An Integrated Approach to Computational Intelligence

Satish Kumar

Professor
Department of Physics & Computer Science
Coordinator, Multimedia Facility
Dayalbagh Educational Institute
Dayalbagh, Agra, India

skumar_db@ieee.org
Seminar Outline

- Brief review of computational intelligence, neural nets and fuzzy systems
- Subsethood neuro-fuzzy model
- Applications
- Evolutionary learning based subsethood model
- Parallel implementation + Applications
Inspiration for Increasing Machine IQ (MIQ)

Two fundamental questions...

- How does the human brain work?
- How can we exploit whatever little we understand of how the human brain computes to build intelligent machines?
Computing Models for MIQ Applications

**Hard Computing**
- Precise Models
  - Symbolic Logic Reasoning (Traditional AI)
  - Traditional Numerical Modeling and Search

**Soft Computing**
- Approximate Models
  - Approximate Reasoning
  - Functional Approximation and Randomized Search

Additional Models:
- Probabilistic Models
- Multivalued & Fuzzy Logics
- Neural Networks
- Evolutionary Computing
Computational Intelligence

- Neural Networks
  - Learning
  - Generalization
- Fuzzy Logic
  - Linguistic information
  - Approximate reasoning
- Evolutionary Algorithms
  - Global search

Hybrids

- Can be augmented with expert knowledge
- Can mimic one or more aspects of carbon-based biological intelligence: the wet stuff
Typical Multilayered Neural Network Architecture
Artificial Neuron
Neural Network Architectures

\[ \mathbf{X} \subset \mathbb{R}^n \]

\[ \mathbf{S} = f(\mathbf{X}) \]

Feedforward

Feedback
Salient Properties of Neural Networks

- Robust and fault tolerant
  - Ability to operate (with some performance loss), in the event of damage to internal structure, and incomplete input data

- Associative recall
  - Ability to invoke related memories from one concept

- Function approximation and generalization
  - Ability to approximate functions by creating internal representations
  - Predict on unseen inputs
  - No mathematical model required

- Adaptation
  - Incorporate powerful learning mechanisms
Humans reason with fuzzy concepts  (Bezdek, 1993)

- Advice to driving student:
  - Begin braking 74 feet from the crosswalk
  - Apply the brakes *pretty soon*

- Children quickly learn to interpret:
  - You must be in bed *around 9 pm*

- We easily...
  - Assimilate imprecise information and vague rules
  - Reason effortlessly using a *fuzzy logic*
Fuzzy Concept: Numbers Close to 4

\[ \mu(X): X \rightarrow [0,1] \]

This mapping is flexible!
Fuzzy Rules in the Real World

- If traffic density is HIGH, then keep green light on LONGER
- If X1 is MEDIUM and X2 is SMALL then Y is LARGE
Fuzzy Systems

1. Fuzzification Interface
2. Rule Base
3. Inference Engine
4. Defuzzifier
The Benefit of Integration

self-organizing substrates based on on-line adaptation numeric processing

linguistic variables method to handle uncertainty and imprecision model expert heuristics
Neuro Fuzzy Modelling Framework

- **X1=0.9**
- **X2=26**
- **X3=LOW**
- **X4=YES**

**Neuro Fuzzy Inference System**

- **Gradient/Evolutionary Learning algorithm**
- **Initialize**
- **Extraction algorithm**

**Numeric database**

**Linguistic Rule base**

**Inference**
Making the Connections Fuzzy: Embedding a Simple Rule

\[ R = \text{If } X1 \text{ is MEDIUM and } X2 \text{ is SMALL then } Y \text{ is LARGE} \]
Asymmetric Subsethood Product Fuzzy Neural Inference System

ASuPFuNIS - Architecture

**Input Layer**
- $x_1 \rightarrow$ (Linguistic)
- $x_i \rightarrow$ (Linguistic)
- $x_m \rightarrow$ (Linguistic)
- $x_{m+1} \rightarrow$ (Numeric)
- $x_n \rightarrow$ (Numeric)

**Rule Layer**
- Antecedent weights $\left(c_{ij}, \sigma_{ij}^l, \sigma_{ij}^r\right)$
- Consequent weights $\left(c_{jk}, \sigma_{jk}^l, \sigma_{jk}^r\right)$

**Output Layer**
- $y_1$
- $y_k$
- $y_p$
Mixed Inputs

Numeric
1.3

Linguistic
HIGH

Membership functions have tunable left and right spreads

Input

Neural Networks Laboratory
Dayalbagh Educational Institute

CMSC 818Z University of Maryland: 15/03/07
Activity or Evidence Aggregation

1.3 Numeric Input

Node: Fuzzifier

Mutual Subsethood

E1 E2

Mutual Subsethood

Z1

Rule Node

Output Node

Numeric input

Linguistic input

Linguistic Input node
Mutual Subsethood Measure

Fuzzified Numeric Input or Fuzzified Input

Fuzzy Antecedent Weight

Mutual Subsethood

\[ E(A,B) = \frac{C(A \cap B)}{C(A) + C(B) - C(A \cap B)} \]

\( C(\bullet) \) is the cardinality of the set defined by

\[
C(A) = \int_{-\infty}^{c} e^{-\left(\frac{x-c}{\sigma^l}\right)^2} dx + \int_{c}^{\infty} e^{-\left(\frac{x-c}{\sigma^r}\right)^2} dx
\]
Mutual Subsethood: Overlap Cases
Activity or Evidence Aggregation

1.3

Numeric input

Forms of Mutual Subsethood

E1

E2

Z1

Rule Node

Output Node

Linguistic input

Linguistic Input node
Activity Aggregation

Product

\[ \prod (A, B) = AB \]

\[ Z_j = \prod_{i=1}^{n} E_{ij} \]
Defuzzification

1.3 Numeric input
Input Fuzzifier

Linguistic input
Linguistic Input node

Mutual Subsethood

E1

E2

Rule Node

Output Node

Z1

Z2

Z3

Mutual Subsethood
Activity Calculation at Output Layer

Defuzzification is done by each node in output layer using area based defuzzification

\[
y_k = \frac{\sum_{j=1}^{q} \nu_j \mu_j c_j}{\sum_{j=1}^{q} \nu_j \mu_j} = \frac{\sum_{j=1}^{q} Z_j \left( c_{jk} + \frac{\sigma^r_{jk} - \sigma^l_{jk}}{\sqrt{\pi}} \right) \left( \sigma^l_{jk} + \sigma^r_{jk} \right)}{\sum_{j=1}^{q} Z_j \left( \sigma^l_{jk} + \sigma^r_{jk} \right)}
\]

The output signal of output node \( k \) is \( S(y_k) = y_k \)
Early Approach: Supervised Gradient Descent Learning

The performance of the output is evaluated by a standard square error function which is given as

$$e = \frac{1}{2} \sum_{k=1}^{p} (d_k - y_k)^2$$

Iterative gradient descent update equation:

$$P(t + 1) = P(t) - \eta \nabla e(t) + a \Delta P(t - 1)$$

where

$$P = \left( x_i^{\sigma}, x_i^{\sigma}, w_{ij}^{c}, w_{ij}^{\sigma}, w_{ij}^{\sigma}, v_{jk}^{c}, v_{jk}^{\sigma}, v_{jk}^{\sigma} \right)$$
Summary of ASuPFuNIS Features

- Handles numeric and linguistic inputs simultaneously
- Composition of inputs and fuzzy weights employs a mutual subsethood measure
- Aggregates activities at rule node using a product operator
- Output is generated using area defuzzification
- Gradient descent employed to train the network
ASuPFuNIS is a Universal Function Approximator

Let \( \phi(\cdot) \) be a non-constant, bounded, and monotone-increasing continuous function. Let \( \mathbb{I}^n \) denote the \( n \)-dimensional unit hypercube \([0, 1]^n\), and the space of continuous functions on \( \mathbb{I}^n \) be denoted by \( C(\mathbb{I}^n) \). Then given any function \( F \in C(\mathbb{I}^n) \) and \( \epsilon > 0 \), there exists an integer \( p \) and sets of real constants \( \alpha_j, \theta_j, \) and \( w_{ij} \), where \( i = 1, \ldots , n \) and \( j = 1, \ldots , p \) such that we may define

\[
f(X, W) = \sum_{j=1}^{p} \alpha_j \phi\left( \sum_{i=1}^{n} w_{ij} x_i - \theta_j \right) \quad X \in \mathbb{I}^n, \ W \in \mathbb{R}^{n \times p} \tag{6.61}
\]

as an approximate realization of the function \( F(X) \) where

\[
|f(X, W) - F(X)| < \epsilon \tag{6.62}
\]

for all \( X \in \mathbb{I}^n \).
Application Domains

- Medical Diagnosis (Hepatitis data)
- Pattern Classification (Iris data, Telugu vowel classification, Ripley synthetic data)
- Function Approximation (Narazaki-Ralescu function, Hang)
- Time Series Prediction (Chaotic Mackey-Glass series)
- Control (Truck backer-upper)
Hepatitis Diagnosis
http://www.ics.uci.edu/~mlearn/MLRepository.html

Problem Statement:

Hepatitis diagnosis requires classifying patient into two classes DIE or LIVE on the basis of 19 features.

- 6 numeric features
- 13 linguistic features

ASuPFuNIS

DIE

LIVE
## Details of Hepatitis Data Features

<table>
<thead>
<tr>
<th>#</th>
<th>Attribute</th>
<th>Type</th>
<th>Range/Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Age</td>
<td>Numeric</td>
<td>20-70</td>
</tr>
<tr>
<td>2</td>
<td>Sex</td>
<td>Linguistic</td>
<td>M/F</td>
</tr>
<tr>
<td>3</td>
<td>Steroid</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>4</td>
<td>Antivirals</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>5</td>
<td>Fatigue</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>6</td>
<td>Malaise</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>7</td>
<td>Anorexia</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>8</td>
<td>Liver Big</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>9</td>
<td>Liver Firm</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>10</td>
<td>Spleen Palpable</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>11</td>
<td>Spiders</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>12</td>
<td>Ascites</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>13</td>
<td>Varices</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
<tr>
<td>14</td>
<td>Bilirubin</td>
<td>Numeric</td>
<td>0.3-4.8</td>
</tr>
<tr>
<td>15</td>
<td>Alk. Phos.</td>
<td>Numeric</td>
<td>26-280</td>
</tr>
<tr>
<td>16</td>
<td>SGOT</td>
<td>Numeric</td>
<td>14-420</td>
</tr>
<tr>
<td>17</td>
<td>Albumin</td>
<td>Numeric</td>
<td>2.1-5</td>
</tr>
<tr>
<td>18</td>
<td>Protime</td>
<td>Numeric</td>
<td>0-100</td>
</tr>
<tr>
<td>19</td>
<td>Histology</td>
<td>Linguistic</td>
<td>Y/N</td>
</tr>
</tbody>
</table>
Experimental Setup and Model for Linguistic Features

- 19-q-2 ASuPFuNIS architecture
- Centers of antecedent and consequent fuzzy weights initialized in the range (0,1)
- Spreads initialized in the range (0.2, 0.9)
- Learning rate and momentum: 0.0001

- The linguistic features are represented by two fuzzy sets. ‘no’ by an asymmetric Gaussian centered at 0 and ‘yes’ by an asymmetric Gaussian centered at 1 with tunable spreads
Hepatitis Data

- 135 patterns
- Five sets each of 70% training-30% test patterns randomly selected
# Hepatitis Diagnosis Results

<table>
<thead>
<tr>
<th>Methods</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASuPFuNIS (2 rules / hep 135)</td>
<td>Avg: 97.5 Best: 100</td>
</tr>
<tr>
<td>Wang-Tseng approach</td>
<td>91.61</td>
</tr>
<tr>
<td>k-NN, k=18</td>
<td>90.2</td>
</tr>
<tr>
<td>Bayes</td>
<td>84.0</td>
</tr>
<tr>
<td>LVQ</td>
<td>83.2</td>
</tr>
<tr>
<td>Assistant-86</td>
<td>83.0</td>
</tr>
<tr>
<td>CN2</td>
<td>80.1</td>
</tr>
</tbody>
</table>
Telugu Vowel Classification
(http://www.isical.ac.in/~sushmita/patterns)

- 6-Class data set comprising 871 Telugu vowel sounds
  - Six vowel classes are \( \partial, a, i, u, e, \) and \( o \)
- Spoken by three male speakers
- Three input features
  - Corresponding to the first, second and third formant frequencies obtained through spectrum analysis of speech data
Telugu Vowel Data Plot
Data Preprocessing

- Vowel data linearly normalized feature-wise in the range (0,1)
- Training set: 87 patterns (randomly choosing 10% of samples of each class)
- Test set: 784 patterns (remaining 90% of patterns)
- Simulation experiments performed on 12 such separate vowel data sets
Experimental Setup

- 3-q-6 ASuPFuNIS architecture
- Centers of fuzzy weights initialized in the range (0,1)
- Spreads initialized in the range (0.2, 0.9)
- Learning rate, momentum: 0.0001
- Percent correct recognition rate used as the performance measure of the network
## Performance Comparison

<table>
<thead>
<tr>
<th>Class → Model ↓</th>
<th># of free parameters</th>
<th>$\partial$</th>
<th>a</th>
<th>i</th>
<th>u</th>
<th>e</th>
<th>o</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pal and Mitra Model (A)†</td>
<td>326</td>
<td>44.6</td>
<td>65.4</td>
<td>79.2</td>
<td>88.3</td>
<td>75.0</td>
<td>85.7</td>
<td>76.8</td>
</tr>
<tr>
<td>Pal and Mitra Model (B)†</td>
<td>358</td>
<td>69.8</td>
<td>72.8</td>
<td>81.8</td>
<td>85.9</td>
<td>75.0</td>
<td>87.2</td>
<td>80.1</td>
</tr>
<tr>
<td>Pal and Mitra Model (C)†</td>
<td>386</td>
<td>51.2</td>
<td>84.4</td>
<td>81.9</td>
<td>86.8</td>
<td>92.4</td>
<td>89.5</td>
<td>84.2</td>
</tr>
<tr>
<td>SuPFuNIS (6 rules) [225] SuPFuNIS (10 rules)</td>
<td>111</td>
<td>60.00</td>
<td>77.50</td>
<td>85.42</td>
<td>69.12</td>
<td>81.72</td>
<td>96.30</td>
<td>80.87</td>
</tr>
<tr>
<td>ASuPFuNIS (3 rules)</td>
<td>87</td>
<td>43.08</td>
<td>80.00</td>
<td>86.45</td>
<td>85.29</td>
<td>87.10</td>
<td>94.44</td>
<td>83.80</td>
</tr>
<tr>
<td>ASuPFuNIS (4 rules)</td>
<td>114</td>
<td>32.31</td>
<td>92.50</td>
<td>87.74</td>
<td>91.18</td>
<td>87.10</td>
<td>88.27</td>
<td>84.18</td>
</tr>
<tr>
<td>ASuPFuNIS (5 rules)</td>
<td>141</td>
<td>35.38</td>
<td>88.75</td>
<td>87.74</td>
<td>88.97</td>
<td>90.32</td>
<td>87.65</td>
<td>84.31</td>
</tr>
</tbody>
</table>
Truck Backer Upper Control Problem

- Backing up a truck to loading dock
  - Loading dock
    - $x = 10 \; \varphi = 90$

$\Phi$ : Angle of the truck with the horizontal
$x, y$ : coordinates in the space
$\Theta$ : steering angle
$b$ : length of truck

$x(t + 1) = x(t) + \cos[\phi(t) + \theta(t)] + \sin[\theta(t)]\sin[\phi(t)]$

$y(t + 1) = y(t) + \sin[\phi(t) + \theta(t)] - \sin[\theta(t)]\cos[\phi(t)]$

$\phi(t + 1) = \phi(t) - \sin^{-1}\left[\frac{2\sin(\theta(t))}{b}\right]$
Design of Truck Backer Upper Control

- y coordinate ignored
- Range of x: 0 to 20
- Range of Φ: -90° to 270°
- Range of θ: -40° to +40°

Control value of θ such that the final state \((x_f, \Phi_f) = (10, 90°)\)
Data Set and Parameter Details

- Numeric data comprises of 238 pairs accumulated from 14 sequences of desired \((x, \Phi, \theta)\) values (Wang/Mendel, 1992)
- The 14 initial states \((x, \Phi, \theta)\) are \((1,0), (1,90), (1,270), (7,0), (7,90), (7,180), (7,270), (13,0), (13,90), (13,180),(13,270), (19,90), (19,180), (19,270)\)
- The data was linearly normalized in the range \([0,1]\)
- The free parameters for this application are \(6r+2\)
- Three initial states, \((x, \Phi, \theta) = (3,30), (10,220), \) and \((13,30)\) were used to test the performance of the controller
Truck Backer Upper Trajectories

3 rules

5 rules
Performance Measure

The performance of the controller is considered good if a proper balance is maintained between the type of trajectory and trajectory destination.

- **Normalized Docking Error (NDE)**
- **Trajectory Error (TE)**

\[
NDE = \sqrt{\left(\frac{\varphi_f - \varphi_d}{360}\right)^2 + \left(\frac{x_f - x_d}{20}\right)^2}
\]

\[
TE = \frac{\text{length of truck trajectory}}{\text{distance(initial position, desired final position)}}
\]
# Docking and Trajectory Errors

<table>
<thead>
<tr>
<th>Initial point ((x, y, \phi^\circ))</th>
<th>Normalized Error</th>
<th>Trajectory Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 rules</td>
<td>3 rules</td>
</tr>
<tr>
<td>((2, 2, 90^\circ))</td>
<td>0.0087</td>
<td>0.0085</td>
</tr>
<tr>
<td>((3, 3, -30^\circ))</td>
<td>0.0064</td>
<td>0.0061</td>
</tr>
<tr>
<td>((5, 3, 180^\circ))</td>
<td>0.0270</td>
<td>0.0255</td>
</tr>
<tr>
<td>((8, 4, -45^\circ))</td>
<td>0.0066</td>
<td>0.0041</td>
</tr>
<tr>
<td>((9, 2, -60^\circ))</td>
<td>0.0017</td>
<td>0.0021</td>
</tr>
<tr>
<td>((10, 4, 220^\circ))</td>
<td>0.0100</td>
<td>0.0094</td>
</tr>
<tr>
<td>((10, 10, 360^\circ))</td>
<td>0.0107</td>
<td>0.0090</td>
</tr>
<tr>
<td>((11, 4, 270^\circ))</td>
<td>0.0080</td>
<td>0.0075</td>
</tr>
<tr>
<td>((13, 3, 30^\circ))</td>
<td>0.0053</td>
<td>0.0047</td>
</tr>
<tr>
<td>((15, 7, 280^\circ))</td>
<td>0.0066</td>
<td>0.0055</td>
</tr>
<tr>
<td>((16, 3, 260^\circ))</td>
<td>0.0054</td>
<td>0.0051</td>
</tr>
<tr>
<td>((18, 2, 50^\circ))</td>
<td>0.0134</td>
<td>1.0814</td>
</tr>
</tbody>
</table>
Augmentation of SuPFuNIS with Expert Linguistic Knowledge

- Training done using reduced set (42 data pairs) considering only first three pairs of data from each of the 14 sequences
- Finer control is done using the linguistic rules constructed from expert knowledge
Truck Backer Upper Trajectories

(a) 5 rules (reduced numeric data)

(b) 5 rules (reduced numeric data) + 5 expert rules
Truck Backer Upper Trajectories

(c) 5 rules (reduced numeric data) + 9 expert rules

Comparison of (a), (b), (c) for 3,3,-30
ASuPFuNIS: High Performance and Economical

- Robust against random data set variations
- Economical in terms of network architecture while being able to yield a high performance
- Capable of handling both the numeric and linguistic information efficiently
- ASuPFuNIS performs well as a
  - Classifier/diagnostic system
  - Predictor
  - Controller
Need for Evolutionary Learning

- Gradient descent:
  - Sensitive to initialization of network parameters
  - Gets trapped in local minima
  - Cannot learn network structure
- Require: Simultaneous learning of structure and parameter estimation
  - Number of rule nodes
  - Essential features
  - Fuzzy set centers and spreads
Principles of Natural Selection

- Concentrate on **population** rather than a single individual
- Each individual has a fitness
- Individuals that are fit enough to survive will reproduce
- Create new individuals from existing ones
  - Crossover
  - Mutation
Specifics: Differential Evolution

- Initialize population
- Evaluate each vector, find best member
- Mutation and recombination
- Select child if better than parent
- Repeat until:
  - Predefined number of generations reached
  - All Vectors have converged
  - No Improvement after $x$ generations
EASuPFuNIS Coding

Feature spreads
Antecedent centers/spreads
Consequent centers/spreads

Antecedent Enable Bits
DE Strategy for Real Part

Old population

\[ V_0 = F^*(X_{R1} - X_{R2}) + X_{\text{best}} \]

Mutant Vector

Current member (R)

Exponential Crossover

Trial Vector (R)
DE Strategy for Enable Bit Part

Different bits: Perturb connection with probability 0.5

Identical bits: No perturbation in connection
Replacement

If the fitness of the trial vector is better than that of $X_1$, include the trial vector in the new population and keep the original member. Repeat this procedure for the entire population.
Parallelization

- Computation times are too large
- Parallel implementation of DE
  - Reduces computation time
  - Improves performance
- Parallelization Strategies
  - Master-slave model
  - Island model
  - Cellular model
Initial Implementation

- Master-slave
Implementation using LAM/MPI

- Local Area Multicomputer/Message Passing Interface (LAM/MPI)
- EASuPFuNIS coding scheme uses both real and integer variables
- Requires the use of MPI Derived Datatypes
  - Saves multiple sends
Cluster Specifications

- LAM/MPI version: 7.1.1
- Cluster:
  - Primary: 10 dedicated IBM X206 Pentium 3.2 GHz Processor based servers (RHL FC4)
  - Secondary: 15 non-dedicated IBM ThinkCentre desktops (RHL FC4)
  - Gbps/Mbps network switch
- Graphical tool used for instrumentation:
  - Jumpshot 4
Benchmarks Parallel DE Learning on ASuPFuNIS

Applications Used
- **MGTS** Time-series Prediction Problem
- **HANG** Function Approximation Problem
Mackey-Glass Time Series Prediction

- Chaotic time series (for $\tau = 17$)

\[
\frac{dx(t)}{dt} = \frac{0.2x(t) - \tau}{1 + x^{10}(t - \tau)} - 0.1x(t)
\]

- The problem involves predicting a future value based on a set of values at certain times less than $t$

\[
x(t + 6) = f(x(t - 18), x(t - 12), x(t - 6), x(t))
\]
MGTS Data generation

- Runge-Kutta procedure with time step size = 0.1
- 171 initial values
  - 170 initial conditions for $t < 0$ are assumed zero
  - Initial condition $x(0) = 1.2$
- 1000 input-output data pairs from $t = 118$ to 1117
  - Training set: First 500 data tuples ($t = 118$ to 517)
  - Test set: Second 500 data tuples ($t = 518$ to 1117)

$[x(t - 18), x(t - 12), x(t - 6), x(t); x(t + 6)]$
Mackey Glass Time Series

- The DE parameters for a 5/10 rule network used in the experiments

<table>
<thead>
<tr>
<th># of Rules</th>
<th>F</th>
<th>CR</th>
<th>Stringsize</th>
<th>Popsize</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.26</td>
<td>0.96</td>
<td>83</td>
<td>830</td>
</tr>
<tr>
<td>10</td>
<td>0.36</td>
<td>0.98</td>
<td>158</td>
<td>1580</td>
</tr>
</tbody>
</table>
Simulation Results

- PEASuPFNIS run times for 5000 generations on MGTS data ($q = 10$)
- Super-linear speedup

<table>
<thead>
<tr>
<th># Slaves</th>
<th>Run Time</th>
<th>$S_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>14 hrs 3 min</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>11 hrs 18 min</td>
<td>1.24</td>
</tr>
<tr>
<td>3</td>
<td>3 hrs 39 min</td>
<td>3.83</td>
</tr>
<tr>
<td>6</td>
<td>1 hr 38 min</td>
<td>8.59</td>
</tr>
<tr>
<td>9</td>
<td>1 hr 24 min</td>
<td>9.92</td>
</tr>
</tbody>
</table>
MGTS Approximation Performance

(a) Zoom plot of function approximation for 3 rule model
(b) Zoom plot of function approximation for 10 rule model

(c) Prediction error for 3 rule model
(d) Prediction error for 10 rule model
## Comparison of ASuPFuNIS NRMSEs with other models for MGTS Data

<table>
<thead>
<tr>
<th>Models</th>
<th>NRMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEFREX (Russo 2000)</td>
<td>0.0061</td>
</tr>
<tr>
<td>ANFIS (Jang 1993)</td>
<td>0.0074</td>
</tr>
<tr>
<td>NSFLS (Mendel et.al., 1997 )</td>
<td>0.0107</td>
</tr>
<tr>
<td>SuPFuNIS (10 rules)</td>
<td>0.016</td>
</tr>
<tr>
<td>(Paul &amp; Kumar 2002)</td>
<td></td>
</tr>
<tr>
<td>ASuPFuNIS (5 rules)</td>
<td>0.0148</td>
</tr>
<tr>
<td>(Velayutham &amp; Kumar2005)</td>
<td></td>
</tr>
<tr>
<td>PEASuPFuNIS (5 rules)</td>
<td>0.013</td>
</tr>
<tr>
<td>PEASuPFuNIS (10 rules)</td>
<td>0.008</td>
</tr>
</tbody>
</table>
Parallel Implementation Based on Migration

- Multi-population (island) model
- Each population evolves on its own
- Exchanges of randomly selected individuals take place at intervals between the islands
Synthetic Function Approximation Problem (HANG)

- Nonlinear system with two inputs and a single output
  \[ y = \left(1 + x_1^{-2} + x_2^{-1.5}\right)^2 \quad 1 \leq x_1, x_2 \leq 5 \]

- 50 randomly picked data points

- Two additional inputs added with values in the range [1,5] to which the output of the system is indifferent

- Performance Index
  \[
  PI = \sqrt{\frac{\sum_{k=1}^{Q} (d_k - y_k)^2}{\sum_{k=1}^{Q} |y_k|}}
  \]
DE Parameters for Hang

- 4-3-1 PE-ASuPFuNIS network used in the experiments
  - Inputs: 4
  - Rules: 3
  - Output: 1

<table>
<thead>
<tr>
<th>F</th>
<th>CR</th>
<th>Stringsize (Real part)</th>
<th>Enable bits (Binary part)</th>
<th>Popsize</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.49</td>
<td>0.96</td>
<td>53</td>
<td>12</td>
<td>530</td>
</tr>
</tbody>
</table>
Performance of PEASuPFuNIS

<table>
<thead>
<tr>
<th>Models</th>
<th>PI</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Fuzzy Model</td>
<td>0.01</td>
</tr>
<tr>
<td>(Sugeno and Yasukawa, 1993)</td>
<td></td>
</tr>
<tr>
<td>A Neuro-Fuzzy Model (Chakraborty and Pal, 2001)</td>
<td>0.01</td>
</tr>
<tr>
<td>PE-ASuPFuNIS</td>
<td>0.0023</td>
</tr>
<tr>
<td>(Singh and Kumar, 2006)</td>
<td>(with successful elimination of redundant features)</td>
</tr>
</tbody>
</table>
General Issues in NF System Design

- Identification of relevant features
- Selection of a minimal architecture
- Mode of representation of knowledge
- Design of inference system
- Learning mechanism
- Interpretation of embedded knowledge
- Computational complexity (!)
PE-ASuPFuNIS: Example of an Integrated CI System

- Robust against random data set variations
- Economical in terms of network architecture while being able to yield a high performance
- Capable of handling both the numeric and linguistic information efficiently
- ASuPFuNIS performs well as a
  - Classifier/diagnostic system
  - Predictor
  - Controller
Future Directions

- Implement more efficient and effective parallelization strategies
  - Island model
- Target real world applications
  - Bioinformatics
Simultaneous Parallel Architecture/Parameter Search

- Implement simultaneous search across multiple network architecture spaces
  - Phase 1: Architecture search
  - Phase 2: Parameter search and feature selection

- Targeting real world applications
Simultaneous Parallel Architecture/Parameter Search

Hybrid population DE

String migrations
Generations: From SuPFuNIS to PEASuPFuNIS

Generation 1
- SuPFuNIS

Generation 2
- ASuPFuNIS

Generation 3
- EASuPFuNIS

Generation 4
- PEASuPFuNIS
Some Relevant Publications

Papers


Books

Thanks from the nnl Team...

Sandeep Paul, C. Shunmuga Velayutham, Lotika Singh, Satish Kumar
Neural Networks Laboratory
Dayalbagh Educational Institute
Dayalbagh, Agra UP 282005 INDIA
skumar_db@ieee.org