Advanced probabilistic methods

Lecture 5: Mixture models and EM

Pekka Marttinen

Aalto University

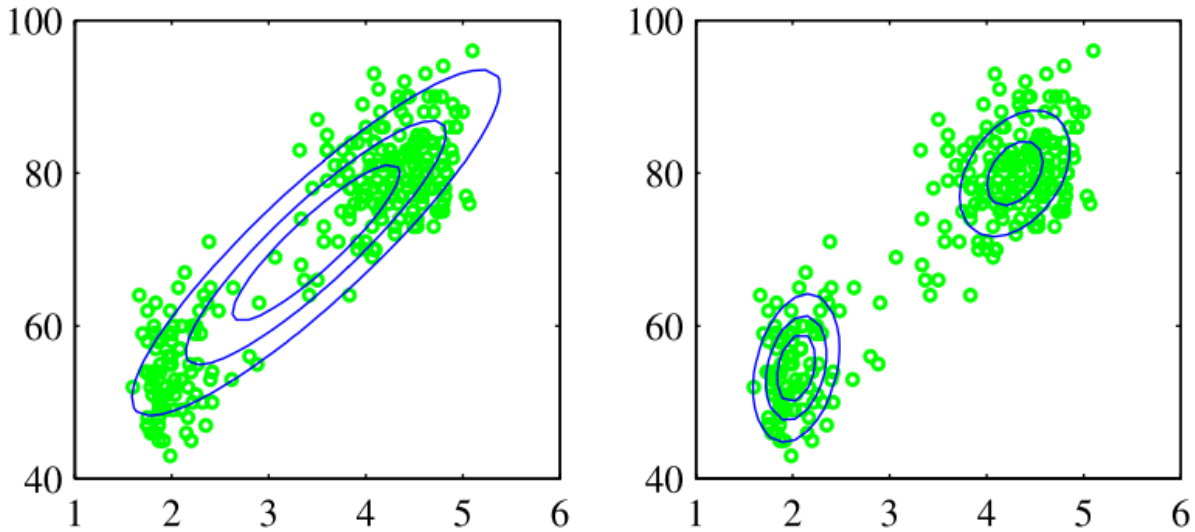
February, 2019

Lecture 5 overview

- Gaussian mixture models (GMMs)
- EM algorithm
- EM for Gaussian mixture models
- Suggested reading: Bishop: *Pattern Recognition and Machine Learning*
  - p. 110-113 (2.3.9): Mixtures of Gaussians
  - *simple_example.pdf*
  - p. 430-443: EM for Gaussian mixtures

Gaussian mixture models (motivation)

- Standard Gaussian model (left) gives bad fit to data with clusters
- Combination of two Gaussians (right) is much better

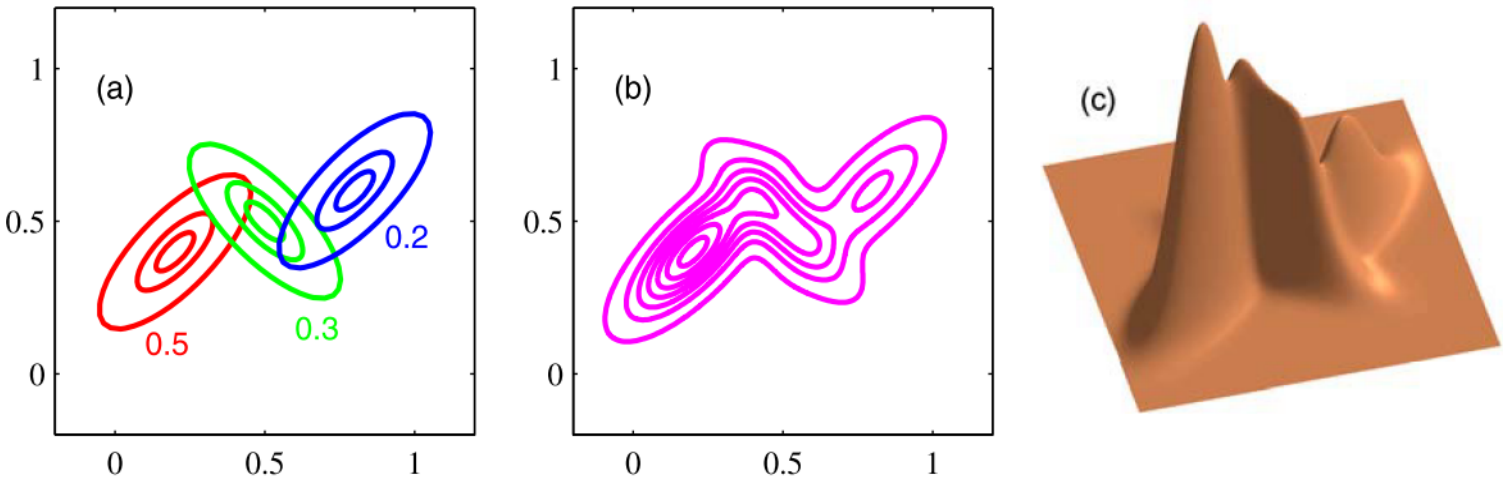


Gaussian mixture models

- Gaussian mixture model with K components has density

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k N(\mathbf{x}|\mu_k, \Sigma_k).$$

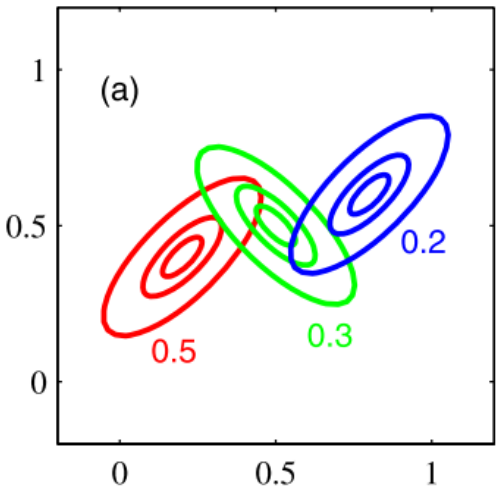
- $N(x|\mu_k, \Sigma_k)$ is a **component** with its own mean μ_k and covariance Σ_k.
- π_k are the **mixing coefficients**, which satisfy $\sum_k \pi_k = 1$, $0 \leq \pi_k \leq 1$.

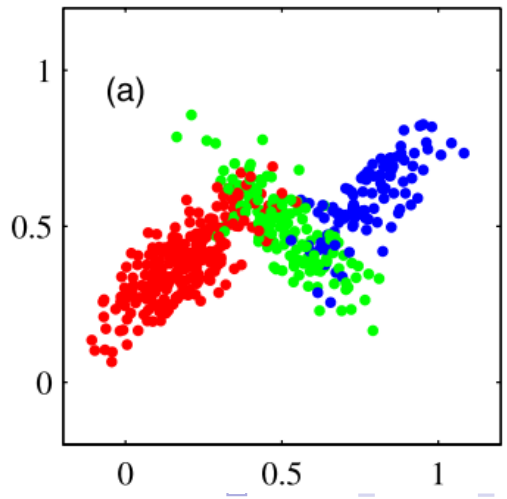


GMMs, latent variable representation (1/2)

- Equivalent formulation is obtained by defining **latent variables** $\mathbf{z}_n = (z_{n1}, \ldots, z_{nK})$ which tell the component for observation $\mathbf{x}_n$
- In detail $\mathbf{z}_n$ is a vector with exactly one element equal to 1 and other elements equal to 0. $z_{nk} = 1$ means that the observation $\mathbf{x}_n$ belongs to component k.

$$\mathbf{z}_n = (0, \ldots, 0, \underbrace{1}_{k^{th} \text{ elem.}}, 0, \ldots, 0)^T$$





GMMs, latent variable representation (2/2)

- Define

$$p(z_{nk} = 1) = \pi_k \qquad \text{and} \qquad p(\mathbf{x}_n | z_{nk} = 1) = N(\mathbf{x}_n | \mu_k, \Sigma_k),$$

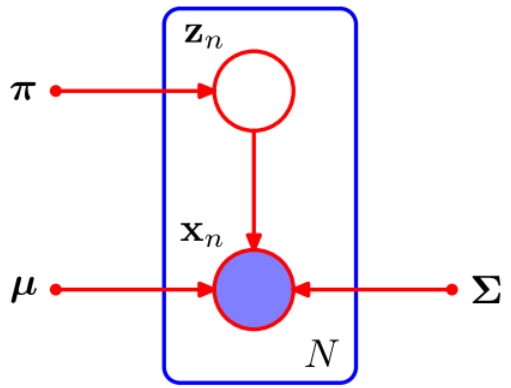
or equivalently

$$p(\mathbf{z}_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}} \quad \text{and} \quad p(\mathbf{x}_n | \mathbf{z}_n) = \prod_{k=1}^{K} N(\mathbf{x}_n | \mu_k, \Sigma_k)^{z_{nk}}$$

- Then

$$p(\mathbf{x}_n) = \sum_{z_n} p(\mathbf{z}_n) p(\mathbf{x}_n | \mathbf{z}_n) = \sum_k \pi_k N(\mathbf{x}_n | \mu_k, \Sigma_k)$$

$\rightarrow \mathbf{x}_n$ has marginally the Gaussian mixture model distribution.



GMM: responsibilities (1/2)

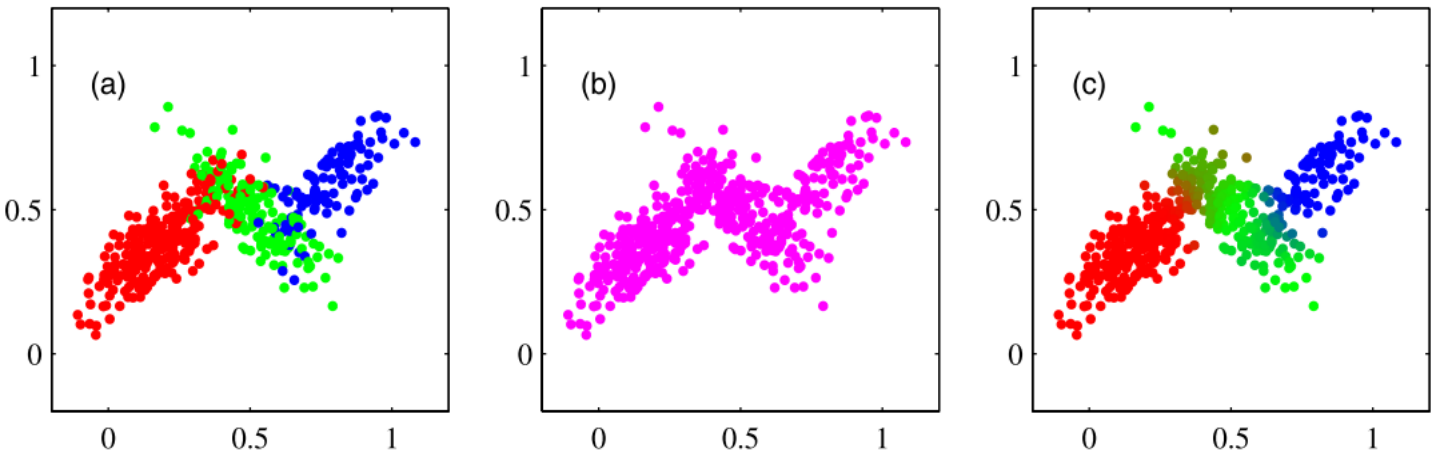
- Posterior probability $p(z_{nk} = 1|\mathbf{x}_n)$ that observation $\mathbf{x}_n$ was generated by component k

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

- $\gamma(z_{nk})$ can be viewed as the **responsibility** that component k takes for explaining the observation $\mathbf{x}_n$

GMM: responsibilities (2/2)

- (left) samples from a joint distribution $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$, showing both cluster labels $\mathbf{z}$ and observations $\mathbf{x}$ (**complete** data)
- (center) samples from the marginal distribution $p(\mathbf{x})$ (**incomplete** data)
- (right) **responsibilities** of the data points, computed using *known* parameters $\pi = (\pi_1, \ldots, \pi_K)$, $\mu = \mu_1, \ldots, \mu_K$, $\Sigma = (\Sigma_1, \ldots, \Sigma_K)$.
- Problem: in practice π, μ, and Σ are usually *unknown*.



Idea of the EM algorithm (1/2)

- Let X denote the observed data, and θ model parameters. The goal in maximum likelihood is to find $\widehat{\theta}$:

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

- If model contains latent variables Z, the log-likelihood is given by

$$\log p(X|\theta) = \log \left\{ \sum_{Z} p(X, Z|\theta) \right\},$$

  which may be difficult to maximize analytically
- Possible solutions: 1) numerical optimization, 2) the EM algorithm (expectation-maximization)

Idea of the EM algorithm (2/2)

- X: **observed** data, Z: **unobserved** latent variables
- $\{X, Z\}$: **complete** data, X: **incomplete** data
- In EM algorithm, we assume that the complete data log-likelihood:

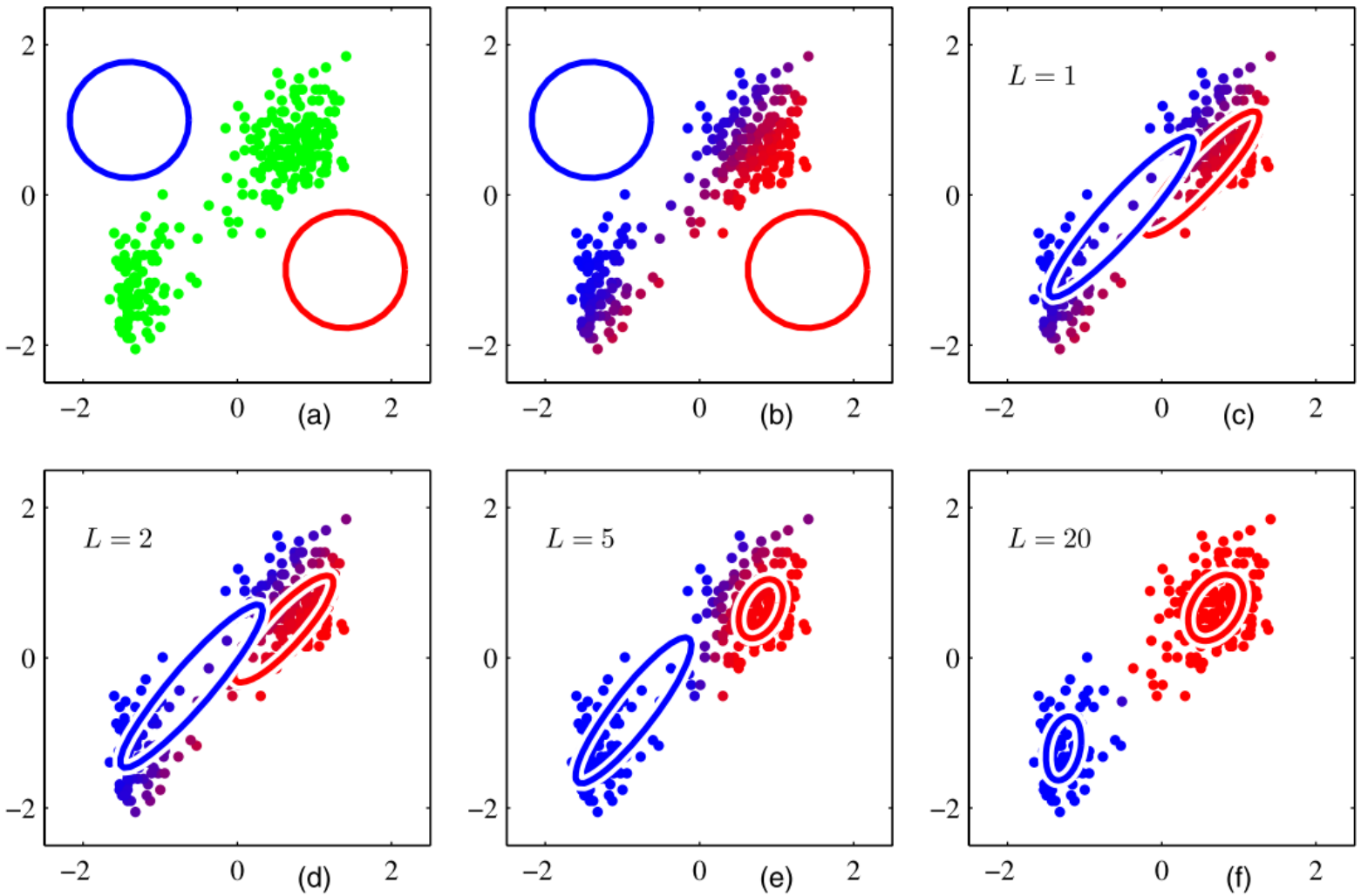
$$\log p(X, Z|\theta)$$

  is easy to maximize.
- Problem: Z is not observed
- Solution: maximize

$$\begin{aligned} Q(\theta, \theta_0) &\equiv E_{Z|X,\theta_0}\left[\log p(X, Z|\theta)\right] \\ &= \sum_Z p(Z|X, \theta_0) \log p(X, Z|\theta) \end{aligned}$$

  where $p(Z|X, \theta_0)$ is the posterior distribution of the latent variables computed using the current parameter estimate θ_0

Illustration of the EM algorithm for GMMs



EM algorithm in detail

Goal: maximize $\log p(X|\theta)$ w.r.t. θ

1. Initialize θ_0
2. **E-step** Evaluate $p(Z|X, \theta_0)$, and then compute

$$Q(\theta, \theta_0) = E_{Z|X,\theta_0}\left[\log p(X, Z|\theta)\right] = \sum_Z p(Z|X, \theta_0) \log p(X, Z|\theta)$$

3. **M-step** Evaluate θ^{new} using

$$\theta^{new} = \arg\max_\theta Q(\theta, \theta_0).$$

   Set $\theta_0 \leftarrow \theta^{new}$
4. Repeat **E** and **M** steps until convergence

Why EM works

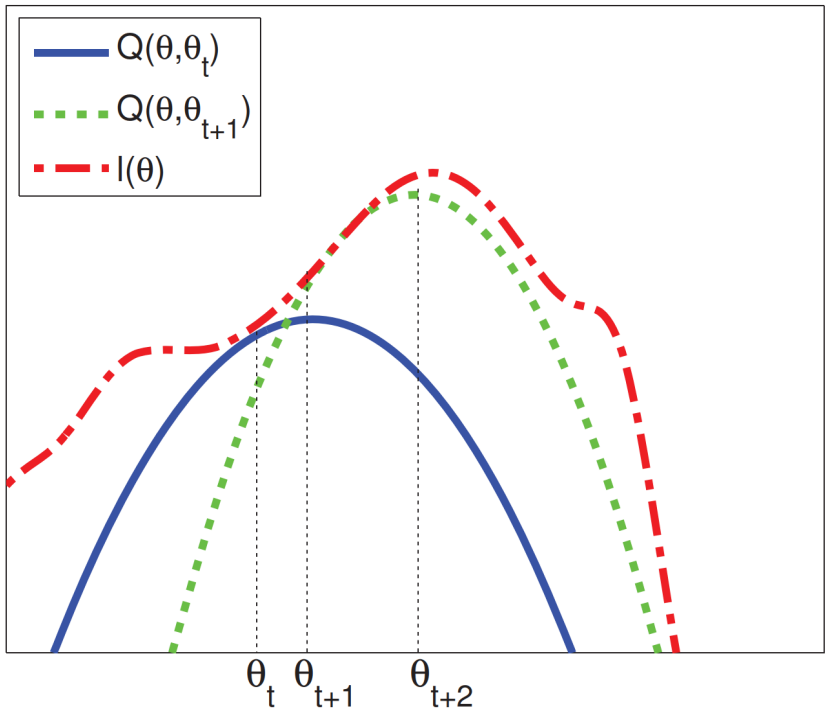


Figure: 11.16 in Murphy (2012)

- $Q(\theta, \theta_0)$ is a lower bound of the log-likelihood $\log p(x|\theta)$ (Bishop, Ch. 9.4)
- EM iterates between 1) updating the lower bound (E-step), 2) maximizing the lower bound (M-step).

EM algorithm, comments

- In general, Z does not have to be discrete, just replace the summation in $Q(\theta, \theta_0)$ by integration.
- EM-algorithm can be used to compute the MAP (*maximum a posteriori*) estimate by maximizing in the M-step $Q(\theta, \theta_0) + \log p(\theta)$.
- In general, EM-algorithm is applicable when the observed data X can be **augmented** into complete data $\{X, Z\}$ such that $\log p(X, Z|\theta)$ is easy to maximize; Z does not have to be latent variables but can represent, for example, unobserved values of missing or censored observations.

EM algorithm, simple example

- Consider N independent observations $\mathbf{x} = (x_1, \ldots, x_N)$ from a two-component mixture of univariate Gaussians

$$p(x_n|\theta) = \frac{1}{2}N(x_n|0,1) + \frac{1}{2}N(x_n|\theta,1). \quad (1)$$

- One unknown parameter, θ, the mean of the second component.
- **Goal**: estimate

$$\widehat{\theta} = \arg\max_{\theta} \{\log p(\mathbf{x}|\theta)\}.$$

- *simple_example.pdf* and *simple_em.m*

EM algorithm for GMMs

$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k N(\mathbf{x}|\mu_k, \Sigma_k)$

1. Initialize parameter μ_k, Σ_k and mizing coefficients π_k. Repeat until convergence:
2. **E-step**: Evaluate the responsibilities using current parameter values

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

3. **M-step:** Re-estimate the parameters using the current responsibilities

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \mu_k^{new})(\mathbf{x}_n - \mu_k^{new})^T$$

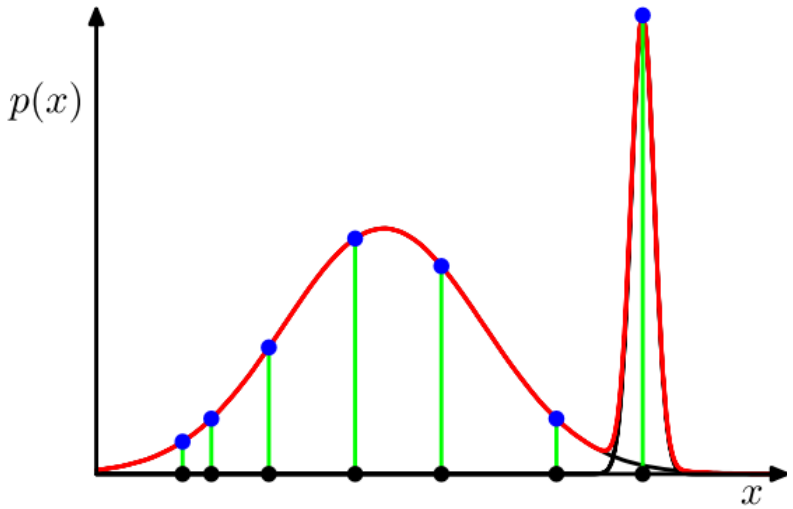
$$\pi_k^{new} = \frac{N_k}{N}$$

Derivation of the EM algorithm for GMMs

- In the **M-step** the formulas for μ_k^{new} and Σ_k^{new} are obtained by differentiating the expected complete data log-likelihood $Q(\theta, \theta_0)$ with respect to the particular parameters, and setting the derivatives to zero.
- The formula for π_k^{new} can be derived by maximizing $Q(\theta, \theta_0)$ under the constraint $\sum_{k=}^{K} \pi_k = 1$. This can be done using the *Lagrange multipliers*.

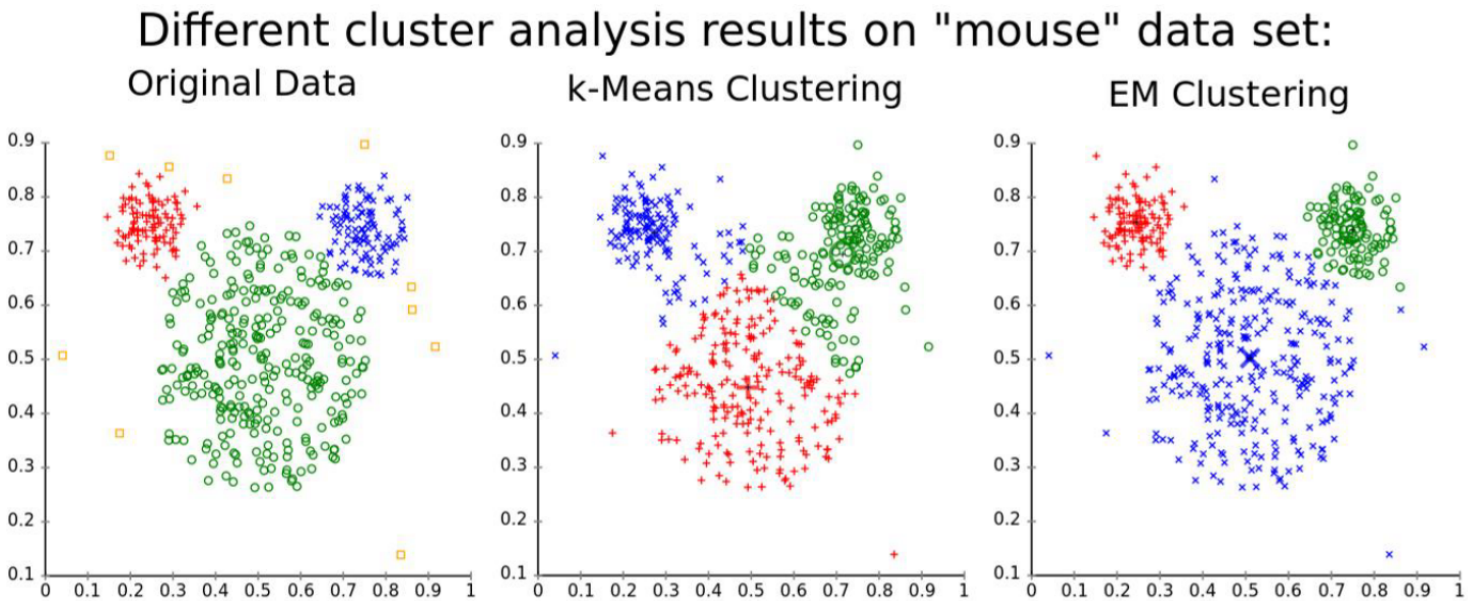
EM for GMM, caveats

- EM converges to a local optimum. In fact, the ML estimation for GMMs is not well-defined due to **singularities**: if $\sigma_k \longrightarrow 0$ for components k with a single data point, likelihood goes to infinity (fig). Remedy: prior on σ_k.
- **Label-switching**: non-identifiability due to the fact that cluster labels can be switched and likelihood remains the same.
- In practice it is recommended to initialize the EM for the GMM by k-means.



GMM vs. k-means

- "Why use GMMs and not just k-means?"



from Wikipedia

1. Clusters can be of different sizes and shapes
2. Probabilistic assignment of data items to clusters
3. Possibility to include prior knowledge (structure of the model/prior distributions on the parameters)

Important points

- Definition of the Gaussian mixture model
- Representing the Gaussian mixture model using discrete latent variables, which specify the components (or clusters) of the observations
- ML-estimation of GMMs can be done using numerical optimization or the EM algorithm.
- The main idea of the EM algorithm is to maximize the expectation of the complete data log-likelihood, where the expectation is computed over the current posterior distribution of the latent variables.