

# VIP Cheatsheet: Học có giám sát

Afshine AMIDI và Shervine AMIDI

Ngày 17 tháng 5 năm 2020

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

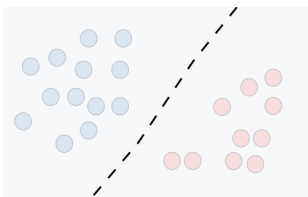
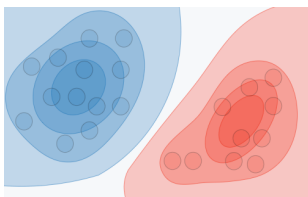
## Giới thiệu về học có giám sát

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

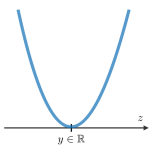
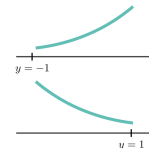
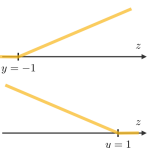
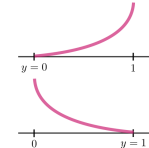
□ **Loại mô hình** – Các mô hình khác nhau được tổng kết trong bảng bên dưới:

	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

## Các kí hiệu và khái niệm tổng quát

□ **Hypothesis** – Hypothesis được kí hiệu là  $h_\theta$ , 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:

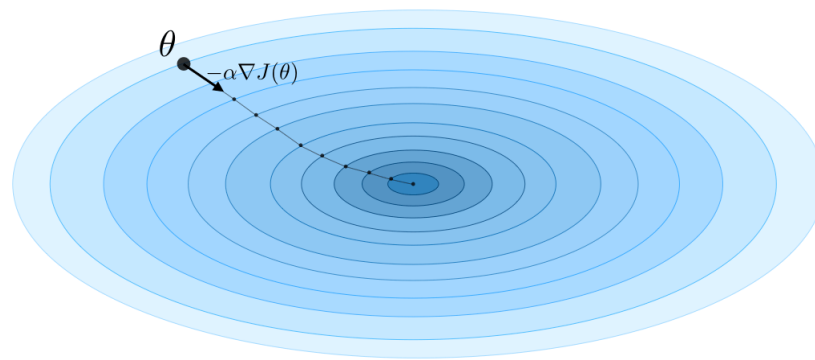
Least squared error	Mất mát Logistic	Mất mát Hinge	Cross-entropy
$\frac{1}{2}(y - z)^2$	$\log(1 + \exp(-yz))$	$\max(0, 1 - yz)$	$-\left[y \log(z) + (1 - y) \log(1 - z)\right]$
			
Hồi quy tuyến tính	Hồi quy Logistic	SVM	Mạng neural

□ **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  $\theta$  được sử dụng để tìm tham số tối ưu  $\theta$  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  $\theta$  thỏa 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:*

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

## 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  $\theta$  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  $\alpha$  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:

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

## 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  $\phi_i$  của mỗi lớp  $i$  bằng với:

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

## 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,  $\eta$ , một số liệu thống kê đầy đủ  $T(y)$  và hàm phân vùng log (log-partition function)  $a(\eta)$  sẽ có dạng như sau:

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

*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	$\eta$	$T(y)$	$a(\eta)$	$b(y)$
Bernoulli	$\log\left(\frac{\phi}{1-\phi}\right)$	$y$	$\log(1 + \exp(\eta))$	1
Gaussian	$\mu$	$y$	$\frac{\eta^2}{2}$	$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right)$
Poisson	$\log(\lambda)$	$y$	$e^{\eta}$	$\frac{1}{y!}$
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.*

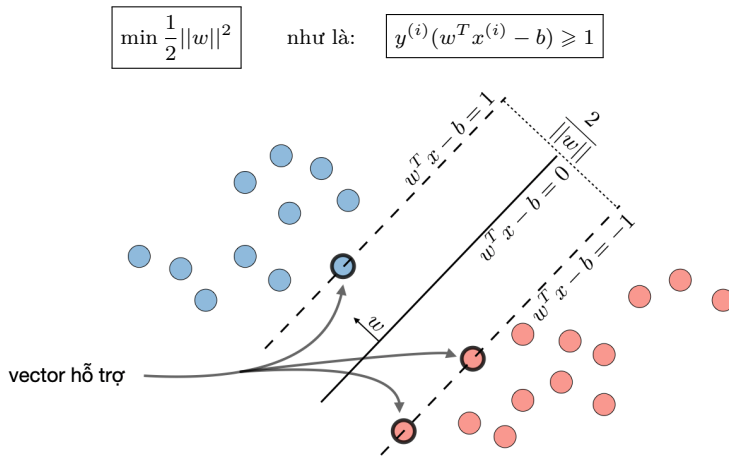
## Máy vector hỗ trợ

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:



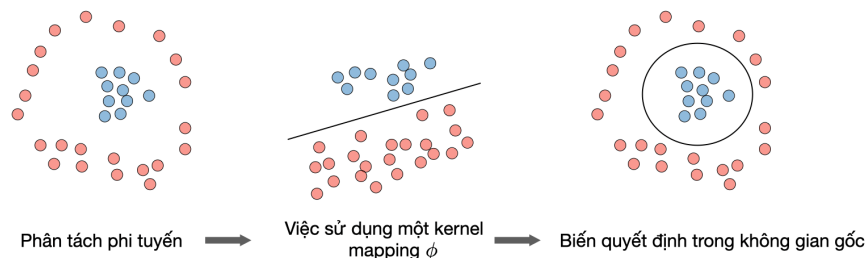
□ **Mất mát Hinge** – Mất mát Hinge được sử dụng trong thiết lập của SVMs và nó được định nghĩa như sau:

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

□ **Kernel (nhân)** – Cho trước feature mapping  $\phi$ , 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  $\phi$ , 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  $\beta_i$  được gọi là bội số Lagrange.

## 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.

## 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:

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

## Naive Bayes

□ **Giải 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  $k \in \{0, 1\}, l \in [1, L]$

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

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

## 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 kỹ 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, kỹ 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.*

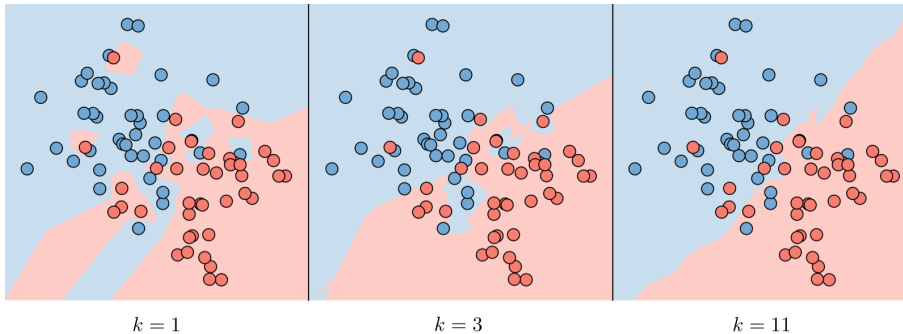
□ **Boosting** – Ý tưởng của các phương thức boosting là kết hợp các phương pháp học yếu hơn để tạo nên phương pháp học mạnh hơn. Những phương thức chính được tổng kết ở bảng dưới đây:

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

## 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.

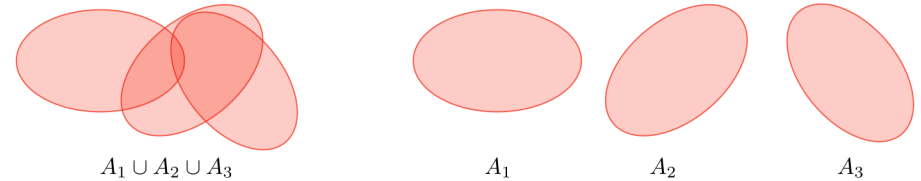
*Chú ý: Tham số  $k$  cao hơn, độ chệch (bias) cao hơn, tham số  $k$  thấp hơn, phương sai cao hơn.*



## 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  $\phi$ . 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 lý 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:

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

□ **Định lý giới hạn trên** – Cho  $\mathcal{H}$  là một finite hypothesis class mà  $|\mathcal{H}| = k$  với  $\delta$ , 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  $\text{VC}(\mathcal{H}) = d$  và  $m$  là số lượng các ví dụ huấn luyện. Với xác suất nhỏ nhất là  $1 - \delta$ , ta có:

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