The Beta distribution: basics

Let us consider the following parametric Beta model

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

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

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

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 moment (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

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

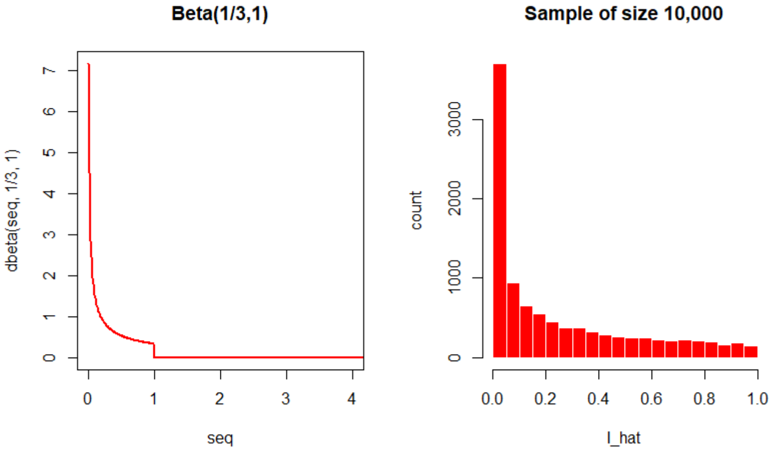
we have that

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

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

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

Let's visualize the PDF and a sample



Expectation

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

$$E[X] = \int_{Support} x f_\theta(x)\ dx \ = \ \int_0^1 x \ \frac{1}{B\left(\boldsymbol{\frac{1}{\theta}}, \mathbf{1}\right)} x^{\frac{1}{\theta}-1}(1-x)^{1-1} dx$$

$$= \int_0^1 \frac{1}{\theta}\ x^{\frac{1}{\theta}}\ dx \ = \ \frac{1}{\theta}\ \left[\frac{1}{\frac{1}{\theta}+1} x^{\frac{1}{\theta}+1}\right]_0^1$$

$$= \frac{1}{\theta}\ \frac{\theta}{1+\theta}\ 1^{\frac{1}{\theta+1}} \ = \ \frac{1}{1+\theta}$$

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

```r
# on average, we should get about 1/4, for theta = 3
set.seed(2023)
theta = 3
mean(rbeta(n = 10000000, shape1 = 1/theta, shape2 = 1) )
# [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 $\frac{1}{\overline{x}} - 1$, with $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

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.

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

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

```r
# Method of Moments estimator
MoBeta <- function(x) {

  n <- length(x)
  sample_moment <- sum(x) / n

  theta_mom <- (1 / sample_moment) -1
  alpha_mom <- 1 / theta_mom

  output <- NULL
  output$alpha_mom <- alpha_mom
  output$theta_mom <- theta_mom

  return(output)
}

# generate artificial data, sample of size 100,000
set.seed(2021)
x <- rbeta(n = 100000, shape1 = 1/3, shape2 = 1)

# apply MoMgamma()
MoBeta(x = x)

# $alpha_mom
# [1] 0.3369428
# $theta_mom
# [1] 2.967863
```

Method of moments in Python

```
import numpy as np
np.random.seed(2023)

from scipy.stats import beta
# one realization of a Beta(1/3, 1)
beta.rvs(a = 1/3, size = 1, b = 1)
# array([0.03613599])

import statistics
# on average, we should get 1/4
statistics.mean(beta.rvs(a = 1/3, size = 100000, b = 1))
# 0.24922457399140857

# Method of Moments estimator
def MoMBeta(x):

  n = len(x)
  sample_moment = np.sum(x) / n

  theta_mom = (1 / sample_moment) -1
  alpha_mom = 1 / theta_mom

  return theta_mom, alpha_mom

# generate artificial data, sample of size 100,000
np.random.seed(2023)
x = beta.rvs(a = 1/3, size = 100000, b = 1)

# apply MoMgamma()
MoMBeta(x = x)
# (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,

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

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

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

Maximum Likelihood estimation (2/2)

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

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

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

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

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

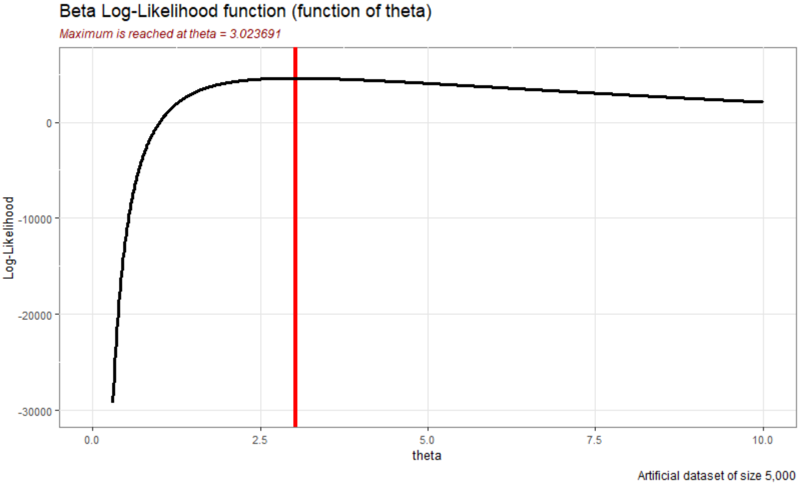
Maximum Likelihood estimation in R

```r
# generate a random sample of n = 5000 from an Beta distribution
set.seed(2023)
n = 5000 ; theta = 3
xi <- rbeta(n = n, shape1 = 1/theta, shape2 = 1)
head(xi)
# [1] 0.332392794 0.448662938 0.001354903 0.002210476 0.064676188 0.021291173

# Closed-form MLE
theta_hat_formula =  1 / (- n / sum(log(xi))  )
theta_hat_formula
# [1] 3.023691

# Numerical approximation of the MLE
mle = optimize(function(theta){sum(dbeta(x = xi, shape1 = 1/theta, shape2 = 1,
     log = TRUE))},
               interval = c(0, 10),
               maximum = TRUE,
               tol = .Machine$double.eps^0.5)

theta_hat = mle$maximum
theta_hat
# [1] 3.023691
```

Maximum Likelihood estimation in Python

```
import numpy as np
# generate a sample of size 5000 from a Beta distribution
np.random.seed(2023)
n = 5000 ;theta = 3 # true value of the parameter, that we wish to estimate
xi = beta.rvs(a = 1/theta, size = 5000, b = 1)
print(xi)
# [0.03613599 0.61631475 0.00599853 ... 0.23542418 0.00548306 0.82538336]

# Closed-form MLE
theta_hat_formula = 1 / ( - (n / np.sum(np.log(xi))) )
theta_hat_formula
# [1] 3.0272294817747585

#T numerical optimization
from scipy import stats

def llikelihood(Theta):
    # log-likelihood function
    ll = -np.sum(stats.beta.logpdf(xi, a = 1/Theta, b = 1))
    return ll

from scipy.optimize import minimize

# Numerical approximation of the MLE using minimize()
mle = minimize(llikelihood,
                    x0 = 4,
                    method = 'BFGS')
print(mle.x)
# [3.02722947]
```

Plot the Log-Likelihood function



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