9. Gaussian Mixture Models

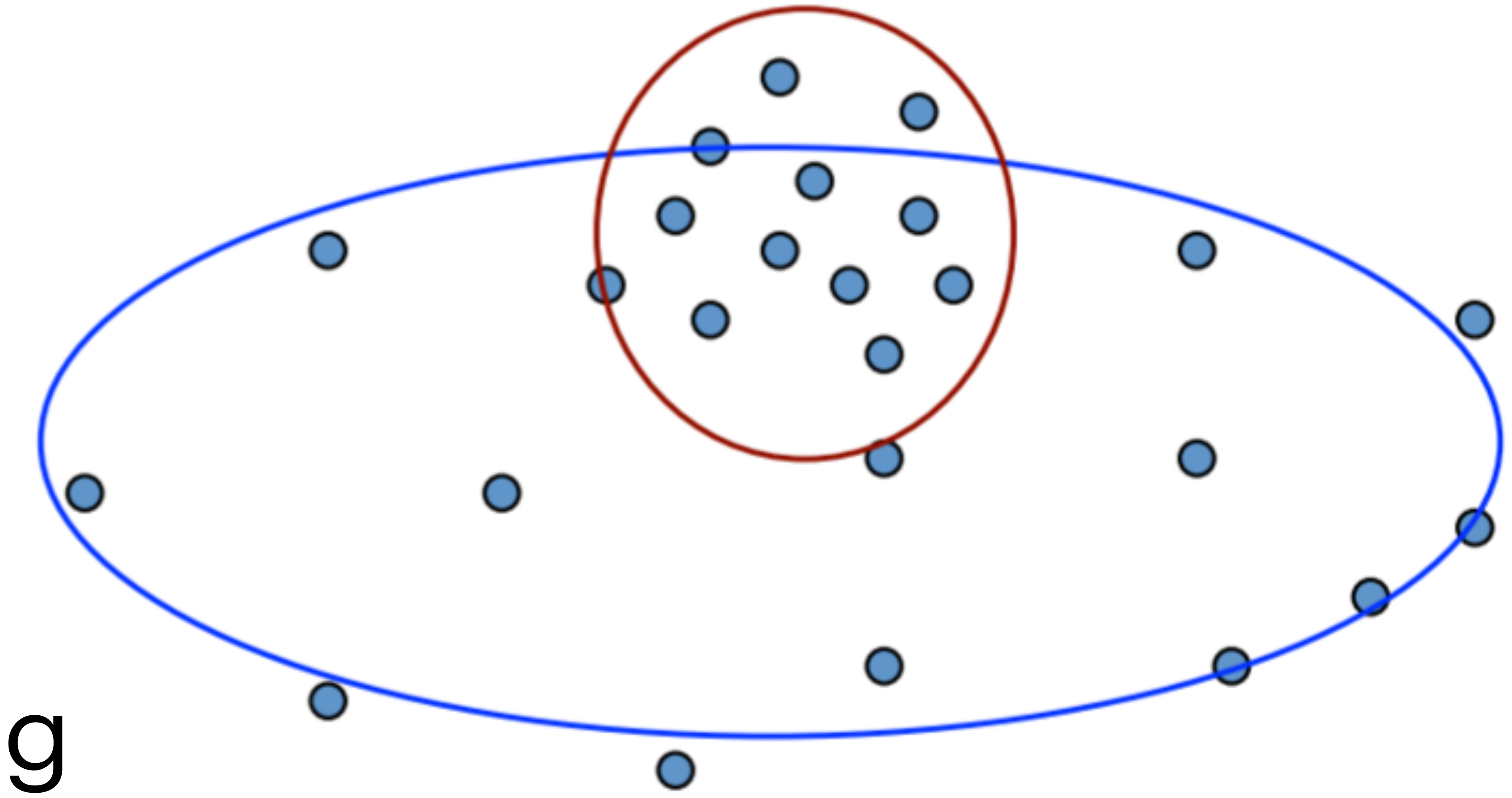
EEC454 Introduction to Machine Learning Systems

2023 Fall, Jaeho Lee

Recap: Clustering by K-means

- **K-means.** Each cluster is represented by the centroid.
  - A datum belongs to the cluster with nearest centroid.

- **Limitations.** Plenty, e.g., cannot handle…
  - overlapping clusters
  - “wider” clusters
  - *Example.* Non-local residents in Pohang
    - POSCO or POSTECH?

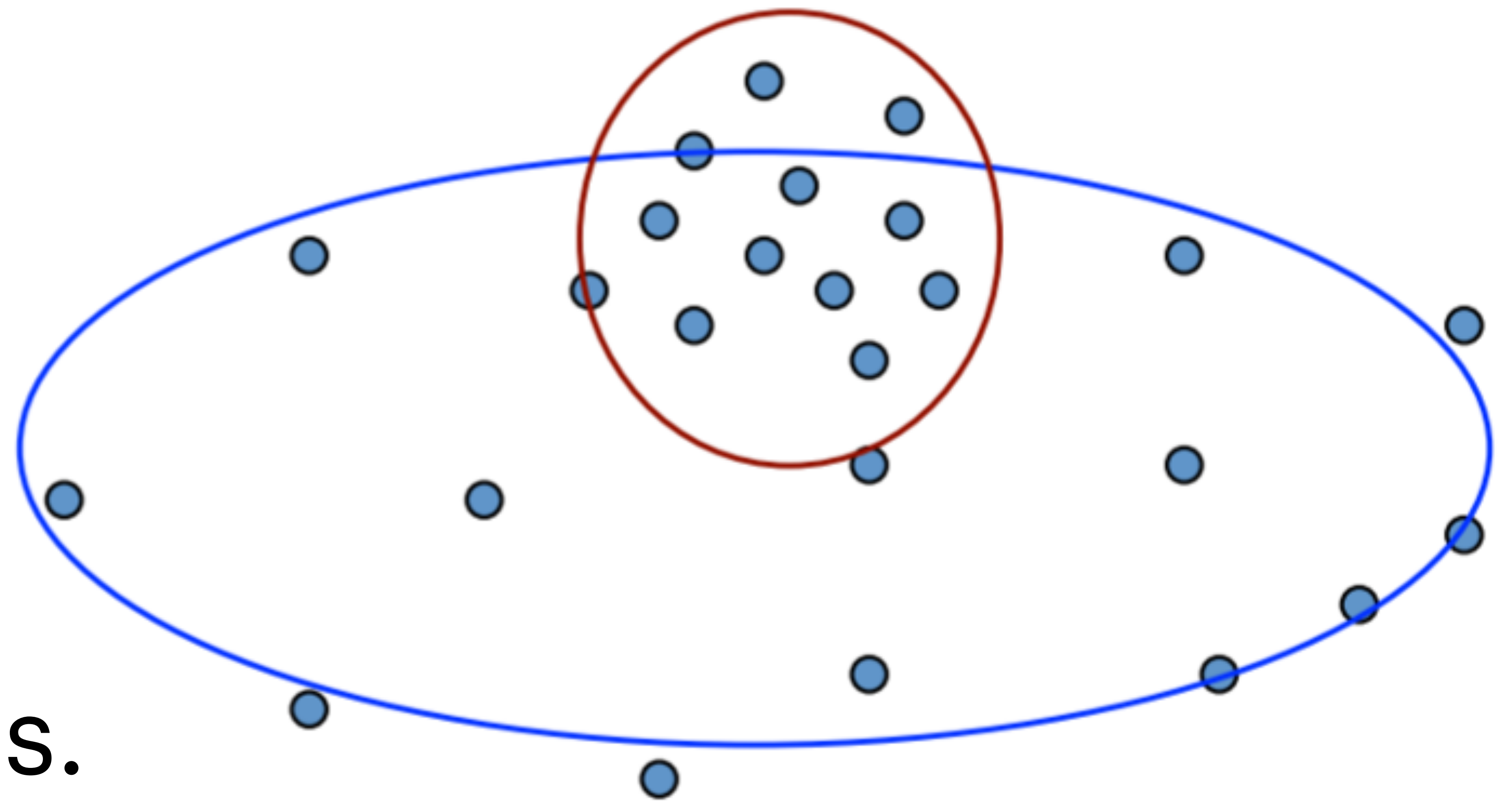


needs a probabilistic approach!

Mixture Models

Mixture models

- **Idea.** Take a generative approach, and fit parameters!
  - *Example*. the previous POSCO vs POSTECH.
    - We draw $Y \in \{0,1\} \sim \mathrm{Bern}(p)$. (0: POSCO, 1: POSTECH)
    - Model the conditional distribution:
      - If $Y = 0$, draw X from $\mathcal{N}(\mu_0, \sigma_0^2)$
      - If $Y = 1$, draw X from $\mathcal{N}(\mu_1, \sigma_1^2)$
  - Allows overlap & can account for wideness.



Mixture models

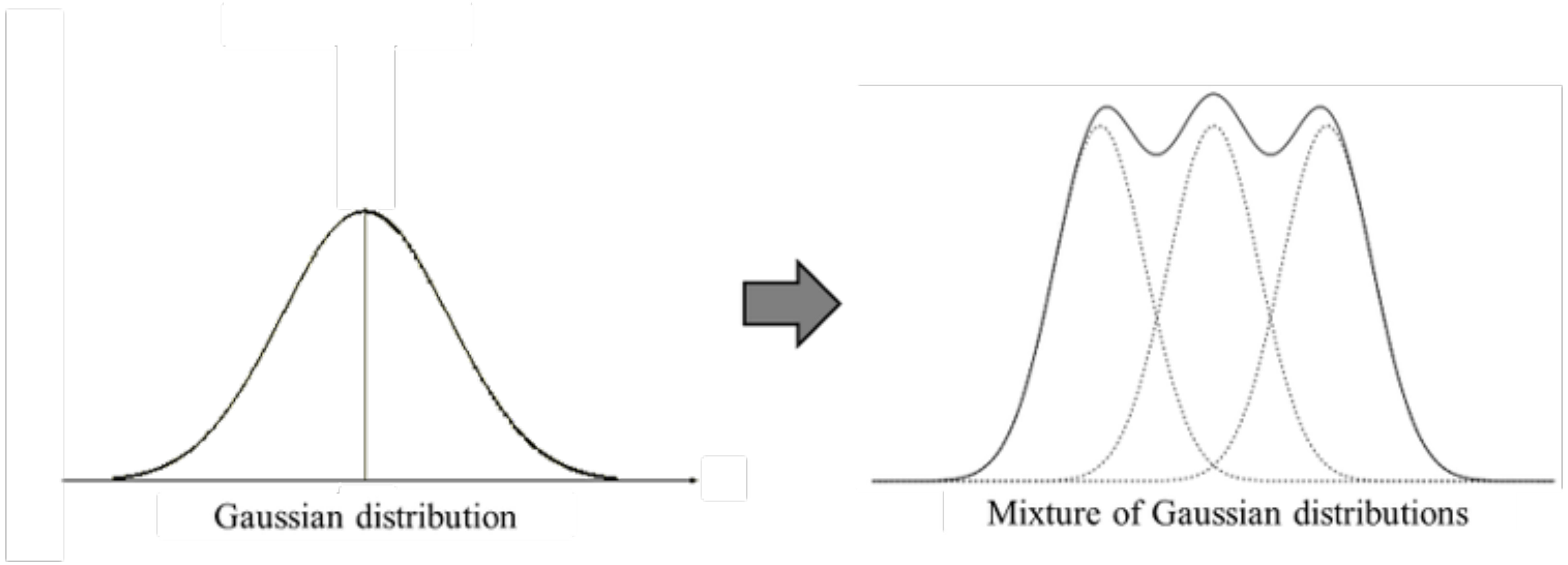
- **Perk.** If you have "learned" a nice probabilistic model from data, you can not only cluster, but also generate a new data.

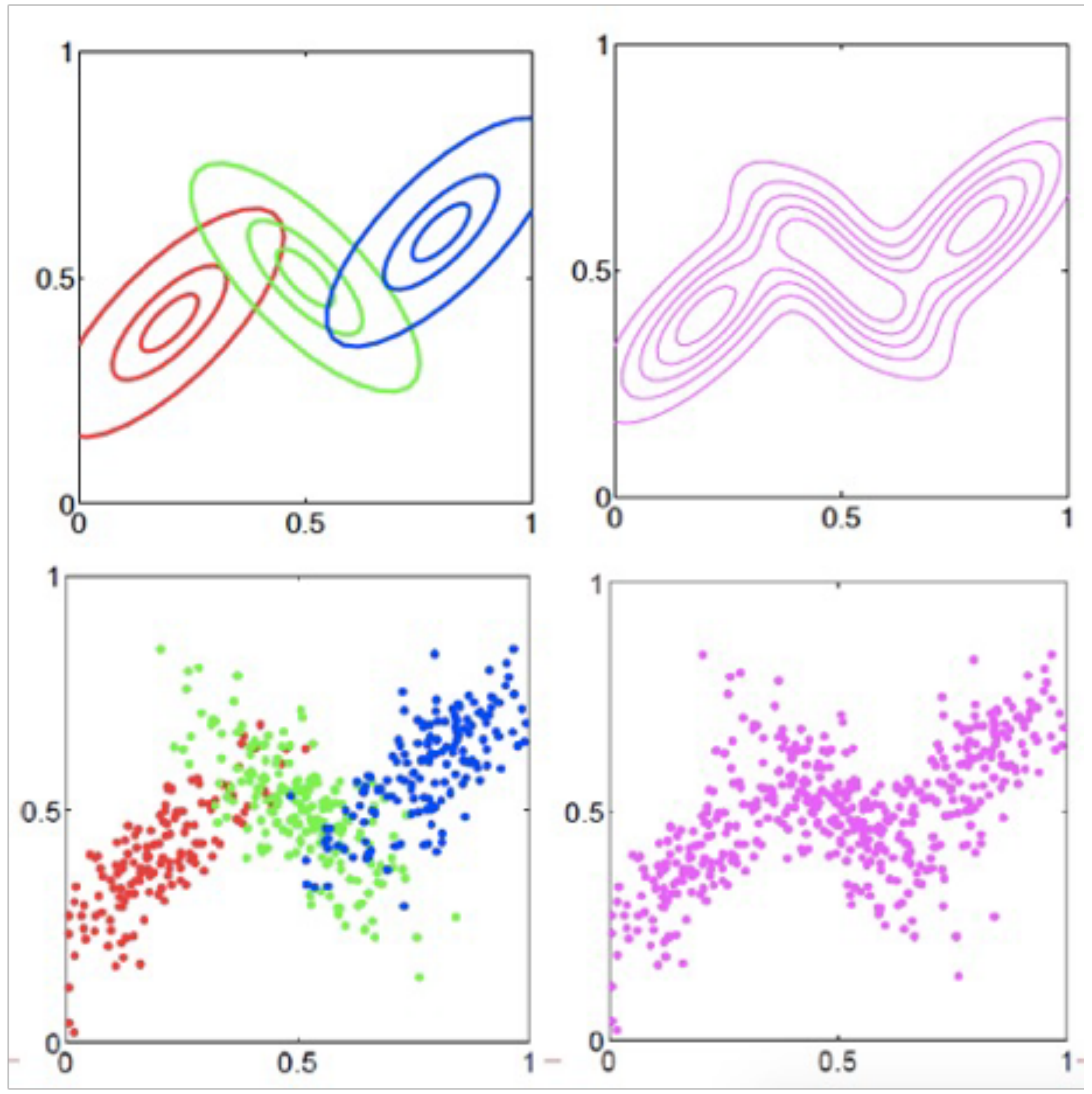
(Note: Example below requires additional text conditioning...)

(finite) Mixture models

- More generally we model the data-generating pdf with

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



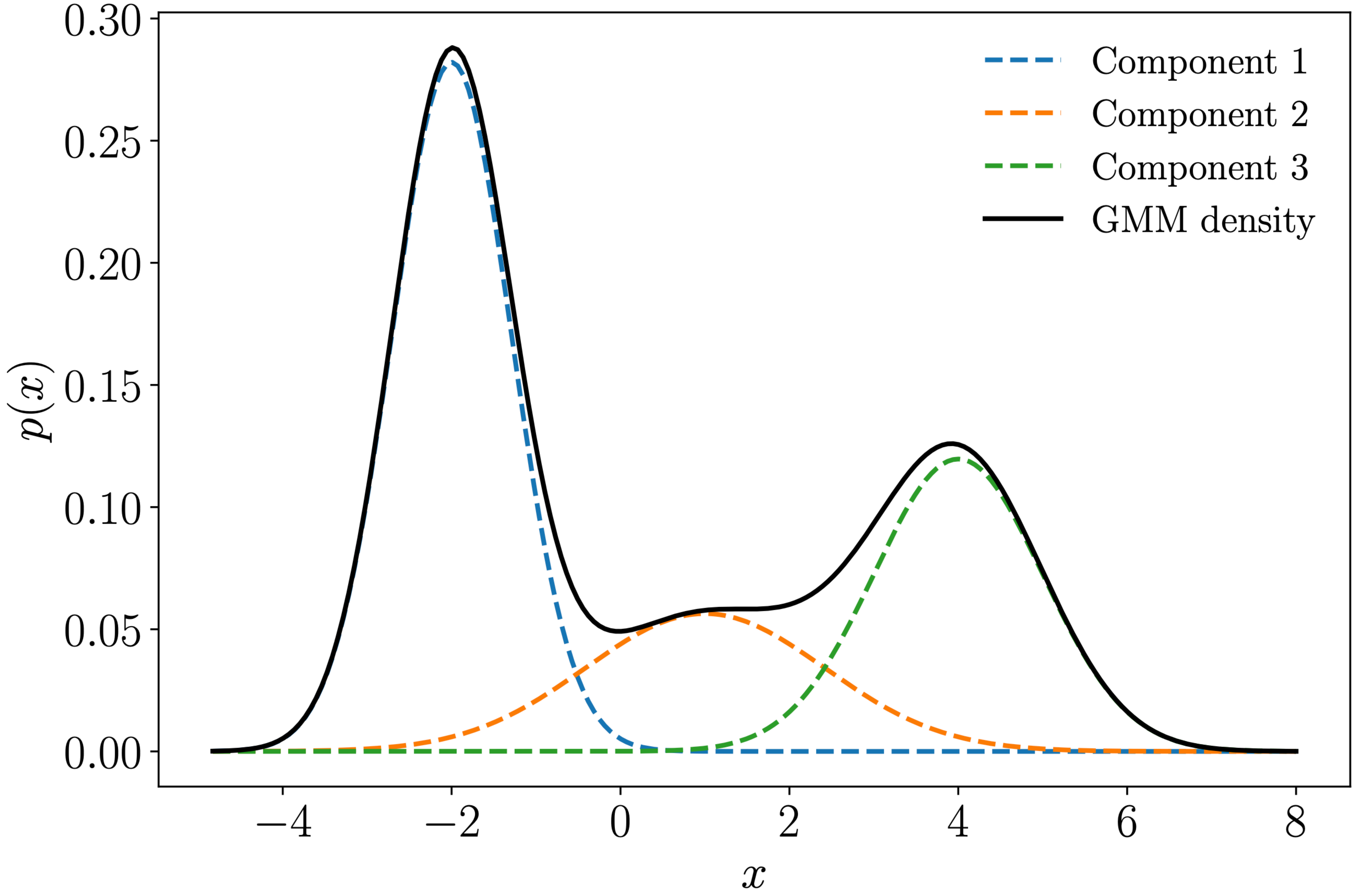


Gaussian mixture models

- Each base distribution is a Gaussian distribution:

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

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



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

Gaussian mixture models

- Each base distribution is a Gaussian distribution:

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

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

- **Question.** How do we fit the parameters, given $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$?
  - **Challenge.** We do not know the true labels!

Maximum Likelihood

- Similar to what we learned in naïve Bayes,
  what we want to try is the maximum likelihood.

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

$$= \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_i \mid \mu_k, \Sigma_k)$$

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

Maximum Log-Likelihood

- We do the usual log trick to make everything summation…

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

  - Normally, you would try to find the optimum by locating the critical point (i.e., gradient = 0)
    - Give it a try! (let me know if you succeed)

Expectation-Maximization

- **Idea.** Fix some variables and optimize others.
  Fix the optimized variables, and optimize the previously fixed.
  Repeat …
  - Generally, we call it expectation-maximization (EM) algorithm.
  - Similar to what we did in K-means!

Algorithm 1 k-means algorithm

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

Expectation-Maximization

- Recall that, in hard K-means…
  - Randomly initialize centroids $\{\mu_k\}$.
  - Fix the centroids $\{\mu_k\}$ and optimize the assignment $\{r_{ik}\}$.
    - Optimal, if **nearest neighbor**.
  - Fix the assignment $\{r_{ik}\}$ and optimize the centroid $\{\mu_k\}$.
    - Optimal, if **mean of the assigned data**.
  - Repeat.

Expectation-Maximization

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

    Non-binary, as in soft K-means
    - Optimal, if?
  - Fixed the *responsibility* $\{r_{ik}\}$ and optimized the parameters θ.
    - Optimal, if?
- Let's think about the optimal conditions...

Recall: Multivariate Gaussian

- Multivariate Gaussians:

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

- Take log, you get:

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

Recall: Responsibilities

- **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

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

Recall: Responsibilities

- **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

$$\underbrace{r_{ik}}_{p(y=k \mid \mathbf{x})} = \frac{\overbrace{\pi_k}^{p(y=k)} \overbrace{\mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)}^{p(\mathbf{x} \mid \mathbf{y}=k)}}{\underbrace{\sum_j \pi_j \mathcal{N}(\mathbf{x} \mid \mu_j, \Sigma_j)}_{p(\mathbf{x})}}$$

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

Recall: Responsibilities

- **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

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

Note. If $\pi_k = 1/K$, $\Sigma_k = \mathbf{I}/\beta$, 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. μ_k is...

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

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

Optimality Condition: Variance

- Do the similar thing, and you get

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

where we use the shorthand $n_k = \sum_{i=1}^{n} r_{ik}$.

see section 11.2.3 of the main textbook

Optimality Condition: Mixture Weights

- Do the similar thing, and you get

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

see section 11.2.4 of the main textbook;

this one is trickier as it's constrained—use Lagrange multipliers!

The full E-M

- Do the similar thing, and you get

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 $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$:

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

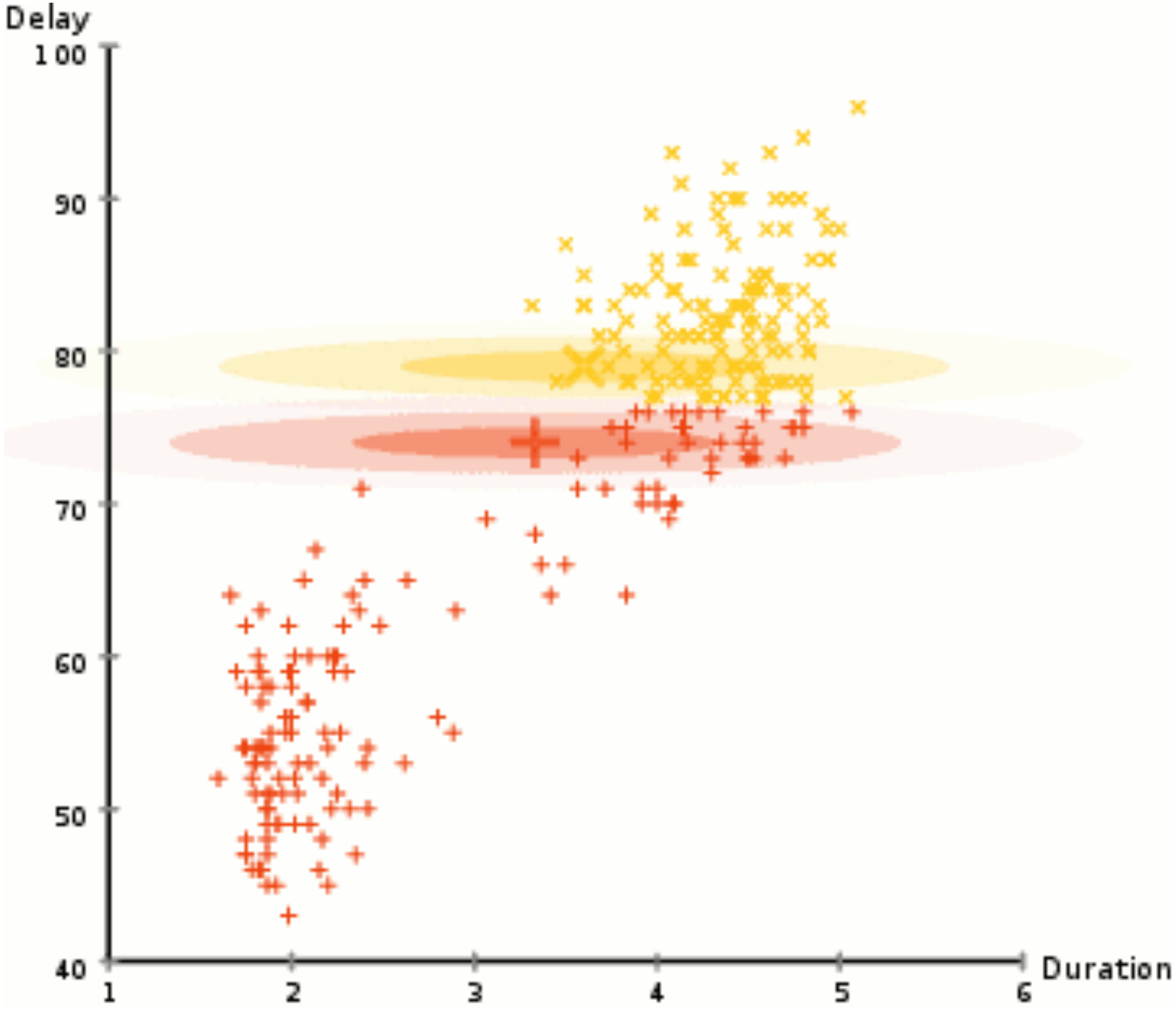
3. *M-step:* Reestimate parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ using the current responsibilities r_{nk} (from E-step):

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

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

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

The full E-M



The full E-M

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

Cheers

- *Next up.* Trees, Random Forest, and Boosting