Mixture Models, EM Algorithms and K-Means

Jun Zhu
dcszj@mail.tsinghua.edu.cn
Department of Computer Science and Technology
Tsinghua University

Oct 18, 2012
Outline

- Mixture of Gaussians
- Maximum Likelihood Estimation
- EM Algorithms
- K-Means
- Hierarchical Clustering
- Discussions
Gaussian Distributions

- **Univariate Gaussian distribution**

\[ p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \]

- **Given parameters**, we can draw samples and plot distributions
Maximum Likelihood Estimation

Given a data set $\mathcal{D} = \{x_1, \ldots, x_N\}$, the likelihood is

$$p(\mathcal{D}|\mu, \sigma^2) = \prod_{n=1}^{N} p(x_n|\mu, \sigma^2)$$

MLE estimates the parameters as

$$(\mu_{ML}, \sigma_{ML}^2) = \arg\max_{\mu, \sigma^2} \log p(\mathcal{D}|\mu, \sigma^2)$$

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^{N} x_n \quad \text{sample mean}$$

$$\sigma_{ML}^2 = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu_{ML})^2 \quad \text{sample variance}$$
Gaussian Distributions

- $d$-dimensional multivariate Gaussian

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)\Sigma^{-1}(x - \mu)\right)$$

- Given parameters, we can draw samples and plot distributions

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \quad \Sigma = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$
Maximum Likelihood Estimation

Given a data set $\mathcal{D} = \{x_1, \ldots, x_N\}$, the likelihood is

$$p(\mathcal{D}|\mu, \Sigma) = \prod_{n=1}^{N} p(x_n|\mu, \Sigma)$$

MLE estimates the parameters as

$$(\mu_{ML}, \Sigma_{ML}) = \arg\max_{\mu, \Sigma} \log p(\mathcal{D}|\mu, \Sigma)$$

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^{N} x_n \quad \text{sample mean}$$

$$\sigma^2_{ML} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu_{ML})^2 \quad \text{sample variance}$$
Other Nice Analytic Properties

- Marginal is Gaussian
- Conditional is Gaussian

![Graphs](image)

1. $x_b = 0.7$
2. $p(x_a, x_b)$
3. $p(x_a | x_b = 0.7)$
4. $p(x_a)$
Limitations of Single Gaussians

- Single Gaussian is unimodal

... can’t fit well multimodal data, which is more realistic!
Mixture of Gaussians

A simple family of multi-modal distributions

- treat unimodal Gaussians as basis (or component) distributions
- superpose multiple Gaussians via linear combination

\[ p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2) \]
Mixture of Gaussians

- A simple family of multi-modal distributions
  - treat unimodal Gaussians as \textit{basis (or component) distributions}
  - superpose multiple Gaussians via \textit{linear combination}

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x | \mu_k, \Sigma_k)$$

\textit{What conditions should the mixing coefficients satisfy?}
Another Motivation …

Some heavy-tailed distributions are mixture of Gaussians

- Laplace distribution

- Student-t distribution
MLE for Mixture of Gaussians

- Log-likelihood

\[
\log p(\mathcal{D}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \right)
\]

- this is complicated … 😞
- … but, we know the MLE for single Gaussians are easy

- A heuristic procedure (can we iterate?)
  - allocate data into different components
  - estimate each component Gaussian analytically
Optimal Conditions

Some math

\[ \mathcal{L}(\mu, \Sigma) = \log p(D|\mu, \Sigma) = \sum_{n=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \right) \]

\[ \frac{\partial \mathcal{L}}{\partial \mu_k} = 0 \quad \rightarrow \quad \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} \Sigma_k^{-1} (x_n - \mu_k) = 0 \]

\[ \gamma(z_{nk}) \]

\[ \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \]

A weighted sample mean!
Optimal Conditions

Some math

\[ \mathcal{L}(\mu, \Sigma) = \log p(\mathcal{D}|\mu, \Sigma) = \sum_{n=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \right) \]

\[ \frac{\partial \mathcal{L}}{\partial \Sigma_k} = 0 \quad \Rightarrow \quad \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^\top \]

\textit{A weighted sample variance!}
Optimal Conditions

Some math

\[ \mathcal{L}(\mu, \Sigma) = \log p(D|\mu, \Sigma) = \sum_{n=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \right) \]

Note: constraints exist for mixing coefficients!

\[ L = \mathcal{L}(\mu, \Sigma) + \lambda \left( \sum_{k=1}^{K} \pi_k - 1 \right) \]

\[ \frac{\partial L}{\partial \pi_k} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} \frac{\mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} + \lambda = 0 \]

\[ \pi_k = \frac{N_k}{N} \]

The ratio of data assigned to component \( k \)!
Optimal Conditions – summary

- The set of couple conditions

\[ \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{n,k}) x_n \]

\[ \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{n,k})(x_n - \mu_k)(x_n - \mu_k)^\top \]

\[ \pi_k = \frac{N_k}{N} \]

- The key factor to get them coupled

\[ \gamma(z_{n,k}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)} \]

- If we know \( \gamma(z_{n,k}) \), each component Gaussian is easy to estimate!
The EM Algorithm

- **E-step:** estimate the responsibilities
  \[
  \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}
  \]

- **M-step:** re-estimate the parameters
  \[
  \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \\
  \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T \\
  \pi_k = \frac{N_k}{N}
  \]

Initialization plays a key role to succeed!
A Running Example

The data and a mixture of two isotropic Gaussians
A Running Example

Initial E-step
A Running Example

Initial M-step
A Running Example

The 2\textsuperscript{nd} M-step
A Running Example

The 5<sup>th</sup> M-step

$L = 5$
A Running Example

$L = 20$

The 20$^{th}$ M-step
Theory

Let’s take the latent variable view of mixture of Gaussians

\[ p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \]

- Indicator (selecting) variable

\[ \mathbf{z} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \]

\[ p(x, z) = \prod_{k=1}^{K} \pi_k^{z_k} \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k} \]

\[ p(x) = \sum_{z} p(x, z) \]

Note: the idea of data augmentation is influential in statistics and machine learning!
Theory

- Re-visit the log-likelihood

\[ \log p(D|\Theta) = \sum_{n=1}^{N} \log \left( \sum_{z_n} p(x_n, z_n) \right) \]

- Jensen’s inequality

\[
\log \frac{x_1 + x_2}{2} \geq \frac{\log x_1 + \log x_2}{2}
\]
Theory

- Re-visit the log-likelihood

\[
\log p(D|\Theta) = \sum_{n=1}^{N} \log \left( \sum_{z_n} p(x_n, z_n) \right)
\]

- Jensen’s inequality

\[
\log \mathbb{E}_{p(x)}[x] \geq \mathbb{E}_{p(x)}[\log x]
\]
Theory

- Re-visit the log-likelihood

\[
\log p(\mathcal{D}|\Theta) = \sum_{n=1}^{N} \log \left( \sum_{\mathbf{z}_n} p(\mathbf{x}_n, \mathbf{z}_n) \right)
\]

- Jensen’s inequality

\[
\log \mathbb{E}_p(x)[x] \geq \mathbb{E}_p(x)[\log x]
\]

- How to apply?

\[
\log p(\mathcal{D}|\Theta) = \sum_{n=1}^{N} \log \left( \sum_{\mathbf{z}_n} q(\mathbf{z}_n) \frac{p(\mathbf{x}_n, \mathbf{z}_n)}{q(\mathbf{z}_n)} \right)
\]

\[
\geq \sum_{n=1}^{N} \sum_{\mathbf{z}_n} q(\mathbf{z}_n) \log \left( \frac{p(\mathbf{x}_n, \mathbf{z}_n)}{q(\mathbf{z}_n)} \right)
\]
Theory

◆ What we have is a lower bound

$$\log p(D|\Theta) \geq \sum_{n=1}^{N} \sum_{z_n} q(z_n) \log \left( \frac{p(x_n, z_n)}{q(z_n)} \right) \triangleq \mathcal{L}(\Theta, q(Z))$$

◆ What’s the GAP?

$$\mathcal{L}(\Theta, q(Z)) = \sum_{n=1}^{N} \left\{ \sum_{z_n} q(z_n) \log p(x_n, z_n) - \sum_{z_n} q(z_n) \log q(z_n) \right\}$$

$$= \sum_{n=1}^{N} \left\{ \sum_{z_n} q(z_n) \log \left( \frac{p(x_n, z_n)}{p(x_n)} \right) + \log p(x_n) - \sum_{z_n} q(z_n) \log q(z_n) \right\}$$

$$= \log p(D|\Theta) + \sum_{n=1}^{N} \left\{ \sum_{z_n} q(z_n) \log p(z_n|x_n) - \sum_{z_n} q(z_n) \log q(z_n) \right\}$$

$$= \log p(D|\Theta) - KL(q(Z)||p(Z|D))$$
Theory

What we have is a lower bound

$$\log p(D|\Theta) \geq \sum_{n=1}^{N} \sum_{z_n} q(z_n) \log \left( \frac{p(x_n, z_n)}{q(z_n)} \right) \triangleq \mathcal{L}(\Theta, q(Z))$$

What’s the GAP?

$$\log p(D|\Theta) - \mathcal{L}(\Theta, q(Z)) = \text{KL}(q(Z)||p(Z|D))$$
K-Means

- Initially, the number of clusters must be known, or chosen, to be $K$ say.
- The initial step is the choose a set of $K$ instances as centres of the clusters. Often chosen such that the points are mutually “farthest apart”, in some way.
- Next, the algorithm considers each instance and assigns it to the cluster which is closest.
- The cluster centroids are recalculated either after each instance assignment, or after the whole cycle of re-assignments.
- This process is iterated.
A Running Example

Ask user how many clusters they’d like (e.g., K=5).
A Running Example

- Ask user how many clusters they’d like (e.g., K=5).
- Randomly guess k cluster center locations
A Running Example

- Ask user how many clusters they’d like (e.g., K=5).
- Randomly guess k cluster center locations.
- Each datapoint finds out which center it’s closest to. (Thus each Center “owns” a set of datapoints)

[Slide courtesy: Andrew Moore]
A Running Example

- Ask user how many clusters they’d like (e.g., K=5).
- Randomly guess k cluster center locations.
- Each datapoint finds out which center it’s closest to. (Thus each Center “owns” a set of datapoints)
- Each Center finds the centroid of the points it owns.

[Slide courtesy: Andrew Moore]
A Running Example

- Ask user how many clusters they’d like (e.g., K=5).
- Randomly guess k cluster center locations
- Each datapoint finds out which center it’s closest to. (Thus each Center “owns” a set of datapoints)
- Each Center finds the centroid of the points it owns
- ...and jumps there
- . . .Repeat until terminated!
K-means Questions

- What is it trying to optimize?
- Are we sure it will terminate?
- Are we sure it will find an optimal clustering?
- How should we start it?
- How could we automatically choose the number of centers?
Theory:
K-Means as an Optimization Problem

The opt. problem

\[ \min_{\{\ell_c\}^K_{c=1}} \sum^K_{c=1} \sum_{x \in \ell_c} \| x - \mu_c \|_2^2 \]

s.t. \[ \mu_c = \frac{1}{|\ell_c|} \sum_{x \in \ell_c} x \]

Theorem: K-means iteratively leads to a non-increasing of the objective, until local minimum is achieved

Proof ideas:
- Each operation leads to non-increasing of the objective
- The objective is bounded and the number of clusters is finite
Trying to find a good optimum

- Idea 1: Be careful about where you start
- Idea 2: Do many runs of k-means, each from a different random start configuration
- Many other ideas floating around.
Relation between GMM and K-Means

Small variance asymptotics:

- The EM algorithm for GMM reduces to K-Means under certain conditions:

E-step:
\[
\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}
\]

M-step:
\[
\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n
\]
\[
\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^\top
\]
\[
\pi_k = \frac{N_k}{N}
\]

let \( \Sigma_k = \sigma I \) and \( \sigma \to 0 \)

\[
\gamma(z_{nk}) = \frac{\pi_k \exp\left(-\frac{1}{2\sigma} \|x_n - \mu_k\|_2^2\right)}{\sum_j \pi_j \exp\left(-\frac{1}{2\sigma} \|x_n - \mu_j\|_2^2\right)}
\]
\[
= k^*, \text{ where } k^* = \arg\min_k \|x_n - \mu_k\|_2^2
\]

- Proof?
Single Linkage Hierarchical Clustering

Start with “every point is its own cluster”
Single Linkage Hierarchical Clustering

- Start with “every point is its own cluster”
- Find “most similar” pairs of clusters
Single Linkage Hierarchical Clustering

- Start with “every point is its own cluster”
- Find “most similar” pairs of clusters
- Merge it into a parent cluster
Single Linkage Hierarchical Clustering

- Start with “every point is its own cluster”
- Find “most similar” pairs of clusters
- Merge it into a parent cluster
- Repeat
Single Linkage Hierarchical Clustering

- Start with “every point is its own cluster”
- Find “most similar” pairs of clusters
- Merge it into a parent cluster
- Repeat

[Slide courtesy: Andrew Moore]
Single Linkage Hierarchical Clustering

- Start with “every point is its own cluster”
- Find “most similar” pairs of clusters
- Merge it into a parent cluster
- Repeat

Key Question:
How do we define similarity between clusters?
=> minimum, maximum, or average distance between points in clusters

[Slide courtesy: Andrew Moore]
How many components are good?

- Can we let the data speak for themselves?
  - let data determine model complexity (e.g., the number of components in mixture models)
  - allow model complexity to grow as more data observed
How many components are good?

Can we let the data speak for themselves?

- we will talk about Dirichlet Process (DP) Mixtures
- and nonparametric Bayesian models
Summary

- Gaussian Mixtures and K-means are effective tools to discover clustering structures.
- EM algorithms can be applied to do MLE for GMMs.
- Relationships between GMMs and K-means are discussed.

Unresolved issues

- How to determine the number of components for mixture models?
- How to determine the number of components for K-means?
Materials to Read

- Chap. 9 of Bishop’s PRML book