ECE 4960

Spring 2017

Lecture 15

Nonlinear Equations and Optimization:
The Newton Method for Nonlinear Optimization

Edwin C. Kan
School of Electrical and Computer Engineering
Cornell University

Optimization of Nonlinear Functions

- Finding the optimization of **a scalar nonlinear objective function** $V(x_1, x_2, \ldots x_n)$ can be defined by the local Taylor expansion with respect to the vector x as:

$$V\left(\vec{x}+\Delta\vec{x}\right)=V\left(\vec{x}\right)+\nabla V\left(\vec{x}\right)\cdot\Delta\vec{x}+\frac{1}{2}\left(\Delta\vec{x}\right)^t\cdot\left[H\right]\Delta\vec{x}$$

- where the gradient function $\nabla V\left(\vec{x}\right)$ and the Hessian matrix $[H]$ are defined by:

$$\nabla V\left(x_1,x_2,\ldots,x_n\right)=\begin{pmatrix}\dfrac{\partial V\left(x_1,x_2,\ldots,x_n\right)}{\partial x_1}\\ \dfrac{\partial V\left(x_1,x_2,\ldots,x_n\right)}{\partial x_2}\\ \ldots\\ \dfrac{\partial V\left(x_1,x_2,\ldots,x_n\right)}{\partial x_n}\end{pmatrix}$$

or in vector form: $\nabla V\left(\vec{x}\right)=\left(\dfrac{\partial V\left(x_1,x_2,\ldots,x_n\right)}{\partial x_j}\right)$

Nonlinear Optimization

- If *V* is only linearly dependent on *x*, then **linear programming** teaches us that the optimal values have to be on the boundary. The solution method is a simple **search on the constraint function**.
- From the equivalency, the solution vector *x* that minimizes *V* will be the *x* that solves $\nabla V(\vec{x}) = 0$, provided that *[H]* is **positive definite.**

$$\nabla V(x_1, x_2, ..., x_n) = \begin{pmatrix} \frac{\partial V(x_1, x_2, ..., x_n)}{\partial x_1} \\ \frac{\partial V(x_1, x_2, ..., x_n)}{\partial x_2} \\ ... \\ \frac{\partial V(x_1, x_2, ..., x_n)}{\partial x_n} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ ... \\ 0 \end{pmatrix}$$

Newton's Method for Nonlinear Optimization

1. Set up the variables *x* and evaluation of *V(x)* AND $\nabla V(\vec{x})$.
2. Make up an **initial guess** $x^{(k)}$, starting from $k = 0$. Notice that this is a difficult choice and has dominant influence on the convergence behavior.
3. Evaluate $\nabla V(\vec{x})$ and the Hessian matrix for the update vector:

$$\Delta\vec{x}^{(k)} = -\underbrace{\left[H\left(\vec{x}^{(k)}\right)\right]}_{[[J]]}^{-1} \underbrace{\nabla V\left(\vec{x}^{(k)}\right)}_{f\left(\vec{x}^{(k)}\right)} \qquad [H] = \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial V}{\partial x_1}\right) & \frac{\partial}{\partial x_2}\left(\frac{\partial V}{\partial x_1}\right) & \cdots & \frac{\partial}{\partial x_n}\left(\frac{\partial V}{\partial x_1}\right) \\ \frac{\partial}{\partial x_1}\left(\frac{\partial V}{\partial x_2}\right) & \frac{\partial}{\partial x_2}\left(\frac{\partial V}{\partial x_2}\right) & \cdots & \frac{\partial}{\partial x_n}\left(\frac{\partial V}{\partial x_2}\right) \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial}{\partial x_1}\left(\frac{\partial V}{\partial x_n}\right) & \frac{\partial}{\partial x_2}\left(\frac{\partial V}{\partial x_n}\right) & \cdots & \frac{\partial}{\partial x_n}\left(\frac{\partial V}{\partial x_n}\right) \end{bmatrix}$$

4. Evaluate the norm of $\|\Delta x^{(k)}\|_2$ and $V(x^{(k)})$. Stop if $\|\Delta x^{(k)}\|_2 <$ tolerance (often set between 10^{-7} to 10^{-9}).
5. Update $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$, $k{+}{+}$, and return to Step 3 to iterate.

(Reminder) The Newton Raphson Method

1. Set up the variables and evaluation of *f(x)* for solving *x* that satisfis the nonlinear equation $f(x) = 0$

2. Make up an **initial guess** $x^{(k)}$, where $k = 0$ initially Notice that this is a difficult choice and has dominant influence on the convergence behavior.

3. Evaluate $f(x^{(k)})$ and its slope $f'(x^{(k)})$ or the Jacobian matrix J for the multi-variate case. Calculate the update vector:
   $\Delta x^{(k)} = -f(x^{(k)})/f'(x^{(k)})$ or $\Delta\vec{x}^{(k)} \cong -\left[J\left(\vec{x}^{(k)}\right)\right]^{-1} \cdot \vec{f}\left(\vec{x}^{(k)}\right)$

4. Evaluate the norm of $\|\Delta x^{(k)}\|_2$ and $\|f(x^{(k)})\|_2$. Stop if $\|\Delta x^{(k)}\|_2$ or $\|f(x^{(k)})\|_2$ < tolerance (often set between 10^{-7} to 10^{-9}).

5. Update $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$, k++, and return to Step 3 to iterate.

Local and Global Minimization

- At the minimum point: $\vec{x}*$

$$\nabla V(\vec{x}*) = 0 \qquad V(\vec{x}* + \Delta\vec{x}) > V(\vec{x}*)$$

- If and only if:

$$(\Delta\vec{x})^t [H(\vec{x}*)] \Delta\vec{x} > 0, \qquad \forall \Delta\vec{x}$$

- Local minimum at $\vec{x}*$: For a given R > 0, if $V(\vec{x}* + \Delta\vec{x}) \geq V(\vec{x}*) \quad \forall \|\Delta\vec{x}\| < R$
- Strict local miminum: For a given R > 0, if $V(\vec{x}* + \Delta\vec{x}) > V(\vec{x}*) \quad \forall \|\Delta\vec{x}\| < R$
- Global minimum: $R \to \infty$
- The minimum is nondegenerate if *[H]* is positive definite.

Hacker Practice

Use the quasi-Newton method with line search to solve the nonlinear optimization function V by making $x^{(0)} = (0, 0)$ and the local analysis of the Hessian matrix by 10^{-4} perturbation.

$$V = (3x_1^2 + x_2 - 4)^2 + (x_1^2 - 3x_2 + 2)^2$$

Report $\|x^{(k)}\|_2$, $\|\Delta x^{(k)}\|_2$, t, $V(x^{(k)})$.

Can you observe the quadratic convergence?

We know there are two local minima where $V = 0$ at (1, 1) and (–1, 1). How will you change the initial guess to get both?

Descent Methods for Nonlinear Optimization

- A descent direction $\Delta x^{(k)}$ is defined as:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + \Delta\vec{x}^{(k)} \quad \Rightarrow \quad V\left(\vec{x}^{(k+1)}\right) = V\left(\vec{x}^{(k)} + \Delta\vec{x}^{(k)}\right) < V\left(\vec{x}^{(k)}\right)$$

- Or equivalently

$$V\left(\vec{x}^{(k)} + \Delta\vec{x}^{(k)}\right) - V\left(\vec{x}^{(k)}\right) = \nabla V\left(\vec{x}^{(k)}\right) \cdot \Delta\vec{x} < 0$$

- The Newton direction is a choice of the descent direction, as

$$\Delta\vec{x}^{(k)} = -\left[H\left(\vec{x}^{(k)}\right)\right]^{-1} \nabla V\left(\vec{x}^{(k)}\right)$$

$$V\left(\vec{x}^{(k)} + \Delta\vec{x}^{(k)}\right) - V\left(\vec{x}^{(k)}\right) \cong \nabla V\left(\vec{x}^{(k)}\right) \cdot \Delta\vec{x}$$

$$= -\nabla V\left(\vec{x}^{(k)}\right) \cdot \left[H\left(\vec{x}^{(k)}\right)\right]^{-1} \nabla V\left(\vec{x}^{(k)}\right) < 0$$

As long as [H] is positive definite

The Steepest Descent Method

- Use of line search will help stabilize most descent methods.
- Evaluation of [*H*] can be too expensive, just like [*J*]. So we will take approximations and sacrifice the quadratic convergence

$$\Delta\vec{x}^{(k)} = -t\nabla V\left(\vec{x}^{(k)}\right) = -t\,\frac{V\left(x_1, x_2, \ldots, x_i + \Delta x_i, \ldots, x_n\right) - V\left(x_1, x_2, \ldots, x_n\right)}{\Delta x_i}$$

- Or similar to the Gauss-Seidel method that uses the best available *V* during evaluation:

$$\Delta\vec{x}^{(k)} = -t\,\frac{V\left(x_1 + \Delta x_1, x_2 + \Delta x_2, \ldots, x_i + \Delta x_i, \ldots, x_n\right) - V\left(x_1 + \Delta x_1, x_2 + \Delta x_2, \ldots, x_i, \ldots x_n\right)}{\Delta x_i}$$

The Conjugate Gradient Method

- During the steepest descent, we can make each $\Delta x^{(k)}$ to be orthogonal to the previous steps of to $\Delta x^{(0)}$ to $\Delta x^{(k-1)}$ by taking superposition with the previous steps.
- For example, for $n = 2$, if $\Delta x^{(0)} = (1, 0)$, and the first calculation of $\Delta x^{(1)}$ is (1, 1). The modified step in the CG method will be:

$(\Delta x^{(1)})_{CG} \cdot \Delta x^{(0)} = 0$ where $(\Delta x^{(1)})_{CG} = a \cdot \Delta x^{(0)} + \Delta x^{(1)}$

- Solving for a, we obtain $(\Delta x^{(1)})_{CG} = (0, 1)$ with $a = -1$.
- When the problem is nearly linear, we can guarantee to find the solution in less than n steps, as the correction vector would have covered the entire space with n orthogonal vectors.
- When V is highly nonlinear, the CG method may not find a minimum in n steps.

Hacker Practice

Use the steepest descent method with line search to solve the nonlinear optimization function V by making $x^{(0)}$ = (0, 0) and the local analysis by $\Delta x_i = 10^{-4} \cdot x_i$ perturbation.

$$V = (3x_1^2 + x_2 - 4)^2 + (x_1^2 - 3x_2 + 2)^2$$

$$\Delta\vec{x}^{(k)} = -t\nabla V\left(\vec{x}^{(k)}\right) = -t\,\frac{V\left(x_1, x_2, \ldots, x_i + \Delta x_i, \ldots, x_n\right) - V\left(x_1, x_2, \ldots, x_n\right)}{\Delta x_i}$$

Report $\|x^{(k)}\|_2$, $\|\Delta x^{(k)}\|_2$, t, $V(x^{(k)})$.

Can you observe the quadratic convergence?

We know there are two local minima where V = 0 at (1, 1) and (–1, 1). How will you change the initial guess to get both?