Statistical Programming
Week 6

THE UNIVERSITY *of* EDINBURGH

School of Mathematics and Maxwell Institute

Directional derivative

REM: tangent plane approximation used to motivate steepest descent

$$f(\boldsymbol{x} + \boldsymbol{\Delta}) \approx f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \boldsymbol{\Delta}$$

The quantity $\nabla f(\boldsymbol{x})^T \boldsymbol{\Delta}$ is called the directional derivative of f along the vector $\boldsymbol{\Delta}$ and is equal to

$$\nabla f(\boldsymbol{x})^T \boldsymbol{\Delta} = \lim_{h \to 0} \frac{f(\boldsymbol{x} + h\boldsymbol{\Delta}) - f(\boldsymbol{x})}{h}$$

Meaning: instantaneous rate of change of the function f, when moving past $\boldsymbol{x}$ at velocity $\boldsymbol{\Delta}$.

Steepest descent

Taylor expansion, $t \in (0, 1)$

$$f(\boldsymbol{x} + \boldsymbol{\Delta}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \boldsymbol{\Delta} + \frac{1}{2} \boldsymbol{\Delta}^T \nabla^2 f(\boldsymbol{x} + t\boldsymbol{\Delta})\, \boldsymbol{\Delta}$$

- As steepest descent approaches the minimum, $\boldsymbol{x}^*$, the directional derivative term retained in the Taylor expansion of f becomes negligible relative to the neglected second derivative term since $\nabla f(\boldsymbol{x}^*) = 0$
- This largely explains the method's poor performance.

Quadratic form approximation

Consider instead the quadratic form approximation

$$\begin{aligned} f(\boldsymbol{x}+\boldsymbol{\Delta}) &= f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \boldsymbol{\Delta} + \frac{1}{2}\boldsymbol{\Delta}^T \nabla^2 f(\boldsymbol{x}+t\boldsymbol{\Delta})\,\boldsymbol{\Delta} \\ &= f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \boldsymbol{\Delta} + \frac{1}{2}\boldsymbol{\Delta}^T \nabla^2 f(\boldsymbol{x})\,\boldsymbol{\Delta} + O(\|\boldsymbol{\Delta}\|^3) \quad (1) \end{aligned}$$

- This is of the form $c + b^T\boldsymbol{\Delta} + \frac{1}{2}\boldsymbol{\Delta}^T H_{\boldsymbol{x}}\,\boldsymbol{\Delta}$, vector b, scalar c.
- $H_{\boldsymbol{x}}$ is square symmetric matrix and has spectral decomposition

$$H_{\boldsymbol{x}} = \Gamma_{\boldsymbol{x}}\,\Lambda_{\boldsymbol{x}}\,\Gamma_{\boldsymbol{x}}^T$$

where $\Lambda_{\boldsymbol{x}}$ is a diagonal matrix of eigenvalues, ordered in decreasing value and $\Gamma_{\boldsymbol{x}}$ is $m \times m$ orthogonal matrix ($\Gamma_{\boldsymbol{x}}^T\Gamma_{\boldsymbol{x}} = I$) of the corresponding eigenvectors.

Quadratic form approximation

Unique quadratic form that passes through $\boldsymbol{x}$ and has the same first and second partial derivatives with f at $\boldsymbol{x}$.

Differentiating the r.h.s. of (1) with respect to $\boldsymbol{\Delta}$ and setting the result to $\boldsymbol{0}$ we have

$$\nabla f(\boldsymbol{x}) + \nabla^2 f(\boldsymbol{x})\,\boldsymbol{\Delta} = 0$$

as the equation to solve for the step $\boldsymbol{\Delta}$. Assuming $\nabla^2 f(\boldsymbol{x})$ is positive definite, we obtain *Newton's direction*

$$\boldsymbol{\Delta} = -\nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x})$$

$$= -\Gamma_{\boldsymbol{x}} \Lambda_{\boldsymbol{x}}^{-1} \left[\Gamma_{\boldsymbol{x}}^T \nabla f(\boldsymbol{x}) \right]$$

Newton's direction

THE UNIVERSITY *of* EDINBURGH

The instantaneous rate of change of f when moving past $\boldsymbol{x}$ at velocity $\boldsymbol{\Delta} = -\nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x})$ is

$$\nabla f(\boldsymbol{x})^T \boldsymbol{\Delta} = -\nabla f(\boldsymbol{x})^T \nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x}) < 0, \tag{2}$$

hence $\boldsymbol{\Delta}$ is a descent direction.

- Newton's direction is reliable when the difference between the true function $f(\boldsymbol{x} + \boldsymbol{\Delta})$ and its quadratic model approximation is not too large.
- Unlike the steepest descent, there is a "natural" step length of 1 associated with Newton's direction. Usually we accept the step, unless it fails to lead to sufficient decrease in f. In the latter case, the step length is repeatedly halved until sufficient decrease is achieved.
- When $\nabla^2 f(\boldsymbol{x})$ is not positive definite, Newton's direction may not even be defined, since $\nabla^2 f(\boldsymbol{x})^{-1}$ may not even exist.

Practical issues

Similarly to equation (2), we see that any step of the form $\boldsymbol{\Delta} = -B_{\boldsymbol{x}}^{-1}\nabla f(\boldsymbol{x})$, with $B_{\boldsymbol{x}}$ positive definite, implies $\boldsymbol{\Delta}$ is a descent direction.

- This observation gives us the freedom to modify $\nabla^2 f(\boldsymbol{x})$ to make it positive definite and still be guaranteed a descent direction.
- One approach is to take the symmetric eigen-decomposition $\Gamma_{\boldsymbol{x}}\Lambda_{\boldsymbol{x}}\Gamma_{\boldsymbol{x}}^T$, reverse the sign of any negative eigenvalues and replace any zero eigenvalues with a small positive number, e.g., for $\lambda_1, \ldots, \lambda_m \neq 0$ tweak

$$\Lambda_{\boldsymbol{x}} = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_m \end{bmatrix} \quad \text{to} \quad \Lambda'_{\boldsymbol{x}} = \begin{bmatrix} |\lambda_1| & 0 & \ldots & 0 \\ 0 & |\lambda_2| & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & |\lambda_m| \end{bmatrix}$$

Practical issues

- Computational cheaper alternatives add a multiple of the identity matrix to $\nabla^2 f(\boldsymbol{x})$, sufficiently large to make the result positive definite, i.e.,

$$B_{\boldsymbol{x}} = \nabla^2 f(\boldsymbol{x}) + c\boldsymbol{I}, \qquad c > 0$$

where

$$\boldsymbol{I} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Practical Newton algorithm

THE UNIVERSITY *of* EDINBURGH

1. Set $k = 0$ and a guesstimate $\boldsymbol{x}^0$
2. Evaluate $f(\boldsymbol{x}^{(k)}), \nabla f(\boldsymbol{x}^{(k)}), \nabla^2 f(\boldsymbol{x}^{(k)})$, implying a quadratic model approximation to f.
3. Test whether $\boldsymbol{x}^{(k)}$ is a minimum and terminate if it is.
4. If $\nabla^2 f(\boldsymbol{x}^{(k)})$ is not positive definite, perturb it so that it is (modifying the quadratic model)
5. The search direction is defined by the solution

$$\boldsymbol{\Delta} = -\nabla^2 f(\boldsymbol{x}^{(k)})^{-1} \nabla f(\boldsymbol{x}^{(k)})$$

6. If $f(\boldsymbol{x}^{(k)} + \boldsymbol{\Delta})$ is not sufficiently lower than $f(\boldsymbol{x}^{(k)})$, repeatedly halve $\boldsymbol{\Delta}$ until it is.
7. Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{\Delta}$, increment k by one, return to step 2.

Comments

- Methods that use the Newton direction have a fast rate of local convergence, typically quadratic.
- After a neighbourhood of the solution is reached, convergence to high accuracy often occurs in just a few iterations.
- The main drawback of the Newton direction is the need for the Hessian $\nabla^2 f(\boldsymbol{x})$. This can be tedious to obtain, especially when the dimension of $\boldsymbol{x}$ is large.
- Finite-difference techniques may be useful in avoiding the need to calculate second derivatives by hand.

[Demo]

Quasi-Newton methods

Is it possible produce methods that only require first derivative information, but with efficiency rivalling Newton's method, rather than steepest descent?

- One approach is to use finite-difference but this is usually more costly than exact evaluation of Hessian.

- Quasi-Newton methods provide an attractive alternative to Newton's method in that they do not require computation of the Hessian and yet still attain a superlinear rate of convergence

- In place of $\nabla^2 f(\boldsymbol{x})$, they use an approximation $B_{\boldsymbol{x}}$, which is updated after each step to take into account of the additional knowledge gained in every step. The updates make use of the fact that changes in the gradient provide information about the second derivative of f along the search direction

Quasi-Newton methods

REM fundamental theorem of calculus, if $f : \mathcal{R} \rightarrow \mathcal{R}$ is a continuous function then

$$f(x) = f(a) + \int_a^x f'(t)\,\mathrm{d}t$$

The multivariate generalisation recasted to the gradient of f is

$$\nabla f(\boldsymbol{x} + \boldsymbol{\Delta}) = \nabla f(\boldsymbol{x}) + \int_0^1 \nabla^2 f(\boldsymbol{x} + t\boldsymbol{\Delta})\,\boldsymbol{\Delta}\,\mathrm{d}t$$

Adding and subtracting $\nabla^2 f(\boldsymbol{x})\,\boldsymbol{\Delta}$ to the r.h.s, we get

$$\begin{aligned}\nabla f(\boldsymbol{x} + \boldsymbol{\Delta}) = \nabla f(\boldsymbol{x}) + \nabla^2 f(\boldsymbol{x})\,\boldsymbol{\Delta}& \\ + \int_0^1 \left[\nabla^2 f(\boldsymbol{x} + t\boldsymbol{\Delta}) - \nabla^2 f(\boldsymbol{x})\right]\,\boldsymbol{\Delta}\,\mathrm{d}t&\end{aligned}$$

Quasi-Newton methods

THE UNIVERSITY
of EDINBURGH

Setting $\boldsymbol{x} = \boldsymbol{x}^{(k)}$ and $\boldsymbol{\Delta} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ we obtain that for small $\|\boldsymbol{\Delta}\|$

$$\nabla f(\boldsymbol{x}^{(k+1)}) - \nabla f(\boldsymbol{x}^{(k)}) \approx \nabla^2 f(\boldsymbol{x}^{(k)})(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) \qquad (3)$$

Let $H^{(k+1)}$ be the approximate positive definite Hessian at the $(k+1)$-th step

- The basic requirement of a quasi Newton method is that the approximation should exactly match $\nabla f(\boldsymbol{x}^{(k)})$, i.e., it should get the gradient vector at the previous point, $\boldsymbol{x}^{(k)}$, exactly right, mathematically

$$\nabla f(\boldsymbol{x}^{(k+1)}) - \nabla f(\boldsymbol{x}^{(k)}) = H^{(k+1)}(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) \qquad (4)$$

so that the new Hessian approximation $H^{(k+1)}$ mimics the property (3)

Quasi-Newton methods

- Let $\boldsymbol{s}_k = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ and $\boldsymbol{y}_k = \nabla f(\boldsymbol{x}^{(k+1)}) - \nabla f(\boldsymbol{x}^{(k)})$

- Then expression (4) is equivalent to

$$H^{(k+1)}\boldsymbol{s}_k = \boldsymbol{y}_k \qquad \textit{(secant equation)}$$

- The secant equation will only be feasible for positive definite $H^{(k+1)}$ under certain conditions on $\boldsymbol{s}_k$ and $\boldsymbol{y}_k$, but we can always arrange for these to be met by appropriate step length selection.
  (Wolfe conditions, see Nocedal & Wright, 2006, §3.1[1]).

- The secant equation alone does not define a unique $B^{(k+1)} = (H^{(k+1)})^{-1}$, and some extra conditions are needed.

[1] Nocedal and Wright (2006). Numerical Optimization, Springer, 2nd edition

Conditions and solution

- $B^{(k+1)}$ satisfies the secant equation
- The difference between $B^{(k)}$ and $B^{(k+1)}$ has low rank, i.e., $B^{(k+1)}$ is as close to $B^{(k)}$ (measured by some appropriate matrix norm)
- $B^{(k+1)}$ is positive definite

The unique solution to this problem is the BFGS update[2]

$$B^{(k+1)} = (\boldsymbol{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T) B^{(k)} (\boldsymbol{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T) + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^T$$

where $\rho_k^{-1} = \boldsymbol{s}_k^T \boldsymbol{y}_k$

[2] BFGS is named after Broyden, Fletcher, Goldfarb and Shanno all of whom independently discovered and published it, around 1970.

Quasi-Newton methods

- The BFGS method then works exactly like Newton's method, but with $B^{(k)}$ in place of the inverse $\nabla^2 f(\boldsymbol{x}^{(k)})$, and without the need for second derivative evaluation or for perturbing the Hessian to achieve positive definiteness

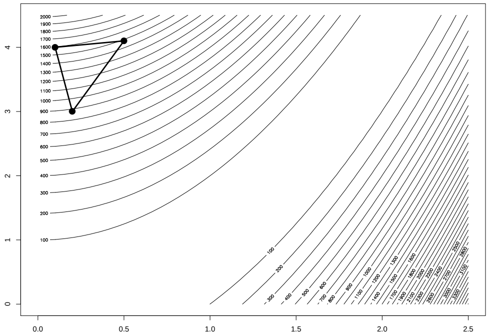
- A finite difference approximation to the Hessian is often used to start the method

[Demo]

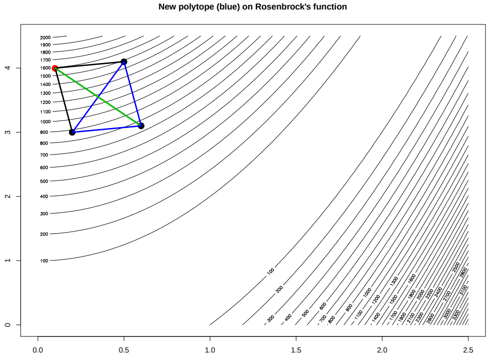
Nelder-Mead polytope

What if even gradient evaluation is too taxing, or if our objective is not smooth enough for Taylor approximation to be valid? The Nelder-Mead polytope provides a successful answer.

- Let m be the dimension of $\boldsymbol{x}$. At each stage of the method we maintain $m+1$ distinct $\boldsymbol{x}$ vector, defining a polytope in $\mathcal{R}^m$ (e.g., for a two dimensional $\boldsymbol{x}$, the polytope is a triangle)


Starting polytope on Rosenbrock's function

Nelder-Mead polytope


New polytope (blue) on Rosenbrock's function

Nelder-Mead polytope

1. The search direction is defined as the vector from the worst point (vertex of polytope with highest value) through the average of of the remaining m points
2. The initial step length is set to twice the distance from the worst point to the centroid of the others. If it succeeds (meaning that the new point is no longer the worst point), then a step length of 1.5 times is tried, and the better of the two is accepted
3. If the previous step did not find a successful new point then step lengths of half and one and a half times the distance from the worst point to the centroid are tried
4. If the last two steps failed to locate a successful then the polytope is reduced in size, by linear rescaling towards the current best point (which remains fixed)

[Demo]

Nelder-Mead polytope

- Variations are possible, in particular with regard to step length and shrinkage factors.
- Slower rate of convergence than Newton type algorithms
- Nelder-Mead is good if the answer does not need to be accurate, and derivatives are difficult to compute.
- Not recommended for general purpose modelling software.

Other methods

THE UNIVERSITY
of EDINBURGH

- In likelihood maximisation contexts, the method of *scoring* is used. This replaces the Hessian in Newton's method with the expected Hessian.

- Conjugate gradient method is another way of getting a Newton type method when only first derivatives are available. It is applicable when the objective is somehow related to a sum of squares of differences between data and fitted values. This is closely related to the algorithm for fitting generalised linear models (GLMs).

Other methods

THE UNIVERSITY
of EDINBURGH

- Simulated annealing is a method for optimising difficult objectives, such as those arising in discrete optimisation problems, or with very spiky objective functions. The basic idea is to propose random changes in $\boldsymbol{x}$, always accepting a change which reduces $f(\boldsymbol{x})$, but also accepting moves which increase the objective, with a probability which decreases with the size of increase, and also decreases as the optimisation progresses.

- The Expectation-Maximisation algorithm is a useful approach to likelihood maximisation when there are missing data or random effects that are difficult to integrate out.

Optim function

R's inbuilt function `optim` performs general-purpose optimization based on Nelder-Mead, quasi-Newton and conjugate-gradient algorithms.

```
optim(par, fn, gr = NULL, ...,
      method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B",
                 "SANN", "Brent"),
      lower = -Inf, upper = Inf, control = list(),
              hessian = FALSE)
```

To read more on optimization

- Nocedal, J. and Wright, S. J. (2006). Numerical Optimization. Springer.
- Givens, H. and Hoeting, J. A. (2012). Computational Statistics. Wiley.