Computational statistics

Lecture 1: Optimizing smooth univariate functions

Thierry Denœux

February 16, 2016

Computational statistics

- Modern methods in statistics and econometrics rely heavily on computational methods, for instance,
  - Nonlinear optimization
  - Monte Carlo simulation
  - Resampling techniques (bootstrap, cross-validation)
  - Non parametric density estimation and smoothing
  - Machine Learning, data mining, big data analysis, etc.
- "Computational statistics" is a branch of Statistics at the intersection with Computer Science. It concerns the study of efficient procedures for solving statistical problems with computers

Contents of this course

- Three parts:
  1. Part I: optimization
  2. Part II: simulation and resampling
  3. Part III: density estimation, smoothing, statistical learning
- We will use the "R" programming language (free, flexible, large collection of available statistical methods)

Part I: Optimization

Many problems in statistics can be seen as optimizing (i.e., minimizing or maximizing) some function,

- maximizing the likelihood
- finding the mode of the posterior density, or highest posterior density intervals
- minimizing risk in Bayesian decision problems
- minimizing empirical risk in machine learning problems, etc.

Categories of optimization problems

- continuous vs. combinatorial optimization
- univariate vs. multivariate
- constrained vs. unconstrained

Contents of this course (Part I)

1. Optimizing smooth univariate functions: Bisection, Newton's method, Fisher scoring, secant method
2. Optimizing smooth multivariate functions: nonlinear Gauss-Seidel iteration, Newton's method, Fisher scoring, Gauss-Newton method, ascent algorithms, discrete Newton method, quasi-Newton methods
3. Combinatorial optimization: local search, ascent algorithms, simulated annealing, genetic algorithms
4. Expectation-Maximization (EM) algorithm for maximizing the likelihood or posterior density

Overview

Introduction to optimization

- In this first part, the real-valued function $g : \mathbb{R}^n \to \mathbb{R}$ to be maximized or minimized will be assumed to be smooth (at least differentiable)
- It may be a likelihood, a profile likelihood, a Bayesian posterior, or some other function
- Maximizing g is equivalent to minimizing $-g$
- Unless otherwise specified, we will consider maximization problems, without loss of generality

Introduction to optimization (continued)

- For maximum likelihood estimation, g is the log likelihood function, ℓ, and $\mathbf{x}$ is the corresponding parameter vector, $\boldsymbol{\theta}$. If $\hat{\boldsymbol{\theta}}$ is a MLE, it maximizes the log likelihood. Therefore $\hat{\boldsymbol{\theta}}$ is a solution to the score equation

$$\ell'(\boldsymbol{\theta}) = \mathbf{0},$$

where $\ell'(\boldsymbol{\theta}) = \left(\frac{\partial \ell(\boldsymbol{\theta})}{\partial \theta_1}, \ldots, \frac{\partial \ell(\boldsymbol{\theta})}{\partial \theta_n}\right)^T$ and $\mathbf{0}$ is a column vector of zeros.

- We see that optimization is intimately linked with solving nonlinear equations. Finding a MLE amounts to finding a root of the score equation.
- The maximum of g is a solution to $\mathbf{g}'(\mathbf{x}) = \mathbf{0}$.

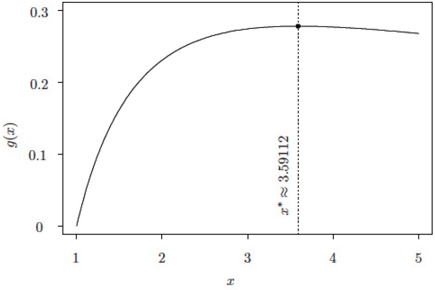
Univariate Optimization for Smooth g

- Example 1: Maximize

$$g(x) = \frac{\log\{x\}}{1+x}$$

with respect to x.

- We cannot find the root of $g'(x) = \frac{1+1/x-\log x}{(1+x)^2}$ analytically.



- The maximum of $g(x) = \frac{\log\{x\}}{1+x}$ occurs at $x^* \approx 3.59112$, indicated by the vertical line

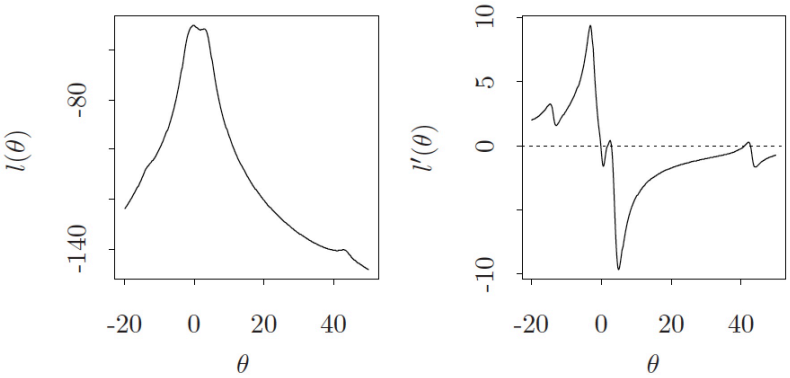
Example 2

- The following data are an i.i.d. sample from a Cauchy$(\theta, 1)$ distribution:
  1.77, −0.23, 2.76, 3.80, 3.47, 56.75, −1.34, 4.24, −2.44, 3.29, 3.71, −2.40, 4.53, −0.07, −1.05, −13.87, −2.53, −1.75, 0.27, 43.21.
- The likelihood function is

$$L(\theta) = \prod_{i=1}^{20} \frac{1}{\pi\left(1 + (x_i - \theta)^2\right)}$$

  Find the MLE for θ.
- The score function has multiple roots requiring numerical solution.

Log likelihood and score function for the Cauchy data



Local vs. global maximum

- A vector $\mathbf{x}_0$ is a local maximum of g if $\exists \epsilon > 0$ such that, for all $\mathbf{x} \in \mathbb{R}^n$,

$$\|\mathbf{x} - \mathbf{x}_0\| \leq \epsilon \Rightarrow g(\mathbf{x}_0) \geq g(\mathbf{x})$$

- A vector $\mathbf{x}_0$ is a global maximum of g if, for all $\mathbf{x} \in \mathbb{R}^n$,

$$g(\mathbf{x}_0) \geq g(\mathbf{x})$$

- We usually want to find a global maximum, but optimization algorithms can only be guaranteed to converge to a local maximum
- Solution: restart the algorithm from different initial conditions, but we can never be sure to have reached a global maximum

Iterative Methods

- Recall the simple example where we seek to maximize

$$g(x) = \frac{\log\{x\}}{1+x}$$

  with respect to x.
- We will rely on successive approximations of the solution.
- If we know that the maximum is around 3, it might be reasonable to use $x^{(0)} = 3.0$ as an initial guess, or starting value.
- An updating equation will be used to produce an improved guess, $x^{(t+1)}$, from the most recent value $x^{(t)}$, for $t = 0, 1, 2, \ldots$ until iterations are stopped.

Overview

Bisection Method

- If g' is continuous on $[a_0, b_0]$ and $g'(a_0)g'(b_0) \leq 0$ then the intermediate value theorem implies that there exists at least one $x^* \in [a_0, b_0]$ for which $g'(x^*) = 0$ and hence x^* is a local optimum of g.
- To find such a root, the bisection method systematically shrinks the interval from $[a_0, b_0]$ to $[a_1, b_1]$ to $[a_2, b_2]$ and so on, where $[a_0, b_0] \supset [a_1, b_1] \supset [a_2, b_2] \supset \cdots$ and so forth.
- If these intervals are chosen to retain $g'(a_i)g'(b_i) \leq 0$, then the ith interval contains a root.

Bisection Method

- Let $x^{(0)} = (a_0 + b_0)/2$ be the starting value.
- The updating equations are

$$[a_{t+1}, b_{t+1}] = \begin{cases} [a_t, x^{(t)}] & \text{if } g'(a_t)g'(x^{(t)}) \leq 0 \\ [x^{(t)}, b_t] & \text{if } g'(a_t)g'(x^{(t)}) > 0 \end{cases}$$

and

$$x^{(t+1)} = (a_{t+1} + b_{t+1})/2.$$

- If g has more than one root in the starting interval, it is easy to see that bisection will find one of them, but will not find the rest.

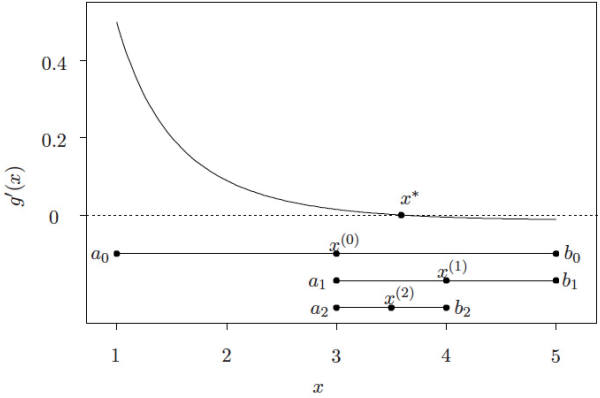
Example

- To find the value of x maximizing

$$g(x) = \frac{\log\{x\}}{1+x},$$

  we might take $a_0 = 1$, $b_0 = 5$, and $x^{(0)} = 3$.
- The following figure illustrates the first few steps of the bisection algorithm.
- For continuous smooth functions, bisection is guaranteed to converge to a root because a root is always in the interval and the length of the interval halves at each iteration. However, the method is slow.

Example



The top portion of this graph shows $g'(x)$ and its root at x^*. The bottom portion shows the first three intervals obtained using the bisection method with $(a_0, b_0) = (1,\ 5)$. The tth estimate of the root is at the center of the tth interval.

Stopping Criteria

- Near the root $g'(x^{(t+1)}) \approx 0$. However, relatively large changes from $x^{(t)}$ to $x^{(t+1)}$ are often seen even when $g'(x^{(t+1)})$ is roughly zero, therefore a stopping rule based directly on $g'(x^{(t+1)})$ is not very reliable.
- On the other hand, a small change from $x^{(t)}$ to $x^{(t+1)}$ is most frequently associated with $g'(x^{(t+1)})$ near zero. Therefore, we typically assess convergence by monitoring $\left|x^{(t+1)} - x^{(t)}\right|$ and use $g'(x^{(t+1)})$ as a backup check.
- The *absolute convergence criterion* mandates stopping when

$$\left|x^{(t+1)} - x^{(t)}\right| < \epsilon,$$

where ϵ is a constant chosen to indicate tolerable imprecision.

Stopping Criteria (continued)

- The *relative convergence criterion* mandates stopping when iterations have reached a point for which

$$\frac{\left|x^{(t+1)} - x^{(t)}\right|}{\left|x^{(t)}\right|} < \epsilon. \tag{1}$$

- This criterion enables the specification of a target precision (e.g., 'within 1%') without worrying about the units of x.
- Preference between the absolute and relative convergence criteria depends on the problem at hand:
  - If the scale of x is huge (or tiny) relative to ϵ, an absolute convergence criterion may stop iterations too reluctantly (or too soon).
  - The relative convergence criterion corrects for the scale of x, but can become unstable if $x^{(t)}$ values (or the true solution) lie too close to zero.
- In this latter case, another option is to monitor relative convergence by stopping when $\frac{\left|x^{(t+1)} - x^{(t)}\right|}{\left|x^{(t)}\right| + \epsilon} < \epsilon$.

Convergence diagnostics

- Also important to include stopping rules that flag a failure to converge:
  - Stop after N iterations, regardless of convergence. Do not devote all affordable iterations to one attempt! Budget time for many smaller attempts, anticipating convergence failures, data corrections, multiple starting values, etc.
  - Could stop if any convergence measure fails to decrease or cycle over several iterations, or if the solution itself cycle unsatisfactorily.
  - It is also sensible to stop if the procedure appears to be converging to a point at which $g(x)$ is inferior to another value you have already found (i.e., a known false peak or local maximum).
- Regardless of which such stopping rules you employ, any indication of poor convergence behavior means that $x^{(t+1)}$ must be discarded and the procedure somehow restarted in a manner more likely to yield successful convergence.

Overview

Newton's Method

- Suppose that g' is continuously differentiable and that $g''(x^*) \neq 0$.
- At iteration t, the approach approximates $g'(x^*)$ by the linear Taylor series expansion:

$$0 = g'(x^*) \approx g'(x^{(t)}) + (x^* - x^{(t)})g''(x^{(t)})$$

- Since g' is approximated by its tangent line at $x^{(t)}$, it seems sensible to approximate the root of g' by the root of the tangent line. Thus, solving for the root,

$$x^* \equiv x^{(t+1)} = x^{(t)} - \frac{g'(x^{(t)})}{g''(x^{(t)})} = x^{(t)} + h^{(t)}$$

- When the optimization of g corresponds to a MLE problem where $\hat{\theta}$ is a solution to $\ell'(\theta) = 0$, the updating equation for Newton's method is

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\ell'(\theta^{(t)})}{\ell''(\theta^{(t)})}.$$

Example

- For the simple function of Example 1,

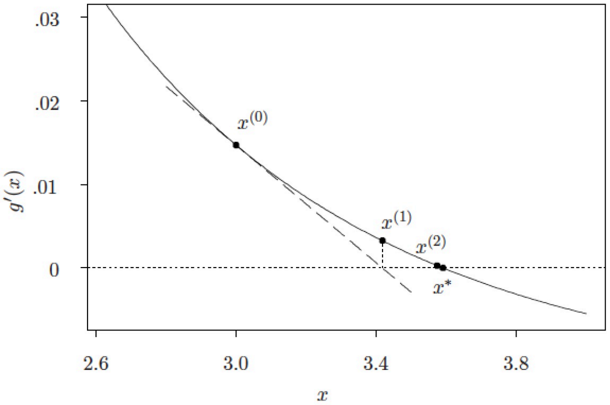
$$g(x) = \frac{\log\{x\}}{1+x},$$

we have

$$h^{(t)} = \frac{(x^{(t)}+1)\left(1+1/x^{(t)}-\log\{x^{(t)}\}\right)}{3+4/x^{(t)}+1/(x^{(t)})^2-2\log\{x^{(t)}\}}.$$

- The following figure illustrates the first several iterations. Starting from $x^{(0)} = 3.0$, Newton's method quickly finds $x^{(4)} \approx 3.59112$. For comparison, the first five decimal places of x^* are not correctly determined by the bisection method until iteration 19.

Example (continued)



At the first step, Newton's method approximates g' by its tangent line at $x^{(0)}$ whose root, $x^{(1)}$, serves as the next approximation of the true root x^*. The next step similarly yields $x^{(2)}$, which is already quite close to the root at x^*.

Convergence rate

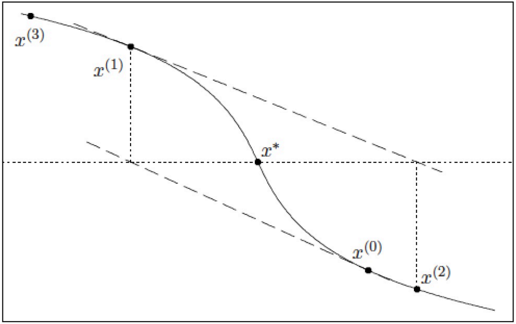
- Define the approximation error at iteration t, $\epsilon^{(t)} = x^{(t)} - x^*$
- A method has convergence of order β if $\lim_{t\to\infty} \epsilon^{(t)} = 0$ and

$$\lim_{t\to\infty} \frac{\left|\epsilon^{(t+1)}\right|}{\left|\epsilon^{(t)}\right|^{\beta}} = c$$

  for some constants $c \neq 0$ and $\beta > 0$.
- Higher orders of convergence are better in the sense that precise approximation of the true solution is more quickly achieved.
- Newton's method has quadratic convergence order, $\beta = 2$
- Unfortunately, high orders are sometimes achieved at the expense of robustness: some slow algorithms are more foolproof than their faster counterparts.

Convergence of Newton's method

Newton's method may fail to converge. For instance



Starting from $x^{(0)}$, Newton's method diverges by taking steps that are increasingly distant from the true root, x^*. In contrast, the bisection method would converge in this case.

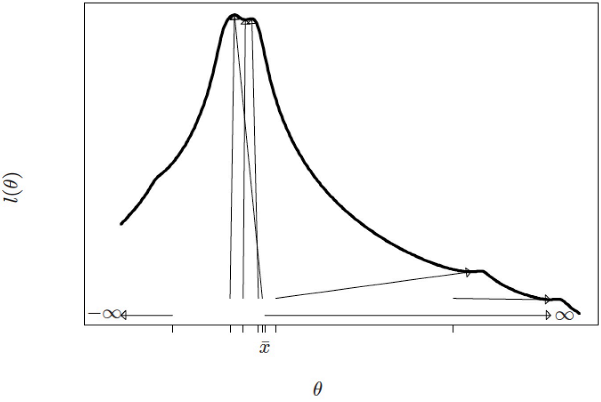
When does Newton's method converge?

- Theorem 1: If g' has two continuous derivatives and $g''(x^*) \neq 0$, then there exists a neighborhood of x^* for which NM converges to x^* when started from some $x^{(0)}$ in that neighborhood
- Theorem 2: If g' is twice continuously differentiable, is convex and has a root, then NM converges to that root from any starting point.

Reminder: a real-valued function f defined on an interval I is convex if the line segment between any two points on the graph of the function lies above or on the graph,

$$\forall x, y \in I, \forall \alpha \in [0,1], f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$$

Importance of the starting point



Log-likelihood for the Cauchy data. Arrows show convergence of Newton's method from several starting values

Fisher Scoring

- Fisher information (for scalar parameter) is

$$I(\theta) = \mathbb{E}\{\ell'(\theta)^2\} =^* -\mathbb{E}\{\ell''(\theta)\}$$

  *under regularity conditions.

- Reminder: for large iid samples, it holds approximately that $\hat{\theta} \sim \mathcal{N}(\theta, I(\theta)^{-1})$.
- Let $J(\hat{\theta}) = -\ell''(\hat{\theta})$ (observed information)
- Usually $I(\hat{\theta}) \approx J(\hat{\theta})$
- This suggests using the increment $h^{(t)} = \ell'(\theta^{(t)})/I(\theta^{(t)})$ where $I(\theta^{(t)})$ is the Fisher information evaluated at $\theta^{(t)}$.
- This yields

$$\theta^{(t+1)} = \theta^{(t)} + \ell'(\theta^{(t)})I(\theta^{(t)})^{-1}$$

Fisher Scoring vs. Newton's method

- Fisher scoring and Newton's method share the same asymptotic properties; either may be easier for a particular problem.
- In particular, $I(\theta)$ may be easier to compute. In the case of iid data, $I_n(\theta) = nI_1(\theta)$.
- The observed information $-\ell''(\theta)$ may be negative (resulting in divergence), specially far from the solution, whereas $I(\theta)$ is always positive.
- Generally, FS makes rapid improvements initially, while NM gives better refinements near the end.
- Case of the linear canonical one-parameter exponential family:

$$f(x;\theta) = b(x)\exp\left[\theta t(x) - c(\theta)\right]$$

We have $-\ell''(\theta) = c''(\theta) = I(\theta)$: FS and NM coincide.

Overview

Introduction

Bisection

Newton's method

Secant method

Secant Method

- When differentiating g' is difficult, we can replace the derivative by the discrete differenced approximation,

$$g''(x^{(t)}) \approx \frac{g'(x^{(t)}) - g'(x^{(t-1)})}{x^{(t)} - x^{(t-1)}}$$

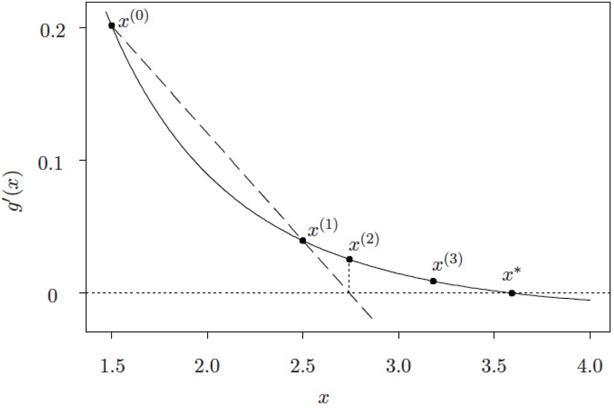
- This yields the update

$$x^{(t+1)} = x^{(t)} - g'(x^{(t)}) \frac{x^{(t)} - x^{(t-1)}}{g'(x^{(t)}) - g'(x^{(t-1)})}$$

  for $t \geq 1$.

- Requires two starting points, $x^{(0)}$ and $x^{(1)}$.
- The following figure illustrates the first steps of the method for maximizing the simple function of Example 1.
- The order of convergence of the secant method is superlinear: $\beta \approx 1.62$

Example



The secant method locally approximates g' using the secant line between $x^{(0)}$ and $x^{(1)}$. The corresponding estimated root, $x^{(2)}$, is used with $x^{(1)}$ to generate the next approximation