Chương 2.
Học có giám sát

Ts. Nguyễn An Tế

Khoa CNTT kinh doanh – ĐH Kinh tế TPHCM

tena@ueh.edu.vn

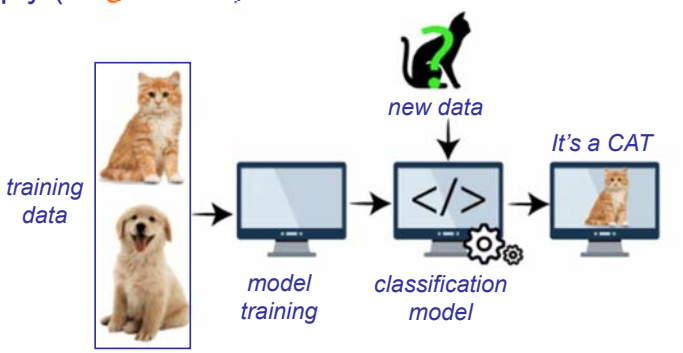
2025

Nội dung

1. **Học có giám sát**
2. **Một số phương pháp học có giám sát**

1. Học có giám sát

- **Học có giám sát** (*Supervised Learning*) **sắp xếp items vào K lớp biết trước → phân lớp** (*Classification*)
  - xây dựng mô hình phân lớp dựa trên các quan sát đã biết (*Learning by Examples*)
  - xác định các lớp/nhãn (*label*): nominal, ordinal data
  - hồi quy (*Regression*): numerical data



1. Học có giám sát

- **Học có giám sát** (*Supervised Learning*)
  - trước khi khảo sát dữ liệu: chưa nhận diện được các lớp
  - sau khi khảo sát dữ liệu: tất cả các lớp đều được nhận diện liên quan đến những đặc trưng của dữ liệu

predefined classes: sau khi khảo sát dữ liệu !

1. Học có giám sát

- **Phân lớp** (*Classification*): **sắp xếp items vào K lớp biết trước**
  - gán nhãn, dự báo
  - mô tả

features

target attribute ∈ { YES, NO }

TRAINING SET

Age	Income	Student	Buy
youth	high	no	NO
youth	high	no	NO
middle	high	no	YES
senior	medium	no	YES
senior	low	yes	YES
senior	low	yes	NO
middle	low	yes	YES
youth	medium	no	NO
youth	medium	yes	YES
senior	medium	yes	YES

classes

các quan sát (observations)

➡ (youth, medium, yes, **?**)

1. Học có giám sát

- **Phân lớp** (*Classification*): **ứng dụng dự báo**

Thời tiết: có mưa hay không ?

Sức gió, độ ẩm,…

Kinh doanh: doanh số trong tháng sẽ tăng hay giảm ?

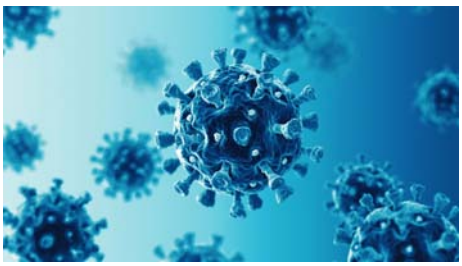
Chỉ số tiêu dùng, yếu tố xã hội, lễ-Tết, sự kiện,…

Thị trường chứng khoán: cổ phiếu X lên hay xuống ?

Giá vàng, giá ngoại tệ, bất động sản,…

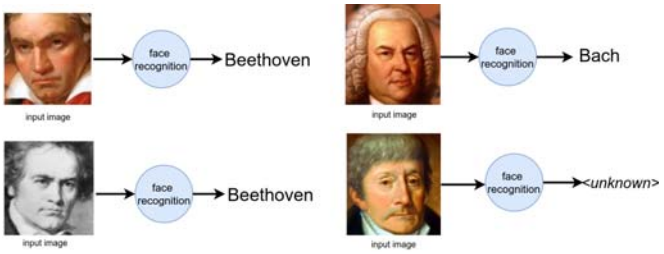
1. Học có giám sát

- **Phân lớp nhị phân** (*Binary Classification*)**: tổng số lớp K = 2**
  - ứng dụng: chẩn đoán y khoa, ngân hàng-tín dụng, phát hiện gian lận, spam, …
  - phương pháp phổ biến: *Logistic Regression, Decision Trees, Support Vector Machine, Naïve Bayes*, …



1. Học có giám sát

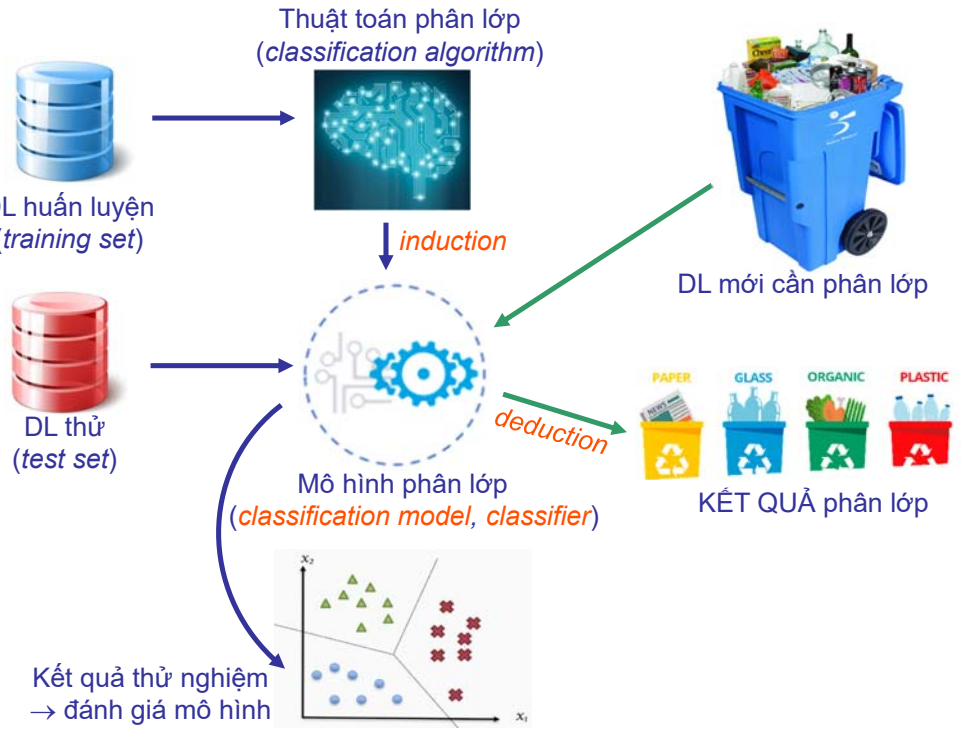
- **Phân lớp đa lớp** (*Multi-class Classification*)**: tổng số lớp K > 2**
  - ứng dụng: nhận dạng khuôn mặt (*Face Recognition*), chữ viết (*Optical Character Recognition*), giống loài sinh vật, …
  - phương pháp phổ biến: *Random Forest, Gradient Boosting*, *Logistic Regression*, *Support Vector Machine*, …



1. Học có giám sát

- **Phân lớp đa nhãn** (*Multi-label Classification*)**: 1 item có thể thuộc nhiều hơn 1 lớp**
  - ứng dụng: phân loại (chủ đề) văn bản/ảnh, tagging, …
  - phương pháp: cải biên từ các phương pháp binary/multi-class

1. Học có giám sát

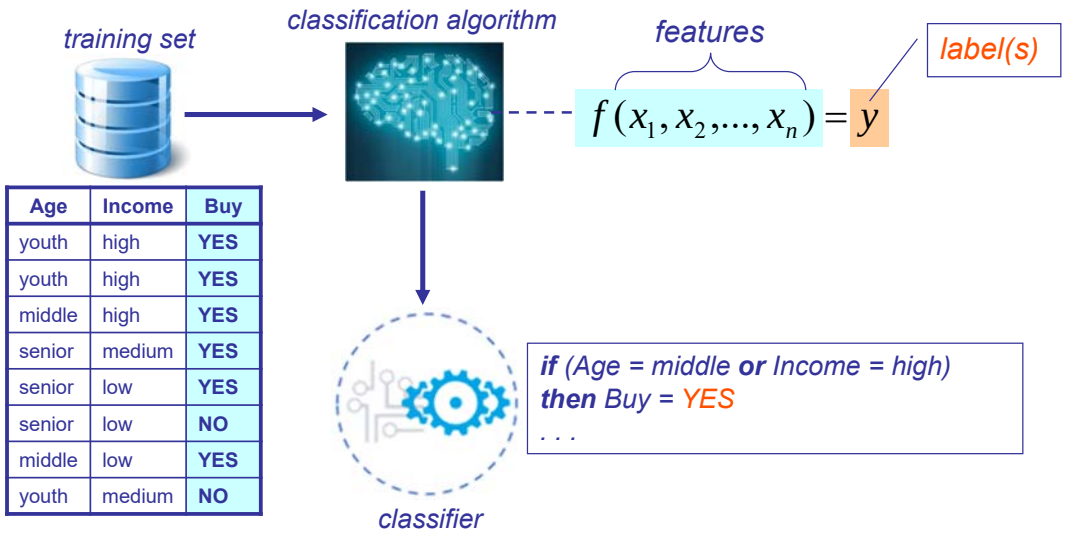


1. Học có giám sát

- **Quy trình 2 bước** (*Two-Step Process*)
  - **B1:** Xây dựng mô hình phân lớp (*Model Construction*)
  - **B2:** Sử dụng mô hình phân lớp (*Model Usage*)
    - + Đánh giá mô hình phân lớp (độ chính xác, …)
    - + Phân lớp những dữ liệu mới

1. Học có giám sát

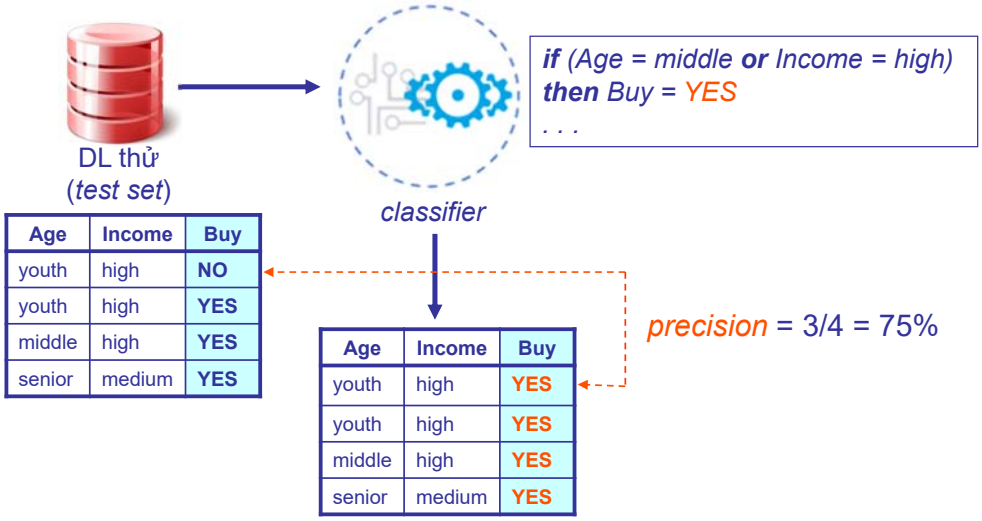
- **Bước B1: Xây dựng mô hình phân lớp**
  - giai đoạn huấn luyện (*training*)



Age	Income	Buy
youth	high	YES
youth	high	YES
middle	high	YES
senior	medium	YES
senior	low	YES
senior	low	NO
middle	low	YES
youth	medium	NO

1. Học có giám sát

- **Bước B2.1: Đánh giá mô hình phân lớp**
  - giai đoạn thử nghiệm, đánh giá (*testing*)

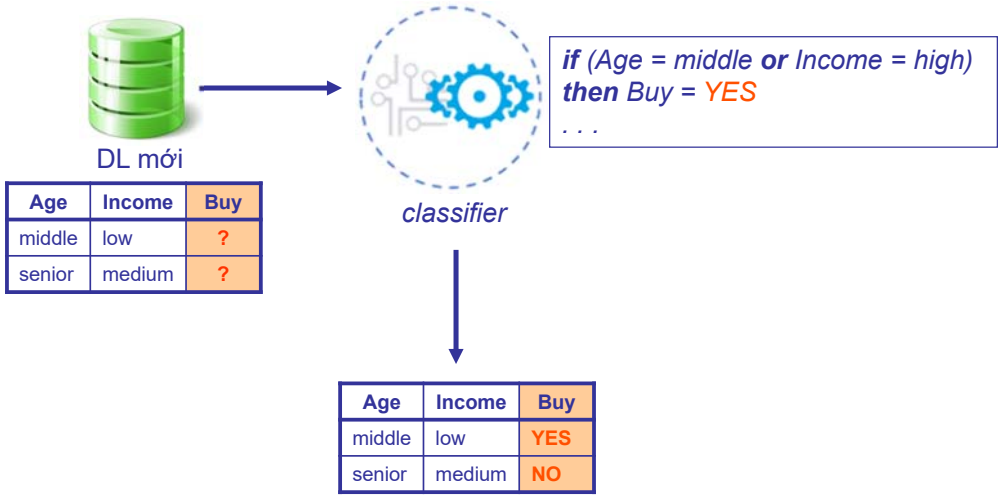


Age	Income	Buy
youth	high	NO
youth	high	YES
middle	high	YES
senior	medium	YES

Age	Income	Buy
youth	high	YES
youth	high	YES
middle	high	YES
senior	medium	YES

1. Học có giám sát

- **Bước B2.2: Phân lớp dữ liệu mới**



Age	Income	Buy
middle	low	?
senior	medium	?

Age	Income	Buy
middle	low	YES
senior	medium	NO

1. Học có giám sát

- **Phương pháp có tham số** (*Parametric Methods*)

output

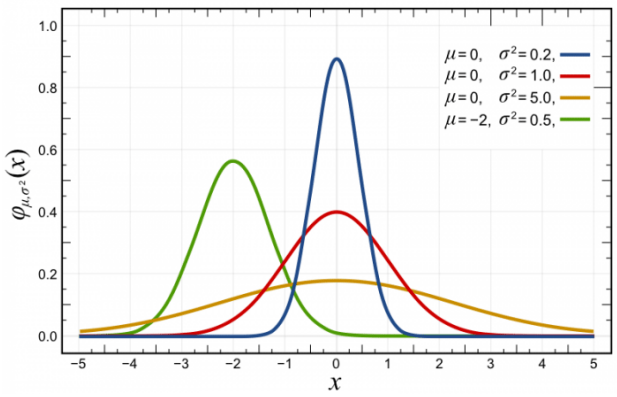
"A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a parametric model." [Russell+]

intput

"… parametric where we assume that the sample is drawn from some distribution that obeys a known model, for example, Gaussian." [Alpaydin]

1. Học có giám sát

- **Phương pháp có tham số** (*Parametric Methods*)
  - mô hình được dựa trên một số lượng parameters không nhiều
  - phương pháp ước lượng các tham số của phân phối giả định: (*Maximum Likelihood Estimation – MLE*)



1. Học có giám sát

- **Phương pháp phi tham số** (*Nonparametric Methods*)

> *"A nonparametric model is one that cannot be characterized by a bounded set of parameters."* [Russell+]

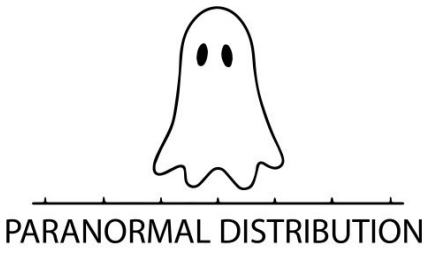
> *"Nonparametric methods do not assume any a priori parametric form for the underlying densities ; … a nonparametric model is not fixed but its complexity depends on the size of the training set."* [Alpaydin]

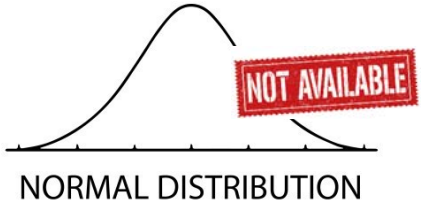
PHI ≠ KHÔNG

Số lượng tham số được xác định TRƯỚC giai đoạn huấn luyện ?

1. Học có giám sát

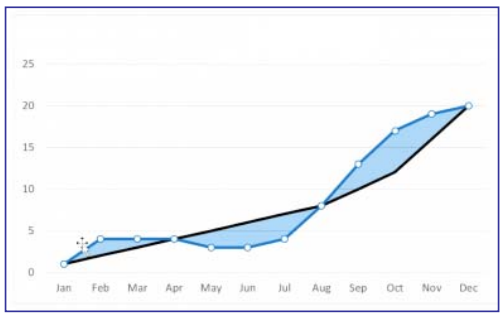
- **Phương pháp phi tham số** (*Nonparametric Methods*)
  - không dựa trên bất kỳ giả thiết nào về phân phối của dữ liệu
  - khai thác những "yếu tố" từ chính bản thân dữ liệu





1. Học có giám sát

- **Phương pháp phi tham số** (*Nonparametric Methods*)
  - "*similar inputs have similar outputs*"
  - tốc độ biến thiên chậm của các hàm
  - tính chất tương đồng trong lân cận (láng giềng)



1. Học có giám sát

- **Phương pháp phi tham số: cách tiếp cận để tạo ouputs**
  - không dựa trên một mô hình toàn cục (*global model*)
  - tìm kiếm những thể hiện tương tự với inputs → áp dụng nhiều mô hình cục bộ (*local models*)
  - áp dụng phương pháp nội suy (*interpolation*)
  - độ phức tạp phụ thuộc vào kích thước dữ liệu

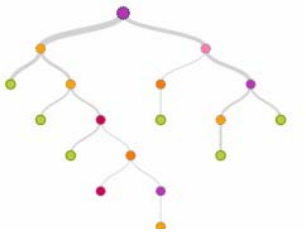
1. Học có giám sát

- **Học phi tham số** *Instance-based* **hay** *Memory-based Learning*
  - lưu trữ dữ liệu huấn luyện (*training instances*) → nội suy
  - độ phức tạp không gian lưu trữ: O(N)
  - độ phức tạp tìm kiếm những thể hiện tương tự với input: O(N)

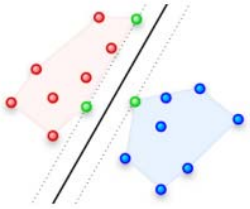
1. Học có giám sát

k-Nearest Neighbors

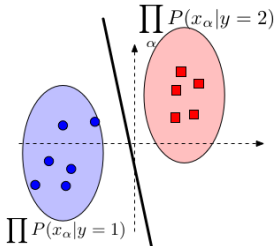
classification algorithms



Decision Tree (ID3, C4.5, …)



Support Vector Machine (SVM)



Naïve Bayes

Nội dung

1. Học có giám sát

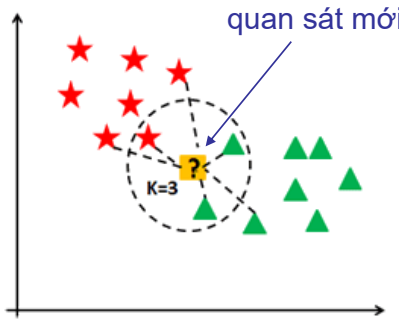
2. **Một số phương pháp học có giám sát**
   - k-NN (*k-Nearest Neighbors*)
   - Cây quyết định (*Decision Tree*)
   - Naïve Bayes Classification
   - SVM (*Support Vector Machine*)

2.1 K-NN (*k-Nearest Neighbors*)

❑ **Phương pháp** *Lazy Learning*
   - target của một quan sát mới được dựa trên những "láng giềng" (gần nhất)
   - trì hoãn việc tính toán, xây dựng mô hình (cục bộ) cho đến khi xuất hiện quan sát mới (>< *Earger Learning*)
     → giai đoạn dự đoán >> giai đoạn "học" (lưu trữ các quan sát)
   - không lưu lại những kết quả trung gian

2.1 K-NN

- **Giá trị target của quan sát mới**
  - *classification*: chọn lớp phổ biến trong số k láng giềng
  - *regression*: trung bình giá trị target của k láng giềng



2.1 K-NN

- **Hai vấn đề quan trọng**
  - độ tương đồng (*similarity*), khoảng cách (*distance*)
  - số lượng láng giềng (k)

2.1 K-NN

- **Độ tương đồng** (*numerical data*)
  - cosine:

$$sim(x,y) = \frac{\sum_{i=1}^{n} x_i . y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

  - tích vô hướng (*scalar product*):

$$sim(x,y) = \langle x, y \rangle = \sum_{i=1}^{n} x_i . y_i$$

Đối với *categorical data*: so sánh giá trị (*Hamming distance*)

2.1 K-NN

- **Hàm khoảng cách** (*numerical data*)
  - khoảng cách Manhattan:

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

  - khoảng cách Euclid:

$$d(x,y) = \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2}$$

  - khoảng cách Minkowski: (p > 2)

$$d(x,y) = \sum_{i=1}^{n} \left(|x_i - y_i|^p \right)^{1/p}$$

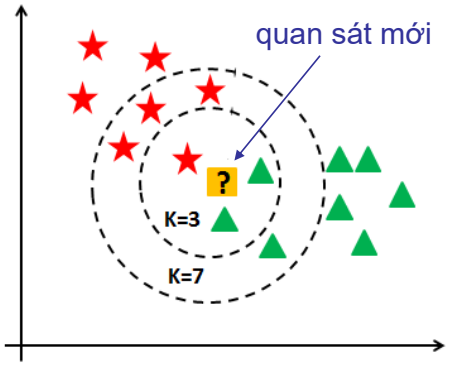
Đối với *categorical data*: so sánh giá trị (*Hamming distance*)

2.1 K-NN

- **Xác định số lượng láng giềng k**

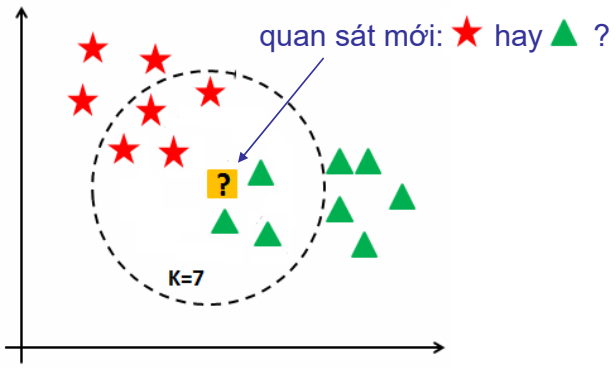
  Với k = 3: ? → ▲

  Với k = 7: ? → ★



2.1 K-NN

- **Vai trò của các láng giềng**
  - vai trò đều như nhau
  - vai trò phụ thuộc vào khoảng cách (trọng số)

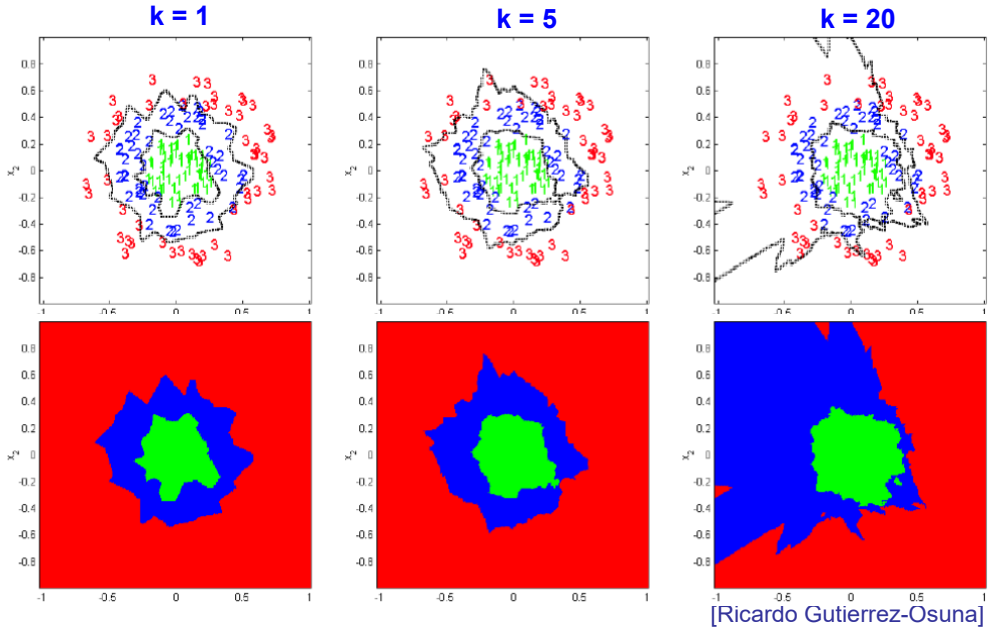


2.1 K-NN

- Xác định số lượng láng giềng k
  - k chẵn hay lẻ ?
  - k = 1: dễ bị ảnh hưởng bởi nhiễu
  - k nhỏ: đường biên quyết định không trơn, dễ gây ra ovetfitting
  - k lớn: phá vỡ những cấu trúc cục bộ (tiềm ẩn) trong dữ liệu
  - khi số lượng quan sát N đủ lớn: k = SQRT(N) / 2

2.1 K-NN

- Xác định số lượng láng giềng k



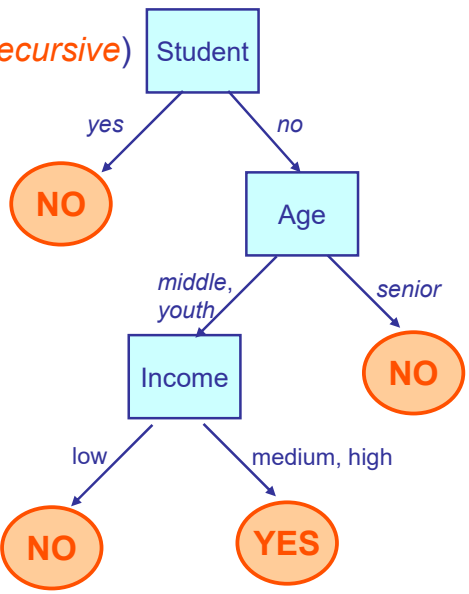
[Ricardo Gutierrez-Osuna]

2.1 K-NN

- **Ưu điểm**
  - đơn giản, dễ triển khai
  - chi phí thấp trong giai đoạn học
  - có thể áp dụng cho classification và regression
  - nhiều khả năng chọn lựa linh hoạt (hàm khoảng cách)
- **Khuyết điểm**
  - xác định giá trị của k
  - chi phí tính toán trong giai đoạn dự đoán
  - kém hiệu quả khi phân phối của target bị lệch

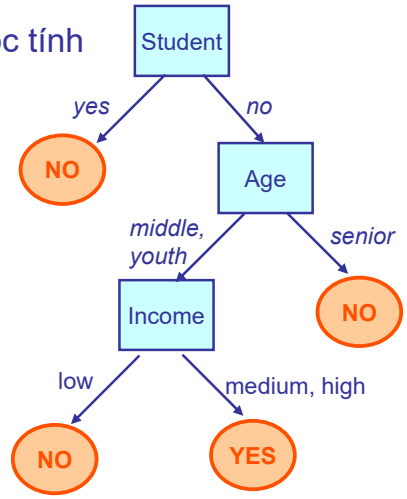
2.2 Cây quyết định (*Decision Tree*)

- **Cách tiếp cận suy diễn theo cấu trúc cây** (*tree induction*)
  - chia để trị (*divide-and-conquer*)
  - đệ quy từ trên xuống (*top-down recursive*)
  - KHÔNG lan truyền ngược (*backpropagation*)



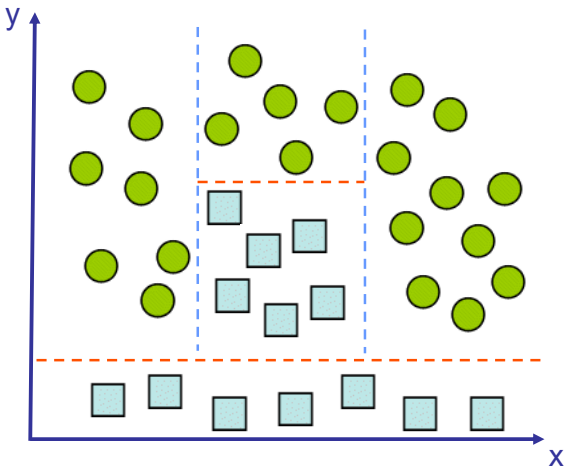
2.2 Cây quyết định

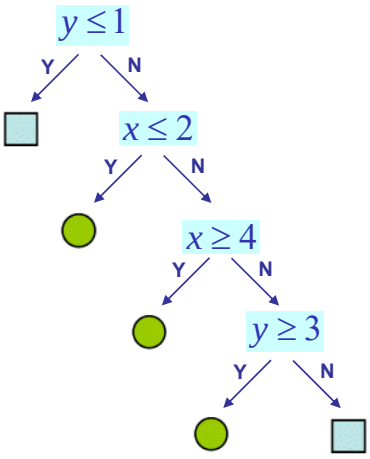
- **Cấu trúc cây quyết định đơn biến** (*univariate tree*)
  - nút lá (*leaf node*): nhãn phân lớp (*decision node*)
  - nút gốc (*root*), nút trong (*internal node*): thuộc tính (kiểm tra)
  - nhánh (*branch*): trường hợp của thuộc tính



2.2 Cây quyết định

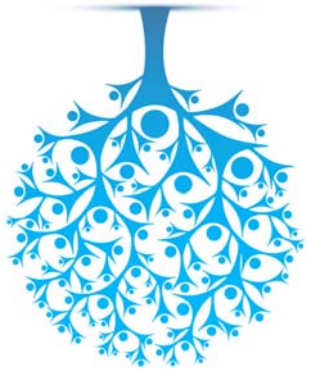
- **Cấu trúc cây quyết định đơn biến** (*univariate tree*)
  - dựa trên các biên quyết định "thẳng" | "phẳng" (*rectilinear decision boundary*)





2.2 Cây quyết định

- **Phương pháp học phi tham số**
  - không cần giả thiết về phân phối của các lớp (nhãn)
  - cấu trúc cây không được xác định trước → gắn liền với dữ liệu quan sát được

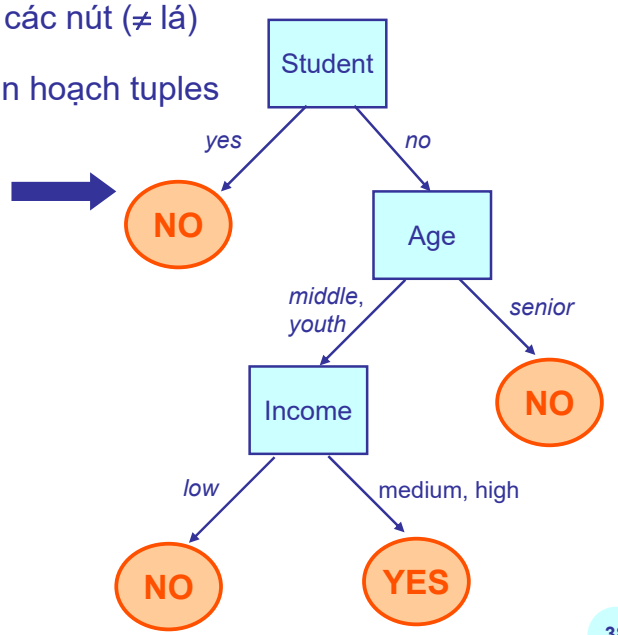


2.2 Cây quyết định

- **Xây dựng cây quyết định từ tập huấn luyện**
  - sắp xếp các thuộc tính → các nút (≠ lá)
  - phân chia các nhánh: phân hoạch tuples

CSDL: training set

Age	Income	Student	Buy
youth	high	no	NO
youth	high	no	NO
middle	high	no	YES
senior	medium	no	YES
senior	low	yes	YES
senior	low	yes	NO
middle	low	yes	YES
youth	medium	no	NO
youth	medium	yes	YES
senior	medium	yes	YES



2.2 Cây quyết định

VD: Tạo phân hoạch các tuples từ thuộc tính Age

Tid	Age	Income	Student	Rating	Buy
T1	youth	high	no	fair	NO
T2	youth	high	no	excellent	NO
T3	middle	high	no	fair	YES
T4	senior	medium	no	fair	YES
T5	senior	low	yes	fair	YES
T6	senior	low	yes	excellent	NO
T7	middle	low	yes	excellent	YES
T8	youth	medium	no	fair	NO
T9	youth	low	yes	fair	YES
T10	senior	medium	yes	fair	YES
T11	youth	medium	yes	excellent	YES
T12	middle	medium	no	excellent	YES
T13	middle	high	yes	fair	YES
T14	senior	medium	no	excellent	NO

Age	Income	Student	Rating	Buy
middle	high	no	fair	YES
middle	low	yes	excellent	YES
middle	medium	no	excellent	YES
middle	high	yes	fair	YES

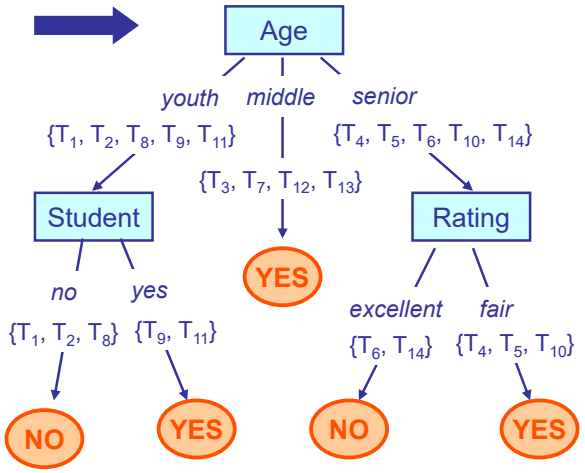
Age	Income	Student	Rating	Buy
senior	medium	no	fair	YES
senior	low	yes	fair	YES
senior	low	yes	excellent	NO
senior	medium	yes	fair	YES
senior	medium	no	excellent	NO

Age	Income	Student	Rating	Buy
youth	high	no	fair	NO
youth	high	no	excellent	NO
youth	medium	no	fair	NO
youth	low	yes	fair	YES
youth	medium	yes	excellent	YES

2.2 Cây quyết định

VD: Tạo các phân hoạch (nhiều cấp)

Tid	Age	Income	Student	Rating	Buy
T1	youth	high	no	fair	NO
T2	youth	high	no	excellent	NO
T3	middle	high	no	fair	YES
T4	senior	medium	no	fair	YES
T5	senior	low	yes	fair	YES
T6	senior	low	yes	excellent	NO
T7	middle	low	yes	excellent	YES
T8	youth	medium	no	fair	NO
T9	youth	low	yes	fair	YES
T10	senior	medium	yes	fair	YES
T11	youth	medium	yes	excellent	YES
T12	middle	medium	no	excellent	YES
T13	middle	high	yes	fair	YES
T14	senior	medium	no	excellent	NO



2.2 Cây quyết định

❑ **Xây dựng cây quyết định từ tập huấn luyện**

- đệ quy từ nút gốc → *top-down*
- thuật toán "tham lam" (*greedy algorithm*), không quay lui
- chia để trị (*divide-and-conquer*): phân hoạch trên các quan sát

Ở mỗi bước, chọn thuộc tính tạo phân hoạch
tốt nhất trên các quan sát liên quan (truyền từ nút cha)

↓

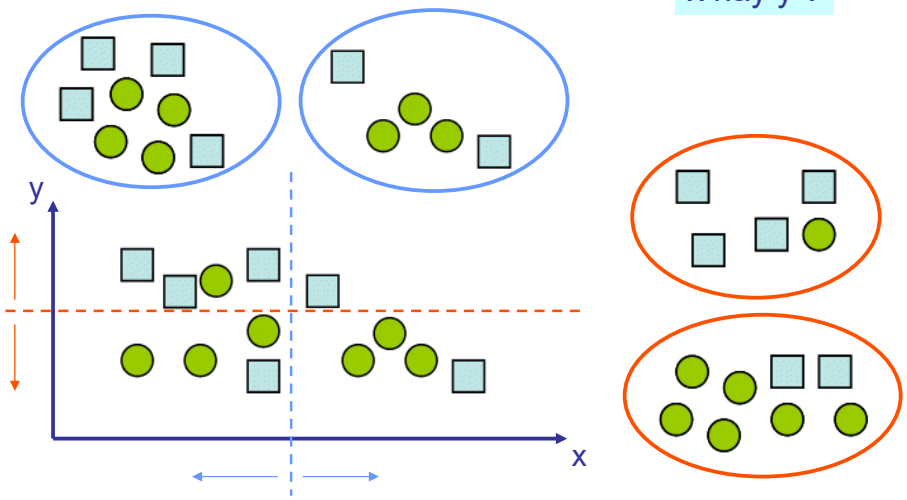
nhiều cách tiếp cận → độ đo

- *điều kiện dừng*: phân hoạch hoàn toàn tất cả quan sát, *hoặc* tất cả các thuộc tính đã được sử dụng (mỗi thuộc tính chỉ được xuất hiện 1 lần trong cây)

2.2 Cây quyết định (tt.)

❑ **Chọn thuộc tính phân tách** (*splitting attribute*)

x hay y ?



biên quyết định (*decision boundary*)

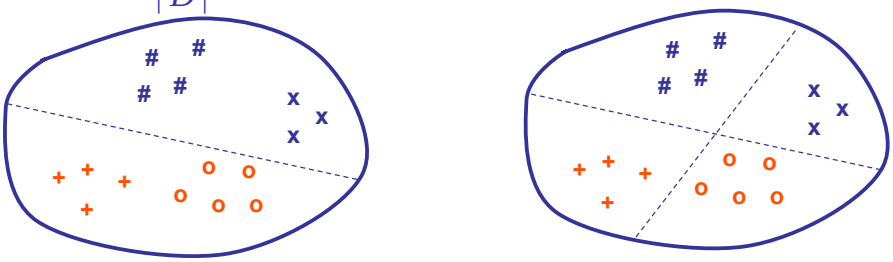
2.2 Cây quyết định

- **Độ đo entropy trong Lý thuyết thông tin** (*Information Theory*, Shannon 1948)**: mức độ hỗn tạp (thuần khiết) của D**

Giả sử phân hoạch D với các lớp C_1, …, C_m.

$$Entropy(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) = Info(D)$$

$p_i = \frac{|C_{i,D}|}{|D|}$: xs để 1 phần tử của D thuộc về lớp C_i (i = 1..m)



2.2 Cây quyết định

<u>VD</u>: Độ đo entropy theo thuộc tính phân lớp (*target attribute*) **Buy**

Tid	Buy
T1	NO
T2	NO
T3	YES
T4	YES
T5	YES
T6	NO
T7	YES
T8	NO
T9	YES
T10	YES
T11	YES
T12	YES
T13	YES
T14	NO

|D| =14

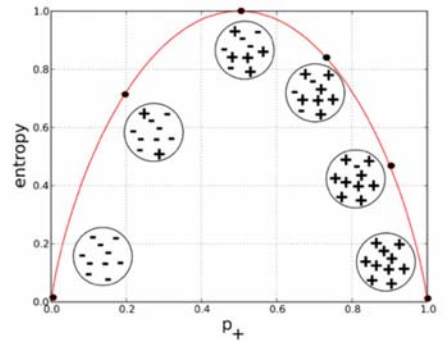
p_{YES} = 9/14

p_{NO} = 5/14

$$Entropy(D) = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right) = 0.940$$

2.2 Cây quyết định

- **Độ đo entropy trong phân lớp nhị phân** (C_+ và C_-)
  - Entropy = 0: $(p_+ * p_-) = 0 \Rightarrow$ S đồng nhất
  - Entropy = 1: $p_+ = p_- = 0.5 \Rightarrow |C_+| = |C_-|$
  - Entropy $\in (0, 1)$: $p_i \in (0, 1) \Rightarrow |C_+| \neq |C_-|$



2.2 Cây quyết định

- **Độ đo *Information Gain*: ước lượng độ sai biệt về thông tin TRƯỚC và SAU khi dùng thuộc tính A để phân hoạch D**

Giả sử DOM(A) = $\{a_1, a_2, \ldots, a_v\}$

Phân hoạch D từ thuộc tính A: $\{D_1^A, D_2^A, \cdots, D_v^A\}$

$$Info(D, A) = \sum_{j=1}^{v} \frac{|D_j^A|}{|D|} * Entropy(D_j^A)$$

Entropy sau khi dùng A để tạo phân hoạch

$$Gain(D, A) = Entropy(D) - Info(D, A)$$

Độ sai biệt về thông tin (trung bình) sau khi dùng A để tạo phân hoạch

càng NHỎ càng tốt

$$A^* = \arg\max_A Gain(D, A)$$

Iterative Dichotomiser 3 – ID.3 [Quinlan, 86]

2.2 Cây quyết định

VD: Độ đo Information Gain

Tid	Age	Income	Student	Rating	Buy
T1	youth	high	no	fair	NO
T2	youth	high	no	excellent	NO
T3	middle	high	no	fair	YES
T4	senior	medium	no	fair	YES
T5	senior	low	yes	fair	YES
T6	senior	low	yes	excellent	NO
T7	middle	low	yes	excellent	YES
T8	youth	medium	no	fair	NO
T9	youth	low	yes	fair	YES
T10	senior	medium	yes	fair	YES
T11	youth	medium	yes	excellent	YES
T12	middle	medium	no	excellent	YES
T13	middle	high	yes	fair	YES
T14	senior	medium	no	excellent	NO

Gain(D, Age) = ?

Gain(D, Income) = ?

Gain(D, Student) = ?

Gain(D, Rating) = ?

2.2 Cây quyết định

VD: Độ đo Information Gain

Thuộc tính Age:

Age	Buy
youth	NO
youth	NO
middle	YES
senior	YES
senior	YES
senior	NO
middle	YES
youth	NO
youth	YES
senior	YES
youth	YES
middle	YES
middle	YES
senior	NO

middle [4/14, YES = 4, NO = 0]

senior [5/14, YES = 3, NO = 2]

youth [5/14, YES = 2, NO = 3]

$$Entropy(middle) = -\frac{4}{4}\log_2\frac{4}{4} = 0$$

$$Entropy(senior) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

$$Entropy(youth) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

$$Info(D, Age) = \frac{4}{14} * 0 + \frac{5}{14} * 0.971 + \frac{5}{14} * 0.971 = 0.694$$

$$Gain(D, Age) = 0.940 - 0.694 = 0.246$$

2.2 Cây quyết định

VD: Độ đo Information Gain

Thuộc tính Income: high [4/14, YES = 2, NO = 2]

low [4/14, YES = 3, NO = 1]

medium [6/14, YES = 4, NO = 2]

Income	Buy
high	NO
high	NO
high	YES
medium	YES
low	YES
low	NO
low	YES
medium	NO
low	YES
medium	YES
medium	YES
medium	YES
high	YES
medium	NO

$$Entropy(high) = -\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4} = 1$$

$$Entropy(low) = -\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4} = 0.811$$

$$Entropy(medium) = -\frac{4}{6}log_2\frac{4}{6} - \frac{2}{6}log_2\frac{2}{6} = 0.918$$

$$Info(D, Income) = \frac{4}{14} * 1 + \frac{4}{14} * 0.811 + \frac{6}{14} * 0.918 = 0.911$$

$$Gain(D, Income) = 0.940 - 0.911 = 0.029$$

2.2 Cây quyết định

VD: Độ đo Information Gain

Thuộc tính Student: no [7/14, YES = 3, NO = 4]

yes [7/14, YES = 6, NO = 1]

Student	Buy
no	NO
no	NO
no	YES
no	YES
yes	YES
yes	NO
yes	YES
no	NO
yes	YES
yes	YES
yes	YES
no	YES
yes	YES
no	NO

$$Entropy(no) = -\frac{3}{7}log_2\frac{3}{7} - \frac{4}{7}log_2\frac{4}{7} = 0.985$$

$$Entropy(yes) = -\frac{6}{7}log_2\frac{6}{7} - \frac{1}{7}log_2\frac{1}{7} = 0.592$$

$$Info(D, Student) = \frac{7}{14} * 0.985 + \frac{7}{14} * 0.592 = 0.789$$

$$Gain(D, Student) = 0.940 - 0.788 = 0.151$$

2.2 Cây quyết định

VD: Độ đo Information Gain

Thuộc tính Rating: excellent [6/14, YES = 3, NO = 3]

fair [8/14, YES = 6, NO = 2]

Rating	Buy
fair	NO
excellent	NO
fair	YES
fair	YES
fair	YES
excellent	NO
excellent	YES
fair	NO
fair	YES
fair	YES
excellent	YES
excellent	YES
fair	YES
excellent	NO

$$Entropy(excellent) = -\frac{3}{6}log_2\frac{3}{6} - \frac{3}{6}log_2\frac{3}{6} = 1$$

$$Entropy(yes) = -\frac{6}{8}log_2\frac{6}{8} - \frac{2}{8}log_2\frac{2}{8} = 0.811$$

$$Info(D, Rating) = \frac{6}{14} * 1 + \frac{8}{14} * 0.811 = 0.892$$

$$Gain(D, Rating) = 0.940 - 0.892 = 0.048$$

2.2 Cây quyết định

VD: Độ đo Information Gain

Age	Income	Student	Rating	Buy
youth	high	no	fair	NO
youth	high	no	excellent	NO
middle	high	no	fair	YES
senior	medium	no	fair	YES
senior	low	yes	fair	YES
senior	low	yes	excellent	NO
middle	low	yes	excellent	YES
youth	medium	no	fair	NO
youth	low	yes	fair	YES
senior	medium	yes	fair	YES
youth	medium	yes	excellent	YES
middle	medium	no	excellent	YES
middle	high	yes	fair	YES
senior	medium	no	excellent	NO

Gain(D, Age) = 0.246

Gain(D, Income) = 0.029

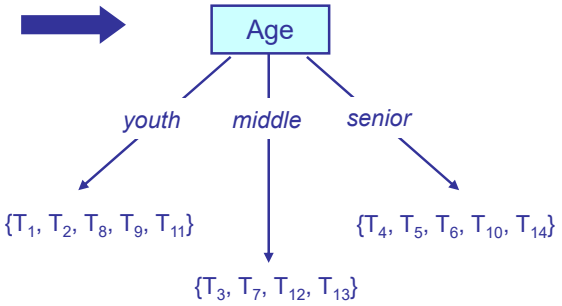
Gain(D, Student) = 0.151

Gain(D, Rating) = 0.048

2.2 Cây quyết định

VD: Tạo các phân hoạch theo thuộc tính Age → đệ quy

Tid	Age	Income	Student	Rating	Buy
T1	youth	high	no	fair	NO
T2	youth	high	no	excellent	NO
T3	middle	high	no	fair	YES
T4	senior	medium	no	fair	YES
T5	senior	low	yes	fair	YES
T6	senior	low	yes	excellent	NO
T7	middle	low	yes	excellent	YES
T8	youth	medium	no	fair	NO
T9	youth	low	yes	fair	YES
T10	senior	medium	yes	fair	YES
T11	youth	medium	yes	excellent	YES
T12	middle	medium	no	excellent	YES
T13	middle	high	yes	fair	YES
T14	senior	medium	no	excellent	NO



2.2 Cây quyết định

- **Trường hợp A là thuộc tính liên tục**

  Sắp xếp DOM(A) = $\{a_1, a_2, \ldots, a_v\}$ tăng dần: $a_i < a_j, \forall i < j$

  Tạo $(v - 1)$ điểm giữa $\{ mp_1, mp_2, \ldots, mp_{v-1} \}$ của các cặp (a_i, a_{i+1})

  Với mỗi mp_i, ta có phân hoạch của D gồm 2 tập con:

  $$D_{left} = \sigma_{\{A \le mpi\}}(D) \text{ và } D_{right} = D - D_{left}$$

  Chọn mp_i sao cho Info(D, A) nhỏ nhất.

2.2 Cây quyết định

- Information Gain có xu hướng "thiên vị" những thuộc tính mang nhiều giá trị

Age	Income	Student	Rating	When	Buy?
youth	high	no	fair	3 pm	NO
youth	high	no	excellent	3 pm	NO
middle	high	no	fair	5 pm	YES
senior	medium	no	fair	4 pm	YES
senior	low	yes	fair	6 pm	YES
senior	low	yes	excellent	7 pm	NO
middle	low	yes	excellent	4 pm	YES
youth	medium	no	fair	5 pm	NO
youth	low	yes	fair	3 pm	YES
senior	medium	yes	fair	3 pm	YES
youth	medium	yes	excellent	6 pm	YES
middle	medium	no	excellent	5 pm	YES
middle	high	yes	fair	6 pm	YES
senior	medium	no	excellent	4 pm	NO

2.2 Cây quyết định

VD: Độ đo Information Gain

Thuộc tính When:

When	Buy
3 pm	NO
3 pm	NO
5 pm	YES
4 pm	YES
6 pm	YES
7 pm	NO
4 pm	YES
5 pm	NO
3 pm	YES
3 pm	YES
6 pm	YES
5 pm	YES
6 pm	YES
4 pm	NO

3pm [4/14, YES = 2, NO = 2]
4pm [3/14, YES = 2, NO = 1]
5pm [3/14, YES = 2, NO = 1]
6pm [3/14, YES = 3, NO = 0]
7pm [1/14, YES = 0, NO = 1]

$$Entropy(3pm) = -\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4} = 1$$

$$Entropy(4pm) = Entropy(5pm) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} == 0.918$$

$$Entropy(6pm) = Entropy(7pm) = 0$$

$$Info(D,When) = \frac{4}{14} * 1 + \frac{3}{14} * 0.918 + \frac{3}{14} * 0.918 = 0.679$$

$$Gain(D,When) = 0.940 - 0.679 = 0.261$$

2.2 Cây quyết định

VD: Độ đo Information Gain

Age	Income	Student	Rating	When	Buy?
youth	high	no	fair	3 pm	NO
youth	high	no	excellent	3 pm	NO
middle	high	no	fair	5 pm	YES
senior	medium	no	fair	4 pm	YES
senior	low	yes	fair	6 pm	YES
senior	low	yes	excellent	7 pm	NO
middle	low	yes	excellent	4 pm	YES
youth	medium	no	fair	5 pm	NO
youth	low	yes	fair	3 pm	YES
senior	medium	yes	fair	3 pm	YES
youth	medium	yes	excellent	6 pm	YES
middle	medium	no	excellent	5 pm	YES
middle	high	yes	fair	6 pm	YES
senior	medium	no	excellent	4 pm	NO

Gain(D, Age) = 0.246

Gain(D, Student) = 0.151

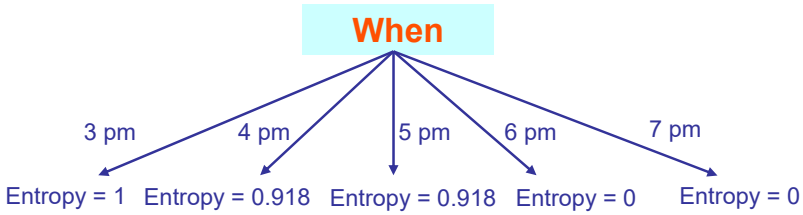
Gain(D, Rating) = 0.048

Gain(D, Income) = 0.029

Gain(D, When) = 0.261

2.2 Cây quyết định

- **Information Gain có xu hướng "thiên vị" những thuộc tính mang nhiều giá trị**



Gain(D, When) = 0.261

Gain(D, Age) = 0.246

Gain(D, Student) = 0.151

Gain(D, Rating) = 0.048

Gain(D, Income) = 0.029

(When = 7pm) ⇒ (Buy = NO)

Confidence ↑

Support ↓

2.2 Cây quyết định

- **Độ đo *Gain Ratio*: chuẩn hoá Information Gain bằng thông tin phân tách (*split information*)**

$$SplitInfo(D,A) = -\sum_{j=1}^{v} \frac{|D_j^A|}{|D|} * \log_2\left(\frac{|D_j^A|}{|D|}\right)$$

$$GainRatio(D,A) = \frac{Gain(D,A)}{SplitInfo(D,A)}$$

C4.5

$$A^* = \arg\max_A GainRatio(D,A)$$ [Quinlan, 93]

2.2 Cây quyết định

VD: Độ đo Gain Ratio

Thuộc tính Age: middle [4/14, YES = 4, NO = 0]

senior [5/14, YES = 3, NO = 2]

youth [5/14, YES = 2, NO = 3]

$$SplitInfo(D,Age) = -\frac{4}{14}\log_2\frac{4}{14} - \frac{5}{14}\log_2\frac{5}{14} - \frac{5}{14}\log_2\frac{5}{14} = 1.577$$

$$GainRatio(D,Age) = \frac{0.246}{1.577} = 0.156$$

2.2 Cây quyết định

VD: Độ đo Gain Ratio

Thuộc tính When: 3pm [4/14, YES = 2, NO = 2]

4pm [3/14, YES = 2, NO = 1]

5pm [3/14, YES = 2, NO = 1]

6pm [3/14, YES = 3, NO = 0]

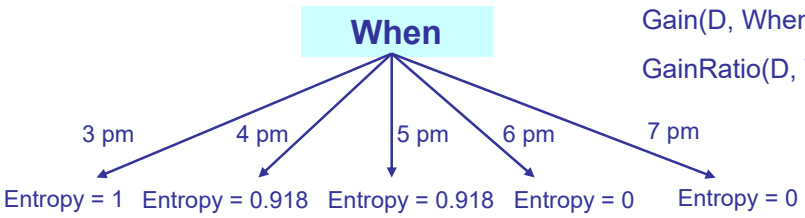
7pm [1/14, YES = 0, NO = 1]

$$SplitInfo(D,When) = -\frac{4}{14}\log_2\frac{4}{14} - 3\left(\frac{3}{14}\log_2\frac{3}{14}\right) - \frac{1}{14}\log_2\frac{1}{14} = 2.217$$

$$GainRatio(D,When) = \frac{0.261}{2.217} = 0.118$$

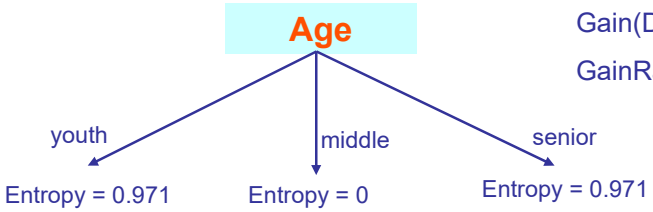
2.2 Cây quyết định

VD:



Gain(D, When) = 0.261

GainRatio(D, When) = 0.118



Gain(D, Age) = 0.246

GainRatio(D, Age) = 0.156

2.2 Cây quyết định

$$SplitInfo(D,A) = -\sum_{j=1}^{v} \frac{|D_j^A|}{|D|} * \log_2\left(\frac{|D_j^A|}{|D|}\right) \quad GainRatio(D,A) = \frac{Gain(D,A)}{SplitInfo(D,A)}$$

SplitInfo $\rightarrow$ 0 ($|D^A_j| \approx |D|$) ?

2.2 Cây quyết định

B1. S = { những tuples chưa được phân hoạch (nút hiện hành) }

A = { những thuộc tính chưa tham gia vào cây quyết định }

B2. Nếu những tuples trong S thuộc cùng 1 class thì S $\rightarrow$ nút lá.

Ngược lại:

+ Chọn A_j "tốt nhất" trong A để tạo phân hoạch cho S.
  Loại A_j khỏi A.

+ Tạo các nhánh xuất phát từ A_j:

  Nếu A_j rời rạc: tạo phân hoạch { D_{jk} | $a_{jk} \in DOM(A_j)$ }

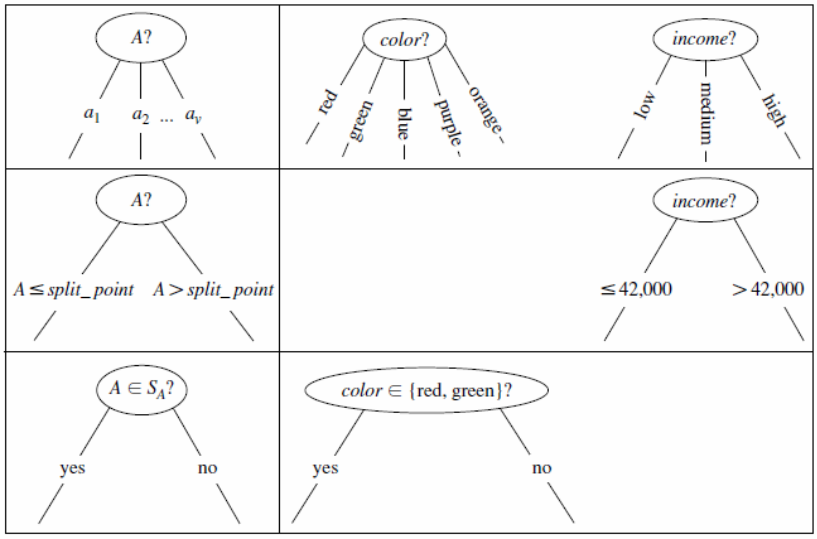
  Nếu A_j liên tục: tạo $D_{left} = \sigma_{\{Aj \le split\_point\}}(D)$ và $D_{right} = D - D_{left}$

  Nếu ($A_j \in V$) ?: tạo D_{Yes} và $D_{No} = D - D_{Yes}$

B3. Đệ quy B1 với các D_{jk} được tạo từ phân hoạch bởi A_j

2.2 Cây quyết định

VD: Tạo các nhánh từ một thuộc tính A



[Han+]

2.2 Cây quyết định

- **Kiểm tra thuật toán thỏa mãn 1 trong các điều kiện dừng:**

  (i) Toàn bộ các tuples trong S đều thuộc cùng 1 class.

  (ii) Không còn A_j nào để tạo phân hoạch cho S trong B2.
  → chọn lớp phổ biến (mặc định)

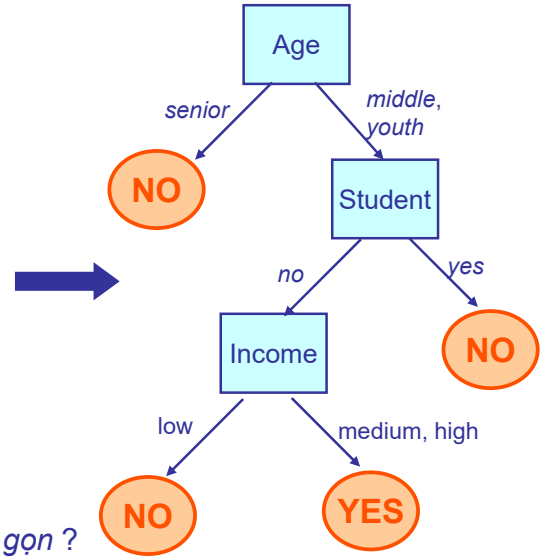
  (iii) Nhánh rỗng, nghĩa là $D_{jk} = \varnothing$ → chọn lớp phổ biến (mặc định)

2.2 Cây quyết định

❑ **Xây dựng cây quyết định từ tập huấn luyện**

- có thể tạo nhiều cây quyết định khác nhau từ 1 tập huấn luyện

Age	Income	Student	Buy
youth	high	no	NO
youth	high	no	NO
middle	high	no	YES
senior	medium	no	YES
senior	low	yes	YES
senior	low	yes	NO
middle	low	yes	YES
youth	medium	no	NO
youth	medium	yes	YES
senior	medium	yes	YES



Tiêu chí chất lượng ?

Nguyên tắc Ockham's Razor: *tinh gọn* ?

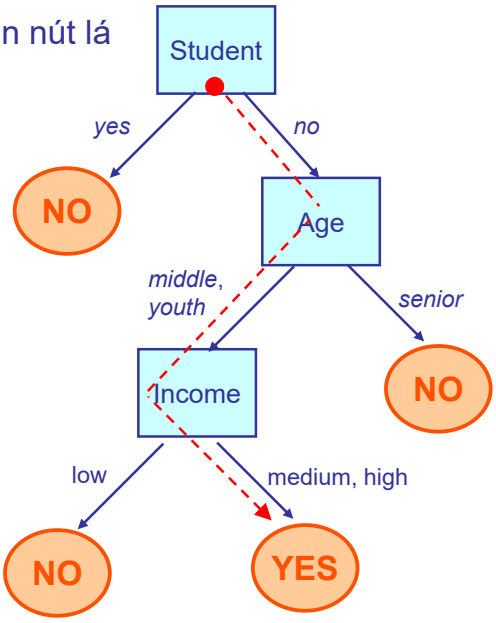
2.2 Cây quyết định

❑ **Phân lớp dựa trên cây quyết định**

- bắt đầu từ nút gốc đi dần đến nút lá

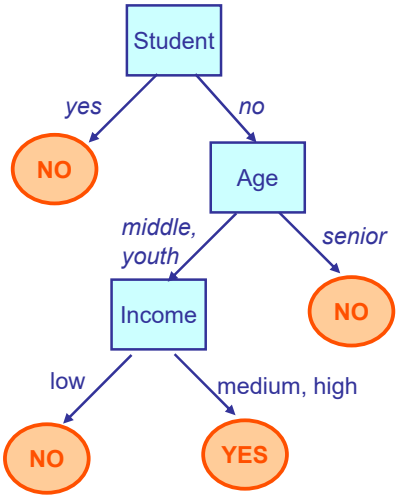
(middle, high, no, **?**)

YES



2.2 Cây quyết định

- **Biến đổi cây quyết định thành tập luật IF-THEN**
  - mỗi lộ trình → 1 luật cơ bản (*rule base*) IF-THEN
  - *rule support*: % dữ liệu (huấn luyện) hỗ trợ cho luật



R_1: IF (Student = yes)
THEN Buy = NO

R_2: IF (Student = no) AND (Age = senior)
THEN Buy = NO

R_3: IF (Student = no) AND (Age ≠ senior)
AND (Income = low)
THEN Buy = NO

R_4: IF (Student = no) AND (Age ≠ senior)
AND (Income ≠ low)
THEN Buy = YES

2.2 Cây quyết định

- **Chỉ số Gini** (*Gini Index*)

$$Gini = 1 - \sum_{i=1}^{k} p_i^2$$

p_i: xác suất để quan sát thuộc lớp (nhãn) C_i

- Gini càng thấp thì mức độ đồng nhất càng cao
- hiệu quả khi có số lượng các lớp khá lớn
  (tính toán nhanh hơn entropy)

2.2 Cây quyết định

- **Chỉ số Gini** (*Gini Index*)

Dễ thấy:

$$1=\left(\sum_{i=1}^{k} p_i\right)^2 \geq \sum_{i=1}^{k} p_i^2 \quad \Rightarrow \quad 0 \leq \left(1-\sum_{i=1}^{k} p_i^2\right)$$

Hơn nữa, theo BĐT Cauchy-Schwarz:

$$\left(\sum_{i=1}^{k} a_i b_i\right)^2 \leq \left(\sum_{i=1}^{k} a_i^2\right)\left(\sum_{i=1}^{k} b_i^2\right)$$

Với $a_i = p_i$ và $b_i = 1$:

$$1=\left(\sum_{i=1}^{k} p_i\right)^2 \leq k\sum_{i=1}^{k} p_i^2 \quad \Rightarrow \quad \frac{1}{k} \leq \sum_{i=1}^{k} p_i^2$$

Vậy: $0 \leq Gini \leq \left(1-\frac{1}{k}\right)$

2.2 Cây quyết định

- **Ưu điểm**
  - dễ hiểu, dễ diễn giải kết quả
  - có thể áp dụng cho nhiều kiểu dữ liệu khác nhau
  - không cần chuẩn hóa dữ liệu
  - không bị tác động bởi vấn đề dữ liệu bị thiếu
  - phân lớp nhanh

2.2 Cây quyết định

- **Khuyết điểm**
  - chi phí (thời gian) xây dựng mô hình: $O(n * |D| * \log_2|D|)$
  - kém hiệu quả với dữ liệu định lượng
  - kém ổn định: sự thay đổi nhỏ trên tập huấn luyện cũng có thể dẫn đến những thay đổi lớn trên cấu trúc cây quyết định

2.2 Cây quyết định

- **Một số mở rộng**
  - Cắt tỉa (*pruning*)
  - Cây quyết định đa biến (*Multivariate Decision Tree*)
  - CART (*Classification And Regression Tree*)

2.2 Cây quyết định

- **Rừng ngẫu nhiên** (*Random Forest*): **số lượng features lớn**
  - nếu KHÔNG giới hạn độ sâu của cây quyết định → tồn tại những nút lá (nhãn) chỉ liên quan đến 1 số lượng nhỏ quan sát
  - nếu giới hạn độ sâu của cây quyết định → có thể bỏ sót những điều kiện kiểm tra (phân nhánh) quan trọng

  ⇒ học kết hợp (*Ensemble Learning*): từ nhiều cây quyết định

2.2 Cây quyết định

- **Rừng ngẫu nhiên** (*Random Forest*)**: tạo nhiều cây quyết định**
  - dựa trên các tập con của tập huấn luyện: chọn ngẫu nhiên
  - dựa trên các tập con features: chọn ngẫu nhiên, theo ngữ cảnh (ý nghĩa, mức độ quan trọng)

2.2 Cây quyết định

- **Rừng ngẫu nhiên** (*Random Forest*)**: các cơ chế kết hợp**
  - bài toán phân lớp
  - bài toán hồi quy

2.3 Phân lớp Naïve Bayes

- **Định lý Bayes**

Tuple (*evidence*) $X = (x_1, x_2, \ldots, x_n)$, $x_j \in DOM(A_j)$

VD: X = (35 tuổi, thu nhập $40K)

Giả thuyết H: $X \in C_k$

VD: C_{YES} = mua laptop (Buy = YES)

Thomas Bayes
(1701-1761)

Xác suất hậu nghiệm (*posterior probability*)

VD: P(H | X): xs SẼ mua laptop nếu là 35 tuổi và thu nhập 40K.

P(X | H): xs để người ĐÃ mua laptop là 35 tuổi và thu nhập 40K.

Xác suất tiên nghiệm (*prior probability*)

VD: P(H): xs sẽ mua laptop, bất kể tuổi tác, thu nhập

P(X): xs để 1 người là 35 tuổi, thu nhập 40K, dù mua hay không

2.3 Phân lớp Naïve Bayes

- **Định lý Bayes**

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

Phân lớp $X \in C_k$ nếu và chỉ nếu: $P(C_k \mid X) \geq P(C_i \mid X)\ \forall i$

$P(C_k \mid X)$: *maximum posterior hypothesis*

$$P(C_k \mid X) = \frac{P(X \mid C_k)P(C_k)}{P(X)}$$ → Hằng số $\forall C_i \in C$

$$\forall i, \quad P(C_i) = \frac{|D|}{|C|} \text{ hoặc } P(C_i) = \frac{|\sigma_{Ci}(D)|}{|D|}$$

$|\sigma_{Ci}(D)|$: Số tuples thuộc C_i

2.3 Phân lớp Naïve Bayes

- **Tính $P(X|C_i)$ với giả định "ngây thơ" về sự độc lập của giá trị giữa các thuộc tính (*class-conditional independence*) đối với C_i**

$$P(X \mid C_i) = \prod_{j=1}^{n} P(x_j \mid C_i)$$

- Nếu A_j <u>rời rạc</u>: $P(x_j \mid C_i) = \frac{|\sigma_{Ci,A_j=x_j}(D)|}{|\sigma_{C_i}(D)|}$
- Nếu A_j liên tục: $P(x_j \mid C_i)$ tuân theo 1 phân phối → PDF

⇒ **Kiểm định tính độc lập giữa các thuộc tính ?**

2.3 Phân lớp Naïve Bayes

VD: Phân lớp (dự đoán) với X = (youth, medium, yes, fair, ?)

Age	Income	Student	Rating	Buy
youth	high	no	fair	NO
youth	high	no	excellent	NO
middle	high	no	fair	YES
senior	medium	no	fair	YES
senior	low	yes	fair	YES
senior	low	yes	excellent	NO
middle	low	yes	excellent	YES
youth	medium	no	fair	NO
youth	low	yes	fair	YES
senior	medium	yes	fair	YES
youth	medium	yes	excellent	YES
middle	medium	no	excellent	YES
middle	high	yes	fair	YES
senior	medium	no	excellent	NO

$P(C_{YES}) = 9/14 = 0.643$

$P(C_{NO}) = 5/14 = 0.357$

$P(Age=youth|C_{YES}) = 2/9 = 0.222$

$P(Age=youth|C_{NO}) = 3/5 = 0.6$

$P(Income=medium|C_{YES}) = 4/9 = 0.444$

$P(Income=medium|C_{NO}) = 2/5 = 0.4$

$P(Student=yes|C_{YES}) = 6/9 = 0.667$

$P(Student=yes|C_{NO}) = 1/5 = 0.2$

$P(Rating=fair|C_{YES}) = 6/9 = 0.667$

$P(Rating=fair|C_{NO}) = 2/5 = 0.4$

2.3 Phân lớp Naïve Bayes

VD: Phân lớp (dự đoán) với X = (youth, medium, yes, fair, ?)

$P(Age=youth|C_{YES}) = 0.222$
$P(Income=medium|C_{YES}) = 0.444$
$P(Student=yes|C_{YES}) = 0.667$
$P(Rating=fair|C_{YES}) = 0.667$

$P(Age=youth|C_{NO}) = 0.6$
$P(Income=medium|C_{NO}) = 0.4$
$P(Student=yes|C_{NO}) = 0.2$
$P(Rating=fair|C_{NO}) = 0.4$

$P(X \mid C_{YES}) = P(Age=youth|C_{YES}) * P(Income=medium|C_{YES}) * P(Student=yes|C_{YES}) * P(Rating=fair|C_{YES}) =$
$= 0.222 * 0.444 * 0.667 * 0.667 = 0.044$

$P(C_{YES}) = 0.643$
$P(C_{NO}) = 0.357$

$\Rightarrow P(X \mid C_{YES}) * P(C_{YES}) = 0.044 * 0.643 = \mathbf{0.028}$

$P(X \mid C_{NO}) = P(Age=youth|C_{NO}) * P(Income=medium|C_{NO}) * P(Student=yes|C_{NO}) * P(Rating=fair|C_{NO}) =$
$= 0.6 * 0.4 * 0.2 * 0. = 0.019$

$\Rightarrow P(X \mid C_{NO}) * P(C_{NO}) = 0.019 * 0.357 = \mathbf{0.007}$

2.3 Phân lớp Naïve Bayes

- **Tính $P(X|C_i)$ với giả định "ngây thơ" về sự độc lập của giá trị giữa các thuộc tính** (*class-conditional independence*) **đối với C_i**

$$P(X \mid C_i) = \prod_{j=1}^{n} P(x_j \mid C_i)$$

Nhận xét: $P(x_j \mid C_i) = 0 \Rightarrow P(X \mid C_i) = 0$

Phép hiệu chỉnh Laplace (*Laplace correction*, *Laplace smoothing*)

$$P(x_j \mid C_i) = \frac{|\sigma_{Ci,A_j=x_j}(D)| + \alpha}{|\sigma_{C_i}(D)| + (\alpha * \beta)}$$

α: tham số hiệu chỉnh (thường = 1)

$\beta = |\ DOM(A_j)\ |$ (nhiều cách khác)

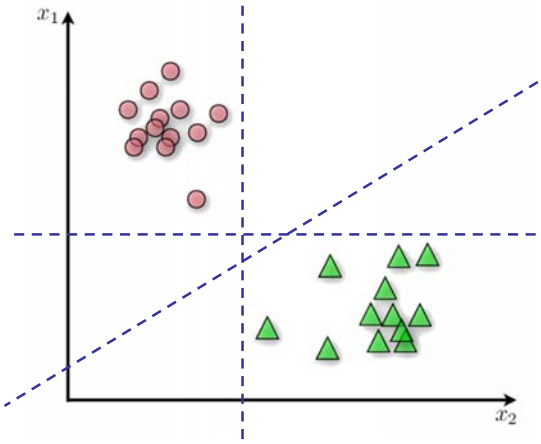
2.3 Phân lớp Naïve Bayes

- **Giả thuyết về phân phối của các thuộc tính**
  - Gaussian
  - Bernoulli
  - Multinomial

2.4 Support Vector Machine (*SVM*)

❑ **Phân lớp SVM tuyến tính** (*Linear SVM Classification*)

- xây dựng siêu phẳng (*hyperplane*) có thể phân cách các lớp

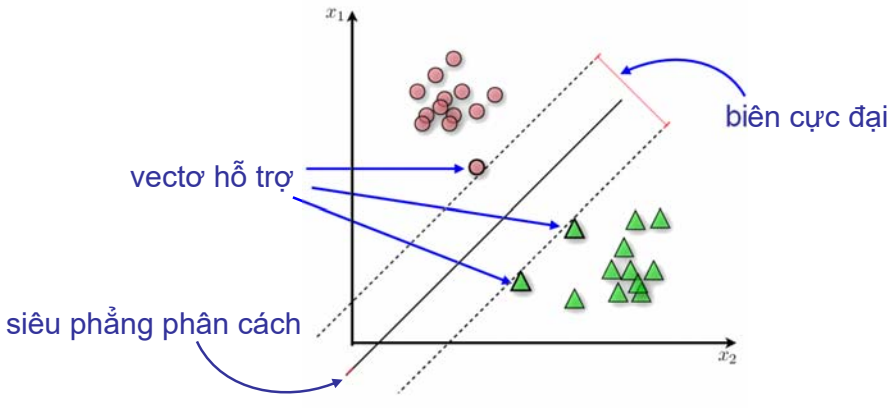


linearly separable

2.4 SVM

❑ **Phân lớp SVM tuyến tính**

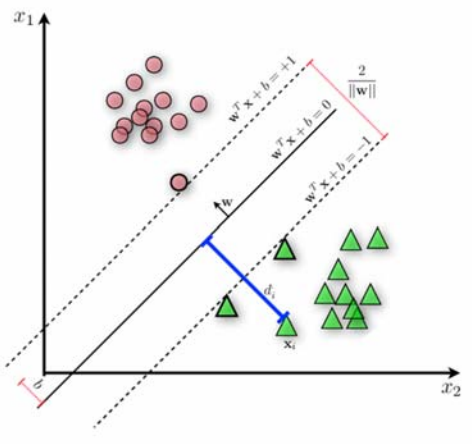
- bài toán tối ưu: xác định biên cực đại (*maximum margin classification*)
- bài toán "đối ngẫu": tìm các vectơ hỗ trợ (*support vectors*) → siêu phẳng phân cách



2.4 SVM

❑ Phân lớp SVM tuyến tính

- biên (*margin*): khoảng cách gần nhất từ siêu phẳng phân cách đến 1 điểm dữ liệu của mỗi lớp xấp xỉ bằng nhau và cực đại



2.4 SVM

❑ Phân lớp SVM tuyến tính

Training set: $T = \{(x^{(i)}, y_i)\}_{i=1}^{m} \quad x^{(i)} \in \mathbb{R}^d \quad y_i \in \{-1, +1\}$

Siêu phẳng H: $w^T x + b = 0 \longrightarrow$ tích vô hướng $w^T x = \langle w, x \rangle$

Khoảng cách từ $(x^{(i)}, y_i)$ đến siêu phẳng H: $d_i = \dfrac{y_i (w^T x^{(i)} + b)}{\|w\|}$

Biên: $$\text{margin} = \min_i \frac{y_i (w^T x^{(i)} + b)}{\|w\|}$$

Bài toán tối ưu, tìm (w, b):

$$(w, b) = \arg\max_{w,b} \left\{ \min_i \frac{y_i (w^T x^{(i)} + b)}{\|w\|} \right\} = \arg\max_{w,b} \left\{ \frac{1}{\|w\|} \min_i (y_i (w^T x^{(i)} + b)) \right\}$$

2.4 SVM

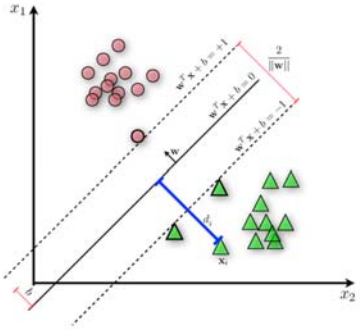
❑ Phân lớp SVM tuyến tính

- thay thế w bằng k.w và b = k.b, với k > 0, thì margin không đổi
  ⇒ có thể giả sử khoảng cách từ H đến những điểm gần nhất:

$$y_i(w^T x^{(i)} + b) = 1 \quad d_i = \frac{y_i(w^T x^{(i)} + b)}{\|w\|} = \frac{1}{\|w\|} \quad \text{max\_margin} = \frac{2}{\|w\|}$$

- biến đổi bài toán tối ưu:

$$(w,b) = \arg\max_{w,b} \left\{ \frac{1}{\|w\|} \right\} \quad y_i(w^T x^{(i)} + b) \geq 1$$



hay:

$$(w,b) = \arg\min_{w,b} \|w\|^2 \quad 1 - y_i(w^T x^{(i)} + b) \leq 0$$

khả vi

2.4 SVM

❑ Phân lớp SVM tuyến tính

- quy hoạch toàn phương (*quadratic programming*)

$$(w,b) = \arg\min_{w,b} \|w\|^2 \quad 1 - y_i(w^T x^{(i)} + b) \leq 0$$

  – hàm mục tiêu là 1 chuẩn (L_2): hàm lồi chặt (*strictly convex funct.*)

  – các bất đẳng thức ràng buộc là tuyến tính ⇒ hàm lồi

  ⇒ nghiệm duy nhất

- phân lớp (dự đoán) dữ liệu mới:

$$\text{class}(x) = sign(w^T x + b)$$

2.4 SVM

- **Phân lớp SVM tuyến tính**
  - bổ sung thêm những quan sát bên ngoài phạm vi 2 đường biên không ảnh hưởng đến mô hình phân lớp
  - mô hình phân lớp được đặc trưng bởi các support vectors
  - SVM là *parametric* hay *nonparametric* ? [Alpaydin, Russell+]

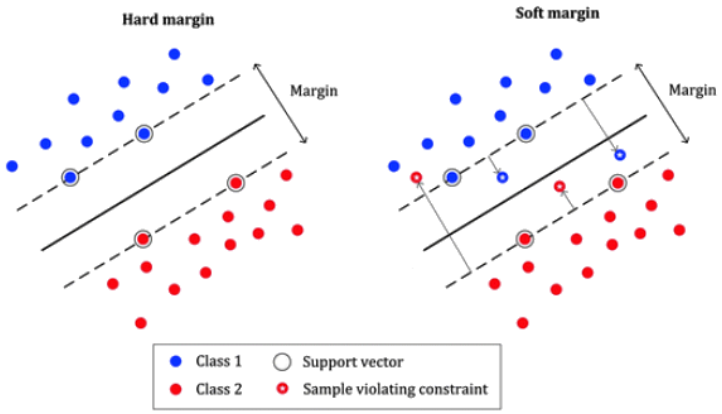
2.4 SVM

- **Một số phương pháp SVM cải biên**
  - *Soft Margin SVM*: xử lý dữ liệu phân tách hầu như tuyến tính (*almost linear separability*)
  - *Kernel SVM*: xử lý dữ liệu phân tách phi tuyến (*non-linear separability*)
  - *Multi-class SVM*: bài toán đa lớp

2.4 SVM

❑ Phân lớp SVM tuyến tính

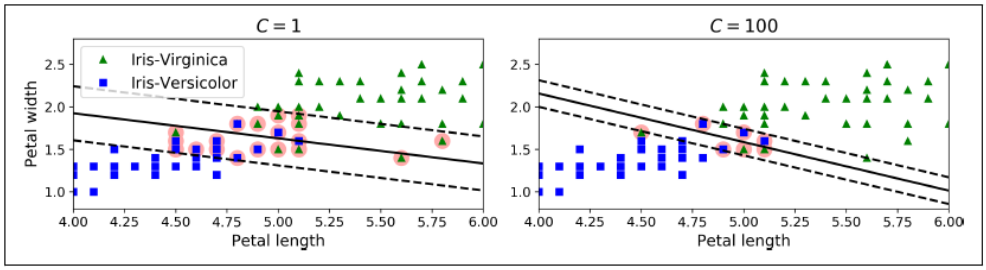
- *hard margin*: phân tách tuyến tính → nhạy cảm với outliers
- *soft margin*: cân đối giữa độ rộng của biên cực đại và giới hạn số lượng quan sát đã vi phạm đường biên (*margin violation*)



2.4 SVM

❑ Thuật toán *Soft Margin SVM*

- sử dụng siêu tham số (*hyperparameter*) để hiệu chỉnh độ rộng của biên cực đại → tăng/giảm số quan sát vi phạm đường biên
- độ rộng đường biên cực đại ↗ thì số lượng quan sát vi phạm ↗



[Géron]

2.4 SVM

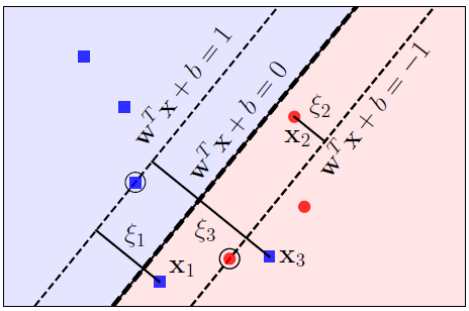
❑ **Thuật toán** *Soft Margin SVM*

- độ "mất mát" (*slack variable*): $\xi_i = \left| w^T x^{(i)} + b - y_i \right|$

  $\xi_i = 0$: x_i được phân cách đúng

  $0 < \xi_i \le 1$: x_i không an toàn nhưng chưa lấn sang lớp sai (x_2)

  $\xi_i > 1$: x_i đã lấn sang lớp sai (x_1 và x_3)



[Vũ Hữu Tiệp]

2.4 SVM

❑ **Thuật toán** *Soft Margin SVM*

- tối ưu hóa hàm mục tiêu với các ràng buộc "mềm":

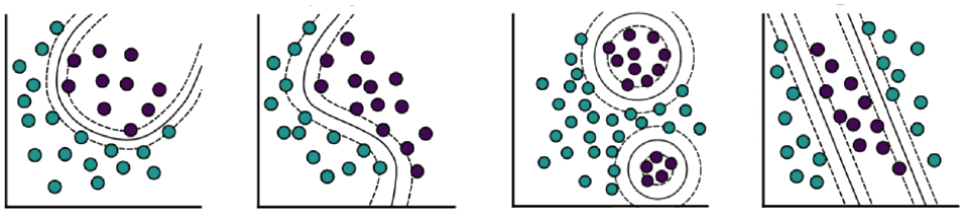
$$(w, b, \xi) = \arg\min_{w,b,\xi} \|w\|^2 + C \sum_{i=1}^{N} \xi_i \quad (\text{const } C > 0)$$

$$1 - \xi_i - y_i (w^T x^{(i)} + b) \le 0$$

2.4 SVM

- **Phân lớp SVM phi tuyến** (*Nonlinear SVM Classification*)
  - dữ liệu không thể phân cách tuyến tính (*non-linear separability*)

  → tạo siêu phẳng phân cách phi tuyến

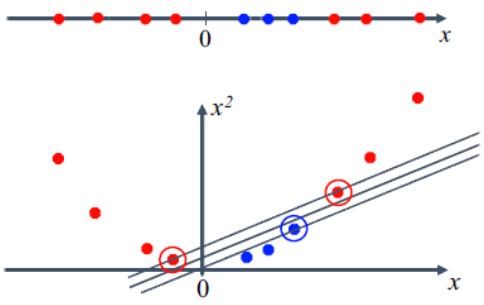


[Hefin I. Rhys]

2.4 SVM

- **Phân lớp SVM phi tuyến** (*Nonlinear SVM Classification*)
  - giải pháp bổ sung features → phân cách tuyến tính

  VD: Bổ sung $x_2 = (x_1)^2$

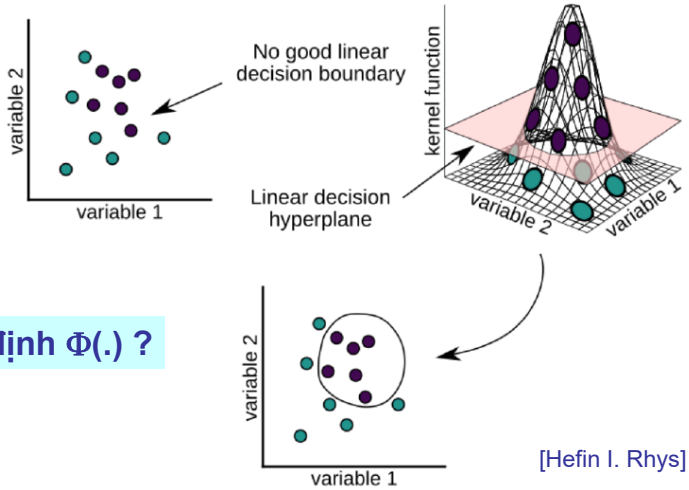


Bậc đa thức NHỎ: kém hiệu quả với những dữ liệu phức tạp

Bậc đa thức LỚN: mô hình chậm vì số lượng lớn features

2.4 SVM

- **Phương pháp** *kernel trick* [Aizerman+, 1964]
  - ánh xạ Φ(.) các quan sát x vào không gian có số chiều cao hơn (*higher-dimensional feature space*) → phân cách tuyến tính



Xác định Φ(.) ?

[Hefin I. Rhys]

2.4 SVM

- **Hàm kernel** (*kernel function*)

$$k(x_i, x_j) = \Phi(x_i)^T . \Phi(x_j)$$

  - ước lượng độ tương đồng giữa x_i và x_j trong không gian mới (thay vì tính tọa độ Φ(x) của từng x trong không gian mới)
  - hàm đối xứng và xác định dương (*positive definite*): điều kiện định lý Mercer nhằm bảo đảm tính lồi của hàm mục tiêu trong bài toán đối ngẫu

[*https://en.wikipedia.org/wiki/Kernel_method*]

2.4 SVM

- **Hàm** *Polynomial kernel*

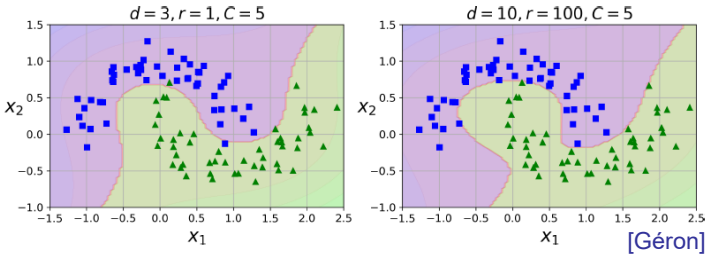
$$k_{Polynomial}(x_i, x_j) = (r.x_i^T.x_j + c)^d$$

Các *hyperparameters*:

d: bậc của đa thức

r, c: hằng số ≥ 0

Đặc biệt *Linear kernel*: $k_{Linear}(x_i, x_j) = x_i^T.x_j$



[Géron]

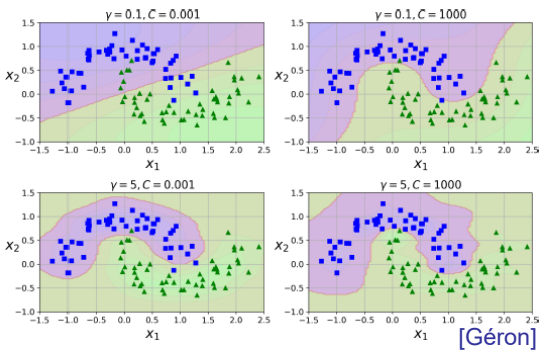
2.4 SVM

- **Hàm** *Radial Basic Function – RBF kernel*

$$k_{RBF}(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right)$$

γ: hằng số > 0

Đặc biệt *Gaussian kernel*: $k_{Gaussian}(x_i, x_j) = \exp\left(\frac{-1}{2\sigma^2}\left\| x_i - x_j \right\|^2\right)$



[Géron]

2.4 SVM

- **Ưu điểm**
  - phân lớp nhanh, tiết kiệm bộ nhớ
  - độ chính xác cao, ít bị overfitting
  - xử lý dữ liệu hiệu quả trong không gian nhiều chiều
  - xử lý cả dữ liệu được phân tách tuyến tính lẫn phi tuyến

2.4 SVM

- **Khuyết điểm**
  - kém hiệu quả với kho dữ liệu lớn (thời gian huấn luyện)
  - kém hiệu quả nếu số chiều lớn hơn số mẫu dữ liệu huấn luyện
  - nhạy cảm với nhiễu
  - thiếu thông tin xác suất phân lớp

Tài liệu tham khảo

Alpaydin, *Introduction to Machine Learning*, 4rd Edition, 2020.

Géron, *Hands-on ML with Scikit-Learn, Keras and TensorFlow*, 2nd Edition, 2019.

Mitchell, *Machine Learning*, 1st Edition, 1997.

Russell and Norvig, *Artificial Intelligence: A Modern Approach*, 4th Edition, 2020.

Vũ Hữu Tiệp, *Machine Learning cơ bản*, 2018.

Thảo luận