Week 9: aim to cover

- Nonlinear least squares, steepest descent, Gauss-Newton (Lecture 17)

- QR factorization; Nonlinear model fitting (Lab 9)

- Levenberg-Marquardt algorithm (as trust region method) (Lecture 18)

Fitting nonlinear models

Recall, we could fit linear models

$$y \sim \Sigma_k x_i \phi_i(t)$$

to data $\{(T_i, Y_i)\}, i = 1 \ldots m$, by finding the least squares solution to an overdetermined linear system

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{x}$ are the parameters or coefficients in the model, $\mathbf{A}_{ij} = \phi_j(T_i)$ and $\mathbf{b}_i = Y_i$.

If the model is nonlinear in (some of) the parameters, we instead have a nonlinear least squares problem.

Example

fitting exponentials

The cost function

In nonlinear models the parameters x_j appear in the residual of the model at each data point

$$r_i(\mathbf{x}) = Y_i - \sum_k \phi_k(T_i, \mathbf{x})$$

We write the residual as

$$\mathbf{R}(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}), \ldots r_m(\mathbf{x}))^T \in \mathbb{R}^m$$

and $\mathbf{x} \in \mathbb{R}^n$ i.e. n parameters and m data points as before.
As before we aim to minimize the 2-norm of the residual, so our cost function to minimize is

$$C(\mathbf{x}) = \frac{1}{2}||\mathbf{R}(\mathbf{x})||_2^2 = \frac{1}{2}\mathbf{R}(\mathbf{x})^T\mathbf{R}(\mathbf{x})$$

The factor of $1/2$ is to simplify later equations.

Nonlinear least squares

We could aim to minimize $C(\mathbf{x})$ using any of the standard methods of nonlinear optimization [see MAST30013] but we instead describe methods that take advantage of the special form of $C(\mathbf{x})$ in the least squares case.

In some sense, nonlinear least squares problems are intermediate between nonlinear systems of equations and nonlinear optimization. It won't be surprising that we need iterative methods. In particular, we solve a sequence of linear least squares problems.

As for root-finding methods, there are optimization methods that use only function information and methods that use gradient information. Newton's method, which uses second derivative information, is regarded as too expensive for real problems. We describe 3 methods that use gradient information.

Steepest descent

The simplest method using gradient information uses the fact that the direction of steepest descent of a function C of several variables is

$$\mathbf{d} = -\nabla C(\mathbf{x}) \in \mathbb{R}^n$$

For the NLSQ case, we have

$$\nabla C = \mathbf{J}(\mathbf{x})^T \mathbf{R}(\mathbf{x})$$

where

$$J_{ij} = \frac{\partial r_i}{\partial x_j}$$

is the Jacobian, an $m \times n$ matrix.
Since a necessary condition for a minimum is $\nabla C(\mathbf{x}) = 0$ we have the condition

$$\mathbf{J}(\mathbf{x}^*)^T \mathbf{R}(\mathbf{x}^*) = \mathbf{0}$$

at a minimum $\mathbf{x}^*$.

Line search

This suggests an algorithm


```
start from an initial guess x_0
until termination criterion is met
   compute descent direction d_k at current position x_c
   x_+ = x_c + a_k d_k
     [step a distance a_k along direction d_k to get trial positio
   test trial position for sufficient descent etc.
   if OK accept x_+ and current a_k; x_c := x_+
   else adjust a_k and repeat attempted step [backtrack]
end
```

Global convergence

Under suitable (reasonable) assumptions, this algorithm is globally convergent to a local minimizer. That does not mean that it finds the global minimum — that is much harder to achieve.
However this algorithm sometimes struggles in the latter stages of the iteration

Example

the Rosenbrock function

and is generally slow (linearly convergent).
We would like a method that converges faster once we've gotten close to the local minimizer.
Think Newton's method!

Gauss-Newton method

For large problems, Newton's method (which requires second derivative information at each step) is regarded as too expensive. For the NLSQ problem, a related method is used to get faster convergence near the minimizer.

If we linearize the residual vector about the current guess, we get the local linear model

$$m(\mathbf{x}) = \mathbf{R}(\mathbf{x}_c) + \mathbf{J}(\mathbf{x}_c)(\mathbf{x} - \mathbf{x}_c) = \mathbf{R}(\mathbf{x}_c) + \mathbf{J}(\mathbf{x}_c)\mathbf{s} = \mathbf{0}$$

and we seek the minimum norm solution of this overdetermined system i.e. we solve a sequence of linear least squares problems

$$\mathbf{J}(\mathbf{x}_c)\mathbf{s} = -\mathbf{R}(\mathbf{x}_c); \quad \mathbf{x}_+ = \mathbf{x}_c + \mathbf{s}$$

This is called a Gauss-Newton step

The corresponding normal equations are

$$\mathbf{J}(\mathbf{x}_c)^T\mathbf{J}(\mathbf{x}_c)\mathbf{s} = -\mathbf{J}(\mathbf{x}_c)^T\mathbf{R}(\mathbf{x}_c); \quad \mathbf{x}_+ = \mathbf{x}_c + \mathbf{s}$$

An example

Newton step

Alternatively, we could minimize the local quadratic model of the cost function

$$\min m(\mathbf{x}) = C(\mathbf{x}_c) + \nabla C(\mathbf{x}_c)\mathbf{s} + \frac{1}{2}\mathbf{s}^T \nabla^2 C \mathbf{s}$$

where the necessary condition $\nabla m(\mathbf{s}) = 0$ gives

$$(\nabla^2 C)\mathbf{s} = -\nabla C(\mathbf{x}_c)$$

Recall the cost function is $C(\mathbf{x}) = \frac{1}{2}\mathbf{R}(\mathbf{x})^T\mathbf{R}(\mathbf{x})$ so

$$\nabla C(\mathbf{x}_c) = \mathbf{J}(\mathbf{x}_c)^T \mathbf{R}(\mathbf{x}_c)$$

and

$$\nabla^2 C(\mathbf{x}_c) = \mathbf{J}(\mathbf{x}_c)^T \mathbf{J}(\mathbf{x}_c) + \Sigma_m r_i(\mathbf{x}_c)\nabla^2 r_i(\mathbf{x}_c)$$

giving

$$\left[\mathbf{J}(\mathbf{x}_c)^T \mathbf{J}(\mathbf{x}_c) + \Sigma_m r_i(\mathbf{x}_c)\nabla^2 r_i(\mathbf{x}_c)\right]\mathbf{s} = -\mathbf{J}^T(\mathbf{x}_c)\mathbf{R}(\mathbf{x}_c); \quad \mathbf{x}_+ = \mathbf{x}_c + \mathbf{s}$$

This Newton step is NOT the same as a Gauss-Newton step.

Connection with Gauss-Newton

The second term in $\nabla^2 C$ i.e. $\Sigma_m r_i(\mathbf{x}_c)\nabla^2 r_i(\mathbf{x}_c)$ requires the evaluation of M Hessian $N \times N$ matrices and so is expensive. This term is missing from the Gauss-Newton step, which requires only first derivative information $\mathbf{J}$.

If the problem is a zero residual problem i.e. $\mathbf{R}(\mathbf{x}^*) = 0$, then the second term vanishes close enough to the minimum, so we might expect the second term to be negligible for small residual problems, or for models that are nearly linear i.e. r_i'' small.

So we could regard the Gauss-Newton step as an approximate Newton step, using the dominant (and cheaper) parts of the Newton step.

Local convergence

We know Newton's method is quadratically convergent, once we are close enough to the minimum. What about the Gauss-Newton step?
Under Standard assumptions, one can prove:

Theorem

$\exists K, \delta > 0$, *such that if* $\mathbf{x}_c \in \mathcal{B}(\delta) \implies$
$||e_+|| \leq K \left(||e_c||^2 + ||\mathbf{R}(\mathbf{x}^*)|| ||e_c|| \right)$

Consequences:

1. for a zero residual problem, Gauss-Newton is quadratically convergent
2. for a small residual problem or close to the minimum, Gauss-Newton is linearly convergent

$$||e_+|| \leq r||e_c|| \quad r < 1$$

and typically faster than steepest descent
3. for a large residual problem or far from the minimum, there is no reason to expect Gauss-Newton to converge

End of Lecture 17