Chapter 6: Derivative-Based Optimization

- Introduction (6.1)
- Descent Methods (6.2)
- The Method of Steepest Descent (6.3)
- Newton’s Methods (NM) (6.4)
- Step Size Determination (6.5)
- Nonlinear Least-Squares Problems (6.8)

Introduction (6.1)

- Goal: Solving minimization nonlinear problems through derivative information

- We cover:
  - Gradient based optimization techniques
  - Steepest descent methods
  - Newton Methods
  - Conjugate gradient methods
  - Nonlinear least-squares problems

- They are used in:
  - Optimization of nonlinear neuro-fuzzy models
  - Neural network learning
  - Regression analysis in nonlinear models
Descent methods (6.2)

- Goal: Determine a point $\theta = \theta^* = [\theta_1^*, \theta_2^*, ..., \theta_n^*]^T$ such that $f(\theta_1, \theta_2, ..., \theta_n)$ is minimum on $\theta = \theta^*$.

- We are looking for a local & not necessarily a global minimum $\theta^*$.

- Let $f(\theta_1, \theta_2, ..., \theta_n) = E(\theta_1, \theta_2, ..., \theta_n)$, the search of this minimum is performed through a certain direction $d$ starting from an initial value $\theta = \theta_0$ (iterative scheme!)

Descent Methods (6.2) (cont.)

$$\theta_{\text{next}} = \theta_{\text{now}} + \eta \, d$$

($\eta > 0$ is a step size regulating the search in the direction $d$)

$$\theta_{k+1} = \theta_k + \eta_k d_k \quad (k = 1, 2, ...$$

The series $\{\theta_k\}_{k=1,2,...}$ should converge to a local minimum $\theta^*$.

- We first need to determine the next direction $d$ & then compute the step size $\eta$.

- $\eta_k d_k$ is called the $k$-th step, whereas $\eta_k$ is the $k$-th step size.

- We should have $E(\theta_{\text{next}}) = E(\theta_{\text{now}} + \eta \, d) < E(\theta_{\text{now}})$.

- The principal differences between various descent algorithms lie in the first procedure for determining successive directions.
Descent Methods (6.2) (cont.)

- Once \( d \) is determined, \( \eta \) is computed as:

\[
\eta = \arg \min_{\eta > 0} \varnothing(\eta)
\]

where: \( \varnothing(\eta) = E(\theta_{\text{now}} + \eta d) \)

- Gradient-based methods

  - Definition: The gradient of a differentiable function \( E: \mathbb{R}^n \rightarrow \mathbb{R} \) at \( \theta \) is the vector of first derivatives of \( E \), denoted as \( g \). That is:

\[
g(\theta) = \nabla E(\theta) = \left[ \frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}, \ldots, \frac{\partial E(\theta)}{\partial \theta_n} \right]^T
\]

Descent Methods (6.2) (cont.)

- Based on a given gradient, downhill directions adhere to the following condition for feasible descent directions:

\[
\varnothing'(0) = \left. \frac{dE(\theta_{\text{now}} + \eta d)}{d\eta} \right|_{\eta=0} = g^T d = \left\| g \right\| \left\| d \right\| \cos(\xi(\theta_{\text{now}})) < 0
\]

Where \( \xi \) is the angle between \( g \) and \( d \) and \( \xi(\theta_{\text{now}}) \) is the angle between \( g_{\text{now}} \) and \( d \) at point \( \theta_{\text{now}} \)
Descent models (6.2) (cont.)

The previous equation is justified by Taylor series expansion:

\[ E(\theta_{\text{now}} + \eta d) = E(\theta_{\text{now}}) + \eta g^T d + O(\eta^2) \]
Descent Methods (6.2) (cont.)

- A class of gradient-based descent methods has the following form in which feasible descent directions can be found by gradient deflection.

- Gradient deflection consists of multiplying the gradient $g$ by a positive definite matrix (pdm) $G$
  \[ d = -Gg \Rightarrow g^T d = -g^T Gg < 0 \] (feasible descent direction)

- The gradient-based method is described therefore by:
  \[ \theta_{\text{next}} = \theta_{\text{now}} - \eta Gg \quad (\eta > 0, \, G \, \text{pdm}) \quad (*) \]

---

Descent Methods (6.2) (cont.)

- **Theoretically**, we wish to determine a value $\theta_{\text{next}}$ such as:
  \[ g(\theta_{\text{next}}) = \frac{\partial E(\theta)}{\partial \theta} \bigg|_{\theta=\theta_{\text{next}}} = 0 \]
  but this is difficult to solve!!

- **But practically**, we stop the algorithm if:
  - The objective function value is sufficiently small
  - The length of the gradient vector $g$ is smaller than a threshold
  - The computation time is exceeded

---

Ch. 6 [sections 6.1-6.5, 6.8]: Derivative-based optimization
The method of Steepest Descent (6.3)

- Despite its slow convergence, this method is the most frequently used nonlinear optimization technique due to its simplicity.

- If $G = I_d$ (identity matrix) then equation (*) expresses the steepest descent scheme:
  \[
  \theta_{\text{next}} = \theta_{\text{now}} - \eta g
  \]

- If $\cos \xi = -1$ (meaning that $d$ points to the same direction of vector $-g$) then the objective function $E$ can be decreased locally by the biggest amount at point $\theta_{\text{now}}$.

The method of Steepest Descent (6.3) (cont.)

- Therefore, the negative gradient direction ($-g$) points to the locally steepest downhill direction.

- This direction may not be a shortcut to reach the minimum point $\theta^*$.

- However, if the steepest descent uses the line minimization technique $(\min \, \varnothing(\eta))$ then $\varnothing'(\eta) = 0$

  \[
  \varnothing'(\eta) = \frac{dE(\theta_{\text{now}} - \eta g_{\text{now}})}{d\eta} = \nabla^T E(\theta_{\text{now}} - \eta g_{\text{now}}) g_{\text{now}}
  \]

  \[
  = g_{\text{next}}^T - g_{\text{now}} = 0
  \]

  $g_{\text{next}}$ is orthogonal to the current gradient vector $g_{\text{now}}$.

  (see figure 6.2; pt X)
The method of Steepest Descent (6.3) (cont.)

- If the contours of the objective function $E$ form hyperspheres (or circles in a 2-dimensional space), the steepest descent methods leads to the minimum in a single step. Otherwise the method does not lead to the minimum point.

Newton’s Methods (NM) (6.4)

- Classical NM
  - Principle: The descent direction $d$ is determined by using the second derivatives of the objective function $E$ if available.
  - If the starting position $\theta_{\text{now}}$ is sufficient close to a local minimum, the objective function $E$ can be approximated by a quadratic form:

$$E(\theta) \approx E(\theta_{\text{now}}) + g^T(\theta - \theta_{\text{now}}) + \frac{1}{2}(\theta - \theta_{\text{now}})^T H(\theta - \theta_{\text{now}})$$

where $H = \nabla^2 E(\theta) = \begin{pmatrix} \frac{\partial^2 E}{\partial^2 \theta} \end{pmatrix}$.
Newton’s Methods (NM) (6.4) (cont.)

- Since the equation defines a quadratic function \( E(\theta) \) in the \( \theta_{\text{now}} \) neighborhood \( \Rightarrow \) its minimum \( \hat{\theta} \) can be determined by differenting & setting to 0. Which gives:
  \[
  0 = g + H(\hat{\theta} - \theta_{\text{now}})
  \]
  Equivalent to:
  \[
  \hat{\theta} = \theta_{\text{now}} - H^{-1}g
  \]

- It is a gradient-based method for \( \eta = 1 \) and \( G = H^{-1} \)

Newton’s Methods (NM) (6.4) (cont.)

- Only when the minimum point \( \hat{\theta} \) of the approximated quadratic function is chosen as the next point \( \theta_{\text{next}} \), we have the so-called NM or the Newton-Raphson method
  \[
  \hat{\theta} = \theta_{\text{now}} - H^{-1}g
  \]
- If \( H \) is positive definite and \( E(\theta) \) is quadratic then the NM directly reaches a local minimum in the single Newton step (single \( - H^{-1}g \))
- If \( E(\theta) \) is not quadratic, then the minimum may nor be reached in a single step & NM should be iteratively repeated
Step Size Determination (6.5)

- Formula of a class of gradient-based descent methods:
  \[ \theta_{\text{next}} = \theta_{\text{now}} + \eta d = \theta_{\text{now}} - \eta \nabla \theta \]

- This formula entails effectively determining the step size \( \eta \)

- \( \nabla^2(\eta) = 0 \) with \( \nabla(\eta) = E(\theta_{\text{now}} + \eta d) \) is often impossible to solve
Step Size Determination (6.5) (cont.)

- **Initial Bracketing**
  
  - We assume that the search area (or specified interval) contains a single relative minimum: $E$ is unimodal over the closed interval
  
  - Determining the initial interval in which a relative minimum must lie is of critical importance
    
    - A scheme, by function evaluation for finding three points to satisfy:
      
      $$E(\theta_{k-1}) > E(\theta_k) < E(\theta_{k+1}); \ 0k-1 < 0k < 0k+1$$
    
    - A scheme, by taking the first derivative, for finding two points to satisfy:
      
      $$E'(\theta_{0k}) < 0, \ E'(\theta_{0k+1}) > 0, \ 0k < 0k+1$$

- **Algorithm for scheme 1:**
  
  An initial bracketing for searching three points $\theta_1$, $\theta_2$ and $\theta_3$

1) Given a starting point $\theta_0$ and $h \in \mathbb{R}$, let $\theta_1$ be $\theta_0 + h$.

   Evaluate $E(\theta_1)$
   
   if $E(\theta_0) \geq E(\theta_1)$, $i \leftarrow 1$
   
   (i.e., go downhill) go to (2)
   
   otherwise $h \leftarrow -h$ (i.e., set backward direction)
   
   $E(\theta_{i-1}) \leftarrow E(\theta_i)$
   
   $\theta_1 \leftarrow \theta_0 + h$
   
   $i \leftarrow 0$
   
   go to (3)

2) Set the next point by; $h \leftarrow 2h$, $\theta_{i+1} \leftarrow \theta_i + h$

3) Evaluate $E(\theta_{i+1})$

   if $E(\theta_i) \geq E(\theta_{i+1})$; $i \leftarrow i + 1$

   (i.e., still go downhill) go to (2)

   Otherwise, Arrange $\theta_{i+1}$, $\theta_i$ and $\theta_{i-1}$ in the decreasing order

   Then, we obtain the three points: $(\theta_1, \theta_2, \theta_3)$

   Stop.
Step Size Determination (6.5) (cont.)

- Line searches
  - The process of determining $\eta^*$ that minimizes a one-dimensional function $\phi(\eta)$ is achieved by searching on the line for the minimum
  - Line search algorithms usually include two components: sectioning (or bracketing), and polynomial interpolation
    - Newton’s method
      When $\phi(\eta_k)$, $\phi'(\eta_k)$, and $\phi''(\eta_k)$ are available, the classical Newton method (defined by $\hat{\theta} = \theta_{\text{norm}} - H^{-1}$) can be applied to solving the equation $\phi'(\eta_k) = 0$:
      \[
      \eta_{k+1} = \eta_k - \frac{\phi'(\eta_k)}{\phi''(\eta_k)} \quad (*)
      \]

Step Size Determination (6.5) (cont.)

- Secant method
  If we use both $\eta_k$ and $\eta_{k-1}$ to approximate the second derivative in equation $(*)$, and if the first derivatives alone are available then we have an estimated $\eta_{k+1}$ defined as:
  \[
  \eta_{k+1} = \eta_k - \frac{\phi'(\eta_k)}{\phi'(\eta_k) - \phi'(\eta_{k-1})} \eta_k - \eta_{k-1}
  \]
  this method is called the secant method.
  - Both the Newton’s and the secant method are illustrated in the following figure.
Newton’s method and secant method to determine the step size

Step Size Determination (6.5) (cont.)

- Sectioning methods
  - It starts with an interval \([a_1, b_1]\) in which the minimum \(\eta^*\) must lie, and then reduces the length of the interval at each iteration by evaluating the value of \(\varnothing\) at a certain number of points
  - The two endpoints \(a_1\) and \(b_1\) can be found by the initial bracketing described previously
  - The bisection method is one of the simplest sectioning method for solving \(\varnothing'(\eta^*) = 0\), if first derivatives are available!
Let $\varnothing'(\eta) = \varphi(\eta)$ then the algorithm is:

Algorithm [bisection method]
(1) Given $\varepsilon \in \mathbb{R}^+$ and an initial interval with 2 endpoints $a_1$ and $a_2$ such that: $a_1 < a_2$ and $\varphi(a_1)\varphi(a_2) < 0$ then set:

$\eta_{\text{left}} \leftarrow a_1$

$\eta_{\text{right}} \leftarrow a_2$

(2) Compute the midpoint $\eta_{\text{mid}}$: $\eta_{\text{mid}} \leftarrow (\eta_{\text{right}} + \eta_{\text{left}}) / 2$

if $\varphi(\eta_{\text{right}})\varphi(\eta_{\text{mid}}) < 0$, $\eta_{\text{left}} \leftarrow \eta_{\text{mid}}$

Otherwise $\eta_{\text{right}} \leftarrow \eta_{\text{mid}}$

(3) Check if $|\eta_{\text{left}} - \eta_{\text{right}}| < \varepsilon$. If it is true then terminate the algorithm, otherwise go to (2)

---

Step Size Determination (6.5) (cont.)

- Golden section search method

This method does not require $\varnothing$ to be differentiable. Given an initial interval $[a_1, b_1]$ that contains $\eta$, the next trial points $(s_k, t_k)$ within the interval are determined by using the golden section ratio $\tau$:

$s_k = b_k - \frac{1}{\tau}(b_k - a_k) = b_k + \frac{\tau - 1}{\tau}(b_k - a_k)$

$t_k = a_k + \frac{1}{\tau}(b_k - a_k)$

where $\tau = \frac{1 + \sqrt{5}}{2} \approx 1.618$
Step Size Determination (6.5) (cont.)

This procedure guarantees the following:
\[ a_k < s_k < t_k < b_k \]
The algorithm generates a sequence of two endpoints \( a_k \) and \( b_k \), according to:

\[
\begin{align*}
\text{If } &\, \emptyset(s_k) > \emptyset(t_k), \quad a_{k+1} = s_k, \quad b_{k+1} = b_k \\
\text{Otherwise} &\quad a_{k+1} = a_k, \quad b_{k+1} = t_k
\end{align*}
\]

The minimum point is bracketed to an interval just \( \frac{2}{3} \) times the length of the preceding interval

Golden section search to determine the step length
Step Size Determination (6.5) (cont.)

- Line searches (cont.)
  - Polynomial interpolation
    - This method is based on curve-fitting procedures
    - A quadratic interpolation is the method that is very often used in practice
    - It constructs a smooth quadratic curve \( q \) that passes through three points \((\eta_1, \varphi_1), (\eta_2, \varphi_2)\) and \((\eta_3, \varphi_3)\):

\[
q(\eta) = \sum_{i=1}^{3} \varphi_i \frac{\prod_{j \neq i}(\eta - \eta_j)}{\prod_{i \neq j}(\eta_i - \eta_j)}
\]

where \( \varphi_i = \varphi(\eta_i), i = 1, 2, 3 \)

Step Size Determination (6.5) (cont.)

- Polynomial interpolation (cont.)
  - Condition for obtaining a unique minimum point is:
    \( q'(\eta) = 0 \), therefore the next point \( \eta_{\text{next}} \) is:

\[
\eta_{\text{next}} = \frac{1}{2} \frac{(\eta_2^3 - \eta_3^3)\varphi_1 + (\eta_3^3 - \eta_1^3)\varphi_2 + (\eta_1^3 - \eta_2^3)\varphi_3}{(\eta_2 - \eta_3)\varphi_1 + (\eta_3 - \eta_1)\varphi_2 + (\eta_1 - \eta_2)\varphi_3}
\]
Step Size Determination (6.5) (cont.)

- Termination rules
  - Line search methods do not provide the exact minimum point of the function $\varnothing$
  - We need a termination rule that accelerate the entire minimization process without affecting too much precision
Step Size Determination (6.5) (cont.)

- Termination rules (cont.)
  - The Goldstein Test
    - This method is based on two definitions:
      - A value of $\eta$ is not too large if with a given $\mu$ ($0 < \mu < \frac{1}{2}$),
        \[
        \emptyset(\eta) \leq \emptyset(0) + \mu \emptyset'(0)\eta
        \]
      - A value of $\eta$ is considered to be not too small if:
        \[
        \emptyset(\eta) > \emptyset(0) + (1 - \mu) \emptyset'(\eta)
        \]

Goldstein test (cont.)

- From the two precedent inequalities, we obtain:
  \[
  (1 - \mu) \emptyset'(0)\eta \leq \emptyset(\eta) - \emptyset(0) = E(\theta_{\text{next}}) - E(\theta_{\text{now}}) \\
  \leq \mu \emptyset'(0)\eta
  \]

  which can be written as:
  \[
  0 < \mu \leq \frac{E(\theta_{\text{next}}) - E(\theta_{\text{now}})}{\eta g d} \leq 1 - \mu < 1
  \]

  where $\emptyset'(0) = g T d < 0$ (Taylor series)
Goal: Optimize a model by minimizing a squared error measure between desired outputs & the model’s output

\[ y = f(x, \theta) \]

Given a set of \( m \) training data pairs \((x_p, t_p)\), \((p = 1, ..., m)\), we can write:

\[
E(\theta) = \sum_{p=1}^{m} (t_p - y_p)^2 = \sum_{p=1}^{m} (t_p - f(x_p, \theta))^2
\]

\[
= \sum_{p=1}^{m} r_p(\theta)^2 = r^T(\theta) \cdot r(\theta)
\]
Nonlinear Least-Squares Problems (6.8) (cont.)

- The gradient is expressed as:
  \[ g = g(\theta) = \frac{\partial E(\theta)}{\partial \theta} = 2 \sum_{p=1}^{m} r_p(\theta) \frac{\partial r_p(\theta)}{\partial \theta} = 2J^T \cdot r \]

where \( J \) is the Jacobian matrix of \( r \).

\[
\left( \begin{array}{c} r, \theta \\ \phi \end{array} \rightarrow (r \cos \theta, r \sin \theta) \right) \quad J_\phi = r
\]

Since \( r_p(\theta) = t_p - f(x_p, \theta) \), this implies that the \( p \)th row of \( J \) is:

\[
-\nabla_\theta f(x_p, \theta)
\]

Nonlinear Least-Squares Problems (6.8) (cont.)

- Gauss-Newton Method

  - Known also as the linearization method

  - Use Taylor series expansion to obtain a linear model that approximates the original nonlinear model

  - Use linear least-squares optimization of chapter 5 to obtain the model parameters
Nonlinear Least-Squares Problems (6.8) (cont.)

- Gauss-Newton Method (cont.)

  - The parameters $\theta^T = (\theta_1, \theta_2, \ldots, \theta_n, \ldots)$ will be computed iteratively.

  - Taylor expansion of $y = f(x, \theta)$ around $\theta = \theta_{\text{now}}$

$$y = f(x, \theta_{\text{now}}) + \sum_{i=1}^{n} \left( \frac{\partial f(x, \theta)}{\partial \theta_i} \bigg|_{\theta=\theta_{\text{now}}} \right) (\theta_i - \theta_{i,\text{now}})$$

Nonlinear Least-Squares Problems (6.8) (cont.)

- Gauss-Newton Method (cont.)

  - $y - f(x, \theta_{\text{now}})$ is linear with respect to $\theta_i - \theta_{i,\text{now}}$ since the partial derivatives are constant

$$E(\theta) = \left( t - f(x, \theta_{\text{now}}) - \frac{\partial f(x, \theta_{\text{now}})}{\partial \theta} (\theta - \theta_{\text{now}}) \right)^2$$

$$= \left\| r + J^T (\theta - \theta_{\text{now}}) \right\|^2 = \left\| r + J^T S \right\|^2$$

where $S = \theta - \theta_{\text{now}}$
Nonlinear Least-Squares Problems (6.8) (cont.)

- Gauss-Newton Method (cont.)

  - The next point $\theta_{next}$ is obtained by:

    \[
    \frac{\partial E(\theta)}{\partial \theta} \bigg|_{\theta=\theta_{next}} = J^T \{ r + J (\theta_{next} - \theta_{now}) \} = 0
    \]

  - Therefore, the following Gauss-Newton formula is expressed as:

    \[
    \theta_{next} = \theta_{now} - (J^TJ)^{-1} J^T r = \theta_{now} - \frac{1}{2} (J^TJ)^{-1} g
    \]

    (since $g = 2J^T r$)