EM Algorithm

Chapter 9, Statistical learning methods

Yu Hao

March 11, 2019

NaMI, Tongji University

Table of contents [1]

[1] Most of the content comes from Aarti Singh, https://www.cs.cmu.edu/~epxing/Class/10701-10s/Lecture/lecture10.pdf

K-means

K-means recap

- Randomly initialize K centers
  - $\boldsymbol{\mu}^{(0)} = \boldsymbol{\mu}_1^{(0)}, \ldots, \boldsymbol{\mu}_K^{(0)}$
- Classify: Assign each point $j \in \{1, \ldots, N\}$ to nearest center:
  - $C^{(t)}(j) \leftarrow \arg\min_i \left\|\boldsymbol{\mu}_i - \boldsymbol{x}_j\right\|^2$
- Recenter: $\boldsymbol{\mu}_i$ becomes centroid of its points
  - $\boldsymbol{\mu}_i^{(t+1)} \leftarrow \arg\min_{\boldsymbol{\mu}} \sum_{j:C(j)=i} \left\|\boldsymbol{\mu} - \boldsymbol{x}_j\right\|^2$
  - Equivalent to $\boldsymbol{\mu}_i \leftarrow$ average of its points!

What is K-means optimizing?

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

$$F(\boldsymbol{\mu}, C) = \sum_{j=1}^{N} \left\| \boldsymbol{\mu}_{C(j)} - \boldsymbol{x}_j \right\|^2$$

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

K-means algorithm

- Optimize potential function:

$$\min_{\boldsymbol{\mu}} \min_{C} F(\boldsymbol{\mu}, C) = \min_{\boldsymbol{\mu}} \min_{C} \sum_{i=1}^{K} \sum_{j:C(j)=i} \left\| \boldsymbol{\mu}_i - \boldsymbol{x}_i \right\|^2$$

- K-means algorithm:
  - ▷ (1) Fix $\boldsymbol{\mu}$, optimize C

$$\min_{C} \sum_{j=1}^{N} \left\| \boldsymbol{\mu}_{C(j)} - x_j \right\|^2 = \sum_{j=1}^{N} \min_{C(j)} \left\| \boldsymbol{\mu}_{C(j)} - \boldsymbol{x}_j \right\|^2$$

Exactly first step: assign each point to the nearest cluster center

K-means algorithm

- Optimize potential function:

$$\min_{\boldsymbol{\mu}} \min_{C} F(\boldsymbol{\mu}, C) = \min_{\boldsymbol{\mu}} \min_{C} \sum_{i=1}^{K} \sum_{j:C(j)=i} \|\boldsymbol{\mu}_i - \boldsymbol{x}_i\|^2$$

- K-means algorithm:
  - ▷ (2) Fix C, optimize $\boldsymbol{\mu}$

$$\min_{\boldsymbol{\mu}} \sum_{i=1}^{K} \sum_{j:C(j)=i} \|\boldsymbol{\mu}_i - \boldsymbol{x}_j\|^2 = \sum_{i=1}^{K} \boxed{\min_{\boldsymbol{\mu}_i} \sum_{j:C(j)=i} \|\boldsymbol{\mu}_i - \boldsymbol{x}_j\|^2}$$

Exactly second step: average of points in cluster i

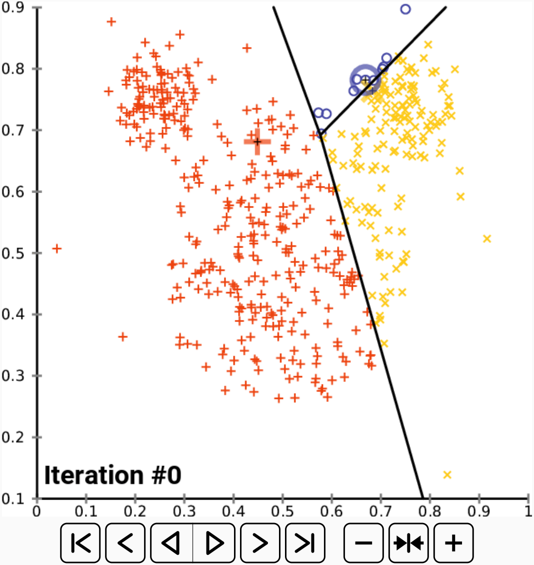
K-means algorithm

- Optimize potential function:

$$\min_{\boldsymbol{\mu}} \min_{C} F(\boldsymbol{\mu}, C) = \min_{\boldsymbol{\mu}} \min_{C} \sum_{i=1}^{K} \sum_{j:C(j)=i} \|\boldsymbol{\mu}_i - \boldsymbol{x}_i\|^2$$

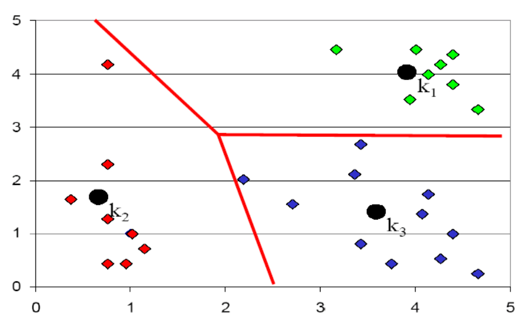
- K-means algorithm:
  - (1) Fix $\boldsymbol{\mu}$, optimize C **Expectation step**
  - (2) Fix C, optimize $\boldsymbol{\mu}$ **Maximization step**
  - Today, we will see a generalization of this approach: **EM algorithm**

Iterations of K-means [2]



[2] gif source: wiki `https://en.wikipedia.org/wiki/K-means_clustering`

K-means decision boundaries



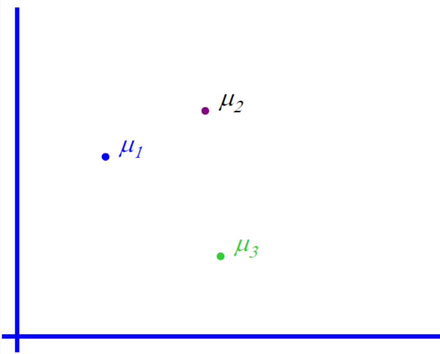
"Linear" Decision Boundaries

- **Generative Model:** Assume data comes from a mixture of K Gaussian distributions with same variance.

K-means: Generative model

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

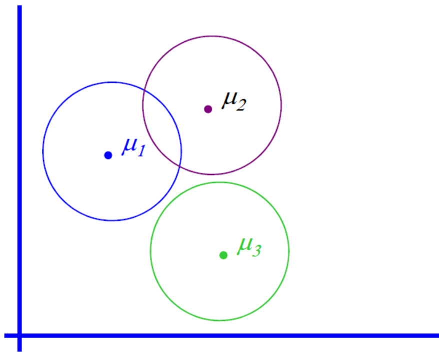
- There are K components
- Component i has an associated mean vector $\boldsymbol{\mu}_i$



K-means: Generative model

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

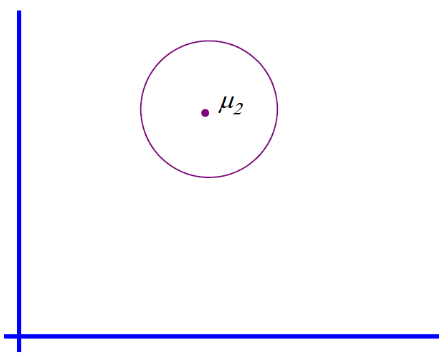
- There are K components
- Component i has an associated mean vector $\boldsymbol{\mu}_i$
- Each component generates data from a Gaussian with mean $\boldsymbol{\mu}_i$ and covariance matrix $\sigma^2\mathbf{I}$



K-means: Generative model

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

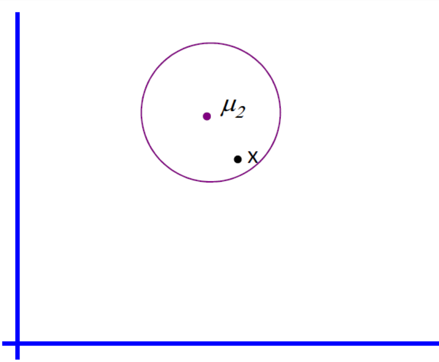
- Each data point is generated according to the following recipe:
- (1) Pick a component at random: choose component i with probability $P(y = i)$



K-means: Generative model

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

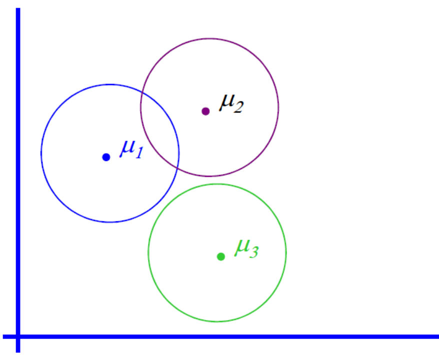
- Each data point is generated according to the following recipe:
- (1) Pick a component at random: choose component i with probability $P(y = i)$
- (2) Data point $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2 \mathbf{I})$



K-means: Generative model

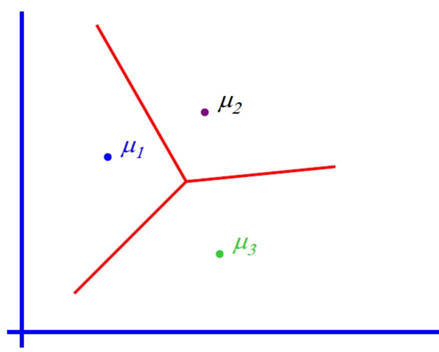
Mixture of K Gaussian distributions: (Multi-model distribution)

- $p(\boldsymbol{x}|y=i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2\mathbf{I})$
- $p(\boldsymbol{x}) = \sum_i p(\boldsymbol{x}|y=i)P(y=i)$
  - ▷ Mixture component
  - ▷ Mixture proportion



K-means: Generative model

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



- $p(\boldsymbol{x}|y = i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2 \mathbf{I})$
- Gaussian Bayes Classifier:
- "Linear Decision boundary" why?

Gaussian Bayes Classifier Decision Boundary [3]

- $p(\mathbf{x}|y=i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2\mathbf{I})$: the covariance is shared between classes,

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

$$\begin{aligned}&\log \pi_i - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\\ =&\log \pi_j - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\end{aligned}$$

$$\begin{aligned}&C + \mathbf{x}^T\Sigma^{-1}\mathbf{x} - 2\boldsymbol{\mu}_i^T\Sigma^{-1}\mathbf{x} + \boldsymbol{\mu}_i^T\Sigma^{-1}\boldsymbol{\mu}_i\\ =&C + \mathbf{x}^T\Sigma^{-1}\mathbf{x} - 2\boldsymbol{\mu}_j^T\Sigma^{-1}\mathbf{x} + \boldsymbol{\mu}_j^T\Sigma^{-1}\boldsymbol{\mu}_j\end{aligned}$$

$$[2(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)]\,\mathbf{x} - (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T\Sigma^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) = C$$

$$\Rightarrow \mathbf{a}^T\mathbf{x} - b = 0$$

[3] Mengye Ren, https://www.cs.toronto.edu/~urtasun/courses/CSC411/tutorial4.pdf

K-means: MLE

- Maximum Likelihood Estimate (MLE)

$$\arg\max_{\boldsymbol{\mu},\sigma^2,P(y)} \prod_i P(y_i, \boldsymbol{x}_i)$$

  But we don't know y_i is!

- Maximize marginal likelihood:

$$\begin{aligned} &\arg\max \prod_j P(\boldsymbol{x}_j) \\ =&\arg\max \prod_j \sum_i^K P(y_j = i, \boldsymbol{x}_j) \\ =&\arg\max \prod_j \sum_i^K P(y_j = i) p(\boldsymbol{x}_j | y_j = i) \end{aligned}$$

K-means: MLE

- Maximize marginal likelihood:

$$\begin{aligned}
&\arg\max \prod_j P(\boldsymbol{x}_j) \\
=&\arg\max \prod_j \sum_i^K P(y_j = i, \boldsymbol{x}_j) \\
=&\arg\max \prod_j \sum_i^K P(y_j = i) p(\boldsymbol{x}_j | y_j = i)
\end{aligned}$$

- Substitute with Gaussian distribution probability:

$$P(y_j = i, \boldsymbol{x}_j) \propto P(y_j = i) \exp\left[-\frac{1}{2\sigma^2} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|^2 \right]$$

K-means: MLE

- If each $\boldsymbol{x}_j$ belongs to one class $C(j)$ (hard assignment), marginal likelihood:

$$P(y_j = i) = \begin{cases} 1 & C(j) = i \\ 0 & else \end{cases}$$

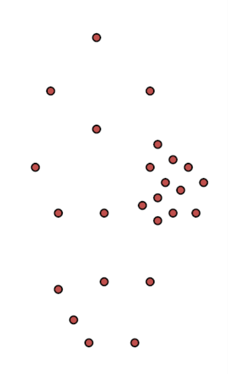
- Then, the log-likelihood function is

$$\ln \prod_{j=1}^{N} \sum_{i=1}^{K} P(y_j = i, x_j) \propto \ln \prod_{j=1}^{N} \exp\left[-\frac{1}{2\sigma^2} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_{C(j)} \right\|^2\right]$$

$$= \sum_{j=1}^{N} -\frac{1}{2\sigma^2} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_{C(j)} \right\|^2$$

Same as K-means!

One bad case for K-means



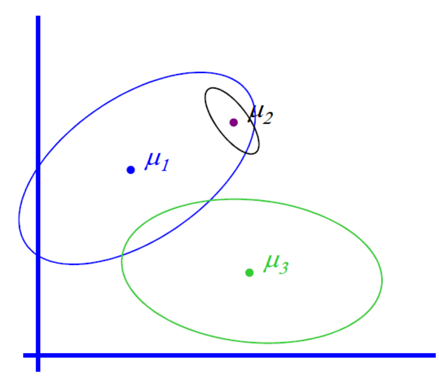
- Clusters may not be linear separable
- Clusters may overlap
- Some clusters may be "wider" than others

Gaussian Mixture Model

General GMM

GMM-Gaussian Mixture Model (Multi-model distribution)

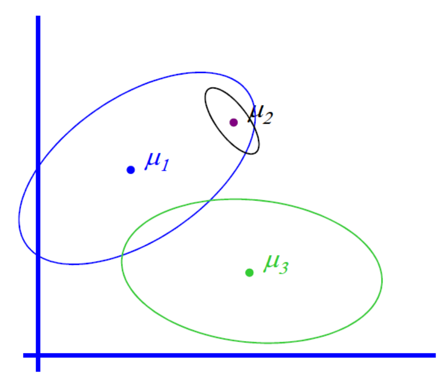
- There are K components
- Component i has an associated mean vector $\boldsymbol{\mu}_i$
- Each component generates data from Gaussian with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$



General GMM

GMM-Gaussian Mixture Model (Multi-model distribution)

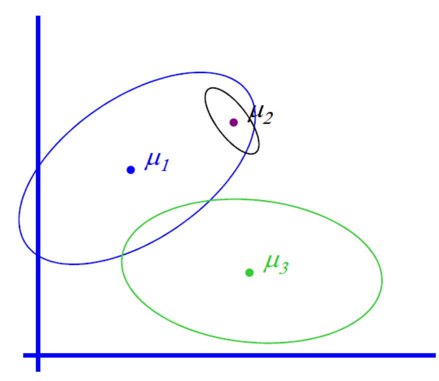
- Each data is generated according to the following recipe:
- (1) Pick a component at random: Choose component i with probability $P(y = i)$
- (2) Data point $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$



General GMM

GMM-Gaussian Mixture Model (Multi-model distribution)

- $p(\mathbf{x}|y=i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
- $p(\mathbf{x}) = \sum_i p(\mathbf{x}|y=i)P(y=i)$
  - ▷ Mixture component
  - ▷ Mixture proportion



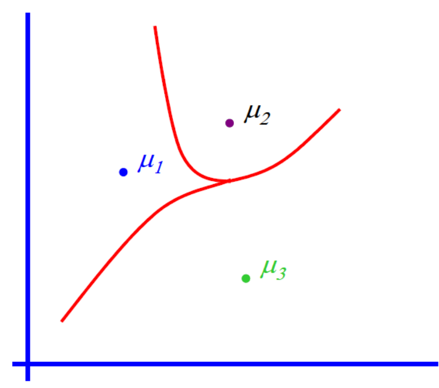
General GMM

GMM-Gaussian Mixture Model (Multi-model distribution)

- $p(\boldsymbol{x}|y=i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
- Gaussian Bayes Classifier:

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

$$\Rightarrow \boldsymbol{x}^T Q \boldsymbol{x} - 2\boldsymbol{b}^T \boldsymbol{x} + b = 0$$



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

GMM: marginal likelihood

- Maximize marginal likelihood:

$$
\begin{aligned}
&\arg\max \prod_j P(\boldsymbol{x}_j) \\
=&\arg\max \prod_j \sum_i^K P(y_j = i, \boldsymbol{x}_j) \\
=&\arg\max \prod_j \sum_i^K P(y_j = i) \color{red}{p(\boldsymbol{x}_j | y_j = i)}
\end{aligned}
$$

GMM: marginal likelihood

- Uncertain about class of each $\boldsymbol{x}_j$ (soft assignment), $P(y_j = i) = P(y = i)$

$$\prod_{j=1}^{N} \sum_{i=1}^{K} P(y_j = i, \boldsymbol{x}_j) \propto$$

$$\prod_{j=1}^{N} \sum_{i=1}^{K} P(y = i) \frac{1}{\sqrt{\det(\boldsymbol{\Sigma}_i)}} \exp\left[-\frac{1}{2} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i (\boldsymbol{x}_j - \boldsymbol{\mu}_i) \right]$$

- How do we find the $\boldsymbol{\mu}_i$'s which give max marginal likelihood?
  - ▷ Set $\frac{\partial F}{\partial \boldsymbol{\mu}_i} = 0$ and solve for $\boldsymbol{\mu}_i$'s. Non-linear non-analytically solvable.
  - ▷ Use gradient decent: Often slow but doable.

EM Algorithm

Expectation-Maximization (EM)

- EM is an optimization strategy for objective functions that can be interpreted as likelihoods in the presence missing data.
- It is much simpler than gradient methods.
- EM is an iterative algorithm with two linked steps:
  - ▷ E-step: fill in hidden values using inference
  - ▷ M-step: apply standard MLE methods
- This procedure monotonically improves the likelihood. Thus it always converges to a local optimum of the likelihood.

EM: A simple case

- We have unlabeled data $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$
- We know there are K classes
- We know $P(y = 1), P(y = 2), \ldots, P(y = K)$
- We don't know $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
- We know common variance σ^2

EM: A simple case

- Problem formulation:

$$
\begin{aligned}
&P(data|\boldsymbol{\mu}_1 \ldots \boldsymbol{\mu}_K) \\
=&P(\boldsymbol{x}_1 \ldots \boldsymbol{x}_N|\boldsymbol{\mu}_1 \ldots \boldsymbol{\mu}_K) \\
=&\prod_{j=1}^{N} p(\boldsymbol{x}_j|\boldsymbol{\mu}_1 \ldots \boldsymbol{\mu}_K) \qquad \textit{Indepentent data} \\
=&\prod_{j=1}^{N} \sum_{i=1}^{K} p(\boldsymbol{x}_j|\boldsymbol{\mu}_i) P(y=i) \qquad \textit{Marginalize over class} \\
\propto&\prod_{j=1}^{N} \sum_{i=1}^{K} \exp\left(-\frac{1}{2\sigma^2}\left\|\boldsymbol{x}_j-\boldsymbol{\mu}_i\right\|^2\right) P(y=i)
\end{aligned}
$$

Expectation (E) step

- If we know $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, then easily compute probability about point $\boldsymbol{x}_j$ belongs to class $y = i$

$$P(y = i | \boldsymbol{x}_j, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K) \propto \exp\left(-\frac{1}{2\sigma^2} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|^2\right) P(y = i)$$

Maximization (M) step

- If we know probability about point $\boldsymbol{x}_j$ belongs to class $y = i$, then MLE for $\boldsymbol{\mu}_i$ is weighted average.
- Imagine multiple copies of each $\boldsymbol{x}_j$, each with weight $P(y = i|\boldsymbol{x}_j)$:

$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{N} P(y = i|\boldsymbol{x}_j)\boldsymbol{x}_j}{\sum_{j=1}^{N} P(y = i|\boldsymbol{x}_j)}$$

EM for spherical, same variance GMMs

- E-step

$$P(y = i|\boldsymbol{x}_j, \boldsymbol{\mu}_1 \ldots \boldsymbol{\mu}_K) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|\boldsymbol{x}_j - \boldsymbol{\mu}_i\right\|^2\right) P(y = i)$$

  - ▷ Compute "expected" classes of all data points for each class
  - ▷ In K-means, we do hard assignment; EM dose soft assignment

- M-step

$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{N} P(y = i|\boldsymbol{x}_j)\boldsymbol{x}_j}{\sum_{j=1}^{N} P(y = i|\boldsymbol{x}_j)}$$

  - ▷ Compute Max. like $\boldsymbol{\mu}$ given our data's class membership distributions.

EM for general GMMs

- Iterate. On iteration t let our estimates be

$$\lambda_t = \boldsymbol{\mu}_1^{(t)}, \boldsymbol{\mu}_2^{(t)}, \ldots, \boldsymbol{\mu}_K^{(t)}, \boldsymbol{\Sigma}_1^{(t)}, \boldsymbol{\Sigma}_2^{(t)}, \ldots, \boldsymbol{\Sigma}_K^{(t)}, p_1^{(t)}, p_2^{(t)}, \ldots, p_K^{(t)}$$

  - $p_i^{(t)}$ is shorthand for estimate of $P(y = i)$

- E-step:

$$P(y = i | \boldsymbol{x}_j, \lambda_t) \propto p_i^{(t)} p(\boldsymbol{x}_j | \boldsymbol{\mu}_i^{(t)}, \boldsymbol{\Sigma}_i^{(t)})$$

EM for general GMMs

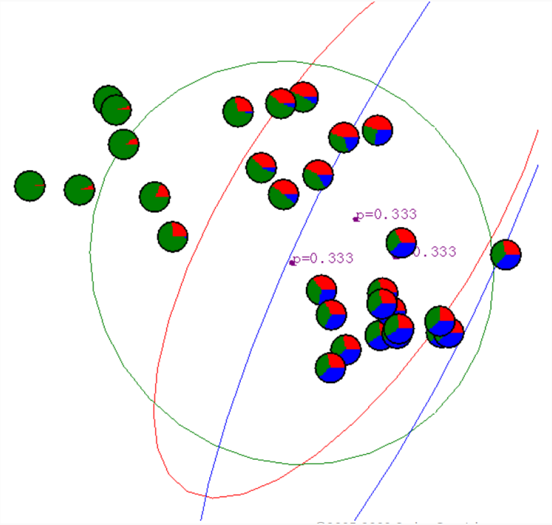
- M-step

$$\boldsymbol{\mu}_i^{(t+1)} = \frac{\sum_{j=1}^{N} P(y = i|\boldsymbol{x}_j, \lambda_t)\boldsymbol{x}_j}{\sum_{j=1}^{N} P(y = i|\boldsymbol{x}_j, \lambda_t)}$$

$$\boldsymbol{\Sigma}_i^{(t+1)} = \frac{\sum_j P(y = i|\boldsymbol{x}_j, \lambda_t)(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{(t+1)})(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{(t+1)})^T}{\sum_j P(y = i|\boldsymbol{x}_j, \lambda_t)}$$

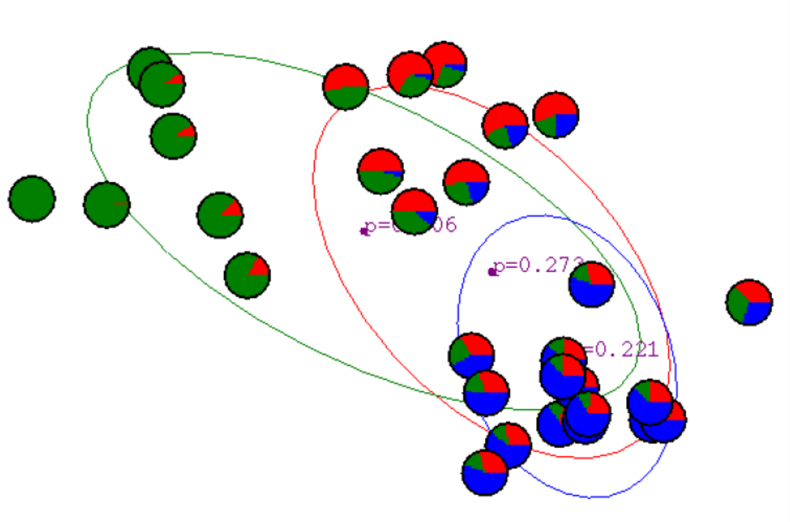
$$p_i^{(t+1)} = \frac{\sum_j P(y = i|\boldsymbol{x}_j, \boldsymbol{\lambda}_t)}{N}$$

EM for general GMMS: Example



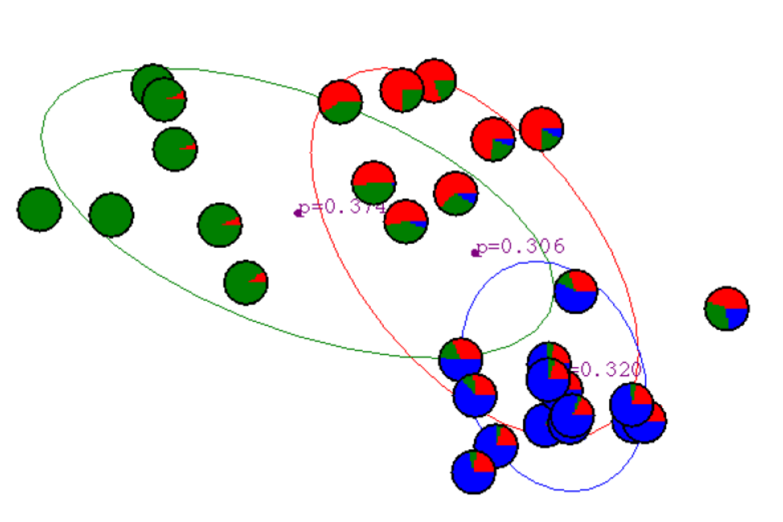
initialization

EM for general GMMS: Example



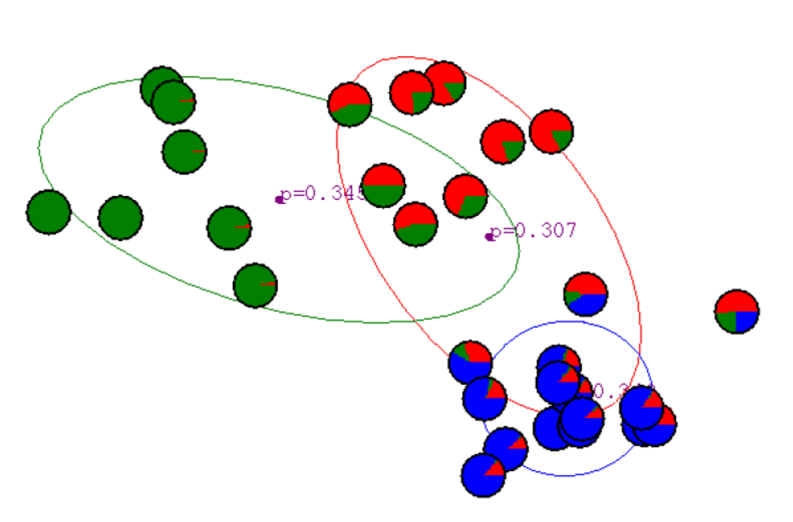
After 1^{st} *iteration*

EM for general GMMS: Example



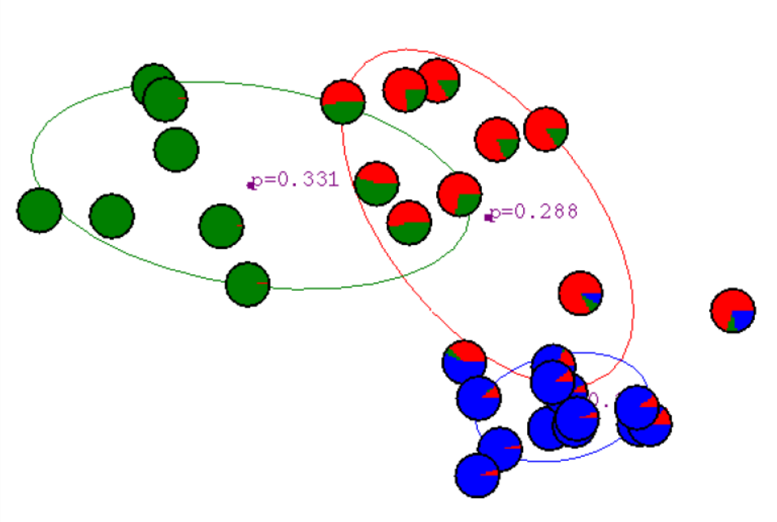
After 2^{nd} *iteration*

EM for general GMMS: Example



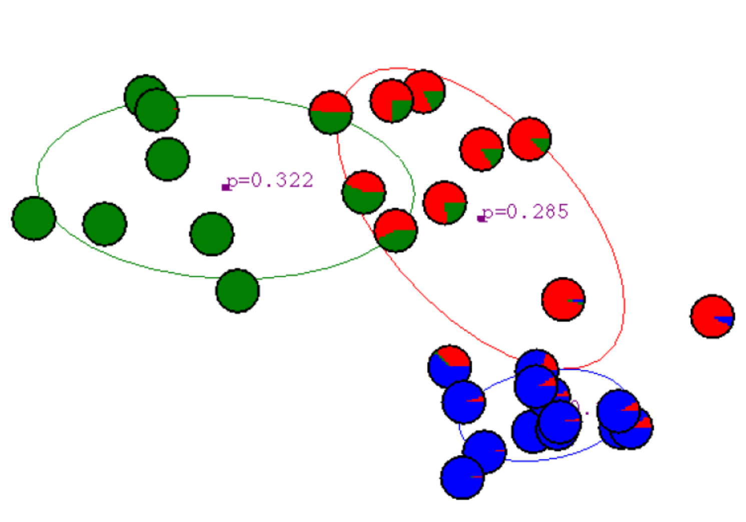
After 3^{rd} *iteration*

EM for general GMMS: Example



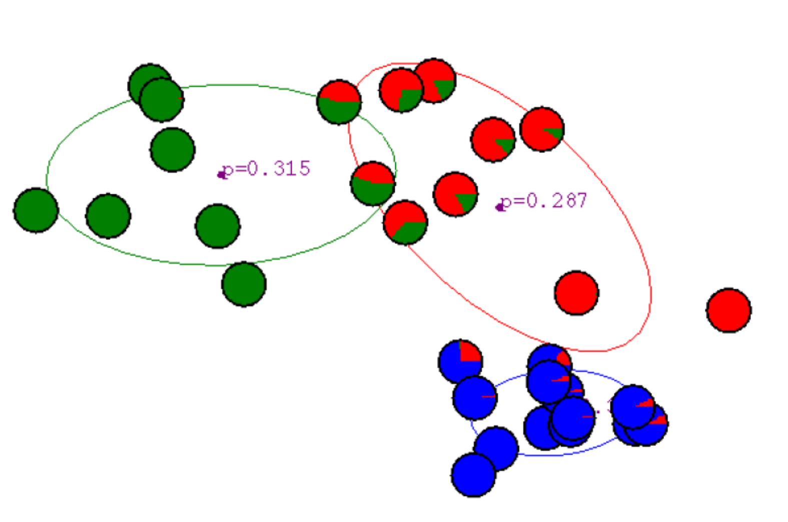
After 4^{th} *iteration*

EM for general GMMS: Example



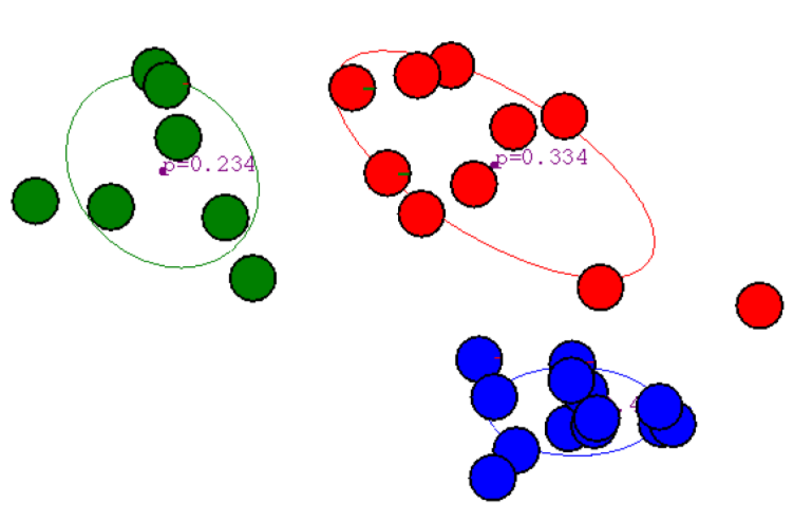
After 5^{th} *iteration*

EM for general GMMS: Example



After 6^{th} *iteration*

EM for general GMMS: Example



After 20^{th} *iteration*

General EM algorithm

- Marginal likelihood: $\mathbf{x}$ is observed, $\mathbf{z}$ is missing:

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

E-step

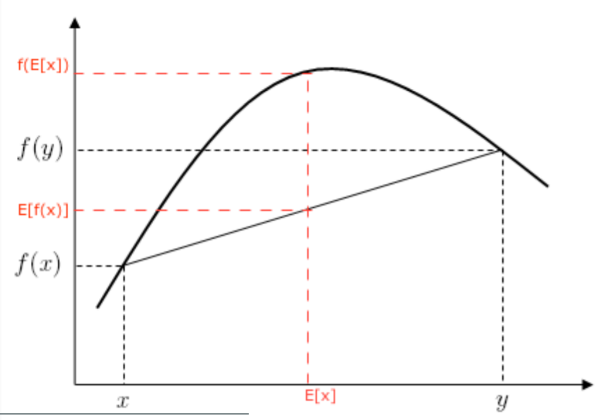
- x is observed, z is missing
- Compute probability of missing data given current choice of θ

$$Q^{(t+1)}(z|x_j) = P(z|x_j, \theta^{(t)})$$

Jensen's inequality [4]

- For a random variable x, if $f(x)$ is concave, then

$$f(E[x]) \geqslant E[f(x)]$$



[4] wiki, `https://en.wikipedia.org/wiki/Jensen%27s_inequality`

Lower-bound on marginal likelihood

$$
\begin{aligned}
P(\mathbf{D};\theta) &= \sum_{j=1}^{N} \log \sum_{z} P(x_j, z|\theta) \\
&= \sum_{j=1}^{N} \log \sum_{z} Q(z|x_j) \frac{P(z, x_j|\theta)}{Q(z|x_j)} \\
&\geqslant \sum_{j=1}^{N} \sum_{z} Q(z|x_j) \log \frac{P(z, x_j|\theta)}{Q(z|x_j)} \quad \textit{Jensen's inequality} \\
&= \sum_{j=1}^{N} \sum_{z} Q(z|x_j) \log P(z, x_j|\theta) + N.H(Q) \quad \textit{entropy of } Q
\end{aligned}
$$

M-step

$$P(\mathbf{D};\theta) \geqslant \sum_{j=1}^{N} \sum_{z} Q(z|\boldsymbol{x}_j) \log P(z, \boldsymbol{x}_j|\theta) + N.H(Q)$$

- Maximize lower bound on marginal likelihood

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

Convergence of EM

$$P(\mathbf{D};\theta) \geqslant F(\theta, Q)$$

- M-step: Fix Q, maximize F over θ

$$\begin{aligned} P(\mathbf{D};\theta) &\geqslant F(\theta, Q^{(T)}) \\ &= \sum_{j=1}^{N} \sum_{\mathbf{z}} Q^{(t)}(\mathbf{z}|\mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j|\theta) + N.H(Q^{(t)}) \end{aligned}$$

Maximizes lower bound F on marginal likelihood

Convergence of EM

- E-step: Fix θ, maximize F over Q

$$
\begin{aligned}
P(\mathbf{D};\theta^{(t)}) &\geqslant F(\theta^{(t)}, Q) \\
&= \sum_{j=1}^{N}\sum_{z} Q(z|\mathbf{x}_j)\log\frac{P(z,\mathbf{x}_j|\theta^{(t)})}{Q(z|\mathbf{x}_j)} \\
&= \sum_{j=1}^{N}\sum_{z} Q(z|\mathbf{x}_j)\log\frac{P(z|\mathbf{x}_j,\theta^{(t)})P(\mathbf{x}_j|\theta^{(t)})}{Q(z|\mathbf{x}_j)} \\
&= \sum_{j=1}^{N}\sum_{z} Q(z|\mathbf{x}_j)\log\frac{P(z|\mathbf{x}_j,\theta^{(t)})}{Q(z|\mathbf{x}_j)} \quad \leftarrow \textit{KL divergence} \\
&+ \sum_{j=1}^{N}\sum_{z} Q(z|\mathbf{x}_j)\log P(\mathbf{x}_j|\theta^{(t)}) \quad \leftarrow P(\mathbf{D};\theta^{(t)})
\end{aligned}
$$

Convergence of EM

- E-step: Fix θ, maximize F over Q

$$P(\mathbf{D};\theta^{(t)}) \geqslant F(\theta^{(t)}, Q)$$
$$= \sum_{j=1}^{N} -KL\left(Q(\mathbf{z}|\mathbf{x}_j), P(\mathbf{z}|\mathbf{x}_j, \theta^{(t)})\right) + P(\mathbf{D};\theta^{(t)})$$

- $KL \geqslant 0$ (why?), F is maximized if KL divergence $= 0$

$$KL(Q, P) = 0 \; if \; Q = P$$

- Recall E-step:

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

Convergence of EM

- M-step: Fix Q, maximize F over θ

$$P(\mathbf{D};\theta) \geqslant F(\theta, Q^{(t)}) = \sum_{j=1}^{N} \sum_{\boldsymbol{z}} Q^{(t)}(\boldsymbol{z}|\boldsymbol{x}_j) \log P(\boldsymbol{z}, \boldsymbol{x}_j|\theta) + N.H(Q^{(t)})$$

Maximize lower bound F on marginal likelihood

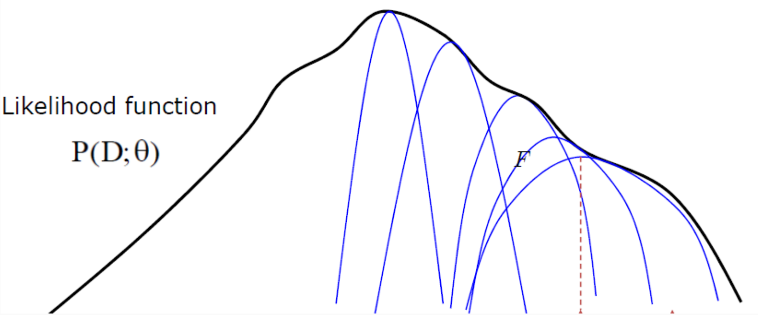
- E-step: Fix θ, maximize F over Q

$$P(\mathbf{D};\theta^{(t)}) \geqslant F(\theta^{(t)}, Q) = P(\mathbf{D};\theta^{(t)}) - \sum_{j=1}^{N} KL\left(Q(\boldsymbol{z}|\boldsymbol{x}_j)||P(\boldsymbol{z}|\boldsymbol{x}_j, \theta^{(t)})\right)$$

Re-aligns F with marginal likelihood

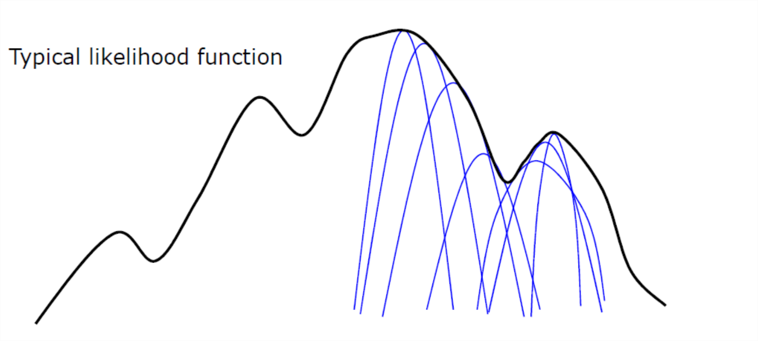
$$F(\theta^{(t)}, Q^{(t+1)}) = P(\mathbf{D};\theta^{(t)})$$

Monotonic convergence of EM



- EM monotonically converges to a local maximum of likelihood

Monotonic convergence of EM



- Different sequence of EM surrogate F-functions depending on initialization.
- Use multiple, randomized initializations in practice

Recent papers related to EM algorithm

- Balakrishnan, Sivaraman, Martin J. Wainwright, and Bin Yu. "Statistical guarantees for the EM algorithm: From population to sample-based analysis." The Annals of Statistics 45.1 (2017): 77-120.
- Schwartz, Boaz, Sharon Gannot, and Emanuël AP Habets. "Online speech dereverberation using Kalman filter and EM algorithm." IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 23.2 (2015): 394-406.
- Zhang, Wen, Ye Yang, and Qing Wang. "Using Bayesian regression and EM algorithm with missing handling for software effort prediction." Information and software technology 58 (2015): 58-70.

Thanks!