Trí Tuệ Nhân Tạo

Nguyễn Nhật Quang

quangnn-fit@mail.hut.edu.vn

Viện Công nghệ Thông tin và Truyền thông
Trường Đại học Bách Khoa Hà Nội

Năm học 2009-2010

Nội dung môn học:

- Giới thiệu về Trí tuệ nhân tạo
- Tác tử
- Giải quyết vấn đề: Tìm kiếm, Thỏa mãn ràng buộc
- Logic và suy diễn
- Biểu diễn tri thức
- Suy diễn với tri thức không chắc chắn
- **Học máy**
  - **Giới thiệu về học máy**
  - **Phân lớp Naïve Bayes**
  - **Học dựa trên các láng giềng gần nhất**
- Lập kế hoạch

Giới thiệu về Học máy

- Các định nghĩa về **Học máy (Machine learning)**
  - → Một quá trình nhờ đó một hệ thống cải thiện hiệu suất (hiệu quả hoạt động) của nó `[Simon, 1983]`
  - → Một quá trình mà một chương trình máy tính cải thiện hiệu suất của nó trong một công việc thông qua kinh nghiệm `[Mitchell, 1997]`
  - → Việc lập trình các máy tính để tối ưu hóa một tiêu chí hiệu suất dựa trên các dữ liệu ví dụ hoặc kinh nghiệm trong quá khứ `[Alpaydin, 2004]`
- Biểu diễn một bài toán học máy `[Mitchell, 1997]`

  Học máy = Cải thiện hiệu quả một công việc thông qua kinh nghiệm
  - Một công việc (nhiệm vụ) **T**
  - Đối với các tiêu chí đánh giá hiệu suất **P**
  - Thông qua (sử dụng) kinh nghiệm **E**

Các ví dụ của bài toán học máy (1)

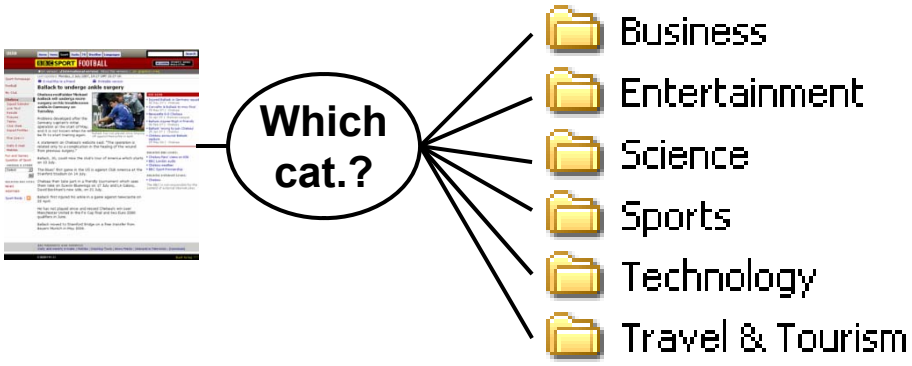
Bài toán lọc các trang Web theo sở thích của một người dùng

- **T**: Dự đoán (để lọc) xem những trang Web nào mà một người dùng cụ thể thích đọc
- **P**: Tỷ lệ (%) các trang Web được dự đoán đúng
- **E**: Một tập các trang Web mà người dùng đã chỉ định là thích đọc và một tập các trang Web mà anh ta đã chỉ định là không thích đọc

Các ví dụ của bài toán học máy (2)

Bài toán phân loại các trang Web theo các chủ đề

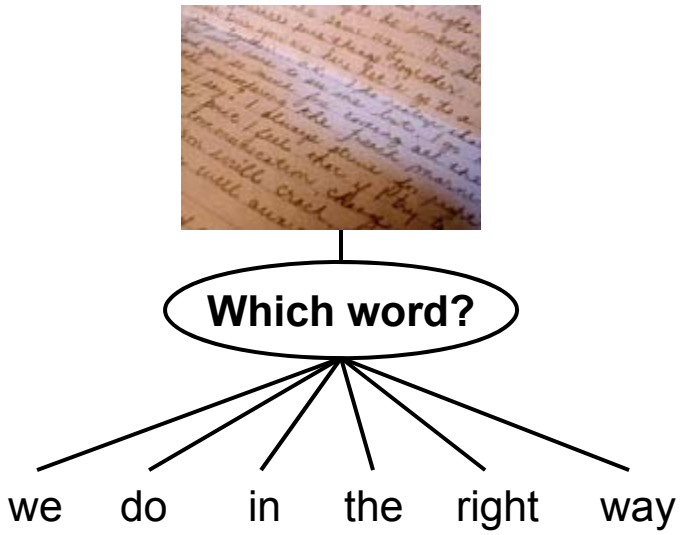
- **T**: Phân loại các trang Web theo các chủ đề đã định trước
- **P**: Tỷ lệ (%) các trang Web được phân loại chính xác
- **E**: Một tập các trang Web, trong đó mỗi trang Web gắn với một chủ đề



Các ví dụ của bài toán học máy (3)

Bài toán nhận dạng chữ viết tay

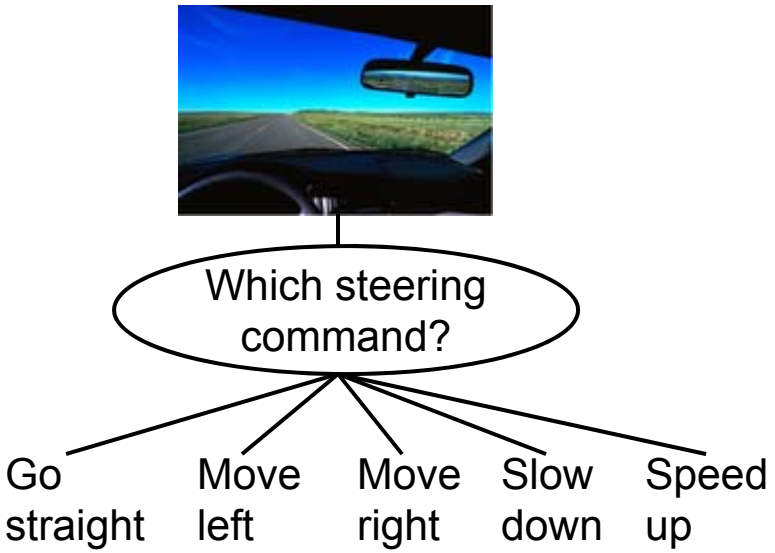
- **T**: Nhận dạng và phân loại các từ trong các ảnh chữ viết tay
- **P**: Tỷ lệ (%) các từ được nhận dạng và phân loại đúng
- **E**: Một tập các ảnh chữ viết tay, trong đó mỗi ảnh được gắn với một định danh của một từ



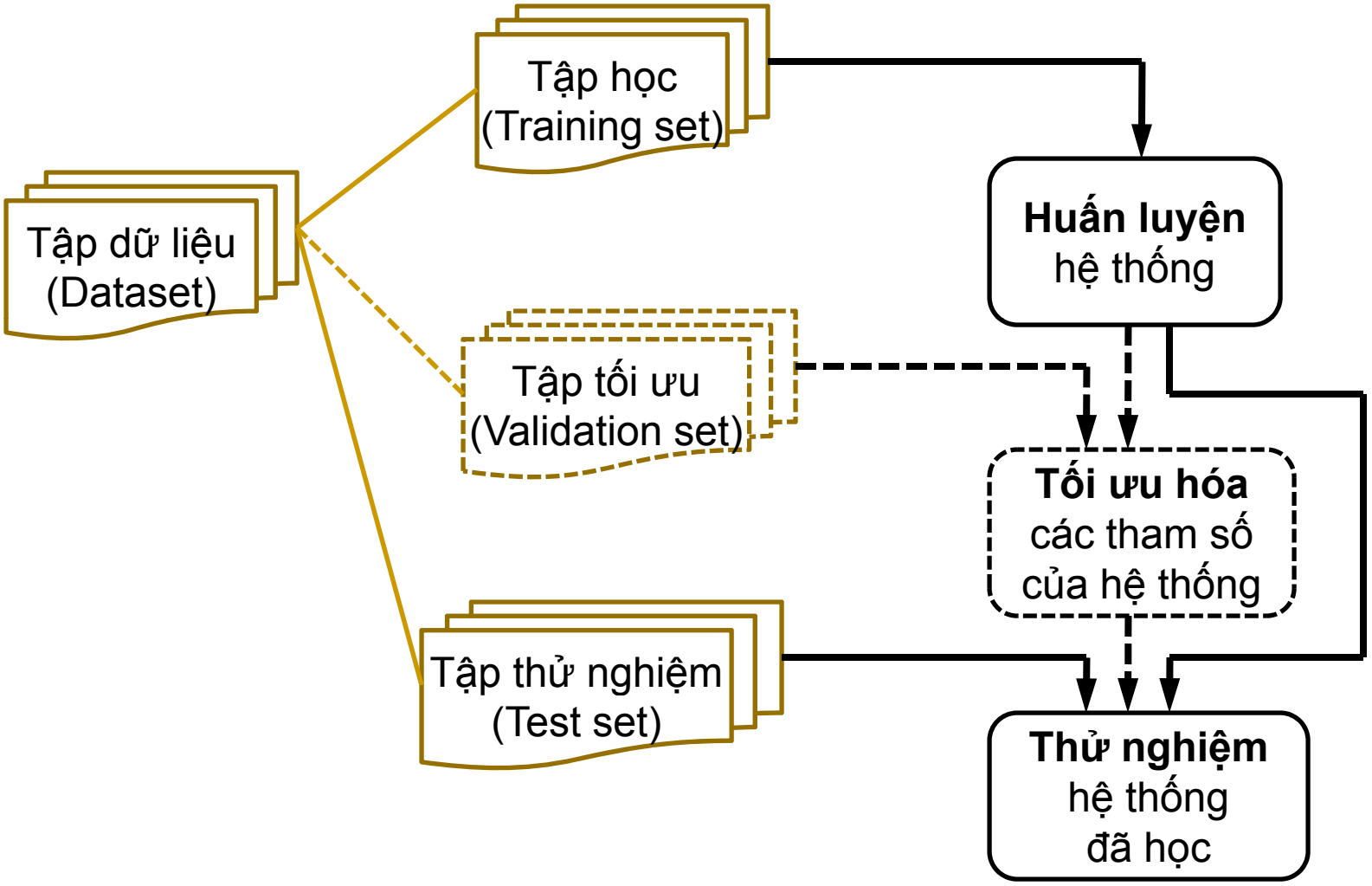
Các ví dụ của bài toán học máy (4)

Bài toán robot lái xe tự động

- **T**: Robot (được trang bị các camera quan sát) lái xe tự động trên đường cao tốc
- **P**: Khoảng cách trung bình mà robot có thể lái xe tự động trước khi xảy ra lỗi (tai nạn)
- **E**: Một tập các ví dụ được ghi lại khi quan sát một người lái xe trên đường cao tốc, trong đó mỗi ví dụ gồm một chuỗi các ảnh và các lệnh điều khiển xe



Quá trình học máy



Học có vs. không có giám sát

- Học có giám sát (supervised learning)
  - Mỗi ví dụ học gồm 2 phần: mô tả (biểu diễn) của ví dụ học, và nhãn lớp (hoặc giá trị đầu ra mong muốn) của ví dụ học đó
  - Bài toán học **phân lớp (classification problem)**

    D_train = {(<Biểu_diễn_của_x>, <Nhãn_lớp_của_x>)}
  - Bài toán học **dự đoán/hồi quy (prediction/regression problem)**

    D_train = {(<Biểu_diễn_của_x>, <Giá_trị_đầu_ra_của_x>)}
- Học không có giám sát (unsupervised learning)
  - Mỗi ví dụ học chỉ chứa mô tả (biểu diễn) của ví dụ học đó - mà không có bất kỳ thông tin nào về nhãn lớp hay giá trị đầu ra mong muốn của ví dụ học đó
  - Bài toán học **phân cụm (Clustering problem)**

    Tập học D_train = {(<Biểu_diễn_của_x>)}

Bài toán học máy – Các thành phần chính (1)

- Lựa chọn các ví dụ học (training/learning examples)
  - Các thông tin hướng dẫn quá trình học (training feedback) được chứa ngay trong các ví dụ học, hay là được cung cấp gián tiếp (vd: từ môi trường hoạt động)
  - Các ví dụ học theo kiểu có giám sát (supervised) hay không có giám sát (unsupervised)
  - Các ví dụ học phải tương thích với (đại diện cho) các ví dụ sẽ được sử dụng bởi hệ thống trong tương lai (future test examples)
- Xác định hàm mục tiêu (giả thiết, khái niệm) cần học
  - F: X → {0,1}
  - F: X → {Một tập các nhãn lớp}
  - F: X → R^+ (miền các giá tri số thực dương)
  - …

Bài toán học máy – Các thành phần chính (2)

- Lựa chọn cách biểu diễn cho hàm mục tiêu cần học
  - Hàm đa thức (a polynomial function)
  - Một tập các luật (a set of rules)
  - Một cây quyết định (a decision tree)
  - Một mạng nơ-ron nhân tạo (an artificial neural network)
  - …
- Lựa chọn một giải thuật học máy có thể học (xấp xỉ) được hàm mục tiêu
  - Phương pháp học hồi quy (Regression-based)
  - Phương pháp học quy nạp luật (Rule induction)
  - Phương pháp học cây quyết định (ID3 hoặc C4.5)
  - Phương pháp học lan truyền ngược (Back-propagation)
  - …

Các vấn đề trong Học máy (1)

- Giải thuật học máy (Learning algorithm)
  - Những giải thuật học máy nào có thể học (xấp xỉ) một hàm mục tiêu cần học?
  - Với những điều kiện nào, một giải thuật học máy đã chọn sẽ hội tụ (tiệm cận) hàm mục tiêu cần học?
  - Đối với một lĩnh vực bài toán cụ thể và đối với một cách biểu diễn các ví dụ (đối tượng) cụ thể, giải thuật học máy nào thực hiện tốt nhất?

Các vấn đề trong Học máy (2)

- Các ví dụ học (Training examples)
  - Bao nhiêu ví dụ học là đủ?
  - Kích thước của tập học (tập huấn luyện) ảnh hưởng thế nào đối với độ chính xác của hàm mục tiêu học được?
  - Các ví dụ lỗi (nhiễu) và/hoặc các ví dụ thiếu giá trị thuộc tính (missing-value) ảnh hưởng thế nào đối với độ chính xác?

Các vấn đề trong Học máy (3)

- Quá trình học (Learning process)
  - Chiến lược tối ưu cho việc lựa chọn thứ tự sử dụng (khai thác) các ví dụ học?
  - Các chiến lược lựa chọn này làm thay đổi mức độ phức tạp của bài toán học máy như thế nào?
  - Các tri thức cụ thể của bài toán (ngoài các ví dụ học) có thể đóng góp thế nào đối với quá trình học?

Các vấn đề trong Học máy (4)

- Khả năng/giới hạn học (Learning capability)
  - Hàm mục tiêu nào mà hệ thống cần học?
    - Biểu diễn hàm mục tiêu: Khả năng biểu diễn (vd: hàm tuyến tính / hàm phi tuyến) vs. Độ phức tạp của giải thuật và quá trình học
  - Các giới hạn (trên lý thuyết) đối với khả năng học của các giải thuật học máy?
  - Khả năng khái quát hóa (generalize) của hệ thống từ các ví dụ học?
    - Để tránh vấn đề "over-fitting" (đạt độ chính xác cao trên tập học, nhưng đạt độ chính xác thấp trên tập thử nghiệm)
  - Khả năng hệ thống tự động thay đổi (thích nghi) biểu diễn (cấu trúc) bên trong của nó?
    - Để cải thiện khả năng (của hệ thống đối với việc) biểu diễn và học hàm mục tiêu

Vấn đề over-fitting (1)

- Một hàm mục tiêu (một giả thiết) học được *h* sẽ được gọi là **quá khớp/quá phù hợp (over-fit)** với một tập học nếu tồn tại một hàm mục tiêu khác *h'* sao cho:
  - *h'* kém phù hợp hơn (đạt độ chính xác kém hơn) *h* đối với tập học, nhưng
  - *h'* đạt độ chính xác cao hơn *h* đối với toàn bộ tập dữ liệu (bao gồm cả những ví dụ được sử dụng sau quá trình huấn luyện)
- Vấn đề over-fitting thường do các nguyên nhân:
  - Lỗi (nhiễu) trong tập huấn luyện (do quá trình thu thập/xây dựng tập dữ liệu)
  - Số lượng các ví dụ học quá nhỏ, không đại diện cho toàn bộ tập (phân bố) của các ví dụ của bài toán học

Vấn đề over-fitting (2)

- Giả sử gọi D là tập toàn bộ các ví dụ, và D_train là tập các ví dụ học
- Giả sử gọi $Err_D(h)$ là mức lỗi mà giả thiết h sinh ra đối với tập D, và $Err_{D\_train}(h)$ là mức lỗi mà giả thiết h sinh ra đối với tập D_train
- Giả thiết h quá khớp (quá phù hợp) tập học D_train nếu tồn tại một giả thiết khác h':
  - $Err_{D\_train}(h) < Err_{D\_train}(h')$, và
  - $Err_D(h) > Err_D(h')$

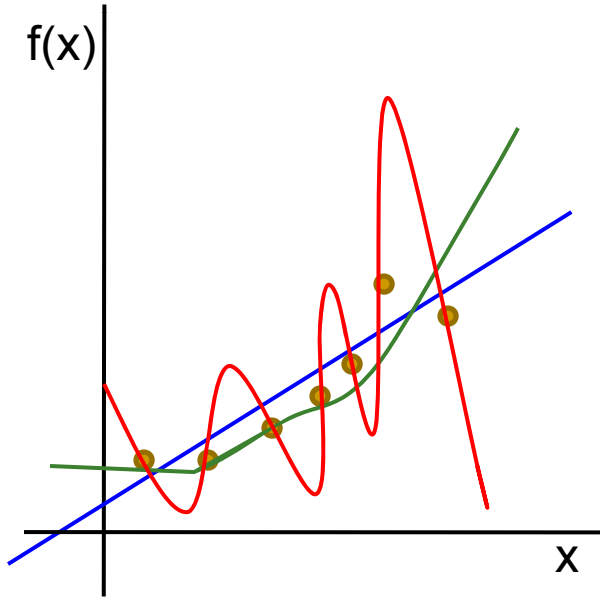
Vấn đề over-fitting (3)

- Trong số các giả thiết (hàm mục tiêu) học được, giả thiết (hàm mục tiêu) nào khái quát hóa tốt nhất từ các ví dụ học?

  Lưu ý: Mục tiêu của học máy là để đạt được độ chính xác cao trong dự đoán đối với các ví dụ sau này, không phải đối với các ví dụ học

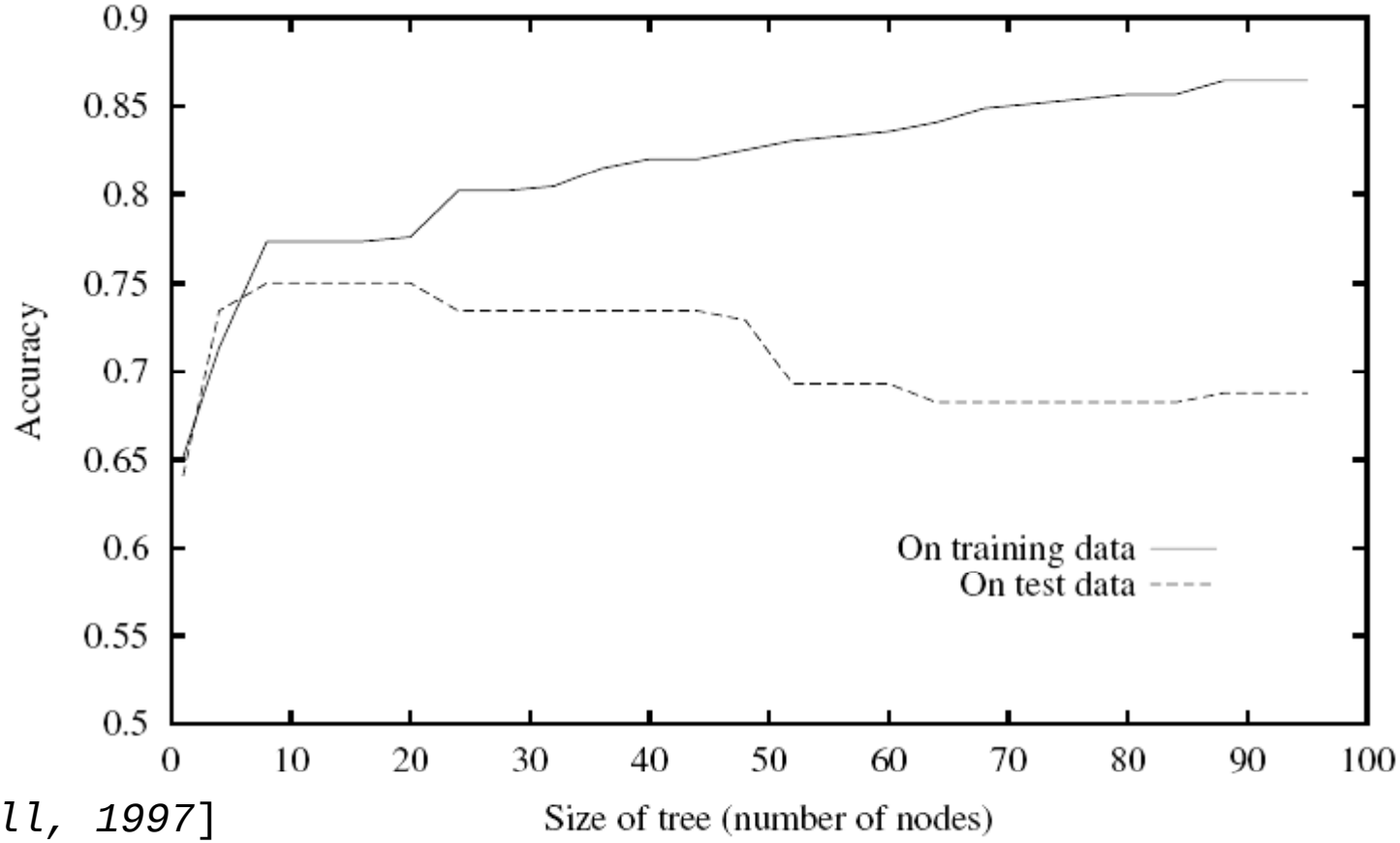
- **Occam's razor**: Ưu tiên chọn hàm mục tiêu đơn giản nhất phù hợp (không nhất thiết hoàn hảo) với các ví dụ học
  - → Khái quát hóa tốt hơn
  - → Dễ giải thích/diễn giải hơn
  - → Độ phức tạp tính toán ít hơn

Hàm mục tiêu *f(x)* nào đạt độ chính xác cao nhất đối với các ví dụ sau này?



Vấn đề over-fitting – Ví dụ

Tiếp tục quá trình học cây quyết định sẽ làm giảm độ chính xác đối với tập thử nghiệm mặc dù tăng độ chính xác đối với tập học



[*Mitchell, 1997*]

Phân lớp Naïve Bayes

- Là các phương pháp học phân lớp có giám sát và dựa trên xác suất
- Dựa trên một mô hình (hàm) xác suất
- Việc phân loại dựa trên các giá trị xác suất của các khả năng xảy ra của các giả thiết
- Là một trong các phương pháp học máy thường được sử dụng trong các bài toán thực tế
- Dựa trên định lý Bayes (Bayes theorem)

Định lý Bayes

$$P(h|D) = \frac{P(D|h).P(h)}{P(D)}$$

- P(h): Xác suất trước (prior probability) rằng giả thiết (phân lớp) h là đúng
- P(D): Xác suất trước rằng tập dữ liệu D được quan sát (thu được)
- P(D|h): Xác suất của việc quan sát được (thu được) tập dữ liệu D, với điều kiện giả thiết h là đúng
- P(h|D): Xác suất của giả thiết h là đúng, với điều kiện tập dữ liệu D được quan sát

Định lý Bayes – Ví dụ (1)

Xét tập dữ liệu sau đây:

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes

[Mitchell, 1997]

Định lý Bayes – Ví dụ (2)

- Tập ví dụ D. Tập các ngày mà thuộc tính *Outlook* có giá trị *Sunny* và thuộc tính *Wind* có giá trị *Strong*
- Giả thiết (phân lớp) h. Anh ta chơi tennis
- Xác suất trước P(h). Xác suất anh ta chơi tennis (không phụ thuộc vào các thuộc tính *Outlook* và *Wind*)
- Xác suất trước P(D). Xác suất của một ngày mà thuộc tính *Outlook* có giá trị *Sunny* và thuộc tính *Wind* có giá trị *Strong*
- P(D|h). Xác suất của một ngày mà thuộc tính *Outlook* có giá trị *Sunny* và Wind có giá trị *Strong*, với điều kiện (nếu biết rằng) anh ta chơi tennis
- P(h|D). Xác suất anh ta chơi tennis, với điều kiện (nếu biết rằng) thuộc tính *Outlook* có giá trị *Sunny* và Wind có giá trị *Strong*

  → Phương pháp phân lớp Naïve Bayes dựa trên xác suất có điều kiện (*posterior probability*) này!

Cực đại hóa xác suất có điều kiện

- Với một tập các giả thiết (các phân lớp) có thể H, hệ thống học sẽ tìm ***giả thiết có thể xảy ra nhất (the most probable hypothesis)*** h ($\in$ H) đối với các dữ liệu quan sát được D
- Giả thiết h này được gọi là giả thiết cực đại hóa xác suất có điều kiện (**maximum a posteriori – MAP**)

$$h_{MAP} = \arg\max_{h \in H} P(h \mid D)$$

$$h_{MAP} = \arg\max_{h \in H} \frac{P(D \mid h).P(h)}{P(D)}$$ (bởi định lý Bayes)

$$h_{MAP} = \arg\max_{h \in H} P(D \mid h).P(h)$$ (P (D) là như nhau đối với các giả thiết h)

MAP – Ví dụ

- Tập H bao gồm 2 giả thiết (có thể)
  - h_1: Anh ta chơi tennis
  - h_2: Anh ta không chơi tennis
- Tính giá trị của 2 xác xuất có điều kiện: $P(h_1|D)$, $P(h_2|D)$
- Giả thiết có thể nhất $h_{MAP}=h_1$ nếu $P(h_1|D) \geq P(h_2|D)$; ngược lại thì $h_{MAP}=h_2$
- Bởi vì $P(D)=P(D,h_1)+P(D,h_2)$ là như nhau đối với cả 2 giả thiết h_1 và h_2, nên có thể bỏ qua đại lượng $P(D)$
- Vì vậy, cần tính 2 biểu thức: $P(D|h_1).P(h_1)$ và $P(D|h_2).P(h_2)$, và đưa ra quyết định tương ứng
  - Nếu $P(D|h_1).P(h_1) \geq P(D|h_2).P(h_2)$, thì kết luận là anh ta chơi tennis
  - Ngược lại, thì kết luận là anh ta không chơi tennis

Đánh giá khả năng xảy ra cao nhất

- Phương pháp MAP: Với một tập các giả thiết có thể H, cần tìm một giả thiết cực đại hóa giá trị: $P(D|h).P(h)$
- Giả sử (assumption) trong phương pháp **đánh giá khả năng xảy ra cao nhất (maximum likelihood estimation – MLE)**: Tất cả các giả thiết đều có giá trị xác suất trước như nhau: $P(h_i)=P(h_j), \forall h_i,h_j \in H$
- Phương pháp MLE tìm giả thiết cực đại hóa giá trị $P(D|h)$; trong đó $P(D|h)$ được gọi là *khả năng xảy ra (likelihood)* của dữ liệu D đối với h
- Giả thiết cực đại hóa khả năng xảy ra (maximum likelihood hypothesis)

$$h_{MLE} = \arg\max_{h \in H} P(D|h)$$

MLE – Ví dụ

- Tập H bao gồm 2 giả thiết có thể
  - h_1: Anh ta chơi tennis
  - h_2: Anh ta không chơi tennis

  D: Tập dữ liệu (các ngày) mà trong đó thuộc tính *Outlook* có giá trị *Sunny* và thuộc tính Wind có giá trị *Strong*
- Tính 2 giá trị khả năng xảy ra (likelihood values) của dữ liệu D đối với 2 giả thiết: $P(D|h_1)$ và $P(D|h_2)$
  - $P(\text{Outlook=Sunny}, \text{Wind=Strong}|h_1) = 1/8$
  - $P(\text{Outlook=Sunny}, \text{Wind=Strong}|h_2) = 1/4$
- Giả thiết MLE $h_{MLE}=h_1$ nếu $P(D|h_1) \geq P(D|h_2)$; và ngược lại thì $h_{MLE}=h_2$

  → Bởi vì $P(\text{Outlook=Sunny}, \text{Wind=Strong}|h_1) < P(\text{Outlook=Sunny}, \text{Wind=Strong}|h_2)$, hệ thống kết luận rằng: *Anh ta sẽ không chơi tennis!*

Phân loại Naïve Bayes (1)

- Biểu diễn bài toán phân loại (classification problem)
  - Một tập học `D_train`, trong đó mỗi ví dụ học x được biểu diễn là một vectơ n chiều: $(x_1, x_2, ..., x_n)$
  - Một tập xác định các nhãn lớp: $C=\{c_1, c_2, ..., c_m\}$
  - Với một ví dụ (mới) z, z sẽ được phân vào lớp nào?
- Mục tiêu: Xác định phân lớp có thể (phù hợp) nhất đối với z

$$c_{MAP} = \arg\max_{c_i \in C} P(c_i \mid z)$$

$$c_{MAP} = \arg\max_{c_i \in C} P(c_i \mid z_1, z_2, ..., z_n)$$

$$c_{MAP} = \arg\max_{c_i \in C} \frac{P(z_1, z_2, ..., z_n \mid c_i).P(c_i)}{P(z_1, z_2, ..., z_n)}$$ (bởi định lý Bayes)

Phân loại Naïve Bayes (2)

- Để tìm được phân lớp có thể nhất đối với z…

$$c_{MAP} = \arg\max_{c_i \in C} P(z_1, z_2, ..., z_n \mid c_i).P(c_i)$$

(P($z_1, z_2, ..., z_n$) là như nhau với các lớp)

- **Giả sử (assumption) trong phương pháp phân loại Naïve Bayes**. Các thuộc tính là *độc lập có điều kiện (conditionally independent)* đối với các lớp

$$P(z_1, z_2, ..., z_n \mid c_i) = \prod_{j=1}^{n} P(z_j \mid c_i)$$

- Phân loại Naïve Bayes tìm phân lớp có thể nhất đối với z

$$c_{NB} = \arg\max_{c_i \in C} P(c_i).\prod_{j=1}^{n} P(z_j \mid c_i)$$

Phân loại Naïve Bayes – Giải thuật

- Giai đoạn học (training phase), sử dụng một tập học

  Đối với mỗi phân lớp có thể (mỗi nhãn lớp) $c_i \in C$

  - Tính giá trị xác suất trước: $P(c_i)$
  - Đối với mỗi giá trị thuộc tính x_j, tính giá trị xác suất xảy ra của giá trị thuộc tính đó đối với một phân lớp c_i: $P(x_j | c_i)$

- Giai đoạn phân lớp (classification phase), đối với một ví dụ mới
  - Đối với mỗi phân lớp $c_i \in C$, tính giá trị của biểu thức:

$$P(c_i).\prod_{j=1}^{n} P(x_j \mid c_i)$$

  - Xác định phân lớp của z là lớp có thể nhất c^*

$$c^* = \arg\max_{c_i \in C} P(c_i).\prod_{j=1}^{n} P(x_j \mid c_i)$$

Phân lớp Naïve Bayes – Ví dụ (1)

Một sinh viên trẻ với thu nhập trung bình và mức đánh giá tín dụng bình thường sẽ mua một cái máy tính?

Rec. ID	Age	Income	Student	Credit_Rating	Buy_Computer
1	Young	High	No	Fair	No
2	Young	High	No	Excellent	No
3	Medium	High	No	Fair	Yes
4	Old	Medium	No	Fair	Yes
5	Old	Low	Yes	Fair	Yes
6	Old	Low	Yes	Excellent	No
7	Medium	Low	Yes	Excellent	Yes
8	Young	Medium	No	Fair	No
9	Young	Low	Yes	Fair	Yes
10	Old	Medium	Yes	Fair	Yes
11	Young	Medium	Yes	Excellent	Yes
12	Medium	Medium	No	Excellent	Yes
13	Medium	High	Yes	Fair	Yes
14	Old	Medium	No	Excellent	No

Phân lớp Naïve Bayes – Ví dụ (2)

- Biểu diễn bài toán phân loại
  - z = (Age=Young, Income=Medium, Student=Yes, Credit_Rating=Fair)
  - Có 2 phân lớp có thể: c_1 ("Mua máy tính") và c_2 ("Không mua máy tính")
- Tính giá trị xác suất trước cho mỗi phân lớp
  - $P(c_1) = 9/14$
  - $P(c_2) = 5/14$
- Tính giá trị xác suất của mỗi giá trị thuộc tính đối với mỗi phân lớp
  - $P(\text{Age=Young}|c_1) = 2/9$; $P(\text{Age=Young}|c_2) = 3/5$
  - $P(\text{Income=Medium}|c_1) = 4/9$; $P(\text{Income=Medium}|c_2) = 2/5$
  - $P(\text{Student=Yes}|c_1) = 6/9$; $P(\text{Student=Yes}|c_2) = 1/5$
  - $P(\text{Credit\_Rating=Fair}|c_1) = 6/9$; $P(\text{Credit\_Rating=Fair}|c_2) = 2/5$

Phân lớp Naïve Bayes – Ví dụ (3)

- Tính toán xác suất có thể xảy ra (likelihood) của ví dụ z đối với mỗi phân lớp
  - Đối với phân lớp c_1

    $P(z|c_1) = P(\text{Age=Young}|c_1).P(\text{Income=Medium}|c_1).P(\text{Student=Yes}|c_1).$
    $P(\text{Credit\_Rating=Fair}|c_1) = (2/9).(4/9).(6/9).(6/9) = 0.044$

  - Đối với phân lớp c_2

    $P(z|c_2) = P(\text{Age=Young}|c_2).P(\text{Income=Medium}|c_2).P(\text{Student=Yes}|c_2).$
    $P(\text{Credit\_Rating=Fair}|c_2) = (3/5).(2/5).(1/5).(2/5) = 0.019$

- Xác định phân lớp có thể nhất (the most probable class)
  - Đối với phân lớp c_1

    $P(c_1).P(z|c_1) = (9/14).(0.044) = 0.028$

  - Đối với phân lớp c_2

    $P(c_2).P(z|c_2) = (5/14).(0.019) = 0.007$

  → Kết luận: *Anh ta (z) sẽ mua một máy tính*!

Phân lớp Naïve Bayes – Vấn đề (1)

- Nếu không có ví dụ nào gắn với phân lớp c_i có giá trị thuộc tính x_j…

  $P(x_j|c_i)=0$, và vì vậy: $P(c_i).\prod_{j=1}^{n} P(x_j|c_i)=0$

- Giải pháp: Sử dụng phương pháp Bayes để ước lượng $P(x_j|c_i)$

$$P(x_j|c_i)=\frac{n(c_i,x_j)+mp}{n(c_i)+m}$$

  - $n(c_i)$: số lượng các ví dụ học gắn với phân lớp c_i
  - $n(c_i,x_j)$: số lượng các ví dụ học gắn với phân lớp c_i có giá trị thuộc tính x_j
  - p: ước lượng đối với giá trị xác suất $P(x_j|c_i)$

    → Các ước lượng đồng mức: p=1/k, với thuộc tính f_j có k giá trị có thể
  - m: một hệ số (trọng số)

    → Để bổ sung cho $n(c_i)$ các ví dụ thực sự được quan sát với thêm m mẫu ví dụ với ước lượng p

Phân lớp Naïve Bayes – Vấn đề (2)

- Giới hạn về độ chính xác trong tính toán của máy tính
  - $P(x_j | c_i) < 1$, đối với mọi giá trị thuộc tính x_j và phân lớp c_i
  - Vì vậy, khi số lượng các giá trị thuộc tính là rất lớn, thì:

$$\lim_{n \to \infty} \left(\prod_{j=1}^{n} P(x_j \mid c_i) \right) = 0$$

- Giải pháp: Sử dụng hàm lôgarit cho các giá trị xác suất

$$c_{NB} = \arg\max_{c_i \in C} \left(\log \left[P(c_i) . \prod_{j=1}^{n} P(x_j \mid c_i) \right] \right)$$

$$c_{NB} = \arg\max_{c_i \in C} \left(\log P(c_i) + \sum_{j=1}^{n} \log P(x_j \mid c_i) \right)$$

Phân loại văn bản bằng NB (1)

- Biểu diễn bài toán phân loại văn bản
  - Tập học D_train, trong đó mỗi ví dụ học là một biểu diễn văn bản gắn với một nhãn lớp: D = {(d_k, c_i)}
  - Một tập các nhãn lớp xác định: C = {c_i}
- Giai đoạn học
  - Từ tập các văn bản trong D_train, trích ra tập các từ khóa (keywords/terms): T = {t_j}
  - Gọi D_c_i ($\subseteq$D_train) là tập các văn bản trong D_train có nhãn lớp c_i
  - Đối với mỗi phân lớp c_i
    - Tính giá trị xác suất trước của phân lớp c_i: $P(c_i) = \frac{|D\_c_i|}{|D|}$
    - Đối với mỗi từ khóa t_j, tính xác suất từ khóa t_j xuất hiện đối với lớp c_i

$$P(t_j \mid c_i) = \frac{\left(\sum_{d_k \in D\_c_i} n(d_k, t_j)\right) + 1}{\left(\sum_{d_k \in D\_c_i} \sum_{t_m \in T} n(d_k, t_m)\right) + |T|}$$

($n(d_k, t_j)$: số lần xuất hiện của từ khóa t_j trong văn bản d_k)

Phân loại văn bản bằng NB (2)

- Để phân lớp cho một văn bản mới d
- Giai đoạn phân lớp
  - Từ văn bản d, trích ra tập T_d gồm các từ khóa (keywords) t_j đã được định nghĩa trong tập T ($T\_d \subseteq T$)
  - **Giả sử (assumption)**. Xác suất từ khóa t_j xuất hiện đối với lớp c_i là độc lập đối với vị trí của từ khóa đó trong văn bản

    $P(t_j \text{ ở vị trí } k \mid c_i) = P(t_j \text{ ở vị trí } m \mid c_i), \ \forall k,m$
  - Đối với mỗi phân lớp c_i, tính giá trị likelihood của văn bản d đối với c_i

    $$P(c_i).\prod_{t_j \in T\_d} P(t_j \mid c_i)$$
  - Phân lớp văn bản d thuộc vào lớp c^*

    $$c^* = \arg\max_{c_i \in C} P(c_i).\prod_{t_j \in T\_d} P(t_j \mid c_i)$$

Học dựa trên láng giềng gần nhất

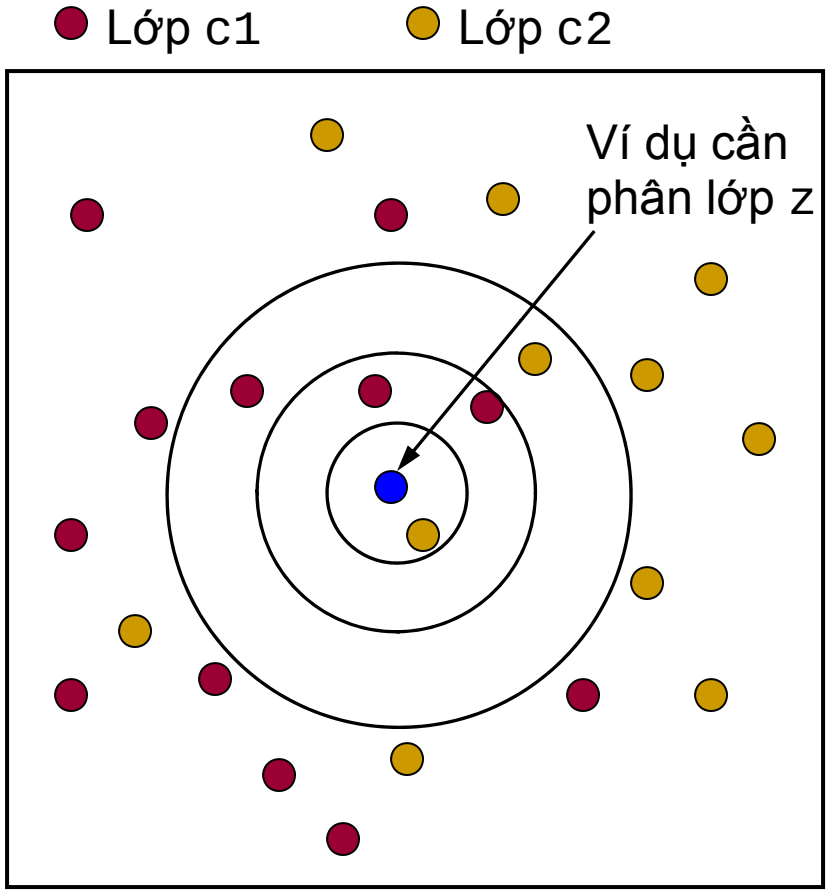
- Một số tên gọi khác của phương pháp học dựa trên láng giềng gần nhất (Nearest neighbor learning)
  - Instance-based learning
  - Lazy learning
  - Memory-based learning
- Ý tưởng của phương pháp học dựa trên láng giềng gần nhất
  - Với một tập các ví dụ học
    - (Đơn giản là) lưu lại các ví dụ học
    - Không cần xây dựng một mô hình (mô tả) rõ ràng và tổng quát của hàm mục tiêu cần học
  - Đối với một ví dụ cần phân loại/dự đoán
    - Kiểm tra (xét) quan hệ giữa ví dụ đó với các ví dụ học để gán giá trị của hàm mục tiêu (một nhãn lớp, hoặc một giá trị thực)

Học dựa trên láng giềng gần nhất

- Biểu diễn đầu vào của bài toán
  - Mỗi ví dụ x được biểu diễn là một vectơ n chiều trong không gian các vectơ $X \in R^n$
  - $x = (x_1, x_2, \ldots, x_n)$, trong đó x_i ($\in R$) là một số thực
- Chúng ta xét 2 kiểu bài toán học
  - Bài toán *phân lớp (classification)*
    - Để học một hàm mục tiêu có giá trị rời rạc (a discrete-valued target function)
    - Đầu ra của hệ thống là một trong số các giá trị rời rạc đã xác định trước (một trong các nhãn lớp)
  - Bài toán *dự đoán/hồi quy (prediction/regression)*
    - Để học một hàm mục tiêu có giá trị liên tục (a continuous-valued target function)
    - Đầu ra của hệ thống là một giá trị số thực

Phân lớp dựa trên NN – Ví dụ

- Xét 1 láng giềng gần nhất
  → Gán z vào lớp c2
- Xét 3 láng giềng gần nhất
  → Gán z vào lớp c1
- Xét 5 láng giềng gần nhất
  → Gán z vào lớp c1



Giải thuật phân lớp k-NN

- Mỗi ví dụ học x được biểu diễn bởi 2 thành phần:
  - Mô tả của ví dụ: $x=(x_1,x_2,\ldots,x_n)$, trong đó $x_i \in R$
  - Nhãn lớp : c ($\in C$, với C là tập các nhãn lớp được xác định trước)
- Giai đoạn học
  - Đơn giản là lưu lại các ví dụ học trong tập học $D = \{x\}$
- Giai đoạn phân lớp: Để phân lớp cho một ví dụ (mới) z
  - Với mỗi ví dụ học$x \in D$, tính khoảng cách giữa x và z
  - Xác định tập NB(z) – các láng giềng gần nhất của z

    → Gồm k ví dụ học trong D gần nhất với z tính theo một hàm khoảng cách d
  - Phân *z* vào lớp chiếm số đông (the majority class) trong số các lớp của các ví dụ học trong NB(z)

Giải thuật dự đoán k-NN

- Mỗi ví dụ học x được biểu diễn bởi 2 thành phần:
  - Mô tả của ví dụ: $x=(x_1,x_2,\ldots,x_n)$, trong đó $x_i \in R$
  - Giá trị đầu ra mong muốn: $y_x \in R$ (là một số thực)
- Giai đoạn học
  - Đơn giản là lưu lại các ví dụ học trong tập học D
- Giai đoạn dự đoán: Để dự đoán giá trị đầu ra cho ví dụ z
  - Đối với mỗi ví dụ học $x \in D$, tính khoảng cách giữa x và z
  - Xác định tập $NB(z)$ – các láng giềng gần nhất của z
    - → Gồm k ví dụ học trong D gần nhất với z tính theo một hàm khoảng cách d
  - Dự đoán giá trị đầu ra đối với z:

$$y_z = \frac{1}{k}\sum_{x \in NB(z)} y_x$$

Xét một hay nhiều láng giềng?

- Việc phân lớp (hay dự đoán) chỉ dựa trên duy nhất một láng giềng gần nhất (là ví dụ học gần nhất với ví dụ cần phân lớp/dự đoán) thường *không* chính xác
  - Nếu ví dụ học này là một ví dụ bất thường, không điển hình (an outlier) – rất khác so với các ví dụ khác
  - Nếu ví dụ học này có nhãn lớp (phân lớp sai) – do lỗi trong quá trình thu thập (xây dựng) tập dữ liệu
- Thường xét *k* (>1) các ví dụ học gần nhất với ví dụ cần phân lớp, và gán ví dụ đó vào lớp chiếm số đông trong số *k* ví dụ học gần nhất này
- *k* thường được chọn là một số lẻ, để tránh cân bằng về tỷ lệ phân lớp (ties in classification)
  - Ví dụ: k= 3, 5, 7,…

Hàm tính khoảng cách (1)

- Hàm tính khoảng cách ***d***
  - Đóng vai trò rất quan trọng trong phương pháp học dựa trên láng giềng gần nhất
  - Thường được xác định trước, và không thay đổi trong suốt quá trình học và phân loại/dự đoán
- Lựa chọn hàm khoảng cách ***d***
  - *Các hàm khoảng cách hình học*: Dành cho các bài toán có các thuộc tính đầu vào là kiểu số thực ($x_i \in R$)
  - *Hàm khoảng cách Hamming*: Dành cho các bái toán có các thuộc tính đầu vào là kiểu nhị phân ($x_i \in \{0,1\}$)
  - *Hàm tính độ tương tự Cosine*: Dành cho các bài toán phân lớp văn bản (x_i là giá trị trọng số TF/IDF của từ khóa thứ *i*)

Hàm tính khoảng cách (2)

- Các hàm tính khoảng cách hình học (Geometry distance functions)
  - Hàm Manhattan: $$d(x,z) = \sum_{i=1}^{n} |x_i - z_i|$$
  - Hàm Euclid: $$d(x,z) = \sqrt{\sum_{i=1}^{n} (x_i - z_i)^2}$$
  - Hàm Minkowski (p-norm): $$d(x,z) = \left(\sum_{i=1}^{n} |x_i - z_i|^p \right)^{1/p}$$
  - Hàm Chebyshev: $$d(x,z) = \lim_{p \to \infty} \left(\sum_{i=1}^{n} |x_i - z_i|^p \right)^{1/p} = \max_i |x_i - z_i|$$

Hàm tính khoảng cách (3)

- Hàm khoảng cách Hamming
  - Đối với các thuộc tính đầu vào là kiểu nhị phân
  - Ví dụ: x=(0,1,0,1,1)

$$d(x,z) = \sum_{i=1}^{n} Difference(x_i, z_i)$$

$$Difference(a,b) = \begin{cases} 1, if\ (a \neq b) \\ 0, if\ (a = b) \end{cases}$$

- Hàm tính độ tương tự Cosine
  - Đối với đầu vào là một vectơ các giá trị trọng số (TF/IDF) của các từ khóa

$$d(x,z) = \frac{x.z}{\|x\|\|z\|} = \frac{\sum_{i=1}^{n} x_i z_i}{\sqrt{\sum_{i=1}^{n} x_i^{\,2}}\sqrt{\sum_{i=1}^{n} z_i^{\,2}}}$$

Chuẩn hóa miền giá trị thuộc tính

- Hàm tính khoảng cách Euclid: $d(x,z) = \sqrt{\sum_{i=1}^{n}(x_i - z_i)^2}$
- Giả sử mỗi ví dụ được biểu diễn bởi 3 thuộc tính: `Age`, `Income` (cho mỗi tháng), và `Height` (đo theo mét)
  - `x` = (`Age`=20, `Income`=12000, `Height`=1.68)
  - `z` = (`Age`=40, `Income`=1300, `Height`=1.75)
- Khoảng cách giữa `x` và `z`
  - `d(x,z)` = $[(20\text{-}40)^2 + (12000\text{-}1300)^2 + (1.68\text{-}1.75)^2]^{1/2}$
  - Giá trị khoảng cách này bị quyết định chủ yếu bởi giá trị khoảng cách (sự khác biệt) giữa 2 ví dụ đối với thuộc tính `Income`

    → Vì: Thuộc tính `Income` có miền giá trị rất lớn so với các thuộc tính khác
- Cần phải chuẩn hóa miền giá trị (đưa về cùng một khoảng giá trị)
  - Khoảng giá trị [0,1] thường được sử dụng
  - Đối với mỗi thuộc tính `i`: $x_i = x_i$/giá_trị_cực_đại_đối_với_thuộc_tính_*i*

Trọng số của các thuộc tính

- Hàm khoảng cách Euclid:

$$d(x,z) = \sqrt{\sum_{i=1}^{n} (x_i - z_i)^2}$$

  - Tất cả các thuộc tính có cùng (như nhau) ảnh hưởng đối với giá trị khoảng cách

- Các thuộc tính khác nhau có thể (nên) có mức độ ảnh hưởng khác nhau đối với giá trị khoảng cách

- Cần phải tích hợp (đưa vào) các giá trị trọng số của các thuộc tính trong hàm tính khoảng cách

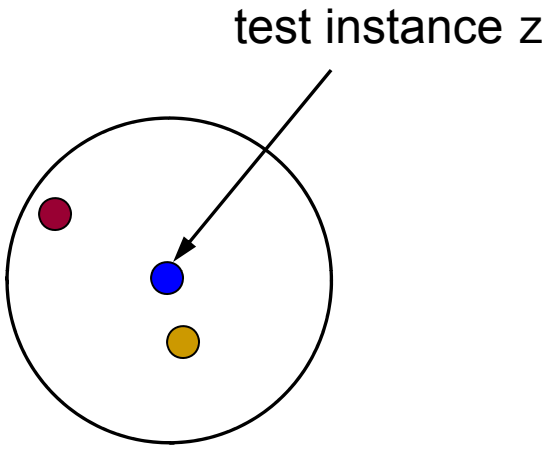
$$d(x,z) = \sqrt{\sum_{i=1}^{n} w_i (x_i - z_i)^2}$$

  - w_i là trọng số của thuộc tính `i`:

- Làm sao để xác định các giá trị trọng số của các thuộc tính?
  - Dựa trên các tri thức cụ thể của bài toán (vd: được chỉ định bởi các chuyên gia trong lĩnh vực của bài toán đang xét)
  - Bằng một quá trình tối ưu hóa các giá trị trọng số (vd: sử dụng một tập học để học một bộ các giá trị trọng số tối ưu)

Khoảng cách của các láng giềng (1)

- Xét tập NB(z) – gồm *k* ví dụ học gần nhất với ví dụ cần phân lớp/dự đoán z
  - Mỗi ví dụ (láng giềng gần nhất) này có khoảng cách khác nhau đến z
  - Các láng giềng này có ảnh hưởng như nhau đối với việc phân lớp/dự đoáncho z? → KHÔNG!
- Cần gán các mức độ ảnh hưởng (đóng góp) của mỗi láng giềng gần nhất tùy theo khoảng cách của nó đến z
  - Mức độ ảnh hưởng cao hơn cho các láng giềng gần hơn!



Khoảng cách của các láng giềng (2)

- Gọi v là hàm xác định trọng số theo khoảng cách
  - Đối với một giá trị d(x,z) – khoảng cách giữa x và z
  - v(x,z) tỷ lệ nghịch với d(x,z)
- Đối với bài toán phân lớp:

$$c(z) = \arg\max_{c_j \in C} \sum_{x \in NB(z)} v(x,z).Identical(c_j, c(x))$$

$$Identical(a,b) = \begin{cases} 1, if\ (a=b) \\ 0, if\ (a \neq b) \end{cases}$$

- Đối với bài toán dự đoán (hồi quy):

$$f(z) = \frac{\sum_{x \in NB(z)} v(x,z).f(x)}{\sum_{x \in NB(z)} v(x,z)}$$

- Lựa chọn một hàm xác định trọng số theo khoảng cách:

$$v(x,z) = \frac{1}{\alpha + d(x,z)} \qquad v(x,z) = \frac{1}{\alpha + [d(x,z)]^2} \qquad v(x,z) = e^{-\frac{d(x,z)^2}{\sigma^2}}$$

Học NN – Khi nào?

- Các ví dụ được biểu diễn là các vectơ trong không gian số thực (R^n)
- Số lượng các thuộc tính (số chiều của không gian) đầu vào không lớn
- Tập học khá lớn (nhiều ví dụ học)
- Các ưu điểm
  - Không cần bước học (hệ thống chỉ đơn giản là lưu lại các ví dụ học)
  - Hoạt động tốt với các bài toán có số lớp khá lớn
    → Không cần phải học riêng rẽ n bộ phân lớp cho n lớp
  - Phương pháp học k-NN (k >>1) có thể làm việc được cả với dữ liệu lỗi
    → Việc phân lớp/dự đoán dựa trên k láng giềng gần nhất
- Các nhược điểm
  - Phải xác định hàm tính khoảng cách phù hợp
  - Chi phí tính toán (về thời gian và bộ nhớ) tại thời điểm phân lớp/dự đoán
  - Có thể phân lớp/dự đoán sai , do các thuộc tính không liên quan (irrelevant attributes)

Tài liệu tham khảo

- E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2004.
- T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- H. A. Simon. *Why Should Machines Learn?* In R. S. Michalski, J. Carbonell, and T. M. Mitchell (Eds.): Machine learning: An artificial intelligence approach, chapter 2, pp. 25-38. Morgan Kaufmann, 1983.