EM Algorithm

David A. Levin

DA-IICT

January 2020

A problem

- Suppose that X has a mixture of Gaussian distributions: Let $f_j(x; \mu_j, \Sigma_j)$ for $j = 1, \ldots, K$ be K Normal densities. The j-th density has mean μ_j and covariance Σ_j. (These may be d-dimensional Gaussians.)

$$f_X(x) = \sum_{j=1}^{k} f_i(x; \mu_j, \Sigma_j)\pi_j.$$

- A random vector with the density above can be obtained by first sampling a **latent**, or hidden, variable $Z \in \{1, \ldots, K\}$ according to a distribution π on $\{1, \ldots, K\}$, so that $\mathbb{P}(Z = j) = \pi_j$.
- Given that $Z = j$, the vector $\boldsymbol{X}$ is generated according to the Normal density $f_j(\boldsymbol{x}; \mu_j, \Sigma_j)$
- Estimation of $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$ and $\boldsymbol{\Sigma} = (\Sigma_1, \ldots, \Sigma_K)$ is no longer tractable.

A problem

- Suppose that X has a mixture of Gaussian distributions: Let $f_j(x; \mu_j, \Sigma_j)$ for $j = 1, \ldots, K$ be K Normal densities. The j-th density has mean μ_j and covariance Σ_j. (These may be d-dimensional Gaussians.)

$$f_X(x) = \sum_{j=1}^{k} f_i(x; \mu_j, \Sigma_j)\pi_j .$$

- A random vector with the density above can be obtained by first sampling a **latent**, or hidden, variable $Z \in \{1, \ldots, K\}$ according to a distribution π on $\{1, \ldots, K\}$, so that $\mathbb{P}(Z = j) = \pi_j$.
- Given that $Z = j$, the vector $\boldsymbol{X}$ is generated according to the Normal density $f_j(\boldsymbol{x}; \mu_j, \Sigma_j)$
- Estimation of $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$ and $\boldsymbol{\Sigma} = (\Sigma_1, \ldots, \Sigma_K)$ is no longer tractable.

A problem

- Suppose that X has a mixture of Gaussian distributions: Let $f_j(x;\mu_j,\Sigma_j)$ for $j=1,\dots,K$ be K Normal densities. The j-th density has mean μ_j and covariance Σ_j. (These may be d-dimensional Gaussians.)

$$f_X(x) = \sum_{j=1}^{k} f_i(x;\mu_j,\Sigma_j)\pi_j\,.$$

- A random vector with the density above can be obtained by first sampling a **latent**, or hidden, variable $Z \in \{1,\dots,K\}$ according to a distribution π on $\{1,\dots,K\}$, so that $\mathbb{P}(Z=j)=\pi_j$.
- Given that $Z=j$, the vector $\boldsymbol{X}$ is generated according to the Normal density $f_j(\boldsymbol{x};\mu_j,\Sigma_j)$
- Estimation of $\boldsymbol{\mu}=(\mu_1,\dots,\mu_K)$ and $\boldsymbol{\Sigma}=(\Sigma_1,\dots,\Sigma_K)$ is no longer tractable.

A problem

- Suppose that X has a mixture of Gaussian distributions: Let $f_j(x; \mu_j, \Sigma_j)$ for $j = 1, \dots, K$ be K Normal densities. The j-th density has mean μ_j and covariance Σ_j. (These may be d-dimensional Gaussians.)

$$f_X(x) = \sum_{j=1}^{k} f_i(x; \mu_j, \Sigma_j)\pi_j.$$

- A random vector with the density above can be obtained by first sampling a **latent**, or hidden, variable $Z \in \{1, \dots, K\}$ according to a distribution π on $\{1, \dots, K\}$, so that $\mathbb{P}(Z = j) = \pi_j$.
- Given that $Z = j$, the vector $\boldsymbol{X}$ is generated according to the Normal density $f_j(\boldsymbol{x}; \mu_j, \Sigma_j)$
- Estimation of $\boldsymbol{\mu} = (\mu_1, \dots, \mu_K)$ and $\boldsymbol{\Sigma} = (\Sigma_1, \dots, \Sigma_K)$ is no longer tractable.

- The log-likelihood function is

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{K} f_j(x_i; \mu_j, \Sigma_j) \right)$$

- If we could observe $(Z_i)_{i=1}^n$, then we could separate out each "class" (the data points X_i with $Z_i = j$), and separately estimate μ_j, Σ_j via the usual MLE procedure.

- Let us see what happens when we look for critical points.
- $$\frac{d\ell}{d\mu_j} = \sum_{i=1}^{n} \frac{\pi_j f_j(x_i;\mu_j,\Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_i;\mu_\ell,\Sigma_\ell)} \Sigma_j^{-1}(x_i - \mu_j)$$
- Define
$$\gamma^{(i)}(j) = \mathbb{P}(Z_i = j \mid x_i; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$N_j = \sum_{i=1}^{n} \gamma^{(i)}(j) = \mathbb{E}[\#X_i\text{'s in class } j \mid x_i] .$$
- The $\gamma_j^{(i)}$ is the **responsibility** of class j for the i-th data point.
- The solution to $\frac{d\ell}{d\mu_j} = 0$ satisfies
$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{n} \gamma^{(i)}(j) x_i .$$

- Let us see what happens when we look for critical points.
- $$\frac{d\ell}{d\mu_j} = \sum_{i=1}^{n} \frac{\pi_j f_j(x_i;\mu_j,\Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_i;\mu_\ell,\Sigma_\ell)} \Sigma_j^{-1}(x_i - \mu_j)$$
- Define
$$\gamma^{(i)}(j) = \mathbb{P}(Z_i = j \mid x_i; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$N_j = \sum_{i=1}^{n} \gamma^{(i)}(j) = \mathbb{E}[\#X_i\text{'s in class } j \mid x_i]\,.$$
- The $\gamma_j^{(i)}$ is the **responsibility** of class j for the i-th data point.
- The solution to $\frac{d\ell}{d\mu_j} = 0$ satisfies
$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{n} \gamma^{(i)}(j) x_i\,.$$

- Let us see what happens when we look for critical points.
- $$\frac{d\ell}{d\mu_j} = \sum_{i=1}^{n} \frac{\pi_j f_j(x_i;\mu_j,\Sigma_j)}{\sum_{\ell=1}^{K}\pi_\ell f_\ell(x_i;\mu_\ell,\Sigma_\ell)} \Sigma_j^{-1}(x_i-\mu_j)$$
- Define

$$\gamma^{(i)}(j) = \mathbb{P}(Z_i = j \mid x_i; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$N_j = \sum_{i=1}^{n} \gamma^{(i)}(j) = \mathbb{E}[\#X_i\text{'s in class } j \mid x_i]\,.$$

- The $\gamma_j^{(i)}$ is the **responsibility** of class j for the i-th data point.
- The solution to $\frac{d\ell}{d\mu_j} = 0$ satisfies

$$\mu_j = \frac{1}{N_j}\sum_{i=1}^{n} \gamma^{(i)}(j)\, x_i\,.$$

- Let us see what happens when we look for critical points.
- $$\frac{d\ell}{d\mu_j} = \sum_{i=1}^{n} \frac{\pi_j f_j(x_i;\mu_j,\Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_i;\mu_\ell,\Sigma_\ell)} \Sigma_j^{-1}(x_i-\mu_j)$$
- Define

$$\gamma^{(i)}(j) = \mathbb{P}(Z_i = j \mid x_i; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$N_j = \sum_{i=1}^{n} \gamma^{(i)}(j) = \mathbb{E}[\#X_i\text{'s in class } j \mid x_i]\,.$$

- The $\gamma_j^{(i)}$ is the **responsibility** of class j for the i-th data point.
- The solution to $\frac{d\ell}{d\mu_j} = 0$ satisfies

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{n} \gamma^{(i)}(j) x_i\,.$$

- Let us see what happens when we look for critical points.
- $$\frac{d\ell}{d\mu_j} = \sum_{i=1}^{n} \frac{\pi_j f_j(x_i; \mu_j, \Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_i; \mu_\ell, \Sigma_\ell)} \Sigma_j^{-1}(x_i - \mu_j)$$
- Define

$$\gamma^{(i)}(j) = \mathbb{P}(Z_i = j \mid x_i; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$N_j = \sum_{i=1}^{n} \gamma^{(i)}(j) = \mathbb{E}[\#X_i\text{'s in class } j \mid x_i] \,.$$

- The $\gamma_j^{(i)}$ is the **responsibility** of class j for the i-th data point.
- The solution to $\frac{d\ell}{d\mu_j} = 0$ satisfies

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{n} \gamma^{(i)}(j) x_i \,.$$

- We can differentiate with respect to the (components of) Σ_j and solve for critical points. Doing so yields

$$\Sigma_j = \frac{1}{N_j} \sum_{i=1}^{n} \gamma^{(i)}(j)(x_i - \mu_j)(x_i - \mu_j)^T.$$

- Note similarity with a single Gaussian, but the sample covariances $(x_i - \mu_j)(x_i - \mu_j)^T$ are weighted by the $\gamma^{(i)}(j)$'s. These are called **responsibilities**.
- Exercise: Work out the details!

- We can differentiate with respect to the (components of) Σ_j and solve for critical points. Doing so yields

$$\Sigma_j = \frac{1}{N_j} \sum_{i=1}^{n} \gamma^{(i)}(j)(x_i - \mu_j)(x_i - \mu_j)^T .$$

- Note similarity with a single Gaussian, but the sample covariances $(x_i - \mu_j)(x_i - \mu_j)^T$ are weighted by the $\gamma^{(i)}(j)$'s. These are called **responsibilities**.
- Exercise: Work out the details!

- We can differentiate with respect to the (components of) Σ_j and solve for critical points. Doing so yields

$$\Sigma_j = \frac{1}{N_j} \sum_{i=1}^{n} \gamma^{(i)}(j)(x_i - \mu_j)(x_i - \mu_j)^T.$$

- Note similarity with a single Gaussian, but the sample covariances $(x_i - \mu_j)(x_i - \mu_j)^T$ are weighted by the $\gamma^{(i)}(j)$'s. These are called **responsibilities**.
- Exercise: Work out the details!

- Differentiating with respect to π_j gives

$$\frac{d\ell}{d\pi_j} = \sum_{i=1}^{n} \frac{f_j(x_i;\mu_j,\Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_i;\mu_\ell,\Sigma_\ell)}$$

- But the π's satisfy the constraint

$$g(\boldsymbol{\pi}) \stackrel{\text{def}}{=} \sum_j \pi_j = 1\,.$$

so we need to use Lagrange multiplier to solve for constrained maxima:

$$\begin{aligned} 0 &= \frac{d\ell}{d\pi_k} - \lambda \frac{dg}{d\pi_k} \\ &= \sum_{i=1}^{n} \frac{f_j(x_i;\mu_j,\Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_i;\mu_\ell,\Sigma_\ell)} - \lambda \\ &= \sum_{i=1}^{n} \gamma^{(i)}(j) - \lambda\pi_j && \text{multiplying by } \pi_j \\ &= N_j - \lambda\pi_j \end{aligned}$$

- Differentiating with respect to π_j gives

$$\frac{d\ell}{d\pi_j} = \sum_{i=1}^{n} \frac{f_j(x_i; \mu_j, \Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_i; \mu_\ell, \Sigma_\ell)}$$

- But the π's satisfy the constraint

$$g(\boldsymbol{\pi}) \stackrel{\text{def}}{=} \sum_j \pi_j = 1 \,.$$

so we need to use Lagrange multiplier to solve for constrained maxima:

$$\begin{aligned}
0 &= \frac{d\ell}{d\pi_k} - \lambda \frac{dg}{d\pi_k} \\
&= \sum_{i=1}^{n} \frac{f_j(x_i; \mu_j, \Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_i; \mu_\ell, \Sigma_\ell)} - \lambda \\
&= \sum_{i=1}^{n} \gamma^{(i)}(j) - \lambda \pi_j && \text{multiplying by } \pi_j \\
&= N_j - \lambda \pi_j
\end{aligned}$$

- Note that if $C_j = |\{i : Z_i = j\}|$, the number of X_i's generated from class j, then

$$\sum_{j=1}^{K} N_j = \sum_j \mathbb{E}\left[C_j \mid \boldsymbol{x}\right] = \mathbb{E}\left[\sum_j C_j \middle| \boldsymbol{x}\right] = n.$$

- Summing over j yields

$$0 = \sum_{j=1}^{K} N_j - \lambda \sum_j \pi_j = n - \lambda .$$

- Thus $\lambda = n$ and substituting back we find

$$\pi_j = \frac{N_j}{n} .$$

- Note that if $C_j = |\{i : Z_i = j\}|$, the number of X_i's generated from class j, then

$$\sum_{j=1}^{K} N_j = \sum_j \mathbb{E}\left[C_j \mid \boldsymbol{x}\right] = \mathbb{E}\left[\sum_j C_j \middle| \boldsymbol{x}\right] = n.$$

- Summing over j yields

$$0 = \sum_{j=1}^{K} N_j - \lambda \sum_j \pi_j = n - \lambda.$$

- Thus $\lambda = n$ and substituting back we find

$$\pi_j = \frac{N_j}{n}.$$

- Note that if $C_j = |\{i : Z_i = j\}|$, the number of X_i's generated from class j, then

$$\sum_{j=1}^{K} N_j = \sum_j \mathbb{E}\left[C_j \mid \boldsymbol{x}\right] = \mathbb{E}\left[\sum_j C_j \middle| \boldsymbol{x}\right] = n.$$

- Summing over j yields

$$0 = \sum_{j=1}^{K} N_j - \lambda \sum_j \pi_j = n - \lambda.$$

- Thus $\lambda = n$ and substituting back we find

$$\pi_j = \frac{N_j}{n}.$$

Recalling that

$$\gamma^{(i)}(j) = \mathbb{P}(Z_i = j \mid x_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \frac{\pi_j f_j(x_i \mid \mu_j, \Sigma_j)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x_i \mid \mu_\ell, \Sigma_\ell)}$$

$$N_j = \sum_{i=1}^{n} \gamma^{(j)}(i) = \mathbb{E}[C_j \mid \boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi]$$

any critical point $(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi)$ of of the log-likelihood function should obey the system of equations

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{n} \gamma^{(i)}(j) x_i \stackrel{\text{def}}{=} \bar{x}_j$$

$$\Sigma_j = \frac{1}{N_j} \sum_{i=1}^{n} \gamma^{(i)}(x_i - \mu_j)(x_i - \mu_j)^T$$

$$\pi_j = \frac{N_j}{n}.$$

- Note that these equations which any critical point must obey are not a closed-form solution, since the γ's depend on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
- They do lend themselves to an iterative scheme.
- The following is a special case of the EM-algorithm:
  - Initialize with any $\boldsymbol{\mu}_0, \Sigma_0, \pi_0$.
  - E-step. Use current $\boldsymbol{\mu}, \Sigma$ and π to calculate the responsibilities γ's.
  - M-step. Solve for $\boldsymbol{\mu}_u, \Sigma_u$ and π_u using these γ's.
  - Iterate.

- Note that these equations which any critical point must obey are not a closed-form solution, since the γ's depend on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
- They do lend themselves to an iterative scheme.
- The following is a special case of the EM-algorithm:
  - Initialize with any $\boldsymbol{\mu}_0, \Sigma_0, \pi_0$.
  - E-step. Use current $\boldsymbol{\mu}, \Sigma$ and π to calculate the responsibilities γ's.
  - M-step. Solve for $\boldsymbol{\mu}_u, \Sigma_u$ and π_u using these γ's.
  - Iterate.

- Note that these equations which any critical point must obey are not a closed-form solution, since the γ's depend on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
- They do lend themselves to an iterative scheme.
- The following is a special case of the EM-algorithm:
  - Initialize with any $\boldsymbol{\mu}_0, \Sigma_0, \pi_0$.
  - E-step. Use current $\boldsymbol{\mu}, \Sigma$ and π to calculate the responsibilities γ's.
  - M-step. Solve for $\boldsymbol{\mu}_u, \Sigma_u$ and π_u using these γ's.
  - Iterate.

- Note that these equations which any critical point must obey are not a closed-form solution, since the γ's depend on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
- They do lend themselves to an iterative scheme.
- The following is a special case of the EM-algorithm:
  - Initialize with any $\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \pi_0$.
  - E-step. Use current $\boldsymbol{\mu}, \Sigma$ and π to calculate the responsibilities γ's.
  - M-step. Solve for $\boldsymbol{\mu}_u, \Sigma_u$ and π_u using these γ's.
  - Iterate.

- Note that these equations which any critical point must obey are not a closed-form solution, since the γ's depend on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
- They do lend themselves to an iterative scheme.
- The following is a special case of the EM-algorithm:
  - Initialize with any $\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \pi_0$.
  - E-step. Use current $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and π to calculate the responsibilities γ's.
  - M-step. Solve for $\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u$ and π_u using these γ's.
  - Iterate.

- Note that these equations which any critical point must obey are not a closed-form solution, since the γ's depend on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
- They do lend themselves to an iterative scheme.
- The following is a special case of the EM-algorithm:
  - Initialize with any $\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \pi_0$.
  - *E*-step. Use current $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and π to calculate the responsibilities γ's.
  - *M*-step. Solve for $\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u$ and π_u using these γ's.
  - Iterate.

- Note that these equations which any critical point must obey are not a closed-form solution, since the γ's depend on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
- They do lend themselves to an iterative scheme.
- The following is a special case of the EM-algorithm:
  - Initialize with any $\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \pi_0$.
  - E-step. Use current $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and π to calculate the responsibilities γ's.
  - M-step. Solve for $\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u$ and π_u using these γ's.
  - Iterate.

- Each iteration is guaranteed to increase the likelihood function.
- In general, there could be local max.
- Also note that the likelihood has singularities in this problem: If you set one of the parameter values μ_j to x_{i_0} for a single i_0, then the likelihood at a such a parameter combination is

$$L(\boldsymbol{\mu},\boldsymbol{\sigma}) = \left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} + \sum_{k\neq j}\pi_k f_k(x_{i_0};\mu_k,\sigma_k)\right] \times \prod_{i\neq i_0}\left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2\sigma_j^2}(x_i-\mu_j)^2} + \sum_{k\neq j}\pi_k f_k(x_i;\mu_k,\sigma_k)\right]$$

- The likelihood tends to ∞ as $\sigma_j \to 0$!
- Note this does not occur in the $K=1$ case.
- In practice it may be assumed that σ_j is not (near) zero for any j, so a local max away from such singularities is desired.

- Each iteration is guaranteed to increase the likelihood function.
- In general, there could be local max.
- Also note that the likelihood has singularities in this problem: If you set one of the parameter values μ_j to x_{i_0} for a single i_0, then the likelihood at a such a parameter combination is

$$L(\boldsymbol{\mu},\boldsymbol{\sigma}) = \left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} + \sum_{k\neq j}\pi_k f_k(x_{i_0};\mu_k,\sigma_k)\right] \times \prod_{i\neq i_0}\left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2\sigma_j^2}(x_i-\mu_j)^2} + \sum_{k\neq j}\pi_k f_k(x_i;\mu_k,\sigma_k)\right]$$

- The likelihood tends to ∞ as $\sigma_j \to 0$!
- Note this does not occur in the $K = 1$ case.
- In practice it may be assumed that σ_j is not (near) zero for any j, so a local max away from such singularities is desired.

- Each iteration is guaranteed to increase the likelihood function.
- In general, there could be local max.
- Also note that the likelihood has singularities in this problem: If you set one of the parameter values μ_j to x_{i_0} for a single i_0, then the likelihood at a such a parameter combination is

$$L(\boldsymbol{\mu},\boldsymbol{\sigma}) = \left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} + \sum_{k\neq j}\pi_k f_k(x_{i_0};\mu_k,\sigma_k)\right]$$

$$\times \prod_{i\neq i_0}\left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2\sigma_j^2}(x_i-\mu_j)^2} + \sum_{k\neq j}\pi_k f_k(x_i;\mu_k,\sigma_k)\right]$$

- The likelihood tends to ∞ as $\sigma_j \to 0$!
- Note this does not occur in the $K=1$ case.
- In practice it may be assumed that σ_j is not (near) zero for any j, so a local max away from such singularities is desired.

- Each iteration is guaranteed to increase the likelihood function.
- In general, there could be local max.
- Also note that the likelihood has singularities in this problem: If you set one of the parameter values μ_j to x_{i_0} for a single i_0, then the likelihood at a such a parameter combination is

$$L(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} + \sum_{k \neq j} \pi_k f_k(x_{i_0}; \mu_k, \sigma_k) \right] \times \prod_{i \neq i_0} \left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2\sigma_j^2}(x_i - \mu_j)^2} + \sum_{k \neq j} \pi_k f_k(x_i; \mu_k, \sigma_k) \right]$$

- The likelihood tends to ∞ as $\sigma_j \to 0$!
- Note this does not occur in the $K = 1$ case.
- In practice it may be assumed that σ_j is not (near) zero for any j, so a local max away from such singularities is desired.

- Each iteration is guaranteed to increase the likelihood function.
- In general, there could be local max.
- Also note that the likelihood has singularities in this problem: If you set one of the parameter values μ_j to x_{i_0} for a single i_0, then the likelihood at a such a parameter combination is

$$L(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} + \sum_{k \neq j} \pi_k f_k(x_{i_0}; \mu_k, \sigma_k) \right]$$
$$\times \prod_{i \neq i_0} \left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2\sigma_j^2}(x_i - \mu_j)^2} + \sum_{k \neq j} \pi_k f_k(x_i; \mu_k, \sigma_k) \right]$$

- The likelihood tends to ∞ as $\sigma_j \to 0$!
- Note this does not occur in the $K = 1$ case.
- In practice it may be assumed that σ_j is not (near) zero for any j, so a local max away from such singularities is desired.

- Each iteration is guaranteed to increase the likelihood function.
- In general, there could be local max.
- Also note that the likelihood has singularities in this problem: If you set one of the parameter values μ_j to x_{i_0} for a single i_0, then the likelihood at a such a parameter combination is

$$L(\boldsymbol{\mu},\boldsymbol{\sigma}) = \left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} + \sum_{k\neq j} \pi_k f_k(x_{i_0};\mu_k,\sigma_k)\right] \times \prod_{i\neq i_0}\left[\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2\sigma_j^2}(x_i-\mu_j)^2} + \sum_{k\neq j}\pi_k f_k(x_i;\mu_k,\sigma_k)\right]$$

- The likelihood tends to ∞ as $\sigma_j \to 0$!
- Note this does not occur in the $K = 1$ case.
- In practice it may be assumed that σ_j is not (near) zero for any j, so a local max away from such singularities is desired.

Another algorithm: k-means

- Problem: How to partition data $x_1, \ldots, x_n$ in $\mathbb{R}^d$ into K distinct classes?
- The goal will be to pick classes and centers $\boldsymbol{\mu}$ to minimize

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{i,k} \|x_i - \mu_k\|^2$$

where

$$r_{i,k} = \begin{cases} 1 & \text{if } x_i \text{ is assigned to class } k \\ 0 & \text{otherwise} \end{cases}$$

- Note that no assumption about a distribution for the data is made here; the goal is simply to minimize the **distortion** measure J.

Another algorithm: k-means

- Problem: How to partition data $x_1, \ldots, x_n$ in $\mathbb{R}^d$ into K distinct classes?
- The goal will be to pick classes and centers $\boldsymbol{\mu}$ to minimize

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{i,k} \|x_i - \mu_k\|^2$$

where

$$r_{i,k} = \begin{cases} 1 & \text{if } x_i \text{ is assigned to class } k \\ 0 & \text{otherwise} \end{cases}$$

- Note that no assumption about a distribution for the data is made here; the goal is simply to minimize the **distortion** measure J.

Another algorithm: k-means

- Problem: How to partition data $x_1, \ldots, x_n$ in $\mathbb{R}^d$ into K distinct classes?
- The goal will be to pick classes and centers $\boldsymbol{\mu}$ to minimize

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{i,k} \|x_i - \mu_k\|^2$$

where

$$r_{i,k} = \begin{cases} 1 & \text{if } x_i \text{ is assigned to class } k \\ 0 & \text{otherwise} \end{cases}$$

- Note that no assumption about a distribution for the data is made here; the goal is simply to minimize the **distortion** measure J.

- E-step: pick r_i's to minimize J, holding the μ_k's constant.
- M-step: Minimize J holding r_i's constant.
- Note the distortion separates into K separate optimization problems in the M-step,

$$J = \sum_{k=1}^{K} \sum_{x_i \in \mathscr{C}_j} \|x_i - \mu_j\|^2$$

  Here, $\mathscr{C}_j = \{i : r_{i,j} = 1\}$ is the set of data points in class j.
- The solution to each is

$$\mu_j = \frac{1}{|\mathscr{C}_j|} \sum_{x_i \in \mathscr{C}_j} x_i$$

  which is the mean of the points in class j.
- In the E-step, the optimizing assignment of classes is to take class j to be all points which are closest to μ_j. (Exercise: Check this!)

- E-step: pick r_i's to minimize J, holding the μ_k's constant.
- M-step: Minimize J holding r_i's constant.
- Note the distortion separates into K separate optimization problems in the M-step,

$$J = \sum_{k=1}^{K} \sum_{x_i \in \mathscr{C}_j} \|x_i - \mu_j\|^2$$

  Here, $\mathscr{C}_j = \{i : r_{i,j} = 1\}$ is the set of data points in class j.
- The solution to each is

$$\mu_j = \frac{1}{|\mathscr{C}_j|} \sum_{x_i \in \mathscr{C}_j} x_i$$

  which is the mean of the points in class j.
- In the E-step, the optimizing assignment of classes is to take class j to be all points which are closest to μ_j. (Exercise: Check this!)

- E-step: pick r_i's to minimize J, holding the μ_k's constant.
- M-step: Minimize J holding r_i's constant.
- Note the distortion separates into K separate optimization problems in the M-step,

$$J = \sum_{k=1}^{K} \sum_{x_i \in \mathscr{C}_j} \|x_i - \mu_j\|^2$$

  Here, $\mathscr{C}_j = \{i : r_{i,j} = 1\}$ is the set of data points in class j.
- The solution to each is

$$\mu_j = \frac{1}{|\mathscr{C}_j|} \sum_{x_i \in \mathscr{C}_j} x_i$$

  which is the mean of the points in class j.
- In the E-step, the optimizing assignment of classes is to take class j to be all points which are closest to μ_j. (Exercise: Check this!)

- E-step: pick r_i's to minimize J, holding the μ_k's constant.
- M-step: Minimize J holding r_i's constant.
- Note the distortion separates into K separate optimization problems in the M-step,

$$J = \sum_{k=1}^{K} \sum_{x_i \in \mathscr{C}_j} \|x_i - \mu_j\|^2$$

  Here, $\mathscr{C}_j = \{i : r_{i,j} = 1\}$ is the set of data points in class j.
- The solution to each is

$$\mu_j = \frac{1}{|\mathscr{C}_j|} \sum_{x_i \in \mathscr{C}_j} x_i$$

  which is the mean of the points in class j.
- In the E-step, the optimizing assignment of classes is to take class j to be all points which are closest to μ_j. (Exercise: Check this!)

- E-step: pick r_i's to minimize J, holding the μ_k's constant.
- M-step: Minimize J holding r_i's constant.
- Note the distortion separates into K separate optimization problems in the M-step,

$$J = \sum_{k=1}^{K} \sum_{x_i \in \mathscr{C}_j} \|x_i - \mu_j\|^2$$

  Here, $\mathscr{C}_j = \{i : r_{i,j} = 1\}$ is the set of data points in class j.
- The solution to each is

$$\mu_j = \frac{1}{|\mathscr{C}_j|} \sum_{x_i \in \mathscr{C}_j} x_i$$

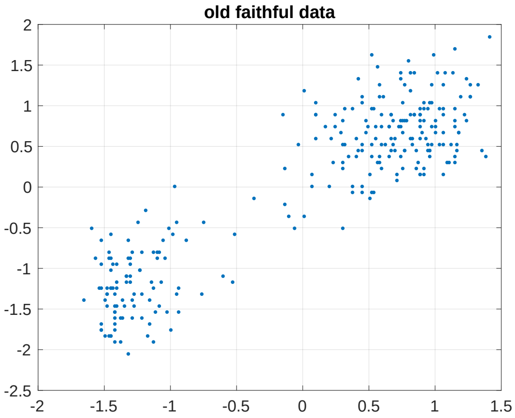
  which is the mean of the points in class j.
- In the E-step, the optimizing assignment of classes is to take class j to be all points which are closest to μ_j. (Exercise: Check this!)

- For estimation on the Gaussian mixture model, $\gamma^{(i)}(j)$ gives the posterior probability that the i-th variable belongs to the j-th component. This is a probabilistic assignment to classes.
- For k-means, a hard assignment is made. Each data point is assigned to one and only one class.
- Note that k-means does not make any model assumption.

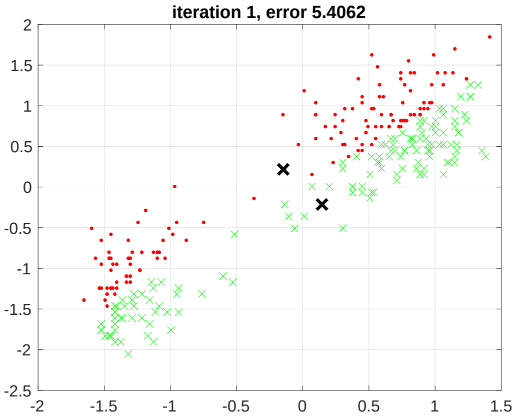
From the `README` file for the PMTK MATLAB code, available at `https://github.com/probml/pmtk3`

> *PMTK is a collection of Matlab/Octave functions, written by Matt Dunham, Kevin Murphy and various other people. The toolkit is Machine learning: a probabilistic perspective, but can also be used independently of this book. The goal is to provide a unified conceptual and software framework encompassing machine learning, graphical models, and Bayesian statistics (hence the logo).*

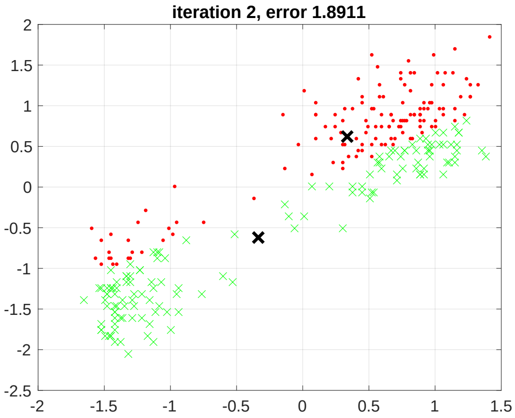
Initialization



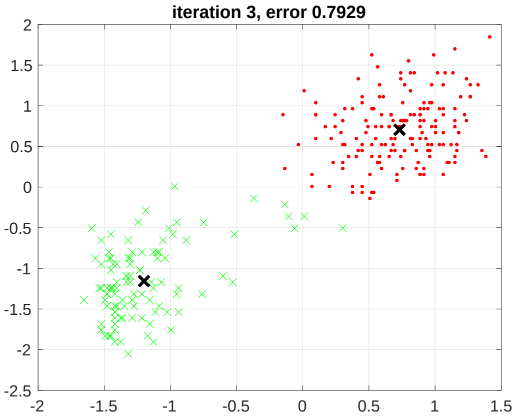
Interation 1



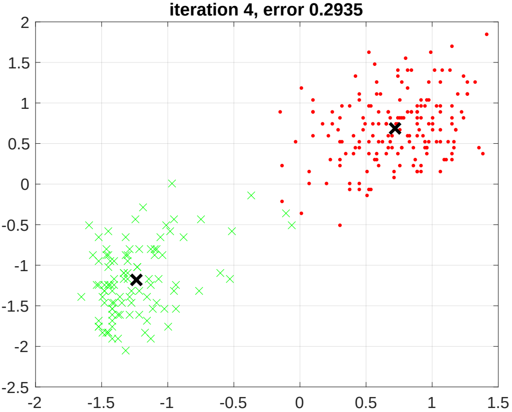
Interation 2



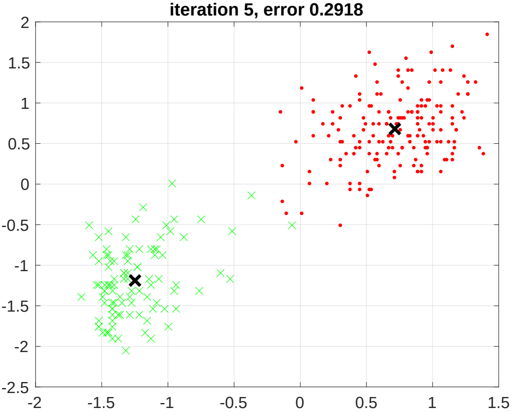
Interation 3



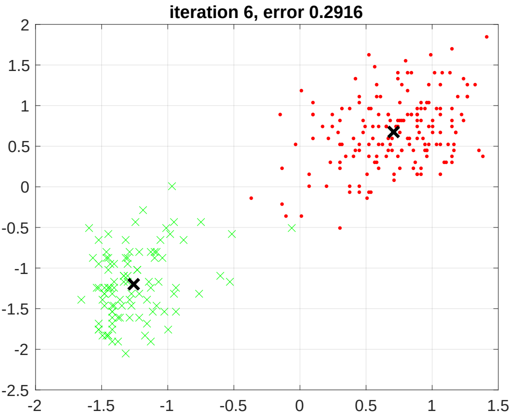
Interation 4



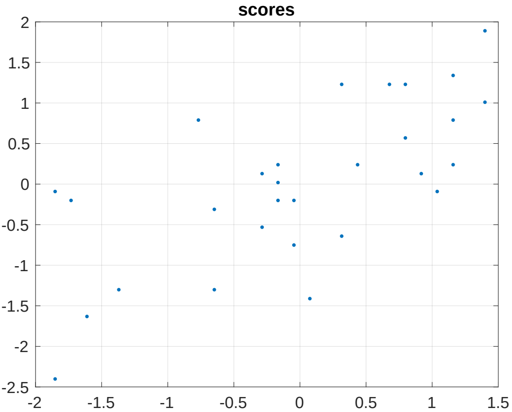
Interation 5



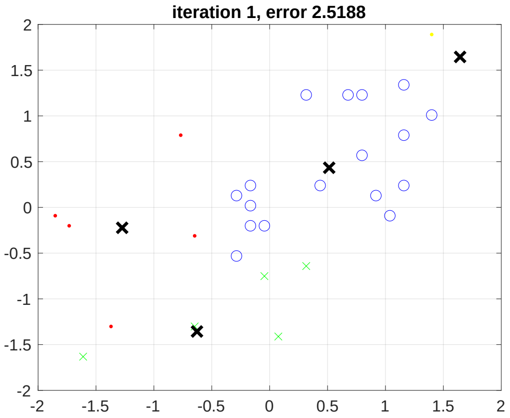
Interation 6



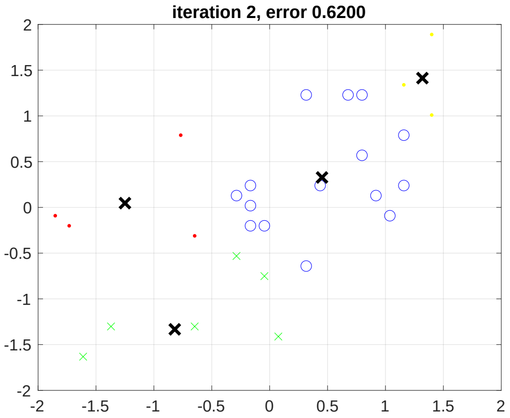
Initialization



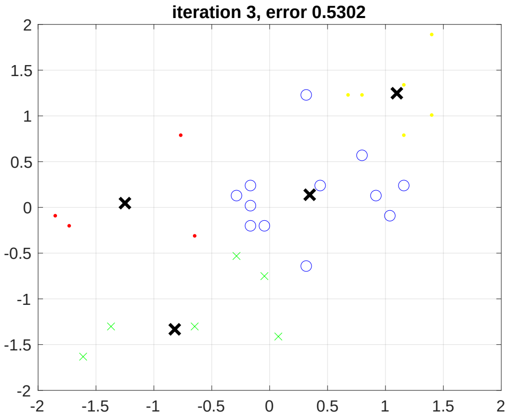
Interation 1



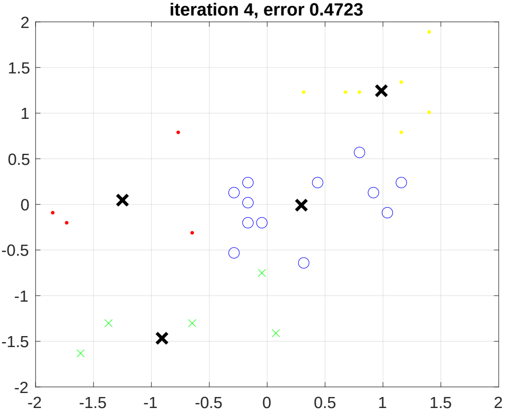
Interation 2



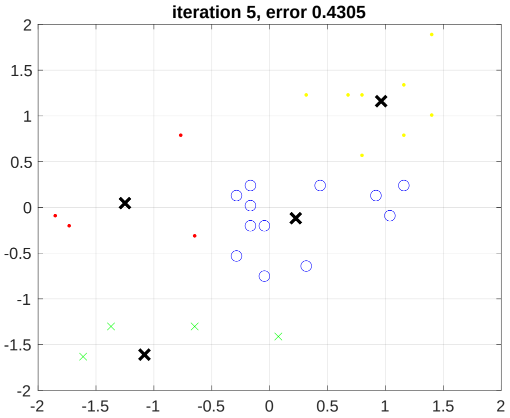
Interation 3



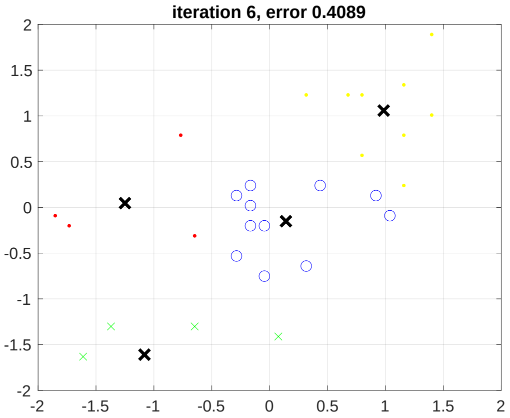
Interation 4



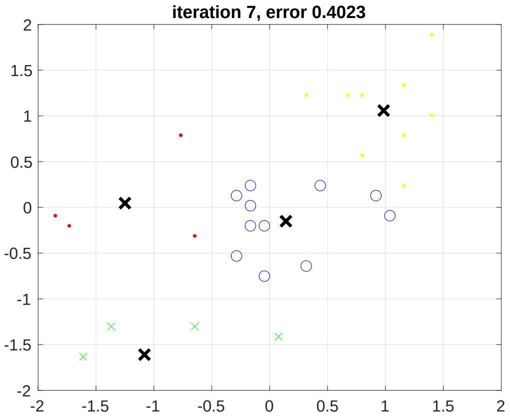
Interation 5



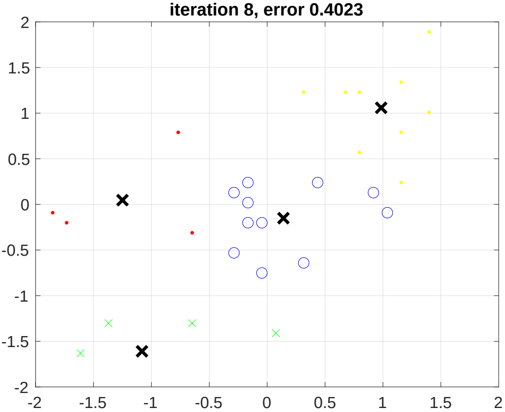
Interation 6



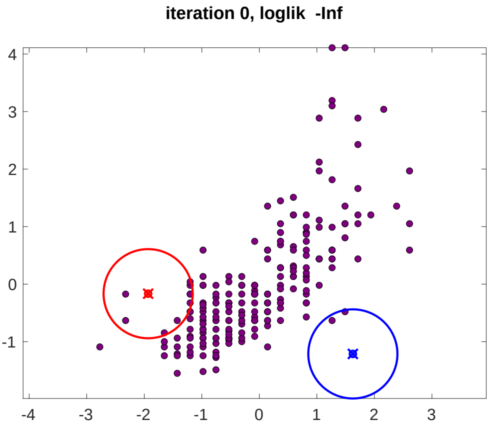
Interation 7



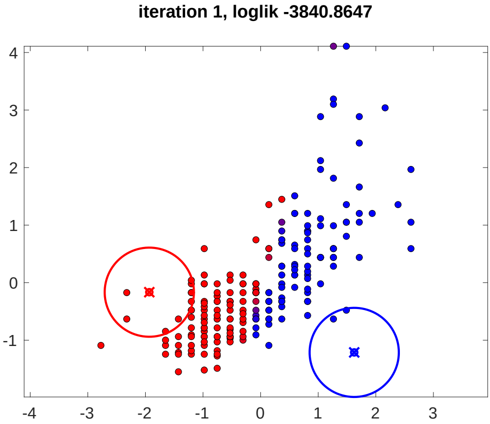
Interation 8



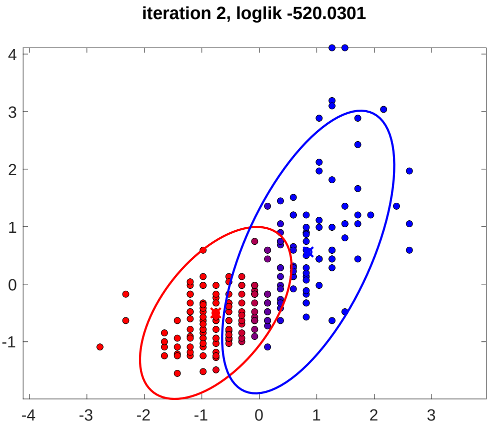
Initialization



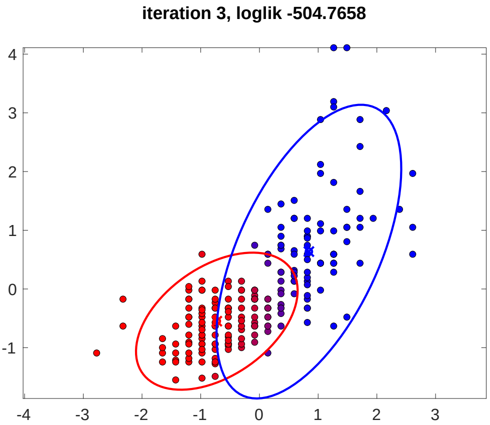
Interation 1



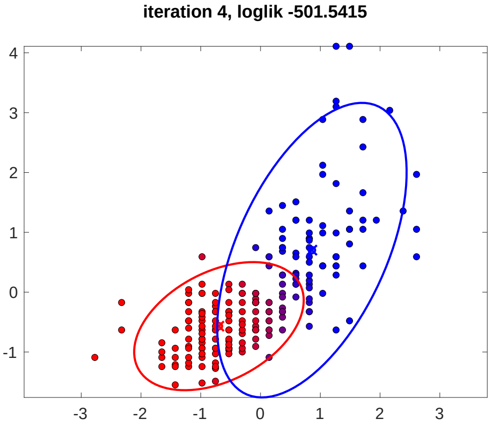
Interation 2



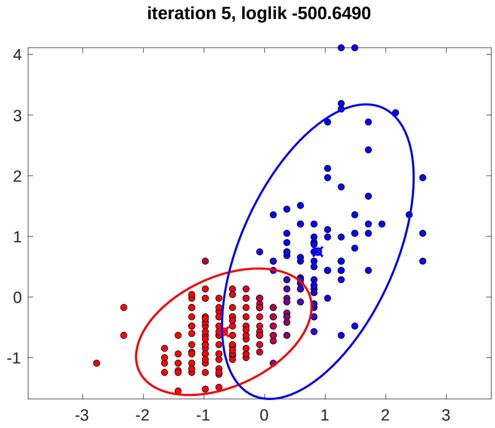
Interation 3



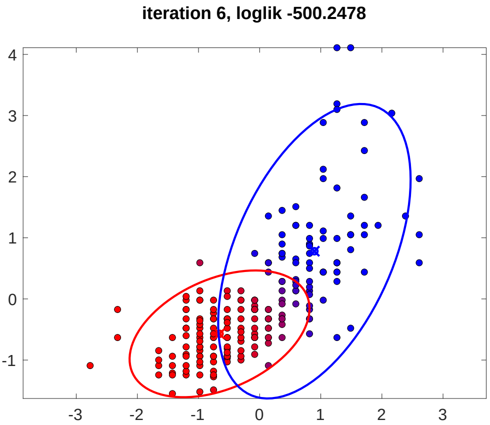
Interation 4

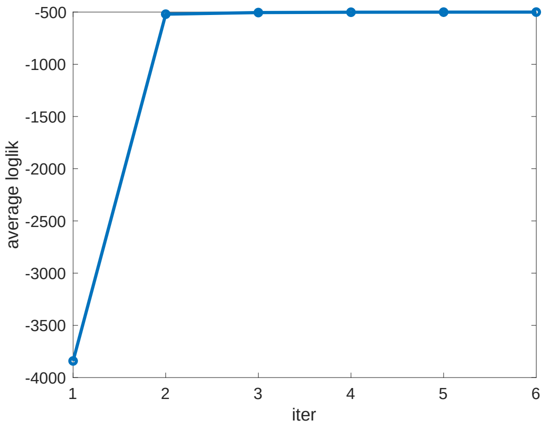


Interation 5



Interation 6





average loglik
-500
-1000
-1500
-2000
-2500
-3000
-3500
-4000
1
2
3
4
5
6
iter

EM in general

- Goal: Maximize $p_X(\boldsymbol{x};\theta)$ over θ.
- We have **latent** variables $\boldsymbol{Z}$ which are unobserved.
- The **complete** likelihood

$$p_{X,Z}(\boldsymbol{x},\boldsymbol{z};\theta)$$

  is often easier to optimize.
- E-step: calculate

$$Q(\theta,\theta_{\text{old}}) := \mathbb{E}_{\boldsymbol{Z}\sim p_{Z|X}(\cdot|\boldsymbol{x};\theta_{\text{old}})}\left[\ln p_{X,Z}(\boldsymbol{x},\boldsymbol{Z};\theta)\right]$$

- M-step: Optimize $Q(\theta,\theta_{\text{old}})$ over θ_{old} to find θ_{new}.

EM in general

- Goal: Maximize $p_X(\boldsymbol{x};\theta)$ over θ.
- We have **latent** variables $\boldsymbol{Z}$ which are unobserved.
- The **complete** likelihood

$$p_{X,Z}(\boldsymbol{x},\boldsymbol{z};\theta)$$

  is often easier to optimize.
- E-step: calculate

$$Q(\theta,\theta_{\text{old}}) := \mathbb{E}_{\boldsymbol{Z}\sim p_{Z|X}(\cdot|\boldsymbol{x};\theta_{\text{old}})}\left[\ln p_{X,Z}(\boldsymbol{x},\boldsymbol{Z};\theta)\right]$$

- M-step: Optimize $Q(\theta,\theta_{\text{old}})$ over θ_{old} to find θ_{new}.

EM in general

- Goal: Maximize $p_X(\boldsymbol{x};\theta)$ over θ.
- We have **latent** variables $\boldsymbol{Z}$ which are unobserved.
- The **complete** likelihood

$$p_{X,Z}(\boldsymbol{x},\boldsymbol{z};\theta)$$

  is often easier to optimize.
- E-step: calculate

$$Q(\theta,\theta_{\text{old}}) := \mathbb{E}_{\boldsymbol{Z}\sim p_{Z|X}(\cdot|\boldsymbol{x};\theta_{\text{old}})}\left[\ln p_{X,Z}(\boldsymbol{x},\boldsymbol{Z};\theta)\right]$$

- M-step: Optimize $Q(\theta,\theta_{\text{old}})$ over θ_{old} to find θ_{new}.

EM in general

- Goal: Maximize $p_X(\boldsymbol{x}; \theta)$ over θ.
- We have **latent** variables $\boldsymbol{Z}$ which are unobserved.
- The **complete** likelihood

$$p_{X,Z}(\boldsymbol{x}, \boldsymbol{z}; \theta)$$

  is often easier to optimize.
- E-step: calculate

$$Q(\theta, \theta_{\text{old}}) := \mathbb{E}_{\boldsymbol{Z} \sim p_{Z|X}(\cdot|\boldsymbol{x}; \theta_{\text{old}})} \left[\ln p_{X,Z}(\boldsymbol{x}, \boldsymbol{Z}; \theta) \right]$$

- M-step: Optimize $Q(\theta, \theta_{\text{old}})$ over θ_{old} to find θ_{new}.

EM in general

- Goal: Maximize $p_X(\boldsymbol{x};\theta)$ over θ.
- We have **latent** variables $\boldsymbol{Z}$ which are unobserved.
- The **complete** likelihood

$$p_{X,Z}(\boldsymbol{x},\boldsymbol{z};\theta)$$

  is often easier to optimize.
- E-step: calculate

$$Q(\theta,\theta_{\text{old}}) := \mathbb{E}_{\boldsymbol{Z}\sim p_{Z|X}(\cdot|\boldsymbol{x};\theta_{\text{old}})}\left[\ln p_{X,Z}(\boldsymbol{x},\boldsymbol{Z};\theta)\right]$$

- M-step: Optimize $Q(\theta,\theta_{\text{old}})$ over θ_{old} to find θ_{new}.

- In the case of Gaussian mixtures, it is convenient to use $Z_i = (Z_{i1}, \ldots, Z_{iK})$, where $Z_{ij} = 1$ if and only if X_i comes from the j-th component Normal, and $Z_{ij} = 0$ otherwise.
- In that case, the joint density for both $\boldsymbol{X}$ and $\boldsymbol{Z}$ is

$$\begin{aligned}\log f_{X,Z}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi)) &= \sum_{i=1}^{n} \log \prod_{j=1}^{K} [\pi_j f_j(x_i \mid \mu_j, \Sigma_j)]^{z_{i,j}} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{K} z_{i,j} \log \pi_j + \sum_{i=1}^{n} \sum_{j=1}^{K} z_{i,j} \log f_j(x_i \mid \mu_j, \Sigma_j)\end{aligned}$$

- In the case of Gaussian mixtures, it is convenient to use $Z_i = (Z_{i1}, \ldots, Z_{iK})$, where $Z_{ij} = 1$ if and only if X_i comes from the j-th component Normal, and $Z_{ij} = 0$ otherwise.
- In that case, the joint density for both $\boldsymbol{X}$ and $\boldsymbol{Z}$ is

$$\begin{aligned}
\log f_{X,Z}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi)) &= \sum_{i=1}^{n} \log \prod_{j=1}^{K} [\pi_j f_j(x_i \mid \mu_j, \Sigma_j)]^{z_{i,j}} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{K} z_{i,j} \log \pi_j + \sum_{i=1}^{n} \sum_{j=1}^{K} z_{i,j} \log f_j(x_i \mid \mu_j, \Sigma_j)
\end{aligned}$$

- Let $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$. Taking an expectation with respect to $\boldsymbol{Z}$ under the distribution $p_{Z|X}(\cdot \mid \boldsymbol{x}; \theta_{\text{old}})$ yields

$$\begin{aligned}
\mathbb{E}[\log p_{X,Z}(\boldsymbol{x}, \boldsymbol{Z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi)] &= \sum_{j=1}^{K} \log \pi_j \sum_{i=1}^{n} \mathbb{P}(Z_{i,j} = 1 \mid \boldsymbol{x}; \theta_{\text{old}}) \\
&\quad + \sum_{j=1}^{K} \sum_{i=1}^{n} \mathbb{P}(Z_{i,j} = 1 \mid \boldsymbol{x}; \theta_{\text{old}}) \log f_j(x_i \mid \mu_j, \Sigma_j) \\
&= \sum_j N_j \log \pi_j + \sum_j \sum_i \gamma^{(i)}(j) f_j(x_i \mid \mu_j, \Sigma_j)
\end{aligned}$$

- The γ's are computed just as before, using Bayes Theorem:

$$\gamma_j^{(i)} = \frac{\pi_j f_j(x_i; \theta_{\text{old}})}{\sum_k \pi_k f_k(x_i; \theta_{\text{old}})}$$

Mixtures of Bernoullis

- Consider a vector of D binary variables X_i, where $i = 1, \ldots, D$.

$$p(x_1, \ldots, x_d; \boldsymbol{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i} (1 - \mu_i)^{1 - x_i} .$$

- Now consider a mixture of these distributions:

$$p(x_1, \ldots, x_d; \mu, \pi) = \sum_{j=1}^{k} \pi_j p(x_1, \ldots, x_d; \mu_j) .$$

- The modeling advantage is that there can now be non-zero correlation between the bits.

Mixtures of Bernoullis

- Consider a vector of D binary variables X_i, where $i = 1, \ldots, D$.

$$p(x_1, \ldots, x_d; \boldsymbol{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i} (1 - \mu_i)^{1 - x_i} \,.$$

- Now consider a mixture of these distributions:

$$p(x_1, \ldots, x_d; \mu, \pi) = \sum_{j=1}^{k} \pi_j p(x_1, \ldots, x_d; \mu_j) \,.$$

- The modeling advantage is that there can now be non-zero correlation between the bits.

Mixtures of Bernoullis

- Consider a vector of D binary variables X_i, where $i = 1, \ldots, D$.

$$p(x_1, \ldots, x_d; \boldsymbol{\mu}) = \prod_{i=1}^{D} \mu_i^{x_i} (1 - \mu_i)^{1 - x_i} \,.$$

- Now consider a mixture of these distributions:

$$p(x_1, \ldots, x_d; \mu, \pi) = \sum_{j=1}^{k} \pi_j p(x_1, \ldots, x_d; \mu_j) \,.$$

- The modeling advantage is that there can now be non-zero correlation between the bits.

- Now sample $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ independently from the above mixture distribution. Note each $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,d})$ has d-components.
-

$$p_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) = \prod_{i=1}^{n} \prod_{j=1}^{K} \left[\prod_{\ell=1}^{D} \mu_{j,\ell}^{x_{i,\ell}} (1 - \mu_{j,\ell})^{1 - x_{i,\ell}} \pi_j \right]^{z_{i,j}}$$

$$\log p_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{n} \sum_{j=1}^{K} z_{i,j} \Big[\log \pi_j + \sum_{\ell=1}^{D} [x_{i,\ell} \log \mu_{j,\ell} + (1 - x_{i,\ell}) \log(1 - \mu_{j,\ell})] \Big].$$

- Let $\theta = (\boldsymbol{\mu}, \boldsymbol{\pi})$. As before,

$$\gamma_j^{(i)} = \mathbb{P}(Z_{i,j} = 1 \mid \boldsymbol{x}_i; \theta_{\text{old}}) = \frac{\pi_j p(\boldsymbol{x}_i; \mu_j)}{\sum_k \pi_k p(\boldsymbol{x}_i; \mu_k)}$$

- Recall

$$Q(\theta, \theta_0) = \mathbb{E}[p_{X,Z}(\boldsymbol{x}, \boldsymbol{Z})],$$

where the expectation is with respect to $\boldsymbol{Z}$ with law $p_{Z|X}(\boldsymbol{z} \mid \boldsymbol{x}; \theta_0)$

- Now sample $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ independently from the above mixture distribution. Note each $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,d})$ has d-components.
-

$$p_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) = \prod_{i=1}^{n} \prod_{j=1}^{K} \left[\prod_{\ell=1}^{D} \mu_{j,\ell}^{x_{i,\ell}} (1 - \mu_{j,\ell})^{1 - x_{i,\ell}} \pi_j \right]^{z_{i,j}}$$

$$\log p_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{n} \sum_{j=1}^{K} z_{i,j} \Big[\log \pi_j + \sum_{\ell=1}^{D} [x_{i,\ell} \log \mu_{j,\ell} + (1 - x_{i,\ell}) \log(1 - \mu_{j,\ell})] \Big].$$

- Let $\theta = (\boldsymbol{\mu}, \boldsymbol{\pi})$. As before,

$$\gamma_j^{(i)} = \mathbb{P}(Z_{i,j} = 1 \mid \boldsymbol{x}_i; \theta_{\text{old}}) = \frac{\pi_j p(\boldsymbol{x}_i; \mu_j)}{\sum_k \pi_k p(\boldsymbol{x}_i; \mu_k)}$$

- Recall

$$Q(\theta, \theta_0) = \mathbb{E}[p_{X,Z}(\boldsymbol{x}, \boldsymbol{Z})],$$

where the expectation is with respect to $\boldsymbol{Z}$ with law $p_{Z|X}(\boldsymbol{z} \mid \boldsymbol{x}; \theta_0)$

- Now sample $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ independently from the above mixture distribution. Note each $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,d})$ has d-components.
-
$$p_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) = \prod_{i=1}^{n} \prod_{j=1}^{K} \left[\prod_{\ell=1}^{D} \mu_{j,\ell}^{x_{i,\ell}} (1 - \mu_{j,\ell})^{1 - x_{i,\ell}} \pi_j \right]^{z_{i,j}}$$
$$\log p_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{n} \sum_{j=1}^{K} z_{i,j} \Bigg[\log \pi_j + \sum_{\ell=1}^{D} [x_{i,\ell} \log \mu_{j,\ell} + (1 - x_{i,\ell}) \log(1 - \mu_{j,\ell})] \Bigg].$$
- Let $\theta = (\boldsymbol{\mu}, \boldsymbol{\pi})$. As before,
$$\gamma_j^{(i)} = \mathbb{P}(Z_{i,j} = 1 \mid \boldsymbol{x}_i; \theta_{\text{old}}) = \frac{\pi_j p(\boldsymbol{x}_i; \mu_j)}{\sum_k \pi_k p(\boldsymbol{x}_i; \mu_k)}$$
- Recall
$$Q(\theta, \theta_0) = \mathbb{E}[p_{X,Z}(\boldsymbol{x}, \boldsymbol{Z})],$$
where the expectation is with respect to $\boldsymbol{Z}$ with law $p_{Z|X}(\boldsymbol{z} \mid \boldsymbol{x}; \theta_0)$

- Now sample $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ independently from the above mixture distribution. Note each $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,d})$ has d-components.
- $$p_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) = \prod_{i=1}^{n} \prod_{j=1}^{K} \left[\prod_{\ell=1}^{D} \mu_{j,\ell}^{x_{i,\ell}} (1 - \mu_{j,\ell})^{1 - x_{i,\ell}} \pi_j \right]^{z_{i,j}}$$
$$\begin{aligned} \log p_{X,Z}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{n} \sum_{j=1}^{K} z_{i,j} \Big[\log \pi_j + \sum_{\ell=1}^{D} [x_{i,\ell} \log \mu_{j,\ell} \\ + (1 - x_{i,\ell}) \log(1 - \mu_{j,\ell})] \Big] . \end{aligned}$$
- Let $\theta = (\boldsymbol{\mu}, \boldsymbol{\pi})$. As before,
$$\gamma_j^{(i)} = \mathbb{P}(Z_{i,j} = 1 \mid \boldsymbol{x}_i; \theta_{\text{old}}) = \frac{\pi_j p(\boldsymbol{x}_i; \mu_j)}{\sum_k \pi_k p(\boldsymbol{x}_i; \mu_k)}$$
- Recall
$$Q(\theta, \theta_0) = \mathbb{E}[p_{X,Z}(\boldsymbol{x}, \boldsymbol{Z})] ,$$
where the expectation is with respect to $\boldsymbol{Z}$ with law $p_{Z|X}(\boldsymbol{z} \mid \boldsymbol{x}; \theta_0)$

- $$Q(\theta,\theta_0) = \underbrace{\sum_{j=1}^{K} \gamma^{(i)}(j) \log \pi_j}_{L_1}$$
$$+ \underbrace{\sum_j \sum_\ell \overbrace{\left[\log \mu_{j,\ell} \sum_i \gamma^{(i)}(j) x_{i,\ell} + \log(1-\mu_{j,\ell}) \sum_i \gamma^{(i)}(j)(1-x_{i,\ell}) \right]}^{L_{2,j,\ell}}}_{L_2}$$
- The maximization given the γ's can be carried out separately for L_1 and L_2.
- L_1 we have seen before and it has maximum at $\pi_j = N_j/N$, where $N_j = \sum_i \gamma_j^{(i)}$.

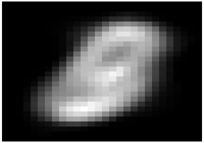
- For L_2, fix j, ℓ.

$$\frac{dL_{2,j,\ell}}{d\mu_{j,\ell}} = \frac{1}{\mu_{j,\ell}} \sum_i \gamma^{(i)}(j) x_{i,\ell} - \frac{1}{(1-\mu_{j,\ell})} (N_j - \sum_i \gamma^{(i)}(j) x_{i,\ell})$$

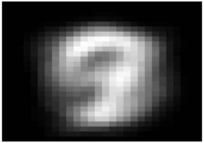
- Setting the above equal to 0 and solving yields, coordinatewise,
-
$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_i \gamma_j^{(i)} \boldsymbol{x}_i,$$

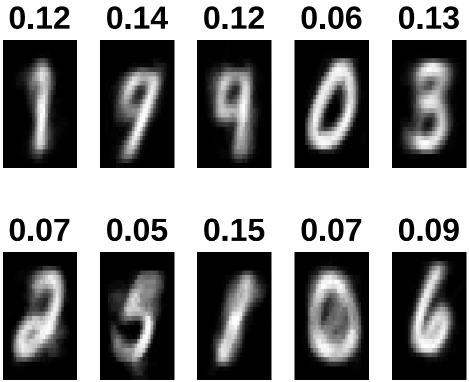
Mixture of Bernoullis

0.51



0.49




0.12 0.14 0.12 0.06 0.13
0.07 0.05 0.15 0.07 0.09