Fuzzy Logic
GA

Sabah Sayed

Department of Computer Science

Faculty of Computers and Artificial Intelligence

Cairo University

Egypt

Evolving Fuzzy Systems using Genetic Algorithms

- We want to evolve a fuzzy rule system using a GA.

- **Things to consider:**
- What will be evolved (what parts of the fuzzy system)
- How system elements are represented
- Population initialization
- Fitness evaluations
- Operators used

What will we evolve?

- Usually the rule base is evolved to get the best rules representing a dataset of a particular problem.

- We can also evolve the membership functions used to find the best types (triangular, trapezoidal, ...etc) and ranges.

Representing Fuzzy Rules

- Each chromosome will represent a single rule set
- Our system will use Mamdani-type fuzzy rules, for example:

  if input_1 is not Low, and input_2 is High, then the output is Medium

  if input_2 is Low, then the output is High

- but computers generally can't handle this kind of input (yet)
- Assuming each variable has three fuzzy sets; Low, Medium, High:

  Let Low=1   Medium=2   High=3

  The above two rules can be numerically represented as:

  `-1 3 2 and 0 1 3 (0 means "don't care")`

Population Initialization

- Random initialization of the population.

- Rules are checked for feasibility.
- Each rule must have a non-zero antecedent, *and* a non-zero consequent.
- A rule such as 2 3 1 1 0 is thus non feasible, and is not included in the rule set.

- Re-generate individuals to compensate for the infeasible chromosomes.

Fitness Evaluation

- Percent correct and mean-square error are often used.

- Additional components of fitness may include the number of rules (minimize).

Steps of Evolving a Fuzzy Rule System

- Given a history of data (dataset), to a particular problem:

x1	x2	x3	......	xm	y
....	....	....	....	....	....
....	....	....	....	....	....
....	....	....	....	....	....

it is required to evolve a fuzzy system that will classify or predict y.

Evolving a Fuzzy Rule System

- **<u>Step 1:</u>**
  Get min and max of each Xi (column) in dataset => $X_{i\text{-}min}$, $X_{i\text{-}max}$

- **<u>Step 2:</u>**
  Design fuzzy sets (select them)
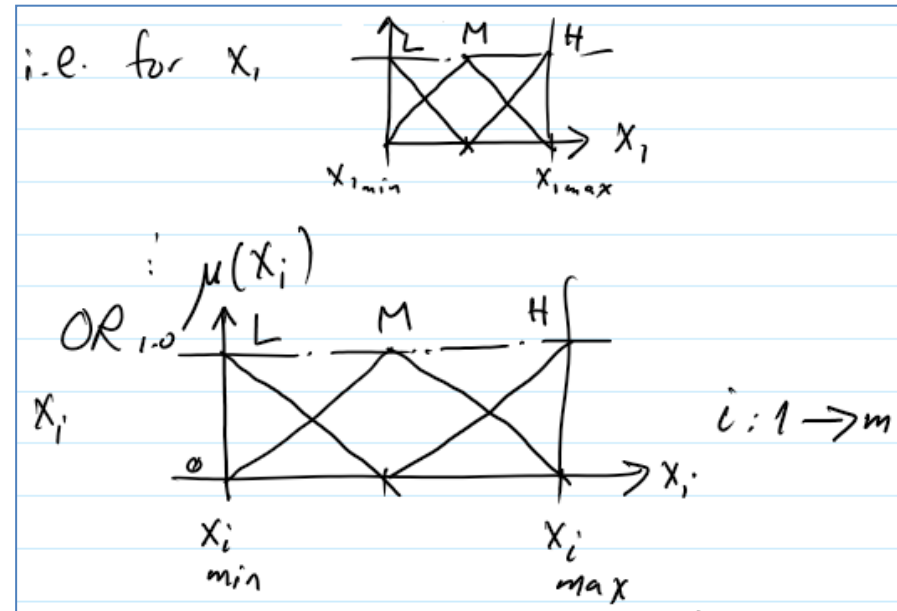
Evolving a Fuzzy Rule System

- **Step 3:**
  Design chromosomes composed of random set of fuzzy rules as follows:

- R_j:
  if x1=M and x2=L
  then y=H
- R_k:
  if x1=H
  then y=L
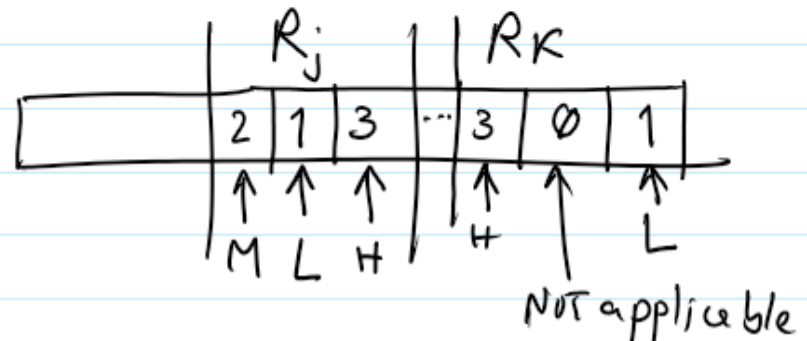
Evolving a Fuzzy Rule System

- **<u>Step 3:</u>**
Note that number of rules in chromosome is variable => variable-length chromosome

chromosome 1:

R1	R2	R3	R4

- **chromosome 2:**

R5	R6	R7

Evolving a Fuzzy Rule System

- ## **Step 4:**
  Design of crossover operator

R1	R2	R3	R4

R5	R6	R7

Random crossover point

Random crossover point

R1	R6	R7

R5	R2	R3	R4

Evolving a Fuzzy Rule System

- **Step 5:**
  Design of mutation operator

- According to random number **r**,
  for each entry in rule, either:
  - Add 1 (r: 0- 0.33)

    ex. L to M
  - Subtract 1 (r: 0.34 – 0.66)

    ex. H to M
  - Negate (r: 0.67 - 1)

    ex. L to Not L
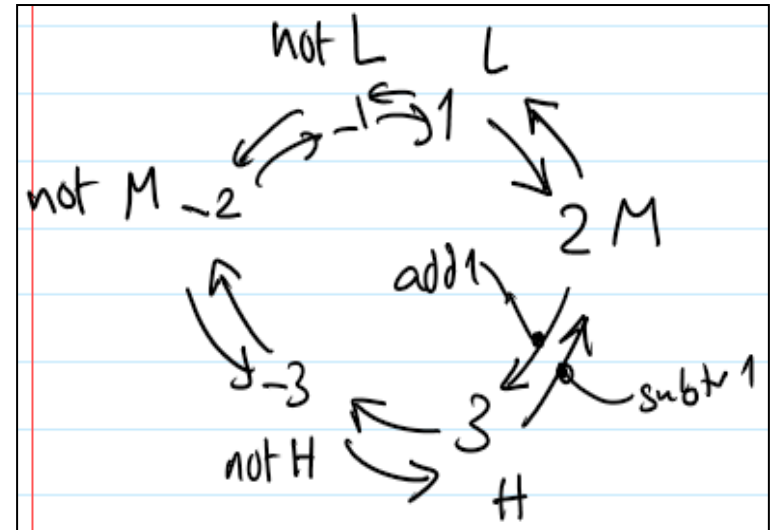
Evolving a Fuzzy Rule System

- **Step 6:**
  Fitness function

- Run each fuzzy system (chromosome) as usual on dataset

- Fitness$_i$ = (number of correctly predicted rows)$_i$

- Or Fitness$_i$ = 1/MSE$_i$ (mean square error)

Calculate Fitness of Chromosome k:
Mean Square Error

chr_K | R_1 | R_2 | R_3 | R_4

Ruleset

for each chromosome chr_K in GA Population:
{
   Sum-of-SEs = 0
   for each record r_i in the dataset $[r_1, r_2 ---- r_p]$:
   {
      ✗ Use values of $x_{1i}, x_{2i}, x_{3i} ---- x_{mi}$ to perform:

       ① Fuzzification
       ② Inference ⟶ Rules $R_1, R_2, R_3 ---$ in chr_K
       ③ Defuzzification ⟶ to get $Y_{i\,(predicted)}$

      ✗ Calculate Square Error $(SE) = \left(Y_{i\,(predicted)} - Y_{i\,(actual)}\right)^2$

      ✗ Sum-of-SEs += SE
   }

   $MSE_k = $ Sum-of-SEs $/P$ $P =$ num. of records in data set

   Fitness of $chr_k = 1/MSE_k$
}

Evolving a Fuzzy Rule System

- **Step 7:**
Selection Strategy:
     Roulette wheel selection

- **Step 8:**
Replacement strategy:
     Elitism Strategy