Maximum Likelihood Estimation: Numerical Solution for Bernoulli Distribution[1]

The Bernoulli distribution has the probability mass function:

$$f\left(y|\theta\right) \;=\; \theta^{y}\left(1-\theta\right)^{1-y} \tag{1}$$

where: $y=\{0,1\}$ and $\theta\in[0,1]$.

Maximum Likelihood Estimation

Let $y_1,\ldots,y_n$, denote the data. Assume, $\forall i : y_i$ independente random variables, share the same parameter from a Bernoulli distribution described in (1).

$$f\left(y_i|\theta\right) \;=\; \theta^{y_i}\left(1-\theta\right)^{1-y_i} \tag{2}$$

then, their join probability distribution is:

$$\begin{aligned} f\left(y_1,\ldots,y_n|\theta\right) &= \prod_{i=1}^{n} f\left(y_i|\theta\right) \\ &= \prod_{i=1}^{n}\theta^{y_i}\left(1-\theta\right)^{1-y_i} \end{aligned} \tag{3}$$

The likelihood function is:

$$\mathscr{L}\left(\theta|y_1,\ldots,y_n\right) \;=\; f\left(y_1,\ldots,y_n|\theta\right) \tag{4}$$

The log-likelihood function is:

$$\begin{aligned} \ell\left(\theta|y_1,\ldots,y_n\right) &= \log\left(\mathscr{L}\left(\theta|y_1,\ldots,y_n\right)\right) \\ &= \log\prod_{i=1}^{n}\theta^{y_i}\left(1-\theta\right)^{1-y_i} \\ &= \sum_{i=1}^{n}\log\left(\theta^{y_i}\left(1-\theta\right)^{1-y_i}\right) \\ &= \sum_{i=1}^{n}\left(y_i\;\log\theta+\left(1-y_i\right)\;\log\;\left(1-\theta\right)\right) \end{aligned} \tag{5}$$

The maximum likelihood estimator (MLE), denoted by $\hat{\theta}$, is such that:

$$\hat{\theta} \;=\; \operatorname{argmax}\left\{\ell\left(\theta|y_1,\ldots,y_n\right)\right\} \tag{6}$$

[1] *To refer this document and the implemented code, please cite as*: Alvarado, M. (2020, August 12). Maximum Likelihood Estimation: Numerical Solution for Bernoulli Distribution (Version v1.0.0). Zenodo. http://doi.org/10.5281/zenodo.3986441. Also at GitHub: https://github.com/miguel-alvarado-stats/MLE_Bernoulli.

To maximize the log-likelihood function (5), requires the derivative with respect to θ. The resulted function is called the Score function, denoted by $U\left(\theta|y_1,\ldots,y_n\right)$.

$$\begin{aligned} U\left(\theta|y_1,\ldots,y_n\right) &= \frac{\partial\ell\left(\theta|y_1,\ldots,y_n\right)}{\partial\theta} \\ &= \sum_{i=1}^{n}\left(y_i\frac{1}{\theta}+\left(1-y_i\right)\frac{1}{1-\theta}\left(-1\right)\right) \\ &= \frac{1}{\theta\left(1-\theta\right)}\sum_{i=1}^{n}y_i-\frac{n}{1-\theta} \end{aligned} \tag{7}$$

Then, the MLE $\hat{\theta}$ is the solution of:

$$U\left(\theta=\hat{\theta}|y_1,\ldots,y_n,\lambda\right) = 0 \tag{8}$$

Maximum Likelihood Estimation: Newton-Raphson Method

Just for notation, let write equation (8) as:

$$U\left(\theta^*\right) = 0 \tag{9}$$

The equation (9), generally, is a nonlinear equation, that can be aproximate by Taylor Series:

$$U\left(\theta^*\right) \approx U\left(\theta^{(t)}\right)+U'\left(\theta^{(t)}\right)\left(\theta^*-\theta^{(t)}\right) \tag{10}$$

Then, using (10) into (9), and solving for θ^*:

$$\begin{aligned} U\left(\theta^{(t)}\right)+U'\left(\theta^{(t)}\right)\left(\theta^*-\theta^{(t)}\right) &= 0 \\ \theta^* &= \theta^{(t)}-\frac{U\left(\theta^{(t)}\right)}{U'\left(\theta^{(t)}\right)} \end{aligned} \tag{11}$$

where U' is the derivative of the Score function (7) respect of θ.

$$\begin{aligned} U'\left(\theta|y_1,\ldots,y_n,\lambda\right) &= \frac{\partial U\left(\theta|y_1,\ldots,y_n,\lambda\right)}{\partial\theta} \\ &= \frac{2\theta-1}{\theta^2\left(1-\theta\right)^2}\sum_{i=1}^{n}y_i-\frac{n}{\left(1-\theta\right)^2} \end{aligned} \tag{12}$$

Then, with the Newton-Raphson method: starting with an initial guess $\theta^{(1)}$ successive approximations are obtained using (13), until the iterative process converges.

$$\theta^{(t+1)} = \theta^{(t)}-\frac{U\left(\theta^{(t)}\right)}{U'\left(\theta^{(t)}\right)} \tag{13}$$

In order to example the use of Newton-Rapshon method, we use the data of total July rainfall (in millimeters) at Quilpie, Australia stored into the data set "quilpie", where the variable y is a dicotomic variable[2].

[2] This data was taken from package: GLMsData.

```
# load data
data("quilpie")
Y <- quilpie$y
```

We load the code developed into our R function MLE_NR_Bernoulli, stored in the R object with the same name.

```
# load the function to solve by Newton-Raphson
load("MLE_NR_Bernoulli.RData")
```

The function MLE_NR_Bernoulli takes $\theta = 0.5$ as a first guess for the iterative process and, besides some other default parameters that can be modified, only needs the data vector Y.

```
# MLE by Newton-Raphson (NR) for Bernoulli distribution
MLE_NR_Bernoulli(Y)

##      ML Estimator Likelihood            Log-Likelihood
## [1,] "0.5000000"  "3.3881317890172e-21"  "-47.1340082780763"
## [2,] "0.5147059"  "3.48927745358427e-21" "-47.1045922714519"
## [3,] "0.5147059"  "3.48927745358427e-21" "-47.1045922714519"
```

Then, the MLE by Newton-Raphson method: $\hat{\theta} =$ 0.5147059.

Maximum Likelihood Estimation: Fisher-Scoring Method

A distribution belongs to the exponential family if it can be written in the form:

$$f\left(y|\theta\right) \quad = \quad \text{exp}\left\{\frac{a\left(y\right)b\left(\theta\right)-c\left(\theta\right)}{\phi}+d\left(y,\phi\right)\right\} \tag{14}$$

Since (1) can be written as a member of exponential family as in (14):

$$\begin{aligned} f\left(y|\theta,\lambda\right) &= \text{exp}\left\{\text{log}\left(\theta^{y}\left(1-\theta\right)^{1-y}\right)\right\} \\ &= \text{exp}\left\{y\ \text{log}\left(\frac{\theta}{1-\theta}\right)-\left(-\text{log}\ \left(1-\theta\right)\right)\right\} \end{aligned} \tag{15}$$

where, $a\left(y\right)=y$, $b\left(\theta\right)=\ \text{log}\left(\frac{\theta}{1-\theta}\right)$, $c\left(\theta\right)=-\text{log}\ \left(1-\theta\right)$, $\phi=1$, and $d\left(y,\phi\right)=0$.

Then, since the Bernoulli distribution belongs to the exponential family, it can be show that the variance of U, denoted by $\mathcal{J}$, is:

$$\mathcal{J} \quad = \quad \text{Var}\left\{U\right\}=-\text{E}\left\{U'\right\} \tag{16}$$

where:

$$\text{E}\left\{U'\right\} \quad = \quad -\frac{1}{\phi}\left(b''\left(\theta\right)\frac{c'\left(\theta\right)}{b'\left(\theta\right)}-c''\left(\theta\right)\right) \tag{17}$$

For MLE, it is common to approximate U' by its expected value $\mathsf{E}\left\{U'\right\}$. In this case:

$$\begin{aligned}
\mathcal{J} &= -\mathsf{E}\left\{U'\right\} \\
&= \mathsf{E}\left\{-U'\right\} \\
&= \mathsf{E}\left\{-\sum_{i=1}^{n} U_i'\right\} \\
&= \sum_{i=1}^{n} -\mathsf{E}\left\{U_i'\right\} \\
&= \sum_{i=1}^{n} -\frac{1}{\phi}\left(b''\left(\theta\right)\frac{c'\left(\theta\right)}{b'\left(\theta\right)} - c''\left(\theta\right)\right) \qquad (18)
\end{aligned}$$

where, using (1), the previous derivaties:

$$\begin{aligned}
b'\left(\theta\right) &= \frac{1}{\theta\left(1-\theta\right)} \\
b''\left(\theta\right) &= \frac{2\theta-1}{\theta^2\left(1-\theta\right)^2} \\
c'\left(\theta\right) &= \frac{1}{1-\theta} \\
c''\left(\theta\right) &= \frac{1}{\left(1-\theta\right)^2} \\
\frac{1}{\phi} &= 1
\end{aligned}$$

Then, replacing them into (18):

$$\begin{aligned}
\mathcal{J} &= \sum_{i=1}^{n} -\left\{\frac{2\theta-1}{\theta^2\left(1-\theta\right)^2}\frac{\frac{1}{1-\theta}}{\frac{1}{\theta(1-\theta)}} - \frac{1}{\left(1-\theta\right)^2}\right\} \\
&= \sum_{i=1}^{n} \frac{1}{\theta\left(1-\theta\right)} \\
&= \frac{n}{\theta\left(1-\theta\right)} \qquad (19)
\end{aligned}$$

Finally:

$$\begin{aligned}
\mathcal{J} &= -\mathsf{E}\left\{U'\right\} = \frac{n}{\theta\left(1-\theta\right)} \\
-\mathcal{J} &= \mathsf{E}\left\{U'\right\} = \frac{n}{\theta\left(1-\theta\right)} \qquad (20)
\end{aligned}$$

Then, approximating U' by its expected value $\mathsf{E}\left\{U'\right\}$, the equation (13) results into:

$$\theta^{(t+1)} = \theta^{(t)} + \frac{U\left(\theta^{(t)}\right)}{\mathcal{J}\left(\theta^{(t)}\right)} \qquad (21)$$

In order to example the use of Fisher-Scoring method, we use the same data used in the Newton-Rapshon method. We load the code developed into our R function `MLE_FS_Bernoulli`, stored in the R object with the same name.

```
# load the function to solve by Fisher-Scoring
load("MLE_FS_Bernoulli.RData")
```

The function MLE_FS_Bernoulli takes $\theta = 0.5$ as a first guess for the iterative process and, besides some other default parameters that can be modified, only needs the data vector Y.

```
# MLE by Fisher-Scoring (FS) for Bernoulli distribution
MLE_FS_Bernoulli(Y)

##      ML Estimator Likelihood            Log-Likelihood
## [1,] "0.5000000"  "3.3881317890172e-21"  "-47.1340082780763"
## [2,] "0.5147059"  "3.48927745358427e-21" "-47.1045922714519"
## [3,] "0.5147059"  "3.48927745358427e-21" "-47.1045922714519"
```

Then, the MLE by Fisher-Scoring method: $\hat{\theta} =$ 0.5147059.

In summary, the same estimate for the MLE is achieved by both approaches: the Newton-Raphson and the Fisher-Scoring method.

Naive Approach

The main idea behind the Maximum Likelihood (ML) method is to choose those estimates for the unknown parameters that maximize the join probability of our observed data (our sample). Keeping in mind this idea, if we want to get the MLE and avoiding to implement a numerical solution, a naive approach is to set a large range of possible values for unknown parameters, evaluate the log-likelihood function (also, the likelihood function) at each point and the point for which the log-likelihood function (also, the likelihood function) reaches its maximum value, will be our MLE looking for.

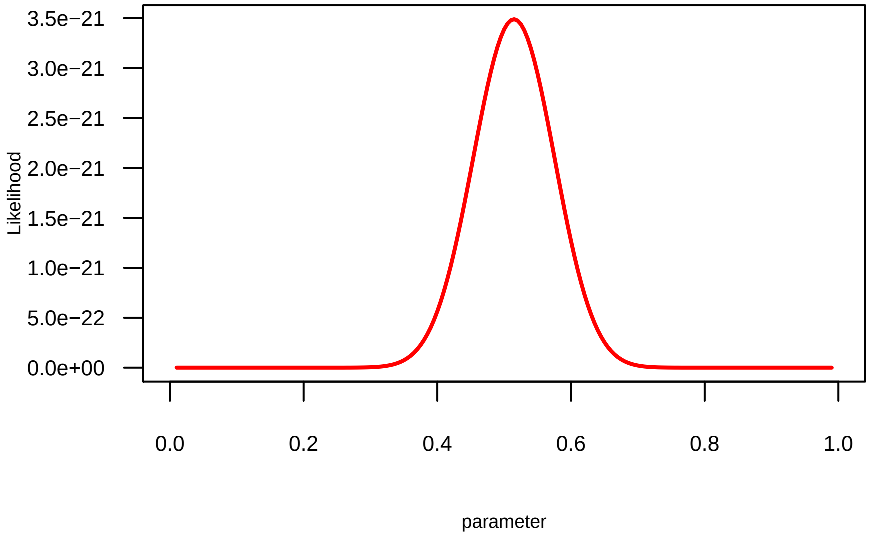
We set a set of values por the parameter, from $\hat{\theta} =$ 0.01 to $\hat{\theta} =$ 0.99, spaced by 0.005; and evaluate (4) and (5) at each point from the set of values.

```
# set a large range of values for the parameter
theta <- seq(0.01, 0.99, 0.005)
tabla <- matrix(c(NA,NA,NA), nrow = length(theta), ncol = 3)
tabla[,1] <- theta

# evaluate the likelihood and the log-likelihood at each point
for (i in 1:length(theta)){
  tabla[i,2] <- prod(dbinom(Y, size = 1, prob = theta[i], log = FALSE))
  tabla[i,3] <- sum(dbinom(Y, size = 1, prob = theta[i], log = TRUE))
}
colnames(tabla) <- c("theta", "Likelihood", "Log-Likelihood")
df_tabla <- as.data.frame(tabla)
```

Then, the plot of likelihood function evaluated at each point:

Figure 1: Likelihooh Function



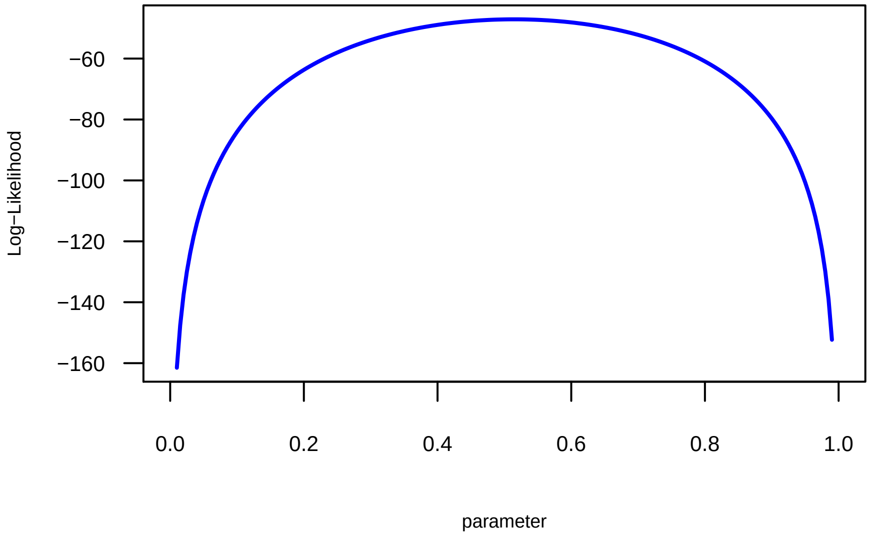
The point for which the likelihood function is maximum, is our MLE from a naive approach:

```
mxl <- max(tabla[,2])
df_tabla %>% filter(`Likelihood` == mxl)

##   theta   Likelihood Log-Likelihood
## 1 0.515 3.489236e-21       -47.1046
```

Also, the plot of log-likelihood function evaluated at each point:

Figure 2: Log−Likelihooh Function



The point for which the log-likelihood function is maximum, which is the same point at the likelihood function reaches its maximum value, is our MLE from a naive approach:

```
mxllog <- max(tabla[,3])
df_tabla %>% filter(`Log-Likelihood` == mxllog)

##   theta   Likelihood Log-Likelihood
## 1 0.515 3.489236e-21       -47.1046
```

As we can see, using this naive approach, we reach a value that is close enough to that which is reached using the numerical solution: the Newton-Raphson and the Fisher-Scoring method.