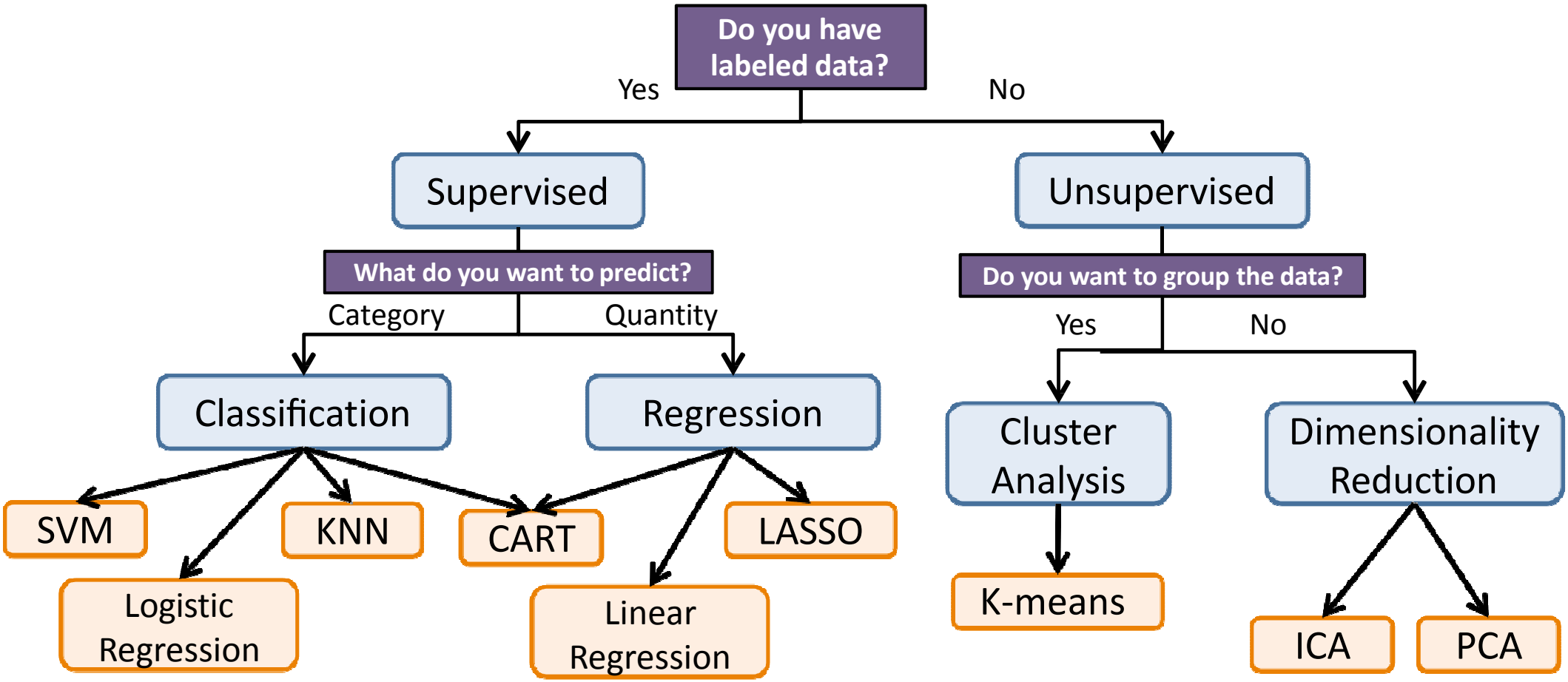
Kỹ thuật đánh giá chéo, hiệu chỉnh mô hình

Nguyễn Thanh Tùng
Khoa Công nghệ thông tin – Đại học Thủy Lợi
tungnt@tlu.edu.vn

Website môn học: https://sites.google.com/a/wru.vn/cse445Fall2017

Bài giảng có sử dụng hình vẽ trong cuốn sách "An Introduction to Statistical Learning with Applications in R" với sự cho phép của tác giả, có sử dụng slides các khóa học CME250 của ĐH Stanford và IOM530 của ĐH Southern California

Các dạng giải thuật Học máy



Nhắc lại
Hồi quy tuyến tính đơn giản

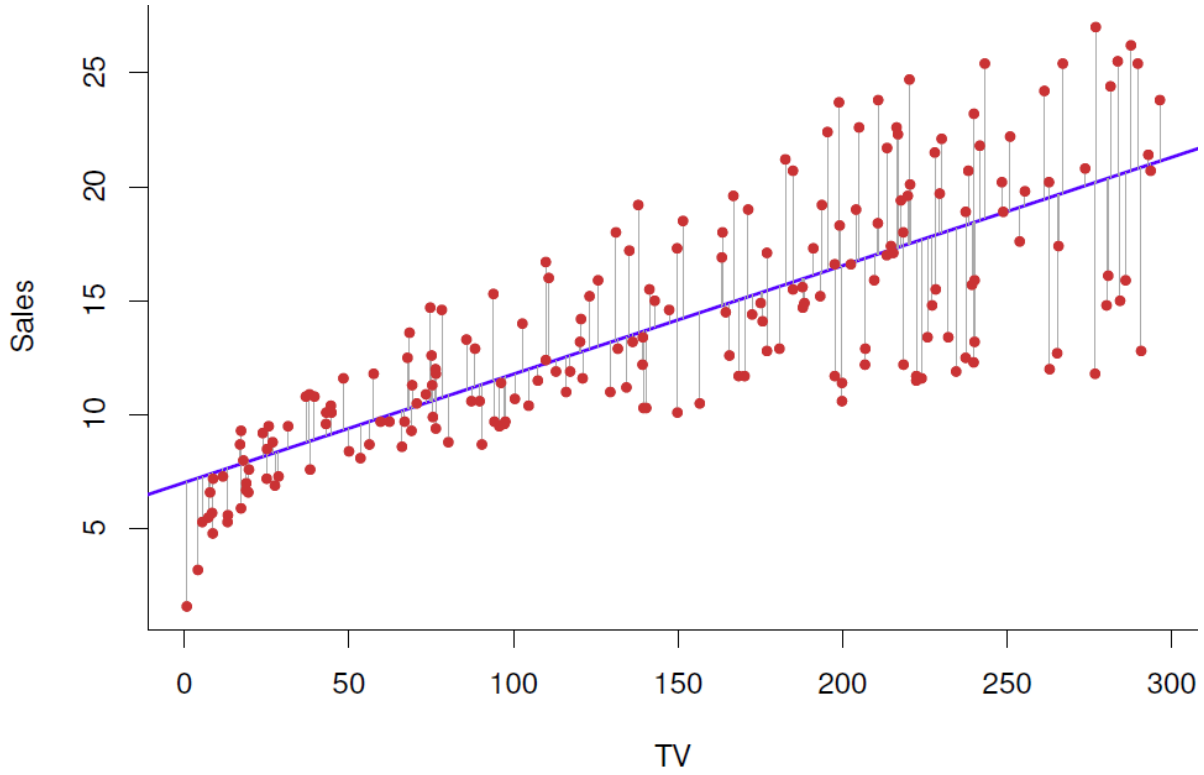


Figure 3.1 , ISL 2013

Nhắc lại
Bình phương nhỏ nhất

- Sử dụng phương pháp bình phương nhỏ nhất để đo lường độ xấp xỉ của mô hình áp dụng trên dữ liệu
- *Phần dư (Residual)*: sai số giữa giá trị quan sát được và giá trị dự đoán.

$$r^{(\mathrm{i})} = Y^{(\mathrm{i})} - \hat{Y}^{(\mathrm{i})}$$

- *Tổng phần dư bình phương-Residual sum of squares* (RSS):

$$RSS = \left(r^{(1)}\right)^2 + \left(r^{(2)}\right)^2 + \cdots + \left(r^{(\mathrm{n})}\right)^2$$

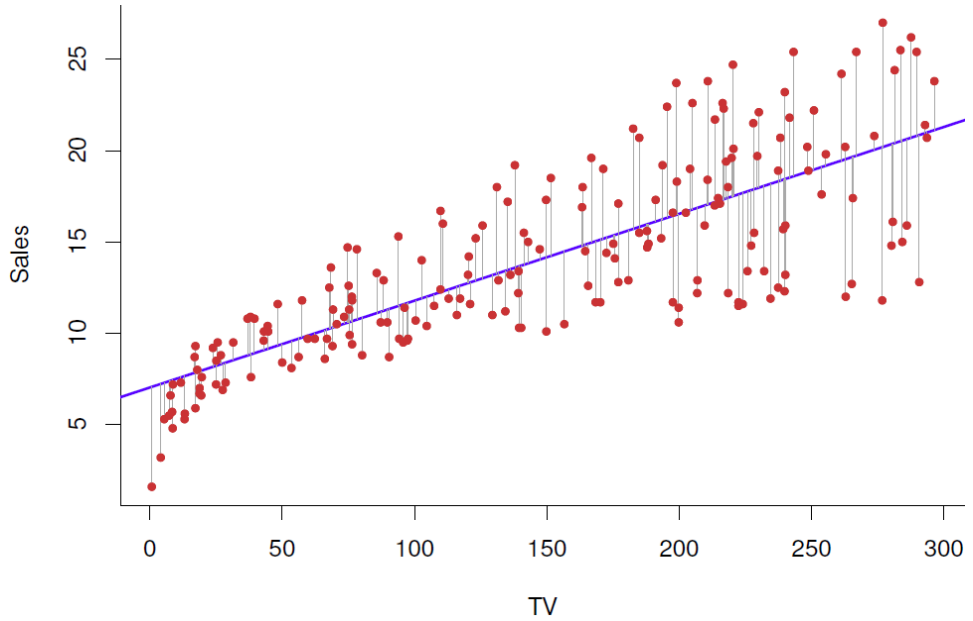
- *Lỗi bình phương trung bình-Mean squared error* (MSE):

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

Hàm tổn thất
Loss Functions

Loss Functions

$$L(\theta_i, \hat{\theta}_i)$$



Loss Functions

$$L(\theta_i, \hat{\theta}_i)$$

Sai số bình phương (Squared error)

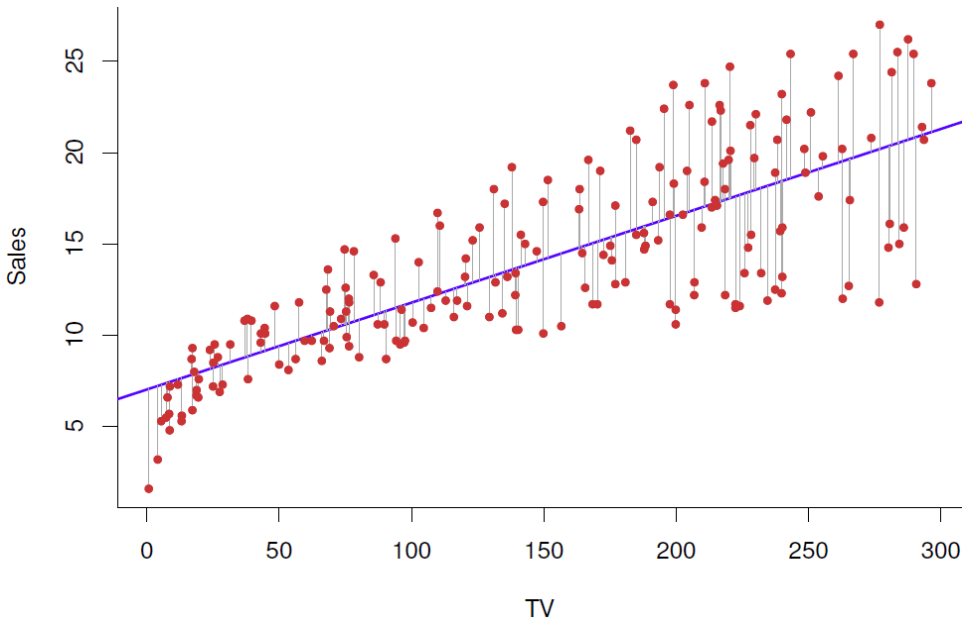
$$\sum_i (\theta_i - \hat{\theta}_i)^2$$

Sai số tuyệt đối (Absolute error)

$$\sum_i \left|\theta_i - \hat{\theta}_i\right|$$

Indicator error

$$\sum_i I(\theta_i \neq \hat{\theta}_i)$$



Học máy chỉ để giải 1 vấn đề

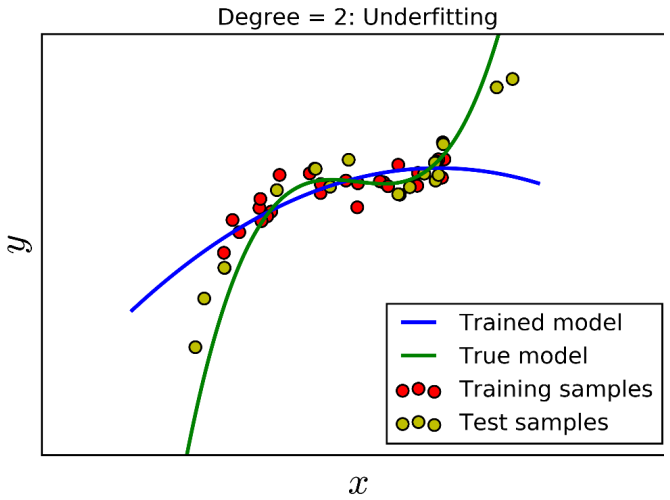
$$\hat{f} = \underset{\tilde{f}}{\operatorname{argmin}} E[L(Y, \tilde{f}(X))]$$

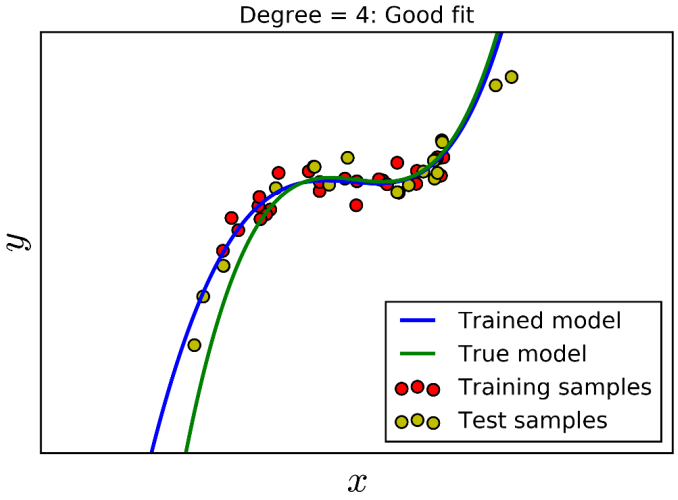
argument minimum: Cho giá trị nhỏ nhất của 1 hàm số trong miền xác định

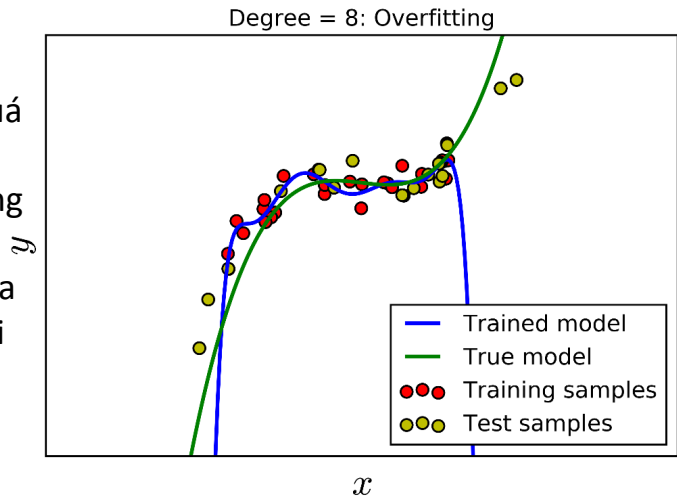
Hiện tượng quá khớp
Overfitting

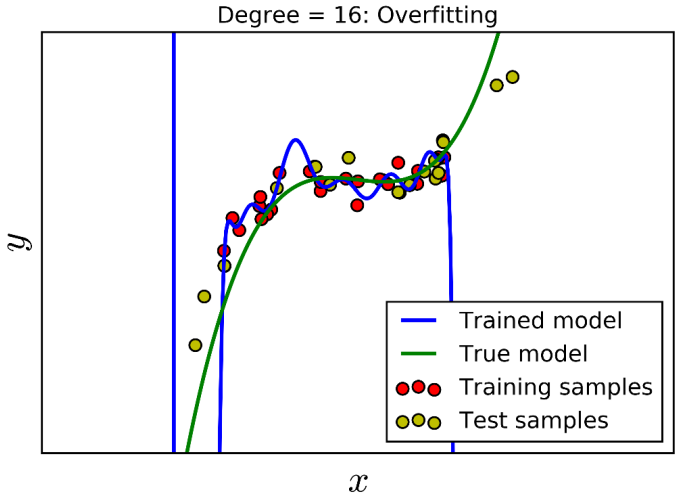
Underfitting và Overfitting

- Có 50 điểm dữ liệu được tạo bằng một đa thức bậc ba cộng thêm nhiễu.
- Đồ thị của đa thức có màu xanh lục (true model).
- 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?









- 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.

Kỹ thuật đánh giá chéo
Cross-validation

Kỹ thuật đánh giá chéo

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

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

Tập đánh giá (Validation)

- Thường chia tập dữ liệu ra thành training data và test data.
- Chú ý: khi xây dựng mô hình, ta không được sử dụng test data.
- Làm cách nào để biết được chất lượng của mô hình với unseen data (tức dữ liệu chưa nhìn thấy bao giờ)?

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. Lúc này, training set là phần còn lại của training set ban đầu.
- Train error được tính trên training set mới này.
- Validation error: Lỗi được tính trên tập validation.

Tập đánh giá (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.
- 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.

Tập đánh giá (Validation)

- Tuy nhiên, khi ta có rất hạn chế số lượng dữ liệu để xây dựng mô hình. 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.
- 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).

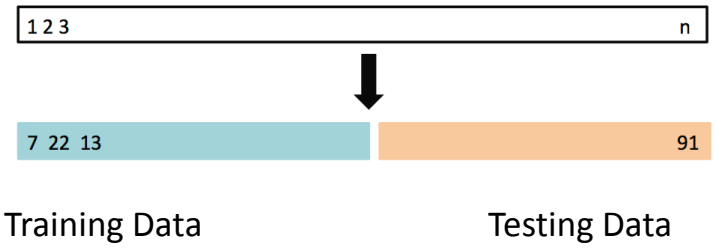
Kỹ thuật đánh giá chéo

- *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 *validata set*. Mô hình sẽ được xây dựng dựa vào hợp của ***k−1*** tập con còn lại.
- 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

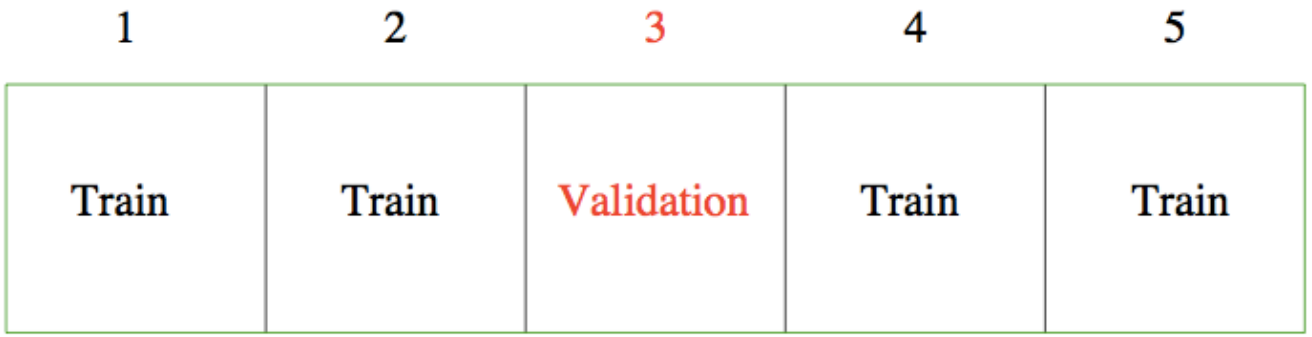
Tập kiểm thử - Test Set

Tập đánh giá - Validation Set



Kỹ thuật đánh giá chéo K–fold

Ví dụ 5–fold

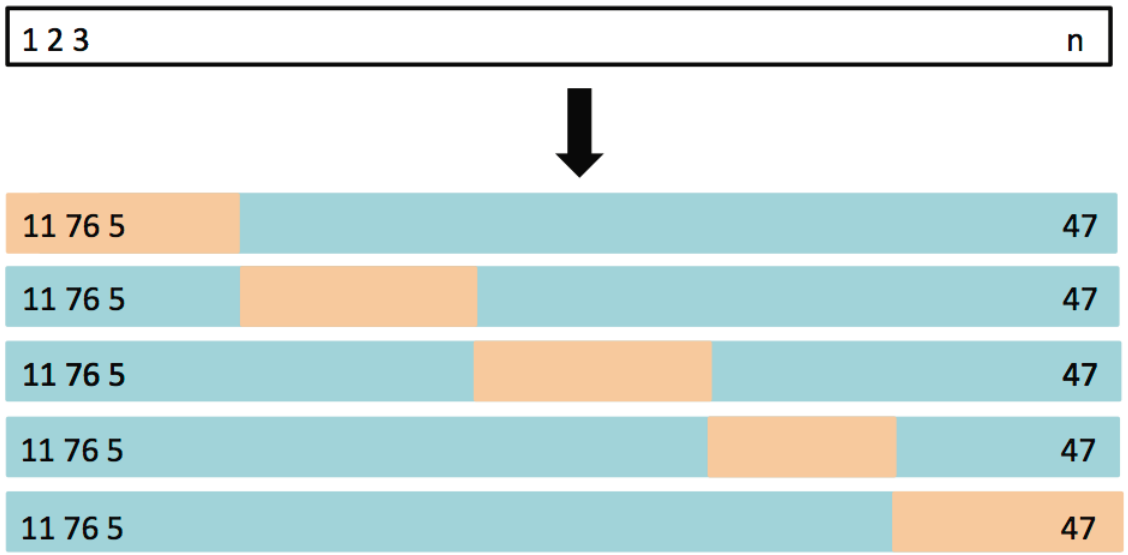


$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i))$$

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

Kỹ thuật đánh giá chéo

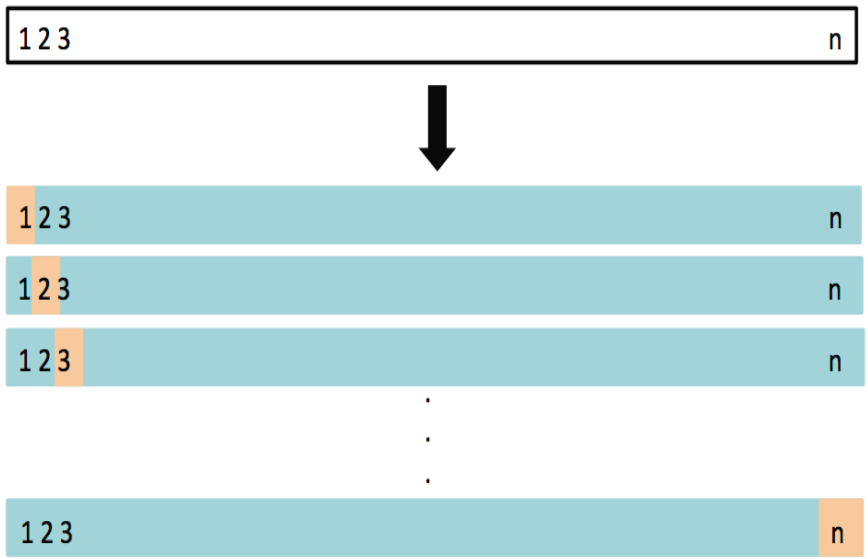
5-fold và 10-fold thường được ưa dùng (lỗi bias cao, phương sai thấp)



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**.

(lỗi bias thấp, phương sai cao)



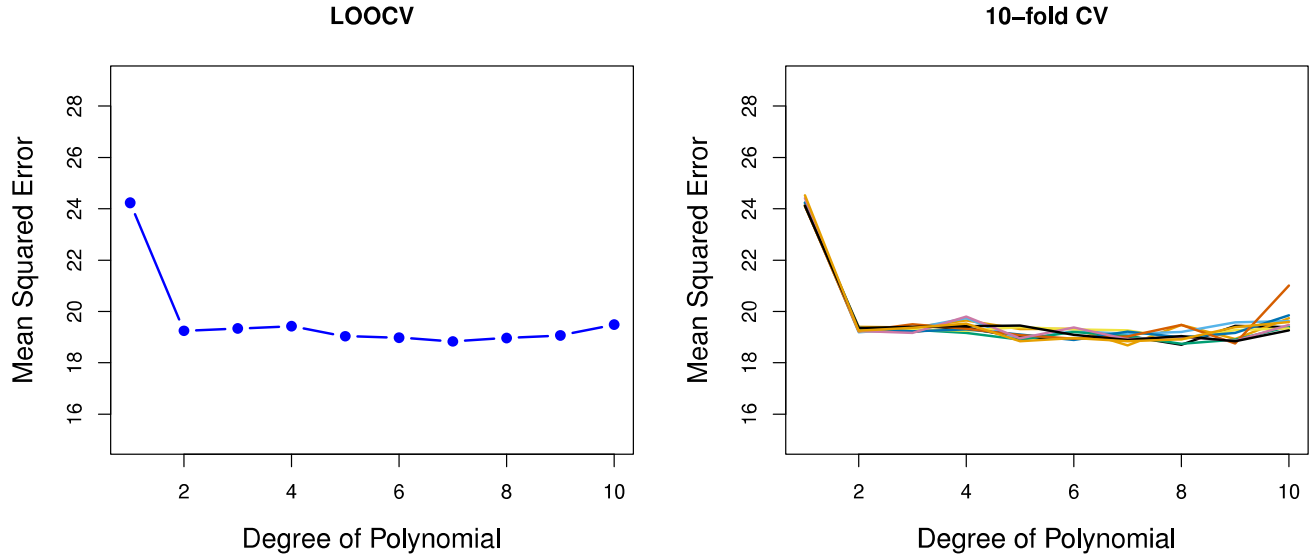
Auto Data: LOOCV vs. K-fold CV

Hình trái: Sai số LOOCV

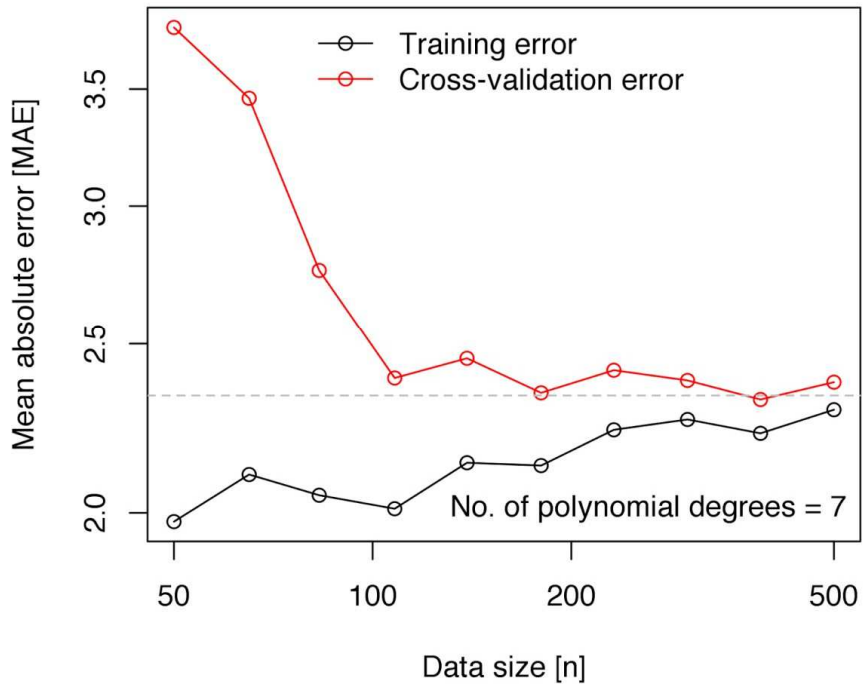
Hình phải: 10-fold CV được chạy nhiều lần, đồ thị biểu diễn sai khác nhỏ về lỗi CV

LOOCV là trường hợp đặc biệt của k-fold, khi k = N

Cả hai đều ổn định, tuy nhiên LOOCV mất nhiều thời gian tính toán hơn!



Kỹ thuật đánh giá chéo



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

Kỹ thuật đánh giá chéo

- 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.
- Vậy việc chỉ xây dựng một mô hình thôi đã rất phức tạp.
- 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 mất mát 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 mất mát có tăng lên một chút.

Mô hình có điều chỉnh

Nhắc lại: Hồi quy tuyến tính đa biến

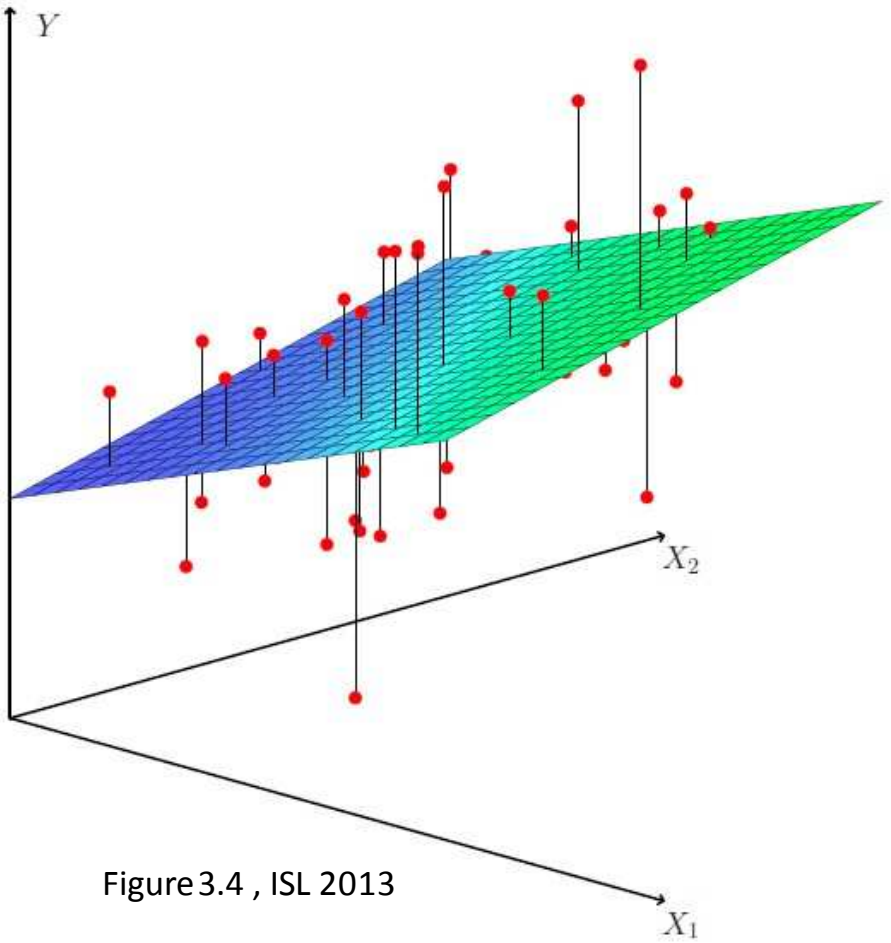


Figure 3.4 , ISL 2013

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2$$

Trường hợp quá nhiều biến

khi có quá nhiều biến đầu vào

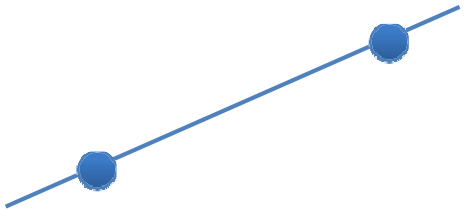
$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

khi có tương tác giữa các biến đầu vào

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot (X_1 X_2) + \beta_4 \cdot X_1^2 + \beta_5 \cdot X_2^2 + \beta_6 \cdot \log(X_1 / X_2) + \beta_7 \cdot \sin(X_1 - X_2)$$

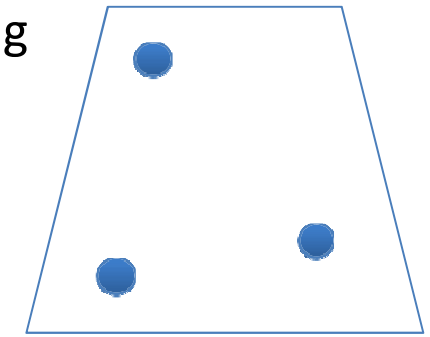
Trường hợp quá nhiều biến

Hai mẫu xác định 1 đường thẳng



$$Y = \beta_0 + \beta_1 \cdot X_1$$

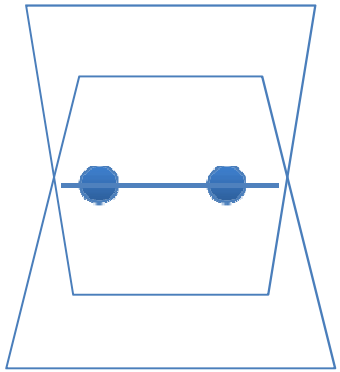
Ba mẫu xác định 1 mặt phẳng



$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2$$

Trường hợp quá nhiều biến

Hai mẫu không xác định một mặt phẳng duy nhất

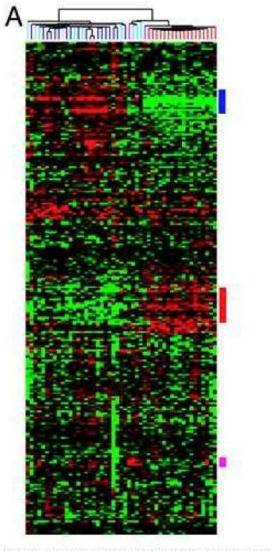


$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2$$

Trường hợp quá nhiều biến

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

Gene expression arrays



Điều gì xảy ra?

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

Câu hỏi: Ta có 8 biến và có hàng trăm mẫu. Hai biến (X_3 và X_4) có tương quan yếu với Y (do đó cũng hữu dụng nhỏ cho dự đoán), tuy nhiên chúng có tương quan cao với các biến khác. Điều gì xảy ra khi diễn giải các hệ số β của hai biến X_3 và X_4?

Đa cộng tuyến (Multi-collinearity)

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

- Theo giả thiết của phương pháp Hồi quy tuyến tính thì các biến độc lập không có mối quan hệ tuyến tính.
- Nếu quy tắc này bị vi phạm thì sẽ có hiện tượng đa cộng tuyến: là hiện tượng các biến độc lập trong mô hình phụ thuộc tuyến tính lẫn nhau và thể hiện được dưới dạng hàm số

Ta cần phải làm gì?

Phạt các hệ số β lớn.

Hiệu chỉnh mô hình

Phạt (Penalty): Đưa vào một *đại lượng điều chỉnh* (regularization term hoặc regularizer) khi đánh giá ϵ_{test}:

$$\boldsymbol{\epsilon_{test} = \epsilon_{train} + penalty}.$$

Khi huấn luyện mô hình, ta tìm $\boldsymbol{w}$ để cực tiểu hóa hàm mục tiêu $J(\boldsymbol{w}) = \epsilon_{train}(\boldsymbol{w}) + penelty(\boldsymbol{w})$

Hiệu chỉnh mô hình

Regularization là việc đưa một đại lượng điều chỉnh (regularizator or regularization term) vào quá trình học để ngăn cản hiện tượng quá khít

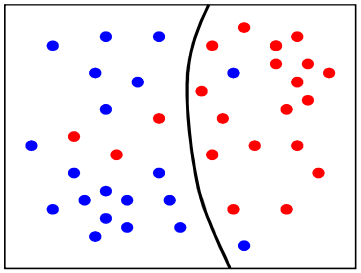
$$\epsilon[f] = \epsilon_{train}[f] + \lambda \times \text{regularizer}[f]$$

Đại lượng điều chỉnh thường liên quan đến độ phức tạp của lời giải

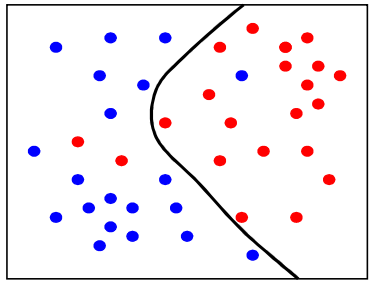
Hạn chế về độ mịn của hàm (smoothness)

Giới hạn chuẩn của không gian vector.

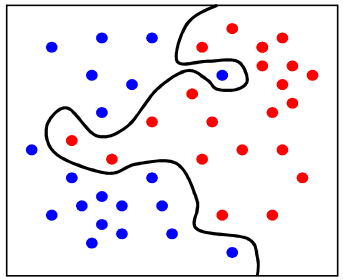
$\epsilon_{train}[f1] = 5/40$



$\epsilon_{train}[f2] = 3/40$



$\epsilon_{train}[f3] = 0$



Hồi quy tuyến tính đa biến

Quay lại hồi quy tuyến tính, ta cố gắng để cực tiểu hóa lỗi bình phương

$$\sum_{\text{các mẫu}} [Y - (\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2)]^2$$

Hồi quy Ridge

Tìm giá trị β để cực tiểu lỗi phạt "penalized", tương đương với

$$\sum_{samples} [Y - (\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2)]^2 + \lambda \cdot (\beta_0^2 + \beta_1^2 + \beta_2^2)$$

L2

Hiệu chỉnh mô hình (Regularization)

Hồi quy Ridge

Tìm giá trị β để cực tiểu lỗi phạt "penalized", tương đương với

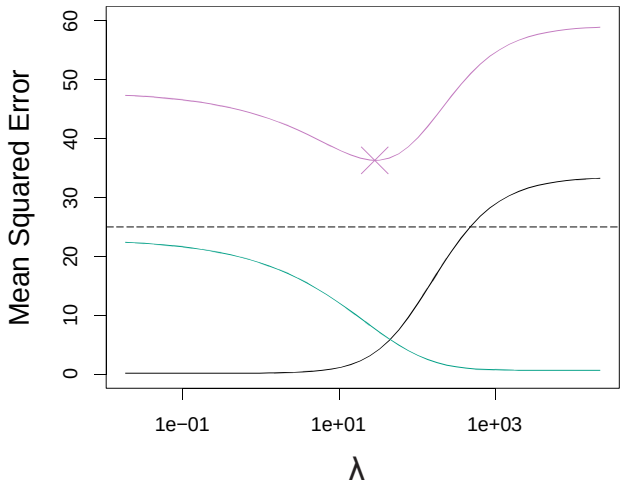
$$\sum_{\text{các mẫu}} [Y - (\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2)]^2 + \lambda \cdot (\beta_0^2 + \beta_1^2 + \beta_2^2)$$

L2

hoặc viết ở dạng khác,

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left(\|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 \right)$$

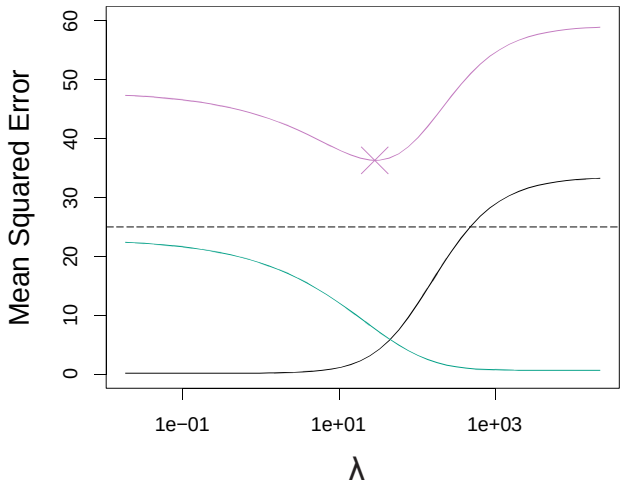
Hồi quy Ridge

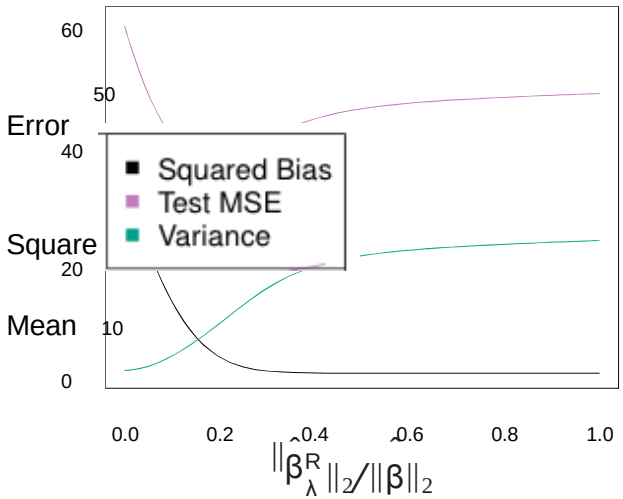


Đường cong nào là lỗi bias, đâu là phương sai, và đâu là lỗi dự đoán trên tập dữ liệu kiểm thử?

Hastie, Trevor, et al. Introduction to statistical learning.

Hồi quy Ridge





Hastie, Trevor, et al. Introduction to statistical learning.

Hiệu chỉnh mô hình

Ta đã xử lý:

- *Underdetermined*
- *Overfitting*
- Đa cộng tuyến (*Multi--collinearity)*

Vậy mô hình thưa là gì (*sparsity)*?

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

$\beta_2 \to 0$, $\beta_4 \to 0$, $\beta_7 \to 0$

Mô hình thưa (Sparsity)

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \beta_7 \cdot X_7 + \beta_8 \cdot X_8$$

$\beta_2 \to 0$, $\beta_4 \to 0$, $\beta_7 \to 0$

- Dùng cho lựa chọn biến (Feature selection)
- Thời gian tính toán lâu (computational efficiency)

Mô hình thưa (Sparsity)

Lasso

"**L**east **a**bsolute **s**hrinkage and **s**election **o**perator"

$$\sum_{samples} [Y - (\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2)]^2 + \lambda \cdot (|\beta_0| + |\beta_1| + |\beta_2|)$$

L1

Mô hình giống như hồi quy Rigde nhưng khác hàm phạt

Tibshirani, Robert. "Regression shrinkage and selection via the lasso." Journal of the Royal Statistical Society. Series B (Methodological)(1996): 267–288.

Lasso

"**L**east **a**bsolute **s**hrinkage and **s**election **o**perator"

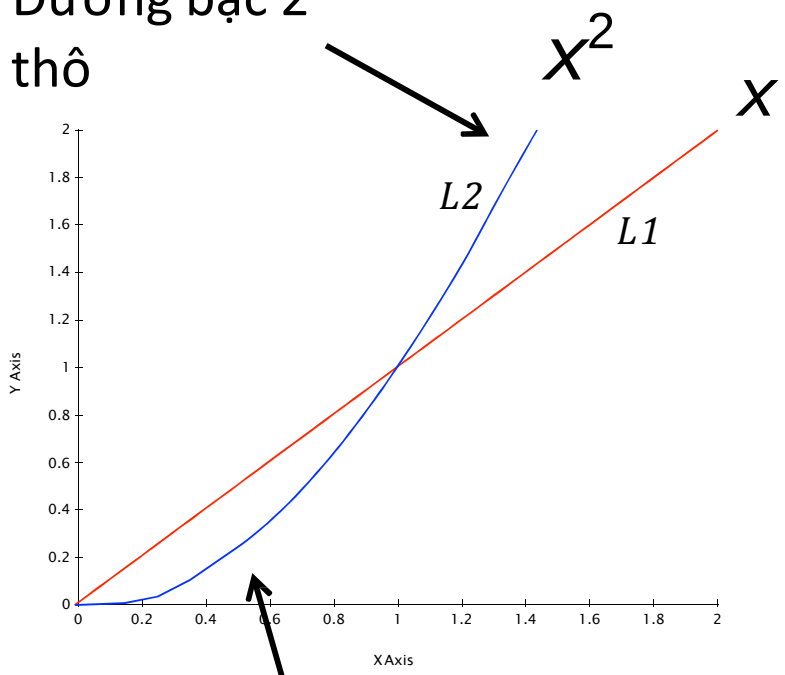
$$\sum_{samples} [Y - (\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2)]^2 + \lambda \cdot (|\beta_0| + |\beta_1| + |\beta_2|)$$

hoặc viết ở dạng khác,

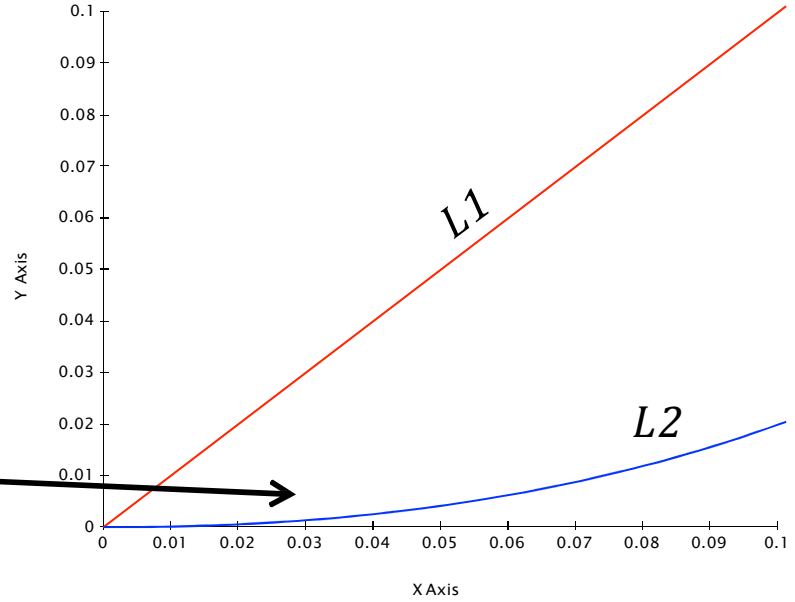
$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left(\|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right)$$

Phương thức phạt (Penalties)

Đường bậc 2 thô



Kiểu đường bậc 2.



Mục tiêu khác: Mô hình thưa

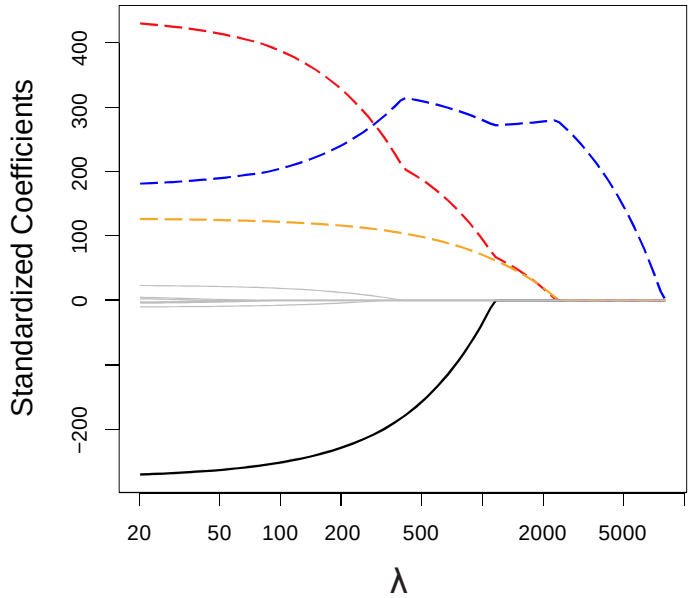
Lasso

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i-1}^{N} \left(y_i - \beta_0 - \sum_{j-1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j-1}^{p} |\beta_j| \right\} \equiv \hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \quad s.t. \sum_{j=1}^{p} |\beta_j| \leq t$$

Ridge

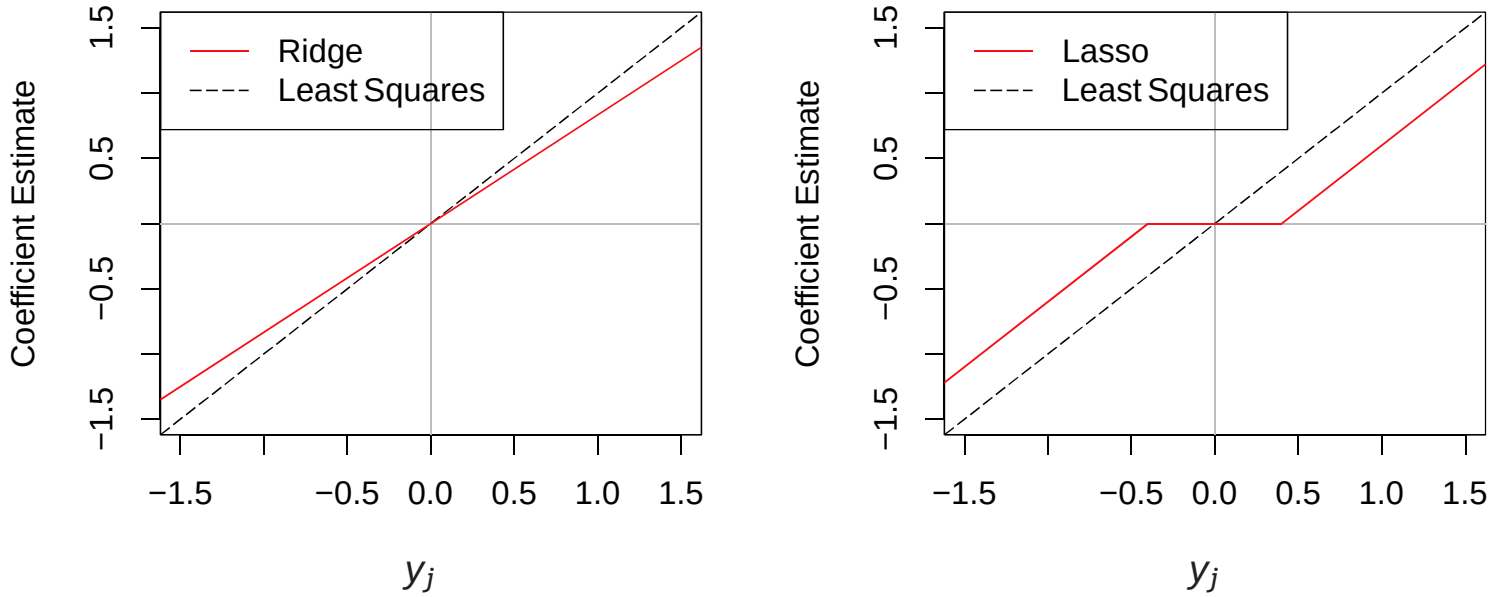
$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i-1}^{N} \left(y_i - \beta_0 - \sum_{j-1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j-1}^{p} \beta_j^2 \right\} \equiv \hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \quad s.t. \sum_{j=1}^{p} \beta_j^2 \leq t$$

L1 (lasso) tính nhanh hơn và thưa



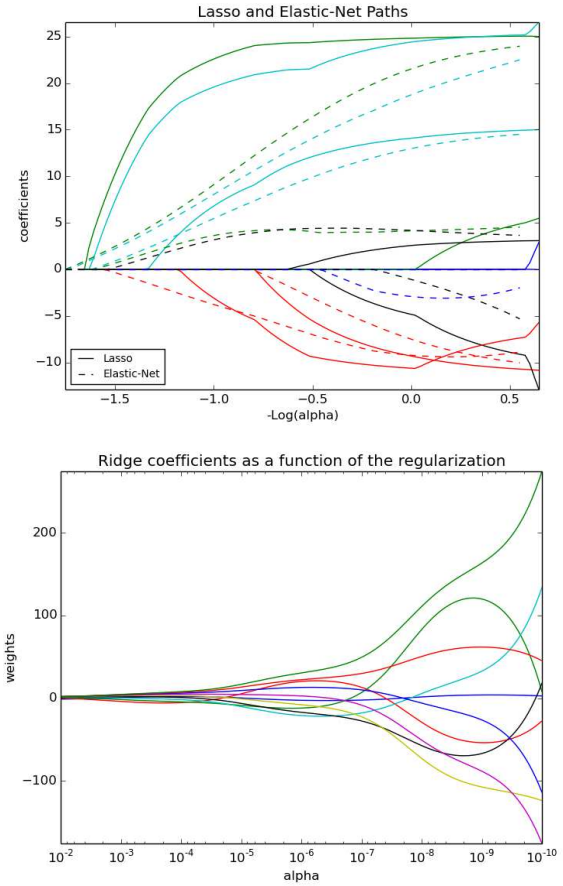
Hastie, Trevor, et al. Introduction to statistical learning.

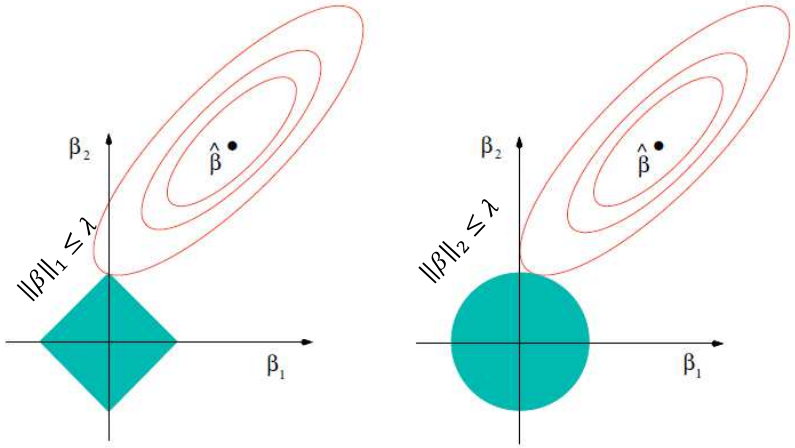
Ridge vs. Lasso: Mô hình thưa



Hastie, Trevor, et al. Introduction to statistical learning.

Ridge và LASSO





Hình ảnh về ước lượng của hồi quy lasso (trái) và ridge (phải). Vùng màu xanh và vùng ràng buộc $|\beta_1| + |\beta_2| \leq t$ và $\beta_1^2 + \beta_2^2 \leq t^2$, và các đường ellipse màu đỏ là đường viền của hàm residual-sum-of-squares. Điểm $\hat{\beta}$ biểu diễn ước lượng bình phương tối thiểu thông thường unconstrained).

Câu hỏi?