ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Học máy cơ bản

Hồi quy tuyến tính

Thân Quang Khoát

Nội dung môn học

- Buổi 1: Giới thiệu về Học máy
- Buổi 2: Quy trình xây dựng hệ thống học máy
- **Buổi 3: Hồi quy tuyến tính**
- Buổi 4: Học dựa trên láng giềng gần nhất (KNN)
- Buổi 5: Cây quyết định và Rừng ngẫu nhiên
- Buổi 6: Naïve Bayes
- Buổi 7: Máy vector hỗ trợ (SVM)
- Buổi 8: Đánh giá hiệu quả của mô hình học máy
- Buổi 9: Phân cụm
- Buổi 10-11: Kiểm tra giữa kỳ và trình bày ý tưởng làm dự án cuối kỳ
- Buổi 12-20: Học sâu

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.

Hồi quy tuyến tính: Giới thiệu

- Bài toán hồi quy: cần học một hàm $y = f(\boldsymbol{x})$ từ một tập học cho trước $\boldsymbol{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_M, y_M)\}$ sao cho $y_i \approx f(\boldsymbol{x}_i)$ với mọi i.
  - Mỗi quan sát được biểu diễn bằng một véctơ *n* chiều, chẳng hạn $\boldsymbol{x}_i = (x_{i1}, \dots, x_{in})^T$.
  - Mỗi chiều biểu diễn một thuộc tính (attribute/feature)
- Mô hình tuyến tính (linear model):
nếu giả thuyết hàm $y = f(\boldsymbol{x})$ là hàm có dạng tuyến tính

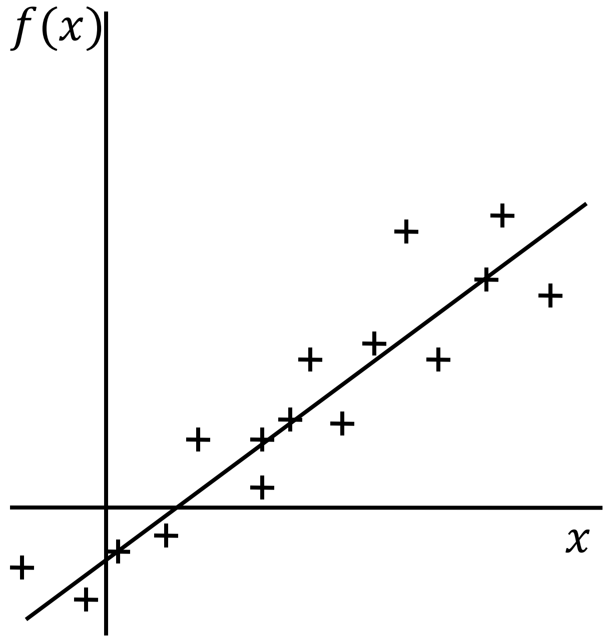
$$f(\boldsymbol{x}) = w_0 + w_1 x_1 + \dots + w_n x_n$$

  - w_0 hay được gọi là độ lệch (bias)
- Học một hàm hồi quy tuyến tính thì tương đương với việc học véctơ trọng số $\boldsymbol{w} = (w_0, w_1, \dots, w_n)^T$

Hồi quy tuyến tính: Ví dụ

Hàm tuyến tính f(**x**) nào phù hợp?

x	y
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: $f(x) = -1.02 + 0.83x$

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)

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

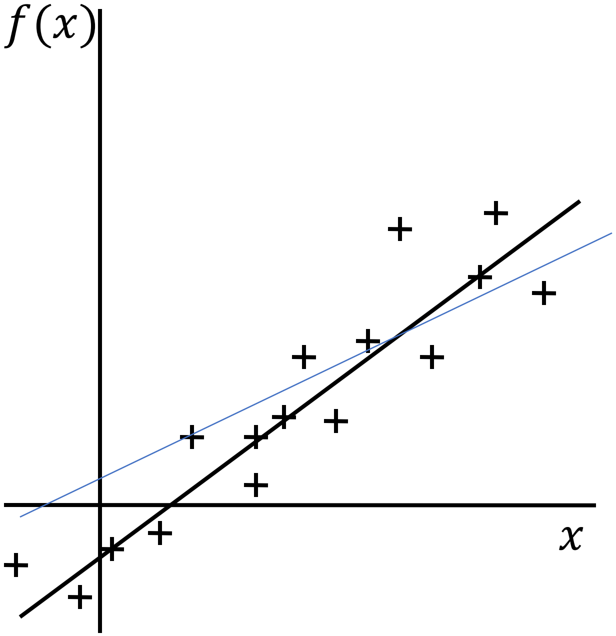
  - Ta thường mong muốn y_x xấp xỉ tốt c_x
- Phán đoán cho quan sát tương lai $\mathbf{z} = (z_1, z_2, ..., z_n)^T$
  - Cần dự đoán giá trị đầu ra, bằng cách áp dụng hàm mục tiêu đã học được `f`:

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

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!!
  $\boldsymbol{H} = \{ f(x, w): \boldsymbol{w} = (w_0, w_1, \dots, w_n) \in \mathbb{R}^{n+1} \}$
  - Làm sao để học? Quy tắc nào?



- Dùng một tiêu chuẩn để đánh giá.
  - Tiêu chuẩn thường dùng là hàm lỗi (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 trung bình* (Expected loss/risk) 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.

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

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

- 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:

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

  - 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; $\mathbf{y} = (y_1, y_2, ..., y_M)^T$.
  - Chú ý: giả thuyết $\mathbf{A}^T\mathbf{A}$ tồn tại nghịch đảo.

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** = $(y_1, y_2, \ldots, y_M)^T$.
  - Chú ý: giả thuyết $\mathbf{A}^T\mathbf{A}$ tồn tại nghịch đảo.

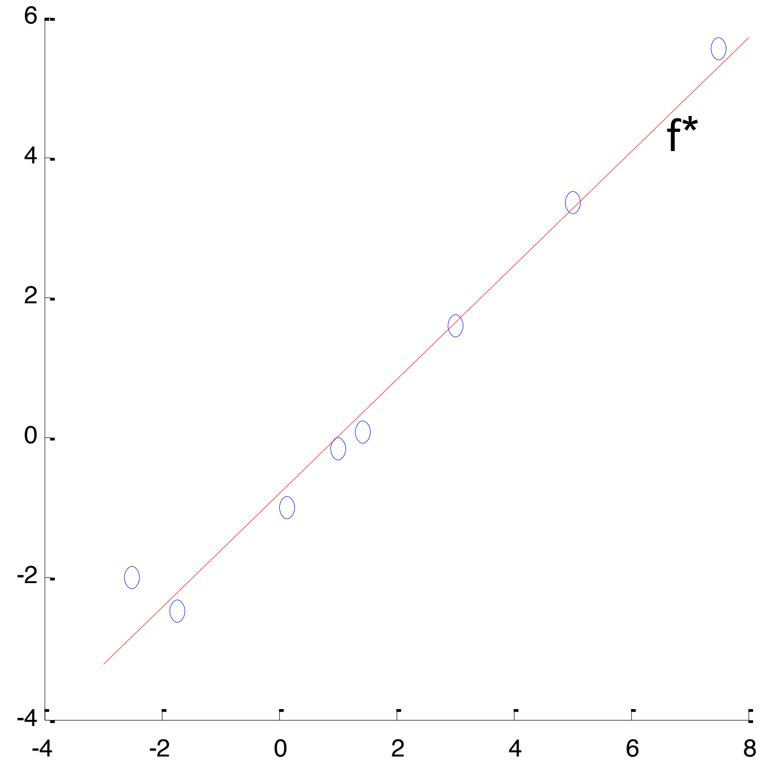
- Phán đoán cho quan sát mới **x**:

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

Bình phương tối thiểu: ví dụ

Kết quả học bằng bình phương tối thiểu

x	y
0.13	-1
1.02	-0.17
3	1.61
-2.5	-2
1.44	0.1
5	3.36
-1.74	-2.46
7.5	5.56



$f^*(x) = 0.81x - 0.78$

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.

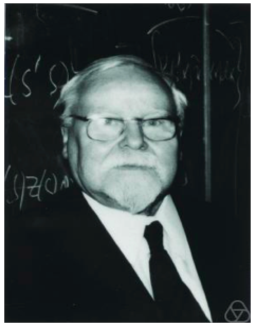
Ridge regression (1)

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

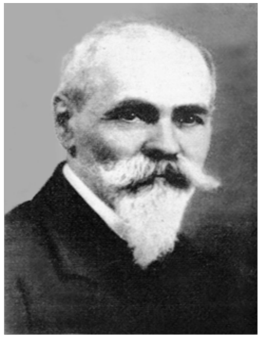
$$f^* = \arg\min_{f \in \boldsymbol{H}} RSS(f) + \lambda \|\boldsymbol{w}\|_2^2$$

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

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:

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

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

  - 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 $(1, x_{i1}, x_{i2}, \ldots, x_{in})$; **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 $\lambda>0$
- 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 $\mathbf{A}_i = (1, x_{i1}, x_{i2}, \ldots, x_{in})$; $\mathbf{B}^{-1}$ là ma trận nghịch đảo; $\mathbf{y} = (y_1, y_2, \ldots, y_M)^T$.
- 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.

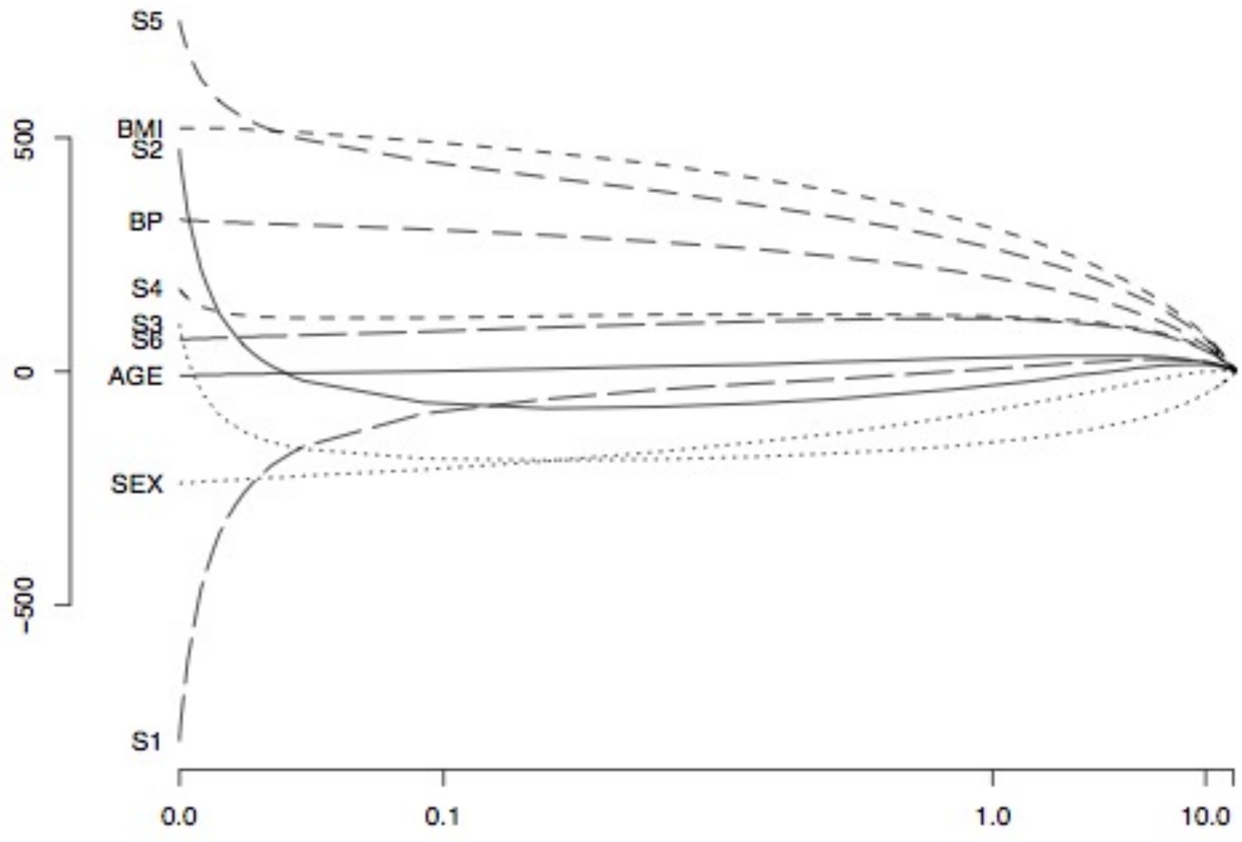
Ridge regression: ví dụ

- 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 L^2 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 L^2 bằng L^1 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

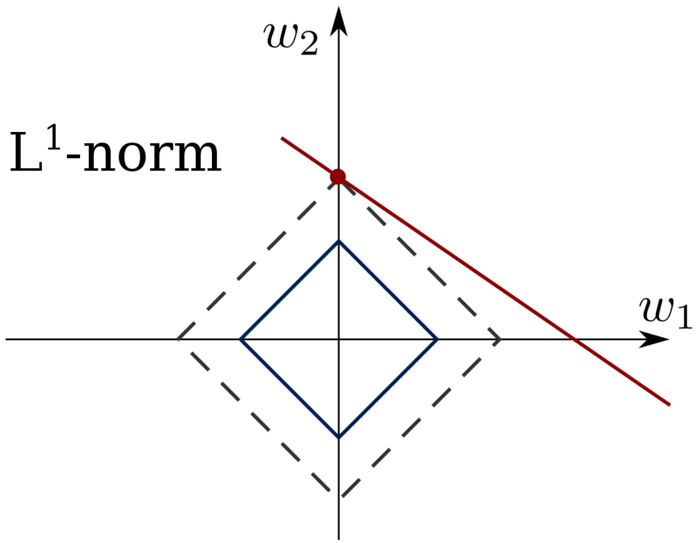


Figure by Nicoguaro - Own work, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=58258966

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

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict