Topic 11:

Probabilistic Mixture Models

- examples of model fitting with multiple modes
- mixture modeling basics
- the expectation-maximization algorithm (EM)
- application: taking a closer look at MLESAC

Important: Read ch.7 of Prince's book

probabilistic mixture models: motivation

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
- segmentation? (... stay tuned)

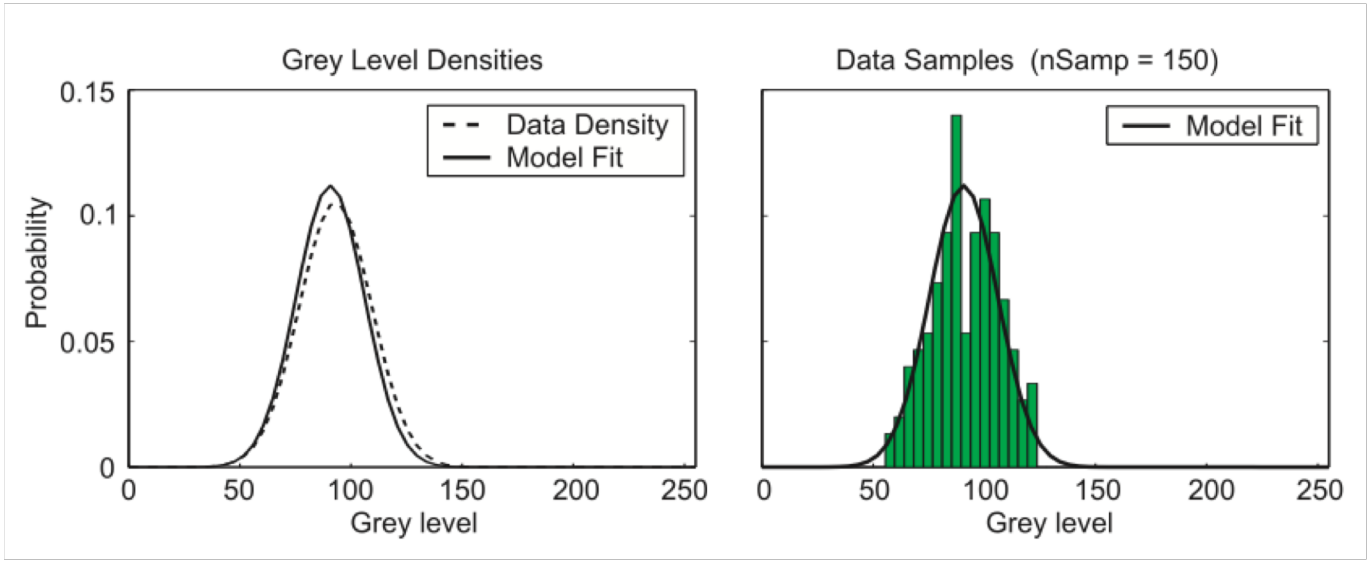
+ robust estimation, etc

probabilistic mixture models: motivation

Let's say we want to model the distribution of grey levels $d_k \equiv d(\vec{\mathbf{x}}_k)$ at pixels, $\{\vec{\mathbf{x}}_k\}_{k=1}^{K}$, within some image region of interest.

Non-parametric model: Compute a histogram.

Parametric model: Fit an analytic density function to the data.



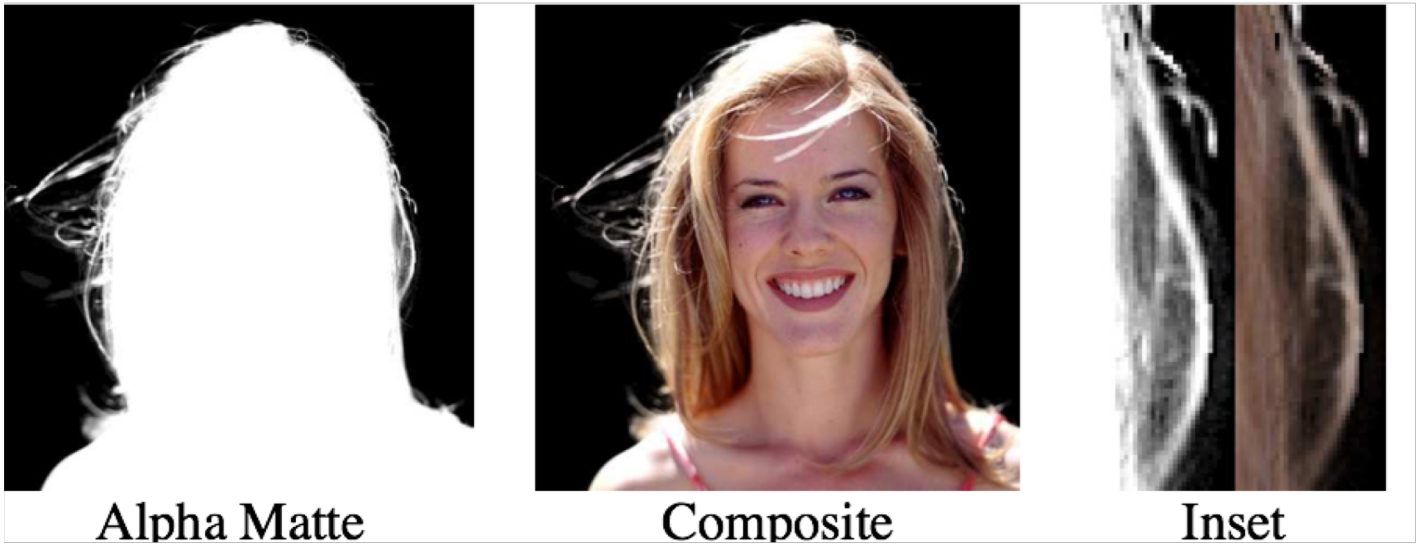
example: modeling intensity distributions

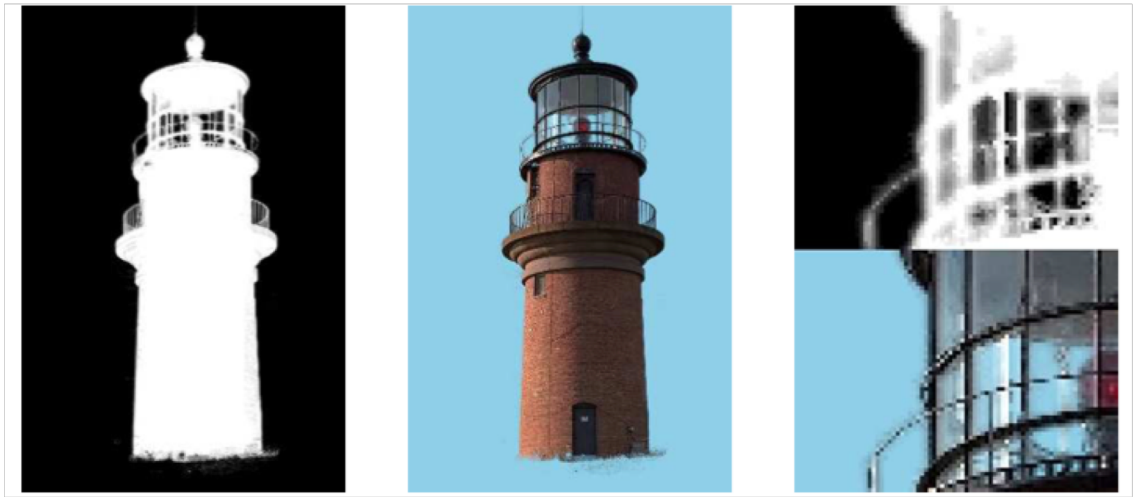
When the data come from an image region with more than one dominant color, perhaps near an occlusion boundary, then a single Gaussian will not fit the data well:

(Chuang et al, CVPR 2001)

example: modeling intensity distributions

When the data come from an image region with more than one dominant color, perhaps near an occlusion boundary, then a single Gaussian will not fit the data well:





maximum likelihood estimation of Gaussian distr.

Let's say we want to model the distribution of grey levels $d_k \equiv d(\vec{\mathbf{x}}_k)$ at pixels, $\{\vec{\mathbf{x}}_k\}_{k=1}^{K}$, within some image region of interest.

Non-parametric model: Compute a histogram.

Parametric model: Fit an analytic density function to the data.

Example: assume iid samples from $G(\mu, \sigma^2)$ with μ, σ^2 unknown

✳ - probability of samples given the model is

$$P(d_1, \dots d_K \mid \mu, \sigma^2) = \prod_{k=1}^{K} P(d_k \mid \mu, \sigma^2)$$

$$= \prod_{k=1}^{K} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(d_k-\mu)^2}{2\sigma^2}}$$

maximum likelihood estimation of Gaussian distr.

Example: assume iid samples from $G(\mu, \sigma^2)$ with μ, σ^2 unknown

- probability of samples given the model is

$$P(d_1, \dots d_K | \mu, \sigma^2) = \prod_{k=1}^{K} P(d_k | \mu, \sigma^2)$$

$$= \prod_{k=1}^{K} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(d_k-\mu)^2}{2\sigma^2}}$$

- negative log likelihood

$$L(\mu, \sigma^2) = K \log \sigma\sqrt{2\pi} + \sum_{k=1}^{K} \frac{(d_k-\mu)^2}{2\sigma^2}$$

$$* \quad \frac{\partial L}{\partial \mu} = \frac{-K\mu + \sum_{k=1}^{K} d_k}{\sigma^2} = 0 \Rightarrow \boxed{\mu_{ML} = \frac{1}{K} \sum_{k=1}^{K} d_k} \quad *$$

maximum likelihood estimation of Gaussian distr.

Example: assume iid samples from $G(\mu, \sigma^2)$ with μ, σ^2 unknown

- probability of samples given the model is

$$P(d_1, \dots d_K | \mu, \sigma^2) = \prod_{k=1}^{K} P(d_k | \mu, \sigma^2)$$

$$= \prod_{k=1}^{K} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(d_k-\mu)^2}{2\sigma^2}}$$

- negative log likelihood

$$L(\mu, \sigma^2) = K \log \sigma\sqrt{2\pi} + \sum_{k=1}^{K} \frac{(d_k-\mu)^2}{2\sigma^2}$$

$$\frac{\partial L}{\partial \sigma} = \frac{K}{\sigma} - \frac{1}{\sigma^3} \sum_{k=1}^{K} (d_k-\mu)^2 = 0 \Rightarrow \sigma^2_{ML} = \frac{1}{K} \sum (d_k-\mu)^2$$

*

maximum likelihood estimation of Gaussian distr.

Example: assume iid samples from $G(\mu, \sigma^2)$ with μ, σ^2 unknown

- but we dont know μ – all we have is our estimate $\mu_{ML} = \frac{1}{K} \sum d_k$
- if we use μ_{ML} instead of μ we have

$$E\left[\frac{1}{K} \sum (d_k - \mu_{ML})^2 \right] = \frac{K-1}{K} \sigma^2 \quad *$$

maximum likelihood estimation of Gaussian distr.

Example: assume iid samples from $G(\mu, \sigma^2)$ with μ, σ^2 unknown

- but we dont know μ – all we have is our estimate $\mu_{ML} = \frac{1}{K}\sum d_k$
- if we use μ_{ML} instead of μ we have

$$E\left[\frac{1}{K}\sum (d_k - \mu_{ML})^2\right] = \frac{K-1}{K}\sigma^2 \neq \sigma^2$$

the estimate is biased

to remove the bias, multiply by $\frac{K}{K-1}$ to get

$$\sigma^2_{ML} = \frac{K}{K-1}\cdot\hat{\sigma}^2 = \frac{1}{K-1}\sum (d_k - \mu_{ML})^2$$

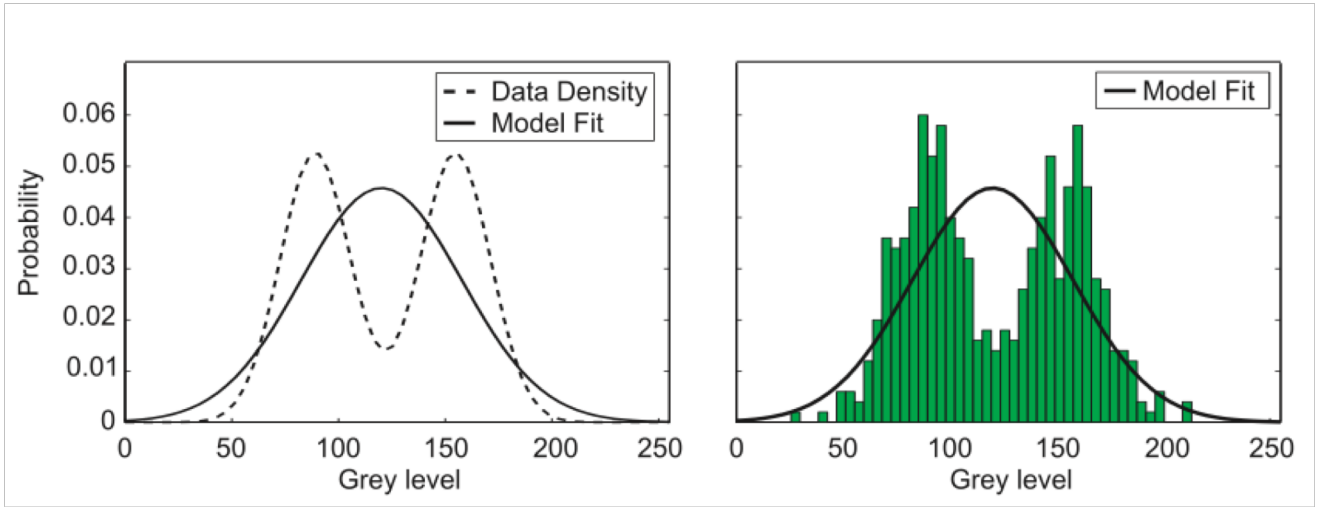
$$\hat{\sigma}^2 = \frac{1}{K}\sum (d_k - \mu)^2$$

example: modeling intensity distributions

When the data come from an image region with more than one dominant color, perhaps near an occlusion boundary, then a single Gaussian will not fit the data well:

example: modeling intensity distributions

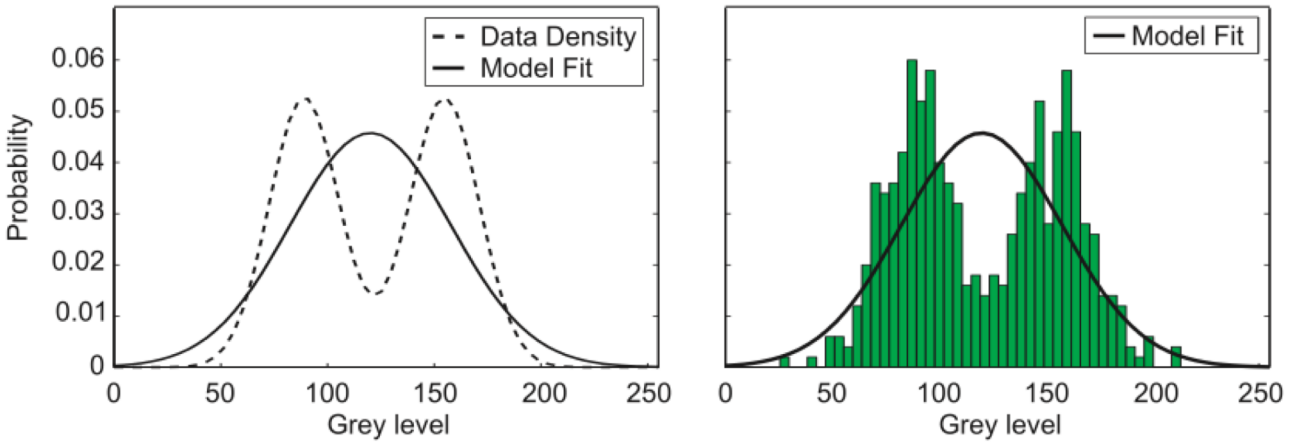
When the data come from an image region with more than one dominant color, perhaps near an occlusion boundary, then a single Gaussian will not fit the data well:



example: modeling intensity distributions

Missing Data: If the assignment of measurements 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.

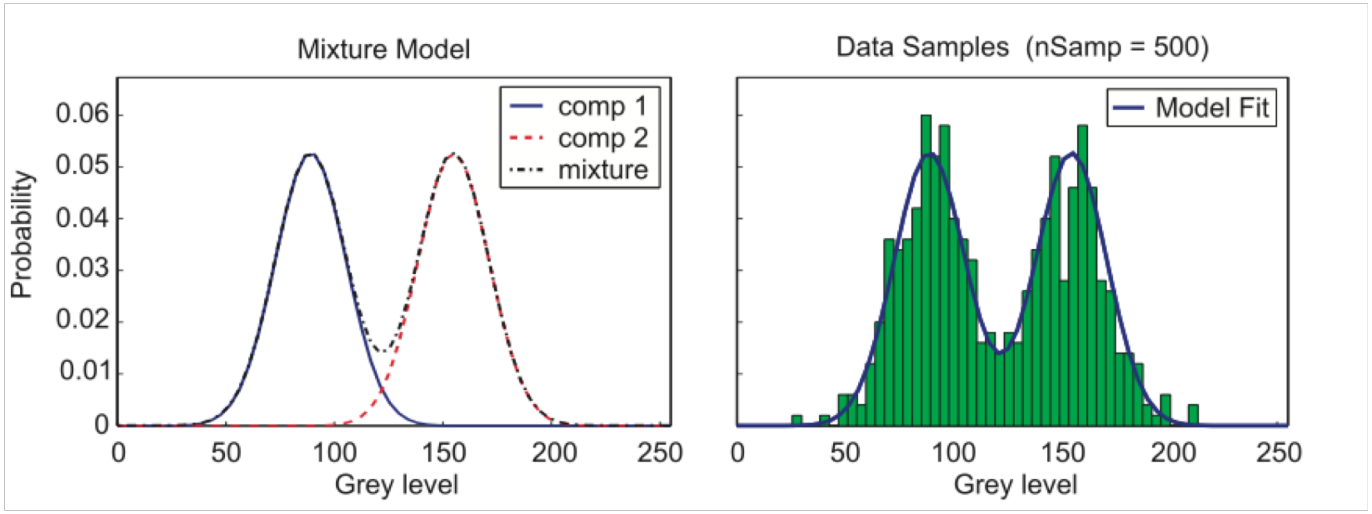
Soft Assignments: But we don't know the assignments of pixels to the two Gaussians. So instead, let's infer them:



example: modeling intensity distributions

Missing Data: If the assignment of measurements 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.

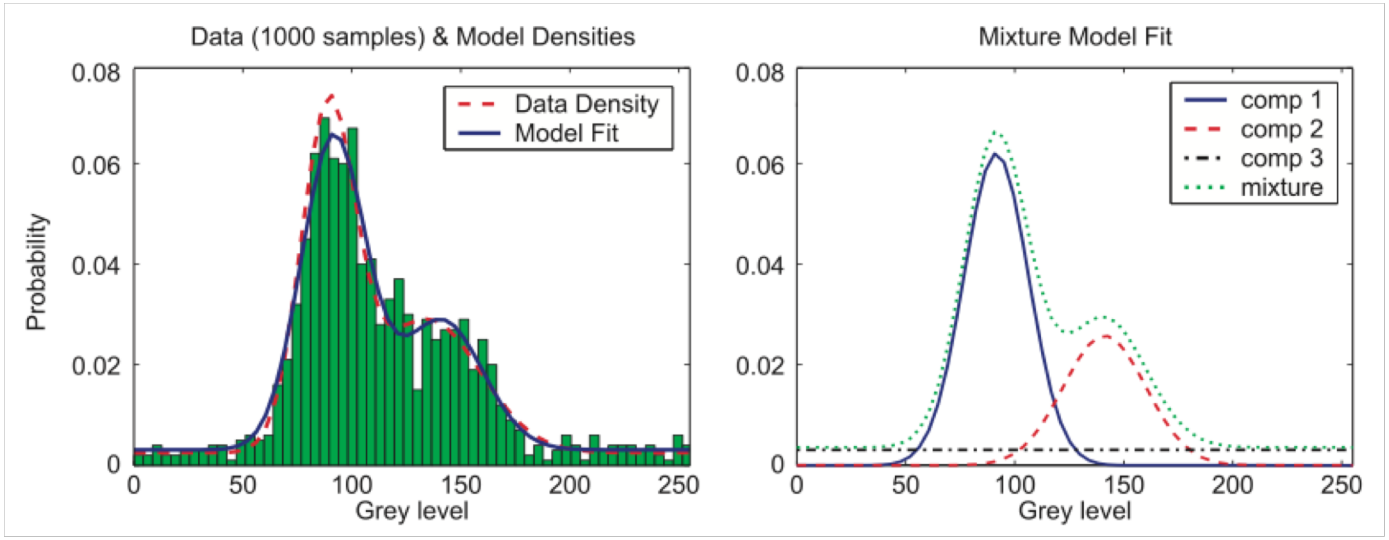
Soft Assignments: But we don't know the assignments of pixels to the two Gaussians. So instead, let's infer them:



example: modeling outliers

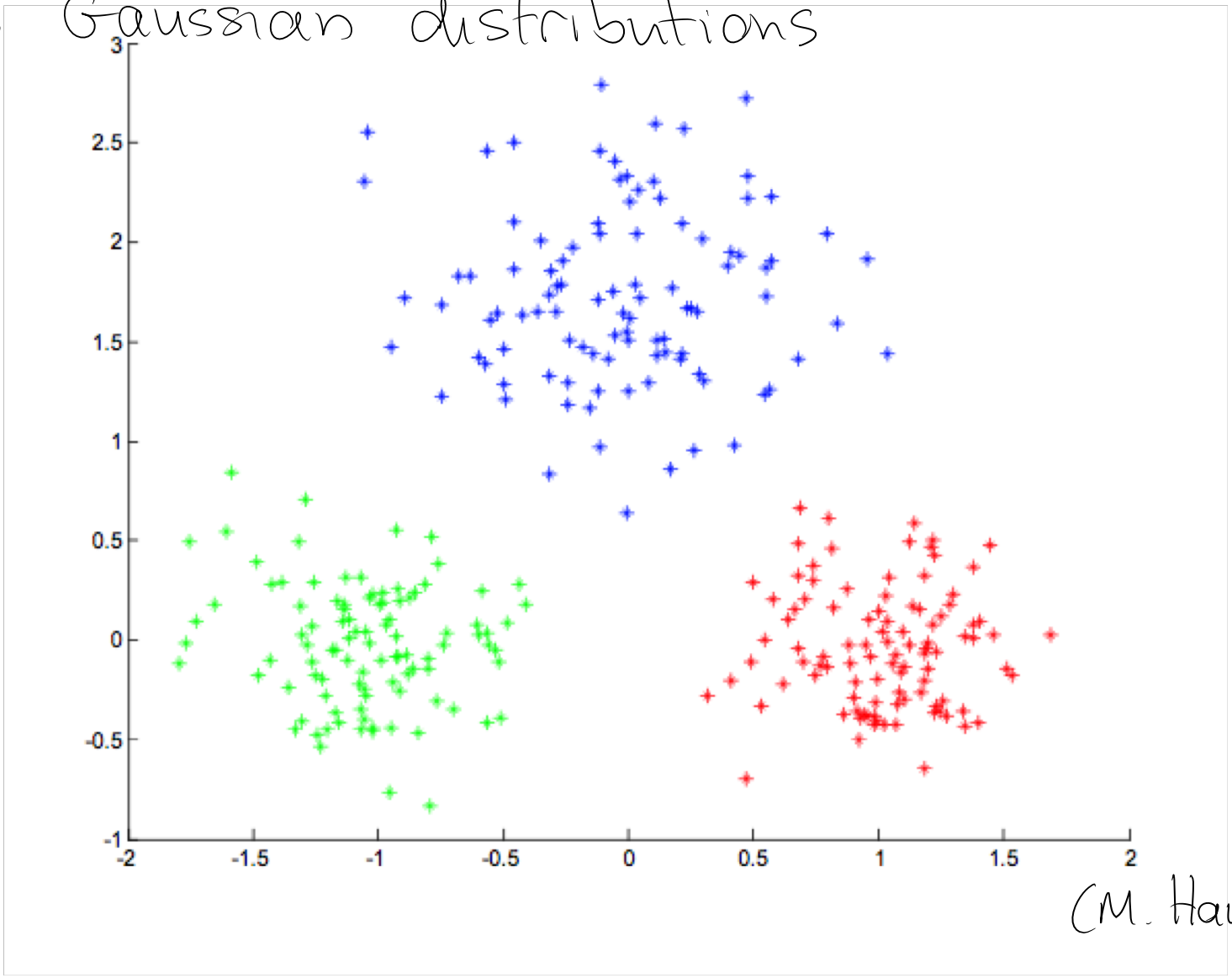
- Suppose we have outliers in addition to the two Gaussians
- A common way to model them is to assume they follow a 3rd distribution which is uniform over the range of values

* agnostic about outlier values



example: modeling 2D distributions

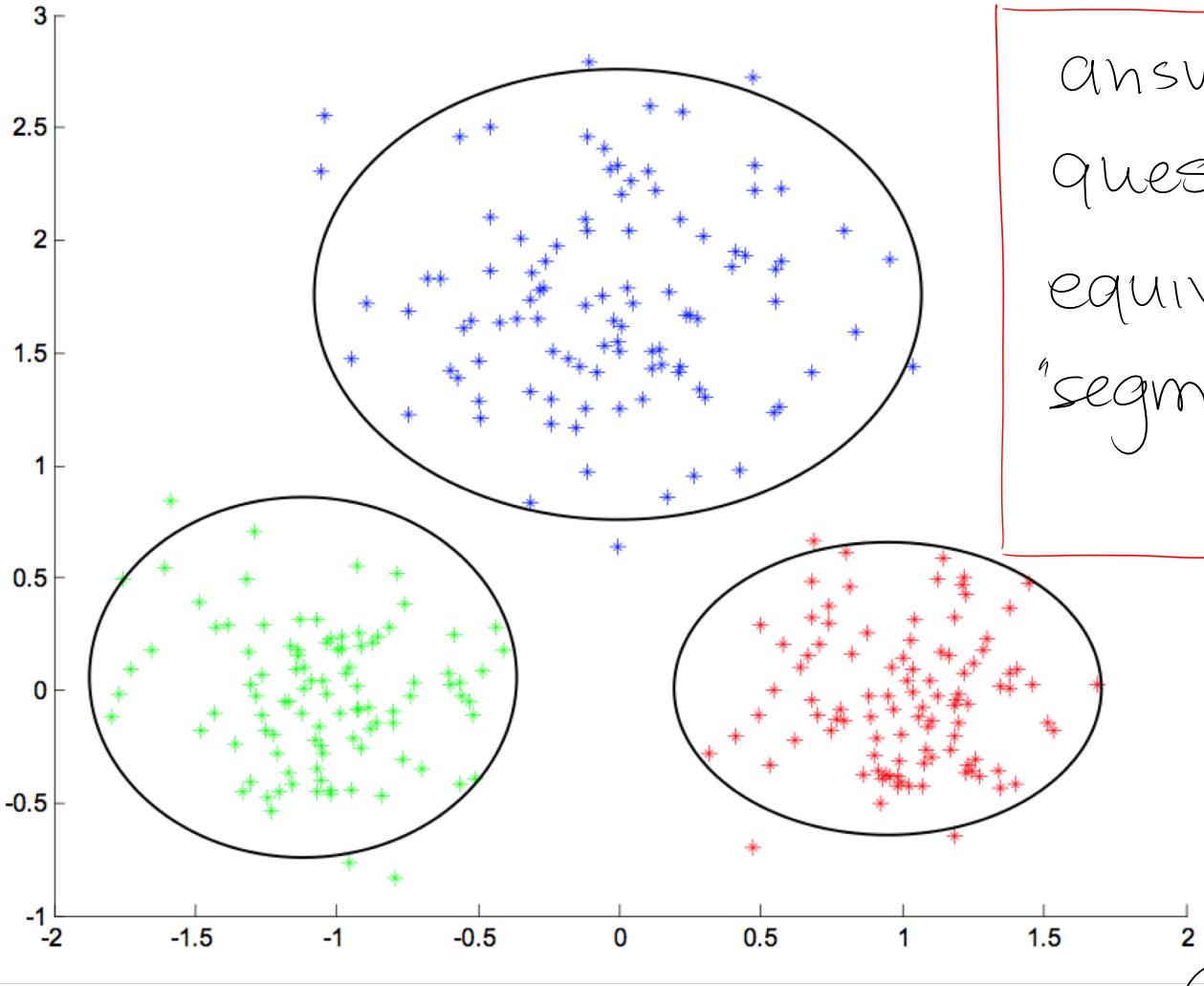
Here the 2D points are samples from 3 2D Gaussian distributions



(M. Hauskrecht)

example: modeling 2D distributions

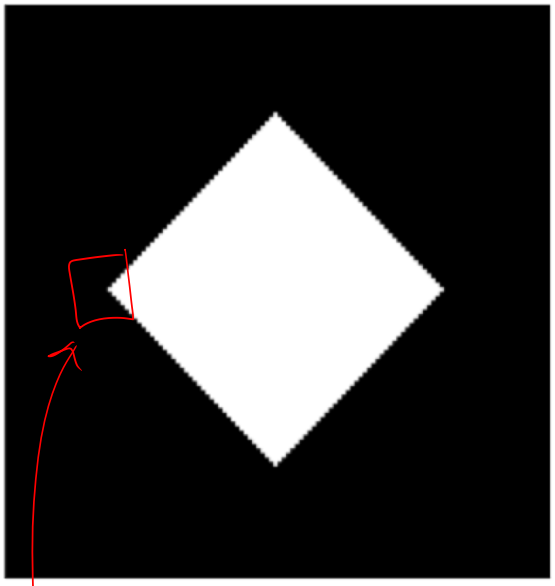
So how can we identify them?



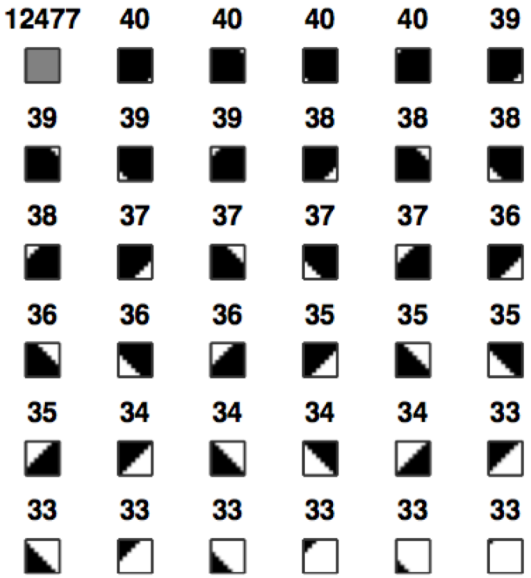
answering this question is equivalent to "segmenting" the points

(M. Hauskrecht)

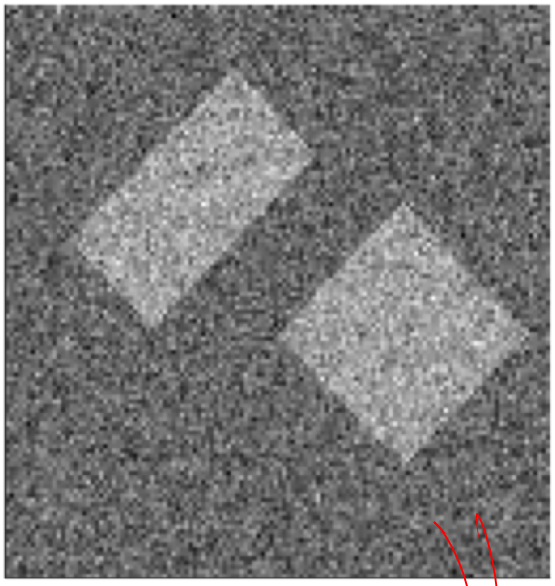
example: modeling image patches



(a) Training Image



(b) Prior Learned



(c) Noisy Image

- N×N patch of pixels ⇒ vector in $\mathbb{R}^{N^2}$
- we can model the distribution of the vectors found in a training image
- we can then use that distribution as a prior to perform to evaluate patch likelihoods in a new photo

(Zoran & Weiss, 2011)

Topic 11:

Probabilistic Mixture Models

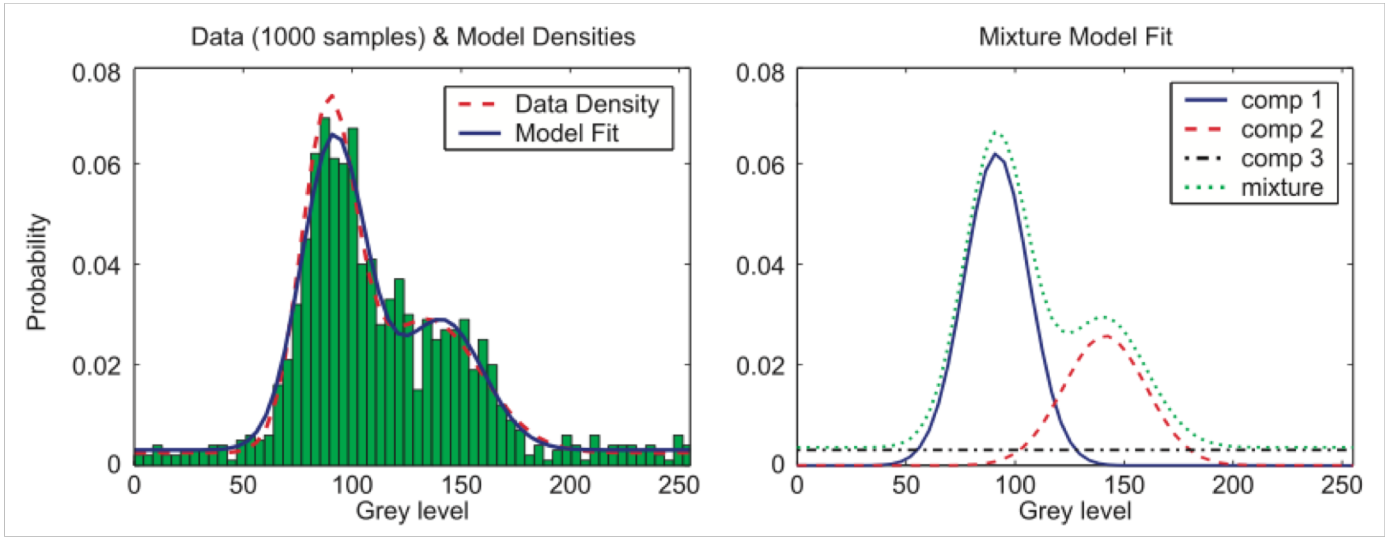
- examples of model fitting with multiple modes
- mixture modeling basics
- the expectation-maximization algorithm (EM)
- application: taking a closer look at MLESAC

basic model

- 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 $p_n(d \mid \vec{\mathbf{a}}_n)$, 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*).

sample model parameters

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 $p_n(d \mid \vec{\mathbf{a}}_n)$.



basic model

- 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 $p_n(d \mid \vec{\mathbf{a}}_n)$, 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*).

sample, model parameters

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 $p_n(d \mid \vec{\mathbf{a}}_n)$.

Probability of a sample given the mixture model:

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

- d: sample
- M: complete model $(\vec{m}, \vec{a}_1, \ldots, \vec{a}_N)$
- m_n: probability the n^{th} process was chosen for this sample
- $p_n(d \mid \vec{a}_n)$: probability of the sample given that is was generated by that process

ownership probabilities

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

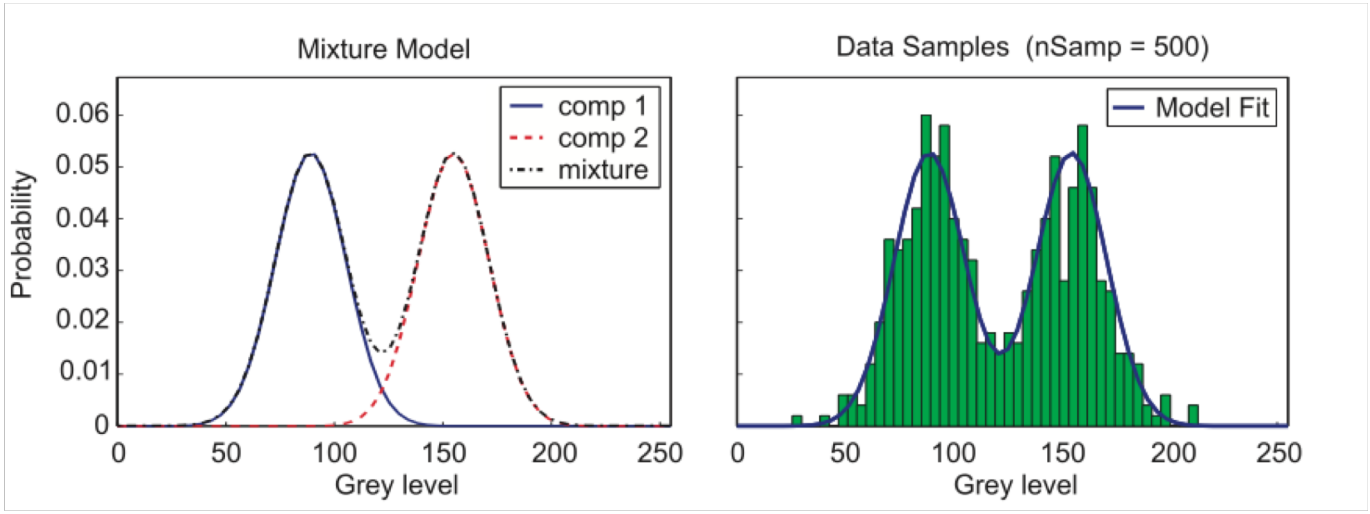
$$q_n(d) \stackrel{\text{def}}{=} P(d \text{ is "owned by" } M_n) \stackrel{\text{def}}{=} P(M_n \mid d)$$

From Bayes rule:

$$q_n(d) = P(M_n \mid d) = \frac{P(d \mid M_n)\, P(M_n)}{P(d)}$$

distribution of n^{th} process

its mixing probability



inference when mixture coefs & params known

Suppose $N = 2$ and $P(M_1) = P(M_2) = \frac{1}{2}$

$$P(d \mid M_n) = G(d; \mu_n, \sigma_n^2)$$

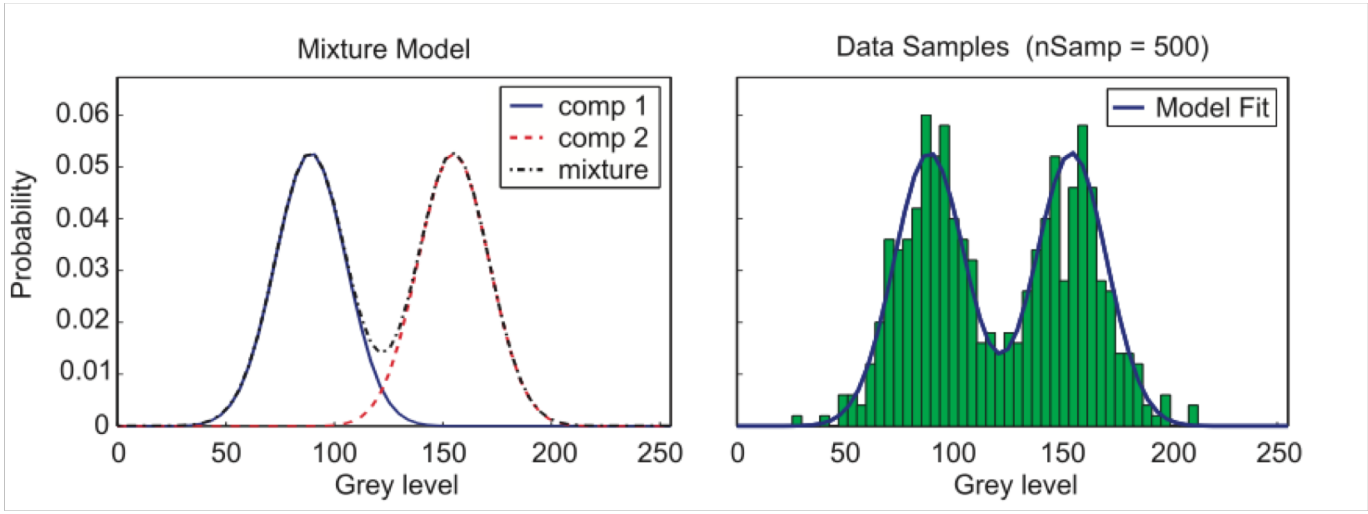
known

From Bayes rule:

$$q_n(d) = P(M_n \mid d) = \frac{P(d \mid M_n)\, P(M_n)}{P(d)}$$

distribution of n^{th} process

its mixing probability



inference when mixture coefs & params known

Suppose $N=2$ and $P(M_1) = P(M_2) = \frac{1}{2}$

$$P(d|M_n) = G(d; \mu_n, \sigma_n^2)$$

μ_n, σ_n^2: known

From Bayes rule:

$$q_n(d) = P(M_n|d) = \frac{P(d|M_n)\,P(M_n)}{P(d)}$$

$P(d|M_n)$: distribution of n^{th} process

$P(M_n)$: its mixing probability

$P(d) \leftarrow \sum_{n=1}^{N} P(d|M_n) \cdot P(M_n)$

$$q_1(d) = \frac{G(d; \mu_1, \sigma_1^2) \cdot \frac{1}{2}}{G(d; \mu_1, \sigma_1^2)\frac{1}{2} + G(d; \mu_2, \sigma_2^2)\frac{1}{2}}$$

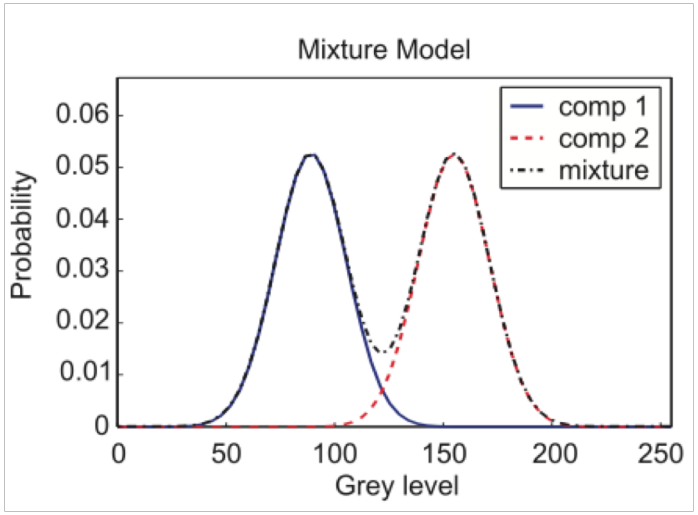
$$q_2(d) = 1 - q_1(d)$$

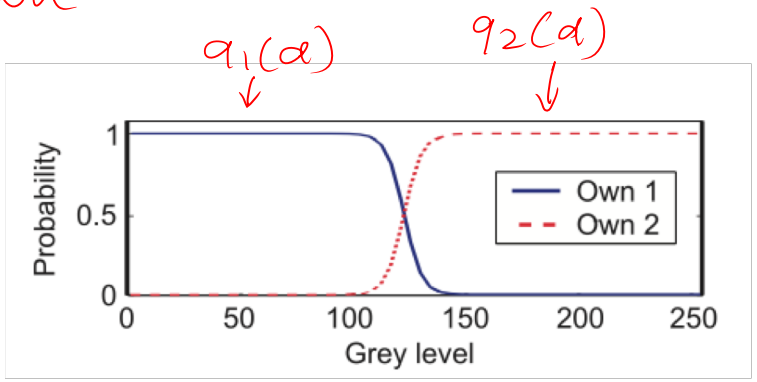
inference when mixture coefs & params known

Suppose $N=2$ and $P(M_1) = P(M_2) = \frac{1}{2}$

$$P(d \mid M_n) = G(d; \mu_n, \sigma_n^2)$$

known





$$q_1(d) = \frac{G(d; \mu_1, \sigma_1^2) \cdot \frac{1}{2}}{G(d; \mu_1, \sigma_1^2)\frac{1}{2} + G(d; \mu_2, \sigma_2^2)\frac{1}{2}}$$

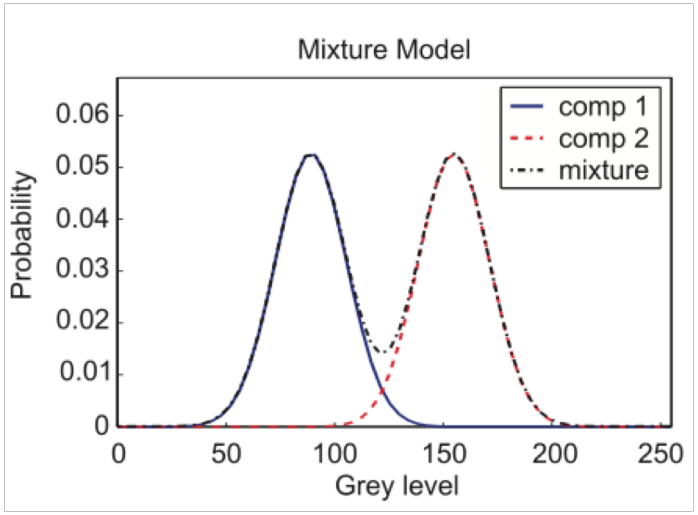
$$q_2(d) = 1 - q_1(d)$$

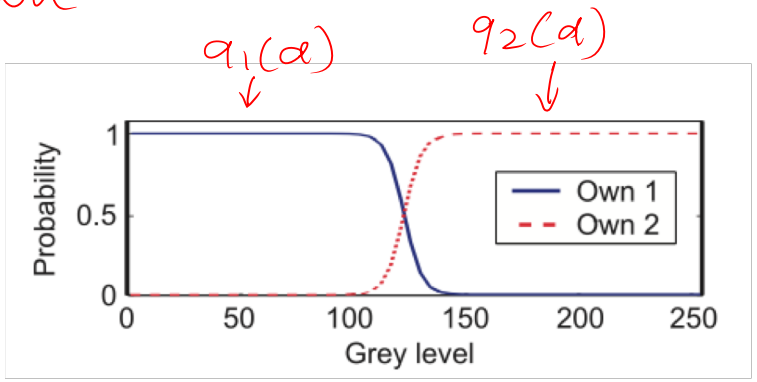
inference when mixture coefs & params known

Suppose $N=2$ and $P(M_1) = P(M_2) = \frac{1}{2}$

$$P(d \mid M_n) = G(d; \mu_n, \sigma_n^2)$$

known





$$q_1(d) = \frac{G(d; \mu_1, \sigma_1^2) \cdot \frac{1}{2}}{G(d; \mu_1, \sigma_1^2)\frac{1}{2} + G(d; \mu_2, \sigma_2^2)\frac{1}{2}}$$

$$q_2(d) = 1 - q_1(d)$$

inference when ownerships are known

Suppose $N=2$ and $P(M_1) = P(M_2) = \frac{1}{2}$

$$P(d|M_n) = G(d; \mu_n, \sigma_n^2)$$

unknown

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

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

Topic 11:

Probabilistic Mixture Models

- examples of model fitting with multiple modes
- mixture modeling basics
- the expectation-maximization algorithm (EM)
- application: taking a closer look at MLESAC

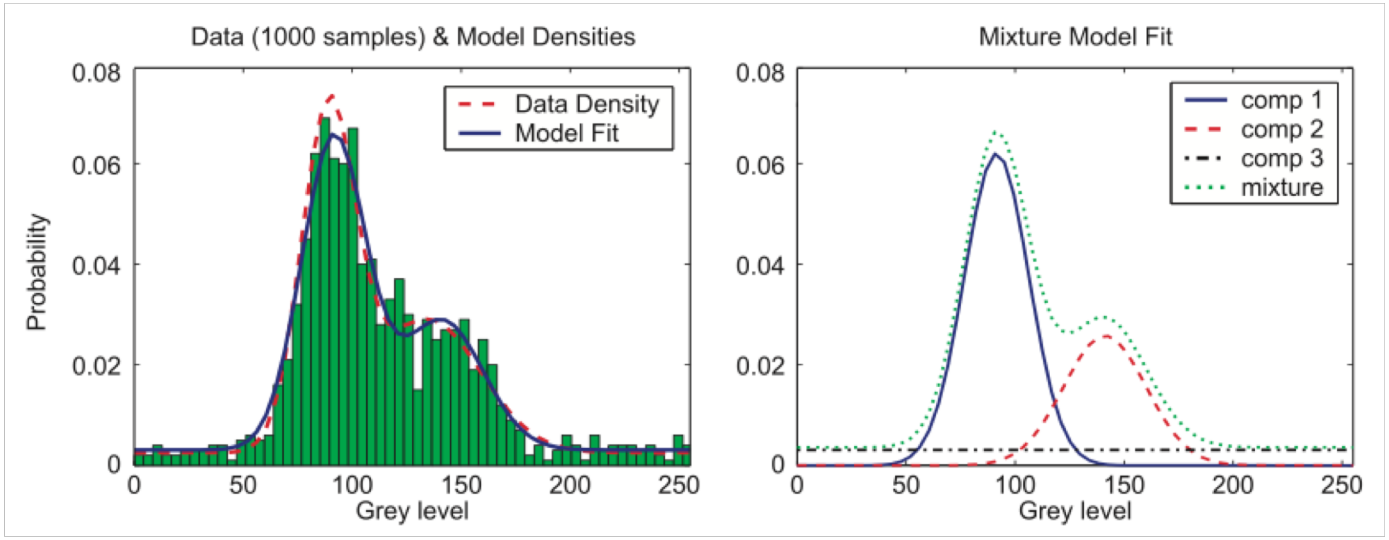
inference when mixture coefficients are unknown

- 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 $p_n(d \mid \vec{\mathbf{a}}_n)$, 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*).

unknown model parameters

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 $p_n(d \mid \vec{\mathbf{a}}_n)$.

unknown mixture coefficients



the 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, $z_n(d_k) \in \{0, 1\}$.

"no" "yes"

was observation d_k generated by model n?

Steps:

1. Fix the parameters of the model components and compute the expected value of the data assignments $E[z_n(d_k)]$

E-step

2. Given the expected value of the assignments, compute an ML-estimate of the model components.

M-step

3. Repeat until convergence

the expectation-maximization (EM) algorithm

- 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).

Steps:

1. Fix the parameters of the model components and compute the expected value of the data assignments $E[z_{n(dk)}]$ — E-step

2. Given the expected value of the assignments, compute an ML-estimate of the model components. — M-step

3. Repeat until convergence

the expectation-maximization (EM) algorithm

- 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).

Steps:

① Fix the parameters of the model components and compute the expected value of the data assignments $E[z_n(d_k)]$

E-step

$$E[z_n(d_k)] = 1 \cdot p(z_n(d_k)=1) + 0 \cdot p(z_n(d_k)=0) = p(z_n(d_k)=1) = g_n(d_k)$$

ownership probability.

the expectation-maximization (EM) algorithm

- 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).

Steps:

1

E-step (equivalent)

Fix the parameters of the model components and compute the ownership probabilities $q_n(d_x)$

(see slide 24)

the expectation-maximization (EM) algorithm

The mixture model likelihood function is given by:

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

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

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

↳ differentiate wrt $\vec{a}_n$ & set to 0

$\vec{m}$

(2) Given the ownership probabilities compute an ML-estimate of the parameters of the mixture components

M-step

the expectation-maximization (EM) algorithm

$$* \quad \sum_k q_n(d_k) \frac{\partial}{\partial \vec{a}_n} \log p_n(d_k | \vec{a}_n) = \vec{0}$$

see notes for derivation

$$\frac{\partial L(\mathcal{M})}{\partial \vec{a}_n} = \vec{0}$$

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

↳ differentiate wrt $\vec{a}_n$ & set to 0

(2) M-step

Given the ownership probabilities compute an ML-estimate of the parameters of the mixture components

the expectation-maximization (EM) algorithm

Solve the weighted least-squares problem

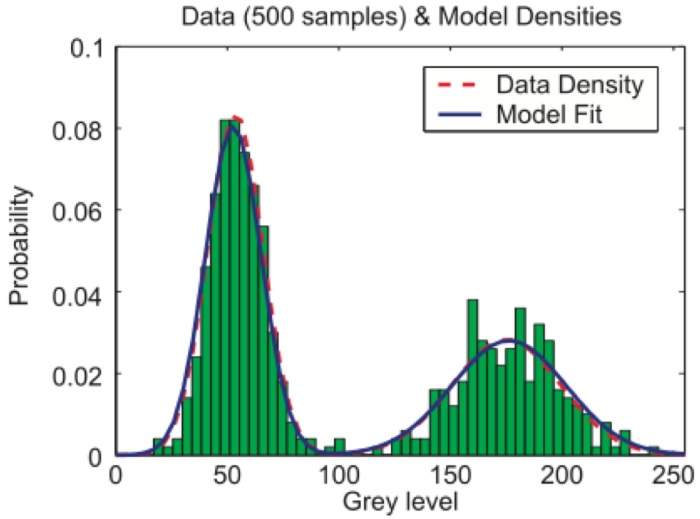
$$\sum_n q_n(d_k) \frac{\partial}{\partial \vec{a}_n} \log p_n(d_k | \vec{a}_n) = \vec{0}$$

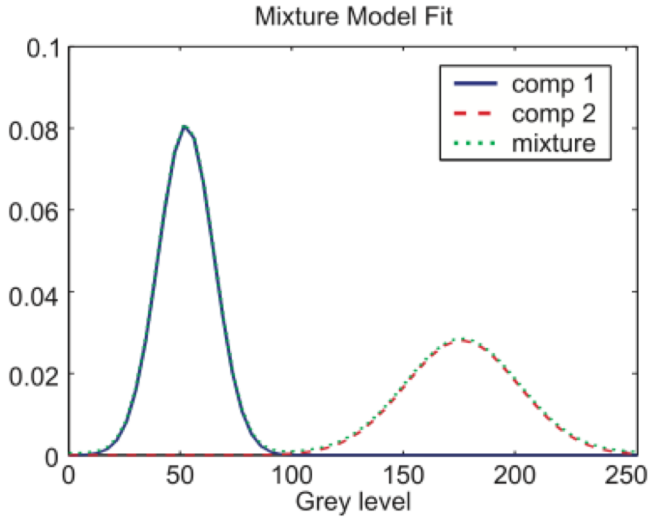
(see notes for derivation & details)

M-step (2) Given the ownership probabilities compute an ML-estimate of the parameters of the mixture components

inferring mixture models using EM

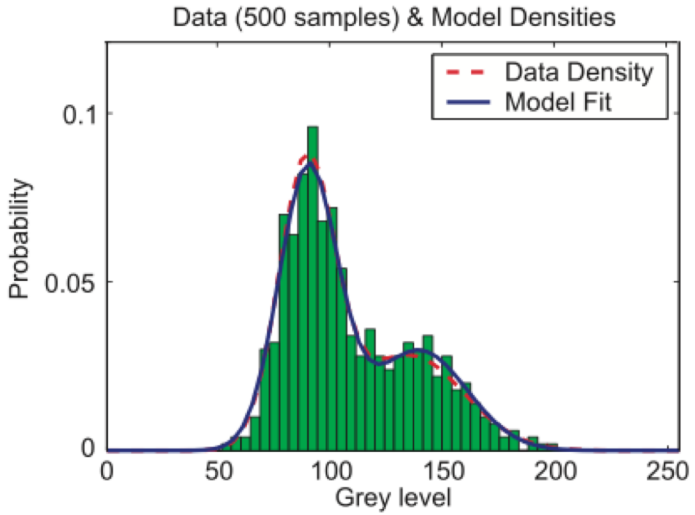
Example 1: Two distant modes. (We don't necessarily need EM here since *hard* assignments would be simple to determine, and reasonably efficient statistically.)

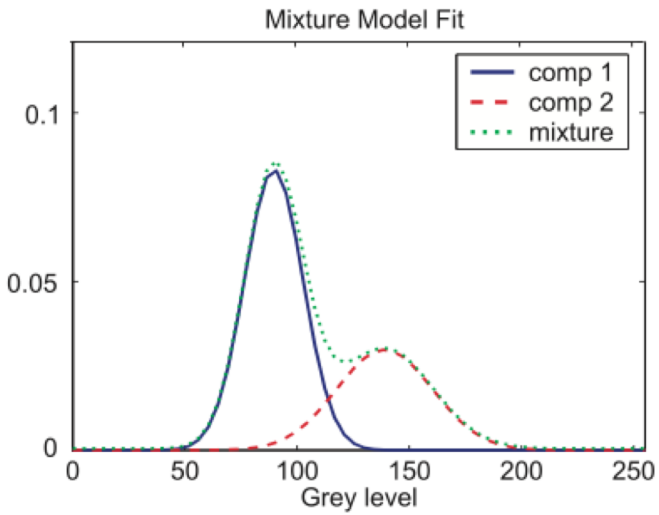




inferring mixture models using EM

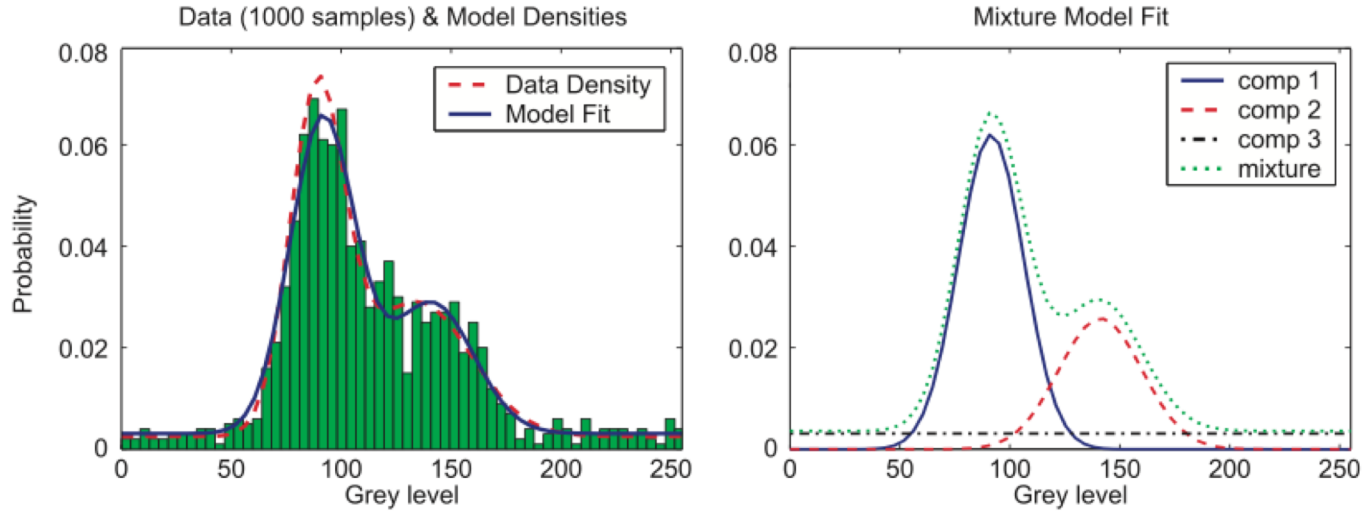
Example 2: Two nearby modes. (Here, the soft ownerships are essential to the estimation of the mode locations and variances.)





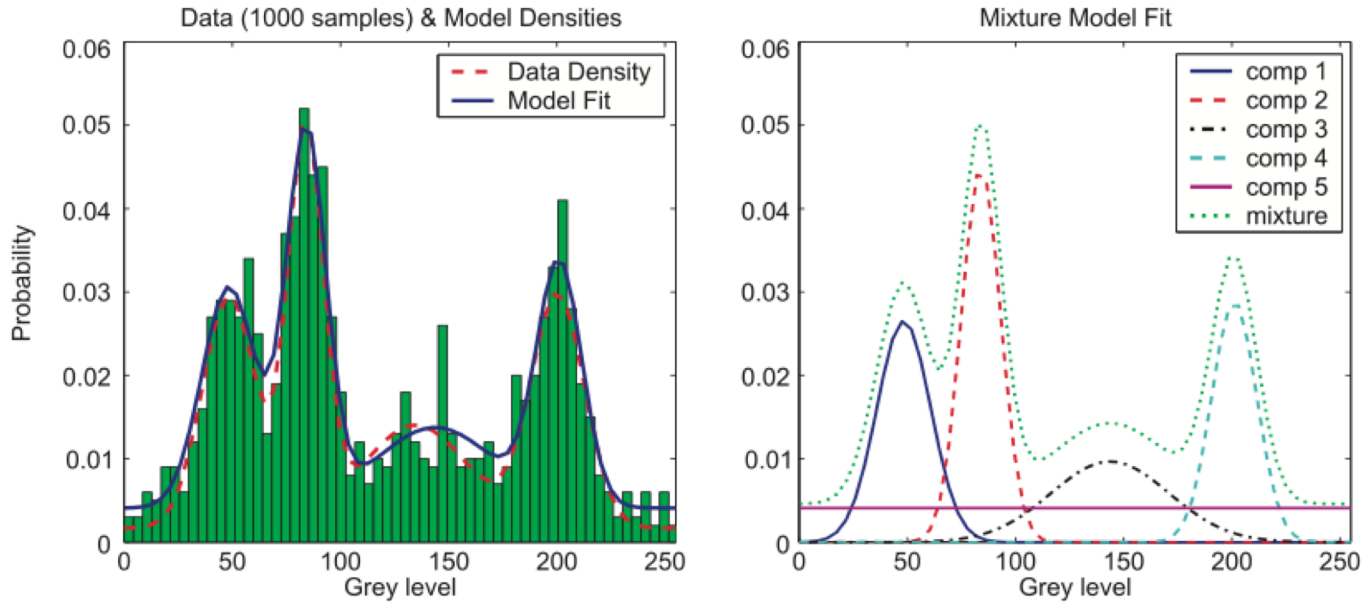
inferring mixture models using EM

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



inferring mixture models using EM

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



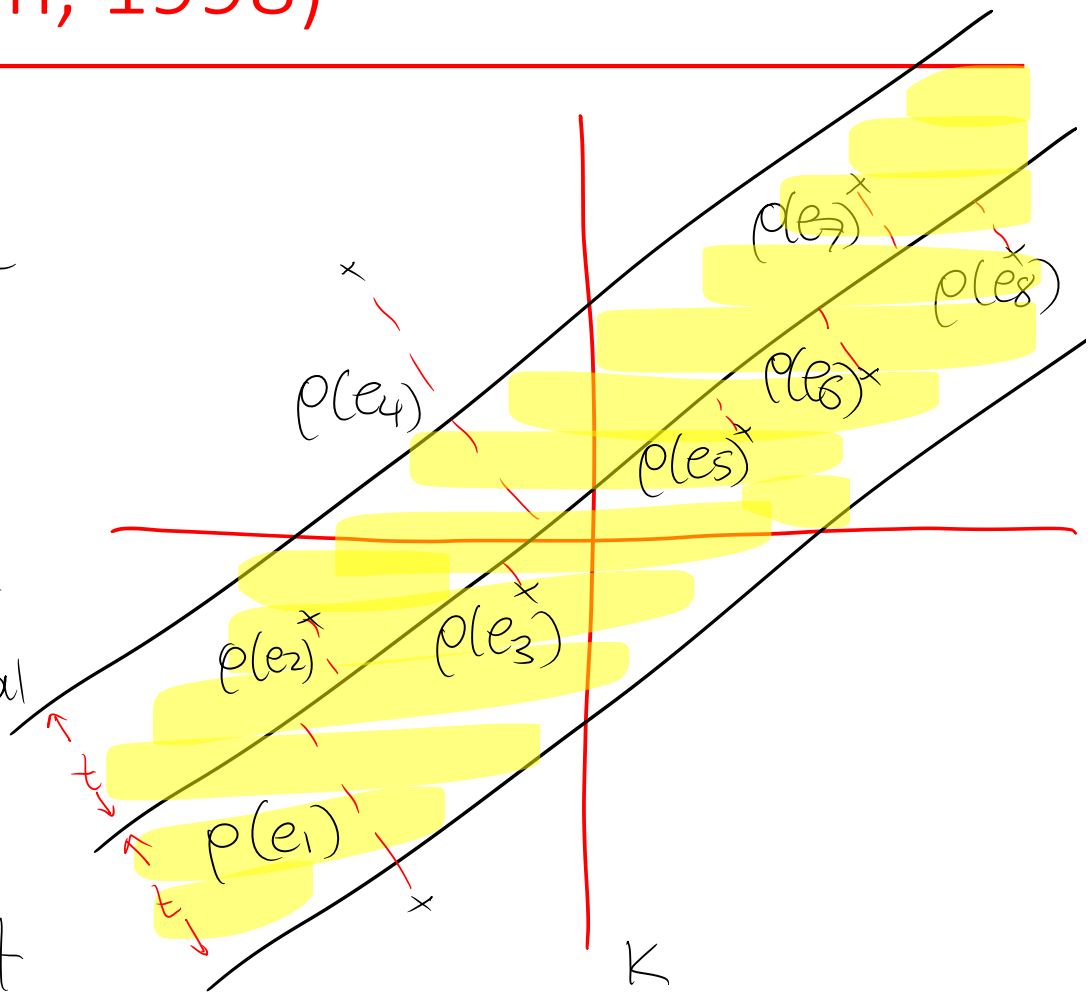
Topic 11:

Probabilistic Mixture Models

- examples of model fitting with multiple modes
- mixture modeling basics
- the expectation-maximization algorithm (EM)
- application: taking a closer look at MLESAC

MSAC (Torr & Zisserman, 1998)

1. Repeat the following:
   a. Select a random sample of minimal size for model fitting
   b. Fit the model
   c. Calculate the fit error using a robust functional
2. Select the best solution over all samples
3. Minimize the robust cost function over all datapoints



$$O(\rho) = \sum_{k=1}^{K} \rho(e_k)$$

$$\rho(e_k) = \begin{cases} e_k^2 & |e_k| < t \\ t^2 & |e_k| \geq t \end{cases}$$

MLESAC (Torr & Zisserman, 2000)

Mixture model for measurements:

$$x_k = \hat{x}_k + e_k$$

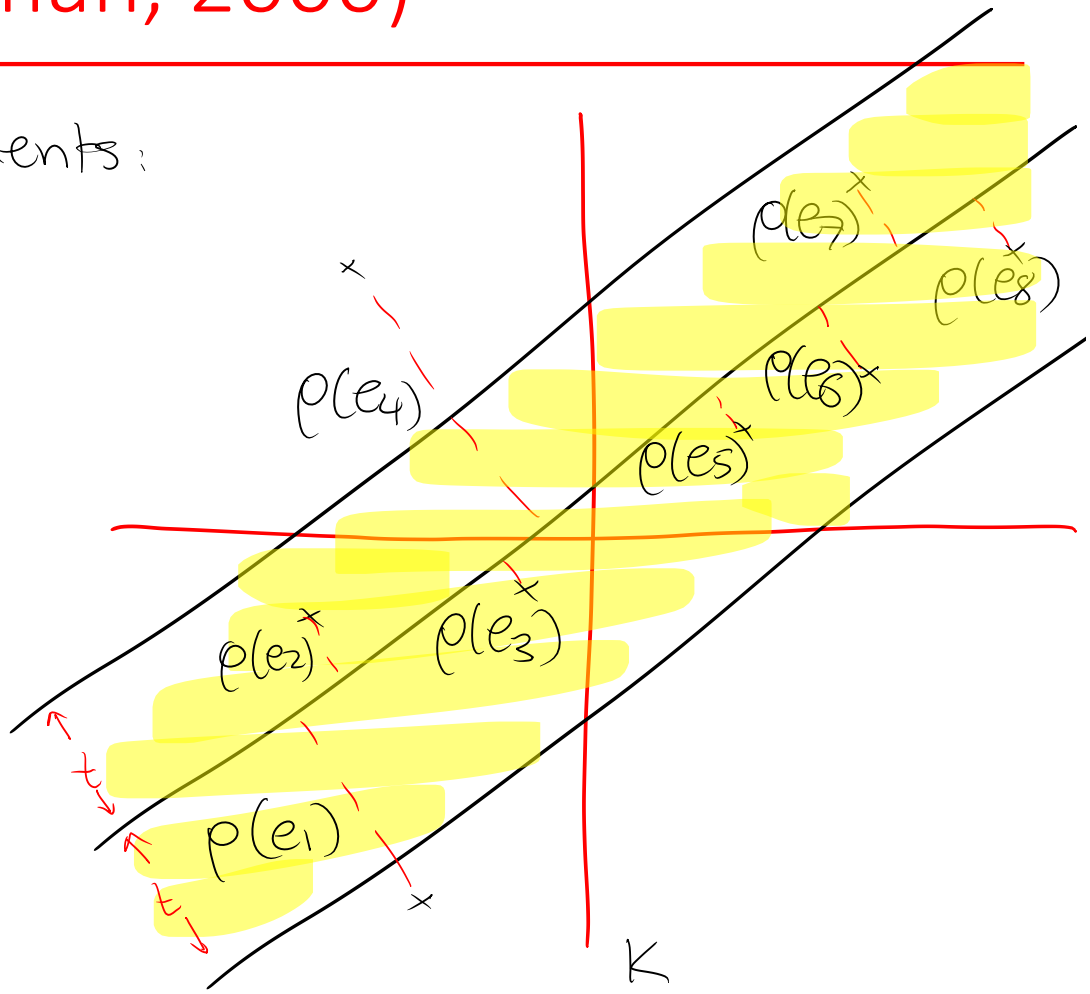
measured "true" error

① inlier errors follow normal distribution

$$e_k \sim \mathcal{N}(0, \Sigma)$$

② outlier errors follow uniform distribution

$$e_k \sim U[-E, E]$$



$$O(p) = \sum_{k=1}^{K} \rho(e_k)$$

$$\rho(e_k) = \begin{cases} e_k^2 & |e_k| < t \\ t^2 & |e_k| \geq t \end{cases}$$

MLESAC (Torr & Zisserman, 2000)

Mixture model for measurements:

$$x_k = \hat{x}_k + e_k$$

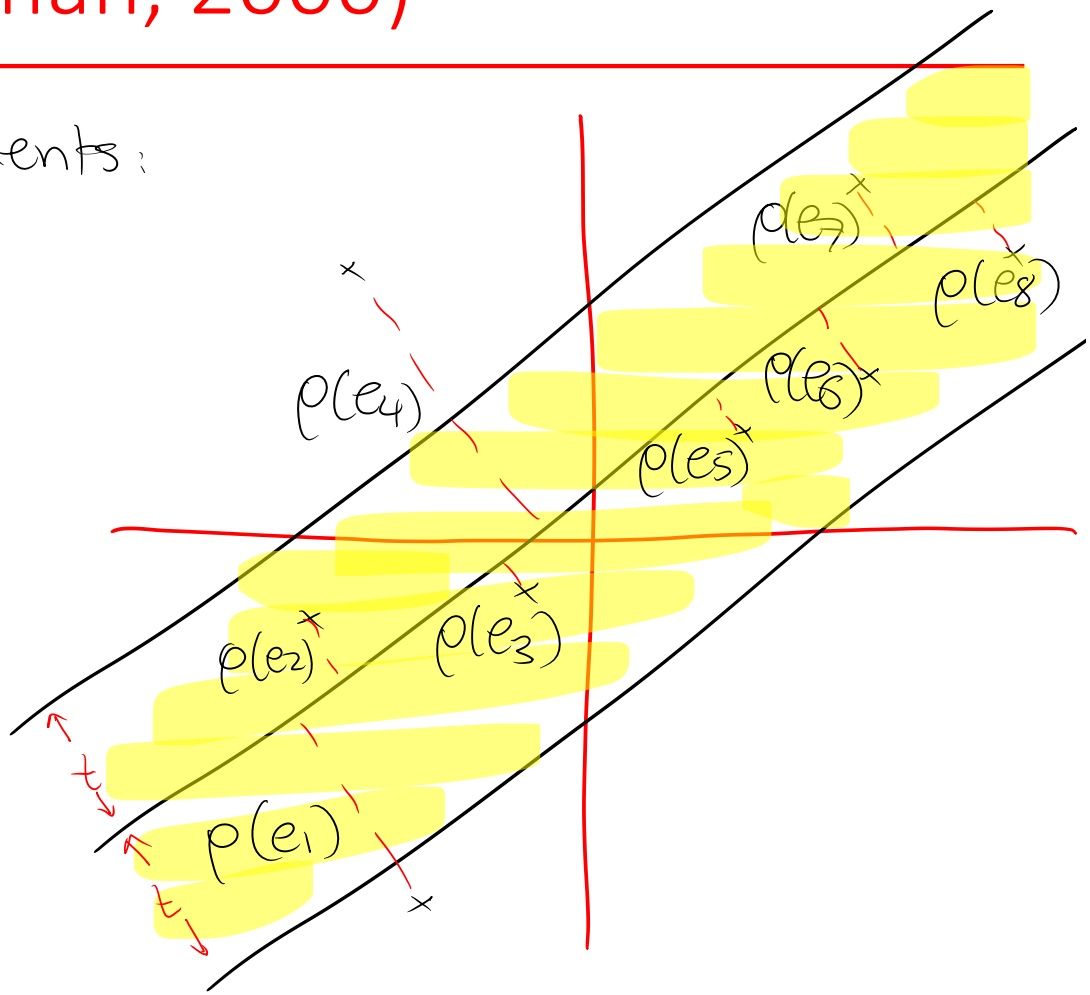
measured "true" error

① inlier errors follow normal distribution

$$e_k \sim \mathcal{N}(0, \Sigma)$$

② outlier errors follow uniform distribution

$$e_k \sim \mathcal{U}[-E, E]$$



③ mixing probabilities

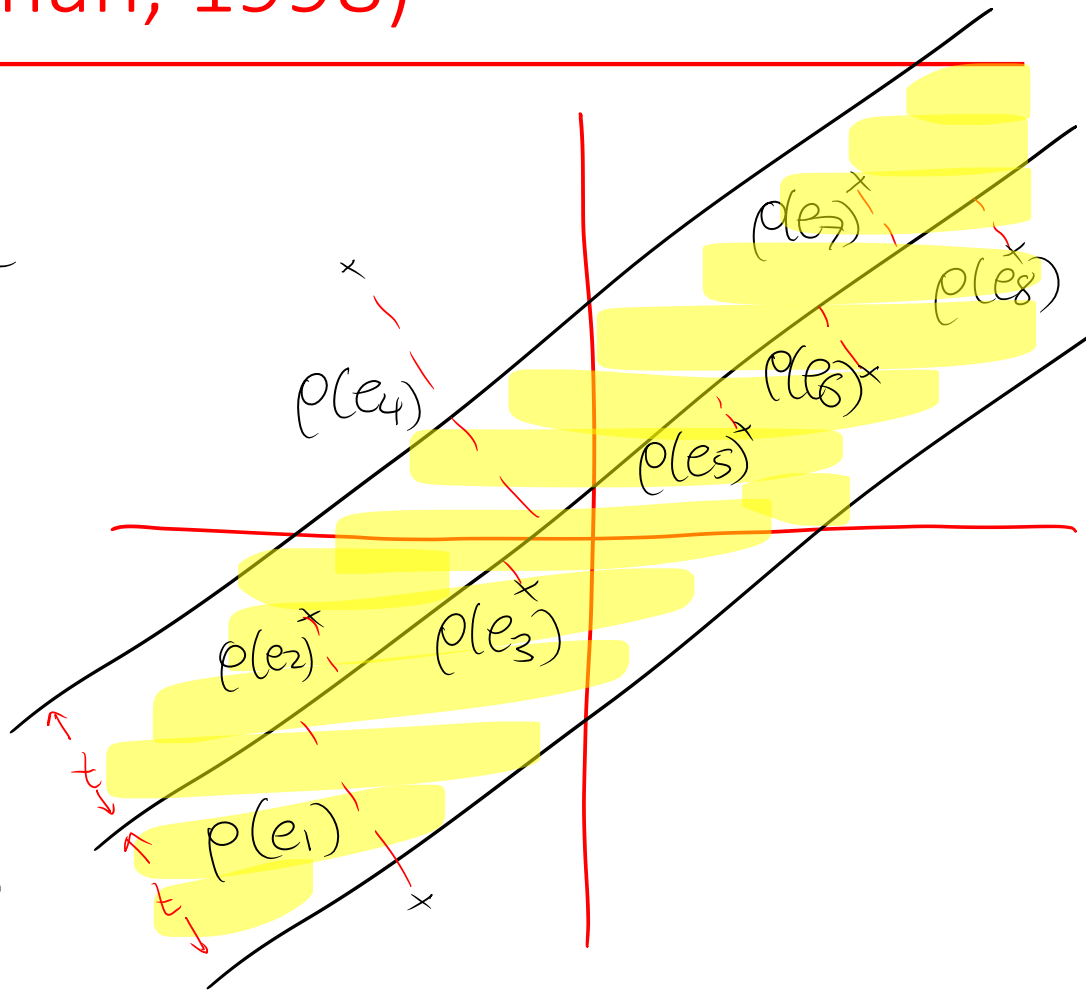
must be estimated

γ for inliers

$1-\gamma$ for outliers

MLESAC (Torr & Zisserman, 1998)

1. Repeat the following:
   a. Select a random sample of minimal size for model fitting
   b. Fit the model
   c. Estimate γ and the log-likelihood using EM
2. Select the best solution over all samples
3. Minimize the robust cost functional over datapoints



MLESAC (Torr & Zisserman, 1998)

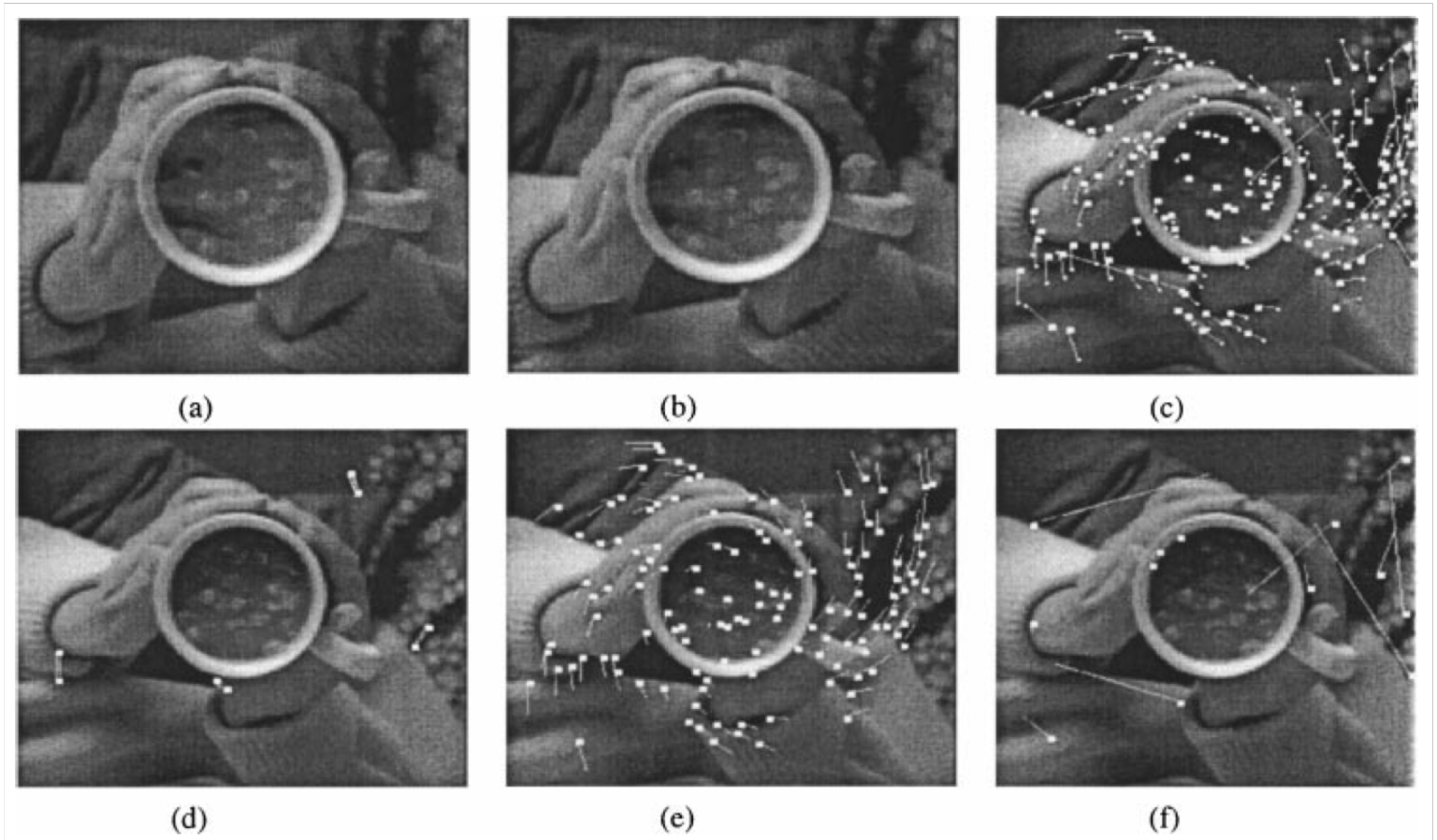


FIG. 5. (a) First image and (b) second image of a cyclotorsion sequence, i.e., rotation of the camera about the optic axis only, combined with image zoom. The camera motion is composed of a large cyclotorsion. (c) matches, (d) basis selected, (e) inliers, and (f) outliers for fitting a projectivity.

To probe further…

Two example applications of probabilistic mixture modeling

- Chuang et al., "A Bayesian Approach to Digital Matting," Proc. CVPR 2001

- Zoran and Weiss, "From learning Models of Natural Image Patches to Whole Image Restoration," Proc. ICCV 2011