

InfoLab21  
Department of Communication Systems

# Evolving Intelligent Systems

## Lecture 2 Algorithms

Dr. Flamen Angelov

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

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

1

InfoLab21  
Department of Communication Systems

## Lecture 2: Algorithms

1. Learning concept, types
2. Data Space partitioning, evolving Clustering
3. On-line input selection
4. Learning Algorithms (LMS, RLS)
5. Procedure
6. Evolving Classifiers - concept

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

2

InfoLab21  
Department of Communication Systems

## 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

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

3

InfoLab21  
Department of Communication Systems

## Learning methods

```

graph TD
    A[Learning methods] --> B[Supervised learning]
    A --> C[Reinforcement learning]
    A --> D[Unsupervised learning]
    B --> E[Parameter identification]
    D --> F[System structure rules identification, clustering]
    G[Learning methods] --> H[On-line]
    G --> I[Off-line]
  
```

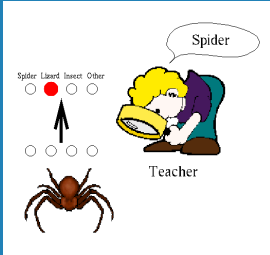
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid

4

InfoLab21  
Department of Communication Systems

## Supervised learning

- In this paradigm the learning algorithm is given a set of input/output pattern pairs.
- The model parameters/coefficients are adjusted so that the system will produce the required output in future.



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

5

InfoLab21  
Department of Communication Systems

## Supervised learning

- In this simple example the algorithm would be given a set of pictures of animals that are classified as 'spider', 'insect', 'lizard' or other.
- If the system is shown a spider, but classifies it as a lizard then the coefficients are adjusted to make the system respond "spider".
- A training set which consists of correct input-output data (target vector) is provided.
- If we denote by  $t$  the  $m$ -dimensional target then the error in prediction is:
 
$$J = \frac{1}{2} \sum_{i=1}^m (t_i - y_i)^2$$

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

6

InfoLab21  
Department of Communication Systems

## Reinforcement learning

- The system is provided with an evaluation of its output given the input and alters the coefficients to try to increase the reinforcement it receives.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 7

InfoLab21  
Department of Communication Systems

## Reinforcement learning

- Learning with **reinforcement (critic)** is similar to learning with a teacher (**supervised**)
- except that instead of being told the correct output given the input, only the fact that **it is** or **is not** correct is provided.
- The algorithm updates the coefficients to **maximize the # of inputs** on which it is correct
- In the very simple example the system/classifier is told that it is **incorrect** when it classifies the spider as a 'lizard' - but it is **NOT** told what the correct classification would be.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 8

InfoLab21  
Department of Communication Systems

## Reinforcement learning

- For this reason, learning with a critic is often more difficult and takes longer.
- Sometimes, however, we do not know what the correct output given an input should be and learning with a critic is **the only learning possible**
- Example of application: a system that controls a chemical plant in such a way that the temperature of the plant **did not exceed a certain bound**.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 9

InfoLab21  
Department of Communication Systems

## Reinforcement learning

- We may not know what settings of the valves should be used at a given time.
- If, however, the temperature rises above the bound we know that what the system just did was **NOT** good.
- That is, we supply **critic information**, but **NOT teacher information**, so a reinforcement algorithm could be used.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 10

InfoLab21  
Department of Communication Systems

## Un-supervised learning

- Unsupervised learning does **NOT** receive information from either a teacher or critic.
- Instead, it relies on an **internal criterion** to guide learning.

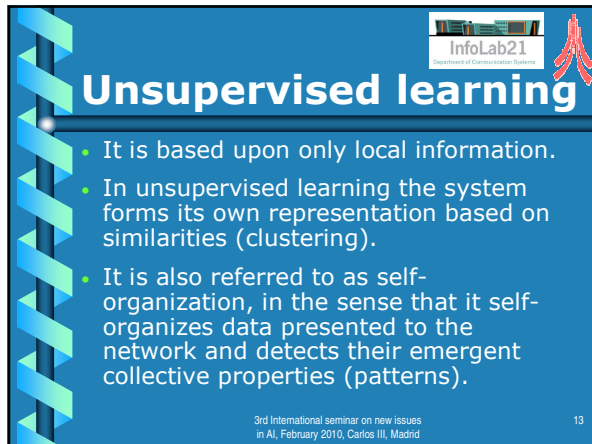
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 11

InfoLab21  
Department of Communication Systems

## Unsupervised Learning

- In **unsupervised** learning the system forms clusters (groupings, regions of data space) based on similarities.
- Systems that use **unsupervised** learning are single-layer NN such as SOM (self-organized maps, Kohonen, 1988), competitive learning etc.

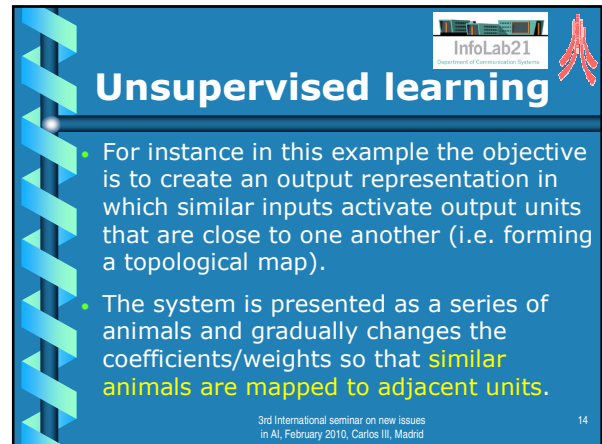
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 12



## Unsupervised learning

- It is based upon only local information.
- In unsupervised learning the system forms its own representation based on similarities (clustering).
- It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties (patterns).

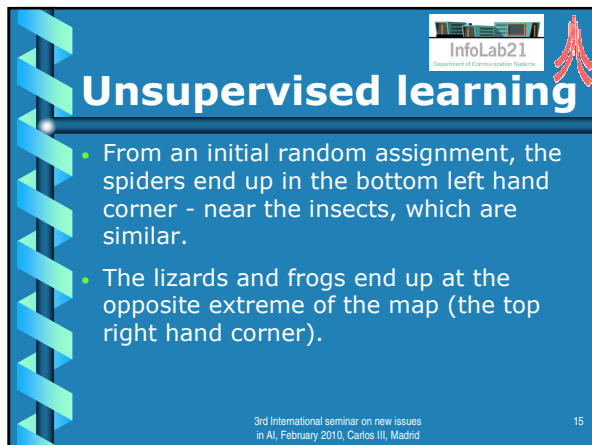
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 13



## Unsupervised learning

- For instance in this example the objective is to create an output representation in which similar inputs activate output units that are close to one another (i.e. forming a topological map).
- The system is presented as a series of animals and gradually changes the coefficients/weights so that **similar animals are mapped to adjacent units**.

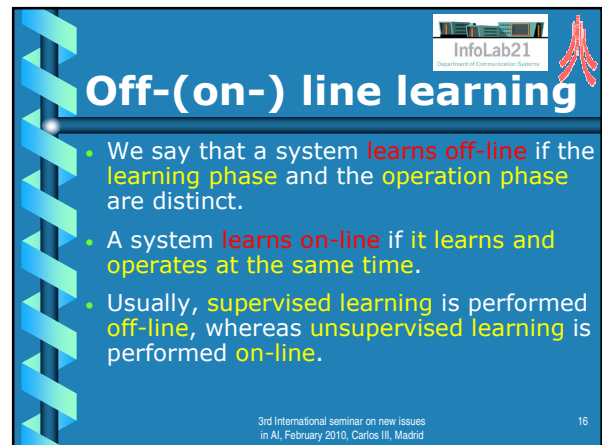
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 14



## Unsupervised learning

- From an initial random assignment, the spiders end up in the bottom left hand corner - near the insects, which are similar.
- The lizards and frogs end up at the opposite extreme of the map (the top right hand corner).

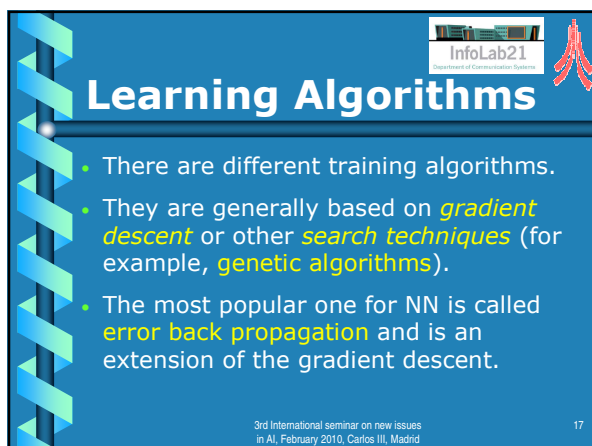
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 15



## Off-(on-) line learning

- We say that a system **learns off-line** if the **learning phase** and the **operation phase** are distinct.
- A system **learns on-line** if it **learns and operates at the same time**.
- Usually, **supervised learning** is performed **off-line**, whereas **unsupervised learning** is performed **on-line**.

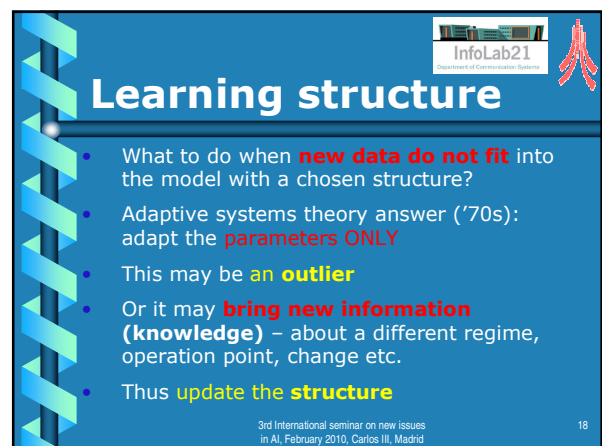
3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 16



## Learning Algorithms

- There are different training algorithms.
- They are generally based on **gradient descent** or other **search techniques** (for example, **genetic algorithms**).
- The most popular one for NN is called **error back propagation** and is an extension of the gradient descent.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 17



## Learning structure

- 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**

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 18

InfoLab21  
Department of Communication Systems

## TS fuzzy model (concept)

InfoLab21  
Department of Communication Systems

## Linear Models

- Consider the data for cars fuel consumption
- Each dot in the figure provides information about the weight in pounds and fuel consumption in *mpg* for one of 74 cars.
- Clearly, weight and fuel consumption are linked, so that, in general, heavier cars use more fuel.
- Source: UCI repository <http://www.ics.uci.edu/~lfd/uciarchive/UCRRepository.html>

InfoLab21  
Department of Communication Systems

## Linear Models

- Now, suppose we are given the weight of a 75<sup>th</sup> car, and asked to predict how much fuel it will use based on these data
- Such questions can be answered by using a **model** - a short mathematical description - of the data
- The simplest useful model here is of the form

$$y = w_1x + w_0$$

InfoLab21  
Department of Communication Systems

## Linear Models

- This is a **linear** model: in the 2-D, *xy*-plot
- This is the equation describes a straight line with slope *w1* and intercept *w0* with the *y*-axis

InfoLab21  
Department of Communication Systems

## The Error function

- The Error** function *E* is usually formulated as **sum-squared error** over the model parameters:

$$E = \frac{1}{2} \sum_{i=1}^N (t_i - y_i)^2$$

InfoLab21  
Department of Communication Systems

## System Structure

- Real-life** systems are **non-linear** but they are usually treated as locally or partially linear
- relatively easy to analyse and straightforward to implement and guaranteed **stable**.

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

InfoLab21  
Department of Communication Systems

## In a 2-D representation

Evolution of Clusters; · - data sample; \* - current data sample; o - focal point

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

25

InfoLab21  
Department of Communication Systems

## Density-based partitioning

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

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

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

$$P_A < P_B$$

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

26

InfoLab21  
Department of Communication Systems

## Density update

- Recursively calculated
- Euclidean:
 
$$P_k(z_k) = \frac{k-1}{(k-1)(1+\alpha_k) + \beta_k - 2\gamma_k}$$

$$a_i = \sum_{j=1}^n x_i^j f_j^i \quad b_i = b_{k-1} + a_{k-1} \quad b_1 = 0$$

$$c_i = \sum_{j=1}^n x_i^j f_j^i \quad f_k^j = f_{k-1}^j + x_{k-1}^j \quad f_1^j = 0$$
- Cosine (eClass)
 
$$a_i^k = a_{i,k-1} + \frac{(x_i^k)^2}{\sum_{i=1}^n (x_i^k)^2} \quad a_i^1 = \frac{(x_i^1)^2}{\sum_{i=1}^n (x_i^1)^2}; i = [1, n]$$

$$P_k(x_k) = \frac{1}{2 - \frac{1}{\sum_{i=1}^n (x_i^k)^2} \sum_{i=1}^n x_i^k a_{i,k-1}}$$

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

27

InfoLab21  
Department of Communication Systems

## Update density around focal points

- Recursively calculated
- Euclidean:
 
$$P_k(z^*) = \frac{(k-1)P_{k-1}(z^*)}{k-2 + P_{k-1}(z^*) + P_{k-1}(z^*) \sum_{j=1}^{p_{k-1}} \|z^* - z_{k-1}^j\|^2}$$
- Cosine (eClass):
 
$$P_k(x^*) = \frac{(k-1)P_{k-1}(x^*)}{k-1 + \left[ (k-2) \left( \frac{1}{P_{k-1}(x^*)} - 1 \right) + d_{\cos}(x^*, x_k) \right]}$$

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

28

InfoLab21  
Department of Communication Systems

## Manage cluster quality

- Adapt the radius:
 
$$(r_{jk}^l)^2 = \rho(r_{j(k-1)}^l)^2 + (1-\rho)(\sigma_{jk}^l)^2$$
- Using the local scatter:
 
$$(\sigma_{jk}^l)^2 = \frac{1}{S_k^l} \sum_{i=1}^{S_k^l} \|x^{i^k} - x_i^l\|_j^2$$
- $$(\sigma_{j0}^l)^2 = 1 \quad (\sigma_{j0}^{N+1})^2 = \frac{1}{N} \sum_{i=1}^N (\sigma_{jk}^l)^2$$

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

29

InfoLab21  
Department of Communication Systems

## Analyze cluster quality

- Support – No of samples per cluster
 
$$S^l \leftarrow S^l + 1 \quad \text{for } l = \operatorname{argmin}_i \|z_k - z^{i^k}\|$$
- Age:
 
$$\text{Age}_i^k(k) = k - \frac{\sum_{l=1}^{N_i(k)} I_l^k}{N_i(k)}; i = [1, R]$$
- $$I_k^l \leftarrow I_k^l + k \quad \text{for } l = \operatorname{argmin}_i \|z_k - z^{i^k}\|$$

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

30

InfoLab21  
Department of Communication Systems

## Detecting shift by cluster/rule age

**Shift** in the data stream can be detected by the **age** of the cluster/rule which corresponds to the **inflexed point of the Age curve** (when the derivative of Age changes its sign).

$\frac{d(Age)}{dk}$

Age analysis on Pepsyrene

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

31

InfoLab21  
Department of Communication Systems

## Cluster/rule utility

- The **utility** of a fuzzy rule represents the degree of support of a fuzzy rule.
- It is defined as the accumulated firing strength of the respective fuzzy rule for the span of its life

$$\eta_i(k) = \frac{\sum_{l=1}^k \lambda_l}{k - t_i} ; i = [1, R]$$

Utility analysis on Pepsyrene

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

32

InfoLab21  
Department of Communication Systems

## Granular System Structure

The high dimensional effect of the Clustering trick (analogy to SVM)

Similarity Measures:

$$\tau_1 = 1 / (x(k) - x_1^*)^2;$$

$$\tau_2 = \exp(- (x(k) - x_1^*)^2 / \sigma_1^2)$$

$$\tau_3 = \exp(-0.5(x - x_1^*)' C_{x1}^{-1}(x - x_1^*))$$

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

33

InfoLab21  
Department of Communication Systems

## On-line inputs selection

- Because in TS fuzzy systems the output is locally linear, the sensitivity analysis reduces to analysis of the consequent parameters:

$$\omega_j(k) = \frac{T_{ij}(k)}{\sum_{r=1}^n T_{ir}(k)} ; i = [1, R]; j = [1, n]$$

$$T_{ij}(k) = \sum_{l=1}^k |\theta_{ij}(l)|$$

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

34

InfoLab21  
Department of Communication Systems

## On-line inputs selection

- The importance of each input (feature) can be evaluated by the ratio of the accumulated sum of the consequent parameters for the specific  $j^{th}$  input (feature) in respect to **all**  $n$  inputs:

$$\exists j^* | \omega_{j^*}(k) < \varepsilon_i \sum_{r=1}^n T_{ir}(k); i = [1, R]; j = [1, n]$$

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

35

InfoLab21  
Department of Communication Systems

## On-line selection of input variables

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

36

InfoLab21  
Department of Communication Systems

## Advanced features

- ✓ Clustering the data space with monitoring and control of the cluster *age*;
- ✓ Identifying *drifts* in the data stream;
- ✓ Learning consequents parameters and using them for on-line input variables sensitivity analysis and selection (accumulated weights);
- ✓ Monitoring the *age* and *utility* of the fuzzy rules.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 37

InfoLab21  
Department of Communication Systems

## eClustering in the context

- The main element of the proposed approach is to estimate the *data density* in a *recursive* (suitable for on-line and real-time applications) manner from data streams and *to react* on the data density variations by *modifying the underlying structure* of the model.
- It is well known that the data density has been estimated off-line as a Gaussian by;
  - kernels in image processing;
  - Parzen windows;
  - GRNN
  - Mountain function
  - potential in subtractive clustering.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 38

InfoLab21  
Department of Communication Systems

## RDE

- We proposed (2003) to use Cauchy function:

$$e^{-\frac{\sum_{i=1}^k |z(i)-z(i)|}{2\sigma^2}} = \frac{1}{e^{\frac{\sum_{i=1}^k |z(i)-z(i)|}{2\sigma^2}}} \approx \frac{1}{1 + \sum_{i=1}^k \frac{|z(i)-z(i)|^2}{2\sigma^2} + \dots}$$

$$P(z(k)) = \frac{1}{1 + \frac{1}{k-1} \sum_{i=1}^{k-1} \frac{|z(k)-z(i)|^2}{2\sigma^2}}$$

- The rationale is that the points with high potential are good candidates for becoming focal points of fuzzy rules.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 39

InfoLab21  
Department of Communication Systems

## 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
 
$$P(z(k)) > \max_{i=1}^R P(z^i(k))$$
- 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
 
$$P(z(k)) < \min_{i=1}^R P(z^i(k))$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 40

InfoLab21  
Department of Communication Systems

## Principles of evolution

- B3) *avoid overlap* and information redundancy in forming new fuzzy rules.
 
$$\exists i, i = [1, R]; \mu_{ij}(x(k)) > e^{-1}; \forall j; j = [1, n]$$
- 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'

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 41

InfoLab21  
Department of Communication Systems

## Principles of evolution

- C4) remove/ignore *old* clusters/rules (with high *age*)
 
$$IF(A_i^k < meanA + std(A)) THEN(\lambda \leftarrow 0)$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 42

InfoLab21  
Department of Communication Systems

## Principles of evolution

- D4) remove/ignore clusters with **low support**  

$$\text{IF } (S_k^i < n(n-1)) \text{ THEN } (\lambda^i \leftarrow 0)$$
- E4) remove/ignore clusters/rules with **low utility**  

$$U_k^i = \frac{1}{k-1} \sum_{l=1}^k \lambda^l$$
- F5) **select on-line the input variables** that contribute most to the output  

$$\text{IF } (U_k^i < \varepsilon_i) \text{ THEN } (\lambda^i \leftarrow 0)$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 43

InfoLab21  
Department of Communication Systems

## Procedure eClustering

Read data sample  $z(k)=[x(k);y(k)]$   
 IF  $(k=1)$  THEN  
 $d_j(k)=0; j=[1, n]; b(k)=0$   
 $x_1^*(1) \leftarrow x(1) \quad P(z_1^*(k)) \leftarrow 1; R \leftarrow 1$   
 Rule: IF  $(x_1 \text{ is } x_1^*) \text{ AND...AND } (x_n \text{ is } x_n^*)$   
 ELSE  
 Recursively calculate density of the current data sample;  
 Update the membership functions of the respective fuzzy sets;  
 Recursively update the density at the existing cluster centers;  
 Check  
 Condition A (P1 OR P2);  
 Conditions B-E (P3, P4, P5)  
 Assign the new point to the nearest cluster.  
 Repeat until end of data stream

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 44

InfoLab21  
Department of Communication Systems

## Procedure eClustering

- Form new rule:
  - Generalisation power/representative  
 IF  $P_k(x_k) > \max P_k(x^{N+1})$  THEN  $x^{(N+1)} \leftarrow x_k$
  - Good coverage/completeness  
 IF  $P_k(x_k) < \min P_k(x^{N+1})$  THEN  $x^{(N+1)} \leftarrow x_k$
- Remove a rule:
  - IF  $\exists i, i=[1, N]; \mu_j^i > e^{-1} \quad \forall j, j=[1, n]$  THEN  $x^{N+1} \rightarrow out$
  - IF  $(Age > \text{mean}(Age) + \text{std}(Age))$  THEN  $x^{N+1} \rightarrow out$
  - IF  $(Utility \text{ of the rule is low})$  THEN  $x^{N+1} \rightarrow out$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 45

InfoLab21  
Department of Communication Systems

## Extracting rules on-line

- As a result of this data density-based on-line clustering procedure the following fuzzy rule base formed of antecedent parts is generated from the data stream:
 
$$R_i \text{ IF } (x_1 \text{ is } x_{1i}^*) \text{ AND } (x_2 \text{ is } x_{2i}^*) \dots (x_n \text{ is } x_{ni}^*); i=[1, R]$$
- It can then be either;
  - stored and later analyzed by an operator;
  - used for prediction at each time step if combined with consequent part identification;
  - used for classification if combined with consequent part identification;
  - used for clustering the data and various applications

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 46

InfoLab21  
Department of Communication Systems

## eTS

$R^i : \text{IF } (x_1 \text{ is close to } x_{1i}^*) \text{ AND...AND } (x_n \text{ is close to } x_{ni}^*) \text{ THEN } (y^i = f^i)$

- MIMO
- TS  $f^i = x^{T_i} \pi^i + sM$  (0 order TS)  $f^i = a^i$

$$\pi^i = \begin{bmatrix} \alpha'_{01} & \alpha'_{02} & \dots & \alpha'_{0m} \\ \alpha'_{11} & \alpha'_{12} & \dots & \alpha'_{1m} \\ \dots & \dots & \dots & \dots \\ \alpha'_{n1} & \alpha'_{n2} & \dots & \alpha'_{nm} \end{bmatrix} \quad a^i = [\alpha'_{01} \quad \alpha'_{02} \quad \dots \quad \alpha'_{0m}]$$

$$y = \sum_{i=1}^N \frac{\prod_{j=1}^n \mu_j^i(x_j)}{\sum_{j=1}^N \prod_{j=1}^n \mu_j^i(x_j)} y^i$$

$$\mu_j^i = e^{-\frac{\|x - x_{j^i}\|^2}{2(\sigma_j^i)^2}}$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 47

InfoLab21  
Department of Communication Systems

## On-line learning

- Two stages:
  - data partitioning by RDE and eClustering;
  - fuzzily weighted RLS of conseq. params.
- Specifics/advantages
  - It has an **evolving** (open, flexible) structure and can start from scratch;
  - very low memory is required for the calculations;
  - suitable for real-time applications;
  - high prediction rates;
  - It has MIMO structure;
  - Can automatically detect **shifts** in the data pattern that reflect different operating regimes.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 48



InfoLab21  
Department of Communication Systems

## Learning – min E

- In learning from data they try to minimise the error (the difference between the expected/predicted value and the real value).
- Learning works in a **close loop** taking a feedback from the error.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 49

InfoLab21  
Department of Communication Systems

## Learning - adapting

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 50

InfoLab21  
Department of Communication Systems

## Learning aim

- Learning aims to minimise distortion (error) **at each time instant**:  

$$e(n) = x(n) - \hat{x}(n)$$
- The error,  $e(n)$  between the system output and the target value is fed back to adjust the coefficients,  $f_n(j)$ ;  
 $j=0, 1, \dots, N-1$
- NOTE: coefficients are **time-varying**

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 51

InfoLab21  
Department of Communication Systems

## Performance surface

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 52

InfoLab21  
Department of Communication Systems

## Iterative solution

- The iterations begin with an **initial estimate** or **guess** (starting point on the  $J$  surface).
- coefficients,  $f(1), f(2), \dots$  are then successively updated:  

$$f(n) = f(n-1) - \alpha(n)p(n)$$
- where  $\alpha(n)$  is a **step-size** parameter  
 $p(n)$  is the search direction vector

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 53

InfoLab21  
Department of Communication Systems

## Steepest Gradient Descent

- The gradient of  $J$  points the direction of the biggest increase of the error function.
- We want to minimise it, therefore we move in the opposite direction:  

$$\nabla J = 2(R_{yy}f - R_{yx})$$

$$f(n) = f(n-1) - \alpha(n) \nabla J(n)$$
- Combining, we get:  

$$f(n) = f(n-1) - 2\alpha(n)(R_{yy}f - R_{yx})$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 54

InfoLab21  
Department of Communication Systems

## Steepest Gradient Descent

- In the neighbourhood of  $f$ , the  $\nabla$  is **normal** (perpendicular) to lines of equal error  $J$ .
- Thus, the gradient direction is the line of steepest ascent in error terms.
- The algorithm has a natural tendency to take large steps when  $f$  is far from  $f^*$ , and smaller steps as  $f$  approaches  $f^*$  and  $\nabla J$  decreases.
- The process eventually converges to the optimal solution.

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 55

InfoLab21  
Department of Communication Systems

## Wiener-Hopf algorithm

- assuming an **estimate** of the gradient instead the exact Wiener-Hopf solution to the normal equations we get:

$$f(n) = f(n-1) - \alpha(n) \hat{\nabla} J(n)$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 56

InfoLab21  
Department of Communication Systems

## Widrow-Hopf algorithm

- The **LMS** algorithm **estimates**  $\nabla J$  via the  $\nabla$  of the **instantaneous** value of the squared error:

$$\hat{\nabla} J(n-1) = -2e(n)y(n)$$

$$e(n) = x(n) - f^T(n-1)y(n)$$

- and the **LMS** adaptive Filter update is:

$$f(n) = f(n-1) + 2\alpha(n)y(n)e(n)$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 57

InfoLab21  
Department of Communication Systems

## LMS procedure

- 1) Initialise by  $f(0)=0$
- 2) Do for  $n=1,2,\dots$ 
  - a)  $\hat{x}(n) = f^T(n-1)y(n)$
  - b)  $e(n) = x(n) - \hat{x}(n)$
  - c)  $f(n) = f(n-1) + 2\alpha(n)y(n)e(n)$
- 3) Increment  $n$  and continue from step 2)

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 58

InfoLab21  
Department of Communication Systems

## RLS Procedure

- 1) Initialise
  - filter coefficients  $f(0)=0$
  - the estimate of  $R_{yy}^{-1}(n)$
 where  $\delta$  is a small number;  
 $I_N$  is the identity matrix
 
$$I_N = \begin{bmatrix} 1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 1 \end{bmatrix}$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 59

InfoLab21  
Department of Communication Systems

## RLS Procedure

- 2) For the next time-steps  $n=1,2,\dots$  DO:
  - a. Find the estimation of the signal using the previous filter coefficients:
 
$$\hat{x}(n) = f^T(n-1)y(n)$$
  - b. The *a priori* estimate of the error:
 
$$e(n) = x(n) - \hat{x}(n)$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 60

InfoLab21  
Department of Communication Systems

## RLS Procedure

- c. Update the inverse auto-correlation matrix:
 
$$R_{yy}^{-1}(n) = R_{yy}^{-1}(n-1) - \frac{R_{yy}^{-1}(n-1)y(n)y^T(n)R_{yy}^{-1}(n-1)}{1 + y^T(n)R_{yy}^{-1}(n-1)y(n)}$$
- d. Update the filter coefficients:
 
$$f(n) = f(n-1) + R_{yy}^{-1}(n)y(n)e(n)$$
- e. Increment  $n$  and continue from step a.

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

InfoLab21  
Department of Communication Systems

## RLS vs. LMS

- Recursive methods are preferred when we wish to improve our estimate of the parameters using the new data
- Computationally RLS is **more expensive** than LMS and it is **more sensitive to errors (less robust)** due to
  - Its recursive nature
  - And in the case when the signal  $x$  has value zero for long time.

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

InfoLab21  
Department of Communication Systems

## RLS vs. LMS convergence

☑ Its **convergence is superior** (converges faster and to lower values of the error).

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

InfoLab21  
Department of Communication Systems

## Learning

$$y = \Psi^T \theta \quad \theta = [\pi^1, \pi^2, \dots, \pi^N]^T$$

$$\Psi = [\lambda^1 x_e^T, \lambda^2 x_e^T, \dots, \lambda^N x_e^T]^T$$

Identification criteria:

$$(y - \Psi^T \theta)^T (y - \Psi^T \theta) \rightarrow \min$$

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

InfoLab21  
Department of Communication Systems

## On-line learning consequent parameters

- Globally optimal (more precise)
  - RLS (kalman filter)
 
$$\theta_k = \theta_{k-1} + C_k \Psi_k (y_k - \Psi_k^T \theta_{k-1}) \quad \theta_0 = 0$$

$$C_k = C_{k-1} \frac{C_{k-1} \Psi_k^T \Psi_k C_{k-1}}{1 + \Psi_k^T C_{k-1} \Psi_k} \quad C_0 = \Omega$$
- Locally optimal (locally meaningful models)
  - wRLS
 
$$\pi_{ik} = \pi_{ik-1} + c_{ik} x_{ek} \lambda_i(x_k) (y_k - x_{ek}^T \pi_{ik-1})$$

$$c_{ik} = c_{ik-1} \frac{\lambda_i(x_k) c_{ik-1} x_{ek} x_{ek}^T c_{ik-1}}{1 + \lambda_i(x_k) x_{ek}^T c_{ik-1} x_{ek}}$$

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

InfoLab21  
Department of Communication Systems

## eTS Fuzzy Models (parameter evolution)

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

InfoLab21  
Department of Communication Systems

## eTS vs ANFIS

- The main difference between the ANFIS and eTS is the fact that the number of neurons at each layer of ANFIS as well as the number of fuzzy rules is **pre-determined** and **fixed**
- while in eTS a new fuzzy rule (and the respective neurons in each layer) is added if the new data is representative enough (very high P) or brings new info (coverage of the data space, very low P).

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 67

InfoLab21  
Department of Communication Systems

## Basic procedure

1. First data - first rule center
2. Collect **new data in real-time**
3. Calculate  $P^{new}$
4. Recursively up-date  $P^*$
5. Up-grade or modify the Rule-base
6. Estimate Consequent parameters
7. Form the final output

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 68

InfoLab21  
Department of Communication Systems

## Flow-Chart

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 69

InfoLab21  
Department of Communication Systems

## Rule-base Evolution

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 70

InfoLab21  
Department of Communication Systems

## Linear classifiers

- Let us have a task to build a classifier to group the following data into two classes:

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

InfoLab21  
Department of Communication Systems

## Non-linear classifier

- A more realistic distribution of the data will look like this:
- Let us have a task to build a classifier to group the following data into two classes:

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

InfoLab21  
Department of Communication Systems

## Non-linear classifier

- Obviously, if we use a **non-linear classifier** a much better result (a lower level of the error) can be obtained.
- The question is **what sort of structure** this function should have.

InfoLab21  
Department of Communication Systems

## Non-linear classifier 3D

IRIS Dataset – Surfaces of membership to three classes

InfoLab21  
Department of Communication Systems

## Non-linear classifier

- In the case of a *linear* function the problem is to find the **parameters** of this classifier.
- If the function is *non-linear* we have to specify its structure as well.

InfoLab21  
Department of Communication Systems

## Adaptive classifiers

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 76

InfoLab21  
Department of Communication Systems

## eClass

77

InfoLab21  
Department of Communication Systems

## Granular System Structure

$$y = f(x) \quad \text{encoding}$$

$$\tau_i = h_i(x(k)) \rightarrow y = \sum_i h_i w_i \quad (\text{RBF NN})$$

$$v_i = \tau_i / (\sum_j \tau_j) \rightarrow y = \sum_i v_i y_i^* \quad (\text{Fuzzy})$$

$$v_i = \tau_i / (\sum_j \tau_j) \rightarrow y = \sum_i v_i (a_{i0} + a_i^T x) \quad (\text{Takagi-Sugeno})$$

$$\psi = [h_1(x(k)), \dots, h_m(x(k))]^T \quad \theta = [w_1, \dots, w_m] \quad y = \psi^T \theta$$

3rd International seminar on new issues in AI, February 2010, Carlos III, Madrid 78

## Parametric Model Learning

$$y_k = \psi_k^T \theta_{k-1}$$

$$\theta_k = \theta_{k-1} + \alpha \psi_k (y_k - \psi_k^T \theta_{k-1}) \quad \text{LMS}$$

$$\theta_k = \theta_{k-1} + (\beta / \psi_k^T \psi_k) \psi_k (y_k - \psi_k^T \theta_{k-1}) \quad \text{Widrow-Hoff}$$

$$\begin{aligned} \theta_k &= \theta_{k-1} + C_k \psi_k (y_k - \psi_k^T \theta_{k-1}) & \theta_0 &= 0 & \text{RLS,} \\ C_k &= C_{k-1} - \frac{C_{k-1} \psi_k \psi_k^T C_{k-1}}{1 + \psi_k^T C_{k-1} \psi_k} & C_0 &= \Omega I & \text{Kalman Filter} \end{aligned}$$

$$\theta_k = \theta_{k-1} + R_k H_k \psi_k (y_k - \psi_k^T \theta_{k-1}) \quad \text{Kohonen SOM}$$

## Lecture 2 Review

1. Learning concept, types
2. Data Space partitioning, evolving Clustering
3. On-line input selection
4. Learning Algorithms (LMS, RLS)
5. Procedure
6. Evolving Classifiers - concept