Fuzzy Logic

1.

“Soft Computing”
- Neural networks
- Fuzzy logic
- Neuro-Fuzzy control
- Genetic algorithms

Reference
Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence

Fuzzy Logic History
- 1937, Max Black: Vagueness, an exercise in logical analysis, Phil. of Science, 4, 427-455
- 19xx, Jan Lukasiewicz
- 1967, Lotfi Zadeh, UCB: Fuzzy Sets, in Information and Control J.
- 1974, E.H. Mamdani, Control Systems
- 1980's-90's: Bart Kosko, USC

Fuzzy Founders/Followers
Jan Lukasiewicz (1878-1956)
Lotfi Zadeh (1921-)
Bart Kosko

Prof. Zadeh:
As the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.
Fuzzy Journals
- IEEE Trans. on Fuzzy Systems
- International J. of Approximate Reasoning
- Intelligent and Fuzzy Systems
- Journal of Cybernetics

Fuzzy Logic Applications
- Subway ride smoothness control
- Camcorder auto-focus and jiggle control
- Braking systems
- Saturn automobile transmission
- Copier quality control
- Rice-cooker temperature control

More Applications of Fuzzy Logic
- Automatic control of dam gates for hydroelectric-powerplants (Tokyo Electric Power)
- Camera aiming for the telecast of sporting events (Omron)
- Cruise-control for automobiles (Nissan, Subaru)
- Positioning of wafer-steppers in the production of semiconductors (Canon)
- Prediction system for early recognition of earthquakes (Inst. of Seismology Bureau of Metrology, Japan)
- Controlling of subway systems in order to improve driving comfort, precision of halting and power economy (Hitachi)

Air-Conditioning System (Mitsubishi)
- Problem description: Industrial air-conditioning system shall be able to react flexibly to changing ambient conditions
- Realization:
  - 50 rules
  - 6 linguistic variables
  - Resolution: 8 bit
  - Input variables: room temperature, wall temperature and temporal evaluation of these signals
- Development:
  - 4 days to create the prototype
  - 20 days for testing and integration
  - 80 days for optimization with real test objects
- Implementation as pure software solution on standard microcontroller
- Results:
  - Reduction of starting processes down to 40 percent of the standard solution
  - Sustaining of the temperature even with interference factors (like open window, etc.) substantially improved
  - Fewer sensors required, Established energy saving by testing: 24 percent

Fuzzy Silver Bullet?
- Fuzzy logic may not provide any new mechanisms that weren’t there before.
- It provides a viewpoint, that helps expedite problem solving.
- Analogy: Object-Oriented Programming didn’t create any new computable functions.

Fuzzy Set Basics
- Classical (“crisp”) sets:
  - Membership in a set is all or nothing
  - Characteristic function \( c_S(x) \): Universe \( \{0, 1\} \)
  - \( c_S(x) = 1 \) iff \( x \in S \)
- Fuzzy sets:
  - Membership in a set is a degree
  - membership function \( c_S \): Universe \( [0, 1] \)
Linguistic Characterizations of Degree of Membership

- Consider the set of “hot” days in Claremont in 2003.
- Was Oct. 29 “hot”? It might have been called one of:
  - “very hot”
  - “sort of hot”
  - “not hot”
- The answer depends on the observer, time, etc.

Sounds similar to probability, but isn’t

- Probability deals with uncertainty or likelihood of occurrence.
- Fuzzy logic deals with ambiguity, vagueness of description.

Fuzzy Membership

Are these disks, cylinders, or rods?

Fuzzy Membership

Which of these is a pile of sand?

Membership function plots

Crisp vs. Fuzzy Membership Functions

<table>
<thead>
<tr>
<th>is a pile</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td># grains of sand</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>is a pile</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>grains of sand</td>
<td>0</td>
</tr>
</tbody>
</table>
More Crisp vs. Fuzzy Membership Functions

Crisp

Fuzzy

Note: Universe can be continuous or discrete, ordered or unordered.

Matlab’s Built-in Membership Functions

Linguistic Modifiers

Example: Composites of Young and Old

Discrete Universe Examples

Fuzzy-Set Operations expressed using membership functions

Fuzzy OR (union)

Fuzzy AND (intersection)
Fuzzy Complement

\[ c_A(x) = 1 - c_A(x). \]

Which Set-Theoretic Rules Hold?

\[
\begin{align*}
A \cap B &= B \cap A \\
A \cup B &= B \cup A \\
(A \cap B) \cap C &= A \cap (B \cap C) \\
(A \cup B) \cap C &= A \cup (B \cap C) \\
(A \cap B) \cup C &= (A \cup C) \cup (B \cup C) \\
(A \cup B) \cup C &= (A \cup C) \cup (B \cup C) \\
A' \cap B' &= A'' \cap B'' \\
A' \cap B' &= A' \cap B'
\end{align*}
\]

etc.

Fuzzy Anomaly?

The intersection of a set with its complement is not necessarily empty.

\[ c_A(x) = 1 - c_A(x). \]

Fuzzy Implication

There is no single standard. A variety of versions exists:

- **Larsen:** \[ x \overset{\Rightarrow}{\land} y = xy \]
- **Lukasiewicz:** \[ x \overset{\Rightarrow}{\land} y = \min(1, 1-x+y) \]
- **Mamdani:** \[ x \overset{\Rightarrow}{\land} y = \min(x, y) \]
- **Standard strict:** \[ x \overset{\Rightarrow}{\land} y = x \leq y ? 1 : 0 \]
- **Goedel:** \[ x \overset{\Rightarrow}{\land} y = x \leq y ? 1 : y \]
- **Gaines:** \[ x \overset{\Rightarrow}{\land} y = x \leq y ? 1 : y/x \]
- **Kleene-Dienes:** \[ x \overset{\Rightarrow}{\land} y = \max(1-x, y) \]
- **Kleene-Dienes-Luk:** \[ x \overset{\Rightarrow}{\land} y = 1-x+yxy \]

Fuzzy If-Then Rules

- **Mamdani style**
  If pressure is high then volume is small

- **Sugeno style (uses equations on rhs)**
  If speed is medium then resistance = 5*speed
**Problem**

- Determine a fuzzy rule-base adequate to specify the pole-cart controller.

**Pole-on-Cart Balancing Example**

- Angle of pole
  - neg.high
  - neg.low
  - pos.low
  - pos.high

**Pole-on-Cart Example**

- Speed of cart
  - neg.high
  - neg.low
  - pos.low
  - pos.high

The speed can have non-zero membership in more than 1 category.

**Pole-on-Cart Balancing Example**

- Angular velocity of pole
  - neg.high
  - neg.low
  - pos.low
  - pos.high

- neg.velocity
  - zero
  - pos.velocity

- neg.velocity
  - zero
  - pos.velocity
Example of a fuzzy-logic rule represented in this table: "If the angular velocity is pos. low and the angle is zero, then set the speed at low."

Fuzzy Rule Base
(Kosko: FAM- Fuzzy Associative Memory)

<table>
<thead>
<tr>
<th>Angle</th>
<th>neg. high</th>
<th>neg. low</th>
<th>zero</th>
<th>pos. low</th>
<th>pos. high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle</td>
<td>neg. high</td>
<td>neg. low</td>
<td>zero</td>
<td>pos. low</td>
<td>pos. high</td>
</tr>
<tr>
<td>pos. low</td>
<td>neg. high</td>
<td>neg. low</td>
<td>zero</td>
<td>pos. low</td>
<td>pos. high</td>
</tr>
<tr>
<td>pos. hi</td>
<td>neg. high</td>
<td>neg. low</td>
<td>zero</td>
<td>pos. low</td>
<td>pos. high</td>
</tr>
</tbody>
</table>

Control speed as a function of angle and angular velocity

Inference in a (Mamdani-style) Fuzzy System

- Start with quantitative input data
- **Fuzzify** the data
- Derive conclusion based on fuzzy data
- **De-fuzzify** the conclusion to get quantitative output

Fuzzification

In this case, the actual angle is a mixture of zero and pos. low.

Fuzzification

Here the actual angular velocity is a mixture of zero and neg. low.

Multiple Applicable Rules:

- Angle is a mixture of zero and pos. low.
- Angular velocity is a mixture of zero and neg. low

Three entries in the rule base are applicable. We must determine how to combine them.

Use the diagrams to determine the degree to which each rule is applicable.

- Consider the rule
  "If angle is zero and angular velocity is zero, the speed is zero."
  The actual value belongs to the fuzzy set zero to a degree of 0.75 for "angle" and to a degree of 0.4 for "angular velocity".
  - Since this is an AND operation, the minimum criterion is used.
    (For OR, the maximum would be used.)
  - The fuzzy set zero of the variable "speed" is cut at 0.4 and the patches are shaded up to that area.
Using min combination for AND  
(This is for one of three speed rules: zero.)

Similarly, for each of the (3) applicable rules we get an inferred speed.

We then combine the results of the three rules to get an output fuzzy set.

Defuzzification

To actually set the speed, we need a number, not a fuzzy set. Various rules can be used to get the number. The most common one is to use the centroid of the fuzzy set.

Centroid Review

Matlab’s Defuzzification Rules
Control Surface for an Inverted Pendulum:
Force = f(Angle, AngularVelocity)

Fuzzy Truck-Backer Applet
/cs/cs152/fuzzy/truck

Rule base

An example simulation

Re-Programmability buttons

Position

90°
A different example simulation

Control Surface for another Truck-Backer

Another Truck Backer using only 9 Rules
(/cs/cs152/fuzzy/fismat/lisdemo)
Kong & Kosko compared Fuzzy vs. Neural Controllers

- **Fuzzy**
  - Regular path followed
  - “Trained” by common-sense hand-coded rules
  - Light-weight controller (comparisons and additions only)

- **Neural**
  - Sometimes followed irregular path
  - Training time-consuming
  - Controller computationally intensive

Kong & Kosko Derived Controller Functions

Kong & Kosko Konclusions

- **Fuzzy**
  - Regular path followed
  - “Trained” by common-sense hand-coded rules
  - Light-weight controller (comparisons and additions only)

- **Neural**
  - Sometimes followed irregular path
  - Training time-consuming
  - Controller computationally intensive

Matlab Fuzzy Ball-Juggler Demo

Applications of Fuzzy Logic to Training MLP Neural Networks

- Control of variable learning rate in backpropagation (Choi, et al.):
  - Let CE denote Change of Error.
  - Let CCE denote Change of CE.
  - Fuzzy Rules:
    - If CE is small with no recent sign changes, then increase LR.
    - If CE has recent sign changes, then decrease LR.
    - If CE is small, and CCE is small with no recent sign changes, then increase LR and momentum.
Application of Neural Networks to Fuzzy Logic

- Learned fuzzy rules for forecasting gas demand.
- Used backpropagation network with five layers.
- The input layer contains two nodes - temperature and the month and the output layer is the gas demand figure.
- Fuzzy rules are then extracted from the net.

Sugeno Control Model

- Extends Mamdani model
- Fuzzy part is still in antecedent of rules, which are used for selection
- Consequent of rules is more complex: some function (e.g. polynomial) of input variables

Fuzzy Inference System Using Sugeno-Style Rules

If speed is low then resistance = 2
If speed is medium then resistance = 4*speed
If speed is high then resistance = 8*speed

MFs low medium high

Rule 1: w1 = .3; r1 = 2
Rule 2: w2 = .8; r2 = 4*2
Rule 3: w3 = .1; r3 = 8*2

Resistance = ∑(wi*ri) / ∑wi = 7.12

Example: 1-input Sugeno

- Rules:
  - If X is small, then Y = 0.1X + 6.4.
  - If X is medium, then Y = -0.5X + 4.
  - If X is large, then Y = X-2.

- The following 2 slides indicate the results of combining these rules using crisp vs. fuzzy logic.
Example: 2-input Sugeno

- Rules:
  - If X is small and Y is small, then \( Z = -X + Y + 1 \).
  - If X is small and Y is large, then \( Z = -Y + 3 \).
  - If X is large and Y is small, then \( Z = -X + 3 \).
  - If X is large and Y is large, then \( Z = X + Y + 2 \).

MFs for demo sug2 (2-input)

Control surface for sug2

Tsukamoto model

- Aggregate rule outputs by a weighted average, rather than by defuzzification.

Example of Hybrids: ANFIS
(Adaptive Neuro-Fuzzy Inference System)

- Developed by J.-S. R. Jang
- Uses Sugeno or Tsukamoto models
- Similar to Radial Basis Function network

ANFIS

- Fuzzy reasoning

- ANFIS (Adaptive Neuro-Fuzzy Inference System)
Contrary Opinion

WHY I DESPISE FUZZY FEEDBACK CONTROL, by Michael Athans
http://www.lit.net/ieee/cdc98/debates/cdc98fuzzy-debate/

Asserts that fuzzy control only shown applicable to trivial (SISO) systems, not more complex (MIMO) systems.

Athans’ Asserted Shortcomings of Fuzzy Controllers

- Fuzzy rules just generate nonlinear static functions
- Performance specifications “vague” or nonexistent
- Cannot generate “differential equation” controller rules
- Not easy to differentiate noisy sensor signals by finite differencing, as almost always done in fuzzy applications
- No utilization of dynamic (e.g. Kalman) filtering
- I have never seen a multiple-input multiple-output (MIMO) fuzzy control application
- Combinatorial complexity for high-order and multivariable applications