Expectation-Maximization

Machine Learning 1

Machine Learning Group, TU Berlin – WS 2016/17

ISIS survey

Outline

K-Means clustering

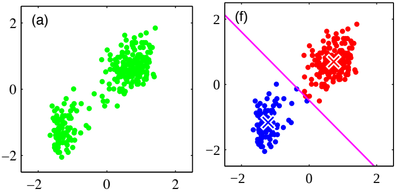
Gaussian Mixture Model (GMM)

Expectation Maximization

- Maximum likelihood for latent variables
- A lower bound on the log-likelihood

Clustering

Given N d-dimensional datapoints $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ (no labels),



partition data into K disjoint sets $\mathcal{S}_k$ based on similarity.

$$\forall k \forall l \; \mathcal{S}_k \cap \mathcal{S}_l = \emptyset, \quad \bigcup_{k=1}^{K} \mathcal{S}_k = \mathcal{D}$$

K-Means

Define clusters by minimum Euclidean distance to cluster mean.

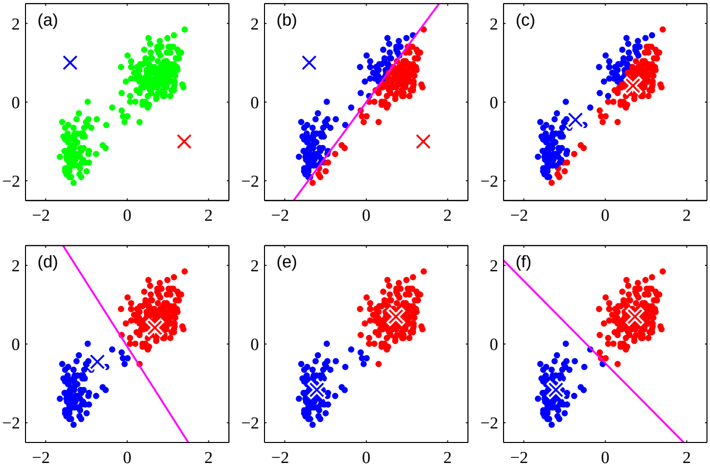
$\rightarrow$ Find $\boldsymbol{\theta} = \{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ that minimize

$$J(\boldsymbol{\theta}) = \textstyle\sum_{k=1}^{K} \sum_{\mathbf{x}_n \in \mathcal{S}_k} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2, \text{ where } \boldsymbol{\mu}_k = \frac{1}{|\mathcal{S}_k|} \sum_{\mathbf{x}_n \in \mathcal{S}_k} \mathbf{x}_n$$

Algorithm ("Expectation Maximization")

1. Choose K random points as initial cluster centers $\boldsymbol{\mu}_1^{(0)}, \ldots, \boldsymbol{\mu}_K^{(0)}$
2. **Assignment (E):** $\mathcal{S}_k^{(t)} = \left\{\mathbf{x}_n : \left\|\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)}\right\|^2 \leq \left\|\mathbf{x}_n - \boldsymbol{\mu}_l^{(t)}\right\|^2 \ \forall l, 1 \leq l \leq K\right\}$
3. **Update (M):** $\boldsymbol{\mu}_k^{(t+1)} = \frac{1}{|\mathcal{S}_k^{(t)}|} \sum_{\mathbf{x}_n \in \mathcal{S}_k^{(t)}} \mathbf{x}_n$
4. Iterate 2. and 3. until convergence to local minimum

K-Means

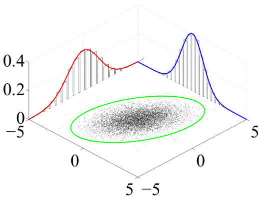


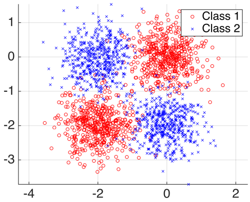
Figures from Bishop 2006

Density estimation

Multivariate Gaussian:

$$p(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right]$$
$$\sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$





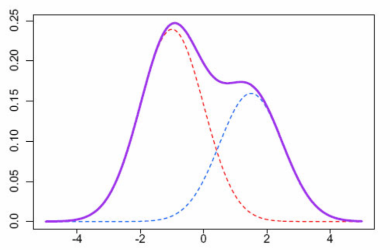
Does not always model data (e.g. class-conditional densities) well.

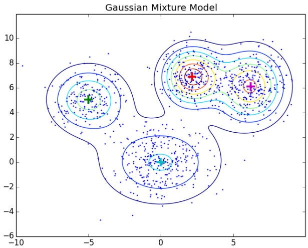
Gaussian Mixture Model (GMM)

$$
\begin{aligned}
p(\mathbf{x}|\boldsymbol{\theta}) &= \sum_{k=1}^{K} \tau_k p_k(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \\
p_k(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) &\sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k) \\
\tau_k &: \text{scaling or ``prior'' of } p(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \\
\sum_{k=1}^{K} \tau_k &= 1 \\
\boldsymbol{\theta} &= \{\tau_1, \ldots, \tau_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \Sigma_1, \ldots, \Sigma_K\}
\end{aligned}
$$

Gaussian Mixture Model (GMM)

GMM's are universal density approximators.





Figures from Eugene Weinstein, Yu Zhu

As a byproduct, they provide a clustering solution.

Fitting a GMM using ML

Log-likelihood:

$$
\begin{aligned}
L(\boldsymbol{\theta}) &= \log\left[p(X|\boldsymbol{\theta})\right] \\
&= \log\left[\prod_{n=1}^{N}\sum_{k=1}^{K}\tau_k p_k(\mathbf{x}_n|\boldsymbol{\mu}_k,\Sigma_k)\right] \\
&= \sum_{n=1}^{N}\log\left[\sum_{k=1}^{K}\tau_k p_k(\mathbf{x}_n|\boldsymbol{\mu}_k,\Sigma_k)\right] \\
&= \sum_{n=1}^{N}\log\left[\sum_{k=1}^{K}\tau_k\left(\frac{2\pi)^{-d/2}}{(|\Sigma_k|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu}_k)^\top\Sigma_k^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_k)\right]\right)\right]
\end{aligned}
$$

Difficult to optimize: $\frac{\partial L(\boldsymbol{\theta})}{\partial\theta_k} = \sum_n \frac{1}{\sum_k f(\boldsymbol{\theta}_k)}\frac{\partial f(\boldsymbol{\theta}_k)}{\theta_k}$

No analytic solution.

Fitting a GMM using EM

Trick: introduce auxiliary variables indicating the membership of each sample to a Gaussian

$$
\begin{aligned}
\mathbf{z}_1, \ldots, \mathbf{z}_N \in \mathbb{R}^K &\sim \mathit{Categorical}\,(\boldsymbol{\tau}) \\
p(z_{nk} = 1) &= \tau_k \\
\forall n\ \exists! k \quad & z_{nk} = 1,\ z_{nj, j \neq k} = 0 \\
\text{e.g. } \mathbf{z}_n &= (0, 0, 0, 1, 0, \ldots, 0)^\top
\end{aligned}
$$

Note: this is also how to sample from a GMM

1. Sample $\mathbf{z} \sim \mathit{Categorical}\,(\boldsymbol{\tau})$
2. Sample $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k)$, where $z_k = 1$

Fitting a GMM using EM: algorithm

1. Initialize t=0, $\boldsymbol{\theta}^{(0)} = \{\tau_1, \ldots, \tau_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \Sigma_1, \ldots, \Sigma_K\}$ (e.g., $\tau_k^{(0)} = 1/K, \Sigma_k^{(0)} = I, \boldsymbol{\mu}_k^{(0)} = \text{rand}$)
2. **Expectation:** compute membership probabilities given $\boldsymbol{\theta}^{(t)}$

$$q^{(t)}(z_{nk}) := p(z_{nk} = 1 | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \overset{\text{Bayes}}{=} \frac{p(\boldsymbol{x}_n | z_{nk}, \boldsymbol{\theta}^{(t)}) p(z_{nk}, \boldsymbol{\theta}^{(t)})}{p(\boldsymbol{x}_n | \boldsymbol{\theta}^{(t)})}$$

$$= \frac{\tau_k^{(t)} p_k(\mathbf{x}_n | \boldsymbol{\mu}_k^{(t)}, \Sigma_k^{(t)})}{\sum_{l=1}^K \tau_l^{(t)} p_l(\mathbf{x}_n | \boldsymbol{\mu}_l^{(t)}, \Sigma_l^{(t)})}$$

3. **Maximization:** update $\boldsymbol{\theta}$ given (soft) cluster assignments

$$\tau_k^{(t+1)} = \frac{1}{N} \sum_{n=1}^N q^{(t)}(z_{nk}) \qquad \boldsymbol{\mu}_k^{(t+1)} = 1/N\tau_k^{(t+1)} \sum_{n=1}^N q^{(t)}(z_{nk}) \boldsymbol{x}_n$$

$$\Sigma_k^{(t+1)} = 1/N\tau_k^{(t+1)} \sum_{n=1}^N q^{(t)}(z_{nk}) \left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t+1)}\right) \left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t+1)}\right)^\top$$

Fitting a GMM using EM: algorithm

Note: also possible to use hard cluster assignments.

1. **Expectation:**

$$q^{(t)}(z_{nk}) = p(z_{nk} = 1 | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)})$$

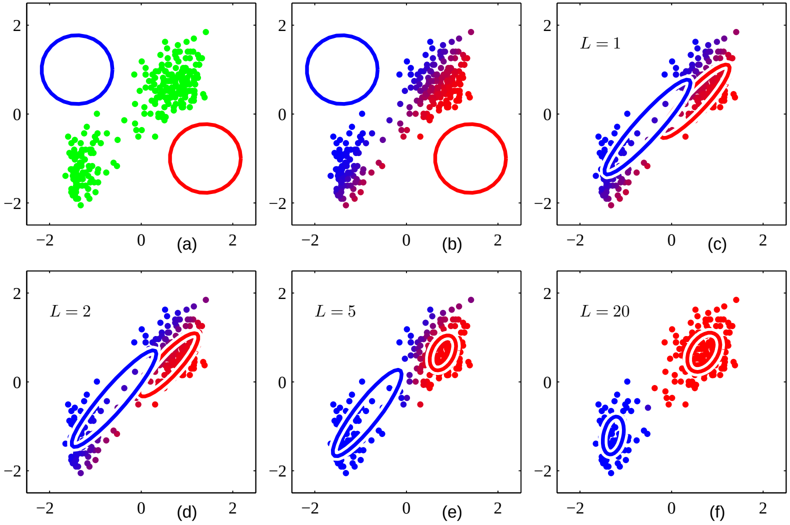
$$z_{nk}^{(t)} = \begin{cases} 1 & \text{if } q^{(t)}(z_{nk}) = \max_l q^{(t)}(z_{nl}) \\ 0 & \text{otherwise} \end{cases}$$

2. **Maximization:** update $\boldsymbol{\theta}$ given hard cluster assignments

$$\tau_k^{(t+1)} = \frac{1}{N} \sum_{n=1}^{N} z_{nk}^{(t)} \qquad \boldsymbol{\mu}_k^{(t+1)} = {}^{1}\!/_{N\tau_k^{(t+1)}} \sum_{n=1}^{N} z_{nk}^{(t)} \boldsymbol{x}_n$$

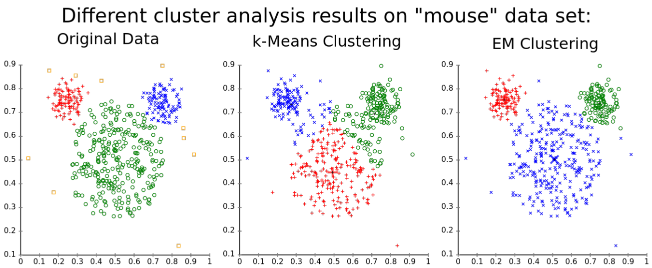
$$\Sigma_k^{(t+1)} = {}^{1}\!/_{N\tau_k^{(t+1)}} \sum_{n=1}^{N} z_{nk}^{(t)} \left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t+1)}\right) \left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t+1)}\right)^\top$$

Fitting a GMM using EM



Figures from Bishop 2006

Fitting a GMM: comparison



Wikipedia

In contrast to K-means, GMM allows for

- Unequal cluster variances
- Unequal cluster probabilities
- Non-spherical clusters
- Soft cluster assignment

Fitting a GMM

Maximum Likelihood: our ultimate goal is to optimize $p(X|\boldsymbol{\theta})$.

Do the update equations optimize $p(X|\boldsymbol{\theta})$?

To answer this, it is easier to look at the EM algorithm in general.

ML for latent variable models

z latent (unobserved variables)

X observed data

θ model parameters

We want to maximize the likelihood of the observed data (= incomplete-data likelihood), $L(\theta|X) = p(X|\theta)$:

$$\hat{\theta} = \arg\max_{\theta} \log\left[p(X|\theta)\right] = \arg\max_{\theta} \log\left[\sum_{z\in\mathcal{Z}} p(X,z|\theta)\right].$$

Maximizing this directly is difficult because of $\log \sum \ldots$.

On the other hand, it is often easy to optimize the complete-data likelihood, $L(\theta|X,z) = \log p(X,z|\theta)$.

Example: GMM

Incomplete-data log-likelihood

$$\log\left[p(X|\theta)\right] = \sum_{n=1}^{N} \log\left[\sum_{k=1}^{K} \tau_k p_k(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)\right]$$

Complete-data log-likelihood

$$\log\left[p(X, z|\theta)\right] = \sum_{n=1}^{N} \log\left[\sum_{k=1}^{K} \delta_{z_{nk}=1} \tau_k p_k(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)\right]$$

$\rightarrow$ Analytic ML estimate for each $\theta_k = (\tau_k, \boldsymbol{\mu}_k, \Sigma_k)$

Problem: We don't know z.

Expectation Maximization

Since we don't know z, we need to estimate it jointly with θ.

Expectation Maximization algorithm:

- Iterate between updates of hidden variables and parameters
- **Theory:** updates are defined in a way such that $p(X|\theta)$ increases in each step

$\rightarrow$ Guaranteed to find local maximum of $p(X|\theta)$
(hard to find global maximum if $p(X|\theta)$ is non-concave)

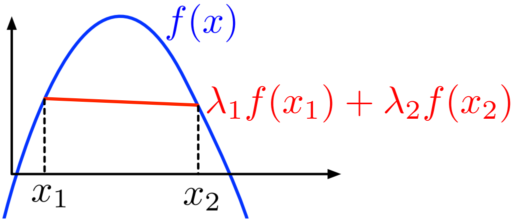
- **Technically:** optimize a lower bound on $p(X|\theta)$ and subsequently improve bound

Jensen's inequality

For any convex combination $\lambda_1, \ldots, \lambda_I, \quad \lambda_i \geq 0, \quad \sum_{i=1}^{I} \lambda_i = 1$

and any concave function f:

$$f(\sum_{i=1}^{I} \lambda_i x_i) \geq \sum_{i=1}^{I} \lambda_i f(x_i) \; .$$



(conversely for convex f, analogous for continuous f)

A lower bound on the log-likelihood

θ: a parameter setting

$q(z)$: a probability mass function of choice on z

$$
\begin{aligned}
\log p(X|\theta) &= \log \sum_z p(X, z|\theta) \\
&= \log \sum_z q(z) \left[\frac{p(X, z|\theta)}{q(z)}\right] \\
\text{Jensen's inequality} &\downarrow \quad \text{(remember that } \sum_z q(z) = 1\text{, log concave)} \\
&\geq \sum_z \underbrace{q(z)}_{\lambda_i} \underbrace{\log}_{f(.)} \underbrace{\left[\frac{p(X, z|\theta)}{q(z)}\right]}_{x_i} \\
&=: F(q(z), \theta)
\end{aligned}
$$

This lower bound is much easier to optimize ($\log \sum$ vs. $\sum \log$).

Expectation Maximization

True objective: maximize the data log likelihood w.r.t. θ

$$\hat{\theta} = \arg\max_{\theta} \log p(X|\theta) \ .$$

This is difficult.

EM objective: maximize, w.r.t. q and θ, the lower bound

$$\hat{q}, \hat{\theta} = \arg\max_{q,\theta} F(q(z), \theta) \ .$$

Maximization of the lower bound

$$F(q(z), \theta) = \sum_z q(z) \log \left[\frac{p(X, z|\theta)}{q(z)} \right]$$

There are two ways to improve the lower bound:

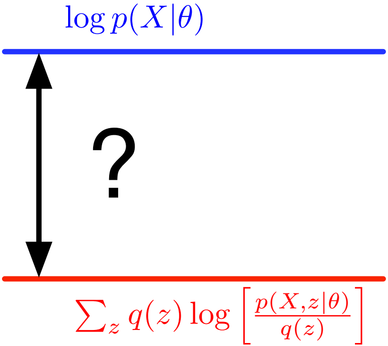
1. **Expectation: improve** $q(z)$ for given θ
2. Maximization: improve θ for given $q(z)$

Expectation: improving $q(z)$

How to select $q(z)$?

The difference between the data log-likelihood and lower bound:

$$\log p(X|\theta) - \sum_z q(z) \log \left[\frac{p(X, z|\theta)}{q(z)} \right]$$



Expectation: improving $q(z)$

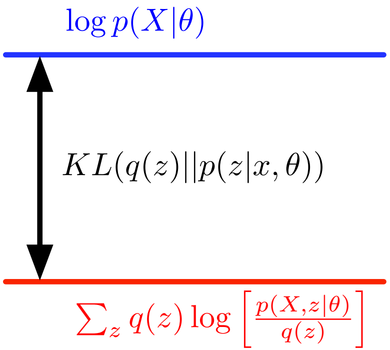
The difference between the data log-likelihood and lower bound:

$$\begin{aligned}
& \log p(X|\theta) - \sum_z q(z) \log\left[\frac{p(X,z|\theta)}{q(z)}\right] \\
= \;& \log p(X|\theta) - \sum_z q(z) \log\left[\frac{p(X|\theta)p(z|X,\theta)}{q(z)}\right] \\
= \;& \underbrace{\log p(X|\theta) - \sum_z q(z) \log p(X|\theta)}_{0} - \sum_z q(z) \log\left[\frac{p(z|X,\theta)}{q(z)}\right] \\
= \;& -\sum_z q(z) \log\left[\frac{p(z|X,\theta)}{q(z)}\right] = \sum_z q(z) \log\left[\frac{q(z)}{p(z|X,\theta)}\right] \\
= \;& KL(q(z)||p(z|x,\theta))
\end{aligned}$$

Expectation: improving $q(z)$

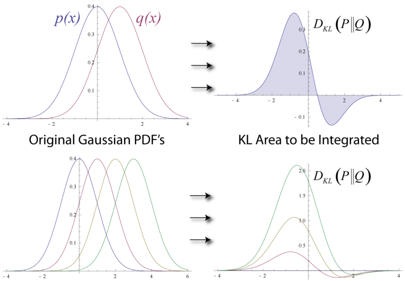
The difference between the data log likelihood and lower bound:

$$\log p(X|\theta) - \sum_z q(z) \log \left[\frac{p(X, z|\theta)}{q(z)} \right] = KL(q(z)||p(z|x,\theta))$$



The Kullback-Leibler divergence

$$KL(P||Q) = \sum_x P(x) \log \frac{Q(x)}{P(x)} = \mathbb{E}_x \left[\log \frac{Q(x)}{P(x)} \right] \geq 0$$



Wikipedia

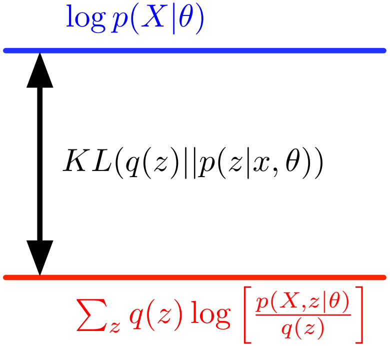
- Measures the distance between two distributions P and Q
- Not a true metric (not symmetric, no triangle inequality)
- Important quantity in information theory

Expectation: improving $q(z)$

$$KL(q(z)||p(z|x,\theta)) = 0 \iff q(z) = p(z|X,\theta)$$

$\Rightarrow$ Lower bound is strict if $q(z) = p(z|X,\theta)$.

Expectation step: set $q^{(t)}(z) = p(z|X,\theta^{(t)})$.



Maximization of the lower bound

$$F(q(z), \theta) = \sum_z q(z) \log \left[\frac{p(X, z|\theta)}{q(z)} \right]$$

There are two ways to improve the lower bound:

1. Expectation: improve $q(z)$ for given θ
2. **Maximization: improve** θ for given $q(z)$

Maximization: improving θ

Goal: maximize $F(q(z), \theta)$ w.r.t. θ.

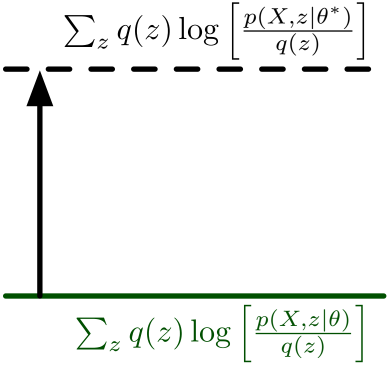
$$\begin{aligned}
\theta^* &= \arg\max_\theta \sum_z q(z) \log \left[\frac{p(X, z|\theta)}{q(z)} \right] \\
&= \arg\max_\theta \sum_z q(z) \log p(X, z|\theta) \underbrace{- \sum_z q(z) \log q(z)}_{\text{“Entropy” } H(z)} \\
&= \arg\max_\theta \sum_z q(z) \log p(X, z|\theta)
\end{aligned}$$

Approach: set gradient to zero ...

Typically, easy (analytic) solution, due to $\sum \log$ rather than $\log \sum$.

Maximization: improving θ

Maximization step: set $\theta^{(t+1)} = \arg\max_\theta F(q(z)^{(t)}, \theta)$.



Iterative optimization

$$F(q(z), \theta) = \sum_z q(z) \log \left[\frac{p(X, z|\theta)}{q(z)} \right]$$

$$\begin{aligned} \log p(X|\theta^{(t)}) &\stackrel{E-Step}{=} F(q^{(t+1)}(z), \theta^{(t)}) \\ &\stackrel{M-Step}{\leq} F(q^{(t+1)}(z), \theta^{(t+1)}) \\ &\stackrel{Jensen}{\leq} \log p(X|\theta^{(t+1)}) \end{aligned}$$

$\rightarrow$ Convergence to local maximum of $L(\theta|X) = \log p(X|\theta)$.

Note: update of $q(z) = p(z|X, \theta) \rightarrow$ update of z.

Iterative optimization

$$\sum_z q_{n+1}(z) \log \left[\frac{p(X,z|\theta_{n+1})}{q_{n+1}(z)} \right]$$

Expectation

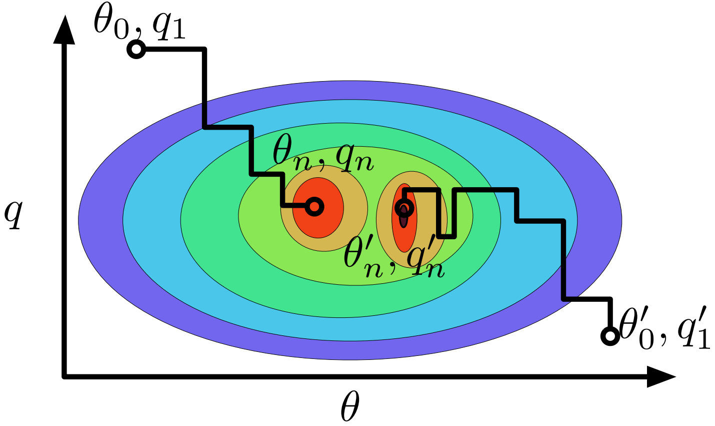
Maximisation

$$\log p(X|\theta_n) = \sum_z q_{n+1}(z) \log \left[\frac{p(X,z|\theta_n)}{q_{n+1}(z)} \right]$$

Expectation

$$\sum_z q_n(z) \log \left[\frac{p(X,z|\theta_n)}{q_n(z)} \right]$$

Convergence to local maximum



"Block coordinate ascent"

EM summary

z latent (unobserved variables)

X observed data

θ model parameters

1. Initialize $\theta^{(0)} = \text{rand}$
2. Expectation: $q^{(t)}(z) = p(z|X, \theta^{(t)})$
3. Maximization: $\theta^{(t+1)} = \arg\max_\theta \sum_z q^{(t)}(z) \log p(X, z|\theta)$
4. Iterate until convergence

Why "Expectation"?

Remember: maximization step

$$\begin{aligned}
\theta^{(t+1)} &= \arg\max_\theta F(q^{(t)}(z), \theta) \\
&= \arg\max_\theta \sum_z q^{(t)}(z) \log p(X, z|\theta) + H(z) \\
&= \arg\max_\theta \sum_z p(z|X, \theta^{(t)}) \log p(X, z|\theta) \\
&= \arg\max_\theta \mathbb{E}_{z|X,\theta^{(t)}} \left[\log p(X, z|\theta)\right] \\
&=: \arg\max_\theta Q(\theta|\theta^{(t)})
\end{aligned}$$

Original "expectation" step (Dempster et al., 1977): compute

$$Q(\theta|\theta^{(t)}) = \mathbb{E}_{z|X,\theta^{(t)}} \left[\log p(X, z|\theta)\right]$$

$\rightarrow$ Boils down to estimating $q^{(t)}(z) = p(z|X, \theta^{(t)})$.

Fitting a GMM using EM: Expectation

Expectation step:

$$q^{(t)}(z_{nk}) := p(z_{nk} = 1 | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) = \frac{\tau_k^{(t)} p_k(\mathbf{x}_n | \boldsymbol{\mu}_k^{(t)}, \Sigma_k^{(t)})}{\sum_{l=1}^{K} \tau_l^{(t)} p_l(\mathbf{x}_n | \boldsymbol{\mu}_l^{(t)}, \Sigma_l^{(t)})}$$

Fitting a GMM using EM: Expectation

$$
\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \sum_{z} q^{(t)}(z) \log p(X, z|\theta) \\
&= \mathsf{E}_{Z|X,\boldsymbol{\theta}^{(t)}} \left[\log p(X, z|\theta)\right] \\
&= \mathsf{E}_{Z|X,\boldsymbol{\theta}^{(t)}} \log \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}) \\
&= \mathsf{E}_{Z|X,\boldsymbol{\theta}^{(t)}} \sum_{n=1}^{N} \log \left[p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta})\right] \\
&= \sum_{n=1}^{N} \mathsf{E}_{Z|X,\boldsymbol{\theta}^{(t)}} \left[\log p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta})\right] \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \log \tau_k p_k(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} q^{(t)}(z_{nk}) \Big[\log \tau_k - \tfrac{1}{2} \log |\Sigma_k| - \tfrac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_k) - \tfrac{d}{2} \log(2\pi) \Big]
\end{aligned}
$$

Fitting a GMM using EM: Maximization

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$
$$= \sum_{n=1}^{N}\sum_{k=1}^{K} q^{(t)}(z_{nk})\big[\log \tau_k - \tfrac{1}{2}\log|\Sigma_k| - \tfrac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_k) - \tfrac{d}{2}\log(2\pi)\big]$$

$$\begin{aligned}
\boldsymbol{\tau}^{(t+1)} &= \arg\max_{\boldsymbol{\tau}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \quad \text{s.t.} \quad \sum_{k=1}^{K}\tau_k = 1 \\
&= \arg\max_{\boldsymbol{\tau}} \sum_{k=1}^{K} \log \tau_k \sum_{n=1}^{N} q^{(t)}(z_{nk}) + \lambda\left(1 - \sum_{k=1}^{K}\tau_k\right) \\
\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})}{\partial \tau_k^{(t+1)}} &= \frac{1}{\tau_k^{(t+1)}} \sum_{n=1}^{N} q^{(t)}(z_{nk}) - \lambda = 0 \\
\sum_{k=1}^{K}\tau_k^{(t+1)} = 1 \quad &\Rightarrow \lambda = N \quad \Rightarrow \tau_k^{(t+1)} = \frac{1}{N}\sum_{n=1}^{N} q^{(t)}\left(z_{nk}\right)
\end{aligned}$$

Fitting a GMM using EM: Maximization

$$
\begin{aligned}
&Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} q^{(t)}(z_{nk})\Big[\log \tau_k - \tfrac{1}{2}\log|\Sigma_k| - \tfrac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_k) - \tfrac{d}{2}\log(2\pi)\Big]
\end{aligned}
$$

$$
\begin{aligned}
(\boldsymbol{\mu}_k^{(t+1)}, \Sigma_k^{(t+1)}) &= \arg\max_{\boldsymbol{\mu}_k, \Sigma_k} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \\
&= \arg\max_{\boldsymbol{\mu}_k, \Sigma_k} \sum_{n=1}^{N} q^{(t)}(z_{nk}) \left(-\tfrac{1}{2}\log|\Sigma_k| - \tfrac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\right)
\end{aligned}
$$

Just a weighted version of the ML estimate for a single Gaussian.

$$
\begin{aligned}
\boldsymbol{\mu}_k^{(t+1)} &= {}^1\!/_{N\tau_k^{(t+1)}} \sum_{n=1}^{N} q^{(t)}\left(z_{nk}\right) \boldsymbol{x}_n \\
\Sigma_k^{(t+1)} &= {}^1\!/_{N\tau_k^{(t+1)}} \sum_{n=1}^{N} q^{(t)}\left(z_{nk}\right) \left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t+1)}\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t+1)}\right)^\top
\end{aligned}
$$

Summary

- EM is a “meta-algorithm” for obtaining local ML estimates
- Also applicable to maximum a-posteriori (MAP) estimation
- Particularly useful in models with latent variables z, where optimizing the incomplete-data likelihood directly is hard, but optimizing the complete-data likelihood $p(X, z|\theta)$ is easy.

$\rightarrow$ Alternate between estimating z and θ

- Can be *applied to* to a GMM, but EM is not equal to a GMM
- Other applications:
  - Hidden Markov Models (Baum-Welch algorithm)
  - Missing/incomplete data
  - Only summary data observed

Properties

Pro

- No stepsize/learning rate
- Each iteration improves likelihood

Con

- "Only" local minima found
- Solution dependent on initialization
- Can be slow

Note: sometimes possible to use generic solvers (e.g. Newton)

But:

- Complicated gradients, update rules
- No improvement guarantee (e.g., Jensen requires densities)

References

- A. P. Dempster, N. M. Laird and D. B. Rubin. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm", Journal of the Royal Statistical Society, B, vol. 39, no. 1, pp. 1-38.
- Neal, R, and Hinton, G (1999). Michael I. Jordan, ed. "A view of the EM algorithm that justifies incremental, sparse, and other variants". Learning in Graphical Models. Cambridge, MA: MIT Press: 355?368.
- Hastie, Tibshirani, and Friedman, The Elements of Statistical Learning, Chapter 8.5,