# Gaussian Mixture Model

# Density Estimation

## Generative approach

$$p(x_1, \ldots, x_n|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$$

- There is a latent parameter $\Theta$
- For all i, draw observed $x_i$ given $\Theta$

**What if the basic model doesn't fit all data?**

$\Rightarrow$ Mixture modelling, Partitioning algorithms

Different parameters for different parts of the domain. $[\theta_1, \ldots, \theta_K]$

# Partitioning Algorithms

- **K-means**
  - **hard assignment**: each object belongs to only one cluster

$$\theta_i \in \{\theta_1, \ldots, \theta_K\}$$

- **Mixture modeling**
  - **soft assignment**: probability that an object belongs to a cluster

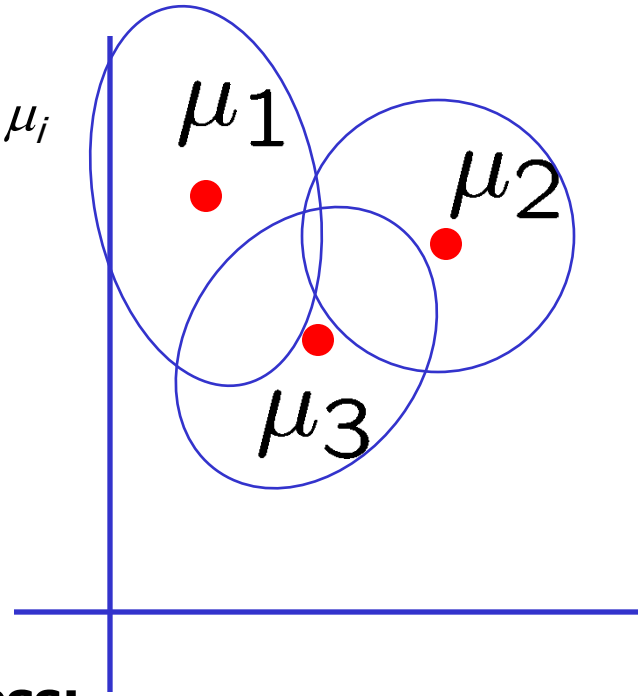
$$(\pi_1, \ldots, \pi_K),\ \pi_i \geq 0,\ \sum_{i=1}^{K} \pi_i = 1$$

# Gaussian Mixture Model

**Mixture of K Gaussians distributions: (Multi-modal distribution)**

- There are K components
- Component *i* has an associated mean vector $\mu_i$

Component *i* generates data from $N(\mu_i, \Sigma_i)$




**Each data point is generated using this process:**

1) Choose component $i$ with probability $\pi_i = P(y = i)$
2) Datapoint $x \sim N(\mu_i, \Sigma_i)$

# Gaussian Mixture Model

**Mixture of K Gaussians distributions: (Multi-modal distribution)**

**Hidden variable**

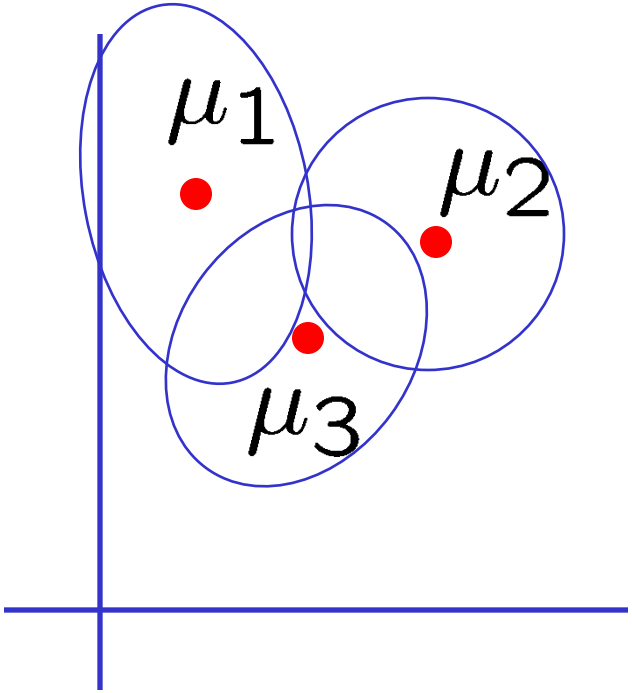
$$p(x|y=i) = N(\mu_i, \Sigma_i)$$

$$p(x) = \sum_{i=1}^{K} p(x|y=i)P(y=i)$$

**Observed data** — $p(x)$

**Mixture component** — $p(x|y=i)$

**Mixture proportion** — $P(y=i)$



# Mixture of Gaussians Clustering

**Assume that**

$\Sigma_i = \sigma^2 \mathbf{I}$, for simplicity.

$p(x|y=i) = N(\mu_i, \sigma^2 \mathbf{I})$

$p(y=i) = \pi_i$

$\mu_1, \ldots, \mu_K, \sigma^2, \pi_1, \ldots, \pi_K$ parameters are all known.

For a given x we want to decide if it belongs to cluster i or cluster j

**Cluster x based on posteriors**:

$$\log \frac{P(y=i|x)}{P(y=j|x)}$$

$$= \log \frac{p(x|y=i)P(y=i)/p(x)}{p(x|y=j)P(y=j)/p(x)}$$

$$= \log \frac{p(x|y=i)\pi_i}{p(x|y=j)\pi_j} = \log \frac{\pi_i \exp(\frac{-1}{2\sigma^2}\|x-\mu_i\|^2)}{\pi_j \exp(\frac{-1}{2\sigma^2}\|x-\mu_j\|^2)}$$

# Mixture of Gaussians Clustering

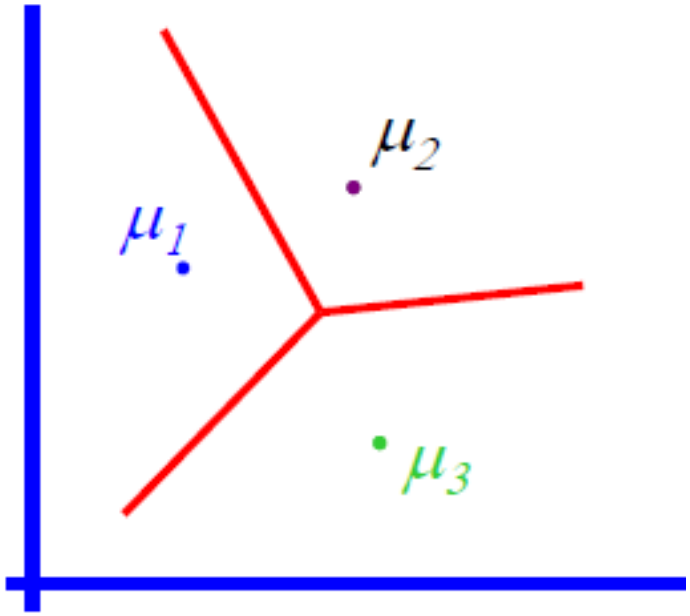
**Assume that**

$\Sigma_i = \sigma^2 \mathbf{I}$, for simplicity. $p(x|y=i) = N(\mu_i, \sigma^2 \mathbf{I})$

$p(y=i) = \pi_i$ $\mu_1, \ldots, \mu_K, \sigma^2, \pi_1, \ldots, \pi_K$ are known.

$$\log \frac{P(y=i|x)}{P(y=j|x)} = \log \frac{p(x|y=i)\pi_i}{p(x|y=j)\pi_j} = \log \frac{\pi_i \exp(\frac{-1}{2\sigma^2}\|x-\mu_i\|^2)}{\pi_j \exp(\frac{-1}{2\sigma^2}\|x-\mu_j\|^2)}$$

# Piecewise linear decision boundary



# MLE for GMM

**What if we don't know the parameters?** $\mu_1, \ldots, \mu_K, \sigma^2, \pi_1, \ldots, \pi_K$?

$\Rightarrow$ **Maximum Likelihood Estimate (MLE)**

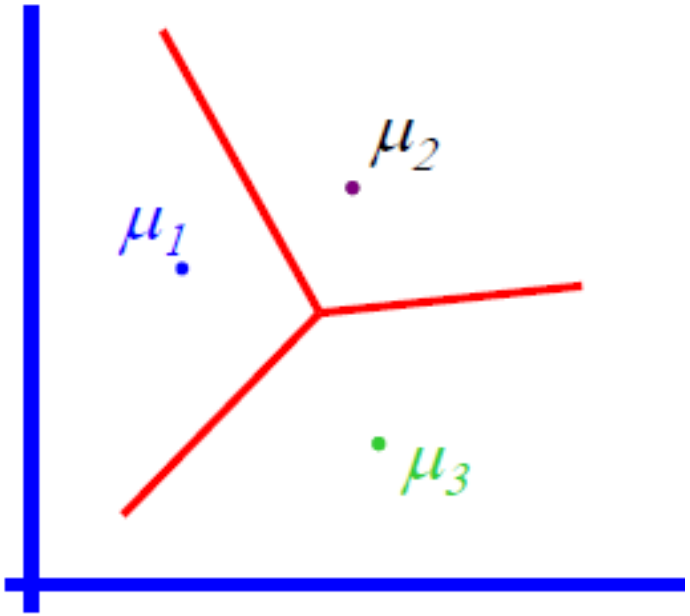
$$\theta = [\mu_1, \ldots, \mu_K, \sigma^2, \pi_1, \ldots, \pi_K]$$

$$\arg\max_\theta \prod_{j=1}^{n} P(x_j|\theta)$$

$$= \arg\max_\theta \prod_{j=1}^{n} \sum_{i=1}^{K} P(y_j = i, x_j|\theta)$$

$$= \arg\max_\theta \prod_{j=1}^{n} \sum_{i=1}^{K} P(y_j = i|\theta) p(x_j|y_j = i|\theta)$$

$$= \arg\max_\theta \prod_{j=1}^{n} \sum_{i=1}^{K} \pi_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}\|x_j - \mu_i\|^2\right)$$



# K-means and GMM

**MLE:** $$\hat{\theta} = \arg\max_{\theta} \prod_{j=1}^{n} \sum_{i=1}^{K} \pi_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}\|x_j - \mu_i\|^2\right)$$

- What happens if we assume **Hard assignment**?

$P(y_j = i) = 1$ if $i = C(j)$

$= 0$ otherwise

**In this case the MLE estimation**:

$$\arg\max_{\theta} \prod_{j=1}^{n} P(x_j|\theta) = \arg\max_{\theta} \prod_{j=1}^{n} \sum_{i=1}^{K} \overbrace{P(y_j = i, x_j|\theta)}^{P(y_j = i)\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}\|x_j - \mu_i\|^2\right)}$$

$$= \arg\max_{\theta} \prod_{j=1}^{n} \exp\left(\frac{-1}{2\sigma^2}\|x_j - \mu_{C(j)}\|^2\right)$$

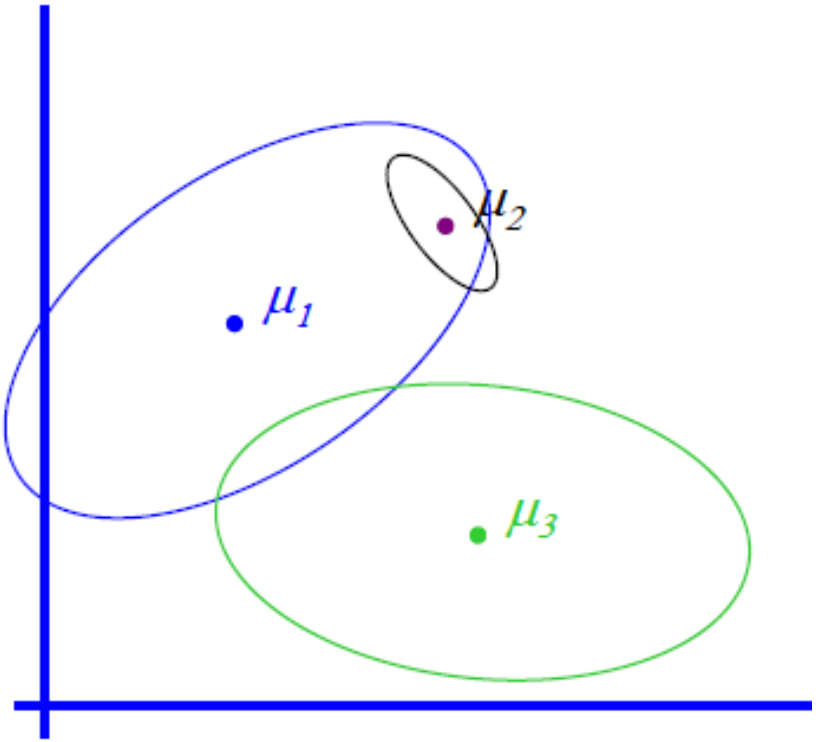
$$= \arg\min_{\mu, C} \sum_{j=1}^{n} \|x_j - \mu_{C(j)}\|^2) = \arg\min_{\mu, C} F(\mu, C)$$

**Same as K-means!!!**

# General GMM

## General GMM –Gaussian Mixture Model (Multi-modal distribution)

- There are k components
- Component *i* has an associated mean vector $\mu_i$
- Each component generates data from a Gaussian with mean $\mu_i$ and covariance matrix $\Sigma_i$. Each data point is generated according to the following recipe:




1) Pick a component at random: Choose component i with probability *P(y=i)*

2) Datapoint x~ N($\mu_I$ ,$\Sigma_i$)

# General GMM

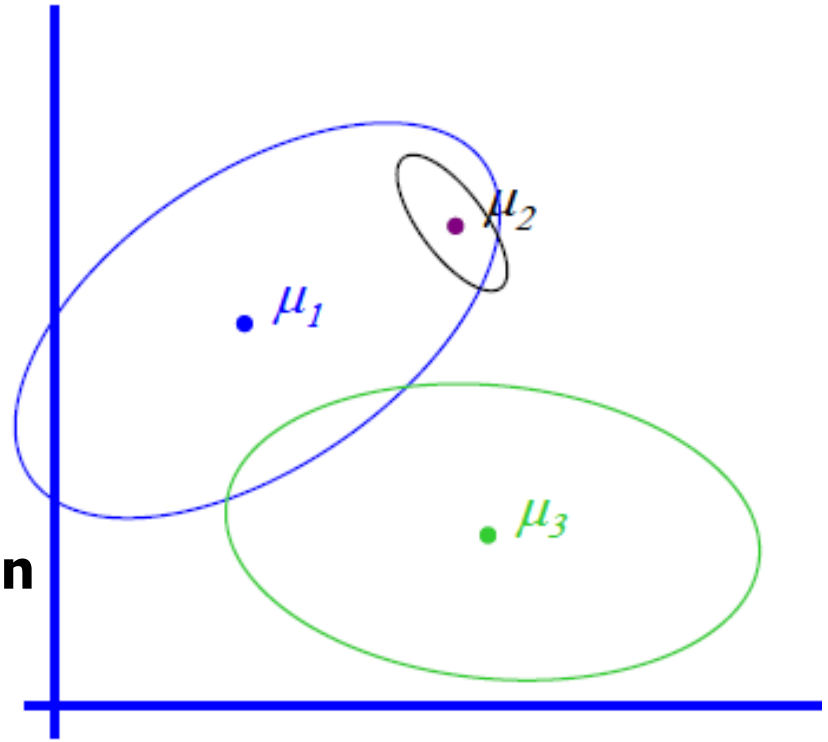
**GMM –Gaussian Mixture Model (Multi-modal distribution)**

$$p(x|y=i) = N(\mu_i, \Sigma_i)$$

$$p(x) = \sum_{i=1}^{K} p(x|y=i)P(y=i)$$

**Mixture component**

**Mixture proportion**



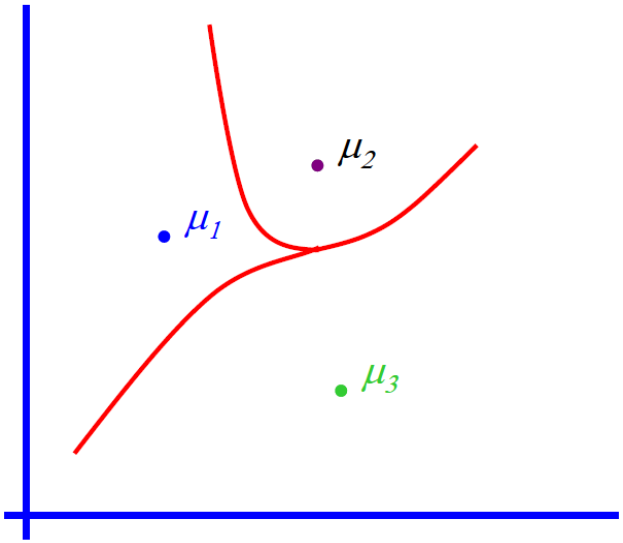
# General GMM

**Assume that**

$\theta = [\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K, \pi_1, \ldots, \pi_K]$ are known.

$p(x|y=i) = N(\mu_i, \Sigma_i)$

$p(y=i) = \pi_i$




**Clustering based on posteriors:**

$$\log \frac{P(y=i|x)}{P(y=j|x)}$$

$$= \log \frac{p(x|y=i)P(y=i)/p(x)}{p(x|y=j)P(y=j)/p(x)}$$

$$= \log \frac{p(x|y=i)\pi_i}{p(x|y=j)\pi_j} = \log \frac{\pi_i \frac{1}{\sqrt{2\pi|\Sigma_i|}} \exp\left[-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right]}{\pi_j \frac{1}{\sqrt{2\pi|\Sigma_j|}} \exp\left[-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\right]}$$

$$= x^T W x + w^T x + c$$

Depends on $\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K, \pi_1, \ldots, \pi_K$

**"Quadratic Decision boundary"** – second-order terms don't cancel out

# General GMM MLE Estimation

**What if we don't know** $\theta = [\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K, \pi_1, \ldots, \pi_K]$?

**⇒ Maximize marginal likelihood (MLE):**

$$\arg\max_\theta \prod_{j=1}^{n} P(x_j|\theta) = \arg\max_\theta \prod_{j=1}^{n} \sum_{i=1}^{K} P(y_j = i, x_j|\theta)$$

$$= \arg\max_\theta \prod_{j=1}^{n} \sum_{i=1}^{K} P(y_j = i|\theta) p(x_j|y_j = i|\theta)$$

$$= \arg\max_\theta \prod_{j=1}^{n} \sum_{i=1}^{K} \pi_i \frac{1}{\sqrt{2\pi|\Sigma_i|}} \exp\left[-\frac{1}{2}(x_j - \mu_i)^T \Sigma_i^{-1}(x_j - \mu_i)\right]$$

How do we find $\theta = [\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K, \pi_1, \ldots, \pi_K]$ which gives max. marginal likelihood?

* Set $\frac{\partial}{\partial \mu_i} \log \text{Prob}(\ldots) = 0$, and solve for $\mu_i$. Non-linear, non-analytically solvable
* Use gradient descent. Doable, but often slow
* Use EM.

# Expectation-Maximization (EM)

**A general algorithm to deal with hidden data, but we will study it in the context of unsupervised learning (hidden class labels = clustering) first.**

- EM is an optimization strategy for objective functions that can be interpreted as likelihoods in the presence of missing data.
- EM is much simpler than gradient methods:
  No need to choose step size.
- EM is an iterative algorithm with two linked steps:
  - o E-step: fill-in hidden values using inference
  - o M-step: apply standard MLE/MAP method to completed data
- We will prove that this procedure monotonically improves the likelihood (or leaves it unchanged). EM always converges to a local optimum of the likelihood.

# Expectation-Maximization (EM)

**A simple case:**

- We have unlabeled data $x_1, x_2, \ldots, x_m$
- We know there are K classes
- We know $P(y=1)=\pi_1$, $P(y=2)=\pi_2$ $P(y=3)$ … $P(y=K)=\pi_K$
- We know common variance $\sigma^2$
- We **don't** know $\mu_1, \mu_2, \ldots \mu_K$ , and we want to learn them

**We can write**

$$p(x_1, \ldots, x_n | \mu_1, \ldots \mu_K) = \prod_{j=1}^{n} p(x_j | \mu_1, \ldots, \mu_K)$$ Independent data

$$= \prod_{j=1}^{n} \sum_{i=1}^{K} p(x_j, y_j = i | \mu_1, \ldots, \mu_K)$$ Marginalize over class

$$= \prod_{j=1}^{n} \sum_{i=1}^{K} p(x_j | y_j = i | \mu_1, \ldots, \mu_K) p(y_j = i)$$

$$\propto \prod_{j=1}^{n} \sum_{i=1}^{K} \exp(-\frac{1}{2\sigma^2} \|x_j - \mu_i\|^2) \pi_i$$ $\Rightarrow$ learn $\mu_1, \mu_2, \ldots \mu_K$

# Expectation (E) step

We want to learn: $\theta = [\mu_1, \ldots, \mu_K]$

Our estimator at the end of iteration t-1: $\theta^{t-1} = [\mu_1^{t-1}, \ldots, \mu_K^{t-1}]$

At iteration t, construct function Q:

$$Q(\theta^t|\theta^{t-1}) = \sum_{j=1}^{n} \sum_{i=1}^{K} P(y_j = i|x_j, \theta^{t-1}) \log P(x_j, y_j = i|\theta^t)$$

**E step**

$$P(y_j = i|x_j, \theta^{t-1}) = P(y_j = i|x_j, \mu_1^{t-1}, \ldots, \mu_K^{t-1})$$

$$\propto P(x_j|y_j = i, \mu_1^{t-1}, \ldots, \mu_K^{t-1}) P(y_j = i)$$

$$\propto \exp(-\frac{1}{2\sigma^2}\|x_j - \mu_i^{t-1}\|^2)\pi_i$$

$$= \frac{\exp(-\frac{1}{2\sigma^2}\|x_j - \mu_i^{t-1}\|^2)\pi_i}{\sum_{i=1}^{K} \exp(-\frac{1}{2\sigma^2}\|x_j - \mu_i^{t-1}\|^2)\pi_i}$$

Equivalent to assigning clusters to each data point in K-means in a soft way

# Maximization (M) step

$$Q(\theta^t|\theta^{t-1}) = \sum_{j=1}^{n}\sum_{i=1}^{K} P(y_j = i|x_j, \theta^{t-1}) \log P(x_j, y_j = i|\theta^t)$$

$$= \sum_{j=1}^{n}\sum_{i=1}^{K} P(y_j = i|x_j, \theta^{t-1})[\log \underbrace{P(x_j|y_j = i, \theta^t)}_{\propto \exp(-\frac{1}{2\sigma^2}\|x_j - \mu_i^t\|^2)} + \log \underbrace{P(y_j = i|\theta^t)}_{\pi_i}]$$

We calculated these weights in the E step

$$R_{i,j}^{t-1} = P(y_j = i|x_j, \theta^{t-1})$$

Joint distribution is simple

**M step** At iteration t, maximize function Q in $\theta^t$:

$$Q(\mu_i^t|\theta^{t-1}) \propto \sum_{j=1}^{n} R_{i,j}^{t-1}(-\frac{1}{2\sigma^2}\|x_j - \mu_i^t\|^2)$$

$$\frac{\partial}{\partial \mu_i^t} Q(\mu_i^t|\theta^{t-1}) = 0 \Rightarrow \sum_{j=1}^{n} R_{i,j}^{t-1}(x_n - \mu_i^t) = 0$$

$$\mu_i^t = \sum_{j=1}^{n} w_j x_j \text{ where } w_j = \frac{R_{i,j}^{t-1}}{\sum_{j=1}^{n} R_{i,j}^{t-1}} = \frac{P(y_j=i|x_j,\theta^{t-1})}{\sum_{l=1}^{n} P(y_l=i|x_l,\theta^{t-1})}$$

Equivalent to updating cluster centers in K-means

# EM for spherical, same variance GMMs

**E-step**

Compute "expected" classes of all datapoints for each class

$$P(y_j = i | x_j, \theta^{t-1}) = \frac{\exp(-\frac{1}{2\sigma^2}\|x_j - \mu_i^{t-1}\|^2)\pi_i^{t-1}}{\sum_{i=1}^{K}\exp(-\frac{1}{2\sigma^2}\|x_j - \mu_i^{t-1}\|^2)\pi_i^{t-1}}$$

In K-means "E-step" we do hard assignment. EM does soft assignment

**M-step**

Compute Max of function Q. [In this example update **μ** given our data's class membership distributions (weights) ]

$$\mu_i^t = \sum_{j=1}^{n} w_j x_j \quad \text{where } w_j = \frac{P(y_j=i|x_j,\theta^{t-1})}{\sum_{l=1}^{n} P(y_l=i|x_l,\theta^{t-1})}$$

Iterate. Exactly the same as MLE with weighted data.

# EM for general GMMs

**The more general case:**

- We have unlabeled data $x_1, x_2, \ldots, x_m$
- We know there are K classes
- We **don't** know P(y=1)=$\pi_1$, P(y=2)=$\pi_2$ P(y=3) … P(y=K)=$\pi_K$
- We **don't** know $\Sigma_1$,… $\Sigma_K$
- We **don't** know $\mu_1$, $\mu_2$, … $\mu_K$

**We want to learn**: $\theta = [\mu_1, \ldots, \mu_K, \pi_1, \ldots, \pi_K, \Sigma_1, \ldots, \Sigma_K]$

Our estimator at the end of iteration t-1:

$$\theta^{t-1} = [\mu_1^{t-1}, \ldots, \mu_K^{t-1}, \pi_1^{t-1}, \ldots, \pi_K^{t-1}, \Sigma_1^{t-1}, \ldots, \Sigma_K^{t-1}]$$

**The idea is the same:**

At iteration t, construct function Q (E step) and maximize it in $\theta^t$ (M step)

$$Q(\theta^t|\theta^{t-1}) = \sum_{j=1}^{n} \sum_{i=1}^{K} P(y_j = i|x_j, \theta^{t-1}) \log P(x_j, y_j = i|\theta^t)$$

# EM for general GMMs

At iteration t, construct function Q (E step) and maximize it in $\theta^t$ (M step)

$$Q(\theta^t|\theta^{t-1}) = \sum_{j=1}^{n}\sum_{i=1}^{K} P(y_j = i|x_j, \theta^{t-1}) \log P(x_j, y_j = i|\theta^t)$$

**E-step**

Compute "expected" classes of all datapoints for each class

$$R_{i,j}^{t-1} = P(y_j = i|x_j, \theta^{t-1}) \quad = \frac{\mathcal{N}(x_j|\mu_i^{t-1}, \Sigma_i^{t-1})\pi_i^{t-1}}{\sum_{i=1}^{K}\mathcal{N}(x_j|\mu_i^{t-1}, \Sigma_i^{t-1})\pi_i^{t-1}}$$

**M-step** $\frac{\partial}{\partial\theta^t}Q(\theta^t|\theta^{t-1}) = 0$

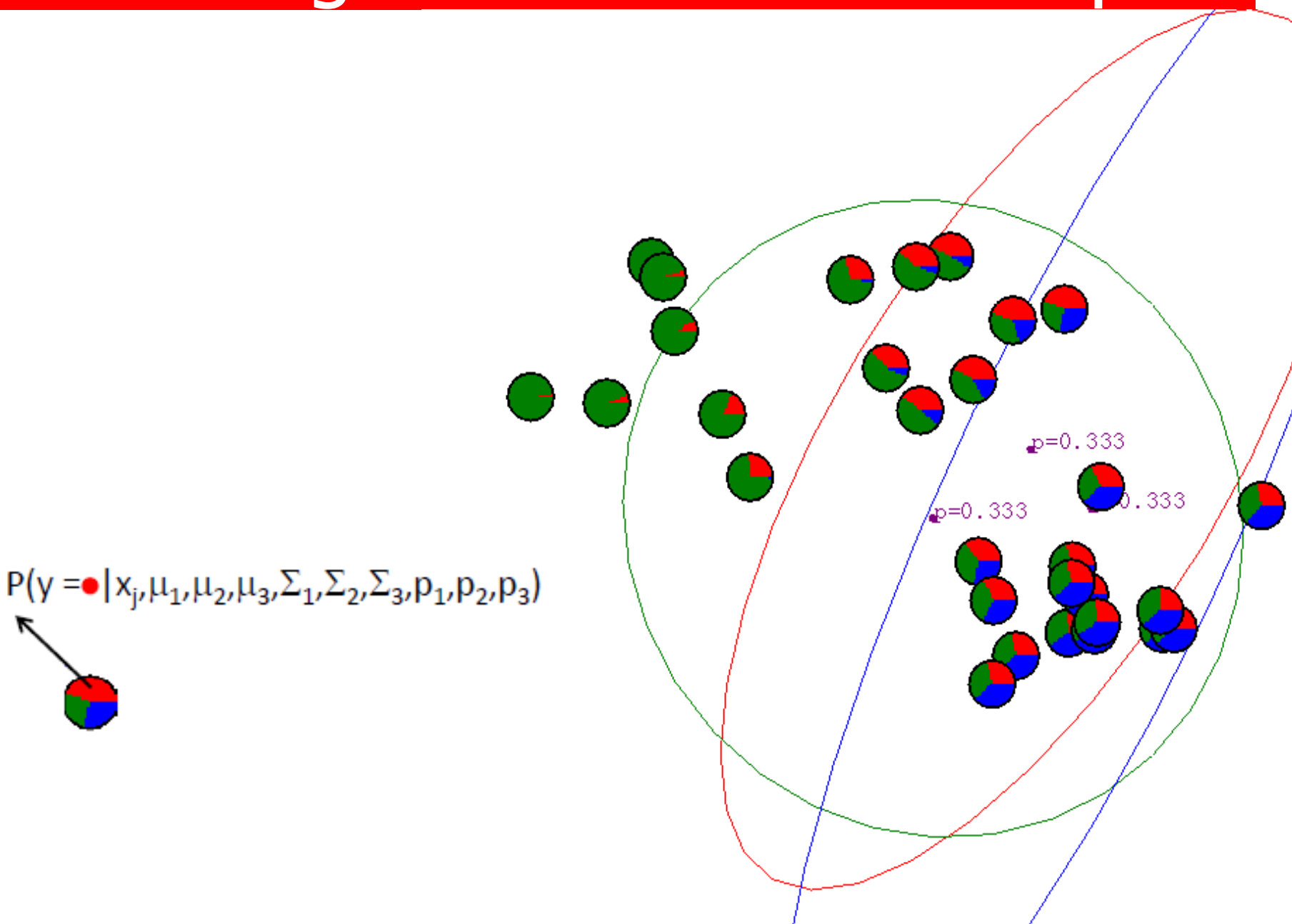
Compute MLEs given our data's class membership distributions (weights)

$$\mu_i^t = \sum_{j=1}^{n} w_j x_j \quad \text{where } w_j = \frac{R_{i,j}^{t-1}}{\sum_{j=1}^{n} R_{i,j}^{t-1}}$$

$$\Sigma_i^t = \sum_{j=1}^{n} w_j (x_j - \mu_i^t)^T (x_j - \mu_i^t)$$

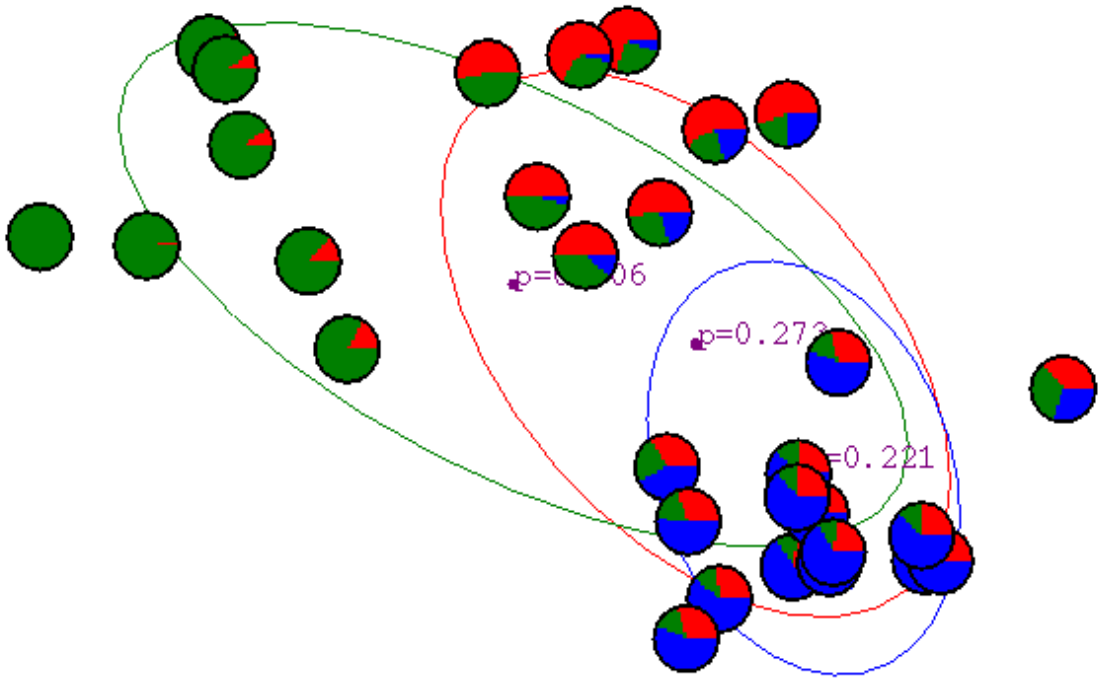
$$\pi_i^t = \frac{1}{n}\sum_{j=1}^{n} R_{i,j}^{t-1}$$

# EM for general GMMs: Example



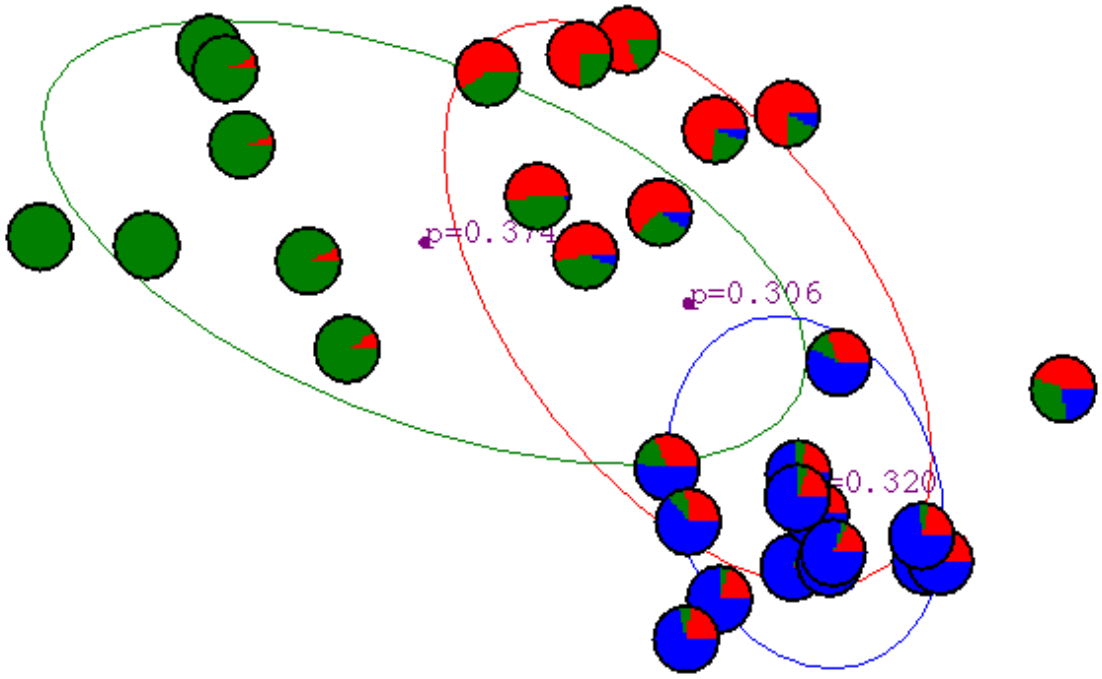
# EM for general GMMs: Example

**After 1st iteration**



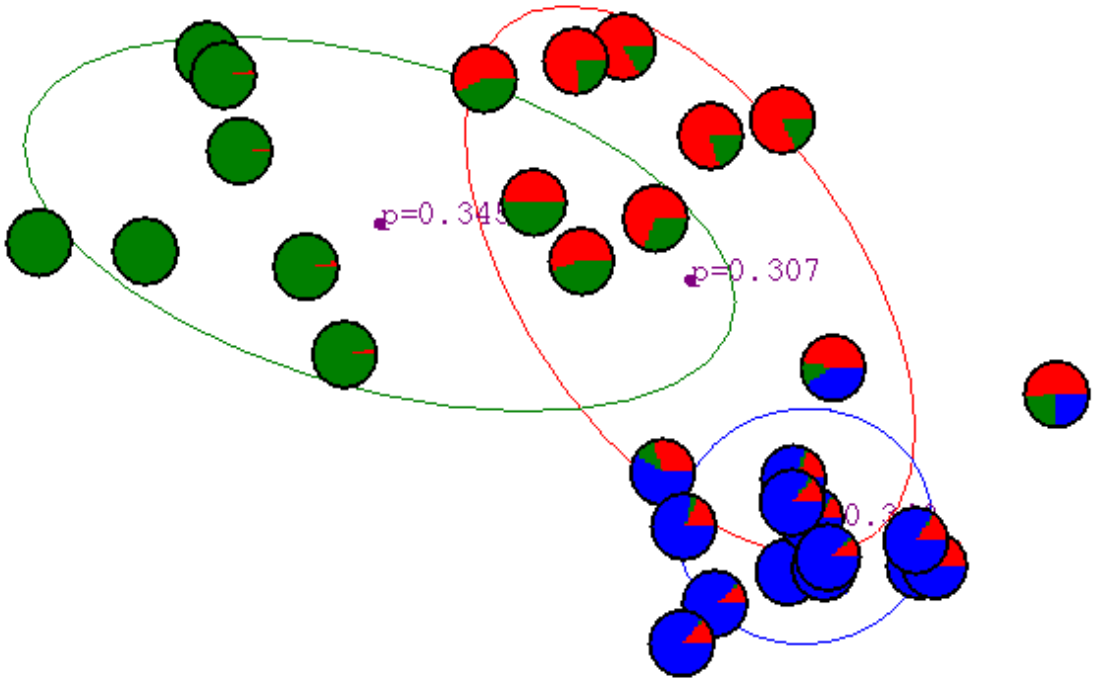
# EM for general GMMs: Example

**After 2nd iteration**



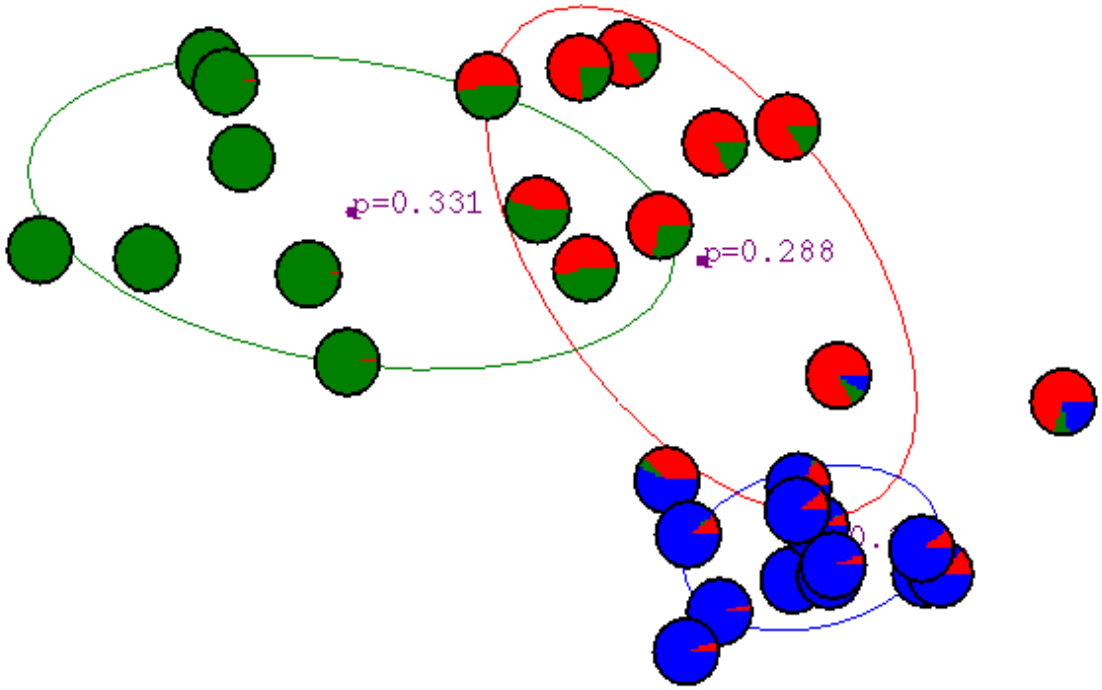
# EM for general GMMs: Example

**After 3rd iteration**



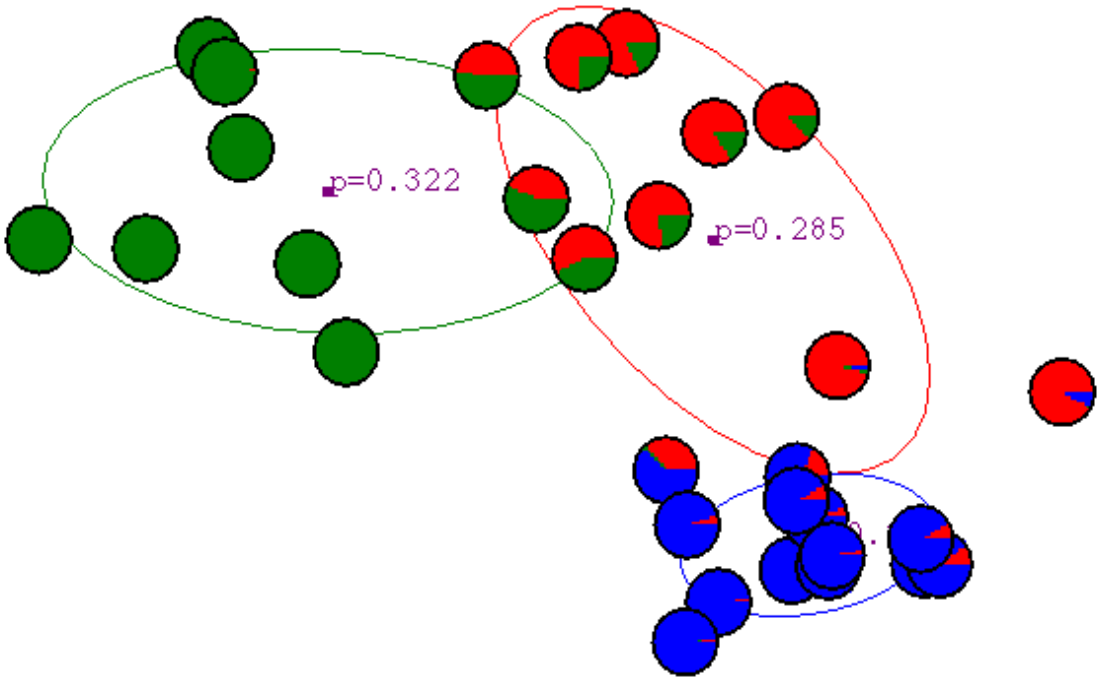
# EM for general GMMs: Example

**After 4th iteration**



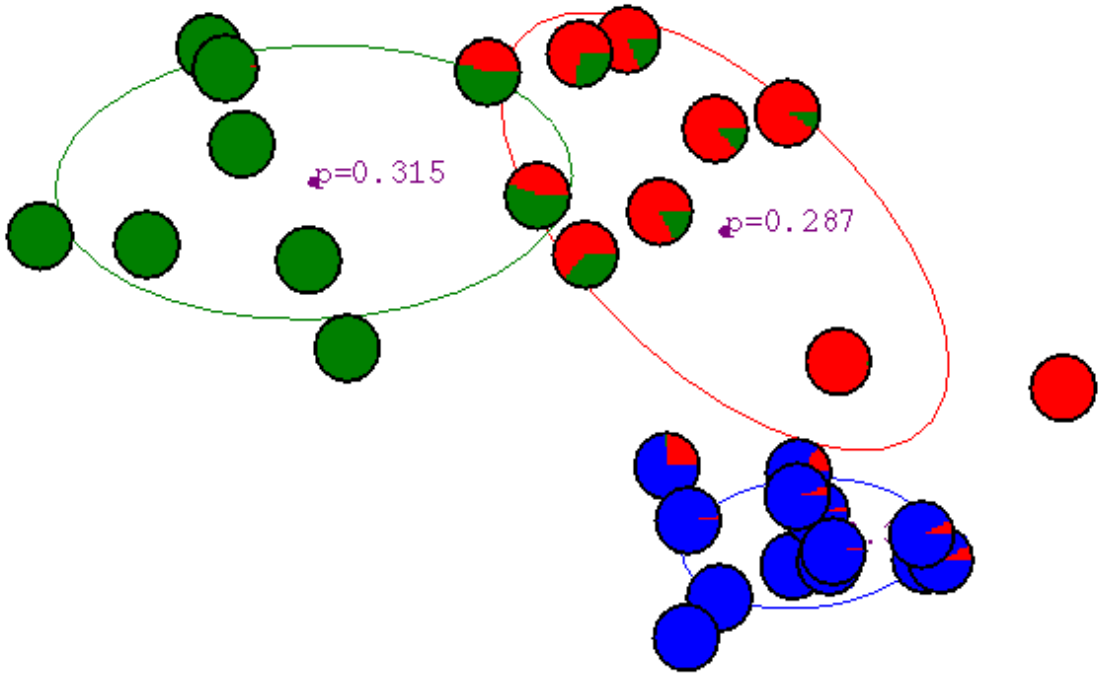
# EM for general GMMs: Example

**After 5th iteration**



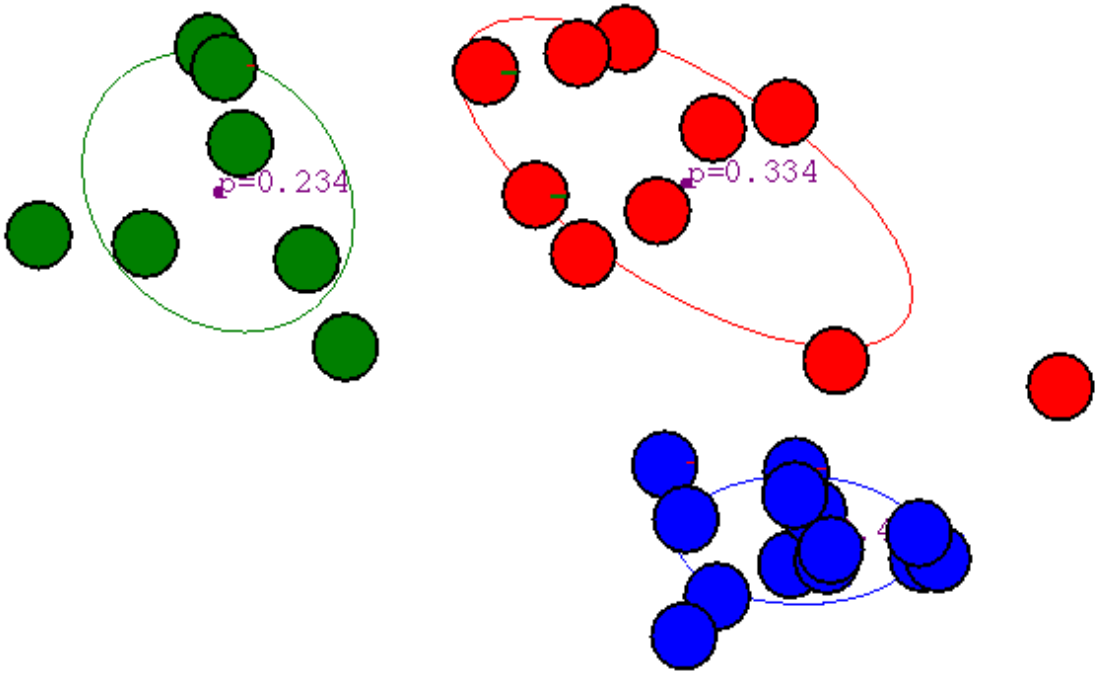
# EM for general GMMs: Example

**After 6th iteration**



# EM for general GMMs: Example

**After 20th iteration**



# GMM for Density Estimation