Expectation-Maximization

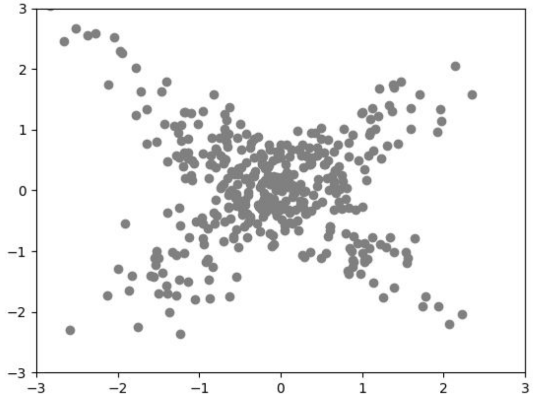
CS 580
Intro to Artificial Intelligence

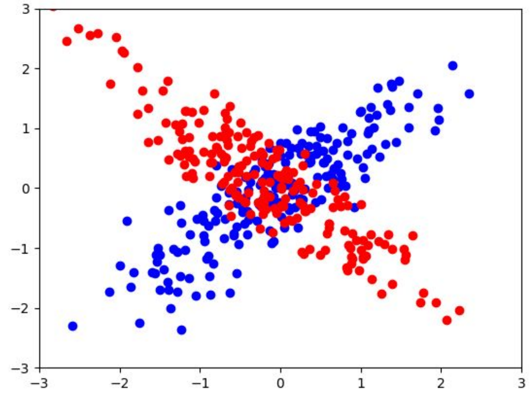
Soft Clustering

For some datasets, it’s useful to have a probabilistic or **soft** assignment of data points to clusters

In this setting, the partition function is less important than the **size**, **shape**, and **location** of the clusters

To compute this, we’re going to assume our data is **generated** by some non-deterministic process



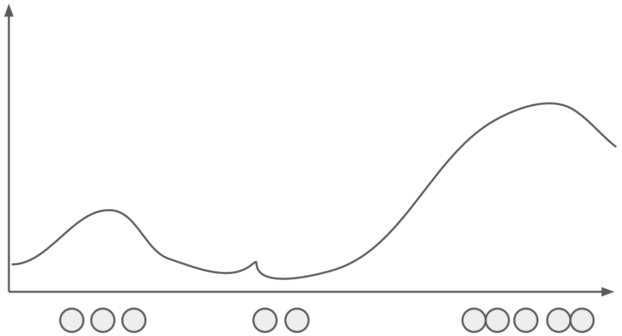


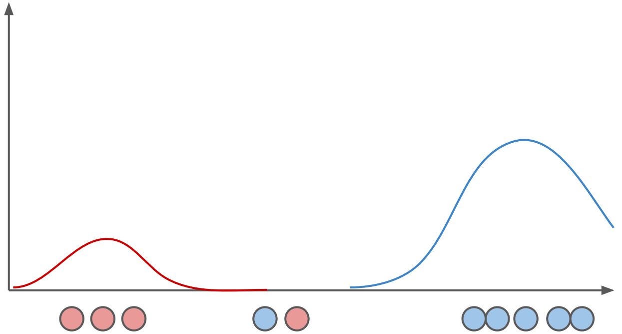
Data generation process

For K clusters, assume there are K distributions which generated the data. Each point is generated in the following way

1. Pick one of the distributions randomly
2. Sample **x** from that distribution

This is known as a **mixture model**, and when the underlying distributions are Gaussian, a **G**aussian **m**ixture **m**odel, or **GMM**.





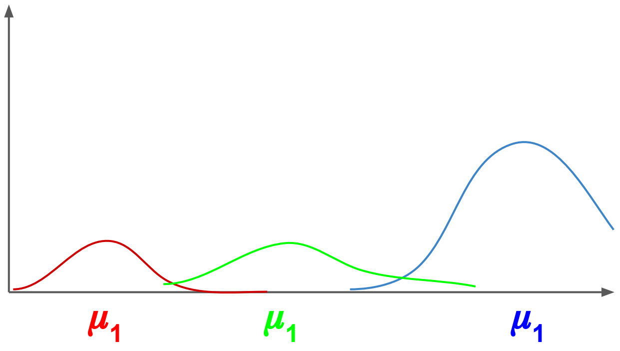
Gaussian parameters

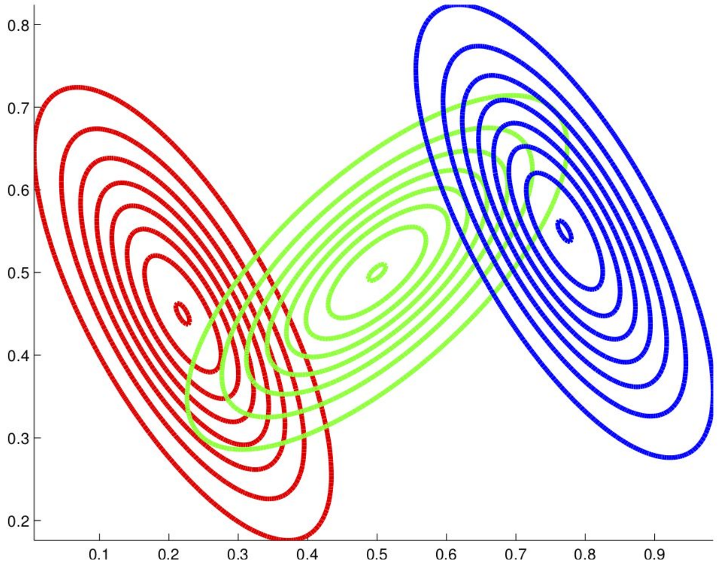
In 1D

$$p(x;\mu_i,\sigma_i^2)=\frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left\{-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right\}$$

In higher dimensions

$$p(\mathbf{x};\mu_i,\Sigma_i)=\frac{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_i)^\top\Sigma_i^{-1}(\mathbf{x}-\mu_i)\right\}}{\sqrt{(2\pi)^d(\det\Sigma_i)}}$$





Estimating parameters (1)

What's the MLE of the parameters of a Gaussian for a set of points? Easy, just the sample mean and covariance!

Unfortunately, it's not a **single** Gaussian, it's a **mixture**. Let $Z^{(i)}$ be the random variable representing which mixture generated $\mathbf{x}^{(i)}$

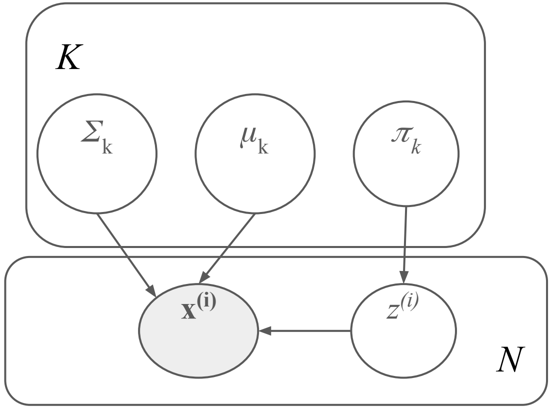
$$\theta = \{(\mu_k, \Sigma_k, \pi_k)\}_{k=1}^{K}$$

$$p(Z^{(i)} = k) = \pi_k$$

$$p(\mathbf{x}, Z = k \mid \theta) = \pi_k \cdot p(\mathbf{x} \mid \mu_k, \Sigma_k)$$

$$p(\mathbf{x} \mid \theta) = \sum_{k=1}^{K} p(\mathbf{x}, Z = k \mid \theta) = \sum_{k=1}^{K} [\pi_k \cdot p(\mathbf{x} \mid \mu_k, \Sigma_k)]$$

The Bayes net reveals an interesting structure.



We want to estimate π_k, μ_k, $\mathbf{\Sigma}_k$ given the data

Estimating parameters (2)

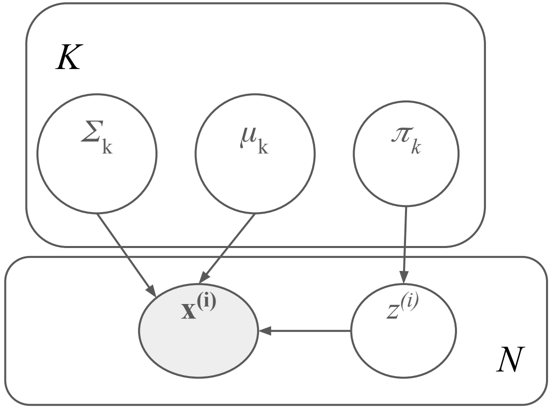
Notice: given **x**, the parameters are *not* independent of one another.

BUT: if Z_k were observed, π_k is independent of μ_k and $\boldsymbol{\Sigma}_k$

A two step process (like k-means):

1. Fix the parameters θ, estimate expected value of Z_k
2. Fix Z_k, compute MLE of the parameters θ

This technique generalizes beyond GMMs and is called **E**xpectation-**M**aximization (or EM)



Maximizing likelihood with hidden variables

For models that combine some **observed** random variables $\mathbf{x}^{(\mathbf{i})}$ and some **hidden** random variables $Z^{(i)}$ we'd like to maximize the log likelihood

$$\ell(\theta) = \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)} \mid \theta) = \sum_{i=1}^{N} \log \left[\sum_{h} p(\mathbf{x}^{(i)}, Z^{(i)} = h \mid \theta) \right]$$

Since we can't move the log inside the second sum, this can be challenging to optimize even for simple distributions (Gaussian, exponential family, etc)

Maximizing “complete data” log likelihood

What if we **knew** what each $Z^{(i)}$ was? We can define the **complete data log likelihood** as

$$\ell_c(\theta) = \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)}, z^{(i)} \mid \theta)$$

We can't compute this directly, since we don't know $Z^{(i)}$ so let's work with the **expected complete data log likelihood**

$$Q(\theta, \theta^{(t-1)}) = \underline{\mathbb{E}[\ell_c(\theta) \mid S, \theta^{(t-1)}]}$$

“Auxiliary” function

Expectation over possible $Z^{(i)}$ values

Expectation-Maximization

Only need to calculate the terms in Q(θ, $\theta^{(t-1)}$) that the argmax in the next step depends on. These are called the **e**xpected **s**ufficient **s**tatistics (ESS)

The E step:

$$\text{Compute } Q(\theta, \theta^{(t-1)})$$

The M step:

$$\theta^{(t)} = \arg\max_{\theta} Q(\theta, \theta^{(t-1)})$$

We can show that alternating between these two steps monotonically increases the log likelihood of the observed data!

EM for GMMs (1)

We can plug in definitions for our GMM to the EM definitions

Definition

Use $z^{(i)}$ to "select" the correct Gaussian

Plug in definition of $p(\mathbf{x}^{(i)}, z^{(i)} | \theta)$ from GMM model

Move log inside product, becomes sum

$$Q(\theta, \theta^{(t-1)}) = \mathbb{E}\left[\sum_{i=1}^{N} \log p(\mathbf{x}^{(i)}, Z^{(i)} \mid \theta)\right]$$

$$= \sum_{i=1}^{N} \mathbb{E}\left[\log\left[\prod_{k=1}^{K} (\pi_k \cdot p(\mathbf{x}^{(i)} \mid \theta_k))^{\mathbb{I}(Z^{(i)}=k)}\right]\right]$$

$$= \sum_{i=1}^{N} \mathbb{E}\left[\sum_{k=1}^{K} \log(\pi_k \cdot p(\mathbf{x}^{(i)} \mid \theta_k))^{\mathbb{I}(Z^{(i)}=k)}\right]$$

EM for GMMs (2)

Log identity:
log a^b = b log a

$$Q(\theta, \theta^{(t-1)}) = \sum_{i=1}^{N} \mathbb{E}\left[\sum_{k=1}^{K} \log(\pi_k \cdot p(\mathbf{x}^{(i)} \mid \theta_k))^{\mathbb{I}(Z^{(i)}=k)}\right]$$

Does not depend on $Z^{(i)}$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{E}\left[\mathbb{I}(Z^{(i)} = k)\right] \boxed{\cdot \log(\pi_k \cdot p(\mathbf{x}^{(i)} \mid \theta_k))}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} p(Z^{(i)} = k \mid \mathbf{x}^{(i)}, \theta^{(t-1)}) \cdot \log(\pi_k \cdot p(\mathbf{x}^{(i)} \mid \theta_k))$$

EM for GMMs (3)

Define the "responsibility" cluster k takes for point $\boldsymbol{x}^{(i)}$

$r_{ik} = \mathrm{p}(Z^{(i)}=k|\mathbf{x}^{(\mathrm{i})},\theta^{(\mathrm{t}-1)})$

$$Q(\theta, \theta^{(t-1)}) = \sum_{i=1}^{N}\sum_{k=1}^{K} \overline{p(Z^{(i)} = k \mid \mathbf{x}^{(i)}, \theta^{(t-1)})} \cdot \log(\pi_k \cdot p(\mathbf{x}^{(i)} \mid \theta_k))$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \cdot (\log \pi_k + \log\ p(\mathbf{x}^{(i)} \mid \theta_k))$$

Log identity:
log a*b = log a + log b

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log\ p(\mathbf{x}^{(i)} \mid \theta_k)$$

With Q in this form, we can compute r_{ik} if θ is fixed, and optimize for θ if r_{ik} is fixed!

$$Q(\theta, \theta^{(t-1)}) = \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log p(\mathbf{x}^{(i)} \mid \theta_k)$$

EM for GMMs - The E step

We can compute the "responsibility" by just normalizing the weighted probability

$$r_{ik} = p(Z^{(i)} = k \mid \mathbf{x}^{(i)}, \theta^{(t-1)})$$

$$= \alpha\, p(\mathbf{x}^{(i)} \mid Z^{(i)} = k, \theta^{(t-1)}) \cdot \underline{p(Z^{(i)} = k \mid \theta^{(t-1)})}$$

$$= \frac{\pi_k \cdot p(\mathbf{x}^{(i)} \mid \theta_k^{(t-1)})}{\sum_{j=1}^{K} \pi_j \cdot p(\mathbf{x}^{(i)} \mid \theta_j^{(t-1)})}$$

Probability under the k^{th} mixture

Bayes rule

Probability of the k^{th} mixture

Normalize!

$$Q(\theta, \theta^{(t-1)}) = \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log p(\mathbf{x}^{(i)} \mid \theta_k)$$

EM for GMMs - The M step (1)

For the mixing coefficients,

$$\text{maximize} \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik} \log \pi_k, \text{ s.t. } \sum_{k=1}^{K} \pi_k = 1$$

$$\Longleftrightarrow$$

$$\pi_k = \frac{1}{N}\sum_{i=1}^{N} r_{ik}$$

Weighted sum of points assigned to cluster

$$Q(\theta, \theta^{(t-1)}) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log \pi_k + \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log p(\mathbf{x}^{(i)} \mid \theta_k)$$

EM for GMMs - The M step (2)

For the gaussian parameters, we use the r_{ik} weighted mean and covariance:

$$\text{maximize} \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log p(\mathbf{x}^{(i)} \mid \theta_k), \text{ s.t. } \Sigma_k \text{ p.s.d.}$$

$$\Longleftrightarrow$$

$$\mu_k = \frac{\sum_{i=1}^{N} r_{ik} \mathbf{x}^{(i)}}{\sum_{i=1}^{N} r_{ik}}, \qquad \Sigma_k = \frac{\sum_{i=1}^{N} r_{ik} (\mathbf{x}^{(i)} - \mu_k)(\mathbf{x}^{(i)} - \mu_k)^\top}{\sum_{i=1}^{N} r_{ik}}$$

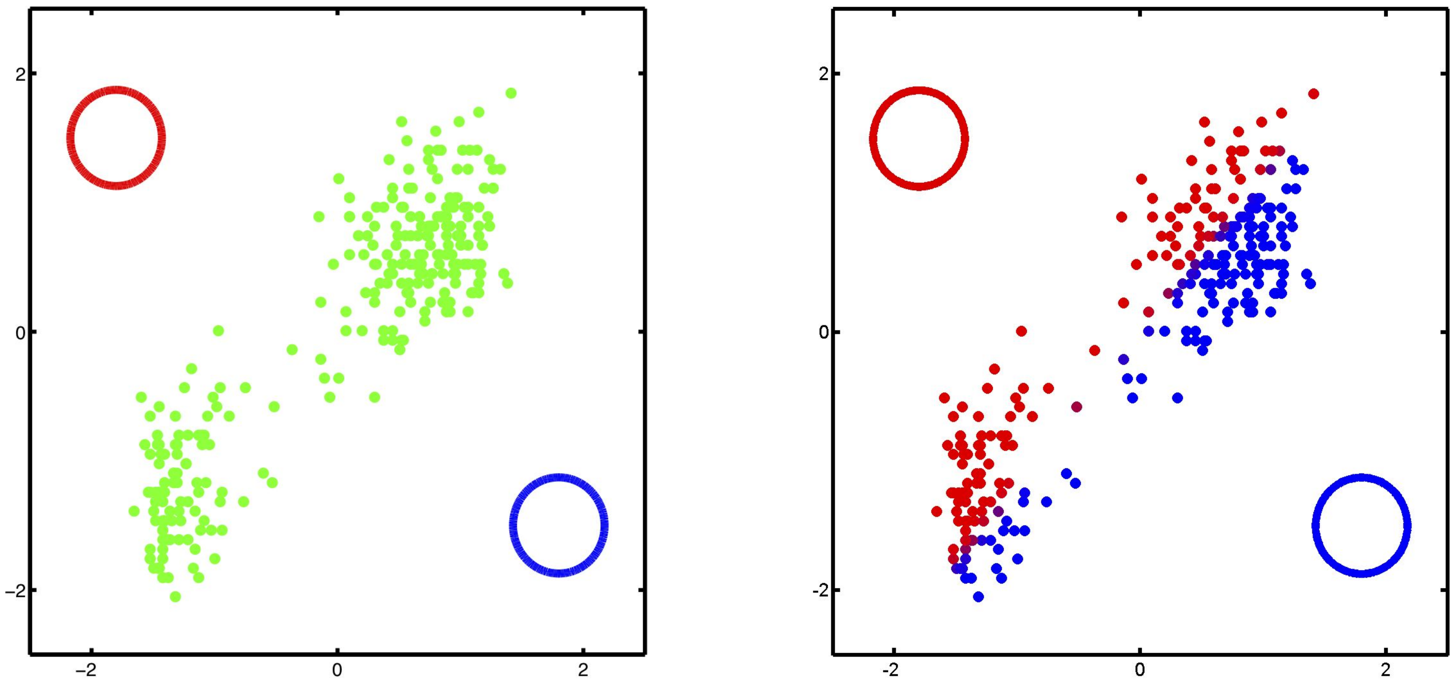
Weighted mean of $x^{(i)}$ assigned to cluster

Weighted empirical covariance

EM for GMMs - example (1)

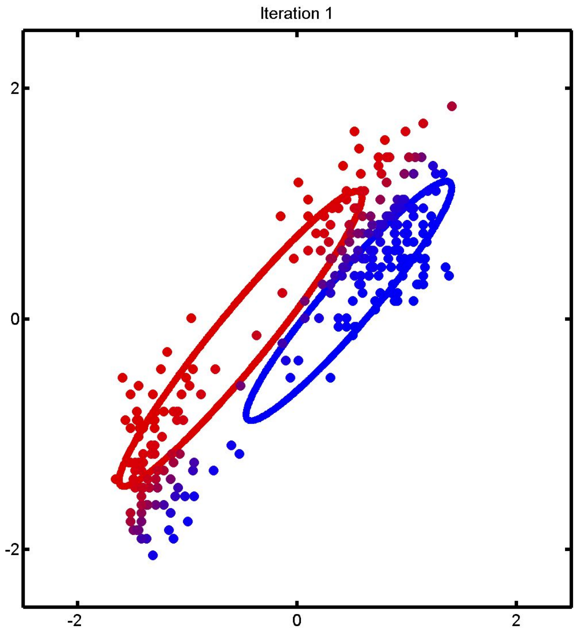
Init random θ

First E step

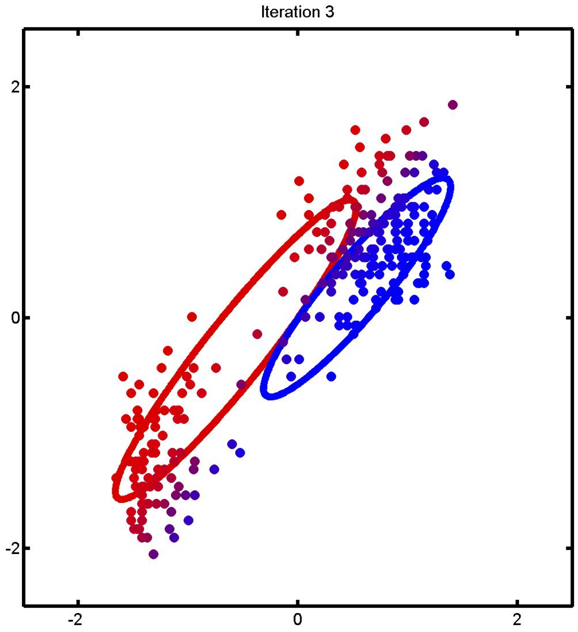


EM for GMMs - example (2)

After first M step

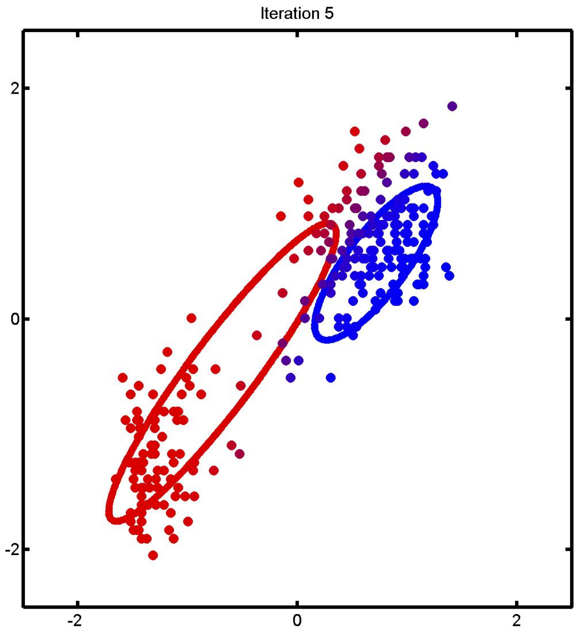


After 3 iterations

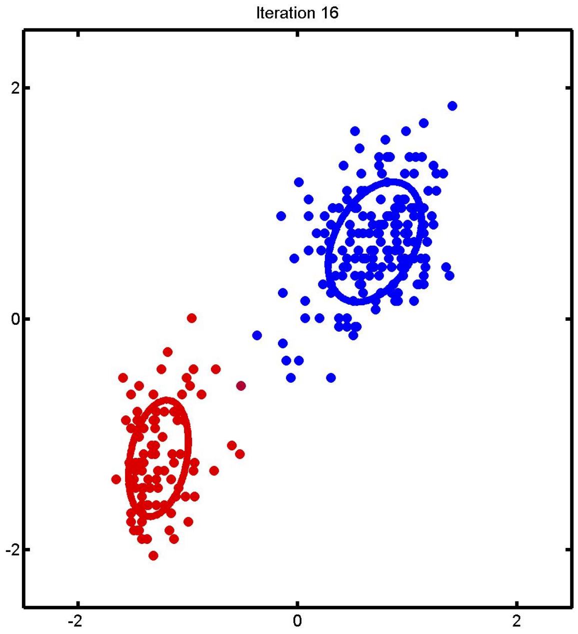


EM for GMMs - example (3)

After 5 iterations



After 16 iterations



EM properties

- Each iteration monotonically improves the likelihood of the data
- Like k-Means finds a **local** optima (Note that swapping cluster labels doesn't change likelihood, so this problem is non-convex)
- Unlike k-Means, no fixed number of iterations (soft assignment means there isn't a finite number of configurations)
- Works on many different problems as long as you can define both the E step and the M step

Summary and preview

Wrapping up

- Gaussian Mixture Models let us perform “soft clustering” where instead of a partition function, we can assign a **probability** of belonging to any of the clusters
- We can fit the parameters of a GMM using a technique known as **E**xpectation **M**aximization (EM): alternating between finding the expected value of the complete data likelihood, and finding the parameters which maximize this expectation

Next time: Hidden Markov Models