FUZZY C MEANS CLUSTERING

Nicolas Guarin Gaviria, Milton Eduardo Rios Salazar

Instituto Tecnológico Metropolitano

INTRODUCCIÓN.

Este documento implementa un condigo en Matlab del programa de agrupación difusa c-means (FCM). El programa FCM es aplicable a una amplia variedad de problemas de análisis de datos; en este caso tenemos 2 ejemplos una BD de un Jugador del cual se toman datos como Rapidez y Resistencia y otra sobre COVID Colombia, los datos están vinculados a grupos que representan el comportamiento difuso de este algoritmo. Para hacer eso, simplemente tenemos que construir una matriz apropiada llamada U cuyos factores son números entre 0 y 1, y representar el grado de pertenencia entre los datos y los centros de los grupos.

El algoritmo Fuzzy c-means (FCM) es un método de agrupación que permite que una pieza de datos pertenezca a dos o más agrupaciones. Este método (desarrollado por Dunn en 1973 y mejorado por Bezdek en 1981) se usa con frecuencia en el reconocimiento de patrones; es una técnica de minería de datos que permite encontrar grupos en un conjunto de datos y es aplicado en diversos campos como clasificación de datos, reconocimiento de patrones, estudio del clima y diagnóstico de enfermedades entre otras.

1. PROCEDIMIENTO

Para la creación del FCM que se expone en este documento se aplica a una base de datos previamente cargada y leída: determinación de los grupos K, Inicio de centroides, Parámetro m (fusividad), número de Iteraciones y Epsilon $\boldsymbol{\varepsilon}$ (desempeño), este grupo es propio de un modelo Kmeans, luego se aplican los siguientes cálculos:

Cálculo de grados de pertenencia

$$\mu_{C_i}(x) = \frac{1}{\sum_{j=1}^{k}\left(\frac{|x-v_i|^2}{|x-v_j|^2}\right)^{\frac{1}{m-1}}} \quad 1 \le i \le k, x \in X$$

Cálculo de nuevos centros

$$v_i = \frac{\sum_{x \in X}(\mu_{C_i}(x))^m \, x}{\sum_{x \in X}(\mu_{C_i}(x))^m} \quad 1 \le i \le k$$

Variación de centros

$$V_{i(t)} = \| V_{i(t)} - V_{i(t-1)} \|$$

2. OBJETIVO GENERAL

Implementar y dasarollar un clasificador FCM para 2 bases de datos una integrada en el contexto de familiarización y otra en el ejercicio de desarrollo de procesos propios.

Objetivos Específicos

- Identificar los conceptos que conforman FCM
- Identificar la estructura y funciones númericas que lo conforman y su grupo de procesos
- Validar y analizar los procesos gráficos otenidos.

3. DESCRIPCIÓN ESQUEMA PROPUESTO EJEMPLO DE CLASE

Se tiene una tabla (Tabla 1) en la cual se evalúa la rapidez y la resistencia de 11 futbolistas. Un valor cercano a 1 indica que el jugador es bastante rápido o resistente según el caso, un valor cercano a 0 muestra que el jugador es lento o poco resistente. Se desea separar el conjunto de datos en 2 grupos (cluster) para ver si se encuentran jugadores con características especiales, para esto se hará uso del algoritmo FCM.
[notas de clase]

JUGADOR	RAPIDEZ	RESISTENCIA
1	0.58	0.33
2	0.90	0.11
3	0.68	0.17
4	0.11	0.44
5	0.47	0.81
6	0.24	0.83
7	0.09	0.18
8	0.82	0.11
9	0.65	0.50
10	0.09	0.63
11	0.98	0.24

Tabla 1 Evaluación de la rapidez y resistencia de 11 futbolistas

Código Principal

```
clc
close all
clear all

rapidez=[0.58 0.90 0.68 0.11 0.47 0.24 0.09 0.82
0.65 0.09 0.98];
resistencia=[0.33 0.11 0.17 0.44 0.81 0.83 0.18
0.11 0.5 0.63 0.24];
%defensa=[0.57 0.8 0.37 0.63 0.1 0.98 0.67 0.17
0.21 0.78 0.5];
ciclo=0;
grupo=[rapidez',resistencia'];
k= N; %reemplazar la N por 2, 3, 4 y 5
centro=rand(k,2);
valores2=centro;
e=0.000001

while e && ciclo <= 100
    v=[];
    vc1=[];
    for t=1:length(grupo)
        for j=1:length(centro)
            c=sum((grupo(t,:)-centro(j,:)).^2);
            v(t,j)=c;
        end
    end

    for t=1:length(v)
        for j=1:length(centro)
            cx=1/sum(((v(t,:)))/v(t,j));
            vc1(t,j)=cx;
            vcx=fliplr(vc1);
        end
    end
    vt=[];
    for f=1:length(v)
        for g=1:length(centro)

v1x=sum(vcx(:,g).^2.*grupo(:,:,1))/sum(vcx(:,g).^2
);
            vt(g,:)=v1x;
        end
    end

    r1=sum(vt,2);
    rr1=sum(r1);
    r2=sum(centro,2);
    rr2=sum(r2);
    e=abs(rr2-rr1)
    centro=vt;
    ciclo=ciclo+1;

    scatter(rapidez,resistencia,30,'*')
    hold on
    scatter(centro(:,1),centro(:,2),'d')
    scatter(valores2(:,1),valores2(:,2),'filled')

end


 %coeficiente de particion... hacer una tabla con
los valores de K...
    %este valor debe de dar cercano a 1
    valor1=[];
    for h=1:length(vcx)
        for u=1:length(centro)
           pc=sum(vcx(:,u).^2);
           valor1(:,u)=pc;
           resultado1=sum(valor1)/length(vcx)
        end
    end
    %coeficiente de entropia se debe de hacer lo
mismo pero su valor debe
    %de estar cerca de 0
    valor2=[];
    valor3=[];
    for h=1:length(vcx)
        for u=1:length(centro)
            ce=sum(vcx(h,u));
            valor2(:,u)=ce;
        end
    end
    for f=1:length(vcx)
        for z=1:length(centro)
             ce1=sum(log10(vcx(f,z)));
             valor3(:,z)=ce1;
        end
    end
resultado2=-sum(valor2.*valor3)/length(vcx)
```

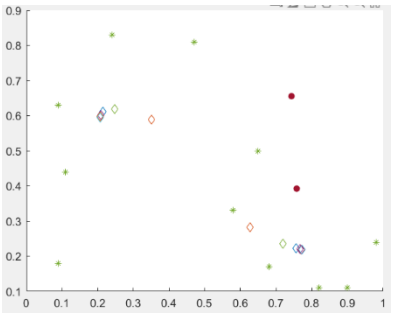
La validez de un algoritmo de agrupamiento (clustering) se estima mediante un criterio objetivo para determinar que tan buena es la partición generada por el algoritmo.

Estos criterios son importantes porque permiten compararlos resultados de diversos algoritmos y permiten determinar el mejor número de clusters

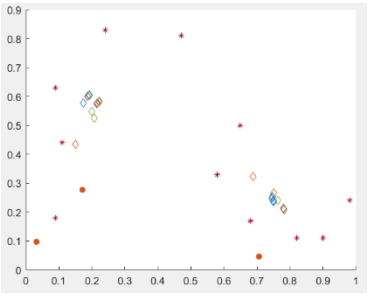
[La validez del grupo se refiere al problema de determinar si la partición difusa obtenida se ajusta adecuadamente a todos los datos.]

tabla de análisis		
K=2	0,8127	0,0086
K=3	0,547	0,0317
K=4	0,4036	0,0359
K=5	0,3229	0,049
	PC	PE

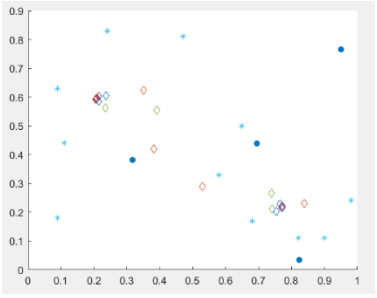
Validación de la clasificación para K = 2



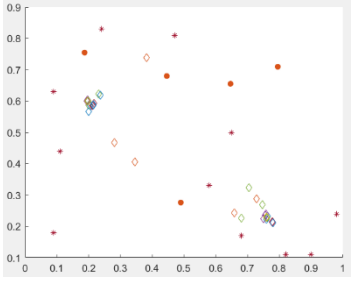
Validación de la clasificación para K = 3



Validación de la clasificación para K = 4



Validación de la clasificación para K = 5



4. IDENTIFICACIÓN DE UNA PLANTA DEESCRIPCIÓN ESQUEMA PROPUESTO EJEMPLO DE CLASE

Se implementa una Base de Datos tomáda de https://github.com/datasets/covid-19, la cual toma los datos a nivel mundial sobre la actual pandemia COVID19-20, por prácticidad aislamos solo los datos de Colombia, en la cual sse comparan lo datos de casos confimados Vs casos recuperados. Fecha de inicio BD 6/03/2020 – Fecha de final de Evaluación BD 1/06/2020

tabla de análisis		
K=2	0,9014	0,0018
K=3	0,5455	0,0035
K=4	0,4507	0,0052
K=5	0,41	0,0067
K=6	0,3151	0,0072
K=7	0,2427	0,0078
K=8	0,2254	0,0087
K=9	0,1818	0,0089
K=10	0,1671	0,0095
	PE	PC

Covid Colombia	
Confirmado	Recuperado
1	0
1	0
1	0
1	0
3	0
3	0
3	0
13	0
22	0
34	0
54	0
65	1
93	1
102	1
128	1
196	1
231	3
277	3
378	6
470	8
491	8
539	10
608	10
702	10
798	15
906	31
1065	39
1161	55
1267	55
1406	85
1485	88
1579	88
1780	100
2054	123
2223	174
2473	197
2709	214
2776	270
2852	319
2979	354
3105	452
3233	550
3439	634
3439	634

Covid Colombia	
Confirmado	Recuperado
3792	711
3977	804
4149	804
4356	870
4561	927
4881	1003
5142	1067
5379	1133
5597	1210
5949	1268
6207	1411
6507	1439
7006	1551
7285	1666
7668	1722
7973	1807
8613	2013
8959	2148
9456	2300
10051	2424
10495	2569
11063	2705
11613	2825
12272	2971
12930	3133
13610	3358
14216	3460
14939	3587
15574	3751
16295	3903
16935	4050
17687	4256
18330	4431
19131	4575
20177	4718
21175	5016
21981	5265
23003	5511
24104	6111
24141	6132
25406	6687
26734	6935
27219	7032
29384	8384

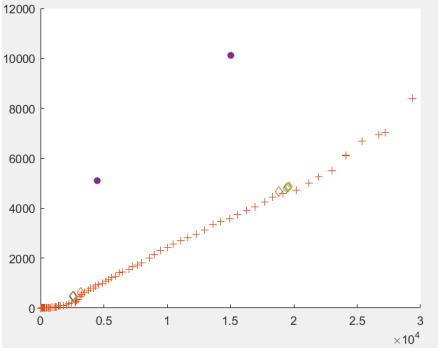
Tabla2 – evaluación de datos casos confimados a casos recuperados

Código Principal

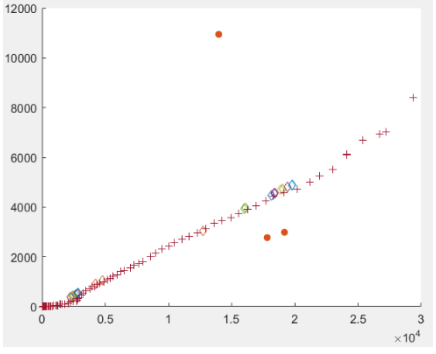
```
clc
close all
clear all

ciclo=0;
grupo=[database(1),database(2)];% valores de la
base de datos
k=2;% diferentes centroides... Hacerlo para 10
centroides maximo...
centro=randi([1,20000],k,2); %como son dos grupos
dejo el 2 indicado
valores2=centro; %gardo los valores en otra varia-
ble para graficarlos
e=0.000001 % error

while e && ciclo <= 100
    v=[];
    vc1=[];
    for t=1:length(grupo)
        for j=1:length(centro)
            c=sum((grupo(t,:)-centro(j,:)).^2);
            v(t,j)=c;
        end
    end

    %ciclo en que la suma de su fila da 1
    for t=1:length(v)
        for j=1:length(centro)
            cx=1/sum(((v(t,:)))/v(t,j));
            vc1(t,j)=cx;
            vcx=fliplr(vc1);
        end
    end
    %procedimineto de suma de nuevos pesos
    vt=[];
    for f=1:length(v)
        for g=1:length(centro)

v1x=sum(vcx(:,g).^2.*grupo(:,:,1))/sum(vcx(:,g).^2
);
            vt(g,:)=v1x;
        end
    end


    %operaciones para el error
    r1=sum(vt,2);
    rr1=sum(r1);
    r2=sum(centro,2);
    rr2=sum(r2);
    e=abs(rr2-rr1);
    centro=vt;
    ciclo=ciclo+1

    %graficas
    scatter(grupo(:,1),grupo(:,2),'+')
    hold on
    scatter(centro(:,1),centro(:,2),'d')
    scatter(valores2(:,1),valores2(:,2),'filled')

end

 %coeficiente de particion... hacer una tabla con
los valores de K...
    %este valor debe de dar cercano a 1
    valor1=[];
    for h=1:length(vcx)
        for u=1:length(centro)
           pc=sum(vcx(:,u).^2);
           valor1(:,u)=pc;
           resultado1=sum(valor1)/length(vcx)
        end
    end
    %coeficiente de entropia se debe de hacer lo
mismo pero su valor debe
    %de estar cerca de 0
    valor2=[];
    valor3=[];
    for h=1:length(vcx)
        for u=1:length(centro)
            ce=sum(vcx(h,u));
            valor2(:,u)=ce;
        end
    end
    for f=1:length(vcx)
        for z=1:length(centro)
             ce1=sum(log10(vcx(f,z)));
             valor3(:,z)=ce1;
        end
    end
  resultado2=-sum(valor2.*valor3)/length(vcx)
```

Validación de la clasificación.
válida para máximo 10 centroides

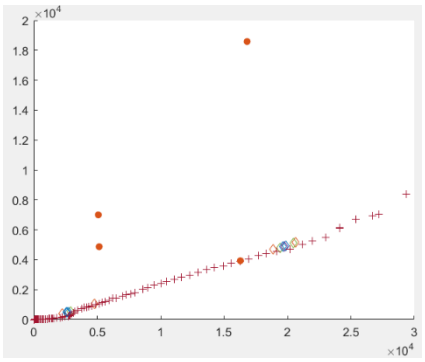
Validación de la clasificación para K = 2



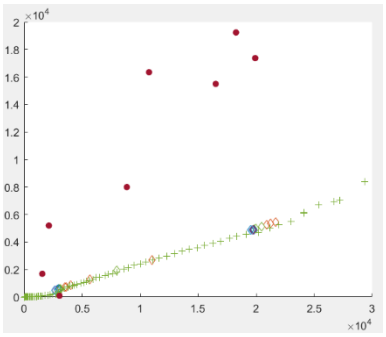
Validación de la clasificación para K = 3



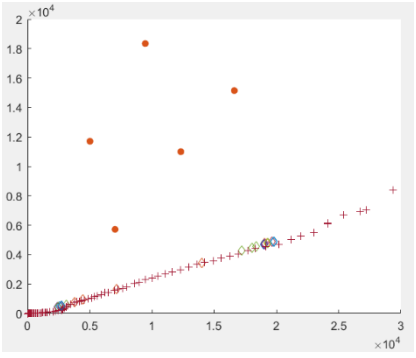
Validación de la clasificación para K = 4



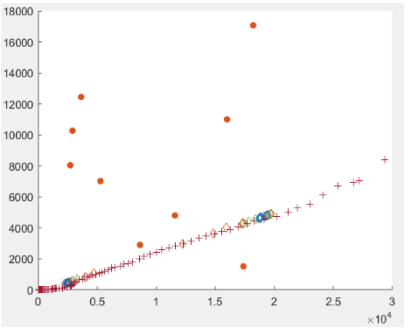
Validación de la clasificación para K = 5



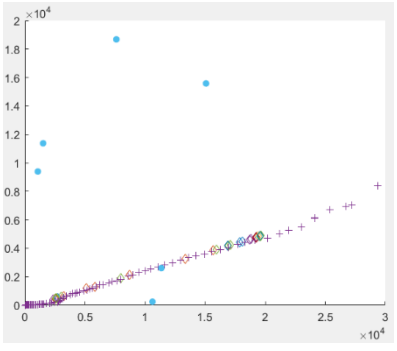
Validación de la clasificación para K = 9



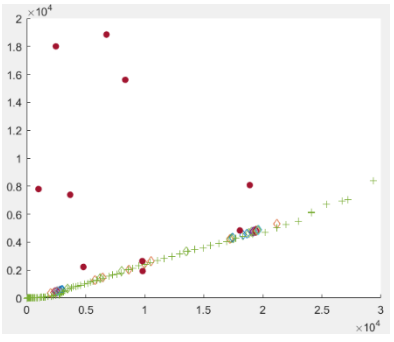
Validación de la clasificación para K = 6

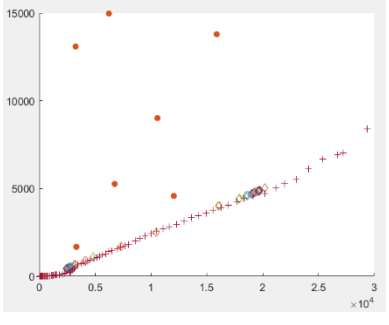


Validación de la clasificación para K = 10



Validación de la clasificación para K = 7





Validación de la clasificación para K = 8

5. ANALISIS Y CONCLUSIONES

El caso más favorable para para el ejemplo de clase es K=2

El caso más favorable para para el ejemplo pro-propuesto es K=2

El uso de FCM en la minería de datos permite analizar y encontrar patrones los cuales podemos asociar y clasificar en grupos grandes y está minería se basa en técnicas estadísticas o algoritmos de la

inteligencia artificial.

El clustering más usado para realizar clustering es CMeans pero este crea una partición dura de los datos incluso si un dato tiene características de grupos diferentes.

Fuzzy C-Means (FCM) es una extensión de CMeans en la cuál el dato puede pertenecer a más de un clustering, además, de calcular los prototipos del cluster, también calcula las funciones de pertenencia de los datos dentro de cada cluster.

FCM produce una partición suave por esto es útil en situaciones en las que los datos poseen características de distintos grupos.

Existen reglas de la valide y estas se basan en características de la función de pertenencia o en el desempeño de la partición.

References

[1] Frawley William J., Piatetsky-Shapiro Gregory, Matheus Cristopher J, "Knowledge Discovery in Databases: An Overview" pages 1--27. AAAI/MIT Press, 1991.

[2] U.K. Chakraborty, D.G. Dastidar (1993). Using reliability analysis to estimate the number of generations to convergence in genetic algorithms. Information Processing Letters, 46, 199-209.

[3] Klir George J, Yuan Bo Fuzzy Sets and Fuzzy Logic theory and applications, New Jersey: Prentice Hall, 1995, p. 357-362.

[4] T.E. Davis, J.C. Principe (1993). A Markov chain framework for the simple genetic algorithm. Evolutionary Computation, 1(3), 269-288.

[5] Díaz Diez Bárbara, Morillas Raya Antonio, "Minería de Datos y Lógica Difusa. Una aplicación al estudio de la rentabilidad económica de las empresas agroalimentarias en Andalucía" Estadística Española. Vol. 46, Núm. 157, 2004 p. 409-430.

[6] Fundamentación de la materia sistemas inteligentes, ITM, Orlando Zpata Cortez

[7] http://revistas.utp.edu.co/index.php/revistaciencia/article/viewFile/3095/1695