ASSIGNMENT 3: EXPECTATION MAXIMIZATION

Johannes Kofler,
Markus Holzleitner
Institute for Machine Learning

JOHANNES KEPLER UNIVERSITY LINZ

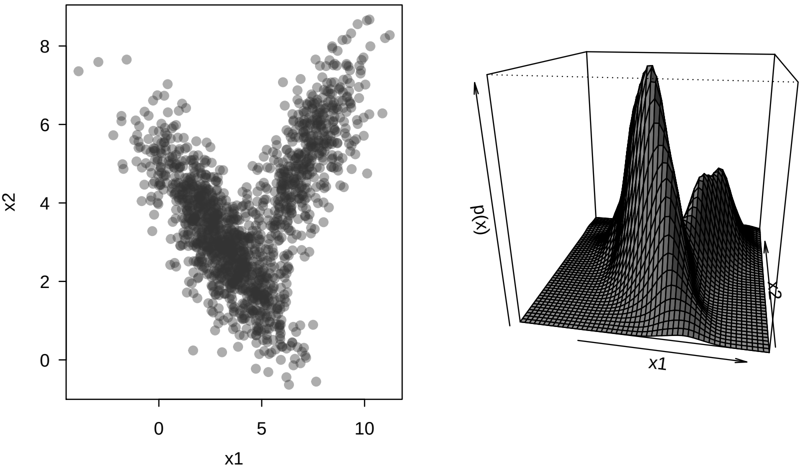
Institute for Machine Learning

Copyright statement:

Agenda:

- Theory of EM
- Example: Mixture Model
- Example: Mixture of Gaussians
- Success Stories

Mixture Model: Mixture of Gaussians (MoG)



Expectation Maximization (EM)

- a framework for optimizing **latent (hidden) variable models**
- suppose we are given data $\{x_i\}$
- our model, however, has the form $p(x, u; \theta)$ with θ a model parameter and u a latent (hidden) variable
- if $\{x_i, u_i\}$ were given, we could apply maximum likelihood

$$\ln \mathcal{L}(\theta) = \sum_{i}^{n} \ln p(x_i; \theta) = \sum_{i=1}^{n} \ln \sum_{u_i \in U} p(x_i, u_i; \theta)$$

- if the model parameter θ was given, we could estimate the hiddens $\{u_i\}$

$$\hat{u}_i = \arg\max_{u} p(u \,|\, x_i; \theta) = \arg\max_{u} \frac{p(x_i, u; \theta)}{\sum_{u \in U} p(x_i, u; \theta)}$$

EM: Intuition

- EM solves this chicken-egg problem in an iterative manner
- we start with a random initialization
- we keep optimizing one given the other, i.e.

1. estimate $\hat{u}_i$ given $\theta = \hat{\theta}$ (E-step)
2. estimate $\hat{\theta}$ given $u_i = \hat{u}_i$ (M-step)

- repeat until convergence

EM: Theory (1/3)

- introduce some candidate distribution $Q(u \,|\, x)$
- improve it iteratively s.t. $Q(u \,|\, x) \to p(u \,|\, x)$

$$
\begin{aligned}
\ln \mathcal{L}(\theta \,|\, x) &= \ln p(x) = \ln \int_U p(x,u) \mathrm{d}u \\
&= \ln \int_U \frac{Q(u \,|\, x)}{Q(u \,|\, x)} p(x,u) \mathrm{d}u \\
&= \ln \mathbb{E}_Q \left(\frac{p(x,u)}{Q(u \,|\, x)} \right) \\
&\geq \mathbb{E}_Q \left(\ln \frac{p(x,u)}{Q(u \,|\, x)} \right)
\end{aligned}
$$

- the last step is Jensen's inequality

EM: Theory (2/3)

- the amount by how much we fail to optimize the true log-likelihood is the Kullback-Leibler divergence of $Q(u \mid x)$ and $p(u \mid x)$

$$
\begin{aligned}
\mathbb{E}_Q \left(\ln \frac{p(x,u)}{Q(u \mid x)} \right) &= \int_U Q(u \mid x) \ln \frac{p(x) p(u \mid x)}{Q(u \mid x)} \, \mathrm{d}u \\
&= \int_U Q(u \mid x) \ln p(x) \, \mathrm{d}u + \int_U Q(u \mid x) \ln \frac{p(u \mid x)}{Q(u \mid x)} \, \mathrm{d}u \\
&= \ln p(x) - D_{\mathrm{KL}}(Q(u \mid x) \,\|\, p(u \mid x)) \\
&= \ln \mathcal{L}(\theta \mid x) - D_{\mathrm{KL}}(Q(u \mid x) \,\|\, p(u \mid x))
\end{aligned}
$$

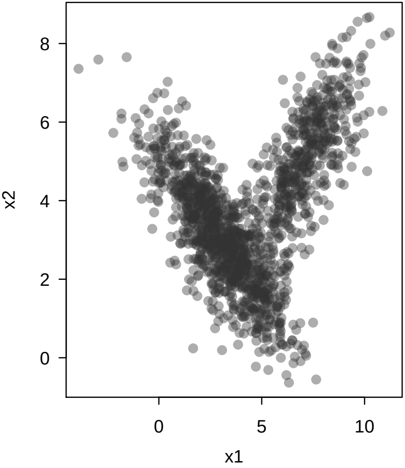
EM: Theory (3/3)

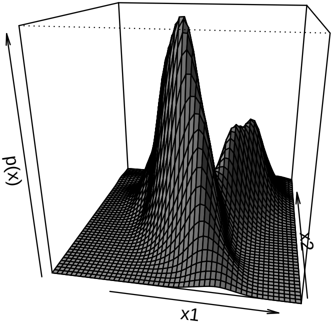
- this gives us another interpretation for what happens in EM
- in the M-step, we obtain a lower bound on the true likelihood function
- in the E-step, we make the bound tight (i.e. equality holds)

$$\theta := \arg\max_{\theta} \mathbb{E}_Q \left(\ln \frac{p(x, u; \theta)}{Q(u \,|\, x)} \right) \qquad \text{M-step}$$

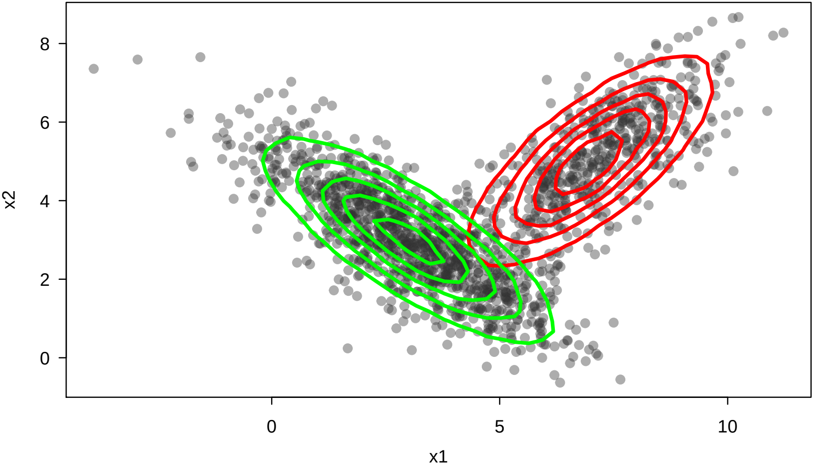
$$Q(u \,|\, x) := p(u \,|\, x; \theta) \qquad \text{E-step}$$

Mixture Model: Mixture of Gaussians (MoG)





Mixture Model: Mixture of Gaussians (MoG)



Mixture Model: Log-Likelihood

$$p(x_i) = \sum_{k=1}^{K} \alpha_k p_k(x_i; \theta_k)$$

$$\ln \mathcal{L}(\Theta \,|\, x_1, \ldots, x_n) = \ln \prod_{i=1}^{n} p(x_i) = \sum_{i=1}^{n} \ln \left(\sum_{k=1}^{K} \alpha_k p_k(x_i; \theta_k) \right)$$

- K is a hyper-parameter
- parameter set $\Theta = \{\alpha_1, \ldots, \alpha_K, \theta_1, \ldots, \theta_K\}$
- mixing coefficients $\alpha_1, \ldots, \alpha_K$
  - follow a categorical distribution, i.e. $\sum_k \alpha_k = 1$
- component parameters $\theta_1, \ldots, \theta_K$
  - govern the K component distributions $p(x; \theta_k)$
  - in case of MoG, we have $\theta_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$
- The data are assumed to be drawn i.i.d. from $p(x)$.

Mixture Model: Optimizing for θ_k

$$
\begin{aligned}
\frac{\partial \ln \mathcal{L}}{\partial \theta_k} &= \sum_{i=1}^{n} \frac{\alpha_k}{\sum_{\ell=1}^{K} \alpha_\ell p(x_i;\theta_\ell)} \frac{\partial p(x_i;\theta_k)}{\partial \theta_k} \\
&= \sum_{i=1}^{n} \frac{\alpha_k}{\sum_{\ell=1}^{K} \alpha_\ell p(x_i;\theta_\ell)} \frac{p(x_i;\theta_k)}{p(x_i;\theta_k)} \frac{\partial p(x_i;\theta_k)}{\partial \theta_k} \\
&= \sum_{i=1}^{n} p(u_i = k \mid x_i) \frac{\partial \ln p(x_i;\theta_k)}{\partial \theta_k} \qquad (1)
\end{aligned}
$$

- we introduced u_i, which are
  - hidden variables indicating component membership
  - realizations of the categorical distribution governed by α_k, i.e. $\alpha_k = p(u_i = k)$ for all i.
- given $p(u_i = k \mid x_i)$ allows us to estimate θ_k the usual way

Mixture Model: Optimizing for α_k

Lagrangian of the log-likelihood and the sum-to-one constraint

$$\Lambda(\Theta, \lambda) = \sum_{i=1}^{n} \ln \left(\sum_{\ell=1}^{K} \alpha_\ell p(x_i; \theta_\ell) \right) + \lambda \left(\sum_{\ell=1}^{K} \alpha_\ell - 1 \right)$$

$$\frac{\partial \Lambda(\Theta, \lambda)}{\partial \alpha_k} = \sum_{i=1}^{n} \frac{p(x_i; \theta_k)}{\sum_{\ell=1}^{m} \alpha_\ell p(x_i; \theta_\ell)} + \lambda = 0$$

$$\sum_{i=1}^{n} p(u_i = k \,|\, x_i) + \alpha_k \lambda = 0$$

$$\sum_{k=1}^{K} \sum_{i=1}^{n} p(u_i = k \,|\, x_i) + \sum_{k=1}^{K} \alpha_k \lambda = 0$$

$$\lambda = -n \quad \Rightarrow \quad \alpha_k = \frac{1}{n} \sum_{i=1}^{n} p(u_i = k \,|\, x_i)$$

Mixture Model: Computing $p(u_i = k \mid x_i)$

- both solutions for θ_k and α_k involve the term $r_{ik} := p(u_i = k \mid x_i)$ ("responsibility").
- r_{ik} is a “soft” assignment of x_i to the K components
  - “soft” meaning in terms of probabilities instead of one-hot
- given θ_k and α_k we can just compute r_{ik} using Bayes’ theorem

$$r_{ik} = \frac{p(u_i = k)p(x_i \mid u_i = k)}{p(x_i)} = \frac{\alpha_k p(x_i; \theta_k)}{\sum_{\ell=1}^{K} \alpha_\ell p(x_i; \theta_\ell)}$$

Mixture Model: Applying EM

- estimates for θ_k and α_k are only optimal for given r_{ik}
- estimates for r_{ik} are only correct for given θ_k and α_k
- instanciate EM for mixture models by performing the following steps

1. optimize θ_k, α_k for given r_{ik} (M-step)
2. compute r_{ik} for given θ_k, α_k (E-step)

Mixture of Gaussians (MoG)

- We choose $\theta_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}, \boldsymbol{\mu}_k \in \mathbb{R}^d, \boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}$ and the likelihood function for the k-th component is

$$p(\mathbf{x}_i; \theta_k) = \det(2\pi\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)\right)$$

- Taking the log simplifies the situation to

$$\ln p(\mathbf{x}_i; \theta_k) = -\frac{d}{2}\ln 2\pi - \frac{1}{2}\left(\ln\det\boldsymbol{\Sigma}_k + (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)\right)$$

  where the first summand is just a constant offset (irrelevant for optimization).
- $\rightarrow$ redefine the log-likelihood function and use

$$\ln p(\mathbf{x}_i; \theta_k) = -\frac{1}{2}\left(\ln\det\boldsymbol{\Sigma}_k + (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)\right)$$

- Computing the maximum likelihood explicitly, like for one Gaussian, is infeasible for mixture model $\rightarrow$ EM-algorithm

MoG: Optimizing for $\boldsymbol{\mu}_k$

$$\begin{aligned}
\frac{\partial \ln p(\mathbf{x}_i;\theta_k)}{\partial \boldsymbol{\mu}_k} &= -(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} = 0 \\
\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}_k^{-1} &= \mathbf{x}_i^T \boldsymbol{\Sigma}_k^{-1} \\
\boldsymbol{\mu}_k &= \mathbf{x}_i
\end{aligned}$$

plugging into equation (1) gives

$$\begin{aligned}
\boldsymbol{\mu}_k \sum_{i=1}^{n} r_{ik} &= \sum_{i=1}^{n} r_{ik}\mathbf{x}_i \\
\boldsymbol{\mu}_k &= \frac{\sum_{i=1}^{n} r_{ik}\mathbf{x}_i}{\sum_{i=1}^{n} r_{ik}}
\end{aligned}$$

MoG: Optimizing for Σ_k

$$\frac{\partial \ln p(\mathbf{x}_i; \theta_k)}{\partial \mathbf{\Sigma}_k} = -\frac{1}{2}\left(\mathbf{\Sigma}_k^{-1} - \mathbf{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1}\right) = 0$$

$$\mathbf{\Sigma}_k^{-1} = \mathbf{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1}$$

$$\mathbf{\Sigma}_k = (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

plugging into equation (1) gives

$$\mathbf{\Sigma}_k \sum_{i=1}^{n} r_{ik} = \sum_{i=1}^{n} r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

$$\mathbf{\Sigma}_k = \frac{\sum_{i=1}^{n} r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{\sum_{i=1}^{n} r_{ik}}$$

MoG: Putting it Together

M-Step:

$$\alpha_k := \frac{1}{n}\sum_{i=1}^{n} r_{ik}$$

$$\boldsymbol{\mu}_k := \frac{\sum_{i=1}^{n} r_{ik}\mathbf{x}_i}{\sum_{i=1}^{n} r_{ik}}$$

$$\boldsymbol{\Sigma}_k := \frac{\sum_{i=1}^{n} r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{\sum_{i=1}^{n} r_{ik}}$$

E-Step:

$$r_{ik} := p(u_i = k \mid x_i; \alpha_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

MoG: Example

taken from Bishop, “Pattern Recognition and Machine Learning” (2006), p437