# Giới thiệu Máy học (Machine Learning - ML)

Thiện Lê

UIUC

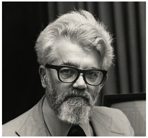
# Sơ lược

# Sơ lược

# Trí tuệ nhân tạo (Artificial Inteligence - AI)

**John McCarthy (1955)**

... making a machine behave in ways that would be called intelligent if a human were so behaving ...



**Marvin Minsky (1968)**

... making machines do things that would require intelligence if done by men ...

# Trí tuệ nhân tạo (Artificial Inteligence - AI)

- *Thuật toán* truyền tải trí thông minh của con người cho máy tính

# Ví dụ truyền tải trí thông minh

**Bài toán**

Viết thuật toán dự đoán email có phải spam hay không (spam filtering)

# Giải đơn giản bài toán spam filtering

## Nhận xét

- Spam email thường chứa những từ ‘quảng cáo’, ‘nhanh tay’, ‘miễn phí’, v.v.
- Spam email thường chứa link độc hại

# Giải đơn giản bài toán spam filtering

## Nhận xét

- Spam email thường chứa những từ ‘quảng cáo’, ‘nhanh tay’, ‘miễn phí’, v.v.
- Spam email thường chứa link độc hại

## Ví dụ thuật toán đơn giản

*nếu* nội dung chứa ‘quảng cáo’ hay ‘nhanh tay’ hay ‘miễn phí’
    *thì* dự đoán ‘spam!’
*còn không*:
    *nếu* nội dung có chứa link độc hại, *thì* dự đoán ‘spam!’
    *còn không*, dự đoán ‘không spam’

# Thuật toán rule-based

- Bắt đầu từ những nhận xét căn bản (rule)
- Xét mọi trường hợp, dùng *câu lệnh* if-else

# Nhận xét thuật toán rule-based

- Dễ viết, chỉ phụ thuộc vào rule
- ...

- Cần rule cụ thể
- Có thể phải xét rất nhiều trường hợp
- ....

**Tóm tắt**

- Dễ làm, nhưng khó để làm tốt!
- Cách nào đỡ "cực" hơn? Thuật toán tự tìm rule?

# Sơ lược

# Liên quan tới bài toán AI

- *Bài toán ML* $\subset$ *bài toán AI*
- Truyền tải trí thông về *dữ liệu* (data)
- *Thuật toán ML* cố gắng giải quyết *bài toán ML*

# Định nghĩa vắn tắt ML

## Định nghĩa vắn tắt I

*Bài toán ML* là bài toán tìm hàm số giải thích dữ liệu (data)

# Định nghĩa vắn tắt ML

**Định nghĩa vắn tắt I**

*Bài toán ML* là bài toán tìm hàm số giải thích dữ liệu (data)

1. Tìm ở đâu?
2. Giải thích dữ liệu như thế nào?

# Định nghĩa vắn tắt ML

**Định nghĩa vắn tắt I**

*Bài toán ML* là bài toán tìm hàm số giải thích dữ liệu (data)

1. Tìm ở đâu?
2. Giải thích dữ liệu như thế nào?

# Tìm ở đâu? – Gia đình hàm (function family)

- *Hàm số (function)*
  - $f$ : miền xác định (domain) $\rightarrow$ miền giá trị (range)
  - $f$ : biến $\mapsto$ giá trị của biến
- *Gia đình hàm*: tập hợp nhiều hàm số có cùng tính chất

# Ví dụ về hàm số

## Ví dụ hàm số, gia đình hàm

- bình phương : $\mathbb{R} \to \mathbb{R} : x \mapsto x^2$

# Ví dụ về hàm số

## Ví dụ hàm số, gia đình hàm

- bình phương : $\mathbb{R} \to \mathbb{R} : x \mapsto x^2$
- phân loại email : $\{\text{email}\} \to \{\text{'spam'}, \text{'không spam'}\}$

# Ví dụ về hàm số

## Ví dụ hàm số, gia đình hàm

- bình phương : $\mathbb{R} \to \mathbb{R} : x \mapsto x^2$
- phân loại email : $\{\text{email}\} \to \{\text{'spam'}, \text{'không spam'}\}$
- gia đình hàm tuyến tính (linear) $\{f : x \mapsto ax + b | a, b \in \mathbb{R}\}$

# Ví dụ ML

## Ví dụ bài toán ML khái quát cho spam filtering

- *Cho:* gia đình hàm
  $\mathcal{H} = \{f : \{\text{email}\} \rightarrow \{\text{‘spam’}, \text{‘không spam’}\}\}$
- *Dữ liệu:* tập hợp nhiều email nhận được trong năm vừa rồi
- *Bài toán ML:* tìm hàm số $f^* \in \mathcal{H}$ giải thích tốt dữ liệu có được

# Ví dụ ML

## Ví dụ bài toán ML khái quát cho spam filtering

- *Cho:* gia đình hàm
  $\mathcal{H} = \{f : \{\text{email}\} \to \{\text{‘spam'}, \text{‘không spam'}\}\}$
- *Dữ liệu:* tập hợp nhiều email nhận được trong năm vừa rồi
- *Bài toán ML:* tìm hàm số $f^* \in \mathcal{H}$ giải thích tốt dữ liệu có được

## Ứng dụng ML cho spam filtering

- Tìm $f^*$ bằng cách giải bài toán ML
- Khi có email $x$ mới, phân loại theo $f^*(x)$

# Định nghĩa vắn tắt ML

**Định nghĩa vắn tắt I**

*Bài toán ML* là bài toán tìm hàm số giải thích dữ liệu (data)

1. Tìm ở đâu?
2. Giải thích dữ liệu như thế nào?

# Định nghĩa vắn tắt ML

**Định nghĩa vắn tắt I**

*Bài toán ML* là bài toán tìm hàm số giải thích dữ liệu (data)

1. Tìm ở đâu?
2. Giải thích dữ liệu như thế nào?

# Giải thích dữ liệu thế nào là tốt (trong bài toán spam filtering)?

- Thông tin di chuyển như thế nào trong bài toán ML?
  - *Thông tin hiện tại*: dữ liệu hiện tại đang có
  - *Thông tin mới*: thông tin về dữ liệu chưa quan sát
  - *Mục đích sau cùng*: thông tin về dữ liệu tương lai đúng càng nhiều càng tốt

# Giải thích dữ liệu thế nào là tốt (trong bài toán spam filtering)?

- Thông tin di chuyển như thế nào trong bài toán ML?
  - *Thông tin hiện tại*: dữ liệu hiện tại đang có
  - *Thông tin mới*: thông tin về dữ liệu chưa quan sát
  - *Mục đích sau cùng*: thông tin về dữ liệu tương lai đúng càng nhiều càng tốt
- Thiết kế hàm mất mát (loss) $L : \mathcal{H} \to \mathbb{R}$
  - $\forall f \in \mathcal{H}, L(f)$ đánh giá xem hàm $f$ có đạt được *mục đích sau cùng* hay không.
  - Thường $L(f)$ càng nhỏ thì $f$ càng làm tốt nhiệm vụ

# Tối ưu hoá (optimization)

## Ví dụ bài toán optimization

*Cho:* hàm số $f : x \mapsto x^2 - 4x + 4$
*Tìm:* $x^* \in \mathbb{R}$ sao cho giá trị của $f(x^*)$ nhỏ nhất

# Tối ưu hoá (optimization)

## Ví dụ bài toán optimization

*Cho:* hàm số $f : x \mapsto x^2 - 4x + 4$
*Tìm:* $x^* \in \mathbb{R}$ sao cho giá trị của $f(x^*)$ nhỏ nhất

## Bài toán optimization khái quát

*Cho:* tập hợp $A$, hàm số $f : A \to \mathbb{R}$
*Tìm:* $x^* \in A$ sao cho giá trị của $f(x^*)$ nhỏ nhất

# Tối ưu hoá (optimization)

## Ví dụ bài toán optimization

*Cho:* hàm số $f : x \mapsto x^2 - 4x + 4$
*Tìm:* $x^* \in \mathbb{R}$ sao cho giá trị của $f(x^*)$ nhỏ nhất

## Bài toán optimization khái quát

*Cho:* tập hợp $A$, hàm số $f : A \to \mathbb{R}$
*Tìm:* $x^* \in A$ sao cho giá trị của $f(x^*)$ nhỏ nhất

## Bài toán ML khái quát

*Cho:* gia đình hàm $\mathcal{H}$, hàm mất mát (loss) $L : \mathcal{H} \to \mathbb{R}$
*Tìm:* hàm số $f^* \in \mathcal{H}$ sao cho giá trị của $L(f^*)$ nhỏ nhất

# ML $\subset$ Optimization

**Định nghĩa vắn tắt II**

*Bài toán ML* là bài toán tìm hàm số giải thích dữ liệu trong một gia đình hàm nào đó bằng optimization.

# Sơ lược

# Bài toán nhỏ

- Hoàn cảnh
  - Cho biến $x, y$
  - Cho biết hàm số $f$ liên hệ $x$ với $y$ có bậc $= 1$
- Dữ liệu
  - Nếu $x = 1$ thì $y = 1$
  - Nếu $x = 2$ thì $y = 5$
- Tìm $f$

# Bài toán lớn hơn

- Hoàn cảnh
  - Cho biến $x$, $y$
  - Cho biết tồn tại hàm $f$ bậc 1 sao cho $f(x)$ 'gần' với $y$
- Dữ liệu
  - Nếu $x = 1$ thì $y = 1$
  - Nếu $x = 2$ thì $y = 5$
  - Nếu $x = 3$ thì $y = 6$
- Tìm $f$

# Bài toán 1 chiều

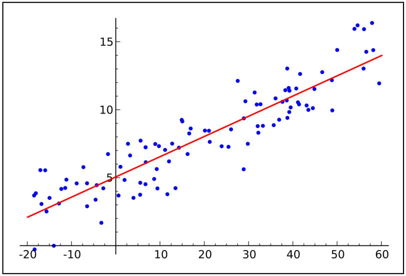
- Dữ liệu
  - $(x_i)_{i=1}^n \in \mathbb{R}$
  - $(y_i)_{i=1}^n \in \mathbb{R}$

# Bài toán 1 chiều

- Dữ liệu
  - $(x_i)_{i=1}^n \in \mathbb{R}$
  - $(y_i)_{i=1}^n \in \mathbb{R}$
- Cho
  - $\mathcal{H} := \{f : x \mapsto ax + b | a, b \in \mathbb{R}\}$
  - $L(f) := \sum_{i=1}^n (f(x_i) - y_i)^2$

# Bài toán 1 chiều

- Dữ liệu
  - $(x_i)_{i=1}^n \in \mathbb{R}$
  - $(y_i)_{i=1}^n \in \mathbb{R}$
- Cho
  - $\mathcal{H} := \{f : x \mapsto ax + b | a, b \in \mathbb{R}\}$
  - $L(f) := \sum_{i=1}^n (f(x_i) - y_i)^2$
- Tìm
  - $f^* \in \mathcal{H}$ sao cho $L(f^*)$ nhỏ nhất



# Bài toán $d$-chiều

- Dữ liệu
  - $(\boldsymbol{x}_i)_{i=1}^n \in \mathbb{R}^d$
  - $(y_i)_{i=1}^n \in \mathbb{R}$
- Cho
  - $\mathcal{H} \subseteq \{f : \boldsymbol{x} \mapsto \langle \boldsymbol{a}, \boldsymbol{x} \rangle + b | \boldsymbol{a} \in \mathbb{R}^d, b \in \mathbb{R}\}$
  - $L(f) = \sum_{i=1}^n (f(x_i) - y_i)^2$
- Tìm
  - $f^* \in \mathcal{H}$ sao cho $L(f^*)$ nhỏ nhất
    (hoặc $\boldsymbol{a} \in \mathbb{R}^d, b \in \mathbb{R}$ sao cho $L(\boldsymbol{x} \mapsto \langle \boldsymbol{a}, \boldsymbol{x} \rangle + b)$ nhỏ nhất)

# Giải

- Bài toán optimization này có *nghiệm giải tích hoàn toàn* (analytical solution), muốn biết kết quả chỉ cần bỏ dữ liệu vào *công thức chuẩn* (normal equation)

# Giải

- Bài toán optimization này có *nghiệm giải tích hoàn toàn* (analytical solution), muốn biết kết quả chỉ cần bỏ dữ liệu vào *công thức chuẩn* (normal equation)
- *Nhận xét 1:* không phải bài toán ML nào cũng phức tạp

# Giải

- Bài toán optimization này có *nghiệm giải tích hoàn toàn* (analytical solution), muốn biết kết quả chỉ cần bỏ dữ liệu vào *công thức chuẩn* (normal equation)
- *Nhận xét 1:* không phải bài toán ML nào cũng phức tạp
- *Nhận xét 2:* $\mathcal{H}$ đơn giản. $L$ là hàm liên tục, khả vi (theo tham số của $\mathcal{H}$)

# Giải

- Bài toán optimization này có *nghiệm giải tích hoàn toàn* (analytical solution), muốn biết kết quả chỉ cần bỏ dữ liệu vào *công thức chuẩn* (normal equation)
- *Nhận xét 1:* không phải bài toán ML nào cũng phức tạp
- *Nhận xét 2:* $\mathcal{H}$ đơn giản. $L$ là hàm liên tục, khả vi (theo tham số của $\mathcal{H}$)
- Trong thực tế, ít xài *công thức chuẩn* vì nó không *ổn định số học (numerically stable)*

# Học có giám sát (Supervised Learning)

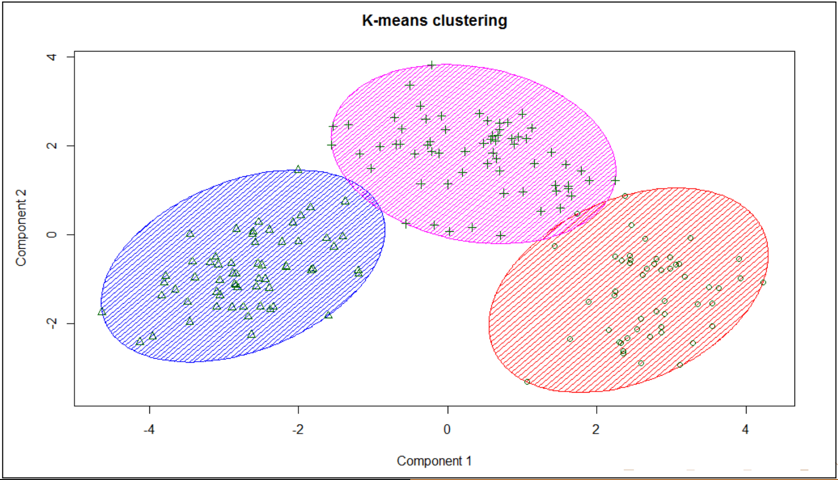
- Linear regression là một ví dụ về học có giám sát
- Dữ liệu được cung cấp có "mác", hàm số đầu ra đoán mác cho dữ liệu tương lai / chưa nhìn thấy

# Hồi quy (Regression) vs. Phân loại (classification)

- *Linear regression* là một ví dụ về bài toán hồi quy
  - Tóm tắt thông tin của dữ liệu cũ để làm thông tin dữ liệu mới
- *Spam filtering* là một ví dụ về bài toán phân loại
  - Dữ liệu cũ được chia thành 2 hay nhiều loại, tìm cách đoán xem dữ liệu mới thuộc loại nào

# Sơ lược

# Minh hoạ



# Bài toán

- Dữ liệu
  - $X = (x_i)_{i=1}^n \in \mathbb{R}^d$
  - *Không có mác!*

# Bài toán

- Dữ liệu
  - $X = (x_i)_{i=1}^n \in \mathbb{R}^d$
  - *Không có mác!*
- Cho
  - $\mathcal{H} \subseteq \{f : X \mapsto \{1, 2, \ldots, k\}\}$ tập hợp các hàm gán điểm trong $x \in X$ vào cluster $S_{f(x)}$
  - $L(f) = \sum_{i=1}^k |S_i| \text{Var}[S_i]$
    - $\text{Var}[S_i]$ đo phương sai của các điểm $x$ có $f(x) = i$

# Bài toán

- Dữ liệu
  - $X = (x_i)_{i=1}^n \in \mathbb{R}^d$
  - *Không có mác!*
- Cho
  - $\mathcal{H} \subseteq \{f : X \mapsto \{1, 2, \ldots, k\}\}$ tập hợp các hàm gán điểm trong $x \in X$ vào cluster $S_{f(x)}$
  - $L(f) = \sum_{i=1}^k |S_i| \text{Var}[S_i]$
    - $\text{Var}[S_i]$ đo phương sai của các điểm $x$ có $f(x) = i$
- Tìm
  - $f^* \in \mathcal{H}$ sao cho $L(f^*)$ nhỏ nhất

# Giải

- Không có nghiệm giải tích hoàn toàn. Dùng thuật toán dự đoán nghiệm (heuristics). Không có định lý về tính đúng sai.

# Giải

- Không có nghiệm giải tích hoàn toàn. Dùng thuật toán dự đoán nghiệm (heuristics). Không có định lý về tính đúng sai.
- *Thuật toán EM (Expectation - Maximization)* (không trong scope)

# Giải

- Không có nghiệm giải tích hoàn toàn. Dùng thuật toán dự đoán nghiệm (heuristics). Không có định lý về tính đúng sai.
- *Thuật toán EM (Expectation - Maximization)* (không trong scope)
- *Nhận xét 1:* Phần lớn bài toán ML đòi hỏi giải một bài optimization khó như kmeans, cần dùng những thuật toán *numerical methods* phức tạp, khó kiểm soát hơn.

# Giải

- Không có nghiệm giải tích hoàn toàn. Dùng thuật toán dự đoán nghiệm (heuristics). Không có định lý về tính đúng sai.
- *Thuật toán EM (Expectation - Maximization)* (không trong scope)
- *Nhận xét 1:* Phần lớn bài toán ML đòi hỏi giải một bài optimization khó như kmeans, cần dùng những thuật toán *numerical methods* phức tạp, khó kiểm soát hơn.
- *Nhận xét 2:* Lựa chọn $\mathcal{H}$ và $L$ cân bằng giữa độ khó của bài toán và ý nghĩa của bài toán.

# Học không giám sát

- *K-means* là ví dụ cho học không giám sát
  - Không có khái niệm dữ kiệu mới, cũ; chỉ đi tìm thông tin trong dữ liệu đang có
  - Dữ liệu đang có không được gán 'mác'

# Sơ lược

# Bài toán lớn hơn

- Hoàn cảnh
  - Cho biến $x$, $y$
  - Cho biết tồn tại hàm $f$ bậc 1 sao cho $f(x)$ 'gần' với $y$
- Dữ liệu
  - Nếu $x = 1$ thì $y = 1$
  - Nếu $x = 2$ thì $y = 5$
  - Nếu $x = 3$ thì $y = 6$
- Tìm $f$

# Gia đình hàm trong hồi quy tuyến tính

- $f(x) \neq y$

# Gia đình hàm trong hồi quy tuyến tính

- $f(x) \neq y$
- $f(x) \approx y$

# Gia đình hàm trong hồi quy tuyến tính

- $f(x) \neq y$
- $f(x) \approx y$
- $f(x) = ax + b + \epsilon$, trong đó $a, b \in \mathbb{R}$ còn $\epsilon$ là sai số ngẫu nhiên

# Mô hình thống kê (Statistical model)

- $\epsilon$ trong $f(x) = ax + b + \epsilon$ là 1 biến ngẫu nhiên

# Mô hình thống kê (Statistical model)

- $\epsilon$ trong $f(x) = ax + b + \epsilon$ là 1 biến ngẫu nhiên
- $f$ là 1 hàm số ngẫu nhiên

# Mô hình thống kê (Statistical model)

- $\epsilon$ trong $f(x) = ax + b + \epsilon$ là 1 biến ngẫu nhiên
- $f$ là 1 hàm số ngẫu nhiên
- $\mathcal{H} = \{x \to ax + b + \epsilon | a, b \in \mathbb{R}\}$ là 1 mô hình thống kê

# Tại sao cần biến ngẫu nhiên

- Mô hình mẫu của dữ liệu
- Lý thuyết ngoại suy, nội suy
- Mục đích của máy học

# Mô hình mẫu

- Bài toán ML = optimization + thống kê
- Dùng xác suất thống kê dựng mô hình mẫu cho dữ liệu

**Định nghĩa vắn tắt III**

*Bài toán ML* là bài toán tìm tham số của mô hình thống kê giải thích dữ liệu (data)

# Ước lượng tham số (Parameter Fitting)

**Định nghĩa vắn tắt III**

*Bài toán ML* là bài toán tìm tham số của mô hình thống kê giải thích dữ liệu (data)

- Tìm tham số như thế nào?

# Ước lượng (Estimator)

- Hoàn cảnh
  - Cho biết $y = ax + b + \epsilon$ với tham số $a$, $b$ nào đó, $\epsilon$ là biến ngẫu nhiên
- Dữ liệu
  - Nếu $x = 1$ thì $y = 1$
  - Nếu $x = 2$ thì $y = 5$
  - Nếu $x = 3$ thì $y = 6$
- Tìm $f$ (hoặc tìm $a, b$)
  - Dùng dữ liệu hữu hạn để đoán tham số $\hat{a}(\epsilon), \hat{b}(\epsilon)$

## Ước lượng

Ước lượng của một tham số là một cách dựa vào dữ liệu để đoán tham số.

# Ước lượng không chệch (unbiased estimator)

- Cho tham số $a$, ước lượng $\hat{a}$ phụ thuộc vào dữ liệu
- Chệch (bias) $:= E[\hat{a}] - a$

# Ước lượng không chệch (unbiased estimator)

- Cho tham số $a$, ước lượng $\hat{a}$ phụ thuộc vào dữ liệu
- Chệch (bias) $:= E[\hat{a}] - a$
- Ước lượng không chệch $\iff$ bias $= 0$

# Ước lượng hiệu quả (Efficient estimator)

- Phương sai của ước lượng $Var[\hat{a}] = E[(E[\hat{a}] - \hat{a})^2]$

# Ước lượng hiệu quả (Efficient estimator)

- Phương sai của ước lượng $Var[\hat{a}] = E[(E[\hat{a}] - \hat{a})^2]$
- Mean squared error (MSE) $:= E[(a - \hat{a})^2]$

# Ước lượng hiệu quả (Efficient estimator)

- Phương sai của ước lượng $Var[\hat{a}] = E[(E[\hat{a}] - \hat{a})^2]$
- Mean squared error (MSE) $:= E[(a - \hat{a})^2]$
- Ước lượng *không chệch* với phương sai nhỏ nhất $:=$ ước lượng hiệu quả

# Cân bằng độ lệch - phương sai

- $MSE =$ phương sai $+ bias^2$

# Cân bằng độ lệch - phương sai

- $MSE =$ phương sai $+ bias^2$
- Phương sai lớn biểu diễn overfit
- Chệch lớn biểu diễn underfit

# Cân bằng độ lệch - phương sai

- $MSE =$ phương sai $+ bias^2$
- Phương sai lớn biểu diễn overfit
- Chệch lớn biểu diễn underfit
- Để tìm ước lượng tốt cần giảm cả 2.

# Sơ lược

# Q&A

Q&A

# (Extra) Giải Kmeans chính xác khó thế nào