Pertemuan 7 — kNN dan Regresi Linier

Menunda pekerjaan, lalu menarik garis — sekaligus rekap menjelang UTS

Hendri Karisma

Semester Ganjil 2026/2027

Program Studi Teknik Informatika, STMIK Tazkia

Alur pertemuan hari ini

1. Dari *eager* ke *lazy*: belajar berbasis contoh
2. Algoritma k-nearest neighbour
3. Metrik jarak, dan mengapa penskalaan wajib
4. Pengaruh k: bias dan variance
5. Pembobotan jarak dan regresi terbobot lokal
6. Kutukan dimensi dan ongkos menjawab
7. Dari klasifikasi ke regresi
8. Galat kuadrat dan bentuk tertutup
9. Asumsi, diagnostik, dan R^2
10. Batas bentuk tertutup, jembatan ke Pertemuan 9
11. Rekapitulasi model Pertemuan 2–7
12. Tugas besar UTS

Sasaran pertemuan ini

Anda dapat menjalankan kNN dan regresi linier dengan tangan pada data kecil, menjelaskan asumsi dan ongkos masing-masing, lalu memilih di antara seluruh model yang sudah kita pelajari dengan alasan yang dapat dipertanggungjawabkan.

Dari eager ke lazy: pembelajaran berbasis contoh

Menyambung dari Pertemuan 6

Pada Pertemuan 6 kita menutup dua model yang bekerja keras sebelum ditanya. Naive Bayes menghitung seluruh peluang kelas dan peluang bersyarat setiap fitur, lalu menyimpannya sebagai tabel. SVM menyelesaikan satu persoalan optimasi berkendala untuk menemukan bidang pemisah bermargin terbesar, lalu cukup menyimpan *support vector*-nya.

Keduanya berbagi satu pola yang sama, dan pola itu berlaku juga untuk pohon keputusan dan ensembel: ada satu **tahap latih** yang memadatkan data menjadi sebuah model, dan setelah itu data latihnya boleh dilupakan.

Pertanyaan hari ini

Bagaimana jika kita menolak memadatkan apa pun? Kita simpan saja seluruh data, dan baru berpikir ketika ada pertanyaan yang masuk. Itulah gagasan *instance-based learning*, dan wakilnya yang paling terkenal adalah kNN.

Sumber: Tom M. Mitchell, *Machine Learning*, Bab 8: Instance-Based Learning.

Eager learner dan lazy learner

Eager — rajin di muka

Membangun satu model global saat latih, lalu membuang datanya.

- Pohon keputusan, Naive Bayes, SVM, regresi linier
- Satu hipotesis untuk seluruh ruang masukan
- Latih mahal, menjawab murah
- Keputusannya dapat dibaca tanpa melihat data lagi

Lazy — menunda sampai ditanya

Tidak ada model yang dibangun. Data disimpan apa adanya.

- kNN, *locally weighted regression*, *case-based reasoning*
- Hipotesisnya dibentuk **lokal**, berbeda untuk setiap pertanyaan
- Latih hampir gratis, menjawab mahal
- Alasannya berupa “mirip dengan contoh-contoh ini”

Perhatikan keuntungan tersembunyi dari menunda: karena hipotesis dibentuk per pertanyaan, kNN tidak perlu satu bentuk fungsi yang cocok untuk seluruh ruang. Batas keputusan yang berliku dilayaninya tanpa usaha tambahan, sedangkan model global harus menambah fitur atau kernel dulu.

k-nearest neighbour: jarak, skala, dan k

Algoritmanya sesingkat itu

Tahap latih. Simpan seluruh pasangan $(\mathbf{x}_i, y_i)$. Selesai.

Tahap menjawab untuk satu pertanyaan $\mathbf{x}_q$:

1. Hitung jarak $d(\mathbf{x}_q, \mathbf{x}_i)$ ke **seluruh** data latih.
2. Ambil k data dengan jarak terkecil. Sebut himpunannya $N_k(\mathbf{x}_q)$.
3. Gabungkan label mereka menjadi satu jawaban.

Klasifikasi — suara terbanyak

$$\hat{y} = \arg\max_{c} \sum_{i \in N_k} \mathbb{1}[y_i = c]$$

Proporsi suara juga dapat dibaca sebagai taksiran peluang kelas.

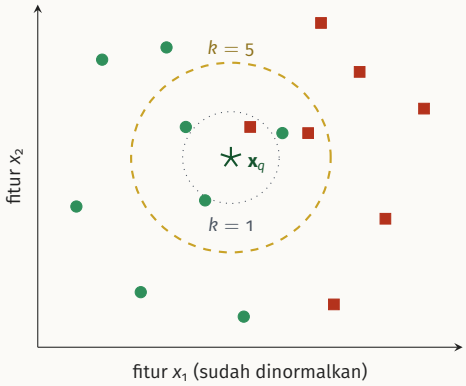
Regresi — rata-rata

$$\hat{y} = \frac{1}{k} \sum_{i \in N_k} y_i$$

Keluaran kontinu diperoleh dengan merata-ratakan nilai tetangga, bukan memilih mayoritas.

Tidak ada parameter yang dilatih. Yang kita tentukan hanyalah k, metrik jarak, dan cara pembobotan — ketiganya **hyperparameter**, ditetapkan lewat validasi, bukan lewat data latih.

Satu pertanyaan, dua jawaban, tergantung k



Bulatan hijau dan kotak merah adalah dua kelas. Bintang di tengah adalah pertanyaan $\mathbf{x}_q$ yang belum berlabel. Lima tetangga terdekatnya:

Tetangga	Jarak	Kelas
1	0,58	merah
2	0,81	hijau
3	0,86	hijau
4	0,89	hijau
5	1,27	merah

Dengan $k = 1$ jawabannya **merah**, karena tetangga terdekat kebetulan satu titik merah yang menyusup. Dengan $k = 5$ suaranya 3 hijau lawan 2 merah, sehingga jawabannya **hijau**.

Satu titik dapat membalikkan keputusan. Itulah alasan k perlu dipilih, bukan ditebak.

Ukuran “mirip” harus dipilih lebih dahulu

Metrik	Rumus	Dipakai ketika
Euclidean (L_2)	$\sqrt{\sum_j (x_j - z_j)^2}$	Baku untuk fitur numerik yang sebanding; satu selisih besar dihukum lebih keras daripada banyak selisih kecil
Manhattan (L_1)	$\sum_j \lvert x_j - z_j \rvert$	Lebih tahan *outlier*, dan lebih masuk akal pada data berdimensi tinggi atau jarang
Minkowski (L_p)	$\left(\sum_j \lvert x_j - z_j \rvert^p\right)^{1/p}$	Bentuk umum: $p = 1$ Manhattan, $p = 2$ Euclidean, $p \to \infty$ hanya selisih terbesar
Cosine	$1 - \dfrac{\mathbf{x} \cdot \mathbf{z}}{\lVert\mathbf{x}\rVert \lVert\mathbf{z}\rVert}$	Ketika **arah** lebih berarti daripada besaran: teks TF-IDF, *embedding*
Hamming	$\sum_j \mathbb{1}[x_j \neq z_j]$	Data kategorikal atau biner: cukup dihitung berapa atribut yang berbeda

Lewat *one-hot*, jarak Euclidean kuadrat persis dua kali jarak Hamming — tetapi atribut dengan 50 kategori lalu menyumbang jauh lebih besar daripada atribut dengan 2 kategori. Untuk data campuran numerik dan kategorikal ada jarak Gower, yang menormalkan ketidaksamaan setiap atribut ke $[0, 1]$ lalu merata-ratakannya. Pemilihan metrik adalah tempat pengetahuan domain masuk ke dalam kNN: karena tidak ada tahap latih, metrik inilah satu-satunya cara kita memberi tahu algoritma apa arti “mirip”.

Mengapa penskalaan fitur wajib: satuan menentukan jawaban

Tiga baris data pemohon kredit dengan dua fitur, gaji bulanan dalam rupiah dan umur dalam tahun.

	Gaji (Rp)	Umur
P_1 (tolak)	8.000.000	25
P_2 (terima)	9.500.000	45
$\mathbf{x}_q$ (?)	8.200.000	44

Jarak Euclidean kuadrat pada satuan aslinya:

$$d(\mathbf{x}_q, P_1)^2 = (200.000)^2 + 19^2 = 4{,}0 \times 10^{10} + 361$$

$$d(\mathbf{x}_q, P_2)^2 = (1.300.000)^2 + 1^2 = 1{,}69 \times 10^{12} + 1$$

Jadi $d(\mathbf{x}_q, P_1) \approx 200.000$ sedangkan $d(\mathbf{x}_q, P_2) \approx 1.300.000$. Tetangga terdekat adalah P_1, sehingga dengan $k = 1$ jawabannya **tolak**.

Intinya

Lihat angka 361 dan 1 itu. Sumbangan umur sekitar seperseratus miliar dari total, sehingga umur secara efektif **tidak ikut dihitung**. Yang kita jalankan sebenarnya hanyalah kNN atas gaji.

Setelah dinormalkan, jawabannya berbalik

Skalakan dengan min–max memakai rentang yang masuk akal: gaji 5–20 juta (rentang 15 juta) dan umur 22–60 tahun (rentang 38).

	gaji′	umur′
P_1	0,200	0,079
P_2	0,300	0,605
$\mathbf{x}_q$	0,213	0,579

Contoh satu sel: gaji′ untuk $\mathbf{x}_q$ adalah $(8{,}2 - 5)/15 = 3{,}2/15 = 0{,}213$.

$$d(\mathbf{x}_q, P_1) = \sqrt{0{,}013^2 + 0{,}500^2} = 0{,}500$$
$$d(\mathbf{x}_q, P_2) = \sqrt{0{,}087^2 + 0{,}026^2} = 0{,}091$$

Sekarang P_2 lebih dekat, lima kali lebih dekat, dan jawabannya menjadi **terima**.

Data yang sama, metrik yang sama, k yang sama — keputusan berlawanan, hanya karena satuannya. Karena itu **penskalaan** bukan pemanis, melainkan bagian dari definisi model. Pilihan yang lazim: min–max ke $[0, 1]$, atau standardisasi $z = (x - \mu)/\sigma$ yang lebih tahan nilai ekstrem. Hitung μ dan σ **hanya dari data latih**, lalu terapkan angka itu ke data uji. Bila tidak, protokol evaluasi Pertemuan 5 sudah bocor sejak awal.

Pengaruh k: dua kesalahan yang berlawanan

k kecil — variance tinggi

Keputusan bergantung pada sedikit titik, sehingga satu titik salah label atau satu pencilan dapat membaliknya. Batas keputusannya berliku dan penuh pulau kecil. Pada $k = 1$ galat pada data latih selalu 0% — setiap titik adalah tetangga terdekat dirinya sendiri — dan angka itu tidak berarti apa pun.

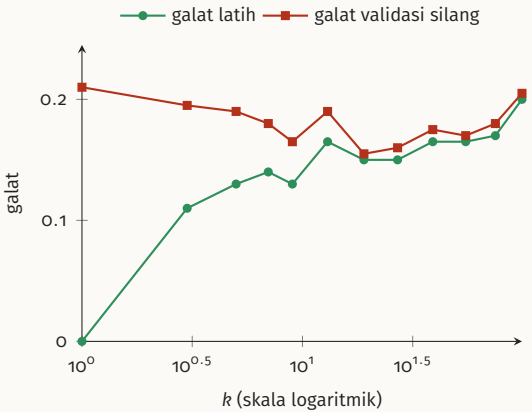
k besar — bias tinggi

Suara diambil dari wilayah yang makin luas, sehingga batasnya makin halus dan struktur lokal yang halus tersapu. Pada ekstremnya, $k = n$ membuat setiap pertanyaan dijawab dengan kelas mayoritas seluruh data: model tidak lagi melihat $\mathbf{x}_q$ sama sekali.

Bias induktif kNN

Asumsinya satu kalimat: **titik yang berdekatan cenderung berlabel sama**. Ini asumsi kemulusan lokal. Bila asumsi itu tidak berlaku — misalnya label ditentukan oleh XOR dua fitur yang tenggelam di antara puluhan fitur lain — kNN akan gagal, dan bukan karena k-nya salah.

Galat sebagai fungsi k, dan cara memilihnya



Data sintetis dua bulan sabit bertumpang ($n = 200$, derau tinggi), validasi silang 10-lipat.

Bacalah dua kurva itu bersama-sama:

- Pada $k = 1$ galat latih 0% tetapi galat validasi justru **tertinggi**, 21%. Inilah wajah *overfitting* pada kNN.
- Galat validasi menurun sampai sekitar $k \approx 9$–19, lalu naik lagi ketika k terlalu besar.
- Kurvanya bergelombang, tidak mulus. Validasi silang sendiri punya derau, sehingga selisih satu persen antara dua k berdekatan jangan diperlakukan sebagai kemenangan.

Cara memilih k: coba deretan nilai, ukur dengan validasi silang pada data latih, ambil yang terbaik, lalu laporkan kinerjanya pada data uji yang belum pernah dipakai.

Pakai k ganjil untuk dua kelas, supaya suara tidak seri. Untuk c kelas, hindari k yang merupakan kelipatan c.

Pembobotan jarak: tetangga dekat bersuara lebih keras

Pada kNN biasa, tetangga di jarak 0,1 dan di jarak 1,2 punya suara yang sama besar. Perbaikannya sederhana: beri bobot $w_i = 1/d(\mathbf{x}_q, \mathbf{x}_i)^2$, lalu pakai suara atau rata-rata terbobot.

$$\hat{y}_{\text{klasifikasi}} = \arg\max_c \sum_{i \in N_k} w_i \, \mathbb{1}[y_i = c], \qquad \hat{y}_{\text{regresi}} = \frac{\sum_{i \in N_k} w_i \, y_i}{\sum_{i \in N_k} w_i}$$

Contoh terhitung, $k = 3$**.**

Tetangga	d_i	$w_i = 1/d_i^2$	kelas
1	0,5	4,000	A
2	1,0	1,000	B
3	1,2	0,694	B

Suara biasa: B menang 2 lawan 1. Suara terbobot: A memperoleh 4,000, B memperoleh $1{,}000 + 0{,}694 = 1{,}694$. **A menang.**

Versi regresi dengan tetangga bernilai $y = 10,\ 12,\ 14$ pada jarak yang sama seperti di kiri:

$$\hat{y} = \frac{4(10) + 1(12) + 0{,}694(14)}{4 + 1 + 0{,}694} = \frac{61{,}72}{5{,}694} = 10{,}84$$

Rata-rata biasa memberi 12,00; pembobotan menariknya mendekati tetangga terdekat.

Efek sampingnya menyenangkan: dengan pembobotan, k yang terlalu besar jadi kurang merusak, karena tetangga jauh otomatis nyaris tak bersuara.

Dari rata-rata terbobot ke regresi terbobot lokal

kNN untuk regresi menjawab dengan satu bilangan datar, yaitu rata-rata tetangga. Akibatnya kurva jawabannya bertangga: ia melompat setiap kali himpunan tetangga berubah.

Locally weighted regression memperbaikinya dengan satu langkah: alih-alih merata-ratakan tetangga, **cocokkan sebuah garis** pada tetangga itu, dengan bobot yang mengecil bersama jarak.

$$\min_{\beta} \sum_{i \in N_k} w_i \left(y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_d x_{id}\right)^2, \qquad w_i = K\left(\frac{d(\mathbf{x}_q, \mathbf{x}_i)}{h}\right)$$

Garis itu dipakai sekali saja untuk menjawab $\mathbf{x}_q$, lalu dibuang; pertanyaan berikutnya mendapat garisnya sendiri. Fungsi K adalah *kernel* pembobot, dan h lebar jendela.

Intinya

Inilah pertemuan dua bagian kuliah hari ini: regresi linier yang **dipasang secara lokal** oleh sebuah metode lazy. Model globalnya tidak ada.

Kutukan dimensi dan ongkos menjawab

Pada dimensi tinggi, semua titik nyaris sama jauh

Ilustrasi 1 — volume menempel di kulit. Pada kubus satuan d dimensi, bagian volume di dalam lapisan setebal 0,05 dari permukaan adalah $1 - 0{,}9^d$:

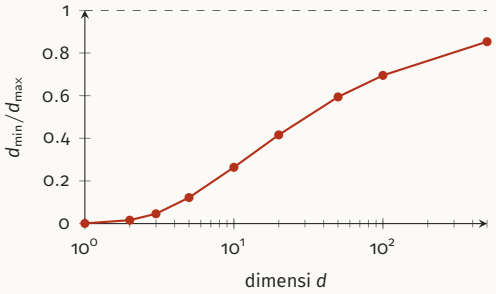
d	1	2	10	50	100
di kulit	10%	19%	65%	99,5%	$\approx$ 100%

Pada $d = 100$ hampir tidak ada "bagian dalam". Setiap titik adalah titik tepi, dan tepi berarti jauh dari titik lain.

Ilustrasi 2 — lingkungan lokal tidak lagi lokal. Untuk memuat 10% data, kubus kecil harus bersisi $0{,}1^{1/d}$ dari rentang setiap fitur:

d	1	2	10	50	100
sisi	0,10	0,32	0,79	0,96	0,98

Pada $d = 10$, "sepuluh persen tetangga terdekat" sudah merentang 79% rentang setiap fitur, sehingga kata "terdekat" kehilangan artinya.



Ilustrasi 3. Seribu titik acak seragam pada $[0, 1]^d$ dan satu titik pertanyaan; yang digambar adalah rasio jarak terdekat terhadap jarak terjauh, dirata-ratakan atas 200 percobaan. Pada $d = 2$ rasionya 0,016; pada $d = 100$ sudah 0,70, dan terus menuju 1. Ketika rasio itu mendekati 1, "tetangga terdekat" dan "titik terjauh" hampir berjarak sama, dan seluruh gagasan kNN ikut runtuh bersamanya.

Menghadapi kutukan dimensi

- **Buang fitur yang tidak relevan.** Pada kNN, fitur tak relevan bukan netral; ia menambah jarak secara acak dan menenggelamkan fitur yang berguna. Pemilihan fitur memberi perbaikan yang lebih besar di kNN daripada di pohon keputusan, yang memang mengabaikan fitur tak berguna dengan sendirinya.
- **Reduksi dimensi.** PCA atau *embedding* memadatkan puluhan fitur menjadi beberapa arah yang benar-benar membawa ragam.
- **Bobot per fitur.** Beri setiap fitur bobot α_j di dalam jarak, lalu setel α dengan validasi silang. Inilah cikal bakal *metric learning*.

Kaitan dengan Pertemuan 2

Kutukan dimensi adalah bentuk lain dari pelajaran ruang hipotesis: makin banyak fitur, makin longgar model, dan makin banyak data yang dibutuhkan untuk mengikatnya. Kebutuhan data tumbuh secara eksponensial terhadap dimensi, sedangkan data kita tidak.

Catatan penting: kNN tetap sangat kuat ketika dimensinya sedang, datanya cukup banyak, dan fiturnya sudah dipilih dengan baik. Pada data tabular kecil, kNN yang dinormalkan dengan benar sering mengalahkan model yang jauh lebih rumit.

Latih hampir gratis, menjawab mahal

	Latih	Satu pertanyaan
Waktu	$O(1)$	$O(nd)$
Memori	$O(nd)$	$O(nd)$ tetap dipegang

Menemukan k terkecil setelah jaraknya dihitung hanya menambah $O(n \log k)$ dengan *heap*, jadi biaya sebenarnya didominasi penghitungan n jarak.

Dengan $n = 1.000.000$ dan $d = 100$, satu pertanyaan memerlukan sekitar 10^8 operasi dasar. Kalikan dengan seribu permintaan per detik, dan sistemnya tidak mungkin dilayani apa adanya.

Kembali ke Pertemuan 1

Di pertemuan pertama kita membedakan **efektivitas** dan **efisiensi**, dan menyebut pola umum ML: latih sekali mahal, menjawab berkali-kali murah. kNN adalah pengecualian yang paling terang.

Ia sering sangat **efektif** — tanpa asumsi bentuk fungsi, tanpa penyetelan yang panjang — tetapi **tidak efisien saat menjawab**, dan seluruh data latih harus ikut dibawa ke produksi. Pada sistem *real-time* dengan anggaran milidetik, itu sering menjadi alasan penolakan yang sama sekali tidak berhubungan dengan akurasinya.

Mempercepat pencarian tetangga

- **KD-tree** — membelah ruang bergantian menurut sumbu fitur, sehingga sebagian besar cabang dapat dilewati. Sangat membantu pada dimensi rendah, kira-kira sampai $d \lesssim 20$. Di atas itu kinerjanya merosot kembali ke pencarian menyeluruh, dan alasannya persis kutukan dimensi tadi.
- **Ball-tree** — membelah ruang dengan bola bersarang, bukan dengan bidang sejajar sumbu. Lebih tahan pada dimensi menengah dan menerima metrik selain Euclidean.
- **Approximate nearest neighbour** — melepaskan jaminan ketepatan untuk menukarnya dengan kecepatan. HNSW membangun graf berlapis yang dapat ditelusuri dalam waktu kira-kira logaritmik; LSH memakai fungsi *hash* yang sengaja menabrakkan titik-titik yang berdekatan. Inilah mesin di balik basis data vektor dan pencarian *embedding* pada sistem RAG hari ini.
- **Kurangi datanya.** Buang contoh yang tidak pernah mengubah keputusan, atau ganti kelompok titik yang berdekatan dengan satu wakil (prototipe). Datanya menyusut, dan kNN ikut menjadi murah.

Perhatikan bahwa gagasan lama ini justru sedang naik lagi. Pencarian tetangga terdekat pada *embedding* adalah kNN, dan ia kini dijalankan pada miliaran vektor.

Regresi linier

Dari klasifikasi ke regresi

Sampai hari ini keluaran yang kita cari selalu berupa label dari himpunan berhingga: spam atau bukan, layak atau tidak, satu dari tiga jenis bunga. Sekarang keluarannya **bilangan kontinu**: harga rumah, lama pengiriman, emisi bulan depan, nilai ujian.

Yang berubah

- Ruang keluaran: $y \in \mathbb{R}$, bukan $y \in \{c_1, \ldots, c_m\}$
- Ukuran galat: bukan lagi benar atau salah, melainkan **seberapa jauh**
- Metrik evaluasi: MSE, RMSE, MAE, R^2, bukan akurasi dan F1

Yang tidak berubah

- Pembagian data latih, validasi, dan uji tetap wajib
- *Overfitting* tetap mengintai
- Bias induktif tetap ada — di sini berupa asumsi bahwa hubungannya linier

Regresi linier adalah model tertua dalam daftar kita, dan tetap model pertama yang sebaiknya dicoba pada masalah kontinu. Bukan karena paling akurat, tetapi karena ia memberi patokan yang jujur dan koefisien yang dapat dibaca.

Modelnya: satu bidang datar di ruang fitur

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_d x_d$$

Dengan kolom konstanta 1 pada setiap baris, seluruhnya menjadi satu perkalian matriks:

$$\hat{\mathbf{y}} = X\beta, \qquad X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{bmatrix} \in \mathbb{R}^{n \times (d+1)}, \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix}$$

Kata “linier” mengacu pada **parameter**, bukan pada bentuk kurvanya. Kita boleh memasukkan x^2, $\log x$, atau $x_1 x_2$ sebagai kolom baru, dan modelnya tetap linier dalam β sehingga seluruh teori berikut tetap berlaku. Yang tidak boleh adalah parameter yang masuk secara tak linier, misalnya $e^{\beta_1 x}$.

Intinya

Melatih regresi linier berarti mencari satu vektor β sepanjang $d + 1$ angka. Tidak lebih.

Mengapa galat *kuadrat*, bukan galat mutlak

Ukuran kesalahan yang kita pakai adalah jumlah kuadrat residu:

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \|\mathbf{y} - X\beta\|^2, \qquad \mathrm{MSE} = \frac{\mathrm{RSS}}{n}$$

Alasan praktis. Fungsinya mulus dan cembung dalam β, sehingga turunannya ada di mana-mana dan minimumnya tunggal. Galat mutlak $|y - \hat{y}|$ tidak dapat diturunkan di nol dan tidak punya bentuk tertutup.

Alasan prinsipil. Kuadrat muncul sendiri dari *maximum likelihood* bila galatnya Gaussian.

Ambil model galat dari Pertemuan 5: $y_i = \mathbf{x}_i^\top \beta + \varepsilon_i$ dengan $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ saling bebas. Maka

$$\log L(\beta) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_i (y_i - \mathbf{x}_i^\top \beta)^2$$

Suku pertama tidak memuat β, sehingga memaksimumkan $\log L$ **sama saja** dengan meminimumkan RSS.

Intinya

Jadi *least squares* bukan pilihan selera; ia penaksir kemungkinan maksimum di bawah asumsi galat Gaussian. Konsekuensinya jujur: bila galatnya berekor tebal atau penuh pencilan, asumsi itu salah, dan galat mutlak atau Huber justru lebih tepat.

Menurunkan bentuk tertutup

Kembangkan RSS, lalu turunkan terhadap β:

$$\mathrm{RSS}(\beta) = (\mathbf{y} - X\beta)^\top (\mathbf{y} - X\beta) = \mathbf{y}^\top \mathbf{y} - 2\beta^\top X^\top \mathbf{y} + \beta^\top X^\top X\beta$$

$$\frac{\partial \,\mathrm{RSS}}{\partial \beta} = -2X^\top \mathbf{y} + 2X^\top X\beta$$

Samakan dengan nol. Hasilnya disebut **persamaan normal**:

$$X^\top X\beta = X^\top \mathbf{y} \qquad \Longrightarrow \qquad \boxed{\hat{\beta} = (X^\top X)^{-1} X^\top \mathbf{y}}$$

Karena RSS cembung — $X^\top X$ selalu semi-definit positif — titik berturunan nol itu pasti minimum, bukan maksimum atau titik sadel. Nama “normal” berasal dari maknanya secara geometris: vektor residu $\mathbf{y} - X\hat{\beta}$ tegak lurus terhadap seluruh kolom X. Dengan kata lain $\hat{\mathbf{y}}$ adalah proyeksi ortogonal $\mathbf{y}$ ke ruang yang direntang kolom-kolom X — titik terdekat yang masih dapat dijangkau oleh model.

Contoh terhitung: lima titik data

Kita mencatat jam belajar mandiri per minggu (x) dan nilai kuis dalam skala 10 (y) untuk lima mahasiswa.

x_i	y_i	$x_i - \bar{x}$	$y_i - \bar{y}$
1	2	-2	-2
2	4	-1	0
3	5	0	1
4	4	1	0
5	5	2	1
$\bar{x} = 3,$	$\bar{y} = 4$		

Garis selalu melewati $(\bar{x}, \bar{y}) = (3, 4)$ — itu akibat langsung dari rumus $\hat{\beta}_0$.

Untuk satu variabel, persamaan normal menyusut menjadi dua rumus:

$$S_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y}) = 4 + 0 + 0 + 0 + 2 = 6, \quad S_{xx} = \sum (x_i - \bar{x})^2 = 10$$

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = 0{,}6, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = 2{,}2 \;\Rightarrow\; \hat{y} = 2{,}2 + 0{,}6\, x$$

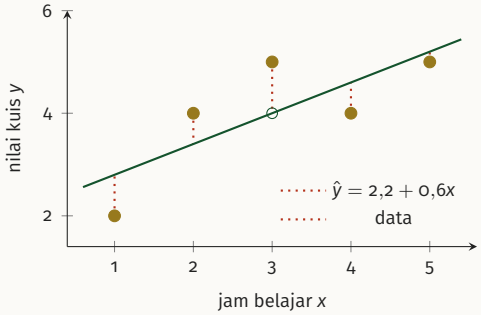
Periksa lewat bentuk matriks. Dengan $\sum x_i = 15$, $\sum x_i^2 = 55$, $\sum y_i = 20$, $\sum x_i y_i = 66$:

$$X^\top X = \begin{bmatrix} 5 & 15 \\ 15 & 55 \end{bmatrix}, \quad X^\top \mathbf{y} = \begin{bmatrix} 20 \\ 66 \end{bmatrix}, \quad \det = 275 - 225 = 50$$

$$\hat{\beta} = \frac{1}{50} \begin{bmatrix} 55 & -15 \\ -15 & 5 \end{bmatrix} \begin{bmatrix} 20 \\ 66 \end{bmatrix} = \frac{1}{50} \begin{bmatrix} 110 \\ 30 \end{bmatrix} = \begin{bmatrix} 2{,}2 \\ 0{,}6 \end{bmatrix}$$

Sama persis. Kelebihan bentuk matriks: rumusnya tidak berubah ketika fiturnya menjadi sepuluh atau seratus — yang berubah hanya ukuran $X^\top X$, dan di situlah pula masalahnya bersembunyi.

Garisnya, residunya, dan R^2



x_i	y_i	$\hat{y}_i$	$e_i = y_i - \hat{y}_i$
1	2	2,8	$-0{,}8$
2	4	3,4	$+0{,}6$
3	5	4,0	$+1{,}0$
4	4	4,6	$-0{,}6$
5	5	5,2	$-0{,}2$
		jumlah	0,0

$\mathrm{RSS} = 0{,}64 + 0{,}36 + 1{,}00 + 0{,}36 + 0{,}04 = 2{,}40$,
sehingga $\mathrm{MSE} = 0{,}48$ dan $\mathrm{RMSE} = 0{,}69$.
$\mathrm{TSS} = \sum(y_i - \bar{y})^2 = 6{,}00$.

$$R^2 = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{2{,}40}{6{,}00} = 0{,}60$$

$$R^2_{\text{adj}} = 1 - (1 - R^2)\tfrac{n-1}{n-p-1} = 1 - 0{,}4 \cdot \tfrac{4}{3} = 0{,}47$$

Jumlah residu selalu nol bila model punya konstanta β_0.

Membaca R^2 dengan hati-hati

R^2 menjawab satu pertanyaan saja: berapa bagian ragam y yang berhasil dijelaskan model, dibandingkan dengan sekadar menebak $\bar{y}$. Nilai 0,60 pada contoh kita berarti 60% ragam nilai kuis terjelaskan oleh jam belajar.

- Menambah fitur apa pun — bahkan kolom bilangan acak — **tidak pernah** menurunkan R^2. Karena itu R^2 tidak boleh dipakai membandingkan model dengan jumlah fitur berbeda.
- R^2_{adj} menghukum penambahan fitur dengan faktor $(n-1)/(n-p-1)$, dan dapat turun. Di contoh kita ia jatuh dari 0,60 ke 0,47 karena n hanya 5.
- R^2 tinggi tidak menjamin modelnya benar, dan R^2 rendah tidak selalu berarti buruk — pada data perilaku manusia, 0,3 sudah bagus.

Kembali ke Pertemuan 5

Semua angka di kiri dihitung pada data latih. Angka yang menentukan tetap galat pada data uji. R^2 pada data latih dapat dibuat mendekati 1 dengan menambah fitur sampai $p \to n$; yang seperti itu bukan model, melainkan hafalan.

Laporkan RMSE atau MAE bersama R^2. RMSE punya satuan yang sama dengan y, sehingga dapat dinilai secara praktis: “salah rata-rata 0,69 poin pada skala 10” jauh lebih mudah ditimbang daripada “$R^2 = 0{,}60$”.

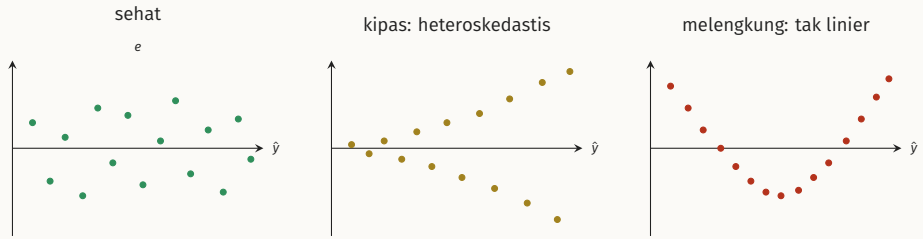
Asumsi yang dipikul regresi linier

- **Linearitas** — hubungan x dan y memang kira-kira berupa garis. Bila sebenarnya melengkung, koefisiennya menyesatkan walaupun R^2-nya lumayan.
- **Galat saling bebas** — satu residu tidak boleh dapat diramal dari residu sebelumnya. Sering dilanggar pada data runtun waktu, dan akibatnya galat baku terlalu kecil sehingga kita terlalu percaya diri.
- **Homoskedastisitas** — ragam galat tetap di seluruh rentang $\hat{y}$. Bila menyebar seperti kipas, taksiran β masih tak berbias tetapi uji dan selang kepercayaannya tidak lagi sah.
- **Normalitas galat** — diperlukan untuk uji t, uji F, dan selang kepercayaan; **tidak** diperlukan agar $\hat{\beta}$ itu sendiri baik. Pada n besar syarat ini paling ringan di antara semuanya.
- **Multikolinearitas rendah** — antar-fitur tidak boleh saling meramal. Bila luas bangunan dalam m² dan dalam kaki² ikut keduanya, $X^\top X$ nyaris singular: koefisiennya melambung, tandanya bisa terbalik, dan berubah drastis hanya karena satu data ditambahkan. Deteksi dengan *variance inflation factor* $\mathrm{VIF}_j = 1/(1 - R_j^2)$; nilai di atas 5 atau 10 pantas dicurigai.

Daftar ini adalah bentuk konkret dari **bias induktif** regresi linier. Model apa pun punya asumsi; keunggulan regresi linier adalah asumsinya dapat dituliskan dan diperiksa.

Diagnostik: bacalah plot residu, jangan hanya angkanya

Gambarkan residu e_i terhadap nilai dugaan $\hat{y}_i$. Yang diharapkan adalah awan tak berpola di sekitar nol. Tiga pola berikut adalah yang paling sering muncul.



Kipas menyarankan mentransformasi y, misalnya $\log y$, atau memakai galat baku yang tangguh. Lengkungan menyarankan menambah suku x^2 atau mengganti model. Selain itu periksa plot Q–Q untuk normalitas, dan jarak Cook untuk titik yang seorang diri menarik garisnya.

Menafsirkan koefisien, dan satu peringatan

Pada contoh kita $\hat{\beta}_1 = 0{,}6$ dibaca begini: setiap tambahan satu jam belajar per minggu **disertai** kenaikan nilai kuis rata-rata 0,6 poin, **dengan fitur lain ditahan tetap**. Tiga catatan menyertainya.

- Frasa "fitur lain ditahan tetap" menjadi tidak bermakna bila fiturnya berkorelasi kuat. Anda tidak dapat menaikkan luas bangunan tanpa menaikkan jumlah kamar.
- Besar koefisien bergantung satuan. Jangan membandingkan β antar-fitur kecuali semuanya sudah distandarkan.
- $\hat{\beta}_0$ adalah dugaan y ketika semua fitur nol. Sering tidak punya arti fisik — rumah berluas nol tidak ada — dan tetap dibutuhkan agar garisnya bebas bergeser.

Korelasi bukan kausalitas

Regresi mengukur **hubungan** dalam data, bukan **sebab**. Koefisien 0,6 itu sama sekali tidak membuktikan bahwa memaksa mahasiswa belajar satu jam lebih banyak akan menaikkan nilainya 0,6 poin. Bisa jadi mahasiswa yang termotivasi belajar lebih lama *dan* mengerjakan kuis lebih baik — motivasi adalah peubah tersembunyi yang menyebabkan keduanya. Klaim sebab-akibat memerlukan eksperimen teracak atau rancangan kausal, bukan sekadar $\hat{\beta}$ dan nilai-p yang kecil.

Batas bentuk tertutup, dan jembatan ke Pertemuan 9

Ongkosnya tumbuh cepat. Menyusun $X^\top X$ memerlukan $O(nd^2)$ operasi, dan membalikkannya $O(d^3)$.

- $d = 100$: ringan sekali.
- $d = 10.000$: sekitar 10^{12} operasi hanya untuk pembalikan, dan matriks $10^4 \times 10^4$ harus muat di memori.
- Data yang tidak muat di memori tidak bisa disusun menjadi satu X sama sekali.

$X^\top X$ **bisa tidak punya invers.** Itu terjadi ketika ada fitur yang merupakan kombinasi linier fitur lain, atau ketika $d > n$ — dan pada data teks atau genomik, $d > n$ adalah keadaan biasa.

Dua jalan keluar.

- **Regularisasi.** Ridge menambah $\lambda\|\beta\|^2$, sehingga $\hat{\beta} = (X^\top X + \lambda I)^{-1} X^\top \mathbf{y}$ — penambahan λI membuat matriksnya selalu dapat dibalik. Lasso memakai $\lambda\|\beta\|_1$ dan memaksa sebagian koefisien menjadi tepat nol, sehingga sekaligus memilih fitur. Keduanya dibahas lebih dalam setelah UTS.
- **Metode numerik.** Jangan cari rumusnya; turuni saja permukaan galatnya: $\beta_{t+1} = \beta_t - \eta \nabla \mathrm{RSS}(\beta_t)$.

Pertemuan 9

Regresi linier adalah satu dari sedikit model yang punya bentuk tertutup, dan bahkan ia pun menyerah pada data besar. Di situlah kita kembali ke pembedaan analitik dan numerik dari Pertemuan 1, lalu masuk ke *loss function*, *gradient descent*, dan SGD — mesin yang menggerakkan seluruh separuh kedua mata kuliah ini.

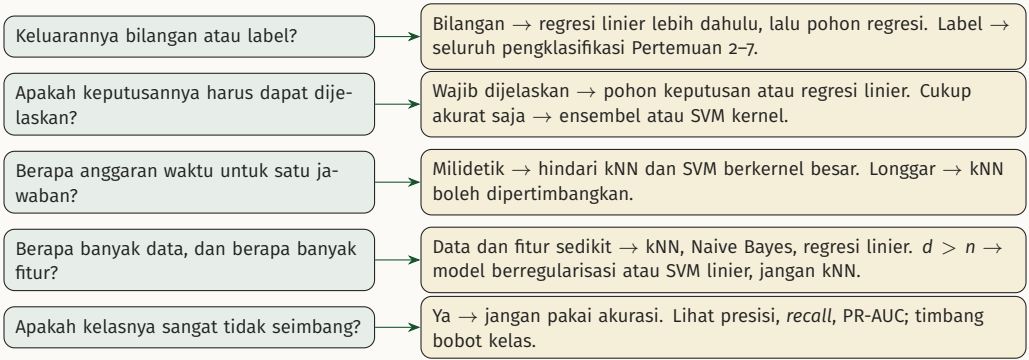
Rekapitulasi menjelang UTS

Semua model Pertemuan 2–7 dalam satu tabel

Model	Masalah	Asumsi / bias induktif	Biaya latih	Biaya inferensi	Keterbacaan
Find-S, *version space* (P2)	klasifikasi biner	Konsep target ada di ruang hipotesis konjungtif; data bebas derau	$O(nd)$	$O(d)$	sangat tinggi: satu aturan
Pohon keputusan ID3, C4.5 (P3)	keduanya	Pohon pendek lebih disukai; batas sejajar sumbu	$O(dn \log n)$	O(kedalaman)	tinggi: terbaca sebagai aturan
Random forest (P4)	keduanya	Banyak pohon beragam saling meniadakan galat	$B\times$ satu pohon	$O(B \cdot$ kedalaman$)$	rendah; ada *feature importance*
XGBoost (P4)	keduanya	Galat sisa dapat dikurangi bertahap oleh pohon kecil	lebih berat, banyak *hyperparameter*	$O(B \cdot$ kedalaman$)$	rendah; SHAP membantu
Naive Bayes (P6)	klasifikasi	Fitur saling bebas bila kelas diketahui; bentuk sebaran diasumsikan	$O(nd)$	$O(dc)$	menengah: peluang terlihat
SVM (P6)	keduanya	Margin terbesar menggeneralisasi terbaik; kernel menentukan bentuk batas	$O(n^2)$–$O(n^3)$	$O(\lvert SV\rvert \cdot d)$	rendah bila memakai kernel
kNN (P7)	keduanya	Titik berdekatan berlabel sama; jarak bermakna; skala seragam	$O(1)$	$O(nd)$ tiap pertanyaan	menengah: tunjukkan tetangganya
Regresi linier (P7)	regresi	Hubungan linier; galat bebas, seragam, Gaussian; kolinearitas rendah	$O(nd^2 + d^3)$	$O(d)$	sangat tinggi: koefisien per fitur

n = jumlah data, d = jumlah fitur, c = jumlah kelas, B = jumlah pohon, $|SV|$ = jumlah *support vector*. Angka kompleksitas adalah gambaran kasar.

Peta memilih model



Aturan kerja yang sudah berulang kali kita sebut: mulailah dari patokan yang sederhana — regresi linier atau pohon tunggal — lalu naikkan kerumitan hanya jika angka pada data validasi benar-benar membaik.

Tugas besar UTS — bobot 35%

Bentuk. Berkelompok 3–4 orang. Keluarannya tiga: repositori kode, laporan 8–12 halaman, dan presentasi 12 menit ditambah tanya jawab.

Isi yang wajib ada.

1. **Perumusan masalah** dalam bahasa Mitchell: sebutkan *T*, *P*, dan *E*, serta mengapa masalah ini layak diselesaikan dengan ML.
2. **Data.** Data publik atau data mitra dengan izin. Jelaskan asal, jumlah baris dan kolom, serta keterbatasannya.
3. **Praproses.** Nilai hilang, pengkodean kategorikal, penskalaan fitur, pencilan. Semua transformasi dipasang **hanya** pada data latih.
4. **Minimal tiga model pembanding** dari Pertemuan 2–7, salah satunya model sederhana sebagai patokan.
5. **Protokol evaluasi** sesuai Pertemuan 5: pemisahan latih/validasi/uji atau validasi silang bersarang, metrik yang cocok, dan selang ketidakpastian — bukan satu angka tunggal.
6. **Pembahasan.** Mengapa model yang menang menang, di mana ia salah, dan apa risikonya bila dipakai sungguhan.

Rubrik ringkas.

Komponen	Bobot
Perumusan masalah dan pemahaman data	15%
Praproses dan rekayasa fitur	15%
Ragam model dan ketepatan penerapannya	20%
Kebenaran protokol evaluasi	25%
Pembahasan, keterbatasan, dan etika	15%
Laporan dan presentasi	10%

Perhatikan bobot terbesar. Kelompok dengan akurasi 84% yang protokolnya benar bernilai lebih tinggi daripada kelompok dengan akurasi 97% yang ternyata membocorkan data uji ke dalam praproses.

Alat bantu AI boleh dipakai, asalkan dituliskan bagian mana yang dibantu dan setiap anggota mampu menjelaskan kodenya. Judul dan data disetujui satu pekan sebelum pengumpulan.

Yang perlu Anda siapkan

1. Baca Mitchell Bab 8 untuk kNN dan *locally weighted regression*, lalu satu bab pengantar regresi linier — misalnya *An Introduction to Statistical Learning* Bab 3.
2. Latihan tangan, tanpa komputer: ulangi contoh lima titik hari ini dengan mengganti y_3 dari 5 menjadi 3. Hitung ulang $\hat{\beta}_1$, $\hat{\beta}_0$, RSS, dan R^2, lalu jelaskan arah perubahannya.
3. Latihan kode: pada satu data tabular pilihan Anda, jalankan kNN dengan dan tanpa penskalaan fitur, untuk $k \in \{1, 3, 5, 9, 15, 25\}$. Gambarkan galat validasi silang terhadap k untuk kedua kasus dalam satu grafik.
4. Bentuk kelompok tugas besar, lalu kirimkan calon masalah dan calon sumber data.

Setelah UTS

Kita meninggalkan model yang punya rumus, dan mulai mencari parameter dengan cara menuruni permukaan galat: *loss function*, *gradient descent*, SGD, lalu perceptron dan jaringan saraf.

Terima kasih

Pertanyaan?