COMS 4771 Lecture 19

1. Mixture models
2. Expectation-Maximization
Mixture models
Unsupervised classification

▶ **Input:** \( x^{(1)}, x^{(2)}, \ldots, x^{(n)} \in \mathbb{R}^d \), target cardinality \( k \in \mathbb{N} \).

▶ **Output:** function \( f : \mathbb{R}^d \rightarrow \{1, 2, \ldots, k\} =: [k] \).

▶ **Typical semantics:** hidden subpopulation structure.
**Gaussian Mixture Model**

\[(X, Y) \sim P_\theta, \text{ a distribution over } \mathbb{R}^d \times [k] \text{ where} \]

\[Y \sim \pi \quad \text{(discrete distribution over } [k]; \Pr_\theta(Y = j) = \pi_j) \]

\[X | Y = j \sim N(\mu_j, \Sigma_j) \quad \text{(Gaussian with mean } \mu_j \text{ and covariance } \Sigma_j) \]

Parameters \( \theta := (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k) \).
Gaussian mixture model

\((X, Y) \sim P_\theta\), a distribution over \(\mathbb{R}^d \times [k]\) where

\[
Y \sim \pi \quad \text{(discrete distribution over \([k]\); \(Pr_\theta(Y = j) = \pi_j\))}
\]

\[
X|Y = j \sim N(\mu_j, \Sigma_j) \quad \text{(Gaussian with mean \(\mu_j\) and covariance \(\Sigma_j\))}
\]

Parameters \(\theta := (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)\).

Looks familiar?
Gaussian mixture model

$$(X, Y) \sim P_\theta, \text{ a distribution over } \mathbb{R}^d \times [k] \text{ where}$$

$$Y \sim \pi \quad \text{(discrete distribution over } [k]; \Pr_\theta(Y = j) = \pi_j)$$

$$X|Y = j \sim N(\mu_j, \Sigma_j) \quad \text{(Gaussian with mean } \mu_j \text{ and covariance } \Sigma_j)$$

Parameters $\theta := (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$.

Looks familiar?

Modeling assumption:
Data $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times [k]$ is iid sample from $P$, 
\((X, Y) \sim P_\theta, \text{ a distribution over } \mathbb{R}^d \times [k] \text{ where}\)

\[Y \sim \pi \quad (\text{discrete distribution over } [k]; \Pr_\theta(Y = j) = \pi_j)\]

\[X \mid Y = j \sim N(\mu_j, \Sigma_j) \quad (\text{Gaussian with mean } \mu_j \text{ and covariance } \Sigma_j)\]

Parameters \(\theta := (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)\).

**Looks familiar?**

**Modeling assumption:**
Data \((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times [k]\) is iid sample from \(P\), but you only get \(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\).
**Gaussian mixture model**

\[(X, Y) \sim P_\theta, \text{ a distribution over } \mathbb{R}^d \times [k] \text{ where}\]

\[Y \sim \pi \quad \text{(discrete distribution over } [k]; \Pr_\theta(Y = j) = \pi_j)\]

\[X | Y = j \sim N(\mu_j, \Sigma_j) \quad \text{(Gaussian with mean } \mu_j \text{ and covariance } \Sigma_j)\]

Parameters \(\theta := (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)\).

**Looks familiar?**

**Modeling assumption:**
Data \((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times [k]\) is iid sample from \(P\), but you only get \(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\).

Models of this sort are called **mixture models**; this one in particular is called the **Gaussian mixture model**.

\[X \sim \sum_{j=1}^{k} \pi_j \ N(\mu_j, \Sigma_j)\]

**Mixing weights** \(\pi\); **mixture components** \(N(\mu_1, \Sigma_1), \ldots, N(\mu_k, \Sigma_k)\).
Gaussian mixture model

\((X, Y) \sim P_\theta\), a distribution over \(\mathbb{R}^d \times [k]\) where

\[ Y \sim \pi \quad \text{(discrete distribution over [k]; Pr_\theta(Y = j) = \pi_j)} \]
\[ X \vert Y = j \sim N(\mu_j, \Sigma_j) \quad \text{(Gaussian with mean \(\mu_j\) and covariance \(\Sigma_j\))} \]

Parameters \(\theta := (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)\).

Looks familiar?

Modeling assumption:
Data \((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times [k]\) is iid sample from \(P\), but you only get \(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\).

Models of this sort are called mixture models; this one in particular is called the Gaussian mixture model.

\[ p_\theta(x) = \sum_{j=1}^{k} \pi_j \cdot (2\pi)^{-d/2} \sqrt{\det(\Sigma_j^{-1})} \exp\left(-\frac{1}{2}(x - \mu_j) \top \Sigma_j^{-1}(x - \mu_j)\right) \]

Mixing weights \(\pi\); mixture components \(N(\mu_1, \Sigma_1), \ldots, N(\mu_k, \Sigma_k)\).
Gaussian mixtures in $\mathbb{R}^1$

\[
\frac{1}{2} \mathcal{N}(0, 1) + \frac{1}{2} \mathcal{N}(2, 1/4)
\]

\[
\frac{4}{5} \mathcal{N}(0, 1) + \frac{1}{5} \mathcal{N}(2, 1/4)
\]
Gaussian mixtures in $\mathbb{R}^2$

Plot of the mixture density.

A sample of size 1000.
Suppose you have the parameters $\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$ of a Gaussian mixture distribution, and further that $(X, Y) \sim P_\theta$. 
Suppose you have the parameters \( \theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k) \) of a Gaussian mixture distribution, and further that \((X, Y) \sim P_\theta\).

Assignment variables \( \Phi = (\Phi_1, \Phi_2, \ldots, \Phi_k) \in \{0, 1\}^k \) (as in \(k\)-means):

\[
\Phi_j := 1\{Y = j\}.
\]

You observe \( X = x \), but \( Y \) (and hence \( \Phi \)) is hidden from you!
Suppose you have the parameters $\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$ of a Gaussian mixture distribution, and further that $(X, Y) \sim P_\theta$.

Assignment variables $\Phi = (\Phi_1, \Phi_2, \ldots, \Phi_k) \in \{0, 1\}^k$ (as in $k$-means):

$$\Phi_j := 1\{Y = j\}.$$ 

You observe $X = x$, but $Y$ (and hence $\Phi$) is hidden from you!

**Soft assignment** of a data point $x \in \mathbb{R}^d$ to component $j \in [k]$:

$$\mathbb{E}_\theta[\Phi_j \mid X = x] = \Pr_\theta[Y = j \mid X = x]$$
Suppose you have the parameters $\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$ of a Gaussian mixture distribution, and further that $(X, Y) \sim P_\theta$.

Assignment variables $\Phi = (\Phi_1, \Phi_2, \ldots, \Phi_k) \in \{0, 1\}^k$ (as in $k$-means):

$$\Phi_j := \mathbb{1}\{Y = j\}.$$ 

You observe $X = x$, but $Y$ (and hence $\Phi$) is hidden from you!

**Soft assignment** of a data point $x \in \mathbb{R}^d$ to component $j \in [k]$:

$$\mathbb{E}_\theta[\Phi_j \mid X = x] = \Pr_\theta[Y = j \mid X = x]$$
$$= \frac{\Pr_\theta[Y = j] \cdot \Pr_\theta[X = x \mid Y = j]}{\Pr_\theta[X = x]}$$
Suppose you have the parameters $\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$ of a Gaussian mixture distribution, and further that $(X, Y) \sim P_\theta$.

Assignment variables $\Phi = (\Phi_1, \Phi_2, \ldots, \Phi_k) \in \{0, 1\}^k$ (as in $k$-means):

$$\Phi_j := 1\{Y = j\}.$$  

You observe $X = x$, but $Y$ (and hence $\Phi$) is hidden from you!

**Soft assignment** of a data point $x \in \mathbb{R}^d$ to component $j \in [k]$:

$$\mathbb{E}_\theta[\Phi_j \mid X = x] = \Pr_\theta[Y = j \mid X = x] = \frac{\Pr_\theta[Y = j] \cdot \Pr_\theta[X = x \mid Y = j]}{\Pr_\theta[X = x]} = \frac{\pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp\left(-\frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j)\right)}{\sum_{j' = 1}^k \pi_{j'} \cdot \sqrt{\det(\Sigma_{j'}^{-1})} \exp\left(-\frac{1}{2}(x - \mu_{j'})^\top \Sigma_{j'}^{-1}(x - \mu_{j'})\right)}.$$
**Example:** a Gaussian mixture with $k = 2$ in $\mathbb{R}^1$.

$$
\Pr_\theta[X = x \mid Y = j], \, j \in \{1, 2\}.
$$

$\Pr_\theta[Y = 1 \mid X = x] = \frac{\pi_1 \cdot \frac{1}{\sigma_1} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right)}{\pi_1 \cdot \frac{1}{\sigma_1} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right) + \pi_2 \cdot \frac{1}{\sigma_2} \exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)}.$
Maximum likelihood estimation of $\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$ given data $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ (assumed to be an i.i.d. sample).

$$\theta_{\text{ML}} := \arg \max_{\theta} \sum_{i=1}^{n} \ln p_{\theta}(x^{(i)})$$
Maximum likelihood estimation of $\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$ given data $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ (assumed to be an i.i.d. sample).

$$\theta_{ML} := \arg \max_\theta \sum_{i=1}^n \ln p_\theta(x^{(i)})$$

$$= \arg \max_\theta \sum_{i=1}^n \ln \left\{ \sum_{j=1}^k \pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp \left( -\frac{1}{2} (x - \mu_j)^\top \Sigma_j^{-1} (x - \mu_j) \right) \right\}$$
Parameter estimation for Gaussian mixtures

Maximum likelihood estimation of \( \theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k) \) given data \( x^{(1)}, x^{(2)}, \ldots, x^{(n)} \) (assumed to be an i.i.d. sample).

\[
\theta_{ML} := \arg \max_{\theta} \sum_{i=1}^{n} \ln p_\theta(x^{(i)})
\]

\[
= \arg \max_{\theta} \sum_{i=1}^{n} \ln \left\{ \sum_{j=1}^{k} \pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp \left( -\frac{1}{2} (x - \mu_j)^\top \Sigma_j^{-1} (x - \mu_j) \right) \right\}
\]

Ack! \( \ln \left\{ \sum_{j=1}^{k} \cdots \right\} \) does not simplify nicely!
MLE for Gaussian mixtures: not a convex optimization problem.

\[
\text{arg max } \sum_{i=1}^{n} \ln \left\{ \sum_{j=1}^{k} \pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp \left( -\frac{1}{2} (x - \mu_j)^\top \Sigma_j^{-1} (x - \mu_j) \right) \right\}
\]
MLE for Gaussian mixtures: **not a convex optimization problem.**

\[
\arg\max_{\theta} \sum_{i=1}^{n} \ln \left\{ \sum_{j=1}^{k} \pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp \left( -\frac{1}{2} (\mathbf{x} - \mu_j)^\top \Sigma_j^{-1} (\mathbf{x} - \mu_j) \right) \right\}
\]

Gradient descent (ascent) may converge to a *local maximizer*, but could be arbitrarily far from / worse than the MLE.
MLE for Gaussian mixtures: not a convex optimization problem.

\[
\arg \max_{\theta} \sum_{i=1}^{n} \ln \left\{ \sum_{j=1}^{k} \pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp \left( -\frac{1}{2} (x - \mu_j)^\top \Sigma_j^{-1} (x - \mu_j) \right) \right\}
\]

Gradient descent (ascent) may converge to a local maximizer, but could be arbitrarily far from / worse than the MLE.

But this is a good thing, because the MLE is degenerate. 
\[\mu_1 = x^{(1)}, \sigma_1^2 \to 0, \text{ likelihood } \to \infty.\]
Saving grace:
If the data are actually generated by a Gaussian mixture with parameters $\theta_*$, then $\theta_*$ may be close to some local maximizer of the log-likelihood.
Local optimization

Saving grace:
If the data are actually generated by a Gaussian mixture with parameters $\theta^\star$, then $\theta^\star$ may be close to some local maximizer of the log-likelihood.

Just need to find the “right” local maximizer ... (i.e., a good, non-degenerate local maximizer).
Local optimization

Saving grace:
If the data are actually generated by a Gaussian mixture with parameters $\theta_\star$, then $\theta_\star$ may be close to some local maximizer of the log-likelihood.

Just need to find the “right” local maximizer . . .
(i.e., a good, non-degenerate local maximizer).

Methods like gradient ascent would work, but there’s a much easier & better local optimization method for this case: the E-M algorithm.
Expectation-Maximization
Motivation

Suppose we had a labeled iid sample:

\[(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times [k].\]
Suppose we had a labeled iid sample:

\[(x^{(1)}, \phi^{(1)}), (x^{(2)}, \phi^{(2)}), \ldots, (x^{(n)}, \phi^{(n)}) \in \mathbb{R}^d \times \{0, 1\}^k.\]
Suppose we had a labeled iid sample:

\[(x^{(1)}, \phi^{(1)}), (x^{(2)}, \phi^{(2)}), \ldots, (x^{(n)}, \phi^{(n)}) \in \mathbb{R}^d \times \{0, 1\}^k.\]

The “complete log-likelihood” of \(\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)\) is

\[
\sum_{i=1}^{n} \sum_{j=1}^{k} \phi^{(i)}_j \ln \left\{ \pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp \left( -\frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j) \right) \right\}
\]

\[
= \sum_{i=1}^{n} \sum_{j=1}^{k} \phi^{(i)}_j \left( \ln \pi_j + \frac{1}{2} \ln \det(\Sigma_j^{-1}) - \frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j) \right),
\]

which can be easily maximized w.r.t. \(\theta\).
Suppose we had a *labeled* iid sample:

\[(x^{(1)}, \phi^{(1)}), (x^{(2)}, \phi^{(2)}), \ldots, (x^{(n)}, \phi^{(n)}) \in \mathbb{R}^d \times \{0, 1\}^k.\]

The “complete log-likelihood” of \(\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)\) is

\[
\sum_{i=1}^{n} \sum_{j=1}^{k} \phi^{(i)}_j \ln\left\{ \pi_j \cdot \sqrt{\det(\Sigma^{-1}_j)} \exp\left( -\frac{1}{2} (x - \mu_j)^\top \Sigma^{-1}_j (x - \mu_j) \right) \right\}
\]

\[
= \sum_{i=1}^{n} \sum_{j=1}^{k} \phi^{(i)}_j \left( \ln \pi_j + \frac{1}{2} \ln \det(\Sigma^{-1}_j) - \frac{1}{2} (x - \mu_j)^\top \Sigma^{-1}_j (x - \mu_j) \right),
\]

which can be easily maximized w.r.t. \(\theta\).

In fact, even easy with *soft assignments* \(w^{(i)}_j := \mathbb{E}_{\hat{\theta}}[\phi^{(i)}_j | X = x^{(i)}]:\)

\[
\sum_{i=1}^{n} \sum_{j=1}^{k} \mathbb{E}_{\hat{\theta}}[\Phi^{(i)}_j | X = x^{(i)}] \left( \ln \pi_j + \frac{1}{2} \ln \det(\Sigma^{-1}_j) - \frac{1}{2} (x - \mu_j)^\top \Sigma^{-1}_j (x - \mu_j) \right).
\]

“Expectation (w.r.t. \(P_{\hat{\theta}}\) conditioned on \(\{x^{(i)}\}\)) of complete log-likelihood.”
Motivation

Suppose we had a labeled iid sample:

\[(x^{(1)}, \phi^{(1)}), (x^{(2)}, \phi^{(2)}), \ldots, (x^{(n)}, \phi^{(n)}) \in \mathbb{R}^d \times \{0, 1\}^k.\]

The “complete log-likelihood” of \(\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)\) is

\[\sum_{i=1}^{n} \sum_{j=1}^{k} \phi^{(i)}_j \ln \left\{ \pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp \left( -\frac{1}{2} (x - \mu_j)^\top \Sigma_j^{-1} (x - \mu_j) \right) \right\}\]

\[= \sum_{i=1}^{n} \sum_{j=1}^{k} \phi^{(i)}_j \left( \ln \pi_j + \frac{1}{2} \ln \det(\Sigma_j^{-1}) - \frac{1}{2} (x - \mu_j)^\top \Sigma_j^{-1} (x - \mu_j) \right),\]

which can be easily maximized w.r.t. \(\theta\).

In fact, even easy with soft assignments \(w^{(i)}_j := \mathbb{E}_{\hat{\theta}}[\phi^{(i)}_j | X = x^{(i)}]:\)

\[\sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i)}_j \left( \ln \pi_j + \frac{1}{2} \ln \det(\Sigma_j^{-1}) - \frac{1}{2} (x - \mu_j)^\top \Sigma_j^{-1} (x - \mu_j) \right).\]

“Expectation (w.r.t. \(P_{\hat{\theta}}\) conditioned on \(\{x^{(i)}\}\)) of complete log-likelihood.”
Expectation-Maximization (E-M)

Initialize $\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$ somehow.
Initialize $\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$ somehow. Then repeat:

1. **E step**: expectation of “hidden variables” w.r.t. $P_\theta$ conditioned on data. For each $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, k\}$,

$$w_{j}^{(i)} := \frac{\pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp\left(-\frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j)\right)}{\sum_{j' = 1}^{k} \pi_{j'} \cdot \sqrt{\det(\Sigma_{j'}^{-1})} \exp\left(-\frac{1}{2}(x - \mu_{j'})^\top \Sigma_{j'}^{-1}(x - \mu_{j'})\right)}$$
**Expectation-Maximization (E-M)**

Initialize $\theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_k, \mu_k, \Sigma_k)$ somehow. Then repeat:

1. **E step**: expectation of “hidden variables” w.r.t. $P_\theta$ conditioned on data. For each $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, k\}$,

   $$w_j^{(i)} := \frac{\pi_j \cdot \sqrt{\det(\Sigma_j^{-1})} \exp\left(-\frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j)\right)}{\sum_{j' = 1}^k \pi_{j'} \cdot \sqrt{\det(\Sigma_{j'}^{-1})} \exp\left(-\frac{1}{2}(x - \mu_{j'})^\top \Sigma_{j'}^{-1}(x - \mu_{j'})\right)}$$

2. **M step**: maximize “expected complete log-likelihood” w.r.t. parameters. For each $j \in \{1, 2, \ldots, k\}$,

   $$\pi_j := \frac{1}{n} \sum_{i=1}^n w_j^{(i)}$$
   $$\mu_j := \frac{1}{n \pi_j} \sum_{i=1}^n w_j^{(i)} x^{(i)}$$
   $$\Sigma_j := \frac{1}{n \pi_j} \sum_{i=1}^n w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top.$$
Sample run of the E-M algorithm

Arbitrary initialization of $\pi_j$, $\mu_j$, $\Sigma_j$ for $j \in \{1, 2\}$. 
Sample run of the E-M algorithm

**E step**: soft assignments $z^{(i)}_j$ for each $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2\}$. 
Sample run of the E-M algorithm

M step: update parameters $\pi_j$, $\mu_j$, $\Sigma_j$ for $j \in \{1, 2\}$.
Sample run of the E-M algorithm

After two rounds of E-M.
Sample run of the E-M algorithm

After five rounds of E-M.
Sample run of the E-M algorithm

After 20 rounds of E-M.
**Using the E-M algorithm**

E-M for Gaussian mixtures

1. **E step**: For each \( i \in [n], j \in [k] \),

\[
w_j^{(i)} \propto \pi_j \cdot p_{\mu_j, \Sigma_j}(x^{(i)})
\]

where \( p_{\mu, \Sigma} \) is the \( N(\mu, \Sigma) \) p.d.f.

2. **M step**: For each \( j \in [k] \),

\[
\begin{align*}
\pi_j &:= \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)} \\
\mu_j &:= \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} x^{(i)} \\
\Sigma_j &:= \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top.
\end{align*}
\]
Using the E-M algorithm

E-M for Gaussian mixtures

1. **E step**: For each \( i \in [n], j \in [k] \),

\[
    w_j^{(i)} \propto \pi_j \cdot p_{\mu_j, \Sigma_j}(x^{(i)})
\]

where \( p_{\mu, \Sigma} \) is the \( N(\mu, \Sigma) \) p.d.f.

2. **M step**: For each \( j \in [k] \),

\[
    \pi_j := \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)}
\]

\[
    \mu_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} x^{(i)}
\]

\[
    \Sigma_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top.
\]

Some details

- **Initialization**: a bit of an art; both \( D^2 \)-sampling and Lloyd’s algorithm are reasonable.

- Starved clusters: problems can occur if \( \pi_j \) becomes too small (e.g., \( \Sigma_j \) could be near singular).

- Remove/replace such components.

- Convergence: E-M is guaranteed to converge to a stationary point (i.e., gradient equals zero).

Run E-M from many random initializations; pick the result with highest likelihood.
Using the E-M algorithm

E-M for Gaussian mixtures

1. **E step**: For each \( i \in [n], j \in [k], \)
   \[
   w_j^{(i)} \propto \pi_j \cdot p_{\mu_j, \Sigma_j}(x^{(i)})
   \]
   where \( p_{\mu, \Sigma} \) is the \( \mathcal{N}(\mu, \Sigma) \) p.d.f.

2. **M step**: For each \( j \in [k], \)
   \[
   \pi_j := \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)}
   \]
   \[
   \mu_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} x^{(i)}
   \]
   \[
   \Sigma_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top.
   \]

Some details

- **Initialization**: a bit of an art; both \( D^2 \)-sampling and Lloyd’s algorithm are reasonable.
- **Starved clusters**: problems can occur if \( \pi_j \) becomes too small (e.g., \( \Sigma_j \) could be near singular).
Using the E-M algorithm

E-M for Gaussian mixtures

1. E step: For each $i \in [n], j \in [k]$,
   \[ w_j^{(i)} \propto \pi_j \cdot p_{\mu_j, \Sigma_j}(x^{(i)}) \]
   where $p_{\mu, \Sigma}$ is the $N(\mu, \Sigma)$ p.d.f.

2. M step: For each $j \in [k]$,
   \[ \pi_j := \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)} \]
   \[ \mu_j := \frac{1}{n \pi_j} \sum_{i=1}^{n} w_j^{(i)} x^{(i)} \]
   \[ \Sigma_j := \frac{1}{n \pi_j} \sum_{i=1}^{n} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top. \]

Some details

- **Initialization**: a bit of an art; both $D^2$-sampling and Lloyd’s algorithm are reasonable.

- **Starved clusters**: problems can occur if $\pi_j$ becomes too small (e.g., $\Sigma_j$ could be near singular).

Remove/replace such components.
E-M for Gaussian mixtures

1. **E step**: For each $i \in [n]$, $j \in [k]$,

   $$w_j^{(i)} \propto \pi_j \cdot p_{\mu_j, \Sigma_j}(x^{(i)})$$

   where $p_{\mu, \Sigma}$ is the $N(\mu, \Sigma)$ p.d.f.

2. **M step**: For each $j \in [k]$,

   $$\pi_j := \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)}$$

   $$\mu_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} x^{(i)}$$

   $$\Sigma_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top.$$

**Some details**

- **Initialization**: a bit of an art; both $D^2$-sampling and Lloyd’s algorithm are reasonable.

- **Starved clusters**: problems can occur if $\pi_j$ becomes too small (e.g., $\Sigma_j$ could be near singular). Remove/replace such components.

- **Convergence**: E-M is guaranteed to converge to a stationary point (i.e., gradient equals zero).
Using the E-M algorithm

E-M for Gaussian mixtures

1. **E step**: For each \( i \in [n], j \in [k], \)

\[
w_j^{(i)} \propto \pi_j \cdot p_{\mu_j, \Sigma_j}(x^{(i)})
\]

where \( p_{\mu, \Sigma} \) is the \( \text{N}(\mu, \Sigma) \) p.d.f.

2. **M step**: For each \( j \in [k], \)

\[
\pi_j := \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)}
\]

\[
\mu_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} x^{(i)}
\]

\[
\Sigma_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^\top.
\]

Some details

- **Initialization**: a bit of an art; both \( D^2 \)-sampling and Lloyd’s algorithm are reasonable.

- **Starved clusters**: problems can occur if \( \pi_j \) becomes too small (e.g., \( \Sigma_j \) could be near singular).

  Remove/replace such components.

- **Convergence**: E-M is guaranteed to converge to a stationary point (i.e., gradient equals zero).

  Run E-M from many random initializations; pick the result with highest likelihood.
Derivation of E-M algorithm

E-M is a general algorithmic template for climbing log-likelihood objectives of models with hidden variables (e.g., cluster assignments).
Derivation of E-M algorithm

E-M is a general algorithmic template for climbing log-likelihood objectives of models with hidden variables (e.g., cluster assignments).

Model gives probability of both observed and unobserved data (i.e., a “complete likelihood”)
e.g., for Gaussian mixtures,

\[
\Pr_\theta(X = x \land Y = j) = \pi_j \cdot p_{\mu_j, \Sigma_j}(x)
\]

(\(X\) is observed, but \(Y\) is hidden).
Derivation of E-M algorithm

E-M is a general algorithmic template for climbing log-likelihood objectives of models with hidden variables (e.g., cluster assignments).

Model gives probability of both observed and unobserved data (i.e., a “complete likelihood”)
e.g., for Gaussian mixtures,

\[
\Pr_\theta (X = x \land Y = j) = \pi_j \cdot p_{\mu_j, \Sigma_j}(x)
\]

\(X\) is observed, but \(Y\) is hidden).

Likelihood of \(\theta\) given \(X = x\) is

\[
\Pr_\theta (X = x) = \sum_{j=1}^{k} \Pr_\theta (X = x \land Y = j) = \sum_{j=1}^{k} \pi_j \cdot p_{\mu_j, \Sigma_j}(x).
\]
For now, just consider one data point $x^{(i)}$.
Log-likelihood of $\theta$ given $x^{(i)}$ is (for any $q_j \neq 0$)

$$
\ln \Pr_\theta(X = x^{(i)}) = \ln \left( \sum_{j=1}^{k} q_j \cdot \Pr_\theta(X = x^{(i)} \wedge Y = j) \right)
$$
Derivation of E-M algorithm

For now, just consider one data point $x^{(i)}$.
Log-likelihood of $\theta$ given $x^{(i)}$ is (for any $q_j \neq 0$)

\[
\ln \Pr_\theta(\mathbf{X} = x^{(i)}) = \ln \left( \sum_{j=1}^{k} \Pr_\theta(\mathbf{X} = x^{(i)} \land Y = j) \right) \\
= \ln \left( \sum_{j=1}^{k} q_j \cdot \frac{\Pr_\theta(\mathbf{X} = x^{(i)} \land Y = j)}{q_j} \right).
\]
**Derivation of E-M algorithm**

For now, just consider one data point $x^{(i)}$.

Log-likelihood of $\theta$ given $x^{(i)}$ is (for any $q_j \neq 0$)

$$
\ln \Pr_\theta(X = x^{(i)}) = \ln \left( \sum_{j=1}^{k} \Pr_\theta(X = x^{(i)} \land Y = j) \right)
= \ln \left( \sum_{j=1}^{k} q_j \cdot \frac{\Pr_\theta(X = x^{(i)} \land Y = j)}{q_j} \right).
$$

By Jensen’s inequality and concavity of $\ln$, if $q = (q_1, q_2, \ldots, q_k)$ is a probability distribution (with $q_j > 0$), then

$$
\ln \left( \sum_{j=1}^{k} q_j \cdot \frac{\Pr_\theta(X = x^{(i)} \land Y = j)}{q_j} \right) \geq \sum_{j=1}^{k} q_j \cdot \ln \left( \frac{\Pr_\theta(X = x^{(i)} \land Y = j)}{q_j} \right).
$$
Derivation of E-M algorithm

Now consider all \( n \) data points \( \{x^{(i)}\}_{i \in [n]} \). Assuming each \( q^{(i)} \) is a probability vector with \( q^{(i)}_j > 0 \),

\[
\mathcal{L}(\theta) \geq \sum_{i=1}^{n} \sum_{j=1}^{k} q^{(i)}_j \cdot \ln \left( \frac{\Pr_{\theta}(X = x^{(i)} \land Y = j)}{q^{(i)}_j} \right) =: \mathcal{L}_{\text{LB}}(\theta). \quad (*)
\]

Bayes’ rule shows that \( \mathcal{L}(\hat{\theta}) = \mathcal{L}_{\text{LB}}(\hat{\theta}) \) when \( q^{(i)}_j = \Pr_{\hat{\theta}}(Y = j \mid X = x^{(i)}) \).
Now consider all $n$ data points $\{x^{(i)}\}_{i \in [n]}$. Assuming each $q^{(i)}$ is a probability vector with $q_j^{(i)} > 0$, 

$$ \mathcal{L}(\theta) \geq \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \ln \left( \frac{\Pr_{\theta}(X = x^{(i)} \land Y = j)}{q_j^{(i)}} \right) =: \mathcal{L}_{LB}(\theta). \quad (\star) $$

Bayes’ rule shows that $\mathcal{L}(\hat{\theta}) = \mathcal{L}_{LB}(\hat{\theta})$ when $q_j^{(i)} = \Pr_{\hat{\theta}}(Y = j \mid X = x^{(i)})$.

**E-M algorithm**: starting with some initial setting of $\hat{\theta}$, repeat the following.

- **E step**: Construct log-likelihood lower-bound $\mathcal{L}_{LB}$ as in $(\star)$ by choosing 

  $$ q_j^{(i)} := \Pr_{\hat{\theta}}(Y = j \mid X = x^{(i)}) $$

  so that lower-bound is tight at current $\hat{\theta}$. 

---

19 / 26
Derivation of E-M algorithm

Now consider all $n$ data points $\{x^{(i)}\}_{i \in [n]}$. Assuming each $q^{(i)}$ is a probability vector with $q_j^{(i)} > 0$,

$$L(\theta) \geq \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \ln \left( \frac{\Pr_\theta(X = x^{(i)} \land Y = j)}{q_j^{(i)}} \right) =: L_{LB}(\theta). \quad (\star)$$

Bayes’ rule shows that $L(\hat{\theta}) = L_{LB}(\hat{\theta})$ when $q_j^{(i)} = \Pr_{\hat{\theta}}(Y = j \mid X = x^{(i)}).

**E-M algorithm**: starting with some initial setting of $\hat{\theta}$, repeat the following.

- **E step**: Construct log-likelihood lower-bound $L_{LB}$ as in $(\star)$ by choosing
  $$q_j^{(i)} := \Pr_{\hat{\theta}}(Y = j \mid X = x^{(i)})$$
  so that lower-bound is tight at current $\hat{\theta}$.

- **M step**: With the $q_j^{(i)}$ fixed, update $\hat{\theta}$ to maximize $L_{LB}$:
  $$L_{LB}(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \ln \Pr_\theta(X = x^{(i)} \land Y = j) - \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \ln q_j^{(i)}$$
Derivation of E-M algorithm

Now consider all \( n \) data points \( \{x^{(i)}\}_{i \in [n]} \). Assuming each \( q^{(i)} \) is a probability vector with \( q_j^{(i)} > 0 \),

\[
\mathcal{L}(\theta) \geq \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \ln \left( \frac{\Pr_\theta(X = x^{(i)} \land Y = j)}{q_j^{(i)}} \right) =: \mathcal{L}_{LB}(\theta). \quad (\star)
\]

Bayes’ rule shows that \( \mathcal{L}(\hat{\theta}) = \mathcal{L}_{LB}(\hat{\theta}) \) when \( q_j^{(i)} = \Pr_{\hat{\theta}}(Y = j \mid X = x^{(i)}) \).

E-M algorithm: starting with some initial setting of \( \hat{\theta} \), repeat the following.

- **E step**: Construct log-likelihood lower-bound \( \mathcal{L}_{LB} \) as in \((\star)\) by choosing

\[
q_j^{(i)} := \Pr_{\hat{\theta}}(Y = j \mid X = x^{(i)})
\]

so that lower-bound is tight at current \( \hat{\theta} \).

- **M step**: With the \( q_j^{(i)} \) fixed, update \( \hat{\theta} \) to maximize \( \mathcal{L}_{LB} \):

\[
\mathcal{L}_{LB}(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \ln \Pr_\theta(X = x^{(i)} \land Y = j) - \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \ln q_j^{(i)}
\]

\[
= \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \left( \ln \pi_j + \ln p_{\mu_j, \Sigma_j}(x^{(i)}) \right) + \text{const.}
\]
CONSTRUCTING AND MAXIMIZING $\mathcal{L}_{LB}$

\[ \mathcal{L}(\theta) \]

\[ \mathcal{L}_{LB}(\theta) \]

E step: construct $\mathcal{L}_{LB}$ such that $\mathcal{L}(\hat{\theta}) = \mathcal{L}_{LB}(\hat{\theta})$ for current $\hat{\theta}$. 

$\theta$
Constructing and maximizing $\mathcal{L}_{LB}$

M step: choose $\hat{\theta}$ to maximize $\mathcal{L}_{LB}$.
Constructing and maximizing $\mathcal{L}_{LB}$

**E step:** construct $\mathcal{L}_{LB}$ such that $\mathcal{L}(\hat{\theta}) = \mathcal{L}_{LB}(\hat{\theta})$ for current $\hat{\theta}$. 

\[
\mathcal{L}(\theta) \\
\mathcal{L}_{LB}(\theta)
\]
Constructing and maximizing $\mathcal{L}_{LB}$

M step: choose $\hat{\theta}$ to maximize $\mathcal{L}_{LB}$. 

\[ \mathcal{L}(\theta) \]

\[ \mathcal{L}_{LB}(\theta) \]
Other hidden variable models

Fairly easy to derive E-M algorithm for other hidden variable models by following general template.

Simple Mechanical Turk (MTurk) model:
- $m$ items, $n$ workers.
- Nature picks correct label for item $i$ to be 1 with probability $\pi_i$ (and 0 otherwise).
- Ask each worker to label each item as 0 or 1.
- Worker $j$ responds with correct label on item $i$ with probability $p_j$.
- All choices of Nature and worker responses are independent.

Parameters are $\theta = (\pi_1, \pi_2, \ldots, \pi_m, p_1, p_2, \ldots, p_n)$.

Random variables:
- (Hidden) $Y_i$ is the correct label for item $i$; $\Pr_\theta(Y_i = 1) = \pi_i$.
- (Observed) $X_{i,j}$ is the response given by worker $j$ for item $i$; $\Pr_\theta(X_{i,j} = Y_i) = p_j$. 
Other hidden variable models

Fairly easy to derive E-M algorithm for other hidden variable models by following general template.

**Simple Mechanical Turk (MTurk) model:** $m$ items, $n$ workers.

- Nature picks correct label for item $i$ to be 1 with probability $\pi_i$ (and 0 otherwise).
- Ask each worker to label each item as 0 or 1.
- Worker $j$ responds with correct label on item $i$ with probability $p_j$.
- All choices of Nature and worker responses are independent.
Other hidden variable models

Fairly easy to derive E-M algorithm for other hidden variable models by following general template.

Simple Mechanical Turk (MTurk) model: $m$ items, $n$ workers.

- Nature picks correct label for item $i$ to be 1 with probability $\pi_i$ (and 0 otherwise).
- Ask each worker to label each item as 0 or 1.
- Worker $j$ responds with correct label on item $i$ with probability $p_j$.
- All choices of Nature and worker responses are independent.

Parameters are $\theta = (\pi, p) = (\pi_1, \pi_2, \ldots, \pi_m, p_1, p_2, \ldots, p_n)$.

Random variables:

- (Hidden) $Y_i$ is the correct label for item $i$;
  \[ \Pr_{\theta}(Y_i = 1) = \pi_i. \]

- (Observed) $X_{i,j}$ is the response given by worker $j$ for item $i$;
  \[ \Pr_{\theta}(X_{i,j} = Y_i) = p_j. \]
For now, pretend there’s only one item $i$; $X_i := (X_{i,1}, X_{i,2}, \ldots, X_{i,n})$ and $Y_i$. Let $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n}) \in \{0, 1\}^n$ be the observed responses.

$$\ln \Pr_\theta(X_i = x_i)$$
For now, pretend there’s only one item $i$; $X_i := (X_{i,1}, X_{i,2}, \ldots, X_{i,n})$ and $Y_i$. Let $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n}) \in \{0, 1\}^n$ be the observed responses.

$$
\ln \Pr_{\theta}(X_i = x_i) = \ln \sum_{y \in \{0, 1\}} q(y) \cdot \frac{\Pr_{\theta}(X_i = x_i \land Y_i = y)}{q(y)}
$$
Log-likelihood for MTurk

For now, pretend there’s only one item $i$; $X_i := (X_{i,1}, X_{i,2}, \ldots, X_{i,n})$ and $Y_i$.

Let $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n}) \in \{0, 1\}^n$ be the observed responses.

\[
\ln \Pr_\theta(X_i = x_i) = \ln \sum_{y \in \{0, 1\}} q(y) \cdot \frac{\Pr_\theta(X_i = x_i \land Y_i = y)}{q(y)} \geq \sum_{y \in \{0, 1\}} q(y) \cdot \ln \Pr_\theta(X_i = x_i \land Y_i = y) - \sum_{y \in \{0, 1\}} q(y) \ln q(y).
\]
Log-likelihood for MTurk

For now, pretend there's only one item $i$; $X_i := (X_{i,1}, X_{i,2}, \ldots, X_{i,n})$ and $Y_i$.

Let $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n}) \in \{0, 1\}^n$ be the observed responses.

$$
\ln \Pr_{\theta}(X_i = x_i) \\
= \ln \sum_{y \in \{0, 1\}} q(y) \cdot \frac{\Pr_{\theta}(X_i = x_i \land Y_i = y)}{q(y)} \\
\geq \sum_{y \in \{0, 1\}} q(y) \cdot \ln \Pr_{\theta}(X_i = x_i \land Y_i = y) - \sum_{y \in \{0, 1\}} q(y) \ln q(y).
$$

For each $y \in \{0, 1\}$, "complete log-likelihood" is

$$
\ln \Pr_{\theta}(X_i = x_i \land Y_i = y) \\
= (1 - y) \left[ \ln(1 - \pi_i) + \sum_{j=1}^{n} (1 - x_{i,j}) \ln p_j + x_{i,j} \ln(1 - p_j) \right] \\
+ y \left[ \ln \pi_i + \sum_{j=1}^{n} x_{i,j} \ln p_j + (1 - x_{i,j}) \ln(1 - p_j) \right].
$$
By independence and Bayes’ rule:

\[
\Pr_{\hat{\theta}}(Y_i = y \mid X_i = x_i) =: q_i^y (1 - q_i)^{1-y}
\]

where

\[
q_i := \Pr_{\hat{\theta}}(Y_i = 1 \mid X_i = x_i)
\]

\[
= \frac{\hat{\pi}_i \prod_{j=1}^{n} \hat{p}_j^{x_{i,j}} (1 - \hat{p}_j)^{1-x_{i,j}}}{\hat{\pi}_i \prod_{j=1}^{n} \hat{p}_j^{x_{i,j}} (1 - \hat{p}_j)^{1-x_{i,j}} + (1 - \hat{\pi}_i) \prod_{j=1}^{n} \hat{p}_j^{1-x_{i,j}} (1 - \hat{p}_j)^{x_{i,j}}}.
\]
Log-likelihood (lower-bound) for MTurk

By independence and Bayes’ rule:

\[
\Pr_\hat{\theta}(Y_i = y \mid X_i = x_i) =: q_i^y (1 - q_i)^{1-y}
\]

where

\[
q_i := \Pr_\hat{\theta}(Y_i = 1 \mid X_i = x_i) = \frac{\hat{\pi}_i \prod_{j=1}^{n} \hat{p}_{j}^{x_{i,j}} (1 - \hat{p}_{j})^{1-x_{i,j}}}{\hat{\pi}_i \prod_{j=1}^{n} \hat{p}_{j}^{x_{i,j}} (1 - \hat{p}_{j})^{1-x_{i,j}} + (1 - \hat{\pi}_i) \prod_{j=1}^{n} \hat{p}_{j}^{1-x_{i,j}} (1 - \hat{p}_{j})^{x_{i,j}}}.
\]

Therefore, using \(q(y) := q_i^y (1 - q_i)^{1-y}\), expected complete log-likelihood is

\[
\sum_{y \in \{0,1\}} q(y) \cdot \ln \Pr_\theta(X = x \land Y = y)
\]

\[
= (1 - q_i) \left[ \ln(1 - \pi_i) + \sum_{j=1}^{n} (1 - x_{i,j}) \ln p_j + x_{i,j} \ln(1 - p_j) \right]
\]

\[
+ q_i \left[ \ln \pi_i + \sum_{j=1}^{n} x_{i,j} \ln p_j + (1 - x_{i,j}) \ln(1 - p_j) \right].
\]
Now consider all $m$ items, and use independence of $(X_1, Y_1), \ldots, (X_m, Y_m)$. 

**Derivation of E step:** given parameter values $\hat{\theta}$, compute 

\[ q_i := \hat{\pi}_i \prod_{j=1}^{n} \hat{p}_{x_{i,j}}(1 - \hat{p}_j)^{1 - x_{i,j}} \times \hat{\pi}_i \prod_{j=1}^{n} \hat{p}_{x_{i,j}}(1 - \hat{p}_j)^{1 - x_{i,j}} + (1 - \hat{\pi}_i) \prod_{j=1}^{n} \hat{p}_1^{1 - x_{i,j}} \times \hat{p}_j^{x_{i,j}}. \]

for all $i \in [m]$, which together determine $L_{LB}$. 

**Derivation of M step:** With $q_1, q_2, \ldots, q_m$ fixed to determine $L_{LB}$, update $\hat{\pi}_i$ and $\hat{p}_j$ to maximize $L_{LB}(\theta) := m \sum_{i=1}^{m} \sum_{y_i \in \{0, 1\}} q_{y_i} (1 - q_i)^{1 - y_i} \cdot \ln Pr_{\theta}(X_i = x_i \land Y_i = y_i) = m \sum_{i=1}^{m} (1 - q_i) \left[ \ln(1 - \pi_i) + n \sum_{j=1}^{n} x_{i,j} \ln p_j + (1 - x_{i,j}) \ln(1 - p_j) \right] + m \sum_{i=1}^{m} q_i \left[ \ln \pi_i + n \sum_{j=1}^{n} x_{i,j} \ln p_j + (1 - x_{i,j}) \ln(1 - p_j) \right].$

(Obtain using first-order condition for optimality—i.e., derivative equals zero.)
Now consider all $m$ items, and use independence of $(X_1, Y_1), \ldots, (X_m, Y_m)$.

**Derivation of E step:** given parameter values $\hat{\theta}$, compute

$$q_i := \frac{\hat{\pi}_i \prod_{j=1}^{n} \hat{p}^{x_{i,j}} (1 - \hat{p}_j)^{1-x_{i,j}}}{\hat{\pi}_i \prod_{j=1}^{n} \hat{p}_j^{x_{i,j}} (1 - \hat{p}_j)^{1-x_{i,j}} + (1 - \hat{\pi}_i) \prod_{j=1}^{n} \hat{p}_j^{1-x_{i,j}} (1 - \hat{p}_j)^{x_{i,j}}}$$

for all $i \in [m]$, which together determine $\mathcal{L}_{LB}$. 
Now consider all \( m \) items, and use independence of \((X_1, Y_1), \ldots, (X_m, Y_m)\).

**Derivation of E step:** given parameter values \( \hat{\theta} \), compute

\[
q_i := \frac{\hat{\pi}_i \prod_{j=1}^n \hat{p}_{j}^{x_{i,j}} (1 - \hat{p}_j)^{1-x_{i,j}}}{\hat{\pi}_i \prod_{j=1}^n \hat{p}_{j}^{x_{i,j}} (1 - \hat{p}_j)^{1-x_{i,j}} + (1 - \hat{\pi}_i) \prod_{j=1}^n \hat{p}_j^{1-x_{i,j}} (1 - \hat{p}_j)^{x_{i,j}}}
\]

for all \( i \in [m] \), which together determine \( \mathcal{L}_{LB} \).

**Derivation of M step:** With \( q_1, q_2, \ldots, q_m \) fixed to determine \( \mathcal{L}_{LB} \), update \( \hat{\pi}_i \) and \( \hat{p}_j \) to maximize

\[
\mathcal{L}_{LB}(\theta) := \sum_{i=1}^m \sum_{y_i \in \{0,1\}} q_i^{y_i} (1 - q_i)^{1-y_i} \cdot \ln \Pr_{\theta}(X_i = x_i \land Y_i = y_i)
\]

\[
= \sum_{i=1}^m (1 - q_i) \left[ \ln(1 - \pi_i) + \sum_{j=1}^n (1 - x_{i,j}) \ln p_j + x_{i,j} \ln(1 - p_j) \right]
\]

\[
+ \sum_{i=1}^m q_i \left[ \ln \pi_i + \sum_{j=1}^n x_{i,j} \ln p_j + (1 - x_{i,j}) \ln(1 - p_j) \right].
\]

(Obtain using first-order condition for optimality—i.e., derivative equals zero.)
**E-M for MTurk model**

**Input**: observed responses $x_{i,j}$ for $i \in [m]$, $j \in [n]$. Initialize $(\hat{\pi}, \hat{p})$ somehow. Then repeat the following.

- **E step**: for all $i \in [m]$,

  $$q_i = \frac{\hat{\pi}_i \prod_{j=1}^n \hat{p}_{j}^{x_{i,j}} (1 - \hat{p}_j)^{1-x_{i,j}}}{\hat{\pi}_i \prod_{j=1}^n \hat{p}_{j}^{x_{i,j}} (1 - \hat{p}_j)^{1-x_{i,j}} + (1 - \hat{\pi}_i) \prod_{j=1}^n \hat{p}_j^{1-x_{i,j}} (1 - \hat{p}_j)x_{i,j}}.$$  

- **M step**:

  $$\hat{\pi}_i := q_i \quad \text{for all } i \in [m];$$
  $$\hat{p}_j := \frac{1}{m} \sum_{i=1}^m \left\{ q_ix_{i,j} + (1 - q_i)(1 - x_{i,j}) \right\} \quad \text{for all } j \in [n].$$

**Output**:

- $\hat{\pi}_i$ = probability that correct label of item $i$ is 1.
- $\hat{p}_j$ = probability that worker $j$ gives the correct label.
Mixture models: hidden variable model for “soft clustering” / modeling hidden subpopulations.

Maximum likelihood usually intractable for hidden variable models (and sometimes gives degenerate solutions anyway!).

E-M algorithm: local optimization algorithm for climbing log-likelihood objective for hidden variable models.

General recipe for deriving E-M algorithm.