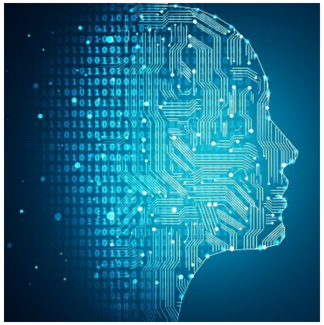
Machine Learning

GMMS and the EM Algorithm

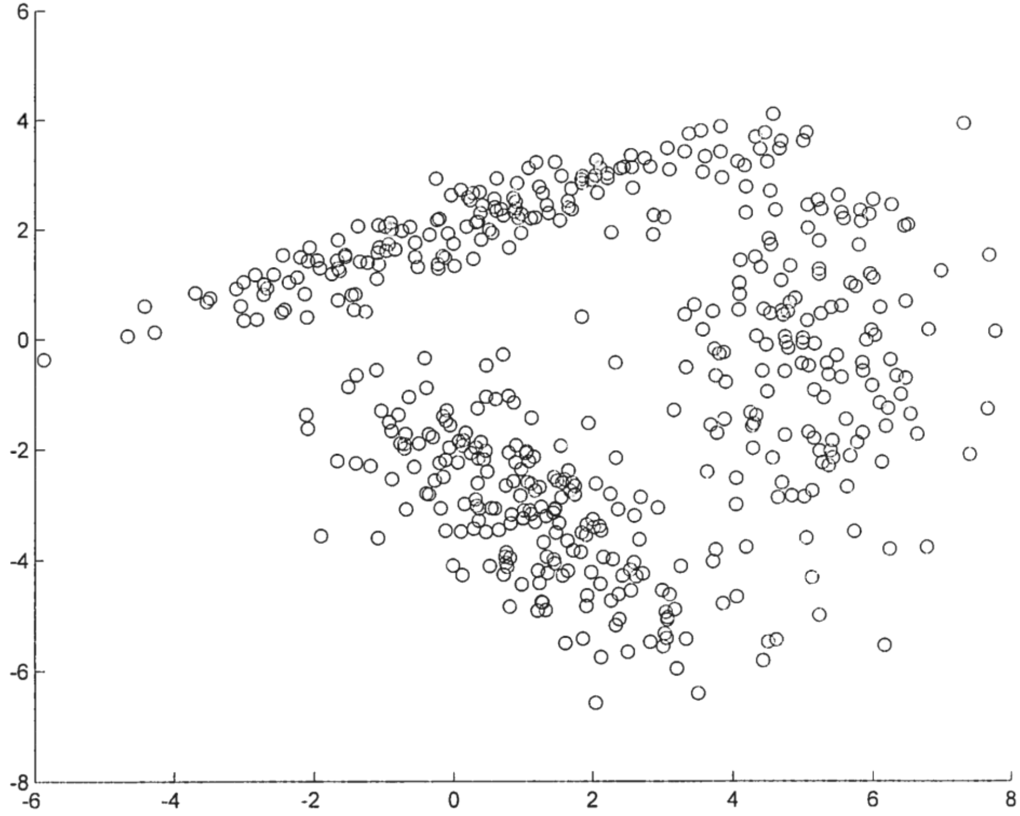
Kevin Moon (kevin.moon@usu.edu)

STAT/CS 5810/6655



Motivating Example

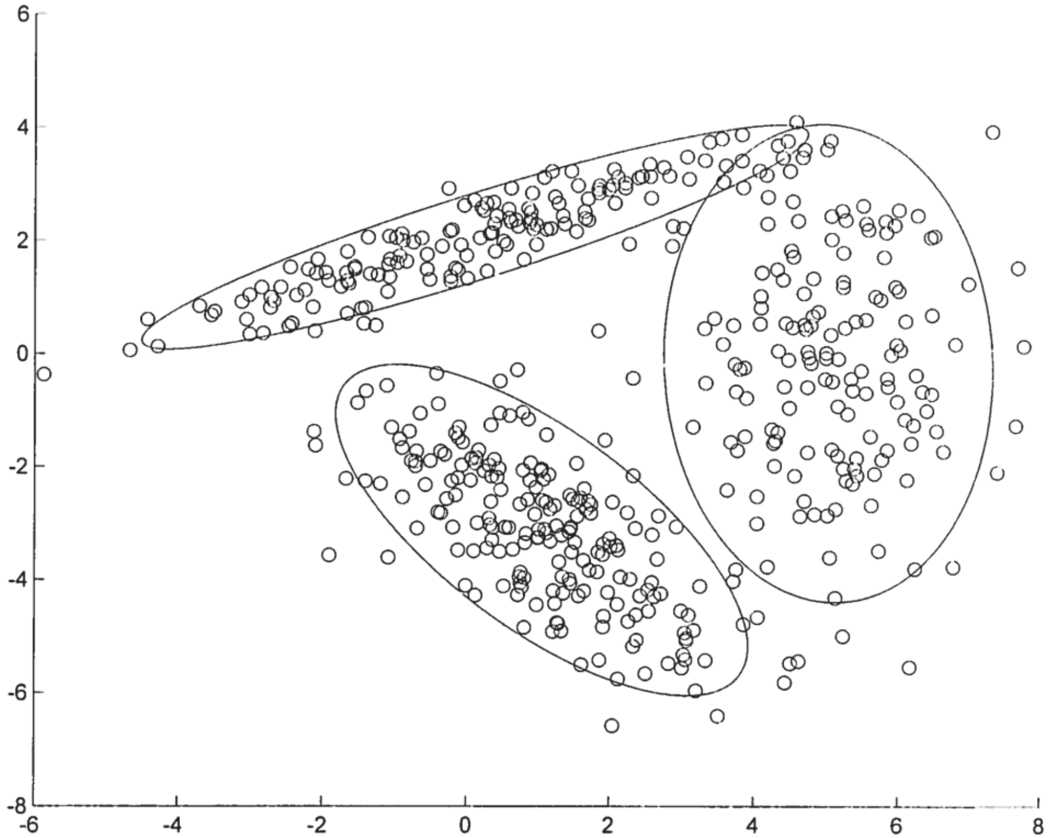
- Suppose we want to cluster the following dataset



- The data naturally cluster into 3 groups which are each described by bivariate Gaussian densities

Motivating Example

- Below ellipses represent 90% contours of a Gaussian mixture model (GMM) learned using MLE



- Today, we'll learn about GMMs and the EM algorithm, an iterative algorithm for MLE

Outline

1. Gaussian Mixture Models
2. MLE of GMMs
3. The EM algorithm for GMMs
4. The EM algorithm in general

Multivariate Gaussian RV

- Probability density:

$$\phi(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)$$

  - $\boldsymbol{x} \in \mathbb{R}^d$
  - $\boldsymbol{\mu} \in \mathbb{R}^d$
  - $\Sigma \in \mathbb{R}^{d\times d}, \Sigma \succ 0$ (PD)

- A random variable $\boldsymbol{X}$ follows a *Gaussian Mixture Model* (GMM) if its probability density function f has the form

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \Sigma_k)$$

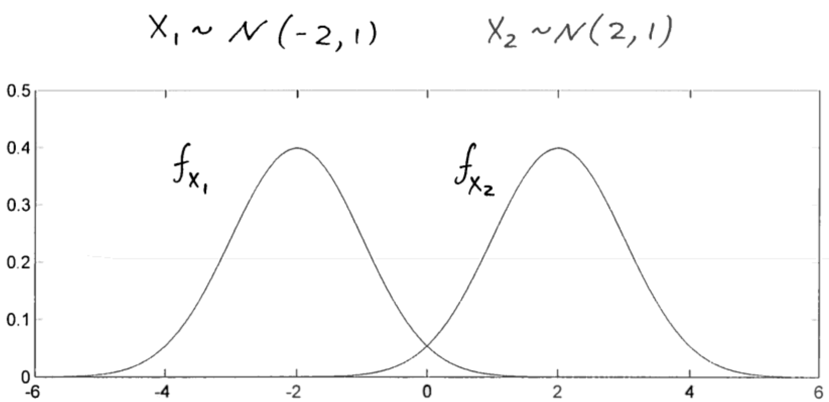
  - $w_k \geq 0, \sum_k w_k = 1$
  - $\boldsymbol{\mu}_k \in \mathbb{R}^d, \Sigma_k \in \mathbb{R}^{d\times d}, \Sigma_k \succ 0$

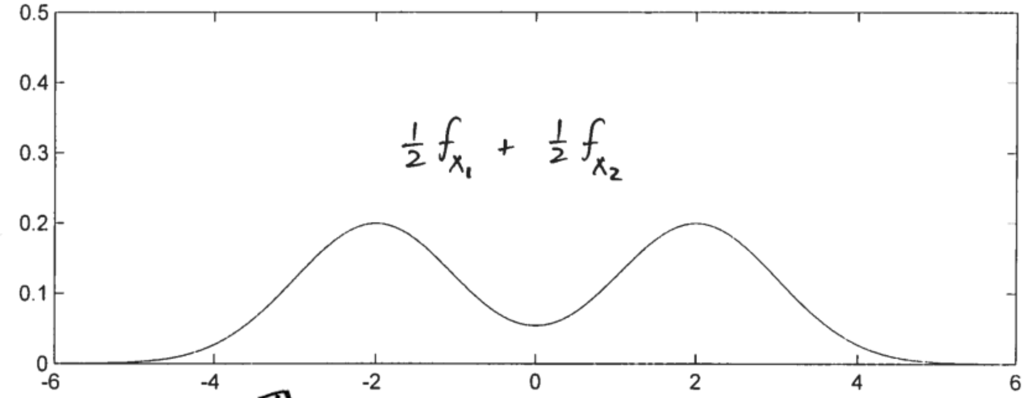
GMM vs. Sum of Gaussians

- It can be easy to confuse a *mixture* of Gaussians with a *sum* of Gaussians
- A sum of Gaussian RVs is another Gaussian
  - Unimodal
- A mixture of Gaussians can be multimodal and therefore not Gaussian

GMM vs Sum of Gaussians

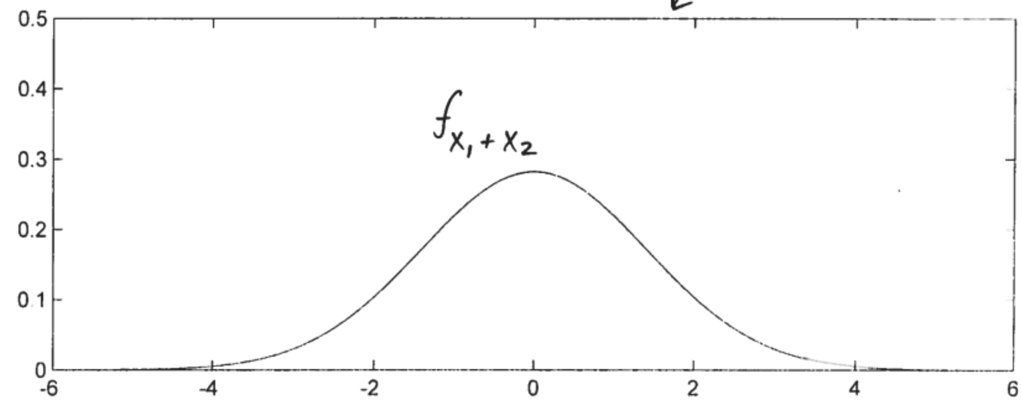
Example





mixture

not a mixture



$X_1 + X_2 \sim \mathcal{N}(0,2)$ if X_1 and X_2 are independent

Simulating a GMM

- Suppose the following are known:

$$\boldsymbol{\theta} = (w_1, \dots, w_K, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K, \Sigma_1, \dots, \Sigma_K)$$

- How can we simulate a realization of the GMM?
    1. Select a “component” k at random weighted according to the weights $w_1, \dots, w_K$
    2. Draw a realization $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$
- Why does this work?

Simulating a GMM

- Let $S \in \{1, \dots, K\}$ be a discrete RV such that
$$\Pr(S = k) = w_k$$
- The pdf $f(\boldsymbol{x})$ of $\boldsymbol{X}$ is such that for any event A,
$$\Pr(\boldsymbol{X} \in A) = \int_A f(\boldsymbol{x}) d\boldsymbol{x}$$
- By the law of total expectation:
$$\begin{aligned}
\Pr(\boldsymbol{X} \in A) &= \sum_{k=1}^{K} \Pr(\boldsymbol{X} \in A | S = k) \cdot \Pr(S = k) \\
&= \sum_{k=1}^{K} \left(\int_A \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \Sigma_k) d\boldsymbol{x} \right) w_k \\
&= \int_A \left(\sum_{k=1}^{K} w_k \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \Sigma_k) \right) d\boldsymbol{x}
\end{aligned}$$
- Hence $f(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \Sigma_k)$ as claimed before

Latent variables

- The variable S is an example of a *state* variable and is said to be *hidden* or *latent* because it is usually unobserved
- We will assume that every realization of a GMM is associated with a hidden state variable

MLE for GMMs

- From observations $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^d$, we want to infer the parameters $\boldsymbol{\theta} = (w_1, \dots, w_K, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K, \Sigma_1, \dots, \Sigma_K)$
  - I.e. to cluster the data
- We will use maximum likelihood estimation viewing K as fixed
- When $K = 1$, the MLE has a closed-form solution:
  - $\widehat{\boldsymbol{\mu}}_1 = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$
  - $\widehat{\Sigma}_1 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)^T$
- However, when $K > 1$, there is no closed-form solution

MLE for GMMs

- Denote $\underline{\boldsymbol{x}} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ for brevity
- Likelihood function:

$$\begin{aligned} L(\boldsymbol{\theta};\, \underline{\boldsymbol{x}}) &:= \prod_{i=1}^{n} f(\boldsymbol{x}_i; \theta) \\ &= \prod_{i=1}^{n} \left(\sum_{k=1}^{K} w_k \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \Sigma_k) \right) \end{aligned}$$

- Log-likelihood:

$$\ell(\boldsymbol{\theta};\, \underline{\boldsymbol{x}}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} w_k \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \Sigma_k) \right)$$

MLE for GMMs

- Since we can't find an analytical solution that maximizes ℓ wrt $\boldsymbol{\theta}$, we'll use an iterative strategy
- This strategy depends critically on the state variables associated with the observations:

$$\underline{\boldsymbol{s}} = (s_1, \ldots, s_n)$$

- A natural idea is an alternating algorithm like in k-means:
  - Given $\boldsymbol{\theta}$, update the estimate of $\underline{\boldsymbol{s}}$
  - Given $\underline{\boldsymbol{s}}$, update the estimate of $\boldsymbol{\theta}$
- Each step can be performed efficiently
- Learning these parameters can be thought of as a variant of k-means where the cluster assignments are *soft*

The Expectation Maximization (EM) Algorithm

- The *complete data* (observations with labels) is

$$\underline{\boldsymbol{z}} = (\underline{\boldsymbol{x}}, \underline{\boldsymbol{s}})$$

- Define the indicator variable

$$\Delta_{i,k} = \begin{cases} 1 & if\ s_i = k \\ 0 & if\ s_i \neq k \end{cases}$$

The EM Algorithm

- The *complete-data log-likelihood* is

$$
\begin{aligned}
\ell(\boldsymbol{\theta};\ \underline{\boldsymbol{x}}, \underline{\boldsymbol{s}}) &= \log\left(\prod_{i=1}^{n} \Pr(S_i = s_i; \theta) f(x_i | s_i; \theta)\right) \\
&= \log\left(\prod_{i=1}^{n} w_{s_i} \cdot \phi\left(\boldsymbol{x}_i; \boldsymbol{\mu}_{s_i}, \Sigma_{s_i}\right)\right) \\
&= \sum_{i=1}^{n} \log\left(w_{s_i} \cdot \phi\left(\boldsymbol{x}_i; \boldsymbol{\mu}_{s_i}, \Sigma_{s_i}\right)\right) \\
&= \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \Delta_{i,k} w_k \cdot \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \Sigma_k)\right) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} \Delta_{i,k} [\log w_k + \log \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \Sigma_k)]
\end{aligned}
$$

Only term that depends on $\underline{\boldsymbol{s}}$

The EM Algorithm

- Ideally, if we knew $S_i/\Delta_{i,k}$, we could maximize the complete data log-likelihood
- Since we don't know these state variables, we can replace $\Delta_{i,k}$ with its expected value and then maximize wrt $\boldsymbol{\theta}$
- However, the expected value of $\Delta_{i,k}$ depends on $\boldsymbol{\theta}$
  - A "chicken and egg" problem
  - Suggests an iterative approach
- The EM algorithm is such an approach
  - Produces a sequence of estimates $\boldsymbol{\theta}^{(1)}, \ \boldsymbol{\theta}^{(2)}, \ldots$
  - Alternates between two steps: the E step and the M step

The E-step

- Calculate the expected complete-data log-likelihood:

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}\right) = \mathbb{E}_{\underline{\boldsymbol{S}}|\underline{\boldsymbol{x}}}\left[\ell\left(\boldsymbol{\theta};\ \underline{\boldsymbol{x}}, \underline{\boldsymbol{s}}\right) | \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}\right]$$

Conditional expectation wrt $\underline{\boldsymbol{S}}|\underline{\boldsymbol{x}}$. S is capitalized here since it is viewed as random.

The pmf of $\underline{\boldsymbol{S}}|\underline{\boldsymbol{x}}$ depends on the GMM parameters; use the current estimate

- This becomes

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{i,k}^{(j)} \left[\log w_k + \log \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \Sigma_k)\right]$$

Let’s find an expression for this

The E-Step

$$
\begin{aligned}
\gamma_{i,k}^{(j)} &= \mathbb{E}\left[\Delta_{i,k} | \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}\right] \\
&= \Pr\left(\Delta_{i,k} = 1 | \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}\right) \\
&= \Pr\left(S_i = k | \underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}\right) \\
&= \frac{\Pr\left(S_i = k; \boldsymbol{\theta}^{(j)}\right) f\left(\boldsymbol{x}_i \middle| S_i = k; \boldsymbol{\theta}^{(j)}\right)}{f\left(\boldsymbol{x}_i \middle| \boldsymbol{\theta}^{(j)}\right)} \\
&= \frac{w_k^{(j)} \cdot \phi\left(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(j)}, \Sigma_k^{(j)}\right)}{\sum_{\ell=1}^{K} w_\ell^{(j)} \cdot \phi\left(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell^{(j)}, \Sigma_\ell^{(j)}\right)}
\end{aligned}
$$

- This is the fraction of the density value at $\boldsymbol{x}_i$ explained by the kth component
  - Sometimes called the *responsibility* of cluster k for $\boldsymbol{x}_i$
  - A soft measure of cluster membership

The M-Step

- Compute

$$\boldsymbol{\theta}^{(j+1)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}\right)$$

- Recall

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{i,k}^{(j)} \left[\log w_k - \frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x_i - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}(x_i - \boldsymbol{\mu}_k)\right]$$

The M-Step

- It can be shown without too much trouble that the solution is

$$\boldsymbol{\mu}_k^{(j+1)} = \frac{\sum_{i=1}^n \gamma_{i,k}^{(j)} \boldsymbol{x}_i}{\sum_{i=1}^n \gamma_{i,k}^{(j)}}$$

Weighted sample mean and covariance

$$\Sigma_k^{(j+1)} = \frac{\sum_{i=1}^n \gamma_{i,k}^{(j)} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(j+1)}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(j+1)}\right)^T}{\sum_{i=1}^n \gamma_{i,k}^{(j)}}$$

$$w_k^{(j+1)} = \frac{1}{n}\sum_{i=1}^n \gamma_{i,k}^{(j)}$$

Fraction of all data explained by kth component

Terminating the EM Algorithm

- The algorithm can be terminated when the increase in the likelihood is small:

$$\ell\left(\boldsymbol{\theta}^{(j+1)}; \underline{\boldsymbol{x}}\right) - \ell\left(\boldsymbol{\theta}^{(j)}; \underline{\boldsymbol{x}}\right) \leq \epsilon$$

  - ϵ is a small number

- We'll see that the likelihood function is increasing with j

Initialization

- The EM algorithm is sensitive to initialization (like k-means)
  - I.e. it is not convex
- One possibility:
  - $\boldsymbol{\mu}_k^{(0)}$ = random $\boldsymbol{x}_i$ (distinct for each k)
  - $\Sigma_k^{(0)}$ = sample covariance of all data (same for all k)
  - $w_k^{(0)} = \frac{1}{K}$
  - As with k-means, it may be beneficial to run the algorithm many times and pick the result with the best likelihood
- Another possibility: initialize with k-means result

Connection to k-means

- k-means is a special case of the EM algorithm with GMM
- Consider the GMM $f(\boldsymbol{x}) = \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \sigma^2 I)$
  - σ^2 is fixed

Common, isotropic covariance

- The EM algorithm iterates over:

$$\boldsymbol{\mu}_k^{(j+1)} = \frac{\sum_{i=1}^{n} \gamma_{i,k}^{(j)} \boldsymbol{x}_i}{\sum_{i=1}^{n} \gamma_{i,k}^{(j)}}$$

$$w_k^{(j+1)} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{i,k}^{(j)}$$

$$\gamma_{i,k}^{(j+1)} = \frac{w_k^{(j)} \cdot \phi\left(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(j)}, \sigma^2 I\right)}{\sum_{\ell=1}^{K} w_\ell^{(j)} \cdot \phi\left(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell^{(j)}, \sigma^2 I\right)}$$

Connection to k-means

- Now as $\sigma^2 \to 0$, we have

$$\gamma_{i,k} \to \begin{cases} 1 & \text{if } k = \arg\min_{\ell} ||\boldsymbol{x}_i - \boldsymbol{\mu}_\ell|| \\ 0 & \text{otherwise} \end{cases}$$

- This is equivalent to the k-means algorithm

The EM Algorithm in General

- The EM algorithm applies to many other MLE problems where unobserved variables would make computation easier
- As before, denote
  - $\underline{\boldsymbol{x}} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$
  - $\underline{\boldsymbol{s}} = (s_1, \dots, s_n)$
  - s_i is some variable that explains how $\boldsymbol{x}_i$ was generated
- Let $\ell(\boldsymbol{\theta}; \underline{\boldsymbol{x}})$ denote the log-likelihood and $\ell(\boldsymbol{\theta}; \underline{\boldsymbol{x}}, \underline{\boldsymbol{s}})$ the complete-data log-likelihood

The EM Algorithm in General

The general EM algorithm:

- Initialize $\boldsymbol{\theta}^{(0)}$
- Repeat

  E-Step: Form
  $$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}\right) = \mathbb{E}_{\underline{\boldsymbol{S}}|\underline{\boldsymbol{x}}}\left[\ell\left(\boldsymbol{\theta};\ \underline{\boldsymbol{x}}, \underline{\boldsymbol{s}}\right)|\underline{\boldsymbol{x}}; \boldsymbol{\theta}^{(j)}\right]$$

  M-Step: Compute
  $$\boldsymbol{\theta}^{(j+1)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(j)}\right)$$

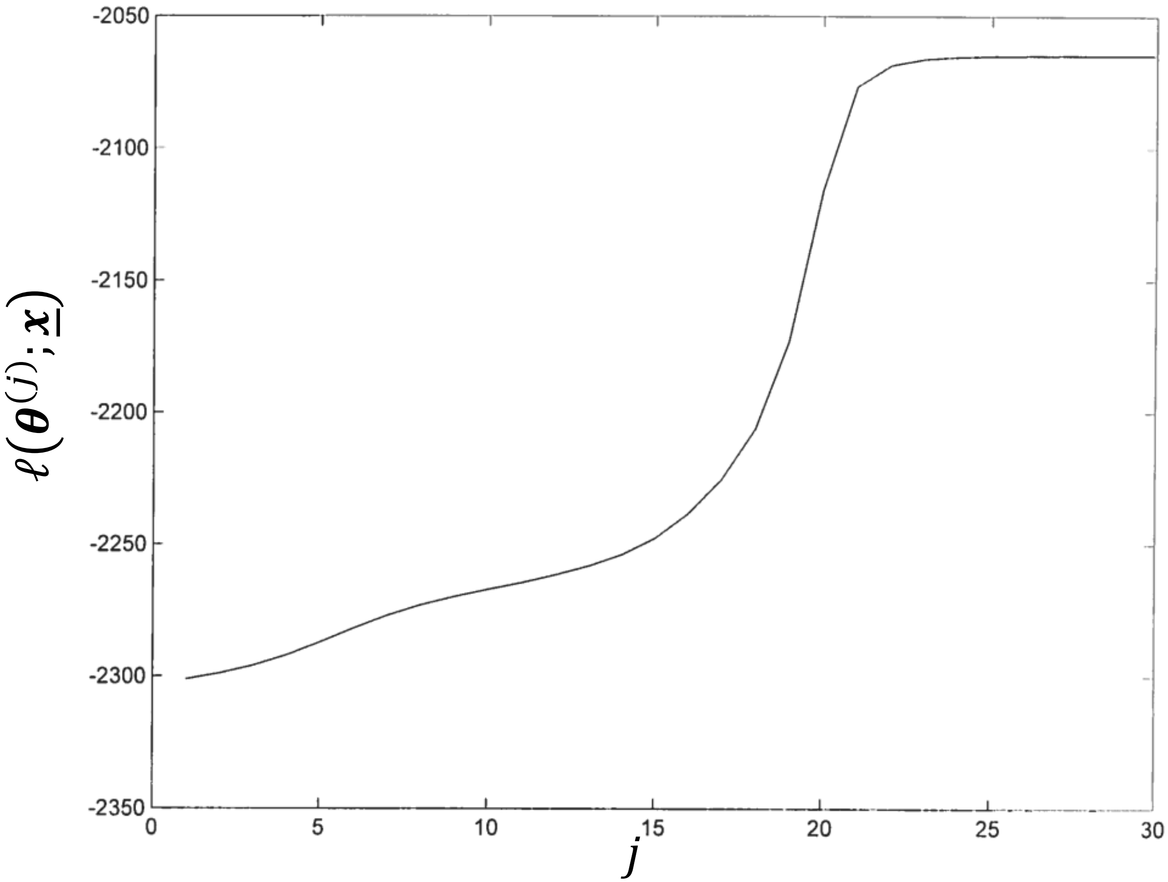
- Until termination criterion satisfied

Ascent property of the EM Algorithm

- **Theorem:** For each $j = 0,1,2,\ldots$

$$\ell\left(\boldsymbol{\theta}^{(j+1)}; \underline{\boldsymbol{x}}\right) \geq \ell\left(\boldsymbol{\theta}^{(j)}; \underline{\boldsymbol{x}}\right)$$

  - 3 Gaussians example from beginning:



Final Thoughts/Summary

- The EM algorithm is a general algorithm for MLE whenever there may be unobserved (latent) or missing variables
- Initialization matters (generally not a convex problem)
- Convergence can sometimes be slow

Further reading

- A tutorial: https://ieeexplore.ieee.org/abstract/document/543975
- Proof of the final theorem: http://web.eecs.umich.edu/~cscott/past_courses/eecs545f16/20_em_gmm.pdf
- ESL Section 8.5