EM Algorithm

K-means Recap …

- Randomly initialize *k* centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$
- **Classify**: Assign each point $j \in \{1, \ldots m\}$ to nearest center:
  - $C^{(t)}(j) \leftarrow \arg\min_i ||\mu_i - x_j||^2$
- **Recenter**: μ_i becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg\min_\mu \sum_{j:C(j)=i} ||\mu - x_j||^2$
  - Equivalent to $\mu_i \leftarrow$ average of its points!

What is K-means optimizing?

- Potential function F($\boldsymbol{\mu}$,C) of centers $\boldsymbol{\mu}$ and point allocations C:

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

- Optimal K-means:
  - $\min_{\boldsymbol{\mu}} \min_{C} F(\boldsymbol{\mu}, C)$

K-means algorithm

- Optimize potential function:

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- **K-means algorithm:**

(1) Fix $\boldsymbol{\mu}$, optimize C

$$\min_{C(1),C(2),\ldots,C(m)} \sum_{j=1}^{m} \|\mu_{C(j)} - x_j\|^2$$

独立

$$= \sum_{j=1}^{m} \underbrace{\min_{C(j)} \|\mu_{C(j)} - x_j\|^2}_{\text{Exactly first step – assign each point to the nearest cluster center}}$$

- Optimize potential function:

$$\min_{\mu}\min_{C} F(\mu, C) = \min_{\mu}\min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- **K-means algorithm:**

(2) Fix C, optimize **μ**

$$\min_{\mu_1,\mu_2,\ldots\mu_K} \sum_{i=1}^{K} \sum_{j:C(j)=i} \|\mu_i - x_j\|^2$$

$$= \sum_{i=1}^{K} \underbrace{\min_{\mu_i} \sum_{j:C(j)=i} \|\mu_i - x_j\|^2}_{\text{Solution: average of points in cluster i}}$$

Solution: average of points in cluster i

Exactly second step (re-center)

K-means algorithm

- Optimize potential function:

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- **K-means algorithm:** (coordinate ascent on F)

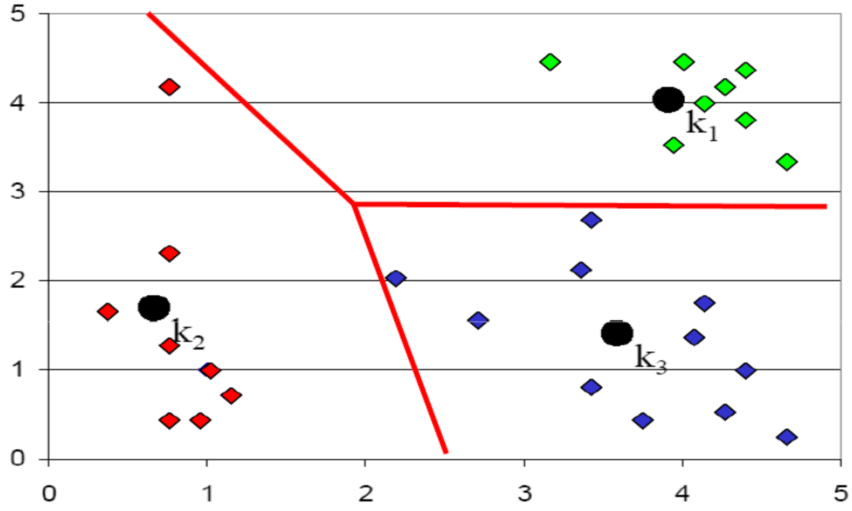
(1) Fix μ, optimize C **Expectation step**

(2) Fix C, optimize μ **Maximization step**

Today, we will see a generalization of this approach:

EM algorithm

K-means Decision boundaries



"Linear" Decision Boundaries

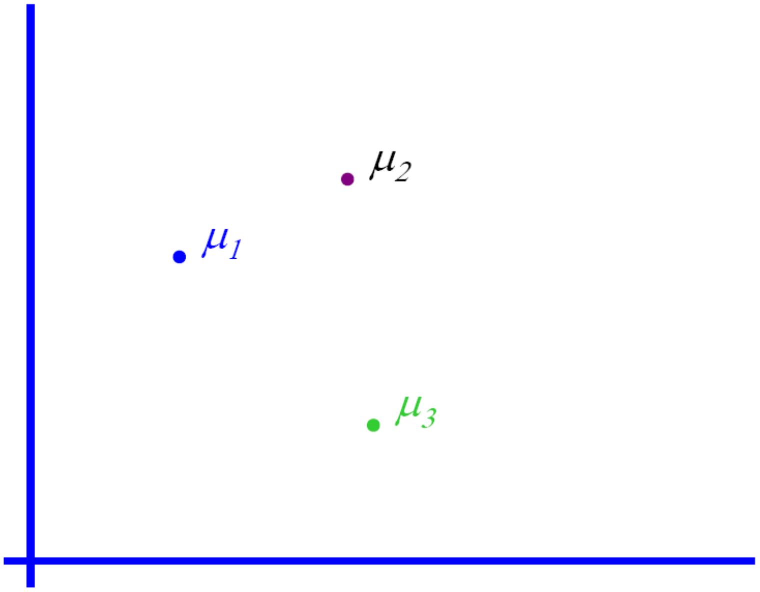
Generative Model:

Assume data comes from a mixture of K Gaussians distributions with same variance

K-means: Generative model

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

- There are k components
- Component i has an associated mean vector μ_i

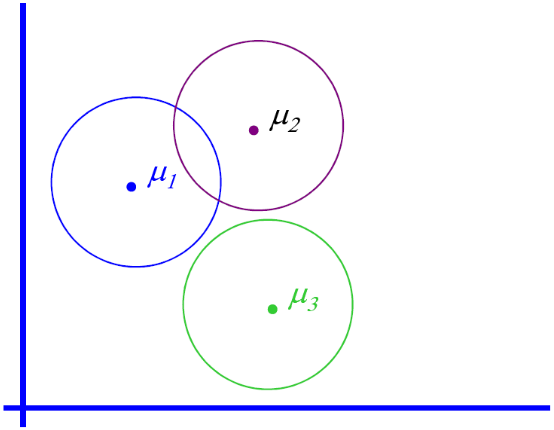


K-means: Generative model

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

- There are k components
- Component i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 I$

Each data point is generated accordingto the following recipe:



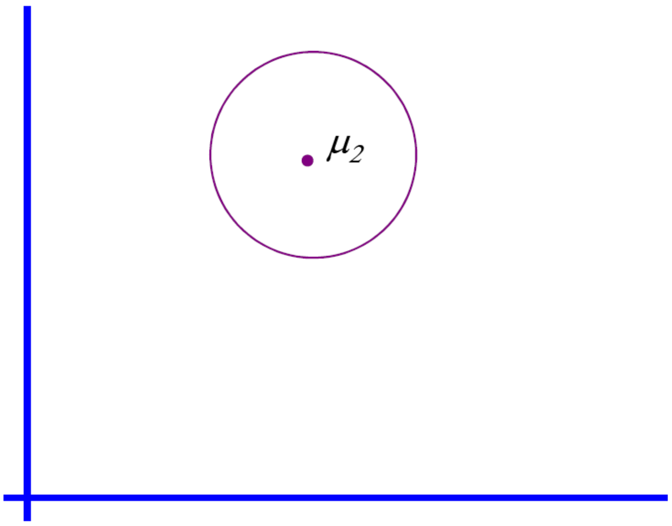
K-means: Generative model

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

- There are k components
- Component i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 I$

Each data point is generated accordingto the following recipe:

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



K-means: Generative model

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

- There are k components
- Component i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 I$

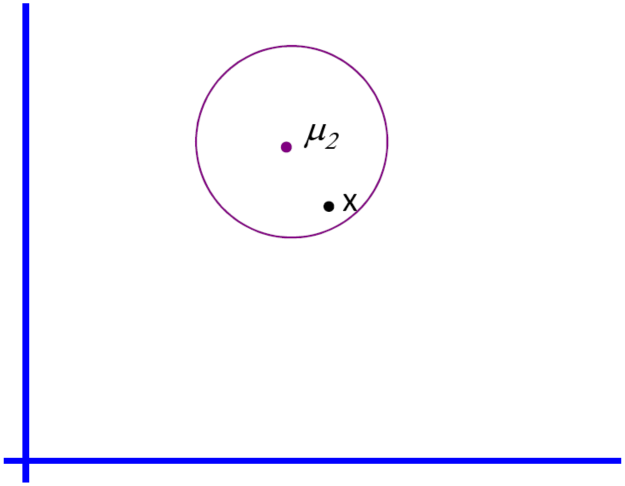
Each data point is generated accordingto the following recipe:

1) Pick a component at random:

   Choose component i with

   probability P(y=i)

2) Datapoint $x \sim N(\mu_i, \sigma^2 I)$

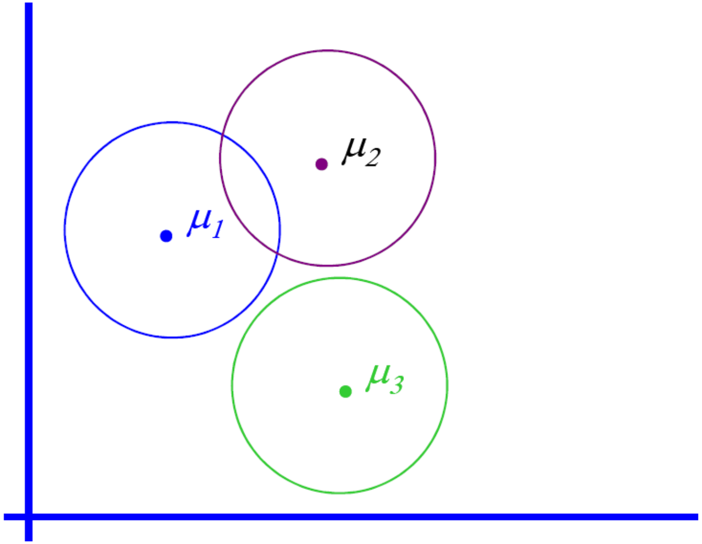


K-means: Generative model

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

$$p(x \mid y=i) \sim \mathrm{N}(\mu_i, \sigma^2 I)$$

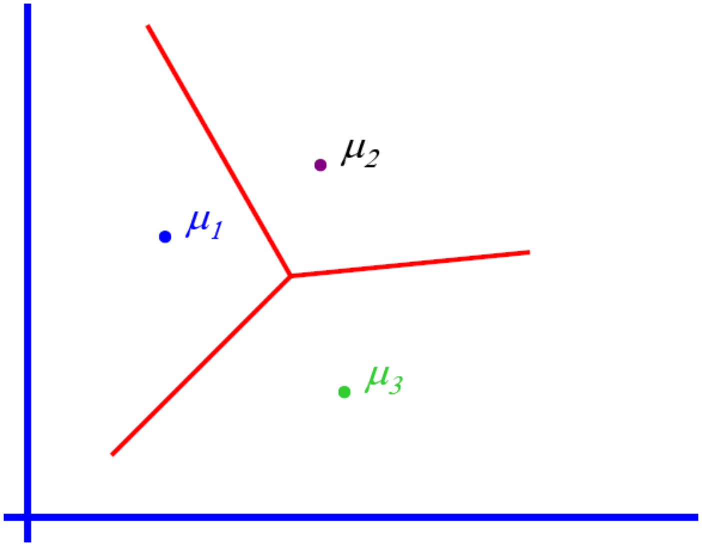
$$p(x) = \sum_i \underbrace{p(x \mid y=i)}_{\text{Mixture component}} \ \underbrace{P(y=i)}_{\text{Mixture proportion}}$$



K-means: Generative model

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

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



Gaussian Bayes Classifier:

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

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

$$= w^T x$$

Depends on $\mu_1, \mu_2, .., \mu_K, \sigma^2$, P(y=1),..., P(Y=k)

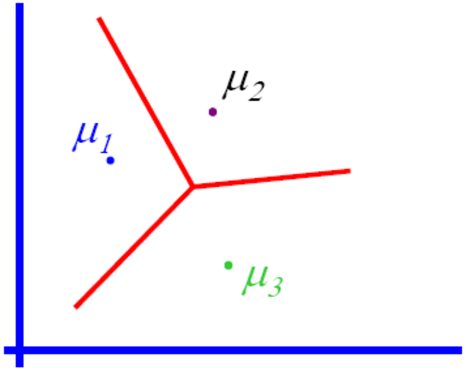
"Linear Decision boundary" – Recall that second-order terms cancel out

K-means: Generative model

Maximum Likelihood Estimate (MLE)

$$\underset{\mu_1, \mu_2, \ldots, \mu_K, \sigma^2, P(y=1), \ldots, P(Y=k)}{\text{argmax}} \prod_i P(y_i, x_i)$$

But we don't know y_i's!!!



Maximize marginal likelihood:

$$\text{argmax} \prod_j P(x_j) = \text{argmax} \prod_j \sum_{i=1}^{K} P(y_j=i, x_j)$$
$$= \text{argmax} \prod_j \sum_i^K P(y_j=i) p(x_j | y_j=i)$$

K-means: Generative model

Maximize marginal likelihood:

$$\operatorname{argmax} \prod_j P(x_j) = \operatorname{argmax} \prod_j \sum_{i=1}^{K} P(y_j=i,x_j)$$

$$= \operatorname{argmax} \prod_j \sum_{i=1}^{K} P(y_j=i)p(x_j \mid y_j=i)$$

$$P(y_j = i, x_j) \propto P(y_j = i) \exp\left[-\frac{1}{2\sigma^2}\left\|x_j - \mu_i\right\|^2\right]$$

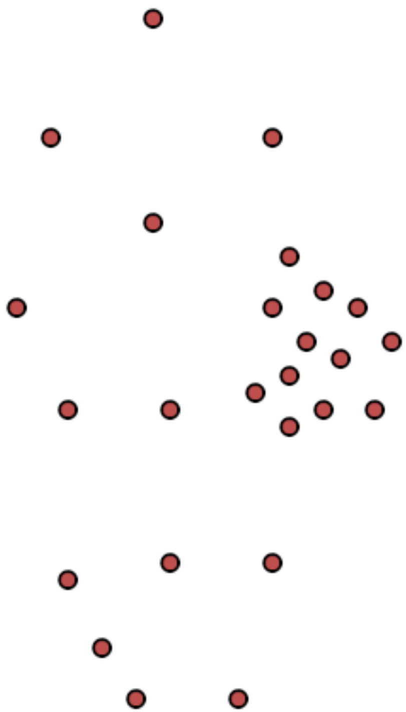
If each x_j belongs to one class $C(j)$ (hard assignment), marginal likelihood:

$P(y_j=i) = 1$ or 0 $\quad\quad$ 1 if $i = C(j)$

$$\prod_{j=1}^{m}\sum_{i=1}^{k} P(y_j = i, x_j) \propto \prod_{j=1}^{m} \exp\left[-\frac{1}{2\sigma^2}\left\|x_j - \mu_{C(j)}\right\|^2\right] = \sum_{j=1}^{m} -\frac{1}{2\sigma^2}\left\|x_j - \mu_{C(j)}\right\|^2$$

Same as K-means!!!

(One) bad case for K-means

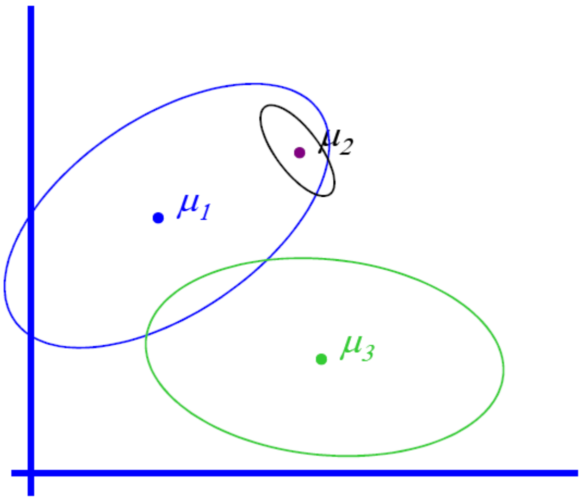


- Clusters may not be linearly separable
- Clusters may overlap
- Some clusters may be “wider” than others

General GMM

GMM – Gaussian Mixture Model (Multi-modal distribution)

- There are k components
- Component i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix Σ_i



Each data point is generated according to the following recipe:

1) Pick a component at random:

   Choose component i with

   probability $P(y=i)$

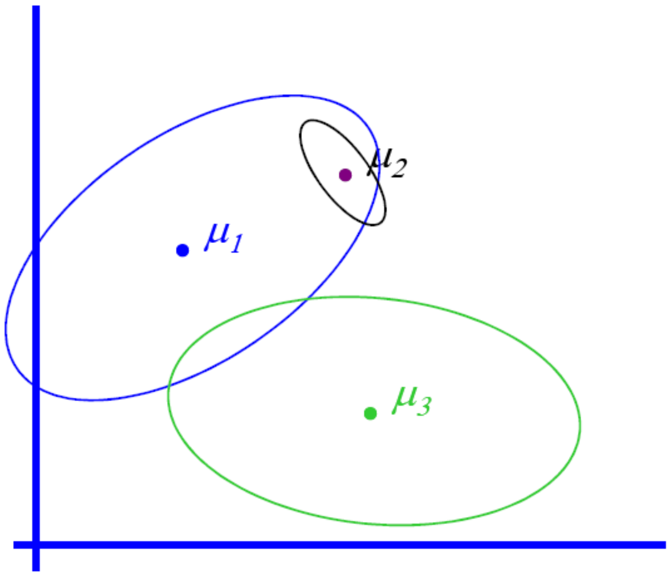
2) Datapoint $x \sim N(\mu_i, \Sigma_i)$

General GMM

GMM – Gaussian Mixture Model (Multi-modal distribution)

$$p(x|y=i) \sim \mathrm{N}(\mu_i, \Sigma_i)$$

$$p(x) = \sum_i \underbrace{p(x|y=i)}_{\text{Mixture component}} \underbrace{P(y=i)}_{\text{Mixture proportion}}$$



General GMM

GMM – Gaussian Mixture Model (Multi-modal distribution)

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

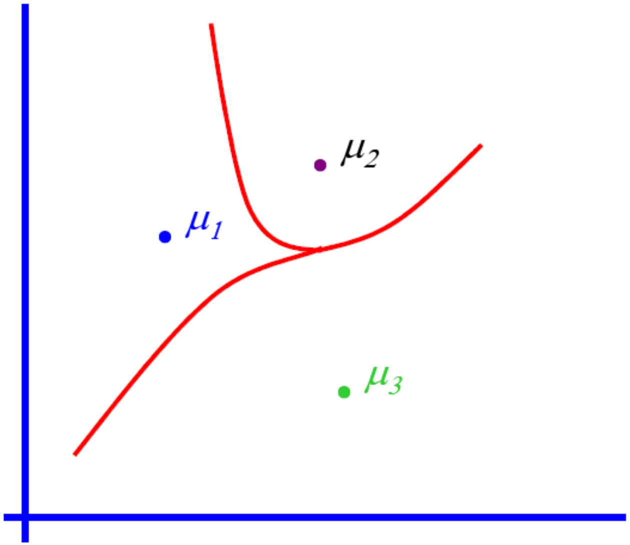
Gaussian Bayes Classifier:

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

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

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

Depend on $\mu_1, \mu_2, .., \mu_K, \Sigma_1, \Sigma_2, .., \Sigma_K$, P(y=1),..., P(Y=k)



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

General GMM

Maximize marginal likelihood:

$$\text{argmax} \prod_j P(x_j) = \text{argmax} \prod_j \sum_{i=1}^{K} P(y_j=i,x_j)$$
$$= \text{argmax} \prod_j \sum_{i=1}^{K} P(y_j=i)p(x_j|y_j=i)$$

Uncertain about class of each x_j (soft assignment), $P(y_j=i) = P(y=i)$

$$\prod_{j=1}^{m}\sum_{i=1}^{k} P(y_j = i, x_j) \propto \prod_{j=1}^{m}\sum_{i=1}^{k} P(y = i)\frac{1}{\sqrt{\det(\Sigma_i)}}\exp\left[-\frac{1}{2}(x_j - \mu_i)^T \Sigma_i (x_j - \mu_i)\right]$$

How do we find the μ_i's which give max. marginal likelihood?

* Set $\frac{\partial}{\partial \mu_i}$ log Prob (....) = 0 and solve for μ_i's. Non-linear non-analytically solvable

* Use gradient descent: Often slow but doable

Expectation-Maximization (EM)

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

- EM is an optimization strategy for objective functions that can be interpreted as likelihoods in the presence of missing data.
- It is much simpler than gradient methods:

  No need to choose step size.
  Enforces constraints automatically.
  Calls inference and fully observed learning as subroutines.

- EM is an Iterative algorithm with two linked steps:

  E-step: fill-in hidden values using inference
  M-step: apply standard MLE/MAP method to completed data

- We will prove that this procedure monotonically improves the likelihood (or leaves it unchanged). Thus it always converges to a local optimum of the likelihood.

Expectation-Maximization (EM)

A simple case:

We have unlabeled data $\boldsymbol{x}_1\ \boldsymbol{x}_2 \ldots \boldsymbol{x}_m$

We know there are k classes

We know P(y=1), P(y=2) P(y=3) … P(y=K)

We <u>don't</u> know $\boldsymbol{\mu}_1\ \boldsymbol{\mu}_2 \ldots \boldsymbol{\mu}_k$

We know common variance σ^2

We can write P(data | $\boldsymbol{\mu}_1 \ldots \boldsymbol{\mu}_k$)

$$= \mathrm{p}\left(x_1 \ldots x_m \middle| \mu_1 \ldots \mu_k\right)$$

$$= \prod_{j=1}^{m} \mathrm{p}\left(x_j \middle| \mu_1 \ldots \mu_k\right)$$

Independent data

$$= \prod_{j=1}^{m} \sum_{i=1}^{k} \mathrm{p}\left(x_j \middle| \mu_i\right) \mathrm{P}(y = i)$$

Marginalize over class

$$\propto \prod_{j=1}^{m} \sum_{i=1}^{k} \exp\left(-\frac{1}{2\sigma^2}\left\|x_j - \mu_i\right\|^2\right) \mathrm{P}(y = i)$$

Expectation (E) step

If we know $\mu_1,\ldots,\mu_k$ $\rightarrow$ easily compute prob. point x_j belongs to class y=i

$$P\left(y=i \middle| x_j, \mu_1 \ldots \mu_k\right) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|x_j-\mu_i\right\|^2\right) P(y=i)$$

Simply evaluate gaussian and normalize

Maximization (M) step

If we know prob. point x_j belongs to class y=i

$\rightarrow$ MLE for μ_i is weighted average

imagine multiple copies of each x_j, each with weight $P(y=i|x_j)$:

$$\mu_i = \frac{\sum_{j=1}^{m} P\left(y=i \middle| x_j\right) x_j}{\sum_{j=1}^{m} P\left(y=i \middle| x_j\right)}$$

EM for spherical, same variance GMMs

E-step

Compute "expected" classes of all datapoints for each class

$$P(y=i|x_j,\mu_1...\mu_k) \propto \exp\left(-\frac{1}{2\sigma^2}\|x_j-\mu_i\|^2\right)P(y=i)$$

In K-means "E-step"
we do hard assignment

EM does soft assignment

M-step

Compute Max. like **μ** given our data's class membership distributions

$$\mu_i = \frac{\sum_{j=1}^{m} P(y=i|x_j)x_j}{\sum_{j=1}^{m} P(y=i|x_j)}$$

EM for axis-aligned GMMs

$$\Sigma_i = \begin{pmatrix} 0 & 0 & \sigma^2_{i,3} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma^2_{i,m-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma^2_{i,m} \end{pmatrix}$$

Iterate. On iteration t let our estimates be

$\lambda_t = \{ \mu_1^{(t)}, \mu_2^{(t)} \ldots \mu_k^{(t)}, \Sigma_1^{(t)}, \Sigma_2^{(t)} \ldots \Sigma_k^{(t)}, p_1^{(t)}, p_2^{(t)} \ldots p_k^{(t)} \}$

$p_i^{(t)} = p^{(t)}(y=i)$

E-step

Compute "expected" classes of all datapoints for each class

$$\mathrm{P}\left(y=i \middle| x_j, \lambda_t\right) \propto p_i^{(t)} \mathrm{p}\left(x_j \middle| \mu_i^{(t)}, \Sigma_i^{(t)}\right)$$

Just evaluate a Gaussian at x_j

M-step

Compute Max. like **μ** given our data's class membership distributions

$$\mu_i^{(t+1)} = \frac{\sum_j \mathrm{P}\left(y=i \middle| x_j, \lambda_t\right) x_j}{\sum_j \mathrm{P}\left(y=i \middle| x_j, \lambda_t\right)}$$

$$p_i^{(t+1)} = \frac{\sum_j \mathrm{P}\left(y=i \middle| x_j, \lambda_t\right)}{m}$$

m = #data points

EM for general GMMs

Iterate. On iteration t let our estimates be

$\lambda_t = \{ \mu_1^{(t)}, \mu_2^{(t)} \dots \mu_k^{(t)}, \Sigma_1^{(t)}, \Sigma_2^{(t)} \dots \Sigma_k^{(t)}, p_1^{(t)}, p_2^{(t)} \dots p_k^{(t)} \}$

$p_i^{(t)}$ is shorthand for estimate of $P(y=i)$ on t'th iteration

E-step

Compute "expected" classes of all datapoints for each class

$$P\left(y = i \middle| x_j, \lambda_t\right) \propto p_i^{(t)} p\left(x_j \middle| \mu_i^{(t)}, \Sigma_i^{(t)}\right)$$

Just evaluate a Gaussian at x_j

M-step

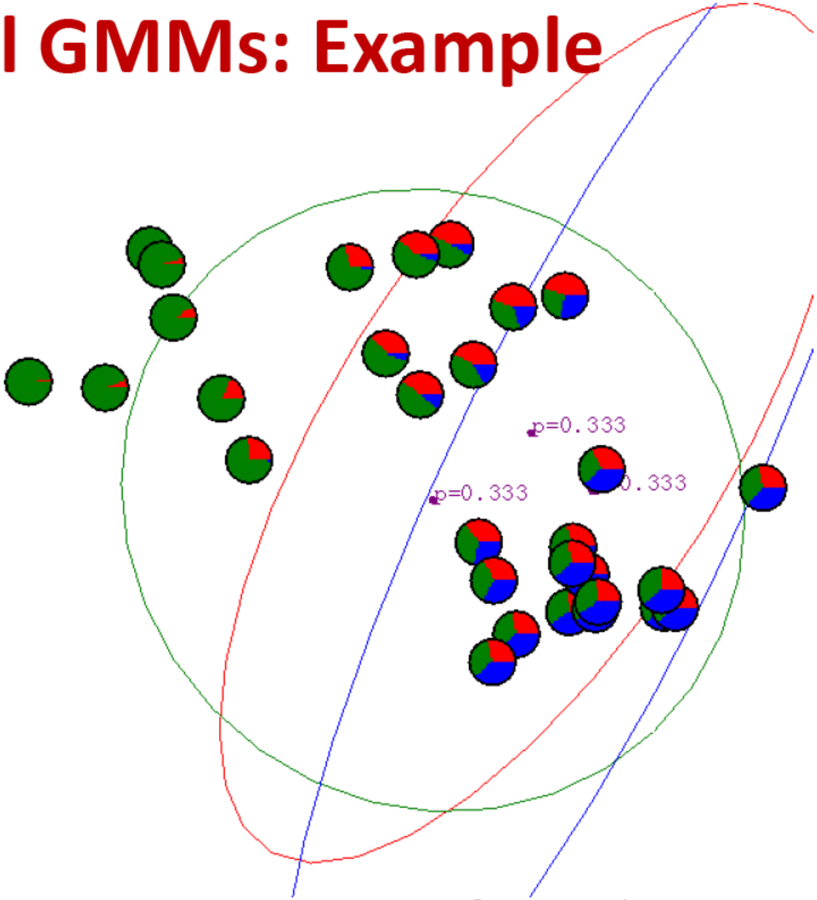
Compute MLEs given our data's class membership distributions (weights)

$$\mu_i^{(t+1)} = \frac{\sum_j P\left(y = i \middle| x_j, \lambda_t\right) x_j}{\sum_j P\left(y = i \middle| x_j, \lambda_t\right)} \qquad \Sigma_i^{(t+1)} = \frac{\sum_j P\left(y = i \middle| x_j, \lambda_t\right)\left(x_j - \mu_i^{(t+1)}\right)\left(x_j - \mu_i^{(t+1)}\right)^T}{\sum_j P\left(y = i \middle| x_j, \lambda_t\right)}$$

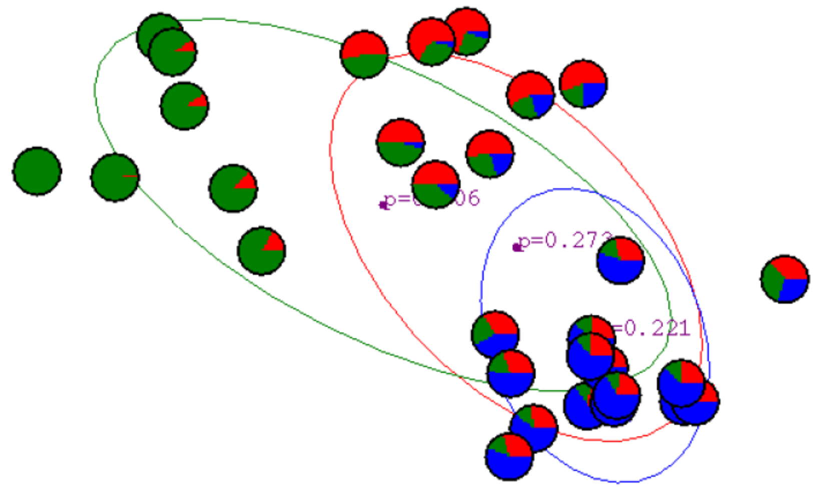
$$p_i^{(t+1)} = \frac{\sum_j P\left(y = i \middle| x_j, \lambda_t\right)}{m}$$

m = #data points

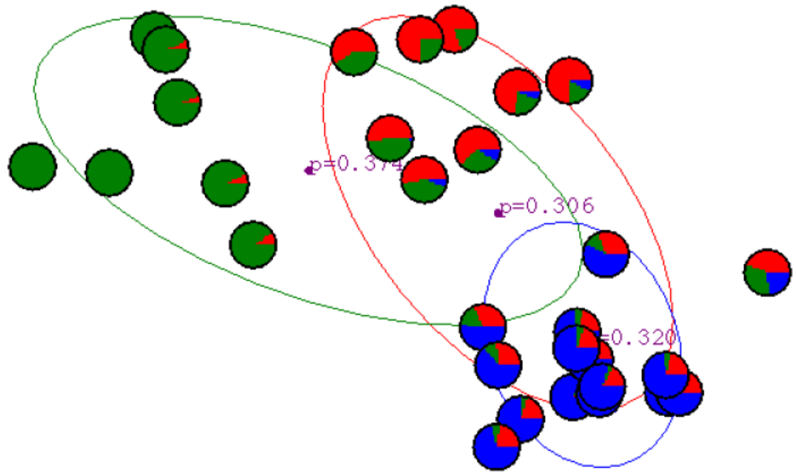
EM for general GMMs: Example



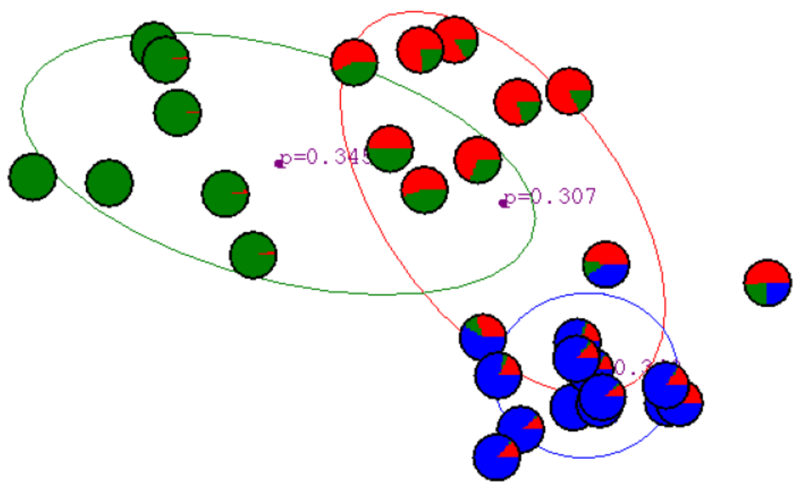
After 1st iteration



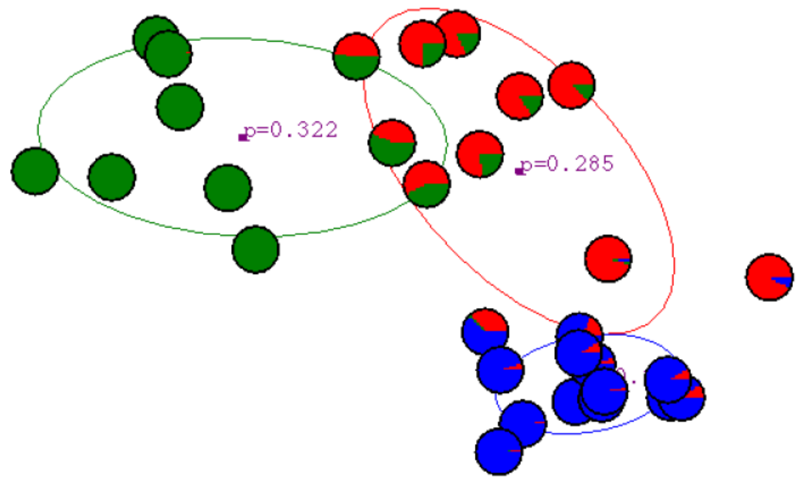
After 2nd iteration



After 3rd iteration



After 5th iteration



After 20th iteration