

# 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 2020-2021

# 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):

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

- 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

$$f(x) = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n = w_0 + \sum_{i=1}^n w_ix_i \quad (w_i, x_i \in \mathbb{R})$$

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

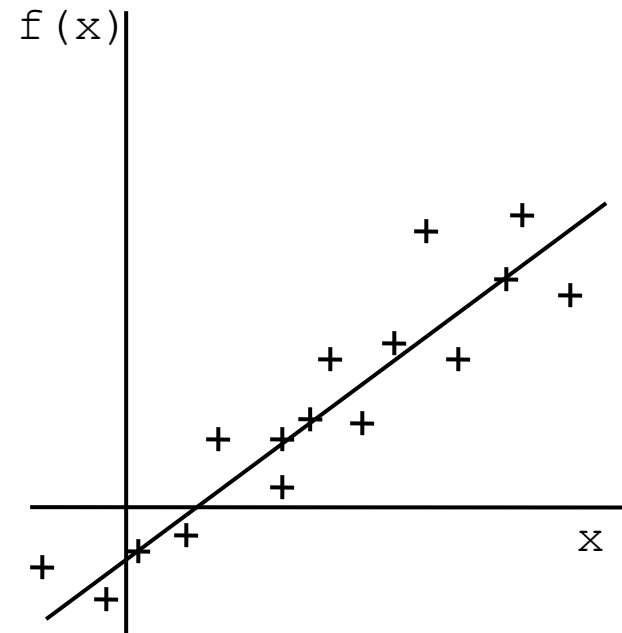
$$f: X \rightarrow Y$$

- $X$ : Miền không gian đầu vào (không gian vector  $n$  chiều –  $\mathbb{R}^n$ )
  - $Y$ : Miền không gian đầu ra (miền các giá trị số thực –  $\mathbb{R}$ )
  - $f$ : Hàm mục tiêu cần học (một hàm ánh xạ tuyến tính)
- Thực chất, là học một vector các trọng số:  $w = (w_0, w_1, w_2, \dots, w_n)$

# 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  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , trong đó  $x_i \in \mathbb{R}$ 
  - Giá trị đầu ra mong muốn  $c_x (\in \mathbb{R})$
  - 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  $\mathbf{z} = (z_1, z_2, \dots, z_n)$ 
  - 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 vector trọng số  $\mathbf{w}$  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 vector trọng số  $\mathbf{w}$
  - Tính toán giá trị lỗi huấn luyện  $E$
  - Cập nhật vector trọng số  $\mathbf{w}$  theo **quy tắc delta (delta rule)**
  - Lặp lại, cho đến khi hội tụ về một giá trị lỗi nhỏ nhất (cực 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  $\mathbf{w}^* = (w^*_0, w^*_1, \dots, w^*_n)$  là vector trọng số đã học được

# Quy tắc delta

- Để cập nhật vector trọng số  $\mathbf{w}$  theo hướng giúp giảm bớt giá trị lỗi huấn luyện  $E$ 
  - $\eta$  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( $D, \eta$ )

```
for each thuộc tính  $f_i$ 
     $w_i \leftarrow$  giá trị (nhỏ) được khởi tạo ngẫu nhiên
while not CONVERGENCE
    for each thuộc tính  $f_i$ 
         $\text{delta\_}w_i \leftarrow 0$ 
        for each ví dụ học  $x \in D$ 
            Tính toán giá trị đầu ra thực tế  $y_x$ 
            for each thuộc tính  $f_i$ 
                 $\text{delta\_}w_i \leftarrow \text{delta\_}w_i + \eta (c_x - y_x) x_i$ 
            for each thuộc tính  $f_i$ 
                 $w_i \leftarrow w_i + \text{delta\_}w_i$ 
    end while
return  $w$ 
```

## LSLR\_incremental(D, $\eta$ )

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  $y_x$

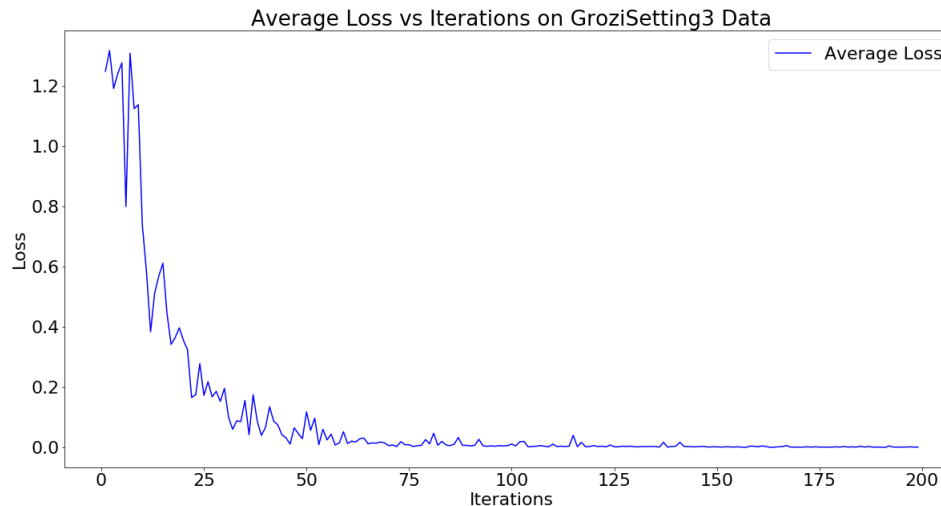
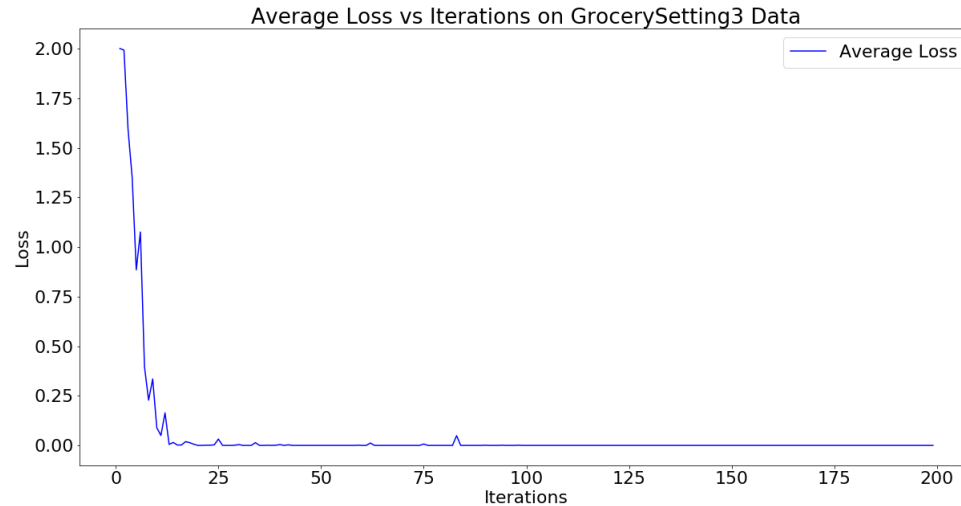
for each thuộc tính  $f_i$

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

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
  - ...