Fuzzy Inference (Expert) System

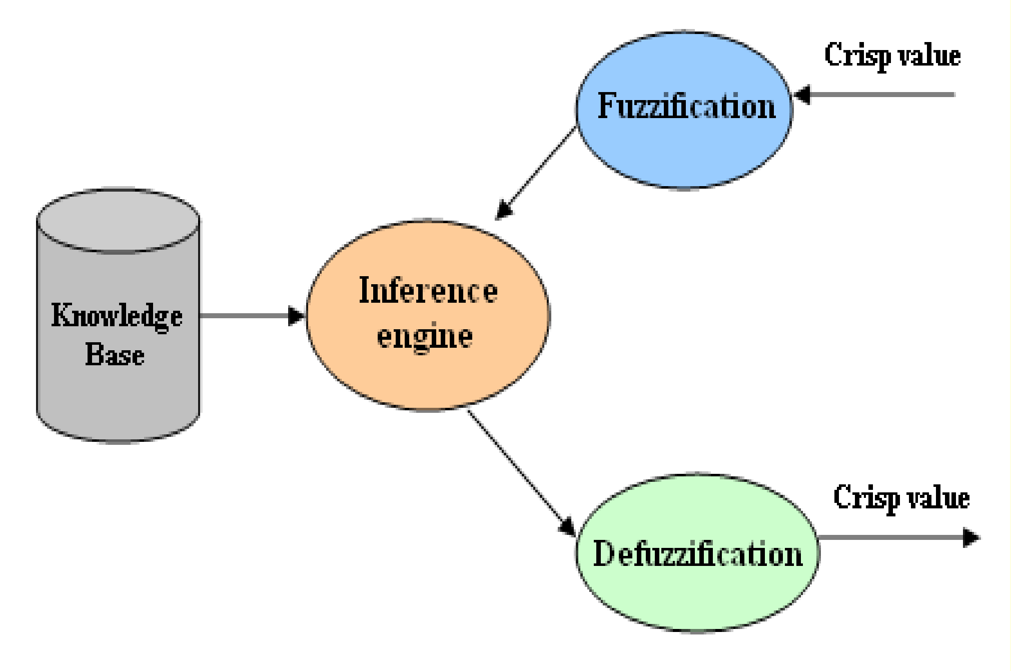
Introduction

- A Fuzzy Inference System (FIS) is a way of mapping an input space to an output space using fuzzy logic
- FIS uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data.
- The rules in FIS (sometimes may be called as fuzzy expert system) are fuzzy production rules of the form:
  - *if p then q,* where p and q are fuzzy statements.
- For example, in a fuzzy rule
  - if x is low and y is high then z is medium.
  - Here x is low; y is high; z is medium are fuzzy statements; x and y are input variables; z is an output variable, low, high, and medium are fuzzy sets.

Cont…

- The antecedent describes to what degree the rule applies, while the conclusion assigns a fuzzy function to each of one or more output variables.
- Most tools for working with fuzzy expert systems allow more than one conclusion per rule.
- The set of rules in a fuzzy expert system is known as *knowledge base*.
- The functional operations in fuzzy expert system proceed in the following steps.
  - Fuzzification
  - Fuzzy Inferencing (apply implication method)
  - Aggregation of all outputs
  - Defuzzification

Structure of a Fuzzy Expert System



Fuzzification

- In the process of fuzzification, membership functions defined on input variables are applied to their actual values so that the degree of truth for each rule premise can be determined.
- Fuzzy statements in the antecedent are resolved to a degree of membership between 0 and 1.
  - If there is only one part to the antecedent, then this is the degree of support for the rule.
  - If there are multiple parts to the antecedent, apply fuzzy logic operators and resolve the antecedent to a single number between 0 and 1.
- Antecedent may be joined by OR; AND operators.
  - For OR -- max
  - For AND -- min

Fuzzy Inferencing

- In the process of inference
  - Truth value for the premise of each rule is computed and applied to the conclusion part of each rule.
  - This results in one fuzzy set to be assigned to each output variable for each rule.
- The use of degree of support for the entire rule is to shape the output fuzzy set.
- The consequent of a fuzzy rule assigns an entire fuzzy set to the output.
- If the antecedent is only partially true, (i.e., is assigned a value less than 1), then the output fuzzy set is truncated according to the implication method.

Cont...

- If the consequent of a rule has multiple parts, then all consequents are affected equally by the result of the antecedent.
- The consequent specifies a fuzzy set to be assigned to the output.
- The implication function then modifies that fuzzy set to the degree specified by the antecedent.
- The following functions are used in inference rules.
- *min* or *prod* are commonly used as inference rules.
  - *min*: truncates the consequent's membership function
  - *prod*: scales it.

Aggregation of all outputs

- It is the process where the outputs of each rule are combined into a single fuzzy set.
- The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule.
- The output of the aggregation process is one fuzzy set for each output variable.
  - Here, all fuzzy sets assigned to each output variable are combined together to form a single fuzzy set for each output variable using a fuzzy aggregation operator.
- Some of the most commonly used aggregation operators are
  - the maximum : point-wise maximum over all of the fuzzy sets
  - the sum : (point-wise sum over all of the fuzzy
  - the probabilistic sum.

Defuzzification

- In Defuzzificztion, the fuzzy output set is converted to a crisp number.
- Some commonly used techniques are the *centroid* and *maximum* methods.
  - In the *centroid method*, the crisp value of the output variable is computed by finding the variable value of the centre of gravity of the membership function for the fuzzy value.
  - In the *maximum method*, one of the variable values at which the fuzzy set has its maximum truth value is chosen as the crisp value for the output variable.
- Some other methods for defuzzification are:
  - bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum, etc.

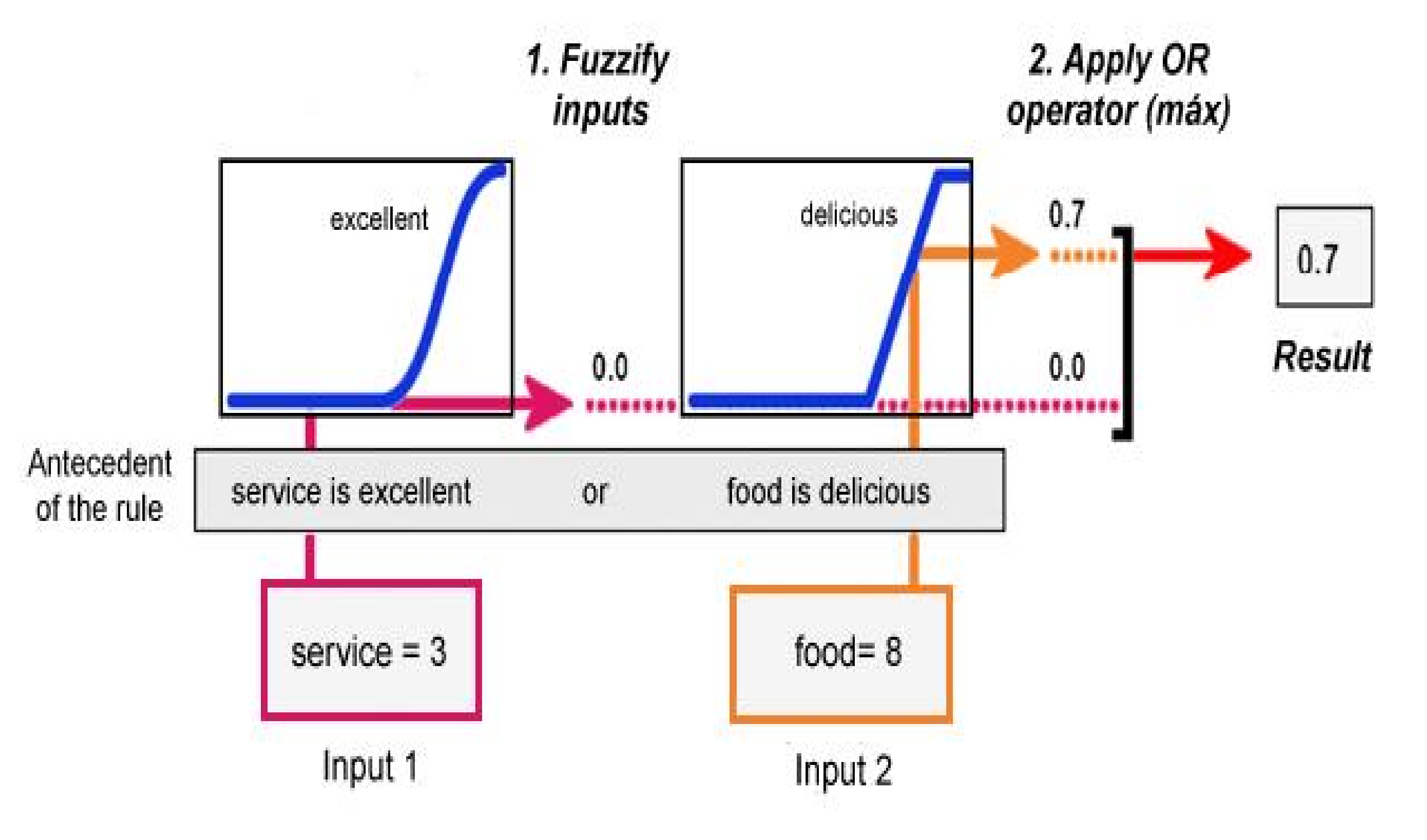
Generic Method

- Main steps are
  - Evaluate the antecedent for each rule
  - Obtain each rule's conclusion
  - Aggregate conclusions
  - Defuzzification
- We will explain these steps using an example of **Tipping Problem**
- **Two inputs** : Quality of food and Service at a restaurant rated at scale from 0-10
- **One output**: Amount of tip to be given
- Tip should reflect the quality of the food and service.
- The tip might be in the range 5-15% of total bill paid.

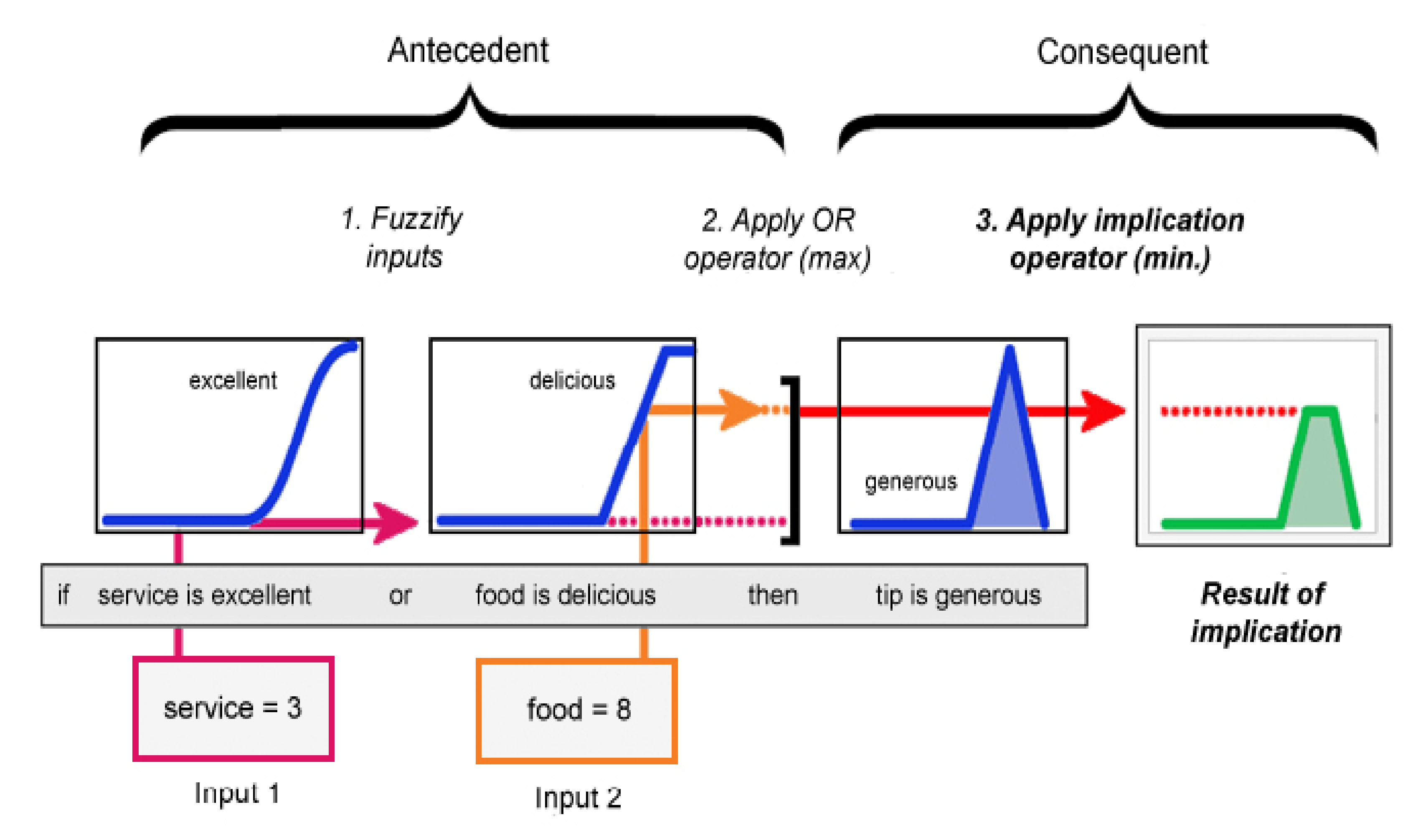
Rules for Tipping

- Let us consider the following three rules
  - If service is **poor** or food is **bad**, then tip is **cheap**
  - If service is **good**, then tip is **average**
  - If service is **excellent** or food is **delicious**, then tip is **generous**
- Input variables
  - Service : represented by poor, good, excellent
  - Food : represented by bad, delicious
- Output Variable:
  - Tip : represented by cheap, average, generous

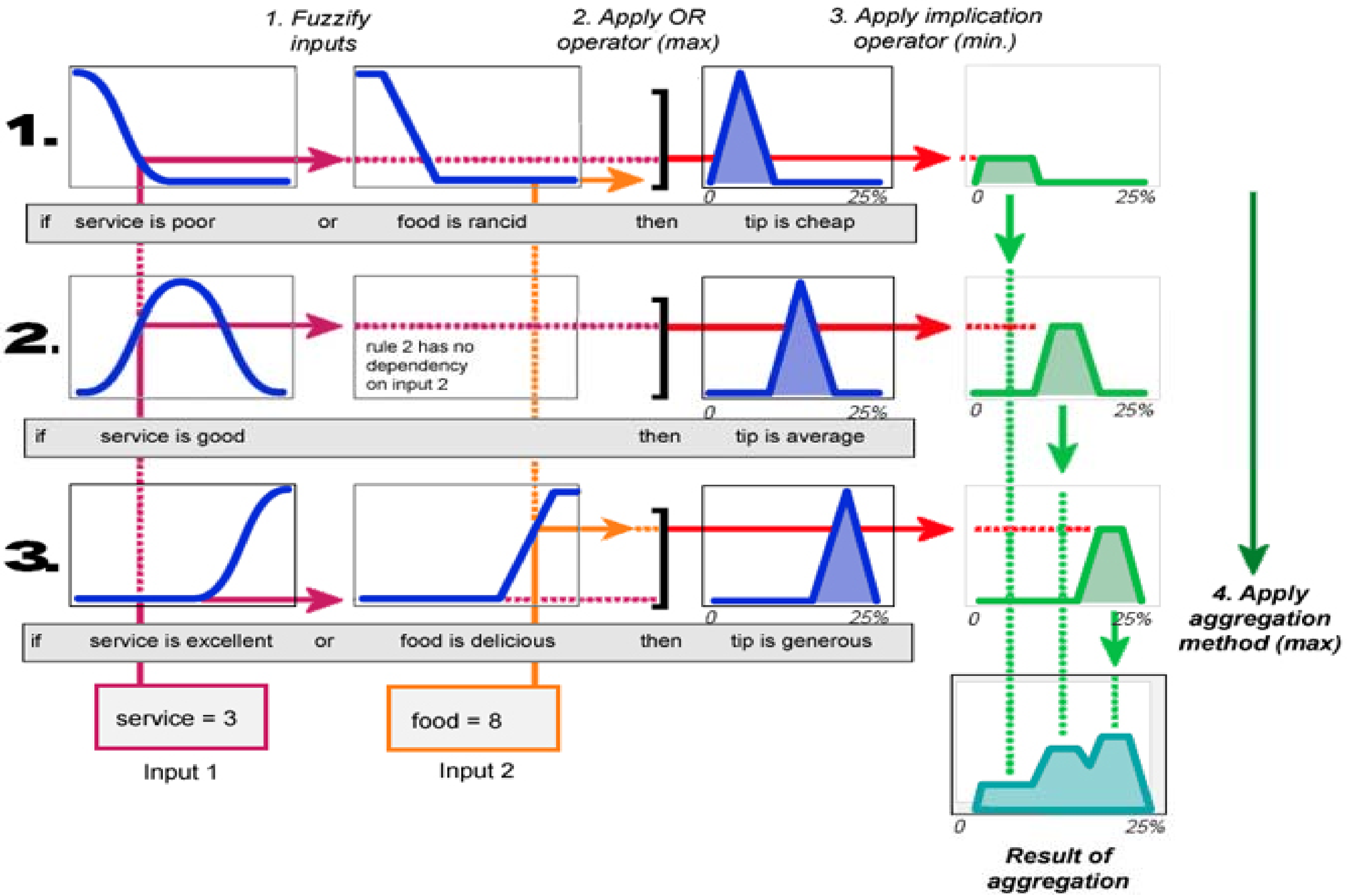
Antecedent for each rule



Rule's Conclusion


Antecedent
Consequent
1. Fuzzify inputs
2. Apply OR operator (max)
3. Apply implication operator (min.)
excellent
delicious
generous
if service is excellent or food is delicious then tip is generous
Result of implication
service = 3
food = 8
Input 1
Input 2

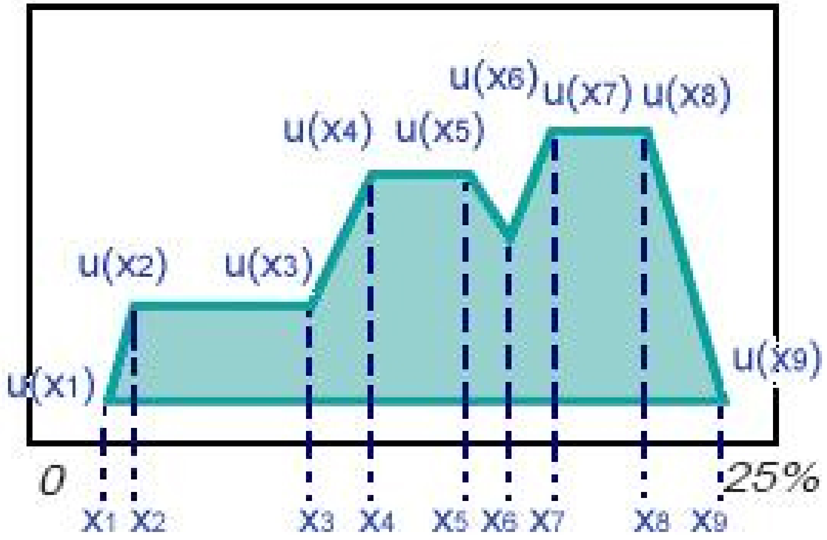
Aggregate Conclusions


1. Fuzzify inputs
2. Apply OR operator (max)
3. Apply implication operator (min.)
1.
if service is poor or food is rancid then tip is cheap
2.
rule 2 has no dependency on input 2
if service is good then tip is average
3.
if service is excellent or food is delicious then tip is generous
0
25%
service = 3
Input 1
food = 8
Input 2
4. Apply aggregation method (max)
Result of aggregation

Defuzzification

Defuzzify the aggregate output (centroid)

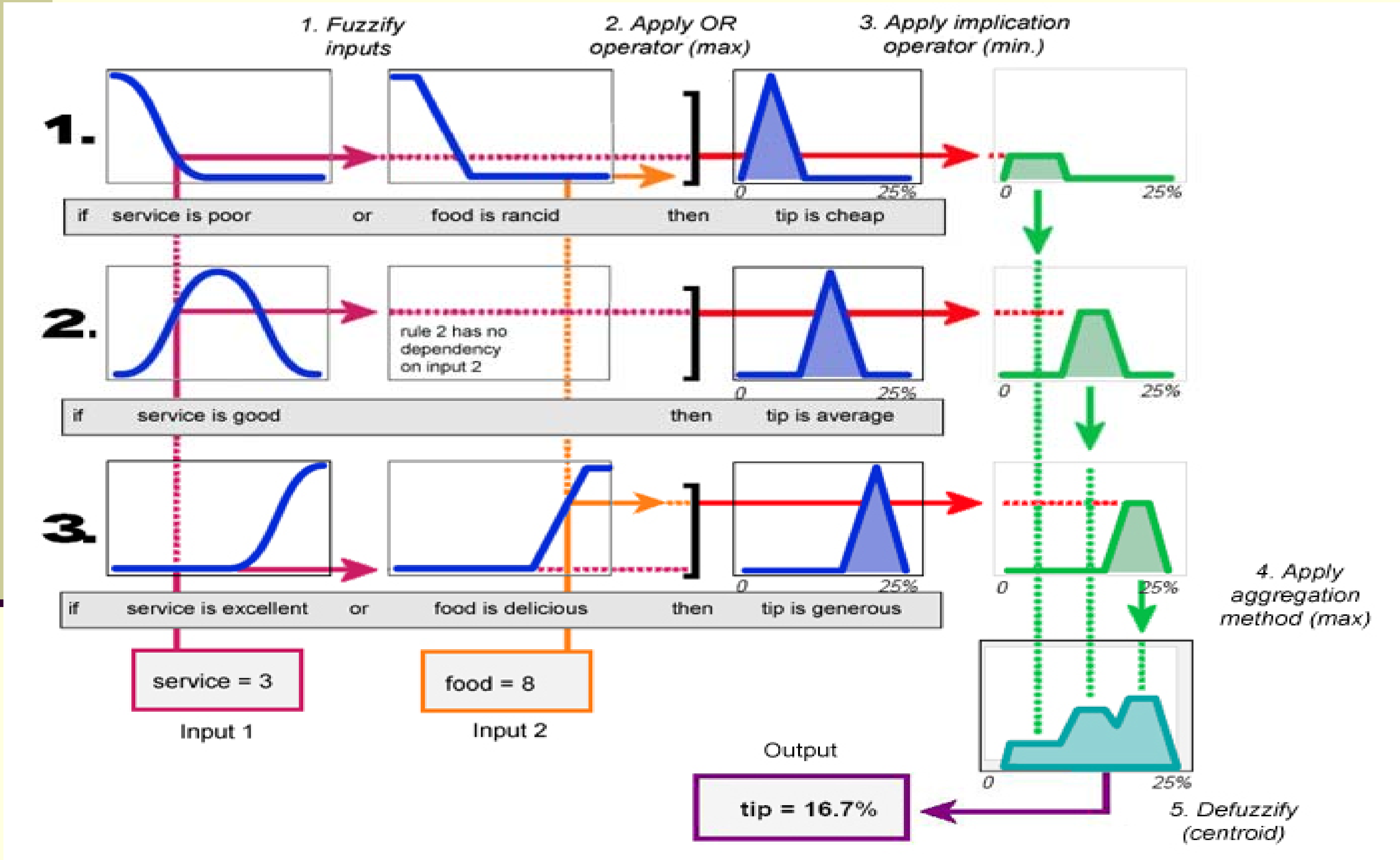
$$g = \frac{\sum_{i=1}^{9} x_i \cdot u(x_i)}{\sum_{i=1}^{9} u(x_i)} = 16{,}7$$



tip= 16,7%

Result of defuzzification

All Steps Together looks like


1. Fuzzify inputs
2. Apply OR operator (max)
3. Apply implication operator (min.)
1.
if service is poor or food is rancid then tip is cheap
0
25%
2.
rule 2 has no dependency on input 2
if service is good then tip is average
3.
if service is excellent or food is delicious then tip is generous
4. Apply aggregation method (max)
service = 3
Input 1
food = 8
Input 2
Output
tip = 16.7%
5. Defuzzify (centroid)

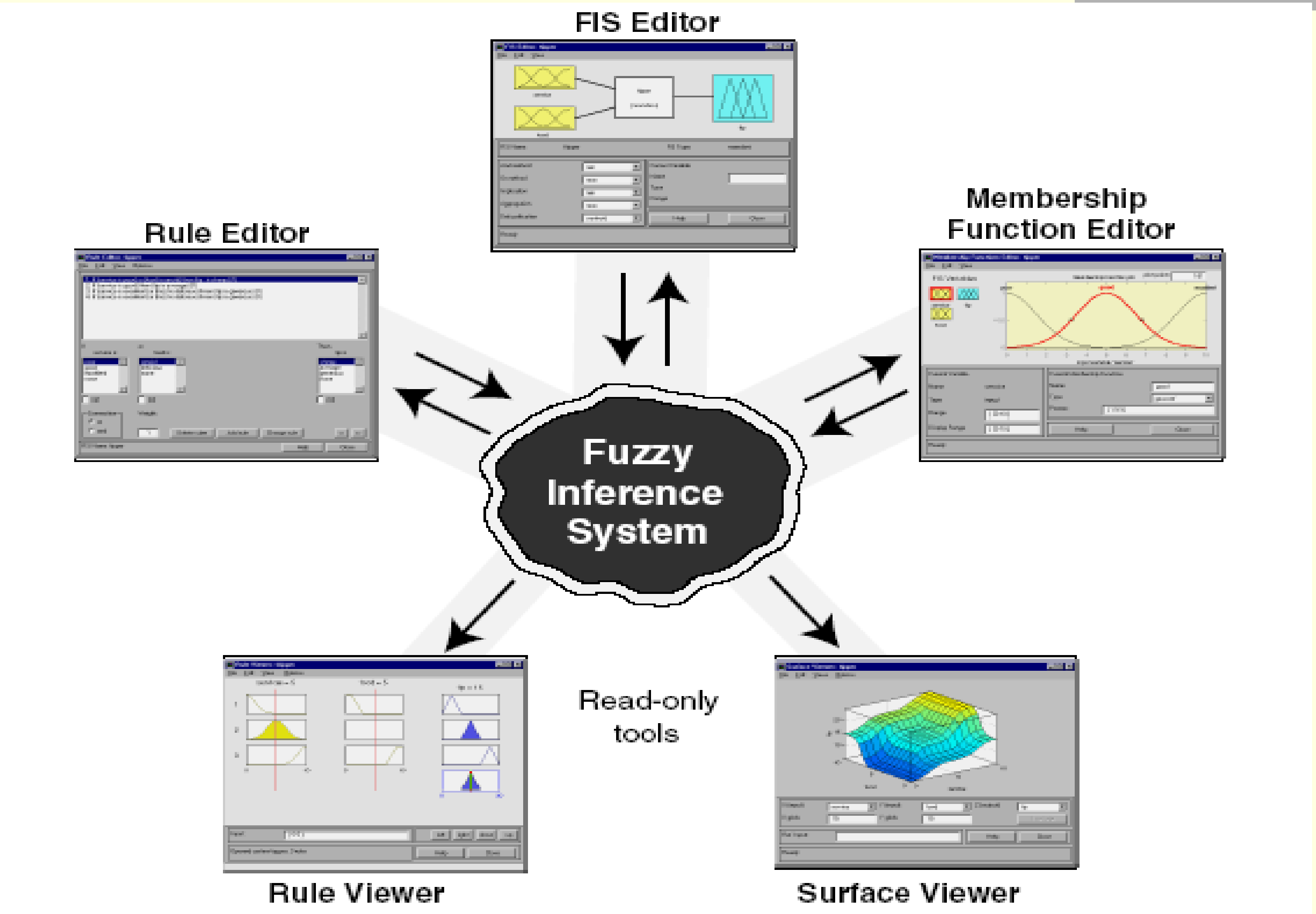
MatLab

Fuzzy Toolkit

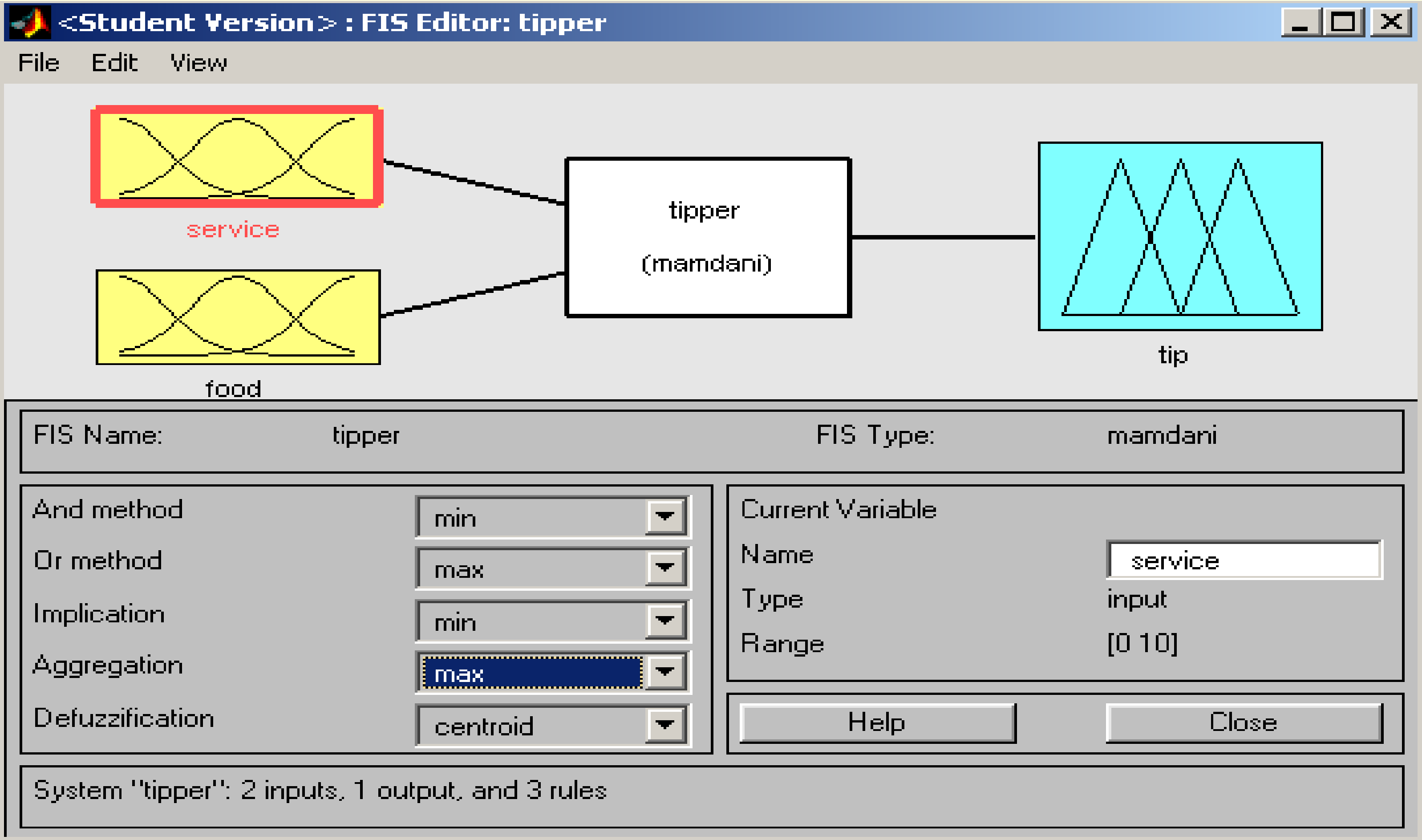
Introduction

- MATLAB fuzzy logic toolbox provides facility for the development of fuzzy-logic systems using
  - graphical user interface (GUI) tools
  - command line functionality
- There are five primary GUI tools
  - Fuzzy Inference System (FIS) Editor
  - Membership Function Editor
  - Rule Editor
  - Rule Viewer
  - Surface Viewer

GUI Tools


FIS Editor
Rule Editor
Membership Function Editor
Fuzzy Inference System
Read-only tools
Rule Viewer
Surface Viewer

Fuzzy Inference System (FIS) Editor



FIS Name: tipper

FIS Type: mamdani

And method: min

Or method: max

Implication: min

Aggregation: max

Defuzzification: centroid

Current Variable

Name: service

Type: input

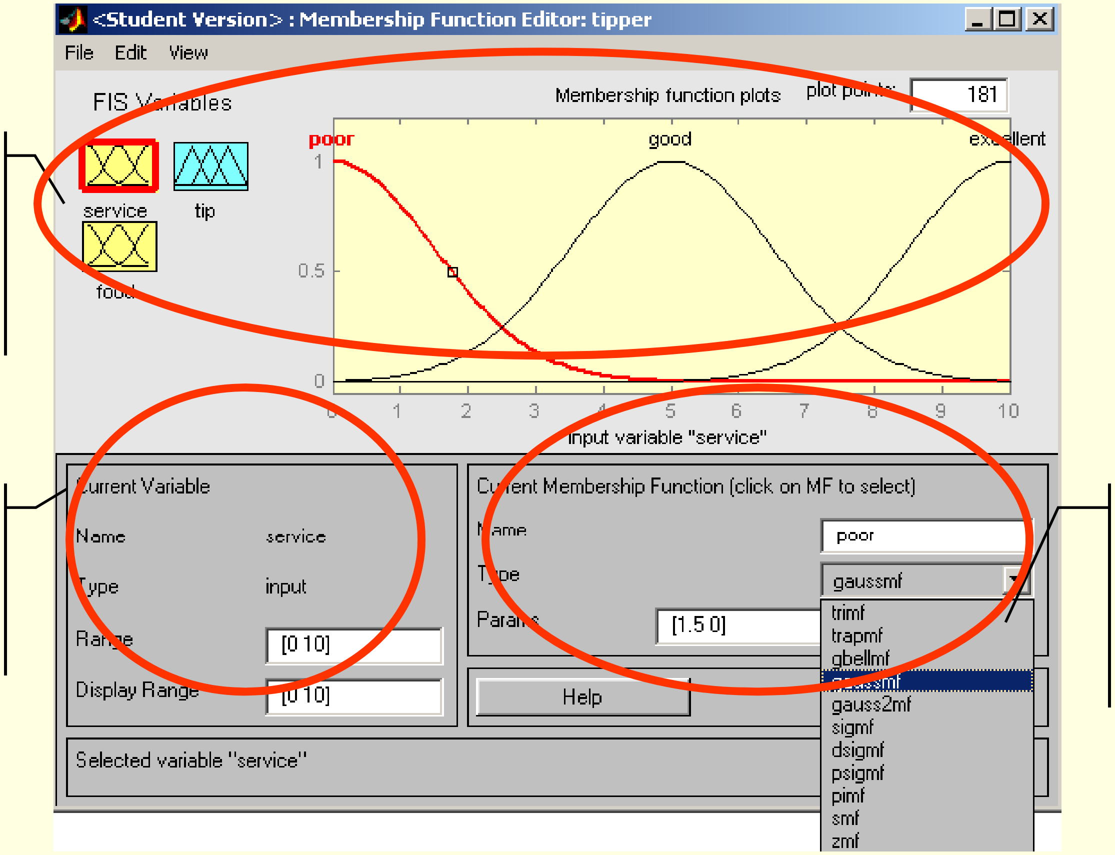
Range: [0 10]

System "tipper": 2 inputs, 1 output, and 3 rules

Membership Function Editor

Select & edit attributes of membership function

Display & edit values of current variable



plot points: 181

Current Variable

Name service

Type input

Range [0 10]

Display Range [0 10]

Current Membership Function (click on MF to select)

Name poor

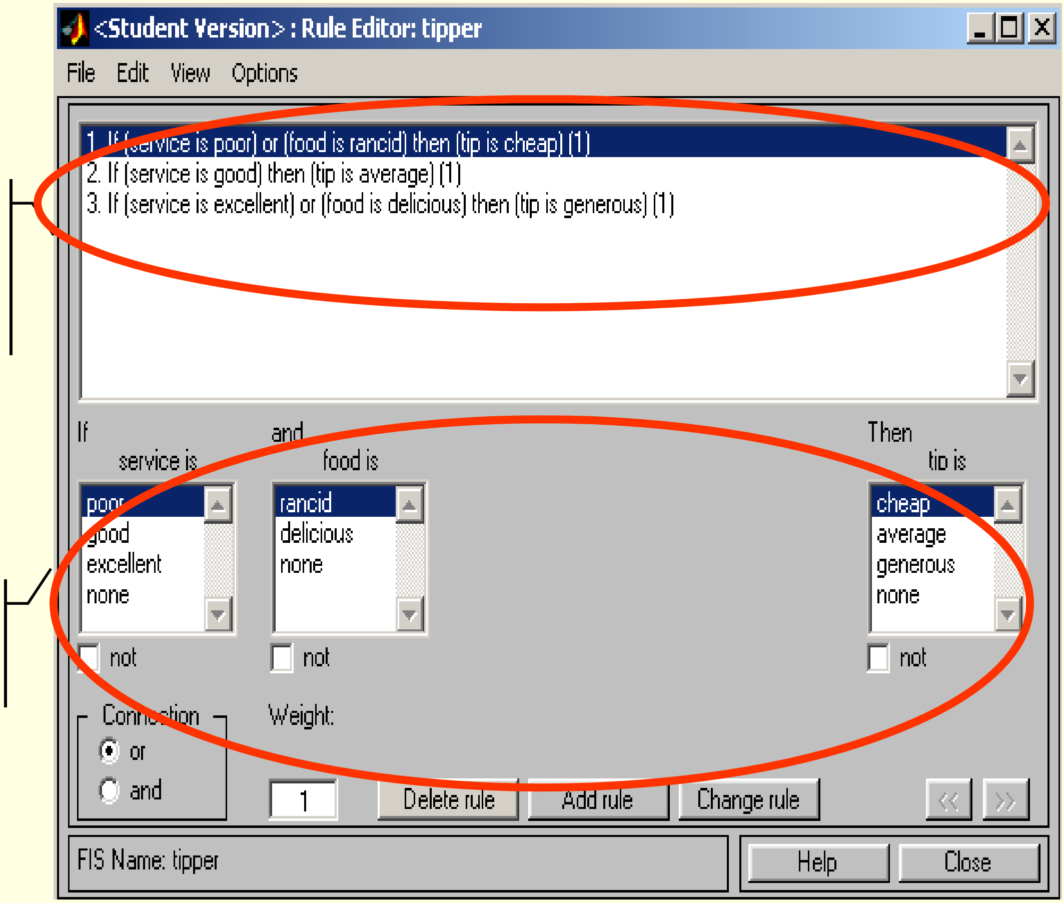
Type gaussmf

Params [1.5 0]

Selected variable "service"

Name & edit parameters of membership function

Rule Editor



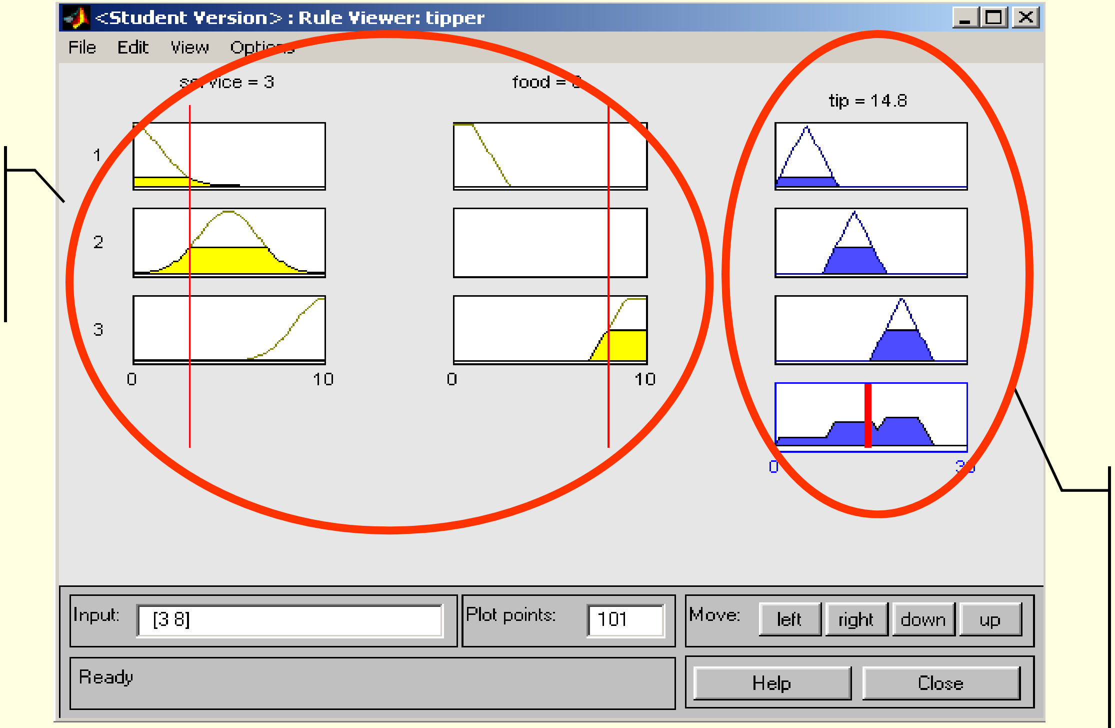
1. If (service is poor) or (food is rancid) then (tip is cheap) (1)
2. If (service is good) then (tip is average) (1)
3. If (service is excellent) or (food is delicious) then (tip is generous) (1)

Rules – automatically updated

Create and edit rules

Rule Viewer

Shows how input variable is used in rules



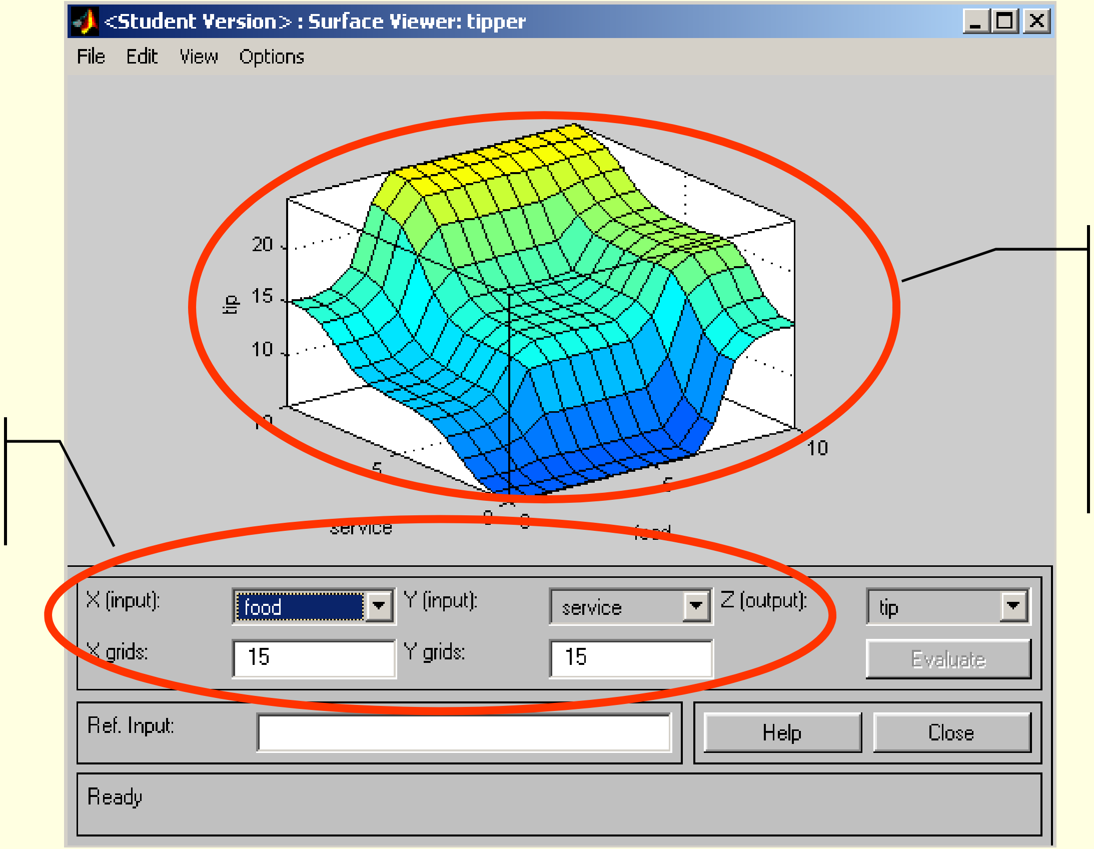
Input: [3 8]

Plot points: 101

Ready

Shows how output variable is used in rules; shows output of fuzzy system

Surface Viewer



X (input): food

Y (input): service

Z (output): tip

X grids: 15

Y grids: 15

Ref. Input:

Ready

Specify input and output variables

Shows output surface for any system output versus any one (or two) inputs