Optimisation

- Many tasks require minimising or maximising a function.
  - When fitting a model to an image, one minimises an error function
    - Fitting lines to data
    - Fitting a PDF to data
    - Finding image features etc
- Suggested Reading:
  - “Numerical Recipes in C”, Press et.al.

Maxima and Minima

- Any maximum point of \( f(x) \) is a minimum of \(-f(x)\)
  - Need only consider minimisation
- Global minima
  - Lowest value of \( f(x) \) across all valid \( x \)
- Local minima
  - \( f(x) \) lower than all neighbours,
  - \( f(x) < f(x+d) \) for all small \(|d|\)

Methods

- Local Methods
  - find local minimum near to a starting point
- Global methods
  - find the global minima over a region (very hard)
  - Typically we polish the resulting global minima by applying a local optimiser

Choice of method

- Choice of method will depend on
  - Whether we need local/global solution
  - Whether we can calculate derivatives efficiently
  - Number of dimensions
  - Amount of memory available

Optimisation in 1D: Brackets

- A minimum is bracketed by three points, \( a < b < c \) such that \( f(b) < f(a) \) and \( f(b) < f(c) \)
- Minimum must be in range \([a, c]\)
Golden section search

- Find initial bracket \([a,b,c]\)
- Choose new point \(x\) in \([a,c]\)
  - If \(x < b\)
    - if \(f(x) < f(b)\) then new interval \([a,x,b]\) else \([x,b,c]\)
  - If \(x > b\)
    - if \(f(x) < f(b)\) then new interval \([b,x,c]\) else \([a,b,x]\)
- New interval always smaller
- Repeat until interval sufficiently small

Examples

- New bracket contains lowest point and points either side

Choice of x

- Wish to ensure continuous, predictable reduction in width
- Require bracket to shrink by fixed amount
- Suppose initial bracket is \([0,k,1]\)
- Suppose we test a point \(a\) fraction \(k\) along \([k,1]\), ie at \(k + k(1-k)\)

\[
\frac{0}{k} \quad \frac{k}{k + k(1-k)} \quad \frac{1}{1 + k(1-k)}
\]

- Final bracket will either be
  - \([0,k,k(2-k)]\) (width \(k(2-k)\)) or
  - \([k,k(2-k),1]\) (width \(1-k\))
- For consistent convergence, we require

\[
\frac{4}{k(2-k)} = 1 - \delta \quad \frac{4}{k(2-k)} = 1 - \frac{3\sqrt{5} - 4 + 1}{3} = 1 - \frac{\sqrt{5}}{2} = 0.381966.
\]

Choice of x

- Choose \(x\) a fraction \(k\) along largest interval, measured from central point
  - \(k = 0.5(3 - \sqrt{5})\) (the Golden section)
- If initial bracket is \([a,b,c]\) with \((b-a)/(c-a) = k\),
  - after \(n\) iterations bracket width will be

\[(1-k)^n(c-a) \approx 0.62^n(c-a)\]
Convergence rates

- Suppose width of bracket at iteration \( i \) is \( w_i \)
- The uncertainty shrinks at each iteration
  \[ w_{i+1} = c \cdot w_i^k \]
  - If \( k=1 \) we have ‘linear’ convergence
  - If \( k>1 \) we have ‘super-linear’ convergence
  - If \( k=2 \) we have ‘quadratic’ convergence

eg for Golden Section : \( w_{i+1} = 0.62 w_i \), so linear.

Obtaining initial bracket

- Plot function and estimate position of min.
- General approach:
  - Guess at two points
  - Step in downward direction until function rises again
  - Bracket is the two points found before the function rose, and the one after.

Golden Section search

- Guaranteed to find the local minimum given an initial bracket
- Reliable but not necessarily fast

When to stop

- In theory can find minima to arbitrary accuracy
- In practice we’re limited by machine precision, \( \epsilon \)
- On this machine: \( 1.0 + \delta = 1.0 \) for \( \delta < \epsilon \)
  - Float (32 bit) \( \epsilon = 10^{-8} \)
  - Double (64 bit) \( \epsilon = 10^{-16} \)

When to stop (II)

- The shape of a function \( f(x) \) around a minimum at \( c \) is given by Taylor’s theorem
  \[
  f(x) = f(c) + f'(c)(x-c) + 0.5 f''(c)(x-c)^2
  \]
  (First derivative is zero at the minimum)

When to stop (III)

- For \( f(x) \) to be distinguishable from \( f(c) \)
  \[
  \frac{0.5 f''(c)(x-c)^2}{f(c)} > \frac{\epsilon}{\sqrt{x-c}|\sqrt{\epsilon}} | \frac{f'(c)}{f'(c)} |
  \]
  \( |x-c| > \sqrt{\epsilon} | \frac{f'(c)}{f'(c)} | = 1 \)

  Bracket width : \( |x-c| > \sqrt{\epsilon} | c | \)

  float: \( \sqrt{\epsilon} = 10^{-4} \)
  double: \( \sqrt{\epsilon} = 10^{-8} \)
Parabolic Interpolation

- Near a minimum, most smooth functions will be approximately parabolic
- Given 3 points, fit parabola to estimate minima

Brent’s method

- 1D optimisation method useful when derivatives not available
- Uses a cunning combination of golden section search and parabolic interpolation
- Fast and reliable

Using Derivatives in 1D search

- Derivative information can be used to do interpolation with cubics or higher order polynomials
- More conservative approach recommended:
  - Given bracket \([a,b,c]\),
    - If \(f'(b)<0\) choose \(x\) in \([b,c]\)
    - If \(f'(b)>0\) choose \(x\) in \([a,b]\)
  - Use linear interpolation to select \(x\)

Using 1D derivatives

Using 2nd derivatives

- Suppose we can efficiently evaluate \(f(x), f'(x), f''(x)\)
- Taylor expansion:
  \[
  f(x+dx) = f(x) + f'(x)dx + 0.5f''(x)dx^2 + O(dx^3)
  \]
  \[
  f'(x+dx) = f'(x) + f''(x)dx
  \]
- If the minima is at \(x+dx\), then \(f'(x+dx)=0\)
  \[
  f'(x) + f''(x)dx = 0 \quad dx = -\frac{f'(x)}{f''(x)}
  \]

Newton-Raphson iterations

- Initial estimate: \(x_0\)
- Repeat \(x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}\)
- Until converged
  - Exhibits quadratic convergence
  - However, no guarantee that it will converge
  - Important to retain a bracket, and to test each new point