

Soft Computing Methods and Applications

Lab Exercise and Assignment #6 (2020)

Fuzzy inference systems

1. Clone your Assignment 05 as a new Assignment 06.
2. If you didn't implement bool ShowSeries flag and your fuzzy set always shows series points on the chart, implement the flag to let data storage of series and line display optional. By default, the series of a fuzzy set is not added and calculated nor displayed in the chart area of the universe. Therefore, the function that updates series points is by default not invoked. Add the flag (bool type) to let user modify the property.
3. Extend the functionalities of the FuzzySet set class to support the development of fuzzy inference systems. Add required virtual and overridden functions and/or read-only properties: e.g.,

```
public virtual double MaxDegree{ get{...} }  
public virtual double COACrispValue{ get{...} }  
public virtual double BOACrispValue{ get{...} }  
...
```

4. In the previous assignment, the list of user specified fuzzy if-then rules is defined in your UI form class. Now it is the time to hand over the list to various fuzzy inference systems. Aggregate objects of the available IfThenFuzzyRule class to define different fuzzy inference systems:

```
class MamdaniFuzzySystem  
class SugenoFuzzySystem  
class TsukamotoFuzzySystem
```

Note: output functions for Sugeno models can/shall be hard-coded on your code. You can define several public functions to cover the sample equations shown in the examples of the textbook, for the user to define the output equation of a Sugeno if-then rule in constructing a Sugeno system. In other words, you should define a SugenoIfThenFuzzyRule class to incorporate these hard-coded functions for the user to select. Moreover, proper UIs should be provided for the user to select output equation. These hard-coded functions can be calculated from this member function:

```
public double GetOutputValue( double[] inputs, int equationID )
```

```

{
    switch(equationID )
    {
        case 0: // Y=0.1X+6.4
            return 0.1 * inputs[0] + 6.4;
            break;
        case 1: // Y=0.5X+4
            return 0.5 * inputs[0] + 4;
            break;
        ...
    }
}

```

5. You need to define interface functions to let user construct an inference system and perform either a single condition inferencing to get a single result (a cut- or scaled- fuzzy output or a crisp output from a defuzzification method) or a complete crisp-in-crisp-out inferencing on all combinations of universe values that yields a complete input-output map. For example:

```

Public class MamdaniFuzzySystem
{
    IfThenFuzzyRules[] allRules;
    // constructor ...

    ...

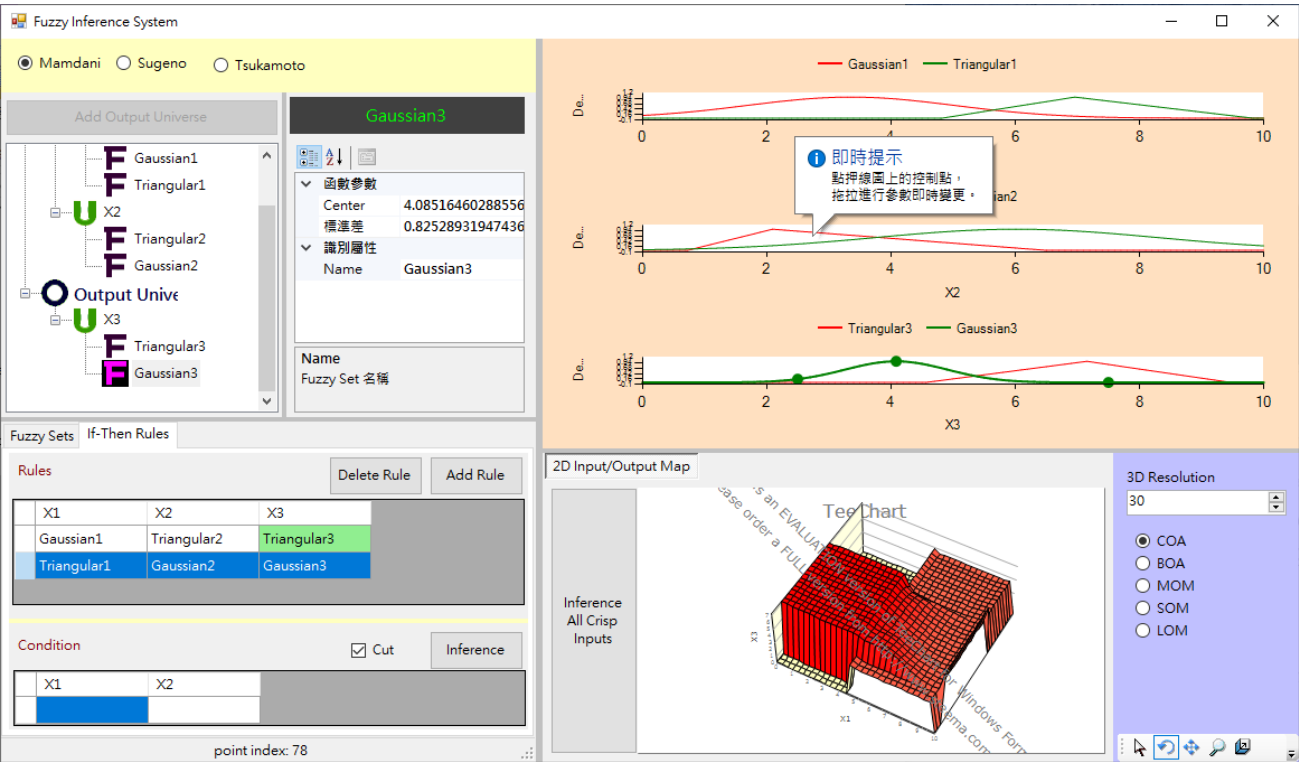
    public FuzzySet FuzzyInFuzzyOutInferencing(FuzzySet[] conditions ) { ... }
    public double CrispInCrispOutIferencing(
        double[] conditions, DefuzzificationType type ) { ... }
    public FuzzySet CrispInFuzzyOutIferencing( double[] conditions) { ... }
}

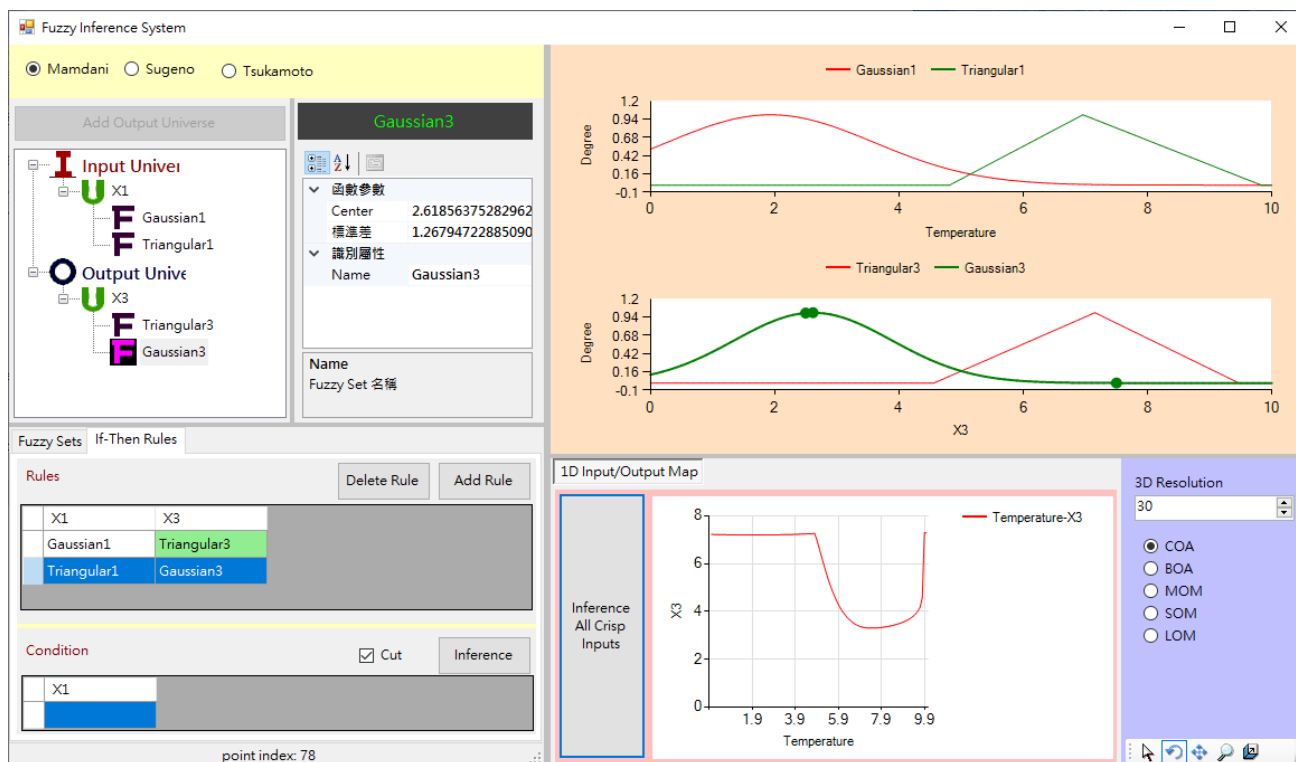
public enum DefuzzificationType { COA, BOA, ... }

```

6. Following the lab instructions to design UIs for inference system construction. The UIs will let the users define a set of fuzzy if-then rules. In addition, defuzzification mode can be selected to generate crisp output values.

7. Provide UIs for the users to execute automatic crisp-in-crisp-out inferencing and obtain the output response lines or surfaces (on 3D chart). MS Chart does not support 3-D surface charting. You can download a trial version of TeeChart component from Steema Inc. Alternatively, you can check our bulletin board for a binary library of TeeChart. Refer to the lab instructions of implementing a 3D surface plot. For a crisp value inferencing, the condition turns out to be an array of real numbers of double type. Notice that, in this crisp-input case, no intersection of two fuzzy sets (a singleton fuzzy set and antecedent fuzzy set) and the maximal degree of it are required for inferencing. We simply use the given crisp input value to evaluate its membership degree, which turns out to be the firing strength of the antecedent term for the given crisp value and the rule.





8. Add model save and read capability, as possible as you could. This will save the tedious construction steps. Run your application to construct fuzzy inference systems and inference the crisp input-output response as shown in the following examples.
9. Video tape the execution of each model to show the modeling and inferencing capability of your App. You don't need to video tape detailed construction steps. Instead, video tape the constructed model and inferencing results, by browsing the tree view, property grid, and interactively rotate the 3D response surface to demonstrate your app did generate the model and inference the results.
10. If your submission exceeds the upload limit to COOL, move your video file to cloud and submit a text file indicating the download link. Submit the link information as a text file and send it with your source code as well.

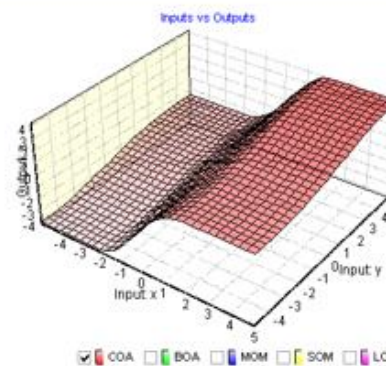
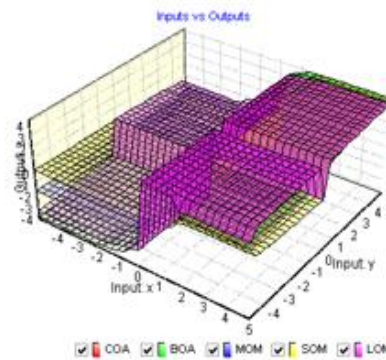
Mamdani fuzzy model shown in Example 4.2 (Fig. 4.6)

If x is small and y is small then z is negative large

If x is small and y is large then z is negative small

If x is large and y is small then z is positive small

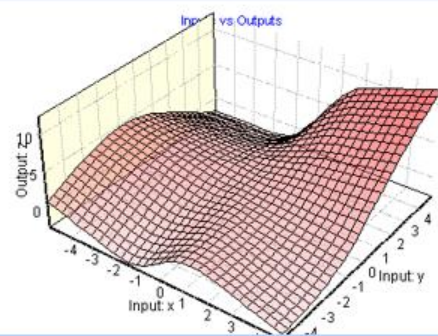
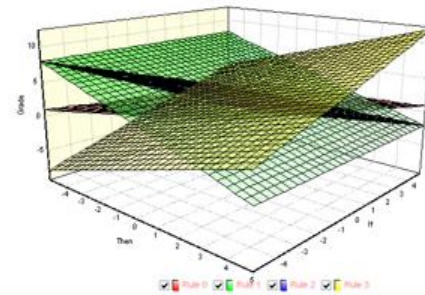
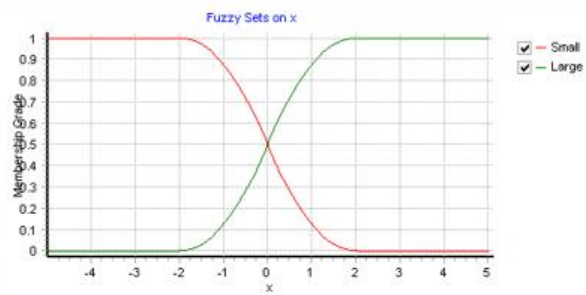
If x is large and y is large then z is positive large



Input-Output
(COA Map)

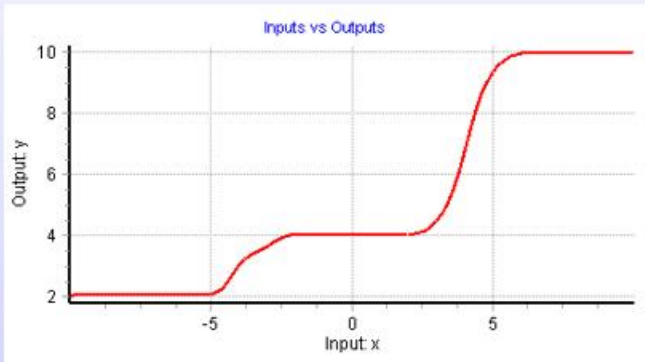
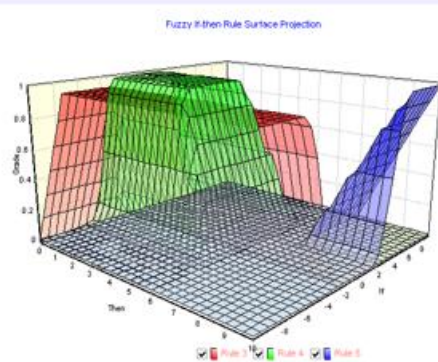
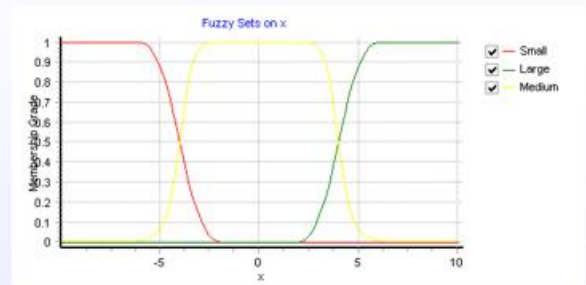
Sugeno fuzzy model shown in Example 4.4 (Fig. 4.10)

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 $z = x + y + 2$

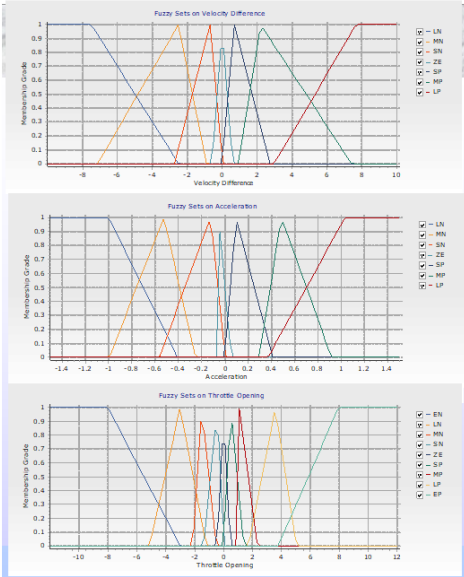


Tsukamoto fuzzy model shown in Example 4.5 (Fig. 4.12)

If x is small then y is c_1
 If x is medium then y is c_2
 If x is large then y is c_3



Platoons of cars (Mamdani)(optional)



		velocity Difference						
Throttle		LN	MN	SN	ZE	SP	MP	LP
Acceleration	LN	ZE	MP	MP	MP	LP	LP	EP
	MN	MN	ZE	ZE	MP	MP	MP	LP
	SN	MN	SN	SN	SP	SP	MP	LP
	ZE	LN	MN	SN	ZE	SP	MP	LP
	SP	LN	MN	SN	SN	SP	SP	MP
	MP	EN	LN	MN	MN	SN	SP	MP
	LP	EN	LN	LN	MN	MN	SN	ZE

