Nhập môn Học máy và Khai phá dữ liệu

(IT3190)

Nguyễn Nhật Quang

quang.nguyennhat@hust.edu.vn

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ thông tin và truyền thông

Năm học 2019-2020

Nội dung môn học:

- **Giới thiệu về**
  - **Học máy**
  - **Khai phá dữ liệu**
  - **Các framework và công cụ phần mềm**
- Tiền xử lý dữ liệu
- Đánh giá hiệu năng của hệ thống
- Hồi quy
- Phân cụm
- Phân loại
- Phát hiện luật kết hợp

Học máy vs. Khai phá dữ liệu

- **Học máy (Machine learning)** vs. **Khai phá dữ liệu (Data mining)**
- Giống nhau:
  - Cần sử dụng dữ liệu; thường là (rất) nhiều dữ liệu
  - Phát hiện tri thức từ dữ liệu (knowledge discovery from data)
- Khác nhau:

	Học máy	**Khai phá dữ liệu**
Trọng tâm:	Tập trung vào việc học (learning) của hệ thống máy tính	Tập trung vào việc hiểu (understanding) dữ liệu
Mục đích sử dụng:	Nhằm dự đoán các kết quả trong tương lai	Nhằm phân tích các dữ liệu hiện có (quá khứ)

Giới thiệu về Học máy

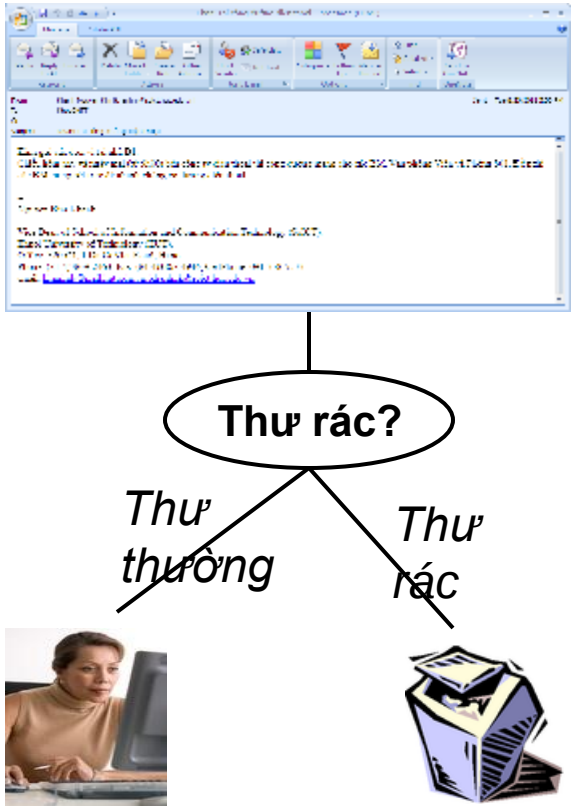
- **Học máy (Machine Learning – ML)** là một lĩnh vực nghiên cứu của Trí tuệ nhân tạo (Artificial Intelligence – AI)
- Các định nghĩa về học máy
  - → Một quá trình nhờ đó một hệ thống cải thiện hiệu suất (hiệu quả hoạt động) của nó `[Simon, 1983]`
  - → Một quá trình mà một chương trình máy tính cải 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, 2020]`
- 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 năng ***P***
  - Thông qua (sử dụng) kinh nghiệm ***E***

Ví dụ bài toán học máy (1)

Lọc thư rác (Email spam filtering)

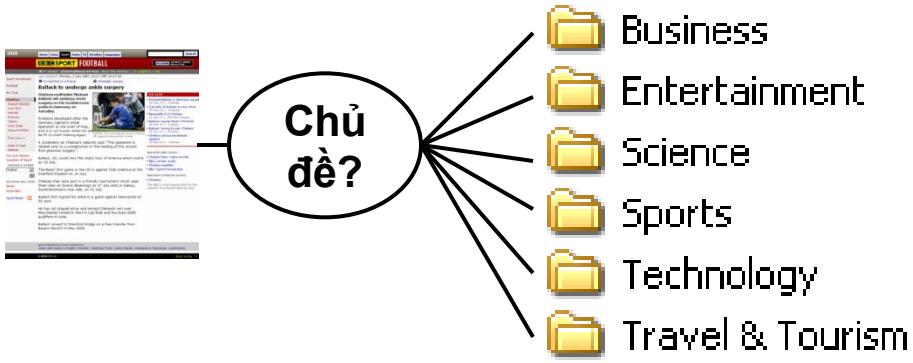
- ***T***: Dự đoán (để lọc) những thư điện tử nào là thư rác (spam email)
- ***P***: % of các thư điện tử gửi đến được phân loại chính xác
- ***E***: Một tập các thư điện tử (emails) mẫu, mỗi thư điện tử được biểu diễn bằng một tập thuộc tính (vd: tập từ khóa) và nhãn lớp (thư thường/thư rác) tương ứng



Ví dụ bài toán học máy (2)

Phân loại các trang Web (Web page categorization/ classification)

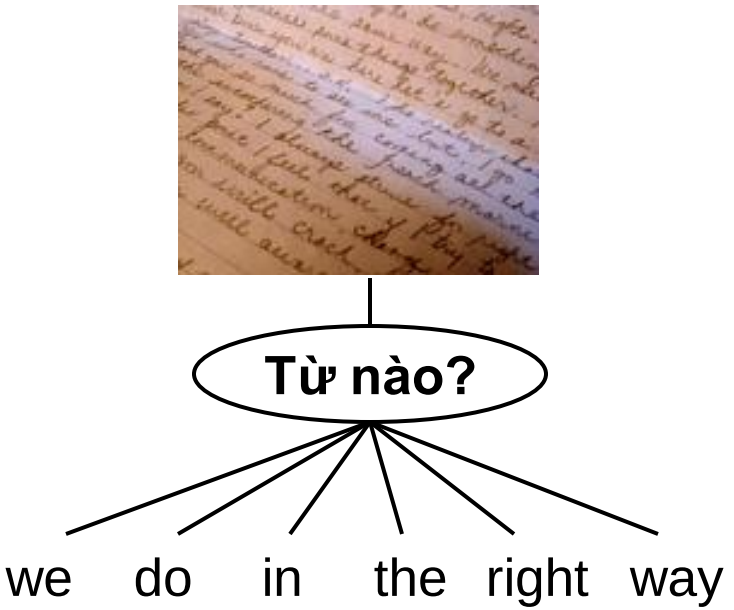
- ***T***: Phân loại các trang Web theo các chủ đề đã định trước
- ***P***: Tỷ lệ (%) các trang Web được phân loại chính xác
- ***E***: Một tập các trang Web, trong đó mỗi trang Web gắn với một chủ đề



Ví dụ bài toán học máy (3)

Nhận dạng chữ viết tay (Handwritten characters recognition)

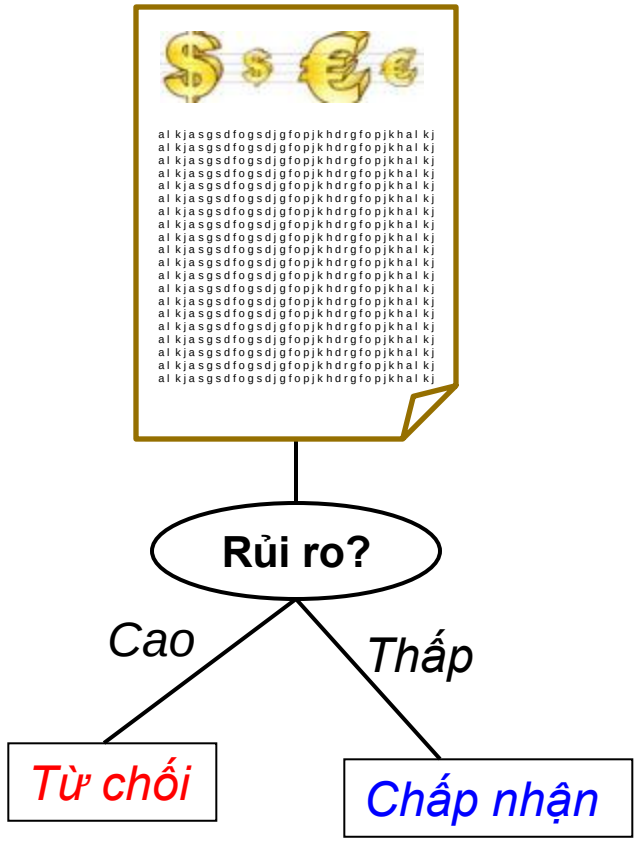
- ***T***: Nhận dạng và phân loại các từ trong các ảnh chữ viết tay
- ***P***: Tỷ lệ (%) các từ được nhận dạng và phân loại đúng
- ***E***: Một tập các ảnh chữ viết tay, trong đó mỗi ảnh được gắn với một định danh của một từ



Ví dụ bài toán học máy (4)

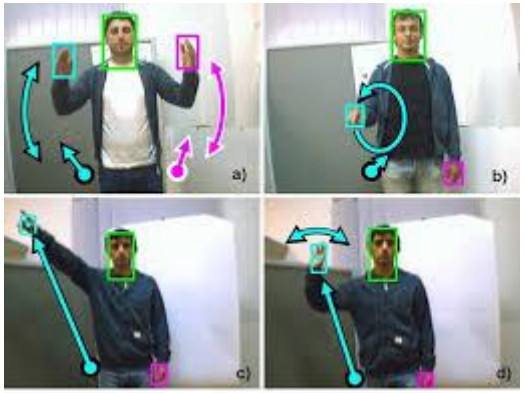
Dự đoán rủi ro cho vay tài chính (Loan risk estimation)

- ***T***: Xác định mức độ rủi ro (vd: 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)



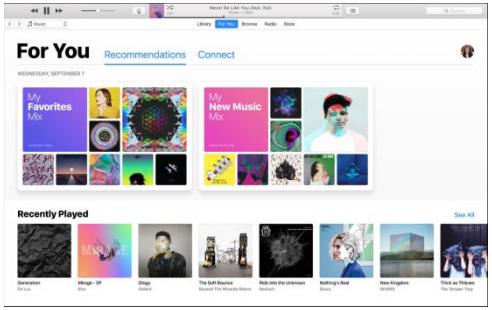
Các ứng dụng thành công của học máy (1)

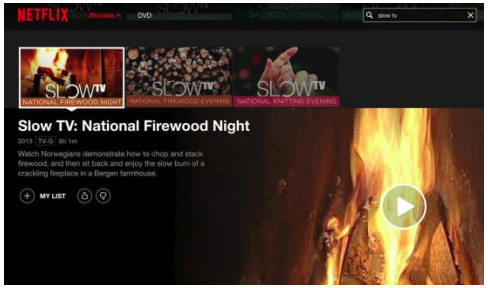
- Tương tác người máy
  - Giọng nói, Cử chỉ, Hiểu ngôn ngữ, …



Các ứng dụng thành công của học máy (2)

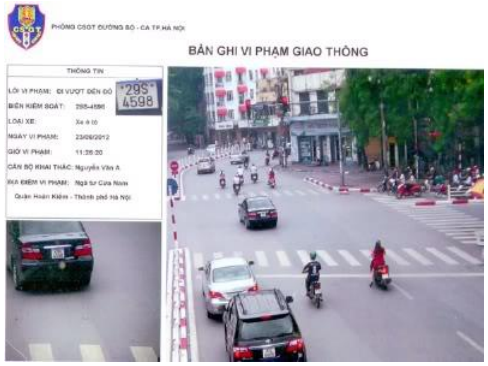
- Giải trí
  - Âm nhạc, Phim ảnh, Trò chơi, Tin tức, Mạng xã hội, …

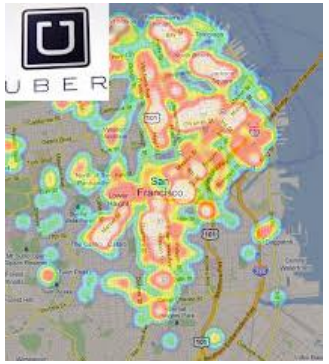




Các ứng dụng thành công của học máy (3)

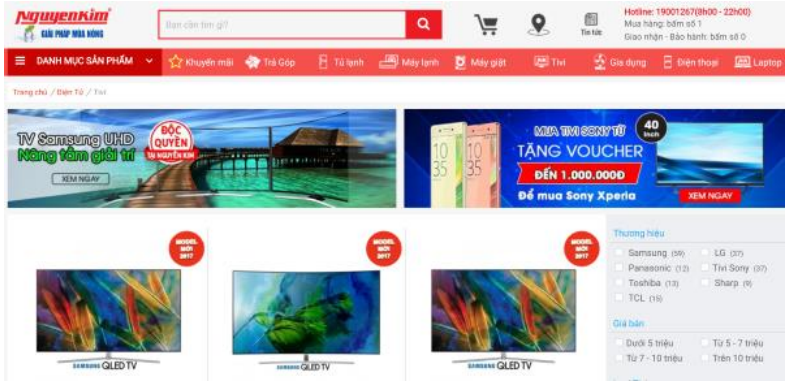
- Giao thông
  - Xe tự động, Giám sát giao thông, Dự đoán nhu cầu đi xe, …





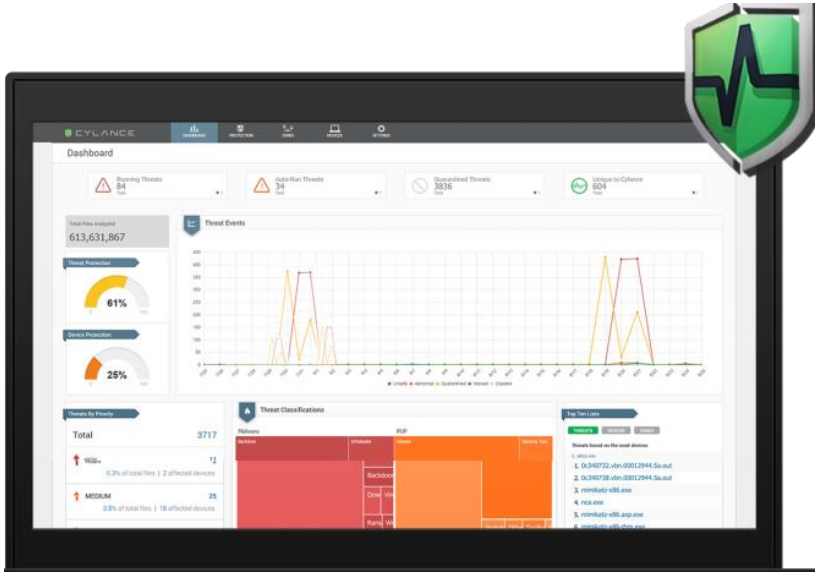
Các ứng dụng thành công của học máy (4)

- Thương mại điện tử
  - Gợi ý các sản phẩm và dịch vụ, Dự đoán nhu cầu khách hàng, Khuyến mại, …



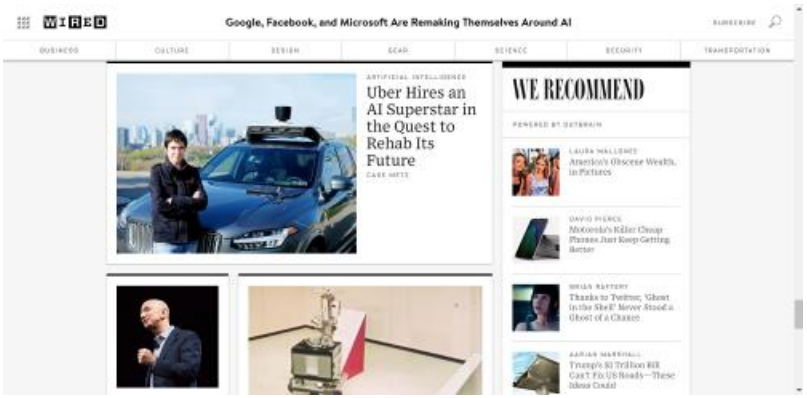
Các ứng dụng thành công của học máy (5)

- An ninh an toàn hệ thống
  - Phát hiện vi rút máy tính, Phát hiện xâm nhập (tấn công) mạng, Lọc thư rác,…

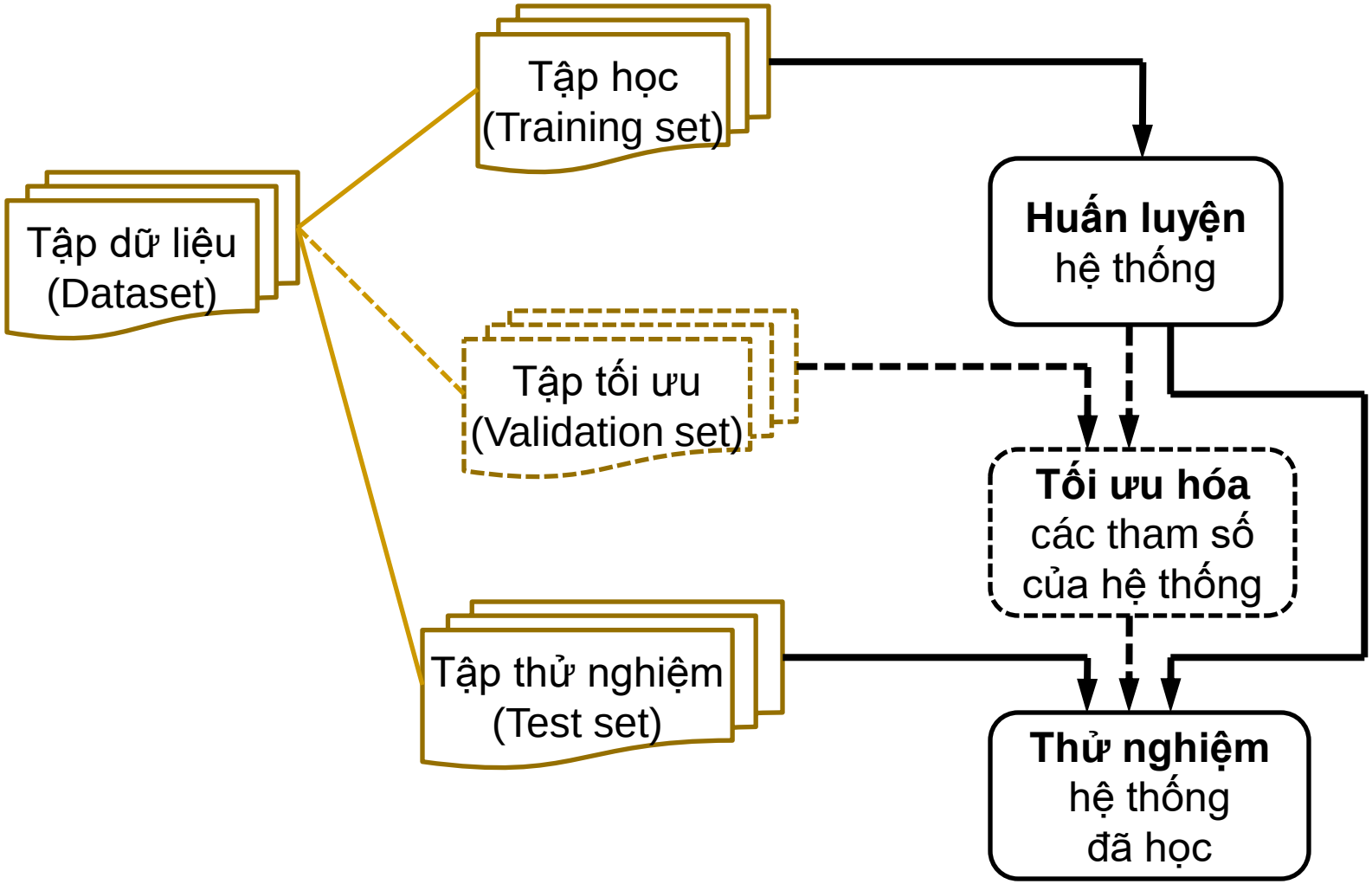


Các ứng dụng thành công của học máy (6)

- Quảng cáo



Quá trình học máy



Các thành phần chính của bài toán học máy (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: $X \rightarrow \{0,1\}$
  - F: $X \rightarrow$ Một tập các nhãn lớp
  - F: $X \rightarrow R^+$ (miền các giá tri số thực dương)
  - …

Các thành phần chính của bài toán học máy (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 thực tế (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 đối với khả năng học của các giải thuật học máy?
  - Khả năng khái quát hóa (generalization) của hệ thống từ các ví dụ học?
    - Vấn đề "học chưa khớp" (under-fitting)
    - Vấn đề "học quá khớp" (over-fitting)
  - 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

Các vấn đề trong Học máy (5)

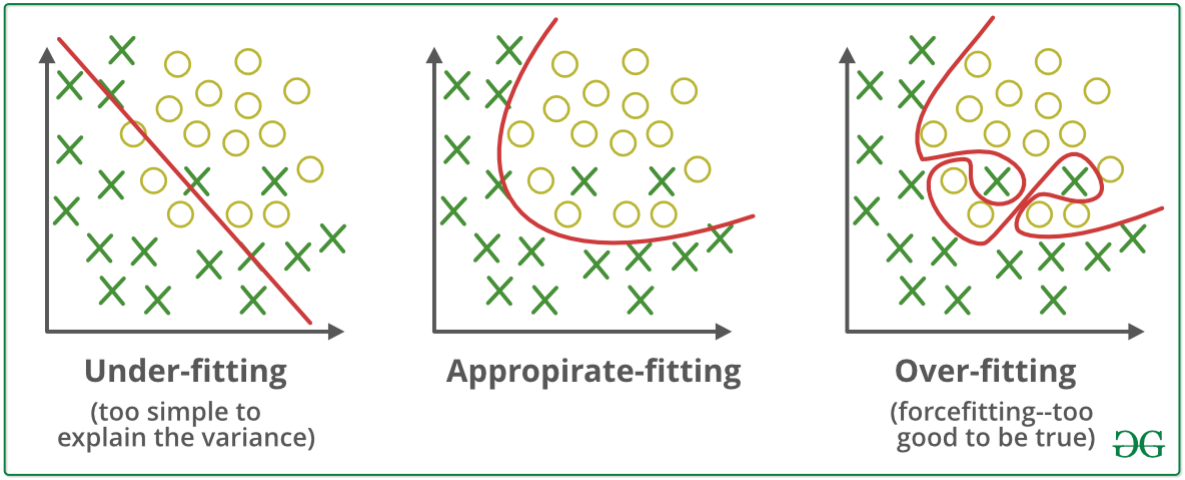
- (WHEN) Khi nào một mô hình (đã được huấn luyện trước đó) cần được huấn luyện lại?
  - Mô hình đã huấn luyện hoạt động (rất) tốt đối với các ví dụ cũ
  - Nhưng đến một thời điểm nhất định, mô hình đó hoạt động kém đi (rất) đáng kể đối với các ví dụ mới đưa vào hệ thống
- (HOW) Một mô hình (đã được huấn luyện trước đó) nên được huấn luyện lại như thế nào?
  - Để phù hợp (thích nghi) với các ví dụ mới đưa vào hệ thống

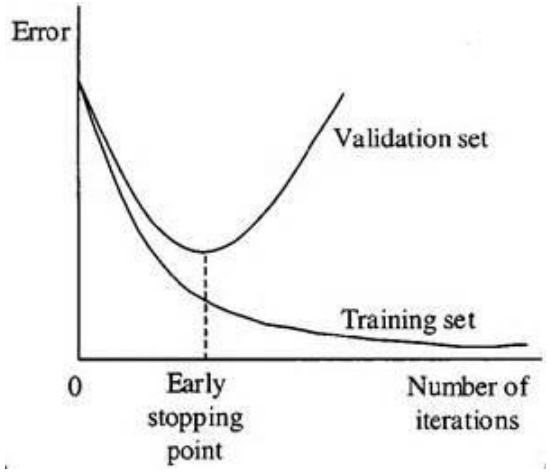
Khả năng khái quát hóa (1)

- Khả năng khái quát hóa (generalization) thể hiện khả năng của mô hình vẫn đạt độ chính xác cao đối với các dữ liệu trong tương lai (unseen data)
  - Lưu ý: Chúng ta không được dùng bất kỳ ví dụ nào trong tập thử nghiệm (test set) trong quá trình huấn luyện/lựa chọn mô hình!
  - Sử dụng tập tối ưu (validation set) – thường được tách ra từ (là 1 phần nhỏ của) tập huấn luyện (training set) ban đầu – để đóng vai trò như là các dữ liệu trong tương lai (unseen data) trong quá trình huấn luyện/lựa chọn mô hình
    - Giả sử: Các đặc điểm dữ liệu giống nhau giữa Tập tối ưu (validation set) và Tập thử nghiệm (test set)!

Khả năng khái quát hóa (2)

- 2 vấn đề thường gặp (và cần tránh!) đối với khả năng khái quát hóa:
  - **Học chưa khớp (under-fitting)**: Đạt độ chính xác thấp trên cả tập học, tập tối ưu và tập thử nghiệm
    - Thường xuyên đưa ra các kết luận sai (Tính chất "*high bias*")
  - **Học quá khớp (over-fitting)**: Đạt độ chính xác cao trên tập học, nhưng đạt độ chính xác thấp trên tập tối ưu và tập thử nghiệm
    - Xu hướng đưa ra các kết luận khác nhau đối với các ví dụ (khá) giống nhau (Tính chất "*high variance*")





(https://towardsdatascience.com/underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6fe4a8a49dbf)

Vấn đề học quá khớp (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 học quá khớp (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 học quá khớp (3)

- 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
  - Mức độ chính xác quá lý tưởng (~100%) đối với tập huấn luyện – Quá trình học hội tụ ở một hàm mục tiêu lý tưởng đối với tập huấn luyện (nhưng không thích hợp với các ví dụ khác trong tương lai)

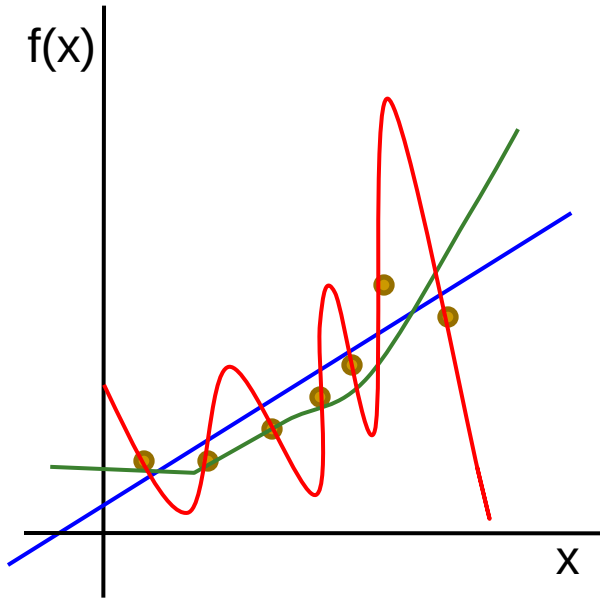
Vấn đề học quá khớp (4)

- 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

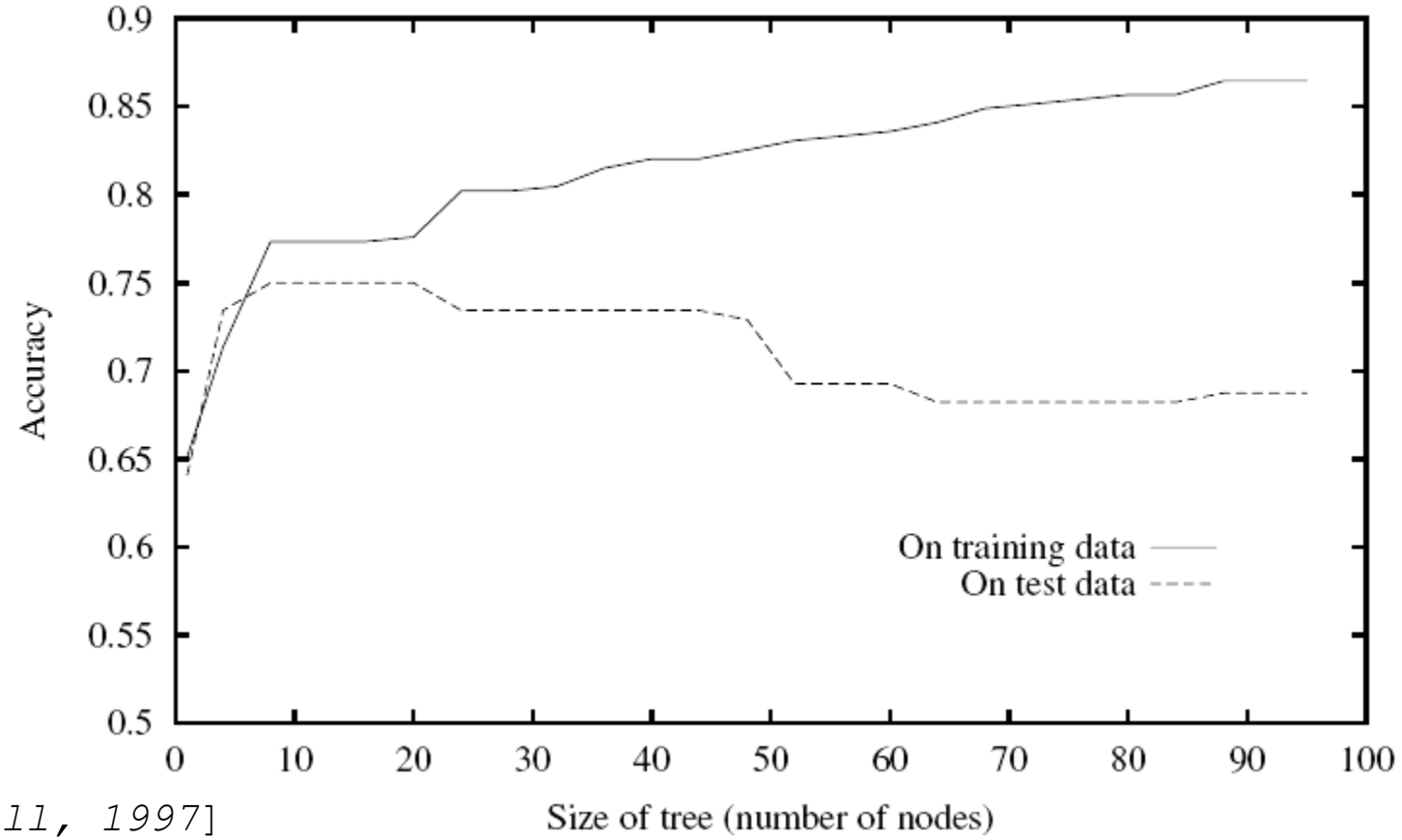
- **Occam's razor**: Ưu tiên chọn hàm mục tiêu đơn giản nhất phù hợp (không nhất thiết hoàn hảo) với các ví dụ học
  - → Khái quát hóa tốt hơn
  - → Dễ giải thích/diễn giải hơn
  - → Độ phức tạp tính toán ít hơn

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



Vấn đề học quá khớp – 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*]

Giới thiệu về Khai phá dữ liệu

- **Khai phá dữ liệu (Data mining – DM)** – Phát hiện tri thức từ dữ liệu (Knowledge discovery from data)
  - Là việc trích rút ra được các mẫu hoặc tri thức *quan trọng* từ một lượng dữ liệu (rất) lớn
    - *quan trọng* = không tầm thường, ẩn, chưa được biết đến, và có thể hữu ích
- Các tên gọi khác
  - Phát hiện tri thức trong các cơ sở dữ liệu (Knowledge discovery in databases - KDD)
  - Trích rút tri thức (Knowledge extraction)
  - Phân tích mẫu/dữ liệu (Data/pattern analysis)
  - …

Tại sao cần khai phá dữ liệu?

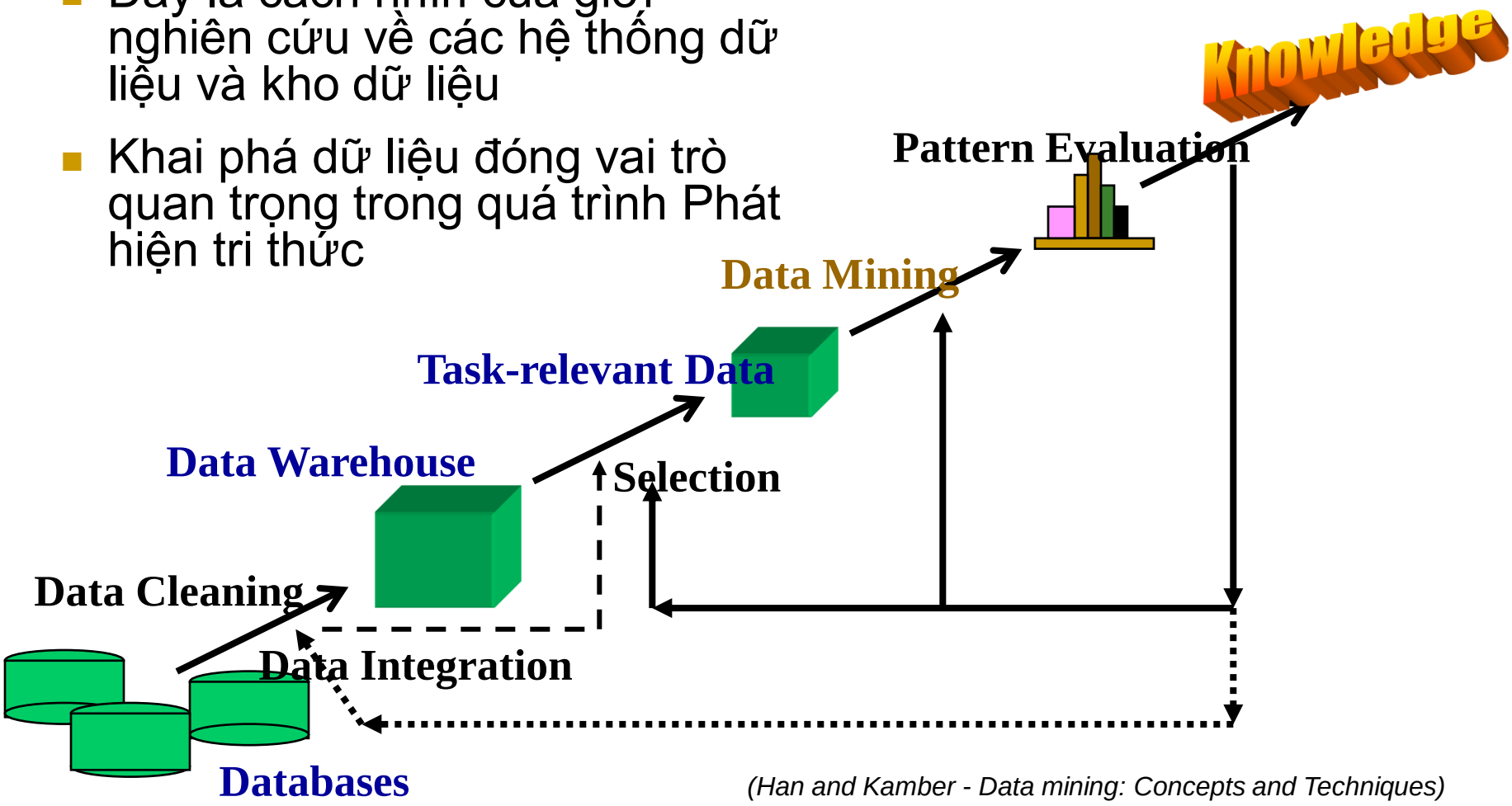
- Sự gia tăng bùng nổ của dữ liệu: Từ mức độ terabytes đến mức độ petabytes
  - Thu thập dữ liệu và sự tồn tại của dữ liệu
    - Các công cụ thu thập dữ liệu tự động, các hệ thống cơ sở dữ liệu, World Wide Web, xã hội số
  - Các nguồn dữ liệu phong phú
    - Kinh doanh: Internet, thương mại điện tử, giao dịch thương mại, chứng khoán,…
    - Khoa học: Tín hiệu cảm biến, tin sinh, thí nghiệm mô phỏng/giả lập,…
    - Xã hội: Tin tức, máy ảnh số, các mạng xã hội
- Chúng ta bị tràn ngập trong dữ liệu – Nhưng lại thiếu (cần) tri thức
- Khai phá dữ liệu: Giúp *tự động* phân tích các tập dữ liệu rất lớn, để phát hiện ra các tri thức

Các bước của quá trình Phát hiện tri thức

1. Tìm hiểu lĩnh vực của bài toán (ứng dụng)
   - Các mục đích của bài toán, các tri thức cụ thể của lĩnh vực
2. Tạo nên (thu thập) một tập dữ liệu phù hợp
3. Làm sạch và tiền xử lý dữ liệu
4. Giảm kích thước của dữ liệu, chuyển đổi dữ liệu
   - Xác định các thuộc tính quan trọng, giảm số chiều (số thuộc tính), biểu diễn bất biến
5. Lựa chọn chức năng khai phá dữ liệu
   - Tóm tắt hóa (summarization), phân loại/phân lớp, hồi quy/dự đoán, kết hợp, phân cụm
6. Lựa chọn/Phát triển (các) giải thuật khai phá dữ liệu phù hợp
7. Tiến hành quá trình khai phá dữ liệu
8. Đánh giá mẫu thu được và biểu diễn tri thức
   - Hiển thị hóa, chuyển đổi, bỏ đi các mẫu dư thừa, …
9. Sử dụng các tri thức được phát hiện

Quá trình Phát hiện tri thức (1)

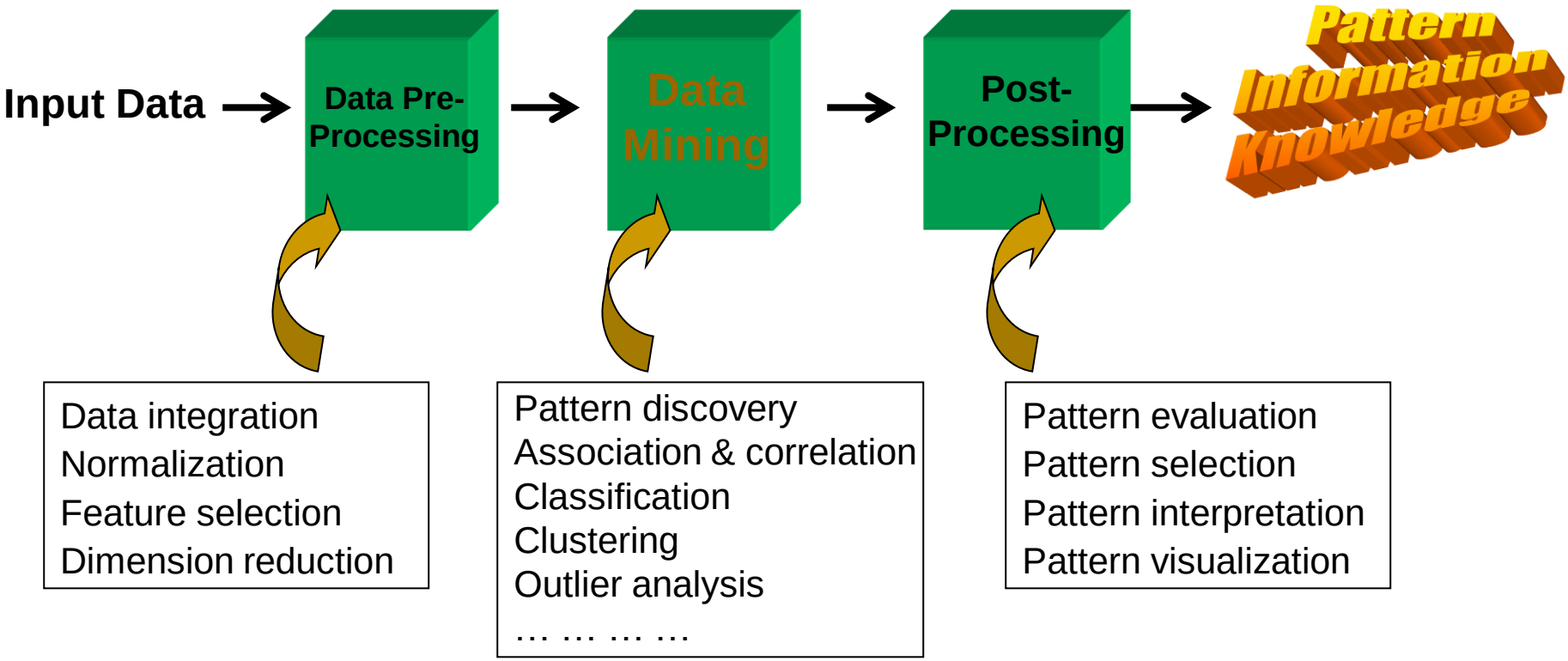
- Đây là cách nhìn của giới nghiên cứu về các hệ thống dữ liệu và kho dữ liệu
- Khai phá dữ liệu đóng vai trò quan trọng trong quá trình Phát hiện tri thức



(Han and Kamber - Data mining: Concepts and Techniques)

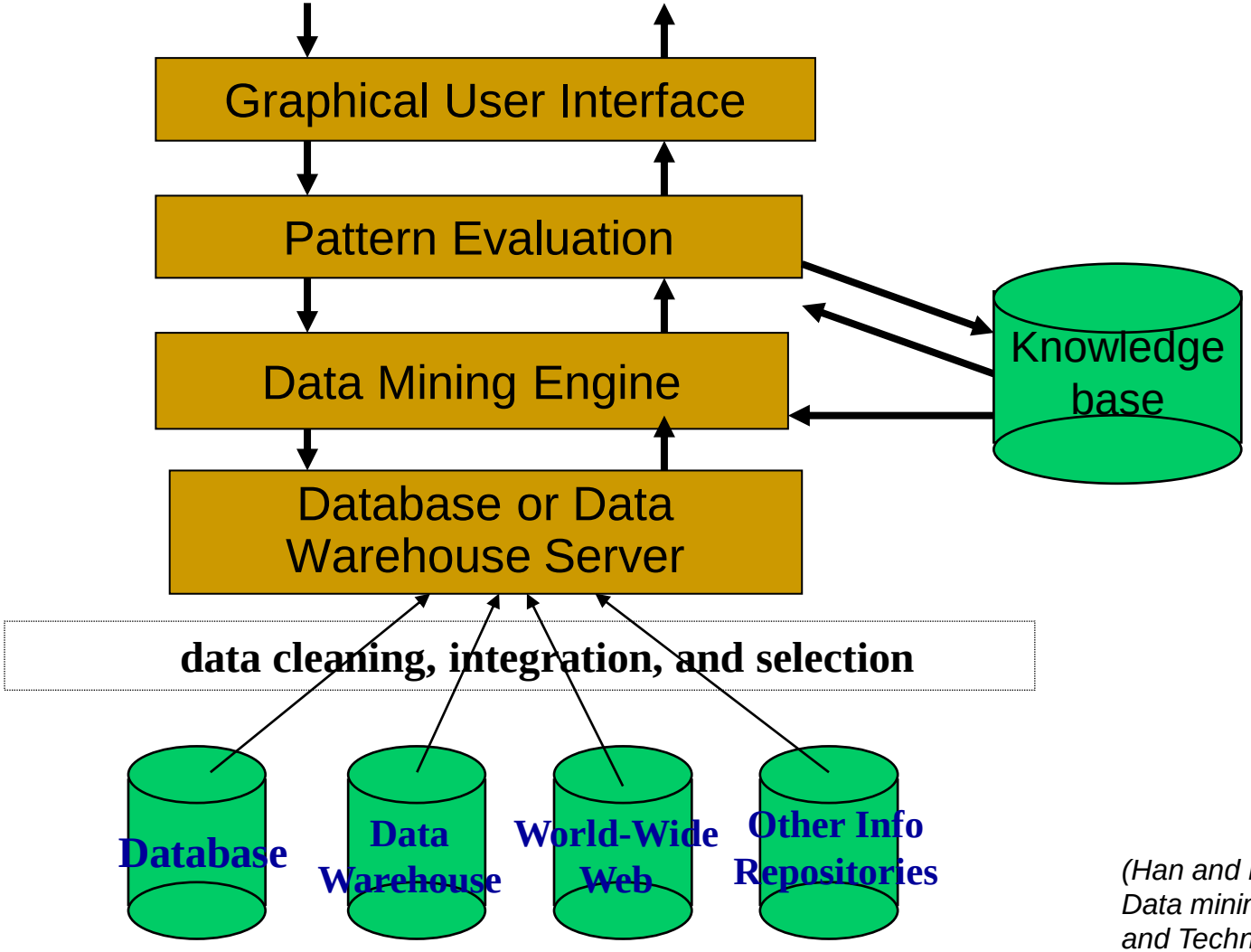
Quá trình Phát hiện tri thức (2)

(Han and Kamber - Data mining: Concepts and Techniques)



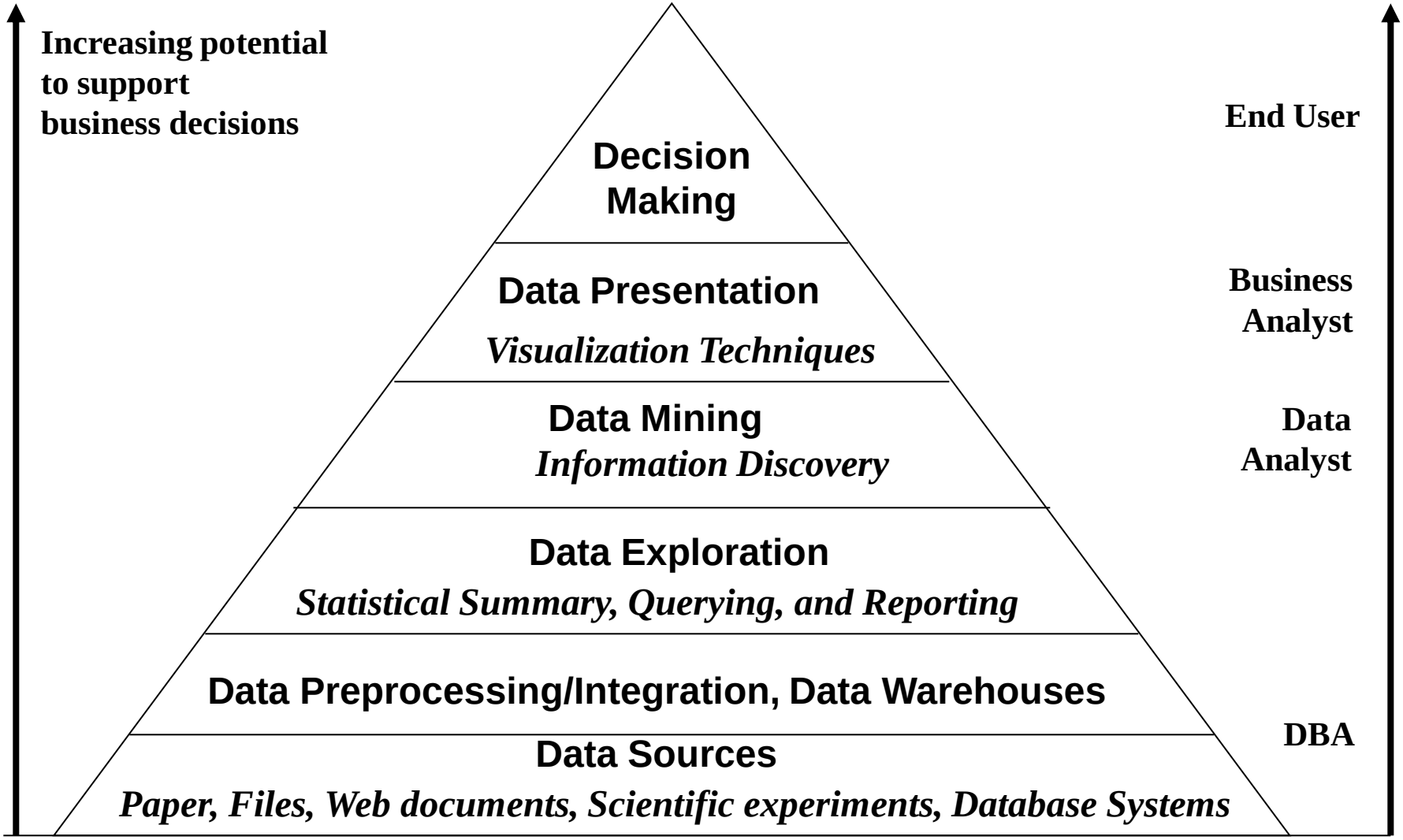
- Đây là cách nhìn của giới nghiên cứu về học máy và thống kê

Kiến trúc hệ thống khai phá dữ liệu



(Han and Kamber - Data mining: Concepts and Techniques)

Khai phá dữ liệu cho kinh doanh



(Han and Kamber - Data mining: Concepts and Techniques)

DM – Các lĩnh vực liên quan

- Công nghệ cơ sở dữ liệu (Database technology)
- Giải thuật (Algorithm)
- Thống kê (Statistics)
- Học máy (Machine learning)
- Nhận dạng mẫu (Pattern recognition)
- Hiển thị hóa (Visualization)
- Tính toán hiệu năng cao (High-performance computing)

Khai phá dữ liệu – Các cách nhìn

- Dữ liệu được khai phá
  - Dữ liệu quan hệ, kho dữ liệu, dữ liệu giao dịch, luồng dữ liệu, dữ liệu hướng đối tượng, dữ liệu phụ thuộc không gian, dữ liệu liên tục theo thời gian, dữ liệu dạng văn bản, dữ liệu đa phương tiện, dữ liệu hỗn tạp, dữ liệu trên WWW, …
- Tri thức được phát hiện
  - Sự đặc trưng, sự phân biệt, luật kết hợp, phân lớp, phân cụm, xu hướng/dịch chuyển, phân tích ngoại lai (outlier)
- Các kỹ thuật được sử dụng
  - Dựa trên cơ sở dữ liệu, phân tích kho dữ liệu, học máy, thống kê, hiển thị hóa, …
- Các ứng dụng (bài toán) thực tế
  - Kinh doanh bán lẻ, viễn thông, ngân hàng, phát hiện gian lận tài chính, khai phá dữ liệu sinh học, phân tích thị trường chứng khoán, khai phá văn bản, khai phá Web, …

DM: Khái quát hóa

- Tích hợp thông tin và xây dựng các kho dữ liệu
  - Làm sạch dữ liệu, chuyển đổi dữ liệu, tích hợp dữ liệu, và mô hình dữ liệu nhiều chiều (multi-dimensional data model)
- Công nghệ khối dữ liệu (data cube)
  - Các phương pháp hiệu quả để tính toán kết hợp nhiều chiều của dữ liệu
  - Xử lý phân tích trực tuyến (Online analytical processing – OLAP)
- Mô tả khái niệm theo nhiều chiều: Sự đặc trưng và sự phân biệt
  - Tổng quát hóa, tóm tắt, và tương phản các đặc tính của dữ liệu

DM: Phân tích kết hợp và tương quan

- Các mẫu hoặc các tập mục (itemsets) thường xuyên
  - Những mục (sản phẩm) nào thường xuyên được mua cùng nhau, trong siêu thị BigC?
- Kết hợp (association), tương quan (correlation), và nguyên nhân (causality)
  - Ví dụ về một luật kết hợp (association rule)
    - Bánh mỳ → Sữa [0.5%, 75%] (độ hỗ trợ – support, độ tin cậy – confidence)
  - Các mục kết hợp ở mức cao, thì cũng tương quan ở mức cao?
- Làm thế nào để phát hiện các mẫu (luật) như vậy trong các tập dữ liệu lớn?

DM: Phân lớp và dự đoán

- Phân lớp (classification) và dự đoán (prediction)
  - Xây dựng các mô hình (các hàm mục tiêu) dựa trên một số ví dụ học/huấn luyện
  - Mô tả và phân biệt các lớp (các khái niệm) cho việc dự đoán trong tương lai
  - Phân lớp các ví dụ mới, hoặc dự đoán các giá trị kiểu số
- Các phương pháp điển hình
  - Cây quyết định (Decision tree learning), Phân lớp Naïve Bayes (Naïve Bayes classification), Máy vectơ hỗ trợ (Support vector machine), Mạng nơ-ron nhân tạo (Artificial neural networks), Học quy nạp luật (Rule induction), Hồi quy tuyến tính (Linear regression), …
- Các ứng dụng điển hình
  - Phát hiện gian lận thẻ tín dụng, quảng cáo trực tiếp (phù hợp với từng người), phân loại/dự đoán các loại bệnh, phân loại các trang Web, …

DM: Phân cụm và phân tích ngoại lai

- Phân cụm (Cluster analysis)
  - Phương pháp học không giám sát (unsupervised learning) – không có thông tin về nhãn lớp
  - Nhóm dữ liệu lại thành các cụm (clusters)
  - Nguyên tắc: Cực đại hóa sự tương tự giữa các đối tượng trong cùng một cụm; nhưng cực tiểu hóa sự tương tự giữa các đối tượng khác cụm
  - Có rất nhiều phương pháp và ứng dụng (bài toán)
- Phân tích ngoại lai (Outlier analysis/detection)
  - Ngoại lai (Outlier): Một đối tượng rất khác biệt với các đối tượng khác (trong một cụm)
  - Nhiễu của dữ liệu, hay là ngoại lệ?
  - Các phương pháp: phân cụm, phân tích hồi quy, …
  - Rất hữu ích trong các bài toán phát hiện gian lận (giả mạo), hoặc phân tích các sự kiện hiếm khi xảy ra

DM: Phân tích xu hướng và tiến triển

- Phân tích chuỗi (sequence), xu hướng (trend), và tiến triển (evolution)
  - Phân tích xu hướng và sự dịch chuyển (khỏi xu hướng)
  - Khai phá các mẫu kiểu chuỗi (sequential patterns)
    - Vd: Đầu tiên mua máy ảnh số, sau đó mua các thẻ nhớ SD dung lượng lớn, …
  - Phân tích tính chu kỳ (Periodicity analysis)
  - Phân tích chuỗi dữ liệu liên tục theo thời gian (time-series) và chuỗi dữ liệu sinh học
  - Phân tích dựa trên sự tương tự (Similarity-based analysis)

DM: Phân tích mạng và cấu trúc

- Khai phá đồ thị dữ liệu (Graph mining)
  - Tìm ra các đồ thị con (các phần của đồ thị ban đầu), các cây (dữ liệu XML), các cấu trúc con (dữ liệu Web) … thường xuyên xảy ra
- Phân tích mạng thông tin (Information network analysis)
  - Các mạng xã hội: các tác nhân (các đối tượng, các nút) và các mối quan hệ (các cạnh)
    - Vd: Mạng các tác giả (học giả) trong lĩnh vực Trí tuệ nhân tạo
  - Các mạng hỗn tạp (khác nhau)
    - Vd: Một người có thể tham gia nhiều mạng khác nhau (bạn bè, gia đình, bạn cùng lớp/trường, những người cùng sở thích nghe nhạc Rock,…)
  - Các liên kết (links) mang rất nhiều thông tin ngữ nghĩa: Khai phá các liên kết (Link mining)
- Khai phá Web (Web mining)
  - WWW là một mạng thông tin khổng lồ: PageRank (Google)
  - Phân tích các mạng thông tin Web
    - Phát hiện cộng đồng Web, Khai phá ý kiến (Opinion mining), Khai phá dữ liệu truy cập Web (usage mining)

Tất cả các mẫu đều quan trọng?

- Quá trình khai phá dữ liệu có thể sinh (phát hiện) ra hàng ngàn mẫu – Không phải tất cả các mẫu đều quan trọng
- Các đánh giá về mức độ quan trọng của các mẫu
  - Một mẫu là quan trọng, nếu nó: dễ hiểu đối với người dùng, vẫn đúng đối với các dữ liệu mới (ở một mức độ chắc chắn nhất định), hữu dụng, mới mẻ, hoặc giúp xác nhận một giả thiết nào đó của một người dùng
- Các đánh giá dựa trên mục tiêu (objective) và dựa trên chủ quan (subjective)
  - Dựa trên mục tiêu (objective): dựa trên sự thống kê và các cấu trúc của các mẫu
    - Vd: dựa trên các giá trị độ hỗ trợ (support), độ tin cậy (confidence)
  - Dựa trên chủ quan (subjective): dựa trên sự tin cậy của người dùng đối với dữ liệu
    - Vd: sự ngạc nhiên, sự mới mẻ, … đối với người dùng

Đánh giá mức độ quan trọng của mẫu

- Mức độ đơn giản (Simplicity)
  - Ví dụ: Độ dài của các luật kết hợp
  - Ví dụ: Kích thước của cây quyết định học được
- Mức độ tin cậy (Certainty/Confidence)
  - Ví dụ: Độ tin cậy (confidence) của các luật kết hợp
  - Ví dụ: Độ chính xác của phân lớp học được
- Mức độ tiện ích (Utility): Khả năng hữu ích của mẫu
  - Ví dụ: Độ hỗ trợ của các luật kết hợp
  - Ví dụ: Ngưỡng nhiễu đối với phân lớp học được
- Tính mới mẻ (Novelty): Mẫu mới, chưa bao giờ được biết đến

Tìm tất cả các mẫu quan trọng?

- Tìm tất cả các mẫu quan trọng: Tính hoàn chỉnh (completeness)
  - Một hệ thống khai phá dữ liệu có thể tìm được *tất cả* các mẫu quan trọng không?
  - Chúng ta có cần phải tìm *tất cả* các mẫu quan trọng không?
  - Tìm kiếm vét cạn (exhaustive) vs. heuristic
- Chỉ tìm các mẫu quan trọng: Bài toán tối ưu
  - Một hệ thống khai phá dữ liệu có thể tìm *chỉ* các mẫu quan trọng?
  - Các phương pháp
    - Trước hết cứ sinh (tìm) ra tất cả các mẫu, sau đó loại bỏ đi các mẫu không quan trọng
    - (Trong quá trình khai phá dữ liệu) Chỉ sinh ra các mẫu quan trọng

Hiển thị các mẫu tìm được

- Các người dùng khác nhau, các mục đích sử dụng khác nhau sẽ yêu cầu các dạng hiển thị khác nhau đối với các mẫu tìm được
  - Hiển thị bằng: các luật, các bảng, biểu đồ so sánh, …
- Phân cấp khái niệm
  - Tri thức phát hiện được có thể sẽ dễ hiểu hơn khi được biểu diễn ở mức khái quát hóa cao hơn
  - Sự phân cấp khái niệm cho phép nhìn (xét) dữ liệu theo các cách nhìn khác nhau
- Các kiểu tri thức khác nhau đòi hỏi các cách biểu diễn khác nhau (đối với các mẫu tìm được)
  - Luật kết hợp
  - Phân lớp,
  - Phân cụm
  - …

DM: Các ứng dụng thành công

- Phân tích dữ liệu cho hỗ trợ ra quyết định
  - Phân tích thị trường
    - Quảng cáo cá nhân (target marketing), quản lý quan hệ khách hàng (CRM), phân tích giỏ hàng, bán hàng liên quan (cross-selling), phân chia thị trường
  - Phân tích rủi ro
    - Dự đoán, giữ khách hàng, phân tích cạnh tranh
  - Phát hiện gian lận (outliers)
- Các ứng dụng khác
  - Khai phá văn bản (nhóm tin – news group, email, tài liệu)
  - Khai phá Web
  - Phân tích dữ liệu sinh học và tin sinh
  - …*(Và rất nhiều các ứng dụng thực tế khác!)*

DM: Các vấn đề thách thức

- Tính hiệu quả (efficiency) và khả năng mở rộng (scalability) của các giải thuật khai phá dữ liệu
- Các phương pháp khai phá dữ liệu *song song*, *phân tán*, *luồng (stream)*, và *tăng cường (incremental)*
- Xử lý với dữ liệu có số chiều (số thuộc tính) lớn
- Xử lý với dữ liệu chứa nhiễu (lỗi), không chắc chắn, không hoàn chỉnh
- Đưa (tích hợp) vào quá trình khai phá dữ liệu các ràng buộc, tri thức chuyên gia, tri thức nền tảng (background knowledge)
- Đánh giá mẫu và tích hợp tri thức
- Khai phá các kiểu dữ liệu rất khác nhau (dữ liệu tin sinh, Web, mạng thông tin,…)
- Tích hợp khai phá dữ liệu vào các thiết bị hoạt động
- Bảo đảm tính an ninh, toàn vẹn, riêng tư trong khai phá dữ liệu

Các frameworks và công cụ phần mềm (1)

- **TensorFlow** (www.tensorflow.org)
  - OS: Linux, Mac OS, Windows, Android
  - Programming language: Python, C++, Java
- **Caffe** (caffe.berkeleyvision.org)
  - OS: Linux, Mac OS, Windows
  - Programming language: Python, Matlab
- **Caffe2** (caffe2.ai), **PyTorch** (pytorch.org)
  - On March, 2018, Caffe2 and PyTorch is merged into a single platform
  - OS: Linux, Mac OS, Windows, iOS, Android, Raspbian
  - Programming language: C++, Python
- **Keras** (keras.io)
  - OS: Linux, Mac OS, Windows
  - Programming language: Python
- **Theano** (deeplearning.net/software/Theano)
  - OS: Linux, Mac OS, Windows
  - Programming language: Python

Các frameworks và công cụ phần mềm (2)

- **CNTK** (www.microsoft.com/en-us/research/product/ cognitive-toolkit/)
  - OS: Windows, Linux
  - Programming language: Python, C++, C#
- **Deeplearning4j** (deeplearning4j.org)
  - OS: Linux, Mac OS, Windows, Android
  - Programming language: Java, Scala, Clojure, Python
- **Apache Mahout** (mahout.apache.org)
  - OS: Any OSs with JVM installed
  - Programming language: Java, Scala
- **MLlib** of Apache Spark (https://spark.apache.org/mllib/)
  - OS: Any OSs with JVM installed
  - Programming language: Java, Python, Scala, R
- **Weka** (http://www.cs.waikato.ac.nz/ml/weka/)
  - OS: Any OSs with JVM installed
  - Programming language: Java

Một số khóa học tự học

- **Statistics-101** (provided by IBM)

  https://cognitiveclass.ai/courses/statistics-101
- **Machine Learning with Python** (provided by IBM)

  https://cognitiveclass.ai/courses/machine-learning-with-python
- **Machine Learning Foundations: A Case Study Approach** (provided by University of Washington)

  https://www.coursera.org/learn/ml-foundations
- **Machine Learning** (provided by Stanford University)

  https://www.coursera.org/learn/machine-learning
- **Predictive Analytics and Data Mining** (provided by University of Illinois at Urbana-Champaign)

  https://www.coursera.org/learn/predictive-analytics-data-mining
- **Data Mining Specialization** (provided by University of Illinois at Urbana-Champaign)

  https://www.coursera.org/specializations/data-mining

Tài liệu tham khảo

- E. Alpaydin. *Introduction to Machine Learning, 4th Edition*. The MIT Press, 2020.
- 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.