Expectation-Maximization Algorithm

ELI CHERTKOV, AIG TALK 2/8/2021

What does the EM algorithm do?

Performs statistical inference on probability models with hidden (latent) variables.

$$\underbrace{P(x, z; \theta)}_{\text{visible variables},\ \text{hidden variables},\ \text{model parameters}} \quad \Rightarrow \quad P(x; \theta) = \sum_z P(x, z; \theta)$$

visible variables　hidden variables　model parameters

EM can be used for:

- Unsupervised learning (learning features of an unlabeled dataset)
- Clustering

EM is commonly used with:

- Gaussian mixture models
- Hidden Markov models (called Baum-Welsch in this context)

Maximum likelihood estimation

MLE is a method for estimating the parameters of a probability model from a dataset.

The idea is to find the parameters θ that maximize the likelihood that the data $x^{(1)}, \ldots, x^{(N)}$ came from the probability distribution $P(x; \theta)$:

$$\hat{\theta} = argmax\ L(\theta) = argmax \prod_{n=1}^{N} P\left(x^{(n)}; \theta\right) \iff \hat{\theta} = argmax\ l(\theta) = argmax \sum_{n=1}^{N} \log P\left(x^{(n)}; \theta\right)$$

For certain simple probability distributions, the optimal MLE parameters have closed form expressions.

Example: Multivariate Gaussian

$$P(x; \theta = \{\mu, \Sigma\}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}}\ e^{-(\mathrm{x}-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu)/2}$$

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} x^{(n)}$$

$$\hat{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} \left(x^{(n)} - \hat{\mu}\right)\left(x^{(n)} - \hat{\mu}\right)^{T}$$

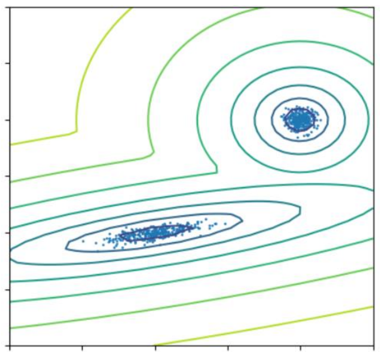
But even slightly more complicated examples do not have closed form solutions! EM can help with this.

Gaussian Mixture Models

We will focus on a particular class of probability distributions (a sum of Gaussians):

$$P(x, z; \theta = \{\phi_1, \dots, \phi_K, \mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\}) = \phi_z P_z(x; \mu_z, \Sigma_z)$$

$K = 2$ example



where $z \in \{1, \dots, K\}$, $P_z(x; \mu_z, \Sigma_z)$ is a Gaussian distribution, and $\sum_k \phi_k = 1$.

Notes:

- You can think of z as being a "cluster label" where we have K Gaussian clusters.
- For large enough K, a GMM can approximate any probability distribution.
- There are closed-form expressions for the MLE parameters of $P(x, z; \theta)$.
- But NO closed-form expressions for the marginal distribution $P(x; \theta) = \sum_z P(x, z; \theta)$.

EM algorithm motivation

By introducing a distribution $Q_n(z)$, we can find a lower bound for the log-likelihood:

$$l(\theta) = \sum_n \log p\left(x^{(n)}; \theta\right) = \sum_n \log \sum_z p\left(x^{(n)}, z; \theta\right)$$

$$= \sum_n \log \sum_z Q_n(z) \frac{p\left(x^{(n)}, z; \theta\right)}{Q_n(z)} \geq \sum_n \sum_z Q_n(z) \log \frac{p\left(x^{(n)}, z; \theta\right)}{Q_n(z)}$$

(since log is concave, by Jensen's inequality)

The LHS and RHS are equal when $Q_n(z) = P\left(z \middle| x^{(n)}; \theta\right) = P(x^{(n)}, z; \theta) / (\sum_{z'} P(x^{(n)}, z'; \theta))$

The idea of the EM algorithm is that we will iteratively construct $Q_n(z)$ distributions
that provide larger and larger lower bounds on the true log-likelihood $l(\theta)$.

EM algorithm

Initialize θ_0

For $t = 1, \ldots, T$ (or until converged)

E-step:

Set $Q_n(z) = P\left(z \middle| x^{(n)}; \theta_{t-1}\right)$ = "expectation that $x^{(n)}$ is in cluster z for parameters θ_{t-1}"

M-step

Find $\theta_t = argmax_\theta \ \sum_n \sum_z Q_n(z) \log \frac{p\left(x^{(n)}, z; \theta_{t-1}\right)}{Q_n(z)}$

EM algorithm for GMMs

Initialize $\phi_k = \frac{1}{K}, \mu_k = random\ x^{(n)}, \Sigma_\mathrm{k} = cov(data)$

For $t = 1, \dots, T$ (or until converged)

E-step:

Set $Q_n(z) = P\left(z \middle| x^{(n)}; \theta_{t-1}\right) = \phi_z P_z\left(x^{(n)}; \theta_{z,t-1}\right) / (\sum_k \phi_k P_k\left(x^{(n)}; \theta_{k,t-1}\right))$

M-step

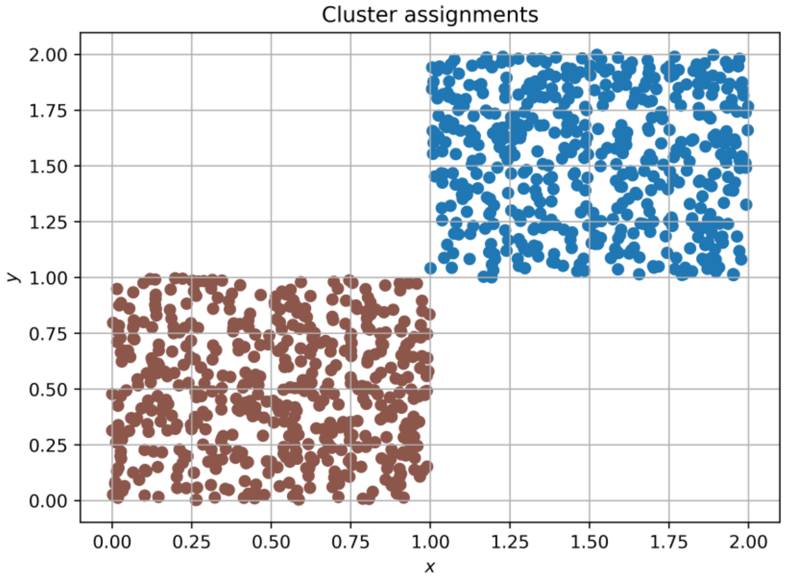
$$\phi_{k,t} = N_k / N \qquad \mu_{k,t} = \frac{1}{N_k} \sum_n Q_n(k)\, x^{(n)} \qquad (\, N_k = \sum_n Q_n(k) \,)$$

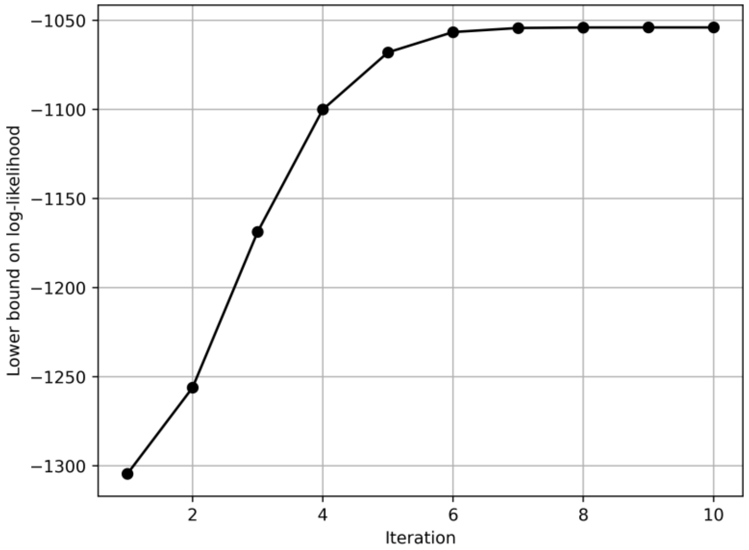
$$\Sigma_{k,t} = \frac{1}{N_k} \sum_n Q_n(k)\, \left(x^{(n)} - \mu_{k,t}\right)\left(x^{(n)} - \mu_{k,t}\right)^T$$

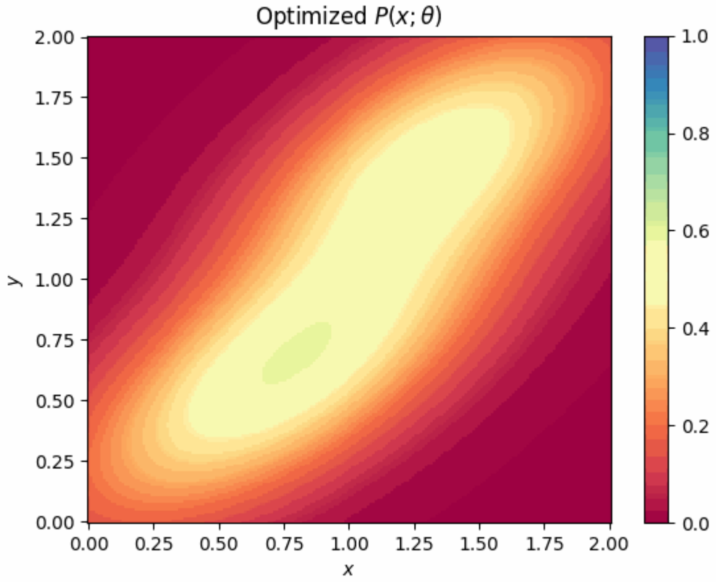
Note: For GMMs, the optimal parameter updates have closed-form expressions.

Example 1

$K = 2$ Gaussian clusters

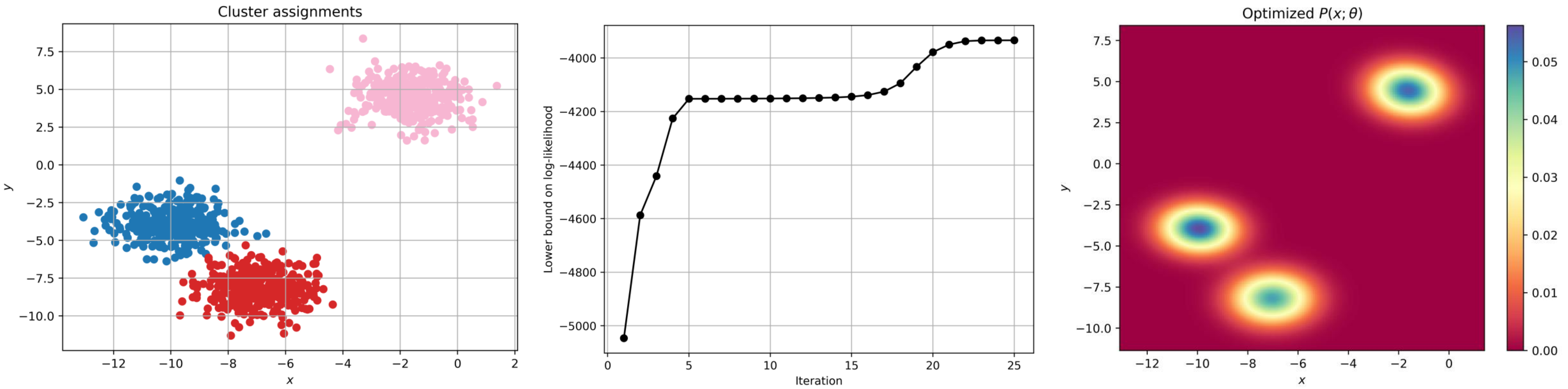






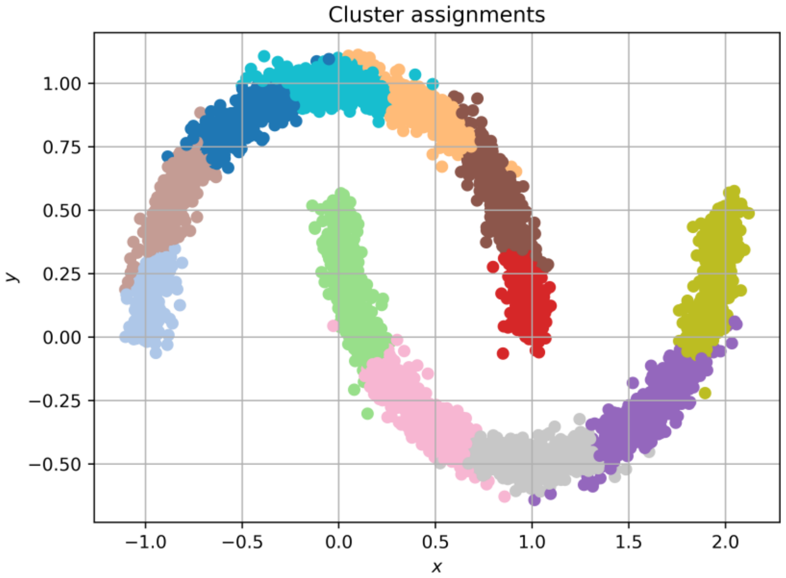
Example 2

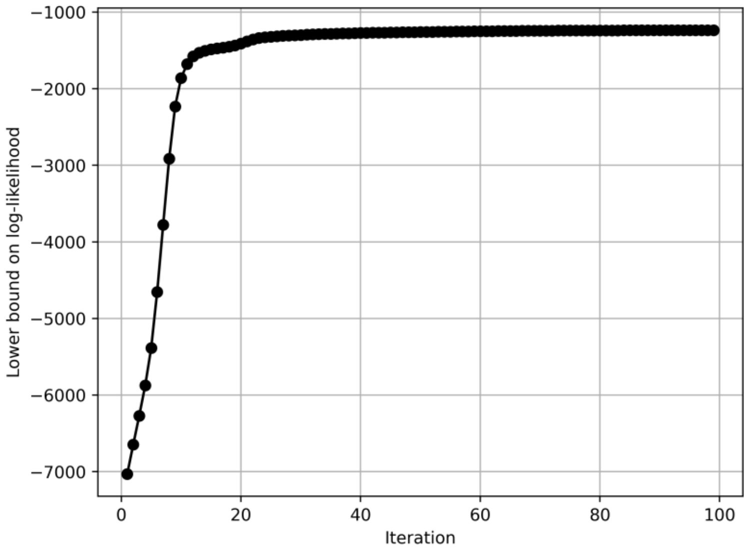
$K = 3$ Gaussian clusters

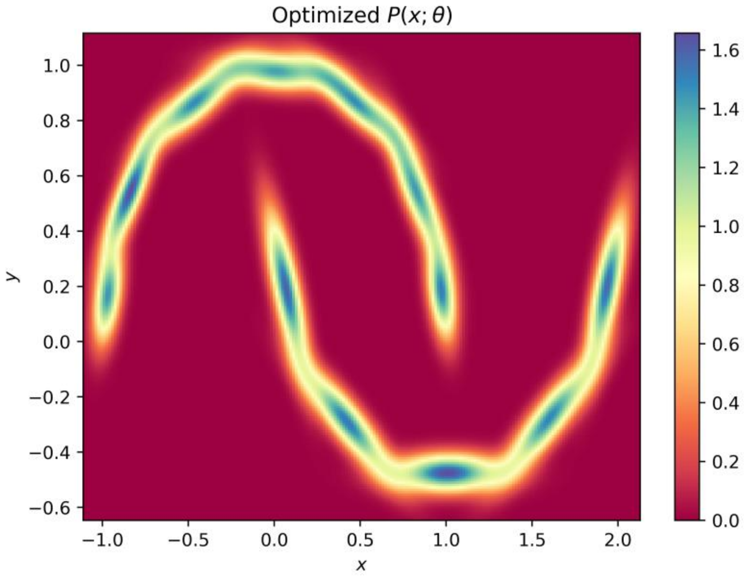


Example 3

$K = 12$ Gaussian clusters







Summary

- EM algorithm does inference on hidden variable models
- Used to fit Gaussian Mixture Models to datasets
- Based on iteratively maximize a lower-bound on log-likelihood

Pros:

- General method that can be applied to all hidden variable models.
- Simple and efficient for GMMs.
- Guaranteed to converge (locally).

Cons:

- Can get stuck in local minima.
- Other methods might work better if EM update rules don't have closed form expressions.

References

Andrew Ng's Stanford CS 229 Lectures

Lecture notes: https://see.stanford.edu/materials/aimlcs229/cs229-notes8.pdf

Video: https://www.youtube.com/watch?v=rVfZHWTwXSA

Padhraic Smyth's UCI CS 274 Lecture notes

https://www.ics.uci.edu/~smyth/courses/cs274/notes/EMnotes.pdf