OPTIMIZATION AND NUMERICAL METHODS

DATA/MSML 603: Principles of Machine Learning

Optimization Problem

- **Optimization problem** (in **standard form**)

$$
\begin{array}{ll}
\text{minimize} & f_0(\mathbf{x}) \\
\text{subject to} & f_i(\mathbf{x}) \leq 0, \;\; i = 1, \ldots, m \\
 & h_j(\mathbf{x}) = 0, \;\; j = 1, \ldots, p
\end{array} \tag{1}
$$

  - $\mathbf{x} \in \mathbb{R}^n$ -- optimization variable(s)
  - $f_0 : \mathbb{R}^n \to \mathbb{R}$ -- objective function or cost function
  - $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, m$ -- inequality constraint functions
  - $h_j : \mathbb{R}^n \to \mathbb{R}, j = 1, \ldots, p$ -- equality constraint function

- Special case: If m=p=0, i.e., no constraints, the optimization problem is said to be "**unconstrained**"

Optimization Problem

- Examples
  - Maximum likelihood estimation: $$\text{maximize}_{\boldsymbol{\theta}} \sum_{k=1}^{n} \log\left(p(\mathbf{x}_k|\boldsymbol{\theta})\right)$$
  - Squared margin perceptron: $$\text{minimize}_{\mathbf{w},b} \sum_{k=1}^{n} \left(\max(0, 1 - y^k(\mathbf{w}^T\mathbf{x}^k + b))\right)^2$$
  - Hard-margin support vector machine: $$\begin{aligned} \text{minimize}_{\mathbf{w},b} \quad & \|\mathbf{w}\|_2^2 \\ \text{subject to} \quad & \max\left(0, 1 - y^k(\mathbf{w}^T\mathbf{x}^k + b)\right) = 0, \\ & k = 1, \ldots, n \end{aligned}$$

Unconstrained Optimization

- We will focus on unconstrained optimization here

$$\text{minimize}_{\mathbf{x}} \ f(\mathbf{x})$$

  - Will formulate it as a minimization problem
  - Maximization problem can be cast as a minimization problem by multiplying the objective function by -1

- We will discuss two basic but widely used numerical techniques for solving unconstrained optimization problems
  - Gradient descent method
  - Newton's method

Unconstrained Optimization

- Key steps in many numerical algorithms

1. Select an initial point $\mathbf{x}^0$ and set k = 0
2. Update the solution to $\mathbf{x}^{k+1}$ by first picking a direction in which we search and determine how much we will move in the indirection
3. Increase k by one and repeat Step 2 till some stopping condition is satisfied

- Common stopping conditions

1. Stop after a prespecified number of iterations
2. Stop if $\|\nabla f(\mathbf{x}^k)\| < \epsilon$ for some small ϵ

Basic Calculus – Linear Approximation

- Suppose $g : \mathbb{R} \to \mathbb{R}$ is a differentiable univariate function
  - First-order or linear approximation of g at $x \in \mathbb{R}$

$$g_1(y) = g(x) + g'(x)(y - x)$$

- Consider a differentiable multivariate function $f : \mathbb{R}^n \to \mathbb{R}$
  - First-order or linear approximation of f at $\mathbf{x} \in \mathbb{R}^n$

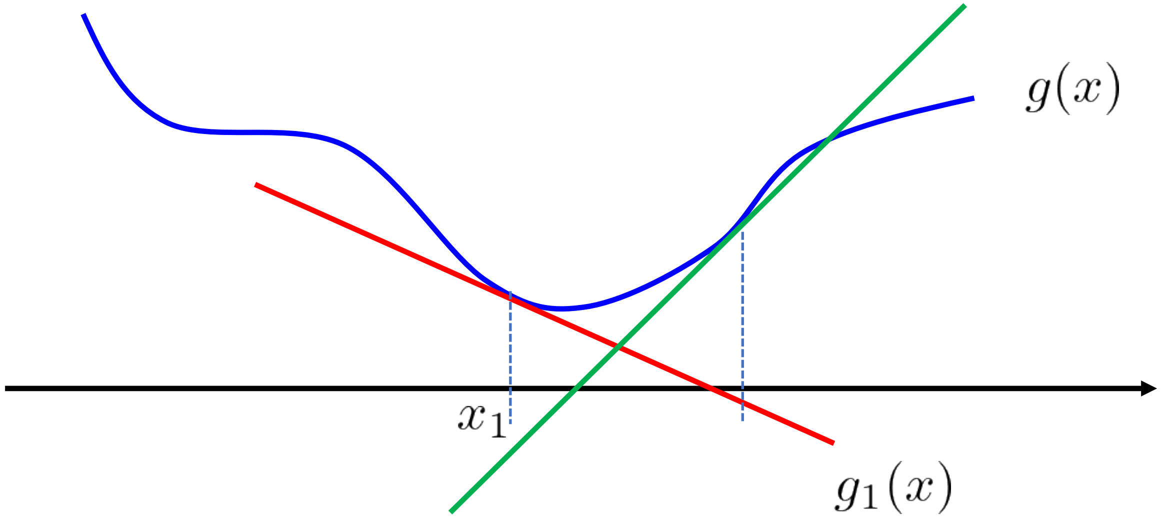
$$f_1(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x})$$

where

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}(\mathbf{x}) \quad \ldots, \frac{\partial f}{\partial x_n}(\mathbf{x}) \right]^T$$

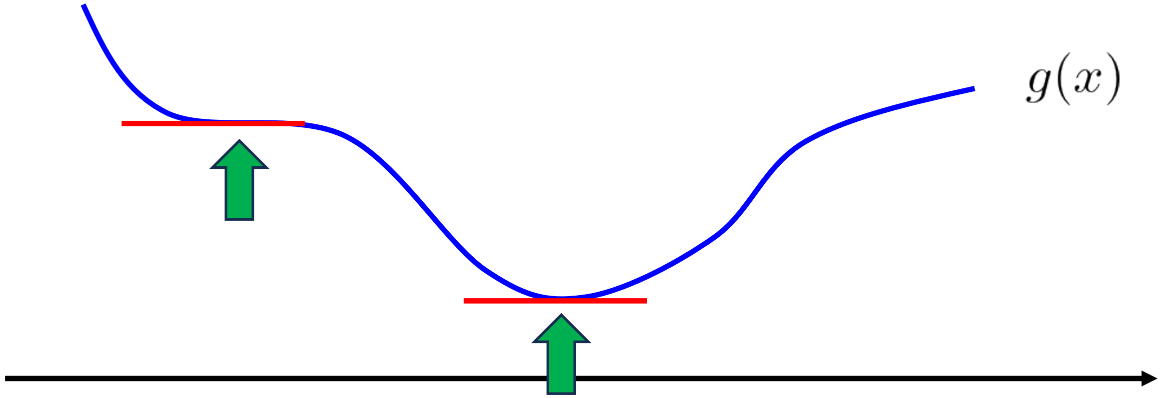
Basic Calculus – Linear Approximation

- Example



First-Order Condition for Optimality

- Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function
- Recall that, for a differentiable univariate function $g : \mathbb{R} \to \mathbb{R}$, a solution to $g'(x) = 0$ is potential minimizer



First-Order Condition for Optimality

- Similarly, for a multivariate objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, a solution to $\nabla f(\mathbf{x}) = \mathbf{0}$ is a potential minimizer
  - Gives us a system of n equations

$$
\begin{gathered}
\frac{\partial f}{\partial x_1}(\mathbf{x}) = 0 \\
\vdots \\
\frac{\partial f}{\partial x_n}(\mathbf{x}) = 0
\end{gathered}
$$

- Definition: A stationary point refers to a minimum, a maximum or a saddle point

First-Order Condition for Optimality

- First-order necessary condition for optimality: A solution $\mathbf{x}^*$ to

$$\text{minimize}_{\mathbf{x}} \; f(\mathbf{x})$$

with a differentiable objective function $f : \mathbb{R}^n \to \mathbb{R}$, must satisfy

$$\boxed{\nabla f(\mathbf{x}^*) = \mathbf{0}} \qquad (7\text{-}1)$$

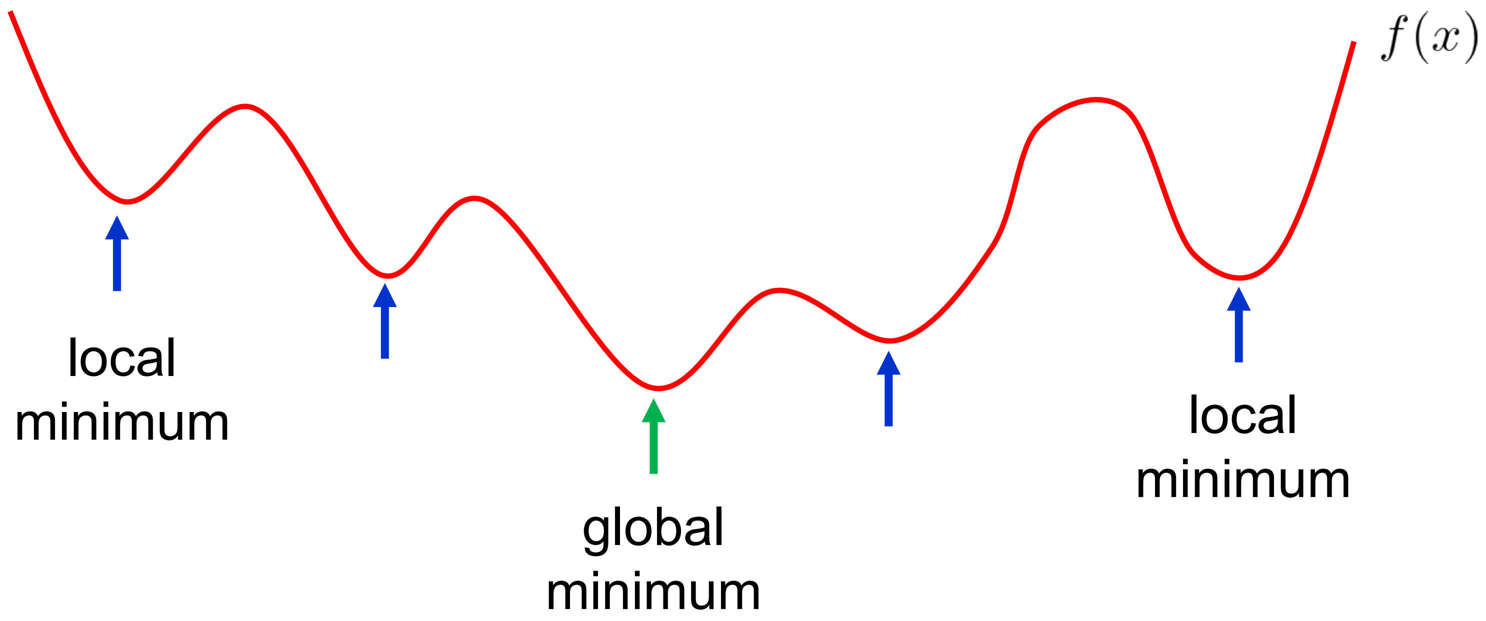
- Example:

$$\text{minimize}_{\mathbf{x}} \; f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r, \quad \mathbf{Q} \in \mathbb{S}_n$$

$$\nabla f(\mathbf{x}^*) = \mathbf{Q}\mathbf{x}^* + \mathbf{q} = \mathbf{0} \quad \Rightarrow \quad \mathbf{Q}\mathbf{x}^* = -\mathbf{q}$$

Issue

- In general, the minimization problem can have many local minimizers that satisfy (7-1)



Issue

- Question: How do we know whether the minimizer we found is a local minimizer or a global minimizer?
- Answer: In general, it is very difficult to tell. Even worse, we cannot even tell how bad the local minimizer is
- Exception: When the objective function is “convex”, any stationary point is a global minimum
- Question: What is a convex function?

Convex Functions

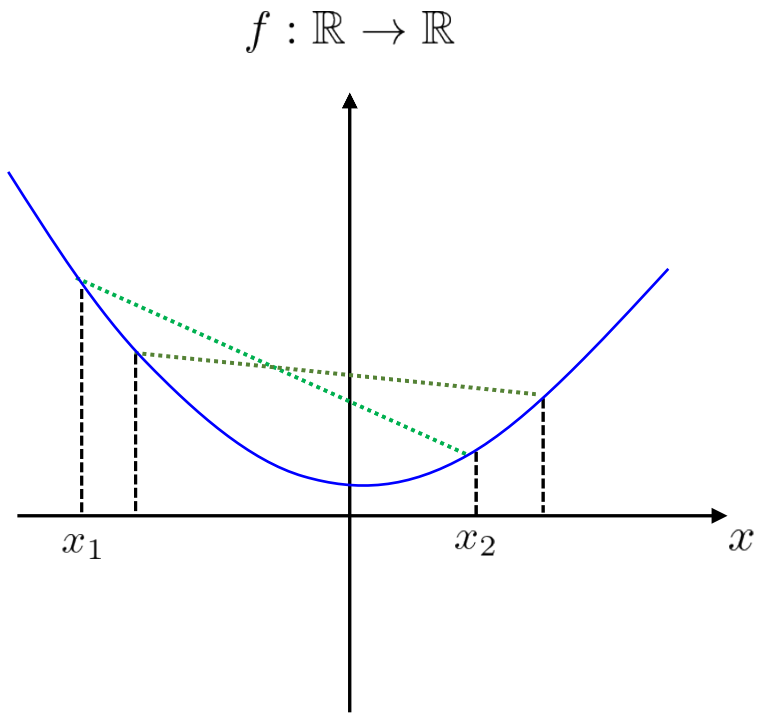
- Definition: A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be convex if, for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and $\alpha \in [0, 1]$, we have

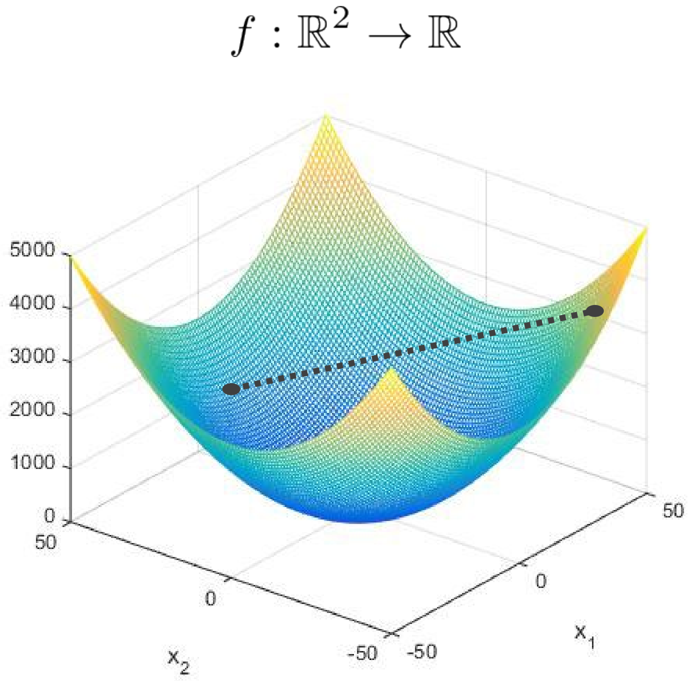
$$f(\alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2)$$

  - The function is said to be "concave" if the inequality goes the other way

- If $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable and its Hessian matrix $\nabla^2 f(\mathbf{x})$ is positive semidefinite for all $\mathbf{x} \in \mathbb{R}^n$, then it is convex

Convex Function (2)





$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta \cdot f(\mathbf{x}) + (1-\theta)f(\mathbf{y}) \quad \text{for all} \quad \theta \in [0,1]$$

Convex Functions

- Example: Quadratic function

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r, \qquad \mathbf{Q} \in \mathbb{S}_n$$

  - Hessian matrix at $\mathbf{x} \in \mathbb{R}^n$

$$\nabla f(\mathbf{x}) = \mathbf{Q}\mathbf{x} + \mathbf{q}$$

$$\nabla^2 f(\mathbf{x}) = \mathbf{Q}$$

  - Convex if and only if is positive semidefinite

Gradient Descent Method

- Use a linear approximation to determine the direction in which the objective function value can be reduced

$$f_1(\mathbf{x}) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k), \;\; k = 0, 1, 2, \ldots$$

  - Take the direction in which the tangent hyperplane angles downward most sharply
    - Called the “steepest descent direction”
    - The negative gradient: $-\nabla f(\mathbf{x}^k)$
- Gradient descent algorithm: $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_{k+1} \nabla f(\mathbf{x}^k), \quad k = 0, 1, 2, \ldots$
  - called the step size, step length, or a learning rate in ML

Gradient Descent Method

- Initialize: $\mathbf{x}^0 \in \mathbb{R}^m$
- Set k = 1
- While stopping condition not met
  - $\mathbf{x}^k = \mathbf{x}^{k-1} - \alpha_k \nabla f(\mathbf{x}^{k-1})$
  - k++

step size

Step Size

- Question: How do we choose the step sizes $\alpha_k, k = 1, 2, \ldots?$

1. Fixed step size
2. Exact line search
3. Backtracking line search

Step Size

- **Step size α_k selection** – Recall that (3) holds only locally
- Question is, how do we choose a finite step size α_k that guarantees a sufficient decrease (rate) in the objective function?
- Note that, once we fix the direction $\mathbf{d}_k$,

$$\phi_{\mathbf{d}_k}(\alpha) = f(\mathbf{x}_k + \alpha \cdot \mathbf{d}_k)$$

is a function of the scalar variable α

Step Size

1. **Exact line search** – a natural choice is

$$\alpha^* \in \arg\min_{\alpha \geq 0} \ \phi_{\mathbf{d}_k}(\alpha)$$

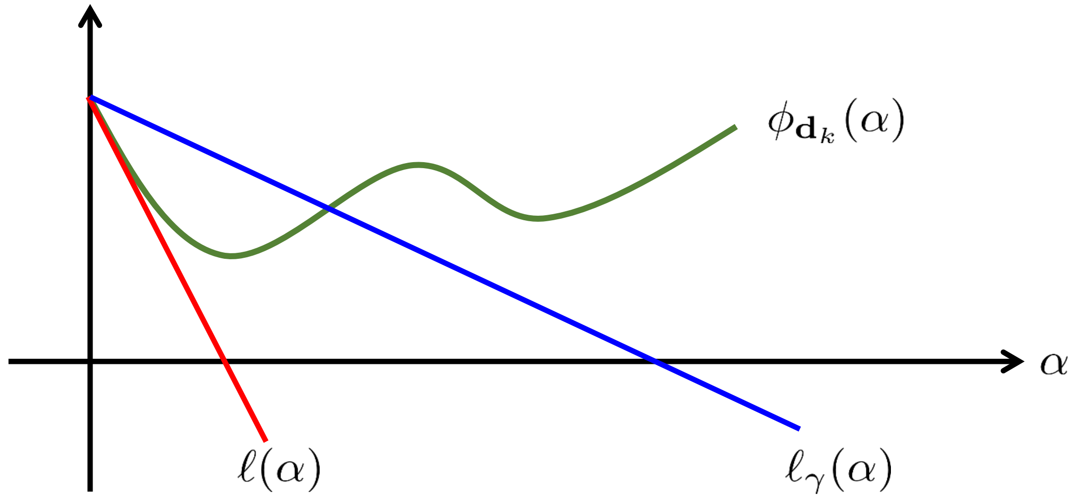
- Requires solving a univariate, generally non-convex optimization problem (unless f is convex)
- Rarely used in practice

Step Size

2. Backtracking line search

- Define (i) $\ell(\alpha) := \phi_{\mathbf{d}_k}(0) + \alpha\Delta_k,$ where $\Delta_k = \nabla f_0(\mathbf{x}_k)^T \mathbf{d}_k$, and

  (ii) for all $\gamma \in (0,1),\ \ell_\gamma(\alpha) = \phi_{\mathbf{d}_k}(0) + \alpha \cdot \gamma \cdot \Delta_k$

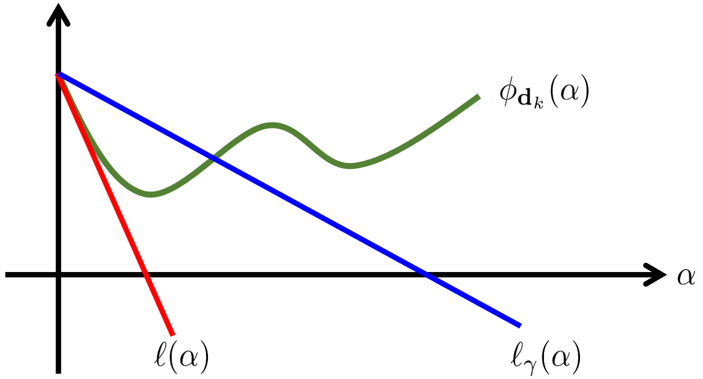


$\ell(\alpha) = \phi_{\mathbf{d}_k}(0) + \nabla f_0(\mathbf{x}_k)^T(\alpha\mathbf{d}_k)$ $\qquad$ $\ell_\gamma(\alpha) = \phi_{\mathbf{d}_k}(0) + \gamma\nabla f_0(\mathbf{x}_k)^T(\alpha\mathbf{d}_k)$

Step Size

- Since $\gamma \in (0,1)$, the line $\ell_\gamma(\alpha)$ lies above $\ell(\alpha)$ and also above $\phi_{\mathbf{d}_k}(\alpha)$ for sufficiently small α
- For all α for which $\phi_{\mathbf{d}_k}(\alpha) \leq \ell_\gamma(\alpha)$, we have

$$f_0(\mathbf{x}_k + \alpha \cdot \mathbf{d}_k) \leq f_0(\mathbf{x}_k) + \alpha \cdot \gamma\left(\nabla f_0(\mathbf{x}_k)^T \mathbf{d}_k\right)$$



  - Called "Armijo condition"
  - Provides a guaranteed "rate" of decrease, namely given by

$$-\gamma\Delta_k = -\gamma\left(\nabla f_0(\mathbf{x}_k)^T \mathbf{d}_k\right)$$

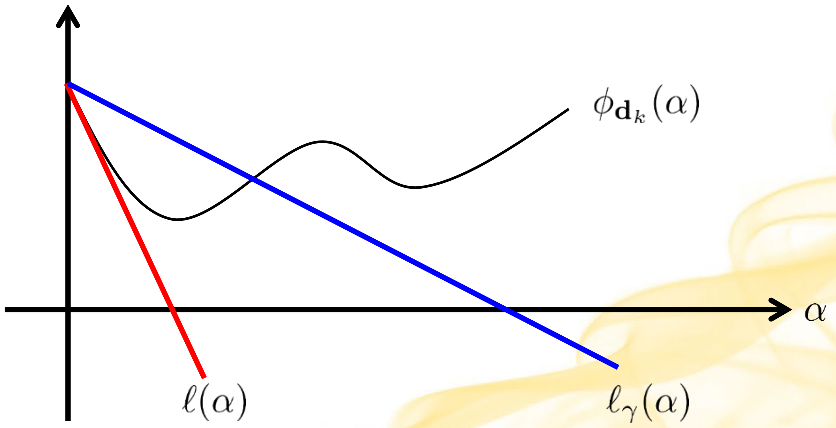
  - When $\phi_{\mathbf{d}_k}$ is bounded below, $\phi_{\mathbf{d}_k}$ and ℓ_γ must cross at least at one point

Step Size

- Backtracking line search looks for a (largest) value of α for which the Armijo condition holds (in a heuristic manner)

- **<u>Backtracking line search algorithm:</u>** $(\gamma, \beta \in (0,1))$

  1. Set $\alpha = \alpha_{\text{init}}, \ \Delta_k = \nabla f_0(\mathbf{x}_k)^T \mathbf{d}_k$
  2. If $f_0(\mathbf{x}_k + \alpha \cdot \mathbf{d}_k) \leq f_0(\mathbf{x}_k) + \alpha \cdot \gamma \cdot \Delta_k$, then return $\alpha_k = \alpha$
  3. Else, let $\alpha \leftarrow \beta \cdot \alpha$ and go to step 2



SCIENCE ACADEMY

Basic Calculus – Quadratic Approximation

- Suppose $g : \mathbb{R} \rightarrow \mathbb{R}$ is a twice differentiable univariate function
  - Second-order or quadratic approximation of g at $x \in \mathbb{R}$

$$g_2(y) = g(x) + g'(x)(y - x) + \frac{1}{2}g''(x)(y - x)^2$$

- Consider a twice differentiable multivariate function $f : \mathbb{R}^n \rightarrow \mathbb{R}$
  - Second-order or quadratic approximation of f at $\mathbf{x} \in \mathbb{R}^n$

$$f_2(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x})$$

where

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2}{\partial x_1^2} f(\mathbf{x}) & \frac{\partial^2}{\partial x_1 \partial x_2} f(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_n} f(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_n \partial x_1} f(\mathbf{x}) & \frac{\partial^2}{\partial x_n \partial x_2} f(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_n^2} f(\mathbf{x})) \end{bmatrix}$$

Newton's Method

- Approximate f using a quadratic approximation around $\mathbf{x}^k$

$$f(\mathbf{x}) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^k)^T \nabla^2 f(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k) \qquad \text{(7-2)}$$

  - We choose $\mathbf{x}^{k+1}$ to be a stationary point of (7-2)

- Question: How do we find a stationary point of (7-2)?

- Answer: Set the gradient of (7-2) to zero and solve for $\mathbf{x}^{k+1}$

$$\nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) = \mathbf{0} \Rightarrow \boxed{\nabla^2 f(\mathbf{x}^k)\mathbf{x}^{k+1} = \nabla^2 f(\mathbf{x}^k)\mathbf{x}^k - \nabla f(\mathbf{x}^k)}$$

System of n equations

Newton's Method

- Initialize: $\mathbf{x}^0 \in \mathbb{R}^m$
- Set k = 0
- While stopping condition not met
  - Solve the system $\nabla^2 f(\mathbf{x}^k)\mathbf{x}^{k+1} = \nabla^2 f(\mathbf{x}^k)\mathbf{x}^k - \nabla f(\mathbf{x}^k)$
  - k++

Newton's Method

- Suppose that the Hessian matrix is invertible

$$\nabla^2 f(\mathbf{x}^k)\mathbf{x}^{k+1} = \nabla^2 f(\mathbf{x}^k)\mathbf{x}^k - \nabla f(\mathbf{x}^k)$$

- Multiply both sides by its invertible

$$\mathbf{x}^{k+1} = \left(\nabla^2 f(\mathbf{x}^k)\right)^{-1}\nabla^2 f(\mathbf{x}^k)\mathbf{x}^k - \left(\nabla^2 f(\mathbf{x}^k)\right)^{-1}\nabla f(\mathbf{x}^k) = \mathbf{x}^k - \left(\nabla^2 f(\mathbf{x}^k)\right)^{-1}\nabla f(\mathbf{x}^k)$$

  - Similar to the gradient descent method where the step size is replaced by the inverse of Hessian matrix

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \left(\nabla^2 f(\mathbf{x}^k)\right)^{-1}\nabla f(\mathbf{x}^k)$$

Newton’s Method

- Question: What if the Hessian matrix is not invertible?
- Answer: Replace the inverse of Hessian matrix with what is called a pseudo-inverse computed using the singular value decomposition

$$\nabla^2 f(\mathbf{x}^k) = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

  - Columns of matrix $\mathbf{U}$ consist of left singular vectors of $\nabla^2 f(\mathbf{x}^k)$
  - Columns of matrix $\mathbf{V}$ consist of right singular vectors of $\nabla^2 f(\mathbf{x}^k)$
  - Diagonal elements of $\mathbf{D}$ are singular values of $\nabla^2 f(\mathbf{x}^k)$

$$\left(\nabla^2 f(\mathbf{x}^k)\right)^\dagger = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T \quad \Rightarrow \quad \mathbf{x}^{k+1} = \mathbf{x}^k - \left(\nabla^2 f(\mathbf{x}^k)\right)^\dagger \nabla f(\mathbf{x}^k)$$