Lecture 9: Clustering

Andreas Wichert
Department of Computer Science and Engineering
Técnico Lisboa

What is Cluster Analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is unsupervised classification: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

- We approach first using a nonprobabilistic technique called the K-means algorithm (Lloyd, 1982).
- Then we introduce the latent variable view of mixture distributions in which the discrete latent variables can be interpreted as defining assignments of data points to specific components of the mixture
- A general technique for finding maximum likelihood estimators in latent variable models is the expectation-maximization (EM) algorithm.

- Training set consists on N observations (sample)

$$X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\eta, \cdots, \mathbf{x}_N)$$

- Our goal is to partition the data set into some number *K* of clusters, where we shall suppose for the moment that the value of *K* is given.
- Clustering is a useful tool for data compression.
- Instead of reducing the dimensionality of a data set, clustering reduces the number of data points.

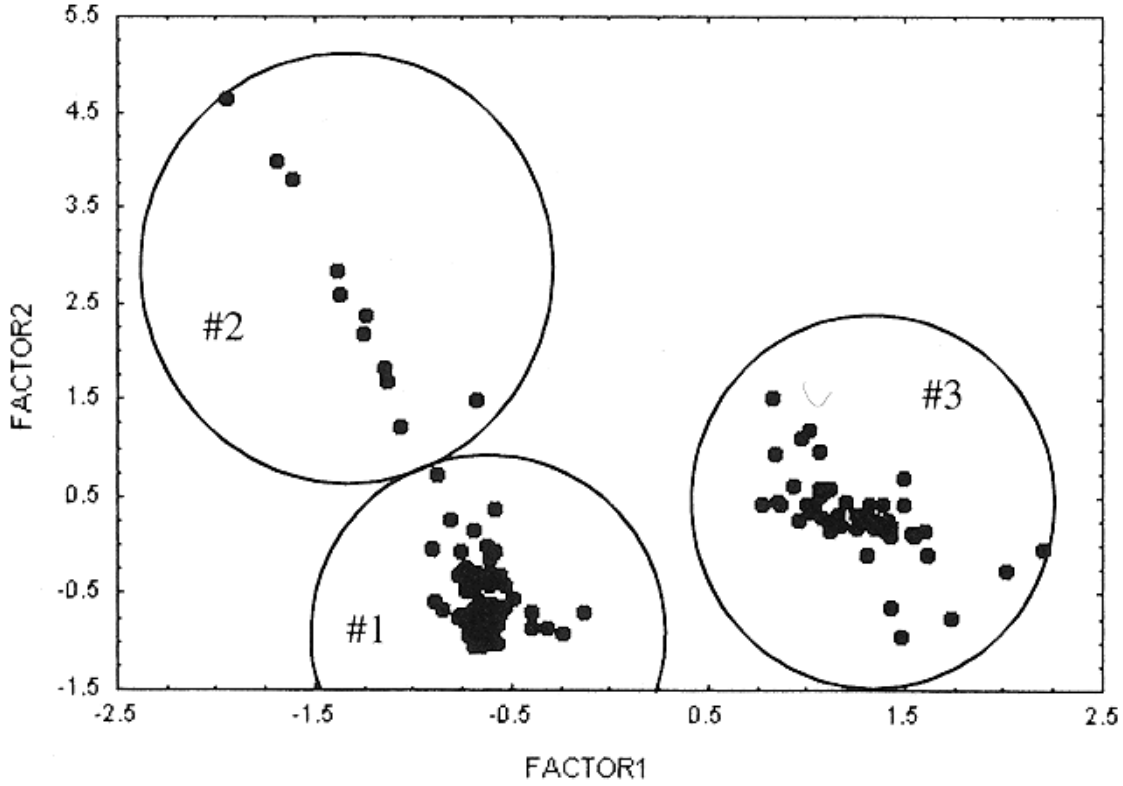
- Custer as comprising a group of data points whose inter-point distances are small compared with the distances to points outside of the cluster
- It groups the data points into clusters according to a distance function.
- The points are similar to one another within the same cluster and dissimilar to the objects in other clusters

- The cluster centers (also called centroids) represent the compressed data set

- The most popular clustering method is *K*-means clustering. We map *N* data points, represented by vectors of dimension *D*, into *K* centroids with

$$K \ll N$$

K-Means Clustering



K-means Clustering

$$d_2(\mathbf{x}, \mathbf{y}) = \|x - y\| = \sqrt{\sum_{i=1}^{D} (x_i - y_i)^2}.$$

In K-means clustering with K vectors called centroids w

$$\mathbf{c}_1, \mathbf{c}_2 \cdots, \mathbf{c}_K$$

and K sets called clusters

$$C_1, C_2 \cdots, C_K.$$

each cluster set is defined as the set of points with where

$$k = 1, ..., K$$

$$C_k = \{\mathbf{x} | d_2(\mathbf{x}, \mathbf{c}_k) = \min_j d_2(\mathbf{x}, \mathbf{c}_j)\}.$$

K-means Clustering

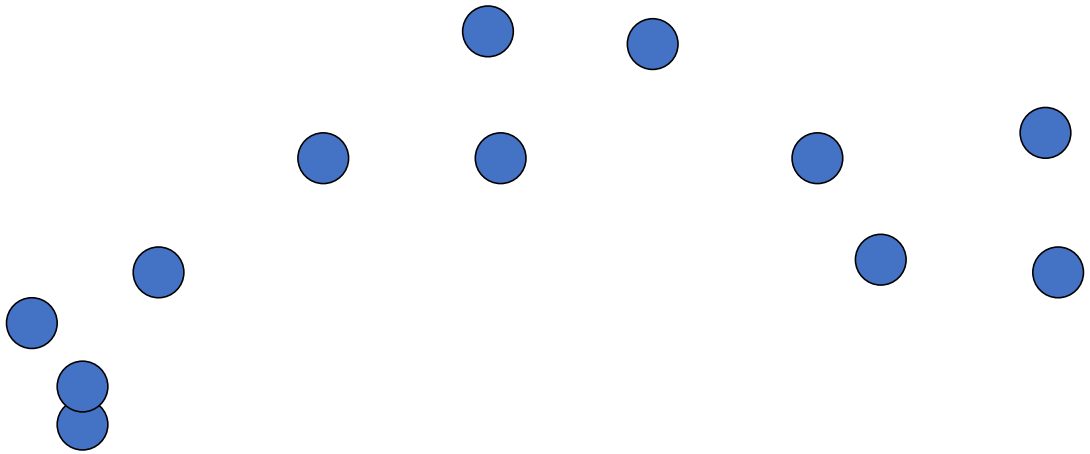
$$k = 1, ..., K$$

$$C_k = \{\mathbf{x} | d_2(\mathbf{x}, \mathbf{c}_k) = \min_j d_2(\mathbf{x}, \mathbf{c}_j)\}.$$

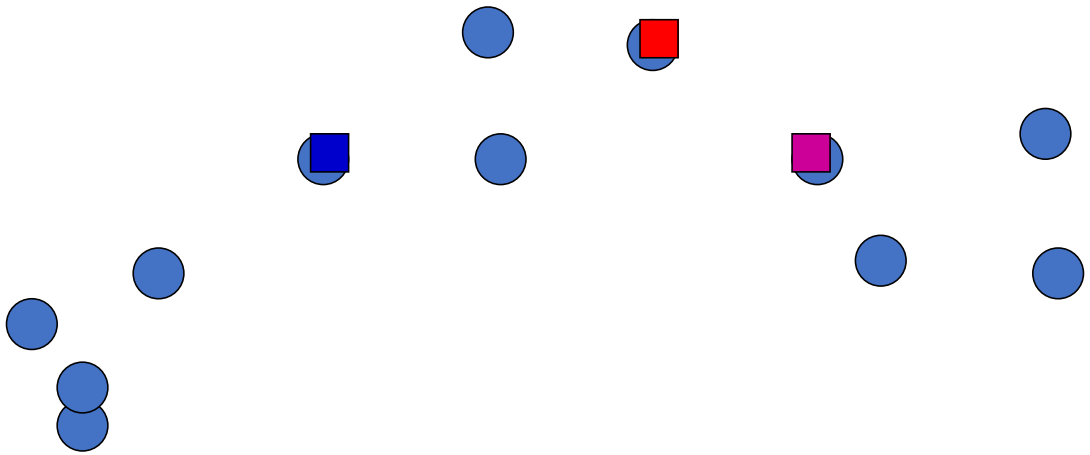
Each cluster C_k contains the points that are closest to the centroid $\mathbf{c}_k$.
centroid $\mathbf{c}_k$ is represented by the mean value of all the points of C_k

$$\mathbf{c}_k = \frac{1}{|C_k|} \cdot \sum_{x \in C_k} \mathbf{x}.$$

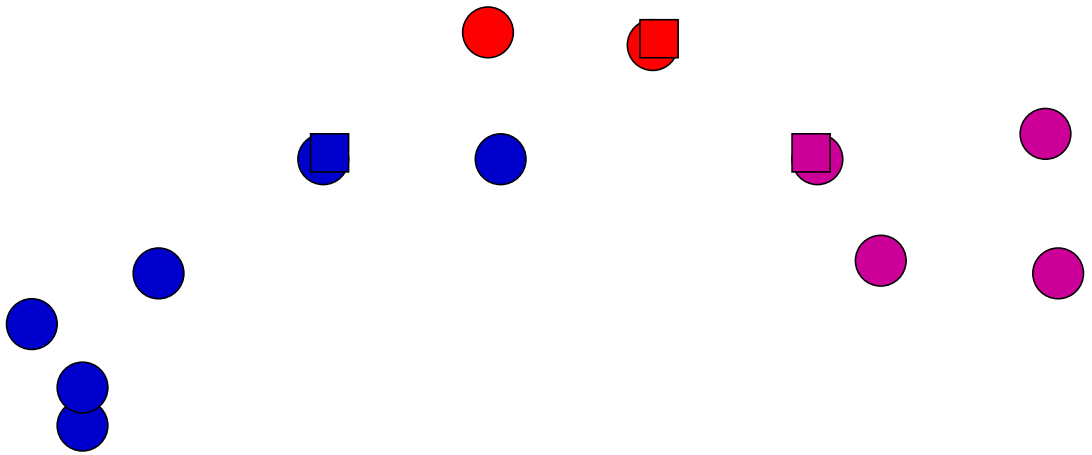
K-means: an example



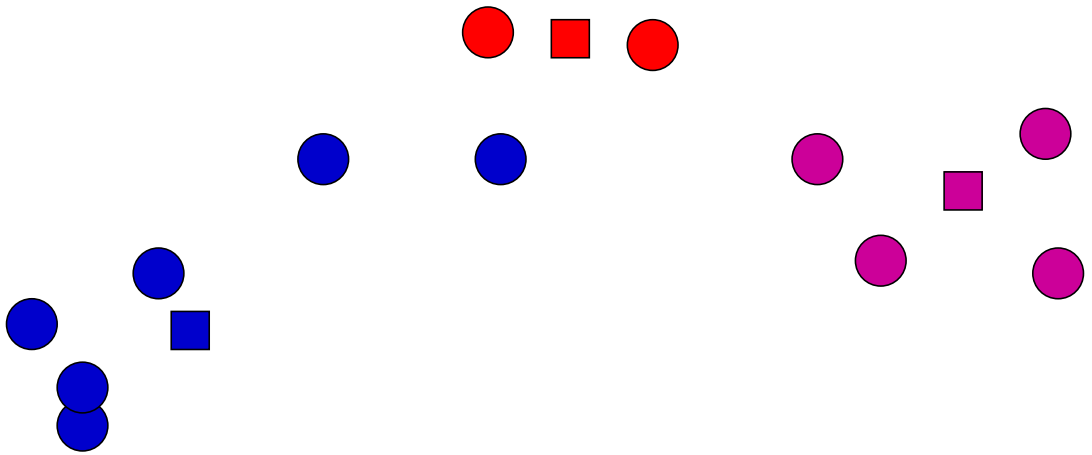
K-means: Initialize centers randomly



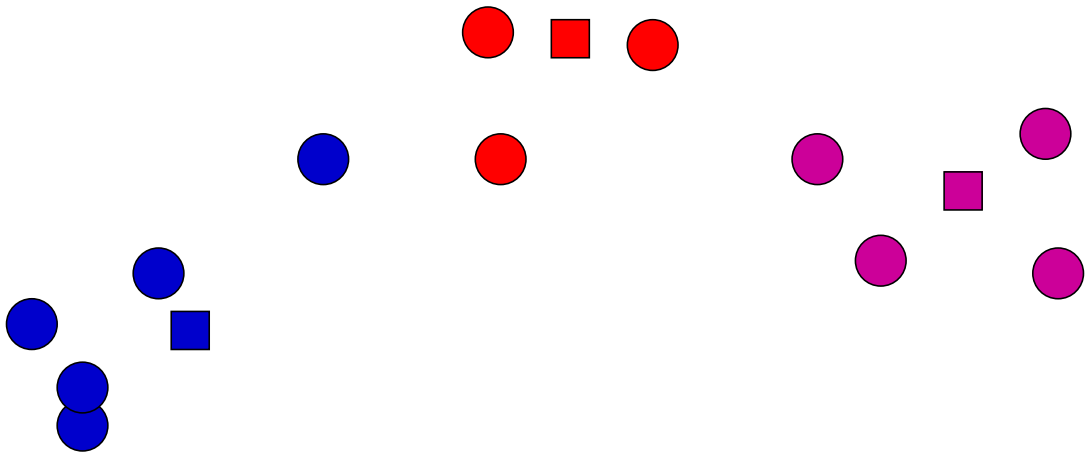
K-means: assign points to nearest center



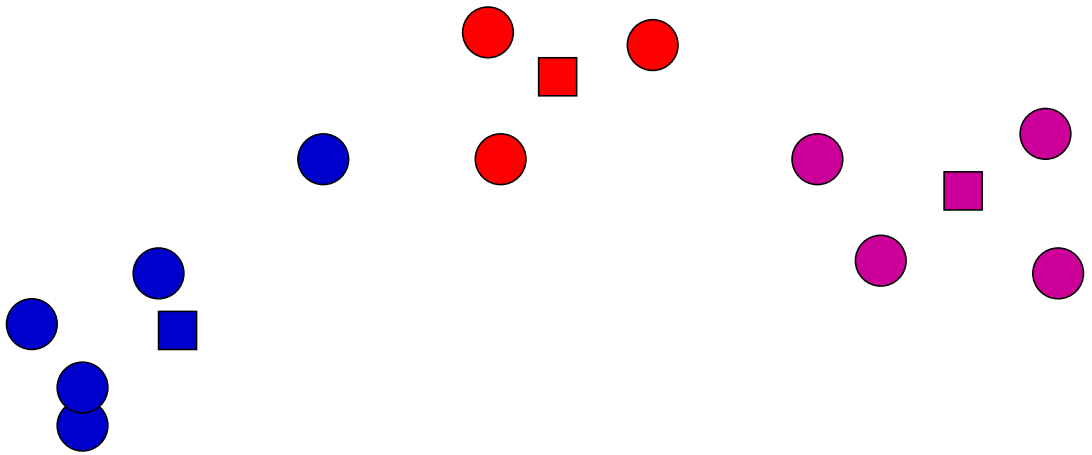
K-means: readjust centers



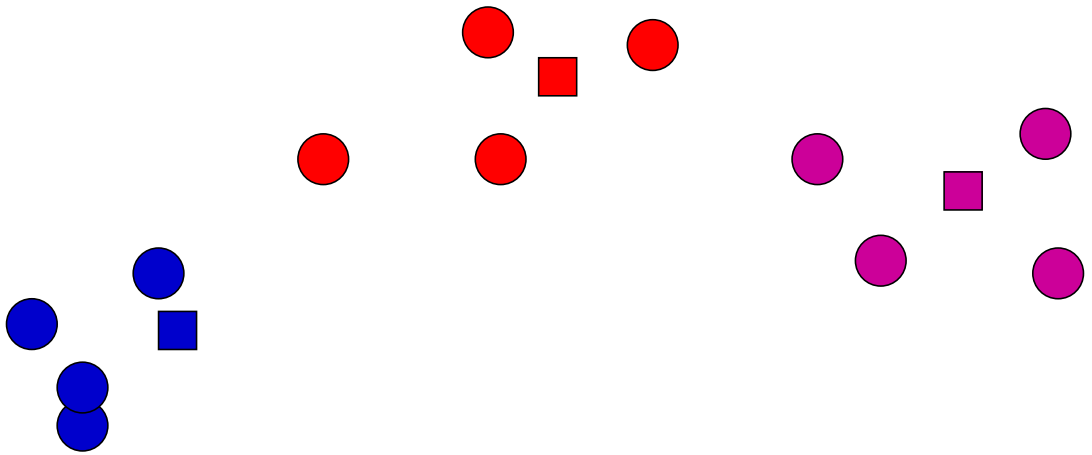
K-means: assign points to nearest center



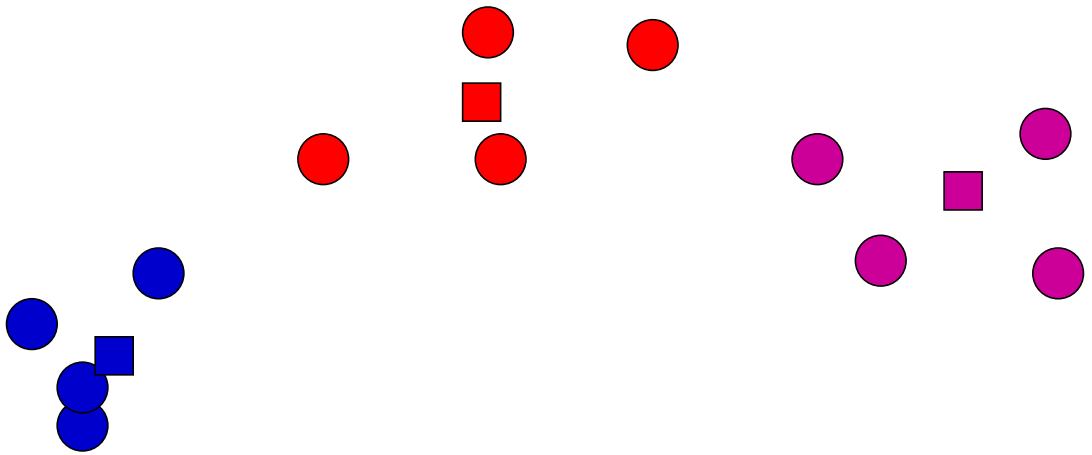
K-means: readjust centers



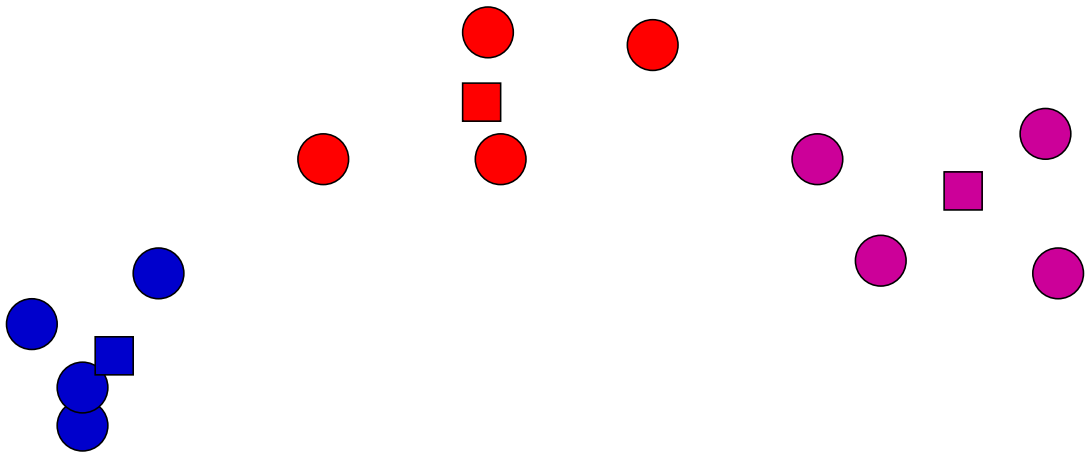
K-means: assign points to nearest center



K-means: readjust centers



K-means: assign points to nearest center



No changes: Done

For each data point $\mathbf{x}_\eta$, we introduce a corresponding set of binary indicator variables

$$r_{\eta k} \in \{0, 1\}$$

with

$$k = 1, ..., K$$

describing which of the K clusters the data point $\mathbf{x}_\eta$, is assigned to, so that if data point $\mathbf{x}_\eta$, is assigned to cluster k then $r_{\eta k} = 1$, and $r_{\eta j} = 0$ for $j \neq k$

This is known as the 1-of-K coding scheme.

$$r_{\eta k} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_\eta - \mathbf{c}_j\|^2 \\ 0 & \text{otrwise} \end{cases}$$

We can then define an objective function, sometimes called a distortion measure, given by

$$J = E = \sum_{\eta=1}^{N} \sum_{k=1}^{K} r_{\eta k} \cdot \|\mathbf{x}_\eta - \mathbf{c}_k\|^2 = \sum_{k=1}^{K} \sum_{x \in C_k} \left(d_2(\mathbf{x}, \mathbf{c}_k)\right)^2$$

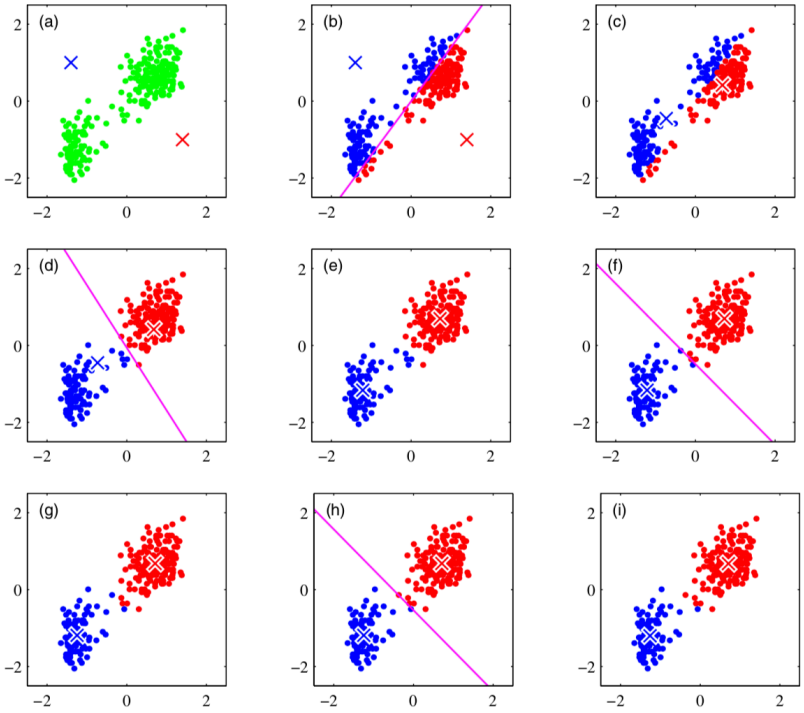
we want to minimize it

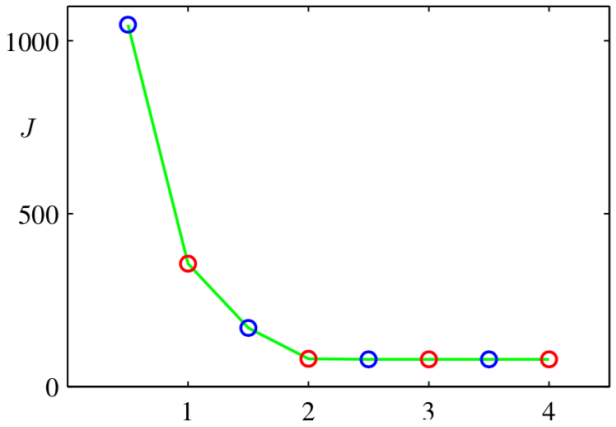
$$\frac{\partial J}{\partial \mathbf{c}_k} = -2 \cdot \sum_{\eta=1}^{N} r_{\eta k} \cdot (\mathbf{x} - \mathbf{c}_k)$$

$$\sum_{\eta=1}^{N} r_{\eta k} \cdot (\mathbf{x} - \mathbf{c}_k) = 0$$

$$\sum_{\eta=1}^{N} r_{\eta k} \cdot \mathbf{x} - \sum_{\eta=1}^{N} r_{\eta k} \cdot \mathbf{c}_k = 0$$

$$\mathbf{c}_k = \frac{\sum_{\eta=1}^{N} r_{\eta k} \cdot \mathbf{x}}{\sum_{\eta=1}^{N} r_{\eta k}} = \frac{1}{|C_k|} \cdot \sum_{x \in C_k} \mathbf{x}$$


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
2
0
−2
−2
0
2


1000
J
500
0
1
2
3
4

Standard K-means

Random initialisation of K centroids;
do
{
assign to each $\mathbf{x}_\eta$ in the dataset the nearest centroid $\mathbf{c}_k$ according to d_2;
compute all new centroids $\mathbf{c}_k = \frac{1}{|C_k|} \cdot \sum_{x \in C_k} \mathbf{x}$;

}
until ($|E_{new} - E_{old}| < \epsilon$ or number of iterations max iterations).

Sequential K-means

For large data sets, the adaptive K-means learning algorithm is given by with sequential update in which, for each data point

$$\mathbf{c}_k^{new} = \mathbf{c}_k^{old} + \eta_\eta \cdot (\mathbf{x}_\eta - \mathbf{c}_k^{old}) = \mathbf{c}_k^{old} + \frac{1}{|C_k^{old}| + 1} \cdot (\mathbf{x}_\eta - \mathbf{c}_k^{old})$$

where η_η is the learning rate parameter, which is typically made to decrease monotonically as more data points are considered and can be represented by $\frac{1}{|C_k^{old}|+1}$.

Sequential K-means

Random initialisation of K centroids;
do
{
chose $\mathbf{x}_\eta$ from the dataset;
determine the nearest centroid $\mathbf{c}_k$ according to d_2;
compute the new centroid $\mathbf{c}_k^{new} = \mathbf{c}_k^{old} + \frac{1}{|C_k^{old}|+1} \cdot (\mathbf{x}_\eta - \mathbf{c}_k^{old})$;
}
until ($|E_{new} - E_{old}| < \epsilon$ or number of iterations max iterations).

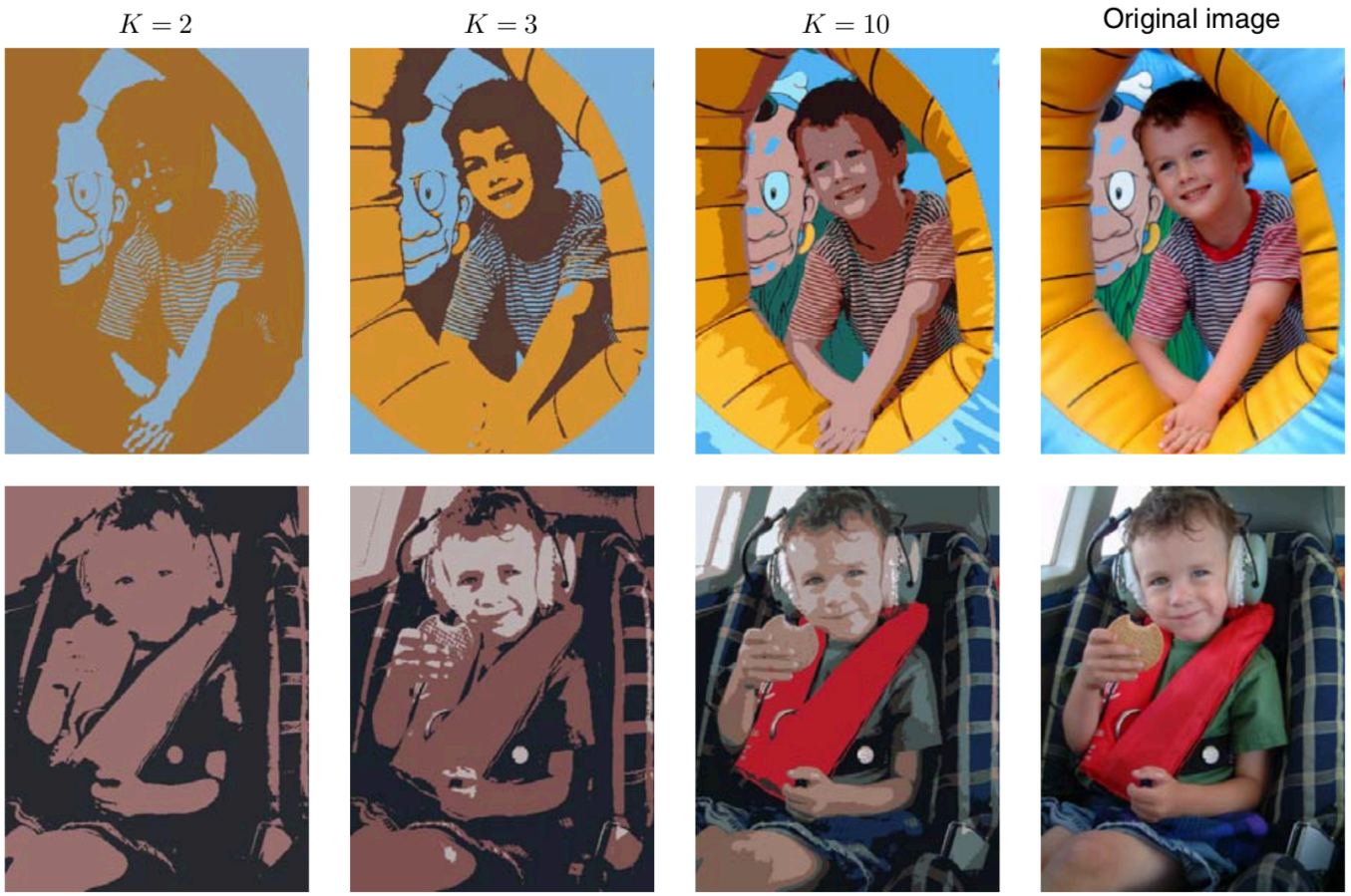
- K-means represents an unsupervised learning; it is an unsupervised classification because no predefined classes are present.

- One notable feature of the K-means algorithm is that at each iteration, every data point is assigned uniquely to one, and only one, of the clusters.

Color reduction

- K-means can be used for color reduction for RGB images.
- K would indicate the number of the reduced colors, and the dimension would be there for *R, G, B*. $x_i = R_i, G_i, B_i$ would correspond to the pixel at position i in a one dimensional array.
- Color segmentation represents a weak segmentation, in which the segmented parts (the same color) do not correspond necessarily to objects.

Color reduction



- How to chose *K* ?
  - You have to know your data!

- Repeated runs of *K*-means clustering on the same data can lead to quite different partition results
  - Why? Because we use random initialization


a)
b)

Adaptive Initialization

- Choose a maximum *radius* within every data point should have a cluster seed after completion of the initialization phase
- In a single sweep go through the data and assigns the cluster seeds according to the chosen r*adius*
  - A data point becomes a new cluster seed, if it is not covered by the spheres with the chosen *radius* of the other already assigned seeds
  - K-MAI clustering (Wichert et al. 2003)

Validity: Relative Criteria

- Here the basic idea is the evaluation of a clustering structure by comparing it to other clustering schemes, resulting by the *same* algorithm but with *different parameter values*

- There are two criteria proposed for clustering evaluation and selection of an optimal clustering scheme (Berry and Linoff, 1996)
  - Compactness, the members of each cluster should be as close to each other as possible. A common measure of compactness is the variance, which should be minimized
  - Separation, the clusters themselves should be widely spaced

Validity index

- Dunn index, a cluster validity index for *K*-means clustering proposed in Dunn (1974)

- Attempts to identify “compact and well separated clusters”

  - *Notation: k the number of clusters*

Dunn index

$$d(C_i, C_j) = \min_{\vec{x} \in C_i, \vec{y} \in C_j} d(\vec{x}, \vec{y})$$

$$diam(C_i) = \max_{\vec{x}, \vec{y} \in C_i} d(\vec{x}, \vec{y})$$

$$D_k = \min_{1 \le i \le k} \left\{ \min_{\substack{1 \le j \le k \\ i \ne j}} \left\{ \frac{d(C_i, C_j)}{\max_{1 \le l \le k} \{diam(C_l)\}} \right\} \right\}$$

- If the dataset contains compact and well-separated clusters, the *distance* between the clusters is expected to be *large* and the *diameter* of the clusters is expected to be *small*
- Large values of the index indicate the presence of compact and well-separated clusters

- The implications of the Dunn index are:

- Considerable amount of time required for its computation
- Sensitive to the presence of noise in datasets, since these are likely to increase the values of *diam(c)*

- The Davies-Bouldin (DB) index (1979)

$$d(C_i, C_j) = \min_{\vec{x} \in C_i, \vec{y} \in C_j} d(\vec{x}, \vec{y})$$

$$diam(C_i) = \max_{\vec{x}, \vec{y} \in C_i} d(\vec{x}, \vec{y})$$

$$DB_k = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \left\{ \frac{diam(C_i) + diam(C_j)}{d(C_i, C_j)} \right\}$$

- Small indexes correspond to good clusters, clusters are compact and their centers are far away
- The DB_k index exhibits no trends with respect to the number of clusters and thus we seek the minimum value of DB_k its plot versus the number of clusters

Mixture of Gaussians

Gaussian distribution or normal is defined over D dimensional space

$$p(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{(2 \cdot \pi)^{D/2}} \cdot \frac{1}{|\Sigma|^{1/2}} \cdot \exp\left(-\frac{1}{2} \cdot (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} \cdot (\mathbf{x} - \boldsymbol{\mu})\right)$$

where

- $\boldsymbol{\mu}$ is the D dimensional mean vector
- Σ is a $D \times D$ covariance matrix
- $|\Sigma|$ is the determinant of Σ

Gaussian Mixture Distribution

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$$

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \cdot \frac{1}{(2 \cdot \pi)^{D/2}} \cdot \frac{1}{|\Sigma_k|^{1/2}} \cdot \exp\left(-\frac{1}{2} \cdot (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \cdot (\mathbf{x} - \boldsymbol{\mu}_k)\right)$$

with

$$0 \leq \pi_k \leq 1$$

and

$$\sum_{k=1}^{K} \pi_k = 1$$

Let us introduce a K-dimensional binary random variable c having a 1-of-K representation in which a particular element c_k is equal to one and all other are equal to 0.

$$c_k \in \{0, 1\}, \qquad \sum_k c_k = 1$$

We define

$$p(\mathbf{x}, \mathbf{c}) = p(\mathbf{x}|\mathbf{c}) \cdot p(\mathbf{c})$$

We can write

$$p(c_k = 1) = \pi_k$$

and we can write

$$p(\mathbf{x}|c_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) = \frac{1}{(2 \cdot \pi)^{D/2}} \cdot \frac{1}{|\Sigma_k|^{1/2}} \cdot \exp\left(-\frac{1}{2} \cdot (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \cdot (\mathbf{x} - \boldsymbol{\mu}_k)\right)$$

Because c_k has a 1-of-K representation with $c_k \in \{0, 1\}$ we can write it to the power

$$p(\mathbf{c}) = \prod_{k=1}^{K} \pi_k^{c_k}$$

and we can write

$$p(\mathbf{x}|\mathbf{c}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)^{c_k}$$

For $\mathbf{x}$ there is a $\mathbf{c}$ and we are able to work with $p(\mathbf{x}, \mathbf{c})$!

$$p(\mathbf{x}) = \sum_{\mathbf{c}} p(\mathbf{x}, \mathbf{c}) = \sum_{\mathbf{c}} p(\mathbf{c}) \cdot p(\mathbf{x}|\mathbf{c}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$$

and

$$p(\mathbf{c}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{c}) \cdot p(\mathbf{c})}{p(\mathbf{x})}$$

We will now use $c_k = 1$ to denote the cluster k and $p(c_k = 1|\mathbf{x})$

$$p(c_k = 1|\mathbf{x}) = \frac{p(\mathbf{x}|c_k = 1) \cdot p(c_k = 1)}{p(\mathbf{x})}$$

$$p(c_k = 1|\mathbf{x}) = \frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \cdot p(c_k = 1)}{\sum_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \cdot p(c_k = 1)}$$

We will now use $c_k = 1$ to denote the cluster k and $p(c_k = 1|\mathbf{x})$

$$p(c_k = 1|\mathbf{x}) = \frac{p(\mathbf{x}|c_k = 1) \cdot p(c_k = 1)}{p(\mathbf{x})}$$

$$p(c_k = 1|\mathbf{x}) = \frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \cdot p(c_k = 1)}{\sum_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \cdot p(c_k = 1)}$$

We define $\gamma(c_k)$ to be equivalent to $p(c_k = 1|\mathbf{x})$

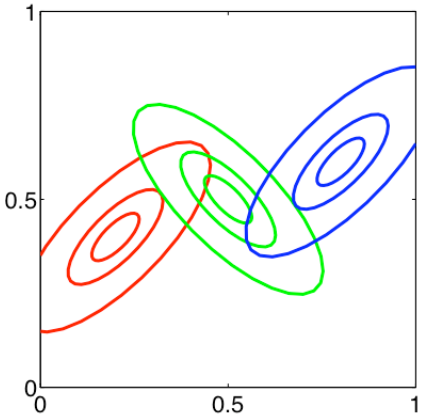
$$\gamma(c_k) \equiv p(c_k = 1|\mathbf{x}) = \frac{\pi_k \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)}$$

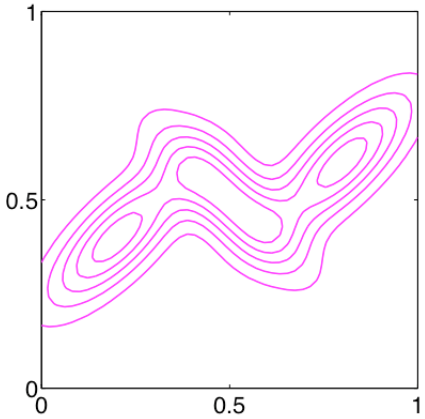
We define $\gamma(c_{\eta k})$ to be equivalent to $p(c_k = 1|\mathbf{x})_\eta$ for a certain pattern $\mathbf{x}_\eta$ with the index η

$$\gamma(c_{\eta k}) \equiv p(c_k = 1|\mathbf{x}_\eta) = \frac{\pi_k \cdot \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)}$$

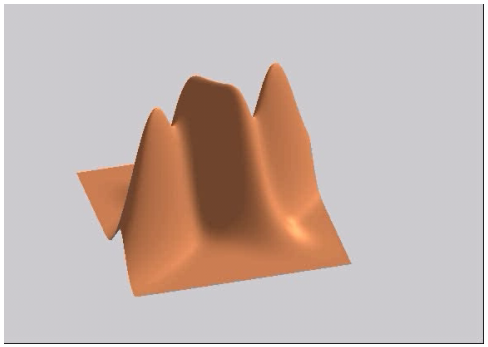
Example: Mixture of 3 Gaussians *k*

$p(\mathbf{x}|k)$

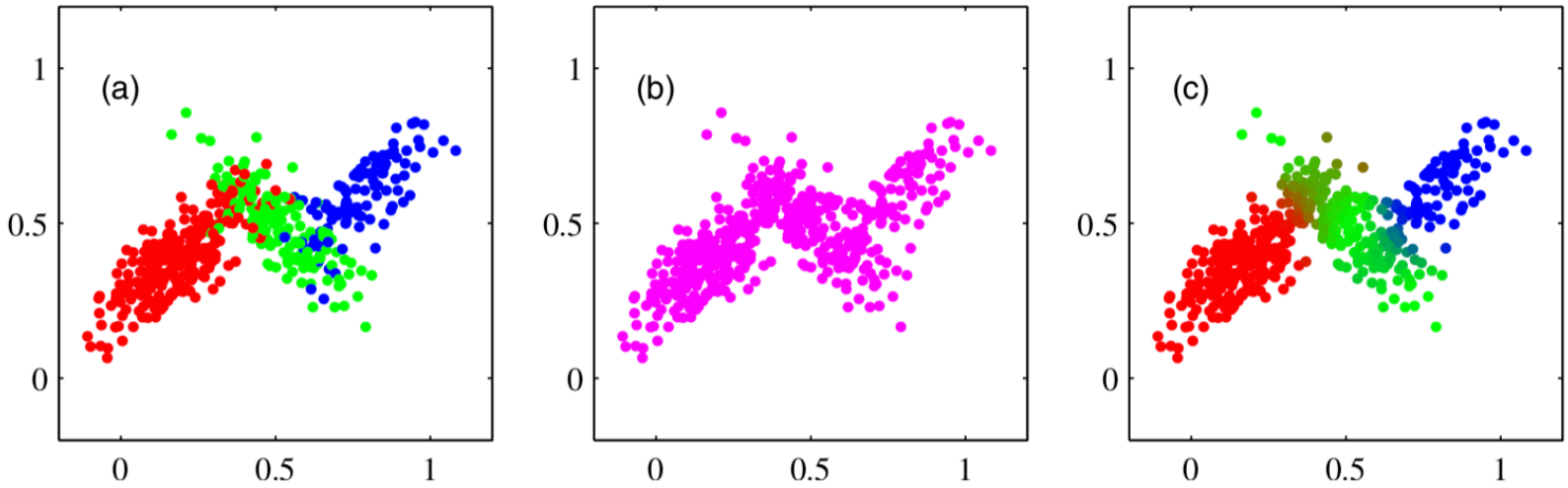




$p(\mathbf{x})$



$p(\mathbf{x})$



- (a) corresponding to the three components of the mixture, are depicted in red, green, and blue,
- (b) the corresponding samples from the marginal distribution ***p(x)***
- (c) The same samples in which the colours represent the value of the responsibilities

$$p(c_k = 1|\mathbf{x}) = \frac{p(\mathbf{x}|c_k = 1) \cdot p(c_k = 1)}{p(\mathbf{x})}$$

Maximum likelihood

Training set consists on N observations (sample)

$$X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\eta, \cdots, \mathbf{x}_N)$$

We can represent the dataset as a design matrix X of the dimension $N \times D$ as before

The log of the likelihood function is given by

$$\log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{\eta=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k) \right)$$

Significant problem associated with the maximum likelihood framework applied to Gaussian mixture models

Significant problem

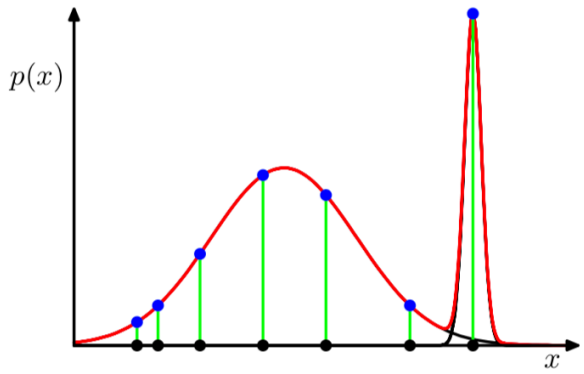
Consider a Gaussian mixture whose components have covariance matrices given by $\Sigma_k = \sigma_k^2 \cdot I$ with I being the identity matrix.

Suppose that one of the components of the mixture model, let us say the j^th component, has its mean $|\boldsymbol{\mu}_k$ exactly equal to one of the data points

$$\boldsymbol{\mu}_k = \mathbf{x}_\eta$$

$$\mathcal{N}(\mathbf{x}_\eta | \mathbf{x}_\eta, \sigma_j^2 \cdot I) = \frac{1}{(2 \cdot \pi)^{1/2}} \cdot \frac{1}{\sigma_j}$$

For $\sigma_j \to 0$ the term goes to infinity



EM for Gaussian mixtures

We maximise $\log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\mu}_k$, then Σ_k and π_k

$$\frac{\partial \log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k,} = \sum_{\eta=1}^{N} \frac{1}{\partial \boldsymbol{\mu}_k,} \log \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k) \right)$$

$$\frac{\partial \log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k,} = \sum_{\eta=1}^{N} \frac{1}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k)} \cdot \frac{1}{\partial \boldsymbol{\mu}_k} \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k) \right)$$

Since the derivative of exponential is exponential again...

$$\frac{\partial \log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k,} = \sum_{\eta=1}^{N} \frac{\pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k)} \cdot \frac{1}{\partial \boldsymbol{\mu}_k} \left(-\frac{1}{2} \cdot (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \cdot (\mathbf{x} - \boldsymbol{\mu}_k) \right)$$

EM for Gaussian mixtures

and we get

$$\frac{\partial \log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k,} = \sum_{\eta=1}^{N} \frac{\pi_k \cdot \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)} \cdot \Sigma_k^{-1} \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k)$$

$$0 = \sum_{\eta=1}^{N} \frac{\pi_k \cdot \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)} \cdot \Sigma_k^{-1} \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k)$$

$$0 = \sum_{\eta=1}^{N} \gamma(c_{\eta k}) \cdot \Sigma_k^{-1} \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k)$$

Multiplying with Σ_k we get

$$0 = \sum_{\eta=1}^{N} \gamma(c_{\eta k}) \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k)$$

EM for Gaussian mixtures

Multiplying with Σ_k we get

$$0 = \sum_{\eta=1}^{N} \gamma(c_{\eta k}) \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k)$$

$$\boldsymbol{\mu}_k \cdot \left(\sum_{\eta=1}^{N} \gamma(c_{\eta k}) \right) = \sum_{\eta=1}^{N} \gamma(c_{\eta k}) \cdot \mathbf{x}_\eta$$

with

$$N_k = \sum_{\eta=1}^{N} \gamma(c_{\eta k})$$

we get

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \cdot \sum_{\eta=1}^{N} \gamma(c_{\eta k}) \cdot \mathbf{x}_\eta$$

EM for Gaussian mixtures

We maximise $\log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to Σ_k

$$\frac{\partial \log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \Sigma_k,} = \sum_{\eta=1}^{N} \frac{1}{\partial \Sigma_k,} \log \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k) \right) = 0$$

we get

$$\Sigma_k = \frac{1}{N_k} \cdot \sum_{\eta=1}^{N} \gamma(c_{\eta k}) \cdot (\mathbf{x} - \boldsymbol{\mu}_k) \cdot (\mathbf{x} - \boldsymbol{\mu}_k)^T$$

with

$$N_k = \sum_{\eta=1}^{N} \gamma(c_{\eta k})$$

EM for Gaussian mixtures

We maximise $\log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to π_k with the constraint

$$\sum_{k=1}^{K} \pi_k = 1$$

This can be achieved using a Lagrange multiplier and maximizing the following quantity

$$\log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

$$\frac{\partial \log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)}{\partial \pi_k} = 0$$

which gives

$$0 = \sum_{\eta=1}^{N} \frac{\mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)} + \lambda$$

EM for Gaussian mixtures

multiplying by π_k we get

$$0 = \sum_{\eta=1}^{N} \frac{\pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k)} + \pi_k \cdot \lambda$$

$$0 = \sum_{\eta=1}^{N} \gamma(c_{\eta k}) + \pi_k \cdot \lambda$$

$$0 = N_k + \pi_k \cdot \lambda$$

summing over k

$$0 = \sum_{k=1}^{K} (N_k + \pi_k \cdot \lambda)$$

$$-\sum_{k=1}^{K} N_k = \sum_{k=1}^{K} \pi_k \cdot \lambda$$

$$-N = \lambda$$

EM for Gaussian mixtures

$$0 = N_k + \pi_k \cdot \lambda$$

summing over k

$$0 = \sum_{k=1}^{K} (N_k + \pi_k \cdot \lambda)$$

$$-\sum_{k=1}^{K} N_k = \sum_{k=1}^{K} \pi_k \cdot \lambda$$

$$-N = \lambda$$

and we get

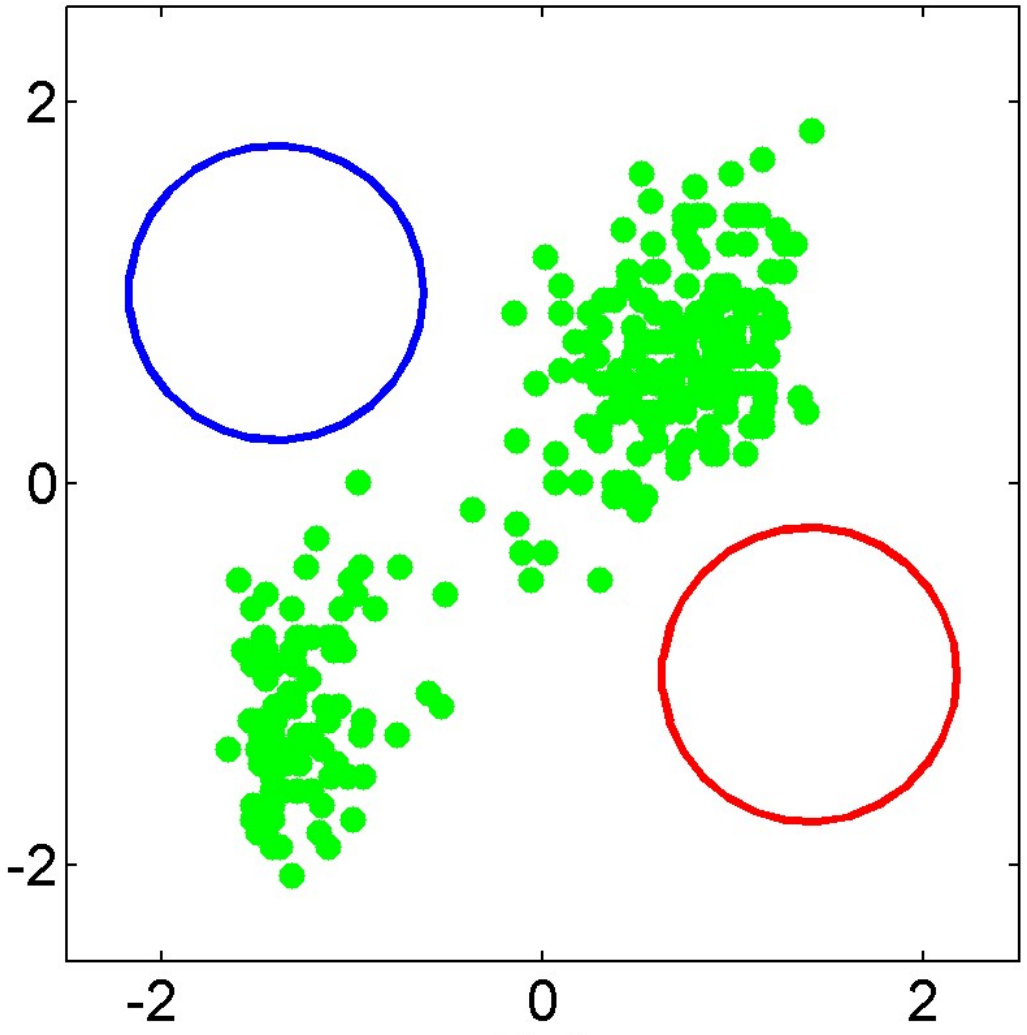
$$0 = N_k + \pi_k \cdot (-N)$$

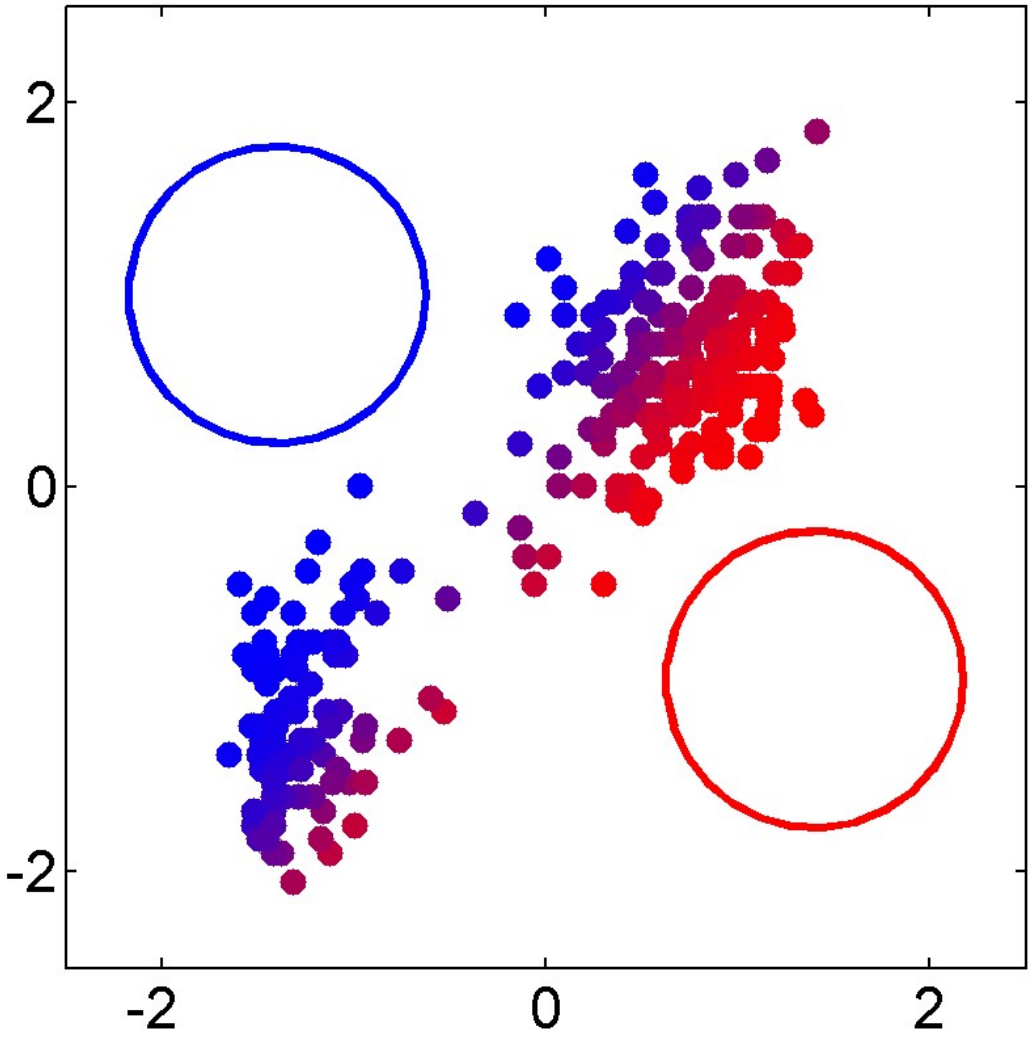
$$\pi_k = \frac{N_k}{N}$$

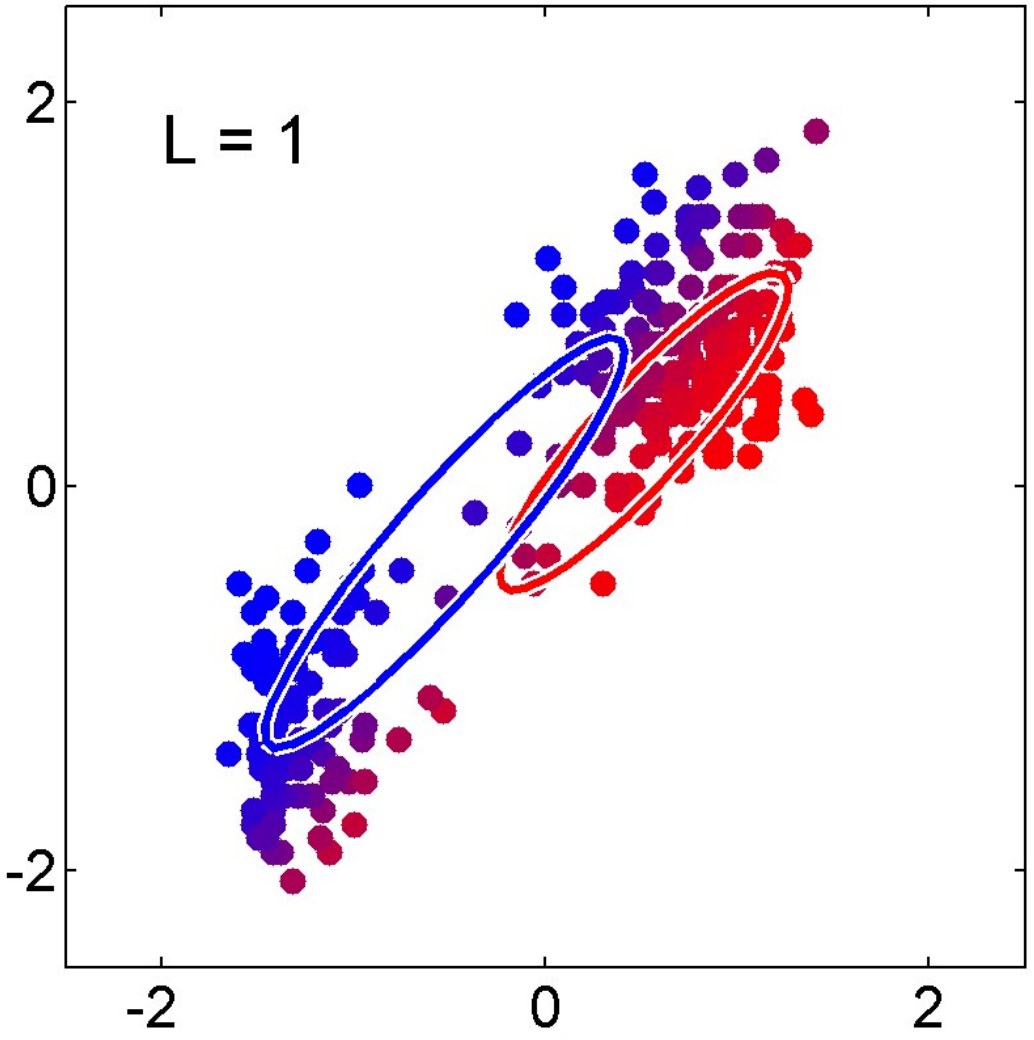
Algorithm: EM for Gaussian mixtures

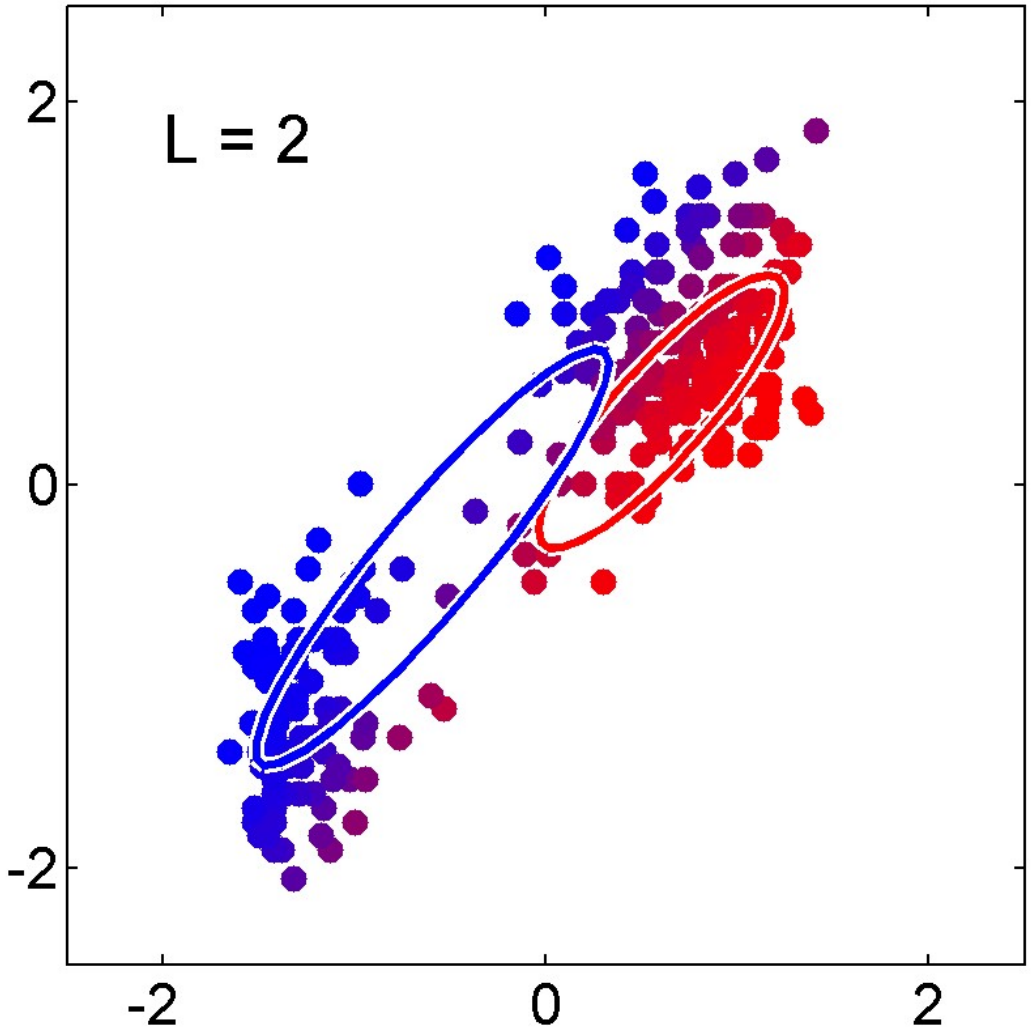
Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).
Training set consists on N observations (sample)

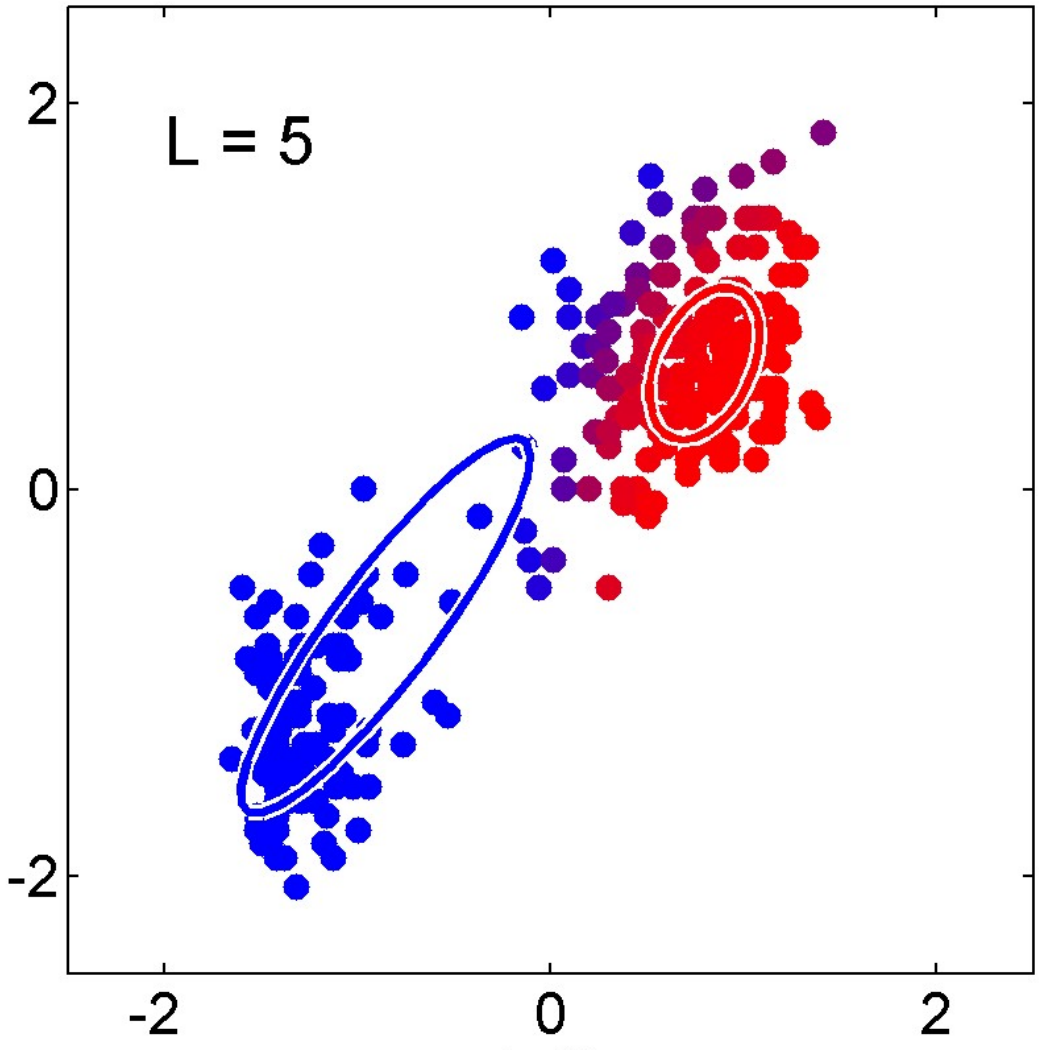
$$X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\eta, \cdots, \mathbf{x}_N)$$

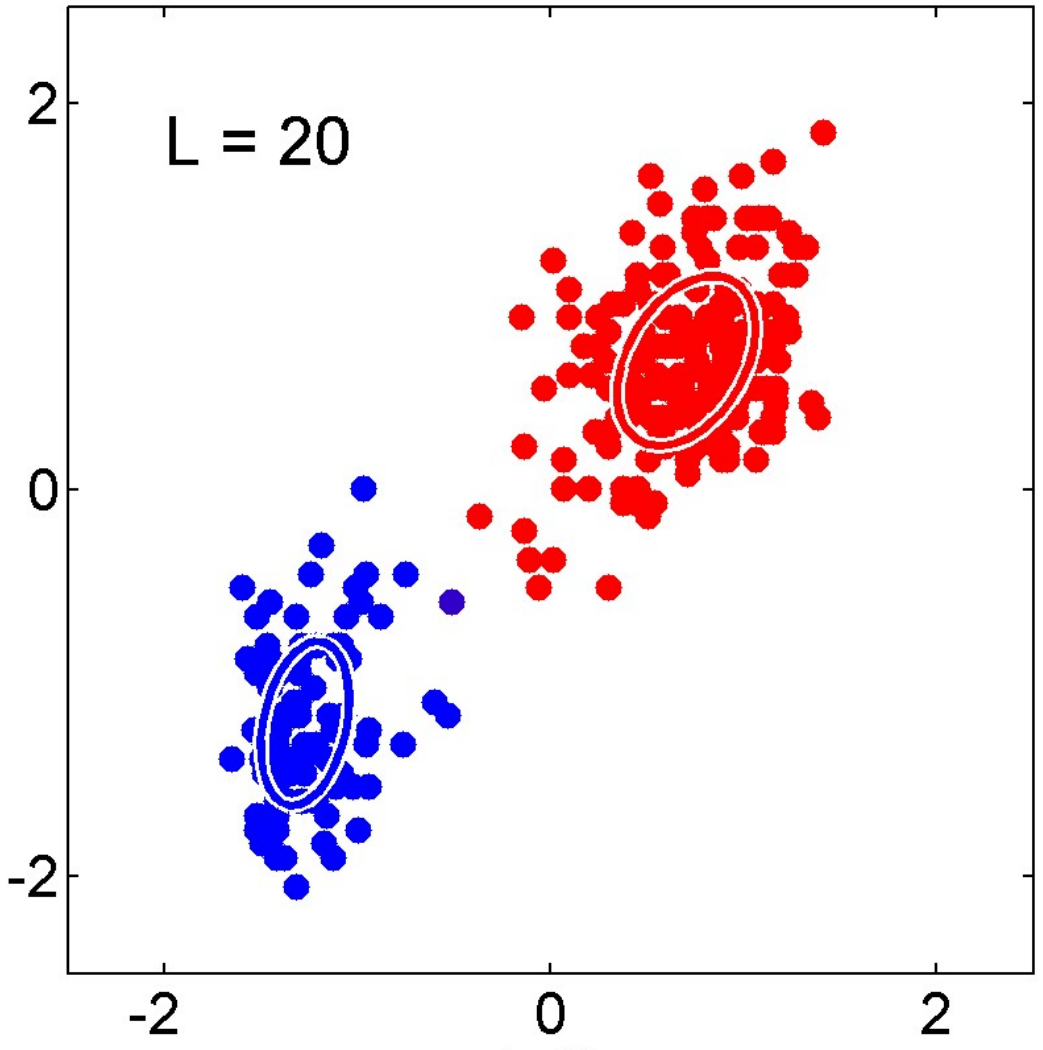

2
0
-2
-2
0
2


2
0
-2
-2
0
2


L = 1
2
0
-2
-2
0
2


L = 2
2
0
-2
-2
0
2


L = 5
2
0
-2
-2
0
2


L = 20
2
0
-2
-2
0
2

Algorithm: EM for Gaussian mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).
Training set consists on N observations (sample)

$$X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\eta, \cdots, \mathbf{x}_N)$$

EM for Gaussian mixtures

1. Initialisation:

   Chose K, number of clusters. Then initialise

   - the means $\boldsymbol{\mu}_k$ (centres of clusters, random data point or random value)
   - Σ_k covariances (shape of clusters, usually we can start with identity matrix I)
   - $\pi_k = p(c_k = 1)$ mixing coefficients (prior, importance of clusters, usually we can start with the value $\frac{1}{K}$, each cluster has the same importance

2. **E-Step** (Expectation):

   Compute for each data point η and each cluster k

$$\gamma(c_{\eta k}) = p(c_k = 1|\mathbf{x}_\eta) = \frac{\pi_k \cdot \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)}$$

   c_k is the mixture, cluster k

$$p(c_k = 1|\mathbf{x}_\eta) = \frac{p(c_k = 1) \cdot p(\mathbf{x}_\eta|c_k = 1)}{p(\mathbf{x}_\eta)}$$

   Usually we can compute the likelihood

$$p(c_k = 1, \mathbf{x}_\eta) = \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta|\boldsymbol{\mu}_k, \Sigma_k)$$

   with

$$p(\mathbf{x}_\eta|c_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) = \frac{1}{(2 \cdot \pi)^{D/2}} \cdot \frac{1}{|\Sigma_k|^{1/2}} \cdot \exp\left(-\frac{1}{2} \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k)\right)$$

EM for Gaussian mixtures

$$p(c_k = 1, \mathbf{x}_\eta) = \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k)$$

with

$$p(\mathbf{x}_\eta | c_k = 1) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_k) = \frac{1}{(2 \cdot \pi)^{D/2}} \cdot \frac{1}{|\Sigma_k|^{1/2}} \cdot \exp\left(-\frac{1}{2} \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k) \right)$$

and after it

$$p(\mathbf{x}_\eta) = \sum_{k=1}^{K} p(c_k = 1, \mathbf{x}_\eta) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k)$$

and normalize

$$\gamma(c_{\eta k}) = p(c_k = 1 | \mathbf{x}_\eta) = \frac{p(c_k = 1, \mathbf{x}_\eta)}{p(\mathbf{x}_\eta)}$$

3. **M-Step** (Maximization):

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \cdot \sum_{\eta=1}^{N} \gamma(c_{\eta k}) \cdot \mathbf{x}_\eta$$

$$\Sigma_k = \frac{1}{N_k} \cdot \sum_{\eta=1}^{N} \gamma(c_{\eta k}) \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k) \cdot (\mathbf{x}_\eta - \boldsymbol{\mu}_k)^T$$

$$\pi_k = p(c_k = 1) = \frac{N_k}{N}$$

with

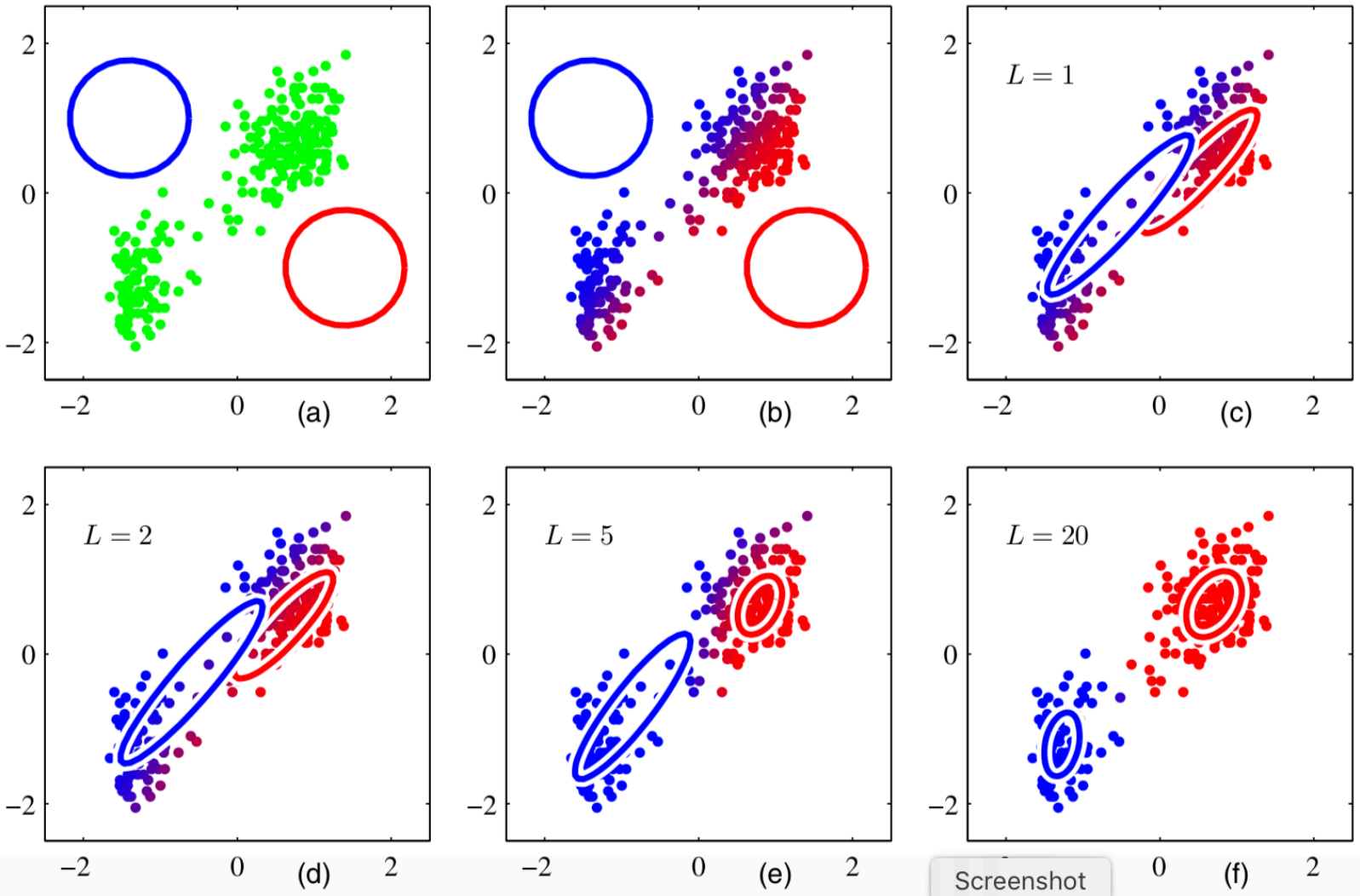
$$N_k = \sum_{\eta=1}^{N} \gamma(c_{\eta k})$$

EM for Gaussian mixtures

4. Evaluate the log likelihood

$$\log p(X|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{\eta=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}_\eta | \boldsymbol{\mu}_k, \Sigma_k) \right)$$

and check for convergence of either the parameters or the log likelihood. the convergence criterion is not satisfied or below number of iterations max iterations, return to step 2.



Screenshot

EM and K-means Clustering

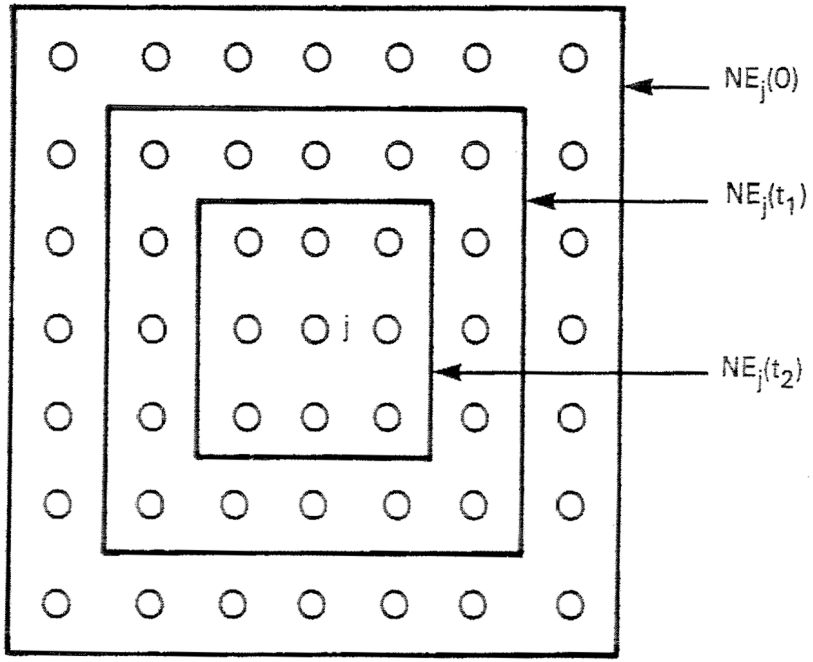
- Comparison of the K-means algorithm with the EM algorithm for Gaussian mixtures shows that there is a close similarity.
- The K-means algorithm performs a hard assignment of data points to clusters, in which each data point is associated uniquely with one cluster.
- The EM algorithm makes a soft assignment based on the posterior probabilities.
- In K-means the shape of the cluster is described by Euclidean distance function

Kohonen Self Organizing Maps

- Unsupervised learning
  - Clustering
  - related to k-Means batch modus
  - Labeling, supervised
- Perform a topologically ordered mapping from high dimensional space onto two-dimensional space
  - Dimension reduction
- The centroids (units) are arranged in a layer (two dimensional space), units physically near each other in a two-dimensional space respond to similar input

- $0 < \alpha(t) < 1$ is a monotonically decreasing scalar function
- *NE(t)* is a neighborhood function is decreasing with time *t*
- The topology of the map is defined by *NE(t)*
  - The dimension of the map is smaller (equal) then the dimension of the data space
  - Usually the dimension of a map is two
- For tow dimensional map the number of the centroids should have a integer valued square root
  - a good value to start is around 10^2 centroids

Neighborhood on the map



SOM Learning (Unsupervised)

Initialization of center vectors ***m***; t=0;

do

{

chose $\boldsymbol{x}_i$ from the dataset

$\boldsymbol{m}_c$ nearest reference vector according to d_2

For all $\boldsymbol{m}_r$ near $\boldsymbol{m}_c$ on the map

$$m_r(t+1) = m_r(t) + \alpha(t)[x_i(t) - m_r(t)] \qquad for \qquad r \in NE_C(t)$$

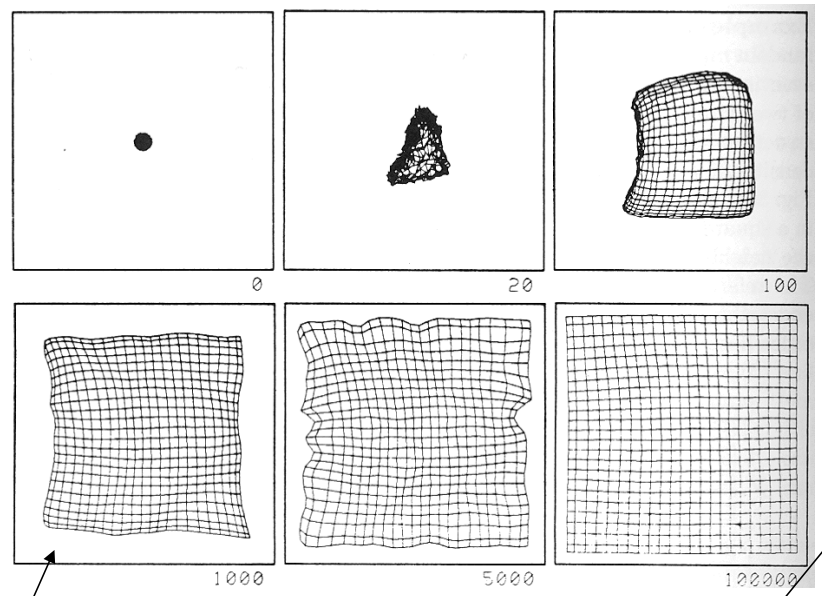
t++;

}

until number of iterations *t max_iterations*

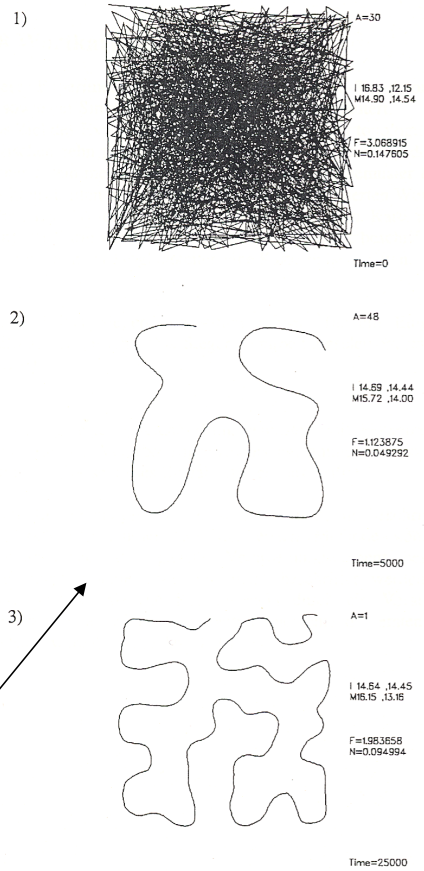
Ordering process of 2 dim data

random 2 dim points



2-dim map

1-dim map



Supervised labeling

- The network can be labeled in two ways

- (A) For each known class represented by a vector the closest centroid is searched and labeled accordingly
- (B) For every centroid we test to which known class represented by a vector it is closest

- Example of labeling of 10 classes, 0,..,9
- 10*10 centroids
- 2-dim map

(A)

#	0	#	#	2	#	#	#	#	
#	#	#	#	#	#	#	#	8	
#	#	#	#	#	#	#	#	#	
#	4	#	#	#	#	3	#	#	
#	#	#	#	#	#	#	#	#	
1	#	#	#	#	#	#	#	#	
#	#	#	#	#	#	#	#	6	
#	#	#	#	#	#	#	#	#	
5	#	#	#	9	#	#	7	#	

(B)

0	0	0	2	2	2	8	8	8	
0	0	4	2	2	2	8	8	8	
4	4	4	4	2	3	8	8	8	
4	4	4	4	3	3	3	3	8	
4	4	4	4	3	3	3	6	6	
1	1	4	4	3	3	3	6	6	
1	1	5	9	9	9	6	6	6	
5	5	5	9	9	9	7	7	7	
5	5	5	9	9	9	7	7	7	

Animal example

Table 3.4. Animal names and their attributes

		dove	hen	duck	goose	owl	hawk	eagle	fox	dog	wolf	cat	tiger	lion	horse	zebra	cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

"birds" occupy the left part of the lattice, "hunters" such as "tiger", "lion" and "cat" are clustered toward the right, and more "peaceful" species such as "zebra", "horse" and "cow" are situated in the upper middle. Within each cluster, a further grouping according to similarity is discernible.

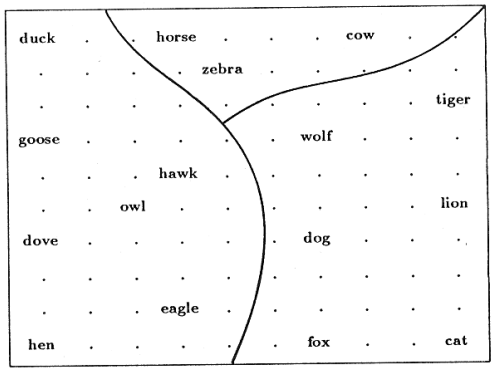
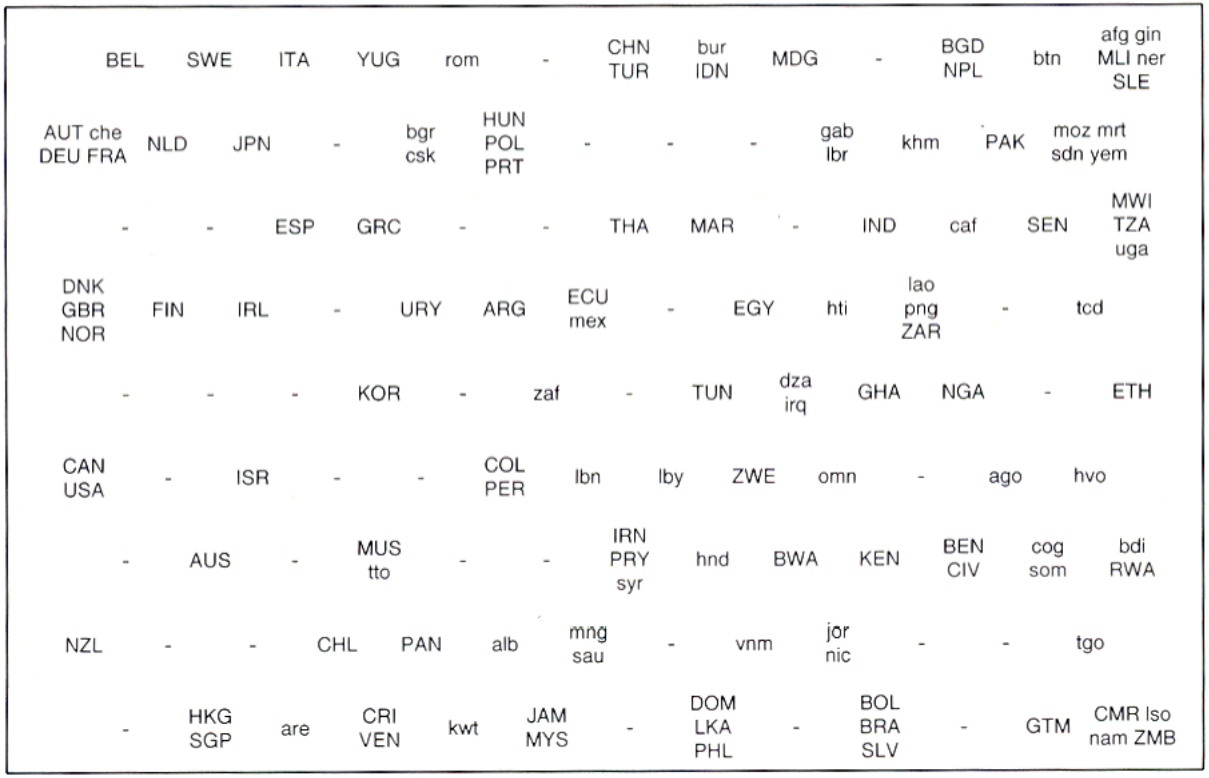
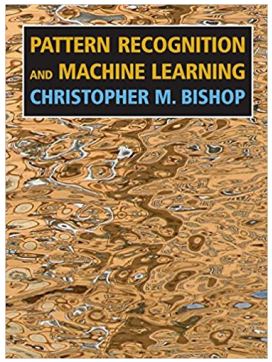


Fig. 3.22. After the network had been trained with inputs describing attribute sets from Table 3.4, the map was calibrated by the columns of Table 3.4 and labeled correspondingly. A grouping according to similarity has emerged

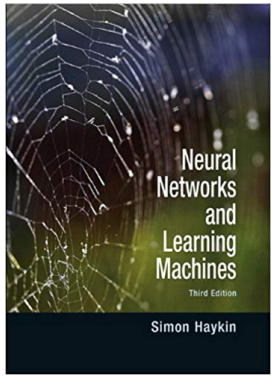
Poverty map of countries



Literature

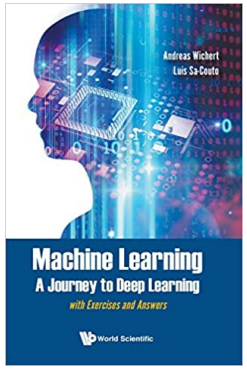


- Christopher M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer 2006
  - Chapter 9



- Simon O. Haykin, Neural Networks and Learning Machine, (3rd Edition), Pearson 2008
  - Chapter 9

Literature



- Machine Learning - A Journey to Deep Learning, A. Wichert, Luis Sa-Couto, World Scientific, 2021
  - Chapter 9