GIỚI THIỆU VỀ HỌC CÓ GIÁM SÁT

Học có giám sát

- Xét: $$Y = f(X) + \epsilon$$
- Các phương pháp học có giám sát:
  - Học bởi các ví dụ (quan sát)-"Learn by example"
  - Xây dựng mô hình $\hat{f}$ sử dụng tập các quan sát đã được gắn nhãn

$$\left(X^{(1)}, Y^{(1)}\right), \ldots, \left(X^{(n)}, Y^{(n)}\right)$$

Nhận dạng đối tượng

Nhận dạng đối tượng

- Trong nhận dạng đối tượng, chúng ta muốn một bộ phân lớp nhận vào 1 ảnh và cho biết lớp của đối tượng có trong ảnh. (Xác thực I'm not robot trên google)
- Các bộ phân lớp thường học bởi các thuật toán học có giám sát.
  - Dữ liệu huấn luyện trong ImageNet gồm: 1000 lớp với hơn 1 triệu ảnh huấn luyện
  - Sai số của bộ phân lớp từ 2010 đến 2015: 28.2, 25.8, 16.4, 11.7, 6.7, 3.6
  - Bây giờ khả năng phần lớp của ImageNet xấp xỉ khả năng phân lớp của con người

Lọc thư rác

- Trong lọc thư rác, cần 1 bộ phân lớp nhận vào 1 email và xác định email đó là thư rác (spam) hay không (ham hoặc not spam)
- Thường được tạo ra bằng cách học.
- Được sử dụng rộng rãi trên hầu hết các tài khoản email của người dùng.

Tuy nhiên:

- Thay vì xác định 1 email là spam hay không, ta có thể đưa ra 1 số thực thể hiện xác suất email đó là spam.
- Thuật toán được xây dựng bằng cách sử dụng các ngưỡng khác nhau để cực tiểu hóa lỗi dương tính sai (false positive)
- Khi đó ta có bài toán hồi quy, cụ thể là hồi quy logistic

Dự đoán giá nhà, giá cổ phiếu

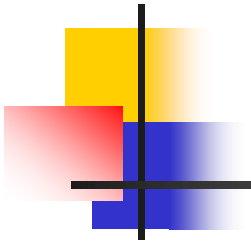
- Dự đoán giá nhà dựa vào vị trí, số phòng, ...
- Dự đoán giá cổ phiếu dựa vào lượng giao dịch trong các ngày trước đó,...

Đây là các bài toán hồi quy.

Học có giám sát

- Giải thuật học có giám sát
  - Lấy hàm ước lượng "tốt nhất" $\hat{f}$ trong tập các hàm
- Ví dụ: Hồi quy tuyến tính
  - Chọn 1 ước lượng tốt nhất từ *dữ liệu học* trong tập các hàm tuyến tính

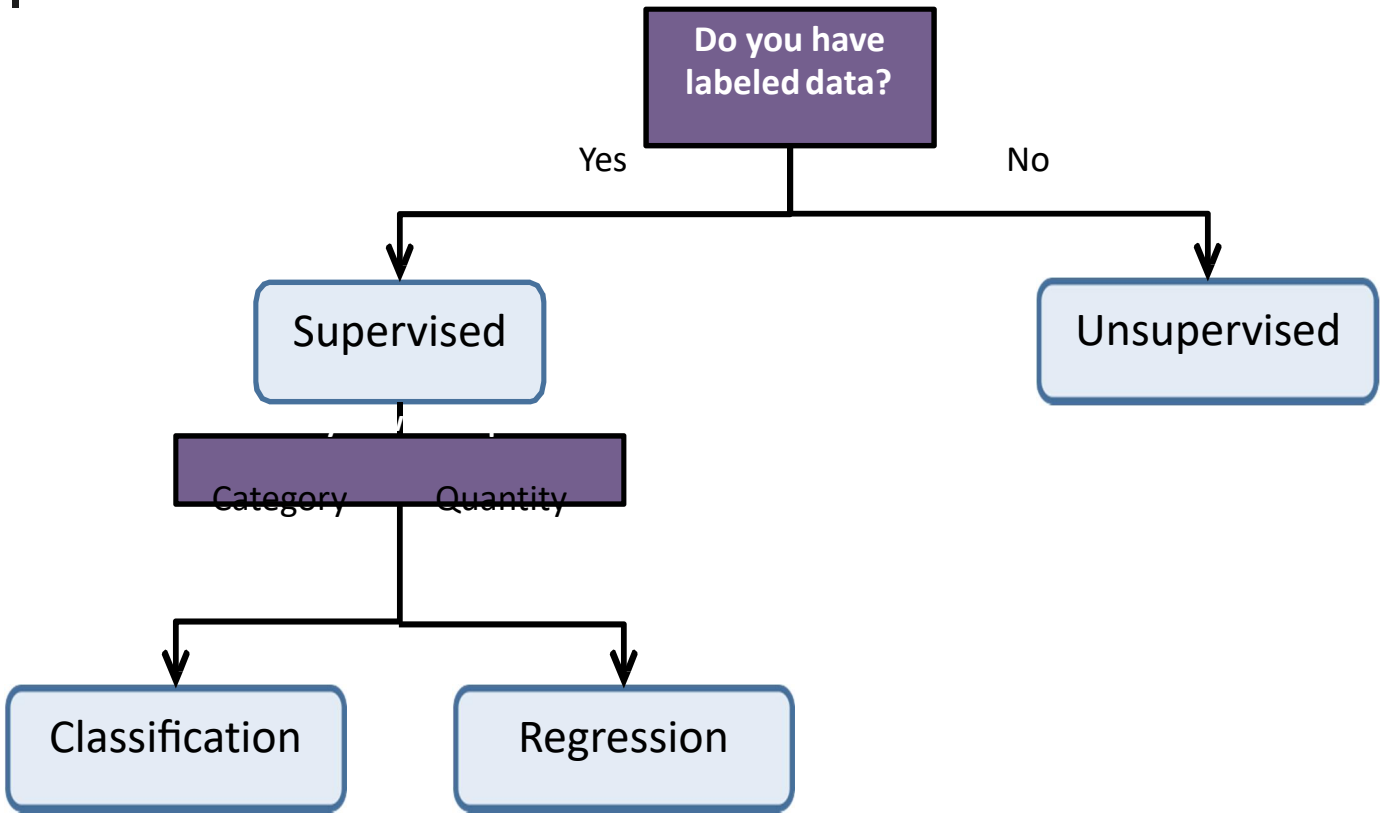
$$f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_d X_d$$



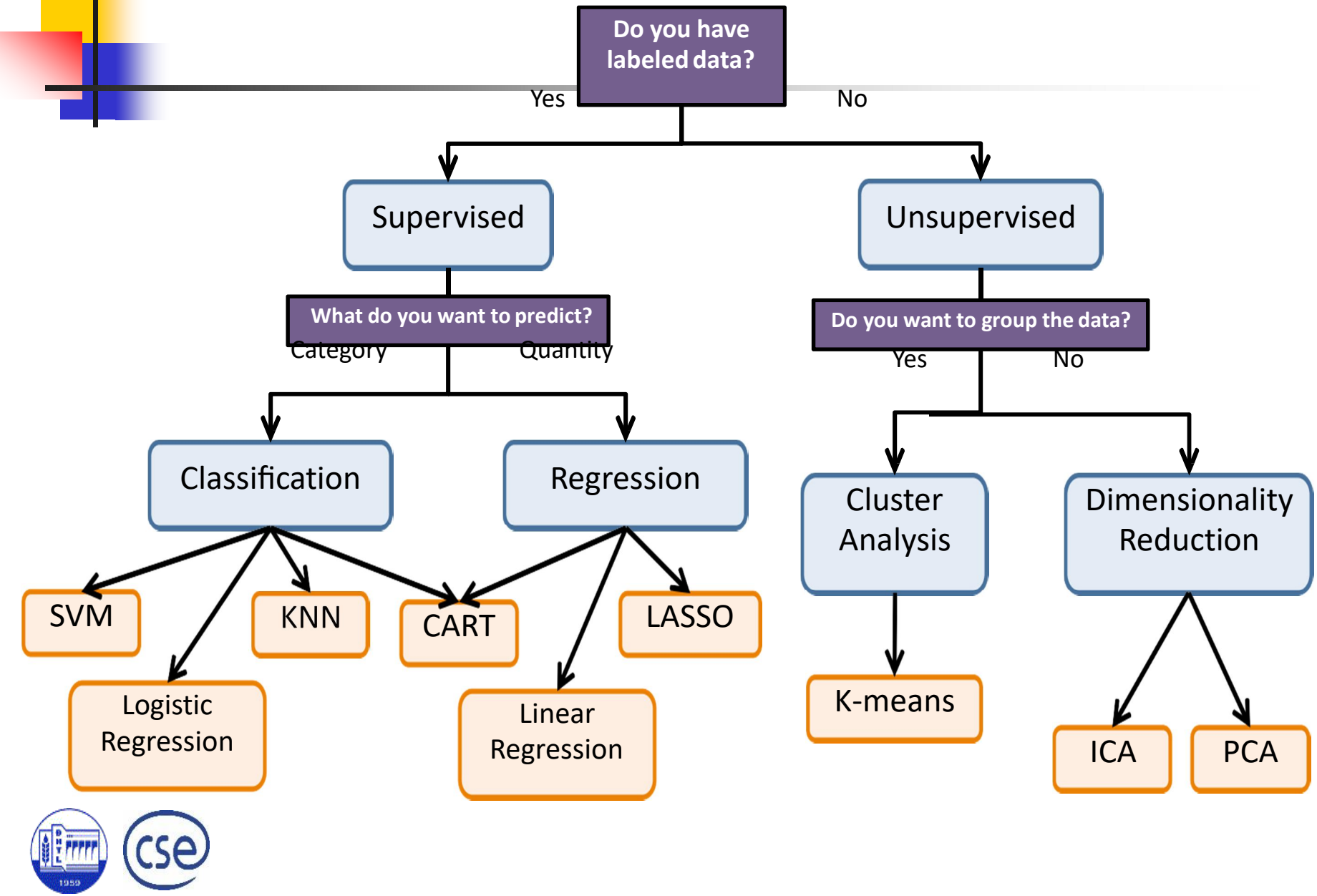
Phân lớp và Hồi quy

- Bài toán học có giám sát gồm 2 dạng:
  - Hồi quy: biến đầu ra Y là định lượng (quantitative)
  - Phân lớp: biến đầu ra Y là định tính/hạng mục/rời rạc

Các dạng giải thuật học máy



Các dạng giải thuật học máy



ĐỘ CHÍNH XÁC CỦA MÔ HÌNH

Đo hiệu năng bài toán hồi quy

- Hàm tổn thất (Loss function): loại hàm dùng để đo lường sai số của mô hình
- Vd: Sai số bình phương trung bình (Mean squared error - MSE)
  - Độ đo thông dụng dùng để tính độ chính xác bài toán hồi quy

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

  - Tập trung đo các sai số lớn hơn là các sai số nhỏ

Đo hiệu năng bài toán hồi quy

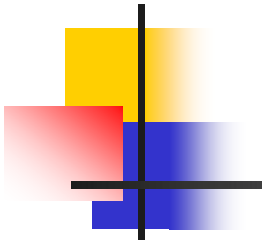
- Mục tiêu: xây dựng mô hình khái quát hóa (*generalizes)*

Ta muốn cực tiểu hóa lỗi trên dữ liệu chưa biết, không phải trên dữ liệu học.

VD: Dự đoán giá cổ phiếu *trong tương lai* dựa trên giá cổ phiếu trong quá khứ

- Chúng ta muốn cực tiểu tổn thất kỳ vọng (*expected* loss):

Vấn đề: Không thể cực tiểu lỗi trên dữ liệu huấn luyện.



Overfitting

- *Quá khớp (Overfitting)*: Học sự biến thiên ngẫu nhiên trong dữ liệu hơn là xu hướng cơ bản
- Đặc điểm của overfitting: Mô hình có hiệu năng cao trên tập dữ liệu học nhưng kém trên tập dữ liệu thử nghiệm.

Underfitting và Overfitting

- Có 50 điểm dữ liệu được tạo bằng một đa thức bậc ba cộng thêm nhiễu.
- Đồ thị của đa thức (true model) có màu xanh lá cây
- Bài toán: Giả sử ta không biết mô hình ban đầu mà chỉ biết các điểm dữ liệu, hãy tìm một mô hình "tốt" để mô tả dữ liệu đã cho?

Underfitting và Overfitting

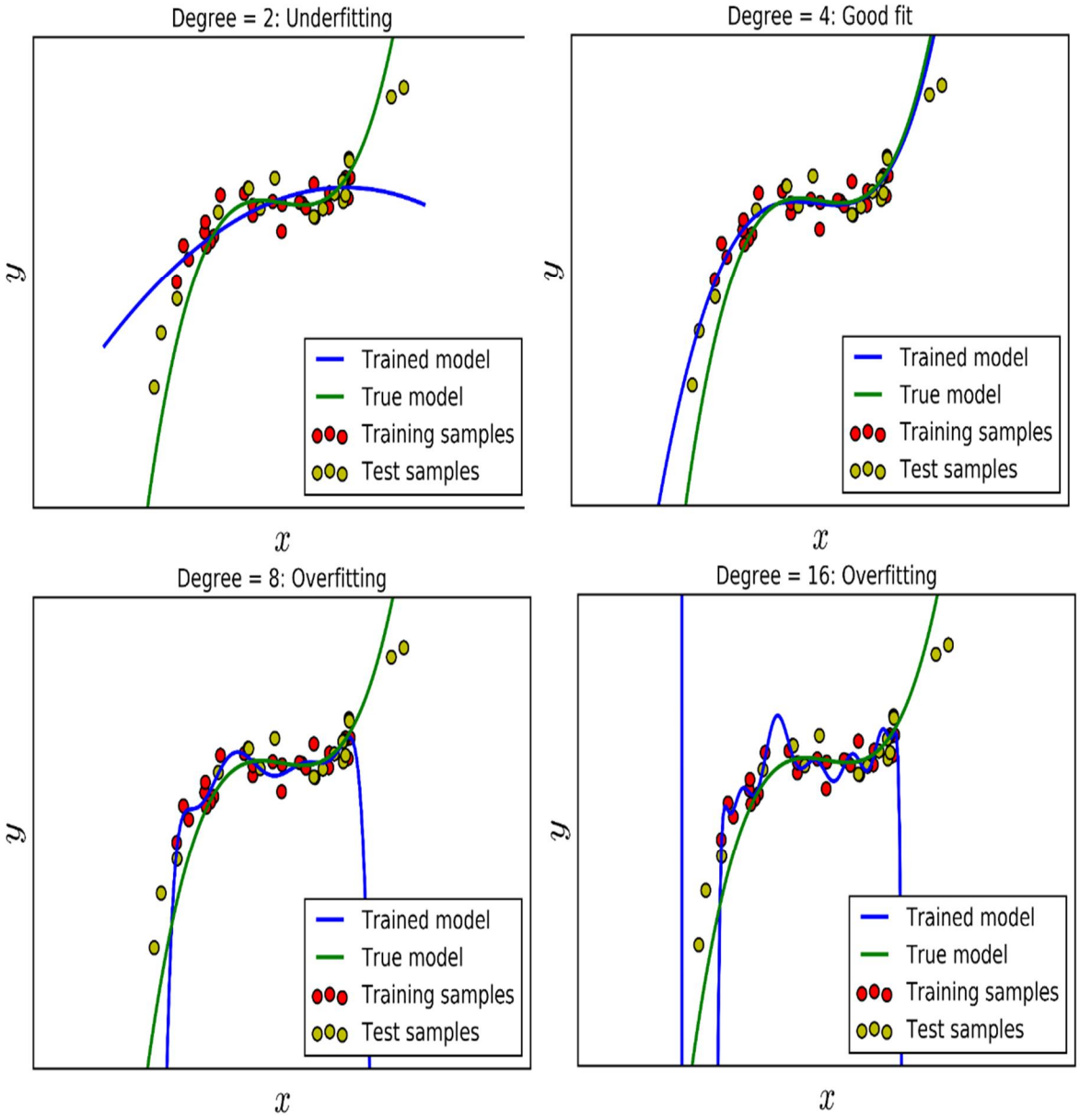
- Với d=2, mô hình không thực sự tốt vì dự đoán quá khác so với mô hình thực

→ *underfitting*

- Với d=8 và d=16, với các điểm dữ liệu trong khoảng của training data, mô hình dự đoán và mô hình thực là khá giống nhau. Tuy nhiên, về phía phải, đa thức bậc 8 và 16 cho kết quả hoàn toàn ngược với xu hướng của dữ liệu

→ *Overfitting.*

- d=4, mô hình tốt nhất.



Đánh giá hiệu năng

- Lỗi huấn luyện và lỗi kiểm thử thể hiện khác nhau.

Khi tính linh hoạt của mô hình tăng lên:

- *Lỗi huấn luyện* giảm
- *Lỗi kiểm thử ban đầu* giảm, nhưng sau đó tăng lên do có hiện tượng overfitting → lỗi kiểm thử dạng chữ U - "U-shaped"

Đánh giá hiệu năng

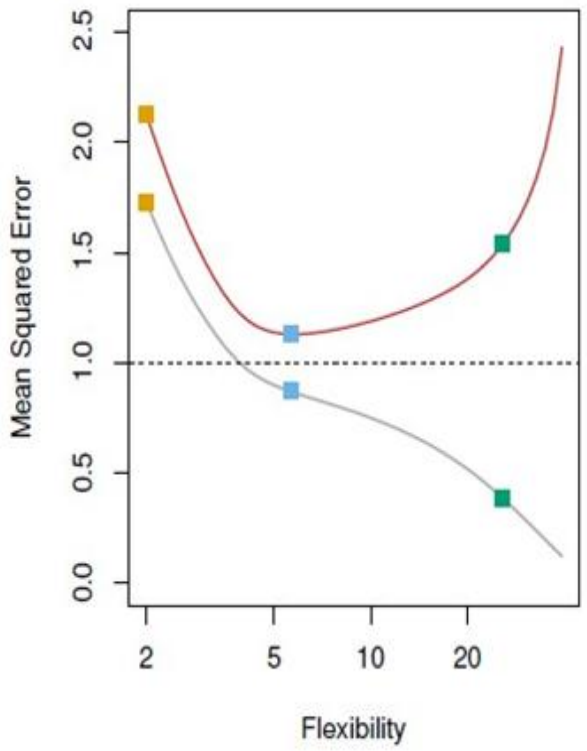


Figure 2.9 , ISL 2013

Đánh giá hiệu năng

- Làm sao để ước lượng lỗi kiểm thử để tìm một mô hình tốt?
- *Kỹ thuật kiểm tra chéo (Cross-validation hay CV)*

Một tập các kỹ thuật nhằm sử dụng dữ liệu huấn luyện để ước lượng lỗi tổng quát (generalization error)

Dữ liệu trong CV

- *Dữ liệu huấn luyện (Training data):* Tập các quan sát (bản ghi) được sử dụng để xây dựng (học) mô hình.
- *Dữ liệu kiểm chứng (Validation data):* Tập các quan sát dùng để ước lượng lỗi nhằm tìm tham số hoặc lựa chọn mô hình.
- *Dữ liệu kiểm thử (Test data):*
  - Tập các quan sát dùng để đánh giá hiệu năng trên dữ liệu chưa biết (unseen) trong tương lai.
  - Dữ liệu này không sử dụng cho giải thuật học máy trong quá trình xây dựng mô hình.

Trade-off: Độ lệch vs. Phương sai

- Lỗi kiểm thử đường cong hình chữ U (U-shaped) xảy ra dựa trên 2 đặc điểm của mô hình học máy:

$$\mathbb{E}\left[\text{test error}\right] = var(\hat{f}) + bias(\hat{f})^2 + var(\epsilon)$$

$var(\hat{f})$: *Phương sai (variance)* của hàm ước lượng

$bias(\hat{f})$: *Độ chệch/sai lệch (bias)* của hàm ước lượng

Trade-off: Độ lệch vs. Phương sai

- **Phương pháp đơn giản:** bias cao, phương sai thấp
- **Phương pháp linh hoạt (phức tạp):** bias thấp, phương sai cao

Trade-off: Độ lệch vs. Phương sai

- Dễ đạt được phương sai thấp/bias cao *hoặc* phương sai cao/bias thấp,
- Tuy nhiên rất khó để đạt được cả phương sai *và* bias cùng thấp

Kỹ thuật đánh giá chéo

"Dùng lỗi trên tập dữ liệu kiểm thử để ước lượng lỗi dự đoán"

$$err = E[L(Y, \hat{f}(X))]$$

Khi xảy ra hiện tượng Underfitting và Overfitting???

Tập đánh giá (Validation)

- Phương pháp: trích từ training data ra một tập con nhỏ và thực hiện việc đánh giá mô hình trên tập con này.
- Tập con nhỏ được trích ra từ training set này được gọi là validation set.
- Training set là phần còn lại của training set ban đầu (sau khi bỏ đi tập đánh giá).

Tập đánh giá (Validation)

- Train error: được tính trên training set mới này.
- Validation error: Lỗi được tính trên tập validation.
- Tìm mô hình sao cho cả *train eror* và *validation error* đều nhỏ, qua đó có thể dự đoán được rằng *test error* cũng nhỏ.

Tập đánh giá (Validation)

- Phương pháp thường được sử dụng là sử dụng nhiều mô hình khác nhau. Mô hình nào cho validation error nhỏ nhất sẽ là mô hình tốt.
- Tuy nhiên, khi ta có số lượng dữ liệu để xây dựng mô hình rất hạn chế, nếu lấy quá nhiều dữ liệu trong tập training ra làm dữ liệu validation, phần dữ liệu còn lại của tập training là không đủ để xây dựng mô hình.

Tập đánh giá (Validation)

- Nếu ta giữ tập validation phải thật nhỏ để có được lượng dữ liệu cho training đủ lớn. Một vấn đề khác nảy sinh, hiện tượng overfitting lại có thể xảy ra với tập training còn lại.
- Giải pháp: Cross-validation (Kỹ thuật đánh giá chéo).

Kỹ thuật đánh giá chéo

- *Cross validation* là một cải tiến của *validation* với lượng dữ liệu trong tập validation là nhỏ nhưng chất lượng mô hình được đánh giá trên nhiều tập *validation* khác nhau.
- Chia tập training ra *k* tập con không có phần tử chung, có kích thước gần bằng nhau.
- Tại mỗi lần kiểm thử, một trong số *k* tập con được lấy ra làm *validation set*. Mô hình sẽ được xây dựng dựa vào hợp của *k–1* tập con còn lại.

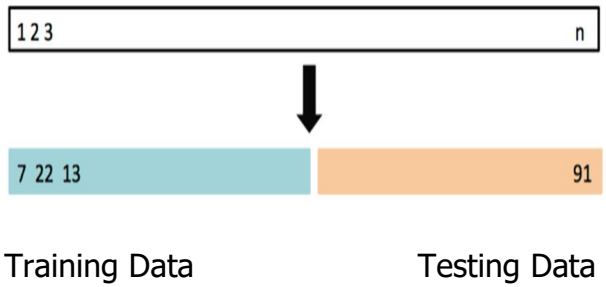
Kỹ thuật đánh giá chéo

- Mô hình cuối được xác định dựa trên trung bình của các train error và validation error. Cách làm này còn có tên gọi là k-fold cross validation.

Tập huấn luyện - Training Set

Tập kiểm thử - Test Set

Tập đánh giá - Validation Set



Kỹ thuật đánh giá chéo K-fold

Ví dụ 5-fold

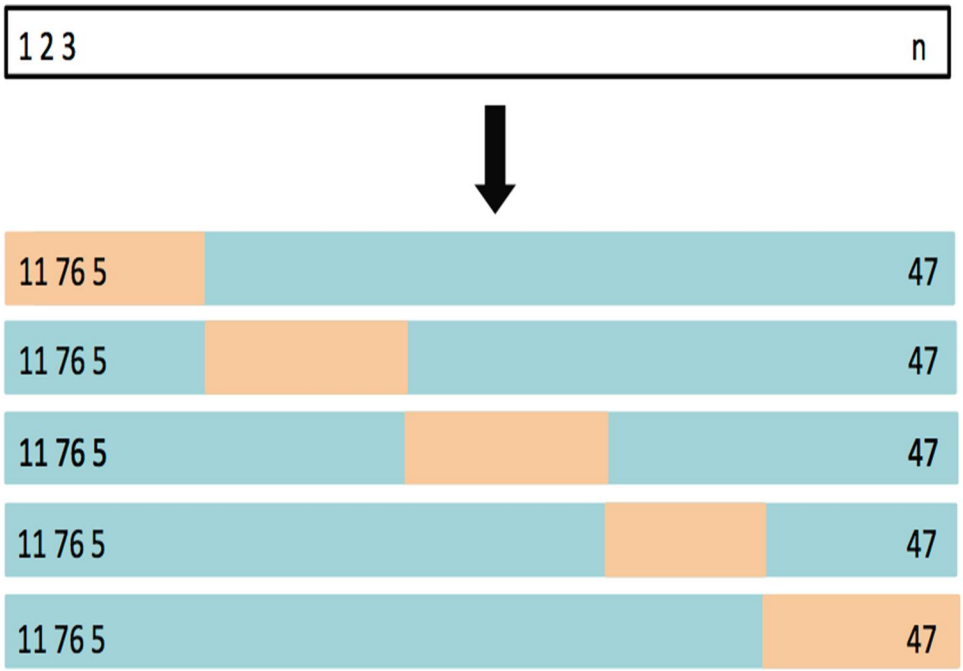
1	2	3	4	5
Train	Train	Validation	Train	Train

$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i))$$

Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Kỹ thuật đánh giá chéo

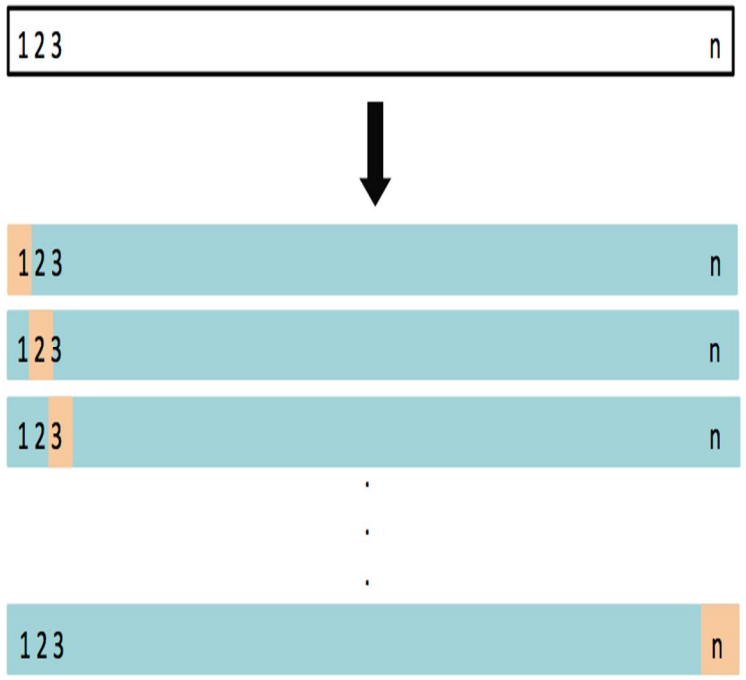
5-fold và 10-fold thường được ưa dùng (lỗi bias cao, phương sai thấp)



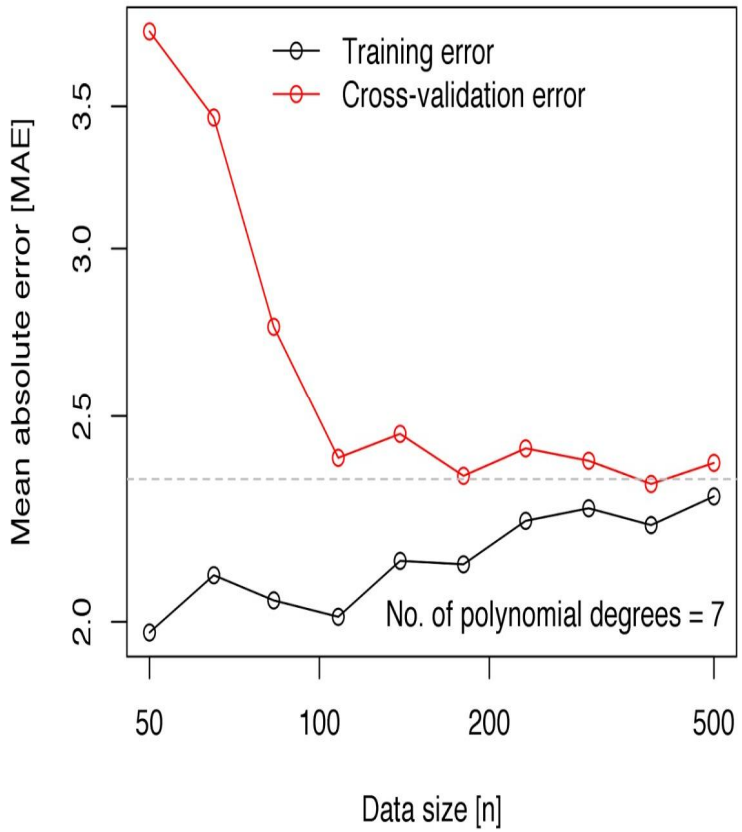
Kỹ thuật đánh giá chéo

- Khi ***k*** bằng với số lượng phần tử N trong tập *training* ban đầu, tức mỗi tập con có đúng 1 phần tử, ta gọi kỹ thuật này là leave-one-out cross validation (LOOCV).

(lỗi bias thấp, phương sai cao)



Kỹ thuật đánh giá chéo



Ta cần thêm biến (mô hình mới) hoặc thêm dữ liệu?

Nhược điểm của CV

- Nhược điểm lớn của *cross-validation* là số lượng *training runs* tỉ lệ thuận với k. Trong các bài toán Machine Learning, lượng tham số cần xác định thường lớn và khoảng giá trị của mỗi tham số cũng rộng. Việc xây dựng một mô hình cũng rất phức tạp.

$\Rightarrow$ Giải pháp giúp số mô hình cần huấn luyện giảm đi nhiều, thậm chí chỉ một mô hình. Cách này có tên gọi chung là *điều chỉnh mô hình* (*regularization*).

Điều chỉnh mô hình

- *Regularization*, một cách cơ bản, là điều chỉnh mô hình một chút để tránh overfitting trong khi vẫn giữ được tính tổng quát của nó (tính tổng quát là tính mô tả được nhiều dữ liệu, trong cả tập training và test).
- Một cách cụ thể hơn, ta sẽ tìm cách *di chuyển* nghiệm của bài toán tối ưu hàm tổn thất tới một điểm gần nó. Hướng di chuyển sẽ là hướng làm cho mô hình *ít phức tạp hơn* mặc dù giá trị của hàm tổn thất có tăng lên một chút.