Trường Đại học Khoa học Tự nhiên
Khoa Công nghệ Thông tin

TÀI LIỆU LÝ THUYẾT KTDL & UD

# Phân Lớp Dữ Liệu (P2)
# Classification

**Giảng viên: ThS. Lê Ngọc Thành**
**Email: lnthanh@fit.hcmus.edu.vn**

Summer 2012

# Nội dung

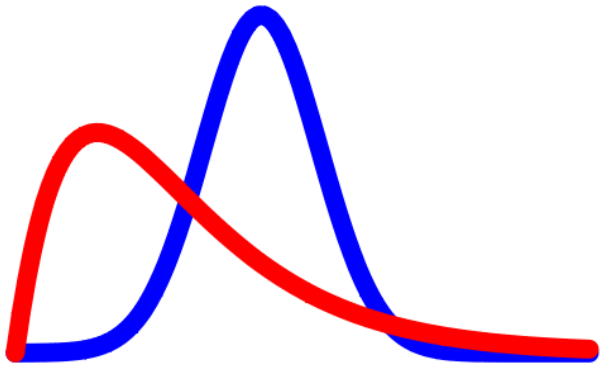
- **Phân lớp dựa trên thống kê**
  - Phân lớp Bayes
  - Định lý Bayes
  - Phân lớp Naïve Bayes
  - Làm trơn Laplace
- Phân lớp dựa trên thể hiện
- Đánh giá phương pháp phân lớp

# Phân lớp Bayes

- Phân lớp Bayes là một bộ phân lớp thống kê thực hiện việc dự đoán xác suất mà lớp sẽ thuộc về dựa trên giá trị của các thuộc tính biết trước (hay gọi là xác suất hậu nghiệm):

$$\Pr(C = c_j \mid a_1 = v_1, \dots, a_{|a|} = v_{|a|})$$

- Mẫu thuộc về lớp $c_j$ khi xác suất trên đạt cực đại



# Bộ phân lớp Naïve Bayes

- Bộ phân lớp Bayes đơn giản, Naïve Bayes, có khả năng thực thi hiệu quả so với cây quyết định hay mạng neuron
- Có thể chạy nhanh trên cơ sở dữ liệu lớn với độ chính xác cao
- Naïve Bayes xem các thuộc tính xảy ra độc lập với nhau (lí do tại sao gọi là “naïve”)

# Định lý Bayes

- Gọi **X** là mẫu dữ liệu chưa biết nhãn
- **$C_i$** là giả thuyết X thuộc về phân lớp $C_i$
- Việc phân lớp là quá trình xác định **$P(C_i|X)$** (xác suất hậu nghiệm), xác suất mà giả thuyết đúng với mẫu dữ liệu X cho trước.
  - Ví dụ: P(class=No | outlook=sunny,windy=true,…)
- **$P(C_i)$** là xác suất tiền nghiệm của lớp, có thể ước lượng từ dữ liệu huấn luyện
  - Ví dụ: xác suất X sẽ mua máy tính mà không quan tâm đến bất kì thông tin nào như tuổi, thu nhập,…
- **$P(X)$** là xác suất mẫu dữ liệu được quan sát
- **$P(X|C_i)$** là khả năng quan sát mẫu X khi cho trước giả thuyết về phân lớp
  - Ví dụ: Cho trước X sẽ "*mua*" máy tính, khả năng mà X ở độ tuổi 31…40, thu nhập trung bình, …

# Định lý Bayes (tt)

- Định lý Bayes:

$$P(C_i|X) = \frac{P(X|C_i)P(Ci)}{P(X)}$$

- Dự đoán X thuộc về lớp $C_i$ khi và chỉ khi $P(C_i|X)$ là cao nhất trong số $P(C_m|X)$ của tất cả m lớp.
- Thực tế, tính toán này đòi hỏi nhiều xác suất khởi tạo và tốn chi phí thực thi đáng kể

# Nhận xét định lý Bayes

- Do P(X) là hằng số cho mọi lớp nên chỉ cần tìm cực đại của:

$$P(C_i|X) = \frac{P(X|C_i)P(Ci)}{\cancel{P(X)}}$$

→ $$P(C_i|X) = P(X|C_i)P(C_i)$$

# Naïve Bayes

- Naïve Bayes giả sử giá trị của mọi thuộc tính đều độc lập nên:

$$P(X|C_i) = \prod_{k=1}^{n} P(xk|\,C_i)$$

$$= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(xn|Ci)$$

- Việc tính trở nên đơn giản hơn:
  - Nếu giá trị thuộc tính là rời rạc, $P(x_k|C_i)$ là tỉ lệ mẫu $x_k$ trong các dòng phân lớp $C_i$
  - Nếu giá trị thuộc tính là liên tục, $P(x_k|C_i)$ thường được tính theo phân phối Gauss

# Ví dụ Naïve Bayes (1/2)

| age | income | student | credit_rating | buy? |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

** Các lớp:*

C1:buys_computer = ‘yes’

C2:buys_computer = ‘no’

** Dữ liệu cần xác định lớp:*

**X** = (age <=30,

Income = medium,

Student = yes,

Credit_rating = fair)

**Cần tính* $\mathbf{P(C_i|X)}$*:*

- $P(C_i)$?
- $P(X|C_i)$?
  - $P(x_{age}|C_i)$, $P(x_{income}|C_i)$, $P(x_{student}|C_i)$, $P(x_{Credit\_rating}|C_i)$?

# Ví dụ Naïve Bayes (2/2)

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Tính $P(X|C_i)$ cho mỗi lớp
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

**$P(X|C_i)$ :** P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

**$P(X|C_i)*P(C_i)$ :** P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

**Vậy X thuộc về lớp ("buys_computer = yes")**

# Sửa lỗi Laplace (1/2)

- Naïve Bayes đòi hỏi mỗi xác suất điều kiện phải ≠ 0.

$$P(X|C_i) = \prod_{k=1}^{n} P(xk|\,C_i)$$

- Ví dụ: dữ liệu huấn luyện ở phân lớp $C_i$ gồm 1000 dòng, xét trên thuộc tính ***income***, nó có thể chứa 0 dòng *low*, 990 dòng *medium* và 10 dòng *high*
- Sửa lỗi Laplace bằng cách bổ sung vào mỗi trường hợp 1 dữ liệu "ảo":
  - P(income=low|$C_i$) = 1/1003
  - P(income=medium|$C_i$) = 991/1003
  - P(income=high|$C_i$) = 11/1003

# Sửa lỗi Laplace (2/2)

- Tổng quát

$$P(C_i) = \frac{\left|C_{i,D}\right| + 1}{|D| + m}$$

$$P(x_k | C_i) = \frac{\#C_{i,D}\{x_k\} + 1}{\left|C_{i,D}\right| + r}$$

*m: số lớp*

*r: số giá trị của thuộc tính*

# Bài tập 1

| Outlook | Temperature | Humidity | Windy | Play? |
|---|---|---|---|---|
| sunny | hot | high | weak | No |
| sunny | hot | high | strong | No |
| overcast | hot | high | weak | Yes |
| rain | mild | High | weak | Yes |
| rain | cool | Normal | weak | Yes |
| rain | cool | normal | strong | No |
| overcast | cool | normal | strong | Yes |
| sunny | mild | high | weak | No |
| sunny | cool | normal | weak | Yes |
| rain | mild | normal | weak | Yes |
| sunny | mild | normal | strong | Yes |
| overcast | mild | high | strong | Yes |
| overcast | hot | normal | weak | Yes |
| rain | mild | high | strong | No |

*Dùng Naïve Bayes để dự đoán lớp cho mẫu:*

*$X_1$=<Outlook=sunny, Temp=cool, Humidity=high, Windy=strong>*

*$X_2$=<Outlook=overcast, Temp=cool, Humidity=high, Windy=strong>*

# Bài tập 1 – Đáp án

| |
|---|
| $P(C_1) = 9/14=0.643$ |
| $P(C_2) = 5/14=0.357$ |

| **Outlook** | |
|---|---|
| **P(sunny \| y) = 2/9** | **P(sunny \| n) = 3/5** |
| **P(overcast \| y) = 4/9** | **P(overcast \| n) = 0** |
| **P(rain \| y) = 3/9** | **P(rain \| n) = 2/5** |
| **Temperature** | |
| **P(hot \| y) = 2/9** | **P(hot \| n) = 2/5** |
| **P(mild \| y) = 4/9** | **P(mild \| n) = 2/5** |
| **P(cool \| y) = 3/9** | **P(cool \| n) = 1/5** |
| **Humidity** | |
| **P(high \| y) = 3/9** | **P(high \| n) = 4/5** |
| **P(normal \| y) = 6/9** | **P(normal \| n) = 1/5** |
| **Windy** | |
| **P(strong \| y) = 3/9** | **P(strong \| n) = 3/5** |
| **P(weak \| y) = 6/9** | **P(weak \| n) = 2/5** |

# Bài tập 1 – Đáp án (tt)

- $P(C_1|X_1) = P(X_1|C_1) * P(C_1)$
  $=P(C_1)*P(sunny|y)*P(cool|y)*P(high|y)* P(strong|y)$
  $=0.005$
- $P(C_2|X_1) = P(X_1|C_2)* P(C_2)$
  $=P(C_2)*P(sunny|n)*P(cool|n)*P(high|n)*P(strong|n)$
  $=0.021$

***Do $P(C_2|X_1) > P(C_1|X_1) \rightarrow X_1$ thuộc lớp $C_2$ ("no")***

- $P(C_1|X_2) = P(X_2|C_1)* P(C_1)$
  $=P(C_1)*P(ovecast|y)*P(cool|y)*P(high|y)*P(strong|y)$
  $=0.011$
- $P(C_2|X_2) = P(X_2|C_2)* P(C_2)$
  $=P(C_2)*P(ovecast|n)*P(cool|n)*P(high|n)*P(strong|n)$
  $=0$

***Xác suất $P(C_2|X_2) = 0$ nên để chính xác ta áp dụng kĩ thuật làm trơn Laplace***

# Bài tập 1 – Làm trơn Laplace

| |
|---|
| $P(C_1) = (9+1)/(14+2) = 10/16$ |
| $P(C_2) = (5+1)/(14+2) = 6/16$ |

| | |
|---|---|
| **Outlook** | |
| **P(sunny \| y) = 3/12** | **P(sunny \| n) = 4/8** |
| **P(overcast \| y) = 5/12** | **P(overcast \| n) = 1/8** |
| **P(rain \| y) = 4/12** | **P(rain \| n) = 3/8** |
| **Temperature** | |
| **P(hot \| y) = 3/12** | **P(hot \| n) = 3/8** |
| **P(mild \| y) = 5/12** | **P(mild \| n) = 3/8** |
| **P(cool \| y) = 4/12** | **P(cool \| n) = 2/8** |
| **Humidity** | |
| **P(high \| y) = 4/11** | **P(high \| n) = 5/7** |
| **P(normal \| y) = 7/11** | **P(normal \| n) = 2/7** |
| **Windy** | |
| **P(strong \| y) = 4/11** | **P(strong \| n) = 4/7** |
| **P(weak \| y) = 7/11** | **P(weak \| n) = 3/7** |

# Bài tập 1 – Làm trơn Laplace (tt)

- $P(C_1|X_2) = P(X_2|C_1)* P(C_1)$
  $=P(C_1)*P(ovecast|y)*P(cool|y)*P(high|y)*P(strong|y)$
  $=0.011$
- $P(C_2|X_2) = P(X_2|C_2)* P(C_2)$
  $=P(C_2)*P(ovecast|n)*P(cool|n)*P(high|n)*P(strong|n)$
  $=0.005$

***Do $P(C_1|X_2) > P(C_2|X_2) \rightarrow X_2$ thuộc lớp $C_1$ ("yes")***

# Nhận xét Naïve Bayes

- Ưu điểm:
  - Dễ dàng thực thi
  - Đạt được kết quả khá tốt trong hầu hết các trường hợp
- Khuyết điểm:
  - Việc giả sử các thuộc tính độc lập có thể sẽ làm giảm độ chính xác vì thực tế có thể tồn tại sự phụ thuộc giữa chúng.
    - Ví dụ: bệnh viện-bệnh nhân-tuổi-lịch sử bệnh trong gia đình-…

# Nội dung

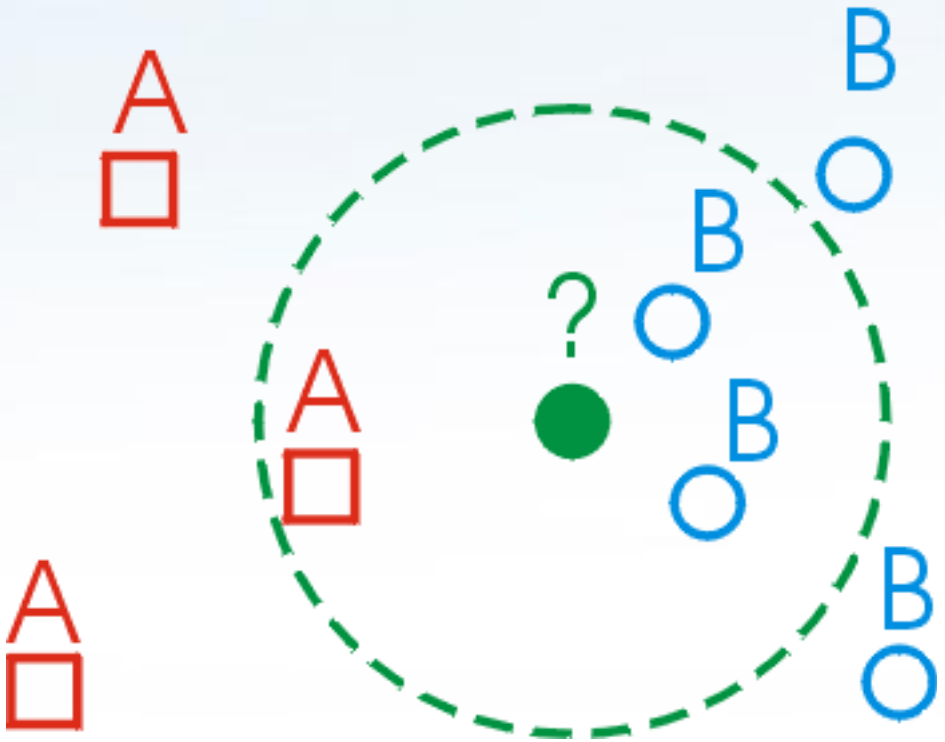
- Phân lớp dựa trên thống kê
- **Phân lớp dựa trên thể hiện**
  - Khái niệm
  - Thuật toán k-NN
  - Tính khoảng cách
  - Chuẩn hóa dữ liệu
  - Giá trị k
- Đánh giá phương pháp phân lớp

# Phân lớp dựa trên thể hiện

- Phương pháp phân lớp dựa trên thể hiện (Instance-based):
  - Lưu trữ các mẫu huấn luyện và chờ cho đến khi có yêu cầu phân lớp một mẫu mới.
- Các phương pháp:
  - Thuật toán k-láng giềng gần nhất (k-NN)
  - Hồi qui với trọng số cục bộ (Locally weighted regression)
  - Suy luận dựa trên trường hợp (Case-based reasoning)

# k-Nearest Neighbor (k-NN)

- Một mẫu mới được gán vào lớp có nhiều mẫu trong số k mẫu gần với nó nhất



# Thuật toán k-NN

**Algorithm** kNN($D$, $d$, $k$)

1 Compute the distance between $d$ and every example in $D$;

2 Choose the $k$ examples in $D$ that are nearest to $d$, denote the set by $P$ ($\subseteq D$);

3 Assign $d$ the class that is the most frequent class in $P$ (or the majority class);

- Tính khoảng cách giữa d và tất cả các mẫu trong tập huấn luyện
- Chọn k mẫu gần nhất với d trong tập huấn luyện
- Gán d vào lớp có nhiều mẫu nhất trong số k mẫu láng giềng đó (hoặc d nhận giá trị trung bình của k mẫu)

# Tính khoảng cách

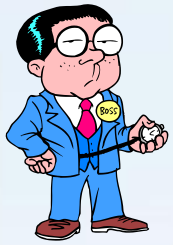
- Khoảng cách có thể dựa trên nhiều độ đo như Euclide, Cosin,…
- Ví dụ, khoảng cách Euclide giữa hai mẫu $X = (x_1,\ldots,x_n)$ và $Y = (y_1,\ldots,y_n)$:

$$D(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Khi thực hiện so sánh, có thể bỏ qua căn bậc 2

# Ví dụ tính khoảng cách

- ***Khoảng cách giữa John và Rachel***



**John:**
Age=35
Income=95K
No. of credit cards=3

**Rachel:**
Age=41
Income=215K
No. of credit cards=2

$$D(John, Rachel) = \sqrt{[(35-41)^2 + (95K-215K)^2 + (3-2)^2]}$$

- Các thuộc tính **có giá trị lớn** sẽ ảnh hưởng nhiều đến khoảng cách giữa các đối tượng (VD: thuộc tính income)
- Các thuộc tính có **miền giá trị khác nhau**

→ ***Cần chuẩn hóa giá trị thuộc tính***

# Chuẩn hóa dữ liệu

- Cần phải chuẩn hoá dữ liệu : ánh xạ các giá trị vào đoạn [0,1] theo công thức :

$$a_i = \frac{v_i - \min v_i}{\max v_i - \min v_i}$$

*với : $v_i$ là giá trị thực tế của thuộc tính i*

*$a_i$ là giá trị của thuộc tính đã chuẩn hóa*

- Ví dụ: nhiệt độ có các giá trị 20, 24, 26, 27, 30
  Giá trị chuẩn hóa của 26 là:

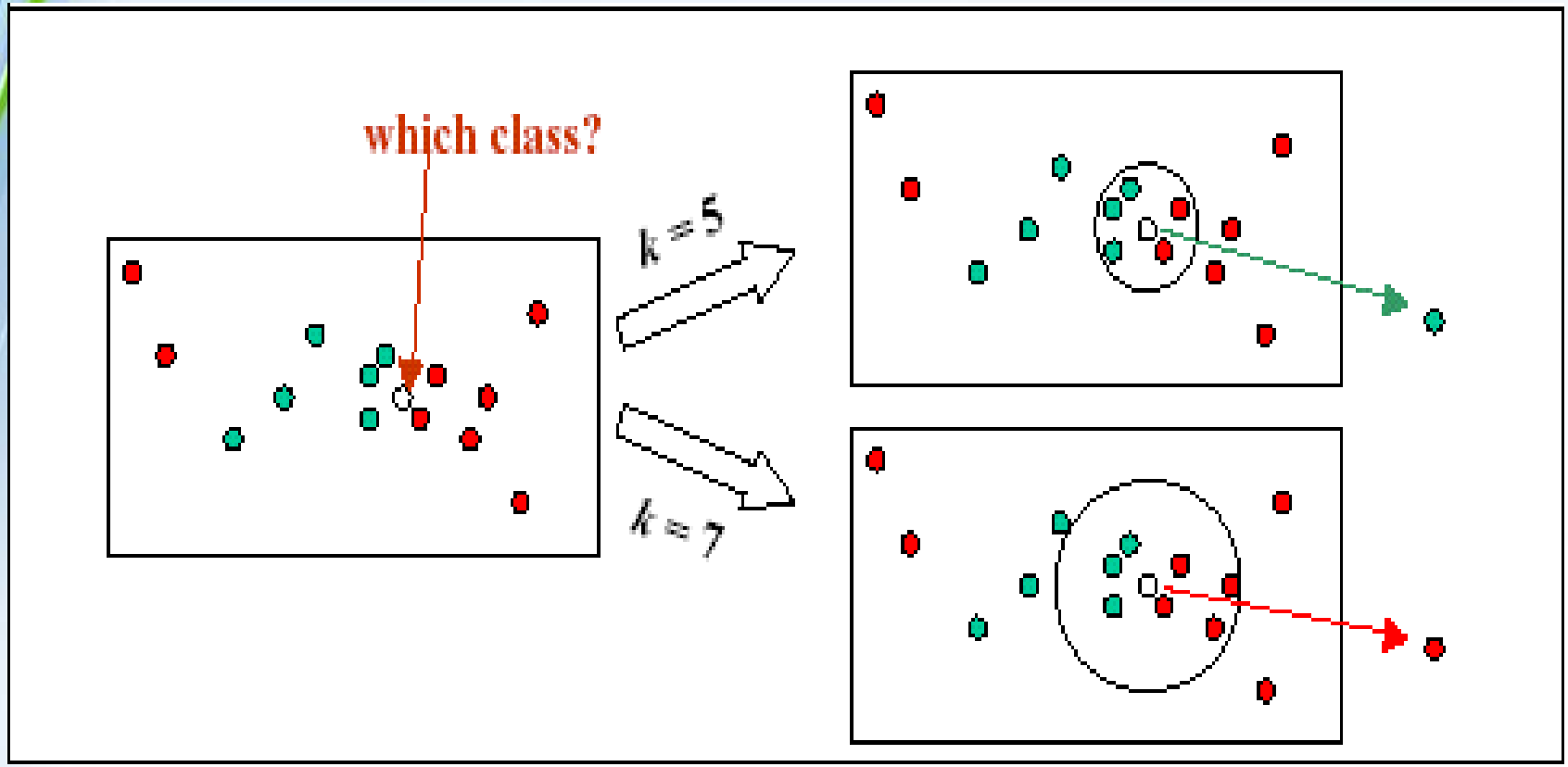
$$\frac{26 - 20}{30 - 20} = 0.6$$

# Nhận xét k-NN

- ***Ưu điểm:***
  - Dễ sử dụng và cài đặt.
  - Xử lý tốt với dữ liệu nhiễu.
- ***Khuyết điểm:***
  - Cần lưu tất cả các mẫu
  - Cần nhiều thời gian để xác định lớp cho một mẫu mới (cần tính và so sánh khoảng cách đến tất cả các mẫu)
  - Phụ thuộc vào giá trị k do người dùng lựa chọn
    - *Nếu k quá nhỏ, nhạy cảm với nhiễu*
    - *Nếu k quá lớn, vùng lân cận có thể chứa các điểm của lớp khác*
  - Thuộc tính phi số?

# Giá trị k

- Phụ thuộc vào giá trị k do người dùng lựa chọn



# Bài tập 2

| Customer | Age | Income (K) | No. cards | Response |
|---|---|---|---|---|
| Lâm | 35 | 35 | 3 | Yes |
| Hưng | 22 | 50 | 2 | No |
| Mai | 28 | 40 | 1 | Yes |
| Lan | 45 | 100 | 2 | No |
| Thủy | 20 | 30 | 3 | Yes |
| Tuấn | 34 | 55 | 2 | No |
| Minh | 63 | 200 | 1 | No |
| Vân | 55 | 140 | 2 | No |
| Thiện | 59 | 170 | 1 | No |
| Ngọc | 25 | 40 | 4 | Yes |
| **Vinh** | **39** | **43** | **1** | **?** |

Sử dụng thuật toán k-NN với **k = 5** để xác định lớp cho "Vinh"

# Nội dung

- Phân lớp dựa trên thống kê
- Phân lớp dựa trên thể hiện
- **Đánh giá phương pháp phân lớp**
  - Các khía cạnh đánh giá
  - Các độ đo chính xác
  - Phương pháp ước lượng độ chính xác
  - So sánh các bộ phân lớp
  - Các kĩ thuật cải thiện độ chính xác

# Khía cạnh đánh giá

- Độ chính xác: khả năng dự đoán của bộ phân lớp
- Tính hiệu quả:
  - Chi phí phát sinh mô hình
  - Chi phí sử dụng mô hình
- Tính mạnh mẽ: khả năng nắm giữ nhiễu hay giá trị bị thiếu
- Tính mở rộng: hiệu quả với dữ liệu lớn
- Tính dễ hiểu
- Các tính chất khác: kích thước cây, số lượng luật, chất lượng luật…

# Độ chính xác

- Tập dữ liệu được chia thành 2 phần hoàn toàn độc lập nhau.
  - Tập huấn luyện (training set – dùng để học mô hình)
  - Tập kiểm chứng (test set)
- Những độ đo để đánh giá độ chính xác: ma trận sai số, tỉ lệ lỗi, …
- Các phương pháp ước lượng độ chính xác của bộ phân lớp:
  - Holdout method, Random Subsampling
  - Cross-validation
  - Bootstrap

# Một số khái niệm về độ đo (1/2)

- Qui ước:
  - Mẫu dương (Positive tuples) là các mẫu thuộc về một lớp chính đang quan tâm
  - Mẫu âm (Negative tuples) là các mẫu thuộc về các lớp còn lại
- Gọi **P** là số mẫu dương, **N** là số mẫu âm thực sự có trong tập test.
- **TP** (True Positives): số mẫu dương được phân lớp đúng
- **TN** (True Negatives): số mẫu âm được phân lớp đúng

# Một số khái niệm về độ đo (2/2)

- **FP** (False Positives): số mẫu âm bị phân lớp sai thành dương
- **FN** (False Negatives): số mẫu dương bị phân lớp sai thành âm.

| | | Predicted class | | |
|---|---|---|---|---|
| | | + | − | Total |
| Actual class | + | *TP* | *FN* | *P* |
| | − | *FP* | *TN* | *N* |
| | Total | *P′* | *N′* | *P* + *N* |

***Ma trận sai số (Confusion Matrix)***

# Ví dụ Ma trận sai số

| A\P | C | ¬C | |
|---|---|---|---|
| C | TP | FN | P |
| ¬C | FP | TN | N |
| | P’ | N’ | All |

- Dữ liệu cửa hàng bán máy tính, mẫu dương P là các mẫu buys_computer = yes

| Actual class\Predicted class | buy_computer = yes | buy_computer = no | Total |
|---|---|---|---|
| buy_computer = yes | **6954** | **46** | 7000 |
| buy_computer = no | **412** | **2588** | 3000 |
| Total | 7366 | 2634 | 10000 |

- Xác định TP, TN, FP, FN?
- Trường hợp lý tưởng, đường chéo phụ nên là 0 hay xấp xỉ 0

# Độ đo Accuracy

- Độ Accuracy: tỉ lệ mẫu trong tập test được phân lớp đúng

$$accuracy = \frac{TP + TN}{P + N}$$

- Ví dụ:

accuracy = (6954 + 2588)/ 10000 = 0.95

| Actual class\Predicted class | buy_computer = yes | buy_computer = no | Total |
|---|---|---|---|
| buy_computer = yes | **6954** | **46** | 7000 |
| buy_computer = no | **412** | **2588** | 3000 |
| Total | 7366 | 2634 | 10000 |

# Độ đo lỗi

- Tỉ lệ lỗi: tỉ lệ mẫu bị phân lớp sai trong tập test (bằng 1 - accuracy)

$$error\ rate = \frac{FP + FN}{P + N}$$

- Ví dụ:

  error rate = (412 + 46)/ 10000 = 0.05

| Actual class\Predicted class | buy_computer = yes | buy_computer = no | Total |
|---|---|---|---|
| buy_computer = yes | **6954** | **46** | 7000 |
| buy_computer = no | **412** | **2588** | 3000 |
| Total | 7366 | 2634 | 10000 |

# Vấn đề mất cân bằng lớp (1/2)

- Lớp quan tâm có thể hiếm xảy ra so với các lớp khác
- Ví dụ:
  - Trong ứng dụng phát hiện lừa đảo, lớp quan tâm là “fraud” nhưng lại xảy ra ít hơn nhiều so với các mẫu thuộc lớp “nonfraudulant”.
  - Trong dữ liệu bệnh án, giả sử thuộc tính phân lớp là “cancer”, tỉ lệ mẫu được gán nhãn “yes” (lớp quan tâm) thấp hơn nhiều so với nhãn “no”.

# Vấn đề mất cân bằng lớp (2/2)

- Bộ phân lớp có thể đoán đúng ở những mẫu âm nhưng có thể phân lớp sai hoàn toàn ở những mẫu dương
- Ví dụ:
  - Một bộ phân lớp có độ accuracy 99% cho thấy khả năng đoán đúng rất cao. Tuy nhiên nếu 1% sai còn lại thuộc về mẫu dương thì đạt 99% trở nên vô nghĩa

→ Giải quyết bằng độ đo *sensitivity* và *specificity*

# Độ đo Sensitivity và Specificity

- Sensitivity: tỉ lệ nhận dạng mẫu dương đúng

$$sensitivity = \frac{TP}{P}$$

- Specificity: tỉ lệ nhận dạng mẫu âm đúng

$$specificty = \frac{TN}{N}$$

# Ví dụ Sensitivity và Specificity

| *Classes* | *yes* | *no* | *Total* | *Recognition (%)* |
|---|---|---|---|---|
| *yes* | **90** | **210** | 300 | 30.00 |
| *no* | **140** | **9560** | 9700 | 98.56 |
| Total | 230 | 9770 | 10,000 | 96.40 |

- Mặc dù bộ phân lớp có độ chính xác cao 96.50%. Tuy nhiên khả năng đánh nhãn đúng lớp dương khá thấp vì độ sensitivity thấp.

# Độ đo Precision và Recall

- Precision: độ đo sự chính xác – là tỉ lệ mẫu mà bộ phân lớp gán nhãn là dương thì thực sự là dương.

$$precision = \frac{TP}{TP + FP}$$

- Recall: độ đo sự toàn vẹn – là tỉ lệ mẫu dương mà bộ phân lớp đã gán nhãn được.

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P}$$

*Giống với độ đo sensitivity*

# Ví dụ Precision và Recall

| *Classes* | *yes* | *no* | *Total* |
|---|---|---|---|
| *yes* | **90** | **210** | 300 |
| *no* | **140** | **9560** | 9700 |
| Total | 230 | 9770 | 10,000 |

- Precision(yes) = 90/230 = 39.13%
- Recall(yes) = 90/300 = 30.00%

# Nhận xét Precision và Recall

- Precision cao nhất là 1.0:
  - Thể hiện mỗi mẫu mà bộ phân lớp đánh nhãn thuộc về lớp dương đều thực sự là dương.
  - Không thể hiện số mẫu dương bị phân lớp sai
- Recall cao nhất là 1.0:
  - Thể hiện mọi mẫu dương đều được đánh nhãn đúng.
  - Không thể hiện bao nhiêu mẫu khác bị đánh nhãn sai thuộc về lớp dương

# F-Score

- Độ đo F: sự kết hợp precision và recall

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- Độ đo F được xem là trung bình điều hòa (harmonic mean) giữa precision và recall
- Đánh trọng bằng nhau giữa precision và recall (β=1)
- Nếu muốn xem trọng một trong hai độ đo đó, vậy? (β=2, β=0.5)

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

# Ví dụ F-Score

| classes | buy_computer = yes | buy_computer = no | total | recognition(%) |
|---|---|---|---|---|
| buy_computer = yes | 6954 | 46 | 7000 | 99.34 |
| buy_computer = no | 412 | 2588 | 3000 | 86.27 |
| total | 7366 | 2634 | 10000 | 95.42 |

**F-measure(M-Yes)= 96.81%**

# Tóm tắt các độ đo

| A\P | C | ¬C | |
|---|---|---|---|
| C | TP | FN | P |
| ¬C | FP | TN | N |
| | P' | N' | All |

| *Measure* | *Formula* |
|---|---|
| accuracy, recognition rate | $\frac{TP+TN}{P+N}$ |
| error rate, misclassification rate | $\frac{FP+FN}{P+N}$ |
| sensitivity, true positive rate, recall | $\frac{TP}{P}$ |
| specificity, true negative rate | $\frac{TN}{N}$ |
| precision | $\frac{TP}{TP+FP}$ |
| $F$, $F_1$, $F$-score, harmonic mean of precision and recall | $\frac{2 \times precision \times recall}{precision + recall}$ |
| $F_\beta$, where $\beta$ is a non-negative real number | $\frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$ |

# Bài tập 3

Dữ liệu liên quan đến việc phân lớp khách hàng là *lừa đảo* hay *không lừa đảo* của một ngân hàng trước khi cho vay:

*Lớp dự đoán*

| *Lớp thực sự* | Lớp | Lừa đảo | Không LĐ | Tổng |
|---|---|---|---|---|
| | **Lừa đảo** | 44 | 15 | 59 |
| | **Không LĐ** | 20 | 146 | 166 |
| | **Tổng** | 64 | 161 | 225 |

a) Giả sử lớp quan tâm là *lừa đảo*. Hãy lập ma trận sai số (confusion matrix)

b) Tính các độ đo accuracy, error rate, sensitivity, specificity, precision, F-Score

c) Trong số các độ đo trên, độ đo nào là tỉ lệ nhận dạng sai, recall, tỉ lệ dương đúng, tỉ lệ âm đúng

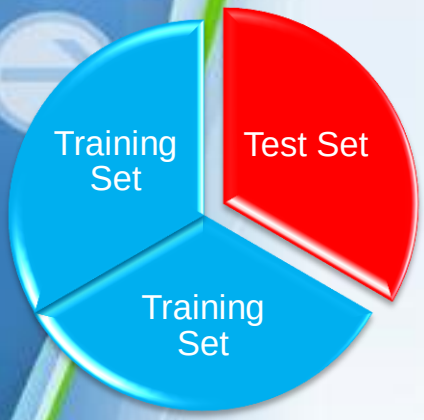
# Phương Pháp Ước Lượng Độ Chính Xác

# Độ tin cậy khi ước lượng

- Liệu các số liệu được tính từ các độ đo có đáng tin cậy và khách quan?
  - Phụ thuộc vào loại dữ liệu
  - Phụ thuộc vào cách thu thập dữ liệu
  - Phụ thuộc vào cách chia dữ liệu thành tập training và tập test.
  - …

→ Cần có phương pháp để ước lượng độ chính xác một cách tin cậy

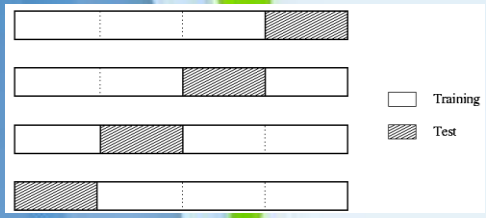
# Holdout Method




- Dữ liệu được *chia ngẫu nhiên* thành 2 phần độc lập
  - Tập huấn luyện chiếm 2/3 để rút ra mô hình
  - Tập kiểm thử chiếm 1/3 để ước lượng độ chính xác

→ *Các mẫu có thể không đại diện cho toàn bộ dữ liệu, thiếu lớp trong tập thử nghiệm*

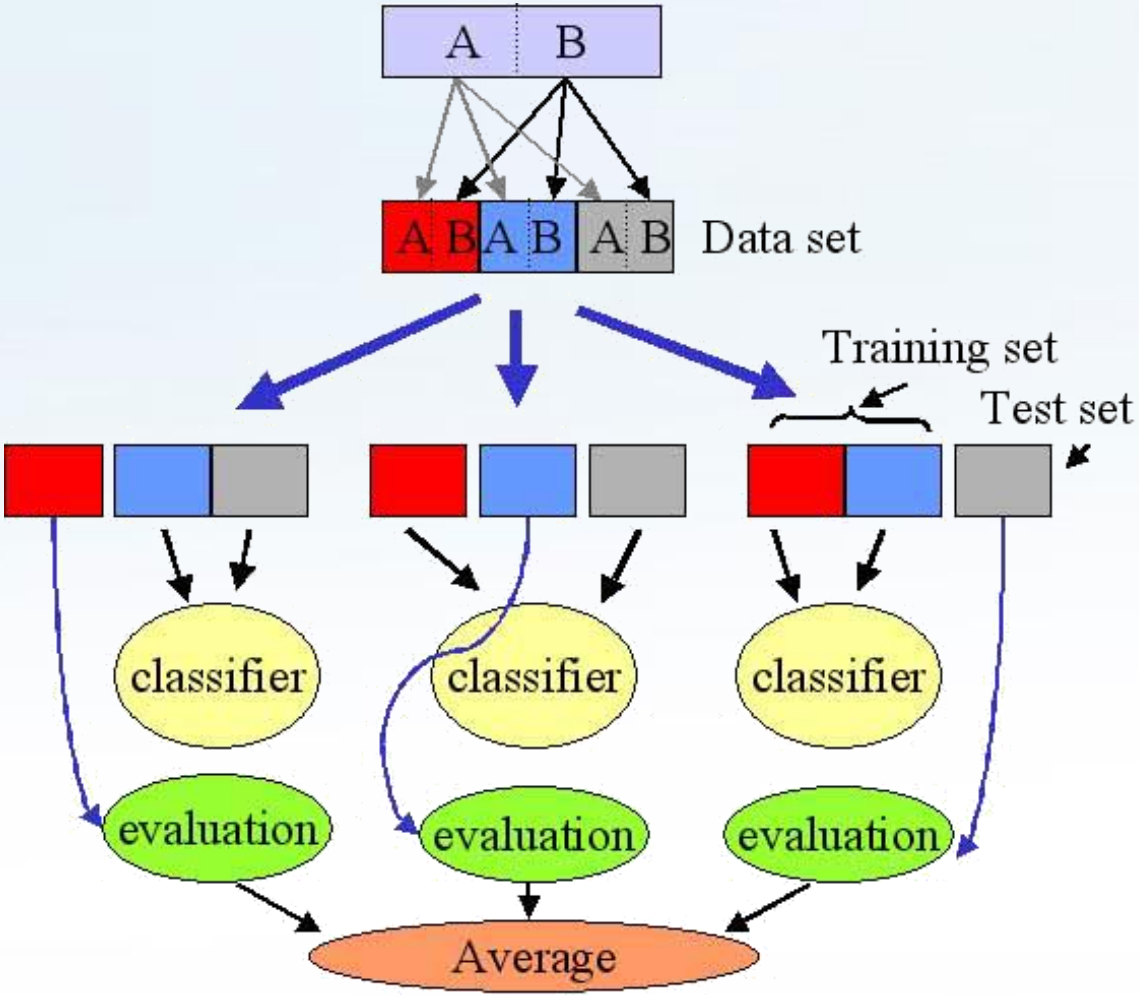
- Random sampling: là biến thể của holdout
  - Lặp lại holdout k lần, độ chính xác là trung bình cộng của các độ chính xác mỗi lần.




# k-fold Cross-Validation

- Chia ngẫu nhiên dữ liệu thành k phần độc lập và có kích thước xấp xĩ bằng nhau. D={$D_1$, $D_2$, …, $D_k$}
- Thực hiện k lần đánh giá.
- Ở lần thứ i, tập $D_i$ được sử dụng như tập test, các tập còn lại dùng để huấn luyện
- k = 10 được sử dụng phổ biến
- ***Leave-one-out***: là k fold với k là số lượng mẫu (chỉ áp dụng khi kích thước dữ liệu nhỏ)
- ***Stratified cross-validation***: phân phối lớp của các mẫu trong mỗi fold thì xấp xĩ giống dữ liệu ban đầu

# Minh họa k-fold Cross-Validation



# Bootstrap

- Thường áp dụng với tập dữ liệu nhỏ
- Mỗi lần một mẫu được chọn, nó đều có khả năng được chọn lại và thêm vào tập huấn luyện
- Có một vài phương pháp bootstrap, phổ biến là ***.632 bootstrap***
  - Tập dữ liệu kích thước *d* sẽ được lấy mẫu bootstrap *d* lần. Do đó tập huấn luyện *d* mẫu. Những mẫu nào không đưa vào tập huấn luyện sẽ dùng để test. Khoảng 63.2 % dữ liệu rơi vào tập huấn luyện và 36.8% cho tập test (vì theo xác suất $(1-1/d)^d \approx e^{-1}=0.368$)
  - Lặp lại việc lấy mẫu k lần và độ chính xác cuối:

$$Acc(M) = \frac{1}{k}\sum_{i=1}^{k}(0.632 \times Acc(M_i)_{test\_set} + 0.368 \times Acc(M_i)_{train\_set})$$

# So sánh các bộ phân lớp


# Chi phí vs. Lợi ích

- Các độ đo TP, TN, FP, FN được sử dụng để thể hiện "*giá phải trả*" và "*lợi ích*" (cost-benefit) liên quan đến mô hình phân lớp.
- "*Giá phải trả*" thường liên quan nhiều đến độ đo FN (False Negative – mẫu dương bị phân lớp sai thành mẫu âm) hơn là độ đo FP (False Positivie – mẫu âm bị phân lớp sai thành dương)
  - Ví dụ: "giá phải trả" khi dự đoán bệnh nhân bị ung thư là không ung thư cao hơn nhiều khi dự đoán bệnh nhân không ung thư là ung thư

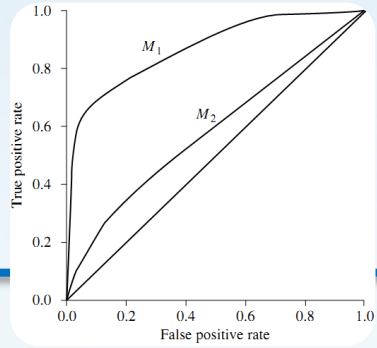
# Chi phí vs. Lợi ích (2/2)

- "Lợi ích" thường gắn liền với TP (True Positive) hơn là TN (True Negative).

→ Cần có sự đánh trọng khác nhau ở TP, TN, FP, FN

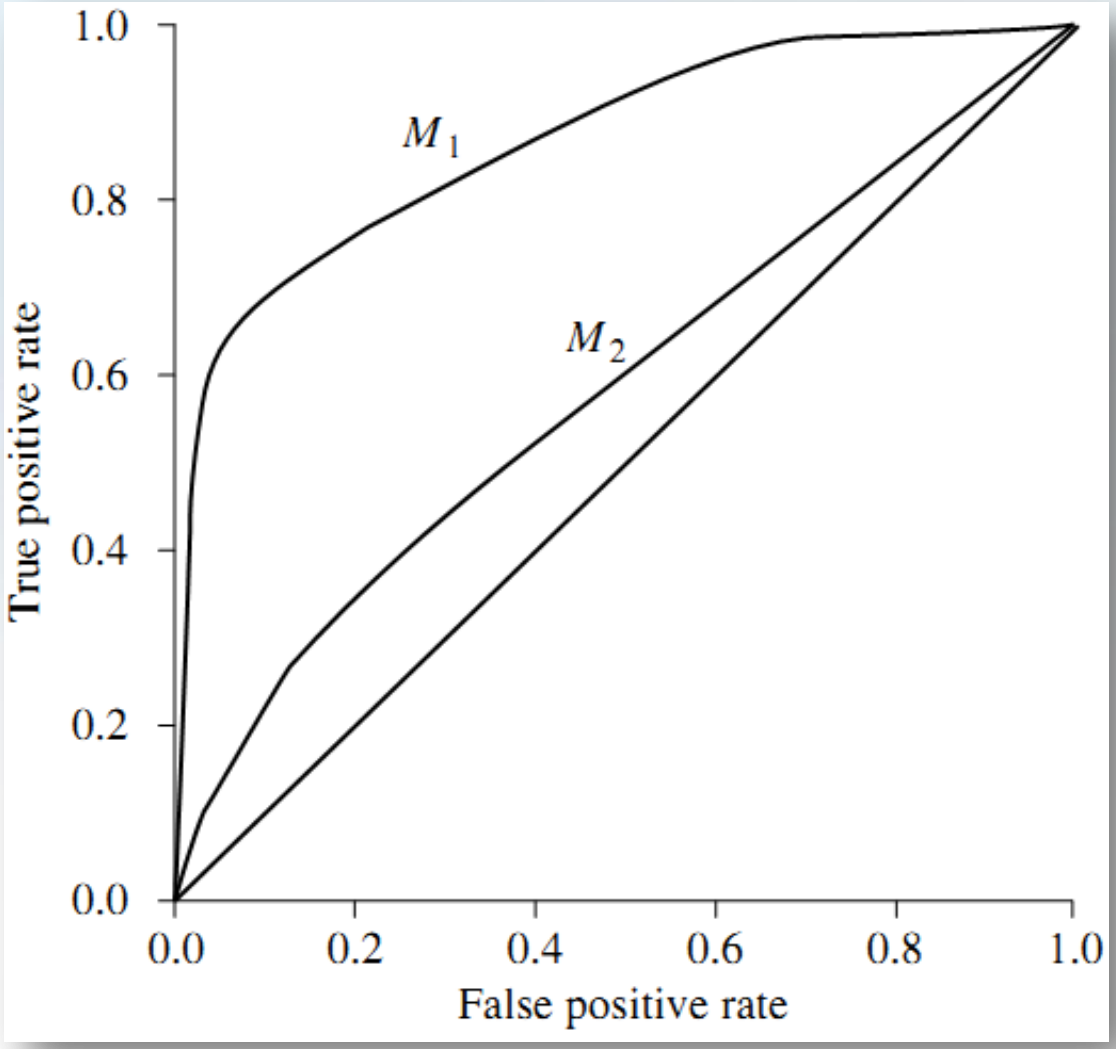
- Tuy nhiên, việc so sánh các bộ phân lớp dựa trên chi phí và lợi ích trở nên không dễ dàng.

# Đường cong ROC




- Đường cong *ROC* (Receiver Operating Characteristics): là công cụ trực quan so sánh 2 mô hình phân lớp
- Thể hiện sự tương quan giữa tỉ lệ ***TPR*** (true positive rate, sensitvity; trục đứng) và ***FPR*** (false postivive rate,1-specificity; trục ngang)
- Để vẽ được, mô hình phân lớp phải trả về xác suất mà mẫu được phân vào lớp dương.
- Xếp hạng các mẫu theo thứ tự giảm dần về xác suất.

# Ví dụ đường cong ROC



# Ví dụ xác suất phân lớp

| Tuple # | Class | Prob. |
|---|---|---|
| 1 | *P* | 0.90 |
| 2 | *P* | 0.80 |
| 3 | *N* | 0.70 |
| 4 | *P* | 0.60 |
| 5 | *P* | 0.55 |
| 6 | *N* | 0.54 |
| 7 | *N* | 0.53 |
| 8 | *N* | 0.51 |
| 9 | *P* | 0.50 |
| 10 | *N* | 0.40 |

*Lớp thực sự của mẫu*

*Số mẫu P: 5*

*Số mẫu N: 5*

*Các mẫu test được sắp xếp giảm dần dựa trên xác suất*

# Vẽ đường cong ROC

- Bắt đầu ở góc trái dưới (TPR=FPR=0)
- Lần lượt kiểm tra nhãn lớp thực sự của các mẫu theo thứ tự sắp xếp:
  - Nếu là dương (mẫu phân lớp đúng), trên đồ thị, ta di chuyển lên và vẽ một điểm.
  - Nếu là âm (mẫu phân lớp sai), ta di chuyển sang phải và vẽ một điểm

***Độ lớn như thế nào?***

# Tính toán TPR, FPR

- Ở mỗi hạng *i*, ta xem giá trị xác suất là ngưỡng để phân lớp:
  - Nếu mẫu nào lớn hơn ngưỡng thì xem là mẫu dương
  - Còn lại là mẫu âm
- Dựa trên đó để tính TPR,FPR

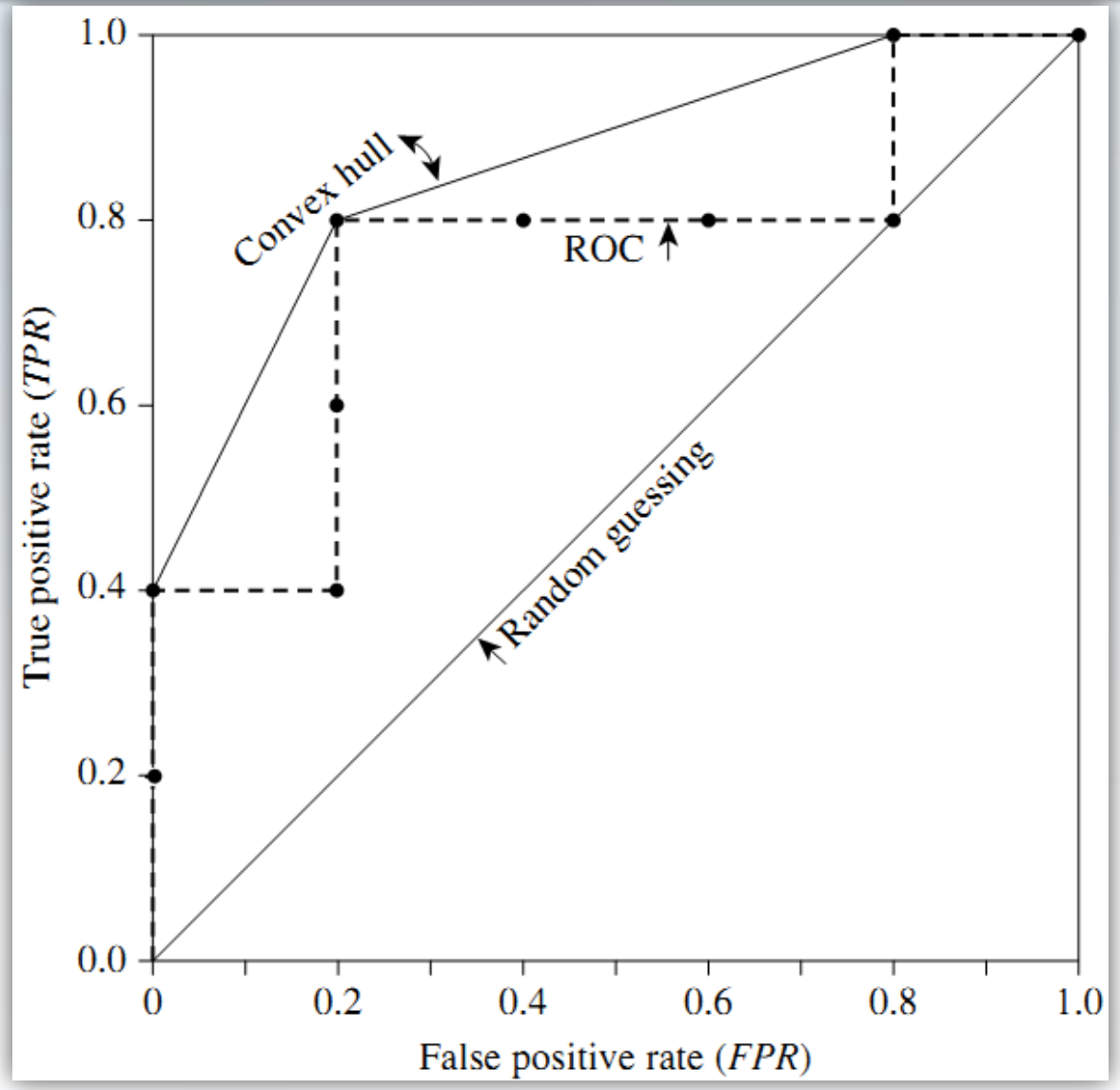
# Ví dụ tính TPR, FPR

| Tuple # | Class | Prob. | TP | FP | TN | FN | TPR | FPR |
|---|---|---|---|---|---|---|---|---|
| 1 | *P* | 0.90 | 1 | 0 | 5 | 4 | 0.2 | 0 |
| 2 | *P* | 0.80 | 2 | 0 | 5 | 3 | 0.4 | 0 |
| 3 | *N* | 0.70 | 2 | 1 | 4 | 3 | 0.4 | 0.2 |
| 4 | *P* | 0.60 | 3 | 1 | 4 | 2 | 0.6 | 0.2 |
| 5 | *P* | 0.55 | 4 | 1 | 4 | 1 | 0.8 | 0.2 |
| 6 | *N* | 0.54 | 4 | 2 | 3 | 1 | 0.8 | 0.4 |
| 7 | *N* | 0.53 | 4 | 3 | 2 | 1 | 0.8 | 0.6 |
| 8 | *N* | 0.51 | 4 | 4 | 1 | 1 | 0.8 | 0.8 |
| 9 | *P* | 0.50 | 5 | 4 | 0 | 1 | 1.0 | 0.8 |
| 10 | *N* | 0.40 | 5 | 5 | 0 | 0 | 1.0 | 1.0 |

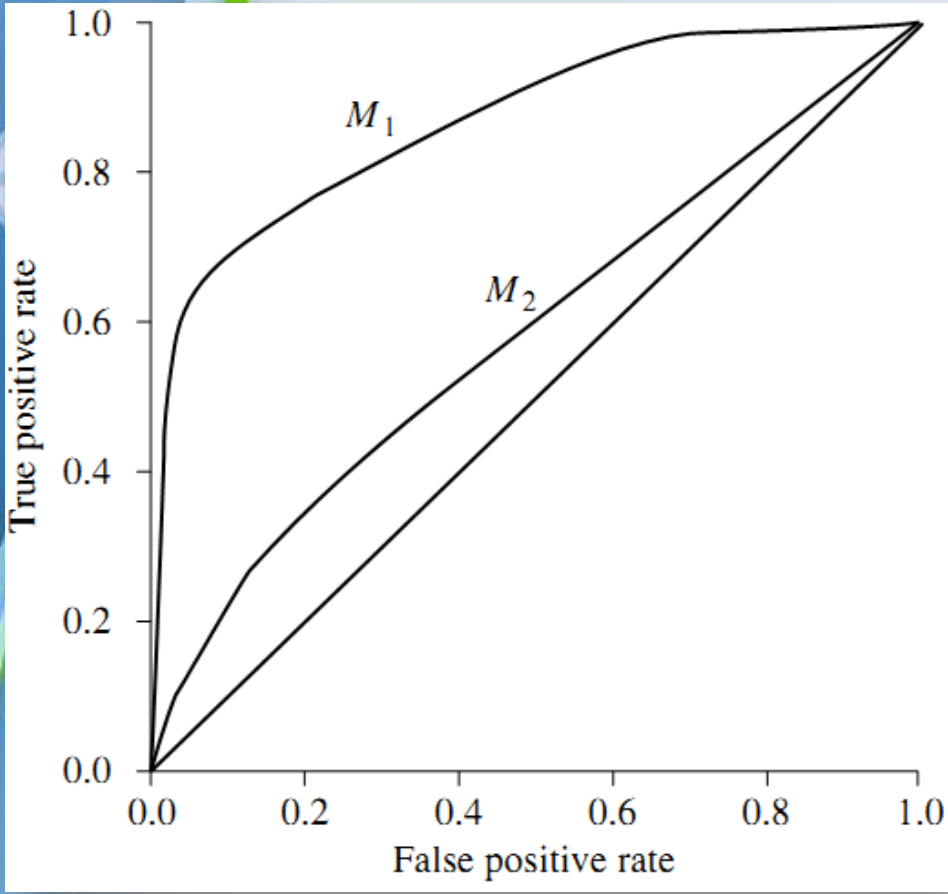
# Ví dụ đường cong ROC

http://en.wikipedia.org/wiki/Convex_hull

| Tuple # | Class | Prob. | TP | FP | TN | FN | TPR | FPR |
|---|---|---|---|---|---|---|---|---|
| 1 | P | 0.90 | 1 | 0 | 5 | 4 | 0.2 | 0 |
| 2 | P | 0.80 | 2 | 0 | 5 | 3 | 0.4 | 0 |
| 3 | N | 0.70 | 2 | 1 | 4 | 3 | 0.4 | 0.2 |
| 4 | P | 0.60 | 3 | 1 | 4 | 2 | 0.6 | 0.2 |
| 5 | P | 0.55 | 4 | 1 | 4 | 1 | 0.8 | 0.2 |
| 6 | N | 0.54 | 4 | 2 | 3 | 1 | 0.8 | 0.4 |
| 7 | N | 0.53 | 4 | 3 | 2 | 1 | 0.8 | 0.6 |
| 8 | N | 0.51 | 4 | 4 | 1 | 1 | 0.8 | 0.8 |
| 9 | P | 0.50 | 5 | 4 | 0 | 1 | 0.8 | 0.8 |
| 10 | N | 0.40 | 5 | 5 | 0 | 0 | 1.0 | 1.0 |



# So sánh sử dụng ROC




- Khu vực dưới đường cong là độ accuracy của mô hình.
- Càng gần đường chéo chính (đường baseline), độ chính xác càng thấp
- Mô hình $M_1$ tốt hơn mô hình $M_2$

# Bài tập 4

- Cho các mẫu dữ liệu được phân lớp xác suất theo thứ tự giảm dần:

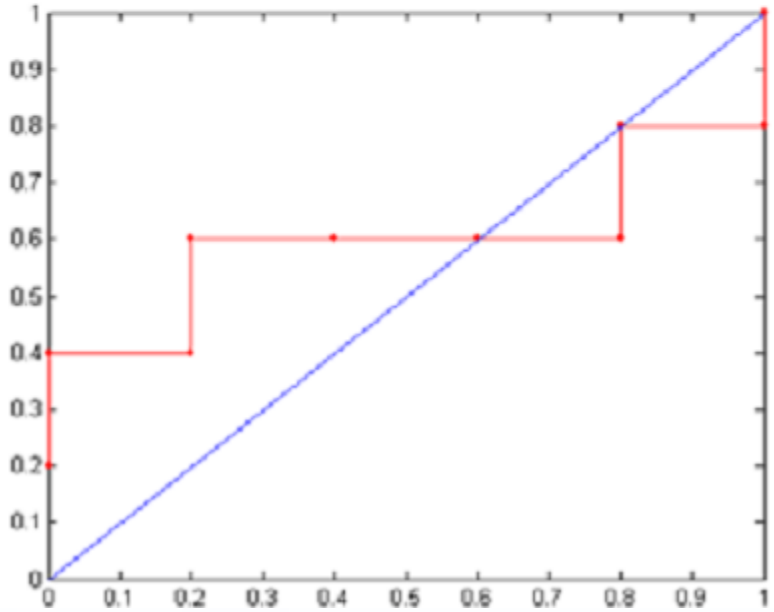
| Instance | P(+\|A) | True Class |
|---|---|---|
| 1 | 0.95 | + |
| 2 | 0.93 | + |
| 3 | 0.87 | - |
| 4 | 0.85 | - |
| 5 | 0.85 | - |
| 6 | 0.85 | + |
| 7 | 0.76 | - |
| 8 | 0.53 | + |
| 9 | 0.43 | - |
| 10 | 0.25 | + |

Hãy lập bảng tính các TPR, FPR và vẽ đường cong ROC thể hiện độ chính xác của mô hình

# Bài tập 4 – Đáp án

| | Class | + | - | + | - | - | - | + | - | + | + | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Threshold >= | | 0.25 | 0.43 | 0.53 | 0.76 | 0.85 | 0.85 | 0.85 | 0.87 | 0.93 | 0.95 | 1.00 |
| | TP | 5 | 4 | 4 | 3 | 3 | 3 | 3 | 2 | 2 | 1 | 0 |
| | FP | 5 | 5 | 4 | 4 | 3 | 2 | 1 | 1 | 0 | 0 | 0 |
| | TN | 0 | 0 | 1 | 1 | 2 | 3 | 4 | 4 | 5 | 5 | 5 |
| | FN | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 4 | 5 |
| → | TPR | 1 | 0.8 | 0.8 | 0.6 | 0.6 | 0.6 | 0.6 | 0.4 | 0.4 | 0.2 | 0 |
| → | FPR | 1 | 1 | 0.8 | 0.8 | 0.6 | 0.4 | 0.2 | 0.2 | 0 | 0 | 0 |

**ROC Curve:**



# Các kĩ thuật cải thiện độ chính xác của phân lớp

# Các kĩ thuật cải thiện

- Đọc thêm trong phần 8.6 cuốn Data Mining: Concepts and Techniques (3rd Edition, J.Han) để biết một số trick dùng cho việc tăng cường độ chính xác của mô hình.

# Tóm tắt

- Phân lớp dựa trên xác suất Naïve Bayes tính xác suất thuộc về một lớp. Lớp nào có xác suất cao nhất, mẫu sẽ được gán nhãn lớp đó.
- Sửa lỗi Laplace dùng để khắc phục trường hợp giá trị mẫu bị thiếu trong phân lớp bằng cách bổ sung "ảo" vào mỗi giá trị một mẫu.
- Phân lớp dựa trên thể hiện không phát sinh mô hình mà thực hiện phân lớp trực tiếp dựa trên các mẫu. Thuật toán k-NN dựa trên k làng giềng gần nhất để quyết định lớp cho mẫu mới.
- Việc đánh giá, so sánh thuật toán dựa trên nhiều yếu tố như độ chính xác, tốc độ, khả năng, độ tốt… Khi quan tâm đến độ chính xác, có các độ đo TP, TN, FP, FN, …, đường cong ROC

# Tài liệu tham khảo


1. J.Han, M.Kamber, Chương 8 – Classification:Basic Concepts và Chương 9 – Classification: Advanced Methods, cuốn "*Data mining: Basic Concepts and Methods*", 3rd edition
2. J.Han, M.Kamber, J.Pei, Chapter 8, http://www.cs.uiuc.edu/homes/hanj/cs412/bk3_slides/08ClassBasic.ppt
3. Bing Liu, Chapter 3 – Suppervised Learning, http://www.cs.uic.edu/~liub/teach/cs583-fall-06/CS583-supervised-learning.ppt

# Hỏi & Đáp