Numerical Methods for Optimization and Control Theory

Lecture 7: Calculating Derivatives

Gergó Lajos (Bognár Gergő)

ELTE Faculty of Informatics

Table of Contents

Motivation

Problem

Automatic calculation or approximation of the derivatives, required by the optimization method.

Approaches

- Finite differencing
- Automatic differentiation
- Symbolic differentiation

Table of Contents

Idea

Estimate the real function f in the neighborhood of x with its tangent line:

$$f(x+\varepsilon) \approx f(x) + f'(x) \cdot \varepsilon \quad (\varepsilon \in \mathbb{R} \setminus \{0\}).$$

Then we can approximate the derivative with a *finite difference*:

$$f'(x) \approx \frac{f(x+\varepsilon) - f(x)}{\varepsilon}.$$

We can then give error estimations based on Taylor's theorem, and extend the formula to multiple variables.

Definition (finite differences)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function, $\varepsilon > 0$, and denote the unit vectors as $e_i \in \mathbb{R}^n, \ (i = 1, \ldots, n)$.

The *forward difference*, or *one-sided difference* approximation of the gradient:

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x + \varepsilon e_i) - f(x)}{\varepsilon} \quad (i = 1, \ldots, n).$$

The *central difference* approximation of the gradient:

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x + \varepsilon e_i) - f(x - \varepsilon e_i)}{2\varepsilon} \quad (i = 1, \ldots, n).$$

Theorem (errors of the finite differences)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function, $\varepsilon > 0$, then the error of the *forward difference* is

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + \varepsilon e_i) - f(x)}{\varepsilon} + O(\varepsilon) \quad (i = 1, \dots, n),$$

Theorem (errors of the finite differences)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a three times continuously differentiable function, $\varepsilon > 0$, then the error of the *central difference* is

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + \varepsilon e_i) - f(x - \varepsilon e_i)}{2\varepsilon} + O(\varepsilon^2) \quad (i = 1, \dots, n).$$

Proof:

The idea of the *finite differences* comes from the Taylor's theorem:

$$f(x+p) = f(x) + \nabla f(x)^T p + O(\|p\|^2) \qquad (\text{if } f \in C^2),$$

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x) + O(||p||^3) \qquad (\text{if } f \in C^3).$$

Let $p = \pm\varepsilon e_i$ ($i = 1, \ldots, n$). Then if $f \in C^2$:

$$f(x+\varepsilon e_i) = f(x) + \varepsilon \frac{\partial f}{\partial x_i}(x) + O(\varepsilon^2)$$

$$\Longrightarrow \frac{\partial f}{\partial x_i}(x) = \frac{f(x+\varepsilon e_i) - f(x)}{\varepsilon} + O(\varepsilon).$$

Similarly, if $f \in C^3$:

$$f(x \pm \varepsilon e_i) = f(x) \pm \varepsilon \frac{\partial f}{\partial x_i}(x) + \frac{1}{2}\varepsilon^2 \frac{\partial^2 f}{\partial x_i^2}(x) + O(\varepsilon^3)$$

$$\Longrightarrow \frac{\partial f}{\partial x_i}(x) = \frac{f(x+\varepsilon e_i) - f(x-\varepsilon e_i)}{2\varepsilon} + O(\varepsilon^2). \quad \square$$

Recommendation (choice of ε)

The good practical choice for ε is $\sqrt{u}$ (*forward difference*) and $\sqrt[3]{u}$ (*central difference*), where u is the *unit roundoff* (the numeric precision of the floating point arithmetic).

Discussion: (*forward difference*)

Let $L > 0$ be a bound on $\|\nabla^2 f\|$, then

$$\left| \frac{\partial f}{\partial x_i}(x) - \frac{f(x + \varepsilon e_i) - f(x)}{\varepsilon} \right| \leq \frac{L\varepsilon}{2}.$$

Let $L_f > 0$ be a bound on $|f|$, and denote the computed values by *fl*, then the floating point error is bounded by uL_f:

$$|fl(f(x)) - f(x)| \leq uL_f, \quad |fl(f(x + \varepsilon e_i)) - f(x + \varepsilon e_i)| \leq uL_f.$$

Then the error bound of the computed finite difference is:

$$\left|\frac{\partial f}{\partial x_i}(x) - fl\left(\frac{f(x+\varepsilon e_i) - f(x)}{\varepsilon}\right)\right| \leq \frac{L\varepsilon}{2} + \frac{2uLf}{\varepsilon}.$$

The optimal choice is to minimize this error: $\varepsilon^2 = 4uL_f/L$.
In practice, assuming that the ratio L_f/L is moderate, the choice

$$\varepsilon = \sqrt{u}$$

is close to optimal. The total error is then close to $\sqrt{u}$.

Discussion: (*central difference*)

A similar discussion leads to the choice

$$\varepsilon = \sqrt[3]{u}.$$

Here, the total error is close to $u^{2/3}$.

Remarks:

- The *forward difference* requires the evaluation of f at $(n+1)$ points: at x and at $x + \varepsilon e_i$ $(i = 1, \ldots, n)$.
- The *central difference* requires evaluation at $(2n+1)$ points.
- Similarly to the *forward difference*, a *backward difference* can be defined, with similar properties:

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x) - f(x - \varepsilon e_i)}{\varepsilon} \quad (i = 1, \ldots, n).$$

  The *central difference* is then the average of the *forward* and *backward* differences.
- The *central difference* is more accurate than the *forward difference*. In theory, error terms are $O(\varepsilon^2)$ and $O(\varepsilon)$. Although in practice, a total error close to $u^{1/2}$ and $u^{2/3}$ can only be achieved.

Second Order Finite Differences

Motivation

Approximation of the Hessian matrix $\nabla^2 f(x)$, or the Hessian-vector product $\nabla^2 f(x)p$, required by some iteration methods.

Approaches

- Finite differencing of the gradient ∇f (if available).
  (Usually a non-symmetric approximation is produced. The symmetry can be recovered by replacing the approximation $\nabla^2 f(x) \approx H$ by $(H + H^T)/2$.)
- Approximation of $\nabla^2 f(x)p$ only (if the gradient is available). Based on Taylor's theorem ($\varepsilon > 0, p \in \mathbb{R}^n$):

$$\nabla f(x + \varepsilon p) = \nabla f(x) + \varepsilon \nabla^2 f(x)p + O(\varepsilon^2),$$

  that leads to the direct approximation:

$$\nabla^2 f(x)p = \frac{\nabla f(x + \varepsilon p) - \nabla f(x)}{\varepsilon}.$$

- Second order finite differencing (if no gradient is available).

Approaches (contd.)

- Second order finite differencing (if no gradient is available). Approximate the Hessian only with function values: approximate the gradient first, then the Hessian.

  In case of *forward difference* approximation:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} \approx \frac{\frac{\partial f}{\partial x_j}(x+\varepsilon e_i) - \frac{\partial f}{\partial x_j}(x)}{\varepsilon} \approx$$

$$\approx \frac{\frac{f(x+\varepsilon e_i+\varepsilon e_j)-f(x+\varepsilon e_i)}{\varepsilon} - \frac{f(x+\varepsilon e_j)-f(x)}{\varepsilon}}{\varepsilon} =$$

$$= \frac{f(x+\varepsilon e_i+\varepsilon e_j) - f(x+\varepsilon e_i) - f(x+\varepsilon e_j) + f(x)}{\varepsilon^2}.$$

Table of Contents

Idea

If the computational representation of a function is known (e.g. during the compilation), then we can produce code for the gradient by manipulating the function code.

Restrictions

The function must be a 'mathematical function', i.e. can be evaluated by performing only:

- addition, multiplication, division, exponentiation,
- trigonometric, exponential, logarithmic function evaluation.

Tool

Most important tool is the *chain rule*: if $f : \mathbb{R}^m \to \mathbb{R}$ and $y \in \mathbb{R}^n \to \mathbb{R}^m$ are differentiable functions, then

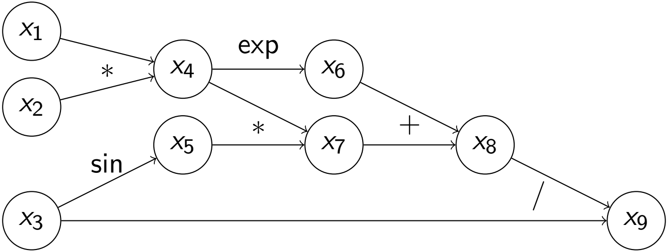
$$\nabla(f \circ y)(x) = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i} \nabla y_i(x) \quad (x \in \mathbb{R}^n).$$

.

Example

$$f(x) = \left(x_1 x_2 \sin x_3 + e^{x_1 x_2}\right) / x_3 \quad (x = (x_1, x_2, x_3) \in \mathbb{R}^3).$$

Computational graph:



Intermediate variables:

$x_4 = x_1 \cdot x_2,\ x_5 = \sin x_3,\ x_6 = e^{x_4},\ x_7 = x_4 \cdot x_5,\ x_8 = x_6 + x_7,\ x_9 = x_8 / x_3.$

Approaches

- *Forward mode*: evaluates and carries forward the derivative of the intermediate variables, concurrently by the evaluation of f (numeric or symbolic evaluation, code generation possibility).
- *Reverse mode*: after the evaluation of f, performs a reverse sweep, assembles the gradient from the partial derivatives of the child nodes in the computational graph (numeric evaluation only).

Example (forward mode)

Calculate the gradient of the intermediate variables $x_4, \ldots, x_9$ sequentially. At node $x_7 = x_4 \cdot x_5$, we already have ∇x_4 and ∇x_5:

$$x_7 = x_4 \cdot x_5 \implies \nabla x_7 = \frac{\partial x_7}{\partial x_4} \nabla x_4 + \frac{\partial x_7}{\partial x_5} \nabla x_5 = x_5 \nabla x_4 + x_4 \nabla x_5.$$