13. Unconstrained minimization

- gradient and Hessian
- Newton’s method
Unconstrained minimization problem

minimize \( g(x_1, x_2, \ldots, x_n) \)

\( g : \mathbb{R}^n \rightarrow \mathbb{R} \) (a function that maps \( n \)-vectors to scalars)

- \( x = (x_1, x_2, \ldots, x_n) \) is \( n \)-vector of optimization variables
- \( g(x) \) is the cost function or objective function
- to solve a maximization problem (i.e., maximize \( g(x) \)), minimize \(-g(x)\)
Local and global optimum

- $x^*$ is an optimal point (or a minimum) if

$$g(x^*) \leq g(x) \quad \text{for all } x$$

also called globally optimal

- $x^*$ is a locally optimal point (local minimum) if for some $R > 0$

$$g(x^*) \leq g(x) \quad \text{for all } x \text{ with } \|x - x^*\| \leq R$$

Example

$y$ is locally optimal

$z$ is (globally) optimal
Gradient

**Gradient** of \( g : \mathbb{R}^n \to \mathbb{R} \) at \( z \in \mathbb{R}^n \) is

\[
\nabla g(z) = \left( \frac{\partial g}{\partial x_1}(z), \frac{\partial g}{\partial x_2}(z), \ldots, \frac{\partial g}{\partial x_n}(z) \right)
\]

- \( \nabla g(z)^T = Dg(z) \) is the derivative matrix of \( g \) at \( z \)
- special case (\( n = 1 \)): \( \nabla g(z) = g'(z) \)

**Affine (first order) approximation** of \( g \) around \( z \) is

\[
\hat{g}(x) = g(z) + \frac{\partial g}{\partial x_1}(z)(x_1 - z_1) + \cdots + \frac{\partial g}{\partial x_n}(z)(x_n - z_n)
\]

\[
= g(z) + \nabla g(z)^T(x - z)
\]
Directional derivative

the *directional derivative* of $g$ at $x$, in the direction $v$, is

$$\nabla g(x)^T v$$

$v$ is a *descent direction* if $\nabla g(x)^T v < 0$

**Interpretation:** define function $h : \mathbb{R} \rightarrow \mathbb{R}$ with $h(t) = g(x + tv)$

- $h$ gives value of $g$ on the line through $x$ in direction $v$
- derivative of $h$ is

$$h'(t) = \frac{\partial g}{\partial x_1}(x + tv)v_1 + \frac{\partial g}{\partial x_2}(x + tv)v_2 + \cdots + \frac{\partial g}{\partial x_n}(x + tv)v_n$$

$$= \nabla g(x + tv)^T v$$

- derivative at $t = 0$ is $h'(0) = \nabla g(x)^T v$
Hessian

Hessian of $g$ at $z$: an $n \times n$ matrix $\nabla^2 g(z)$ with

$$\nabla^2 g(z)_{ij} = \frac{\partial^2 g}{\partial x_i \partial x_j}(z)$$

• a symmetric matrix
• the derivative matrix $Df(x)$ of $f(x) = \nabla g(x)$ at $z$
• special case ($n = 1$): $\nabla^2 g(z) = g''(z)$

Quadratic (second order) approximation of $g$ around $z$:

$$g_q(x) = g(z) + \nabla g(z)^T(x - z) + \frac{1}{2}(x - z)^T\nabla^2 g(z)(x - z)$$
Examples

• affine function \( g(x) = a^T x + b \)

\[
\nabla g(x) = a, \quad \nabla^2 g(x) = 0
\]

• quadratic function \( g(x) = x^T P x + q^T x + r \) with \( P \) symmetric

\[
\nabla g(x) = 2Px + q, \quad \nabla^2 g(x) = 2P
\]

• least squares cost: \( g(x) = \|Ax - b\|^2 = x^T A^T Ax - 2b^T Ax + b^T b \)

\[
\nabla g(x) = 2A^T Ax - 2A^T b, \quad \nabla^2 g(x) = 2A^T A
\]
Properties

Linear combination: if \( g(x) = \alpha_1 g_1(x) + \alpha_2 g_2(x) \), then

\[
\nabla g(x) = \alpha_1 \nabla g_1(x) + \alpha_2 \nabla g_2(x)
\]

\[
\nabla^2 g(x) = \alpha_1 \nabla^2 g_1(x) + \alpha_2 \nabla^2 g_2(x)
\]

Composition with affine mapping: if \( g(x) = h(Cx + d) \), then

\[
\nabla g(x) = C^T \nabla h(Cx + d)
\]

\[
\nabla^2 g(x) = C^T \nabla^2 h(Cx + d)C
\]
Example

\[ g(x_1, x_2) = e^{x_1+x_2-1} + e^{x_1-x_2-1} + e^{-x_1-1} \]

Gradient

\[ \nabla g(x) = \begin{bmatrix} e^{x_1+x_2-1} + e^{x_1-x_2-1} - e^{-x_1-1} \\ e^{x_1+x_2-1} - e^{x_1-x_2-1} \end{bmatrix} \]

Hessian

\[ \nabla^2 g(x) = \begin{bmatrix} e^{x_1+x_2-1} + e^{x_1-x_2-1} + e^{-x_1-1} & e^{x_1+x_2-1} - e^{x_1-x_2-1} \\ e^{x_1+x_2-1} - e^{x_1-x_2-1} & e^{x_1+x_2-1} + e^{x_1-x_2-1} \end{bmatrix} \]
Gradient and Hessian via composition property

express $g$ as $g(x) = h(Cx + d)$ with $h(y_1, y_2, y_3) = e^{y_1} + e^{y_2} + e^{y_3}$ and

$$C = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 0 \end{bmatrix}, \quad d = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

**Gradient:** $\nabla g(x) = C^T \nabla h(Cx + d)$

$$\nabla g(x) = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} e^{x_1+x_2-1} \\ e^{x_1-x_2-1} \\ e^{-x_1-1} \end{bmatrix}$$

**Hessian:** $\nabla^2 g(x) = C^T \nabla^2 h(Cx + d)C$

$$\nabla^2 g(x) = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} e^{x_1+x_2-1} & 0 & 0 \\ 0 & e^{x_1-x_2-1} & 0 \\ 0 & 0 & e^{-x_1-1} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 0 \end{bmatrix}$$
Optimality conditions for twice differentiable $g$

**Necessary condition**: if $x^*$ is locally optimal, then

$$\nabla g(x^*) = 0 \quad \text{and} \quad \nabla^2 g(x^*) \text{ is positive semidefinite}$$

**Sufficient condition**: if $x^*$ satisfies

$$\nabla g(x^*) = 0 \quad \text{and} \quad \nabla^2 g(x^*) \text{ is positive definite}$$

then $x^*$ is locally optimal

**Necessary and sufficient condition for convex functions**

- $g$ is called *convex* if $\nabla^2 g(x)$ is positive semidefinite everywhere
- if $g$ is convex then $x^*$ is optimal if and only if $\nabla g(x^*) = 0$
Examples ($n = 1$)

• $g(x) = \log(e^x + e^{-x})$

\[
\begin{align*}
  g'(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}}, \\
  g''(x) &= \frac{4}{(e^x + e^{-x})^2}
\end{align*}
\]

$g''(x) \geq 0$ everywhere; $x^* = 0$ is the unique optimal point

• $g(x) = x^4$

\[
\begin{align*}
  g'(x) &= 4x^3, \\
  g''(x) &= 12x^2
\end{align*}
\]

$g''(x) \geq 0$ everywhere; $x^* = 0$ is the unique optimal point

• $g(x) = x^3$

\[
\begin{align*}
  g'(x) &= 3x^2, \\
  g''(x) &= 6x
\end{align*}
\]

$g'(0) = 0, g''(0) = 0$ but $x = 0$ is not locally optimal
Examples

- \( g(x) = x^T P x + q^T x + r \) (\( P \) is positive definite)

\[
\nabla g(x) = 2P x + q, \quad \nabla^2 g(x) = 2P
\]

\( \nabla^2 g(x) \) is positive definite everywhere, hence the unique optimal point is

\[
x^* = -(1/2)P^{-1}q
\]

- \( g(x) = \|Ax - b\|^2 \) (\( A \) is a matrix with linearly independent columns)

\[
\nabla g(x) = 2A^T Ax - 2A^T b, \quad \nabla^2 g(x) = 2A^T A
\]

\( \nabla^2 g(x) \) is positive definite everywhere, hence the unique optimal point is

\[
x^* = (A^T A)^{-1} A^T b
\]
Examples

element of page 13-9: we can express $\nabla^2 g(x)$ as

$$\nabla^2 g(x) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} e^{x_1+x_2-1} & 0 & 0 \\ 0 & e^{x_1-x_2-1} & 0 \\ 0 & 0 & e^{-x_1-1} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 0 \end{bmatrix}$$

this shows that $\nabla^2 g(x)$ is positive definite for all $x$

therefore $x^*$ is optimal if and only if

$$\nabla g(x^*) = \begin{bmatrix} e^{x^*_1+x^*_2-1} + e^{x^*_1-x^*_2-1} - e^{-x^*_1-1} \\ e^{x^*_1+x^*_2-1} - e^{x^*_1-x^*_2-1} \end{bmatrix} = 0$$

two nonlinear equations in two variables
Newton’s method for minimizing a convex function

If \( \nabla^2 g(x) \) is positive definite everywhere, we can minimize \( g(x) \) by solving

\[
\nabla g(x) = 0
\]

**Algorithm:** choose \( x^{(0)} \) and repeat for \( k = 0, 1, 2, \ldots \)

\[
x^{(k+1)} = x^{(k)} - \nabla^2 g(x^{(k)})^{-1} \nabla g(x^{(k)})
\]

- \( v = -\nabla^2 g(x)^{-1} \nabla g(x) \) is called the *Newton step* at \( x \)
- converges if started sufficiently close to the solution
- Newton step computed by a Cholesky factorization of the Hessian
Interpretations of Newton step

Affine approximation of gradient

- affine approximation of $f(y) = \nabla g(y)$ around $x$ is

$$\hat{f}(y) = \nabla g(x) + \nabla^2 g(x)(y - x)$$

- Newton update $x + v$ is solution of linear equation $\hat{f}(y) = 0$

Quadratic approximation of function

- quadratic approximation of $g(y)$ around $x$ is

$$g_q(y) = g(x) + \nabla g(x)^T(y - x) + \frac{1}{2}(y - x)^T \nabla^2 g(x)(y - x)$$

- Newton update $x + v$ is minimizer of $g_q$ (solution of $\nabla g_q(y) = 0$)
Example \((n = 1)\)

\[
g_q(y) = g(x) + g'(x)(y - x) + \frac{g''(x)}{2}(y - x)^2
\]
Example

\[ g(x) = \log(e^x + e^{-x}), \quad g'(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad g''(x) = \frac{4}{(e^x + e^{-x})^2} \]

\[ g(x) \]

\[ g'(x) \]

\[ g''(x) \]

does not converge when started at \( x = 1.15 \)
Newton method with line search

use update \( x + tv \) and choose \( t \) so that \( g(x + tv) < g(x) \)

**Algorithm:** choose \( x^{(0)} \) and repeat for \( k = 0, 1, 2, \ldots \)

- compute Newton step \( v = -\nabla^2 g(x^{(k)})^{-1} \nabla g(x^{(k)}) \)
- find largest \( t \) in \( \{1, 0.5, 0.5^2, 0.5^3, \ldots \} \) that satisfies

\[
g(x^{(k)} + tv) \leq g(x^{(k)}) + \alpha t \nabla g(x^{(k)})^T v
\]

and take \( x^{(k+1)} = x^{(k)} + tv \)

- \( \alpha \) is an algorithm parameter (small and positive, e.g., \( \alpha = 0.01 \))
- \( t \) is called the *step size*
Interpretation of line search

to determine a suitable step size, consider the function \( h : \mathbb{R} \rightarrow \mathbb{R} \)

\[
h(t) = g(x + tv)
\]

\( x = x^{(k)} \) is the current iterate; \( v \) is the Newton step at \( x \)

- \( h'(0) = \nabla g(x)^T v \) is directional derivative of \( g \) at \( x \) in direction \( v \)
- affine approximation of \( h \) at \( t = 0 \) is

\[
\hat{h}(t) = h(0) + h'(0)t = g(x) + t\nabla g(x)^T v
\]

- line search accepts step size \( t \) if \( g(x + tv) \leq g(x) + \alpha t\nabla g(x)^T v \); i.e.,

\[
h(t) - h(0) \leq \alpha(\hat{h}(t) - h(0))
\]

decrease \( h(t) - h(0) \) is at least \( \alpha \) times what is expected based on \( \hat{h} \)
Interpretation of line search

start with $t = 1$; divide $t$ by 2 until $h(t) \leq h(0) + \alpha h'(0)t$

- works if $h'(0) = \nabla g(x)^Tv < 0$ ($v$ is a descent direction)
- if $\nabla^2 g(x)$ is positive definite, the Newton step is a descent direction

$$h'(0) = \nabla g(x)^Tv = v^T\nabla^2 g(x)v < 0$$
Example

\[ g(x) = \log(e^x + e^{-x}), \quad x^{(0)} = 4 \]

close to the solution: very fast convergence, no backtracking steps
Example

equation of page 13-9

\[ g(x_1, x_2) = e^{x_1 + x_2 - 1} + e^{x_1 - x_2 - 1} + e^{-x_1 - 1} \]

Newton method with line search started at \( x = (-2, 2) \)
Newton method for nonconvex functions

if $\nabla^2 g(x)$ is not positive definite, it is possible that Newton step $v$ satisfies

$$\nabla g(x)^T v = -\nabla g(x)^T \nabla^2 g(x)^{-1} \nabla g(x) < 0$$

- if Newton step is not descent direction, replace it with descent direction
- simplest choice is $v = -\nabla g(x)$; practical methods make other choices