Inference and Representation

Rachel Hodos

New York University

Lecture 5, October 6, 2015
Outline:

- Unsupervised learning
- Example: clustering
- Review k-means clustering
- Probabilistic perspective -> GMMs
- EM algorithm for GMMs
- General derivation of EM algorithm
- Identifiability
\textbf{K-}Means and Gaussian Mixture Models

David Rosenberg

New York University

June 15, 2015
Example: Old Faithful Geyser

- Looks like two clusters.
- How to find these clusters algorithmically?
$k$-Means: By Example

- Standardize the data.
- Choose two cluster centers.

From Bishop’s *Pattern recognition and machine learning*, Figure 9.1(a).
k-means: by example

- Assign each point to closest center.

From Bishop's *Pattern recognition and machine learning*, Figure 9.1(b).
k-means: by example

- Compute new class centers.

---

From Bishop's *Pattern recognition and machine learning*, Figure 9.1(c).
k-means: by example

- Assign points to closest center.

---

From Bishop’s *Pattern recognition and machine learning*, Figure 9.1(d).
Compute cluster centers.

From Bishop's *Pattern recognition and machine learning*, Figure 9.1(e).
k-means: by example

- Iterate until convergence.

From Bishop’s *Pattern recognition and machine learning*, Figure 9.1(i).
The clustering for $k = 3$ below is a local minimum, but suboptimal:

Would be better to have one cluster here

... and two clusters here

Let’s consider a **generative model** for the data.

**Suppose**

1. There are $k$ clusters.
2. We have a probability density for each cluster.

**Generate a point as follows**

1. Choose a random cluster $z \in \{1, 2, \ldots, k\}$.
   - $Z \sim \text{Multi}(\pi_1, \ldots, \pi_k)$.
2. Choose a point from the distribution for cluster $Z$.
   - $X \mid Z = z \sim p(x \mid z)$. 


Gaussian Mixture Model \((k = 3)\)

1. Choose \(Z \in \{1, 2, 3\} \sim \text{Multi} \left( \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right) \).
2. Choose \(X \mid Z = z \sim \mathcal{N}(X \mid \mu_z, \Sigma_z)\).
Gaussian Mixture Model: Joint Distribution

- Factorize joint according to Bayes net:

\[ p(x, z) = p(z)p(x | z) = \pi_z \mathcal{N}(x | \mu_z, \Sigma_z) \]

- \( \pi_z \) is probability of choosing cluster \( z \).
- \( X \mid Z = z \) has distribution \( \mathcal{N}(\mu_z, \Sigma_z) \).
- \( z \) corresponding to \( x \) is the true cluster assignment.
Latent Variable Model

- Back in reality, we observe $X$, not $(X, Z)$.
- Cluster assignment $Z$ is called a hidden variable.

**Definition**

A latent variable model is a probability model for which certain variables are never observed.

- e.g. The Gaussian mixture model is a latent variable model.
Model-Based Clustering

- We observe $X = x$.
- The conditional distribution of the cluster $Z$ given $X = x$ is
  \[ p(z \mid X = x) = \frac{p(x, z)}{p(x)} \]
- The conditional distribution is a **soft assignment** to clusters.
- A **hard assignment** is
  \[ z^* = \arg \min_{z \in \{1, \ldots, k\}} \mathbb{P}(Z = z \mid X = x). \]
- So if we have the model, clustering is trivial.
Estimating/Learning the Gaussian Mixture Model

- We’ll use the common acronym GMM.
- What does it mean to “have” or “know” the GMM?
- It means knowing the parameters
  
  Cluster probabilities: \( \pi = (\pi_1, \ldots, \pi_k) \)
  
  Cluster means: \( \mu = (\mu_1, \ldots, \mu_k) \)
  
  Cluster covariance matrices: \( \Sigma = (\Sigma_1, \ldots, \Sigma_k) \)

- We have a probability model: let’s find the MLE.
- Suppose we have data \( D = \{x_1, \ldots, x_n\} \).
- We need the model likelihood for \( D \).
Since we only observe $X$, we need the **marginal distribution**:

$$p(x) = \sum_{z=1}^{k} p(x, z)$$

$$= \sum_{z=1}^{k} \pi_z \mathcal{N}(x | \mu_z, \Sigma_z)$$

- Note that $p(x)$ is a convex combination of probability densities.
- This is a common form for a probability model...
Definition

A probability density \( p(x) \) represents a mixture distribution or mixture model, if we can write it as a convex combination of probability densities. That is,

\[
p(x) = \sum_{i=1}^{k} w_i p_i(x),
\]

where \( w_i \geq 0 \), \( \sum_{i=1}^{k} w_i = 1 \), and each \( p_i \) is a probability density.

- In our Gaussian mixture model, \( X \) has a mixture distribution.
- More constructively, let \( S \) be a set of probability distributions:
  1. Choose a distribution randomly from \( S \).
  2. Sample \( X \) from the chosen distribution.

- Then \( X \) has a mixture distribution.
EM Algorithm for GMM: Overview

1. Initialize parameters $\mu$, $\Sigma$, $\pi$.

2. “E step”. Evaluate the responsibilities using current parameters:

   \[
   \gamma_{ij} = \frac{\pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}{\sum_{c=1}^{k} \pi_c \mathcal{N}(x_i \mid \mu_c, \Sigma_c)},
   \]

   for $i = 1, \ldots, n$ and $j = 1, \ldots, k$.

3. “M step”. Re-estimate the parameters using responsibilities:

   \[
   \mu_{c}^{\text{new}} = \frac{1}{n_c} \sum_{i=1}^{n} \gamma_{ci} x_i
   \]

   \[
   \Sigma_{c}^{\text{new}} = \frac{1}{n_c} \sum_{i=1}^{n} \gamma_{ci} (x_i - \mu_{\text{MLE}}) (x_i - \mu_{\text{MLE}})^T
   \]

   \[
   \pi_{c}^{\text{new}} = \frac{n_c}{n},
   \]

4. Repeat from Step 2, until log-likelihood converges.
EM for GMM

- Initialization

From Bishop's *Pattern recognition and machine learning*, Figure 9.8.
First soft assignment:

From Bishop’s *Pattern recognition and machine learning*, Figure 9.8.
First soft assignment:

From Bishop's *Pattern recognition and machine learning*, Figure 9.8.
After 5 rounds of EM:

From Bishop's *Pattern recognition and machine learning*, Figure 9.8.
After 20 rounds of EM:

From Bishop’s *Pattern recognition and machine learning*, Figure 9.8.
Relation to $K$-Means

- EM for GMM seems a little like $k$-means.
- In fact, there is a precise correspondence.
- First, fix each cluster covariance matrix to be $\sigma^2 I$.
- As we take $\sigma^2 \to 0$, the update equations converge to doing $k$-means.
- If you do a quick experiment yourself, you’ll find
  - Soft assignments converge to hard assignments.
  - Has to do with the tail behavior (exponential decay) of Gaussian.
Motivation:
- With hidden variables, MLE is harder to compute (not always closed form solution).
- Also, we may want to estimate the expected states of the hidden variables.

EM algorithm can help with both
- EM is iterative algorithm to maximize log-likelihood