

# **MVE550 - Second Assignment**

Adam Andersson, Albin Ekström  
& Joakim Osterman

December 2, 2022



# 1 Discrete-Time Markov Chain with $\{1,2,3\}$

a) We count the transitions between 1,2, and 3 in a transition matrix.

$$T = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix}$$

Adding a pseudo count to the transition matrix we can get the posterior distributions for  $P_1$ ,  $P_2$ ,  $P_3$  and thus  $P$ .

$$\begin{aligned}\pi(P_1|y) &= \text{Diriclet}(1, 3, 2) \\ \pi(P_2|y) &= \text{Diriclet}(2, 1, 3) \\ \pi(P_3|y) &= \text{Diriclet}(2, 3, 1)\end{aligned}$$

$$\therefore \pi(P|y) = \begin{bmatrix} \pi(P_1|y) \\ \pi(P_2|y) \\ \pi(P_3|y) \end{bmatrix}$$

We use Equation 1.1 to calculate the expected posterior for  $P_1$ ,  $P_2$ , and  $P_3$  respectively. Where  $c_i$  is the transition counts observed from our data, as a vector. And  $k$  is the added pseudo-counts, i.e. 3 in our case.

$$E(P_i|x_o, \dots, x_n) = \frac{(1, 1, \dots, 1) + c_i}{k + c_{i1} + \dots + c_{ik}} \quad (1.1)$$

Hence,  $E(P_1|y)$ ,  $E(P_2|y)$ , and  $E(P_3|y)$  are:

$$\begin{aligned}E(P_1|y) &= (0.1666667, 0.5000000, 0.3333333) \\ E(P_2|y) &= (0.3333333, 0.1666667, 0.5000000) \\ E(P_3|y) &= (0.3333333, 0.5000000, 0.1666667)\end{aligned}$$

Thus,

$$\therefore E(P|y) = \begin{bmatrix} 0.1666667 & 0.5000000 & 0.3333333 \\ 0.3333333 & 0.1666667 & 0.5000000 \\ 0.3333333 & 0.5000000 & 0.1666667 \end{bmatrix}$$

**b)** A histogram of a simulation of the chain for a length of 400 samples is shown in Figure 1.1.

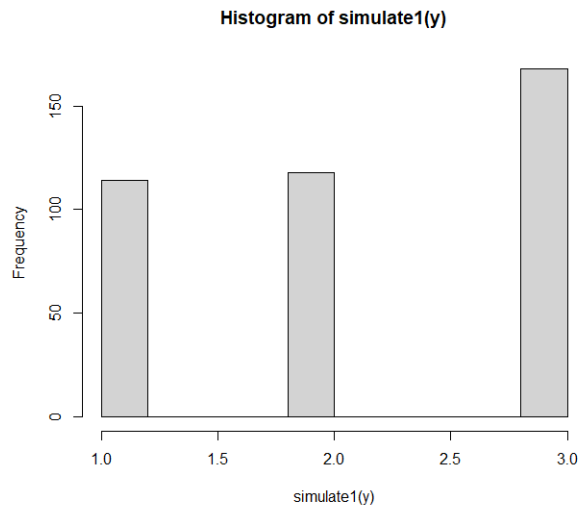


Figure 1.1: Histogram plot for the continuation of the chain above into a chain of length 400 samples.

**c)** Using a similar function as in **b)**, we sample 1000 values of the  $E(P_{12}|\text{actual and simulated data})$  into a histogram shown in Figure 1.2.

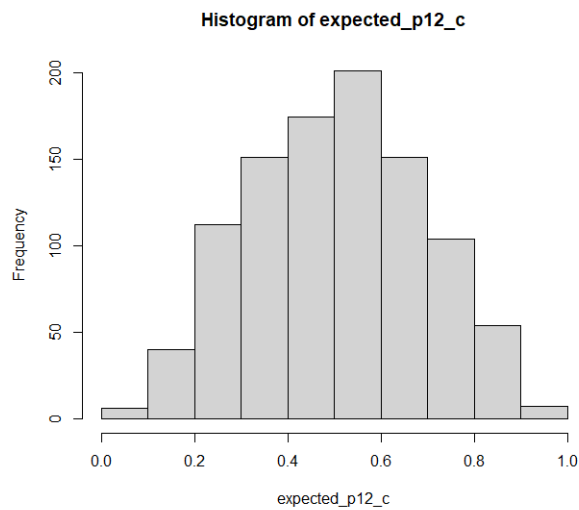


Figure 1.2: Histogram of  $E(P_{12}|\text{actual and simulated data})$

**d)** In Figure 1.2 we update the count matrix with the new simulated value, but in this task, the predicted values should only depend on the first 10 observed values. The histogram is shown in 1.3

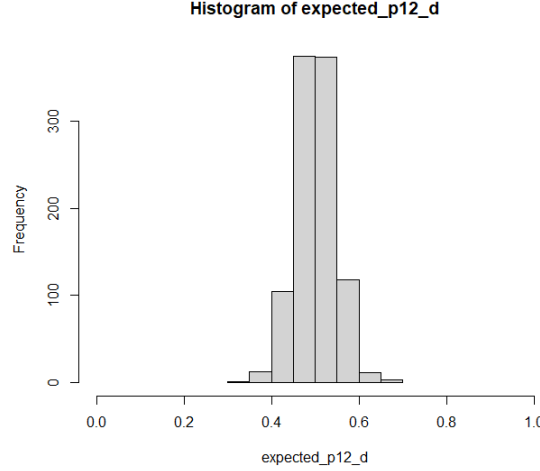


Figure 1.3: Histogram for the prediction only depending on the first 10 observed values

The differences occurring between Figure (histogram) 1.2 and 1.3 are because they are plotted from different chains of data. For the histogram shown in Figure 1.2 every new iteration plots the result based on the actual data and the recently simulated data as well. Therefore it is reasonable to get a much broader spread in our final histogram when compared to Figure 1.3. Since we in Figure 1.3 only use our original counted data to simulate the new state chain, hence the narrow distribution.

**e)** Figure 1.4 shows a histogram of the new states in the chain where each sampled  $E(P_{12}|y)$  is based on a Dirichlet distributed sampled posterior.

The histograms shown in Figure 1.3 and 1.4 are very similar even though the sampling is different. Since we are sampling to form the expected distribution of  $P_{12}$  we can assume with the law of large numbers that we get a similar distribution as in Figure 1.3. If the number of samples increases, the histogram will give an even more similar result to Figure 1.3

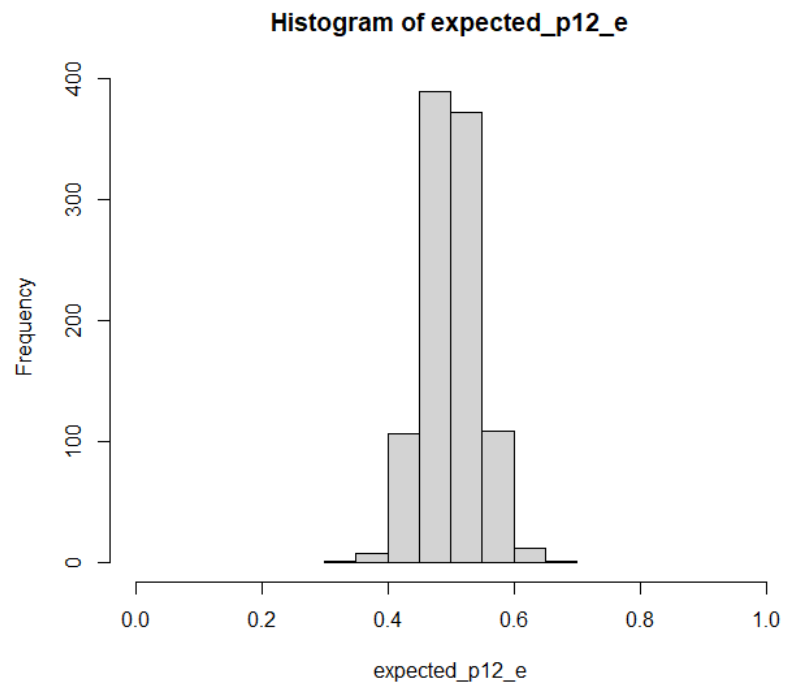


Figure 1.4: Histogram of the probability  $P$  sampled 1000 times from the posterior.

## 2 A Branching Process

a) To determine the posterior distribution of  $\lambda$  we first create a vector based on the observed offspring distribution, see figure 2.1. Resulting in a vector  $y = [1, 2, 3, 2, 0, 0, 1, 4, 2]$ .

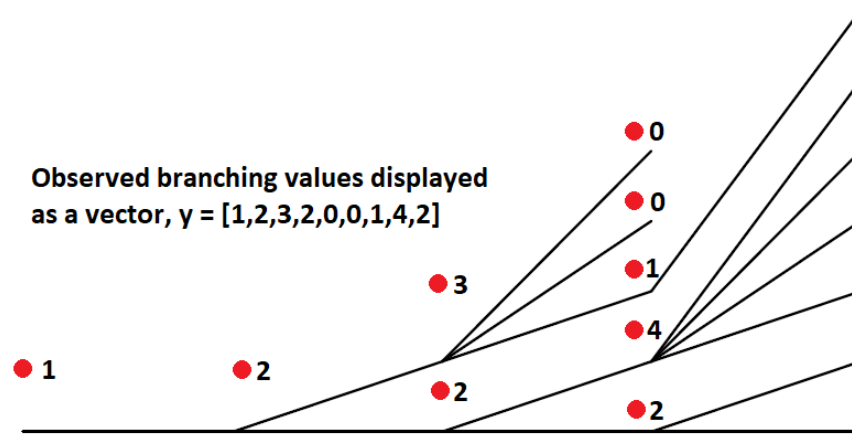


Figure 2.1: Observed branching process, displayed as a vector

To display the posterior distribution of  $\lambda$  we use a gamma distribution due to the Poisson gamma conjugacy where  $\alpha$  is the sum of the observed children and  $\beta$  is the count of the observed values  $y$ . See Figure 2.2 for the resulting posterior distribution of  $\lambda$ .

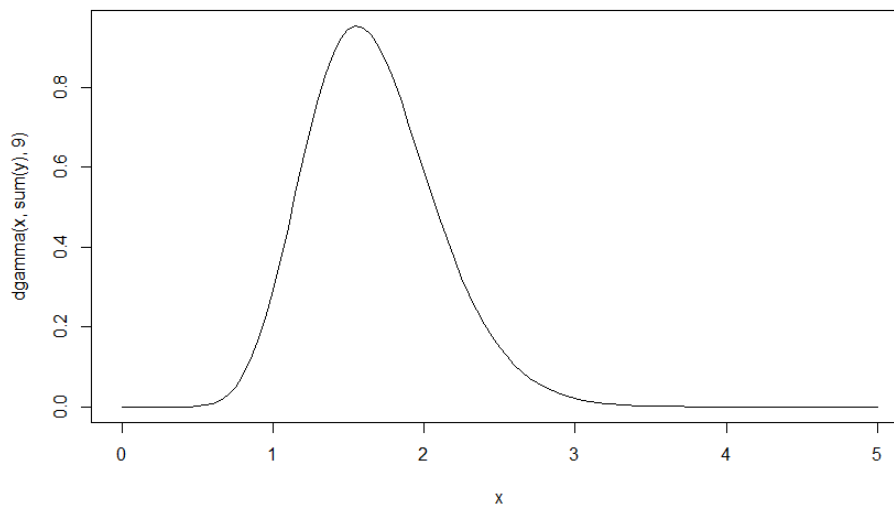


Figure 2.2: Posterior distribution of  $\lambda$

**b)** Considering the continuation of the generations in the branching process we create a function in R to determine the rate of extinction, which takes a value for  $\lambda$  and outputs the probability of becoming extinct. To calculate the probability of whether a branching process becomes extinct or not we use a probability-generating function (pgf)  $G(s)$  for extinction (see Equation 2.1).

$$G(s) = \sum_{k=0}^{\infty} s^k \alpha_k \quad (2.1)$$

To calculate the probability generating function it needs the offspring distribution  $\alpha_k$  which is a Poisson distribution. Using the `optimize` function in R on the absolute value of the function  $G(s) - s = 0$ : we find the smallest root for  $s$ . Then we can simply plot our extinction probability as a function where  $x = \lambda$ , see figure 2.3.

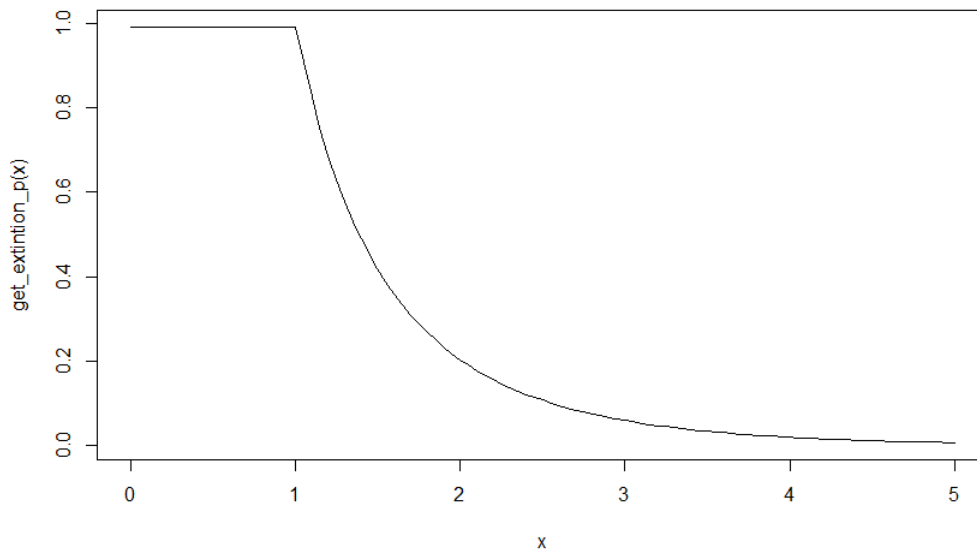


Figure 2.3: The rate of extinction as a function of  $x = \lambda$



**c)** Equation 2.2 computes the probability of extinction taking uncertainty in  $\lambda$  into account. In Equation 2.2  $\pi(\lambda|y)$  is the gamma distribution.

$$P(ext.|\pi(\lambda|y)) * \pi(\lambda|y) \quad (2.2)$$

Then, simply integrate this product to get the extinction probability as in Equation 2.3.

$$\int_0^\infty P(ext.|\pi(\lambda|y)) * \pi(\lambda|y) \quad (2.3)$$

$\therefore$  Thus providing the probability  $P_f = 0.3997901$

**d)** To simulate from the posterior found in **a)** sample various values of  $\lambda$  from the posterior distribution. Also plugging each of these values into the function determined in **b)**. Resulting in a list of probabilities of going extinct for various values of  $\lambda$ . Then, compute the mean of these values to get an approximated value of going extinct. Note that this probability varies a small amount since we used "only" a thousand samples of  $\lambda$ . Hence, if the number of samples is increased (according to the law of large numbers) the sampled value will get more accurate and eventually go to the expected. Thus, the result for the simulated probability of extinction is,

$$\therefore P_{sim} \approx 0.3940227$$

**e)** Using `optimize` the lambda is found with the highest likelihood from the function in Equation 2.4.

$$f(\lambda) = \prod_{i=0}^k P(y_i|\lambda) \quad (2.4)$$

This function gives that the maximum likelihood for  $\lambda \approx 1.67$ . Thus, providing the probability of extinction as,

$$\therefore P_{maxLikelihood} = 0.3242287$$

Listing 2.1: Assignment 1.r

```

library(LearnBayes)

## a) Compute the expectation of the posterior for P

# 10 experiments, 3 ones, 4 twos, 3 threes
# Pi - Dirichlet(1,1,1)

# Functions

get_tmatrix <- function(y)
{
  # Get counts of transitions in transition matrix
  m = matrix(0, nrow = 3, ncol = 3, byrow = T)
  for (i in 1:length(y)-1)
  {
    v1=y[i]
    v2=y[i+1]
    m[v1,v2]=m[v1,v2]+1
  }
  return (m)
}

get_expected_m <- function(y)
{
  m = get_tmatrix(y)
  P1 = m[1,]
  P2 = m[2,]
  P3 = m[3,]

  # E(Pi | data) Pg 28. in comp.
  EP1 = (c(1, 1, 1) + P1) / (length(P1) + sum(P1))
  EP2 = (c(1, 1, 1) + P2) / (length(P2) + sum(P2))
  EP3 = (c(1, 1, 1) + P3) / (length(P3) + sum(P3))

  # Expectation of the posterior for P
  return(matrix(c(EP1, EP2, EP3), 3, 3, byrow = TRUE))
}

simulate1 <- function(y) {
  m=get_expected_m(y)

  for (i in (length(y)+1):400) {
    # Gets the last value of chain
    t = tail(y,1)

    # Sample a value with prob. from value n-1
    y[i] = sample(c(1,2,3),1,prob = m[t,])
  }
}

```

```

    # New expected expected posterior
    m = get_expected_m(y)
  }
  return (y)
}

simulate2 <- function(y){
  # Return  $E(P_{12} \mid \text{actual and simulated data})$ 
  m=get_expected_m(y)

  for (i in (length(y)+1):400) {
    t = tail(y,1)
    y[i]=sample(c(1,2,3),1,prob = m[t,])
    m=get_expected_m(y)
  }
  return (m[1,2])
}

simulate3 <- function(y){
  m1 = get_expected_m(y)

  # Store the expected posterior
  m = m1
  for (i in (length(y)+1):400)
  {
    t = tail(y,1)
    y[i]=sample(c(1,2,3),1,prob = m1[t,])
    m=get_expected_m(y)
  }
  return (m[1,2])
}

simulate4 <- function(y){
  m1 = get_expected_m(y)
  m = m1
  for (i in (length(y)+1):400)
  {
    t = tail(y,1) # last seen value
    Psamp = rdirichlet(1,m1[t,])
    y[i]=sample(c(1,2,3),1,prob = Psamp)
    m=get_expected_m(y)
  }
  return (m[1,2])
}

# 1.a
# observed values
y = c(1,2,3,2,3,1,2,1,3,2)

```

```

P = get_expected_m(y)

# How to get  $E(P|y)$ 

# 1.b

hist(simulate1(y))

# 1.c
expected_p12_c = replicate(1000, simulate2(y))
hist(expected_p12_c, xlim = c(0,1))

# 1.d
expected_p12_d = replicate(1000, simulate3(y))
hist(expected_p12_d, xlim = c(0,1))

# 1.e
expected_p12_e = replicate(1000, simulate4(y))
hist(expected_p12_e, xlim = c(0,1))

#m_count = matrix(c(length(which(y == 1)),
#                    length(which(y == 2)),
#                    length(which(y == 3))),
#                  1,3, byrow = T)

```

Listing 2.2: Assignment 1.r

```
# Data
y=c(1,2,3,2,0,0,1,4,2)

# Functions
#a <- function(lambda) {dpois(0:50,lambda)} # Child dist.

G <- Vectorize(function(s,lambda)
{
  a=dpois(0:50,lambda)
  sum(a[1:length(a)]*s^(0:(length(a)-1)))
})

get_extinction_p = Vectorize(function(lambda) {
  optimize(Vectorize(function(x) {abs(G(x,lambda)-x)}),
    interval =c(0,0.99))$minimum
})

posterior <-function(x) {dgamma(x,sum(y),9)}

#2.a) Plotting the posterior of lambda
curve(dgamma(x,sum(y),9),to=5)

#2.b) Plotting the rate of extinction based on different
# values of lambda = x in the curve
curve(get_extinction_p(x), to = 5)

#2.c)

g_ext <- function(p) posterior(p)*get_extinction_p(p)

integrate(Vectorize(g_ext), 0, Inf)$value

#2.d)
p <- seq(0, 10, length.out=500)
ext_list = c()

# Sample extinctions
for (i in 1:1000){
  lambda_sample <- sample(p, 1, prob = posterior(p))
  ext_list[i]=get_extinction_p(lambda_sample)
}

# avg. extinction / y
```

```
mean(ext_list)

#2.e
get_lh <- function(x)
{
  likelyhood = 1
  for (i in 1:length(y))
  {
    likelyhood = likelyhood*dpois(y[i],x)
  }
  return (likelyhood)
}

max_lambda = optimize(Vectorize(get_lh),c(0,50),maximum = T)
max_lambda$maximum
get_extinction_p(max_lambda$maximum)
```