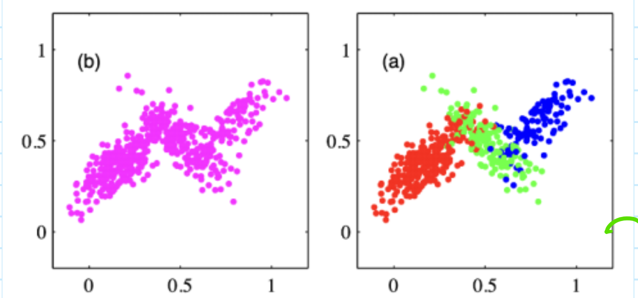
Lecture 4: GMM

I, Soft vs Hard Assignment

- **Hard assignment:** Mỗi điểm chỉ được assign cho 1 cluster một cách rõ ràng, Kmeans là một thuật toán hard clustering. However, for point near the decision boundary, this may not be such a good idea.
- **Soft assignment:** there will be possibility of a point belong to each cluster and the point will be assigned based on that possibility. GMM is a soft assignment algo.

II, Gaussian Mixture Model



- GMM models data as a combination of Gaussians, tức là có nhiều Gaussian distribution trong data như hình.

- The mixture of Gaussians is a generative model (mô hình sinh: từ dữ liệu dựng nên cấu trúc model; >< discriminative model: cho sẵn giả thiết về dữ liệu và model)
- Tóm tắt model: Từ data gốc ta muốn mô phỏng một cấu trúc Gaussian mixture bằng cách tìm ra các tham số μ, Σ của các Gaussian distribution và size của các distribution đó trong bộ dữ liệu.
- Cho một điểm dữ liệu x_n, để xuất hiện điểm đó trong training set
  - +, Cần tạo một giá trị discrete $z_n \in \{1, \dots K\}$ mô tả điểm đó thuộc cluster nào.
  - +, Sau đó x_n được tạo ra theo cluster mà nó thuộc về bằng cách cho x_n theo distribution của cluster đó. Trong trường hợp này là $x_n \sim \mathcal{N}(x \mid \mu_K, \Sigma_K)$.

→ Như vậy, để tìm được các Gaussians trong dataset, ta sẽ mô phỏng cách các điểm đc tạo thành ở 2 bước nêu trên rồi tìm cách optimize để tìm đc các tham số z, μ, Σ của mô hình giả định đó.

(*) Mô tả các ký hiệu trong model

- Biến z_n là vector đại diện cho việc điểm x_n thuộc cluster nào trong đó $z_n = (z_{n1}, z_{n2} \dots z_{nK})$ với $z_{nk} \in \{0, 1\}$ (VD: $z_n = (0, 1, \dots 0)$)

↳ z_n thực chất là latent variable (biến ngầm), nó được sử dụng để việc formulate bài toán trở nên tường minh hơn chứ ko phải tham số mà model cần tìm hay biến đầu ra nào. Nó cũng là unobserved variable, ta đang giả định nó để build mô hình rồi tìm tham số chứ ko mang ý nghĩa là gtrị quan sát hay đc suy ra từ gtri quan sát thực.

III, Problem Formulation

- Mục tiêu của GMM là generate ra cấu trúc các Gaussians trong dữ liệu, về mặt toán học, ta sẽ đi tìm $p(x)$ là xác suất một điểm trong dataset và $p(x_i) = p(x_i \text{ thuộc cluster } k) \times p(\text{vị trí của } x_i \text{ trong cluster } k) = p(z_i) \times \mathcal{N}(x_i \mid \mu_k, \Sigma_k)$. Công thức của các $p(x_i)$ phản ánh đc cấu trúc dữ liệu gồm size (probability) của các cluster ($p(z)$) cũng như prob density function của từng cluster ($\mathcal{N}(x \mid \mu, \Sigma)$) nên đó là ý nghĩa của việc tại sao ta cần formulate và optimize dựa trên nó (in my opinion tho).

⇒ Vậy trong phần này sẽ trình bày các bước để derive $p(x)$ formula

⊕ Ta có $p(z_{nk})$ là prior probability của mô hình, thể hiện xác suất 1 điểm thuộc cluster k.

- Do z_{nk} chỉ nhận gtrị 0 hoặc 1 nên ta sẽ sử dụng phân phối Bernoulli cho nó. Ta kí hiệu $\pi_k = p(z_{nk} = 1)$ là xsuất 1 điểm bất kỳ thuộc cluster k, hay thực chất là size của cluster k trong dataset. π_k là parameter của mô hình.
- Tính chất $0 \le \pi_k \le 1$ và $\sum \pi_k = 1$
- Vậy $p(z_n) = \prod \pi_k^{z_{nk}}$ — $p(z_n)$ nôm na là xsuất điểm x_n thuộc 1 cụm bất kỳ. Trong các z_{nk} chỉ có 1 cái bằng 1 nên thực chất đây là cthuc $p(z_{nk})$ nhưng viết tổng quát cho mọi điểm dữ liệu.

⊕ Tiếp theo là $p(x_n \mid z_n)$: nghĩa là xsuất điều kiện: đã cho $x_n \in$ cụm k xác suất x_n nằm ở đâu trong cụm đó (vị trí gần hay xa tâm μ_k).

$$\Rightarrow p(x_n \mid z_{nk} = 1) = \mathcal{N}(x_n \mid \mu_k, \Sigma_k)$$

$$\Leftrightarrow p(x_n \mid z_n) = \prod_{k=1}^{K} \mathcal{N}(x_n \mid \mu, \Sigma)^{z_{nk}}$$ → cthuc tổng quát cho mọi đ' và mọi cluster. (K: số cluster)

⊕ Vậy ta tìm được joint distribution $p(x, z)$ như sau:

$$p(x, z) = \prod_{k=1}^{K} p(z_n)\, p(x_n \mid \mu_k, \Sigma_k) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}}\, \mathcal{N}(x_n \mid \mu_k, \Sigma_k)^{z_{nk}}$$

⊕ Vậy xác suất marginal $p(x_n)$ đã nêu trên sẽ đc tính bằng

$$p(x_n) = \Sigma_k\, p(x_n, z_{nk}) = \Sigma_k\, p(z_{nk})\, p(x_n \mid z_{nk})$$
$$= \sum_{k=1}^{K} \pi_k\, \mathcal{N}(x_n \mid \mu_k, \Sigma_k)$$

⊕ Đến bước này ta thấy được các parameter của mô hình gồm $\{\pi_k, \mu_k, \Sigma_k\}$ và ý tưởng rằng:

→ Nếu biết z_n, fitting Gaussian is easy

→ Nếu biết các phân phối vs μ_k, Σ_k, π_k tương ứng, finding z_n is easy

→ ý tưởng optimize song song tương tự như Kmeans

⊕ Thay vì dùng latent variable z_n, giờ ta sẽ sử dụng khái niệm responsibility hay chính là x suất $p(z_{nk} = 1 \mid x_n)$. kí hiệu:

$$\begin{aligned}
\gamma(z_{nk}) &= p(z_{nk} = 1|x_n) \\
&= \frac{p(z_{nk}1)p(x_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(x_n|z_{nj} = 1)} \\
&= \boxed{\frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}}
\end{aligned}$$

→ $\gamma(n_k)$ là responsibility của cluster k cho điểm x_n

⊕ Bây giờ ta đã xác định được bộ tham số $\theta = \{\pi_k, \mu_k, \Sigma_k\}$

Objective function là hàm log likelihood của data vì ta cần tìm các Gaussians có thể mô phỏng chính xác bộ dữ liệu → maximum likelihood criterion. Vậy objective function là:

$$l(\theta) = \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\right) \rightarrow \theta = \text{argmax}\, l(\theta)$$

→ Bài toán optimization nên ta lấy đạo hàm riêng của $l(\theta)$ theo từng tham số bằng 0, ta được kqua:

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

→ nhận thấy 3 tham số đều phụ thuộc $\gamma(z_{nk})$ nên có thể áp dụng iterative scheme như Kmeans

III, GMM steps

- Đầu tiên ta khởi tạo ngẫu nhiên các parameter, sau đó iterate 2 bước sau: (Repeat until convergence)

+) E-step: Calculate responsibilities using current parameters

$$\gamma(z_{nk}) = \frac{p(z_{nk}1)p(x_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(x_n|z_{nj} = 1)}$$

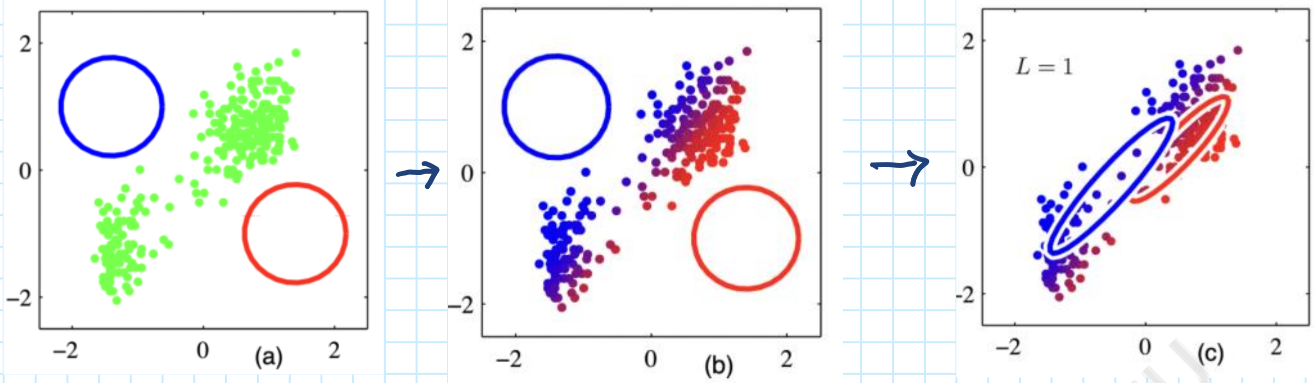
+) M-step: Re-estimate parameters using these $\gamma(z_{nk})$

$$\left\{\begin{aligned}
\mu_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \\
\Sigma_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T \\
\pi_k &= \frac{N_k}{N}
\end{aligned}\right.$$

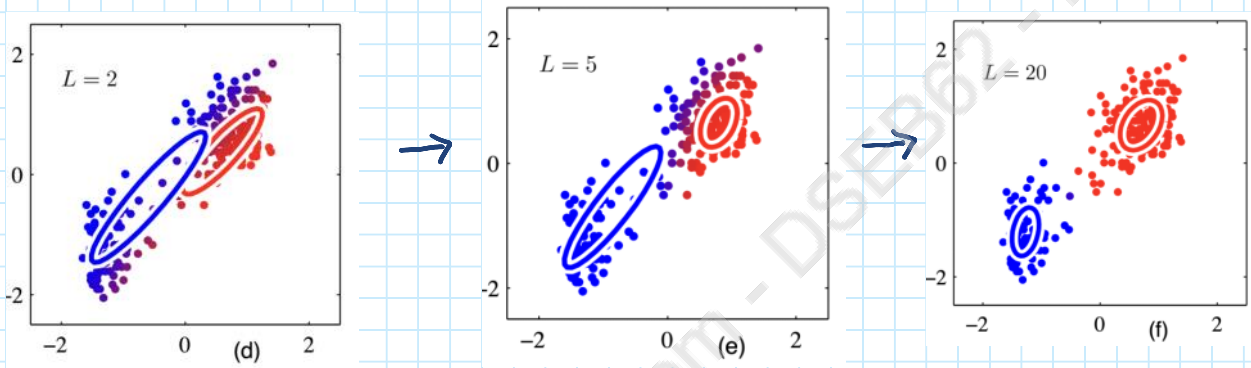
Hai bước E-step và M step như bên được gọi là Expectation-Maximization algorithm.

- Convergence criteria: thử số lần lặp ≠ nhau.

IV, Minh hoạ GMM



Khởi tạo param → Tính $\gamma(z_{nk})$ loop 1 → Tính param loop 1



Sau loop 2 → Sau loop 5 → Sau loop 20

V, GMM pros and cons

Advantages of Gaussian mixture models

- **Probabilistic estimates of belonging to each cluster**. One of the main advantages of gaussian mixture models is that they provide estimates of the probability that each data point belongs to each cluster. This provides a lot more contextual information than the standalone cluster assignment that most other clustering algorithms provide. These probability estimates can be very useful when examining ambiguous data points that fall at the border of two clusters.
- **Does not assume spherical clusters**. Another advantage that gaussian mixture models have over other models like k-means clustering is that they do not assume that all clusters are uniformly shaped spheres. Instead, gaussian mixture models can be used to accommodate clusters of varying shapes (so long as they are roughly elliptical).
- **Handles clusters of differing sizes**. In addition to being able to accommodate clusters of varying shapes, gaussian mixture models can also be used to accommodate clusters of varying sizes. This provides even more flexibility in the types of clusters that can be handled.
- **Less sensitive to scale**. Gaussian mixture models are generally less sensitive to scale than other clustering algorithms. That means that you may not need to rescale your variables before using them for clustering.

Disadvantages of Gaussian mixture models

- **Difficult to incorporate categorical features**. One of the main disadvantages of clustering with gaussian mixture models is that it is difficult to incorporate categorical variables. Gaussian mixture models operate under the assumption that all of your features are normally distributed, so they are not easily adapted to categorical data.
- **Assumes a normal distribution for features**. In addition to being struggling with categorical features, gaussian mixture models may also struggle with numeric variables that are not normally distributed. This means that you should take some time to look at the distributions of your features before reaching for this clustering algorithm.
- **Make some assumptions about cluster shape**. While gaussian mixture models are able to handle clusters of varying shapes and sizes, they do make some assumptions about the shape of the clusters. Specifically, the clusters are assumed to be elliptic. This means that gaussian mixture models will not perform as well in cases where clusters are very irregularly shaped.
- **Needs sufficient data for each cluster**. Since you need to estimate a covariance matrix in order to use gaussian mixture models, you should make sure that you have enough data points in each cluster to adequately estimate the covariance. The amount of data required is not huge, but it is larger than simple algorithms that do not estimate a covariance matrix.
- **Need to specify number of clusters**. Another disadvantage of gaussian mixture models is that you need to specify the number of clusters you want to use in your analysis ahead of time. This can be a non-trivial task when you do not have intuition about the number of clusters there should be.

- **Somewhat sensitive towards outliers**. Since gaussian mixture models operate under the assumption that your features are normally distributed, they can be thrown off by cases where there are many outliers in the data. That being said, some implementations of gaussian mixture models allow for outliers to be separated out into a separate cluster.
- **Somewhat sensitive to initialization conditions**. Gaussian mixture models are somewhat sensitive to initialization conditions of the algorithm such as the seed that is used and the starting points that are used for cluster centers. This means you may get different results if you run the algorithm multiple times.
- **Somewhat slow**. One final disadvantage of gaussian mixture models is that they tend to be slower than similar clustering algorithms like k-means clustering. This is especially true when there are many features in your dataset.