Gaussian Mixture Models

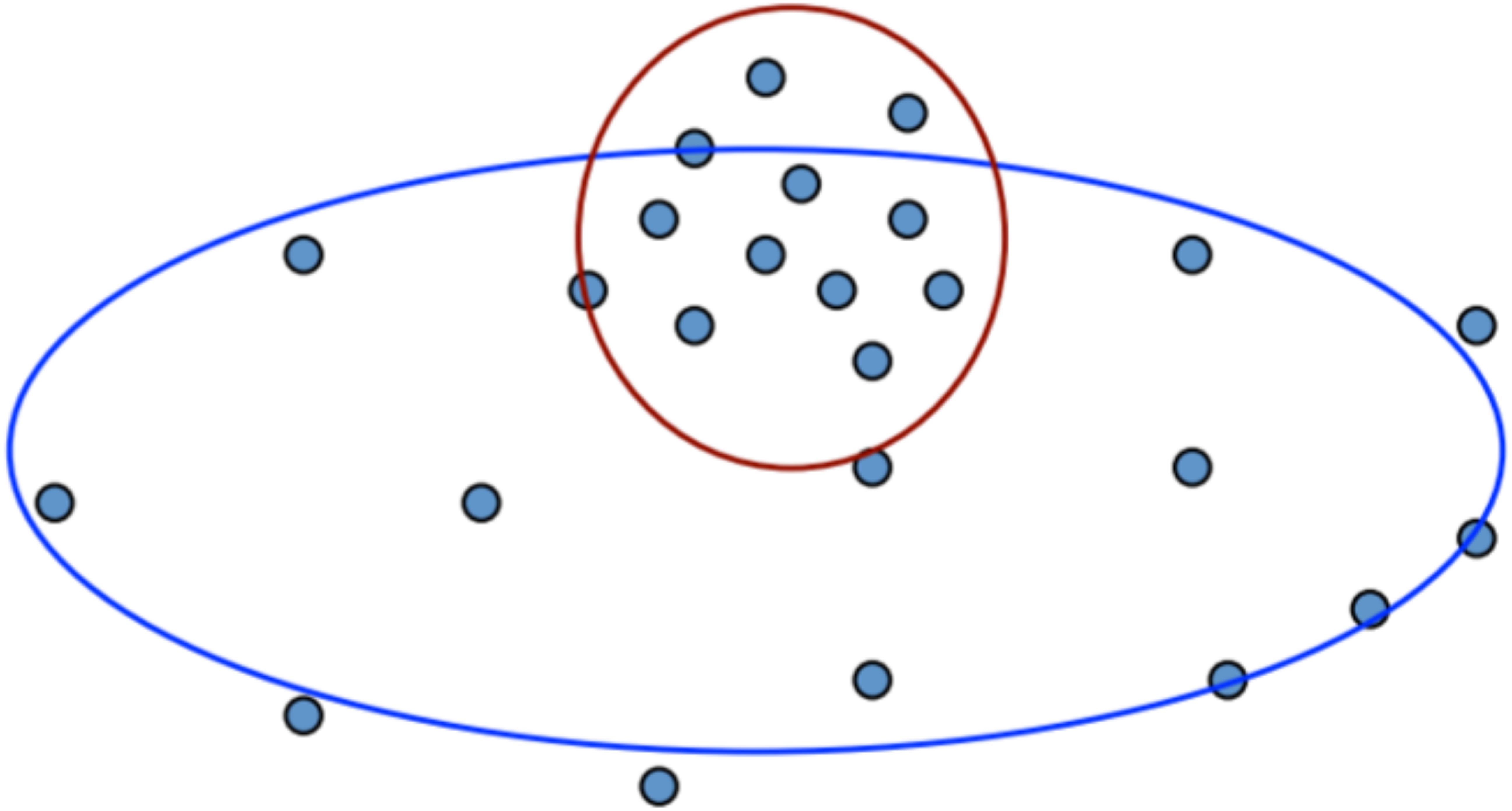
EECE454 Intro. to Machine Learning Systems

Fall 2024

Recap

- **K-means.** Each cluster is represented by centroid
  - Each datum belongs to a cluster with the nearest centroid
- **Limitations.** Plenty, e.g., cannot handle
  - Overlapping clusters
  - Wider cluster
  - Example. Residents in Pohang
    - Student vs. Locals

    => Take a more probabilistic approach



Mixture models

Mixture models

- **Idea.** Take a **generative approach**
  - Modeling
  - Fitting

Mixture models

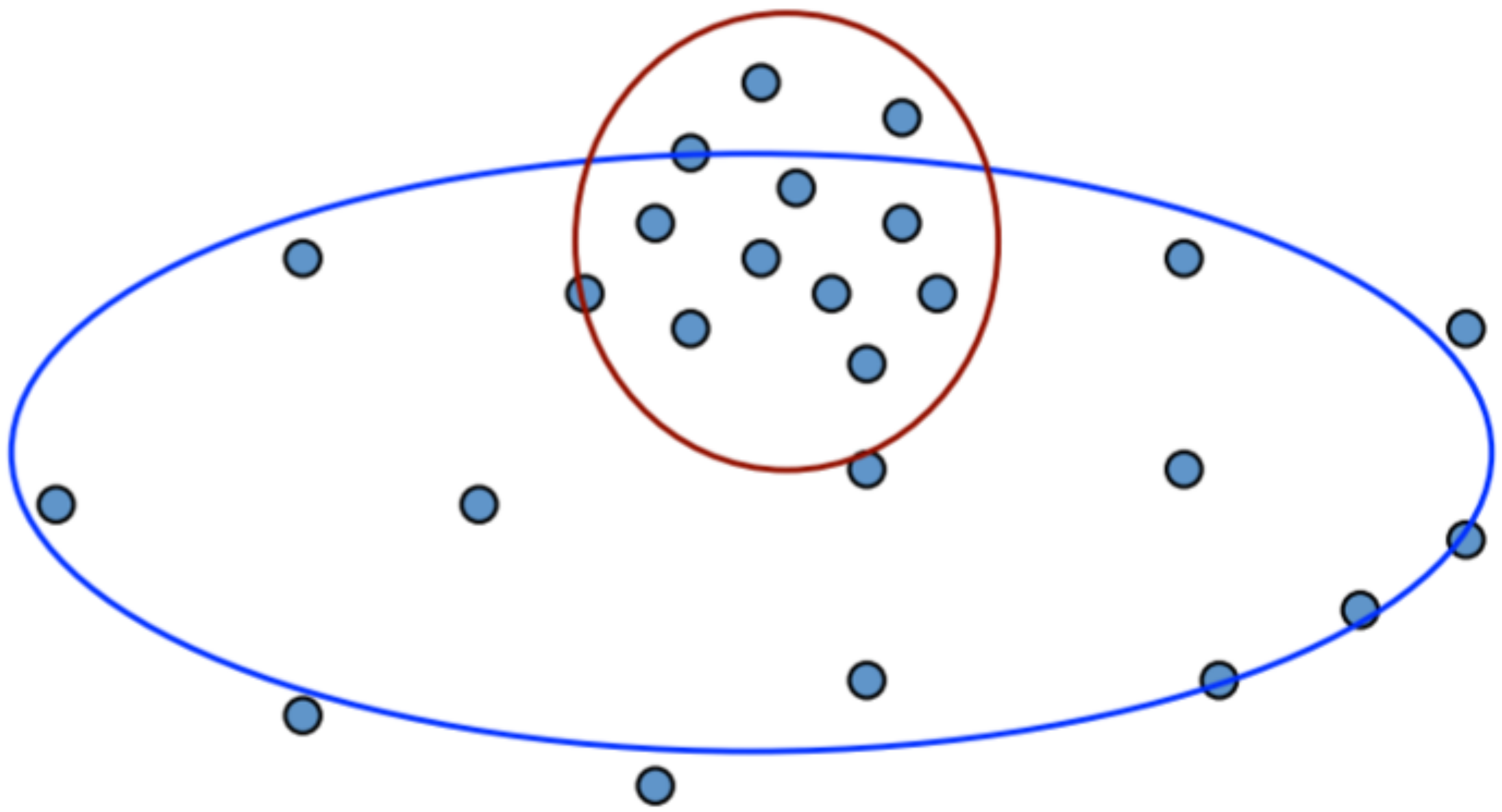
- **Idea.** Take a **generative approach**
  - Modeling: Model data generation with probability distributions
    - For mixture models:
      - $P_\phi(\text{cluster})$: (Latent) Group identity <— Not a "real" thing; a human artifact
      - $P_\theta(\text{feature} \mid \text{cluster})$: Data distribution of each cluster
  - Fitting

Mixture models

- **Idea.** Take a **generative approach**.
  - Modeling: Model data generation with probability distributions
    - For mixture models:
      - $P_\phi(\text{cluster})$: (Latent) Group identity <— Not a “real” thing; a human artifact
      - $P_\theta(\text{feature} \mid \text{cluster})$: Data distribution of each cluster
  - Fitting: Use training data to fit the parameters
      - $P_{\text{train}} \approx P_{\theta,\phi}(\text{feature})$

Mixture models

- Example. Previous example
  - Draw $Y \in \{0,1\} \sim \mathrm{Bern}(p)$ (0: local, 1: students)
    - If $Y = 0$, $X \sim \mathcal{N}(\mu_0, \sigma_0^2)$
    - If $Y = 1$, $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$
  - Allows overlap, account for wideness



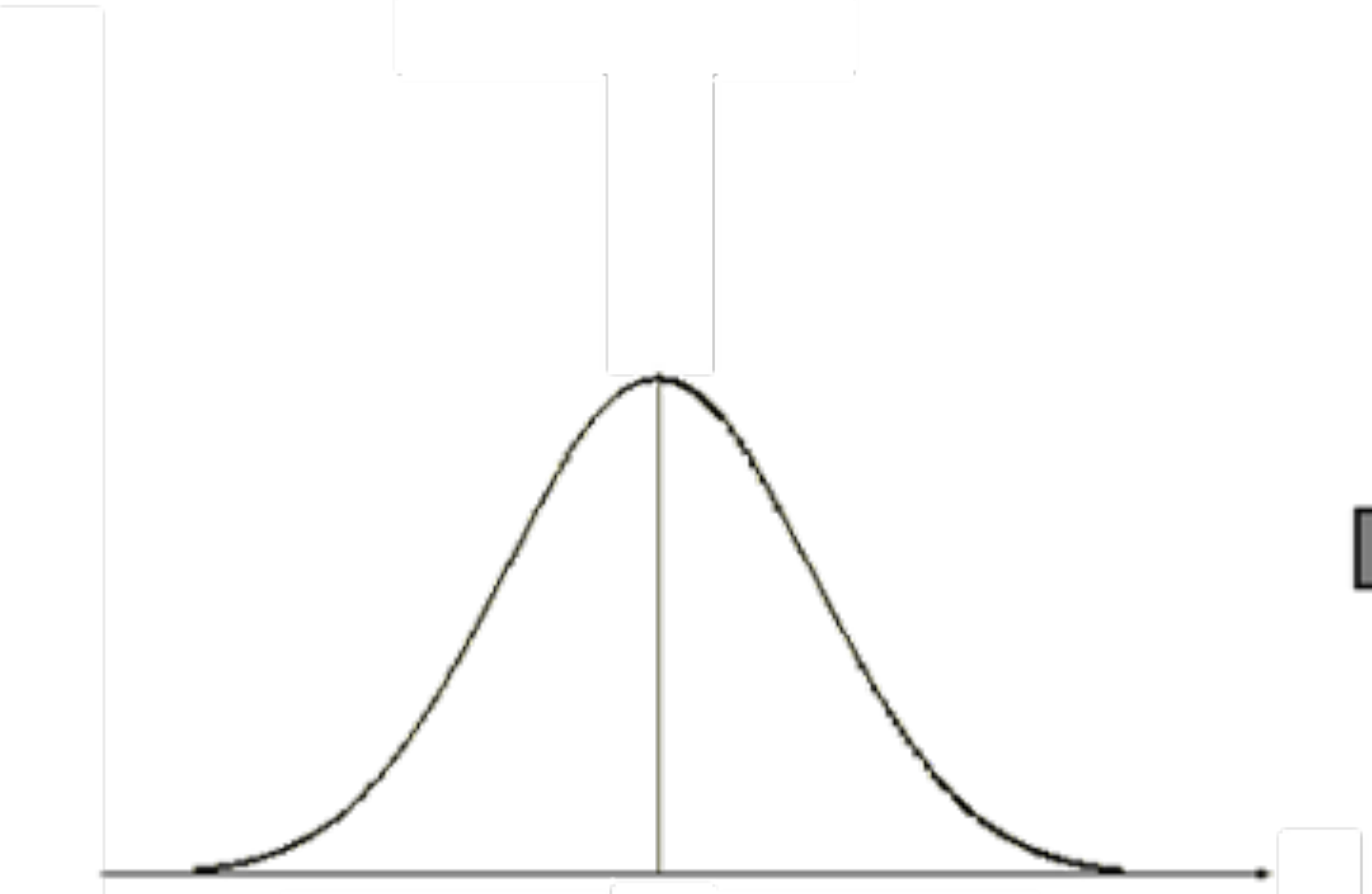
Generative approach

- **Perk.** If you have learned a nice probabilistic model from data,
  you can also sample a new data from $P_{\theta,\phi}(\,\cdot\,)$

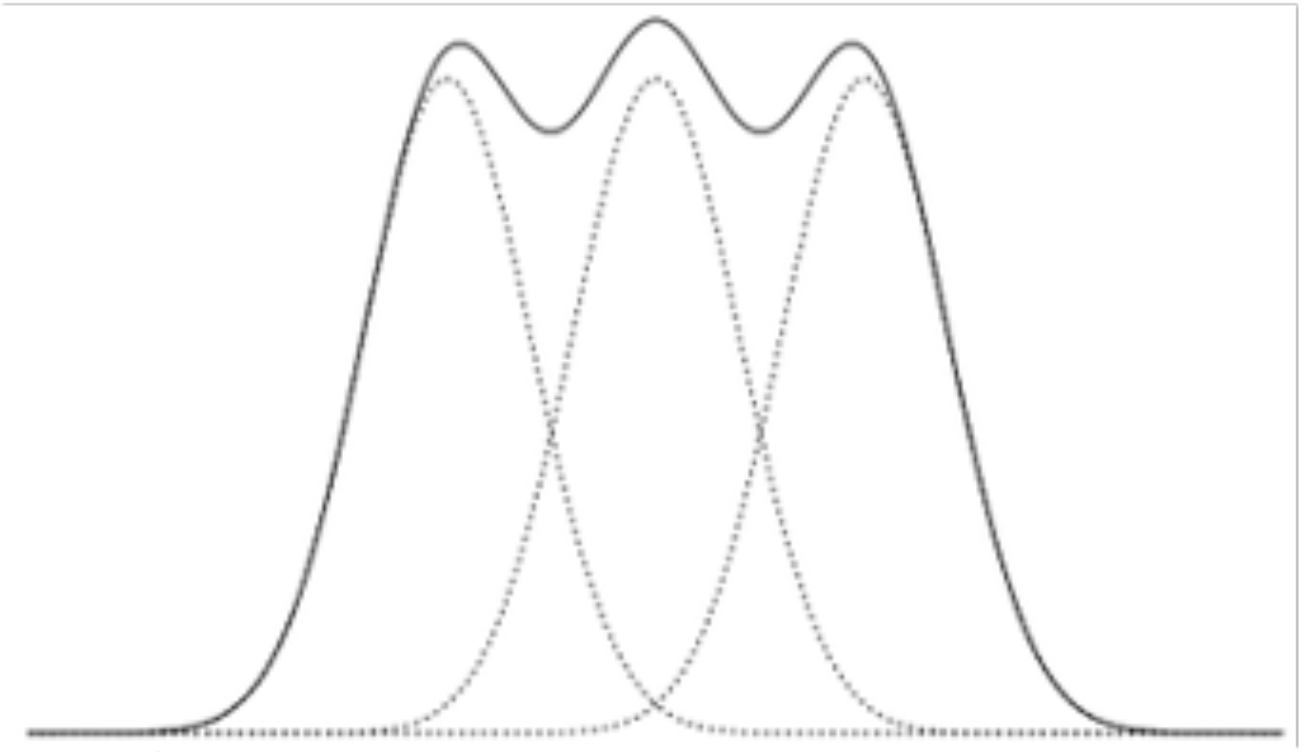
(finite) Mixture models

- **Mixture models.** A special set of generative models where $P(\,\cdot\,)$ takes the form:

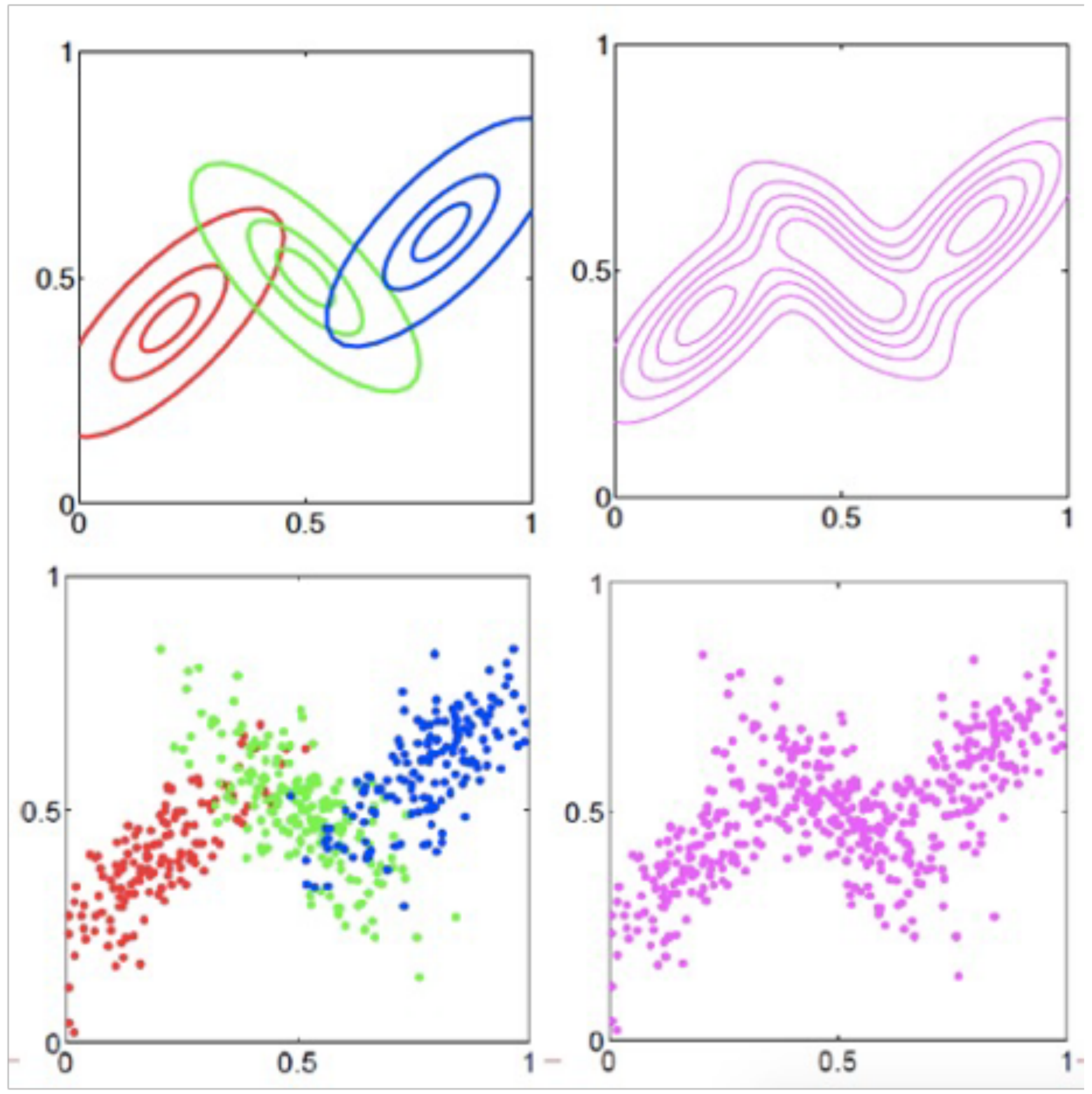
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \cdot p_k(\mathbf{x}), \qquad \pi_k \in [0,1], \sum \pi_k = 1.$$



Gaussian distribution



Mixture of Gaussian distributions



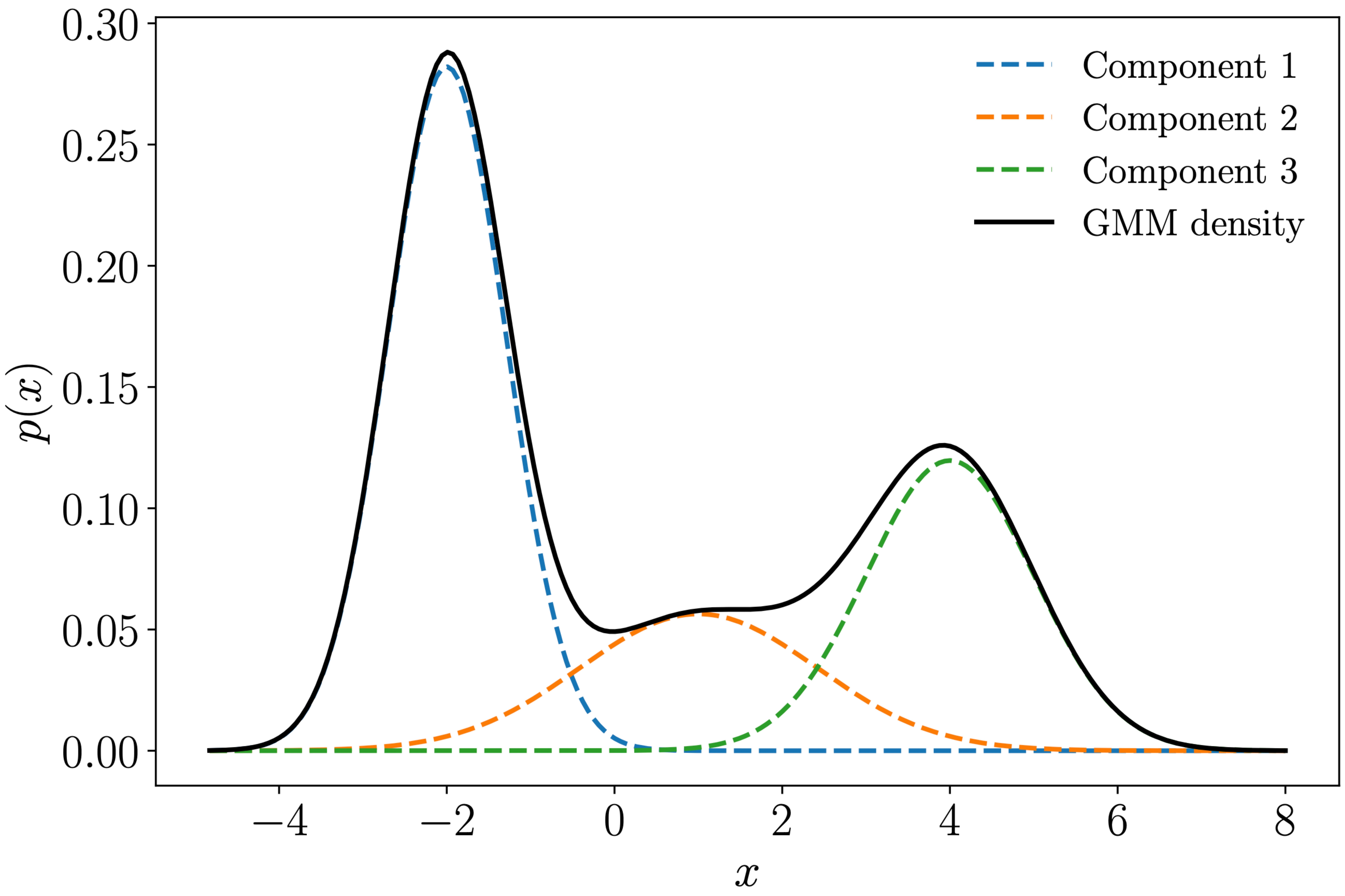
Gaussian Mixture models

- **Gaussian MM.** Each base distribuion p_k is a Gaussian distribution

$$p(\mathbf{x} \mid \theta) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$$

  - Here, $\theta = (\mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K, \pi_1, \ldots, \pi_K)$ is the total parameter set

Gaussian Mixture models



$$p(x \mid \boldsymbol{\theta}) = 0.5\mathcal{N}\left(x \mid -2, \tfrac{1}{2}\right) + 0.2\mathcal{N}\left(x \mid 1, 2\right) + 0.3\mathcal{N}\left(x \mid 4, 1\right)$$

Gaussian Mixture models

- **Gaussian MM.** Each base distribuion p_k is a Gaussian distribution

$$p(\mathbf{x}\,|\,\theta) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}\,|\,\mu_k, \Sigma_k)$$

  - Here, $\theta = (\mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K, \pi_1, \ldots, \pi_K)$ is the total parameter set

- **Question.** How do we fit the parameters, given the training data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$?

  - Note. Here, we do not care about "real" group identities.

Maximum likelihood

- **Basic strategy.** As in Naïve Bayes, we rely on the maximum likelihood principle:

Maximum likelihood

- **Basic strategy.** As in Naïve Bayes, we rely on the maximum likelihood principle:

  - Directly consider the likelihood for the mixture distribution

$$p(\mathbf{x}_{1:n} \mid \theta) = \prod_{i=1}^{n} p(\mathbf{x}_i \mid \theta)$$

$$= \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_{k(i)} \cdot \mathcal{N}(\mathbf{x}_i \mid \mu_{k(i)}, \Sigma_{k(i)})$$

  - Maximize this quantity by tuning $\theta = \{\mu_k, \Sigma_k, \pi_k \mid k \in [K]\}$

Maximum likelihood

- Transform to the usual log likelihood to make everything about summations:

$$\mathscr{L} := \log p(\mathbf{x}_{1:n} \mid \theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_i \mid \mu_k, \Sigma_k) \right)$$

- **Goal.** Solve $\max_{\theta} \mathscr{L}$

Maximum likelihood

- Transform to the usual log likelihood to make everything about summations:

$$\mathscr{L} := \log p(\mathbf{x}_{1:n} \,|\, \theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_i \,|\, \mu_k, \Sigma_k) \right)$$

  - **Goal.** Solve $\max_{\theta} \mathscr{L}$

- Normally, we would have tried to find optimum by critical point analysis
  - However, getting an closed-form solution is very difficult…
    - Give it a try, and let me know if you succeed

Expectation-Maximization

- **Idea.** Repeat the following steps:
  - Fix some variables and optimize others
  - Fix the optimized variables and optimize the previously fixed

Expectation-Maximization

- **Idea.** Repeat the following steps:
  - Fix some variables and optimize others
  - Fix the optimized variables and optimize the previously fixed
- Generally, this is a special case of **expectation-maximization (EM)** algorithm
  - Similar to what we did in K-means

Algorithm 1 k-means algorithm

Specify the number k of clusters to assign.
Randomly initialize k centroids.
repeat
    expectation: Assign each point to its closest centroid.
    maximization: Compute the new centroid (mean) of each cluster.
until The centroid positions do not change.

EM in K-means

- Recall that in Hard K-means:
  - Randomly initialize centroids $\{\mu_k\}$
  - Fix the centroid $\{\mu_k\}$ and optimize the assignment $\{r_{ik}\}$
    - Optimal, if <u>nearest neighbor</u>
  - Fix the assignment $\{r_{ik}\}$ and optimize the centroid $\{\mu_k\}$
    - Optimal, if <u>mean of the assigned data</u>
  - Repeat

EM in GMM

- Similarly, what we want to do is:
  - Randomly initialize parameters $\theta = \{\mu_k, \Sigma_k, \pi_k\}$
  - Fix the parameters θ and optimize the **responsibility** $\{r_{ik}\}$

    Non-binary, as in soft K-means
    - Optimal, if?
  - Fix the **responsibility** $\{r_{ik}\}$ and optimize the parameters θ
    - Optimal, if?
  - Repeat
- Let us think about the optimality conditions...

Recall: Multivariate Gaussian

- Multivariate Gaussians:

$$\mathcal{N}(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu)\right)$$

- By taking log, we get

$$\log \mathcal{N}(\mathbf{x} \mid \mu, \Sigma) = -\frac{1}{2} \cdot \left(d \log(2\pi) + \log|\Sigma| + (\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu)\right)$$

Recall: Responsibilities

- **Responsibilities.** How likely each data belongs to a certain cluster

  - Soft K-means. The softmax value

$$r_{ik} = \frac{\exp(-\beta \|\mathbf{x}_i - \mu_k\|_2^2)}{\sum_j \exp(-\beta \|\mathbf{x}_i - \mu_j\|_2^2)}$$

Recall: Responsibilities

- **Responsibilities.** How likely each data belongs to a certain cluster
  - Soft K-means. The softmax value

$$r_{ik} = \frac{\exp(-\beta \|\mathbf{x}_i - \mu_k\|_2^2)}{\sum_j \exp(-\beta \|\mathbf{x}_i - \mu_j\|_2^2)}$$

  - GMM. We use

$$r_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x} \,|\, \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x} \,|\, \mu_j, \Sigma_j)}$$

Recall: Responsibilities

- **Responsibilities.** How likely each data belongs to a certain cluster
  - Soft K-means. The softmax value

$$r_{ik} = \frac{\exp(-\beta \|\mathbf{x}_i - \mu_k\|_2^2)}{\sum_j \exp(-\beta \|\mathbf{x}_i - \mu_j\|_2^2)}$$

- GMM. We use

$p(y = k)$

$p(\mathbf{x} \mid \mathbf{y} = k)$

$$r_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x} \mid \mu_j, \Sigma_j)}$$

$p(\mathbf{x})$

$$p(y = k \mid \mathbf{x}) = \frac{p(\mathbf{x}, y = k)}{p(\mathbf{x})}$$

Recall: Responsibilities

- **Responsibilities.** How likely each data belongs to a certain cluster
  - Soft K-means. The softmax value

$$r_{ik} = \frac{\exp(-\beta\|\mathbf{x}_i - \mu_k\|_2^2)}{\sum_j \exp(-\beta\|\mathbf{x}_i - \mu_j\|_2^2)}$$

  - GMM. We use

$$r_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x} \,|\, \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x} \,|\, \mu_j, \Sigma_j)}$$

    - **Note.** If $\pi_k = 1/K$ and $\sigma_k = 1/\beta$, then this is identical to soft K-means

Optimality condition: Mean

- Recall that

$$\mathscr{L} := \log p(\mathbf{x}_{1:n} \mid \theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_i \mid \mu_k, \Sigma_k) \right)$$

- Partial derivative w.r.t. $\boldsymbol{\mu}_k$ is:

$$\nabla_{\mu_k} \mathscr{L} = \sum_{i=1}^{n} \frac{\pi_k \cdot \nabla_{\mu_k} \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)}{\sum \pi_j \mathcal{N}(\mathbf{x}_i \mid \mu_j, \Sigma_j)} = \sum_{i=1}^{n} r_{ik} (\mathbf{x}_i - \mu_k)^\top \Sigma_k^{-1} = \mathbf{0}$$

$$\Rightarrow \mu_k = \frac{\sum_i r_{ik} \mathbf{x}_i}{\sum_i r_{ik}}$$

Optimality condition: Variance

- Do the similar thing, an get

$$\Sigma_k = \frac{1}{n_k} \sum_{i=1}^{n} r_{ik}(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top$$

  - Here, we are using the shorthand $n_k = \sum_{i=1}^{n} r_{ik}$
  - For derivation, see section 11.2.3 of the main textbook

Optimality condition: Mixture weights

- Do the similar thing, and you get

$$\pi_k = \frac{n_k}{n}$$

  - For derivation, see section 11.2.3 of the main textbook
  - This one is trickier, as this is constrained; use Lagrange multipliers!

The full algorithm

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

$$r_{nk} = \frac{\pi_k \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)}{\sum_j \pi_j \mathcal{N}\big(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_j,\, \boldsymbol{\Sigma}_j\big)}\,. \tag{11.53}$$

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

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n\,, \tag{11.54}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top\,, \tag{11.55}$$

$$\pi_k = \frac{N_k}{N}\,. \tag{11.56}$$


Original Data
k-Means Clustering
EM Clustering
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Next lecture

- A bit more about the EM algorithm, in general
    - Why do we call such algorithms Expectation-Maximization?
    - Why does EM algorithm converge?
    - (Somewhat advanced; not in mid-term)

Cheers