

VIP Cheatsheet: Học không có giám sát

Afshine AMIDI và Shervine AMIDI

Ngày 17 tháng 5 năm 2020

Dịch bởi Trần Tuấn Anh và Đàm Minh Tiến

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:

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

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	$\mathcal{N}(0, I)$	$\mathcal{N}(\mu + \Lambda z, \psi)$	$\mu_j \in \mathbb{R}^n$

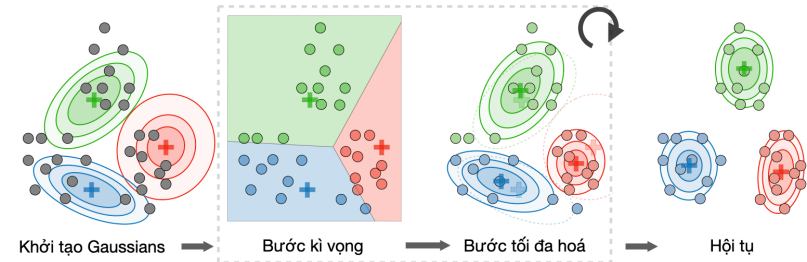
□ **Thuật toán** – Thuật toán tối đa hoá kì vọng (EM) mang lại một phương thức có hiệu quả trong việc ước lượng tham số θ thông qua tối đa hoá giá trị ước lượng likelihood bằng cách lặp lại việc tạo nên một cận dưới cho likelihood (E-step) và tối ưu hoá cận dưới (M-step) như sau:

- 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:

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



Phân cụm k -means

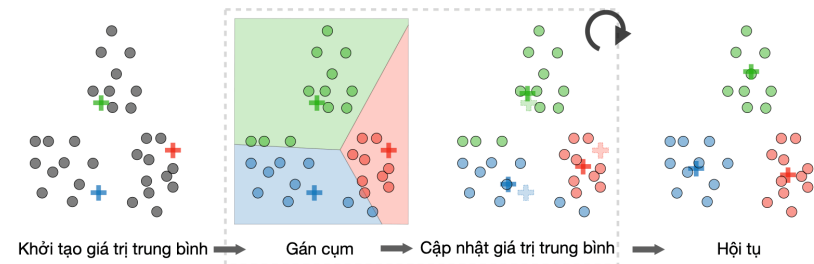
Chúng ta kí hiệu $c^{(i)}$ là cụm của điểm dữ liệu i và μ_j là điểm trung tâm của cụm j .

□ **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:

$$c^{(i)} = \operatorname{argmin}_j \|x^{(i)} - \mu_j\|^2$$

và

$$\mu_j = \frac{\sum_{i=1}^m 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{c^{(i)}=j\}}}$$



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

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

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

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

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

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:

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

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:

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

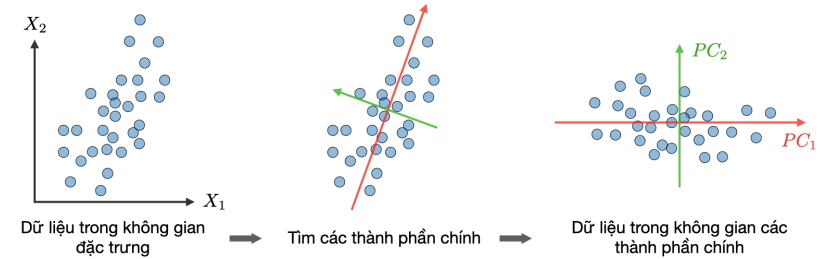
Chú thích: vector riêng tương ứng với giá trị riêng lớn nhất được gọi là vector riêng chính của ma trận A .

□ **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.

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

- 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 $\text{span}_{\mathbb{R}}(u_1, \dots, u_k)$.



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:

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

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:

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