STAT3*13

Semester 2	Statistical Inference	2011

Week 10 - Computer Exercise

Suppose we wish to fit a gamma distribution with density $f(x;\alpha,\beta) = x^{\alpha-1}e^{-x/\beta}\beta^{-\alpha}/\Gamma(\alpha)$ for $x > 0$ to some data $x_1, x_2, \ldots, x_n$ by maximum likelihood.

The log-likelihood is

$$L(\alpha,\beta) = (\alpha-1)\sum_{i=1}^{n}\log x_i - \frac{1}{\beta}\sum_{i=1}^{n} x_i - n\psi(\alpha) - n\alpha\log\beta\,.$$

where $\psi(\alpha) = \log\Gamma(\alpha)$. Setting partial derivatives equal to zero yields the likelhood equations

$$\begin{aligned}\sum_{i=1}^{n}\log x_i - n\psi'(\alpha) - n\log\beta &= 0\\ \frac{1}{\beta^2}\sum_{i=1}^{n} x_i - \frac{n\alpha}{\beta} &= 0\end{aligned}$$

Writing the solution to these equations (assuming that it exists) by $(\hat\alpha,\hat\beta)$ the second equation gives

$$\hat\beta = \bar x/\hat\alpha \tag{1}$$

and substituting back into the first and rearranging gives

$$\begin{aligned}0 &= \sum_{i=1}^{n}\log x_i - n\psi'(\hat\alpha) - n\log(\bar x/\hat\alpha)\\ &= \sum_{i=1}^{n}\log(x_i/\bar x) - n\psi'(\hat\alpha) + n\log\hat\alpha\,.\end{aligned}$$

Dividing by n gives

$$m + \log\hat\alpha - \psi'(\hat\alpha) = g(\hat\alpha) = 0 \tag{2}$$

where $m = \frac{1}{n}\sum_{i=1}^{n}\log(x_i/\bar x)$. Given a set of actual data, m can be computed and this equation solved using the Newton-Raphson method, which finds a root of an equation $g(\alpha) = 0$ by starting at some initial guess α_0 and then iteratively updating via

$$\alpha_{j+1} = \alpha_j - g(\alpha_j)/g'(\alpha_j)\,.$$

This is feasible using e.g. R since the derivatives of $\psi(\alpha) = \log\Gamma(\alpha)$ are available as $\psi'(\alpha) =$ `digamma`(α) and $\psi''(\alpha) =$ `trigamma`(α). All we need is a starting value, which we might as well take as the method-of-moments estimator for α given by $\bar x/s^2$ where s^2 is the sample variance of the x_i's. Once $\hat\alpha$ is obtained, we use (1) to get $\hat\beta$.

In this exercise we firstly work out how to obtain the mle's for α and β and then perform a simulation to compare their performance to the easier-to-obtain method-of-moment estimators. We also consider a third, related, intermediate estimation method.

1. Generate a sample of size 200 from a gamma distribution with `shape=5` and `scale=2`, calling it `x`.

2. Obtain the method-of-moments estimates, calling them `a.mom` and `b.mom` (recall that $Var(X)/E(X) = \beta$). Also define `m` as $\frac{1}{n}\sum_{i=1}^{n} \log(x_i/\bar{x})$.

3. The code below performs a single Newton-Raphson iteration attempting to solve (2), starting at $\alpha_0 =$ `a.mom`; it then performs another 9 iterations

```
a0=a.mom
print(a0)
g=m+log(a0)-digamma(a0)
gd=(1/a0)-trigamma(a0)
a1=a0-g/gd
print(a1)

a=a1
for(i in 1:9){
g=m+log(a)-digamma(a)
gd=(1/a)-trigamma(a)
a=a-g/gd
print(a)
}
```

   You should see that the biggest change is after just the first iteration, and that the subsequent adjustments are much smaller by comparison.

4. If you are happy that 10 iterations is enough for convergence, set `a.mle` = `a` and compute the mle for β, calling it `b.mle` using (1). Also obtain an estimate for β in a similar way but using `a1` (the one-step estimate of α) instead of `a.mle`; call it `b1`.

5. You are now ready for the simulation part, where we compare the performance of the *three* different estimation methods: method-of-moments, one-step Newton-Raphson and maximum-likelihood:

   - Generate a large random sample from the same gamma distribution of length 200,000, forming it into a 1000-by-200 matrix `M`; each row of `M` is then interpreted as a sample of size 200.
   - Create empty vectors `A.mom`, `B.mom`, `A1`, `B1`, `A.mle` and `B.mle`; e.g. `A.mom=0`.
   - Perform a loop of 1000 iterations where at the `j`-th iteration you compute the method-of-moments, one-step Newton-Raphson and maximum likelihood estimates of each of α and β based on the `j`-th row of `M`, saving the results in the `j`-th elements of `A.mom`, `B.mom`, `A1`, `B1`, `A.mle` and `B.mle` respectively (doing $1+9 = 10$ Newton-Raphson iterations each time should be enough for the mle; if not you can make it more).

6. Once you have obtained your 6 vectors of length 1000, compare `A.mom`, `A1` and `A.mle` by computing the average squared error for each (e.g. `mean((A.mom-5)^2)`); similarly compare `B.mom`, `B1` and `B.mle`. **Comment** on the relative performances of the three methods.