Announcements

Assignments

- HW9 (online)
  - Due Thu 4/16, 11:59 pm

TA Applications:

- Please apply to TA with us! Link for MLD applications will be in piazza.

Participation points

- Starting now, we're capping the denominator (57 polls) in the participation points calculation

$$\frac{N_{answered}}{N=57} \geq 80\% \qquad 5\%$$

Introduction to Machine Learning

Clustering
GMM and EM

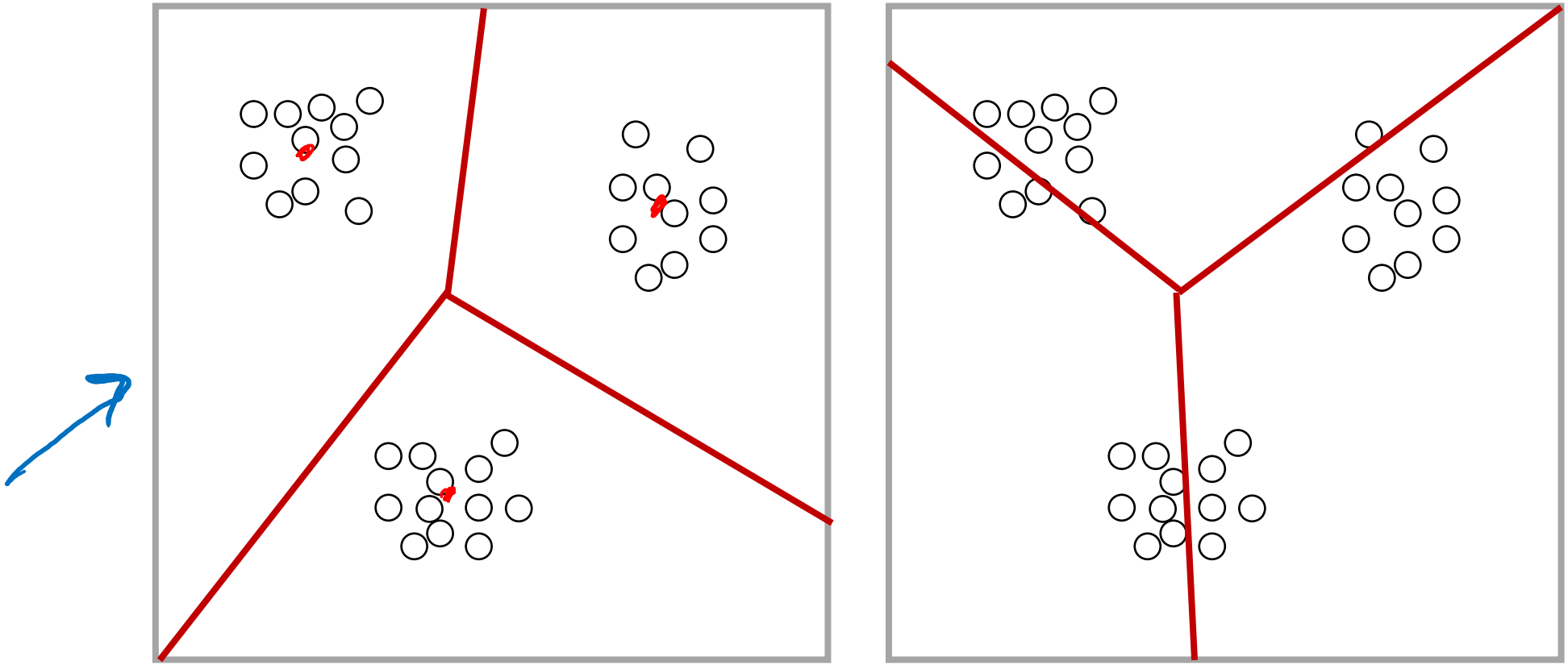
Instructor: Pat Virtue

Slide credits: Aarti Singh, Eric Xing, Carlos Guestrin

K-means Optimization

Question: Which of these partitions is “better”?

min



Slide credit: CMU MLD Matt Gormley

K-means Optimization

Input: $\vec{x}^{(1)} \dots \vec{x}^{(N)}$ K $x^{(i)} \in \mathbb{R}^M$

$z^{(1)} \dots z^{(N)}$

Output: $C_1, \dots C_k$ C $\vec{c}_k \in \mathbb{R}^M$

$$C = \underset{C}{\operatorname{argmin}} \sum_{i=1}^{N} \min_{j \in \{1..K\}} \|\vec{x}^{(i)} - \vec{c}_j\|_2^2$$

$$C, \vec{z} = \underset{C}{\operatorname{argmin}} \sum_{i=1}^{N} \min_{z^{(i)} \in \{1..K\}} \|\vec{x}^{(i)} - \vec{c}_{z^{(i)}}\|_2^2$$

$$= \underset{C, \vec{z}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\vec{x}^{(i)} - \vec{c}_{z^{(i)}}\|_2^2$$

$z^{(i)} \in \{1 \dots K\}$

Slide credit: CMU MLD Matt Gormley

K-means Optimization

Alternating minimization

a) $\vec{z} = \underset{z}{\operatorname{argmin}} \sum_{i=1}^{N} \|x^i - c_{z^i}\|_2^2$

b) $C = \underset{C}{\operatorname{argmin}} \sum_{i=1}^{N} \|x^i - c_{z^i}\|_2^2$

---

a) $z^{(1)} = \underset{k}{\operatorname{argmin}} \|x^{(1)} - c_k\|_2^2$

$\vdots$

$z^{(N)} = \quad "$

b) $\vec{c}_1 = \underset{c_1}{\operatorname{argmin}} \sum_{i=z^{(i)}=1} \|x^{(i)} - c_1\|_2^2$

$\vdots$

$\vec{c}_K = \underset{c_K}{\operatorname{argmin}} \sum_{i=z^{(i)}=K} \|x^{(i)} - c_k\|_2^2$

Slide credit: CMU MLD Matt Gormley

Piazza Poll 1

[True/False] The alternating minimization algorithm will find the global minimum of the k-means objective.

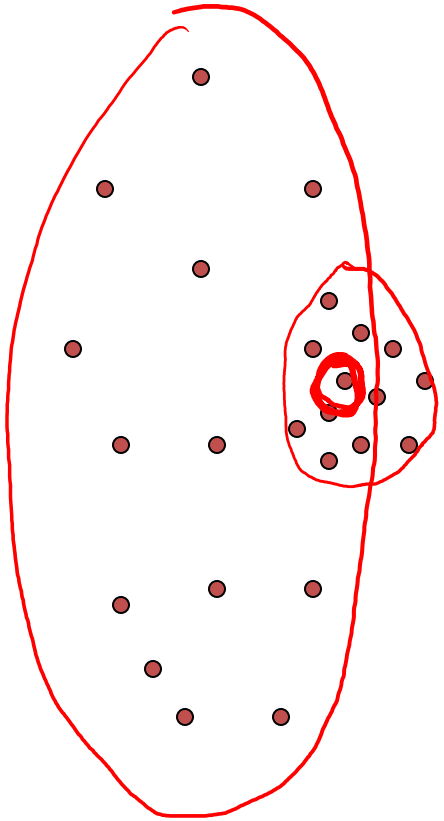
$$\boldsymbol{C}, \boldsymbol{z} = \underset{\boldsymbol{C}, \boldsymbol{z}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{c}_{z^{(i)}} \right\|_2^2$$

A. Calamity

B. True

C. False

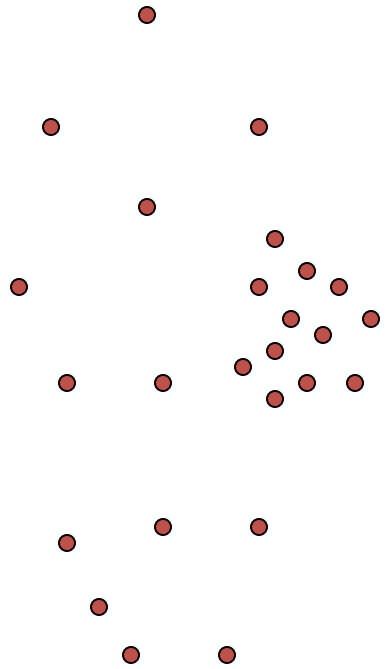
(One) bad case for K-means



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

$$p(z_{n,k}=1 \mid x_n)$$

(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

$$p(x) = \sum_y p(x, y)$$
$$= \sum_y p(x|y)\,p(y)$$

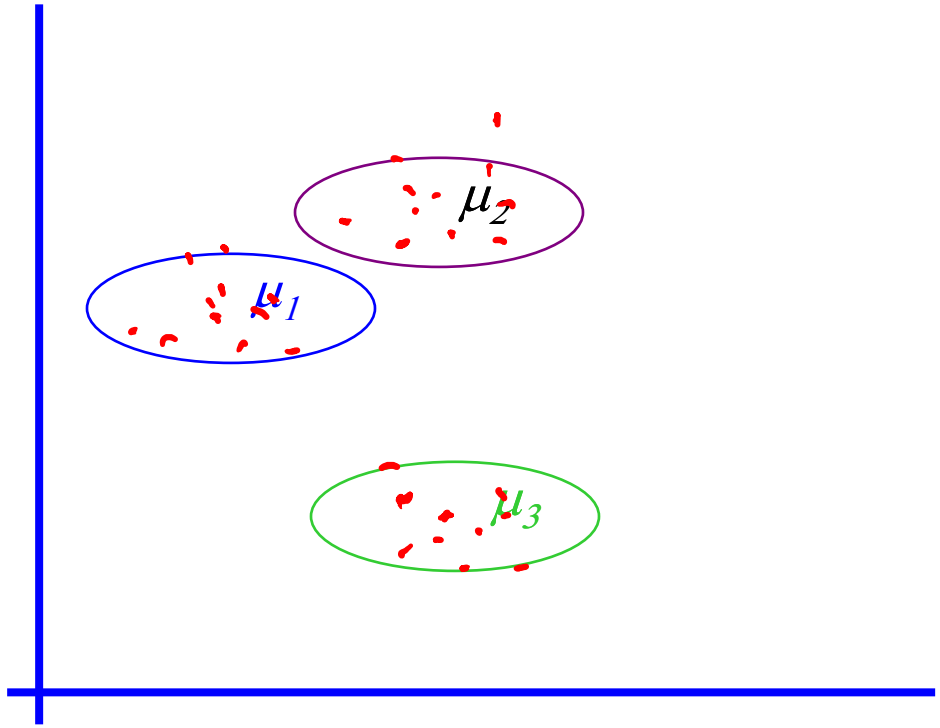
Gaussian Mixture Model

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

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

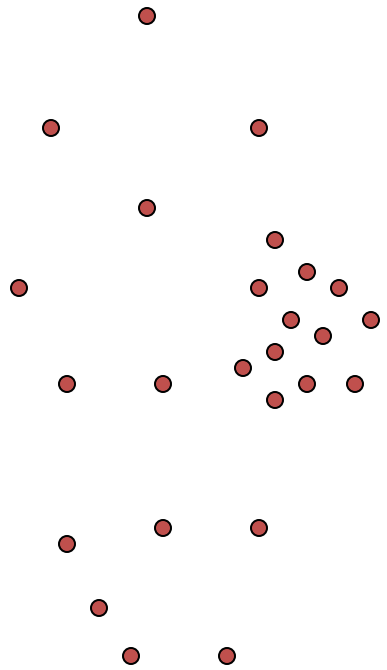
$$p(x) = \sum_i p(x|y=i)\, P(y=i)$$

Mixture component ↓ — ↓ **Mixture proportion**



$y \longrightarrow x \sim p(x|y)$

(One) bad case for K-means



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

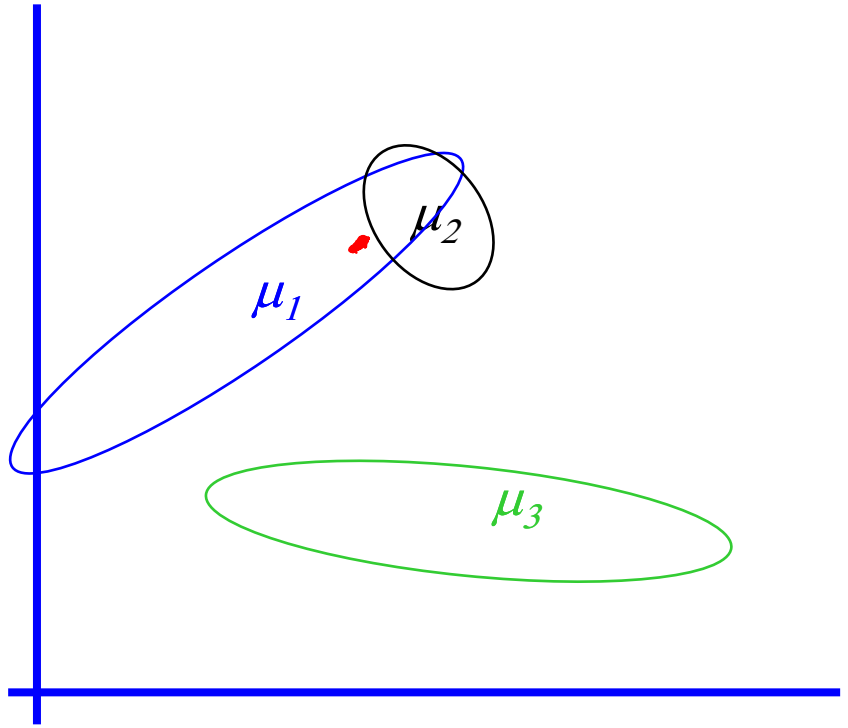
General GMM

GMM – Gaussian Mixture Model (Multi-modal distribution)

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

$$p(x) = \sum_i p(x|y=i)\, P(y=i)$$

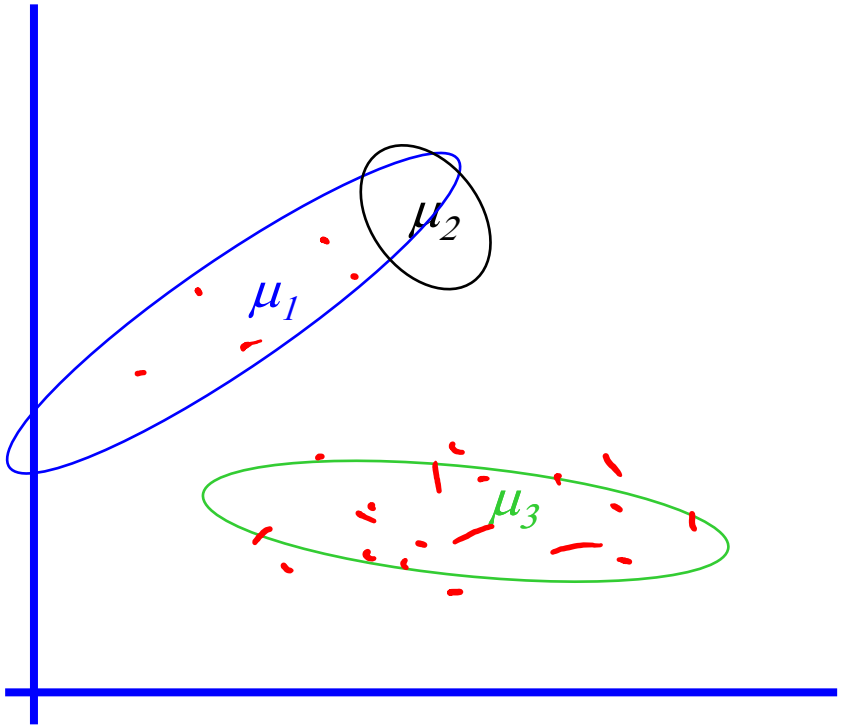
Mixture component **Mixture proportion**



General GMM

GMM – Gaussian Mixture Model (Multi-modal distribution)

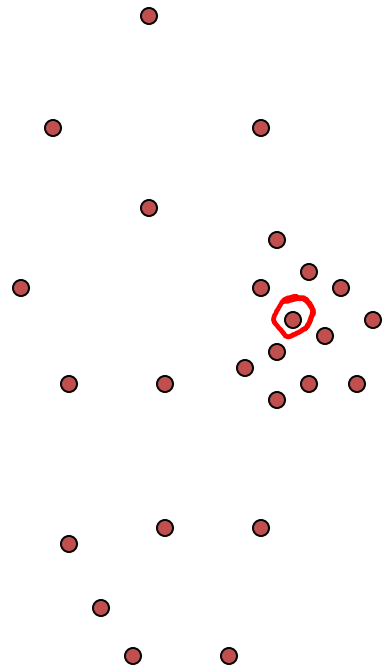
- There are k components
- Component *i* has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix Σ_i



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) Datapoint 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

$p(y=1|x)$

$p(y=2|x)$

$p(y=3|x)$

$p(y=1)$

General GMM

GMM – Gaussian Mixture Model (Multi-modal distribution)

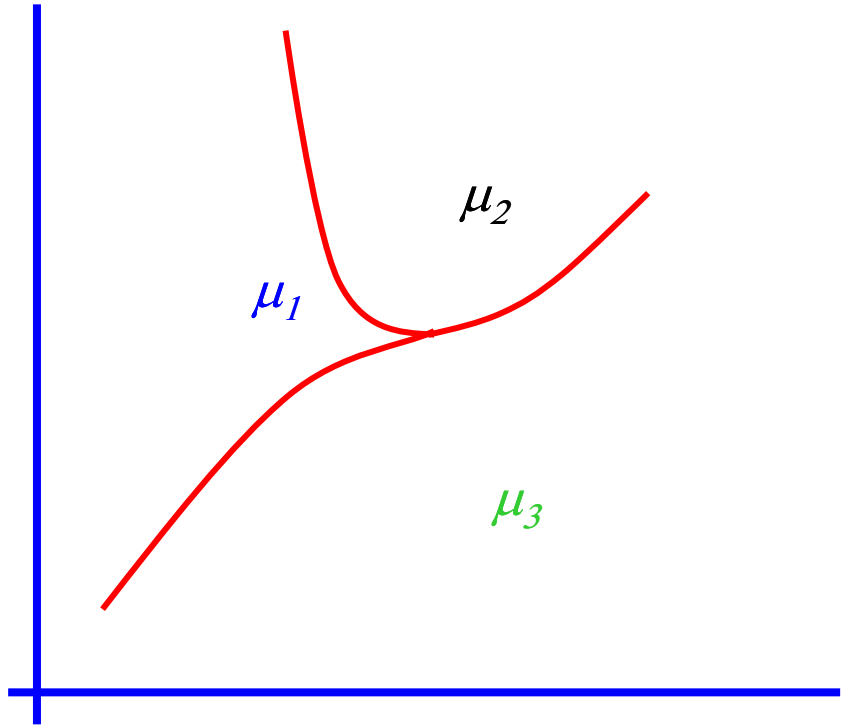
$$p(x|y=i) \sim \mathrm{N}(\mu_i, \Sigma_i)$$

Decision boundary when probabilities are equal:

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

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

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



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

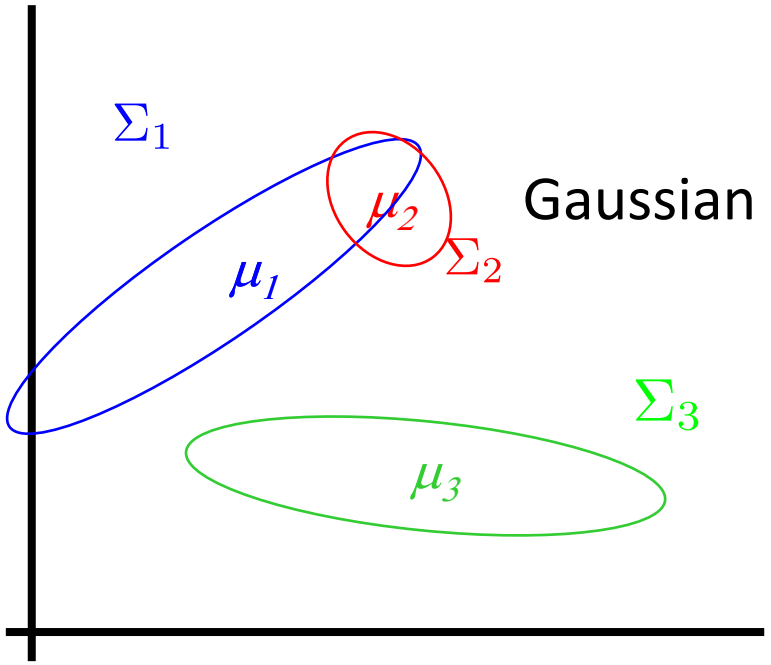
"Quadratic Decision boundary" – second-order terms don't cancel out

Learning General GMM

$$x_1, \ldots, x_m \sim p(x) = \sum_{i=1}^{k} p(x|Y=i)P(Y=i)$$

Mixture component ↓ $p(x|Y=i)$

Mixture proportion, p_i ↓ $P(Y=i)$



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? Max Likelihood
  But don't know labels Y

Learning General GMM

Maximize marginal likelihood:

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

$P(y_j=i) = P(y=i)$ 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 μ_i, Σ_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 not-analytically solvable

* Use gradient descent: Doable, but often slow

MLE Optimization

$p(D|\theta)$

MLE x

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(x|\theta)$$

$$= \prod_{i=1}^{N} p(x^{(i)}|\theta)$$

$$= \log \prod p(x^{(i)}|\theta)$$

$$= \arg\max_{\theta} \sum_{i=1}^{N} \log p(x^{(i)}|\theta)$$

M(c)LE $y|x$

$$\hat{\theta}_{ML} = \arg\max \prod p(y^{(i)}|x^{(i)}\theta)$$

$$= \prod_{i=1}^{N} p(x^{(i)}|y^{(i)}\theta)p(y^{(i)}|\theta)$$

$$= \sum_{i=1}^{N} \log\left[p(x|y\theta)\,p(y|\theta)\right]$$

$$\hat{\theta}_{ML} = \arg\max \sum_{i=1}^{N} \left[\log p(x^{i}|y^{i}\theta) + \log p(y^{i}|\theta)\right]$$

MLE Optimization

$$\hat{\theta}_{ML} = \arg\max \prod_i p(x_i)$$

$$= \arg\max \prod_{i=1}^{N} \sum_z p(x_i, z)$$

$$= \log \prod_i \sum_z p(x_i, z)$$

$$\sum_{i=1}^{N} \log \sum_z p(x_i, z)$$

$$p(z)p(x|z)$$

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**?

$P(y=1|x)$

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

$= 0$ otherwise

Same as k-means!

$$\text{argmax} \prod_j P(x_j) = \text{argmax} \prod_j p(x_j | y_j=C(j))$$
$$= \text{argmax} \prod_{j=1}^{n} \exp\left(\frac{-1}{2\sigma^2} \|x_j - \mu_{C(j)}\|^2\right)$$
$$= \text{argmin} \sum_{j=1}^{n} \|x_j - \mu_{C(j)}\|^2) = \arg\min_{\mu, C} F(\mu, C)$$

C z

z

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 $E[Y|x] \leftarrow \theta^t$

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

$\{p_i, \mu_i, \Sigma_i\}_{i=1}^k$ $\theta^{t+1} \leftarrow E[y|x]$

- 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.

$p(x|\theta^{(t)}) \leq p(x|\theta^{(t+1)} \leq p(x|\theta^{(t+2)})$

EM for spherical, same variance GMMs

E-step

Compute "expected" classes of all datapoints for each class

$$p(y_k^{(i)} = 1 \mid x, \theta^{(t)}) \leftarrow \theta^{(t)}$$

M-step

Compute Max. like $\boldsymbol{\mu}$ given our data's class membership distributions (weights)

$$\theta^{(t+1)} \leftarrow p(y_k^{(i)} = 1 \mid x, \theta^{(t)})$$

Iterate.

EM for spherical, same variance GMMs

E-step

Compute "expected" classes of all datapoints for each class

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

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 (weights)

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

Exactly same as MLE with weighted data

$$\frac{\sum_{j=1}^{M} z_i^{(j)} x_j}{\sum_{j=1}^{N} z_i^{(j)}}$$

← sum X in i

← N points in i

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

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

Just evaluate a Gaussian at x_j

M-step

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

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

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

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

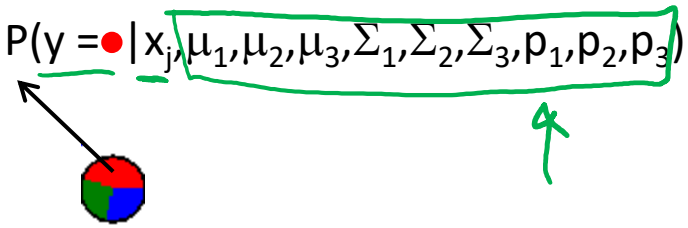
m = #data points

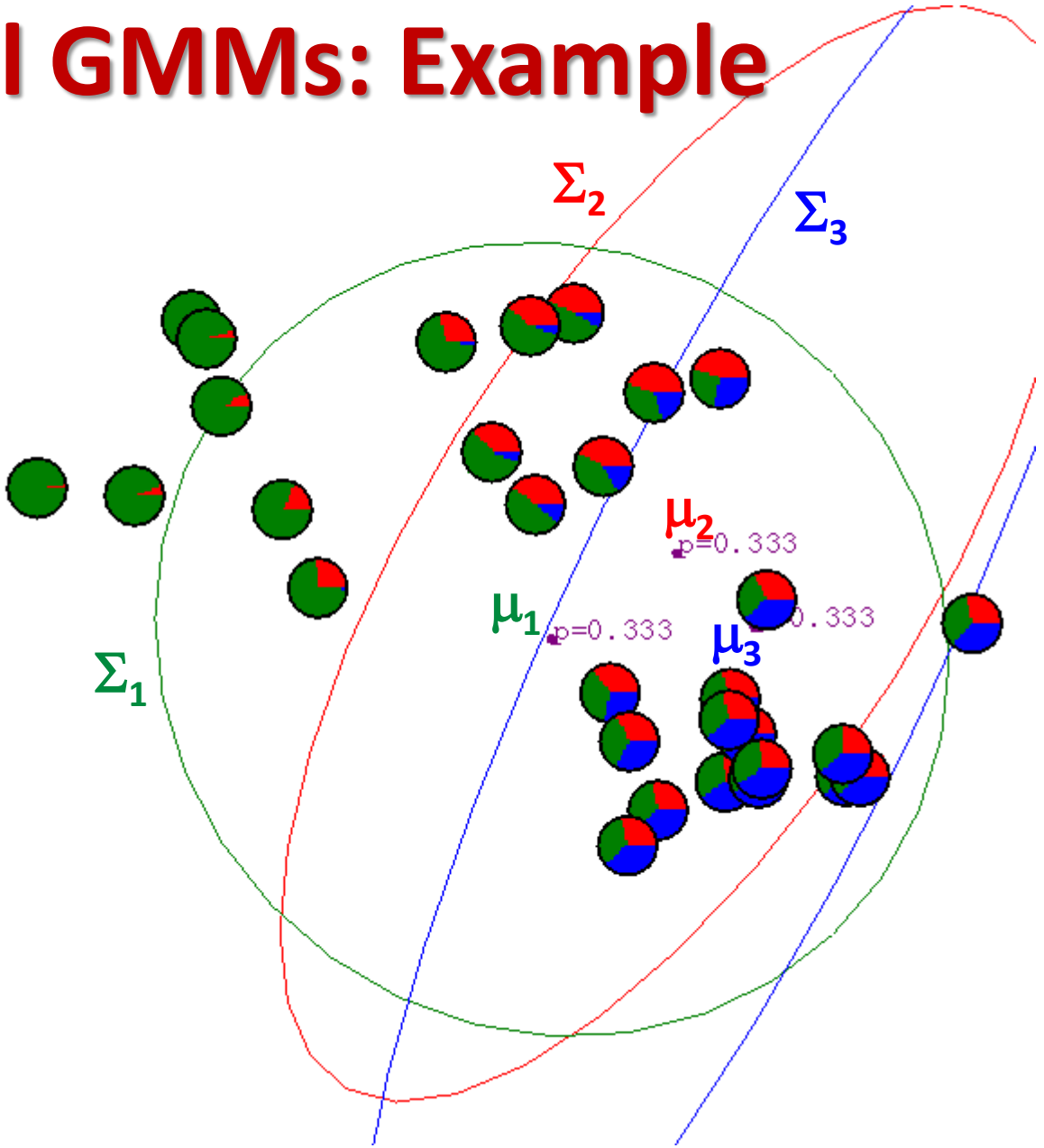
EM Convergence

$$p(x \mid \lambda^{(t)}) \leq p(x \mid \lambda^{(t+1)})$$

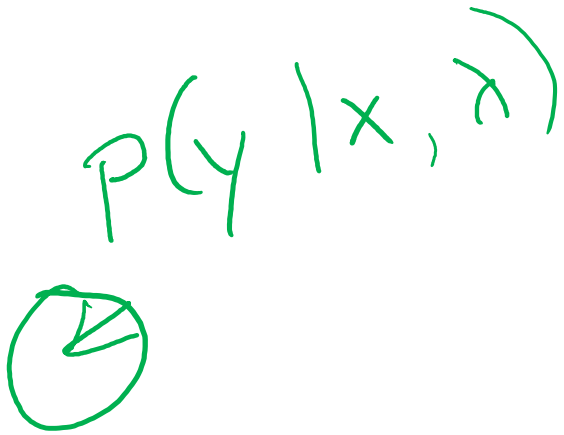
$$p(x \mid \lambda^{*})$$

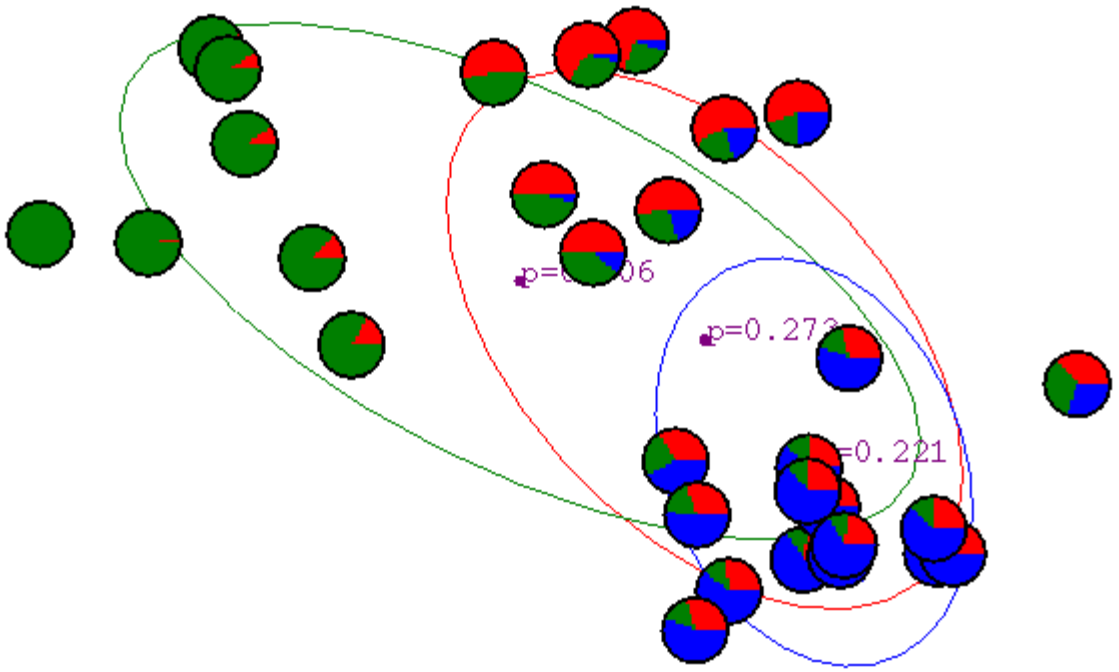
EM for general GMMs: Example



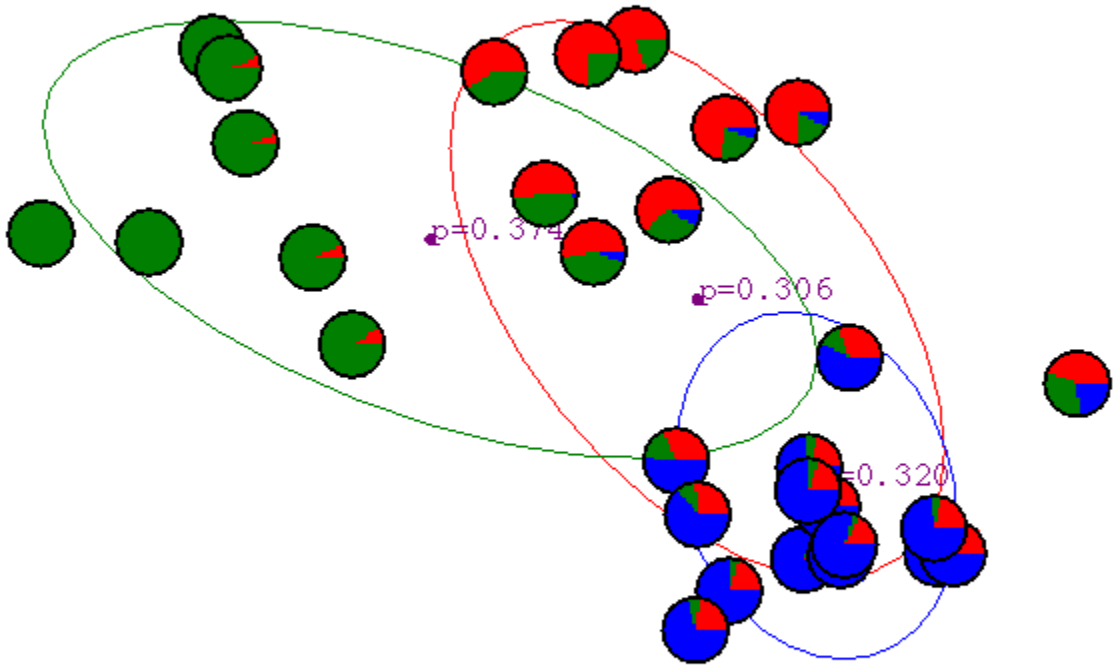


After 1st iteration

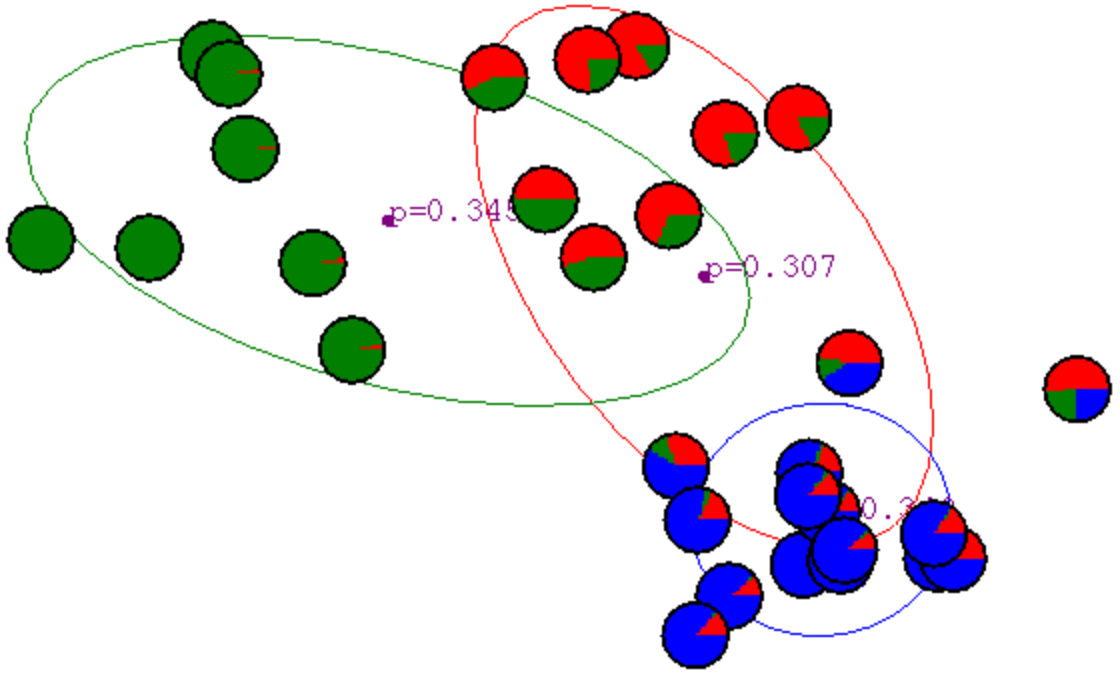




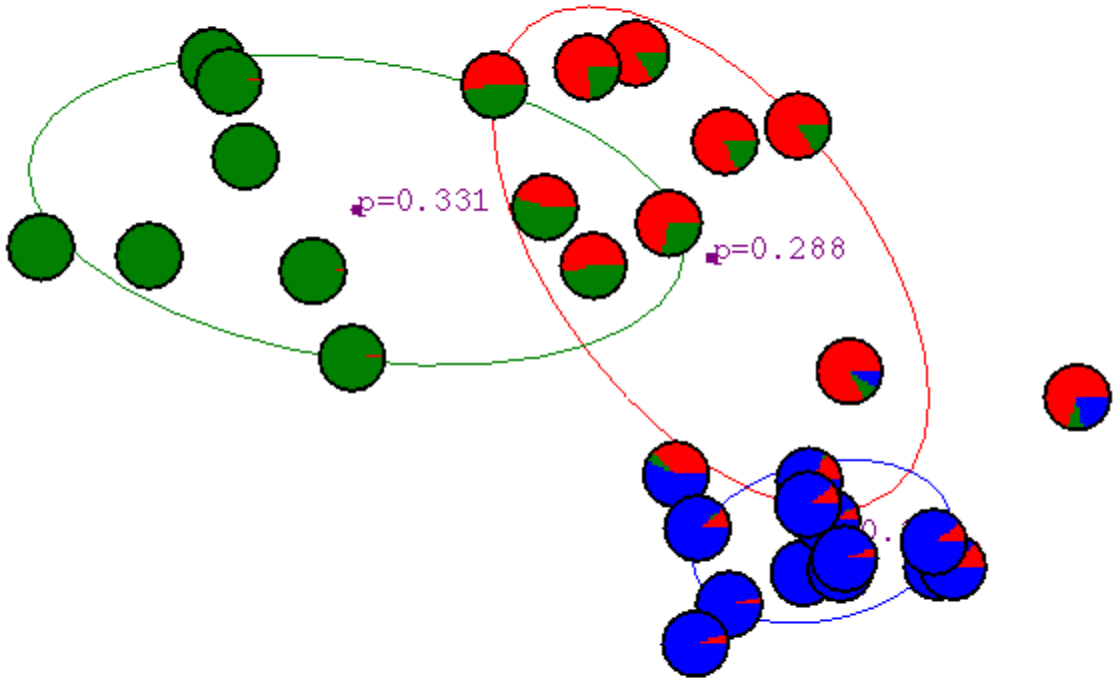
After 2nd iteration



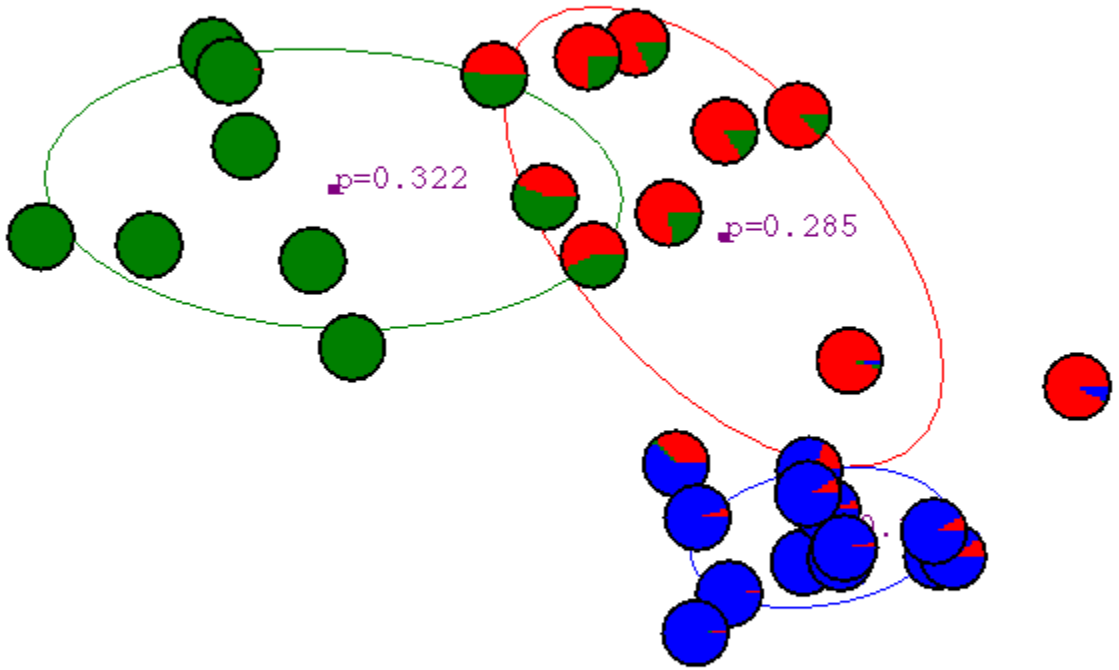
After 3rd iteration



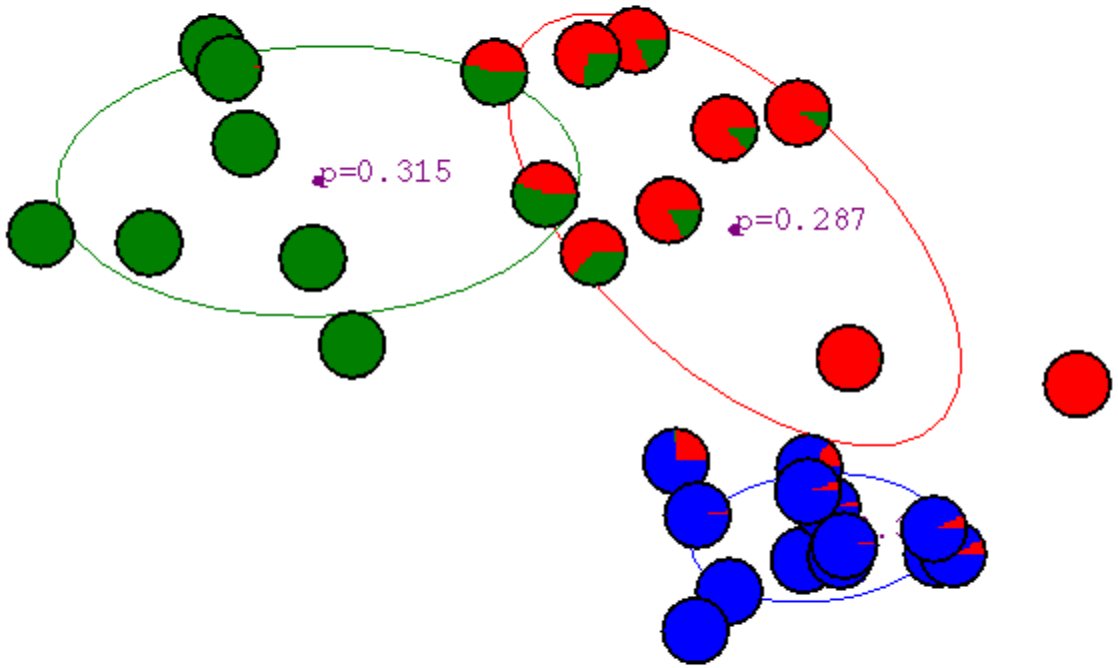
After 4th iteration



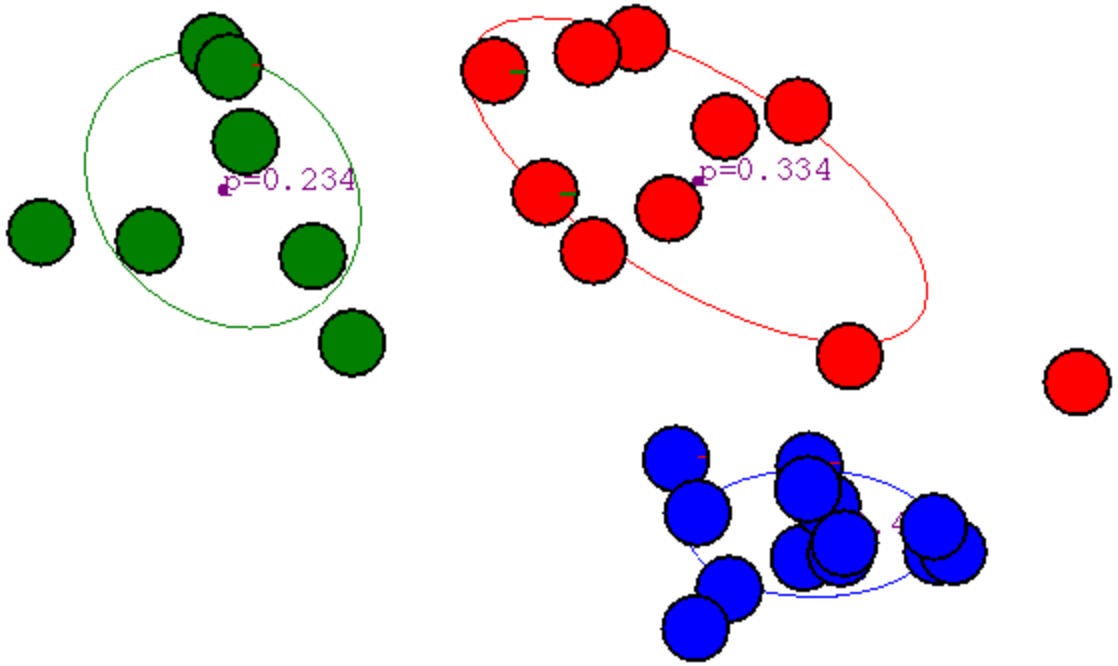
After 5th iteration



After 6th iteration



After 20th iteration



General GMM

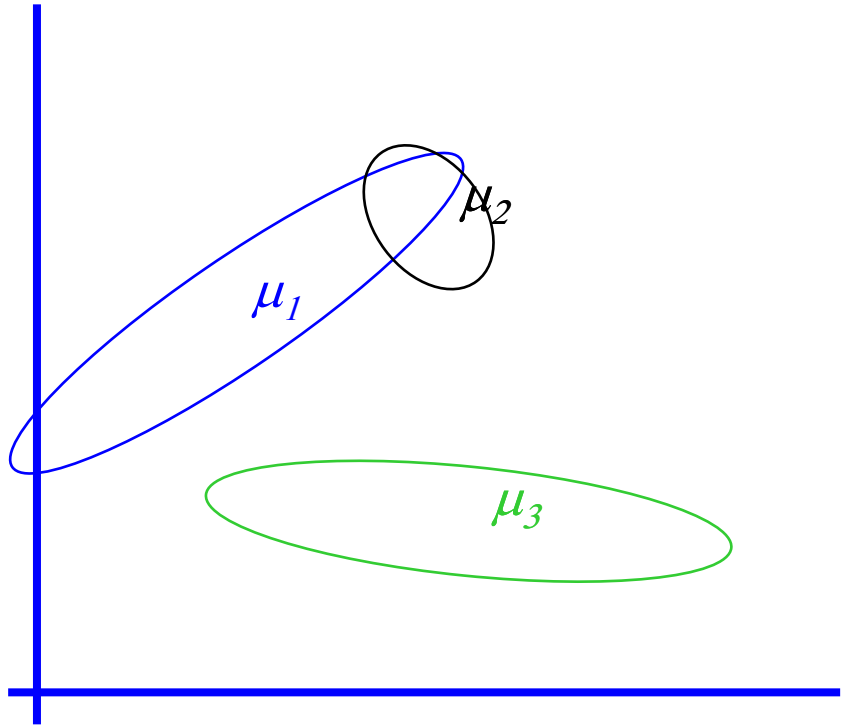
GMM – Gaussian Mixture Model (Multi-modal distribution)

$$p(x) = \sum_i p(x|y=i)\, P(y=i)$$

Mixture component — $p(x|y=i)$

Mixture proportion — $P(y=i)$

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



What you need to know…

- Hierarchical clustering algorithms
  - Single-linkage
  - Complete-linkage
  - Centroid-linkage
  - Average-linkage

- Partition based clustering algorithms
  - K-means
    - Coordinate descent
    - Seeding
    - Choosing K
  - Mixture models

    EM algorithm

General EM algorithm

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

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

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

θ - model parameter(s)

How to maximize marginal likelihood using EM?

Lower-bound on marginal likelihood

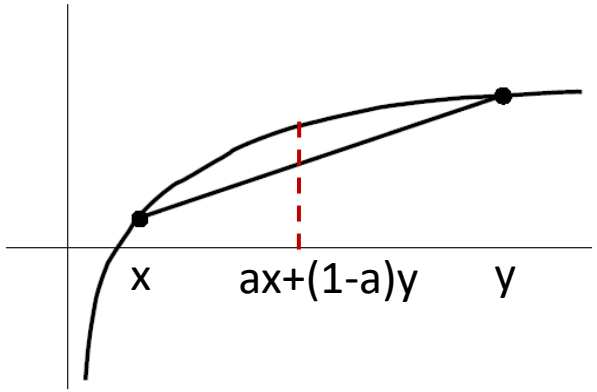
$$\log P(D;\theta) = \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{x}_j, \mathbf{z} \mid \theta)$$

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

P(z) f(z)

Variational approach

Jensen's inequality: $\log \sum_z P(z)\, f(z) \geq \sum_z P(z) \log f(z)$



log: concave function

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

Lower-bound on marginal likelihood

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

$$= \sum_{j=1}^{m} \log \sum_{z} Q(z \mid x_j) \frac{P(z, x_j \mid \theta)}{Q(z \mid x_j)}$$

P(z) f(z)

Jensen's inequality: $\log \sum_z P(z)\, f(z) \geq \sum_z P(z) \log f(z)$

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

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

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

E-step: Fix θ, maximize F over Q

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

E-step maximizes lower bound F on marginal likelihood => doesn't decrease the marginal likelihood

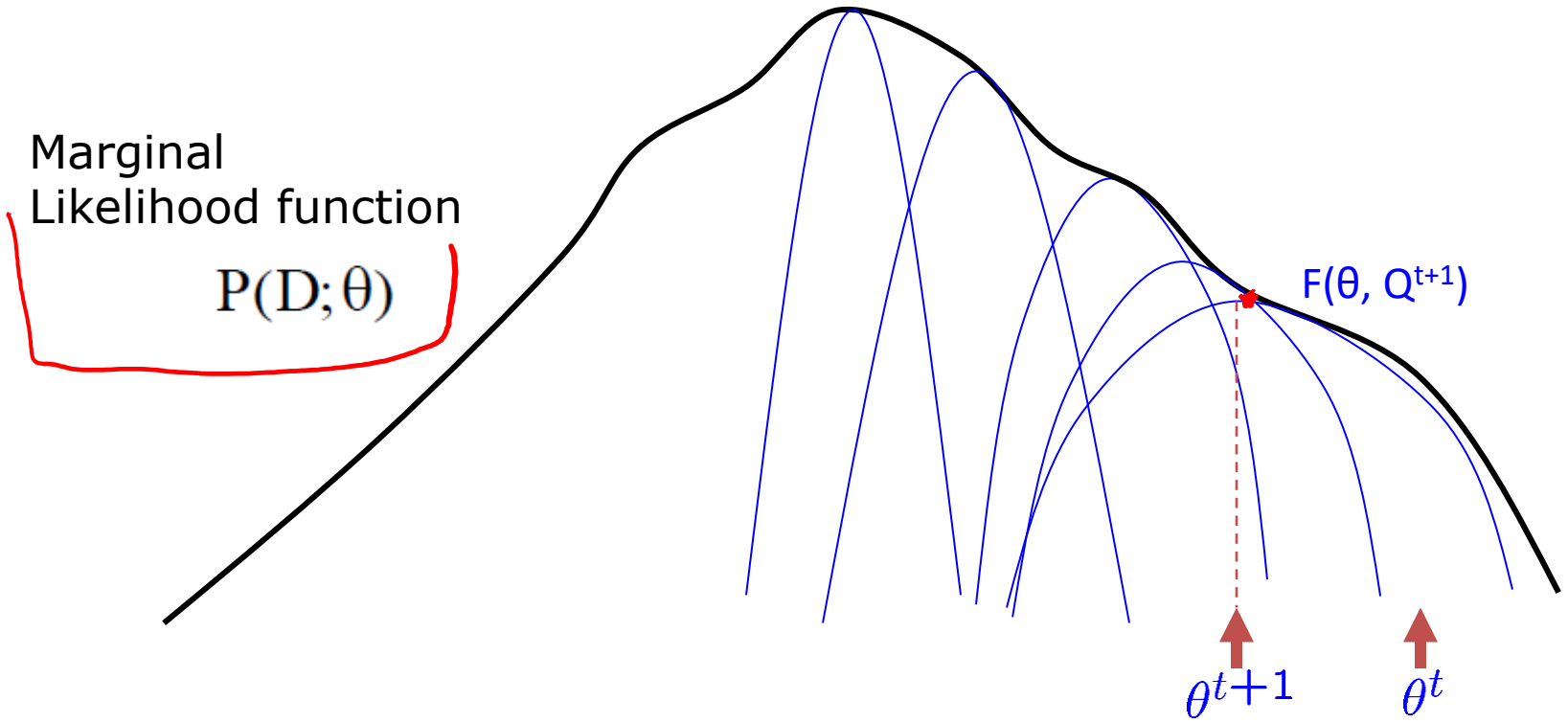
M-step: Fix Q, maximize F over θ

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

M-step maximizes lower bound F on marginal likelihood => doesn't decrease the marginal likelihood

Since marginal likelihood is bounded, convergence follows!

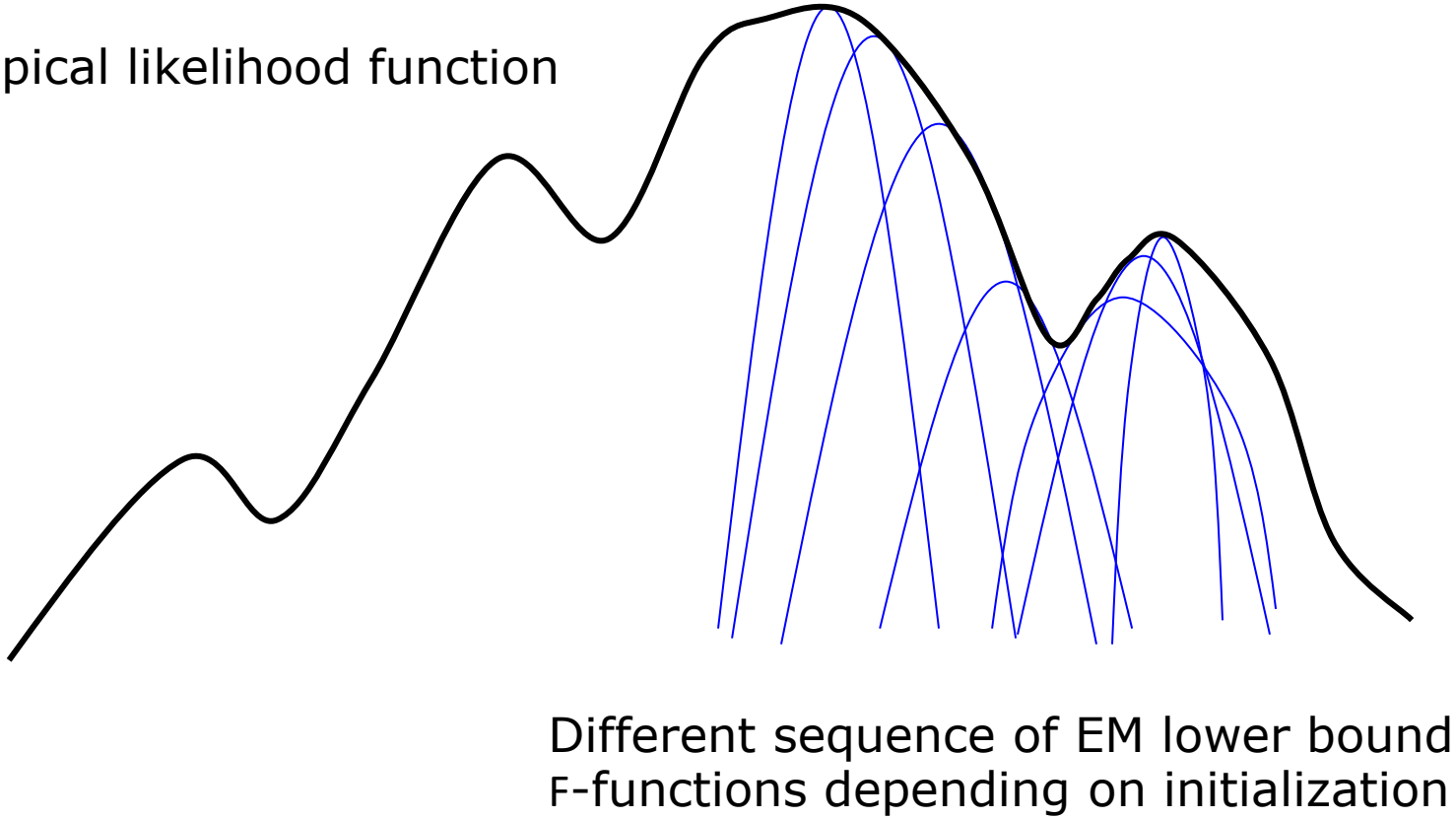
Convergence of EM



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

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

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

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

1

KL divergence between two distributions

E step

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

E-step: Fix θ, maximize F over Q

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

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

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

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

M-step: Fix Q, maximize F over θ

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

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

Fixed (Independent of θ)

=

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

Log likelihood if z was known

Expected log likelihood wrt Q

M step

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

M-step: Fix Q, maximize F over θ

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

Fixed (Independent of θ)

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

Expected log likelihood wrt $Q^{(t+1)}$

Use expected counts instead of counts when computing MLE:

If learning requires Count(**x**,**z**), Use $E_{Q(t+1)}$[Count(**x**,**z**)]

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

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.