Expectation Maximization
Unsupervised Learning
Machine Learning

Torsten Möller
Reading

• Chapter 8 of “Machine Learning—An Algorithmic Perspective” by Marsland
• Chapter 9 of “Pattern Recognition and Machine Learning” by Bishop
• Chapter 14 of “The Elements of Statistical Learning” by Hastie, Tibshirani, Friedman
Learning Parameters to Probability Distributions

- Learning thus far meant to properly set parameters of a model based on training data.
- However, in many settings not all variables are observed (labelled) in the training data: $x_i = (x_i, h_i)$
  - e.g. Speech recognition: have speech signals, but not phoneme labels.
  - e.g. Object recognition: have object labels (car, bicycle), but not part labels (wheel, door, seat).
- Unobserved variables are called latent variables.

figs from Fergus et al.
Outline

K-Means

Gaussian Mixture Models

Expectation-Maximization
Outline

K-Means

Gaussian Mixture Models

Expectation-Maximization

©Möller/Mori
Unsupervised Learning

- We will start with an unsupervised learning (clustering) problem:
- Given a dataset \( \{ \mathbf{x}_1, \ldots, \mathbf{x}_N \} \), each \( \mathbf{x}_i \in \mathbb{R}^D \), partition the dataset into \( K \) clusters
  - Intuitively, a **cluster** is a group of points, which are close together and far from others
Distortion Measure

- Formally, introduce prototypes (or cluster centers) $\mathbf{\mu}_k \in \mathbb{R}^D$
- Use binary $r_{nk}$, 1 if point $n$ is in cluster $k$, 0 otherwise (1-of-$K$ coding scheme again)
- Find $\{\mathbf{\mu}_k\}, \{r_{nk}\}$ to minimize distortion measure:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| \mathbf{x}_n - \mathbf{\mu}_k \|^2$$
Minimizing Distortion Measure

• Minimizing $J$ directly is hard

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

• However, two things are easy
  • If we know $\boldsymbol{\mu}_k$, minimizing $J$ wrt $r_{nk}$
  • If we know $r_{nk}$, minimizing $J$ wrt $\boldsymbol{\mu}_k$

• This suggests an iterative procedure
  • Start with initial guess for $\boldsymbol{\mu}_k$
  • Iteration of two steps:
    • Minimize $J$ wrt $r_{nk}$
    • Minimize $J$ wrt $\boldsymbol{\mu}_k$
  • Rinse and repeat until convergence
Minimizing Distortion Measure

- Minimizing $J$ directly is hard

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2$$

- However, two things are easy
  - If we know $\mu_k$, minimizing $J$ wrt $r_{nk}$
  - If we know $r_{nk}$, minimizing $J$ wrt $\mu_k$

- This suggests an iterative procedure
  - Start with initial guess for $\mu_k$
  - Iteration of two steps:
    - Minimize $J$ wrt $r_{nk}$
    - Minimize $J$ wrt $\mu_k$
  - Rinse and repeat until convergence
Minimizing Distortion Measure

- Minimizing $J$ directly is hard

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2$$

- However, two things are easy
  - If we know $\mu_k$, minimizing $J$ wrt $r_{nk}$
  - If we know $r_{nk}$, minimizing $J$ wrt $\mu_k$

- This suggests an iterative procedure
  - Start with initial guess for $\mu_k$
  - Iteration of two steps:
    - Minimize $J$ wrt $r_{nk}$
    - Minimize $J$ wrt $\mu_k$
  - Rinse and repeat until convergence

©Möller/Mori
Determining Membership Variables

- Step 1 in an iteration of K-means is to minimize distortion measure $J$ wrt cluster membership variables $r_{nk}$

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| \mathbf{x}_n - \mu_k \|^2$$

- Terms for different data points $\mathbf{x}_n$ are independent, for each data point set $r_{nk}$ to minimize

$$\sum_{k=1}^{K} r_{nk} \| \mathbf{x}_n - \mu_k \|^2$$

- Simply set $r_{nk} = 1$ for the cluster center $\mu_k$ with smallest distance
Determining Membership Variables

• Step 1 in an iteration of K-means is to minimize distortion measure $J$ wrt cluster membership variables $r_{nk}$

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2$$

• Terms for different data points $x_n$ are independent, for each data point set $r_{nk}$ to minimize

$$\sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2$$

• Simply set $r_{nk} = 1$ for the cluster center $\mu_k$ with smallest distance
Determining Membership Variables

- Step 1 in an iteration of K-means is to minimize distortion measure $J$ wrt cluster membership variables $r_{nk}$

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2$$

- Terms for different data points $x_n$ are independent, for each data point set $r_{nk}$ to minimize

$$\sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2$$

- Simply set $r_{nk} = 1$ for the cluster center $\mu_k$ with smallest distance
Determining Cluster Centers

- Step 2: fix $r_{nk}$, minimize $J$ wrt the cluster centers $\mu_k$

\[ J = \sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk} \| x_n - \mu_k \|^2 \]  

- So we can minimize wrt each $\mu_k$ separately

- Take derivative, set to zero:

\[ 2 \sum_{n=1}^{N} r_{nk} (x_n - \mu_k) = 0 \]

\[ \Leftrightarrow \mu_k = \frac{\sum_{n} r_{nk} x_n}{\sum_{n} r_{nk}} \]

i.e. mean of datapoints $x_n$ assigned to cluster $k$.
Determining Cluster Centers

- Step 2: fix $r_{nk}$, minimize $J$ wrt the cluster centers $\mu_k$

$$J = \sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk} \| \mathbf{x}_n - \mu_k \|^2$$

- So we can minimize wrt each $\mu_k$ separately
- Take derivative, set to zero:

$$2 \sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \mu_k) = 0$$

$$\Leftrightarrow \mu_k = \frac{\sum_{n} r_{nk} \mathbf{x}_n}{\sum_{n} r_{nk}}$$

i.e. mean of datapoints $\mathbf{x}_n$ assigned to cluster $k$
K-means Algorithm

- Start with initial guess for $\mu_k$
- Iteration of two steps:
  - Minimize $J$ wrt $r_{nk}$
    - Assign points to nearest cluster center
  - Minimize $J$ wrt $\mu_k$
    - Set cluster center as average of points in cluster
- Rinse and repeat until convergence
K-Means example
K-means example
K-Means example
K-means example

(d)
K-means example
K-means example

(f)
K-means example
K-means example
K-means example

Next step doesn’t change membership – stop
K-means Convergence

- Repeat steps until no change in cluster assignments
- For each step, value of $J$ either goes down, or we stop
- Finite number of possible assignments of data points to clusters, so we are guaranteed to converge eventually
- Note it may be a local maximum rather than a global maximum to which we converge
K-means Example - Image Segmentation

- K-means clustering on pixel colour values
- Pixels in a cluster are coloured by cluster mean
- Represent each pixel (e.g. 24-bit colour value) by a cluster number (e.g. 4 bits for $K = 10$), compressed version
- This technique known as vector quantization
  - Represent vector (in this case from RGB, $\mathbb{R}^3$) as a single discrete value
Outline

K-Means

Gaussian Mixture Models

Expectation-Maximization
Hard Assignment vs. Soft Assignment

- In the K-means algorithm, a **hard assignment** of points to clusters is made.
- However, for points near the decision boundary, this may not be such a good idea.
- Instead, we could think about making a **soft assignment** of points to clusters.
• The **Gaussian mixture model** (or *mixture of Gaussians* MoG) models the data as a combination of Gaussians

• Above shows a dataset generated by drawing samples from three different Gaussians
• The mixture of Gaussians is a generative model
• To generate a datapoint $x_n$, we first generate a value for a discrete variable $z_n \in \{1, \ldots, K\}$
• We then generate a value $x_n \sim \mathcal{N}(x|\mu_k, \Sigma_k)$ for the corresponding Gaussian
Graphical Model

- Full graphical model using plate notation
  - Note \( z_n \) is a latent variable, unobserved
- Need to give conditional distributions \( p(z_n) \) and \( p(x_n | z_n) \)
- The one-of-\( K \) representation is helpful here: \( z_{nk} \in \{0, 1\} \), 
  \( z_n = (z_{n1}, \ldots, z_{nK}) \)
• Use a Bernoulli distribution for $p(z_n)$
  
  • i.e. $p(z_{nk} = 1) = \pi_k$
  
  • Parameters to this distribution $\{\pi_k\}$
  
  • Must have $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$

  • $p(z_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$
Graphical Model - Observed Variable

- Use a Gaussian distribution for $p(x_n|z_n)$
- Parameters to this distribution $\{\mu_k, \Sigma_k\}$

\[
p(x_n|z_{nk} = 1) = \mathcal{N}(x_n|\mu_k, \Sigma_k)
\]

\[
p(x_n|z_n) = \prod_{k=1}^{K} \mathcal{N}(x_n|\mu_k, \Sigma_k)^{z_{nk}}
\]

©Möller/Mori
Graphical Model - Joint distribution

The full joint distribution is given by:

\[
p(x, z) = \prod_{n=1}^{N} p(z_n)p(x_n | z_n)
\]

\[
= \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(x_n | \mu_k, \Sigma_k)^{z_{nk}}
\]
MoG Marginal over Observed Variables

• The marginal distribution \( p(\mathbf{x}_n) \) for this model is:

\[
 p(\mathbf{x}_n) = \sum_{z_n} p(\mathbf{x}_n, z_n) = \sum_{z_n} p(z_n)p(\mathbf{x}_n | z_n)
\]

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

• A mixture of Gaussians
MoG Conditional over Latent Variable

- The conditional $p(z_{nk} = 1|x_n)$ will play an important role for learning.

- It is denoted by $\gamma(z_{nk})$ can be computed as:

$$
\gamma(z_{nk}) \equiv p(z_{nk} = 1|x_n) = \frac{p(z_{nk} = 1)p(x_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(x_n|z_{nj} = 1)}
$$

$$
= \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}
$$

- $\gamma(z_{nk})$ is the responsibility of component $k$ for datapoint $n$
• The conditional $p(z_{nk} = 1 | \mathbf{x}_n)$ will play an important role for learning.
• It is denoted by $\gamma(z_{nk})$ can be computed as:

$$\gamma(z_{nk}) \equiv p(z_{nk} = 1 | \mathbf{x}_n) = \frac{p(z_{nk} = 1)p(\mathbf{x}_n | z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(\mathbf{x}_n | z_{nj} = 1)} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \mu_j, \Sigma_j)}$$

• $\gamma(z_{nk})$ is the responsibility of component $k$ for datapoint $n$. 

©Möller/Mori
• The conditional $p(z_{nk} = 1|x_n)$ will play an important role for learning.

• It is denoted by $\gamma(z_{nk})$ can be computed as:

$$
\gamma(z_{nk}) \equiv p(z_{nk} = 1|x_n) = \frac{p(z_{nk} = 1)p(x_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(x_n|z_{nj} = 1)}
= \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}
$$

• $\gamma(z_{nk})$ is the responsibility of component $k$ for datapoint $n$. 
MoG Learning

• Given a set of observations \( \{ \mathbf{x}_1, \ldots, \mathbf{x}_N \} \), without the latent variables \( z_n \), how can we learn the parameters?
  • Model parameters are \( \theta = \{ \pi_k, \mu_k, \Sigma_k \} \)

• Answer will be similar to k-means:
  • If we know the latent variables \( z_n \), fitting the Gaussians is easy
  • If we know the Gaussians \( \mu_k, \Sigma_k \), finding the latent variables is easy

• Rather than latent variables, we will use responsibilities \( \gamma(z_{nk}) \)
MoG Learning

• Given a set of observations \( \{x_1, \ldots, x_N\} \), without the latent variables \( z_n \), how can we learn the parameters?
  • Model parameters are \( \theta = \{\pi_k, \mu_k, \Sigma_k\} \)

• Answer will be similar to k-means:
  • If we know the latent variables \( z_n \), fitting the Gaussians is easy
  • If we know the Gaussians \( \mu_k, \Sigma_k \), finding the latent variables is easy

• Rather than latent variables, we will use responsibilities \( \gamma(z_{nk}) \)
MoG Learning

• Given a set of observations \( \{x_1, \ldots, x_N\} \), without the latent variables \( z_n \), how can we learn the parameters?
  • Model parameters are \( \theta = \{\pi_k, \mu_k, \Sigma_k\} \)

• Answer will be similar to k-means:
  • If we know the latent variables \( z_n \), fitting the Gaussians is easy
  • If we know the Gaussians \( \mu_k, \Sigma_k \), finding the latent variables is easy

• Rather than latent variables, we will use responsibilities
  \( \gamma(z_{nk}) \)
MoG Maximum Likelihood Learning

• Given a set of observations \( \{x_1, \ldots, x_N\} \), without the latent variables \( z_n \), how can we learn the parameters?
  • Model parameters are \( \theta = \{ \pi_k, \mu_k, \Sigma_k \} \)

• We can use the maximum likelihood criterion:

\[
\theta_{ML} = \arg \max_{\theta} \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)
\]

\[
= \arg \max_{\theta} \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right\}
\]

• Unfortunately, closed-form solution not possible this time – log of sum rather than log of product
MoG Maximum Likelihood Learning

- Given a set of observations \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}, without the latent variables \mathbf{z}_n, how can we learn the parameters?
  - Model parameters are \( \theta = \{\pi_k, \mu_k, \Sigma_k\} \)
- We can use the maximum likelihood criterion:

\[
\theta_{ML} = \arg \max_\theta \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)
\]

\[
= \arg \max_\theta \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k) \right\}
\]

- Unfortunately, closed-form solution not possible this time – log of sum rather than log of product
MoG Maximum Likelihood Learning - Problem

• Maximum likelihood criterion, 1-D:

\[ \theta_{ML} = \arg \max_{\theta} \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{2\pi\sigma_k}} e\left\{ -\frac{(x_n-\mu_k)^2}{2\sigma_k^2} \right\} \right\} \]
MoG Maximum Likelihood Learning - Problem

• Maximum likelihood criterion, 1-D:

\[ \theta_{ML} = \arg \max_\theta \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{2\pi \sigma_k}} e^{-\frac{(x_n - \mu_k)^2}{2\sigma_k^2}} \right\} \]

• Suppose we set \( \mu_k = x_n \) for some \( k \) and \( n \), then we have one term in the sum:

\[
\pi_k \frac{1}{\sqrt{2\pi \sigma_k}} \exp \left\{ -\frac{(x_n - \mu_k)^2}{2\sigma_k^2} \right\}
= \pi_k \frac{1}{\sqrt{2\pi \sigma_k}} \exp \left\{ -(0)^2 / (2\sigma_k^2) \right\}
\]
MoG Maximum Likelihood Learning - Problem

- Maximum likelihood criterion, 1-D:

\[
\theta_{ML} = \arg \max_{\theta} \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{2\pi\sigma_k}} e^{-\frac{(x_n - \mu_k)^2}{2\sigma_k^2}} \right\}
\]

- Suppose we set \( \mu_k = x_n \) for some \( k \) and \( n \), then we have one term in the sum:

\[
\pi_k \frac{1}{\sqrt{2\pi\sigma_k}} \exp \left\{ -\frac{(x_n - \mu_k)^2}{2\sigma_k^2} \right\} = \pi_k \frac{1}{\sqrt{2\pi\sigma_k}} \exp \left\{ -\frac{0^2}{2\sigma_k^2} \right\}
\]

- In the limit as \( \sigma_k \to 0 \), this goes to \( \infty \)
  - So ML solution is to set some \( \mu_k = x_n \), and \( \sigma_k = 0! \)
ML for Gaussian Mixtures

- Keeping this problem in mind, we will develop an algorithm for ML estimation of the parameters for an MoG model
  - Search for a local optimum
- Consider the log-likelihood function

\[
\ell(\theta) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right\}
\]

- We can try taking derivatives and setting to zero, even though no closed form solution exists
Maximizing Log-Likelihood - Means

\[
\ell(\theta) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}
\]

\[
\frac{\partial}{\partial \mu_k} \ell(\theta) = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} \Sigma_k^{-1}(x_n - \mu_k)
\]

\[
= \sum_{n=1}^{N} \gamma(z_{nk}) \Sigma_k^{-1}(x_n - \mu_k)
\]

• Setting derivative to 0, and multiply by \( \Sigma_k \)

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

\[
\Leftrightarrow \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \quad \text{where} \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk})
\]
Maximizing Log-Likelihood - Means

\[
\ell(\theta) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k) \right\}
\]

\[
\frac{\partial}{\partial \mu_k} \ell(\theta) = \sum_{n=1}^{N} \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n | \mu_j, \Sigma_j)} \Sigma_k^{-1}(x_n - \mu_k)
\]

\[
= \sum_{n=1}^{N} \gamma(z_{nk}) \Sigma_k^{-1}(x_n - \mu_k)
\]

- Setting derivative to 0, and multiply by \( \Sigma_k \)

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

\[
\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \text{ where } N_k = \sum_{n=1}^{N} \gamma(z_{nk})
\]
Maximizing Log-Likelihood - Means and Covariances

• Note that the mean $\mu_k$ is a weighted combination of points $x_n$, using the responsibilities $\gamma(z_{nk})$ for the cluster $k$

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

• $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$ is the effective number of points in the cluster

• A similar result comes from taking derivatives wrt the covariance matrices $\Sigma_k$:

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$
Maximizing Log-Likelihood - Means and Covariances

- Note that the mean $\mu_k$ is a weighted combination of points $x_n$, using the responsibilities $\gamma(z_{nk})$ for the cluster $k$

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

- $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$ is the effective number of points in the cluster

- A similar result comes from taking derivatives wrt the covariance matrices $\Sigma_k$:

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$
Maximizing Log-Likelihood - Mixing Coefficients

• We can also maximize wrt the mixing coefficients $\pi_k$
• Note there is a constraint that $\sum_k \pi_k = 1$
  • Use Lagrange multipliers,
• End up with:

$$\pi_k = \frac{N_k}{N}$$

average responsibility that component $k$ takes
Three Parameter Types and Three Equations

• These three equations a solution does not make

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

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

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

• All depend on \( \gamma(z_{nk}) \), which depends on all 3!
• But an iterative scheme can be used
EM for Gaussian Mixtures

- Initialize parameters, then iterate:
  - **E step**: Calculate responsibilities using current parameters
    \[
    \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}
    \]
  - **M step**: Re-estimate parameters using these \( \gamma(z_{nk}) \)
    \[
    \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n
    \]
    \[
    \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T
    \]
    \[
    \pi_k = \frac{N_k}{N}
    \]

- This algorithm is known as the expectation-maximization algorithm (EM)
  - Next we describe its general form, why it works, and why it’s called EM (but first an example)
EM for Gaussian Mixtures

- Initialize parameters, then iterate:
  - **E step**: Calculate responsibilities using current parameters
    \[
    \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}
    \]
  - **M step**: Re-estimate parameters using these \( \gamma(z_{nk}) \)
    \[
    \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n
    \]
    \[
    \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T
    \]
    \[
    \pi_k = \frac{N_k}{N}
    \]

- This algorithm is known as the expectation-maximization algorithm (EM)
- Next we describe its general form, why it works, and why it’s called EM (but first an example)
EM for Gaussian Mixtures

• Initialize parameters, then iterate:
  • **E step**: Calculate responsibilities using current parameters

\[
\gamma(z_{nk}) = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n | \mu_j, \Sigma_j)}
\]

• **M step**: Re-estimate parameters using these \(\gamma(z_{nk})\)

\[
\begin{align*}
\mu_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \\
\Sigma_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T \\
\pi_k &= \frac{N_k}{N}
\end{align*}
\]

• This algorithm is known as the expectation-maximization algorithm (EM)

  • Next we describe its general form, why it works, and why it’s called EM (but first an example)
• Same initialization as with K-means before
  • Often, K-means is actually used to initialize EM
MoG EM - Example

- Calculate responsibilities $\gamma(\tilde{z}_{nk})$
• Calculate model parameters $\{\pi_k, \mu_k, \Sigma_k\}$ using these responsibilities
MoG EM - Example

- Iteration 2
MoG EM - Example

- Iteration 5
MoG EM - Example

- Iteration 20 - converged
Outline

K-Means

Gaussian Mixture Models

Expectation-Maximization
General Version of EM

• In general, we are interested in maximizing the likelihood

\[ p(X|\theta) = \sum_Z p(X, Z|\theta) \]

where \( X \) denotes all observed variables, and \( Z \) denotes all latent (hidden, unobserved) variables

• Assume that maximizing \( p(X|\theta) \) is difficult (e.g. mixture of Gaussians)

• But maximizing \( p(X, Z|\theta) \) is tractable (everything observed)

  • \( p(X, Z|\theta) \) is referred to as the complete-data likelihood function, which we don’t have
General Version of EM

• In general, we are interested in maximizing the likelihood

\[ p(X|\theta) = \sum_{Z} p(X, Z|\theta) \]

where \( X \) denotes all observed variables, and \( Z \) denotes all latent (hidden, unobserved) variables

• Assume that maximizing \( p(X|\theta) \) is difficult (e.g. mixture of Gaussians)

• But maximizing \( p(X, Z|\theta) \) is tractable (everything observed)
  • \( p(X, Z|\theta) \) is referred to as the complete-data likelihood function, which we don’t have
A Lower Bound

• The strategy for optimization will be to introduce a lower bound on the likelihood
  • This lower bound will be based on the complete-data likelihood, which is easy to optimize
• Iteratively increase this lower bound
• Make sure we’re increasing the likelihood while doing so
A Decomposition Trick

• To obtain the lower bound, we use a decomposition:

\[
\ln p(X, Z | \theta) = \ln p(X | \theta) + \ln p(Z | X, \theta) \quad \text{product rule}
\]

\[
\ln p(X | \theta) = \mathcal{L}(q, \theta) + KL(q || p)
\]

\[
\mathcal{L}(q, \theta) \equiv \sum_Z q(Z) \ln \left\{ \frac{p(X, Z | \theta)}{q(Z)} \right\}
\]

\[
KL(q || p) \equiv -\sum_Z q(Z) \ln \left\{ \frac{p(Z | X, \theta)}{q(Z)} \right\}
\]

• \(KL(q || p)\) is known as the Kullback-Leibler divergence (KL-divergence), and is \(\geq 0\) (see p.55 PRML, next slide)
  • Hence \(\ln p(X | \theta) \geq \mathcal{L}(q, \theta)\)
Kullback-Leibler Divergence

- $KL(p(x)||q(x))$ is a measure of the difference between distributions $p(x)$ and $q(x)$:

$$KL(p(x)||q(x)) = - \sum_x p(x) \log \frac{q(x)}{p(x)}$$

- Motivation: average additional amount of information required to encode $x$ using code assuming distribution $q(x)$ when $x$ actually comes from $p(x)$
  - Note it is not symmetric: $KL(q(x)||p(x)) \neq KL(p(x)||q(x))$ in general
  - It is non-negative:
    - Jensen’s inequality: $- \ln(\sum_x x p(x)) \leq - \sum_x p(x) \ln x$
    - Apply to $KL$:

$$KL(p||q) = - \sum_x p(x) \log \frac{q(x)}{p(x)} \geq - \ln \left( \sum_x \frac{q(x)}{p(x)} p(x) \right) = - \ln \sum_x q(x) = 0$$
Kullback-Leibler Divergence

- $KL(p(x)||q(x))$ is a measure of the difference between distributions $p(x)$ and $q(x)$:

$$KL(p(x)||q(x)) = - \sum_x p(x) \log \frac{q(x)}{p(x)}$$

- Motivation: average additional amount of information required to encode $x$ using code assuming distribution $q(x)$ when $x$ actually comes from $p(x)$
- Note it is not symmetric: $KL(q(x)||p(x)) \neq KL(p(x)||q(x))$ in general
- It is non-negative:
  - Jensen’s inequality: $- \ln(\sum x x p(x)) \leq - \sum x p(x) \ln x$
  - Apply to $KL$:

$$KL(p||q) = - \sum_x p(x) \log \frac{q(x)}{p(x)} \geq - \ln \left( \sum_x \frac{q(x)}{p(x)} p(x) \right) = - \ln \sum_x q(x) = 0$$
Kullback-Leibler Divergence

- $KL(p(x)||q(x))$ is a measure of the difference between distributions $p(x)$ and $q(x)$:

$$KL(p(x)||q(x)) = - \sum_x p(x) \log \frac{q(x)}{p(x)}$$

- Motivation: average additional amount of information required to encode $x$ using code assuming distribution $q(x)$ when $x$ actually comes from $p(x)$
- Note it is not symmetric: $KL(q(x)||p(x)) \neq KL(p(x)||q(x))$ in general
- It is non-negative:
  - Jensen’s inequality: $- \ln(\sum_x x p(x)) \leq - \sum_x p(x) \ln x$
  - Apply to $KL$:

$$KL(p||q) = - \sum_x p(x) \log \frac{q(x)}{p(x)} \geq - \ln \left( \sum_x \frac{q(x)}{p(x)} p(x) \right) = - \ln \sum_x q(x) = 0$$
Kullback-Leibler Divergence

- $KL(p(x)||q(x))$ is a measure of the difference between distributions $p(x)$ and $q(x)$:

$$KL(p(x)||q(x)) = - \sum_x p(x) \log \frac{q(x)}{p(x)}$$

- Motivation: average additional amount of information required to encode $x$ using code assuming distribution $q(x)$ when $x$ actually comes from $p(x)$

- Note it is not symmetric: $KL(q(x)||p(x)) \neq KL(p(x)||q(x))$ in general

- It is non-negative:
  - Jensen’s inequality: $- \ln(\sum_x xp(x)) \leq - \sum_x p(x) \ln x$
  - Apply to $KL$:

$$KL(p||q) = - \sum_x p(x) \log \frac{q(x)}{p(x)} \geq - \ln \left( \sum_x \frac{q(x)}{p(x)} p(x) \right) = - \ln \sum_x q(x) = 0$$

©Möller/Mori
Increasing the Lower Bound - E step

- EM is an iterative optimization technique which tries to maximize this lower bound: \( \ln p(X|\theta) \geq \mathcal{L}(q, \theta) \)

- **E step**: Fix \( \theta^{old} \), maximize \( \mathcal{L}(q, \theta^{old}) \) wrt \( q \)
  - i.e. Choose distribution \( q \) to maximize \( \mathcal{L} \)
  - Reordering bound:
  \[
  \mathcal{L}(q, \theta^{old}) = \ln p(X|\theta^{old}) - KL(q||p)
  \]

- \( \ln p(X|\theta^{old}) \) does not depend on \( q \)
- Maximum is obtained when \( KL(q||p) \) is as small as possible
  - Occurs when \( q = p \), i.e. \( q(Z) = p(Z|X, \theta) \)
  - This is the posterior over \( Z \), recall these are the responsibilities from MoG model
Increasing the Lower Bound - E step

- EM is an iterative optimization technique which tries to maximize this lower bound: $\ln p(X|\theta) \geq \mathcal{L}(q, \theta)$

- **E step**: Fix $\theta^{old}$, maximize $\mathcal{L}(q, \theta^{old})$ wrt q
  - i.e. Choose distribution $q$ to maximize $\mathcal{L}$
  - Reordering bound:
    \[
    \mathcal{L}(q, \theta^{old}) = \ln p(X|\theta^{old}) - KL(q||p)
    \]

- $\ln p(X|\theta^{old})$ does not depend on $q$
- Maximum is obtained when $KL(q||p)$ is as small as possible
  - Occurs when $q = p$, i.e. $q(Z) = p(Z|X, \theta)$
  - This is the posterior over $Z$, recall these are the responsibilities from MoG model
Increasing the Lower Bound - E step

• EM is an iterative optimization technique which tries to maximize this lower bound: \( \ln p(X|\theta) \geq \mathcal{L}(q, \theta) \)

• **E step**: Fix \( \theta^{old} \), maximize \( \mathcal{L}(q, \theta^{old}) \) wrt \( q \)
  
  • i.e. Choose distribution \( q \) to maximize \( \mathcal{L} \)
  
  • Reordering bound:

\[
\mathcal{L}(q, \theta^{old}) = \ln p(X|\theta^{old}) - KL(q||p)
\]

• \( \ln p(X|\theta^{old}) \) does not depend on \( q \)

• Maximum is obtained when \( KL(q||p) \) is as small as possible
  
  • Occurs when \( q = p \), i.e. \( q(Z) = p(Z|X, \theta) \)
  
  • This is the posterior over \( Z \), recall these are the responsibilities from MoG model
Increasing the Lower Bound - E step

- EM is an iterative optimization technique which tries to maximize this lower bound: $\ln p(X|\theta) \geq \mathcal{L}(q, \theta)$

- **E step**: Fix $\theta^{old}$, maximize $\mathcal{L}(q, \theta^{old})$ wrt $q$
  - i.e. Choose distribution $q$ to maximize $\mathcal{L}$
  - Reordering bound:

  $$\mathcal{L}(q, \theta^{old}) = \ln p(X|\theta^{old}) - KL(q||p)$$

- $\ln p(X|\theta^{old})$ does not depend on $q$
- Maximum is obtained when $KL(q||p)$ is as small as possible
  - Occurs when $q = p$, i.e. $q(Z) = p(Z|X, \theta)$
  - This is the posterior over $Z$, recall these are the responsibilities from MoG model
Increasing the Lower Bound - M step

- **M step**: Fix $q$, maximize $\mathcal{L}(q, \theta)$ wrt $\theta$

- The maximization problem is on

\[
\mathcal{L}(q, \theta) = \sum_Z q(Z) \ln p(X, Z|\theta) - \sum_Z q(Z) \ln q(Z)
\]

\[
= \sum_Z p(Z|X, \theta^{old}) \ln p(X, Z|\theta) - \sum_Z p(Z|X, \theta^{old}) \ln p(Z|X, \theta^{old})
\]

- Second term is constant with respect to $\theta$
- First term is $\ln$ of complete data likelihood, which is assumed easy to optimize
  - *Expected complete log likelihood* – what we think complete data likelihood will be
Increasing the Lower Bound - M step

- **M step**: Fix $q$, maximize $\mathcal{L}(q, \theta)$ wrt $\theta$
- The maximization problem is on

$$
\mathcal{L}(q, \theta) = \sum_{Z} q(Z) \ln p(X, Z | \theta) - \sum_{Z} q(Z) \ln q(Z)
$$

$$
= \sum_{Z} p(Z | X, \theta^{old}) \ln p(X, Z | \theta) - \sum_{Z} p(Z | X, \theta^{old}) \ln p(Z | X, \theta^{old})
$$

- Second term is constant with respect to $\theta$
- First term is ln of complete data likelihood, which is assumed easy to optimize
  - Expected complete log likelihood – what we think complete data likelihood will be
Increasing the Lower Bound - M step

- **M step**: Fix $q$, maximize $\mathcal{L}(q, \theta)$ wrt $\theta$
- The maximization problem is on

$$
\mathcal{L}(q, \theta) = \sum_Z q(Z) \ln p(X, Z | \theta) - \sum_Z q(Z) \ln q(Z)
$$

$$
= \sum_Z p(Z | X, \theta^{old}) \ln p(X, Z | \theta) - \sum_Z p(Z | X, \theta^{old}) \ln p(Z | X, \theta^{old})
$$

- Second term is constant with respect to $\theta$
- First term is $\ln$ of complete data likelihood, which is assumed easy to optimize
  - Expected complete log likelihood – what we think complete data likelihood will be
Increasing the Lower Bound - M step

- **M step**: Fix $q$, maximize $\mathcal{L}(q, \theta)$ wrt $\theta$
- The maximization problem is on

$$
\mathcal{L}(q, \theta) = \sum_Z q(Z) \ln p(X, Z | \theta) - \sum_Z q(Z) \ln q(Z)
$$

$$
= \sum_Z p(Z | X, \theta^{old}) \ln p(X, Z | \theta) - \sum_Z p(Z | X, \theta^{old}) \ln p(Z | X, \theta^{old})
$$

- Second term is constant with respect to $\theta$
- First term is $\ln$ of complete data likelihood, which is assumed easy to optimize
  - **Expected complete log likelihood** – what we think complete data likelihood will be
Why does EM work?

• In the M-step we changed from $\theta^{old}$ to $\theta^{new}$
• This increased the lower bound $\mathcal{L}$, unless we were at a maximum (so we would have stopped)
• This must have caused the log likelihood to increase
  • The E-step set $q$ to make the KL-divergence 0:
    $$\ln p(X|\theta^{old}) = \mathcal{L}(q, \theta^{old}) + KL(q||p) = \mathcal{L}(q, \theta^{old})$$
  • Since the lower bound $\mathcal{L}$ increased when we moved from $\theta^{old}$ to $\theta^{new}$:
    $$\ln p(X|\theta^{old}) = \mathcal{L}(q, \theta^{old}) < \mathcal{L}(q, \theta^{new})$$
    $$= \ln p(X|\theta^{new}) - KL(q||p^{new})$$
• So the log-likelihood has increased going from $\theta^{old}$ to $\theta^{new}$
Why does EM work?

- In the M-step we changed from \( \theta^{old} \) to \( \theta^{new} \)
- This increased the lower bound \( \mathcal{L} \), unless we were at a maximum (so we would have stopped)
- This must have caused the log likelihood to increase
  - The E-step set \( q \) to make the KL-divergence 0:
    \[
    \ln p(X|\theta^{old}) = \mathcal{L}(q,\theta^{old}) + KL(q||p) = \mathcal{L}(q,\theta^{old})
    \]
  - Since the lower bound \( \mathcal{L} \) increased when we moved from \( \theta^{old} \) to \( \theta^{new} \):
    \[
    \ln p(X|\theta^{old}) = \mathcal{L}(q,\theta^{old}) < \mathcal{L}(q,\theta^{new})
    \]
    \[
    = \ln p(X|\theta^{new}) - KL(q||p^{new})
    \]
  - So the log-likelihood has increased going from \( \theta^{old} \) to \( \theta^{new} \)
Why does EM work?

- In the M-step we changed from $\theta^{old}$ to $\theta^{new}$
- This increased the lower bound $\mathcal{L}$, unless we were at a maximum (so we would have stopped)
- This must have caused the log likelihood to increase
  - The E-step set $q$ to make the KL-divergence 0:
    \[
    \ln p(X|\theta^{old}) = \mathcal{L}(q, \theta^{old}) + KL(q||p) = \mathcal{L}(q, \theta^{old})
    \]
    
    - Since the lower bound $\mathcal{L}$ increased when we moved from $\theta^{old}$ to $\theta^{new}$:
      \[\ln p(X|\theta^{old}) = \mathcal{L}(q, \theta^{old}) < \mathcal{L}(q, \theta^{new})\]
      \[= \ln p(X|\theta^{new}) - KL(q||p^{new})\]
      
    - So the log-likelihood has increased going from $\theta^{old}$ to $\theta^{new}$
Why does EM work?

• In the M-step we changed from $\theta^{\text{old}}$ to $\theta^{\text{new}}$

• This increased the lower bound $\mathcal{L}$, unless we were at a maximum (so we would have stopped)

• This must have caused the log likelihood to increase
  • The E-step set $q$ to make the KL-divergence 0:
    \[
    \ln p(X|\theta^{\text{old}}) = \mathcal{L}(q, \theta^{\text{old}}) + KL(q||p) = \mathcal{L}(q, \theta^{\text{old}})
    \]

  • Since the lower bound $\mathcal{L}$ increased when we moved from $\theta^{\text{old}}$ to $\theta^{\text{new}}$:
    \[
    \ln p(X|\theta^{\text{old}}) = \mathcal{L}(q, \theta^{\text{old}}) < \mathcal{L}(q, \theta^{\text{new}})
    \]
    \[
    = \ln p(X|\theta^{\text{new}}) - KL(q||p^{\text{new}})
    \]

  • So the log-likelihood has increased going from $\theta^{\text{old}}$ to $\theta^{\text{new}}$
Bounding Example

Consider 2 component 1-D MoG with known variances (example from F. Dellaert) [http://www.cc.gatech.edu/~dellaert/em-paper.pdf]
• True likelihood function
  • Recall we’re fitting means $\theta_1, \theta_2$
Bounding Example

- Lower bound the likelihood function using averaging distribution $q(Z)$
  - $\ln p(X|\theta) = \mathcal{L}(q, \theta) + KL(q(Z)||p(Z|X, \theta))$
  - Since $q(Z) = p(Z|X, \theta^{old})$, bound is tight (equal to actual likelihood) at $\theta = \theta^{old}$
Bounding Example

- Lower bound the likelihood function using averaging distribution \( q(Z) \)
  - \( \ln p(X|\theta) = \mathcal{L}(q, \theta) + KL(q(Z)||p(Z|X, \theta)) \)
  - Since \( q(Z) = p(Z|X, \theta^{old}) \), bound is tight (equal to actual likelihood) at \( \theta = \theta^{old} \)

©Möller/Mori
Bounding Example

- Lower bound the likelihood function using averaging distribution $q(Z)$
  - $\ln p(X|\theta) = \mathcal{L}(q, \theta) + KL(q(Z)||p(Z|X, \theta))$
  - Since $q(Z) = p(Z|X, \theta^{old})$, bound is tight (equal to actual likelihood) at $\theta = \theta^{old}$
Bounding Example

- Lower bound the likelihood function using averaging distribution $q(Z)$
  - $\ln p(X|\theta) = \mathcal{L}(q, \theta) + KL(q(Z) || p(Z|X, \theta))$
  - Since $q(Z) = p(Z|X, \theta^{old})$, bound is tight (equal to actual likelihood) at $\theta = \theta^{old}$
EM - Summary

• EM finds local maximum to likelihood

\[ p(X|\theta) = \sum_z p(X, Z|\theta) \]

• Iterates two steps:
  • **E step** “fills in” the missing variables \( Z \) (calculates their distribution)
  • **M step** maximizes expected complete log likelihood (expectation wrt **E step** distribution)

• This works because these two steps are performing a coordinate-wise hill-climbing on a lower bound on the likelihood \( p(X|\theta) \)
Conclusion

- Readings: Ch. 9.1, 9.2, 9.4
- K-means clustering
- Gaussian mixture model
- What about K?
  - Model selection: either cross-validation or Bayesian version (average over all values for K)
- Expectation-maximization, a general method for learning parameters of models when not all variables are observed