Mixture Models and EM

Goal: Introduction to probabilistic mixture models and the expectation-maximization (EM) algorithm.

Motivation:

- simultaneous fitting of multiple model instances
- unsupervised clustering of data
- coping with missing data

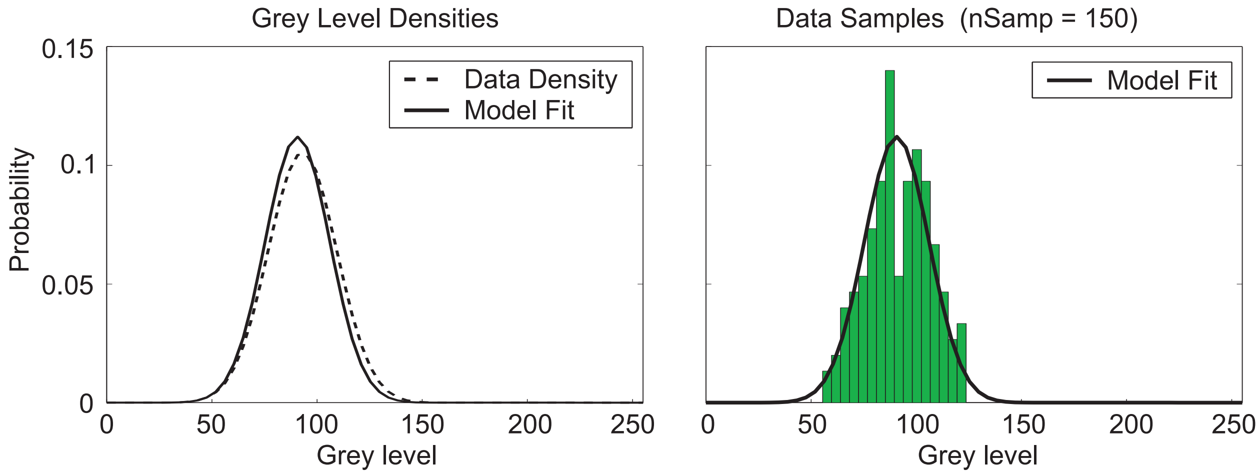
Model Fitting: Density Estimation

Let's say we want to model the distribution of a set of scalar observations $\{d_k\}_{k=1}^K$.

Non-parametric model: Compute a histogram.

Parametric model: Fit an analytic density function to the data.
For example, if we assume the samples were drawn from a Gaussian distribution, then we could fit a Gaussian density to the data by computing the sample mean and variance:

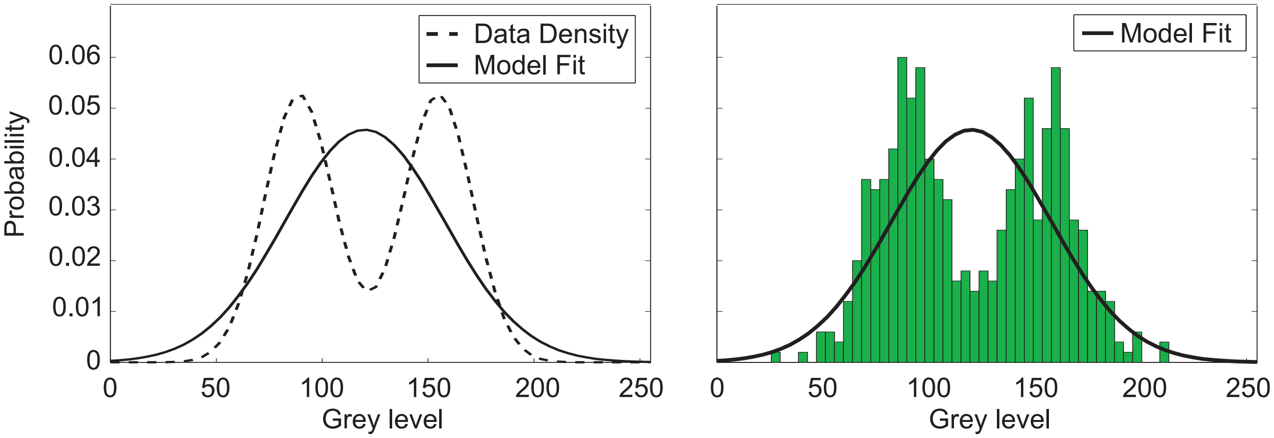
$$\mu = \frac{1}{K}\sum_k d_k \quad , \qquad \sigma^2 = \frac{1}{K-1}\sum_k (d_k - \mu)^2$$



Right plot shows a histogram of 150 IID samples drawn from the Gaussian density on the left (dashed). Overlaid is the estimated Gaussian model (solid).

Model Fitting: Multiple Data Modes

When the data come from two different sources (e.g., two distinct physical processes), then a single Gaussian density function will not fit the data well.



Missing Data: If the assignment of observations to the two modes were *known*, then we could easily solve for the means and variances using sample statistics, as before, but only incorporating those data assigned to their respective models.

Soft Assignments: But we don't know the assignments of pixels to the two Gaussians. So instead, let's infer them:

Using Bayes' rule, the probability that d_k is owned (i.e., generated) by model $\mathcal{M}_n$ is

$$p(\mathcal{M}_n \,|\, d_k) \;=\; \frac{p(d_k \,|\, \mathcal{M}_n)\, p(\mathcal{M}_n)}{p(d_k)}$$

Ownership (example)

Above we drew samples from two Gaussians in equal proportions, so

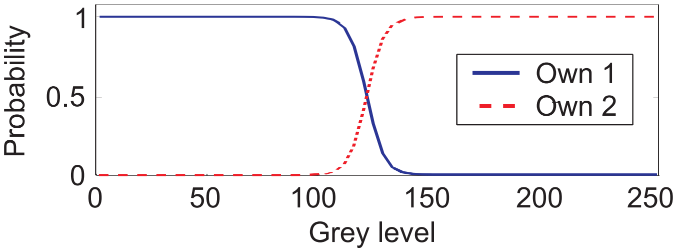
$$p(\mathcal{M}_1) = p(\mathcal{M}_2) = \frac{1}{2} \quad , \quad \text{and} \quad p(d_k \,|\, \mathcal{M}_n) \;=\; G(d_k;\, \mu_n, \sigma_n^2)$$

where $G(d;\, \mu, \sigma^2)$ is a Gaussian pdf with mean μ and variance σ^2 evaluated at d. And remember $p(d_k) = \sum_n p(d_k \,|\, \mathcal{M}_n)\, p(\mathcal{M}_n)$.

So, the *ownerships*, $q_n(d_k) \equiv p(\mathcal{M}_n \,|\, d_k)$, then reduce to

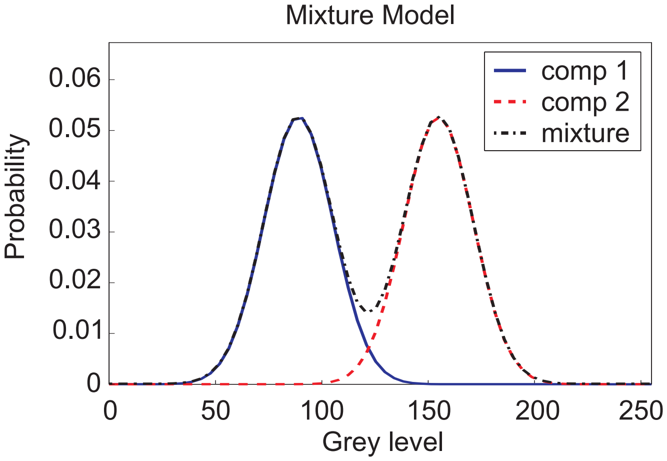
$$q_1(d_k) \;=\; \frac{G(d_k;\, \mu_1, \sigma_1^2)}{G(d_k; \mu_1, \sigma_1^2) + G(d_k; \mu_2, \sigma_2^2)} \quad , \quad \text{and} \quad q_2(d_k) \;=\; 1 - q_1(d_k)$$

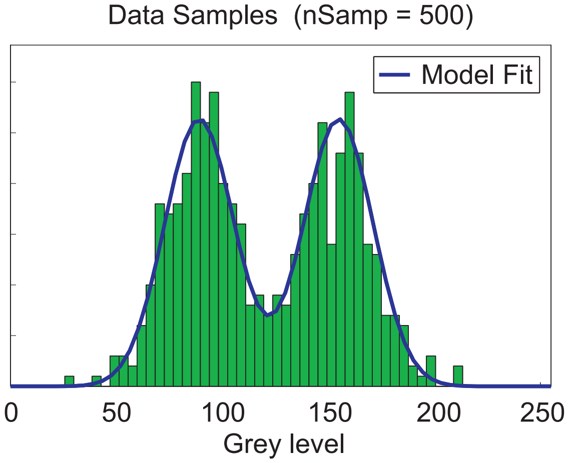
For the 2-component density below:



Then, the Gaussian parameters are given by weighted sample stats:

$$\mu_n = \frac{1}{S_n} \sum_k q_n(d_k) d_k \quad , \quad \sigma_n^2 = \frac{1}{S_n} \sum_k q_n(d_k)(d_k - \mu_n)^2 \; , \quad S_n = \sum_k q_n(d_k)$$





Mixture Model

Assume

- N processes, $\{\mathcal{M}_n\}_{n=1}^N$, each of which generates some data (or measurements).
- Each sample d from process $\mathcal{M}_n$ is IID with density $p_n(d \,|\, \vec{\mathbf{a}}_n)$, where $\vec{\mathbf{a}}_n$ denotes parameters for process $\mathcal{M}_n$.
- The proportion of the entire data set produced solely by $\mathcal{M}_n$ is denoted $m_n = p(\mathcal{M}_n)$ (it's called a *mixing probability*).

Generative Process: First, randomly select one of the N processes according to the mixing probabilities, $\vec{\mathbf{m}} \equiv (m_1, ..., m_N)$. Then, given n, generate a sample from the observation density $p_n(d \,|\, \vec{\mathbf{a}}_n)$.

Mixture Model Likelihood: The probability of observing a datum d from the collection of N processes is given by their linear mixture:

$$p(d \,|\, \mathcal{M}) \;=\; \sum_{n=1}^{N} m_n \, p_n(d \,|\, \vec{\mathbf{a}}_n)$$

The *mixture model*, $\mathcal{M}$, comprises $\vec{\mathbf{m}}$, and the parameters, $\{\vec{\mathbf{a}}_n\}_{n=1}^N$.

Mixture Model Inference: Given K IID observations (the data), $\{d_k\}_{k=1}^K$, our goal is to estimate the mixture model parameters.

Remarks: One may also wish to estimate N and the parametric form of each component, but that's outside the scope of these notes.

Expectation-Maximization (EM) Algorithm

EM is an iterative algorithm for parameter estimation, especially useful when one formulates the estimation problem in terms of *observed* and *missing* data.

- Observed data are the K intensities. Missing data are the assignments of observations to model components, $z_n(d_k) \in \{0, 1\}$.

Each EM iteration comprises an E-step and an M-step:

E-Step: Compute the expected values of the missing data given the current model parameter estimate. For mixture models one can show this gives the ownership probability: $\mathrm{E}[z_n(d_k)] = q_n(d_k)$.

M-Step: Compute ML model parameters given observed data and the expected value of the missing data. For mixture models this yields a weighted regression problem for each model component:

$$\sum_{k=1}^{K} q_n(d_k) \, \frac{\partial}{\partial \vec{\mathbf{a}}_n} \log p_n(d_k \,|\, \vec{\mathbf{a}}_n) \;=\; \vec{\mathbf{0}} \, .$$

and the mixing probabilities are $m_n \;=\; \frac{1}{K} \sum_{k=1}^{K} q_n(d_k)$.

Remarks:

- Each EM iteration can be shown to increase the likelihood of the observed data given the model parameters.
- EM converges to local maxima (not necessarily global maxima).
- An initial guess is required (e.g., random ownerships).

Derivation of EM for Mixture Models

The mixture model likelihood function is given by:

$$p\left(\{d_k\}_{k=1}^K \,|\, \mathcal{M}\right) \;=\; \prod_{k=1}^K p\,(d_k \,|\, \mathcal{M}) \;=\; \prod_{k=1}^K \sum_{n=1}^N m_n\, p_n(d_k \,|\, \vec{\mathbf{a}}_n)$$

where $\mathcal{M} \equiv \left(\vec{\mathbf{m}},\ \{\vec{\mathbf{a}}_n\}_{n=1}^N\right)$. The log likelihood is then given by

$$L(\mathcal{M}) \;=\; \log p\left(\{d_k\}_{k=1}^K \,|\, \mathcal{M}\right) \;=\; \sum_{k=1}^K \log\left(\sum_{n=1}^N m_n\, p_n(d_k \,|\, \vec{\mathbf{a}}_n)\right)$$

Our goal is to find extrema of the log likelihood function subject to the constraint that the mixing probabilities sum to 1. The constraint that $\sum_n m_n = 1$ can be included with a Lagrange multiplier. Accordingly, the following conditions can be shown to hold at the extrema of the objective function:

$$\frac{1}{K}\sum_{k=1}^K q_n(d_k) \;=\; m_n$$

and

$$\frac{\partial L}{\partial \vec{\mathbf{a}}_n} \;=\; \sum_{k=1}^K q_n(d_k)\, \frac{\partial}{\partial \vec{\mathbf{a}}_n} \log p_n(d_k \,|\, \vec{\mathbf{a}}_n) \;=\; \vec{\mathbf{0}}\,.$$

The first condition is easily derived from the derivative of the log likliehood with respect to m_n, along with the Lagrange multiplier.

The second condition is more involved as we show here, beginning with form of the derivative of the log likelihood with respect to the motion parameters for the m^{th} component:

$$\begin{aligned}
\frac{\partial L}{\partial \vec{\mathbf{a}}_n} \;&=\; \sum_{k=1}^K \frac{1}{\sum_{n=1}^N m_n\, p_n(d_k \,|\, \vec{\mathbf{a}}_n)}\, \frac{\partial}{\partial \vec{\mathbf{a}}_n}\left(\sum_{n=1}^N m_n\, p_n(d_k \,|\, \vec{\mathbf{a}}_n)\right) \\
&=\; \sum_{k=1}^K \frac{m_n}{\sum_{n=1}^N m_n\, p_n(d_k \,|\, \vec{\mathbf{a}}_n)}\, \frac{\partial}{\partial \vec{\mathbf{a}}_n} p_n(d_k \,|\, \vec{\mathbf{a}}_n) \\
&=\; \sum_{k=1}^K \frac{m_n p_n(d_k \,|\, \vec{\mathbf{a}}_n)}{\sum_{n=1}^N m_n\, p_n(d_k \,|\, \vec{\mathbf{a}}_n)}\, \frac{\partial}{\partial \vec{\mathbf{a}}_n} \log p_n(d_k \,|\, \vec{\mathbf{a}}_n)
\end{aligned}$$

The last step is an algebraic manipulation that uses the fact that $\frac{\partial \log p(a)}{\partial a} = \frac{1}{p(a)} \frac{\partial p(a)}{\partial a}$.

Derivation of EM for Mixture Models (cont)

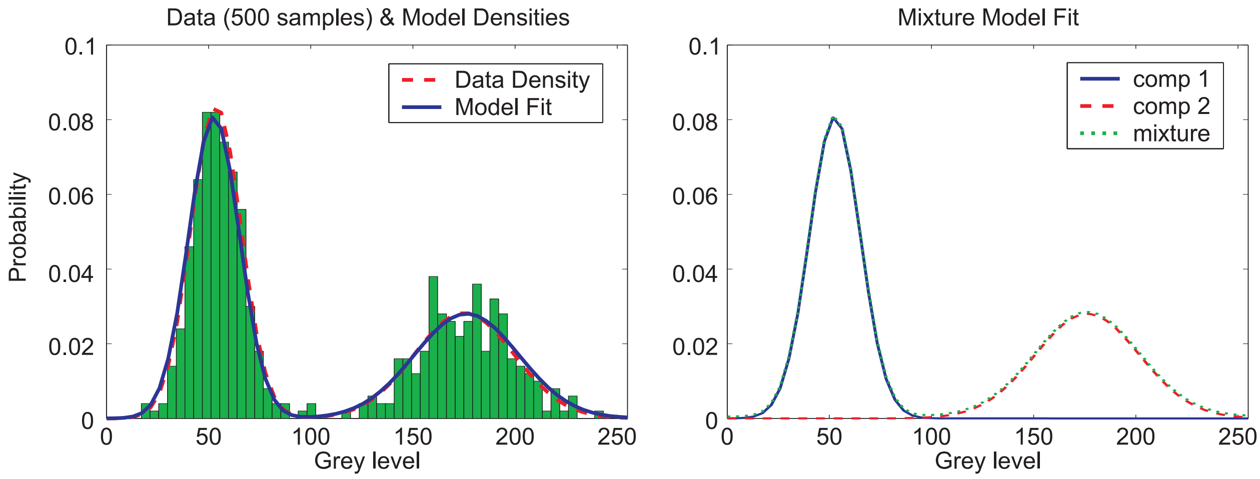
Notice that this equation can be greatly simplified because each term in the sum is really the product of the ownership probability $q_n(d_k)$ and the derivative of the component log likelihood. Therefore

$$\frac{\partial L}{\partial \vec{\mathbf{a}}_n} \;=\; \sum_{k=1}^{K} q_n(d_k)\,\frac{\partial}{\partial \vec{\mathbf{a}}_n}\log p_n(d_k \,|\, \vec{\mathbf{a}}_n)$$

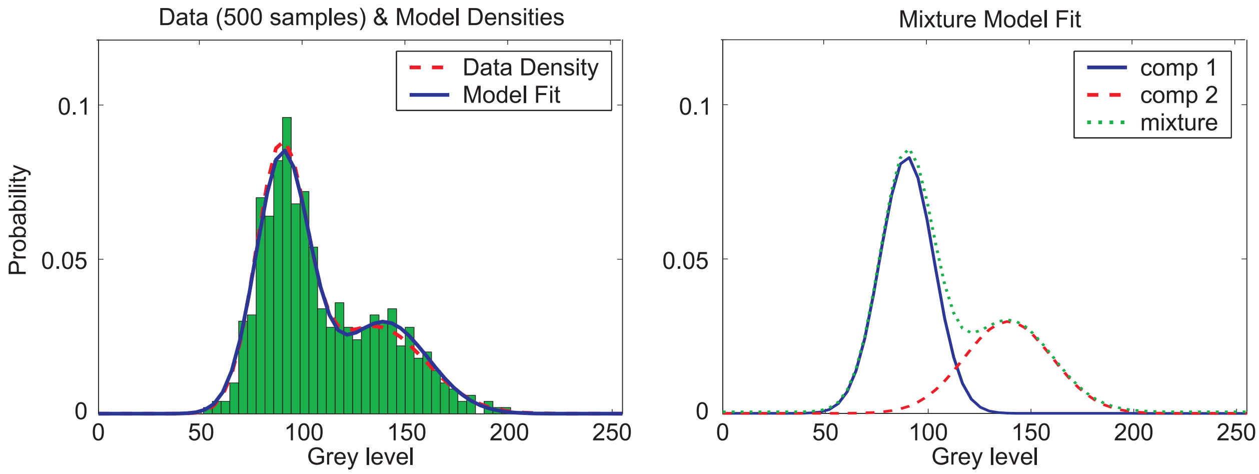
This is just a weighted log likelihood. In the case of a Gaussian component likelihood, $p_n(d_k \,|\, \vec{\mathbf{a}}_n)$, this is the derivative of a weighted least-squares error. Thus, setting $\partial L/\partial \vec{\mathbf{a}}_n = \vec{\mathbf{0}}$ in the Gaussian case yields a weighted least-squares estimate for $\vec{\mathbf{a}}_n$.

Examples

Example 1: Two distant modes. (We don't necessarily need EM here since *hard* assignments would be simple to determine, and reasonably efficient statistically.)

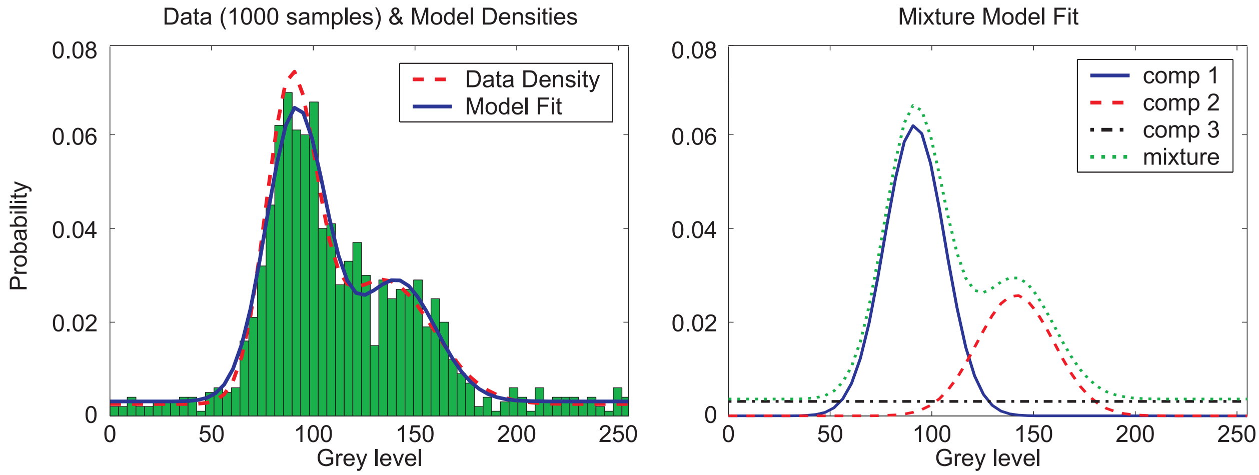


Example 2: Two nearby modes. (Here, the soft ownerships are essential to the estimation of the mode locations and variances.)

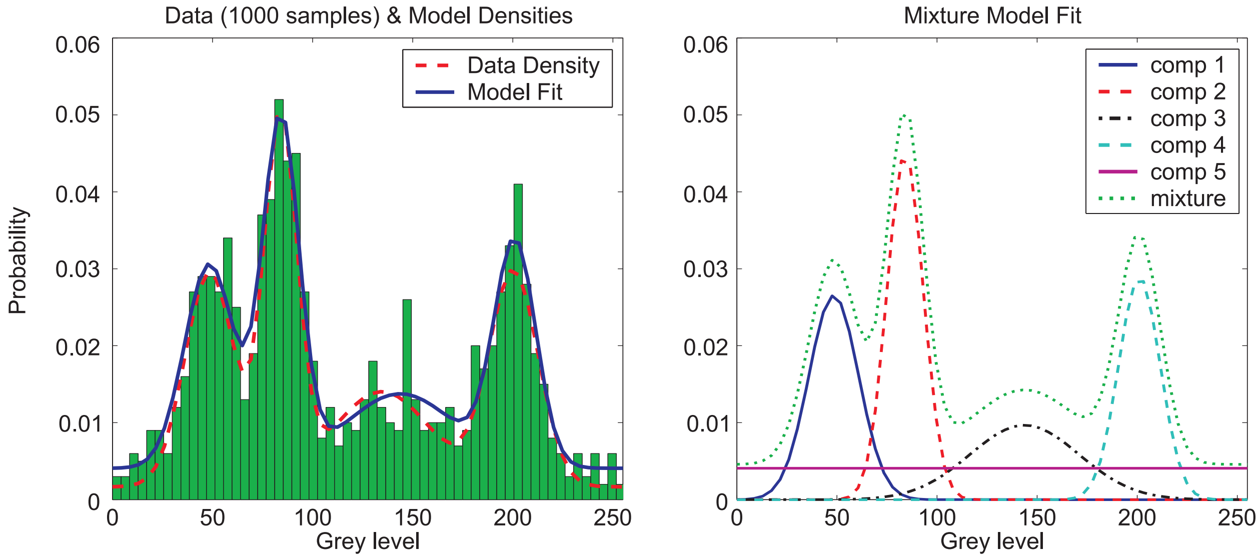


More Examples

Example 3: Nearby modes with uniformly distributed outliers. The model is a mixture of two Gaussians and a uniform outlier process.



Example 4: Four modes and uniform noise present a challenge to EM. With only 1000 samples the model fit is reasonably good.


Data (1000 samples) & Model Densities
Data Density
Model Fit
Probability
Grey level
Mixture Model Fit
comp 1
comp 2
comp 3
comp 4
comp 5
mixture
Grey level