Gaussian Mixture Models

Pradeep Ravikumar

Co-instructor: Manuela Veloso

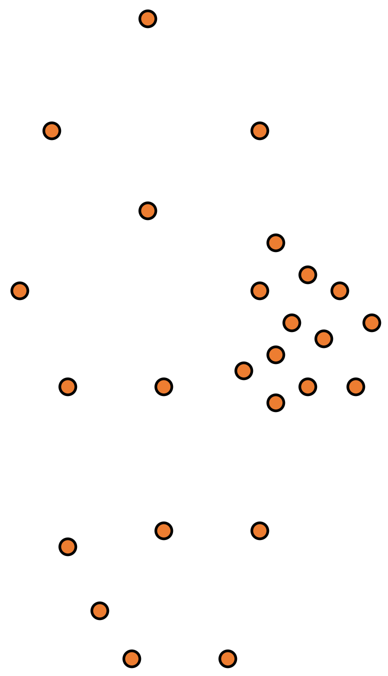
Machine Learning 10-701

Some slides courtesy of Eric Xing, Carlos Guestrin

MACHINE LEARNING DEPARTMENT

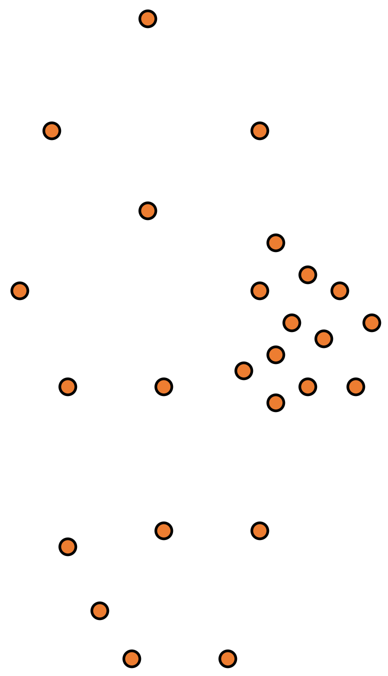
Carnegie Mellon.
School of Computer Science

(One) bad case for K-means



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

(One) bad case for K-means



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

Partitioning Algorithms

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

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

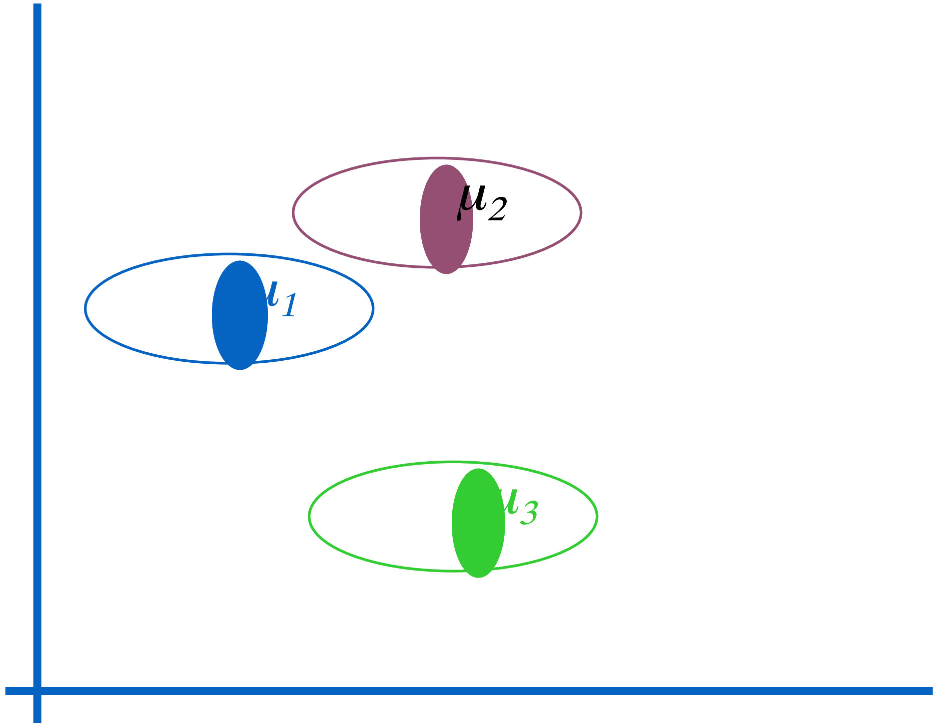
  Generative approach: think of each cluster as a component distribution, and any data point is drawn from a "mixture" of multiple component distributions

Gaussian Mixture Model

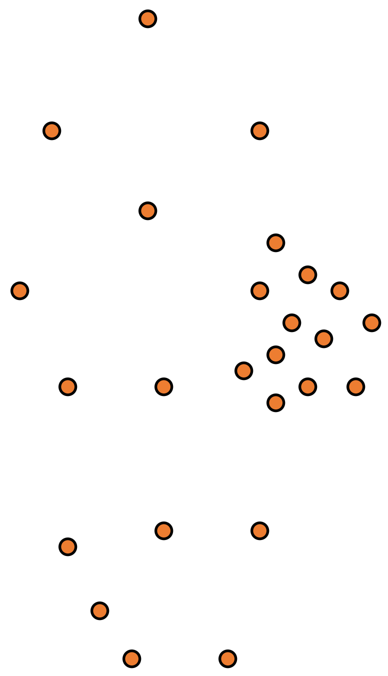
Mixture of K Gaussian distributions: (Multi-modal distribution)

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

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



(One) bad case for K-means



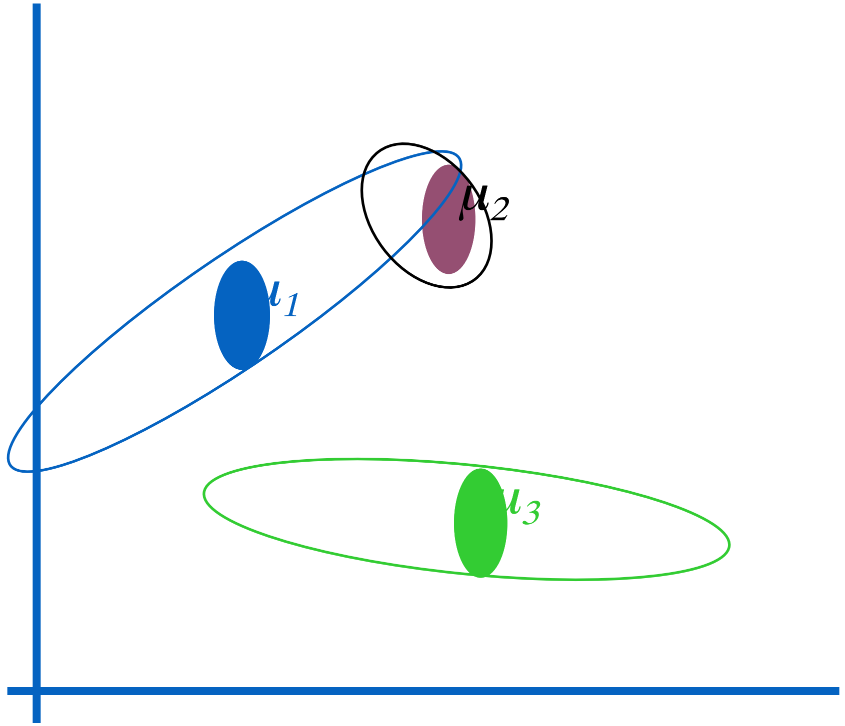
- Clusters may overlap
- Some clusters may be “wider” than others
- Clusters may not be linearly separable

General GMM

GMM – Gaussian Mixture Model (Multi-modal distribution)

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

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



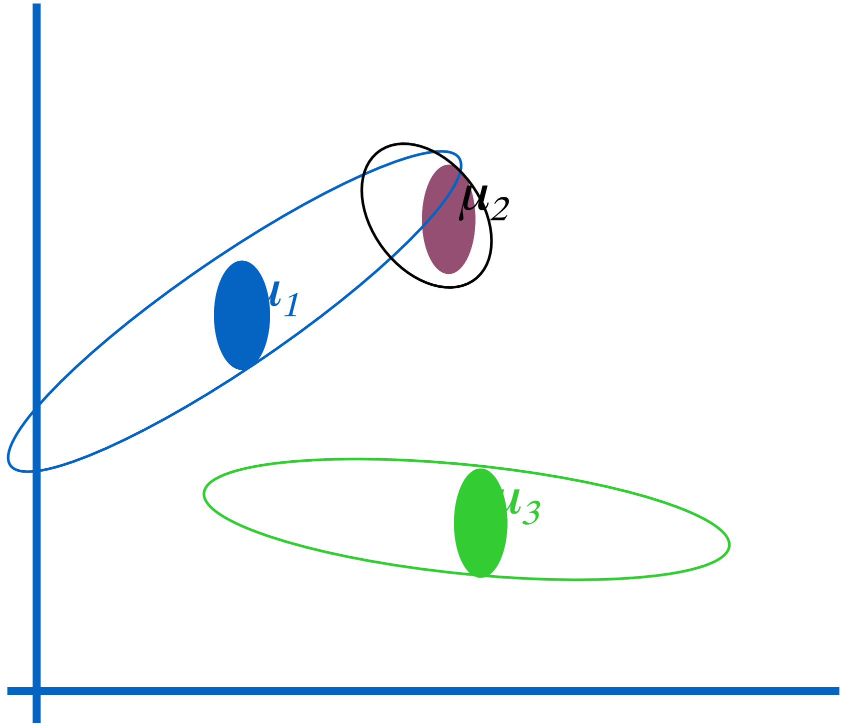
General GMM

GMM – Gaussian Mixture Model (Multi-modal distribution)

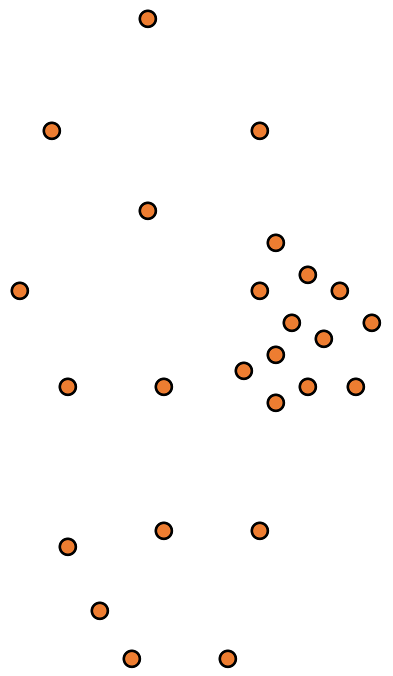
- There are k components
- Component *i* has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix Σ_i

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) Data-point x ~ $N(\mu_i, \Sigma_i)$



(One) bad case for K-means



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

General GMM

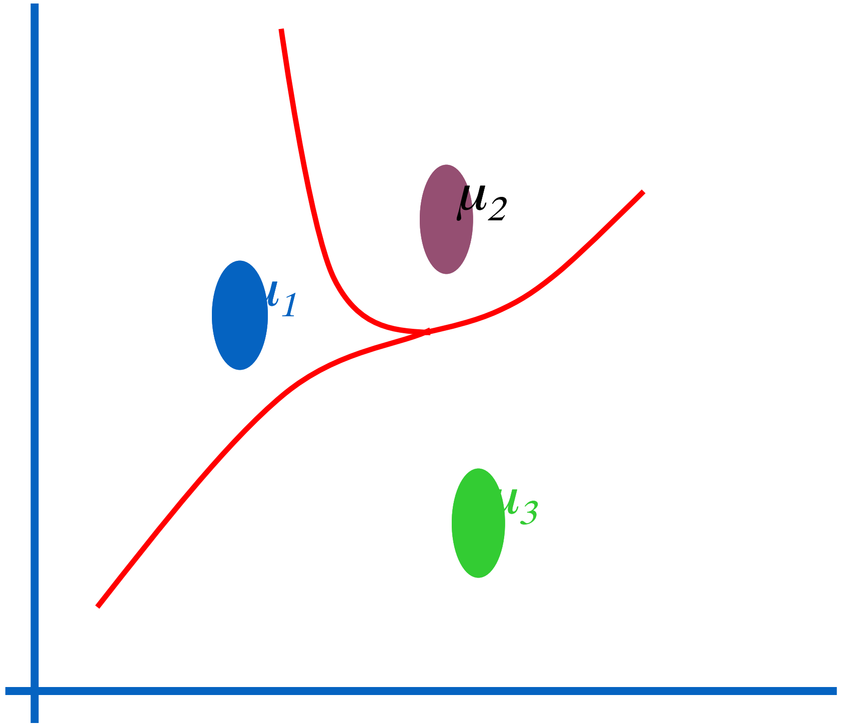
GMM – Gaussian Mixture Model (Multi-modal distribution)

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

Gaussian Bayes Classifier:

$$\log \frac{P(y=i\,|\,x)}{P(y=j\,|\,x)} = \log \frac{p(x\,|\,y=i)P(y=i)}{p(x\,|\,y=j)P(y=j)} = x^T W x + w^T x$$

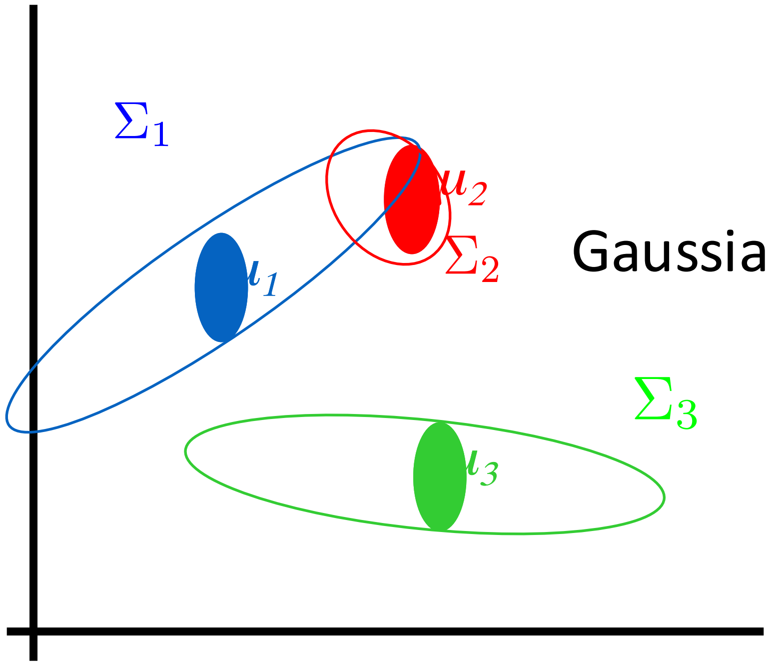
Depend on $\mu_1, \mu_2, \ldots, \mu_K, \Sigma_1, \Sigma_2, \ldots, \Sigma_K$, P(y=1),…, P(Y=k)



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

Learning General GMM

$$x_1, \ldots, x_m \sim p(x) = \sum_{i=1}^{k} \underbrace{p(x|Y=i)}_{\text{Mixture component}} \underbrace{P(Y=i)}_{\text{Mixture proportion, } p_i}$$



Gaussian mixture model

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

Parameters: $\{p_i, \mu_i, \Sigma_i\}_{i=1}^{K}$

- How to estimate parameters? Maximum Likelihood
  But don't know labels Y (recall Gaussian Bayes classifier)

Learning General GMM

Maximize marginal likelihood:

$\arg\max \prod_j P(x_j) = \arg\max \prod_j \sum_{i=1}^{K} P(y_j=i,x_j)$ ... marginalizing y_j

$= \arg\max \prod_j \sum_{i=1}^{K} P(y_j=i)p(x_j|y_j=i)$

$P(y_j=i) = P(y=i)$ Mixture component i is chosen with prob $P(y = i)$

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

How do we find the $\boldsymbol{\mu}_i$'s and P(y=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: Doable, but often slow

GMM vs. k-means

Maximize marginal likelihood:

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

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

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

$$P(y_j = i) = 1 \text{ if } i = C(j)$$

$$= 0 \text{ otherwise}$$

$$\text{argmax} \prod_j P(x_j) = \text{argmax} \prod_j p(x_j | y_j=C(j))$$

Same as k-means!

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

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

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

- No need to choose step size as in Gradient methods.

- EM is an Iterative algorithm with two linked steps:

  E-step: fill-in hidden data (Y) using inference

  M-step: apply standard MLE/MAP method to estimate parameters

  $\{p_i, \mu_i, \Sigma_i\}^k_{i=1}$

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

EM for spherical, same variance GMMs

E-step

Compute "expected" classes of all datapoints for each class

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

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

EM does soft assignment

M-step

Compute MLE for $\boldsymbol{\mu}$ given our data's class membership distributions (weights)

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

Similar to K-means, but with
weighted data

Iterate.

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)

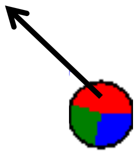
$$\mu_i^{(t+1)} = \frac{\sum_j P\left(y = i \middle| x_j, \lambda_t\right) x_j}{\sum_j P\left(y = i \middle| x_j, \lambda_t\right)} \qquad \Sigma_i^{(t+1)} = \frac{\sum_j P\left(y = i \middle| x_j, \lambda_t\right)\left(x_j - \mu_i^{(t+1)}\right)\left(x_j - \mu_i^{(t+1)}\right)^T}{\sum_j P\left(y = i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

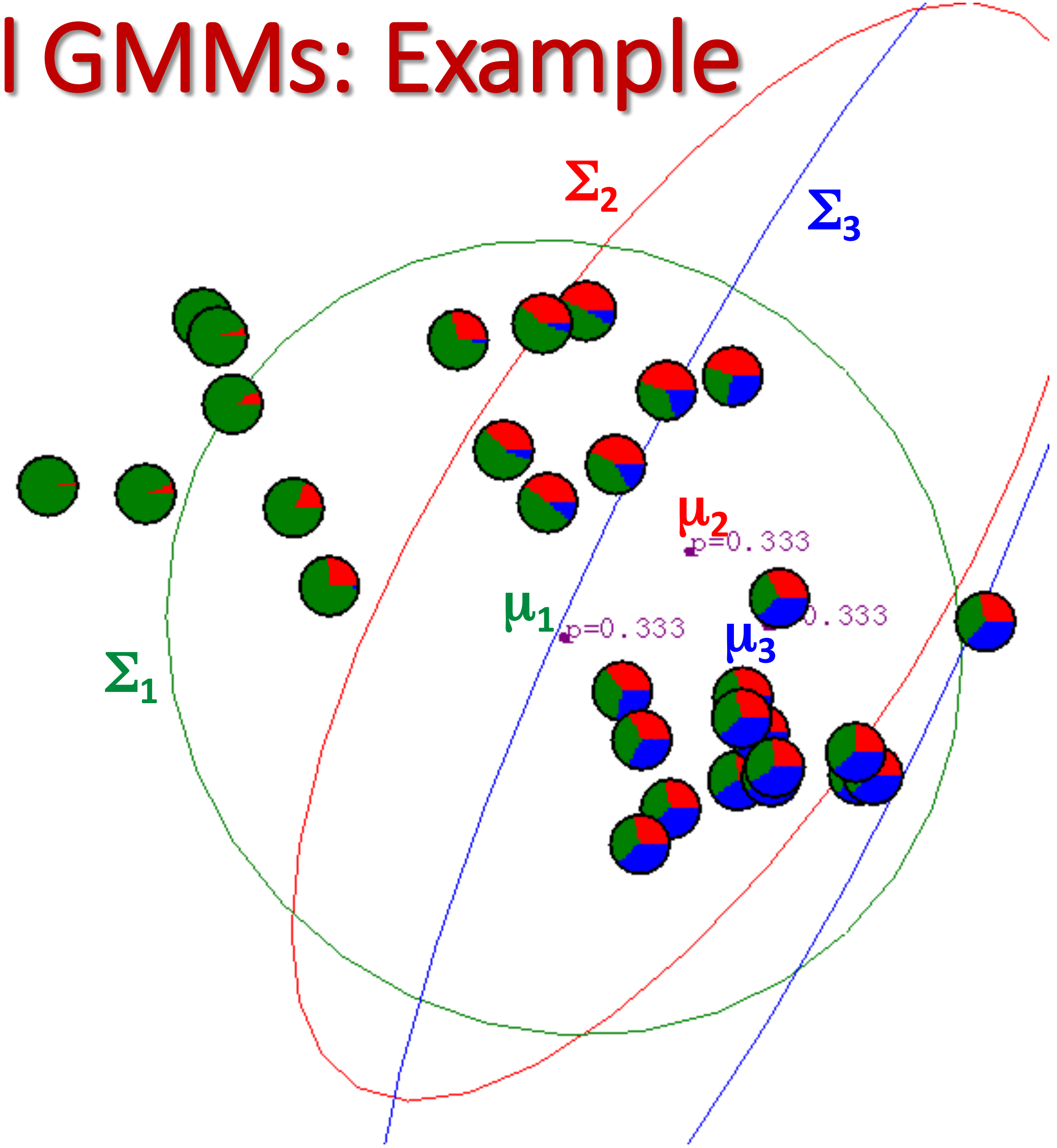
$$p_i^{(t+1)} = \frac{\sum_j P\left(y = i \middle| x_j, \lambda_t\right)}{m}$$

m = #data points

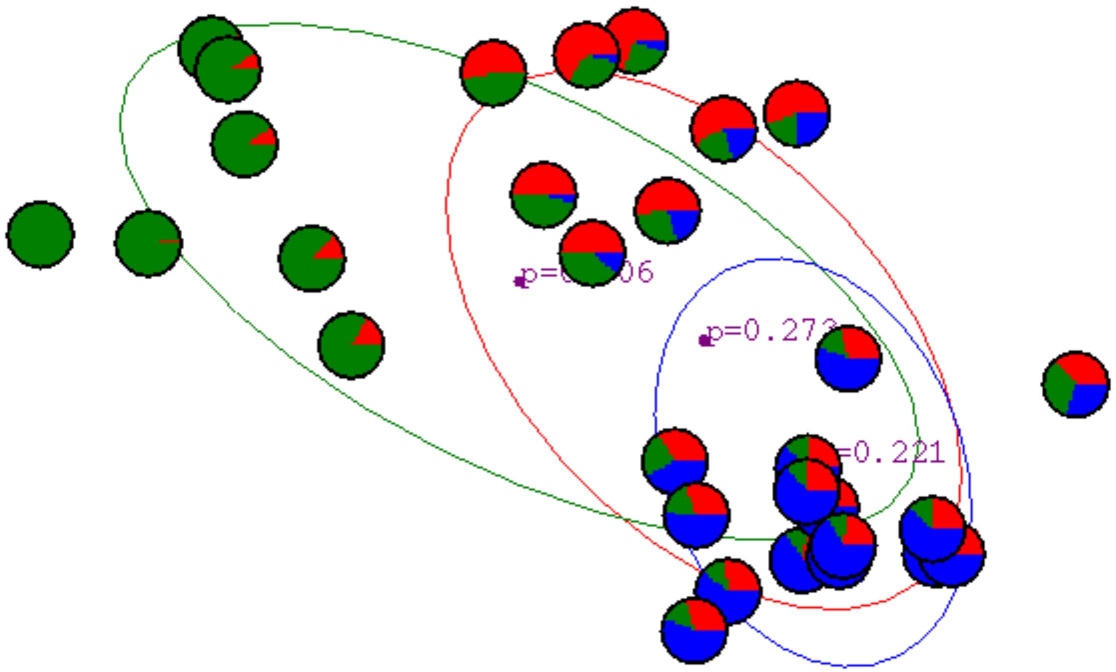
EM for general GMMs: Example

$P(y = \bullet \mid x_j, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3, p_1, p_2, p_3)$

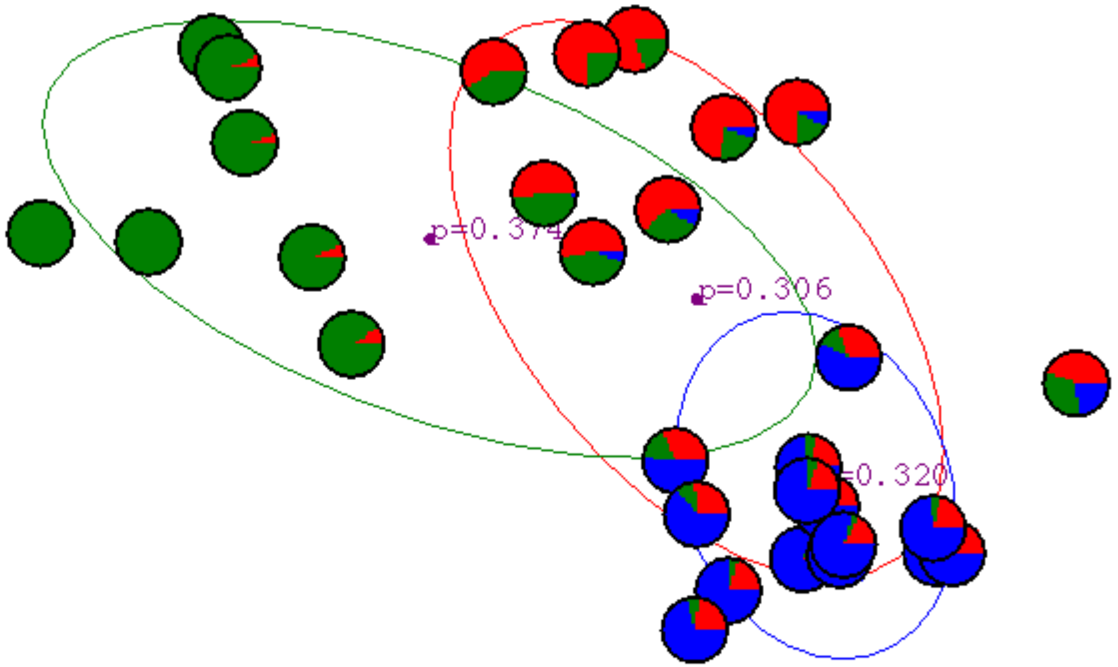




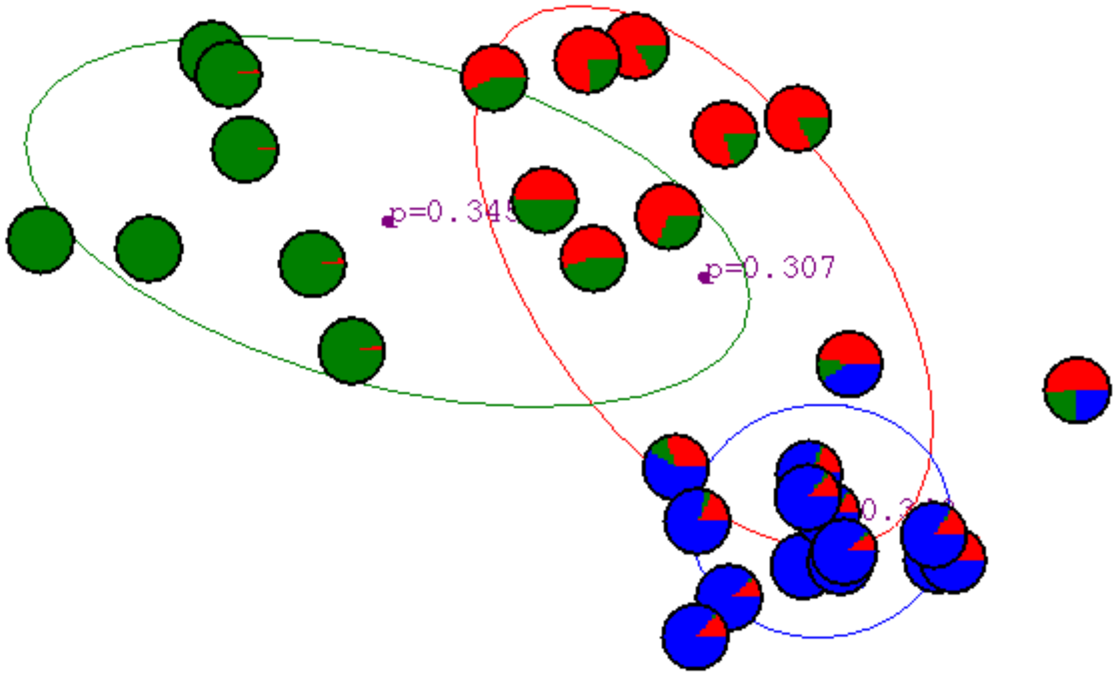
After 1st iteration



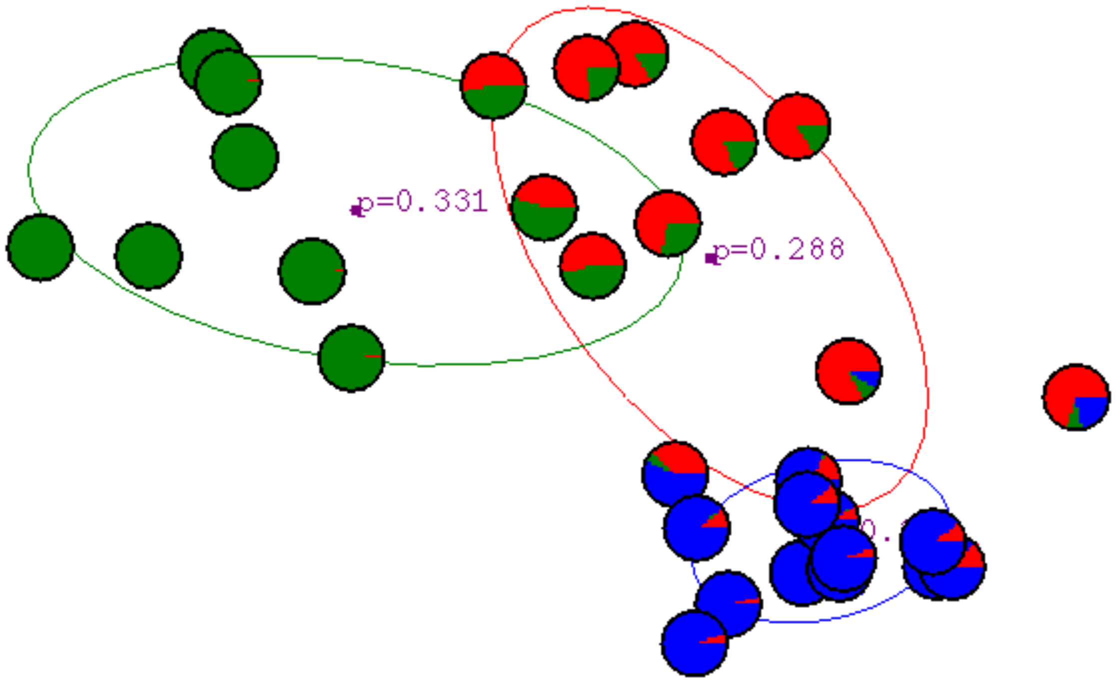
After 2nd iteration



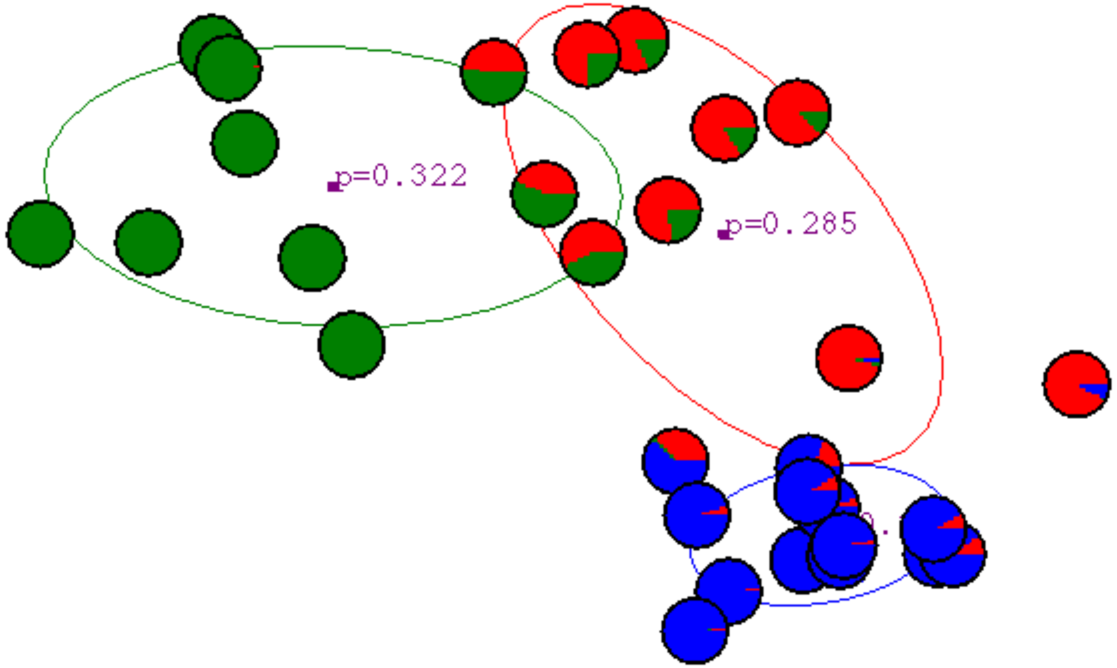
After 3rd iteration



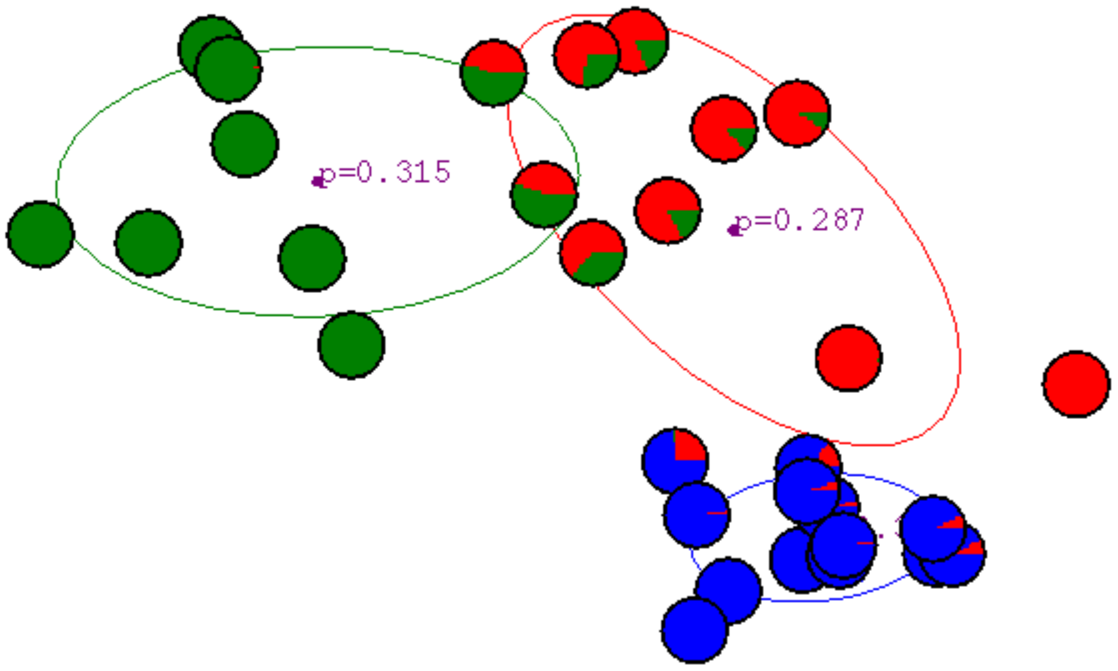
After 4th iteration



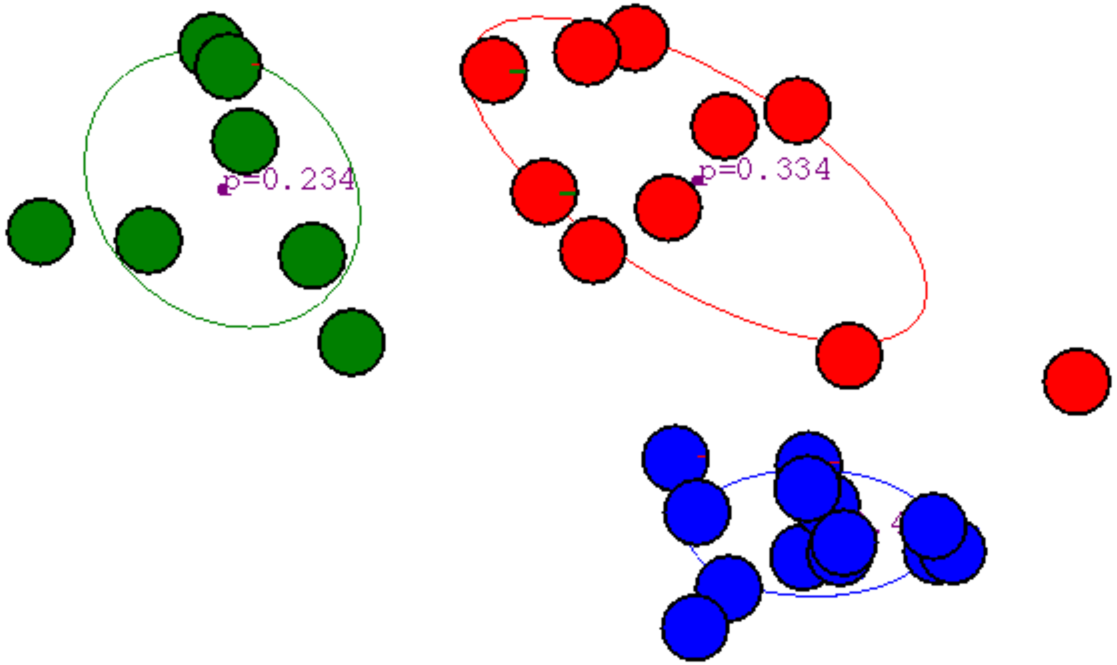
After 5th iteration



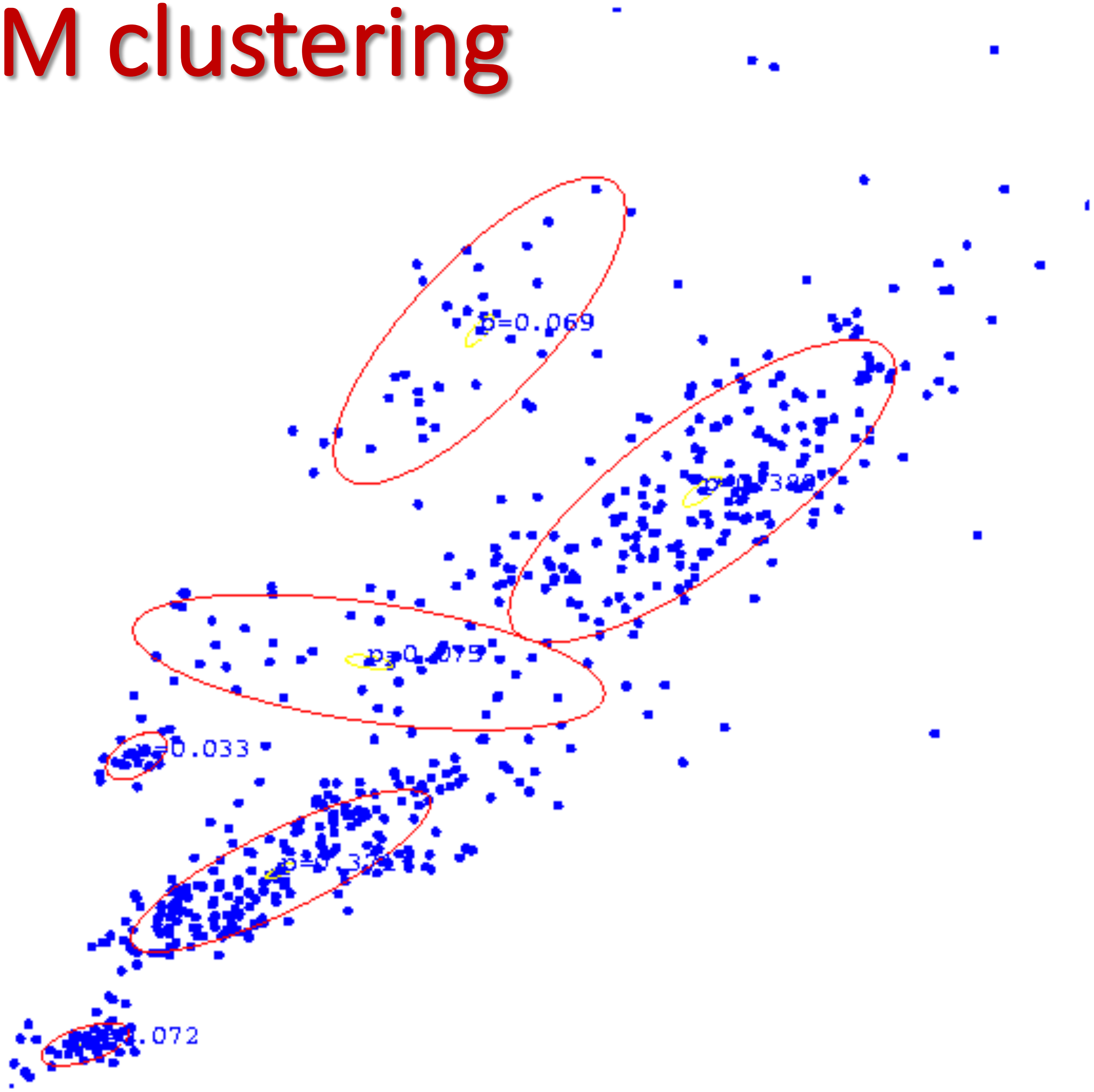
After 6th iteration



After 20th iteration



Example: GMM clustering

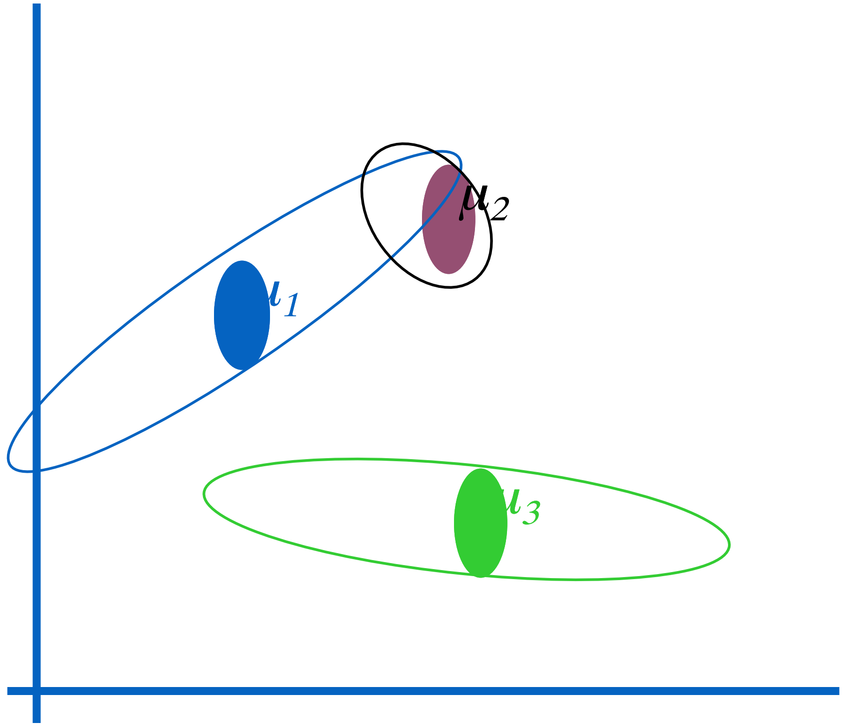


General GMM

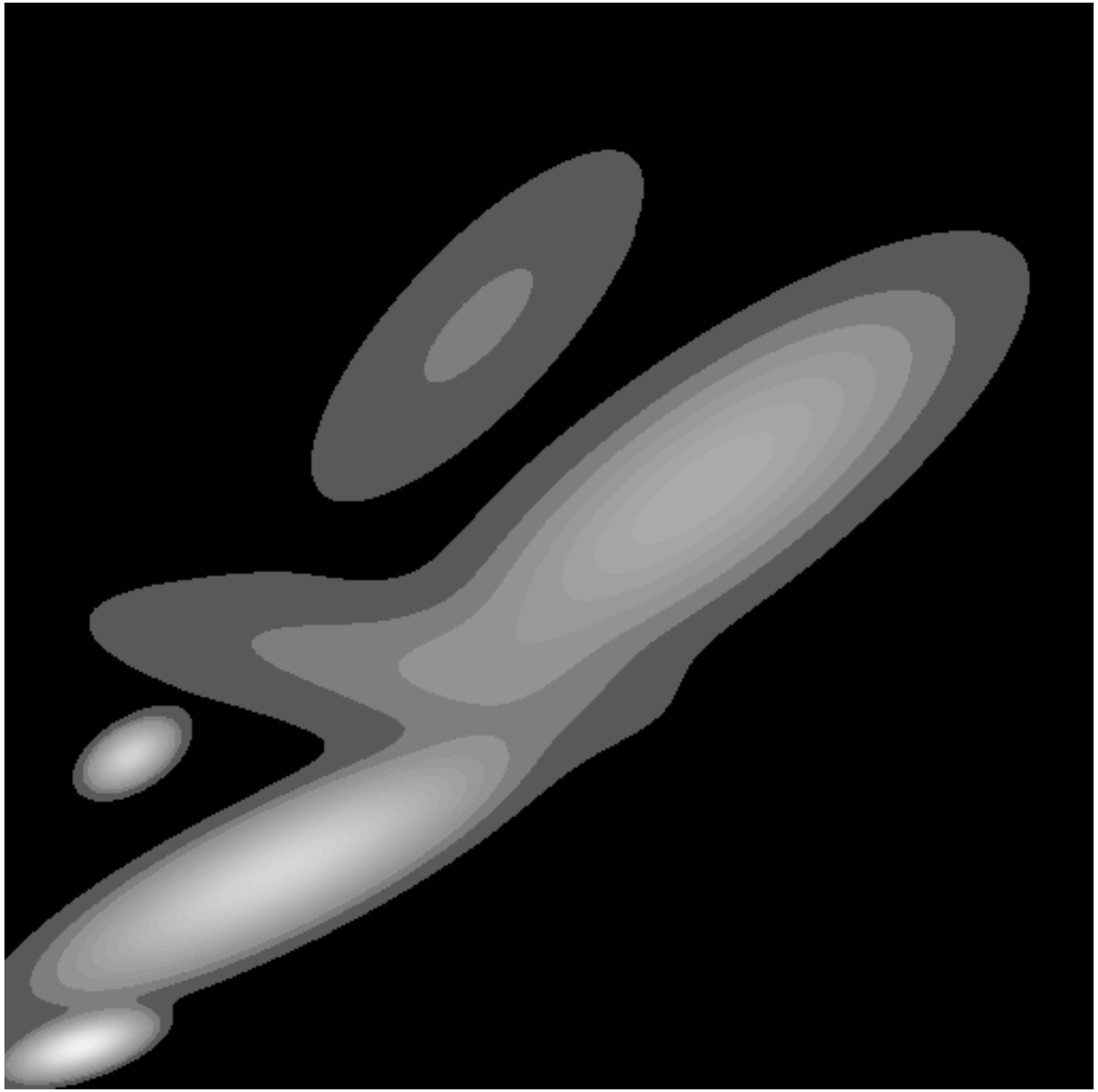
GMM – Gaussian Mixture Model (Multi-modal distribution)

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

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



Resulting Density Estimator



General EM algorithm

Marginal likelihood – **x** is observed, **z** is missing:

$$
\begin{aligned}
\log \mathrm{P}(\mathrm{D};\theta) &= \log \prod_{j=1}^{m} P(\mathbf{x}_j \mid \theta) \\
&= \sum_{j=1}^{m} \log P(\mathbf{x}_j \mid \theta) \\
&= \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{x}_j, \mathbf{z} \mid \theta)
\end{aligned}
$$

$$\mathrm{D} = \{\mathrm{x}_j\}_{j=1}^{m}$$

θ - model parameter(s)

How to maximize marginal likelihood using EM?

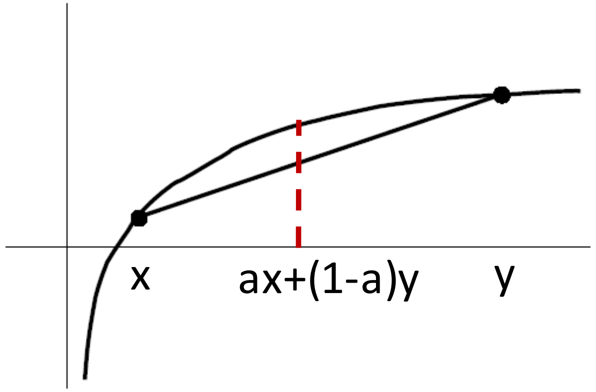
Lower-bound on marginal likelihood

$$\log P(D;\theta) = \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{x}_j, \mathbf{z} \mid \theta)$$

$$= \sum_{j=1}^{m} \log \sum_{\mathbf{z}} \underbrace{Q(\mathbf{z} \mid \mathbf{x}_j)}_{P(\mathbf{z})} \underbrace{\frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta)}{Q(\mathbf{z} \mid \mathbf{x}_j)}}_{f(\mathbf{z})}$$

Variational approach

Jensen's inequality: $\log \sum_{\mathbf{z}} P(\mathbf{z})\, f(\mathbf{z}) \geq \sum_{\mathbf{z}} P(\mathbf{z}) \log f(\mathbf{z})$



log: concave function

log(ax+(1-a)y) ≥ a log(x) + (1-a) log(y)

Lower-bound on marginal likelihood

$$\log \mathrm{P}(\mathrm{D};\theta) = \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{x}_j, \mathbf{z} \mid \theta)$$

$$= \sum_{j=1}^{m} \log \sum_{\mathbf{z}} \underbrace{Q(\mathbf{z} \mid \mathbf{x}_j)}_{\mathrm{P}(\mathbf{z})} \underbrace{\frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta)}{Q(\mathbf{z} \mid \mathbf{x}_j)}}_{\mathrm{f}(\mathbf{z})}$$

Jensen's inequality: $\log \sum_{\mathbf{z}} \mathrm{P}(\mathbf{z})\, \mathrm{f}(\mathbf{z}) \geq \sum_{\mathbf{z}} \mathrm{P}(\mathbf{z}) \log \mathrm{f}(\mathbf{z})$

$$\geq \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta)}{Q(\mathbf{z} \mid \mathbf{x}_j)} \quad =: \quad F(\theta, Q)$$

EM as Coordinate Ascent

$$\log \mathrm{P}(\mathrm{D};\theta) \ \geq \ F(\theta, Q)$$

E-step: Fix θ, maximize F over Q

$$\mathrm{Q}^{t+1} = \arg\max_{\mathrm{Q}} \mathrm{F}(\theta^{t}, \mathrm{Q})$$

M-step: Fix Q, maximize F over θ

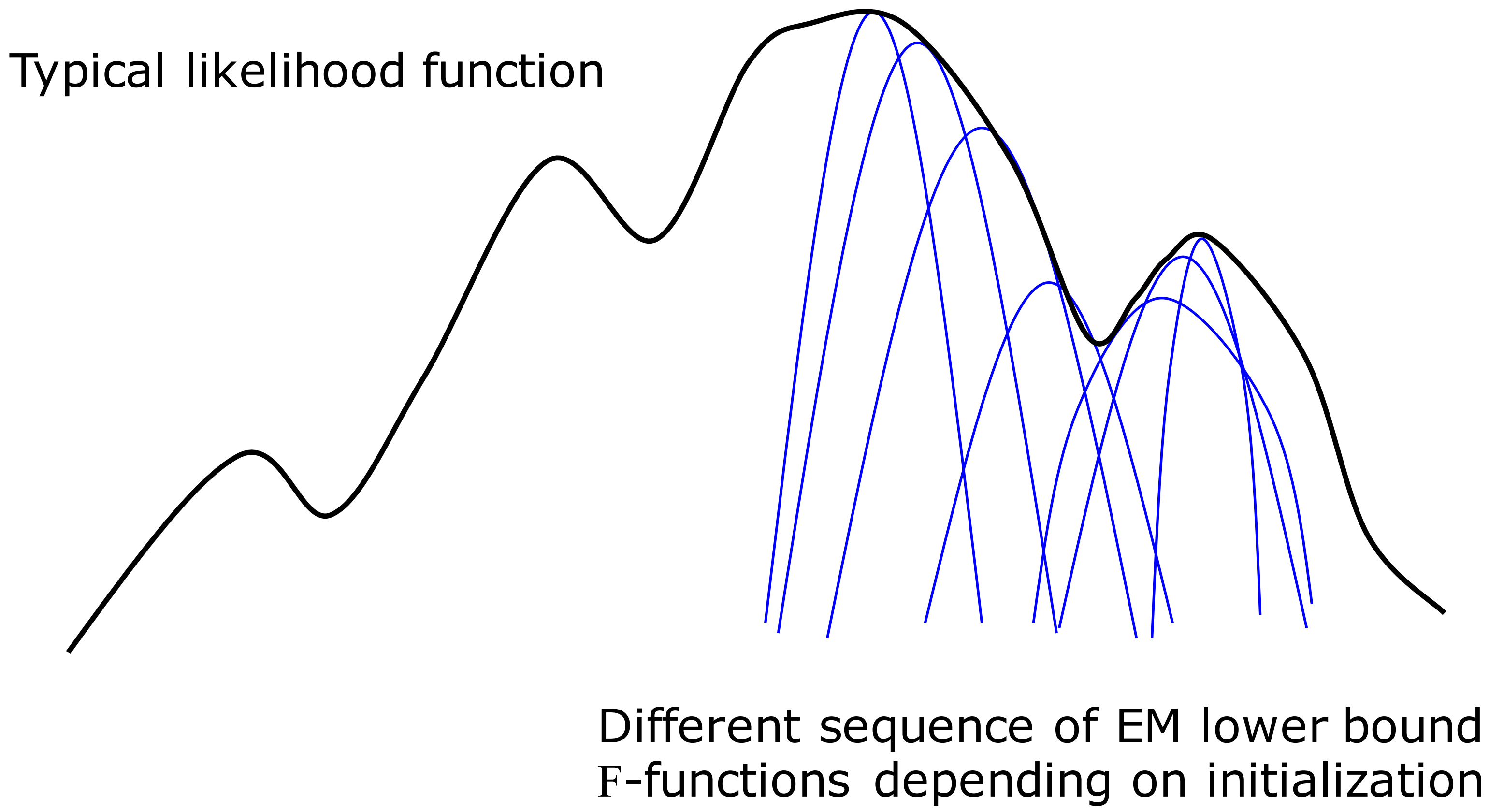
$$\theta^{t+1} = \arg\max_{\theta} \mathrm{F}(\theta, \mathrm{Q}^{t+1})$$

Convergence of EM

Sequence of EM lower bound F-functions

EM monotonically converges to a local maximum of likelihood !

EM & Local Maxima



Use multiple, randomized initializations in practice

EM as Coordinate Ascent

$$\log \mathrm{P}(\mathrm{D};\theta) \geq F(\theta, Q)$$

E-step: Fix θ, maximize F over Q

$$\mathrm{Q}^{t+1} = \arg\max_{\mathrm{Q}} \mathrm{F}(\theta^{t}, \mathrm{Q})$$

M-step: Fix Q, maximize F over θ

$$\theta^{t+1} = \arg\max_{\theta} \mathrm{F}(\theta, \mathrm{Q}^{t+1})$$

E step

$$\log \mathrm{P}(\mathrm{D};\theta) \;\geq\; F(\theta, Q)$$

E-step: Fix θ, maximize F over Q

$$\log \mathrm{P}(\mathrm{D};\theta^{(t)}) \;\geq\; F(\theta^{(t)}, Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)})}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

$$= \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})\; P(\mathbf{x}_j|\theta^{(t)})}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

$$= \underbrace{\sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})}{Q(\mathbf{z} \mid \mathbf{x}_j)}}_{-KL(Q(\mathbf{z}|\mathbf{x}_j),\, P(\mathbf{z}|\mathbf{x}_j, \theta^{(t)}))} + \underbrace{\sum_{j=1}^{m} \cancelto{1}{\sum_{\mathbf{z}} Q(\mathbf{z}|\mathbf{x}_j)} \log P(\mathbf{x}_j|\theta^{(t)})}_{\log \mathrm{P}(\mathrm{D};\theta^{(t)})}$$

KL divergence between two distributions

E step

$$\log \mathrm{P(D;\theta)} \geq F(\theta, Q)$$

E-step: Fix θ, maximize F over Q

$$\log \mathrm{P(D;\theta^{(t)})} \geq F(\theta^{(t)}, Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)})}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

$$= \sum_{j=1}^{m} -KL(Q(\mathbf{z}|\mathbf{x}_j), P(\mathbf{z}|\mathbf{x}_j, \theta^{(t)})) + \log \mathrm{P(D;\theta^{(t)})}$$

KL>=0, above expression is maximized if KL divergence = 0

KL(Q,P) = 0 iff Q = P

Therefore,

E step: $Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) = P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})$

E step

$$\log\ \mathrm{P}(\mathrm{D};\theta)\ \geq\ F(\theta, Q)$$

E-step: Fix θ, maximize F over Q

$$\log \mathrm{P}(\mathrm{D};\theta^{(t)}) \geq\ F(\theta^{(t)}, Q) = \sum_{j=1}^{m} -KL(Q(\mathbf{z}|\mathbf{x}_j), P(\mathbf{z}|\mathbf{x}_j, \theta^{(t)})) + \log\ \mathrm{P}(\mathrm{D};\theta^{(t)})$$

$$\rightarrow\quad Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j)\ =\ P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})$$

Compute probability of missing data z given current choice of θ

Re-aligns F with marginal likelihood !!

$$F(\theta^{(t)}, Q^{(t+1)})\ =\ \log \mathrm{P}(\mathrm{D};\theta^{(t)})$$

M step

$$\log \mathrm{P(D;\theta)} \geq F(\theta, Q)$$

M-step: Fix Q, maximize F over θ

$$\log \mathrm{P(D;\theta)} \geq F(\theta, Q^{(t+1)}) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta)}{Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j)}$$

$$= \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta) + \underbrace{\sum_{j=1}^{m} H(Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j))}_{\text{Fixed (Independent of } \theta)}$$

$$\parallel$$

$$\underbrace{\sum_{\mathbf{z}} \boxed{\sum_{j=1}^{m} \log P(\mathbf{z}, \mathbf{x}_j \mid \theta)} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j)}_{\text{Expected log likelihood wrt Q}}$$

Log likelihood if z was known

M step

$$\log \mathrm{P}(\mathrm{D};\theta) \geq F(\theta, Q)$$

M-step: Fix Q, maximize F over θ

$$\log \mathrm{P}(\mathrm{D};\theta) \geq F(\theta, Q^{(t+1)}) = \sum_{\mathbf{z}} \sum_{j=1}^{m} \log P(\mathbf{z}, \mathbf{x}_j \mid \theta) Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) + \sum_{j=1}^{m} \underbrace{H(Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j))}_{\text{Fixed (Independent of } \theta)}$$

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \underbrace{\sum_{\mathbf{z}} \sum_{j=1}^{m} \log P(\mathbf{z}, \mathbf{x}_j \mid \theta) Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j)}_{\text{Expected log likelihood wrt } Q^{(t+1)}}$$

EM as Coordinate Ascent

$$\log \mathrm{P}(\mathrm{D};\theta) \geq F(\theta, Q)$$

E-step: Fix θ, maximize F over Q

$$\mathrm{Q}^{t+1} = \arg\max_{\mathrm{Q}} \mathrm{F}(\theta^{t}, \mathrm{Q})$$

$$Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) = P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})$$

E.g., $\mathrm{P}\left(y = i \middle| x_j, \mu_t\right)$

Compute probability of missing data given current choice of θ

M-step: Fix Q, maximize F over θ

$$\theta^{t+1} = \arg\max_{\theta} \mathrm{F}(\theta, \mathrm{Q}^{t+1})$$

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{\mathbf{z}} \sum_{j=1}^{m} \log P(\mathbf{z}, \mathbf{x}_j \mid \theta) Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j)$$

Compute estimate of θ by maximizing marginal likelihood using $Q^{(t+1)}(z|x_j)$

Summary: EM Algorithm

- A way of maximizing likelihood function for hidden variable models. Finds MLE of parameters when the original (hard) problem can be broken up into two (easy) pieces:
  1. Estimate some “missing” or “unobserved” data from observed data and current parameters.
  2. Using this “complete” data, find the maximum likelihood parameter estimates.

- Alternate between filling in the latent variables using the best guess (posterior) and updating the parameters based on this guess:
  1. E-step: $Q^{t+1} = \arg\max_{Q} F(\theta^{t}, Q)$
  2. M-step: $\theta^{t+1} = \arg\max_{\theta} F(\theta, Q^{t+1})$

- In the M-step we optimize a lower bound on the likelihood. In the E-step we close the gap, making bound=likelihood.

- EM performs coordinate ascent on F, can get stuck in local minima.

- BUT Extremely popular in practice.