

Trí Tuệ Nhân Tạo

(Artificial Intelligence)

Lê Thanh Hương

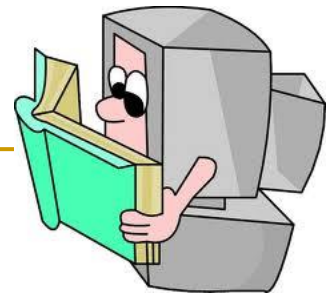
Viện Công nghệ thông tin và Truyền thông
Trường Đại Học Bách Khoa Hà Nội

Nội dung môn học

- Chương 1. Tổng quan
- Chương 2. Tác tử thông minh
- Chương 3. Giải quyết vấn đề
- Chương 4. Tri thức và suy diễn
- Chương 5. Biểu diễn tri thức
- **Chương 6. Học máy**
 - Giới thiệu về học máy
 - K láng giềng gần
 - Phân lớp Naïve Bayes
 - Học cây quyết định
 - Mạng nơron

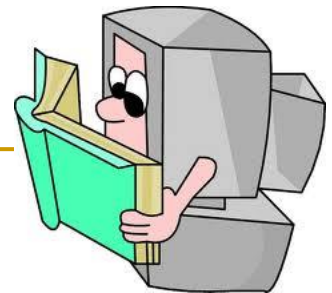
Giới thiệu về Học máy

- Học máy (ML - Machine Learning) là một lĩnh vực nghiên cứu của Trí tuệ nhân tạo (Artificial Intelligence)
- Câu hỏi trung tâm của ML:
 - *“How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?”* [Mitchell, 2006]
- Vài quan điểm về học máy:
 - Một quá trình nhờ đó một hệ thống cải thiện hiệu suất (hiệu quả hoạt động) của nó [Simon, 1983]
 - Việc lập trình các máy tính để tối ưu hóa một tiêu chí hiệu suất dựa trên các dữ liệu hoặc kinh nghiệm trong quá khứ [Alpaydin, 2010]



Một máy học

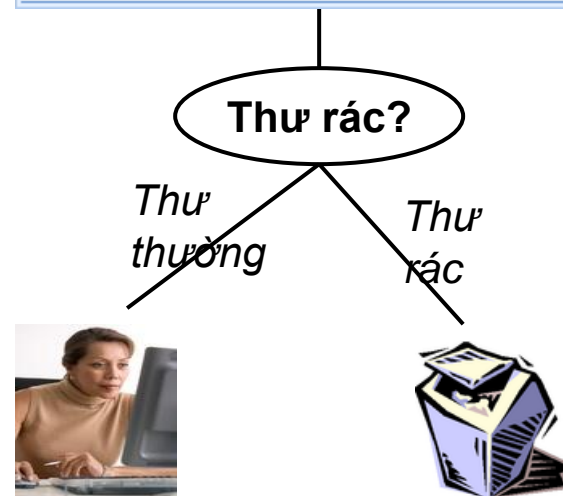
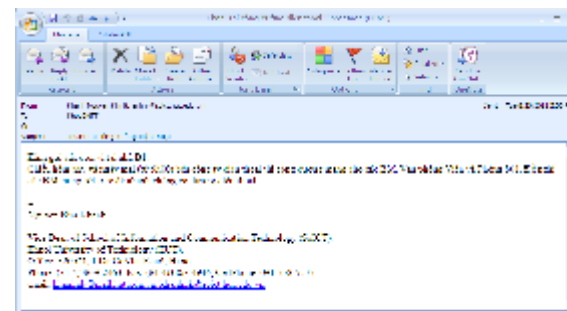
- Ta nói một máy tính *có khả năng học* nếu nó tự cải thiện hiệu suất hoạt động P cho một công việc T cụ thể, dựa vào kinh nghiệm E của nó.
- Như vậy *một bài toán học máy* có thể biểu diễn bằng 1 bộ (T, P, E)
 - T : một công việc (nhiệm vụ)
 - P : tiêu chí đánh giá hiệu năng
 - E : kinh nghiệm



Ví dụ bài toán học máy (1)

Lọc thư rác (email spam filtering)

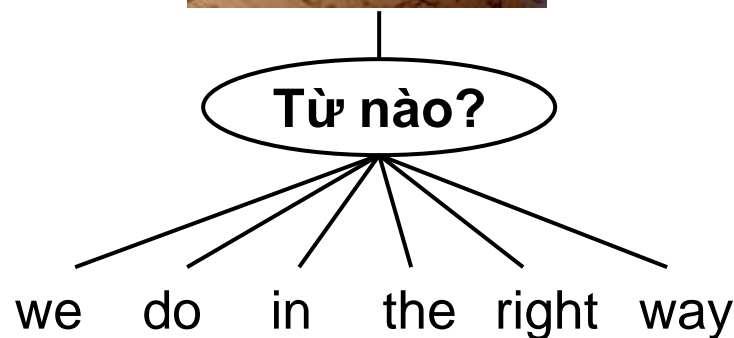
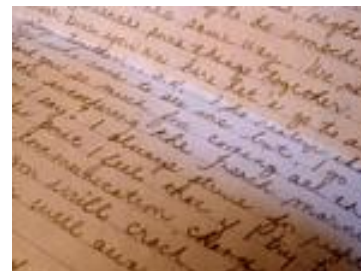
- **T**: Dự đoán (để lọc) những thư điện tử nào là thư rác (spam email)
- **P**: số lượng thư điện tử gửi đến được phân loại chính xác
- **E**: Một tập các thư điện tử (emails) mẫu, mỗi thư điện tử được biểu diễn bằng một tập thuộc tính (vd: tập từ khóa) và nhãn lớp (thư thường/thư rác) tương ứng



Ví dụ bài toán học máy (2)

Nhận dạng chữ viết tay

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



Ví dụ bài toán học máy (3)

Gán nhãn ảnh

- **T**: đưa ra một vài mô tả ý nghĩa của 1 bức ảnh
- **P**: ?
- **E**: Một tập các bức ảnh, trong đó mỗi ảnh đã được gán một tập các từ mô tả ý nghĩa của chúng



FISH WATER OCEAN
TREE CORAL



PEOPLE MARKET PATTERN
TEXTILE DISPLAY



BIRDS NEST TREE
BRANCH LEAVES

Máy học (1)

- Học một ánh xạ (hàm):

$$f : x \mapsto y$$

- x : quan sát (dữ liệu), kinh nghiệm
 - y : phán đoán, tri thức mới, kinh nghiệm mới, ...
- **Hồi quy** (regression): nếu y là một số thực
 - **Phân loại** (classification): nếu y thuộc một tập rời rạc (tập nhãn lớp)
-

Máy học (2)

- Học từ đâu?

- Từ các quan sát trong quá khứ (tập học).

$$\{x_1, x_2, \dots, x_N\}; \{y_1, y_2, \dots, y_M\}$$

- Sau khi đã học:

- Thu được một mô hình, kinh nghiệm, tri thức mới.

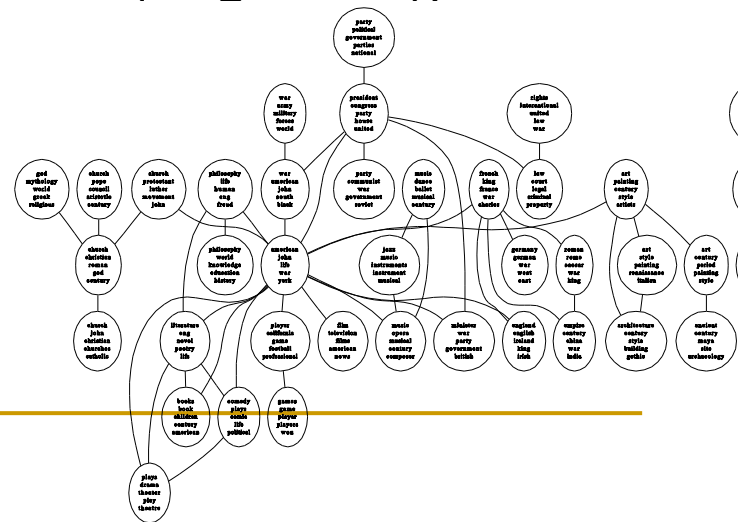
- Dùng nó để suy diễn (phán đoán) cho quan sát trong tương lai.

$$Y = f(x)$$

Hai bài toán học cơ bản

- **Học có giám sát (supervised learning):** cần học một hàm $y = f(x)$ từ tập học $\{\{x_1, x_2, \dots, x_N\}; \{y_1, y_2, \dots, y_N\}\}$ sao cho $y_i \in f(x_i)$.
 - *Phân loại* (phân lớp): nếu y chỉ nhận giá trị từ một tập rời rạc, chẳng hạn {cá, cây, quả, mèo}
 - *Hồi quy*: nếu y nhận giá trị số thực
- **Học không giám sát (unsupervised learning):** cần học một hàm $y = f(x)$ từ tập học cho trước $\{x_1, x_2, \dots, x_N\}$.

- Y có thể là các cụm dữ liệu.
- Y có thể là các cấu trúc ẩn.



Học có giám sát: ví dụ

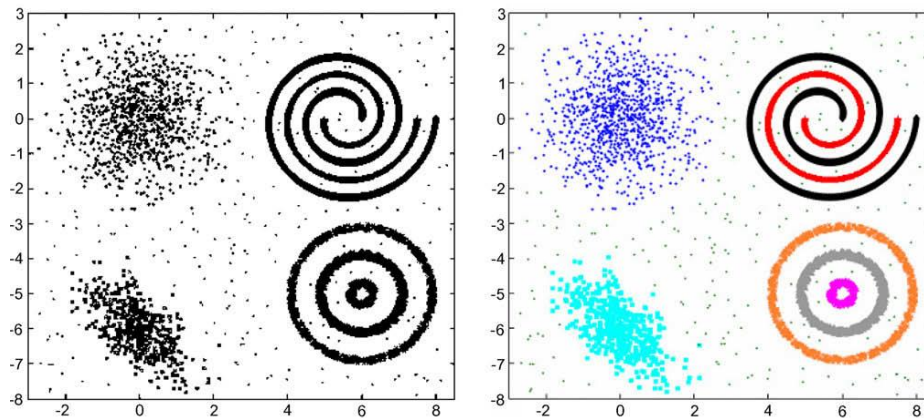
- Lọc thư rác
- Phân loại trang web
- Dự đoán rủi ro tài chính
- Dự đoán biến động chỉ số chứng khoán
- Phát hiện tấn công mạng



Học không giám sát: ví dụ (1)

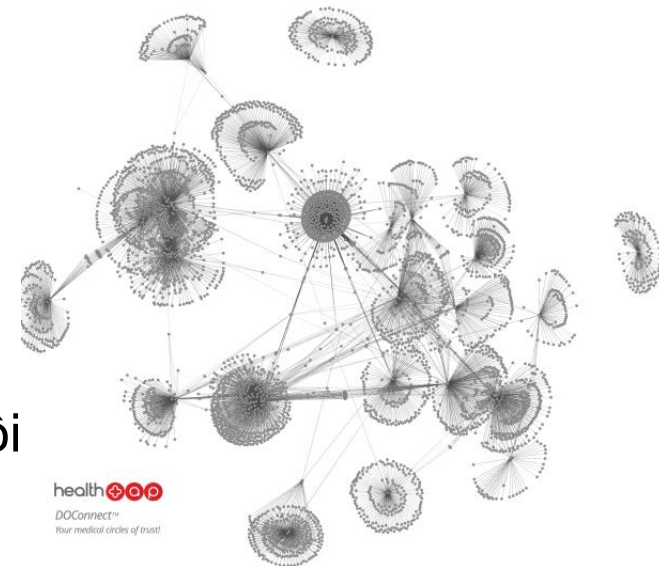
■ Phân cụm (clustering)

- Phát hiện các cụm dữ liệu, cụm tính chất,...



■ Community detection

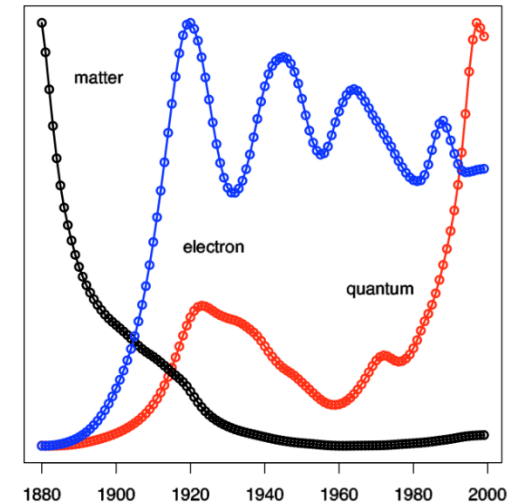
- Phát hiện các cộng đồng trong mạng xã hội



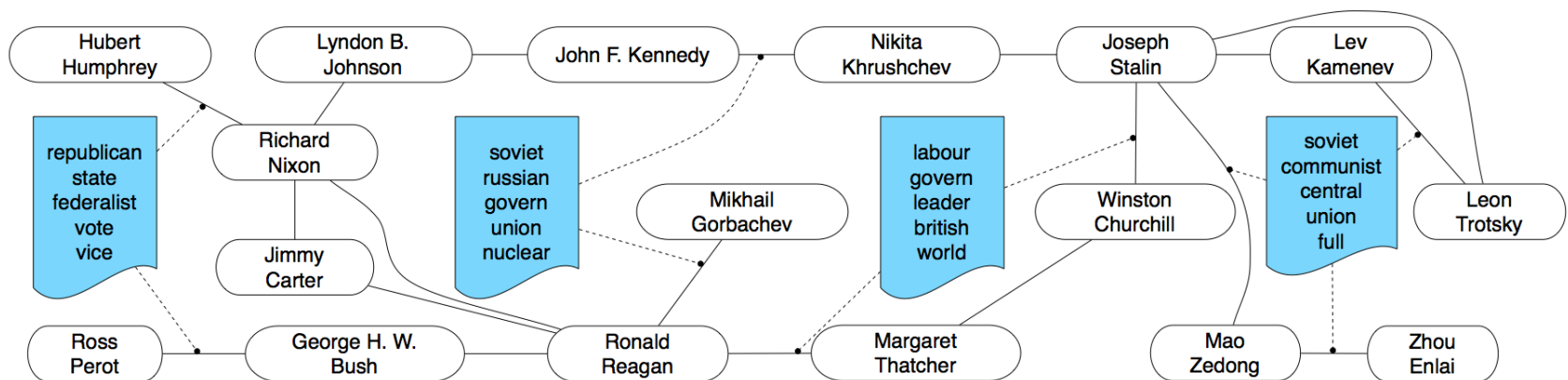
Học không giám sát: ví dụ (2)

■ Trends detection

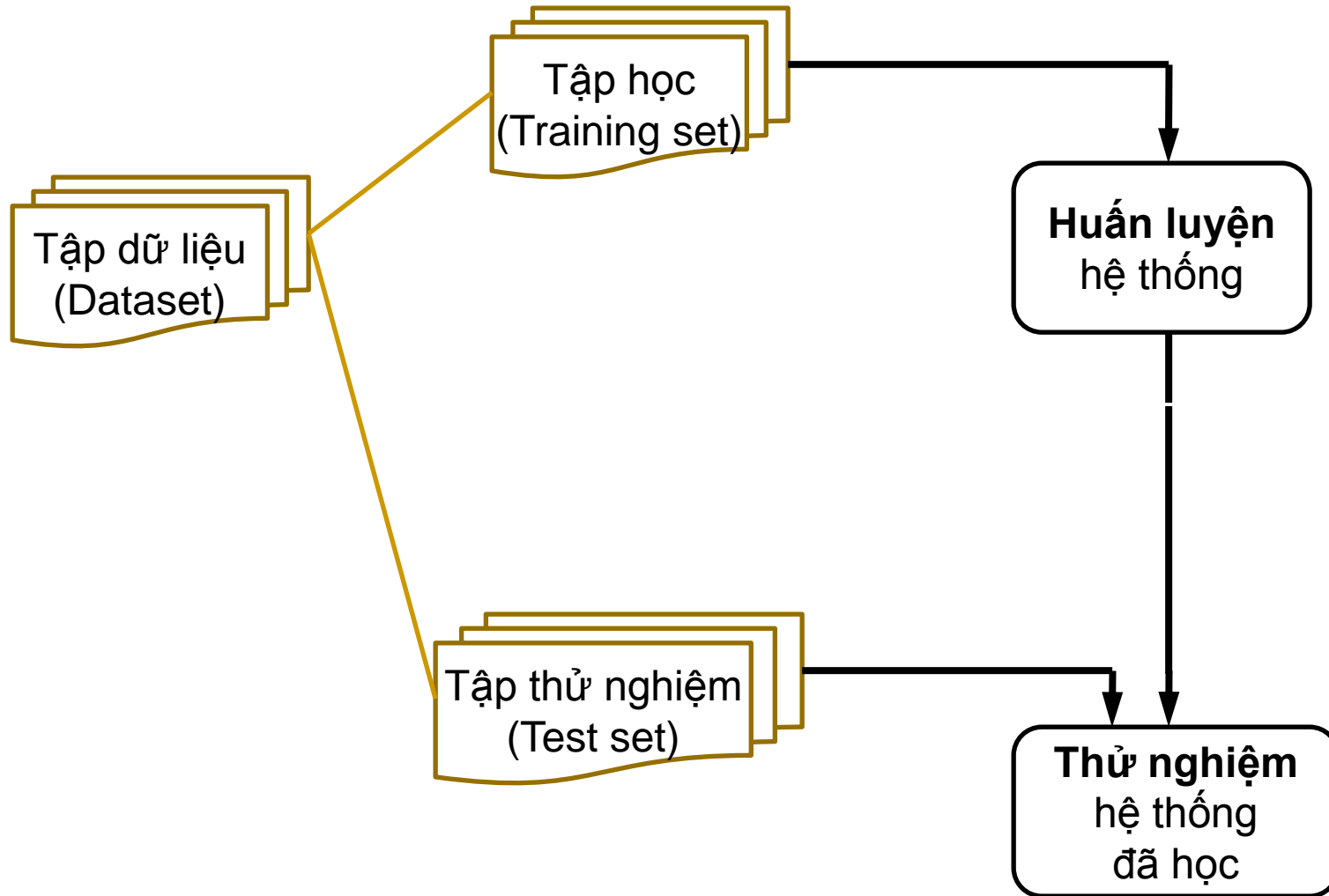
- Phát hiện xu hướng, thị yếu,...



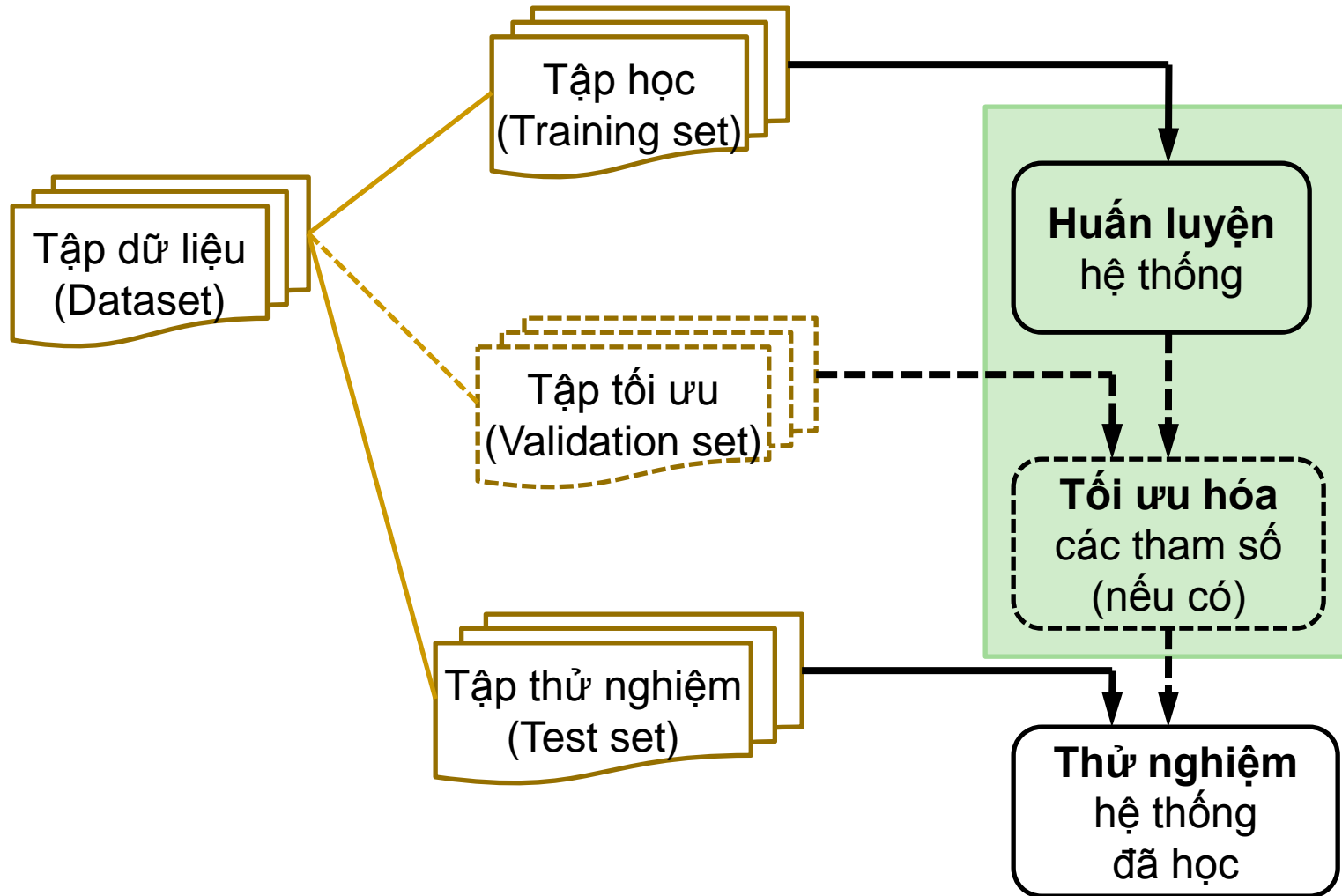
■ Entity-interaction analysis



Quá trình học máy: cơ bản



Quá trình học máy: toàn diện



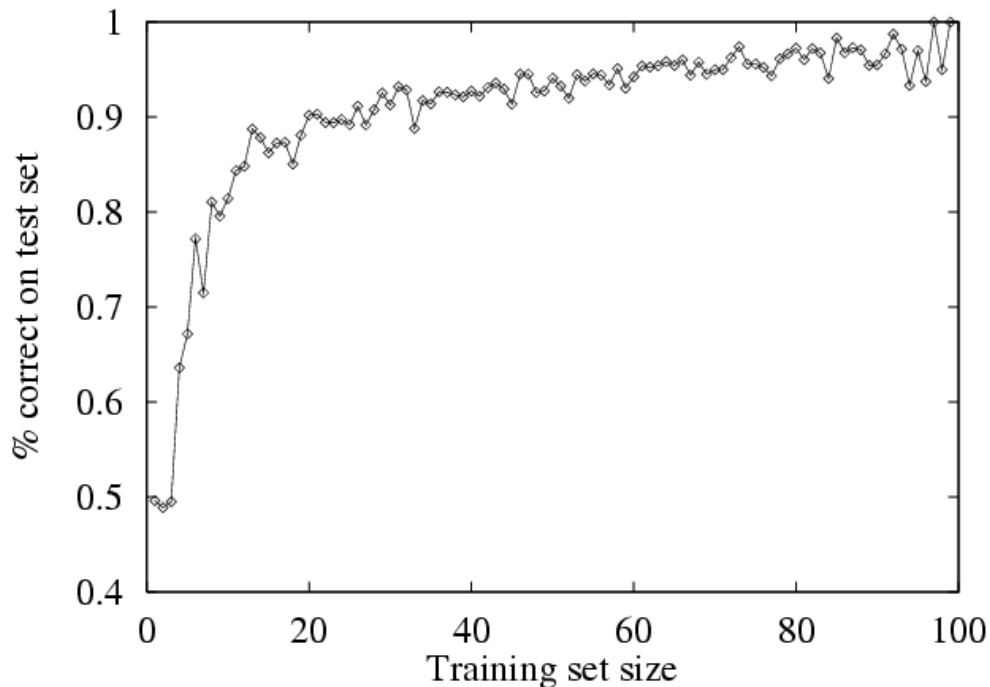
Đánh giá và lựa chọn mô hình

- Cho trước tập quan sát D , ta cần lựa chọn tham số λ (model selection) cho phương pháp học A và đánh giá (assessment) chất lượng tổng thể của A .
 - Chọn tập hữu hạn S mà chứa các giá trị tiềm năng cho λ .
 - Chọn độ đo P để đánh giá hiệu năng.
 - Chia tập D thành 3 tập rời nhau: D_{train} , $T_{\text{validation}}$, và T_{test}
 - Với mỗi giá trị $\lambda \in S$:
 - Học A từ tập học D_{train} với tham số đầu vào λ . Đo hiệu năng trên tập $T_{\text{validation}} \rightarrow$ thu được P_λ
 - Chọn λ^* mà có P_λ tốt nhất.
 - Huấn luyện A trên tập $D_{\text{train}} \cup T_{\text{validation}}$, với tham số đầu vào λ^* .
 - Đo hiệu năng của hệ thống trên tập T_{test} .
- Có thể thay Hold-out bằng kỹ thuật khác (cross-validation).

Đo độ chính xác

- Làm sao để biết $h \approx f$?
- Sử dụng lý thuyết tính toán
 1. Thử giả thiết h trên 1 tập các ví dụ mới (tập thử) (sử dụng cùng 1 mức độ phân bố các mẫu như tập luyện)

Learning curve = % chính xác trên tập thử, sử dụng hàm xây dựng trên tập luyện



Thiết kế một hệ thống học (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 nên tương thích với (đại diện cho) các ví dụ sẽ được làm việc bởi hệ thống trong tương lai (future test examples)
 - Xác định hàm mục tiêu (giả thiết, khái niệm) cần học
 - $F: X \rightarrow \{0,1\}$
 - $F: X \rightarrow$ Một tập các nhãn lớp
 - $F: X \rightarrow \mathbb{R}^+$ (miền các giá trị số thực dương)
 - ...
-

Thiết kế một hệ thống học (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 (nhiều) 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?
-

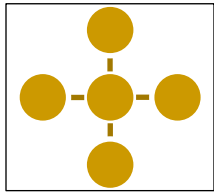
Các vấn đề trong Học máy (4)

- Khả năng/giới hạn học (Learnability)
 - 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 (generalization) của hệ thống?
 - 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
-

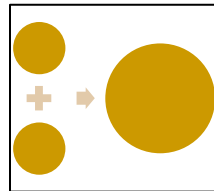
Phương pháp phân loại

Các bạn phân loại thể nào?

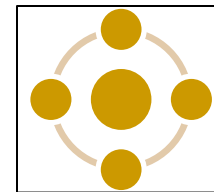
Class a



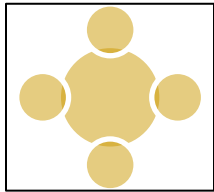
Class b



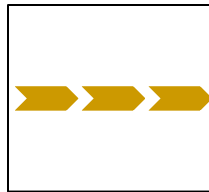
Class a



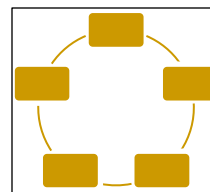
Class a



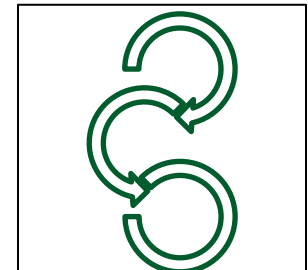
??



Class a



Class b



Học dựa trên các láng giềng gần nhất

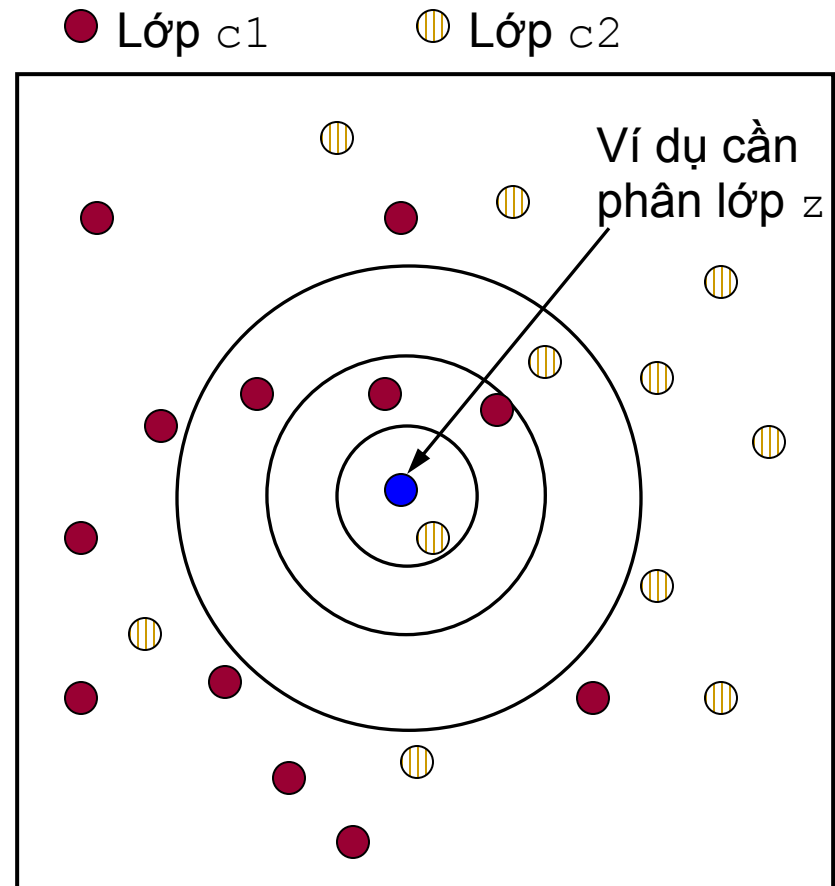
- ***K-nearest neighbors*** (k-NN) là một trong số các phương pháp phổ biến trong học máy. Vài tên gọi khác như:
 - Instance-based learning
 - Lazy learning
 - Memory-based learning
 - Ý tưởng của phương pháp
 - Không xây dựng một mô hình (mô tả) rõ ràng cho hàm mục tiêu cần học.
 - Quá trình học chỉ lưu lại các dữ liệu huấn luyện.
 - Việc dự đoán cho một quan sát mới sẽ dựa vào các hàng xóm gần nhất trong tập học.
-

k-NN

- Hai thành phần chính:
 - Độ đo tương đồng (similarity measure/distance) giữa các đối tượng.
 - Các hàng xóm sẽ dùng vào việc phán đoán.
 - *Trong một số điều kiện thì k-NN có thể đạt mức lỗi tối ưu Bayes (mức lỗi mong muốn của bất kỳ phương pháp nào)* [Gyader and Hengartner, JMLR 2013]
 - Thậm chí khi chỉ dùng 1 hàng xóm gần nhất thì nó cũng có thể đạt đến mức lỗi tối ưu Bayes. [Kontorovich & Weiss, AISTATS 2015]
-

Ví dụ: bài toán phân lớp

- 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 k-NN cho phân lớp

- Mỗi ví dụ học x được biểu diễn bởi 2 thành phần:
 - Mô tả của ví dụ: $x = (x_1, x_2, \dots, x_n)$, 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
 - 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 trong $NB(z)$
-

Giải thuật k-NN cho hồi quy

- Mỗi ví dụ học x được biểu diễn bởi 2 thành phần:
 - Mô tả của ví dụ: $x = (x_1, x_2, \dots, x_n)$, 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 :
$$y_z = \frac{1}{k} \sum_{x \in NB(z)} y_x$$

k-NN: Các vấn đề cốt lõi

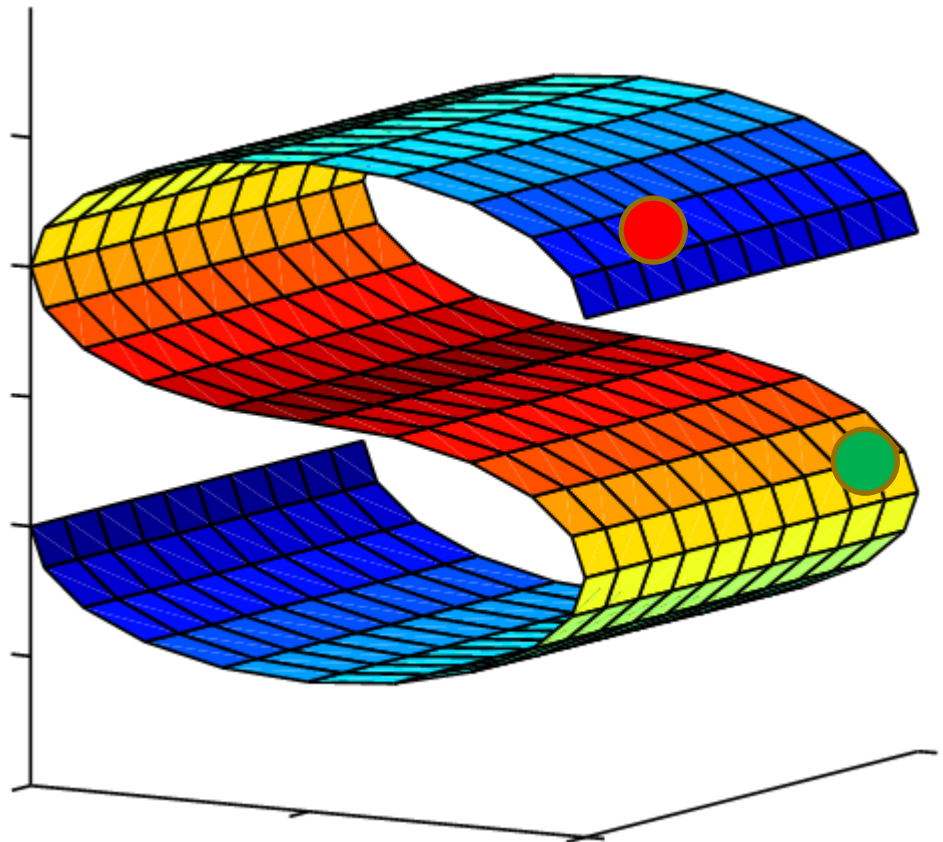


Suy
nghĩ
khác
nhau!

k-NN: Các vấn đề cốt lõi

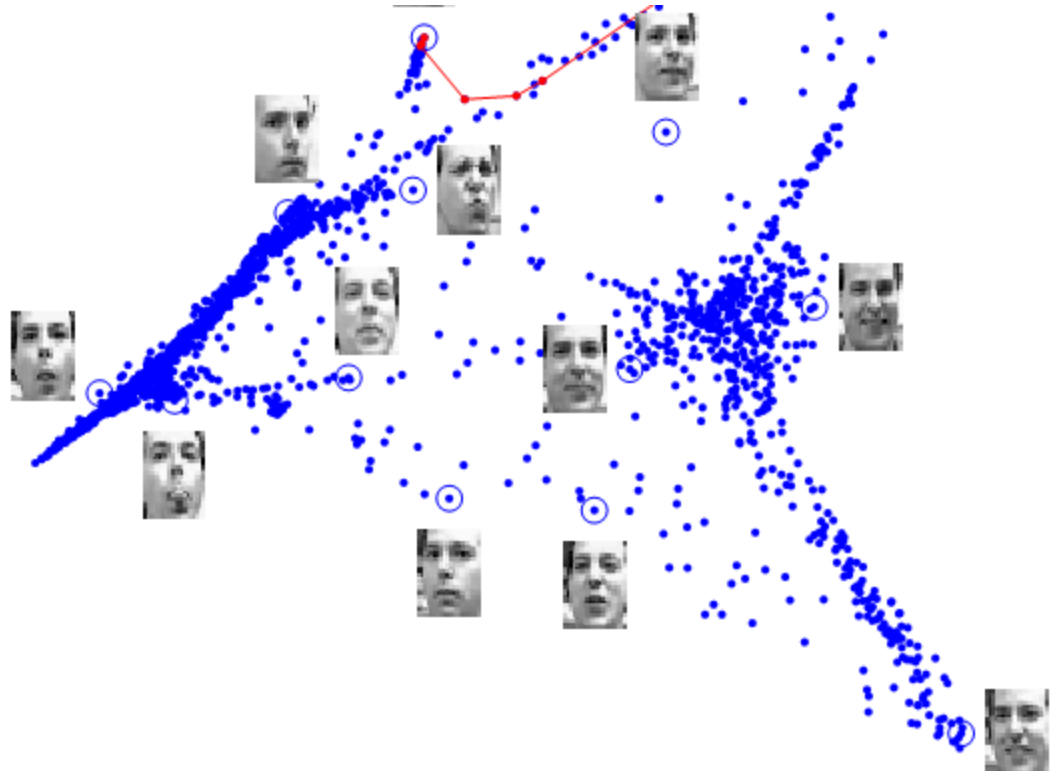
■ Hàm khoảng cách

- Mỗi hàm sẽ tương ứng với một cách nhìn về dữ liệu.
- Vô hạn hàm!!!
- Chọn hàm nào?



k-NN: Các vấn đề cốt lõi

- Chọn tập láng giềng $NB(z)$
 - Chọn bao nhiêu láng giềng?
 - Giới hạn chọn theo vùng?



k-NN: một hay nhiều láng giềng?

- Về lý thuyết thì 1-NN cũng có thể là một trong số các phương pháp tối ưu.
- k-NN là một phương pháp tối ưu Bayes nếu gặp một số điều kiện, chẳng hạn: y bị chặn, cỡ M của tập học lớn, hàm hồi quy liên tục, và

$$k \rightarrow \infty, (k/M) \rightarrow 0, (k/\log M) \rightarrow +\infty$$

- Trong thực tiễn ta nên lấy nhiều hàng xóm ($k > 1$) khi cần phân lớp/dự đoán, nhưng không quá nhiều. Lý do:
 - Tránh ảnh hưởng của lỗi/nhiều nếu chỉ dùng 1 hàng xóm.
 - Nếu quá nhiều hàng xóm thì sẽ phá vỡ cấu trúc tiềm ẩn trong dữ liệu.
-

Hàm tính khoảng cách (1)

■ Hàm tính khoảng cách d

- Đóng vai trò rất quan trọng trong phương pháp học dựa trên các 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 ($x_i \in \mathbb{R}$)
 - *Hàm khoảng cách Hamming*: Dành cho các bài toán có các thuộc tính đầu vào là kiểu nhị phân ($x_i \in \{0, 1\}$)
-

Hàm tính khoảng cách (2)

- Các hàm tính khoảng cách hình học (Geometry distance functions)

- Hàm Minkowski (p -norm):

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

- Hàm Manhattan ($p = 1$):

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

- Hàm Euclid ($p = 2$):

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

- Hàm Chebyshev ($p = \infty$):

$$\begin{aligned} d(x, z) &= \lim_{p \rightarrow \infty} \left(\sum_{i=1}^n |x_i - z_i|^p \right)^{1/p} \\ &= \max_i |x_i - z_i| \end{aligned}$$

Hàm tính khoảng cách (3)

■ Hàm khoảng cách Hamming

- Đối với các thuộc tính đầu vào là kiểu nhị phân ()

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

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

k-NN: Ưu nhược điểm

■ Các ưu điểm

- Chi phí thấp cho quá trình huấn luyện (chỉ việc lưu lại các ví dụ học)
- Hoạt động tốt với các bài toán phân loại gồm nhiều lớp
 - Không cần phải học c bộ phân loại cho c lớp
- Phương pháp học k-NN ($k \gg 1$) có khả năng xử lý nhiễu cao
 - Phân loại/dự đoán được thực hiện dựa trên k láng giềng gần nhất
- Rất Linh động trong việc chọn hàm khoảng cách.
 - Có thể dùng độ tương tự (similarity): cosine
 - Có thể dùng độ đo khác, chẳng hạn Kullback-Leibler divergence, Bregman divergence, ...

■ Các nhược điểm

- Phải lựa chọn hàm tính khoảng cách (sự khác biệt) thích hợp với bài toán
 - Chi phí tính toán (thời gian, bộ nhớ) cao tại thời điểm phân loại/dự đoán
 - Có thể cho kết quả kém/sai với các thuộc tính không liên quan
-

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

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

- $P(h)$: Xác suất trước (tiên nghiệm) của giả thiết h
 - $P(D)$: Xác suất trước (tiên nghiệm) của việc quan sát được dữ liệu D
 - $P(D | h)$: Xác suất (có điều kiện) của việc quan sát được dữ liệu D , nếu biết giả thiết h là đúng. (likelihood)
 - $P(h | D)$: Xác suất (hậu nghiệm) của giả thiết h là đúng, nếu quan sát được dữ liệu D
 - **Nhiều phương pháp phân loại dựa trên xác suất sẽ sử dụng xác suất hậu nghiệm (*posterior probability*) này!**
-

Định lý Bayes: Ví dụ (1)

Giả sử chúng ta có tập dữ liệu sau (dự đoán 1 người có chơi tennis)?

Ngày	Ngoài trời	Nhiệt độ	Độ ẩm	Gió	Chơi tennis
N1	Nắng	Nóng	Cao	Yếu	Không
N2	Nắng	Nóng	Cao	Mạnh	Không
N3	Âm u	Nóng	Cao	Yếu	Có
N4	Mưa	Bình thường	Cao	Yếu	Có
N5	Mưa	Mát mẻ	Bình thường	Yếu	Có
N6	Mưa	Mát mẻ	Bình thường	Mạnh	Không
N7	Âm u	Mát mẻ	Bình thường	Mạnh	Có
N8	Nắng	Bình thường	Cao	Yếu	Không
N9	Nắng	Mát mẻ	Bình thường	Yếu	Có
N10	Mưa	Bình thường	Bình thường	Yếu	Có
N11	Nắng	Bình thường	Bình thường	Mạnh	Có
N12	Âm u	Bình thường	Cao	Mạnh	Có

Định lý Bayes: Ví dụ (2)

- Dữ liệu D . *Ngoài trời là nắng và Gió là mạnh*
 - Giả thiết (phân loại) h . Anh ta chơi tennis
 - Xác suất trước $P(h)$. Xác suất rằng anh ta chơi tennis (bất kể *Ngoài trời* như thế nào và *Gió* ra sao)
 - Xác suất trước $P(D)$. Xác suất rằng *Ngoài trời là nắng và Gió là mạnh*
 - $P(D|h)$. Xác suất *Ngoài trời là nắng và Gió là mạnh*, nếu biết rằng anh ta chơi tennis
 - $P(h|D)$. Xác suất anh ta chơi tennis, nếu biết rằng *Ngoài trời là nắng và Gió là mạnh*
-

Xác suất hậu nghiệm cực đại (MAP)

- 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 có xác suất hậu nghiệm cực đại (**Maximum a posteriori – MAP**)

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

$$h_{MAP} = \arg \max_{h \in H} \frac{P(D | h) \cdot P(h)}{P(D)} \quad (\text{bởi định lý Bayes})$$

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

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 suất có điều kiện: $P(h_1 | D)$, $P(h_2 | D)$
 - Giả thiết có thể nhất $h_{\text{MAP}} = h_1$ nếu $P(h_1 | D) \geq P(h_2 | D)$; ngược lại thì $h_{\text{MAP}} = h_2$
 - Vì vậy, cần tính 2 biểu thức: $P(D | h_1) \cdot P(h_1)$ và $P(D | h_2) \cdot P(h_2)$, và đưa ra quyết định tương ứng
 - Nếu $P(D | h_1) \cdot P(h_1) \geq P(D | h_2) \cdot P(h_2)$, thì kết luận là anh ta chơi tennis
 - Ngược lại, thì kết luận là anh ta không chơi tennis
-

Đánh giá khả năng có thể nhất (MLE)

- Phương pháp MAP: Với một tập các giả thiết có thể H , cần tìm một giả thiết cực đại hóa giá trị: $P(D|h) \cdot P(h)$
- Giả sử (assumption) trong phương pháp **đánh giá khả năng có thể 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 có thể (likelihood)* của dữ liệu D đối với h
- Giả thiết có khả năng nhất (maximum likelihood hypothesis)

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

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 $P(D|h_2)$
 - $P(\text{Outlook}=\text{Sunny}, \text{Wind}=\text{Strong}|h_1) = 1/8$
 - $P(\text{Outlook}=\text{Sunny}, \text{Wind}=\text{Strong}|h_2) = 1/4$
- Giả thiết MLE $h_{\text{MLE}}=h_1$ nếu $P(D|h_1) \geq P(D|h_2)$; và ngược lại thì $h_{\text{MLE}}=h_2$
 - Bởi vì $P(\text{Outlook}=\text{Sunny}, \text{Wind}=\text{Strong}|h_1) < P(\text{Outlook}=\text{Sunny}, \text{Wind}=\text{Strong}|h_2)$, 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 vector n chiều: (x_1, x_2, \dots, x_n)
 - Một tập xác định các nhãn lớp: $C = \{c_1, c_2, \dots, c_m\}$
 - Với một ví dụ (mới) z , thì z sẽ được phân vào lớp nào?
- Mục tiêu: Xác định phân lớp có thể (phù hợp) nhất đối với z

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

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

$$c_{MAP} = \arg \max_{c_i \in C} \frac{P(z_1, z_2, \dots, z_n | c_i) \cdot P(c_i)}{P(z_1, z_2, \dots, z_n)} \quad (\text{bởi định lý Bayes})$$

Phân loại Naïve Bayes (2)

- Để tìm được phân lớp có thể nhất đối với $z \dots$

$$c_{MAP} = \arg \max_{c_i \in C} P(z_1, z_2, \dots, z_n | c_i) \cdot P(c_i) \quad (P(z_1, z_2, \dots, z_n) \text{ là như nhau với các lớp})$$

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

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

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

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

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 tiên nghiệm: $P(c_i)$
 - 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:

$$P(c_i) \cdot \prod_{j=1}^n P(x_j | c_i)$$

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

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

Phân loại 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 loại Naïve Bayes: Ví dụ (2)

■ Biểu diễn bài toán phân loại

- $z = (\text{Age}=\text{Young}, \text{Income}=\text{Medium}, \text{Student}=\text{Yes}, \text{Credit_Rating}=\text{Fair})$
- 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

- | | |
|---|--|
| • $P(\text{Age}=\text{Young} c_1) = 2/9;$ | $P(\text{Age}=\text{Young} c_2) = 3/5$ |
| • $P(\text{Income}=\text{Medium} c_1) = 4/9;$ | $P(\text{Income}=\text{Medium} c_2) = 2/5$ |
| • $P(\text{Student}=\text{Yes} c_1) = 6/9;$ | $P(\text{Student}=\text{Yes} c_2) = 1/5$ |
| • $P(\text{Credit_Rating}=\text{Fair} c_1) = 6/9;$ | $P(\text{Credit_Rating}=\text{Fair} c_2) = 2/5$ |
-

Phân loại 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
$$P(z|c_1) = P(\text{Age}=\text{Young}|c_1).P(\text{Income}=\text{Medium}|c_1).P(\text{Student}=\text{Yes}|c_1).P(\text{Credit_Rating}=\text{Fair}|c_1) = (2/9).(4/9).(6/9).(6/9) = 0.044$$
 - Đối với phân lớp c_2
$$P(z|c_2) = P(\text{Age}=\text{Young}|c_2).P(\text{Income}=\text{Medium}|c_2).P(\text{Student}=\text{Yes}|c_2).P(\text{Credit_Rating}=\text{Fair}|c_2) = (3/5).(2/5).(1/5).(2/5) = 0.019$$
 - 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 loại Naïve Bayes: Vấn đề (1)

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

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

- 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: $p = 1/k$, nếu thuộc tính f_j có k giá trị
- 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 loại 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:

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

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

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

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

Phân loại văn bản bằng NB (1)

■ Biểu diễn bài toán phân loại văn bản

- Tập học D , 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 , trích ra tập các từ khóa $T = \{t_j\}$
- Gọi D_{c_i} là tập các văn bản trong D 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

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

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

Phân loại văn bản bằng NB (2)

■ Giai đoạn phân lớp đối với một văn bản mới d

- Từ văn bản d , trích ra tập T_d gồm các từ khóa (keywords) t_j đã được định nghĩa trong tập 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

$$P(t_j \text{ ở vị trí } k | c_i) = P(t_j \text{ ở vị trí } m | c_i), \quad \forall k, m$$

- Đối với mỗi phân lớp c_i , tính xác suất hậu nghiệm của văn bản d đối với c_i

$$P(c_i) \cdot \prod_{t_j \in T_d} P(t_j | c_i)$$

- Phân lớp văn bản d thuộc vào lớp c^*

$$c^* = \arg \max_{c_i \in C} P(c_i) \cdot \prod_{t_j \in T_d} P(t_j | c_i)$$

Tài liệu tham khảo

- E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2010.
 - T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
 - T. M. Mitchell. *The discipline of machine learning*. CMU technical report, 2006.
 - 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.
 - A. Kontorovich and Weiss. *A Bayes consistent 1-NN classifier*. Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS). JMLR: W&CP volume 38, 2015.
 - A. Guyader, N. Hengartner. *On the Mutual Nearest Neighbors Estimate in Regression*. Journal of Machine Learning Research 14 (2013) 2361-2376.
 - L. Gottlieb, A. Kontorovich, and P. Nisnevitch. *Near-optimal sample compression for nearest neighbors*. Advances in Neural Information Processing Systems, 2014.
-