Pertemuan 5 — Evaluasi Hipotesis dan Pembelajaran Bayesian

Membuktikan model mana yang lebih baik, lalu belajar dengan peluang

Hendri Karisma

Semester Ganjil 2026/2027

Program Studi Teknik Informatika, STMIK Tazkia

Alur pertemuan hari ini

Bagian A — Evaluasi hipotesis

1. Mengapa akurasi pada data latih menipu
2. *Sample error* dan *true error*
3. Selang kepercayaan atas galat
4. Matriks konfusi, presisi, *recall*, F1
5. Protokol latih, validasi, uji
6. *k-fold cross validation*
7. Membandingkan dua algoritma

Bagian B — Pembelajaran Bayesian

8. Teorema Bayes dan empat istilahnya
9. Hipotesis MAP dan *maximum likelihood*
10. *Maximum likelihood* dan kuadrat terkecil
11. *Minimum description length*
12. *Bayes optimal classifier* dan Gibbs

Sambungan dari Pertemuan 4

Sampai pekan lalu kita sudah punya banyak model: pohon tunggal, *random forest*, XGBoost. Semuanya keluar dengan satu angka akurasi, dan angka itu selalu terlihat meyakinkan. Hari ini kita belajar **membuktikan** model mana yang benar-benar lebih baik, lalu mengenal cara berpikir Bayesian yang menjadi dasar semua pengukuran itu.

Dari punya model menjadi punya bukti

Akurasi pada data latih hampir selalu menipu

Bayangkan Anda melatih pohon keputusan tanpa pemangkasan pada 200 baris data dan mendapat akurasi latih 100%. Apakah itu kabar baik?

- Pohon itu boleh saja menumbuhkan satu daun untuk setiap baris data. Ia tidak menemukan pola; ia **menghafal**.
- Ukuran yang kita pakai dihitung pada data yang sama yang dipakai memilih model. Itu seperti menyusun soal ujian sendiri lalu mengumumkan nilai sendiri.
- Setiap keputusan yang Anda ambil sambil melihat angka itu — kedalaman pohon, jumlah estimator, ambang batas — menempelkan sedikit informasi data tersebut ke dalam model.

Intinya

Yang ingin kita ketahui bukan “seberapa baik model mengingat masa lalu”, melainkan “seberapa baik ia akan bekerja pada data yang belum pernah ia lihat”. Dua pertanyaan ini berbeda, dan yang kedua tidak pernah bisa dijawab dengan pasti, hanya ditaksir.

Sample error dan true error

Sample error — yang bisa dihitung

Proporsi salah pada satu sampel S berukuran n:

$$\text{error}_S(h) = \frac{1}{n} \sum_{x \in S} \delta\big(f(x) \neq h(x)\big)$$

δ bernilai 1 bila jawabannya salah, 0 bila benar. Angka inilah yang keluar dari `scikit-learn`.

True error — yang kita inginkan

Peluang model salah pada satu contoh yang diambil acak dari seluruh populasi $\mathcal{D}$:

$$\text{error}_{\mathcal{D}}(h) = \Pr_{x \sim \mathcal{D}} \big[f(x) \neq h(x)\big]$$

Angka ini **tidak pernah** kita ketahui. Ia hanya ditaksir dari sampel.

Seluruh Bab 5 Mitchell berangkat dari dua pertanyaan sederhana. Pertama, seberapa dekat $\text{error}_S(h)$ kepada $\text{error}_{\mathcal{D}}(h)$? Kedua, kalau h_1 terlihat lebih baik daripada h_2 pada satu sampel, seberapa yakin kita bahwa ia memang lebih baik?

Sumber: Tom M. Mitchell, *Machine Learning*, Bab 5.

Galat sebagai variabel acak dan selang kepercayaan

Menguji model itu sama dengan melempar koin

Ambil 40 contoh acak dari populasi, jalankan model, hitung berapa yang salah. Ulangi dengan 40 contoh lain: hasilnya hampir pasti berbeda. **Galat yang terukur adalah variabel acak**, bukan sifat tetap dari model.

Setiap pengujian adalah percobaan Bernoulli dengan peluang gagal $p = \mathrm{error}_{\mathcal{D}}(h)$, sehingga jumlah kesalahan r mengikuti **sebaran binomial**:

$$\Pr(r) = \binom{n}{r} p^r (1-p)^{n-r}$$

- $E[r] = np$, sehingga $E[\mathrm{error}_S(h)] = p$. Artinya error_S adalah **estimator tak bias**: rata-rata jangka panjangnya tepat pada nilai sebenarnya.
- Simpangan bakunya $\sqrt{\dfrac{p(1-p)}{n}} \approx \sqrt{\dfrac{\mathrm{err}(1-\mathrm{err})}{n}}$

Dua kesalahan yang berbeda

Bias berarti penaksirnya meleset secara sistematis — ini yang terjadi bila galat diukur pada data latih: hasilnya terlalu optimistis.

Ragam berarti penaksirnya berayun dari sampel ke sampel — ini yang terjadi bila data ujinya sedikit, walaupun penaksirnya tak bias.

Selang kepercayaan: melaporkan angka beserta ketidakpastiannya

Untuk n yang cukup besar (patokan praktis: $n \geq 30$ dan $n \cdot \mathrm{err} \geq 5$), binomial dapat dihampiri dengan sebaran normal. Selang kepercayaan $N\%$ untuk $\mathrm{error}_{\mathcal{D}}(h)$ menjadi:

Rumus yang harus Anda hafal

$$\mathrm{err} \;\pm\; z_N \sqrt{\frac{\mathrm{err}\,(1-\mathrm{err})}{n}}$$

Tingkat kepercayaan $N\%$	50%	68%	80%	90%	95%	98%	99%
Pengali z_N	0,67	1,00	1,28	1,64	1,96	2,33	2,58

Artinya: bila percobaan ini diulang banyak kali, sekitar $N\%$ dari selang yang terbentuk akan memuat galat sebenarnya. Perhatikan bahwa z_N naik pelan, sedangkan lebar selang turun hanya sebanding $1/\sqrt{n}$.

Contoh terhitung: 12 salah dari 40 sampel

Diketahui $n = 40$ dan $r = 12$, sehingga

$$\text{err} = \frac{12}{40} = 0{,}30$$

Simpangan bakunya:

$$\sqrt{\frac{0{,}30 \times 0{,}70}{40}} = \sqrt{0{,}00525} = 0{,}0725$$

Untuk tingkat kepercayaan 95%, $z_N = 1{,}96$:

$$1{,}96 \times 0{,}0725 = 0{,}142$$

Sehingga selangnya:

$$0{,}30 \pm 0{,}142 \;=\; [\,0{,}158\;;\;0{,}442\,]$$

Bacalah dengan jujur

Model ini boleh dilaporkan bergalat 30%. Tetapi dengan bukti yang kita punya, galat sebenarnya bisa saja 16%, dan bisa juga 44%.

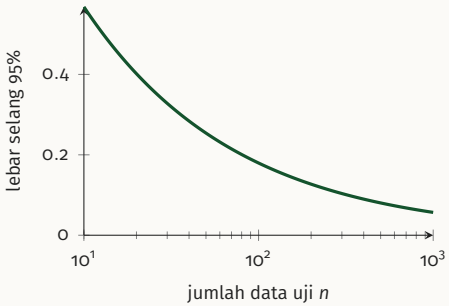
Rentang itu **hampir tiga kali** dari ujung bawah ke ujung atas. Menyatakan "akurasi model kami 70%" tanpa menyebut selangnya bukan sekadar kurang teliti, tetapi menyesatkan pembacanya.

Perhatikan juga bahwa selang ini hanya sah bila 40 sampel itu **tidak pernah** dipakai untuk melatih maupun menyetel model.

Betapa lebar selang itu ketika n kecil

n	Selang 95%	Lebar
10	[0,016 ; 0,584]	0,568
40	[0,158 ; 0,442]	0,284
100	[0,210 ; 0,390]	0,180
400	[0,255 ; 0,345]	0,090
1.000	[0,272 ; 0,328]	0,057
10.000	[0,291 ; 0,309]	0,018

Semua baris memakai $\mathrm{err} = 0{,}30$ yang sama. Yang berubah hanya banyaknya data uji.



Untuk memotong lebar selang menjadi setengah, data uji harus **empat kali** lebih banyak.

Inilah sebabnya klaim “model kami 2% lebih akurat” yang diuji pada 50 baris data sebaiknya tidak dipercaya: selangnya jauh lebih lebar daripada selisih yang diklaim.

Mengukur lebih dari sekadar akurasi

Matriks konfusi: empat angka, bukan satu

		Prediksi model	
		Positif	Negatif
Kenyataan	Positif	TP	FN
	Negatif	FP	TN

- **TP** (*true positive*) — penipuan yang tertangkap
- **FN** (*false negative*) — penipuan yang lolos
- **FP** (*false positive*) — transaksi wajar yang ditahan sia-sia
- **TN** (*true negative*) — transaksi wajar yang dilewatkan

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Presisi} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2 \cdot \text{Presisi} \cdot \text{Recall}}{\text{Presisi} + \text{Recall}}$$

Presisi menjawab: dari yang saya tuduh, berapa yang benar bersalah? **Recall** menjawab: dari yang benar bersalah, berapa yang saya tangkap?

Akurasi meringkas empat angka menjadi satu, dan dalam prosesnya membuang justru perbedaan yang paling menentukan.

Contoh terhitung: penipuan 1 dari 1.000

Dari 10.000 transaksi, 10 di antaranya penipuan. Tiga model diuji pada data yang sama.

Model	TP	FN	FP	TN	**Akurasi**	**Presisi**	**Recall**	**F_1**
A — selalu menjawab “aman”	0	10	0	9.990	**99,90%**	—	0,00	0,00
B — ambang longgar	8	2	90	9.900	99,08%	0,082	**0,80**	0,148
C — ambang ketat	5	5	10	9.980	99,85%	0,333	0,50	**0,400**

Model A punya akurasi **tertinggi** dan sama sekali tidak berguna: ia tidak pernah menangkap satu pun penipuan. Model B menangkap 8 dari 10, tetapi 90 nasabah jujur ikut tertahan. Model C hanya menangkap separuh, dengan alarm palsu yang jauh lebih sedikit.

Intinya

Pada data timpang (*imbalanced*), akurasi hampir tidak membawa informasi. Pilihan antara B dan C bukan persoalan statistika, melainkan persoalan **ongkos**: berapa nilai satu penipuan yang lolos dibanding satu nasabah jujur yang tersinggung?

Memilih metrik dengan sadar

Utamakan *recall* bila...

kesalahan melewatkan kasus positif jauh lebih mahal.

- Penapisan kanker: lebih baik memanggil pasien sehat untuk uji ulang daripada melewatkan satu tumor
- Deteksi kebocoran data, peringatan dini bencana

Utamakan presisi bila...

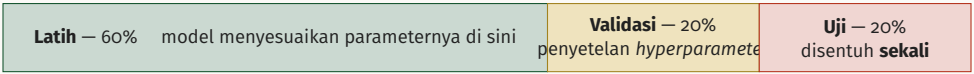
setiap tuduhan salah punya ongkos nyata.

- Penyaring *spam*: menahan satu surel penting lebih merugikan daripada meloloskan sepuluh iklan
- Pemblokiran akun, rekomendasi yang tampil di halaman depan

Presisi dan *recall* bergerak berlawanan: menurunkan ambang keputusan menaikkan *recall* dan menurunkan presisi. F_1 adalah rata-rata harmonik keduanya, dipakai ketika kita belum punya alasan untuk memihak salah satu. Bila ongkos kedua jenis kesalahan sudah diketahui angkanya, ukuran yang paling tepat justru **biaya total yang diharapkan**, bukan F_1.

Protokol evaluasi dan validasi silang

Tiga tumpuk data, tiga peran berbeda



- **Latih** — dari sini model mengambil bobot, cabang pohon, atau parameternya.
- **Validasi** — di sini Anda membandingkan kandidat: kedalaman 5 atau 10, $C = 1$ atau $C = 10$, 100 atau 500 pohon. Boleh dipakai berulang kali.
- **Uji** — di sini Anda hanya melaporkan. Sekali dipakai untuk mengambil keputusan, ia berubah menjadi data validasi, dan angkanya berhenti bisa dipercaya.

Intinya

Setiap kali Anda melihat angka pada data uji lalu mengubah sesuatu, sedikit informasi data uji itu bocor ke dalam model. Setelah dua puluh kali, “akurasi uji” Anda sudah menjadi akurasi validasi yang berbaju uji.

Kebocoran data: kegagalan yang paling sering dan paling halus

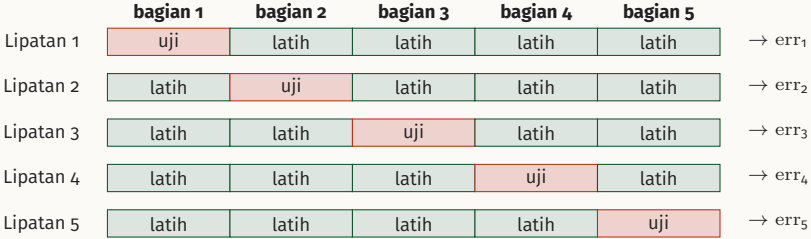
Data leakage terjadi ketika model, secara langsung atau tidak, ikut melihat informasi yang seharusnya hanya ada di masa depan atau hanya ada di data uji.

- **Normalisasi sebelum pembagian.** Menghitung rata-rata dan simpangan baku pada seluruh data, lalu baru membaginya: statistik data uji sudah masuk ke penskalaan.
- **Pengisian nilai hilang secara global**, seleksi fitur, atau *oversampling* SMOTE sebelum pembagian — bentuk kebocoran yang sama.
- **Fitur yang mengandung jawaban.** Kolom “tanggal pelunasan klaim” pada model penilai klaim.
- **Pembagian acak pada data berurut waktu.** Model jadi boleh melihat hari Kamis untuk memprediksi hari Rabu; untuk runtun waktu, potong menurut waktu.
- **Duplikat dan kelompok yang terpecah.** Satu pasien dengan lima kunjungan harus seluruhnya berada di satu sisi.

Intinya

Gejalanya khas: akurasi validasi tinggi dan mencurigakan rapi, lalu jatuh begitu dipakai sungguhan. Obatnya satu: masukkan seluruh pra-pemrosesan ke dalam `Pipeline` yang dipasang hanya pada data latih.

k-fold cross validation



Data dibagi menjadi k bagian berukuran sama. Model dilatih k kali; setiap kali satu bagian menjadi data uji dan sisanya menjadi data latih. Hasil akhirnya:

$$\bar{\mathrm{err}} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{err}_i, \qquad s = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(\mathrm{err}_i - \bar{\mathrm{err}})^2}$$

Intinya

Keuntungannya dua: **setiap baris data pernah menjadi data uji**, dan kita memperoleh k angka sehingga bisa melihat ragamnya, bukan satu angka tunggal. Ongkosnya jelas: melatih k kali.

Varian validasi silang dan kapan masing-masing dipakai

Varian	Cara kerja	Dipakai ketika
k-fold, $k = 5$ atau 10	Bagi acak menjadi k bagian	Pilihan baku untuk data ukuran sedang; $k = 10$ bila melatihnya murah
Stratified k-fold	Proporsi tiap kelas dijaga sama di setiap lipatan	Kelasnya timpang, atau datanya kecil sehingga satu lipatan bisa kehilangan kelas minoritas — baku untuk klasifikasi
Leave-one-out ($k = n$)	Satu baris menjadi data uji, $n - 1$ sisanya melatih	Data sangat sedikit dan melatih model itu murah; nyaris tak bias, tetapi ragamnya besar dan ongkosnya n kali latih
Group k-fold	Semua baris satu kelompok masuk ke lipatan yang sama	Ada pasien, pengguna, atau lokasi yang menyumbang banyak baris
Time series split	Melatih pada masa lalu, menguji pada masa depan	Data berurut waktu; pembagian acak di sini adalah kebocoran
Nested CV	CV di dalam untuk menyetel, CV di luar untuk menilai	Anda menyetel *hyperparameter* dan tetap ingin taksiran kinerja yang jujur

Untuk tugas besar mata kuliah ini, *stratified* 5-fold di dalam dan satu data uji yang dikunci sejak awal sudah cukup.

Membandingkan dua algoritma

Paired t-test pada hasil k-fold

Karena kedua algoritma diuji pada **lipatan yang sama**, kita boleh membandingkan selisihnya lipatan demi lipatan. Inilah arti kata *paired*.

Lipatan	Pohon tunggal	*Random forest*	d_i
1	0,22	0,16	0,06
2	0,26	0,21	0,05
3	0,19	0,15	0,04
4	0,24	0,20	0,04
5	0,21	0,18	0,03
Rata-rata	0,224	0,180	**0,044**

$$t = \frac{\bar{d}}{\sqrt{\dfrac{1}{k(k-1)} \sum_{i=1}^{k} (d_i - \bar{d})^2}}$$

$\sum (d_i - \bar{d})^2 = 0{,}00052$, sehingga penyebutnya $\sqrt{0{,}00052/20} = 0{,}0051$, dan

$$t = \frac{0{,}044}{0{,}0051} \approx 8{,}6$$

Dengan $k - 1 = 4$ derajat kebebasan, nilai kritis dua sisi pada 95% adalah 2,776. Karena $8{,}6 > 2{,}776$, selisihnya **nyata**.

Yang membuat kesimpulan ini kuat bukan besarnya selisih 4,4%, melainkan bahwa *random forest* menang di **setiap** lipatan dengan jarak yang konsisten.

Arti *p-value*, dan satu peringatan

p-value secara intuitif

Andaikan kedua algoritma sebenarnya sama baik. *p-value* adalah peluang kita melihat selisih sebesar ini atau lebih besar, semata-mata karena keberuntungan pembagian data.

- $p = 0{,}001$: sulit dijelaskan sebagai kebetulan
- $p = 0{,}35$: persis seperti yang diharapkan dari kebetulan
- p **bukan** peluang bahwa model B lebih baik, dan **bukan** ukuran seberapa besar selisihnya

Lipatan	Model X	Model Y	d_i
1	0,182	0,174	0,008
2	0,201	0,200	0,001
3	0,177	0,181	−0,004
4	0,194	0,182	0,012
5	0,188	0,190	−0,002
Rata-rata	0,1884	0,1854	**0,003**

$\sum(d_i - \bar{d})^2 = 0{,}000184 \Rightarrow$ penyebut 0,00303, sehingga $t = 0{,}003/0{,}00303 \approx 0{,}99$. Jauh di bawah 2,776: **tidak ada bukti**.

Peringatan yang perlu Anda bawa seumur karier

Selisih 0,3% pada satu dataset, tanpa uji dan tanpa pengulangan, biasanya bukan bukti apa-apa: ia bisa hilang hanya dengan mengubah *random seed*. Sebelum mengatakan “model kami lebih baik”, tunjukkan bahwa selisihnya bertahan di banyak lipatan, beberapa *seed*, dan bila mungkin lebih dari satu dataset.

Teorema Bayes dan pembelajaran Bayesian

Teorema Bayes: cara memperbarui keyakinan

Teorema Bayes

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

- **Prior** $P(h)$ — seberapa besar kita mempercayai hipotesis h **sebelum** melihat data. Di sinilah pengetahuan awal masuk secara terbuka.
- **Likelihood** $P(D \mid h)$ — seberapa besar peluang data seperti ini muncul **jika** h benar. Inilah yang dihitung model.
- **Posterior** $P(h \mid D)$ — keyakinan kita atas h **setelah** melihat data. Ini hasil yang kita cari.
- **Evidence** $P(D) = \sum_{h_i \in H} P(D \mid h_i) P(h_i)$ — peluang data itu muncul menurut semua hipotesis. Ia tidak bergantung pada h, sehingga sering cukup diperlakukan sebagai penormal.

Cara membacanya sederhana: **posterior sebanding dengan likelihood dikali prior**. Data yang kuat dapat mengalahkan prior yang lemah, dan prior yang kuat menuntut data lebih banyak untuk digeser.

Sumber: Mitchell, *Machine Learning*, Bab 6.

Contoh terhitung: uji penyakit langka

Sebuah penyakit menyerang 8 dari 1.000 orang. Uji laboratoriumnya bagus: sensitivitas 98% (benar mendeteksi yang sakit) dan spesifisitas 97% (benar melewatkan yang sehat). Hasil uji Anda **positif**. Berapa peluang Anda benar-benar sakit?

Yang diketahui:

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

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

Hitung dua jalur menuju hasil positif:

$$0{,}008 \times 0{,}98 = 0{,}00784$$

$$0{,}992 \times 0{,}03 = 0{,}02976$$

$$P(+) = 0{,}00784 + 0{,}02976 = 0{,}03760$$

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

9.920 sehat (80 sakit = pita merah tipis)

Bayangkan 10.000 orang diuji

80 sakit → 78 positif benar, 2 lolos
9.920 sehat → 298 positif palsu,
9.622 negatif benar

Total positif 376; yang benar sakit 78

$78/376 = 0{,}207$ — sama dengan hitungan di sebelah kiri, hanya dinyatakan sebagai orang, bukan peluang.

Mengapa intuisi orang meleset di sini

Kebanyakan orang, termasuk banyak dokter, menjawab “sekitar 95%”. Jawabannya 21%. Dua sebab yang selalu sama:

- **Prior diabaikan.** Yang diingat hanya angka 98% dan 97%, sedangkan prevalensi 0,008 dilupakan. Kekeliruan ini punya nama: *base rate fallacy*.
- **Dua arah bersyarat ditukar.** $P(+ \mid \text{sakit}) = 0{,}98$ adalah sifat **alat ujinya**. Yang Anda butuhkan $P(\text{sakit} \mid +)$, dan keduanya tidak sama. Menukarkannya disebut *prosecutor’s fallacy*.
- **Yang sehat jauh lebih banyak.** Alarm palsu 3% dari 9.920 orang (298 orang) mengalahkan 78 deteksi benar, semata-mata karena kelompok sehat jauh lebih besar. Ini persis pelajaran yang sama dengan kasus penipuan 1:1.000 tadi.

Dan inilah sebabnya uji penapisan diulang. Bila uji kedua yang independen juga positif, posteriornya melompat:

$$\frac{0{,}008 \times 0{,}98^2}{0{,}008 \times 0{,}98^2 + 0{,}992 \times 0{,}03^2} = \frac{0{,}0076832}{0{,}0085760} \approx 0{,}896$$

Posterior dari uji pertama menjadi prior bagi uji kedua. Begitulah pembelajaran Bayesian bekerja: keyakinan diperbarui bertahap, bukan diputuskan sekali.

MAP, maximum likelihood, dan panjang kode

Hipotesis MAP dan maximum likelihood

MAP — *maximum a posteriori*

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

Penyebut $P(D)$ dibuang karena sama untuk semua h. MAP memakai prior, sehingga pengetahuan awal ikut diperhitungkan.

ML — *maximum likelihood*

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

Hanya melihat kecocokan dengan data. Inilah yang dipakai hampir semua rutin pelatihan baku.

Kapan keduanya sama

Bila priornya **seragam**, yaitu $P(h_i) = P(h_j)$ untuk semua pasangan hipotesis, maka $P(h)$ menjadi faktor tetap yang tidak mengubah urutan, sehingga $h_{MAP} = h_{ML}$. Memakai *maximum likelihood* tanpa menyadarinya berarti diam-diam menyatakan “semua hipotesis sama mungkin sebelum saya melihat data”.

Kembali ke uji penyakit tadi, dengan $H = \{\text{sakit}, \text{sehat}\}$ dan $D = \{+\}$: h_{ML} membandingkan 0,98 dengan 0,03 dan menjawab **sakit**; h_{MAP} membandingkan 0,00784 dengan 0,02976 dan menjawab **sehat**. Pada kejadian langka, mengabaikan prior bukan sekadar kurang teliti — ia mengubah kesimpulan.

Maximum likelihood dan kuadrat terkecil adalah satu hal yang sama

Andaikan data berbentuk $d_i = f(x_i) + \varepsilon_i$ dengan galat $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ yang saling bebas. Maka:

$$h_{ML} = \arg\max_h \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(d_i - h(x_i))^2}{2\sigma^2}\right) \qquad \text{likelihood Gaussian}$$

$$= \arg\max_h \sum_{i=1}^{m} \left[-\ln\sqrt{2\pi\sigma^2} - \frac{(d_i - h(x_i))^2}{2\sigma^2}\right] \qquad \text{logaritma tidak menggeser letak maksimum}$$

$$= \arg\max_h \sum_{i=1}^{m} -(d_i - h(x_i))^2 \qquad \text{buang suku dan faktor yang tetap}$$

$$= \arg\min_h \sum_{i=1}^{m} (d_i - h(x_i))^2 \qquad \text{inilah jumlah kuadrat galat}$$

Intinya

Meminimalkan jumlah kuadrat galat bukan pilihan sembarangan, melainkan akibat langsung dari satu asumsi: **galatnya Gaussian, berpusat nol, dan ragamnya sama**. Ketika asumsi itu gugur — misalnya ada *outlier* berat atau galatnya menumpuk di satu sisi — kuadrat terkecil ikut gugur, dan kita berpindah ke fungsi galat lain seperti Huber.

Minimum Description Length: belajar sama dengan memampatkan

Teori informasi Shannon mengatakan pesan berpeluang p paling ringkas dikodekan dengan $-\log_2 p$ bit. Ambil $-\log_2$ dari $P(D \mid h)P(h)$ pada rumus h_{MAP}, dan tanda maksimum berbalik menjadi minimum:

Prinsip MDL

$$h_{MDL} = \arg\min_{h \in H} \Big[\underbrace{-\log_2 P(h)}_{\text{panjang kode hipotesis}} + \underbrace{-\log_2 P(D \mid h)}_{\text{panjang kode data, diberi } h} \Big]$$

Bayangkan Anda mengirim data kepada rekan melalui saluran yang mahal. Anda boleh mengirim **modelnya** dulu, lalu hanya **daftar kesalahan** model itu. Hipotesis terbaik adalah yang membuat jumlah kedua kiriman paling pendek.

- Hipotesis yang rumit mahal di suku pertama, walaupun suku keduanya nol.
- Hipotesis terlalu sederhana murah di suku pertama, tetapi mahal di suku kedua.
- Inilah *Occam's razor* dalam satuan bit, bukan selera — sekaligus alasan formal di balik bias **"pohon pendek lebih disukai"** pada ID3 di Pertemuan 3.

Bayes optimal classifier dan jalan menuju naive Bayes

Bayes optimal classifier: jangan pilih satu hipotesis

Sejauh ini kita memilih satu hipotesis terbaik lalu memakainya. Tetapi pertanyaan sebenarnya bukan "hipotesis mana yang paling mungkin", melainkan **"label mana yang paling mungkin"**. Keduanya bisa berbeda jawaban.

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

Setiap hipotesis memberi suara, dan beratnya sebesar posteriornya.

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

Hasilnya berbeda

$h_{MAP} = h_1$, sehingga bila kita memakai satu hipotesis terbaik, jawabannya $+$.

Tetapi bobot posterior untuk label $-$ adalah $0{,}3 + 0{,}3 = 0{,}6$, sedangkan untuk $+$ hanya 0,4. *Bayes optimal classifier* menjawab $-$.

Tidak ada metode lain, dengan ruang hipotesis dan prior yang sama, yang rata-rata bisa mengalahkannya. Ia adalah batas atas kinerja.

Mengapa yang optimal itu tak terpakai, dan apa penggantinya

- **Ongkosnya tak terbayar.** Jumlahnya harus diambil atas **seluruh** ruang hipotesis. Untuk konjungsi atas 10 atribut biner saja jumlahnya sudah ribuan; untuk pohon keputusan atau jaringan saraf, ruangnya tak hingga — dan setiap satu prediksi menuntut seluruh ruang dijelajahi.
- **Priornya tak diketahui.** $P(h)$ harus ditetapkan, dan pada masalah nyata kita tidak punya dasar untuk menetapkannya.
- **Algoritma Gibbs** — alternatif murahnya: ambil satu hipotesis **secara acak menurut sebaran posterior**, lalu pakai hipotesis itu saja untuk menjawab. Hasil teoretisnya mengejutkan: galat yang diharapkan paling buruk **dua kali** galat *Bayes optimal classifier*. Gagasan yang sama muncul kembali pada *ensemble* dan *dropout*.
- **Nilainya sebagai tolok ukur.** Walaupun tidak dijalankan, ia memberi tahu batas terbaik yang mungkin, sehingga kita tahu seberapa jauh model kita dari batas itu.

Jembatan ke Pertemuan 6

Pertemuan depan kita mengambil jalan pintas yang jauh lebih praktis: berhenti menjumlah atas hipotesis, dan langsung menaksir $P(v \mid a_1, \ldots, a_n)$ dari data. Agar bisa dihitung, satu asumsi berani ditambahkan — **atribut saling bebas bila kelasnya diketahui**. Itulah *naive Bayes*: asumsinya hampir selalu salah, dan hasilnya sering sangat baik.

Rangkuman satu halaman

Bagian A — evaluasi

- Yang kita punya *sample error*; yang kita inginkan *true error*
- Galat terukur adalah variabel acak binomial, dengan simpangan baku $\sqrt{\mathrm{err}(1-\mathrm{err})/n}$
- Laporkan $\mathrm{err} \pm z_N \sqrt{\mathrm{err}(1-\mathrm{err})/n}$, bukan satu angka
- Pada data timpang, akurasi menyesatkan; pakai presisi, *recall*, F_1, atau ongkos
- Data uji disentuh sekali; pra-pemrosesan masuk `Pipeline`
- k-fold memberi rata-rata sekaligus ragam
- Selisih kecil tanpa uji bukan bukti

Bagian B — Bayesian

- $P(h \mid D) \propto P(D \mid h)P(h)$: posterior sebanding likelihood kali prior
- Mengabaikan prior pada kejadian langka mengubah kesimpulan
- $h_{MAP} = h_{ML}$ hanya bila priornya seragam
- Kuadrat terkecil adalah *maximum likelihood* dengan galat Gaussian
- MDL: $-\log_2 P(h) - \log_2 P(D \mid h)$, yaitu *Occam's razor* dalam satuan bit
- *Bayes optimal* adalah batas atas yang tak terjangkau; Gibbs paling buruk dua kali lebih jelek

Tugas untuk pertemuan berikutnya

1. **Bacaan.** Mitchell Bab 5 dan Bab 6 (khususnya 6.1–6.7). Bishop Bab 1.2 sebagai pendalaman.
2. **Protokol evaluasi tugas besar** — tulis satu halaman untuk dataset kelompok Anda:
   - pembagian latih, validasi, dan uji beserta proporsinya, dan alasan mengapa pembagiannya acak, bertingkat (*stratified*), berkelompok, atau menurut waktu;
   - skema k-fold yang dipakai pada bagian latih, beserta nilai k dan alasannya;
   - metrik utama yang akan dilaporkan, dengan alasan berbasis ongkos kesalahan, bukan sekadar "karena umum dipakai";
   - daftar langkah pra-pemrosesan dan penjelasan bagaimana Anda mencegah kebocoran pada masing-masing langkah.
3. **Latihan soal.** (a) Sebuah model salah pada 27 dari 150 contoh uji. Hitung *sample error* dan selang kepercayaan 90% serta 99%. (b) Penapisan penyakit dengan prevalensi 0,002, sensitivitas 95%, spesifisitas 90%: hitung $P(\text{sakit} \mid +)$, lalu hitung ulang bila uji kedua yang independen juga positif. (c) Tunjukkan bahwa meminimalkan *cross-entropy* pada klasifikasi biner setara dengan memaksimalkan likelihood Bernoulli.
4. **Praktikum.** Bandingkan pohon tunggal dan *random forest* pada dataset kelompok Anda memakai `cross_val_score` dengan *stratified* 10-fold, lalu jalankan `scipy.stats.ttest_rel` atas kedua deret hasilnya. Laporkan t, p, dan satu paragraf kesimpulan yang jujur — termasuk bila kesimpulannya "belum ada bukti".

Terima kasih

Pertanyaan?