Gaussian Mixtures and the EM Algorithm

Reading: Chapter 7.4, Prince book

Review: The Gaussian Distribution

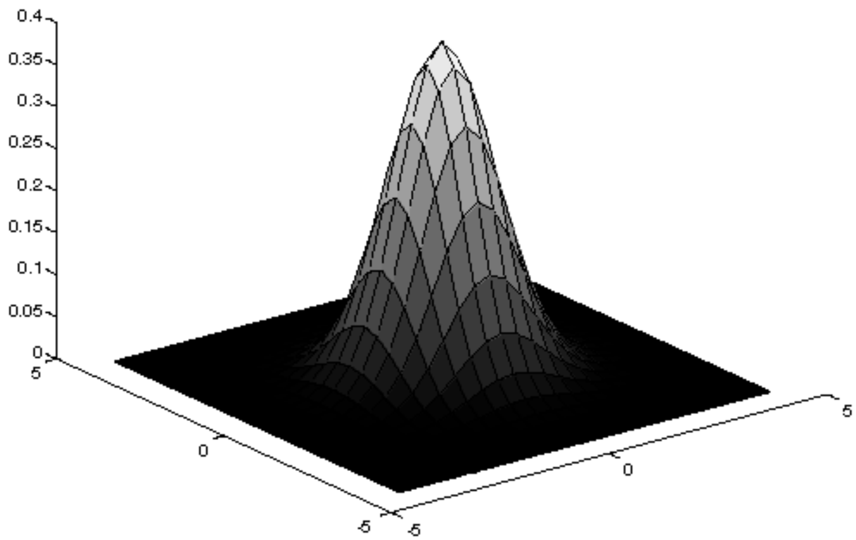
- Multivariate Gaussian

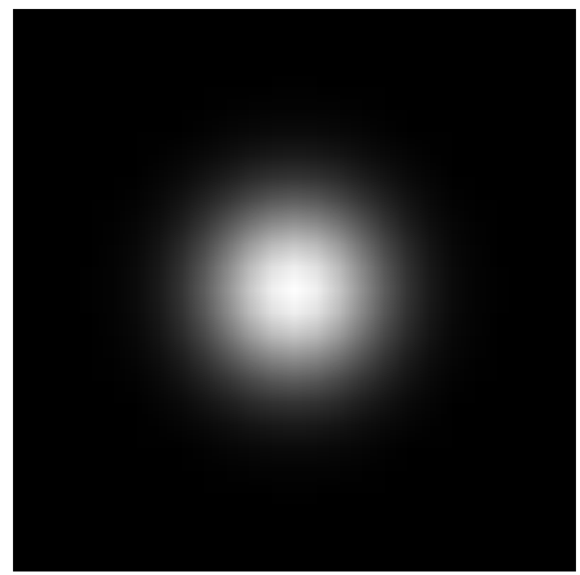
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{((2\pi)^d|\boldsymbol{\Sigma}|)^{1/2}} \exp\underbrace{\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}}_{\text{Evaluates to a number}}$$

mean
dx1 vector

covariance
dxd matrix

Isotropic (spherical) if covariance is diag(σ^2, σ^2,..., σ^2)





Bishop, 2003

Likelihood Function

- Data set

$$D = \{\mathbf{x}_n\} \quad n = 1, \ldots, N$$

- Assume observed data points generated independently

$$p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Viewed as a function of the parameters, this is known as the *likelihood function*

Bishop, 2003

Maximum Likelihood

- Set the parameters by maximizing the likelihood function
- Equivalently maximize the log likelihood

$$\ln p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{Nd}{2}\ln(2\pi) - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})$$

Bishop, 2003

Maximum Likelihood Solution

- Maximizing w.r.t. the mean gives the *sample mean*

$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n$$

- Maximizing w.r.t covariance gives the *sample covariance*

$$\boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\mathrm{T}}$$

Note: if N is small you want to divide by N-1 when computing sample covariance to get an unbiased estimate.

Bishop, 2003

Comments

Gaussians are well understood and easy to estimate

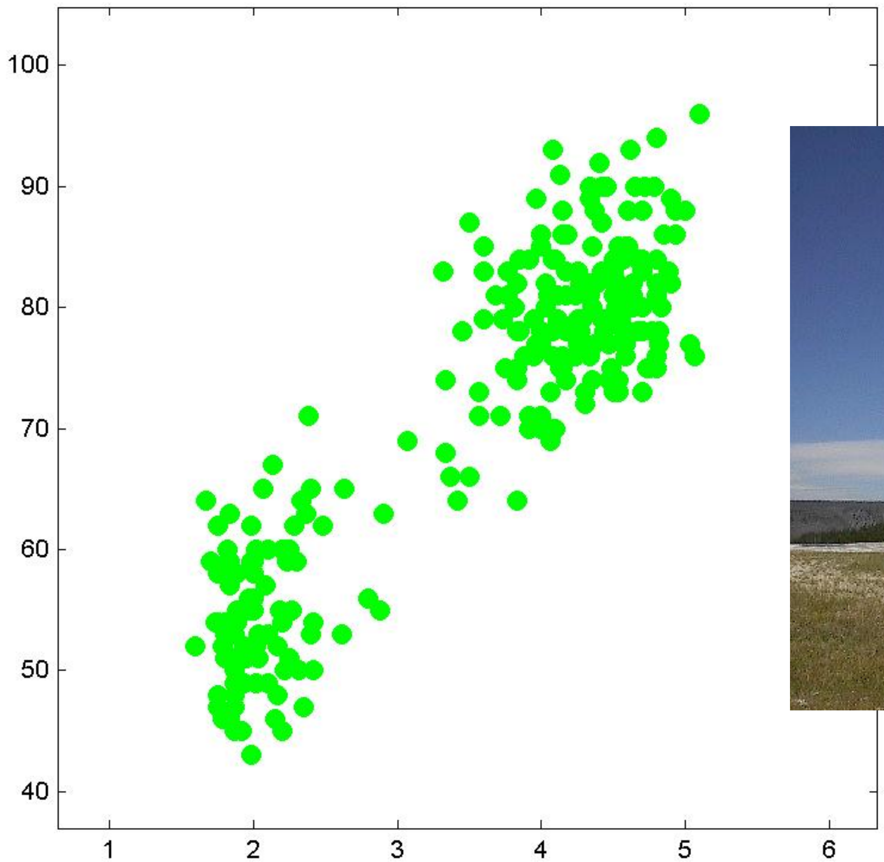
However, they are unimodal, thus cannot be used to represent inherently multimodal datasets

Fitting a single Gaussian to a multimodal dataset is likely to give a mean value in an area with low probability, and to overestimate the covariance.

Robert Collins
CSE586

Old Faithful Data Set

Time between eruptions (minutes)



Duration of eruption (minutes)

Bishop, 2004

Idea: Use a Mixture of Gaussians

- Convex Combination of Distributions

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Normalization and positivity require

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

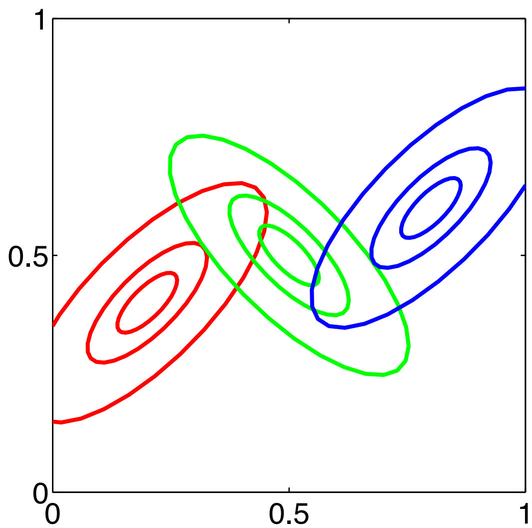
- Can interpret the mixing coefficients as prior probabilities

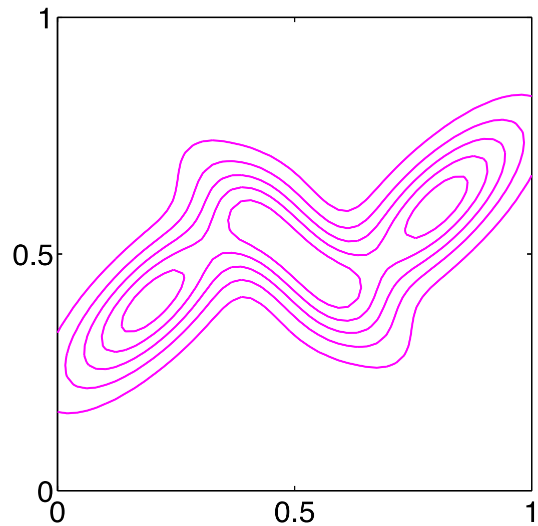
$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k) p(\mathbf{x}|k)$$

Bishop, 2003

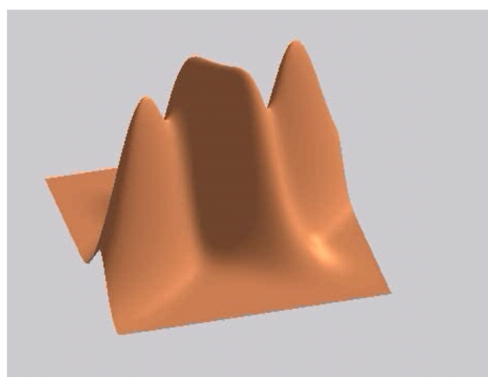
Example: Mixture of 3 Gaussians

$p(\mathbf{x}|k)$





$p(\mathbf{x})$

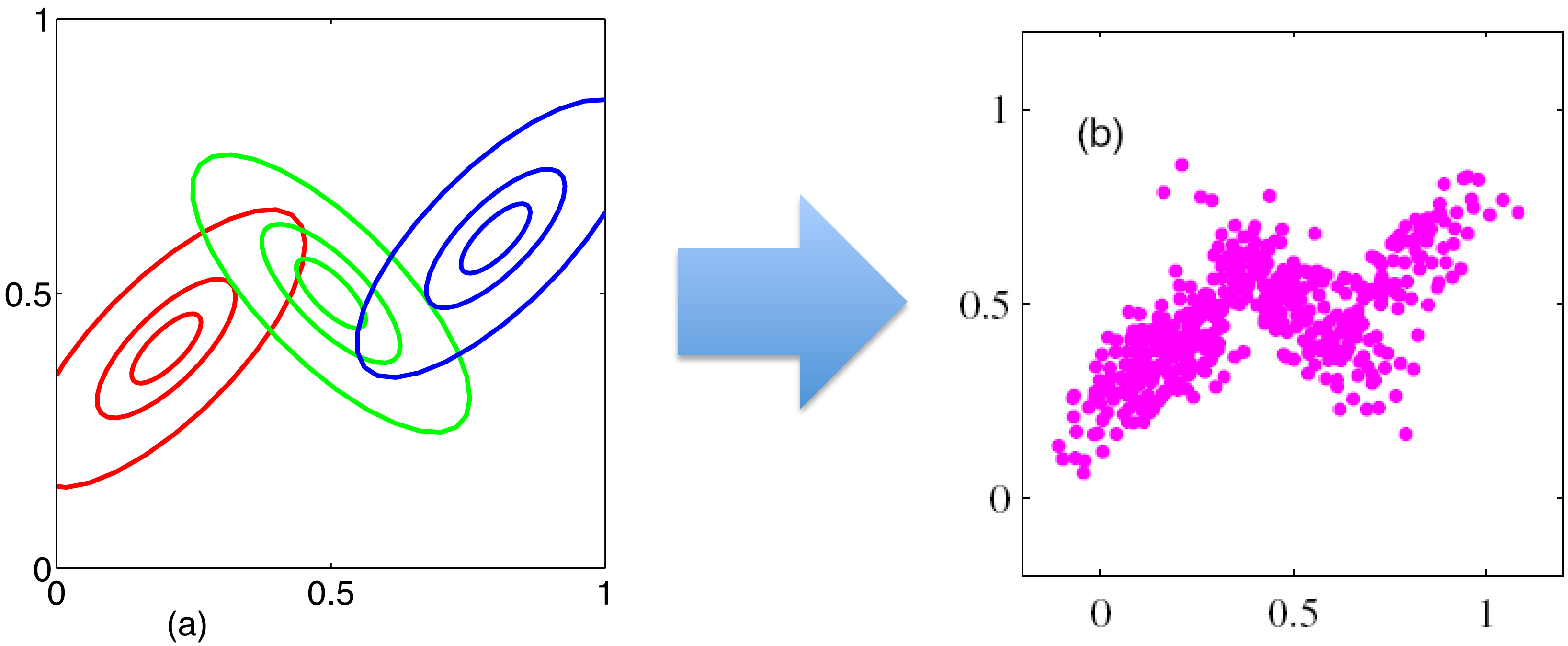


$p(\mathbf{x})$

Bishop, 2004

Aside: Sampling from a Mixture Model

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



Aside: Sampling from a Mixture Model

Generate u = uniform random number between 0 and 1

If $u < \pi_1$

generate $x \sim N(x \mid \mu_1, \Sigma_1)$

elseif $u < \pi_1 + \pi_2$

generate $x \sim N(x \mid \mu_2, \Sigma_2)$

⋮

elseif $u < \pi_1 + \pi_2 + \ldots + \pi_{K-1}$

generate $x \sim N(x \mid \mu_{K-1}, \Sigma_{K-1})$

else

generate $x \sim N(x \mid \mu_K, \Sigma_K)$

MLE of Mixture Parameters

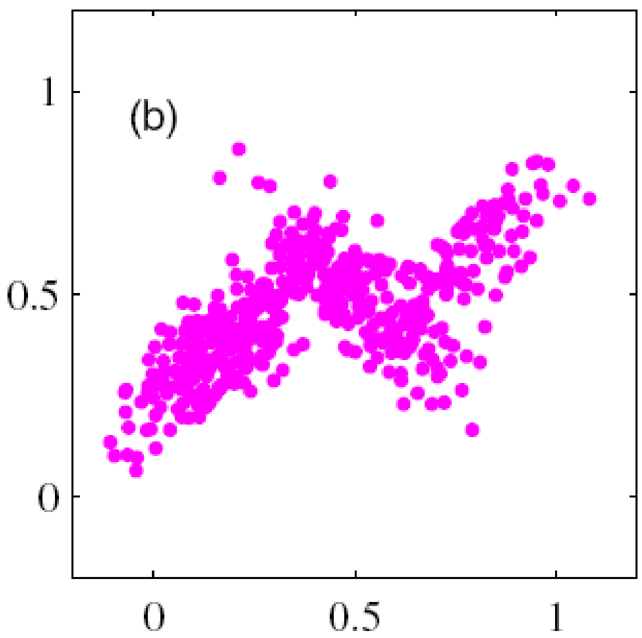
- However, MLE of mixture parameters is HARD!
- Joint distribution:

$$p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \underbrace{\sum_{k=1}^{K}}_{\text{Uh-oh, log of a sum}} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

EM Algorithm



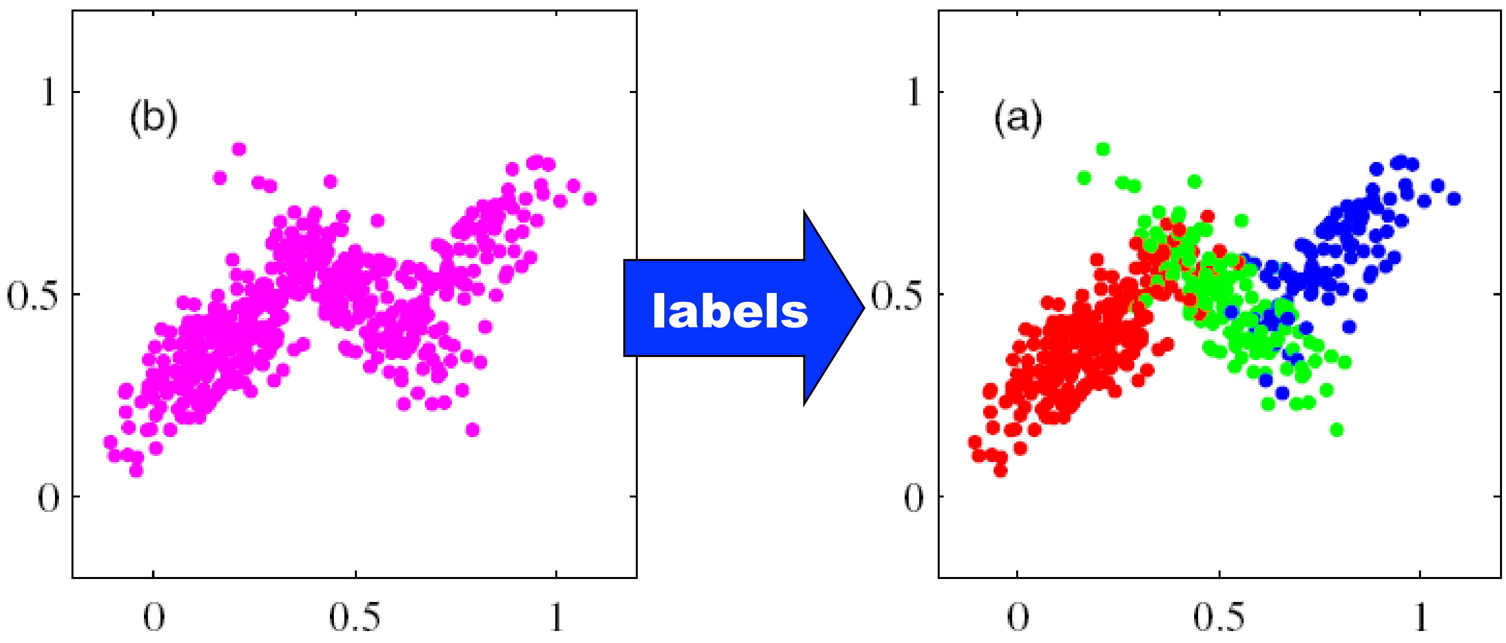
What makes this estimation problem hard?

1) It is a mixture, so log-likelihood is messy

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

2) We don’t directly see what the underlying process is

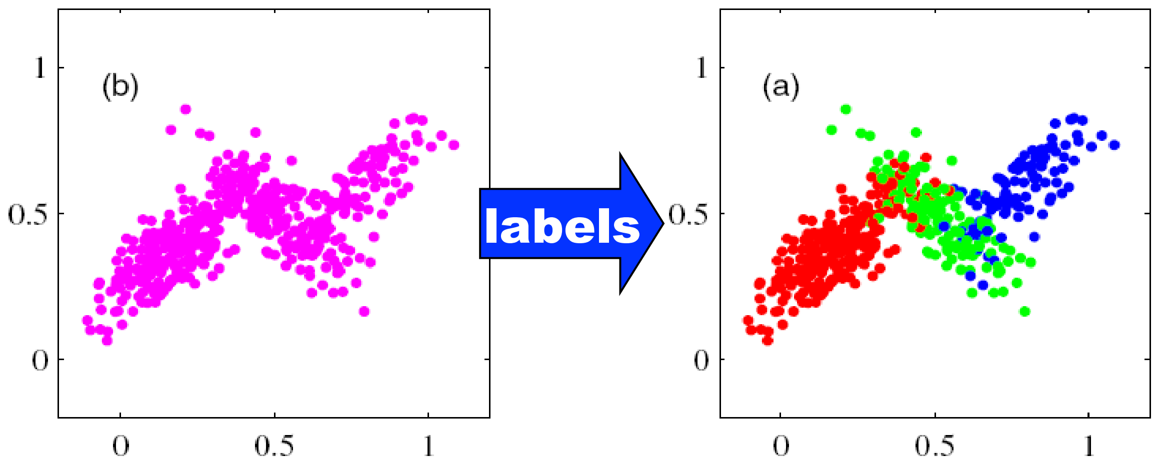
EM Algorithm



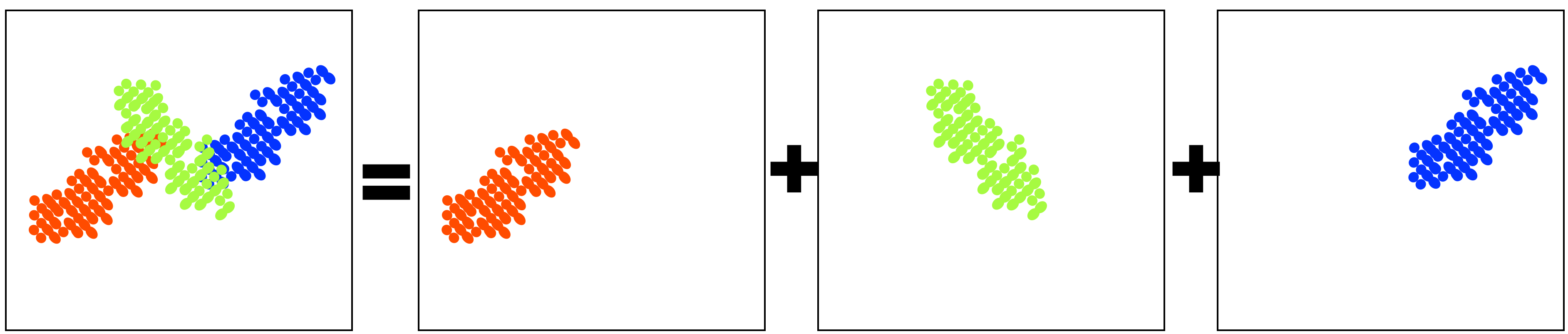
Suppose some oracle told us which point comes from which Gaussian.

How? By providing a “latent” variable z_nk which is 1 if point n comes from the kth component Gaussian, and 0 otherwise (a 1 of K representation)

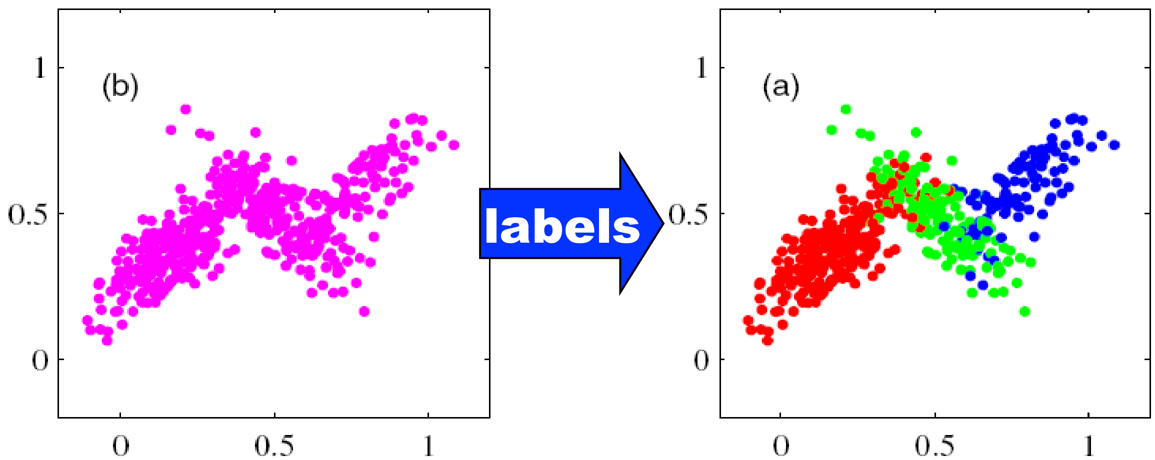
EM Algorithm



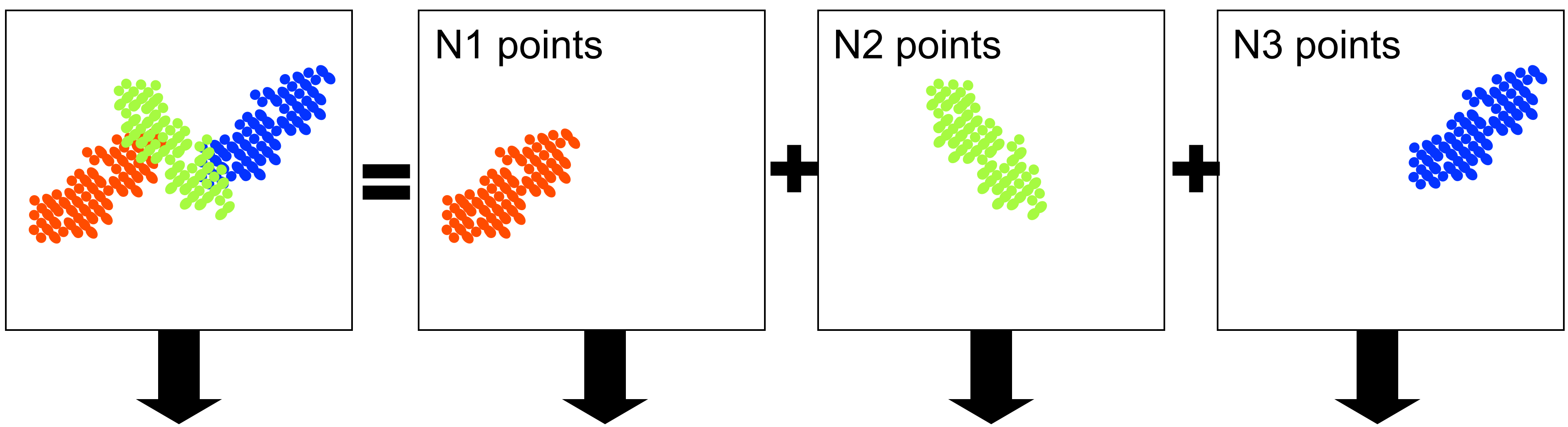
This lets us recover the underlying generating process decomposition:



EM Algorithm



And we can easily estimate each Gaussian, along with the mixture weights!



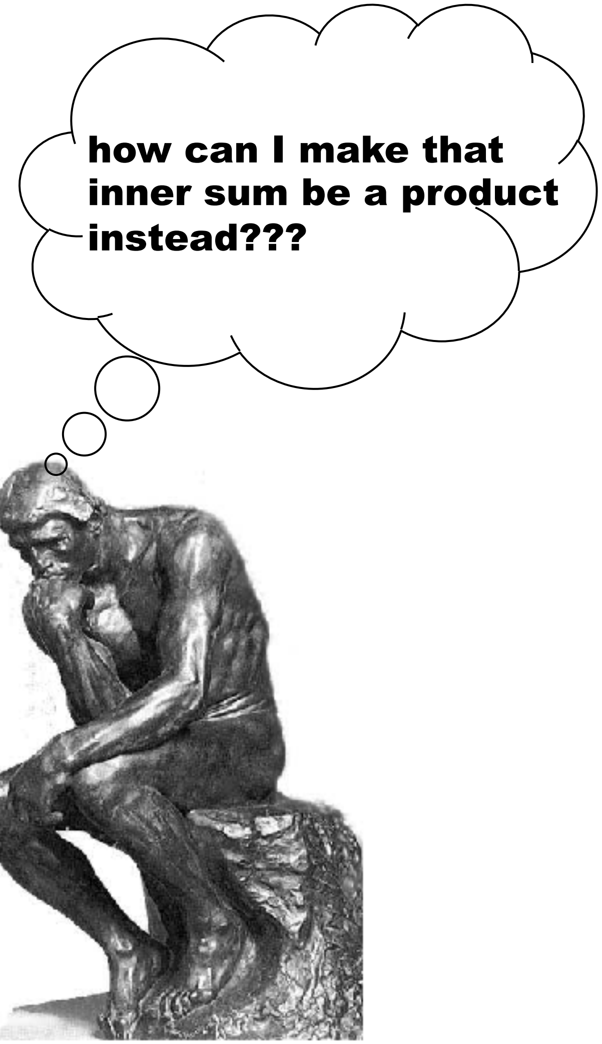
estimate
pi_k = Nk / N

estimate
mu_1, Sigma_1

estimate
mu_2, Sigma_2

estimate
mu_3, Sigma_3

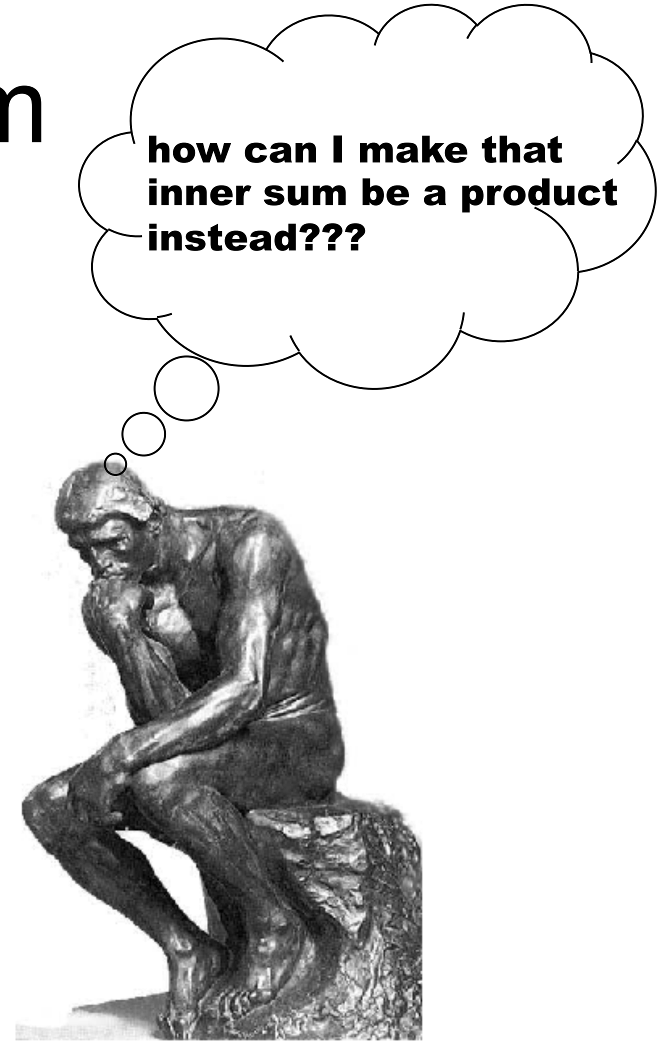
EM Algorithm



Remember that this was a problem...

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

EM Algorithm



Remember that this was a problem...

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Again, if an oracle gave us the values of the latent variables (component that generated each point) we could work with the complete log likelihood

$$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

and the log of that looks much better!

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left\{ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}.$$

Latent Variable View

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\} .$$

note: for a given n, there are k of these latent variables, and only ONE of them is 1 (all the rest are 0)

Latent Variable View

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}.$$

note: for a given n, there are k of these latent variables, and only ONE of them is 1 (all the rest are 0)

This is thus equivalent to

$$\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \pi_1 + \ln \mathcal{N}(x_n|\mu_1, \Sigma_1)$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \pi_2 + \ln \mathcal{N}(x_n|\mu_2, \Sigma_2) \quad + \quad \cdots \quad +$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \pi_K + \ln \mathcal{N}(x_n|\mu_K, \Sigma_K)$$

Latent Variable View

$$\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \pi_1 + \sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \mathcal{N}(x_n|\mu_1, \Sigma_1)$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \pi_2 + \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \mathcal{N}(x_n|\mu_2, \Sigma_2)$$

$$+ \quad \bullet\bullet\bullet \quad +$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \pi_K + \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \mathcal{N}(x_n|\mu_K, \Sigma_K)$$

Robert Collins
CSE586

Latent Variable View

$$\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \pi_1 + \boxed{\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \mathcal{N}(x_n|\mu_1, \Sigma_1)}$$

can be estimated separately

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \pi_2 + \boxed{\sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \mathcal{N}(x_n|\mu_2, \Sigma_2)}$$

can be estimated separately

$$+ \ \cdots \ +$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \pi_K + \boxed{\sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \mathcal{N}(x_n|\mu_K, \Sigma_K)}$$

can be estimated separately

Latent Variable View

$$\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \pi_1 + \boxed{\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \mathcal{N}(x_n|\mu_1,\Sigma_1)}$$

can be estimated separately

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \pi_2 + \boxed{\sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \mathcal{N}(x_n|\mu_2,\Sigma_2)}$$

can be estimated separately

$$+ \cdots +$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \pi_K + \boxed{\sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \mathcal{N}(x_n|\mu_K,\Sigma_K)}$$

can be estimated separately

these are coupled because the mixing weights all sum to 1, but it is no big deal to solve

EM Algorithm

Unfortunately, oracles don't exist (or if they do, they won't talk to us)

So we don't know values of the the z_nk variables

What EM proposes to do:

1) compute p(Z|X,theta), the posterior distribution over z_nk, given our current best guess at the values of theta

2) compute the expected value of the log likelihood ln(p(X,Z|theta)) with respect to the distribution p(Z|X,theta)

3) find theta_new that maximizes that function. This is our new best guess at the values of theta.

4) iterate...

Insight

Since we don’t know the latent variables, we instead take the expected value of the log likelihood with respect to their posterior distribution P(z|x,theta). In the GMM case, this is equivalent to “softening” the binary latent variables to continuous ones (the expected values of the latent variables)

$$\ln p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \underbrace{z_{nk}}_{\text{unknown discrete value 0 or 1}} \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

$$\mathrm{E}_{\mathbf{z}}[\ln p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})] = \sum_{n=1}^{N} \sum_{k=1}^{K} \underbrace{\gamma_k(\mathbf{x}_n)}_{\text{known continuous value between 0 and 1}} \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

Where $\gamma_j(\mathbf{x}_n)$ is $P(z_{nk} = 1)$

Insight

So now, after replacing the binary latent variables with their continuous expected values:

all points contribute to the estimation of all components

each point has unit mass to contribute, but splits it across the K components

the amount of weight a point contributes to a component is proportional to the relative likelihood that the point was generated by that component

Latent Variable View (with an oracle)

$$\boxed{\sum_{\substack{\text{all n for which}\\ z_{n,1}=1}} \ln \pi_1 + \sum_{\substack{\text{all n for which}\\ z_{n,2}=1}} \ln \pi_2 + \cdots + \sum_{\substack{\text{all n for which}\\ z_{n,K}=1}} \ln \pi_K}$$

$$+ \boxed{\sum_{\substack{\text{all n for which}\\ z_{n,1}=1}} \ln \mathcal{N}(x_n|\mu_1,\Sigma_1)}$$

can be estimated separately

$$+ \boxed{\sum_{\substack{\text{all n for which}\\ z_{n,2}=1}} \ln \mathcal{N}(x_n|\mu_2,\Sigma_2)}$$

can be estimated separately

$$+ \boxed{\sum_{\substack{\text{all n for which}\\ z_{n,K}=1}} \ln \mathcal{N}(x_n|\mu_K,\Sigma_K)}$$

can be estimated separately

these are coupled because the mixing weights all sum to 1, but it is no big deal to solve

Latent Variable View (with EM, $\gamma^{i}_{n,k}$ a constant at iteration i)

$$\boxed{\begin{array}{l} \sum_N \sum_K \gamma^{i}_{n,k} \ln \pi_1 \\ + \sum_N \sum_K \gamma^{i}_{n,k} \ln \pi_2 \\ + \dots + \\ + \sum_N \sum_K \gamma^{i}_{n,k} \ln \pi_K \end{array}}$$

$$+ \boxed{\sum_N \sum_K \gamma^{i}_{n,k} \ln \mathcal{N}(x_n | \mu_1, \Sigma_1)}$$ can be estimated separately

$$+ \boxed{\sum_N \sum_K \gamma^{i}_{n,k} \ln \mathcal{N}(x_n | \mu_2, \Sigma_2)}$$ can be estimated separately

$$+ \boxed{\sum_N \sum_K \gamma^{i}_{n,k} \ln \mathcal{N}(x_n | \mu_K, \Sigma_K)}$$ can be estimated separately

these are coupled because the mixing weights all sum to 1, but it is no big deal to solve

Robert Collins
CSE586

EM Algorithm for GMM

E

$$\gamma_j(\mathbf{x}_n) = \frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_k \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

ownership weights

M

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

means

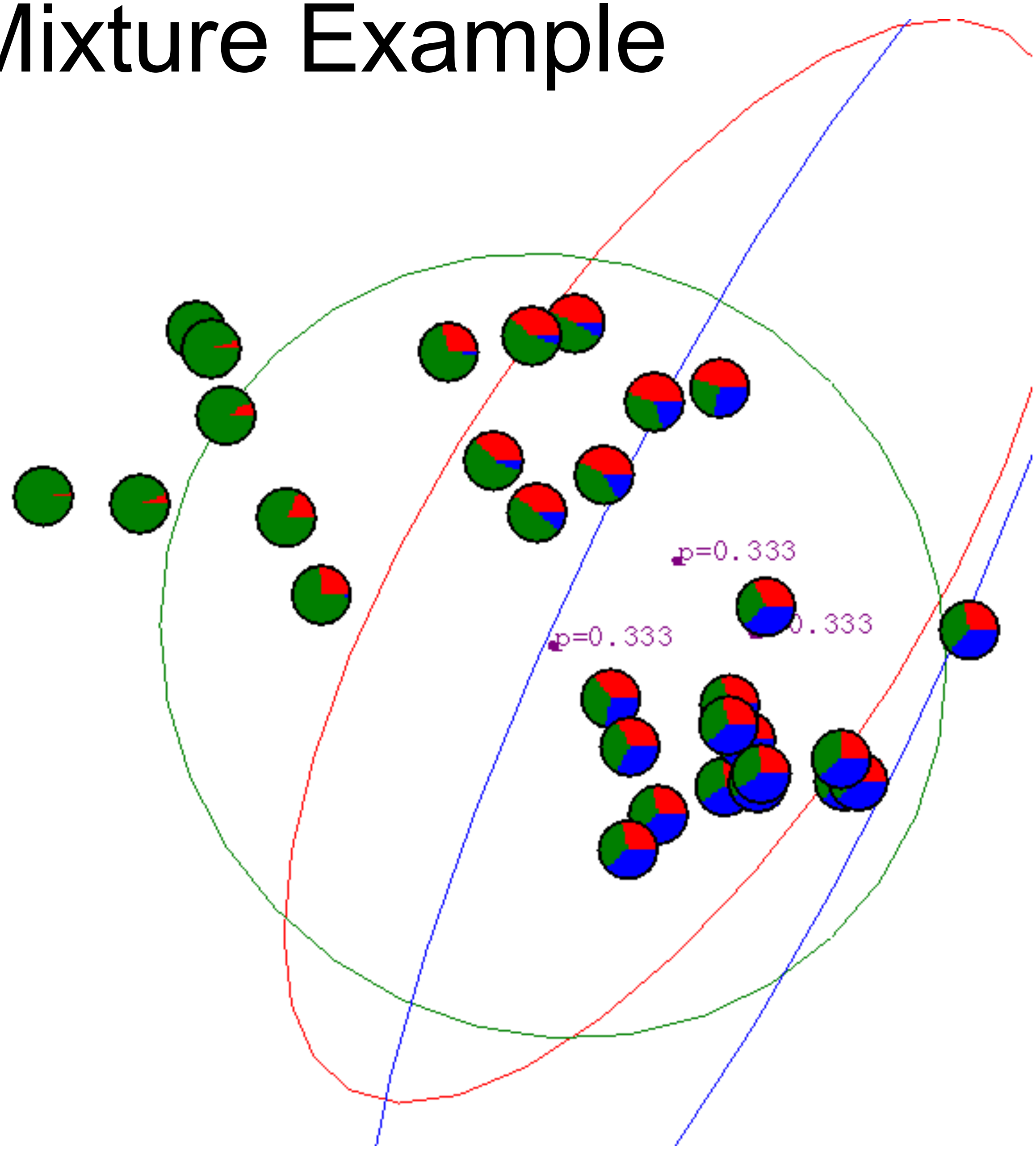
$$\boldsymbol{\Sigma}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_j)(\mathbf{x}_n - \boldsymbol{\mu}_j)^\top}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

covariances

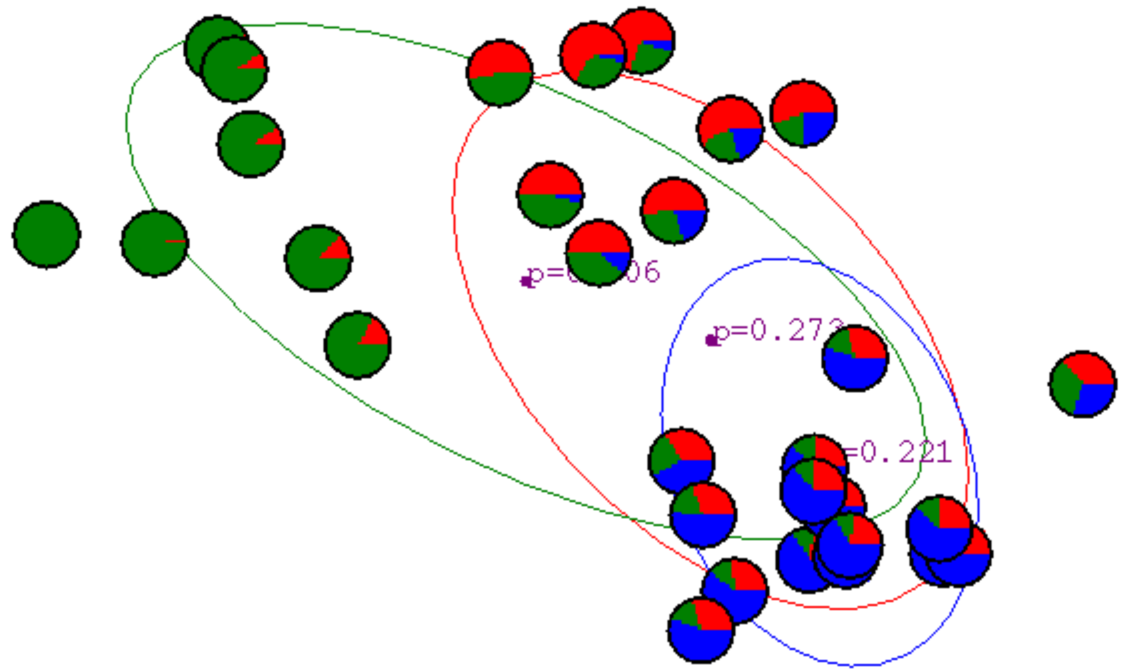
$$\pi_j = \frac{1}{N}\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)$$

mixing probabilities

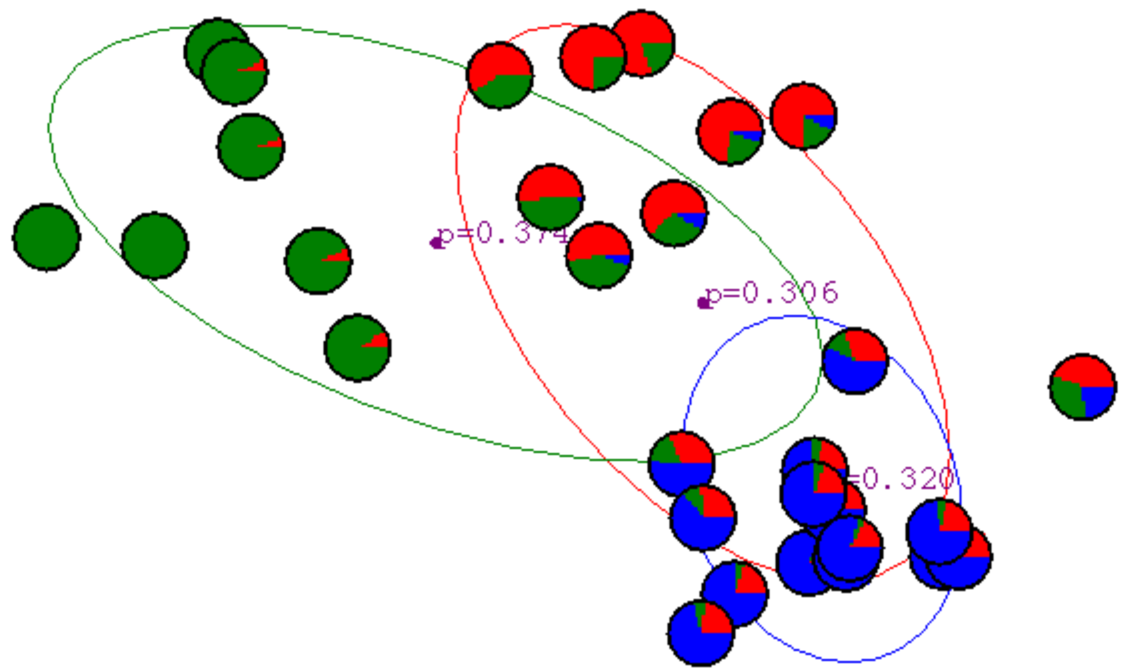
Gaussian Mixture Example



After first iteration

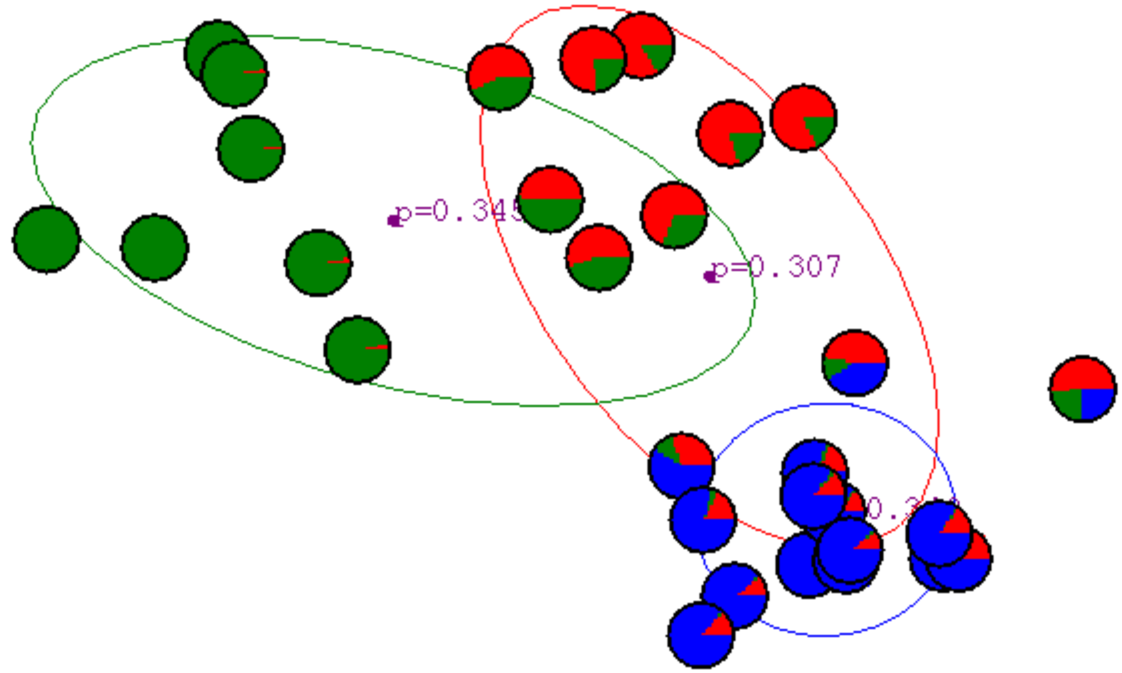


After 2nd iteration

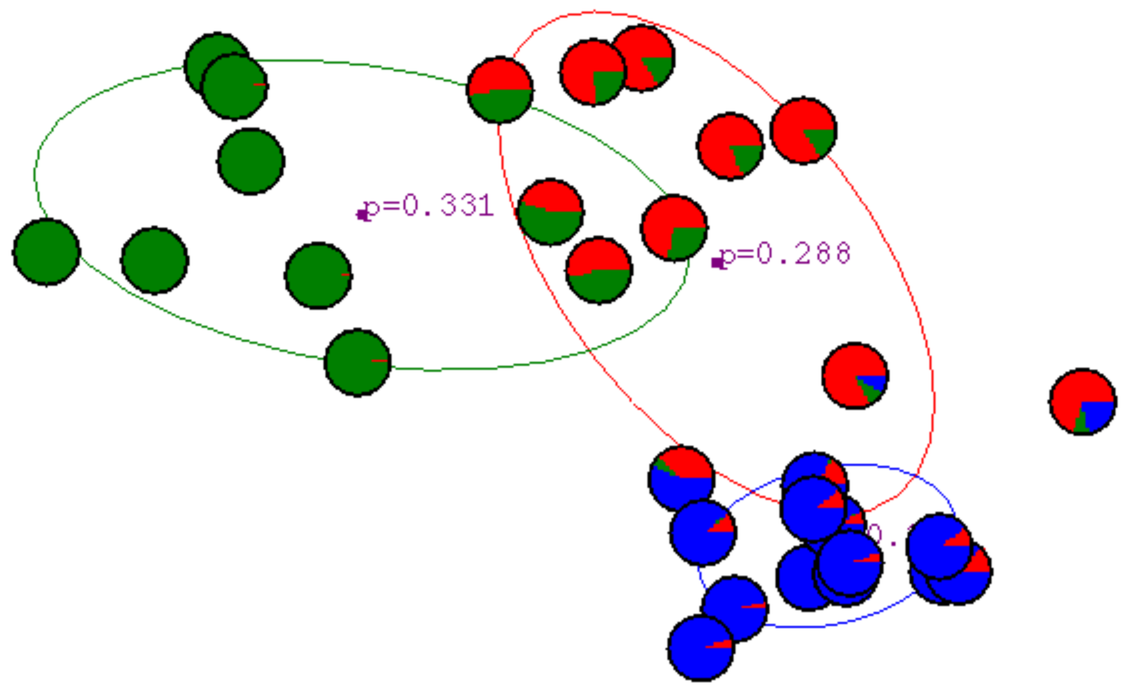


Robert Collins
CSE586

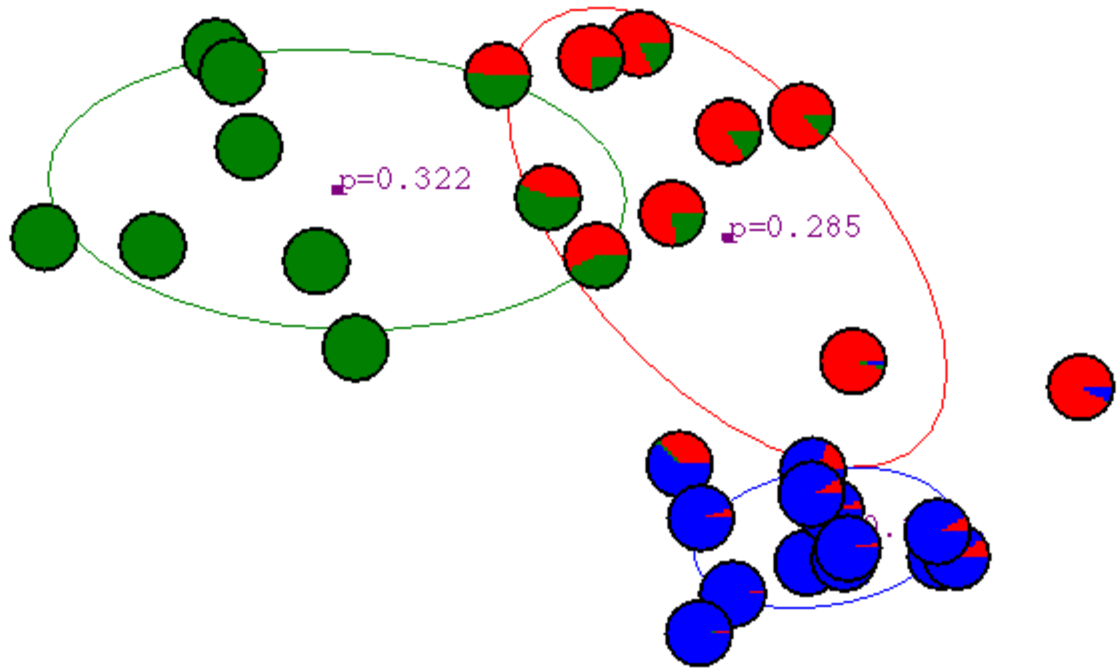
After 3rd iteration



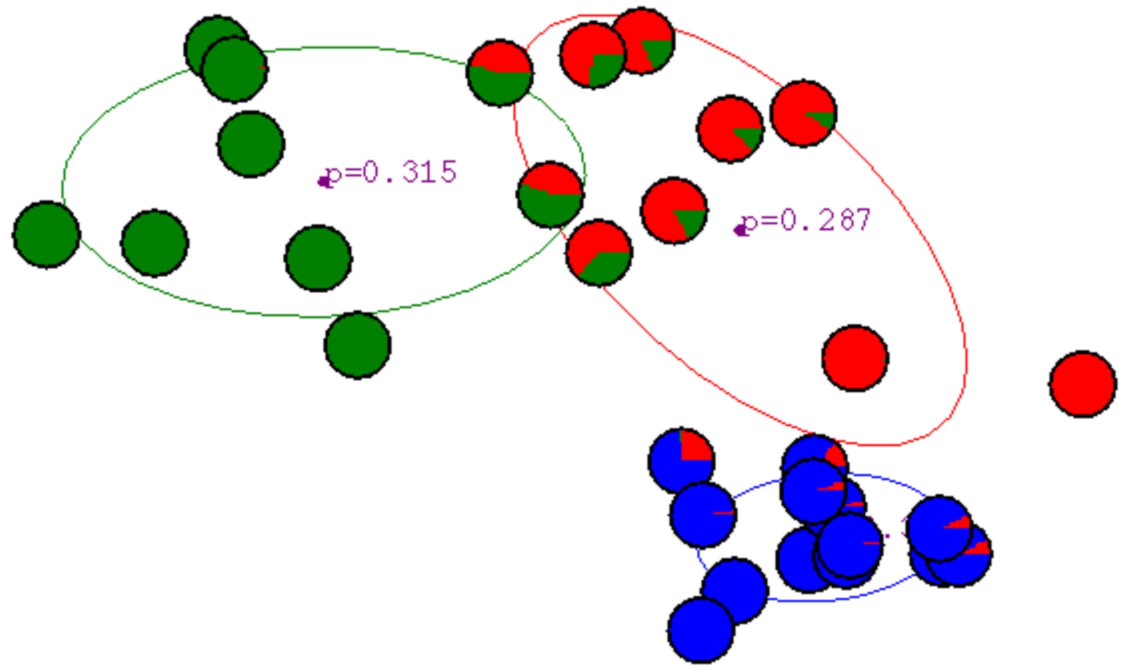
After 4th iteration



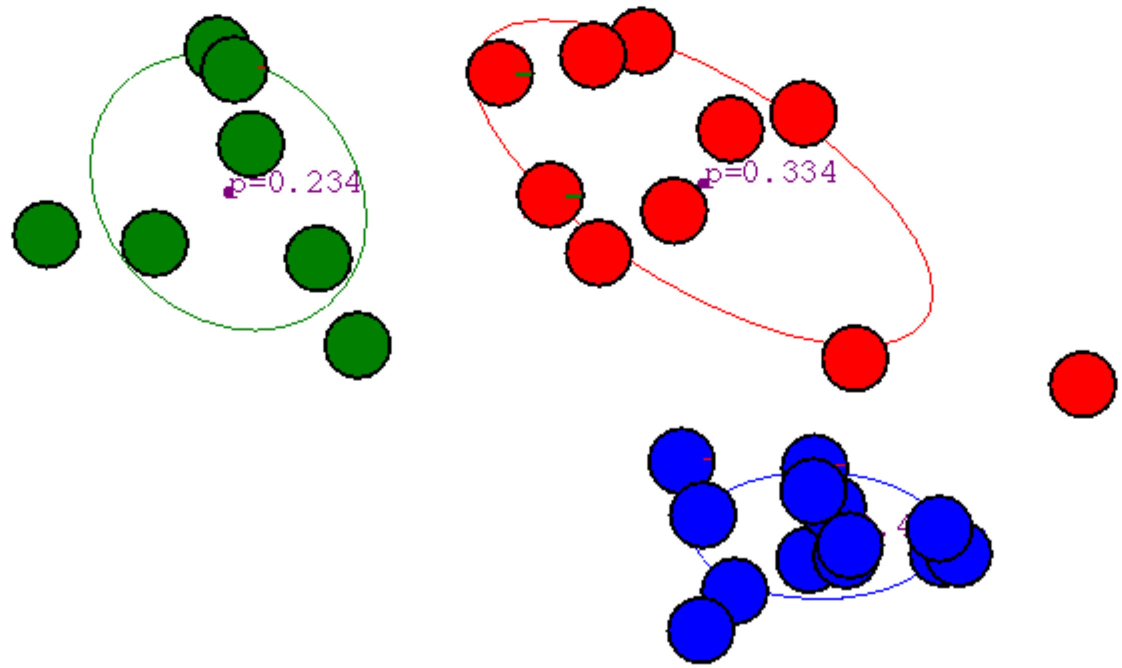
After 5th iteration



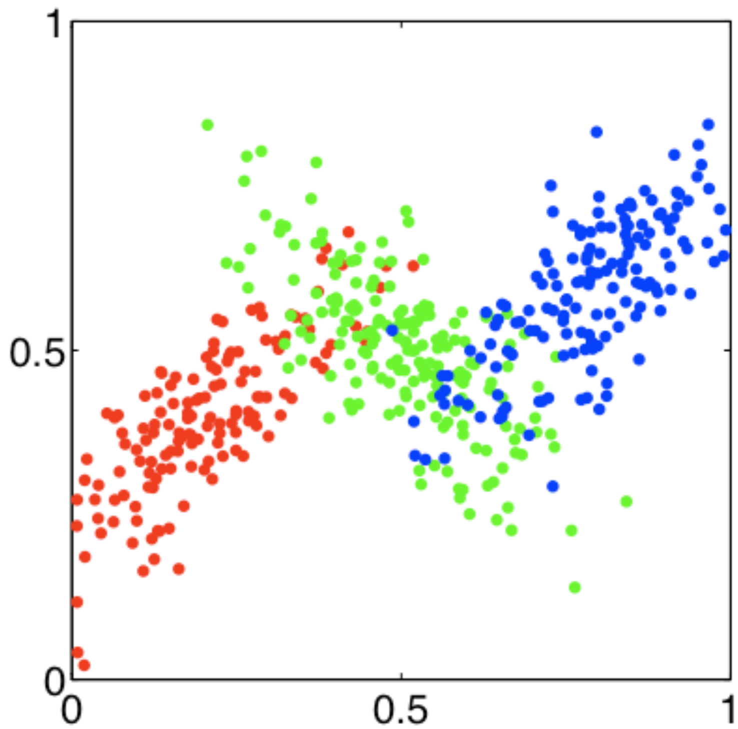
After 6th iteration



After 20th iteration

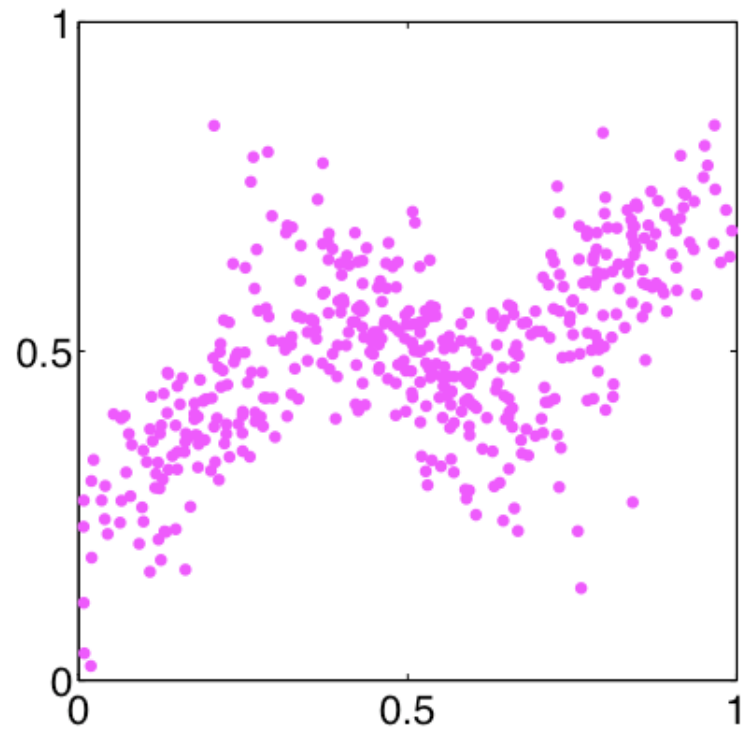


Recall: Labeled vs Unlabeled Data



labeled

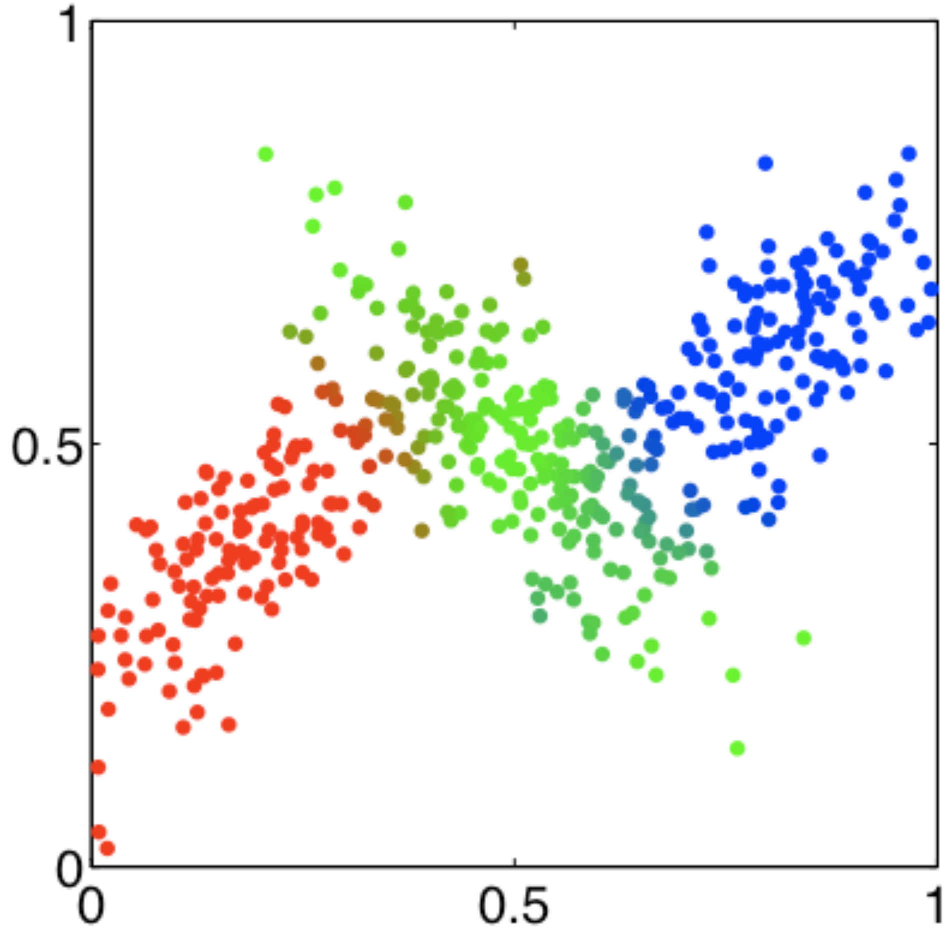
Easy to estimate params
(do each color separately)



unlabeled

Hard to estimate params
(we need to assign colors)

EM produces a “Soft” labeling



each point makes a weighted contribution
to the estimation of ALL components

Review of EM for GMMs

E

$$\gamma(z_{nj}) = \frac{\pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_k \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

ownership weights
(soft labels)

M

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma(z_{nj})\, \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nj})}$$

means

$$\boldsymbol{\Sigma}_j = \frac{\sum_{n=1}^{N} \gamma(z_{nj})\, (\mathbf{x}_n - \boldsymbol{\mu}_j)(\mathbf{x}_n - \boldsymbol{\mu}_j)^{\top}}{\sum_{n=1}^{N} \gamma(z_{nj})}$$

covariances

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nj})$$

mixing weights

Alternate E and M steps to convergence.

From EM to K-Means

Alternative explanation of K-means!

- Fix all mixing weights to 1/K
  [drop out of the estimation]
- Fix all covariances to σ^2 I
  [drop out of the estimation so we only have to estimate the means; each Gaussian likelihood becomes inversely proportional to distance from a mean]
- Take limit as σ^2 goes to 0
  [this forces soft weights to become binary]

From EM to K-Means

E $$\gamma(z_{nj}) = \frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j)}{\sum_k \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}$$

ownership weights (soft labels)

$$= \frac{\exp(-\frac{1}{2\sigma^2}||x_n - \mu_j||^2)}{\sum_{k=1}^{K} \exp(-\frac{1}{2\sigma^2}||x_n - \mu_k||^2)}$$

after fixing mixing weights and covariances as described on last slide

now divide top and bottom by $\exp(-\frac{1}{2\sigma^2} d^2_{min})$ where $d^2_{min} = \min_k ||x_n - \mu_k||^2$

and take limit as σ^2 goes to 0

$$\gamma(z_{nj}) = z_{nj} = \begin{cases} 1 & \text{if } \mu_j \text{ is closest mean to } x_n \\ 0 & \text{otherwise} \end{cases}$$

hard labels, as in the K-means algorithm

K-Means Algorithm

- Given N data points $x_1, x_2, \dots, x_N$
- Find K cluster centers $\mu_1, \mu_2, \dots, \mu_K$ to minimize $\sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \|x_n - \mu_j\|^2$
  (z_{nk} is 1 if point n belongs to cluster k; 0 otherwise)
- Algorithm:
  – initialize K cluster centers $\mu_1, \mu_2, \dots, \mu_K$
  – repeat
    - set z_{nk} labels to assign each point to closest cluster center
    - revise each cluster center μ_j to be center of mass of points in that cluster $\mu_j = \frac{\sum_{n=1}^{N} z_{nj}\, x_n}{\sum_{n=1}^{N} z_{nj}}$
  – until convergence (e.g. z_{nk} labels don't change)

E

M