Worksheet: Chain Binomial models

Nathan Green, Imperial College London

14/01/2020

1 Introduction

In this practical we will explore the use of the chain Binomial models for infectious diseases, introduced in the lectures.

2 Inference

First we will infer the paramters of the model for the measles example given in the lecture.

For chain observed sample sizes a, b, c, d and probability of infection and avoidance p and $q = 1 - p$, respectively. Define a log-likelihood function corresponding to the data as follows.

$$l = a\ln(q^2) + b\ln(2pq^2) + c\ln(2p^2q) + d\ln(p^2)$$

This can be coded in R as

```
loglik_measles <- function(a,b,c,d) {

  function(p){
    q <- 1 - p
    a*log(q^2) + b*log(2*p*q^2) +c*log(2*p^2*q) + d*log(p^2)
  }
}
```

This is an example of a *closure* which returns a function with parameters `a,b,c,d` defined. We can check this with the data from the lecture.

```
loglik_dat <- loglik_measles(a = 34,
                             b = 25,
                             c = 36,
                             d = 239)

loglik_dat
```

```
## function(p){
##     q <- 1 - p
##     a*log(q^2) + b*log(2*p*q^2) +c*log(2*p^2*q) + d*log(p^2)
##   }
## <environment: 0x0000000012f322a8>
```

To find the MLE we identify the probability corresponding to the largest likelihood in the sequence of discrete probabilities we just created.

```
p <- seq(0, 1, 0.005)

p_hat <- p[loglik_dat(p) >= max(loglik_dat(p))]
p_hat # 0.79
```

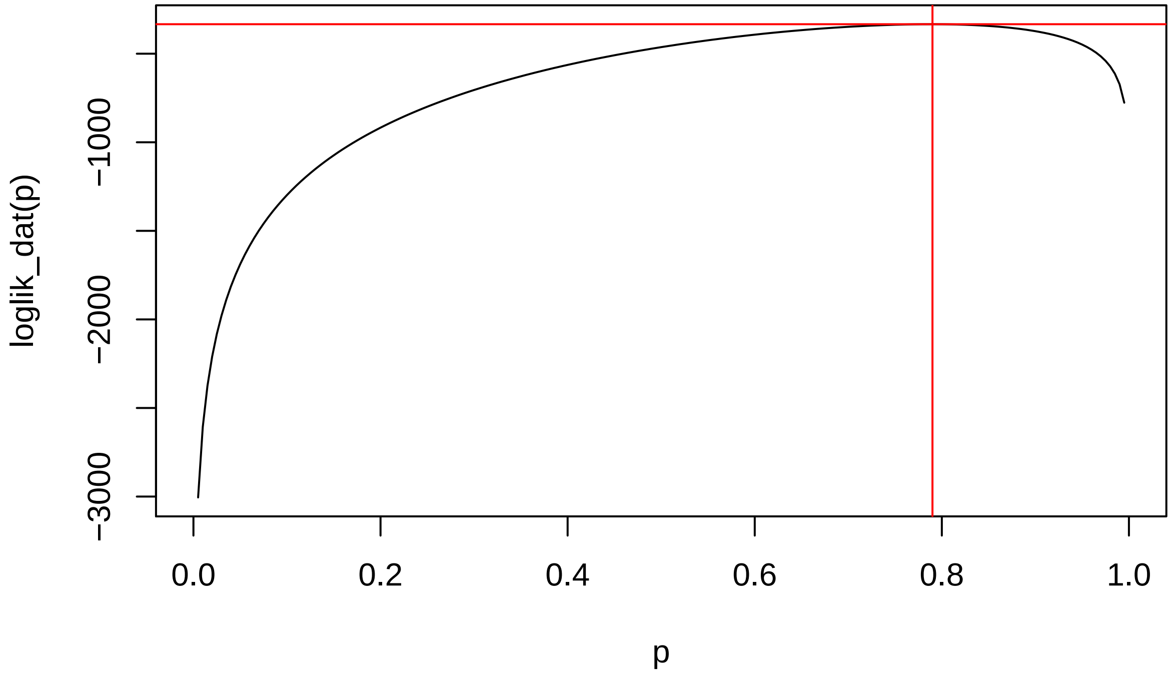
```
## [1] 0.79
```

```
q_hat <- 1 - p_hat
q_hat # 0.21
```

```
## [1] 0.21
```

In the same way as for the previous likelihood practical, let's plot the log-likelihood curve. We indicate the MLE $\hat{p}$ with a red line.

```
plot(p, loglik_dat(p), type = "l")
abline(v = p_hat, col = "red")
abline(h = max(loglik_dat(p)), col = "red")
```



We are now in a position to calculate the expected number of households for each types of chain from the measles data. The total number of households is $n = 334$.

```r
n <- 334

# {1}
n*q_hat^2
```

```
## [1] 14.7294
```

```r
# {1~2}
n*2*p_hat*q_hat^2
```

```
## [1] 23.27245
```

```r
# {1~3}
n*2*p_hat^2*q_hat
```

```
## [1] 87.54875
```

```r
# {12}
n*p_hat^2
```

```
## [1] 208.4494
```

2.1 Your turn

Can you extend the previous model to households of size 4. For data $n = 500$, and $a = 11, b = 12, c = 14, d = 27, e = 49, f = 63, g = 152, h = 172$.

- Define the log-likelihood function.
- plot the log-likelihood curve.
- What is the MLE and the expected number of households for each type of chain.

ANSWERS:

```r
loglik_measles <- function(a,b,c,d,e,f,g,h) {

  function(p){
    q <- 1 - p
    a*log(q^3) + b*log(3*p*q^4) + c*log(6*p^2*q^4) +
      d*log(6*p^3*q^3) + e*log(3*p^3*q^2) + f*log(3*p^2*q^2) +
      g*log(3*p^3*q) + h*log(p^3)
  }
}
```

```r
loglik_dat <- loglik_measles(a = 11,
                             b = 12,
                             c = 14,
                             d = 27,
                             e = 49,
                             f = 63,
                             g = 152,
                             h = 172)
```

```
p_hat <- p[loglik_dat(p) >= max(loglik_dat(p))]
p_hat
```

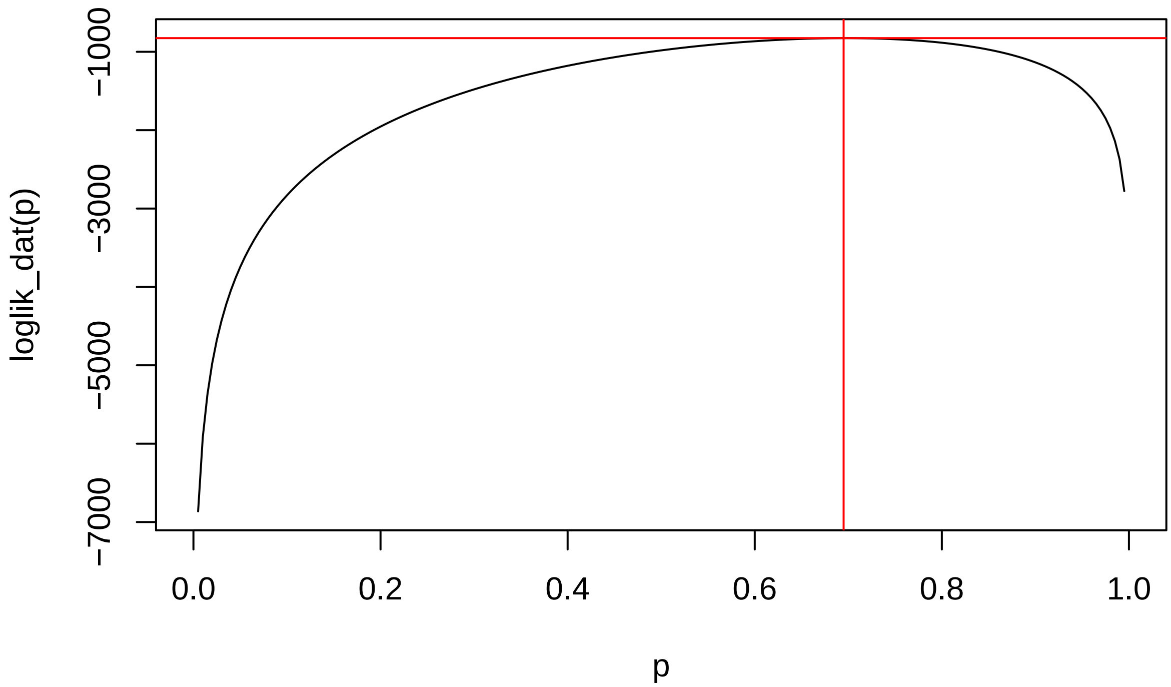
```
## [1] 0.695
```

```
q_hat <- 1 - p_hat
q_hat
```

```
## [1] 0.305
```

The true value is $p = 0.7$,

```
plot(p, loglik_dat(p), type = "l")
abline(v = p_hat, col = "red")
abline(h = max(loglik_dat(p)), col = "red")
```



```
n <- 500

# {1}
n*q_hat^3
```

```
## [1] 14.18631
```

```
# {1^2}
n*3*p_hat*q_hat^4
```

```
## [1] 9.021431
```

```
# {1^3}
n*6*p_hat^2*q_hat^3
```

```
## [1] 41.11406
```

```
# {12}
n*3*p_hat^2*q_hat^2
```

```
## [1] 67.4001
```

3 Stochastic simulation

We can generate our own chain Binomial data using forward simulation. We can do this in term of the final sizes i.e. chain frequencies via a multinomial distirbution or explicitly modelling the chain trajectories with the SIR chain Binomial model.

3.1 Multinomial distibution

A simple way of doing this is to sample from a multinomial distibutions with probabilites coresponding to each of the types of chain. For this formulation its easy to see the connection with the likelihood approach above.

For example, for households of size 3:

```
p <- 0.8
q <- 1 - p
rmultinom(n = 1,
          size = 334,
          prob = c(q^2, p*q^2, 2*p^2*q, p^2))
```

```
##      [,1]
## [1,]   11
## [2,]   13
## [3,]   79
## [4,]  231
```

Households of size 4:

```
p <- 0.7
q <- 1 - p
rmultinom(n = 1,
          size = 500,
          prob = c(q^3, 3*p*q^4, 6*p^2*q^4, 6*p^3*q^3, 3*p^3*q^2, 3*p^2*q^2, 3*p^3*q, p^3))
```

```
##      [,1]
## [1,]   16
## [2,]    6
## [3,]   15
## [4,]   27
## [5,]   64
## [6,]   67
## [7,]  154
## [8,]  151
```

3.2 chain Binomial model

We can define the model using the transition probabilities to update the number of susceptibles, infectives and removed at each time step.

$$S_{t+1} = S_t - I_{t+1}, \qquad R_{t+1} = R_t + I_t$$

Let us create a function to simulate this process. The input parameters are defined as follows:

- `q` is athe pairwise probability of *avoiding* potentially infectious contact
- `R0` is the basic reproduction number
- `MAXTIME` is the maximum number of time steps the simulation will run for
- `I0` is the initial number of infectives
- `S0` is the initial number of susceptibles

```r
sir_cb <- function(q = NA,
                   R0 = NA,
                   N,
                   MAXTIME = 100,
                   I0 = 1,
                   S0 = N - I0){
  if (is.na(q))
    q <- 1 - R0/N

  # initial conditions
  I <- I0
  S <- S0

  for(t in seq_len(MAXTIME)){

    I <- c(I, rbinom(n = 1,
                     size = S[t],
                     prob = 1 - q^I[t]))

    S <- c(S, S[t] - I[t + 1])

    if(I[t + 1] == 0) break

  }

  return(
    data.frame(t = 0:t,
                I = I,
                S = S))
}
```

At each time step a susceptible becomes infected with probability `1 - q^I[t]`. When there are no more infectives then the function stop, indicated by the `break` function.

Let us check that this function does what we expect. For a single household we get the following.

```
sir_cb(q = 0.5, N = 3)
```

```
##   t I S
## 1 0 1 2
## 2 1 0 2
```

We can now replicate the output from the measles example given in the lecture. First we create a list object to put the results in to. Then we simulate $n = 334$ households of size 3 by looping using the `for` command. While we're at it we also extract the number of infectives into a separate variable called `inf`.

```
res <- list()
inf <- NULL

for (i in 1:334){

  res[[i]] <- sir_cb(q = 0.21, N = 3)
  inf[i] <- paste(res[[i]]$I, collapse = ",")
}
```

```
table(inf)
```

```
## inf
##     1,0   1,1,0 1,1,1,0   1,2,0
##      17      27      74     216
```

3.3 Your turn

- Change the `q` and `N` parameters to explore the model behaviour.
- What would you expect to see for larger q or larger N?

The `sir_cb()` function above used the Reed-Frost formulation of the model. Modify it to use the Greenwood version.

- Rerun the analysis for households of size 4 and compare the results for the two different types of model. What do you see?