Math 475: Final Exam

Hannah Zmuda

12/26/2020

Problem 1: The Peppered Moths

The goal of this problem is to find and create and EM Algorithm for looking at Peppered moths species in an area/from a sample population. ### Step 1 (E): Find the Q function

$$Q(\theta|\theta^t) = E[logL(\theta|Y)|x, \theta^t]$$

Following the notation, we first have our observed data x. $x = (n_c, n_I, n_T)$ where n is the number of moths based on one of three phenotype (C, I, T). The complete data $y = (n_{CC}, n_{CI}, n_{CT}, n_{II}, n_{IT}, n_{TT})$ where y represents the number of moths based on allele frequency (genotype). For this problem, **we wish to estimate the allele probabilities**: $ p = (p_C, p_I, p_T)$. Because the allele T is recessive to I and I is recessive to C, p only needs to depend on allele counts for C and I. This means $p_T = 1 - p_C - p_I$ and $p = (p_C, p_I, 1)$. From there we can compute the complete log likelihood:

$$logf_Y(y|p) = n_{CC}log(p_C^2) + n_{CI}log(2p_Cp_I) + n_{CT}log(2p_Cp_T) + n_{II}log(p_I^2) + n_{IT}log(2p_Ip_T) + n_{TT}log(p_T^2)$$

$$+log\binom{n}{n_{CC} \quad n_{CI} \quad n_{CT} \quad n_{II} \quad n_{IT} \quad n_{TT}}$$

The only complete data variable entirely observed is n_{TT} because it is a recessive allele. Therefore the complete data becomes $Y = (N_{CC}, N_{CI}, N_{CT}, N_{II}, N_{IT}, n_{TT}$. Using this, we can calculate the expectation for each variable in the complete data set Y:

$$E[N_{CC}|n_C, n_I, n_T, p^{(t)}] = n_{CC}^{(t)} = \frac{n_C(p_C^{(t)})^2}{(p_C^{(t)})^2 + 2p_C^{(t)}p_I^{(t)} + 2p_C^{(t)}p_T^{(t)}}$$

$$E[N_{CC}|n_C, n_I, n_T, p^{(t)}] = n_{CI}^{(t)} = \frac{2n_Cp_C^{(t)}p_I^{(t)}}{(p_C^{(t)})^2 + 2p_C^{(t)}p_I^{(t)} + 2p_C^{(t)}p_T^{(t)}}$$

$$E[N_{CC}|n_C, n_I, n_T, p^{(t)}] = n_{CT}^{(t)} = \frac{2n_Cp_C^{(t)}p_T^{(t)}}{(p_C^{(t)})^2 + 2p_C^{(t)}p_I^{(t)} + 2p_C^{(t)}p_T^{(t)}}$$

$$E[N_{CC}|n_C, n_I, n_T, p^{(t)}] = n_{II}^{(t)} = \frac{n_I(p_I^{(t)})^2}{(p_I^{(t)})^2 + 2p_I^{(t)}p_T^{(t)}}$$

$$E[N_{CC}|n_C, n_I, n_T, p^{(t)}] = n_{IT}^{(t)} = \frac{2n_Ip_I^{(t)}p_T^{(t)}}{(p_I^{(t)})^2 + 2p_I^{(t)}p_T^{(t)}}$$

We then get a Q function like the following:

$$Q(p|p^{(t)}) = n_{CC}log(p_C^2)+n_{CI}log(2p_Cp_I)+n_{CT}log(2p_Cp_T)+n_{II}log(p_I^2)+n_{IT}log(2p_Ip_T)+n_{TT}log(p_T^2)+k(n_C, n_I, n_T, p^{(t)}$$

where k is a conditional expectation. This will become unimportant in the M step because it does not depend on p.

Step 2 (M): Maximize the Q function and set next variable to equal maximizer.

To Maximize the Q function, we will take the derivative of the Q function. Because the Q function is multinomial, we will need to take two separate derivatives in regard to p_C and p_I.

$$\frac{dQ(p|p^{(t)})}{dp_C} = \frac{2n_{CC}^{(t)} + n_{CI}^{(t)} + n_{CT}^{(t)}}{p_C} - \frac{2n_{TT}^{(t)} + n_{CT}^{(t)} + n_{IT}^{(t)}}{1 - p_C - p_I}$$

$$p_C^{(t)} = \frac{2n_{CC}^{(t)} + n_{CI}^{(t)} + n_{CT}^{(t)}}{2(n_C + n_I + n_T)}$$

$$\frac{dQ(p|p^{(t)})}{dp_I} = \frac{2n_{II}^{(t)} + n_{IT}^{(t)} + n_{CI}^{(t)}}{p_I} - \frac{2n_{TT}^{(t)} + n_{CT}^{(t)} + n_{IT}^{(t)}}{1 - p_C - p_I}$$

$$p_I^{(t)} = \frac{2n_{II}^{(t)} + n_{IT}^{(t)} + n_{CI}^{(t)}}{2(n_C + n_I + n_T)}$$

$$p_T^{(t)} = \frac{2n_{TT}^{(t)} + n_{CT}^{(t)} + n_{IT}^{(t)}}{2(n_C + n_I + n_T)}$$

Step 3: Return to step 1 (E step) unless stopping criterion has been

Implementation in R

```
set.seed(475)

#initialize variables and initial values
x <- c(85, 196, 341) #observed data x = (nC, nI, nT)
n <- rep(0,6) #complete data (y)
#1:nCC,2:nCI,3:nCT,4:nII,5:nIT,6:nTT
n.old <- rep(0,6,3)
p <- rep(1/3,3) #starting probability values for each allele frequency 1:pC, 2:pI, 3:pT
p.old <- p
I <- 8 #Number of EM algorithm iterations

#E step: Updated n(t) value
moth.e <- function(x,p){
  n.cc <- (x[1]*p[1]*p[1])/(p[1]*p[1] + 2*p[1]*p[2] + 2*p[1]*p[3])#1
  n.ci <- (2*x[1]*p[1]*p[2])/(p[1]*p[1] + 2*p[1]*p[2] + 2*p[1]*p[3])#2
  n.ct <- (2*x[1]*p[1]*p[3])/(p[1]*p[1] + 2*p[1]*p[2] + 2*p[1]*p[3])#3
  n.ii <- (x[2]*p[2]*p[2])/(p[2]*p[2] + 2*p[2]*p[3])#4
  n.it <- (2*x[2]*p[2]*p[3])/(p[2]*p[2] + 2*p[2]*p[3])#5
  n.tt <- x[3]#6
  return(c(n.cc,n.ci,n.ct,n.ii,n.it,n.tt))
}
#M step: Updated p(t+1) value
moth.m <- function(x,n){
  p.c <- (2*n[1] + n[2] + n[3])/(2*sum(x))
  p.i <- (2*n[4] + n[2] + n[5])/(2*sum(x))
  p.t <- (2*n[6] + n[3] + n[5])/(2*sum(x))
  return(c(p.c,p.i,p.t))
}
#Main Loop
```

```r
for(i in 1:I)
{
  n.old <- n
  p.old <- p
  n <- moth.e(x,p)
  p <- moth.m(x,n)
}
p
```

```
## [1] 0.07083691 0.18873679 0.74042630
```

Problem 2

```r
set.seed(475)
#initialize values
x <- c(85, 196, 341) #observed data x = (nC, nI, nT)
n <- rep(0,6) #complete data (y)
#1:nCC,2:nCI,3:nCT,4:nII,5:nIT,6:nTT
p <- rep(1/3,3) #starting probability values for each allele frequency 1:pC, 2:pI, 3:pT
it <- 8 #Number of

#part a: MH Algorithm to est alpha, beta, mu, lambda (sigma)
m.h <- function(x,p,n){

}
```

Problem 3

Problem 4

Problem 5