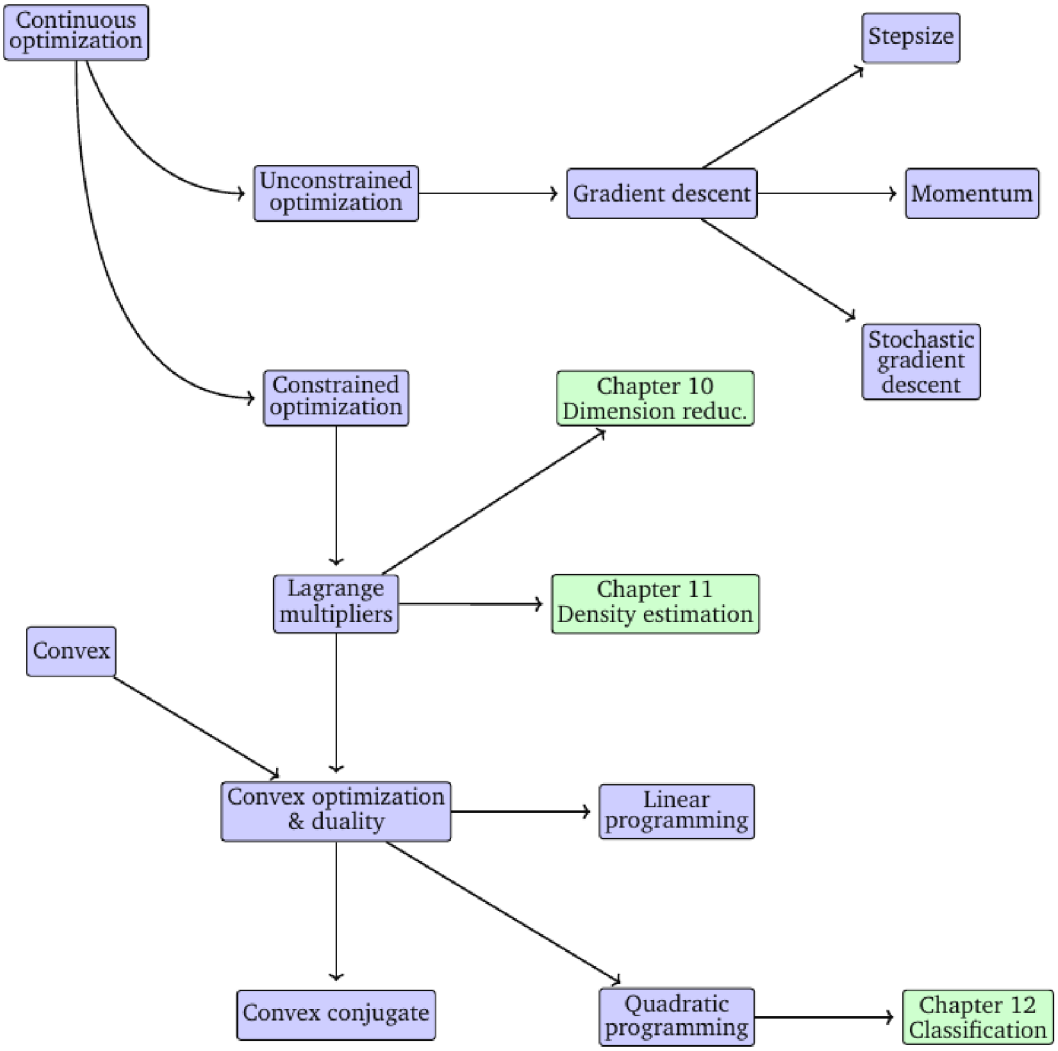
Part 7 - Continuous Optimization



7.1 Optimization menggunakan Gradient Descent

Gradient descent adalah metode optimisasi matematis tanpa batasan (*unconstrained*) yang merupakan algoritma iteratif orde pertama untuk menemukan nilai minimum dari sebuah fungsi multivariabel yang dapat diturunkan (*differentiable*). Metode ini sangat berguna dalam *machine learning* untuk meminimalkan fungsi biaya atau kerugian (*cost or loss function*).

Prinsip dasar di balik *gradient descent* adalah observasi bahwa jika sebuah fungsi multivariabel $f(x)$ terdefinisi dan dapat diturunkan di sekitar titik a, maka nilai $f(x)$ akan menurun paling cepat jika kita bergerak dari titik a ke arah gradien negatif dari f di titik a, yaitu $-\nabla f(a)$. Arah ini dikenal sebagai arah penurunan paling curam (*steepest descent*).

Gradient Descent

Jika kita ingin menemukan minimum lokal $f(x_*)$ dari suatu fungsi $f : \mathbb{R}^n \to \mathbb{R}$, kita mulai dari tebakan awal x_0 lalu kita iterasi berdasarkan:

$$x_{i+1} = x_i - \gamma_i(\nabla f(x_i))^T$$

Dengan $\gamma \in \mathbb{R}_+$ disebut *step size* atau *learning rate* yang mengontrol seberapa jauh kita bergerak ke arah yang berlawanan dengan gradien.

Dengan memilih ukuran langkah γ yang cukup kecil, setiap iterasi dijamin akan menghasilkan nilai fungsi yang lebih kecil atau sama dengan sebelumnya.

$$f(x_0) \geq f(x_1) \geq f(x_2) \geq \dots$$

Dengan urutan ini, barisan (x_n) akan konvergen atau mendekati titik minimum lokal yang diinginkan.

7.1.1 Step Size

Memilih ukuran *step size* sangat penting. Jika terlalu kecil, gradient descent akan lambat. Jika terlalu besar, gradient descent akan melampaui titik minimum, gagal konvergen, bahkan divergen. Dua heuristik sederhana yang bisa digunakan adalah:

- Jika nilai fungsi meningkat setelah langkah gradien, maka *step size* terlalu besar. Kembalikan langkah dan kurangi nilai *step size*.
- Jika nilai fungsi menurun ukuran *step size* bisa lebih besar. Coba tambah nilai *step size*.

7.1.2 Gradient Descent dengan Momentum

Konvergensi *gradient descent* bisa menjadi sangat lambat, terutama pada permukaan optimisasi yang memiliki "lembah" panjang dan sempit (disebut sebagai *poorly conditioned*). Dalam kondisi seperti ini, langkah-langkah *gradient descent* cenderung "zig-zag" atau melompat-lompat di antara dinding lembah sambil membuat kemajuan yang sangat kecil menuju titik optimum.
Ide utama Gradient Descent dengan Momentum adalah memberikan "memori" pada *gradient descent* dengan menambahkan suku tambahan yang mengingat apa yang terjadi pada iterasi sebelumnya.

Intuisi: Mekanisme ini bekerja dengan cara Meredam osilasi dan memperhalus pembaruan gradien. Dianalogikan seperti bola berat yang enggan mengubah arahnya, sehingga ia terus melaju ke arah yang konsisten menuruni bukit dan tidak mudah berbelok karena "zig-zag".

Rumus pembaruan untuk *gradient descent* dengan momentum adalah:

$$
\begin{aligned}
x_{i+1} &= x_i - \gamma_i (\nabla f(xi))^\top + \alpha \Delta x_i \\
\Delta x_i &= x_i - x_{i-1}
\end{aligned}
$$

dengan suatu $\alpha \in [0, 1]$ adalah parameter momentum. Suku $\alpha \Delta x_i$ inilah yang berperan sebagai "memori".

7.1.3 Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) adalah sebuah pendekatan stokastik dari metode *gradient descent* yang digunakan untuk meminimalkan fungsi objektif, terutama yang berbentuk sebagai penjumlahan dari beberapa fungsi. Bentuk umum fungsi objektifnya adalah: $L(\theta) = \sum_{n=1}^{N} L_n(\theta)$ di mana θ adalah vektor parameter model dan L_n adalah fungsi *loss* untuk data ke-n.

Gradient descent standar, yang juga dikenal sebagai *batch gradient descent*, mengharuskan perhitungan gradien dari seluruh N data dalam *training set* untuk melakukan satu kali pembaruan parameter:

$$
\theta_{i+1} = \theta_i - \gamma_i \sum_{n=1}^{N} (\nabla L_n(\theta_i))^\top
$$

Proses ini menjadi sangat mahal secara komputasi dan memakan waktu, terutama ketika ukuran *training set* sangat besar.

SGD mengatasi masalah ini dengan menggunakan aproksimasi gradien yang "murah" dengan menggunakan subset acak dari data (disebut *mini-batch*) untuk mengestimasi gradien di setiap langkahnya. Dalam kasus ekstrem, estimasi gradien bahkan bisa dilakukan hanya dengan satu data acak. Kata "stokastik" merujuk bahwa gradien yang digunakan hanyalah sebuah aproksimasi.

Kunci keberhasilan SGD adalah bahwa untuk konvergensi, estimasi gradien yang digunakan hanya perlu menjadi estimasi yang tidak bias (*unbiased estimate*) dari gradien sebenarnya. Penggunaan sampel acak (subsample) dari data sudah cukup untuk memenuhi syarat ini. Dengan laju pembelajaran (*learning rate*) yang menurun secara tepat, SGD secara

teoretis dijamin akan konvergen menuju minimum lokal. Tujuan dari machine learning adalah performa generalisasi, bukan sekadar menemukan nilai minimum yang presisi dari fungsi objektif pada data latih. Oleh karena itu, pendekatan dengan gradien aproksimatif seperti SGD akan menjadi lebih efektif.

7.2 Optimasi Terkendala dan Pengali Lagrange

> **Optimasi Terkendala**
>
> Optimisasi terkendala adalah proses untuk menemukan nilai maksimum atau minimum (nilai optimal) dari sebuah fungsi, dengan syarat bahwa variabel-variabelnya harus memenuhi serangkaian batasan atau kondisi yang disebut kendala (*constraints*). Berbeda dengan optimisasi tak terkendala di mana kita mencari titik terendah atau tertinggi di seluruh "permukaan" fungsi, optimisasi terkendala membatasi pencarian kita hanya pada wilayah yang diizinkan oleh kendala-kendala tersebut.

Secara formal, masalah ini dapat ditulis sebagai:

- Minimalkan (atau maksimalkan): $f(x)$
- Dengan kendala: $g_i(x) \leq 0$ untuk setiap $i = 1, \ldots, m$ dan/atau $h_j(x) = 0$ untuk setiap $j = 1, \ldots, n$.
  Ide visual tentang persinggungan ini memiliki landasan matematis yang kuat menggunakan konsep gradien (∇).
- Gradien dari sebuah fungsi pada suatu titik selalu tegak lurus terhadap garis kontur fungsi tersebut di titik itu.
- Ketika dua kurva (garis kontur f dan kurva kendala g) bersinggungan, mereka memiliki garis singgung yang sama di titik tersebut. Akibatnya, vektor gradien mereka juga harus sejajar, artinya keduanya menunjuk ke arah yang sama atau berlawanan.
  Secara matematis, ini berarti gradien satu fungsi adalah kelipatan skalar dari gradien fungsi lainnya. Hubungan ini dinyatakan sebagai:

  $$\nabla f(x) = \lambda \nabla g(x)$$

  Konstanta proporsionalitas λ ini disebut Pengali Lagrange (*Lagrange Multiplier*).

Untuk menyelesaikan masalah optimisasi terkendala, kita membangun sebuah sistem persamaan yang terdiri dari persamaan gradien di atas dan persamaan kendala aslinya.

Secara lebih formal, pendekatan ini menggunakan fungsi Lagrangian, $\mathcal{L}(x, \lambda)$, yang menggabungkan fungsi objektif dan kendala menjadi satu fungsi tunggal:

$$\mathcal{L}(x, \boldsymbol{\lambda}) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)$$

Di mana λ_i adalah Pengali Lagrange untuk setiap kendala $g_i(x)$.

Contoh 1: Memaksimalkan Fungsi pada Sebuah Lingkaran

Misalkan kita ingin memaksimalkan fungsi $f(x, y) = x^2 y$ dengan kendala $g(x, y) = x^2 + y^2 = 1$.
Pertama, hitung gradien dari kedua fungsi:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right] = [2xy, x^2]$$
$$\nabla g = \left[\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right] = [2x, 2y]$$

didapat:

$$\begin{aligned} 2xy &= \lambda(2x) \\ y &= \lambda, \\ x^2 &= \lambda(2y) = 2y^2 \\ x^2 + y^2 &= 1 \end{aligned}$$

Selesaikan sistem persamaan tersebut untuk menemukan nilai x, y, dan λ. Solusi dari sistem ini akan memberikan titik-titik kandidat di mana nilai maksimum atau minimum berada.

7.3 Optimisasi Cembung

Optimisasi cembung adalah subbidang optimisasi yang sangat penting karena memiliki properti yang kuat: *setiap minimum lokal yang ditemukan juga merupakan minimum global.* Hal ini menghilangkan masalah terjebak dalam solusi suboptimal yang sering terjadi pada fungsi non-cembung. Selain itu, untuk masalah optimisasi cembung, berlaku *strong duality*, yang berarti nilai optimal dari masalah primal dan masalah dual adalah sama.

> **Himpunan Cembung (Convex Set)**
>
> Sebuah himpunan C disebut cembung jika untuk setiap dua titik $x, \mathbf{y} \in C$ dan skalar θ dengan $0 \leq \theta \leq 1$, garis lurus yang menghubungkan kedua titik tersebut sepenuhnya

berada di dalam himpunan:

$$\theta x + (1-\theta)\mathbf{y} \in C$$

Fungsi Cembung (Convex Function)

Sebuah fungsi $f : \mathbb{R}^D \to \mathbb{R}$ yang domainnya adalah himpunan cembung disebut fungsi cembung jika ruas garis yang menghubungkan dua titik mana pun pada grafiknya selalu berada di atas atau pada grafik fungsi itu sendiri. Secara matematis, untuk setiap x, y dalam domain f dan untuk setiap skalar θ dengan $0 \leq \theta \leq 1$, berlaku:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

. Ketidaksetaraan ini juga dikenal sebagai Ketidaksetaraan Jensen. Secara visual, fungsi cembung berbentuk seperti "mangkuk". Fungsi cekung (*concave function*) adalah negatif dari fungsi cembung.

Jika sebuah fungsi $f(x)$ dapat diturunkan, terdapat cara lain untuk menentukan kecembungannya:

- Syarat Orde Pertama: Fungsi $f(x)$ adalah cembung jika dan hanya jika grafik fungsi tersebut selalu berada di atas garis singgungnya di setiap titik. Secara matematis, untuk setiap titik x, y berlaku:

$$f(y) \geq f(x) + (\nabla_x f(x))^\top (y - x)$$

- Syarat Orde Kedua: Jika fungsi $f(x)$ dapat diturunkan dua kali, maka $f(x)$ adalah cembung jika dan hanya jika matriks Hessian-nya, $\nabla_x^2 f(x)$, adalah positif semidefinit untuk semua nilai x dalam domainnya.

Contoh 2: Menentukan kecembungan

Pertimbangkan fungsi $f(x) = x^4$. Untuk memeriksa kecembungannya, kita dapat menggunakan syarat orde kedua. Pertama hitung turunan pertama $f'(x) = 4x^3$ dan turunan keduanya $f''(x) = 12x^2$. Karena $f''(x) = 12x^2 \geq 0$ untuk semua $x \in \mathbb{R}$, maka Hessian (yang dalam kasus satu variabel hanyalah turunan kedua) selalu non-negatif.

Artinya $f(x) = x^4$ adalah fungsi cembung di seluruh domain real. Sebagai verifikasi dengan syarat orde pertama, ambil $x = 1$ dan $y = 2$. Nilai $f(2) = 16$, sedangkan $f(1) + f'(1)(2 - 1) = 1 + 4(1)(1) = 5$. Karena $16 \geq 5$, maka syarat $f(y) \geq f(x) + f'(x)(y - x)$ juga terpenuhi. Dengan demikian kedua syarat konsisten menyatakan bahwa $f(x) = x^4$ adalah fungsi cembung.

Sebuah masalah optimisasi terkendala disebut sebagai masalah optimisasi cembung jika ia memiliki bentuk:

$$\min_x f(x)$$

Tunduk pada:

$$\begin{aligned} g_i(x) &\leq 0 \quad \text{untuk semua } i = 1, \ldots, m \\ h_j(x) &= 0 \quad \text{untuk semua } j = 1, \ldots, n \end{aligned}$$

di mana fungsi objektif $f(x)$ dan semua fungsi kendala pertidaksamaan $g_i(x)$ adalah fungsi cembung, dan semua fungsi kendala persamaan $h_j(x)$ mendefinisikan himpunan cembung.

Dalam konteks ini, berlaku dualitas kuat (strong duality), yang berarti solusi optimal dari masalah dual sama dengan solusi optimal dari masalah primal.

7.3.1 Linear Programming (LP)

Ini adalah kasus khusus di mana fungsi objektif dan semua fungsi kendala adalah linear. Bentuk primal dari LP adalah:

$$\min_{x \in \mathbb{R}^d} c^\top x \quad \text{dengan} \quad Ax \leq b$$

Masalah dual yang bersesuaian adalah:

$$\max_{\lambda \in \mathbb{R}^m} -b^\top \lambda \quad \text{dengan} \quad c + A^\top \lambda = 0, \quad \lambda \geq 0$$

7.3.2 Quadratic Programming (QP)

Ini adalah kasus di mana fungsi objektif adalah fungsi kuadratik cembung dan kendalanya adalah afin (linear). Bentuk primal dari QP adalah:

$$\min_{x \in \mathbb{R}^d} \frac{1}{2} x^\top Q x + c^\top x \quad \text{dengan} \quad Ax \leq b$$

di mana matriks Q adalah simetris dan positif definit, yang memastikan fungsi objektifnya cembung. Masalah dual yang bersesuaian adalah:

$$\max_{\lambda \in \mathbb{R}^m} -\frac{1}{2}(c + A^\top \lambda)^\top Q^{-1}(c + A^\top \lambda) - \lambda^\top b \quad \text{dengan} \quad \lambda \geq 0$$

7.3.2 Konjugat Cembung (Convex Conjugate)

Konjugat cembung, atau dikenal juga sebagai *transformasi Legendre-Fenchel*, adalah cara lain untuk memahami dualitas dalam optimisasi cembung. Secara intuitif, transformasi ini mendeskripsikan ulang sebuah fungsi cembung bukan berdasarkan titik-titiknya $(x, f(x))$, melainkan berdasarkan hiperbidang penyokongnya (garis singgung).

Definisi formal dari konjugat cembung untuk fungsi $f : \mathbb{R}^D \to \mathbb{R}$ adalah fungsi $f^\star$ yang didefinisikan sebagai:

$$f^*(s) = \sup_{x \in \mathbb{R}^D} \left(\langle s, x \rangle - f(x) \right)$$

Di sini, s dapat dianggap sebagai gradien atau "kemiringan" dari hiperbidang penyokong. Untuk fungsi cembung, menerapkan transformasi konjugat dua kali akan mengembalikan fungsi aslinya. Alat ini sangat berguna untuk menurunkan masalah dual, terutama dalam konteks machine learning.