Introduction to Machine Learning
Topic 6: Gaussian processes and mixture models
Lecture 6b

Mattias Villani

Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University

Topic overview

- **K-means clustering**
- **Mixture models**
- **EM algorithm** for mixtures.

k-Means clustering

- Data in $\mathbb{R}^D$: $\mathbf{x}_1, ..., \mathbf{x}_N$.
- Aim: **partition the data into K clusters**.
- Each observation $\mathbf{x}_n$ is represented by a **centroid** $\mu_k \in \mathbb{R}^D$.
- Let $r_{nk} = 1$ if $\mathbf{x}_n$ belongs to μ_k.
- **k-means clustering** minimizes

$$\sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| \mathbf{x}_n - \mu_k \right\|^2$$

  with respect to the r_{nk} and the $\mu_1, ..., \mu_K$.
- Iterative **algorithm**: initialize $\mu_1, ..., \mu_K$, then:
  - Allocate ($r_{nk} = 1$) each observations to nearest centroid
  - Recompute centroids as means of allocated observations.

k-Means clustering of old faithful data

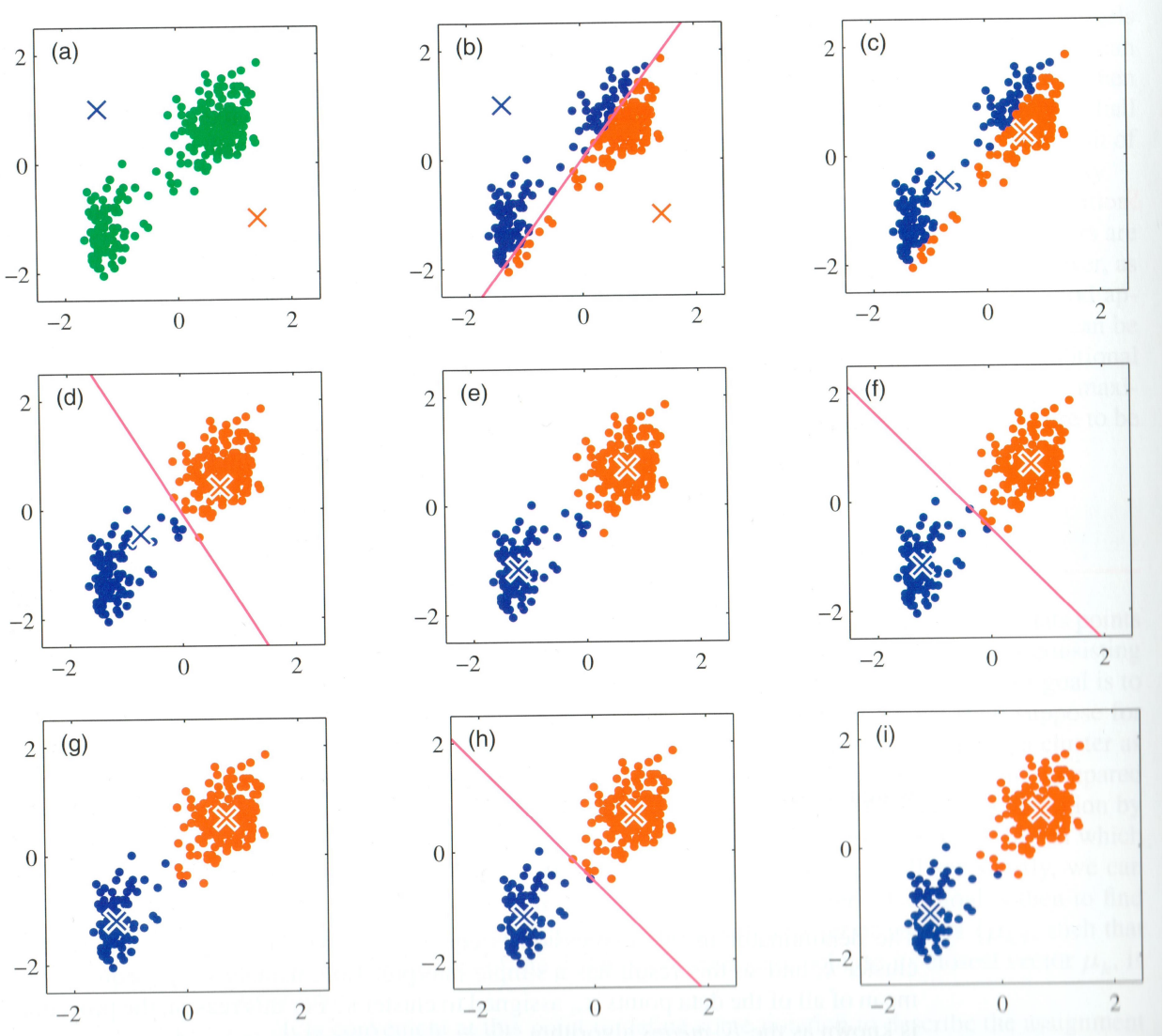
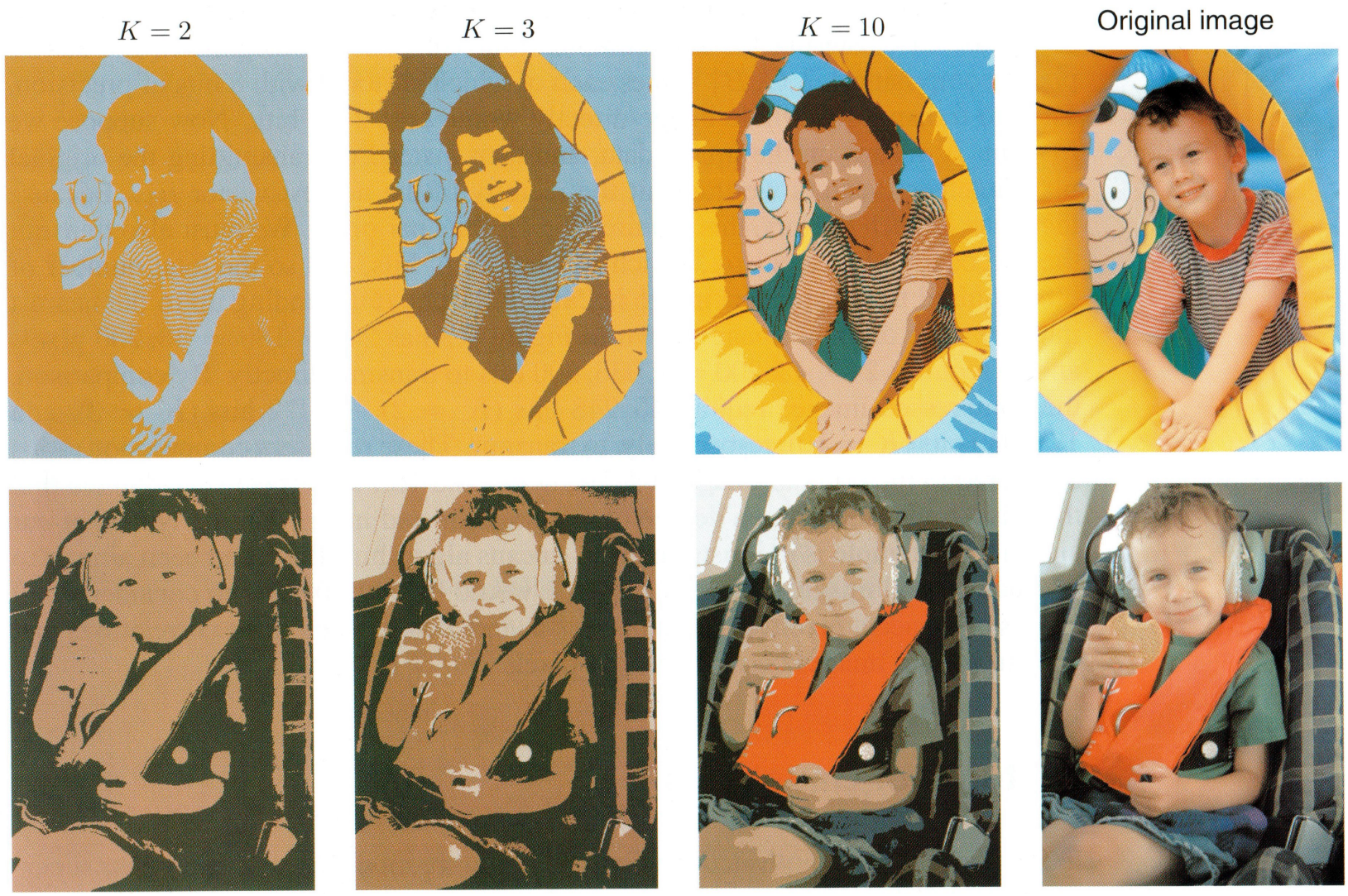


Image compression by k-means



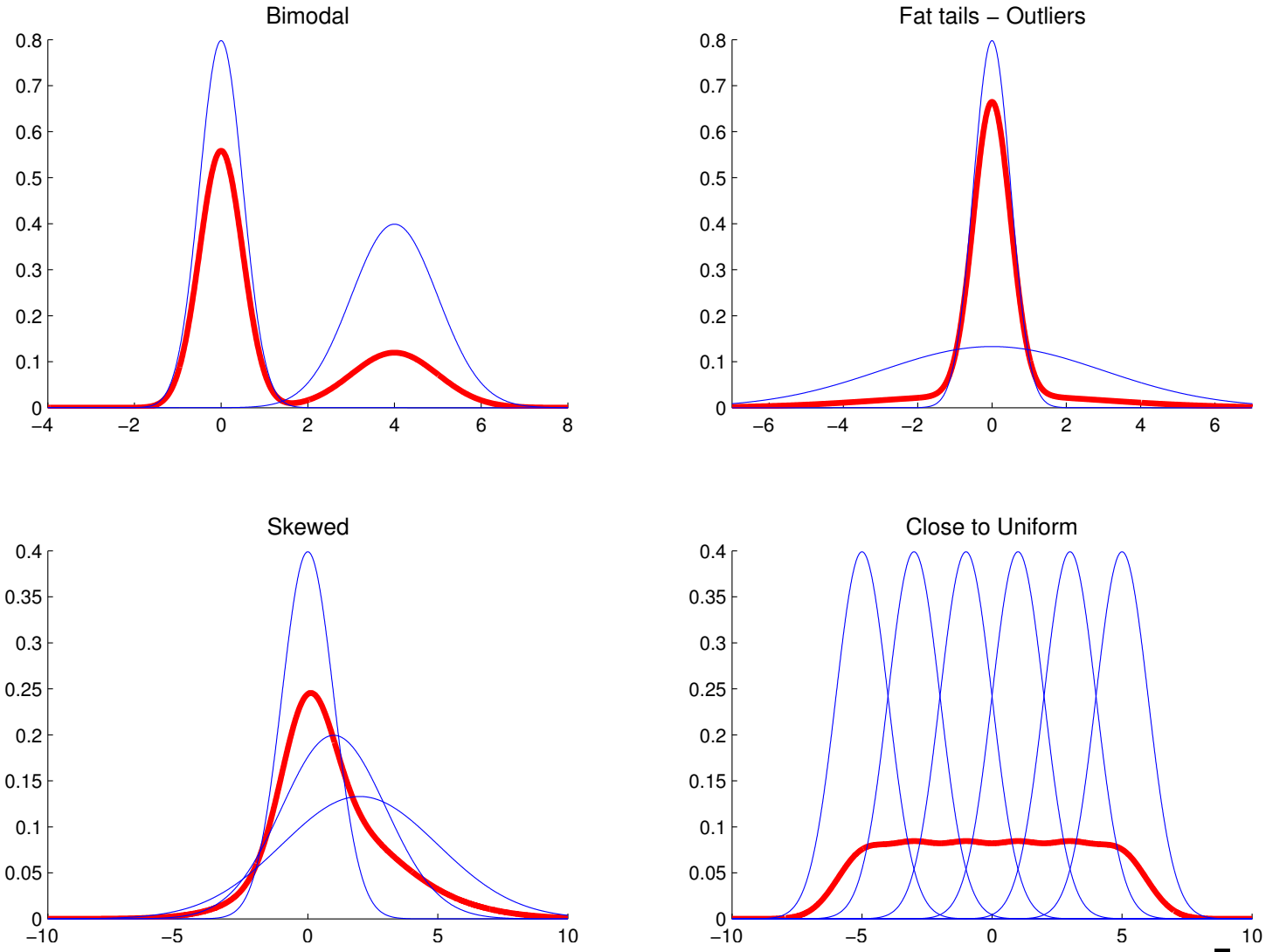
Mixtures of Gaussians

- Let $\mathcal{N}(x|\mu, \sigma^2)$ denotes the **PDF** of a **normal** variable $X \sim N(\mu, \sigma^2)$.
- **Two-component mixture of normals** [MN(2)]

$$p(x) = \pi \cdot \mathcal{N}(x|\mu_1, \sigma_1^2) + (1 - \pi) \cdot \mathcal{N}(x|\mu_2, \sigma_2^2)$$

- **Simulate** from a MN(2):
  - Simulate an indicator $I \in \{1, 2\}$, where $I \sim Bern(\pi)$, and $\pi = \Pr(I = 1)$.
  - If $I = 1$, simulate x from $N(\mu_1, \sigma_1^2)$
  - If $I = 2$, simulate x from $N(\mu_2, \sigma_2^2)$.

Illustration of mixture distributions



Mixture distributions, cont.

- K-component mixture of normals

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2), \quad \sum_{k=1}^{K} \pi_k = 1$$

- K-component mixture of multivariate normals

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

- General mixtures ($f(x|\theta_k)$ can be any distribution)

$$p(x) = \sum_{k=1}^{K} \pi_k f(x|\theta_k).$$

- Example: **count data**. Poisson is elegant, but restrictive ($E(X) = Var(X)$). A **mixture of Poissons** is more flexible:

$$p(x) = \sum_{k=1}^{K} \pi_k \cdot \mathrm{Poisson}(x|\theta_k).$$

Mixture distributions, cont.

- Not easy to estimate directly - the likelihood is a product of sums

$$p(x_1, ..., x_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2).$$

- **Assume** that we knew which of the component density each observation came from:

$$I_i = \begin{cases} 1 \text{ if } x_i \text{ came from Density 1} \\ 2 \text{ if } x_i \text{ came from Density 2} \\ \vdots \\ K \text{ if } x_i \text{ came from Density K} \end{cases}.$$

- If we know $I_1, ..., I_n$: just estimate $\pi_1, ..., \pi_K$, $\mu_1, ..., \mu_K, \sigma_1^2, ..., \sigma_K^2$ by separating the sample according to the I's.
- But we do **not** know $I_1, ..., I_n$! [**Gibbs sampling**; Bayesian Learning].
- **EM-algorithm** is 'poor man's Gibbs sampling'. Maximum likelihood estimators.

EM-algorithm for mixture of Gaussians

- Define z_i to be a **1-of-K variable** that indicates which mixture component observation i belongs to.
- Observation i belongs to component $k \Rightarrow$ the kth element of z_i is unity and the other elements are all zero.
- Let $\mathbf{Z}$ be the $N \times K$ **component membership matrix** where the ith row is z_i. Latent (unobserved variables).
- Complete data likelihood: $p(x_1, ..., x_N, \mathbf{Z}|\theta)$. But $\mathbf{Z}$ is not observed.
- **Expectations-Maximization (EM) algorithm.** **Maximum likelihood** for models with latent variables.
- Iterative method. Parameters are updated and the likelihood increases at every iteration.

EM-algorithm for mixture of Gaussians

1. Initialize the π's, μ's and σ^2s. Compute the log likelihood.
2. (**E-step**) Given the component parameters (the π's, μ's and σ^2's) compute the expected values for the latents z_{nk}

$$\gamma\left(z_{nk}\right) = \frac{\pi_k \mathcal{N}\left(x_n | \mu_k, \sigma_k^2\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(x_n | \mu_j, \sigma_j^2\right)}$$

   and $N_k = \sum_{n=1}^{N} \gamma\left(z_{nk}\right)$.
3. (**M-step**) Given the $\gamma(z_{nk})$, maximize the expected (with respect to $\mathbf{Z}$) log-likelihood for the component parameters

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma\left(z_{nk}\right) x_n$$

$$\sigma_k^2 = \frac{1}{N_k} \sum_{n=1}^{N} \gamma\left(z_{nk}\right) \left(x_n - \mu_k^{new}\right)^2$$

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

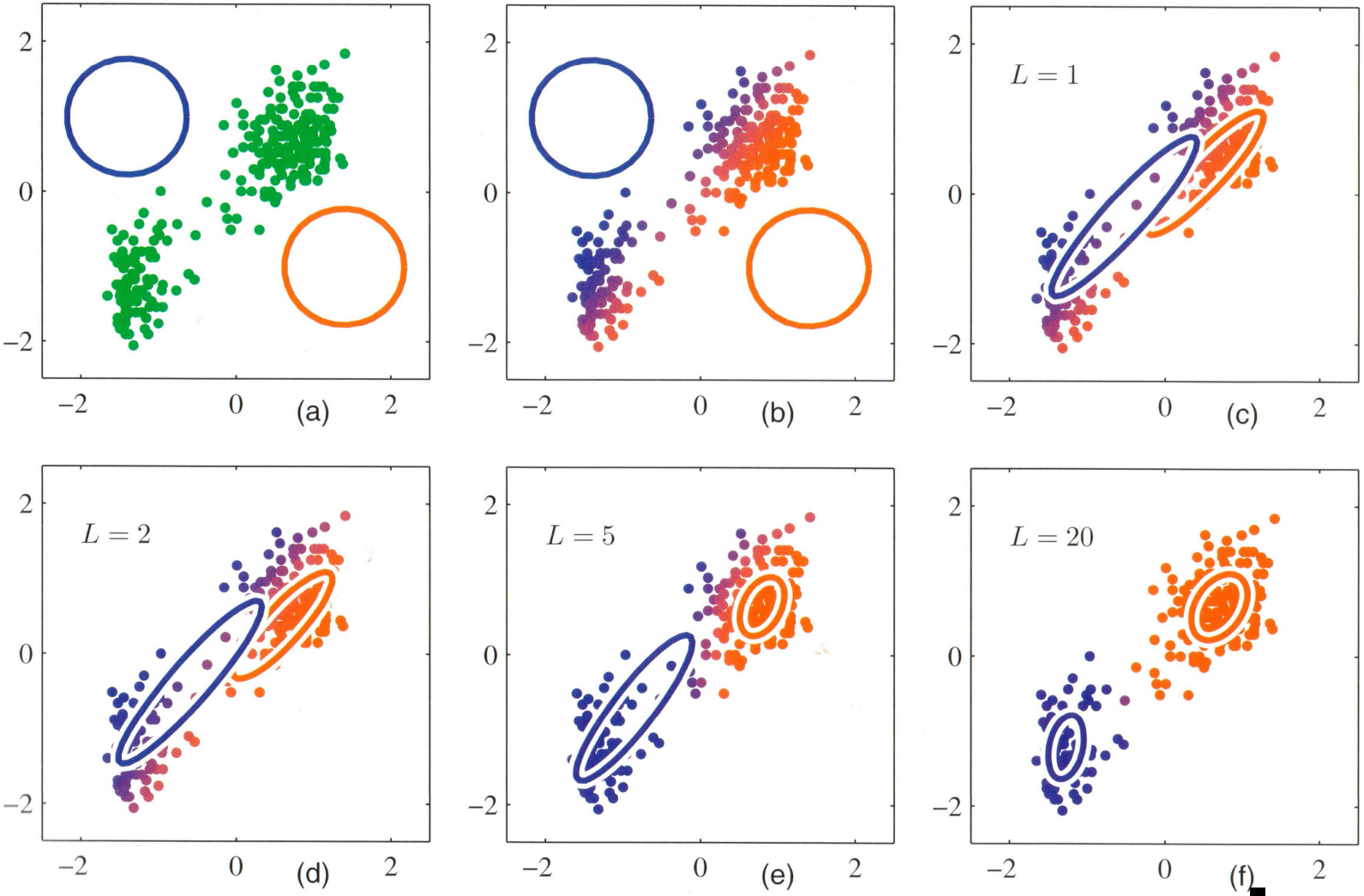
EM-algorithm for mixture of Gaussians

4. Evaluate the log-likelihood at the new parameter values

$$\log p\left(x_1, ..., x_N | \mu, \sigma^2, \pi\right) = \log \left\{ \sum_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \sigma_k^2) \right\}$$

and check for convergence of the log-likelihood.

Mixture of bivariate Gaussians for old Faithful



EM-algorithm for mixture of Gaussians

- EM: the log-likelihood is **guaranteed to increase** at every iteration.
- EM is typically **slow**. Switch to Newton-Raphson when you come closer to the maximum.
- EM solution often depends on the **initial values**. Restart EM with many different initial values and pick the best solution.
- **Label-switching** for mixtures. There are actually $K!$ maxima.
- EM tends to find a finite maximum even though the **likelihood is actually unbounded**. Singularities. Likelihood becomes arbitraly large if kth component is $\mathcal{N}(x_j, \sigma_k^2)$ with $\sigma_k^2 \to 0$.