Mixture models

- Nonparametric model of density: makes no assumptions about the structure of the distribution, but needs to keep all the training data around.

- Single global parametric model (e.g., Gaussian): compact, but overly restrictive.
Mixture models

- Nonparametric model of density: makes no assumptions about the structure of the distribution, but needs to keep all the training data around.

- Single global parametric model (e.g., Gaussian): compact, but overly restrictive.

- Mixture model: suppose that training examples come from a small number of distinct “types,” and the distribution of each type can be described by a simple parametric model (e.g., a Gaussian).
Examples

- Images of the same person under different conditions: with/without glasses, smiling or frowning, frontal or profile.
- Images of the same category but different sorts of objects: dogs of different breeds.
- Multiple topics within the same document.
- Different ways of pronouncing the same phonemes.
Mixture models

- Assumptions:
  - $k$ underlying types (components);
  - $y_i$ is the identity of the component “responsible” for $x_i$;
  - $y_i$ is a hidden (latent) variable: never observed.

The mixture density is characterized by:

- Mixing probabilities $p(y = c)$ (have to add up to 1)
- Component densities $p(x | y = c)$

The mixture density is $p(x) = \sum_{c=1}^{k} p(y = c) p(x | y = c)$. 
Mixture models

Assumptions:
- \(k\) underlying types (components);
- \(y_i\) is the identity of the component “responsible” for \(x_i\);
- \(y_i\) is a hidden (latent) variable: never observed.

The mixture density is characterized by:
- Mixing probabilities \(p(y = c)\) (have to add up to 1)
- Component densities \(p(x \mid y = c)\)
Mixture models

Assumptions:
- $k$ underlying types (components);
- $y_i$ is the identity of the component “responsible” for $x_i$;
- $y_i$ is a hidden (latent) variable: never observed.

The mixture density is characterized by:
- Mixing probabilities $p(y = c)$ (have to add up to 1)
- Component densities $p(x | y = c)$

The mixture density is
Mixture models

Assumptions:
- $k$ underlying types (components);
- $y_i$ is the identity of the component “responsible” for $x_i$;
- $y_i$ is a hidden (latent) variable: never observed.

The mixture density is characterized by:
- Mixing probabilities $p(y = c)$ (have to add up to 1)
- Component densities $p(x | y = c)$

The mixture density is

$$p(x) = \sum_{c=1}^{k} p(y = c) p(x | y = c).$$
Example: Gaussian mixture

\[ 0.7 \times \text{Gaussian 1} + 0.3 \times \text{Gaussian 2} = \text{Mixture Density} \]
The generative process with \( k \)-component mixture:

- The parameters \( \theta_c \) for each component \( c \) are fixed.
- Draw \( y_i \sim [p_1, \ldots, p_k] \);
- Given \( y_i \), draw \( x_i \sim p(x; \theta_{y_i}) \).
The generative process with $k$-component mixture:

- The parameters $\theta_c$ for each component $c$ are fixed.
- Draw $y_i \sim [p_1, \ldots, p_k]$;
- Given $y_i$, draw $x_i \sim p(x; \theta_{y_i})$.

The graphical model representation:

\[
p(x, y; \theta, p) = p(y) \cdot p(x; \theta_{y})
\]

\[
p(x; \theta, p) = \sum_{c=1}^{k} p_c \cdot p(x; \theta_c)
\]

($p$ denotes the vector of mixing probabilities and $\theta$ denotes the vector of parameters of the mixture components)
Suppose each component is a multivariate Gaussian:

\[ p(x \mid y = c) = \mathcal{N}(x; \mu_c, \Sigma_c), \]

\[ \mathcal{N}(x; \mu_c, \Sigma_c) = \frac{1}{(2\pi)^{d/2} |\Sigma_c|^{1/2}} e^{-\frac{1}{2}(x-\mu_c)^T \Sigma_c^{-1}(x-\mu_c)}. \]

The Gaussian mixture density is

\[ p(x; \theta, p) = \sum_{c=1}^{k} p_c \cdot \mathcal{N}(x; \mu_c, \Sigma_c), \]

where \( \theta = [\mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k] \).
Maximum likelihood estimation of Gaussian parameters

1. Suppose we draw samples $x_1, \ldots, x_N$ from a Gaussian distribution with mean $\mu$ and covariance matrix $\Sigma$.

2. The likelihood of the data is

$$
\prod_{i=1}^{N} \mathcal{N}(x_i; \mu, \Sigma).
$$

3. Maximum likelihood (ML) estimates of parameters are obtained by maximizing the likelihood (or the log of the likelihood) as a function of $\mu$ and $\Sigma$. 

$$
\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i,

\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{\mu})(x_i - \hat{\mu})^T.
$$
Maximum likelihood estimation of Gaussian parameters

- Suppose we draw samples $x_1, \ldots, x_N$ from a Gaussian distribution with mean $\mu$ and covariance matrix $\Sigma$.

- The likelihood of the data is

$$
\prod_{i=1}^{N} \mathcal{N}(x_i; \mu, \Sigma).
$$

- Maximum likelihood (ML) estimates of parameters are obtained by maximizing the likelihood (or the log of the likelihood) as a function of $\mu$ and $\Sigma$.

- ML estimates of Gaussian mean and covariance are

$$
\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad \hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{\mu})(x_i - \hat{\mu})^T.
$$
For each component $c$ and each data point $i$, let us introduce a binary \textit{indicator variable}

$$z_{ic} = \begin{cases} 
1 & \text{if } y_i = c, \\
0 & \text{otherwise.}
\end{cases}$$
For each component $c$ and each data point $i$, let us introduce a binary *indicator variable*:

$$z_{ic} = \begin{cases} 
1 & \text{if } y_i = c, \\
0 & \text{otherwise.}
\end{cases}$$

The count of examples from $c$-th component:

$$N_c = \sum_{i=1}^{N} z_{ic}.$$
Suppose we know all the assignments $z_{ic}$. How do we obtain maximum likelihood estimates of the parameters $p_c, \mu_c, \Sigma_c$ for each component?

\[
\hat{p}_c = \frac{1}{N_c} \sum_{i=1}^{N} z_{ic}, \\
\hat{\mu}_c = \frac{1}{N_c} \sum_{i=1}^{N} z_{ic} x_i, \\
\hat{\Sigma}_c = \frac{1}{N_c} \sum_{i=1}^{N} z_{ic} (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^T.
\]
Suppose we know all the assignments $z_{ic}$. How do we obtain maximum likelihood estimates of the parameters $p_c$, $\mu_c$, $\Sigma_c$ for each component?

$$\hat{p}_c = \frac{N_c}{N},$$

$$\hat{\mu}_c = \frac{1}{N_c} \sum_{i=1}^{N} z_{ic} x_i,$$

$$\hat{\Sigma}_c = \frac{1}{N_c} \sum_{i=1}^{N} z_{ic} (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^T.$$
Credit assignment

- When we don’t know $y_i$, we face a credit assignment problem: which component is responsible for $x_i$?

- Suppose for a moment that we do know component parameters $\theta = [\mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k]$ and mixing probabilities $p = [p_1, \ldots, p_k]$. 
When we don’t know $y_i$, we face a credit assignment problem: which component is responsible for $x_i$?

Suppose for a moment that we do know component parameters $\theta = [\mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k]$ and mixing probabilities $p = [p_1, \ldots, p_k]$.

Then, for each point $x_i$, we can compute the posterior probability of each label using the Bayes rule:

$$\hat{p}(y = c|x_i; \theta, p) = \frac{p_c \cdot p(x_i; \mu_c, \Sigma_c)}{\sum_{l=1}^{k} p_l \cdot p(x_i; \mu_l, \Sigma_l)}$$
When we don’t know \( y_i \), we face a credit assignment problem: which component is responsible for \( x_i \)?

Suppose for a moment that we do know component parameters \( \theta = [\mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k] \) and mixing probabilities \( p = [p_1, \ldots, p_k] \).

Then, for each point \( x_i \), we can compute the posterior probability of each label using the Bayes rule:

\[
\hat{p}(y = c|x_i; \theta, p) = \frac{p_c \cdot p(x_i; \mu_c, \Sigma_c)}{\sum_{l=1}^{k} p_l \cdot p(x_i; \mu_l, \Sigma_l)}
\]

We will call \( \gamma_{ic} = \hat{p}(y = c|x_i; \theta, p) \) the responsibility of the \( c \)-th component for \( x_i \).

Note: \( \sum_{c=1}^{k} \gamma_{ic} = 1 \).
The “complete data” likelihood (when $z$ are known):

$$p(X_N, Z_N; p, \theta) = \prod_{i=1}^{N} \prod_{c=1}^{k} (p_c \mathcal{N}(x_i; \mu_c, \Sigma_c))^{z_{ic}}.$$
The “complete data” likelihood (when $z$ are known):

$$p(X_N, Z_N; \mathbf{p}, \theta) = \prod_{i=1}^{N} \prod_{c=1}^{k} (p_c \mathcal{N}(x_i; \mu_c, \Sigma_c))^{z_{ic}}.$$ 

Instead of maximizing the likelihood, it is more convenient to maximize the log of the likelihood:

$$\log p(X_N, Z_N; \mathbf{p}, \theta) =$$
The “complete data” likelihood (when $z$ are known):

$$p(X_N, Z_N; p, \theta) = \prod_{i=1}^{N} \prod_{c=1}^{k} (p_c \mathcal{N}(x_i; \mu_c, \Sigma_c))^{z_{ic}}.$$ 

Instead of maximizing the likelihood, it is more convenient to maximize the log of the likelihood:

$$\log p(X_N, Z_N; p, \theta) = \sum_{i=1}^{N} \sum_{c=1}^{k} z_{ic} (\log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c)).$$
The “complete data” likelihood (when $z$ are known):

$$p(X_N, Z_N; p, \theta) = \prod_{i=1}^{N} \prod_{c=1}^{k} (p_c \mathcal{N}(x_i; \mu_c, \Sigma_c))^{z_{ic}}.$$

Instead of maximizing the likelihood, it is more convenient to maximize the log of the likelihood:

$$\log p(X_N, Z_N; p, \theta) = \sum_{i=1}^{N} \sum_{c=1}^{k} z_{ic} (\log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c)).$$

When we don’t know $z$, we can’t compute the complete likelihood, but can take its expectation w.r.t. the posterior distribution of $z$ given estimates $p, \theta$ of the mixture model parameters:

$$\mathbb{E}_z [\log p(x_i, z_{ic}; \theta, p)].$$
The “complete data” likelihood (when $z$ are known):

$$p(X_N, Z_N; p, \theta) = \prod_{i=1}^{N} \prod_{c=1}^{k} (p_c \mathcal{N}(x_i; \mu_c, \Sigma_c))^{z_{ic}}.$$ 

Instead of maximizing the likelihood, it is more convenient to maximize the log of the likelihood:

$$\log p(X_N, Z_N; p, \theta) = \sum_{i=1}^{N} \sum_{c=1}^{k} z_{ic} (\log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c)).$$

When we don’t know $z$, we can’t compute the complete likelihood, but can take its expectation w.r.t. the posterior distribution of $z$ given estimates $p, \theta$ of the mixture model parameters:

$$\mathbb{E}_z [\log p(x_i, z_{ic}; \theta, p)].$$

In the posterior distribution of $z$, the probability of each $z_{ic} = 1$ is simply its responsibility $\gamma_{ic}$. 
Computing the expected likelihood

Source: G. Shakhnarovich

\[ E_z \left[ \sum_{i=1}^{N} \sum_{c=1}^{k} z_{ic} (\log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c)) \right] \]
Computing the expected likelihood

$$\mathbb{E}_z \left[ \sum_{i=1}^{N} \sum_{c=1}^{k} z_{ic} \left( \log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c) \right) \right]$$

$$= \sum_{i=1}^{N} \sum_{c=1}^{k} \mathbb{E}_{z_{ic}} \left[ z_{ic} \left( \log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c) \right) \right].$$
Computing the expected likelihood

\[
\mathbb{E}_Z \left[ \sum_{i=1}^{N} \sum_{c=1}^{k} z_{ic} \left( \log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c) \right) \right]
\]

\[
= \sum_{i=1}^{N} \sum_{c=1}^{k} \mathbb{E}_{z_{ic}} \left[ z_{ic} \left( \log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c) \right) \right].
\]

\[
\mathbb{E}_{z_{ic}} [z_{ic}L] = \Pr[z_{ic} = 0](0 \cdot L) + \Pr[z_{ic} = 1](1 \cdot L) = \gamma_{ic}L.
\]
Computing the expected likelihood

\[
\mathbb{E}_z \left[ \sum_{i=1}^N \sum_{c=1}^k z_{ic} \left( \log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c) \right) \right] \\
= \sum_{i=1}^N \sum_{c=1}^k \mathbb{E}_{z_{ic}} \left[ \underbrace{z_{ic} \left( \log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c) \right)}_{L} \right].
\]

\[
\mathbb{E}_{z_{ic}} [z_{ic}L] = \Pr[z_{ic} = 0](0 \cdot L) + \Pr[z_{ic} = 1](1 \cdot L) = \gamma_{ic} L.
\]

Therefore,

\[
\mathbb{E}_z \left[ \log p(X_N, Z_N; p, \theta) \right] = \sum_{i=1}^N \sum_{c=1}^k \gamma_{ic} \left( \log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c) \right).
\]
\[ E_z [\log p(X_N, Z_N; p, \theta)] = \sum_{i=1}^{N} \sum_{c=1}^{k} \gamma_{ic} (\log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c)). \]

We can find \( p, \theta \) that maximize the expected likelihood by setting derivatives to zero and, for \( p \), using Lagrange multipliers to enforce \( \sum_c p_c = 1 \).
\[ \mathbb{E}_z [\log p(X_N, Z_N; \mathbf{p}, \theta)] = \sum_{i=1}^{N} \sum_{c=1}^{k} \gamma_{ic} (\log p_c + \log \mathcal{N}(x_i; \mu_c, \Sigma_c)). \]

We can find \( \mathbf{p}, \theta \) that maximize the expected likelihood by setting derivatives to zero and, for \( \mathbf{p} \), using Lagrange multipliers to enforce \( \sum_c p_c = 1 \).

\[
\hat{p}_c = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ic}, \\
\hat{\mu}_c = \frac{1}{\sum_{i=1}^{N} \gamma_{ic}} \sum_{i=1}^{N} \gamma_{ic} x_i, \\
\hat{\Sigma}_c = \frac{1}{\sum_{i=1}^{N} \gamma_{ic}} \sum_{i=1}^{N} \gamma_{ic} (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^T.
\]
What if we know the **indicators** (assignments of components to data points) but not the **parameters** (means, covariances, and component weights)?
What if we know the **indicators** (assignments of components to data points) but not the **parameters** (means, covariances, and component weights)?

- We can do ML estimation of the parameters – and we are done.

What if we know the **parameters** but not the **indicators**?

First, we compute the posteriors of indicators. With known posteriors, we can estimate parameters that maximize the expected likelihood – and then we are done.

But in reality we know neither the parameters nor the indicators. What to do?
What if we know the **indicators** (assignments of components to data points) but not the **parameters** (means, covariances, and component weights)?

- We can do ML estimation of the parameters – and we are done.

What if we know the **parameters** but not the **indicators**?
What if we know the **indicators** (assignments of components to data points) but not the **parameters** (means, covariances, and component weights)?

- We can do ML estimation of the parameters – and we are done.

What if we know the **parameters** but not the **indicators**?

- First, we compute the posteriors of indicators.
What if we know the **indicators** (assignments of components to data points) but not the **parameters** (means, covariances, and component weights)?

- We can do ML estimation of the parameters – and we are done.

What if we know the **parameters** but not the **indicators**?

- First, we compute the posteriors of indicators.
- With known posteriors, we can estimate parameters that maximize the *expected* likelihood – and then we are done.
What if we know the **indicators** (assignments of components to data points) but not the **parameters** (means, covariances, and component weights)?

- We can do ML estimation of the parameters – and we are done.

What if we know the **parameters** but not the **indicators**?

- First, we compute the posteriors of indicators.
- With known posteriors, we can estimate parameters that maximize the *expected* likelihood – and then we are done.

But in reality we know neither the parameters nor the indicators. What to do?
The EM algorithm

- Start with a guess of $\theta, p$.
  - For example, random $\theta$ and $p_c = 1/k$. 

Source: G. Shakhnarovich
The EM algorithm

- Start with a guess of $\theta$, $p$.
  - For example, random $\theta$ and $p_c = 1/k$.
- Iterate between:
  
  **E-step**  Compute values of expected assignments, i.e. calculate $\gamma_{ic}$, using current estimates of $\theta$, $p$. 

The EM algorithm

- Start with a guess of $\theta, p$.
  - For example, random $\theta$ and $p_c = 1/k$.
- Iterate between:
  - **E-step** Compute values of expected assignments, i.e. calculate $\gamma_{ic}$, using current estimates of $\theta, p$.
  - **M-step** Obtain new estimates of $\theta, p$ by maximizing the expected likelihood under current assignments $\gamma_{ic}$. 

Source: G. Shakhnarovich
EM for Gaussian mixtures – summary

- Initialize: random $\mu_{c}^{old}$, $\Sigma_{c}^{old}$, $p_{c}^{old} = 1/k$ for $c = 1, \ldots, k$.
- Iterate until convergence:
EM for Gaussian mixtures – summary

- Initialize: random $\mu_{c}^{old}$, $\Sigma_{c}^{old}$, $p_{c}^{old} = 1/k$ for $c = 1, \ldots, k$.
- Iterate until convergence:

**E-step** estimate responsibilities:

$$\gamma_{ic} = \frac{p_{c}^{old} \mathcal{N}(x_{i}; \mu_{c}^{old}, \Sigma_{c}^{old})}{\sum_{l=1}^{k} p_{l}^{old} \mathcal{N}(x_{i}; \mu_{l}^{old}, \Sigma_{l}^{old})}$$
EM for Gaussian mixtures – summary

- Initialize: random $\mu_{c}^{old}$, $\Sigma_{c}^{old}$, $p_{c}^{old} = 1/k$ for $c = 1, \ldots, k$.

- Iterate until convergence:

  **E-step** estimate responsibilities:

  $$
  \gamma_{ic} = \frac{p_{c}^{old} \mathcal{N}(x_i; \mu_{c}^{old}, \Sigma_{c}^{old})}{\sum_{l=1}^{k} p_{l}^{old} \mathcal{N}(x_i; \mu_{l}^{old}, \Sigma_{l}^{old})}
  $$

  **M-step** re-estimate mixture parameters:

  $$
  \hat{p}_{c}^{new} = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ic},
  $$

  $$
  \hat{\mu}_{c}^{new} = \frac{1}{\sum_{i=1}^{N} \gamma_{ic}} \sum_{i=1}^{N} \gamma_{ic} x_i,
  $$

  $$
  \hat{\Sigma}_{c}^{new} = \frac{1}{\sum_{i=1}^{N} \gamma_{ic}} \sum_{i=1}^{N} \gamma_{ic} (x_i - \hat{\mu}_{c}^{new})(x_i - \hat{\mu}_{c}^{new})^T.
  $$
Colors represent $\gamma_{ic}$ after the E-step.
EM for Gaussian mixtures: an example

Colors represent $\gamma_{ic}$ after the E-step.

1st iteration 2nd iteration 3rd iteration
EM for Gaussian mixtures: an example

Colors represent $\gamma_{ic}$ after the E-step.

1st iteration
2nd iteration
3rd iteration
EM for Gaussian mixtures: an example

Colors represent $\gamma_{ic}$ after the E-step.

1st iteration 2nd iteration 3rd iteration
4th iteration 7th iteration 9th iteration
Colors represent $\gamma_{ic}$ after the E-step.
EM for Gaussian mixtures: an example

Colors represent $\gamma_{ic}$ after the E-step.
Colors represent $\gamma_{ic}$ after the E-step.
Problem with EM: Bad local minima

- We can be very unlucky with the initial guess.

1st iteration

2nd

4th

The problem:
$$\lim_{\sigma^2 \to 0} N(x; \mu = x, \Sigma = \sigma^2 I) = \infty.$$
Problem with EM: Bad local minima

- We can be very unlucky with the initial guess.

The problem:

$$\lim_{\sigma^2 \to 0} \mathcal{N}(x; \mu = x, \Sigma = \sigma^2 I) = \infty.$$
Problem with EM: Bad local minima

- We can be very unlucky with the initial guess.

1st iteration

2nd

4th

- The problem:

\[
\lim_{\sigma^2 \to 0} \mathcal{N}(x; \mu = x, \Sigma = \sigma^2 I) = \infty.
\]
Avoiding bad local minima

- Impose a prior on $\theta$: Instead of maximizing the likelihood in the M-step, maximize the regularized posterior:

$$\argmax_{\theta} \left\{ \mathbb{E}_Z \left[ \log p(X, Z; \theta) \right] + \log p(\theta) \right\}.$$
Avoiding bad local minima

- Impose a prior on $\theta$: Instead of maximizing the likelihood in the M-step, maximize the regularized posterior:

$$\argmax_{\theta} \{ \mathbb{E}_Z \left[ \log p(X, Z; \theta) \right] + \log p(\theta) \}.$$ 

- Restrict the form of the covariance matrices (diagonal or spherical).
Avoiding bad local minima

- Impose a prior on $\theta$: Instead of maximizing the likelihood in the M-step, maximize the regularized posterior:

  $\arg\max_{\theta} \{ \mathbb{E}_Z [\log p(X, Z; \theta)] + \log p(\theta) \}$.

- Restrict the form of the covariance matrices (diagonal or spherical).

- Start with a very large covariance estimate.
So far we have assumed known $k$.
Can we just select the $k$ that maximizes the likelihood?
So far we have assumed known $k$. Can we just select the $k$ that maximizes the likelihood?

- The result would be a mixture model with a separate, very narrow Gaussian component for every training example.
- This solution yields infinite log-likelihood!
So far we have assumed known $k$.

Can we just select the $k$ that maximizes the likelihood?

- The result would be a mixture model with a separate, very narrow Gaussian component for every training example.
- This solution yields infinite log-likelihood!

We need a criterion to penalize such models.

- Bayesian Information Criterion (BIC) score:

$$BIC(\mathcal{M}) = L^*(\mathcal{M}) - \frac{d(\mathcal{M})}{2} \log N,$$

where $L^*(\mathcal{M})$ is the likelihood of estimated model $\mathcal{M}$, $d(\mathcal{M})$ is the number of parameters in the model, and $N$ is the number of data points.
The EM algorithm in general

- Observed data $X$, hidden variables $Z$.
  - E.g., *missing data*.
- Complete data log-likelihood: $\ell(X, Z; \theta)$

Initialize $\theta^{old}$, and iterate until convergence:

**E-step:** Compute the expected likelihood as a function of $\theta$.

$$Q\left(\theta; \theta^{old}\right) = \mathbb{E}_{p(Z | X, \theta^{old})} \left[ \ell(X, Z; \theta) \mid X, \theta^{old} \right]$$

**M-step:** Compute

$$\theta^{new} = \arg\max_{\theta} Q\left(\theta; \theta^{old}\right).$$
Why does EM work?

Ultimately, we want to maximize likelihood of the \textit{observed} data

$$\theta^* = \arg\max_{\theta} \log p(X; \theta).$$

Let $\ell^{(t)}$ be $\log p(X; \theta_{\text{new}})$ after $t$ iterations.

Can show (not in this class):

$$\ell^{(0)} \leq \ell^{(1)} \leq \ldots \leq \ell^{(t)} \ldots$$