Gaussian Mixture Model (GMM)

- Generative process
  - Chose a mixture component (cluster), *m*, with probability *p(m)*
  - For the component m, consult the particular Gaussian distribution
  - Generate a sample from that distribution
- This models the distribution

$$p(\mathbf{x}) = \sum_m p(m)p(\mathbf{x} \mid m) \qquad \text{where} \qquad p(\mathbf{x} \mid m) = \mathbb{N}(\boldsymbol{\mu}_m, \Sigma)$$

$$\Theta = \{\Theta_m\}$$

- And for multiple points

$$p(\{\mathbf{x}_i\}) = \prod_i \left(\sum_m p(m)p(\mathbf{x} \mid m) \right)$$

Segmentation/Grouping by EM

- Since we don't know which point comes from which segment, we have to use an **estimate** of the **probabilities** that a given point belongs to a given segment.
- Formally, these probabilities can be denoted $p(m \mid \mathbf{x}_i, \Theta^{(m)})$
- This is the probability that $\mathbf{x}_i$ is in cluster *m*, given the model
- If we assume we know these estimates for the probabilities of the missing values, we can then estimate the means of the Gaussians for each segment.
- Specifically, we compute means and variances by **weighting** the standard formulas by these probabilities.

Segmentation/Grouping by EM

- We estimate the mean for each segment by:

Iteration (step)

$$\mu_m^{(s+1)} = \frac{\sum_{i=1}^r \mathbf{x}_i p(m \mid \mathbf{x}_i, \Theta_m^{(s)})}{\sum_{i=1}^r p(m \mid \mathbf{x}_i, \Theta_m^{(s)})}$$

- Variances/covariances work similarly

We can sort out the chicken!

Details optional

Segmentation/Grouping--E Step

- Given parameters, the probability that a given point is associated with each cluster is can be computed by:

$$p(m \mid \mathbf{x}_l, \Theta_m^{(s)}) = \frac{\alpha_m^{(s)} p(\mathbf{x}_l \mid \theta_m^{(s)})}{\sum_{k=1}^M \alpha_k^{(s)} p(\mathbf{x}_l \mid \theta_k^{(s)})} \qquad (s \text{ indexes iterations})$$

- The book uses $\overline{I_{lm}}$ for $p(m \mid \mathbf{x}_i, \Theta^{(s)})$ (*I* suggests "indicator variable")
- (Also, my copy of the book's version of the above equation looks wrong to me--the index *l* applies to points and the index for theta should refer to groups)

Details optional

$\alpha_m^{(s)} = p(m)$ (Standard notation)

$p(m \mid \mathbf{x}_l, \Theta^{(s)}) = \frac{p(\mathbf{x}_l \mid m, \theta_m^{(s)}) p(m)}{p(\mathbf{x}_l \mid \theta_m^{(s)})}$ (Bayes)

$p(\mathbf{x}_l \mid \theta_m^{(s)}) = \sum_{k=1}^{M} p(\mathbf{x}_l, m, \theta_k^{(s)})$ (Marginalization)

$p(\mathbf{x}_l \mid \theta_m^{(s)}) = \sum_{k=1}^{M} p(m) p(\mathbf{x}_l \mid m, \theta_k^{(s)})$ (Definition of "|")

Therefore

$p(m \mid \mathbf{x}_l, \Theta^{(s)}) = \frac{\alpha_m^{(s)} p(\mathbf{x}_l \mid \theta_m^{(s)})}{\sum_{k=1}^{M} \alpha_k^{(s)} p(\mathbf{x}_l \mid \theta_k^{(s)})}$

We can do the egg!

Segmentation/Grouping by EM

- This is a lot like K-means
- Instead of binary cluster membership, each point has some probability of being in each cluster
- In addition to computing means, we generally also compute variances
  - Setting all variances equal in advance ("tied parameters") is simplest, but often having different variances is important.
  - Can fit different variances to each cluster (most common)
  - Can fit covariance matrices instead of variances (usually not possible if the dimension is over five or so)

Segmentation with EM

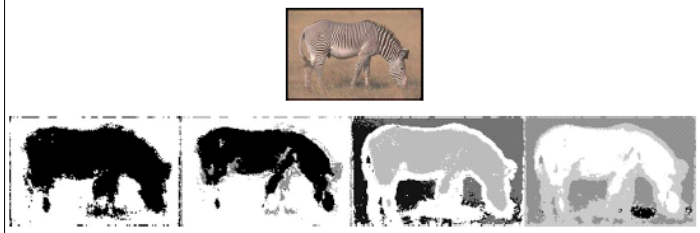


Figure from "Color and Texture Based Image Segmentation Using EM and Its Application to Content Based Image Retrieval",S.J. Belongie et al., Proc. Int. Conf. Computer Vision, 1998, c1998, IEEE

Motion segmentation with EM (one)

- Recall the baby on the couch
- Alternative algorithms based on previous examples?

Motion segmentation with EM (one)

- Can treat background/foreground assignment as missing values!

Motion segmentation with EM (two)

- Model image sequence as consisting of regions (layers) of parametric motion
  - For example, affine motion is popular

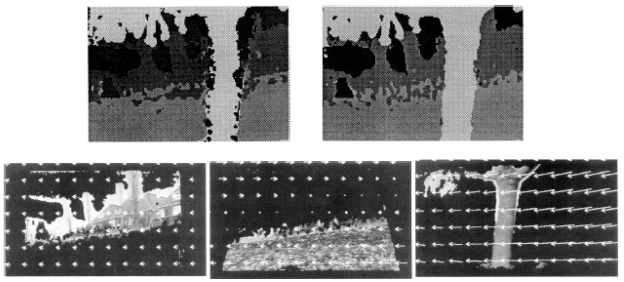
$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

- Now we need to
  - Determine which pixels belong to which region
  - Estimate parameters
  - Yet another example of a missing value problem!

Three frames from the MPEG "flower garden" sequence

Figure from "Representing Images with layers,", by J. Wang and E.H. Adelson, IEEE Transactions on Image Processing, 1994, c 1994, IEEE

Grey level shows region no. with highest probability



Segments and motion fields associated with them

Figure from "Representing Images with layers,", by J. Wang and E.H. Adelson, IEEE Transactions on Image Processing, 1994, c 1994, IEEE

If we use multiple frames to estimate the appearance of a segment, we can fill in occlusions; so we can re-render the sequence with some segments removed.

Figure from "Representing Images with layers,", by J. Wang and E.H. Adelson, IEEE Transactions on Image Processing, 1994, c 1994, IEEE

RANSAC versus EM

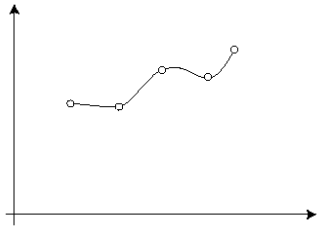
- Many, but not all problems that can be attacked with EM can also be attacked with RANSAC
  - For RANSAC, we need to be able to get a parameter estimate with a manageably small number of random choices.
  - RANSAC is often better

Important

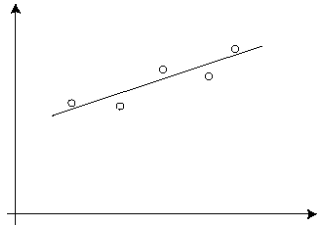
Model Selection

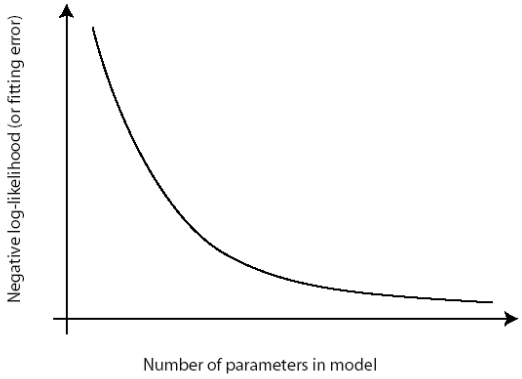
- In general, models with more parameters will fit a dataset better, but are poorer at prediction
- This means we can't simply look at the negative log-likelihood (or fitting error)

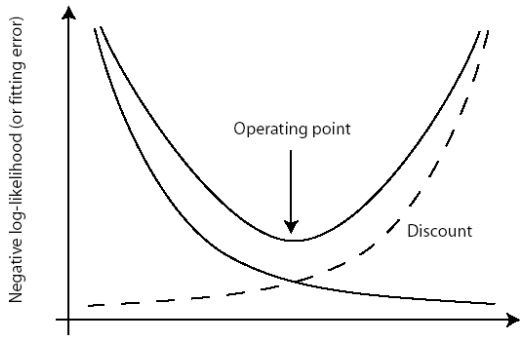
Important



Top is not necessarily a better fit than bottom
(actually, almost always worse)







We can discount the fitting error with some term in the number of parameters in the model.

Discounts

- Let N be the number of data points, p the number of parameters
- AIC (an information criterion)
  - choose model with smallest value of $-2L(D;\theta^*)+2p$
- BIC (Bayes information criterion)
  - choose model with smallest value of $-2L(D;\theta^*)+p\log N$
- Minimum description length
  - same criterion as BIC, but derived in a completely different way

Important

Cross-validation

- Split data set into two pieces, fit to one, and compute negative log-likelihood on the **other**
- One set is "training data", the other is "testing data" or "held out data"
- Average over different splits
- This estimates the quality of your model
  - Often (rightfully so) used to compare algorithms
- If you are doing model selection, then you choose the model with the smallest value of this average
  - This works because adding parameters causes over fitting of the training data which gives worse performance on test data
  - However, it ignores priors over models