Hidden Variables, the EM Algorithm, and Mixtures of Gaussians

Computer Vision

CS 543 / ECE 549

University of Illinois

Derek Hoiem

Today's Class

- Examples of Missing Data Problems
  - Detecting outliers
  - Latent topic models (HW 2, problem 3)
  - Segmentation (HW 2, problem 4)
- 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

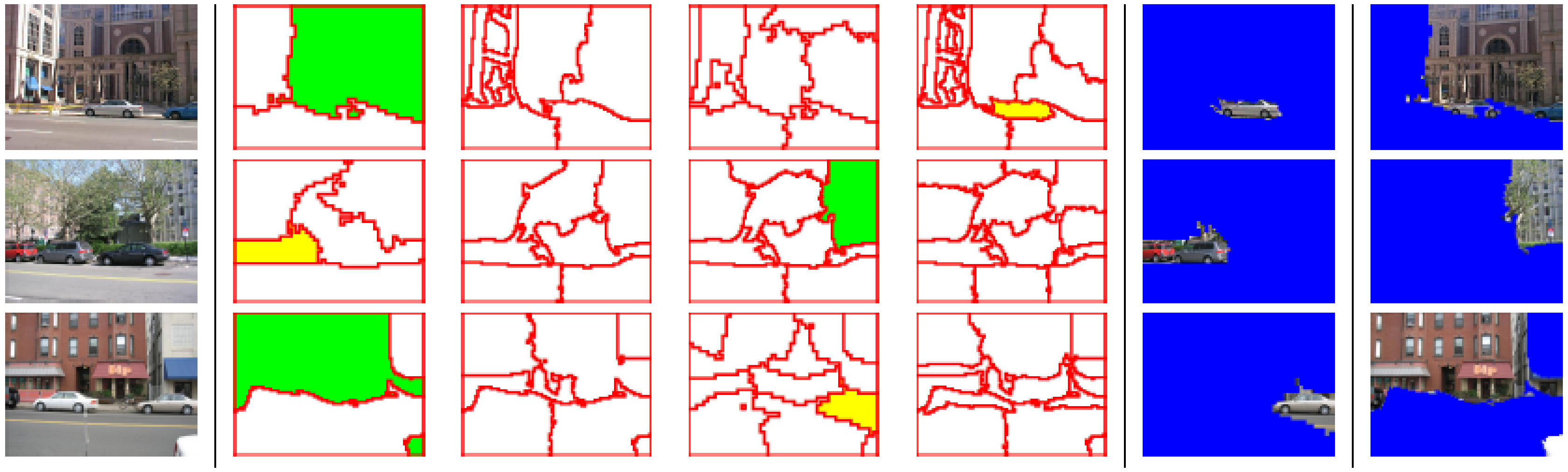
10
8
9
2
8

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.

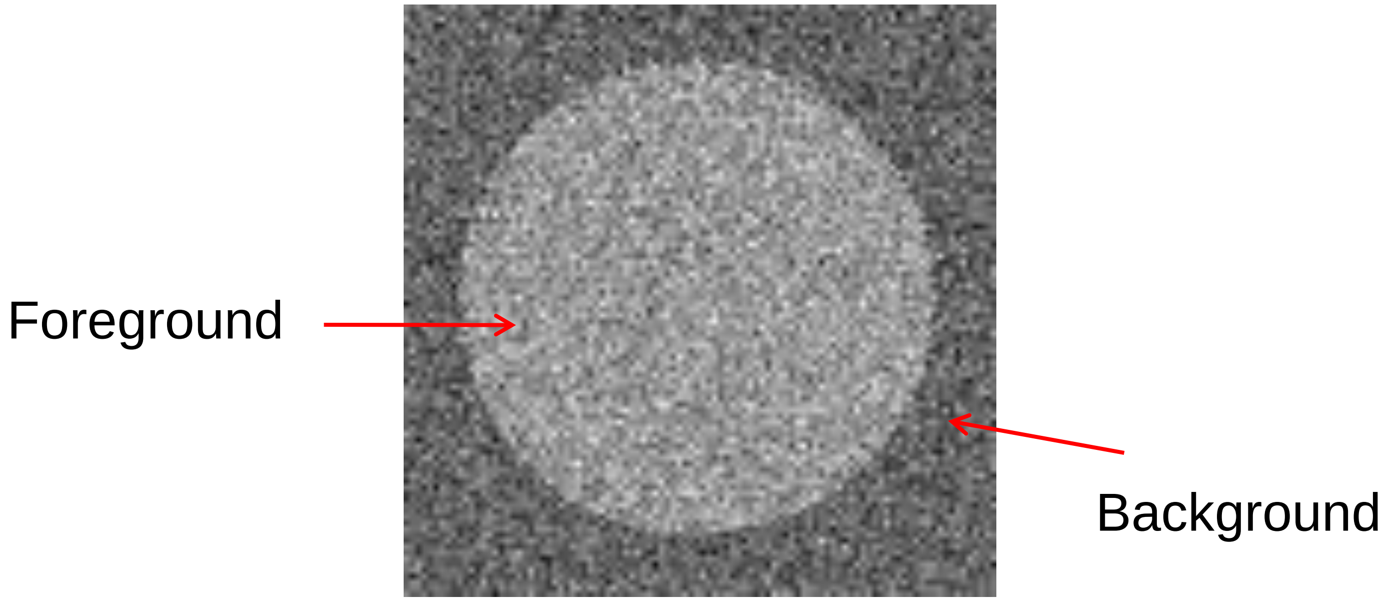


http://www.robots.ox.ac.uk/~vgg/publications/papers/russell06.pdf

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.

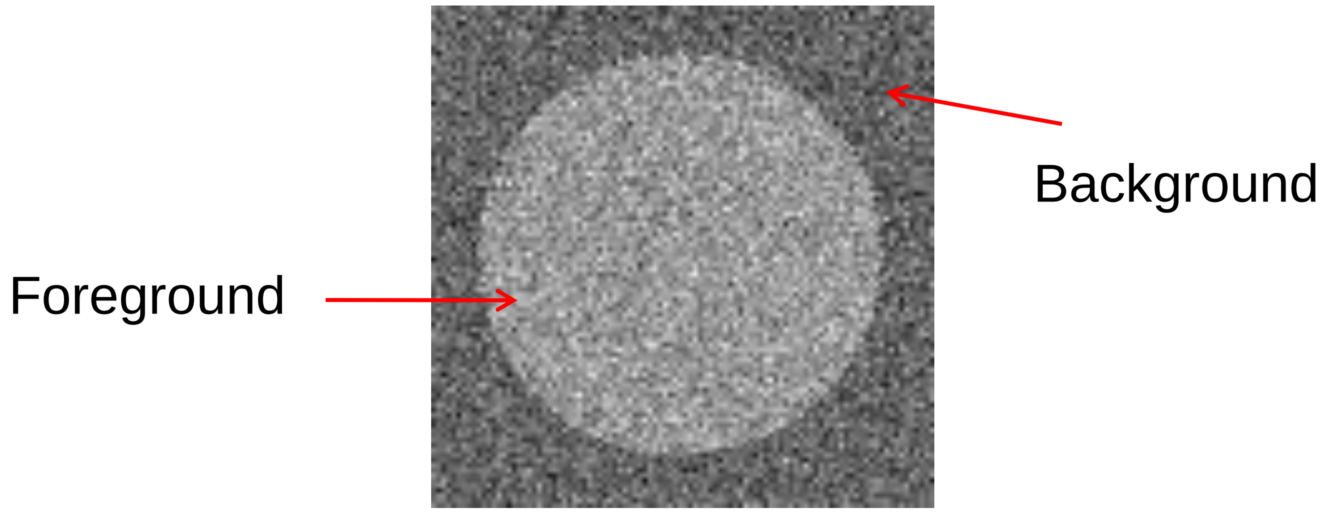


Missing Data Problems: Segmentation

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

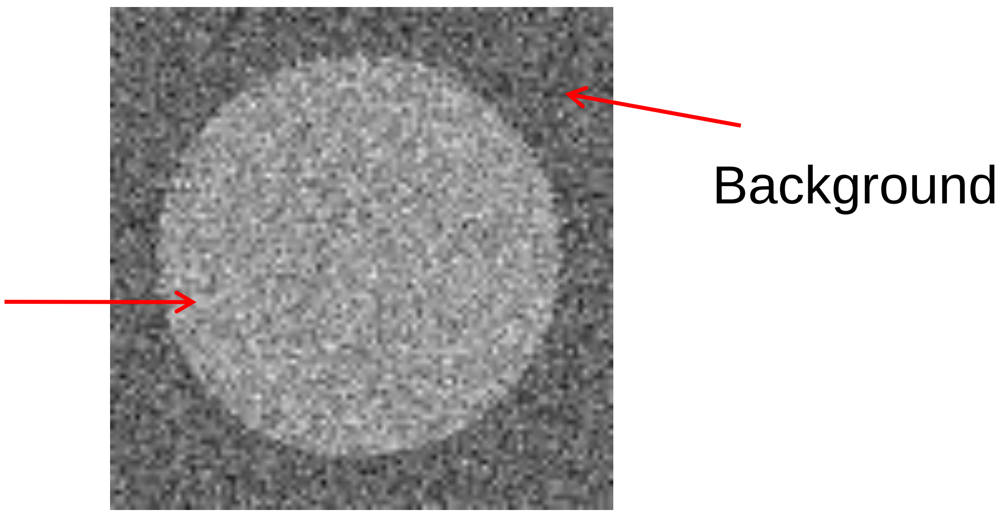
Three steps:

1. If we had labels, how could we model the appearance of foreground and background?
2. Once we have modeled the fg/bg appearance, how do we compute the likelihood that a pixel is foreground?
3. How can we get both labels and appearance models at once?



Maximum Likelihood Estimation

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



Maximum Likelihood Estimation

data

$$\mathbf{x} = \left\{ x_1 .. x_N \right\}$$

parameters

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

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

Maximum Likelihood Estimation

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

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

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

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

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

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

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

Gaussian Distribution

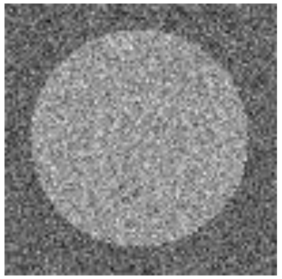
$$p(x_n \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n - \mu)^2}{2\sigma^2} \right)$$

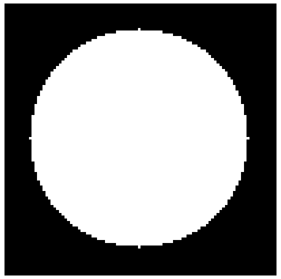
$$\hat{\mu} = \frac{1}{N} \sum_n x_n \qquad \hat{\sigma}^2 = \frac{1}{N} \sum_n (x_n - \hat{\mu})^2$$

Example: MLE

Parameters used to Generate

```
fg: mu=0.6, sigma=0.1
bg: mu=0.4, sigma=0.1
```

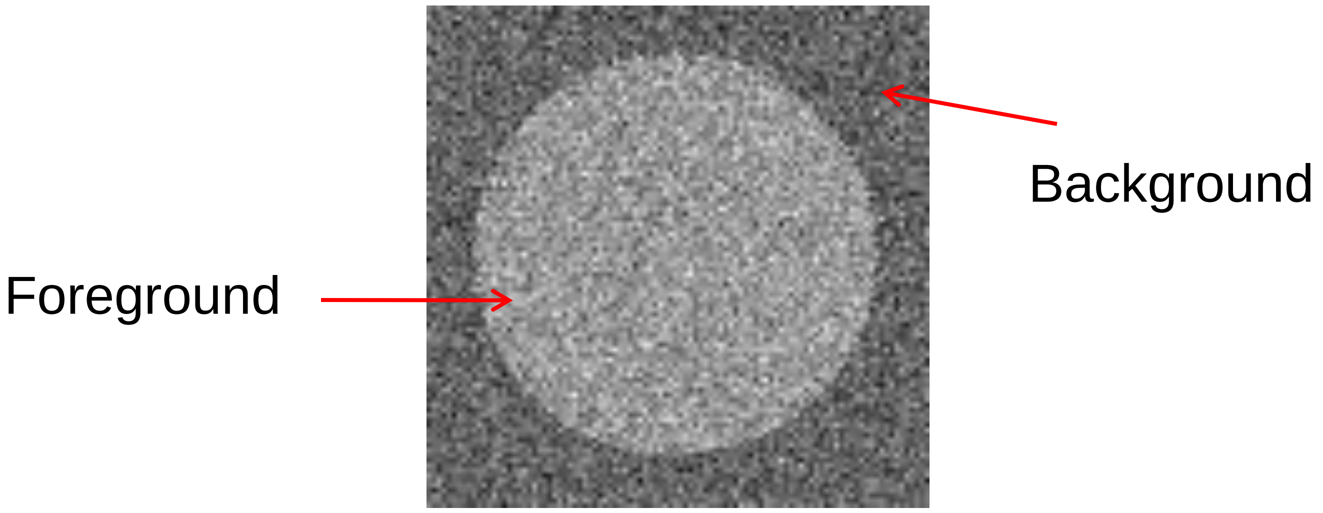

im


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

$$p(\overset{\text{component or label}}{\overset{\searrow}{z_n}} = m \mid x_n, \theta)$$

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

$$p(\overset{\substack{\text{component or label}\\ \searrow}}{z_n} = m \mid x_n, \theta) = \frac{p\left(z_n = m, x_n \mid \theta_m\right)}{p\left(x_n \mid \theta\right)}$$

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

component or label

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

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

Probabilistic Inference

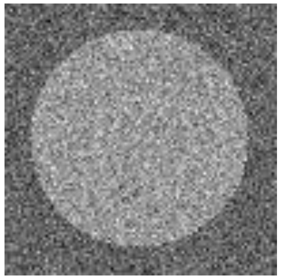
Compute the likelihood that a particular model generated a sample

$$p(\overset{\substack{\text{component or label}\\ \searrow}}{z_n} = m \mid x_n, \theta) = \frac{p\left(z_n = m, x_n \mid \theta_m\right)}{p\left(x_n \mid \theta\right)}$$

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

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

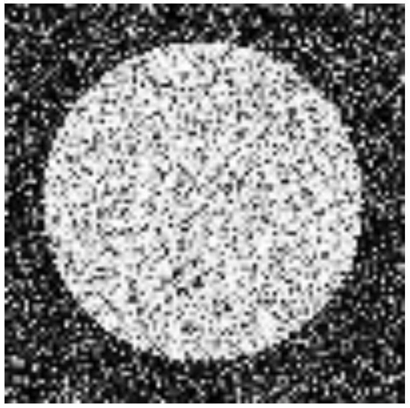
Example: Inference


im

Learned Parameters

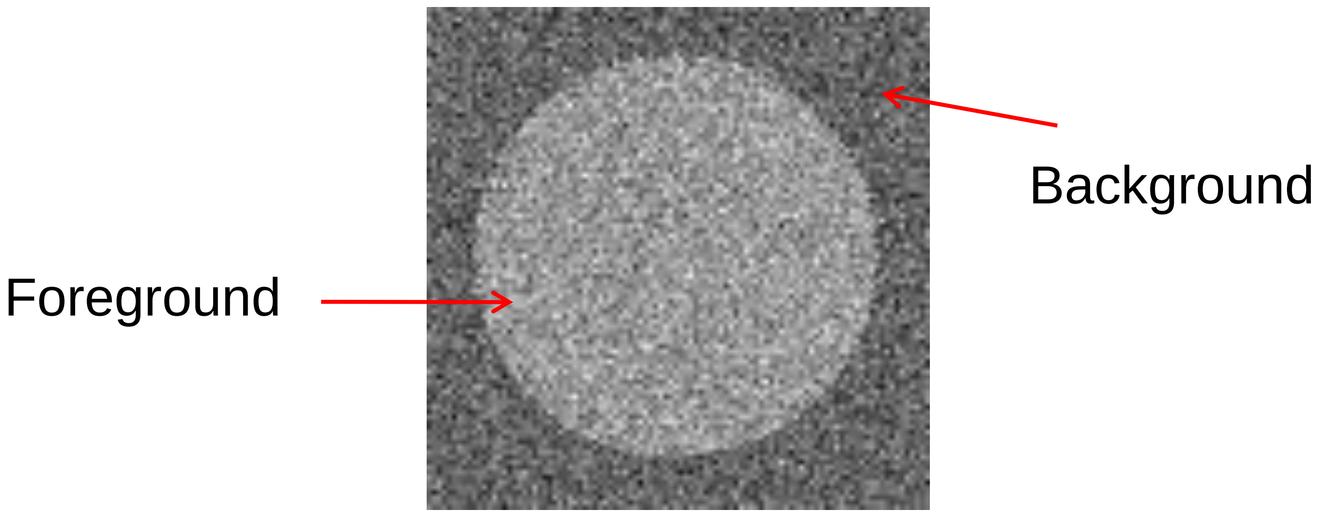
```
fg: mu=0.6, sigma=0.1
bg: mu=0.4, sigma=0.1
```

```
>> 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(fg | 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

$$p\left(x_n \mid \underbrace{\boldsymbol{\mu}, \boldsymbol{\sigma}^2}_{\text{component model parameters}}, \underbrace{\boldsymbol{\pi}}_{\text{component prior}}\right) = \sum_m \underbrace{p\left(x_n, z_n = m \mid \mu_m, \sigma_m^{\,2}, \pi_m\right)}_{\text{mixture component}}$$

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

Mixture of Gaussians

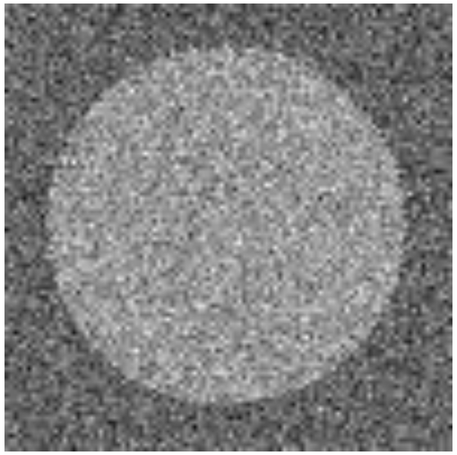
With enough components, can represent any probability density function

- 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



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

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

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

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

Expectation Maximization (EM) Algorithm

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

$\nwarrow$ Log of sums is intractable

Jensen's Inequality

$$f\left(\mathrm{E}[X]\right) \geq \mathrm{E}\left[f(X)\right]$$

for concave functions f(x)

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

Expectation Maximization (EM) Algorithm

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

1. E-step: compute

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

2. M-step: solve

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

Expectation Maximization (EM) Algorithm

log of expectation of P(x|z)

Goal: $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \log\left(\sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} \mid \theta) \right)$ $\qquad f(\mathrm{E}[X]) \geq \mathrm{E}[f(X)]$

1. E-step: compute

expectation of log of P(x|z)

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

2. M-step: solve

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

EM for Mixture of Gaussians (by hand)

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

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

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

EM for Mixture of Gaussians (by hand)

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

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

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

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

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

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

EM Demos

- Mixture of Gaussian demo
- Simple segmentation demo

"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

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.

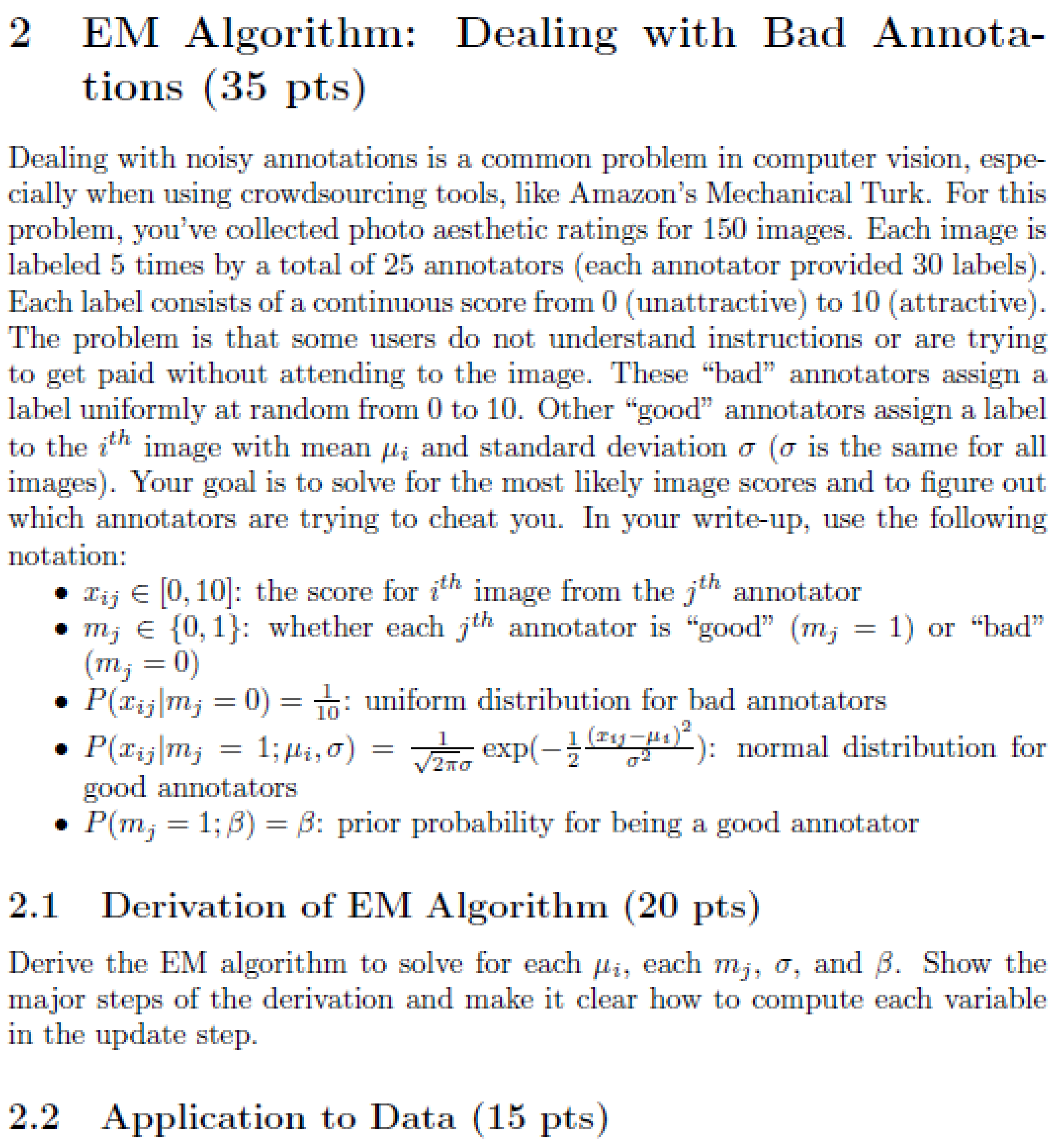
Challenge: Determine which people to trust and the average rating by accurate annotators.

Annotator Ratings

10
8
9
2
8

Photo: Jam343 (Flickr)

HW 4, problem 2



2 EM Algorithm: Dealing with Bad Annotations (35 pts)

Dealing with noisy annotations is a common problem in computer vision, especially when using crowdsourcing tools, like Amazon's Mechanical Turk. For this problem, you've collected photo aesthetic ratings for 150 images. Each image is labeled 5 times by a total of 25 annotators (each annotator provided 30 labels). Each label consists of a continuous score from 0 (unattractive) to 10 (attractive). The problem is that some users do not understand instructions or are trying to get paid without attending to the image. These "bad" annotators assign a label uniformly at random from 0 to 10. Other "good" annotators assign a label to the i^{th} image with mean μ_i and standard deviation σ (σ is the same for all images). Your goal is to solve for the most likely image scores and to figure out which annotators are trying to cheat you. In your write-up, use the following notation:

- $x_{ij} \in [0, 10]$: the score for i^{th} image from the j^{th} annotator
- $m_j \in \{0, 1\}$: whether each j^{th} annotator is "good" ($m_j = 1$) or "bad" ($m_j = 0$)
- $P(x_{ij}|m_j = 0) = \frac{1}{10}$: uniform distribution for bad annotators
- $P(x_{ij}|m_j = 1; \mu_i, \sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2}\frac{(x_{ij}-\mu_i)^2}{\sigma^2})$: normal distribution for good annotators
- $P(m_j = 1; \beta) = \beta$: prior probability for being a good annotator

2.1 Derivation of EM Algorithm (20 pts)

Derive the EM algorithm to solve for each μ_i, each m_j, σ, and β. Show the major steps of the derivation and make it clear how to compute each variable in the update step.

2.2 Application to Data (15 pts)

The false scores come from a uniform distribution

The true scores for each image have a Gaussian distribution

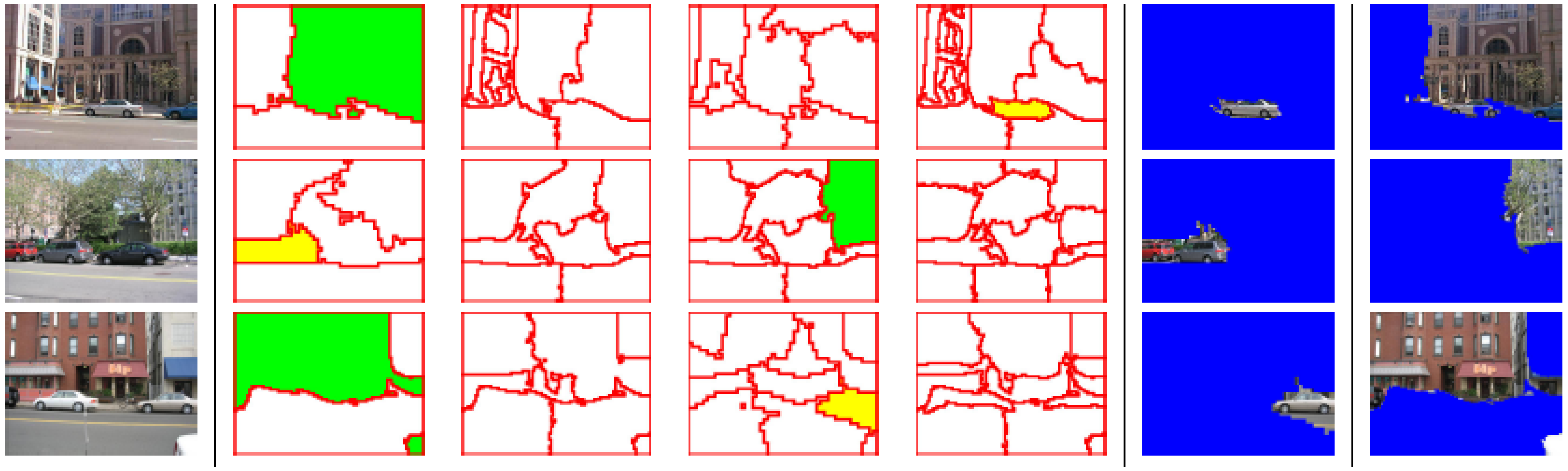
Annotators are always "bad" or always "good"

The "good/bad" label of each annotator is the missing data

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.



http://www.robots.ox.ac.uk/~vgg/publications/papers/russell06.pdf

Next class

- MRFs and Graph-cut Segmentation