In the name of God

Producer:

Mehrab Atighi

Subject:

Monte carlo method for Normal-Cuashy bayes estimator

Date:

2020/27/12

Supervisor:

Dr.Seyed Noorullah Mousavi

**Issue:**

For the normal-Cauchy Bayes estimator

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.

**Solve:**

We know that according to the Monte carlo methods we have:

step1 :we set the , the caushy normal distrubtion Caushu(0,1)

step2 :we set the other functions infront of the integral,g(x)

so we have:

so now we set the

then:

Forwe have:

> rm(list=ls())

> memory.limit(size=999999999)

[1] 1e+09

> x=c(0,2,4)

> N=10^5

> g1<-c();g2<-c()

> g.1<-c();g.2<-c()

> delta<-matrix(c(rep(0,3\*N)),ncol = N)

> theta<-rcauchy(N)

> par(mfrow=c(1,3))

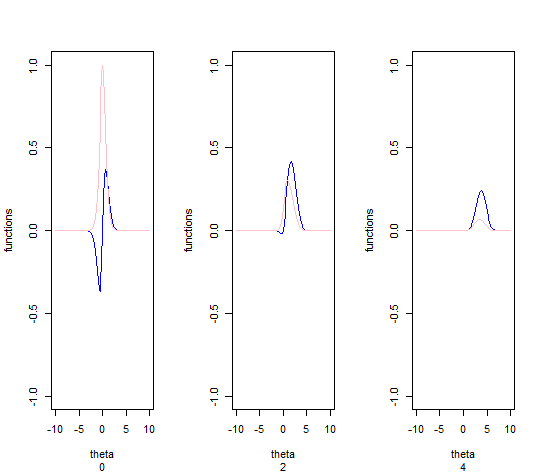
> for(i in 1:3){

+ f.1=function(theta){ theta/(1+(theta)^2)\*exp(-(x[i]-theta)^2/2)}

+ f.2=function(theta){ 1/(1+(theta)^2)\*exp(-(x[i]-theta)^2/2)}

+ plot(f.1,xlab="theta",ylab="functions",type ="l" ,xlim = c(-10,10),ylim=c(-1,1) ,col="blue" , sub=x[i])

+ plot(f.2,add=TRUE,type="l" , col="pink" , xlim = c(-10,10))}



+ for(i in 1:N){

+ g1[i]<-theta[i]\*exp(-(1/2\*(x[j]-theta[i])^2))

+ g2[i]<-1\*exp(-(1/2\*(x[j]-theta[i])^2))

+ delta[j,i]<-mean(g1[1:i])/mean(g2[1:i])}

+ print(paste("the delta(x) value with x=",x[j],"is:",mean(g1)/mean(g2)))}

[1] "the delta(x) value with x= 0 is: -0.00233317554413187"

[1] "the delta(x) value with x= 2 is: 1.27888732761474"

[1] "the delta(x) value with x= 4 is: 3.42736175946439"

> o<-c()

> for(j in 1:3){

+ g.1<-function(t){(t/(1+(t^2)))\*exp(-((1/2)\*(x[j]-t)^2))}

+ g.2<-function(t){(1/(1+(t^2)))\*exp(-((1/2)\*(x[j]-t)^2))}

+ print(o[j]<-(integrate(g.1,-Inf,Inf)$val)/(integrate(g.2,-Inf,Inf)$val))}

[1] 0

[1] 1.282195

[1] 3.435062

> for(j in 1:3){

+ r<-which((delta[j,]=="NaN"))

+ for(w in 1:length(r)){

+ delta[j,r[w]]<-o[j]}}

> (sd<-apply(delta,1,sd))

[1] 0.01037757 0.01340714 0.02649938

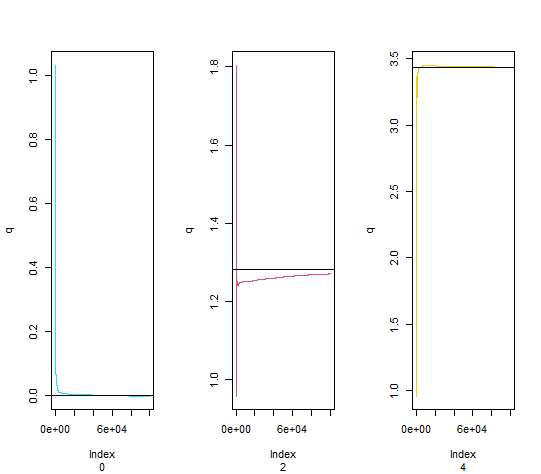
> par(mfrow=c(1,3))

> for(j in 1:3){

+ q<-cumsum(delta[j,])/1:N

+ plot(q,type="l" ,sub = x[j],col=85\*j,xlim=c(0,N),ylim=c(min(q),max(q)))

+ abline(h=o[j] ,col="black")}



**Now we want to make it for normal:**

==

Now if

If we set ,

So We have :

The two front function plot is same with cuashy plot.(so I don’t show it again.)

Now we want to estimate ,(according to up)with Normal-standard density function:

> #normal part:

> rm(list=ls())

> x=c(0,2,4)

> N=10^5

> h1<-c();h2<-c()

> h.1<-c();h.2<-c()

> delta2<-matrix(c(rep(0,3\*N)),ncol = N)

> for(i in 1:3){

+ f.1=function(theta){ theta/(1+(theta)^2)\*exp(-(x[i]-theta)^2/2)}

+ f.2=function(theta){ 1/(1+(theta)^2)\*exp(-(x[i]-theta)^2/2)}

+ plot(f.1,xlab="theta",ylab="functions",type ="l" ,xlim = c(-10,10),ylim=c(-1,1) ,col=85 , sub=x[i])

+ plot(f.2,add=TRUE,type="l" , col="purple" , xlim = c(-10,10))}

> for( j in 1:3){theta<-rnorm(N,mean = 0,sd=1)

+ for(i in 1:N){

+ h1[i]<-(((theta[i])/(1+(theta[i])^2))\*(exp((-1/2\*(theta[i])^2)+(theta[i]\*x[j]))))

+ h2[i]<-(((1)/(1+(theta[i])^2))\*(exp((-1/2\*(theta[i])^2)+(theta[i]\*x[j]))))

+ delta2[j,i]<-mean(h1[1:i])/mean(h2[1:i])}

+ print(paste("the delta(x) value with x=",x[j],"is:",mean(h1)/mean(h2)))}

[1] "the delta(x) value with x= 0 is: 0.000377986859489263"

[1] "the delta(x) value with x= 2 is: 0.685799582291693"

[1] "the delta(x) value with x= 4 is: 1.58736740680525"

> o<-c()

> for(j in 1:3){

+ h.1<-function(t){(t/(1+(t^2)))\*exp(-((1/2)\*(x[j]-t)^2))}

+ h.2<-function(t){(1/(1+(t^2)))\*exp(-((1/2)\*(x[j]-t)^2))}

+ print(o[j]<-(integrate(h.1,-Inf,Inf)$val)/(integrate(h.2,-Inf,Inf)$val))}

[1] 0

[1] 1.282195

[1] 3.435062

> for(j in 1:3){

+ r<-which((delta2[j,]=="NaN"))

+ for(w in 1:length(r)){

+ delta2[j,r[w]]<-o[j]}}

> (sd<-apply(delta2,1,sd))

[1] 0.007764371 0.007211865 0.036078876

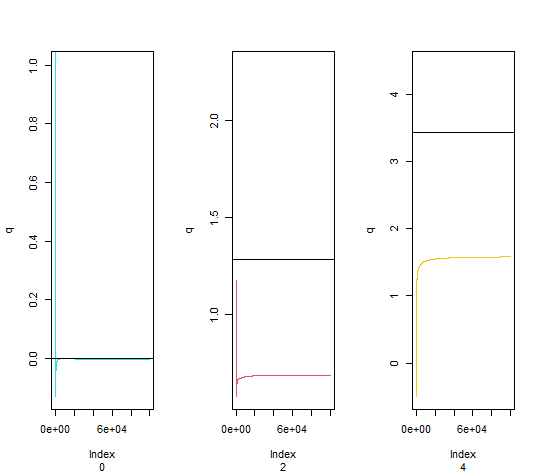
> par(mfrow=c(1,3))

> for(j in 1:3){

+ q<-cumsum(delta2[j,])/1:N

+ plot(q,type="l" ,sub = x[j],col=85\*j,xlim=c(0,N),ylim=c(min(q),o[j]+1))

+ abline(h=o[j] ,col="black")}



**Conclusion:** the estiamate for with normal-standard distrubtion is not good but the Caushy estimate is good for that,we can see that in estimate with cuashy random variable converges very quickly to .