

Super VIP Cheatsheet: Học máy

Afshine AMIDI và Shervine AMIDI

Ngày 19 tháng 5 năm 2020

Mục lục

1	Học có giám sát	2	4	Mẹo và thủ thuật	11
1.1	Giới thiệu về học có giám sát	2	4.1	Độ đo phân loại	11
1.2	Các kí hiệu và khái niệm tổng quát	2	4.2	Độ đo hồi quy	11
1.3	Các mô hình tuyến tính	3	4.3	Lựa chọn model (mô hình)	12
1.3.1	Hồi quy tuyến tính	3	4.4	Dự đoán	12
1.3.2	Phân loại và logistic hồi quy	3			
1.3.3	Mô hình tuyến tính tổng quát	3	5	Refreshers	13
1.4	Máy vector hỗ trợ	3	5.1	Xác suất và thống kê	13
1.5	Generative Learning	4	5.2	Giới thiệu về Xác suất và Tổ hợp	13
1.5.1	Gaussian Discriminant Analysis	4	5.3	Xác suất có điều kiện	13
1.5.2	Naive Bayes	4	5.4	Biến ngẫu nhiên	14
1.6	Các phương thức Tree-based và ensemble	4	5.5	Phân phối đồng thời biến ngẫu nhiên	14
1.7	Các cách tiếp cận phi-tham số khác	5	5.6	Ước lượng tham số	15
1.8	Lý thuyết học	5	5.7	Đại số và vi tích phân	15
			5.8	Kí hiệu chung	15
			5.9	Các phép toán ma trận	16
			5.9.1	Phép nhân	16
			5.9.2	Một số phép toán khác	16
			5.10	Những tính chất của ma trận	16
			5.10.1	Định nghĩa	16
			5.11	Giải tích ma trận	17
2	Học không có giám sát	6			
2.1	Giới thiệu về học không giám sát	6			
2.2	Phân cụm	6			
2.2.1	Tối đa hoá kì vọng	6			
2.2.2	Phân cụm k -means	7			
2.2.3	Phân cụm phân cấp	7			
2.2.4	Các số liệu đánh giá phân cụm	7			
2.3	Giảm số chiều dữ liệu	7			
2.3.1	Phép phân tích thành phần chính	7			
2.3.2	Phân tích thành phần độc lập (ICA)	8			
3	Học sâu	9			
3.1	Mạng Neural	9			
3.2	Mạng neural tích chập (Convolutional Neural Networks)	9			
3.3	Mạng neural hồi quy (Recurrent Neural Networks)	10			
3.4	Học tăng cường (Reinforcement Learning) và điều khiển	10			

1 Học có giám sát

Dịch bởi Trần Tuấn Anh, Đàm Minh Tiến, Hung Nguyễn và Nguyễn Trí Minh

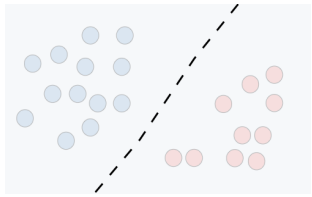
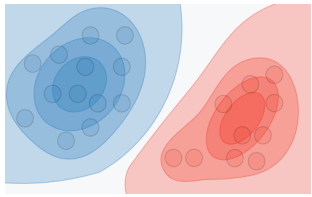
1.1 Giới thiệu về học có giám sát

Cho một tập hợp các điểm dữ liệu $\{x^{(1)}, \dots, x^{(m)}\}$ tương ứng với đó là tập các đầu ra $\{y^{(1)}, \dots, y^{(m)}\}$, chúng ta muốn xây dựng một bộ phân loại học được cách dự đoán y từ x .

□ **Loại dự đoán** – Các loại mô hình dự đoán được tổng kết trong bảng bên dưới:

	Hồi quy	Phân loại
Đầu ra	Liên tục	Lớp
Các ví dụ	Hồi quy tuyến tính	Hồi quy Logistic, SVM, Naive Bayes

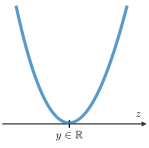
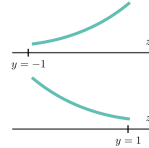
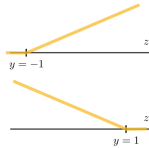
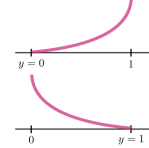
□ **Loại mô hình** – Các mô hình khác nhau được tổng kết trong bảng bên dưới:

	Mô hình phân biệt	Mô hình sinh
Mục tiêu	Ước lượng trực tiếp $P(y x)$	Ước lượng $P(x y)$ để tiếp tục suy luận $P(y x)$
Những gì học được	Biên quyết định	Phân bố xác suất của dữ liệu
Hình minh họa		
Các ví dụ	Hồi quy, SVMs	GDA, Naive Bayes

1.2 Các kí hiệu và khái niệm tổng quát

□ **Hypothesis** – Hypothesis được kí hiệu là h_θ , là một mô hình mà chúng ta chọn. Với dữ liệu đầu vào cho trước $x^{(i)}$, mô hình dự đoán đầu ra là $h_\theta(x^{(i)})$.

□ **Hàm mất mát** – Hàm mất mát là một hàm số dạng: $L : (z, y) \in \mathbb{R} \times Y \mapsto L(z, y) \in \mathbb{R}$ lấy đầu vào là giá trị dự đoán được z tương ứng với đầu ra thực tế là y , hàm có đầu ra là sự khác biệt giữa hai giá trị này. Các hàm mất mát phổ biến được tổng kết ở bảng dưới đây:

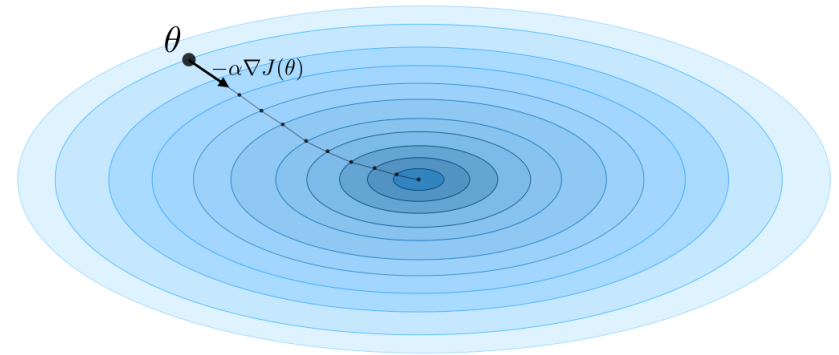
Least squared error	Mất mát Logistic	Mất mát Hinge	Cross-entropy
$\frac{1}{2}(y - z)^2$	$\log(1 + \exp(-yz))$	$\max(0, 1 - yz)$	$-[y \log(z) + (1 - y) \log(1 - z)]$
			
Hồi quy tuyến tính	Hồi quy Logistic	SVM	Mạng neural

□ **Hàm giá trị (Cost function)** – Cost function J thường được sử dụng để đánh giá hiệu năng của mô hình và được định nghĩa với hàm mất mát L như sau:

$$J(\theta) = \sum_{i=1}^m L(h_\theta(x^{(i)}), y^{(i)})$$

□ **Gradient descent** – Bằng việc kí hiệu $\alpha \in \mathbb{R}$ là tốc độ học, việc cập nhật quy tắc/ luật cho gradient descent được mô tả với tốc độ học và cost function J như sau:

$$\theta \leftarrow \theta - \alpha \nabla J(\theta)$$



Chú ý: Stochastic gradient descent (SGD) là việc cập nhật tham số dựa theo mỗi ví dụ huấn luyện, và batch gradient descent là dựa trên một lô (batch) các ví dụ huấn luyện.

□ **Likelihood** – Likelihood của một mô hình $L(\theta)$ với tham số θ được sử dụng để tìm tham số tối ưu θ thông qua việc cực đại hoá likelihood. Trong thực tế, chúng ta sử dụng log-likelihood $\ell(\theta) = \log(L(\theta))$ để dễ dàng hơn trong việc tối ưu hoá. Ta có:

$$\theta^{\text{opt}} = \arg \max_{\theta} L(\theta)$$

□ **Giải thuật Newton** – Giải thuật Newton là một phương thức số tìm θ thoả mãn điều kiện $\ell'(\theta) = 0$. Quy tắc cập nhật của nó là như sau:

$$\theta \leftarrow \theta - \frac{\ell'(\theta)}{\ell''(\theta)}$$

Chú ý: Tổng quát hoá đa chiều, còn được biết đến như là phương thức Newton-Raphson, có quy tắc cập nhật như sau:

$$\theta \leftarrow \theta - \left(\nabla_{\theta}^2 \ell(\theta) \right)^{-1} \nabla_{\theta} \ell(\theta)$$

1.3 Các mô hình tuyến tính

1.3.1 Hồi quy tuyến tính

Chúng ta giả sử ở đây rằng $y|x; \theta \sim \mathcal{N}(\mu, \sigma^2)$

□ **Phương trình chuẩn** – Bằng việc kí hiệu X là ma trận thiết kế, giá trị của θ làm cực tiểu hoá cost function là một phương pháp dạng đóng như sau:

$$\theta = (X^T X)^{-1} X^T y$$

□ **Giải thuật LMS** – Bằng việc kí hiệu α là tốc độ học, quy tắc cập nhật của giải thuật Least Mean Squares (LMS) cho tập huấn luyện của m điểm dữ liệu, còn được biết như là quy tắc học Widrow-Hoff, là như sau:

$$\forall j, \quad \theta_j \leftarrow \theta_j + \alpha \sum_{i=1}^m [y^{(i)} - h_{\theta}(x^{(i)})] x_j^{(i)}$$

Chú ý: Luật cập nhật là một trường hợp đặc biệt của gradient ascent.

□ **LWR** – Hồi quy trọng số cục bộ, còn được biết với cái tên LWR, là biến thể của hồi quy tuyến tính, nó sẽ đánh trọng số cho mỗi ví dụ huấn luyện trong cost function của nó bởi $w^{(i)}(x)$, được định nghĩa với tham số $\tau \in \mathbb{R}$ như sau:

$$w^{(i)}(x) = \exp \left(-\frac{(x^{(i)} - x)^2}{2\tau^2} \right)$$

1.3.2 Phân loại và logistic hồi quy

□ **Hàm Sigmoid** – Hàm sigmoid g , còn được biết đến như là hàm logistic, được định nghĩa như sau:

$$\forall z \in \mathbb{R}, \quad g(z) = \frac{1}{1 + e^{-z}} \in]0, 1[$$

□ **Hồi quy logistic** – Chúng ta giả sử ở đây rằng $y|x; \theta \sim \text{Bernoulli}(\phi)$. Ta có công thức như sau:

$$\phi = p(y = 1|x; \theta) = \frac{1}{1 + \exp(-\theta^T x)} = g(\theta^T x)$$

Chú ý: không có giải pháp dạng đóng cho trường hợp của hồi quy logistic.

□ **Hồi quy Softmax** – Hồi quy softmax, còn được gọi là hồi quy logistic đa lớp, được sử dụng để tổng quát hoá hồi quy logistic khi có nhiều hơn 2 lớp đầu ra. Theo quy ước, chúng ta thiết lập $\theta_K = 0$, làm cho tham số Bernoulli ϕ_i của mỗi lớp i bằng với:

$$\phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^K \exp(\theta_j^T x)}$$

1.3.3 Mô hình tuyến tính tổng quát

□ **Họ số mũ** – Một lớp của phân phối được cho rằng thuộc về họ số mũ nếu nó có thể được viết dưới dạng một thuật ngữ của tham số tự nhiên, cũng được gọi là tham số kinh điển (canonical parameter) hoặc hàm kết nối, η , một số liệu thống kê đầy đủ $T(y)$ và hàm phân vùng log (log-partition function) $a(\eta)$ sẽ có dạng như sau:

$$p(y; \eta) = b(y) \exp(\eta T(y) - a(\eta))$$

Chú ý: chúng ta thường có $T(y) = y$. Đồng thời, $\exp(-a(\eta))$ có thể được xem như là tham số chuẩn hoá sẽ đảm bảo rằng tổng các xác suất là một.

Ở đây là các phân phối mũ phổ biến nhất được tổng kết ở bảng bên dưới:

Phân phối	η	$T(y)$	$a(\eta)$	$b(y)$
Bernoulli	$\log \left(\frac{\phi}{1-\phi} \right)$	y	$\log(1 + \exp(\eta))$	1
Gaussian	μ	y	$\frac{\eta^2}{2}$	$\frac{1}{\sqrt{2\pi}} \exp \left(-\frac{y^2}{2} \right)$
Poisson	$\log(\lambda)$	y	e^{η}	$\frac{1}{y!}$
Geometric	$\log(1 - \phi)$	y	$\log \left(\frac{e^{\eta}}{1 - e^{\eta}} \right)$	1

□ **Giả thuyết GLMs** – Mô hình tuyến tính tổng quát (GLM) với mục đích là dự đoán một biến ngẫu nhiên y như là hàm cho biến $x \in \mathbb{R}^{n+1}$ và dựa trên 3 giả thuyết sau:

$$(1) \quad y|x; \theta \sim \text{ExpFamily}(\eta) \quad (2) \quad h_{\theta}(x) = E[y|x; \theta] \quad (3) \quad \eta = \theta^T x$$

Chú ý: Bình phương nhỏ nhất thông thường và hồi quy logistic đều là các trường hợp đặc biệt của các mô hình tuyến tính tổng quát.

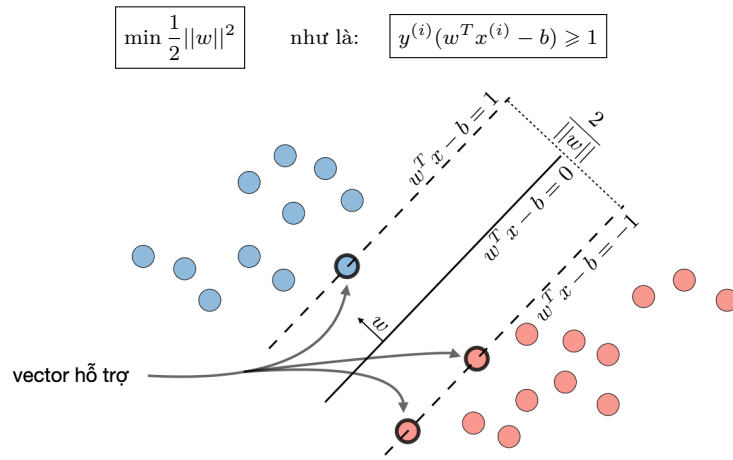
1.4 Máy vector hỗ trợ

Mục tiêu của máy vector hỗ trợ là tìm ra dòng tối đa hoá khoảng cách nhỏ nhất tới dòng.

□ **Optimal margin classifier** – Optimal margin classifier h là như sau:

$$h(x) = \text{sign}(w^T x - b)$$

với $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ là giải pháp cho vấn đề tối ưu hoá sau đây:



Chú ý: đường thẳng có phương trình là $w^T x - b = 0$.

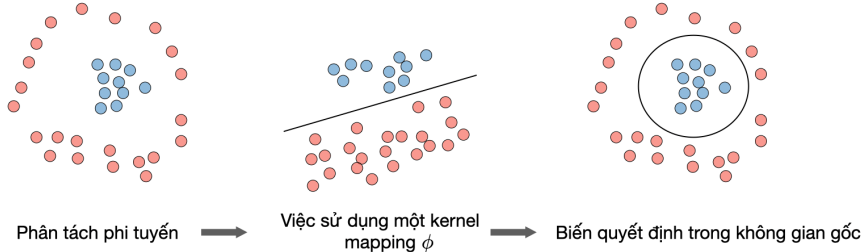
□ **Mất mát Hinge** – Mất mát Hinge được sử dụng trong thiết lập của SVMs và nó được định nghĩa như sau:

$$L(z, y) = [1 - yz]_+ = \max(0, 1 - yz)$$

□ **Kernel (nhân)** – Cho trước feature mapping ϕ , chúng ta định nghĩa kernel K như sau:

$$K(x, z) = \phi(x)^T \phi(z)$$

Trong thực tế, kernel K được định nghĩa bởi $K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$ được gọi là Gaussian kernel và thường được sử dụng.



Chú ý: chúng ta nói rằng chúng ta sử dụng "kernel trick" để tính toán cost function sử dụng kernel bởi vì chúng ta thực sự không cần biết đến ánh xạ tường minh ϕ , nó thường khá phức tạp. Thay vào đó, chỉ cần biết giá trị $K(x, z)$.

□ **Lagrangian** – Chúng ta định nghĩa Lagrangian $\mathcal{L}(w, b)$ như sau:

$$\mathcal{L}(w, b) = f(w) + \sum_{i=1}^l \beta_i h_i(w)$$

Chú ý: hệ số β_i được gọi là bội số Lagrange.

1.5 Generative Learning

Một mô hình sinh đầu tiên cố gắng học cách dữ liệu được sinh ra thông qua việc ước lượng $P(x|y)$, sau đó chúng ta có thể sử dụng $P(x|y)$ để ước lượng $P(y|x)$ bằng cách sử dụng luật Bayes.

1.5.1 Gaussian Discriminant Analysis

□ **Thiết lập** – Gaussian Discriminant Analysis giả sử rằng y và $x|y = 0$ và $x|y = 1$ là như sau:

$$y \sim \text{Bernoulli}(\phi)$$

$$x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma) \quad \text{và} \quad x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$$

□ **Sự ước lượng** – Bảng sau đây tổng kết các ước lượng mà chúng ta tìm thấy khi tối đa hoá likelihood:

$\hat{\phi}$	$\hat{\mu}_j \quad (j = 0, 1)$	$\hat{\Sigma}$
$\frac{1}{m} \sum_{i=1}^m 1_{\{y^{(i)}=1\}}$	$\frac{\sum_{i=1}^m 1_{\{y^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{y^{(i)}=j\}}}$	$\frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$

1.5.2 Naive Bayes

□ **Giả thiết** – Mô hình Naive Bayes giả sử rằng các features của các điểm dữ liệu đều độc lập với nhau:

$$P(x|y) = P(x_1, x_2, \dots | y) = P(x_1|y)P(x_2|y) \dots = \prod_{i=1}^n P(x_i|y)$$

□ **Giải pháp** – Tối đa hoá log-likelihood đưa ra những lời giải sau đây, với $k \in \{0, 1\}, l \in \llbracket 1, L \rrbracket$

$$P(y = k) = \frac{1}{m} \times \#\{j | y^{(j)} = k\} \quad \text{và} \quad P(x_i = l | y = k) = \frac{\#\{j | y^{(j)} = k \text{ và } x_i^{(j)} = l\}}{\#\{j | y^{(j)} = k\}}$$

Chú ý: Naive Bayes được sử dụng rộng rãi cho bài toán phân loại văn bản và phát hiện spam.

1.6 Các phương thức Tree-based và ensemble

Các phương thức này có thể được sử dụng cho cả bài toán hồi quy lẫn bài toán phân loại.

□ **CART** – Cây phân loại và hồi quy (CART), thường được biết đến là cây quyết định, có thể được biểu diễn dưới dạng cây nhị phân. Chúng có các ưu điểm có thể được diễn giải một cách dễ dàng.

□ **Rừng ngẫu nhiên** – Là một kĩ thuật dựa trên cây (tree-based), sử dụng số lượng lớn các cây quyết định để lựa chọn ngẫu nhiên các tập thuộc tính. Ngược lại với một cây quyết định đơn, kĩ thuật này khá khó diễn giải nhưng do có hiệu năng tốt nên đã trở thành một giải thuật khá phổ biến hiện nay.

Chú ý: rừng ngẫu nhiên là một loại giải thuật ensemble.

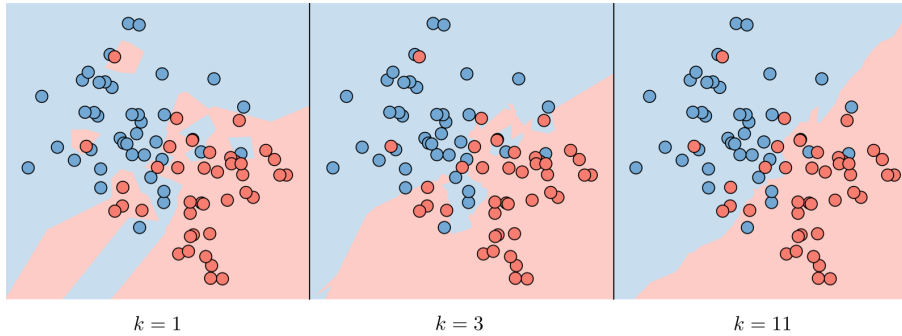
□ **Boosting** – Ý tưởng của các phương thức boosting là kết hợp các phương pháp học yếu hơn để tạo nên phương pháp học mạnh hơn. Những phương thức chính được tổng kết ở bảng dưới đây:

Adaptive boosting	Gradient boosting
<ul style="list-style-type: none"> - Các trọng số có giá trị lớn được đặt vào các phần lỗi để cải thiện ở bước boosting tiếp theo - "Adaboost" 	<ul style="list-style-type: none"> - Các phương pháp học yếu huấn luyện trên các phần lỗi còn lại

1.7 Các cách tiếp cận phi-tham số khác

□ **k -nearest neighbors** – Giải thuật k -nearest neighbors, thường được biết đến là k -NN, là cách tiếp cận phi-tham số, ở phương pháp này phân lớp của một điểm dữ liệu được định nghĩa bởi k điểm dữ liệu gần nó nhất trong tập huấn luyện. Phương pháp này có thể được sử dụng trong quá trình thiết lập cho bài toán phân loại cũng như bài toán hồi quy.

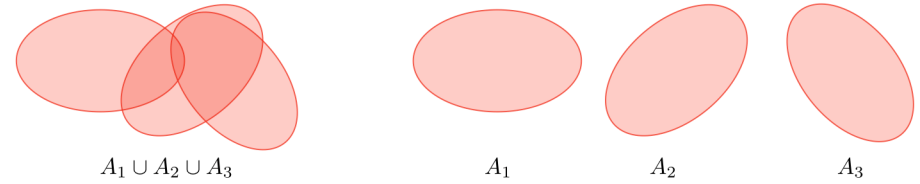
Chú ý: Tham số k cao hơn, độ chệch (bias) cao hơn, tham số k thấp hơn, phương sai cao hơn.



1.8 Lý thuyết học

□ **Union bound** – Cho k sự kiện là A_1, \dots, A_k . Ta có:

$$P(A_1 \cup \dots \cup A_k) \leq P(A_1) + \dots + P(A_k)$$



□ **Bất đẳng thức Hoeffding** – Cho Z_1, \dots, Z_m là m biến iid được đưa ra từ phân phối Bernoulli của tham số ϕ . Cho $\hat{\phi}$ là trung bình mẫu của chúng và $\gamma > 0$ cố định. Ta có:

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2 \exp(-2\gamma^2 m)$$

Chú ý: bất đẳng thức này còn được biết đến như là ràng buộc Chernoff.

□ **Lỗi huấn luyện (Training error)** – Cho trước classifier h , ta định nghĩa training error $\hat{\epsilon}(h)$, còn được biết đến là empirical risk hoặc empirical error, như sau:

$$\hat{\epsilon}(h) = \frac{1}{m} \sum_{i=1}^m 1_{\{h(x^{(i)}) \neq y^{(i)}\}}$$

□ **Probably Approximately Correct (PAC)** – PAC là một framework với nhiều kết quả về lí thuyết học đã được chứng minh, và có tập hợp các giả thiết như sau:

- tập huấn luyện và test có cùng phân phối
- các ví dụ huấn luyện được tạo ra độc lập

□ **Shattering (Chia nhỏ)** – Cho một tập hợp $S = \{x^{(1)}, \dots, x^{(d)}\}$, và một tập hợp các classifiers \mathcal{H} , ta nói rằng \mathcal{H} chia nhỏ S nếu với bất kì tập các nhãn $\{y^{(1)}, \dots, y^{(d)}\}$ nào, ta có:

$$\exists h \in \mathcal{H}, \quad \forall i \in \llbracket 1, d \rrbracket, \quad h(x^{(i)}) = y^{(i)}$$

□ **Định lí giới hạn trên** – Cho \mathcal{H} là một finite hypothesis class mà $|\mathcal{H}| = k$ với δ , kích cỡ m là cố định. Khi đó, với xác suất nhỏ nhất là $1 - \delta$, ta có:

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + 2 \sqrt{\frac{1}{2m} \log \left(\frac{2k}{\delta} \right)}$$

□ **VC dimension** – Vapnik-Chervonenkis (VC) dimension của class infinite hypothesis \mathcal{H} cho trước, kí hiệu là $VC(\mathcal{H})$ là kích thước của tập lớn nhất được chia nhỏ bởi \mathcal{H} .

Chú ý: VC dimension của $\mathcal{H} = \{\text{tập hợp các linear classifiers trong 2 chiều}\}$ là 3.



□ **Định lý (Vapnik)** – Cho \mathcal{H} với $VC(\mathcal{H}) = d$ và m là số lượng các ví dụ huấn luyện. Với xác suất nhỏ nhất là $1 - \delta$, ta có:

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + O \left(\sqrt{\frac{d}{m} \log \left(\frac{m}{d} \right)} + \frac{1}{m} \log \left(\frac{1}{\delta} \right) \right)$$

2 Học không có giám sát

Dịch bởi Trần Tuấn Anh và Đàm Minh Tiến

2.1 Giới thiệu về học không giám sát

□ **Động lực** – Mục tiêu của học không giám sát là tìm được quy luật ẩn (hidden pattern) trong tập dữ liệu không được gán nhãn $\{x^{(1)}, \dots, x^{(m)}\}$.

□ **Bất đẳng thức Jensen** – Cho f là một hàm lồi và X là một biến ngẫu nhiên. Chúng ta có bất đẳng thức sau:

$$E[f(X)] \geq f(E[X])$$

2.2 Phân cụm

2.2.1 Tối đa hoá kì vọng

□ **Các biến Latent** – Các biến Latent là các biến ẩn/ không thấy được khiến cho việc dự đoán trở nên khó khăn, và thường được kí hiệu là z . Đây là các thiết lập phổ biến mà các biến latent thường có:

Thiết lập	Biến Latent z	$x z$	Các bình luận
Sự kết hợp của k Gaussians	Multinomial(ϕ)	$\mathcal{N}(\mu_j, \Sigma_j)$	$\mu_j \in \mathbb{R}^n, \phi \in \mathbb{R}^k$
Phân tích hệ số	$\mathcal{N}(0, I)$	$\mathcal{N}(\mu + \Lambda z, \psi)$	$\mu_j \in \mathbb{R}^n$

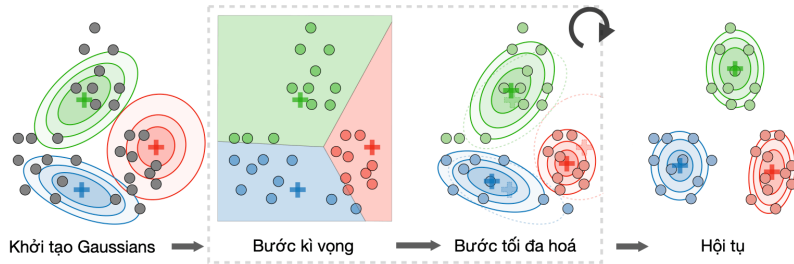
□ **Thuật toán** – Thuật toán tối đa hoá kì vọng (EM) mang lại một phương thức có hiệu quả trong việc ước lượng tham số θ thông qua tối đa hoá giá trị ước lượng likelihood bằng cách lặp lại việc tạo nên một cận dưới cho likelihood (E-step) và tối ưu hoá cận dưới (M-step) như sau:

- E-step: Đánh giá xác suất hậu nghiệm $Q_i(z^{(i)})$ cho mỗi điểm dữ liệu $x^{(i)}$ đến từ một cụm $z^{(i)}$ cụ thể như sau:

$$Q_i(z^{(i)}) = P(z^{(i)} | x^{(i)}; \theta)$$

- M-step: Sử dụng xác suất hậu nghiệm $Q_i(z^{(i)})$ như các trọng số cụ thể của cụm trên các điểm dữ liệu $x^{(i)}$ để ước lượng lại một cách riêng biệt cho mỗi mô hình cụm như sau:

$$\theta_i = \operatorname{argmax}_{\theta} \sum_i \int_{z^{(i)}} Q_i(z^{(i)}) \log \left(\frac{P(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right) dz^{(i)}$$

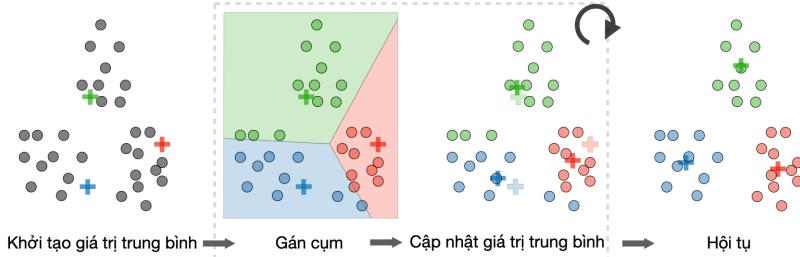


2.2.2 Phân cụm k-means

Chúng ta kí hiệu $c^{(i)}$ là cụm của điểm dữ liệu i và μ_j là điểm trung tâm của cụm j .

□ **Thuật toán** – Sau khi khởi tạo ngẫu nhiên các tâm cụm (centroids) $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$, thuật toán k -means lặp lại bước sau cho đến khi hội tụ:

$$c^{(i)} = \arg \min_j \|x^{(i)} - \mu_j\|^2 \quad \text{và} \quad \mu_j = \frac{\sum_{i=1}^m 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{c^{(i)}=j\}}}$$



□ **Hàm Distortion** – Để nhận biết khi nào thuật toán hội tụ, chúng ta sẽ xem xét hàm distortion được định nghĩa như sau:

$$J(c, \mu) = \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

2.2.3 Phân cụm phân cấp

□ **Thuật toán** – Là một thuật toán phân cụm với cách tiếp cận phân cấp kết tập, cách tiếp cận này sẽ xây dựng các cụm lồng nhau theo một quy tắc nối tiếp.

□ **Các loại** – Các loại thuật toán hierarchical clustering khác nhau với mục tiêu là tối ưu hoá các hàm đối tượng khác nhau sẽ được tổng kết trong bảng dưới đây:

Liên kết Ward	Liên kết trung bình	Liên kết hoàn chỉnh
Tối thiểu hoá trong phạm vi khoảng cách của một cụm	Tối thiểu hoá khoảng cách trung bình giữa các cặp cụm	Tối thiểu hoá khoảng cách tối đa giữa các cặp cụm

2.2.4 Các số liệu đánh giá phân cụm

Trong quá trình thiết lập học không giám sát, sẽ khá khó khăn để đánh giá hiệu năng của một mô hình vì chúng ta không có các nhãn đủ tin cậy như trong trường hợp của học có giám sát.

□ **Hệ số Silhouette** – Bằng việc kí hiệu a và b là khoảng cách trung bình giữa một điểm mẫu với các điểm khác trong cùng một lớp, và giữa một điểm mẫu với các điểm khác thuộc cụm kế cận gần nhất, hệ số silhouette s đối với một điểm mẫu đơn được định nghĩa như sau:

$$s = \frac{b - a}{\max(a, b)}$$

□ **Chỉ số Calinski-Harabaz** – Bằng việc kí hiệu k là số cụm, các chỉ số B_k và W_k về độ phân tán giữa và trong một cụm lần lượt được định nghĩa như là

$$B_k = \sum_{j=1}^k n_{c(j)} (\mu_{c(j)} - \mu)(\mu_{c(j)} - \mu)^T, \quad W_k = \sum_{i=1}^m (x^{(i)} - \mu_{c(i)})(x^{(i)} - \mu_{c(i)})^T$$

Chỉ số Calinski-Harabaz $s(k)$ cho biết khả năng phân cụm tốt đến đâu của một mô hình phân cụm, ví dụ như với score cao hơn thì sẽ dày đặc hơn và việc phân cụm tốt hơn. Nó được định nghĩa như sau:

$$s(k) = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{N - k}{k - 1}$$

2.3 Giảm số chiều dữ liệu

2.3.1 Phép phân tích thành phần chính

Là một kĩ thuật giảm số chiều dữ liệu, kĩ thuật này sẽ tìm các hướng tối đa hoá phương sai để chiếu dữ liệu lên trên đó.

□ **Giá trị riêng, vector riêng** – Cho ma trận $A \in \mathbb{R}^{n \times n}$, λ là giá trị riêng của A nếu tồn tại một vector $z \in \mathbb{R}^n \setminus \{0\}$, gọi là vector riêng, như vậy ta có:

$$Az = \lambda z$$

□ **Định lý Spectral** – Với $A \in \mathbb{R}^{n \times n}$. Nếu A đối xứng thì A có thể chéo hoá bởi một ma trận trực giao $U \in \mathbb{R}^{n \times n}$. Bằng việc kí hiệu $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, ta có:

$$\exists \Lambda \text{ đường chéo, } A = U\Lambda U^T$$

Chú thích: vector riêng tương ứng với giá trị riêng lớn nhất được gọi là vector riêng chính của ma trận A .

□ **Thuật toán** – Phép phân tích thành phần chính (Principal Component Analysis, PCA) là một kĩ thuật giảm số chiều dữ liệu, nó sẽ chiếu dữ liệu lên k chiều bằng cách tối đa hoá phương sai của dữ liệu như sau:

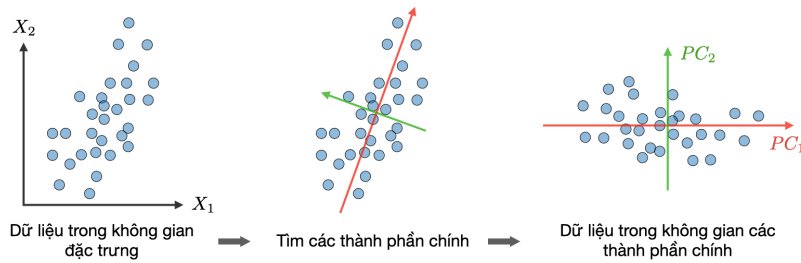
- **Bước 1:** Chuẩn hoá dữ liệu để có giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1.

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j} \quad \text{where} \quad \mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)} \quad \text{và} \quad \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$$

Vì thế, quy tắc học của stochastic gradient ascent là với mỗi ví dụ huấn luyện $x^{(i)}$, chúng ta sẽ cập nhật W như sau:

$$W \leftarrow W + \alpha \left(\begin{pmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{pmatrix} x^{(i)T} + (W^T)^{-1} \right)$$

- **Bước 2:** Tính $\Sigma = \frac{1}{m} \sum_{i=1}^m x^{(i)} x^{(i)T} \in \mathbb{R}^{n \times n}$, là đối xứng với các giá trị riêng thực.
- **Bước 3:** Tính $u_1, \dots, u_k \in \mathbb{R}^n$ là k vector riêng trực giao của Σ , tức các vector trực giao riêng của k giá trị riêng lớn nhất.
- **Bước 4:** Chiếu dữ liệu lên $\text{span}_{\mathbb{R}}(u_1, \dots, u_k)$.



2.3.2 Phân tích thành phần độc lập (ICA)

Là một kĩ thuật tìm các nguồn tạo cơ bản.

□ **Giả định** – Chúng ta giả sử rằng dữ liệu x được tạo ra bởi vector nguồn n -chiều $s = (s_1, \dots, s_n)$, với s_i là các biến ngẫu nhiên độc lập, thông qua một ma trận mixing và non-singular A như sau:

$$x = As$$

Mục tiêu là tìm ma trận unmixing $W = A^{-1}$.

□ **Giải thuật Bell và Sejnowski ICA** – Giải thuật này tìm ma trận unmixing W bằng các bước dưới đây:

- Ghi xác suất của $x = As = W^{-1}s$ như là:

$$p(x) = \prod_{i=1}^n p_s(w_i^T x) \cdot |W|$$

- Ghi log likelihood cho dữ liệu huấn luyện $\{x^{(i)}, i \in [1, m]\}$ và kí hiệu g là hàm sigmoid như sau:

$$l(W) = \sum_{i=1}^m \left(\sum_{j=1}^n \log \left(g'(w_j^T x^{(i)}) \right) + \log |W| \right)$$

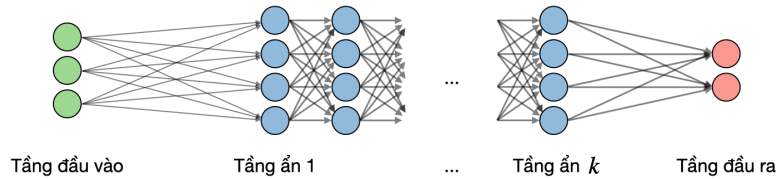
3 Học sâu

Dịch bởi Trần Tuấn Anh, Phạm Hồng Vinh, Đàm Minh Tiến, Nguyễn Khánh Hưng, Hoàng Vũ Đạt và Nguyễn Trí Minh

3.1 Mạng Neural

Mạng Neural là 1 lớp của các mô hình (models) được xây dựng với các tầng (layers). Các loại mạng Neural thường được sử dụng bao gồm: Mạng Neural tích chập (Convolutional Neural Networks) và Mạng Neural hồi quy (Recurrent Neural Networks).

□ **Kiến trúc** – Các thuật ngữ xoay quanh kiến trúc của mạng neural được mô tả như hình phía dưới:



Bằng việc kí hiệu i là tầng thứ i của mạng, j là hidden unit (đơn vị ẩn) thứ j của tầng, ta có:

$$z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]}$$

chúng ta kí hiệu w, b, z tương ứng với trọng số (weights), bias và đầu ra.

□ **Hàm kích hoạt (Activation function)** – Hàm kích hoạt được sử dụng ở phần cuối của đơn vị ẩn để đưa ra độ phức tạp phi tuyến tính (non-linear) cho mô hình (model). Đây là những trường hợp phổ biến nhất:

Sigmoid	Tanh	ReLU	Leaky ReLU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ với $\epsilon \ll 1$

□ **Lỗi (loss) Cross-entropy** – Trong bối cảnh của mạng neural, hàm lỗi cross-entropy $L(z, y)$ thường được sử dụng và định nghĩa như sau:

$$L(z, y) = - \left[y \log(z) + (1 - y) \log(1 - z) \right]$$

□ **Tốc độ học (Learning rate)** – Tốc độ học, thường được kí hiệu bởi α hoặc đôi khi là η , chỉ ra tốc độ mà trọng số được cập nhật. Thông số này có thể là cố định hoặc được thay đổi tùy biến. Phương thức (method) phổ biến nhất hiện tại là Adam, đó là phương thức thay đổi tốc độ học một cách phù hợp nhất có thể.

□ **Backpropagation (Lan truyền ngược)** – Backpropagation là phương thức dùng để cập nhật trọng số trong mạng neural bằng cách tính toán đầu ra thực sự và đầu ra mong muốn. Đạo hàm theo trọng số w được tính bằng cách sử dụng quy tắc chuỗi (chain rule) theo như cách dưới đây:

$$\frac{\partial L(z, y)}{\partial w} = \frac{\partial L(z, y)}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w}$$

Như kết quả, trọng số được cập nhật như sau:

$$w \leftarrow w - \eta \frac{\partial L(z, y)}{\partial w}$$

□ **Cập nhật trọng số** – Trong mạng neural, trọng số được cập nhật như sau:

- **Bước 1:** Lấy một mẻ (batch) dữ liệu huấn luyện (training data).
- **Bước 2:** Thực thi lan truyền tiến (forward propagation) để lấy được lỗi (loss) tương ứng.
- **Bước 3:** Lan truyền ngược lỗi để lấy được gradients (độ dốc).
- **Bước 4:** Sử dụng gradients để cập nhật trọng số của mạng (network).

□ **Dropout** – Dropout là thuật ngữ kĩ thuật dùng trong việc tránh overfitting tập dữ liệu huấn luyện bằng việc bỏ đi các đơn vị trong mạng neural. Trong thực tế, các neurons hoặc là bị bỏ đi bởi xác suất p hoặc được giữ lại với xác suất $1 - p$.

3.2 Mạng neural tích chập (Convolutional Neural Networks)

□ **Yêu cầu của tầng tích chập (Convolutional layer)** – Bằng việc ghi chú W là kích cỡ của volume đầu vào, F là kích cỡ của neurons thuộc convolutional layer, P là số lượng zero padding, khi đó số lượng neurons N phù hợp với volume cho trước sẽ như sau:

$$N = \frac{W - F + 2P}{S} + 1$$

□ **Batch normalization (chuẩn hoá)** – Đây là bước mà các hyperparameter γ, β chuẩn hoá batch $\{x_i\}$. Bằng việc kí hiệu μ_B, σ_B^2 là giá trị trung bình, phương sai mà ta muốn gán cho batch, nó được thực hiện như sau:

$$x_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

Nó thường được tính sau fully connected/convolutional layer và trước non-linearity layer và mục tiêu là cho phép tốc độ học cao hơn cũng như giảm đi sự phụ thuộc mạnh mẽ vào việc khởi tạo.

3.3 Mạng neural hồi quy (Recurrent Neural Networks)

□ **Các loại cổng** – Đây là các loại cổng (gate) khác nhau mà chúng ta sẽ gặp ở một mạng neural hồi quy điển hình:

Cổng đầu vào	cổng quên	cổng đầu ra	cổng
Ghi vào cell hay không?	Xoá cell hay không?	Cần tiết lộ bao nhiêu về cell?	Ghi bao nhiêu vào cell?

□ **LSTM** – Mạng bộ nhớ dài-ngắn (LSTM) là 1 loại RNN model tránh vấn đề vanishing gradient (gradient biến mất đột ngột) bằng cách thêm vào cổng 'quên' ('forget' gates).

3.4 Học tăng cường (Reinforcement Learning) và điều khiển

Mục tiêu của học tăng cường đó là cho tác tử (agent) học cách làm sao để tối ưu hoá trong một môi trường.

□ **Tiến trình quyết định Markov (Markov decision processes)** – Tiến trình quyết định Markov (MDP) là một dạng 5-tuple $(\mathcal{S}, \mathcal{A}, \{P_{sa}\}, \gamma, R)$ mà ở đó:

- \mathcal{S} là tập hợp các trạng thái (states)
- \mathcal{A} là tập hợp các hành động (actions)
- $\{P_{sa}\}$ là xác suất chuyển tiếp trạng thái cho $s \in \mathcal{S}$ và $a \in \mathcal{A}$
- $\gamma \in [0, 1]$ là discount factor
- $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ hoặc $R : \mathcal{S} \rightarrow \mathbb{R}$ là reward function (hàm định nghĩa phần thưởng) mà giải thuật muốn tối đa hoá

□ **Policy** – Policy π là 1 hàm $\pi : \mathcal{S} \rightarrow \mathcal{A}$ có nhiệm vụ ánh xạ states tới actions.

Chú ý: Ta quy ước rằng ta thực thi policy π cho trước nếu cho trước state s ta có action $a = \pi(s)$.

□ **Hàm giá trị (Value function)** – Với policy cho trước π và state s , ta định nghĩa value function V^π như sau:

$$V^\pi(s) = E \left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots | s_0 = s, \pi \right]$$

□ **Phương trình Bellman** – Phương trình tối ưu Bellman đặc trưng hoá value function V^{π^*} của policy tối ưu (optimal policy) π^* :

$$V^{\pi^*}(s) = R(s) + \max_{a \in \mathcal{A}} \gamma \sum_{s' \in \mathcal{S}} P_{sa}(s') V^{\pi^*}(s')$$

Chú ý: ta quy ước optimal policy π^ đối với state s cho trước như sau:*

$$\pi^*(s) = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{sa}(s') V^*(s')$$

□ **Giải thuật duyệt giá trị (Value iteration)** – Giải thuật duyệt giá trị gồm 2 bước:

- 1) Ta khởi tạo giá trị:

$$V_0(s) = 0$$

- 2) Ta duyệt qua giá trị dựa theo giá trị phía trước:

$$V_{i+1}(s) = R(s) + \max_{a \in \mathcal{A}} \left[\sum_{s' \in \mathcal{S}} \gamma P_{sa}(s') V_i(s') \right]$$

□ **Ước lượng khả năng tối đa (Maximum likelihood estimate)** – Ước lượng khả năng tối đa cho xác suất chuyển tiếp trạng thái (state) sẽ như sau:

$$P_{sa}(s') = \frac{\# \text{thời gian hành động } a \text{ tiêu tốn cho state } s \text{ và biến đổi nó thành } s'}{\# \text{thời gian hành động } a \text{ tiêu tốn cho state (trạng thái) } s}$$

□ **Q-learning** – Q-learning là 1 dạng phán đoán phi mô hình (model-free) của Q, được thực hiện như sau:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[R(s, a, s') + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

4 Mẹo và thủ thuật

Dịch bởi Trần Tuấn Anh, Nguyễn Trí Minh, Vinh Phạm và Đàm Minh Tiến

4.1 Độ đo phân loại

Đối với phân loại nhị phân (binary classification) là các độ đo chính, chúng khá quan trọng để theo dõi (track), qua đó đánh giá hiệu năng của mô hình (model).

□ **Ma trận nhầm lẫn (Confusion matrix)** – Confusion matrix được sử dụng để có kết quả hoàn chỉnh hơn khi đánh giá hiệu năng của model. Nó được định nghĩa như sau:

		Lớp dự đoán	
		+	-
lớp thực sự	+	TP True Positives	FN False Negatives Type II error
	-	FP False Positives Type I error	TN True Negatives

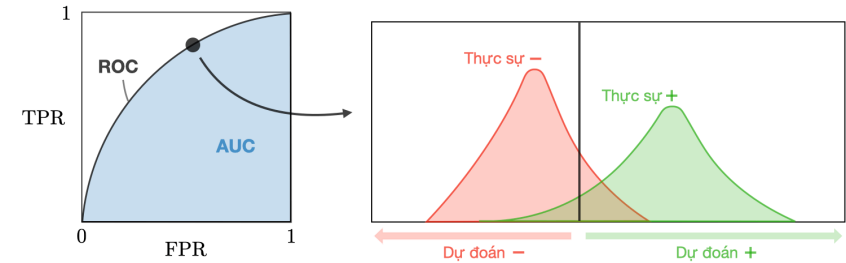
□ **Độ đo chính** – Các độ đo sau thường được sử dụng để đánh giá hiệu năng của mô hình phân loại:

Độ đo	Công thức	Diễn giải
chính xác	$\frac{TP + TN}{TP + TN + FP + FN}$	Hiệu năng tổng thể của mô hình
Precision	$\frac{TP}{TP + FP}$	Độ chính xác của các dự đoán positive
Recall Sensitivity	$\frac{TP}{TP + FN}$	Bao phủ các mẫu thử chính xác (positive) thực sự
Specificity	$\frac{TN}{TN + FP}$	Bao phủ các mẫu thử sai (negative) thực sự
Điểm F1	$\frac{2TP}{2TP + FP + FN}$	Độ đo Hybrid hữu ích cho các lớp không cân bằng (unbalanced classes)

□ **ROC** – Đường cong thao tác nhận, được kí hiệu là ROC, là minh họa của TPR với FPR bằng việc thay đổi ngưỡng (threshold). Các độ đo này được tổng kết ở bảng bên dưới:

Độ đo	Công thức	Tương đương
True Positive Rate TPR	$\frac{TP}{TP + FN}$	Recall, sensitivity
False Positive Rate FPR	$\frac{FP}{TN + FP}$	1-specificity

□ **AUC** – Khu vực phía dưới đường cong thao tác nhận, còn được gọi tắt là AUC hoặc AUROC, là khu vực phía dưới ROC như hình minh họa phía dưới:



4.2 Độ đo hồi quy

□ **Độ đo cơ bản** – Cho trước mô hình hồi quy f , độ đo sau được sử dụng phổ biến để đánh giá hiệu năng của mô hình:

Tổng của tổng các bình phương	Mô hình tổng bình phương	Tổng bình phương dư
$SS_{\text{tot}} = \sum_{i=1}^m (y_i - \bar{y})^2$	$SS_{\text{reg}} = \sum_{i=1}^m (f(x_i) - \bar{y})^2$	$SS_{\text{res}} = \sum_{i=1}^m (y_i - f(x_i))^2$

□ **Hệ số quyết định** – Hệ số quyết định, thường được kí hiệu là R^2 hoặc r^2 , cung cấp độ đo mức độ tốt của kết quả quan sát đầu ra (được nhân rộng bởi mô hình), và được định nghĩa như sau:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

□ **Độ đo chính** – Độ đo sau đây thường được sử dụng để đánh giá hiệu năng của mô hình hồi quy, bằng cách tính số lượng các biến n mà độ đo đó sẽ cân nhắc:

Mallow's Cp	AIC	BIC	Adjusted R^2
$\frac{SS_{\text{res}} + 2(n+1)\hat{\sigma}^2}{m}$	$2[(n+2) - \log(L)]$	$\log(m)(n+2) - 2\log(L)$	$1 - \frac{(1-R^2)(m-1)}{m-n-1}$

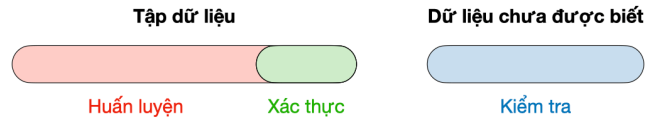
trong đó L là khả năng và $\hat{\sigma}^2$ là giá trị ước tính của phương sai tương ứng với mỗi response (hồi đáp).

4.3 Lựa chọn model (mô hình)

□ **Vocabulary** – Khi lựa chọn mô hình, chúng ta chia tập dữ liệu thành 3 tập con như sau:

Tập huấn luyện	Tập xác thực	Tập kiểm tra (testing)
<ul style="list-style-type: none"> Mô hình được huấn luyện Thường là 80 tập dữ liệu 	<ul style="list-style-type: none"> mô hình được xác thực Thường là 20% tập dữ liệu Cũng được gọi là hold-out hoặc development set (tập phát triển) 	<ul style="list-style-type: none"> mô hình đưa ra dự đoán Dữ liệu chưa được biết

Khi mô hình đã được chọn, nó sẽ được huấn luyện trên tập dữ liệu đầu vào và được test trên tập dữ liệu test hoàn toàn khác. Tất cả được minh hoạ ở hình bên dưới:



□ **Cross-validation** – Cross-validation, còn được gọi là CV, một phương thức được sử dụng để chọn ra một mô hình không dựa quá nhiều vào tập dữ liệu huấn luyện ban đầu. Các loại khác nhau được tổng kết ở bảng bên dưới:

<i>k</i> -fold	Leave- <i>p</i> -out
<ul style="list-style-type: none"> Huấn luyện trên $k - 1$ phần và đánh giá trên 1 phần còn lại Thường thì $k = 5$ hoặc 10 	<ul style="list-style-type: none"> Huấn luyện trên $n - p$ phần và đánh giá trên p phần còn lại Trường hợp $p = 1$ được gọi là leave-one-out

Phương thức hay được sử dụng được gọi là *k*-fold cross-validation và chia dữ liệu huấn luyện thành k phần, đánh giá mô hình trên 1 phần trong khi huấn luyện mô hình trên $k - 1$ phần còn lại, tất cả k lần. Lỗi sau đó được tính trung bình trên k phần và được đặt tên là cross-validation error.

Phần	Tập dữ liệu	Lỗi xác thực	Lỗi cross-xác thực
1		ϵ_1	$\frac{\epsilon_1 + \dots + \epsilon_k}{k}$
2		ϵ_2	
\vdots	\vdots	\vdots	
k		ϵ_k	
	Huấn luyện (red) Xác thực (green)		

□ **Chuẩn hoá** – Mục đích của thủ tục chuẩn hoá là tránh cho mô hình bị overfit với dữ liệu, do đó gặp phải vấn đề phương sai lớn. Bảng sau đây sẽ tổng kết các loại kĩ thuật chuẩn hoá khác nhau hay được sử dụng:

LASSO	Ridge	Elastic Net
<ul style="list-style-type: none"> Giảm hệ số xuống còn 0 Tốt cho việc lựa chọn biến 	Làm cho hệ số nhỏ hơn	Thay đổi giữa chọn biến và hệ số nhỏ hơn
$\dots + \lambda \theta _1$ $\lambda \in \mathbb{R}$	$\dots + \lambda \theta _2^2$ $\lambda \in \mathbb{R}$	$\dots + \lambda \left[(1 - \alpha) \theta _1 + \alpha \theta _2^2 \right]$ $\lambda \in \mathbb{R}, \alpha \in [0, 1]$

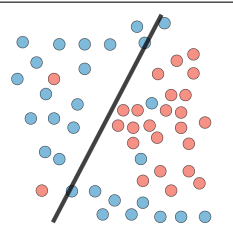
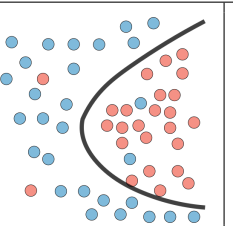
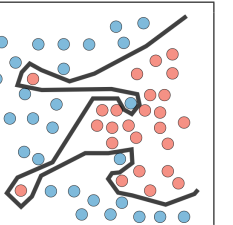
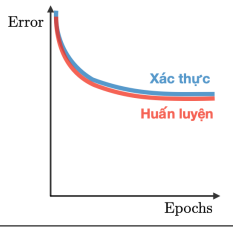
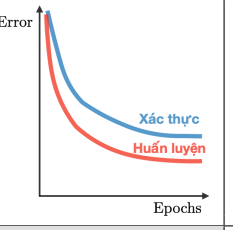
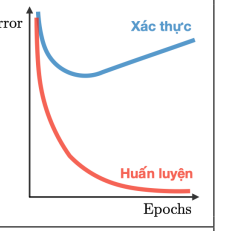
4.4 Dự đoán

□ **Bias** – Bias của mô hình là sai số giữa dự đoán mong đợi và dự đoán của mô hình trên các điểm dữ liệu cho trước.

□ **Phương sai** – Phương sai của một mô hình là sự thay đổi dự đoán của mô hình trên các điểm dữ liệu cho trước.

□ **Sự đánh đổi bias/phương sai** – Mô hình càng đơn giản bias càng lớn, mô hình càng phức tạp phương sai càng cao.

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"> Lỗi huấn luyện cao Lỗi huấn luyện tiến gần tới lỗi test Bias cao 	<ul style="list-style-type: none"> Lỗi huấn luyện thấp hơn một chút so với lỗi test 	<ul style="list-style-type: none"> Lỗi huấn luyện rất thấp Lỗi huấn luyện thấp hơn lỗi test rất nhiều Phương sai cao
Minh hoạ hồi quy			

Mình hoạ phân loại			
Mình hoạ deep learning (học sâu)			
Biện pháp khắc phục có thể dùng	- Mô hình phức tạp - Thêm nhiều đặc trưng - Huấn luyện lâu hơn		- Thực hiện chuẩn hóa - Lấy nhiều dữ liệu hơn

□ **Phân tích lỗi** – Phân tích lỗi là phân tích nguyên nhân của sự khác biệt trong hiệu năng giữa mô hình hiện tại và mô hình lý tưởng.

□ **Phân tích Ablative** – Phân tích Ablative là phân tích nguyên nhân của sự khác biệt giữa hiệu năng của mô hình hiện tại và mô hình cơ sở.

5 Refreshers

5.1 Xác suất và thống kê

Dịch bởi Hoàng Minh Tuấn và Hung Nguyễn

5.2 Giới thiệu về Xác suất và Tổ hợp

□ **Không gian mẫu** – Một tập hợp các kết cục có thể xảy ra của một phép thử được gọi là không gian mẫu của phép thử và được kí hiệu là S .

□ **Sự kiện (hay còn gọi là biến cố)** – Bất kỳ một tập hợp con E nào của không gian mẫu đều được gọi là một sự kiện. Một sự kiện là một tập các kết cục có thể xảy ra của phép thử. Nếu kết quả của phép thử chứa trong E , chúng ta nói sự kiện E đã xảy ra.

□ **Tiên đề của xác suất** – Với mỗi sự kiện E , chúng ta kí hiệu $P(E)$ là xác suất sự kiện E xảy ra.

$$(1) \quad 0 \leq P(E) \leq 1 \quad (2) \quad P(S) = 1 \quad (3) \quad P\left(\bigcup_{i=1}^n E_i\right) = \sum_{i=1}^n P(E_i)$$

□ **Hoán vị** – Hoán vị là một cách sắp xếp r phần tử từ một nhóm n phần tử, theo một thứ tự nhất định. Số lượng cách sắp xếp như vậy là $P(n, r)$, được định nghĩa như sau:

$$P(n, r) = \frac{n!}{(n-r)!}$$

□ **Tổ hợp** – Một tổ hợp là một cách sắp xếp r phần tử từ n phần tử, không quan trọng thứ tự. Số lượng cách sắp xếp như vậy là $C(n, r)$, được định nghĩa như sau:

$$C(n, r) = \frac{P(n, r)}{r!} = \frac{n!}{r!(n-r)!}$$

Ghi chú: Chúng ta lưu ý rằng với $0 \leq r \leq n$, ta có $P(n, r) \geq C(n, r)$

5.3 Xác suất có điều kiện

□ **Định lý Bayes** – Với các sự kiện A và B sao cho $P(B) > 0$, ta có:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Ghi chú: ta có $P(A \cap B) = P(A)P(B|A) = P(A|B)P(B)$

□ **Phân vùng** – Cho $\{A_i, i \in [1, n]\}$ sao cho với mỗi i , $A_i \neq \emptyset$. Chúng ta nói rằng $\{A_i\}$ là một phân vùng nếu có:

$$\forall i \neq j, A_i \cap A_j = \emptyset \quad \text{và} \quad \bigcup_{i=1}^n A_i = S$$

Ghi chú: với bất cứ sự kiện B nào trong không gian mẫu, ta có $P(B) = \sum_{i=1}^n P(B|A_i)P(A_i)$.

□ **Định lý Bayes mở rộng** – Cho $\{A_i, i \in [1, n]\}$ là một phân vùng của không gian mẫu. Ta có:

$$P(A_k|B) = \frac{P(B|A_k)P(A_k)}{\sum_{i=1}^n P(B|A_i)P(A_i)}$$

□ **Sự kiện độc lập** – Hai sự kiện A và B được coi là độc lập khi và chỉ khi ta có:

$$P(A \cap B) = P(A)P(B)$$

5.4 Biến ngẫu nhiên

□ **Biến ngẫu nhiên** – Một biến ngẫu nhiên, thường được kí hiệu là X , là một hàm nối mỗi phần tử trong một không gian mẫu thành một số thực.

□ **Hàm phân phối tích lũy (CDF)** – Hàm phân phối tích lũy F , là một hàm đơn điệu không giảm, sao cho $\lim_{x \rightarrow -\infty} F(x) = 0$ và $\lim_{x \rightarrow +\infty} F(x) = 1$, được định nghĩa là:

$$F(x) = P(X \leq x)$$

Ghi chú: chúng ta có $P(a < X \leq b) = F(b) - F(a)$.

□ **Hàm mật độ xác suất (PDF)** – Hàm mật độ xác suất f là xác suất mà X nhận các giá trị giữa hai giá trị thực liên tiếp của biến ngẫu nhiên.

□ **Mối quan hệ liên quan giữa PDF và CDF** – Dưới đây là các thuộc tính quan trọng cần biết trong trường hợp rời rạc (D) và liên tục (C).

Trường hợp	CDF F	PDF f	Thuộc tính của PDF
(D)	$F(x) = \sum_{x_i \leq x} P(X = x_i)$	$f(x_j) = P(X = x_j)$	$0 \leq f(x_j) \leq 1$ và $\sum_j f(x_j) = 1$
(C)	$F(x) = \int_{-\infty}^x f(y)dy$	$f(x) = \frac{dF}{dx}$	$f(x) \geq 0$ và $\int_{-\infty}^{+\infty} f(x)dx = 1$

□ **Phương sai** – Phương sai của một biến ngẫu nhiên, thường được kí hiệu là $\text{Var}(X)$ hoặc σ^2 , là một độ đo mức độ phân tán của hàm phân phối. Nó được xác định như sau:

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

□ **Độ lệch chuẩn** – Độ lệch chuẩn của một biến ngẫu nhiên, thường được kí hiệu σ , là thước đo mức độ phân tán của hàm phân phối của nó so với các đơn vị của biến ngẫu nhiên thực tế. Nó được xác định như sau:

$$\sigma = \sqrt{\text{Var}(X)}$$

□ **Kỳ vọng và moment của phân phối** – Dưới đây là các biểu thức của giá trị kỳ vọng $E[X]$, giá trị kỳ vọng tổng quát $E[g(X)]$, moment bậc k $E[X^k]$ và hàm đặc trưng $\psi(\omega)$ cho các trường hợp rời rạc và liên tục:

Trường hợp	$E[X]$	$E[g(X)]$	$E[X^k]$	$\psi(\omega)$
(D)	$\sum_{i=1}^n x_i f(x_i)$	$\sum_{i=1}^n g(x_i) f(x_i)$	$\sum_{i=1}^n x_i^k f(x_i)$	$\sum_{i=1}^n f(x_i) e^{i\omega x_i}$
(C)	$\int_{-\infty}^{+\infty} x f(x) dx$	$\int_{-\infty}^{+\infty} g(x) f(x) dx$	$\int_{-\infty}^{+\infty} x^k f(x) dx$	$\int_{-\infty}^{+\infty} f(x) e^{i\omega x} dx$

□ **Biến đổi các biến ngẫu nhiên** – Đặt các biến X và Y được liên kết với nhau bởi một hàm. Kí hiệu f_X và f_Y lần lượt là các phân phối của X và Y , ta có:

$$f_Y(y) = f_X(x) \left| \frac{dx}{dy} \right|$$

□ **Quy tắc tích phân Leibniz** – Gọi g là hàm của x và có khả năng c , và a, b là các ranh giới có thể phụ thuộc vào c . Chúng ta có:

$$\frac{\partial}{\partial c} \left(\int_a^b g(x) dx \right) = \frac{\partial b}{\partial c} \cdot g(b) - \frac{\partial a}{\partial c} \cdot g(a) + \int_a^b \frac{\partial g}{\partial c}(x) dx$$

□ **Bất đẳng thức Chebyshev** – Gọi X là biến ngẫu nhiên có giá trị kỳ vọng μ . Với $k, \sigma > 0$, chúng ta có bất đẳng thức sau:

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

5.5 Phân phối đồng thời biến ngẫu nhiên

□ **Mật độ có điều kiện** – Mật độ có điều kiện của X với Y , thường được kí hiệu là $f_{X|Y}$, được định nghĩa như sau:

$$f_{X|Y}(x) = \frac{f_{XY}(x, y)}{f_Y(y)}$$

□ **Tính chất độc lập** – Hai biến ngẫu nhiên X và Y độc lập nếu ta có:

$$f_{XY}(x, y) = f_X(x) f_Y(y)$$

□ **Mật độ biên và phân phối tích lũy** – Từ hàm phân phối mật độ đồng thời f_{XY} , ta có

Trường hợp	Mật độ biên	Hàm tích lũy
(D)	$f_X(x_i) = \sum_j f_{XY}(x_i, y_j)$	$F_{XY}(x, y) = \sum_{x_i \leq x} \sum_{y_j \leq y} f_{XY}(x_i, y_j)$
(C)	$f_X(x) = \int_{-\infty}^{+\infty} f_{XY}(x, y) dy$	$F_{XY}(x, y) = \int_{-\infty}^x \int_{-\infty}^y f_{XY}(x', y') dx' dy'$

□ **Tính chất độc lập** – Hai biến ngẫu nhiên X và Y độc lập nếu ta có:

$$\psi_{X+Y}(\omega) = \psi_X(\omega) \times \psi_Y(\omega)$$

□ **Hiệp phương sai** – Chúng ta xác định hiệp phương sai của hai biến ngẫu nhiên X và Y , thường được kí hiệu σ_{XY}^2 hay $\text{Cov}(X, Y)$, như sau:

$$\text{Cov}(X, Y) \triangleq \sigma_{XY}^2 = E[(X - \mu_X)(Y - \mu_Y)] = E[XY] - \mu_X \mu_Y$$

□ **Hệ số tương quan** – Kí hiệu σ_X, σ_Y là độ lệch chuẩn của X và Y , chúng ta xác định hệ số tương quan giữa X và Y , kí hiệu ρ_{XY} , như sau:

$$\rho_{XY} = \frac{\sigma_{XY}^2}{\sigma_X \sigma_Y}$$

Ghi chú 1: chúng ta lưu ý rằng với bất cứ biến ngẫu nhiên X, Y nào, ta luôn có $\rho_{XY} \in [-1, 1]$.

Ghi chú 2: Nếu X và Y độc lập với nhau thì $\rho_{XY} = 0$.

□ **Các phân phối chính** – Dưới là các phân phối chính cần ghi nhớ:

Loại	Phân phối	PDF	$\psi(\omega)$	$E[X]$	$\text{Var}(X)$
(D)	$X \sim \mathcal{B}(n, p)$ Binomial	$P(X = x) = \binom{n}{x} p^x q^{n-x}$ $x \in \llbracket 0, n \rrbracket$	$(pe^{i\omega} + q)^n$	np	npq
	$X \sim \text{Po}(\mu)$ Poisson	$P(X = x) = \frac{\mu^x}{x!} e^{-\mu}$ $x \in \mathbb{N}$	$e^{\mu(e^{i\omega} - 1)}$	μ	μ
(C)	$X \sim \mathcal{U}(a, b)$ Uniform	$f(x) = \frac{1}{b-a}$ $x \in [a, b]$	$\frac{e^{i\omega b} - e^{i\omega a}}{(b-a)i\omega}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
	$X \sim \mathcal{N}(\mu, \sigma)$ Gaussian	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ $x \in \mathbb{R}$	$e^{i\omega\mu - \frac{1}{2}\omega^2\sigma^2}$	μ	σ^2
	$X \sim \text{Exp}(\lambda)$ Exponential	$f(x) = \lambda e^{-\lambda x}$ $x \in \mathbb{R}_+$	$\frac{1}{1 - \frac{i\omega}{\lambda}}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$

5.6 Ước lượng tham số

□ **Mẫu ngẫu nhiên** – Mẫu ngẫu nhiên là tập hợp của n biến ngẫu nhiên X_1, \dots, X_n độc lập và được phân phối giống hệt với X .

□ **Công cụ ước tính** – Công cụ ước tính (estimator) là một hàm của dữ liệu được sử dụng để suy ra giá trị của một tham số chưa biết trong mô hình thống kê.

□ **Thiên vị** – Thiên vị (bias) của Estimator $\hat{\theta}$ được định nghĩa là chênh lệch giữa giá trị kì vọng của phân phối $\hat{\theta}$ và giá trị thực, tức là

$$\text{Bias}(\hat{\theta}) = E[\hat{\theta}] - \theta$$

Ghi chú: một công cụ ước tính được cho là không thiên vị (unbiased) khi chúng ta có $E[\hat{\theta}] = \theta$.

□ **Giá trị trung bình mẫu** – Giá trị trung bình mẫu của mẫu ngẫu nhiên được sử dụng để ước tính giá trị trung bình thực μ của phân phối, thường được kí hiệu \bar{X} và được định nghĩa như sau:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

Ghi chú: trung bình mẫu là không thiên vị (unbiased), nghĩa là $E[\bar{X}] = \mu$.

□ **Phương sai mẫu** – Phương sai mẫu của mẫu ngẫu nhiên được sử dụng để ước lượng phương sai thực sự σ^2 của phân phối, thường được kí hiệu là s^2 hoặc $\hat{\sigma}^2$ và được định nghĩa như sau:

$$s^2 = \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

Ghi chú: phương sai mẫu không thiên vị (unbiased), nghĩa là $E[s^2] = \sigma^2$.

□ **Định lý giới hạn trung tâm** – Giả sử chúng ta có một mẫu ngẫu nhiên X_1, \dots, X_n theo một phân phối nhất định với trung bình μ và phương sai σ^2 , sau đó chúng ta có:

$$\bar{X} \underset{n \rightarrow +\infty}{\sim} \mathcal{N}\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

5.7 Đại số và vi tích phân

Dịch bởi Hoàng Minh Tuấn và Phạm Hồng Vinh

5.8 Kí hiệu chung

□ **Vectơ** – Chúng ta kí hiệu $x \in \mathbb{R}^n$ là một vectơ với n phần tử, với $x_i \in \mathbb{R}$ là phần tử thứ i :

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

□ **Ma trận** – Kí hiệu $A \in \mathbb{R}^{m \times n}$ là một ma trận với m hàng và n cột, $A_{i,j} \in \mathbb{R}$ là phần tử nằm ở hàng thứ i , cột j :

$$A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

Ghi chú: vectơ x được xác định ở trên có thể coi như một ma trận $n \times 1$ và được gọi là vectơ cột.

□ **Ma trận đơn vị** – Ma trận đơn vị $I \in \mathbb{R}^{n \times n}$ là một ma trận vuông với các phần tử trên đường chéo chính bằng 1 và các phần tử còn lại bằng 0:

$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Ghi chú: với mọi ma trận vuông $A \in \mathbb{R}^{n \times n}$, ta có $A \times I = I \times A = A$.

□ **Ma trận đường chéo** – Ma trận đường chéo $D \in \mathbb{R}^{n \times n}$ là một ma trận vuông với các phần tử trên đường chéo chính khác 0 và các phần tử còn lại bằng 0:

$$D = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & d_n \end{pmatrix}$$

Ghi chú: chúng ta kí hiệu D là $\text{diag}(d_1, \dots, d_n)$.

5.9 Các phép toán ma trận

5.9.1 Phép nhân

□ **Vectơ/vectơ** – Có hai loại phép nhân vectơ/vectơ:

- phép nhân inner: với $x, y \in \mathbb{R}^n$, ta có:

$$x^T y = \sum_{i=1}^n x_i y_i \in \mathbb{R}$$

- phép nhân outer: với $x \in \mathbb{R}^m, y \in \mathbb{R}^n$, ta có:

$$xy^T = \begin{pmatrix} x_1 y_1 & \cdots & x_1 y_n \\ \vdots & & \vdots \\ x_m y_1 & \cdots & x_m y_n \end{pmatrix} \in \mathbb{R}^{m \times n}$$

□ **Ma trận/vectơ** – Phép nhân giữa ma trận $A \in \mathbb{R}^{m \times n}$ và vectơ $x \in \mathbb{R}^n$ là một vectơ có kích thước \mathbb{R}^m :

$$Ax = \begin{pmatrix} a_{r,1}^T x \\ \vdots \\ a_{r,m}^T x \end{pmatrix} = \sum_{i=1}^n a_{c,i} x_i \in \mathbb{R}^m$$

với $a_{r,i}^T$ là các vectơ hàng và $a_{c,j}$ là các vectơ cột của A , và x_i là các phần tử của x .

□ **Ma trận/ma trận** – Phép nhân giữa ma trận $A \in \mathbb{R}^{m \times n}$ và $B \in \mathbb{R}^{n \times p}$ là một ma trận kích thước $\mathbb{R}^{m \times p}$:

$$AB = \begin{pmatrix} a_{r,1}^T b_{c,1} & \cdots & a_{r,1}^T b_{c,p} \\ \vdots & & \vdots \\ a_{r,m}^T b_{c,1} & \cdots & a_{r,m}^T b_{c,p} \end{pmatrix} = \sum_{i=1}^n a_{c,i} b_{r,i}^T \in \mathbb{R}^{n \times p}$$

với $a_{r,i}^T, b_{r,i}^T$ là các vectơ hàng và $a_{c,j}, b_{c,j}$ lần lượt là các vectơ cột của A và B .

5.9.2 Một số phép toán khác

□ **Chuyển vị** – Chuyển vị của một ma trận $A \in \mathbb{R}^{m \times n}$, kí hiệu A^T , khi các phần tử hàng cột hoán đổi vị trí cho nhau:

$$\forall i, j, \quad A_{i,j}^T = A_{j,i}$$

Ghi chú: với ma trận A, B , ta có $(AB)^T = B^T A^T$

□ **Nghịch đảo** – Nghịch đảo của ma trận vuông khả đảo A được kí hiệu là A^{-1} và chỉ tồn tại duy nhất:

$$AA^{-1} = A^{-1}A = I$$

Ghi chú: không phải tất cả các ma trận vuông đều khả đảo. Ngoài ra, với ma trận A, B , ta có $(AB)^{-1} = B^{-1}A^{-1}$

□ **Truy vết** – Truy vết của ma trận vuông A , kí hiệu $\text{tr}(A)$, là tổng của các phần tử trên đường chéo chính của nó:

$$\text{tr}(A) = \sum_{i=1}^n A_{i,i}$$

Ghi chú: với ma trận A, B , chúng ta có $\text{tr}(A^T) = \text{tr}(A)$ và $\text{tr}(AB) = \text{tr}(BA)$

□ **Định thức** – Định thức của một ma trận vuông $A \in \mathbb{R}^{n \times n}$, kí hiệu $|A|$ hay $\det(A)$ được tính đệ quy với $A_{\setminus i, \setminus j}$, ma trận A xóa đi hàng thứ i và cột thứ j :

$$\det(A) = |A| = \sum_{j=1}^n (-1)^{i+j} A_{i,j} |A_{\setminus i, \setminus j}|$$

Ghi chú: A khả đảo nếu và chỉ nếu $|A| \neq 0$. Ngoài ra, $|AB| = |A||B|$ và $|A^T| = |A|$.

5.10 Những tính chất của ma trận

5.10.1 Định nghĩa

□ **Phân rã đối xứng** – Một ma trận A đã cho có thể được biểu diễn dưới dạng các phần đối xứng và phản đối xứng của nó như sau:

$$A = \underbrace{\frac{A + A^T}{2}}_{\text{Đối xứng}} + \underbrace{\frac{A - A^T}{2}}_{\text{Phản đối xứng}}$$

□ **Chuẩn** – Một chuẩn (norm) là một hàm $N : V \rightarrow [0, +\infty[$ mà V là một không gian vectơ, và với mọi $x, y \in V$, ta có:

- $N(x + y) \leq N(x) + N(y)$
- $N(ax) = |a|N(x)$ với a là một số
- nếu $N(x) = 0$, thì $x = 0$

Với $x \in V$, các chuẩn thường dùng được tổng hợp ở bảng dưới đây:

Chuẩn	Kí hiệu	Định nghĩa	Trường hợp dùng
Manhattan, L^1	$\ x\ _1$	$\sum_{i=1}^n x_i $	LASSO chính quy hóa
Euclidean, L^2	$\ x\ _2$	$\sqrt{\sum_{i=1}^n x_i^2}$	Ridge chính quy hóa
p -norm, L^p	$\ x\ _p$	$\left(\sum_{i=1}^n x_i^p\right)^{\frac{1}{p}}$	Hölder bất đẳng thức
Infinity, L^∞	$\ x\ _\infty$	$\max_i x_i $	Uniform convergence

□ **Sự phụ thuộc tuyến tính** – Một tập hợp các vectơ được cho là phụ thuộc tuyến tính nếu một trong các vectơ trong tập hợp có thể được biểu diễn bởi một tổ hợp tuyến tính của các vectơ khác.

Ghi chú: nếu không có vectơ nào có thể được viết theo cách này, thì các vectơ được cho là độc lập tuyến tính

□ **Hạng ma trận (rank)** – Hạng của một ma trận A kí hiệu $\text{rank}(A)$ và là số chiều của không gian vectơ được tạo bởi các cột của nó. Điều này tương đương với số cột độc lập tuyến tính tối đa của A .

□ **Ma trận bán xác định dương** – Ma trận $A \in \mathbb{R}^{n \times n}$ là bán xác định dương (PSD) kí hiệu $A \succeq 0$ nếu chúng ta có:

$$A = A^T \quad \text{và} \quad \forall x \in \mathbb{R}^n, \quad x^T A x \geq 0$$

Ghi chú: tương tự, một ma trận A được cho là xác định dương và được kí hiệu $A \succ 0$, nếu đó là ma trận PSD thỏa mãn cho tất cả các vectơ khác không x , $x^T A x > 0$.

□ **Giá trị riêng, vectơ riêng** – Cho ma trận $A \in \mathbb{R}^{n \times n}$, λ được gọi là giá trị riêng của A nếu tồn tại một vectơ $z \in \mathbb{R}^n \setminus \{0\}$, được gọi là vectơ riêng, sao cho:

$$Az = \lambda z$$

□ **Định lý phổ** – Cho $A \in \mathbb{R}^{n \times n}$. Nếu A đối xứng, thì A có thể chéo hóa bởi một ma trận trực giao thực $U \in \mathbb{R}^{n \times n}$. Bằng cách kí hiệu $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, chúng ta có:

$$\exists \Lambda \text{ đường chéo}, \quad A = U \Lambda U^T$$

□ **Phân tích giá trị suy biến** – Đối với một ma trận A có kích thước $m \times n$, Phân tích giá trị suy biến (SVD) là một kỹ thuật phân tích nhân tố nhằm đảm bảo sự tồn tại của đơn vị U $m \times m$, đường chéo Σ $m \times n$ và đơn vị V $n \times n$ ma trận, sao cho:

$$A = U \Sigma V^T$$

5.11 Giải tích ma trận

□ **Gradient** – Cho $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ là một hàm và $A \in \mathbb{R}^{m \times n}$ là một ma trận. Gradient của f đối với A là ma trận $m \times n$, được kí hiệu là $\nabla_A f(A)$, sao cho:

$$\left(\nabla_A f(A) \right)_{i,j} = \frac{\partial f(A)}{\partial A_{i,j}}$$

Ghi chú: gradient của f chỉ được xác định khi f là hàm trả về một số.

□ **Hessian** – Cho $f : \mathbb{R}^n \rightarrow \mathbb{R}$ là một hàm và $x \in \mathbb{R}^n$ là một vectơ. Hessian của f đối với x là một ma trận đối xứng $n \times n$, ghi chú $\nabla_x^2 f(x)$, sao cho:

$$\left(\nabla_x^2 f(x) \right)_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

Ghi chú: hessian của f chỉ được xác định khi f là hàm trả về một số.

□ **Các phép toán của gradient** – Đối với ma trận A, B, C , các thuộc tính gradient sau cần để lưu ý:

$$\nabla_A \text{tr}(AB) = B^T$$

$$\nabla_{A^T} f(A) = (\nabla_A f(A))^T$$

$$\nabla_A \text{tr}(ABA^T C) = CAB + C^T A B^T$$

$$\nabla_A |A| = |A| (A^{-1})^T$$