Lecture 11: Density Estimation with Gaussian Mixture Models

Yi, Yung (이융)

Mathematics for Machine Learning
https://yung-web.github.io/home/courses/mathml.html
KAIST EE

April 3, 2021

Warm-Up

Please watch this tutorial video by Luis Serrano on Gaussian Mixture Model.

https://www.youtube.com/watch?v=q71Niz856KE

Roadmap

(1) Gaussian Mixture Model

(2) Parameter Learning: MLE

(3) Latent-Variable Perspective for Probabilistic Modeling

(4) EM Algorithm

Roadmap

(1) Gaussian Mixture Model

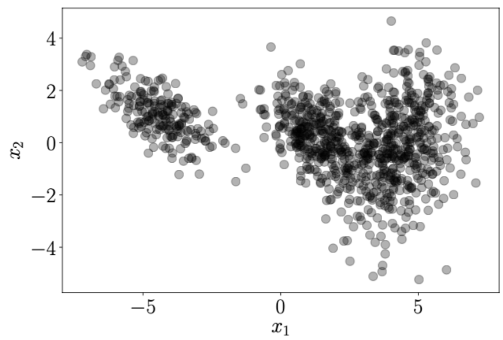
(2) Parameter Learning: MLE

(3) Latent-Variable Perspective for Probabilistic Modeling

(4) EM Algorithm

Density Estimation

- Represent data compactly using a density from a parametric family, e.g., Gaussian or Beta distribution
- Parameters of those families can be found by MLE and MAPE
- However, there are many cases when simple distributions (e.g., just Gaussian) fail to approximate data.



Mixture Models

- More expressive family of distribution
- Idea: Let's mix! A convex combination of K "base" distributions

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{x}), \quad 0 \le \pi_k \le 1, \quad \sum_{k=1}^{K} \pi_k = 1$$

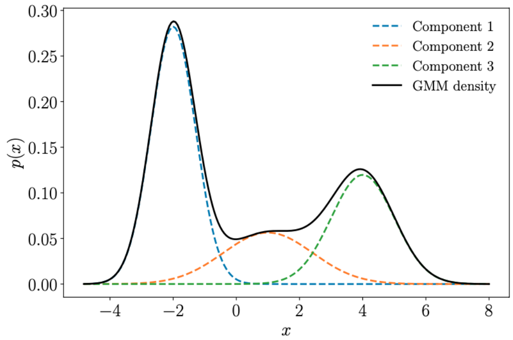
- Multi-modal distributions: Can be used to describe datasets with multiple clusters
- Our focus: Gaussian mixture models
- Want to finding the parameters using MLE, but cannot have the closed form solution (even with the mixture of Gaussians) $\rightarrow$ some iterative methods needed

Gaussian Mixture Model

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad 0 \le \pi_k \le 1, \quad \sum_{k=1}^{K} \pi_k = 1,$$

where the parameters $\boldsymbol{\theta} := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k : k = 1, \ldots, K\}$

- Example. $p(x|\boldsymbol{\theta}) = 0.5\mathcal{N}(x|-2, 1/2) + 0.2\mathcal{N}(x|1, 2) + 0.3\mathcal{N}(x|4, 1)$



Roadmap

(1) Gaussian Mixture Model

(2) Parameter Learning: MLE

(3) Latent-Variable Perspective for Probabilistic Modeling

(4) EM Algorithm

Parameter Learning: Maximum Likelihood

KAIST EE

- Given a iid dataset $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the log-likelihood is:

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathcal{X}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\mathbf{x}_n|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- $\boldsymbol{\theta}_{\text{ML}} = \arg\min_{\boldsymbol{\theta}}(-\mathcal{L}(\boldsymbol{\theta}))$
- Necessary condition for $\boldsymbol{\theta}_{\text{ML}}$: $\left.\frac{d\mathcal{L}}{d\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}_{\text{ML}}} = 0$
- However, the closed-form solution of $\boldsymbol{\theta}_{\text{ML}}$ does not exist, so we rely on an iterative algorithm (also called EM algorithm).
- We show the algorithm first, and then discuss how we get the algorithm.

Responsibilities

KAIST EE

- Definition. Responsibilities. Given n-th data point $\mathbf{x}_n$ and the parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k : k = 1, \ldots, K)$,

$$r_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- How much is each component k responsible, if the data $\mathbf{x}_n$ is sampled from the current mixture model?
- $\mathbf{r}_n = (r_{nk} : k = 1, \ldots, K)$ is a probability distribution, so $\sum_{k=1}^{K} r_{nk} = 1$
- Soft assignment of $\mathbf{x}_n$ to the K mixture components

EM Algorithm: MLE in Gaussian Mixture Models

EM for MLE in Gaussian Mixture Models

S1. Initialize $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$

S2. **E-step:** Evaluate responsibilities r_{nk} for every data point $\boldsymbol{x}_n$ using the current $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$:

$$r_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}, \quad N_k = \sum_{n=1}^{N} r_{nk}$$

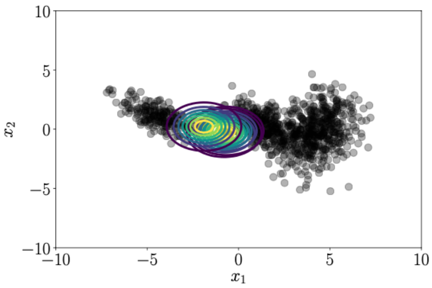
S3. **M-step:** Reestimate parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ using the current responsibilities r_{nk}:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n, \quad \boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}, \quad \pi_k = \frac{N_k}{N},$$

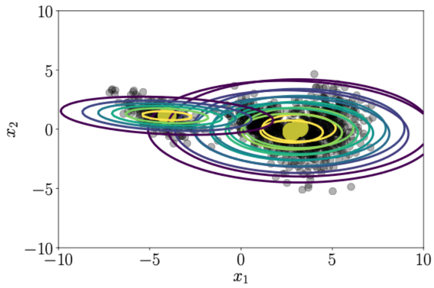
and go to **S2.**

- The update equation in **M-step** is still mysterious, which will be covered later.

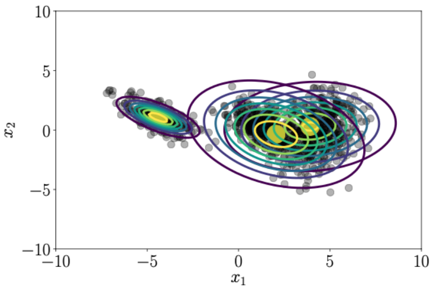
Example: EM Algorithm



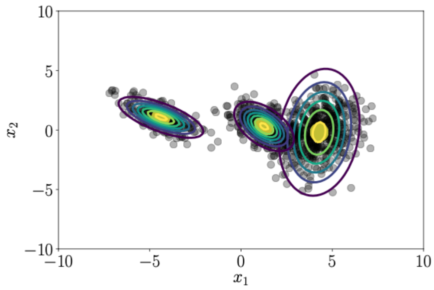
(c) EM initialization.



(d) EM after one iteration.



(e) EM after 10 iterations.



(f) EM after 62 iterations.

M-Step: Towards the Zero Gradient

- Given $\mathcal{X}$ and r_{nk} from E-step, the new updates of $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k should be made, such that the followings are satisfied:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = 0^{\mathsf{T}} \Longleftrightarrow \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = 0^{\mathsf{T}}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = 0 \Longleftrightarrow \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \pi_k} = 0 \Longleftrightarrow \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \pi_k} = 0$$

- Nice thing: the new updates of $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, π_k are all expressed by the responsibilities $[r_{nk}]$
- Let's take a look at them one by one!

M-Step: Update of $\boldsymbol{\mu}_k$

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n}{\sum_{n=1}^{N} r_{nk}}, k = 1, \ldots, K$$

-

M-Step: Update of Σ_k

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}, k = 1, \ldots, K$$

-

M-Step: Update of π_k

$$\pi_k^{\text{new}} = \frac{\sum_{n=1}^{N} r_{nk}}{N}, k = 1, \ldots, K$$

-

Roadmap

(1) Gaussian Mixture Model

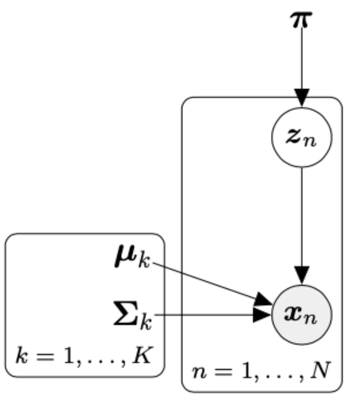
(2) Parameter Learning: MLE

(3) Latent-Variable Perspective for Probabilistic Modeling

(4) EM Algorithm

Latent-Variable Perspective

- Justify some ad hoc decisions made earlier
- Allow for a concrete interpretation of the responsibilities as posterior distributions
- Iterative algorithm for updating the model parameters can be derived in a principled manner



Generative Process

- Latent variable $\boldsymbol{z}$: One-hot encoding random vector $\boldsymbol{z} = [z_1, \ldots, z_K]^{\mathsf{T}}$ consisting of $K-1$ many 0s and exactly one 1.
- An indicator rv $z_k = 1$ represents whether k-th component is used to generate the data sample $\boldsymbol{x}$ or not.
- $p(\boldsymbol{x}|z_k = 1) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- Prior for $\boldsymbol{z}$ with $\pi_k = p(z_k = 1)$

$$p(\boldsymbol{z}) = \boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]^{\mathsf{T}}, \quad \sum_{k=1}^{K} \pi_k = 1$$

- Sampling procedure
  1. Sample which component to use $z^{(i)} \sim p(\boldsymbol{z})$
  2. Sample data according to i-th Gaussian $\boldsymbol{x}^{(i)} \sim p(\boldsymbol{x}|z^{(i)})$

Joint Distribution, Likelihood, and Posterior (1)

- Joint distribution

$$p(\boldsymbol{x}, \boldsymbol{z}) = \begin{pmatrix} p(\boldsymbol{x}, z_1 = 1) \\ \vdots \\ p(\boldsymbol{x}, z_K = 1) \end{pmatrix} = \begin{pmatrix} p(\boldsymbol{x}|z_1 = 1)p(z_1 = 1) \\ \vdots \\ p(\boldsymbol{x}|z_K = 1)p(z_K = 1) \end{pmatrix} = \begin{pmatrix} \pi_1 \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ \vdots \\ \pi_K \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K) \end{pmatrix}$$

- Likelihood for an arbitrary single data $\boldsymbol{x}$: By summing out all latent variables[1],

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{\theta}) = \sum_{k=1}^{K} p(\boldsymbol{x}|\boldsymbol{\theta}, z_k = 1) p(z_k = 1|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- For all the data samples $\mathcal{X}$, the log-likelihood is:

$$\log p(\mathcal{X}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Compare: Page 7

[1] In probabilistic PCA, $\boldsymbol{z}$ was continuous, so we integrated them out.

Joint Distribution, Likelihood, and Posterior (2)

KAIST EE

- Posterior for the k-th z_k, given an arbitrary single data $\boldsymbol{x}$:

$$p(z_k = 1|\boldsymbol{x}) = \frac{p(z_k = 1)p(\boldsymbol{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- Now, for all data samples $\mathcal{X}$, each data $\boldsymbol{x}_n$ has $\boldsymbol{z}_n = [z_{n1}, \ldots, z_{nK}]^{\mathsf{T}}$, but with the same prior $\boldsymbol{\pi}$.

$$p(z_{nk} = 1|\boldsymbol{x}_n) = \frac{p(z_{nk} = 1)p(\boldsymbol{x}_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(\boldsymbol{x}_n|z_{nj} = 1)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = r_{nk}$$

- Responsibilities are mathematically interpreted as posterior distributions.

Roadmap

KAIST EE

(1) Gaussian Mixture Model

(2) Parameter Learning: MLE

(3) Latent-Variable Perspective for Probabilistic Modeling

(4) EM Algorithm

Revisiting EM Algorithm for MLE

S1. Initialize $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$

S2. **E-step:**

$$r_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

S3. **M-step:** Update $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$ using r_{nk} and go to **S2.**

- **E-step.** Expectation over $\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}$: Given the current $\boldsymbol{\theta}^{(t)} = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)$, calculates the expected log-likelihood

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}^{(t)}}[\log p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})] = \int \log p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}$$

- **M-step.** Maximization of the computation results in E-step for the new model parameters.

- Only guarantee of just local-optimum because the original optimization is not necessarily a convex optimization. L7(4)

Other Issues

- Model selection for finding a good K, e.g., using nested cross-validation
- Application: Clustering
  - K-means: Treat the means in GMM as cluster centers and ignore the covariances.
  - K-means: hard assignment, GMM: soft assignment
- EM algorithm: Highly generic in the sense that it can be used for parameter learning in general latent-variable models
- Standard criticism for MLE exists such as overfitting. Also, fully-Bayesian approach assuming some priors on the parameters is possible, but not covered in this notes.
- Other density estimation methods
  - Histogram-based method: non-parametric method
  - Kernel-density estimation: non-parametric method

Questions?

Review Questions

1)