Part 11 - Estimasi Kepadatan dengan Gaussian Mixture Models

Estimasi kepadatan bertujuan untuk merepresentasikan data secara ringkas menggunakan sebuah fungsi kepadatan dari keluarga parametrik, seperti distribusi Gaussian. Namun, distribusi tunggal seperti Gaussian sering kali memiliki kemampuan pemodelan yang terbatas dan tidak dapat merepresentasikan data yang kompleks secara memadai, misalnya data dengan beberapa "gugus" (multimodal). *Mixture models* menawarkan solusi yang lebih ekspresif dengan merepresentasikan sebuah distribusi $p(x)$ sebagai kombinasi cembung dari K distribusi dasar (komponen) yang sederhana:

$$p(x) = \sum_{k=1}^{K} \pi_k p_k(x)$$

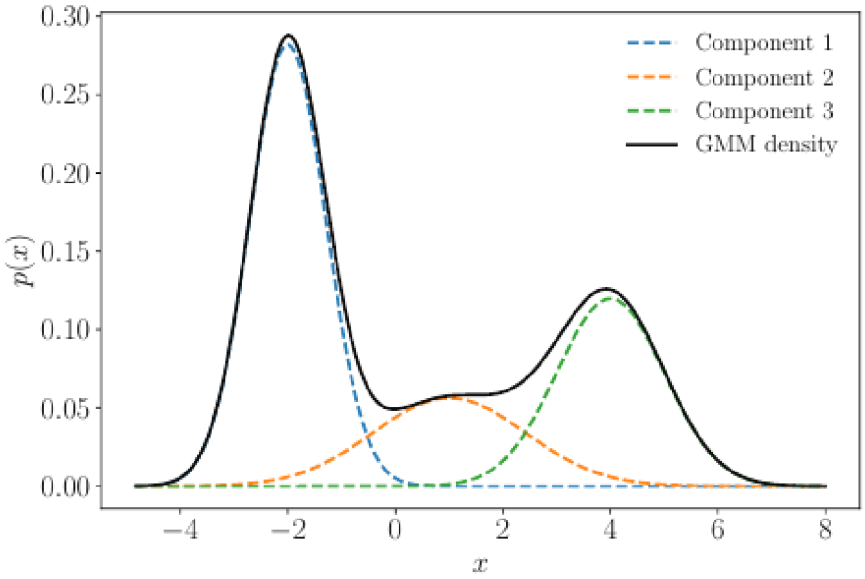
dengan kendala pada bobot campuran π_k yaitu $0 \leq \pi_k \leq 1$ dan $\sum_{k=1}^{K} \pi_k = 1$. Bab ini berfokus pada *Gaussian Mixture Models* (GMM), di mana distribusi dasarnya adalah Gaussian. Karena tidak ada solusi bentuk-tertutup untuk estimasi parameter GMM, kita akan menggunakan skema iteratif.

11.1 Gaussian Mixture Model

Sebuah *Gaussian Mixture Model* (GMM) adalah sebuah model kepadatan di mana kita menggabungkan sejumlah terbatas K distribusi Gaussian $\mathcal{N}(x|\mu_k, \Sigma_k)$ sehingga:

$$p(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

di mana $\theta := \{\mu_k, \Sigma_k, \pi_k : k = 1, \ldots, K\}$ adalah kumpulan semua parameter model. Kombinasi cembung ini memberikan fleksibilitas yang jauh lebih besar untuk memodelkan kepadatan yang kompleks dibandingkan dengan distribusi Gaussian tunggal. Contohnya adalah pada gaussian mixture berikut:



$$p(x|\theta) = 0.5\mathcal{N}\left(x| -2, \frac{1}{2}\right) + 0.2\mathcal{N}(x|1,2) + 0.3\mathcal{N}(x|4|1)$$

11.2 Pembelajaran Parameter melalui Maximum Likelihood

Dengan asumsi kita memiliki set data $X = x_1, \ldots, x_N$ yang ditarik dari distribusi yang tidak diketahui, tujuan kita adalah mencari parameter GMM yang dapat merepresentasikan distribusi ini dengan baik. Kita menggunakan estimasi *maximum likelihood. Log-likelihood* dari data didefinisikan sebagai:

$$\log p(X|\theta) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right)$$

Tantangan utamanya adalah keberadaan logaritma di luar penjumlahan, yang menghalangi kita untuk mendapatkan solusi bentuk-tertutup. Sebagai gantinya, kita akan mencari kondisi optimalitas dengan mengatur gradien dari *log-likelihood* terhadap setiap parameter menjadi nol.

11.2.1 Responsibilities

Kita mendefinisikan sebuah kuantitas sentral yang disebut *responsibility* (tanggung jawab) dari komponen campuran ke-k untuk titik data ke-n sebagai:

$$r_{nk} := \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

Nilai r_{nk} ini dapat diinterpretasikan sebagai probabilitas bahwa titik data x_n telah

dihasilkan oleh komponen campuran ke-k. Ini adalah bentuk "penugasan lunak" (*soft assignment*) dari setiap titik data ke K komponen, di mana untuk setiap n, $\sum_k r_{nk} = 1$.

11.2.2 Memperbarui Rata-rata (Means)

Dengan mengatur turunan parsial dari *log-likelihood* terhadap μ_k menjadi nol, kita mendapatkan aturan pembaruan untuk rata-rata:

$$\mu_k^{\text{new}} = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}}$$

Pembaruan ini merupakan rata-rata tertimbang dari titik-titik data, di mana bobotnya adalah *responsibility*. Kita mendefinisikan $N_k := \sum_{n=1}^{N} r_{nk}$ sebagai total *responsibility* dari komponen ke-k.

11.2.3 Memperbarui Kovariansi

Dengan cara yang sama, aturan pembaruan untuk matriks kovariansi Σ_k diperoleh dengan mengatur turunan parsial *log-likelihood* menjadi nol:

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (x_n - \mu_k)(x_n - \mu_k)^\top$$

Ini dapat diartikan sebagai estimasi kovariansi tertimbang untuk setiap komponen.

11.2.4 Memperbarui Bobot Campuran

Untuk memperbarui bobot campuran π_k dengan kendala $\sum_k \pi_k = 1$, kita menggunakan Pengali Lagrange. Hasilnya adalah aturan pembaruan:

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

Bobot baru untuk komponen ke-k adalah rasio dari total *responsibility*-nya terhadap jumlah total titik data.

11.3 Algoritma EM

Karena persamaan pembaruan untuk semua parameter saling bergantung satu sama lain melalui *responsibilities*, kita tidak dapat memperoleh solusi bentuk-tertutup. Namun, ini mengarah pada skema iteratif yang sederhana yang disebut algoritma *Expectation-*

Maximization (EM). Algoritma EM untuk GMM bergantian antara dua langkah berikut hingga konvergensi:

1. **Langkah-E (Ekspektasi):** Evaluasi *responsibilities* r_{nk} menggunakan nilai parameter saat ini π_k, μ_k, Σ_k.
2. **Langkah-M (Maksimisasi):** Estimasi ulang parameter π_k, μ_k, Σ_k menggunakan *responsibilities* yang baru dihitung dari Langkah-E dengan persamaan pembaruan yang telah diturunkan. Setiap iterasi dari algoritma EM dijamin akan meningkatkan (atau tidak menurunkan) nilai *log-likelihood*.

11.4 Perspektif Variabel Laten

GMM dapat dipandang sebagai model dengan variabel laten diskrit, yang memberikan landasan teoretis yang lebih kokoh untuk algoritma EM. Kita memperkenalkan variabel laten biner $z_k \in 0, 1$ yang mengindikasikan komponen mana yang menghasilkan sebuah titik data. Vektor $z = [z_1, \ldots, z_K]^\top$ adalah representasi *one-hot*, dengan hanya satu entri bernilai 1.

- Distribusi prior pada variabel laten ini adalah $p(z_k = 1) = \pi_k$.
- Distribusi kondisional dari data adalah $p(x|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k)$. Untuk mendapatkan *likelihood* $p(x|\theta)$, kita melakukan marginalisasi terhadap variabel laten z:

  $$p(x|\theta) = \sum_z p(x|\theta, z)p(z|\theta) = \sum_{k=1}^{K} p(x|\theta, z_k = 1)p(z_k = 1|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

  Persamaan ini sama persis dengan definisi GMM di awal. Dari perspektif ini, *responsibility* r_{nk} ternyata adalah probabilitas posterior dari variabel laten, yaitu $p(z_{nk} = 1|x_n)$. Ini memberikan interpretasi matematis yang sahih untuk *responsibilities*.