Sensor Signal Processing

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

FB Elektrotechnik und Informationstechnik
Technische Universität Kaiserslautern

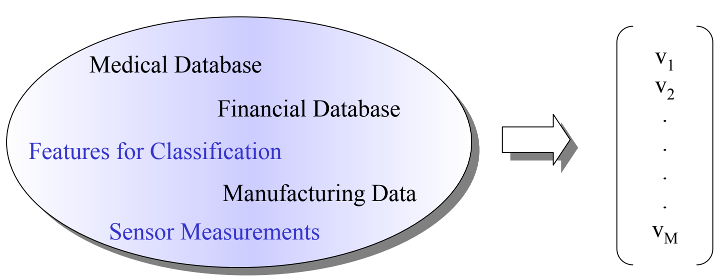
Fall Semester 2005

Course Contents

Chapter Contents

Motivation

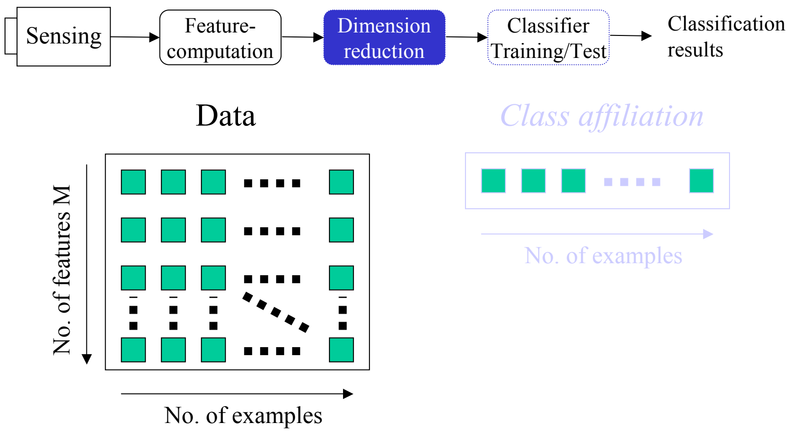
- Technical problems are often characterized by sets of high dimensional data
- Significance, correlations, redundancy, or irrelevancy of the variables v_i with regard to the given application a priori unknown



- Computational complexity and *Curse of Dimensionality* promote efficient dimensionality reduction, visualization can provide transparence & insight
- Dimensionality reduction is an ubiquitous problem in many disciplines !

Motivation

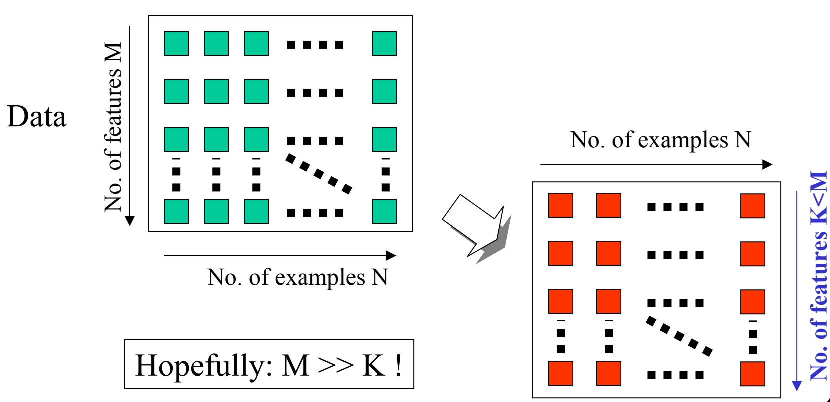
- After signal processing and feature computation further condensing or compression of the data in terms of attributes or features is desirable
- Supervised as well as unsupervised dimensionality reduction techniques can serve for that purpose !



Sensor Signal Processing
Dimensionality Reduction

Motivation

- There is a close relation between data analysis and compression based on dimensionality reduction techniques
- The purpose is to represent the data by a reduced number of variables or attributes or features according to appropriate assessment functions !



Motivation

- Projection of multivariate data by DR mappings & ensuing visualization and interactive analysis is a research topic of interest for more than 3 decades
- Recently, data mining/warehouse & knowledge discovery applications give renewed strong incentive & drive to the field
- For classification, ***feature extraction*** & ***selection*** for vectors $\vec{v} = [v_1, v_2, \ldots, v_o]^T$ are typically defined as (see, e.g., [Kittler 86])

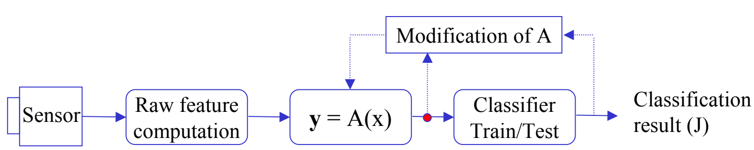
$$\boxed{\begin{array}{ll} \textit{Feature Extraction}: & J(A) = \max_A J(A(\vec{v})) \\ \textit{Feature Selection}: & J(X) = \max_\chi J(\chi) \end{array}}$$

A mapping $\Phi : I\!R^o \rightarrow I\!R^d$ is optimized with regard to assessment criterion J and with d<o and $\vec{y} = [y_1, y_2, \ldots, y_d]^T$

- Methods for classification may be salient for visualization and vice versa
- $\vec{y} = A(\vec{v})$ can be a linear or a nonlinear mapping

Motivation

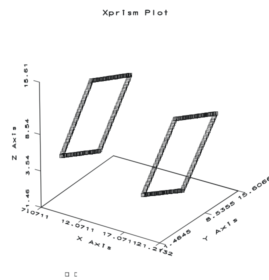
- The dimensionality reducing mapping can be unsupervised or employ supervised information



- Optimization criteria can be, e.g., signal, topology, distance preservation or discriminance gain
- Dimensionality reduction (DR) methods can be salient both for multivariate data classification and visualization
- Classification: discriminance optimization with DR constraint for lean and well performing recognition system
- Visualization: DR fixed to dimension two or three with, e.g., structure preservation constraint for data analysis, advanced MMI & system design

Motivation

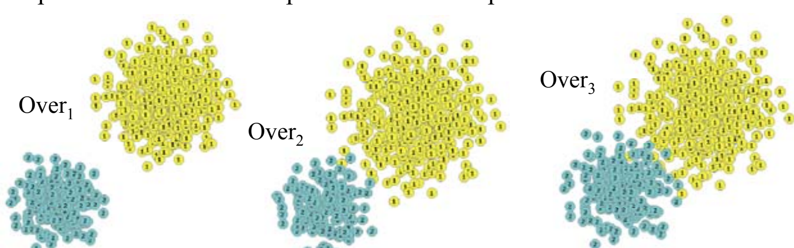
- Benchmark Data is required for method demonstration & assessment
- ***Cube*** data: Artificial data, 3 dimensions, 400 vectors, 8 classes. Data points on 8 edges of two opposite sides of a cube rotated by 45°

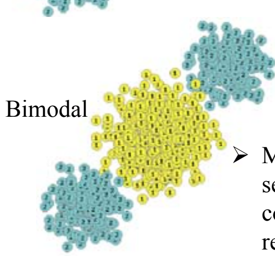


- ***Iris*** data: Well known Iris flower data, 4 dimensions, training & test set with 75 vectors each, 3 classes virginica, setosa, and versicolor
- ***Mech$_x$*** data: Mechatronic data from turbine jet engine compressor monitoring. Five data sets with 375 24-D vectors each, 4 classes corresponding to operating regions and compressor stability
- ***X-Ray*** data: X-ray inspection of ball grid array packages in electronics manufacturing. Two sets, 40 10-D vectors each, 3 classes for ok & defect type

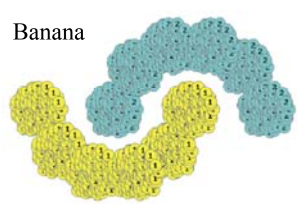
Motivation

- Supervised Assessment Options of Feature Spaces



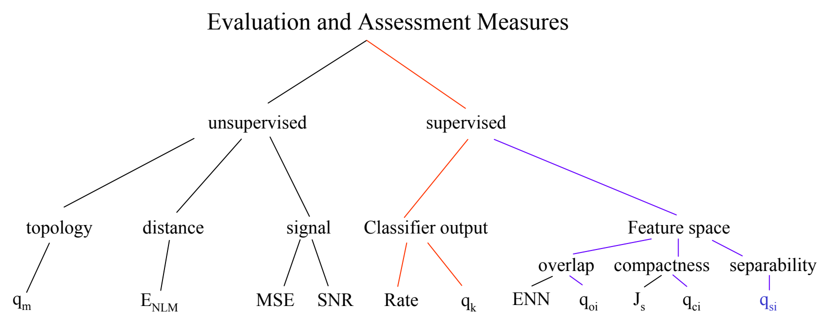


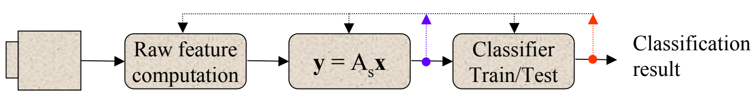
- Measure discriminance by separability, overlap, compactness, ..., of class regions in feature space



Assessment Functions

- Taxonomy of Evaluation and Assessment Measures





Assessment Functions

- Simple Parametric Overlap Measure
- Class specific distributions modelled by Gaussian functions
- Pairwise overlap can be computed from mean values μ_i, μ_j and standard deviations σ_i, σ_j by

$$q_{v_{lij}} = \frac{|\mu_i - \mu_j|}{(N_i - 1)\sigma_i + (N_j - 1)\sigma_j} \tag{5.1}$$

- The feature merit for separating one class from all others is given by

$$q_{v_{li}} = \frac{1}{L-1} \sum_{j \neq i}^{L} q_{v_{lij}} \tag{5.2}$$

- The merit of a single feature to distinguish all classes could be computed by

$$q_{v_l} = \frac{1}{L} \sum_{i=1}^{L} q_{v_{li}} \tag{5.3}$$

- Global summation can be misleading in some cases
- Simple & efficient measure for fast *parametric first order feature selection*

Assessment Functions

- Inspired by the work of Fukunaga et al. on scatter matrices & nonlinear mappings, a nonparametric compactness measure can be derived based on intra/inter class distances

$$q_{ci} = \frac{\frac{1}{L}\sum_{l=1}^{L}\frac{2}{N_l(N_l-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\partial(\omega_i,\omega_j)*\partial(\omega_i,l)*d_{X_{ij}}}{\frac{1}{N^B}\sum_{i=1}^{N}\sum_{j=i+1}^{N}(1-\partial(\omega_i,\omega_j))*d_{X_{ij}}} \quad (5.4)$$

$$with \quad d_{X_{ij}} = \sqrt{\sum_{q=1}^{M}(x_{iq}-x_{jq})^2} \quad and \quad \partial(\omega_i,\omega_j) = \begin{cases} 1 & if \quad \omega_i = \omega_j \\ 0 & if \quad \omega_i \neq \omega_j \end{cases}$$

- The Kronecker Delta $\partial(\omega_i,l)$ assures, that only vectors of class l are regarded
  - \+ Nonparametric, multivariate compactness measure
  - \+ The measure is free of user-definable parameters
  - – Measure has $O(N^2)$ complexity
  - – *Normalization properties* only allow observation of relative changes

Assessment Functions

- Inspired by probability estimation in Edited-Nearest-Neighbor method (ENN) or Leave-One-Out (LOO) classification
- For each vector, a number of k nearest-neighbors are computed & according to the neighbors´ affiliation to same/different class a nonparametric overlap measure is computed

$$q_{oi} = \frac{1}{L}\sum_{c=1}^{L}\frac{1}{N_c}\sum_{j=1}^{N_c}\frac{\sum_{i=1}^{k} q_{NN_{ji}} + \sum_{i=1}^{k} n_i}{2*\sum_{i=1}^{k} n_i} \quad (5.5)$$

$$with \quad n_i = 1 - \frac{d_{NN_{ji}}}{d_{NN_{jk}}} \quad and \quad q_{NN_{ji}} = \begin{cases} n_i & if \quad \omega_j = \omega_i \\ -n_i & if \quad \omega_j \neq \omega_i \end{cases}$$

- Simplification of the measure can be achieved by only computing the rank or just the number of nearest neighbors in the measure instead of the distances

Assessment Functions

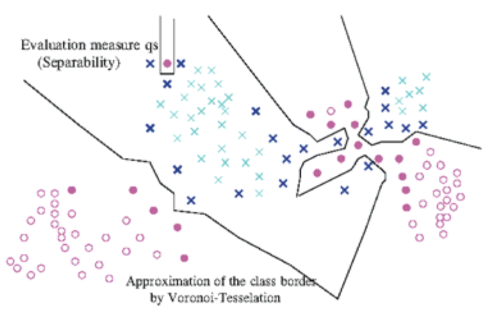
- Assessment of the nonparametric overlap measure:
  - \+ q_{oi} is well normalized in [0,1]; 1.0 indicates no overlap of class regions
  - \+ q_{oi} is fine grained and thus well suited for optimization
  - – The method has one required parameter *k* (*k* typically set to 5-10)
  - – q_{oi} has $O(N^2)$ complexity
- Application example of q_o for synthetic & application data:

Data set	q_o	q_o'	q_o''
$Over_1$	1,000000	1,000000	1,000000
$Over_2$	0,991100	0,993100	0,992800
$Over_3$	0,975300	0,979400	0,978200
Banana	1,000000	1,000000	1,000000
Bimodal	0,990900	0,989800	0,990000
Iris train	0,915600		
Pins	0,952700		
$Mech_1$	0,978800		

Sensor Signal Processing
Dimensionality Reduction

Assessment Functions

- Nonparametric Separability Measure exploits the fast training run of a Reduced-Nearest-Neighbor-Classifier (RNN)
- Separability is proportional to the number of chosen reference vectors T_{RNNi} per class during dynamic RNN configuration



$$q_{si} = \frac{1}{L}\sum_{i=1}^{L}\frac{N_i - (T_{RNN_i} - 1)}{N_i} \qquad (5.6)$$

- Here, N_i denotes the number of patterns per class & L the number of classes
  - \+ Fast due to O(N) complexity, no required user-definable parameters
  - \+ normed response [0,1], where 1.0 means linear separability
  - – Coarse grained & thus less well suited for optimization

Assessment Functions

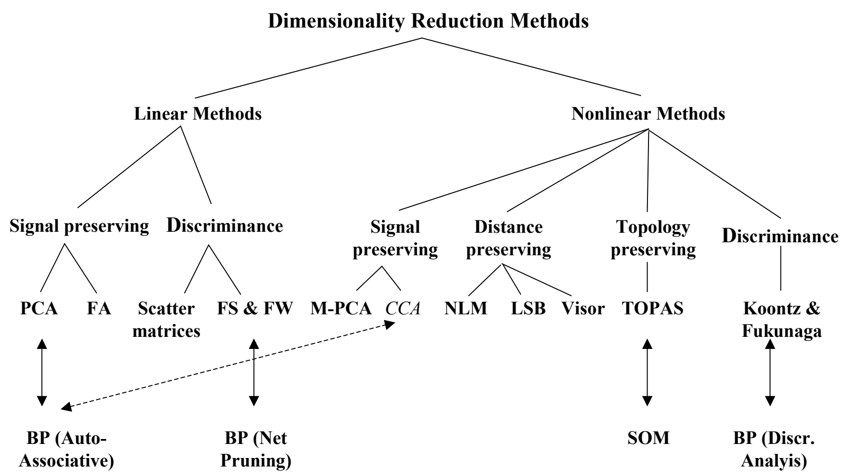
- Application example of q_{si} to synthetic and application data in comparison with Fukunaga´s measure J from scatter matrices

Data set	q_s	J_s
Over$_1$	1,000000	0,030863
Over$_2$	0,985612	0,025812
Over$_3$	0,960884	0,026084
Banana	0,993421	0,002372
Bimodal	0,983333	0,007937
Iris train	0,906667	2,659621
Pins	0,908108	3,461134
Mech$_1$	0,989333	4,824150

- Synthetic data asssessment results show the sensitivity to underlying & gradually decreasing separability
- Identical vectors in the data set with different class affiliations return an assessment value of $q_{si} = 0.0$

Feature Selection

- Taxonomy of Dimensionality Reduction Methods:



Feature Selection

- **<u>Quest:</u>** Find minimum feature subset with optimum discriminance
- AFS chooses for a given sample set the subset of variables or configuration X that maximizes a given cost function J:

$$J(X) = \max_{\chi} \; J(\chi) \tag{5.7}$$

- AFS is a linear mapping, based on the selection matrix A_S with $\vec{y}_j = A_s \vec{x}_j$
- ***Feature selection***: $c_i \in \{0,1\}$ switch variables, 2^M combinations
- ***Feature weighting***: $c_i \in [0,1]$ or arbitrary real numbers

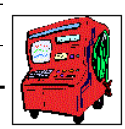
$$A_S = \begin{pmatrix} c_1 & 0 & 0 & \ldots & 0 \\ 0 & c_2 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & c_{M-1} & 0 \\ 0 & 0 & 0 & \ldots & c_M \end{pmatrix} \tag{5.8}$$

- Cost function J can be one of the measures given in the taxonomy, e.g., q_{si}
- Combinatorial optimization problem, **NP-complete**
- Exhaustive search for global optimum only feasible for small *M*
- **Way out:** Apply heuristics & optimization strategies to find at least a local optimum with bounded time and effort

Feature Selection

- **Simplification**: J is computed for each individual feature and a selected combination of classes, e.g., pairwise class discriminance
- Considerable computational savings by neglecting possible higher order correlations between feature pairs or tupels
- $q_{l_{ij}}$ could serve as simple **parametric** measure (1rst order par. selection)
- For each class separation feature are ranked according to their J value
- Rank table example for ***Iristrain*** data:

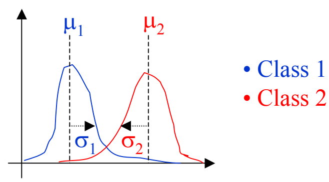
Feature	Rank	C 1-2	Rank	C 1-3	Rank	C 2-3
v_1	4	1.020	3	1.482	3	0.442
v_2	3	1.065	4	0.890	4	0.255
v_3	2	4.139	**1**	**5.451**	2	1.218
v_4	**1**	**4.387**	2	5.180	**1**	**1.660**

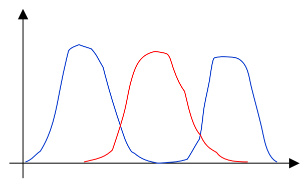


- Selection takes place by choosing the features on top rank positions, e.g., features 3 & 4 for first rank position ($C=[0, 0, 1, 1]^T$)

Assessment of Parametric First-Order Feature Selection (FOFS):

- The method is very fast and its complexity is O(M)
- However, for pairwise class separation, the complexity grows with O(L)
- The rank tables grow exponentially, visual inspection becomes infeasible !
- Only a local optimum solution can be expected
- The method can be extended to different class separation (one vs. all, all)
- Inclusion of lower ranking features can affect the solution
- **Summary**: If parametric assumption is met, method can be fast & effective
- **Problem**: Nonparametric and multimodal one-dimensional distributions:





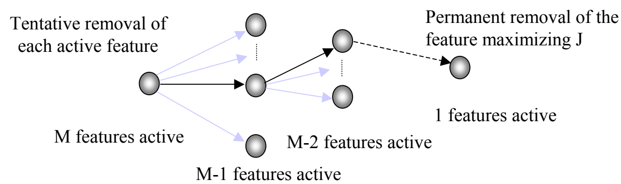
- **Remedy**: A nonparametric J is computed, e.g., q_{oi} restricted to one dimension
- Example for application data with two classes from visual inspection:

Feature	Rank Par.)	C 1-2	Rank (Nonpar.)	C 1-2
v_1	5	0.2917	5	0.6977
v_2	4	0.6489	3	0.8211
v_3	**1**	**1.1558**	4	0.7058
v_4	3	0.8547	2	0.8808
v_5	2	1.0047	**1**	**0.9270**

- The global solution $C=[0, 0, 0, 1, 1]^T$ was determined by exhaustive search
- The nonparametric FOFS found the global optimum, parametric FOFS failed
- However, the number of rank positions salient for the given problem is not immediately obvious
- FOFS can serve as first step to confine search space in selection hierarchy

Feature Selection

- Nonparametric cost function is computed for each configuration, i.e., the currently selected subset of features
- Heuristic search strategy is applied to confine the search space
- Sequential-Backward/Forward-Selection (SBS/SFS):



- M*(M-1)/2+M combinations have to be assessed for SBS/SFS
  $\Longrightarrow$ $O(M^2)$ complexity
- For M=16, a local optimum solution will be found in approx. 136 s (1 s per assessment assumed) in contrast to 18h for exhaustive search

Feature Selection

- Application example of SBS for ***Iris*** train:

```
1 2 3 4    0.90667
- 2 3 4    0.94667
- - 3 4    0.96000
- - - 4    0.00000
Optimum quality:    0.96
Significant Features:3 4
```

- Application example of SBS for visual inspection data:

```
1 2 3 4 5    0.903846
- 2 3 4 5    0.942308
- - 3 4 5    0.961538
- - - 4 5    0.961538
- - - - 5    0.913462
Optimum quality: 0.961538
Significant Features: 4 5
```

- Further applicable heuristics: Branch & bound, floating search, etc.

Feature Selection

- Stochastic methods, e.g., Simulated Annealing (SA), applicable as heuristic
- Perturbation Method (PM) is a simple variant, based on random state changes
- PM application for ***Iristrain***:

```
                 Proposed Accepted
1  -  -  -  :    0.00000  0.000000
1  -  3  -  :    0.88000  0.880000
1  -  3  -  :    0.00000  0.880000
1  -  3  4  :    0.94667  0.946667
1  -  3  4  :    0.92000  0.946667
1  -  3  4  :    0.90667  0.946667
1  -  3  4  :    0.90667  0.946667
1  -  3  4  :    0.88000  0.946667
1  -  3  4  :    0.88000  0.946667
1  -  3  4  :    0.88000  0.946667
-  -  3  4  :    0.96000  0.960000
-  -  3  4  :    0.00000  0.960000
-  -  3  4  :    0.00000  0.960000
-  -  3  4  :    0.00000  0.960000
-  -  3  4  :    0.94667  0.960000
Best Quality: 0.96000
Best Features: 3  4
```

- Random initial state
- Random selection of variable to propose state transition
- Selected variable is toggled and transition is accepted if

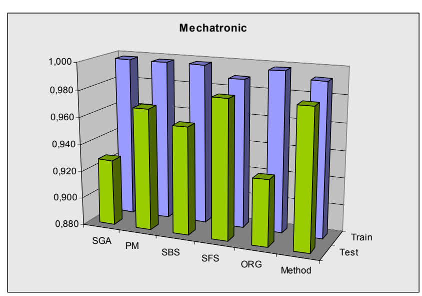
$$\Delta q_{si} = q_{si}^{new} - q_{si}^{old} \qquad (5.9)$$

- Random tossing surprisingly effective at moderate effort
- Different results for each run !
- Several enhancements possible (multiple variable change or SA)

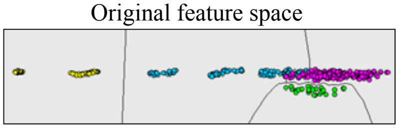
Feature Selection

- Additional optimization techniques can be applied to feature selection, e.g., from evolutionary computation like genetic algorithms & swarm intelligence
- **Results for $Mech_1$ & $Mech_2$:**
  - SFS and FSSPEA find same optimum solution !
  - Inclusion of further constraints

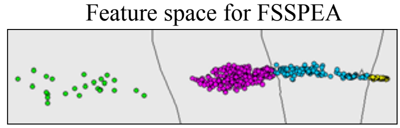
Method	$Mech_1$	$Mech_2$	Features
ORG	0,992500	0,981667	24
→ SFS	0,997500	0,928333	3
SBS	0,989333	0,981667	12
PM	0,997500	0,959167	6
SGA	0,997333	0,969167	12
→ FSSPEA	0,997500	0,928333	3



Original feature space

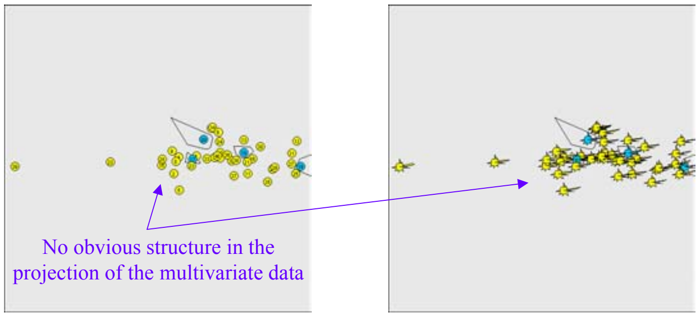


Feature space for FSSPEA



Feature Selection

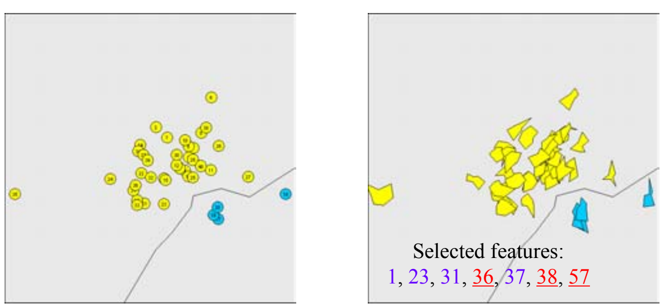
- Small application example from microelectronic manufacturing: Wafer Data Analysis
- Simple example of DR methodology potential for general fab data analysis
- The database was gathered from an MPC-Run with 10 Wafers and 4 Parameter Extraction Sites per Wafer, 59 parameters each:



- All chips with unsatisfactory behavior on Wafer 5 (blue; 1-4, 6-10 yellow)
- **Assume Wafer 5 processing was abnormal, what makes it different ?**

Feature Selection

- The lack of obvious structure can be due to absence of abnormity or to occlusion by a majority of normal parameters and high intrinsic dimension
- Thus, AFS is employed to find parameters supporting abnormity hypothesis



- From 59 parameters AFS chose 3 for q_{si}/SBS and 7 for q_{oi}/SBS
- **There is (weak) hypothesis support:** ***Can these features be responsible ?***

Feature Selection

Sensor Signal Processing
Dimensionality Reduction

➢ Analyzing the AFS choice by probing the parameter meaning:

Parameter	RCNTMPN	RDIFFN	RDIFFP	RMET	RMET2	RNWELL	RON3NH	RPOLY2	RPOLYN	RVIA
	Ohm/cont	Ohm/[]	Ohm/[]	mOhm/[]	mOhm/[]	kOhm/[]	kOhm.um	Ohm/[]	Ohm/[]	mOhm/via
min		20	25			1	15	21	20	
Spec max	20	35	60	140	80	1,4	23	33	28	300
Mitelwert	4,9079	23,5965	43,0638	70,4023	37,2045	1,1110	19,6393	27,6148	25,5955	#####
Varianz	0,0717	0,0240	0,6963	1,5038	0,7536	0,0000	0,0036	0,1757	1,0725	23,1676
Wafer 1 Site 1	0,3271	-0,1765	-0,3637	-1,7822	0,1855	0,0010	-0,0493	-0,5148	-0,1755	-3,8600
Wafer 1 Site 2	0,0081	-0,2565	-0,8738	-0,3422	-1,2045	0,0030	0,0007	-0,4548	0,8745	6,5400
Wafer 1 Site 3	0,1151	-0,1265	-0,6037	0,2278	0,2855	-0,0020	-0,0493	-0,1048	0,2245	0,1400
Wafer 1 Site 4	-0,1089	-0,1665	-0,3438	2,3378	0,3355	-0,0030	-0,0293	0,0752	0,9645	3,5400
Wafer 2 Site 1	0,0801	-0,0765	-0,7137	-2,0622	1,1855	0,0050	-0,0293	-0,5948	0,2245	2,1400
Wafer 2 Site 2	-0,0499	-0,2165	-1,1238	0,4578	0,6355	0,0060	0,0007	-0,8148	0,7645	11,1400
Wafer 2 Site 3	-0,0719	-0,1265	-0,8537	0,2778	0,8255	-0,0030	-0,0893	-0,3048	0,3845	5,9400
Wafer 2 Site 4	-0,0209	-0,1365	-0,5437	1,9178	1,9155	0,0020	-0,0793	-0,0448	0,5745	9,1400
Wafer 3 Site 1	0,2131	-0,0265	-0,2837	-1,6122	-0,4245	-0,0030	-0,0593	-0,4348	0,4145	-9,2600
Wafer 3 Site 2	0,0221	-0,1565	-0,7638	-0,5823	-1,0545	0,0080	-0,0193	-0,5448	1,0845	-0,4600
Wafer 3 Site 3	0,1241	-0,0965	-0,7638	0,4078	-0,0845	-0,0010	-0,0893	-0,0948	0,3645	-4,6600
Wafer 3 Site 4	0,0461	-0,0865	-0,3038	0,7678	-0,1045	-0,0030	-0,0493	0,0852	0,6845	-1,8600
Wafer 4 Site 1	0,0821	0,2435	0,8963	-0,6822	-0,4845	-0,0040	0,0007	-0,3148	0,4545	-6,2600
Wafer 4 Site 2	-0,1059	0,0735	0,7763	0,2278	-0,8945	0,0040	0,0307	-0,3348	1,0245	2,5400
Wafer 4 Site 3	-0,0949	0,0435	0,2263	-0,4822	0,5755	-0,0050	0,1207	0,1052	0,5745	-2,2600
Wafer 4 Site 4	-0,3129	0,1135	0,8363	1,6078	-0,4245	-0,0040	0,0607	0,0852	1,0245	-0,4600
Wafer 5 Site 1	0,3841	0,0035	-0,5938	-1,2723	-0,1045	-0,0010	0,1107	0,6552	-0,4155	-8,6600
Wafer 5 Site 2	0,0521	-0,1565	-0,8738	-0,0422	-1,8645	0,0020	0,1507	0,3952	0,6645	-2,6600
Wafer 5 Site 3	0,1741	-0,0865	-0,8637	0,4678	0,3555	-0,0080	0,0107	0,8052	0,1445	-6,0600
Wafer 5 Site 4	-0,0479	-0,1565	-0,2937	1,7678	0,0855	-0,0020	0,0007	0,7052	0,7645	-3,8600
Wafer 6 Site 1	-0,5139	0,1335	0,4462	-1,8823	0,3155	0,0030	0,0007	0,4052	-2,0855	-5,8600
Wafer 6 Site 2	-0,5469	0,0135	0,4162	-0,2723	-0,8445	0,0040	0,0607	0,0852	-2,0555	2,3400
Wafer 6 Site 3	-0,3849	0,0335	-0,0737	-0,1522	0,4255	-0,0010	-0,0793	0,6552	-2,0855	-1,6600
Wafer 6 Site 4	-0,4699	0,0935	0,7763	1,5278	0,3855	0,0050	0,0507	0,3252	-1,8155	0,7400
Wafer 7 Site 1	-0,2969	-0,0765	-0,8537	-1,2923	-0,3445	0,0010	0,0007	-0,1248	-2,4955	-3,6600
Wafer 7 Site 2	-0,3859	-0,1765	-1,3138	-0,5823	-1,5845	0,0020	0,0907	-0,1848	-1,0355	5,7400
Wafer 7 Site 3	-0,3589	-0,0865	-1,1038	0,0678	0,8555	-0,0130	-0,0493	0,2252	-1,9555	1,9400
Wafer 7 Site 4	-0,4209	-0,1765	-0,7037	1,3378	0,3755	-0,0090	-0,0793	0,1352	-1,0055	4,5400
Wafer 8 Site 1	0,4341	0,1435	0,9263	-1,2723	0,1855	0,0050	0,0007	0,2452	-0,5755	-5,6600
Wafer 8 Site 2	0,0911	0,0335	0,6363	0,1578	-1,1945	0,0110	0,0507	-0,0648	0,5045	5,9400
Wafer 8 Site 3	0,1611	0,0635	0,2763	-0,1322	0,3055	-0,0020	-0,0493	0,3652	-0,1655	-1,0600
Wafer 8 Site 4	0,0641	0,0935	0,8762	1,8378	0,0155	0,0040	0,0007	0,2652	0,5945	2,3400
Wafer 9 Site 1	0,4741	0,3935	1,6663	-0,9622	-0,1045	-0,0020	0,0607	0,2952	-0,2955	-3,6600

➢ The subcluster was found for the assumption of Wafer 5 abnormity

➢ The parameters must be checked for physical plausibility with regard to observed chip behavior

➢ The visualization & analysis result is imposed on the original database in Excel

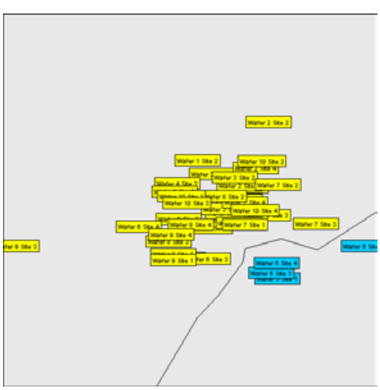
➢ In the given case, the detected abnormity in the selected parameters cannot be held responsible

➢ Probably a design error, not a manufacturing problem !

Feature Selection

Sensor Signal Processing
Dimensionality Reduction

➢ This simple example only scratched the surface of the application potential
➢ Feasibility demonstration of DR methodology for microelectronic data



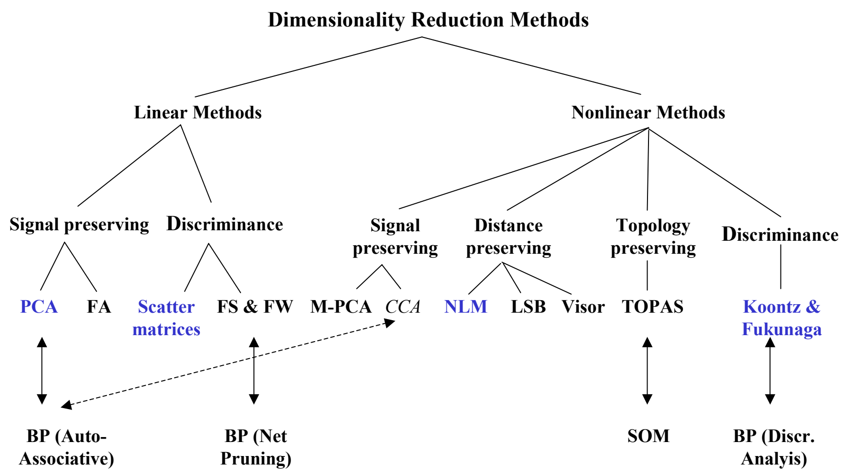
➢ In this case found parameters distinguish Wafer 5 but are not relevant to observed problem

➢ The outlined approach offers fast, efficient, and transparent access to multivariate, complex data met in today´s manufacturing processes

➢ Other nonobvious information can be found & employed for process optimization & centering

➢ Circuit design & design centering is another potential application field

- Taxonomy of Dimensionality Reduction Methods:



- Dimensionality reduction by Principal Component Analysis
- Dimensionality reduction method from communication science
- Objective: Find a linear transformation with $\vec{y}_i = W_{1..m}\vec{v}_i$ and $\hat{\vec{v}}_i = W^T_{1..m}\vec{y}_i$ so that the reconstruction error is minimum for given m

$$E = \frac{1}{N}\sum_{i=1}^{N}(\vec{v}_i - \hat{\vec{v}}_i)^2 \qquad (5.10)$$

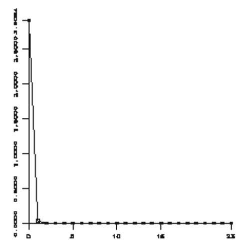
- Visualization employs the first two (three) principal components
- Compute covariance matrix of the database:

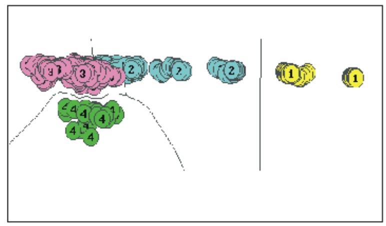
$$K = \frac{1}{N}\sum_{i=1}^{N}(\vec{v}_i - \vec{\mu}_j)(\vec{v}_i - \vec{\mu}_j)^T \quad \text{with} \quad \vec{\mu} = \frac{1}{N}\sum_{i=1}^{N}\vec{v}_i \qquad (5.11)$$

- Compute Eigenvalues λ_i and Eigenvectors ψ_i of matrix K
- Data is decorrelated by applying $\vec{y}_i = \Psi\vec{v}_i$ (5.12)
- Largest Eigenvalue corresponds to component with largest variance in the data
- Eigenvalues are sorted and the *m* Eigenvectors corresponding to the largest Eigenvalues are selected for projection: $\vec{y}_{i_{1..m}} = \Psi_{1..m}\vec{v}_i$ (5.13)

Feature Extraction

- For compression in channel coding or classification, *m* can be chosen according to Scree-plot or achieved classification rate **R**
- PCA is a linear, signal-preserving approach based on the assumption of unimodal Gaussian data. Example for Mech_1 data:

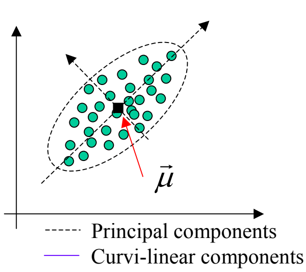


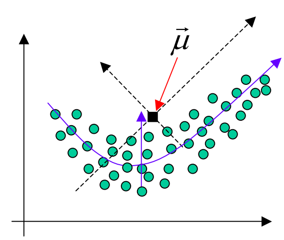


- Most variance is embodied in the first two principal components
- **R** grows with M^2 for M-D data (Numerical & accuracy problems)
- Bad visualization for high ***intrinsic dimensionality*** or nonlinear components

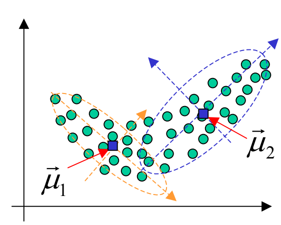
Feature Extraction

- For nonlinear data, the assumptions of PCA do not hold:

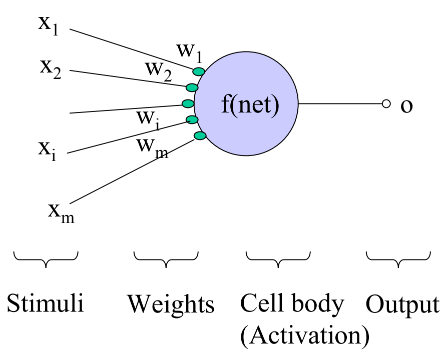




- **Remedy:** Curvi-linear components (CCA)
- Piecewise linear approximation in an hierarchical approach using clustering & local PCA computation (M-PCA)

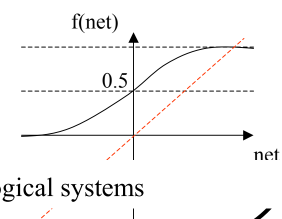


- Artificial neural networks are commonly applied for function approximation in classification and dimensionality reduction
- Simple artificial neuron model:



$$o = f\left(\sum_{i=1}^{m} x_i w_i\right) \tag{5.14}$$

$$f(net) = \frac{1}{1+e^{-net}} \tag{5.15}$$



- Coarse abstraction of the natural nerve cell in biological systems

- Adaptation of a neuron weight is commonly achieved by gradient descent based on an error function:

$$E = \frac{1}{2}\sum_{k=1}^{N}\left(y^k - f\left(\sum_{i=1}^{m} x_i^k w_i\right)\right)^2 \tag{5.16}$$

- Every weight is adapted after (random) initialization according to:

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \tag{5.17}$$

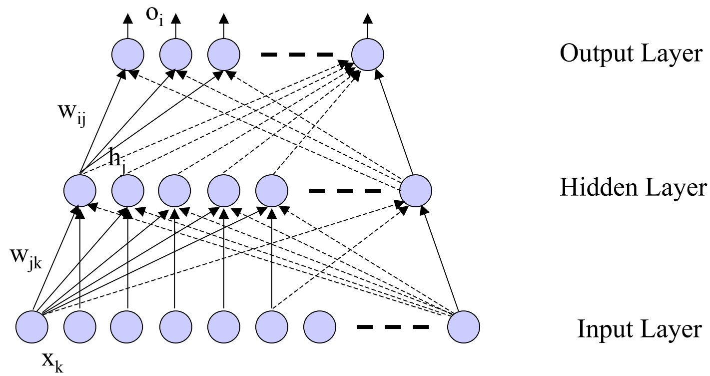
- The gradient is computed as:

$$\frac{\partial E}{\partial w_i} = -\sum_{k=1}^{N}\left(y^k - f\left(\sum_{i=1}^{m} x_i^k w_i\right)\right)\cdot f'\left(\sum_{i=1}^{m} x_i^k w_i\right)\cdot x_i^k \tag{5.18}$$

- Inserting (5.18) in (5.17) return the batch learning rule, reducing the batch to one returns on-line learning rule with immediate weight adaptation

Feature Extraction

- Commonly, a multi-layered network of multiple of such neuron models is applied, denoted as multi-layer feedforward neural network :



- The network can be extended by more hidden layers, but already the given topology is proven to be a universal function approximator
- **A** learning rule is required for this network, in particular the hidden layer

Feature Extraction

- Introducing the following abbreviations using the notation given with the network structure:

$$o_i = f\left(\sum_j h_j w_{ij}\right) \quad ; h_j = f\left(\sum_k x_k w_{jk}\right) \tag{5.19}$$

- With (5.19) the error can be expressed as:

$$E = \frac{1}{2}\sum_{\mu=1}^{N}\sum_{i=1}^{L}\left(y_i^\mu - f\left(\sum_{i=1}^{m} h_j^\mu w_{ij}\right)\right)^2 \tag{5.20}$$

- Every weight is adapted after (random) initialization according to:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \tag{5.21}$$

- The gradient for the output layer weights is computed as:

$$\frac{\partial E}{\partial w_{ij}} = -\sum_{\mu=1}^{N}\left(y_i^\mu - f\left(\sum_j h_j^\mu w_{ij}\right)\right)\cdot f'\left(\sum_j h_j^\mu w_{ij}\right)\cdot h_j^\mu \tag{5.22}$$

- This can be expressed employing the abreviations of (5.19) as

$$\frac{\partial E}{\partial w_{ij}} = -\sum_{\mu=1}^{N}\left(y_i^\mu - o_i^\mu\right)\cdot o'^{\mu}_i \cdot h_j^\mu \qquad (5.23)$$

- Inserting in (5.21) gives the output layer batch adaptation rule

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta \sum_{\mu=1}^{N}\left(y_i^\mu - o_i^\mu\right)\cdot o'^{\mu}_i \cdot h_j^\mu \qquad (5.24)$$

- For the hidden layer adaptation rule, the error function must be expanded:

$$E = \frac{1}{2}\sum_{\mu=1}^{N}\sum_{i=1}^{L}\underbrace{\left(y_i^\mu - \underbrace{f\left(\underbrace{\sum_j w_{ij}\, \underbrace{f\left(\underbrace{\sum_k x_k^\mu w_{jk}}_{net_j} \right)}_{h_j}}_{net_i} \right)}_{o_i} \right)}_{e_i}^2 \qquad (5.25)$$

- Every hidden weight is adapted after (random) initialization according to:

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} \qquad (5.26)$$

- The gradient for the hidden layer weights is computed by application of the chain rule:

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial e_i}\cdot\frac{\partial e_i}{\partial o_i}\cdot\frac{\partial o_i}{\partial net_i}\cdot\frac{\partial net_i}{\partial h_j}\cdot\frac{\partial h_j}{\partial net_j}\cdot\frac{\partial net_j}{\partial w_{jk}} \qquad (5.27)$$

$$\frac{\partial E}{\partial w_{jk}} = -\sum_{\mu=1}^{N}\sum_{i=1}^{L}\left(y_i^\mu - f\left(\sum_j h_j^\mu w_{ij}\right)\right)\cdot f'\left(\sum_j h_j^\mu w_{ij}\right)\cdot w_{ij}\cdot f'\left(\sum_j x_k^\mu w_{jk}\right)\cdot x_k^\mu \qquad (5.28)$$

- This can again be expressed employing the abreviations of (5.19) as

$$\frac{\partial E}{\partial w_{jk}} = -\sum_{\mu=1}^{N}\sum_{i=1}^{L}\left(y_i^\mu - o_i^\mu\right)\cdot o'^{\mu}_i \cdot w_{ij}\cdot h'^{\mu}_j \cdot x_k^\mu \qquad (5.29)$$

Feature Extraction

- Introduction of error or δ-terms with

$$\delta_i^\mu = \left(y_i^\mu - o_i^\mu\right) \cdot o'^\mu_i \tag{5.30}$$

$$\delta_j^\mu = h'^\mu_j \cdot \sum_{i=1}^{L} w_{ij} \delta_i^\mu \tag{5.31}$$

- ... allows a compact representation of the adaptation rules

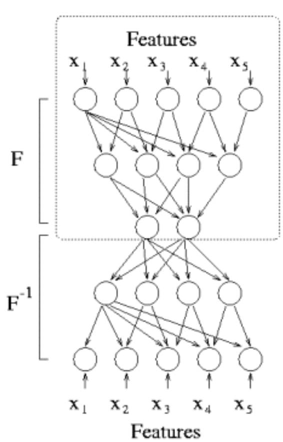
$$\Delta w_{ij} = \eta \sum_{\mu=1}^{N} \delta_i^\mu \cdot h_j^\mu \tag{5.32}$$

$$\Delta w_{jk} = \eta \sum_{\mu=1}^{N} \delta_j^\mu \cdot x_k^\mu \tag{5.33}$$

- This learning rule is denoted as error-backpropagation learning rule
- Numerous variants of this vanilla approach are in existence to improve learning behavior, e.g., introduction of a momentum term or adaptive η

Feature Extraction

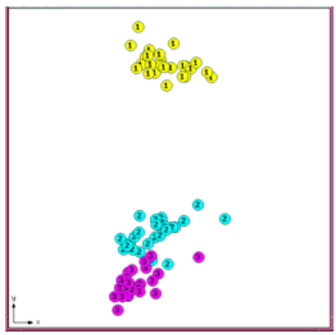
- Nonlinear Signal Preserving Mappings
- Application of backpropagation networks in autoassociative mode:

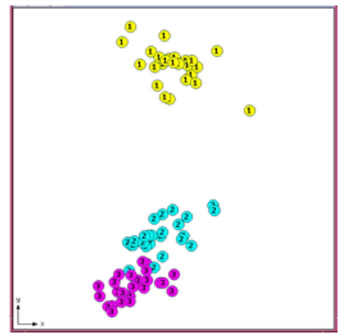


- Network learning tries to preserve the feature values over the bottleneck layer
- 3L networks perform similar to PCA
- Number of bottleneck layer neurons define projection dimension *d*
- Example with 4-2-4 BP network
- 5L networks performs nonlinear compression, extracting principal curves
- Topology must be specified by user
- Hard to train and to interpret
- Example with 4-9-2-9-4 BP network

Feature Extraction

- Visualization of ***Iristrain*** data by 3L-BP in autoassociative mode:

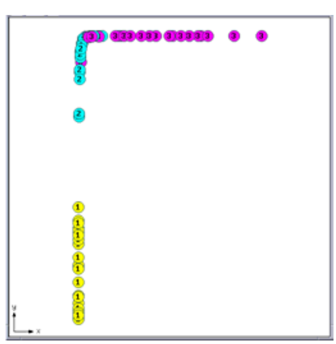


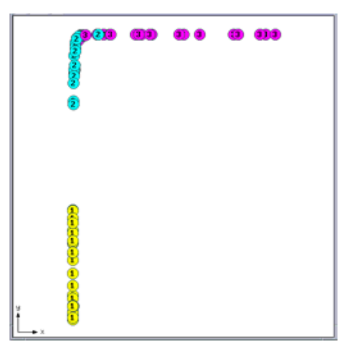


- Visualization of ***Iristest*** data by 3L-BP in autoassociative mode:

Feature Extraction

- Visualization of ***Iristrain*** data by 5L-BP in autoassociative mode:





- Visualization of ***Iristest*** data by 3L-BP in autoassociative mode:

- Distance Preserving Nonlinear Mapping: Sammon´s NLM
- Particular case of Multi-dimensional-scaling (MDS) approach
- Sammon´s stress assesses distortion of distances while mapping the data vectors from **X** space to corresponding pivot points in **Y** space by his NLM:

$$E(m) = \frac{1}{c}\sum_{j=1}^{N}\sum_{i=1}^{j}\frac{(d_{X_{ij}} - d_{Y_{ij}}(m))^2}{d_{X_{ij}}} \quad with \quad c = \sum_{j=1}^{N}\sum_{i=1}^{j} d_{X_{ij}} \tag{5.34}$$

- Here, d_{Xij} and d_{Yij} denote the interpoint distance in feature space **X** and projection space **Y**, respectively:

$$d_{Y_{ij}}(m) = \sqrt{\sum_{q=1}^{d}(y_{iq}(m) - y_{jq}(m))^2} \tag{5.35}$$

$$d_{X_{ij}} = \sqrt{\sum_{q=1}^{M}(x_{iq} - x_{jq})^2} \tag{5.36}$$

- Minimization of Sammon´s stress using gradient descent is achieved by iterative computation of new coordinates for the pivot vectors in **Y** space:

$$y_{iq}(m+1) = y_{iq}(m) - MF * \Delta y_{iq}(m) \tag{5.37}$$

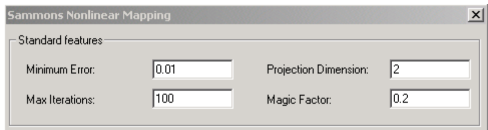
$$with \qquad \Delta y_{iq}(m) = \frac{\partial E(m)}{\partial y_{iq}(m)} \Big/ \left|\frac{\partial^2 E(m)}{\partial y_{iq}(m)^2}\right| \tag{5.38}$$

- The partial derivatives for gradient descent are given by

$$\frac{\partial E(m)}{\partial y_{iq}} = \frac{-2}{c}\sum_{\substack{j=1 \\ j\neq i}}^{N}\left[\frac{1}{d_{Y_{ij}}} - \frac{1}{d_{X_{ij}}}\right](y_{iq} - y_{jq}) \tag{5.39}$$

$$\frac{\partial^2 E(m)}{\partial y_{iq}^{\ 2}} = \frac{-2}{c}\sum_{\substack{j=1 \\ j\neq i}}^{N}\left[\frac{1}{d_{Y_{ij}}} - \frac{1}{d_{X_{ij}}} - \frac{(y_{iq} - y_{jq})^2}{d_{Y_{ij}}^{\ 3}}\right] \tag{5.40}$$

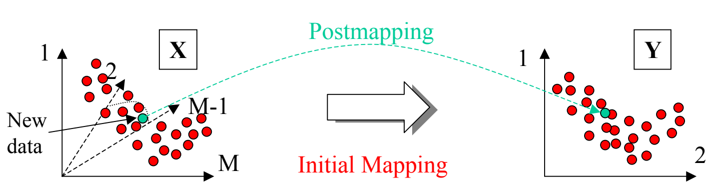
- NLM maps each point and preserves distances & structure in data
- Salient visualization method for direct data projection & visual analysis
- Mapping control:



- Error E(m)>0.1 indicates unacceptable mapping result [Sammon 69]
- Computation of NLM has $O(N^2)$ complexity
- Method becomes infeasible for large databases
- Gradient descent optimization is not capable to exactly preserve all distances
- Inevitable mapping error is introduced

Acceleration for approximated mapping by heuristics !

- After initial dimensionality reducing mapping computation for a given data set, single or groups of data points shall be subject to the same mapping, too
- In classification, test & application data shall be mapped in recall
- In visualization, new data points shall be mapped & displayed



- In both cases, the mapping of additional points shall take place without recomputation of the entire mapping
- PCA & ANN´s satisfy this requirement, NLM originally does not
- **Remedy 1:** Train an ANN with NLM mapping data & use it for recall
- **Remedy 2:** Introduce NLM recall mapping (NLMR)

- Introduction of a NLM recall method (NLMR) basing on the previously computed NLM
- New data vectors $\vec{v}_i^r$ subject to recall are placed according to their distances
- $d_{\hat{X}_{ij}}$ to the training data points $\vec{v}_j^t$
- Mutual distances between test vectors are neglected in NLMR
- Thus the cost function is modified to:

$$\hat{E}_i(m) = \frac{1}{\hat{c}} \sum_{j=1}^{K} \left(\frac{(d_{\hat{X}_{ij}} - d_{\hat{Y}_{ij}}(m))^2}{d_{\hat{X}_{ij}}} \right) \qquad (5.41)$$

$$with \quad d_{\hat{X}_{ij}} = \sqrt{\sum_{q=1}^{M} (\vec{v}_{iq}^r - \vec{v}_{jq}^t)^2} \quad and \quad \hat{c} = \sum_{j=1}^{K} d_{\hat{X}_{ij}}$$

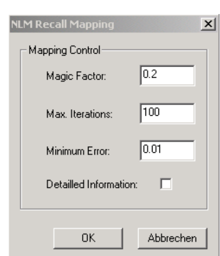
- K corresponds to the number of initially mapped training samples

- Each point is randomly initialized in **Y** space
- Iterative adjustments by gradient descent take place according to:

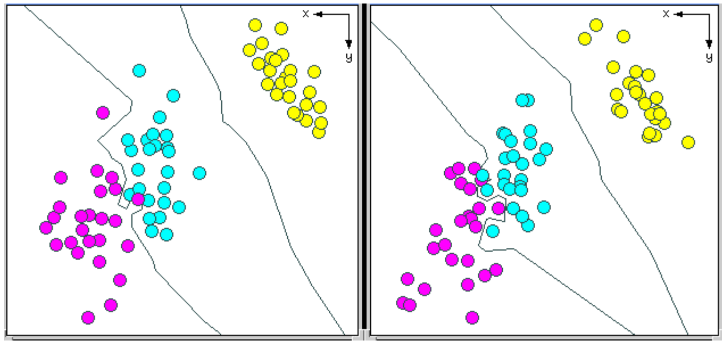
$$\hat{y}_{iq}(m+1) = \hat{y}_{iq}(m) - MF * \Delta\hat{y}_{iq}(m) \qquad (5.42)$$

$$with \quad \Delta\hat{y}_{iq}(m) = \frac{\partial \hat{E}_i(m)}{\partial \hat{y}_{iq}(m)} \Big/ \left| \frac{\partial^2 \hat{E}_i(m)}{\partial \hat{y}_{iq}(m)^2} \right| \quad and \quad 0 < MF \le 1$$

- Mapping of test data for recall or postmapping is now feasible
- The number of iterations required per data vector mapping vary considerably
- Computational savings can be achieved for given Minimum Error threshold



- NLMR Application example for ***Iristrain*** and ***Iristest*** data:



- NLM(R) can now serve as nonlinear alternative to PCA. Quantitative discriminance comparison showed NLM(R) to be superior for given dimension

- Quantitative Assessment of selected unsupervised Mapping Techniques
- Compression from M- to 2D data for linear/nonlinear unsupervised methods
- Evaluation by overlap q_o and separability q_s measures for training & test sets

Method	Dim.	Train	q_o	q_s	Test	q_o	q_s
Original	4	Iristrain	0.95503	0.90666	Iristest	0.91683	0.88000
PCA	2	Iristrain	0.91329	0.86667	Iristest	0.91970	0.89333
NLM	2	Iristrain	0.94250	0.89333	Iristest	0.93128	0.89333
BP	2	Iristrain	0.93008	0.90666	Iristest	0.91630	0.90667
Original	24	$Mech_1$	1.000	0.98933	$Mech_2$	0.99799	0.96308
PCA	2	$Mech_1$	0.98959	0.97067	$Mech_2$	0.94889	0.91384
NLM	2	$Mech_1$	0.97898	0.94400	$Mech_2$	0.95730	0.91384
BP	2	$Mech_1$	0.97861	0.94400	$Mech_2$	0.93619	0.89538

- BP has 4-9-2-9-4 network topology, 1000 epochs of quickprop learning
- PCA is simple and excellent when applicable to the given data
- NLM(R) outperforms PCA for nonlinear data & is easy to use
- BP results are very hard to obtain & approximately as those of NLM(R)
- This comparison assessed discriminance not structure preservation !

- Supervised linear technique for dimensionality reduction [Fukunaga 90]
- **Objective:** Find a linear transformation that minimizes the within or intraclass distances between data points and maximizes the between or interclass distances, i.e., class regions shall become compact & well separated

- Scatter matrices are computed for this aim. In the parametric approach interclass S_b and intraclass S_w scatter matrices are computed by:

$$S_w = \frac{1}{N}\sum_{i=1}^{L}\sum_{j=1}^{N_i}(\vec{v}_j^{\omega_i} - \vec{\mu}_j^{\omega_i})(\vec{v}_j^{\omega_i} - \vec{\mu}_j^{\omega_i})^T \quad \text{with} \quad \vec{\mu}^{\omega_i} = \frac{1}{N_i}\sum_{j=1}^{N_i}\vec{v}_j^{\omega_i} \qquad (5.43)$$

$$S_b = \frac{1}{N}\sum_{i=1}^{L}\frac{N_i}{N}(\vec{\mu}_j^{\omega_i} - \vec{\mu})(\vec{\mu}_j^{\omega_i} - \vec{\mu})^T \quad \text{with} \quad \vec{\mu} = \frac{1}{N}\sum_{j=1}^{N}\vec{v}_j = \sum_{j=1}^{N}\frac{N_i}{N}\vec{\mu}^{\omega_i} \qquad (5.44)$$

- One possible measure of the underlying separability is given by J_s with:

$$J_s = Tr(S_w^{-1}S_b) \qquad (5.45)$$

- Computation of Eigenvalues λ_i and Eigenvectors ψ_i of matrix $(S_w^{-1}S_b)$
- Largest Eigenvalues correspond to components with largest ***discriminance***
- Eigenvalues are sorted and the *m* Eigenvectors corresponding to the largest Eigenvalues are selected for dimensionality reduction:

$$\vec{y}_{i_{1..m}} = \Psi_{1..m}\vec{v}_i \qquad (5.46)$$

- Dimensionality is reduced and discriminance improved in one step
- For classification, *m* is chosen according to Scree plot of λ_i or one of the measures given in the taxonomy of section 2
- Visualization by 2 (3) ψ_i, for $m>2$ combination with unsupervised methods
- The parametric approach is limited in scope and ability !

Feature Extraction

- Extension of scatter matrices for nonparametric or even multimodal data [Fukunaga 90]
- Interclass scatter is computed based on k-nearest-neighbor technique for nonparametric scatter matrix $\mathbf{S}_b$ by:

$$S_b = \frac{1}{N}\sum_{i=1}^{L}\sum_{j=1}^{N_i} g_i(\vec{v}_j^{\omega_i} - \vec{\mu}_{jNN}^{\neq\omega_i})(\vec{v}_j^{\omega_i} - \vec{\mu}_{jNN}^{\neq\omega_i})^T \tag{5.47}$$

- Here, $\vec{\mu}_{jNN}^{\neq\omega_i} = \frac{1}{k}\sum_{l=1}^{k}\vec{v}_{l_{NN}}^{\neq\omega_i}$ denotes the mean vector of the k-nearest-neighbors of $\vec{v}_j^{\omega_i}$ with different class affiliations
- The weighting factor for off-class-borders vectors g_j is given by:

$$g_j = \frac{\min\{d^{\varsigma}(v_j^{\omega_i}, \mu_{jNN}^{\omega_i}), d^{\varsigma}(v_j^{\omega_i}, \mu_{jNN}^{\omega_i})\}}{d^{\varsigma}(v_j^{\omega_i}, \mu_{jNN}^{\omega_i}) + d^{\varsigma}(v_j^{\omega_i}, \mu_{jNN}^{\omega_i})} \tag{5.48}$$

with weight decay factor $\varsigma = [0, \infty[$

Feature Extraction

- The parametric intraclass scatter matrix S_w is retained
- Decorrelation with regard to S_w, i.e., S_w in space **Y** will be $S_w = \mathbf{I}$
- Pattern data is transformed by

$$\vec{y}_j = (\Psi_{1..M}^{w}(\Lambda^{w})^{-\frac{1}{2}})^T \vec{v}_j \tag{5.49}$$

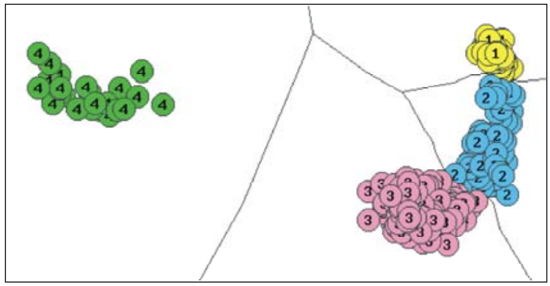
- In space Y, now $Tr(S_w^{-1}\mathbf{S}_b) = Tr(\mathbf{S}_b)$ holds (whitening according to S_w)
- Computation of Eigenvalues λ_i and Eigenvectors ψ_i of matrix $\mathbf{S}_b$
- Eigenvalues are sorted and the *m* Eigenvectors corresponding to the largest Eigenvalues are selected for dimensionality reduction:

$$\vec{y}_{j_{1..m}}^* = (\Psi_{1..m}^{b})^T \vec{v}_j^* \tag{5.50}$$

- Finally, the dimensionality reducing projection is computed by:

$$\vec{y}_{j_{1..m}}^* = (\Psi_{1..m}^{b})^T \vec{y}_j = (\Psi_{1..m}^{b})^T (\Psi_{1..M}^{w}(\Lambda^{w})^{-\frac{1}{2}})^T \vec{v}_j \tag{5.51}$$

- Visualization example of nonparametric scatter matrix (NPSCM) application for ***Mech$_1$*** data:



- Numerical assessment of the method with regard to discriminance will follow after presentation of nonlinear supervised mapping methods

- Koontz & Fukunaga extended Sammon´s stress in the NLM by an additional term dedicated to assessment of intraclass distances
- The KFM mapping pursues intraclass reduction & structure preservation and thus nonlinear discriminance analysis by the following cost function:

$$E(m) = \frac{1}{c}\sum_{j=1}^{N}\sum_{i=1}^{j}\frac{\delta(\omega_i,\omega_j)d_{Y_{ij}}^2(m) + \hat{\lambda}(d_{X_{ij}} - d_{Y_{ij}}(m))^2}{d_{X_{ij}}} \tag{5.52}$$

- Here, c, d_{Xij} and d_{Yij} are the same terms defined for Sammon´s stress in the NLM and

$$\delta(\omega_i,\omega_j) = \begin{cases} \hat{\alpha} & : \quad \omega_i = \omega_j \\ 0 & : \quad \omega_i \neq \omega_j \end{cases} \quad with \quad \hat{\alpha} = 1 \tag{5.53}$$

- Minimization of KFM cost function using gradient descent is achieved by iterative computation of new coordinates for the pivot vectors in **Y** space:

$$y_{iq}(m+1) = y_{iq}(m) - MF * \Delta y_{iq}(m) \qquad (5.54)$$

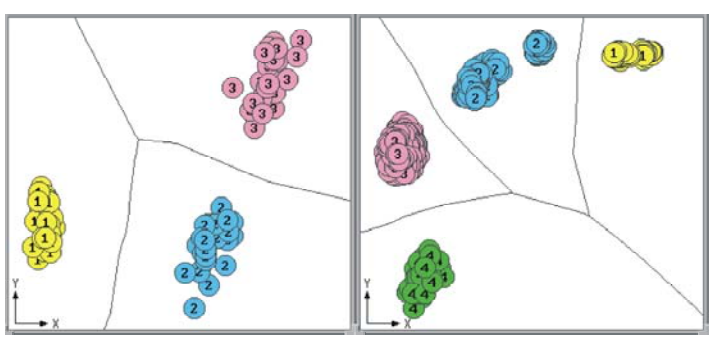
$$with \qquad \Delta y_{iq}(m) = \frac{\partial E(m)}{\partial y_{iq}(m)} \Big/ \left| \frac{\partial^2 E(m)}{\partial y_{iq}(m)^2} \right|$$

- The partial derivatives for gradient descent are given by

$$\frac{\partial E(m)}{\partial y_{iq}} = \frac{2}{c} \sum_{\substack{j=1 \\ j \neq i}}^{N} \left[\frac{\delta(\omega_i, \omega_j)}{d_{X_{ij}}} - \hat{\lambda} \left(\frac{1}{d_{Y_{ij}}} - \frac{1}{d_{X_{ij}}} \right) \right] (y_{iq} - y_{jq}) \qquad (5.55)$$

$$\frac{\partial^2 E(m)}{\partial y_{iq}{}^2} = \frac{2}{c} \sum_{\substack{j=1 \\ j \neq i}}^{N} \left[\frac{\delta(\omega_i, \omega_j)}{d_{X_{ij}}} - \hat{\lambda} \left(\frac{1}{d_{Y_{ij}}} - \frac{1}{d_{X_{ij}}} - \frac{(y_{iq} - y_{jq})^2}{d_{Y_{ij}}{}^3} \right) \right] \qquad (5.56)$$

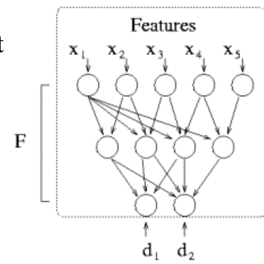
- Modification of the original approach by setting $\hat{\alpha} = 1 - \hat{\lambda}$ reduces parameters and gives convenient control from pure structure preservation ($\hat{\lambda} = 0.0$) to pure separability achievement ($\hat{\lambda} = 1.0$)
- Visualization of ***Iristrain*** and ***Mech$_1$*** with $\hat{\lambda} = 0.3$ after 100 iterations



Feature Extraction

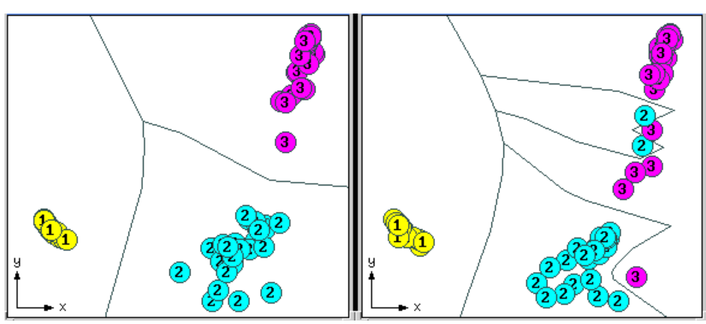
- As for NLM, a recall procedure is not straight forward available for KFM
- Koontz & Fukunaga presented a (restricted) method using polynomial distance approximation and a pivot point approach for mapping
- Employment of neural network, e.g., BP, as universal function approximator is an interesting alternative

- Network is supplied with original feature data at the input & KFM data at the output
- Training phase aims on mapping error minimization with generalization
- Test data is mapped later for classification or visualization employing the trained net
- **Problem:** Net configuration & convergence



Feature Extraction

- Visualization of KFM applied to ***Iristrain*** and ***Iristest*** data with a 4-9-5-2 BP network (training & test data mapped by BP network !)

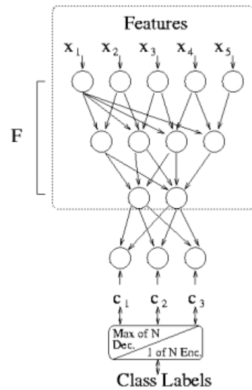


- Achieving network convergence & good approximation properties is not straightforward for combined KFM/BP nonlinear mapping

Feature Extraction

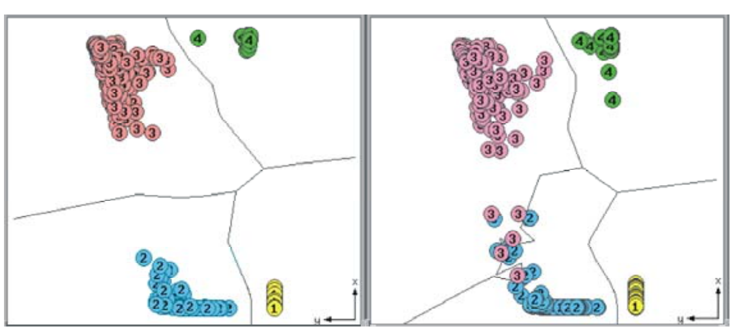
- Employment of BP neural network with a bottleneck topology for nonlinear mapping or nonlinear discriminance analysis (NDA) is an interesting alternative:

- Network is supplied with original feature data at the input & class data at the output
- Training phase aims on classification error minimization with given bottleneck
- The number of neurons in the bottleneck layer determine projection dimension ***d***
- The trained net is cut after bottleneck
- Test data is mapped later for classification or visualization employing the cut net
- **Problem:** Net configuration & convergence



Feature Extraction

- Result of BP_NDA for ***Mech$_1$*** and ***Mech$_2$*** data employing a 24-9-2-4 BP neural network with a bottleneck topology (d=2):



- Fast and rather easy training, only 130-150 epochs were required
- Much easier & faster than KFM/BP, but no structure preservation

Feature Extraction

- Compression from M- to 2D data for linear/nonlinear supervised methods
- Evaluation by overlap q_o and separability q_s measures for training & test sets

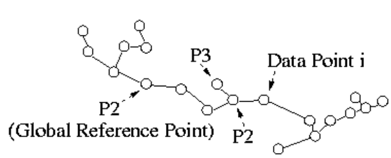
Method	Dim.	Train	q_o	q_s	Test	q_o	q_s
Original	4	Iristrain	0.95503	0.90666	Iristest	0.91683	0.88000
NPSCM	2	Iristrain	0.98224	0.94666	Iristest	0.95788	0.90666
BP	2	Iristrain	0.97536	1.000	Iristest	0.95295	0.94667
KFM/BP	2	Iristrain	1.000	1.000	Iristest	0.94023	0.92000
Original	24	$Mech_1$	1.000	0.98933	$Mech_2$	0.99799	0.96308
NPSCM	2	$Mech_1$	0.99908	0.98200	$Mech_2$	0.98716	0.97536
BP	2	$Mech_1$	0.99988	0.99733	$Mech_2$	0.99373	0.99385

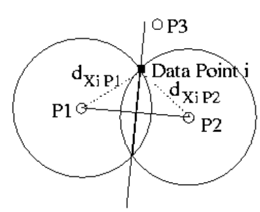
- KFM with trained BP network (only for ***Iris*** data, hard to train for ***Mech*** data)
- BP with 24-9-2-4 network topology for $Mech_1$ fast to train & good results
- Linear NPSCM is simple and excellent when applicable to the given data
- Nonlinear BP, KFM/BP outperform NPSCM. Problem: overfitting/overlap
- BP & KFM/BP similar but KFM/BP results are very hard to obtain

Accelerated Methods

Lee, Slaggle & Blum´s Triangulation Mapping:

- Compute *Minimal-Spanning-Tree* (MST) of the data (O(N^2) complexity !)
- Select initial data points from the MST and map to 2D space
- Step through MST & map each of the remaining points by triangulation





- Only 2N-3 distances are exactly preserved
- The algorithm tends to unfold circular structures, misleading in analysis !
- Option: Choose a fixed, global reference point for focusing the mapping (ROI)

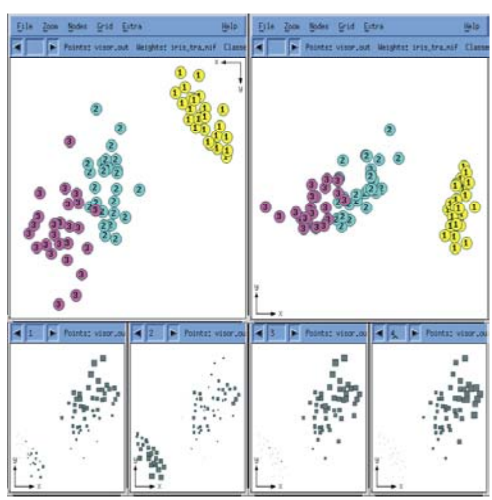
Accelerated Methods

Visor Mapping [König et al. 94]:

- Fast mapping with only O(N) complexity as data previewer
- Bases on the assumption to find three suited global pivot points for ensuing triangulation mapping of the remaining N-3 data points
- Mapping steps (no parameters):
  1. Compute centroid of the data set
  2. Determine data point most distant from centroid
  3. Determine two more data points with maximum from centroid & each other
  4. Place these pivot points in the 2D-plane, exactly preserving their mutual distances
  5. Place the remaining N-3 points by triangulation based on pivot points
- The concept is amenable to enhancement by reference or multiple pivot points

Accelerated Methods ,,

- NLM and Visor mapping results are displayed for ***Iristrain*** data
- Mappings can be subject to rotation & mirroring
- Basic global data structure is quite similar
- The relevant information, e.g., for ***classification system design*** can be extracted by fast preview
- Quantitative analysis of mapping error required !



Summary

- The chapter addressed the issue of decreasing the dimensionality of data sets to achieve visualization and/or improved classification
- A survey of the basic idea and selected conventional as well as neural supervised and unsupervised methods was provided
- In particular, pontential cost functions for assessment and optimization of dimensionality reducing mappings have been regarded and applied
- Special attention was given to automatic feature selection (AFS)
- AFS allows the selection of a subset of salient attributes or features from an initially large set and the pruning of the architecture to a lean system
- Another focus was on unsupervised methods, e.g., NLM, to project arbitrary, prepferably to two dimensions in the wake of system design
- Numerous more advanced and more recent methods can be found today
- Application is in system design and system operation