



Học viện Công nghệ Bưu chính Viễn thông
Khoa Công nghệ thông tin 1

Nhập môn trí tuệ nhân tạo

Giới thiệu về học máy

- ❑ Giới thiệu
- ❑ Học cây quyết định
- ❑ Phân loại Bayes đơn giản
- ❑ Học dựa trên ví dụ

Tài liệu tham khảo

- ▶ N. Nilsson. Introduction to machine learning
<http://ai.stanford.edu/people/nilsson/mlbook.html>
- ▶ T. Mitchell. Machine learning. McGraw-Hill, 1997.
- ▶ E. Alpaydin. Introduction to machine learning. MIT Press, 2004.
- ▶ M. Mohri, A. Rostamizadeh, A. Talwalkar. Foundations of Machine Learning. MIT Press, 2012.

Công cụ và dữ liệu

- ▶ Bộ công cụ Weka
 - <http://www.cs.waikato.ac.nz/~ml/weka>
- ▶ Kho dữ liệu mẫu UC Irvine
 - <http://www.ics.uci.edu/~mlearn/ML/Repository.html>

Một số ứng dụng của học máy (1/3)

- ▶ Những ứng dụng khó lập trình theo cách thông thường do không tồn tại hoặc khó giải thích kinh nghiệm, kỹ năng của con người
 - Nhận dạng chữ viết, âm thanh, hình ảnh
 - Lái xe tự động, thám hiểm sao Hỏa

- ▶ Chương trình máy tính có khả năng thích nghi: lời giải thay đổi theo thời gian hoặc theo tình huống cụ thể
 - Chương trình trợ giúp cá nhân
 - Định tuyến mạng

Một số ứng dụng của học máy (2/3)

- ▶ Khai phá (phân tích) dữ liệu
 - Hồ sơ bệnh án → tri thức y học
 - Dữ liệu bán hàng → quy luật kinh doanh



Một số ứng dụng của học máy (3/3)

- ▶ Hầu hết các ứng dụng trí tuệ nhân tạo ngày nay có sử dụng học máy

...

- Web search
- Speech recognition
- Handwriting recognition
- Machine translation
- Information extraction
- Document summarization
- Question answering
- Spelling correction
- Image recognition
- 3D scene reconstruction
- Human activity recognition
- Autonomous driving
- Music information retrieval
- Automatic composition
- Social network analysis

...

...

- Product recommendation
- Advertisement placement
- Smart-grid energy optimization
- Household robotics
- Robotic surgery
- Robot exploration
- Spam filtering
- Fraud detection
- Fault diagnostics
- AI for video games
- Character animation
- Financial trading
- Protein folding
- Medical diagnosis
- Medical imaging

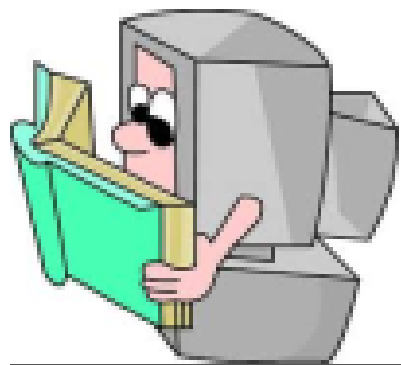
...

Học máy là gì?

- 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]

Học máy là gì?

- 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: Dữ liệu (lọc) những thư điện tử nào là thư rác (spam email)

P: Tỷ lệ % của các 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 (ví dụ: tập từ khóa) và nhãn lớp (thư thường/thư rác) tương ứng



Ví dụ bài toán học máy (2)

- **Phân loại các loại trang Web**

T: Phân loại các trang web theo các chủ đề đã định trước

P: Tỷ lệ % của các trang web được phân loại chính xác

E: Một tập các trang web, trong đó mỗi trang Web được biểu diễn bằng một tập thuộc tính và nhãn lớp là một chủ đề



Ví dụ bài toán học máy (3)

- **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



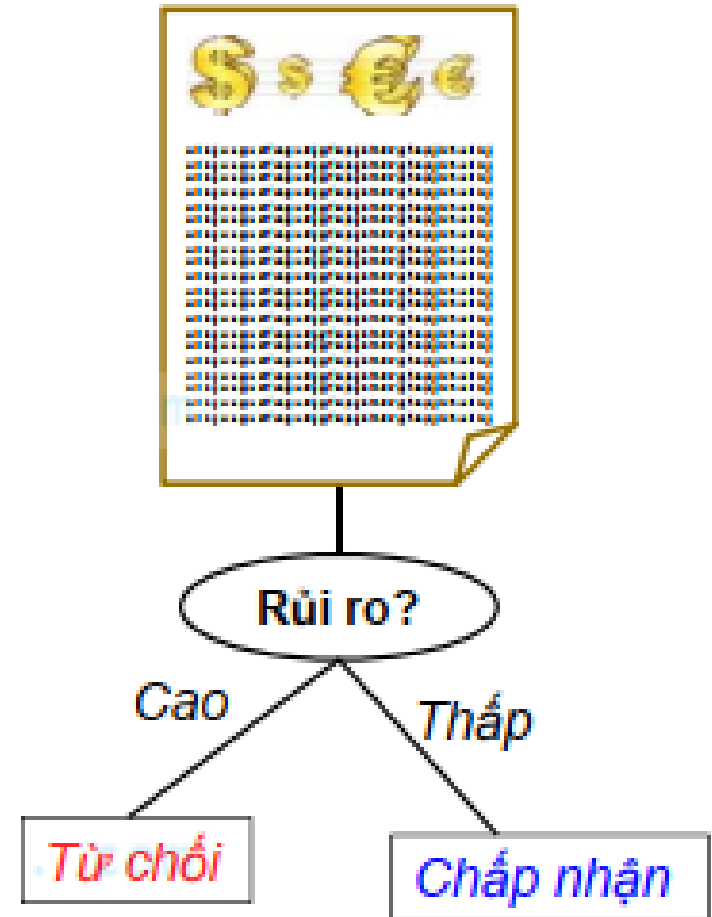
Ví dụ bài toán học máy (4)

- **Dự đoán rủi ro cho vay tài chính**

T: Xác định mức độ rủi ro (ví dụ: cao/thấp) đối với các hồ sơ xin vay tài chính

P: Tỷ lệ % các hồ sơ xin vay có mức độ rủi ro cao (không trả lại tiền vay) được xác định chính xác

E: Một tập các hồ sơ xin vay; mỗi hồ sơ được biểu diễn bởi một tập các thuộc tính và mức độ rủi ro (cao/thấp)



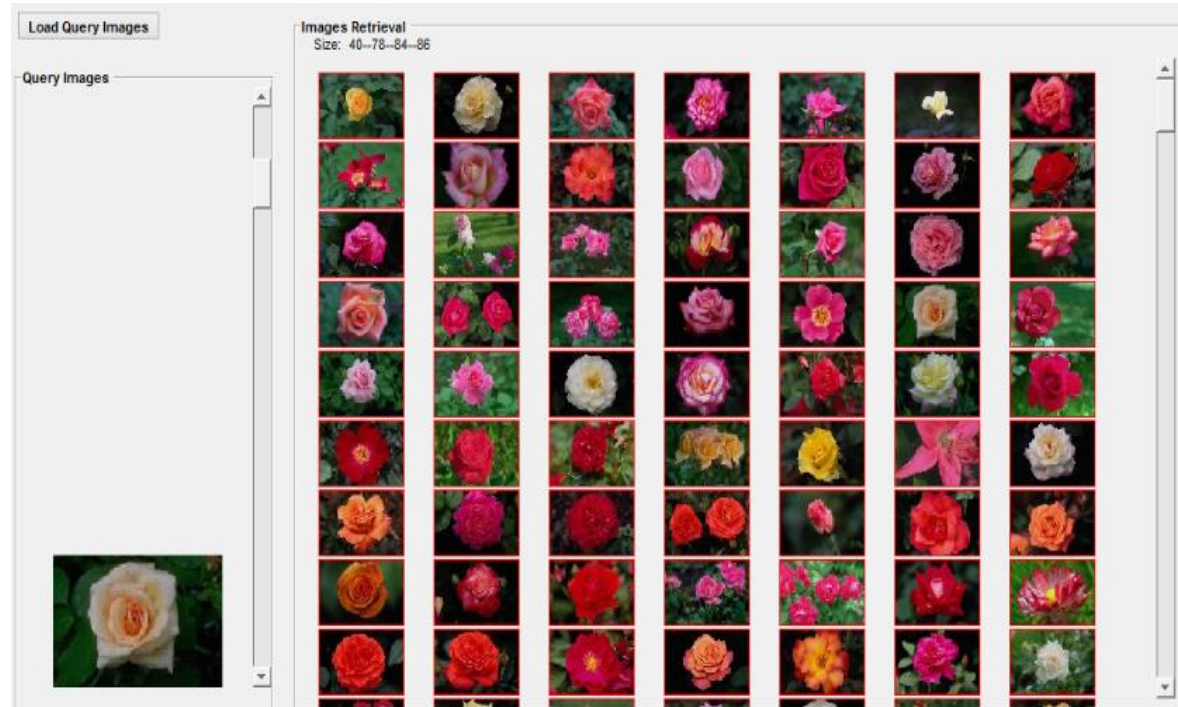
Ví dụ bài toán học máy (5)

- **Tra cứu ảnh**

T: Đưa ra những ảnh trong CSDL ảnh cùng mô tả tới một đối tượng của ảnh truy vấn người dùng đưa vào

P: Tỷ lệ % ảnh mô tả cùng đối tượng và số ảnh trả về

E: Một bộ sưu tập các chủ đề ảnh, mỗi ảnh được biểu diễn bởi véc tơ đặc trưng và thông tin chủ đề của ảnh (có thể có hoặc không)



Học máy (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)

Anh ta thích nghe



+



→ Trẻ hay Già ?

Học máy (2)

■ Học từ đâu?

- Từ các quan sát trong quá khứ (tập học – training set).
 $\{\{x_1, x_2, \dots, x_N\}; \{y_1, y_2, \dots, y_M\}\}$
- x_i là các quan sát của x trong quá khứ
- y_h là *nhãn (label)* hoặc *phản hồi (response)* hoặc *đầu ra (output)*

■ Sau khi đã học:

- Thu được một mô hình, kinh nghiệm, tri thức mới (f).
- Dùng nó để **suy diễn (infer)** hoặc **phán đoán (predict)** cho quan sát trong tương lai.
 $y_z = f(z)$

Một số khái niệm

- ▶ **Mẫu**, hay ví dụ (samples): là đối tượng cần xử lý (ví dụ phân loại)
 - Ví dụ: khi lọc thư rác thì mỗi thư là một mẫu
- ▶ **Mẫu** thường được mô tả bằng tập thuộc tính hay **đặc trưng** (features)
 - Ví dụ: trong chuẩn đoán bệnh, thuộc tính là triệu chứng của người bệnh, và các tham số khác như chiều cao, cân nặng, ...
- ▶ **Nhãn** phân loại (label): thể hiện loại của đối tượng mà ta cần dự đoán
 - Ví dụ: nhãn phân loại thư rác có thể là "rác" hoặc "bình thường"

Học có giám sát – Supervised learning

- **Học có giám sát (Supervised learning):** cần học một hàm $y = f(x)$ từ các cặp dữ liệu $\{(x_i, y_i) \in X \times Y\}$, từ tập dữ liệu này chúng ta cần tạo ra một **hàm ánh xạ** mỗi phần tử từ tập X sang một phần tử (xấp xỉ) ở tập Y .

$$y_i \approx f(x_i), \forall i = 1, 2, \dots, N$$

- **Phân loại (phân lớp):** nếu y chỉ nhận giá trị từ một tập rời rạc, chẳng hạn $\{\text{cá, cây, quả, mèo}\}$

- **Hồi quy:** nếu y nhận giá trị số thực

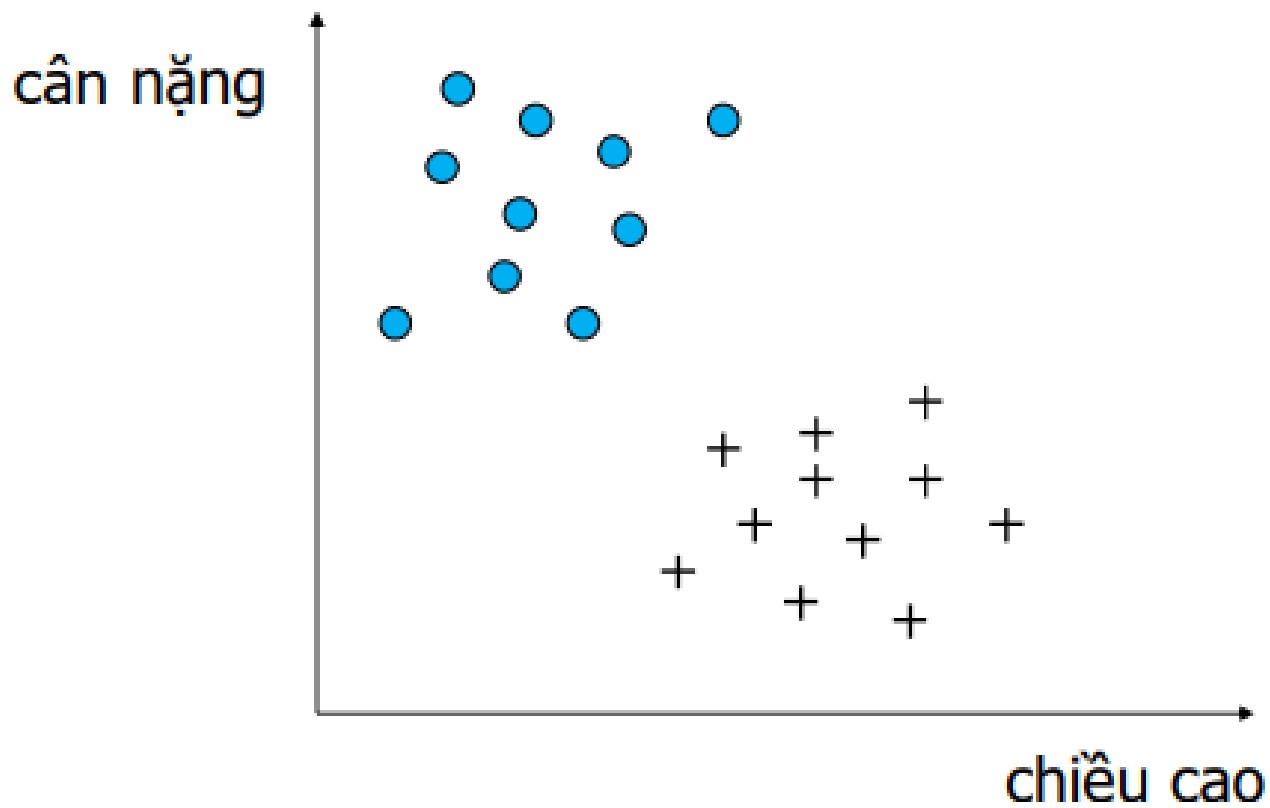
- Mục đích là xấp xỉ hàm **f** thật tốt để khi có một dữ liệu $x_{\text{mới}}$, chúng ta có thể tính được nhãn tương ứng của nó $y = f(x)$

Học có giám sát: Ví dụ

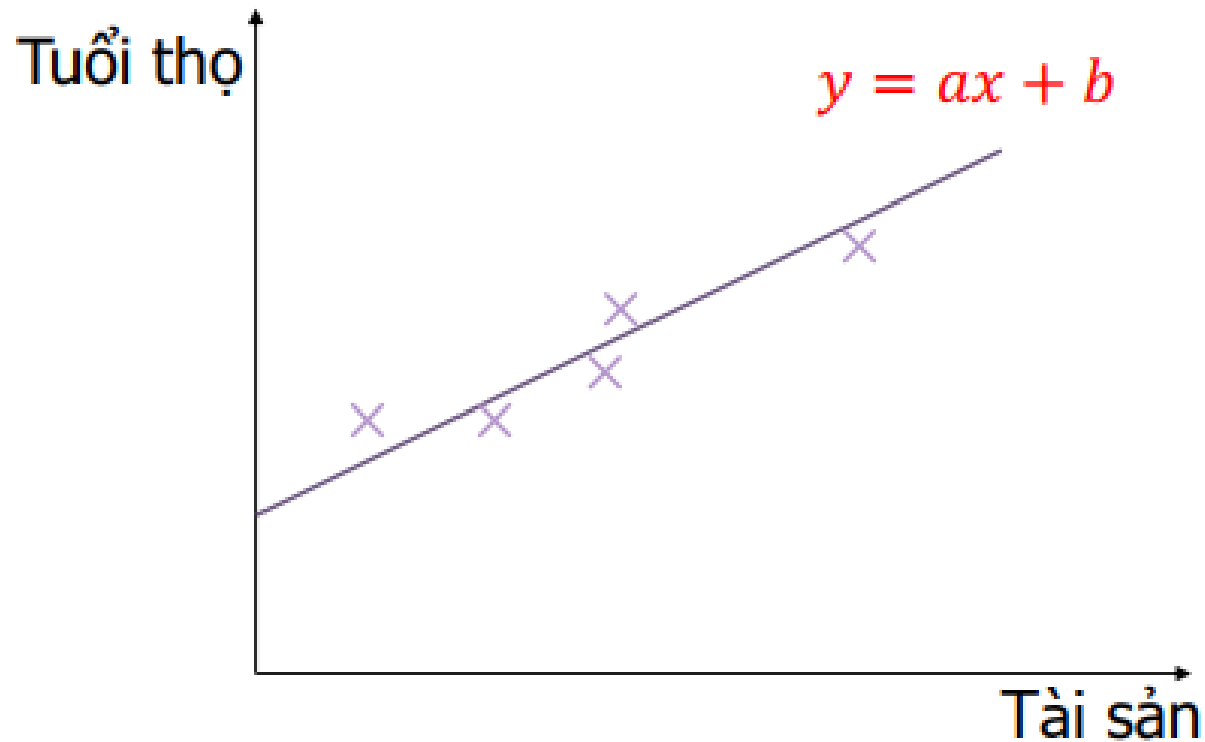
- Lọc thư rác
- Phân loại trang web
- Dự đoán rủi ro tài chính
- Dự đoán biến động chỉ số chứng khoán
- Phát hiện tấn công mạng



Phân lớp



Hồi quy (Regression)



Ứng dụng: dự đoán giá cả, lái xe,...

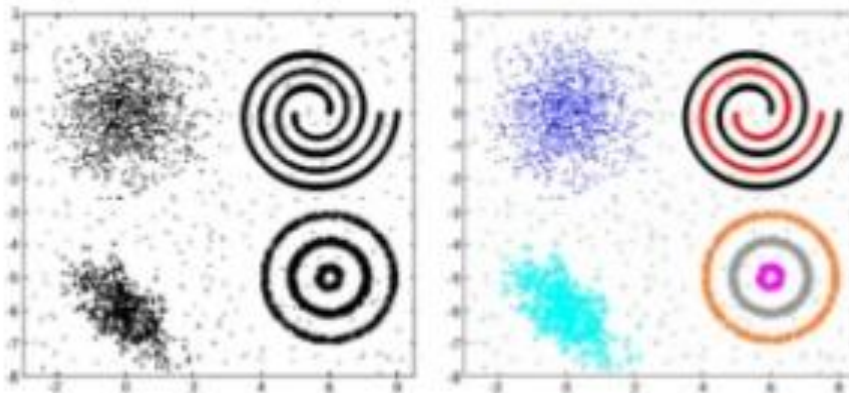


- Học không có giám sát (Unsupervised learning): cần học từ các tập dữ liệu $\{(x_i) \in X\}, \forall i = 1, 2, \dots, N$, là tập dữ liệu không có nhãn.
- - Có thể là các cụm dữ liệu
 - Có thể là cấu trúc ẩn

Học không có giám sát: Ví dụ

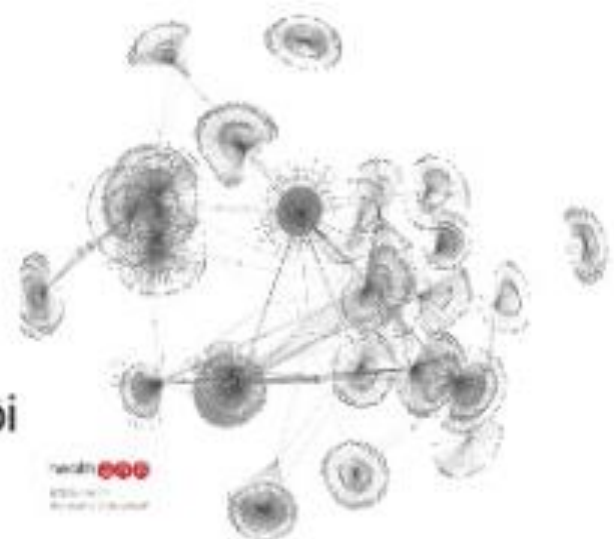
■ Phân cụm (clustering)

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



■ Community detection

- Phát hiện các cộng đồng trong mạng xã hội



Học luật kết hợp

- ▶ Ví dụ

- Phân tích giao dịch, mua bán (hóa đơn mua hàng)

- ▶ $P(Y|X)$

- Xác suất người mua hàng X còn mua hàng Y

- ▶ Ví dụ luật kết hợp

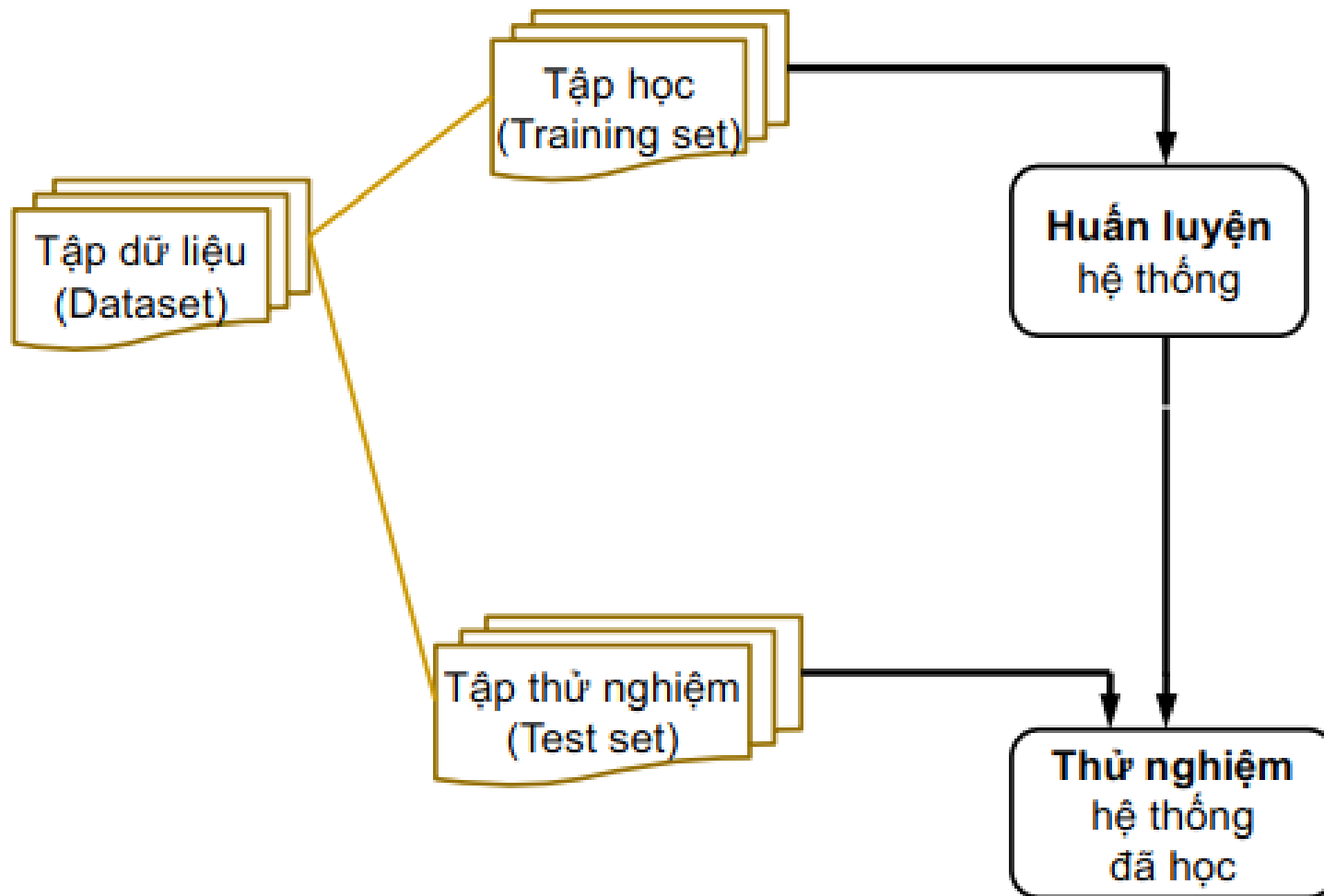
- Người mua bánh mì thường mua bơ
 - Người mua lạc rang thường mua bia

- ▶ Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước

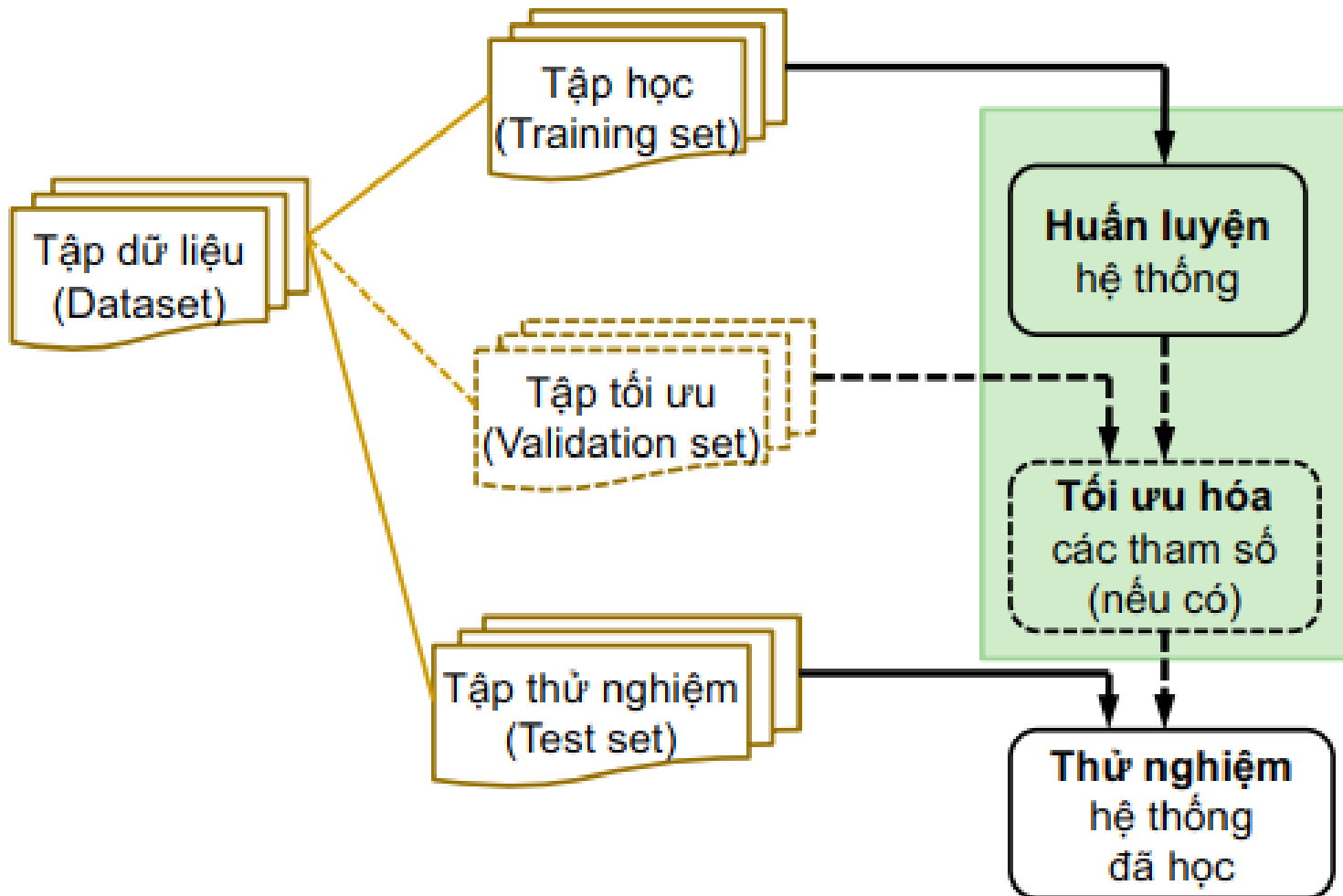
Học tăng cường

- ▶ Kinh nghiệm không được cho trực tiếp dưới dạng đầu vào / đầu ra
- ▶ Hệ thống nhận được một giá trị thưởng (reward) là kết quả cho một chuỗi hành động nào đó
- ▶ Thuật toán cần học cách hành động để cực đại hóa giá trị thưởng
- ▶ Ví dụ: học đánh cờ
 - Hệ thống không được chỉ cho nước đi nào là hợp lý cho từng tình huống cụ thể
 - Chỉ biết kết quả thắng thua sau một chuỗi nước đi

Quá trình học máy: **cơ bản**



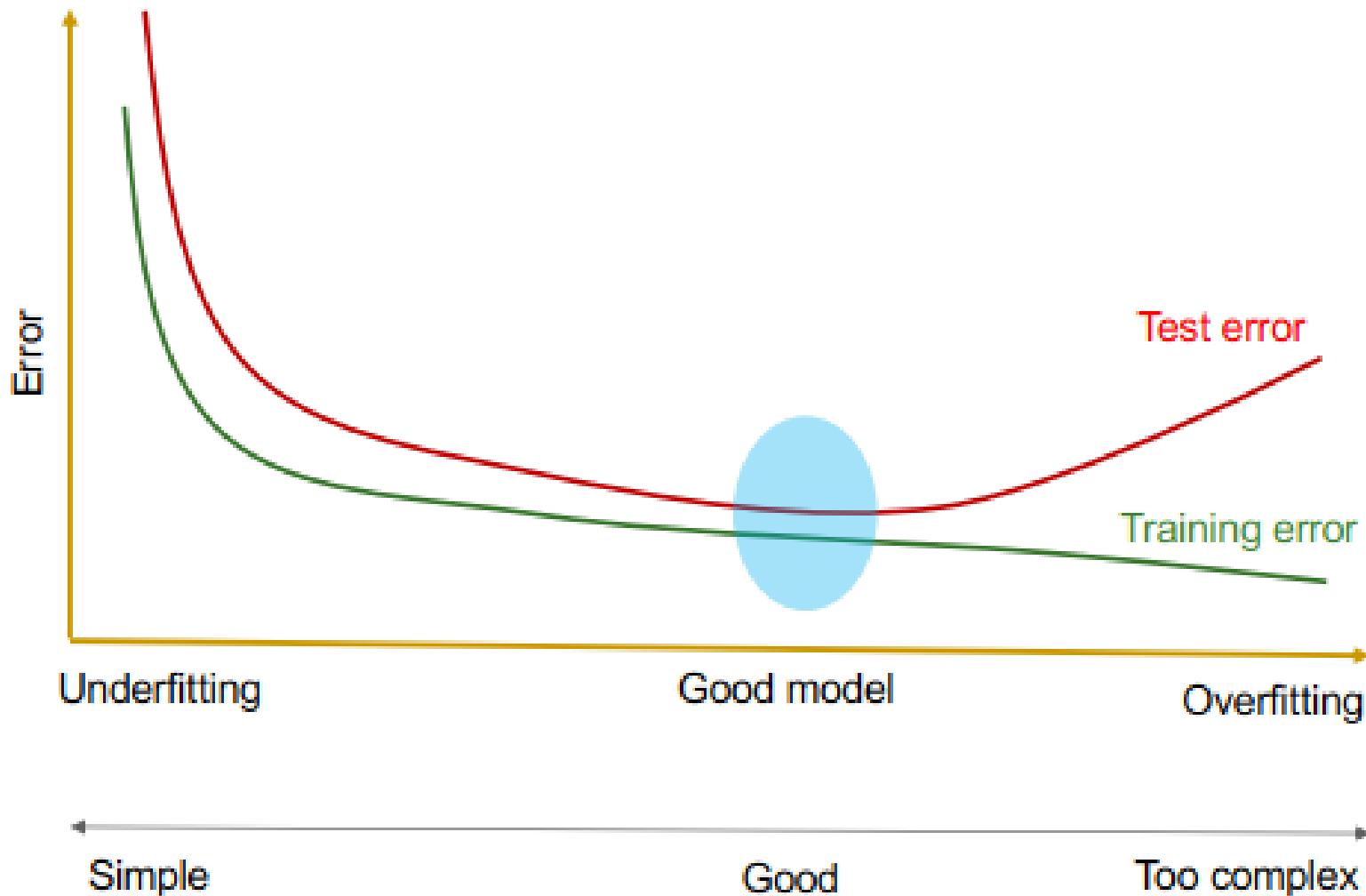
Quá trình học máy: **toàn diện**



Vấn đề overfitting

- Một hàm mục tiêu (một giả thiết) học được h sẽ được gọi là **quá khớp/quá phù hợp (overfit)** 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)
- Vài nguyên nhân:
 - Lỗi (nhiều) 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 overfitting: minh họa



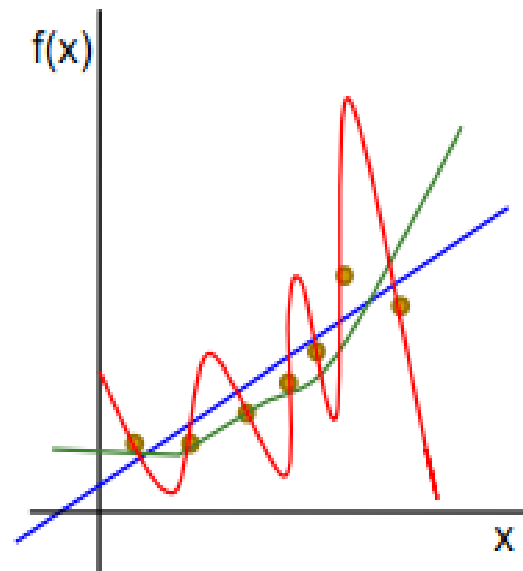
Vấn đề overfitting: giải pháp

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

- Hiệu chỉnh (Regularization):** hạn chế không gian 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?



Weka – Giới thiệu

- WEKA là một công cụ phần mềm viết bằng Java, phục vụ cho lĩnh vực học máy và khai phá dữ liệu
- Các tính năng chính:
 - Một tập các công cụ tiền xử lý dữ liệu, các giải thuật học máy, khai phá dữ liệu, và các phương pháp thí nghiệm đánh giá
 - Giao diện đồ họa (gồm cả tính năng hiển thị chuẩn hóa dữ liệu)
 - Môi trường cho phép so sánh các giải thuật học máy và khai phá dữ liệu

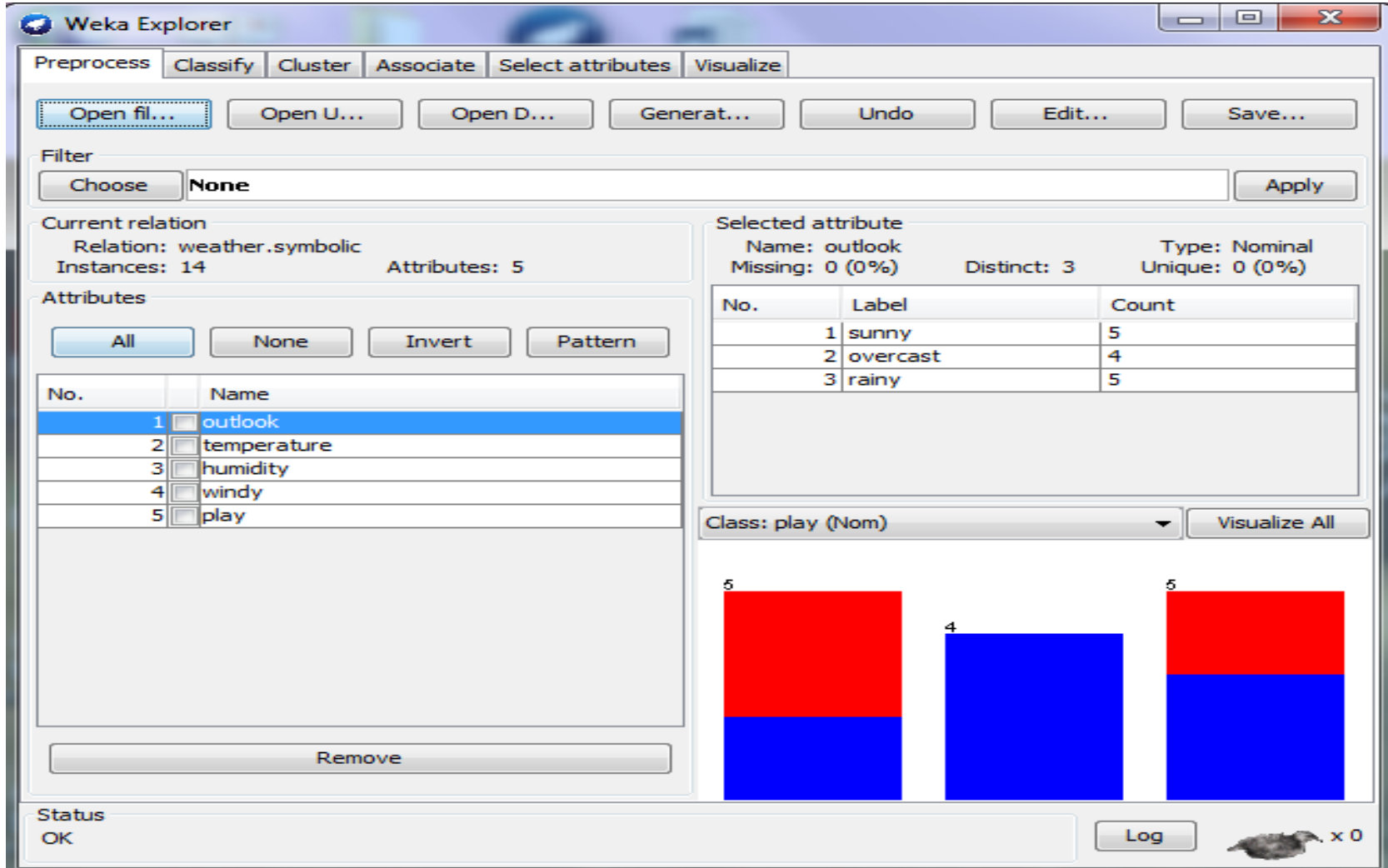
Weka – Giới thiệu



Weka – Các môi trường chính

- **Simple CLI:** Giao diện đơn giản kiểu dòng lệnh (MS-DOS)
- **Explorer:** Môi trường cho phép sử dụng tất cả các khả năng của WEKA để khám phá dữ liệu
- **Experimenter:** Môi trường cho phép tiến hành các thí nghiệm và thực hiện các kiểm tra thống kê (statistical tests) giữa các mô hình học máy
- **KnowledgeFlow:** Môi trường cho phép bạn tương tác đồ họa kéo/thả để thực hiện thiết kế các thành phần của một thí nghiệm

Weka – Các môi trường chính



Weka Explorer

Preprocess | **Classify** | Cluster | Associate | Select attributes | Visualize

Open fil... | Open U... | Open D... | Generat... | Undo | Edit... | Save...

Filter: Choose **None** [Apply]

Current relation:
Relation: weather.symbolic
Instances: 14 Attributes: 5

Attributes: All | None | Invert | Pattern

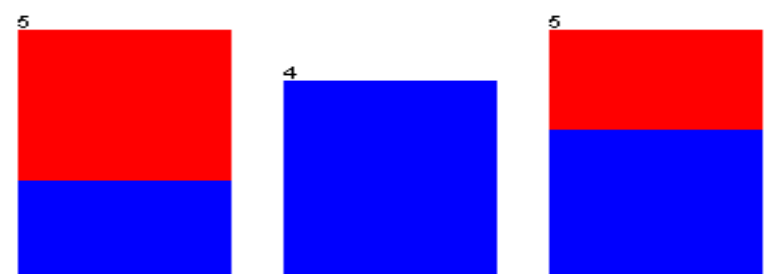
No.	Name
1	<input checked="" type="checkbox"/> outlook
2	<input type="checkbox"/> temperature
3	<input type="checkbox"/> humidity
4	<input type="checkbox"/> windy
5	<input type="checkbox"/> play

[Remove]

Selected attribute:
Name: outlook
Missing: 0 (0%) Distinct: 3 Type: Nominal
Unique: 0 (0%)

No.	Label	Count
1	sunny	5
2	overcast	4
3	rainy	5

Class: play (Nom) [Visualize All]



Status: OK [Log] x 0

Weka – Môi trường Explorer

- **Preprocess:** Để chọn và thay đổi (xử lý) dữ liệu làm việc
- **Classify:** Để huấn luyện và kiểm tra các mô hình học máy (phân loại, phân lớp, hồi quy/dự đoán)
- **Cluster:** Để học các nhóm từ dữ liệu (phân cụm)
- **Associate:** Để khám phá các luật kết hợp
- **Select attributes:** Để xác định và lựa chọn các thuộc tính liên quan (quan trọng) nhất của dữ liệu
- **Visualize:** Để xem (hiển thị) biểu đồ tương tác 2 chiều đối với dữ liệu

Nội dung

- ❑ Giới thiệu
- ❑ Học cây quyết định (decision tree learning)
- ❑ Phân loại Bayes đơn giản
- ❑ Học dựa trên ví dụ

Dữ liệu huấn luyện

Ngày	Trời	Nhiệt độ	Độ ẩm	Gió	Chơi tennis
D1	nắng	nóng	cao	yếu	không
D2	nắng	nóng	cao	mạnh	không
D3	u ám	nóng	cao	yếu	có
D4	mưa	trung bình	cao	yếu	có
D5	mưa	lạnh	bình thường	yếu	có
D6	mưa	lạnh	bình thường	mạnh	không
D7	u ám	lạnh	bình thường	mạnh	có
D8	nắng	trung bình	cao	yếu	không
D9	nắng	lạnh	bình thường	yếu	có
D10	mưa	trung bình	bình thường	yếu	có
D11	nắng	trung bình	bình thường	mạnh	có
D12	u ám	trung bình	cao	mạnh	có
D13	u ám	nóng	bình thường	yếu	có
D14	mưa	trung bình	cao	mạnh	không

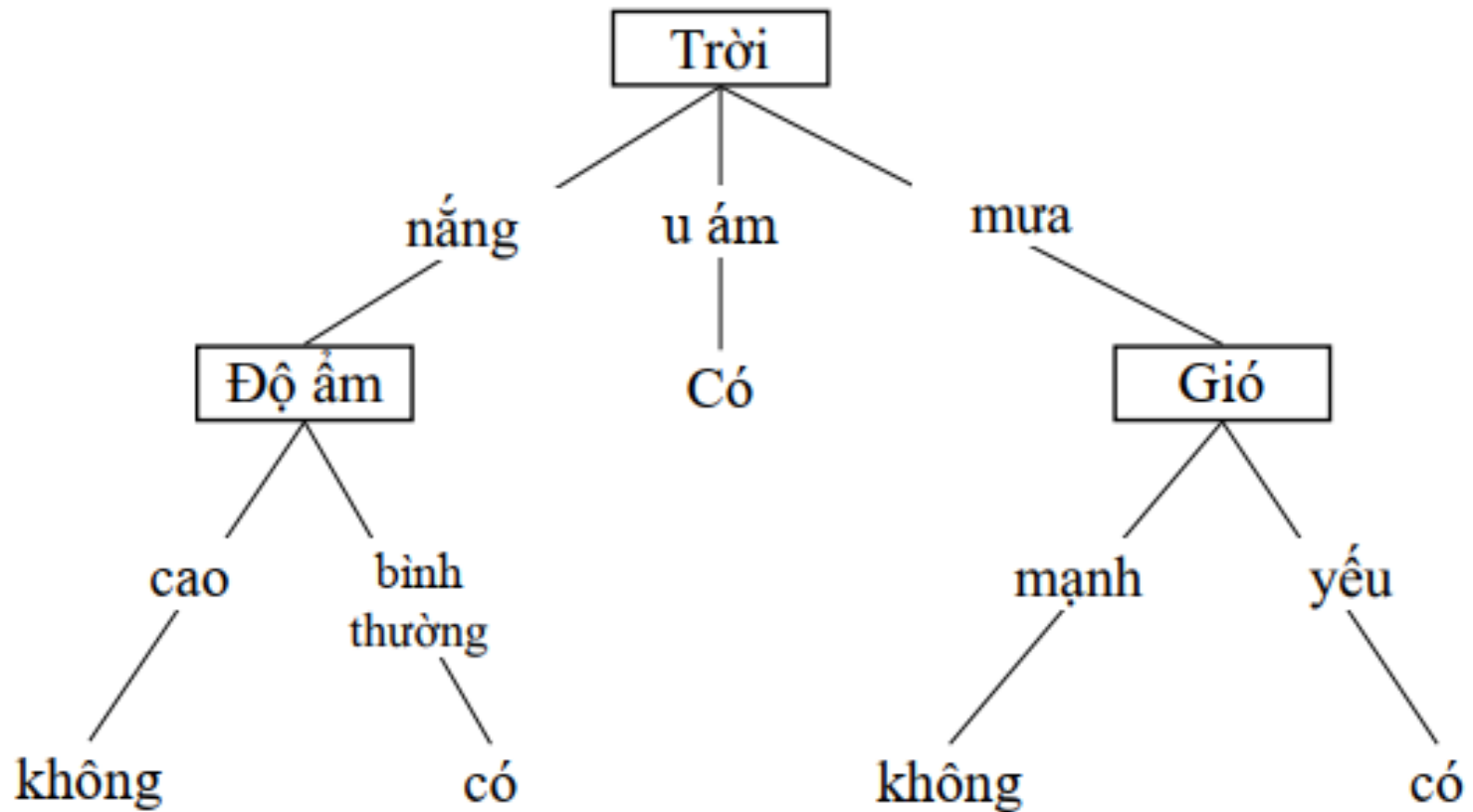
thuộc tính

nhãn

mẫu

Ngày	Trời	Nhiệt độ	Độ ẩm	Gió	Chơi tennis
D1	nắng	nóng	cao	yếu	không
D2	nắng	nóng	cao	mạnh	không
D3	u ám	nóng	cao	yếu	có
D4	mưa	trung bình	cao	yếu	có
D5	mưa	lạnh	bình thường	yếu	có
D6	mưa	lạnh	bình thường	mạnh	không
D7	u ám	lạnh	bình thường	mạnh	có
D8	nắng	trung bình	cao	yếu	không
D9	nắng	lạnh	bình thường	yếu	có
D10	mưa	trung bình	bình thường	yếu	có
D11	nắng	trung bình	bình thường	mạnh	có
D12	u ám	trung bình	cao	mạnh	có
D13	u ám	nóng	bình thường	yếu	có
D14	mưa	trung bình	cao	mạnh	không

Ví dụ cây quyết định



Cây quyết định là gì?

- ▶ **Là mô hình phân loại có dạng cây**
 - Mỗi nút trung gian (không phải lá) ứng với một phép kiểm tra thuộc tính, mỗi nhánh của nút ứng với một giá trị của thuộc tính tại nút đó
 - Mỗi nút lá ứng với một nhãn phân loại
- ▶ **Quá trình phân loại thực hiện như sau**
 - Mẫu phân loại đi từ gốc cây xuống dưới
 - Tại mỗi nút trung gian, thuộc tính tương ứng với nút được kiểm tra, tùy giá trị thuộc tính, mẫu được chuyển xuống nhánh tương ứng
 - Khi tới nút lá, mẫu được nhận nhãn phân loại của nút

Biểu diễn dưới dạng quy tắc

- ▶ Cây quyết định có thể biểu diễn tương đương dưới dạng các quy tắc logic
- ▶ Mỗi cây là tuyến của các quy tắc, mỗi quy tắc bao gồm các phép hội
- ▶ Ví dụ

$(\text{Trời} = \text{nắng} \wedge \text{Độ ẩm} = \text{bình_thường})$
 $\vee (\text{Trời} = \text{u_ám})$
 $\vee (\text{Trời} = \text{mưa} \wedge \text{Gió} = \text{yếu})$

Học cây quyết định

- ▶ Cây quyết định được học (xây dựng) từ dữ liệu huấn luyện
- ▶ Với mỗi bộ dữ liệu có thể xây dựng nhiều cây quyết định
 - Chọn cây nào?
- ▶ Quá trình học là quá trình tìm kiếm cây quyết định phù hợp với dữ liệu huấn luyện
 - Cho phép phân loại đúng dữ liệu huấn luyện

Thuật toán ID3

- ▶ Xây dựng lần lượt các nút của cây bắt đầu từ gốc
- ▶ Thuật toán
 - **Khởi đầu**: nút hiện thời là nút gốc chứa toàn bộ tập dữ liệu huấn luyện
 - Tại nút hiện thời n , lựa chọn thuộc tính
 - Chưa được sử dụng ở nút tổ tiên
 - Cho phép phân chia tập dữ liệu hiện thời thành các tập con **một cách tốt nhất**
 - Với mỗi giá trị thuộc tính được chọn thêm một nút con bên dưới
 - Chia các ví dụ ở nút hiện thời về các nút con theo giá trị thuộc tính được chọn
 - **Lặp** (đệ quy) cho tới khi
 - Tất cả các thuộc tính đã được sử dụng ở các nút phía trên, hoặc
 - Tất cả ví dụ tại nút hiện thời có cùng nhãn phân loại
 - Nhãn của nút được lấy theo đa số nhãn của ví dụ tại nút hiện thời

Lựa chọn thuộc tính tại mỗi nút thế nào?

Tiêu chuẩn chọn thuộc tính của ID₃

- ▶ Tại mỗi nút n
 - Tập (con) dữ liệu ứng với nút đó
 - Cần lựa chọn thuộc tính cho phép phân chia tập dữ liệu tốt nhất
- ▶ Tiêu chuẩn:
 - Dữ liệu sau khi phân chia càng đồng nhất càng tốt
 - Đo bằng độ tăng thông tin (Information Gain - IG)
 - **Chọn thuộc tính có độ tăng thông tin lớn nhất**
 - IG dựa trên entropy của tập (con) dữ liệu

Entropy

- ▶ Trường hợp tập dữ liệu S có 2 loại nhãn: đúng (+) hoặc sai (-)

$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

p_+ : % số mẫu đúng, p_- : % số mẫu sai

- ▶ Trường hợp tổng quát: có C loại nhãn

$$\text{Entropy}(S) = \sum_{i=1}^C -p_i \log_2 p_i$$

p_i : % ví dụ của S thuộc loại i

- ▶ Ví dụ

$$\begin{aligned} \text{Entropy}([9^+, 5^-]) &= -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) \\ &= 0.94 \end{aligned}$$

Độ tăng thông tin IG

Với tập (con) mẫu S và thuộc tính A

$$IG(S, A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Trong đó:

$values(A)$: tập các giá trị của A

S_v là tập con của S bao gồm các mẫu có giá trị của A bằng v

$|S|$ số phần tử của S

► Tính $IG(S, \text{Gió})$

$$\text{values}(\text{Gió}) = \{\text{yếu}, \text{mạnh}\}$$

$$S = [9+, 5-], H(S) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.94$$

$$S_{\text{yếu}} = [6+, 2-], H(S_{\text{yếu}}) = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8} = 0.811$$

$$S_{\text{mạnh}} = [3+, 3-], H(S_{\text{mạnh}}) = -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6} = 1$$

$$\begin{aligned} IG(S, \text{Gió}) &= H(S) - \frac{8}{14} H(S_{\text{yếu}}) - \frac{6}{14} H(S_{\text{mạnh}}) \\ &= 0.94 - \frac{8}{14} 0.811 - \frac{6}{14} 1 \\ &= 0.048 \end{aligned}$$

Các đặc điểm của ID3

- ▶ ID3 là thuật toán tìm kiếm cây quyết định phù hợp với dữ liệu huấn luyện
- ▶ Tìm kiếm theo kiểu tham lam, bắt đầu từ cây rỗng
- ▶ Hàm đánh giá là độ tăng thông tin
- ▶ ID3 có khuynh hướng (bias) lựa chọn cây đơn giản
 - Ít nút
 - Các thuộc tính có độ tăng thông tin lớn nằm gần gốc

Training error và Test error (1/2)

- ▶ **Training error (lỗi huấn luyện)**
 - Là lỗi đo được trên tập **dữ liệu huấn luyện**
 - Thường đo bằng **sự sai khác** giữa giá trị tính toán của mô hình và giá trị thực của dữ liệu huấn luyện
 - Trong quá trình học ta cố gắng làm **giảm tới mức tối thiểu lỗi huấn luyện**
- ▶ **Test error (lỗi kiểm tra)**
 - Là lỗi đo được trên tập **dữ liệu kiểm tra**
 - Là cái ta thực sự quan tâm

Làm sao ta có thể tác động tới hiệu quả của mô hình trên tập dữ liệu kiểm tra khi ta chỉ quan sát được tập dữ liệu huấn luyện?



Overfitting



Chống quá vưả dữ liệu bằng cắt tỉa cây

- ▶ Chia dữ liệu thành hai phần
 - Huấn luyện
 - Kiểm tra
- ▶ Tạo cây đủ lớn trên dữ liệu huấn luyện
- ▶ Tính độ chính xác của cây trên tập kiểm tra
- ▶ Loại bỏ cây con sao cho kết quả trên dữ liệu kiểm tra được cải thiện nhất
- ▶ Lặp lại cho đến khi không còn cải thiện được kết quả nữa

Chống quá vưả dữ liệu bằng cách tia luật (C4.5)

- ▶ Biến đổi cây thành các luật
- ▶ Tia mỗi luật độc lập với các luật khác
 - Bỏ một số phần trong vế trái của luật
- ▶ Sắp xếp các luật sau khi tia theo mức độ chính xác của luật

Sử dụng thuộc tính có giá trị liên tục

- ▶ Tạo ra những thuộc tính **rời rạc** mới
- ▶ Ví dụ, với thuộc tính liên tục A , tạo ra thuộc tính rời rạc A_c như sau
 - $A_c = \text{true}$ nếu $A > c$
 - $A_c = \text{false}$ nếu $A \leq c$
- ▶ Xác định ngưỡng c thế nào?
 - Thường chọn sao cho A_c đem lại độ tăng thông tin lớn nhất
- ▶ Có thể chia thành nhiều khoảng với nhiều ngưỡng

Nhiệt độ	45	56	60	74	80	90
Chơi tennis	không	không	có	có	có	không

Các độ đo khác

- ▶ Độ đo Information Gain (IG) ưu tiên thuộc tính có nhiều giá trị, ví dụ, thuộc tính ngày sẽ có độ tăng thông tin cao nhất

- ▶ Thông tin chia

$$SplitInformation(S, A) = - \sum_{i=1}^c \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- ▶ Tiêu chuẩn đánh giá thuộc tính

$$GainRatio = \frac{InformationGain(S, A)}{SplitInformation(S, A)}$$



-

Phương pháp Phân loại Bayes (1/2)

- ▶ Trong giai đoạn huấn luyện ta có một tập mẫu, mỗi mẫu được cho bởi cặp $\langle x_i, y_i \rangle$, trong đó
 - x_i là vector đặc trưng (thuộc tính)
 - y_i là nhãn phân loại, $y_i \in C$ (C là tập các nhãn)
- ▶ Sau khi huấn luyện xong, bộ phân loại cần dự đoán nhãn y cho mẫu mới $x = \langle x_1, x_2, \dots, x_n \rangle$

$$y = \operatorname{argmax}_{c_j \in C} P(c_j | x_1, x_2, \dots, x_n)$$

- ▶ Sử dụng quy tắc Bayes

$$\begin{aligned} y &= \operatorname{argmax}_{c_j \in C} \frac{P(x_1, x_2, \dots, x_n | c_j) P(c_j)}{P(x_1, x_2, \dots, x_n)} \\ &= \operatorname{argmax}_{c_j \in C} P(x_1, x_2, \dots, x_n | c_j) P(c_j) \end{aligned}$$

Phương pháp phân loại Bayes (2/2)

Tần xuất quan sát thấy nhãn c_j trên tập dữ liệu D :

$$\frac{\text{count}(c_j)}{|D|}$$

$$y = \operatorname{argmax}_{c_j \in C} P(x_1, x_2, \dots, x_n | c_j) P(c_j)$$

Sử dụng giả thiết về tính độc lập (**Đơn giản!!!**)

$$P(x_1, x_2, \dots, x_n | c_j) = P(x_1 | c_j) P(x_2 | c_j) \dots P(x_n | c_j)$$

Số lần xuất hiện x_i cùng với c_j chia cho số lần xuất hiện c_j : $\frac{\text{count}(x_i, c_j)}{\text{count}(c_j)}$

Ví dụ

► Xác định nhãn phân loại cho mẫu sau

< Trời = nắng, Nhiệt độ = trung bình, Độ ẩm = cao, Gió = mạnh >

$$y = \underset{c \in \{có, không\}}{\operatorname{argmax}} P(\text{Trời} = \text{nắng} | c) P(\text{Nhiệt độ} = \text{trung bình} | c) \\ P(\text{Độ ẩm} = \text{cao} | c) P(\text{Gió} = \text{mạnh} | c) P(c)$$



-

Nguyên tắc chung

- ▶ Không xây dựng mô hình
- ▶ Chỉ lưu lại các mẫu huấn luyện
- ▶ Xác định nhãn cho mẫu mới dựa trên những mẫu giống mẫu mới nhất
- ▶ Gọi là học lười (lazy learning)

Thuật toán k láng giềng gần nhất

- ▶ k -nearest neighbors (k -NN)
- ▶ Chọn k mẫu **giống** mẫu cần phân loại nhất, gọi là k hàng xóm
- ▶ Gán nhãn phân loại cho mẫu chỉ sử dụng thông tin của k hàng xóm này
 - Ví dụ lấy theo đa số trong số k hàng xóm
- ▶ **Chọn hàng xóm thế nào?**

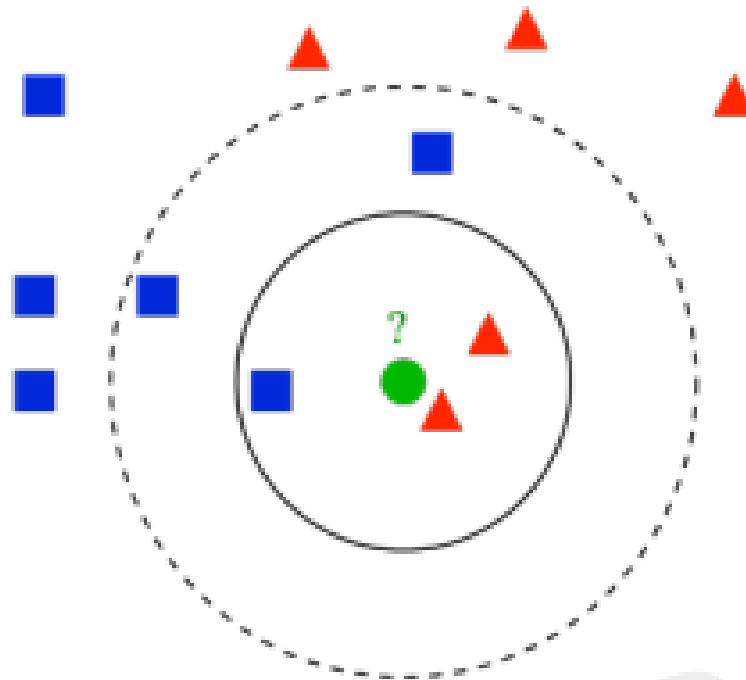
Tính khoảng cách

- ▶ Giả sử mẫu x có giá trị thuộc tính là $< a_1(x), a_2(x), \dots, a_n(x) >$, thuộc tính là số thực
- ▶ Khoảng cách giữa hai mẫu x_i và x_j là khoảng cách Euclidean

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^n (a_l(x_i) - a_l(x_j))^2}$$

- ▶ Phương pháp k-NN thì biểu diễn các mẫu dưới dạng véc tơ trong không gian Euclidean và sử dụng hàm khoảng cách Euclidean để tính độ tương tự.

Minh họa Thuật toán k -NN



Hình 5.11. Kết quả phân loại của k -NN với $k=3$ và $k=5$.

Thuật toán k -NN

Giai đoạn học (huấn luyện)

Lưu các mẫu huấn luyện có dạng $\langle x, f(x) \rangle$ vào cơ sở dữ liệu

Giai đoạn phân loại

Đầu vào: tham số k

Với mẫu x cần phân loại:

1. Tính khoảng cách $d(x, x_i)$ từ x tới tất cả mẫu x_i trong cơ sở dữ liệu
2. Tìm k mẫu có $d(x, x_i)$ nhỏ nhất, giả sử k mẫu đó là x_1, x_2, \dots, x_k .
3. Xác định nhãn phân loại $f'(x)$ là nhãn chiếm đa số trong tập $\{x_1, x_2, \dots, x_k\}$

Thuật toán k -NN

Thuật toán k -NN có thể dùng cho trường hợp Hồi quy, trong đó mẫu x có thể có nhãn phân loại $f(x)$ với $f(x)$ là một số thực. Thuật toán k -NN sẽ thay đổi ở bước 3 như sau:

$$f'(x) = \frac{\sum_{i=1}^k f(x_i)}{k}$$

Yêu cầu

- Thực nghiệm thật toán KNN trên bộ dữ liệu Iris flower dataset tải từ trang UCI. Bộ dữ liệu này bao gồm thông tin của ba loại hoa Iris : Iris setosa, Iris virginica và Iris versicolor. Mỗi loại có 50 bông hoa được đo với dữ liệu là 4 thông tin: chiều dài, chiều rộng đài hoa (sepal), và chiều dài, chiều rộng cánh hoa (petal).
- Tách 150 dữ liệu trong Iris flower dataset ra thành 2 phần, gọi là ***training set*** và ***test set***. Thuật toán KNN sẽ dựa vào thông tin ở *training set* để dự đoán xem mỗi dữ liệu trong *test set* tương ứng với loại hoa nào. Dữ liệu được dự đoán này sẽ được đối chiếu với loại hoa thật của mỗi dữ liệu trong *test set* để đánh giá hiệu quả của KNN.
- Đánh giá:

$$\text{accuracy} = \frac{\text{Số mẫu trong test data được phân loại đúng}}{\text{Số mẫu trong test data}}$$

Cho dữ liệu huấn luyện như ở trong bảng, trong đó A1, A2, A3 là các thuộc tính, f là nhãn phân loại.

- a) Hãy tìm nút gốc cho cây quyết định sử dụng thuật toán ID3. Trong trường hợp có nhiều thuộc tính tốt tương đương thì chọn theo thứ tự lần lượt A1, A2, A3.
- b) Sử dụng thuật toán K láng giềng gần nhất với $k = 5$, tìm nhãn phân loại cho mẫu sau:
- **A1 = 1, A2 = 0, A3 = 1**

A1	A2	A3	f
0	0	1	+
0	0	2	+
0	0	3	+
0	0	4	+
0	1	1	-
0	1	2	-
0	1	3	-
1	0	4	-
1	1	1	+
1	1	2	+