

## HW 4 (Due Nov 17, in D2L)

### 1) Finite Mixture Model

In a finite Mixture model, the marginal distribution of a random variable (RV, say  $x$ ) is a weighted sum of a finite number of densities. The following is an example of a mixture of three normal densities

$$f(x) = \pi_1 N(x|\mu_1, \sigma_1^2) + \pi_2 N(x|\mu_2, \sigma_2^2) + (1 - \pi_1 - \pi_2) N(x|\mu_3, \sigma_3^2) \quad (1)$$

where  $N(x|\mu_*, \sigma_*^2) = \frac{e^{-\frac{(x-\mu_*)^2}{2\sigma_*^2}}}{\sqrt{2\pi\sigma_*^2}}$

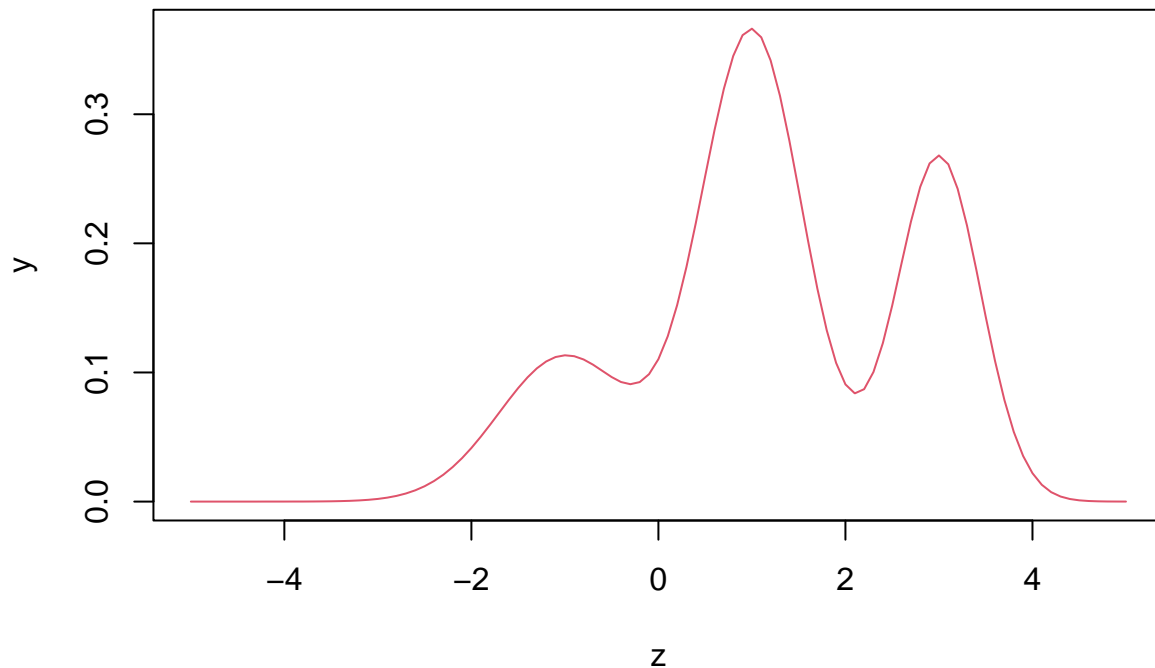
The following is a density plot from the above mixture with:

- $\mu_1 = -1, \sigma_1^2 = 0.5$ ,
- $\mu_2 = 1, \sigma_2^2 = 0.3$ ,
- $\mu_3 = 3, \sigma_3^2 = 0.2$ , and
- $\pi_1 = .2, \pi_2 = .5$ .

```
n=100000
mu=c(-1,1,3)
v=c(.5,.3,.2)
pi=c(.2,.5,.3)

f=function(x,prop,means,variances){
  nComponents=length(prop)
  n=length(x)
  y=rep(0,n)
  for(i in 1:nComponents){
    y=y+prop[i]*dnorm(x=x,mean=means[i],sd=sqrt(variances[i]))
  }
  return(y)
}

z=seq(from=-5,to=5,by=.1)
y=f(z,pi,mu,v)
plot(y~z,col=2,type='l')
```



A mixture model can also be represented using the joint distribution of two RVs  $p(x, z) = p(x|z)p(z)$  where  $z$  is an indicator variable for the mixture component ( $z \in \{1, 2, 3\}$ ,  $p(z = 1) = \pi_1$ ,  $p(z = 2) = \pi_2$ , and  $p(z = 3) = 1 - \pi_1 - \pi_2$ ) and  $p(x|z) = N(x|\mu_z, \sigma_z^2)$ .

The marginal distribution of  $x$  is

$$p(x) = p(z = 1)N(x|\mu_1, \sigma_1^2) + p(z = 2)N(x|\mu_2, \sigma_2^2) + p(z = 3)N(x|\mu_3, \sigma_3^2)$$

which is the same as expression (1).

#### Task:

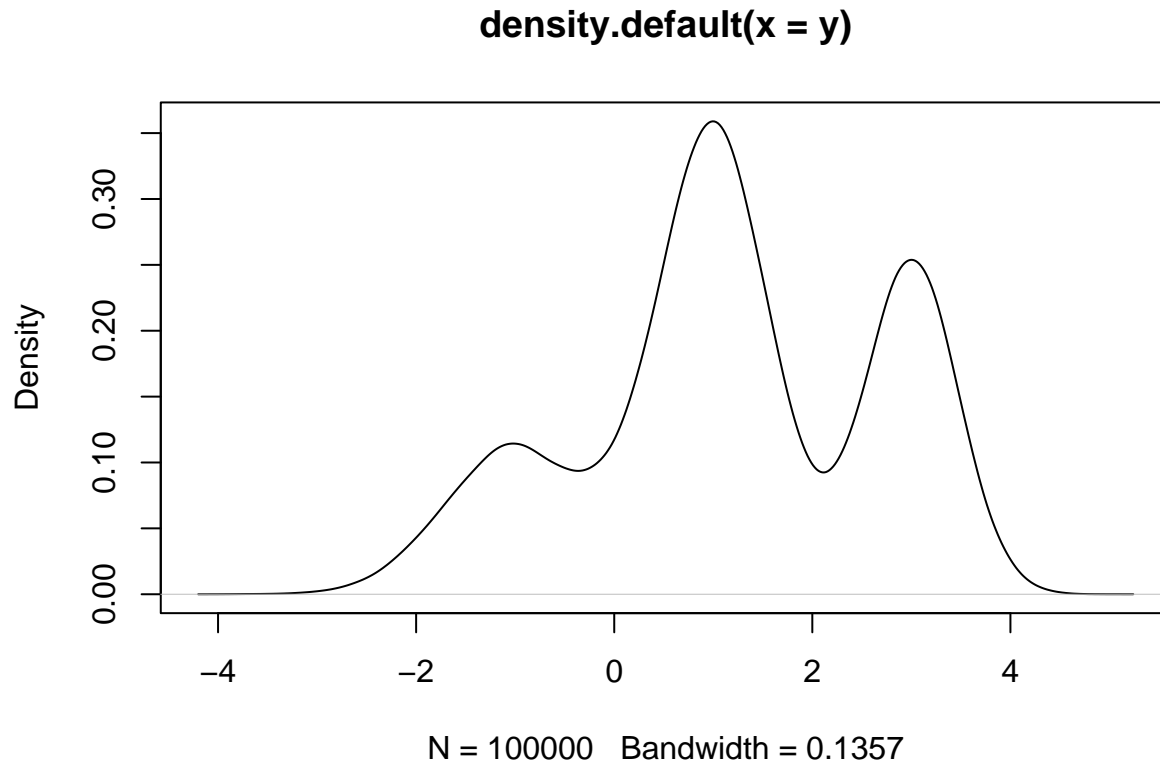
- Use composition sampling to generate 100,000 samples from  $p(z, x) = p(x|z)p(z)$ .
- Plot the empirical density of  $x$  using `plot(density(x))` where  $x$  are the samples you generated.
- Hint:  $z$  is a multinomial RV that can take values 1, 2, and 3 with probabilities  $\pi_1$ ,  $\pi_2$ , and  $\pi_3$ , then you can sample it using

```
pi_1=0.2
pi_2=0.5
pi_3=1-pi_1-pi_2

z=sample(1:3,prob=c(pi_1,pi_2,pi_3),size=n,replace=TRUE)
```

#### Solution to Problem 1

```
n=100000
u=runif(n)
z=ifelse(u<pi[1],1,ifelse(u<sum(pi[1:2]),2,3))
y=rnorm(n)*sqrt(v[z])+mu[z]
plot(density(y))
```



## 2) The Metropolis Algorithm

Let

- $p(x)$  be the density of  $x$  (or a function proportional to it), and
- $q(x_t|x_{t-1})$  be a symmetric distribution satisfying  $q(x_t|x_{t-1}) = q(x_{t-1}|x_t)$

In Metropolis, to generate samples from  $p(x)$  we:

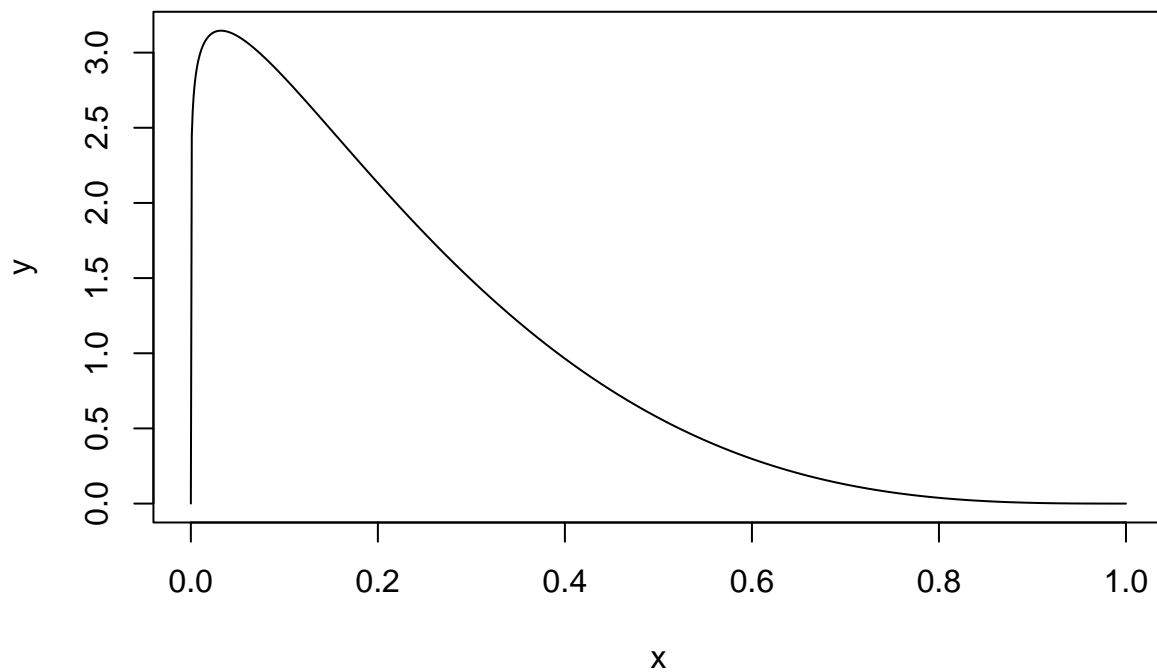
- Initialize  $x_0$  with some value with non-zero probability and then,
- In a loop for  $t$  in  $1:n$  we:
  - Generate a candidate  $w \sim q(x_t|x_{t-1})$
  - Compute the acceptance ratio  $\alpha = \min\{1, f(w)/f(x_{t-1})\}$
  - With probability  $\alpha$  set  $x_t = w$  and with probability  $1 - \alpha$  set  $x_t = x_{t-1}$ .

Example:

- Target density beta with shape1=1.1, and shape2=4 `dbeta(shape1, shape2)`
- $q(x_t|x_{t-1}) = Uniform(0, 1)$

### The Beta Density

```
x=seq(from=0,to=1,length=1000)
y=dbeta(shape1=1.1,shape2=4,x=x)
plot(y~x,type='l')
```

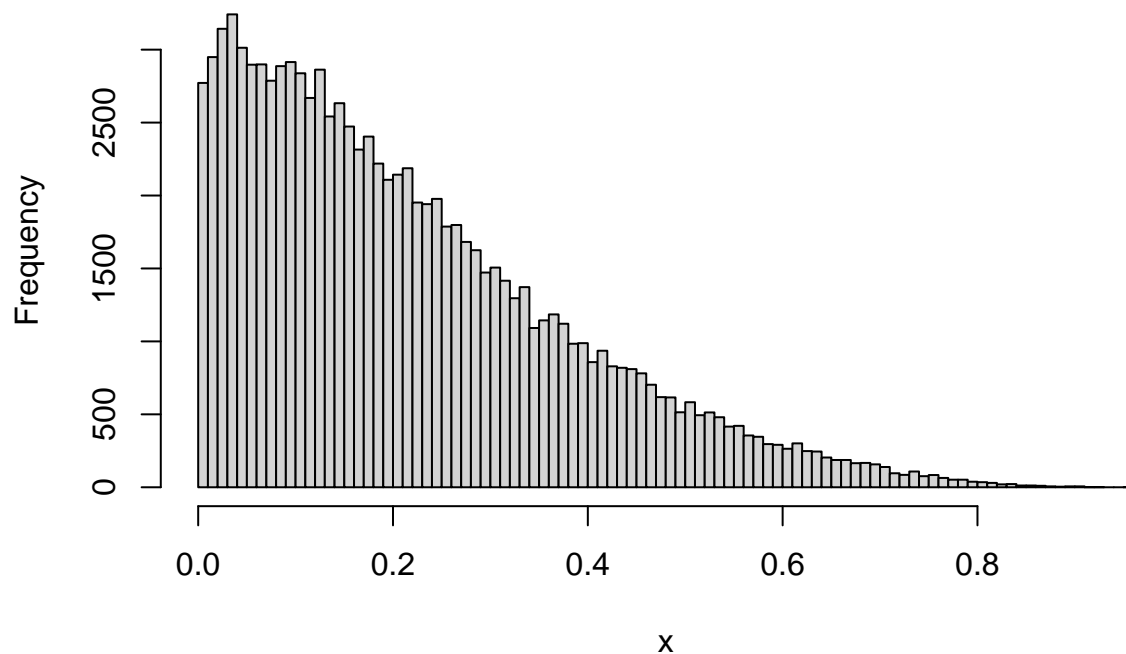


Sampling from a Beta distribution using Metropolis

```
n=100000
x=rep(NA,n)
x[1]=.5

for(i in 2:n){
  candidate=runif(1)
  alpha=min(1,dbeta(shape1=1.1,shape2=4,x=candidate)/dbeta(shape1=1.1,shape2=4,x=x[i-1]))
  z=runif(1)<alpha
  x[i]=ifelse(z,candidate,x[i-1])
}
hist(x,100)
```

## Histogram of x



### Tasks:

- Generate 100,000 samples from the mixture distribution of the first problem using  $N(x[t-1], 1)$  to generate candidates.
- Hint: Create a function to evaluate the target density: `f=function(x,proportions, means, variances){ ...}`. Internally, the function should evaluate the mixture density for  $x$ , which is a weighted sum of `dnorm()` (see expression [1]).

### Solution to problem 2

```
n=100000

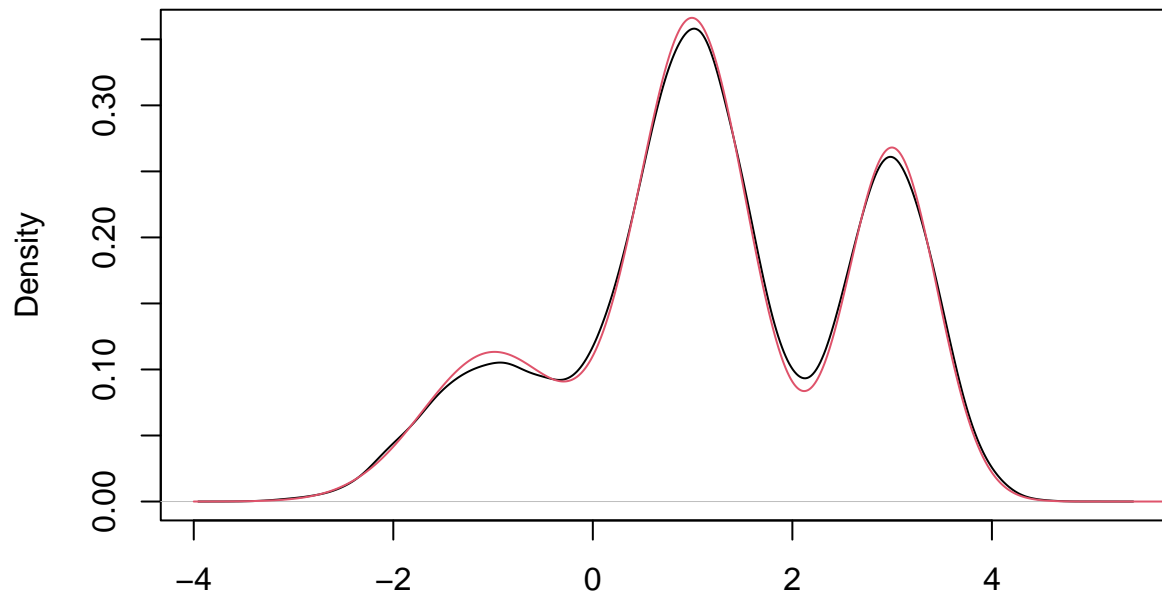
y=rep(NA,n)
y[1]=0

for(i in 2:n){
  candidate=rnorm(mean=y[i-1],1) # here I use as the mean of the candidate generator to be the previous value
  acceptanceProb=min(1,f(candidate,pi,mu,v)/f(y[i-1],pi,mu,v))
  accept=(runif(1)<acceptanceProb)
  if(accept){
    y[i]=candidate
  }else{
    y[i]=y[i-1]
  }
}

plot(density(y))
```

```
z=seq(from=-4,to=8,by=.01)
y2=f(z,pi,mu,v)
lines(x=z,y=y2,col=2)
```

**density.default(x = y)**



N = 100000 Bandwidth = 0.135

```
plot(y,xlim=c(1,300),type='o')
```

