Mixture Model (of Gaussians) and Expectation Maximization (EM)

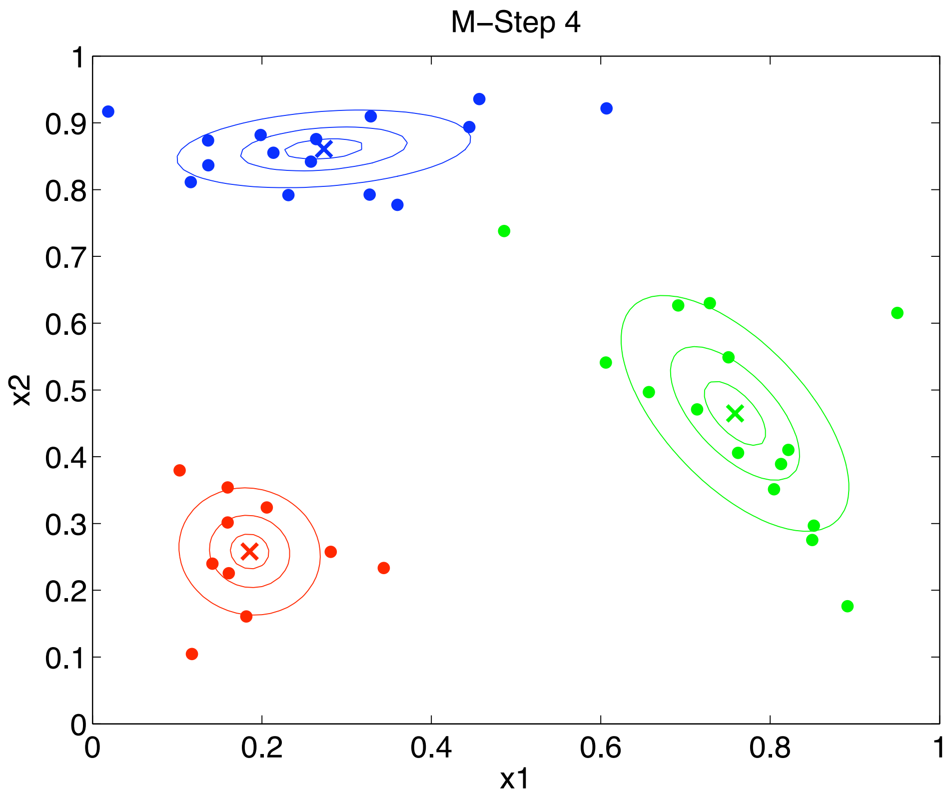
- Semi-supervised learning and clustering as a missing data problem
- Gaussian Mixture Model (GMM)
- Expectation Maximization (EM)
- EM for Gaussian mixture models

Problem formulation

- Suppose you have a classification data set, with data coming from K classes
- But someone erased all or part of the class labels
- You would like to know to what class each example belongs

- In **semi-supervised** learning, some y's are observed some not. Even so, using the $\mathbf{x}$'s with unobserved y's can be helpful.
- In the **clustering problem**, y is *never* observed, so we only have the second term above.

Illustration of the objective



More generally: Missing values / semi-supervised learning

- Suppose we have a generative model of supervised learning data with parameters θ.
- The likelihood of the data is given as $L(\theta) = P(\text{observations}|\theta)$.
  - The goal is to increase the likelihood, i.e. finding a good model.
  - For many applications, the natural logarithm of the likelihood function, called the log-likelihood, is more convenient to work with.

More generally: Missing values / semi-supervised learning

- Under the i.i.d. assumption, the log-likelihood of the data can be written as:

$$\log L(\theta) = \sum_{\text{complete data}} \log P(\mathbf{x}_i, y_i|\theta) + \sum_{\text{incomplete data}} \log P(\mathbf{x}_i|\theta)$$

- For the second term, we must consider *all possible values* for y:

$$\sum_{\text{incomplete data}} \log P(\mathbf{x}_i|\theta) = \sum_{\text{incomplete data}} \log \left(\sum_y P(\mathbf{x}_i, y|\theta) \right)$$

The parameters in a Gaussian mixture model

We will look at a model with one gaussian per class. The parameters of the model[1] are:

- The prior probabilities, $P(y = k)$.
- Mean and covariance matrix, μ_k, Σ_k, defining a multivariate Gaussian distribution for examples in class k.

[1] For D-dimensional data, we have for each Gaussian:

1. A Symmetric full DxD covariance matrix where (D*D - D)/2 is the number of off-diagonal elements and D is the number of diagonal elements
2. A D dimensional mean vector giving D parameters
3. A mixing weight giving another parameter

The overall number of parameters is (D*D - D)/2 + 2D + 1 for each gaussian.

Data likelihood with missing values

Complete data	Missing values
Log-likelihood has a unique maximum in the case of a mixture of gaussians model	There are many local maxima! Maximizing the likelihood becomes a non-linear optimization problem
Under certain assumptions, there is a nice, closed-form solution for the parameters	Closed-form solutions cannot be obtained

Two solutions

1. *Gradient ascent:* follow the gradient of the likelihood with respect to the parameters
2. *Expectation maximization:* use the current parameter setting to construct a local approximation of the likelihood which is "nice" and can be optimized easily

Gradient ascent

- Move parameters in the direction of the gradient of the log-likelihood
- Note: It is easy to compute the gradient at any parameter setting
- Pro: We already know how to do this!
- Cons:
  - We need to ensure that we get "legal" probability distributions or probability density functions (e.g., the gradient needs to be *projected on the space of legal parameters*)
  - Sensitive to parameters (e.g. learning rates) and possibly slow

Expectation Maximization (EM)

- A general purpose method for learning from incomplete data
- Main idea:
  - If we had complete data we could easily maximize the likelihood
  - But because the data is incomplete, we get a summation inside the $\log$, which makes the optimization much harder
  - So in the case of missing values, we will "fantasize" what they should be, based on the current parameter setting
  - In other words, we *fill in the missing values based on our current expectation*
  - Then we *compute new parameters, which maximize the likelihood* of the completed data

In summary, we estimate y given θ, then we reestimate θ given y, then we reestimate y given the new θ, . . .

Maximum likelihood solution

- Let $\delta_{ik} = 1$ if $y_i = k$ and 0 otherwise
- The class probabilities are determined by the empirical frequency of examples in each class:

$$P(y = k) = p_k = \frac{\sum_i \delta_{ik}}{\sum_k \sum_i \delta_{ik}}$$

- The mean and covariance matrix for class k are the empirical mean and covariance of the examples in that class:

$$\begin{aligned} \mu_k &= \frac{\sum_i \delta_{ik} \mathbf{x}_i}{\sum_i \delta_{ik}} \\ \Sigma_k &= \frac{\sum_i \delta_{ik} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{\sum_i \delta_{ik}} \end{aligned}$$

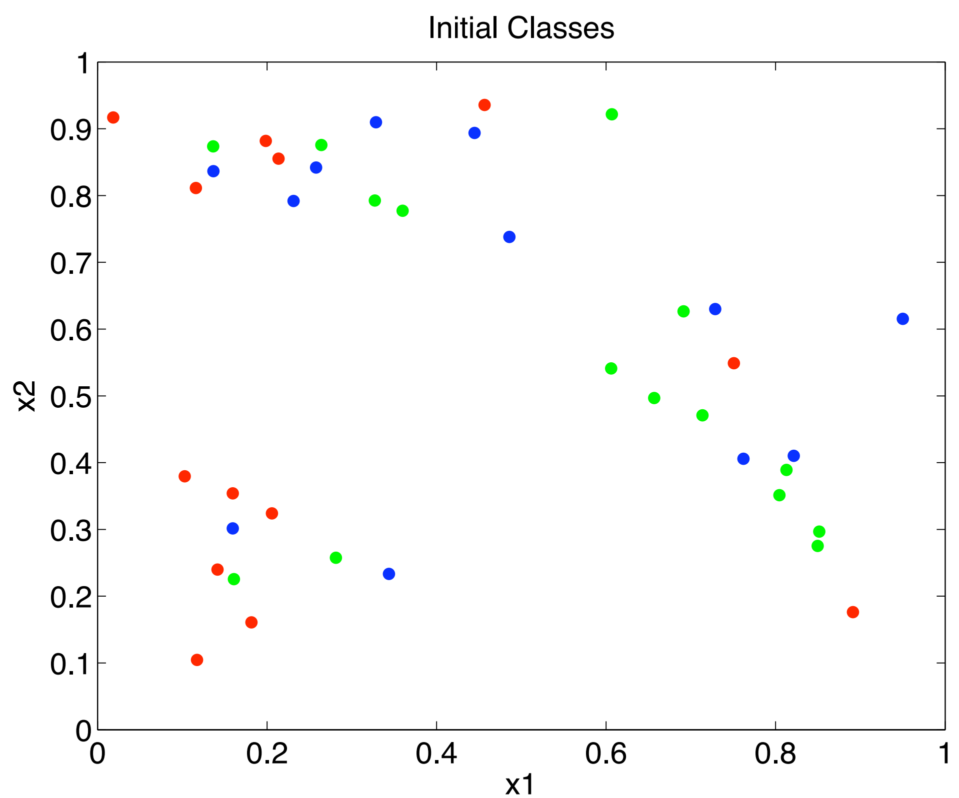
EM for Mixture of Gaussians

- We start with an initial guess for the parameters p_k, μ_k, Σ_k
- We will alternate an:
  - *expectation step (E-step)*, in which we "complete" the data—estimating the y_i
  - *maximization step (M-step)*, in which we re-compute the parameters P_k, μ_k, Σ_k
- In the *hard EM* version, completing the data means that each data point is assumed to be generated by *exactly one Gaussian*—taken to be the most likely assignment.
  (This is roughly equivalent to the setting of K-means clustering.)
- In the *soft EM* version (also usually known as EM), we assume that each data point could have been generated from *any component*
  - We estimate probabilities $P(y_i = k) = P(\delta_{ik} = 1) = E(\delta_{ik})$
  - Each $\mathbf{x}_i$ contributes to the mean and variance estimate of each component.

Hard EM for Mixture of Gaussians

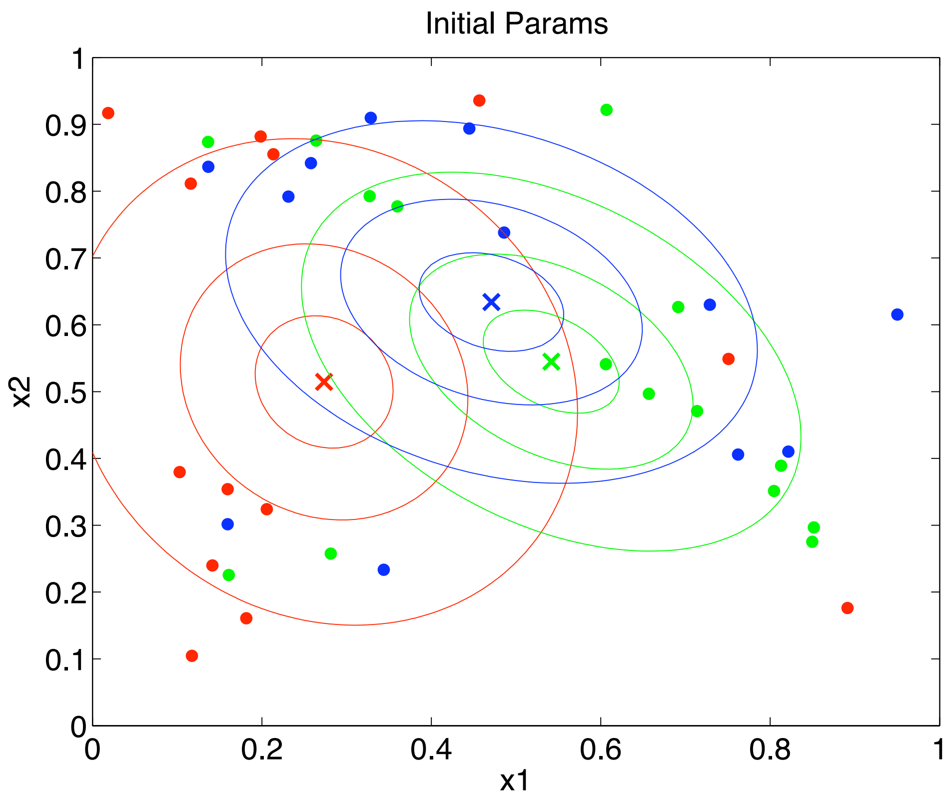
1. Guess initial parameters p_k, μ_k, Σ_k for each class k
2. Repeat until convergence:
   (a) *E-step:* For each instance i and class j, assign each instance to most likely class:
   $$y_i = \arg\max_k P(y_i = k|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|y_i)P(y_i)}{P(\mathbf{x}_i)}$$
   (b) *M-step:* Update the parameters of the model to maximize the likelihood of the data
   $$p_j = \frac{1}{m}\sum_{i=1}^{m}\delta_{ij} \qquad \mu_j = \frac{\sum_{i=1}^{m}\delta_{ij}\mathbf{x}_i}{\sum_{i=1}^{m}\delta_{ij}}$$
   $$\Sigma_j = \frac{\sum_{i=1}^{m}\delta_{ij}\left(\mathbf{x}_i - \mu_j\right)\left(\mathbf{x}_i - \mu_j\right)^T}{\sum_{i=1}^{m}\delta_{ij}}$$

Hard EM for Mixture of Gaussians: Example



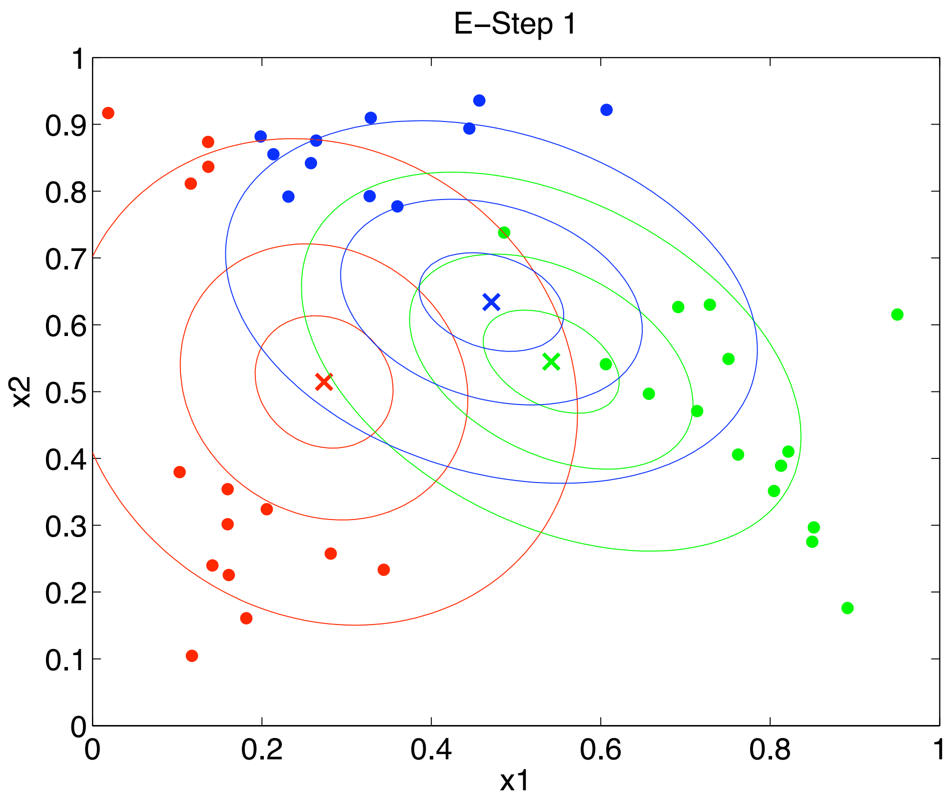
$K = 3$, initial assignment of points to components is random

Hard EM for Mixture of Gaussians: Example

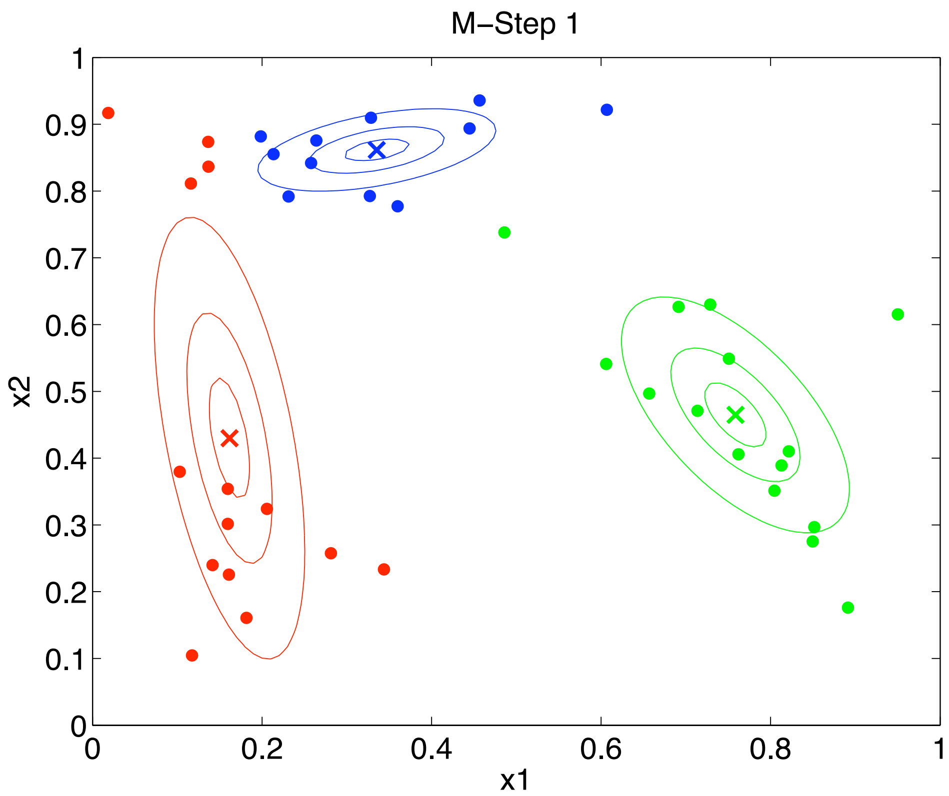


Initial parameters (means and variances) computed from initial assignments

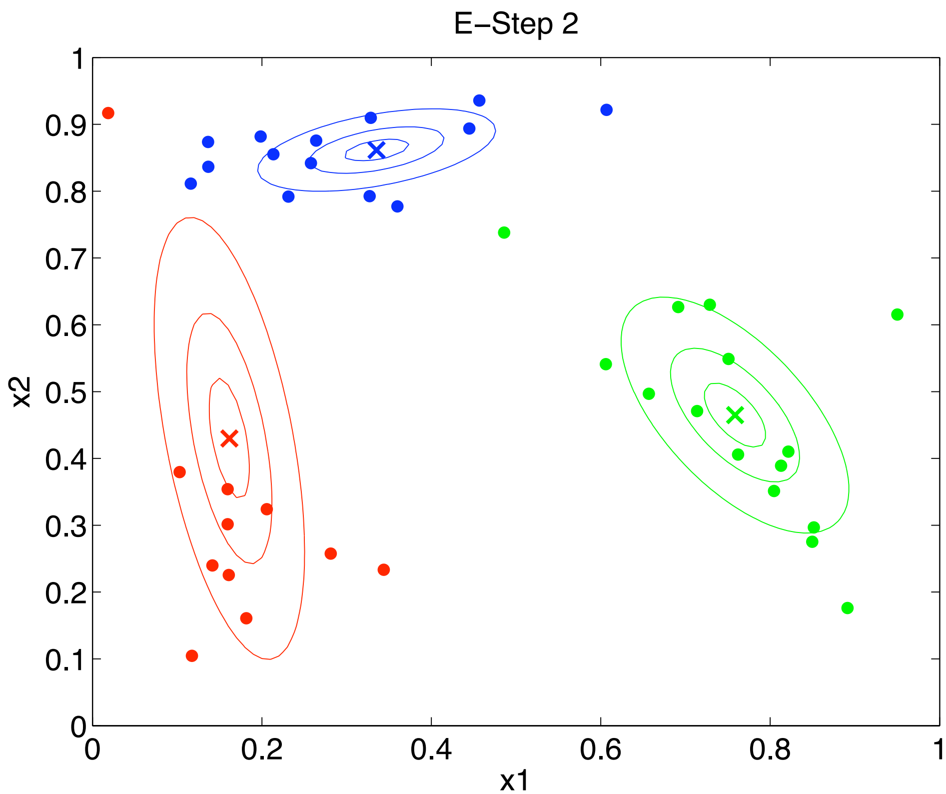
Hard EM for Mixture of Gaussians: Example



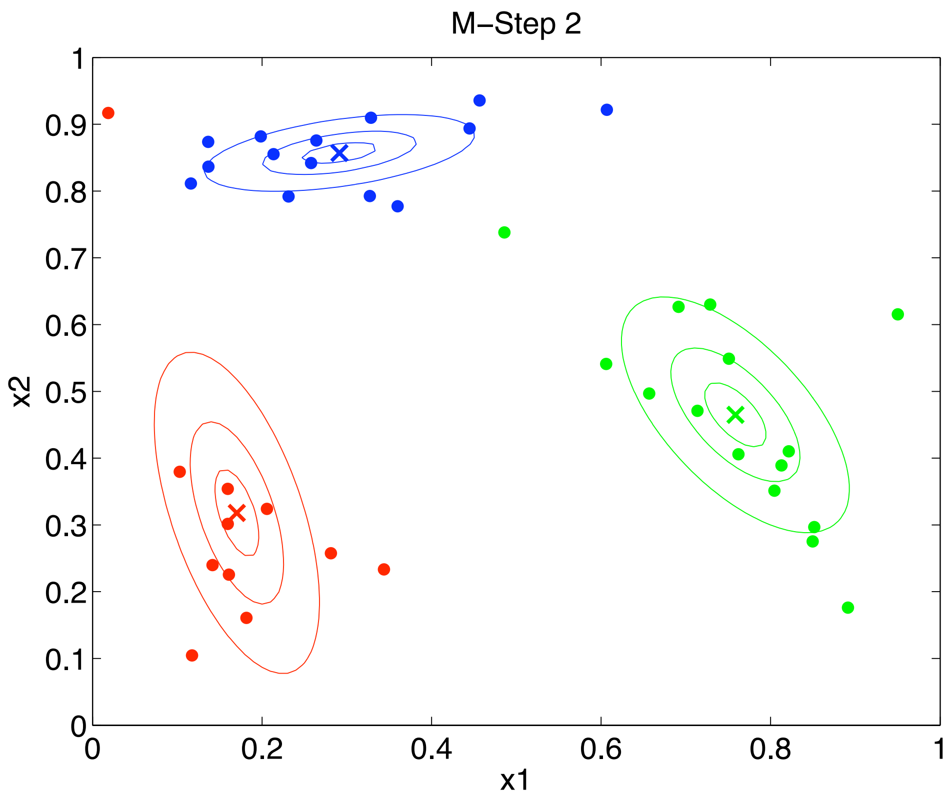
Hard EM for Mixture of Gaussians: Example



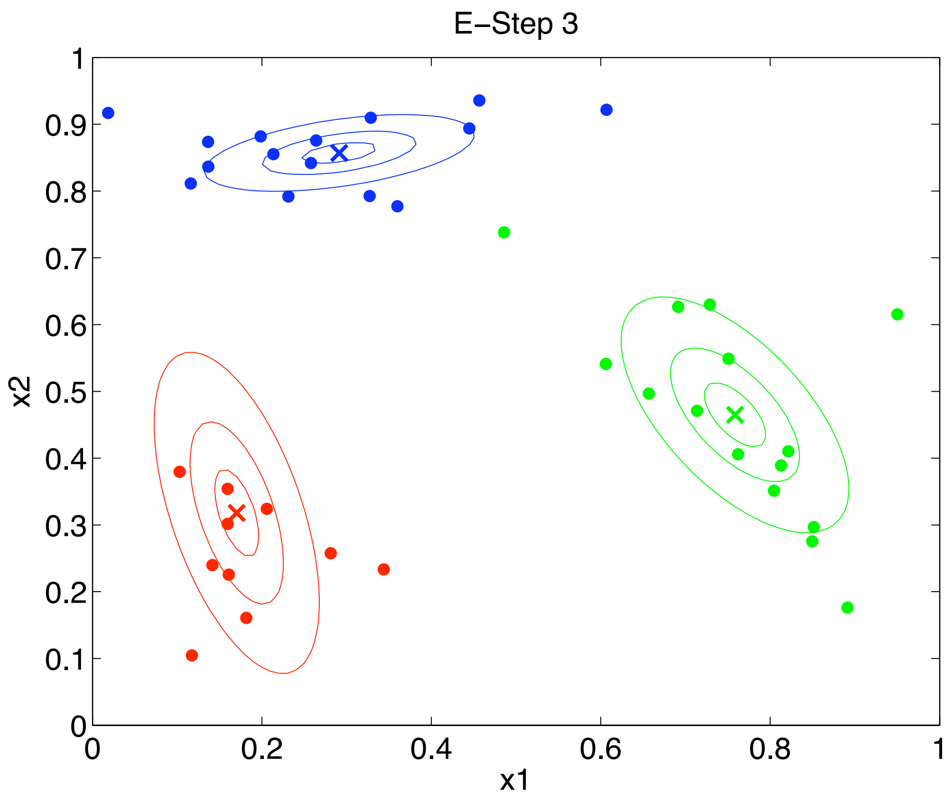
Hard EM for Mixture of Gaussians: Example



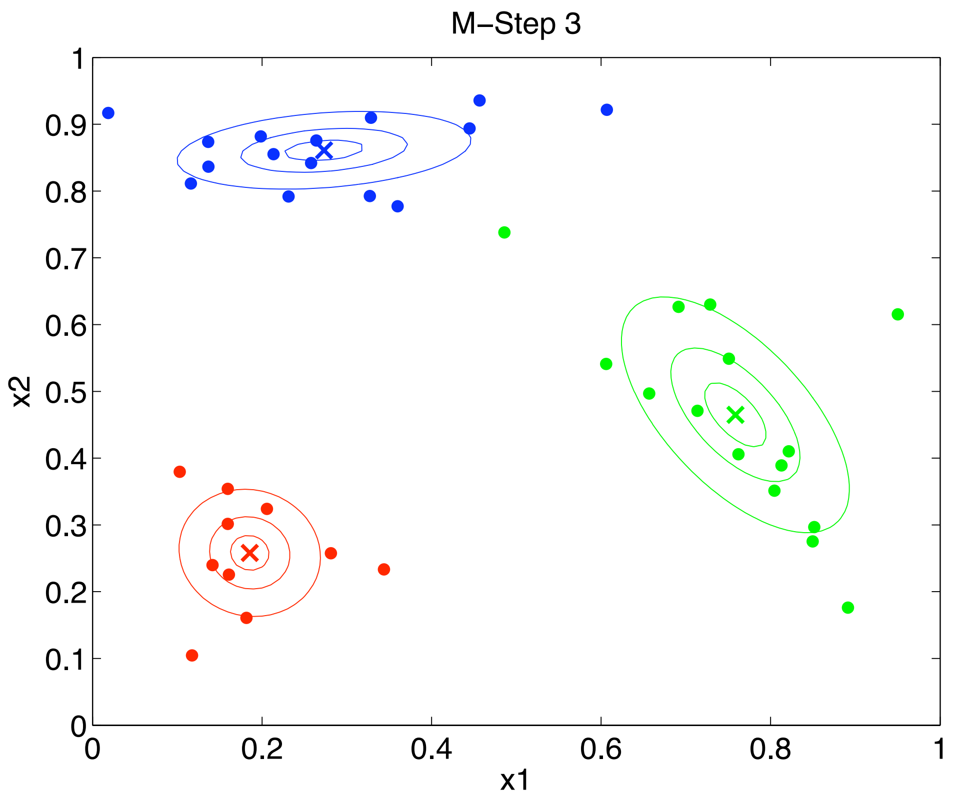
Hard EM for Mixture of Gaussians: Example



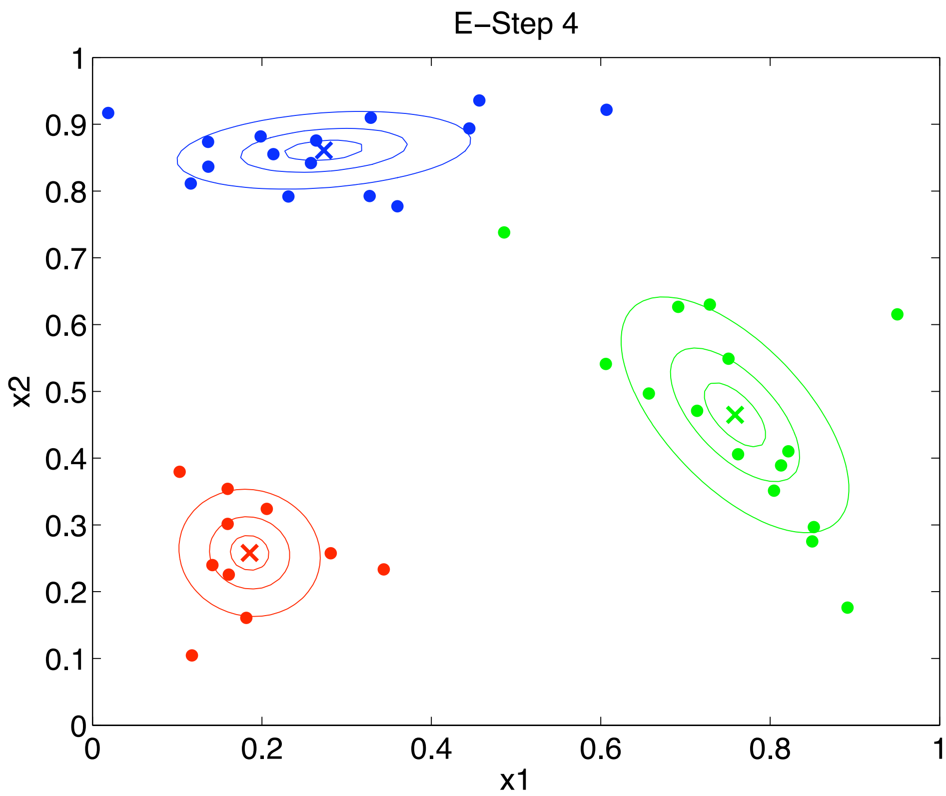
Hard EM for Mixture of Gaussians: Example



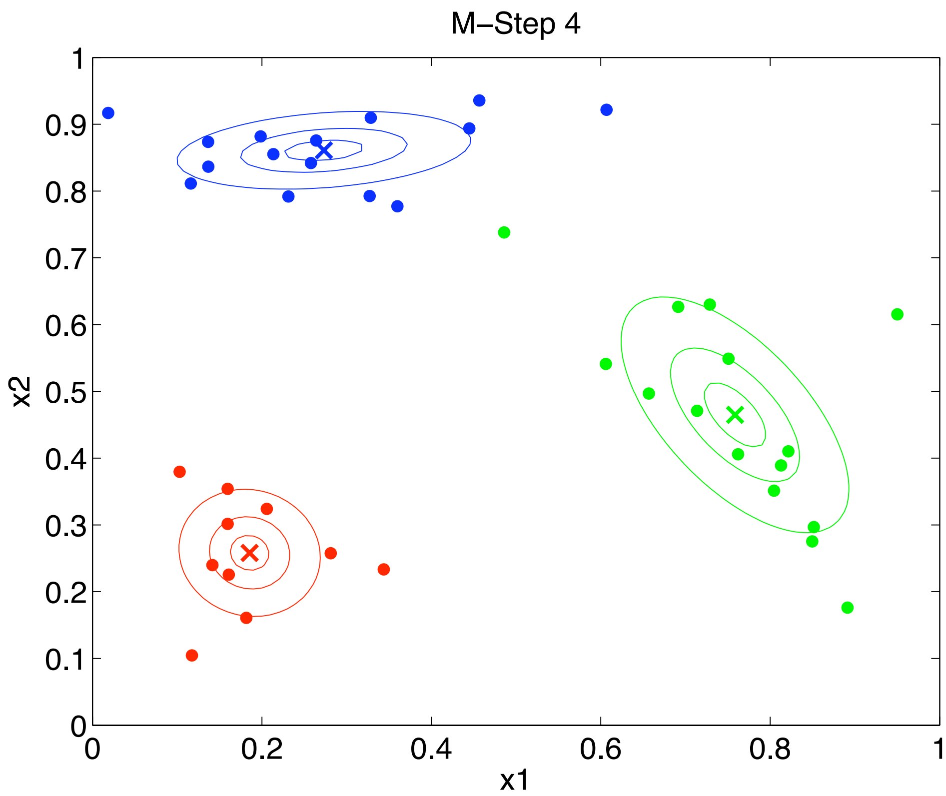
Hard EM for Mixture of Gaussians: Example



Hard EM for Mixture of Gaussians: Example



Hard EM for Mixture of Gaussians: Example



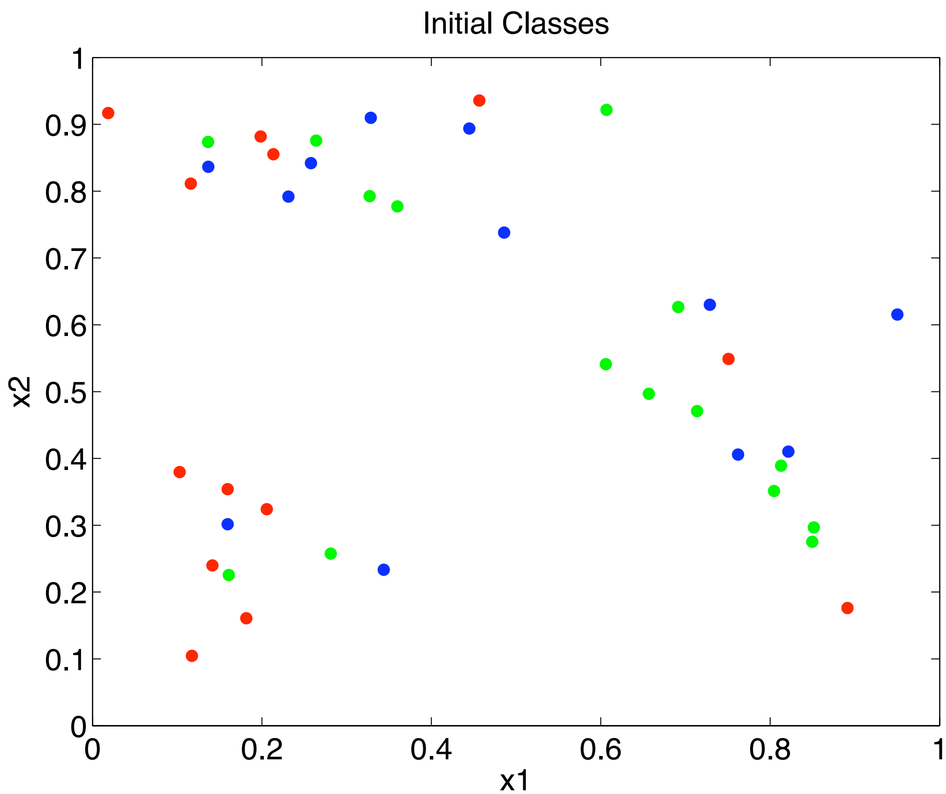
Soft EM for Mixture of Gaussians

1. Guess initial parameters p_k, μ_k, Σ_k for each class k
2. Repeat until convergence:
   (a) *E-step:* For each instance i and class j, compute the probabilities of class membership: $w_{ij} = P(y_i = j|\mathbf{x}_i)$
   (b) *M-step:* Update the parameters of the model to maximize the likelihood of the data

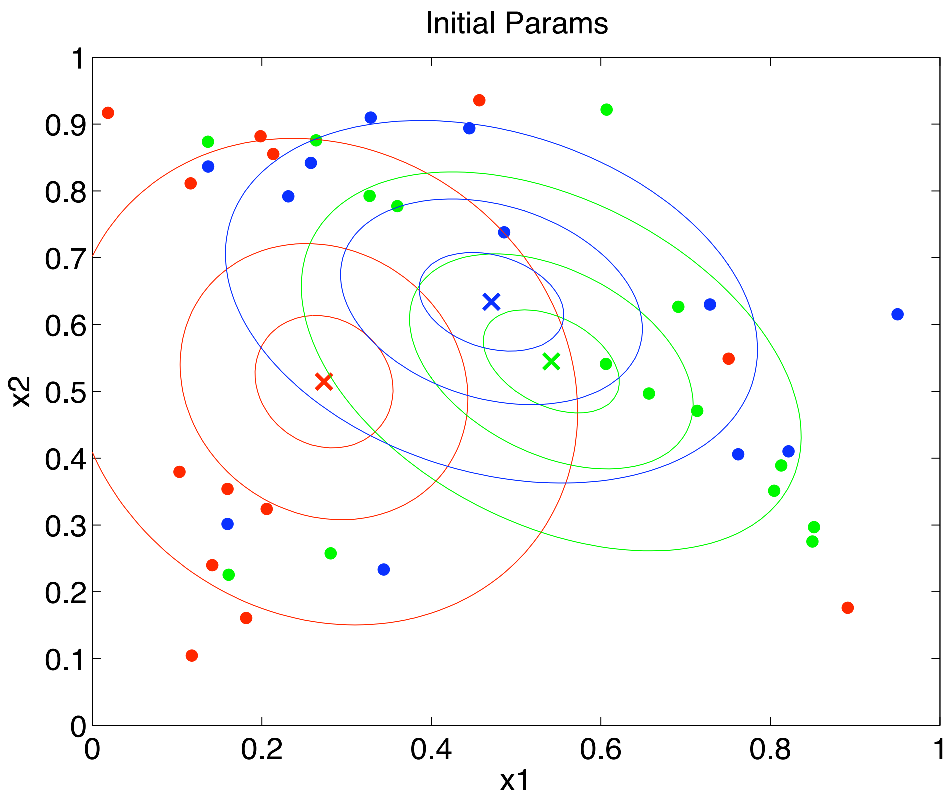
$$p_j = \frac{1}{m}\sum_{i=1}^{m} w_{ij} \qquad \mu_j = \frac{\sum_{i=1}^{m} w_{ij}\mathbf{x}_i}{\sum_{i=1}^{m} w_{ij}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_{ij}\left(\mathbf{x}_i - \mu_j\right)\left(\mathbf{x}_i - \mu_j\right)^T}{\sum_{i=1}^{m} w_{ij}}$$

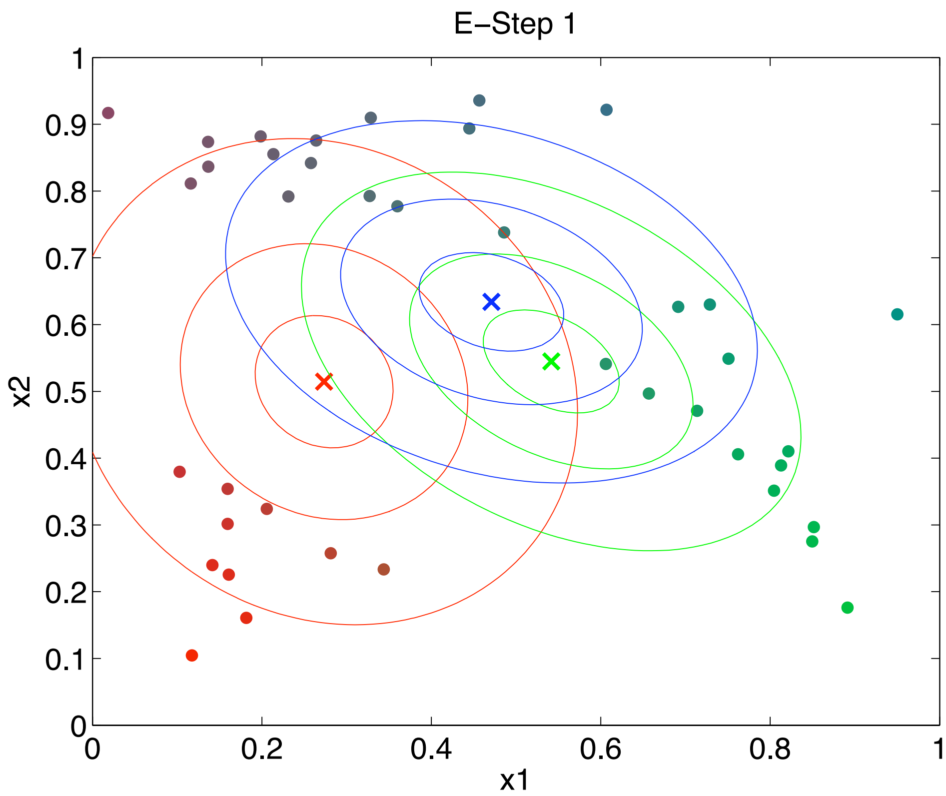
Soft EM for Mixture of Gaussians: Example



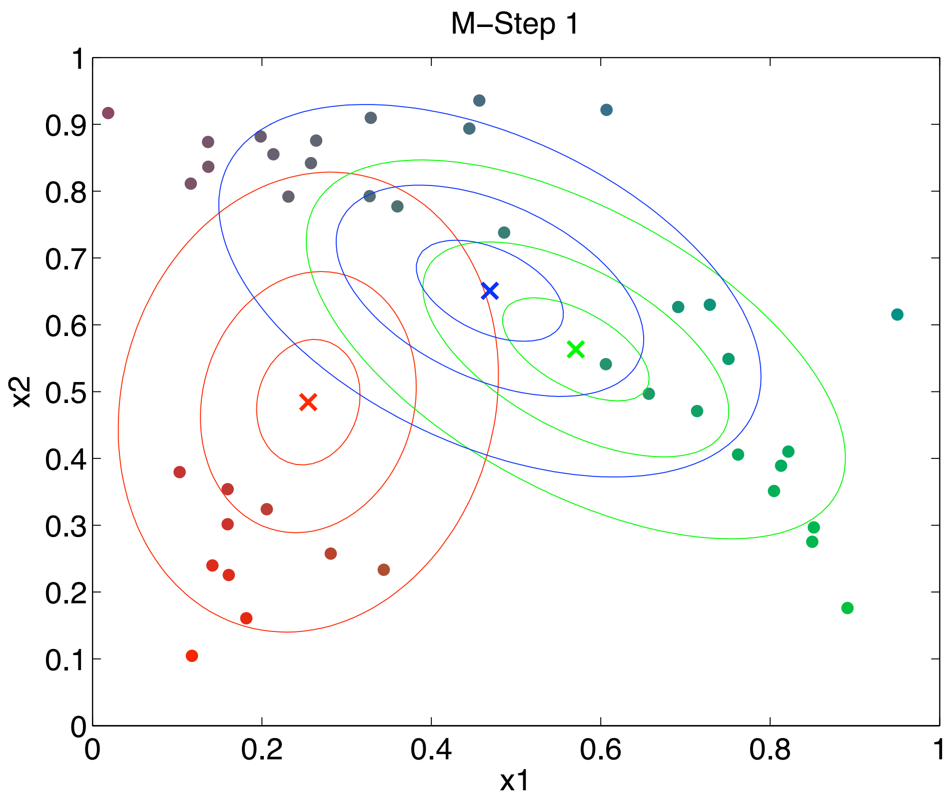
Soft EM for Mixture of Gaussians: Example



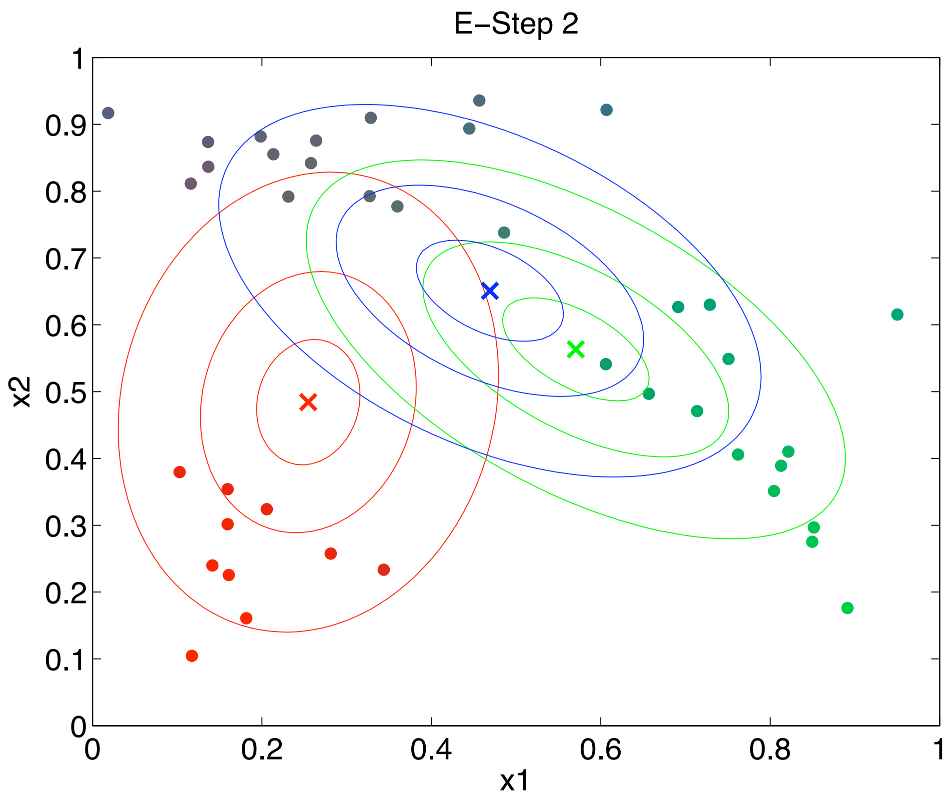
Soft EM for Mixture of Gaussians: Example



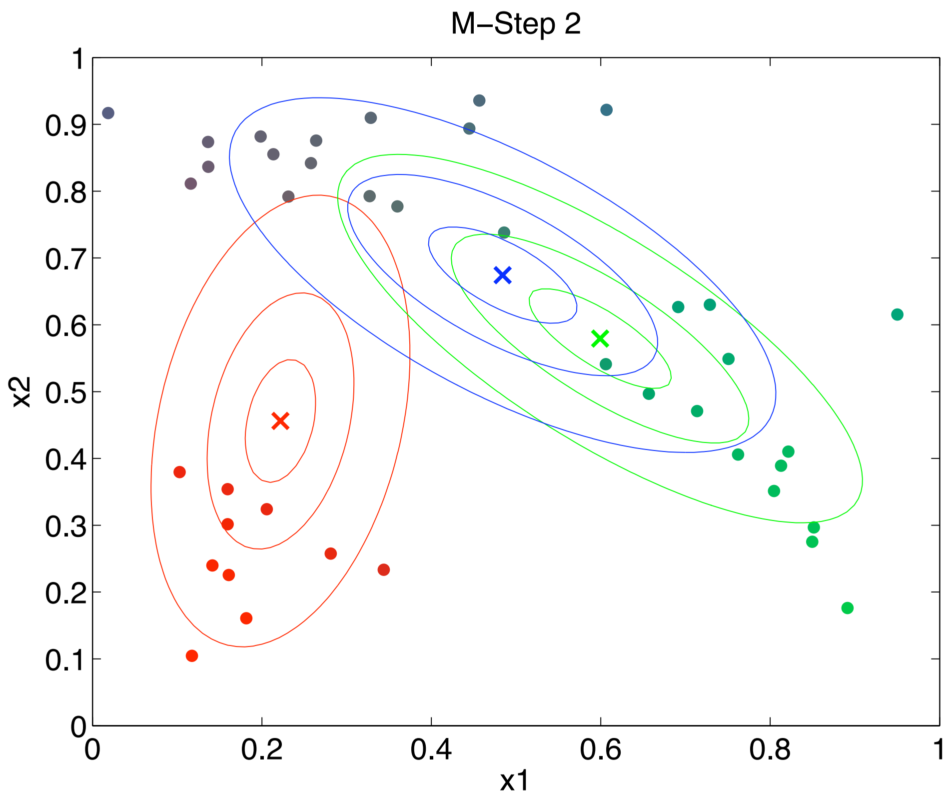
Soft EM for Mixture of Gaussians: Example



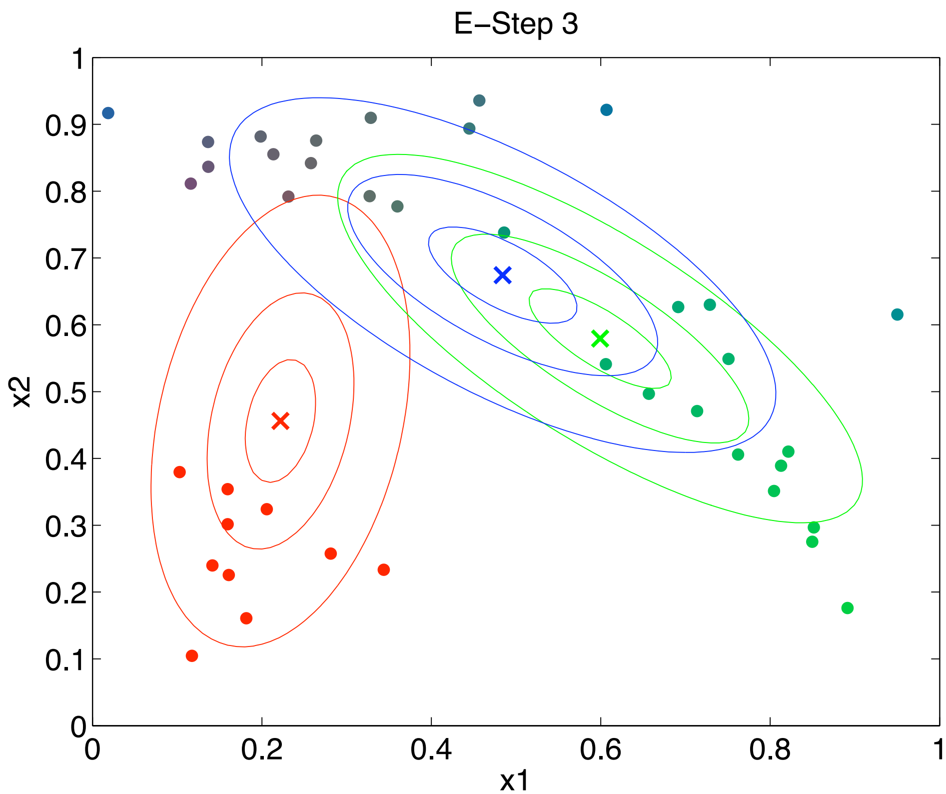
Soft EM for Mixture of Gaussians: Example



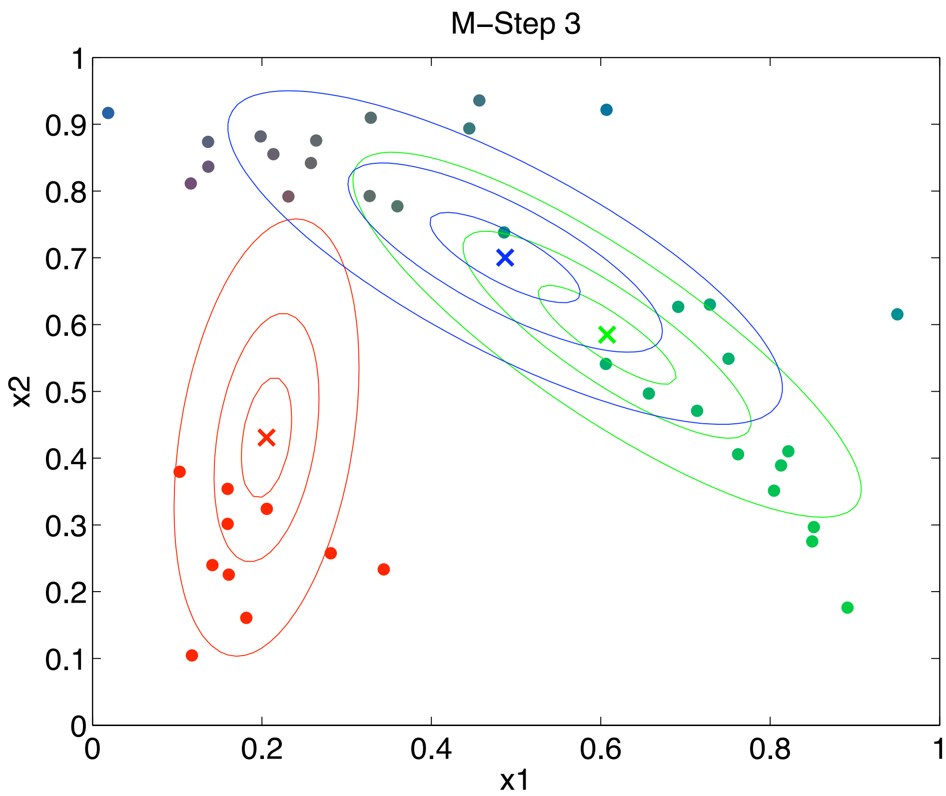
Soft EM for Mixture of Gaussians: Example



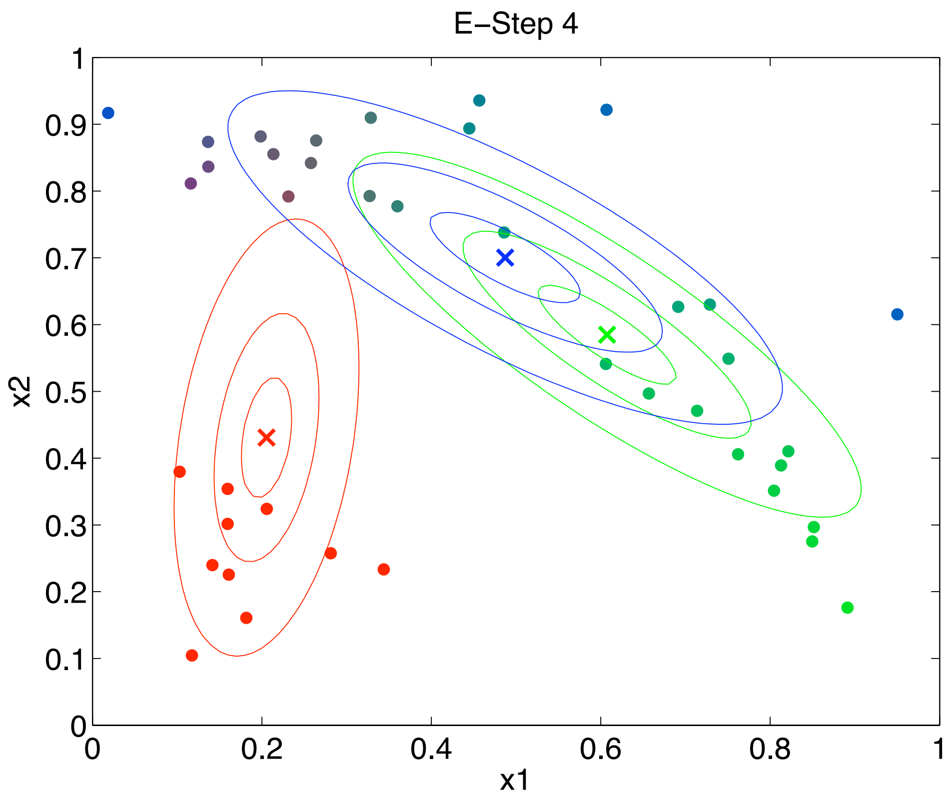
Soft EM for Mixture of Gaussians: Example



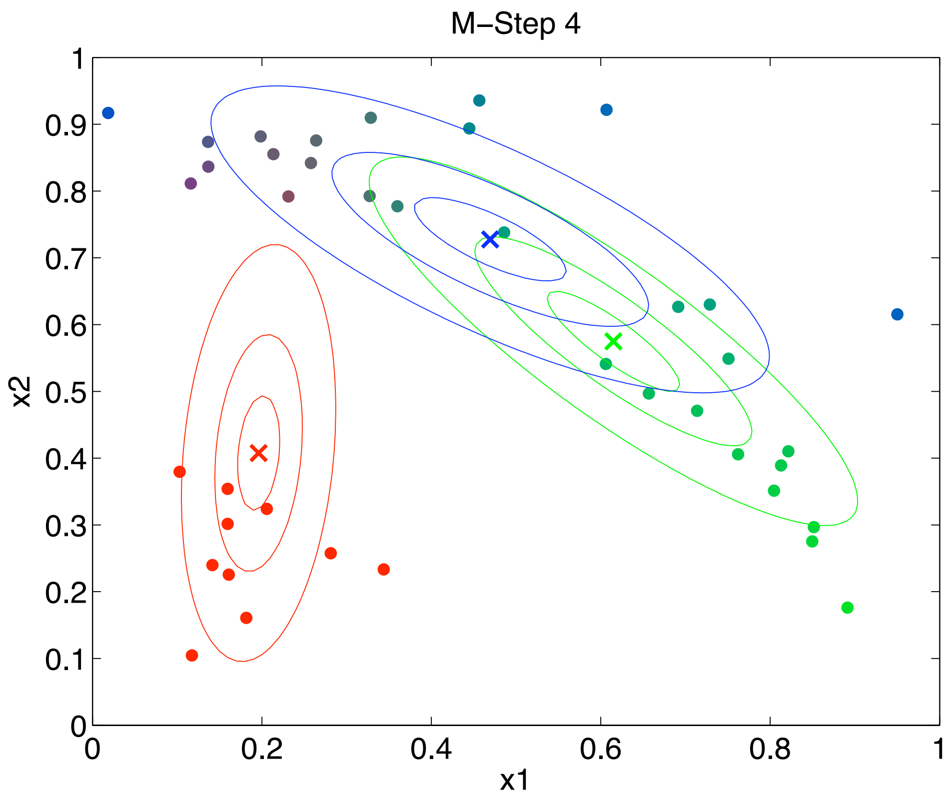
Soft EM for Mixture of Gaussians: Example



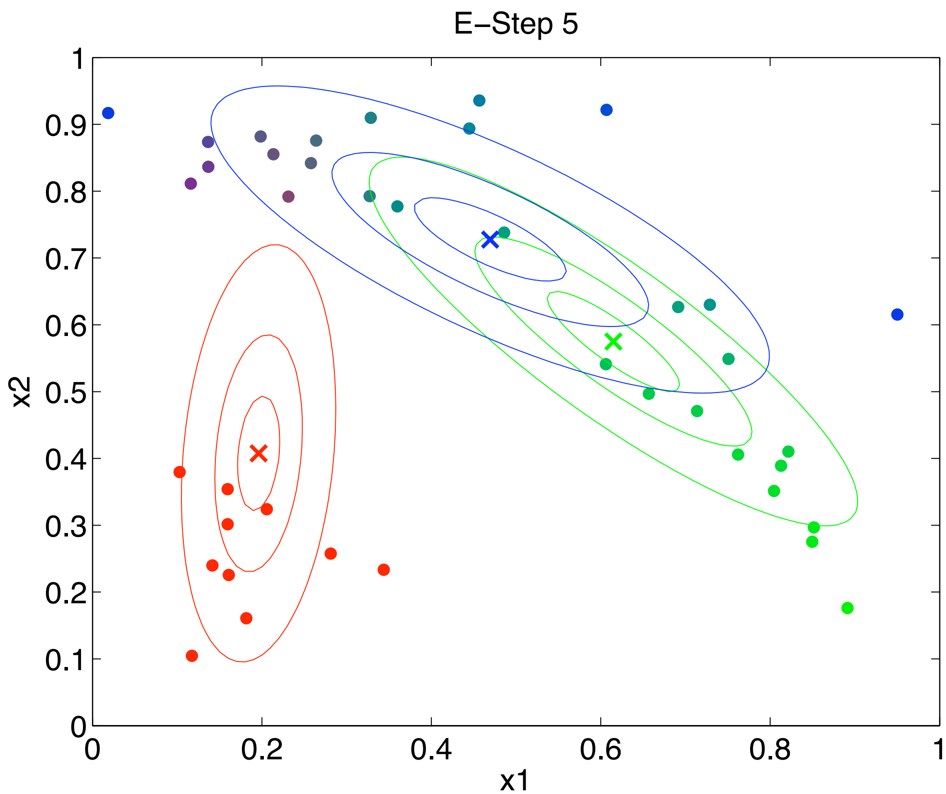
Soft EM for Mixture of Gaussians: Example



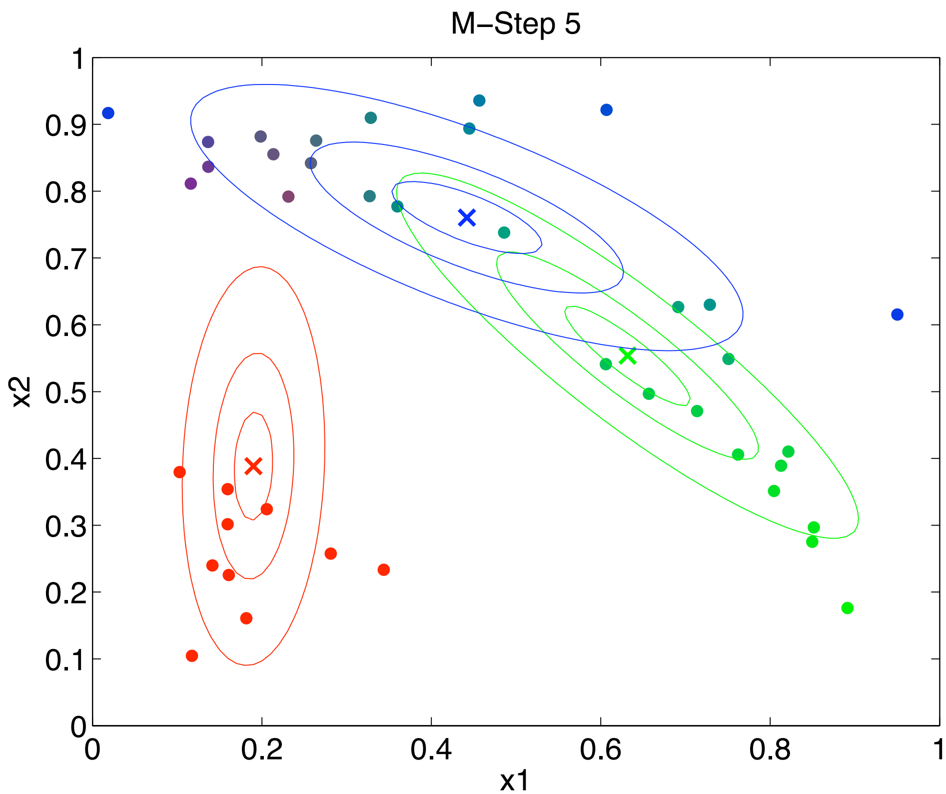
Soft EM for Mixture of Gaussians: Example



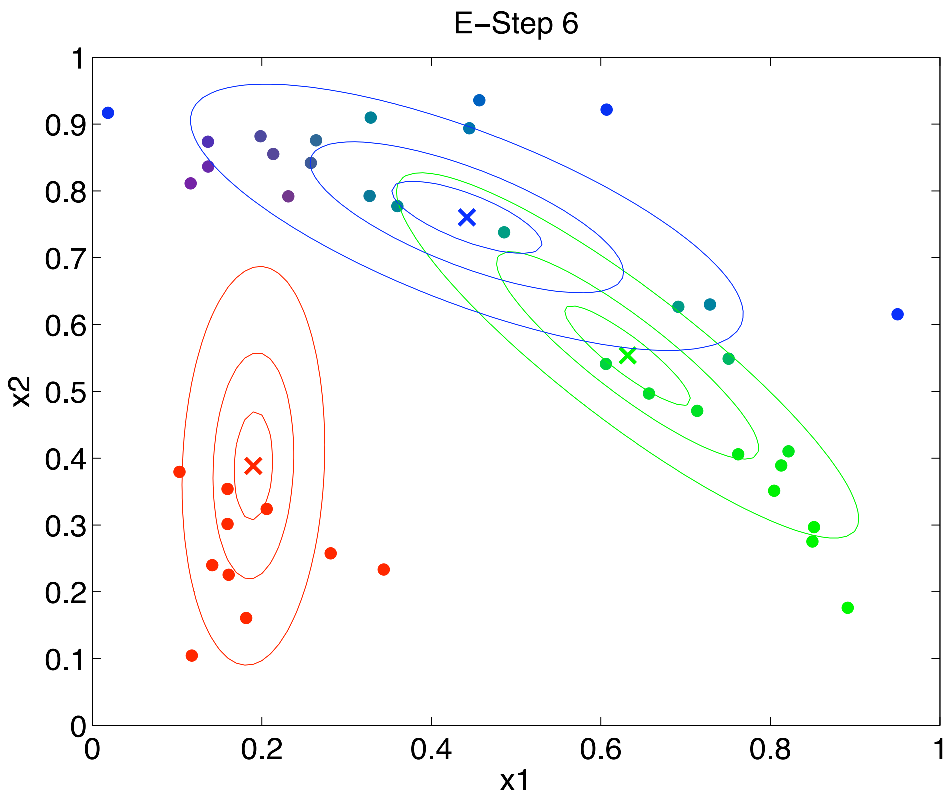
Soft EM for Mixture of Gaussians: Example



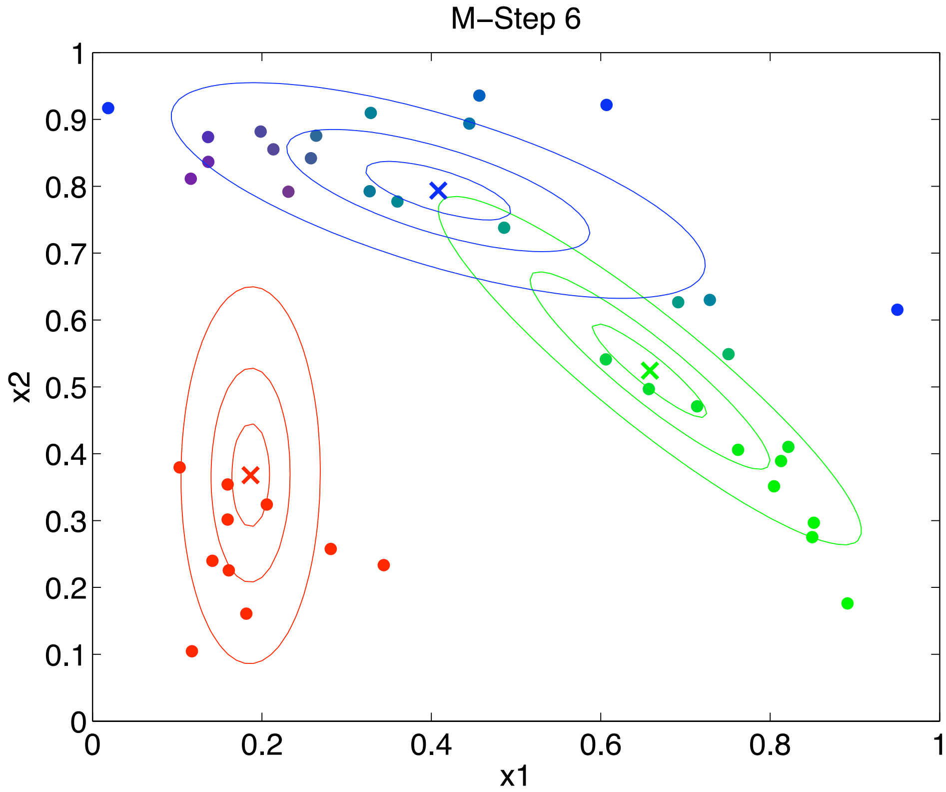
Soft EM for Mixture of Gaussians: Example



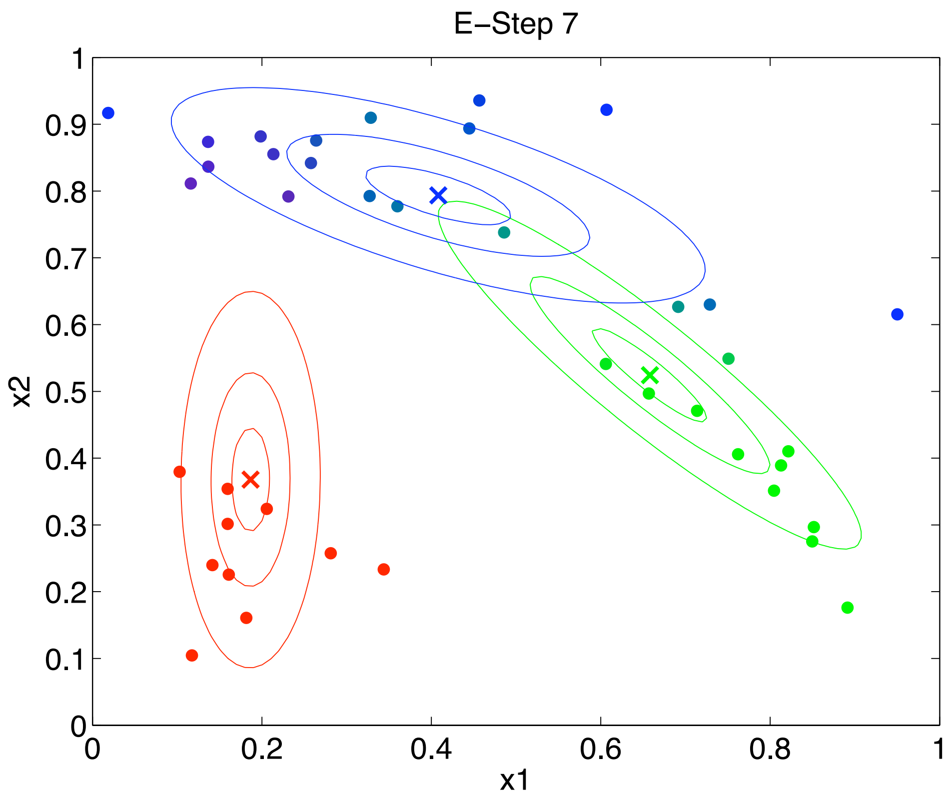
Soft EM for Mixture of Gaussians: Example



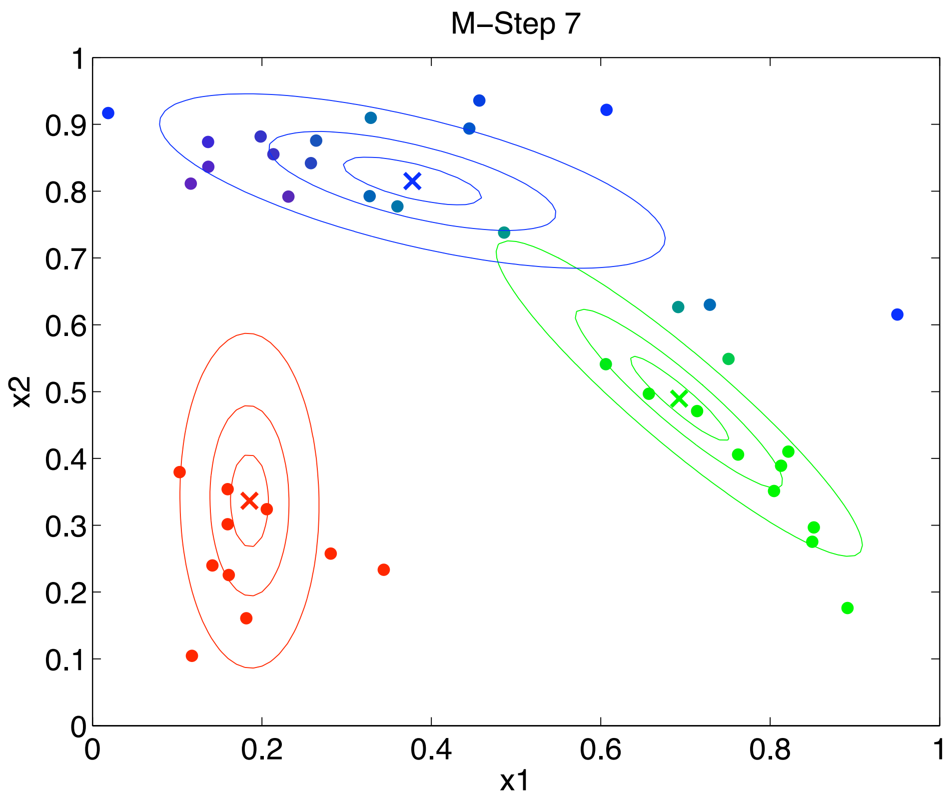
Soft EM for Mixture of Gaussians: Example



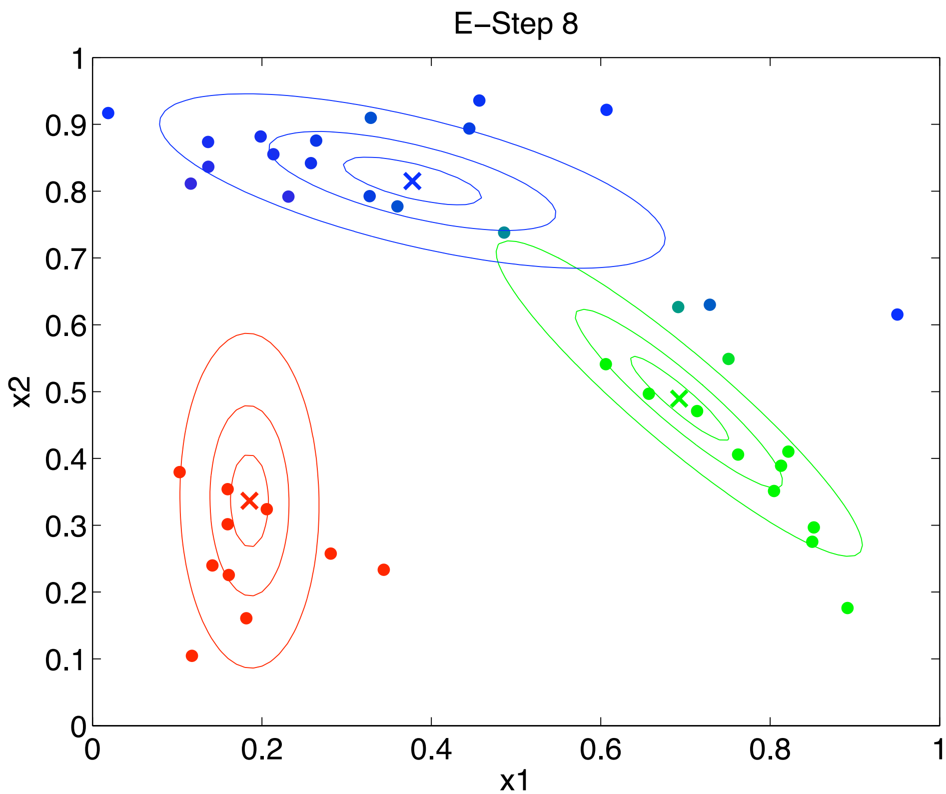
Soft EM for Mixture of Gaussians: Example



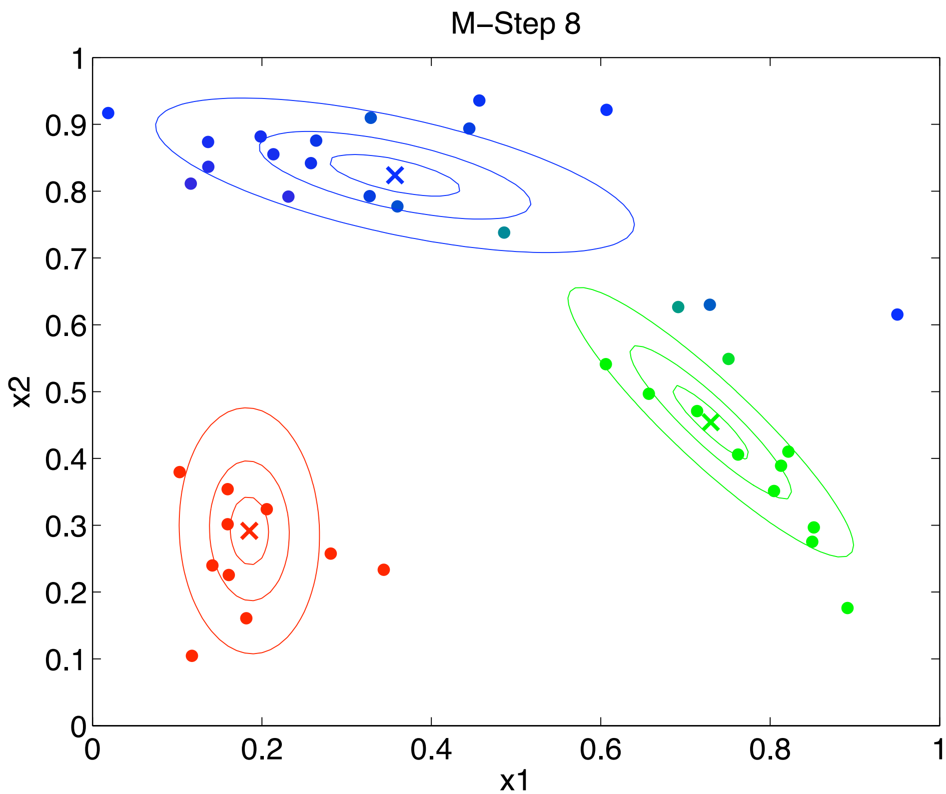
Soft EM for Mixture of Gaussians: Example



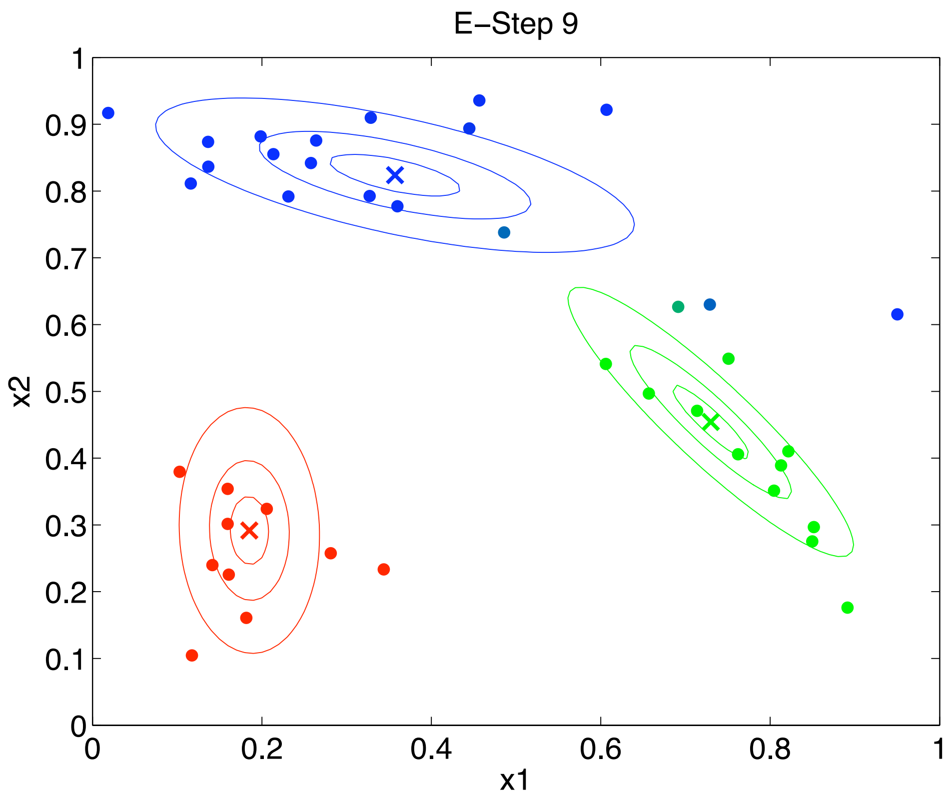
Soft EM for Mixture of Gaussians: Example



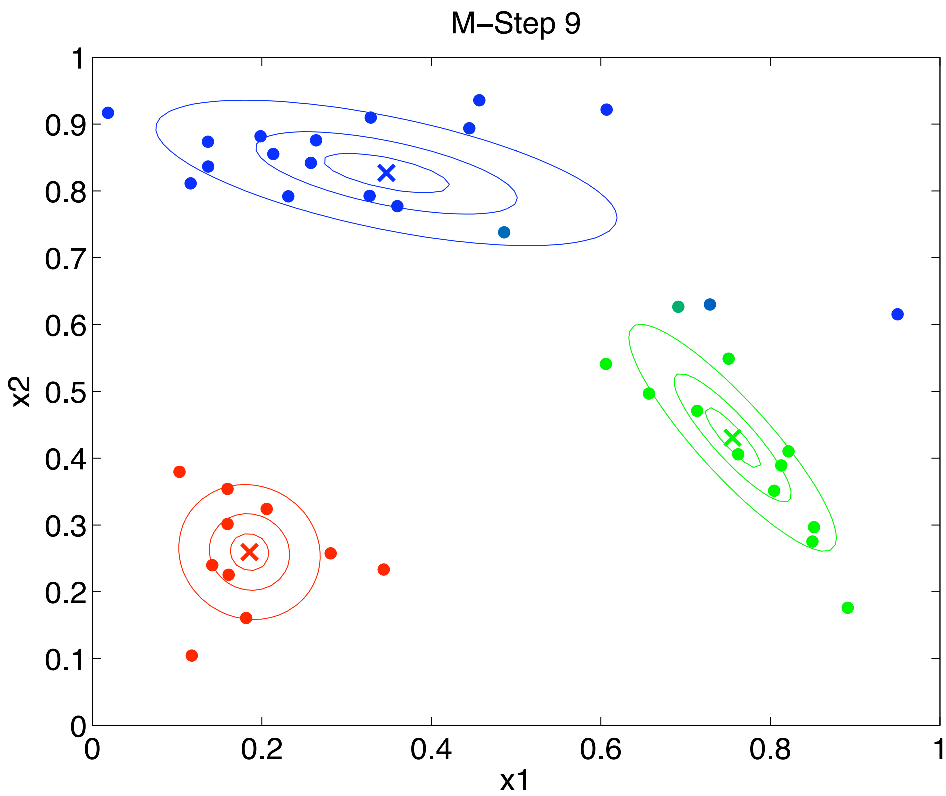
Soft EM for Mixture of Gaussians: Example



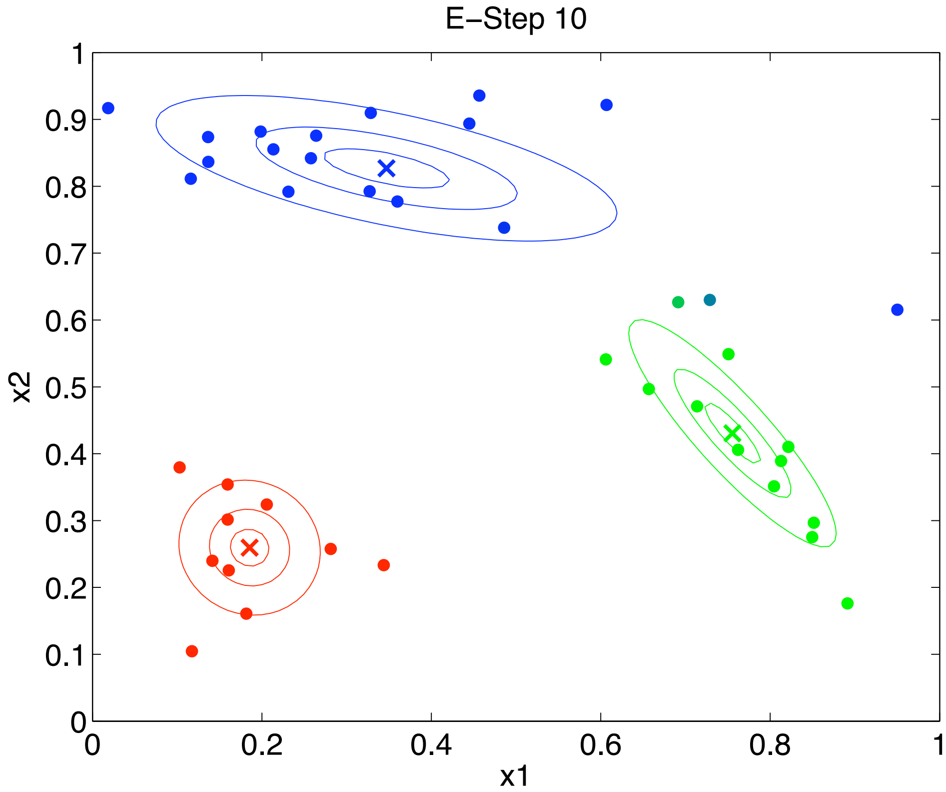
Soft EM for Mixture of Gaussians: Example



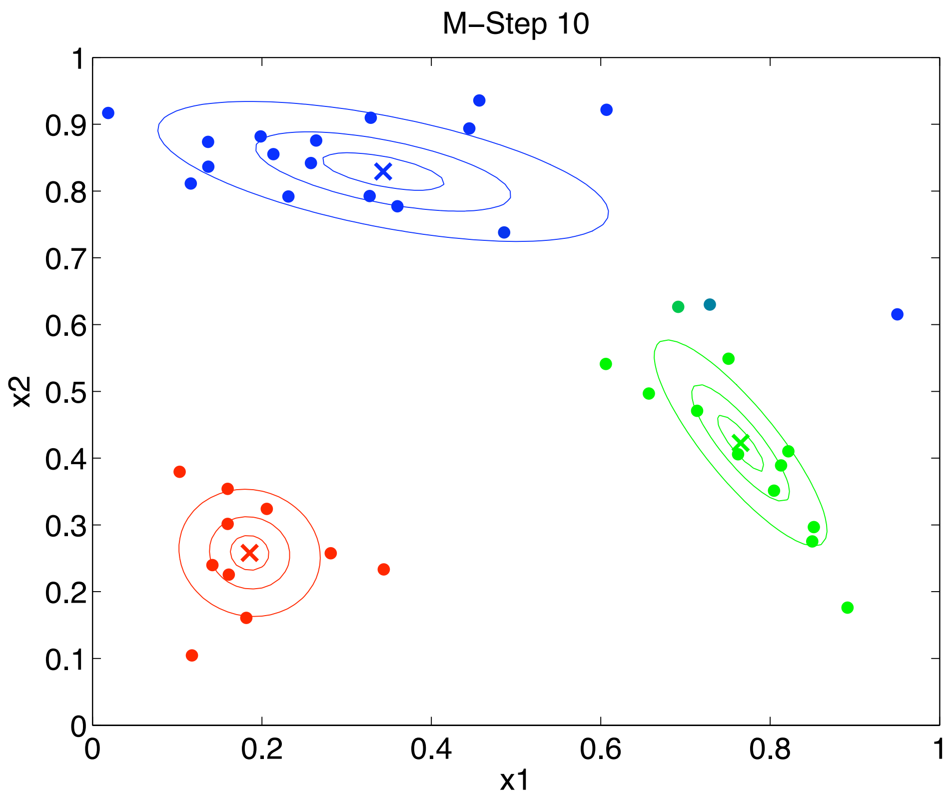
Soft EM for Mixture of Gaussians: Example



Soft EM for Mixture of Gaussians: Example



Soft EM for Mixture of Gaussians: Example



Comparison of hard EM and soft EM

- Soft EM does not commit to a particular value of the missing item. Instead, it considers all possible values, with some probability
- This is a pleasing property, given the uncertainty in the value
- Soft EM is almost always the method of choice (and often when people say "EM", they mean the soft version)
- The complexity of each iteration of the two versions is pretty much the same.
- Soft EM might take more iterations, if we stop it based on numerical value converegnce.

Theoretical properties of EM

- Each iteration improves the likelihood of the data given the class assignments, p_j, μ_j, and Σ_j.
  - Straightforward for Hard EM.
  - Less obvious for Soft EM.
- The algorithm works by making a convex approximation to the log-likelihood (by filling in the data)
- If the parameters do not change in one iteration, then the gradient of the log-likelihood function is zero

Warning: mixture components converging to a point

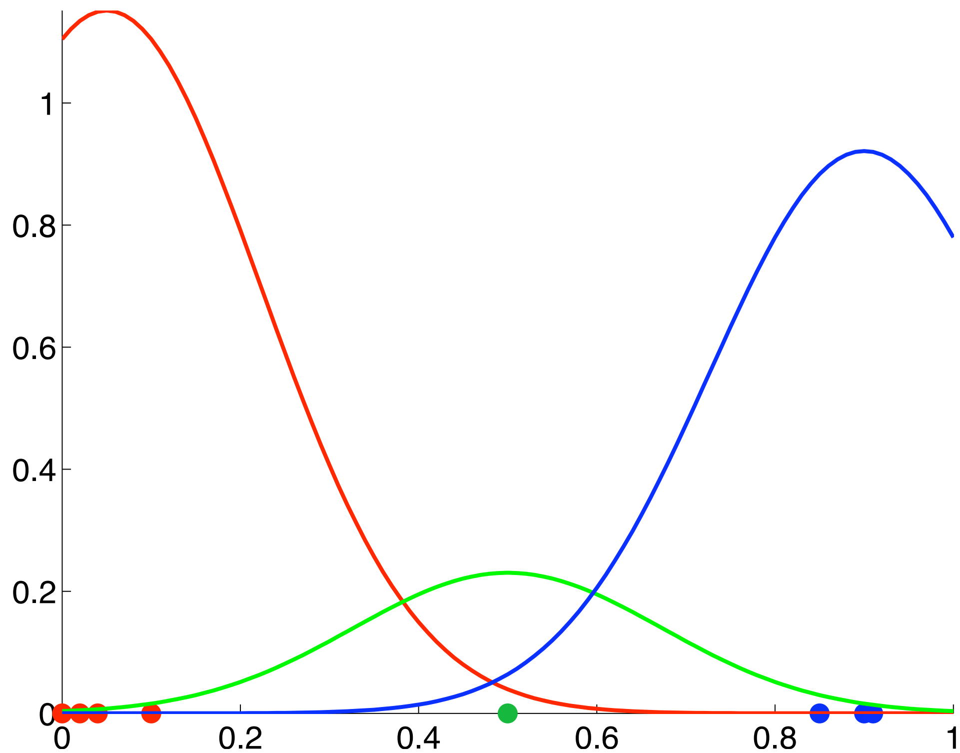
- What happens in Hard EM if a class contains a single point?

$\Rightarrow$ The covariance matrix is not defined!

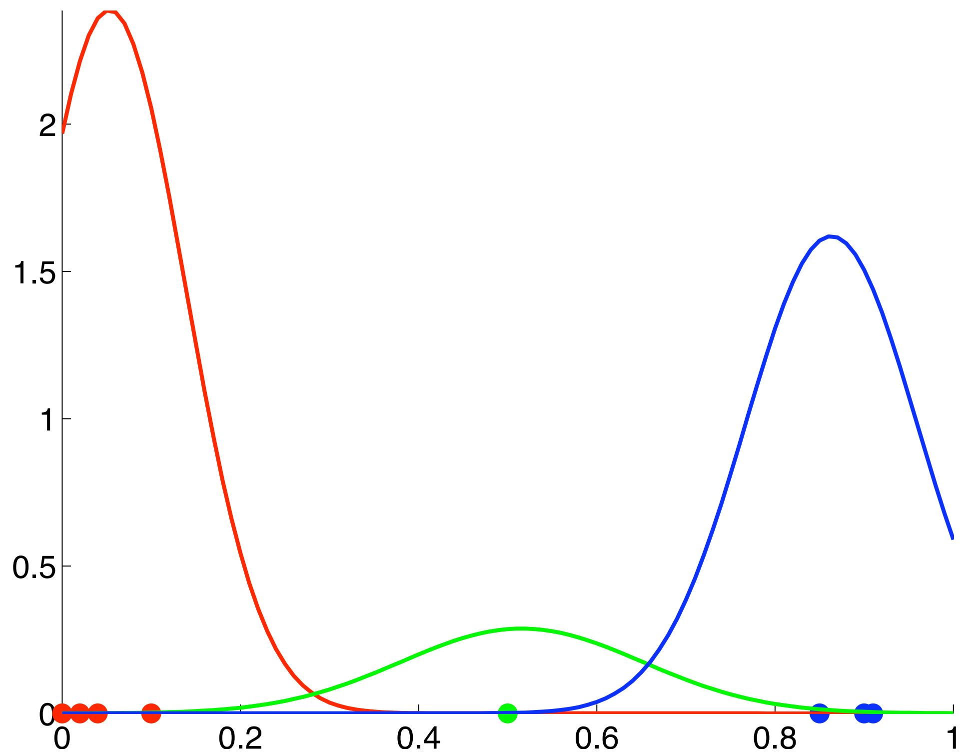
- Similarly, what happens in Soft EM if a class focusses more and more on a single point over iterations?

$\Rightarrow$ The covariance matrix goes to zero! And the likelihood of the data goes to $+\infty$! (See following slides.)

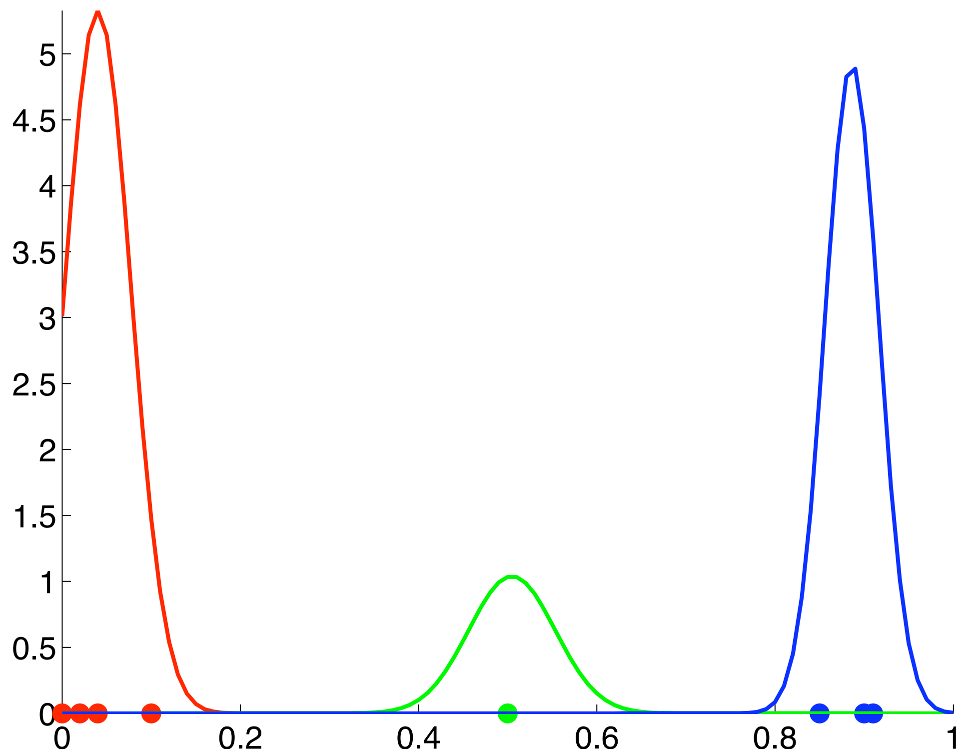
Mixture converging to a point: Example



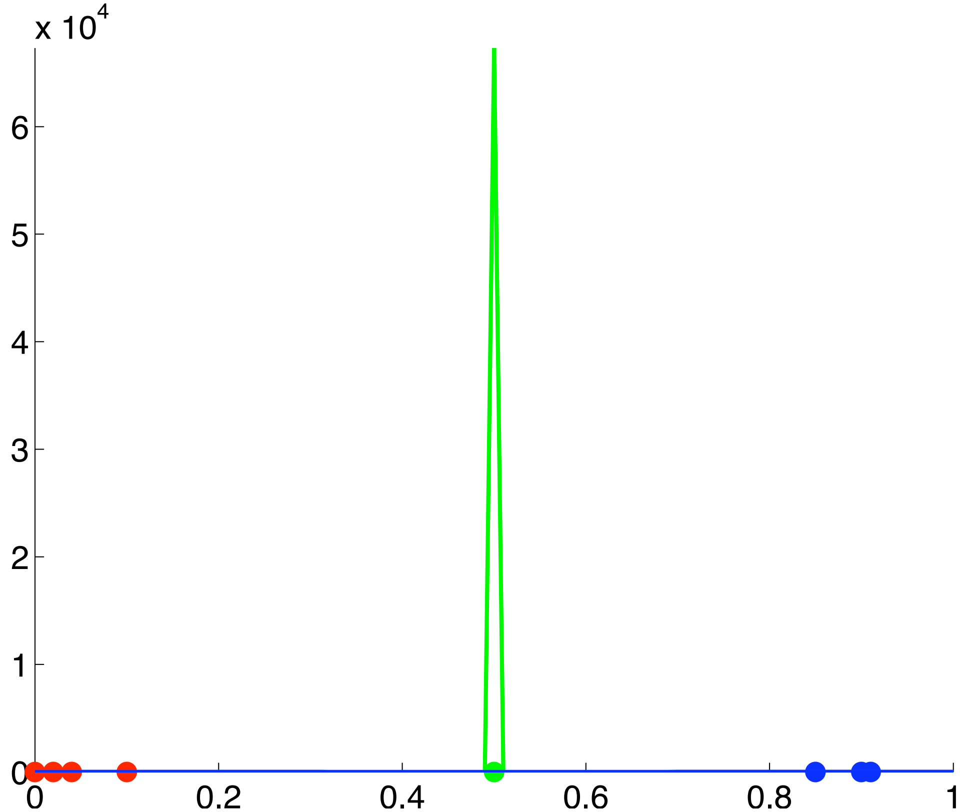
Mixture converging to a point: Example



Mixture converging to a point: Example



Mixture converging to a point: Example



Variations

- If only some of the data is incomplete, the likelihood will have one component based on the complete instances and another ones based on incomplete instances
- Sparse EM: Only compute probability at a few data points (most values will be close to 0 anyway)
- Instead of a complete M-step, just improve the likelihood a bit
- Note that EM can be stuck in *local minima*, so it has to be restarted!
- It works very well for low-dimensional problems, but can have problems if θ is high-dimensional.

Summary of EM

- EM is guaranteed to converge to a local optimum of the likelihood function. Since the optimum is *local*, starting with different values of the initial parameters is necessary
- Can be used for virtually any application with missing data/latent variables
- The algorithm can be stopped when no more improvement is achieved between iterations.
- A big hammer that fits all sorts of practical problems