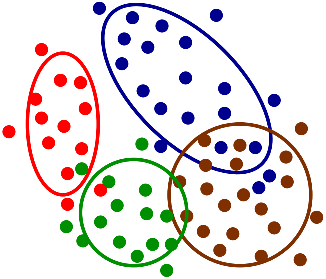
COMS 4771 Lecture 19

1. Mixture models
2. Expectation-Maximization

Mixture models

Unsupervised classification

- **Input**: $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- **Output**: function $f \colon \mathbb{R}^d \to \{1, 2, \ldots, k\} =: [k]$.
- **Typical semantics**: hidden subpopulation structure.



Gaussian mixture model

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

$$
\begin{aligned}
Y &\sim \boldsymbol{\pi} && \text{(discrete distribution over } [k];\ \Pr_{\boldsymbol{\theta}}(Y = j) = \pi_j\text{)} \\
\boldsymbol{X} \big| Y = j &\sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) && \text{(Gaussian with mean } \boldsymbol{\mu}_j \text{ and covariance } \boldsymbol{\Sigma}_j\text{)}
\end{aligned}
$$

Parameters $\boldsymbol{\theta} := (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Gaussian mixture model

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

$$
\begin{aligned}
Y &\sim \boldsymbol{\pi} && \text{(discrete distribution over } [k]\text{; } \Pr_{\boldsymbol{\theta}}(Y=j)=\pi_j\text{)} \\
\boldsymbol{X} \big| Y = j &\sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) && \text{(Gaussian with mean } \boldsymbol{\mu}_j \text{ and covariance } \boldsymbol{\Sigma}_j\text{)}
\end{aligned}
$$

Parameters $\boldsymbol{\theta} := (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \dots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Looks familiar?

Gaussian mixture model

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

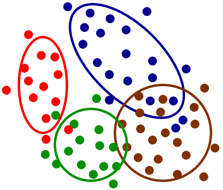
$$\begin{aligned} \boxed{Y \;\sim\; \boldsymbol{\pi}} & \qquad \text{(discrete distribution over } [k]\text{; } \Pr_{\boldsymbol{\theta}}(Y = j) = \pi_j\text{)} \\ \boldsymbol{X} \big| Y = j \;\sim\; \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) & \qquad \text{(Gaussian with mean } \boldsymbol{\mu}_j \text{ and covariance } \boldsymbol{\Sigma}_j\text{)} \end{aligned}$$

sub-population

Parameters $\boldsymbol{\theta} := (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Looks familiar?

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



Gaussian mixture model

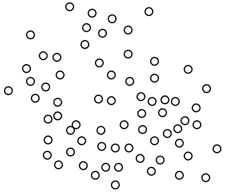
$(\boldsymbol{X}, Y) \sim P_{\boldsymbol{\theta}}$, a distribution over $\mathbb{R}^d \times [k]$ where

$$\begin{aligned} Y &\sim \boldsymbol{\pi} && \text{(discrete distribution over } [k]\text{; } \Pr_{\boldsymbol{\theta}}(Y = j) = \pi_j\text{)} \\ \boldsymbol{X} \big| Y = j &\sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) && \text{(Gaussian with mean } \boldsymbol{\mu}_j \text{ and covariance } \boldsymbol{\Sigma}_j\text{)} \end{aligned}$$

Parameters $\boldsymbol{\theta} := (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Looks familiar?

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



Gaussian mixture model

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

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

$$\boldsymbol{X} \big| Y = j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \qquad \text{(Gaussian with mean } \boldsymbol{\mu}_j \text{ and covariance } \boldsymbol{\Sigma}_j\text{)}$$

Parameters $\boldsymbol{\theta} := (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Looks familiar?

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

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

$$\boldsymbol{X} \sim \boxed{\sum_{j=1}^{k} \pi_j} \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

it indicates the X could come from all those different gussian distributions

Mixing weights $\boldsymbol{\pi}$; **mixture components** $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \ldots, \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Gaussian mixture model

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

$$
\begin{aligned}
Y &\sim \boldsymbol{\pi} && \text{(discrete distribution over } [k]\text{; } \Pr_{\boldsymbol{\theta}}(Y = j) = \pi_j\text{)} \\
\boldsymbol{X} \big| Y = j &\sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) && \text{(Gaussian with mean } \boldsymbol{\mu}_j \text{ and covariance } \boldsymbol{\Sigma}_j\text{)}
\end{aligned}
$$

Parameters $\boldsymbol{\theta} := (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Looks familiar?

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

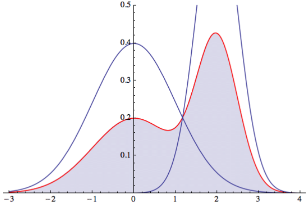
Models of this sort are called **mixture models**;
this one in particular is called the **Gaussian mixture model**.

$$
p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j \cdot (2\pi)^{-d/2} \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) \right)
$$

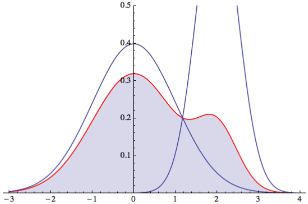
Mixing weights $\boldsymbol{\pi}$; **mixture components** $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \ldots, \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Gaussian mixtures in $\mathbb{R}^1$

interesting!

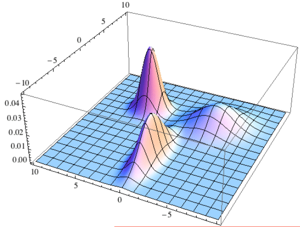


$$\frac{1}{2}\mathcal{N}(0,1) + \frac{1}{2}\mathcal{N}(2,1/4)$$

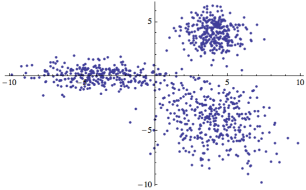


$$\frac{4}{5}\mathcal{N}(0,1) + \frac{1}{5}\mathcal{N}(2,1/4)$$

Gaussian mixtures in $\mathbb{R}^2$



Plot of the mixture density.



A sample of size 1000.

Soft clustering

Suppose you have the parameters $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ of a Gaussian mixture distribution, and further that $(\boldsymbol{X}, Y) \sim P_{\boldsymbol{\theta}}$.

Soft clustering

Suppose you have the parameters $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ of a Gaussian mixture distribution, and further that $(\boldsymbol{X}, Y) \sim P_{\boldsymbol{\theta}}$.

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

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

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

Soft clustering

Suppose you have the parameters $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \dots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ of a Gaussian mixture distribution, and further that $(\boldsymbol{X}, Y) \sim P_{\boldsymbol{\theta}}$.

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

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

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

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

$$\mathbb{E}_{\boldsymbol{\theta}}[\Phi_j \mid \boldsymbol{X} = \boldsymbol{x}] \;=\; \Pr_{\boldsymbol{\theta}}[Y = j \mid \boldsymbol{X} = \boldsymbol{x}]$$

Soft clustering

Suppose you have the parameters $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ of a Gaussian mixture distribution, and further that $(\boldsymbol{X}, Y) \sim P_{\boldsymbol{\theta}}$.

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

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

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

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

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\theta}}[\Phi_j \mid \boldsymbol{X} = \boldsymbol{x}] \; &= \; \Pr_{\boldsymbol{\theta}}[Y = j \mid \boldsymbol{X} = \boldsymbol{x}] \\ &= \; \frac{\Pr_{\boldsymbol{\theta}}[Y = j] \cdot \Pr_{\boldsymbol{\theta}}[\boldsymbol{X} = \boldsymbol{x} \mid Y = j]}{\Pr_{\boldsymbol{\theta}}[\boldsymbol{X} = \boldsymbol{x}]} \end{aligned}$$

Soft clustering

Suppose you have the parameters $\boldsymbol{\theta} = (\boxed{\pi_1}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ of a Gaussian mixture distribution, and further that $(\boldsymbol{X}, Y) \sim P_{\boldsymbol{\theta}}$.

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

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

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

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

where to get this one???

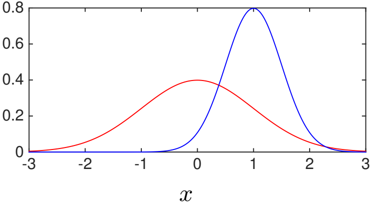
$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\theta}}[\Phi_j \mid \boldsymbol{X} = \boldsymbol{x}] \ &= \ \Pr_{\boldsymbol{\theta}}[Y = j \mid \boldsymbol{X} = \boldsymbol{x}] \\
&= \ \frac{\Pr_{\boldsymbol{\theta}}[Y = j] \cdot \Pr_{\boldsymbol{\theta}}[\boldsymbol{X} = \boldsymbol{x} \mid Y = j]}{\Pr_{\boldsymbol{\theta}}[\boldsymbol{X} = \boldsymbol{x}]} \\
&= \ \frac{\boxed{\pi_j} \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right)}{\sum_{j'=1}^{k} \pi_{j'} \cdot \sqrt{\det(\boldsymbol{\Sigma}_{j'}^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_{j'})^\top \boldsymbol{\Sigma}_{j'}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_{j'})\right)}.
\end{aligned}$$

Soft clustering

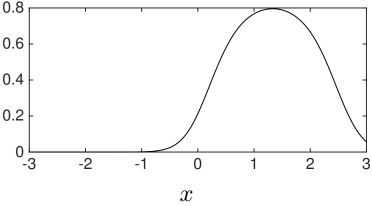
interesting converstion

Example: a Gaussian mixture with $k = 2$ in $\mathbb{R}^1$.

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



$\Pr_{\boldsymbol{\theta}}[Y = 1 \,|\, X = x]$



$$\Pr_{\boldsymbol{\theta}}[Y = 1 \,|\, X = x] = \frac{\pi_1 \cdot \frac{1}{\sigma_1} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right)}{\pi_1 \cdot \frac{1}{\sigma_1} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right) + \pi_2 \cdot \frac{1}{\sigma_2} \exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)}.$$

Parameter estimation for Gaussian mixtures

Maximum likelihood estimation of $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ given data $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}$ (assumed to be an i.i.d. sample).

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

Parameter estimation for Gaussian mixtures

Maximum likelihood estimation of $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ given data $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}$ (assumed to be an i.i.d. sample).

$$\begin{aligned} \boldsymbol{\theta}_{\mathsf{ML}} &:= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) \\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln \left\{ \sum_{j=1}^{k} \pi_j \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) \right) \right\} \end{aligned}$$

Parameter estimation for Gaussian mixtures

Maximum likelihood estimation of $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ given data $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}$ (assumed to be an i.i.d. sample).

come from all distributation!

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

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln \left\{ \sum_{j=1}^{k} \pi_j \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) \right) \right\}$$

Interesting!!!
we get \theta through
this magic way!!!

Ack! $\ln \left\{ \sum_{j=1}^{k} \cdots \right\}$ does not simplify nicely!

Text

MLE for Gaussian mixtures: **not a convex optimization problem**.

$$\arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln\left\{ \sum_{j=1}^{k} \pi_j \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_j) \right) \right\}$$

MLE for Gaussian mixtures

MLE for Gaussian mixtures: **not a convex optimization problem**.

$$\arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln\left\{\sum_{j=1}^{k} \pi_j \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right)\right\}$$

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

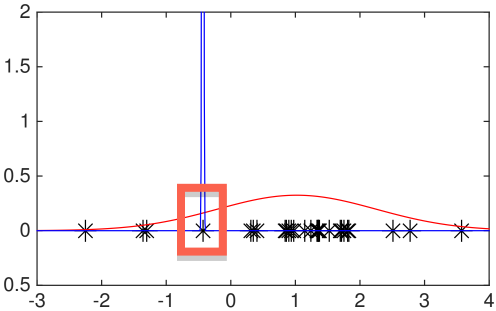
MLE for Gaussian mixtures

MLE for Gaussian mixtures: **not a convex optimization problem**.

$$\arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ln \left\{ \sum_{j=1}^{k} \boxed{\pi_j} \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) \right) \right\}$$

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

could come from all distribution!



one component for single point

the distribution for one point!!!

What a bad!

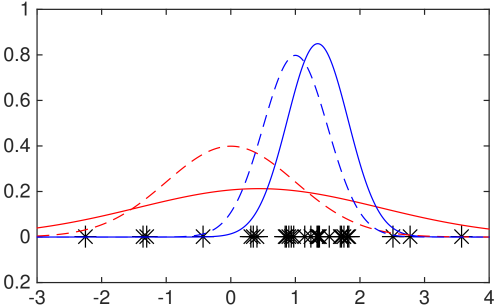
But this is a good thing, because the **MLE is degenerate**: $\mu_1 = x^{(1)}$, $\sigma_1^2 \to 0$, likelihood $\to \infty$.

Local optimization

Saving grace:
If the data are actually generated by a Gaussian mixture with parameters $\boldsymbol{\theta}_\star$,
then $\theta_\star$ may be <u>close</u> to some <u>local maximizer</u> of the log-likelihood.

Saving grace:
If the data are actually generated by a Gaussian mixture with parameters $\boldsymbol{\theta}_\star$,
then $\boldsymbol{\theta}_\star$ may be <u>close</u> to some <u>local maximizer</u> of the log-likelihood.

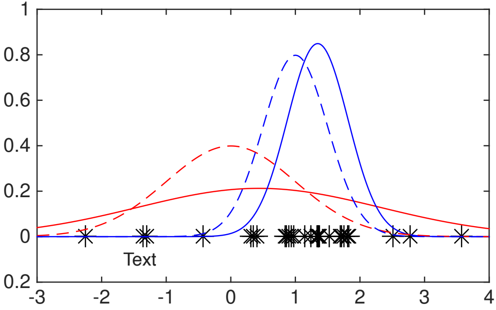
Just need to find the "right" local maximizer . . .
(i.e., a good, non-degenerate local maximizer).



Local optimization

Saving grace:
If the data are actually generated by a Gaussian mixture with parameters $\boldsymbol{\theta}_\star$, **then $\boldsymbol{\theta}_\star$ 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).



catch the local maximizer, which is near \theta star

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:

$$(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \dots, (\boldsymbol{x}^{(n)}, y^{(n)}) \in \mathbb{R}^d \times [k].$$

Motivation

Suppose we had a *labeled* iid sample:

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

Suppose we had a *labeled* iid sample:

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

The "complete log-likelihood" of $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is

$$\sum_{i=1}^{n} \sum_{j=1}^{k} \phi_j^{(i)} \ln\left\{\pi_j \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right)\right\}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{k} \phi_j^{(i)} \left(\ln \pi_j + \frac{1}{2} \ln \det(\boldsymbol{\Sigma}_j^{-1}) - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right),$$

which can be easily maximized w.r.t. $\boldsymbol{\theta}$.

Motivation

Suppose we had a *labeled* iid sample:

hot point

$$(\boldsymbol{x}^{(1)}, \boldsymbol{\phi}^{(1)}), (\boldsymbol{x}^{(2)}, \boldsymbol{\phi}^{(2)}), \dots, (\boldsymbol{x}^{(n)}, \boldsymbol{\phi}^{(n)}) \in \mathbb{R}^d \times \{0,1\}^k.$$

The "complete log-likelihood" of $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \dots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is

$$\sum_{i=1}^{n}\sum_{j=1}^{k} \phi_j^{(i)} \ln\left\{\pi_j \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right)\right\}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{k} \phi_j^{(i)} \left(\ln \pi_j + \frac{1}{2}\ln\det(\boldsymbol{\Sigma}_j^{-1}) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right),$$

which can be easily maximized w.r.t. $\boldsymbol{\theta}$.

In fact, even easy with *soft assignments* $w_j^{(i)} := \mathbb{E}_{\boldsymbol{\theta}}[\phi_j^{(i)} \mid \boldsymbol{X} = \boldsymbol{x}^{(i)}]$:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} \mathbb{E}_{\boldsymbol{\theta}}\left[\Phi_j^{(i)} \mid \boldsymbol{X} = \boldsymbol{x}^{(i)}\right]\left(\ln \pi_j + \frac{1}{2}\ln\det(\boldsymbol{\Sigma}_j^{-1}) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right).$$

"Expectation (w.r.t. $P_{\boldsymbol{\theta}}$ conditioned on $\{\boldsymbol{x}^{(i)}\}$) of complete log-likelihood."

Motivation

Suppose we had a *labeled* iid sample:

if we have label

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

The "complete log-likelihood" of $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is

from sample!!!

soft assignment

$$\sum_{i=1}^{n} \sum_{j=1}^{k} \phi_j^{(i)} \ln\left\{ \pi_j \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) \right) \right\}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{k} \boxed{\phi_j^{(i)}} \left(\ln \pi_j + \frac{1}{2} \ln\det(\boldsymbol{\Sigma}_j^{-1}) - \frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) \right),$$

which can be easily maximized w.r.t. $\boldsymbol{\theta}$.

assignment expectation!

In fact, even easy with *soft assignments* $w_j^{(i)} := \mathbb{E}_{\boldsymbol{\theta}}[\phi_j^{(i)} \mid \boldsymbol{X} = \boldsymbol{x}^{(i)}]$:

compute the expectation of assignment!!

$$\sum_{i=1}^{n} \sum_{j=1}^{k} w_j^{(i)} \left(\ln \pi_j + \frac{1}{2} \ln\det(\boldsymbol{\Sigma}_j^{-1}) - \frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j) \right).$$

"Expectation (w.r.t. $P_{\boldsymbol{\theta}}$ conditioned on $\{\boldsymbol{x}^{(i)}\}$) of complete log-likelihood."

Expectation-Maximization (E-M)

Initialize $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ somehow.

Expectation-Maximization (E-M)

Initialize $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \dots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ somehow. Then repeat:

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

$$w_j^{(i)} := \frac{\pi_j \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right)}{\sum_{j'=1}^{k} \pi_{j'} \cdot \sqrt{\det(\boldsymbol{\Sigma}_{j'}^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_{j'})^\top \boldsymbol{\Sigma}_{j'}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_{j'})\right)}$$

Expectation-Maximization (E-M)

Initialize $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ somehow. Then repeat:

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

note: we use soft assignment all the way!

$$w_j^{(i)} := \frac{\pi_j \cdot \sqrt{\det(\boldsymbol{\Sigma}_j^{-1})} \exp\left(-\frac{1}{2}(\boxed{\boldsymbol{x}} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_j)\right)}{\sum_{j'=1}^{k} \pi_{j'} \cdot \sqrt{\det(\boldsymbol{\Sigma}_{j'}^{-1})} \exp\left(-\frac{1}{2}(\boxed{\boldsymbol{x}} - \boldsymbol{\mu}_{j'})^\top \boldsymbol{\Sigma}_{j'}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{j'})\right)}$$

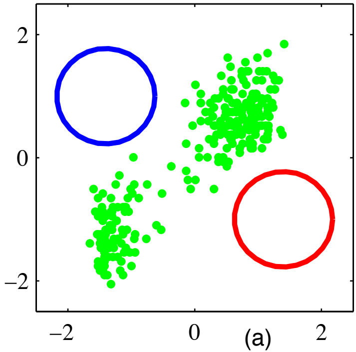
2. **M step**: maximize "expected complete log-likelihood" w.r.t. parameters.
   For each $j \in \{1, 2, \ldots, k\}$,

through soft assignment, we get the weighted label, then we try to maximize the weighted llabel.

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

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

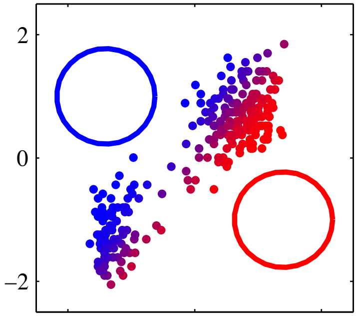
$$\boldsymbol{\Sigma}_j := \frac{1}{n\pi_j}\sum_{i=1}^{n} w_j^{(i)} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^\top.$$



(a)

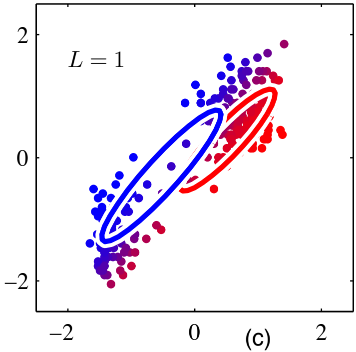
Arbitrary initialization of $\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$ for $j \in \{1, 2\}$.

covariance matrix: Identity Matrix



(b)

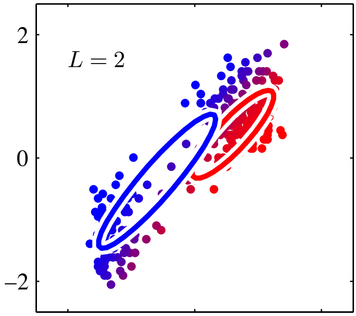
E step: soft assignments $z_j^{(i)}$ for each $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2\}$.



(c)

M step: update parameters $\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$ for $j \in \{1, 2\}$.

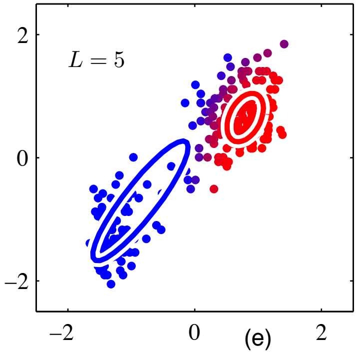
Sample run of the E-M algorithm



(d)

After **two rounds** of E-M.

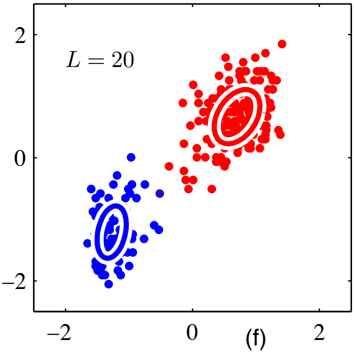
Sample run of the E-M algorithm



(e)

After **five rounds** of E-M.

Sample run of the E-M algorithm



(f)

After **20 rounds** of E-M.

E-M for Gaussian mixtures

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

$$w_j^{(i)} \propto \pi_j \cdot p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}^{(i)})$$

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

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

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

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

$$\boldsymbol{\Sigma}_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^\top.$$

E-M for Gaussian mixtures

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

$$w_j^{(i)} \propto \pi_j \cdot p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}^{(i)})$$

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

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

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

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

$$\boldsymbol{\Sigma}_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^\top.$$

Some details

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

Using the E-M algorithm

E-M for Gaussian mixtures

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

$$w_j^{(i)} \propto \pi_j \cdot p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}^{(i)})$$

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

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

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

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

$$\boldsymbol{\Sigma}_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^\top.$$

Some details

- **Initialization**: a bit of an art; both D^2-sampling and Lloyd's algorithm are reasonable.
- **Starved clusters**: problems can occur if π_j becomes too small (e.g., $\boldsymbol{\Sigma}_j$ could be near singular).

E-M for Gaussian mixtures

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

$$w_j^{(i)} \propto \pi_j \cdot p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}^{(i)})$$

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

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

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

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

$$\boldsymbol{\Sigma}_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^\top.$$

Some details

- **Initialization**: a bit of an art; both D^2-sampling and Lloyd's algorithm are reasonable.
- **Starved clusters**: problems can occur if π_j becomes too small (e.g., $\boldsymbol{\Sigma}_j$ could be near singular).

  Remove/replace such components.

Using the E-M algorithm

E-M for Gaussian mixtures

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

$$w_j^{(i)} \propto \pi_j \cdot p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}^{(i)})$$

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

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

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

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

$$\boldsymbol{\Sigma}_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^\top.$$

Some details

- **Initialization**: a bit of an art; both D^2-sampling and Lloyd's algorithm are reasonable.
- **Starved clusters**: problems can occur if π_j becomes too small (e.g., $\boldsymbol{\Sigma}_j$ could be near singular).

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

E-M for Gaussian mixtures

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

$$w_j^{(i)} \propto \pi_j \cdot p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}^{(i)})$$

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

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

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

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

$$\boldsymbol{\Sigma}_j := \frac{1}{n\pi_j} \sum_{i=1}^{n} w_j^{(i)} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^{\top}.$$

Some details

random initialization

- **Initialization**: a bit of an art; both D^2-sampling and Lloyd's algorithm are reasonable.
- **Starved clusters**: problems can occur if π_j becomes too small (e.g., $\boldsymbol{\Sigma}_j$ 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.
e.g., for Gaussian mixtures,

$$\Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x} \wedge Y = j) \; = \; \pi_j \cdot p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x})$$

($\boldsymbol{X}$ is observed, but Y is hidden).

Derivation of E-M algorithm

note the hidden variables, and the expectation of hidden variables.

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.
e.g., for Gaussian mixtures,

$$\Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x} \wedge Y = j) \;=\; \pi_j \cdot p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x})$$

($\boldsymbol{X}$ is observed, but Y is hidden).

Likelihood of $\boldsymbol{\theta}$ given $\boldsymbol{X} = \boldsymbol{x}$ is

Y is actually assigned by you!

$$\boxed{\Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x})} = \boxed{\sum_{j=1}^{k} \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x} \wedge Y = j)} = \sum_{j=1}^{k} \pi_j \cdot p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}).$$

Actually, this is the initial goal to maximize!

Derivation of E-M algorithm

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

Log-likelihood of $\boldsymbol{\theta}$ given $\boldsymbol{x}^{(i)}$ is

$$\ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)}) \quad = \quad \ln\left(\sum_{j=1}^{k} \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)} \wedge Y = j)\right)$$

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

Log-likelihood of $\boldsymbol{\theta}$ given $\boldsymbol{x}^{(i)}$ is

$$\begin{aligned}
\ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)}) &= \ln\left(\sum_{j=1}^{k} \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)} \wedge Y = j)\right) \\
&= \ln\left(\sum_{j=1}^{k} q_j \cdot \frac{\Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)} \wedge Y = j)}{q_j}\right).
\end{aligned}$$

Derivation of E-M algorithm

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

note: here is for a single point!

Log-likelihood of $\boldsymbol{\theta}$ given $\boldsymbol{x}^{(i)}$ is

$$\begin{aligned}
\ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)}) &= \ln \left(\sum_{j=1}^{k} \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)} \wedge Y = j) \right) \\
&= \ln \left(\sum_{j=1}^{k} q_j \cdot \frac{\Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)} \wedge Y = j)}{q_j} \right).
\end{aligned}$$

By Jensen's inequality and concavity of $\ln$, if $\boldsymbol{q} = (q_1, q_2, \ldots, q_k)$ is a probability distribution, then

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

Derivation of E-M algorithm

Now consider all n data points $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}$. By independence,

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

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

Now consider all n data points $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}$. By independence,

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

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

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

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

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

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

Derivation of E-M algorithm

Now consider all n data points $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(n)}$. By independence,

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

Bayes' rule shows that $\boxed{\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathrm{LB}}(\boldsymbol{\theta}) \text{ when } q_j^{(i)} = \Pr_{\boldsymbol{\theta}}(Y = j \mid \boldsymbol{X} = \boldsymbol{x}^{(i)}).}$

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

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

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

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

- **M step**: Update $\boldsymbol{\theta}$ to maximize $\mathcal{L}_{\mathrm{LB}}$

$$\mathcal{L}_{\mathrm{LB}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)} \wedge Y = j) - \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \ln q_j^{(i)}$$

use the lower bound of inequation. update
on \theta to maximize L(\theta)

with the same \theta, the left part must larger than right part. thus we should keep on increasing right part to maximize the left part.

underlying principle

Now consider all n data points $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}$. By independence,

note the decomposition

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

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

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

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

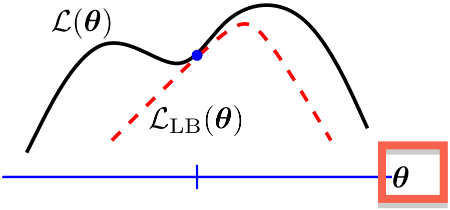
  $$q_j^{(i)} := \Pr_{\boldsymbol{\theta}}(Y = j \mid \boldsymbol{X} = \boldsymbol{x}^{(i)})$$

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

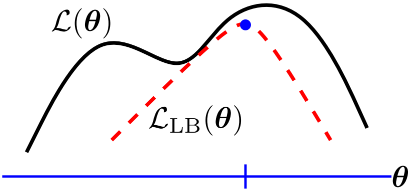
- **M step**: Update $\boldsymbol{\theta}$ to maximize $\mathcal{L}_{\mathrm{LB}}$

$$\begin{aligned}\mathcal{L}_{\mathrm{LB}}(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}^{(i)} \wedge Y = j) - \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \ln q_j^{(i)} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} \cdot \left(\ln \pi_j + \ln p_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}^{(i)})\right) + \text{const.}\end{aligned}$$

compute qj(i)



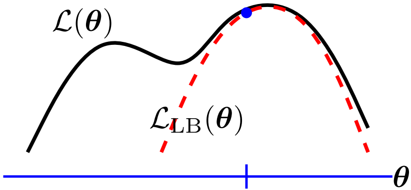
E step: construct $\mathcal{L}_{\text{LB}}$ such that $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{LB}}(\boldsymbol{\theta})$ for current $\boldsymbol{\theta}$.



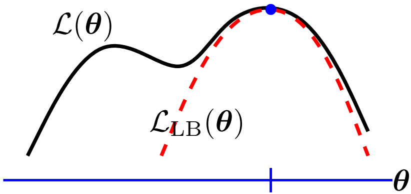
M step: choose $\boldsymbol{\theta}$ to maximize $\mathcal{L}_{\mathrm{LB}}$.

note the change of
lower bond~~

CONSTRUCTING AND MAXIMIZING $\mathcal{L}_{\mathrm{LB}}$



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



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

Other hidden variable models

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

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 π_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.

just 0, 1

Simple Mechanical Turk (MTurk) model: $\boxed{m \text{ items,}}$ n workers.

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

worker make the correct label on item i
(correct label is the nature label)

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

Random variables:

- (Hidden) Y_i is the correct label for item i;

the nature label

pj is the same for all items

$$\Pr_{\boldsymbol{\theta}}(Y_i = 1) = \pi_i.$$

- (Observed) $\boxed{X_{i,j}}$ is the response given by worker j for item i;

response???

could be 1 or 0

$$\Pr_{\boldsymbol{\theta}}(X_{i,j} = \boxed{Y_i}) = p_j.$$

Xi,j is the right label(nature label)

Log-likelihood for MTurk

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

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

$$\ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i)$$

LOG-LIKELIHOOD FOR MTURK

just consider this as a label responded by worker

For now, pretend there's only one item i; $\boldsymbol{X}_i := \boxed{(X_{i,1}, X_{i,2}, \ldots, X_{i,n})}$ and Y_i.
Let $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n}) \in \{0,1\}^n$ be the observed responses.

$$\begin{aligned}
&\ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i) \\
&\quad = \ln \sum_{y \in \{0,1\}} q(y) \cdot \frac{\Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i \wedge Y_i = y)}{q(y)}
\end{aligned}$$

Log-likelihood for MTurk

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

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

$$
\begin{aligned}
&\ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i) \\
&\quad = \quad \ln \sum_{y \in \{0,1\}} q(y) \cdot \frac{\Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i \wedge Y_i = y)}{q(y)} \\
&\quad \geq \quad \sum_{y \in \{0,1\}} q(y) \cdot \ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i \wedge Y_i = y) - \sum_{y \in \{0,1\}} q(y) \ln q(y).
\end{aligned}
$$

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

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

we have n workers

$$\ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i)$$

xi could appear in both condition. when y = 0 or y = 1

maximize on this part

$$= \ln \sum_{y \in \{0,1\}} q(y) \cdot \frac{\Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i \wedge Y_i = y)}{q(y)}$$

$$\geq \boxed{\sum_{y \in \{0,1\}} q(y) \cdot \ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i \wedge Y_i = y)} - \sum_{y \in \{0,1\}} q(y) \ln q(y).$$

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

when y = 0, x(i, j) = 1,
the guess is wrong.
prob = (1- /pi) * (1 - pj)

read carefully, it includes all situation

$$\ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i \wedge Y_i = y)$$

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

note (1-y) and y. when y in {0, 1}. only one of them could be 1, the other must be 0

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

Log-likelihood (lower-bound) for MTurk

By independence and Bayes' rule:

$$\Pr_{\boldsymbol{\theta}}(Y_i = y \mid \boldsymbol{X}_i = \boldsymbol{x}_i) =: q_i^y (1 - q_i)^{1-y}$$

where

$$\begin{aligned} q_i &:= \Pr_{\boldsymbol{\theta}}(Y_i = 1 \mid \boldsymbol{X}_i = \boldsymbol{x}_i) \\ &= \frac{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1 - p_j)^{1 - x_{i,j}}}{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1 - p_j)^{1 - x_{i,j}} + (1 - \pi_i) \prod_{j=1}^n p_j^{1 - x_{i,j}} (1 - p_j)^{x_{i,j}}}. \end{aligned}$$

Log-likelihood (lower-bound) for MTurk

By independence and Bayes' rule:

some kind of qi

$$\Pr_{\boldsymbol{\theta}}(Y_i = y \mid \boldsymbol{X}_i = \boldsymbol{x}_i) =: q_i^y (1 - q_i)^{1-y}$$

where This is for E

the qi is in this form

$$q_i := \Pr_{\boldsymbol{\theta}}(Y_i = 1 \mid \boldsymbol{X}_i = \boldsymbol{x}_i)$$

$$= \frac{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1 - p_j)^{1 - x_{i,j}}}{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1 - p_j)^{1 - x_{i,j}} + (1 - \pi_i) \prod_{j=1}^n p_j^{1 - x_{i,j}} (1 - p_j)^{x_{i,j}}}.$$

Therefore, when $q(y) = q_i^y (1 - q_i)^{1-y}$,

when q(y) by chance equal to qi at here. qi depends y

This is for M

When Yi = 0

$$\sum_{y \in \{0,1\}} q(y) \cdot \ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x} \wedge \boldsymbol{Y} = \boldsymbol{y})$$

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

it could be perfectly replaced with qi

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

E-M for MTurk model

Now consider all m items, and use independence of $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_m, Y_m)$.

E-M for MTurk model

Now consider all m items, and use independence of $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_m, Y_m)$.

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

$$q_i := \frac{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1-p_j)^{1-x_{i,j}}}{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1-p_j)^{1-x_{i,j}} + (1-\pi_i) \prod_{j=1}^n p_j^{1-x_{i,j}} (1-p_j)^{x_{i,j}}}$$

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

E-M for MTurk model

Now consider all m items, and use independence of $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_m, Y_m)$.

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

$$q_i := \frac{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1-p_j)^{1-x_{i,j}}}{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1-p_j)^{1-x_{i,j}} + (1-\pi_i) \prod_{j=1}^n p_j^{1-x_{i,j}} (1-p_j)^{x_{i,j}}}$$

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

Derivation of M step: With $q_1, q_2, \ldots, q_m$ fixed to determine $\mathcal{L}_{\mathrm{LB}}$, choose π_i and p_j to maximize

$$\mathcal{L}_{\mathrm{LB}}(\boldsymbol{\theta}) := \sum_{i=1}^m \sum_{y_i \in \{0,1\}} \boxed{q_i^{y_i} (1-q_i)^{1-y_i}} \cdot \ln \Pr_{\boldsymbol{\theta}}(\boldsymbol{X}_i = \boldsymbol{x}_i \wedge Y_i = y_i)$$

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

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

(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 $(\boldsymbol{\pi}, \boldsymbol{p})$ somehow. Then repeat the following.

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

$$q_i = \frac{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1-p_j)^{1-x_{i,j}}}{\pi_i \prod_{j=1}^n p_j^{x_{i,j}} (1-p_j)^{1-x_{i,j}} + (1-\pi_i) \prod_{j=1}^n p_j^{1-x_{i,j}} (1-p_j)^{x_{i,j}}}.$$

- **M step**:

$$\pi_i := q_i \quad \text{for all } i \in [m];$$

$$p_j := \frac{1}{m} \sum_{i=1}^m \left\{ q_i x_{i,j} + (1-q_i)(1-x_{i,j}) \right\} \quad \text{for all } j \in [n].$$

Output:

- π_i = probability that correct label of item i is 1.
- p_j = probability that worker j gives the correct label.

correct label could be either 0 or 1

the important part is to extract qi

it seems extraordinarily important!

Recap

- **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.