

The Beta distribution: basics

Let us consider the following parametric Beta model

$$\left\{ \text{Beta}\left(\frac{1}{\theta}, 1\right) ; \theta > 0 \right\}$$

and suppose that a sample x_1, \dots, x_n is observed. In addition, let us recall the PDF of a standard $\text{Beta}(\alpha, \beta)$ distribution, parametrized by two real numbers α and β , $B(\alpha, \beta)$ being the Beta function.

$$f_{\alpha, \beta}(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} \mathbb{1}_{[0,1]}(x)$$

The distribution according to our initial model reduces to

$$f_{\theta}(x) = \frac{1}{\theta} x^{\frac{1}{\theta}-1} \mathbb{1}_{[0,1]}(x)$$

Some considerations

Our goal is to find a Method Of Moments (MoM) estimator and a Maximum Likelihood (ML) estimator for θ .

First, let us show that $B\left(\frac{1}{\theta}, 1\right) = \theta$. Indeed, since

$$B(\alpha, \beta) = \frac{\Gamma(\alpha) \Gamma(\beta)}{\Gamma(\alpha + \beta)}$$

we have that

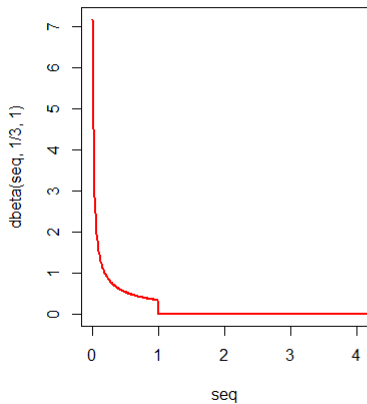
$$B\left(\frac{1}{\theta}, 1\right) = \frac{\Gamma(\theta^{-1}) \Gamma(1)}{\Gamma(\frac{1}{\theta} + 1)} = \frac{\Gamma(\frac{1}{\theta})}{\frac{1}{\theta} \Gamma(\frac{1}{\theta})} = \frac{1}{\frac{1}{\theta}} = \theta$$

Suppose we choose $\theta = 3$, the distribution (also called a Standard Power Function distribution) becomes

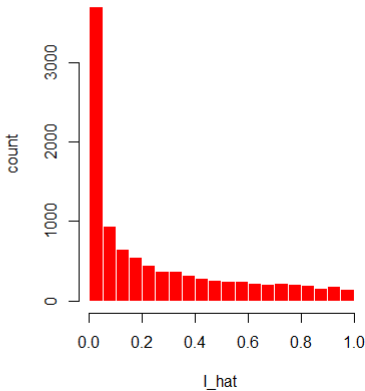
$$f_{\theta}(x) = \frac{1}{3} x^{\frac{-2}{3}} \mathbf{1}_{[0,1]}(x)$$

Let's visualize the PDF and a sample

Beta(1/3,1)



Sample of size 10,000



Expectation

Then, let us show that $E[X] = \frac{1}{1+\theta}$.

$$\begin{aligned} E[X] &= \int_{\text{Support}} x f_{\theta}(x) dx = \int_0^1 x \frac{1}{B\left(\frac{1}{\theta}, 1\right)} x^{\frac{1}{\theta}-1} (1-x)^{1-1} dx \\ &= \int_0^1 \frac{1}{\theta} x^{\frac{1}{\theta}} dx = \frac{1}{\theta} \left[\frac{1}{\frac{1}{\theta} + 1} x^{\frac{1}{\theta} + 1} \right]_0^1 \\ &= \frac{1}{\theta} \frac{\theta}{1 + \theta} 1^{\frac{1}{\theta} + 1} = \frac{1}{1 + \theta} \end{aligned}$$

In our example, we have $E[X] = \frac{1}{1+3} = 1/4$.

```
1 # on average, we should get about 1/4, for theta = 3
2 set.seed(2023)
3 theta = 3
4 mean(rbeta(n = 10000000, shape1 = 1/theta, shape2 = 1) )
5 # [1] 0.2500084
```

Method of Moments estimation

For our Beta model, the **Method Of Moments estimator** (MOM)

$\hat{\theta}_{MOM}$ for θ is given by the expression $\frac{1}{\bar{x}} - 1$, with $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$.

MOM: If $E[\varphi(X)] = h(\theta)$, then we have that $\hat{\theta}_{MOM} = h^{-1}\left(\frac{1}{n} \sum_{i=1}^n \varphi(x_i)\right)$ is a method of moment estimator for θ .

$$\bar{x} = E[X] = \frac{1}{1+\theta} \quad \Leftrightarrow \quad \hat{\theta}_{MOM} = \frac{1 - \bar{x}}{\bar{x}} = \frac{1}{\bar{x}} - 1$$

In our example, we have $E[X] = \frac{1}{1+3} \Leftrightarrow \hat{\theta}_{MOM} = \frac{1-1/4}{1/4} = \frac{3/4}{1/4} = 3$
or equivalently $\frac{1}{1/4} - 1 = 4 - 1 = 3$.

Method of Moments in R

```
1 # Method of Moments estimator
2 MoBeta <- function(x) {
3
4   n <- length(x)
5   sample_moment <- sum(x) / n
6
7   theta_mom <- (1 / sample_moment) -1
8   alpha_mom <- 1 / theta_mom
9
10  output <- NULL
11  output$alpha_mom <- alpha_mom
12  output$theta_mom <- theta_mom
13
14  return(output)
15 }
16
17 # generate artificial data, sample of size 100,000
18 set.seed(2021)
19 x <- rbeta(n = 100000, shape1 = 1/3, shape2 = 1)
20
21 # apply MoMgamma()
22 MoBeta(x = x)
23
24 # $alpha_mom
25 # [1] 0.3369428
26 # $theta_mom
27 # [1] 2.967863
```

Method of Moments in Python

```
1 import numpy as np
2 np.random.seed(2023)
3
4 from scipy.stats import beta
5 # one realization of a Beta(1/3, 1)
6 beta.rvs(a = 1/3, size = 1, b = 1)
7 # array([0.03613599])
8
9 import statistics
10 # on average, we should get 1/4
11 statistics.mean(beta.rvs(a = 1/3, size = 100000, b = 1))
12 # 0.24922457399140857
13
14 # Method of Moments estimator
15 def MoMBeta(x):
16
17     n = len(x)
18     sample_moment = np.sum(x) / n
19
20     theta_mom = (1 / sample_moment) - 1
21     alpha_mom = 1 / theta_mom
22
23     return theta_mom, alpha_mom
24
25 # generate artificial data, sample of size 100,000
26 np.random.seed(2023)
27 x = beta.rvs(a = 1/3, size = 100000, b = 1)
28
29 # apply MoMgamma()
30 MoMBeta(x = x)
31 # (2.9968621974937864, 0.3336823430974835)
```

Maximum Likelihood estimation (1/2)

For our Beta model, the **Maximum Likelihood Estimator** $\hat{\theta}_{MLE}$

for θ is given by the expression $-\frac{\sum_{i=1}^n \ln(x_i)}{n}$. Indeed,

$$\mathcal{L}(\theta \mid \mathbf{x}) = \prod_{i=1}^n f_{\theta}(x_i) = \prod_{i=1}^n \frac{1}{B\left(\frac{1}{\theta}, 1\right)} x_i^{\frac{1}{\theta}-1} (1-x_i)^{1-1} \mathbf{1}_{[0,1]}(x_i)$$

$$= \frac{1}{\theta^n} \left(\prod_{i=1}^n x_i \right)^{\frac{1}{\theta}-1} \prod_{i=1}^n \mathbf{1}_{[0,1]}(x_i)$$

$$l(\theta \mid \mathbf{x}) = -n \ln(\theta) + \left(\frac{1}{\theta} - 1 \right) \sum_{i=1}^n \ln(x_i) + \sum_{i=1}^n \ln(\mathbf{1}_{[0,1]}(x_i))$$

Maximum Likelihood estimation (2/2)

Next, we differentiate the log-likelihood function with respect to the model parameter θ , so that we have

$$\frac{\partial l(\theta | \mathbf{x})}{\partial \theta} = -n \frac{1}{\theta} - \frac{1}{\theta^2} \sum_{i=1}^n \ln(x_i)$$

Setting this derivative equal to 0 and solving for θ then yields

$$-\frac{n}{\theta} = \frac{1}{\theta^2} \sum_{i=1}^n \ln(x_i) \quad \Leftrightarrow \quad \hat{\theta}_{MLE} = -\frac{1}{n} \sum_{i=1}^n \ln(x_i)$$

which is our MLE. Let's perform ML estimation with R then with Python next

Maximum Likelihood estimation in R

```
1 # generate a random sample of n = 5000 from an Beta distribution
2 set.seed(2023)
3 n = 5000 ; theta = 3
4 xi <- rbeta(n = n, shape1 = 1/theta, shape2 = 1)
5 head(xi)
6 # [1] 0.332392794 0.448662938 0.001354903 0.002210476 0.064676188 0.021291173
7
8 # Closed-form MLE
9 theta_hat_formula = 1 / (- n / sum(log(xi)) )
10 theta_hat_formula
11 # [1] 3.023691
12
13 # Numerical approximation of the MLE
14 mle = optimize(function(theta){sum(dbeta(x = xi, shape1 = 1/theta, shape2 = 1,
15     log = TRUE))},
16     interval = c(0, 10),
17     maximum = TRUE,
18     tol = .Machine$double.eps^0.5)
19 theta_hat = mle$maximum
20 theta_hat
21 # [1] 3.023691
```

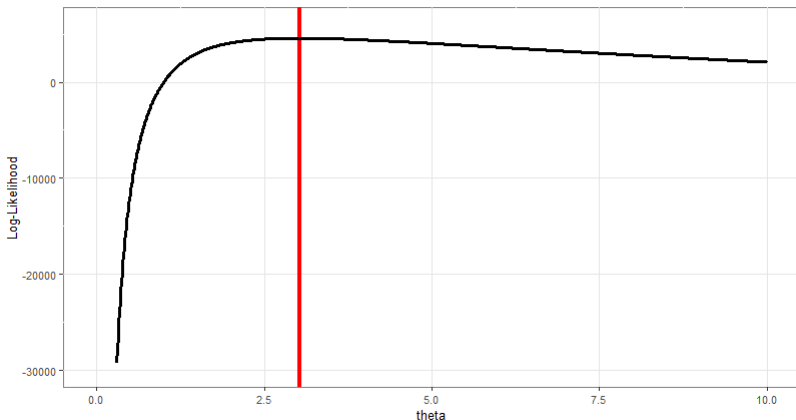
Maximum Likelihood estimation in Python

```
1 import numpy as np
2 # generate a sample of size 5000 from a Beta distribution
3 np.random.seed(2023)
4 n = 5000 ; theta = 3 # true value of the parameter, that we wish to estimate
5 xi = beta.rvs(a = 1/theta, size = 5000, b = 1)
6 print(xi)
7 # [0.03613599 0.61631475 0.00599853 ... 0.23542418 0.00548306 0.82538336]
8
9 # Closed-form MLE
10 theta_hat_formula = 1 / ( - (n / np.sum(np.log(xi))) )
11 theta_hat_formula
12 # [1] 3.0272294817747585
13
14 #T numerical optimization
15 from scipy import stats
16
17 def llikelihood(Theta):
18     # log-likelihood function
19     ll = -np.sum(stats.beta.logpdf(xi, a = 1/Theta, b = 1))
20     return ll
21
22 from scipy.optimize import minimize
23
24 # Numerical approximation of the MLE using minimize()
25 mle = minimize(lllikelihood,
26               x0 = 4,
27               method = 'BFGS')
28 print(mle.x)
29 # [3.02722947]
```

Log-Likelihood function in R

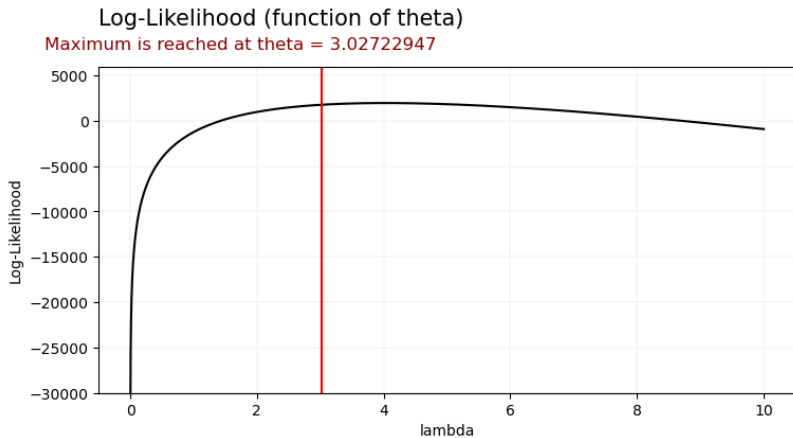
Beta Log-Likelihood function (function of theta)

Maximum is reached at theta = 3.023691



Artificial dataset of size 5,000

Log-Likelihood function in Python



artificial dataset of size 5,000

Further reading and code

Rizzo, M.L. (2019). Statistical Computing with R, Second Edition (2nd ed.). Chapman and Hall/CRC.

<https://doi.org/10.1201/9780429192760>

The R Project for Statistical Computing:

<https://www.r-project.org/>

Python:

<https://www.python.org/>

Accessing R and Python code:

<https://github.com/JRigh/Beta-Model-Estimation>