Sensor Signal Processing

Prof. Dr.-Ing. Andreas König
Lehrstuhl Integrierte Sensorsysteme

FB Elektrotechnik und Informationstechnik
Technische Universität Kaiserslautern

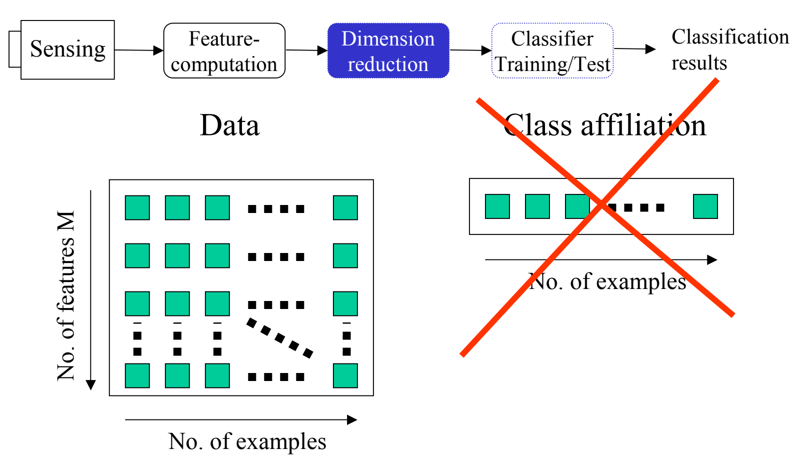
Fall Semester 2005

Course Contents

Chapter Contents

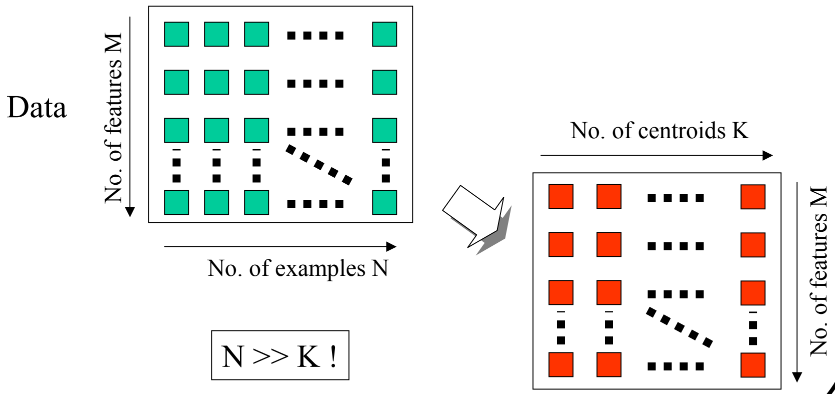
Motivation

- After signal processing and feature computation further condensing or compression of the data is desirable
- Unsupervised clustering techniques can serve for that purpose



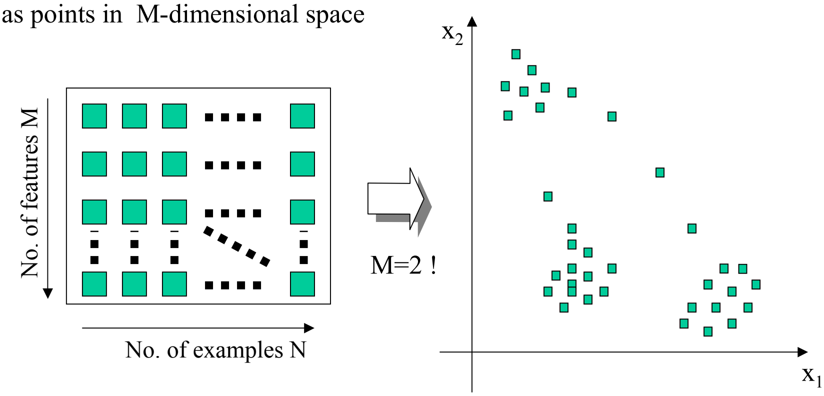
Motivation

- There is a close relation between data analysis and compression based on clustering techniques
- The purpose is to represent the data by a reduced number of prototypes or cluster centers (centroids)



Motivation

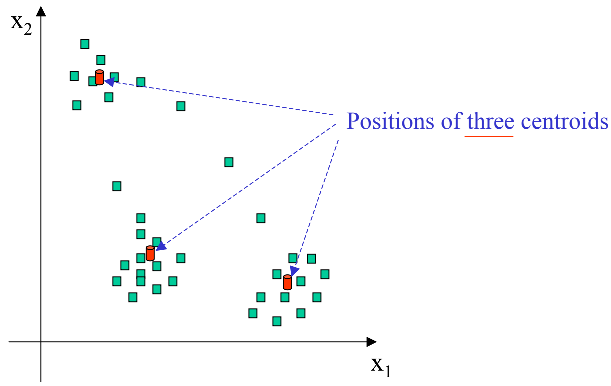
- According to a chosen metric, e.g., Euclidean distance or Mahalanobis distance, clustering takes place in feature space
- Feature space representation relates to the interpretation of feature vectors as points in M-dimensional space



- In case of 2D-representation (M=2) a standard scatter plot shows the feature space

Motivation

- Proximity in feature space expresses similarity and, thus, relations and structure in the data
- This can be exploited for a compact representation of the original data set in the sense of data compression of for data analysis

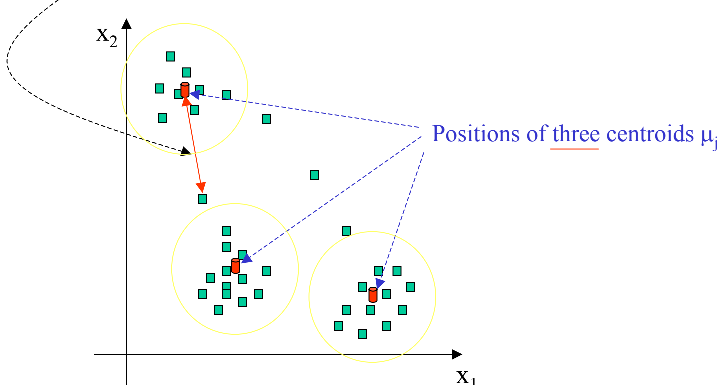


Sensor Signal Processing
Cluster Analysis

Motivation

- The notion of proximity in feature space depends on the chosen metric
- The Euclidean distance implies circular cluster shapes:

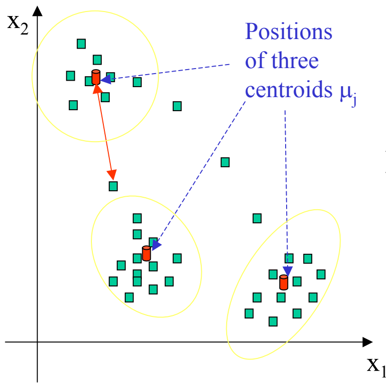
$$d_{ij} = \sqrt{\sum_{m=0}^{M-1} \left(x_{im} - \mu_{jm} \right)^2} \qquad (4.1)$$



Motivation

- The Mahalanobis distance incorporates covariance information and implies elipsoidal cluster shapes:

$$d_{ij} = \left(\vec{x}_i - \vec{\mu}_j\right)^T \Sigma_j^{-1} \left(\vec{x}_i - \vec{\mu}_j\right) \tag{4.2}$$



$$\Sigma_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \left(\vec{x}_i - \vec{\mu}_j\right)\left(\vec{x}_i - \vec{\mu}_j\right)^T \tag{4.3}$$

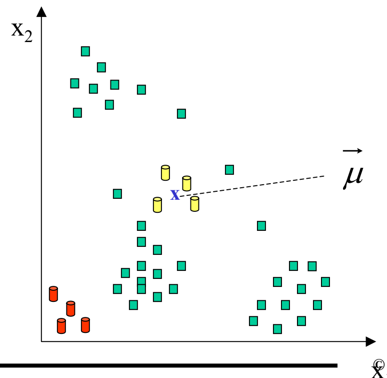
$$\vec{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \vec{x}_i \tag{4.4}$$

Motivation

- Clustering algorithms can be grouped according to certain aspects or characteristics [s. e.g., 3]
  - ✓ Static or dynamic centroid allocation
  - ✓ Initialization of centroids
  - ✓ Flat/hierarchical clustering
  - ✓ Growing/pruning/alternating approaches
  - ✓ Underlying optimization strategy
  - ✓ Crispness / fuzzyness of cluster membership

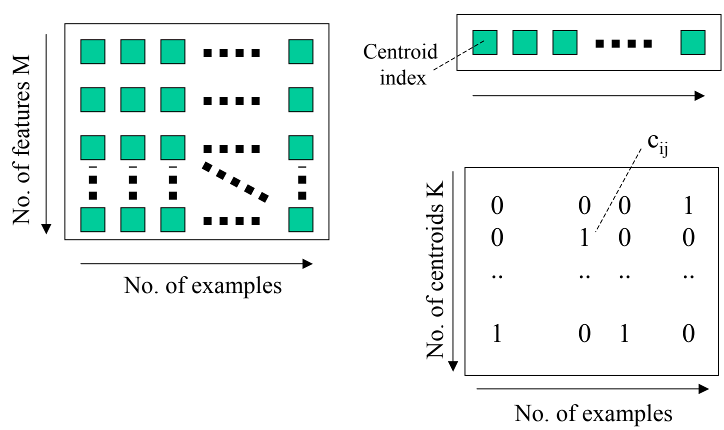
Conventional clustering

- A straight forward clustering technique was introduced by the authors Linde, Buzo, and Gray, denoted as LBG algorithm
- This iterative method starts with the choice of an arbitrary but fixed number of centroids
- The centroids have to be initialized at the beginning:
  - Random initialization by small, random values
  - Selection of a sequence/random collection of data samples



Conventional clustering

- For the process, each data or feature vector is enhanced by an additional entry c_i for the cluster affiliation expressed by the centroid index
- Alternatively, a membership matrix expresses the same with 0,1 for the c_{ij}



Conventional clustering

- The algorithm proceeds as follows:
  1. Initialization
  2. Compute cluster affiliation for every data vector:

$$c_{ij} = \begin{cases} 1 & if \quad d_{ij} = \min_k \left(\sum_{m=0}^{M-1} \left(x_{im} - \mu_{km} \right)^2 \right) \\ 0 & else \end{cases} \tag{4.5}$$

  3. Compute new centroids:

$$\overrightarrow{\mu_j} = \frac{1}{N_j} \sum_{i=0}^{N-1} c_{ij} \vec{x}_i \quad with \quad N_j = \sum_{i=0}^{N-1} c_{ij} \tag{4.6}$$

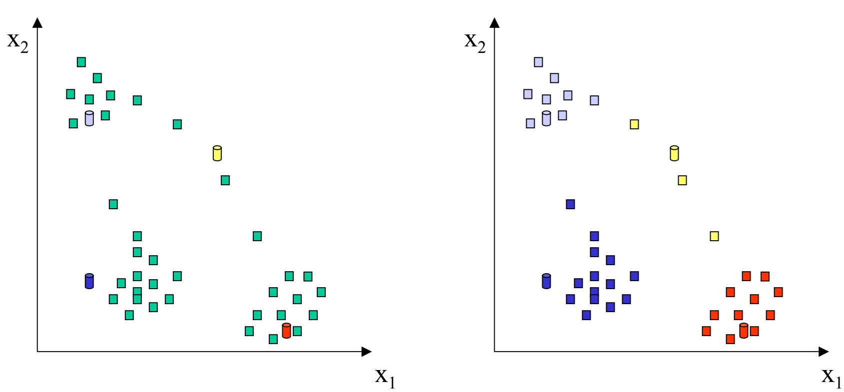
  4. Compute the achieved quantization error:

$$E(k) = \sum_{i=0}^{N-1} \sum_{j=0}^{K-1} \sum_{m=0}^{M-1} c_{ij} \left(x_{im} - \mu_{jm} \right)^2 \tag{4.7}$$

  4. If k<maxstep and E(k)>ErrThresh k++; Goto 2 else Break

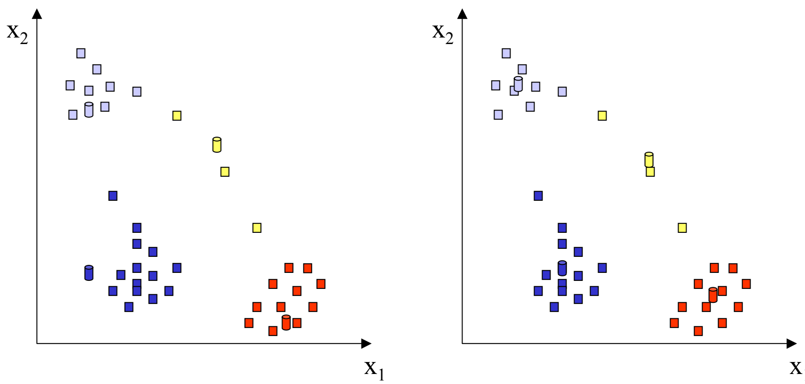
Conventional clustering

- Illustration of step 2



Conventional clustering

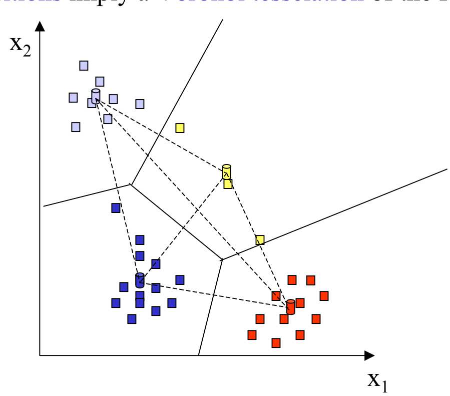
- Illustration of step 3



- In the regarded static case, it is not guaranteed that all centroids will be properly employed (dead codebook vectors)
- Extensions of the basic algorithm to exploit all centroids

Conventional clustering

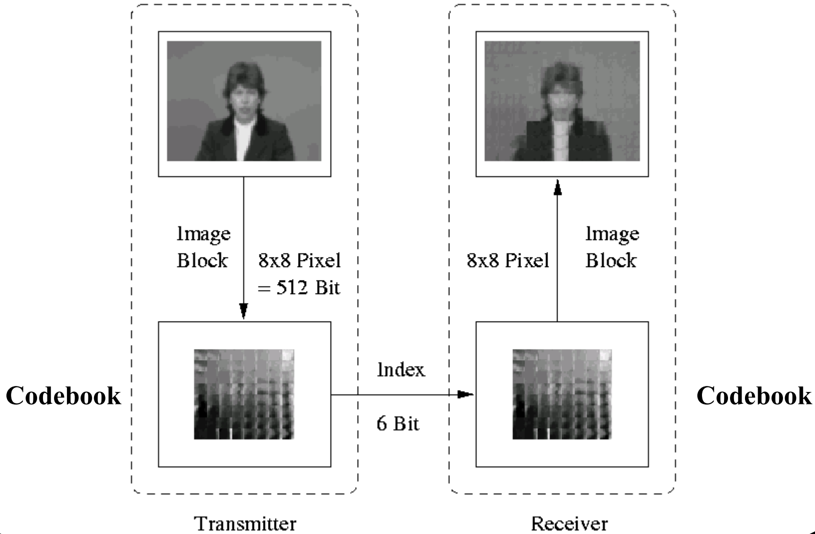
- The previously introduced procedure is also known as the k-means or c-means clustering algorithm [3]
- The centroid positions imply a Voronoi tesselation of the feature space



- Alternatives to clustering criterion J and optimization of J [3]

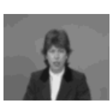
Vector quantization

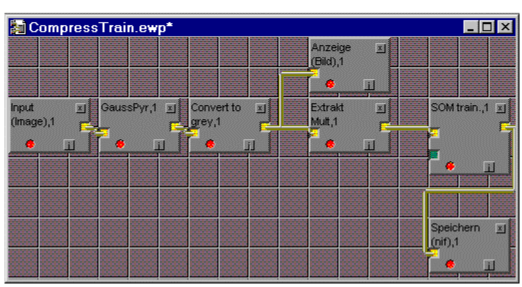
- The quantization property of clustering techniques is best displayed for data compression, e.g., **vector quantization** in image and signal coding:

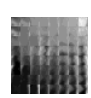


Vector quantization

- Codebook generation and vector quantization (*using SOM*):

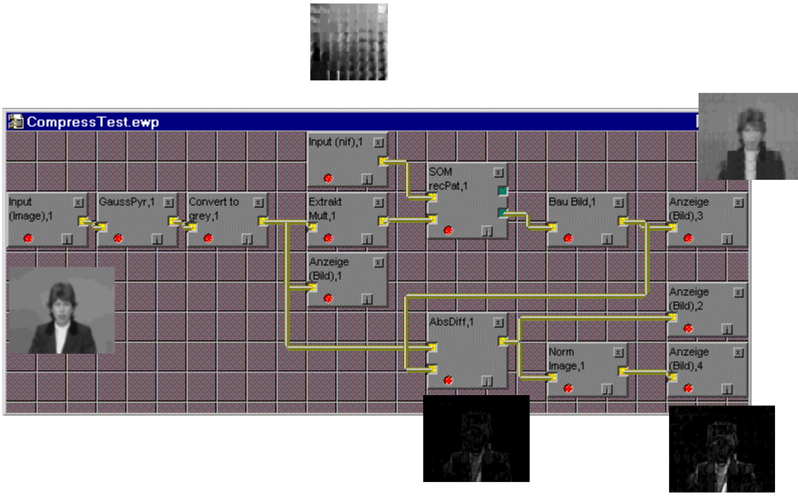






Vector quantization

- Codebook generation and vector quantization (*using SOM*):



Vector quantization

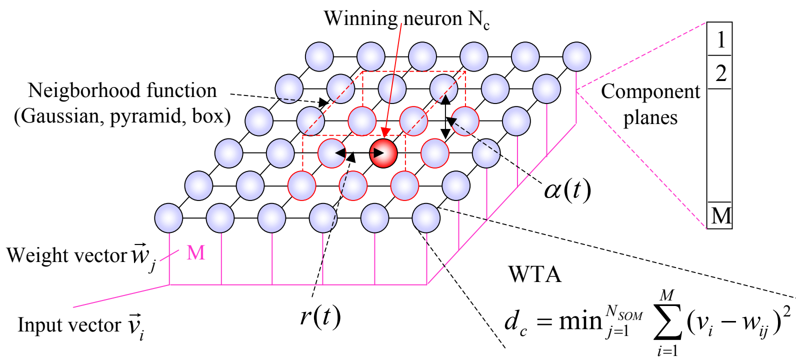
- Results for different frames:

Codebook generation

Codebook adaptation

Self-organizing feature map

- The Self-Organizing feature Map (SOM), introduced by Teuvo Kohonen is the probably most well-known and applied neural network
- The SOM was derived from physiological evidence observed in the somato sensory cortex

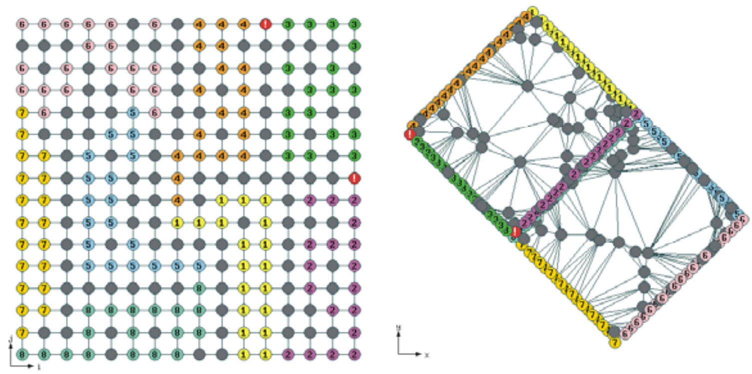


Self-organizing feature map

- The SOM features the properties of data quantization, probability density approximation, topology preserving dimensionality reducing mapping
- Typically, 1D- or 2D-SOM neuron grids are employed (3D in Robotics)
- SOM learning in its common technical implementation:
  1. Random initialization of neuron weight vectors $\vec{w}_j$
  2. Iterative presentation of stimuli vectors $\vec{v}_i$ and computation of the winner neuron N_c
  $$d_c = \min_{j=1}^{N_{SOM}} \sum_{i=1}^{M} (v_i - w_{ij})^2$$
  3. Adaptation of the winning neuron and the neigbors
  $$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + \alpha(t) N_c(r(t))(v_i^k - w_{ij}(t)) & for \quad j \in N_c(r(t)) \\ w_{ij}(t) & for \quad j \notin N_c(r(t)) \end{cases}$$
  4. Reduce $\alpha(t)$ and r(t); Terminate learning by max. steps/error

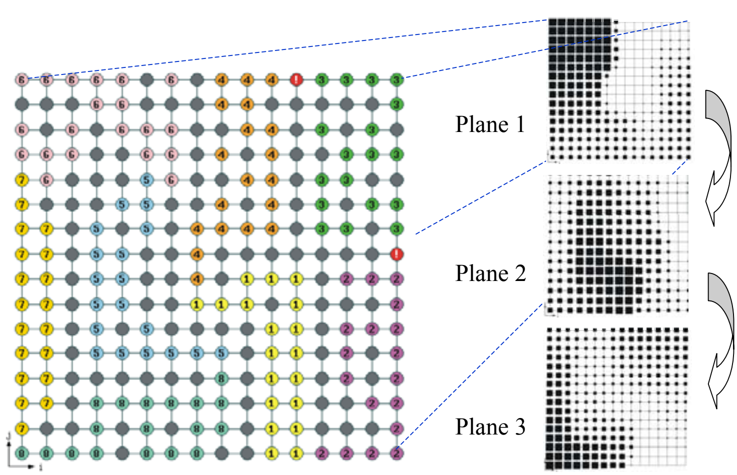
Self-organizing feature map

- During the training process, the SOM unfolds in the multivariate pattern space and creates a topology preserving mapping to the 2D neuron grid
- Example of SOM visualization for ***Cube***-data:



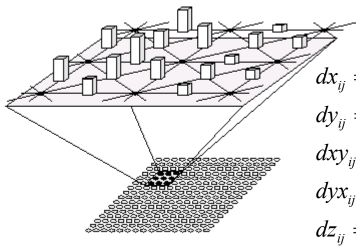
Self-organizing feature map

- SOM component planes for ***Cube***-data:



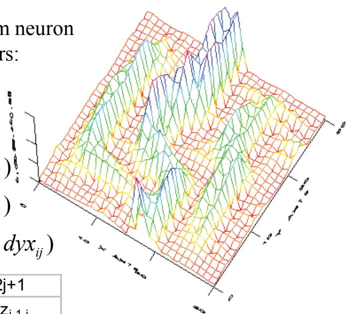
Self-organizing feature map

- SOM only indicates the presence of cluster, yet intra/inter cluster distances remain obscure
- The ***Unified-Distance-Matrix*** complements SOM with distance information



Distances from neuron uij to neighbors:

$$dx_{ij} = d(u_{ij}, u_{ij+1})$$
$$dy_{ij} = d(u_{ij}, u_{i+1j})$$
$$dxy_{ij} = d(u_{ij}, u_{i+1j+1})$$
$$dyx_{ij} = d(u_{ij+1}, u_{i+1j})$$
$$dz_{ij} = 0.5 * (dxy_{ij} + dyx_{ij})$$



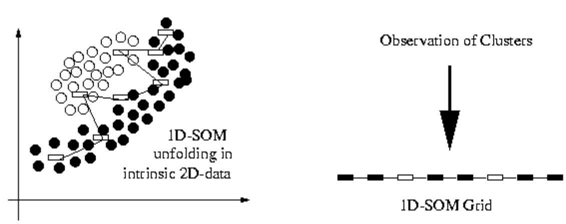
U-Matrix	U-Matrix	U-Matrix	U-Matrix	2j+1
2i-1	...	$dz_{i-1,j-1}$	$dy_{i-1,j}$	$dz_{i-1,j}$
2i	...	$dx_{i,j-1}$	$du_{i,j}$	$dx_{i,j}$
2i+1	...	$dz_{i,j-1}$	$dy_{i,j}$	$dz_{i,j}$

du(i,j) arbitrarily, set to mean of 8 neighbors

Self-organizing feature map

- SOM quantizes the data, i.e., SOM weight vectors are representatives of clusters of database vectors, which themselves are not directly visible
- SOM interpolation properties cause the placement of weight vectors in feature space regions actually void of data samples
- For high intrinsic dimension larger than SOM dimension the SOM tends to fold and twist in the attempt to establish a mapping to the 2D neural grid

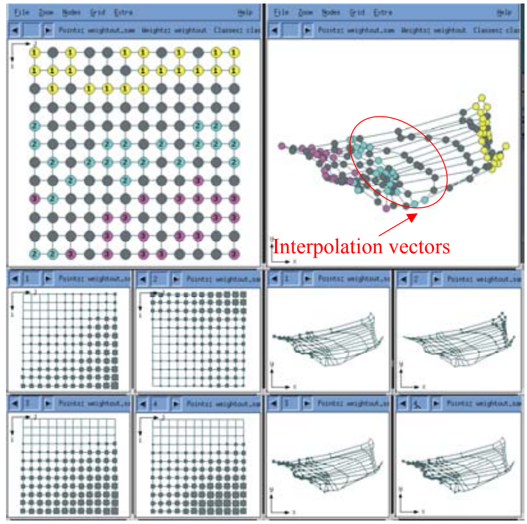
- SOM only indicates the presence of clusters. Distance information requires, e.g., U-Matrix



- Alternative: Sammon´s Nonlinear distance preserving mapping (NLM)

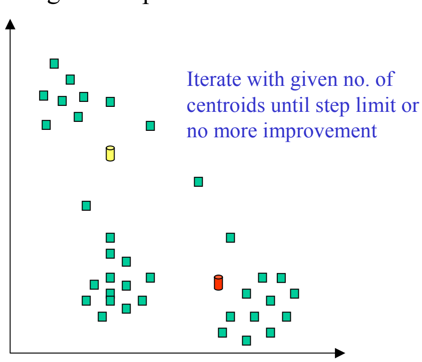
Self-organizing feature map

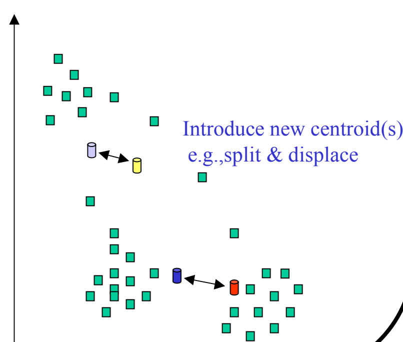
- After SOM training, the computed weight vectors are subject to NLM
- Comparison of SOM & SOM/NLM mapping for ***Iristrain*** data
- Intra/inter cluster distances become overt
- SOM unfolding & effect of interpolation vectors clearer
- The idea can be extended to a generalization of the component planes



Dynamic extensions

- In a realistic application case, the number of clusters is a priori not known
- Also, the cluster shape and the appropriate metric and number of centroids is unknown
- Dynamic or growing algorithms provide powerful alternatives
- Prominent examples are Fritzke's Growing Cells, the Neural Gas, or the Dynamic vector quanitization algorithm (DVQ)
- Rough concept:





Summary

- The chapter gave a survey of the concept and selected conventional as well as neural clustering methods
- Clustering techniques can serve for compressing and for analyzing data
- In classification, clustering helps to find an appropriate number and location of local models to employ in a classifier
- The centroids represent the original data set with (much) less storage and computational requirements
- Due to the unsupervised nature of clustering, fine tuning might be required for classification (supervised) application
- The achievable quantization error depends strongly on the optimization technique employed (to be revisited)
- Choice of metric crucial but difficult: Euclidean distance common choice
- Dynamic growing techniques powerful but not necessarily easier to use