Newton's Method and Scoring

Optimization in multiparameter situation

As before, let $L(\theta)$ be the log likelihood function.

So we want to find θ such that $\nabla L(\theta) = 0$

Similarly to before, take a Taylor series approximation of this giving

$$\nabla L(\theta) = \nabla L(\theta_n) + d^2 L(\theta_n)(\theta - \theta_n)$$

where $d^2 L(\theta)$ is the matrix of second partial derivatives of $L(\theta)$.

Notation: For some reason, Lange defines

$$dL(\theta) = \nabla L(\theta)^T,$$

the gradient as a row vector.

This leads to an updating formula of

$$\theta_{n+1} = \theta_n - (d^2 L(\theta_n))^{-1} dL\left(\theta_n\right)^T$$

which is a direct analogue to the univariate

$$\theta_{n+1} = \theta_n - \frac{L'\left(\theta_n\right)}{L''\left(\theta_n\right)}$$

Example: MLEs for gamma distribution

$$f(x;\lambda,k)=\frac{x^{k-1}}{\lambda^{k}\Gamma(k)}e^{-x/\lambda}$$

$$E[X]=\lambda k,\ \mathrm{Var}(X)=\lambda^{2}k$$

So the log likelihood for a sample of size n is

$$L(\lambda,k)=(k-1)\sum\log x_i-\frac{1}{\lambda}\sum x_i - nk\log\lambda - n\log\Gamma(k)$$

Define $\psi(k)=\frac{d}{dk}\log\Gamma(k)$. This function is sometimes known as the psi function or the digamma function. Its derivative $\psi'(k)$ is often referred to as the trigamma function.

Then the derivatives of the log likelihood are

$$\frac{\partial}{\partial\lambda}L(\lambda,k)=\frac{1}{\lambda^{2}}\sum x_i-\frac{nk}{\lambda}$$

$$\frac{\partial}{\partial k}L(\lambda,k)=\sum\log x_i-n\log\lambda-n\psi(k)$$

$$\frac{\partial^{2}}{\partial\lambda^{2}}L(\lambda,k)=\frac{nk}{\lambda^{2}}-\frac{2}{\lambda^{3}}\sum x_i$$

$$\frac{\partial^2}{\partial k^2} L(\lambda, k) = -n\psi'(k)$$

$$\frac{\partial^2}{\partial \lambda \partial k} L(\lambda, k) = -\frac{n}{\lambda}$$

Thus Newton is easy to implement with

$$dL(\lambda, k)^T = \begin{bmatrix} \frac{1}{\lambda^2}\sum x_i - \frac{nk}{\lambda} \\ \sum \log x_i - n\log\lambda - n\psi(k) \end{bmatrix}$$

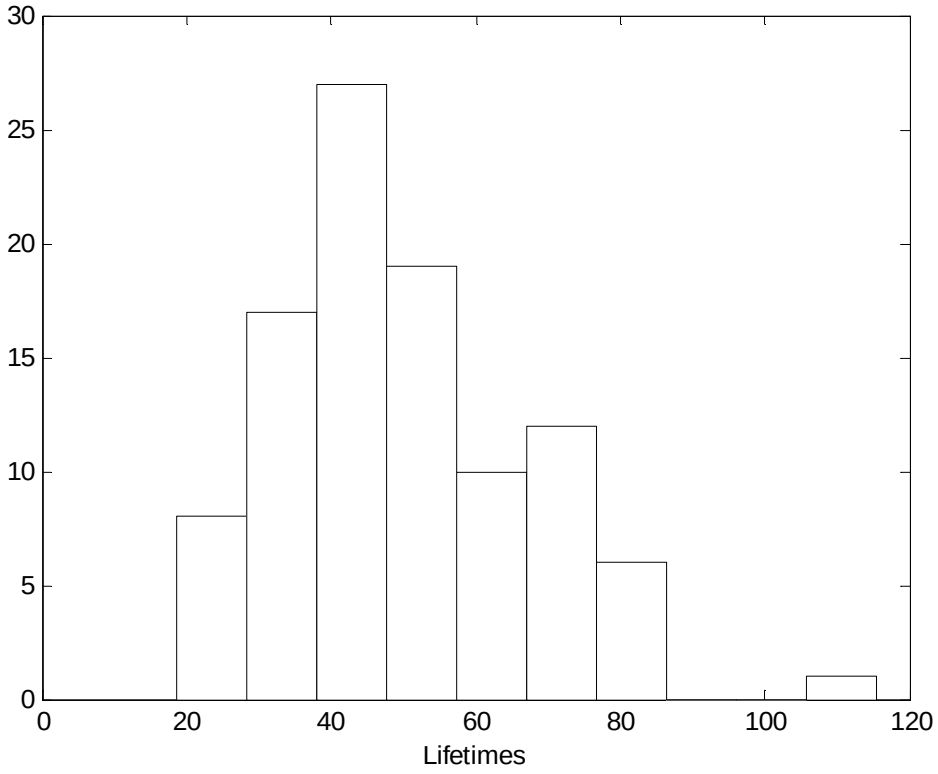
and

$$d^2 L(\lambda, k) = n \begin{bmatrix} \frac{k}{\lambda^2} - \frac{2}{\lambda^3}\sum x_i & -\frac{1}{\lambda} \\ -\frac{1}{\lambda} & -\psi'(k) \end{bmatrix}$$

Implementation notes:

Function	S-Plus/R	Matlab
$\psi(k)$	`digamma(k)`	`psi(k)` or `psi(0,k)`
$\psi'(k)$	`trigamma(k)`	`psi(1,k)`

To exhibit the properties of Newton-Raphson in this case n = 100 observations were generated from a gamma distribution with λ = 5 and k = 10.



$$\bar{x} = 49.8948$$

$$s_x^2 = 293.3081$$

$$s_x = 17.1262$$

For comparison, the method of moments estimators are

$$\lambda_{MOM} = \frac{s_x^2}{\bar{x}} = 5.8785$$

$$k_{MOM} = \frac{\bar{x}^2}{s_x^2} = 8.4876$$

The maximum likelihood estimates are

$$\hat{\lambda} = 5.6831$$

$$\hat{k} = 8.7794$$

and the observed information is

$$I_{obs} = \begin{bmatrix} 27.1825 & 17.5959 \\ 17.5959 & 12.0635 \end{bmatrix}$$

which gives

$$\mathrm{Var}([\lambda\ k]) = I_{obs}^{-1} = \begin{bmatrix} 0.6592 & -0.9615 \\ -0.9615 & 1.4853 \end{bmatrix}$$

The sequence of iterations for this example, based on a convergence criterion of $\|\theta_{n+1} - \theta_n\|_\infty$ is

Iteration	λ_n	k_n
0	5.0000	8.0000
1	5.3097	9.2384
2	5.5421	8.9787
3	5.6705	8.7937
4	5.6829	8.7798
5	5.6831	8.7794
6	5.6831	8.7794

Convergence of multiparameter Newton-Raphson

Like the single parameter case, this procedure has quadratic convergence, so it is usually fast, assuming that you don't make any coding errors.

However accessing convergence during a run is a bit more difficult.

Since there are many parameters, they may be on different scales.

An example is related to question 3 on the first assignment. It involves estimating the lifetime risk for disease ($0 < p < 1$) and the mean and variance of the age of onset distribution for those at risk for the disease.

So using a convergence criterion of

$$\left|\theta_{n+1}^{i} - \theta_{n}^{i}\right| < 0.01$$

might be fine when talking about a mean age around 50, but isn't as good when talking about a lifetime risk around 0.1

So often it makes more sense to use something like

$$\max \frac{\left|\theta_{n+1}^{i} - \theta_{n}^{i}\right|}{\left|\theta_{n}^{i}\right|} < TOL$$

though possible in combination with one based on $\left|\theta_{n+1}^{i} - \theta_{n}^{i}\right|$ as well.

There are two potential problems with Newton's method.

First, it may be computationally expensive to calculate the observed information $-d^2L(\theta)$.

Second, when θ_n is far from $\hat{\theta}$, Newton may head for a minimum instead of a maximum.

Newton's method is not an ascent algorithm, e.g.

$$L(\theta_{n+1}) > L(\theta_n)$$

does NOT have to hold.

There are procedures that are ascent algorithms, such as EM.

This problem usually occurs when $-d^2L(\theta_n)$ is not positive definite.

One solution is to replace $-d^2L(\theta_n)$ with a positive definite approximation A_n.

With this change, the proposed increment $\Delta\theta_n = A_n^{-1}dL(\theta_n)^T$, possibly contracted, forces an increase in $L(\theta)$.

This can be justified by

$$L\left(\theta_n + \alpha\Delta\theta_n\right) - L(\theta_n) = dL(\theta_n)\alpha\Delta\theta_n + o(\alpha)$$
$$= \alpha dL(\theta_n)A_n^{-1}dL(\theta_n)^T + o(\alpha)$$

where the error ratio $o(\alpha)/\alpha$ tends to 0 as α goes to 0.

How to chose α?

Common approach is half step back tracking.

Try $\alpha = 1$. If leads to increase in likelihood, stop.

If not, try $\alpha = ½$, then ¼, etc until you get an increase in $L(\theta)$.

How to choose A_n?

Steepest ascent: $A_n = I$.

Scoring:

Replace observed information with expected information $J(\theta) = E\left[-d^2L(\theta)\right]$.

Since $J(\theta) = E\left[-d^2L(\theta)\right] = Var\left(dL(\theta)^T\right)$ so $J(\theta)$ is positive definite

The update equation for scoring is

$$\theta_{n+1} = \theta_n + J(\theta_n)^{-1} dL(\theta_n)^T$$

(remember the sign switch)

One thing to remember with the scoring algorithm is that you use the same gradient function, which depends on your data. Just replace the information matrix.

Since we are not using the optimal direction early on, the convergence is slightly slower, but is still quadratic. Often this means it might take a couple of iterations longer to converge than basic Newton (as we will see in a minute)

For the gamma example

$$J(\lambda, k) = n \begin{bmatrix} \frac{k}{\lambda^2} & \frac{1}{\lambda} \\ \frac{1}{\lambda} & \psi'(k) \end{bmatrix}$$

which is derived by replacing $\sum x_i$ with its expectation $n\lambda k$.

For the same data and convergence criterion, the following sequence was observed

Iteration	λ_n	k_n
0	5.0000	8.0000
1	7.8163	5.4728
2	7.2107	6.8075
3	5.7308	8.3167
4	5.7952	8.6129
5	5.6831	8.7762
6	5.6832	8.7794
7	5.6831	8.7794
8	5.6831	8.7794

Note that these are the same estimates, up to the number of digits produced.

$$J\left(\hat{\theta}\right) = \begin{bmatrix} 27.1825 & 17.5959 \\ 17.5959 & 12.0635 \end{bmatrix} = I_{obs}\left(\hat{\theta}\right)$$

which gives

$$\mathrm{Var}\left([\lambda\ k]\right) = I_{obs}^{-1} = \begin{bmatrix} 0.6592 & -0.9615 \\ -0.9615 & 1.4853 \end{bmatrix}$$

Normally $J\left(\hat{\theta}\right) \neq I_{obs}\left(\hat{\theta}\right)$, though usually they are close since asymptotically they are the same.

There is something special with this gamma example which leads to the equality.