

MEMORIA: Construcción de un sistema de clasificación basado en reglas difusas

Índice:

1. Introducción y objetivos:	pág 3
2. Generación de reglas:	pág 4
3. Clasificación:	pág 9
4. Modificaciones del algoritmo:	pág 10
5. Conclusión final del trabajo:	pág 16

Introducción y objetivos:

En la práctica final se ha construido un sistema de clasificación basado en reglas difusas. Antes de ello, se han tenido que crear los conjuntos de las variables difusas y las reglas generadas por el conjunto de datos proporcionado. El conjunto de datos se llama Iris y contiene 150 ejemplos de flores formados por cuatro variables numéricas (largo sépalo, ancho sépalo, largo pétalo y ancho pétalo). Cada ejemplo corresponde a una de las 3 clases (denominadas Iris setosa, Iris versicolor e Iris virginica).

Las reglas se generarán a partir del conjunto de datos de entrenamiento, y la clasificación se realizará con los datos finales para el testeo. Para las reglas, es necesaria la creación previa de los conjuntos bajo, medio y alto de cada una de las cuatro variables. Una vez se tienen los conjuntos y se obtiene el índice correspondiente a cada conjunto, se podrá obtener el grado de pertenencia.

Como puede haber reglas redundantes (reglas con mismos antecedentes, pero distinta clase o grado de certeza), se tendrán que eliminar para únicamente escoger las reglas no redundantes (se queda la regla con mayor grado de certeza).

Una vez que se obtiene el conjunto de reglas totales, se debe calcular el grado de asociación de cada ejemplo del conjunto final con cada una de las respectivas reglas. Para cada clase, se escogerá el mayor grado de asociación que tenga ese ejemplo. Finalmente, el argumento de la clase con mayor grado de asociación será el propuesto por el clasificador. Como se ofrece una matriz con las etiquetas de cada ejemplo, se podrá comprobar si la clase propuesta por el clasificador corresponde con la clase real de su etiqueta.

Generación de reglas:

Como se ha explicado anteriormente, para construir las reglas es necesario haber obtenido de forma previa los conjuntos. Se proporciona un dataset que es una matriz de 4 columnas correspondiente al número de variables, y con tantas filas como número de ejemplos hay (150).

En primer lugar, tendremos que crear 4 conjuntos, uno para cada variable, y cada conjunto será obtenido a partir de una columna del conjunto de datos. Como no se proporciona el referencial de cada conjunto, es necesario calcular cuál es el mínimo valor y el máximo valor que puede tomar en el conjunto. Una vez se obtienen ambos valores, el enunciado explica que la precisión son los decimales. Con lo cual, se tomarán todos los valores desde el mínimo al máximo yendo de 0.1 en 0.1.

En segundo lugar, se utilizan 3 etiquetas lingüísticas (bajo, medio, alto). Por lo tanto, cada uno de los 4 conjuntos tendrá 3 etiquetas lingüísticas y, por tanto, otros 3 conjuntos para cada etiqueta.

Para la realización de cada conjunto se ha usado una función triangular. El conjunto bajo comienza teniendo valor máximo en el mínimo del referencial hasta llegar al punto medio, donde tomaría valor nulo hasta el máximo del referencial. El conjunto medio comienza y finaliza en 0, pero tiene su valor máximo en el punto medio del referencial. Por último, el conjunto alto comienza en 0 hasta el punto medio, donde empezaría a incrementar el valor hasta alcanzar el pico en el máximo del referencial. Se escoge el punto medio como la división entre 2 de la suma entre el máximo y mínimo.

En tercer lugar, se va a explicar el por qué de los argumentos de la tabla de la función *trimf* para la creación de cada conjunto triangular. Para el conjunto bajo, se tiene que introducir en los dos primeros argumentos el mínimo para que el primer valor del referencial tome el valor 1, y a continuación, ese valor decrezca hasta el punto medio. Para el conjunto medio, simplemente hay que introducir los argumentos de forma general siendo ambos 3 distintos para que el segundo argumento tome el valor máximo en dicho punto. Para finalizar, el conjunto final deberá tener segundo y tercer argumento igual para mostrar que el conjunto tiene que finalizar con el máximo valor.

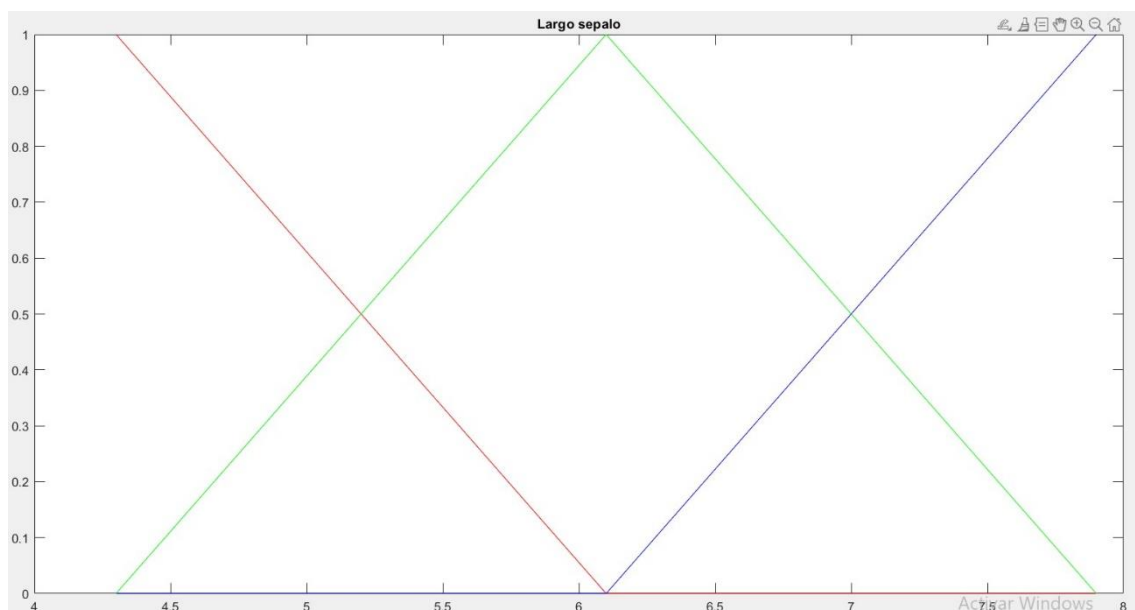
A continuación, se tiene que construir cada una de las reglas. Para ello, se utiliza la función *find* para calcular el índice de cada conjunto, y a través del índice, calcular para cada conjunto cual es la etiqueta a la que pertenece en mayor medida. Para facilitar dicho cálculo, se introducen los valores en un vector para que la función *sort* los ordene de mayor a menor y así poder obtener el índice de la etiqueta correspondiente.

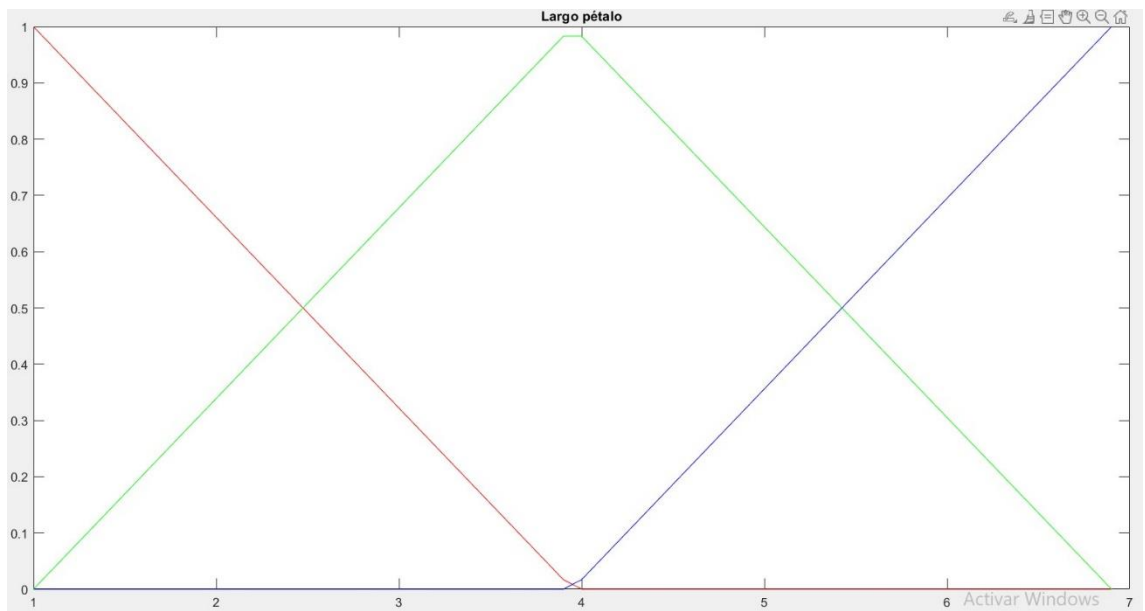
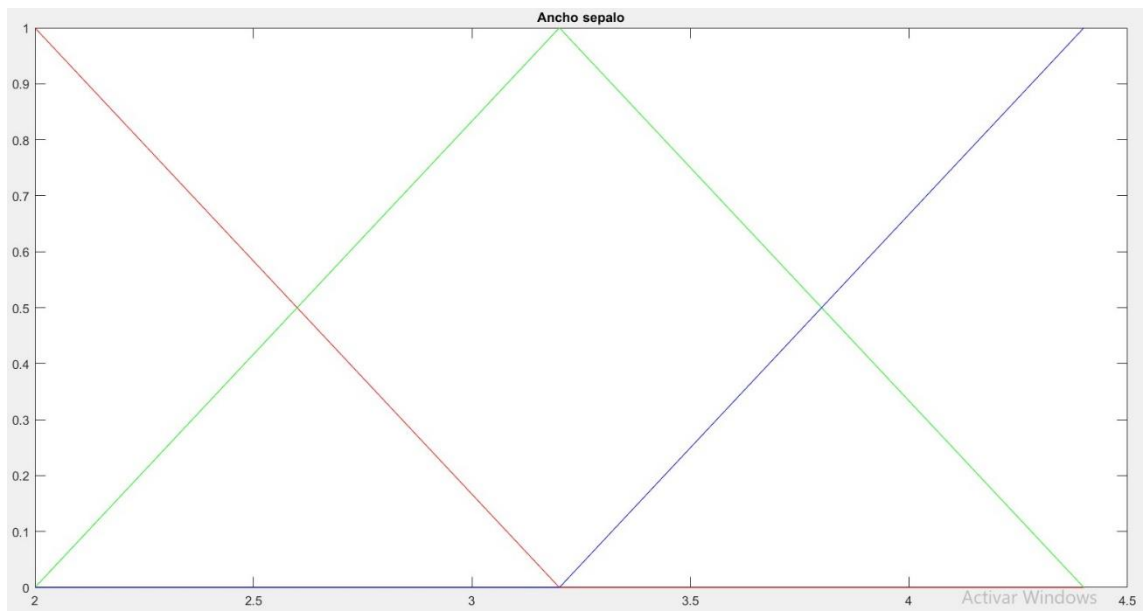
En cuarto lugar, se creará la matriz de reglas. La matriz de reglas tendrá tantas filas como número de ejemplos, y 6 columnas. Las primeras 4 columnas tendrán el índice de la etiqueta de cada conjunto. La quinta columna será el grado de certeza de la regla, y la

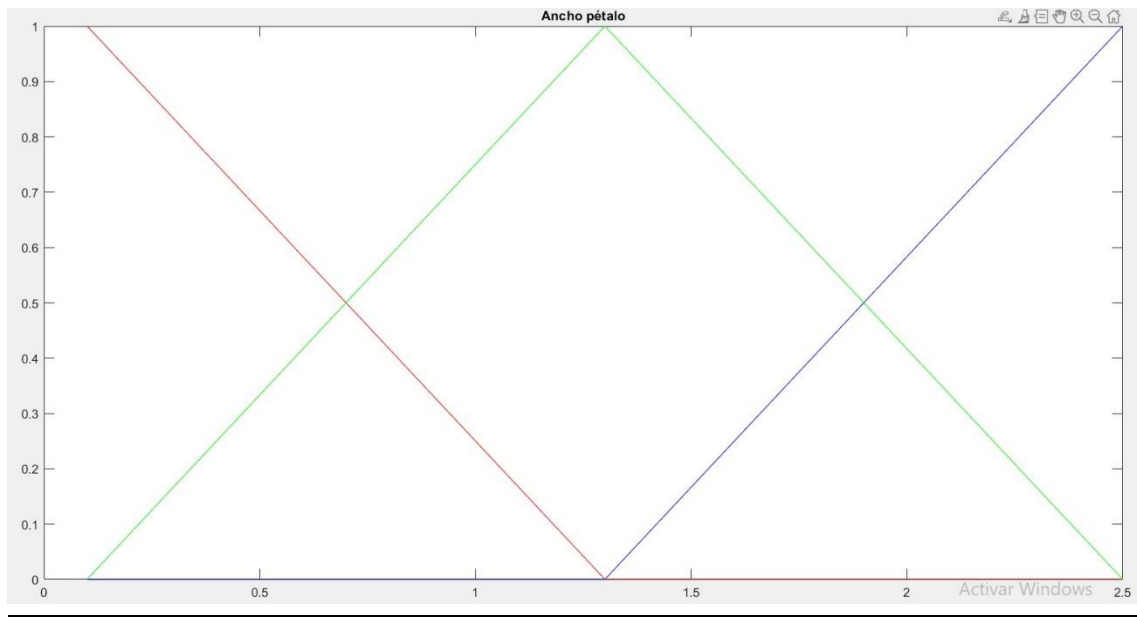
sexta columna será la clase. La clase se obtendrá de la matriz de las etiquetas proporcionada.

Por último, se tienen que eliminar las reglas redundantes. Para ello, hay que fijarse en las combinaciones repetidas de las primeras 4 columnas de cada fila. La función *unique* devuelve un vector columna con el mismo número de filas que la matriz previa. Lo bueno que proporciona *unique* es que todas aquellas 4 columnas que tengan la misma combinación en la matriz reglas, tendrán el mismo valor en el vector del tercer argumento. Además, se toman valores desde el 1 hasta el número máximo de combinaciones de las 4 columnas diferentes. Por lo tanto, se recorre cada una de las combinaciones viendo si dicha combinación está repetida. Si la combinación está repetida y el grado de certeza es mayor que el de otra misma combinación, se guardará el grado de certeza máximo y su clase correspondiente.

Para mostrar por pantalla las reglas se utiliza un diccionario para asociar 1 con bajo, 2 con medio y, 3 con alto. De esta forma, será sencillo imprimir por pantalla las reglas con cada etiqueta de los antecedentes, cada clase, y cada grado de certeza. El conjunto bajo se representa en rojo, el conjunto medio en verde y el conjunto alto en azul.







```
>> main
El número de reglas es 14 y son las siguientes:
IF Largo Sépalo IS BAJO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS MEDIO AND
Ancho Pétalo IS MEDIO
THEN Clase 2 con Certeza = 0.39

IF Largo Sépalo IS BAJO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS BAJO AND
Ancho Pétalo IS BAJO
THEN Clase 1 con Certeza = 0.81

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS MEDIO AND
Ancho Pétalo IS MEDIO
THEN Clase 2 con Certeza = 0.58

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS MEDIO AND
Ancho Pétalo IS ALTO
THEN Clase 3 con Certeza = 0.17

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND
Ancho Pétalo IS MEDIO
THEN Clase 3 con Certeza = 0.14

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS BAJO AND
Ancho Pétalo IS BAJO
THEN Clase 1 con Certeza = 0.36

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS MEDIO AND
Ancho Pétalo IS MEDIO
THEN Clase 2 con Certeza = 0.66

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS MEDIO AND
Ancho Pétalo IS ALTO
THEN Clase 3 con Certeza = 0.38

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS ALTO AND
Ancho Pétalo IS MEDIO
THEN Clase 3 con Certeza = 0.22

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS ALTO AND
Ancho Pétalo IS ALTO
THEN Clase 3 con Certeza = 0.57

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS ALTO AND Largo Pétalo IS BAJO AND
Ancho Pétalo IS BAJO
THEN Clase 1 con Certeza = 0.48

IF Largo Sépalo IS ALTO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND Ancho
Pétalo IS ALTO
THEN Clase 3 con Certeza = 0.37

IF Largo Sépalo IS ALTO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS ALTO AND
Ancho Pétalo IS MEDIO
THEN Clase 3 con Certeza = 0.23
```

Clasificación:

Para el proceso de clasificación se ha creado un clasificador basado en reglas difusas. Es necesario disponer tanto de los conjuntos de datos y etiquetas de testeo como de las reglas y conjuntos difusos. Para obtener la precisión del sistema de clasificación se tiene que recorrer cada uno de los ejemplos del conjunto de datos para comprobar si a través de las reglas, se infiere una etiqueta igual a la asignada en el conjunto de datos dataset. Y.

Para ello, se recorre cada ejemplo y se calcula el grado de asociación de cada ejemplo con cada regla. El grado de asociación se calcula mediante una multiplicación del grado de compatibilidad (t-norma entre el valor asociado de cada atributo del ejemplo con el conjunto) y el grado de certeza de la regla. Para cada clase se guarda el mayor grado de asociación. Una vez que se ha obtenido el máximo valor de cada clase, se escogerá la clase de las tres que tenga mayor grado de asociación. Dicha clase será la inferida por el sistema de clasificación para el ejemplo dado.

Para finalizar, se consulta que cantidad de ejemplos son analizados correctamente comparando el resultado del clasificador con su etiqueta correspondiente. Si coincide, el resultado es correcto. Además, se crea una matriz de confusión con la etiqueta real en las filas, y con la etiqueta inferida en las columnas. De esta manera, se introducirá cada ejemplo analizado para visualizar de forma sencilla cuáles son los ejemplos y de que tipo son con los que suele fallar más el clasificador.

MATLAB Command Window

Page 2

```
IF Largo Sépalo IS ALTO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS ALTO AND
Ancho Pétalo IS ALTO
THEN Clase 3 con Certeza = 0.42
```

```
El sistema de clasificación nos da una accuracy de 0.89333(89.3333%).
```

```
La matriz de confusión es:
```

25	0	0
0	25	0
0	8	17

```
>>
```

Como se puede observar, la precisión es de aproximadamente 89%. Cada 100 ejemplos, el sistema de clasificación únicamente fallará en 11 de ellos. Nos encontramos ante un sistema con bastante precisión, en el que a pesar de que las reglas no tengan un alto grado de certeza, se infiere la clase de forma precisa en la mayoría de los casos.

Gracias a la matriz de confusión se puede inferir cuáles son las clases de los ejemplos en los que falla en mayor medida el clasificador. Todos aquellos ejemplos cuyo resultado de la clase sea el correcto, se encontrarán en la diagonal principal porque la fila y la columna será la misma (esto es porque el valor real (fila) será igual al valor inferido (columna)).

Prácticamente todos los valores de la matriz que no estén en la diagonal principal son valores nulos. Únicamente hay una excepción que se encuentra en la posición (3, 2). Además, el 32% de los ejemplos de la clase 3 se infieren de forma incorrecta (8 de 25), y todos ellos son inferidos para la misma clase (clase 2). También, se puede observar que hay 25 ejemplos por cada clase y que todos los ejemplos de la clase 1 y 2 se infieren correctamente.

De las 14 reglas obtenidas previamente, solamente tenemos 3 para la clase 1 y otras 3 para la clase 2. Por lo tanto, más de la mitad de las reglas están destinadas a la clase 3. Para una mayor precisión es probable que se tenga que aumentar el número de reglas destinadas a la clase 2 para encontrar alguna regla con alto grado de certeza que aumente la precisión de la clase 2.

Modificaciones del algoritmo:

Se van a realizar varias modificaciones al algoritmo. Cada una de las modificaciones se encontrará comentada en el código principal.

En primer lugar, se va a utilizar una función de pertenencia trapezoidal para cada uno de los 4 conjuntos.

```
conjuntos.A = [trapmf(conjuntos.R_A, [minimoA, minimoA, (minimoA + maximoA)/2, maximoA]);
               trapmf(conjuntos.R_A, [minimoA, 5.5, 6.5, maximoA]);
               trapmf(conjuntos.R_A, [minimoA, (minimoA + maximoA)/2, maximoA, maximoA]);

conjuntos.B = [trapmf(conjuntos.R_B, [minimoB, minimoB, (minimoB + maximoB)/2, maximoB]);
               trapmf(conjuntos.R_B, [minimoB, 3, 3.5, maximoB]);
               trapmf(conjuntos.R_B, [minimoB, (minimoB + maximoB)/2, maximoB, maximoB]);

conjuntos.C = [trapmf(conjuntos.R_C, [minimoC, minimoC, (minimoC + maximoC)/2, maximoC]);
               trapmf(conjuntos.R_C, [minimoC, 3.5, 4.5, maximoC]);
               trapmf(conjuntos.R_C, [minimoC, (minimoC + maximoC)/2, maximoC, maximoC]);

conjuntos.D = [trapmf(conjuntos.R_D, [minimoD, minimoD, (minimoD + maximoD)/2, maximoD]);
               trapmf(conjuntos.R_D, [minimoD, 1, 1.5, maximoD]);
               trapmf(conjuntos.R_D, [minimoD, (minimoD + maximoD)/2, maximoD, maximoD]);
```

En segundo lugar, se va a modificar la t-norma que se utiliza para calcular el grado de certeza de las reglas. En lugar de utilizar la t-norma producto, se va a utilizar la t-norma mínimo. Para ello, se añade todos los valores en un vector para simplificar el cálculo.

```
vectorGradoCerteza = [valores1(1) valores2(2) valores3(1) valores4(1)];  
reglas(i,5) = min(vectorGradoCerteza);
```

En tercer lugar, se cambia la t-norma aplicada para calcular el grado de compatibilidad de cada ejemplo con la regla correspondiente. En vez de utilizar la t-norma producto, se va a utilizar la t-norma de Lukasiewicz ($L(x, y) = \max(0, x+y-1)$). Como son 4 valores por cada conjunto, se aplica la propiedad asociativa de la t-norma y así, ir calculando de 2 en 2 el grado de asociación final.

```
compatibilidad(i, j) = max(0, valorLargoPetal+valorAnchoPetal-1);  
compatibilidad(i, j) = max(0, compatibilidad(i, j)+valorAnchoSepalo-1);  
compatibilidad(i, j) = max(0, compatibilidad(i, j)+valorLargoSepalo-1);
```

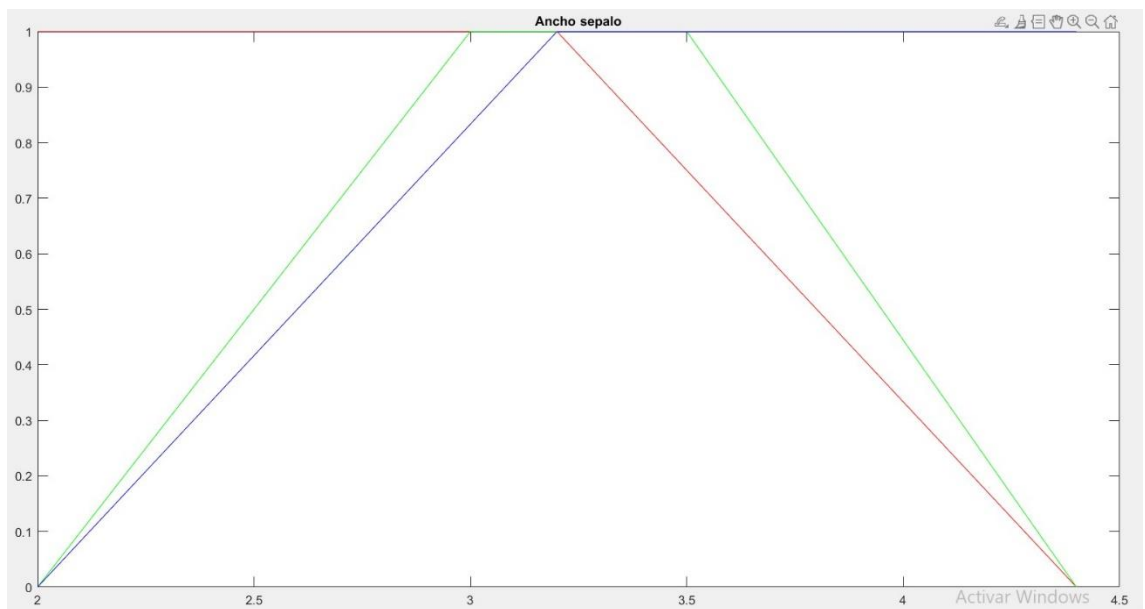
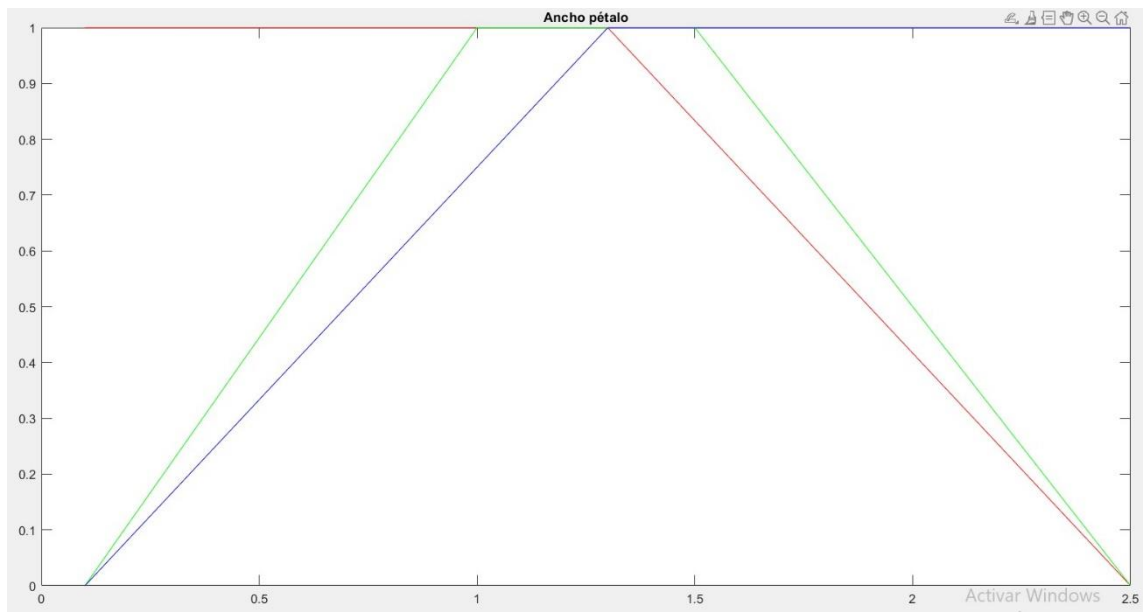
En cuarto lugar, se va a utilizar como función para calcular la asociación la implicación de Zadeh ($I_{Goedel}(A,B)=\min(1,A+B)$).

```
asociacion(i, j) = min(1, compatibilidad(i, j) + reglas(j,4));
```

Finalmente, vamos a modificar la función de agregación por clases para utilizar el máximo.

```
if reglas(j, 6) == 1 && asociacion(i, j) > maximoClases(1)  
    maximoClases(1) = asociacion(i, j);  
  
elseif reglas(j, 6) == 2 && asociacion(i, j) > maximoClases(2)  
    maximoClases(2) = asociacion(i, j);  
  
elseif reglas(j, 6) == 3 && asociacion(i, j) > maximoClases(3)  
    maximoClases(3) = asociacion(i, j);  
  
end
```

A continuación, se mostrarán los resultados con todas las modificaciones realizadas anteriormente.



```
>> main
El número de reglas es 21 y son las siguientes:
IF Largo Sépalo IS BAJO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS BAJO AND Ancho Pétalo IS BAJO
THEN Clase 1 con Certeza = 1.00

IF Largo Sépalo IS BAJO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS BAJO AND Ancho Pétalo IS MEDIO
THEN Clase 2 con Certeza = 0.70

IF Largo Sépalo IS BAJO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS MEDIO AND Ancho Pétalo IS BAJO
THEN Clase 2 con Certeza = 0.80

IF Largo Sépalo IS BAJO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS MEDIO AND Ancho Pétalo IS MEDIO
THEN Clase 2 con Certeza = 1.00

IF Largo Sépalo IS BAJO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS MEDIO AND Ancho Pétalo IS ALTO
THEN Clase 3 con Certeza = 0.50

IF Largo Sépalo IS BAJO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND Ancho Pétalo IS BAJO
THEN Clase 2 con Certeza = 0.80

IF Largo Sépalo IS BAJO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND Ancho Pétalo IS MEDIO
THEN Clase 2 con Certeza = 0.90

IF Largo Sépalo IS BAJO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND Ancho Pétalo IS ALTO
THEN Clase 2 con Certeza = 1.00

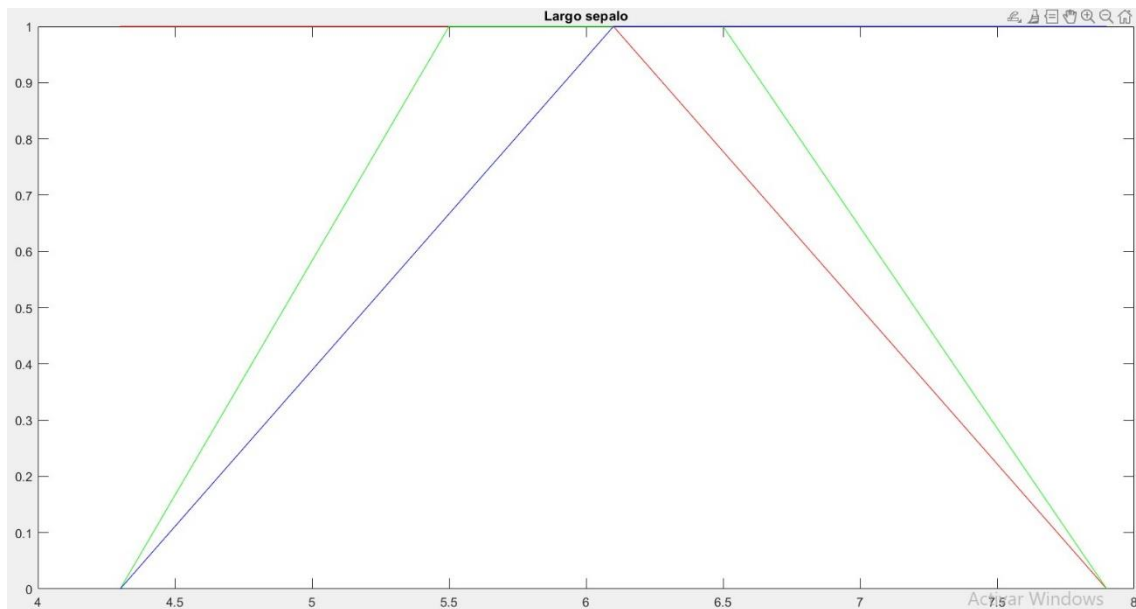
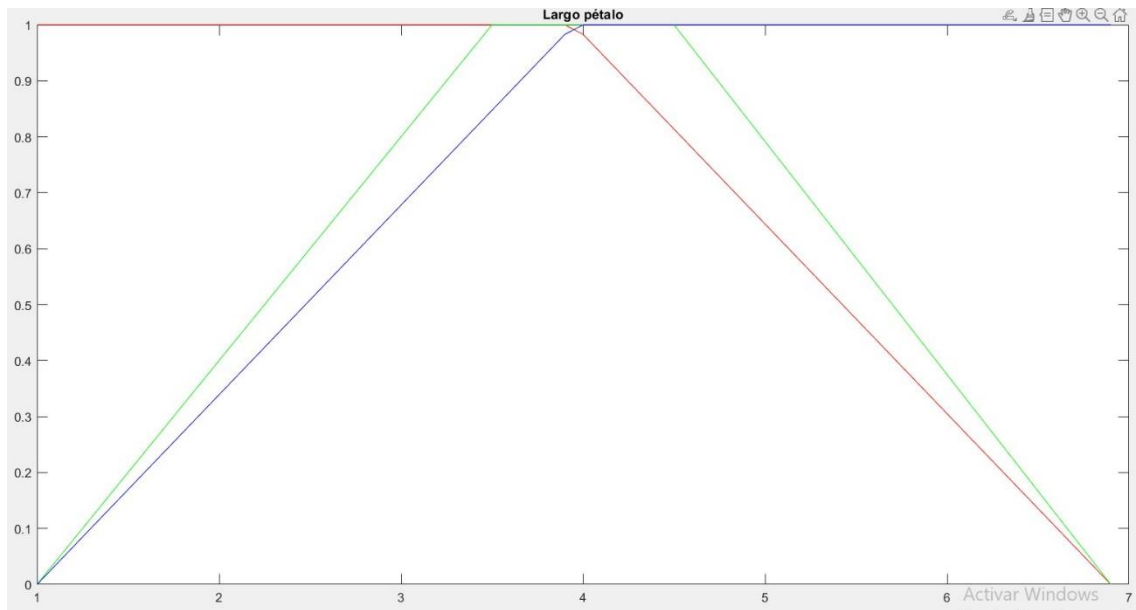
IF Largo Sépalo IS BAJO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS BAJO AND Ancho Pétalo IS BAJO
THEN Clase 1 con Certeza = 1.00

IF Largo Sépalo IS BAJO AND Ancho Sépalo IS ALTO AND Largo Pétalo IS BAJO AND Ancho Pétalo IS BAJO
THEN Clase 1 con Certeza = 0.89

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS MEDIO AND Ancho Pétalo IS BAJO
THEN Clase 2 con Certeza = 0.90

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS MEDIO AND Ancho Pétalo IS MEDIO
THEN Clase 2 con Certeza = 1.00

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND Ancho Pétalo IS MEDIO
THEN Clase 2 con Certeza = 0.80
```



```

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND
Ancho Pétalo IS ALTO
THEN Clase 3 con Certeza = 1.00

IF Largo Sépalo IS MEDIO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS ALTO AND
Ancho Pétalo IS ALTO
THEN Clase 2 con Certeza = 1.00

IF Largo Sépalo IS ALTO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS MEDIO AND
Ancho Pétalo IS MEDIO
THEN Clase 2 con Certeza = 1.00

IF Largo Sépalo IS ALTO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND Ancho
Pétalo IS BAJO
THEN Clase 2 con Certeza = 0.90

IF Largo Sépalo IS ALTO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND Ancho
Pétalo IS MEDIO
THEN Clase 2 con Certeza = 1.00

IF Largo Sépalo IS ALTO AND Ancho Sépalo IS BAJO AND Largo Pétalo IS ALTO AND Ancho
Pétalo IS ALTO
THEN Clase 3 con Certeza = 1.00

IF Largo Sépalo IS ALTO AND Ancho Sépalo IS MEDIO AND Largo Pétalo IS ALTO AND
Ancho Pétalo IS ALTO
THEN Clase 3 con Certeza = 1.00

IF Largo Sépalo IS ALTO AND Ancho Sépalo IS ALTO AND Largo Pétalo IS ALTO AND Ancho
Pétalo IS ALTO
THEN Clase 3 con Certeza = 0.89

El sistema de clasificación nos da una accuracy de 0.33333(33.3333%).
La matriz de confusión es:
    25     0     0
    25     0     0
    25     0     0

>>

```

Como se puede observar, al modificar las funciones de pertenencia a una función trapezoidal, los valores que toman los conjuntos y el dibujo de la gráfica variarán. Además, como los conjuntos son necesarios para el resto de los pasos a realizar, los resultados se verán modificados en cada uno de ellos.

Se destaca que el número de reglas se modifica desde las 14 reglas a las 21, incrementándose en 7. Aumentan las reglas que tienen la clase 2 como consecuencia. Sin embargo, sigue habiendo únicamente 3 reglas para la clase 1.

Por último, la precisión del sistema de clasificación desciende desde un 89.33% hasta un 33.33%. Prácticamente la precisión tras realizar las modificaciones es de un tercio de la anterior sin modificaciones. De cada 100 ejemplos, el resultado del clasificador será erróneo en más de 66.

Se puede observar en la matriz de confusión como todos los ejemplos de la clase 1 se han clasificado correctamente. Sin embargo, las clases 2 y 3 que antes clasificaba por

ejemplo la clase 2 todos los datos correctamente ahora no se clasifican ningún dato bien. Por ello, la precisión decae hasta el 33.33%.

Conclusión final del trabajo:

Como conclusión final del trabajo, el primer clasificador utilizando los datos y funciones que se proporcionan, infiere los datos de una manera muy precisa acertando en la mayoría de los casos. Sin embargo, al realizar las modificaciones, la precisión es pésima debido a que todos los ejemplos de la clase 2 y de la clase 3 se clasifican de forma incorrecta. Para aumentar la precisión sería necesario modificar principalmente las funciones que se utilizan en el grado de compatibilidad y en el grado de asociatividad para que las reglas sean más precisas y así provocar un aumento en los aciertos de las clases 2 y 3.