Assignment-6

GROUP PROJECT: SriKaavya Toodi(st938), Rithvik Ananth(ra830),Aravind reddy(ad1452),Sowmya(s

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.

```
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)))
```

```
}

 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)
 # Solving these using jacobian matric
 J <- solve(matrix(c(trigamma(a2+b2)-trigamma(a2), trigamma(a2+b2), trigamma(a2+b2), trigamma(a2+b2)-
 # 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)
  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 <- cov / 10000
  print(covariance)
  print(paste0("Uniform distribution parameters - Mu value is ", mean_data,
               " Sigma value is ", covariance))
}

}
```

Testing distributions

Testing different distributions below:

```
# 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 is0.724"
```

```
#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 7.507"
```

```
# 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 0.332927162195455"
```

```
# 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 3.00335"
```

```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 1.00011313054711 b value is 99.9999742298387"
```

```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 20.0066271044654"
## [1] "Standard deviation is 2.00539385828563"
```

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

```
## [1] "Exponential distribution parameters - rate value is 4.99205457870902"
```

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

```
## [1] "Estimate of alpha is 1.98007338730874"
## [1] "Estimate of beta is 2.97438096380393"
```

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

```
## [1] "Estimate of alpha is 1.98320520401338"
## [1] "Estimate of beta is 7.89446294878032"
```

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

```
## [1] 0.21230 0.41725 0.05250 0.31795
```

```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,] 9.582251  6.004751
## [2,] 6.004751 16.076323
## [1] "Uniform distribution parameters - Mu value is 4.00804491351362 Sigma value is 9.5822506437674"
## [2] "Uniform distribution parameters - Mu value is 4.99465011934618 Sigma value is 6.00475084439262"
## [3] "Uniform distribution parameters - Mu value is 4.00804491351362 Sigma value is 6.00475084439262"
## [4] "Uniform distribution parameters - Mu value is 4.99465011934618 Sigma value is 16.0763226488718"
```