**Sistemas Inteligentes**
Grado en Ingeniería Informática
elisa.guerrero@uca.es
Departamento Ingeniería Informática

## Building a Fuzzy System from Scratch:

You can build o modify a fuzzy inference system (FIS) using Fuzzy Logic Toolbox commands as an alternative to the GUI tools. For instance, to load the tipping system from disk:

```
a = readfis('tipper.fis')

a =       name: 'tipper'

          type: 'mamdani'

          andMethod: 'min'

          orMethod: 'max'

          defuzzMethod: 'centroid'

          impMethod: 'min'

          aggMethod: 'max'

          input: [1x2 struct]

          output: [1x1 struct]

          rule: [1x3 struct]
```

**RULES**

Each variable, input, or output, has an index number, and each membership function has an index number. The rules are built from statements such as the following:

- **If input1 is MF1 or input2 is MF3, then output1 is MF2 (weight = 0.5)**

This rule is turned into a structure according to the following logic. If there are m inputs to a system and n outputs, then the first m vector entries of the rule structure correspond to inputs 1 through m.

- The entry in column 1 is the index number for the membership function associated with input 1.
- The entry in column 2 is the index number for the membership function associated with input 2, and so on.
- The next n columns work the same way for the outputs.
- Column m + n + 1 is the weight associated with that rule (typically 1) and column m + n + 2 specifies the connective used (where **AND = 1** and OR = 2).

- The structure associated with the preceding rule is: **1 3 2 0.5 2,** for fuzzy system X:

```
X.rule(1).antecedent=[1 3];

X.rule(1).consequent=[2];

X.rule(1).weight=0.5;
```

```
a.rule(1).connection=2;
```

Alternatively, you can build the entire tipping system from the command line using Fuzzy Logic Toolbox commands: newfis, addvar, addmf, and addrule.

```
a=newfis('tipper');

a=addvar(a,'input','service',[0 10]);

a=addmf(a,'input',1,'poor','gaussmf',[1.5 0]);

a=addmf(a,'input',1,'good','gaussmf',[1.5 5]);

a=addmf(a,'input',1,'excellent','gaussmf',[1.5 10]);

a=addvar(a,'input','food',[0 10]);

a=addmf(a,'input',2,'rancid','trapmf',[-2 0 1 3]);

a=addmf(a,'input',2,'delicious','trapmf',[7 9 10 12]);

a=addvar(a,'output','tip',[0 30]);

a=addmf(a,'output',1,'cheap','trimf',[0 5 10]);

a=addmf(a,'output',1,'average','trimf',[10 15 20]);

a=addmf(a,'output',1,'generous','trimf',[20 25 30]);

ruleList=[ ...

1 1 1 1 2

2 0 2 1 1

3 2 3 1 2 ];

a=addrule(a,ruleList);
```

**EVALUATION**

To evaluate the output of a fuzzy system for a given input, use the function evalfis. For example, the following script evaluates tipper at the input, [1 2].

```
a = readfis('tipper');

evalfis([1 2], a)

ans =

  5.5586
```

This function can also be used for multiple collections of inputs, because different input vectors are represented in different parts of the input structure.

```
evalfis([3 5; 2 7], a)

ans =

   12.2184

    7.7885
```

**Defuzzification**

```
x1 = defuzz(x,mf1,'centroid')
```

**System Display Functions**

Because the variable, a, designates the fuzzy tipping system, you can display any of the GUIs for the tipping system directly from the command line. Any of the following functions will display the tipping system with the associated GUI:

- **fuzzy**(a) displays the FIS Editor.
- **mfedit**(a) displays the Membership Function Editor
- **ruleedit**(a) displays the Rule Editor.
- **ruleview**(a) displays the Rule Viewer.
- **surfview**(a) displays the Surface Viewer.

There are three functions designed to give you a high-level view of your fuzzy inference system from the command line:

- **plotfis**(a)
- **plotmf**(a,'input',1): after closing any open MATLAB figures or GUI windows, the function plotmf plots all the membership functions associated with a given variable as follows.
- **gensurf**(a)