Mathematical Modelling
Lecture 6 – Optimisation

Phil Hasnip
phil.hasnip@york.ac.uk

Overview of Course

- Model construction $\longrightarrow$ dimensional analysis
- Experimental input $\longrightarrow$ fitting
- Finding a ‘best’ answer $\longrightarrow$ optimisation
- Tools for constructing and manipulating models $\longrightarrow$ networks, differential equations, integration
- Tools for constructing and simulating models $\longrightarrow$ randomness
- Real world difficulties $\longrightarrow$ chaos and fractals

A First Course in Mathematical Modeling by Giordano, Weir & Fox, pub. Brooks/Cole. Today we’re in chapters 7 and 12.

Aim

- Optimisation: given a model, what is the ‘best’ possible output?

‘Best?’

What do we mean by ‘best’? Depends on the situation! E.g.

- Electrons in a crystal
  Find the lowest energy state
- Performance of shares
  Show me the money! Find the shares that give maximum profit
- Population
  Find the equilibrium population (stationary point)

It’s up to us!

Optimisation

Most optimisation problems can be converted into a minimisation problem

E.g. $g(x) = -f(x)$ turns a maximisation problem into a minimisation one.

The function we wish to optimise is called the objective function.

There might be more than one objective function...

Today

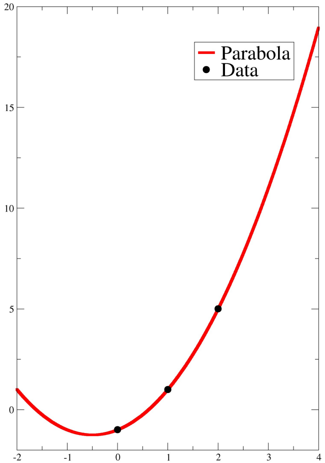
- One variable
- Many variables
- Many minima

Models with one variable

If we have a simple (differentiable!) model of only one variable, then we can use ordinary calculus:

- $f = f(x)$
- Differentiate to get $\frac{df}{dx}$
- Find the stationary points $\frac{df}{dx} = 0$
- Classify stationary points (min, max, inflexion)

Example



Models with many variables

If we have a (differentiable!) model of more than one variable:

- $f = f(x, y, z, \ldots)$
- Multivariate calculus: derivative is now a vector $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \ldots\right)$
- Find the stationary points $\nabla f = (0, 0, \ldots)$
- Classify stationary points (min, max, inflexion, saddle point)

But solving $\nabla f = (0, 0, \ldots)$ can be extremely difficult!

Constraints

Sometimes our solution might have to obey some constraints, for example:

- Number of particles is constant
- Money available to invest is constant
- No two electrons are in the same state
- Must stay on the road

Optimisation summary

Optimise the set of objective functions

$$f_i\left(x_1, x_2, \ldots, x_N\right)$$

subject to the set of constraints

$$g_j\left(x_1, x_2, \ldots, x_N\right) = b_j$$

where b_j are constants, to find the optimal inputs

$$\vec{X} = \left(x_1, x_2, \ldots, x_N\right)$$

Optimisation summary

There are two particular special cases

- f and g are linear in $\vec{X}$ $\longrightarrow$ linear programming
- x_i are integers $\longrightarrow$ integer programming

Examples

Models with many variables

In practice we can't usually solve for the stationary points exactly.

Suppose we want to find the minimum of the function $f(x)$. First we pick a starting point (guess!), then:

1. Differentiate to get vector $\nabla f = \left(\frac{df}{dx}, \frac{df}{dy}, \ldots \right)$ at that point
2. Move from point along $-\nabla f$ to find minimum
   line minimisation
3. Repeat step 1

Steepest descent

This method always moves in the negative gradient direction, so is called the method of steepest descent.

Beyond steepest descents

There are lots of methods that aim to do better than steepest descents. One of the simplest such methods is called conjugate gradients.

Most of these methods will get to the minimum of a quadratic function in one step.

All functions are approximately quadratic near the minimum! Taylor expansion:

$$f(x_0 + \delta x) \approx f(x_0) + \left.\frac{\partial f}{\partial x}\right|_{x_0} \delta x + \frac{1}{2}\left.\frac{\partial^2 f}{\partial x^2}\right|_{x_0} \delta x^2$$

Non-differentiable models

If we can't differentiate our function then we can't use these gradient methods. Guess some points, try to bracket minimum, then try to search in that region.

- Binary searches
- Golden section searches

Multiple minima

If there is more than one minimum, how do we know the one we've found is the lowest?

We don't!

- Monte-carlo
- Simulated annealing
- Genetic algorithms

Summary

When trying to use your model to determine the ‘best’ inputs:

- Decide objective function and what you mean by ‘best’ value
- Function of one variable – can often solve directly
- Function of many variables – usually need to solve iteratively (steepest descent, conjugate gradients)
- Many local minima – need a global optimisation method (simulated annealing, Monte Carlo, genetic algorithms)