10-701 Introduction to Machine Learning

The EM Algorithm

Spring 2019

Ameet Talwalkar
(slide credit: Virginia Smith)

Outline

1. Gaussian mixture models

2. GMMs and Incomplete Data

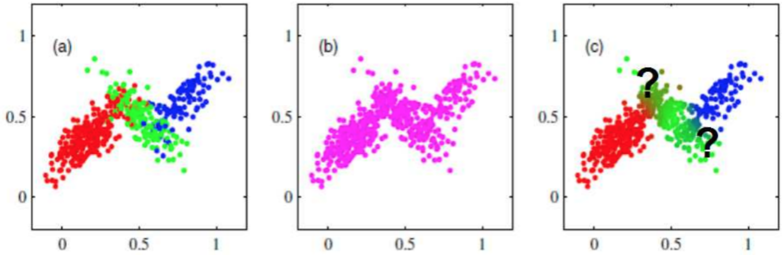
3. EM Algorithm

Gaussian mixture models

Potential issue with k-means . . .

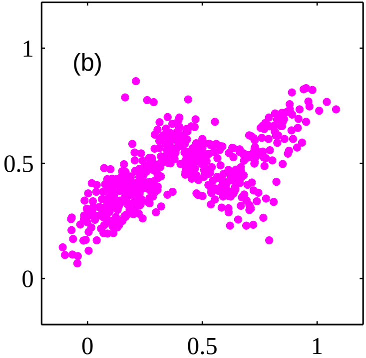
Data points are assigned *deterministically* to one (and only one) cluster

In reality, clusters may overlap, and it may be better to identify the *probability* that a point belongs to each cluster



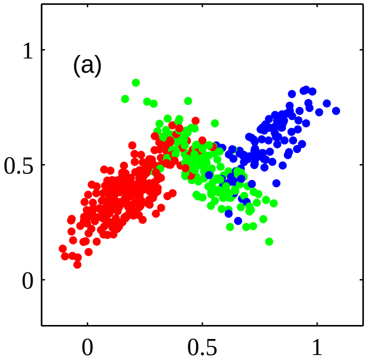
Probabilistic interpretation of clustering?

How can we model $p(\boldsymbol{x})$ to reflect our intuition that points stay close to their cluster centers?



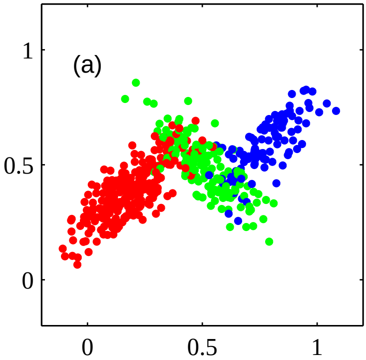
- Points seem to form 3 clusters
- We cannot model $p(\boldsymbol{x})$ with simple and known distributions
- E.g., the data is not a Gaussian b/c we have 3 distinct concentrated regions

Gaussian mixture models: intuition



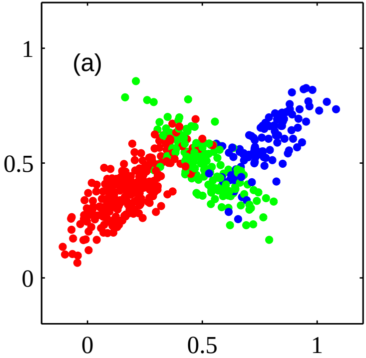
- **Key idea:** Model *each* region with a distinct distribution

Gaussian mixture models: intuition



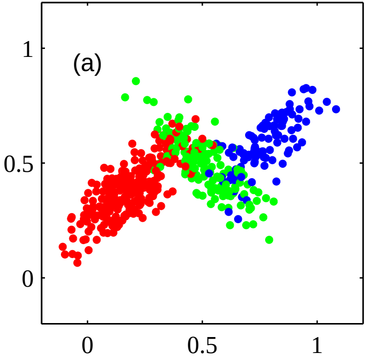
- **Key idea:** Model *each* region with a distinct distribution
- Can use Gaussians — Gaussian mixture models (GMMs)

Gaussian mixture models: intuition



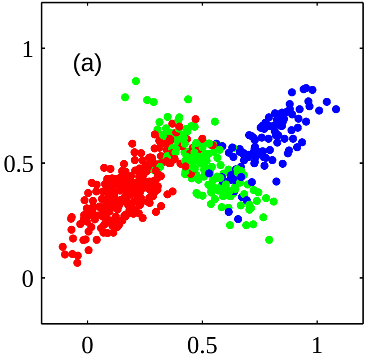
- **Key idea:** Model *each* region with a distinct distribution
- Can use Gaussians — Gaussian mixture models (GMMs)

Gaussian mixture models: intuition



- **Key idea:** Model *each* region with a distinct distribution
- Can use Gaussians — Gaussian mixture models (GMMs)
- *However*, we don't know *cluster assignments* (label), *parameters* of Gaussians, or *mixture components*!

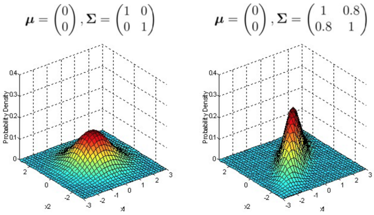
Gaussian mixture models: intuition



- **Key idea:** Model *each* region with a distinct distribution
- Can use Gaussians — Gaussian mixture models (GMMs)
- *However*, we don't know *cluster assignments* (label), *parameters* of Gaussians, or *mixture components*!
- Must learn from *unlabeled* data $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$

Recall: Gaussian (normal) distributions

$$\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$



Gaussian mixture models: formal definition

GMM 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: number of Gaussians — they are called mixture components

Gaussian mixture models: formal definition

GMM 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: number of Gaussians — they are called mixture components
- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$: mean and covariance matrix of k-th component

Gaussian mixture models: formal definition

GMM 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: number of Gaussians — they are called mixture components
- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$: mean and covariance matrix of k-th component
- ω_k: mixture weights (or priors) represent how much each component contributes to final distribution. They satisfy 2 properties:

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

These properties ensure $p(\boldsymbol{x})$ is in fact a probability density function

GMM as the marginal distribution of a joint distribution

Consider the following joint distribution

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

where z is a discrete random variable taking values between 1 and K.

GMM as the marginal distribution of a joint distribution

Consider the following joint distribution

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

where z is a discrete random variable taking values between 1 and K.

Denote

$$\omega_k = p(z = k)$$

Now, assume the conditional distributions are Gaussian distributions

$$p(\boldsymbol{x}|z = k) = N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

GMM as the marginal distribution of a joint distribution

Consider the following joint distribution

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

where z is a discrete random variable taking values between 1 and K.

Denote

$$\omega_k = p(z = k)$$

Now, assume the conditional distributions are Gaussian distributions

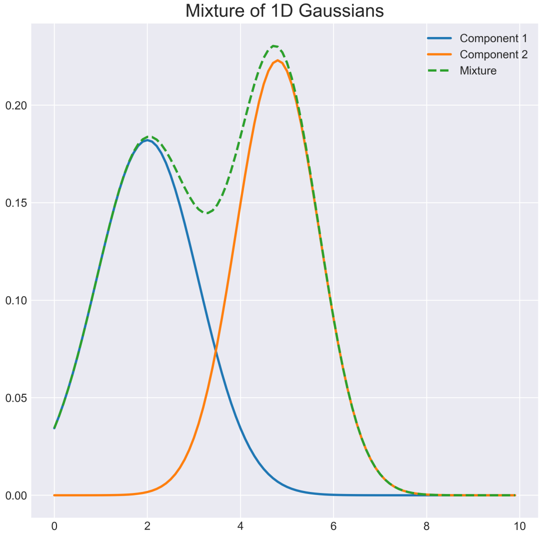
$$p(\boldsymbol{x}|z = k) = N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Then, the marginal distribution of $\boldsymbol{x}$ is

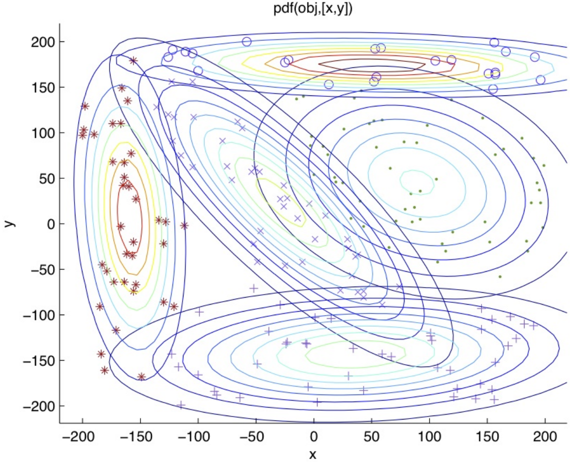
$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Namely, the Gaussian mixture model

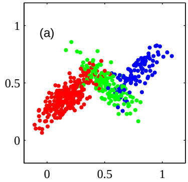
Gaussian mixtures in 1D



Gaussian mixture model for clustering



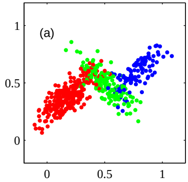
GMMs: example

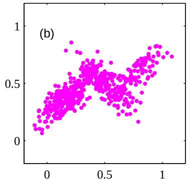


The conditional distribution between $\boldsymbol{x}$ and z (representing color) are

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

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

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

Parameter estimation for Gaussian mixture models

The parameters in GMMs are:

Parameter estimation for Gaussian mixture models

The parameters in GMMs are:

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

Let's first consider the simple/unrealistic case where *we have labels z*

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

- $\mathcal{D}'$ is the **complete** data
- $\mathcal{D}$ the **incomplete** data

How can we learn our parameters?

Parameter estimation for Gaussian mixture models

The parameters in GMMs are:

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

Let's first consider the simple/unrealistic case where *we have labels z*

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

- $\mathcal{D}'$ is the **complete** data
- $\mathcal{D}$ the **incomplete** data

How can we learn our parameters?

Given $\mathcal{D}'$, the maximum likelihood estimation of the $\boldsymbol{\theta}$ is given by

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

Parameter estimation for GMMs: complete data

The complete likelihood is decomposable

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

where we have grouped data by cluster labels z_n.

Parameter estimation for GMMs: complete data

The complete likelihood is decomposable

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

where we have grouped data by cluster labels z_n.

Let $\gamma_{nk} \in \{0, 1\}$ be a binary variable that indicates whether $z_n = k$:

Parameter estimation for GMMs: complete data

The complete likelihood is decomposable

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

where we have grouped data by cluster labels z_n.

Let $\gamma_{nk} \in \{0, 1\}$ be a binary variable that indicates whether $z_n = k$:

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

Parameter estimation for GMMs: complete data

The complete likelihood is decomposable

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

where we have grouped data by cluster labels z_n.

Let $\gamma_{nk} \in \{0, 1\}$ be a binary variable that indicates whether $z_n = k$:

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

Note: 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

From our previous discussion, we have

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

Parameter estimation for GMMs: complete data

From our previous discussion, we have

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

Regrouping, we have

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

Parameter estimation for GMMs: complete data

From our previous discussion, we have

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

Regrouping, we have

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

The term inside the braces depends on k-th component's parameters. It is now easy to show that (left as an exercise) the MLE is:

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

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

What's the intuition?

Intuition

Since γ_{nk} is binary, the previous solution is nothing but:

- ω_k: fraction of total data points whose cluster label z_n is k
  - note that $\sum_k \sum_n \gamma_{nk} = N$
- $\boldsymbol{\mu}_k$: mean of all data points whose z_n is k
- $\boldsymbol{\Sigma}_k$: covariance of all data points whose z_n is k

Intuition

Since γ_{nk} is binary, the previous solution is nothing but:

- ω_k: fraction of total data points whose cluster label z_n is k
  - note that $\sum_k \sum_n \gamma_{nk} = N$
- $\boldsymbol{\mu}_k$: mean of all data points whose z_n is k
- $\boldsymbol{\Sigma}_k$: covariance of all data points whose z_n is k

Recall that this depends on us knowing the true cluster labels z_n

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

GMMs and Incomplete Data

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 $\boldsymbol{\theta}$:

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

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)

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

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 (i.e., 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 $\boldsymbol{\theta}$

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

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

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!

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

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

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!

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

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

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 $\sigma \rightarrow 0$, 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

GMMs vs. k-means

Pros/Cons

- k-means is a simpler, more straightforward method, but might not be as accurate because of deterministic clustering

GMMs vs. k-means

Pros/Cons

- k-means is a simpler, more straightforward method, but might not be as accurate because of deterministic clustering
- GMMs can be more accurate, as they model more information (soft clustering, variance), but can be more expensive to compute

GMMs vs. k-means

Pros/Cons

- k-means is a simpler, more straightforward method, but might not be as accurate because of deterministic clustering
- GMMs can be more accurate, as they model more information (soft clustering, variance), but can be more expensive to compute
- Both methods have a similar set of practical issues (having to select k, the distance, and the initialization)

EM Algorithm

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 $\boldsymbol{\theta}$:

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

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

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

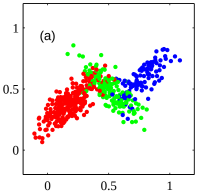
A lower bound

- 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

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

- 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 $q(z)$ to choose?

Recall:

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

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

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

Which $q(z)$ to choose?

Recall:

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

- 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 $q_n(\cdot)$ such that our lower bound holds *with equality* at $\boldsymbol{\theta}^t$
- $f(\mathbb{E}X) = \mathbb{E}f(X)$?

Which $q(z)$ to choose?

Recall:

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

- 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 $q_n(\cdot)$ 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 $q(z)$ to choose?

Recall:

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

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

- 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 $q_n(\cdot)$ 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 $q_n(\boldsymbol{z}_n) \propto p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)$!

Which $q(z)$ to choose?

Recall:

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

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

- 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 $q_n(\cdot)$ 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 $q_n(\boldsymbol{z}_n) \propto p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)$! Since $q_n(\cdot)$ is a distribution, we have

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

Which $q(z)$ to choose?

Recall:

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

- 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 $q_n(\cdot)$ 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 $q_n(\boldsymbol{z}_n) \propto p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)$! Since $q_n(\cdot)$ is a distribution, we have

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

- 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 $q_n(\boldsymbol{z}_n) = p(\boldsymbol{z}_n | \boldsymbol{x}_n; \boldsymbol{\theta}^t)$

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

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

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

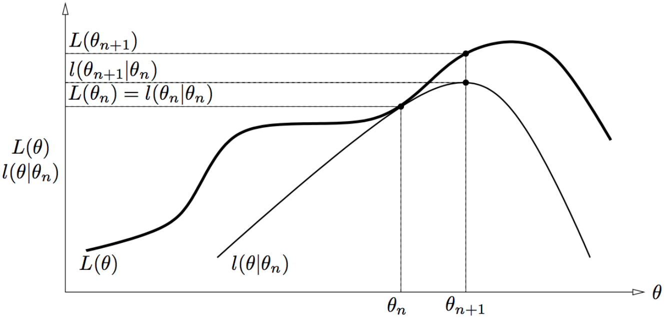
E-Step: For all n, compute $q_n(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$

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., $\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$

EM in Pictures



(Figure from tutorial by Sean Borman)

Example: applying EM to GMMs

What is the E-step in GMM?

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

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

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = \sum_n \sum_k p(z = k | \boldsymbol{x}_n; \boldsymbol{\theta}^{(t)}) \log p(\boldsymbol{x}_n, z = k | \boldsymbol{\theta})$$

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

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) &= \sum_n \sum_k p(z = k|\boldsymbol{x}_n; \boldsymbol{\theta}^{(t)}) \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \\ &= \sum_n \sum_k \gamma_{nk} \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \end{aligned}$$

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

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) &= \sum_n \sum_k p(z = k | \boldsymbol{x}_n; \boldsymbol{\theta}^{(t)}) \log p(\boldsymbol{x}_n, z = k | \boldsymbol{\theta}) \\ &= \sum_n \sum_k \gamma_{nk} \log p(\boldsymbol{x}_n, z = k | \boldsymbol{\theta}) \\ &= \sum_k \sum_n \gamma_{nk} \log p(z = k) p(\boldsymbol{x}_n | z = k) \end{aligned}$$

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 $\ell(\boldsymbol{\theta}^{t+1}) \geq \ell(\boldsymbol{\theta}^{t})$

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:

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

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

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

You should know . . .

- EM is a general procedure for maximizing a likelihood with *latent (unobserved) variables*

You should know ...

- EM is a general procedure for maximizing a likelihood with *latent (unobserved) variables*
- The two steps of EM:

You should know ...

- EM is a general procedure for maximizing a likelihood with *latent (unobserved) variables*
- The two steps of EM:
  - (1) Estimating unobserved data from observed data and current parameters

You should know . . .

- EM is a general procedure for maximizing a likelihood with *latent (unobserved) variables*
- The two steps of EM:
  - (1) Estimating unobserved data from observed data and current parameters
  - (2) Using this "complete" data to find the maximum likelihood parameter estimates

You should know ...

- EM is a general procedure for maximizing a likelihood with *latent (unobserved) variables*
- The two steps of EM:
  - (1) Estimating unobserved data from observed data and current parameters
  - (2) Using this "complete" data to find the maximum likelihood parameter estimates
- Pros: Guaranteed to converge, no parameters to tune (e.g., compared to gradient methods)

You should know . . .

- EM is a general procedure for maximizing a likelihood with *latent (unobserved) variables*
- The two steps of EM:
  - (1) Estimating unobserved data from observed data and current parameters
  - (2) Using this "complete" data to find the maximum likelihood parameter estimates
- Pros: Guaranteed to converge, no parameters to tune (e.g., compared to gradient methods)
- Cons: Can get stuck in local optima, can be expensive

You should know . . .

- EM is a general procedure for maximizing a likelihood with *latent (unobserved) variables*
- The two steps of EM:
  - (1) Estimating unobserved data from observed data and current parameters
  - (2) Using this "complete" data to find the maximum likelihood parameter estimates
- Pros: Guaranteed to converge, no parameters to tune (e.g., compared to gradient methods)
- Cons: Can get stuck in local optima, can be expensive
- Why is EM useful for unsupervised learning?

You should know . . .

- EM is a general procedure for maximizing a likelihood with *latent (unobserved) variables*
- The two steps of EM:
  - (1) Estimating unobserved data from observed data and current parameters
  - (2) Using this "complete" data to find the maximum likelihood parameter estimates
- Pros: Guaranteed to converge, no parameters to tune (e.g., compared to gradient methods)
- Cons: Can get stuck in local optima, can be expensive
- Why is EM useful for unsupervised learning?
  - EM is a general method to deal with hidden data; we have studied it in the context of hidden *labels* (unsupervised learning)