

Review and Motivation

We can model and visualize multimodal datasets by using multiple unimodal (Gaussian-like) clusters.

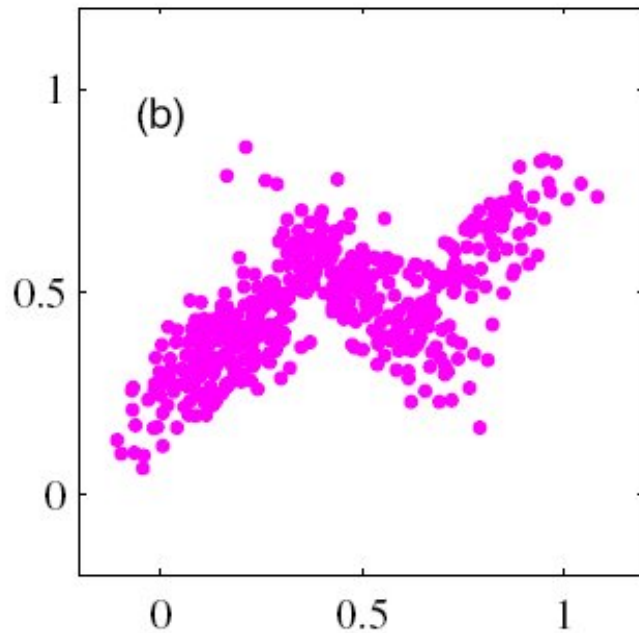
K-means gives us a way of partitioning points into N clusters. Once we know which points go to which cluster, we can estimate a Gaussian mean and covariance for that cluster.

We have introduced the idea of writing what you want to do as a function to be optimized (maximized or minimized). Maximum likelihood estimation to fit mean and covariance parameters of a Gaussian is a good example of this.

Review and Motivation

- want to do MLE of mixture of Gaussian parameters
- But this is hard, because of the summation in the mixture of Gaussian equation (can't take the log of a sum).
- If we knew which point contribute to which Gaussian component, the problem would be a lot easier (we can rewrite so that the summation goes away)
- So... let's guess which point goes with which component, and proceed with the estimation.
- We were unlikely to guess right the first time, but based on our initial estimation of parameters, we can now make a better guess at pairing points with components.
- Iterate
- This is the basic idea underlying the EM algorithm.

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

What is the underlying process?

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k).$$

procedure to generate a mixture of gaussians

for i=1:N

 generate a uniform $U(0,1)$ random number to determine
 which of K components to draw a sample from (based on
 probabilities π_k)

 generate a sample from a Gaussian $\mathcal{N}(\mu_k, \Sigma_k)$

end

EM Algorithm

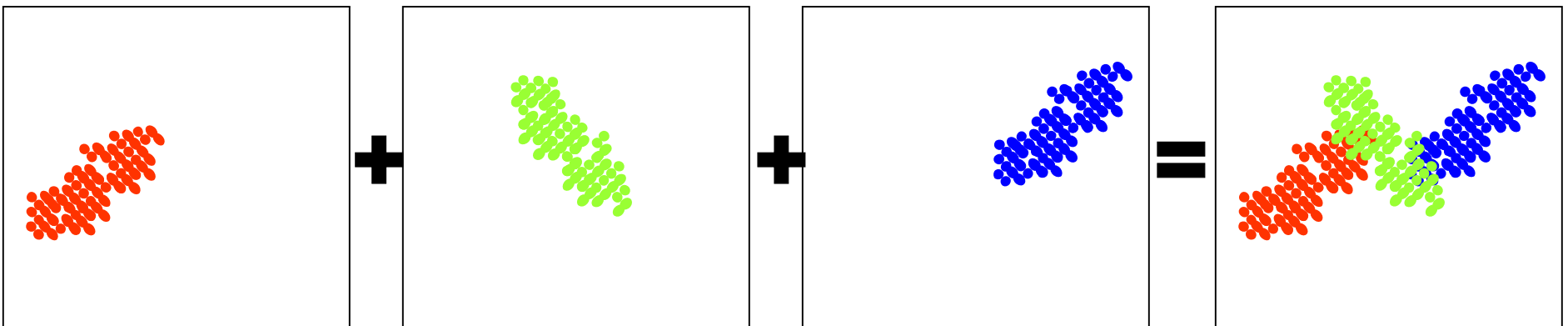
equivalent procedure to generate a mixture of gaussians

for $k=1:K$

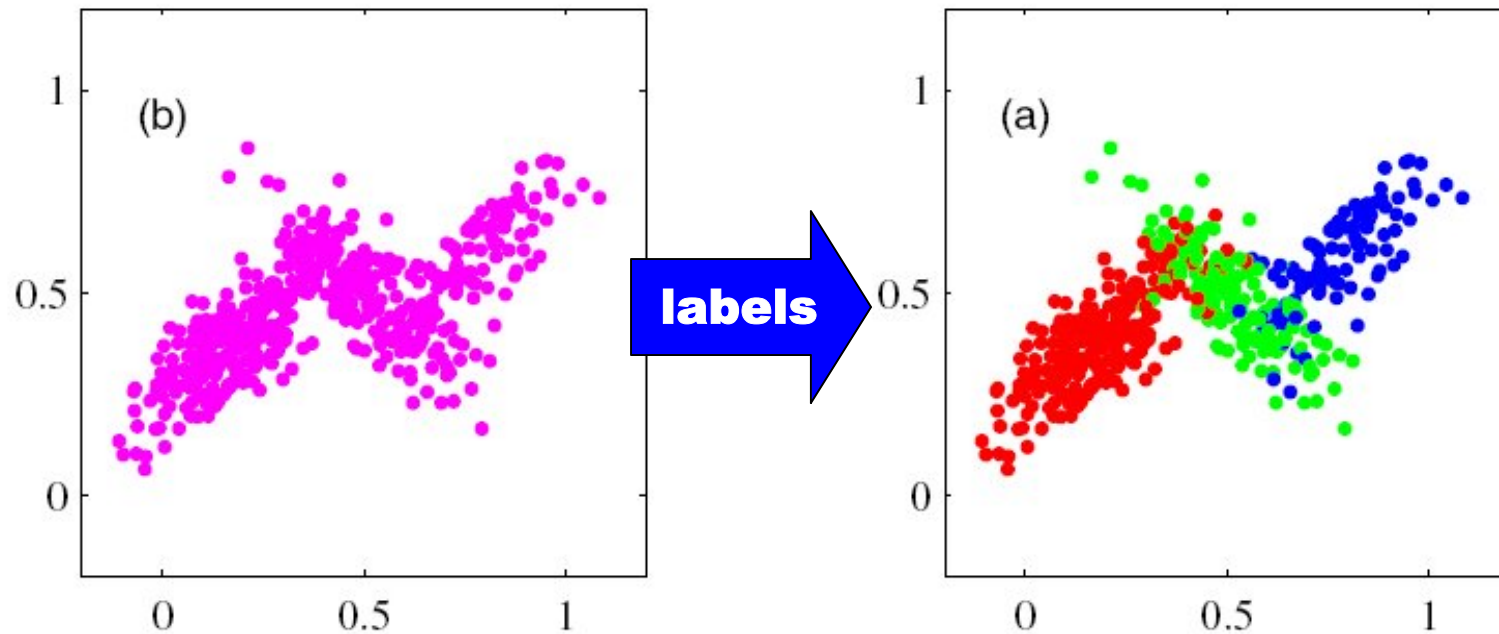
compute number of samples $n_k = \text{round}(N * \pi_k)$ to draw
from the k -th component Gaussian

generate n_k samples from Gaussian $N(\mu_k, \Sigma_k)$

end



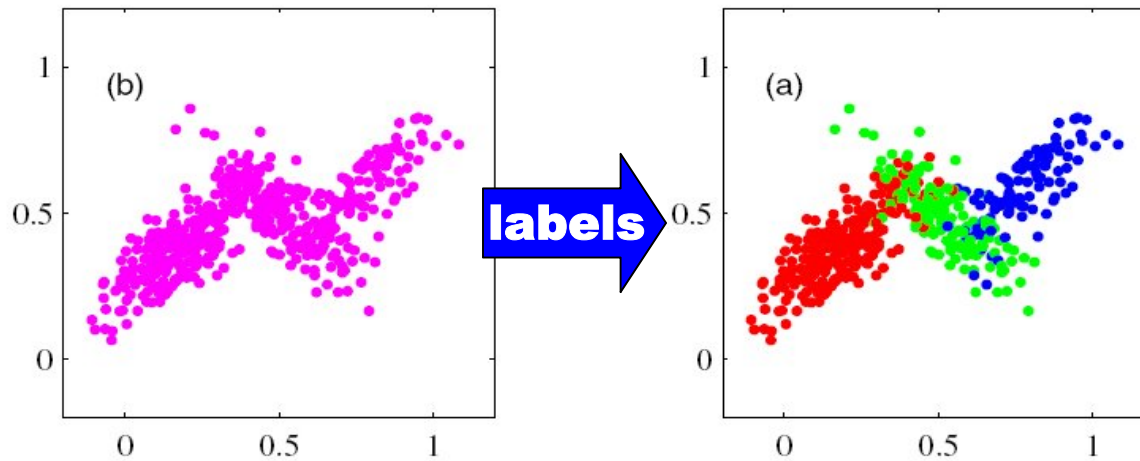
EM Algorithm



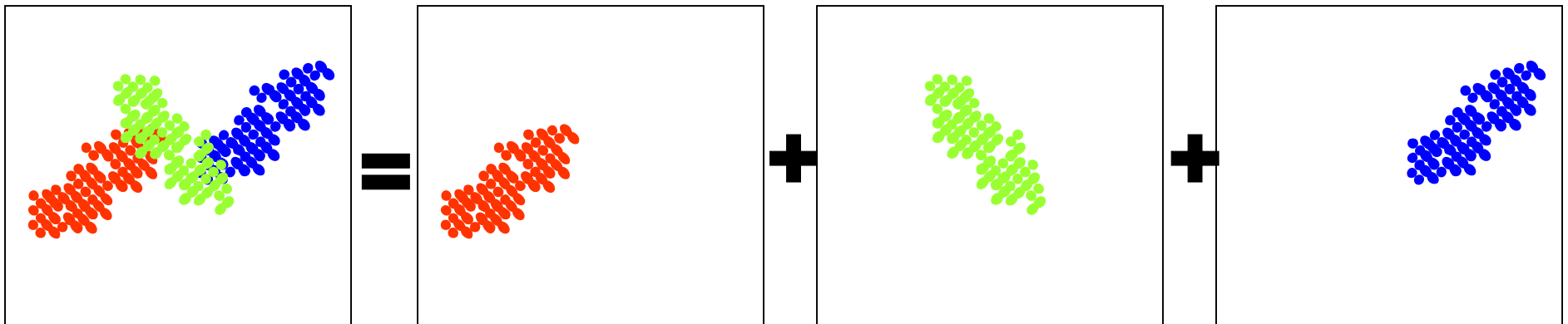
Suppose some oracle told us which point comes from which Gaussian.

How? By providing a “latent” variable z_{nk} which is 1 if point n comes from the k th component Gaussian, and 0 otherwise (a 1 of K representation)

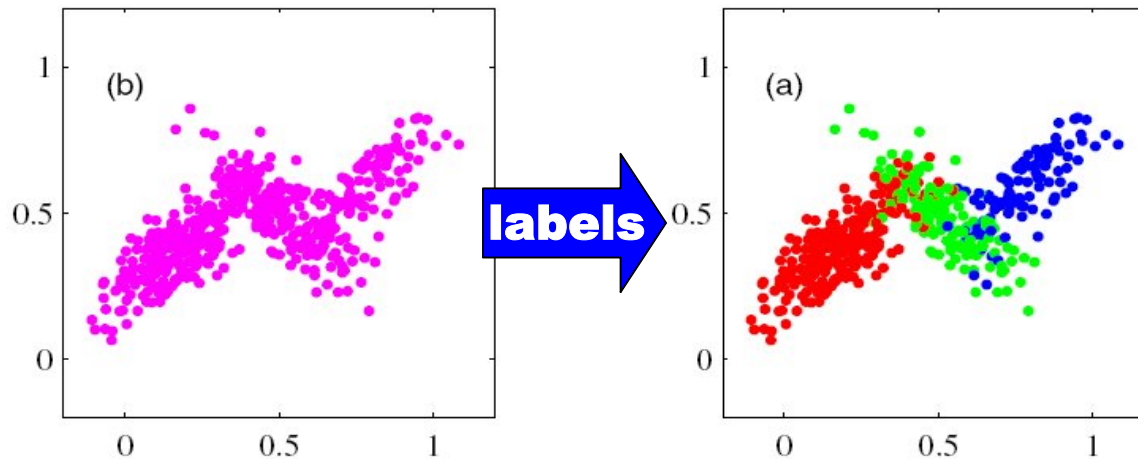
EM Algorithm



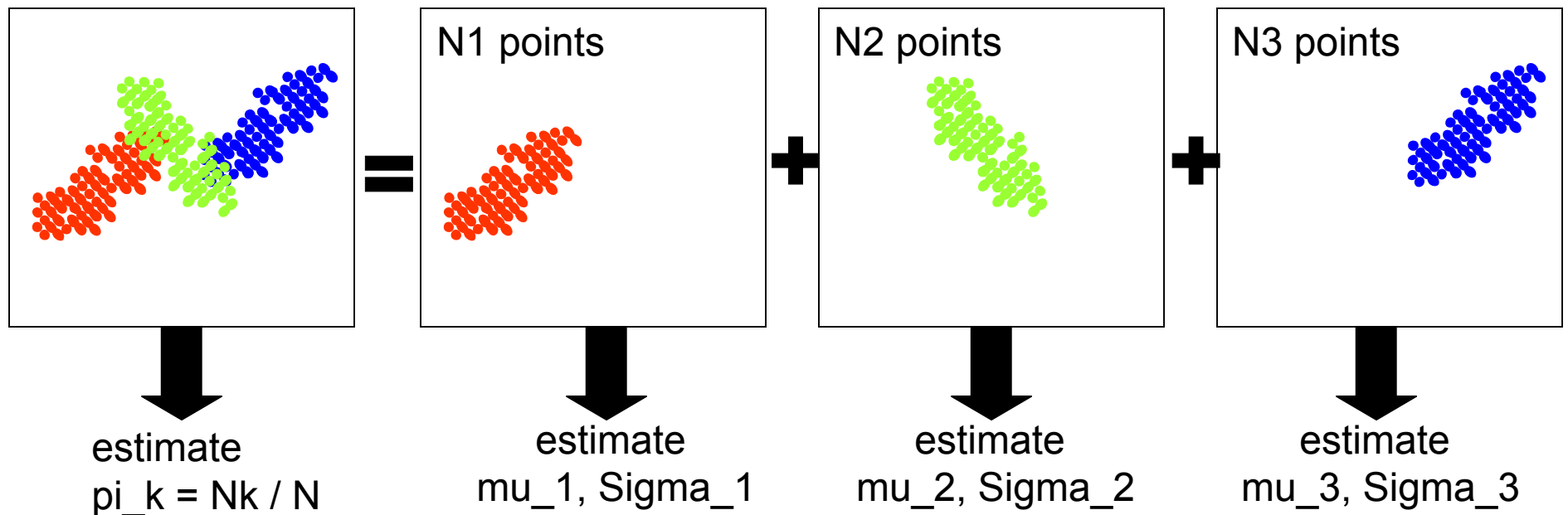
This lets us recover the underlying generating process decomposition:



EM Algorithm



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



EM Algorithm

Remember that this was a problem...

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

**how can I make that
inner sum be a product
instead???**



EM Algorithm

Remember that this was a problem...

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

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

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

and the log of that looks much better!

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

how can I make that inner sum be a product instead???



EM Algorithm

Remember that this was a problem...

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

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

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


and the log of that looks much better!

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

how can I make that inner sum be a product instead???



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

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

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

$$\begin{aligned} & \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,1}=1}} \ln \pi_1 + \ln \mathcal{N}(x_n | \mu_1, \Sigma_1) \\ + & \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,2}=1}} \ln \pi_2 + \ln \mathcal{N}(x_n | \mu_2, \Sigma_2) \quad + \quad \dots \quad + \\ + & \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,K}=1}} \ln \pi_K + \ln \mathcal{N}(x_n | \mu_K, \Sigma_K) \end{aligned}$$

Latent Variable View

$$\begin{aligned} & \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,1}=1}} \ln \pi_1 + \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,1}=1}} \ln \mathcal{N}(x_n | \mu_1, \Sigma_1) \\ & + \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,2}=1}} \ln \pi_2 + \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,2}=1}} \ln \mathcal{N}(x_n | \mu_2, \Sigma_2) \\ & + \dots + \\ & + \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,K}=1}} \ln \pi_K + \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,K}=1}} \ln \mathcal{N}(x_n | \mu_K, \Sigma_K) \end{aligned}$$

Latent Variable View

$$\begin{aligned} & \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,1}=1}} \ln \pi_1 + \boxed{\sum_{\substack{\text{all } n \text{ for which} \\ z_{n,1}=1}} \ln \mathcal{N}(x_n | \mu_1, \Sigma_1)} \quad \text{can be estimated separately} \\ & + \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,2}=1}} \ln \pi_2 + \boxed{\sum_{\substack{\text{all } n \text{ for which} \\ z_{n,2}=1}} \ln \mathcal{N}(x_n | \mu_2, \Sigma_2)} \quad \text{can be estimated separately} \\ & + \dots + \\ & + \sum_{\substack{\text{all } n \text{ for which} \\ z_{n,K}=1}} \ln \pi_K + \boxed{\sum_{\substack{\text{all } n \text{ for which} \\ z_{n,K}=1}} \ln \mathcal{N}(x_n | \mu_K, \Sigma_K)} \quad \text{can be estimated separately} \end{aligned}$$

Latent Variable View

$$\begin{aligned}
 & \sum_{\text{all } n \text{ for which } z_{n,1}=1} \ln \pi_1 + \sum_{\text{all } n \text{ for which } z_{n,1}=1} \ln \mathcal{N}(x_n | \mu_1, \Sigma_1) && \text{can be estimated separately} \\
 + & \sum_{\text{all } n \text{ for which } z_{n,2}=1} \ln \pi_2 + \sum_{\text{all } n \text{ for which } z_{n,2}=1} \ln \mathcal{N}(x_n | \mu_2, \Sigma_2) && \text{can be estimated separately} \\
 + & \dots + \\
 + & \sum_{\text{all } n \text{ for which } z_{n,K}=1} \ln \pi_K + \sum_{\text{all } n \text{ for which } z_{n,K}=1} \ln \mathcal{N}(x_n | \mu_K, \Sigma_K) && \text{can be estimated separately}
 \end{aligned}$$

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

EM Algorithm

Unfortunately, oracle's don't exist (or if they do, they don't want to talk to us)

So we don't know values of 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 θ
- 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 θ_{new} that maximizes that function.
This is our new best guess at the values of θ .
- 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 distribution. In the GMM case, this is equivalent to “softening” the binary latent variables to continuous ones (the expected values of the latent variables)

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

unknown discrete value 0 or 1

$$\mathbb{E}_{\mathbf{z}}[\ln p(\mathbf{x}, \mathbf{z} | \boldsymbol{\theta})] = \sum_{n=1}^N \sum_{i=1}^K \underbrace{\gamma_i(\mathbf{x}_n)}_{\text{known continuous value between 0 and 1}} \{ \ln \pi_i + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \}$$

known continuous value between 0 and 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 oracle)

$$\begin{aligned} & \sum_{\text{all } n \text{ for which } z_{n,1}=1} \ln \pi_1 + \sum_{\text{all } n \text{ for which } z_{n,1}=1} \ln \mathcal{N}(x_n | \mu_1, \Sigma_1) && \text{can be estimated separately} \\ + & \sum_{\text{all } n \text{ for which } z_{n,2}=1} \ln \pi_2 + \sum_{\text{all } n \text{ for which } z_{n,2}=1} \ln \mathcal{N}(x_n | \mu_2, \Sigma_2) && \text{can be estimated separately} \\ + & \dots + \\ + & \sum_{\text{all } n \text{ for which } z_{n,K}=1} \ln \pi_K + \sum_{\text{all } n \text{ for which } z_{n,K}=1} \ln \mathcal{N}(x_n | \mu_K, \Sigma_K) && \text{can be estimated separately} \end{aligned}$$

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

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

$$\begin{aligned}
 & \left(\sum_N \sum_K \gamma_{n,k}^i \ln \pi_1 + \sum_N \sum_K \gamma_{n,k}^i \ln \mathcal{N}(x_n | \mu_1, \Sigma_1) \right) \text{ can be estimated separately} \\
 + & \left(\sum_N \sum_K \gamma_{n,k}^i \ln \pi_2 + \sum_N \sum_K \gamma_{n,k}^i \ln \mathcal{N}(x_n | \mu_2, \Sigma_2) \right) \text{ can be estimated separately} \\
 + & \dots + \\
 + & \left(\sum_N \sum_K \gamma_{n,k}^i \ln \pi_K + \sum_N \sum_K \gamma_{n,k}^i \ln \mathcal{N}(x_n | \mu_K, \Sigma_K) \right) \text{ can be estimated separately}
 \end{aligned}$$

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

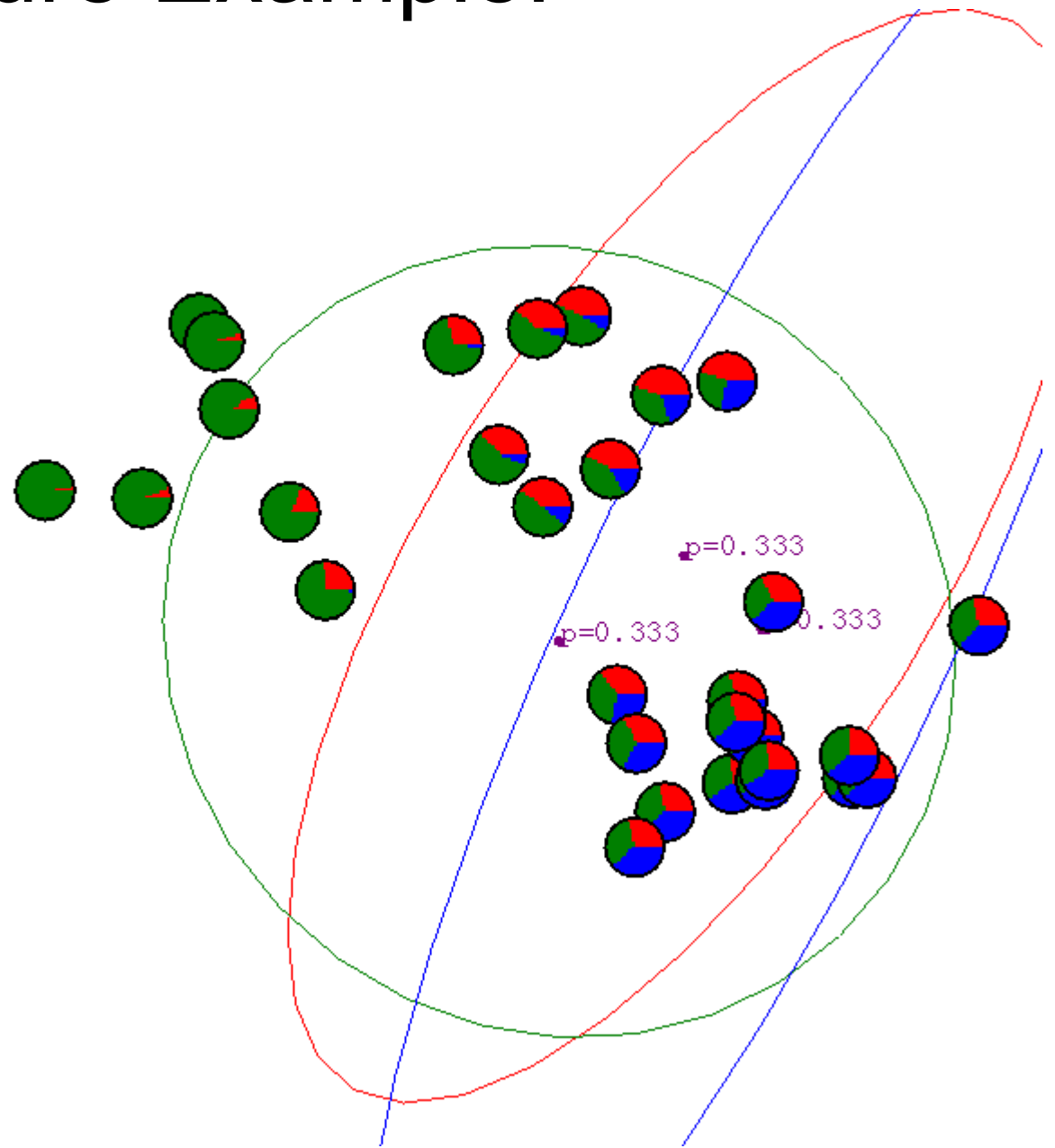
EM Algorithm for GMM

$$\mathbf{E} \quad \frac{\pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\underbrace{\sum_k \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\gamma_j(\mathbf{x}_n)}} \quad \text{ownership weights}$$

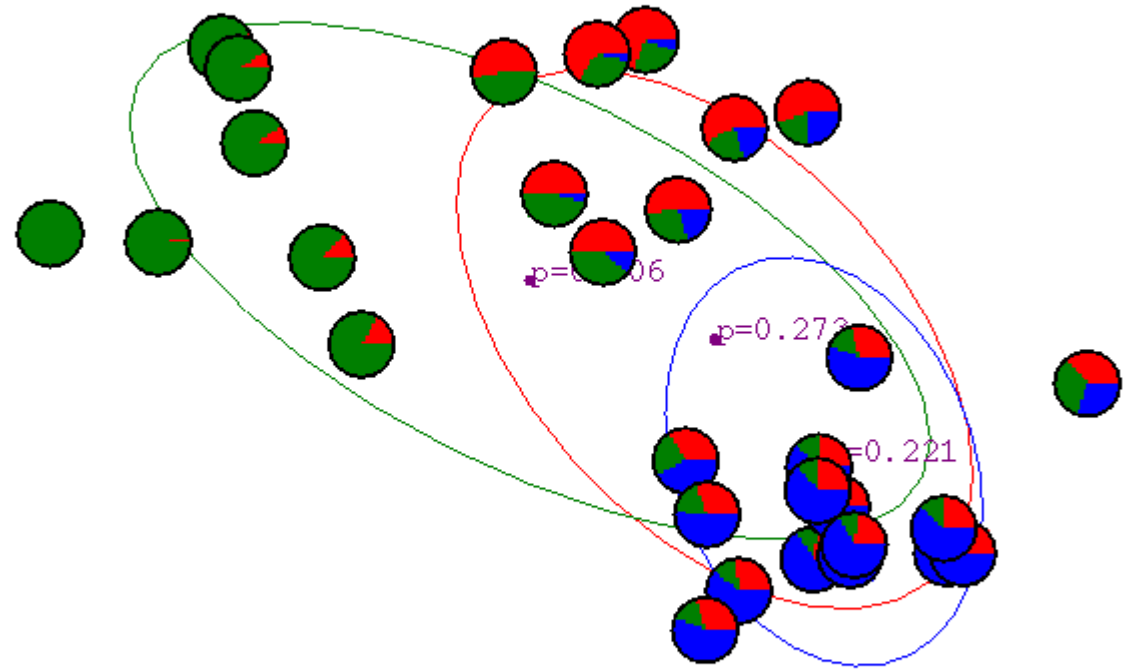
$$\mathbf{M} \quad \mu_j = \frac{\sum_{n=1}^N \gamma_j(\mathbf{x}_n) \mathbf{x}_n}{\sum_{n=1}^N \gamma_j(\mathbf{x}_n)} \quad \text{means} \quad \Sigma_j = \frac{\sum_{n=1}^N \gamma_j(\mathbf{x}_n) (\mathbf{x}_n - \boldsymbol{\mu}_j) (\mathbf{x}_n - \boldsymbol{\mu}_j)^\top}{\sum_{n=1}^N \gamma_j(\mathbf{x}_n)} \quad \text{covariances}$$

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

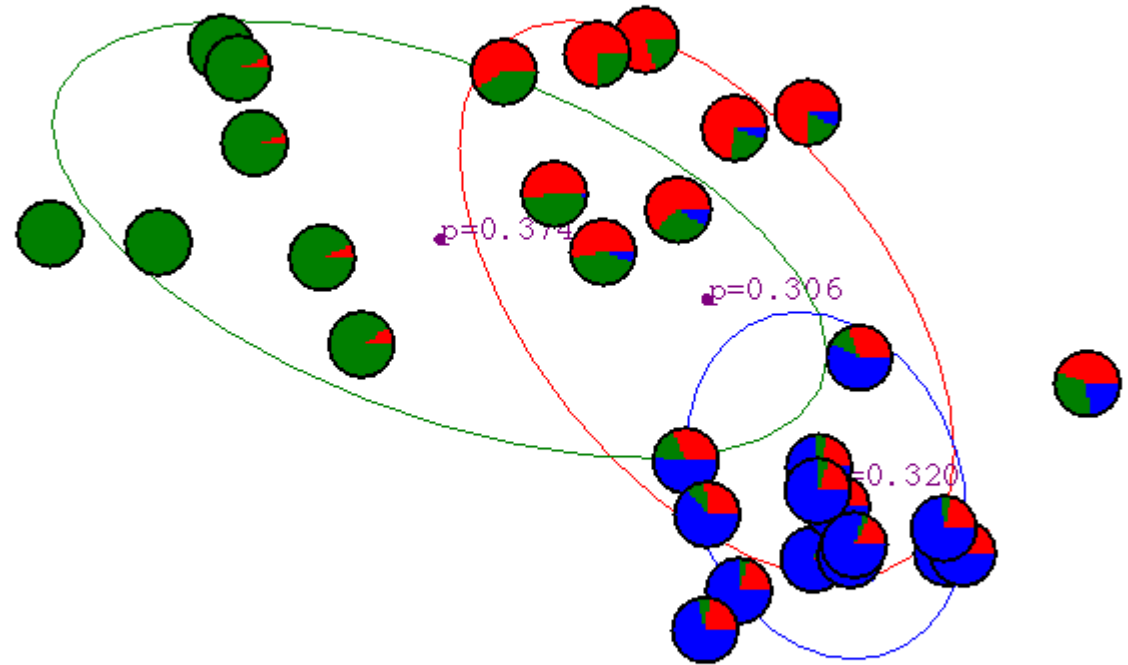
Gaussian Mixture Example: Start



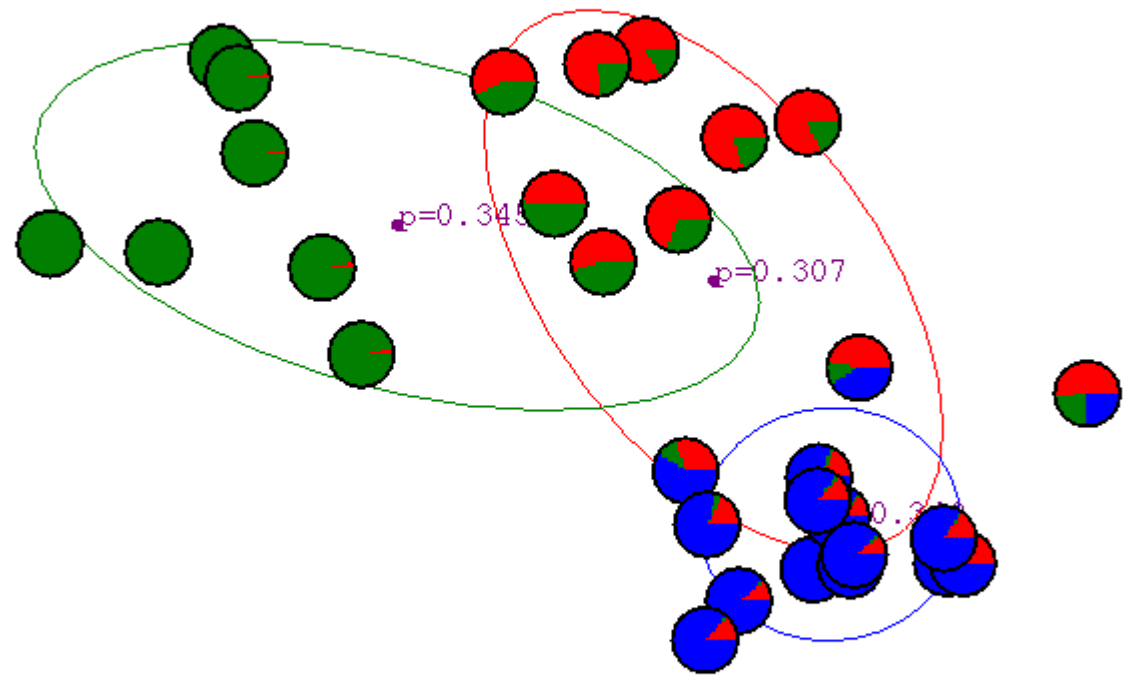
After first iteration



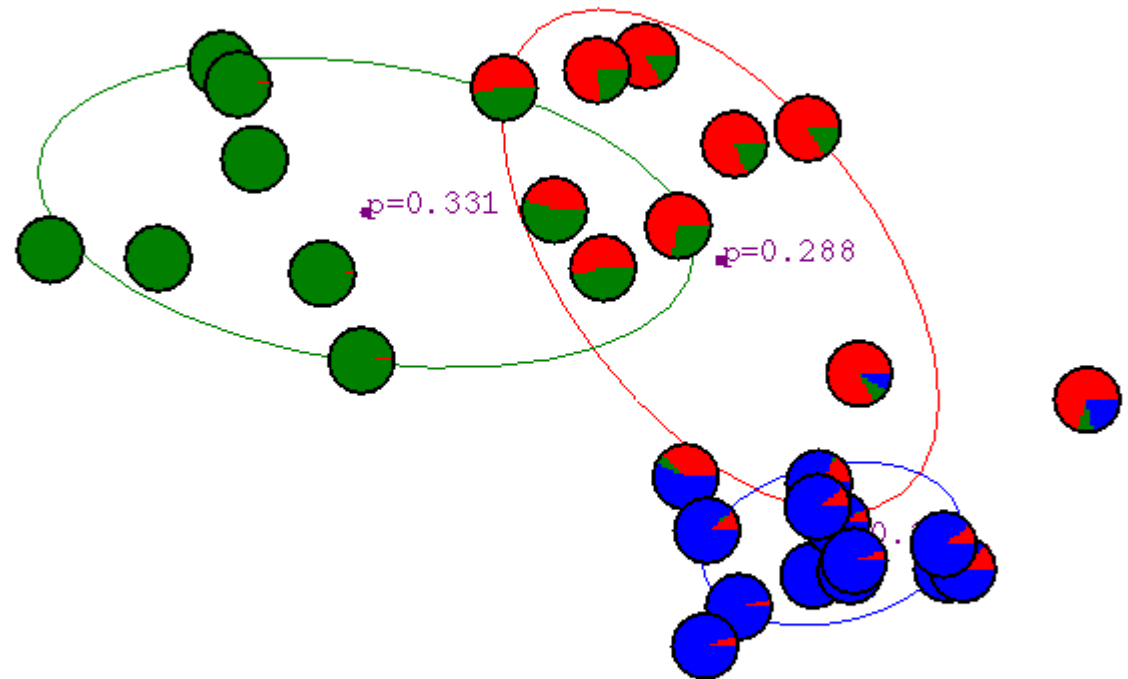
After 2nd iteration



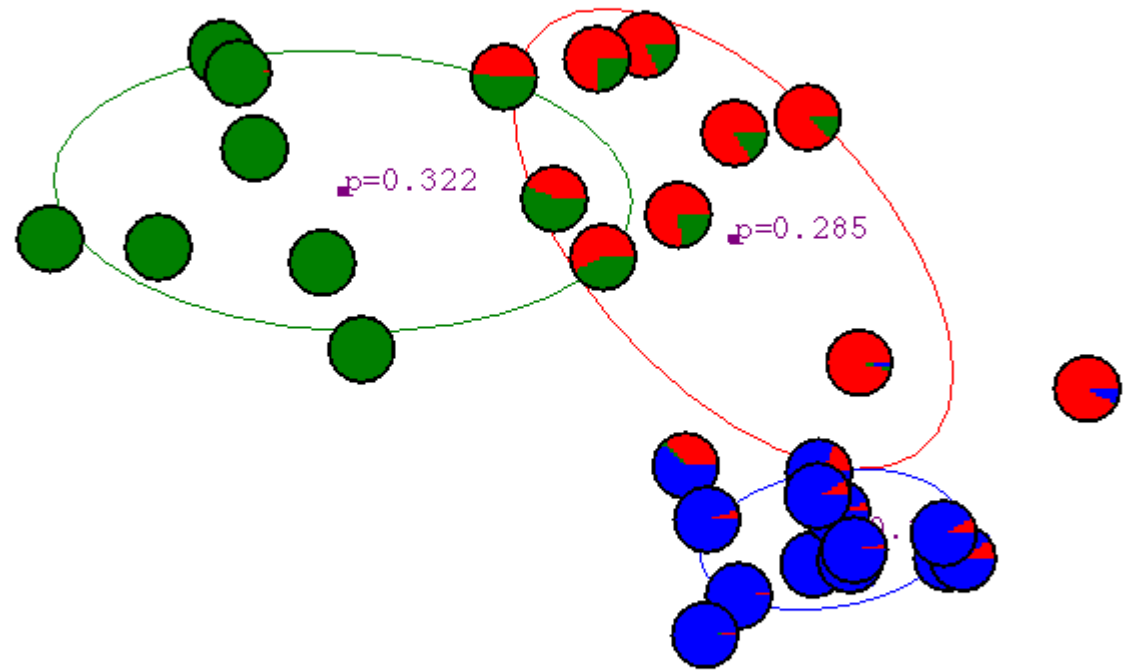
After 3rd iteration



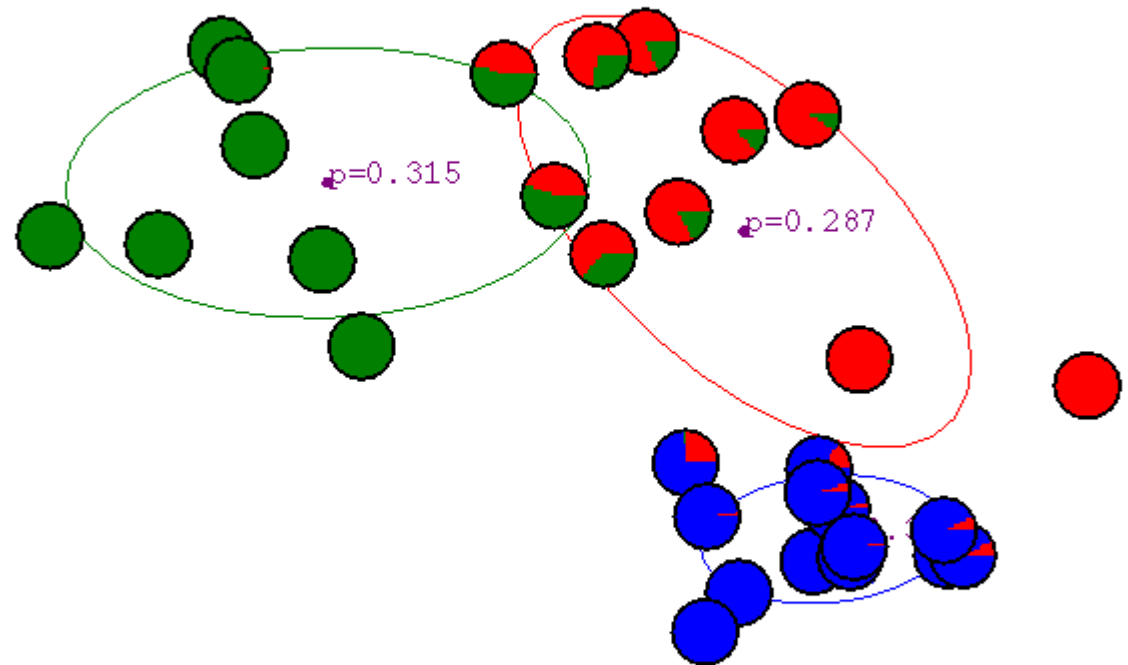
After 4th iteration



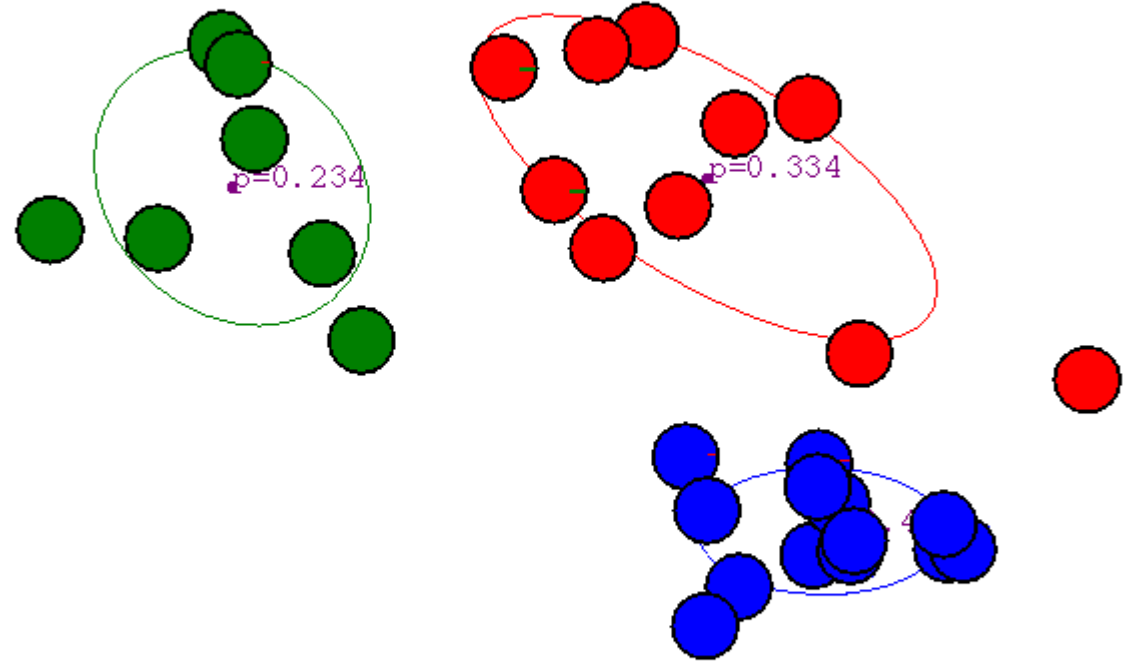
After 5th iteration



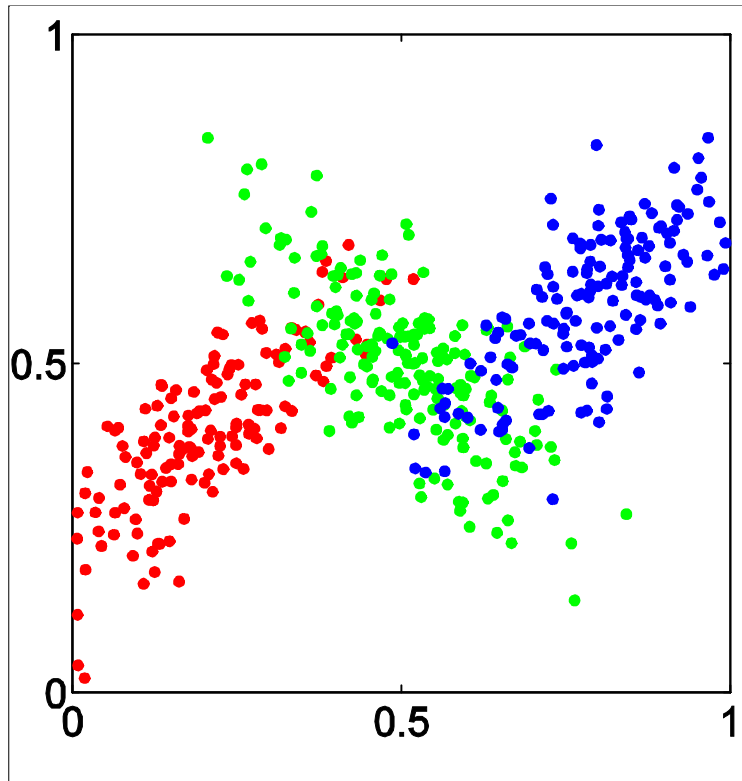
After 6th iteration



After 20th iteration

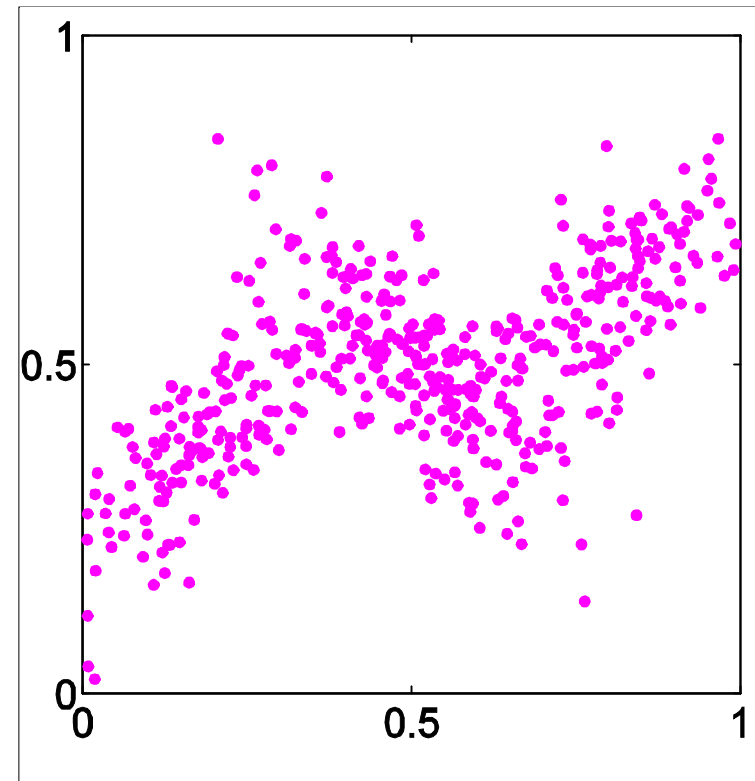


Recall: Labeled vs Unlabeled Data



labeled

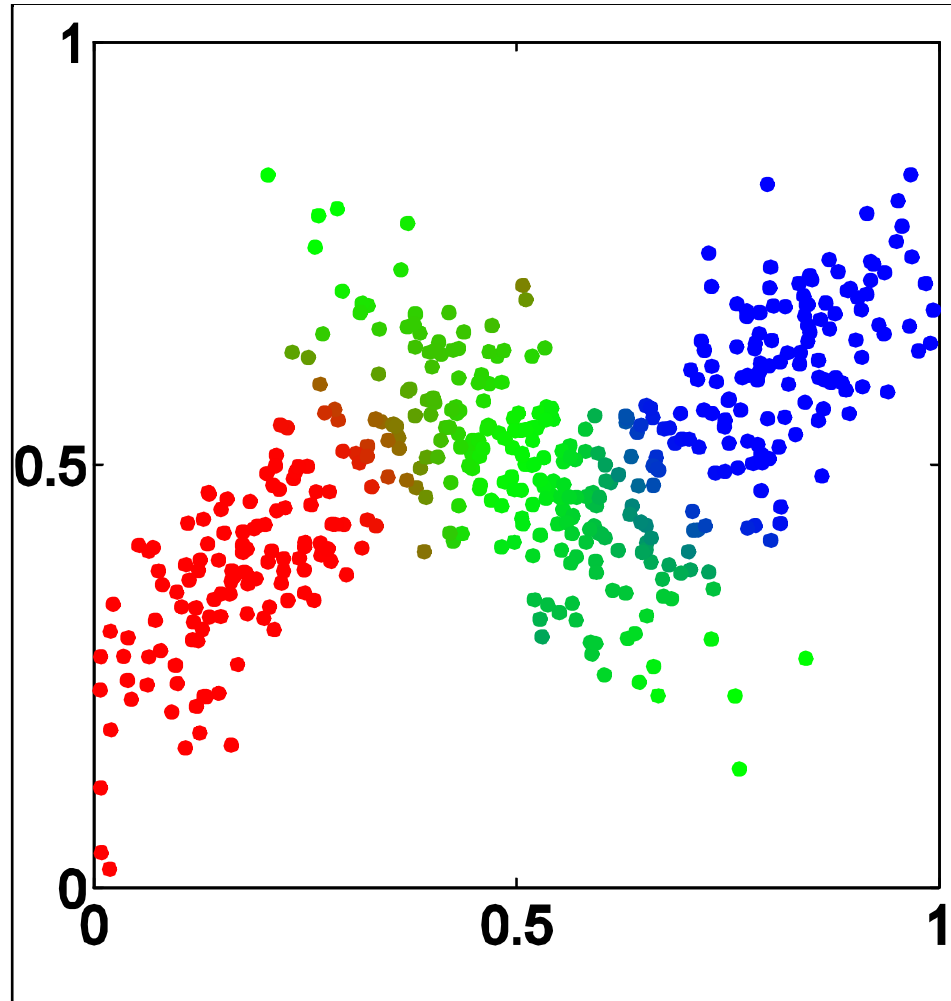
Easy to estimate params
(do each color separately)



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

General EM

The General EM Algorithm

Given a joint distribution $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ over observed variables \mathbf{X} and latent variables \mathbf{Z} , governed by parameters $\boldsymbol{\theta}$, the goal is to maximize the likelihood function $p(\mathbf{X}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

1. Choose an initial setting for the parameters $\boldsymbol{\theta}^{\text{old}}$.
2. **E step** Evaluate $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$.

Evaluate

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}). \quad (9.33)$$

3. **M step** Evaluate $\boldsymbol{\theta}^{\text{new}}$ given by

$$\boldsymbol{\theta}^{\text{new}} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) \quad (9.32)$$

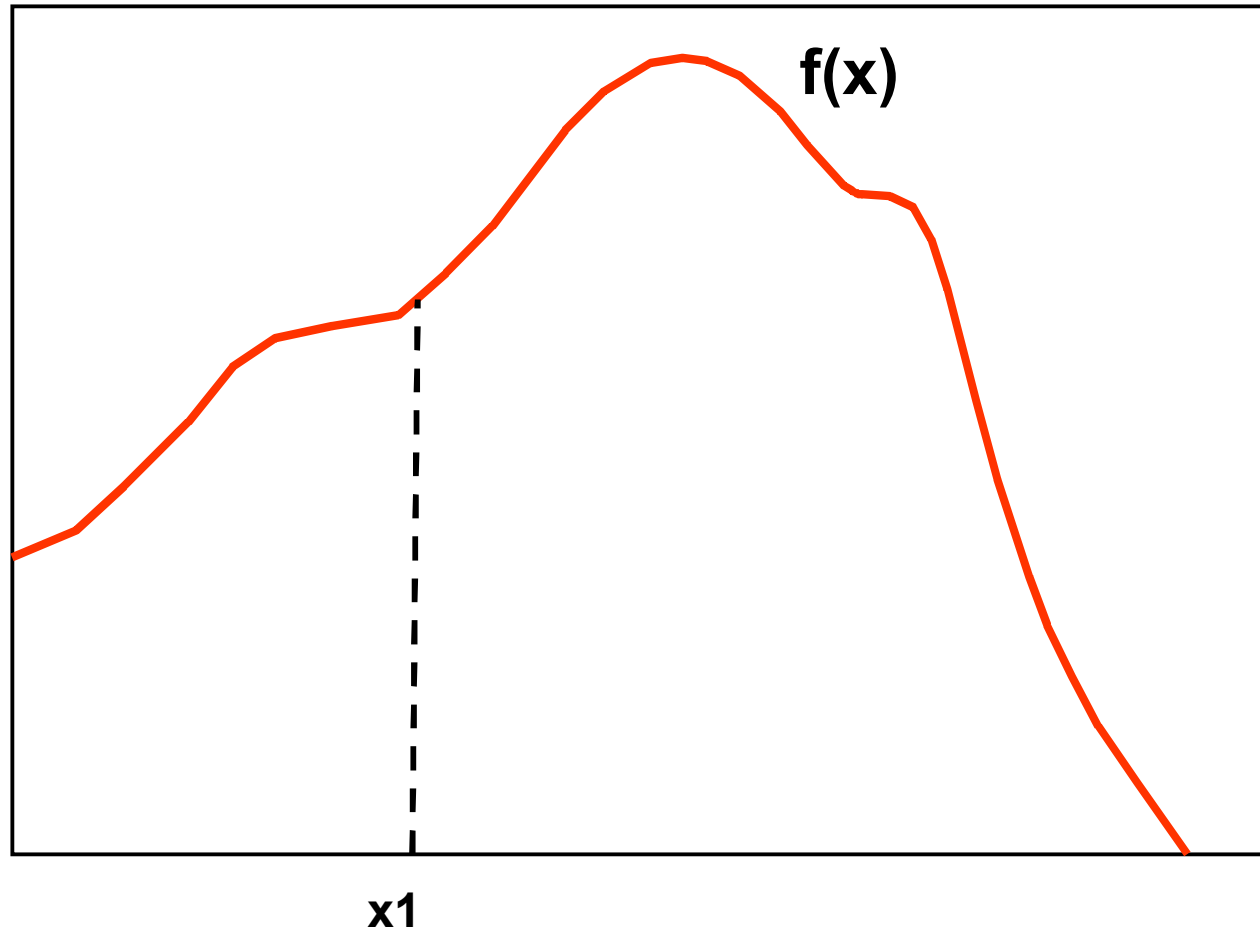
4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

$$\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}} \quad (9.34)$$

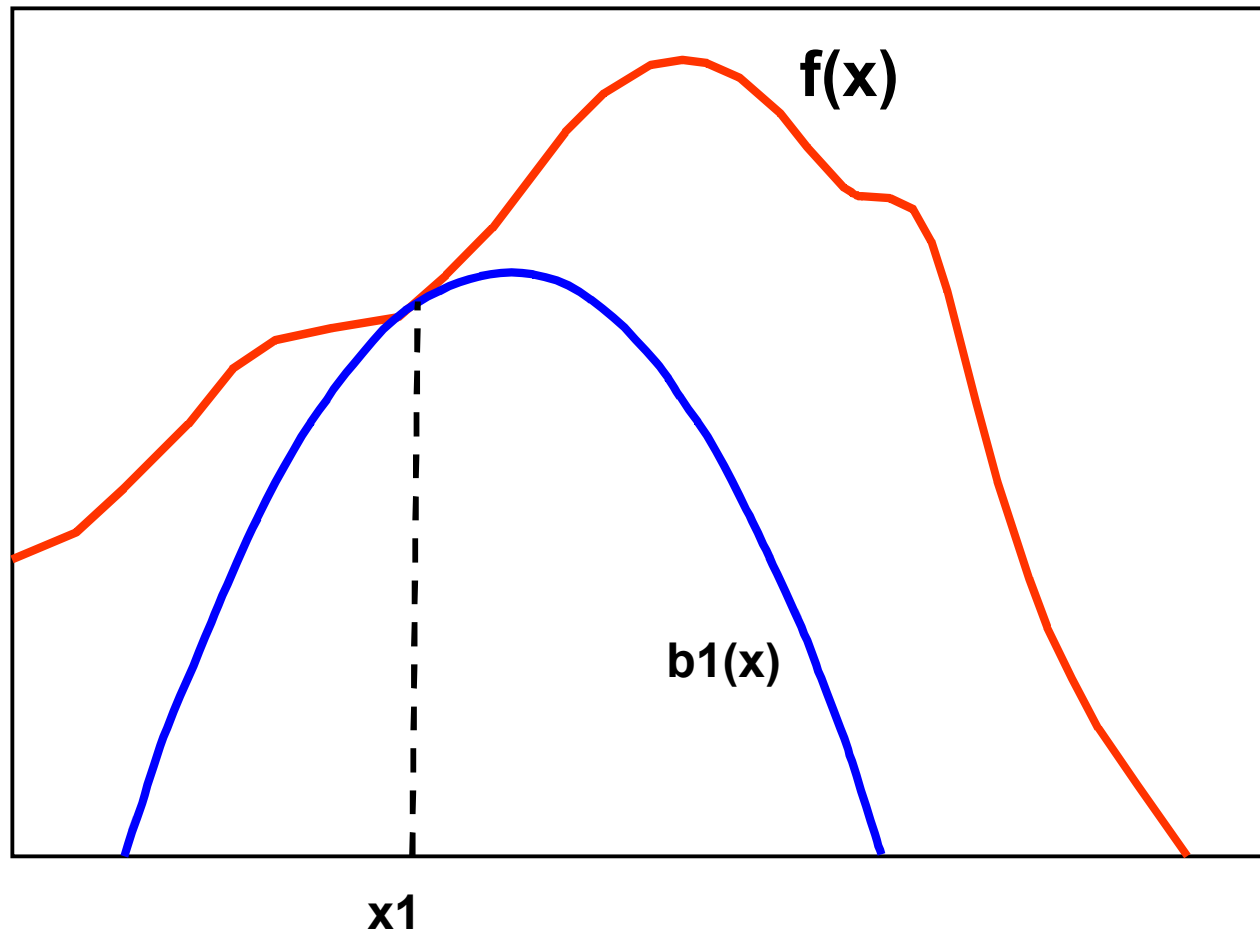
and return to step 2.

Intuitive Explanation

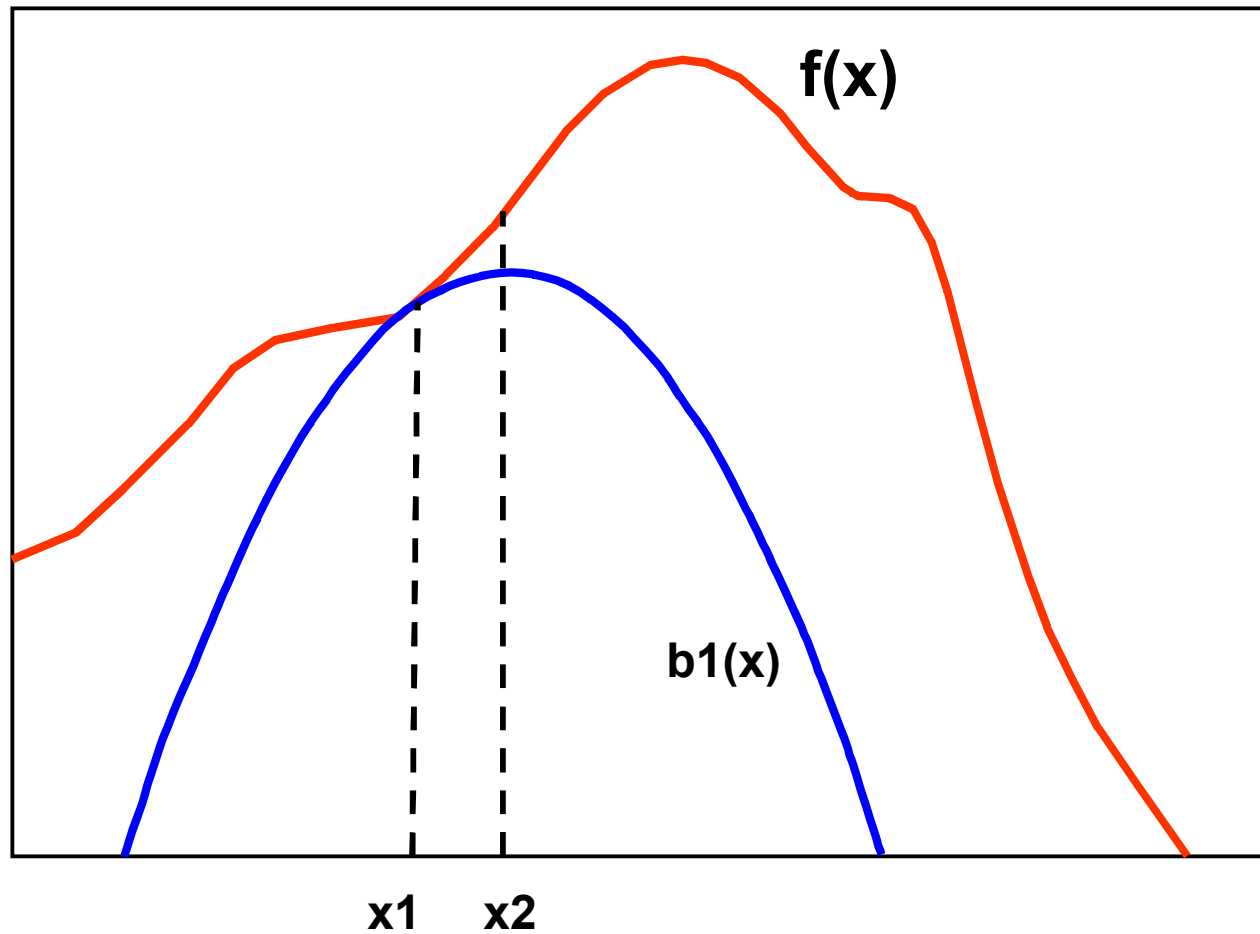
(in terms of function maximization and lower bounds)



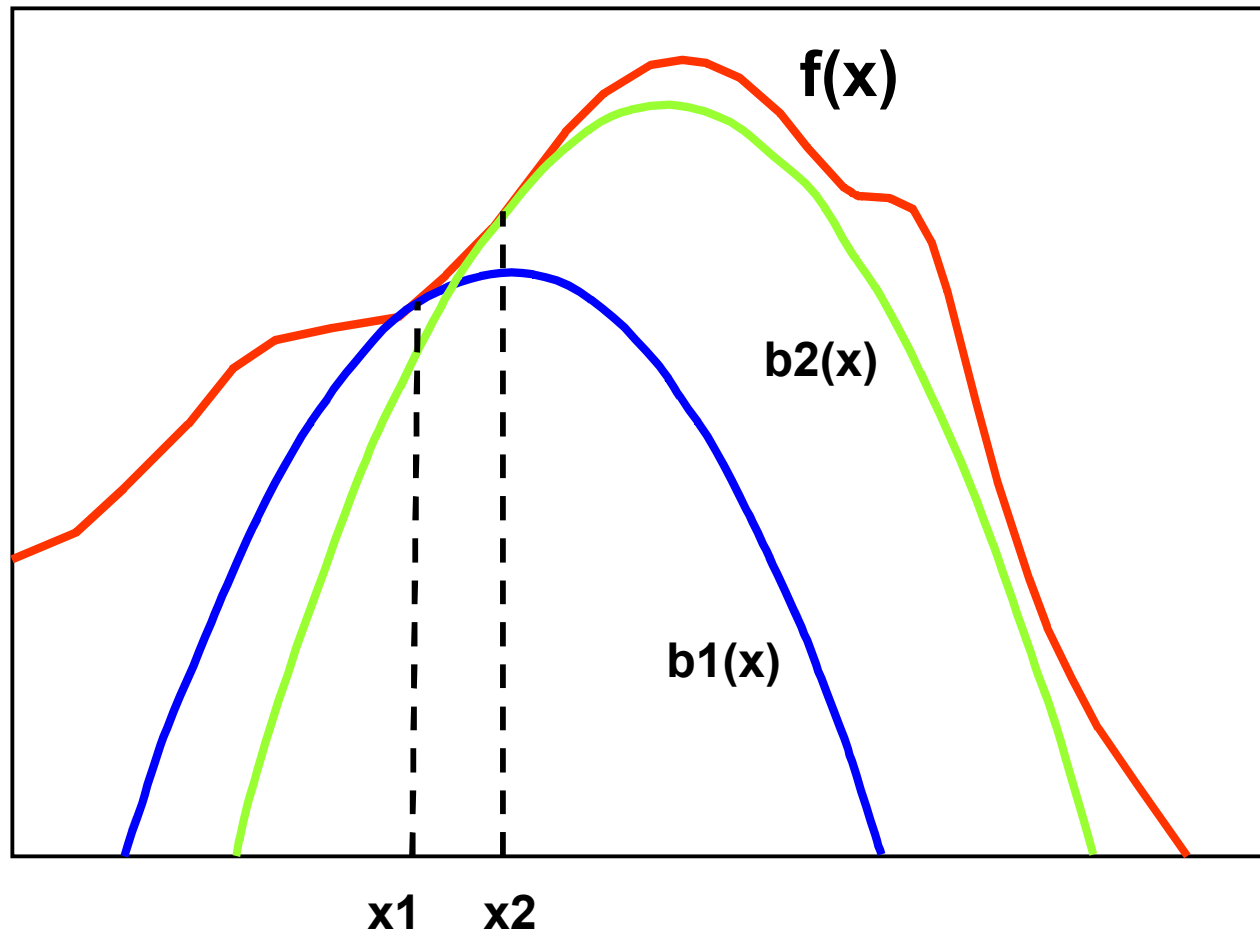
Intuitive Explanation



Intuitive Explanation



Intuitive Explanation



Intuitive Explanation

Why does this work?

By construction, $b_1(x_1) = f(x_1)$

$b_1(x_2) \geq b_1(x_1)$ [it is a maximum]

$f(x_2) \geq b_1(x_2)$ [b_1 is a lower bound]

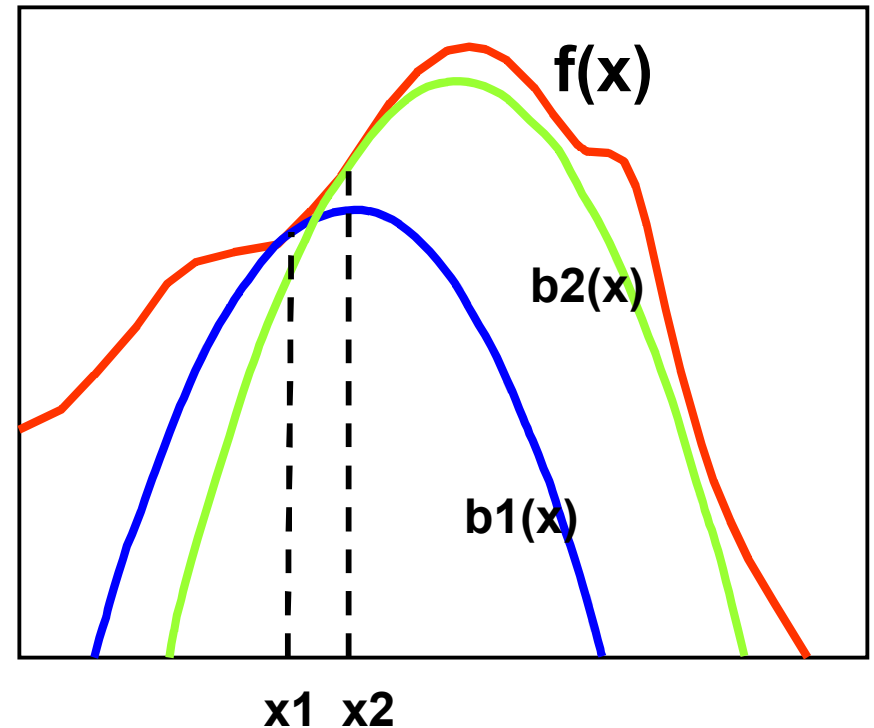
so, it is guaranteed that

$$f(x_2) \geq f(x_1)$$

and in general, at each iteration

$$f(x_{\text{new}}) \geq f(x_{\text{old}})$$

If $f(x)$ is bounded above, then process should converge to a (local) maximum



More Rigorous Proof

We will use Jensen's inequality for convex functions (see, for example, Bishop, PRML, p 56)

$$f\left(\sum_{i=1}^M \lambda_i x_i\right) \leq \sum_{i=1}^M \lambda_i f(x_i)$$

where $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$, for any set of points $\{x_i\}$.

With some manipulation, and reversing the inequality because log is a concave rather than convex function...

$$\ln \sum_k a_k = \ln \sum_k \lambda_k \frac{a_k}{\lambda_k} \geq \sum_k \lambda_k \ln \left(\frac{a_k}{\lambda_k} \right)$$

Proof that EM works

$\ln p(X|\theta)$: this is function $f(x)$ in our earlier picture

$= \ln \sum_Z P(X, Z|\theta)$ definition of probability. Now use Jensen's inequality...

$$\geq \underbrace{\sum_Z p(Z|X, \theta^{\text{old}}) \ln \frac{p(X, Z|\theta)}{p(Z|X, \theta^{\text{old}})}}_{\text{lower bound b(x) in our earlier picture. Bishop calls this } L(q, \theta)}$$

lower bound $b(x)$ in our earlier picture.
Bishop calls this $L(q, \theta)$

Proof that EM works

note, that when $\theta = \theta_{\text{old}}$

$$\sum_Z p(Z|X, \theta^{\text{old}}) \ln \frac{p(X, Z|\theta^{\text{old}})}{p(Z|X, \theta^{\text{old}})}$$

if we expand

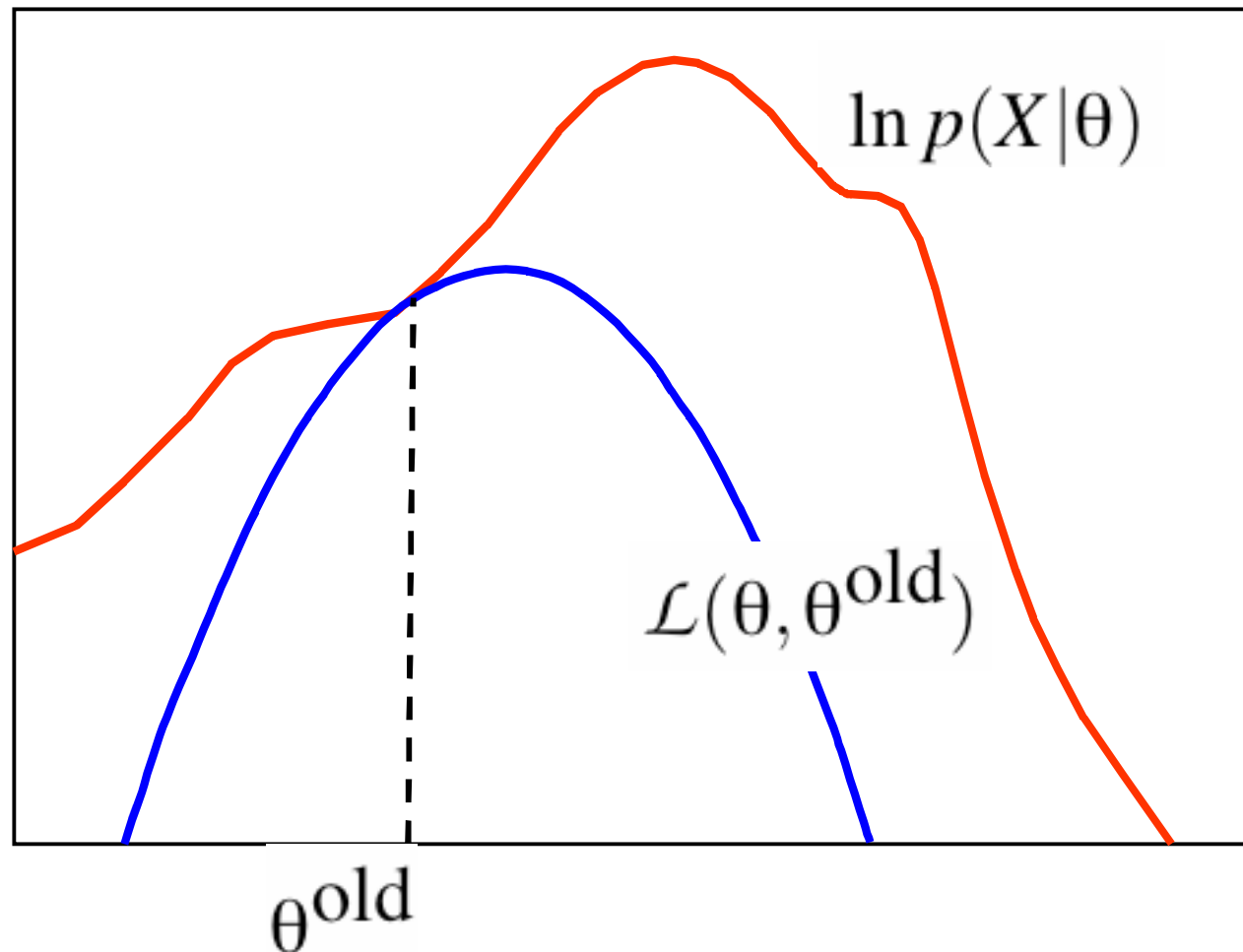
$$p(X, Z|\theta^{\text{old}}) = p(Z|X, \theta^{\text{old}}) p(X|\theta^{\text{old}})$$

this equation $L(q, \theta_{\text{old}})$ becomes just

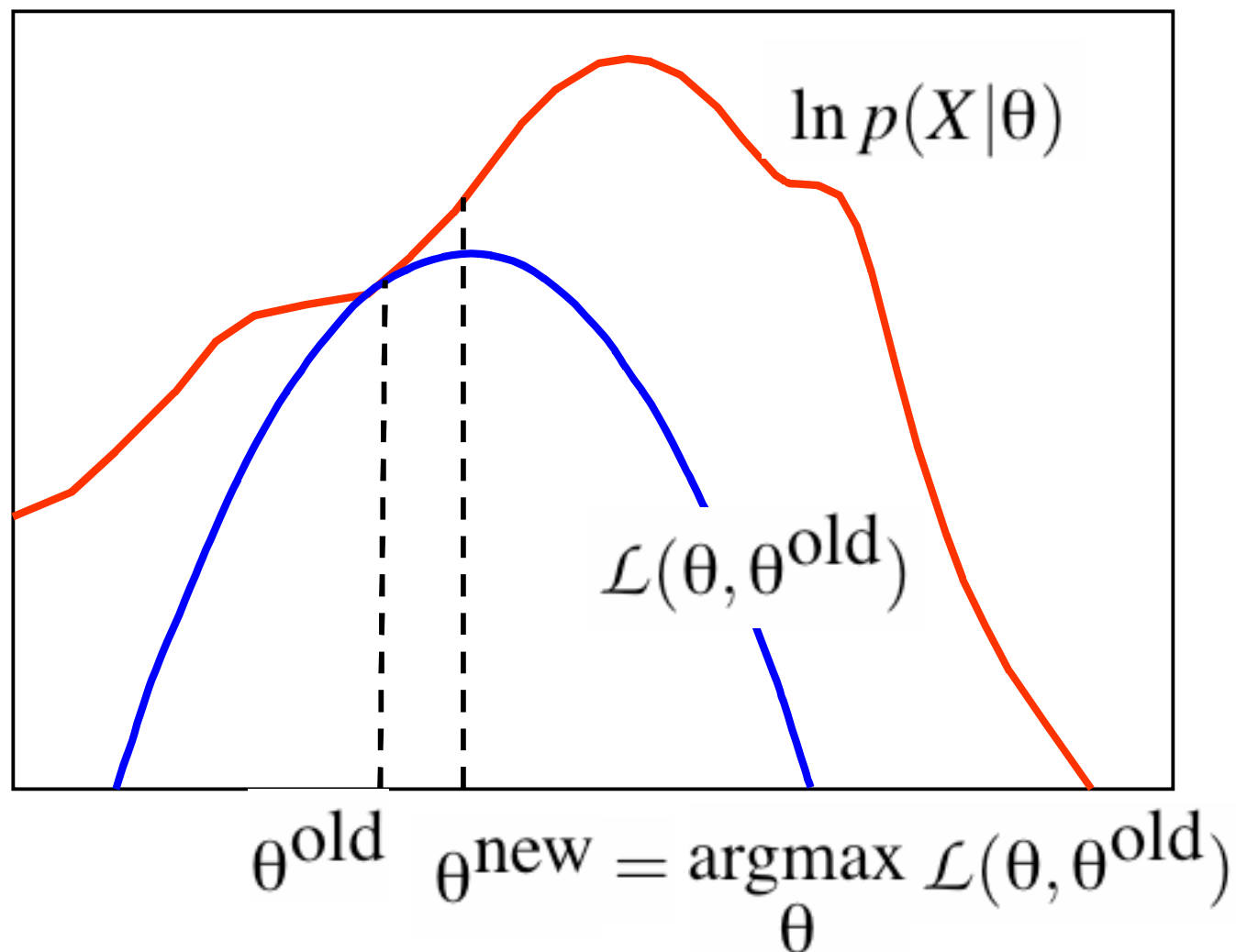
$$\ln p(X|\theta^{\text{old}})$$

Showing that our lower bound “touches” the function $\ln p(x|\theta)$ at the current estimate θ_{old} (as promised by our earlier picture!)

Proof that EM works



Proof that EM works



Proof that EM works

So...

$$\ln p(X|\theta^{\text{new}}) \geq \ln p(X|\theta^{\text{old}})$$

and we have increased our log likelihood.

Therefore, finding argmax of $L(\theta, \theta_{\text{old}})$ is a good thing to do.

But wait, there's more...

Proof that EM works

$$\mathcal{L}(\theta, \theta^{\text{old}}) = \sum_Z p(Z|X, \theta^{\text{old}}) \ln \frac{p(X, Z|\theta)}{p(Z|X, \theta^{\text{old}})}$$

$$= \underbrace{\sum_Z p(Z|X, \theta^{\text{old}}) \ln p(X, Z|\theta)}_{\text{This is the expected value } Q(\theta, \theta^{\text{old}}) \text{ computed in the E-step of EM!!!!}} - \underbrace{\sum_Z p(Z|X, \theta^{\text{old}}) \ln p(Z|X, \theta^{\text{old}})}_{\text{this doesn't depend on } \theta. \text{ ignore it.}}$$

Proof that EM works

$$\mathcal{L}(\theta, \theta^{\text{old}}) = \sum_Z p(Z|X, \theta^{\text{old}}) \ln \frac{p(X, Z|\theta)}{p(Z|X, \theta^{\text{old}})}$$

$$= \underbrace{\sum_Z p(Z|X, \theta^{\text{old}}) \ln p(X, Z|\theta)}_{\text{This is the expected value } Q(\theta, \theta^{\text{old}}) \text{ computed in the E-step of EM!!!!}} - \underbrace{\sum_Z p(Z|X, \theta^{\text{old}}) \ln p(Z|X, \theta^{\text{old}})}_{\text{this doesn't depend on } \theta. \text{ ignore it.}}$$

$$\operatorname{argmax}_{\theta} \mathcal{L}(\theta, \theta^{\text{old}}) \equiv \operatorname{argmax}_{\theta} Q(\theta, \theta^{\text{old}}) \quad \text{therefore, EM is optimizing the right thing!}$$