MAXIMUM LIKELIHOOD ESTIMATOR OF THE PARAMETER α OF GAMMA(α,1) DISTRIBUTION

SARTHAK BAGARIA

Solution to Q1: Gamma Sample Generation

We use the standard routine *rgamma* in R language, which uses the following method for generation of Gamma sample:

For $\boldsymbol{\alpha < 1}$. A rejection technique [1] is used based on the majoring function

$$g(x) = x^{\alpha-1}/\Gamma(\alpha) \quad \text{if } 0 < x \leq 1; \qquad g(x) = e^{-x}/\Gamma(\alpha) \quad \text{if } 1 \leq x.$$

Since $e^{-x} \leq 1$ if $0 < x$ and $x^{\alpha-1} \leq 1$ if $\alpha \leq 1$ and $1 \leq x$ the inequality $e^{-x}x^{\alpha-1}/\Gamma(\alpha) = f(x) \leq g(x)$ is valid for all $x > 0$. The function

$$h(x) = x^{\alpha-1}ea/(e+a) \quad \text{if } 0 < x \leq 1; \qquad h(x) = e^{-x}ea/(e+a) \quad \text{if } 1 \leq x$$

is a probability density that is proportional to g(x). Sampling from h(x) is no problem since both parts have easily invertible integrals: with a probability of $e/(e+a)$ an x below 1 (first part of h(x)) is sampled, otherwise the second part of h(x) is used. The rejection test is based on $f(x)/g(x)$ which is e^{-x} or $x^{\alpha-1}$.

For $\boldsymbol{\alpha \geq 1}$. Ahrens and Dieter's modified rejection method [2] is used.

Applying the transformation $x = (\sqrt{\alpha - 1/2} + t/2)^2$, the resulting transformed function g(t) is close to the standard normal density f(t). The mode of g(t) is at $t = 0$, but g(0) is a little larger than $f(0) = 1/\sqrt{2\pi}$. Also, g(t) intersects the standard normal density f(t) only once at some $t = \tau(a) < 0$. Consequently, $g(t) \geq f(t)$ for all $t \geq 0$. This calls for the following modification of Von Neumann's acceptance-rejection technique:
Generate a standard normal deviate T [3]. If $T \geq 0$, accept $x = (\sqrt{\alpha - 1/2} + T/2)^2$ as a gamma(α) sample. For $T \leq \tau(\alpha)$, where f(t) majorizes g(t), the ratio $r(T) = g(T)/f(T)$ can be compared with a (0, 1)-uniform deviate U for an ordinary rejection test. (For simplicity this test is also applied when $\tau(a) < T < 0$. In this case $r(t) > 1$ and T is always accepted.) Obviously rejection occurs with probability $P(H) = \int_{-\infty}^{\tau}(f(t) - g(t))dt = \int_{\tau}^{\infty}(g(t) - f(t))$. Hence, whenever a negative T is rejected, it must be replaced with a new $T \geq \tau(\alpha)$, and this has to be a sample from the difference distribution whose probability density function is proportional to $g(t) - f(t)$ in $[\tau, \infty)$. Sampling from this difference is done by means of a double-exponential hat.

Solution to Q2: Maximum Likelihood Estimation of α

We use the standard routine *fitdistr* in R language, with BFGS method. For Gamma distribution reasonable starting values for the iterative method can be computed by the routine itself.

BFGS Method [4]**.** In Newtons method, we find the new iterate x_{k+1} as a function of x_k as follows. For any point x define $p = x - x_k$, the second order Taylor expansion around x_k is given by

$$m_k(p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T B_k p$$

This defines a quadratic model of the function near the point x_k . Its gradient with respect to x is $m_k(p) = \nabla f_k + B_k p$, and it is minimized at $p_k = -B_k^{-1} \nabla f_k$.

Working with the inverse Hessian H_k in place of B_k, the secant equation becomes $H_k y_{k-1} = s_{k-1}$. The optimization is then : minimize $||H - H_{k-1}||_W$ subject to $H = H^T$, $H y_{k-1} = s_{k-1}$, which has the unique solution

$$H_k = (I - p_k s_{k-1} y_{k-1}^T) H_{k-1} (I - p_{k-1} y_{k-1} s_{k-1}^T) + s_{k-1} p_{k-1} s_{k-1}^T$$

where $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = \nabla f_k - \nabla f_{k-1}$, W is any matrix satisfying $W y_{k-1} = s_{k-1}$, and $||H - H_{k-1}||_W = ||W^{\frac{1}{2}} (H - H_{k-1}) W^{\frac{1}{2}}||$.

Each step of the BFGS method has the form

$$x_{k+1} = x_k - \alpha_k H_k \nabla f_k, \qquad k = 0, 1, 2, \ldots$$

where α_k is computed from a line search procedure to satisfy the Wolfe conditions:

$$f(x_k + \alpha_k p_k) \le f(x_k) + c_1 \alpha_k \nabla f_k^T p_k,$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \ge c_2 \nabla f_k^T p_k$$

with $0 < c_1 < c_2 < 1$.

Solution to Q3 & Q4: Results

The following graphs were plotted for values of MLE of α against the values of α_0, the parameter used for generation of sample. α_0 ranges from .1 to 10 in steps of .1, and for each value of α_0, estimates for 20 samples are calculated. n is the number of Gamma variates in a single sample.

Example outputs from different samples with α for $\alpha_0 = 5$:

$n = 10$:

5.3144648(0.6950958); 4.5355942(0.6368612); 4.6917132(0.6489457);
5.743960(0.725238); 4.0315324(0.5962093); 5.2970827(0.6938487);

$n = 50$:

4.6669183(0.2893656);	4.6980724(0.2904354);	4.9866184(0.3001651);
4.7338926(0.2916606);	4.7629318(0.2926501);	4.927377(0.298193);

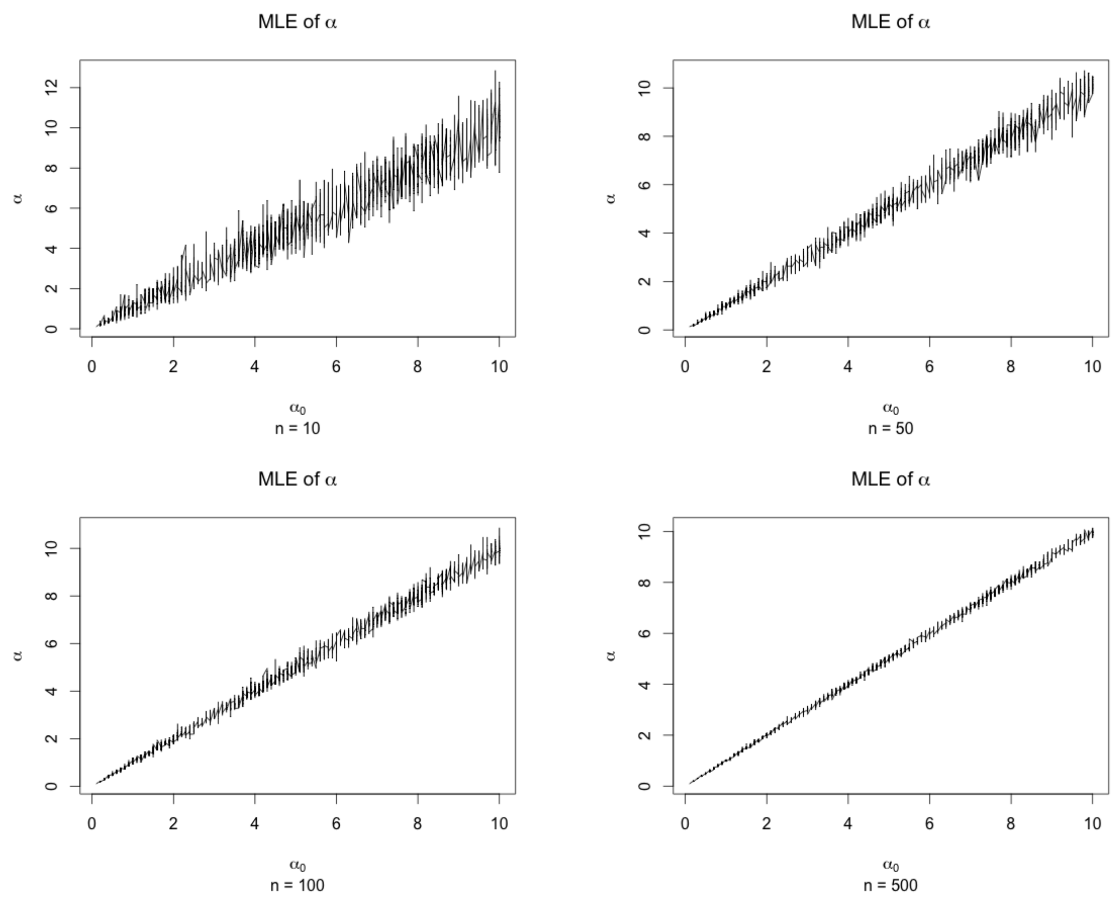
$n = 100$:

5.4493586(0.2228456);	5.1090057(0.2151015);	5.5133350(0.2242719);
4.9219525(0.2107262);	5.2803015(0.2190328);	5.3109010(0.2197278);

$n = 500$:

4.99358132(0.09499359);	5.15714169(0.09669357);	5.15428979(0.09666418);
5.11426871(0.09625083);	5.01874406(0.09525706);	4.95975300(0.09463824);

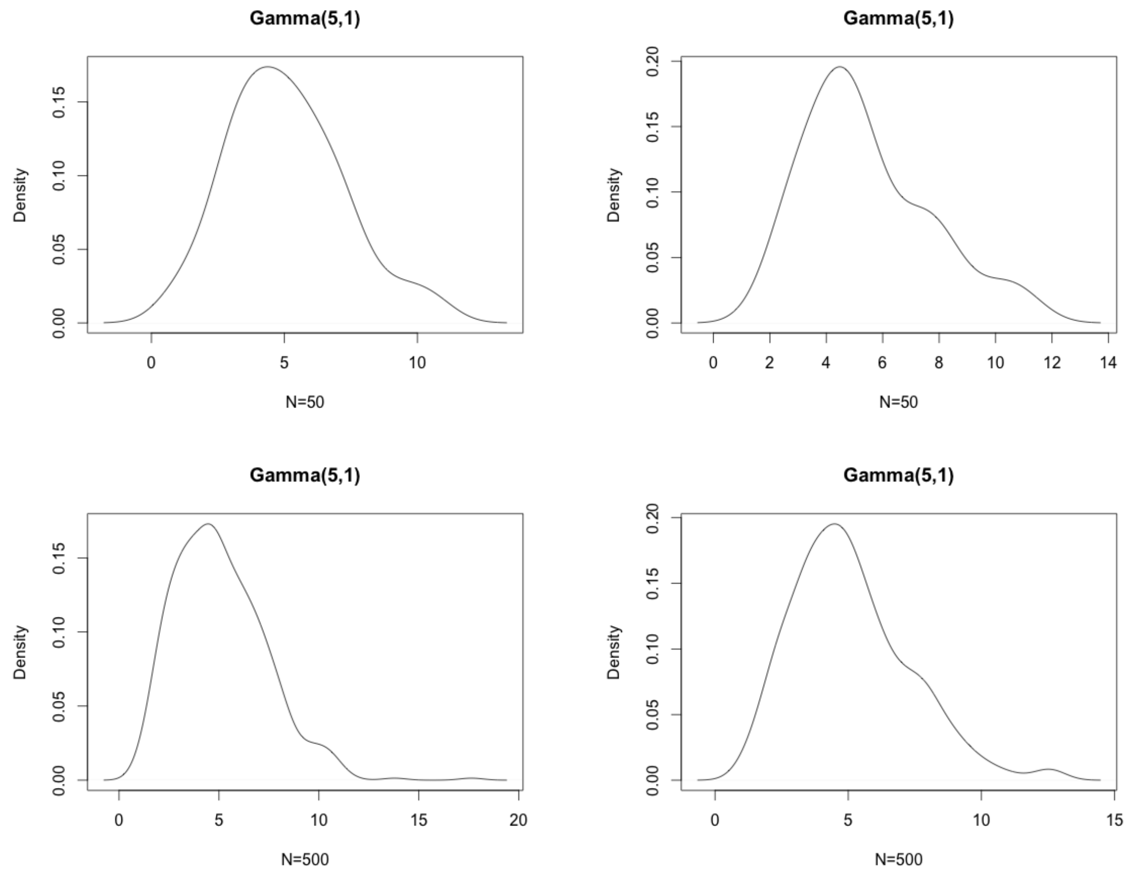
The quantities in brackets are the estimated standard errors due to numerical optimization, which, we see, decrease as the sample size n is increased. Also, the variance of the estimates is less for large sample sizes.



And though we have different algorithms for sample generation for the cases $\alpha_0 < 1$ and $\alpha_0 \geq 1$, the sampling and estimation methods do not differentiate between integer and non-integer α_0 values, and the precision achieved is similar in two cases.

For example, when $n = 50$ and $\alpha = 5.1$:
5.3183637(0.3109812); 4.9230182(0.2980474); 4.8010873(0.2939454);
5.1262415(0.3047635); 5.3411929(0.3117119); 5.0274508(0.3015171);

Examples of generated Samples:



Code for plots (in R language)

```
library(MASS)
n=50                        #number of variates in each sample
alpha=.1*(1:100)            #range of alpha for sample generation
s=20                        #number of samples for each value of alpha
assign("x", c())
assign("y", c())
assign("z", c())
for (a in alpha)
  for (i in 1:s) {
    x <- c(x,(a))
    y <- c(y,fitdistr(rgamma(n,(a)), "gamma", rate=1, method="BFGS"))
  }
for (i in 1:(s*length(alpha))) z[i] = y[[5*i-4]]
plot(
  x,z, type="l", main=expression(paste("MLE of ",alpha)),
  sub=substitute(paste("n = ", n), list(n=n)),
  xlab=expression(alpha[0]), ylab=expression(alpha)
)
```

Code for single estimation (in R language)

```
n=50                        #number of variates in each sample
a=2                         #value of alpha for generation of sample
fitdistr(rgamma(n,a), "gamma", rate=1, method="BFGS")
```

References

[1] Ahrens, J. H. and Dieter, U. (1974). *Computer methods for sampling from gamma, beta, Poisson and binomial distributions.* Computing, 12, 223246

[2] Ahrens, J. H. and Dieter, U. (1982). *Generating gamma variates by a modified rejection technique.* Communications of the ACM, 25, 4754

[3] Ahrens, J. H. and Dieter, U. (1972). *Computer methods for sampling from the exponential and normal distributions.* Communications of the ACM, 15, 873-882

[4] Jorge Nocedal and Stephen J. Wright (2000). *Numerical Optimization*