

PRÁCTICA 3

Sistemas Borrosos

Inteligencia Artificial en las Organizaciones

Grado en Ingeniería Informática

Curso 2020/2021

INTRODUCCIÓN

Cuando se habla de un sistema borroso, normalmente se hace referencia a un software que actúa de acuerdo a condiciones borrosas en la que los límites quedan definidos de forma gradual (¿qué es ser alto, bajo, joven o mayor?). El razonamiento humano hace uso habitualmente de estos términos vagos e inconcretos, por lo que cualquier sistema que deba interactuar con personas debe ser capaz de manejarlos, tanto si son la información de entrada, como si es preciso la aceptación de la decisión concluida por parte de un usuario humano.

Existen muchas herramientas de desarrollo de sistemas que manejan la incertidumbre de forma probabilística con la teoría de Dempster-Shafer, modelos bayesianos y grados de certeza, pero no integran esta posibilidad con la representación y uso de conceptos y reglas borrosas. El Fuzzy Logic Toolbox (FLT) de MATLAB sí lo hace (Figura 1). Dentro de éste, el Fuzzy Logic Designer permite, a alto nivel, manejar los principales aspectos del sistema borroso (variables de entrada/salida, nombres de variables, etc.).

En esta práctica, y con la ayuda del FLT, se pretenden desarrollar un sistema basado en lógica difusa como soporte a la decisión en la actual pandemia que se está viviendo a causa del COVID-19.

OBJETIVO

Construir un sistema basado en lógica borrosa en el ámbito del COVID cuyas reglas puedan decidir si:

- a. Dada una población debe confinarse o no.

- b. Dado un paciente con síntomas relacionados con el COVID-19 y/o contactos con algún paciente positivo, determinar si es candidato o no a realizarle el test PCR.

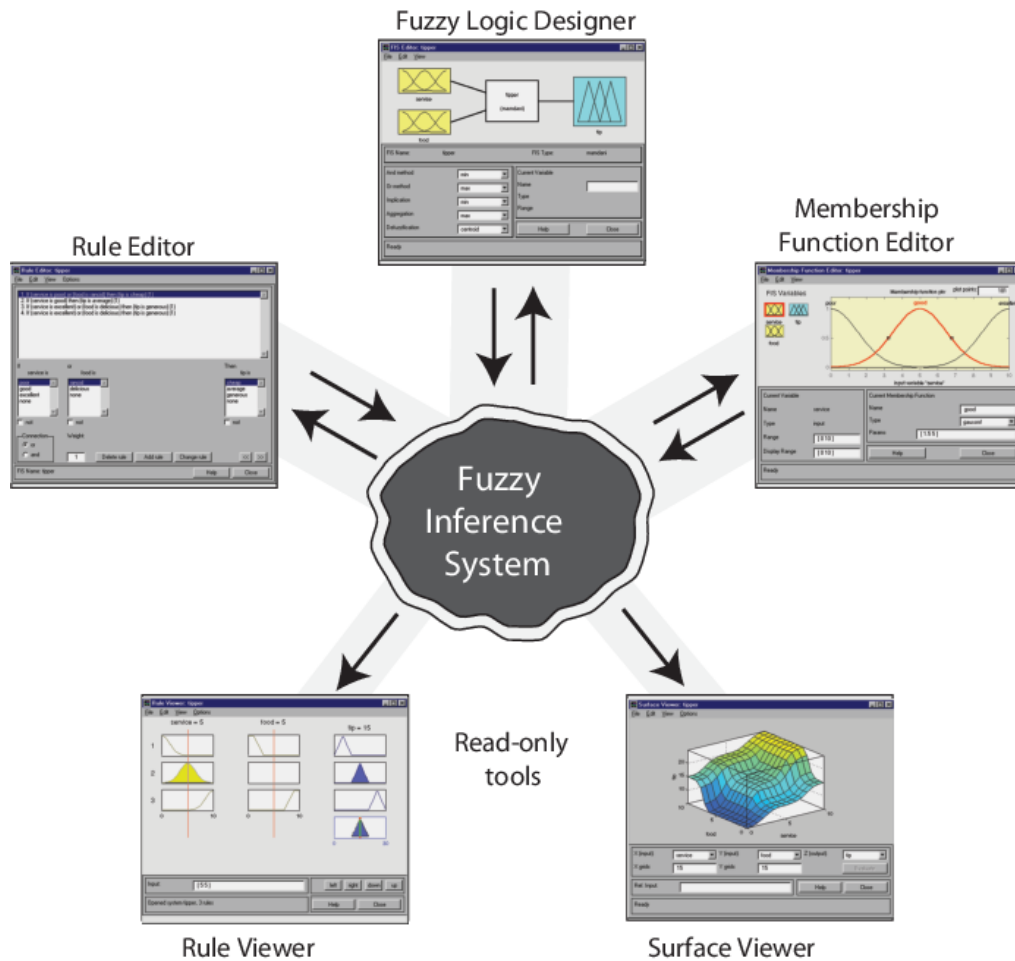


Figura 1. Componentes del Fuzzy Logic Toolbox (Fuente: mathworks.com)

Cada una de estas decisiones dependen de varias variables cuyos valores de referencia deben ser ajustados para aplicarse de forma gradual y no abrupta. En las Tablas 1 y 2, se muestran, a modo de ejemplo, algunas de las posibles variables que se pueden utilizar para los sistemas. Se pueden (y se recomienda) añadir las variables que se estimen oportunas. Dado que los datos se trabajan a dos niveles de abstracción, se recomienda abordar los dos objetivos de la práctica por separado.

Variable	Descripción
Incidencia Acumulada	Casos por 100mil habitantes
Positividad	Tanto por cien
Ocupación UCI	Tanto por cien
Capacidad de rastreo	Baja, media, alta
Capacidad de hacer test	Baja, media, alta
Obediencia de la población	Baja, media, alta
Saturación de Centros de Salud	Baja, media, alta

Tabla 1. Variables aplicables a una población

Variable	Descripción
Distancia mantenida	cm
Mascarilla puesta	Siempre/a ratos/nunca
Tipo lugar	Cerrado o abierto
Tiempo transcurrido	Minutos
Fiebre	Grados
Tos	Si/No
PCR	Positivo/Negativo/No realizado
Cansancio	Nada/Poco/Mucho
Dolor de garganta	Nada/Poco/Mucho
Perdida de olfato	Si/No

Tabla 2. Variables aplicables a un paciente

Para el desarrollo de este sistema, se deberá:

1. Definir (Fuzzy Logic Designer/Membership Function Editor)
 - a. Los conjuntos borrosos para las variables de entrada.
 - b. El conjunto borroso para la variable de salida, que también será una variable borrosa con 2 niveles de recomendación (grado de confinamiento y necesidad de PCR).
 - c. Definir las reglas para la recomendación (Rule Editor/Rule Viewer/Surface Viewer). Dichas reglas podrán aplicar términos borrosos a variables de entrada que no lo son (por ejemplo: si la ocupación de la UCI es muy alta, entonces ...).
2. Validar el sistema (Matlab):
 - a. Utilice un listado de 10 casos de poblaciones para validar el sistema.
 - b. Utilice 20 de pacientes para verificar el correcto funcionamiento del sistema.

Evaluación de la práctica

En esta práctica se deberá realizar y entregar lo siguiente:

- Los ficheros .fis con el sistema de inferencia borroso.
- Los hechos (fichero de texto con el detalle de los casos utilizados)
- Los resultados de la evaluación del sistema (fichero de texto)
- Analizar, estudiar y detallar los resultados
- Conclusiones de la práctica.

Aspectos a evaluar en la corrección de la práctica:

- Planteamiento y desarrollo del problema: 25%.
- Resultados del problema: 25%.
- Análisis de resultados y conclusiones: 25%.
- Presentación: 15%.
- Contexto de la práctica (**Información complementaria sobre el desarrollo de la práctica**): 10%.

Debe darse importancia a la presentación para mostrar los resultados, el análisis de los mismos, conclusiones, etc. Es importante tener en cuenta que el contexto de la práctica en la que se incluirá cualquier información relevante conocida por el estudiante, casos similares, noticias al respecto o cualquier otra información de interés y relacionada con la práctica.

Entrega de la práctica

- La práctica deberá realizarse en grupos de 3/4 personas (y entregarse únicamente por uno de los integrantes del grupo).
- Esta práctica no está dividida en partes. En este documento se detalla la práctica al completo. Todos los ficheros que se entregan deberán describirse brevemente en un fichero de texto.
- La entrega de esta Práctica 3 se realizará con fecha máxima de entrega (por Aula Global):
 - Grupo 83: 24 de noviembre de 2020 – 12:00h
 - Grupo 80 y 84: 26 noviembre de 2020 – 12:00h
- No hay un formato de entrega establecido. Sin embargo, es importante que toda la información se muestre de forma clara y su presentación también es considerada para la nota de la práctica.

Anexo I – Información práctica

Inicialización:

Para iniciar el Fuzzy Toolbox en Matlab:

```
>> fuzzy
```

Para guardar el sistema creado, utilice la opción “Export” del Fuzzy Logic Designer en el menú “File”.

Utilización y validación del sistema:

```
%% para cargar el sistema de inferencia borroso creado
```

```
>> fis = readfis('Mi_sistema')
```

```
%% para evaluar el sistema y almacenar la salida.
```

```
%% en este ejemplo solo hay dos variables de entrada
```

```
>> output = evalfis([500 300], fis)
```

```
%% la salida del sistema una vez defuzzificada es:
```

```
output =  
      5
```

```
%% para la evaluación de varios ejemplos a la vez
```

```
>> input = [500 100;  
           1000 500;  
           5000 500];
```

```
>> output = evalfis(input, fis)
```

```
output =  
  
      1.3560  
      5.0165  
      8.3229
```