MACHINE LEARNING

Cao Văn Chung
cvanchung@hus.edu.vn
Informatics Dept., MIM, HUS, VNU Hanoi

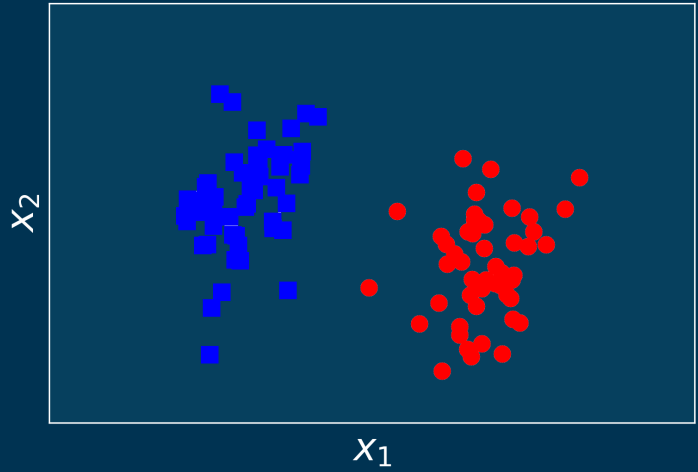
Latent variable - Mixture Models

Multi-Dimentional Gaussian distribution

- Ta xét lại ví dụ tạo dữ liệu trong đoạn lệnh dưới đây (bài SVM - Hard Margin)

```
means = [[1, 2], [5, 1]]
cov = [[.5, .2], [.2, .5]]
N = 50
# create class 1
X0 = np.random.multivariate_normal(means[0], cov, N)
# create class -1
X1 = np.random.multivariate_normal(means[1], cov, N)
# rearange all data
X = np.concatenate((X0.T, X1.T), axis = 1)
# labels
y = np.concatenate((np.ones((1, N)), -1*np.ones((1, N))), axis = 1)
```

Distribution Data from Code



Multi-Dimentional Gaussian distribution

Nhận xét

- Trong các ví dụ với dữ liệu nhân tạo ở phần trước, ta thường chọn tạo dữ liệu với phân phối Gauss (Gaussian distribution).
- Khi thực hiện cách tiếp cận ước lượng hợp lý cực đại để xác định tham số (MLE) ta cũng thường dựa vào giả thiết dữ liệu tuân theo phân bố chuẩn.
- Lý do?

MLE for Multi-Dimentional Gaussian distribution

- Xét bộ dữ liệu N quan sát độc lập $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. Trong đó $\mathbf{x}_n \in \mathbb{R}^d$, $n = 1, 2, \ldots, N$.
- Giả sử các $\mathbf{x}_n, \quad n = 1, 2, \ldots, N$ được lấy mẫu từ phân phối Gaussian đa chiều (d chiều).
- Ta cần ước lượng các tham số của các phân phối tác động lên $\mathbf{X}$ thông qua ước lượng hợp lý cực đại MLE.

MLE for Multi-Dimentional Gaussian distribution

- Giả thiết các điểm quan sát $\mathbf{x}_n$ là được lấy từ một mẫu ngẫu nhiên phân bố chuẩn $\mathbf{X} \sim \mathcal{N}(\mu, \sigma^2)$.
- Lúc đó hàm mật độ của điểm dữ liệu $\mathbf{x}_n$, cũng là ước lượng hợp lý của nó, sẽ là

$$f_{\mathbf{x}_n}(\mathbf{x}; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\mathbf{x}-\mu)^2}{2\sigma^2}\right).$$

- Từ giả thiết N mẫu quan sát độc lập, ta có hàm ước lượng hợp lý cho toàn tập dữ liệu là

$$L(\mu, \Sigma|\mathbf{X}) = \prod_{n=1}^{N} f_{\mathbf{x}_n}(\mathbf{x}_n|\mu, \Sigma).$$

- Với ma trận vuông A, ký hiệu $|A| = \det(A)$ - định thức của A, ta sử dụng hàm log hợp lý cực đại:

MLE for Multi-Dimentional Gaussian distribution

$$
\begin{aligned}
l(\mu, \Sigma|\mathbf{X}) &= \log\left[\prod_{n=1}^{N} f_{\mathbf{x}_n}(\mathbf{x}_n|\mu, \Sigma)\right] \\
&= \log\left[\prod_{n=1}^{N} \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_n - \mu)^{\intercal}\Sigma^{-1}(\mathbf{x}_n - \mu)\right)\right] \\
&= \sum_{n=1}^{N}\left(-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(\mathbf{x}_n - \mu)^{\intercal}\Sigma^{-1}(\mathbf{x}_n - \mu)\right) \qquad (1) \\
&= -\frac{N}{2}\log|\Sigma| - \sum_{n=1}^{N}\frac{1}{2}(\mathbf{x}_n - \mu)^{\intercal}\Sigma^{-1}(\mathbf{x}_n - \mu) - \underbrace{\frac{Nd}{2}\log(2\pi)}_{C} \\
&= -\frac{N}{2}\log|\Sigma| - \sum_{n=1}^{N}\frac{1}{2}(\mathbf{x}_n - \mu)^{\intercal}\Sigma^{-1}(\mathbf{x}_n - \mu) + C
\end{aligned}
$$

MLE for Multi-Dimentional Gaussian distribution

- Lấy đạo hàm $l(\mu, \Sigma|\mathbf{X})$ theo μ ta dùng công thức: $\frac{\partial \mathbf{w}^T\mathbf{A}\mathbf{w}}{\partial \mathbf{w}} = 2\mathbf{A}\mathbf{w}$. Lúc đó đặt $\Sigma^{-1} = \mathbf{A}$ và $\mathbf{x}_i - \mu = \mathbf{w}$ ta có

$$\frac{\partial l(\mu, \Sigma|\mathbf{X})}{\partial \mu} = -\sum_{i=1}^{N} \Sigma^{-1}(\mathbf{x}_i - \mu) = \Sigma^{-1}(N\mu - \sum_{i=1}^{N} \mathbf{x}_i) = 0$$

- Do Σ tồn tại nên từ đẳng thức cuối ta suy ra tham số tối ưu thỏa mãn

$$N\hat{\mu} - \sum_{i=1}^{N} \mathbf{x}_i = 0 \Leftrightarrow \hat{\mu} = \frac{\sum_{i=1}^{N} \mathbf{x}_i}{N} \quad (2)$$

MLE for Multi-Dimentional Gaussian distribution

Lấy đạo hàm $l(\mu, \Sigma|\mathcal{N})$ theo Σ ta cần dùng các công thức sau

- Nếu các phép nhân ma trận thực hiện được thì

$$\text{trace}(\mathbf{ABC}) = \text{trace}(\mathbf{CAB}) = \text{trace}(\mathbf{BCA})$$

- Nếu $\mathbf{x}^\intercal\mathbf{Ax} \in \mathbb{R}$ - đại lượng vô hướng thì:

$$\mathbf{x}^\intercal\mathbf{Ax} = \text{trace}(\mathbf{x}^\intercal\mathbf{Ax}) = \text{trace}(\mathbf{x}^\intercal\mathbf{xA})$$

- Các đạo hàm: $\dfrac{\partial\ \text{trace}(\mathbf{AB})}{\partial\mathbf{A}} = \dfrac{\partial\ \text{trace}(\mathbf{BA})}{\partial\mathbf{A}} = \mathbf{B}^\intercal$ và $\dfrac{\partial \log(\mathbf{A})}{\partial\mathbf{A}} = \mathbf{A}^{-\intercal}$.
- Ký hiệu $|A| = \det(A)-$ định thức ma trận vuông A, thì $|\mathbf{A}| = \dfrac{1}{|\mathbf{A}^{-1}|}$.

MLE for Multi-Dimentional Gaussian distribution

- Áp dụng để lấy đạo hàm $l(\mu, \Sigma|\mathcal{N})$ theo Σ ta có

$$
\begin{aligned}
l(\mu, \Sigma|\mathcal{D}) &= C - \frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \mu)^\intercal \Sigma^{-1}(\mathbf{x}_i - \mu) \\
&= C + \frac{N}{2}\log|\Sigma^{-1}| - \frac{1}{2}\sum_{i=1}^{N}\text{trace}\left[(\mathbf{x}_i - \mu)^\intercal(\mathbf{x}_i - \mu)\Sigma^{-1}\right]
\end{aligned}
$$

Và do đó

$$
\begin{aligned}
\frac{\partial l(\mu, \Sigma|\mathcal{D})}{\partial \Sigma^{-1}} &= \frac{N}{2}\Sigma^\intercal - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^\intercal \\
&= \frac{N}{2}\Sigma - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^\intercal
\end{aligned}
$$

MLE for Multi-Dimentional Gaussian distribution

- Đẳng thức cuối suy ra từ tính đối xứng của Σ.
- Cho đạo hàm bằng 0, ta có tham số tối ưu

$$\frac{N}{2}\hat{\Sigma} - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^\intercal = 0 \Leftrightarrow \hat{\Sigma} = \frac{\sum_{i=1}^{N}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^\intercal}{N} \quad (3)$$

- Từ (2) và (3) ta thấy ước lượng tối ưu cho tham số: kỳ vọng chính là trung bình cộng mẫu và Σ chính là ma trận hiệp phương sai mẫu.

$$\begin{cases} \hat{\mu} &= \frac{\sum_{i=1}^{N}\mathbf{x}_i}{N} = \mathbb{E}(\mathbf{X}) \\ \hat{\Sigma} &= \frac{\sum_{i=1}^{N}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^\intercal}{N} = \mathbb{C}\mathbf{ov}(\mathbf{X}) \end{cases} \quad (4)$$

Mixture models

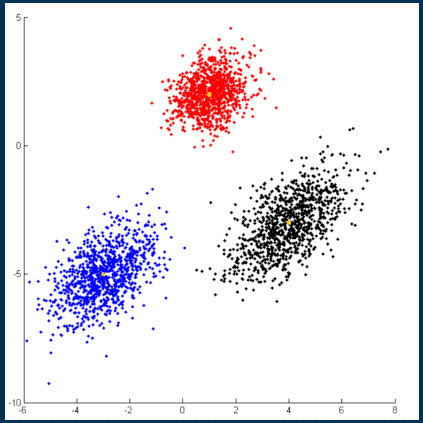
- Trong phần tính toán trên, ta đã ước lượng tham số tối ưu cho trường hợp toàn bộ tập dữ liệu tuân theo chỉ 01 phân bố Gaussian duy nhất.
- Thực tế, một dữ liệu tổng thể có thể tổng hợp từ nhiều quần thể dữ liệu nhỏ, trong đó mỗi quần thể dữ liệu chịu ảnh hưởng bởi một/một số quy luật thống kê riêng.
- Mô hình xác suất cho phép biểu thị sự hiện diện của các quần thể con trong một tổng thể, nhưng không bắt buộc phải xác định rõ mỗi dữ liệu quan sát thuộc về quần thể con nào gọi là **Mô hình trộn - Mô hình hỗn hợp - Mixture models**.
- Trong mô hình này, mỗi dữ liệu quan sát chịu tác động của một số phân phối xác suất.
- Phần tiếp theo ta sẽ áp dụng các dạng mô hình hỗn hợp để phân cụm dữ liệu - một phương pháp học máy không giám sát.

Latent variable models

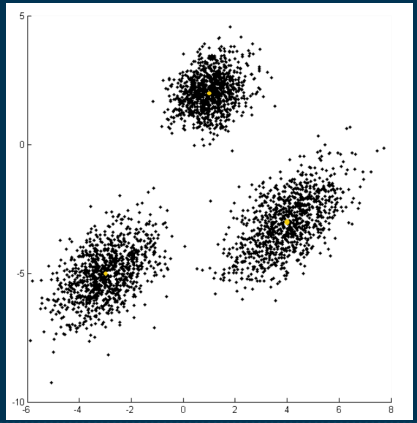
Quan sát các tập dữ liệu dưới đây:

Hình (a) là tập dữ liệu phân bố thành 03 cụm - được gán nhãn (theo màu sắc đỏ - xanh - đen). Hình (b) vẫn là dữ liệu đó nhưng không có nhãn (tất cả cùng màu).

(a)



(b)



Latent variable models

- **Nhắc lại:**
  - Trong các mô hình học máy có huấn luyện (supervised), training dataset với biến quan sát đầu vào $\mathbf{x}_n$ và biến đầu ra (nhãn hoặc giá trị) y_n, giống như hình (a) ở trên, mỗi nhóm dữ liệu được gán nhãn (ví dụ màu sắc) khác nhau.
  - Trong lúc đó, với học không giám sát, ta chỉ có dữ liệu quan sát đầu vào $\mathbf{x}_n$ mà không có nhãn, như là tập dữ liệu testing dataset, giống như hình (b).
- Tuy nhiên, dữ liệu trong cả hai trường hợp thực tế đều phân bố vào các cụm.
- **Câu hỏi:** Nếu chỉ có tập dữ liệu thô (chưa gán nhãn) thì có cách nào xác định được dữ liệu thực sự phân bố thành các cụm như thế nào hay không? Giả thiết các cụm đó đều có phân bố chuẩn (Gaussian).

$\Longrightarrow$ **Phương pháp phân cụm dữ liệu - Clustering.**

Latent variable models

- Trong các mô hình học có huấn luyện (supervised): Ngoài (x_n, y_n) là các biến quan sát và nhãn, mô hình sẽ tính toán & sử dụng các biến trung gian, ví dụ để xác định phân lớp c phù hợp
  - Mô hình Hồi quy Softmax sử dụng $g(z_c) = \frac{\exp(w_c^T x)}{\sum_{k=1}^{C} \exp(w_k^T x)}$;
  - Mô hình Naive Bayes sử dụng $f_c(x) = P(c)P(x|c)$;
  - Mô hình Multi-Class SVM sử dụng $z_c(x) = w_c^T x$...
- Các biến trung gian này được tính từ dữ liệu dựa trên một công thức xác suất - một giả thiết về dữ liệu.

Mô hình với cách tiếp cận: Thay vì biết nhãn y_n, ta biết thông tin phân bố của x_n, dựa vào đó xây dựng mô hình thống kê để tính toán và sử dụng các biến trung gian cho phân lớp dữ liệu $\Longrightarrow$

Gọi là **Mô hình biến ẩn - Latent variable models**.

Gaussian Mixture Model

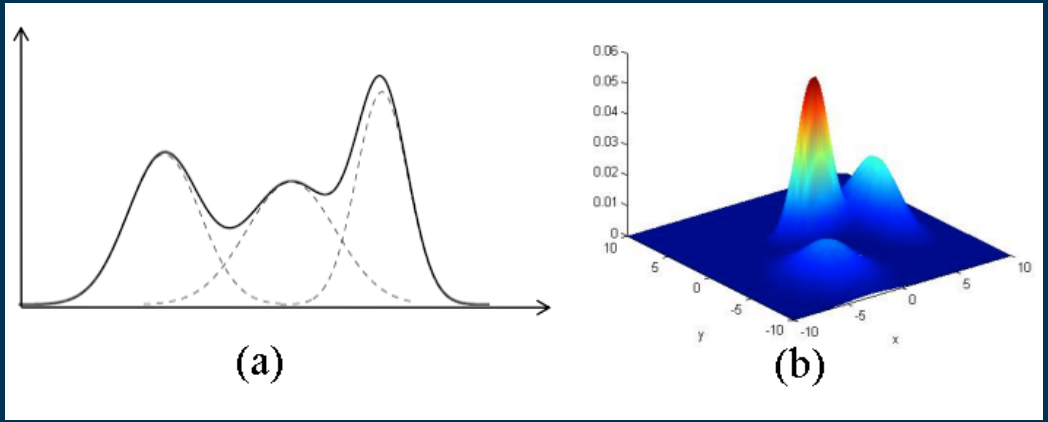
- Giả sử dữ liệu phân bố thành C cụm, dữ liệu trong mỗi một cụm lại có dạng phân phối Gaussian nhiều chiều.
- Chú ý: Xác suất của mỗi điểm dữ liệu không chỉ phụ thuộc vào một phân phối Gaussian duy nhất mà chịu tác động kết hợp từ nhiều phân phối Gaussian khác nhau từ mỗi cụm.

$\Rightarrow$ Gọi là **Mô hình trộn Gaussian - Gaussian Mixture Model - GMM**.

- Mục tiêu của GMM là ước lượng tham số phù hợp nhất cho C cụm thông qua phương pháp ước lượng hợp lý cực đại MLE.
- Giả thiết của mô hình:
  - Dữ liệu mỗi cụm tuân theo phân phối Gaussian d chiều với tập tham số $\{(\mu_c, \Sigma_c)\}_{c=1}^{C}$
  - Tồn tại ước lượng ngẫu nhiên $z_c = z_c(\mathbf{x})$: $z_c = 1$ nếu dữ liệu $\mathbf{x}$ thuộc cụm thứ c; ngược lại $z_c(\mathbf{x}) = 0$.

Gaussian Mixture Model

Hình (a) Phân bố hỗn hợp từ 03 phân phối Gaussian 01 chiều. Hình (b) Phân bố hỗn hợp từ 03 phân phối Gaussian 02 chiều.



Gaussian Mixture Model

- z_c được coi như là một biến ẩn (**latent variable**) - Ta không quan sát được giá trị của nó. Vậy GMM là mô hình biến ẩn.
- Xác suất $P(z_c = 1|\mathbf{x})$ giúp ta xác định tham số phân phối của mô hình Gaussian Mixture.
- Tập các z_c của mỗi cụm c tạo thành phân phối xác suất $(\pi_1, \pi_2, \ldots, \pi_C)$, với $\pi_k = p(z_k = 1|\mathbf{x})$.
- Một xác suất hỗn hợp tác động lên một điểm dữ liệu $\mathbf{x}$ được tính theo công thức Bayes như sau:

$$p(\mathbf{x}) = \sum_{c=1}^{C} p(z_c)p(\mathbf{x}|z_c) = \sum_{c=1}^{C} p(z_c = 1)p(\mathbf{x}|\mu_c, \Sigma_c)$$
$$= \sum_{c=1}^{C} \pi_c p(\mathbf{x}|\mu_c, \Sigma_c) = \sum_{c=1}^{C} \pi_c N(\mathbf{x}|\mu_c, \Sigma_c)$$

MLE for Gaussian Mixture Model

- Xác suất $p(\mathbf{x}|\mu_i, \Sigma_i)$ được tính từ phân phối Gaussian d chiều và cũng là đại lượng mục tiêu cần tham số hóa.
- Giả sử chúng có một tập dữ liệu $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ tuân theo phân phối hỗn hợp trên. Tìm ước lượng hợp lý cực đại cho tham số θ sao cho mô hình GMM là phù hợp nhất.
- Nghiệm tối ưu θ^*

$$\theta^* = \arg\max_\theta p(\mathbf{X}|\theta) = \arg\max_\theta \prod_{n=1}^N p(\mathbf{x}_n|\theta)$$

- Việc giải trực tiếp phương trình đạo hàm bằng 0 cho hàm logarith $\log p(\mathbf{X}|\theta)$ như trường hợp chỉ có 01 cụm là bất khả thi!

EM Algorithm

Thực tế cần sử dụng thuật toán EM (**Expectation-Maximization**) để giải lặp tìm θ tối ưu. Các bước của thuật toán:

- Xuất phát từ θ^0 nào đó.
- Từ θ^t, $t > 0$ - Lặp lại quá trình tính toán ước lượng hợp lý để cập nhật θ^{t+1}.
- Mỗi bước lặp có 02 thao tác (02 bước huấn luyện)
  - **E-Step**: Ước lượng phân phối của biến ẩn z_c thể hiện phân phối xác suất của các cụm c tương ứng với dữ liệu và bộ tham số phân phối.
  - **M-Step**: Cực đại hoá phân phối xác suất đồng thời (*join distribution probability*) của dữ liệu **X** và biến ẩn **Z**.

Do vậy thuật toán lặp này có tên gọi là EM - Expectation Maximization.

EM Algorithm

- Để cập nhật θ tại bước lặp $t > 0$, thay cho hàm ước lượng hợp lý, ta xét hàm tích lũy (*auxilary*):

$$
\begin{aligned}
Q(\theta, \theta_t) &= \mathbb{E}_z(\log p(\mathbf{X}, \mathbf{Z}|\theta_t)) = \sum_z p(z|\mathbf{X}, \theta_t) \log p(\mathbf{X}, \mathbf{Z}|\theta) \\
&= \sum_z p(z|\mathbf{X}, \theta_t) \log [\ p(\mathbf{Z}|\mathbf{X}, \theta) p(\mathbf{X}|\theta)] \\
&= \sum_z p(z|\mathbf{X}, \theta_t) \log p(\mathbf{Z}|\mathbf{X}, \theta) + \underbrace{\left[\sum_z p(z|\mathbf{X}, \theta)\right]}_{1} \log p(\mathbf{X}|\theta) \\
&= \sum_z p(z|\mathbf{X}, \theta_t) \log p(\mathbf{Z}|\mathbf{X}, \theta) + \log p(\mathbf{X}|\theta)
\end{aligned}
$$

EM Algorithm

- $Q(\theta, \theta_t)$ là kì vọng của logarith xác suất chung của $\mathbf{X}$ và $\mathbf{Z}$ và bằng tổng (có trọng số) của $p(z|\mathbf{X}, \theta_t)$ - xác suất tiên nghiệm trên từng cụm.
- Ta sẽ chỉ ra $Q(\theta, \theta_t)$ tăng kéo theo hàm ước lượng hợp lý tăng:

$$
\begin{aligned}
Q(\theta, \theta_t) - Q(\theta_t, \theta_t) &= \log p(\mathbf{X}|\theta) - \log p(\mathbf{X}|\theta_t) - \sum_z p(z|\mathbf{X}, \theta_t) \log \frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{p(\mathbf{Z}|\mathbf{X}, \theta_t)} \\
&= \log p(\mathbf{X}|\theta) - \log p(\mathbf{X}|\theta_t) - \underbrace{\mathrm{KL}(p(\mathbf{Z}|\mathbf{X}, \theta), p(\mathbf{Z}|\mathbf{X}, \theta_t))}_{\geq 0} \\
&\leq \log p(\mathbf{X}|\theta) - \log p(\mathbf{X}|\theta_t)
\end{aligned}
$$

do tính chất $\sum_z p(z|\mathbf{X}, \theta_t) \log \frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{p(\mathbf{Z}|\mathbf{X}, \theta_t)}$ là khoảng cách Kullback - Leibler - Divergence giữa hai phân phối. Vậy $Q(\theta, \theta_t)$ tăng thì $\log p(\mathbf{X}|\theta)$ tăng.

EM Algorithm

- Xác suất xảy ra tại một điểm dữ liệu có thể được biểu diễn:

$$p(\mathbf{x}_n, \mathbf{z}|\theta) = \prod_{c=1}^{C} [p(\mathbf{x}_n, z_c|\theta)]^{z_c} = \prod_{c=1}^{C} [p(\mathbf{x}_n|z_c, \theta)p(z_c|\theta)]^{z_c} = \prod_{c=1}^{C} [p(\mathbf{x}_n|z_c, \theta)\pi_c]^{z_c}$$

- Hàm hợp lý của phân phối xác suất đồng thời của $\mathbf{X}$ và $\mathbf{Z}$ có thể được viết

$$\mathcal{L}(\mathbf{X}, \mathbf{Z}|\theta) = p(\mathbf{X}, \mathbf{Z}|\theta) = \prod_{n=1}^{N} \prod_{c=1}^{C} [p(\mathbf{x}_n, z_c|\theta)]^{z_c} = \prod_{n=1}^{N} \prod_{c=1}^{C} [p(\mathbf{x}_n|z_c, \theta)\pi_c]^{z_c}$$

- Do đó

$$\log[p(\mathbf{X}, \mathbf{Z})] = \sum_{n=1}^{N} \sum_{c=1}^{C} z_c \log p(\mathbf{x}_n|z_c, \theta) + z_c \log \pi_c$$

EM Algorithm

Ta suy ra

$$
\begin{aligned}
Q(\theta, \theta_t) &= \mathbb{E}_z \left[\log p(\mathbf{X}, \mathbf{Z})|\theta_t\right] \\
&= \mathbb{E}_z \left[\sum_{n=1}^{N}\sum_{c=1}^{C} z_c \log p(\mathbf{x}_n|z_c, \theta) + z_c \log \pi_c|\theta_t\right] \\
&= \sum_{n=1}^{N}\sum_{c=1}^{C} \mathbb{E}_z[z_c|\theta_t] \log p(\mathbf{x}_n|z_c, \theta) + \mathbb{E}_z[z_c|\theta_t] \log \pi_c \\
&= \sum_{n=1}^{N}\sum_{c=1}^{C} p(z_c|\mathbf{x}_n, \theta_t) \left[\log p(\mathbf{x}_n|z_c, \theta) + \log \pi_c\right] = \ldots
\end{aligned}
$$

EM Algorithm

$$
\begin{aligned}
Q(\theta, \theta_t) &= \ldots \\
&= \sum_{n=1}^{N} \sum_{c=1}^{C} p(z_c|\mathbf{x}_n, \theta_t) \left[\log \frac{\exp\left(-\frac{1}{2}(\mathbf{x}_n - \mu_c)^\intercal \Sigma_c^{-1}(\mathbf{x}_n - \mu_c)\right)}{(2\pi)^{d/2}|\Sigma_c|^{1/2}} + \log \pi_c \right] \\
&= \sum_{n=1}^{N} \sum_{c=1}^{C} p(z_c|\mathbf{x}_n, \theta_t) \Bigg[-\frac{1}{2}\log|\Sigma_c| - \frac{1}{2}(\mathbf{x}_n - \mu_c)^\intercal \Sigma_c^{-1}(\mathbf{x}_n - \mu_c) \\
&\qquad + \log \pi_c + \text{Const}_c \Bigg]
\end{aligned}
$$

EM Algorithm

E-step:

$$\begin{aligned}\mathbb{E}_z(z_c|\mathbf{x}_n,\theta_t) &= 1\times p(z_c=1|\mathbf{x}_n,\theta_t)+0\times p(z_c=0|\mathbf{x}_n,\theta_t)\\ &= p(z_c|\mathbf{x}_n,\theta_t) = \frac{p(z_c|\theta_t)p(\mathbf{x}_n|z_c,\theta_t)}{p(\mathbf{x}_c|\theta_t)}\\ &= \frac{\pi_c N(\mu_{ct},\Sigma_{ct}|\mathbf{x}_n)}{\sum_{c=1}^{C}\pi_c N(\mu_{ct},\Sigma_{ct}|\mathbf{x}_n)}\end{aligned}$$

- π_c chính là xác suất tiên nghiệm (posteriori probability) bằng với tỷ lệ các quan sát thuộc về cụm c ở vòng lặp thứ t - xem lại cách tính ở phần Naive Bayes.
- $N(\mu_{ct},\Sigma_{ct}|\mathbf{x}_n)$ là xác suất $\mathbf{x}_n$ rơi vào cụm c theo công thức phân phối Gaussian.

EM Algorithm

M-step: Tính toán và cho đạo hàm $\frac{\partial Q(\theta, \theta_t)}{\partial \theta} = 0$, chú ý θ là bộ tham số $\{\pi_c, \mu_c, \Sigma_c\}_{c=1}^{C}$. Ta tính đạo hàm theo từng biến μ_c và Σ_c như phần trước:

$$\frac{\partial Q(\theta, \theta_t)}{\partial \mu_c} = \frac{\partial}{\partial \mu_c} \sum_{n=1}^{N} \sum_{c=1}^{C} p(z_c|\mathbf{x}_n, \theta_t) \Bigg[-\frac{1}{2} \log |\Sigma_c| - \frac{1}{2} (\mathbf{x}_n - \mu_c)^\intercal \Sigma_c^{-1} (\mathbf{x}_n - \mu_c) + \log \pi_c + C_c \Bigg]$$

$$= \frac{\partial}{\partial \mu_c} p(z_c|\mathbf{x}_n, \theta_t) \left[\sum_{n=1}^{N} \Sigma_c^{-1} (\mu_c - \mathbf{x}_n) \right]$$

$$= \frac{\partial}{\partial \mu_c} \Sigma_c^{-1} \left[\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)(\mu_c - \mathbf{x}_n) \right] = 0$$

EM Algorithm

M-step:

Giải phương trình $\frac{\partial Q(\theta, \theta_t)}{\partial \mu_c} = 0$ ta được nghiệm tối ưu

$$\mu_c^* = \frac{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)\mathbf{x}_n}{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)}.$$

Chú ý ở đây $p(z_c|\mathbf{x}_n, \theta_t)$ là xác suất để $\mathbf{x}_n$ rơi vào cụm c mà ta đã tính được trong bước E-step.

EM Algorithm

M-step:

Tiếp theo ta tính đạo hàm theo Σ_c và giải phương trình đạo hàm bằng 0:

$$\frac{\partial Q(\theta, \theta_t)}{\partial \Sigma_c^{-1}} = \frac{\partial}{\partial \mu_c} \sum_{n=1}^{N} \sum_{c=1}^{k} p(z_c|\mathbf{x}_n, \theta_t) \Bigg[-\frac{1}{2} \log |\Sigma_c| - \frac{1}{2} (\mathbf{x}_n - \mu_c)^\intercal \Sigma_c^{-1} (\mathbf{x}_n - \mu_c) + \log \pi_c + \text{Const}_c \Bigg]$$

$$= \sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t) \left[\frac{1}{2} \Sigma_c - \frac{1}{2} (\mathbf{x}_n - \mu_c)(\mathbf{x}_n - \mu_c)^\intercal \right] = 0$$

Suy ra tham số Σ_c tối ưu

$$\Sigma_c^* = \frac{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)[(\mathbf{x}_n - \mu_c)(\mathbf{x}_n - \mu_c)^\intercal]}{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)}.$$

EM Algorithm

M-step:
Vậy tham số tối ưu ở mỗi cụm c sẽ được cập nhật tại bước thứ t theo công thức:

$$\mu_c^* = \frac{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)\mathbf{x}_n}{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)};$$

$$\Sigma_c^* = \frac{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)[(\mathbf{x}_n - \mu_c)(\mathbf{x}_n - \mu_c)^\intercal]}{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)}.$$

EM Algorithm

M-step:
Tuy nhiên để tính toán cho bước tiếp theo $t+1$, ta còn cần tính các xác suất π_c, $c = 1, 2, \ldots, C$. Do có các ràng buộc trên π_c:

$$\pi_c \geq 0 \quad \forall c = 1, \ldots, C \quad \text{và} \quad \sum_{c=1}^{C} \pi_c = 1$$

nên ta dùng phương pháp nhân tử Lagrange. Hàm Lagrange ứng với $Q(\theta, \theta_t)$ là:

$$J(\theta, \theta_t) = Q(\theta, \theta_t) + \lambda \left[1 - \sum_{c=1}^{C} \pi_c \right]$$

EM Algorithm

M-step:
Lấy đạo hàm và cho bằng 0 ta được

$$\frac{\partial J(\theta, \theta_t)}{\partial \pi_c} = \frac{\partial Q(\theta, \theta_t)}{\partial \pi_c} - \lambda = \frac{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)}{\pi_c} - \lambda = 0.$$

Đẳng thức cuối suy ra $\pi_c = \frac{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)}{\lambda}$, kết hợp với điều kiện $\sum_{c=1}^{C} \pi_c = 1$ ta được $\sum_{j=1}^{k} \pi_j = \frac{\sum_{i=1}^{N} \sum_{j=1}^{k} p(z_j|\mathbf{x}_i, \theta_t)}{\lambda} = \frac{N}{\lambda} = 1$ hay $\lambda = N$ và thu được tham số tối ưu:

$$\pi_c^* = \frac{\sum_{n=1}^{N} p(z_c|\mathbf{x}_n, \theta_t)}{N}.$$

Đến đây ta có đầy đủ các ước lượng để tính cho bước lặp tiếp theo $t + 1$.

k-means Algorithm

- Trong thuật toán EM ở trên, từ khởi tạo ban đầu cho C cụm dữ liệu, tại mỗi bước ta thực hiện các thao tác
  - Với C cụm được phân từ bước trước, tính kỳ vọng mẫu μ_c và phương sai mẫu Σ_c cho dữ liệu trong mỗi cụm c.
  - Xét từng điểm dữ liệu, tính xác suất để điểm đó thuộc về mỗi cụm $c \in [1..C]$. Từ đó xác định lại việc mỗi điểm dữ liệu thuộc về cụm nào và chuyển sang bước lặp tiếp theo.
- Giả sử dữ liệu thuộc một không gian vector, độ sai lệch giữa các dữ liệu quan sát có thể tính theo khoảng cách Euclide;
- Lúc đó có thể xây dựng thuật toán phân cụm tương tự EM nhưng đơn giản hơn - **Thuật toán k-means**.

k-means Algorithm

Thuật toán k-means

- Cho tập dữ liệu $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$ và số cụm cần phân chia $0 < k < N$.
- Khởi tạo: Lấy k điểm dữ liệu khác nhau $c_1, c_2, \ldots, c_k$ từ tập $\mathbf{X}$ - là các điểm trung tâm mỗi cụm. Tại bước lặp thứ t:
  - Gán điểm dữ liệu vào cụm:
    - Với k tâm cụm từ bước trước, với mỗi điểm dữ liệu $\mathbf{x}_n$, tính các khoảng cách $d_{n,j} = \text{dist}(\mathbf{x}_n, c_j)$ từ $\mathbf{x}_n$ đến tâm cụm hiện tại c_j.
    - Giả sử l là chỉ số cụm có $d_{n,l} = \text{dist}(\mathbf{x}_n, c_l)$ nhỏ nhất.
    - Gán điểm $\mathbf{x}_n$ vào cụm thứ l, tức cụm có tâm hiện tại là c_l.
    - Sau thao tác gán, dữ liệu $\mathbf{X}$ được chia vào các cụm $S_1, S_2, \ldots, S_k$.
  - Xác định lại tâm cụm: Tâm cụm c_j sẽ được xác định lại là trung bình cộng các điểm trong cụm S_j đó: $c_j^{new} := \dfrac{\sum_{x_n \in S_j} x_n}{|S_j|}$. Gán $t = t + 1$ và lặp lại.
- Thuật toán dừng khi tâm cụm tìm được ở bước $t + 1$ hoàn toàn trùng với tâm cụm tìm được ở bước t.

k-means Algorithm

Một số lưu ý khi thực hiện thuật toán

- Số cụm k phù hợp nhất: Tùy theo số điểm dữ liệu N. Số cụm k tối ưu thường được chọn bằng các xét một dãy các giá trị của k và tìm giá trị phù hợp nhất.
- Chọn tâm cụm ở bước khởi tạo
  - Không chọn các điểm khởi tạo quá gần nhau.
  - Chọn điểm c_1 bất kỳ
  - Chọn c_2 ở xa c_1 nhất có thể (theo khoảng cách Euclide).
  - Chọn c_3 sao cho $\text{dist}(c_1, c_3) + \text{dist}(c_2, c_3)$ lớn nhất có thể.
  - Tiếp tục như vậy cho đến tâm cụm chỉ số k.