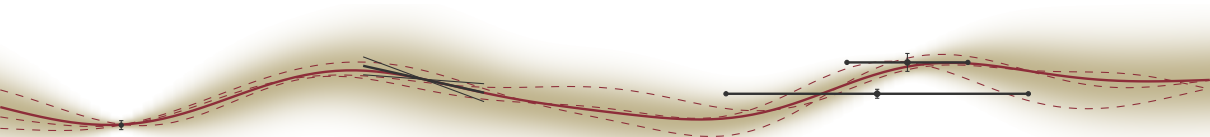
Probabilistic Inference and Learning
Lecture 19
The EM algorithm

Philipp Hennig

07 January 2019

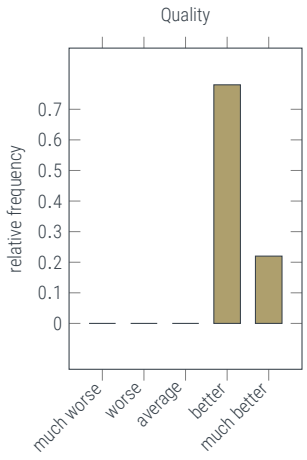
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

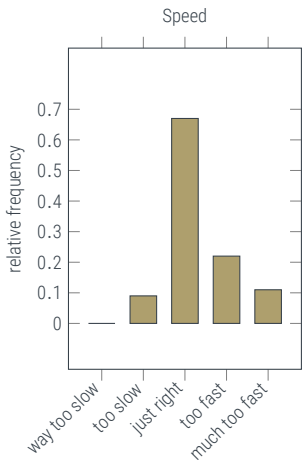
Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

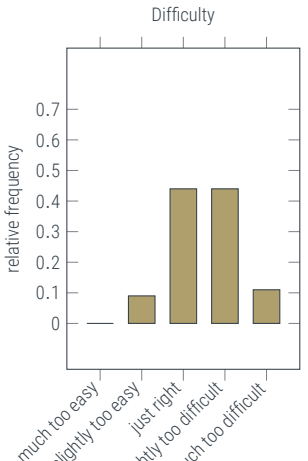


Last Lecture: Debrief

Feedback dashboard







Last Lecture: Debrief

Detailed Feedback

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Things you did not like:

- **second half was too fast** (especially Lagrange part)
- some of the variables were not defined on the slides (R_i)

Things you did not understand:

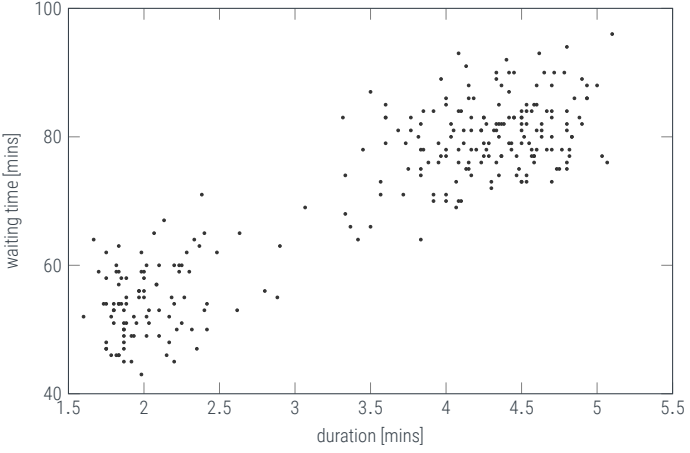
- how to set β
- in what way does introducing z decouple μ, Σ from π?

Things you enjoyed:

- **introduction to k-means**
- example exam

Recap: Clustering and Mixture Models

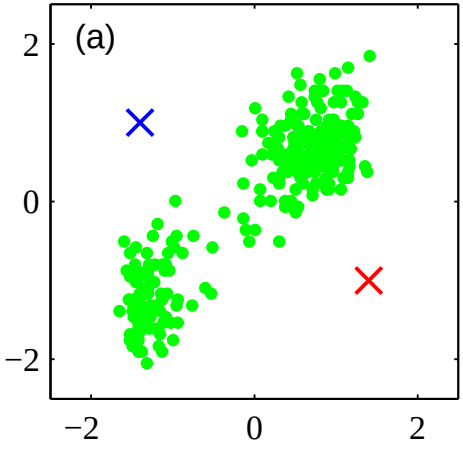
before the Christmas Break



Recap: *k*-Means

Example on Old Faithful

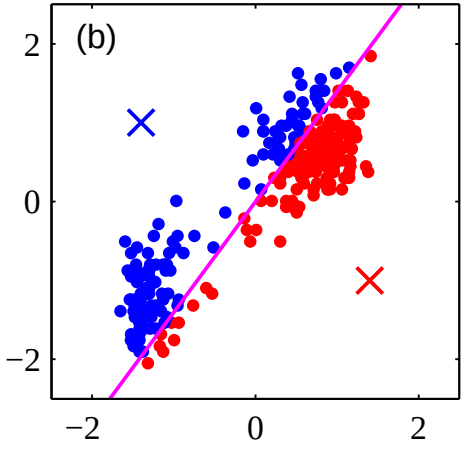
[Figure 9.1 in Bishop, 2006]



Recap: *k*-Means

Example on Old Faithful

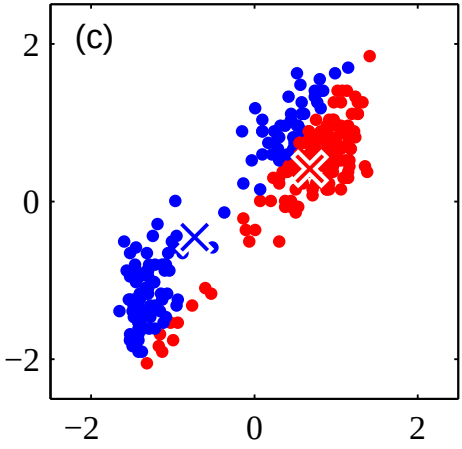
[Figure 9.1 in Bishop, 2006]



Recap: *k*-Means

Example on Old Faithful

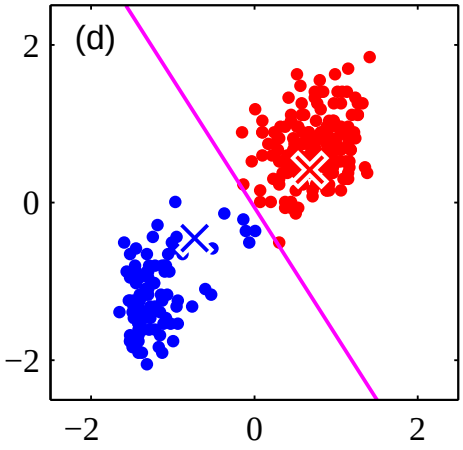
[Figure 9.1 in Bishop, 2006]



Recap: *k*-Means

Example on Old Faithful

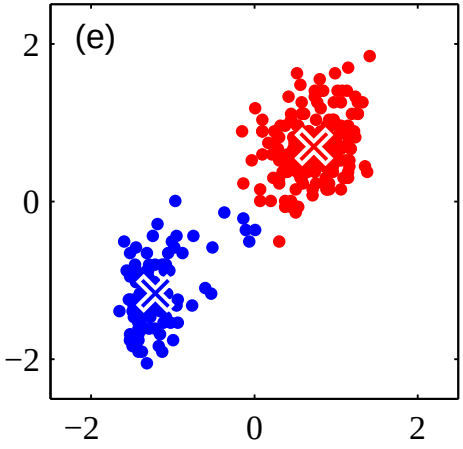
[Figure 9.1 in Bishop, 2006]



Recap: *k*-Means

Example on Old Faithful

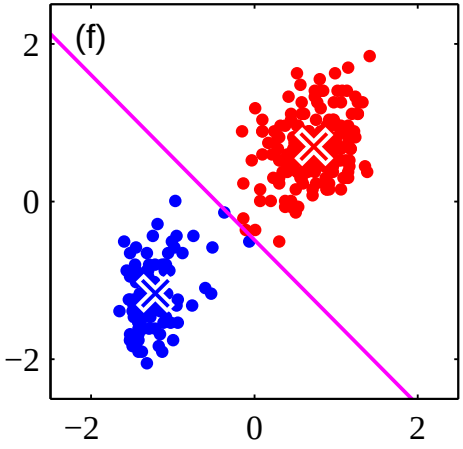
[Figure 9.1 in Bishop, 2006]



Recap: *k*-Means

Example on Old Faithful

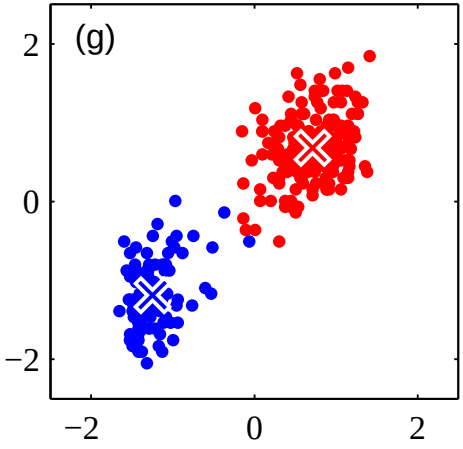
[Figure 9.1 in Bishop, 2006]



Recap: *k*-Means

Example on Old Faithful

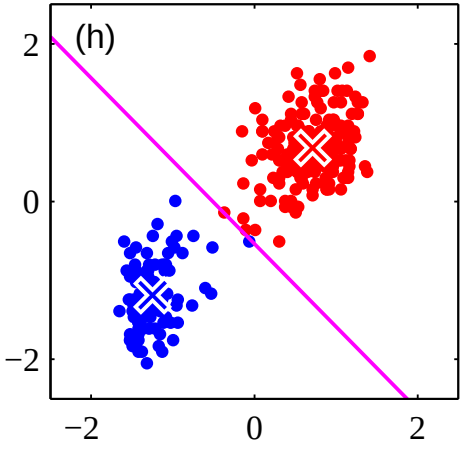
[Figure 9.1 in Bishop, 2006]



Recap: *k*-Means

Example on Old Faithful

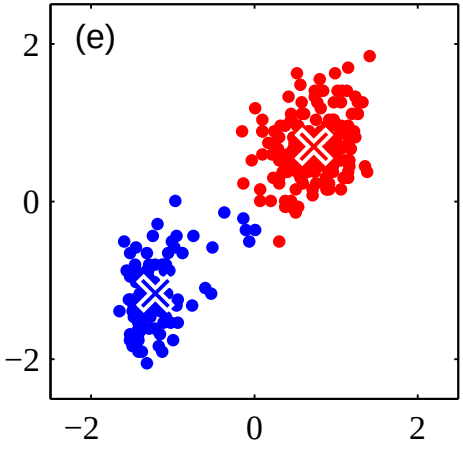
[Figure 9.1 in Bishop, 2006]



Recap: *k*-Means

Example on Old Faithful

[Figure 9.1 in Bishop, 2006]

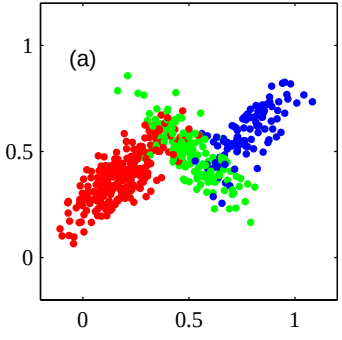


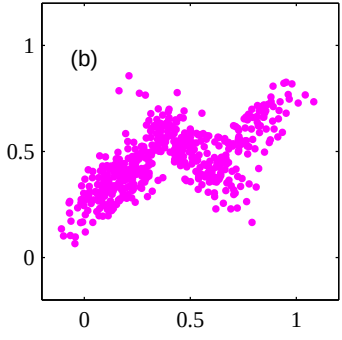
Gaussian Mixtures

A generative model for *k*-means

[Figure from Bishop, PRML, 2006]

$$p(x \mid \pi, \mu, \Sigma) = \sum_{j}^{k} \pi_j \mathcal{N}(x; \mu_j, \Sigma_j) \qquad \pi_j \in [0, 1], \quad \sum_j \pi_j = 1$$





$$p(x \mid \pi, \mu, \Sigma) = \sum_{j}^{k} \pi_j \mathcal{N}(x; \mu_j, \Sigma_j) \qquad \pi_j \in [0, 1], \quad \sum_j \pi_j = 1$$

In Lecture 18 we saw that …

- *k*-means finds a (local) maximum-likelihood solution for the Gaussian mixture model if $\Sigma \to 0$
- for $\Sigma = \beta \cdot I$, we get "soft *k*-means"
- for Gaussian mixtures, the maximum likelihood solution has no analytic form and must be found iteratively (e.g. by *k*-means)
- for the general parameter set (π, μ, Σ), there is an algorithm that alternates between setting π and μ, Σ, known as **Expectation Maximization (EM)**
- actually, EM alternates between setting μ, Σ to their maximum likelihood values and computing the *expected probability* of their class label z_{ij} (but max.-lik. for π is just a sum over z_{ij})

$$r_{ij} = p(z_{ij} = 1 \mid x_i, \mu, \Sigma) = \frac{\pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}{\sum_{j'} \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}$$

Today:

- another, alternative look at EM
- an intuition/proof sketch for why it converges

Why is this hard?

Latent Variables make Maximum Likelihood tricky

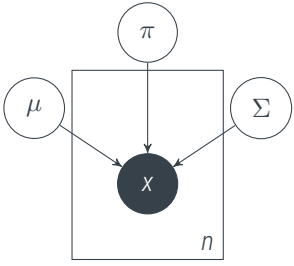
$$p(x, z \mid \pi, \mu, \Sigma) = \prod_i^n \prod_j^k \pi_j^{z_{ij}} \mathcal{N}(x_i; \mu_j, \Sigma_j)^{z_{ij}}$$

$$p(z) = \prod_j^k \pi_j^{z_j} \qquad p(x_i \mid z_{ij} = 1) = \mathcal{N}(x_i; \mu_j, \Sigma_j)$$

- We can see from the graph that $\mu, \Sigma \not\perp\!\!\!\perp z \mid x$ (Note how hard this would be from the joint!).
- Thus, even maximum likelihood inference is hard!

$$p(x \mid \pi, \mu, \Sigma) = \prod_i^n \sum_j^k \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)$$

- in the EM algorithm, we found an *iterative* way to maximize it

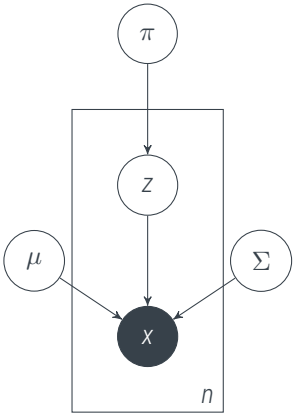


Why is this hard?

Latent Variables make Maximum Likelihood tricky

$$p(x, z \mid \pi, \mu, \Sigma) = \prod_i^n \prod_j^k \pi_j^{z_{ij}} \mathcal{N}(x_i; \mu_j, \Sigma_j)^{z_{ij}}$$

$$p(z) = \prod_j^k \pi_j^{z_j} \qquad p(x_i \mid z_{ij} = 1) = \mathcal{N}(x_i; \mu_j, \Sigma_j)$$



- We can see from the graph that $\mu, \Sigma \not\perp\!\!\!\perp z \mid x$ (Note how hard this would be from the joint!).
- Thus, even maximum likelihood inference is hard!

$$p(x \mid \pi, \mu, \Sigma) = \prod_i^n \sum_j^k \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)$$

- in the EM algorithm, we found an *iterative* way to maximize it

Reminder: Deriving the EM algorithm

iterative maximum likelihood for the Gaussian mixture model

Let's try to maximize the likelihood ($\star$) for π, μ, Σ (tool 1)

$$\log p(x \mid \pi, \mu, \Sigma) = \sum_{i}^{n} \log \left(\sum_{j}^{k} \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j) \right)$$

To maximize w.r.t. μ set gradient of log likelihood to 0:

$$\nabla_{\mu_j} \log p(x \mid \pi, \mu, \Sigma) = -\frac{1}{2} \sum_{i}^{n} \underbrace{\frac{\pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}{\sum_{j'} \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}}_{=: r_{ji}} \Sigma_j (x_i - \mu_j) + \text{const.}$$

$$\nabla_{\mu_j} \log p = 0 \quad \Rightarrow \mu_j = \frac{1}{R_j} \sum_{i}^{n} r_{ji} x_i \qquad R_j := \sum_{i} r_{ji}$$

Reminder: Deriving the EM algorithm

iterative maximum likelihood for the Gaussian mixture model

Let's try to maximize the likelihood ($\star$) for π, μ, Σ (tool 1)

$$\log p(x \mid \pi, \mu, \Sigma) = \sum_i^n \log \left(\sum_j^k \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j) \right)$$

To maximize w.r.t. Σ set gradient of log likelihood to 0:

$$\nabla_{\Sigma_j} \log p(x \mid \pi, \mu, \Sigma) = -\frac{1}{2} \sum_i^n \underbrace{\frac{\pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}{\sum_{j'} \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}}_{=:r_{ji}} (x_i - \mu_j)(x_i - \mu_j)^\intercal + \text{const.}$$

$$\nabla_{\Sigma_j} \log p = 0 \quad \Rightarrow \Sigma_j = \frac{1}{R_j} \sum_i^n r_{ji} (x_i - \mu_j)(x_i - \mu_j)^\intercal$$

Reminder: Deriving the EM algorithm

iterative maximum likelihood for the Gaussian mixture model

Let's try to maximize the likelihood ($\star$) for π, μ, Σ (tool 1)

$$\log p(x \mid \pi, \mu, \Sigma) = \sum_i^n \log \left(\sum_j^k \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j) \right)$$

To maximize w.r.t. π, enforce $\sum_j \pi_j = 1$ by introducing Lagrange multiplier λ and optimize

$$\nabla_{\pi_j} \left[\log p(x \mid \pi, \mu, \Sigma) + \lambda \left(\sum_j \pi_j - 1 \right) \right] = \sum_i^n \frac{\mathcal{N}(x_i; \mu_j, \Sigma_j)}{\sum_{j'} \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)} + \lambda$$

$$0 = \sum_i^n \pi_j \frac{\mathcal{N}(x_i; \mu_j, \Sigma_j)}{\sum_{j'} \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)} + \lambda \pi_j = \sum_i^n r_{ij} + \lambda \pi_j$$

$$\sum_j \pi_j = 1 \Rightarrow \lambda = -n \qquad \Rightarrow \qquad \pi_j = \frac{\sum_i r_{ij}}{n} =: \frac{R_j}{n}$$

The EM Algorithm (for Gaussian mixtures)

iterative maximum likelihood for the Gaussian mixture model

If we know the responsibilities r_{ij}, we can optimize μ, π analytically. And if we know μ, π, we can set r_{ij}! (As we could have guessed from the graph!) Thus

- initialize μ, π (e.g. random μ, uniform π). Then iterate:

E Set

$$r_{ij} = \frac{\pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}{\sum_{j'}^{k} \pi_{j'} \mathcal{N}(x_i; \mu_{j'}, \Sigma_{j'})}$$

M Set

$$R_j = \sum_i r_{ji} \qquad \mu_j = \frac{1}{R_j} \sum_i^n r_{ij} x_i \qquad \Sigma_j = \frac{1}{R_j} \sum_i^n r_{ij} (x_i - \mu_j)(x_i - \mu_j)^\intercal \qquad \pi_j = \frac{R_j}{n}$$

- Note that π is essentially given through r_{ij}, thus can be incorporated into the first step
- What is r_{ij}? It is the marginal posterior probability (**[E]xpectation**) for $z_{ij} = 1$:

$$p(z_{ij} = 1 \mid x, \mu, \Sigma, \pi) = \frac{p(z_{ij} = 1) p(x \mid z_{ij} = 1)}{\sum_{j'}^{k} p(z_{j'i} = 1) p(x \mid z_{j'i} = 1)} = \frac{\pi_j \mathcal{N}(x; \mu_j, \Sigma_j)}{\sum_{j'}^{k} \pi_{j'} \mathcal{N}(x_i; \mu_{j'}, \Sigma_{j'})}$$

Taking the easy way out

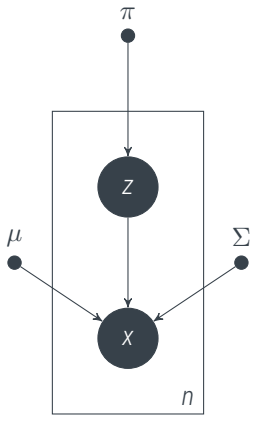
Just pretend you know that variable that causes trouble

$$p(x, z \mid \pi, \mu, \Sigma) = \prod_i^n \prod_j^k \pi_j^{z_{ij}} \mathcal{N}(x_i; \mu_j, \Sigma_j)^{z_{ij}}$$

$$p(x \mid z, \pi, \mu, \Sigma) = \prod_i^n \pi_{k_i} \mathcal{N}(x_i; \mu_{k_i}, \Sigma_{k_i})$$

$$\pi_j \leftarrow \frac{N_j}{N} \qquad N_j = \sum_i z_{ij}$$

$$\mu_j \leftarrow \frac{1}{N_j} \sum_i z_{ij} x_i \qquad \Sigma_j \leftarrow \frac{1}{N_j} \sum_i z_{ij} (x_i - \mu_j)(x_i - \mu_j)^\mathsf{T}$$



But we didn't have z! So, for EM, we replaced it with its expectation!

Generic EM Algorithm

Maximize **expected** log likelihoods

Setting:

- Want to find *maximum likelihood* (or MAP) estimate for a model involving a **latent** variable

$$\theta_* = \arg\max_\theta \left[\log p(x \mid \theta)\right] = \arg\max_\theta \left[\log\left(\sum_z p(x, z \mid \theta)\right)\right]$$

- Assume that the summation inside the log makes analytic optimization intractable
- but that optimization would be analytic if z were known (i.e. if there were only one term in the sum)

Idea: Initialize θ_0, then iterate between

1. Compute $p(z \mid x, \theta_{\text{old}})$
2. Set θ_{new} to the **Maximum** of the **Expectation** of the *complete-data* log likelihood:

$$\theta_{\text{new}} = \arg\max_\theta \sum_z p(z \mid x, \theta_{\text{old}}) \log p(\underbrace{x, z}_{!} \mid \theta) = \arg\max_\theta \mathbb{E}_{p(z|x,\theta_{\text{old}})} \left[\log p(x, z \mid \theta)\right]$$

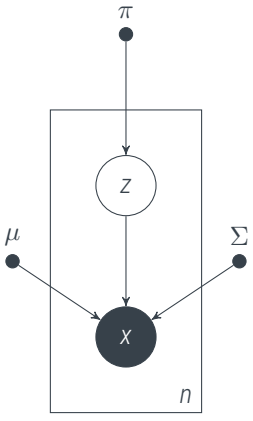
3. Check for convergence of either the log likelihood, or θ.

EM for Gaussian Mixtures

re-written in generic form

- Want to maximize, as function of $\theta := (\pi_j, \mu_j, \Sigma_j)_{j=1,\dots,k}$

$$\log p(x \mid \pi, \mu, \Sigma) = \sum_i \log \left(\sum_j \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j) \right)$$



EM for Gaussian Mixtures

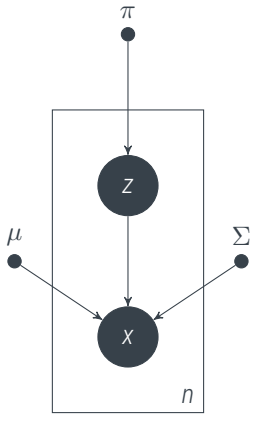
re-written in generic form

- Want to maximize, as function of $\theta := (\pi_j, \mu_j, \Sigma_j)_{j=1,\dots,k}$

$$\log p(x \mid \pi, \mu, \Sigma) = \sum_i \log \left(\sum_j \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j) \right)$$

- Instead, maximizing the "complete data" likelihood is easier:

$$\begin{aligned} \log p(x, z \mid \pi, \mu, \Sigma) &= \log \prod_i^n \prod_j^k \pi_j^{z_{ij}} \mathcal{N}(x_i; \mu_j, \Sigma_j)^{z_{ij}} \\ &= \sum_i \sum_j z_{nk} \underbrace{\left(\log \pi_j + \log \mathcal{N}(x_i; \mu_j, \Sigma_j) \right)}_{\text{easy to optimize (exponential families!)}} \end{aligned}$$



EM for Gaussian Mixtures

re-written in generic form

1. Compute $p(z \mid x, \theta)$:

$$p(z_{ij} = 1 \mid x_i, \mu, \Sigma) = \frac{p(z_{ij} = 1)p(x_i \mid z_{ij} = 1)}{\sum_{j'}^{k} p(z_{ij'} = 1)p(x_i \mid z_{ij'} = 1)} = \frac{\pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)}{\sum_{j'} \pi_j \mathcal{N}(x_i; \mu_j, \Sigma_j)} =: r_{ij}$$

2. Maximize

$$\mathbb{E}_{p(z|x,\theta)} \left(\log p(x, z \mid \theta) \right) = \sum_i \sum_j r_{ij} \left(\log \pi_j + \log \mathcal{N}(x_i; \mu_j, \Sigma_j) \right)$$

(see earlier slides on how to solve this, much easier problem)

The EM algorithm

Instead of trying to maximize

$$\log p(x \mid \theta) = \log \sum_z p(x, z \mid \theta),$$

instead maximize

$$\mathbb{E}_z \log p(x, z \mid \theta) = \sum_z p(z \mid x, \theta) \log p(x, z \mid \theta),$$

then re-compute $p(z \mid x, \theta)$, and repeat.

Why is this a good idea?

The EM algorithm

Instead of trying to maximize

$$\log p(x \mid \theta) = \log \sum_z p(x, z \mid \theta),$$

instead maximize

$$\mathbb{E}_z \log p(x, z \mid \theta) = \sum_z p(z \mid x, \theta) \log p(x, z \mid \theta),$$

then re-compute $p(z \mid x, \theta)$, and repeat.

Why is this a good idea?

An observation: By Jensen's inequality (log is concave!)

$$\sum_z \log p(x, z \mid \theta) \leq \log \sum_z p(x, z \mid \theta)$$

Lemma

Consider the probability distribution $p(x, z)$ and an arbitrary probability distribution $q(z)$ such that $q(z) > 0$ whenever $p(z) = \sum_x p(x, z) > 0$. Then the following equality holds:

$$\log p(x) = \mathcal{L}(q) + D_{KL}(q \| p)$$

$$\text{where} \quad \mathcal{L}(q) := \sum_z q(z) \log \left(\frac{p(x, z)}{q(z)} \right) \quad \text{and} \quad D_{KL}(q \| p) := -\sum_z q(z) \log \left(\frac{p(z \mid x)}{q(z)} \right).$$

Proof:

$$\log p(x, z) = \log p(z \mid x) + \log p(x) \qquad \text{(product rule)}$$

$$\Rightarrow \quad \mathcal{L}(q) = \sum_z q(z) \left(\log p(z \mid x) + \log p(x) - \log q(z) \right)$$

$$\Rightarrow \quad \mathcal{L}(q) + D_{\mathrm{KL}}(q \| p) = \sum_z q(z) \log p(x) = \log p(x) \qquad (q \text{ is probability distr.}) \quad \square$$

A Mathematical Insight

free energy / expectation lower bounds

[this exposition from Bishop, PRML, 2006]

Lemma

Consider the probability distribution $p(x, z)$ and an arbitrary probability distribution $q(z)$ such that $q(z) > 0$ whenever $p(z) = \sum_x p(x, z) > 0$. Then the following equality holds:

$$\log p(x) = \mathcal{L}(q) + D_{KL}(q \| p)$$

$$\text{where} \quad \mathcal{L}(q) := \sum_z q(z) \log \left(\frac{p(x, z)}{q(z)} \right) \quad \text{and} \quad D_{KL}(q \| p) := -\sum_z q(z) \log \left(\frac{p(z \mid x)}{q(z)} \right).$$

- $-\mathcal{L}(q)$ is known as the **Variational Free Energy** in physics, because

$$-\mathcal{L}(q) = -\mathbb{E}_q(\log p(x, z)) - \mathbb{H}(q) \quad \text{cf.} \quad F = U - TS$$

- note that $D_{KL}(q \| p) \geq 0$ with "$=$" iff $p \equiv q$. Thus $\mathcal{L}(q) \leq p(x)$, and it is also known as the **Expectation Lower Bound (ELBO)**

Historical Side-Note

Machine Learning is the application of scientific modelling to *everything*

Hermann
v. Helmholtz
1821–1894

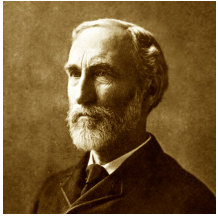
image:L. Meder

"Energy"

Ludwig Boltzmann
1844–1906

image:wikipedia

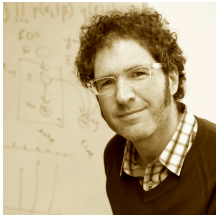
"Entropie"



Josia W. Gibbs
1839–1903

image:unknown

"Enthalpy"



David M. Blei

image:Columbia U

"ELBO"

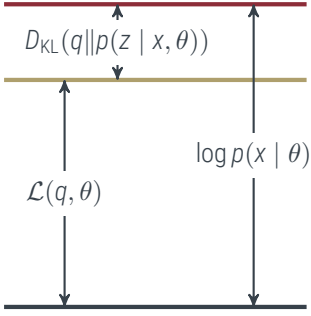
EM maximizes the ELBO / minimizes Free Energy

a more general view

$$\log p(x \mid \theta) = \mathcal{L}(q, \theta) + D_{\mathrm{KL}}(q \| p(z \mid x, \theta))$$

$$\mathcal{L}(q, \theta) = \sum_z q(z) \log \left(\frac{p(x, z \mid \theta)}{q(z)} \right)$$

$$D_{\mathrm{KL}}(q \| p(z \mid x, \theta)) = -\sum_z q(z) \log \left(\frac{p(z \mid x, \theta)}{q(z)} \right)$$



EM maximizes the ELBO / minimizes Free Energy

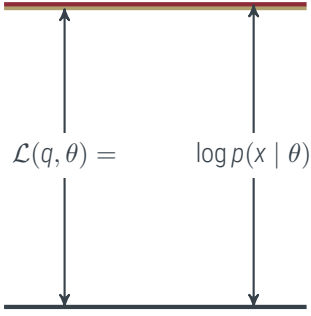
a more general view

$$\log p(x \mid \theta) = \mathcal{L}(q, \theta) + D_{\mathrm{KL}}(q \| p(z \mid x, \theta))$$

$$\mathcal{L}(q, \theta) = \sum_z q(z) \log \left(\frac{p(x, z \mid \theta)}{q(z)} \right)$$

$$D_{\mathrm{KL}}(q \| p(z \mid x, \theta)) = -\sum_z q(z) \log \left(\frac{p(z \mid x, \theta)}{q(z)} \right)$$

E -step: $q(z) = p(z \mid x, \theta_{\mathrm{old}})$, thus $D_{\mathrm{KL}}(q \| p(z \mid x, \theta_i)) = 0$



EM maximizes the ELBO / minimizes Free Energy

a more general view

$$\log p(x \mid \theta) = \mathcal{L}(q, \theta) + D_{\mathrm{KL}}(q \| p(z \mid x, \theta))$$

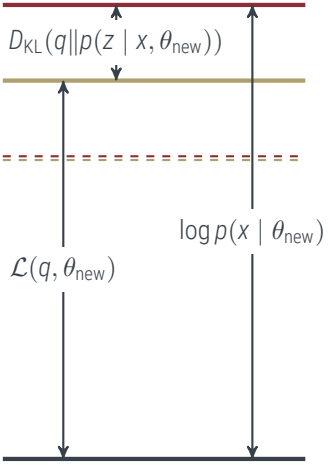
$$\mathcal{L}(q, \theta) = \sum_z q(z) \log \left(\frac{p(x, z \mid \theta)}{q(z)} \right)$$

$$D_{\mathrm{KL}}(q \| p(z \mid x, \theta)) = -\sum_z q(z) \log \left(\frac{p(z \mid x, \theta)}{q(z)} \right)$$

E -step: $q(z) = p(z \mid x, \theta_{\mathrm{old}})$, thus $D_{\mathrm{KL}}(q \| p(z \mid x, \theta_i)) = 0$

M -step: **Maximize ELBO / minimize Free Energy**

$$\begin{aligned} \theta_{\mathrm{new}} &= \arg\max_\theta \sum_z q(z) \log p(x, z \mid \theta) \\ &= \arg\max_\theta \mathcal{L}(q, \theta) + \sum_z q(z) \log q(z) \end{aligned}$$



EM Algorithm — General Form

for further generalization

Setting:

- Want to find *maximum likelihood* (or MAP) estimate for a model involving a **latent** variable

$$\theta_* = \arg\max_\theta \left[\log p(x \mid \theta)\right] = \arg\max_\theta \left[\log \left(\sum_z p(x, z \mid \theta)\right)\right]$$

- Assume that the summation inside the log makes analytic optimization intractable
- but that optimization would be analytic if z was known (i.e. if there were only one term in the sum)

Idea: Initialize θ_0, then iterate between

1. Compute $q(z) = p(z \mid x, \theta_{\text{old}})$, thereby setting $D_{\text{KL}}(q \| p(z \mid x, \theta)) = 0$
2. Set θ_{new} to the **Maximize** the **Expectation Lower Bound / minimize dhe Variational Free Energy**

$$\theta_{\text{new}} = \arg\max_\theta \mathcal{L}(q, \theta) = \arg\max_\theta \sum_z q(z) \log \left(\frac{p(x, z \mid \theta)}{q(z)}\right)$$

3. Check for convergence of either the log likelihood, or θ.

Some Observations

for future reference

- When we set $q(z) = p(z \mid x, \theta_{\text{old}})$, we set D_{KL} to its **minimum** $D_{\text{KL}}(q \| p(z \mid x, \theta) = 0$, thus

$$\begin{aligned} \nabla_\theta \log p(x \mid \theta_{\text{old}}) &= \nabla_\theta \mathcal{L}(q, \theta_{\text{old}}) + \nabla_\theta D_{\text{KL}}(q \| p(z \mid x, \theta_{\text{old}})) \\ &= \nabla_\theta \mathcal{L}(q, \theta_{\text{old}}) \end{aligned}$$

  So we could also use an optimizer based on this gradient to **numerically** optimize $\mathcal{L}$.
  This is known as **generalized EM**

- If $p(x, z \mid \theta)$ is an **exponential family** with θ as the natural parameters, then

$$\begin{aligned} p(x, z) &= \exp(\phi(x, z)^\intercal \theta - \log Z(\theta)) \\ \mathcal{L}(q(z), \theta) &= \mathbb{E}_{q(z)}(\phi(x, z)^\intercal \theta - \log Z(\theta)) \end{aligned}$$

  and optimization might be very easy, even analytic (Example: Gaussian mixtures).

- it is straightforward to extend EM to maximize a **posterior** instead of a likelihood (just add a log prior for θ)

The EM algorithm:

- to find *maximum likelihood* (or MAP) estimate for a model involving a **latent** variable

$$\theta_* = \arg\max_\theta \left[\log p(x \mid \theta)\right] = \arg\max_\theta \left[\log \left(\sum_z p(x, z \mid \theta)\right)\right]$$

- Initialize θ_0, then iterate between

E Compute $p(z \mid x, \theta_{\text{old}})$, thereby setting $D_{\text{KL}}(q \| p(z \mid x, \theta) = 0$

M Set θ_{new} to the **Maximize** the **Expectation Lower Bound / minimize the Variational Free Energy**

$$\theta_{\text{new}} = \arg\max_\theta \mathcal{L}(q, \theta) = \arg\max_\theta \sum_z q(z) \log \left(\frac{p(x, z \mid \theta)}{q(z)}\right)$$

- Check for convergence of either the log likelihood, or θ.

Next time: What if we can't even compute $p(z \mid x, \theta)$?