Knowledge-Based Control Systems
(SC4081)

Lecture 3: Construction of Fuzzy Systems

Alfredo Núñez
Section of Railway Engineering
CiTG, Delft University of Technology
The Netherlands

a.a.nunezvicencio@tudelft.nl
tel: 015-27 89355

Robert Babuška
Delft Center for Systems and Control
3mE, Delft University of Technology
The Netherlands

r.babuska@tudelft.nl
tel: 015-27 85117

Outline

1. Singleton and Takagi–Sugeno fuzzy system.
2. Dynamic fuzzy systems.
3. Knowledge based fuzzy modeling.
4. Data-driven construction.

Singleton Fuzzy Model

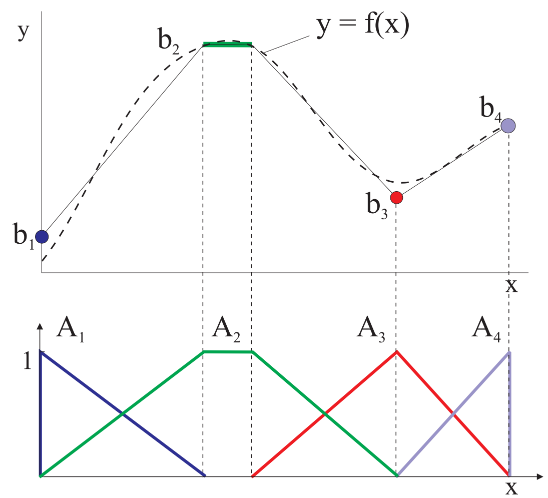
If x is A_i then $y = b_i$

Inference/defuzzification:

$$y = \frac{\sum_{i=1}^{K} \mu_{A_i}(x) b_i}{\sum_{i=1}^{K} \mu_{A_i}(x)}$$

- well-understood approximation properties
- straightforward parameter estimation

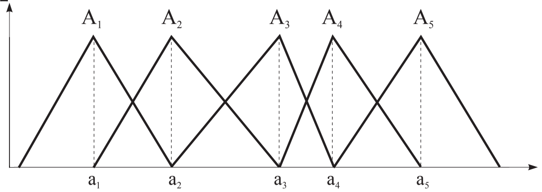
Piece-wise Linear Approximation



Linear Mapping with a Singleton Model

$$y = \mathbf{k}^T\mathbf{x} + q = \sum_{j=1}^{p} k_j x_j + q$$

- **Triangular partition:**



- **Consequent singletons are equal to:**

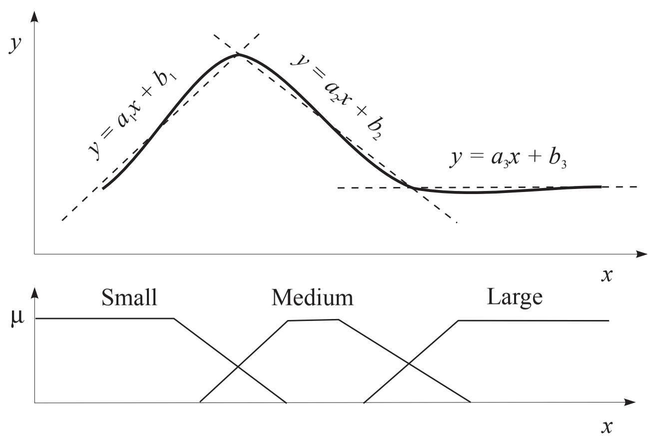
$$b_i = \sum_{j=1}^{p} k_j a_{i,j} + q$$

Takagi–Sugeno (TS) Fuzzy Model

If x **is** A_i **then** $y_i = a_i x + b_i$

$$y = \frac{\sum_{i=1}^{K} \mu_{A_i}(x) y_i}{\sum_{i=1}^{K} \mu_{A_i}(x)} = \frac{\sum_{i=1}^{K} \mu_{A_i}(x)(a_i x + b_i)}{\sum_{i=1}^{K} \mu_{A_i}(x)}$$

Input-Output Mapping of the TS Model



Consequents are approximate local linear models of the system.

TS Model is a Quasi-Linear System

$$y = \frac{\sum_{i=1}^{K} \mu_{A_i}(\mathbf{x}) y_i}{\sum_{j=1}^{K} \mu_{A_j}(\mathbf{x})} = \frac{\sum_{i=1}^{K} \mu_{A_i}(\mathbf{x})(\mathbf{a}_i^T\mathbf{x} + b_i)}{\sum_{j=1}^{K} \mu_{A_j}(\mathbf{x})}$$

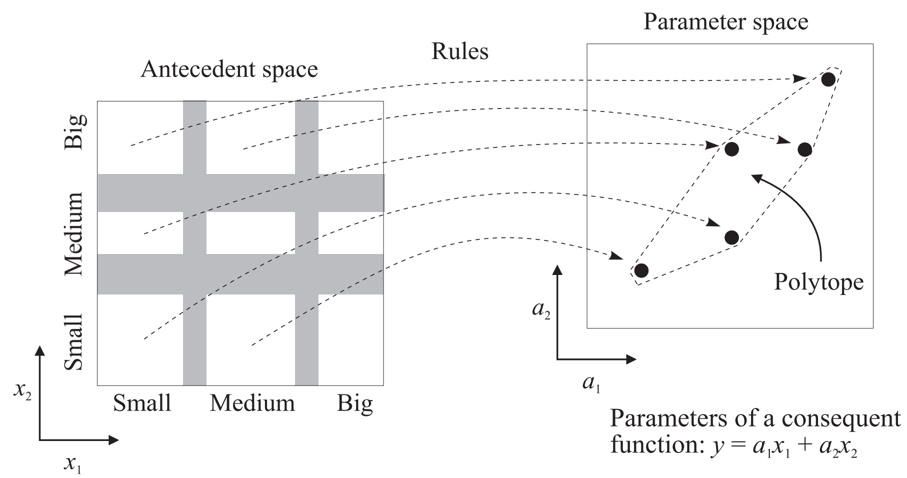
TS Model is a Quasi-Linear System

$$y = \frac{\sum_{i=1}^{K} \mu_{A_i}(\mathbf{x}) y_i}{\sum_{j=1}^{K} \mu_{A_j}(\mathbf{x})} = \frac{\sum_{i=1}^{K} \mu_{A_i}(\mathbf{x})(\mathbf{a}_i^T \mathbf{x} + b_i)}{\sum_{j=1}^{K} \mu_{A_j}(\mathbf{x})}$$

$$y = \underbrace{\left(\sum_{i=1}^{K} \gamma_i(\mathbf{x}) \mathbf{a}_i^T \right)}_{\mathbf{a}(\mathbf{x})} \mathbf{x} + \underbrace{\sum_{i=1}^{K} \gamma_i(\mathbf{x}) b_i}_{\mathbf{b}(\mathbf{x})}$$

linear in parameters a_i and b_i, pseudo-linear in x (LPV)

TS Model is a Polytopic System



Modeling of Dynamic Systems

Nonlinear regression model:

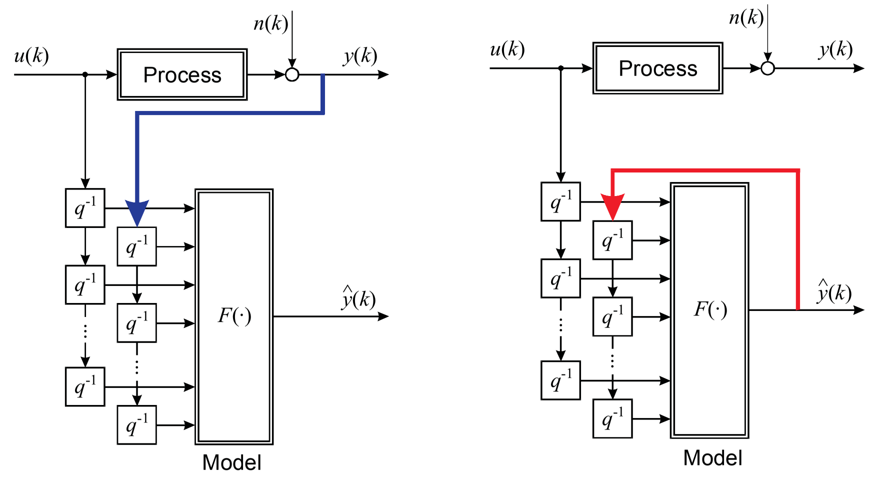
$$\hat{y}(k+1) = F\left[y(k), \ldots, y(k-n_y+1), u(k), \ldots, u(k-n_u+1)\right]$$

In rule-based form (example TS model):

If $y(k)$ **is Small and** $u(k)$ **is Large**

then $\hat{y}(k+1) = \sum_{j=1}^{n_y} a_{ij} y(k-j+1) + \sum_{j=1}^{n_u} b_{ij} u(k-j+1) + c_i$

One-Step-Ahead Prediction vs. Simulation



Construction of Fuzzy Models

Modeling Paradigms

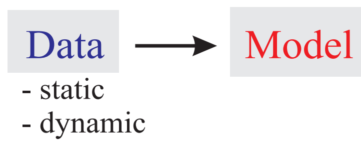
- Mechanistic (white-box, physical)
- Qualitative (naive physics, knowledge-based)
- Data-driven (black-box, inductive)

Often combination of different approaches semi-mechanistic, gray-box modeling.

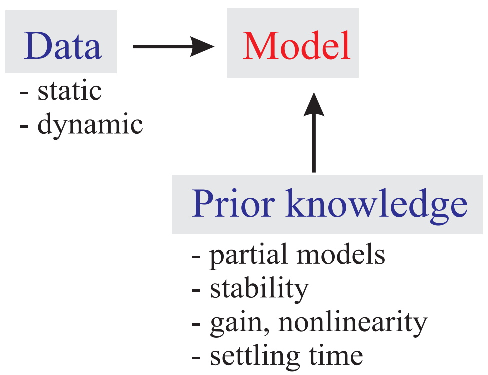
Parameterization of nonlinear models

- polynomials, splines
- look-up tables
- fuzzy systems
- neural networks
- (neuro-)fuzzy systems
- radial basis function networks
- wavelet networks
- ...

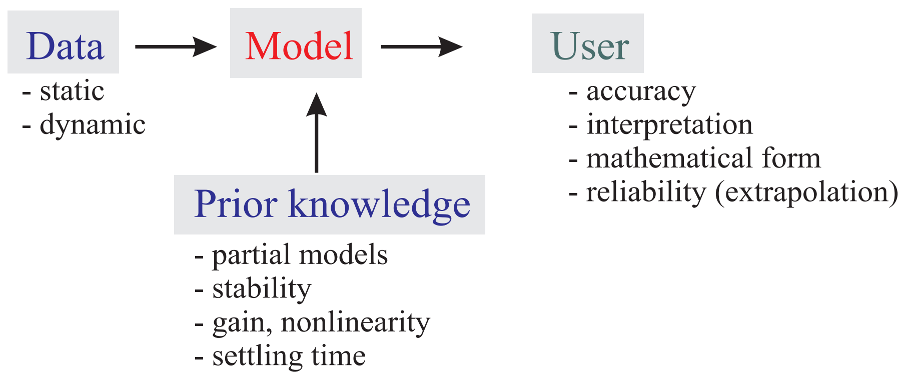
Modeling of Complex Systems



Modeling of Complex Systems



Modeling of Complex Systems



Building Fuzzy Models

Knowledge-based approach:

- expert knowledge $\rightarrow$ rules & membership functions
- fuzzy model of human operator
- linguistic interpretation

Building Fuzzy Models

Knowledge-based approach:

- expert knowledge $\rightarrow$ rules & membership functions
- fuzzy model of human operator
- linguistic interpretation

Data-driven approach:

- nonlinear mapping, universal approximation
- extract rules & membership functions from data

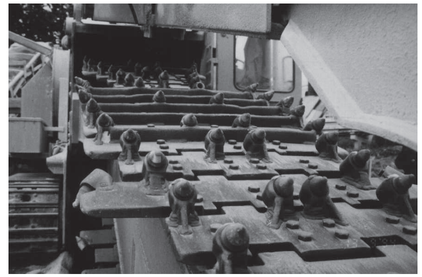
Knowledge-Based Modeling

- Problems where little or no data are available.
- Similar to expert systems.
- Presence of both quantitative and qualitative variables or parameters.

Typical applications: fuzzy control and decision support, but also modeling of poorly understood processes

Wear Prediction for a Trencher

Trencher T-850 (Vermeer)



Chain Detail

Goal: Given the terrain properties, predict bit wear and production rate of trencher.

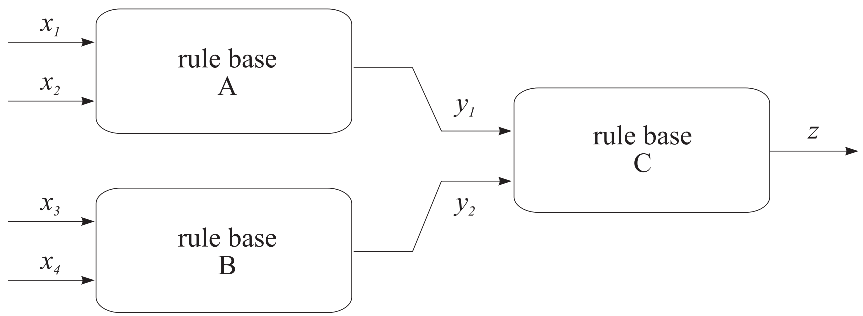
Why Knowledge-Based Modeling?

- Interaction between tool and environment is complex, dynamic and highly nonlinear, rigorous mathematical models are not available.
- Little data (15 data points) to develop statistical regression models.
- Input data are a mixture of numerical measurements (rock strength, joint spacing, trench dimensions) and qualitative information (joint orientation).
- Precise numerical output not needed, qualitative assessment is sufficient.

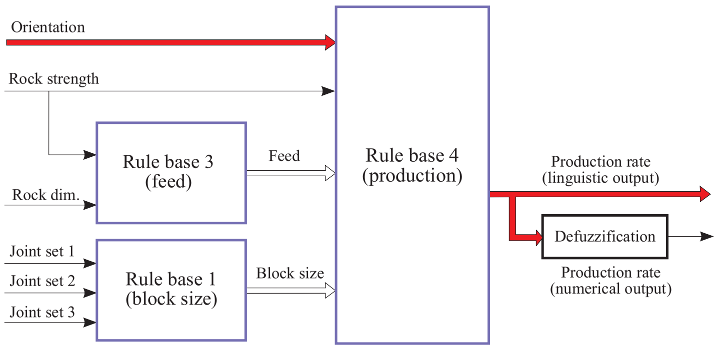
Dimensionality Problem: Hierarchical Structure

Assume 5 membership functions for each input

625 rules in a flat rule base vs. 75 rules in a hierarchical one

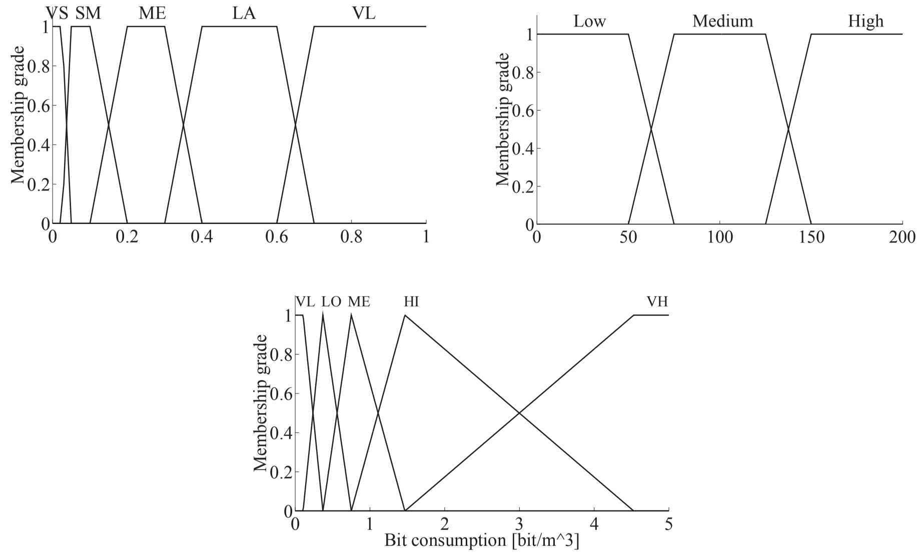


Trencher: Fuzzy Rule Bases




```
If TRENCH-DIM is SMALL  and STRENGTH is LOW    Then FEED is VERY-HIGH;
If TRENCH-DIM is SMALL  and STRENGTH is MEDIUM Then FEED is HIGH;
....
If JOINT-SP is EXTRA-LARGE and FEED is VERY-HIGH Then PROD is VERY-HIGH
```


Example of Membership Functions



Output: Prediction of Production Rate

data no.	measured value	predicted linguistic value(s)			
1	2.07	VERY-LOW	1.00		
2	5.56	HIGH	1.00		
3	23.60	VERY-HIGH	0.50		
4	11.90	HIGH	0.40	VERY-HIGH	0.60
5	7.71	MEDIUM	1.00		
6	7.17	LOW	0.72		
7	8.05	MEDIUM	0.80		
8	7.39	LOW	1.00		
9	4.58	LOW	0.50		
10	8.74	MEDIUM	1.00		
11	134.84	EXTREMELY-HIGH	1.00		

Data-Driven Construction

Structure and Parameters

Structure:

- Input and output variables. For dynamic systems also the representation of the dynamics.
- Number of membership functions per variable, type of membership functions, number of rules.

Parameters:

- Consequent parameters (least squares).
- Antecedent membership functions (various methods).

Least-Squares Estimation of Singletons

$$R_i\text{:}\quad \textbf{If } \mathbf{x} \textbf{ is } A_i \textbf{ then } y = b_i$$

- Given A_i and a set of input–output data:

$$\{(\mathbf{x}_k, y_k) \mid k = 1, 2, \dots, N\}$$

- Estimate optimal consequent parameters b_i.

Least-Squares Estimation of Singletons

1. Compute the membership degrees $\mu_{A_i}(\mathbf{x}_k)$

2. Normalize

$$\gamma_{ki} = \mu_{A_i}(\mathbf{x}_k) / \sum_{j=1}^{K} \mu_{A_j}(\mathbf{x}_k)$$

(Output: $y_k = \sum_{i=1}^{K} \gamma_{ki} b_i$, in a matrix form: $\mathbf{y} = \mathbf{\Gamma b}$)

3. Least-squares estimate: $\mathbf{b} = \left[\mathbf{\Gamma}^T\mathbf{\Gamma}\right]^{-1}\mathbf{\Gamma}^T\mathbf{y}$

Least-Square Estimation of TS Consequents

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{W}_i = \begin{bmatrix} \gamma_{i1} & 0 & \cdots & 0 \\ 0 & \gamma_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_{iN} \end{bmatrix}$$

$$\boldsymbol{\theta}_i = \left[\mathbf{a}_i^T \quad b_i\right]^T, \quad \mathbf{X}_e = [\mathbf{X} \quad \mathbf{1}]$$

Least-Square Estimation of TS Consequents

- **Global LS**: $\boldsymbol{\theta}' = \left[(\mathbf{X}')^T\mathbf{X}'\right]^{-1}(\mathbf{X}')^T\mathbf{y}$

with $\mathbf{X}' = [\mathbf{W}_1\mathbf{X}_e \quad \mathbf{W}_2\mathbf{X}_e \quad \dots \quad \mathbf{W}_c\mathbf{X}_e]$

and $\boldsymbol{\theta}' = \left[\boldsymbol{\theta}_1^T \quad \boldsymbol{\theta}_2^T \quad \dots \quad \boldsymbol{\theta}_c^T\right]^T$

Least-Square Estimation of TS Consequents

- **Global LS**: $\boldsymbol{\theta}' = \left[(\mathbf{X}')^T\mathbf{X}'\right]^{-1}(\mathbf{X}')^T\mathbf{y}$

with $\mathbf{X}' = [\mathbf{W}_1\mathbf{X}_e \quad \mathbf{W}_2\mathbf{X}_e \quad \dots \quad \mathbf{W}_c\mathbf{X}_e]$

and $\boldsymbol{\theta}' = \left[\boldsymbol{\theta}_1^T \quad \boldsymbol{\theta}_2^T \quad \dots \quad \boldsymbol{\theta}_c^T\right]^T$

- **Local LS**: $\boldsymbol{\theta}_i = \left[\mathbf{X}_e^T\mathbf{W}_i\mathbf{X}_e\right]^{-1}\mathbf{X}_e^T\mathbf{W}_i\mathbf{y}$

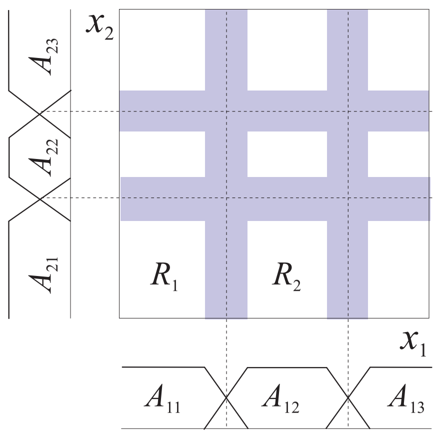
Antecedent Membership Functions

- templates (grid partitioning),
- nonlinear optimization (neuro-fuzzy methods),
- tree-construction
- product space fuzzy clustering

Template-Based Modeling

Determine membership functions a priori (shape, number).

- Only for small problems (1 to 3 inputs).

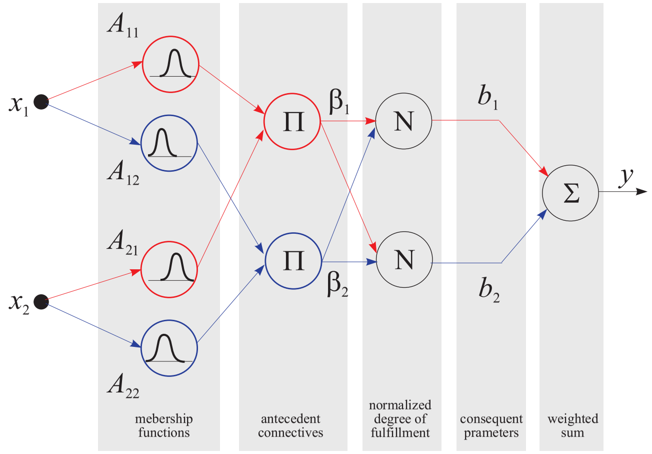


$K = \prod_{j=1}^{p} N_j$

Nonlinear Optimization (Neuro–Fuzzy Learning)

If x_1 is A_{11} and x_2 is A_{21} then $y = b_1$

If x_1 is A_{12} and x_2 is A_{22} then $y = b_2$



Smooth Membership Functions

$$\mu(x; c, \sigma) = \exp\left(-(\frac{x-c}{2\sigma})^2\right)$$

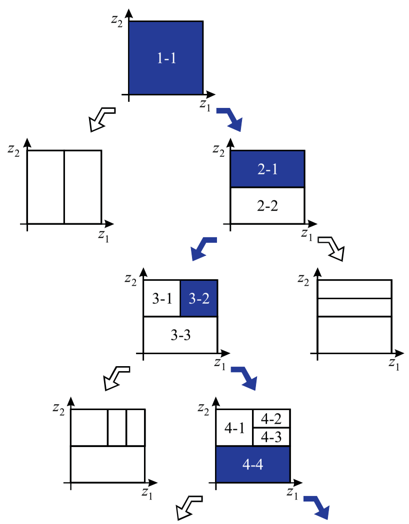
$$y = \frac{\sum_{i=1}^{K} \exp\left(-(\frac{x-c_i}{2\sigma_i})^2\right) b_i}{\sum_{i=1}^{K} \exp\left(-(\frac{x-c_i}{2\sigma_i})^2\right)}$$

Adjust parameters c_i and σ_i (nonlinear optimization):

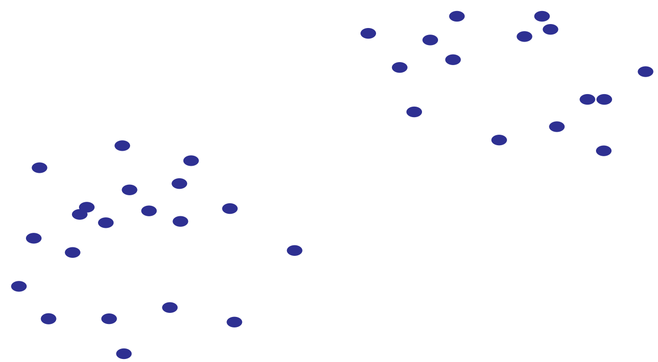
- Gradient-based: (back-propagation, Levenberg-Marquardt).
- Gradient-free: (Nelder-Mead, GA, simulated annealing).

Tree-Construction Methods

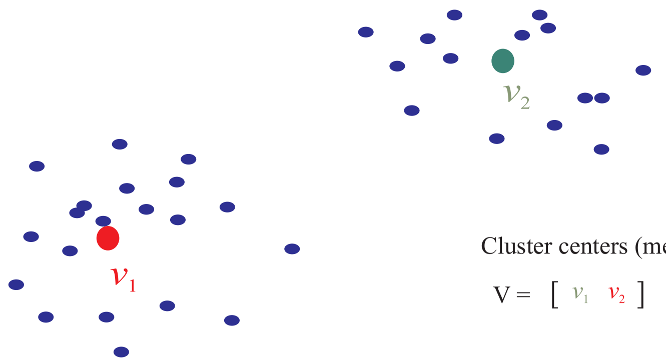
- Growing: Adds one LLM/rule in each iteration
- Axis-orthogonal partition of the input space
- Placement of the membership functions
  1. Division of the worst performing LLM
  2. Test division in each input dimension
  3. Best performing division is realized
- Separate estimation of the newly generated LLMs (weighted least squares)



Fuzzy Clustering: Data



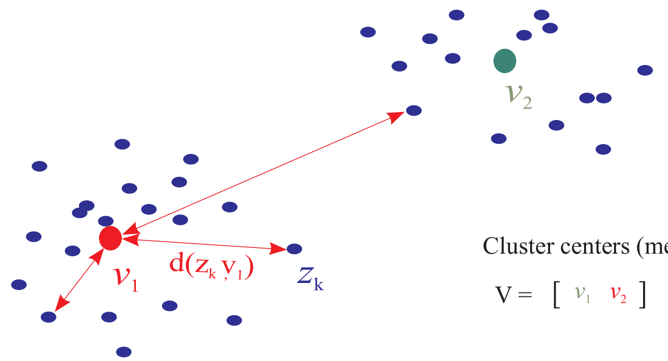
Fuzzy Clustering: Prototypes



Cluster centers (means):

$V = [\ v_1\ \ v_2\ ]$

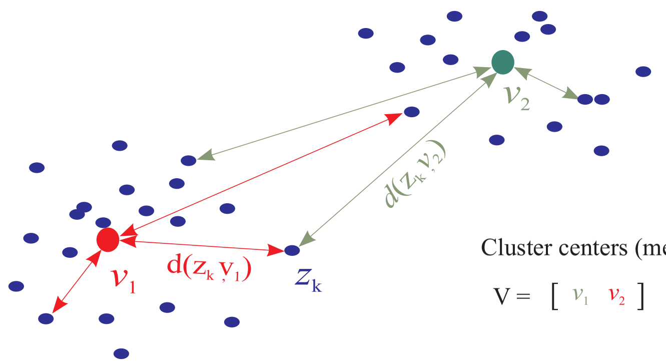
Fuzzy Clustering: Distance



Cluster centers (means):

$V = [\ v_1\ \ v_2\ ]$

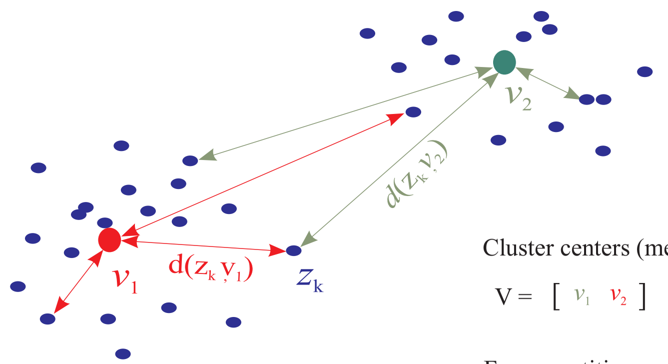
Fuzzy Clustering: Distance



Cluster centers (means):

$V = [\ v_1\ \ v_2\ ]$

Fuzzy Clustering: Partition Matrix



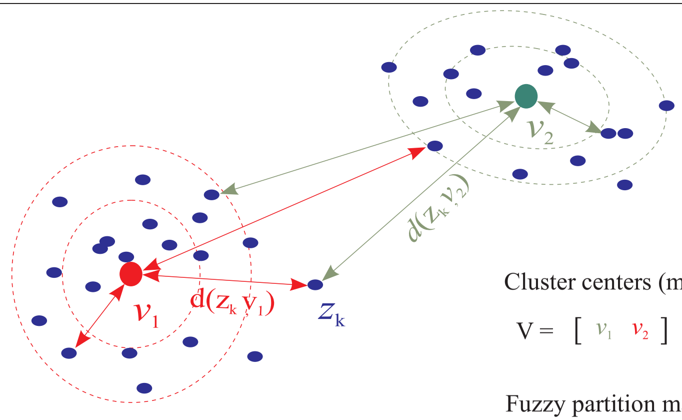
Cluster centers (means):

$V = [\ v_1\ \ v_2\ ]$

Fuzzy partition matrix:

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2N} \end{bmatrix}$$

Fuzzy Clustering: Shapes



Cluster centers (means):

V = [v_1 v_2]

Fuzzy partition matrix:

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2N} \end{bmatrix}$$

Fuzzy Clustering Problem

Given the data:

$$\mathbf{z}_k = [z_{1k}, z_{2k}, \ldots, z_{nk}]^T \in \mathbb{R}^n, \quad k = 1, \ldots, N$$

Find:

the fuzzy partition matrix:

$$\mathbf{U} = \begin{bmatrix} \mu_{11} & \ldots & \mu_{1k} & \ldots & \mu_{1N} \\ \vdots & \ldots & \vdots & \ldots & \vdots \\ \mu_{c1} & \ldots & \mu_{ck} & \ldots & \mu_{cN} \end{bmatrix}$$

and the cluster centers:

$$\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}, \quad \mathbf{v}_i \in \mathbb{R}^n$$

Fuzzy Clustering: an Optimization Approach

Objective function (least-squares criterion):

$$J(Z; V, U, A) = \sum_{i=1}^{c} \sum_{j=1}^{N} \mu_{i,j}^m d_{A_i}^2(\mathbf{z}_j, \mathbf{v}_i)$$

subject to constraints:

$$0 \le \mu_{i,j} \le 1, \qquad i = 1, \ldots, c, \; j = 1, \ldots, N \qquad \textbf{membership degree}$$

$$0 < \sum_{j=1}^{N} \mu_{i,j} < 1, \qquad i = 1, \ldots, c \qquad \textbf{no cluster empty}$$

$$\sum_{i=1}^{c} \mu_{i,j} = 1, \qquad j = 1, \ldots, N \qquad \textbf{total membership}$$

Fuzzy c-Means Algorithm

Repeat:

1. Compute cluster prototypes (means): $v_i = \dfrac{\sum_{k=1}^{N} \mu_{i,k}^m \mathbf{z}_k}{\sum_{k=1}^{N} \mu_{i,k}^m}$

Fuzzy c-Means Algorithm

Repeat:

1. Compute cluster prototypes (means): $v_i = \frac{\sum_{k=1}^{N} \mu_{i,k}^m \mathbf{z}_k}{\sum_{k=1}^{N} \mu_{i,k}^m}$

2. Calculate distances: $d_{ik} = (\mathbf{z}_k - \mathbf{v}_i)^T(\mathbf{z}_k - \mathbf{v}_i)$

Fuzzy c-Means Algorithm

Repeat:

1. Compute cluster prototypes (means): $v_i = \frac{\sum_{k=1}^{N} \mu_{i,k}^m \mathbf{z}_k}{\sum_{k=1}^{N} \mu_{i,k}^m}$

2. Calculate distances: $d_{ik} = (\mathbf{z}_k - \mathbf{v}_i)^T(\mathbf{z}_k - \mathbf{v}_i)$

3. Update partition matrix: $\mu_{ik} = \frac{1}{\sum_{j=1}^{c}(d_{ik}/d_{jk})^{1/(m-1)}}$

until $\|\Delta \mathrm{U}\| < \epsilon$

Distance Measures

- Euclidean norm:

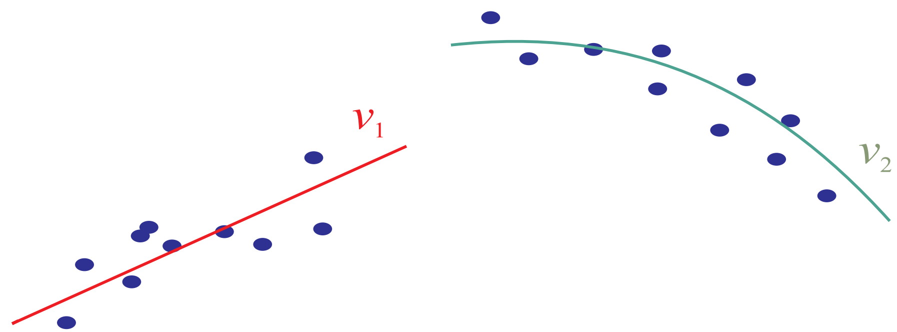
$$d^2(\mathbf{z}_j, \mathbf{v}_i) = (\mathbf{z}_j - \mathbf{v}_i)^T(\mathbf{z}_j - \mathbf{v}_i)$$

- Inner-product norm:

$$d^2_{A_i}(\mathbf{z}_j, \mathbf{v}_i) = (\mathbf{z}_j - \mathbf{v}_i)^T A_i(\mathbf{z}_j - \mathbf{v}_i)$$

- Many other possibilities ...

Generalized Prototypes



lines, circles, ellipses, functions, etc.

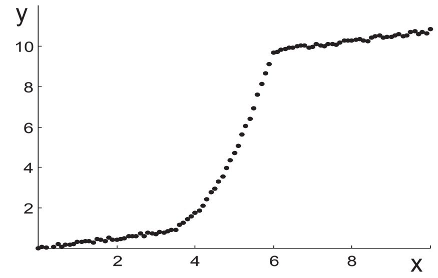
Fuzzy Clustering – Demo

1. Fuzzy c-means

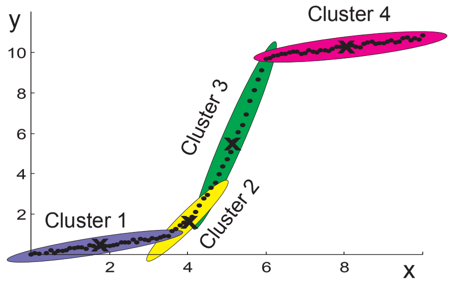
2. Clustering with adaptive distance measure

3. Line detection by clustering

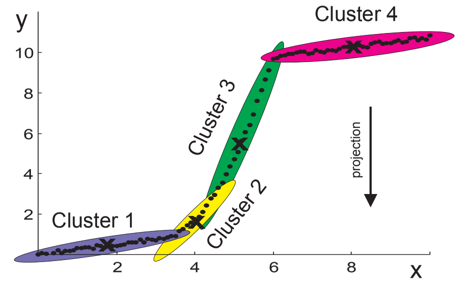
Extraction of Rules by Fuzzy Clustering



Extraction of Rules by Fuzzy Clustering



Extraction of Rules by Fuzzy Clustering

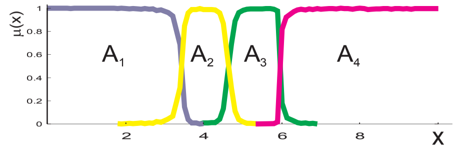


Takagi-Sugeno model

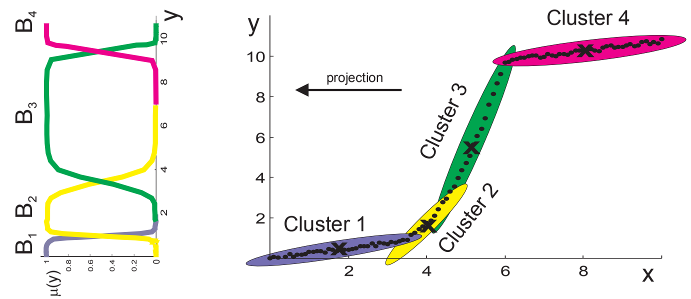
Rule-based description:

If x is A_1 then $y = a_1x + b_1$
If x is A_2 then $y = a_2x + b_2$

etc...



Extraction of Rules by Fuzzy Clustering



Rule-based description:

If y is B_1 then $x = a_1 y + b_1$
If y is B_2 then $x = a_2 y + b_2$

etc...

Inverse Takagi-Sugeno model

Rule Extraction – Demo

- Extraction of Takagi–Sugeno rules

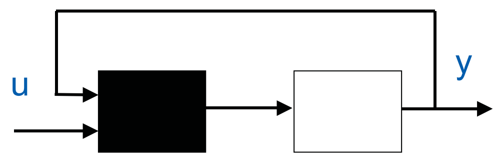
Semi-Mechanistic Modeling

- White-box model developed for well-known parts.
- Black-box or knowledge-based model for unknown relationships.

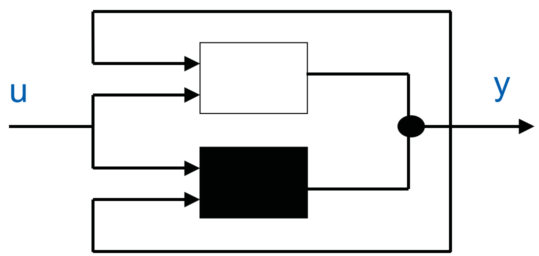
+ effective use of all available information
+ extrapolation, scalability
+ short development time

Semi-Mechanistic Modeling – Structures

Serial

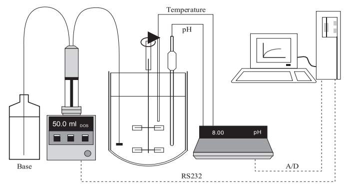


Parallel



Semi-Mechanistic Modeling: Example

PenG + H_2O $\xrightarrow{\text{enzyme}}$ 6-APA + PhAH



Mechanistic model (balance equations)

$$
\begin{aligned}
B_{k+1} &= B_k + \Delta T \frac{[E]_k \cdot V_k \cdot r_e}{M_B} \\
V_{k+1} &= V_k + \Delta T \frac{[E]_k \cdot V_k \cdot r_e}{M_B} \\
[\text{PenG}]_{k+1} &= \frac{V_k}{V_{k+1}} ([\text{PenG}]_k \quad \Delta T \cdot R_{\text{G}} \cdot [E]_k \cdot r_e) \\
[6-\text{APA}]_{k+1} &= \frac{V_k}{V_{k+1}} ([6-\text{APA}]_k + \Delta T \cdot R_{\text{A}} \cdot [E]_k \cdot r_e) \\
[\text{PhAH}]_{k+1} &= \frac{V_k}{V_{k+1}} ([\text{PhAH}]_k + \Delta T \cdot R_{\text{P}} \cdot [E]_k \cdot r_e) \\
[E]_{k+1} &= \frac{V_k}{V_{k+1}} [E]_k
\end{aligned}
$$