Nhập môn Học máy và Khai phá dữ liệu

(IT3190)

Nguyễn Nhật Quang

quang.nguyennhat@hust.edu.vn

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ thông tin và truyền thông

Năm học 2021-2022

Nội dung môn học:

- Giới thiệu về Học máy và Khai phá dữ liệu
- Tiền xử lý dữ liệu
- Đánh giá hiệu năng của hệ thống
- **Hồi quy**
  - **Bài toán hồi quy**
  - **Hồi quy tuyến tính (Linear regression)**
- Phân lớp
- Phân cụm
- Phát hiện luật kết hợp

Bài toán hồi quy

- Hồi quy (regression) thuộc nhóm bài toán học có giám sát (supervised learning)
- Mục tiêu của bài toán hồi quy là dự đoán một vector các giá trị liên tục (số thực)

$$f: X \rightarrow Y$$

trong đó *Y* là một vector các giá trị số thực

Bài toán hồi quy: Đánh giá hiệu năng

- Giá trị (kết quả) đầu ra của hệ thống là một giá trị số
- Hàm đánh giá lỗi
  - MAE (mean absolute error):

$$MAE - Error(x) = \frac{\sum_{i=1}^{n} |d(x) - o(x)|}{n}$$

  - RMSE (root mean squared error):

$$RMSE - Error(x) = \sqrt{\frac{\sum_{i=1}^{n} \left(d(x) - o(x)\right)^2}{n}}$$

  - Lỗi tổng thể trên toàn bộ tập thử nghiệm:

$$Error = \frac{1}{|D\_test|} \sum_{x \in D\_test} Error(x);$$

  - n: Số lượng các đầu ra (outputs)
  - $o(x)$: Vector các giá trị đầu ra dự đoán bởi hệ thống đối với ví dụ x
  - $d(x)$: Vector các giá trị đầu ra thực sự (đúng/mong muốn) đối với ví dụ x
- Độ chính xác (*Accuracy*) là một hàm đảo (inverse function) đối với hàm lỗi (*Error*)

Hồi quy tuyến tính – Giới thiệu

- Một phương pháp học máy đơn-giản-nhưng-hiệu-quả phù hợp khi hàm mục tiêu (cần học) là một hàm tuyến tính

$$f(x) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n = w_0 + \sum_{i=1}^{n} w_i x_i \qquad (w_i, x_i \in R)$$

- Cần học (xấp xỉ) một hàm mục tiêu `f`

$$f: X \rightarrow Y$$

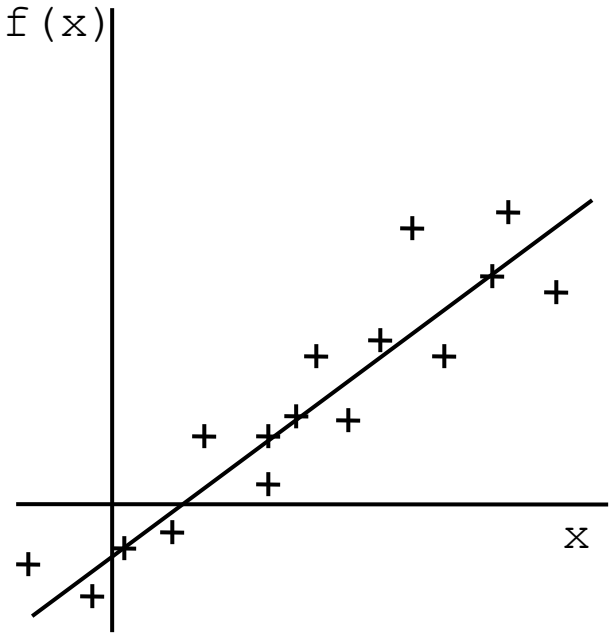
  - `X`: Miền không gian đầu vào (không gian vectơ `n` chiều – R^n)
  - `Y`: Miền không gian đầu ra (miền các giá trị số thực – `R`)
  - `f`: Hàm mục tiêu cần học (một hàm ánh xạ tuyến tính)

- Thực chất, là học một vectơ các trọng số: $w = (w_0, w_1, w_2, \ldots, w_n)$

Hồi quy tuyến tính – Ví dụ

Hàm tuyến tính `f(x)` nào phù hợp?

x	f(x)
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`

Các ví dụ học/kiểm thử

- Đối với mỗi ví dụ học $x=(x_1,x_2,...,x_n)$, trong đó $x_i \in R$
  - Giá trị đầu ra mong muốn c_x $(\in R)$
  - Giá trị đầu ra thực tế (tính bởi hệ thống) $y_x = w_0 + \sum_{i=1}^{n} w_i x_i$

  → w_i là đánh giá hiện thời của hệ thống đối với giá trị trọng số của thuộc tính thứ *i*

  → Giá trị đầu ra thực tế y_x được mong muốn (xấp xỉ) bằng c_x

- Đối với mỗi ví dụ kiểm thử $z=(z_1,z_2,...,z_n)$
  - Cần dự đoán (tính) giá trị đầu ra
  - Bằng cách áp dụng hàm mục tiêu đã học được f

Hàm đánh giá lỗi

- Giải thuật học hồi quy tuyến tính cần phải xác định Hàm đánh giá lỗi (Error function)
  - → Đánh giá mức độ lỗi của hệ thống trong giai đoạn huấn luyện
  - → Còn được gọi là Hàm mất mát (Loss function)
- Định nghĩa hàm lỗi E
  - Lỗi của hệ thống đối với mỗi ví dụ học x:

$$E(x) = \frac{1}{2}(c_x - y_x)^2 = \frac{1}{2}\left(c_x - w_0 - \sum_{i=1}^{n} w_i x_i\right)^2$$

  - Lỗi của hệ thống đối với toàn bộ tập huấn luyện D:

$$E = \sum_{x \in D} E(x) = \frac{1}{2}\sum_{x \in D}(c_x - y_x)^2 = \frac{1}{2}\sum_{x \in D}\left(c_x - w_0 - \sum_{i=1}^{n} w_i x_i\right)^2$$

Hồi quy tuyến tính – Giải thuật

- Việc học hàm mục tiêu `f` là tương đương với việc học vectơ trọng số **`w`** sao cho cực tiểu hóa giá trị lỗi huấn luyện `E`

  → Phương pháp này có tên gọi là "*Least-Square Linear Regression*"

- Giai đoạn huấn luyện
  - Khởi tạo vectơ trọng số **`w`**
  - Tính toán giá trị lỗi huấn luyện `E`
  - Cập nhật vectơ trọng số **`w`** theo **quy tắc delta (delta rule)**
  - Lặp lại, cho đến khi hội tụ về một giá trị lỗi nhỏ nhất (cục bộ) `E`

- Giai đoạn dự đoán

  Đối với một ví dụ mới `z`, giá trị đầu ra được dự đoán bằng:

$$f(z) = w^*_0 + \sum_{i=1}^{n} w^*_i \, z_i$$

Trong đó $w^* = (w^*_0, w^*_1, \ldots, w^*_n)$ là vectơ trọng số đã học được

Quy tắc delta

- Để cập nhật vectơ trọng số **w** theo hướng giúp giảm bớt giá trị lỗi huấn luyện E
  - η là tốc độ học (là một hằng số dương)

    → Xác định mức độ thay đổi đối với các giá trị trọng số tại mỗi bước học
  - Cập nhật theo từng ví dụ (Instance-to-instance/incremental update):
    $w_i \leftarrow w_i + \eta(c_x - y_x)x_i$
  - Cập nhật theo đợt/lô (Batch update): $w_i \leftarrow w_i + \eta \sum_{x \in D} (c_x - y_x) x_i$
- Các tên gọi khác của quy tắc delta
  - LMS (least mean square) rule
  - Adaline rule
  - Widrow-Hoff rule

Cập nhật theo đợt/theo từng ví dụ

- Cập nhật theo đợt/lô (Batch update)
  - Tại mỗi bước học, các giá trị trọng số được cập nhật sau khi ***tất cả*** *các ví dụ học của lô (batch) hiện tại được học bởi hệ thống*
    - Giá trị lỗi được tính tích lũy đối với tất cả các ví dụ học của lô hiện tại
    - Các giá trị trọng số được cập nhật theo giá trị lỗi tích lũy tổng thể của lô hiện tại
- Cập nhật theo từng ví dụ (Instance-to-instance/incremental update)
  - Tại mỗi bước học, các giá trị trọng số được cập nhật *ngay lập tức sau khi* ***mỗi*** *ví dụ học được học bởi hệ thống*
    - Giá trị lỗi (riêng biệt) được tính cho ví dụ học đưa vào
    - Các giá trị trọng số được cập nhật ngay lập tức theo giá trị lỗi này

LSLR_batch(D, η)

for each thuộc tính f_i

$w_i \leftarrow$ giá trị (nhỏ) được khởi tạo ngẫu nhiên

while not CONVERGENCE

for each thuộc tính f_i

$delta\_w_i \leftarrow 0$

for each ví dụ học $x \in D$

Tính toán giá trị đầu ra thực tế y_x

for each thuộc tính f_i

$delta\_w_i \leftarrow delta\_w_i + \eta(c_x - y_x)x_i$

for each thuộc tính f_i

$w_i \leftarrow w_i + delta\_w_i$

end while

return w

LSLR_incremental(D, η)

for each thuộc tính f_i

$w_i \leftarrow$ giá trị (nhỏ) được khởi tạo ngẫu nhiên

while not CONVERGENCE

for each ví dụ học $x \in D$

Tính toán giá trị đầu ra thực tế y_x

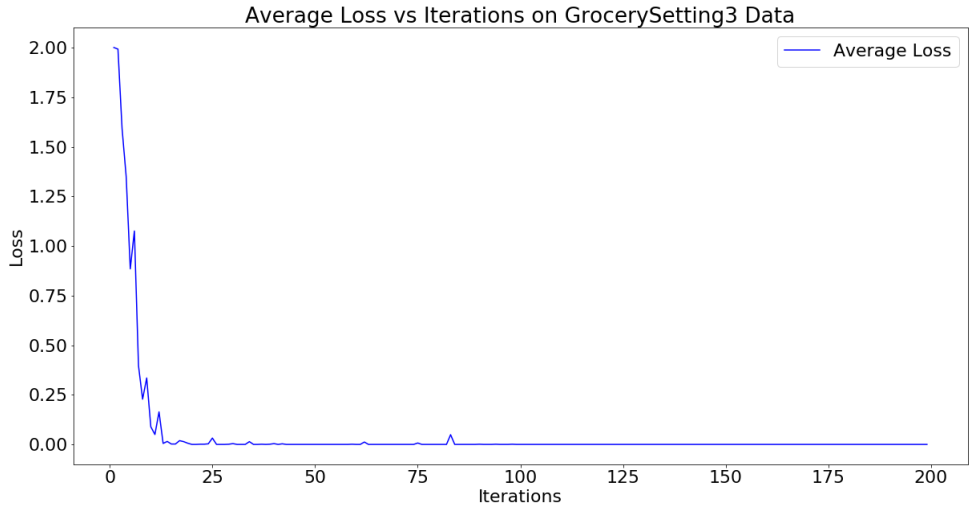
for each thuộc tính f_i

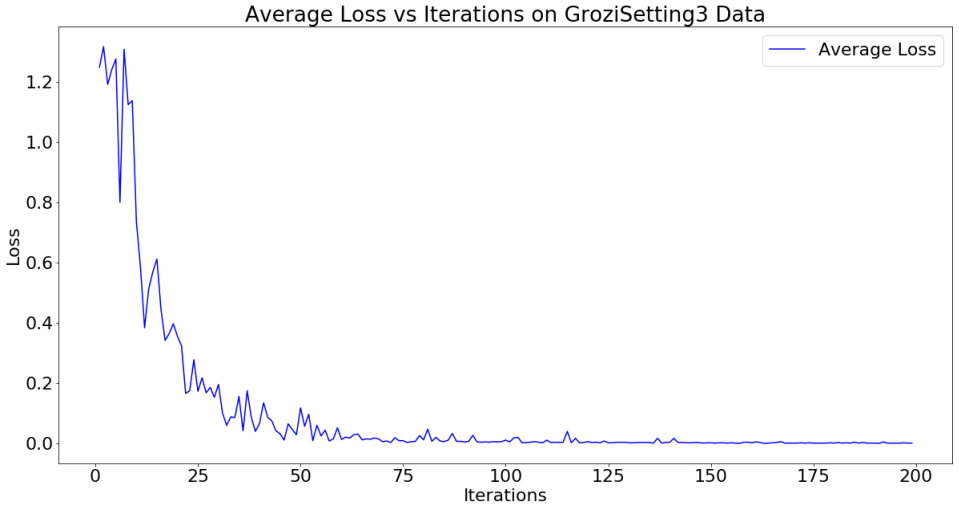
$w_i \leftarrow w_i + \eta (c_x - y_x) x_i$

end while

return w

Các điều kiện kết thúc quá trình học





Các điều kiện kết thúc quá trình học

- Trong các giải thuật `LSLR_batch` và `LSLR_incremental`, quá trình học kết thúc khi các điều kiện được chỉ định bởi `CONVERGENCE` được thỏa mãn
- Các điều kiện kết thúc học thường được định nghĩa dựa trên một số tiêu chí đánh giá hiệu năng hệ thống
  - Kết thúc, nếu giá trị lỗi nhỏ hơn giá trị ngưỡng
  - Kết thúc, nếu giá trị lỗi ở một bước học lớn hơn giá trị lỗi ở bước học trước
  - Kết thúc, nếu sự khác biệt giữa các giá trị lỗi ở 2 bước học liên tiếp nhỏ hơn giá trị ngưỡng
  - ...

Hàm lỗi thực nghiệm

- Với một tập huấn luyện **D** = {($\mathbf{x}_1$, y_1), ($\mathbf{x}_2$, y_2), …, ($\mathbf{x}_M$, y_M)}, cần học hàm f
- **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 lỗi **E** (của hàm *f* học được) trên tập học ***D***

- $\left|\frac{1}{M} RSS(f) - \boldsymbol{E}\right|$ thường được gọi là **lỗi tổng quát hoá** (generalization error) của hàm *f* học được

Bình phương tối thiểu (OLS)

- Với tập huấn luyện **D**, cần tìm (học) hàm *f* sao cho *RSS* nhỏ nhất.

$$\mathbf{E} = \arg\min_{f \in \boldsymbol{H}} RSS(f)$$

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

- Đây được gọi là **phương pháp bình phương tối thiểu** (ordinary least squares - OLS)

Bình phương tối thiểu (OLS)

- Các nhược điểm của phương pháp OLS:
  - Không phù hợp nếu số chiều (n) lớn
  - Khả năng bị học quá khớp (overfitting) là cao vì việc học hàm *f* **chỉ quan tâm việc tối thiểu lỗi đối với tập học**

Hồi quy Ridge (1)

- Với một tập học **D** = {($\mathbf{x}_1$, y_1), ($\mathbf{x}_2$, y_2), …, ($\mathbf{x}_M$, y_M)}, cần tìm (học) hàm *f* sao cho:

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

trong đó: $\|\boldsymbol{w}\|_2^2 = \sum_{j=0}^{n} {w_j}^2$

λ(>0) là **một hằng số phạt** (penalty)

Hồi quy Ridge (2)

- Đại lượng hiệu chỉnh (phạt) $\lambda\|\boldsymbol{w}\|_2^2$
  - Có vai trò hạn chế độ lớn của vector trọng số **w***
- Lựa chọn giá trị λ phù hợp để cân bằng giữa:
  - Chất lượng (vd: độ chính xác) của hàm f học được đối với tập học **D**,
  - Khả năng phán đoán tốt hơn với các ví dụ trong tương lai (khả năng khái quát hóa của hàm *f*)
- Nếu giá trị λ quá bé (rất gần 0), thì Hồi quy Ridge trở thành Hồi quy OLS
- Nếu giá trị λ quá lớn, thì lỗi thực nghiệm (RSS(f) không còn quan trọng (đối với E), và thành phần phạt sẽ làm cho các giá trị trọng số w_i rất nhỏ (rất gần 0)

Hồi quy Ridge (3)

- So với phương pháp bình phương tối thiểu (OLS), thì phương pháp hồi quy Ridge:
  - Giảm khả năng học quá khớp (overfitting)
  - Lỗi trên tập học có thể cao hơn
- Chất lượng của hàm mục tiêu (*f*) học được phụ thuộc rất nhiều vào sự lựa chọn giá trị phù hợp của tham số phạt λ

Hồi quy Ridge: Ví dụ

- Xét tập dữ liệu Prostate gồm 67 ví dụ dùng để huấn luyện và 31 ví dụ dùng để kiểm thử. Mỗi ví dụ được biểu diễn bởi 8 thuộc tính.

w_i	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**

Hồi quy Lasso

- Hồi quy Ridge sử dụng chuẩn L^2 cho đại lượng hiệu chỉnh (phạt): $\|\boldsymbol{w}\|_2^2$
- **Hồi quy Lasso** sử dụng **chuẩn L^1** cho đại lượng hiệu chỉnh (phạt): $\|\boldsymbol{w}\|_1$
  - Với một tập học **D** = {($\mathbf{x}_1$, y_1), ($\mathbf{x}_2$, y_2), …, ($\mathbf{x}_M$, y_M)}, cần tìm (học) hàm *f* sao cho:

$$f^* = \arg\min_{f \in \boldsymbol{H}} RSS(f) + \lambda \|\boldsymbol{w}\|_1$$

- Hồi quy Lasso thường tạo ra nghiệm thưa, tức là nhiều thành phần của vector **w** có giá trị là 0
  - Hồi quy Lasso thực hiện đồng thời 2 việc: hạn chế độ phức tạp của mô hình và lựa chọn thuộc tính

Thử nghiệm: OLS vs. Ridge vs. Lasso

- Xét tập dữ liệu Prostate gồm 67 ví dụ dùng để huấn luyện và 31 ví dụ dùng để kiểm thử. Mỗi ví dụ được biểu diễn bởi 8 thuộc tính.

w_i	OLS	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 =0
→ Chỉ ra các thuộc tính không quan trọng