Maximum Likehood Estimator

Introduction

maximum likelihood estimation is one common method for estimating paremater in a parametric model, just like method of moments. We assume $X_1, X_2..., X_n$ be IID with PDF $f(x;\theta)$, define the likelihood function as

$$\mathcal{L}(\theta) = \Pi_{i=1}^{n} f(X_i;\theta)$$

And the log likelihood function is defined by $\ell_n(\theta) = log\mathcal{L}_n(\theta) = \sum_{i=1}^{n} l\,ogf(X_i;\theta)$.

The *maximum likelihood estimator* MLE, denoted by $\widehat{\theta}_n$, is the value of θ that maximizes $\mathcal{L}_n(\theta)$

Derivation

Point mass at a

Notice the probability mass function for point mass distribution is $f(x) = 1$ in $x = a$ and 0 elsewhere. In fact, f have only two values:1 and 0, so does the $\mathcal{L}(\theta)$ which is the product of many fs. We choose the MLE for point mass distribution to be mode among X_n.

Bernoulli

The probability function is $f(x;p) = p^x(1-p)^{1-x}$ for $x = 0,1$, the unknown parameter is p.

$$\mathcal{L}_n(p) = \prod_{i=1}^{n} f\,(X_i;p) = \prod_{i=1}^{n} p^{X_i}\,(1-p)^{1-X_i} = p^S(1-p)^{n-S}$$

where $S = \sum_i X_i$. Hence,

$$\ell_n(p) = Slogp + (n-S)log(1-p)$$

Take the derivation and set it equal to 0 gave us $\widehat{p}_n = \frac{S}{n}$

Binomial(N,p)

$$\mathcal{L}(p) = \prod_{i}^{n}(f(y_i)) = \prod_{i}^{n}\left[\binom{N}{y_i} p^{y_i}(1-p)^{N-yi}\right] = \left[\prod_{i}^{n}\binom{N}{y_i}\right] p^{\sum_1^n y_i}(1-p)^{nN-\sum_1^n y_i}\,.$$

$$\widehat{p} = \underset{p}{\operatorname{argmax}} \left[\left[\prod_{i}^{n} \binom{N}{y_i} \right] p^{\sum_1^n y_i} (1-p)^{nN - \sum_1^n y_i} \right]$$

$$\begin{aligned} \widehat{p} &= \underset{p}{\operatorname{argmax}} \ \ln \left[\left[\prod_{i}^{n} \binom{N}{y_i} \right] p^{\sum_1^n y_i} (1-p)^{nN - \sum_1^n y_i} \right] \\ &= \underset{p}{\operatorname{argmax}} \sum_{i=1}^{n} \left[\ln \binom{N}{y_i} + \left(\sum_{i=1}^{n} y_i \right) \ln(p) + \left(nN - \sum_{i=1}^{n} y_i \right) \ln(1-p) \right] \end{aligned}$$

$$0 + \frac{\left(\sum_{i=1}^{n} y_i\right)}{\widehat{p}} - \frac{\left(nN - \sum_{i=1}^{n} y_i\right)}{1-\widehat{p}} = 0\,.$$

$$\widehat{p} = \frac{\sum_{i=1}^{n} y_i}{nN} = \frac{\overline{y}}{N}$$

Geometric(p)

$P(X = k) = p(1-p)^k$ where $k \geq 1$

$$\mathcal{L}(p) = (1-p)^{x_1 - 1} p (1-p)^{x_2 - 1} p \ldots (1-p)^{x_n - 1} p = p^n (1-p)^{\sum_1^n x_i - n}$$

$$ln\mathcal{L}(p) = \sum_{i=1}^{n} x_i \, ln(p) + \left(n - \sum_{i=1}^{n} x_i \right) ln(1-p)$$

$$\frac{dln\mathcal{L}(p)}{dp} = \frac{1}{p} \sum_{i=1}^{n} x_i + \frac{1}{1-p} \left(n - \sum_{i=1}^{n} x_i \right) = 0$$

$$\widehat{p} = \frac{n}{\left(\sum_1^n x_i\right)}$$

Poisson Distribution(λ)

$$\mathcal{L}(\lambda) = \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} = e^{-n\lambda} \frac{\lambda^{\sum_1^n x_i}}{\prod_{i=1}^{n} x_i!}$$

$$lnL(\lambda) = -n\lambda + \sum_{1}^{n} x_i \, ln(\lambda) - ln \left(\prod_{i=1}^{n} x_i! \right)$$

$$\frac{dlnL(\lambda)}{dp} = -n + \sum_{1}^{n} \frac{x_i}{\lambda}$$

$$\hat{\lambda} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Uniform(a,b)

$$\mathcal{L}_n(a,b) = \frac{1}{(b-a)^n}$$

if all $a \le X_1, X_2, \ldots, X_n \le b$ else $\mathcal{L} = 0$ To maximize, we need to minize $b - a$, the boundary condition is to choose $\hat{a} = min_i(X_i)$ and $\hat{b} = max_i(X_i)$

Normal(μ, σ^2)

$$\mathcal{L}_n(\mu,\sigma) = \prod_i \frac{1}{\sigma} exp\{-\frac{1}{2\sigma^2}(X_i - \mu)^2\}$$

$$= \sigma^{-n} exp\{-\frac{1}{2\sigma^2}\sum_i (X_i - \mu)^2\}$$

$$= \sigma^{-n} exp\{-\frac{nS^2}{2\sigma^2}\} exp\{-\frac{n(\bar{X} - \mu)^2}{2\sigma^2}\}$$

---

where $\overline{X} = n^{-1}\sum_i X_i$, $S^2 = n^{-1}\sum_i (X_i - \overline{X})^2$

$$\ell(\mu,\sigma) = -nlog\sigma - \frac{nS^2}{2\sigma^2} - \frac{n(\overline{X} - \mu)^2}{2\sigma^2}$$

Solving eqations $\frac{\partial \ell(\mu,\sigma)}{\partial \mu} = 0$ and $\frac{\partial \ell(\mu,\sigma)}{\partial \ell \sigma}$ we conclude that $\hat{\mu} = \overline{X}$ and $\hat{\sigma} = S$

Exponential(β)

$$\mathcal{L}(\beta) = \mathcal{L}(\beta; X_1, X_2 \ldots X_n) = \left(\frac{1}{\beta} e^{\frac{-X_1}{\beta}}\right)\left(\frac{1}{\theta} e^{\frac{-X_2}{\beta}}\right) \ldots \left(\frac{1}{\theta} e^{\frac{-X_n}{\beta}}\right) = \frac{1}{\beta^n} exp\left(\frac{-\sum_1^n X_i}{\beta}\right)$$

$$ln\mathcal{L}(\beta) = -nln(\beta) - \frac{1}{\beta}\sum_1^n X_i \, , 0 < \beta < \infty$$

$$\frac{d[ln\mathcal{L}(\theta)]}{d\beta} = \frac{-n}{\beta} + \frac{1}{\beta^2}\sum_1^n X_i = 0$$

$$\hat{\beta} = \frac{\sum_1^n X_i}{n}$$

Gamma(α, β)

$$\mathcal{L}(\alpha,\beta) = (\frac{1}{\beta^{\alpha}\Gamma(\alpha)})^n exp^{-\frac{\sum_i X_i}{\beta}} \prod_{i=1}^{n} X_i^{\alpha-1}$$

$$\ell_n(\alpha,\beta) = -n\alpha log\beta - nlog\Gamma(\alpha) - \frac{\sum_i X_i}{\beta} + \sum_{i=1}^{n}(\alpha-1)logX_i$$

Solve $\frac{\partial \ell_n(\alpha,\beta)}{\partial \beta} = 0$ and $\frac{\partial l_n(\alpha,\beta)}{\partial \alpha} = 0$ yield:

$$\widehat{\beta} = \frac{\sum_i X_i}{n\alpha}$$

and

$$\sum_{i=1}^{n} l\,ogX_i = nog\frac{\sum X_i}{n\alpha} + n\psi(\alpha)$$

where

$$\psi(\alpha) = \frac{dlog\Gamma(\alpha)}{d\alpha}$$

, is called Digamma Function However, no closed form exists for $\widehat{\alpha}$, only numerical solution exist.

we calculate the numerical estimation using Newton-Raphson Refernce

$$l_n{}'(\alpha) = nlog(\frac{\alpha}{\overline{X}}) - n\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} + \sum_{i=1}^{n} l\,ogX_i$$

$$l_n{}''(\alpha) = \frac{n}{\alpha} - n(\frac{\Gamma'(\alpha)}{\Gamma(\alpha)})'$$

$$\hat{\alpha}^{(k)} = \hat{\alpha}^{(k-1)} - \frac{l_n'(\hat{\alpha}^{(k-1)})}{l_n''(\hat{\alpha}^{(k-1)})}$$

$$\hat{\beta} = \frac{\bar{X}}{\hat{\alpha}}$$

Beta(α, β)

$$\mathcal{L}_n(\alpha,\beta) = (\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)})^n \prod_{i=1}^{n}(X_i)^{\alpha-1}\prod_{i=1}^{n}(1-X_i)^{\beta-1}$$

$$\ell_n(\alpha,\beta) = nlog(\Gamma(\alpha+\beta)) - nlog(\Gamma(\beta)) + (\alpha-1)\sum_{i=1}^{n} log(X_i) + (\beta-1)\sum_{i=1}^{n} log(1-X_i)$$

solve $\frac{\partial \ell_n(\alpha,\beta)}{\partial\alpha}$ and $\frac{\partial \ell_n(\alpha,\beta)}{\partial\beta}$ yield no analytical solution. However, numerical solution via Newton-Raphson mehod does exist. For reference, check page 27 on this paper

We set $\widehat{\theta} = (\widehat{\alpha}, \widehat{\beta})$ iteratively:

$$\hat{\theta}_{i+1} = \hat{\theta}_i - G^{-1}g$$

$$g = [g1, g2]$$

$$g1 = \psi(\hat{\alpha}) - \psi(\hat{\alpha}+\hat{\beta}) - \frac{1}{n}\sum_{i=1}^{n} log(X_i)$$

$$g2 = \psi(\hat{\beta}) - \psi(\hat{\alpha}+\hat{\beta}) - \frac{1}{n}\sum_{i=1}^{n} log(1-X_i)$$

$$G = \begin{bmatrix} \frac{dg1}{d\alpha} & \frac{dg2}{d\beta} \\ \frac{dg2}{d\alpha} & \frac{dg2}{d\beta} \end{bmatrix}$$

$$\frac{dg1}{d\alpha} = \psi'(\alpha) - \psi'(\alpha+\beta)$$

$$\frac{dg1}{d\beta} = \frac{dg2}{d\beta} = -\psi'(\alpha+\beta)$$

$$\frac{dg2}{d\beta} = \psi'(\beta) - \psi'(\alpha+\beta)$$

Student t distribution

$$\mathcal{L}_n(v) = (\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma\frac{\nu}{2}})^n \frac{1}{\prod_{i=1}^{n}(1+\frac{X_i^2}{\nu})^{\frac{\nu+1}{2}}}$$

$$l_n(\nu) = nlog(\Gamma(\frac{\nu+1}{2})) - nlog(\Gamma(\frac{\nu}{2})) - \sum_{i=1}^{n} \frac{\nu+1}{2} log(1+\frac{X_i^2}{\nu})$$

$$\frac{\partial l_n(\nu)}{\partial \nu} = \frac{\nu+1}{2}\sum_{i=1}^{n}\frac{X_i^2}{\nu^2} - \frac{1}{2}\sum_{i=1}^{n}(\frac{X_i^2}{\nu}+1) + \frac{n\psi(\frac{\nu+1}{2})}{2} - \frac{n\psi(\frac{\nu}{2})}{2} = 0$$

We cannot get closed-form solution.

χ_p^2 Chi-Squared Distriubtion

$$\mathcal{L}_n(p) = \frac{1}{(\Gamma(p/2)2^{p/2})^n} e^{\frac{-\sum_{i=1}^{n} X_i}{2}} \prod_{i=1}^{n} X_i^{p/2-1}$$

$$l_n(p) = -nlog(\Gamma(p/2)) - \frac{pn}{2}log2 - \sum_{i=1}^{n}\frac{X_i}{2} + (\frac{p}{2}-1)\sum_{i=1}^{n} l\, ogX_i$$

$$\frac{\partial l_n(p)}{\partial p} = -\frac{n}{2}\psi(p/2) - \frac{n}{2}log2 + \frac{1}{2}\sum_{i=1}^{n} l\, ogX_i = 0$$

$$\hat{p} = 2\psi^{-1}(\frac{\sum_{i=1}^{n} l\, ogX_i}{n} - log2)$$

But in fact, digamma function is not monotone. Therefore it doesn't have an inverse function at all. So the formula above is not right! In other words, the student t distributions doesn't have a closed-form expression for MLE.

Program

```
MLE_Beta<-function(x, iteration=100){
  mu=mean(x)
  s2<-var(x)
  theta<-c(mu*(-1+ mu*(1-mu)/s2),
    (1-mu)*(-1+ mu*(1-mu)/s2)
  )
  G<-matrix(0, ncol=2, nrow=2)
  g<-matrix(0, nrow=2, ncol=1)
  for(i in 1:iteration){
    g[1,1]=digamma(theta[1])-digamma(theta[1] + theta[2])-
sum(log(x))/length(x)
    g[2,1]=digamma(theta[2])-digamma(theta[1] + theta[2])-sum(log(1-
x))/length(x)
    G[1,1]=trigamma(theta[1])-trigamma(theta[1]+theta[2])
    G[1,2]=-trigamma(theta[1]+theta[2])
    G[2,1]=-trigamma(theta[1]+theta[2])
    G[2,2]=trigamma(theta[2])-trigamma(theta[1]+theta[2])
```

```
    theta<-theta-solve(G) %*% g
  }
  return( list(shape1=theta[1], shape2=theta[2] ) )
}

MLE_Gamma<-function(x, iteratin=100){
  n<-length(x)
  mean_x<-mean(x)
  alpha<-n*(mean_x^2)/sum((x-mean_x)^2)
  beta<-sum((x-mean_x)^2)/n/mean_x
  for(i in 1:iteratin){
    #first derivative of alpha_k-1
    der1<-n*log(alpha/mean_x)-n*digamma(alpha)+sum(log(x))
    #second derivative of alpha_k-1
    der2<-n/alpha-n*trigamma(alpha)
    #calculate next alpha
    alpha<-alpha-der1/der2
    beta<-mean_x/alpha
  }

  return(list(shape=alpha, scale=beta))
}

MLE<-function(x, type, bino=5){
  if(type=="PointMass") { ux <- unique(x) ; return(
ux[which.max(tabulate(match(x, ux)))]) }
  if (type=="Bernoulli") return(  list(size=1, prob=sum(x)/length(x)) )
  if(type=="Binomial") return(  list(size=bino, prob=mean(x)/bino) )
  if(type=="Geometric") return( list(prob=length(x)/mean(x) ))
  if(type=="Poisson") return( list( lambda=mean(x) ) )
  if(type=="Uniform") return( list(min=min(x), max=max(x) ) )
  if(type=="Normal" ) return( list(mean=mean(x), sd=sqrt(  sum( (x-mean(x)
)^2)/ length(x)  )))
  if(type=="Exponential") return( list(rate=mean(x) ) )
  if(type=="Gamma") return(MLE_Gamma(x))
  if(type=="Beta") return(MLE_Beta(x))
  #if(type=="t") return(2*(a1^2+a2)/(a1^2+a2-1))
  #if(type=="chi") return(2* )
}
```

Test

We combine parametric bootstrapping and KS-test to test the goodness of fit.

Parametric Bootstrapping

In parametric bootstrapping, we sample from our "estimated" distribution $f(x;\hat{\theta}_n)$ instead of the empirical distribution $\hat{F}_n(x)$

Kolmogorov-Smirnov test

In statistics, the Kolmogorov-Smirnov test (K-S test or KS test) is a nonparametric test of the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K-S test), or to compare two samples (two-sample K-S test). It is named after Andrey Kolmogorov and Nikolai Smirnov.

The Kolmogorov-Smirnov statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples. The null distribution of this statistic is calculated under the null hypothesis that the sample is drawn from the reference distribution (in the one-sample case) or that the samples are drawn from the same distribution (in the two-sample case). In each case, the distributions considered under the null hypothesis are continuous distributions but are otherwise unrestricted.

The two-sample K-S test is one of the most useful and general nonparametric methods for comparing two samples, as it is sensitive to differences in both location and shape of the empirical cumulative distribution functions of the two samples.(https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test)

Bootstrap and KS-test code

I created two *partial* functions for later use, which can create a parameters-partially-fed function.

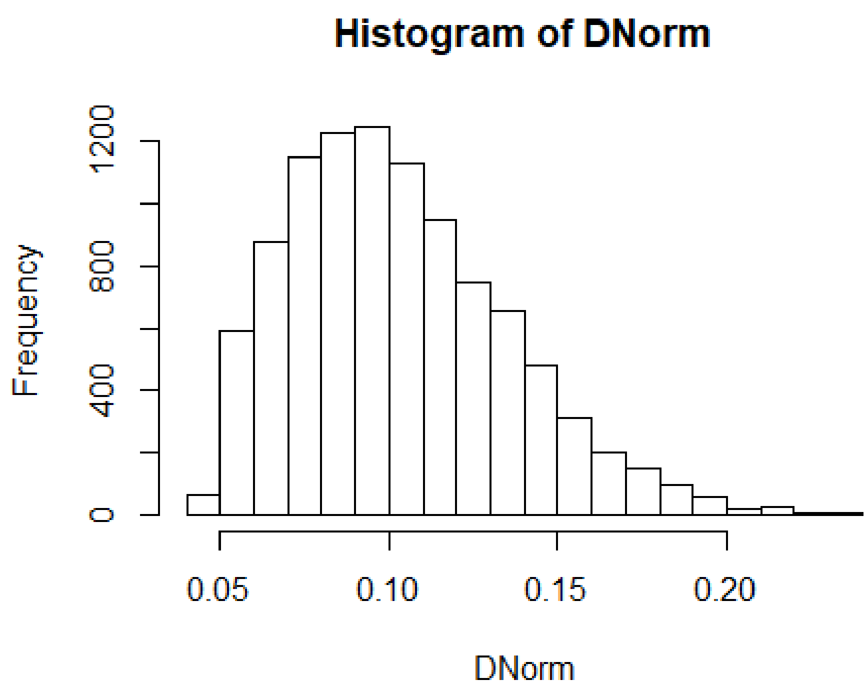
```
partial <- function(f, ...) {
  l <- list(...)
  function(...) {
    do.call(f, c(l, list(...)))
  }
}
partial_list<-function(f, L){
  function(...){
    do.call(f, c(L, list(...)))
  }
}


BootKS<-function(x, n0 , nboot=10000, rvecFUN, Distritype, FUN)
{
Dvec<-NULL
for(i in 1:nboot){
  vec<-rvecFUN(n=n0)
  theta<-MLE(vec, type = Distritype)
  Dvec<-c(Dvec, unname(ks.test(x, partial_list(FUN, theta))$statistic ))
}
```

```
return(Dvec)
}
```

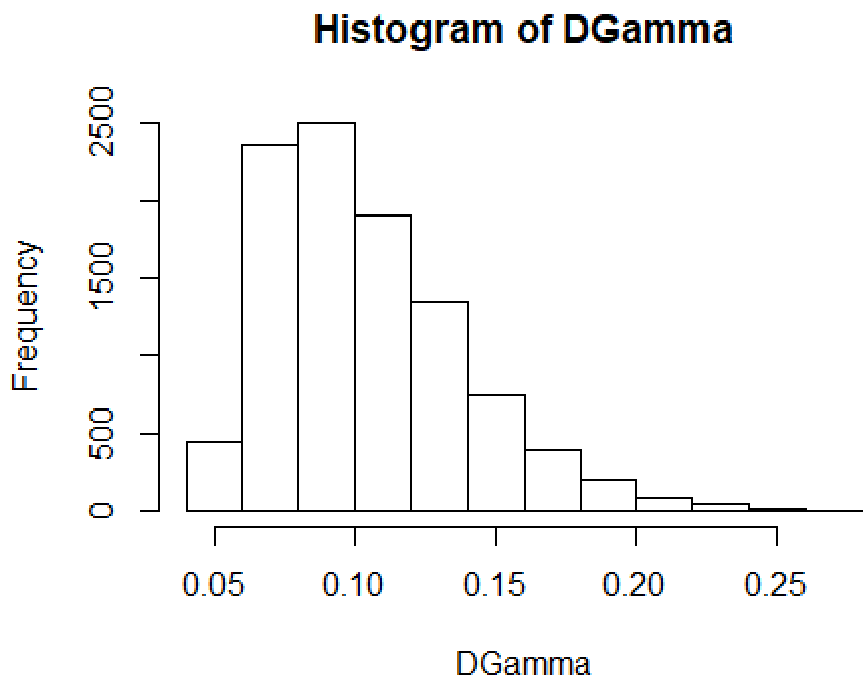
Test with Normal Distribution

```
set.seed(553)
x0<-rnorm(n=100, mean=, sd=6)
theta0<-MLE(x0, type = "Normal")
DNorm<-BootKS(x0, n0=50, rvecFUN = partial_list(rnorm, theta0), Distritype =
"Normal", FUN = pnorm)
hist(DNorm)
```



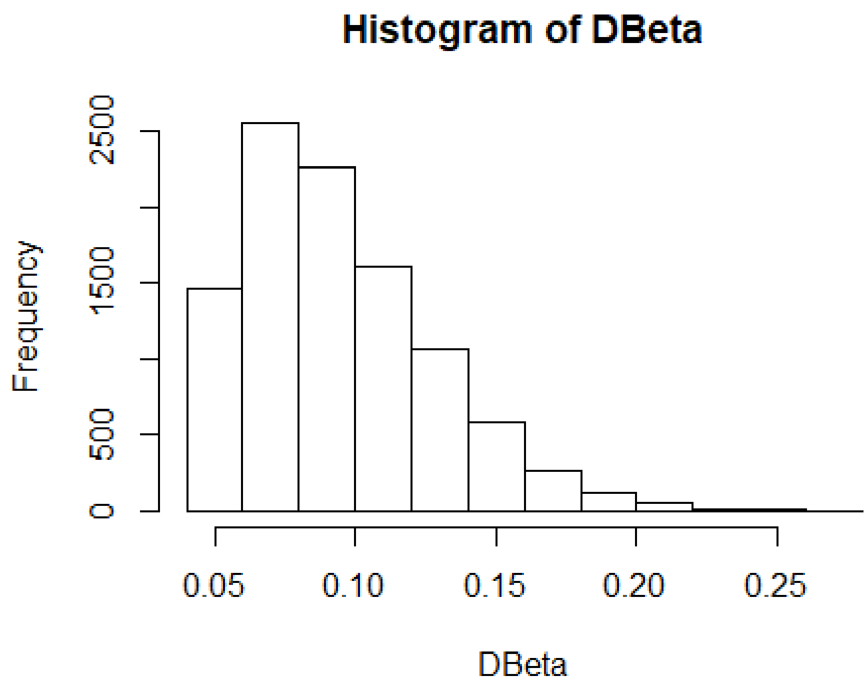
Test with Gamma Distribution

```
set.seed(239)
x1<-rgamma(n=100, shape=5, scale=6)
theta1<-MLE(x1, type = "Gamma")
DGamma<-BootKS(x1, n0=50, rvecFUN = partial_list(rgamma, theta1), Distritype
= "Gamma", FUN=pgamma)
hist(DGamma)
```



Test with Beta Distribution

```
set.seed(461)
x2<-rbeta(n=100, shape1=45, shape2 = 13)
theta2<-MLE(x2, type="Beta")
DBeta<-BootKS(x2,n0=50, rvecFUN = partial_list(rbeta, theta2), Distritype =
"Beta", FUN = pbeta)
hist(DBeta)
```


Histogram of DBeta
Frequency
0
500
1500
2500
0.05
0.10
0.15
0.20
0.25
DBeta