Evolving Intelligent Systems

Dr. Plamen Angelov

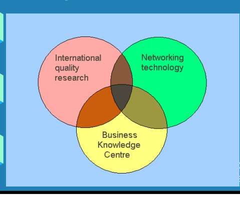
Director Intel.& Robotic Systems Program
Dept of Communications Systems
InfoLab21
Lancaster University, UK

Lancaster University

One of the top 10% research-led Universities in UK

InfoLab21

- **ICT force of 250+** researchers
- Research project income from Industry (Nokia, Philips, BAE Systems, QinetiQ, Ford), Government, EU, USA etc. £15M+



Aim

The aim of this series of lectures is to introduce you with

- *the fundamentals;*
- *methodology;*
- *design;*
- *algorithms;*
- *implementation and applications*

*of **evolving intelligent system** paradigm.*

Context

The lectures will introduce the rationale and basics of this emerging topic while drawing close links with well established methodologies and algorithmic solutions for complex problems such as:

- machine learning
- clustering
- classification
- adaptive systems
- mining data streams
- feature selection, etc.

Style

We will mix

- the fundamental and theoretical presentation with
- more practice oriented experimental results that stem from our industry-oriented research

The style will also attempt a mixture of

- didactic with
- research,
- applications,
- and colloquial aspects

Outline

Lecture 1 – Introduction, Concept

1. Motivation, Challenges
2. System structure and its identification
3. NN, Fuzzy Systems, NF Systems
4. from Adapting to Evolving
5. Basics of Evolving Systems
6. Evolving Fuzzy Systems concept
7. Data space partitioning -> ES structure
8. Principles of evolution

Lecture 2 – Algorithms

Lecture 3 – Applications, Examples

Recommended Readings

- P. Angelov, Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems, Physica-Verlag: Heidelberg, February 2002, ISBN 3-7908-1457-1.
- N. Kasabov, Evolving Connectionist Systems: Methods and Applications in BioInformatics, Brain Study and Intelligent Machines, Springer, London, 2003, ISBN: 1-85233-400-2.
- P. Angelov, D. Filev and N. Kasabov (Eds.), Evolving Intelligent Systems: Methodology and Applications, 484pp., John Willey and Sons, IEEE Press Series on Computational Intelligence, April 2010, ISBN: 978-0-470-28719-4

Motivation

- ✓ Process plants in industry are heavily instrumented with a variety of sensors and control loops,

- ✓ But the level of human involvement at various stages of the process life-cycle is still high

Motivation

- ✓ But process plants are not static environments.
- ✓ They undergo continuous process changes, developments and updates, their components get older, contaminated
- ✓ Due to limited self-learning or self-adapting capabilities of present monitoring and control techniques a considerable amount plant capacity is lost, product quality varies, energy consumption and emissions get higher than under optimal conditions

Motivation

- ✓ A real illustrative example (Evonik/Degussa, Germany):
- ✓ A typical polymerisation plant producing roughly 300 000 ton pa
- ✓ Due to unwanted polymerisation such plants can require cleaning once a month for ≈3 days for each downtime, the costs result in about €1.5 M pa

Motivation

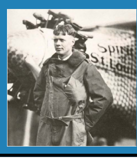
- ✓ If 80% of these downtimes can be avoided by applying evolving intelligent sensors;
- ✓ This can result in savings of about €1.2 M plus the increased capacity of the plant (about 30 000 ton pa)
- ✓ The chemical industry in Europe and US has hundreds of such plants with.
- ✓ This emphasises the obvious large scale of the overall potential savings.

The challenge

'We are drowning in information and starving for knowledge'

R.D. Roger

- Huge volumes of streaming data
  - Defence (autonomous systems)
  - Surveillance and security
  - Advanced industrial processes
  - Internet



The challenge

- Data streams cannot be analyzed in a batch mode, since storing and manipulating the complete data is often impossible

- Instead, systems have to be developed that extract *knowledge* from the data streams in *real-time, on-line*

The challenge

- Knowledge extraction refers to identifying valid and interpretable *structure*
- In control theory – structure identification
- For non-stationary data it is logical to assume a dynamic/evolving structure
- Evolving (Oxford dictionary) – gradually developing, higher level of adaptation

The challenge

- The environment in which real systems (surveillance and robotic/autonomous systems, technological processes etc.) operate is (unpredictably) changing
- The challenge - to develop systems capable of higher level adaptation to the environment and to internal changes (changing behaviour, wearing, faults, regimes, tiredness etc.)

Challenges

There are several challenges that we address:

- even such routine pre-processing techniques like *normalisation and standardisation* when applied to data streaming on-line are not trivial
- the mechanisms for *extracting model/sensor structure form streaming data* with recursive techniques (low memory and comput.) costs
- On-line input variables selection

Challenges

- learning mechanisms with evolving structure and (globally) non-linear and non-stationary characteristics
- to design an algorithm that does not have any user- or problem-specific parameter (thresholds, number of clusters, rules, neurons, learning rates etc.) will bring the level of automation higher
- but is a serious challenge if that algorithm is to be successful in a wide range of applications

Challenges

- ✓ to combine

The mathematical rigour and computational efficiency of data-driven learning techniques with the transparency and human-intelligible form of the knowledge extracted from the data streams

in real-time

- ✓ To preserve the cause-effect relations
- ✓ scalability

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 19

The challenge

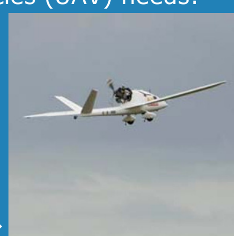
- Current systems that posses *Computational Intelligence* usually rely on fixed rule-bases or NN
- Trained off-line, do not adapt to the environment
- They do not develop their structure (do not evolve)

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 20

Example UAV

Unmanned Aerial Vehicles (UAV) needs:

- to be flexible
- to adapt
- to learn new environment



Herti UAV courtesy of BAE →

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 21

Example Mobile robot

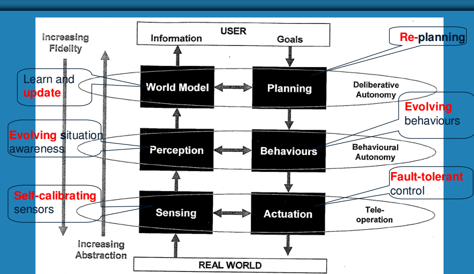
Mobile robots needs:

- to capture new knowledge
- to adapt
- to learn new environment

the de-miner ELTA →

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 22

Autonomous Systems



3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 23

Autonomous Systems, UxV

- Need for autonomy (adapt routing, tasks)
- Need to learn new environment and adapt to internal and external changes

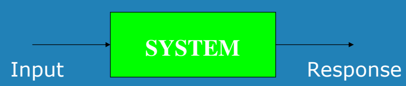
UAV acting as ROV Oct. 2008 Arizona

Land-based, ROV Talon, QQ

- Same problems, different environmental parameters (speed, terrain, constraints)

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 24

System Modeling



- predict object reactions;
- control it;
- detect faults;
- study process performance

System Modeling

A. Linear
B. Non-linear

A. Stationary
B. Non-stationary

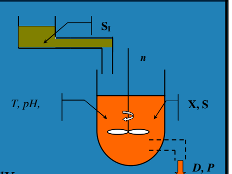
A. *Conventional* Models
  - First Principles Models
  - Black-box models

B. Fuzzy Models

First Principle Model

A. First Principle Models

- Example: Fermentation process
  - ✓ **transparent, close to nature** (mass- and energy conservation in closed systems)
  - × **tedious**, even impossible, (highly) non-linear



$$\frac{dX}{dt} = \mu_X X - DX$$

$$\frac{dS}{dt} = -q_S X + D(S_I - S)$$

$$\frac{dP}{dt} = q_P X - DP$$

Black-box Models

- Linear state-space models

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k) + Du(k)$$

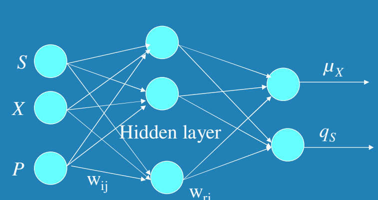
- Polynomial models

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2 + a_6x_1^2x_2^2 + \ldots$$

- ARMA models

$$y(k) = a_0 + a_1x(k-1) + a_2x(k-2) + \ldots + b_0 + b_1y(k-1) + b_2y(k-2) + \ldots$$

Neural Networks



Fuzzy Rule-based Models

- FRB systems are universal approximators
- Thus they are suitable for extracting interpretable knowledge
- And are a promising framework for designing powerful prognostic, classification, and control systems

Fuzzy sets

"So far as the laws of mathematics refer to **reality**, they are **not certain**. And so far they are **certain**, they **do not** refer to **reality**."

Albert Einstein

"Everything is a matter of **a degree**"
Bart Kosko

Fuzzy sets

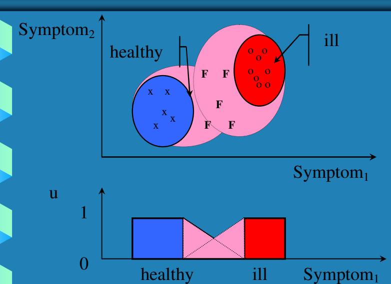
- A fuzzy set - extension of the normal set
- with the main difference that an object can *belong partially* to the fuzzy set.
- Remember than in normal set theory, an object has two options in that respect:
- a) to belong to the set, $x_j \in C_i$
- b) not to belong to the set, $x_j \notin C_i$

Partial membership

- If we use the following notation for membership we can write for the first case, $\mu_{ij}=1$
- For the second case we will have, $\mu_{ij}=0$
- Assuming ***fuzzy set*** describing the membership to cluster we have $0 \leq \mu_{ij} \leq 1$ *where*

$$\sum_{i=1}^{k} \mu_{ij} = 1$$

Fuzzy partitioning

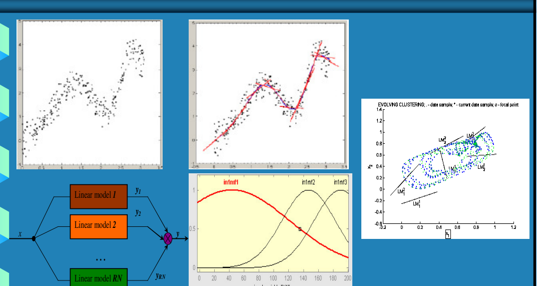


TSK Models (1985)

- TSK systems – important tool for system modeling and identification
  - ✓ Computational efficiency (local linearity)
  - ✓ Universal approximators
  - ✓ Good transparency
  - ✓ Convenient for data-driven design

$$R_i : IF(x_1 is X_1) AND ... AND (x_n is X_n)$$
$$THEN(y_i = a_{i1}x_1 + ... + a_{in}x_n + b_i)$$

TS fuzzy model (concept)



Fuzzy and Neuro-fuzzy

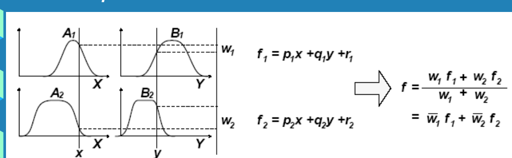
- (Fuzzy) Rule-based Systems – convenient vehicle to *integrate* human (expert) knowledge and knowledge learned from data
- Evolving (Neuro-) Fuzzy Rule-based Systems– *autonomous learning in a human intelligible way from experience in real-time in a dynamic environment*
- *Deterministic, not random ➡ predictable and reproducible*
- *Evolving (self-developing) is a higher level of adaptation (system structure + parameters)*

ANFIS

- ANFIS combines learning capabilities of NN and expressiveness of fuzzy systems
- Suppose, that the rule-base contains two fuzzy IF-THEN rules of TSK type:
- *Rule 1: IF (x is A_1) AND (y is B_1) THEN ($f_1=p_1x+q_1y+r_1$)*
- *Rule 2: IF (x is A_2) AND (y is B_2) THEN ($f_2=p_2x+q_2y+r_2$)*

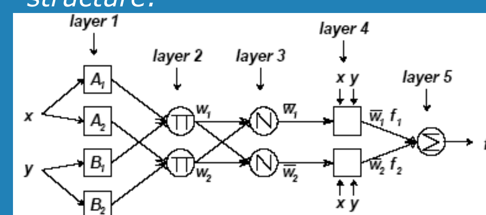
ANFIS

- *Then the reasoning/inference can be represented as follows:*



ANFIS

This is equivalent to the following structure:



ANFIS Functioning Layer 1

- *Every node in Layer 1 performs the following function*

$$O_i^1 = \mu_{A_i}(x)$$

where x is the input to node i;

*A_i is the linguistic label (**small**, **large**, etc.) associated with this node function.*

O_i is the membership function of A_i (it specifies the degree to which given x satisfies the quantifier A_i)

ANFIS Functioning Layer 2

Every node in this layer is a AND node labelled ∏ which multiplies the incoming signals and sends the product (multiplication, denoted also by ×) out:

$$\omega_i = \mu_{A_i}(x) \times \mu_{A_i}(x)$$

Each node output represents the firing strength of a rule.

ANFIS Functioning Layer 3

Every node in this layer is a normalisation node labelled N.

The i^{th} node calculates the ratio of the ith rule's firing strength to the sum of all rules' firing strengths:

$$\overline{\omega_i} = \frac{\omega_i}{\omega_1 + \omega_2}; i = 1,2,...$$

Outputs of this layer are called *normalised firing strengths*.

ANFIS Functioning Layer 4

Every node, i in this layer has a function

$$O_i^4 = \overline{\omega}_i f_i = \overline{\omega}_i (p_i x + q_i y + r_i)$$

where $\overline{\omega}_i$ *is the output of layer 3, and*

$p_i x + q_i y + r_i$ *is the consequent parameter set*

ANFIS Functioning Layer 5

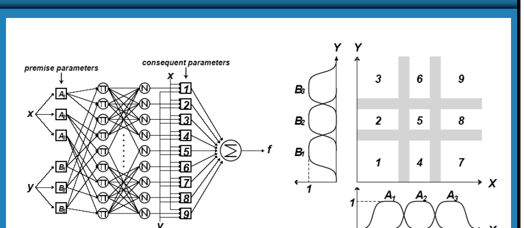
The single node in this layer is a circle node labelled ∑ that computes the overall output as the summation of all incoming signals, i.e.:

$$O_i^5 = overall \quad output = \sum_i \overline{\varpi}_i f_i = \frac{\sum_i \varpi_i f_i}{\sum_i \varpi_i}$$

ANFIS Architecture

- Next Figure shows an ANFIS with 2 inputs (*x* and *y*) and 9 fuzzy rules.
- Three membership functions are associated with each input, so the input space is partitioned into 9 fuzzy subspaces, each of which governed by a fuzzy rule.
- The premise part of a rule defines a fuzzy sub-space, while the consequent part specifies the output within this subspace (see the right plot)

ANFIS Architecture



ANFIS Rule-base

- *Rule 1: IF (x is A1) AND (y is B1) THEN (f1=p1x+q1y+r)*
- *Rule 2: IF (x is A2) AND (y is B1) THEN (f2=p2x+q2y+r2)*
- *Rule 3: IF (x is A3) AND (y is B1) THEN (f3=p3x+q3y+r3)*
- *Rule 4: IF (x is A1) AND (y is B2) THEN (f4=p4x+q4y+r4)*
- *Rule 5: IF (x is A2) AND (y is B2) THEN (f5=p5x+q5y+r5)*
- *Rule 6: IF (x is A3) AND (y is B2) THEN (f6=p6x+q6y+r6)*
- *Rule 7: IF (x is A1) AND (y is B3) THEN (f7=p7x+q7y+r7)*
- *Rule 8: IF (x is A2) AND (y is B3) THEN (f8=p8x+q8y+r8)*
- *Rule 9: IF (x is A3) AND (y is B3) THEN (f9=p9x+q9y+r9)*

ANFIS Learning

- *To train an ANFIS we need to determine two groups of parameters (weights):*
  - *Parameters of the antecedent (premise) part: these include the centres, c_i and spreads, σ_i the Gaussian membership functions;*
  - *Parameters of the consequent part (p_i, q_i, r_i).*

ANFIS Learning

- *The consequent parameters are determined used so-called Least-Squares (LS) or its recursive version (RLS). This will be detailed in the next lecture (L2)*
- *The premise parameters are found by ∇-based algorithm, which we have already considered in the previous lecture (the version of the ∇ approach applied to NN is called error back-propagation).*

ANFIS Learning

- *Learning of ANFIS is very efficient and they are suitable for off-line training and adaptation, but the EBP algorithm is iterative and the application for on-line (real-time) problems is difficult.*
- *Very often validation errors of an ANFIS are high despite that the training errors are low (their generalization capabilities are not very high.*
- *You can find ANFIS in MATLAB program anfis.m which is part of the Fuzzy Logic Toolbox.*

A Historical Perspective

- Until 1990s *intelligent* systems were designed primarily based on *'expert' knowledge* and *fixed structure* models
- Since mid-'90s *data-driven* design methods that aim to *extract knowledge* from the data
- Since 2000– a new trend– *evolving* intelligent systems, *on-line*, *real-time* processing data *streams* - Lancaster is a recognized World leader in this emerging area of research

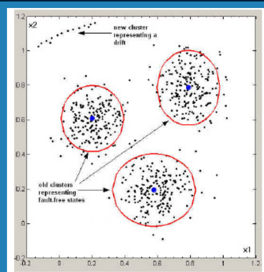
Evolving Intelligent Systems

- *Evolving* Intelligent systems = *i*S with *self-adaptive structure*
- *Evolving* is more than just:
  - *Adaptive*
  - *On-line*
  - *Incremental*
  - *Multi-model* (fixed set of models)
- *Evolving* is different from *evolutionary*

Adaptive → Evolving

- What to do when **new data do not fit** into the model with a chosen structure?
- Adaptive systems theory answer ('70s): adapt the parameters ONLY
- This may be an **outlier**
- Or it may **bring new information (knowledge)** – about a different regime, operation point, change etc.
- Thus update the **structure**

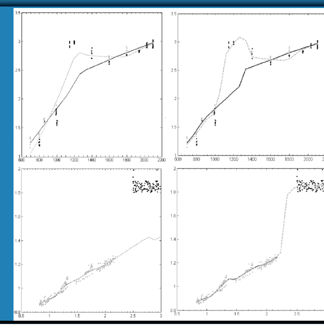
Outlier or a new (cluster/rule)



3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid

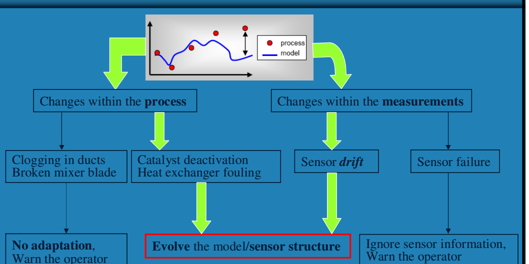
55

Adaptive vs Evolving



56

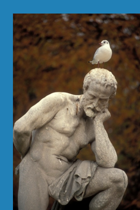
When to evolve?

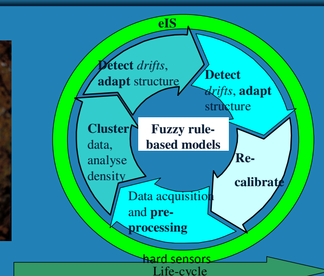


3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid

57

The approach





3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid

58

Industrial Recognition

"Evolving Intelligent Systems have a high potential for implementation in industry"

Dr. A. Kordon, R&D Leader, Dow Chemical, TX

Symposia on this subject – *regularly* since 2006 (Ambleside), London (2007), Witten, Germany (2008), Nashville, USA (2009), Leicester (2010), Paris (2011)

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid

59

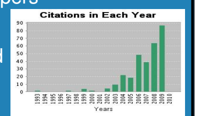
Academic Recognition

- Founding a Task Force on Evolving and Adaptive Fuzzy Systems, Computational Intelligence Society, IEEE
- A special issue of IEEE Transactions on *Evolving* fuzzy systems
- Founding *International Journal on Evolving Systems* by Springer
- A number of invited plenary lectures and key note talks

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid

Publications

- Hundreds of high impact journal papers
- Hundreds of peer reviewed major conference papers including invited
- patents
- Books (research monograph, edited volumes)
- Special sessions
- Tutorials at major conferences, etc…

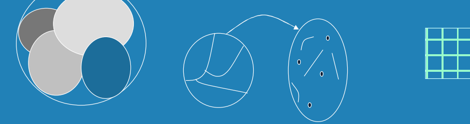


3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 61

Evolving Systems Concept

System Decomposition to Granules and Corresponding Nonlinear Mappings

$\Re^i$: **IF** (x_1 is $\aleph_1^i$) **AND … AND** (x_n is $\aleph_n^i$)
THEN $y^i = f_i(x)$ i=[1, R]



3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 62

Evolving Systems Concept

Universal Approximators of Nonlinear Mappings

$$y = f(x)$$

Evolving Systems = Granular Systems with Open Multiple Model Structure, Learning & Summarization Capability

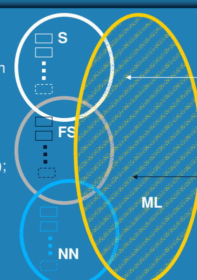
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 63

Evolving Systems in context

Multiple Linear Models; Boolean Switching

Multiple Rule or Linear Models (TS); Soft Switching

Multiple Neurons (RBF); Linear Combiner



Evolving Linear Systems

Evolving Fuzzy Systems

Evolving Neural Systems

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 64

Rationale for Evolving Systems

- Reflects human ability to acquire, summarize, and manage knowledge from facts "on a fly"
- Real time learning of systems that have variable structure and parameters, i.e. multiple operating modes and scenarios
- Capability for learning, managing, fusing, and aggregating multiple models
- Evolving concept has its roots in the area of Computational Intelligence

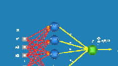
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 65

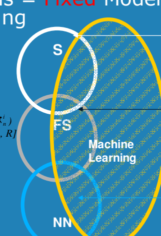
Evolving Systems vs. Adaptive Systems

Adaptive Systems = Fixed Model Structures + Real Time Learning

$y(t) = f(\mathbf{x}, \mathbf{u}(t - td))$

$\Re^i$: IF (x_1 is $\aleph_1^i$) AND … AND (x_n is $\aleph_n^i$) THEN $y^i = f_i(x)$ i=[1, R]





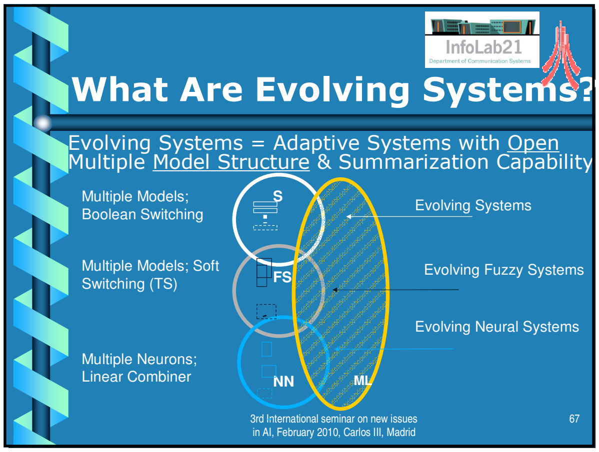
Adaptive Systems

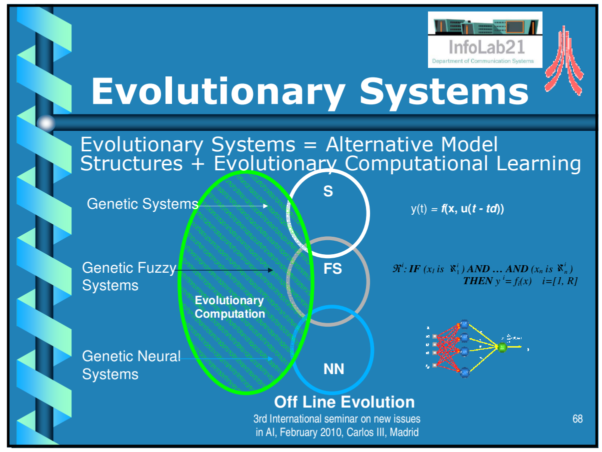
Adaptive Fuzzy Systems

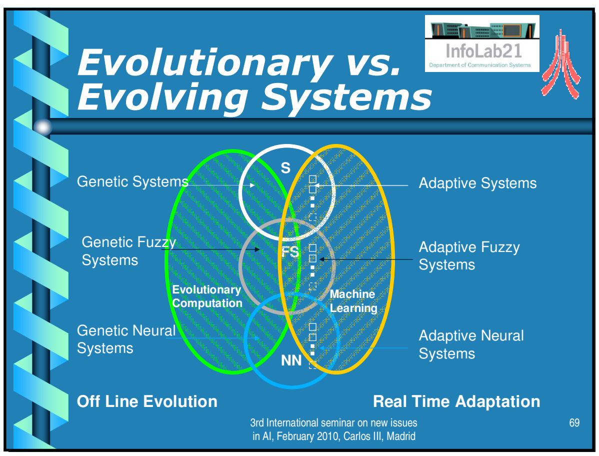
Adaptive Neural Systems

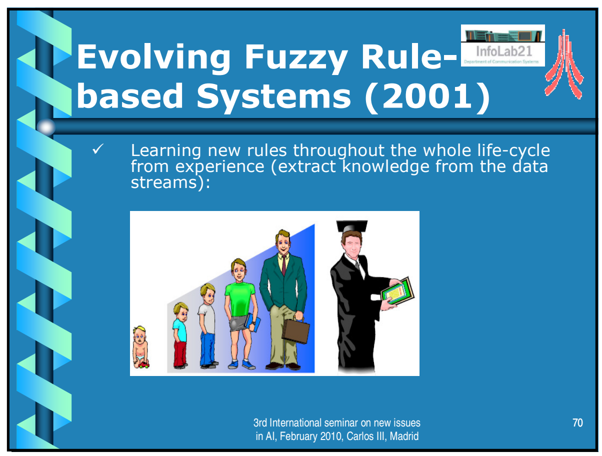
Real Time Adaptation

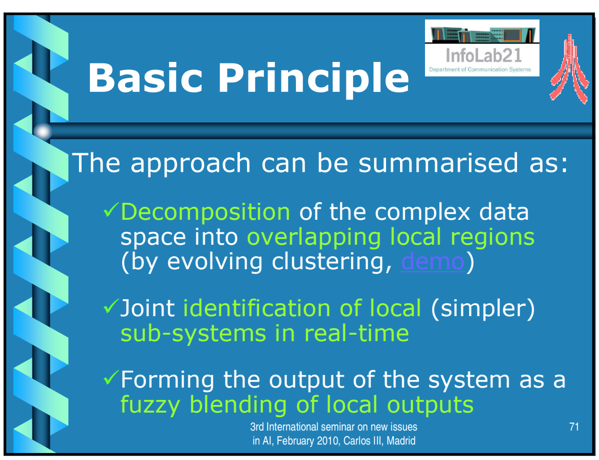
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 66

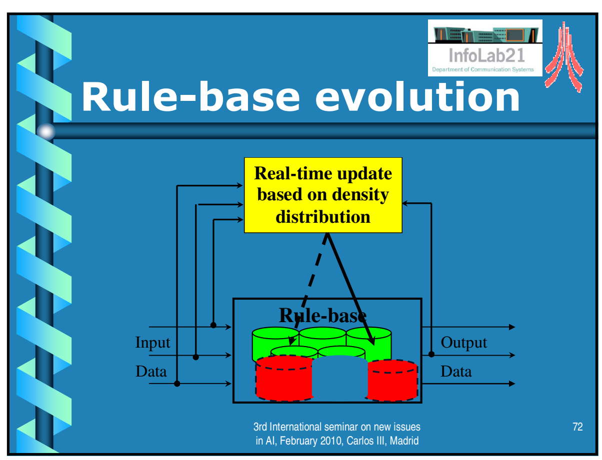

InfoLab21
What Are Evolving Systems?
Evolving Systems = Adaptive Systems with Open Multiple Model Structure & Summarization Capability
Multiple Models; Boolean Switching
Multiple Models; Soft Switching (TS)
Multiple Neurons; Linear Combiner
S
FS
NN
ML
Evolving Systems
Evolving Fuzzy Systems
Evolving Neural Systems
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid
67


InfoLab21
Evolutionary Systems
Evolutionary Systems = Alternative Model Structures + Evolutionary Computational Learning
Genetic Systems
Genetic Fuzzy Systems
Genetic Neural Systems
Evolutionary Computation
S
FS
NN
y(t) = f(x, u(t - td))
Off Line Evolution
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid
68

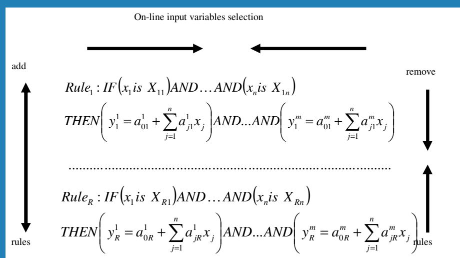

InfoLab21
Evolutionary vs. Evolving Systems
Genetic Systems
Genetic Fuzzy Systems
Genetic Neural Systems
Evolutionary Computation
S
FS
NN
Machine Learning
Adaptive Systems
Adaptive Fuzzy Systems
Adaptive Neural Systems
Off Line Evolution
Real Time Adaptation
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid
69


InfoLab21
Evolving Fuzzy Rule-based Systems (2001)
✓ Learning new rules throughout the whole life-cycle from experience (extract knowledge from the data streams):
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid
70


InfoLab21
Basic Principle
The approach can be summarised as:
✓Decomposition of the complex data space into overlapping local regions (by evolving clustering, demo)
✓Joint identification of local (simpler) sub-systems in real-time
✓Forming the output of the system as a fuzzy blending of local outputs
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid
71


InfoLab21
Rule-base evolution
Real-time update based on density distribution
Rule-base
Input Data
Output Data
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid
72

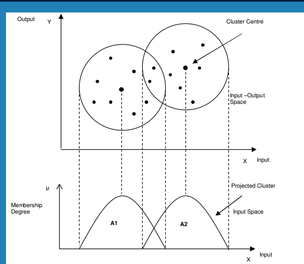
Rule-base evolution



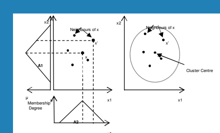
Data Space partitioning

- Goal: Divide and conquer – complex (non-linear problems to be decomposed)
- Technique: Identify sub-structures in the data space that will be mapped (projected) onto the input/output axes (features/classes).
- Approaches:
- divide evenly
- Cluster on-line without a pre-fixed structure

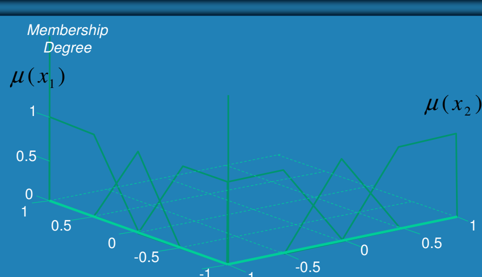
Data Space partitioning



Data Space partitioning

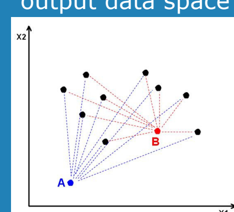


Equal partitioning



Density-based partitioning

Key notion – spatial proximity in the input/ output data space (Caushy type kernel)



$$P_A = \frac{1}{1 + 1/(k_A - 1)\sum_{i=1}^{k_A} d_{A,i}}$$

$$P_B = \frac{1}{1 + 1/(k_B - 1)\sum_{i=1}^{k_B} d_{B,i}}$$

$$P_A < P_B$$

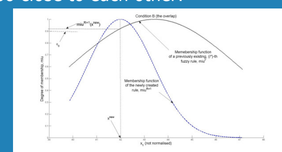
Principles of evolution

The fuzzy rule-base is formed according to the following principles:

- A1) a data sample that have *high density* is eligible to be a focal point of a fuzzy rule
  $$P(z(k)) > \max_{i=1}^{R} P(z^*(k))$$
- A2) a data sample that lies in an area of data space **not covered** by other fuzzy rules is also eligible to form a fuzzy rule $P(z(k)) < \min_{i=1}^{R} P(z^*(k))$
- B3) avoid overlap and information redundancy in forming new fuzzy rules.
  $$\exists i, i = [1, R]; \mu_{ij}(x(k)) > e^{-1}; \forall j; j = [1, n]$$

Principles of evolution

- The third principle, 3) (Condition B) gives the possibility of the rule-base to *gradually shrink*
- This is important for the focal points formed based on 1) which lie too close to each other.



- It leads to simpler rule-base compare to other clustering methods such as ART, VQ etc. which require 'pruning'

Principles of evolution

- C4) remove/ignore *old* clusters/rules (with high *age*)
  $$IF\left(A_k^i < meanA + std(A)\right) THEN\left(\lambda^i \leftarrow 0\right)$$
- D4) remove/ignore clusters with *low support*
  $$IF\left(S_k^i < n(n-1)\right) THEN\left(\lambda^i \leftarrow 0\right)$$
- E4) remove/ignore clusters/rules with *low utility*
  $$U_k^l = \frac{1}{k - I_k^l}\sum_{l=1}^{I_k^l}\lambda^l \qquad IF\left(U_k^l < \varepsilon_1\right) THEN\left(\lambda^l \leftarrow 0\right)$$
- F5) select on-line the input variables that contribute most to the output

Lecture 1 Review

- Lecture 1 – Introduction, Concept
  1. Motivation, Challenges
  2. System structure and its identification
  3. NN, Fuzzy Systems, NF Systems
  4. from Adapting to Evolving
  5. Basics of Evolving Systems
  6. Evolving Fuzzy Systems concept
  7. Data space partitioning -> ES structure
  8. Principles of evolution
- Next Lecture – Algorithms