Probability and Inference HW 3b

Animesh

11/8/2021

Import the necessary libraries

```
library(purrr)
library(MASS)
```

Functions to calculate first and second derivatives for Newton Raphson method

```
f1 <- function(a,n,x){
  return(-n*digamma(a)- n*log(mean(x)) + n*log(a) + sum(log(x)))
}
f2 <- function(a,n){
  return(-n*trigamma(a) + n/a)
}
```

This function is used to calculate parameters for different distributions of data using mean and variance

```
get_dist_params <- function(dist, data) {

  mu <- mean(data); var <- var(data)
  # For the beta distribution, run Newton-Raphson 20 times
  if (dist=="beta") {
    x <- data; a2 <- 1; b2 <- 1
    for (i in 1:20){
    # Matrices to hold estimates
      estim <- matrix(c(a2,b2), nrow=2)
    # Compute the partials 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)

      # Compute partial derivatives by solving the Jacobian matrix
      J <- solve(matrix(c(trigamma(a2+b2)-trigamma(a2),
                          trigamma(a2+b2), trigamma(a2+b2),
                          trigamma(a2+b2)-trigamma(b2)), nrow=2,ncol=2))

      f2 <- estim - J%*%f
      a2 <- f2[1,]
      b2 <- f2[2,]
    }
    cat("Beta dist alpha:",a2,'\n')
    cat("Beta dist beta: ",b2,'\n')
  }
  else if (dist == "bernoulli") {
    cat("Bernoulli p:", mu,'\n')
```

```r
  }
  else if (dist=="binomial") {
    cat("Binomial p: ", mu/length(data),'\n')
  }
  else if (dist == "exponential") {
    cat("Exponential rate value:", 1/mu,'\n')
  }
  # For gamma distribution, we run Newton-Raphson for 60 iterations
  else if (dist=="gamma") {
    x <- data; n <- length(data); a <- 1

    for(i in 1:60){
      a <- a - f1(a,n,x)/f2(a,n)
    }
    # Beta value
    b <- mean(x)/a
    cat("Gamma alpha: ",a,'\n')
    cat("Gamma beta: ",1/b,'\n')
  }
  else if (dist=="geometric") {
    p <- 1/mu
    cat("Geometric p: ", p,'\n')
  }
  else if (dist=="multinomial") {
    n_row <- nrow(data); prob <- c(0,0,0,0);
    p <- data/length(data)
    prob <- rowSums(p)
    cat('Multinomial prob: ',prob,'\n')
  }
  else if (dist=="multivariate_normal") {
    mu <- colMeans(data)
    x_sub_mean <- data - mu
    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, 3)
    cat(covariance,'\n')
    cat("Multivariate mean: ", mu," Multivariate Sigma: ", covariance,'\n')
  }
  else if (dist=="normal") {
    cat("Normal Mean: value: ", mu,'\n')
    cat("Normal Standard deviation: ", sqrt(var),'\n')
  }
  else if (dist=="poisson") {
    cat("Poisson Lambda: ", mu,'\n')
  }
  else if (dist=="uniform") {
    cat("Uniform a: ",min(data)," Uniform b: ", max(data),'\n')
  }
}
```

Let's test it now

Bernoulli with size 1

```
# Bernoulli is Binomial with size as 1
bernoulli_data <- rbinom(1000, 1, 0.75)
dist <- 'Bernoulli'
get_dist_params(dist,bernoulli_data)
```

Binomial Distribution

```
#Binomial distribution
binom_data <- rbinom(100, 1000, 0.75)
dist <- 'Binomial'
get_dist_params(dist,binom_data)
```

Geometric Distribution

```
geom_data <- rgeom(100000, 0.25)
dist <- 'geometric'
get_dist_params(dist,geom_data)
```

```
## Geometric p:  0.3325286
```

Poisson Distribution

```
poisson_data <- rpois(40000, lambda = 3)
dist <- 'poisson'
get_dist_params(dist,poisson_data)
```

```
## Poisson Lambda:  3.020975
```

Uniform Distribution

```
uniform_data <- runif(1000000, 1, 100)
dist <- 'uniform'
get_dist_params(dist,uniform_data)
```

```
## Uniform a:  1.00011  Uniform b:  99.99995
```

Normal Distribution

```
norm_data <- rnorm(100000, 20, 2)
dist <- 'normal'
get_dist_params(dist,norm_data)
```

```
## Normal Mean: value:  19.99056
## Normal Standard deviation:  1.999912
```

Exponential Distribution

```
exp_data <- rexp(100000, 5)
dist <- 'exponential'
get_dist_params(dist,exp_data)
```

```
## Exponential rate value: 4.98074
```

```
gamma_data <- rgamma(10000, 2, 3)
dist <- 'gamma'
get_dist_params(dist,gamma_data)
```

```
## Gamma alpha:  1.97426
```

```
## Gamma beta:  3.010907
```

Beta Distribution

```r
beta_data <- rbeta(10000, 2, 8)
dist <- 'beta'
get_dist_params(dist,beta_data)
```

```
## Beta dist alpha: 1.993774
## Beta dist beta:  7.897157
```

Multinomial Distribution

```r
p = c(0.20,0.40,0.05,0.30)
data <- rmultinom(10000,size=4,p)
dist <- 'multinomial'
get_dist_params(dist, data)
```

```
## Multinomial prob:  0.211375 0.420825 0.05355 0.31425
```

Multivariate Normal Distribution

```r
# Multi variate normal distribution
Sum <- matrix(c(9,6,6,16),2,2)
data <- mvrnorm(n = 10000, c(4, 5), Sum)
dist <- 'multivariate_normal'
get_dist_params(dist, data)
```

```
## 9.448 5.95 5.95 16.563
## Multivariate mean:  4.000595 5.011745  Multivariate Sigma:  9.448 5.95 5.95 16.563
```