike(thvec[i],y,w)}
    lines(thvec,n*(llvec-opt1$objective),lty=3,col='blue')
    abline(v=opt1$maximum,lty=2,col='blue')
}
legend(-4,0,c(expression(l(theta)),'resampled'),col=c('black','blue'),lty=c(1,3))
```



In this plot,

- each log-likelihood is plotted relative to its maximum value, that is

$$d(\theta) = \ell_n(\theta) - \ell_n(\widehat{\theta})$$

  is plotted.

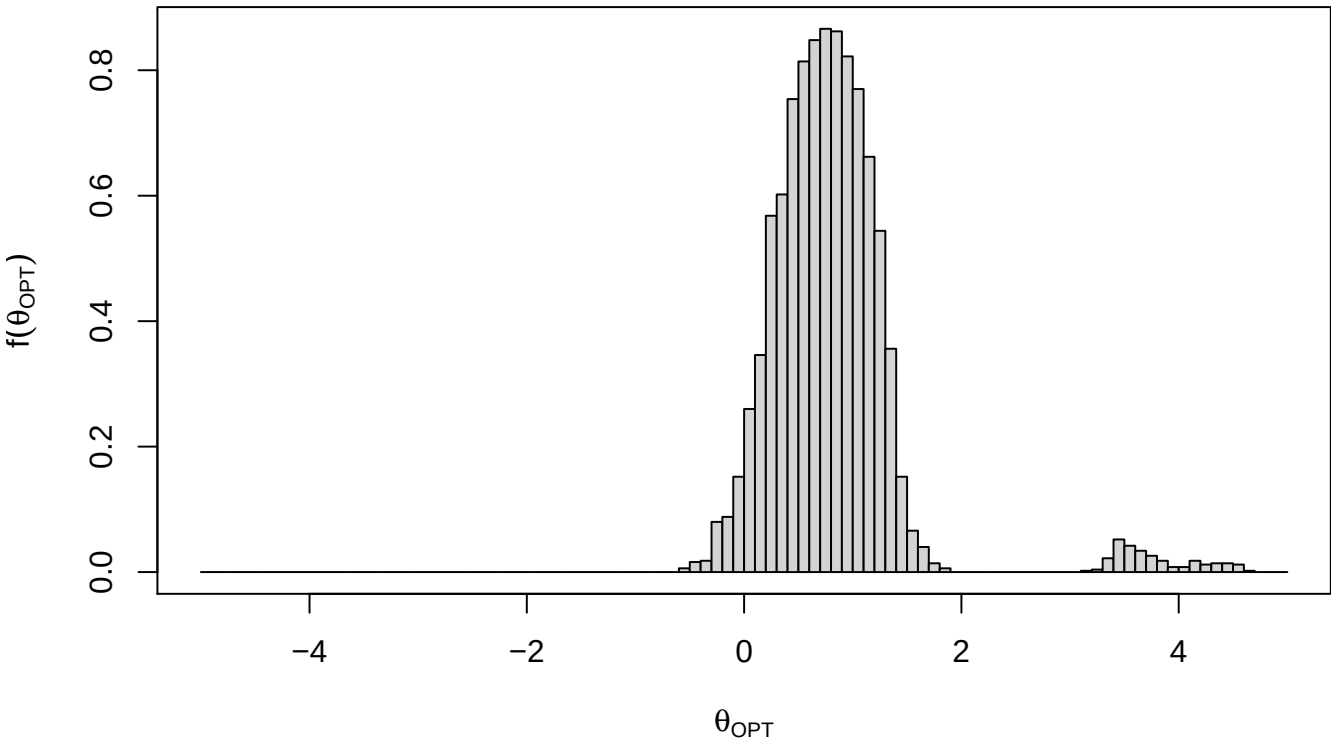
- for the Dirichlet re-weighted log-likelihoods, to match scales correctly, the objective function is multiplied by n, that is

$$d_W(\theta) = n \sum_{i=1}^{n} w_i \ell(y_i; \theta)$$

- the position of the maximizing value θ_{OPT} is plotted as a vertical blue dashed line for each of the re-weighted cases, and as a vertical solid red line for the parametric maximum likelihood estimator.

Using $B = 5000$, we may generate from the posterior distribution for θ_{OPT}.

```
B<-5000
th.samp<-rep(0,B)
for(b in 1:B){
    w<-rgamma(n,1);w<-w/sum(w)
    opt1<-optimize(f=wlike,interval=c(-100,100),yv=y,wv=w,maximum=TRUE)
    th.samp[b]<-opt1$max
}
par(mar=c(4,4,3,0))
hist(th.samp,freq=FALSE,breaks=seq(-5,5,by=0.1),main=' ',
             xlab=expression(theta[OPT]),ylab=expression(f(theta[OPT])))
box()
```



In this problem, if we specify a uniform prior for θ on $(-5, 5)$, then

$$\pi_n(\theta) \propto \prod_{i=1}^{n} \frac{1}{1 + (y_i - \theta)^2} \qquad -5 < \theta < 5$$

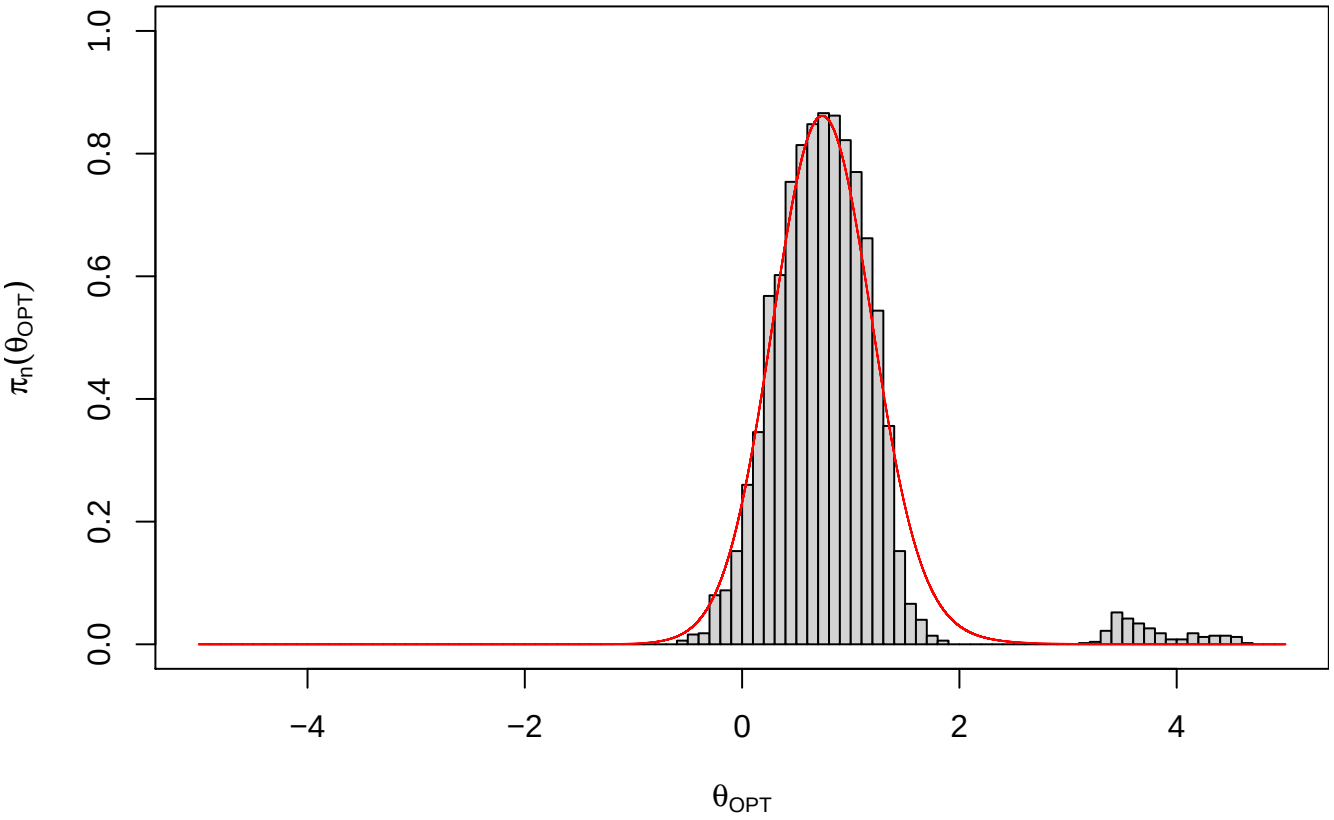
and can compute the normalizing constant using numerical approximation, for example, Simpson's rule, and from this compute the posterior under correct specification in which case $\theta_{\text{OPT}} = \theta_0 = 0$. Simpson's rule uses the approximation

$$\int_{x_0}^{x_N} g(x)\, dx \doteq \frac{(x_N - x_0)}{3N} \left[g(x_0) + 4 \sum_{j=1}^{N/2} g(x_{2j-1}) + 2 \sum_{j=1}^{N/2-1} g(x_{2j}) + + g(x_N) \right]$$

and in the calculation below, we use $N = 500000$, and

$$h = \frac{(x_N - x_0)}{N} = 0.00001.$$

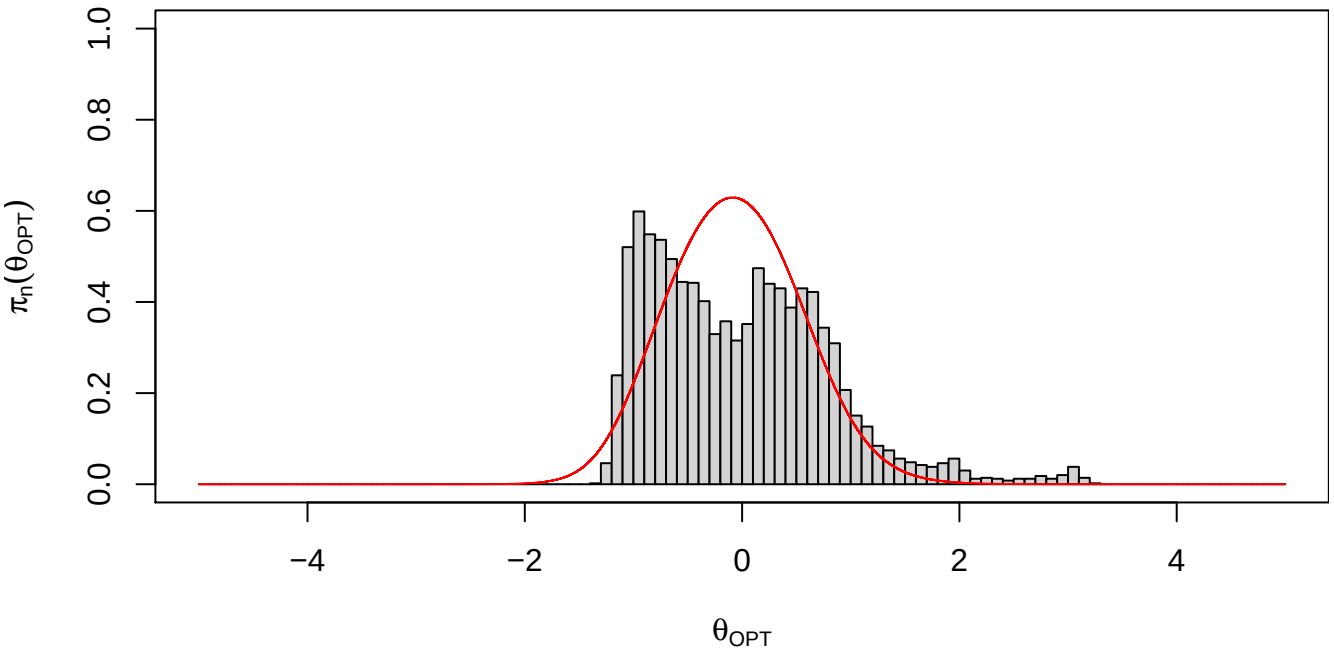
```
w<-rep(1,n)
h<-0.00001
thvec<-seq(-5,5,by=h);llvec<-thvec*0
for(i in 1:length(thvec)){llvec[i]<-wlike(thvec[i],y,w)}
like.n<-exp(llvec-max(llvec))
halfn<-(length(thvec)-1)/2
evens<-c(1:(halfn-1))*2
odds<-c(1:halfn)*2-1
const<-(like.n[1]+4*sum(like.n[odds])+2*sum(like.n[evens])+like.n[length(thvec)])*h/3
post.n<-like.n/const
par(mar=c(4,4,1,0))
hist(th.samp,freq=FALSE,breaks=seq(-5,5,by=0.1),main=' ',ylim=range(0,1),
              xlab=expression(theta[OPT]),ylab=expression(pi[n](theta[OPT])))
box()
lines(thvec,post.n,col='red')
```



When the sample size is small, the two methods can differ appreciably.

```
set.seed(784638)
n<-10
y<-rcauchy(n)
wlike<-function(th,yv,wv){
    logl<--wv*log(1+(yv-th)^2)
    return(sum(logl))
}
w<-rep(1,n)
opt0<-optimize(f=wlike,interval=c(-100,100),yv=y,wv=w,maximum=TRUE)
thvec<-seq(-5,5,by=0.001);llvec<-thvec*0
for(i in 1:length(thvec)){llvec[i]<-wlike(thvec[i],y,w)}
B<-5000
th.samp<-rep(0,B)
for(b in 1:B){
```

```
    w<-rgamma(n,1);w<-w/sum(w)
    opt1<-optimize(f=wlike,interval=c(-100,100),yv=y,wv=w,maximum=TRUE)
    th.samp[b]<-opt1$max
}
w<-rep(1,n)
h<-0.00001
thvec<-seq(-5,5,by=h);llvec<-thvec*0
for(i in 1:length(thvec)){llvec[i]<-wlike(thvec[i],y,w)}
like.n<-exp(llvec-max(llvec))
halfn<-(length(thvec)-1)/2
evens<-c(1:(halfn-1))*2
odds<-c(1:halfn)*2-1
const<-(like.n[1]+4*sum(like.n[odds])+2*sum(like.n[evens])+like.n[length(thvec)])*h/3
post.n<-like.n/const
par(mar=c(4,4,3,0))
thv<-th.samp[th.samp > -5 & th.samp < 5]
hist(thv,freq=FALSE,breaks=seq(-5,5,by=0.1),main=' ',ylim=range(0,1),
              xlab=expression(theta[OPT]),ylab=expression(pi[n](theta[OPT])));box()
lines(thvec,post.n,col='red')
```



When the sample size is $n = 100$, the two methods coincide more closely.

```
set.seed(784638)
n<-100
y<-rcauchy(n)
wlike<-function(th,yv,wv){
    logl<--wv*log(1+(yv-th)^2)
    return(sum(logl))
}
w<-rep(1,n)
opt0<-optimize(f=wlike,interval=c(-100,100),yv=y,wv=w,maximum=TRUE)
thvec<-seq(-5,5,by=0.001);llvec<-thvec*0
for(i in 1:length(thvec)){llvec[i]<-wlike(thvec[i],y,w)}
B<-5000
th.samp<-rep(0,B)
```

```
for(b in 1:B){
    w<-rgamma(n,1);w<-w/sum(w)
    opt1<-optimize(f=wlike,interval=c(-100,100),yv=y,wv=w,maximum=TRUE)
    th.samp[b]<-opt1$max
}
w<-rep(1,n)
h<-0.00001
thvec<-seq(-2.5,2.5,by=h);llvec<-thvec*0
for(i in 1:length(thvec)){llvec[i]<-wlike(thvec[i],y,w)}
like.n<-exp(llvec-max(llvec))
halfn<-(length(thvec)-1)/2
evens<-c(1:(halfn-1))*2
odds<-c(1:halfn)*2-1
const<-(like.n[1]+4*sum(like.n[odds])+2*sum(like.n[evens])+like.n[length(thvec)])*h/3
post.n<-like.n/const
par(mar=c(4,4,3,0))
thv<-th.samp[th.samp > -2.5 & th.samp < 2.5]
hist(thv,freq=FALSE,breaks=seq(-2.5,2.5,by=0.05),main=' ',ylim=range(0,3.5),
              xlab=expression(theta[OPT]),ylab=expression(pi[n](theta[OPT])));box()
lines(thvec,post.n,col='red')
```