EM Algorithm

Professor Ameet Talwalkar

Slide Credit: Professor Fei Sha

Outline

1. Administration
2. Review of last lecture
3. GMMs and Incomplete Data
4. EM Algorithm

Grading

- Midterm and Project Proposal grades are available online
- Midterm: Median (88), Mean (84.7), Standard Deviation (13)
- Proposal: Scores from 0-3 (unaccepatable to exceptional; vast majority of projects were 2s)
- HW5 grades available next Tuesday

HW6

- Will be posted online this afternoon
- Due in section on Friday 12/4
- 1-day extension because:
  - I am posting it late
  - One question is on PCA, which I will cover next Tuesday

Upcoming Class Schedule

- Today: EM
- Tuesday, 12/1: PCA
- Thursday, 12/3: In-class office hours for project (9:00-11am)
- Friday 12/4: Nikos section (covers midterm, HW6 questions)
- Friday, 12/11: Poster Presentation + Project Report
  - I will post project report guideline soon

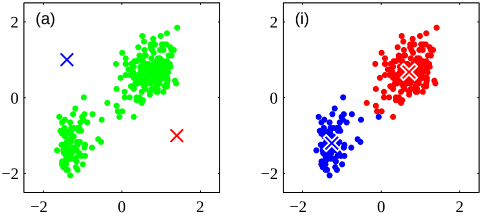
Outline

Clustering

Setup Given $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$ and K, we want to output

- $\{\boldsymbol{\mu}_k\}_{k=1}^K$: centroids of clusters
- $A(\boldsymbol{x}_n) \in \{1, 2, \ldots, K\}$: the cluster membership, i.e., the cluster ID assigned to $\boldsymbol{x}_n$

Toy Example Cluster data into two clusters.



Applications

- Identify communities within social networks
- Find topics in news stories
- Group similiar sequences into gene families

K-means clustering

Intuition Data points assigned to cluster k should be close to $\boldsymbol{\mu}_k$,

Distortion measure (clustering objective function)

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|_2^2$$

where $r_{nk} \in \{0, 1\}$ is an indicator variable

$$r_{nk} = 1 \quad \text{if and only if} \quad A(\boldsymbol{x}_n) = k$$

Algorithm

Minimize distortion measure alternative optimization between $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$

- **Step 0** Initialize $\{\boldsymbol{\mu}_k\}$ to some values

Algorithm

Minimize distortion measure alternative optimization between $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$

- **Step 0** Initialize $\{\boldsymbol{\mu}_k\}$ to some values
- **Step 1** Assume the current value of $\{\boldsymbol{\mu}_k\}$ fixed, minimize J over $\{r_{nk}\}$, which leads to the following cluster assignment rule

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

Algorithm

Minimize distortion measure alternative optimization between $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$

- **Step 0** Initialize $\{\boldsymbol{\mu}_k\}$ to some values
- **Step 1** Assume the current value of $\{\boldsymbol{\mu}_k\}$ fixed, minimize J over $\{r_{nk}\}$, which leads to the following cluster assignment rule

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- **Step 2** Assume the current value of $\{r_{nk}\}$ fixed, minimize J over $\{\boldsymbol{\mu}_k\}$, which leads to the following rule to update the centroids of the clusters

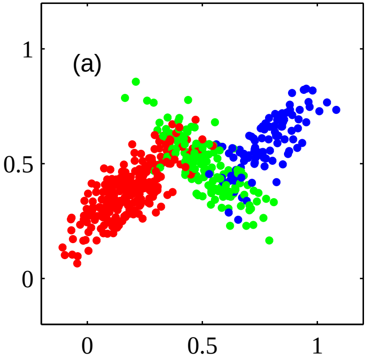
$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\boldsymbol{x}_n}{\sum_n r_{nk}}$$

- **Step 3** Determine whether to stop or return to Step 1

Remarks

- Centroid $\boldsymbol{\mu}_k$ is the mean of data points assigned to the cluster k, hence 'K-means' (you'll look at an alternative in HW6)
- The procedure reduces J in both Step 1 and Step 2 and thus makes improvements on each iteration
- No guarantee we find the global solution; quality of local optimum depends on initial values at Step 0 (k-means++ is a clever approximation algorithm)

Gaussian mixture models: intuition



- Probabalistic interpretation of K-means
- We can model *each* region with a distinct distribution, e.g., Gaussian mixture models (GMMs)

- Can be viewed as generative model

Gaussian mixture models: formal definition

A Gaussian mixture model has the following density function for $\boldsymbol{x}$

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian mixture models: formal definition

A Gaussian mixture model has the following density function for $\boldsymbol{x}$

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- K: the number of Gaussians — they are called (mixture) components
- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$: mean and covariance matrix of the k-th component
- ω_k: mixture weights – priors on each component that satisfy:

$$\forall\ k,\ \omega_k > 0, \quad \text{and} \quad \sum_k \omega_k = 1$$

- Given *unlabeled* data, $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$, we must learn:

Gaussian mixture models: formal definition

A Gaussian mixture model has the following density function for $\boldsymbol{x}$

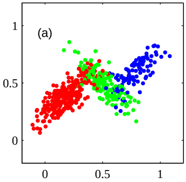
$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- K: the number of Gaussians — they are called (mixture) components
- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$: mean and covariance matrix of the k-th component
- ω_k: mixture weights – priors on each component that satisfy:

$$\forall\ k,\ \omega_k > 0, \quad \text{and} \quad \sum_k \omega_k = 1$$

- Given *unlabeled* data, $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$, we must learn:
  - *parameters* of Gaussians
  - *mixture components*

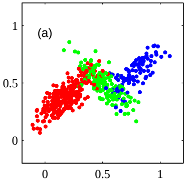
GMMs: example

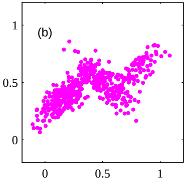


The conditional distribution between $\boldsymbol{x}$ and z (representing color) are

$$p(\boldsymbol{x}|z = red) = N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$
$$p(\boldsymbol{x}|z = blue) = N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$
$$p(\boldsymbol{x}|z = green) = N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

GMMs: example





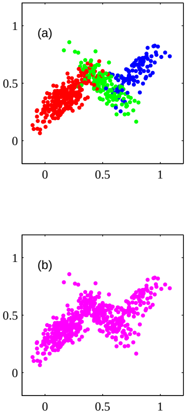
The conditional distribution between $\boldsymbol{x}$ and z (representing color) are

$$p(\boldsymbol{x}|z = red) = N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$
$$p(\boldsymbol{x}|z = blue) = N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$
$$p(\boldsymbol{x}|z = green) = N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

The marginal distribution is thus

$$p(\boldsymbol{x}) = p(red)N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + p(blue)N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + p(green)N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

GMMs: example



The conditional distribution between $\boldsymbol{x}$ and z (representing color) are

$$p(\boldsymbol{x}|z = red) = N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$
$$p(\boldsymbol{x}|z = blue) = N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$
$$p(\boldsymbol{x}|z = green) = N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

The marginal distribution is thus

$$p(\boldsymbol{x}) = p(red)N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + p(blue)N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + p(green)N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

Given a model $\boldsymbol{\theta}$, how would we choose a cluster assignment for $\boldsymbol{x}$?

Parameter estimation for GMMs: complete data

GMM Parameters

$$\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$$

Complete Data: We (unrealistically) assume z is observed for every $\boldsymbol{x}$,

$$\mathcal{D}' = \{\boldsymbol{x}_n, z_n\}_{n=1}^N$$

Parameter estimation for GMMs: complete data

GMM Parameters

$$\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

Complete Data: We (unrealistically) assume z is observed for every $\boldsymbol{x}$,

$$\mathcal{D}' = \{\boldsymbol{x}_n, z_n\}_{n=1}^{N}$$

MLE: Maximize the complete likelihood

$$\boldsymbol{\theta} = \arg\max \log \mathcal{D}' = \sum_n \log p(\boldsymbol{x}_n, z_n)$$

Parameter estimation for GMMs: complete data

Group likelihood by values of z_n

$$\sum_n \log p(\boldsymbol{x}_n, z_n) = \sum_n \log p(z_n)p(\boldsymbol{x}_n|z_n) = \sum_k \sum_{n:z_n=k} \log p(z_n)p(\boldsymbol{x}_n|z_n)$$

Parameter estimation for GMMs: complete data

Group likelihood by values of z_n

$$\sum_n \log p(\boldsymbol{x}_n, z_n) = \sum_n \log p(z_n) p(\boldsymbol{x}_n | z_n) = \sum_k \sum_{n: z_n = k} \log p(z_n) p(\boldsymbol{x}_n | z_n)$$

Introduce dummy variables

$\gamma_{nk} \in \{0, 1\}$ indicate whether $z_n = k$:

$$\sum_n \log p(\boldsymbol{x}_n, z_n) = \sum_k \sum_n \gamma_{nk} \log p(z = k) p(\boldsymbol{x}_n | z = k)$$

In the complete setting the γ_{nk} just add to the notation, but later we will 'relax' these variables and allow them to take on fractional values

Parameter estimation for GMMs: complete data

We can simplify the complete likelihood as follows:

$$
\begin{aligned}
\sum_n \log p(\boldsymbol{x}_n, z_n) &= \sum_k \sum_n \gamma_{nk} \log p(z=k) p(\boldsymbol{x}_n | z=k) \\
&= \sum_k \sum_n \gamma_{nk} \left[\log \omega_k + \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right] \\
&= \sum_k \sum_n \gamma_{nk} \log \omega_k + \sum_k \left\{ \sum_n \gamma_{nk} \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}
\end{aligned}
$$

Parameter estimation for GMMs: complete data

We can simplify the complete likelihood as follows:

$$
\begin{aligned}
\sum_n \log p(\boldsymbol{x}_n, z_n) &= \sum_k \sum_n \gamma_{nk} \log p(z=k) p(\boldsymbol{x}_n | z=k) \\
&= \sum_k \sum_n \gamma_{nk} \left[\log \omega_k + \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right] \\
&= \sum_k \sum_n \gamma_{nk} \log \omega_k + \sum_k \left\{ \sum_n \gamma_{nk} \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}
\end{aligned}
$$

ω_k appears only in left term, and the k-th component's parameters only appear inside braces of right term. We can easily compute MLE (exercise):

$$
\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \boldsymbol{\mu}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \boldsymbol{x}_n
$$

$$
\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}
$$

What's the intuition?

Intuition

Since γ_{nk} is binary, the previous solution is simply:

- For ω_k: count the number of data points whose z_n is k and divide by the total number of data points (note that $\sum_k \sum_n \gamma_{nk} = N$)
- For $\boldsymbol{\mu}_k$: get all the data points whose z_n is k, compute their mean
- For $\boldsymbol{\Sigma}_k$: get all the data points whose z_n is k, compute their covariance matrix

This intuition is going to help us to develop an algorithm for estimating $\boldsymbol{\theta}$ when we do not know z_n (incomplete data).

Outline

Parameter estimation for GMMs: Incomplete data

GMM Parameters

$$\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

Incomplete Data

Our data contains observed and unobserved data, and hence is incomplete

- Observed: $\mathcal{D} = \{\boldsymbol{x}_n\}$
- Unobserved (hidden): $\{\boldsymbol{z}_n\}$

Parameter estimation for GMMs: Incomplete data

GMM Parameters

$$\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

Incomplete Data

Our data contains observed and unobserved data, and hence is incomplete

- Observed: $\mathcal{D} = \{\boldsymbol{x}_n\}$
- Unobserved (hidden): $\{\boldsymbol{z}_n\}$

Goal Obtain the maximum likelihood estimate of $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = \arg\max \ell(\boldsymbol{\theta}) = \arg\max \log \mathcal{D} = \arg\max \sum_n \log p(\boldsymbol{x}_n|\boldsymbol{\theta})$$

Parameter estimation for GMMs: Incomplete data

GMM Parameters

$$\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$$

Incomplete Data

Our data contains observed and unobserved data, and hence is incomplete

- Observed: $\mathcal{D} = \{\boldsymbol{x}_n\}$
- Unobserved (hidden): $\{\boldsymbol{z}_n\}$

Goal Obtain the maximum likelihood estimate of $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = \arg\max \ell(\boldsymbol{\theta}) = \arg\max \log \mathcal{D} = \arg\max \sum_n \log p(\boldsymbol{x}_n|\boldsymbol{\theta})$$

$$= \arg\max \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})$$

The objective function $\ell(\boldsymbol{\theta})$ is called the *incomplete* log-likelihood.

Issue with Incomplete log-likelihood

No simple way to optimize the incomplete log-likelihood (exercise: try to take derivative with respect to parameters, set it to zero and solve)

EM algorithm provides a strategy for iteratively optimizing this function

Two steps as they apply to GMM:

- E-step: 'guess' values of the z_n using existing values of $\boldsymbol{\theta}$
- M-step: solve for new values of $\boldsymbol{\theta}$ given imputed values for z_n

Issue with Incomplete log-likelihood

No simple way to optimize the incomplete log-likelihood (exercise: try to take derivative with respect to parameters, set it to zero and solve)

EM algorithm provides a strategy for iteratively optimizing this function

Two steps as they apply to GMM:

- E-step: 'guess' values of the z_n using existing values of $\boldsymbol{\theta}$
- M-step: solve for new values of $\boldsymbol{\theta}$ given imputed values for z_n (maximize complete likelihood!)

E-step: Soft cluster assignments

We define γ_{nk} as $p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\theta})$

- This is the posterior distribution of z_n given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}$
- Recall that in complete data setting γ_{nk} was binary
- Now it's a "soft" assignment of $\boldsymbol{x}_n$ to k-th component, with $\boldsymbol{x}_n$ assigned to each component with some probability

E-step: Soft cluster assignments

We define γ_{nk} as $p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\theta})$

- This is the posterior distribution of z_n given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}$
- Recall that in complete data setting γ_{nk} was binary
- Now it's a "soft" assignment of $\boldsymbol{x}_n$ to k-th component, with $\boldsymbol{x}_n$ assigned to each component with some probability

Given an estimate of $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$, we can compute γ_{nk} as follows:

$$\gamma_{nk} = p(z_n = k|\boldsymbol{x}_n)$$

$$=$$

E-step: Soft cluster assignments

We define γ_{nk} as $p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\theta})$

- This is the posterior distribution of z_n given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}$
- Recall that in complete data setting γ_{nk} was binary
- Now it's a "soft" assignment of $\boldsymbol{x}_n$ to k-th component, with $\boldsymbol{x}_n$ assigned to each component with some probability

Given an estimate of $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$, we can compute γ_{nk} as follows:

$$\begin{aligned}\gamma_{nk} &= p(z_n = k|\boldsymbol{x}_n) \\ &= \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)}\end{aligned}$$

E-step: Soft cluster assignments

We define γ_{nk} as $p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\theta})$

- This is the posterior distribution of z_n given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}$
- Recall that in complete data setting γ_{nk} was binary
- Now it's a "soft" assignment of $\boldsymbol{x}_n$ to k-th component, with $\boldsymbol{x}_n$ assigned to each component with some probability

Given an estimate of $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$, we can compute γ_{nk} as follows:

$$\begin{aligned}\gamma_{nk} &= p(z_n = k|\boldsymbol{x}_n) \\ &= \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)} \\ &= \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{\sum_{k'=1}^K p(\boldsymbol{x}_n|z_n = k')p(z_n = k')}\end{aligned}$$

M-step: Maximimize complete likelihood

Recall definition of complete likelihood from earlier:

$$\sum_n \log p(\boldsymbol{x}_n, z_n) = \sum_k \sum_n \gamma_{nk} \log \omega_k + \sum_k \left\{ \sum_n \gamma_{nk} \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Previously γ_{nk} was binary, but now we define $\gamma_{nk} = p(z_n = k | \boldsymbol{x}_n)$ (E-step)

M-step: Maximimize complete likelihood

Recall definition of complete likelihood from earlier:

$$\sum_n \log p(\boldsymbol{x}_n, z_n) = \sum_k \sum_n \gamma_{nk} \log \omega_k + \sum_k \left\{ \sum_n \gamma_{nk} \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Previously γ_{nk} was binary, but now we define $\gamma_{nk} = p(z_n = k | \boldsymbol{x}_n)$ (E-step)

We get the same simple expression for the MLE as before!

$$\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \boldsymbol{\mu}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

Intuition: Each point now contributes some fractional component to each of the parameters, with weights determined by γ_{nk}

EM procedure for GMM

Alternate between estimating γ_{nk} and estimating $\boldsymbol{\theta}$

- Initialize $\boldsymbol{\theta}$ with some values (random or otherwise)
- Repeat
  - E-Step: Compute γ_{nk} using the current $\boldsymbol{\theta}$
  - M-Step: Update $\boldsymbol{\theta}$ using the γ_{nk} we just computed
- Until Convergence

EM procedure for GMM

Alternate between estimating γ_{nk} and estimating $\boldsymbol{\theta}$

- Initialize $\boldsymbol{\theta}$ with some values (random or otherwise)
- Repeat
  - E-Step: Compute γ_{nk} using the current $\boldsymbol{\theta}$
  - M-Step: Update $\boldsymbol{\theta}$ using the γ_{nk} we just computed
- Until Convergence

Questions to be answered next

- How does GMM relate to K-means?
- Is this procedure reasonable, i.e., are we optimizing a sensible criterion?
- Will this procedure converge?

GMMs and K-means

GMMs provide probabilistic interpretation for K-means

GMMs and K-means

GMMs provide probabilistic interpretation for K-means

GMMs reduce to K-means under the following assumptions (in which case EM for GMM parameter estimation simplifies to K-means):

- Assume all Gaussians have $\sigma^2 \boldsymbol{I}$ covariance matrices
- Further assume $\sigma \to 0$, so we only need to estimate $\boldsymbol{\mu}_k$, i.e., means

K-means is often called "hard" GMM or GMMs is called "soft" K-means

The posterior γ_{nk} provides a probabilistic assignment for $\boldsymbol{x}_n$ to cluster k

Outline

EM algorithm: motivation and setup

- EM is a general procedure to estimate parameters for probabilistic models with hidden/latent variables
- Suppose the model is given by a joint distribution

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$$

EM algorithm: motivation and setup

- EM is a general procedure to estimate parameters for probabilistic models with hidden/latent variables
- Suppose the model is given by a joint distribution

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$$

- Given incomplete data $\mathcal{D} = \{\boldsymbol{x}_n\}$ our goal is to compute MLE of $\boldsymbol{\theta}$:

$$\begin{aligned}\boldsymbol{\theta} &= \arg\max \log \mathcal{D} = \arg\max \sum_n \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) \\ &= \arg\max \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})\end{aligned}$$

The objective function $\ell(\boldsymbol{\theta})$ is called *incomplete* log-likelihood

A lower bound

- log-sum form of incomplete log-likelihood is difficult to work with
- EM: construct lower bound on $\ell(\boldsymbol{\theta})$ (E-step) and optimize it (M-step)

A lower bound

- log-sum form of incomplete log-likelihood is difficult to work with
- EM: construct lower bound on $\ell(\boldsymbol{\theta})$ (E-step) and optimize it (M-step)
- If we define $q(\boldsymbol{z})$ as a distribution over $\boldsymbol{z}$, then

$$\ell(\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})$$

A lower bound

- log-sum form of incomplete log-likelihood is difficult to work with
- EM: construct lower bound on $\ell(\boldsymbol{\theta})$ (E-step) and optimize it (M-step)
- If we define $q(\boldsymbol{z})$ as a distribution over $\boldsymbol{z}$, then

$$\begin{aligned}\ell(\boldsymbol{\theta}) &= \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}) \\ &= \sum_n \log \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}\end{aligned}$$

A lower bound

- log-sum form of incomplete log-likelihood is difficult to work with
- EM: construct lower bound on $\ell(\boldsymbol{\theta})$ (E-step) and optimize it (M-step)
- If we define $q(\boldsymbol{z})$ as a distribution over $\boldsymbol{z}$, then

$$
\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}) \\
&= \sum_n \log \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)} \\
&\geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)}
\end{aligned}
$$

- Last step follows from Jensen's inequality, i.e., $f(\mathbb{E}X) \geq \mathbb{E}f(X)$ for concave function f

GMM Example

- Consider the previous model where $\boldsymbol{x}$ could be from 3 regions
- We can choose $q(\boldsymbol{z})$ as any valid distribution
- e.g., $q(z = k) = 1/3$ for any of 3 colors
- e.g., $q(z = k) = 1/2$ for red and blue, 0 for green

Which $q(\boldsymbol{z})$ should we choose?

Which $q(\boldsymbol{z})$ to choose?

$$\ell(\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$
$$\geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

- The lower bound we derived for $\ell(\boldsymbol{\theta})$ holds for all choices of $q(\cdot)$
- We want a *tight* lower bound

Which $q(\boldsymbol{z})$ to choose?

$$\begin{aligned}\ell(\boldsymbol{\theta}) &= \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)} \\ &\geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}\end{aligned}$$

- The lower bound we derived for $\ell(\boldsymbol{\theta})$ holds for all choices of $q(\cdot)$
- We want a *tight* lower bound, and given some current estimate $\boldsymbol{\theta}^t$, we will pick $q(\cdot)$ such that our lower bound holds *with equality* at $\boldsymbol{\theta}^t$
- $f(\mathbb{E}X) = \mathbb{E}f(X)$?

Which $q(\boldsymbol{z})$ to choose?

$$\ell(\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$
$$\geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

- The lower bound we derived for $\ell(\boldsymbol{\theta})$ holds for all choices of $q(\cdot)$
- We want a *tight* lower bound, and given some current estimate $\boldsymbol{\theta}^t$, we will pick $q(\cdot)$ such that our lower bound holds *with equality* at $\boldsymbol{\theta}^t$
- $f(\mathbb{E}X) = \mathbb{E}f(X)$? It is sufficient for X to be a constant random variable!

Which $q(\boldsymbol{z})$ to choose?

$$\ell(\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

$$\geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

- The lower bound we derived for $\ell(\boldsymbol{\theta})$ holds for all choices of $q(\cdot)$
- We want a *tight* lower bound, and given some current estimate $\boldsymbol{\theta}^t$, we will pick $q(\cdot)$ such that our lower bound holds *with equality* at $\boldsymbol{\theta}^t$
- $f(\mathbb{E}X) = \mathbb{E}f(X)$? It is sufficient for X to be a constant random variable!
- Choose $q(\boldsymbol{z}_n) \propto p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)$!

Which $q(\boldsymbol{z})$ to choose?

$$\ell(\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$
$$\geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

- The lower bound we derived for $\ell(\boldsymbol{\theta})$ holds for all choices of $q(\cdot)$
- We want a *tight* lower bound, and given some current estimate $\boldsymbol{\theta}^t$, we will pick $q(\cdot)$ such that our lower bound holds *with equality* at $\boldsymbol{\theta}^t$
- $f(\mathbb{E}X) = \mathbb{E}f(X)$? It is sufficient for X to be a constant random variable!
- Choose $q(\boldsymbol{z}_n) \propto p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)$! Since $q(\cdot)$ is a distribution, we have

$$q(\boldsymbol{z}_n) = \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)}{\sum_k p(\boldsymbol{x}_n, z_n = k|\boldsymbol{\theta})} = \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)}{p(\boldsymbol{x}_n|\boldsymbol{\theta}^t)} = p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)$$

Which $q(\boldsymbol{z})$ to choose?

$$\ell(\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

$$\geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

- The lower bound we derived for $\ell(\boldsymbol{\theta})$ holds for all choices of $q(\cdot)$
- We want a *tight* lower bound, and given some current estimate $\boldsymbol{\theta}^t$, we will pick $q(\cdot)$ such that our lower bound holds *with equality* at $\boldsymbol{\theta}^t$
- $f(\mathbb{E}X) = \mathbb{E}f(X)$? It is sufficient for X to be a constant random variable!
- Choose $q(\boldsymbol{z}_n) \propto p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)$! Since $q(\cdot)$ is a distribution, we have

$$q(\boldsymbol{z}_n) = \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)}{\sum_k p(\boldsymbol{x}_n, z_n = k|\boldsymbol{\theta})} = \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)}{p(\boldsymbol{x}_n|\boldsymbol{\theta}^t)} = p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)$$

- This is the posterior distribution of z_n given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}^t$

E and M Steps

Our simplified expression

$$\ell(\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)}{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t))}$$

E and M Steps

Our simplified expression

$$\ell(\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)}{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t))}$$

E-Step: For all n, compute $q(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$

Why is this called the E-Step?

E and M Steps

Our simplified expression

$$\ell(\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)\log\frac{p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta}^t)}{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t))}$$

E-Step: For all n, compute $q(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$

Why is this called the E-Step? Because we can view it as computing the *expected (complete) log-likelihood*:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)\log p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta}) = \mathbb{E}_q \sum_n \log p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta})$$

E and M Steps

Our simplified expression

$$\ell(\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)}{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t))}$$

E-Step: For all n, compute $q(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$

Why is this called the E-Step? Because we can view it as computing the *expected (complete) log-likelihood*:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}) = \mathbb{E}_q \sum_n \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})$$

M-Step: Maximize $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$, i.e., $\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$

Example: applying EM to GMMs

What is the E-step in GMM?

$$\gamma_{nk} = p(z = k|\boldsymbol{x}_n; \boldsymbol{\theta}^{(t)})$$

What is the M-step in GMM? The Q-function is

$$\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) &= \sum_n \sum_k p(z = k|\boldsymbol{x}_n; \boldsymbol{\theta}^{(t)}) \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \\
&= \sum_n \sum_k \gamma_{nk} \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \\
&= \sum_k \sum_n \gamma_{nk} \log p(z = k) p(\boldsymbol{x}_n|z = k) \\
&= \sum_k \sum_n \gamma_{nk} \left[\log \omega_k + \log N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]
\end{aligned}$$

We have recovered the parameter estimation algorithm for GMMs that we previously discussed

Iterative and monotonic improvement

- We can show that $\ell(\boldsymbol{\theta}^{t+1}) \geq \ell(\boldsymbol{\theta}^t)$
- Recall that we chose $q(\cdot)$ in the E-step such that:

$$\ell(\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)}{q(\boldsymbol{z}_n)}$$

- However, in the M-step, $\boldsymbol{\theta}^{t+1}$ is chosen to maximize the right hand side of the equation, thus proving our desired result
- Note: the EM procedure converges but only to a local optimum