Statistical Programming
Week 5

THE UNIVERSITY *of* EDINBURGH

School of Mathematics and Maxwell Institute

Optimization

THE UNIVERSITY of EDINBURGH

Many problems in statistics can be characterised as solving

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

where $f : \mathcal{R}^m \to \mathcal{R}$ is a real scalar valued function of vector $\boldsymbol{x} \in \mathcal{R}^m$, known as the objective function. The function f can represent

- negative log-likelihood
- negative posterior distribution
- dissimilarity measure
- distance, energy, cost, etc.

Local vs global optimisation

THE UNIVERSITY *of* EDINBURGH

- Minimization methods generally operate by seeking a local minimum of f. That is they seek a point $\boldsymbol{x}^*$, such that

$$f(\boldsymbol{x}^* + \boldsymbol{\Delta}) \geq f(\boldsymbol{x}^*)$$

for any sufficiently small perturbation $\boldsymbol{\Delta}$.

- Unless f is a strictly convex function, it is not possible to guarantee that such a $\boldsymbol{x}^*$ is a global minimum. That is, it is not possible to guarantee that

$$f(\boldsymbol{x}^* + \boldsymbol{\Delta}) \geq f(\boldsymbol{x}^*)$$

for any arbitrary $\boldsymbol{\Delta}$.

Optimization algorithms

THE UNIVERSITY of EDINBURGH

- Beginning at $\boldsymbol{x}^{(0)}$ optimization algorithms generate a sequence of iterates $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ that terminate when either no more progress can be made or when it seems that a solution point has been approximated with sufficient accuracy.

- In deciding how to move from one iterate $\boldsymbol{x}^{(k)}$ to the next, the algorithms use information about the function f at $\boldsymbol{x}^{(k)}$, and possibly also information from earlier iterates $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(k-1)}$

Optimization methods are myopic

THE UNIVERSITY *of* EDINBURGH

- Optimization methods are short-sighted. At any stage of operation all that optimization methods "know" about a function are a few properties of the function at the *current best* $\boldsymbol{x}$
- It is on the basis of such information that a method tries to find an improved best $\boldsymbol{x}$. The methods have no overview of the function being optimized.

Search direction and Trust region algorithms

THE UNIVERSITY of EDINBURGH

Algorithms start from the initial guesstimate $\boldsymbol{x}^{(0)}$. Set $k = 0$, then:

1. Evaluate $f(\boldsymbol{x}^{(k)})$, and possibly the first and second derivatives of f with respect to the elements of $\boldsymbol{x}$, at $\boldsymbol{x}^{(k)}$
2. Either
   1. use the information from step 1 (and possibly previous executions of step 1) to find a *search direction*, $\boldsymbol{\Delta}$, such that for some $\alpha > 0$, $f(\boldsymbol{x}^{(k)} + \alpha\boldsymbol{\Delta})$ will provide sufficient decrease relative to $f(\boldsymbol{x}^{(k)})$. Search for such an α and set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha\boldsymbol{\Delta}$.
   2. use the information from step 1 (and possibly previous executions of step 1) to construct a local model of f, and find $\boldsymbol{x}$ minimizing this model within a defined region around $\boldsymbol{x}^{(k)}$. If this value of $\boldsymbol{x}$ leads to a sufficient decrease in f, accept it as $\boldsymbol{x}^{(k+1)}$, otherwise shrink the search region until it does.
3. Test whether a minimum has yet been reached.

Inspecting objective functions

THE UNIVERSITY of EDINBURGH

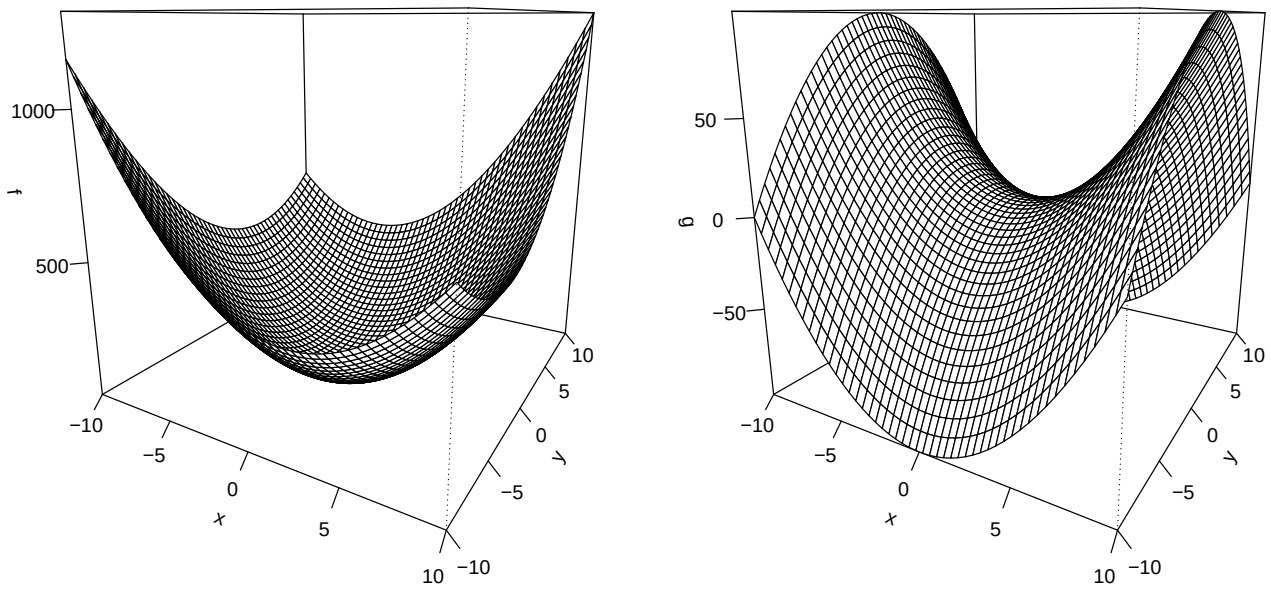

Figure: Left: $f(x, y) = 5x^2 + 4y^2 + 3xy + 7x + 20$.
Right: $g(x, y) = x^2 - y^2$

Contour plots

THE UNIVERSITY of EDINBURGH

A contour plot shows all points where $f(x, y) =$ fixed constant chosen at regular intervals, e.g.,

$$\{(x, y) : f(x, y) = -2\}$$
$$\{(x, y) : f(x, y) = -1\}$$
$$\{(x, y) : f(x, y) = 0\}$$
$$\{(x, y) : f(x, y) = 1\}$$
$$\{(x, y) : f(x, y) = 2\}$$

Contour plots

THE UNIVERSITY of EDINBURGH

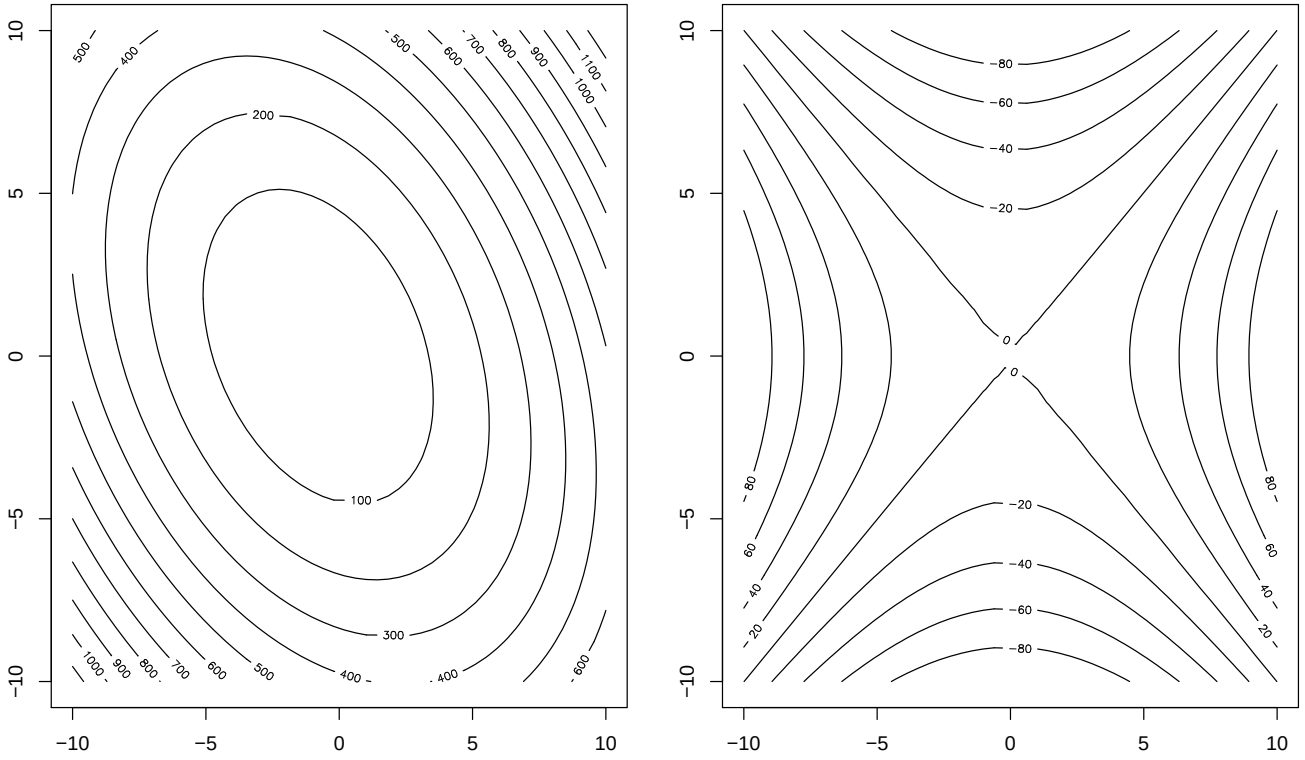

Figure: Left: $f(x, y) = 5x^2 + 4y^2 + 3xy + 7x + 20$.
Right: $g(x, y) = x^2 - y^2$

Perspective and Contour plots

THE UNIVERSITY of EDINBURGH

```
x <- seq(-10,10,len=50)
y <- seq(-10,10,len=50)
f <- matrix(nrow=length(x),ncol=length(y))
g <- matrix(nrow=length(x),ncol=length(y))
for(i in 1:length(x))
for(j in 1:length(y)){
f[i,j]      <- 5*x[i]^2+4*y[j]^2+3*x[i]*y[j]+7*x[i]+20
g[i,j]      <- x[i]^2-y[j]^2
}

par(mfrow=c(1,2))
res <- persp(x,y,f,theta=30,las=1,ticktype="detailed")
res <- persp(x,y,g,theta=30,las=1,ticktype="detailed")

par(mfrow=c(1,2))
contour(x,y,f); contour(x,y,g)
```

Partial derivatives

THE UNIVERSITY of EDINBURGH

Let f be a real scalar valued function of vector (x, y).

- f_x : Rate with which f changes as we vary x and keep $y = y_0$ constant.

$$f_x(x_0, y_0) = \frac{\partial f}{\partial x}(x_0, y_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x, y_0) - f(x_0, y_0)}{\Delta x}$$

- f_y : Rate with which f changes as we vary y and keep $x = x_0$ constant.

$$f_y(x_0, y_0) = \frac{\partial f}{\partial y}(x_0, y_0) = \lim_{\Delta y \to 0} \frac{f(x_0, y_0 + \Delta y) - f(x_0, y_0)}{\Delta y}$$

Multivariate Taylor's theorem

THE UNIVERSITY of EDINBURGH

Suppose that $f : \mathcal{R}^m \to \mathcal{R}$ is twice continuously differentiable function of $\boldsymbol{x}$ and that $\boldsymbol{\Delta}$ is of the same dimension as $\boldsymbol{x}$. Then

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

for some $t \in (0, 1)$, where

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \end{bmatrix} \quad \text{and} \quad \nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

are the *gradient vector* and *Hessian matrix*, respectively.

Conditions for minimum: From Taylor's theorem, it can be established that the conditions for $f(\boldsymbol{x}^*)$ to be a minimum are that

$$\nabla f(\boldsymbol{x}^*) = 0 \quad \text{and} \quad \nabla^2 f(\boldsymbol{x}^*) \text{ is nonnegative definite.}$$

Steepest descent

THE UNIVERSITY OF EDINBURGH

Suppose we are located at an arbitrary point $\boldsymbol{x}$ in $\mathcal{R}^m$. We ask, which direction leads to the most rapid decrease in f for sufficiently small step, and use this as the *search direction*. If $\boldsymbol{\Delta}$ is small enough, then

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

This is the tangent plane approximation of f at $\boldsymbol{x}$.

- For $\boldsymbol{\Delta}$ of fixed length, the greatest decrease will be achieved by minimising the inner product

  $$\nabla f(\boldsymbol{x})^T \boldsymbol{\Delta} = \|\nabla f(\boldsymbol{x})\| \|\boldsymbol{\Delta}\| cos(\theta)$$

  that is, by choosing $\boldsymbol{\Delta} \propto -\nabla f(\boldsymbol{x})$. Here $\|\cdot\|$ denotes Euclidean distance.

- Set

  $$\boldsymbol{\Delta} = -\nabla f(\boldsymbol{x}) / \|\nabla f(\boldsymbol{x})\|$$

Steepest descent

THE UNIVERSITY of EDINBURGH

Given some direction, we need a method for choosing how far to move from $\boldsymbol{x}$ along this direction.

Trade-off: Need to make reasonable progress, but do not want to spend too much time choosing the exactly optimal distance to move, since the point will only be abandoned at the next iteration.

Best to try and choose the step length $\alpha > 0$ so that $f(\boldsymbol{x} + \alpha\boldsymbol{\Delta})$ is sufficiently much lower than $f(\boldsymbol{x})$, and also that the magnitude of the gradient of f in the $\boldsymbol{\Delta}$ direction is sufficiently reduced by the step. At the moment we assume we have such a method.

Steepest descent

THE UNIVERSITY of EDINBURGH

Given that we initially started at the point $\boldsymbol{x}^{(0)}$, the gradient descent algorithm implements the following iterative procedure for finding the minimum of a function f.

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha \frac{\nabla f(\boldsymbol{x}^{(k)})}{\|\nabla f(\boldsymbol{x}^{(k)})\|}$$

Rosenbrock's function

THE UNIVERSITY of EDINBURGH

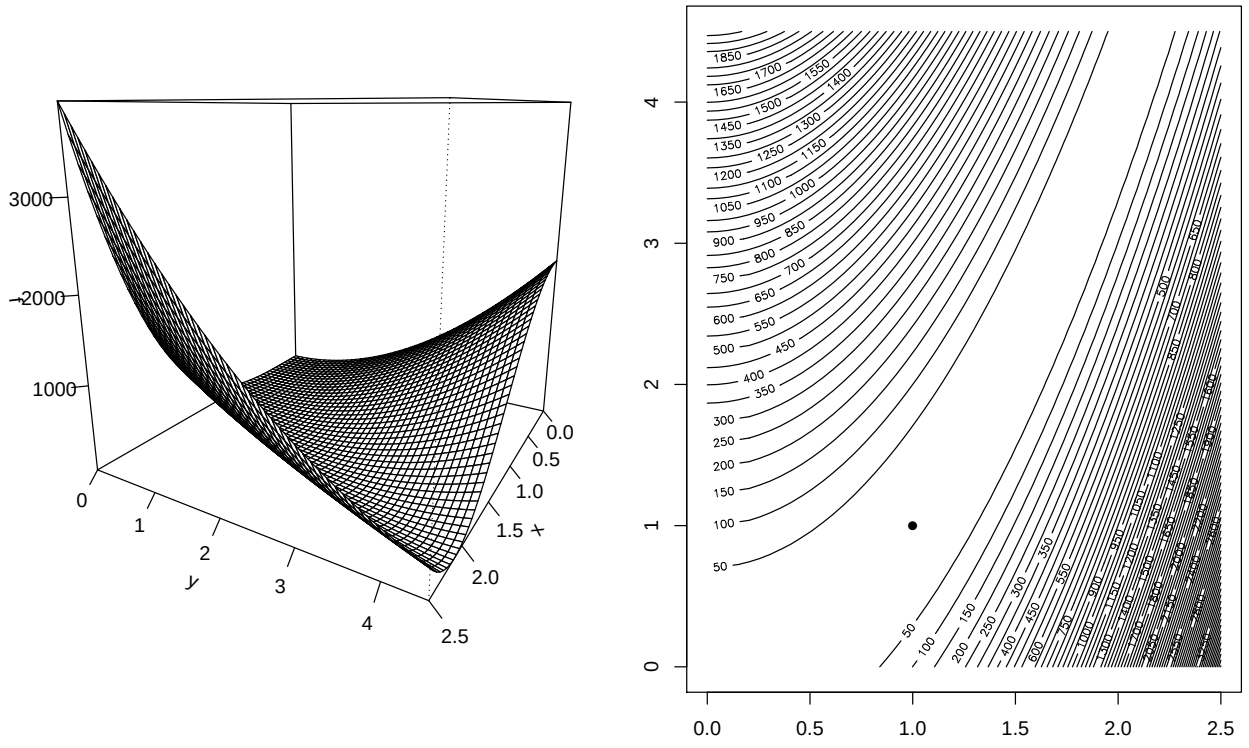

Figure: $f(x, y) = (a - x)^2 + b(y - x^2)^2$ for $a = 1$ and $b = 100$. The black dot on the contour plot shows the minimum of the function.

[Demo]

Drawbacks

THE UNIVERSITY of EDINBURGH

- Scaling: linearly rescaling the z-axis changes all steepest descent directions
- Slow rate of convergence (zig-zag behaviour of algorithm if curvature in different directions is very different)