

EM and GMM

Slides taken from Huang Jia Bin Lectures

What we cover now?

- Examples of Missing Data Problems
 - Detecting outliers
 - Latent topic models
 - Segmentation
- Background
 - Maximum Likelihood Estimation
 - Probabilistic Inference
- Dealing with “Hidden” Variables
 - EM algorithm, Mixture of Gaussians
 - Hard EM

Today's Class

- **Examples of Missing Data Problems**
 - Detecting outliers
 - Latent topic models
 - Segmentation
- Background
 - Maximum Likelihood Estimation
 - Probabilistic Inference
- Dealing with “Hidden” Variables
 - EM algorithm, Mixture of Gaussians
 - Hard EM

Missing Data Problems: Outliers

You want to train an algorithm to predict whether a photograph is attractive. You collect annotations from Mechanical Turk. Some annotators try to give accurate ratings, but others answer randomly.

Challenge: Determine which people to trust and the average rating by accurate annotators.



Annotator
Ratings

10
8
9
2
8

Photo: Jam343 (Flickr)

Missing Data Problems: Object Discovery

You have a collection of images and have extracted regions from them. Each is represented by a histogram of “visual words”.

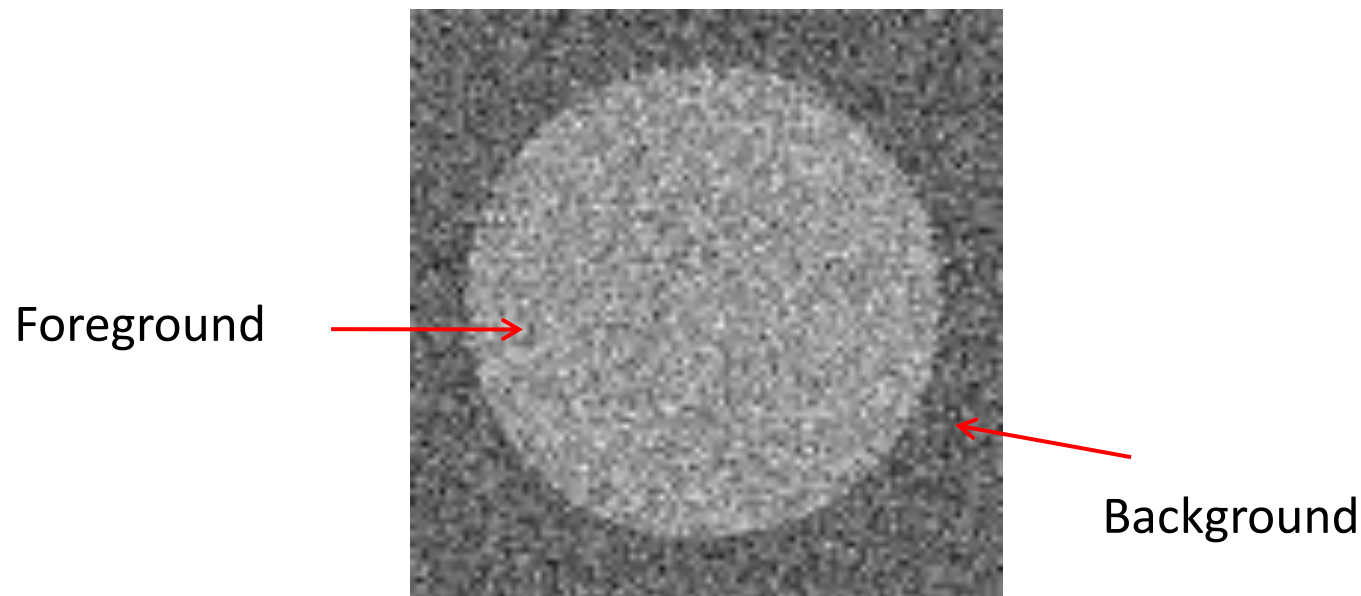
Challenge: Discover frequently occurring object categories, without pre-trained appearance models.



Missing Data Problems: Segmentation

You are given an image and want to assign foreground/background pixels.

Challenge: Segment the image into figure and ground without knowing what the foreground looks like in advance.



Missing Data Problems: Segmentation

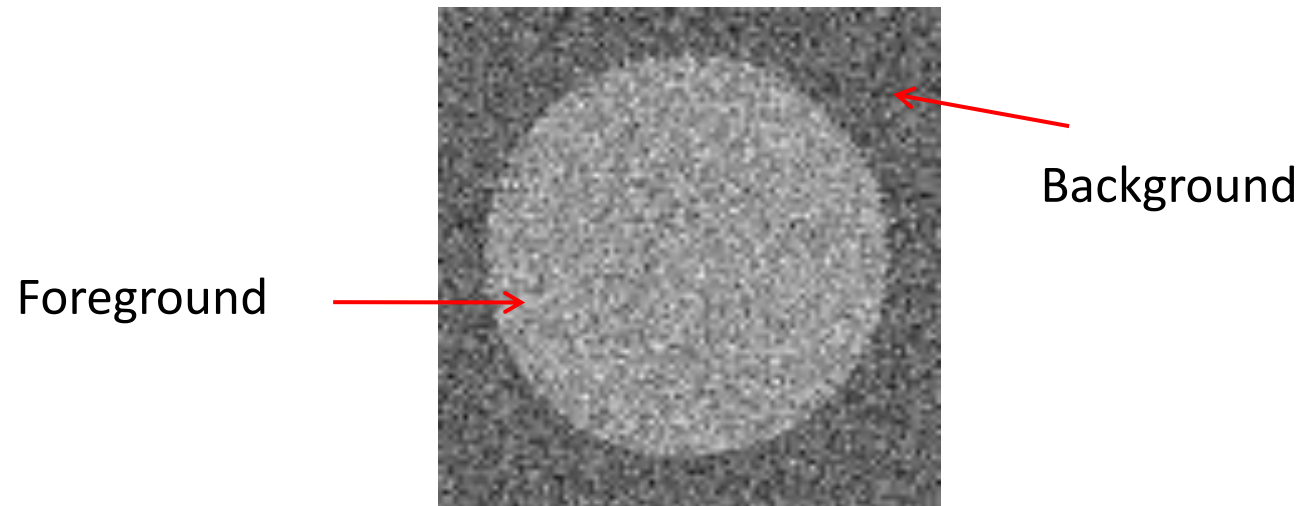
Challenge: Segment the image into figure and ground without knowing what the foreground looks like in advance.

Three steps:

1. If we had labels, how could we model the appearance of foreground and background?
 - **Maximum Likelihood Estimation**
2. Once we have modeled the fg/bg appearance, how do we compute the likelihood that a pixel is foreground?
 - **Probabilistic Inference**
3. How can we get both labels and appearance models at once?
 - **Expectation-Maximization (EM) Algorithm**

Maximum Likelihood Estimation

1. If we had labels, how could we model the appearance of foreground and background?



Maximum Likelihood Estimation

data \rightarrow $\mathbf{x} = \{x_1 \dots x_N\}$

$\hat{\theta} = \operatorname{argmax}_{\theta} p(\mathbf{x} \mid \theta)$ parameters \swarrow

$$\hat{\theta} = \operatorname{argmax}_{\theta} \prod_n p(x_n \mid \theta)$$

Maximum Likelihood Estimation

$$\mathbf{x} = \{x_1 \dots x_N\}$$

$$\hat{\theta} = \operatorname{argmax}_{\theta} p(\mathbf{x} \mid \theta)$$

$$\hat{\theta} = \operatorname{argmax}_{\theta} \prod_n p(x_n \mid \theta)$$

Gaussian Distribution

$$p(x_n \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n - \mu)^2}{2\sigma^2}\right)$$

Maximum Likelihood Estimation

Gaussian Distribution $p(x_n | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n - \mu)^2}{2\sigma^2}\right)$

$$\hat{\theta} = \operatorname{argmax}_{\theta} p(\mathbf{x} | \theta) = \operatorname{argmax}_{\theta} \log p(\mathbf{x} | \theta)$$

Log-Likelihood


$$\hat{\theta} = \operatorname{argmax}_{\theta} \sum_n \log(p(x_n | \theta)) = \operatorname{argmax}_{\theta} L(\theta)$$

$$L(\theta) = \frac{-N}{2} \log(2\pi) - \frac{-N}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_n (x_n - \mu)^2$$

$$\frac{\partial L(\theta)}{\partial \mu} = \frac{1}{\sigma^2} \sum_n (x_n - \mu) = 0 \quad \rightarrow \quad \hat{\mu} = \frac{1}{N} \sum_n x_n$$

$$\frac{\partial L(\theta)}{\partial \sigma} = \frac{N}{\sigma} - \frac{1}{\sigma^3} \sum_n (x_n - \mu)^2 = 0 \quad \rightarrow \quad \sigma^2 = \frac{1}{N} \sum_n (x_n - \hat{\mu})^2$$

Maximum Likelihood Estimation

$$\mathbf{x} = \{x_1 \dots x_N\}$$

$$\hat{\theta} = \operatorname{argmax}_{\theta} p(\mathbf{x} \mid \theta)$$

$$\hat{\theta} = \operatorname{argmax}_{\theta} \prod_n p(x_n \mid \theta)$$

Gaussian Distribution

$$p(x_n \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n - \mu)^2}{2\sigma^2}\right)$$

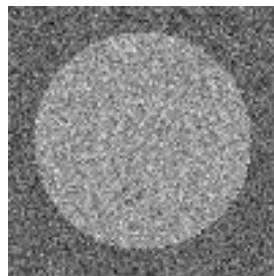
$$\hat{\mu} = \frac{1}{N} \sum_n x_n \quad \hat{\sigma}^2 = \frac{1}{N} \sum_n (x_n - \hat{\mu})^2$$

Example: MLE

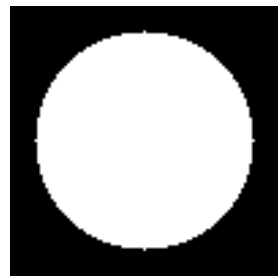
Parameters used to Generate

fg: $\mu=0.6$, $\sigma=0.1$

bg: $\mu=0.4$, $\sigma=0.1$



im

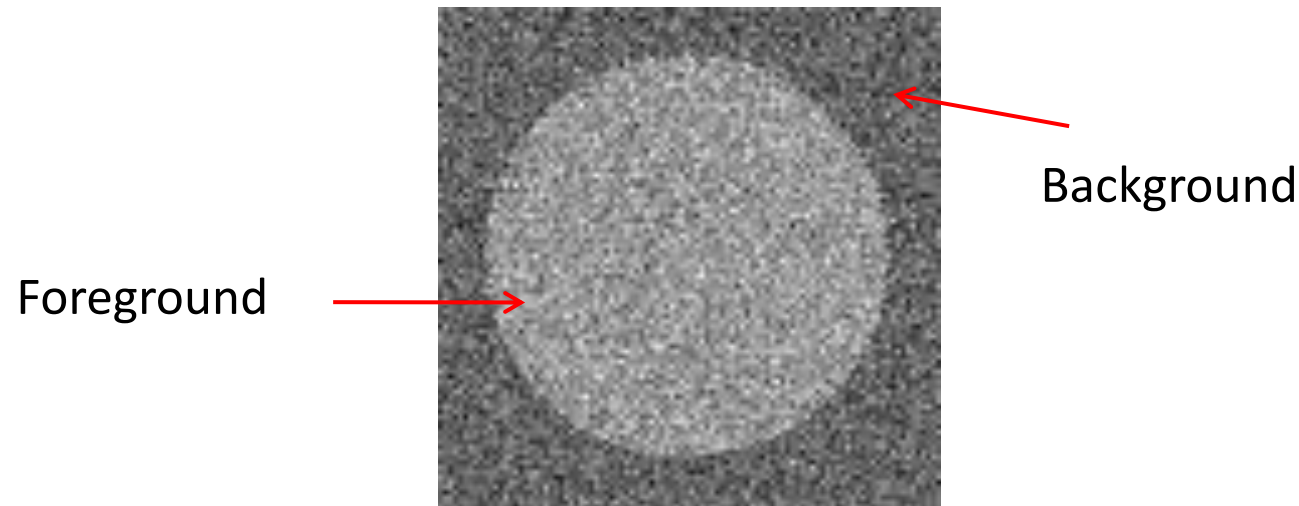


labels

```
>> mu_fg = mean(im(labels))  
      mu_fg = 0.6012  
>> sigma_fg = sqrt(mean((im(labels)-mu_fg).^2))  
      sigma_fg = 0.1007  
>> mu_bg = mean(im(~labels))  
      mu_bg = 0.4007  
>> sigma_bg = sqrt(mean((im(~labels)-mu_bg).^2))  
      sigma_bg = 0.1007  
>> pfg = mean(labels(:));
```

Probabilistic Inference


2. Once we have modeled the fg/bg appearance, how do we compute the likelihood that a pixel is foreground?



Probabilistic Inference

Compute the likelihood that a particular model
generated a sample

component or label


$$p(z_n = m \mid x_n, \theta)$$

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

component or label

$$p(z_n = m \mid x_n, \theta) = \frac{p(z_n = m, x_n \mid \theta_m)}{p(x_n \mid \theta)}$$

← Conditional probability

$$P(A|B) = \frac{P(A, B)}{P(B)}$$

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

component or label

$$\downarrow$$
$$p(z_n = m \mid x_n, \theta) = \frac{p(z_n = m, x_n \mid \theta_m)}{p(x_n \mid \theta)}$$

$$= \frac{p(z_n = m, x_n \mid \theta_m)}{\sum_k p(z_n = k, x_n \mid \theta_k)}$$

← Marginalization

$$P(A) = \sum_k P(A, B = k)$$

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

component or label

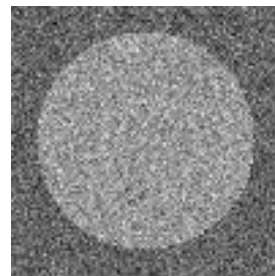
$$p(z_n = m | x_n, \theta) = \frac{p(z_n = m, x_n | \theta_m)}{p(x_n | \theta)}$$

$$= \frac{p(z_n = m, x_n | \theta_m)}{\sum_k p(z_n = k, x_n | \theta_k)}$$

Joint distribution
 $P(A, B) = P(B)P(A|B)$

$$= \frac{p(x_n | z_n = m, \theta_m) p(z_n = m | \theta_m)}{\sum_k p(x_n | z_n = k, \theta_k) p(z_n = k | \theta_k)}$$

Example: Inference



im

Learned Parameters

fg: $\mu=0.6$, $\sigma=0.1$

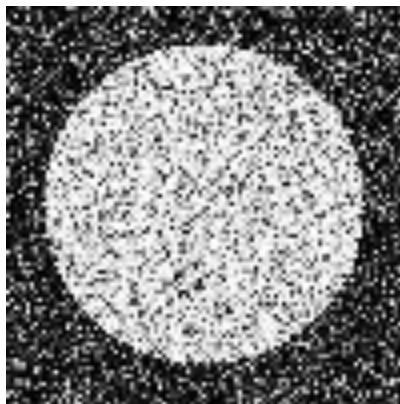
bg: $\mu=0.4$, $\sigma=0.1$

```
>> pfg = 0.5;
```

```
>> px_fg = normpdf(im, mu_fg, sigma_fg);
```

```
>> px_bg = normpdf(im, mu_bg, sigma_bg);
```

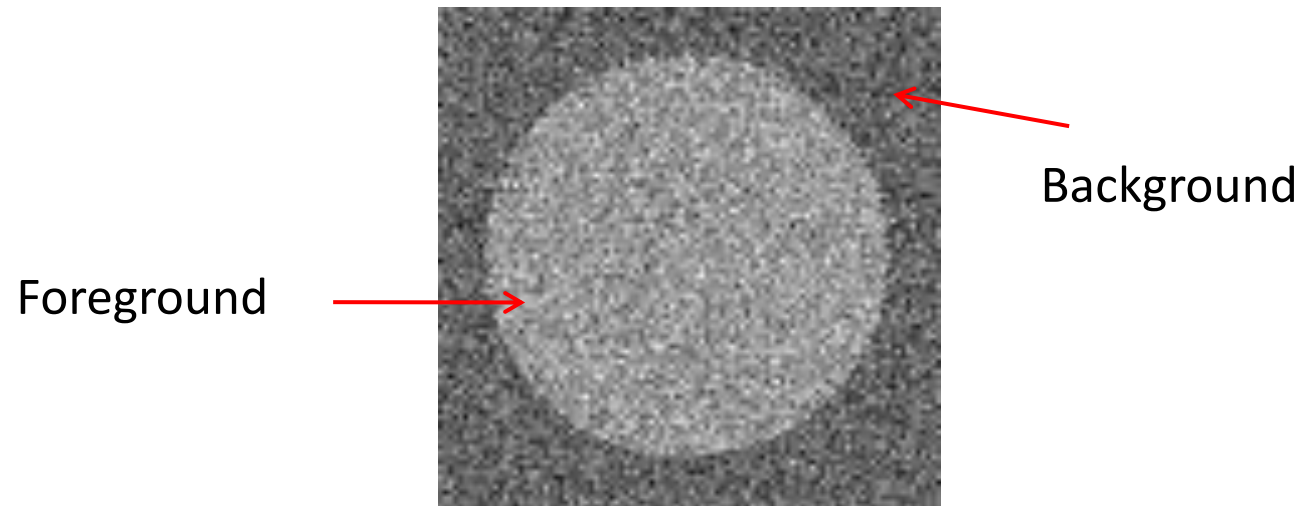
```
>> pfg_x = px_fg*pfg ./ (px_fg*pfg + px_bg*(1-pfg));
```



$p(\text{fg} | \text{im})$

Dealing with Hidden Variables

3. How can we get both labels and appearance parameters at once?



Mixture of Gaussians

component model parameters component prior mixture component

$$p(x_n | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\pi}) = \sum_m p(x_n, z_n = m | \mu_m, \sigma_m^2, \pi_m)$$

$$\begin{aligned} p(x_n, z_n = m | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\pi}) &= p(x_n, z_n = m | \mu_m, \sigma_m^2, \pi_m) \\ &= p(x_n | \mu_m, \sigma_m^2) p(z_n = m | \pi_m) \\ &= \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{(x_n - \mu_m)^2}{2\sigma_m^2}\right) \cdot \pi_m \end{aligned}$$

Mixture of Gaussians

With enough components, can represent any probability density function

- Widely used as general purpose pdf estimator

Segmentation with Mixture of Gaussians

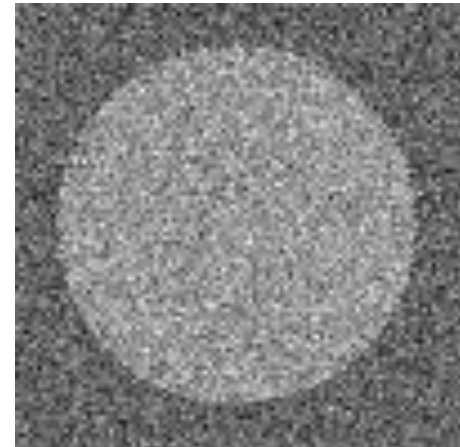
Pixels come from one of several Gaussian components

- We don't know which pixels come from which components
- We don't know the parameters for the components

Problem:

- Estimate the parameters of the Gaussian Mixture Model.

What would you do?



Simple solution

1. Initialize parameters
2. Compute the probability of each hidden variable given the current parameters
3. Compute new parameters for each model, weighted by likelihood of hidden variables
4. Repeat 2-3 until convergence

Mixture of Gaussians: Simple Solution

1. Initialize parameters
2. Compute likelihood of hidden variables for current parameters

$$\alpha_{nm} = p(z_n = m | x_n, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \boldsymbol{\pi}^{(t)})$$

3. Estimate new parameters for each model, weighted by likelihood

$$\hat{\mu}_m^{(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} x_n \quad \hat{\sigma}_m^{2(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} (x_n - \hat{\mu}_m)^2 \quad \hat{\pi}_m^{(t+1)} = \frac{\sum_n \alpha_{nm}}{N}$$

Expectation Maximization (EM) Algorithm

$$\text{Goal: } \hat{\theta} = \operatorname{argmax}_{\theta} \log \left(\sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} \mid \theta) \right)$$

↑
Log of sums is intractable

Jensen's Inequality $f(E[X]) \geq E[f(X)]$ for concave functions $f(x)$
(so we maximize the lower bound!)

Maximum Likelihood from Incomplete Data Via the **EM Algorithm**

[AP Dempster, NM Laird...](#) - Journal of the Royal ..., 1977 - Wiley Online Library

A broadly applicable **algorithm** for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the **algorithm** is derived. Many examples are sketched ...

☆ ⓘ Cited by 54643 Related articles All 61 versions Web of Science: 23929 Import into BibTeX

See here for proof: www.stanford.edu/class/cs229/notes/cs229-notes8.ps

Expectation Maximization (EM) Algorithm

$$\text{Goal: } \hat{\theta} = \operatorname{argmax}_{\theta} \log \left(\sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} \mid \theta) \right)$$

1. E-step: compute

$$\mathbb{E}_{\mathbf{z} \mid \mathbf{x}, \theta^{(t)}} [\log(p(\mathbf{x}, \mathbf{z} \mid \theta))] = \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} \mid \theta)) p(\mathbf{z} \mid \mathbf{x}, \theta^{(t)})$$

2. M-step: solve

$$\theta^{(t+1)} = \operatorname{argmax}_{\theta} \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} \mid \theta)) p(\mathbf{z} \mid \mathbf{x}, \theta^{(t)})$$

log of expectation of $P(\mathbf{x}|\mathbf{z})$

Goal: $\hat{\theta} = \operatorname{argmax}_{\theta} \log \left(\sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} | \theta) \right) \quad f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$

1. E-step: compute

expectation of log of $P(\mathbf{x}|\mathbf{z})$

$$\mathbb{E}_{\mathbf{z}|\mathbf{x}, \theta^{(t)}} [\log(p(\mathbf{x}, \mathbf{z} | \theta))] = \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} | \theta)) p(\mathbf{z} | \mathbf{x}, \theta^{(t)})$$

2. M-step: solve

$$\theta^{(t+1)} = \operatorname{argmax}_{\theta} \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} | \theta)) p(\mathbf{z} | \mathbf{x}, \theta^{(t)})$$

EM for Mixture of Gaussians - derivation

$$p(x_n | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\pi}) = \sum_m p(x_n, z_n = m | \mu_m, \sigma_m^2, \pi_m) = \sum_m \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{(x_n - \mu_m)^2}{\sigma_m^2}\right) \cdot \pi_m$$

1. E-step: $E_{z|x, \theta^{(t)}} [\log(p(\mathbf{x}, \mathbf{z} | \theta))] = \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} | \theta)) p(\mathbf{z} | \mathbf{x}, \theta^{(t)})$
2. M-step: $\theta^{(t+1)} = \operatorname{argmax}_{\theta} \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} | \theta)) p(\mathbf{z} | \mathbf{x}, \theta^{(t)})$

EM for Mixture of Gaussians

$$p(x_n | \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\pi}) = \sum_m p(x_n, z_n = m | \mu_m, \sigma_m^2, \pi_m) = \sum_m \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{(x_n - \mu_m)^2}{\sigma_m^2}\right) \cdot \pi_m$$

1. E-step: $E_{z|x, \theta^{(t)}} [\log(p(\mathbf{x}, \mathbf{z} | \theta))] = \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} | \theta)) p(\mathbf{z} | \mathbf{x}, \theta^{(t)})$

2. M-step: $\theta^{(t+1)} = \operatorname{argmax}_{\theta} \sum_{\mathbf{z}} \log(p(\mathbf{x}, \mathbf{z} | \theta)) p(\mathbf{z} | \mathbf{x}, \theta^{(t)})$

$$\alpha_{nm} = p(z_n = m | x_n, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \boldsymbol{\pi}^{(t)})$$

$$\hat{\mu}_m^{(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} x_n \quad \hat{\sigma}_m^{2(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} (x_n - \hat{\mu}_m)^2 \quad \hat{\pi}_m^{(t+1)} = \frac{\sum_n \alpha_{nm}}{N}$$

EM algorithm - derivation

$$p(\mathbf{x}|\Theta) = \sum_{i=1}^M \alpha_i p_i(\mathbf{x}|\theta_i)$$

$$\log(\mathcal{L}(\Theta|\mathcal{X})) = \log \prod_{i=1}^N p(x_i|\Theta) = \sum_{i=1}^N \log \left(\sum_{j=1}^M \alpha_j p_j(x_i|\theta_j) \right)$$

$$\log(\mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y})) = \log(P(\mathcal{X}, \mathcal{Y}|\Theta)) = \sum_{i=1}^N \log(P(x_i|y_i)P(y_i)) = \sum_{i=1}^N \log(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i}))$$

$$p(y_i|x_i, \Theta^g) = \frac{\alpha_{y_i}^g p_{y_i}(x_i|\theta_{y_i}^g)}{p(x_i|\Theta^g)} = \frac{\alpha_{y_i}^g p_{y_i}(x_i|\theta_{y_i}^g)}{\sum_{k=1}^M \alpha_k^g p_k(x_i|\theta_k^g)}$$

$$p(\mathbf{y}|\mathcal{X}, \Theta^g) = \prod_{i=1}^N p(y_i|x_i, \Theta^g)$$

EM algorithm – E-Step

$$\begin{aligned}
 Q(\Theta, \Theta^g) &= \sum_{\mathbf{y} \in \Upsilon} \log(\mathcal{L}(\Theta | \mathcal{X}, \mathbf{y})) p(\mathbf{y} | \mathcal{X}, \Theta^g) \\
 &= \sum_{\mathbf{y} \in \Upsilon} \sum_{i=1}^N \log(\alpha_{y_i} p_{y_i}(x_i | \theta_{y_i})) \prod_{j=1}^N p(y_j | x_j, \Theta^g) \\
 &= \sum_{y_1=1}^M \sum_{y_2=1}^M \dots \sum_{y_N=1}^M \sum_{i=1}^N \log(\alpha_{y_i} p_{y_i}(x_i | \theta_{y_i})) \prod_{j=1}^N p(y_j | x_j, \Theta^g) \\
 &= \sum_{y_1=1}^M \sum_{y_2=1}^M \dots \sum_{y_N=1}^M \sum_{i=1}^N \sum_{\ell=1}^M \delta_{\ell, y_i} \log(\alpha_{\ell} p_{\ell}(x_i | \theta_{\ell})) \prod_{j=1}^N p(y_j | x_j, \Theta^g) \\
 &= \sum_{\ell=1}^M \sum_{i=1}^N \log(\alpha_{\ell} p_{\ell}(x_i | \theta_{\ell})) \underbrace{\sum_{y_1=1}^M \sum_{y_2=1}^M \dots \sum_{y_N=1}^M \delta_{\ell, y_i} \prod_{j=1}^N p(y_j | x_j, \Theta^g)}_{p(\ell | x_i, \Theta^g)}
 \end{aligned}$$

EM algorithm – E-Step

$$\begin{aligned} Q(\Theta, \Theta^g) &= \sum_{\ell=1}^M \sum_{i=1}^N \log(\alpha_{\ell} p_{\ell}(x_i | \theta_{\ell})) p(\ell | x_i, \Theta^g) \\ &= \sum_{\ell=1}^M \sum_{i=1}^N \log(\alpha_{\ell}) p(\ell | x_i, \Theta^g) + \sum_{\ell=1}^M \sum_{i=1}^N \log(p_{\ell}(x_i | \theta_{\ell})) p(\ell | x_i, \Theta^g) \end{aligned}$$

EM algorithm – M-Step

$$\frac{\partial}{\partial \alpha_\ell} \left[\sum_{\ell=1}^M \sum_{i=1}^N \log(\alpha_\ell) p(\ell | x_i, \Theta^g) + \lambda \left(\sum_{\ell} \alpha_\ell - 1 \right) \right] = 0$$

$$\sum_{i=1}^N \frac{1}{\alpha_\ell} p(\ell | x_i, \Theta^g) + \lambda = 0$$

$$\alpha_\ell = \frac{1}{N} \sum_{i=1}^N p(\ell | x_i, \Theta^g)$$

EM algorithm – M-Step

$$\begin{aligned} & \sum_{\ell=1}^M \sum_{i=1}^N \log(p_{\ell}(x_i | \mu_{\ell}, \Sigma_{\ell})) p(\ell | x_i, \Theta^g) \\ &= \sum_{\ell=1}^M \sum_{i=1}^N \left(-\frac{1}{2} \log(|\Sigma_{\ell}|) - \frac{1}{2} (x_i - \mu_{\ell})^T \Sigma_{\ell}^{-1} (x_i - \mu_{\ell}) \right) p(\ell | x_i, \Theta^g) \end{aligned}$$

Take derivative with respect to μ_l

$$\sum_{i=1}^N \Sigma_{\ell}^{-1} (x_i - \mu_{\ell}) p(\ell | x_i, \Theta^g) = 0$$

$$\mu_{\ell} = \frac{\sum_{i=1}^N x_i p(\ell | x_i, \Theta^g)}{\sum_{i=1}^N p(\ell | x_i, \Theta^g)}$$

EM algorithm – M-Step

Take derivative with respect to Σ_ℓ^{-1}

$$\Sigma_\ell = \frac{\sum_{i=1}^N p(\ell|x_i, \Theta^g) N_{\ell,i}}{\sum_{i=1}^N p(\ell|x_i, \Theta^g)} = \frac{\sum_{i=1}^N p(\ell|x_i, \Theta^g) (x_i - \mu_\ell)(x_i - \mu_\ell)^T}{\sum_{i=1}^N p(\ell|x_i, \Theta^g)}$$

EM Algorithm for GMM

$$\alpha_{\ell}^{new} = \frac{1}{N} \sum_{i=1}^N p(\ell|x_i, \Theta^g)$$

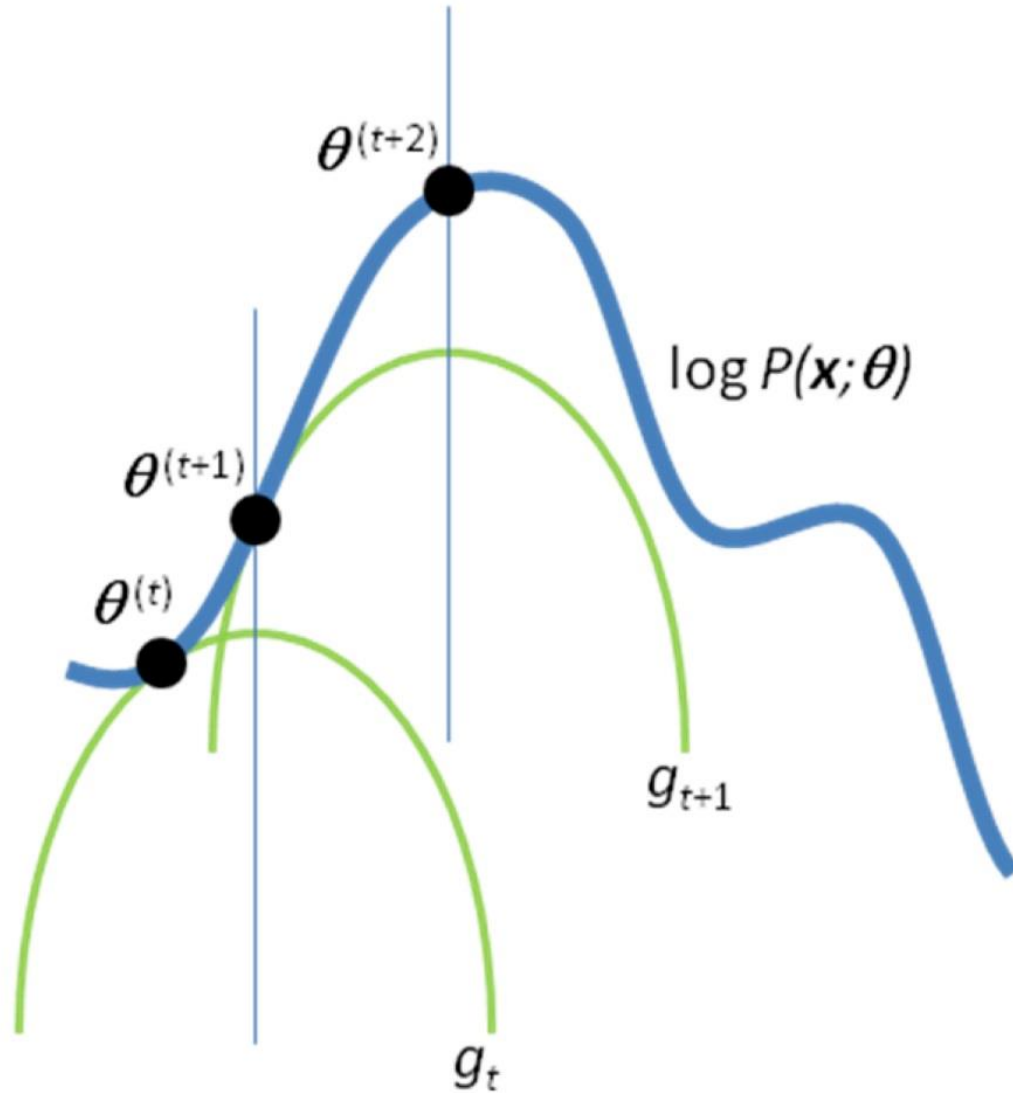
$$\mu_{\ell}^{new} = \frac{\sum_{i=1}^N x_i p(\ell|x_i, \Theta^g)}{\sum_{i=1}^N p(\ell|x_i, \Theta^g)}$$

$$\Sigma_{\ell}^{new} = \frac{\sum_{i=1}^N p(\ell|x_i, \Theta^g) (x_i - \mu_{\ell}^{new})(x_i - \mu_{\ell}^{new})^T}{\sum_{i=1}^N p(\ell|x_i, \Theta^g)}$$

EM Algorithm

- Maximizes a lower bound on the data likelihood at each iteration
- Each step increases the data likelihood
 - Converges to *local maximum*
- Common tricks to derivation
 - Find terms that sum or integrate to 1
 - Lagrange multiplier to deal with constraints

Convergence of EM Algorithm



“Hard EM”

- Same as EM except compute z^* as most likely values for hidden variables
- K-means is an example
- Advantages
 - Simpler: can be applied when cannot derive EM
 - Sometimes works better if you want to make hard predictions at the end
- But
 - Generally, pdf parameters are not as accurate as EM