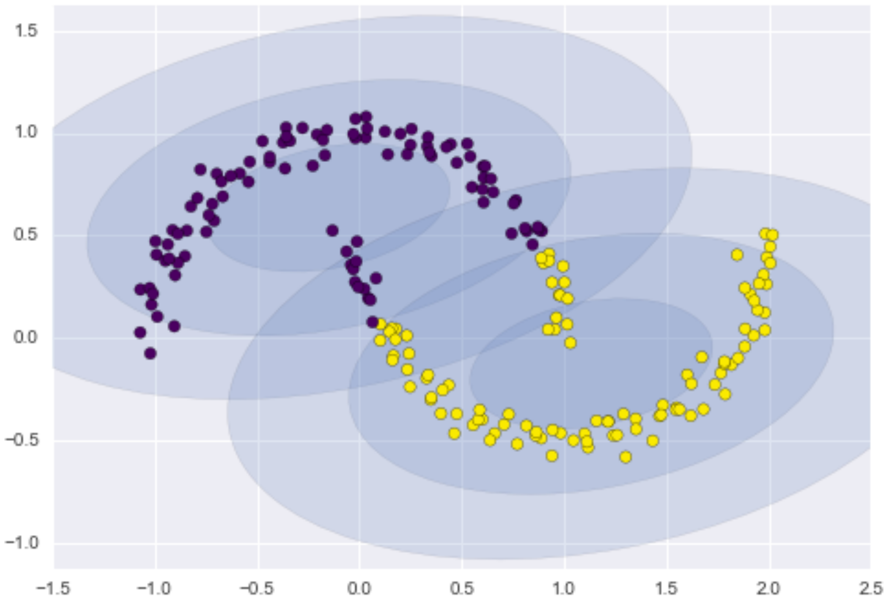
Gaussian Mixture Models (GMM)

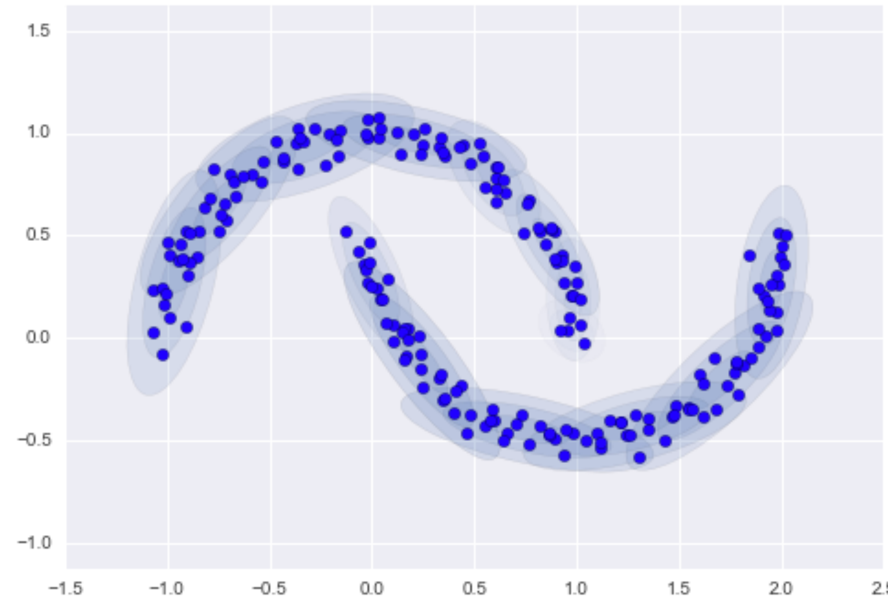
ECE57000: Artificial Intelligence, Fall 2019

David I. Inouye

<u>Gaussian mixture models</u> (GMM) can be used for (1) density estimation and (2) flexible clustering

1. General density estimation

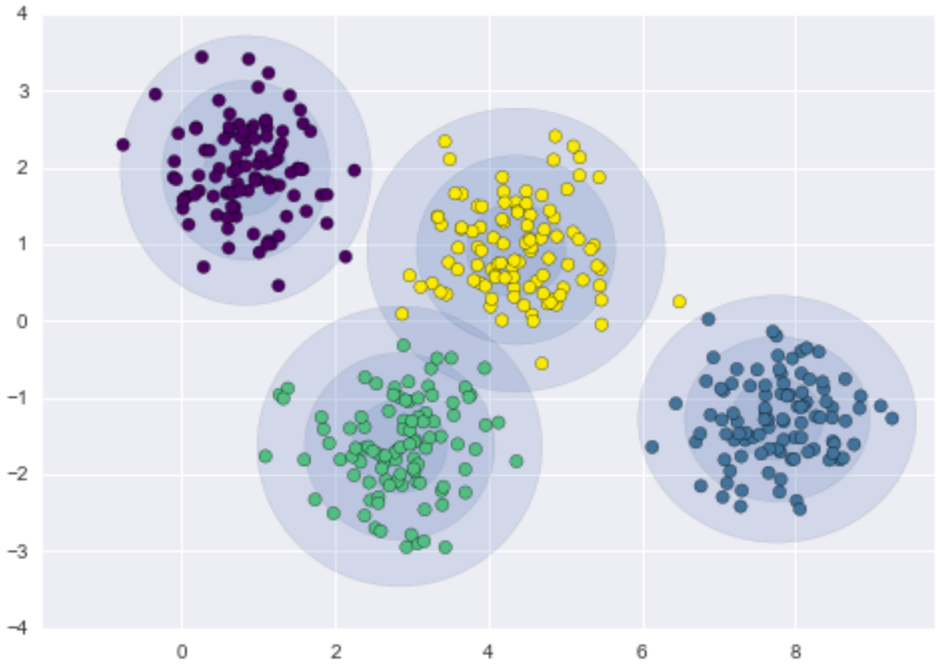




https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

Even if each component distribution is independent, the mixture may **not** be independent

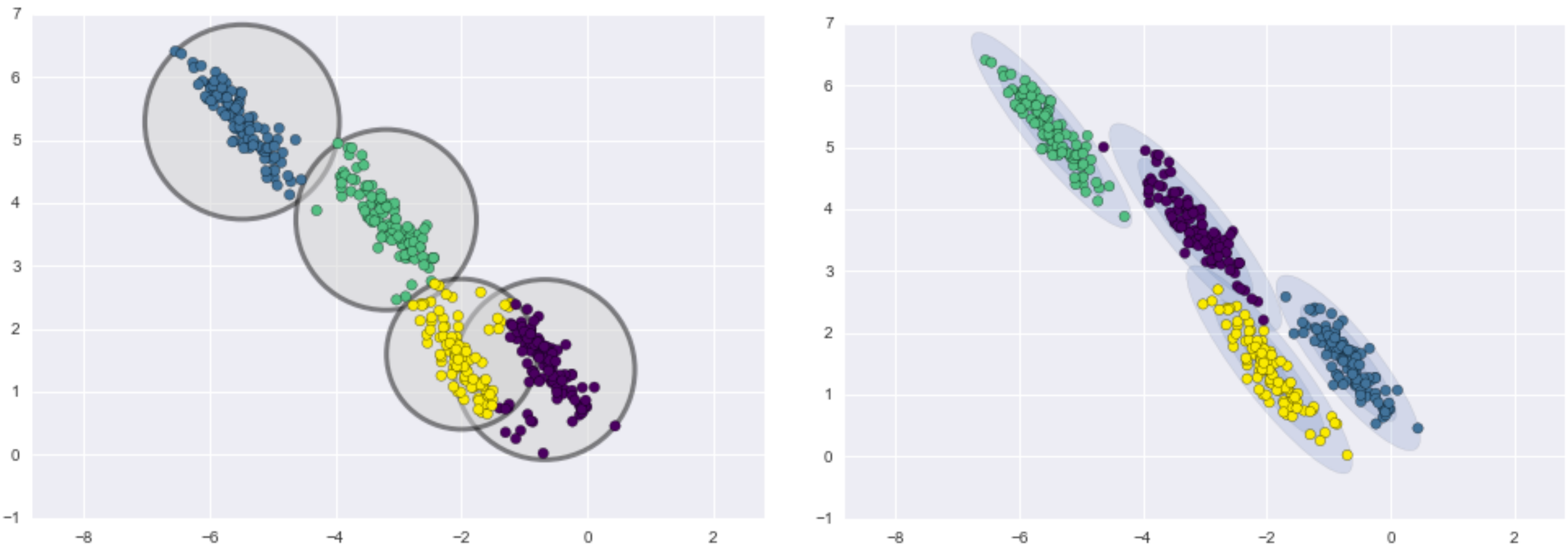
- Formally, $p_j\left(x; \mu_j, \Sigma = \sigma_j^2 I\right), \forall j \in \{1, \dots, k\}$



https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

<u>Gaussian mixture models</u> (GMM) can be used for (1) density estimation and (2) flexible clustering

2. Flexible clustering



https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

Mixture distributions are weighted averages of component distributions

- Mixture distribution
  - Component weights $0 \leq \pi_j, \leq 1$ s.t. $\sum_{j=1}^{k} \pi_j = 1$
  - Component distributions $p_j(x)$
- Simple form of mixture

$$p_{\mathrm{mixture}}(x) = \sum_{j=1}^{k} \pi_j p_j(x)$$

- (check that integrates to 1)

Mixture models can be viewed as <u>latent (or "hidden") variable models</u>

- Simple form of mixture
$$p_{\text{mixture}}(x) = \sum_{j=1}^{k} \pi_j p_j(x)$$
- Note that π_j form a discrete distribution
- Let $z \in \{1, \dots, k\}$ be an *auxiliary* **<u>indicator variable</u>** that denotes which component the point is from
- Let $p(z = j) = \pi_j$, then the joint density model is:
$$p(x, z) = p(z)p(x|z)$$
- Because z are unobserved, we need the marginal distribution of x

$$p_{\text{mixture}}(x) = \sum_j p(x, z = j) = \sum_j p(z = j)p(x|z = j)$$

Gaussian mixture models (GMM) are one of the most common mixture distributions

- Form of Gaussian mixture model

$$p_{\mathrm{GMM}}(x) = \sum_{j=1}^{k} \pi_j p_{\mathcal{N}}(x; \mu_j, \Sigma_j) = \sum_{j=1}^{k} p(z = j) p_{\mathcal{N}}(x; z = j)$$

Machine Learning, Murphy, 2012.

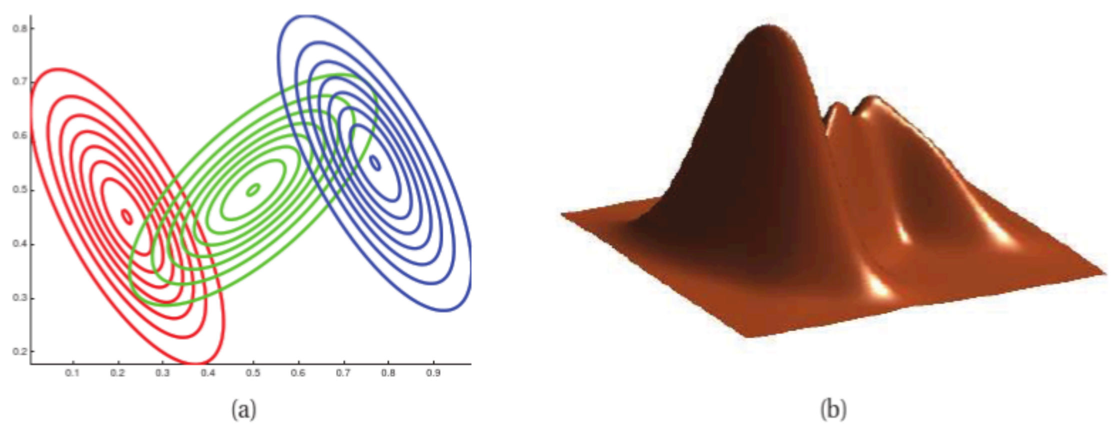


Figure 11.3 A mixture of 3 Gaussians in 2d. (a) We show the contours of constant probability for each component in the mixture. (b) A surface plot of the overall density. Based on Figure 2.23 of (Bishop 2006a). Figure generated by `mixGaussPlotDemo`.

MLE for mixtures is difficult
Reason 1: The algebraic form is more complex

- The mixture log likelihood cannot be simplified

$$\arg\max_{\pi,\mu_j,\Sigma_j} \log \prod_i p_{\mathrm{GMM}}(x_i; \mu_1, \dots, \mu_k, \Sigma_1, \dots, \Sigma_k)$$

$$\sum_i \log p_{\mathrm{GMM}}(x_i; \mu_1, \dots, \mu_k, \Sigma_1, \dots, \Sigma_k)$$

$$\sum_i \log \sum_j p(z_i = j) p_{\mathcal{N}}(x_i; z_i = j)$$

$$\sum_i \log \sum_j \pi_j \exp\left\{-\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j) - \frac{1}{2}\log|\Sigma_j|\right\}$$

- **Cannot exchange log and summation to cancel exp**

MLE for mixtures is difficult
Reason 2: Problem is non-convex
(and could have multiple local optima)

- The intuition is similar to the problem with k-means clustering

See [ML, Ch. 11, pp. 347-348] for more detailed analysis.

Observation: If we knew z_i, then optimizing the complete log likelihood is easy

- Observed/marginal log likelihood
(if z_i is **unknown**)

$$\log \prod_i \left(\sum_j p(z_i) p(x_i | z_i) \right)$$

- Complete log likelihood (if z_i is **known**)

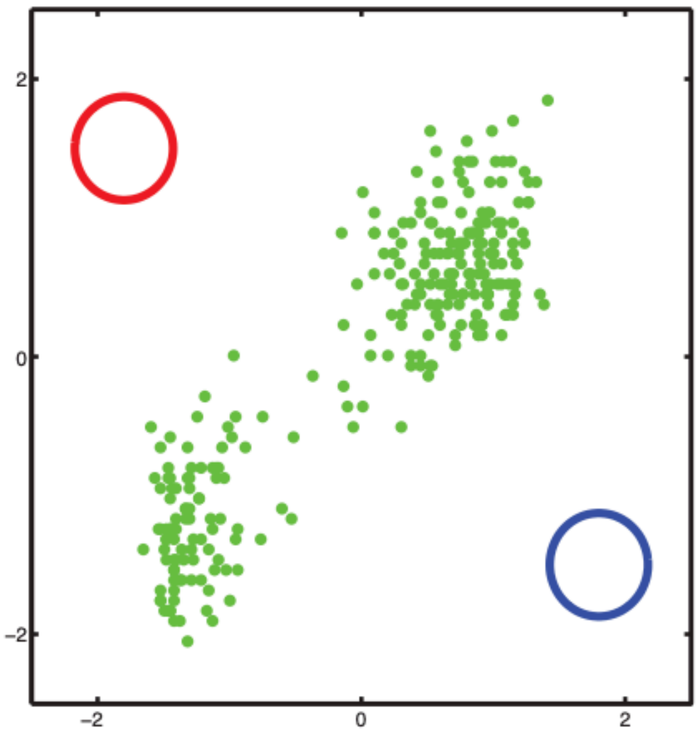
$$\log \prod_i p(x_i, z_i; \theta) = \log \prod_i \pi_j p_{\mathcal{N}}\left(x_i; \mu_{z_i}, \Sigma_{z_i}\right)$$

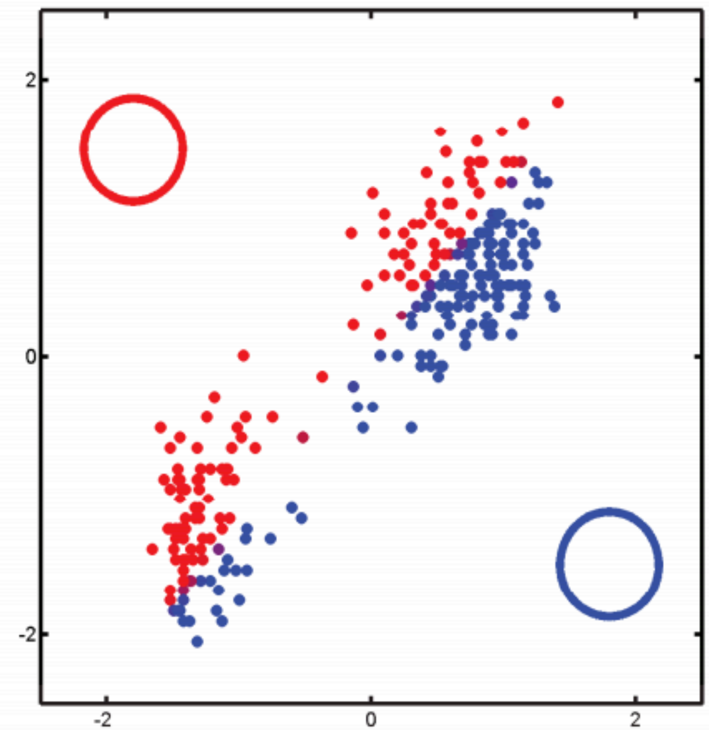
  - For GMMs, this is convex and easy to solve

The Expectation-Maximization (EM) algorithm can be seen as a generalization of k-means

- The EM algorithm for GMM alternates between
  - Probabilistic/soft assignment of points
  - Estimation of Gaussian for each component
- Similar to k-means which alternates between
  - Hard assignment of points
  - Estimation of mean of points in each cluster

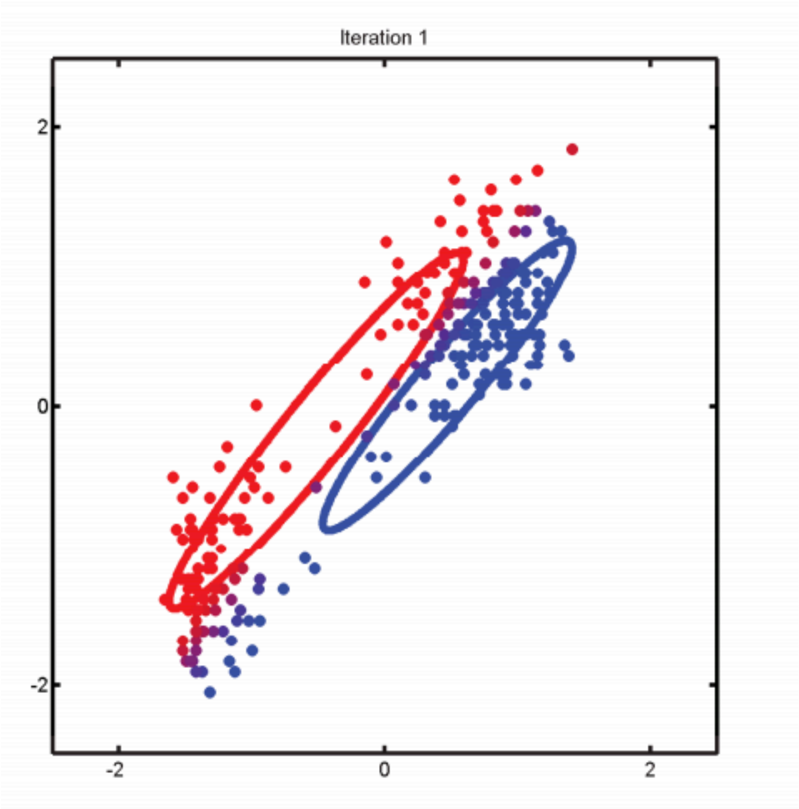
EM Algorithm: Initialization

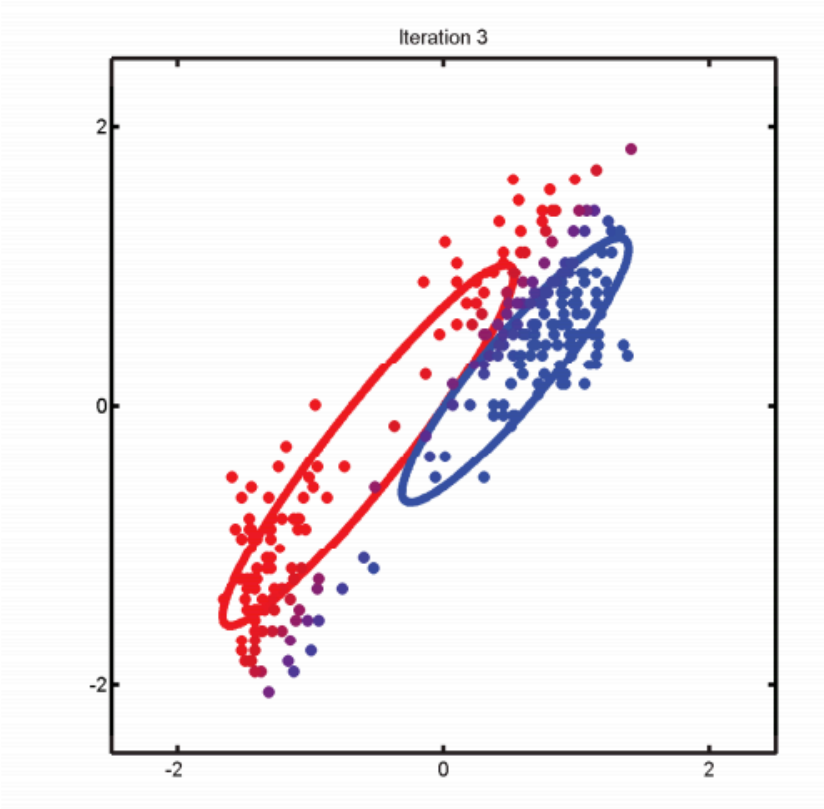




Machine Learning: A probabilistic perspective, Murphy, 2012.

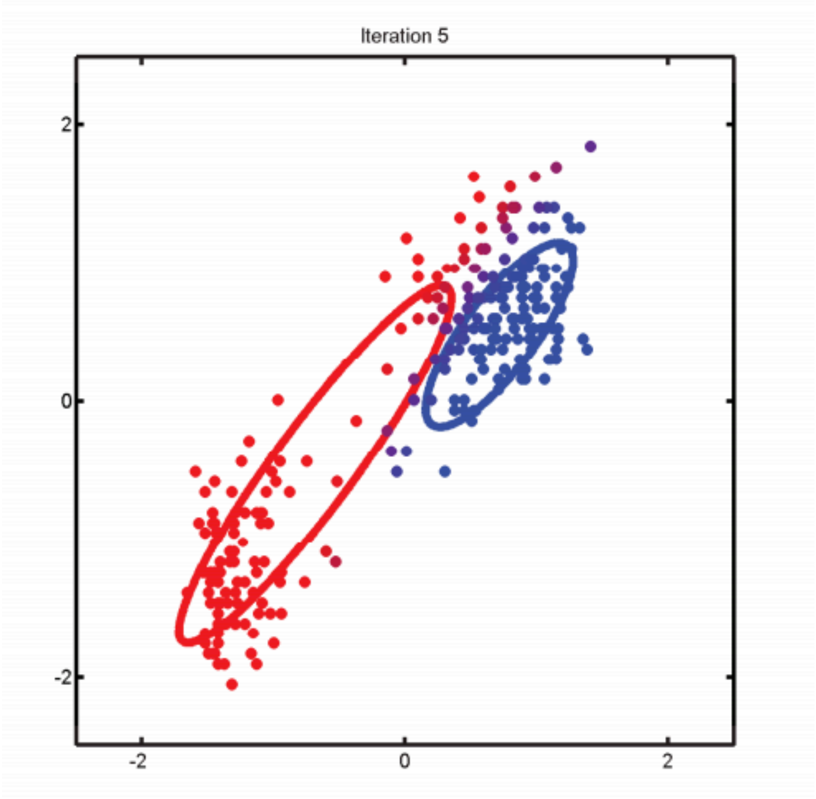
EM Algorithm: Iteration 1 and 3

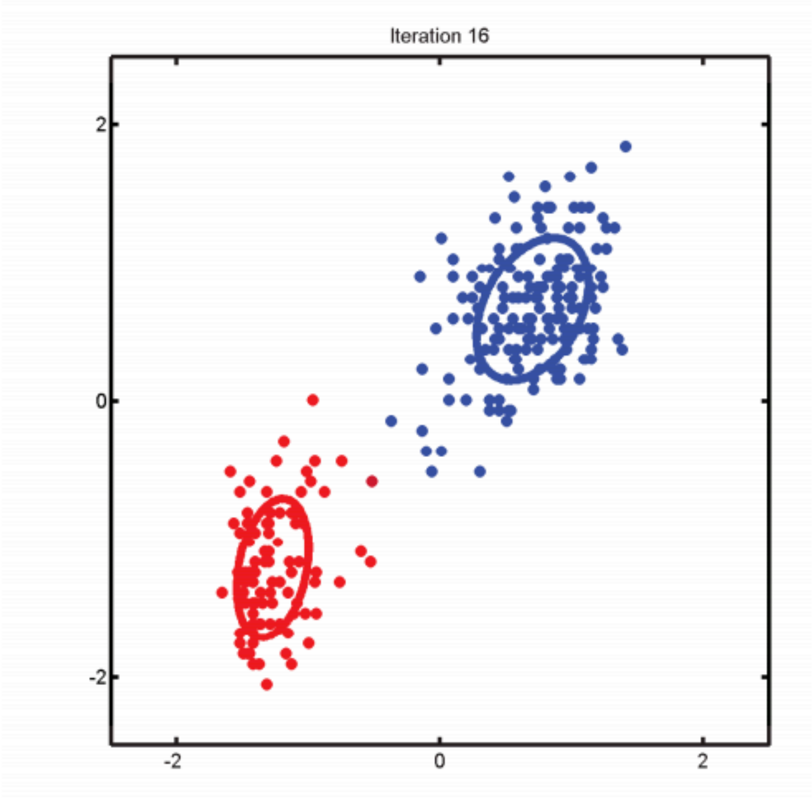




Machine Learning: A probabilistic perspective, Murphy, 2012.

EM Algorithm: Iteration 5 and 16





Machine Learning: A probabilistic perspective, Murphy, 2012.

EM algorithm is **guaranteed** to increase *observed* likelihood, i.e., $\prod_i p_{mixture}(x_i)$