Data Analysis with Neuro-Fuzzy Systems

Rudolf Kruse
Dept. Computer Science
University of Magdeburg

Outline

- Fuzzy Data Analysis
- Neuro Fuzzy Systems
- NEFCLASS

Example: Continuously Adapting Gear Shift Schedule in VW New Beetle

- classification of driver / driving situation by fuzzy logic
- Mamdani controller with 7 rules
- Optimized program
  - 24 Byte RAM
  - 702 Byte ROM on Digimat
- Runtime 80 ms
  - 12 times per second a new sport factor is assigned
- How to generate knowledge automatically from data?
Function Approximation with Fuzzy Rules

- If \( x \) is large then \( y \) is large

Learning from Examples (Observations, Databases)

- **Statistics:** parameter fitting, structure identification, inference method, model selection
- **Machine Learning:** computational learning (PAC learning), inductive learning, learning decision trees, concept learning, ...
- **Neural Networks:** learning from data
- **Cluster Analysis:** unsupervised classification

- Learning Problem is transformed into an optimization problem
- How to use these methods in fuzzy systems?

How to Derive a Fuzzy Controller Automatically from Observed Process Data

- Function approximation
- Perform fuzzy cluster analysis of input-output data (FCM, GK, GG, ...)
- Project clusters
- Obtain fuzzy rules of the kind: "If \( x \) is small then \( y \) is medium"
Data Mining Tasks

- Classification
  *Is this a good customer?*

- Concept Description
  *What makes a good customer? (age, income, ...)*

- Segmentation (Clustering)
  *What kind of customers do I have?*

- Prediction
  *What will be the demand for my product?*

- Dependency Analysis
  *80% of customers who buy diapers buy beer, too*

- Deviation Analysis
  *Why do we sell less insurances in Cleveland?*

Fuzzy Methods in Information Mining: Examples

- here: Exploiting quantitative and qualitative information

- Fuzzy Data Analysis (Projects with Siemens)

- Information Fusion (EC Project)

- Dependency Analysis (Project with Daimler/Chrysler)

Fuzzy Data Analysis

**Strong law of large numbers** (Ralescu, Kruse, Miyakoshi, ...)

Let \( \{x_k \mid k \geq 1\} \) be independent and identically distributed fuzzy random variables such that \( E[\|\text{supp } x\|] < \infty \). Then

\[
\frac{1}{n} \left( x_1 + x_2 + \ldots + x_n, E(\text{co}(x_i)) \right) \to 0
\]

**Books:** Kruse, Meyer: Statistics with Vague Data, Reidel, 1987

Bandemer, Näther: Fuzzy Data Analysis, Kluwer, 1992

Seising, Tanaka and Guo, Wolkenhauer, Viertl, ...

Example: *Prognosis of the Daily Proportional Changes of the DAX at the Frankfurter Stock Exchange (Siemens)*

**Database: time series from 1986 - 1997**

<table>
<thead>
<tr>
<th>DAX</th>
<th>Composite DAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>German 3 month interest rates</td>
<td>Return Germany</td>
</tr>
<tr>
<td>Morgan Stanley index</td>
<td>Dow Jones industrial index</td>
</tr>
<tr>
<td>DM / US-$</td>
<td>US treasury bonds</td>
</tr>
<tr>
<td>Gold price</td>
<td>Nikkei index Japan</td>
</tr>
<tr>
<td>Morgan Stanley index Europe</td>
<td>Price earning ratio</td>
</tr>
</tbody>
</table>
**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 $\mu_1$ AND $x_2$ is $\mu_2$  
  **THEN** $y = \eta$  
  **WITH** weight $k$

---

**Return-on-Investment Curves of the Different Models**

Validation data from March 01, 1994 until April 1997

---

**Neuro-Fuzzy Architecture**

---

**Neuro-Fuzzy Data Analysis**
Neuro-Fuzzy Data Analysis

- Introduction: Neural Networks and Fuzzy-Systems
- Learning Fuzzy Rules
- Learning Fuzzy Sets
- NEFCLASS - An Example
- Interpretable Fuzzy Systems
- Conclusions

Introduction

- 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

Neural Networks

- Usual areas of application:
  - function approximation and classification
- Certain types of NN are universal function approximators
- NN are model free estimators
  (in fact they represent a very general model, but their parameters have no interpretation)
- The parameters (weights) of a NN are estimated by a learning algorithm
- The structure of a NN must be specified

Non-linear model, universal function approximator, connection weights found by "learning".
Classification with Neural Networks

- Multilayer Perceptron: global classification with hyperplanes
- Radial Basis Function Networks: local classification with hyperspheres

NN: Learning by Gradient Descent

- Learning in NN: estimate weights iteratively by gradient descent
- Learning method: Error Backpropagation (BP) or variations like Resilient Propagation (RPROP: adaptive learning rate for each weight, just sign of gradient used, faster than BP)
- Problems: local minima, oscillations, can be time-consuming

Fuzzy Systems

- Usual areas of application: function approximation, classification, control
- FS use fuzzy rules (linguistic rules) to partially describe a function (by vague samples)
- FS use fuzzy sets to represent vague terms like small, medium, large, fast, slow, hot, cold, etc.
- FS are knowledge-based and can be interpreted
- FS can be learned from data (neuro-fuzzy)

Classification with Fuzzy Rules

if $x$ is large and $y$ is large then class 3
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)

Neuro-Fuzzy Data Analysis

- Introduction: Neural Networks and Fuzzy-Systems
- Learning Fuzzy Rules
- Learning Fuzzy Sets
- NEFCLASS - An Example
- Interpretable Fuzzy Systems
- Conclusions

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
**Cluster-Oriented Rule Learning**

**Fuzzy-Cluster Analysis:**

Minimize \( J(X, U, v) = \frac{1}{c} \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{i,k})^m d^2(v_i, x_k) \)

with \( \sum_{i=1}^{c} u_{i,k} = 1 \) and \( \sum_{k=1}^{n} u_{i,k} > 0 \)

\( X \): data set, \( m \): fuzzifier (usually \( 1 < m < 2 \))

\( u_{i,k} \): degree of membership of \( x_k \) to cluster \( i \)

\( v_i \): prototype of cluster \( i \), \( d \): distance measure

**Fuzzy Cluster Analysis**

- **Fuzzy C-Means**: simple, looks for spherical clusters of same size, uses Euclidean distance
- **Gustafson & Kessel**: looks for hyper-ellipsoidal clusters of same size, distance via matrices
- **Gath & Geva**: looks for hyper-ellipsoidal clusters of arbitrary size, distance via matrices
- **Axis-parallel variations** exist that use diagonal matrices (computationally less expensive and less loss of information when rules are created)

---

**Construct Fuzzy Sets by Cluster Projection**

Approximation by a triangular fuzzy set

Convex hull of the discrete degrees of membership

Connection of the discrete degrees of membership

Projecting a cluster means to project the degrees of membership of the data on the single dimensions: Histograms are obtained

---

**Projection of Clusters to Create Rules**

- Problem: There is a loss of information, clusters and rules are not identical.
- The resulting fuzzy sets must be labeled with suitable linguistic terms.
# Projection of Clusters to Create Rules

- Fuzzy sets created by projection may be hard to interpret.
- Unusual distributions of fuzzy sets can be obtained.
- Axis-parallel ellipsoids reduce the loss of information due to projection.

# 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 analysis:
  - 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).
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

Structure-Oriented Rule Learning: Algorithm

- Find all antecedent by using only the numeric attributes.
  Result: list of antecedents.
- If there are symbolic attributes:
  - For each antecedent create so many rule base candidates as there are classes, count the symbolic features, create fuzzy sets for them, and thus complete each rule base candidate.
  - Else: compute best consequent for each antecedent.
  Result: list of rule base candidates.
- Resolve conflicts if necessary and select the final rule base.
  Result: final rule base (now fuzzy sets can be trained).

Learning Fuzzy Rules: Problems in Control

- To build fuzzy controllers, reinforcement learning must be used (note: the correct output is unknown)
- Online rule learning is very difficult in control
  - Decremental learning: create all possible rules and delete bad rules (computationally expensive, not feasible for large domains)
  - Incremental learning: create rules on the fly, guess consequents and improve later
- Example: NEFCON (Nauck/Kruse, 1993) online, structure-oriented, incremental rule learning

Missing Values in Rule Learning

Two input variables, one variable is missing.

Generate 3 antecedents:
- $x$ is large and $y$ is small
- $x$ is large and $y$ is medium
- $x$ is large and $y$ is large

Input pattern: $(x_0, ?)$
Fuzzy Rule Learning: Other Approaches

- Fuzzy Decision Trees
  - Advantage: aims at creating small rule bases by selecting „good“ attributes first when building the tree
  - Drawback: may miss important variables

- Genetic Algorithms
  - Advantage: can find „optimal“ rule base
  - Drawback: computationally expensive

- Combinatorical Approaches
  - Try out combinations of variables, only feasible for small domains

Neuro-Fuzzy Data Analysis

- Introduction: Neural Networks and Fuzzy Systems
- Learning Fuzzy Rules
- Learning Fuzzy Sets
- NEFCLASS - An Example
- Interpretable Fuzzy Systems
- Conclusions

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: Gradient Descent

Assume a Sugeno-type fuzzy system with triangular fuzzy sets and rules like „\( R_k \): if \( x \) is \( \mu_k \) then \( y = c_k \)“

\[
\mu(x) = \begin{cases} 
1 - \frac{2|x - a|}{b} & \text{if } a - \frac{b}{2} \leq x \leq a + \frac{b}{2} \\ 0 & \text{otherwise} 
\end{cases}
\]

\[
y = \frac{\sum_{k=1}^{r} \tau_k c_k}{\sum_{k=1}^{r} \tau_k}, \quad \tau_k = \prod_{i=1}^{n} \mu_k^{(i)}(x_i)
\]
Parameter updates for one pattern, do not compute if $\mu(x) = 0$ or $\mu(x) = 1$ (non-differentiable points of triangular fuzzy sets)

$$
\Delta a = \left( \sum_{k: \mu \in \text{An}(R_k)} \tau_k (c - y) \right) \frac{\sigma \tau}{\sum_k \tau_k} (\hat{y} - y) \frac{2 \text{sgn}(x - a)}{b \mu(x)},
$$

$$
\Delta b = \left( \sum_{k: \mu \in \text{An}(R_k)} \tau_k (c - y) \right) \frac{\sigma \tau}{\sum_k \tau_k} (\hat{y} - y) \frac{1 - \mu(x)}{b \mu(x)},
$$

$$
\Delta c = \frac{\sigma \tau}{\sum_k \tau_k} (\hat{y} - y), \text{ where } \hat{y} \text{ is the target output}
$$

Heuristic algorithm: shift fuzzy sets to (away from) $x$ and enlarge (reduce) their support to increase (decrease) the degree of membership of $x$

Example: Fuzzy Classifier
NEFCLASS (Nauck/Kruse, 1995)

$$
\mu(x) = \begin{cases} 
\frac{x - a}{b - a} & \text{if } x \in [a,b) \\
\frac{c - x}{c - b} & \text{if } x \in [b,c] \\
0 & \text{otherwise}
\end{cases}
$$

Rules of the form:
if $x_1$ is $\mu_1$ and ... and $x_n$ is $\mu_n$
then pattern $(x_1,...,x_n)$ belongs to class $c$
Learning Fuzzy Sets: Numerical Attributes

\[ \Delta b = f \cdot e \cdot (c - a) \cdot \text{sgn}(x - b) \]
\[ \Delta a = -f \cdot e \cdot (b - a) + \Delta b \]
\[ \Delta c = f \cdot e \cdot (c - b) + \Delta b \]

with \( f = \begin{cases} \sigma \cdot \mu(x) & \text{if } e < 0 \\ \sigma \cdot (1 - \mu(x)) & \text{otherwise} \end{cases} \)

\[ e = e_c \cdot (\tau - (1 - \tau) + \varepsilon) \]
\[ e_c = t - o \]

Learning Fuzzy Sets: Symbolic Attributes

Fuzzy sets for symbolic attributes are represented by vectors \( \mathbf{m} = (m_1, ..., m_q) \), \( m[p_j] \) is the entry for the current input value \( p_j \)

\[ \Delta \mathbf{m}[p_j] = \sigma \cdot m \cdot e \]

\[ m = \begin{cases} \mathbf{m}[p_j] & \text{if } e < 0 \\ 1 - \mathbf{m}[p_j] & \text{if } e > 0 \end{cases} \]

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.

Learning Fuzzy Sets: Problems in Control

- Reinforcement learning must be used to compute an error value (note: the correct output is unknown)

- After an error was computed, any fuzzy set learning procedures can be used

- Example: GARIC (Berenji/Kedhkar 1992) online approximation to gradient-descent

- Example: NEFCON (Nauck/Kruse 1993) online heuristic fuzzy set learning using a rule-based fuzzy error measure
Different Neuro-Fuzzy Approaches

- **ANFIS** (Jang, 1993)
  no rule learning, gradient descent fuzzy set learning, function approximator

- **GARIC** (Berenji/Kedhkar, 1992)
  no rule learning, gradient descent fuzzy set learning, controller

- **NEFCON** (Nauck/Kruse, 1993)
  structure-oriented rule learning, heuristic fuzzy set learning, controller

- **FuNe** (Halgamuge/Glesner, 1994)
  combinational rule learning, gradient descent fuzzy set learning, classifier

- **Fuzzy RuleNet** (Tschichold-Gürman, 1995)
  hyperbox-oriented rule learning, no fuzzy set learning, classifier

- **NEFCLASS** (Nauck/Kruse, 1995)
  structure-oriented rule learning, heuristic fuzzy set learning, classifier

- **Learning Fuzzy Graphs** (Berthold/Huber, 1997)
  hyperbox-oriented rule learning, no fuzzy set learning, function approximator

- **NEFPROX** (Nauck/Kruse, 1997)
  structure-oriented rule learning, heuristic fuzzy set learning, function approximator

Neuro-Fuzzy Data Analysis

- **Introduction: Neural Networks and Fuzzy-Systems**
- **Learning Fuzzy Rules**
- **Learning Fuzzy Sets**
- **NEFCLASS - An Example**
- **Interpretable Fuzzy Systems**
- **Conclusions**

Example: NEFCLASS-J

- 699 cases (16 cases with missing values)
- 2 classes: benign (458 cases), malign (241 cases)
- 9 attributes from \{1, ..., 10\}
  - actually an ordinal scale, but usually treated as metric values
- experiment: we treated \(x_3\) and \(x_6\) as symbolic attributes
- \(x_3\) and \(x_6\) are „important“ attributes
**NEFCLASS-J: Learning Result (Rules)**

\[ R_1: \text{if uniformity of cell size is small and bare nuclei is fuzzy0 then benign} \]

\[ R_2: \text{if uniformity of cell size is large then malign} \]

**NEFCLASS-J: Learning Result (Fuzzy Sets)**

![Graph showing fuzzy sets for uniformity of cell size and bare nuclei.]

**NEFCLASS-J: Learning Result (Performance)**

<table>
<thead>
<tr>
<th>Predicted Class</th>
<th>malign</th>
<th>benign</th>
<th>not classified</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>malign</td>
<td>228 (32.62%)</td>
<td>13 (1.86%)</td>
<td>0 (0%)</td>
<td>241 (34.48%)</td>
</tr>
<tr>
<td>benign</td>
<td>15 (2.15%)</td>
<td>443 (63.38%)</td>
<td>0 (0%)</td>
<td>458 (65.52%)</td>
</tr>
<tr>
<td>sum</td>
<td>243 (34.76%)</td>
<td>456 (65.24%)</td>
<td>0 (0%)</td>
<td>699 (100.00%)</td>
</tr>
</tbody>
</table>

- **NEFCLASS-J:** 96.0%
- **Discriminant:** 96.1%
- **C 4.5:** 95.1%
- **NEFCLASS-X:** 95.1%
- **Neural Net:** 94.8%
- **C 4.5 Rules:** 95.4%

**Neuro-Fuzzy Data Analysis**

- Introduction: Neural Networks and Fuzzy-Systems
- Learning Fuzzy Rules
- Learning Fuzzy Sets
- NEFCLASS - An Example
- Interpretable Fuzzy Systems
- Conclusions
When is a Fuzzy System Interpretable?

- If it is a Mamdani-type fuzzy system with few rules that use
  - few variables which are partitioned by
  - few meaningful fuzzy sets.
  - no rule weights . . .

- Control: Sugeno-type fuzzy systems with meaningful local models

- Neuro-fuzzy approaches
  - Often: train a fuzzy system to improve accuracy
  - Problem: danger to lose interpretability
  - Solution: constrain the learning procedures

- If interpretation is important, then a loss of accuracy must be tolerated.

Reasons to Use Fuzzy Systems

- Vague knowledge can be included into the solution, i.e. we know something about our data or a possible solution.

- The solution is interpretable in form of linguistic rules, i.e. we want to learn something about our data/problem.

- From an applicational point of view the solution should be easy to implement, to use and to understand.

- Interpretation is more important than performance

Neuro-Fuzzy Learning Should

- be computationally efficient,
- be easy to influence,
- support an exploratory approach,
- be easy to understand,
- be constrained.

Conclusions

- Neuro-Fuzzy Methods are tools to support the development of fuzzy systems
- Neuro-Fuzzy Methods are not automatic fuzzy system creators, the user must be involved
- The learning algorithms are heuristics, a successful learning outcome cannot be guaranteed
- If applied suitably neuro-fuzzy methods can be very helpful in designing fuzzy systems
Resources

Detlef Nauck, Frank Klawonn & Rudolf Kruse:

**Foundations of Neuro-Fuzzy Systems**

Neuro-Fuzzy Software (NEFCLASS, NEFCON, NEFPROX):
http://www.neuro-fuzzy.de

Beta-Version of NEFCLASS-J:
http://www.neuro-fuzzy.de/nefclass/nefclassj

http://fuzzy.cs.uni-magdeburg.de