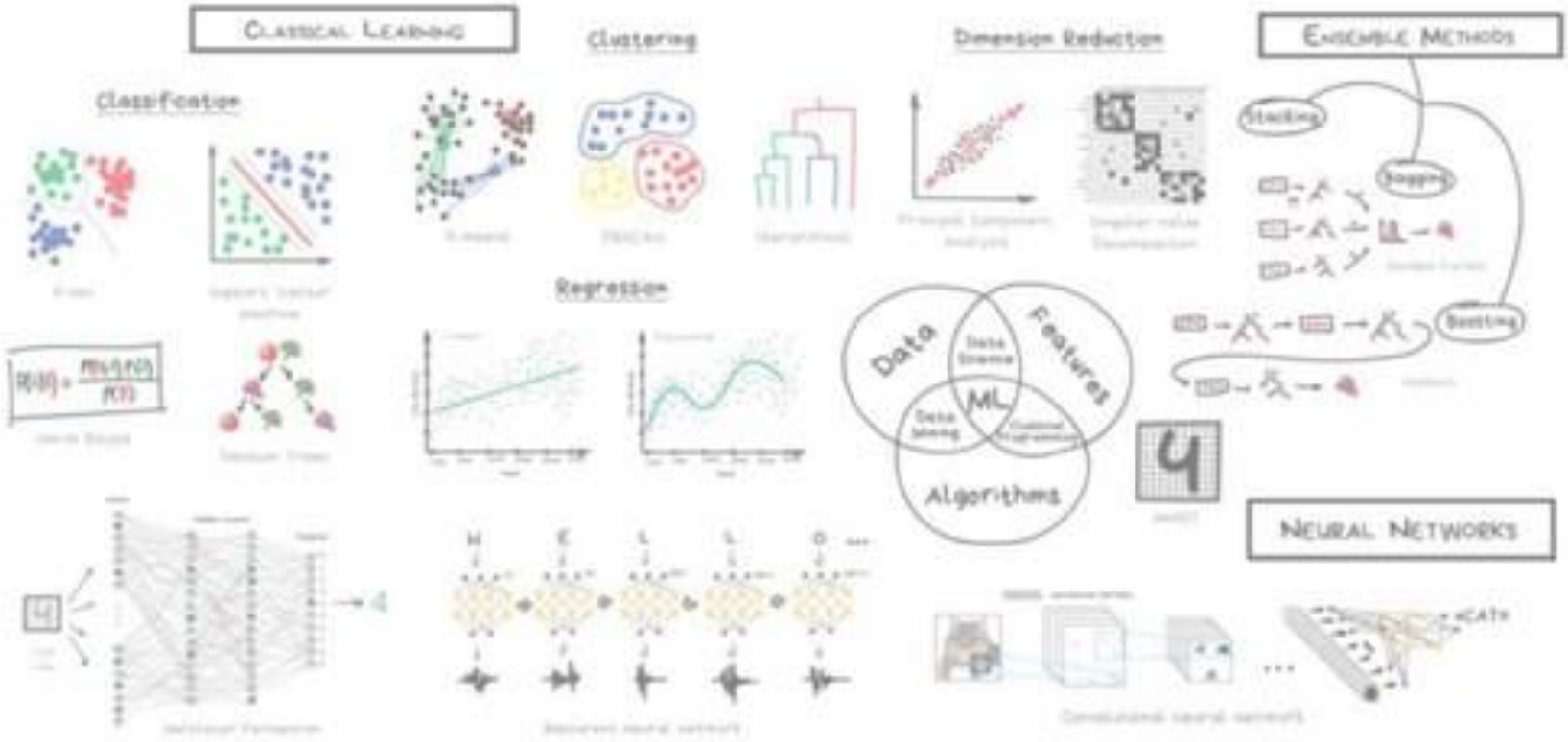
Основи машинного навчання



Лекція № 5

Постановка задачі навчання

- Розділити дані на дві частини:
  - навчальна вибірка (більша частина)
  - тестувальна вибірка
- Навчити машину за існуючою базою даних (навчальною вибіркою) приймати необхідне рішення – побудувати алгоритм прийняття рішень
- Перевірити побудований алгоритм на тестувальній вибірці

Постановка задачі навчання

- Множина об'єктів X^N (інформаційний стан): $\{xi, \dots, xN\}$
- Множина відповідей Y (оцінка, передбачення, прогноз)
- Target function $y: X \to Y$ - невідома залежність яка для кожного інформаційного стану ставить у відповідність певну відповідь

Дано

- Навчальна вибірка $\{x_1, \dots, x_\ell \subset X\}$
- Відомі відповіді $y_i = y(x_i), i = 1, \dots, \ell$

Знайти

- $a: X \to Y$ алгоритм, що визначає функцію y та наближає її на всій множині X

Ознаковий опис

$f_j: X \to Dj, j = 1, \ldots, n$ ознаки об'єктів

- $D_j = \{0,1\}$ — бінарна ознака f_j
- $|D_j| < \infty$ — номінальна ознака f_j
- $|D_j| < \infty, D_j$ упорядковано — порядкова ознака f_j
- $D_j = \mathbb{R}$ — кількісна ознака f_j

Вектор $(f_1(x), \ldots, fn(x))$ — ознаковий опис об'єкта x

Набір даних - матриця «об'єкти-ознаки»:

ℓ об'єктів з n признаками

$$F = \|f_j(xi)\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \cdots & f_n(x_1) \\ \cdots & \cdots & \cdots \\ f_1(x_\ell) & \cdots & f_n(x_\ell) \end{pmatrix}$$

Відповіді та приклади задач

Задачі навчання з учителем

Задачі класифікації (classification)

- $Y = \{-1, +1\}$ – класифікація на два класи
- $Y = \{1, M\}$ – класифікація на M класів, що не перетинаються
- $Y = \{0,1\}^M$ – класифікація M класів, що можуть перетинатися

Задачі відтворення регресії (regression)

- $Y = R$ або $Y = RM$

Задачі ранжування (ranking,learning to rank)

- Y – кінцева упорядкована множина

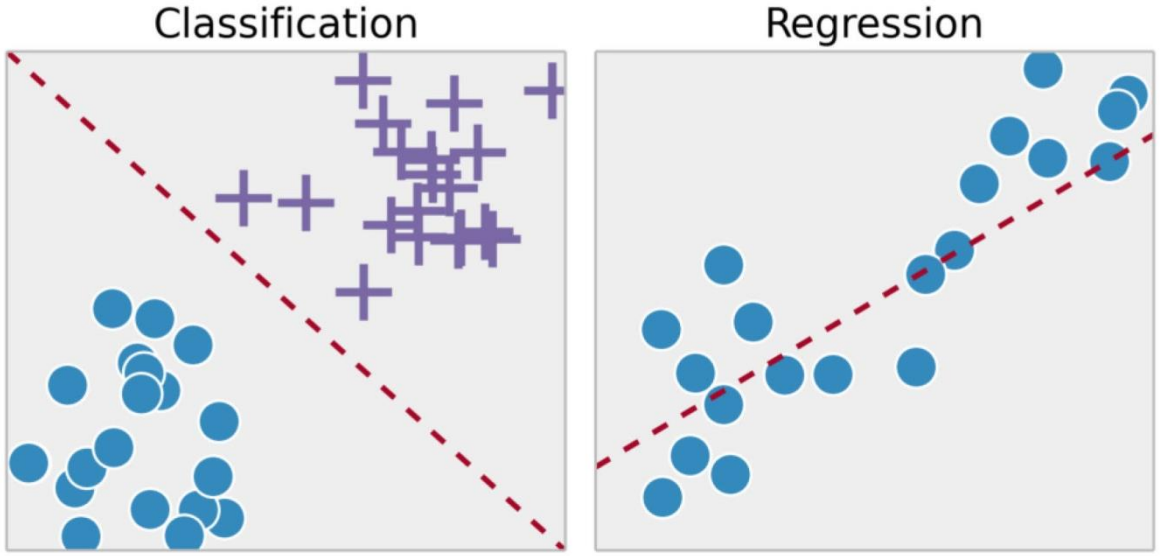
Задачі навчання без учителя

- Відповідей немає. Необхідно щось робити із самими об’єктами.

Статистичне навчання з учителем

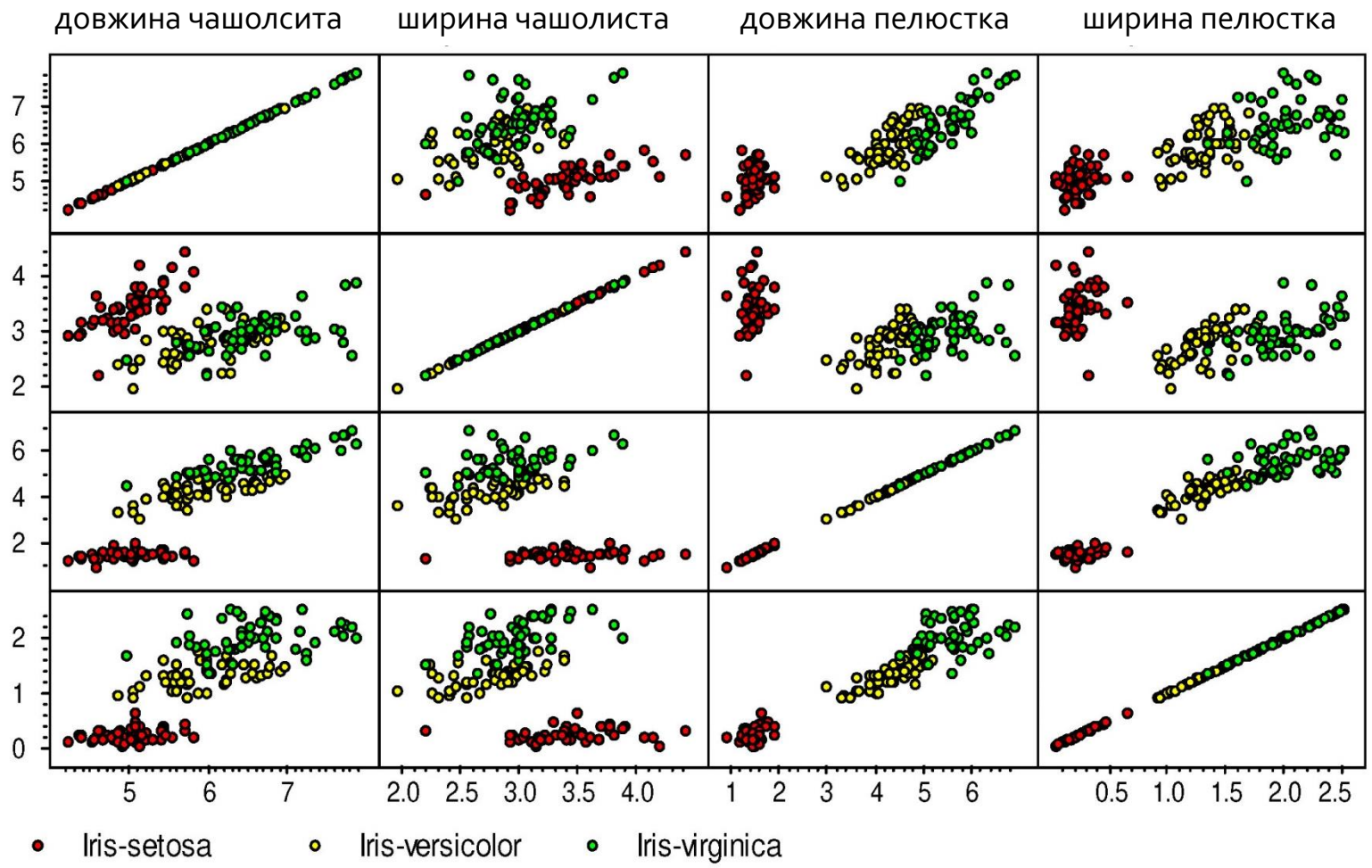
- Навчання за прецедентами
- Відтворення залежностей за існуючими емпіричними даними
- Передбачувальне моделювання
- Апроксимація функцій за заданими точками

Два основних типи задач: класифікація та регресія



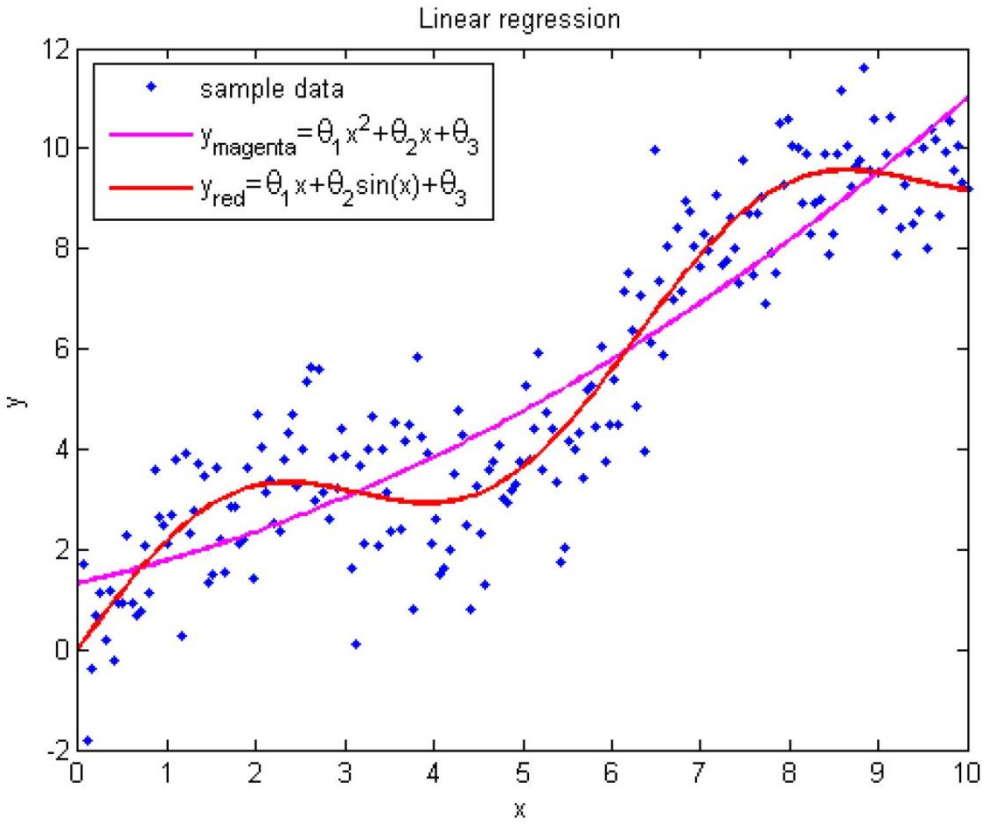
Приклад: класифікація квітів ірису (Фішер 1936)

n = 4 ознаки, Y = 3 класи, довжина вибірки ℓ = 150



Приклад: задача регресії

$X = Y = \mathbb{R}$, $\ell = 200$, $n = 3$ ознаки: $\{x, x^2, 1\}$ або $\{x, \sin x, 1\}$



- Генерація ознак (features generation) збагачує модель
- На практиці дуже важливо «правильно вгадати модель»

Модель алгоритмів (передбачувана модель)

Модель (predictive model) – параметричне сімейство функцій

$$A = \{g(x, \theta) \mid \theta \in \Theta\},$$

де $g : X \times \Theta \to Y$ — фіксована функція

Θ — Множина допустимих значень параметра θ

Приклад

Лінійна модель з вектором параметрів $\theta = (\theta_1, \ldots, \theta_n) \in \mathbb{R}^n$:

$g(x, \theta) = \sum_{j=1}^{n} \theta_j f_j(x)$ — для регресії та ранжування, $Y = \mathbb{R}$

$g(x, \theta) = \text{sign} \sum_{j=1}^{n} \theta_j f_j(x)$ – для класифікації, $Y = \{-1, +1\}$

Метод навчання

Етап навчання (train) :

Метод навчання (training algorithm) $\mu\colon (X \times Y)^\ell \to A$

За вибіркою $X^\ell = (x_i, y_i)_{i=1}^\ell$ будує алгоритм $a = \mu(X^\ell)$:

$$\begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix} \xrightarrow{y} \begin{pmatrix} y_1 \\ \dots \\ y_\ell \end{pmatrix} \xrightarrow{\mu} a$$

Етап застосування (test) :

Алгоритм a для нових об'єктів x'_i видає відповіді $a(x'_i)$:

$$\begin{pmatrix} f_1(x'_1) & \dots & f_n(x'_1) \\ \dots & \dots & \dots \\ f_1(x'_k) & \dots & f_n(x'_k) \end{pmatrix} \xrightarrow{a} \begin{pmatrix} a(x'_1) \\ \dots \\ a(x'_k) \end{pmatrix}$$

Функціонали якості

$\mathscr{L}(a, x)$ — функція втрат (loss function) – величина помилки алгоритму $a \in A$ на об'єкті $x \in X$

Функція втрат для задач класифікації:

- $\mathscr{L}(a, x) = \left[a(x) \neq y(x)\right]$ — індикатор помилки

Функція втрат для задач регресії:

- $\mathscr{L}(a, x) = \left|a(x) - y(x)\right|$ — абсолютне значення помилки
- $\mathscr{L}(a, x) = \left(a(x) - y(x)\right)^2$ — квадратична помилка

Емпіричний ризик – функціонал якості алгоритму a на вибірці X^ℓ

$$Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(a, x_i)$$

Зведення задачі навчання до задачі оптимізації

Метод *мінімізації емпіричного ризику* (Empirical Risk Minimization, ERM):

$$\mu(X^\ell) = \arg\min_{a \in A} Q(a, X^\ell)$$

Приклад: задача регресії $Y = \mathbb{R}$;

n числових ознак $f_j(x),\ j = 1, \ldots, n$;

лінійна модель регресії: $g(x, \theta) = \sum_{j=1}^{n} \theta_j f_j(x),\ \theta \in \mathbb{R}^n$;

квадратична функція втрат: $\mathscr{L}(a, x) = \big(a(x) - y(x)\big)^2$.

Метод найменший квадратів – окремий випадок ERM:

$$\mu(X^\ell) = \arg\min_{\theta} \sum_{i=1}^{\ell} \big(g(x_i, \theta) - y_i\big)^2$$

Приклад Рунге
Апроксимація функції поліномом

Функція $y(x) = \dfrac{1}{1+25x^2}$ на відрізку $x \in [-2, 2]$

Ознаковий опис об'єкта $x \mapsto (1, x^1, x^2, \ldots, x^n)$

Модель поліноміальної регресії

$a(x, \theta) = \theta_0 + \theta_1 x + \cdots + \theta_n x^n$ — поліном ступеня n

Навчання методом найменших квадратів:

$$Q(\theta, X^\ell) = \sum_{i=1}^{\ell} (\theta_0 + \theta_1 x_i + \cdots + \theta_n x_i^n - y_i)^2 \to \min_{\theta_0,\ldots,\theta_n}$$

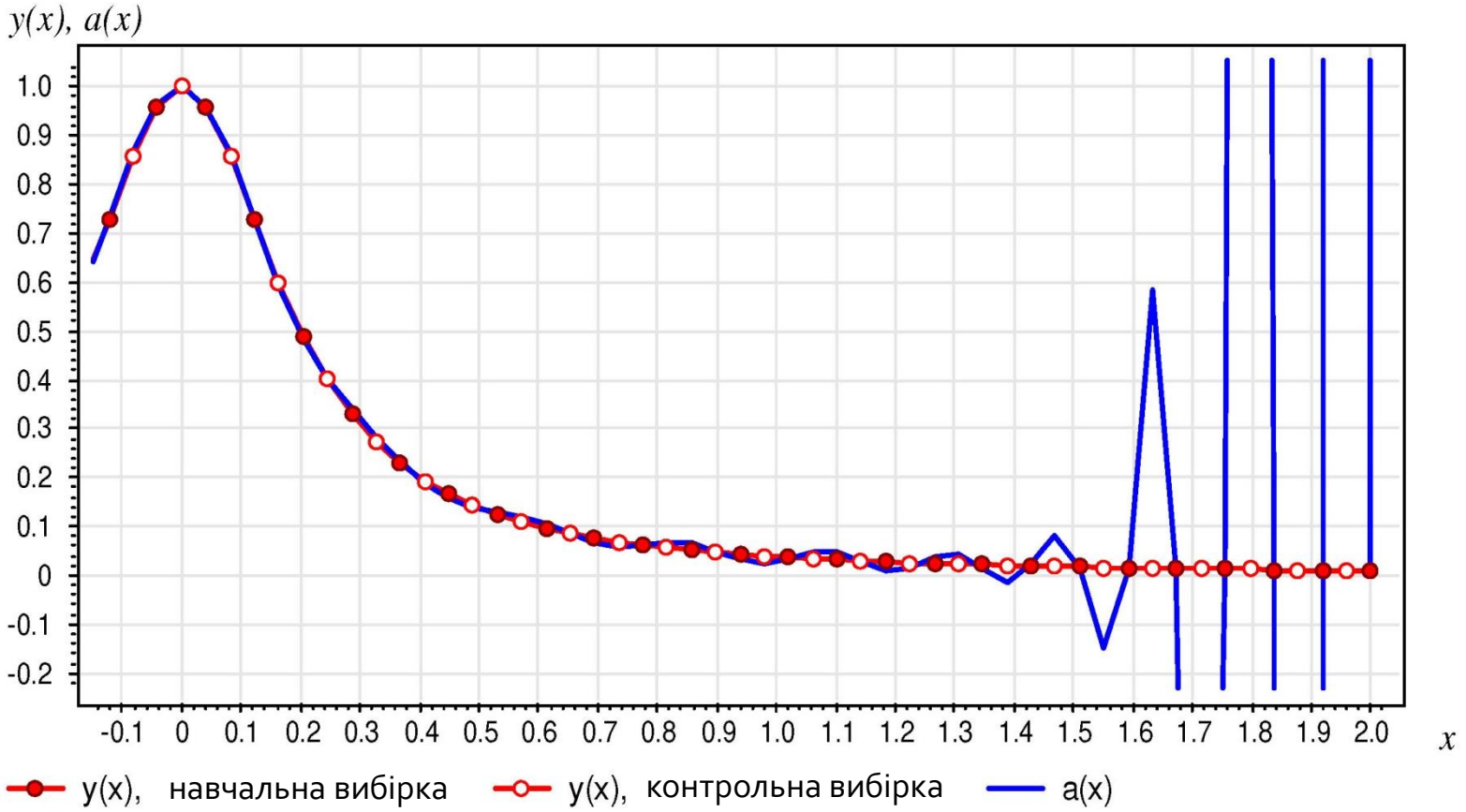
Навчальна вибірка: $X^\ell = \left\{ x_i = 4\frac{i-1}{\ell-1} - 2 \;\middle|\; i = 1, \ldots, \ell \right\}$

Контрольна вибірка: $X^k = \left\{ x_i = 4\frac{i-0.5}{\ell-1} - 2 \;\middle|\; i = 1, \ldots, \ell-1 \right\}$

Що відбувається з $Q(\theta, X^\ell)$ та $Q(\theta, X^k)$ при збільшенні n ?

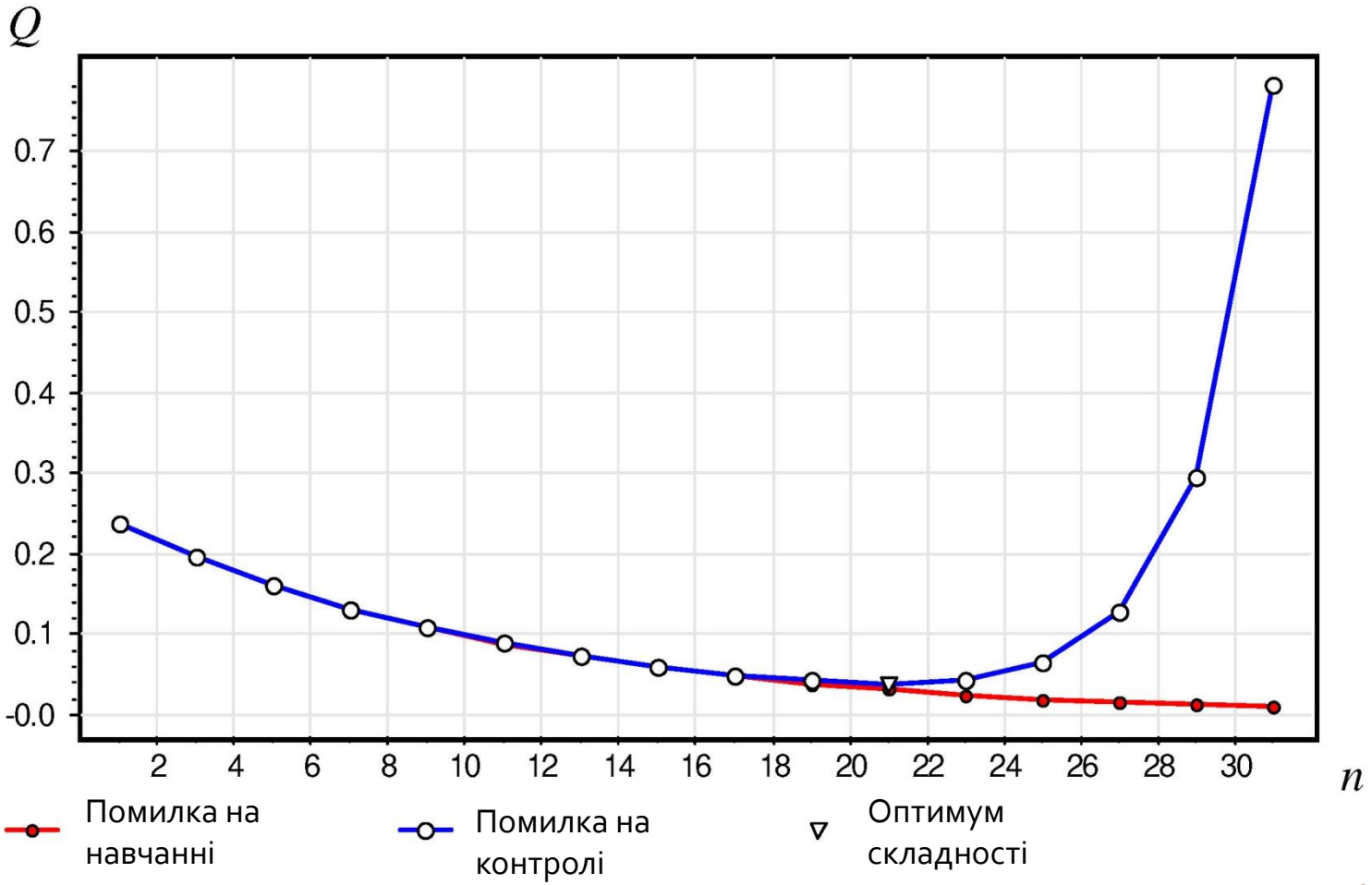
Приклад Рунге
Перенавчання при $n = 38, \ell = 50$

$y(x) = \dfrac{1}{1 + 25x^2}$; $a(x)$ — поліном ступеня n = 38

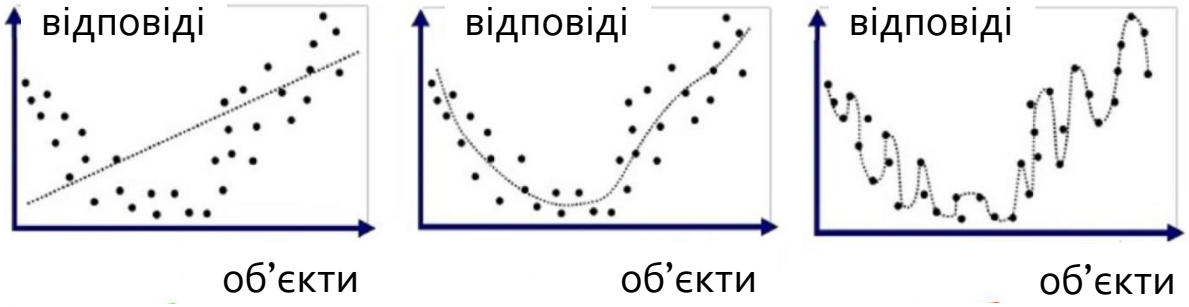


Приклад Рунге
Залежність Q від ступеня поліному n

Перенавчання – це коли $Q\left(\mu(X^\ell), X^k\right) \gg Q\left(\mu(X^\ell), X^\ell\right)$:

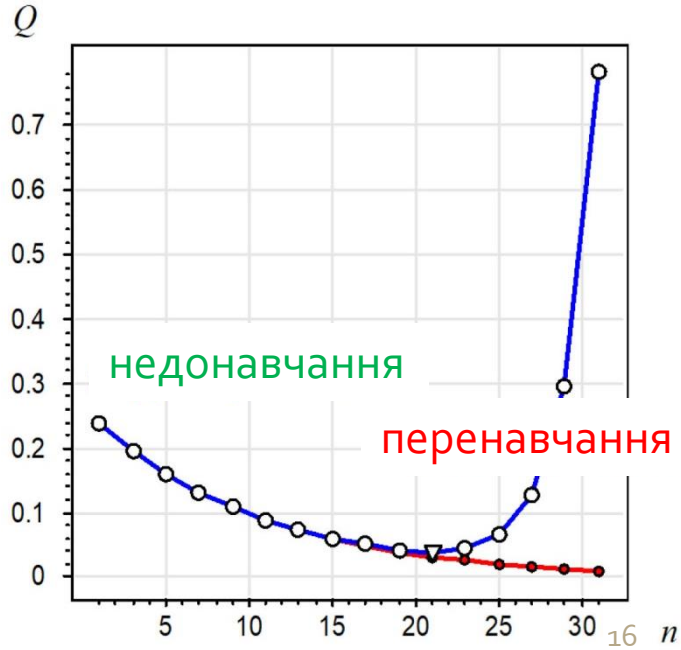


Проблеми недонавчання та перенавчання



недонавчання перенавчання

- Недонавчання (underfitting): модель досить проста, недостатня кількість параметрів *n* (ознак)
- Перенавчання (overfitting:) модель досить складна, завелика кількість параметрів *n* (ознак)



Навчання регресії – оптимізація

Навчальна вибірка: $X^\ell = (x_i, y_i)_{i=1}^\ell, \quad x_i \in \mathbb{R}^n, \quad y_i \in \mathbb{R}$

1. Модель регресії – *лінійна*:

$$a(x, w) = \langle x, w \rangle = \sum_{j=1}^{n} w_j f_j(x), \qquad w \in \mathbb{R}^n$$

2. Функція втрат – *квадратична*:

$$\mathscr{L}(a, y) = (a - y)^2$$

3. Метод навчання – *метод найменших квадратів*:

$$Q(w) = \sum_{i=1}^{\ell} \big(a(x_i, w) - y_i\big)^2 \to \min_w$$

4. Перевірка по тестовій вибірці $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:

$$\bar{Q}(w) = \frac{1}{k} \sum_{i=1}^{k} \big(a(\tilde{x}_i, w) - \tilde{y}_i\big)^2$$

Навчання класифікації – оптимізація

Навчальна вибірка: $X^\ell = (x_i, y_i)_{i=1}^\ell, \quad x_i \in \mathbb{R}^n, \quad y_i \in \{-1, +1\}$

1. Модель класифікації – *лінійна*:

$$a(x, w) = \operatorname{sign}\langle x, w\rangle = \operatorname{sign} \sum_{j=1}^{n} w_j f_j(x)$$

2. Функція втрат – *бінарна або її апроксимація*:

$$\mathscr{L}(a, y) = \big[ay < 0\big] = \big[\langle x, w\rangle y < 0\big] \leqslant \mathscr{L}\big(\langle x, w\rangle y\big)$$

3. Метод навчання – *мінімізація емпіричного ризику*:

$$Q(w) = \sum_{i=1}^{\ell} \big[\langle x_i, w\rangle y_i < 0\big] \leqslant \sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle y_i\big) \to \min_w$$

4. Перевірка по тестовій вибірці $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:

$$\bar{Q}(w) = \frac{1}{k} \sum_{i=1}^{k} \big[\langle \tilde{x}_i, w\rangle \tilde{y}_i < 0\big]$$

Поняття відступу (marging) для розділяючих класифікаторів

Розділяючий класифікатор: $a(x, w) = \text{sign}\, g(x, w)$

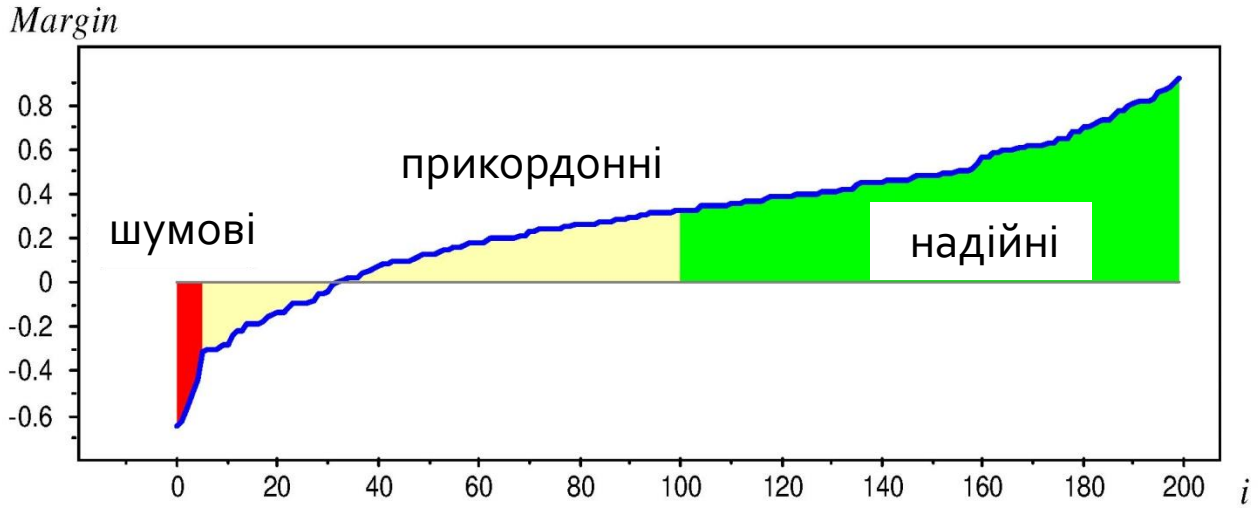
$g(x, w)$ — Розділяюча (дискримінантна) функція

$g(x, w) = 0$ — Рівняння розділяюччої поверхні

$M_i(w) = g(x_i, w)y_i$ — Відступ (marging) об'єкта x_i

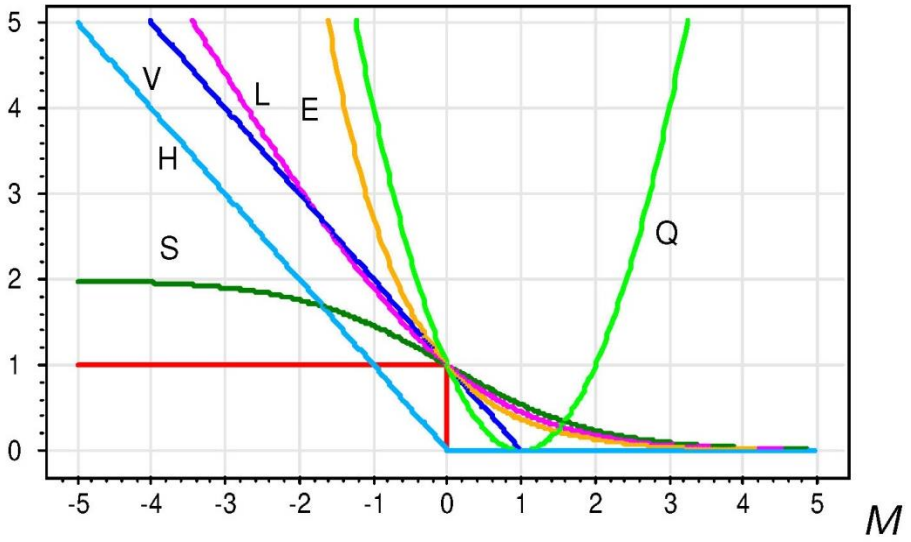
$M_i(w) < 0 \iff$ алгоритм $a(x, w)$ помиляється на x_i

Ранжування об'єктів за зростанням відступів $M_i(w)$:



Неперервні апроксимації порогової функції втрат

Часто використовувані неперервні функції втрат $\mathscr{L}(M)$:



$V(M) = (1 - M)_+$ – кусочно-лінійна (SVM);

$H(M) = (-M)_+$ – кусочно-лінійна (Hebb's rule);

$L(M) = \log_2(1 + e^{-M})$ – логарифмічна (LR);

$Q(M) = (1 - M)^2$ – квадратична (FLD);

$S(M) = 2(1 + e^M)^{-1}$ – сигмоїдна (ANN);

$E(M) = e^{-M}$ – експоненціальна (AdaBoost);

$[M < 0]$ – порогова функція втрат

Метод градієнтного спуску для мінімізації емпіричного ризику

Мінімізація емпіричного ризику (регресія, класифікація):

$$Q(w) = \sum_{i=1}^{\ell} \mathscr{L}_i(w) \to \min_w .$$

Мінімізація методом градієнтного спуску:

$w^{(0)}$ – початкове наближення

$$w^{(t+1)} := w^{(t)} - h \cdot \nabla Q(w^{(t)}), \qquad \nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j} \right)_{j=0}^{n},$$

де h – градієнтний крок, також називається темпом навчання

$$w^{(t+1)} := w^{(t)} - h \sum_{i=1}^{\ell} \nabla \mathscr{L}_i\left(w^{(t)}\right).$$

Ідея прискорення збіжності алгоритму:

брати (x_i, y_i) по одному й одразу поновлювати вектор ваг

Перенавчання — ключов проблема в машинному навчанні

- **Через що виникає перенавчання?**
  - Надлишкові параметри в моделі $g(x, \theta)$ «витрачаються» на надмірно точну підгонку за навчальною вибіркою
  - Вибір алгоритму a з A відбувається за неповною інформацією X^ℓ
- **Як виявити перенавчання?**
  - Емпірично, шляхом розбиття вибірки на навчальну (train) та тестову (test), причому на тестовій вибірці мають бути відомі правильні відповіді
- **Позбутися його не можна. Як його мінімізувати?**
  - Накладати обмеження на θ (регуляризація)
  - Мінімізувати одну з теоретичних оцінок

Дякую за увагу