

Machine Learning

Lecture 9 - Unsupervised learning, mixture models and clustering

Mattias Villani

Department of Statistics
Stockholm University

Department of Computer and Information Science
Linköping University



mattiasvillani.com



@matvil



mattiasvillani

Lecture overview

- Supervised Mixture-of-Normals
- Unsupervised Mixture-of-Normals
- k-means clustering

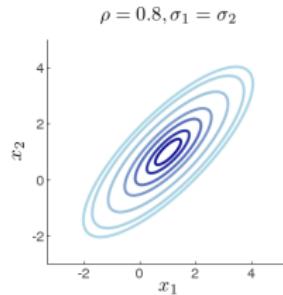
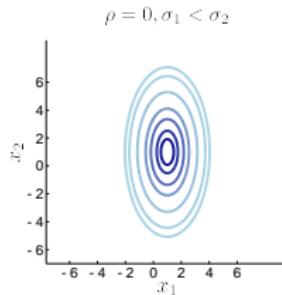
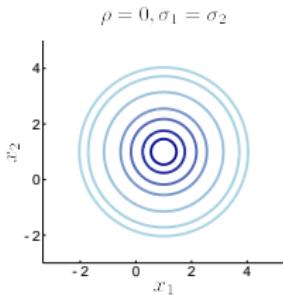
Multivariate normal distribution

- $x \in \mathbb{R}^p$ is a **multivariate normal**, $x \sim \mathcal{N}(\mu, \Sigma)$, with **density**

$$p(x) = |2\pi\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right)$$

- **Mean** and **covariance** matrix

$$\mathbb{E}(x) = \mu \quad \text{and} \quad \text{Cov}(x) = \Sigma$$



Multivariate normal distribution - properties

- $x \in \mathbb{R}^p$ is **multivariate normal**, $x \sim \mathcal{N}(\mu, \Sigma)$ with

$$\mathbb{E}(x) = \mu \text{ and } \text{Cov}(x) = \Sigma$$

- Decompose

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \text{ and } \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

- **Marginal distributions** are normal

$$x_1 \sim \mathcal{N}(\mu_1, \Sigma_{11})$$

- **Conditional distributions** are normal

$$x_1 | x_2 \sim \mathcal{N}(\tilde{\mu}_1, \tilde{\Sigma}_1)$$

with

$$\tilde{\mu}_1 = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$$

$$\tilde{\Sigma}_1 = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

Discriminative vs Generative models

■ **Discriminative models:** modeling of $p(y|x)$. No model for x .

- ▶ Logistic regression
- ▶ Regression trees and ensembles
- ▶ Deep neural networks

■ **Generative models:** *joint* distribution of labels and features

$$p(y, x) = p(x|y)p(y).$$

- ▶ Discriminant analysis
- ▶ Mixture models

■ Generative models require more effort:

- ▶ **need also a model for x**
- ▶ features may be high-dimensional. Hard!

■ Generative models:

- ▶ **better understanding** of the mechanisms.
- ▶ can be extended to **unsupervised** (all labels missing) and **semi-supervised** (some labels missing).

Gaussian mixture model - supervised case

- Let $\mathcal{N}(x|\mu, \Sigma)$ denote the density function of $x \sim \mathcal{N}(\mu, \Sigma)$
- Joint model for label $y \in \{1, \dots, M\}$ and features $x \in \mathbb{R}^p$:

$$p(y, x) = p(x|y)p(y)$$

where the **class-conditional densities** are normal:

$$p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$$

- Known labels: **MLE for each class separately**

$$\hat{\mu}_m = \frac{1}{n_m} \sum_{i:y_i=m} x_i$$

$$\hat{\Sigma}_m = \frac{1}{n_m} \sum_{i:y_i=m} (x_i - \hat{\mu}_m)(x_i - \hat{\mu}_m)^\top$$

- MLE for discrete labels** $p(y = m) = p_m$

$$\hat{p}_m = \frac{n_m}{n}.$$

Discriminant analysis

- Classification probability by Bayes' theorem

$$p(y|x) = \frac{p(y,x)}{p(x)} = \frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y)$$

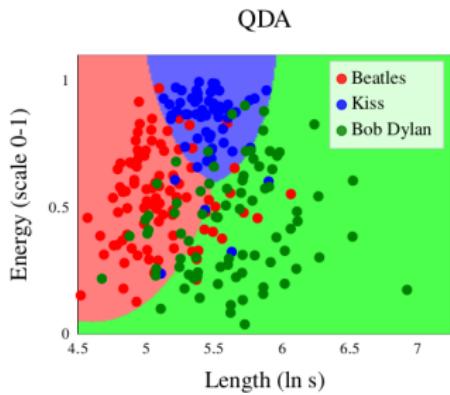
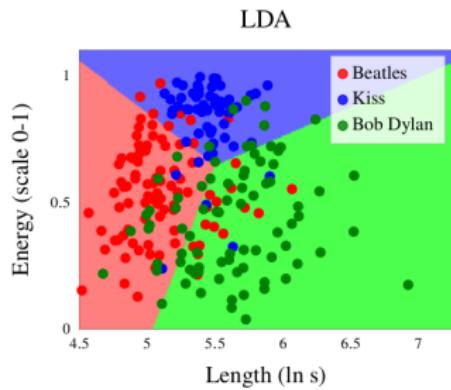
- Quadratic discriminant analysis (QDA)

$$\begin{aligned} & \arg \max_m (\log \hat{p}(x_\star | y = m) + \log \hat{p}(y = m)) \\ &= \arg \max_m (\log \mathcal{N}(x_\star | \hat{\mu}_m, \hat{\Sigma}_m) + \log \hat{p}_m) \\ &= \arg \max_m \left(-\frac{1}{2} \log |\hat{\Sigma}_m| - \frac{1}{2} (x_\star - \hat{\mu}_m)^\top \hat{\Sigma}_m^{-1} (x_\star - \hat{\mu}_m) + \log \hat{p}_m \right) \end{aligned}$$

- So QDA has a quadratic decision boundary.
- Linear discriminant analysis (LDA) if we assume

$$\Sigma_1 = \Sigma_2 = \dots = \Sigma_M = \Sigma$$

Discriminant analysis



Gaussian mixture model - unsupervised case

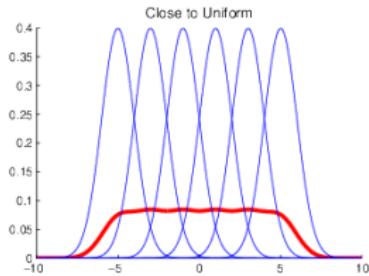
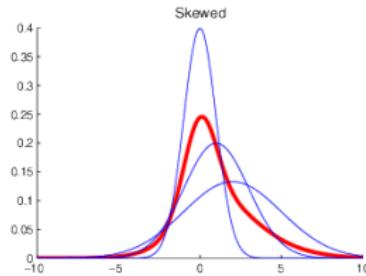
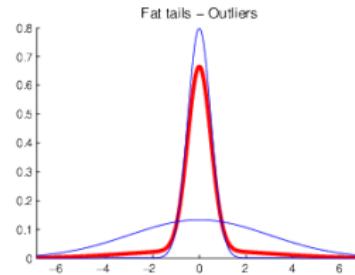
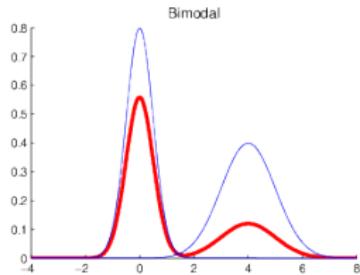
- **Unsupervised**: no labels.
- Don't know from which distribution $\mathcal{N}(x|\mu_m, \Sigma_m)$ each observation x_i comes from.
- **Two-component univariate mixture of normals**

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

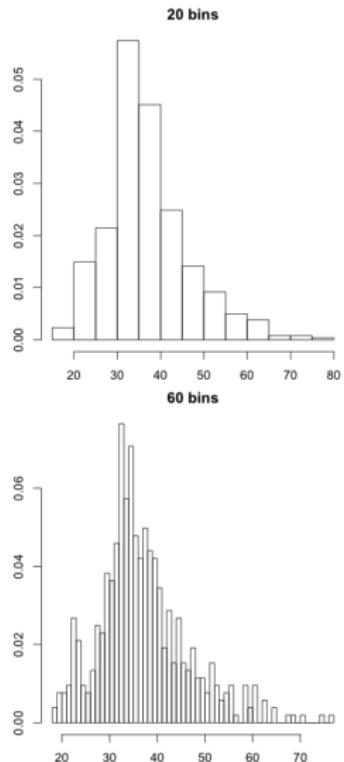
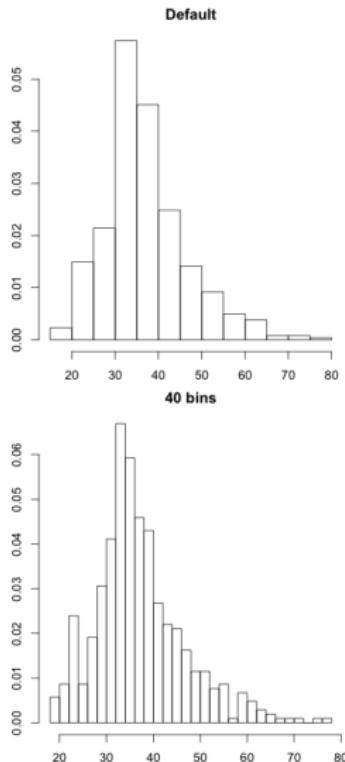
- **Simulate** from a two-component mixture of normals:
 - ▶ Simulate $y_i \in \{1, 2\}$, with $\Pr(y_i = 1) = \pi$.
 - ▶ If $y_i = 1$, simulate x from $N(\mu_1, \sigma_1^2)$
 - ▶ If $y_i = 2$, simulate x from $N(\mu_2, \sigma_2^2)$.
- **M-component mixture of multivariate normals**

$$p(x) = \sum_{m=1}^M \pi_m \mathcal{N}(x|\mu_m, \Sigma_m), \quad \sum_{m=1}^M \pi_m = 1$$

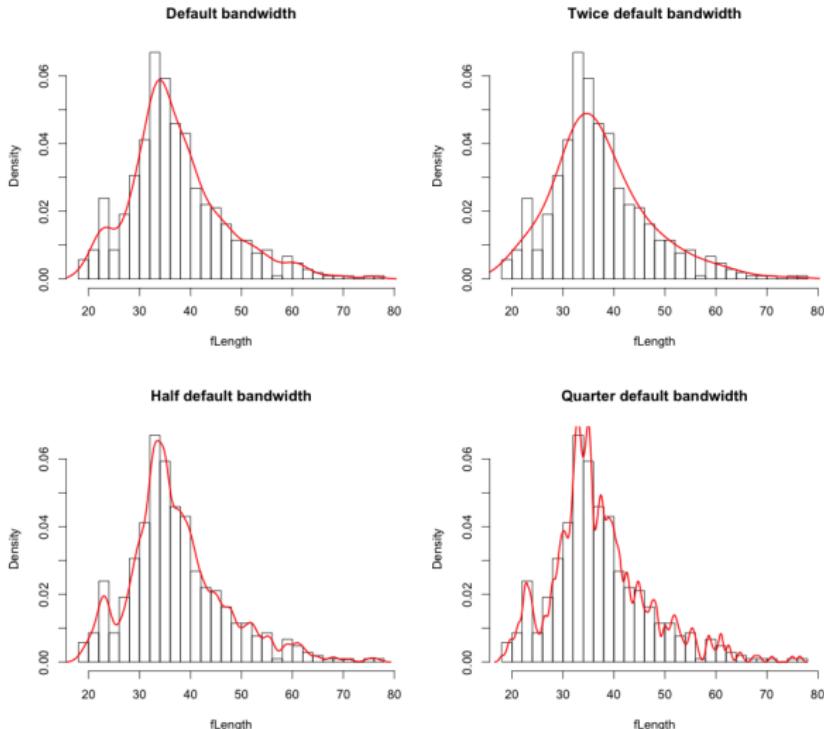
Illustration of mixture distributions



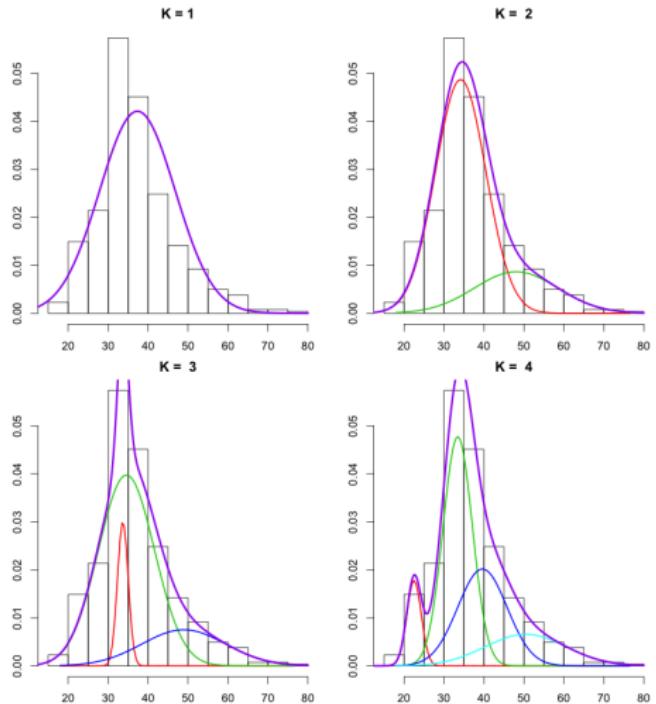
Fish length - Histogram density estimates



Fish length - Kernel density estimates



Fish length - Mixture of normals



- See code GMM_EM.R on web page.

EM algorithms for unsupervised Gaussian mixtures

- GMM - the likelihood is a messy product of sums

$$p(x_1, \dots, x_n | \mu_{1:M}, \Sigma_{1:M}, \pi_{1:M}) = \prod_{i=1}^n \sum_{m=1}^M \pi_m \mathcal{N}(x_i | \mu_m, \Sigma_m).$$

- Complete data likelihood:

$$p(x_1, \dots, x_n, y_1, \dots, y_n | \mu_{1:M}, \Sigma_{1:M}, \pi_{1:M}) = \prod_{i=1}^n \pi_{y_i} \mathcal{N}(x_i | \mu_{y_i}, \Sigma_{y_i}).$$

- Let $\theta = (\mu_{1:M}, \Sigma_{1:M}, \pi_{1:M})$ be all model parameters.
- Iterative **EM-algorithm** for MLE. Given previous estimate $\hat{\theta}$
 - E-step:** Compute $Q(\theta) \equiv \mathbb{E}_y (\log p(X, y | \theta) | X, \hat{\theta})$
 - M-step:** Update $\hat{\theta} \leftarrow \arg \max_{\theta} Q(\theta)$
- The **expected log-likelihood** is

$$Q(\theta) = \sum_{i=1}^n \sum_{m=1}^M w_i(m) [\log \mathcal{N}(x_i | \mu_m, \Sigma_m) + \log \pi_m]$$

where $w_i(m) \equiv p(y_i = m | \hat{\theta}, x_i)$.

EM-algorithm for a Gaussian mixture model

- (E-step) Compute probabilities for the latent y_i

$$w_i(m) = \frac{\pi_m \mathcal{N}(x_i | \mu_m, \Sigma_m)}{\sum_{j=1}^M \pi_j \mathcal{N}(x_i | \mu_j, \Sigma_j)} \quad \text{and} \quad \hat{n}_m = \sum_{i=1}^n w_i(m)$$

- (M-step) Given $w_i(m)$, compute ML estimates by maximizing $Q(\theta)$:

$$\hat{\pi}_m = \frac{\hat{n}_m}{n}$$

$$\hat{\mu}_m = \frac{1}{\hat{n}_m} \sum_{i=1}^n w_i(m) x_i$$

$$\hat{\Sigma}_m = \frac{1}{\hat{n}_m} \sum_{i=1}^n w_i(m) (x_i - \hat{\mu}_m)(x_i - \hat{\mu}_m)^\top$$

- Iterate until log-likelihood

$$\sum_{i=1}^N \log \left\{ \sum_{m=1}^M \pi_m \mathcal{N}(x_i | \mu_m, \Sigma_m) \right\}$$

satisfies some stopping rule.

Gaussian mixture model for old Faithful data

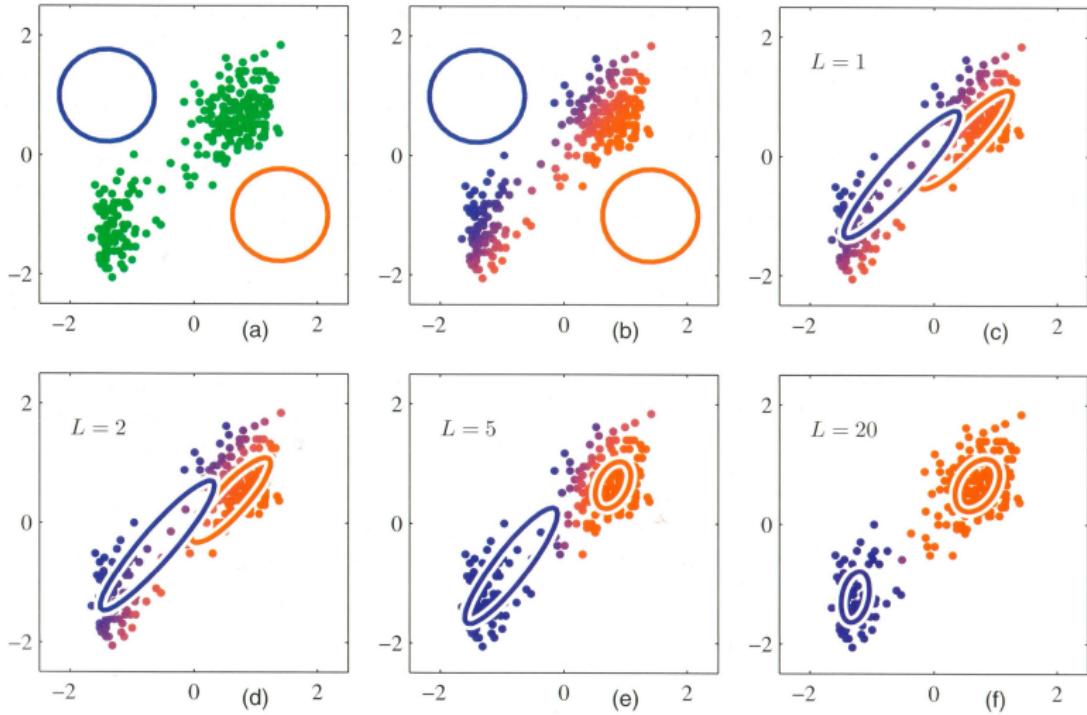


Figure from Bishop (2011).

EM-algorithm for a Gaussian mixture model

- Log-likelihood is **guaranteed to not decrease** at any iteration
- EM is typically **slow**.
Switch to Newton-Raphson when closer to the maximum.
- EM solution often depends on the **initial values**.
Restart with different initial values and pick the best solution.
- **Label-switching** for mixtures. There are actually $M!$ maxima.
- EM tends to find a finite maximum even though the **likelihood is unbounded**. Singularities. Likelihood becomes arbitrarily large if m th component is $\mathcal{N}(x_j, \Sigma_m)$ with $\Sigma_m \rightarrow 0$.
- (pssst: use a Bayesian prior!)

k-Means clustering

- Data in \mathbb{R}^p : x_1, \dots, x_n .
- Aim: **partition** the data into M **clusters**.
- Each observation x_i is represented by a **centroid** $\mu_m \in \mathbb{R}^p$.
- Let responsibility $r_{im} = 1$ if x_i belongs to μ_m .
- **k-means clustering** minimizes

$$\sum_{i=1}^n \sum_{m=1}^M r_{im} \|x_i - \mu_m\|^2$$

with respect to the r_{im} and the μ_1, \dots, μ_M .

- Iterative algorithm: initialize μ_1, \dots, μ_M , then:
 - ▶ Allocate each observations to nearest centroid ($r_{im} = 1$)
 - ▶ Recompute each centroid as mean of its allocated observations.

k-Means clustering of old faithful data

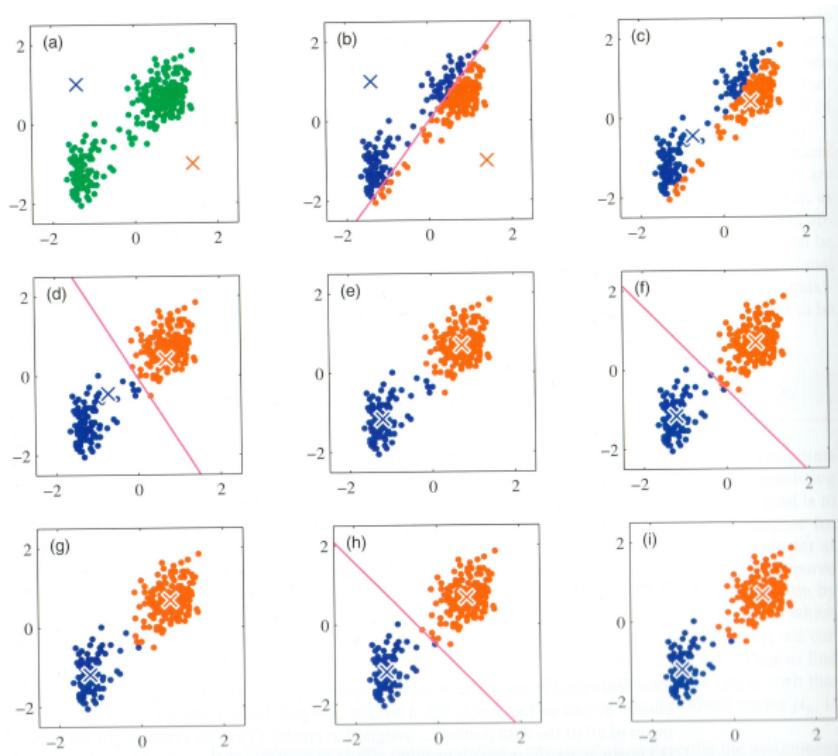


Figure from Bishop (2011).

Clusterings songs by k-means

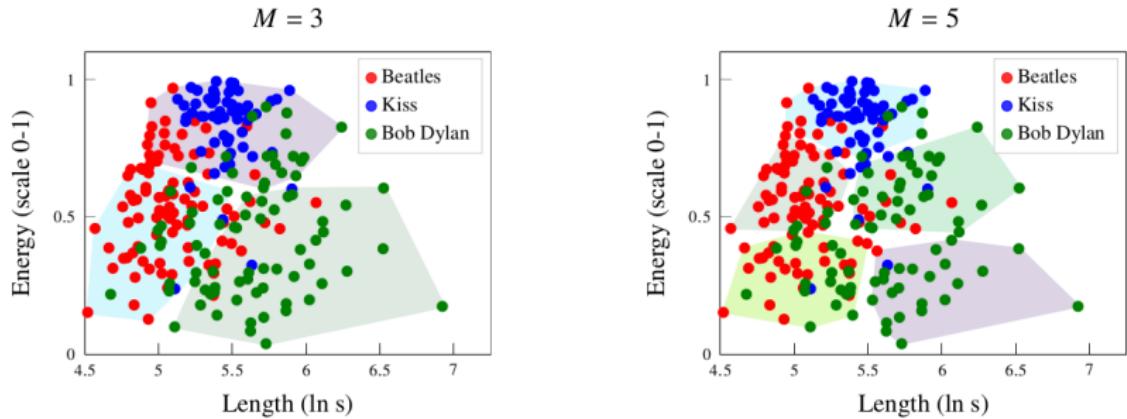


Figure from Lindholm et al. (2021).

Selecting the number of clusters - elbow

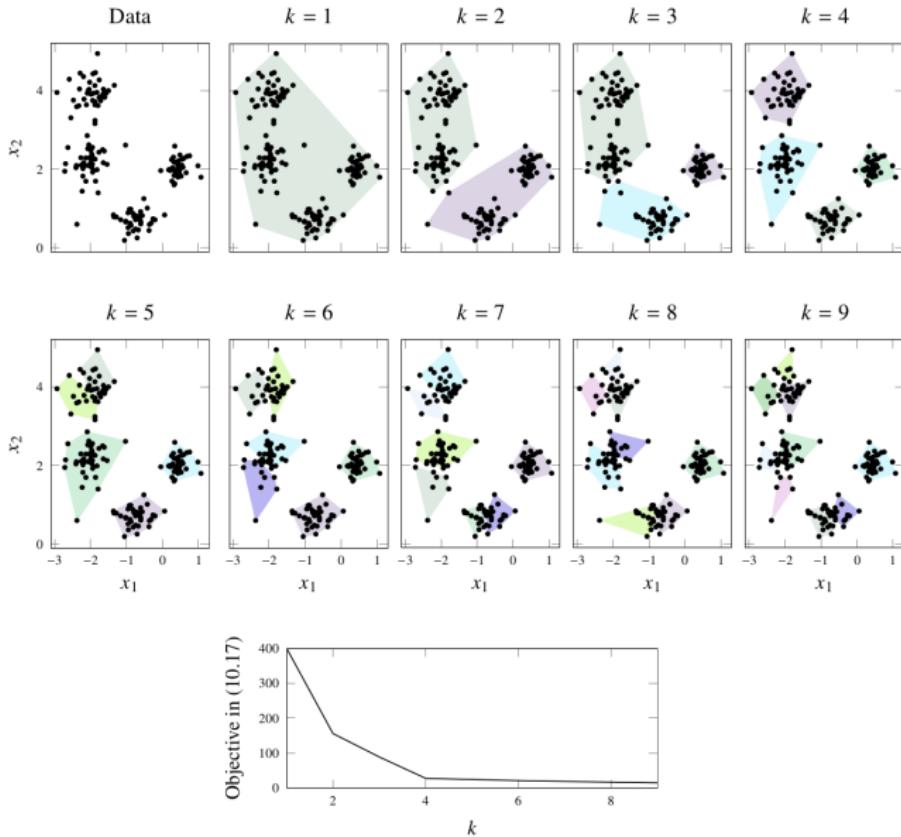


Image compression by k-means



Figure from Bishop (2011).