Ch 7. Continuous Optimization

Find $\min_x f(x)$, minimum value of $f(x)$

- Unconstrained Optimization(Optimization using Gradient Descent)
  - 1. Gradient Descent
  - 2. Gradient Descent with Momentum
  - 3. Stochastic Gradient Descent SGD
- Constrained Optimization
  - 1. Linear Programming
  - 2. Quadratic Programming
  - 3. Convex set - Convex Optimization

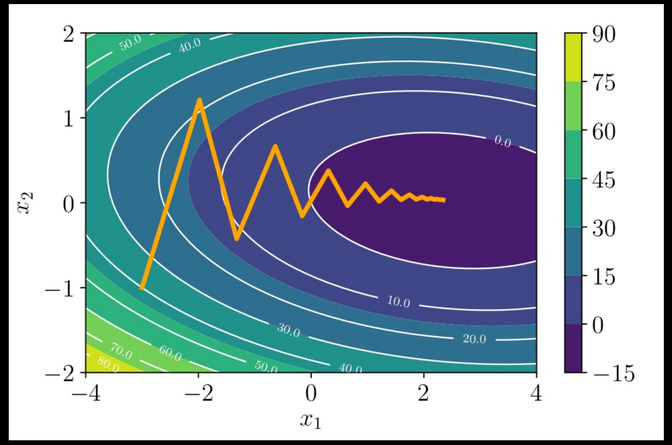
한지수

Optimization using Gradient Descent

Unconstrained Optimization

- 1. Gradient Descent & Step-size/Learning rate
- 2. Gradient Descent with Momentum
- 3. Stochastic Gradient Descent SGD & Standard Gradient Descent

Optimization using Gradient Descent



- 1. Gradient Descent
  - $x_1 = x_0 - \gamma((\nabla f)(x_0))^\top$ for small step size γ
  - $\implies x_{i+1} = x_i - \gamma_i((\nabla f)(x_i))^\top$
  - Relatively slow close to the minimum, increasingly zigzags as the near point-oscillation problem
- Step-size/Learning rate
  - Too small - slow, too large - fail to converge, diverge
  - Speed of convergence of gradient-descent > Condition number $\kappa = \frac{\sigma(A)_{max}}{\sigma(B)_{max}}$

Optimization using Gradient Descent

- 2. Gradient descent with Momentum
  - An additional term to remember what happened in the previous iteration
    - Dampen oscillation, smoothes out the gradient updates
  - $\implies x_{i+1} = x_i - \gamma_i((\nabla f)(x_0))^\top + \alpha \Delta x_i$
    - $\Delta x_i = x_i - x_{i-1} = \alpha \Delta x_{i-1} - \gamma_{i-1}((\nabla f)(x_{i-1}))^\top$
  - Momentum term averages out different noisy estimates of the gradient.

Optimization using Gradient Descent

- 3. Stochastic Gradient Descent SGD
  - By constraining the probability distribution of the approximate gradients
  - $\theta_{i+1} = \theta_i - \gamma_i(\Delta L(\theta_i))^\top$
    - Loss $L(\theta) = \sum_{n=1}^{N} L_n(\theta) = -\sum_{n=1}^{N} logp(y_n \mid x_n, \theta)$
- Standard Gradient Descent/Batch Optimization
  - When training set is enormous and/or no simple formulas exist -> evaluation is expensive
  - Mini-batch GD: Randomly choose a single L_n to estimate the gradient

Constrained Optimization

- 1. Linear Programming
- 2. Quadratic Programming
- 3. Legendre-Fenchel Transform and Convex Conjugate + Convex Set

Constrained Optimization

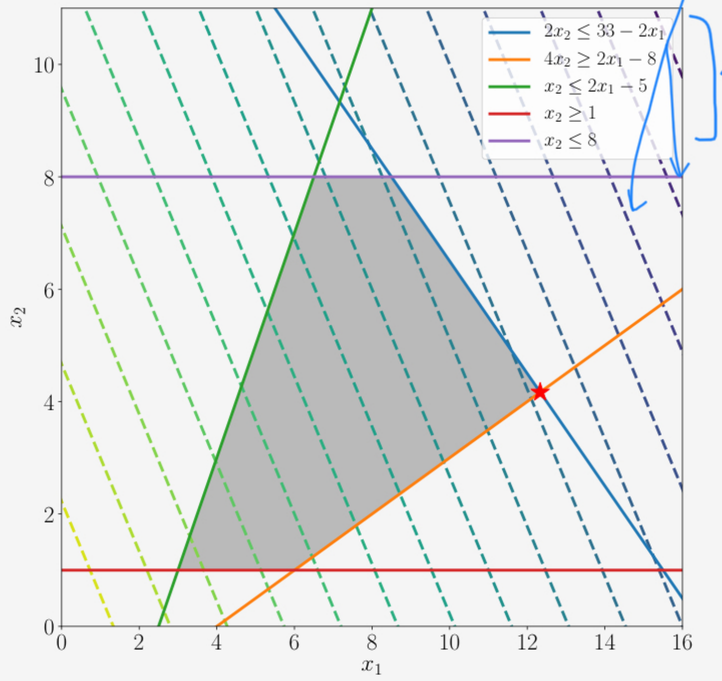
- 1. Linear Programming
  - $\min_{x \in R^d} c^\top x$ subject to $Ax \leq b$
  - $\mathscr{L}(x, \lambda) = f(x) + \lambda^\top g(x) = c^\top x + \lambda^\top (Ax - b)$
  - $\mathscr{L}(x, \lambda) = (c + A^\top \lambda)^\top x - \lambda^\top b$
  - Derivative $\Rightarrow c + A^\top \lambda = 0$
  - $\Rightarrow \mathscr{D}(\lambda) = -\lambda^\top b$
- Dual Optimization Problem
  - $\min_{\lambda \in R^m} -b^T \lambda$ (subject to $c + A^\top \lambda = 0, \lambda \geq 0$)

Linear Programming

Example 7.5 (Linear Program)
Consider the linear program

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^2} \quad & -\begin{bmatrix}5\\3\end{bmatrix}^\top \begin{bmatrix}x_1\\x_2\end{bmatrix} \\ \text{subject to} \quad & \begin{bmatrix}2 & 2\\ 2 & -4\\ -2 & 1\\ 0 & -1\\ 0 & 1\end{bmatrix} \begin{bmatrix}x_1\\x_2\end{bmatrix} \leqslant \begin{bmatrix}33\\8\\5\\-1\\8\end{bmatrix} \end{aligned} \tag{7.44}$$

with two variables. This program is also shown in Figure 7.9. The objective function is linear, resulting in linear contour lines. The constraint set in standard form is translated into the legend. The optimal value must lie in the shaded (feasible) region, and is indicated by the star.



Constrained Optimization

- 2. Quadratic Programming
  - $\min_{x \in R^d} \frac{1}{2} x^\top Q x + c^\top x$ subject to $Ax \leq b$
  - $\mathscr{L}(x, \lambda) = \frac{1}{2} x^\top Q x + c^\top x + \lambda^\top (Ax - b)$
  - Derivative $\Rightarrow Qx + (c + A^\top \lambda) = 0$
    - $\Rightarrow x = -Q^{-1}(c + A^\top \lambda)$
  - $\Rightarrow \mathscr{D}(\lambda) = -\frac{1}{2}(c + A^\top \lambda)^\top Q^{-1}(c + A^\top \lambda) - \lambda^\top b$
- Dual Optimization Problem
  - $\min_{\lambda \in R^m} -\frac{1}{2}(c + A^\top \lambda)^\top Q^{-1}(c + A^\top \lambda) - \lambda^\top b \lambda$ (subject to $\lambda \geq 0$)

Quadratic Programming

Example 7.6 (Quadratic Program)
Consider the quadratic program

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \quad \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^\top \begin{bmatrix}2 & 1\\1 & 4\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} + \begin{bmatrix}5\\3\end{bmatrix}^\top \begin{bmatrix}x_1\\x_2\end{bmatrix} \tag{7.46}$$

$$\text{subject to} \quad \begin{bmatrix}1 & 0\\-1 & 0\\0 & 1\\0 & -1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} \leqslant \begin{bmatrix}1\\1\\1\\1\end{bmatrix} \tag{7.47}$$

of two variables. The program is also illustrated in Figure 7.4. The objective function is quadratic with a positive semidefinite matrix $\boldsymbol{Q}$, resulting in elliptical contour lines. The optimal value must lie in the shaded (feasible) region, and is indicated by the star.

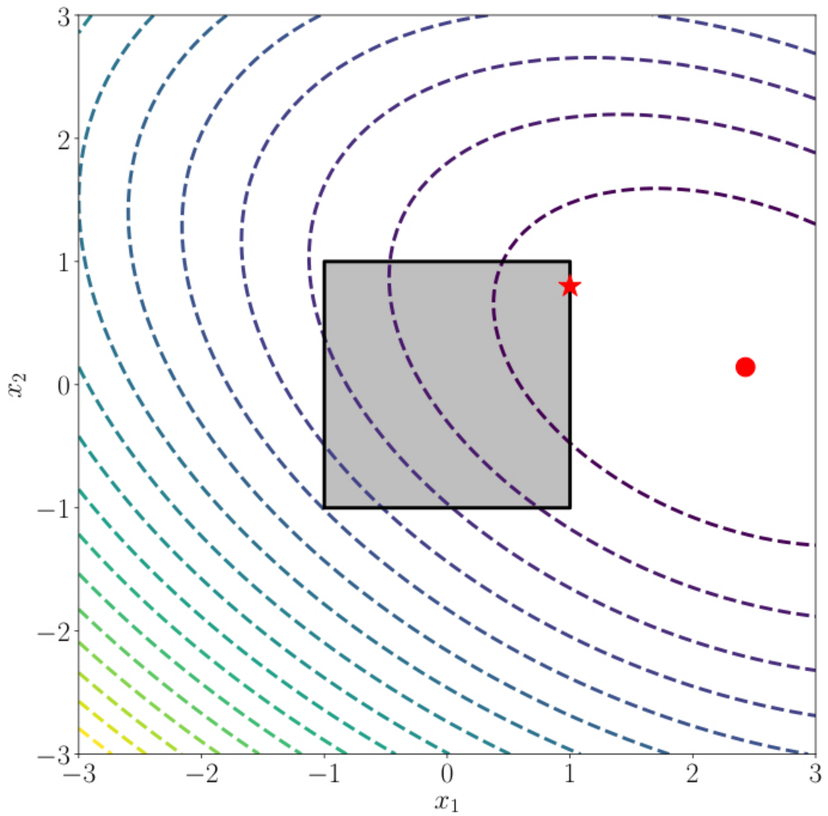


Figure 7.4
Illustration of constrained optimization. The unconstrained problem (indicated by the contour lines) has a minimum on the right side (indicated by the circle). The box constraints ($-1 \leqslant x \leqslant 1$ and $-1 \leqslant y \leqslant 1$) require that the optimal solution is within the box, resulting in an optimal value indicated by the star.

Constrained Optimization - Convex Set

- Convex set
  - Set C for any $x, y \in C, 0 \leq \theta \leq 1 : \theta x + (1 - \theta)y \in C$
  - Convex function: $f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$
  - -> described by its supporting hyperplanes
  - -> described by a function of their gradients : Legendre transform/Convex conjugate
- Convex Optimization Problem
  - $\min_x f(x)$ subject to $g_i(x) \leq 0$ for all $i = 1...m$, $h_j(x) = 0$ for all $j = 1...n$
    - f,g are convex functions, h are convex sets

Constrained Optimization

- 3. Legendre-Fenchel transform / Convex conjugate
  - $f^*(x) = \sup_{x \in R^D} (< s, x > - f(x))$

Example 7.7 (Convex Conjugates)
To illustrate the application of convex conjugates, consider the quadratic function

$$f(\boldsymbol{y}) = \frac{\lambda}{2}\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} \tag{7.59}$$

based on a positive definite matrix $\boldsymbol{K} \in \mathbb{R}^{n\times n}$. We denote the primal variable to be $\boldsymbol{y} \in \mathbb{R}^n$ and the dual variable to be $\boldsymbol{\alpha} \in \mathbb{R}^n$.

Applying Definition 7.4, we obtain the function

$$f^*(\boldsymbol{\alpha}) = \sup_{\boldsymbol{y}\in\mathbb{R}^n} \langle \boldsymbol{y}, \boldsymbol{\alpha}\rangle - \frac{\lambda}{2}\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y}\,. \tag{7.60}$$

Since the function is differentiable, we can find the maximum by taking the derivative and with respect to $\boldsymbol{y}$ setting it to zero.

$$\frac{\partial\left[\langle \boldsymbol{y}, \boldsymbol{\alpha}\rangle - \frac{\lambda}{2}\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y}\right]}{\partial \boldsymbol{y}} = (\boldsymbol{\alpha} - \lambda \boldsymbol{K}^{-1}\boldsymbol{y})^\top \tag{7.61}$$

and hence when the gradient is zero we have $\boldsymbol{y} = \frac{1}{\lambda}\boldsymbol{K}\boldsymbol{\alpha}$. Substituting into (7.60) yields

$$f^*(\boldsymbol{\alpha}) = \frac{1}{\lambda}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} - \frac{\lambda}{2}\left(\frac{1}{\lambda}\boldsymbol{K}\boldsymbol{\alpha}\right)^\top \boldsymbol{K}^{-1}\left(\frac{1}{\lambda}\boldsymbol{K}\boldsymbol{\alpha}\right) = \frac{1}{2\lambda}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha}\,. \tag{7.62}$$