EM Algorithm

Chapter 9, Statistical learning methods

Yu Hao

January 17, 2019

NaMI, Tongji University

Table of contents

1. K-means
2. Gaussian Mixture Model
3. EM Algorithm

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:
  - ▷ $C^{(t)}(j) \leftarrow \arg\min_i \|\boldsymbol{\mu}_i - \boldsymbol{x}_j\|^2$
- Recenter: $\boldsymbol{\mu}_i$ becomes centroid of its points
  - ▷ $\boldsymbol{\mu}_i^{(t+1)} \leftarrow \arg\min_{\boldsymbol{\mu}} \sum_{j:C(j)=i} \|\boldsymbol{\mu} - \boldsymbol{x}_j\|^2$
  - ▷ 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:

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

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

K-means algorithm

- Optimize potential function:

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

- 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:

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

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

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

Exactly second step: average of points in cluster i

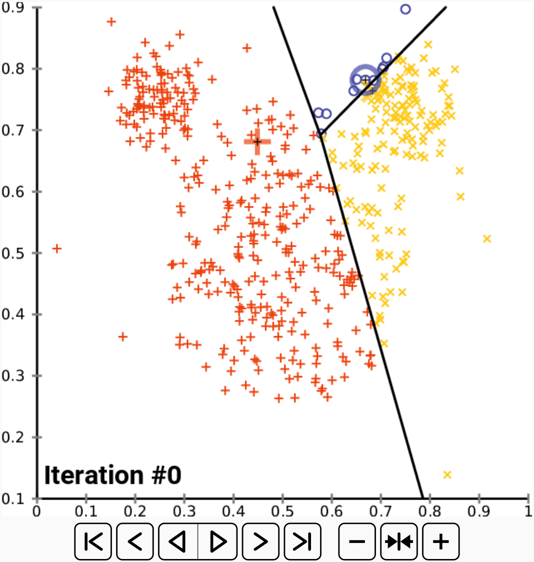
K-means algorithm

- Optimize potential function:

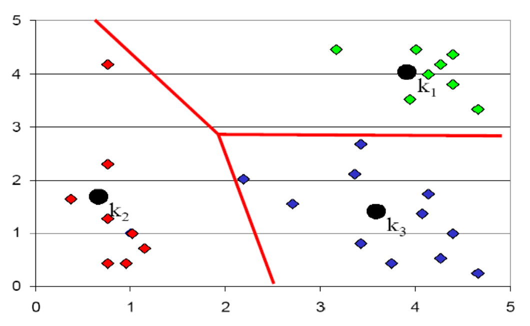
$$\min_{\boldsymbol{\mu}} \min_{C} F(\boldsymbol{\mu}, C) = \min_{\boldsymbol{\mu}} \min_{C} \sum_{i=1}^{K} \sum_{j:C(j)=i} \left\| \boldsymbol{\mu}_i - \boldsymbol{x}_i \right\|^2 \quad (2)$$

- 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



K-means decision boundaries



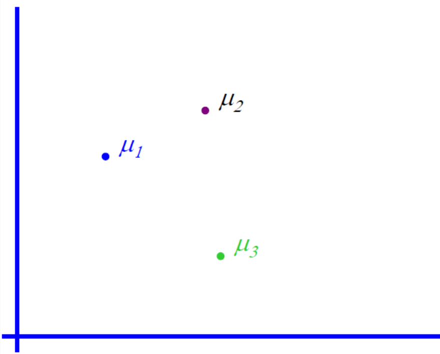
"Linear" Decision Boundaries

- **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-model distribution)

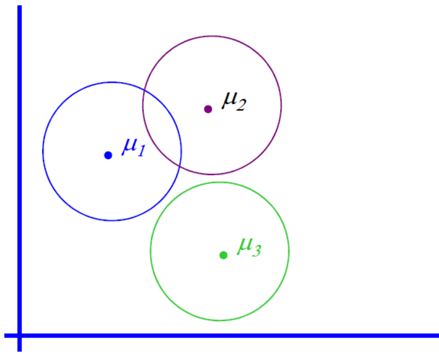
- There are K components
- Component i has an associated mean vector $\boldsymbol{\mu}_i$



K-means: Generative model

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

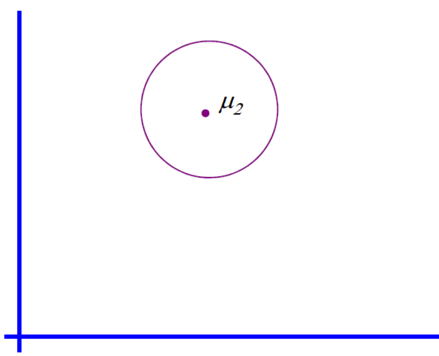
- There are K components
- Component i has an associated mean vector $\boldsymbol{\mu}_i$
- Each component generates data from a Gaussian with mean $\boldsymbol{\mu}_i$ and covariance matrix $\sigma^2\mathbf{I}$



K-means: Generative model

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

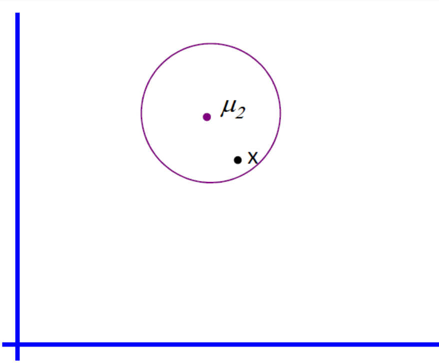
- Each data point is generated according to the following recipe:
- (1) Pick a component at random: choose component i with probability $P(y = i)$



K-means: Generative model

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

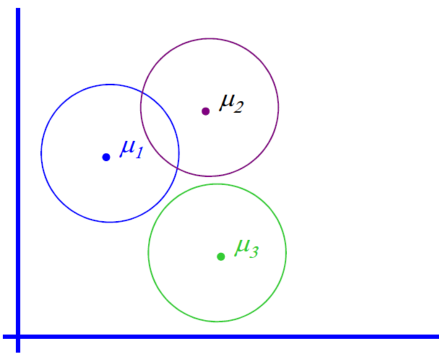
- Each data point is generated according to the following recipe:
- (1) Pick a component at random: choose component i with probability $P(y = i)$
- (2) Data point $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2 \mathbf{I})$



K-means: Generative model

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

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

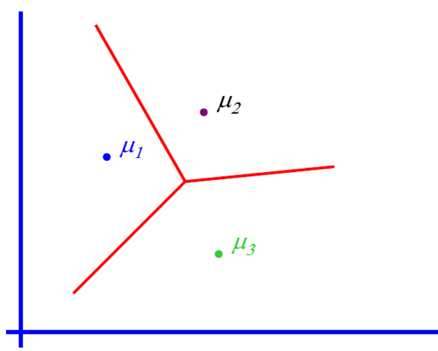
- Gaussian Bayes Classifier:

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

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

$$= \boldsymbol{w}^T \boldsymbol{x}$$

- $\boldsymbol{w}$ depends on $\boldsymbol{\mu}, \sigma^2, P(y)$



"Linear Decision boundary"
second-order terms cancel out

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:

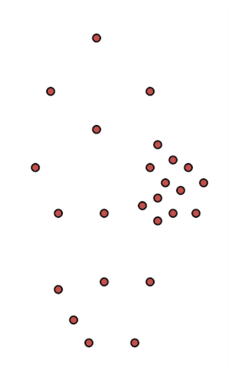
$$P(y_j = i) = \begin{cases} 1 & C(j) = i \\ 0 & else \end{cases}$$

- Then, the log-likelihood function is

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

Same as K-means!

One bad case for K-means



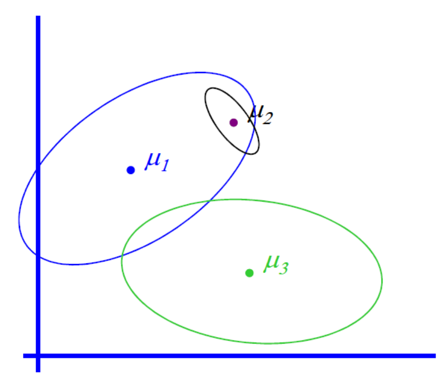
- CLusters may not be linear separable
- Clusters may overlap
- Some clusters may be "wider" than others

Gaussian Mixture Model

General GMM

GMM-Gaussian Mixture Model (Multi-model distribution)

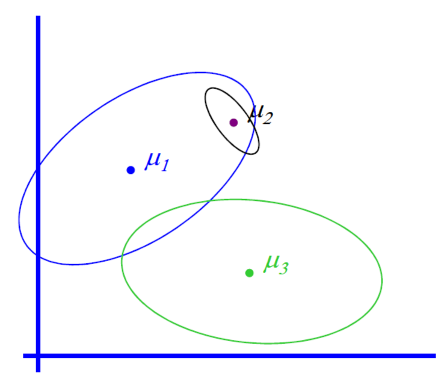
- There are K components
- Component i has an associated mean vector $\boldsymbol{\mu}_i$
- Each component generates data from Gaussian with mean $\boldsymbol{\mu}_i$ and covariance matrix Σ_i



General GMM

GMM-Gaussian Mixture Model (Multi-model distribution)

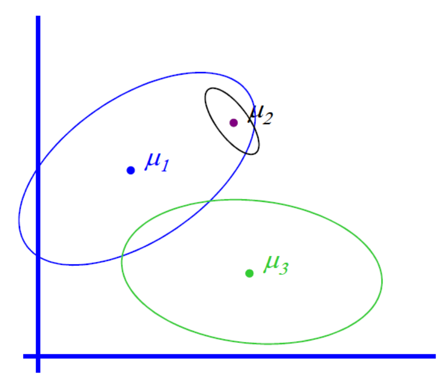
- Each data is generated according to the following recipe:
- (1) Pick a component at random: Choose component i with probability $P(y = i)$
- (2) Data point $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$



General GMM

GMM-Gaussian Mixture Model (Multi-model distribution)

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



General GMM

GMM-Gaussian Mixture Model (Multi-model distribution)

- $p(\boldsymbol{x}|y=i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$

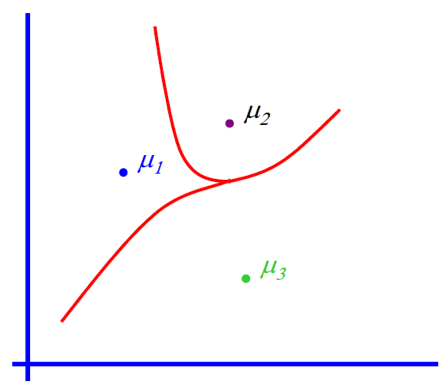
- Gaussian Bayes Classifier:

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

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

$$= \boldsymbol{x}^T \mathbf{W} \boldsymbol{x} + \boldsymbol{w}^T \boldsymbol{x}$$

- $\mathbf{W}$, $\boldsymbol{w}$ depends on $\boldsymbol{\mu}, \Sigma, P(y)$



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

GMM: marginal likelihood

- Maximize marginal likelihood:

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

GMM: marginal likelihood

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

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

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

- 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:

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

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.
- Image multiple copies of each $\boldsymbol{x}_j$, each with weight $P(y = i|\boldsymbol{x}_j)$:

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