Soft Computing Methods and Applications

Lab Exercise and Assignment #5 (2020)

Develop Fuzzy If-Then Rule Class

(1) Enhance your Assignment 04 to the new Assignment 05, using the tricks you did in the last assignment.

(2) Make sure you have added a ParameterChanged event (or MaximumChanged, MinimumChanged, ResolutionChanged events separately) to your Universe class. Make sure your FuzzySet class has subscribed the event, since a fuzzy set object always depends on the universe. Make sure that when users change the range of the universe, fuzzy set displays are updated accordingly.

(3) Make sure your fuzzy set has implemented a ParameterChanged event and fire the event when a particular parameter is changed. Make sure your UnaryOperated and BinaryOperated fuzzy sets have subscribed parameter changed events from the dependent fuzzy sets and operators.

(4) As it is suggested to override C# operators for unary and binary operations on fuzzy sets. Make sure you have completed the operator overloading implementations for your fuzzy sets. For example, operator ! for Not operation (!A), operator & for And (Intersect) operation (A & B), operator | for Or (Union) operation (A | B), binary operator * for Scaling operation (0.3 * A) that generates UnaryOperatedFuzzySet, binary operator – for Cutting operation (0.3 –A) or unary operator – for Cutting operation (–A). Available operators to be overridden include: (binary) &,|,&&,||,+, -,*,/, <, >, <<, >>, ^, %; (unary) !,~, +, -, ++, --.

(5) Enhance your Fuzzy Set to add a virtual public property (or operator), say double MaxDegree, to the FuzzySet class, which will be used to inference a fuzzy rule. Override the property in the derived classes.

(6) Add a class for fuzzy if-then rule creations; e.g. class IfThenFuzzyRule.

- Investigate the data structure required for a fuzzy if-then rule to define its data fields and properties (and possible events). For example, the antecedent part should be a list of fuzzy sets on different universes and the conclusion part is a fuzzy set on another universe.
- In the class, design required public methods from callers' perspective, to support fuzzy inferencing. For example,
  public FuzzySet FuzzyInFuzzyOutInferencing(List<FuzzySet> condition),
  where condition is a given list of condition FuzzySets.
- Fault-proof mechanism should be adopted; e.g., antecedent fuzzy sets should be defined in different input universes and the output fuzzy set should be defined in different universe from input universes.

(7) Prepare UIs for the users to create and show fuzzy if-then rules (Button & DataGridView). In addition, provide UIs for the users to define a condition and then inference the result (output fuzzy set) against the defined rules.

(8) In particular,upgrade your TreeView UI to have the first level divided into input and output parts.

When one output universe is defined, no more universes can be defined in the output part. Use two DataGridView controls to host the rules and the condition for inferencing. Dynamically update the column structure of the grids following the creation and deletion of universes. Prepare button to let user add a data row for defining an if-then rule and click on the cells to specify fuzzy sets on the rule. In condition DataGridView, only one row of antecedent part is defined in advance. User can specity the condition part for inference. Add a button to let user inference the specified condition to get the resulting fuzzy set shown on the output universe.

(9) Adjust the FuzzySet creation logic to have no series points generated by default. During the fuzzy rule inferencing, a number of operated fuzzy sets are generated to facilitate the computation process. Only the final resulting fuzzy set is to be displayed. To save memory usage, by default a fuzzy set does not instantiate points of series and does not show up in the chart area.