260677676_Assignment_3_MATH447

Dan Yunheum Seol

4.19

$$T := \min\{n : Z_n = 0\}$$

be the extinction time for a branching process.

Show that

$$P(T = n) = G_n(0) - G_{n-1}(0) \quad n \geq 1$$

We show it by induction.

Base case: We defined $G_0(s) = s$ and $G_1(s) = G(s)$. Assume $Z_0 = 1$. It follows that

$$Z_n = \sum_{i=1}^{Z_{n-1}} X_i$$

We obtain

$$Z_1 = X_1$$

and

$$P(T = 1) = P(\min n : Z_n = 0 = 1) = P(Z_1 = 0) = P(X_1 = 0) = G(0) = \\ G(0) - 0 = G_1(0) - G_0(0)$$

Inductive hypothesis: Suppose the claim holds for some $n = 1, ..., k$

Inductive step: We need to show

$$P(T = k + 1) = G_{k+1}(0) - G_k(0)$$

Remark that

$$P(T \leq k) = \sum_{i=1}^{k} P(T = i) =$$

By inductive hypothesis, this is a telescoping series:

$$(G_k(0) - G_{k-1}(0)) + (G_{k-1}(0) - G_{k-2}(0)) + ... + (G_1(0) - G_0(0)) = \\ G_k(0) - G_0(0) = G_k(0) - 0 = G_k(0)$$

Remark

$$G_{k+1}(0) = P(Z_{k+1} = 0) = P(Z_{k+1} = 0, Z_k = 0) + P(Z_{k+1} = 0, Z_k \neq 0) = \\ P(Z_k = 0) + P(Z_{k+1} = 0, Z_k \neq 0) = G_k(0) + P(T = k + 1) \implies \\ G_{k+1}(0) - G_k(0) = P(T = k + 1)$$

So our claim is proved $\square$.

4.25

Let $T_n = \sum_{i=0}^{n} Z_i$ be the total number of individuals up to generation n.

$$\phi_n(s) = E[s^{T_n}]$$

be the pgf of T_n.

(a)

Show that

$$\phi_n(s) = sG(\phi_{n-1}(s))$$

$$\phi_n(s) = E[s^{T_n}] = E[s^{\sum_{i=0}^{n} Z_i}]$$

We show it by induction.

Base case: We assume $Z_0 = 1$ We know

$$T_0 = Z_0 = 1$$

implying

$$\phi_0(s) = E[s^1] = sE[1] = s$$

$$\phi_1(s) = E[s^{T_1}] = E[s^{Z_0+Z_1}] = E[s^{Z_0+1}] = E[ss^{Z_0}] = sE[s^{Z_1}] =$$

$$sE[s^{X_1}] = sE[s^X] = sG(s) = sG(\phi_0(s))$$

Inductive hypothesis:

Suppose the recurrence relation holds for $n = 1, ..., k$.

Inductive step:

We need to show that

$$\phi_{k+1}(s) = sG(\phi_k(s))$$

We know that

$$\phi_{k+1}(s) = E[s^{T_k+Z_{k+1}}] = E[s * s^{\sum_{i=1}^{k+1} Z_i}] = sE[E[s^{\sum_{i=1}^{k+1} Z_i}|Z_1]] = sE[s^{\sum_{i=1}^{k+1} Z_i}] =$$

$$s(E[s^{\sum_{i=1}^{k+1} Z_i}|Z_1 = 0] * P(Z_1 = 0) + E[s^{\sum_{i=1}^{k+1} Z_i}|Z_1 > 0] * P(Z_1 > 0)) =$$

Since if $Z_1 = 0 \implies Z_j = 0$ for all $j \geq 2$.

$$s(1 * P(Z_1 = 0) + E[s^{\sum_{i=1}^{k+1} Z_i}|Z_1 > 0] * P(Z_1 > 0)) =$$

$$s(G(0) + \sum_{j=1}^{\infty} E[s^{\sum_{i=1}^{k+1} Z_i}|Z_1 = j] * P(Z_1 = j)) =$$

$$s(\sum_{j=0}^{\infty} E[s^{\sum_{i=1}^{k+1} Z_i}|Z_1 = j] * P(Z_1 = j))$$

Remark that

$$E[s^{\sum_{i=1}^{k+1} Z_i}|Z_1 = j]$$

is like calculating another total progeny $\tilde{T}_i = T_i - 1$ but now with different number of starting individuals.

4.16(a) states that since starting individuals make offsprings independently with each other

$$E[s^{\sum_{i=1}^{k+1} Z_i}|Z_1 = j] = \phi_k(s)^j$$

It follows that

$$s(\sum_{j=0}^{\infty} E[s^{\sum_{i=1}^{k+1} Z_i}|Z_1 = j] * P(Z_1 = j)) =$$

$$s(\sum_{j=0}^{\infty} \phi_k(s)^j * P(Z_1 = j)) = s(\sum_{j=0}^{\infty} \phi_k(s)^j * P(X = j)) =$$

$$sE[\phi_k(s)^Z] = sG(\phi_k(s))$$

concluiding the proof.

(b)

If

$$T = \lim_{n\to\infty} T_n$$

exists, we also obtain that

$$\lim_{n-1\to\infty} T_n = T$$

as well. Since pgf is an increasing function on the interval $(0, 1]$

$$\lim_{k\to\infty} \phi_k(s) = \lim_{k\to\infty} sG(\phi_{k-1}(s))$$

otherwise limit does not exist. It follows that

$$\phi(s) = sG(\phi(s))$$

(c)

$$\frac{d\phi}{ds} = \frac{d}{ds} sG(\phi(s)) =$$

$$G(\phi(s)) + s \sum_{j\in\mathbb{N}^+} j\phi(s)^{j-1} \frac{d\phi}{ds} P(X = j) =$$

$$\sum_{\in\mathbb{N}_{\not{}}} j\phi(s)^j P(X = j) + s \sum_{j\in\mathbb{N}^+} j\phi(s)^{j-1} \frac{d\phi}{ds} P(X = j) =$$

$$\frac{d\phi}{ds}(1 - s \sum_{j\in\mathbb{N}^+} j\phi(s)^{j-1} P(X = j)) = \sum_{j\in\mathbb{N}_{\not{}}} \phi(s)^j P(X = j)$$

$$\implies \frac{d\phi}{ds} = \frac{\sum_{j\in\mathbb{N}_{\not{}}} \phi(s)^j P(X = j)}{(1 - s\sum_{j\in\mathbb{N}^+} j\phi(s)^{j-1} P(X = j))}$$

It follows that

$$\frac{d\phi}{ds}(1) = \frac{\sum_{j\in\mathbb{N}_{\not{}}} \phi(1)^j P(X = j)}{(1 - 1\sum_{j\in\mathbb{N}^+} j\phi(1)^{j-1} P(X = j))} =$$

since $E[1^r] = 1$ for all integer, consequently

$$\frac{\sum_{j \in \mathbb{N}_{\nvdash}} E[1^T]P(X=j)}{(1 - 1\sum_{j \in \mathbb{N}^+} jE[1^T]P(X=j))} = \frac{\sum_{j \in \mathbb{N}_{\nvdash}} P(X=j)}{(1 - \sum_{j \in \mathbb{N}^+} jP(X=j))} =$$

$$\frac{1}{1-\mu}$$

In the subcritical case

4.26

We define our random variable Wbe the count of matching numbers. It is a hypergeometric random variable with parameters N, K, n, i.e.

$$W \sim \text{ Hypergeometric}(N = 100, K = 3, n = 3)$$

with pmf

$$P(W = w) = \frac{\binom{K}{w}\binom{N-K}{n-w}}{\binom{N}{n}}$$

Now, define a function that maps count of matching numbers to the amount you will be winning

$$h : \{0, 1, 2, 3\} \to \mathbb{R}$$

defined as below:

$$h : \begin{cases} 0 \mapsto & -1 \\ 1 \mapsto & 2 \\ 2 \mapsto & 14 \\ 3 \mapsto & 999 \end{cases}$$

(a) Then we are finding $E_X[h(X)]$

We will have

$$-1 * \frac{\binom{3}{0}\binom{97}{3}}{\binom{100}{3}} + 2\frac{\binom{3}{1}\binom{97}{2}}{\binom{100}{3}} + 14 * \frac{\binom{3}{2}\binom{97}{1}}{\binom{100}{3}} + 999 * \frac{\binom{3}{3}\binom{97}{0}}{\binom{100}{3}} = 0.7076$$

```
v <- c(-1*((choose(97, 3))/(choose(100, 3))), 2*((3*choose(97, 2))/(choose(100, 3))),14*(3*97/(choose(1
sum(v)
```

```
## [1] -0.7076747
```

which is about -70.8 cents.

(b)

In case of parlaying we have the offspring distribution

$$a = (a_0, 0, 0, a_3, 0, 0, ..., 0, a_{15})$$

where every component besides a_0, a_3 and a_15 is zero.

Naturally

$$a_3 = P(W = 1) = \frac{\binom{3}{1}\binom{97}{2}}{\binom{100}{3}}$$

and

$$a_{15} = P(W = 2) = \frac{\binom{3}{2}\binom{97}{1}}{\binom{100}{3}}$$

a_0 consists of two cases : firstly where you got none of the matchiong numbers correctly, and secondly where you hit the jackpot. Thus we have

$$a_0 = P(W = 0) + P(W = 3) = \frac{\binom{97}{3} + 1}{\binom{100}{3}}$$

. For the mean we will have

$$\mu = 0.286141 < 1$$

Thus the process is subcritical

```
a <- c(((choose(97, 3))/(choose(100, 3))), ((3*choose(97, 2))/(choose(100, 3))),(3*97/(choose(100, 3)))
k <- c(0, 3, 15, 0)
mu <- sum(a*k)
mu
```

```
## [1] 0.286141
```

(c)

If T denotes the duration of the process, the probability $P(T = k)$ for $k = 1, ..., 4$ would be

$$P(T = n) = G_n(0) - G_{n-1}(0)$$

Proved from 4.19.

First, let's define some functions in R:

```
G<- function(s){
#mgf for offpsring distribution
  a <- c(((choose(97, 3)+1)/(choose(100, 3))), ((3*choose(97, 2))/(choose(100, 3))),(3*97/(choose(100, 
  v <- c(1, s^3, s^15)
  return(sum(v*a))
}
```

```
G_n <- function(n, s){
#composing G(s)
  if(n==0){
    return(s)
  } else{
    return(G_n(n-1, G(s)))
  }
}
```

```
T_n = function(n){
#implementing  G_n(0) - G_n-1(0)
  return((G_n(n,0)-G_n(n-1, 0)))
  }
#print(G)
#print(G_n)
#print(T_n)
```

```
result <- c(0,0,0,0)
for(i in 1:4){
  result[i] = T_n(i)
}
result
```

```
## [1] 0.911818182 0.065936689 0.016092238 0.004409053
```

(d)

We will first find $p/(1-m)$ since we know p and m. We have

$$p = \frac{1}{\binom{100}{3}}$$

and

$$\mu = 0.286141$$

Thus our asymptotic probability of winning is

```
p <- 1/choose(100, 3)

p/(1-mu)
```

```
## [1] 8.663184e-06
```

```
1/(1-mu)
```

```
## [1] 1.400837
```

5.7

We first construct a block of pseudocode describing Metropolis-Hastings algorithm obtaining a binomial sample from the uniform distribution proposed. We will calculate the acceptance function first. We have

$$\forall k \ \pi_k = \binom{n}{k} p^k (1-p)^{n-k}$$

And we have

$$T_{kl} = P(T_1 = l | T_0 = k) = P(T_1 = l) = \frac{1}{n+1}$$

since we independently choose from the discrete uniform distribution that has state space $\{0, ..., n+1\}$

Since T_{kl} is constant for all possible k and l, our acceptance function would be

$$a(i,j) = \min\{1, \frac{\pi_j T_{ji}}{\pi_i T_{ij}}\} = \min\{1, \frac{\pi_j}{\pi_i}\} = \min\{1, \frac{\binom{n}{j}}{\binom{n}{i}} p^{j-i} (1-p)^{i-j}\}$$

By this we can see

$$a(i,i) = 1 \ \forall i$$

```
MetropolisHastingsBinom(trials, n, p): numeric(N)

  Define samples = numeric(N)
  set the first component of samples as 0
```

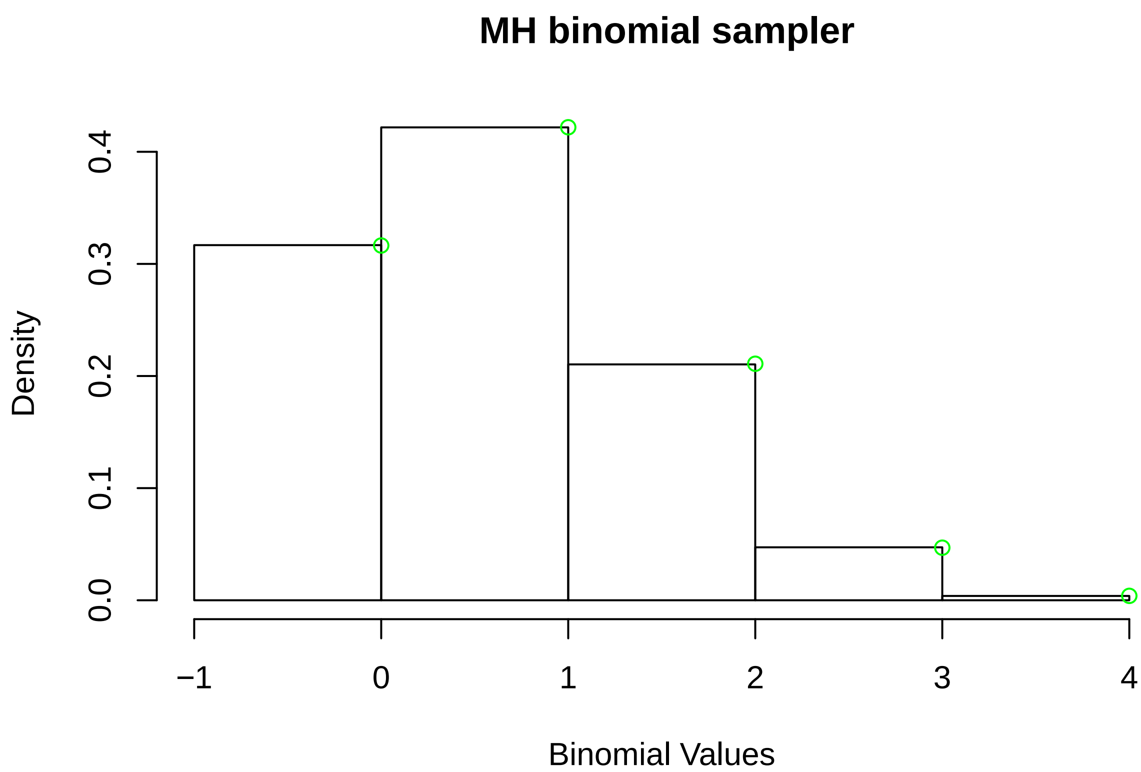
```
  for k = 2, ..., N do
    i = (i-1)th component of vector samples
    j ~ discreteUniform({0,1, ..., n-1, n})

    a(i, j) = min(1, (choose(n, j)/choose(n, k)) * p^(i-j)* (1-p)^(j-i))

    U ~ Uniform(0, 1)

    if(U < a(i, j)) then set  k-th component of samples as j
    else set it to i
  done
return samples
```

Let's implement this into an actual R function.

```r
#5.7
MHbinom <- function(N, n, p){
  simlist <- numeric(N)
  simlist[1] = 0
  for(k in 2:N){
     i = simlist[k-1]
     j = sample(0:n,1)

     acc <- min(1, choose(n, j)/choose(n, i) * p^(j-i)*(1-p)^(i-j))

     if (runif(1)<acc){
       simlist[k] = j
     } else{
       simlist[k] = i
     }

  }

  return(simlist)

}
```

We make a simulation from question #5.8:

```r
simbinom <- MHbinom(1000000, 4, .25)
hist(simbinom,breaks=seq(-1,4,by=1), xlab="Binomial Values",main=" MH binomial sampler",prob=T)
xref <- seq(0,4,by = 1)
yref <- dbinom(xref,4,.25)
points(xref,yref,col="green")
```



which shows an amazing fit.

We check mean and variance as well.

```
muCompare = c(1, mean(simbinom))
sigmaCompare = c(4*(0.25)*(0.75), var(simbinom))
muCompare
```

```
## [1] 1.000000 0.999645
```

```
sigmaCompare
```

```
## [1] 0.7500000 0.7506756
```

5.20

We need to obtain a Poisson sample with $\lambda = 3$ from a geometric distribution with $p = \frac{1}{3}$ proposed.

We have

$$\forall k \ \pi_k = e^{-\lambda} \frac{\lambda^k}{k!}$$

and

$$\forall i \ j \ T_{ij} = P(X_1 = j | X_0 = i) = P(X_1 = j) = (1-p)^{j-1} p$$

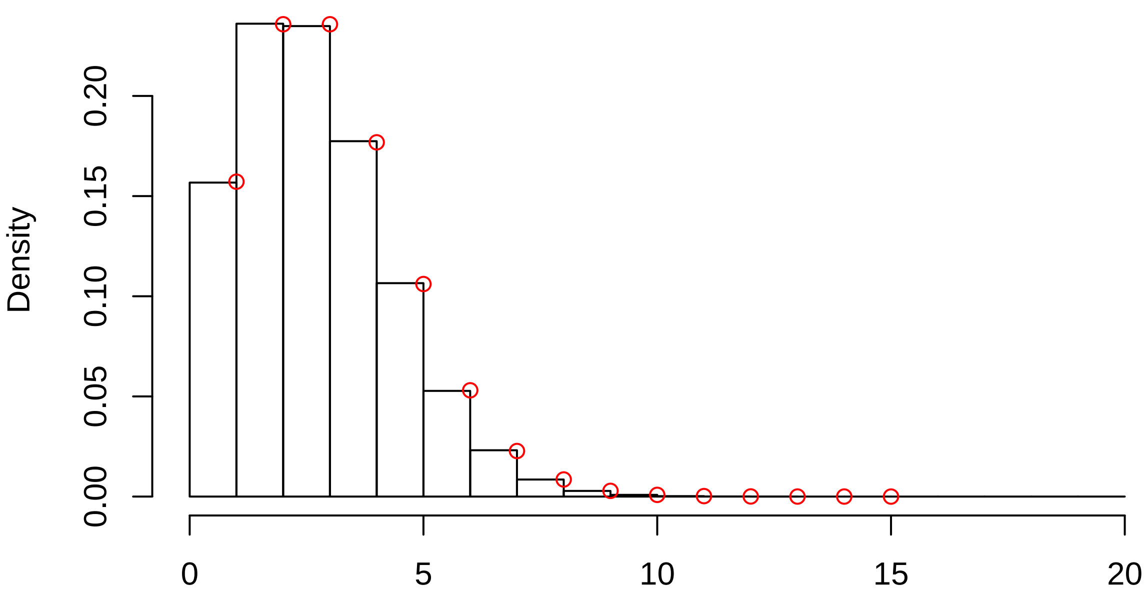
So our acceptance function would be

$$a(i,j) = \frac{\pi_j T_{ji}}{\pi_i T_{ij}} = \frac{e^{-\lambda} \frac{\lambda^j}{j!}}{e^{-\lambda} \frac{\lambda^i}{i!}} \frac{T_{ji}}{T_{ij}} = \frac{i!}{j!} (\lambda^{j-i}) \frac{(1-p)^{i-1} p}{(1-p)^{j-1} p} = \frac{i!}{j!} (\lambda^{i-j}) (1-p)^{i-j}$$

If we implement this into an R function:

```r
#5.20
MHPoisson <- function(N, lambda, p){
  # We sample from Poisson random variable with mean lambda
  # We propose geometric distribution with proportion p
  # We use the "time-till-the-first-success" definition of geometric distribution
  # P(X=k; p) = (1-p)^(k-1)p

  simlist <- numeric(N)
  simlist[1] <- 1 # we assume a positive integer state space

  for(k in 2:N){
    i = simlist[k-1]
    j = rgeom(1, p)+1

    acc <- lambda^(j-i)*(factorial(i)/factorial(j))*(1-p)^(i-j)

    if(runif(1) < acc){
      simlist[k] = j

    } else {
      simlist[k] = i
    }

  }
  return(simlist)
}
```

```r
simlistPois <- MHPoisson(1000000, 3, 1/3)

xref <- seq(min(simlistPois), max(simlistPois), by=1)
yref <- dpois(xref,3)/(1-exp(-3))

hist(simlistPois, breaks=seq(0,20,by=1) ,xlab="",main="",freq=F)
points(xref,yref,col="red")
```



It shows a great fit as well.

We also compare mean and variance

```r
#compare mean and variance to 3, the "supposed" value
c(3, mean(simlistPois), var(simlistPois))
```

```
## [1] 3.000000 3.159582 2.662642
```

We can have a decrease in variance since we conditioned our Poisson sample to be nonzero. Besides that, it matches.