Unsupervised Learning:
Gaussian Mixture Models
& Expectation Maximization

These slides were assembled by Byron Boots, with only minor modifications from Eric Eaton's slides and grateful acknowledgement to the many others who made their course materials freely available online. Feel free to reuse or adapt these slides for your own academic purposes, provided that you include proper attribution.

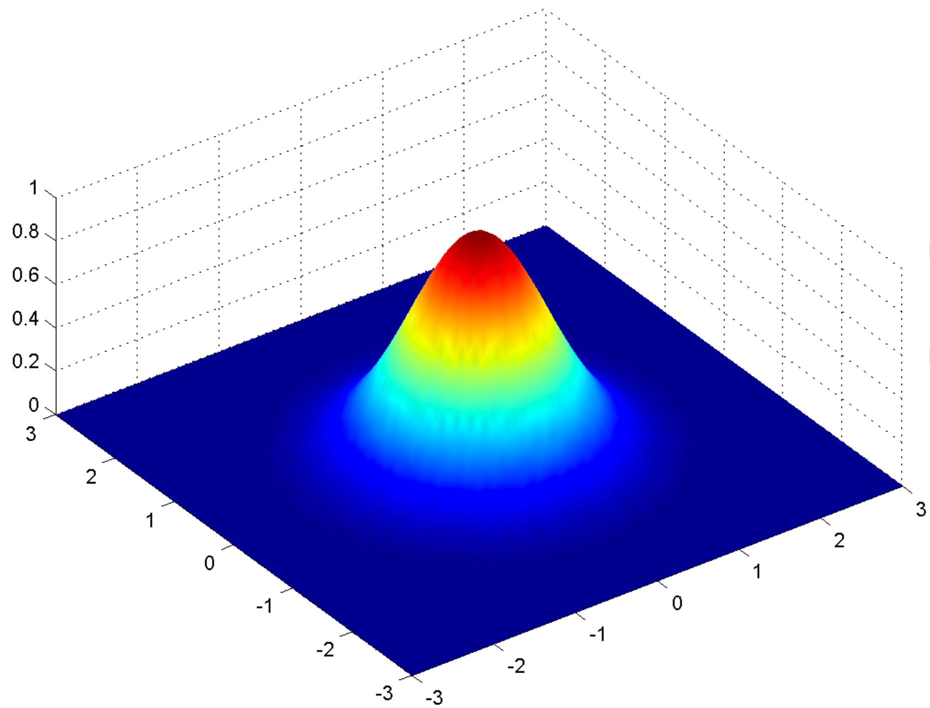
Soft Clustering

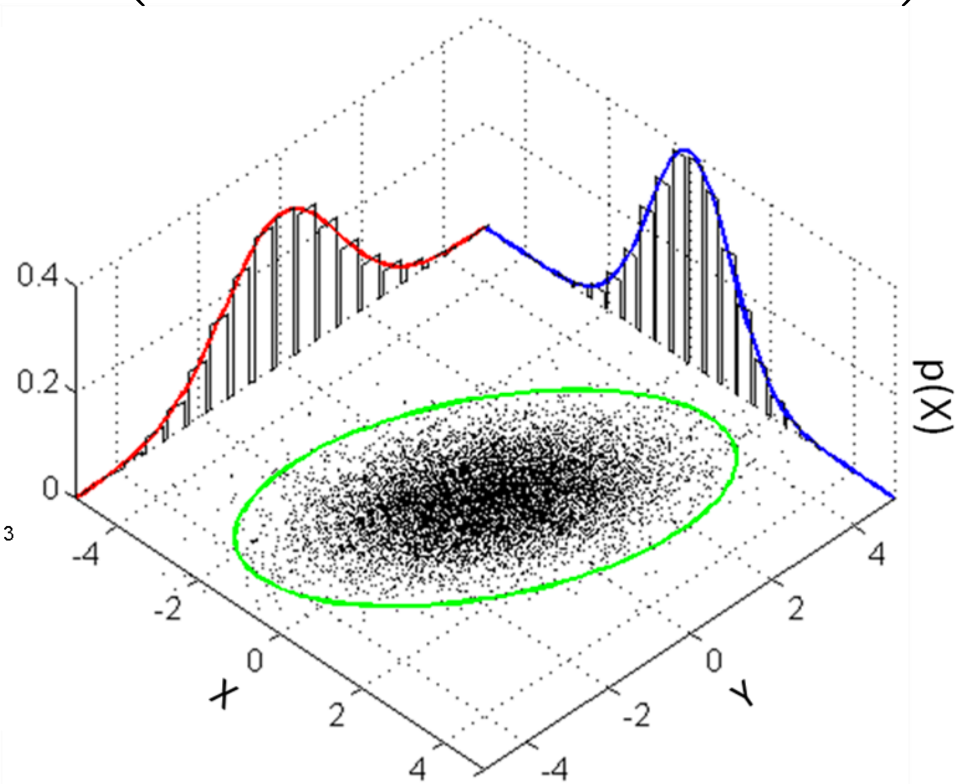
- Clustering typically assumes that each instance is given a "hard" assignment to exactly one cluster.
- Does not allow uncertainty in class membership or for an instance to belong to more than one cluster.
- *Soft clustering* gives probabilities that an instance belongs to each of a set of clusters.
- Each instance is assigned a probability distribution across a set of discovered categories (probabilities of all categories must sum to 1).

Gaussian Mixture Models

- Recall the Gaussian distribution:

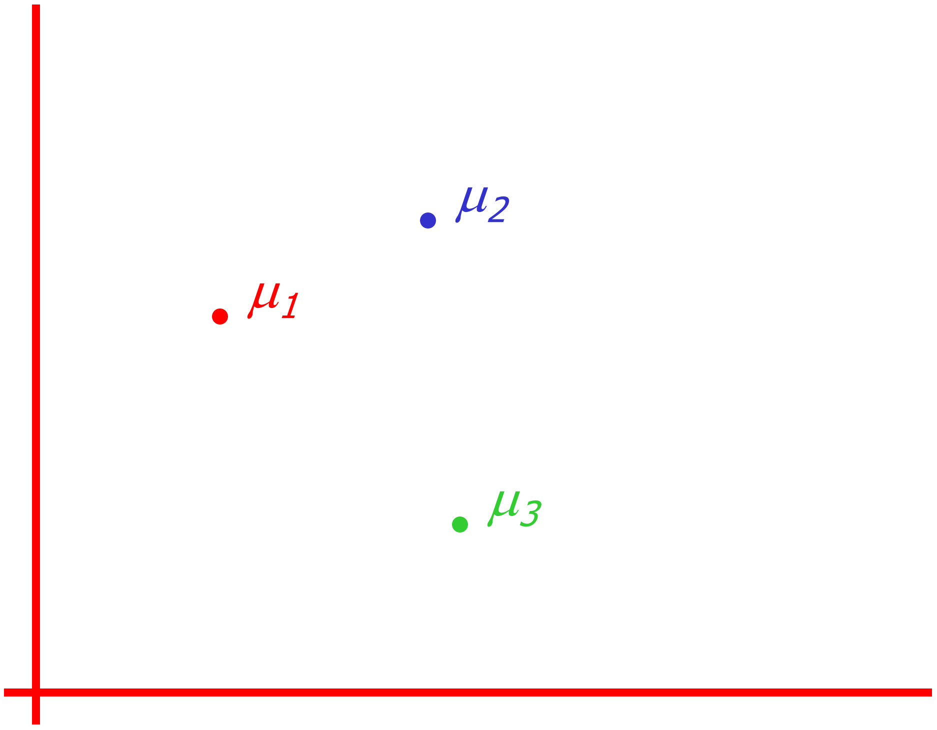
$$P(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$





The GMM assumption

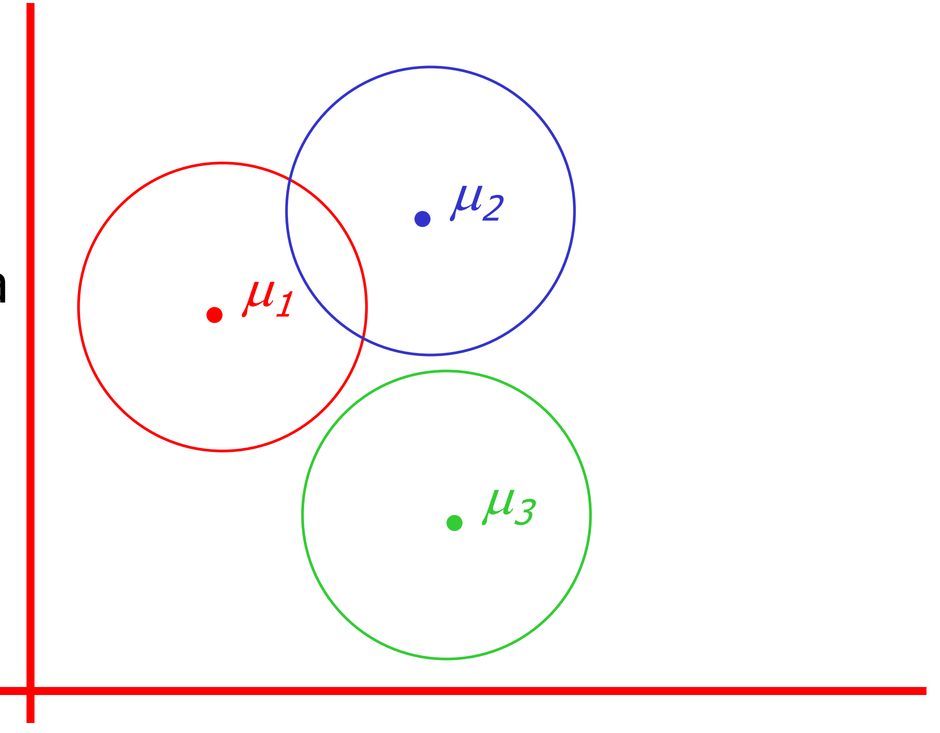
- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i



The GMM assumption

- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$

Assume that each datapoint is generated according to the following recipe:

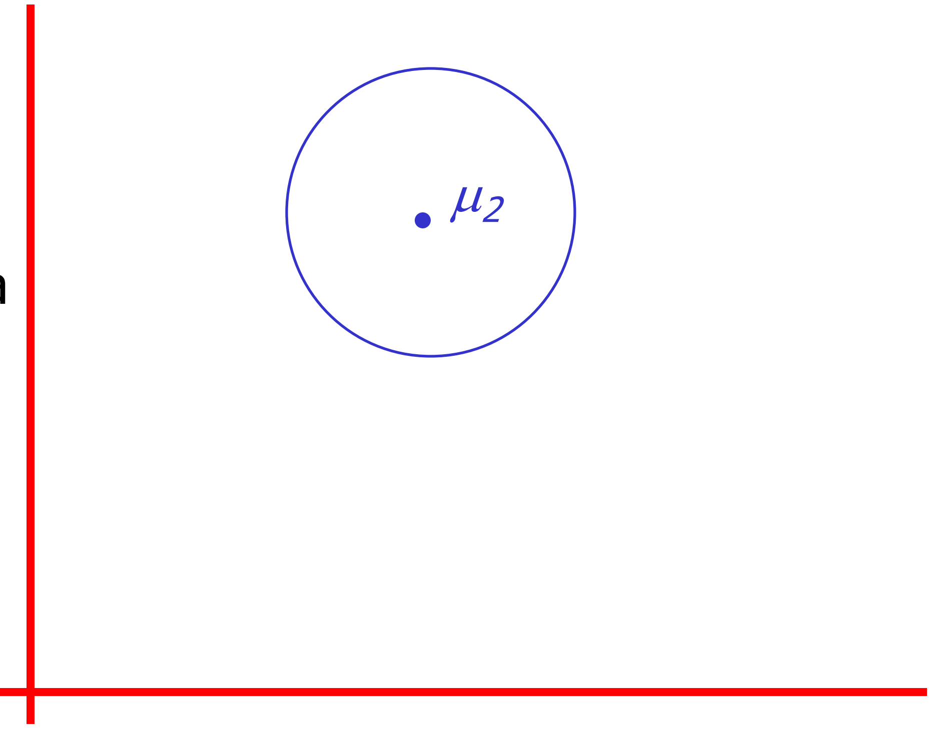


The GMM assumption

- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random: choose component i with probability $P(\omega_i)$.

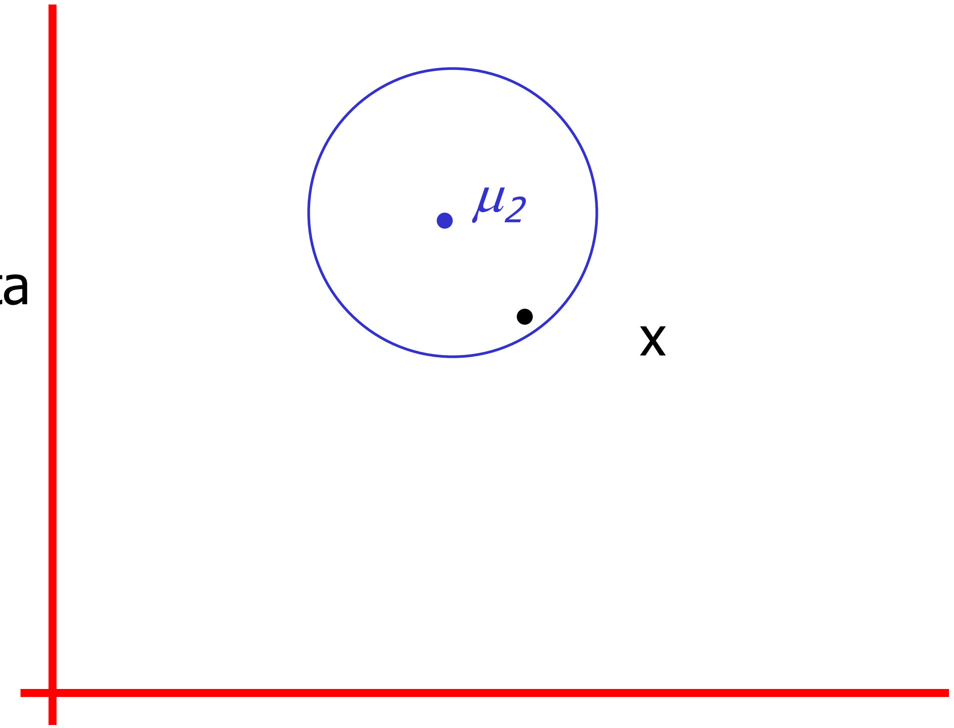


The GMM assumption

- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \mathbf{I}$

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random: choose component *i* with probability $P(\omega_i)$.
2. Datapoint ~ $N(\mu_i, \sigma^2 \mathbf{I})$

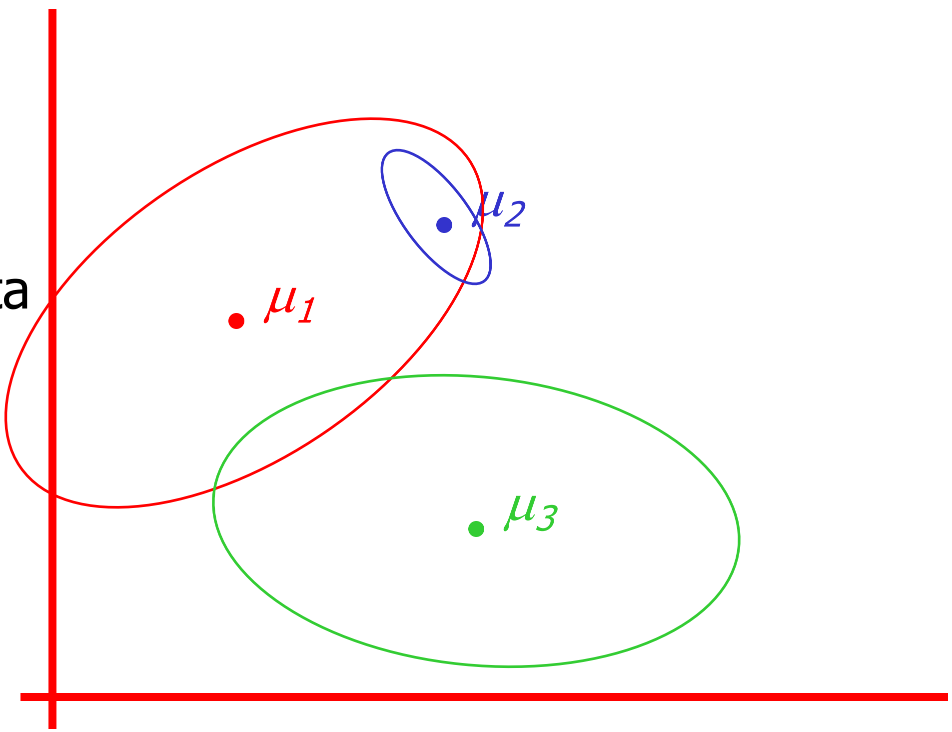


The General GMM assumption

- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix Σ_i

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random: choose component i with probability $P(\omega_i)$.
2. Datapoint ~ N(μ_i, Σ_i)



Mixture Models

- Formally a Mixture Model is the weighted sum of a number of pdfs where the weights are determined by a distribution π

$$p(x) = \pi_0 f_0(x) + \pi_1 f_1(x) + \pi_2 f_2(x) + \ldots + \pi_k f_k(x)$$

$$\text{where } \sum_{i=0}^{k} \pi_i = 1$$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i f_i(x)}$$

Gaussian Mixture Models

- GMM: the weighted sum of a number of Gaussians where the weights are determined by a distribution π

$$p(x) = \pi_0 N(x|\mu_0, \Sigma_0) + \pi_1 N(x|\mu_1, \Sigma_1) + \ldots + \pi_k N(x|\mu_k, \Sigma_k)$$

$$\text{where } \sum_{i=0}^{k} \pi_i = 1$$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i N(x|\mu_k, \Sigma_k)}$$

Expectation-Maximization for GMMs

Iterate until convergence:

On the *t* th iteration let our estimates be

$$\lambda_t = \{ \mu_1(t), \mu_2(t) \ldots \mu_c(t) \}$$

Just evaluate a Gaussian at x_k

E-step: Compute "expected" classes of all datapoints for each class

$$\mathrm{P}(w_i | x_k, \lambda_t) = \frac{\mathrm{p}(x_k | w_i, \lambda_t)\mathrm{P}(w_i | \lambda_t)}{\mathrm{p}(x_k | \lambda_t)} = \frac{\mathrm{p}(x_k | w_i, \mu_i(t), \sigma^2 \mathbf{I}) p_i(t)}{\sum_{j=1}^{c} \mathrm{p}(x_k | w_j, \mu_j(t), \sigma^2 \mathbf{I}) p_j(t)}$$

M-step: Estimate **μ** given our data's class membership distributions

$$\mu_i(t+1) = \frac{\sum_k \mathrm{P}(w_i | x_k, \lambda_t) x_k}{\sum_k \mathrm{P}(w_i | x_k, \lambda_t)}$$

E.M. for General GMMs

$p_i(t)$ is shorthand for estimate of $P(\omega_i)$ on t' th iteration

Iterate. On the t'th iteration let our estimates be

$$\lambda_t = \{ \mu_1(t), \mu_2(t) \ldots \mu_c(t), \Sigma_1(t), \Sigma_2(t) \ldots \Sigma_c(t), p_1(t), p_2(t) \ldots p_c(t) \}$$

E-step: Compute "expected" clusters of all datapoints

$$P(w_i \mid x_k, \lambda_t) = \frac{p(x_k \mid w_i, \lambda_t) P(w_i \mid \lambda_t)}{p(x_k \mid \lambda_t)} = \frac{p(x_k \mid w_i, \mu_i(t), \Sigma_i(t)) p_i(t)}{\sum_{j=1}^{c} p(x_k \mid w_j, \mu_j(t), \Sigma_j(t)) p_j(t)}$$

Just evaluate a Gaussian at x_k

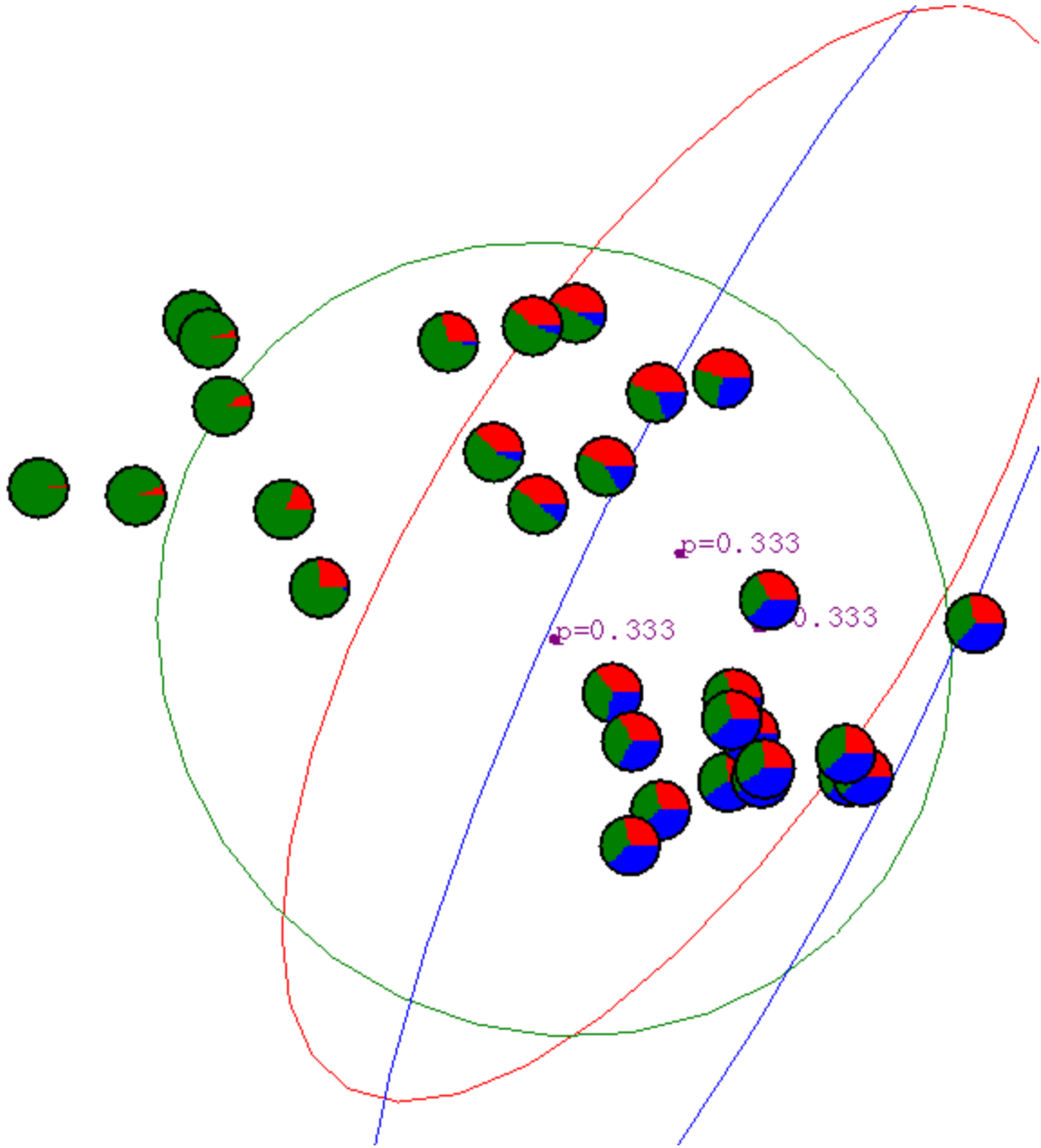
M-step: Estimate **μ, Σ** given our data's class membership distributions

$$\mu_i(t+1) = \frac{\sum_k P(w_i \mid x_k, \lambda_t) x_k}{\sum_k P(w_i \mid x_k, \lambda_t)} \qquad \Sigma_i(t+1) = \frac{\sum_k P(w_i \mid x_k, \lambda_t) [x_k - \mu_i(t+1)][x_k - \mu_i(t+1)]^T}{\sum_k P(w_i \mid x_k, \lambda_t)}$$

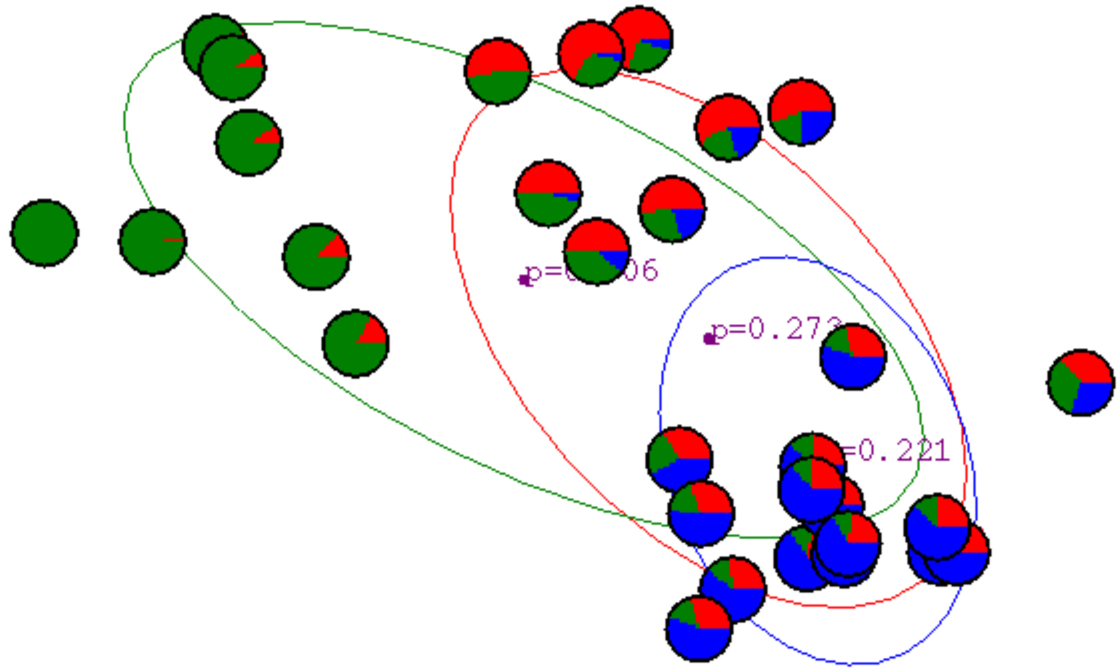
$$p_i(t+1) = \frac{\sum_k P(w_i \mid x_k, \lambda_t)}{R}$$

R = #records

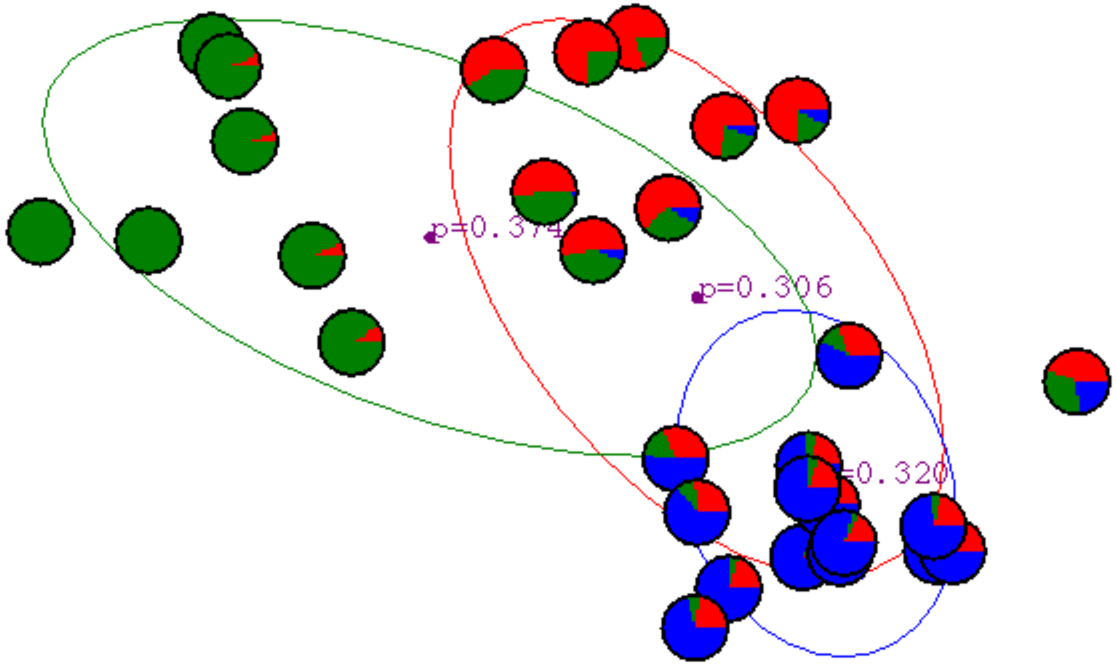
Gaussian Mixture Example: Start



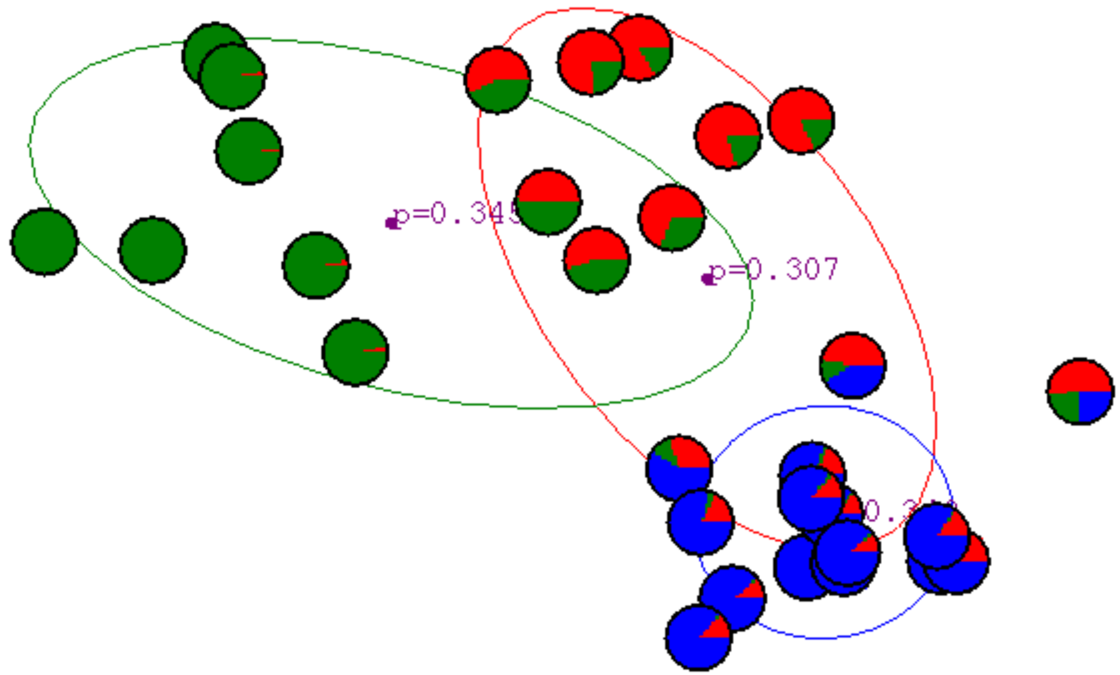
After first iteration



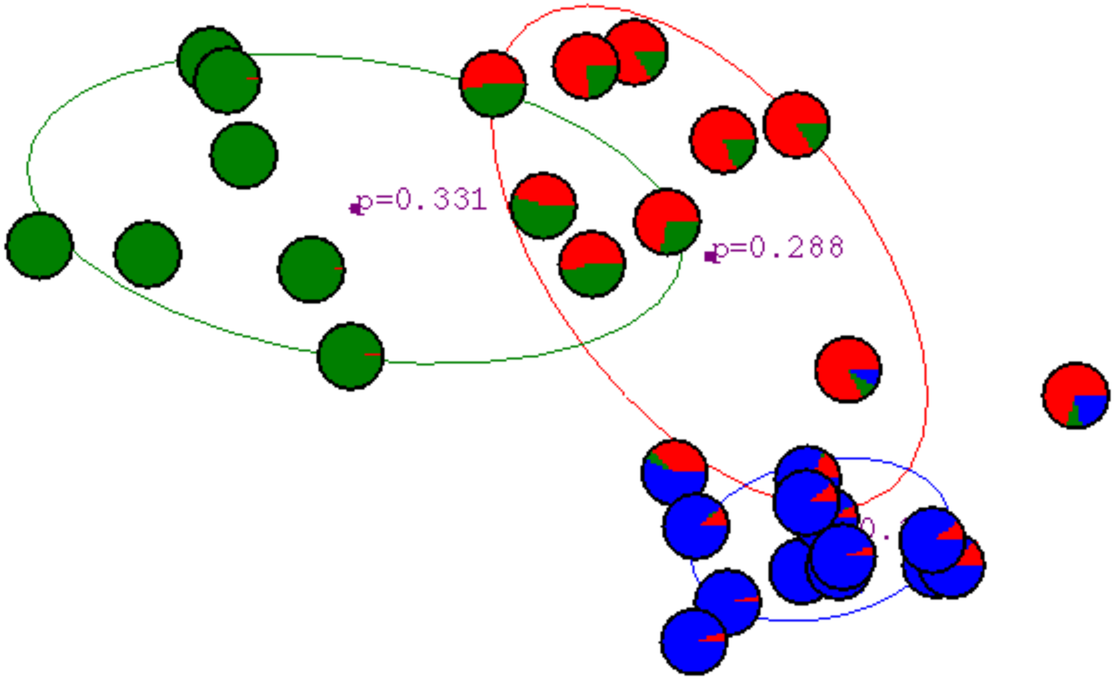
After 2nd iteration



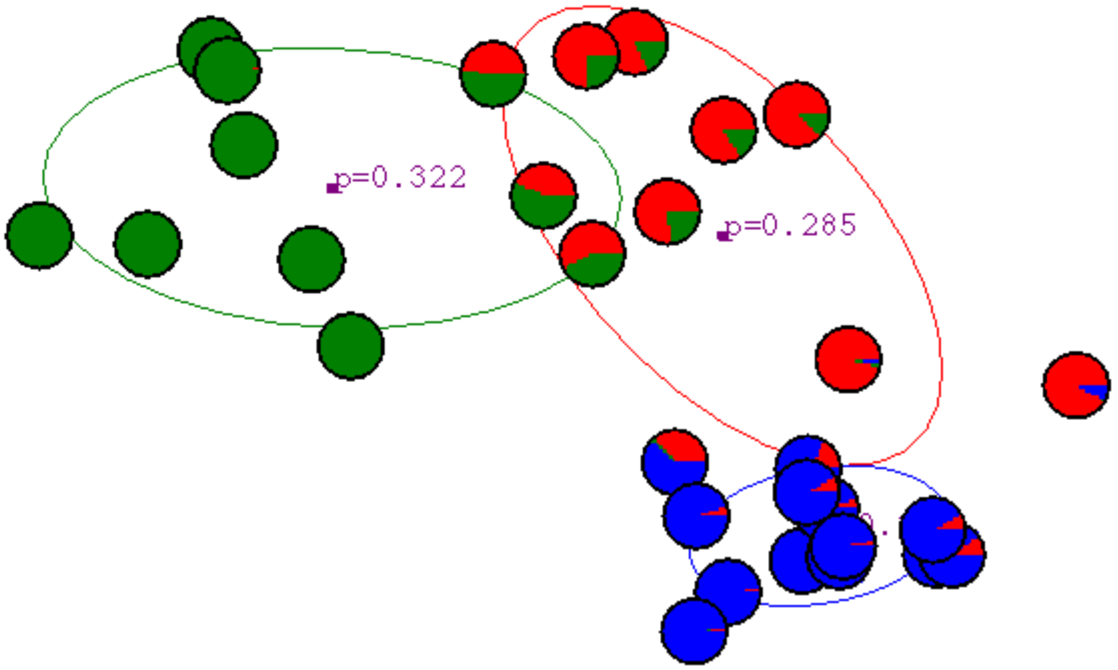
After 3rd iteration



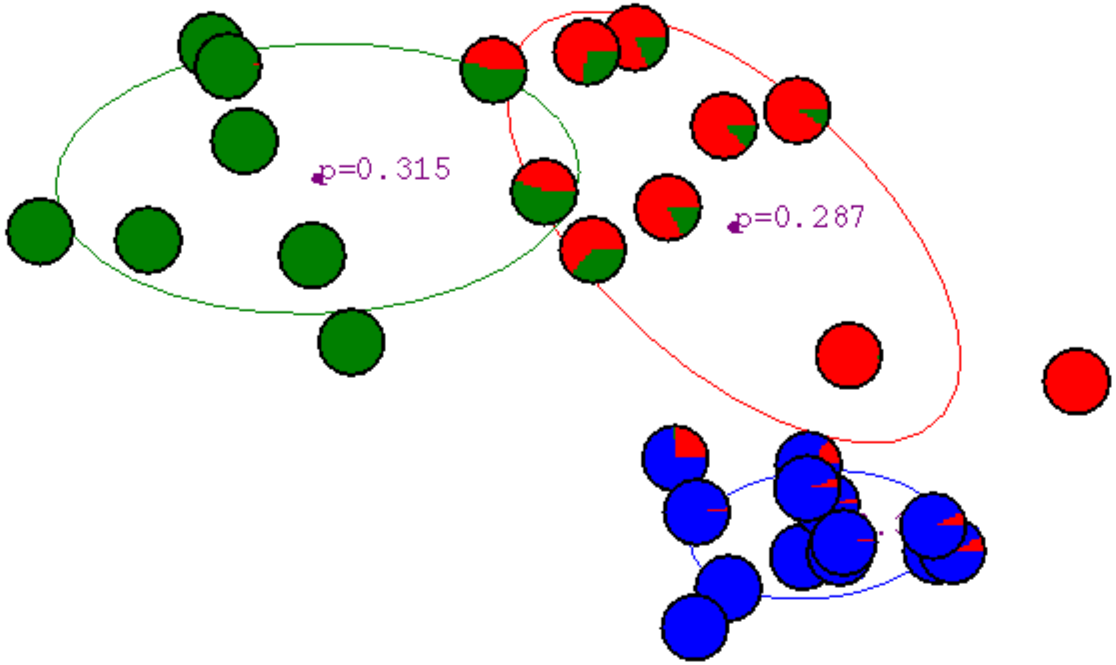
After 4th iteration



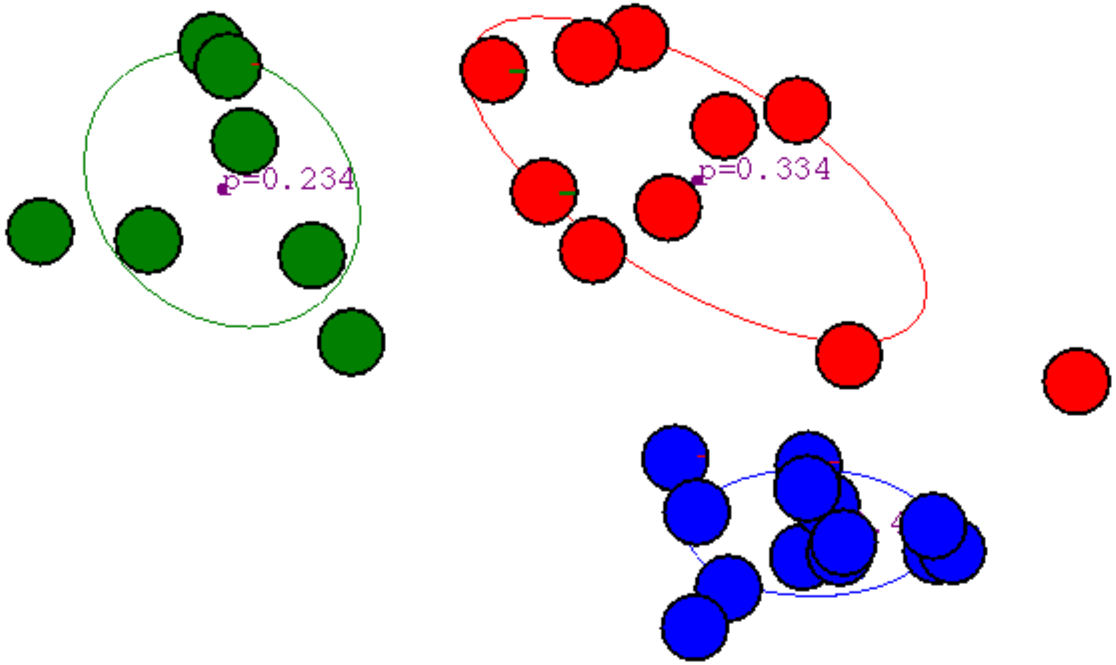
After 5th iteration



After 6th iteration



After 20th iteration



Closing Thoughts

- GMMs are a ”soft” clustering algorithm, that can be learned using EM.
- If you keep iterating EM, you will converge, but only to a local optimum.
- You will see EM in other contexts as well, when doing inference with graphical models is hard – like Hidden Markov Models