# INF367A: Probabilistic machine learning

## Lecture 9: The EM algorithm and Gaussian mixture models

Pekka Parviainen

University of Bergen

3.3.2020

# Outline

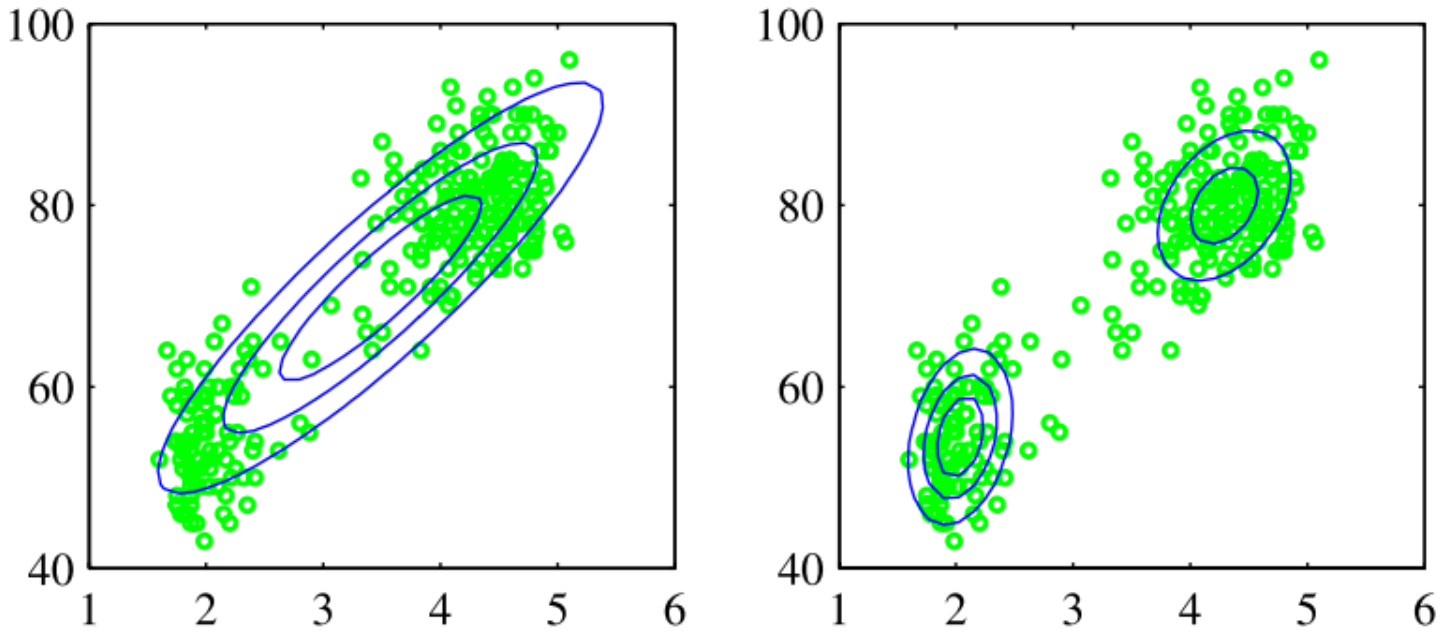


# EM algorithm

- Finding MAP or ML solutions when not all relevant data is observed
- Applicable in many different settings
- Departure from Bayesian framework
  - Used to find point estimates
  - We derive the algorithm for maximum likelihood estimation because the formulas are simpler but the algorithm can be used also for MAP estimates

# Example: Gaussian mixture models

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

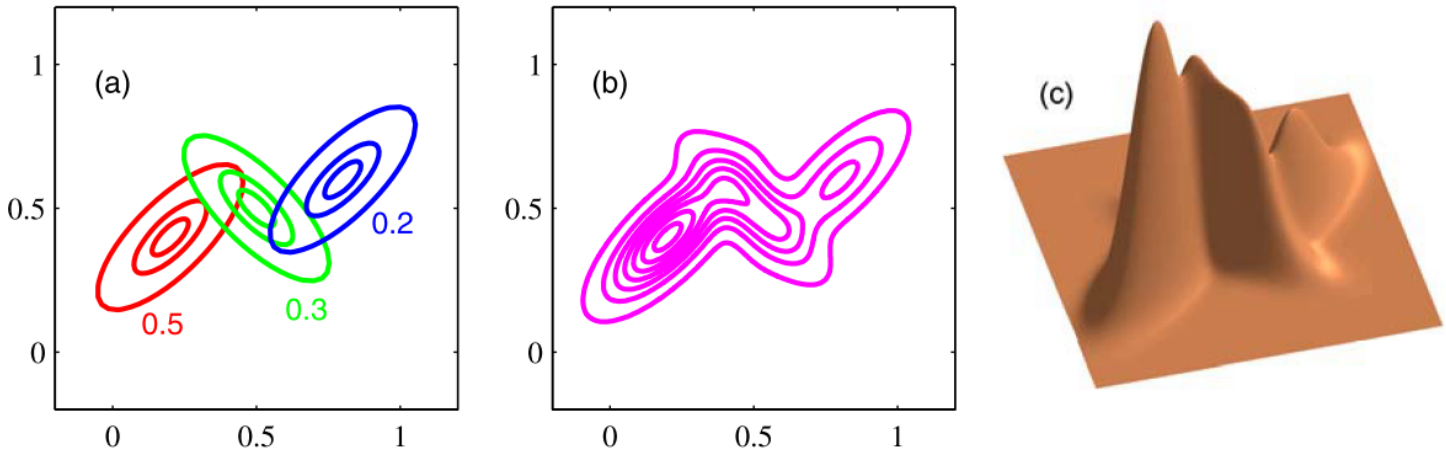



# Gaussian mixture models

- 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 $\mu_k$ and covariance $\Sigma_k$.
- $\pi_k$ are the **mixing coefficients**, which satisfy $\sum_k \pi_k = 1$, $0 \leq \pi_k \leq 1$.



# Gaussian mixture models

- A GMM has a probability density

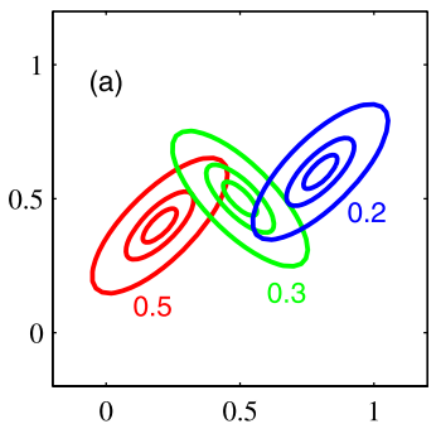
$$P(\mathbf{x}) = \sum_{k=1}^{k} \pi_k N(\mathbf{x}|\mu_k, \Sigma_k)$$

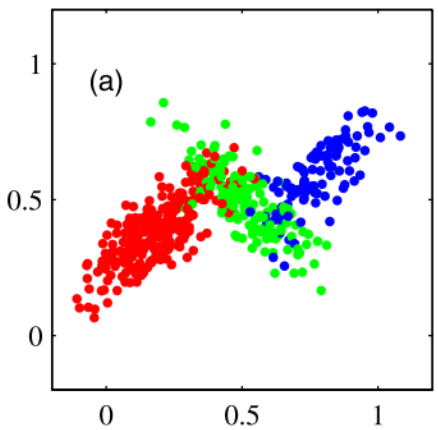
- Two questions:
  1. Given a data point $\mathbf{x}$, what is the probability that it belongs to the cluster $k$?
  2. In general, $\pi$, $\mu$, and $\Sigma$ are not known. How to estimate them given data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$?

# 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

$$P(z_k = 1) = \pi_k \qquad \text{and} \qquad P(\mathbf{x}\,|\,z_k = 1) = N(\mathbf{x}\,|\,\mu_k, \Sigma_k),$$

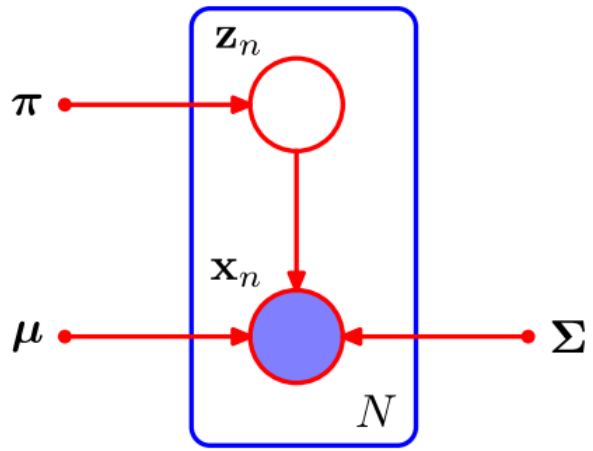
or equivalently

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

- Then

$$P(\mathbf{x}) = \sum_{\mathbf{z}} P(\mathbf{z})P(\mathbf{x}\,|\,\mathbf{z}) = \sum_{k} \pi_k N(\mathbf{x}\,|\,\mu_k, \Sigma_k)$$

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




# GMM: responsibilities (1/2)

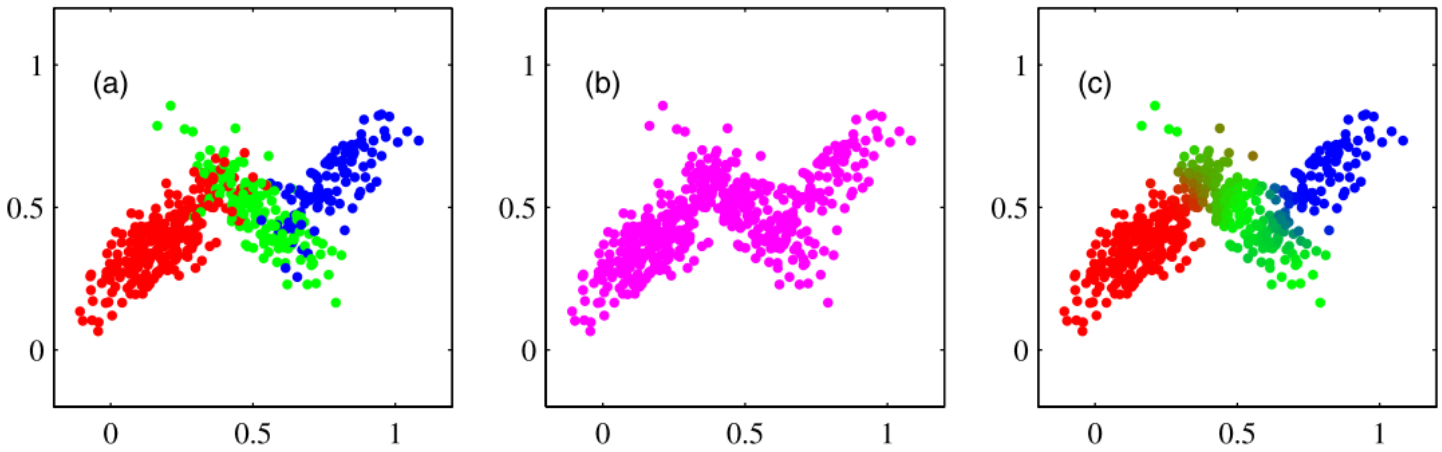
- Posterior probability $P(z_{nk} = 1|\mathbf{x}_n)$ that observation $\mathbf{x}_n$ was generated by component $k$ given $\theta = (\pi, \mu, \Sigma)$

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

- $\gamma(z_{nk})$ 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 $\pi$, $\mu$, and $\Sigma$ are usually *unknown*.



# Idea of the EM algorithm

- Three types of variables: Data $\mathbf{x}$ (known), latent variables $\mathbf{z}$ (unknown) and parameters $\theta$ (unknown)
- Likelihood of complete data

$$P(\mathbf{x}, \mathbf{z} \mid \theta) = P(\mathbf{x} \mid \mathbf{z}, \theta) P(\mathbf{z} \mid \theta)$$

- If we knew $\mathbf{z}$ (cluster assigments) then it would be easy to find maximum likelihood parameters for $\theta$
  - Compute sample means and covariances for each cluster
- If we knew parameters $\theta$ then we could compute posterior probabilities for cluster assignments $\mathbf{z}$
  - Responsibilities
- Idea: alternate between computing responsibilities for $\mathbf{z}$ (E-step) and maximizing (log-)likelihood (M-step)
  - Hence, we have an Expectation-Maximization (EM) algorithm

# Derivation of the EM algorithm

- $\mathbf{x}$: **observed** data, $\mathbf{z}$: **unobserved** latent variables
- $\{\mathbf{x}, \mathbf{z}\}$: **complete** data, $\mathbf{x}$: **incomplete** data
- Goal: maximize the incomplete data log-likelihood

$$\hat{\theta} = \arg\max_{\theta} \Big\{ \log P(\mathbf{x}|\theta) \Big\}$$

- If there are latent variables the incomplete data log-likelihood is given by

$$\log P(\mathbf{x}|\theta) = \log \Big\{ \sum_{\mathbf{z}} P(\mathbf{x}, \mathbf{z}|\theta) \Big\},$$

where $P(\mathbf{x}, \mathbf{z}|\theta)$ is the complete data likelihood

# Derivation of the EM algorithm

- Assume that the complete data log-likelihood

$$\log P(\mathbf{x}, \mathbf{z} | \theta)$$

  is easy to maximize
- Problem: $\mathbf{z}$ is not observed
- Solution: Maximize

$$\begin{aligned} Q(\theta, \theta_t) &= E_{\mathbf{z}|\mathbf{x},\theta_t}[\log P(\mathbf{x}, \mathbf{z} | \theta)] \\ &= \sum_{\mathbf{z}} P(\mathbf{z} | \mathbf{x}, \theta_t) \log P(\mathbf{x}, \mathbf{z} | \theta) \end{aligned}$$

  where $P(\mathbf{z} | \mathbf{x}, \theta_t)$ is the posterior distribution of the latent variables computed using the current parameter estimate $\theta_t$

# EM algorithm

**Goal**: maximize $\log P(\mathbf{x}|\theta)$ w.r.t. $\theta$

1. Initialize $\theta_0$
2. **E-step** Evaluate $P(\mathbf{z}|\mathbf{x}, \theta_t)$, and then compute

$$Q(\theta, \theta_t) = E_{\mathbf{z}|\mathbf{x},\theta_t}[\log P(\mathbf{x}, \mathbf{z}|\theta)] = \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x}, \theta_t) \log P(\mathbf{x}, \mathbf{z}|\theta)$$

3. **M-step** Find $\theta_{t+1}$ using

$$\theta_{t+1} = \arg\max_{\theta} Q(\theta, \theta_t)$$

4. Repeat **E** and **M** steps until convergence

# EM algorithm, a simple example

simple_example.pdf

# EM algorithm for GMMs

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

1. Initialize parameter $\mu_k$, $\Sigma_k$ and mixing coefficients $\pi_k$. 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

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

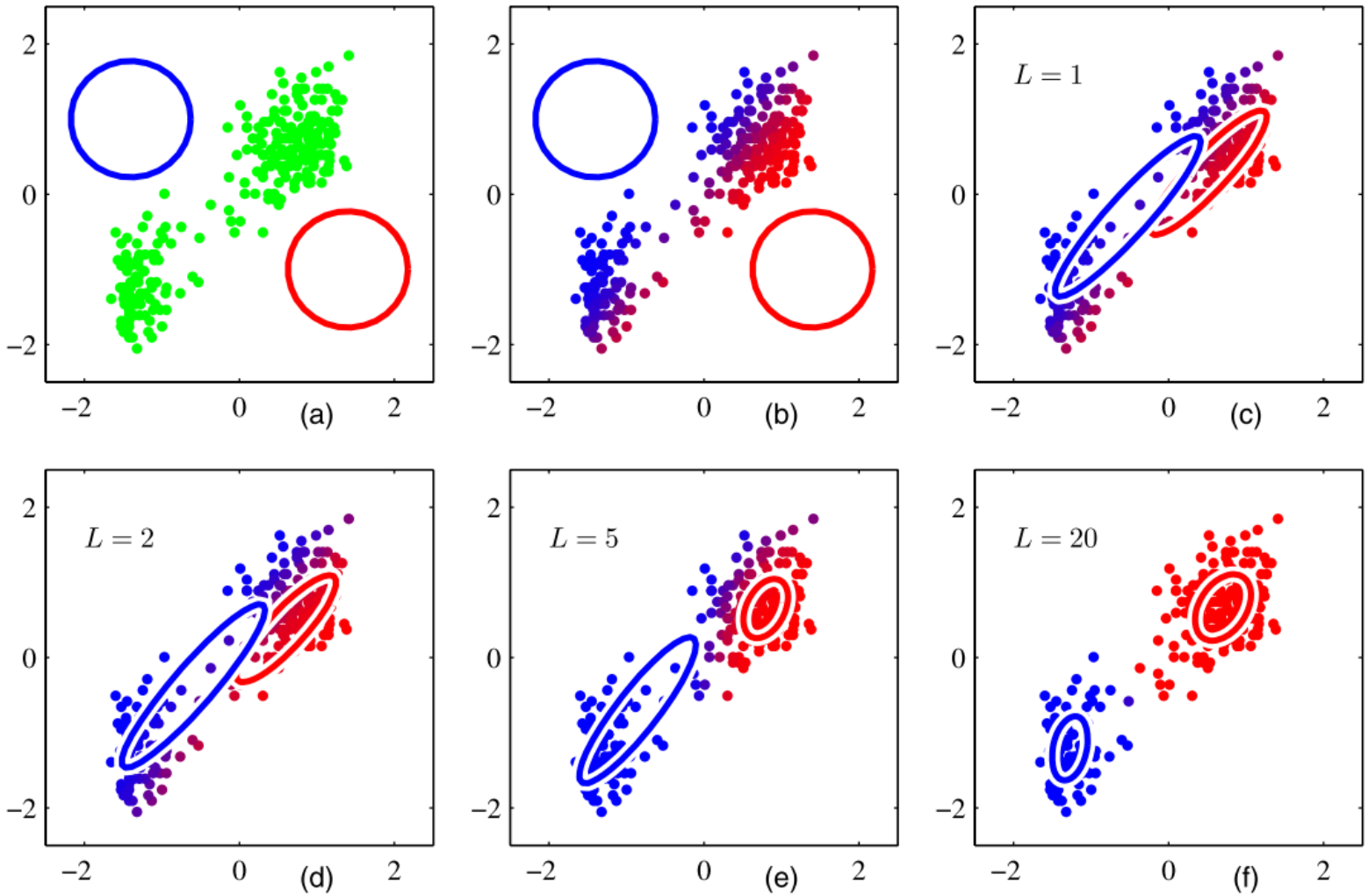
$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \mu_k^{new})(\mathbf{x}_n - \mu_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}, \text{ where } N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

# Derivation of the EM algorithm for GMMs

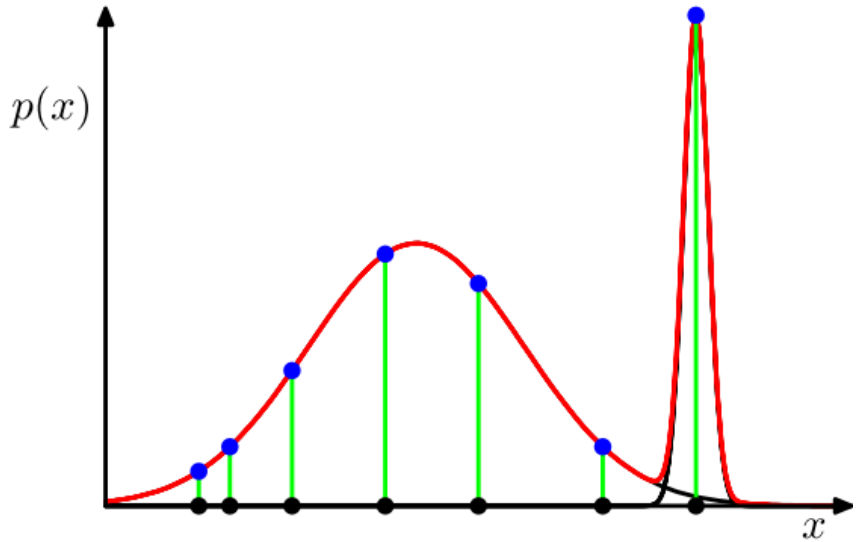
- In the **M-step** the formulas for $\mu_k^{new}$ and $\Sigma_k^{new}$ are obtained by differentiating the expected complete data log-likelihood $Q(\theta, \theta_t)$ with respect to the particular parameters, and setting the derivatives to zero.
- The formula for $\pi_k^{new}$ can be derived by maximizing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

# Illustration of the EM algorithm for GMMs



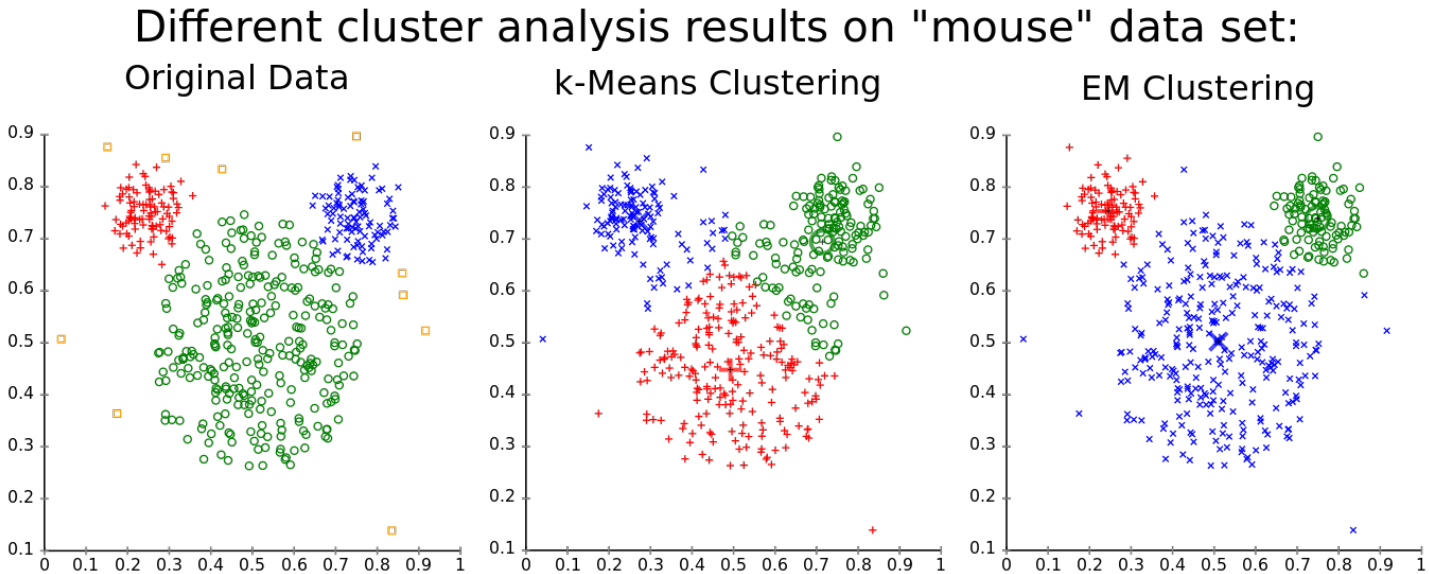

# EM for GMM, caveats

- EM converges to a local optimum. In fact, the ML estimation for GMMs is an ill-posed problem due to **singularities**: if $\sigma_k \to 0$ for components $k$ with a single data point, likelihood goes to infinity (fig). Remedy: prior on $\sigma_k$.
- **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 (1/2)

- "Why use GMMs in the first place 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)

# EM algorithm, comments

- In general, $\mathbf{z}$ does not have to be discrete, just replace the summation in $Q(\theta, \theta_t)$ by integration.
- EM algorithm can be used to compute the MAP (*maximum a posteriori*) estimate by maximizing in the M-step $Q(\theta, \theta_t) + \log P(\theta)$.
- EM algorithm can be applied more generally in situations where the observed data $x$ can be **augmented** into complete data $\{\mathbf{x}, \mathbf{z}\}$ such that $\log P(\mathbf{x}, \mathbf{z} \mid \theta)$ is easy to maximize; that is, $\mathbf{z}$ does not have to be latent variables but can represent, for example, the unobserved values of missing or censored observations.
- EM algorithm converges into a local optimum. Each iteration increases the log-likelihood of observed data $\log P(\mathbf{x} \mid \theta_t)$.
  - You can use the log-likelihood of the observed data as a sanity check

# Further readings

- Bishop 9