Unit 9

Complementary Topics

Overview

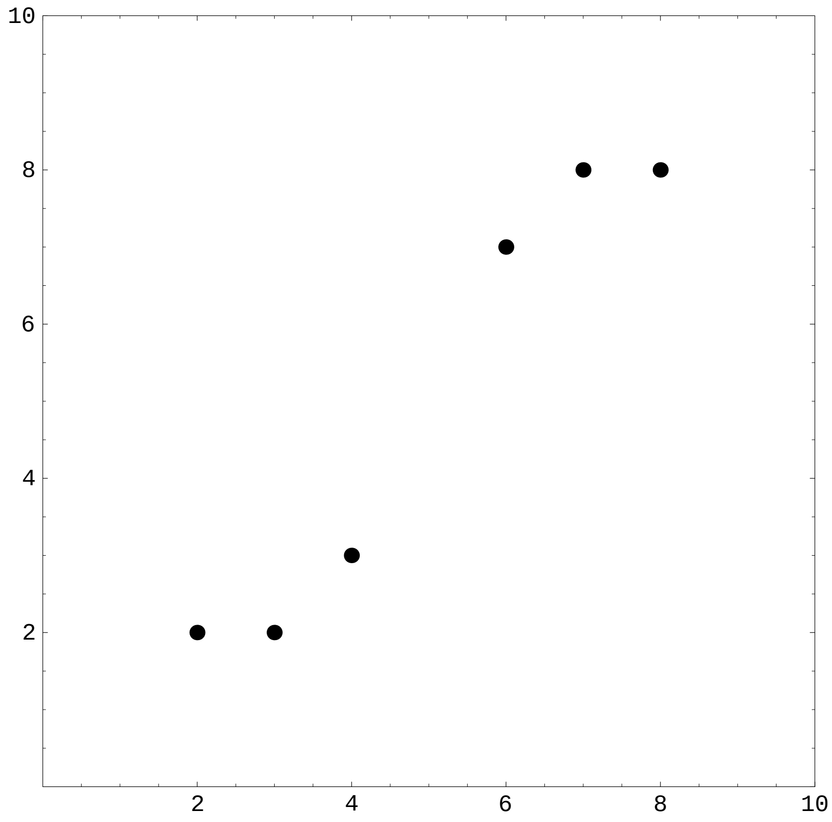
- Overview of fuzzy clustering
  - Important representative: fuzzy c-means
- Overview of learning and tuning methods
  - Inductive learning of fuzzy rules
  - Fuzzy decision trees
  - Numerical optimization of fuzzy systems
  - Overview of other models and methods

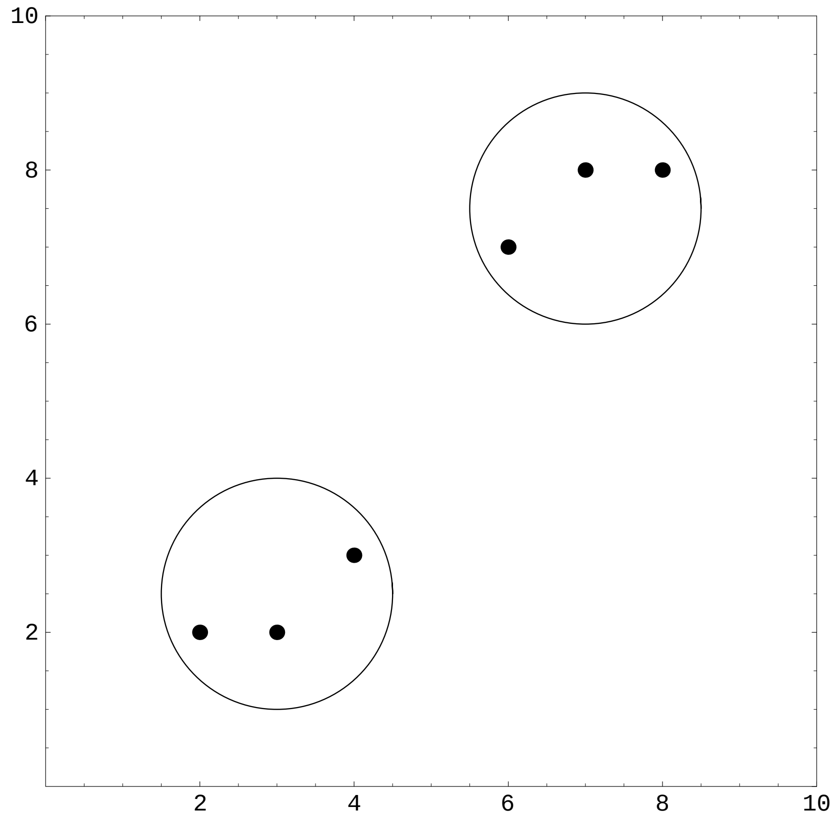
Clustering: Motivation

- Labeled data (data whose classification is known) are sometimes not available
- Sometimes not even the classes and their characteristic features are known
- Data reduction
- For such purposes, it is necessary to identify significant groups of data points, so-called *clusters*

Clusters are data groups in which the points have small distances/high similarity, where the different clusters have a large distance/low similarity.

A Simple Example





Basic Requirements

1. $C_1 \cup \cdots \cup C_K = X$
2. $C_i \neq \emptyset$ for $i = 1, \ldots, K$
3. $C_i \cap C_j = \emptyset$ for $i = 1, \ldots, K$ with $i \neq j$

Prototype-Based Clustering

- Instead of a complete set description, every cluster C_i is represented by a typical value $\mathbf{v}_i$, which can usually be interpreted as the center of the cluster
- The distance to the nearest prototype determines to which cluster a data point belongs, i.e. $\mathbf{x}_k \in C_i$ if

$$\|\mathbf{x}_k - \mathbf{v}_i\| = \min_{j=1}^{K} \|\mathbf{x}_k - \mathbf{v}_j\|$$

The c-Means (CM) Model

Objective function to be minimized:

$$J_{CM}(U,V) = \sum_{i=1}^{K} \sum_{\mathbf{x}_k \in C_i} \|\mathbf{x}_k - \mathbf{v}_i\|^2 = \sum_{i=1}^{K} \sum_{k=1}^{M} u_{ik} \|\mathbf{x}_k - \mathbf{v}_i\|^2$$

Computation of prototypes:

$$\mathbf{v}_i = \frac{1}{|C_i|} \cdot \sum_{\mathbf{x}_k \in C_i} \mathbf{x}_k = \frac{\sum_{k=1}^{n} u_{ik} \mathbf{x}_k}{\sum_{k=1}^{M} u_{ik}} \qquad (1)$$

The c-Means Algorithm

1. Given: data set $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\} \subseteq \mathbb{R}^n$, norm $\|.\|$ on $\mathbb{R}^n$, predefined number of clusters K, maximum number of iterations $t_{\max}$, distance measure $\|.\|_v$, threshold ε
2. Initialization: $V^{(0)} \subseteq \mathbb{R}^n$
3. For $t = 1, \ldots, t_{\max}$ do:
   - Determine $U^{(t)}(V^{(t)})$ (nearest prototype)
   - Determine $V^{(t)}(U^{(t)})$ (by Eq. (1))
   - if $\|V^{(t)} - V^{(t-1)}\|_v \leq \varepsilon$, stop
4. Output: partition matrix U, set of prototypes V

The Fuzzy c-Means (FCM) Model

Objective function to be minimized:

$$J_{FCM}(U,V) = \sum_{i=1}^{K} \sum_{k=1}^{M} u_{ik}^{m} \|\mathbf{x}_k - \mathbf{v}_i\|^2$$

Computation of prototypes:

$$\mathbf{v}_i = \frac{\sum_{k=1}^{M} u_{ik}^{m} \mathbf{x}_k}{\sum_{k=1}^{M} u_{ik}^{m}} \qquad (2)$$

Update of partition matrix:

$$u_{ik} = 1 \Big/ \sum^{K} \left(\frac{\|\mathbf{x}_k - \mathbf{v}_i\|}{\| \quad \|} \right)^{\frac{2}{m-1}} \qquad (3)$$

The Fuzzy c-Means Algorithm

1. Given: data set $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\} \subseteq \mathbb{R}^n$, norm $||.||$ on $\mathbb{R}^n$, predefined number of clusters K, sharpness exponent m, maximum number of iterations t_{max}, distance measure $||.||_v$, threshold ε
2. Initialization: $V^{(0)} \subseteq \mathbb{R}^n$
3. For $t = 1, \ldots, t_{\text{max}}$ do:
   - Determine $U^{(t)}(V^{(t)})$ (by Eq. (3))
   - Determine $V^{(t)}(U^{(t)})$ (by Eq. (2))
   - if $||V^{(t)} - V^{(t-1)}||_v \leq \varepsilon$, stop
4. Output: partition matrix U, set of prototypes V

Adaptive Variants

Different distributions and sizes of clusters usually lead to suboptimal results with CM/FCM. In order to adapt to different structures in data, problem-specific distance measures can be used.

The Gustafson-Kessel (GK) Model:

$$J_{GK}(U,V) = \sum_{i=1}^{K} \sum_{k=1}^{M} u_{ik}^{m} \big((\mathbf{x}_k - \mathbf{v}_i)^T A_i (\mathbf{x}_k - \mathbf{v}_i) \big)$$

$$A_i = \sqrt[p]{\rho_i \text{det}(S_i)} S_i^{-1}$$

$$S_i = \sum_{k=1}^{M} u_{ik}^{m} (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T$$

Learning and Tuning: Motivation

- In all our studies so far, we considered the fuzzy sets and rules as given.
  But where do they actually come from?
- Often they are provided by experts that have sufficient knowledge of the given control/classification task.
  Even in such a case, how can we optimize the parameters?
- In many cases, however, there is nothing known.
  What do we do then?

According to these motivations, numerous methods for constructing/optimizing fuzzy systems from example data have been developed.

Learning and Tuning: The Basic Setup

- Suppose we have a problem in which an output y should be assigned to an n-dimensional input vector $(x_1, \ldots, x_n)$, where the output is either a class label (classification) or a numerical value (interpolation/control/prediction)
- Suppose we have M data samples $(x_1^j, \ldots, x_n^j; y^j)$ ($j = 1, \ldots, M$)
- If we denote the output of an appropriate fuzzy system with $F(x_1, \ldots, x_n)$, the goal is to find parameters (fuzzy sets, rules) such that the output for each input sample $(x_1^j, \ldots, x_n^j)$ is as close to y^j as possible. Simple variant of an error measure:

$$\sum_{j=1}^{M} \left(F(x_1^j, \ldots, x_n^j) - y^j\right)^2$$

Example: The Wine Data Set

Inputs: Chemical Parameters: Alcohol, Malic Acid, Ash, Alkalinity of Ash, Magnesium, Total Phenols, Flavonoids, Non-Flavonoid Phenols, Proanthocyanin, OD280/OD315 of Diluted Wines, Proline Optical Properties: Color Intensity, Hue

Output parameter: vineyard

Goal: identify relationships between properties of wines and the vineyard they originate from

Overview of FS-FOIL

- FS-FOIL is based on a well-known machine learning method (**F**irst-**O**rder **I**nductive **L**earner)
- It tries to describe the data samples fulfilling a certain goal predicate by means of assertions about the input variable; this is done by sequential coverage; the choice of predicates for this stepwise refinement is based on an information gain measure
- Fuzzy sets are chosen a priori according to the distribution of sample data (by means of clustering)

The Language

$$t(\text{“}x \text{ IS } A\text{”}|x_0) = \mu_A(x_0)$$

$$t(\text{“}x \text{ IS NOT } A\text{”}|x_0) = 1 - \mu_A(x_0)$$

$$t(\text{“}x \text{ IS AT LEAST } A\text{”}|x_0) = \sup\{\mu_A(y) \mid y \leq x_0\}$$

$$t(\text{“}x \text{ IS AT MOST } A\text{”}|x_0) = \sup\{\mu_A(y) \mid y \geq x_0\}$$

Example: FS-FOIL Rules for the Wine Data Set

	IF	THEN
Rule 1:	(Flavonoids IstAtLeast High AND Proline IsAtLeast High)	Class Is 1
Rule 2:	(Alcohol IsAtMost Low) OR (Flavonoids Is High AND Alcohol Is High AND Proline IsAtMost Low)	Class Is 2
Rule 3:	(OD280OD315OfDilutedWines IsAtMost Low)	Class Is 3

Descriptive Data Analysis with FS-FOIL

Descriptive Data Analysis with FS-FOIL

Descriptive Data Analysis with FS-FOIL

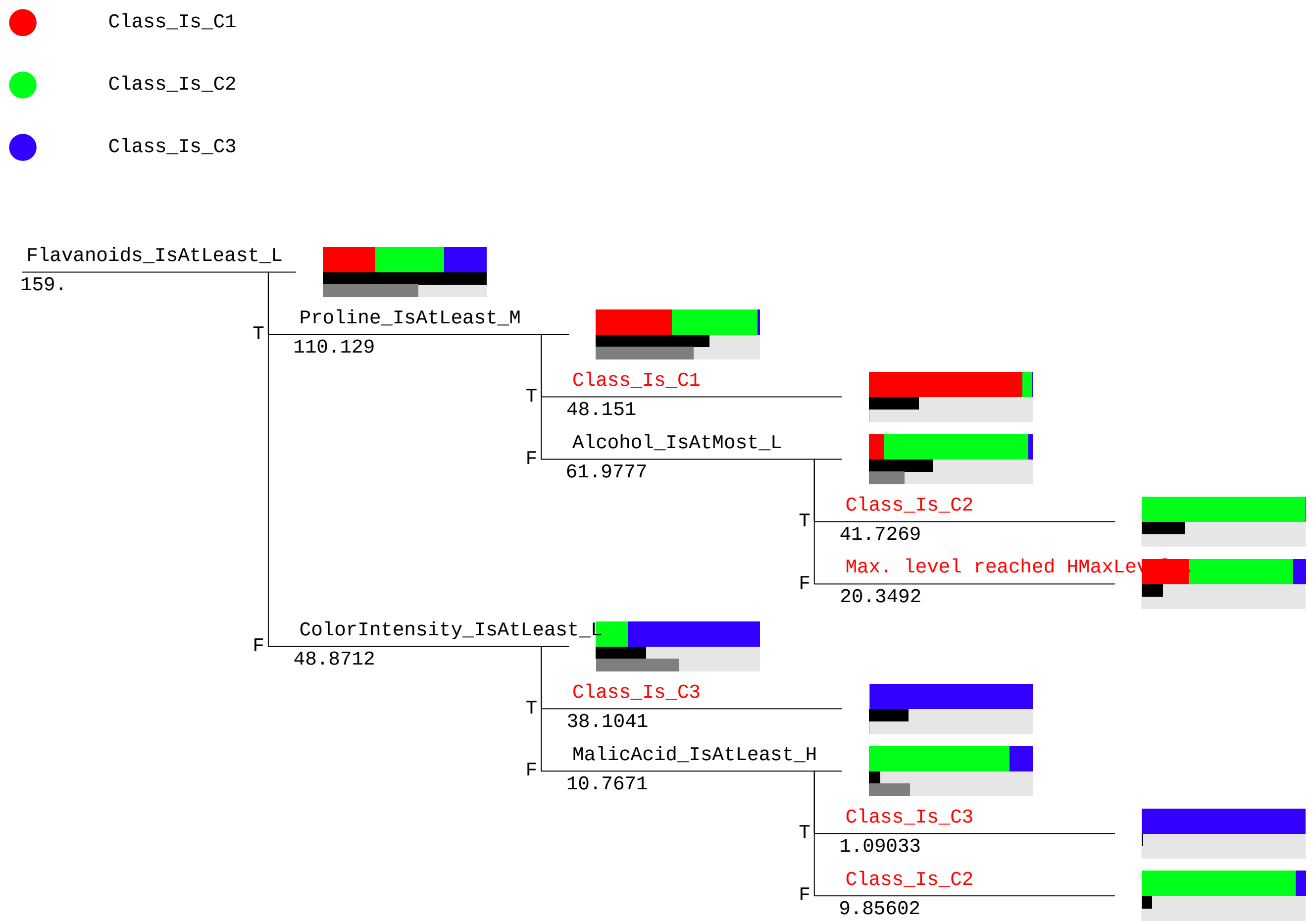
Descriptive Data Analysis with FS-FOIL

	Description
Cluster 1:	(Blue Is High) OR (Red IsAtMost Low AND Blue IsAtLeast VeryHigh)
Cluster 2:	Lightness IsAtMost Dark
Cluster 3:	Lightness IsAtLeast Light
Cluster 4:	(Hue Is Orange) OR (Hue Is Red) OR (Hue Is Yellow) OR (Hue Is Green AND Lightness Is Normal)

Overview of FS-ID3

- FS-ID3 is based on a well-known decision tree induction method (ID3)
- It tries to split the data samples hierarchically by means of a decision tree such that the data sets in the leaf nodes are as homogeneous as possible
- In order to do the splits, FS-ID3 uses an information gain measure
- Fuzzy sets are chosen a priori according to the distribution of sample data (by means of clustering)

Example: FS-ID3 Decision Tree for the Wine Data Set



Numerical Optimization of Sugeno/TSK Fuzzy Systems

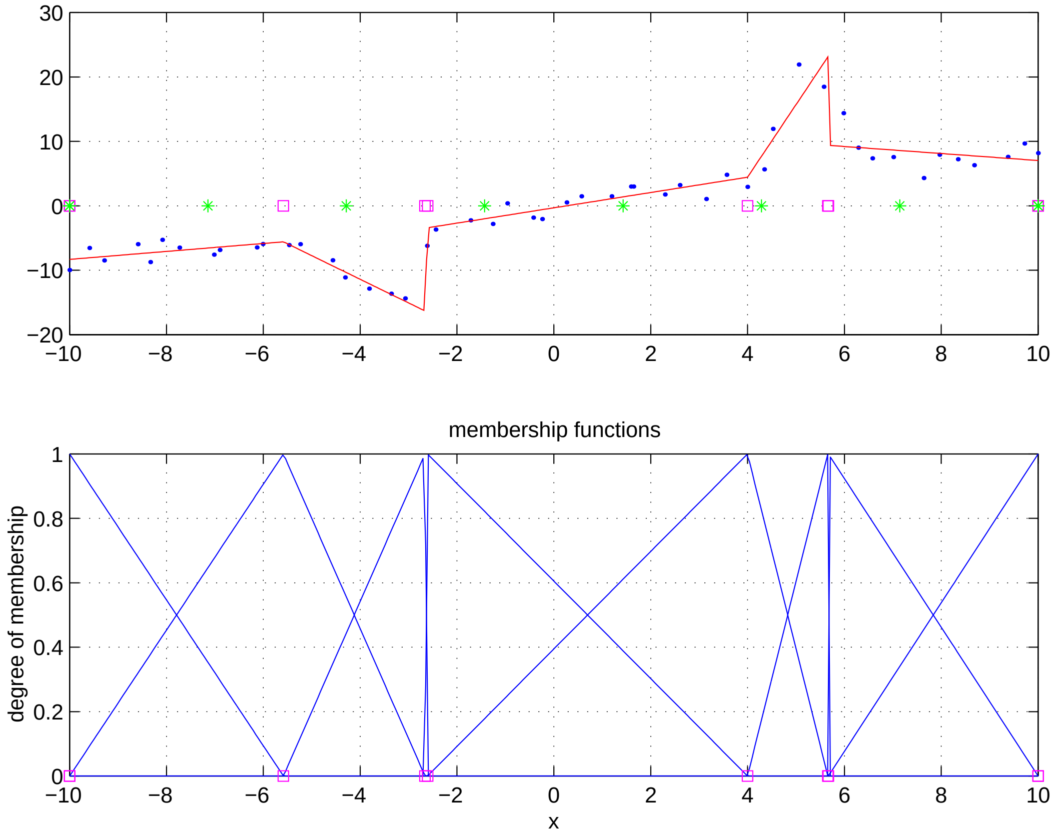
- The optimization problem is linear w.r.t. the coefficients and highly non-linear w.r.t. the parameters describing the fuzzy sets
- Taking interpretability into account results in a relatively large number of constraints
- The problem is *ill-posed*, i.e. the solution of the data fitting problem depends on the data samples in a discontinuous way; therefore, the solution is *unstable* with respect to perturbations (in particular, noise) in the data

RENO

- . . . stands for **Re**gularized **N**umerical **O**ptimization of Fuzzy Systems
- RENO is a highly efficient numerical method for optimizing Sugeno/TSK fuzzy systems with the use of regularization
- RENO can also be applied to the a posteriori tuning of fuzzy systems constructed with FS-ID3/FS-FOIL

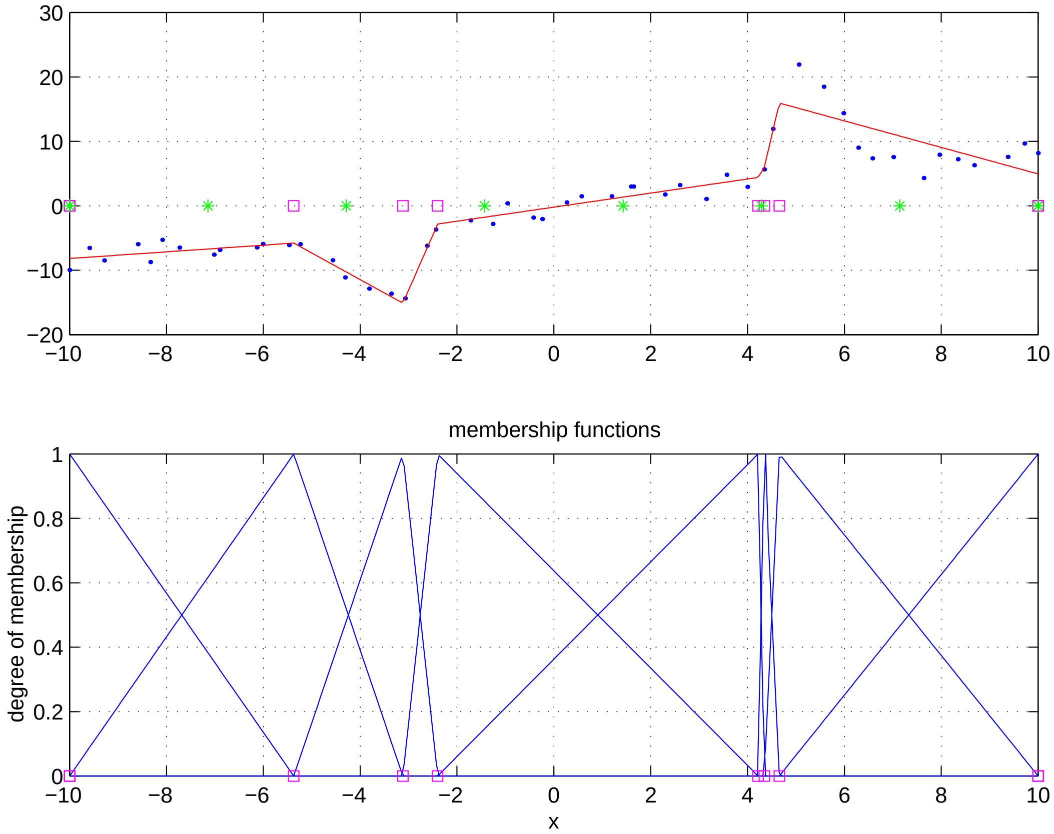
Spectral Data Function with Noisy Measurements

Raw
approximation
without
regularization



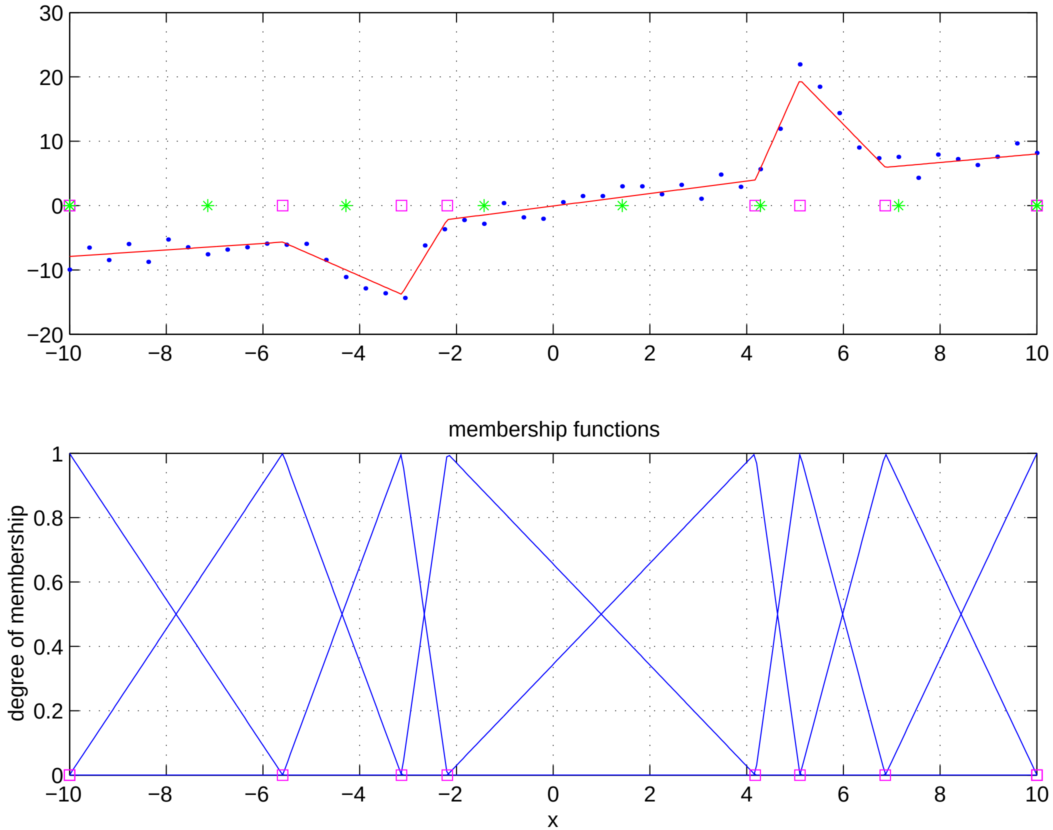
Spectral Data Function with Noisy Measurements

Smoothing



Spectral Data Function with Noisy Measurements

Tikhonov regularization



Sugeno Rule Base Identified from Noisy Data

Rule: Antecedent		Consequent singleton	Consequent label
R1 : If x is *Negative Big*	then	y= -7.908	*Negative Medium*
R2 : If x is *Negative Medium*	then	y= -5.671	*Negative Medium*
R3 : If x is *Negative Small*	then	y=-13.784	*Negative Big*
R4 : If x is *Negative very Small*	then	y= -1.960	*Negative Small*
R5 : If x is *Positive very Small*	then	y= 2.367	*Positive Small*
R6 : If x is *Positive Small*	then	y= 19.524	*Positive Big*
R7 : If x is *Positive Medium*	then	y= 5.943	*Positive Medium*
R8 : If x is *Positive Big*	then	y= 8.022	*Positive Medium*

Overview of Other Learning/Tuning Methods

- Methods based on clustering (ANFIS, GENFIS, etc.)
- Neuro-fuzzy networks
- Genetic optimization of fuzzy systems