Chapter 8:

Neuro-Fuzzy Systems

Neuro-Fuzzy Systems

- Building a fuzzy system requires
  - prior knowledge (fuzzy rules, fuzzy sets)
  - manual tuning: *time consuming and error-prone*
- Therefore: Support this process by learning
  - learning fuzzy rules (structure learning)
  - learning fuzzy set (parameter learning)

Approaches from Neural Networks can be used

- Database: time series from 1986 - 1997

DAX	Composite DAX
German 3 month interest rates	Return Germany
Morgan Stanley index Germany	Dow Jones industrial index
DM / US-$	US treasury bonds
Gold price	Nikkei index Japan
Morgan Stanley index Europe	Price earning ratio

Fuzzy Rules in Finance

- Trend Rule

  IF DAX = decreasing AND US-$ = decreasing

  THEN DAX prediction = decrease

  WITH high certainty

- Turning Point Rule

  IF DAX = decreasing AND US-$ = increasing

  THEN DAX prediction = increase

  WITH low certainty

- Delay Rule

  IF DAX = stable AND US-$ = decreasing

  THEN DAX prediction = decrease

  WITH very high certainty

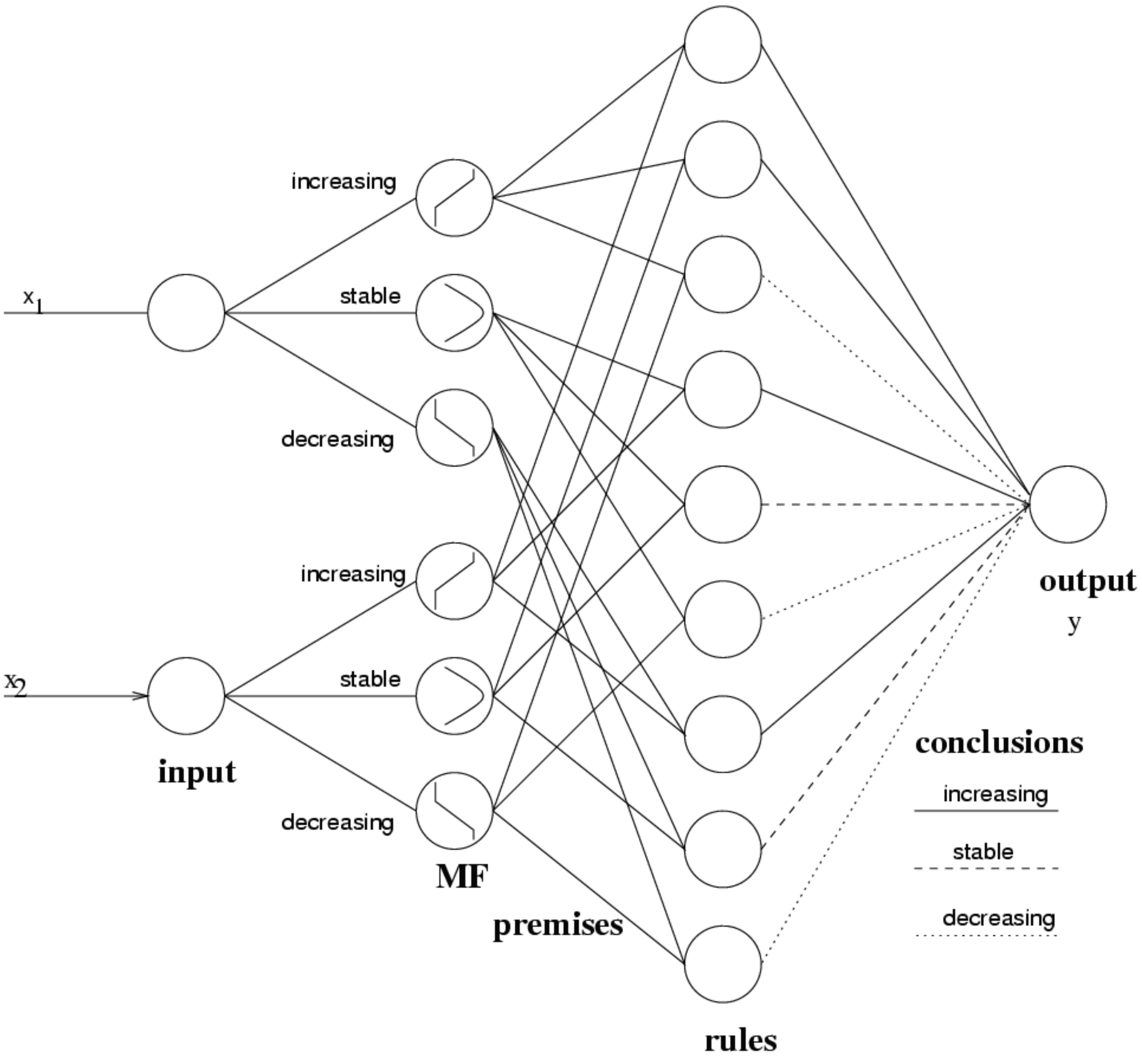
- In general

  IF x_1 is μ_1 AND x_2 is μ_2

  THEN $y = \eta$

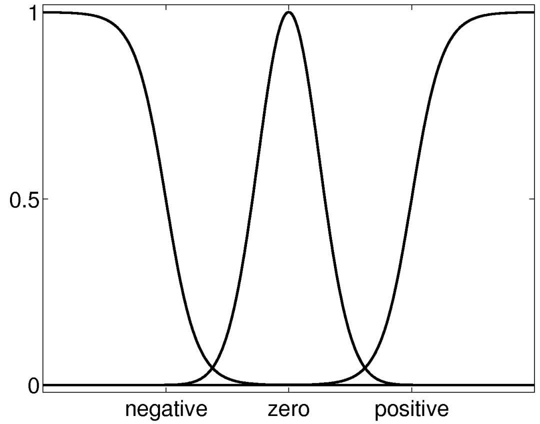
  WITH weight k

Neuro-Fuzzy Architecture



From Rules to Neural Networks

1. Evaluation of membership degrees



2. Evaluation of rules (rule activity)

$$\mu_l\colon \mathbb{R}^n \to [0,1]^r, \quad \underline{x} \Rightarrow \prod_{j=1}^{D_l} \mu_{c,s}^{(j)}(x_i)$$

3. Accumulation of rule inputs and normalization

$$\mathrm{NF}\colon \mathbb{R}^n \to \mathbb{R}, \quad \underline{x} \Rightarrow \sum_{l=1}^{r} w_l \frac{k_l \mu_l(\underline{x})}{\sum_{j=1}^{r} k_j \mu_j(\underline{x})}$$

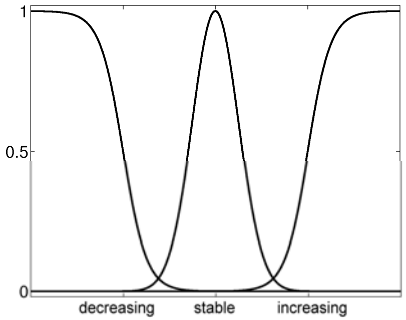
Reduction of the dimension of the weight space

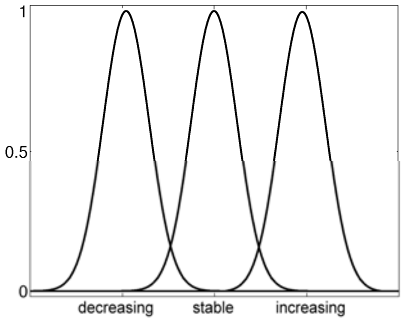
1. Membership functions of different inputs share their parameters, e.g.

$$\mu_{dax}^{stable} \equiv \mu_{cdax}^{stable}$$

2. Membership functions of the same input variable are not allowed to pass each other, they must keep their original order, e.g.

$$\mu^{decreasing} < \mu^{stable} < \mu^{increasing}$$

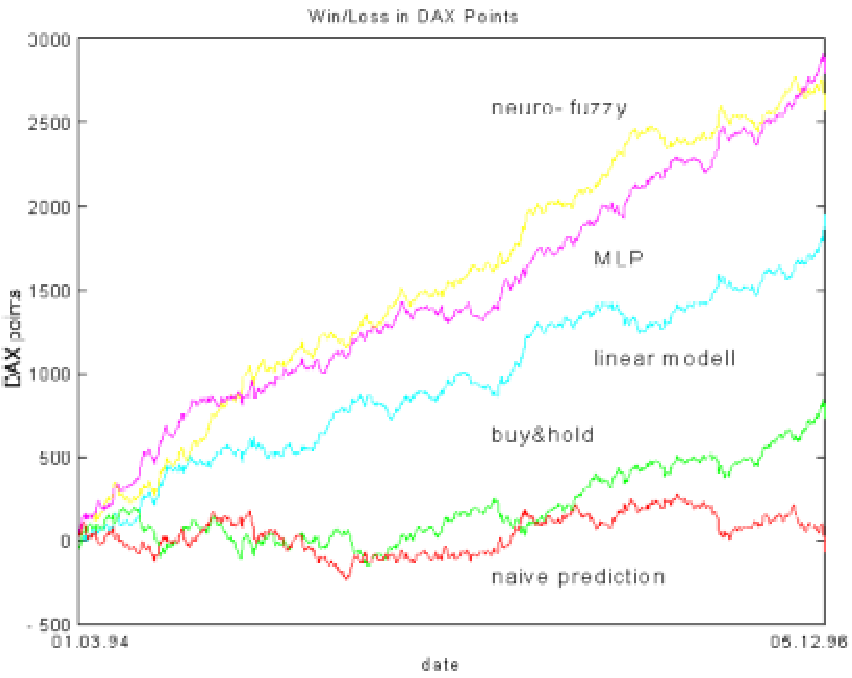


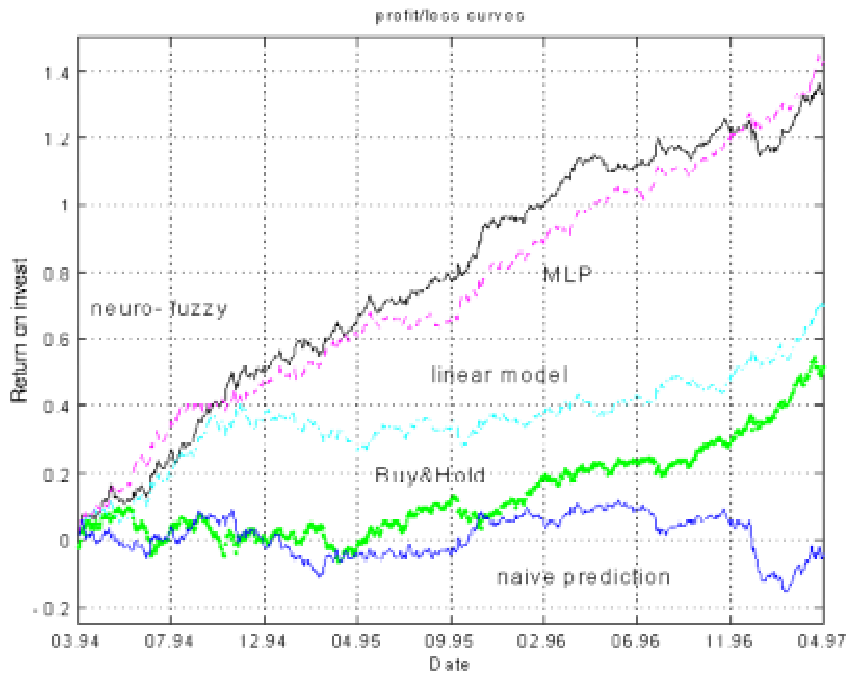


Benefits:
- the optimized rule base can still be interpreted
- the number of free parameters is reduced

Return-on-Investment Curves of the Different Models

Validation data from March 01, 1994 until April 1997





Neuro-Fuzzy Systems in Data Analysis

- Neuro-Fuzzy System:
  - System of linguistic rules (fuzzy rules).
  - Not rules in a logical sense, but function approximation.
  - Fuzzy rule = vague prototype / sample.

- Neuro-Fuzzy-System:
  - Adding a learning algorithm inspired by neural networks.
  - Feature: local adaptation of parameters.

A Neuro-Fuzzy System

- is a fuzzy system trained by heuristic learning techniques derived from neural networks
- can be viewed as a 3-layer neural network with fuzzy weights and special activation functions
- is always interpretable as a fuzzy system
- uses constraint learning procedures
- is a function approximator (classifier, controller)

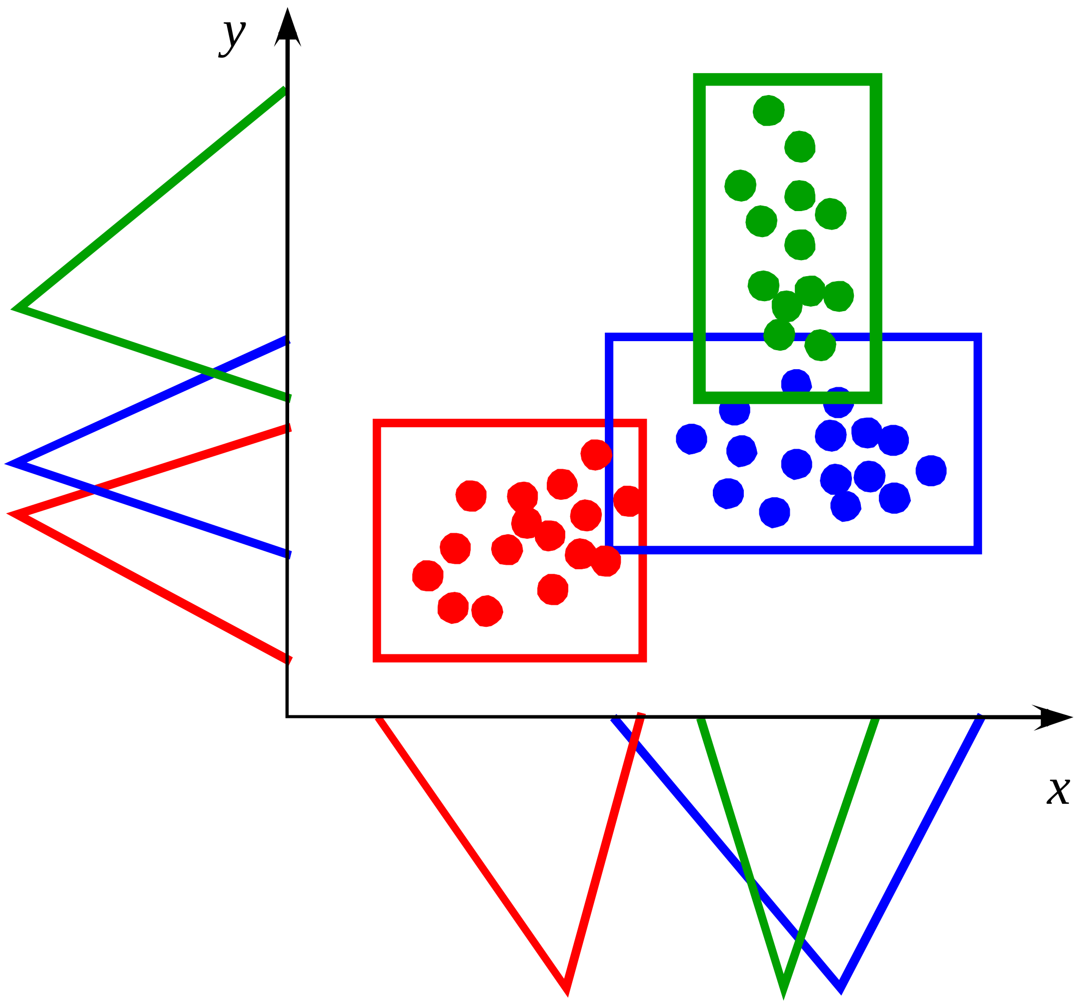
Learning Fuzzy Rules

- Cluster-oriented approaches
  => find clusters in data, each cluster is a rule

- Hyperbox-oriented approaches
  => find clusters in the form of hyperboxes

- Structure-oriented approaches
  => used predefined fuzzy sets to structure the
  data space, pick rules from grid cells

Hyperbox-Oriented Rule Learning



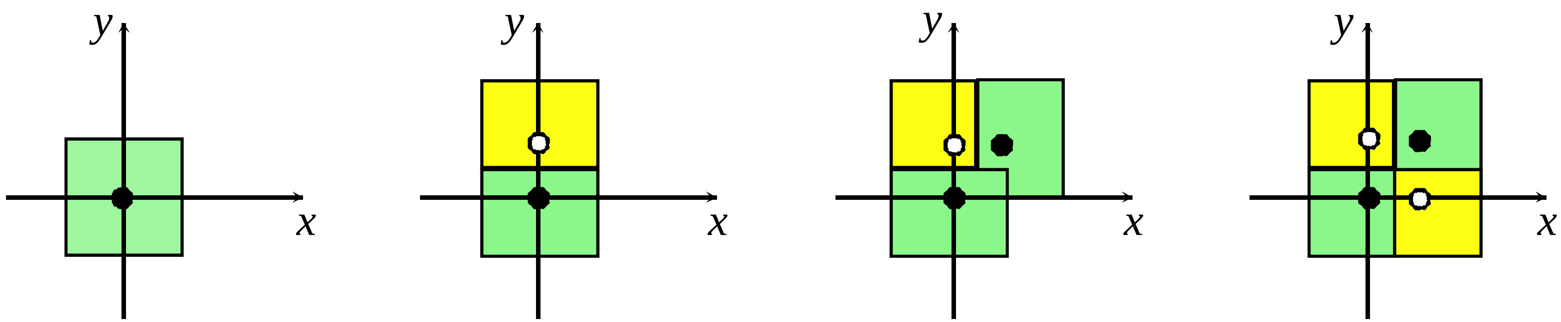
Search for hyperboxes in the data space

Create fuzzy rules by projecting the hyperboxes

Fuzzy rules and fuzzy sets are created at the same time

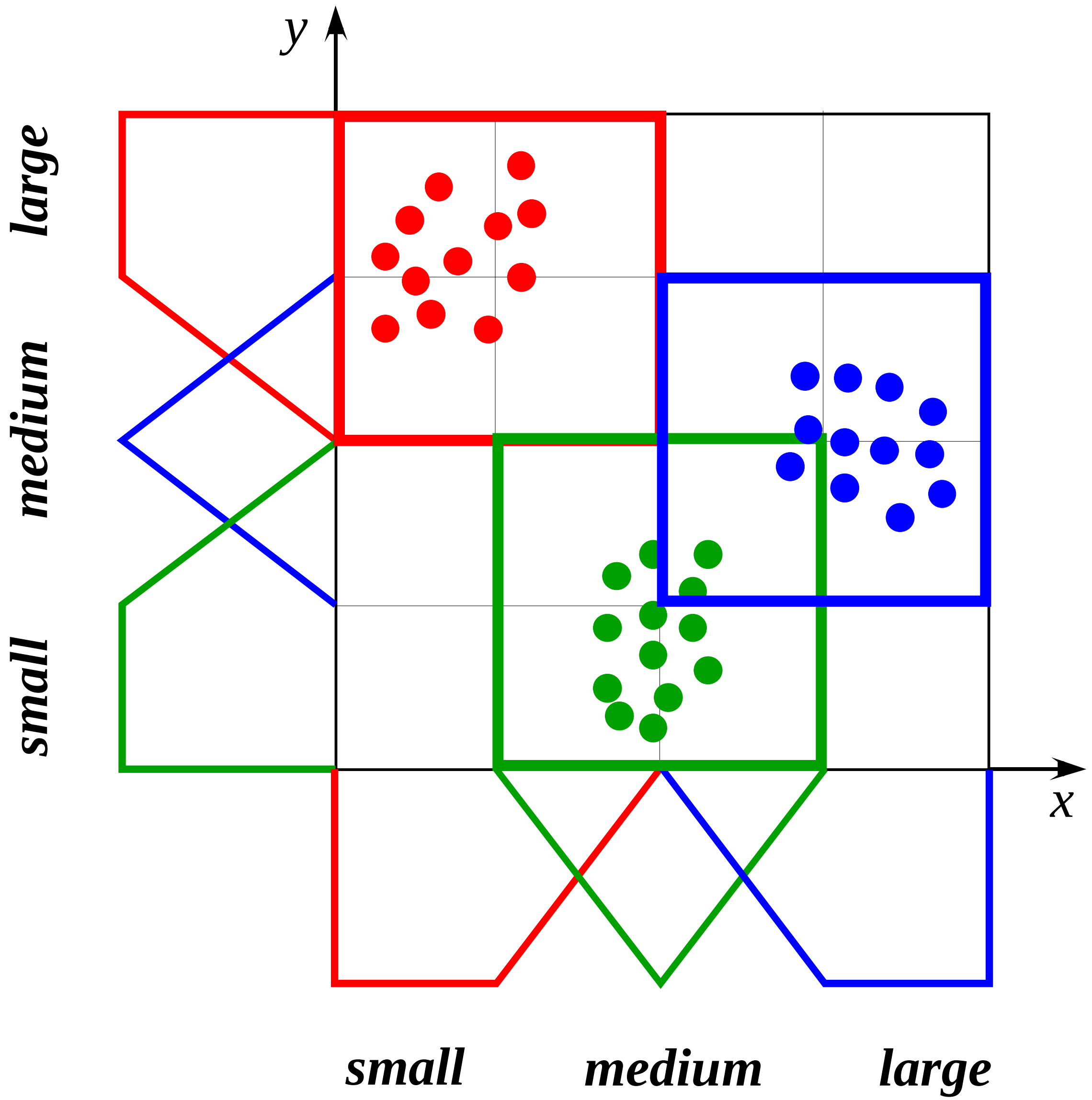
Usually very fast

Hyperbox-Oriented Rule Learning



- Detect hyperboxes in the data, example: XOR function
- Advantage over fuzzy cluster anlysis:
  - No loss of information when hyperboxes are represented as fuzzy rules
  - Not all variables need to be used, don‘t care variables can be discovered
- Disadvantage: each fuzzy rules uses individual fuzzy sets, i.e. the rule base is complex.

Structure-Oriented Rule Learning



Provide initial fuzzy sets for all variables.

The data space is partitioned by a fuzzy grid

Detect all grid cells that contain data (approach by Wang/Mendel 1992)

Compute best consequents and select best rules (extension by Nauck/Kruse 1995, NEFCLASS model)

Structure-Oriented Rule Learning

- Simple: Rule base available after two cycles through the training data
  - 1. Cycle: discover all antecedents
  - 2. Cycle: determine best consequents

- Missing values can be handled
- Numeric and symbolic attributes can be processed at the same time (mixed fuzzy rules)

- Advantage: All rules share the same fuzzy sets
- Disadvantage: Fuzzy sets must be given

Learning Fuzzy Sets

- Gradient descent procedures
  only applicable, if differentiation is possible, e.g. for Sugeno-type fuzzy systems.

- Special heuristic procedures that do not use gradient information.

- The learning algorithms are based on the idea of backpropagation.

Learning Fuzzy Sets: Constraints

- Mandatory constraints:
  - Fuzzy sets must stay normal and convex
  - Fuzzy sets must not exchange their relative positions (they must not „pass“ each other)
  - Fuzzy sets must always overlap
- Optional constraints
  - Fuzzy sets must stay symmetric
  - Degrees of membership must add up to 1.0
- The learning algorithm must enforce these constraints.

Example: Medical Diagnosis

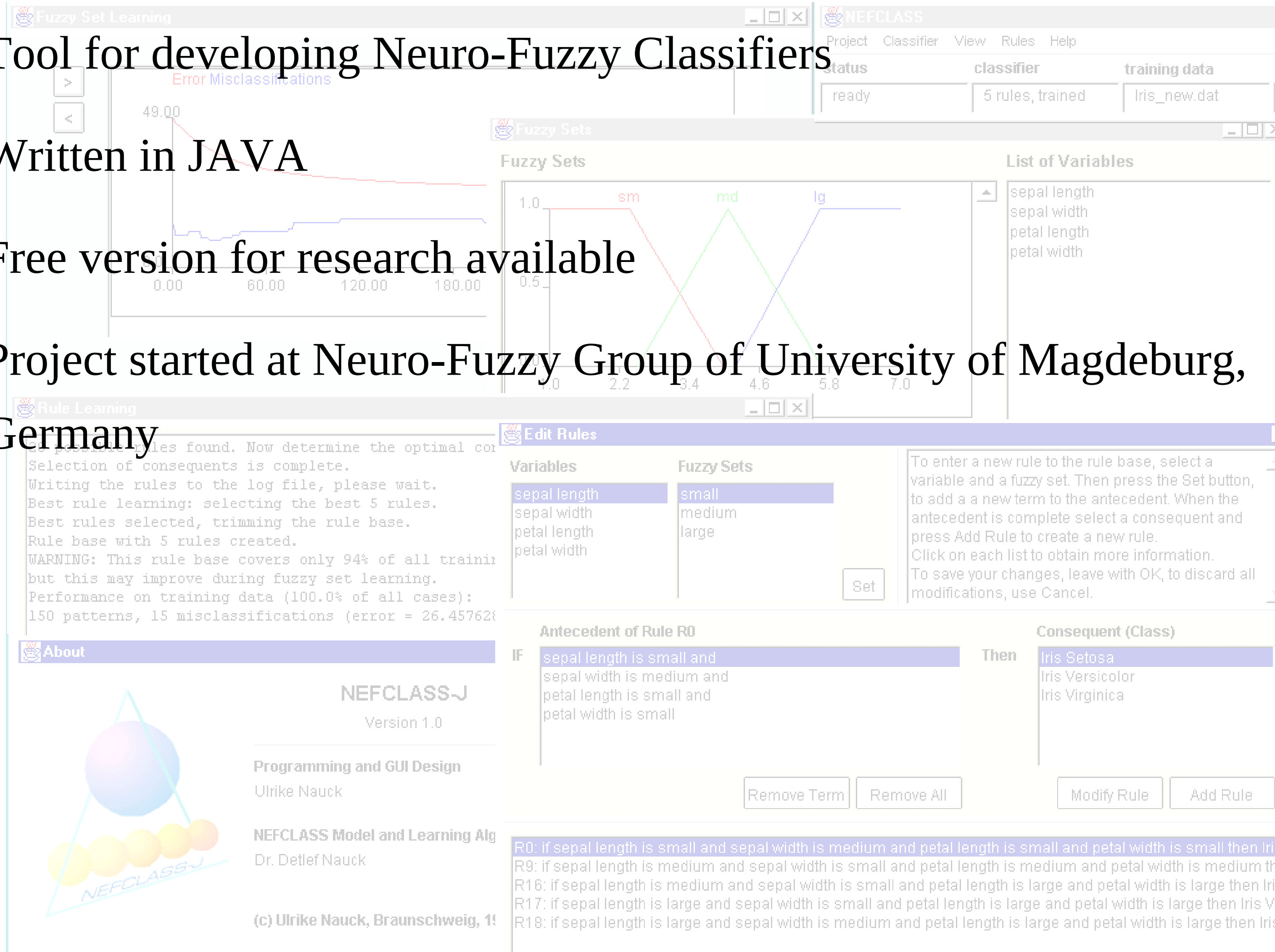
- Results from patients tested for breast cancer (Wisconsin Breast Cancer Data).
- Decision support: Do the data indicate a malignant or a benign case?
- A surgeon must be able to check the classification for plausibility.
- We are looking for a simple and interpretable classifier: ⇨**knowledge discovery**.

Example: WBC Data Set

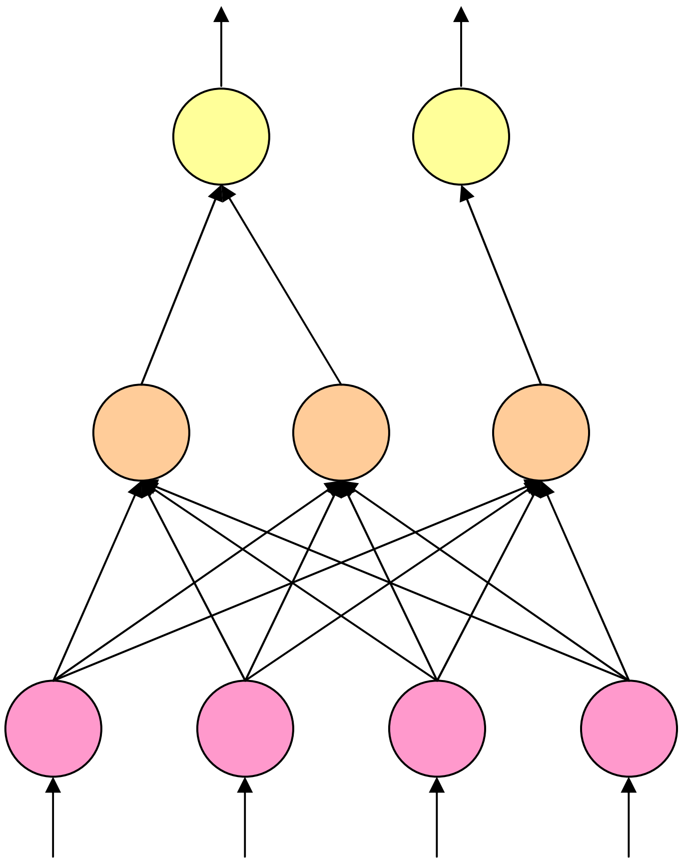
- 699 cases (16 cases have missing values).
- 2 classes: benign (458), malignant (241).
- 9 attributes with values from {1, ... , 10}
  (ordinal scale, but usually interpreted as a numerical scale).
- Experiment: x_3 and x_6 are interpreted as nominal attributes.
- x_3 and x_6 are usually seen as „important“ attributes.

Applying NEFCLASS-J

- Tool for developing Neuro-Fuzzy Classifiers
- Written in JAVA
- Free version for research available
- Project started at Neuro-Fuzzy Group of University of Magdeburg, Germany



NEFCLASS: Neuro-Fuzzy Classifier



Output variables (class labels)

Unweighted connections

Fuzzy rules

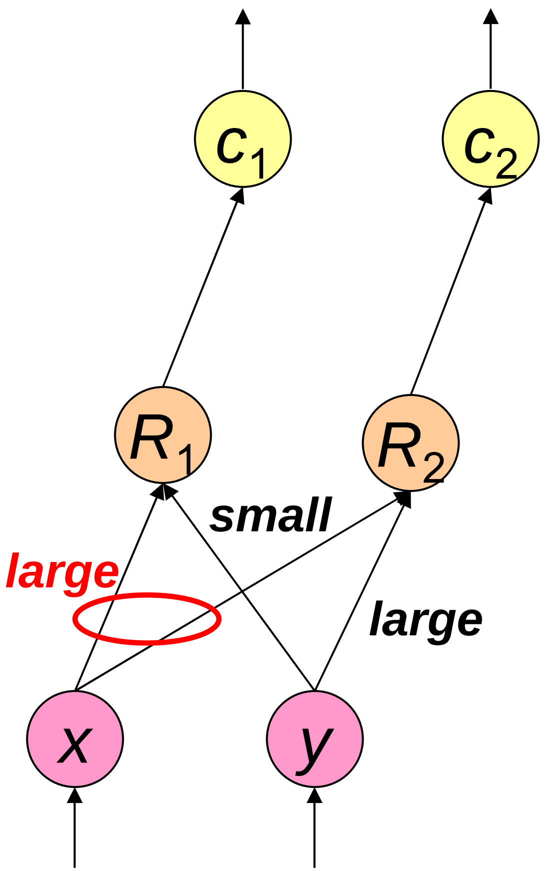
Fuzzy sets (antecedents)

Input variables (attributes)

NEFCLASS: Features

- Automatic induction of a fuzzy rule base from data
- Training of several forms of fuzzy sets
- Processing of numeric and symbolic attributes
- Treatment of missing values (no imputation)
- Automatic pruning strategies
- Fusion of expert knowledge and knowledge obtained from data

Representation of Fuzzy Rules



Example: 2 Rules

R_1: if x is *large* and y *is small*, then class is c_1.

R_2: if x *is large* and y *is large*, then class is c_2.

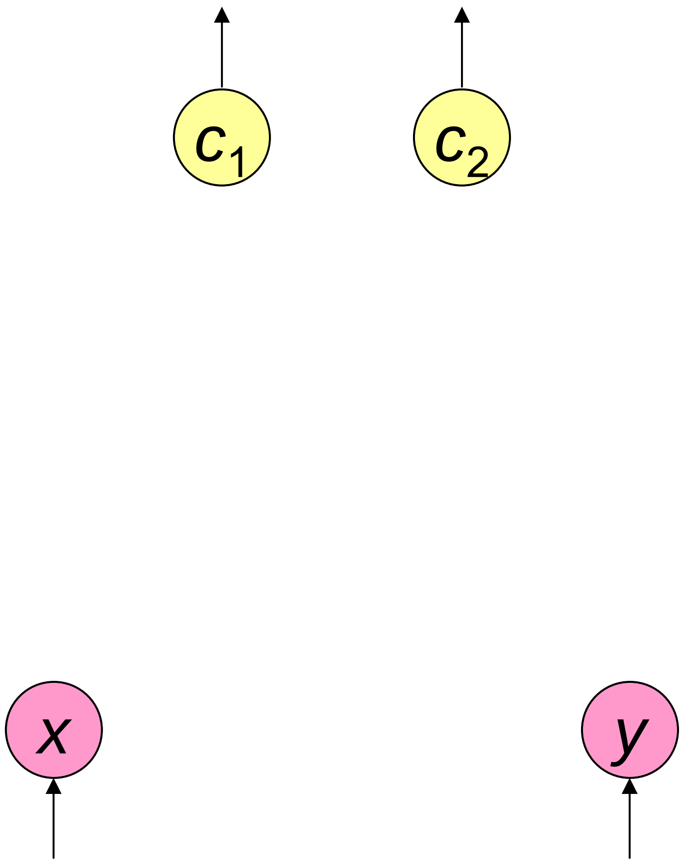
The connections $x \rightarrow R_1$ and $x \rightarrow R_2$ are linked.

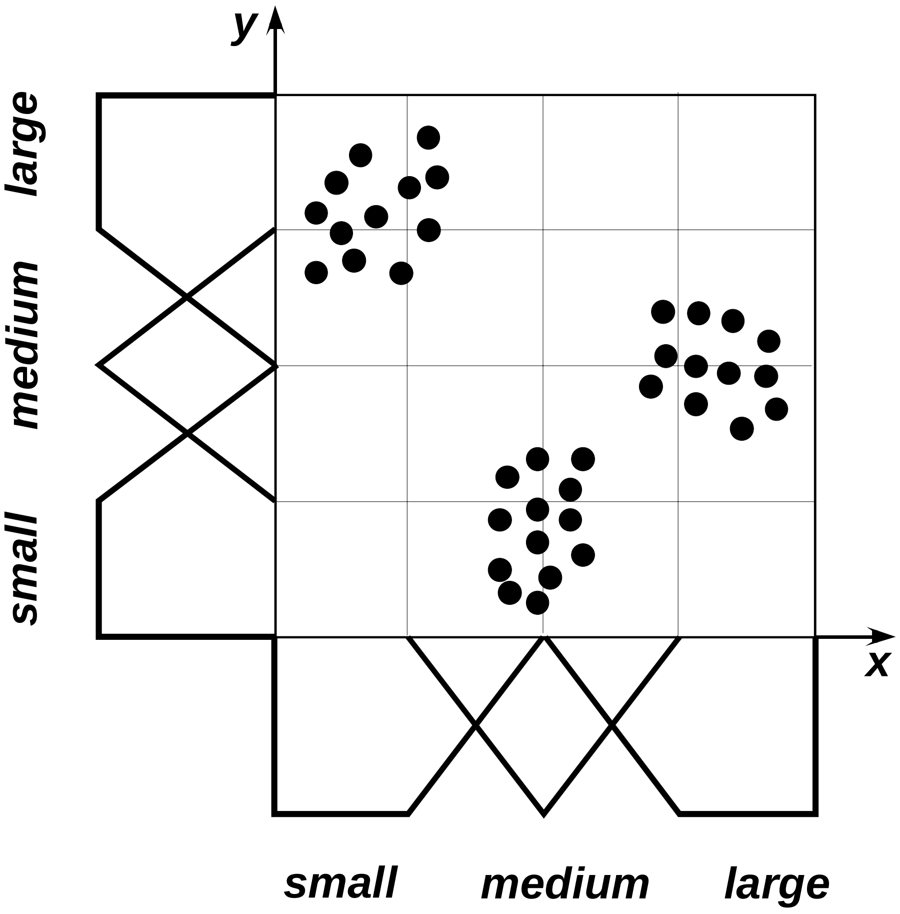
The fuzzy set *large* is a shared weight.

That means the term *large* has always the same meaning in both rules.

1. Training Step: Initialisation

Specify initial fuzzy partitions for all input variables





2. Training Step: Rule Base

Algorithm:

for (all patterns p) **do**
  find antecedent A,
  such that $A(p)$ is maximal;
  if ($A \notin L$) **then** add A to L;
end;

for (all antecedents $A \in L$) **do**
  find best consequent C for A;
  create rule base candidate $R = (A,C)$;
  Determine the performance of R;
  Add R to B;
end;

Select a rule base from B;

Variations:

Fuzzy rule bases can also be created by using prior knowledge, fuzzy cluster analysis, fuzzy decision trees, genetic algorithms, ...

Selection of a Rule Base

Performance of a Rule:

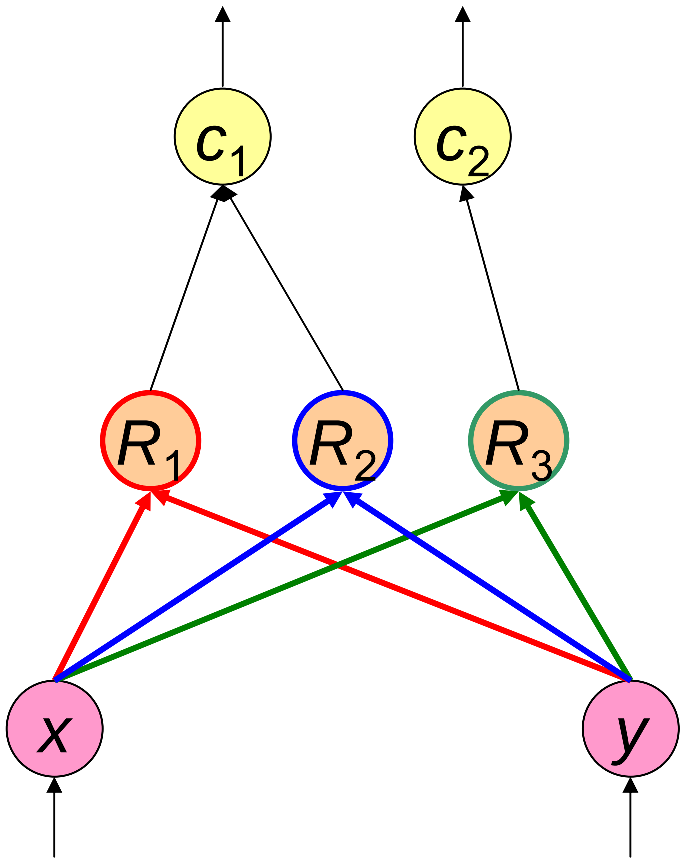
$$P_r = \frac{1}{N}\sum_{p=1}^{N}(-1)^c R_r(\mathbf{x}_p), \text{ with}$$

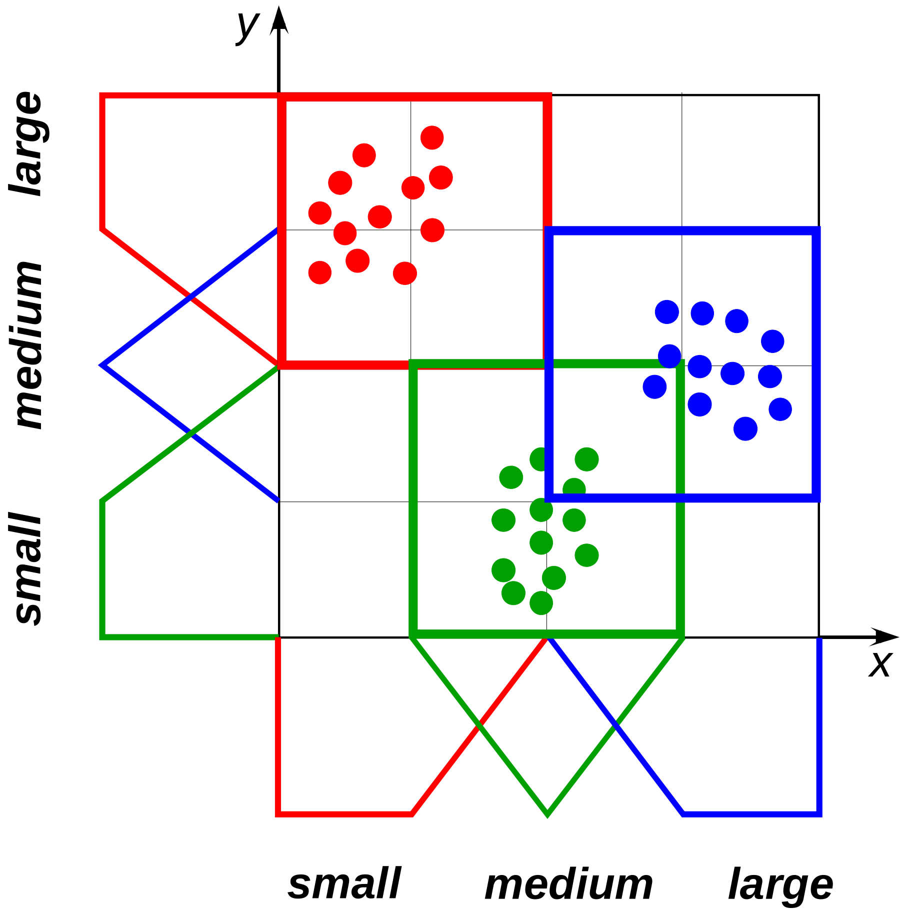
$$c = \begin{cases} 0 & \text{if } \mathrm{class}(\mathbf{x}_p) = \mathrm{con}(R_r), \\ 1 & \text{otherwise.} \end{cases}$$

- Order rules by performance.
- Either select the best r rules or the best r/m rules per class.
- r is either given or is determined automatically such that all patterns are covered.

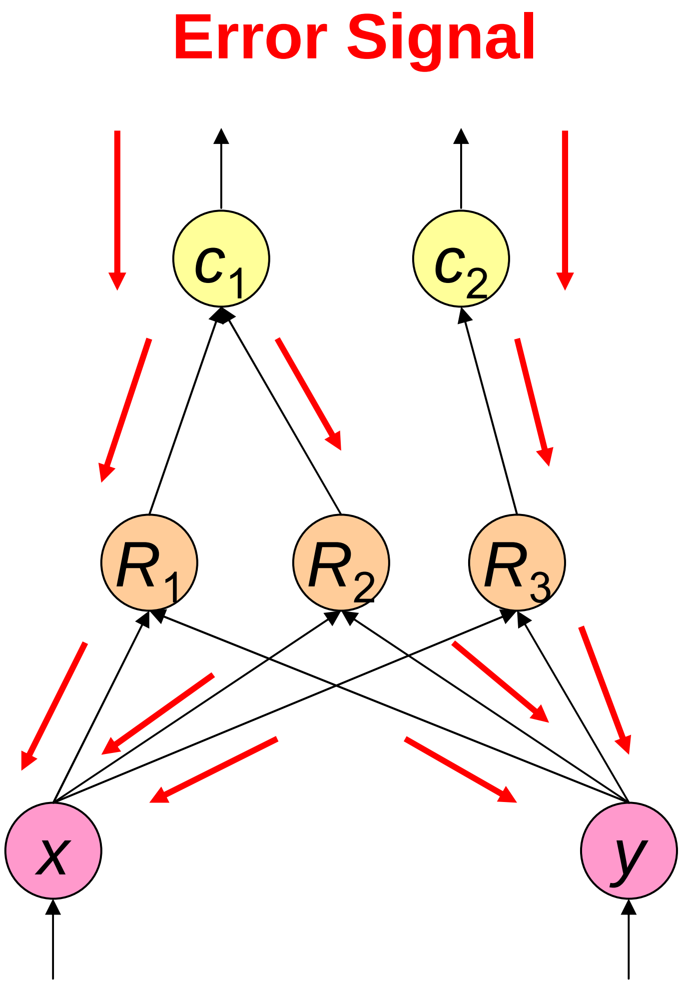
Rule Base Induction

NEFCLASS uses a modified Wang-Mendel procedure





Computing the Error Signal



Fuzzy Error (*j*th output):

$$E_j = \operatorname{sgn}(d)\cdot\left(1-\gamma(d)\right), \text{ with } d = t_j - o_j$$

$$\text{and } \gamma : \Re \to [0, 1],\ \gamma(d) = e^{-\left(\frac{a\cdot d}{d_{\max}}\right)^2}$$

(t : correct output, o : actual output)

Rule Error:

$$E_r = \left(\tau_r\left(1-\tau_r\right)+\varepsilon\right)E_{\text{con}(R_r)}, \text{ with } 0 < \varepsilon \ll 1$$

3. Training Step: Fuzzy Sets

Example: triangular membership function.

$$\mu_{a,b,c} : \Re \to [0,1],\ \mu_{a,b,c}(x) = \begin{cases} \dfrac{x-a}{b-a} & \text{if } x \in [a,b) \\ \dfrac{c-x}{c-b} & \text{if } x \in [b,c] \\ 0 & \text{otherwise} \end{cases}$$

Parameter updates for an antecedent fuzzy set.

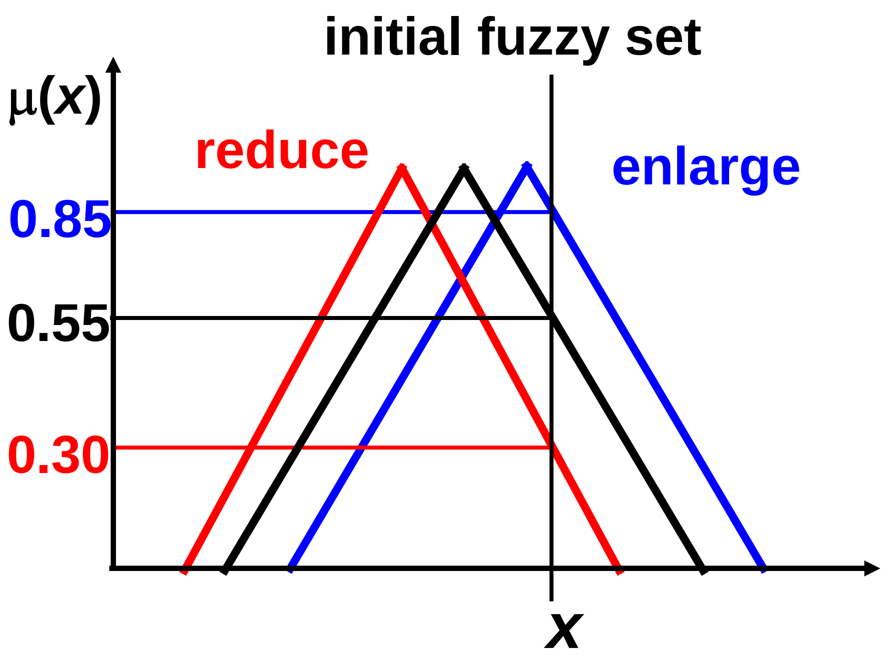
$$f = \begin{cases} \sigma\ \mu(x) & \text{if } \mathsf{E} < 0 \\ \sigma\ (1-\mu(x)) & \text{otherwise} \end{cases}$$

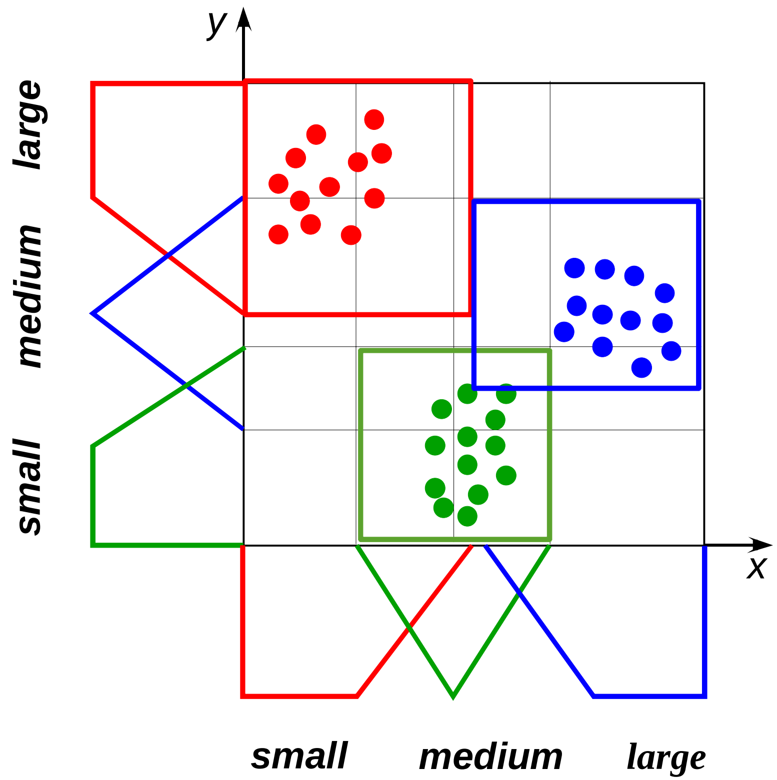
$$\Delta b = f \cdot \mathsf{E} \cdot (c-a) \cdot \operatorname{sgn}(x-b)$$

$$\Delta a = -f \cdot \mathsf{E} \cdot (b-a) + \Delta b$$

$$\Delta c = f \cdot \mathsf{E} \cdot (c-b) + \Delta b$$

Training of Fuzzy Sets



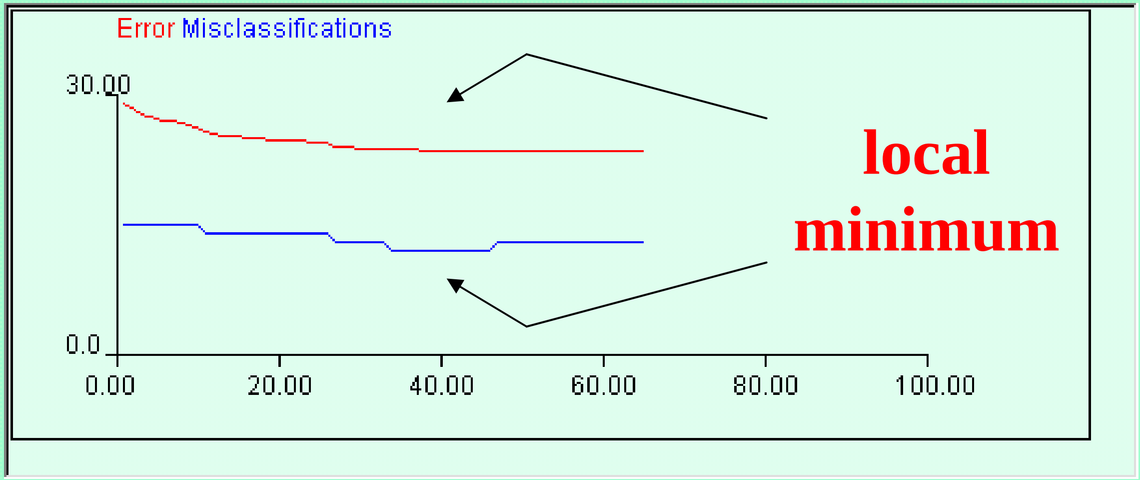


Heuristics: a fuzzy set is moved away from *x* (towards *x*) and its support is reduced (enlarged), in order to reduce (enlarge) the degree of membership of x.

Training of Fuzzy Sets

Algorithm:

repeat
    for (all patterns) **do**
        accumulate parameter updates;
        accumulate error;
    end;
    modify parameters;
until (no change in error);



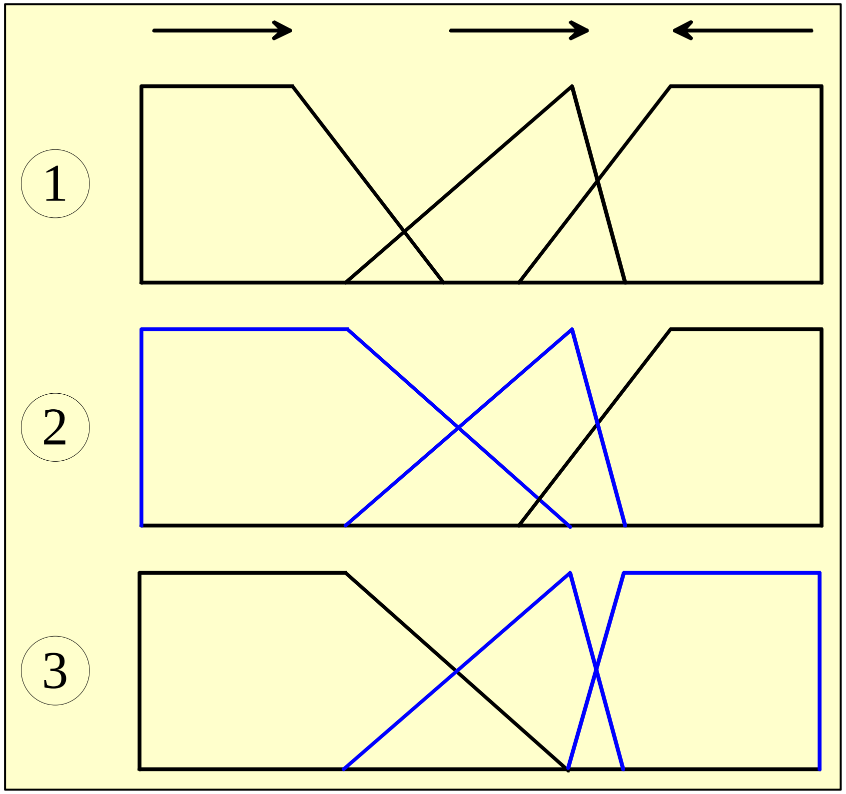
Variations:

- Adaptive learning rate
- Online-/Batch Learning
- optimistic learning (n step look ahead)

Observing the error on a validation set

Constraints for Training Fuzzy Sets

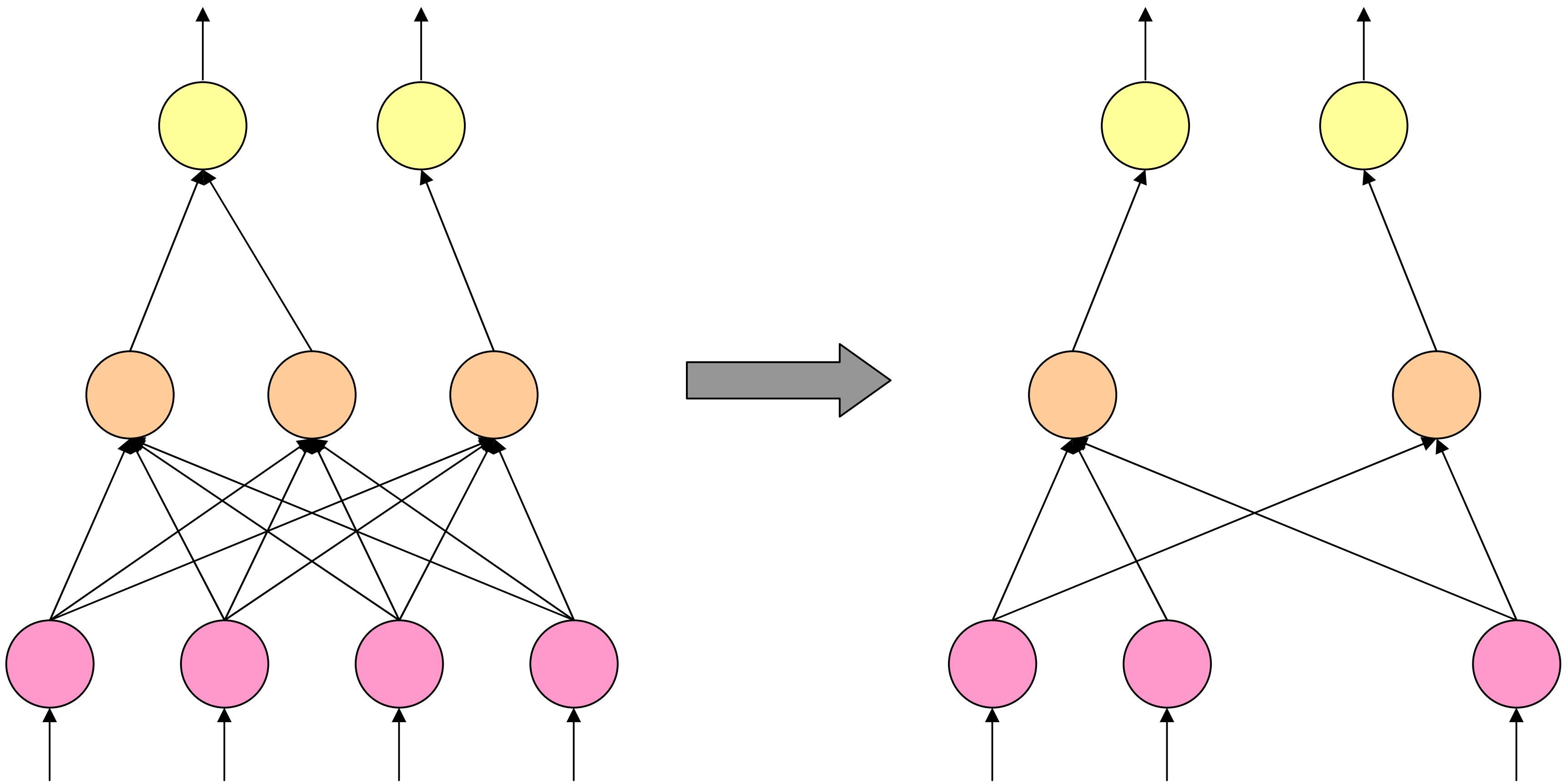
- Valid parameter values
- Non-empty intersection of adjacent fuzzy sets
- Keep relative positions
- Maintain symmetry
- Complete coverage (degrees of membership add up to 1 for each element)



Correcting a partition after modifying the parameters

4. Training Step: Pruning

Goal: remove variables, rules and fuzzy sets, in order to improve interpretability and generalisation.



Pruning

Algorithm:

repeat
  select pruning method;

  repeat
    execute pruning step;
    train fuzzy sets;

    if (no improvement)
    then undo step;

  until (no improvement);

until (no further method);

Pruning Methods:

1. Remove variables
   (use correlations, information gain etc.)

2. Remove rules
   (use rule performance)

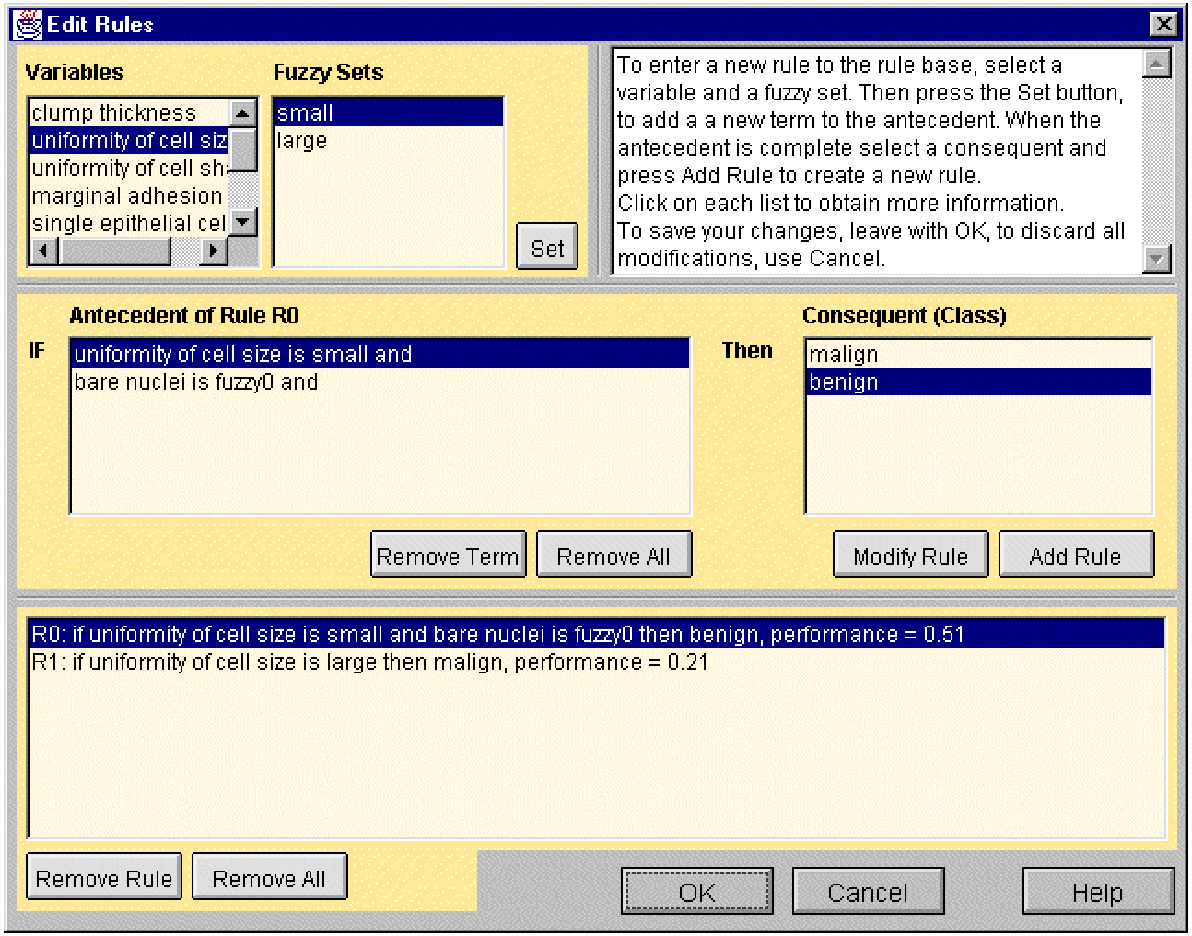
3. Remove terms
   (use degree of fulfilment)

4. Remove fuzzy sets
   (use fuzziness)

WBC Learning Result: Fuzzy Rules

R_1: if uniformity of cell size is *small* and bare nuclei is fuzzy0 then *benign*

R_2: if uniformity of cell size is *large* then *malignant*



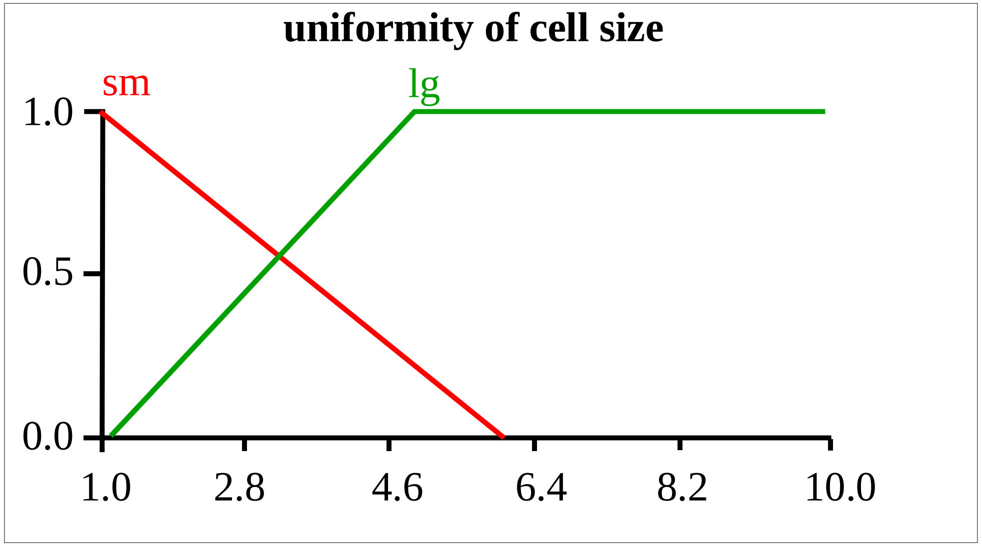
WBC Learning Result: Classification Performance

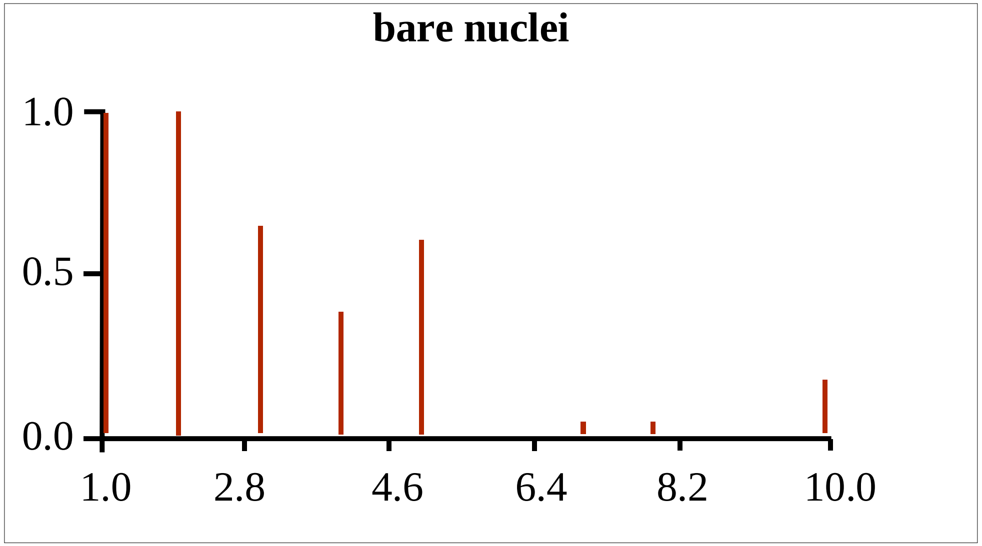
	Predicted Class			
	malign	benign	not classified	sum
malign	228 (32.62%)	13 (1.86%)	0 (0%)	241 (34.99%)
benign	15 (2.15%)	443 (63.38%)	0 (0%)	458 (65.01%)
sum	243 (34.76%)	456 (65.24%)	0 (0%)	699 (100.00%)

Estimated Performance on Unseen Data (Cross Validation)

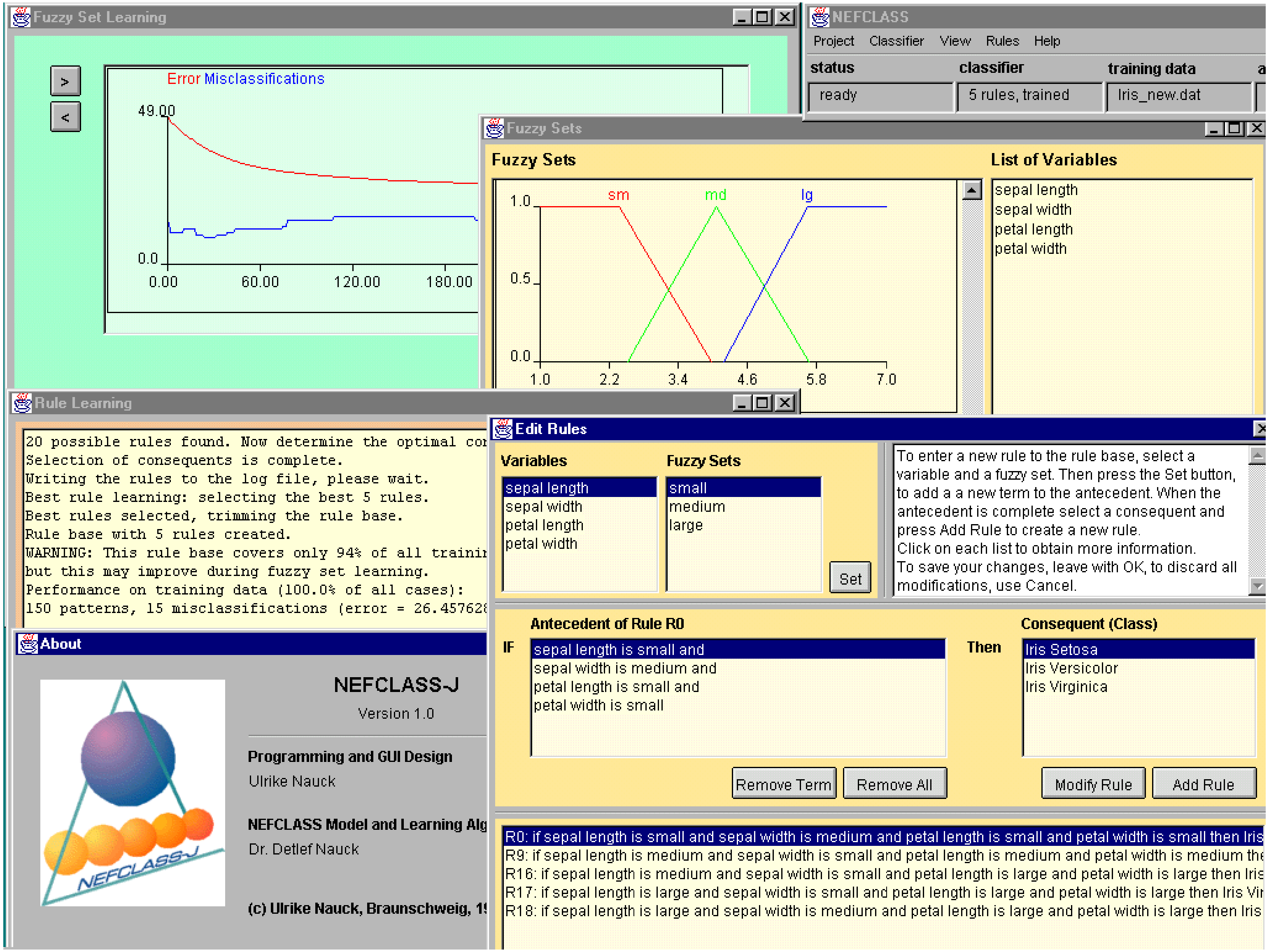
- NEFCLASS-J: 95.42%
- Discriminant Analysis: 96.05%
- C 4.5: 95.10%
- NEFCLASS-J (numeric): 94.14%
- Multilayer Perceptron: 94.82%
- C 4.5 Rules: 95.40%

WBC Learning Result: Fuzzy Sets





NEFCLASS-J



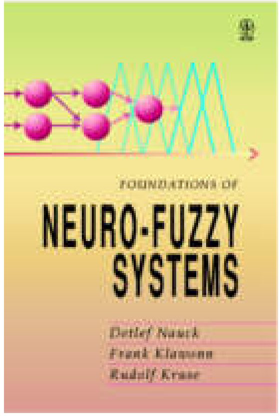
Fuzzy Set Learning
Error Misclassifications
49.00
0.0
0.00
60.00
120.00
180.00
NEFCLASS
Project Classifier View Rules Help
status
classifier
training data
ready
5 rules, trained
Iris_new.dat
Fuzzy Sets
Fuzzy Sets
List of Variables
sm
md
lg
1.0
0.5
0.0
1.0 2.2 3.4 4.6 5.8 7.0
sepal length
sepal width
petal length
petal width
Rule Learning
20 possible rules found. Now determine the optimal co
Selection of consequents is complete.
Writing the rules to the log file, please wait.
Best rule learning: selecting the best 5 rules.
Best rules selected, trimming the rule base.
Rule base with 5 rules created.
WARNING: This rule base covers only 94% of all traini
but this may improve during fuzzy set learning.
Performance on training data (100.0% of all cases):
150 patterns, 15 misclassifications (error = 26.45762
About
NEFCLASS-J
Version 1.0
Programming and GUI Design
Ulrike Nauck
NEFCLASS Model and Learning Alg
Dr. Detlef Nauck
(c) Ulrike Nauck, Braunschweig, 1
NEFCLASS
Edit Rules
Variables
Fuzzy Sets
sepal length
sepal width
petal length
petal width
small
medium
large
Set
To enter a new rule to the rule base, select a variable and a fuzzy set. Then press the Set button, to add a a new term to the antecedent. When the antecedent is complete select a consequent and press Add Rule to create a new rule.
Click on each list to obtain more information.
To save your changes, leave with OK, to discard all modifications, use Cancel.
Antecedent of Rule R0
Consequent (Class)
IF
sepal length is small and
sepal width is medium and
petal length is small and
petal width is small
Then
Iris Setosa
Iris Versicolor
Iris Virginica
Remove Term
Remove All
Modify Rule
Add Rule
R0: if sepal length is small and sepal width is medium and petal length is small and petal width is small then Iris
R9: if sepal length is medium and sepal width is small and petal length is medium and petal width is medium the
R16: if sepal length is medium and sepal width is small and petal length is large and petal width is large then Iris
R17: if sepal length is large and sepal width is small and petal length is large and petal width is large then Iris Vi
R18: if sepal length is large and sepal width is medium and petal length is large and petal width is large then Iris

Resources

Detlef Nauck, Frank Klawonn & Rudolf Kruse:

Foundations of Neuro-Fuzzy Systems

Wiley, Chichester, 1997, ISBN: 0-471-97151-0



Neuro-Fuzzy Software (NEFCLASS, NEFCON, NEFPROX):

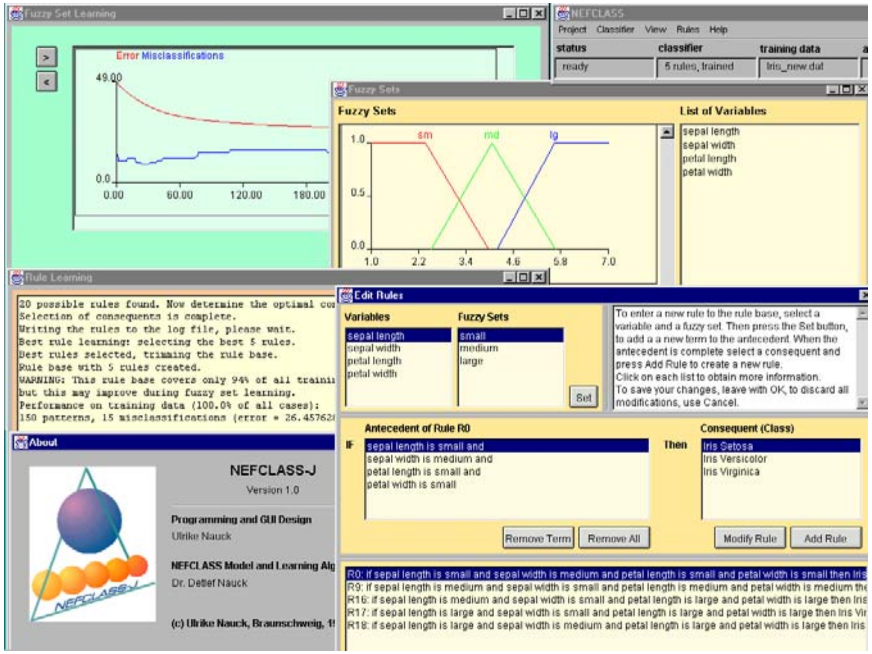
http://www.neuro-fuzzy.de

Beta-Version of NEFCLASS-J:

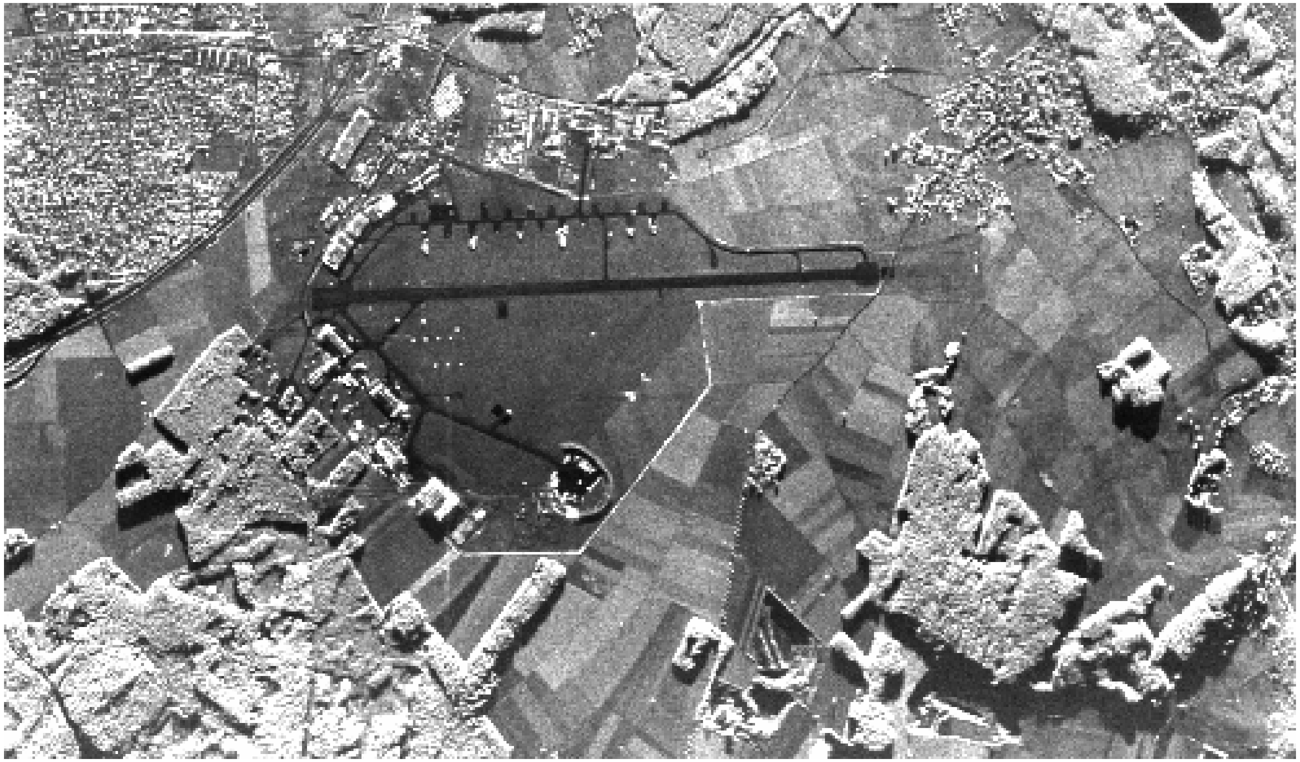
http://www.neuro-fuzzy.de/nefclass/nefclassj

Download NEFCLASS-J

Download the free version of NEFCLASS-J at http://fuzzy.cs.uni-magdeburg.de

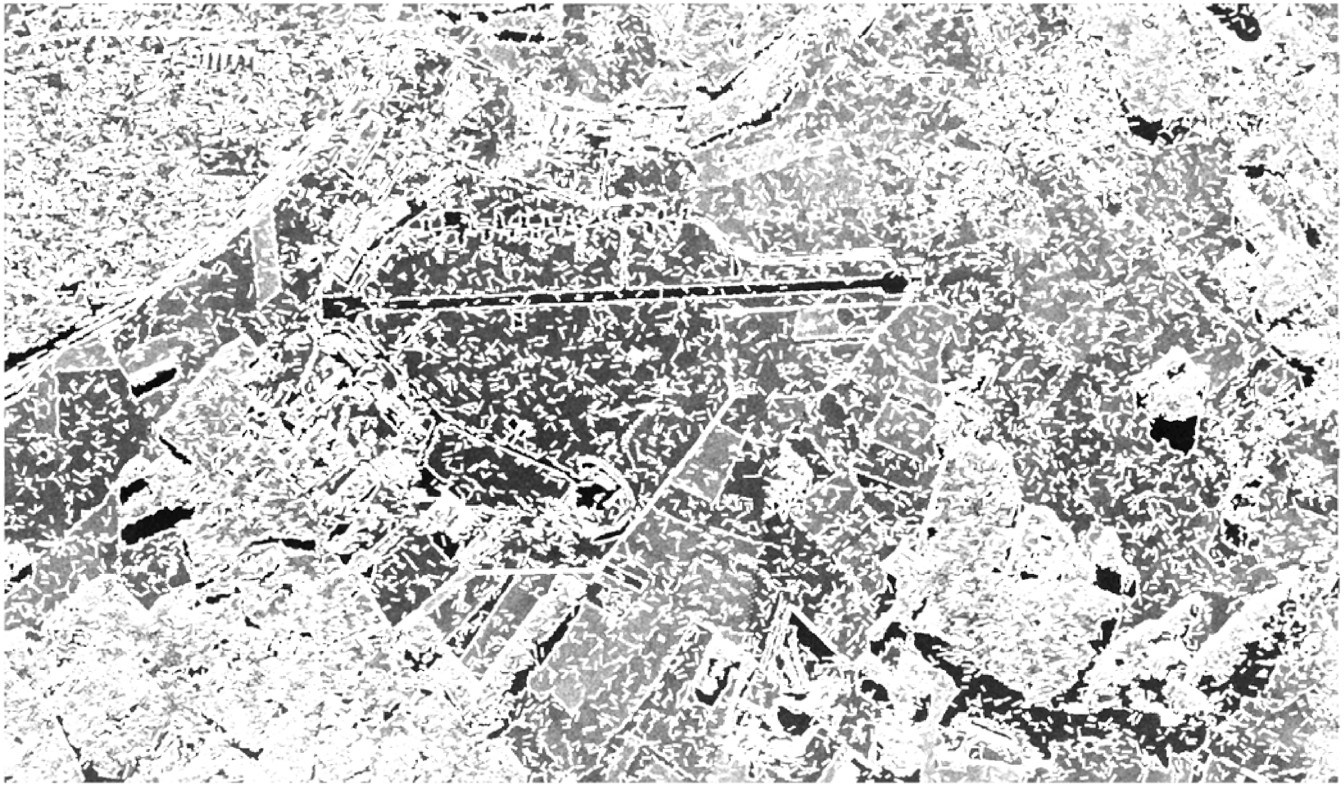


Example: Line Filtering



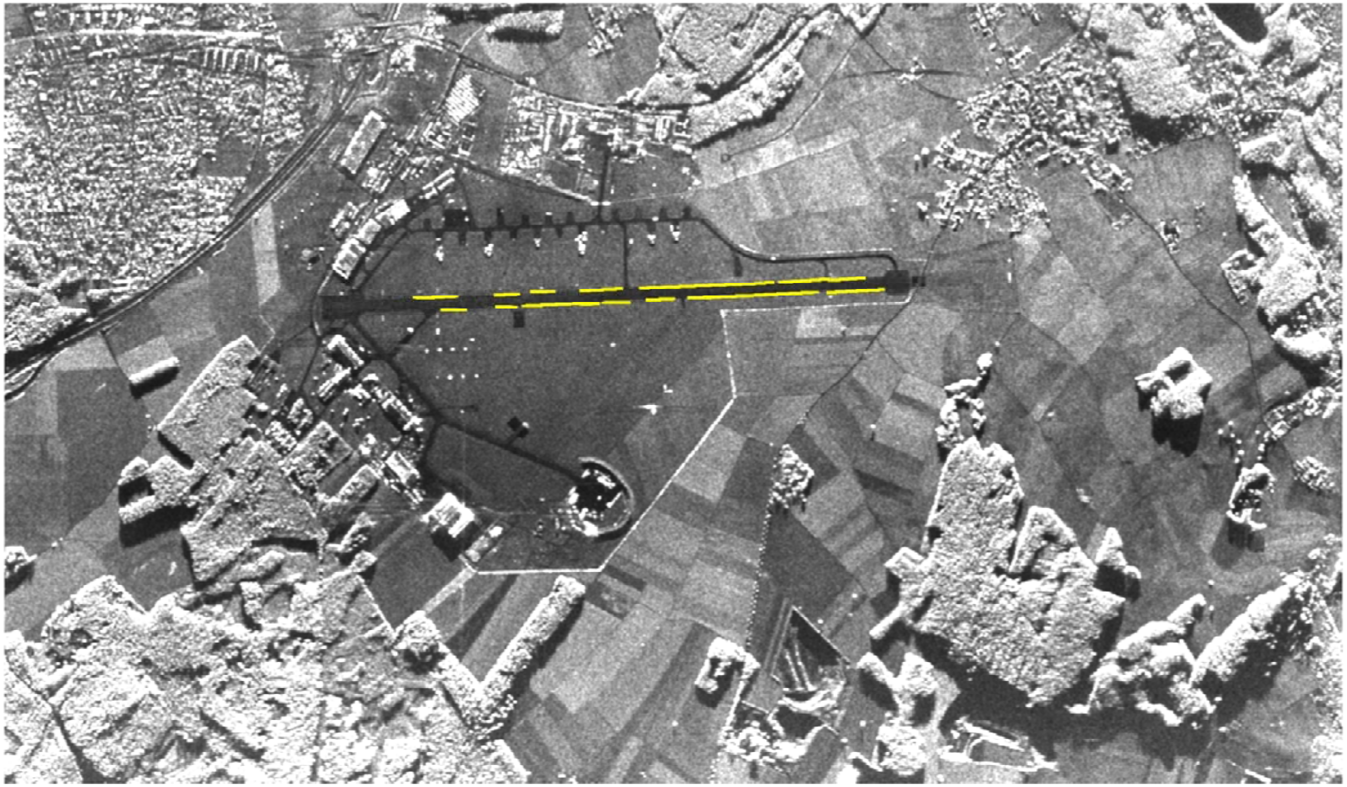
- Extraction of edge segments (Burns' operator)
- Production net:
  edges $\rightarrow$ lines $\rightarrow$ long lines $\rightarrow$ parallel lines $\rightarrow$ runways

Example: Line Filtering

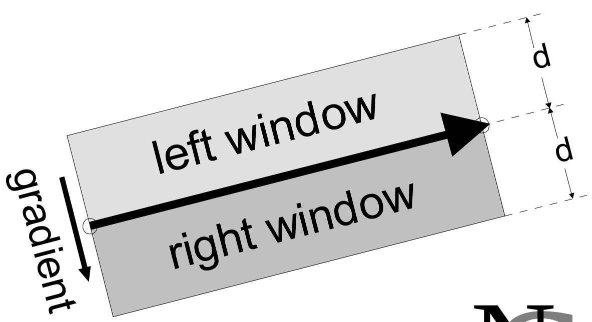


- Problems
  - extremely many lines due to distorted images
  - long execution times of production net

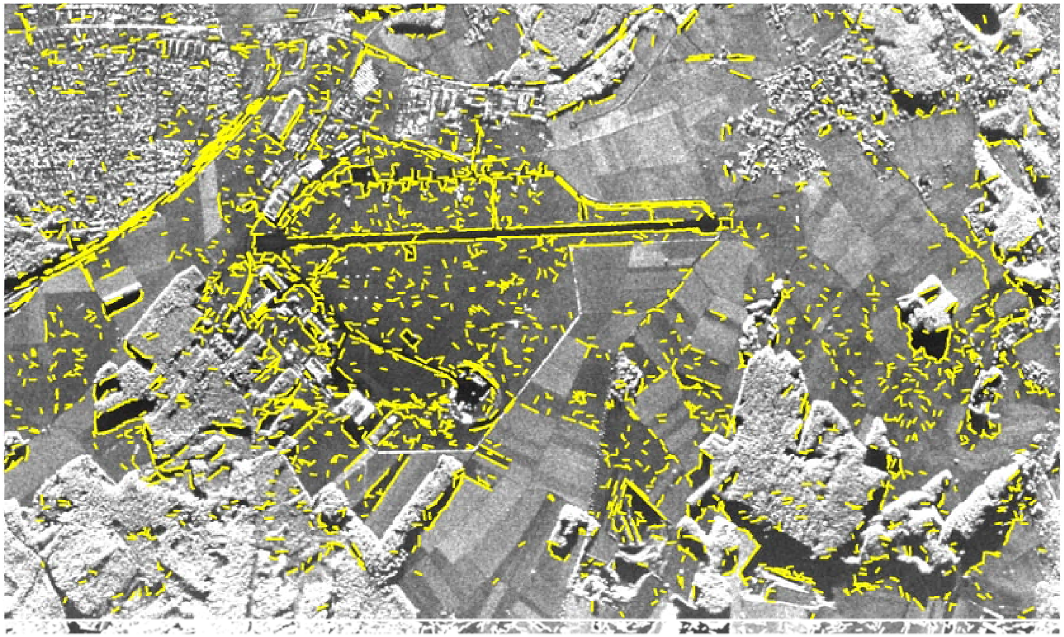
Example: Line Filtering



- Only few lines used for runway assembly
- Approach:
  - Extract textural features of lines
  - Identify and discard superfluous lines



Example: Line Filtering



- Several classifiers:
  - minimum distance, k-nearest neighbor, decision trees, NEFCLASS
- Problems: classes are overlapping and extremely unbalanced
- Result above with modified NEFCLASS:
  - all lines for runway construction found
  - reduction to 8.7% of edge segments

Example: Surface Quality Control

■ Today's Approach

The current surface quality control is done manually → an experienced worker treats the exterior surfaces with a grindstone. The experts classify surface form deviations by means of linguistic descriptions.

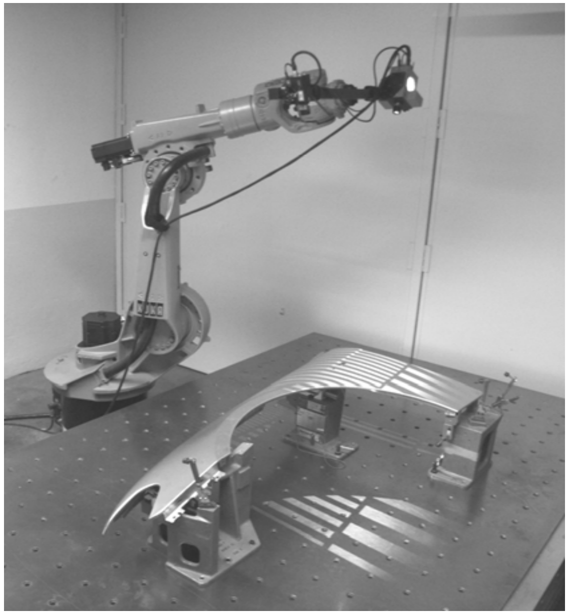
Cumbersome – Subjective - Error Prone Time Consuming

■ The Proposed Approach

Our Approach is based on the digitization of the exterior body panel surface with an optical measuring system.

We characterize the form deviation by mathematical properties that are close to the subjective properties that the experts used in their linguistic description.

Topometric 3-D measuring system

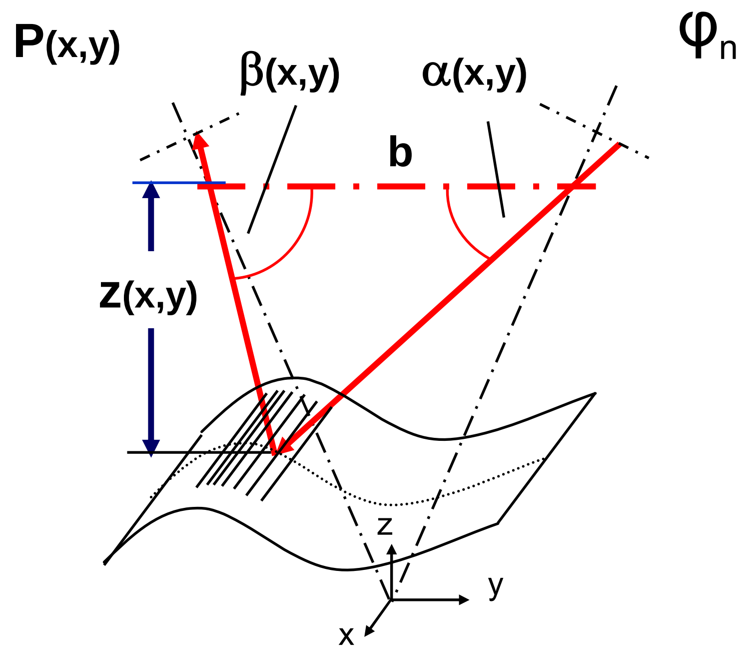


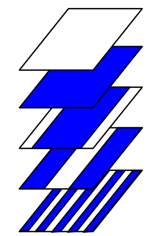
breuckmann

Triangulation and Gratings Projection

0
1
0
0
⋮

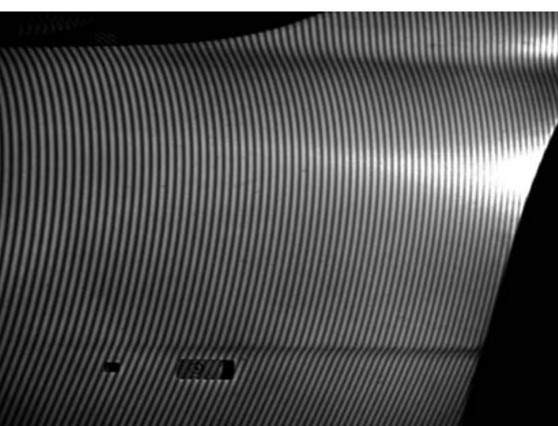
Pixel coding





⋮

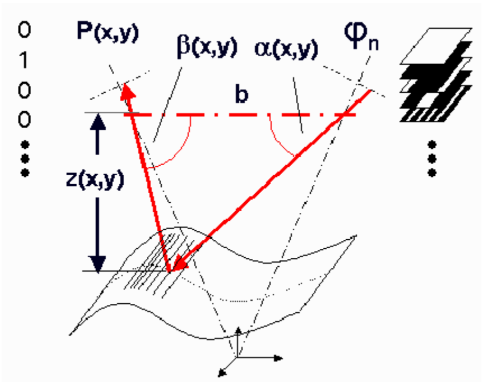
Miniaturized Projection Technique (Grey Code Phase shift)



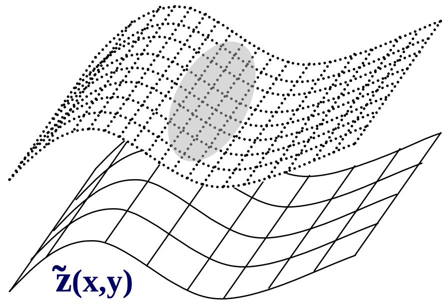
- High Point Density
- Fast Data Collection
- Measurement Accuracy
- Contact less and Non-destructive

NEURO FUZZY

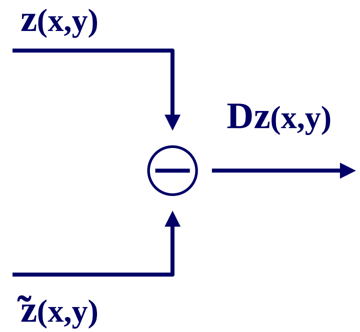
Data Processing



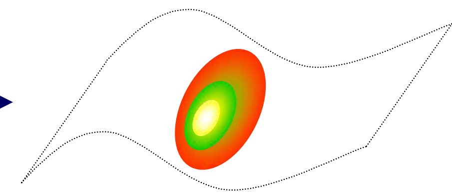
• Approximation by a Polynomial Surface



• Difference

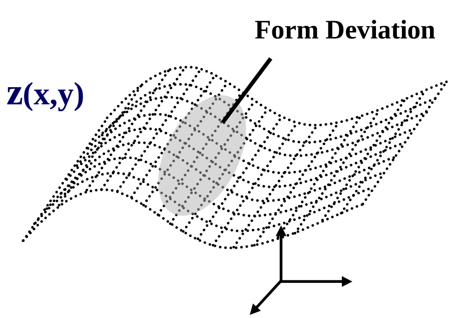


• Colour-Coded Visualization

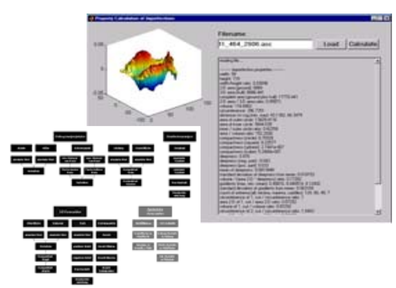


3-D Data Acquisition → Post-Processing → Detection of Form Deviation → Features Analysis

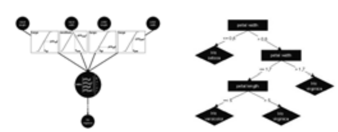
• 3-D-Point Cloud



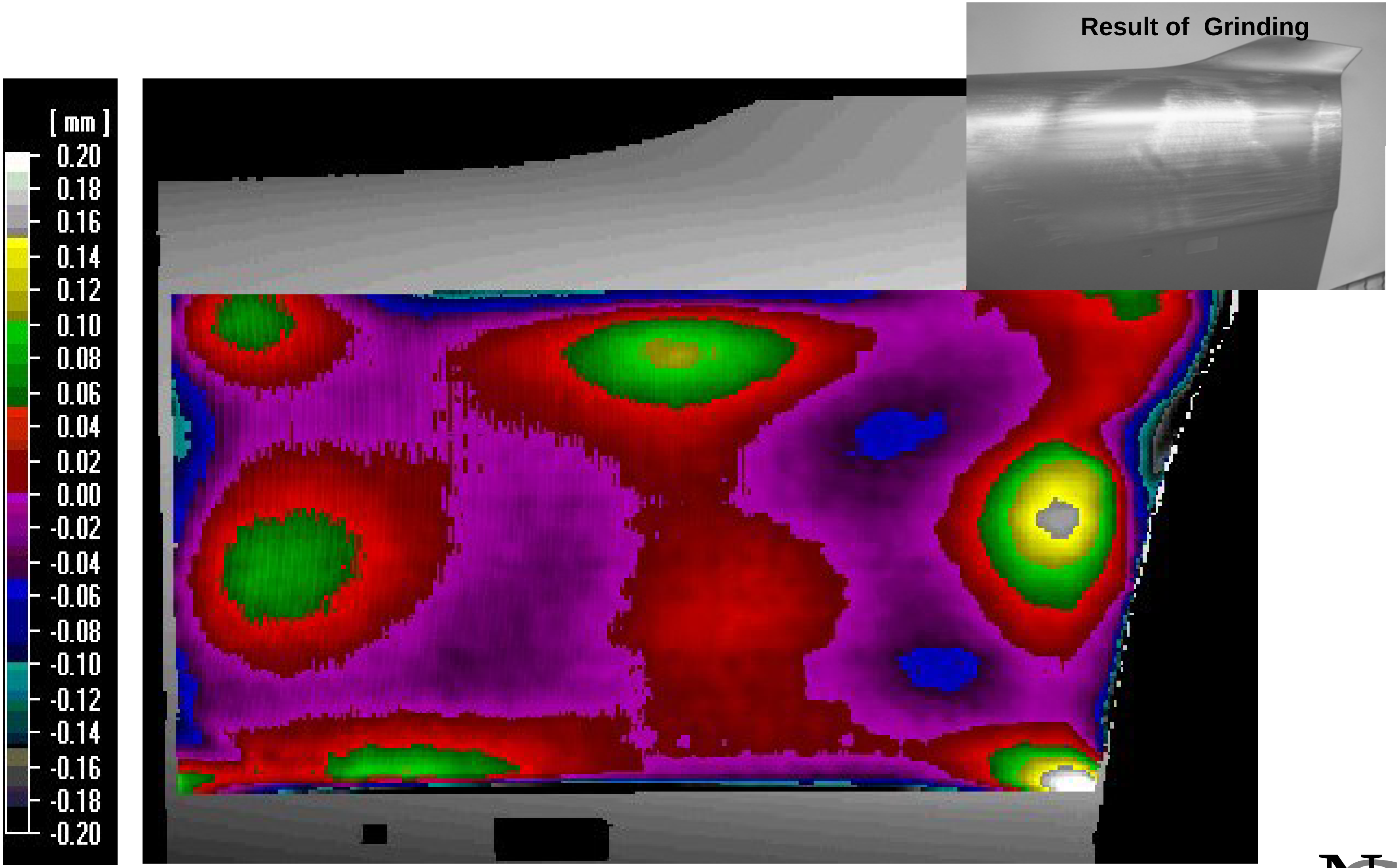
• Feature Calculation



• Classification (Data-Mining)

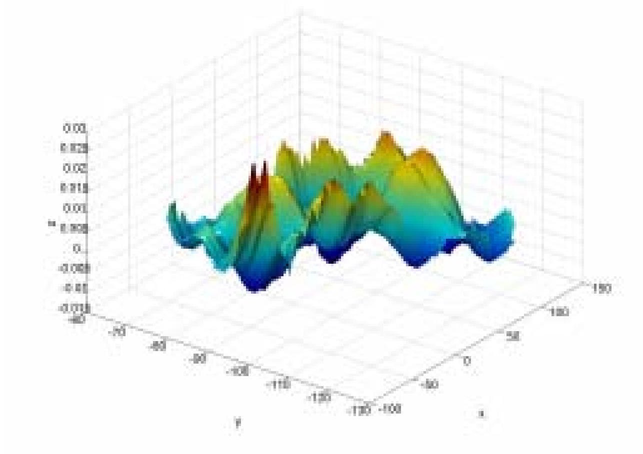


Color Coded Visualization

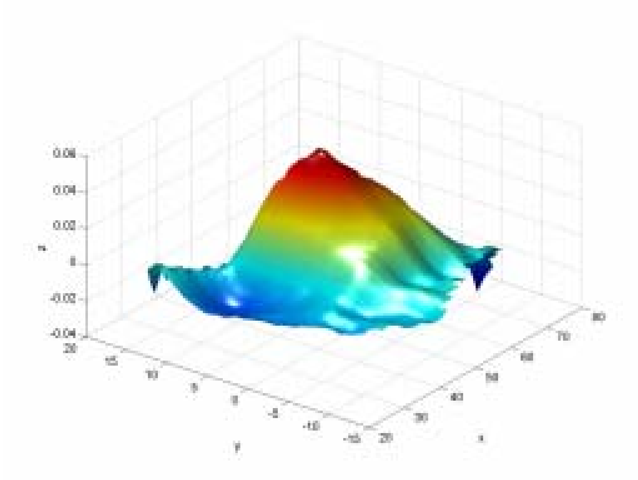


3D Visualization of Local Surface Defects

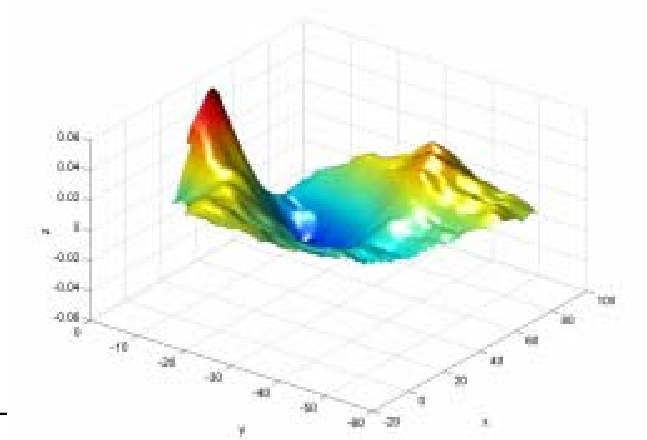
Uneven Surface
(several sink marks in series or adjoined)



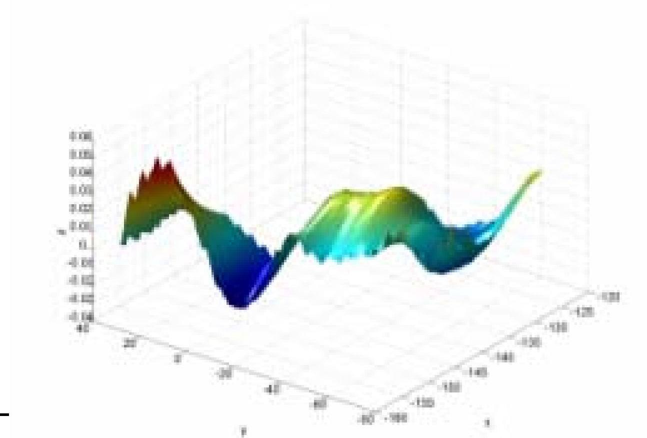
Press Mark
(local smoothing of (micro-)surface)



Sink Mark
(slight flat based depression inward)

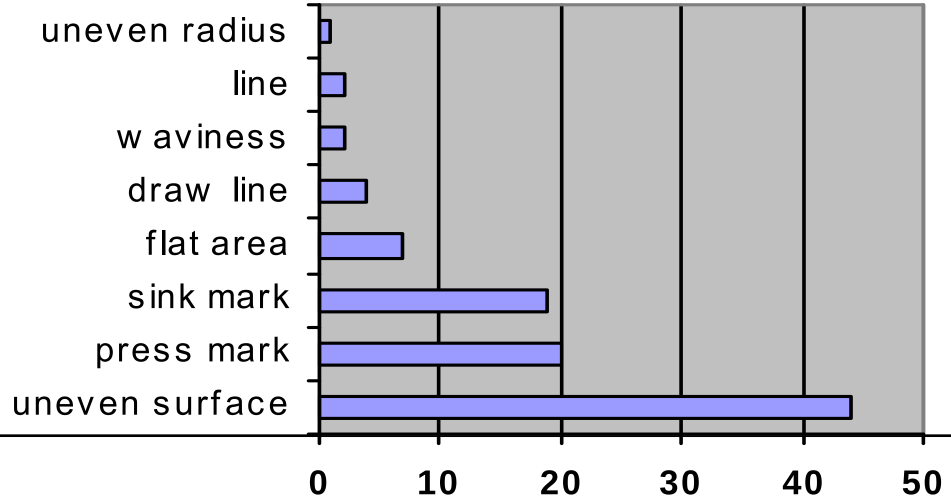


Waviness
(several heavier wrinklings in series)



Data Characteristics

- We analysed 9 master pieces with a total number of 99 defects
- For each defect we calculated 42 features
- The types are rather unbalanced
- We discarded the rare classes
- We discarded some of the extremely correlated features (31 features left)
- We ranked the 31 features by importance
- We use stratified 4-fold cross validation for the experiment.



Application and Results

The Rule Base for NEFCLASS

Rule base
- Rule 1: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 1 AND max_extrema IS fun 1) THEN type IS press_mark
- Rule 2: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND max_extrema IS fun 2) THEN type IS sink_mark
- Rule 3: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
- Rule 4: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
- Rule 5: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND min_extrema IS fun 2) THEN type IS press_mark
- Rule 6: IF (max_distance_to_cog IS fun 3 AND all_extrema IS fun 2 AND max_extrema IS fun 3) THEN type IS uneven_surface
- Rule 7: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 3) THEN type IS uneven_surface

Classification Accuracy

	NBC	DTree	NN	NEFCLASS	DC
Train Set	89.0%	94.7%	90%	81.6%	46.8%
Test Set	75.6%	75.6%	85.5%	79.9%	46.8%

Conclusions

- Neuro-Fuzzy-Systems can be useful for knowledge discovery.
- Interpretability enables plausibility checks and improves acceptance.
- (Neuro-)Fuzzy systems exploit tolerance for sub-optimal solutions.
- Neuro-fuzzy learning algorithms must observe constraints in order not to jeopardise the semantics of the model.
- Not an automatic model creator, the user must **work** with the tool.
- Simple learning techniques support explorative data analysis.