

Obtención De Reglas de Inferencia a partir de Datos de Entrada y Salida de un Sistema Difuso

Erick Muñiz Torres

Matrícula 769664

Resumen - En el presente artículo se describe el desarrollo y pruebas de un sistema para obtener las reglas de inferencia de un sistema difuso a partir de datos de entrada y salida, el cual, esta basado en las técnicas de agrupamiento. Este sistema se desarrolló utilizando una técnica diferente a la que se reporta en los artículos relacionados al tema, los cuales generalmente utilizan Fuzzy c – means como algoritmo de obtención de reglas de inferencia, en cambio, en este trabajo se utiliza Agrupamiento mediante el vecino más cercano (Nearest Neighborhood Clustering) únicamente. El artículo contiene una introducción a la modelación de sistemas difusos, la justificación para utilizar el algoritmo, se describe y presenta la codificación del algoritmo, documentación de las pruebas que se efectuaron y por último las conclusiones del presente trabajo.

Palabras claves – Modelación de sistemas difusos, Agrupamiento mediante el vecino más cercano, Fuzzy c-means, Reglas de inferencia.

I. INTRODUCCION

La característica central de los sistemas difusos es que están basados en el concepto de particionar la información [Emami, Türksen, Goldenberg, 1998]. La habilidad del sistema para la toma de decisiones depende de la existencia de reglas y de un mecanismo de razonamiento difuso. Este tipo de sistemas puede ser interpretado por

modelos difusos que consisten de reglas del tipo IF – THEN [Emami, Türksen, Goldenberg, 1998]. Sin embargo, el problema de la generación de reglas difusas del tipo IF – THEN es uno de los más importantes dentro del desarrollo de sistemas difusos [Delgado, Gómez-Skarmeta, Martin, 1997] ya que se deben establecer el número de reglas que se desean para cada uno de los sistemas a modelar. Muchas reglas pueden crear un sistema difuso demasiado complejo para el problema y otras veces se puede caer en un sistema con pocas reglas en cual puede ser insuficiente para el sistema que se esta modelando [Wang, 1997].

Se cuenta con diversos métodos para el diseño de un sistema difuso entre los que se pueden mencionar: búsqueda en una tabla de esquemas (del inglés Table look-up scheme), entrenamiento de gradiente y mínimos cuadrados recursivos. Sin embargo, ninguno de éstos propone un método sistemático para determinar el número de reglas en el sistema difuso [Wang, 1997].

Además de los métodos descritos anteriormente existe el agrupamiento mediante el vecino más cercano, en el cual, el número de reglas se consideran como un parámetro de diseño y se determina basándose en los pares de entrada y salida de datos [Wang, 1997].

La idea básica de este método es agrupar los pares de entrada y salida dentro de grupos o clusters y crear una regla por cada uno de los grupos que se formaron [Wang, 1997].

Los grupos en este algoritmo se crean a partir de la distancia que existe entre un dato y el centro de los grupos existentes, si esta dentro de la distancia permitida el dato forma parte del grupo, sino se crea un nuevo grupo a partir de ese dato.

Otra variación de esta técnica es el algoritmo Fuzzy c-means, el cual, puede ser visto como un clasificador de información que puede utilizarse en problemas de reconocimiento de patrones. Este algoritmo es considerado como el más famoso en lo que corresponde a agrupamiento porque ha sido ampliamente utilizado [Wang, 1997].

II. JUSTIFICACIÓN

Se propone utilizar el algoritmo de agrupamiento mediante el vecino más cercano aún y cuando en la revisión de trabajos relacionados se observó exclusivamente el uso del algoritmo Fuzzy c-means.

Sin embargo, para dicho algoritmo se requiere especificar el número de grupos que se desean formar, y un post procesamiento para la obtención de las reglas del sistema, lo que se considera una desventaja con respecto al algoritmo de agrupamiento mediante el vecino más cercano, que es capaz de crear las reglas que él considere necesarias a partir del radio de los grupos.

Con el algoritmo de agrupamiento mediante el vecino más cercano si se requiere un sistema robusto y complejo se disminuye el radio de los grupos y el sistema determinará el número de reglas, así mismo, si se desea un sistema con menor complejidad únicamente se requiere

aumentar el radio de los grupos y el sistema ajustará el número de reglas.

III. DESARROLLO

Para el desarrollo de este sistema se utiliza el método de agrupamiento mediante el vecino más cercano, el cual es uno de los métodos de agrupamiento más sencillo.

El método que se implementó consiste en obtener los grupos a partir de los pares de datos de entradas y salidas, una vez que se crean los grupos, cada uno de ellos se toma como una regla. El centro de cada uno de ellas, es decir, donde su función de membresía toma el valor de uno es el centro de cada uno de los grupos que fueron creados.

El algoritmo funciona de la siguiente forma: se toma el primer par de datos como el centro del primer grupo. Si la distancia entre los siguientes datos es menor al radio establecido al inicio del algoritmo, el dato pertenece a dicho grupo. Si la distancia no es menor al radio se designa ese valor como el centro de un nuevo grupo.

El sistema para obtener las reglas de inferencia a partir de pares de datos de entrada y salida de un sistema difuso fue desarrollado en el lenguaje Matlab.

A continuación se presenta la codificación del algoritmo utilizado.

```
%crea el primer cluster
Y(1) = X(entradas+1);
B(1) = 1;
Xc(reglas,:) = X(1:entradas);
Sigma(par,1) = Sigmaini;
Sigma(par,2) = Sigmaini;
```

```

for par = 2:datos,
    X = parnum(par);
    %calcula la distancia
    [dist regla] = calcdist(X);
    % si la distancia es menor que el
    radio pertenece a dicho cluster
    if dist<r,
        B(regla) = B(regla)+1;
        Y(regla) = Y(regla)+
X(entradas+1);
    end
    %si la distancia es mayor que el
    radio se crea un nuevo cluster
    if dist>=r,
        reglas = reglas +1;
        Xc(reglas,:) = X(1:entradas);
        Sigma(reglas,:) =
Sigma(1:entradas);
        B(reglas)=1;
        Y(reglas)=X(entradas+1);
    end
end
end

```

Una vez que se generan los grupos, se obtiene el centro de cada uno de ellos como el centro de las reglas, es decir, el valor de cada uno de los centros de los grupos pasaran a ser el valor para el cual la regla tiene una función de membresía igual a uno. Este procedimiento se realiza tanto para las entradas como para la salida.

Las reglas se crean a partir del siguiente fragmento de código:

```

for numero=1:reglas,
    for col=1:entradas,
        sistema(numero,col) =
Xc(numero,col);
    end
    sistema(numero,entradas+1)
=Y(numero);
end

```

Las reglas son presentadas en forma indexada, esto es, se presentan tres columnas con los datos para los cuales cada uno de los parámetros de la

regla tiene una función de membresía igual a uno.

IV. PRUEBAS

El experimento para observar el desempeño de este método fue diseñado de la siguiente forma:

Los datos de entrada se muestran en las figuras 1 y 2. Los datos de salida se muestran en la fig. 3.

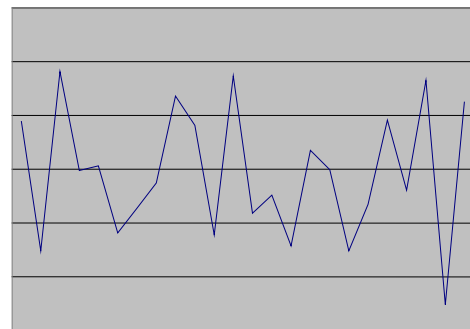


Figura 1. Datos de entrada 1

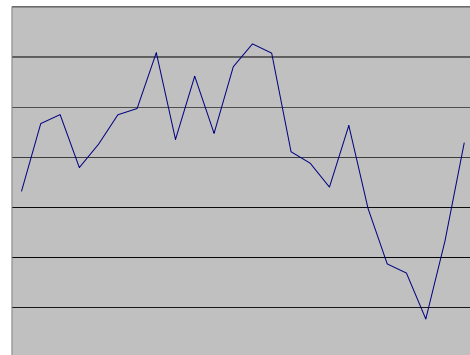


Figura 2. Datos de entrada 2

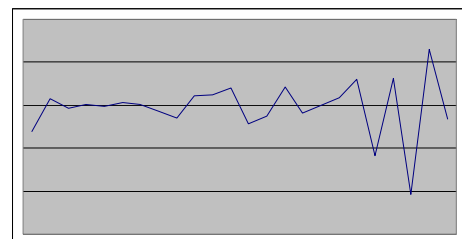


Figura 3. Datos de salida

La figura 4 presenta el sistema difuso que se desea modelar.

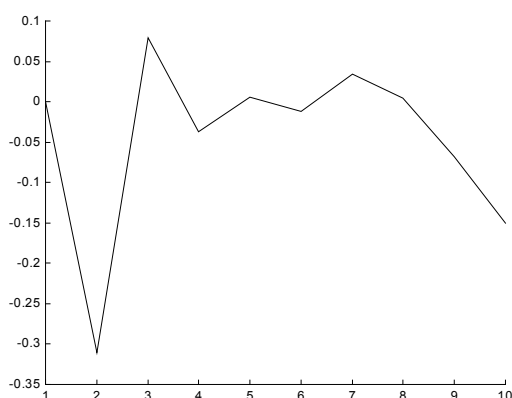


Figura 4. Sistema difuso a modelar

Se hicieron un total de 25 simulaciones del sistema para observar el comportamiento del algoritmo propuesto, los resultados de estos experimentos se condensan en la tabla 1.

Tabla 1. Resultados de los experimentos

Experimento	DATOS			
	Radio	Sigma	Reglas	Error
1	.1	.2	23	.0439
2	.1	.1	23	.0098
3	.1	.3	23	.0823
4	.1	.4	23	.1312
5	.3	.2	15	.0922
6	.3	.3	15	.1004
7	.3	.4	15	.1450
8	.3	.1	15	.0945
9	.5	.1	12	.1312
10	.5	.2	12	.1457
11	.5	.3	12	.1498
12	.5	.4	12	.1766
13	.7	.1	8	.3972
14	.7	.2	8	.3876
15	.7	.3	8	.3809
16	.7	.4	8	.3700
17	.9	.1	6	.3054
18	.9	.2	6	.3049
19	.9	.3	6	.3372
20	.9	.4	6	.3484
21	1.1	.1	3	.6611
22	1.1	.2	3	.6298
23	1.1	.3	3	.5932
24	1.1	.4	3	.5783
25	1.63	.3	2	.07690

Para presentar los resultados se toma como ejemplo el experimento con un radio igual a 0.1 y Sigma igual a 0.1 con el fin de observar en la figura 5 las reglas que fueron obtenidas y en la figura 6 la unión del sistema difuso y el sistema obtenido por este método.

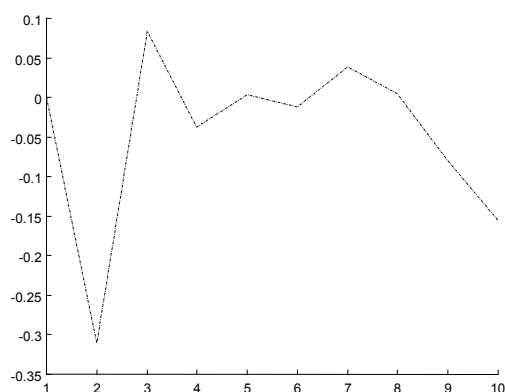


Figura 5. Reglas obtenidas para $r=0.1$ y $S=0.1$

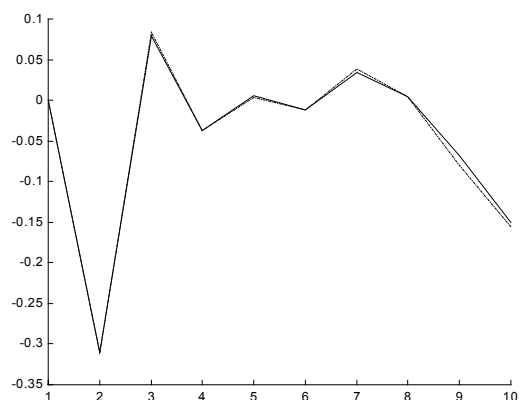


Figura 6. Sistema difuso y reglas obtenidas para $r=0.1$ y $S=0.1$

Las reglas que fueron generadas por el sistema a partir del experimento en el cual se tomó un radio igual a 0.9 y Sima igual a 0.2 son las siguientes:

0	0	0.2184
0.4435	-0.8382	-1.0424
0.8675	0.4023	0.1991
-0.1932	-1.6545	0.3082
0.8283	-2.1126	-1.0379
-1.2556	-1.3346	0.6504

La interpretación de este resultado es la siguiente:

IF x1 is 0 AND x2 is 0
THEN B is 0.2184

IF x1 is 0.4435 AND -0.8382
THEN B is -1.0424

IF x1 is 0.8675 AND x2 is 0.4023
THEN B is 0.1991

IF x1 is -0.1932 AND x2 is -1.6545
THEN B is 0.3082

IF x1 is 0.8283 AND x2 is -2.1126
THEN B is -1.0379

IF x1 is -1.2556 AND x2 is -1.3346
THEN B is 0.6504

La gráfica del sistema difuso y las reglas obtenidas con los parámetros de radio igual a 0.9 y Sigma igual a 0.2 se muestran en la figura 7.

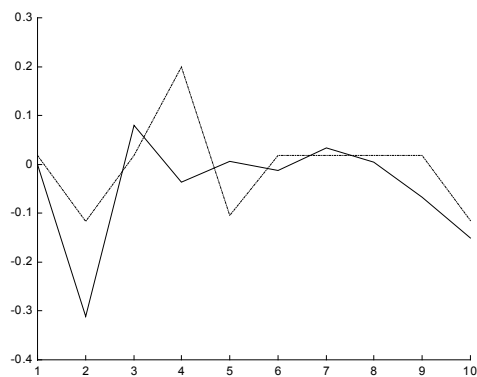


Figura 7. Sistema difuso y reglas obtenidas para $r=0.9$ y $S=0.2$

V. CONCLUSIONES

En lo correspondiente al funcionamiento del algoritmo de agrupamiento mediante el vecino más cercano se pudo observar que el parámetro del radio dicta el número de reglas o complejidad del sistema a

modelar, es decir, a menor radio mayor complejidad.

Mientras que el parámetro Sigma se encuentra relacionado con la suavidad para realizar los grupos influyendo directamente en el error cuadrático entre el sistema y el modelo.

Esto comprueba que el funcionamiento del algoritmo es correcto debido a que el comportamiento es tal y como se describe en la teoría [Wang, 1997].

Se concluye que es factible utilizar el método de agrupamiento mediante el vecino más cercano como metodología para obtener el modelo de un sistema difuso.

Este método puede obtener el modelo del sistema sin tener que indicar el número de reglas que se desean obtener, tal y como se realiza utilizando Fuzz c-means, con lo que se convierte en otra alternativa viable para la identificación de sistemas difusos a partir de los pares de datos de entrada y salida.

REFERENCIAS

Delgado M. Gómez-Skarmeta F., and Martin F. (1997). *A Fuzzy Clustering-Based Rapid Prototyping for Fuzzy Rule-Based Modeling*. IEEE Transactions on Fuzzy Systems. Vol. 5. No. 2.

Emami M., Türksen B. And Goldenberg A. (1998). *Development of a Systematic Methodology of Fuzzy Logic Modeling*. IEEE Transactions on Fuzzy Systems. Vol 6. No. 3.

Wang L. C. (1997). *A course in Fuzzy Systems and Control*. Prentice Hall.

APENDICE A. CODIFICACIÓN DEL ALGORITMO DE AGRUPAMIENTO MEDIANTE VECINO MÁS CERCANO

```

%programa de implementación de fuzzy clustering
%limpia la memoria
clear all
clear global
clc
%definición e inicialización de variables
global datosent Xc reglas entradas
global Y Sigma B
global a b
load datosent.dat
[datos columnas]=size(datosent);
entradas = columnas-1;
clear columnas;
reglas=1;
par = 1;

%ingreso de datos
r=input ('Radio r = ');
Sigmaini= input ('sigma = ');
X = parnum(par);

%Crea el primer cluster con los primeros datos de entrada y salida
Y(1) = X(entradas+1);
B(1) = 1;
Xc(reglas,:) = X(1:entradas);
Sigma(par,1) = Sigmaini;
Sigma(par,2) = Sigmaini;

%ciclo para verificar si los demás datos pertenecen al cluster creado
%o se requiere crear un nuevo cluster
for par = 2:datos,
    X = parnum(par);
    %calcula la distancia
    [dist regla] = calcdist(X);
    % si la distancia es menor que el radio pertenece a dicho cluster
    if dist<r,
        B(regla) = B(regla)+1;
        Y(regla) = Y(regla)+ X(entradas+1);
    end
    %si la distancia es mayor que el radio se crea un nuevo cluster
    if dist>=r,
        reglas = reglas +1;
        Xc(reglas,:) = X(1:entradas);
        Sigma(reglas,:) = Sigma(1:entradas);
        B(reglas)=1;
        Y(reglas)=X(entradas+1);
    end
end

```

```

end

%para comparar el sistema con la identificación realizada por el método
SalidaMetodo = saldered(datos);
SalidaSistema= datosent(:,entradas + 1);
error =0.5 * norm((SalidaMetodo - SalidaSistema'));

% una vez que se tienen las reglas se acomodan para presentar al
%usuario el conjunto de reglas
for numero=1:reglas,
    for col=1:entradas,
        sistema(numero,col) = Xc(numero,col);
    end
    sistema(numero,entradas+1) =Y(numero);
end

%Crea la figura con las gráficas de salida
figure
hold on
Sr=plot(SalidaMetodo(1:10),'-','color','k');
Sp=plot(SalidaSistema(1:10),'k');

%calcula la distancia entre X y Xc
function [dist, regla]= calcdist(X)
global Xc reglas entradas
dist=0;
for numregla=1:reglas,
    dista(numregla)=norm(X(1:entradas)-Xc(numregla,:));
end
[dist regla] = min(dista);

%funcion que obtiene fk(X)
function [SalidaRed]= saldered(datos)
for par=1:datos,
    k=parnum(par);
    SalidaRed(par) = salred(k);
end

%Salida del sistema difuso Fk(X)
function [SalRed]= salred(X)
global reglas entradas Xc Y B Sigma
global a b
a=0;b=0;
for m=1:reglas,

```

```
Suma = 0;
  for n=1:entradas,
    Suma = Suma + ( ( X(n) - Xc(m,n) ) / Sigma(m,n) )^2;
  end
z(m)=exp(-Suma);
a = a + Y(m)*z(m);
b = b + B(m)*z(m);
end
SalRed=a/b;
```