BÀI GIẢNG NHẬP MÔN KHAI PHÁ DỮ LIỆU

CHƯƠNG 5. PHÂN LỚP

TS. Trần Mai Vũ
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
ĐẠI HỌC QUỐC GIA HÀ NỘI

Charu C. Aggarwal. *Data Classification: Algorithms*. CRC Press, 2014.

Nội dung

Giới thiệu phân lớp

Phân lớp học giám sát

Phân lớp học bán giám sát

Bài toán phân lớp

- Đầu vào
  - Tập dữ liệu D = $\{d_i\}$
  - Tập các lớp $C_1, C_2, \ldots, C_k$ mỗi dữ liệu d thuộc một lớp C_i
  - Tập ví dụ $D_{exam} = D_1 + D_2 + \ldots + D_k$ với $D_i = \{d\ D_{exam}: d \text{ thuộc } C_i\}$
  - Tập ví dụ D_{exam} đại diện cho tập D
  - D gồm *m* dữ liệu d_i thuộc không gian *n* chiều
- Đầu ra
  - Mô hình phân lớp: ánh xạ từ D sang C
- Sử dụng mô hình
  - d D \ D_{exam} : xác định lớp của đối tượng d

Phân lớp: Quá trình hai pha

- Xây dựng mô hình: Tìm mô tả cho tập lớp đã có
  - Cho trước tập lớp C = {C_1, C_2, …, C_k}
  - Cho ánh xạ (chưa biết) từ miền D sang tập lớp C
  - Có tập ví dụ $D_{exam}=D_1+D_2+ …+ D_k$ với D_i={d D_{exam}: d C_i}
    D_{exam} được gọi là tập ví dụ mẫu.
  - Xây dựng ánh xạ (mô hình) phân lớp trên: Dạy bộ phân lớp.
  - **Mô hình**: Luật phân lớp, cây quyết định, công thức toán học…
- Pha 1: Dạy bộ phân lớp
  - Tách D_{exam} thành D_{train} (2/3) + D_{test} (1/3). D_{train} và D_{test} "tính đại diện" cho miền ứng dụng
  - D_{train} : xây dựng mô hình phân lớp (xác định tham số mô hình)
  - D_{test} : đánh giá mô hình phân lớp (các độ đo hiệu quả)
  - Chọn mô hình có chất lượng nhất
- Pha 2: Sử dụng mô hình (bộ phân lớp)
  - d D \ D_{exam} : xác định lớp của d.

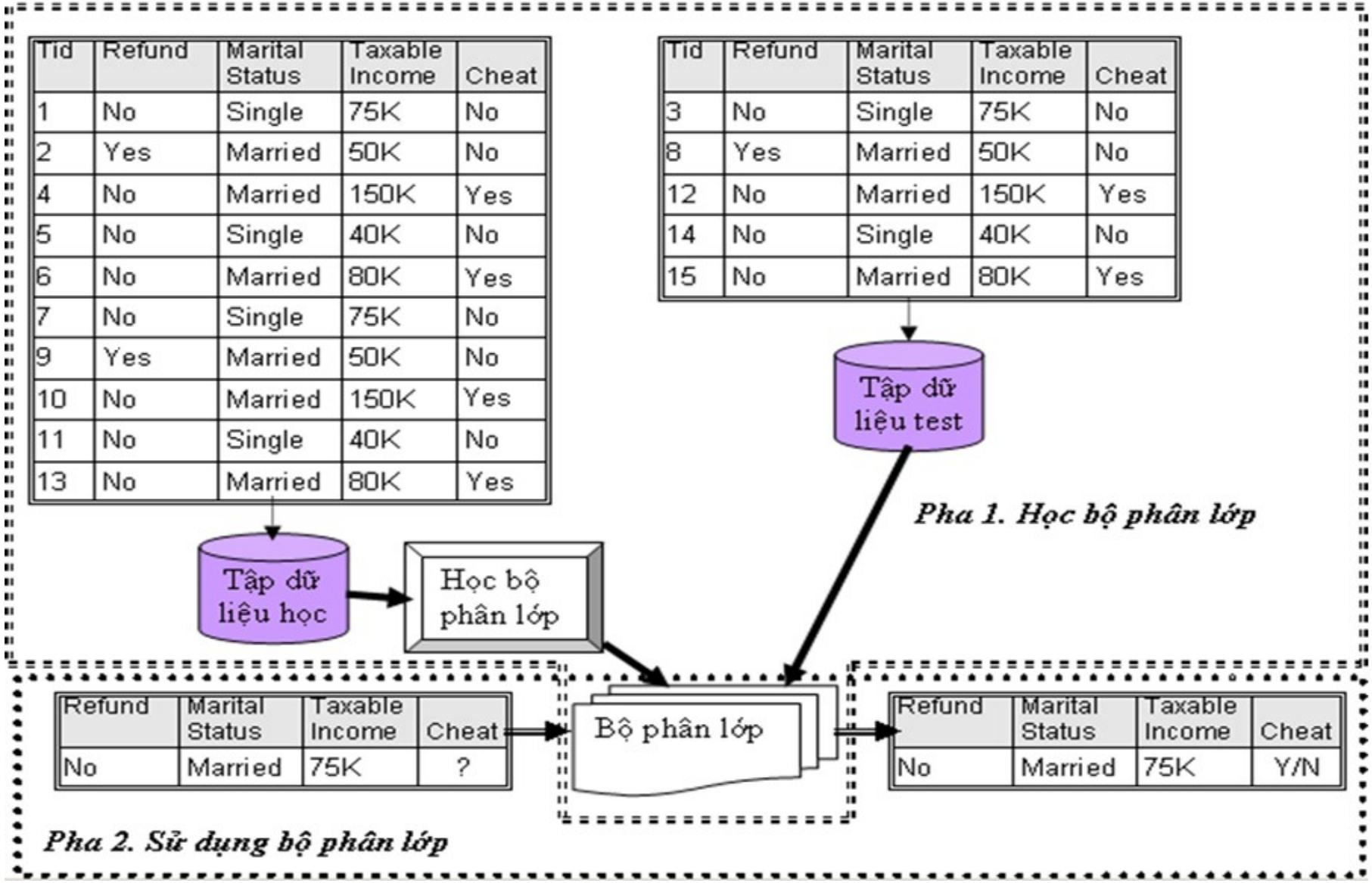
Ví dụ phân lớp: Bài toán cho vay

Tid	*Refund*	*Marital Status*	*Taxable Income*	**Cheat**
1	No	Single	75K	**No**
2	Yes	Married	50K	**No**
3	No	Single	75K	**No**
4	No	Married	150K	**Yes**
5	No	Single	40K	**No**
6	No	Married	80K	**Yes**
7	No	Single	75K	**No**
8	Yes	Married	50K	**No**
9	Yes	Married	50K	**No**
10	No	Married	150K	**Yes**
11	No	Single	40K	**No**
12	No	Married	150K	**Yes**
13	No	Married	80K	**Yes**
14	No	Single	40K	**No**
15	No	Married	80K	**Yes**

Ngân hàng cần cho vay: trả nợ, hôn nhân, thu nhập

“Lớp” liên quan tới **cheat (gian lận)**: hai lớp YES/NO

Phân lớp: Quá trình hai pha



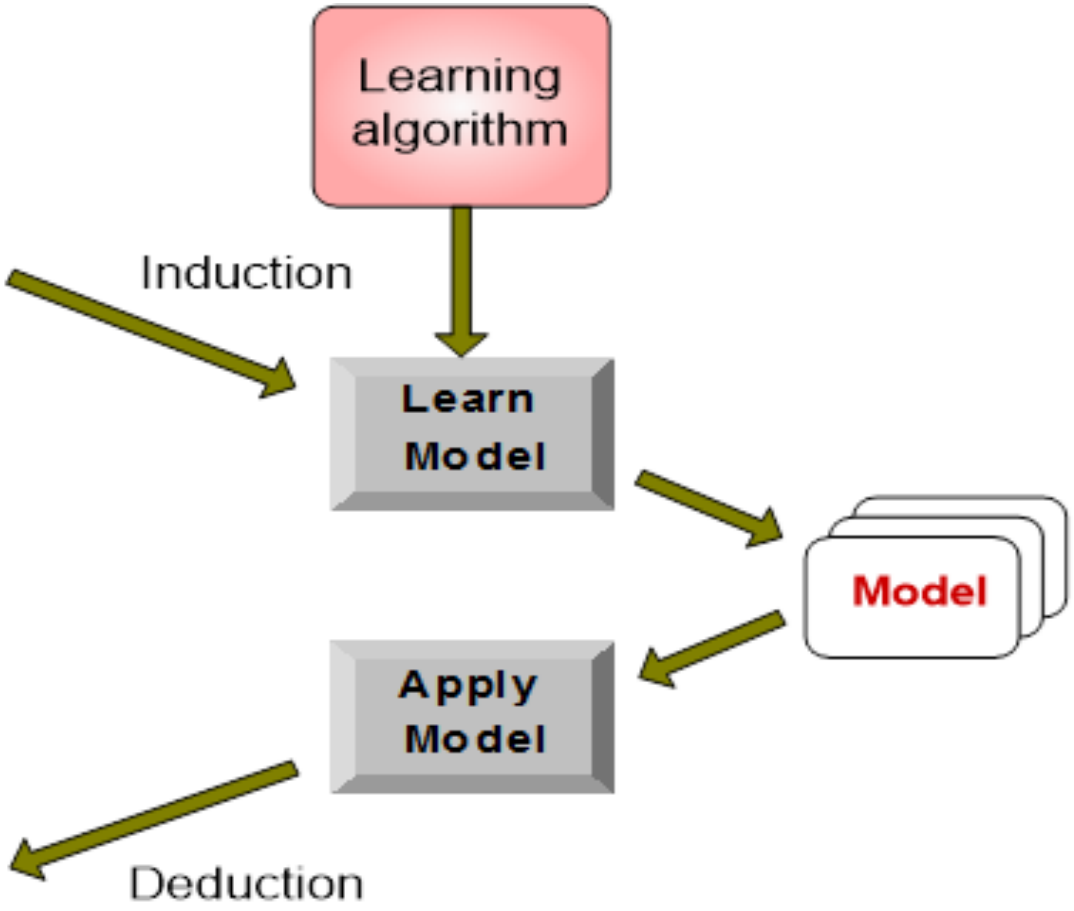
Phân lớp: Quá trình hai pha

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Học mô hình : Quá trình quy nạp
Áp dung mô hình: Quá trình suy diễn

Các loại phân lớp

- Phân lớp nhị phân/đa lớp

  Nhị phân: hai lớp (|C| = 2)

  Đa lớp: số lượng lớp > 2 (|C| > 2)
- Phân lớp đơn nhãn/đa nhãn/phân cấp

  Đơn nhãn: Một đối tượng chỉ thuộc duy nhất một lớp
- Đa nhãn: Một đối tượng thuộc một hoặc nhiều lớp
- Phân cấp: Lớp này là con của lớp kia

Các vấn đề đánh giá mô hình

- Các phương pháp đánh giá hiệu quả

  Câu hỏi: Làm thế nào để đánh giá được hiệu quả của một mô hình?
- Độ đo để đánh giá hiệu quả

  Câu hỏi: Làm thế nào để có được ước tính đáng tin cậy?
- Phương pháp so sánh mô hình

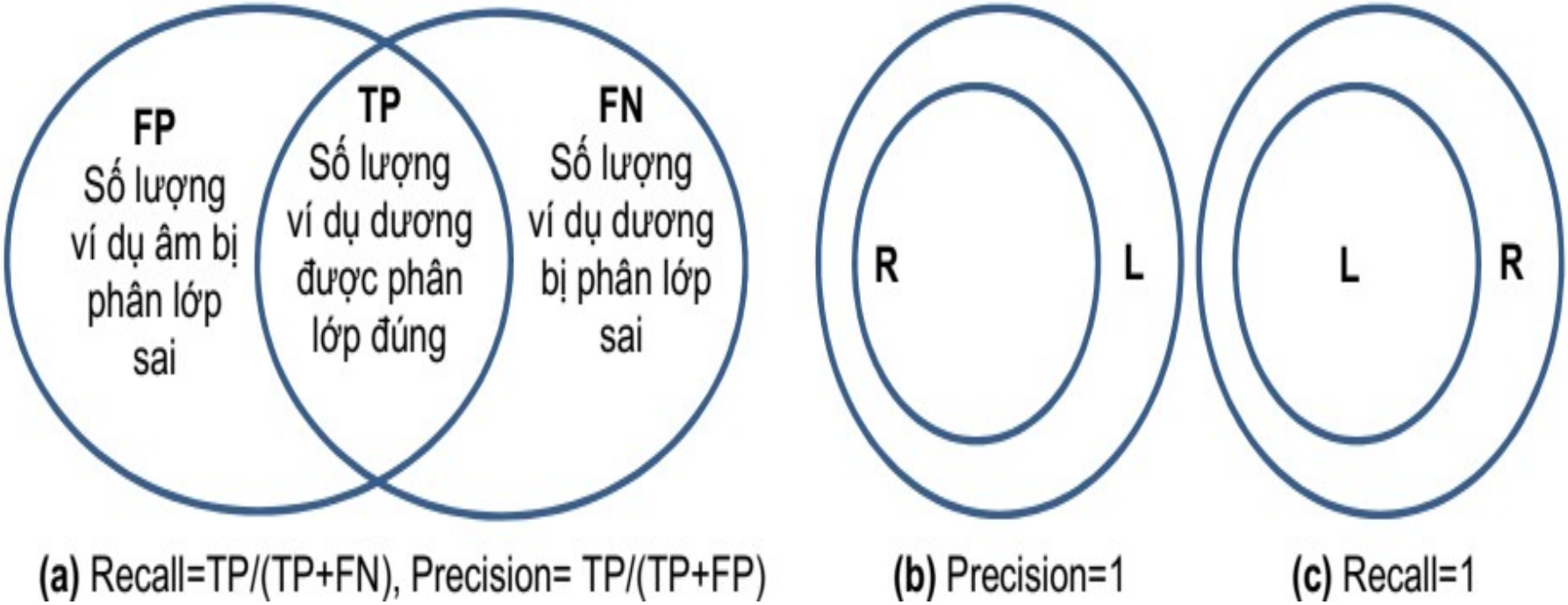
  Câu hỏi: Làm thế nào để so sánh hiệu quả tương đối giữa các mô hình có tính cạnh tranh?

Đánh giá phân lớp nhị phân

- Theo dữ liệu test
- Giá trị thực: P dương / N âm; Giá trị qua phân lớp: T đúng/F sai. : còn gọi là *ma trận nhầm lẫn*
- Sử dụng các ký hiệu TP (true positives), TN (true negatives), FP (false positives), FN (false negatives)
  - TP: số ví dụ dương P mà thuật toán phân đúng (T) cho dương P
  - TN: số ví dụ âm N mà thuật toán phân đúng (T) cho âm N
  - FN: số ví dụ dương P mà thuật toán phân sai (F) cho âmN
  - FP: số ví dụ âm N mà thuật toán phân sai (F) cho dương P
- Độ hồi tưởng , độ chính xác , các độ đo F_1 và F

$$\rho = \frac{TP}{TP + FN} \qquad \pi = \frac{TP}{TP + FP} \qquad f_\beta = \frac{(\beta^2 + 1)\pi\rho}{\beta^2\pi + \rho} \qquad f_1 = \frac{2\pi\rho}{\pi + \rho}$$

Đánh giá phân lớp nhị phân: minh họa



(a) Recall=TP/(TP+FN), Precision= TP/(TP+FP) **(b)** Precision=1 **(c)** Recall=1

R là tập ví dụ kiểm thử được bộ phân lớp gán nhãn dương, L là tập vị dụ kiểm thử thực tế có nhãn dương

Đánh giá phân lớp nhị phân

- Phương án khác đánh giá mô hình nhị phân theo độ chính xác (accuracy) và hệ số lỗi (Error rate)
- *Ma trận nhầm lẫn*

		Lớp dự báo	
		Lớp = 1	Lớp = 0
Lớp thực sự	Lớp = 1	f_{11}	f_{10}
	Lớp = 0	f_{01}	f_{00}

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

So sánh hai phương án

- Tập test có 9990 ví dụ lớp 0 và 10 ví dụ lớp 1. Kiểm thử: mô hình dự đoán cả 9999 ví dụ là lớp 0 và 1 ví dụ lớp 1 (chính xác: TP)
  - Theo phương án (precision, recall) có
    = 1/10=0.1; =1/1=1; f_1 = 2*0.1/(0.1+1.0)= 0.18
  - Theo phương án (accurary, error rate) có
    accurary=0.9991; error rate = 9/10000 = 0.0009
    Được coi là rất chính xác !
  - f_1 thể hiện việc đánh giá nhạy cảm với giá dữ liệu

Đánh giá phân lớp đa lớp

- Bài toán ban đầu: C gồm có k lớp
- Đối với mỗi lớp C_i , cho thực hiện thuật toán với các dữ liệu thuộc D_{test} nhận được các đại lượng TP_i, TF_i, FP_i, FN_i (như bảng dưới đây)

Lớp C_i		Giá trị thực	
		Thuộc lớp C_i	Không thuộc lớp C_i
Giá trị qua bộ phân lớp đa lớp	Thuộc lớp C_i	TP_i	FP_i
	Không thuộc lớp C_i	FN_i	TN_i

Đánh giá phân lớp đa lớp

- Tương tự bộ phân lớp hai lớp (nhị phân)
  - Độ chính xác Pr_i của lớp C_i là tỷ lệ số ví dụ dương được thuật toán phân lớp cho giá trị đúng trên tổng số ví dụ được thuật toán phân lớp vào lớp C_i :

$$\mathrm{Pr}_i = \frac{TP_i}{TP_i + FP_i}$$

  - Độ hồi tưởng Re_i của lớp C_i là tỷ lệ số ví dụ dương được thuật toán phân lớp cho giá trị đúng trên tổng số ví dụ dương thực sự thuộc lớp C_i:

$$\mathrm{Re}_i = \frac{TP_i}{TP_i + FN_i}$$

Đánh giá phân lớp đa lớp

- Các giá trị $_i$ và $_i$: độ hồi phục và độ chính xác đối với lớp C_i.
- Đánh giá theo các độ đo
  - trung bình mịn (micro – average, được ưa chuộng) và
  - trung bình thô (macro- average) ^{M} và ^{M}

$$\rho^M = \frac{1}{K}\sum_{c=1}^{K}\rho_c \qquad \pi^M = \frac{1}{K}\sum_{c=1}^{K}\pi_c$$

$$\rho^\mu = \frac{\sum_{c=1}^{K} TP_c}{\sum_{c=1}^{K}(TP_c + FN_c)} \qquad \pi^\mu = \frac{\sum_{c=1}^{K} TP_c}{\sum_{c=1}^{K}(TP_c + FN_c)}$$

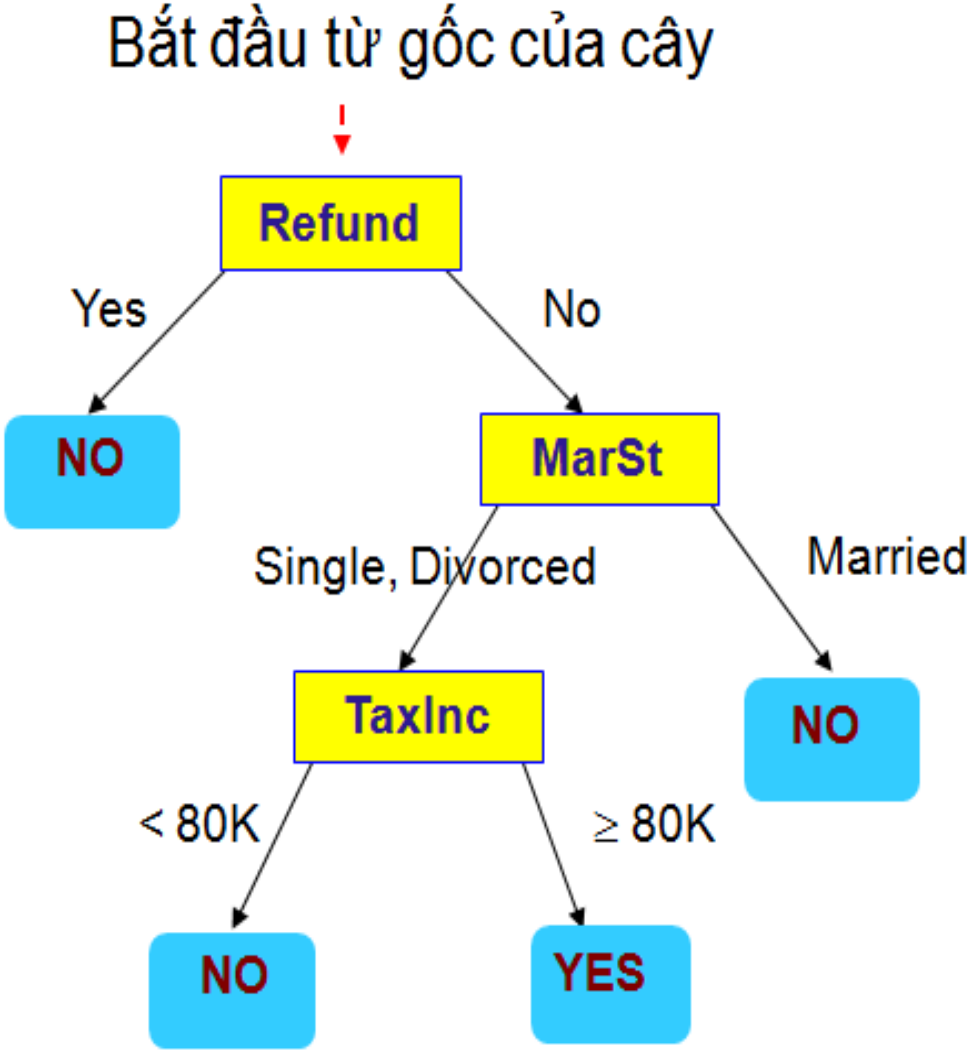
Các kỹ thuật phân lớp

- Các phương pháp cây quyết định
  Decision Tree based Methods
- Các phương pháp dựa trên luật
  Rule-based Methods
- Các phương pháp Bayes «ngây thơ» và mạng tin cậy Bayes
  Naïve Bayes and Bayesian Belief Networks
- Các phương pháp máy vector hỗ trợ
  Support Vector Machines
- Lập luận dưa trên ghi nhớ
  Memory based reasoning
- Các phương pháp mạng nơron
  Neural Networks
- Một số phương pháp khác

Phân lớp cây quyết định

- Mô hình phân lớp là cây quyết định
- Cây quyết định
  - Gốc: tên thuộc tính; không có cung vào + không/một số cung ra
  - Nút trong: tên thuộc tính; có chính xác một cung vào và một số cung ra (gắn với điều kiện kiểm tra giá trị thuộc tính của nút)
  - Lá hoặc nút kết thúc: giá trị lớp; có chính xác một cung vào + không có cung ra.
  - Ví dụ: xem trang tiếp theo
- Xây dựng cây quyết định
  - Phương châm: “chia để trị”, “chia nhỏ và chế ngự”. Mỗi nút tương ứng với một tập các ví dụ học. Gốc: toàn bộ dữ liệu học
  - Một số thuật toán phổ biến: Hunt, họ ID3+C4.5+C5.x
- Sử dụng cây quyết định
  - Kiểm tra từ gốc theo các điều kiện

Ví dụ cây quyết định và sử dụng



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Kết luận: Gán giá trị NO (không gian lận) vào trường Cheat cho bản ghi

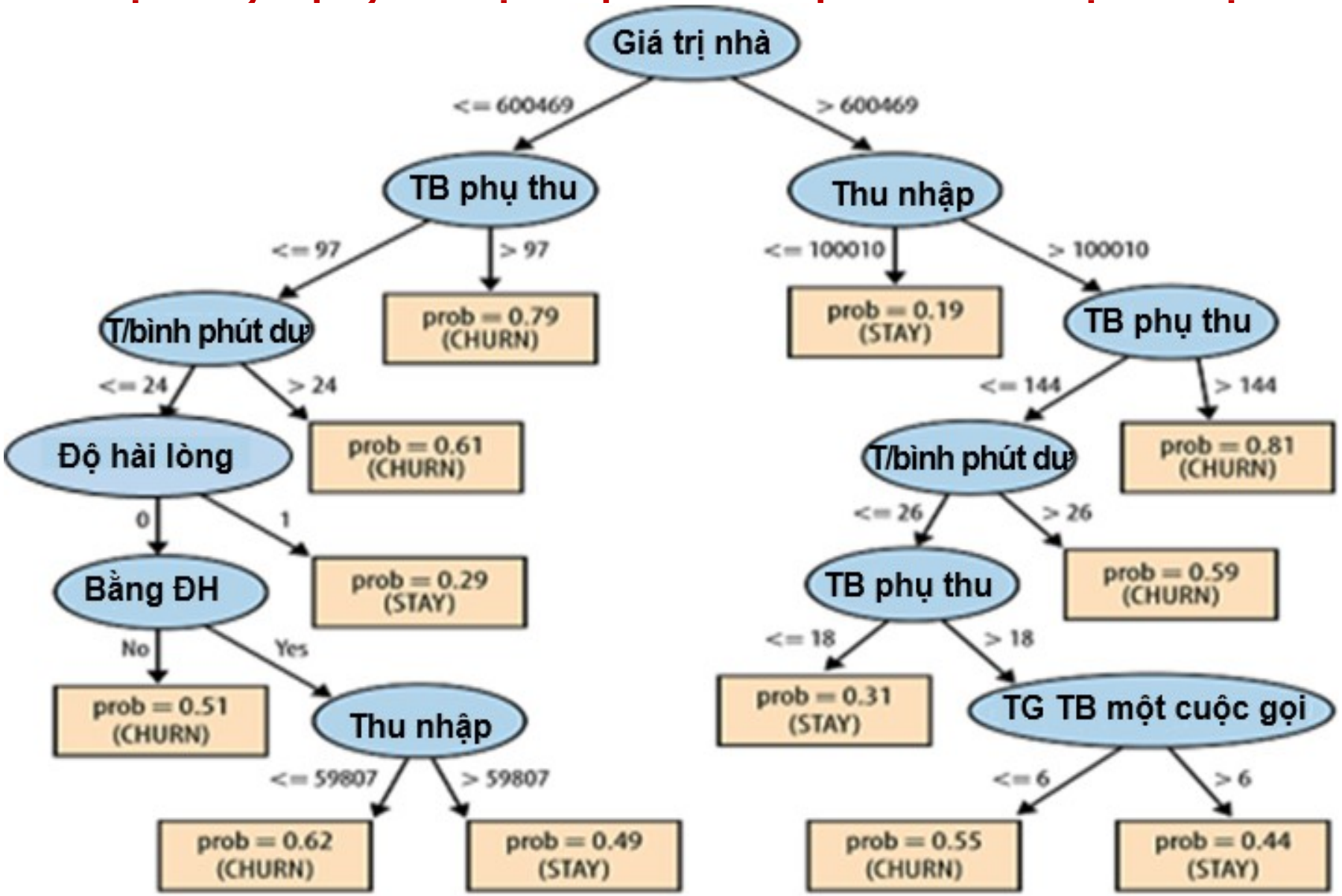
Ví dụ phân lớp: Bài toán rời bỏ dịch vụ

Biến		*Giải thích*
COLLEGE	**Bằng ĐH**	Khách hàng được đào tạo bậc đại học hay không? Biến này nhận giá trị YES (có) và NO (không)
INCOME	**Thu nhập**	Thu nhập hàng năm là tổng số tiền thu nhập mà khách hàng có trong một năm
OVERAGE	**TB phụ thu**	Trung bình phụ thu mỗi tháng
LEFTOVER	**T/bình phút dư**	Trung bình số phút còn dư mỗi tháng
HOUSE	**Giá trị nhà**	Giá trị ước tính nhà của khách hàng từ điều tra dân số
HANDSET_PRICE		Giá trị điện thoại cầm tay mà khách hàng sử dụng
LONG_CALLS_PER_MONTH		Trung bình số cuộc gọi dài (15 phút trở lên) theo tháng
AVERAGE_CALL_DURATION	**TGTB một cuộc gọi**	Thời gian trung bình một cuộc gọi
SATISFACTION	**Độ hài lòng**	Mức độ hài lòng của khách hàng theo báo cáo
REPORTED_USAGE_LEVEL		Mức sử dụng do người dùng tự đánh giá
LEAVE (*biến mục tiêu*)		Khách hàng đã ở lại hay rời mạng? Biến này nhận một trong hai giá trị là STAY (ở lại) và CHURN (rời bỏ)

Công ty điện thoại di động: các thuộc tính như liệt kê

"Lớp" liên quan tới **leave (rời bỏ)**

Ví dụ cây quyết định phân lớp rời bỏ dịch vụ



Dựng cây quyết định: thuật toán Hunt

- Thuật toán dựng cây quyết định sớm nhất, đệ quy theo nút của cây, bắt đầu từ gốc
- Input
  - Cho nút t trên cây quyết định đang được xem xét
  - Cho tập các ví dụ học D_t.
  - Cho tập nhãn lớp (giá trị lớp) y_1, y_1, … y_k. (k lớp)
- Output
  - Xác định nhãn nút t và các cung ra (nếu có) của t
- Nội dung

1: Nếu mọi ví dụ trong D_t đều thuộc vào một lớp y thì nút t là một lá và được gán nhãn y.

2: Nếu D_t chứa các ví dụ thuộc nhiều lớp thì

2.1. Chọn 1 thuộc tính A để phân hoạch D_t và gán nhãn nút t là A

2.2. Tạo phân hoạch D_t theo tập giá trị của A thành các tập con

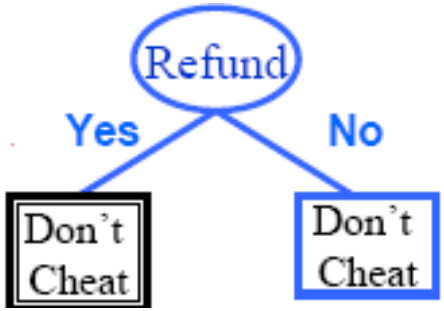
2.3. Mỗi tập con theo phân hoạch của D_t tương ứng với một nút con u của t: cung nối t tới u là miền giá trị A theo phân hoạch, tập con nói trên được xem xét vơi u tiếp theo. Thực hiện thuật toán với từng nút con u của t.

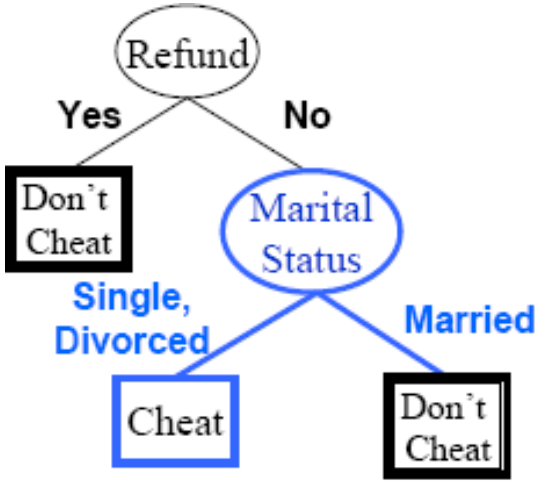
Ví dụ: thuật toán Hunt

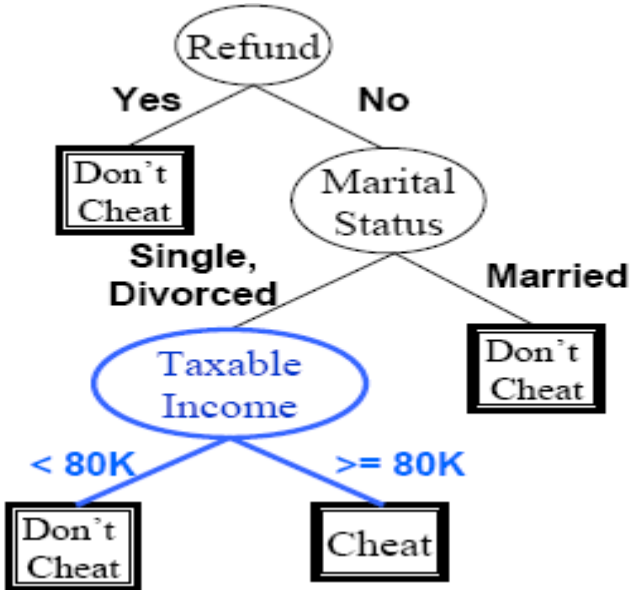
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Giải thích

- Xuất phát từ gốc với 10 bản ghi
- Thực hiện bước 2: **chọn thuộc tính Refund** có hai giá trị Yes, No. Chia thành hai tập gồm 3 bản ghi có Refund = Yes và 7 bản ghi có Refund = No
- Xét hai nút con của gốc từ trái sang phải. Nút trái có 3 bản ghi cùng thuộc lớp Cheat=No (Bước 1) nên là lá gán No (Don't cheat). Nút phải có 7 bản ghi có cả No và Yes nên áp dụng bước 2. Chọn thuộc tính Marital Status với phân hoạch Married và hai giá trị kia…







Thuật toán cây quyết định ID3

ID3 (*Examples, Target_attribute, Attributes*)

Ở đây: *Examples* là tập ví dụ học; *Target_attribute* là các thuộc tính đầu ra (lớp) cho cây quyết định dự đoán; *Attributes* là danh sách các thuộc tính khác tham gia trong quá trình học của cây quyết định. Kết quả thủ tục trả về cây quyết định phân lớp đúng các mẫu ví dụ đưa ra.

1. Tạo một nút gốc *Root* cho cây quyết định.
2. Nếu toàn bộ *Examples* đều là các ví dụ thuộc cùng một lớp thì trả lại cây *Root* một nút đơn với nhãn + (nếu các ví dụ thuộc lớp +) hoặc với nhãn - (nếu các ví dụ thuộc lớp -).
3. Nếu *Attributes* là rỗng thì trả lại cây *Root* một nút đơn với nhãn gán bằng giá trị phổ biến nhất của *Target_attribute* trong *Examples*.
4. Còn lại

   Begin

   4.1. Gán $A \leftarrow$ thuộc tính từ tập *Attributes* mà phân lớp tốt nhất tập *Examples*.

   4.2. Thuộc tính quyết định cho *Root* $\leftarrow A$

   4.3. Lặp với các giá trị có thể v_i của A,

   - Cộng thêm một nhánh cây con ở dưới *Root*, phù hợp với biểu thức kiểm tra $A = v_i$.
   - Đặt $Examples_{v_i}$ là một tập con của tập các ví dụ có giá trị v_i cho A
   - Nếu $Examples_{v_i}$ rỗng
     + Thì dưới mỗi nhánh mới thêm một nút lá với nhãn = giá trị phổ biến nhất của *Target_attribute* trong tập *Examples*.
     + Ngược lại thì dưới nhánh mới này thêm một cây con

       ID3($Examples_{v_i}$, Target_attribute, Attribute - {A}).

   End

5. Return *Root*.

Rút gọn cây

- Chiến lược tham lam
  - Phân chia tập dữ liệu dựa trên việc kiểm tra các thuộc tính **“chọn thuộc tính”** làm chiến lược tối ưu hóa
- Vấn đề cần giải quyết
  - Xác định cách phân chia tập dữ liệu
    - Cách xác định điều kiện kiểm tra thuộc tính
    - Cách xác định cách chia tốt nhất
    - Theo một số độ đo
  - Khi nào thì dừng phân chia (bước 2)
    - Tất cả các dữ liệu thuộc về cùng một lớp
    - Tất cả các dữ liệu có giá trị “tương tự nhau”
    - Ràng buộc dừng phân chia khác: (i) số lượng dữ liệu nhỏ thua ngưỡng cho trước, (ii) test khi-bình phương cho thấy phân bố lớp không phụ thuộc các thuộc tính hiện có; (iii) nếu phân chia không cải thiện chất lượng

Chọn thuộc tính: Độ đo Gini

- Bước 4.1. chọn thuộc tính A tốt nhất gán cho nút t.
- Tồn tại một số độ đo: Gini, Information gain…
- Độ đo Gini
  - Đo tính hỗn tạp của một tập ví dụ mẫu theo "lớp"
  - Công thức tính độ đo Gini cho nút t: $Gini(t) = 1 - \sum_{j=1} [p(j|t)]^2$

    Trong đó p(j|t) là tần suất liên quan của lớp j tại nút t
  - Gini (t) lớn nhất = 1-1/n_c (với n_c là số các lớp tại nút t): khi các bản ghi tại t phân bố đều cho n_c lớp; tính hỗn tạp cao nhất, không có phân biệt giữa các lớp
  - Gini (t) nhỏ nhất = 0 khi tất cả các bản ghi thuộc một lớp duy nhất.
- Ví dụ: Bốn trường hợp

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Chia tập theo độ đo Gini

- Dùng trong các thuật toán CART, SLIQ, SPRINT
- Khi một nút t được phân hoạch thành k phần (k nút con của t) thì chất lượng của việc chia tính bằng

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

trong đó

- n là số bản ghi của tập bản ghi tại nút t,
- .n_i là số lượng bản ghi tại nút con I (của nút t).

Chia tập theo độ đo Gini: Ví dụ

- Tính toán GINI cho Refund (Yes, No), Marital Status (Single&Divorced, Married) và Taxable Income (<80K, 80K).
- Refund: 3/10 * (0) + 7/10 * (1-(3/7)^2 – (4/7)^2) = 7/10*(24/49) = 24/70
- Marital Status: 4/10 * 0 + 6/10 * (1- (3/6)^2 – (3/6)^2) = 6/10 * ½ = 3/10
- Taxable Income: thuộc tính liên tục cần chia khoảng (tồn tại một số phương pháp theo Gini, kết quả 2 thùng và 80K là mốc)

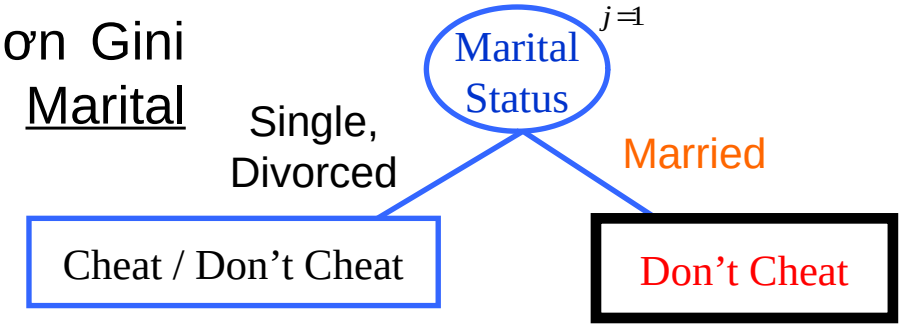
  3/10 * (0) + 7/10 * (1-(3/7)^2 – (4/7)^2) = 7/10*(24/49) = 24/70

Như vậy, Gini của Refund và Taxable Income bằng nhau (24/70) và lớn hơn Gini của Marital Status (3/10) nên chọn Marital Status cho gốc cây quyết định !

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

$$Gini(t) = 1 - \sum_{j=1} [p(j \mid t)]^2$$



Chọn thuộc tính: Information Gain

- Độ đo Information Gain
  - Thông tin thu được sau khi phân hoạch tập ví dụ
  - Dùng cho các thuật toán ID3, họ C4.5
- Entropy
  - Công thức tính entropy nút t:
    $$Entropy(t) = -\sum_j p(j|t)\log p(j|t)$$
    Trong đó p(j|t) là tần suất liên quan của lớp j tại nút t
    độ không đồng nhất tại nút t.
  - Entropy (t) lớn nhất = log (n_c) (với n_c là số các lớp tại nút t): khi các bản ghi tại t phân bố đều cho n_c lớp; tính hỗn tạp cao nhất, không có phân biệt giữa các lớp
  - Entropy (t) nhỏ nhất = 0 khi tất cả các bản ghi thuộc một lớp duy nhất.
  - Lấy loga cơ số 2 thay cho loga tự nhiên
- Tính toán entropy (t) cho một nút tương tự như Gini (t)

Chọn thuộc tính: Information Gain

- Độ đo Information Gain

$$Gain_{chia} = entropy(t) - \sum_{i=1}^{k} \frac{n_i}{n} entropy(i)$$

Trong đó, n là số lượng bản ghi tại nút t, k là số tập con trong phân hoạch, n_i là số lượng bản ghi trong tập con thứ i.

Độ đo giảm entropy sau khi phân hoạch: chọn thuộc tính làm cho Gain đạt lớn nhất.

C4.5 là một trong 10 thuật toán KPDL phổ biến nhất.

  - *Hạn chế*: Xu hướng chọn phân hoạch chia thành nhiều tập con

- Cải tiến

$$GainRATIO = \frac{Gain_{chia}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

  - Dùng GainRatio để khắc phục xu hướng chọn phân hoạch nhiều tập con

- Áp dụng: Tự tiến hành

Phân lớp dựa trên luật

- Giới thiệu
  - Phân lớp các bản ghi dựa vào tập các luật "kiểu" if … then
- Luật
  - Luật: <điều kiện> y
    Trong đó:
    <điều kiện> là sự kết nối các thuộc tính (còn gọi là tiên đề/điều kiện của luật: LHS bên trái)
    y là nhãn lớp (còn gọi là kết quả của luật: RHS bên phải).
  - Ví dụ
    Refund = 'Yes" Cheat = "No"
    (Refund = "No") (Marital Status = "Married") Cheat = "No"
- Sử dụng luật
  - Một luật được gọi là "bảo đảm" thể hiện r (bản ghi) nếu các thuộc tính của r đáp ứng điều kiện của luật.
  - Khi đó, vế phải của luật cũng được áp dụng cho thể hiện.

Xây dựng luật phân lớp

- Giới thiệu
  - Trực tiếp và gián tiếp
- Trực tiếp
  - Trích xuất luật trực tiếp từ dữ liệu
  - Ví dụ: RIPPER, CN2, Holte’s 1R
  - Trích xuất luật trực tiếp từ dữ liệu
    1. Bắt đầu từ một tập rỗng
    2. Mở rộng luật bằng hàm Học_một_luật
    3. Xóa mọi bản ghi “bảo đảm” bởi luật vừa được học
    4. Lặp các bước 2-3 cho đến khi gặp điều kiện dừng
- Gián tiếp
  - Trích xuất luật từ mô hình phân lớp dữ liệu khác, chẳng hạn, mô hình cây quyết định, mô hình mạng nơ ron, …
  - Ví dụ:C4.5Rule

Mở rộng luật: một số phương án

- Sử dụng thống kê
  - Thống kê các đặc trưng cho ví dụ
  - Tìm đặc trưng điển hình cho từng lớp
- Thuật toán CN2
  - Khởi đầu bằng liên kết rỗng: {}
  - Bổ sung các liên kết làm cực tiểu entropy: {A}, {A, B}…
  - Xác định kết quả luật theo đa số của các bản ghi đảm bảo luật

Table 3. The CN2 induction algorithm.

```
Let E be a set of classified examples.
Let SELECTORS be the set of all possible selectors.

Procedure CN2(E)
   Let RULE_LIST be the empty list.
   Repeat until BEST_CPX is nil or E is empty:
      Let BEST_CPX be Find_Best_Complex(E).
      If BEST_CPX is not nil,
      Then let E' be the examples covered by BEST_CPX.
           Remove from E the examples E' covered by BEST_CPX.
           Let C be the most common class of examples in E'.
           Add the rule 'If BEST_CPX then the class is C'
             to the end of RULE_LIST.
   Return RULE_LIST.

Procedure Find_Best_Complex(E)
   Let STAR be the set containing the empty complex.
   Let BEST_CPX be nil.
   While STAR is not empty,
      Specialize all complexes in STAR as follows:
      Let NEWSTAR be the set {x ∧ y|x ∈ STAR, y ∈ SELECTORS}.
      Remove all complexes in NEWSTAR that are either in STAR (i.e.,
         the unspecialized ones) or null (e.g., big = y ∧ big = n).
      For every complex Ci in NEWSTAR:
           If Ci is statistically significant and better than
              BEST_CPX by user-defined criteria when tested on E,
           Then replace the current value of BEST_CPX by Ci.
      Repeat until size of NEWSTAR ≤ user-defined maximum:
         Remove the worst complex from NEWSTAR.
      Let STAR be NEWSTAR.
   Return BEST_CPX.
```

Mở rộng luật: một số phương án

- Thuật toán RIPPER
  - Bắt đầu từ một luật rỗng: {} lớp
  - Bổ sung các liên kết làm cực đại lợi ích thông tin FAIL
  - R0: {} => lớp (luật khởi động)
  - R1: {A} => lớp (quy tắc sau khi thêm liên kết)
  - Gain (R0, R1) = t [log (p1 / (p1 + n1)) - log (p0 / (p0 + n0))]

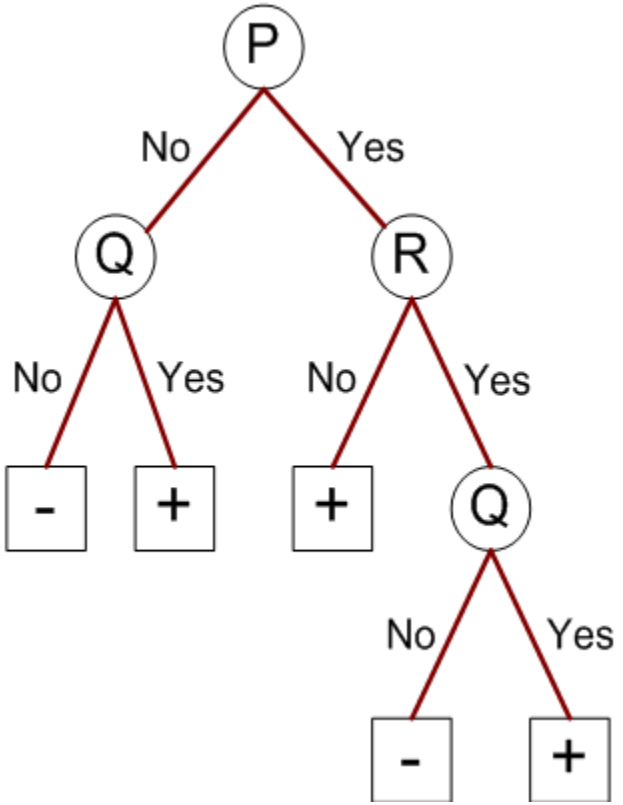
  với t: số thể hiện đúng đảm bảo cả hai R0 và R1

  p0: số thể hiện đúng được bảo đảm bởi R0
    - n0: số thể hiện sai được đảm bảo bởi R0
    - P1: số thể hiện đúng được bảo đảm bởi R1
    - n 1: số trường hợp sai được đảm bảo bởi R1

```
function MaxGainTest(S, i)
1   Visited := ∅;
2   TotalCount[+]:=0; TotalCount[−]:=0;
3   for each example ⟨x, y⟩ in the sample S do
4       for each string s ∈ x_i do
5           Visited := Visited ∪{s};
6           ElemCount[s, y] := ElemCount[s, y]+1;
7       endfor
8       TotalCount[y] := TotalCount[y]+1;
9   endfor
10  BestEntropy = −1;
11  for each s ∈ Visited do
12      p := ElemCount[s,+]; n := ElemCount[s,−];
13      if (Entropy(p, n) > BestEntropy) then
14          BestTest := "s ∈ u_i";
15          BestEntropy := Entropy(p, n);
16      endif
17      p' := TotalCount[+] − ElemCount[s,+];
18      n' := TotalCount[−] − ElemCount[s,−];
19      if (Entropy(p', n') > BestEntropy) then
20          BestTest := "s ∉ u_i";
21          BestEntropy := Entropy(p', n');
22      endif
23      ElemCount[s,+] := 0;
24      ElemCount[s,−] := 0
25  endfor
26  return BestTest
```

Figure 1: Finding the best element-of test

Luật phân lớp: từ cây quyết định



Tập luật

Liệt kê các đường đi từ gốc

r1: (P=No,Q=No) ==> -

r2: (P=No,Q=Yes) ==> +

r3: (P=Yes,R=No) ==> +

r4: (P=Yes,R=Yes,Q=No) ==> -

r5: (P=Yes,R=Yes,Q=Yes) ==> +

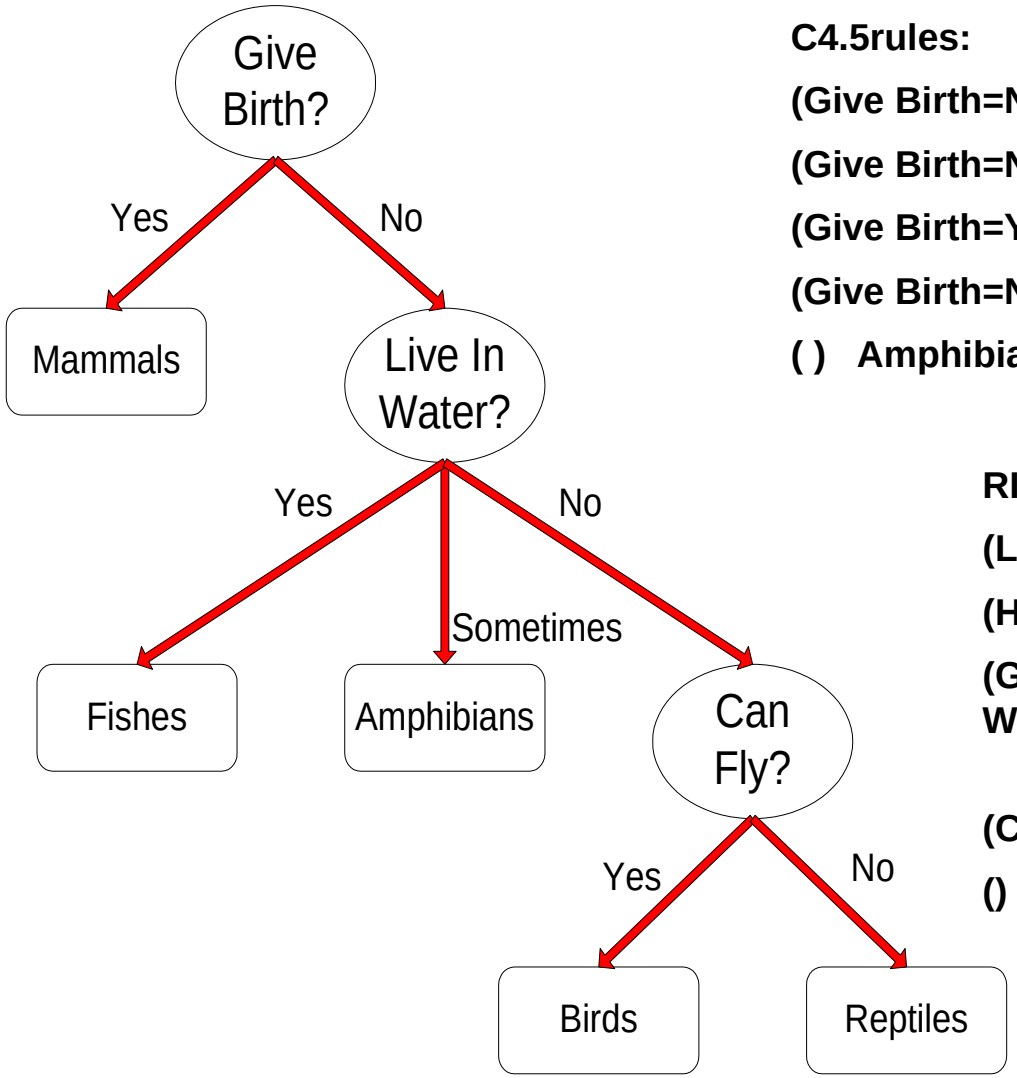
Sinh luật gián tiếp: C4.5rules

- Trích xuất luật từ cây quyết định chưa cắt tỉa
- Với mỗi luật, r: A → y
  - Xem xét luật thay thế r': A' → y, trong đó A' nhận được từ A bằng cách bỏ đi một liên kết
  - So sánh tỷ lệ lỗi r so với các r'
  - Loại bỏ các r' có lỗi thấp hơn r
  - Lặp lại cho đến khi không cải thiện được lỗi tổng thể
- Thay thế sắp xếp theo luật bằng sắp xếp theo tập con của luật (thứ tự lớp)
  - Mỗi tập con là một tập các luật với cùng một kết quả (lớp)
  - Tính toán độ dài mô tả của mỗi tập con
  - Độ dài mô tả = L(lỗi) + g* L(mô hình)
  - g : tham số đếm sự hiện diện của các thuộc tính dư thừa trong một tập luật (giá trị chuẩn, g=0.5)

C4.5rules: Ví dụ

Name	Give Birth	Lay Eggs	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	no	yes	mammals
python	no	yes	no	no	no	reptiles
salmon	no	yes	no	yes	no	fishes
whale	yes	no	no	yes	no	mammals
frog	no	yes	no	sometimes	yes	amphibians
komodo	no	yes	no	no	yes	reptiles
bat	yes	no	yes	no	yes	mammals
pigeon	no	yes	yes	no	yes	birds
cat	yes	no	no	no	yes	mammals
leopard shark	yes	no	no	yes	no	fishes
turtle	no	yes	no	sometimes	yes	reptiles
penguin	no	yes	no	sometimes	yes	birds
porcupine	yes	no	no	no	yes	mammals
eel	no	yes	no	yes	no	fishes
salamander	no	yes	no	sometimes	yes	amphibians
gila monster	no	yes	no	no	yes	reptiles
platypus	no	yes	no	no	yes	mammals
owl	no	yes	yes	no	yes	birds
dolphin	yes	no	no	yes	no	mammals
eagle	no	yes	yes	no	yes	birds

C4.5rules: Ví dụ



C4.5rules:

(Give Birth=No, Can Fly=Yes) Birds

(Give Birth=No, Live in Water=Yes) Fishes

(Give Birth=Yes) Mammals

(Give Birth=No, Can Fly=No, Live in Water=No) Reptiles

( ) Amphibians

RIPPER:

(Live in Water=Yes) Fishes

(Have Legs=No) Reptiles

(Give Birth=No, Can Fly=No, Live In Water=No)
Reptiles

(Can Fly=Yes,Give Birth=No) Birds

() Mammals

Phân lớp Bayes (19/10)

- Giới thiệu
  - Khung xác suất để xây dựng bộ phân lớp
  - Mô hình phân lớp: Tập công thức tính xác suất
- Cơ sở khoa học: X/suất có điều kiện, đ/lý Bayes
  - Xác suất có điều kiện Hai biến cố A và C

$$P(C \mid A) = \frac{P(A,C)}{P(A)}; P(A \mid C) = \frac{P(A,C)}{P(C)}$$

  - Định lý Bayes: **P(c|x) = P(x|c).P(c)/P(x)**
  - P(x) bằng nhau cho tất cả các lớp
  - Tìm c sao cho P(c|x) lớn nhất Tìm c sao cho P(x|c).P(c) lớn nhất
  - P(c): tần suất xuất hiện của các tài liệu thuộc lớp c
  - Vấn đề: cách thức tính P(x|c)?

Định lý Bayes: Ví dụ

- Cho biết
  - Bệnh nhân viêm màng não M có triệu chứng cứng cổ S S| M: 50%
  - Xác suất một bệnh nhân bị viêm màng não M là 1/50.000
  - Xác suất một bệnh nhân bị cứng cổ S là 1/20
- Với một bệnh nhân bị cứng cổ S, hỏi xác suất anh/ cô ta bị viêm màng não M ?

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Pang-Ning Tan, Michael Steinbach, Vipin Kumar. Introduction to Data Mining (Chapter 5: Classification: Alternative Techniques), *Addison Wesley*, 2005, http://www.cs.uu.nl/docs/vakken/dm/dmhc13.pdf

Phân lớp Bayes

- n+1 thuộc tính (bao gồm lớp) là các biến ngẫu nhiên.
- Cho một bản ghi với các giá trị thuộc tính (A_1, A_2, …, A_n) là giá trị quan sát được các biến ngẫu nhiên
  - Cần dự báo nhãn c
  - Tìm lớp c để cực đại xác suất $P(C|A_1, A_2, …, A_n)$
- Có thể tính xác suất $P(C|A_1, A_2, …, A_n)$ từ dữ liệu học?

Pang-Ning Tan, Michael Steinbach, Vipin Kumar. Introduction to Data Mining (Chapter 5: Classification: Alternative Techniques), *Addison Wesley*, 2005, http://www.cs.uu.nl/docs/vakken/dm/dmhc13.pdf

Phân lớp Naïve Bayes

- Giả thiết Naïve Bayes:
  - giả thiết độc lập: xác suất xuất hiện của thuộc tính trong đối tượng độc lập với ngữ cảnh và vị trí của nó trong đối tượng:

$$p(c \mid x, \tau) = \sum_{T \, in \, \tau} p(c \mid x, T) p(T \mid \overline{x})$$

$$\mathrm{P}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k} \mid \mathrm{C}) = \mathrm{P}(\mathrm{x}_1 \mid \mathrm{C}) \cdot \ldots \cdot \mathrm{P}(\mathrm{x}_\mathrm{k} \mid \mathrm{C})$$

Phân lớp Naïve Bayes

- Cho
  - Tập ví dụ $D_{exam} = D_{learn} + D_{test}$
  - Tập lớp C= {C1, C2, …, Cn} với mỗi C_i một ngưỡng $_i > 0$
- Tính xác suất tiên nghiệm
  - Trên tập ví dụ học D_{learn}
  - Xác suất $p(C_i) = M_i/M$, $M = ||D_{learn}||$, $M_i = ||X \quad D_{learn} \quad C_i||$
  - Xác suất một giá trị đặc trưng f_j thuộc lớp C:

$$p(f_j|C) = \frac{1 + TF(f_j, C)}{|F_j| + \sum_{l=1}^{|F_j|} TF(f_l, C)} = \frac{1 + TF(f_j, C)}{|F_j| + |\{d \in D_C\}|}$$

F_j : Tập các giá trị phân biệt của thuộc tính A_j

D_C: Tập ví dụ có nhãn lớp C

TF (f_j, C): số lần giá trị đặc trưng f_j tại thuộc tính Aj xuất hiện trong C

Phân lớp Naïve Bayes

- Cho dữ liệu X mới
  - Tính xác suất hậu nghiệm

$$P(C \mid X) = \frac{p(C) * \prod_{j \in 1..n} (p(f_j \mid C)}{\sum_{i=1}^{k} p(C_i) * \prod_{j \in 1..n} (p(f_j \mid C_i)}$$

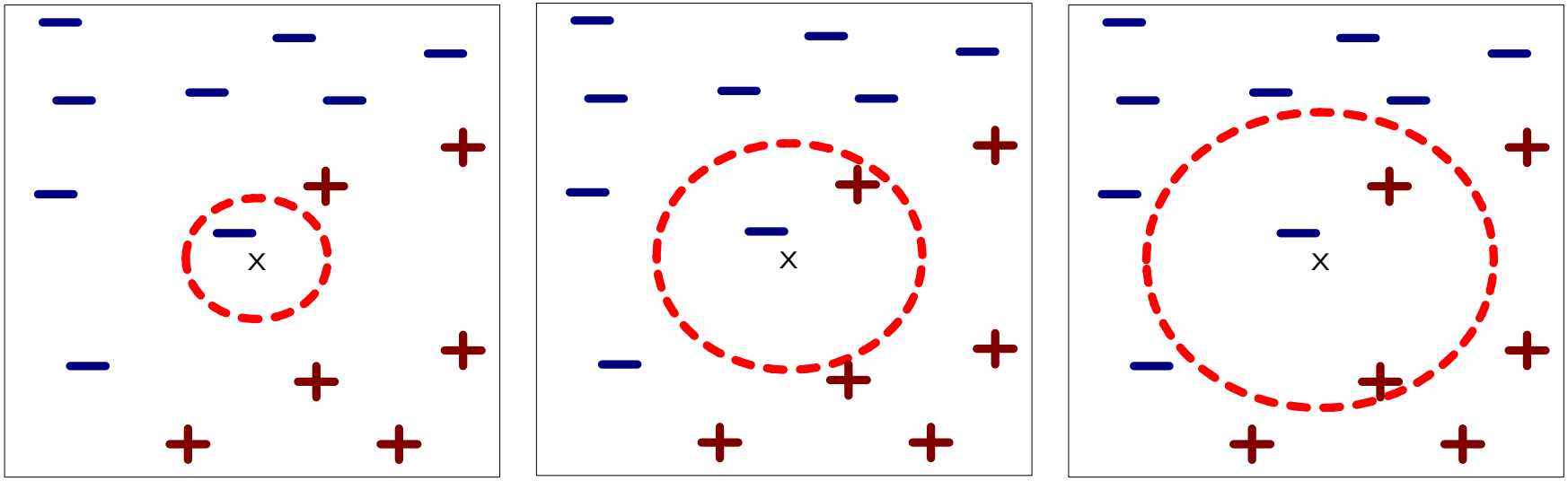
  - Nếu P(C|X) > $_C$ thì X C
  - *n* là số lượng thuộc tính, k là số lượng nhãn

Phân lớp k-NN

$$Sm(X,Y) = Cos(X,Y) = \frac{\sum_l X_l * Y_l}{\sqrt{\sum_l X_l^2 \sum_l Y_l^2}}$$

- Cho trước
  - Một tập D các đối tượng dữ liệu biểu diễn bản ghi các đặc trưng
  - Một đo đo khoảng cách (Ơcơlit) hoặc tương tự (như trên)
  - Một số k > 0 (láng giềng gần nhất
- Phân lớp đối tượng mới Xc được biểu diễn
  - Tính khoảng cách (độ tương tự) từ X tới tất cả dữ liệu thuộc D
  - Tìm k dữ liệu thuộc D gần X nhất
  - Dùng nhãn lớp của k-láng giềng gần nhất để xác định nhãn lớp của X: nhãn nhiều nhất trong k-láng giềng gần nhất

Phân lớp k-NN: Ví dụ



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

- Ba trường hợp như hình vẽ
  - 1-NN: Chọn lớp “-”: láng giềng có nhãn “-” là nhiều nhất
  - 2-NN: Chọn lớp “-”: hai nhãn có số lượng như nhau, chọn nhãn có tổng khoảng cách gần nhất
  - 3-NN: Chọn lớp “+”: láng giềng có nhãn “+” là nhiều nhất

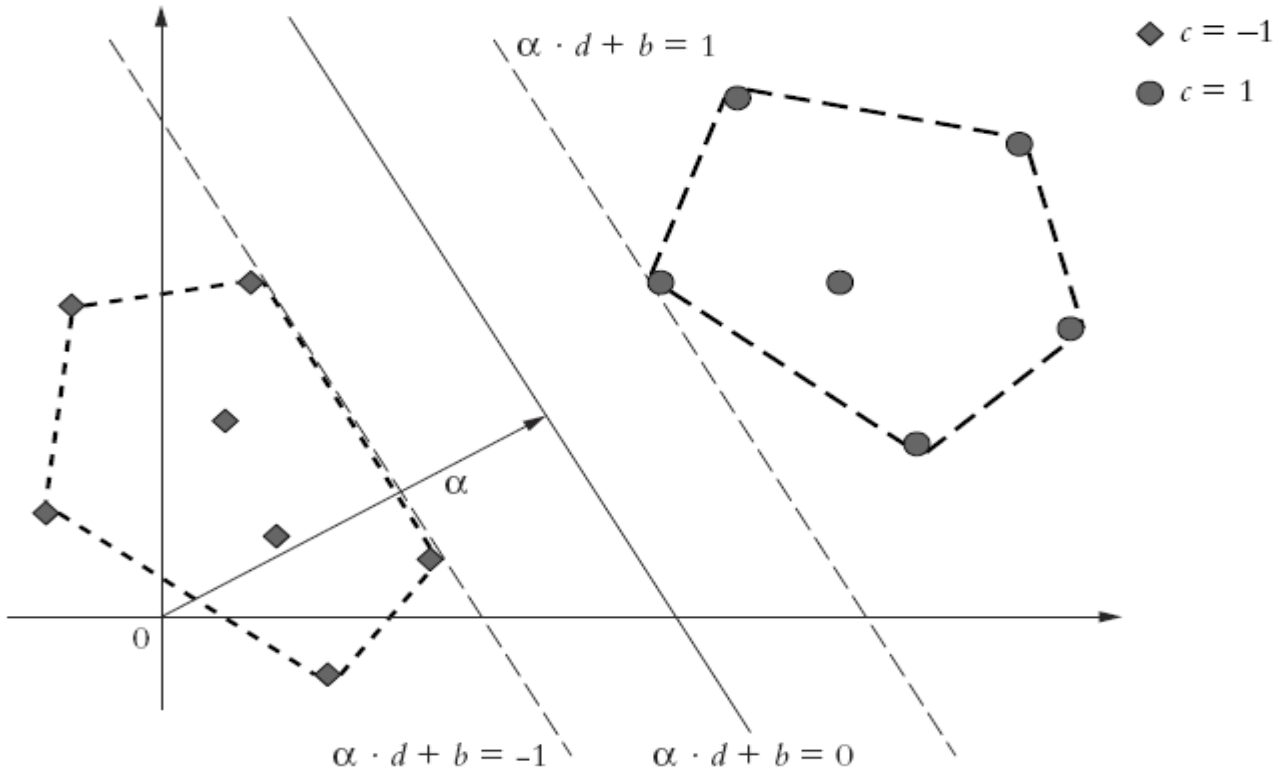
Thuật toán SVM

- Thuật toán máy vector hỗ trợ (Support Vector Machine – SVM): được Corters và Vapnik giới thiệu vào năm 1995.
- SVM rất hiệu quả để giải quyết các bài toán với dữ liệu có số chiều lớn (như các vector biểu diễn văn bản).

Thuật toán SVM

- Tập dữ liệu học: D= $\{(X_i, C_i), i=1,\ldots n\}$
  - $C_i \in \{-1,1\}$ xác định dữ liệu dương hay âm
- Tìm một siêu phẳng: $\alpha_{SVM} .\boldsymbol{d} + \boldsymbol{b}$ phân chia dữ liệu thành hai miền.
- Phân lớp một tài liệu mới: xác định dấu của
  - $f(d) = \alpha_{SVM} .\boldsymbol{d} + \boldsymbol{b}$
  - Thuộc lớp dương nếu $f(d) > 0$
  - Thuộc lớp âm nếu $f(d) < 0$

Thuật toán SVM



Thuật toán SVM

- Nếu dữ liệu học là tách rời tuyến tính:
  - Cực tiểu: $\frac{1}{2}\alpha.\alpha \quad \left(=\frac{1}{2}\|\alpha\|^2\right) \quad (1)$
  - Thỏa mãn: $c_i\left(\alpha.d_i+b\right)\geq 1 \quad \forall i=1,\ldots,n \quad (2)$
- Nếu dữ liệu học không tách rời tuyến tính: thêm biến {$\xi 1 \ldots \xi n$}:
  - Cực tiểu: $\frac{1}{2}\alpha.\alpha + C\sum_{i=1}^{n}\xi_i \ (3)$
  - Thỏa mãn: $c_i\left(\alpha.d_i+b\right)\geq 1-\xi_i \quad \forall i=1,\ldots,n$

$$\xi_i \geq 0 \ \forall i=1,\ldots,n \quad (4)$$

Phân lớp bán giám sát

- Giới thiệu phân lớp bán giám sát
  - Khái niệm sơ bộ
  - Tại sao học bán giám sát
- Nội dung phân lớp bán giám sát
  - Một số cách tiếp cận cơ bản
  - Các phương án học bán giám sát phân lớp
- Phân lớp bán giám sát trong NLP

Sơ bộ về học bán giám sát

- Học bán giám sát là gì ? Xiaojin Zhu [1] FQA
  - Học giám sát: tập ví dụ học đã được gán nhãn (ví dụ gắn nhãn) là tập các cặp (tập thuộc tính, nhãn)
  - ví dụ gắn nhãn
    - Thủ công: khó khăn chuyên gia tốn thời gian, tiền
    - Tự động: như tự động sinh corpus song hiệu quả chưa cao
  - ví dụ chưa gắn nhãn
    - Dễ thu thập nhiều
      - xử lý tiếng nói: bài nói nhiều, xây dựng tài nguyên đòi hỏi công phu
      - xử lý văn bản: trang web vô cùng lớn, ngày càng được mở rộng
    - Có sẵn có điều kiện tiến hành tự động gắn nhãn
  - Học bán giám sát: dùng cả ví dụ có nhãn và ví dụ chưa gắn nhãn
    - Tạo ra bộ phân lớp tốt hơn so với chỉ dùng học giám sát: học bán giám sát đòi hỏi điều kiện về dung lượng khối lượng

Cơ sở của học bán giám sát

- Biểu diễn dữ liệu chưa mô tả hết ánh xạ gán nhãn trên dữ liệu
  - chẳng hạn, nghịch lý “hiệu quả như nhau” trong biểu diễn văn bản
- Ánh xạ gán nhãn có liên quan mô hình dữ liệu (mô hình / đặc trưng/ nhân / hàm tương tự) mô hình đã có theo tự nhiên hoặc giả thiết dữ liệu tuân theo.

Hiệu lực của học bán giám sát

- Dữ liệu chưa nhãn không luôn luôn hiệu quả
  - Nếu giả thiết mô hình không phù hợp giảm hiệu quả
  - Một số phương pháp cần điều kiện về miền quyết định: tránh miền có mật độ cao:
    - Transductive SVM (máy hỗ trợ vector lan truyền)
    - Information Regularization (quy tắc hóa thông tin)
    - mô hình quá trinh Gauxơ với nhiễu phân lớp bằng không
    - phương pháp dựa theo đồ thị với trọng số cạnh là khoảng cách
  - “Tồi” khi dùng phương pháp này song lại “tốt” khi dùng phương pháp khác

Phương pháp học bán giám sát

- Các phương pháp học bán giám sát điển hình
  - EM với mô hình trộn sinh
  - Self-training
  - Co-training
  - TSVM
  - Dựa trên đồ thị
  - ...
- So sánh các phương pháp
  - Đòi hỏi các giả thiết mô hình mạnh. Giả thiết mô hình phù hợp cấu trúc dữ liệu: khó kiểm nghiệm
  - Một số định hướng lựa chọn
    - Lớp phân cụm tốt: dùng EM với mô hình sinh trộn.
    - Đặc trưng phân thành hai phần riêng rẽ: co-training
    - Nếu hai điểm tương tự hướng tới một lớp: dựa trên đồ thị
    - Đã sử dụng SVM thì mở rộng TSVM
    - Khó nâng cấp học giám sát đã có: dùng self-traning
    - …

Phương pháp học bán giám sát

- Dùng dữ liệu chưa gán nhãn
  - Hoặc biến dạng hoặc thay đổi thứ tự giả thiết thu nhờ chỉ dữ liệu có nhãn
  - Mô tả chung
    - Giả thiết dưới dạng p(y|x) còn dữ liệu chưa có nhãn p(x)
    - Mô hình sinh có tham số chung phân bố kết nối p(x, y)
    - Mô hình trộn với EM mở rộng thêm self-training
    - Nhiều phương pháp là phân biệt: TSVM, quy tắc hóa thông tin, quá trình Gauxơ, dựa theo đồ thị
  - Có dữ liệu không nhãn: nhận được xác suất p(x)
- Phân biệt “học lan truyền” với “học bán giám sát”
  - Đa dạng về cách gọi. Hạn chế bài toán phân lớp.
  - “Bán giám sát”
    - dùng ví dụ có / không có nhãn,
    - “học dữ liệu nhãn/không nhãn,
    - “học dữ liệu phân lớp/có nhãn bộ phận”.
    - Có cả lan truyền hoặc quy nạp.
  - Lan truyền để thu hẹp lại cho quy nạp: học chỉ dữ liệu sẵn. Quy nạp: có thể liên quan tới dữ liệu chưa có.

Mô hình sinh: Thuật toán EM

- Sơ bộ
  - Mô hình sớm nhất, phát triển lâu nhất
  - Mô hình có dạng p(x,y) = p(y)*p(x|y)
  - Với số lượng nhiều dữ liệu chưa nhãn cho P(x|y) mô hình trộn đồng nhất. Miền tài liệu được phân thành các thành phần,
  - Lý tưởng hóa tính "Đồng nhất": chỉ cần một đối tượng có nhãn cho mỗi thành phần
- Tính đồng nhất
  - Là tính chất cần có của mô hình
  - Cho họ phân bố {p} là đồng nhất nếu $_1$ $_2$ thì p$_1$ p$_2$ cho tới một hoán đối vị trí các thành phần tính khả tách của phân bố tới các thành phần

Mô hình sinh: Thuật toán EM

- Tính xác thực của mô hình
  - Giả thiết mô hình trộn là chính xác dữ liệu không nhãn sẽ làm tăng độ chính xác phân lớp
  - Chú ý cấu trúc tốt mô hình trộn: nếu tiêu đề được chia thành các tiêu đề con thì nên mô hình hóa thành đa chiều thay cho đơn chiều
- Cực đại EM địa phương
  - Miền áp dụng
    - Khi mô hình trộn chính xác
  - Ký hiệu
    - D: tập ví dụ đã có (có nhẫn /chưa có nhãn)
    - D^K: tập ví dụ có nhãn trong D ($|D^K| << |D|$)

Mô hình sinh: Thuật toán EM

- Nội dung thuật toán

1: Cố định tập tài liệu không nhãn D^U $D \setminus D^K$ dùng trong E-bước và M-bước
2: dùng D^K xây dựng mô hình ban đầu $_0$
3: **for** i = 0, 1, 2, . . . cho đến khi kết quả đảm bảo **do**
4: **for** mỗi tài liệu d D^U **do**
5: E-bước: dùng phân lớp Bayes thứ nhất xác định P$(c|d, _i)$
6: **end for**
7: **for** mỗi lớp c và từ khóa t **do**
8: M-bước: xác định $_{c,t}$ dùng công thức (*) để xây dựng mô hình i+1
9: **end for**
10: **end for**

$$\mathrm{P}(d|c) = \mathrm{P}(L = \ell_d|c) \binom{\ell_d}{\{n(d,t)\}} \prod_{t \in d} \theta_t^{n(d,t)}$$

$$\theta_{c,t} = \frac{1 + \sum_{d \in D} \mathrm{P}(c|d)\, n(d,t)}{|W| + \sum_{d \in D} \sum_{\tau} \mathrm{P}(c|d)\, n(d,\tau)} \qquad \mathrm{P}(c) = \frac{1}{|D|} \sum_{d \in D} \mathrm{P}(c|d)$$

Mô hình sinh: Thuật toán EM

- Một số vấn đề với EM
  - Phạm vi áp dụng: mô hình trộn chính xác
  - Nếu cực trị địa phương khác xa cực trị toàn cục thì khai thác dữ liệu không nhãn không hiệu quả
  - "Kết quả đảm bảo yêu cầu": đánh giá theo các độ đo hồi tưởng, chính xác, F_1...
  - Một số vấn đề khác cần lưu ý:
    - Thuật toán nhân là Bayes naive: có thể chọn thuật toán cơ bản khác
    - Chọn điểm bắt đầu bằng học tích cực

Mô hình sinh: Thuật toán khác

- Phân cụm - và - Nhãn
  - Sử dụng phân cụm cho toàn bộ ví dụ
    - cả dữ liệu có nhãn và không có nhãn
    - dành tập D_{test} để đánh giá
  - Độ chính xác phân cụm cao
    - Mô hình phân cụm phù hợp dữ liệu
    - Nhãn cụm (nhãn dữ liệu có nhãn) làm nhãn dữ liệu khác
- Phương pháp nhân Fisher cho học phân biệt
  - Phương pháp nhân là một phương pháp điển hình
  - Nhân là gốc của mô hình sinh
  - Các ví dụ có nhãn được chuyển đổi thành vector Fisher để phân lớp

Self-Training

- Giới thiệu
  - Là kỹ thuật phổ biến trong SSL
    - EM địa phương là dạng đặc biệt của seft-training
- Nội dung

  Gọi

  L : Tập các dữ liệu gán nhãn.

  U : Tập các dữ liệu chưa gán nhãn

  Lặp (cho đến khi U =)

  Huấn luyện bộ phân lớp giám sát h trên tập L

  Sử dụng h để phân lớp dữ liệu trong tập U

  Tìm tập con U’ U có độ tin cậy cao nhất:

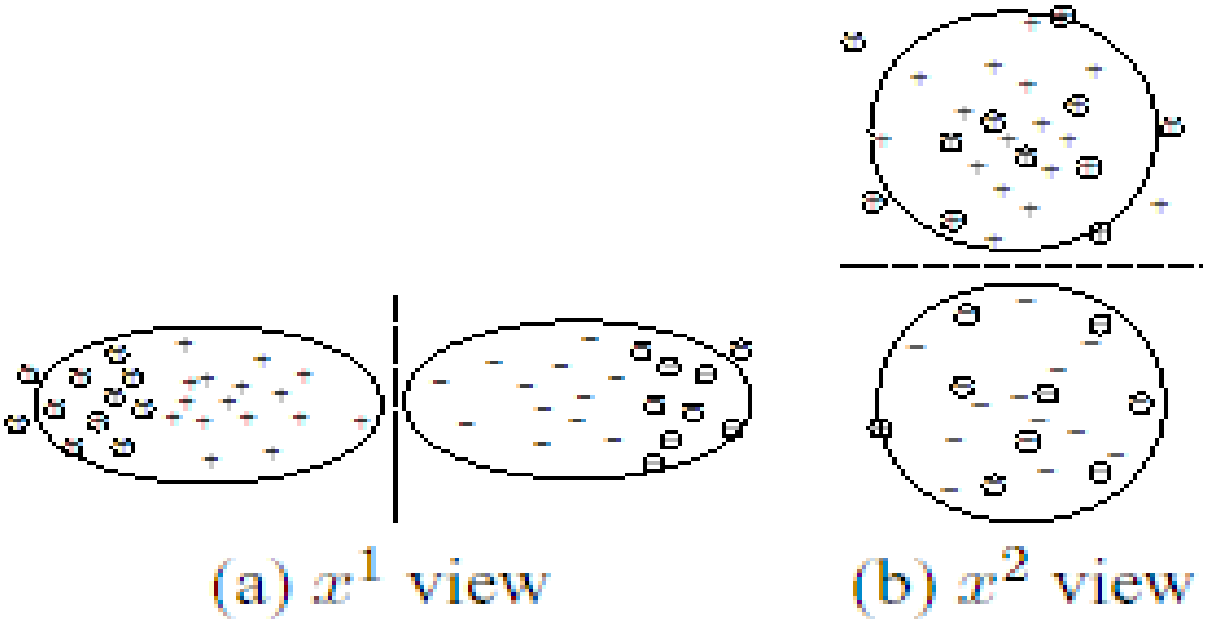
  L + U’ L

  U – U’ U

  Vấn đề tập U' có "độ tin cậy cao nhất"
- Thủ tục "bootstrapping"
- Thường được áp dụng cho các bài toán NLP

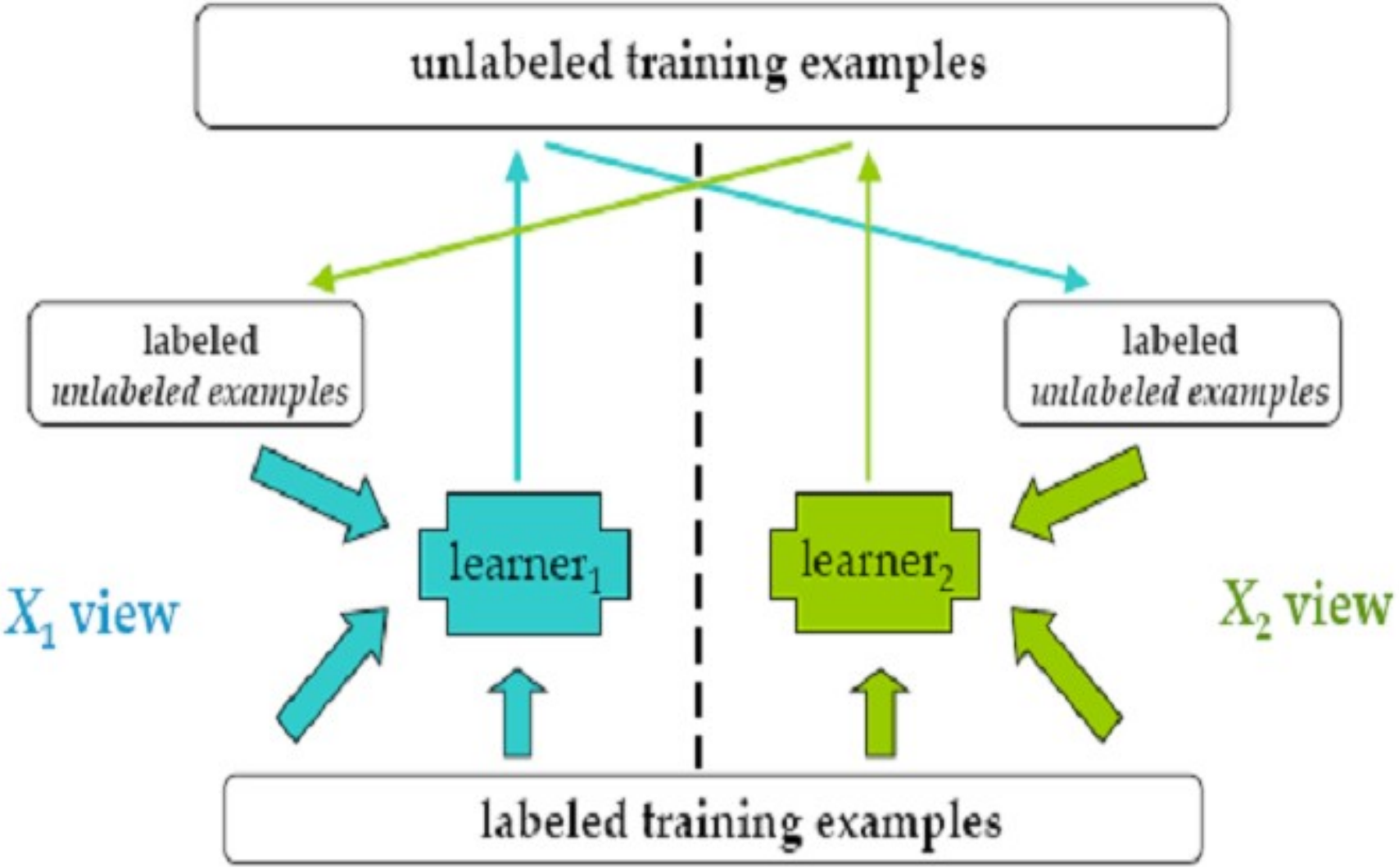
Co-Training

- Tư tưởng
  - Một dữ liệu có hai khung nhìn
  - Ví dụ, các trang web
    - Nội dung văn bản
    - Tiêu đề văn bản



Co-Training

- Mô hình thuật toán



Co-Training

- Điều kiện dừng
  - hoặc tập dữ liệu chưa gán nhãn là rỗng
  - hoặc số vòng lặp đạt tới ngưỡng được xác định trước
- Một số lưu ý
  - Tập dữ liệu gán nhãn có ảnh hưởng lớn đến co-training
    - Quá ít: không hỗ trợ co-training
    - Quá nhiều: không thu lợi từ co-training
  - Cơ sở tăng hiệu quả co-training: thiết lập tham số
    - Kích cỡ tập dữ liệu gán nhãn
    - Kích cỡ tập dữ liệu chưa gán nhãn
    - Số các mẫu thêm vào sau mỗi vòng lặp
  - Bộ phân lớp thành phần rất quan trọng

Chặn thay đổi miền dày đặc

- Transductive SVMs (S3VMs)
  - Phương pháp phân biệt làm việc trên p(y|x) trực tiếp
  - Khi p(x) và p(y|x) không tương thích đưa p(x) ra khỏi miền dầy đặc
- Quá trình Gauxơ

Mô hình đồ thị

- Biểu diễn dữ liệu chưa mô tả hết ánh xạ gán nhãn trên dữ liệu (chẳng hạn, nghịch lý “hiệu quả như nhau” trong biểu diễn văn bản)
- Ánh xạ gán nhãn có liên quan mô hình dữ liệu (mô hình / đặc trưng/ nhân / hàm tương tự) mô hình đã có theo tự nhiên hoặc giả thiết dữ liệu tuân theo.