Assignment-6

MLE Estimators: SriKaavya(st938), Rithvik (ra830), Aravind (ad1452), Sowmya(sa1607)

10-21-2020

Function for Maximum likely hood estimation.

Method to estimate parameters for different distributions is below. The input parameters are:- "distribution_type" : Type of distribution. To be given as a string input "data_input" : Data generated using the specific distribution to be given as input

Mean and variance and given data is calculated and using maximum likelihood estimation, calculations will be done to estimate specific parameter values. These will be printed after calculations.

```r
library(purrr)
library(MASS)
 Parameter_finder <- function( distribution_type , data_input ) {
  # This function is used to calculate parameters for
  #different distributions of data using mean and variance
  mean_data = mean(data_input)
  var_data = var(data_input)

  if ( distribution_type == "Poisson") {
    print(paste0("Poisson distribution parameters - Lambda value is ", mean_data))
  }

  else if (  distribution_type == "Bernoulli") {
    print(paste0("Bernoulli distribution parameters - p value is", mean_data))
  }
  else if (  distribution_type == "Exponential") {
    print(paste0("Exponential distribution parameters - rate value is ", 1/mean_data))
  }
   else if (distribution_type == "Geometric") {
    p = 1/mean_data
    print(paste0("Geometric distribution parameters - p value is ", p))
   }
  else if (  distribution_type == "Normal") {
    print(paste0("Normal distribution parameters - Mean value is ", mean_data))
    print(paste0("Standard deviation is ", sqrt(var_data)))
  }
  else if (  distribution_type == "Binomial") {
    print(paste0("Binomial distribution parameters - p value is ",
                 mean_data/length(data_input)))
  }
  else if (  distribution_type == "Uniform") {
    print(paste0("Uniform distribution parameters - a value is ", min(data_input),
                 " b value is ", max(data_input)))
```

```r
}

 else if (  distribution_type == "Gamma") {
  x<- data_input
  n = length(data_input)
  a <- 1
 # Defining derivative functions for newton-raphson
 f1 <- function(a) -n*digamma(a)- n*log(mean(x)) + n*log(a) + sum(log(x))
 f2 <- function(a) -n*trigamma(a) + n/a

 #Newton-Raphson for 60 iterations
 for( i in 1:60) a <- a - f1(a)/f2(a)
 # Beta value
 b <- mean(x)/a

 print(paste0("Estimate of alpha is ",a))
 # As input to R is inverse of beta
 print(paste0("Estimate of beta is ",1/b))
 }


else if (  distribution_type == "Beta") {
  x <- data_input
 # Initial guesses
 a2 <- 1
 b2<- 1
 # Running Newton-Raphson 20 times
 for (i in 1:20){
   # Matrices to hold estimates
 parm <- matrix(c(a2,b2), nrow=2)
 # Two components by partial derivatives with respect to alpha and beta
 f<- matrix(c(digamma(a2+b2) - digamma(a2) + mean(log(x)),
             digamma(a2+b2)- digamma(b2)+ mean(log(1-x))), nrow=2)
 # Solving these using jacobian matric
 J <- solve(matrix(c(trigamma(a2+b2)-trigamma(a2),
                    trigamma(a2+b2), trigamma(a2+b2),
                    trigamma(a2+b2)-trigamma(b2)), nrow=2,ncol=2))
 # Matrix multiplication
 f2 <- parm -  J%*%f
 # Getting estimates and putting them back inside
 a2 <- f2[1,]
 b2 <- f2[2,]
 }

 print(paste0("Estimate of alpha is ",a2))
 # As input to R is inverse of beta
 print(paste0("Estimate of beta is ",b2))

}

else if (distribution_type=="multinomial")
{
  n_row = nrow(data_input)
```

```r
    prob = c(0,0,0,0)
    p=data_input/length(data_input)
    for(i in 1:n_row)
      prob[i] = sum(p[i,])
    print(prob)
    #print(paste0("Uniform multinomial parameters - n ", list (n=n)))
    #print(paste0("Uniform multinomial parameters - p ",prob))
  }
  else if (distribution_type=="multivariatenormal")
  {
    mean_data <- colMeans(data_input)
    x_sub_mean <- data-mean_data
    cov <- matrix(c(0,0,0,0),2,2)
    for (i in 1:10000) {
      prod <- x_sub_mean[i,] %*% t(x_sub_mean[i,])
      cov <- cov + prod
    }
    covariance <- round(cov / 10000, 2)
    print(covariance)
    print(paste0("Uniform distribution parameters - mean value is ", mean_data,
                 " Sigma value is ", covariance))
  }

 }
```

Testing distributions

Testing different distributions below:

```r
# Bernoulli is Binomial with size as 1
 bernoulli_data <- rbinom(1000, 1, 0.75)
 distribution_type = 'Bernoulli'
 Parameter_finder(distribution_type,bernoulli_data)
```

```
## [1] "Bernoulli distribution parameters - p value is0.747"
```

```r
 #Binomial distribution
 binom_data <- rbinom(100, 1000, 0.75)
 distribution_type = 'Binomial'
 Parameter_finder(distribution_type,binom_data)
```

```
## [1] "Binomial distribution parameters - p value is 7.4865"
```

```r
 # Geometric distribution. Here parameter couln't
 # be estimated even with really high value of N
 geom_data <- rgeom(100000, 0.25)
 distribution_type = 'Geometric'
 Parameter_finder(distribution_type,geom_data)
```

```
## [1] "Geometric distribution parameters - p value is 0.335200399558876"
```

```r
# Poisson distribution. Lamdba is accurately estimated
poisson_data <- rpois(40000, lambda = 3)
distribution_type = 'Poisson'
Parameter_finder(distribution_type,poisson_data)
```

```
## [1] "Poisson distribution parameters - Lambda value is 2.9942"
```

```r
#Uniform distribution data.
#Here a and B are accurately estimated if gap between them is large enough.
#If not,it seems inaccurate
 uniform_data <- runif(1000000, 1, 100)
distribution_type = 'Uniform'
Parameter_finder(distribution_type,uniform_data)
```

```
## [1] "Uniform distribution parameters - a value is 1.0001723235473 b value is 99.9998862701468"
```

```r
# Normal distribution
norm_data <- rnorm(100000, 20, 2)
distribution_type = 'Normal'
Parameter_finder(distribution_type,norm_data)
```

```
## [1] "Normal distribution parameters - Mean value is 20.0055546946028"
## [1] "Standard deviation is 2.00606157648233"
```

```r
# Exponential distribution
exp_data <- rexp(100000, 5)
distribution_type = 'Exponential'
Parameter_finder(distribution_type,exp_data)
```

```
## [1] "Exponential distribution parameters - rate value is 4.99393900877355"
```

```r
# Gamma distribution
gamma_data <- rgamma(10000, 2, 3)
distribution_type = 'Gamma'
Parameter_finder(distribution_type,gamma_data)
```

```
## [1] "Estimate of alpha is 1.98028553040969"
## [1] "Estimate of beta is 2.96187072271242"
```

```r
# Beta distribution
beta_data <- rbeta(10000, 2, 8)
distribution_type = 'Beta'
Parameter_finder(distribution_type,beta_data)
```

```
## [1] "Estimate of alpha is 2.03872132968364"
## [1] "Estimate of beta is 8.22648791428468"
```

```r
#Multinomial Distribution
 p = c(0.20,0.40,0.05,0.30)
 data = rmultinom(10000,size=4,p)
 Parameter_finder("multinomial", data)
```

```
## [1] 0.212825 0.418100 0.051075 0.318000
```

```r
# Multi variate normal distribution
Sum = matrix(c(9,6,6,16),2,2)
data = mvrnorm(n = 10000, c(4, 5), Sum)
distribution_type = 'multivariatenormal'
Parameter_finder(distribution_type, data)
```

```
##      [,1]  [,2]
## [1,] 9.55  6.05
## [2,] 6.05 16.71
## [1] "Uniform distribution parameters - mean value is 3.95815593839186 Sigma value is 9.55"
## [2] "Uniform distribution parameters - mean value is 4.98533949385021 Sigma value is 6.05"
## [3] "Uniform distribution parameters - mean value is 3.95815593839186 Sigma value is 6.05"
## [4] "Uniform distribution parameters - mean value is 4.98533949385021 Sigma value is 16.71"
```

Maximum likely hood Estimators

Finding MLE usually involves techniques of differential calculus.

To maximize likelihood $L(\theta ; x)$ with respect to θ:

- Calculate the derivative of $L(\theta ; x)$ with respect to θ
- Set the derivative to 0
- Solve the resulting equation of θ.

The computations can be often simplified by maximizing log likelihood function

$$l(\theta ; x) = \log L(\theta ; x)$$

~~Suppose~~

Bernoulli's distribution

Let $(x_1, x_2 \ldots x_n) = x$ be the outcomes of n bernoullis trials, each with success probability p

Likelihood of p is based on joint probability distribution of $x_1, x_2 \ldots x_n$

$$L(p, x) \approx f(x ; p) = \prod_{i=1}^{n} f(x_i, p) = \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i}$$

Scanned with CamScanner

~~diff~~ $\log(L\theta) = \sum_{i=1}^{n} \log p^{x_i}(1-p)^{1-x_i}$

~~diff set to 0~~

$$= \sum_{i=1}^{n} x_i \log p + \left(n - \sum_{i=1}^{n} x_i\right)\log(1-p)$$

differentiate & set it to 0

$$\frac{1}{p}\left[\sum_{i=1}^{n} x_i\right] + \left(n - \sum_{i=1}^{n} px_i\right)\frac{-1}{1-p} = 0$$

Solving for P

$$P = \frac{\sum_{i=1}^{n} x_i}{n}$$

$\therefore$ The MLE estimate is simply the mean of the data.

Exponential distribution

pdf $\Rightarrow f(x) = \left(\frac{1}{\theta}\right) ~~exp~~ e^{(-x/\theta)}$.

$$L(\theta) = \left(\frac{1}{\theta^n}\right) e^{\left(\sum_{i=1}^{n} \frac{-x_i}{\theta}\right)}$$

$$\ln(L(\theta)) = -n \cdot \ln(\theta) - \left(\frac{1}{\theta}\right)\sum_{i=1}^{n} x_i$$

$$\frac{d}{d\theta}\left[\ln(L(\theta))\right] = \left(\frac{-n}{\theta}\right) + \sum_{i=1}^{n}\left(\frac{-x_i}{\theta^2}\right)$$

$$\hat{\theta} = \frac{\sum_{i=1}^{n} x_i}{n} = \overline{x}$$

$\therefore$ MLE is the mean of data

Scanned with CamScanner

Geometric

$$f(x) = (1-p)^{x-1} \cdot p \quad - \text{(pdf)}$$

(likelyhood function)

$$L(p) = p^{n}(1-p)^{\sum_{i=1} x_i - n}$$

$$\ln(L(p)) = \ln(p) + \sum_{i=1}^{n} x_i - n \cdot \ln(1-p)$$

$$\frac{d}{dp}\left(\ln(L(p))\right) = \frac{n}{p} - \left(\sum_{i=1}^{n} x_i - n / (1-p)\right) = 0$$

$$\hat{p} = \frac{n}{\sum_{i=1}^{n} x_i} = \frac{1}{\bar{x}}$$

$\therefore$ MLE is 1/mean(data.

Binomial distribution

$$\text{pdf} \Rightarrow f(x) = \frac{n!}{x!(n-x)!} p^{x}(1-p)^{n-x}$$

$$L(p) = \prod_{i=1}^{n} f(x_i) \Rightarrow \prod_{i=1}^{N} \frac{n!}{x_i!(n-x_i)!} p^{x_i}(1-p)^{n-x_i}$$

$$\ln(L(p)) \Rightarrow \sum_{i=1}^{N} \ln(n!) - \sum_{i=1}^{N} \ln(x_i!) - \sum_{i=1}^{N} (n-x_1)!$$

$$+ \sum_{i=1}^{N} x_i \cdot \ln(p) + \left(n - \sum_{i=1}^{N} x_i\right) \cdot \ln(1-p)$$

$$\frac{d}{dp}\left(\ln(Lp)\right) = \frac{1}{p}\sum_{i=1}^{N} x_i - \frac{1}{1-p}\sum_{i=1}^{N}(n - x_i) = 0$$

$$\hat{p} = \frac{1}{N}\left(\frac{\sum_{i=1}^{N} x_i}{n}\right) \Rightarrow \boxed{\frac{\text{mean}}{\text{len(data)}}}$$

Poisson distribution

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

$$L(\lambda) = \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} \Rightarrow e^{-\lambda n}\left[\frac{\lambda^{\prod_{i=1}^{n} x_i}}{\prod_{i=1}^{n}(x_i)!}\right]$$

$$\ln(L(\lambda)) = -\lambda n + \sum_{i=1}^{n} x_i \ln(\lambda) - \ln\left(\prod_{i=1}^{n} x_i\right)$$

$$\frac{d}{d\lambda}\left(\ln(L(\lambda))\right) = -n + \sum_{i=1}^{n} x_i \frac{1}{\lambda} = 0$$

$$\hat{\lambda} = \frac{\sum_{i=1}^{n} x_i}{n} \Rightarrow \text{mean of the data}$$

Uniform distribution

$$0 < x_i < \theta$$

$$\begin{cases} f(x_i) = \frac{1}{\theta} \\ f(x) = 0 & \text{else} \end{cases}$$

$$L(\theta) = \prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n}\left(\frac{1}{\theta}\right) = \theta^{-n}$$

$$\ln(L\theta) = -n\ln(\theta)$$

Scanned with CamScanner

$$\frac{d}{d\theta}\left(\ln(L(\theta))\right) = -\frac{n}{\theta}$$

this is <0 for $\theta > 0$

$L(\theta)$ is a decreasing function &

is maximized at $\theta = x_n$.

$\therefore$ MLE $\Rightarrow$ $\hat{\theta} = x_n$ max(data)

Normal distribution

$$f(x) = \left(2\pi\sigma^2\right)^{-1/2} e^{\left(-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}\right)}$$

$$L(\mu, \sigma^2) \Rightarrow \left(2\pi\sigma^2\right)^{-n/2} e^{\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}(x_j-\mu)^2\right)}$$

$$\ln\left(L(\mu, \sigma^2)\right) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^{n}(x_j-\mu)^2$$

Now for mean

$$\frac{d}{d\mu}\left(\ln(L(\mu))\right) \Rightarrow \frac{1}{\sigma^2}\left(\sum_{j=1}^{n} x_j - \mu\right) = 0$$

$$\sum_{j=1}^{n} x_j - n\mu = 0$$

$$\hat{\mu} = \frac{\sum_{j=1}^{n} x_j}{n} \Rightarrow \text{mean}$$

Scanned with CamScanner

for $\hat{\sigma}^2$

$$\frac{d}{d\sigma^2}\left[\ln\left(L(\sigma^2)\right)\right] = \frac{-n}{2\sigma^2} - \left[\frac{1}{2}\sum_{j=1}^{n}(x_j-\mu)^2\right] \times \frac{d}{d\sigma^2}\left(\frac{1}{\sigma^2}\right)$$

$$\Rightarrow \frac{-n}{2\sigma^2} + \left[\frac{1}{2}\sum_{j=1}^{n}(x_j-\mu)^2\right]\left(\frac{1}{(\sigma^2)^2}\right)$$

$$\Rightarrow \frac{1}{2\sigma^2}\left[\frac{1}{\sigma^2}\sum_{j=1}^{n}(x_j-\mu)^2 - n\right] = 0$$

if we rule out $\sigma^2 = 0$, is equal too

only if $\sigma^2 = \frac{\sum_{j=1}^{n}(x_j-\mu)^2}{n}$

Multivariate normal distribution (variance)

Joint probability distribution function is

$$f(x) = (2\pi)^{-k/2}\left|\det(V_0)\right|^{-1/2} e^{-1/2(x-\mu)^T V_0^{-1}(x-\mu)}$$

$\mu \rightarrow$ mean $\quad V_0 \rightarrow$ Covariance matrix $(k \times k)$

$$L(\mu, V) = (2\pi)^{-nk/2}\left(\det(V)\right)^{-n/2} e^{-\frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^T V^{-1}(x_i-\mu)}$$

$$\ln\left(L(\mu, V)\right) \Rightarrow \frac{-nk}{2}\ln(2\pi) - \frac{n}{2}\ln\left(\det(V)\right) - \frac{1}{2}\sum_{j=1}^{n}(x_j-\mu)^T V^{-1}(x_j-\mu)$$

Scanned with CamScanner

with respect to mean vector

$$\Rightarrow \sum_{i=1}^{n} V^{-1}(x_j - \mu)$$

$$= \frac{V}{V}\frac{1}{V} \sum_{i=1}^{n} x_j - n\mu = 0$$

$$\boxed{\hat{\mu} = \frac{1}{n} \sum_{i=n}^{n} x_i}$$ mean of the data

gradient of log likelyhood with respect to precision matrix

$$\Rightarrow \frac{n}{2} \frac{d}{dV^{-1}} \ln(\det(V^{-1})) - \frac{1}{2} \frac{d}{dV^{-1}} \left(\sum_{i=1}^{n} \operatorname{tr}\left[(x_j - \mu)^T V^{-1} \times (x_i - \mu) \right] \right)$$

$$= \frac{n}{2} V^T - \frac{1}{2} \frac{d}{dV^{-1}} \left(\sum_{i=1}^{n} \operatorname{tr}\left[(x_j - \mu)(x_j - \mu)^T V^{-1} \right] \right)$$

$$= \frac{n}{2} V^T - \frac{1}{2} \left(\sum_{j=1}^{n} (x_i - \mu)(x_i - \mu)^T \right)^T$$

Transpose on both sides after setting it to 0

we get $\hat{V} = \frac{1}{n} \left(\sum_{i=1}^{n} (x - \mu)(x_i - \mu)^T \right)$

Multinomial distribution

The pdf is defined by

$$f_P(n) = n! \prod_x \frac{p_x^{n_x}}{n_x!}$$

for some fixed number of observations n, $L(p) = f_p(n)$ & constraint $C(p) = 1$

$C(p) = \sum_x p_x$.

To maximize L, the gradient of L & gradient of C are colinear, that is there exists λ such that for every x

$$\frac{d}{dp_x} L(p) = \lambda \frac{d}{dp_x} C(p)$$

$$\frac{n_x}{p_x} L(p) = \lambda$$

ie p_x should be proportional to n_x

$\because \sum_x p_x = 1$, we finally get $\hat{p}_x = \frac{n_x}{n}$ for every x.

$\Rightarrow$ In the code probability ~~each~~ $n = 10000$ & the size is given & we estimate $\hat{p}$ for give probabilities.

Gamma distribution derivation

Distribution is given by

$$f(x) = \frac{1}{\beta^{\alpha}\,\Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta} \quad , x > 0$$

$$\Pi(f_{x_1 \dots n}) = \left(\frac{x_1^{\alpha-1} e^{-x_1/\beta}}{\beta^{\alpha}\Gamma(\alpha)}\right) \cdots \frac{\left(x_n^{\alpha-1} e^{-x_n/\beta}\right)}{\beta^{\alpha}\Gamma(\alpha)}$$

$$= \frac{(x_1 x_2 x_3 \dots x_n)^{(\alpha-1)} e^{-\frac{1}{\beta}\sum x_i}}{\beta^{n\alpha}\left(\Gamma(\alpha)\right)^{n}}$$

Taking log

$$\ln(\Pi(f_{x_1 \dots n})) = (\alpha-1)\ln(x_1 x_2 \dots x_n) - \frac{1}{\beta}\sum x_i - n\alpha\ln\beta - n\ln(\Gamma(\alpha))$$

$$= (\alpha-1)\sum_{i=1}^{n}\ln x - \frac{1}{\beta}\sum x_i - n\alpha\ln\beta - n\ln(\Gamma(\alpha))$$

Taking derivative with respect to β we get

$$= \frac{\sum x_i}{\beta^2} - \frac{n\alpha}{\beta} = 0 \Rightarrow n\alpha = \frac{\sum x_i}{\beta}$$

$$\Rightarrow \boxed{\beta = \frac{\bar{x}}{\alpha}}$$

Using $\beta = \frac{\bar{x}}{\alpha}$ and putting in original function, we get $f(\alpha)$ in terms of α

Taking derivative w.r.t α

$$\Rightarrow \frac{d}{d(\alpha)} \Rightarrow \sum_{i=1}^{n} \ln(x_i) - n \ln\beta - n \frac{d}{d\alpha} \ln(\Gamma(\alpha))$$

This is digamma

Using $\beta = \frac{\bar{x}}{\alpha}$

$$\Rightarrow \sum_{i=1}^{n} \ln x_i - n \ln\left(\frac{\bar{x}}{\alpha}\right) - n \, \text{digamma}(\alpha)$$

$$\Rightarrow = n \, \text{digamma}(\alpha) - n \log(\bar{x}) + n \ln(\alpha) + \sum_{i=1}^{n} \ln x_i$$

Equate this to 0

So second derivative function, lets call this $f_1(x)$

is $f_1(x) = -n \, \text{digamma}(\alpha) - n \log(\bar{x}) + n \ln(\alpha) + \sum_{i=1}^{n} \ln x_i$

Taking second derivative w.r.t α

$$f_2(x) = -n \, \text{trigamma}(\alpha) + \frac{n}{\alpha} = 0$$

We use these two functions in Newton-Raphson method until we find a value of α. Number of iterations will be 50. But can be any number until it converges

Beta distribution:

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \quad \alpha,\beta>0$$

MLE

$$\prod(f(x)) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x_1^{\alpha-1}(1-x_1)^{\beta-1} \cdots$$

$$\cdots \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x_n^{\alpha-1}(1-x_n)^{\beta-1}$$

~~Taking ln~~

$$\Rightarrow \left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right)^n \left(\prod x_i\right)^{\alpha-1}\left(\prod(1-x_i)\right)^{\beta-1}$$

Taking ln $\Rightarrow$

$$\ln\left(\prod f(x)\right) \Rightarrow n\ln(\Gamma(\alpha+\beta)) - n\ln(\Gamma(\alpha)) - n\ln(\Gamma(\beta))$$

$$+ (\alpha-1)\sum_{i=1}^{n}\ln(x_i) + (\beta-1)\sum_{i=1}^{n}\ln(1-x_i)$$

Partial w.r.t. α

$$\frac{\partial l}{\partial \alpha} = \frac{n.\Gamma'(\alpha+\beta)}{\Gamma(\alpha+\beta)} - n\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} + \sum_{i=1}^{n} \ln(x_i)$$

$$= 0$$

w.r.t β

$$\frac{\partial l}{\partial \beta} = \frac{n\Gamma'(\alpha+\beta)}{\Gamma(\alpha+\beta)} - n\frac{\Gamma'(\beta)}{\Gamma(\beta)} + \sum_{i=1}^{n} \ln(1-x_i)$$

$$= 0$$

dividing by n

$$\frac{\partial l}{\partial \alpha} \Rightarrow \frac{\text{digamma}(\alpha+\beta)}{\cancel{\Gamma(\alpha+\beta)}} - \text{digamma}(\alpha) + \frac{1}{n}\sum_{i=1}^{n} \ln(x_i)$$

$$\frac{\partial l}{\partial \beta} \Rightarrow \text{digamma}(\alpha+\beta) - \text{digamma}(\beta) + \frac{1}{n}\sum_{i=1}^{n} \ln(1-x_i)$$

To solve these equations, we need Jacobian matrix

$$J = \begin{vmatrix} \psi'(\alpha+\beta) - \psi'(\alpha) & \psi'(\alpha+\beta) \\ \psi'(\alpha+\beta) & \psi'(\alpha+\beta) - \psi'(\beta) \end{vmatrix}$$

Solving this with a certain initial guess gives us values of α and β.