FEM21045-20 & FEM31002-20

Machine Learning (in Finance)

Unsupervised Learning - part 2

Dick van Dijk

Econometric Institute

Erasmus University Rotterdam

e-mail: djvandijk@ese.eur.nl

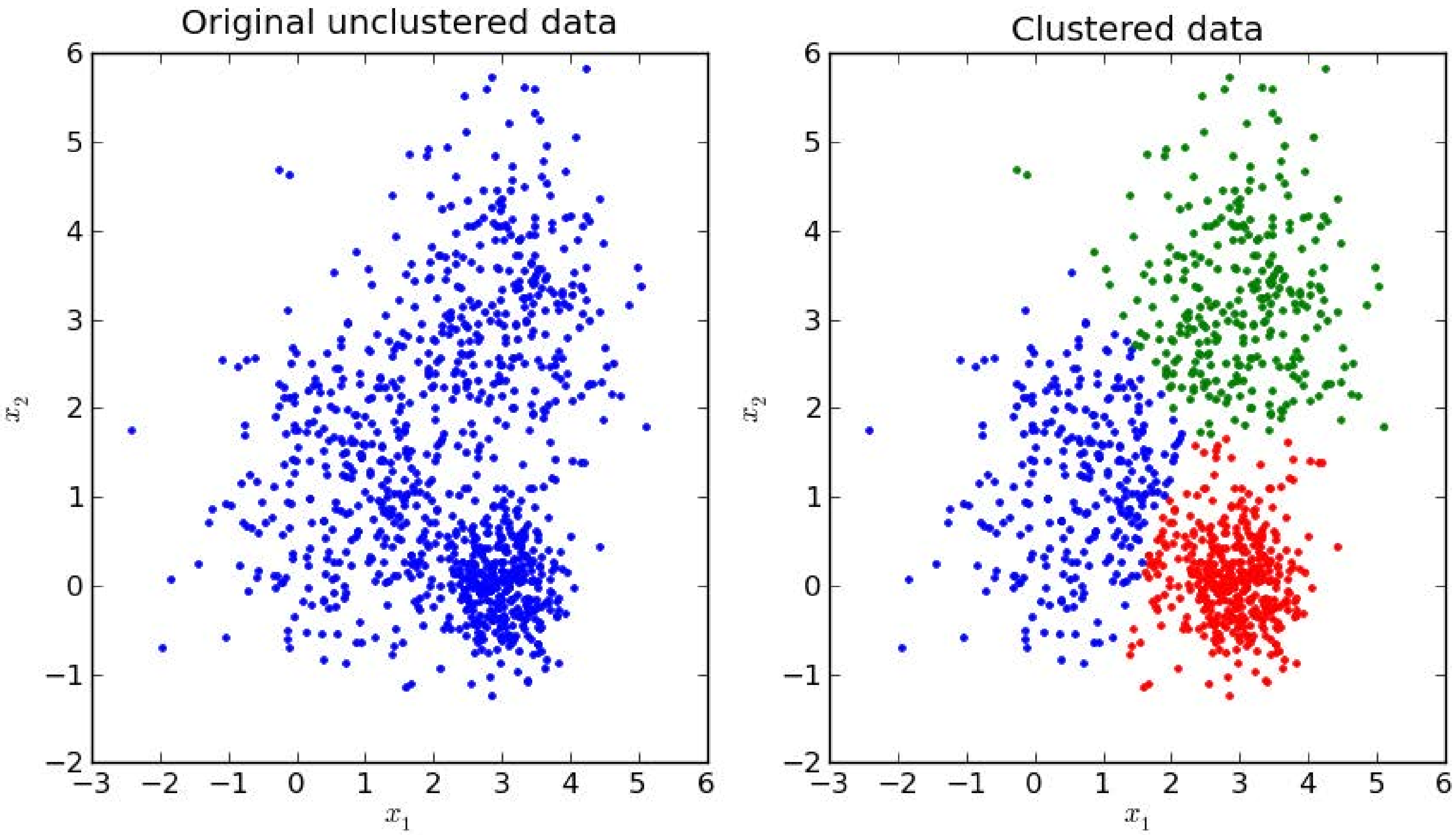
Block 1 (Sep-Oct 2020)

Unsupervised Learning

Outline

- ★ Unsupervised Learning: What and why?
- ★ Principal Components Analysis
- ★ Non-negative Matrix Factorization
  (or: The secret behind online recommendation systems)
- ★ K-means clustering
- ★ Hierarchical clustering
- ★ Gaussian mixture models and the EM algorithm

Cluster analysis



Cluster analysis

★ We have a training sample $\{x_i; i = 1, \ldots, N\}$, where $x_i^T = (x_{i1}, x_{i2}, \ldots, x_{ip})$ is p-dimensional, and p can be large.

★ Observations i may represent individuals (consumers, firms, assets, countries) or time periods (days, months, years)

★ Key question: are there distinct subgroups of observations in the data with substantially different properties?

★ Two crucial ingredients

1. Measure of (dis)similarity [or distance] between observations
2. Clustering algorithm

Dissimilarity measures

★ Quantify the dissimilarity / distance between any two data points i and i'.

★ Typically based on the observed variables/features x_{ij}

★ Default choice is squared Euclidean distance

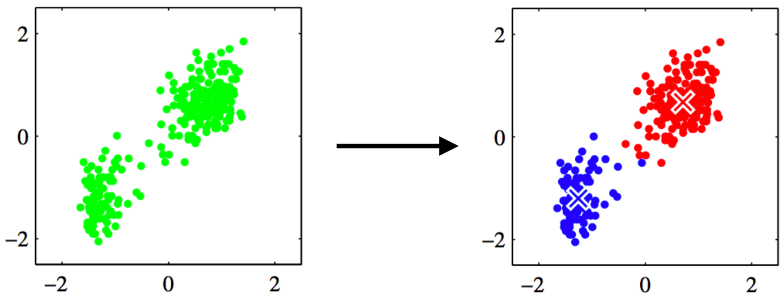
$$D(x_i, x_{i'}) = \|x_i - x_{i'}\|^2 = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

• Works well for quantitative/continuous features x_{ij}; categorical and ordinal variables require some special care

• Possibly use weighted sum $\sum_{j=1}^{p} w_j (x_{ij} - x_{i'j})^2$ (with $w_j \geq 0$ and $\sum_{j=1}^{p} w_j = 1$), because some features may be more important than others, or to account for differences in scale.

Clustering algorithms

★ Goal: partition the training sample into K groups ("clusters") of 'similar' observations



★ Combinatorial algorithms vs. mixture models

★ ('Nonparametric') Combinatorial algorithms minimize the within-cluster dissimilarity

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} D(x_i, x_{i'}),$$

where $C(i) = k$ is an *encoder* assigning observation i to cluster k.

K-means clustering

★ Use latent variable to represent encoder

$$z_{ik} = \begin{cases} 1 & \text{if observation } i \text{ is assigned to cluster } k \\ 0 & \text{otherwise} \end{cases}$$

★ K-means clustering solves

$$\begin{aligned} \min_{m,z} \; & \sum_{k=1}^{K} \sum_{i=1}^{N} z_{ik} \|x_i - m_k\|^2 \\ \text{s.t.} \; & \sum_{k=1}^{K} z_{ik} = 1, \quad i = 1, 2, \ldots, N \\ & z_{ik} \in \{0, 1\}, \quad i = 1, 2, \ldots, N; k = 1, 2, \ldots, K \\ & m_k \in \mathbb{R}^p, \quad k = 1, 2, \ldots, K. \end{aligned}$$

K-means clustering

★ NP-hard problem, but intuitive heuristic available:

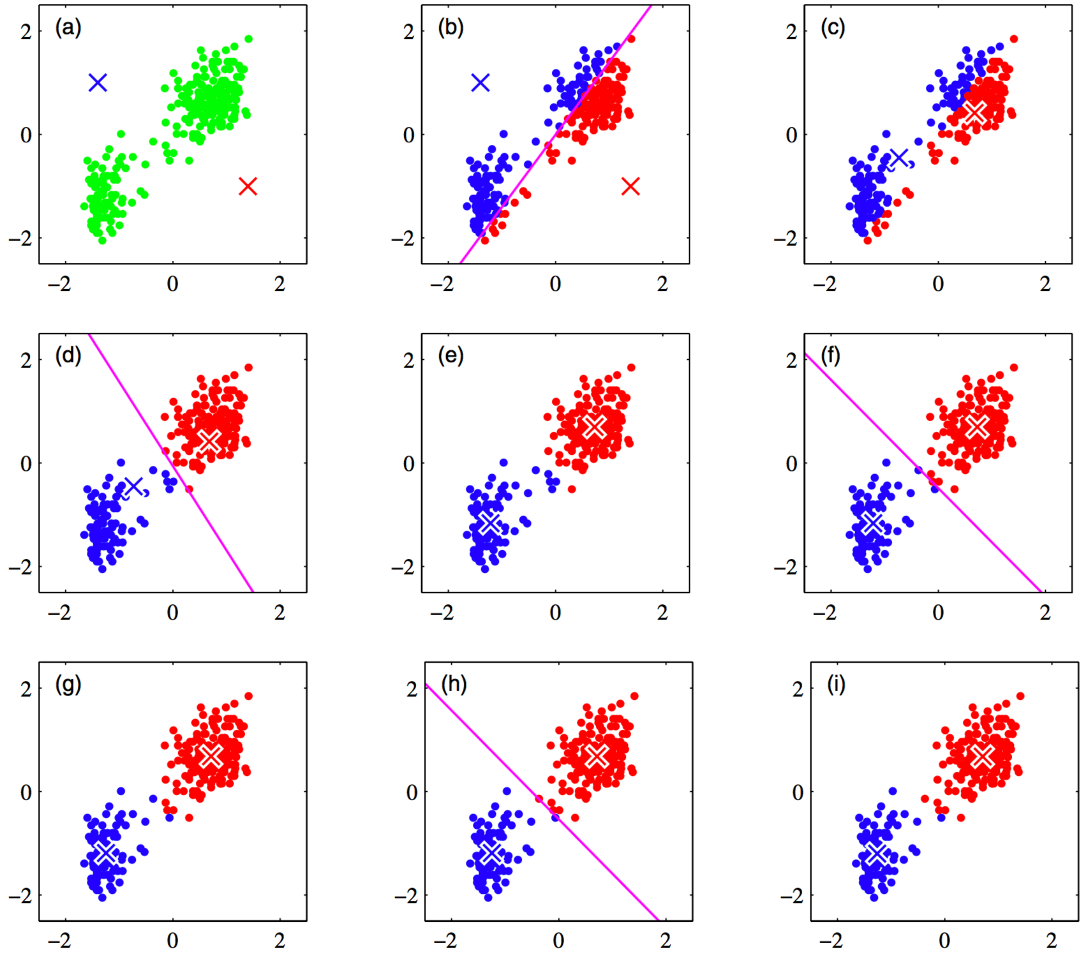
- Initialize with random means $m_k \in \mathbb{R}^p$
- Iterate until convergence:
  - ★ **E-step**: Given the current means, update the assignments

$$z_{ik} = \begin{cases} 1 & \text{if } k = \text{argmin}_j \|x_i - m_j\|^2 \\ 0 & \text{otherwise} \end{cases}$$

  - ★ **M-step**: Given the current assignments, update the means

$$m_k = \frac{\sum_{i=1}^{N} z_{ik} x_i}{\sum_{i=1}^{N} z_{ik}}$$

K-means clustering



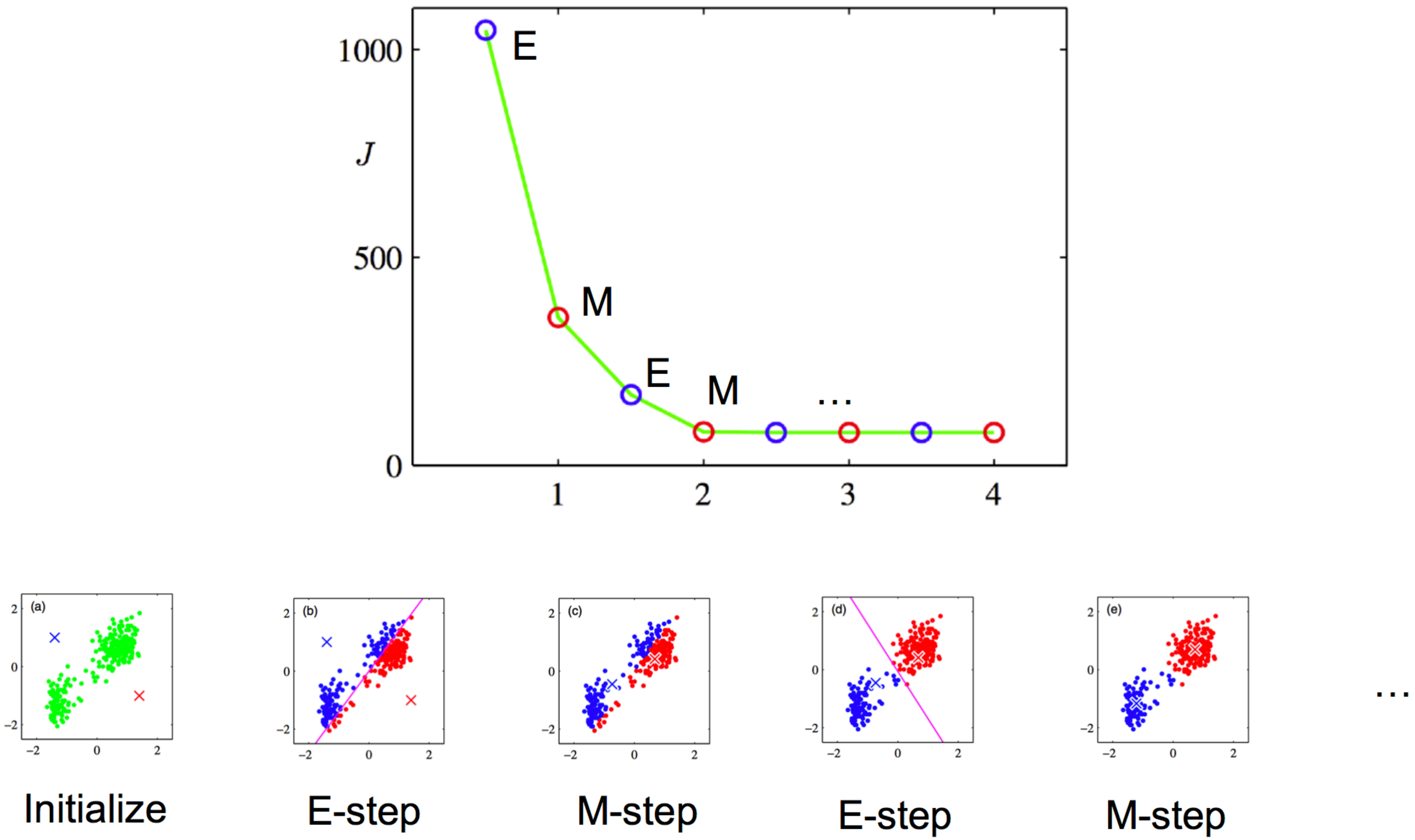
K-means clustering - convergence

★ At each step, the objective function does not deteriorate $\Rightarrow$ convergence is guaranteed

★ Convergence is to **local** minimum only

★ To improve chances of finding the **global** minimum, run the algorithm with different initial cluster means.

K-means clustering - convergence



K-means clustering - pros and cons

★ Good

- Simple to implement
- Fast

★ Bad

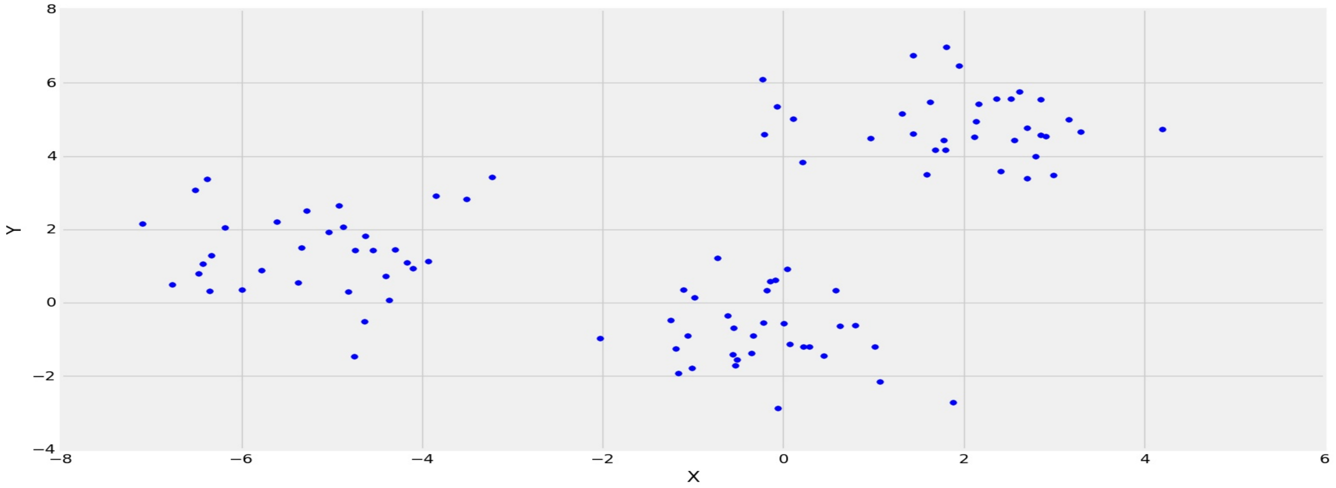
- Local minima
- Can fail miserably in case of 'non-spherical' clusters
- Sensitive to the features scale
- Number of clusters K to be chosen in advance
- Cluster assignments are 'hard, not 'soft'/probabilistic (ict Gaussian Mixture Model)

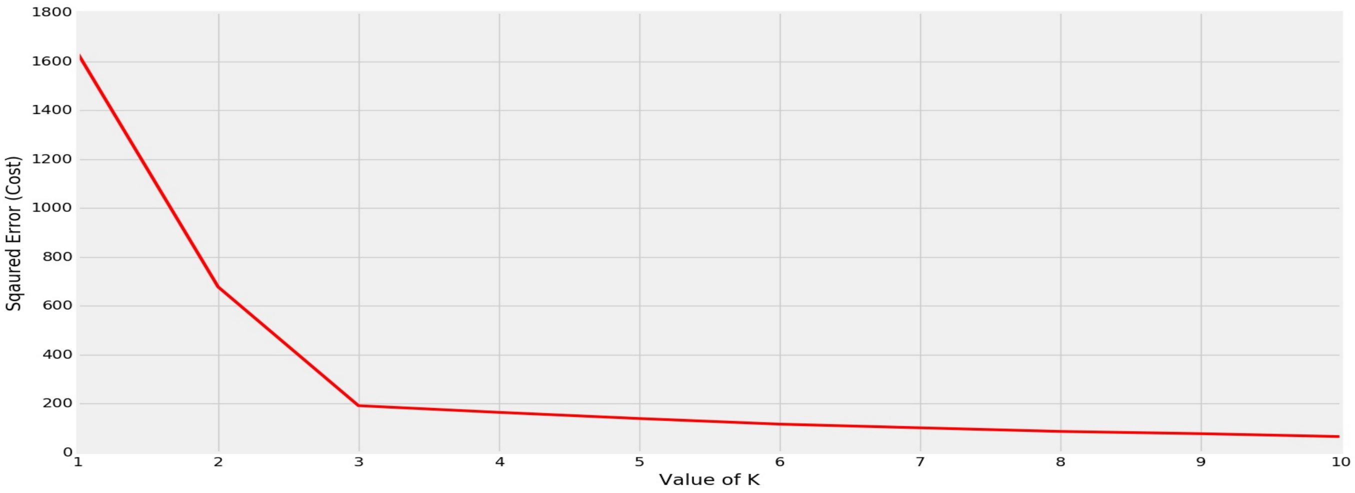
K-means clustering - the mysterious K

★ How to choose the number of clusters (hyperparameter!) K?

- May be given in application
- Using statistical measures – requires distributional assumptions
- Elbow plot: Increasing K gradually and monitoring within-cluster dissimilarity values

K-means clustering - the elbow plot





Hierarchical clustering

★ Produces a hierarchy of clusterings for all $K = 1, 2, \ldots, N$

★ Requires dissimilarity measure between **groups** of observations

★ Approach can be

- *Agglomerative* (bottom-up): start with N singleton clusters and successively merge the two 'closest' clusters
- *Divisive* (top-down): start with a single cluster and successively split a cluster resulting in two new clusters with the largest possible between-group dissimilarity.

Hierarchical clustering - dissimilarity measures

★ Single linkage ('nearest-neighbors')

$$d_{SL}(G,H) = \min_{i \in G, i' \in H} D(x_i, x_{i'})$$

★ Complete linkage ('furthest-neighbors')

$$d_{CL}(G,H) = \max_{i \in G, i' \in H} D(x_i, x_{i'})$$

★ Group average

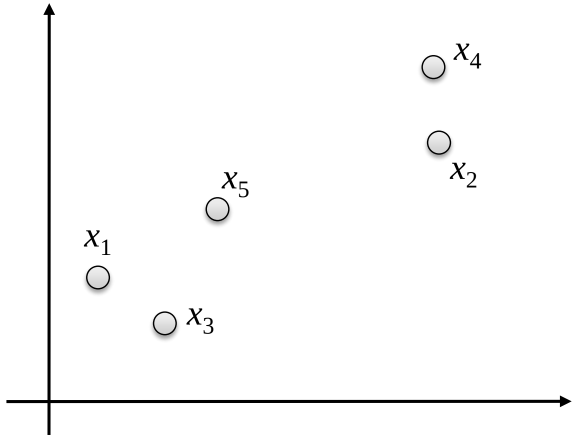
$$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{i' \in H} D(x_i, x_{i'})$$

$\Rightarrow$ Results may be sensitive to the dissimilarity measure used...

Hierarchical clustering

Ilker Birbil, 2020

Hierarchical Clustering (Complete Linkage)



Clusters: $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}$

$s(\{x_1\}|\{x_2\}) = 5.0$ $\boxed{s(\{x_1\}|\{x_3\}) = 0.5}$

$s(\{x_1\}|\{x_4\}) = 4.5$ $s(\{x_1\}|\{x_5\}) = 2.0$

$s(\{x_2\}|\{x_3\}) = 4.7$ $s(\{x_2\}|\{x_4\}) = 0.6$

$s(\{x_2\}|\{x_5\}) = 3.0$ $s(\{x_3\}|\{x_4\}) = 4.0$

$s(\{x_3\}|\{x_5\}) = 2.2$ $s(\{x_4\}|\{x_5\}) = 2.5$

Clusters: $\{x_1, x_3\}, \{x_2\}, \{x_4\}, \{x_5\}$

$s(\{x_1, x_3\}|\{x_2\}) = 5.0$ $s(\{x_1, x_3\}|\{x_4\}) = 4.5$ $s(\{x_1, x_3\}|\{x_5\}) = 2.2$

$\boxed{s(\{x_2\}|\{x_4\}) = 0.6}$ $s(\{x_2\}|\{x_5\}) = 3.0$ $s(\{x_4\}|\{x_5\}) = 2.5$

Clusters: $\{x_1, x_3\}, \{x_2, x_4\}, \{x_5\}$

$s(\{x_1, x_3\}|\{x_2, x_4\}) = 5.0$ $\boxed{s(\{x_1, x_3\}|\{x_5\}) = 2.2}$ $s(\{x_2, x_4\}|\{x_5\}) = 3.0$

Clusters: $\{x_1, x_3, x_5\}, \{x_2, x_4\}$

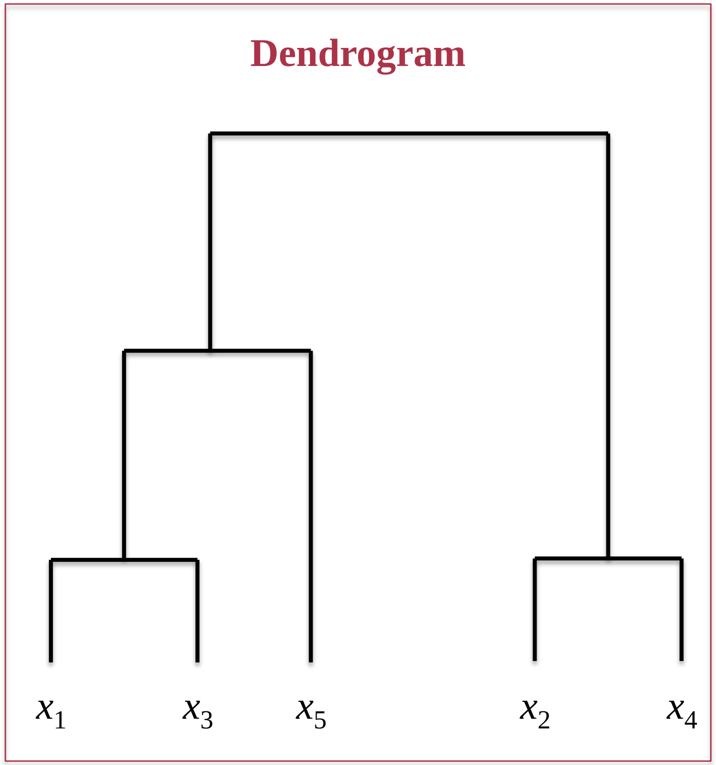
$\boxed{s(\{x_1, x_3, x_5\}|\{x_2, x_4\}) = 5.0}$

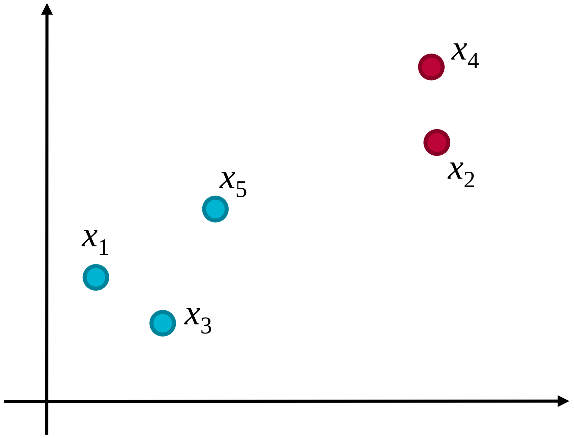
Erasmus Universiteit Rotterdam

Hierarchical clustering

Ilker Birbil, 2020

Hierarchical Clustering





If we want **three** clusters, then these would be $\{x_1, x_3\}, \{x_5\}$ and $\{x_2, x_4\}$

Gaussian mixture models

★ 'Generative model' or Data Generating Process (DGP)

- There are K clusters in X. The latent multinomial variable $z_i \in \{1, 2, \ldots, K\}$ represents the relevant cluster for observation i, with prior probability

$$\Pr(z_i = k) = \pi_k, \quad \text{with } \pi_k \in [0, 1] \text{ and } \sum_{k=1}^{K} \pi_k = 1$$

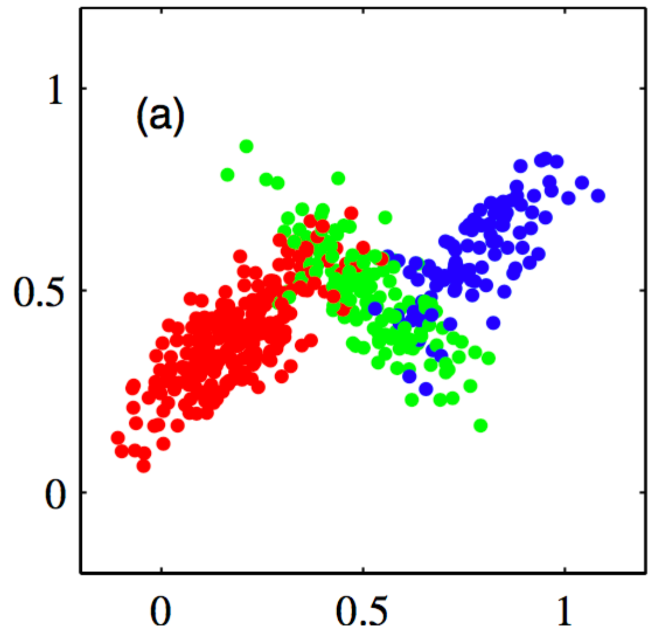
- Observations in cluster k are normally distributed with mean μ_k and covariance matrix Σ_k, that is

$$f(x_i | z_i = k) = \phi(x_i; \mu_k, \Sigma_k)$$

- The unconditional/marginal distribution of an observation x_i then is a mixture of normals

$$f(x_i) = \sum_{z_i} f(x_i, z_i) = \sum_{k=1}^{K} f(x_i | z_i = k) \Pr(z_i = k) = \sum_{k=1}^{K} \pi_k \phi(x_i; \mu_k, \Sigma_k)$$

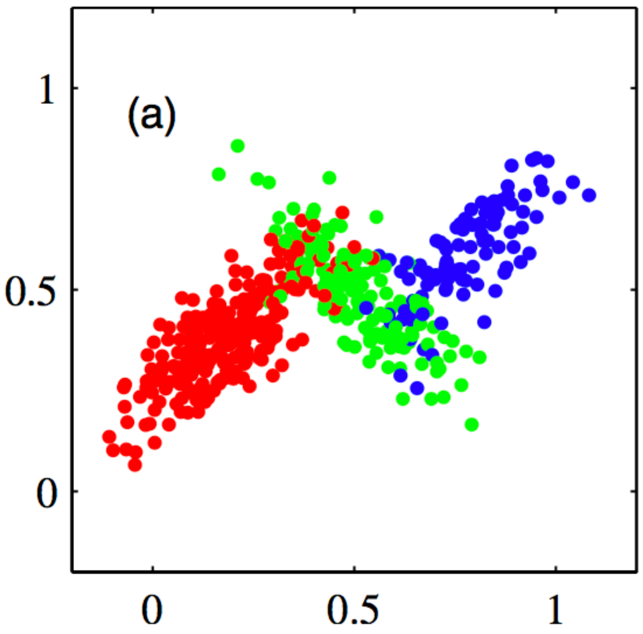
Gaussian mixture models



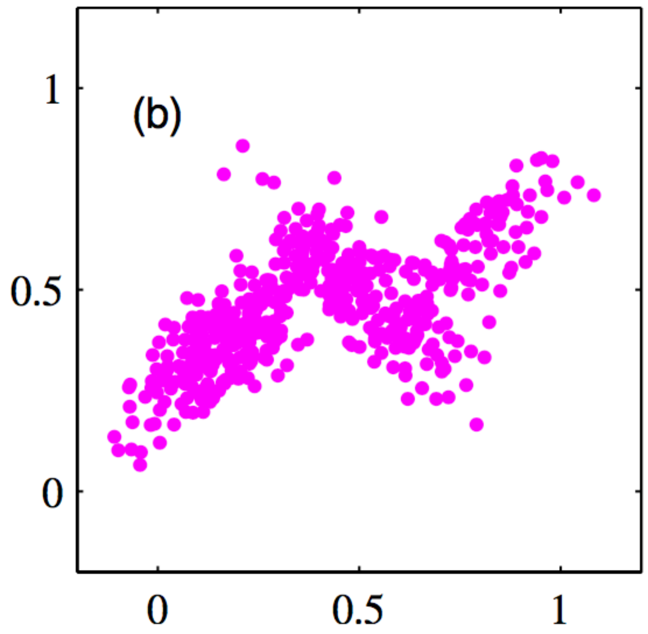
Original data

★ Observations x_i generated from GMM with $K = 3$

Gaussian mixture models



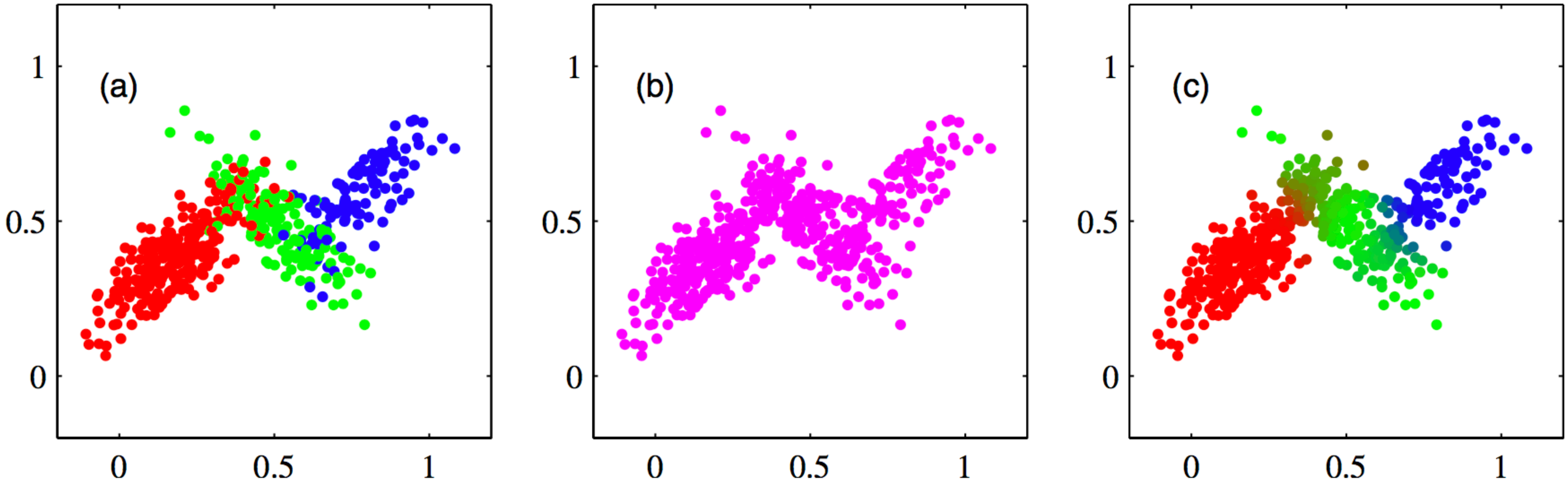
Original data



Unlabelled sample

★ But... we only observe x_i as an 'unlabelled sample' – we do not know/observe z_i!

Gaussian mixture models



Original data | Unlabelled sample | GMM: soft-clustering

★ But... we can estimate the parameters π_k, μ_k and Σ_k in the Gaussian mixture. Moreover, we can obtain posterior probabilities

$$\Pr(z_i = k | x_i), \qquad \text{for } k = 1, \ldots, K,$$

which can be used for '**soft clustering**' of the observations.

Gaussian mixture models - soft clustering

The posterior cluster probabilities are easily obtained as

$$
\begin{aligned}
\text{Pr}(z_i = k|x_i) &= \frac{f(x_i, z_i = k)}{f(x_i)} \\
&= \frac{\text{Pr}(z_i = k) f(x_i|z_i = k)}{\sum_{k=1}^{K} f(x_i|z_i = k)\text{Pr}(z_i = k)} \\
&= \frac{\pi_k \phi(x_i; \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \phi(x_i; \mu_k, \Sigma_k)}
\end{aligned}
$$

Gaussian mixture models - estimation

★ Estimation is done by deriving a **likelihood function** from the GMM – although we need to do a bit more than 'standard' maximum likelihood...

★ Normally, the **likelihood function** is taken to be the joint density function of the N observations $\mathbf{X}^T = (x_1, x_2, \ldots, x_N)$:

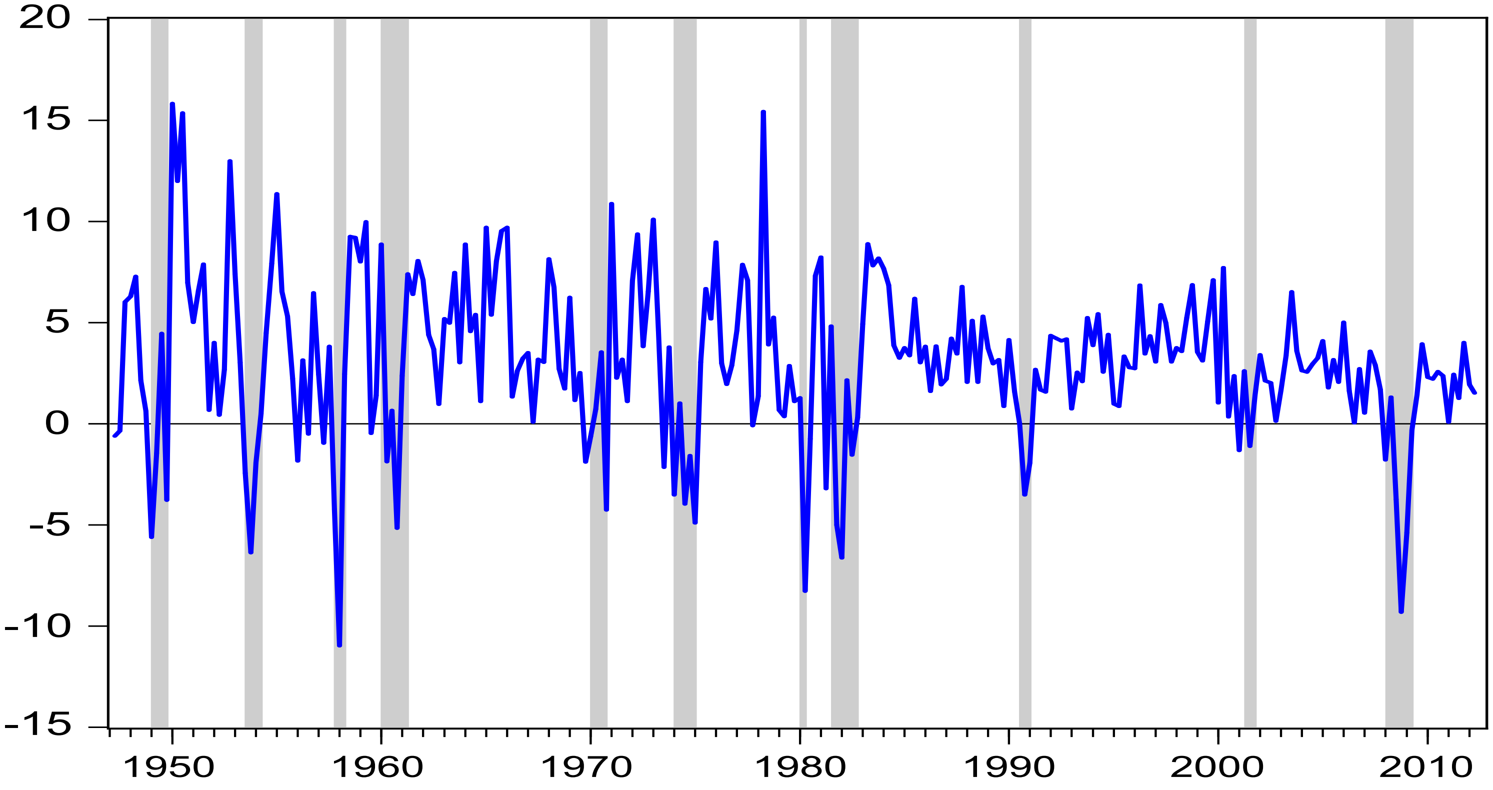
$$\mathcal{L}(\mathbf{X}|\boldsymbol{\theta}) = f(x_1, x_2, \ldots, x_N; \boldsymbol{\theta}) = \prod_{i=1}^{N} f(x_i; \boldsymbol{\theta}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \phi(x_i; \mu_k, \Sigma_k)$$

where $\boldsymbol{\theta}$ is the vector of all parameters in the model.

★ Here this is not convenient, because we get a summation inside the logarithm, when we (as usual) consider the log-likelihood:

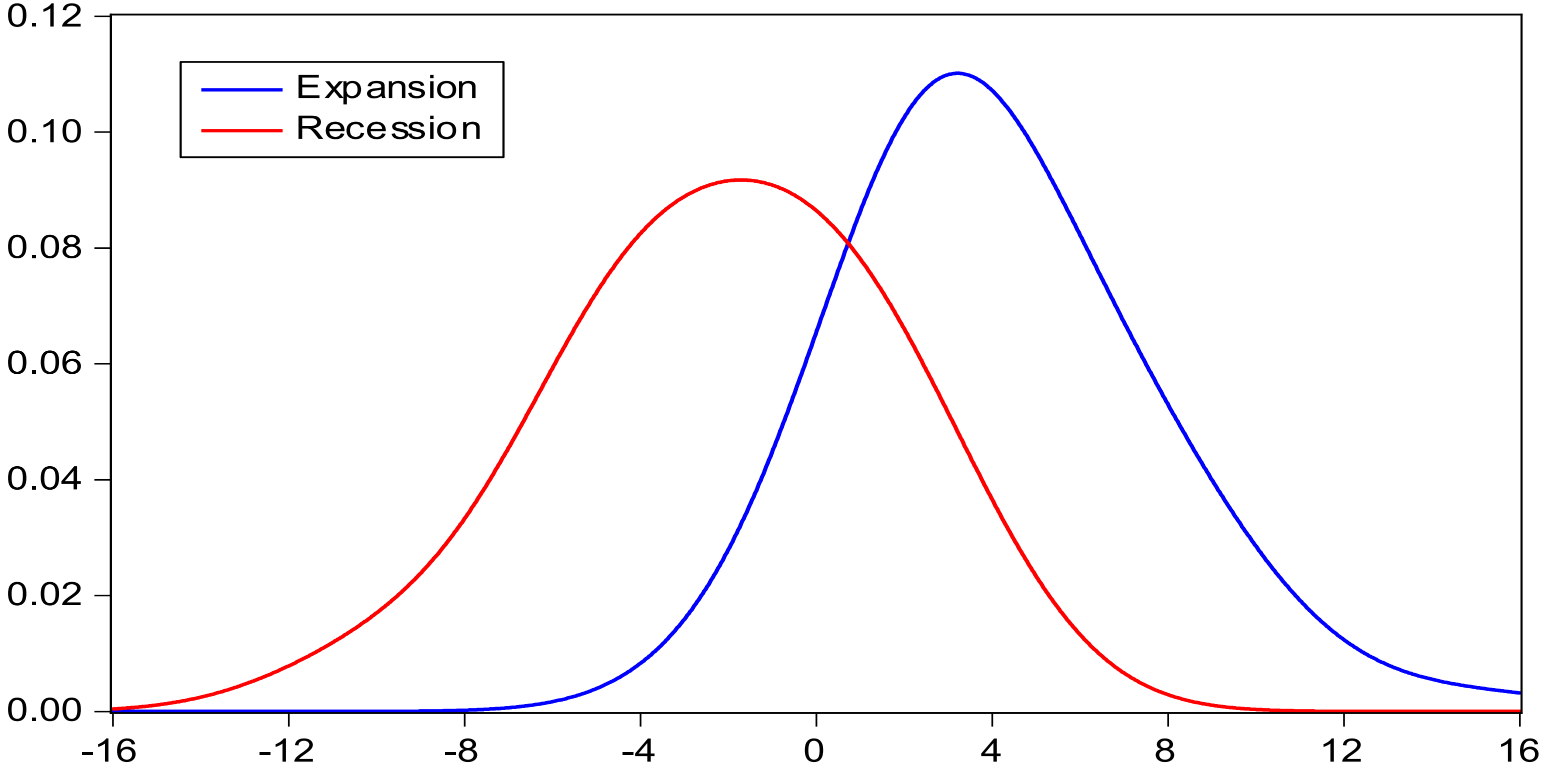
$$\updownarrow(\mathbf{X}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \phi(x_i; \mu_k, \Sigma_k) \right).$$

Gaussian mixture models



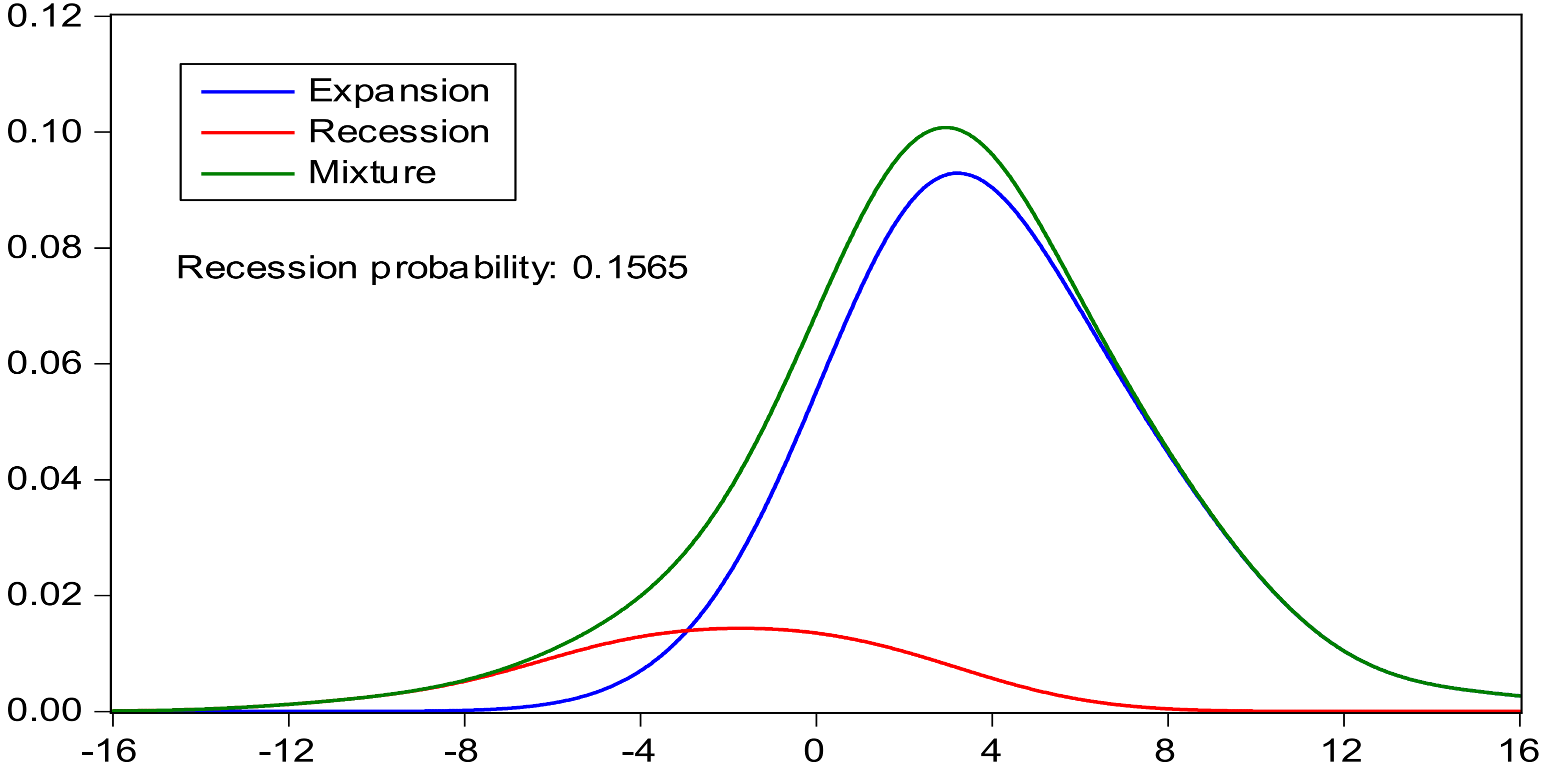
Quarterly growth rates US real GDP, 1947Q2 - 2012Q2

Gaussian mixture models



Quarterly growth rates US real GDP, 1947Q2 - 2012Q2
Densities in NBER recessions and expansions

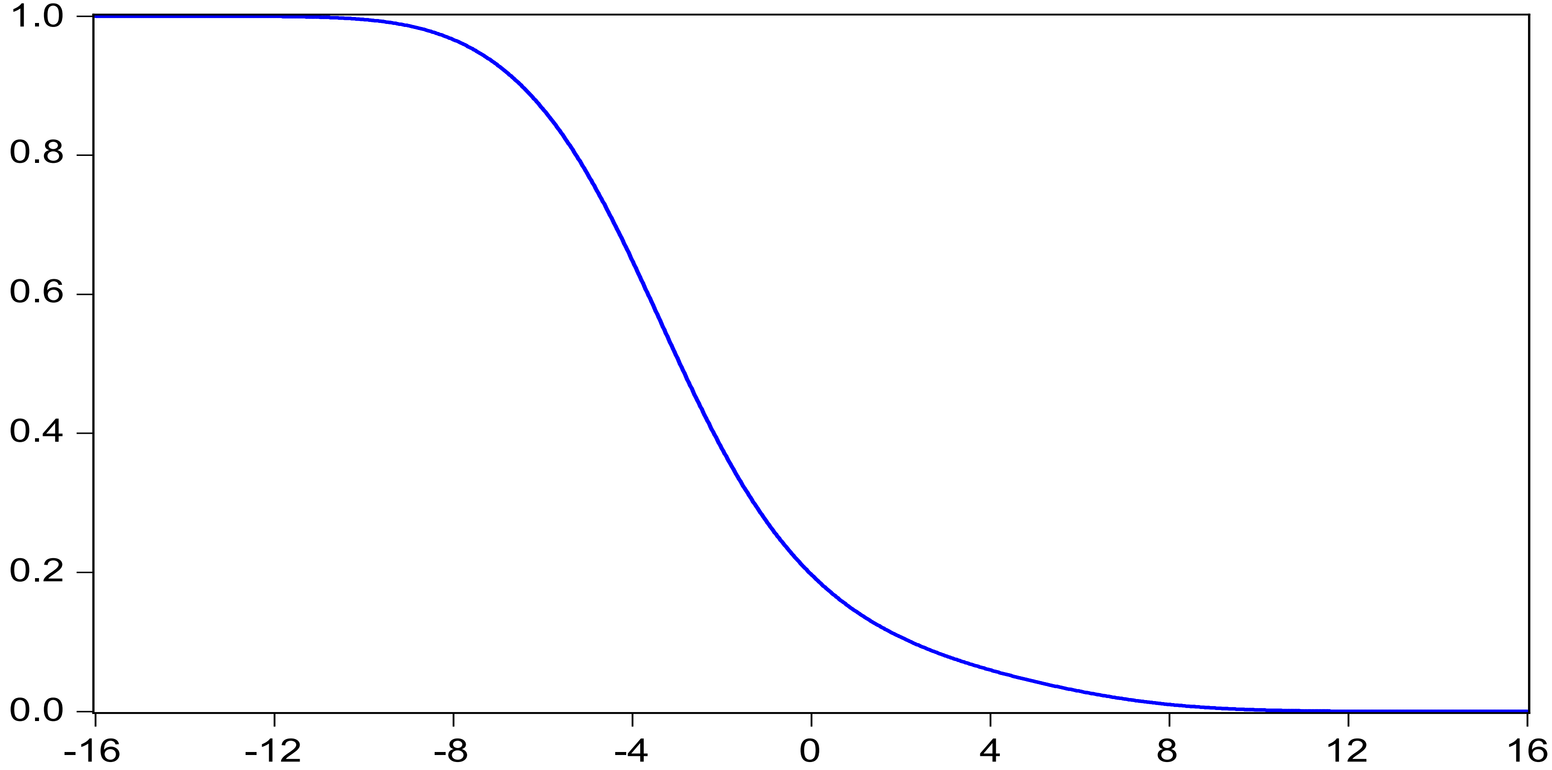
Gaussian mixture models



Quarterly growth rates US real GDP, 1947Q2 - 2012Q2

Mixture density

Gaussian mixture models



Quarterly growth rates US real GDP, 1947Q2 - 2012Q2
Posterior probability of recession

Parameter estimation: EM algorithm

Parameter estimation in the GMM may be done using the **EM algorithm**. This (general) estimation method provides a (local) maximum of the log likelihood function using an iterative two-step procedure:

- **E = Expectation**
- **M = Maximization**

Parameter estimation: EM algorithm

Key idea: Although z_i is unobserved, we may consider it to be part of the dataset. We then consider the so-called **complete data likelihood function**, based on the joint density of x_i and z_i.

Recall that the joint density of x_i and z_i is given by

$$f(x_i, z_i; \boldsymbol{\theta}) = f(x_i | z_i; \boldsymbol{\theta}) \mathrm{Pr}(z_i = k),$$

which may also be written as

$$f(x_i, z_i; \boldsymbol{\theta}) = \left[\pi_1 \phi(x_i; \mu_1, \sigma_1^2)\right]^{\mathrm{I}[z_i=1]} \left[(1-\pi_1)\phi(x_i; \mu_2, \sigma_2^2)\right]^{\mathrm{I}[z_i=2]}.$$

Parameter estimation: EM algorithm

Hence, the **complete data likelihood function** is given by

$$f(\mathbf{x},\mathbf{z};\boldsymbol{\theta}) = \prod_{i=1}^{N} \Big(\left[\pi_1\phi(x_i;\mu_1,\sigma_1^2)\right]^{\mathrm{I}[z_i=1]} \left[(1-\pi_1)\phi(x_i;\mu_2,\sigma_2^2)\right]^{\mathrm{I}[z_i=2]} \Big),$$

where $\mathbf{x}^T = (x_1, x_2, \ldots, x_N)$ and $\mathbf{z}^T = (z_1, z_2, \ldots, z_N)$.

$\Rightarrow$ This is the function that should be maximized to find the maximum likelihood parameter estimates of $\boldsymbol{\theta}$ (plus the desired soft clustering probability Pr$(z_i = 1|x_i;\boldsymbol{\theta})$).

The problem that z_i is unobserved is solved by using the EM algorithm.

Parameter estimation: EM algorithm

The log complete data likelihood function is given by

$$\ln f(\mathbf{x},\mathbf{z};\boldsymbol{\theta}) = \sum_{i=1}^{N} \Big(\mathtt{I}[z_i = 1] \ln \pi_1 + \mathtt{I}[z_i = 2] \ln(1-\pi_1) + \mathtt{I}[z_i = 1] \ln \phi(x_i;\mu_1,\sigma_1^2) + \mathtt{I}[z_i = 2] \ln \phi(x_i;\mu_2,\sigma_2^2) \Big).$$

The iterative two-step estimation procedure consists of:

- **E-step**: Take the expectation of the log complete data likelihood function with respect to $\mathbf{z}$, given $\mathbf{x}$ and $\boldsymbol{\theta}$

$$\mathsf{E}_{\mathbf{z}}[\ln f(\mathbf{x},\mathbf{z};\boldsymbol{\theta})].$$

- **M-step**: Maximize the expected value with respect to the parameter $\boldsymbol{\theta}$

$$\max_{\boldsymbol{\theta}} \mathsf{E}_{\mathbf{z}}[\ln f(\mathbf{x},\mathbf{z};\boldsymbol{\theta})].$$

Parameter estimation: EM algorithm

E-step: We have to compute the expectation of z_i (or essentially $\mathtt{I}[z_i = 1]$) given x_i and $\boldsymbol{\theta}$.

We know that $\mathsf{E}[\mathtt{I}[z_i = 1]|x_i; \boldsymbol{\theta}] = \mathsf{Pr}(z_i = 1|x_i; \boldsymbol{\theta})$.

Assuming the parameters $\boldsymbol{\theta}$ are known, this conditional (posterior) probability denoted by π^*_{1i} is

$$
\begin{aligned}
\mathsf{Pr}(z_i = 1|x_i; \boldsymbol{\theta}) &= \frac{f(x_i, z_i = 1; \boldsymbol{\theta})}{f(x_i; \boldsymbol{\theta})} \\
&= \frac{f(x_i|z_i = 1; \boldsymbol{\theta})\mathsf{Pr}(z_i = 1)}{\sum_{k=1}^{2} f(x_i|z_i = k; \boldsymbol{\theta})\mathsf{Pr}(z_i = k)} \\
&= \frac{\pi_1 \phi(x_i; \mu_1, \sigma_1^2)}{\pi_1 \phi(x_i; \mu_1, \sigma_1^2) + (1 - \pi_1)\phi(x_i; \mu_2, \sigma_2^2)} \equiv \pi^*_{1i}
\end{aligned}
$$

Parameter estimation: EM algorithm

From the log complete data likelihood function

$$\ln f(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) = \sum_{i=1}^{N} \Big(\mathrm{I}[z_i = 1] \ln \pi_1 + \mathrm{I}[z_i = 2] \ln(1 - \pi_1) + \mathrm{I}[z_i = 1] \ln \phi(x_i; \mu_1, \sigma_1^2) + \mathrm{I}[z_i = 2] \ln \phi(x_i; \mu_2, \sigma_2^2) \Big).$$

we then arrive at the '**expected**' log likelihood function

$$\mathrm{E}_{\mathbf{z}}[\ln f(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})] = \sum_{i=1}^{N} \underbrace{\pi_{1i}^* \ln \pi_1 + (1 - \pi_{1i}^*) \ln(1 - \pi_1)} + \underbrace{\pi_{1i}^* \ln \phi(x_i; \mu_1, \sigma_1^2)} + \underbrace{(1 - \pi_{1i}^*) \ln \phi(x_i; \mu_2, \sigma_2^2)}$$

Parameter estimation: EM algorithm

<u>M-step</u>: The maximization step is easy, as we can consider the three parts of the expected log likelihood function separately. The first part provides the new estimate of π_1

$$\pi_1 = \frac{1}{N} \sum_{i=1}^{N} \pi_{1i}^*.$$

The second part of the expected log likelihood function can be written as

$$\sum_{i=1}^{N} \pi_{1i}^*(-\ln \sigma_1 - \frac{1}{2}(x_i - \mu_1)^2/\sigma_1^2)$$

such that μ_1 and σ_1^2 can be updated as follows

$$\mu_1 = \frac{\sum_{i=1}^{N} \pi_{1i}^* x_i}{\sum_{i=1}^{N} \pi_{1i}^*} \qquad \text{and} \qquad \sigma_1^2 = \frac{\sum_{i=1}^{N} \pi_{1i}^* (x_i - \mu_1)^2}{\sum_{i=1}^{N} \pi_{1i}^*}.$$

The third part proceeds in the same way.

EM iterations for the GMM