Pertemuan 5 — *Bayesian Learning*, *Naive Bayes*, GMM, dan EM

Dari peluang berlabel ke campuran Gaussian tanpa label

Hendri Karisma

Semester Ganjil 2026/2027

Program Studi Teknik Informatika, STMIK Tazkia

Tujuan dan alur pertemuan

Alur

1. Dasar probabilitas, teorema Bayes, MAP, *maximum likelihood*, *Bayes optimal*
2. *Naive Bayes*: PlayTennis, teks, Laplace, log
3. *Gaussian naive Bayes* sebagai jembatan
4. Gaussian univariat dan multivariat
5. GMM, variabel laten, dan algoritma EM
6. Optimum lokal, BIC, campuran Bernoulli
7. Kode, praktikum, Ringkasan, kuis

Setelah pertemuan ini Anda dapat

- menghitung *posterior* dan membedakan h_{MAP} dari h_{ML};
- menghitung *naive Bayes* dengan tangan, dengan Laplace dan log;
- membedakan *density estimation* dan *clustering*;
- menjalankan E-step dan M-step GMM dengan tangan;
- memakai `GaussianMixture` dan memilih K dengan BIC.

Sumber: buku ajar Bab 5 dan Bab 6; Mitchell Bab 6; Bishop Bab 2 dan 9.

Teorema Bayes

Dasar probabilitas: peluang selalu dihitung di dalam sebuah dunia

Possibility dan *probability*

Possibility menjawab ya atau tidak: bisakah kejadian itu terjadi? ***Probability*** (peluang) adalah angka 0 sampai 1 di dalam ruang hasil yang sudah ditentukan. Angka 6 dan 1 pada dadu sama-sama *possible*; pada dadu seimbang peluang keduanya $1/6$, pada dadu berat sebelah tidak lagi sama.

Sample space dan *event*

Sample space Ω = himpunan semua hasil, misalnya $\Omega = \{1, \ldots, 6\}$. ***Event*** $A \subseteq \Omega$, misalnya $A = \{2, 4, 6\}$, sehingga $P(\text{genap}) = 3/6$.

Di ML, Ω dibangun dari *dataset*

PlayTennis berisi 14 hari, 9 berlabel Yes: $P(\text{Play} = \text{Yes}) = 9/14 \approx 0{,}643$. Angka ini berlaku di dunia 14 hari itu. Data dari kota atau musim lain membangun Ω lain, jadi peluang dari data adalah taksiran (*estimate*).

Aksioma Kolmogorov

$0 \leq P(A) \leq 1$; $P(\Omega) = 1$; bila $A \cap B = \emptyset$ maka $P(A \cup B) = P(A) + P(B)$.

Notasi: $\subseteq$: himpunan bagian dari; $A \cap B$ (irisan): termasuk *A dan B*; $A \cup B$ (gabungan): termasuk *A atau B*; $\emptyset$: himpunan kosong.

Sumber: buku ajar Subbab 5.1.1; Bertsekas dan Tsitsiklis, *Introduction to Probability*, 2008.

Conditional probability, *independent*, dan *dependent*

Conditional probability

$P(A \mid B)$ dibaca "peluang A bila B diketahui terjadi". Dunianya menyempit dari Ω menjadi B:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \quad P(B) > 0$$

PlayTennis punya 5 hari Sunny, 2 di antaranya Yes: $P(\text{Yes} \mid \text{Sunny}) = 2/5 = 0{,}4$. Urutannya penting: $P(\text{Sunny} \mid \text{Yes}) = 2/9 \approx 0{,}22$.

Independent dan *dependent*

A dan B ***independent*** bila $P(A \mid B) = P(A) \Leftrightarrow P(A \cap B) = P(A)\,P(B)$. Dua lemparan koin: $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. Bila tidak berlaku, keduanya ***dependent***. Di PlayTennis $0{,}4 \neq 0{,}643$, jadi Yes dan Sunny *dependent*; ketergantungan inilah yang dipakai model untuk memprediksi.

Hati-hati

Tidak beririsan bukan berarti *independent*: bila $A \cap B = \emptyset$, terjadinya A memastikan B tidak terjadi, jadi keduanya *dependent*.

Dari rumus di kiri, $P(A \cap B) = P(A \mid B)\,P(B) = P(B \mid A)\,P(A)$ (*product rule*). Membagi dengan $P(B)$ menghasilkan teorema Bayes pada slide berikut.

Notasi: garis tegak $\mid$ di dalam $P(\cdot)$: "dengan syarat"; $\Leftrightarrow$: setara (berlaku dua arah); $\neq$: tidak sama dengan.

Sumber: buku ajar Subbab 5.1.1.

Teorema Bayes dan empat istilahnya

Teorema Bayes

$$P(h \mid D) = \frac{P(D \mid h)\, P(h)}{P(D)}$$

- ***Prior*** $P(h)$: keyakinan atas hipotesis h **sebelum** melihat data; pengetahuan awal masuk di sini secara terbuka.
- ***Likelihood*** $P(D \mid h)$: peluang data seperti ini muncul **jika** h benar. Bagian ini yang dihitung model.
- ***Posterior*** $P(h \mid D)$: keyakinan atas h **setelah** melihat data.
- ***Evidence*** $P(D) = \sum_{h_i} P(D \mid h_i) P(h_i)$: tidak bergantung pada h, jadi cukup diperlakukan sebagai penormal.

Ringkasnya $P(h \mid D) \propto P(D \mid h)\, P(h)$ ($\propto$: sebanding dengan). Makin kuat *prior*-nya, makin banyak data yang diperlukan untuk menggesernya.

Sumber: Mitchell, *Machine Learning*, Bab 6; buku ajar Subbab 5.1.

Contoh terhitung: uji penyakit langka

Penyakit menyerang 8 dari 1.000 orang. Sensitivitas uji 98%, spesifisitas 97%. Hasil uji Anda positif. Berapa peluang Anda benar-benar sakit?

Yang diketahui:

$$P(\text{sakit}) = 0{,}008 \qquad P(\text{sehat}) = 0{,}992$$
$$P(+ \mid \text{sakit}) = 0{,}98 \qquad P(+ \mid \text{sehat}) = 0{,}03$$

Dua jalur menuju hasil positif:

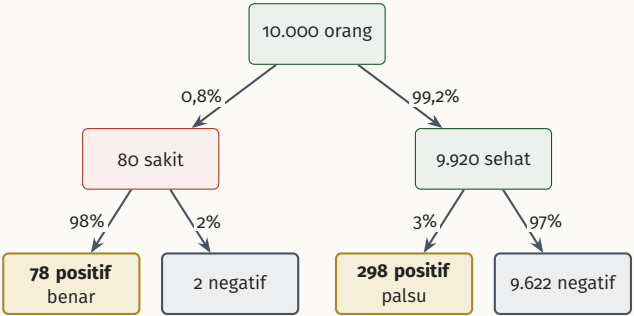
$$0{,}98 \times 0{,}008 = 0{,}00784$$
$$0{,}03 \times 0{,}992 = 0{,}02976$$
$$P(+) = 0{,}03760$$

$$P(\text{sakit} \mid +) = \frac{0{,}00784}{0{,}03760} \approx \mathbf{0{,}209}$$

Ringkasan

Hanya sekitar 21%. Uji yang “98% akurat” tetap menghasilkan lebih banyak positif palsu daripada positif benar bila penyakitnya langka.

Pohon frekuensi dan tiga kekeliruan



Dari 376 hasil positif hanya 78 yang sakit: $78/376 = 0{,}207$.

- ***Prior* diabaikan.** Yang diingat hanya 98% dan 97%; prevalensi 0,008 dilupakan (*base rate fallacy*).
- **Dua arah bersyarat ditukar.** $P(+ \mid \text{sakit})$ adalah sifat alat uji, sedangkan yang dibutuhkan $P(\text{sakit} \mid +)$ (*prosecutor's fallacy*).
- **Yang sehat jauh lebih banyak.** Alarm palsu 3% dari 9.920 orang mengalahkan 78 deteksi benar. Gejala yang sama muncul pada deteksi penipuan transaksi.

Posterior menjadi *prior*: uji berulang

Uji kedua yang independen memakai *posterior* uji pertama sebagai *prior*:

$$P(\text{sakit} \mid +,+) = \frac{0{,}008 \times 0{,}98^2}{0{,}008 \times 0{,}98^2 + 0{,}992 \times 0{,}03^2} = \frac{0{,}0076832}{0{,}0085760} \approx 0{,}896$$

```python
def posterior(prior, sensitivitas, spesifisitas):
    jalur_sakit = prior * sensitivitas              # P(+|sakit) P(sakit)
    jalur_sehat = (1 - prior) * (1 - spesifisitas)  # P(+|sehat) P(sehat)
    return jalur_sakit / (jalur_sakit + jalur_sehat)

p = 0.008
for uji in (1, 2, 3):
    p = posterior(p, 0.98, 0.97)   # posterior menjadi prior berikutnya
    print(f"setelah {uji} uji positif: {p:.3f}")
# 0.209, 0.896, 0.996
```

Keyakinan bergerak $0{,}008 \to 0{,}209 \to 0{,}896 \to 0{,}996$; setiap hasil menjadi titik awal bagi bukti berikutnya. Pola ini inti *Bayesian learning*.

Sumber: `bab05-bayesian-naive-bayes/bayes_penyakit.py`.

MAP, *maximum likelihood*, dan *Bayes optimal*

Hipotesis MAP dan *maximum likelihood*

MAP (*maximum a posteriori*)

$$h_{MAP} = \arg\max_{h \in H} P(D \mid h)\, P(h)$$

Notasi: $\arg\max_h$: nilai h yang membuat ekspresi paling besar, bukan nilai maksimumnya.

Penyebut $P(D)$ dibuang karena sama untuk semua h.

ML (*maximum likelihood*)

$$h_{ML} = \arg\max_{h \in H} P(D \mid h)$$

Sama dengan MAP bila *prior*-nya seragam.

Uji penyakit, $H = \{\text{sakit}, \text{sehat}\}$ dan $D = \{+\}$: h_{ML} membandingkan 0,98 dengan 0,03 dan menjawab **sakit**; h_{MAP} membandingkan 0,00784 dengan 0,02976 dan menjawab **sehat**.

Koin yang dilempar tiga kali dan selalu keluar angka: $\hat{\theta}_{ML} = 3/3 = 1$, terlalu berani untuk tiga lemparan. Dengan *prior* Beta$(2, 2)$, $\hat{\theta}_{MAP} = \frac{\text{angka}+a-1}{n+a+b-2} = \frac{3+1}{3+2} = 0{,}8$. *Prior* menarik taksiran menjauhi nilai ekstrem. Gagasan ini muncul lagi sebagai *Laplace smoothing*.

ML dengan *error* Gaussian menghasilkan *least squares*, dan ML Bernoulli menghasilkan *cross-entropy* (diturunkan di Pertemuan 9).

Bayes optimal classifier dan algoritma Gibbs

Untuk contoh baru yang dicari adalah **label** paling mungkin:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i)\, P(h_i \mid D)$$

Setiap hipotesis memberi suara dengan *weight posterior*-nya.

Hipotesis	$P(h_i \mid D)$	Jawaban untuk x
h_1	0,4	$+$
h_2	0,3	$-$
h_3	0,3	$-$

$h_{MAP} = h_1$ menjawab $+$. *Weight* untuk $-$ adalah 0,6, sehingga *Bayes optimal* menjawab $-$.

- Dengan ruang hipotesis dan *prior* yang sama, tidak ada metode yang rata-rata mengalahkannya.
- ***Cost.*** Jumlah atas **seluruh** ruang hipotesis untuk setiap *prediction*; pada *decision tree* atau *neural network* ruangnya tak hingga.
- ***Prior*** $P(h)$ jarang diketahui dengan dasar yang kuat.
- **Gibbs**: ambil satu h acak menurut *posterior*. *Error* harapannya paling buruk dua kali *error* harapan *Bayes optimal*, bila konsep sasaran diambil menurut *prior*.

Jalan praktis

Berhenti menjumlah atas hipotesis, taksir $P(v \mid a_1, \ldots, a_d)$ langsung dari data. Langkah ini membawa kita ke *naive Bayes*.

Naive Bayes

Tabel peluang penuh terlalu mahal

$$v_{MAP} = \arg\max_{v \in V} P(a_1, \ldots, a_d \mid v)\, P(v)$$

Menaksir $P(a_1, \ldots, a_d \mid v)$ berarti menghitung frekuensi **setiap kombinasi** nilai atribut per kelas. PlayTennis: $3 \times 3 \times 2 \times 2 = 36$ kombinasi per kelas (sekitar 70 bilangan) dari 14 baris data. Teks dengan 10.000 kata biner: 2^{10000}.

Asumsi *conditional independence*

Setelah kelasnya diketahui, nilai satu atribut tidak memberi informasi tentang atribut lain:

$$P(a_1, \ldots, a_d \mid v) = \prod_{i=1}^{d} P(a_i \mid v) \qquad (\textstyle\prod$, pi kapital: perkalian semua suku)

Jumlah parameter turun dari perkalian menjadi penjumlahan: $3 + 3 + 2 + 2 = 10$ taksiran per kelas, untuk teks 2×10.000. Asumsinya keliru (hari hujan cenderung lembap), tetapi semua parameternya kini dapat ditaksir.

Aturan *naive Bayes* dan cara melatihnya

Classifier naive Bayes

$$v_{NB} = \arg\max_{v \in V} P(v) \prod_{i=1}^{d} P(a_i \mid v)$$

- $\hat{P}(v) = n_v / n$
- $\hat{P}(a_i \mid v) = n_{v,a_i} / n_v$
- Satu lintasan atas data; tanpa iterasi dan tanpa *learning rate*.

Algoritma: *naive Bayes* kategorikal

Input: *training data* $\{(\mathbf{a}^{(j)}, v^{(j)})\}_{j=1}^{n}$, *smoothing* $\alpha \geq 0$

for setiap kelas v **do**
  $n_v \leftarrow$ banyak contoh berkelas v; $\hat{P}(v) \leftarrow n_v / n$
  for setiap atribut i dan nilai a **do**
    $n_{v,a} \leftarrow$ banyak contoh kelas v dengan $a_i = a$
    $\hat{P}(a \mid v) \leftarrow (n_{v,a} + \alpha)/(n_v + \alpha k_i)$

Prediction untuk contoh baru $\mathbf{a}$:

for setiap kelas v **do**
  $s_v \leftarrow \log \hat{P}(v) + \sum_i \log \hat{P}(a_i \mid v)$
return $\arg\max_v s_v$

Notasi: $\leftarrow$: pemberian nilai (*assignment*), nilai di kanan disimpan ke variabel di kiri.

Sumber: buku ajar Algoritma 5.1; k_i = banyaknya nilai atribut i.

Contoh terhitung: data PlayTennis

	Outlook	Temp	Humidity	Wind	Play		Outlook	Temp	Humidity	Wind	Play
1	Sunny	Hot	High	Weak	No	8	Sunny	Mild	High	Weak	No
2	Sunny	Hot	High	Strong	No	9	Sunny	Cool	Normal	Weak	Yes
3	Overcast	Hot	High	Weak	Yes	10	Rain	Mild	Normal	Weak	Yes
4	Rain	Mild	High	Weak	Yes	11	Sunny	Mild	Normal	Strong	Yes
5	Rain	Cool	Normal	Weak	Yes	12	Overcast	Mild	High	Strong	Yes
6	Rain	Cool	Normal	Strong	No	13	Overcast	Hot	Normal	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes	14	Rain	Mild	High	Strong	No

Empat belas baris: **Yes** 9 baris, **No** 5 baris. Keputusan untuk hari baru

$$\langle \text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong} \rangle\ ?$$

Kombinasi persis ini tidak ada di *training data*. *Naive Bayes* tetap bisa menjawab karena hanya memerlukan potongan-potongannya.

Sumber: Mitchell, *Machine Learning*, Tabel 3.2.

PlayTennis: tabel frekuensi, lalu kalikan

Taksiran	Yes	No
$P(v)$	$9/14 = 0{,}643$	$5/14 = 0{,}357$
$P(\text{Sunny} \mid v)$	$2/9 = 0{,}222$	$3/5 = 0{,}600$
$P(\text{Cool} \mid v)$	$3/9 = 0{,}333$	$1/5 = 0{,}200$
$P(\text{High} \mid v)$	$3/9 = 0{,}333$	$4/5 = 0{,}800$
$P(\text{Strong} \mid v)$	$3/9 = 0{,}333$	$3/5 = 0{,}600$

Di antara 9 hari Yes ada 2 hari Sunny (hari 9 dan 11), jadi $P(\text{Sunny} \mid \text{Yes}) = 2/9$; di antara 5 hari No ada 3 (hari 1, 2, 8), jadi $3/5$. Setiap taksiran dihitung **di dalam kelasnya sendiri**.

Kelas **Yes**:

$$0{,}643 \times 0{,}222 \times 0{,}333^3 = \mathbf{0{,}0053}$$

Kelas **No**:

$$0{,}357 \times 0{,}600 \times 0{,}200 \times 0{,}800 \times 0{,}600 = \mathbf{0{,}0206}$$

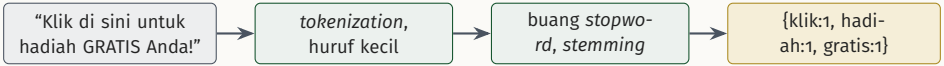
Jadi $v_{NB} = \text{No}$. Penyebutnya sama untuk kedua kelas, sehingga cukup dinormalkan:

$$P(\text{No} \mid \text{hari itu}) \approx \frac{0{,}0206}{0{,}0206 + 0{,}0053} = 0{,}795$$

Ringkasan

Humidity (0,800 lawan 0,333) dan Outlook (0,600 lawan 0,222) mendorong keputusan ke arah No. Atribut yang menentukan keputusan dapat dibaca langsung dari tabel.

Teks: *bag-of-words* dan korpus mini



Varian **multinomial**: $\hat{P}(w \mid c) = \dfrac{\text{jumlah kemunculan } w \text{ di kelas } c}{\text{total token kelas } c}$.

SPAM: 3 *email*, 10 token

“menang hadiah gratis”, “gratis hadiah klik sekarang”, “klik gratis menang”.

Hitungan: gratis 3, hadiah 2, menang 2, klik 2, sekarang 1.

BUKAN SPAM: 3 *email*, 10 token

“rapat besok pagi”, “jadwal rapat proyek”, “kirim laporan proyek besok”.

Hitungan: rapat 2, besok 2, proyek 2, pagi, jadwal, kirim, laporan 1.

$|V| = 12$, $P(\text{spam}) = P(\text{bukan}) = 0{,}5$. Taksiran mentah: $\hat{P}(\text{gratis} \mid \text{spam}) = 3/10 = 0{,}3$, $\hat{P}(\text{rapat} \mid \text{spam}) = 0/10 = 0$.

Masalah peluang nol

Email baru **“gratis rapat besok”** dengan taksiran mentah:

$$\text{SPAM:}\quad 0{,}5 \times \underbrace{0{,}3}_{\text{gratis}} \times \underbrace{0}_{\text{rapat}} \times \underbrace{0}_{\text{besok}} = 0$$

$$\text{BUKAN:}\quad 0{,}5 \times \underbrace{0}_{\text{gratis}} \times \underbrace{0{,}2}_{\text{rapat}} \times \underbrace{0{,}2}_{\text{besok}} = 0$$

Kedua kelas memberi nol, dan model tidak dapat memutuskan apa pun.

- Satu faktor nol membatalkan seluruh bukti dari faktor lain.
- Dengan kosakata 20.000 kata, kata yang belum pernah muncul di suatu kelas adalah hal biasa; sebagian besar sel tabel memang kosong.
- Taksiran 0/10 menyatakan kata itu mustahil, padahal datanya hanya menunjukkan kata itu belum terlihat.

Laplace smoothing dan *m-estimate*

Laplace (*add-one*)

$$\hat{P}(a_i \mid v) = \frac{n_{v,a_i} + 1}{n_v + k}$$

k = banyaknya nilai atribut (teks multinomial: $k = |V|$). Satu pengamatan semu ditambahkan pada setiap nilai.

m-estimate

$$\hat{P}(a_i \mid v) = \frac{n_{v,a_i} + m\,p}{n_v + m}$$

p = peluang awal yang diyakini, m = *weight* keyakinan dalam "jumlah data semu". Laplace: $p = 1/k$, $m = k$.

- Bila n_v besar, tambahan itu tidak terasa; bila n_v kecil, tambahan itulah yang mencegah nol.
- Ini taksiran MAP dengan *prior* Beta (contoh koin tadi), diperluas ke *prior* Dirichlet untuk atribut dengan lebih dari dua nilai.
- Di scikit-learn besarnya diatur parameter `alpha`; `alpha=1` adalah Laplace.

Smoothing dengan angka

PlayTennis. Tidak ada hari Overcast berkelas No, jadi $\hat{P}(\text{Overcast} \mid \text{No}) = 0/5$. Outlook punya $k = 3$ nilai:

$$\text{Laplace: } \frac{0+1}{5+3} = 0{,}125 \qquad \textit{m-estimate } (p = \tfrac{1}{3},\ m = 3)\text{: } \frac{0+3\cdot\frac{1}{3}}{5+3} = 0{,}125$$

$P(\text{Sunny} \mid \text{No})$ bergeser dari 0,600 ke $(3+1)/8 = 0{,}500$. Hari baru tadi kini bernilai 0,0071 (Yes) lawan 0,0182 (No); keputusannya tetap No.

Korpus *email* ($n_c = 10$, $|V| = 12$, penyebut $10 + 12 = 22$):

	SPAM		BUKAN SPAM	
Kata	mentah	Laplace	mentah	Laplace
gratis	$3/10 = 0{,}300$	$4/22 = 0{,}182$	0	$1/22 = 0{,}045$
klik, hadiah	$2/10 = 0{,}200$	$3/22 = 0{,}136$	0	$1/22 = 0{,}045$
rapat, besok	0	$1/22 = 0{,}045$	$2/10 = 0{,}200$	$3/22 = 0{,}136$

Keputusan penyaring *spam* dan perhitungan log

Email **“gratis klik hadiah”** dengan taksiran Laplace:

$$\text{SPAM:}\quad 0{,}5 \times \tfrac{4}{22} \times \tfrac{3}{22} \times \tfrac{3}{22} = 1{,}69 \times 10^{-3} \qquad \text{BUKAN:}\quad 0{,}5 \times \tfrac{1}{22} \times \tfrac{1}{22} \times \tfrac{1}{22} = 4{,}70 \times 10^{-5}$$

Perbandingannya 36 : 1, jadi $P(\text{SPAM} \mid \textit{email}) = 36/37 = 0{,}973$. “Gratis rapat besok” yang tadi macet di nol kini memberi $1{,}88 \times 10^{-4}$ lawan $4{,}23 \times 10^{-4}$, sehingga peluang *spam*-nya 0,308: bukan *spam*.

Underflow dan *log-sum-exp*

Email 200 kata berarti mengalikan 200 bilangan sekitar 10^{-4}. Hasilnya 10^{-800}, di bawah batas *double-precision floating point* (sekitar 10^{-308}), dan menjadi nol. Implementasi nyata bekerja dalam log:

$$v_{NB} = \arg\max_{v} \left[\log P(v) + \sum_{i} \log P(a_i \mid v) \right]$$

Untuk mendapatkan peluang, kurangi semua skor dengan skor terbesar sebelum dieksponenkan.

Kode: *naive Bayes* multinomial dari nol (1)

```python
import math
from collections import Counter

def latih_nb(data, alpha=1.0):
    kelas = Counter(c for _, c in data)
    hitung = {c: Counter() for c in kelas}
    for teks, c in data:
        hitung[c].update(teks.split())
    V = {w for c in hitung for w in hitung[c]}          # kosakata
    log_prior = {c: math.log(n / len(data)) for c, n in kelas.items()}
    log_lik = {}
    for c in kelas:
        penyebut = sum(hitung[c].values()) + alpha * len(V)
        log_lik[c] = {w: math.log((hitung[c][w] + alpha) / penyebut) for w in V}
        log_lik[c]["<unk>"] = math.log(alpha / penyebut)  # kata baru
    return log_prior, log_lik
```

`latih_nb` hanya menghitung: *prior* log per kelas, lalu log peluang setiap kata dengan Laplace (`alpha=1`). Token `<unk>` menampung kata yang tidak ada di kosakata.

Sumber: `bab05-bayesian-naive-bayes/naive_bayes_manual.py`.

Kode: *naive Bayes* multinomial dari nol (2)

```
def prediksi(model, teks):
    log_prior, log_lik = model
    skor = {c: log_prior[c] + sum(log_lik[c].get(w, log_lik[c]["<unk>"])
                                  for w in teks.split()) for c in log_prior}
    m = max(skor.values())                          # log-sum-exp
    z = sum(math.exp(s - m) for s in skor.values())
    return {c: math.exp(s - m) / z for c, s in skor.items()}
# prediksi(model, "gratis klik hadiah") -> spam 0.973
```

Skor setiap kelas adalah jumlah log, sehingga *email* sepanjang apa pun aman dari *underflow*. Baris `m = max(...)` adalah trik *log-sum-exp*: semua skor dikurangi skor terbesar sebelum dieksponenkan, lalu dinormalkan menjadi peluang.

Kode: `MultinomialNB` di scikit-learn

```python
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline

teks = ["menang hadiah gratis", "gratis hadiah klik sekarang",
        "klik gratis menang", "rapat besok pagi", "jadwal rapat proyek",
        "kirim laporan proyek besok"]
label = ["spam"] * 3 + ["bukan"] * 3
model = make_pipeline(CountVectorizer(), MultinomialNB(alpha=1.0)).fit(teks, label)
print(model.predict_proba(["gratis klik hadiah"]))  # [[0.027 0.973]]

vek, nb = model[0], model[1]
i = list(nb.classes_).index("spam")
bukti = nb.feature_log_prob_[i] - nb.feature_log_prob_[1 - i]
kata = vek.get_feature_names_out()
print([kata[j] for j in bukti.argsort()[::-1][:3]])  # gratis, klik, menang
```

Angka 0,973 sama dengan hitungan tangan. Selisih $\log P(w \mid \text{spam}) - \log P(w \mid \text{bukan})$ menunjukkan kata dengan bukti terkuat.

Sumber: `bab05-bayesian-naive-bayes/naive_bayes_sklearn.py`.

Mengapa asumsi yang salah tetap bekerja

- **Keputusan hanya butuh urutan.** Selama kelas yang benar tetap teratas pada arg max, besaran peluangnya boleh melenceng.
- **Hitungan ganda** dari atribut yang berkorelasi sering membesarkan kelas yang memang didukung bukti itu.
- **Sedikit parameter, *variance* kecil.** Pada data sedikit, pertukaran bias–*variance* ini menguntungkan.

Hati-hati dengan angka peluangnya

Naive Bayes terkenal *overconfident* (0,999 atau 0,001). Untuk keputusan berbiaya, kalibrasi dulu dengan `CalibratedClassifierCV`.

Varian	*Feature* dan taksiran
Gaussian	numerik kontinu; $\mu_{i,v}$ dan $\sigma_{i,v}$ per kelas
Multinomial	hitungan kata; $(n_{v,w} + \alpha)/(n_v + \alpha\lvert V\rvert)$
Bernoulli	biner; ketidakhadiran kata ikut dihitung sebagai bukti

Cost latih $O(nd)$, satu lintasan. *Prediction* $O(dc)$ untuk c kelas. Karena semurah itu, *naive Bayes* layak dijadikan *baseline*.

Sumber: Domingos dan Pazzani (1997); McCallum dan Nigam (1998).

Dari data berlabel ke data tanpa label

Gaussian naive Bayes sebagai jembatan

Untuk atribut numerik (suhu, saldo), tabel frekuensi diganti kepadatan normal per kelas:

$$P(a_i \mid v) = \frac{1}{\sqrt{2\pi\sigma_{i,v}^2}} \exp\left(-\frac{(a_i - \mu_{i,v})^2}{2\sigma_{i,v}^2}\right)$$

Notasi: μ (dibaca *mu*): rata-rata; σ (*sigma*): simpangan baku; π (*pi*) $\approx 3{,}14$.

Bila densitas **seluruh** data ditulis menurut model itu, hasilnya *weighted sum* dari beberapa Gaussian:

$$p(\mathbf{x}) = \sum_{v} P(v)\,\mathcal{N}\big(\mathbf{x} \mid \boldsymbol{\mu}_v, \operatorname{diag}(\sigma_{1,v}^2, \ldots, \sigma_{d,v}^2)\big)$$

Label diketahui

Pisahkan data per kelas, lalu hitung rata-rata dan *variance* setiap kelompok (`GaussianNB`). Pada Iris: *accuracy* 0,953 dengan *5-fold CV*.

Label hilang

Rata-rata per kelas tak bisa dihitung tanpa label, dan label tak bisa ditebak tanpa rata-rata. Label menjadi **variabel laten**; EM memecah lingkaran ini.

Supervised dan *unsupervised learning*

	Supervised	***Unsupervised***
Data	pasangan $(\mathbf{x}_n, y_n)$	hanya $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
Contoh model	*decision tree*, *naive Bayes*	GMM, *K-Means*
Tugas	*classification*, *regression*	*density estimation*, *clustering*
Menilai mutu	*accuracy*, *precision*, RMSE	*likelihood* validasi, BIC, kegunaan

- ***Density estimation***: menaksir $p(\mathbf{x})$, untuk menilai seberapa wajar titik baru atau membangkitkan data sintetis.
- ***Clustering***: mengelompokkan titik yang saling mirip tanpa label.
- Pada model campuran keduanya bertemu: setiap komponen $p(\mathbf{x})$ ditafsirkan sebagai satu kelompok.

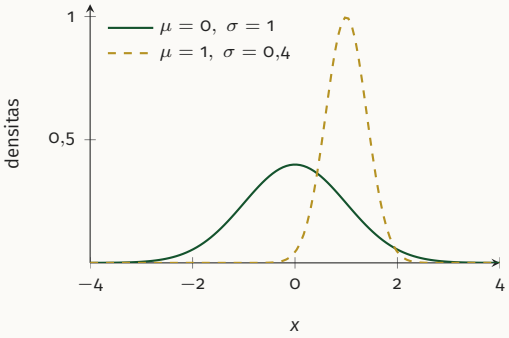
Data tanpa label paling sering dijumpai: jutaan transaksi bank tanpa segmen, ribuan jam audio tanpa transkrip.

Distribusi Gaussian

Gaussian univariat

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- Puncak di μ, lebar ditentukan σ.
- Sekitar 68% massa di $\mu \pm \sigma$, 95% di $\mu \pm 2\sigma$, 99,7% di $\mu \pm 3\sigma$.
- Faktor $1/\sqrt{2\pi\sigma^2}$ membuat luas di bawah kurva sama dengan satu.
- Untuk σ kecil nilai densitas di puncak dapat melebihi 1. Densitas berbeda dari peluang, dan sifat ini nanti memicu singularitas GMM.

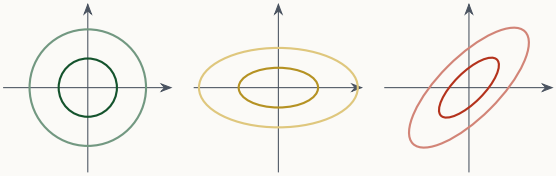


Sumber: Bishop Subbab 1.2.4 dan 2.3; buku ajar Subbab 6.2.

Gaussian multivariat dan jarak Mahalanobis

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\tfrac{1}{2}\Delta^2\right), \qquad \Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

- Σ_{ii}: *variance feature* i; Σ_{ij}: *covariance feature* i dan j.
- $\boldsymbol{\Sigma}$ (*sigma* kapital) simetris dan *positive definite*.
- Δ (*delta* kapital): jarak Mahalanobis. Kontur Δ tetap berbentuk elips.
- Dari $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ ($\boldsymbol{\Lambda}$: *lambda* kapital): vektor eigen memberi arah sumbu, $\sqrt{\lambda_i}$ panjang setengah sumbunya. Untuk $\Sigma_{12} = 1{,}5$ (diagonal 2): $\lambda = 3{,}5$ dan $0{,}5$.



(a) $\boldsymbol{\Sigma} = \mathbf{I}$ (b) diag $(4;\ 1)$ (c) $\Sigma_{12} = 1{,}5$

Kontur $\Delta = 1$ (tebal) dan $\Delta = 2$ (muda).

Log-likelihood dan MLE satu Gaussian

Untuk data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ yang independen:

$$\ln p(\mathbf{X} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{Nd}{2}\ln(2\pi) - \frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N}\Delta_n^2$$

Suku kedua menghukum *covariance* yang terlalu lebar, suku ketiga menghukum titik jauh dari pusat. *Derivative* terhadap $\boldsymbol{\mu}$ disamakan dengan nol memberi **bentuk tertutup**:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n, \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \hat{\boldsymbol{\mu}})(\mathbf{x}_n - \hat{\boldsymbol{\mu}})^{\top}$$

Data nasabah sintetis ($N = 500$, $d = 2$)

$x_1 = \log_{10}$ saldo, x_2 = transaksi digital per bulan; tiga segmen (250, 200, 50). Satu Gaussian: $\hat{\boldsymbol{\mu}} = (6{,}393;\ 24{,}965)$, *log-likelihood* $-2234{,}45$.

Pembagi N atau $N-1$

MLE membagi dengan N; `np.cov` memakai $N-1$, tambahkan `bias=True` untuk MLE.

Gaussian mixture model

Definisi GMM

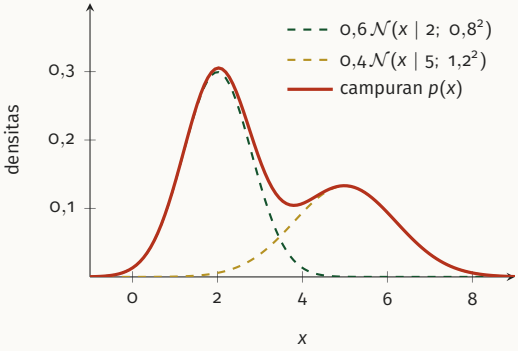
GMM dengan K komponen

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$\pi_k \geq 0$, $\sum_k \pi_k = 1$. Parameter $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$.

- π_k: *mixing coefficient* (*weight* campuran).
- Campuran dua Gaussian dapat berpuncak dua (*bimodal*), hal yang mustahil bagi satu Gaussian.
- Dengan komponen cukup banyak, GMM mendekati hampir semua densitas kontinu yang *smooth*.

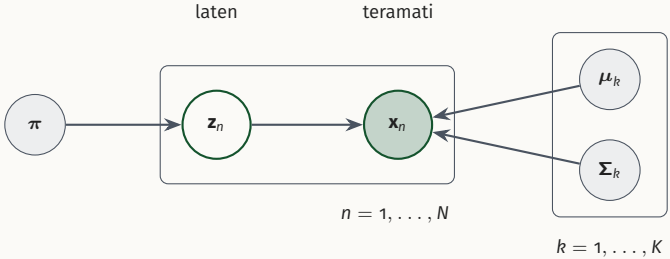
Sumber: Bishop Subbab 2.3.9 dan 9.2.



Variabel laten: cerita pembangkitan data

Komponen asal titik ke-n ditulis $\mathbf{z}_n = (z_{n1}, \ldots, z_{nK})$ dengan tepat satu unsur bernilai 1 (*1-of-K*). Untuk setiap titik:

1. pilih komponen dengan peluang $p(z_{nk} = 1) = \pi_k$;
2. ambil $\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ dari komponen terpilih ($\sim$ dibaca "berdistribusi"; $\mathcal{N}$: Gaussian).



Yang teramati hanya $\mathbf{x}_n$. Menjumlahkan $p(\mathbf{x}, z_k = 1) = \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ atas semua k (*marginalization*) memberi kembali $p(\mathbf{x})$.

Responsibility: teorema Bayes untuk komponen

Posterior komponen setelah melihat x

$$\gamma(z_k) \equiv p(z_k = 1 \mid \mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- π_k berperan sebagai *prior*, $\mathcal{N}(\cdot)$ sebagai *likelihood*; rumus yang sama dengan uji penyakit.
- $\sum_k \gamma(z_k) = 1$ untuk setiap titik.
- GMM memberi ***soft assignment***.

Contoh: nasabah baru

Nasabah (6,2; 27) mendapat *responsibility* (0,717; 0,283; 0): 71,7% segmen aktif bersaldo kecil, 28,3% segmen karyawan. Informasi “seberapa yakin” ini hilang bila hanya memakai `predict`.

Mengapa MLE campuran tidak punya bentuk tertutup

$$\ln p(\mathbf{X} \mid \boldsymbol{\theta}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

Ada penjumlahan di dalam logaritma, sehingga log tidak bisa "masuk" ke setiap Gaussian. *Derivative* terhadap $\boldsymbol{\mu}_k$ disamakan nol memberi

$$\boldsymbol{\mu}_k = \frac{\sum_n \gamma(z_{nk})\, \mathbf{x}_n}{\sum_n \gamma(z_{nk})},$$

padahal $\gamma(z_{nk})$ sendiri bergantung pada $\boldsymbol{\mu}_k$. Kesulitan lain:

- **Banyak optimum**: fungsinya tidak konveks.
- ***Label switching***: menukar nomor komponen memberi densitas sama ($K!$ susunan setara), jadi komponen dinamai dari isinya.
- **Singularitas**: komponen di atas satu titik dengan *variance* $\to$ 0 membuat *log-likelihood* tak hingga.

Seandainya γ diketahui, $\boldsymbol{\mu}_k$ mudah dihitung; seandainya parameter diketahui, γ mudah dihitung. EM bergantian di antara keduanya.

Algoritma EM

E-step dan M-step

E-step (*expectation*): dengan parameter saat ini, hitung

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

M-step (*maximization*): anggap $\gamma(z_{nk})$ tetap, lalu

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}), \qquad \pi_k^{\text{baru}} = \frac{N_k}{N},$$

$$\boldsymbol{\mu}_k^{\text{baru}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})\, \mathbf{x}_n, \qquad \boldsymbol{\Sigma}_k^{\text{baru}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{baru}})(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{baru}})^\top$$

- N_k: banyaknya titik efektif komponen k (tidak harus bulat).
- $\gamma(z_{nk}) \in \{0, 1\}$: M-step = MLE Gaussian per kelas berlabel (QDA bila *covariance* penuh, *Gaussian naive Bayes* bila diagonal).

Sumber: Bishop Subbab 9.2.2.

Pseudocode EM untuk GMM

Algoritma: EM untuk *Gaussian mixture model*

***Input*:** $\mathbf{x}_1, \ldots, \mathbf{x}_N$; K; *tolerance* ε; *regularization* $\lambda \geq 0$

1 *initialization* $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ ▷ acak, atau dari *K-Means*
2 $\ell_{\text{lama}} \leftarrow -\infty$
3 **repeat**
4 $\quad a_{nk} \leftarrow \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ ▷ E-step
5 $\quad \ell_n \leftarrow \text{logsumexp}_k(a_{nk})$; $\ell \leftarrow \sum_n \ell_n$
6 $\quad \gamma(z_{nk}) \leftarrow \exp(a_{nk} - \ell_n)$
7 $\quad N_k \leftarrow \sum_n \gamma(z_{nk})$; $\pi_k \leftarrow N_k / N$ ▷ M-step
8 $\quad \boldsymbol{\mu}_k \leftarrow \frac{1}{N_k} \sum_n \gamma(z_{nk}) \mathbf{x}_n$
9 $\quad \boldsymbol{\Sigma}_k \leftarrow \frac{1}{N_k} \sum_n \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top + \lambda \mathbf{I}$
10 $\quad \Delta \leftarrow \ell - \ell_{\text{lama}}$; $\ell_{\text{lama}} \leftarrow \ell$
11 **until** $\Delta < \varepsilon$ atau iterasi maksimum
12 **return** $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$, $\gamma(z_{nk})$

- Baris 4–6: E-step di ruang log. $\ln \sum_k e^{a_k} = m + \ln \sum_k e^{a_k - m}$ dengan $m = \max_k a_k$, teknik yang sama dengan *naive Bayes*.
- Tanpa log, titik yang jauh dari semua komponen memberi $0/0$.
- $\lambda \mathbf{I}$ di baris 9 mencegah singularitas.

Contoh tangan: data $\{1, 2, 3, 6, 7, 8\}$, E-step iterasi 1

Dua komponen, tebakan awal sengaja kurang tepat: $\pi_1 = \pi_2 = 0{,}5$, $\mu_1 = 2$, $\mu_2 = 4$, $\sigma_1^2 = \sigma_2^2 = 1$. *Weight* dan faktor *normalization* saling menghapus:

$$\gamma(z_{n1}) = \frac{\mathcal{N}(x_n \mid 2, 1)}{\mathcal{N}(x_n \mid 2, 1) + \mathcal{N}(x_n \mid 4, 1)}$$

x_n	$\mathcal{N}(x_n \mid 2, 1)$	$\mathcal{N}(x_n \mid 4, 1)$	$\gamma(z_{n1})$	$\gamma(z_{n2})$
1	0,24197	0,00443	0,9820	0,0180
2	0,39894	0,05399	0,8808	0,1192
3	0,24197	0,24197	0,5000	0,5000
6	$1{,}34 \times 10^{-4}$	0,05399	0,0025	0,9975
7	$1{,}49 \times 10^{-6}$	0,00443	0,0003	0,9997
8	$6{,}08 \times 10^{-9}$	$1{,}34 \times 10^{-4}$	0,0000	1,0000
jumlah (N_k)			2,3657	3,6343

Untuk $x = 2$:
$0{,}39894/(0{,}39894 + 0{,}05399) = 0{,}8808$.

Titik 3 berjarak sama dari kedua rata-rata, jadi $\gamma(z_{31}) = 0{,}5$ tepat.

Titik 6, 7, 8 lebih dekat ke $\mu_2 = 4$, sehingga hampir seluruhnya milik komponen 2. Densitas $6{,}08 \times 10^{-9}$ menunjukkan alasan E-step dihitung di ruang log.

Contoh tangan: M-step iterasi 1 dan seterusnya

Dari tabel E-step:

$$\pi_1 = 2{,}3657/6 = 0{,}3943, \qquad \pi_2 = 3{,}6343/6 = 0{,}6057,$$

$$\mu_1 = \frac{0{,}9820(1) + 0{,}8808(2) + 0{,}5(3) + 0{,}0025(6) + \ldots}{2{,}3657} = \frac{4{,}2612}{2{,}3657} = 1{,}8013,$$

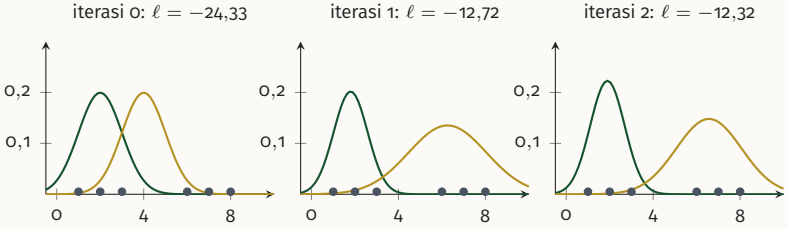
$$\mu_2 = \frac{22{,}7388}{3{,}6343} = 6{,}2567,$$

$$\sigma_1^2 = \frac{1{,}4381}{2{,}3657} = 0{,}6079, \qquad \sigma_2^2 = \frac{11{,}6170}{3{,}6343} = 3{,}1964.$$

Log-likelihood naik dari $-24{,}3314$ menjadi $-12{,}7183$.

- Iterasi 2: $\gamma(z_{31})$ naik dari 0,5 ke 0,7063; $\pi = (0{,}4421;\ 0{,}5579)$, $\mu = (1{,}8949;\ 6{,}5641)$, $\sigma^2 = (0{,}6267;\ 2{,}2626)$, $\ell = -12{,}3179$.
- Setelah enam iterasi: $\pi = (0{,}5;\ 0{,}5)$, $\mu = (2;\ 7)$, $\sigma^2 = (0{,}6667;\ 0{,}6667)$, yaitu MLE untuk $\{1, 2, 3\}$ dan $\{6, 7, 8\}$ secara terpisah: $\frac{1}{3}(1 + 0 + 1) = 2/3$.

Perjalanan EM pada data $\{1, 2, 3, 6, 7, 8\}$



Kurva hijau dan emas: $\pi_k \mathcal{N}(x \mid \mu_k, \sigma_k^2)$; titik abu-abu: data.

Urutan ℓ: $-24{,}33 \to -12{,}72 \to -12{,}32 \to -11{,}69 \to -11{,}46$, selalu naik.

Mengapa *log-likelihood* tidak pernah turun

Untuk sembarang distribusi $q(\mathbf{Z})$ atas variabel laten:

$$\ln p(\mathbf{X} \mid \boldsymbol{\theta}) = \underbrace{\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{q(\mathbf{Z})}}_{\mathcal{L}(q, \boldsymbol{\theta})} + \underbrace{\mathrm{KL}\big(q \,\|\, p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})\big)}_{\geq 0}$$

Jadi $\mathcal{L}(q, \boldsymbol{\theta})$ selalu **batas bawah** (*lower bound*) bagi *log-likelihood*.

- **E-step**: $q = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{\text{lama}})$ (*responsibility*). KL menjadi nol, batas bawah menyentuh *log-likelihood*.
- **M-step**: maksimumkan $\mathcal{L}$ terhadap $\boldsymbol{\theta}$ dengan q tetap; batas bawah naik atau tetap.

$$\ln p(\mathbf{X} \mid \boldsymbol{\theta}^{\text{baru}}) \geq \mathcal{L}(q, \boldsymbol{\theta}^{\text{baru}}) \geq \mathcal{L}(q, \boldsymbol{\theta}^{\text{lama}}) = \ln p(\mathbf{X} \mid \boldsymbol{\theta}^{\text{lama}})$$

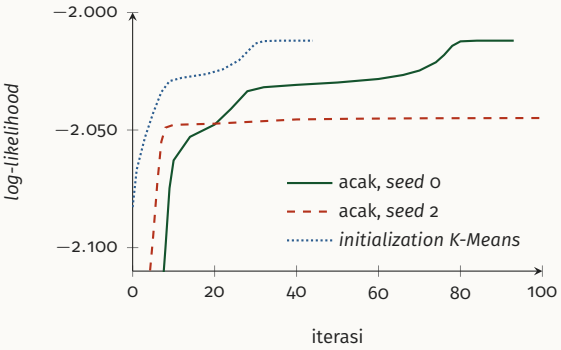
Dipakai saat menulis kode

Bila *log-likelihood* implementasi Anda turun lebih dari *error* pembulatan, hampir pasti ada *bug*. Jaminannya hanya sampai titik stasioner, biasanya maksimum lokal, tidak sampai maksimum global.

Optimum lokal dan *initialization*

Data nasabah, $K = 3$, sepuluh *seed* acak: lima berakhir di $\ell = -2012{,}04$ (tiga segmen asli), lima berhenti sekitar $-2044{,}62$ (segmen aktif dipecah dua, karyawan dan prioritas digabung).

- **Titik data acak**: murah, separuh percobaan terjebak.
- ***K-Means*** (bawaan scikit-learn): mulai dari $-2082{,}8$, mencapai $-2012{,}04$ dalam 45 iterasi.
- **Mulai ulang** (`n_init`): peluang 10 percobaan acak semuanya terjebak kira-kira $0{,}5^{10} \approx 0{,}001$.



Kurva *seed* 0 hampir datar di iterasi 30–70 lalu naik lagi; `tol` yang longgar menghentikan EM di dataran itu.

Memilih model

Bentuk *covariance* dan singularitas

Bentuk	*Covariance*	Kapan dipakai	Total
`full`	$K\,d(d+1)/2$	data banyak, *feature* berkorelasi	17
`tied`	$d(d+1)/2$	bentuk sama, posisi beda	11
`diag`	$K\,d$	dimensi tinggi (audio)	14
`spherical`	K	*feature* distandardisasi	11

Kolom Total: data nasabah ($K = 3$, $d = 2$), termasuk 6 rata-rata dan 2 *weight* bebas. `diag` adalah asumsi *naive Bayes* per komponen.

Singularitas

Komponen kedua diletakkan di $x = 1$ pada data $\{1, 2, 3, 6, 7, 8\}$ dan *variance*-nya dikecilkan:

variance	1	10^{-8}	10^{-16}
ℓ	$-14{,}87$	$-7{,}41$	$1{,}80$

Tanpa batas atas. Penangkalnya `reg_covar` ($\lambda\mathbf{I}$, bawaan 10^{-6}): dengan $\lambda = 10^{-2}$, ℓ tertahan di $-14{,}30$.

Penyebab umum singularitas: baris ganda, *feature* hampir konstan, K terlalu besar, atau $d > N_k$. Periksa itu dulu sebelum menaikkan `reg_covar`.

Memilih jumlah komponen dengan BIC

Log-likelihood latih selalu naik bila K ditambah. Kriteria informasi menambahkan hukuman untuk p parameter:

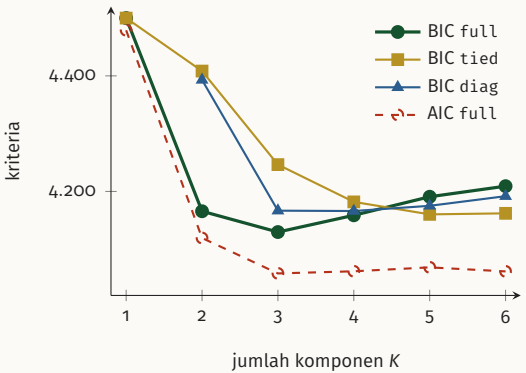
$$\mathrm{AIC} = -2 \ln \hat{L} + 2p$$
$$\mathrm{BIC} = -2 \ln \hat{L} + p \ln N$$

Makin kecil makin baik. Untuk $N \geq 8$, $\ln N > 2$, jadi BIC menghukum lebih berat.

$K = 3$, `full`, $p = 17$, $N = 500$:

$$\mathrm{BIC} = 4024{,}08 + 17(6{,}2146) = 4129{,}73$$

$K = 4$: $\ell = -2007{,}9$ tetapi BIC 4158,7; kenaikan ℓ sekitar 4 tidak menutup hukuman $6 \ln 500 = 37{,}3$.



BIC terkecil: `full`, $K = 3$, sesuai cara data dibangkitkan.

Campuran Bernoulli untuk data biner

Untuk $\mathbf{x} \in \{0,1\}^D$ (Bishop Subbab 9.3.3):

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \prod_{i=1}^{D} \mu_{ki}^{x_i} (1-\mu_{ki})^{1-x_i}$$

Di dalam satu komponen *feature* dianggap independen, persis *Bernoulli naive Bayes*, hanya kelasnya laten. M-step: $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_n \gamma(z_{nk})\, \mathbf{x}_n$.

Contoh: delapan nasabah, empat layanan (QRIS, *top-up*, deposito, reksa dana)

Pola: (1, 1, 0, 0), (1, 1, 0, 0), (1, 1, 1, 0), (1, 0, 0, 0), (0, 0, 1, 1), (0, 1, 1, 1), (0, 0, 1, 1), (1, 0, 1, 1). Awal $\boldsymbol{\mu}_1 = (0{,}6;\ 0{,}5;\ 0{,}4;\ 0{,}5)$, $\boldsymbol{\mu}_2 = (0{,}4;\ 0{,}5;\ 0{,}6;\ 0{,}5)$, $\pi = (0{,}5;\ 0{,}5)$.

Nasabah 1: $0{,}6 \cdot 0{,}5 \cdot 0{,}6 \cdot 0{,}5 = 0{,}09$ lawan $0{,}4 \cdot 0{,}5 \cdot 0{,}4 \cdot 0{,}5 = 0{,}04$, jadi $\gamma(z_{11}) = 0{,}09/0{,}13 = 0{,}692$.

Setelah delapan iterasi: $\boldsymbol{\mu}_1 \approx (1;\ 0{,}75;\ 0{,}25;\ 0)$ (pembayaran digital) dan $\boldsymbol{\mu}_2 \approx (0{,}25;\ 0{,}25;\ 1;\ 1)$ (investasi).

Peluang Bernoulli paling besar 1, sehingga *likelihood*-nya terbatas dan singularitas tidak terjadi.

Implementasi

Kode: EM untuk GMM dengan NumPy (1)

```python
import numpy as np
from scipy.special import logsumexp
from gaussian_mle import log_gauss

def em_gmm(X, K, n_iter=500, tol=1e-6, reg_covar=1e-6, seed=0, init="acak"):
    n, d = X.shape
    rng = np.random.default_rng(seed)
    pi, mu, Sigma = inisialisasi(X, K, init, rng)
    riwayat = []
    for it in range(n_iter):
        # E-step: log(pi_k N(x_n|mu_k,Sigma_k)), lalu normalkan per baris
        log_p = np.column_stack([np.log(pi[k]) + log_gauss(X, mu[k], Sigma[k])
                                 for k in range(K)])
        log_px = logsumexp(log_p, axis=1)          # log p(x_n)
        riwayat.append(log_px.sum())               # log-likelihood
        R = np.exp(log_p - log_px[:, None])        # responsibility, n x K
```

E-step di ruang log: kolom ke-k `log_p` berisi $\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, `logsumexp` memberi $\ln p(\mathbf{x}_n)$, dan `R` adalah matriks *responsibility* $N \times K$ yang barisnya berjumlah satu.

Sumber: `bab06-gmm-em/gmm_numpy.py`.

Kode: EM untuk GMM dengan NumPy (2)

```
        # M-step
        Nk = R.sum(axis=0)
        pi = Nk / n
        mu = (R.T @ X) / Nk[:, None]
        for k in range(K):
            D = X - mu[k]
            Sigma[k] = (R[:, k, None] * D).T @ D / Nk[k] + reg_covar * np.eye(d)
        if it > 0 and riwayat[-1] - riwayat[-2] < tol:
            break
    return pi, mu, Sigma, R, riwayat
```

Output `em_gmm(X, K=3, seed=0)`, komponen diurutkan menurut saldo:

```
iterasi: 94
pi   : [0.504 0.396 0.1  ]
mu   :
 [[ 5.802 33.823]
 [ 6.791 18.052]
 [ 7.801  7.695]]
log-likelihood tidak pernah turun: True
```

- M-step: `Nk`, `pi`, `mu`, dan `Sigma` adalah *weighted average* dengan `R`; `reg_covar` menambah $\lambda\mathbf{I}$.
- Pembangkit data: *weight* (0,5; 0,4; 0,1), rata-rata (5,8; 34), (6,8; 18), (7,8; 8).
- $\ell = -2012{,}04$, lebih tinggi 222 satuan daripada satu Gaussian ($-2234{,}45$); BIC menilai apakah kenaikan itu sepadan.

Kode: `GaussianMixture` di scikit-learn

```python
import numpy as np
from sklearn.mixture import GaussianMixture
from data_nasabah import buat_data

X, y = buat_data()
gmm = GaussianMixture(n_components=3, covariance_type="full", tol=1e-6,
                      max_iter=500, n_init=5, random_state=0).fit(X)
urut = np.argsort(gmm.means_[:, 0])      # namai komponen menurut saldo
print("bobot   :", gmm.weights_[urut].round(3))
print(f"logL total: {gmm.score(X) * len(X):.2f}")  # score = rata-rata per titik

baru = np.array([[5.9, 33.0], [6.2, 27.0], [7.3, 11.0]])
P = gmm.predict_proba(baru)[:, urut]     # responsibility
for b, p in zip(baru, P):
    print(f"nasabah {b}: responsibility = {p.round(3)}")
```

```
bobot   : [0.504 0.396 0.1  ]
logL total: -2012.04
nasabah [ 5.9 33. ]: responsibility = [1. 0. 0.]
nasabah [ 6.2 27. ]: responsibility = [0.717 0.283 0.   ]
nasabah [ 7.3 11. ]: responsibility = [0.    0.827 0.173]
```

Dengan bawaan `tol=1e-3` dan `max_iter=100`, model berhenti setelah 11 iterasi di $-2027{,}07$. Periksa `converged_` dan `n_iter_`. ARI terhadap segmen asli: 0,966.

Praktikum Pertemuan 5

Bagian A: *naive Bayes*

1. Jalankan kode *naive Bayes* manual dan `MultinomialNB`; pastikan peluang *spam* “gratis klik hadiah” $0{,}973$.
2. Pada SMS Spam Collection, latih `MultinomialNB` dan `BernoulliNB` untuk `alpha` $0{,}01$; $0{,}1$; 1. Laporkan *precision*, *recall*, dan sepuluh kata dengan bukti terkuat.
3. Gambar kurva kalibrasi sebelum dan sesudah `CalibratedClassifierCV`.

Bagian B: GMM dan EM

4. Tulis E-step dan M-step 1D; jalankan pada $\{1, 2, 3, 6, 7, 8\}$ dan cocokkan angka iterasi 1 dan 2 dengan slide.
5. Jalankan `em_gmm` pada data nasabah untuk sepuluh *seed*; catat *log-likelihood* akhir dan periksa bahwa tidak pernah turun.
6. Pakai `GaussianMixture` pada Iris yang distandardisasi; pilih bentuk *covariance* dan K dengan BIC, lalu hitung ARI terhadap spesies.

Yang dikumpulkan

Notebook yang dapat dijalankan ulang dari awal, beserta satu halaman pembahasan: varian *naive Bayes* yang Anda pilih, berapa *seed* yang terjebak optimum lokal, dan apakah BIC serta ARI sepakat.

Kriteria penilaian

Naive Bayes dan kalibrasi	25%
EM 1D sesuai hitungan tangan	25%
Analisis optimum lokal	20%
Pemilihan model dengan BIC	20%
Kerapian *notebook*	10%

Ringkasan

Topik	Pokok yang perlu diingat
Teorema Bayes	$P(h \mid D) \propto P(D \mid h)P(h)$. Uji penyakit: positif berarti 21% sakit; uji kedua menaikkan ke 0,896.
MAP, ML, *Bayes optimal*	$h_{MAP} = h_{ML}$ bila *prior* seragam; *least squares* dan *cross-entropy* adalah ML. *Bayes optimal* terbaik secara teori, Gibbs paling buruk dua kali *error*-nya.
Naive Bayes	$v_{NB} = \arg\max_v P(v) \prod_i P(a_i \mid v)$; PlayTennis memberi No ($0{,}0206 > 0{,}0053$). Laplace mencegah nol, log mencegah *underflow*.
Gaussian	MLE satu Gaussian berbentuk tertutup; $\boldsymbol{\Sigma}$ menentukan elips; *Gaussian naive Bayes* adalah campuran yang labelnya diketahui.
GMM dan EM	$p(\mathbf{x}) = \sum_k \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$; E-step menghitung $\gamma(z_{nk})$, M-step *weighted average*. Data $\{1, 2, 3, 6, 7, 8\}$ konvergen ke $\mu = (2;\ 7)$, $\sigma^2 = 2/3$.
Optimum lokal dan BIC	ℓ tidak pernah turun, tetapi separuh *seed* acak berhenti di $-2044{,}62$. Pakai *K-Means*, `n_init`, `tol` ketat. BIC memilih `full`, $K = 3$ (4129,7).

Sumber: Mitchell Bab 6; Bishop Bab 2 dan 9; Dempster, Laird, dan Rubin (1977).

Kuis dan tugas

1. Prevalensi 0,002, sensitivitas 95%, spesifisitas 90%. Hitung $P(\text{sakit} \mid +)$. $\approx 0{,}0187$
2. Koin: 7 angka dari 10 lemparan. Hitung $\hat{\theta}_{ML}$ dan $\hat{\theta}_{MAP}$ dengan Beta$(5, 5)$. $0{,}7$; $11/18 \approx 0{,}611$
3. Hitung keputusan *naive Bayes* untuk $\langle$Overcast, Hot, Normal, Weak$\rangle$, tanpa dan dengan Laplace.
4. Mengapa *naive Bayes* tetap akurat walaupun asumsinya dilanggar, dan kapan peluangnya perlu dikalibrasi?
5. Mulai EM pada $\{1, 2, 3, 6, 7, 8\}$ dari $\mu_1 = \mu_2 = 4{,}5$ (*weight* dan *variance* sama). Apa yang terjadi setelah satu iterasi, dan mengapa?
6. Hitung jumlah parameter GMM $K = 8$, $d = 13$ untuk keempat bentuk *covariance*. `full` 839; `diag` 215

Tugas dan pertemuan depan

Bacaan: buku ajar Bab 5 dan Bab 6, Bishop Subbab 9.1–9.3. Kerjakan latihan 3 dan 4 Bab 6 (lanjutan iterasi EM, *initialization* simetris). Pertemuan 6 membahas kNN dan *K-Means*; *K-Means* ternyata kasus batas GMM dengan $\boldsymbol{\Sigma}_k = \sigma^2 \mathbf{I}$ dan $\sigma^2 \to 0$.

Terima kasih

Pertanyaan?