***Abstract*** *--* Develop a system based on fuzzy rules with triangular sets that determine the solution of a problem.

**INTRODUCCIÓN**

En este artículo se llevará a cabo la explicación de un ejemplo conceptual de las leyes de fuzzy, en dónde se predecirán clientes esperados en una tienda de helados dada una temperatura exterior particular.

Es de vital importancia entender que la lógica difusa es una metodología que permite simular los procedimientos de razonamiento humano en sistemas basados en el conocimiento. En general la lógica difusa imita como una persona toma decisiones basada en información con las características mencionadas. Una de las ventajas de la lógica difusa es la posibilidad de implementar sistemas basados en ella tanto en hardware como en software o en combinación de ambos.

La teoría de la lógica difusa proporciona un marco matemático que permite modelar la incertidumbre de los procesos cognitivos humanos de forma que pueda ser tratable por un computador.

**DESARROLLO DEL CONTENIDO**

***Tema: Tienda de helados.***

***Descripción:*** Realizaremos un ejemplo de un sistema basado en lógica difusa, para predecir los clientes esperados en una tienda de helados dada una temperatura exterior particular.

Las premisas que nos indica el propietario de la heladería son las siguientes:

* Si la temperatura es alta, la heladería está llena.
* Si la temperatura es moderada, la heladería está ocupada.
* Si la temperatura es buena, la heladería está quieta.

***Solución:***1. Para iniciar a desarrollar el sistema, debemos instalar el paquete *scikit-fuzz,* de la siguiente forma:



2. Procedemos a importar las librerías.



3. Creamos las variables universales para representar la temperatura en grados Fahrenheit y los clientes para una capacidad de 35 personas. Utilizando la función “arange” de numpy, que permite generar un array con rango de valores, enviando como parámetros el inicio del intervalo, el fin del intervalo y el espaciado entre valores.



4. Creamos las funciones difusas de membrecía que representen el grado de verdad relacionado con una variable continua.

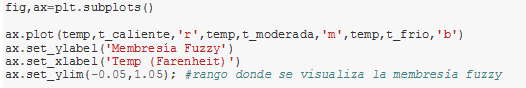
- Para las variables de temperaturas caliente, moderada y fría, utilizamos la instrucción “trimf” de skfuzzy, que permite generar una función triangular, enviando como parámetros el array de la temperatura y la matriz de longitud.

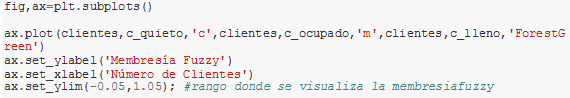


- Para las variables de clientes lleno, moderado y quieto, utilizamos la instrucción “trimf” de skfuzzy, que permite generar una función triangular, enviando como parámetros el array de los clientes y la matriz de longitud.



5. Luego para visualizar las funciones de membresía de la temperatura y de los clientes, usamos el código:



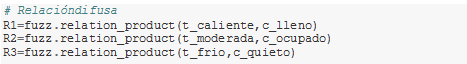


Con la instrucción “fig,ax” estamos indicando que crearemos una figura con un objeto y que utilizaremos el diseño “subplots”.

Posteriormente indicamos que vamos a generar un gráfico de líneas con la función “plot”, enviándole como argumento las variables de entrada tanto de la temperatura como de los clientes, con un respectivo color para cada línea del gráfico.

Adicionalmente utilizamos las funciones, “set\_ylabel” y “set\_xlabel” para nombrar con unas etiquetas el eje “Y” y el eje “X”.

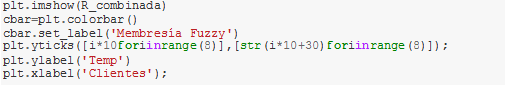
5. Creamos las reglas que son necesarias para cumplir con las premisas iniciales. Utilizando la función “relation\_product” que permite determinar la matriz de relación difusa, indicando que la “Regla1” será cuando la temperatura esté caliente y la heladería llena de clientes; la “Regla2” será cuando la temperatura esté moderada y la heladería esté ocupada de clientes y la “Regla3” será cuando la temperatura esté fría y no haya clientes en la heladería.



Con la siguiente instrucción estamos comparando dos matrices y nos devolverá una nueva matriz que contiene los máximos elementos de las reglas 1,2 y 3.



Para visualizar la nueva matriz con los valores máximos de nuestras reglas, utilizamos la función “imshow” que permite indicar que el diseño del gráfico será en 2D. Luego con la función “colorbar” indicamos que vamos a agregar una barra de color al diagrama y con la instrucción “set\_label” le asigno un nombre a la barra de color. Con la función “yticks” queremos obtener las ubicaciones actuales del eje “Y”, y enviamos como parámetro un for en donde están la lista de posiciones.



**DESCRIPCIÓN DE LOS RESULTADOS**

A continuación se explicarán las gráficas que nos generó el código expuesto anteriormente, primero visualizáramos la gráfica de la temperatura, posteriormente la de los clientes y finalmente la gráfica de nuestras reglas, que fueron definidas al inicio de éste artículo.

***Fig. 1*** Se asigna un 1 a los valores que pertenecen al conjunto especificado y se asigna el rango completo de posibilidades, entre 0 y 1, cuando existe algún nivel de posible pertenencia (cuanto mayor sea el número, mayor será la posibilidad), esta grafica podemos observar como la línea roja comienza a incrementar su valor en la posición de 65 grados Fahrenheit que representa la temperatura cuando esta alta, así mismo podemos observar las demás líneas.

La línea roja de la imagen que se muestra a continuación representa una transformación lineal de pendiente positiva con un mínimo de 65 grados y un máximo de 100 grados. A todo valor menor que 65 se asignará un 0 y a todo valor mayor que 100, un 1.

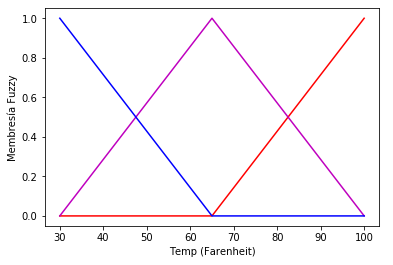
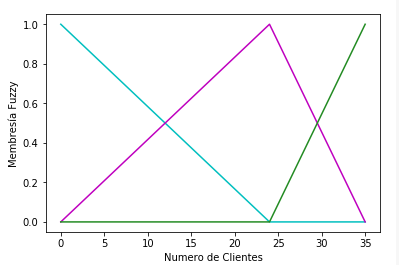


Fig. 1 Gráfico No. 1

***Fig. 2*** Se asigna un 1 a los valores que pertenecen al conjunto especificado y se asigna el rango completo de posibilidades, entre 0 y 1, cuando existe algún nivel de posible pertenencia (cuanto mayor sea el número, mayor será la posibilidad), En esta grafica podemos observar la línea verde como comienza a incrementar cuando llega al valor de 24 clientes, esta línea nos indica que el establecimiento se está llenado.  
La línea verde de la imagen que se muestra a continuación representa una transformación lineal de pendiente positiva con un mínimo de 24 clientes y un máximo de 35 clientes grados. A todo valor menor que 24 se asignará un 0 y a todo valor mayor que 35, un 1.

  
Fig. 2 Gráfico No. 2

***Fig. 3*** En esta grafica podemos observar cómo mayor temperatura hay mayor posibilidad según la escala de membresía fuzzy de que ingresen personas al establecimiento, también nos muestra cómo según la relación de t\_caliente-c\_lleno, t\_moderada-c\_ocupado, t\_frio-c\_quieto, tornan de color amarillo en la gráfica según la membresía fuzzy.

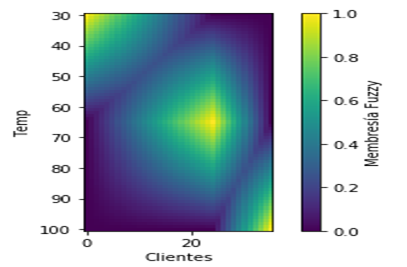


Fig. 3 Gráfico No. 3

**CONCLUSIONES**

1. La lógica difusa es diferente de probabilidad. Con probabilidad, intentamos determinar algo sobre el resultado potencial de los acontecimientos definidos que pueden ocurrir al azar. Con lógica difusa, determinamos algo sobre la naturaleza del acontecimiento.
2. Diseñar sistemas difusos es fácil y rápido. Los sistemas difusos reflejan lo que la gente piensa realmente de un problema. Posteriormente, después de una prueba o experiencia, se puede ajustar el sistema con características exactas. Con los sistemas basados en la lógica difusa se pueden evaluar mayor cantidad de variables, simulando el conocimiento humano.
3. Los sistemas borrosos son estables y fácilmente ajustables y pueden ser validados. Es más rápido crear sistemas borrosos que crear sistemas basados en el conocimiento convencionales, puesto que la lógica difusa maneja todos los grados de libertad que intervienen. Se relaciona entradas y salidas, sin tener que entender todas las variables, permitiendo que el sistema pueda ser más confiable.
4. La lógica difusa es una herramienta de gran alcance y versátil para la información imprecisa, ambigua y vaga de la representación. No puede solucionar todos los problemas, pero nos ayuda a modelar problemas difíciles, incluso intratables.
5. Se simplifica también la adquisición y representación del conocimiento y unas pocas reglas abarcan gran cantidad de complejidades.