Nội dung môn học

- Lecture 1: Giới thiệu về Học máy và khai phá dữ liệu
- Lecture 2: Thu thập và tiền xử lý dữ liệu
- **Lecture 3: Hồi quy tuyến tính (Linear regression)**
- Lecture 4+5: Phân cụm
- Lecture 6: Phân loại và Đánh giá hiệu năng
- Lecture 7: dựa trên láng giềng gần nhất (KNN)
- Lecture 8: Cây quyết định và Rừng ngẫu nhiên
- Lecture 9: Học dựa trên xác suất
- Lecture 10: Mạng nơron (Neural networks)
- Lecture 11: Máy vector hỗ trợ (SVM)
- Lecture 12: Khai phá tập mục thường xuyên và các luật kết hợp
- Lecture 13: Thảo luận ứng dụng học máy và khai phá dữ liệu trong thực tế

Học có giám sát

- **Học có giám sát (Supervised learning)**
  - Tập dữ liệu học (*training data*) bao gồm các quan sát (*examples, observations*), mà mỗi quan sát được *gắn kèm với một giá trị đầu ra mong muốn.*
  - Mục đích là học một hàm (vd: một phân lớp, một hàm hồi quy,...) phù hợp với tập dữ liệu hiện có và khả năng tổng quát hoá cao.
  - Hàm học được sau đó sẽ được dùng để dự đoán cho các quan sát mới.
  - *Phân loại (classification):* nếu đầu ra (output – y) thuộc tập rời rạc và hữu hạn.
  - *Hồi quy (regression):* nếu đầu ra (output – y) là các số thực: Dự đoán đầu ra liên tục.

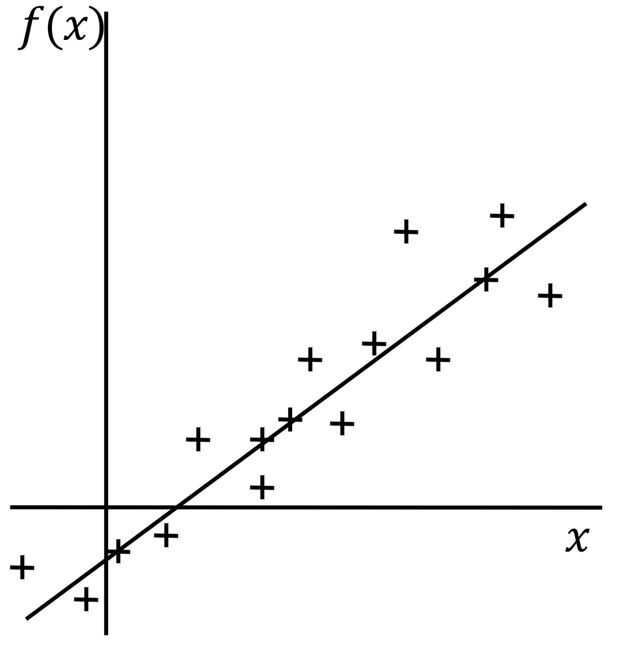
Hồi quy tuyến tính: Giới thiệu

- Bài toán hồi quy: cần học một hàm y = f(**x**) từ một tập học cho trước **D** = $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_M, y_M)\}$ trong đó $y_i \approx f(\mathbf{x}_i)$ với mọi i.
  - Mỗi quan sát được biểu diễn bằng một vécto *n* chiều, chẳng hạn $\mathbf{x}_i =$
  - $(x_{i1}, x_{i2}, \ldots, x_{in})^T$.
  - Mỗi chiều biểu diễn một thuộc tính (attribute/feature)
- Mô hình tuyến tính: nếu giả thuyết hàm y = f(**x**) là hàm có
  - dạng tuyến tính
    - $f(\mathbf{x}) = w_0 + w_1x_1 + \ldots + w_nx_n$
- Học một hàm hồi quy tuyến tính thì tương đương với việc học
  - véctơ trọng số $\mathbf{w} = (w_0, w_1, \ldots, w_n)^T$

Hồi quy tuyến tính: Ví dụ

Hàm tuyến tính f(**x**) nào phù hợp?

0.13	-0.91
1.02	-0.17
3.17	1.61
-2.76	-3.31
1.44	0.18
5.28	3.36
-1.74	-2.46
7.93	5.56
...	...



Ví dụ hàm f(x) có dạng: $f(x) = w_0 + w_1 x$;
HS tự tính ra $f(x) = -1.02 + 0.83x$
theo công thức sau

Phán đoán tương lai

- Đối với mỗi quan sát $\mathbf{x} = (x_1, x_2, ..., x_n)^T$:
  - Giá trị đầu ra mong muốn c_x
    (Không biết trước đối với các quan sát trong tương lai)
  - Giá trị phán đoán (bởi hệ thống)

$$y_x = w_0 + w_1x_1 + ... + w_nx_n$$

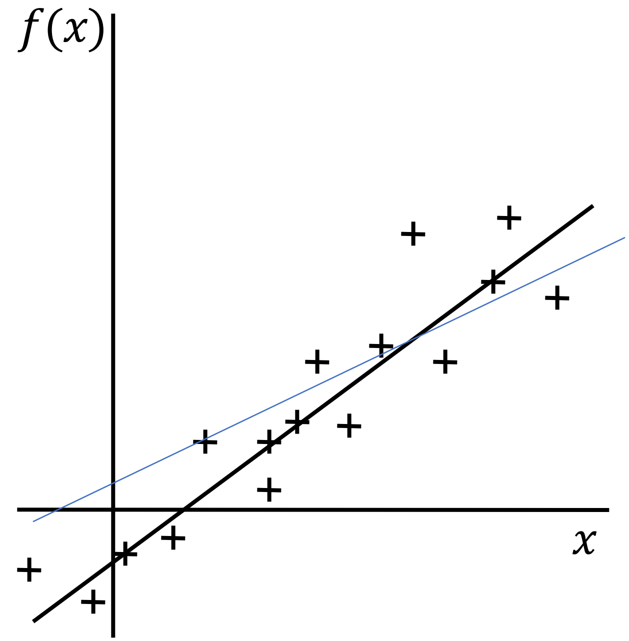
  - Ta thường mong muốn y_x xấp xỉ tốt c_x

- Phán đoán cho quan sát tương lai $\mathbf{z} = (z_1, z_2, ..., z_n)^T$
  - Cần dự đoán giá trị đầu ra, bằng cách áp dụng hàm mục tiêu đã học được `f`:

$$f(\mathbf{z}) = w_0 + w_1z_1 + ... + w_nz_n$$

Học hàm hồi quy

- Mục tiêu học: *học một hàm f* sao cho khả năng phán đoán trong tương lai là tốt nhất.*
  - Tức là sai số $|c_z - f(\mathbf{z})|$ là nhỏ nhất cho các quan sát tương lai **z**.
  - Khả năng tổng quát hóa (generalization) là tốt nhất.
- Vấn đề: Có vô hạn hàm tuyến tính!!
  - Làm sao để học? Quy tắc nào?



- Dùng một tiêu chuẩn để đánh giá.
  - Tiêu chuẩn thường dùng là hàm lỗi (generalization error, loss function, ...)

Hàm đánh giá lỗi (loss function)

- Định nghĩa hàm lỗi E
  - Lỗi (error/loss) phán đoán cho quan sát **x** = $(x_1, x_2, \ldots, x_n)^T$

$$r(\mathbf{x}) = [c_x - f^*(\mathbf{x})]^2 = (c_x - w_0 - w_1 x_1 - \ldots - w_n x_n)^2$$

- Lỗi của hệ thống trên toàn bộ không gian của **x**:

$$E = \mathbf{E}_x[r(\mathbf{x})] = \mathbf{E}_x[c_x - f^*(\mathbf{x})]^2$$

Cost, risk

- Mục tiêu học là tìm hàm f* mà E là nhỏ nhất: $f^* = \arg\min_{f \in \boldsymbol{H}} \boldsymbol{E}_x\left[r(\boldsymbol{x})\right]$
- Trong đó ***H*** là không gian của hàm *f*.

- Nhưng: trong quá trình học ta không thể làm việc được với bài toán này.

Hàm lỗi thực nghiệm

- Ta chỉ quan sát được một tập **D** = {($\mathbf{x}_1$, y_1), ($\mathbf{x}_2$, y_2), …, ($\mathbf{x}_M$, y_M)}. Cần học hàm f từ **D**.
- Lỗi thực nghiệm (empirical loss; residual sum of squares)

$$RSS(f) = \sum_{i=1}^{M} (y_i - f(\boldsymbol{x}_i))^2 = \sum_{i=1}^{M} (y_i - w_0 - w_1 x_{i1} - \cdots - w_n x_{in})^2$$

  - *RSS/M* là một xấp xỉ của $\mathbf{E}_x[r(\mathbf{x})]$ trên tập học ***D***

- $\left|\frac{1}{M} RSS(f) - \boldsymbol{E}_x[r(\boldsymbol{x})]\right|$ thường được gọi là **lỗi tổng quát hoá** (generalization error) của hàm f.
- Nhiều phương pháp học thường gắn với RSS.

Bình phương tối thiểu (OLS)

- Cho trước **D**, ta đi tìm hàm *f* mà có *RSS* nhỏ nhất.

$$f^* = \arg\min_{f \in \boldsymbol{H}} RSS(f)$$

$$\Leftrightarrow \boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum^{M} (y_i - w_0 - w_1 x_{i1} - \cdots - w_n x_{in})^2 \quad (1)$$

$$\min_w \sum_{i=1}^{M} (y_i - w^T x_i)^2$$

- Đây được gọi là bình phương tối thiểu (least squares).

- Tìm nghiệm **w*** bằng cách lấy đạo hàm của RSS và giải phương trình RSS' = 0. Thu được: $\boldsymbol{w}^* = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$

(cm công thức slide 10)

  - Trong đó **A** là ma trận dữ liệu cỡ Mx(n+1) mà hàng thứ i là $\mathbf{A}_i = (1, x_{i1}, x_{i2}, \ldots, x_{in})$; $\mathbf{B}^{-1}$ là ma trận nghịch đảo; $\mathbf{y} = (y_1, y_2, \ldots, y_M)^T$.
  - Chú ý: giả thuyết $\mathbf{A}^T\mathbf{A}$ tồn tại nghịch đảo.

Viết dưới dạng ma trận(mxn+1) (M mẫu, n thuộc tính)

$\min_w \sum (y_i - w^T x_i)^2$ Tương đương $\min_w \|Aw - y\|^2$

Gộp tất cả các dòng dữ liệu thành **ma trận A** (thêm 1 vào đầu mỗi dòng để tính bias w_0):

$$A = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{M1} & x_{M2} & \dots & x_{Mn} \end{bmatrix} \quad (M \times (n+1)) \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}$$

$\sum_{i=1}^{M} (y_i - w^T x_i)^2$ Tương đương $L(w) = \|A$ $\min_w \sum (y_i - w^T x_i)^2$ $^T(Aw - y)$

Lấy đạo hàm gradient theo w:

$$\nabla_w L(w) = 2A^T(Aw - y)$$

Giải phương trình đạo hàm bằng 0 (điều kiện cực trị):

$$A^T A w = A^T y$$

$$w^* = (A^T A)^{-1} A^T y$$

Bình phương tối thiểu: thuật toán

- Input: **D** = {($\mathbf{x}_1$, y_1), ($\mathbf{x}_2$, y_2), …, ($\mathbf{x}_M$, y_M)}
- Output: **w***

- Học **w*** bằng cách tính:

$$\boldsymbol{w}^* = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$$

  - Trong đó **A** là ma trận dữ liệu cỡ Mx(n+1) mà hàng thứ i là $\mathbf{A}_i$ = (1, x_{i1}, x_{i2}, …, x_{in}); $\mathbf{B}^{-1}$ là ma trận nghịch đảo; **y** = $(y_1, y_2, …, y_M)^T$.
  - Chú ý: giả thuyết $\mathbf{A}^T\mathbf{A}$ tồn tại nghịch đảo.

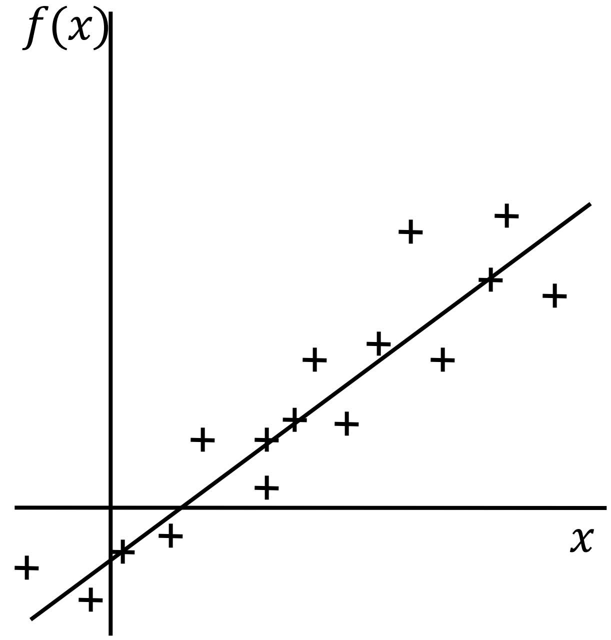
- Phán đoán cho quan sát mới **x**:

$$y_x = w_0^* + w_1^* x_1 + \cdots + w_n^* x_n$$

Hồi quy tuyến tính: Ví dụ

Hàm tuyến tính f(**x**) nào phù hợp?

0.13	-0.91
1.02	-0.17
3.17	1.61
-2.76	-3.31
1.44	0.18
5.28	3.36
-1.74	-2.46
7.93	5.56
...	...



Ví dụ hàm f(x) có dạng: $f(x) = w_0 + w_1 x$;

HS tự tính ra $f(x) = -1.02 + 0.83x$

theo công thức sau

$$\mathbf{A} = \begin{pmatrix} 1 & 0.13 \\ 1 & 1.02 \\ 1 & 3.17 \\ 1 & -2.76 \\ 1 & 1.44 \\ 1 & 5.28 \\ 1 & -1.74 \\ 1 & 7.93 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} -0.91 \\ -0.17 \\ 1.61 \\ -3.31 \\ 0.18 \\ 3.36 \\ -2.46 \\ 5.56 \end{pmatrix}$$

$$\mathbf{A}^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0.13 & 1.02 & 3.17 & -2.76 & 1.44 & 5.28 & -1.74 & 7.93 \end{pmatrix}$$

$$\mathbf{A}^T\mathbf{A} = \begin{pmatrix} 8 & 14.47 \\ 14.47 & 114.5883 \end{pmatrix}$$

Đối với một ma trận 2x2 $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, ma trận nghịch đảo được tính bằng công thức: $\frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$.

Với $\mathbf{A}^T\mathbf{A} = \begin{pmatrix} 8 & 14.47 \\ 14.47 & 114.5883 \end{pmatrix}$:

$$(\mathbf{A}^T\mathbf{A})^{-1} = \frac{1}{707.3255}\begin{pmatrix} 114.5883 & -14.47 \\ -14.47 & 8 \end{pmatrix}$$

Vậy, ma trận $(\mathbf{A}^T\mathbf{A})^{-1}$ là:

$$(\mathbf{A}^T\mathbf{A})^{-1} = \begin{pmatrix} 0.16200222 & -0.02045734 \\ -0.02045734 & 0.01131021 \end{pmatrix}$$

$$\mathbf{A}^T\mathbf{y} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0.13 & 1.02 & 3.17 & -2.76 & 1.44 & 5.28 & -1.74 & 7.93 \end{pmatrix} \begin{pmatrix} -0.91 \\ -0.17 \\ 1.61 \\ -3.31 \\ 0.18 \\ 3.36 \\ -2.46 \\ 5.56 \end{pmatrix}$$

$$\mathbf{A}^T\mathbf{y} = \begin{pmatrix} 3.86 \\ 80.3168 \end{pmatrix}$$

$$\mathbf{w}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$$

$$\mathbf{w}^* = \begin{pmatrix} 0.16200222 & -0.02045734 \\ -0.02045734 & 0.01131021 \end{pmatrix} \begin{pmatrix} 3.86 \\ 80.3168 \end{pmatrix}$$

$$\mathbf{w}^* = \begin{pmatrix} w_0^* \\ w_1^* \end{pmatrix} \approx \begin{pmatrix} -1.0167 \\ 0.8295 \end{pmatrix}$$

f(x)=−1.0167+0.8295x

Bình phương tối thiểu: nhược điểm

- Nếu $\mathbf{A}^T\mathbf{A}$ không tồn tại nghịch đảo thì không học được.
  - Nếu các thuộc tính (cột của A) có phụ thuộc lẫn nhau.
- Độ phức tạp tính toán lớn do phải tính ma trận nghịch đảo. →Không làm việc được nếu số chiều n lớn.
- Khả năng overfitting cao vì việc học hàm f chỉ quan tâm tối thiểu lỗi đối với tập học đang có.

Ridge regression

- Hồi quy Ridge là một phương pháp regular hóa (regularization) mạnh mẽ cho hồi quy tuyến tính, giúp cải thiện độ ổn định và hiệu suất của mô hình khi đối mặt với đa cộng tuyến và overfitting bằng cách thêm một hình phạt dựa trên bình phương của các hệ số.

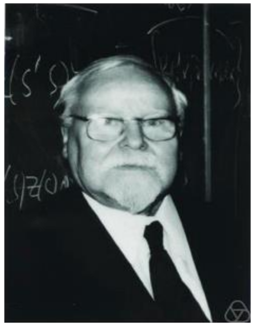
Ridge regression (1):

- Cho trước **D** = {($\mathbf{x}_1$, y_1), ($\mathbf{x}_2$, y_2), ..., ($\mathbf{x}_M$, y_M)}, ta đi giải bài toán:

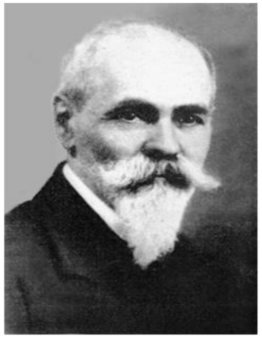
$$\boldsymbol{f}^* = \arg\min_{f \in \boldsymbol{H}} RSS(f) + \lambda \|\boldsymbol{w}\|_2^2$$

$$\Leftrightarrow \boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{M} (y_i - \boldsymbol{A_i w})^2 + \lambda \sum_{j=0}^{n} w_j^2 \qquad (2)$$

Trong đó $\mathbf{A}_i$ = (1, x_{i1}, x_{i2}, ..., x_{in}) được tạo ra từ $\mathbf{x}_i$; λ là một hằng số phạt (λ> 0).



Tikhonov, smoothing an ill-posed problem



Zaremba, model complexity minimization

Bayes: priors over parameters

Andrew Ng: need no maths, but it prevents overfitting!

Ridge regression (2)

- Giải bài toán (2) tương đương với việc giải bài toán sau:

$$w^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{M} (y_i - \boldsymbol{A}_i \boldsymbol{w})^2 \quad (3)$$

$$\text{sao cho } \sum_{j=0}^{n} w_j^2 \leq t$$

  - t là một hằng số nào đó.

- Đại lượng hiệu chỉnh (phạt) $\lambda \|\boldsymbol{w}\|_2^2$
  - Có vai trò hạn chế độ lớn của **w*** (hạn chế không gian hàm f).
  - Đánh đổi chất lượng của hàm f đối với tập học **D**, để có khả năng phán đoán tốt hơn với quan sát tương lai.

Ridge regression (3)

- Tìm nghiệm **w*** bằng cách lấy đạo hàm của RSS và giải phương trình RSS’ = 0. Thu được:

$$\boldsymbol{w}^* = (\boldsymbol{A}^T\boldsymbol{A} + \lambda \boldsymbol{I}_{n+1})^{-1}\boldsymbol{A}^T\boldsymbol{y}$$

  - Trong đó **A** là ma trận dữ liệu cỡ Mx(n+1) mà hàng thứ i là $(1, x_{i1}, x_{i2}, \ldots, x_{in})$; $\mathbf{y} = (y_1, y_2, \ldots, y_M)^T$; $\mathbf{I}_{n+1}$ là ma trận đơn vị cỡ n+1.
- So sánh với phương pháp bình phương tối thiểu:
  - Tránh được trường hợp ma trận dữ liệu suy biến. Hồi quy Ridge luôn làm việc được.
  - Khả năng overfitting thường ít hơn.
  - Lỗi trên tập học có thể nhiều hơn.
- *Chú ý: chất lượng của phương pháp phụ thuộc rất nhiều vào sự lựa chọn của tham số λ.*

Ridge regression: thuật toán

- Input: **D** = {($\mathbf{x}_1$, y_1), ($\mathbf{x}_2$, y_2), …, ($\mathbf{x}_M$, y_M)}, hằng số λ>0
- Output: **w***
- Học **w*** bằng cách tính:

$$\boldsymbol{w}^* = (\boldsymbol{A}^T\boldsymbol{A} + \lambda \boldsymbol{I}_{n+1})^{-1}\boldsymbol{A}^T\boldsymbol{y}$$

  - Trong đó **A** là ma trận dữ liệu cỡ Mx(n+1) mà hàng thứ i là $\mathbf{A}_i$ = (1, x_{i1}, x_{i2}, …, x_{in}); $\mathbf{B}^{-1}$ là ma trận nghịch đảo; **y** = $(y_1, y_2, \ldots, y_M)^T$.

- Phán đoán cho quan sát mới **x**:

$$y_x = w_0^* + w_1^* x_1 + \cdots + w_n^* x_n$$

- ***Chú ý:*** để tránh vài ảnh hưởng xấu từ độ lớn của y, ta nên loại bỏ thành phần w_0 trong đại lượng phạt ở công thức (2). Khi đó nghiệm **w*** sẽ thay đổi một chút.

VD

$$\mathbf{w}^* = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_{n+1})^{-1}\mathbf{A}^T\mathbf{y}$$

Trong đó, n là số lượng đặc trưng (feature). Trong trường hợp này, chúng ta chỉ có một đặc trưng x, nên $n = 1$. Do đó, ma trận đơn vị $\mathbf{I}_{n+1}$ sẽ là $\mathbf{I}_2$ (ma trận đơn vị cấp 2x2).

$$\mathbf{A}^T\mathbf{A} = \begin{pmatrix} 8 & 14.47 \\ 14.47 & 114.5883 \end{pmatrix} \qquad \mathbf{A}^T\mathbf{y} = \begin{pmatrix} 3.86 \\ 80.3168 \end{pmatrix}$$

Chúng ta đã chọn $\lambda = 0.1$. Ma trận đơn vị $\mathbf{I}_2$ là $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

Vậy, $\lambda\mathbf{I}_2 = 0.1 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$.

Bây giờ, cộng ma trận này vào $\mathbf{A}^T\mathbf{A}$:

$$\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_2 = \begin{pmatrix} 8 & 14.47 \\ 14.47 & 114.5883 \end{pmatrix} + \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$$

Thực hiện phép cộng từng phần tử:

$$\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_2 = \begin{pmatrix} 8 + 0.1 & 14.47 + 0 \\ 14.47 + 0 & 114.5883 + 0.1 \end{pmatrix} = \begin{pmatrix} 8.1 & 14.47 \\ 14.47 & 114.6883 \end{pmatrix}$$

$$(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_2)^{-1} = \frac{1}{719.59433} \begin{pmatrix} 114.6883 & -14.47 \\ -14.47 & 8.1 \end{pmatrix} \qquad (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_2)^{-1} \approx \begin{pmatrix} 0.1593895 & -0.0201088 \\ -0.0201088 & 0.0112563 \end{pmatrix}$$

Tính toán $\mathbf{w}^* = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_2)^{-1}\mathbf{A}^T\mathbf{y}$

$$\mathbf{w}^* \approx \begin{pmatrix} 0.1593895 & -0.0201088 \\ -0.0201088 & 0.0112563 \end{pmatrix} \begin{pmatrix} 3.86 \\ 80.3168 \end{pmatrix} \qquad \mathbf{w}^* = \begin{pmatrix} w_0^* \\ w_1^* \end{pmatrix} \approx \begin{pmatrix} -1.0002 \\ 0.8262 \end{pmatrix}$$

Hàm tuyến tính phù hợp nhất với dữ liệu quan sát theo Hồi quy Ridge là: f(x)=−1.0002+0.8262x

So sánh với Hồi quy Tuyến tính thông thường (OLS):

- OLS: f(x)=−1.0167+0.8295x
- Ridge (λ=0.1): f(x)=−1.0002+0.8262x

Bạn có thể thấy rằng các hệ số có một chút thay đổi (co rút nhẹ) so với OLS. Giá trị w0 từ -1.0167 co rút lên -1.0002 (tiến gần hơn về 0). Giá trị w1 từ 0.8295 co rút xuống 0.8262 (cũng tiến gần hơn về 0). Mức độ co rút phụ thuộc vào giá trị của λ. Nếu λ lớn hơn, các hệ số sẽ co rút về 0 mạnh hơn nữa.

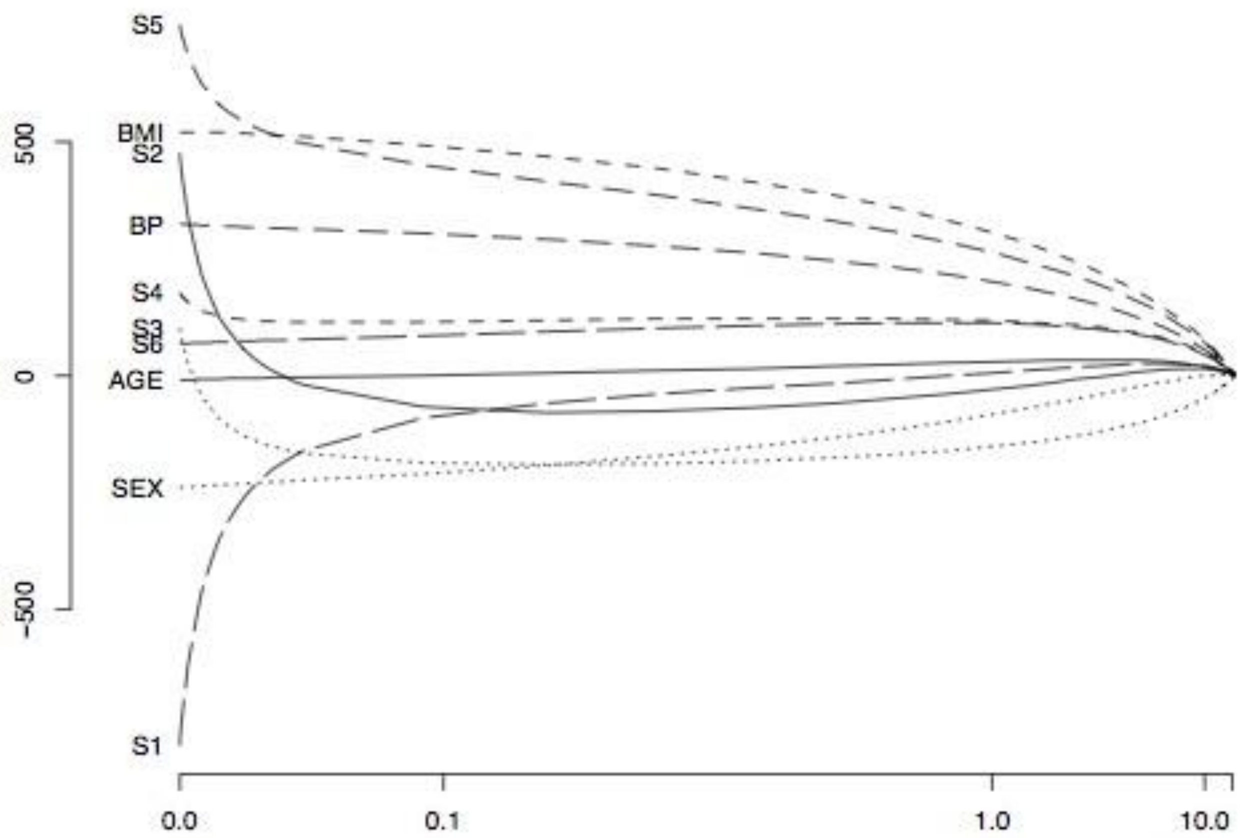
Ridge regression: ví dụ

- Xét tập dữ liệu Prostate gồm 67 quan sát dùng để học, và 31 quan sát dùng để kiểm thử. Dữ liệu gồm 8 thuộc tính.

w	Least squares	Ridge
0	2.465	2.452
lcavol	0.680	0.420
lweight	0.263	0.238
age	−0.141	−0.046
lbph	0.210	0.162
svi	0.305	0.227
lcp	−0.288	0.000
gleason	−0.021	0.040
pgg45	0.267	0.133
Test RSS	**0.521**	**0.492**

Ridge regression: ảnh hưởng của λ

- **W*** = (w_0, S1, S2, S3, S4, S5, S6, AGE, SEX, BMI, BP) thay đổi khi cho λ thay đổi.



LASSO

- Hồi quy Ridge sử dụng chuẩn L² cho đại lượng hiệu chỉnh:

  $w^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{M} (y_i - \boldsymbol{A}_i \boldsymbol{w})^2$, sao cho $\sum_{j=0}^{n} w_j^2 \leq t$

- Thay L² bằng L¹ thì ta sẽ thu được phương pháp LASSO:

$$w^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{M} (y_i - \boldsymbol{A}_i \boldsymbol{w})^2$$

sao cho $\sum_{j=0}^{n} |w_j| \leq t$

- Hoặc có thể viết lại:

$$w^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{M} (y_i - \boldsymbol{A}_i \boldsymbol{w})^2 + \lambda \|\boldsymbol{w}\|_1$$

- Hàm mục tiêu của bài toán là không trơn. Do đó việc giải nó có thể khó hơn hồi quy Ridge.

LASSO: đại lượng hiệu chỉnh

- Các kiểu hiệu chỉnh khác nhau sẽ tạo ra các miền khác nhau cho **w**.
- LASSO thường tạo ra nghiệm thưa, tức là nhiều thành phần của **w** có giá trị là 0.
  - Vì thế LASSO thực hiện đồng thời việc hạn chế và lựa chọn đặc trưng

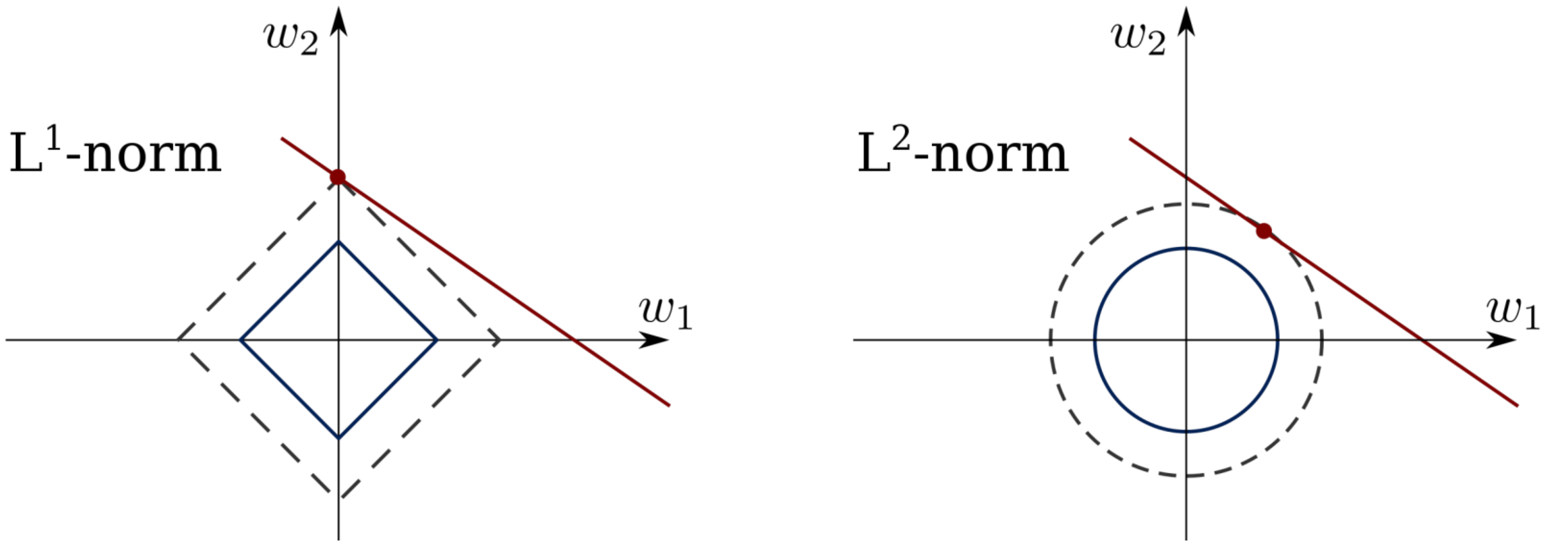


Figure by Nicoguaro - Own work, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=58258966

OLS, Ridge, LASSO

- Xét tập dữ liệu Prostate gồm 67 quan sát dùng để học, và 31 quan sát dùng để kiểm thử. Dữ liệu gồm 8 thuộc tính.

w	Ordinary Least Squares	Ridge	LASSO
0	2.465	2.452	2.468
lcavol	0.680	0.420	0.533
lweight	0.263	0.238	0.169
age	−0.141	−0.046	
lbph	0.210	0.162	0.002
svi	0.305	0.227	0.094
lcp	−0.288	0.000	
gleason	−0.021	0.040	
pgg45	0.267	0.133	
Test RSS	**0.521**	**0.492**	**0.479**

Một số trọng số là 0
→ Chúng có thể không quan trọng

Sai số: Tổng bình phương sai số (Sum of Squared Errors - SSE)
Sai số bình phương trung bình (Mean Squared Error - MSE): =SSE/n
Căn bậc hai của sai số bình phương trung bình (Root Mean Squared Error - RMSE

x (Giá trị đầu vào)	y (Giá trị thực tế)	ŷ (Giá trị dự đoán)	Sai số (y - ŷ)	Sai số bình phương $(y - \hat{y})^2$
0.13	-1	-0.67	-0.33	0.1089
1.02	-0.17	0.05	-0.22	0.0484
3	1.61	1.65	-0.04	0.0016
-2.5	-2	-2.81	0.81	0.6561
1.44	0.1	0.39	-0.29	0.0841
5	3.36	3.27	0.09	0.0081

SSE = 1.0477