

Mục lục

I	Lý thuyết	1
1	Học có giám sát	2
1.1	Cơ sở lý thuyết	2
1.2	Các mô hình tuyến tính	9

Phần I

Lý thuyết

Chương 1

Học có giám sát

1.1 Cơ sở lý thuyết

1.1.1 Chuẩn của vectơ và ma trận

Trong học máy, các đối tượng thường được mô tả bởi vectơ hoặc ma trận thực. Vì vậy, chúng ta cần một thước đo để đánh giá độ lớn của từng đối tượng cũng như độ khác biệt (hay tương đồng) giữa các đối tượng.

Chuẩn của vectơ

Trong không gian vectơ thực \mathbb{R}^n , một vectơ \mathbf{x} , ký hiệu bằng chữ cái đậm, có các thành phần là số thực, ký hiệu bằng chữ cái thường. Vectơ luôn được viết dưới dạng cột

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^\top \quad \text{hay} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (1.1)$$

Xét chuẩn Minkowski, $\|\mathbf{x}\|_p$, với $p \geq 1$, đọc là *chuẩn ℓ_p của \mathbf{x}* , là số thực không âm xác định bởi.

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}. \quad (1.2)$$

Một số chuẩn Minkowski hay dùng:

- 1) Với $p = 2$, ta có chuẩn Euclid, thường được dùng làm chuẩn mặc định. Đó là

căn bậc hai của tổng bình phương các tọa độ:

$$\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}. \quad (1.3)$$

2) Với $p = 1$, ta có chuẩn Manhattan, bằng tổng trị tuyệt đối các thành phần

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|. \quad (1.4)$$

3) Với $p = \infty$, chuẩn vô cùng của \mathbf{x} , được định nghĩa bởi

$$\|\mathbf{x}\|_\infty = \lim_{p \rightarrow \infty} \|\mathbf{x}\|_p = \lim_{p \rightarrow \infty} \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}. \quad (1.5)$$

Ta có thể chứng minh chuẩn vô cùng của \mathbf{x} bằng giá trị lớn nhất của trị tuyệt đối các thành phần:

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|. \quad (1.6)$$

Chuẩn Minkowski có các tính chất chung của chuẩn, như sau:

1) Tính xác định dương:

$$\|\mathbf{x}\| \geq 0, \forall \mathbf{x} \in \mathbb{R}^n, \quad \text{và} \quad \|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}. \quad (1.7)$$

Lưu ý, khi viết $\|\mathbf{x}\| \geq 0$ hay $\|\mathbf{x}\| = 0$, ta hiểu 0 là số thực, nhưng khi viết $\mathbf{x} = \mathbf{0}$, ta phải hiểu $\mathbf{0}$ là vectơ không, $(0, 0, \dots, 0)$, gồm n số thực bằng 0. Như vậy cùng ký hiệu là 0 hoặc $\mathbf{0}$, nhưng ý nghĩa trong mỗi tình huống hoàn toàn khác nhau, người ta thường gọi là tính *tương thích* theo tình huống. Chúng ta cần nhớ điều này đặc biệt trong các bài trình bày bằng chữ viết tay, nơi người ta viết 0 và $\mathbf{0}$ là như nhau, và người đọc phải tự nhận diện đó là gì, cụ thể số thực không $0 \in \mathbb{R}$, hay là vectơ không $\mathbf{0} \in \mathbb{R}^n$, tức là đối tượng trong không gian nào. Cũng tương tự như vậy, ký hiệu đậm đặc trưng cho vectơ \mathbf{x} cũng được viết là x trong trình bày viết tay, người đọc cũng cần nhận dạng nhanh kiểu dữ liệu nó.

2) Tính thuần nhất dương:

$$\|k\mathbf{x}\| = |k| \|\mathbf{x}\| \quad \forall \mathbf{x} \in \mathbb{R}^n, k \in \mathbb{R}. \quad (1.8)$$

3) Bất đẳng thức tam giác:

$$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \quad (1.9)$$

Ví dụ 1. Cho $\mathbf{x} = (6, -2, 3)^\top \in \mathbb{R}^3$. Tính các chuẩn Euclid, chuẩn Manhattan, và chuẩn vô cùng của \mathbf{x} .

Giải. Các chuẩn Euclid, chuẩn Manhattan, và chuẩn vô cùng của \mathbf{x} lần lượt là

$$\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \sqrt{6^2 + 2^2 + 3^2} = 7$$

$$\|\mathbf{x}\|_1 = 6 + 2 + 3 = 11$$

$$\|\mathbf{x}\|_\infty = \max\{6, 2, 3\} = 6.$$

Mã lệnh Python để tính các chuẩn trên:

```
1 import numpy as np
2 x = np.array([6, -2, 3])
3 np.linalg.norm(x)           # hoặc np.linalg.norm(x,
                               2),  $\rightarrow \|\mathbf{x}\|$ 
4 np.linalg.norm(x, 1)        #  $\|\mathbf{x}\|_1$ 
5 np.linalg.norm(x, np.inf)   #  $\|\mathbf{x}\|_\infty$ 
```

□

Chuẩn của ma trận

Cho ma trận thực \mathbf{A} cỡ $m \times n$ gồm m hàng và n cột, trong đó phần tử ở hàng i cột j , $1 \leq i \leq m$, $1 \leq j \leq n$, được ký hiệu là $a_{i,j}$, hoặc a_{ij} nếu không gây nhầm lẫn. Ta viết $\mathbf{A} = (a_{ij})_{m \times n} \in \mathbb{R}^{m \times n}$, hoặc chi tiết

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \dots & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}. \quad (1.10)$$

Với $p \geq 1$, chuẩn p của \mathbf{A} được xác định bởi:

$$\|\mathbf{A}\|_p = \sup_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p}. \quad (1.11)$$

trong đó \mathbf{x} là vectơ có cỡ *tương thích*, tức là phép nhân \mathbf{Ax} thực hiện được. Vì \mathbf{A} có cỡ $m \times n$, nên \mathbf{x} có cỡ $n \times 1$, tức là $\mathbf{x} \in \mathbb{R}^n$. Như vậy chuẩn $\|\mathbf{x}\|_p$ là chuẩn p trong \mathbb{R}^n , còn $\|\mathbf{Ax}\|_p$ là chuẩn p trong \mathbb{R}^m .

Chuẩn của ma trận được định nghĩa trên cơ sở chuẩn của vectơ, nên ta cũng có các chuẩn thông dụng tương ứng của ma trận tùy theo giá trị của p .

1) Với $p = \infty$, chuẩn vô cùng của \mathbf{A} xác định bởi:

$$\|\mathbf{A}\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|, \quad (1.12)$$

tức là giá trị lớn nhất của các tổng trị tuyệt đối các phần tử trên mỗi hàng.

2) Với $p = 1$, chuẩn 1 của \mathbf{A} có công thức khá tương tự (1.12), chỉ khác ở chỗ thay đổi thứ tự chỉ số hàng i và cột j trong tính toán:

$$\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|, \quad (1.13)$$

tức là giá trị lớn nhất của các tổng trị tuyệt đối các phần tử trên mỗi cột.

3) Với $p = 2$, chuẩn 2 của \mathbf{A} được sử dụng nhiều, nhưng quy trình tính toán lại khá phức tạp. Đó là căn bậc hai của giá trị riêng lớn nhất của ma trận $\mathbf{A}^\top \mathbf{A}$:

$$\|\mathbf{A}\|_2 = \sqrt{\max_{1 \leq i \leq n} \lambda_i(\mathbf{A}^\top \mathbf{A})}. \quad (1.14)$$

Ngoài ra, một chuẩn khá phổ biến của ma trận, được tính tương tự chuẩn Euclid cho vectơ, bằng cách xem ma trận như là một vectơ được “ngắt” thành nhiều đoạn và xếp chồng lên nhau. Chuẩn này được gọi là chuẩn Frobenius, gọi tắt là chuẩn F:

$$\|\mathbf{A}\| = \|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}. \quad (1.15)$$

Ví dụ 2. Tính các chuẩn vô cùng, chuẩn 1, chuẩn 2, và chuẩn Frobenius của

$$\mathbf{A} = \begin{bmatrix} 5 & 0 & 2 \\ 3 & -2 & -4 \end{bmatrix}.$$

Giải. 1) Chuẩn vô cùng của \mathbf{A} :

$$\|\mathbf{A}\|_{\infty} = \max\{5 + 0 + 2, 3 + 2 + 4\} = \max\{7, 9\} = 9.$$

2) Chuẩn 1 của \mathbf{A} :

$$\|\mathbf{A}\|_1 = \max\{5 + 3, 0 + 2, 2 + 4\} = \max\{8, 2, 6\} = 8.$$

3) Để tính chuẩn 2 của \mathbf{A} , ta tiến hành từng bước sau:

$$\text{a) } \mathbf{A}^{\top} = \begin{bmatrix} 5 & 3 \\ 0 & -2 \\ 2 & -4 \end{bmatrix}$$

$$\text{b) } \mathbf{A}^{\top} \mathbf{A} = \begin{bmatrix} 5 & 3 \\ 0 & -2 \\ 2 & -4 \end{bmatrix} \begin{bmatrix} 5 & 0 & 2 \\ 3 & -2 & -4 \end{bmatrix} = \begin{bmatrix} 34 & -6 & -2 \\ -6 & 4 & 8 \\ -2 & 8 & 20 \end{bmatrix}$$

c) Đa thức đặc trưng của $\mathbf{A}^{\top} \mathbf{A}$:

$$\begin{aligned} P(\lambda) &= |\mathbf{A}^{\top} \mathbf{A} - \lambda \mathbf{I}| = \begin{vmatrix} 34 - \lambda & -6 & -2 \\ -6 & 4 - \lambda & 8 \\ -2 & 8 & 20 - \lambda \end{vmatrix} \\ &= (-\lambda)^3 + (34 + 4 + 20)(-\lambda)^2 \\ &\quad + \left(\begin{vmatrix} 34 & -6 \\ -6 & 4 \end{vmatrix} + \begin{vmatrix} 34 & -2 \\ -2 & 20 \end{vmatrix} + \begin{vmatrix} 4 & 8 \\ 8 & 20 \end{vmatrix} \right) (-\lambda) \\ &\quad + \begin{vmatrix} 34 & -6 & -2 \\ -6 & 4 & 8 \\ -2 & 8 & 20 \end{vmatrix} \\ &= -\lambda^3 + 58\lambda^2 - 792\lambda \end{aligned}$$

d) Các giá trị riêng của $\mathbf{A}^{\top} \mathbf{A}$ là nghiệm của đa thức đặc trưng $P(\lambda)$:

$$P(\lambda) = -\lambda(\lambda - 36)(\lambda - 22) = 0 \Leftrightarrow \begin{cases} \lambda = 36 & = \lambda_1 \\ \lambda = 22 & = \lambda_2 \\ \lambda = 0 & = \lambda_3 \end{cases}$$

e) Và cuối cùng, ta được chuẩn 2 của \mathbf{A} :

$$\|\mathbf{A}\|_2 = \sqrt{\max\{36, 22, 0\}} = \sqrt{36} = 6.$$

4) Chuẩn F của \mathbf{A} bằng

$$\|\mathbf{A}\|_F = \sqrt{(5^2 + 0^2 + 2^2) + (3^2 + 2^2 + 4^2)} = \sqrt{58}.$$

Mã lệnh Python để tính chuẩn của ma trận cũng tương tự đối với chuẩn của véc tơ:

```
1 import numpy as np
2 A = np.array([[5, 0, 2],
3               [3, -2, -4]])
4 np.linalg.norm(A, np.inf) #  $\|\mathbf{A}\|_\infty$ 
5 np.linalg.norm(A, 1)      #  $\|\mathbf{A}\|_1$ 
6 np.linalg.norm(A, 2)      #  $\|\mathbf{A}\|_2$ 
7 np.linalg.norm(A)          # hoặc  $\text{np.linalg.norm}(A,$ 
                             #  $\text{'fro'})$ ,  $\rightarrow \|\mathbf{A}\| = \|\mathbf{A}\|_F$ 
```

□

1.1.2 Bài toán bình phương tối thiểu

Cho ma trận \mathbf{A} cỡ $m \times n$, và véc tơ m chiều \mathbf{b} . Ta cần tìm véc tơ n chiều \mathbf{x} sao cho $\|\mathbf{Ax} - \mathbf{b}\|$ đạt giá trị nhỏ nhất. Bài toán được viết dưới dạng

$$\arg \min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|. \quad (1.16)$$

Phương trình chuẩn

Định lý 1.1. Nếu \mathbf{A} có hạng đầy đủ theo cột, tức là $\text{rank}(\mathbf{A}) = n$, thì nghiệm bình phương tối thiểu của bài toán (1.16) là

$$\mathbf{x} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}. \quad (1.17)$$

Công thức (1.17) còn gọi là *phương trình chuẩn*.

1.1.3 Khai triển kỳ dị

Xét ma trận \mathbf{A} cỡ $m \times n$ có hạng bằng r . Trường hợp phổ biến khi $m \geq n$, *khai triển kỳ dị* của \mathbf{A} có dạng

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad (1.18)$$

trong đó

- $\mathbf{\Sigma}$ là ma trận đường chéo cỡ $m \times n$, có các phần tử trên đường chéo chính là $\sigma_1, \sigma_2, \dots, \sigma_r, 0, 0, \dots, 0$:

$$\mathbf{\Sigma} = \text{diag}_{m \times n}(\sigma_1, \sigma_2, \dots, \sigma_r, 0, \dots, 0) = \left[\begin{array}{cccc|cccc} \sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \ddots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_r & 0 & \dots & 0 \\ \hline 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{array} \right]_{m \times n} \quad (1.19)$$

với các giá trị kỳ dị $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ xác định bởi

$$\sigma_i = \sqrt{\lambda_i}, \quad i = 1, \dots, n \quad (1.20)$$

trong đó $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ là các giá trị riêng của $\mathbf{A}^\top \mathbf{A}$, kể cả các giá trị riêng bằng 0.

- Cột thứ i của \mathbf{V} , $\mathbf{v}_i = \mathbf{V}_{:,i}$, gọi là *các vectơ riêng phải*, là các vectơ riêng của $\mathbf{A}^\top \mathbf{A}$ ứng với mỗi giá trị riêng λ_i :

$$\mathbf{A}^\top \mathbf{A} \mathbf{v}_i = \lambda_i \mathbf{v}_i \quad (1.21)$$

\mathbf{V} được thiết kế để thỏa mãn tính trực giao, tức là

$$\mathbf{V}^\top \mathbf{V} = \mathbf{I}_n, \quad \mathbf{I}_n \text{ là ma trận đơn vị cấp } n. \quad (1.22)$$

- Với $\sigma_i > 0$, dựng *các vectơ riêng trái*

$$\mathbf{u}_i = \frac{1}{\sigma_i} \mathbf{A} \mathbf{v}_i. \quad (1.23)$$

tương ứng là cột thứ i của \mathbf{U} . Sau đó tiếp tục bổ sung hệ vectơ cơ sở trực chuẩn trong $\ker(\mathbf{A}^\top)$ vào \mathbf{U} để đạt được ma trận trực giao cấp m :

$$\mathbf{U}^\top \mathbf{U} = \mathbf{I}_m. \quad (1.24)$$

Khai triển kỳ dị cho ta xác định ma trận *giả nghịch đảo Moore – Penrose* của \mathbf{A} :

$$\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T, \quad (1.25)$$

trong đó

$$\mathbf{\Sigma}^+ = \text{diag}_{n \times m} \left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r}, 0, \dots, 0 \right) = \left[\begin{array}{cccc|cccc} \frac{1}{\sigma_1} & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \ddots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{\sigma_r} & 0 & \dots & 0 \\ \hline 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{array} \right]_{n \times m} \quad (1.26)$$

để từ đó tìm được nghiệm bình phương tối thiểu của bài toán (1.16).

Định lý 1.2. *Nghiệm bình phương tối thiểu của bài toán (1.16) là*

$$\mathbf{x} = \mathbf{A}^+ \mathbf{b}. \quad (1.27)$$

Đặt

$$\mathbf{\Sigma}_r = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \quad (\text{ma trận vuông cấp } r) \quad (1.28)$$

$$\mathbf{\Sigma}_r^+ = \text{diag} \left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r} \right) \quad (1.29)$$

$$\mathbf{U}_r = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r] \quad (r \text{ cột đầu của } \mathbf{U}, \text{ cỡ } m \times r) \quad (1.30)$$

$$\mathbf{V}_r = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r] \quad (r \text{ cột đầu của } \mathbf{V}, \text{ cỡ } n \times r), \quad (1.31)$$

ta thu được khai triển kỳ dị rút gọn của \mathbf{A} :

$$\mathbf{A} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T, \quad (1.32)$$

đồng thời

$$\mathbf{A}^+ = \mathbf{V}_r \mathbf{\Sigma}_r^+ \mathbf{U}_r^T. \quad (1.33)$$

1.2 Các mô hình tuyến tính

1.2.1 Phương pháp bình phương tối thiểu

Cho hàm số $f(\mathbf{x})$, với $\mathbf{x} = (x_1, x_2, \dots, x_d)^T \in \mathbb{R}^d$ là tập các thuộc tính hay đặc trưng. Giả sử chúng ta biết giá trị của hàm số tại m điểm $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)})^T$,

$i = 1, 2, \dots, m$:

$$y_i = f(\mathbf{x}^{(i)}), \quad i = 1, 2, \dots, m. \quad (1.34)$$

Các thông tin trên thường được mô tả dưới dạng bảng dữ liệu

$\mathbf{x}^{(i)\top}$	x_1	x_2	\dots	x_d	f
$\mathbf{x}^{(1)\top}$	$x_1^{(1)}$	$x_2^{(1)}$	\dots	$x_d^{(1)}$	y_1
$\mathbf{x}^{(2)\top}$	$x_1^{(2)}$	$x_2^{(2)}$	\dots	$x_d^{(2)}$	y_2
\dots	\dots	\dots	\dots	\dots	\dots
$\mathbf{x}^{(m)\top}$	$x_1^{(m)}$	$x_2^{(m)}$	\dots	$x_d^{(m)}$	y_m

(1.35)

Ta cần tìm một hàm $\hat{f}(\mathbf{x})$ để dự đoán $f(\mathbf{x})$ sao cho sai số bình phương

$$SE = \sum_{i=1}^m [\hat{f}(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(i)})]^2 = \sum_{i=1}^m (\hat{y}_i - y_i)^2 = \|\hat{\mathbf{y}} - \mathbf{y}\|^2 \quad (1.36)$$

đạt giá trị nhỏ nhất. Ở đây $\hat{y}_i = \hat{f}(\mathbf{x}^{(i)})$, $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_m)^\top$, và $\mathbf{y} = (y_1, y_2, \dots, y_m)^\top$. Bài toán bình phương tối thiểu thường, tức là tìm \hat{f} làm cực tiểu hàm mục tiêu SE, được ký hiệu dưới dạng

$$\arg \min_{\hat{f}} SE. \quad (1.37)$$

Xét không gian vectơ V gồm các hàm xác định tại các điểm $\mathbf{x}^{(i)}$, $i = 1, \dots, m$. Trong V , xét hệ n hàm độc lập tuyến tính

$$\mathbf{f}_{\text{base}} = (f_1, f_2, \dots, f_n)^\top. \quad (1.38)$$

Ta sẽ tìm hàm \hat{f} sinh bởi hệ hàm \mathbf{f}_{base} , tức là

$$\hat{f} = \sum_{j=1}^n w_j f_j = \mathbf{w}^\top \mathbf{f}_{\text{base}}, \quad \text{trong đó } \mathbf{w} = (w_1, w_2, \dots, w_n)^\top. \quad (1.39)$$

Tại mỗi điểm dữ liệu $\mathbf{x}^{(i)}$, $i = 1, \dots, m$, ta có

$$\hat{y}_i = \hat{f}(\mathbf{x}^{(i)}) = \sum_{j=1}^n w_j f_j(\mathbf{x}^{(i)}). \quad (1.40)$$

Ma trận thiết kế chứa giá trị tại mỗi điểm dữ liệu của từng hàm trong hệ hàm:

$$\mathbf{A} = (a_{ij})_{m \times n} \quad \text{với} \quad a_{ij} = f_j(\mathbf{x}^{(i)}), \quad (1.41)$$

và ký hiệu hàng thứ i của \mathbf{A} là $\mathbf{A}_{i, \cdot}$. Khi đó

$$\hat{y}_i = \mathbf{A}_{i, \cdot} \mathbf{w} \quad (1.42)$$

Suy ra

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_m \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1,:} \mathbf{w} \\ \vdots \\ \mathbf{A}_{m,:} \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1,:} \\ \vdots \\ \mathbf{A}_{m,:} \end{bmatrix} \mathbf{w} = \mathbf{A} \mathbf{w} \quad (1.43)$$

Bài toán bình phương tối thiểu (1.37) trở thành

$$\arg \min_{\mathbf{w}} \|\mathbf{A} \mathbf{w} - \mathbf{y}\|^2, \quad (1.44)$$

có nghiệm được cho bởi phương trình chuẩn (1.17), tức là

$$\mathbf{w} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{y}, \quad (1.45)$$

nếu đáp ứng được điều kiện $\text{rank}(\mathbf{A}) = n$, hoặc nếu không, sử dụng (1.27) theo khai triển kỳ dị:

$$\mathbf{w} = \mathbf{A}^+ \mathbf{y}. \quad (1.46)$$

Lưu ý: \hat{f} làm cực tiểu sai số bình phương $\text{SE} = \|\hat{\mathbf{y}} - \mathbf{y}\|^2$ tương đương với làm cực tiểu sai số bình phương trung bình

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 = \frac{1}{m} \|\hat{\mathbf{y}} - \mathbf{y}\|^2 = \frac{1}{m} \|\mathbf{A} \mathbf{w} - \mathbf{y}\|^2, \quad (1.47)$$

cũng như căn bậc hai của sai số bình phương trung bình

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2} = \frac{1}{\sqrt{m}} \|\hat{\mathbf{y}} - \mathbf{y}\| = \frac{1}{\sqrt{m}} \|\mathbf{A} \mathbf{w} - \mathbf{y}\|, \quad (1.48)$$

tức là các bài toán tối ưu $\arg \min_{\hat{f}} \text{SE}$, $\arg \min_{\hat{f}} \text{MSE}$, và $\arg \min_{\hat{f}} \text{RMSE}$ có chung nghiệm \hat{f} (nếu tồn tại), chỉ khác ở chỗ giá trị của hàm mục tiêu. Ta viết

$$\arg \min_{\hat{f}} \text{SE} = \arg \min_{\hat{f}} \text{MSE} = \arg \min_{\hat{f}} \text{RMSE}. \quad (1.49)$$

Mô hình hồi quy tuyến tính

Mô hình hồi quy tuyến tính có hàm dự báo

$$\hat{f}(\mathbf{x}) = b + \mathbf{w}^\top \mathbf{x}, \quad (1.50)$$

trong đó $b \in \mathbb{R}$ là hệ số chặn, vectơ $\mathbf{w} = (w_1, w_2, \dots, w_d)^\top \in \mathbb{R}^d$ gồm các hệ số (hay trọng số) hồi quy. Ta viết lại hàm dự báo dưới dạng

$$\hat{f}(\mathbf{x}) = b \cdot 1 + w_1 x_1 + w_2 x_2 + \dots + w_d x_d, \quad (1.51)$$

dẫn đến việc xét hệ hàm và vectơ tham số:

$$\mathbf{f}_{\text{base}} = \begin{bmatrix} f_0(\mathbf{x}) = 1 \\ f_1(\mathbf{x}) = x_1 \\ \vdots \\ f_d(\mathbf{x}) = x_d \end{bmatrix}, \quad \tilde{\mathbf{w}} = \begin{bmatrix} 1 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} 1 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \quad (1.52)$$

để quy về bài toán bình phương tối thiểu

$$\hat{f}(\mathbf{x}) = \tilde{\mathbf{w}}^\top \mathbf{f}_{\text{base}}. \quad (1.53)$$

Trong trường hợp này, ma trận thiết kế \mathbf{A} các giá trị tại các điểm dữ liệu của hệ hàm theo công thức (1.41) được xác định bởi

$$\mathbf{A} = \begin{bmatrix} f_0(\mathbf{x}^{(1)}) & f_1(\mathbf{x}^{(1)}) & \dots & f_d(\mathbf{x}^{(1)}) \\ f_0(\mathbf{x}^{(2)}) & f_1(\mathbf{x}^{(2)}) & \dots & f_d(\mathbf{x}^{(2)}) \\ \dots & \dots & \dots & \dots \\ f_0(\mathbf{x}^{(m)}) & f_1(\mathbf{x}^{(m)}) & \dots & f_d(\mathbf{x}^{(m)}) \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_d^{(1)} \\ 1 & x_1^{(2)} & \dots & x_d^{(2)} \\ \dots & \dots & \dots & \dots \\ 1 & x_1^{(m)} & \dots & x_d^{(m)} \end{bmatrix}. \quad (1.54)$$

Trong bảng dữ liệu (1.35), ký hiệu *ma trận đặc trưng*

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(1)\top} \\ \mathbf{x}^{(2)\top} \\ \vdots \\ \mathbf{x}^{(m)\top} \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & \dots & x_d^{(1)} \\ x_1^{(2)} & \dots & x_d^{(2)} \\ \dots & \dots & \dots \\ x_1^{(m)} & \dots & x_d^{(m)} \end{bmatrix}, \quad (1.55)$$

thì

$$\mathbf{A} = \begin{bmatrix} \mathbf{1}_{m \times 1}, \mathbf{X} \end{bmatrix}, \quad (1.56)$$

trong đó $\mathbf{1}_{m \times 1}$ là ma trận cỡ $m \times 1$ gồm toàn số 1, còn gọi là *cột chặn*. Trong nhiều trường hợp, nếu không sợ gây nhầm lẫn, ta đơn giản chỉ cần viết $\mathbf{1}$.

Một ví dụ ở dạng đơn giản nhất, khi hàm dự báo chỉ phụ thuộc vào một thuộc tính duy nhất. Tuy nhiên, ví dụ sẽ giúp chúng ta trực quan hóa dữ liệu, góp phần giải thích tính đúng đắn của các phương pháp.

Ví dụ 3. Cho bảng dữ liệu của hàm số $f(x)$:

#	x	f
#1	1	0
#2	2	3
#3	3	2
#4	3	4
#5	5	5
#6	4	?

Giả sử dữ liệu tuân theo mô hình hồi quy tuyến tính, tức là có hàm dự báo $\hat{f}(x) = b + wx$.

- a) Thực thi mã lệnh với thư viện `scikit-learn` xác định các tham số b, w , và dự đoán $f(4)$. Từ đó
 - i) Xác định giá trị hàm mục tiêu MSE sau khi tối ưu mô hình.
 - ii) Trực quan hóa dữ liệu bao gồm: các điểm dữ liệu, điểm dự đoán, và đường hồi quy.
- b) Xác định các tham số b, w bằng phương trình chuẩn. So sánh kết quả với ý (a).
- c) Xác định các tham số b, w bằng phương pháp khai triển kỳ dị. So sánh kết quả với ý (a).

1.2.2 Hồi quy và phân loại Ridge

Hồi quy

$$\arg \min_w \|Aw - y\|^2 + \alpha \|w\|^2. \quad (1.57)$$

1.2.3 Lasso

$$\arg \min_w \frac{1}{2m} \|Aw - y\|^2 + \alpha \|w\|^2. \quad (1.58)$$

1.2.4 Elastic-Net

$$\arg \min_w \frac{1}{2m} \|Aw - y\|^2 + \alpha \left(\rho \|w\|_1 + \frac{1 - \rho}{2} \|w\|^2 \right). \quad (1.59)$$

1.2.5 Hồi quy logistic

Cho dữ liệu phân loại nhị phân

$\mathbf{x}^{(i)\top}$	x_1	x_2	y
$\mathbf{x}^{(1)\top}$	1	2	1
$\mathbf{x}^{(2)\top}$	2	3	1
$\mathbf{x}^{(3)\top}$	3	1	1
$\mathbf{x}^{(4)\top}$	2	5	0
$\mathbf{x}^{(5)\top}$	4	3	0
---	1	4	?

```

1 import numpy as np
2 from sklearn.linear_model import
   LogisticRegression
3
4 X = np.array([[1, 2],
5               [2, 3],
6               [3, 1],
7               [2, 5],
8               [4, 3]])
9 Y = np.array([1, 1, 1, 0, 0])
10
11 model = LogisticRegression()
12 model.fit(X, Y)
13
14 X_pred = np.array([[1, 4]])
15 Y_pred = model.predict(X_pred) # → array([1])
16
17 model.coef_ # → array([[-0.6455, -0.8844]])
18 model.intercept_ # → array([4.5923])
19
20 w1, w2 = model.coef_[0]
21 b = model.intercept_[0]

```

```
17 # Vẽ biểu đồ
18 plt.figure()

19 # (1) Vẽ các điểm gán nhãn 1
20 plt.scatter(
21     X[Y == 1][:, 0], X[Y == 1][:, 1],
22     color='blue', label='Các điểm gán nhãn 1'
23 )

24 # (2) Vẽ các điểm gán nhãn 0
25 plt.scatter(
26     X[Y == 0][:, 0], X[Y == 0][:, 1],
27     color='red', label='Các điểm gán nhãn 0'
28 )

29 # (3) Vẽ các điểm mới với nhãn dự báo
30 plt.scatter(
31     X_pred[0, 0], X_pred[0, 1],
32     color='purple', marker='x', s=100,
33     label='Điểm với nhãn dự báo là: 1'
34 )

35 # (4) Xây dựng đường phân tách
36 a = - w1 / w2
37 b_ = - b / w2
38 X1_grid = np.linspace(0, 5, 100)
39 X2_grid = a * X1_grid + b_
40 plt.plot(
41     X1_grid, X2_grid,
42     color='green', label='Đường phân tách'
43 )

44 # Gán nhãn cho các thành phần, đặt tên trục và tiêu đề
45 plt.legend()
46 plt.xlabel('x1')
47 plt.ylabel('x2')

```

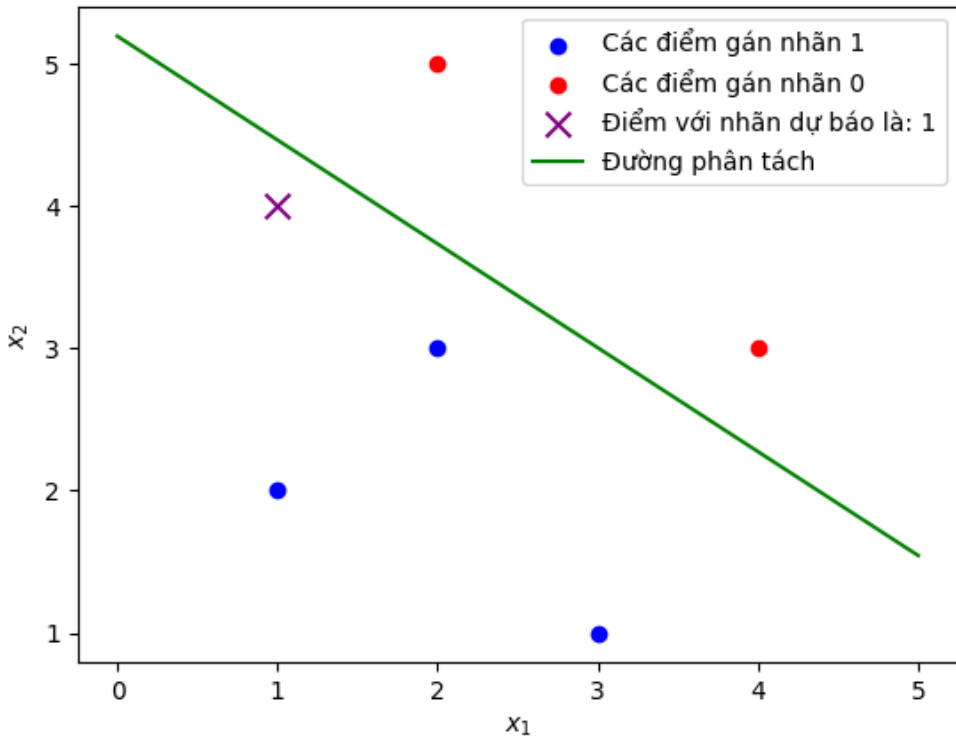


```

48 # Hiển thị biểu đồ
49 plt.show()

```

Mã 1.1: Hồi quy logistic: thư viện scikit-learn



```

1 import numpy as np
2 from scipy.optimize import minimize
3 import matplotlib.pyplot as plt

4 X = np.array([[1, 2],
5               [2, 3],
6               [3, 1],
7               [2, 5],
8               [4, 3]])
9 Y = np.array([1, 1, 1, 0, 0])

10 # Thêm cột bias (ma trận toàn 1) vào X
11 X = np.concatenate((np.ones(shape=(len(X), 1)),
12                     X), axis=1)

```

```

12 # Hàm sigmoid
13 def sigmoid(z):
14     return 1 / (1 + np.exp(-z))

15 # Hàm mất mát (log-likelihood hay cross-entropy loss)
16 def loss(w):
17     L = 0
18     for x, y in zip(X, Y):
19         if y == 0:
20             L += - np.log(1 - sigmoid(np.dot(w,
21 x)))
22         else:
23             L += - np.log(sigmoid(np.dot(w, x)))
24     return L / len(X)

25 # Hàm tính gradient của hàm mất mát
26 def gradient(theta):
27     G = 0 # mặc dù là số nhưng sau khi thực hiện vòng
lặp lại là véctơ
28     for x, y in zip(X, Y):
29         G += (sigmoid(np.dot(theta, x)) - y) * x
30     return G / len(X)

31 # Khởi tạo các tham số
32 # vector tham số ban đầu với kích thước (n+1)
33 initial_theta = np.zeros(X.shape[1])
34 # Sử dụng scipy.optimize.minimize để tối ưu hóa hàm mất mát
35 result = minimize(loss, initial_theta,
36     method='BFGS', jac=gradient)
37 result

```

```

message: Optimization terminated successfully.
success: True
status: 0
      fun: 2.39804740936585e-06
         x: [ 8.653e+01 -1.303e+01 -1.637e+01]
        nit: 24

```

```

        jac: [-2.230e-06 -4.293e-06 -6.690e-06]
    hess_inv: [[ 6.837e+06 -1.104e+06 -1.215e+06]
               [-1.104e+06  1.784e+05  1.962e+05]
               [-1.215e+06  1.962e+05  2.158e+05]]
    nfev: 25
    njev: 25

```

```

1  w = result.x

2  # Dự đoán nhãn cho điểm mới (1, 4)
3  x_pred = np.array([1, 1, 4]) # thêm 1 cho bias
4  prob_pred = sigmoid(np.dot(w, x_pred))

5  y_pred = 1 if prob_pred >= 0.5 else 0

6  a = - w[1] / w[2]
7  b = - w[0] / w[2]

8  # Trực quan hóa các điểm và đường phân tách
9  plt.figure()

10 # (1) Vẽ các điểm gán nhãn 1
11 plt.scatter(X[Y == 1][:, 1], X[Y == 1][:, 2],
              color='blue', label='Nhãn
              1')

12 # (2) Vẽ các điểm gán nhãn 0
13 plt.scatter(X[Y == 0][:, 1], X[Y == 0][:, 2],
              color='red', label='Nhãn
              0')

14 # (3) Xây dựng đường phân tách
15 X1 = np.linspace(0, 5, 100)
16 X2 = a * X1 + b
17 plt.plot(X1, X2, label='Đường phân
              tách', color='green')

18 # (4) Vẽ điểm mới (1, 4) với nhãn dự báo
19 plt.scatter(x_pred[1], x_pred[2], color='purple',

```

```

    marker='x',
20         s=100, label=f'Diểm (1, 4), nhãn dự
        báo: {y_pred}')

21 # Đặt tên trục và tiêu đề
22 plt.xlabel('x1')
23 plt.ylabel('x2')
24 plt.legend()

25 # Hiển thị biểu đồ
26 plt.show()

```

Mã 1.2: Hồi quy logistic: tối ưu bằng scipy

```

1  import numpy as np
2  import matplotlib.pyplot as plt

3  X = np.array([[1, 2],
4                [2, 3],
5                [3, 1],
6                [2, 5],
7                [4, 3]])
8  Y = np.array([1, 1, 1, 0, 0])

9  X = np.concatenate((np.ones(shape=(len(X), 1)),
10                      X), axis=1)

11 def sigmoid(z):
12     return 1 / (1 + np.exp(-z))

13 def loss(w):
14     L = 0
15     for x, y in zip(X, Y):
16         if y == 0:
17             L += - np.log(1 - sigmoid(np.dot(w,
18                                             x)))
19         else:
20             L += - np.log(sigmoid(np.dot(w, x)))
21     return L / len(X)

```

```

20 def gradient(w):
21     G = 0
22     for x, y in zip(X, Y):
23         G += (sigmoid(np.dot(w, x)) - y) * x
24     return G / len(X)

25 # Phương pháp hướng giảm
26 w0 = np.zeros(X.shape[1])
27 learning_rate = 0.1
28 tolerance = 1e-5
29 max_iters = 1e4

30 w = w0
31 num_iters = 0
32 loss_history = [loss(w)]
33 while True:
34     num_iters += 1
35     w -= learning_rate * gradient(w)
36     loss_history.append(loss(w))
37     if np.linalg.norm(gradient(w)) < tolerance or
num_iters == max_iters:
38         break

39 num_iters

40 w

41 loss_history

42 x_pred = np.array([1, 1, 4])
43 prob_pred = sigmoid(np.dot(w, x_pred))
44 y_pred = 1 if prob_pred >= 0.5 else 0

45 a = - w[1] / w[2]
46 b = - w[0] / w[2]

47 plt.figure()

```

```

48 plt.scatter(X[Y == 1][:, 1], X[Y == 1][:, 2],
    color='blue')
49 plt.scatter(X[Y == 0][:, 1], X[Y == 0][:, 2],
    color='red')
50 plt.scatter(x_pred[1], x_pred[2], color='purple',
    marker='x', s=100)

51 X1 = np.linspace(0, 5, 100)
52 X2 = a*X1 + b
53 plt.plot(X1, X2, color='green')
54 plt.show()

```

Mã 1.3: Hồi quy logistic: phương pháp hướng giảm

$$\tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}, \quad \tilde{\mathbf{w}} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \quad (1.60)$$

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_d^{(1)} \\ 1 & x_1^{(2)} & \dots & x_d^{(2)} \\ \dots & \dots & \dots & \dots \\ 1 & x_1^{(m)} & \dots & x_d^{(m)} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_{m \times 1}, \mathbf{X} \end{bmatrix} \quad (1.61)$$

$$z = b + \mathbf{w}^\top \mathbf{x} = \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} \quad (1.62)$$

Để tránh phức tạp về ký hiệu, từ bây giờ, ta thay $\tilde{\mathbf{x}}$ bởi \mathbf{x} , $\tilde{\mathbf{w}}$ bởi \mathbf{w} , và $\tilde{\mathbf{X}}$ bởi \mathbf{X} . Ta viết lại

$$z = \mathbf{w}^\top \mathbf{x} \quad (1.63)$$

$$\nabla_{\mathbf{w}} z = \mathbf{x}. \quad (1.64)$$

$$p = \sigma(z) \Rightarrow \frac{dp}{dz} = \sigma(z)(1 - \sigma(z)) = p(1 - p) \quad (1.65)$$

$$\frac{d}{dz} \log p = \frac{1}{p} \frac{dp}{dz} = \frac{1}{p} p(1 - p) = 1 - p \quad (1.66)$$

$$\frac{d}{dz} \log(1 - p) = \frac{1}{1 - p} \frac{d(1 - p)}{dz} = \frac{1}{1 - p} \left(-\frac{dp}{dz} \right) = -\frac{1}{1 - p} p(1 - p) = -p \quad (1.67)$$

$$\ell = \ell(\mathbf{w}; \mathbf{x}, y) = -y \log p - (1 - y) \log(1 - p), \quad (1.68)$$

với $y \in \{0, 1\}$, $p = \sigma(z) = \sigma(\mathbf{w}^\top \mathbf{x})$. Xét hai trường hợp

Trường hợp 1: $y = 1 \Rightarrow \ell = -\log p \Rightarrow \nabla_{\mathbf{w}} \ell = \frac{d\ell}{dz} \nabla_{\mathbf{w}} z = \frac{d(-\log p)}{dz} \nabla_{\mathbf{w}} z =$
 $-(1 - p) \mathbf{x} = [\sigma(\mathbf{w}^\top \mathbf{x}) - y] \mathbf{x}.$

Trường hợp 2: $y = 0 \Rightarrow \ell = -\log(1 - p) \Rightarrow \nabla_{\mathbf{w}} \ell = \frac{d\ell}{dz} \nabla_{\mathbf{w}} z = -\frac{d\log(1 - p)}{dz} \nabla_{\mathbf{w}} z =$
 $-(-p) \mathbf{x} = [\sigma(\mathbf{w}^\top \mathbf{x}) - y] \mathbf{x}.$

Tóm lại

$$\nabla_{\mathbf{w}} \ell = \nabla_{\mathbf{w}} \ell(\mathbf{w}; \mathbf{x}, y) = [\sigma(\mathbf{w}^\top \mathbf{x}) - y] \mathbf{x}. \quad (1.69)$$

Tại điểm dữ liệu $(\mathbf{x}^{(i)}, y_i)$

$$\nabla_{\mathbf{w}} \ell = \nabla_{\mathbf{w}} \ell(\mathbf{w}; \mathbf{x}^{(i)}, y_i) = [\sigma(\mathbf{w}^\top \mathbf{x}^{(i)}) - y_i] \mathbf{x}^{(i)}. \quad (1.70)$$

Hàm mất mát entropy chéo

$$L = L(\mathbf{w}; \mathbf{X}, \mathbf{Y}) = \frac{1}{m} \sum_{i=1}^m \ell(\mathbf{w}; \mathbf{x}^{(i)}, y_i) \quad (1.71)$$

$$= \frac{1}{m} \sum_{i=1}^m [-y_i \log p_i - (1 - y_i) \log(1 - p_i)].$$

trong đó $p_i = \sigma(\mathbf{w}^\top \mathbf{x}^{(i)})$.

Ta có

$$\nabla_{\mathbf{w}} L = \frac{1}{m} \sum_{i=1}^m \nabla_{\mathbf{w}} \ell(\mathbf{w}; \mathbf{x}^{(i)}, y_i) = \frac{1}{m} \sum_{i=1}^m [\sigma(\mathbf{w}^\top \mathbf{x}^{(i)}) - y_i] \mathbf{x}^{(i)} \quad (1.72)$$

$$= \frac{1}{m} [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}] [\quad (1.73)$$

$$(1.74)$$

Lưu ý rằng với cách viết ma trận ô

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(1)\top} \\ \mathbf{x}^{(2)\top} \\ \vdots \\ \mathbf{x}^{(m)\top} \end{bmatrix} \quad (1.75)$$

thì chuyển vị có dạng

$$\mathbf{X}^\top = \left[\left(\mathbf{x}^{(1)\top} \right)^\top, \left(\mathbf{x}^{(2)\top} \right)^\top, \dots, \left(\mathbf{x}^{(m)\top} \right)^\top \right] = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)} \right] \quad (1.76)$$

Ngoài ra, $\mathbf{w}^\top \mathbf{x}^{(i)}$ là đại lượng vô hướng, nên $\mathbf{w}^\top \mathbf{x}^{(i)} = \left(\mathbf{w}^\top \mathbf{x}^{(i)} \right)^\top = \mathbf{x}^{(i)\top} \left(\mathbf{w}^\top \right)^\top = \mathbf{x}^{(i)\top} \mathbf{w}$. Như vậy

$$\mathbf{X}\mathbf{w} = \begin{bmatrix} \mathbf{x}^{(1)\top} \\ \vdots \\ \mathbf{x}^{(m)\top} \end{bmatrix} \mathbf{w} = \begin{bmatrix} \mathbf{x}^{(1)\top} \mathbf{w} \\ \vdots \\ \mathbf{x}^{(m)\top} \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{w}^\top \mathbf{x}^{(1)} \\ \vdots \\ \mathbf{w}^\top \mathbf{x}^{(m)} \end{bmatrix} \quad (1.77)$$

Nếu quy ước hàm sigmoid có thể tác động lên từng phần tử của vectơ, thì

$$\sigma(\mathbf{X}\mathbf{w}) = \begin{bmatrix} \sigma(\mathbf{w}^\top \mathbf{x}^{(1)}) \\ \vdots \\ \sigma(\mathbf{w}^\top \mathbf{x}^{(m)}) \end{bmatrix} \quad (1.78)$$

Ta viết lại (1.72) dưới dạng vectơ

$$\nabla_{\tilde{\mathbf{w}}} L(\tilde{\mathbf{w}}; \tilde{\mathbf{x}}, y) = \left[\sigma(\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}) - y \right] \tilde{\mathbf{x}} \quad (1.79)$$

Tài liệu tham khảo

1. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., và Duchesnay, É. *scikit-learn Machine Learning in Python*. phiên bản 1.7.1 (2025). <https://scikit-learn.org/>.
2. Bonaccorso, G. *Machine Learning Algorithms*. In lần thứ 2. 514 trang (Packt, 2018).

