computational_statistics_lab03_group14

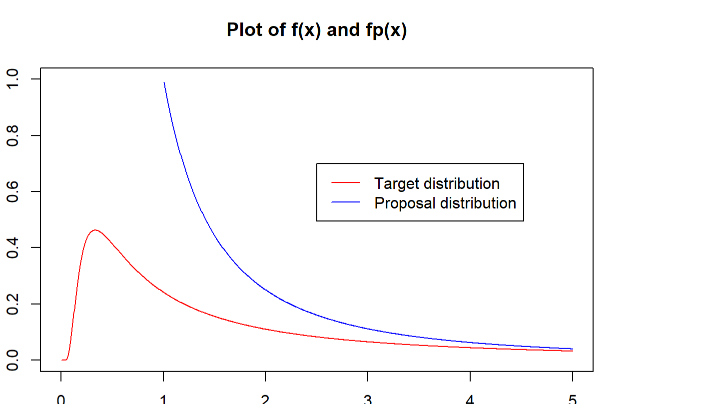
Akshath Srinivas, Samira Goudarzi

2022-11-28

Question 1

subquestion 1

The below plot is drawn for function $f(x)$ when $c = 1$ and $f_p(x)$ when $\alpha = 2$ and $T_{min} = 1$, as we can observe from the plot that there is a problem when when $x < T_{min}$. So we cannot use only one density function $f_p x$(power-law distribution) to generate samples.



Implementing another density function for support of x from range (0,tmin).

The mixture density can be found by below method.

$$M = \int_0^\infty f_m(x) = \int_0^{T_{min}} f_q(x)dx + \int_{T_{min}}^\infty f_p(x)dx$$

Calculating $f_p(x)$ from (T_{min}, ∞)

$$\int_{T_{min}}^\infty f_p(x)dx = \int_{T_{min}}^\infty \frac{\alpha - 1}{T_{min}} (\frac{x}{T_{min}})^{-\alpha} = 1$$

when $x = T_{min}$ we get $f_q(x) = \frac{\alpha-1}{T_{min}}$ which is a normalizing constant of power-law distribution we use this as $f_q(x)$ to sample from $(0, T_{min})$

$$\int_0^{T_{min}} f_q(x)dx = \int_0^{T_{min}} \frac{\alpha - 1}{T_{min}} = \alpha - 1$$

Calculating M

$$M = 1 + \alpha - 1 = \alpha$$

We can get mixture of density functions by $f_m(x)/\alpha$

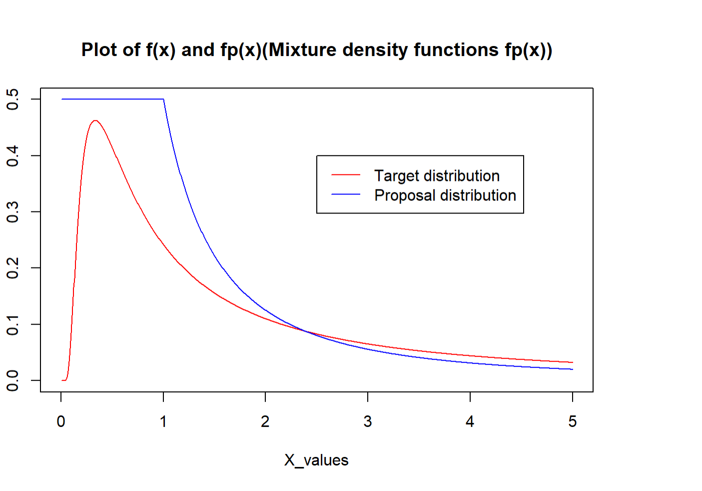
Solving further,we get majorizing density function below for x values between $(0, T_{min})$

$$\frac{\alpha - 1}{T_{min}\alpha}$$

Solving further,we get majorizing density function below for x values between (T_{min}, ∞)

$$\frac{\alpha - 1}{T_{min}\alpha} (\frac{x}{T_{min}})^{-\alpha}$$

The below plot is drawn for $c = 1$, $\alpha = 2$ and $T_{min} = 1$ by using the mixture density functions calculated above.



subquestion 2

Implementing acceptance and rejection method to generate samples after sampling the values from **runif()** and **rplcon()** function. We accept the sample if the below condition is satisfied, otherwise we reject it.

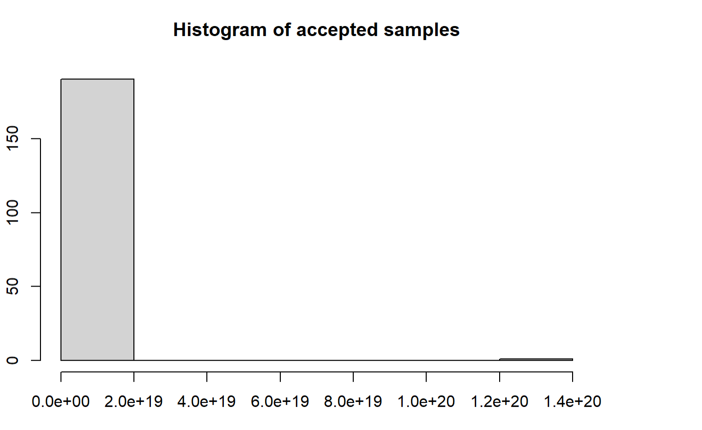
$$u <= f_X(y)/(mc * f_Y(y))$$

where u is generated from $runif(1)$, $f_X(y)$ is target stable distribution which is f_s_f(x,c) in our case, mc is majorizing constant and $f_Y(y)$ is the proposal distribution having two density functions(mixture density), which is f_p_a(x,t_min,alpha) in our case and presenting the accepted samples on histogram.

calculating probabilities for sampling.

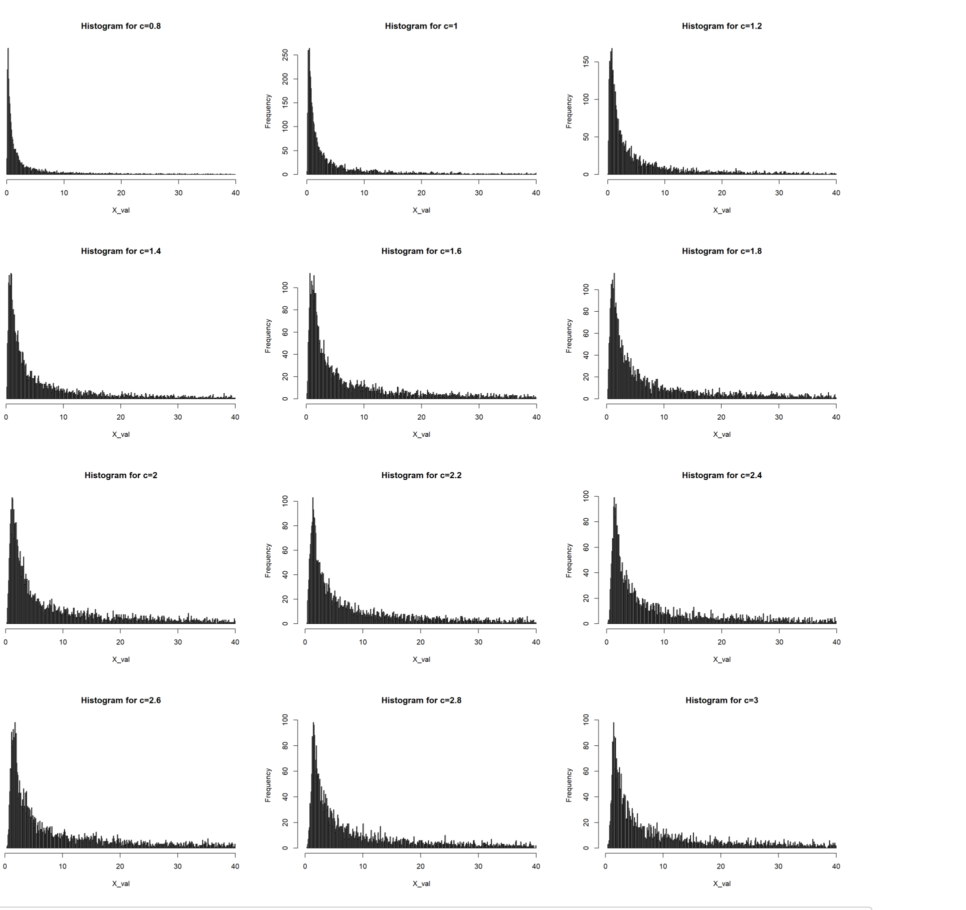
$$\int_0^{T_{min}} \frac{\alpha - 1}{T_{min}\alpha} = \frac{\alpha - 1}{\alpha}$$

$$\int_{T_{min}}^\infty \frac{\alpha - 1}{T_{min}\alpha} (\frac{x}{T_{min}})^{-\alpha} = \frac{1}{\alpha}$$



subquestion 3

Generating the samples from the implemented sampler in sub question 2 for c values ranging from 0.8 to 3 with step of 0.2 and presenting the accepted sample values on histogram.




```
##    c_values         mean     variance rejection_rate majorizing_constant
## 1       0.8 4.268962e+03 2.213876e+10      0.4836484           1.9513448
## 2       1.0 2.989256e+04 3.264996e+12      0.4454445           1.7793103
## 3       1.2 1.938119e+04 1.485385e+12      0.4627463           1.8345941
## 4       1.4 2.244449e+04 9.178191e+11      0.4694469           1.7890121
## 5       1.6 1.713396e+04 3.306561e+11      0.4480448           1.6624600
## 6       1.8 1.562987e+05 1.157391e+14      0.4289429           1.4793439
## 7       2.0 9.677137e+04 3.875279e+13      0.4291429           1.2647699
## 8       2.2 3.989123e+04 3.171733e+12      0.4047405           1.0413784
## 9       2.4 5.095318e+04 2.416728e+12      0.3888389           0.8272199
## 10      2.6 1.911939e+06 2.089367e+16      0.3768377           0.6347845
## 11      2.8 7.799798e+06 3.501928e+17      0.3535354           0.4710573
## 12      3.0 4.037910e+08 1.000817e+21      0.3495350           0.3383134
```

Mean and variance increases as we increase c values.

As we can see above, c parameter in target distribution has impact on majorizing constant. We observed that when majorizing constant is less the rejection rate is also less, but when majorizing constant is large rejection rate is also large.

Question 2

subquestion 1

Laplace distribution formula is given by.

$$DE(\mu, \sigma) = \frac{\alpha}{2} exp(-\alpha|x - \mu|)$$

For DE(0,1), we get below distribution when we substitute μ and σ as 0 and 1 respectively.

$$DE(0, 1) = \frac{1}{2} e^{-|x|}$$

We get CDF when x<0 or μ.

$$\int_{-\infty}^x \frac{1}{2} e^x dx = \frac{1}{2} e^x$$

We get CDF when x>=0 or μ.

$$\int_{-\infty}^0 \frac{1}{2} e^x dx + \int_0^x \frac{1}{2} e^{-x} dx = 1 - \frac{1}{2} e^x$$

Set y from uniform distribution, according to inverse CDF, we get below for y<0.5.

$$y = \frac{1}{2} e^x$$

Further solving, we get.

$$x = ln(2y)$$

Set y from uniform distribution, according to inverse CDF, we get below for y>=0.5

$$y = 1 - \frac{1}{2} e^x$$

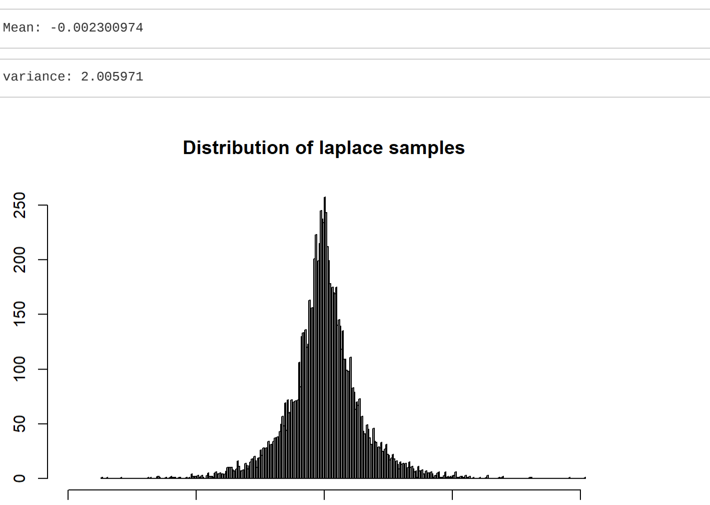
Further solving, we get.

$$x = -ln(2 - 2y)$$

The histogram plotted below from the samples looks like the probability density function of Laplace distribution. The mean and variance of the Laplace distribution should be 0 and 2, mean and variance calculated for our samples is close to these values.

```
## Mean: -0.002300974
```

```
## variance: 2.005971
```



subquestion 2

Getting majorizing constant c value.

$$c * DE(x) >= N(x)$$

Substituting density function.

$$c * \frac{1}{2} e^{-|x|} >= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

Further solving, we get.

$$c >= \frac{2e^{-\frac{1}{2}x^2}}{\sqrt{2\pi} e^{-|x|}}$$

When x>=0.

$$c >= \frac{2e^{-\frac{1}{2}x^2 - x}}{\sqrt{2\pi}}$$

When x<0

$$c >= \frac{2e^{-\frac{1}{2}x^2 + x}}{\sqrt{2\pi}}$$

Applying ln on both sides for above two equations to get $-\frac{1}{2}x^2 - x$ and$-\frac{1}{2}x^2 + x$. The maximum value of these equations will be is 1/2. So, on solving we get **c>=1.3155.**

Acceptance and rejection method is done by taking 2000 samples of distribution(Laplace) from the function we implemented in sub question 1. A random sample will be generated from runif(1), if the condition below satisfies then we accept the sample otherwise we reject it.

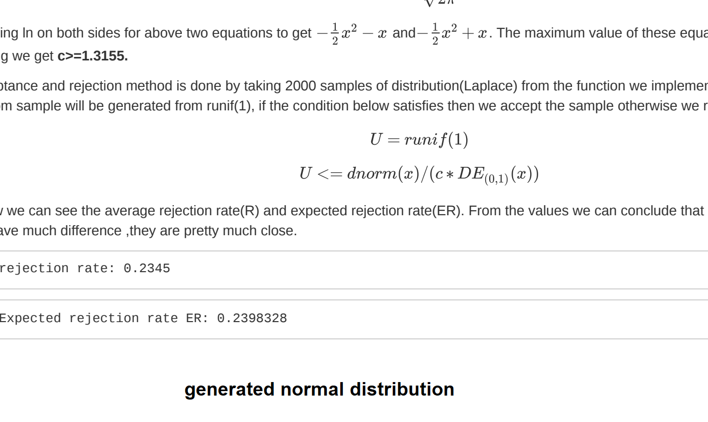
$$U = runif(1)$$

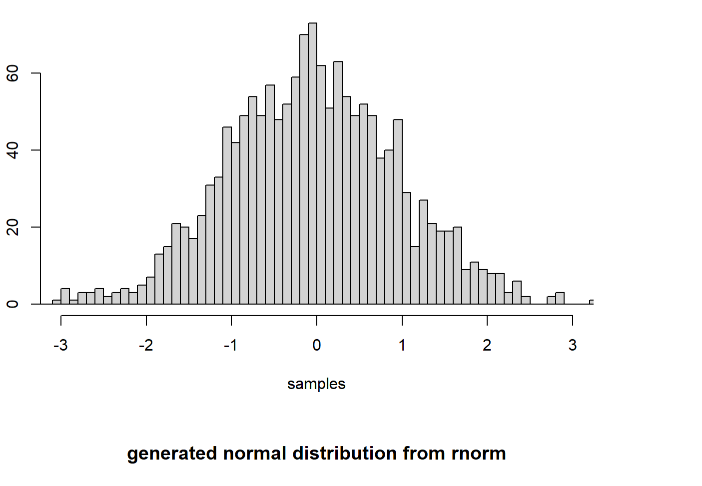
$$U <= dnorm(x)/(c * DE_{(0,1)}(x))$$

Below we can see the average rejection rate(R) and expected rejection rate(ER). From the values we can conclude that R calculated and ER does not have much difference ,they are pretty much close.

```
## rejection rate: 0.2345
```

```
## Expected rejection rate ER: 0.2398328
```





The two histograms generated above from acceptance/rejection method and rnorm(2000,0,1) have similar distribution as samples.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
library(poweRlaw)

##### question 1 #####

# target distribution
f_s_f<-function(x,c){
  f_s_v<-c()
  for(i in 1:length(x)){
    f_s<- c*((sqrt(2*pi))^-1)*exp((-c^2)/(2*x[i]))*(x[i]^(-3/2))
    f_s_v<-c(f_s_v,f_s)
  }
  return(f_s_v)
}

# proposal distribution

f_p_f<-function(x,t_min,alpha){
  f_p_v<-c()
  for(i in 1:length(x)){
    f_p<-((alpha-1)/t_min)*(x[i]/t_min)^(-alpha)
    f_p_v<-c(f_p_v,f_p)
  }
  return(f_p_v)
}
fn<-f_p_f(seq(1.005,5,by=0.005),1,2)

plot(seq(0.005,5,0.005),f_s_f(seq(0.005,5,0.005),1),type='l',col='red',xlim=c(0,5),ylim=c(0,1),xlab='X_values',yl
ab='Function_values',main='Plot of f(x) and fp(x)')
lines(seq(1.005,5,by=0.005),fn,col='blue')
legend (2.5,0.7,legend=c('Target distribution','Proposal distribution'),col = c('red','blue'),lty = rep(1,2))


#Two density functions, after deriving majorizing density
f_p_a<-function(x,t_min,alpha){
  f_p<-c()
  f_p_l<-c()
  for(i in 1:length(x)){
    if(x[i]>t_min){
      f_p<-c(f_p,(alpha-1)/t_min*(x[i]/t_min)^(-alpha))
    }else{
      f_p_l<-c(f_p_l,((alpha-1)/t_min))
    }
  }
  return(c(f_p_l,f_p)/alpha)
}

#t_min is 0.45
plot(seq(0.005,5,0.005),f_s_f(seq(0.005,5,0.005),1),type='l',col='red',xlim=c(0,5),ylim=c(0,0.5),xlab='X_values',
ylab='Function_values',main='Plot of f(x) and fp(x)(Mixture density functions fp(x))')
lines(seq(0.005,5,0.005),f_p_a(seq(0.005,5,0.005),1,2),col='blue')
legend (2.5,0.4,legend=c('Target distribution','Proposal distribution'),col = c('red','blue'),lty = rep(1,2))



#sampling
sampling<-function(samples,alpha,t_min){
  n1 <- samples*(alpha-1)/(alpha)
  n2 <- samples - n1
  left <- runif(n1,0,t_min)
  right <- rplcon(n2,t_min,alpha)
  final_samples<-c(left,right)
  return(final_samples)
}

# accept and rejection method
x_val<-function(sampling,mc,alpha,t_min,c){
  x_val<-c()
  for(i in 1:length(sampling)){
    u<-runif(1)
    px<-f_s_f(sampling[i],c)
    if(sampling[i] <= t_min)
    {
      gy <- (alpha-1)/t_min/alpha
    }else
    {
      gy <- f_p_f(sampling[i],t_min,alpha)/alpha
    }
    if(u<=px/(mc*gy)){
      x_val<-c(x_val,sampling[i])
    }
  }
  return(x_val)
}

alpha<-1.2
t_min<-0.45
#majorizing constant
mc<-max(f_s_f(seq(0.0001,2,0.0001),5)/f_p_a(seq(0.0001,2,0.0001),0.45,1.2))
y<-x_val(sampling(200,1.2,0.45),mc,3,0.45,1.1)
hist(y,xlab='X_values',main = 'Histogram of accepted samples')


#for different choices of c
rej<-list()
x_list<-list()
i<-0
diff_c<-function(c,alpha,t_min){
  mc_v<-c()
  for (i in 1:length(c)){
    mc<-max(f_s_f(seq(0.0001,t_min+1,0.0001),c[i])/f_p_a(seq(0.0001,t_min+1,0.0001),t_min,alpha))
    mc_v<-c(mc_v,mc)
    final_samples<-sampling(10000,1.2,0.45)
    x<-x_val(final_samples,mc,alpha,t_min,c[i])
    .GlobalEnv$i<- .GlobalEnv$i+1
    .GlobalEnv$x_list[[i]]<-x
    .GlobalEnv$rej[[i]]<-(length(final_samples)-length(x))/length(final_samples)

  }
  return(list(x_list,rej,mc_v))
}
x<-diff_c(seq(0.8,3,by=0.2),1.2,0.45)


hist(x[[1]][[1]][which(x[[1]][[1]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=0.8')
hist(x[[1]][[2]][which(x[[1]][[2]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=1')
hist(x[[1]][[3]][which(x[[1]][[3]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=1.2')


hist(x[[1]][[4]][which(x[[1]][[4]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=1.4')
hist(x[[1]][[5]][which(x[[1]][[5]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=1.6')
hist(x[[1]][[6]][which(x[[1]][[6]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=1.8')


hist(x[[1]][[7]][which(x[[1]][[7]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=2')
hist(x[[1]][[8]][which(x[[1]][[8]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=2.2')
hist(x[[1]][[9]][which(x[[1]][[9]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=2.4')


hist(x[[1]][[10]][which(x[[1]][[10]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=2.6')
hist(x[[1]][[11]][which(x[[1]][[11]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=2.8')
hist(x[[1]][[12]][which(x[[1]][[12]]<=40)],breaks = 400,xlab = 'X_val',main='Histogram for c=3')


#majorizing constants
majorizing_constant<-x[[3]]
#means
cal_mean<-sapply(x[[1]],mean)
#variances
cal_var<-sapply(x[[1]],var)

df<-data.frame(c_values=seq(0.8,3,by=0.2),mean=cal_mean,variance=cal_var,rejection_rate=unlist(x[[2]]),majorizing
_constant=x[[3]])
df

##### question 2 #####

#Getting Laplace samples distribution from samples of unif(0,1) by inverse CDF method
inverse_cdf_laplace<-function(x){
  inv<-c()
    for(i in 1:length(x)){
      # uniform distribution values which is greater than mu(mean of uniform distribution) uses below condition
      if (x[i] >= 0.5 ){
        inv <- c(inv,-log(2-(2*x[i])))
      }
      # less than mu uses below condition
      else{
        inv <- c(inv,log(2*x[i]))
      }
    }
  return(inv)
}
#calculating mean and variance
mean<-mean(inverse_cdf_laplace(runif(10000,0,1)))
cat('Mean:',mean)
variance<-var(inverse_cdf_laplace(runif(10000,0,1)))
cat('variance:',variance)

# distribution of samples
hist(inverse_cdf_laplace(runif(10000,0,1)),breaks=400,xlim=c(-10,10),xlab = 'Laplace_samples',
     main = 'Distribution of laplace samples')


#acceptance and rejection method

c<-1.3155
acc_rej<-function(x){
  reject<-0
  x_val<-c()
  for(i in 1:2000){
    r<-runif(1)
    if(r<=dnorm(x[i])/(c*(1/2 *exp(-abs(x[i]))))){
      x_val<-c(x_val,x[i])
    }else{
      reject<-reject+1
    }
  }
  return(list(x_val,reject))
}
laplace_samples<-inverse_cdf_laplace(runif(2000,0,1))
x<-acc_rej(laplace_samples)

#average rejection rate
rejection_rate<-x[[2]]/2000
cat('rejection rate:',rejection_rate)

#expected rejection rate
ER<-1-1/c
cat('Expected rejection rate ER:',ER)


hist (x[[1]],breaks=50,xlim=c(-3,3),xlab = 'samples',main='generated normal distribution')


r<-rnorm(2000,mean=0,sd=1)
hist(r,breaks = 50, main='generated normal distribution from rnorm')
```