CS 70 - Foundations Of Applied Computer Science

Carter Kruse

Reading Assignment 7

CS 70 - Chapter 12 (Non-Linear Systems)

Overview: We consider two flavors of non-linear systems: root finding and optimization.

Root Finding: We are given some function $f(\boldsymbol{x})$ and we are interested in finding where $f(\boldsymbol{x}) = 0$. We have in fact already seen this problem for the linear case: $A\boldsymbol{x} = \boldsymbol{b}$ or $f(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b} = 0$, but now we want to consider non-linear problems where $f(\boldsymbol{x})$ could involve logarithms, exponentials, trigonometric functions, polynomials, etc of $\boldsymbol{x}$.

Optimization: We have a function $f(\boldsymbol{x})$ and we want to find $\boldsymbol{x}$ where $f(\boldsymbol{x})$ takes on its mimimum value. This is often written $\boldsymbol{x}^* = \text{argmin}_x f(\boldsymbol{x})$. Again, we have already seen this for the linear case in the form of linear least squares: $\text{argmin}_x |A\boldsymbol{x} - \boldsymbol{b}|$, but now we want to consider non-linear problems.

1D Root Finding

Given a function $f(x) : \mathbb{R} \to \mathbb{R}$, find a solution for $f(x) = 0$. We typically assume $f(x)$ is continuous or differentiable.

Bisection Method: This method is guaranteed to converge to the correct result, but at a linear rate, meaning the number of correct significant digits increases by roughly one in each iteration. (View Textbook)

Newton's Method: This method is not guaranteed to converge, but if it is started sufficiently close to a root, it has quadratic convergence, meaning the number of correct significant digits roughly doubles in each iteration.

Start with a first-order Taylor expansion of f at x:

$$f(x+h) \approx f(x) + f'(x)\,h$$

Solve for the root of the Taylor approximation of f:

$$f(x) + f'(x)\,h = 0 \to h = -\frac{f(x)}{f'(x)}$$

Given initial guess x_0, update the guess repeatedly by solving for the root of the Taylor approximation:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Secant Method: View Textbook

ND Root Finding

Given a function $f(\boldsymbol{x}) : \mathbb{R}^m \to \mathbb{R}^n$, find a solution for $f(\boldsymbol{x}) = 0$.

Newton's Method: Newton's method generalizes to higher dimensions via the multivariate Taylor expansion.

Start with a first-order multi-variate Taylor expansion of f at $\boldsymbol{x}$, where ∇f is the Jacobian matrix of first-order partial derivatives of f:

$$f(\boldsymbol{x} + \boldsymbol{h}) \approx f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})\, \boldsymbol{h}$$

Solve for the root of the Taylor approximation of f:

$$f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})\, \boldsymbol{h} = 0 \to h = -\left[\nabla f(\boldsymbol{x})\right]^{-1} f(\boldsymbol{x})$$

Given initial guess x_0, update the guess repeatedly by solving for the root of the Taylor approximation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left[\nabla f(\boldsymbol{x})\right]^{-1} f(\boldsymbol{x})$$

1D Optimization

Find the potential minima of a differentiable function $f : \mathbb{R} \to \mathbb{R}$.

Newton's Method: Since f is differentiable, a necessary (but not sufficient) condition for a minimum is that the derivative $f'(x) = 0$. Hence, we could find the root of the 1st derivative using Newton's method.

Start with a first-order Taylor expansion of the derivative f' at x:

$$f'(x + h) \approx f'(x) + f''(x)\, h$$

Solve for the root of the Taylor approximation of f':

$$f'(x) + f''(x)\, h = 0 \to h = -\frac{f'(x)}{f''(x)}$$

Given initial guess x_0, update the guess repeatedly by solving for the root of the Taylor approximation:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

We need to check whether x is a local minimum as opposed to a local maximum.

- If $f''(x) > 0$, then $f'(x) = 0$ is a local minimum.
- If $f''(x) < 0$, then $f'(x) = 0$ is a local maximum.
- If $f''(x) = 0$, then $f'(x) = 0$ is an inflection point.

ND Optimization

Find the potential mimima of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$.
Newton's Method: Since f is differentiable, a necessary (but not sufficient) condition for a minimum is that $\nabla f(\boldsymbol{x}) = 0$. Hence, we could find the root of the gradient using Newton's method.

Start with a first-order multi-variate Taylor expansion of f' at $\boldsymbol{x}$, where $g(\boldsymbol{x}) = \nabla f(\boldsymbol{x})$ and H_f is the Hessian matrix, which encodes the second-order derivatives of f in its elements as $H_{ij} = \left(\partial^2 f\right) / \left(\partial x_i \partial x_j\right)$:

$$g(\boldsymbol{x} + \boldsymbol{h}) \approx g(\boldsymbol{x}) + \nabla g(\boldsymbol{x})\, \boldsymbol{h} = \nabla f(\boldsymbol{x}) + H_f(\boldsymbol{x})\, \boldsymbol{h}$$

Solve for the root of the Taylor approximation of ∇f:

$$g(\boldsymbol{x}) + \nabla g(\boldsymbol{x})\,\boldsymbol{h} = 0 \to h = -\left[\nabla g(\boldsymbol{x})\right]^{-1} g(\boldsymbol{x}) = -H_f^{-1}(\boldsymbol{x})\,\nabla f(\boldsymbol{x})$$

Given initial guess x_0, update the guess repeatedly by solving for the root of the Taylor approximation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - H_f^{-1}(\boldsymbol{x})\,\nabla f(\boldsymbol{x})$$

We need to check whether x is a local minimum, a local maximum, or a saddle point by checking the eigenvalues of $H_f(\boldsymbol{x})$.

- If $H_f(\boldsymbol{x})$ is positive definite (i.e. all positive eigenvalues, so for any $\boldsymbol{h}$, we have $\boldsymbol{h}^T H_f(\boldsymbol{x})\,\boldsymbol{h} > 0$), then $\nabla f(\boldsymbol{x}) = 0$ is a local minimum.
- If $H_f(\boldsymbol{x})$ is negative definite (i.e. all negative eigenvalues, so for any $\boldsymbol{h}$, we have $\boldsymbol{h}^T H_f(\boldsymbol{x})\,\boldsymbol{h} < 0$), then $\nabla f(\boldsymbol{x}) = 0$ is a local maximum.
- If $H_f(\boldsymbol{x})$ is neither positive or negative definite (i.e. eigenvalues have mixed sign), then $\nabla f(\boldsymbol{x}) = 0$ is a saddle point.

Gradient Descent

The key idea is that if $\nabla f(\boldsymbol{x}) \neq 0$, we can always find a smaller $f(\boldsymbol{x})$ by taking a small step downhill, moving $\boldsymbol{x}$ in the negative gradient direction $-\nabla f(\boldsymbol{x})$. (View Textbook)

This method is much less expensive per iteration that Newton's method for optimization, as we do not have to form (nor invert) the Hessian.

The step size α_k for each iteration is called the learning rate in machine learning.

It is sometimes possible for find the best α_k in each iteration by solving a 1D optimization problem. This is called line search.

We can take many (possibly non-constant) steps along $\nabla f(\boldsymbol{x}_k)$ per iteration, while checking that we are still making downhill progress, before goign to the next iteration and re-evaluating a new gradient $\nabla f(\boldsymbol{x}_{k+1})$.
