



Universidad Veracruzana

Maestría en Inteligencia Artificial

Lógica difusa

**Tarea 11. Comparativa entre K-medias y Fuzzy
C-means aplicado sobre iris en MATLAB.**

Ángel García Báez

Dr. Sergio Hernández Méndez

15 de mayo de 2025

Índice

1. Introducción	2
2. Definiciones	3
2.1. Distancia	3
2.2. Distancia Euclidiana	3
2.3. Distancia Mahalanobis	3
2.4. Distancia Manhattan	4
2.5. Algoritmo para fuzzy c-means	5
2.5.1. Paso 1	5
2.5.2. Paso 2	5
2.5.3. Paso 3	6
2.5.4. Paso 4	6
3. Algoritmo de K-medias	7
4. Algoritmo de Componentes Principales PCA.	9
5. Metodología	12
6. Resultados	13
6.1. Resultados para la distancia Euclidiana	13
6.2. Resultados para la distancia de Mahattan	15
7. Conclusiones	17
8. Referencias	18
9. Anexos	19

1. Introducción

En el presente reporte se plantea la idea de aplicar el algoritmo de agrupación con lógica difusa "Fuzzy C-means.^a la base de datos iris para contrastar sus resultados respecto al algoritmo de "K-medias".

2. Definiciones

2.1. Distancia

Acorde con Wang and Sun (2015), para un conjunto de datos tal que si $x, y, z \in M$ son vectores de datos de la misma dimensionalidad, se puede definir $D : M \times M \rightarrow R$ como una distancia métrica si se satisfacen las siguientes propiedades:

- No negatividad: $D(x, y)$.
- Coincidencia: $D(x, y) = 0$ si y solo si $x = y$.
- Simetría: $D(x, y) = D(y, x)$
- Subaditividad: $D(x, y) + D(y, z) \geq D(x, z)$

Las siguientes definiciones de distancias se obtuvieron del artículo de revisión de Wang and Sun (2015):

2.2. Distancia Euclidiana

$$d(x_1, x_2) = \sqrt{\sum (x_1 - x_2)^2}$$

Donde:

- x_1 es el vector fila de tamaño $1 \times P$.
- x_2 es el vector fila de tamaño $1 \times P$.
- $\sum (x_1 - x_2)^2$ Es la suma de las diferencias al cuadrado de cada componente de los vectores, el resultado es un escalar.

2.3. Distancia Mahalanobis

$$d(x_1, x_2) = \sqrt{(x_1 - x_2)\Sigma^{-1}(x_1 - x_2)^T}$$

Donde:

- x_1 es el vector fila de tamaño $1 \times P$.
- x_2 es el vector fila de tamaño $1 \times P$.
- Σ^{-1} es la inversa de la matriz de varianzas y covarianzas de todo el conjunto de datos al que pertenecen los vectores.

2.4. Distancia Manhattan

$$d(x_1, x_2) = \sum |(x_1 - x_2)|$$

- x_1 es el vector fila de tamaño $1 \times P$.
- x_2 es el vector fila de tamaño $1 \times P$.
- $\sum (x_1 - x_2)^2$ Es la suma de las diferencias en valor absoluto de las componentes de los vectores, el resultado es un escalar.

2.5. Algoritmo para fuzzy c-means

El algoritmo fuzzy c-means fue propuesto en un inicio por Nascimento et al. (2000) como una alternativa para el agrupamiento de datos en el contexto del aprendizaje no supervisado de tal forma que se le pudiera agregar esta capa de computo suave al proceso con el fin de llegar a mejores resultados.

A continuación se muestra una versión del algoritmo que es explicada de forma más clara y concisa que fue obtenida de Edla et al. (2020):

2.5.1. Paso 1

Inicializa la matriz $U(t)$ (matriz de pertenencias) de tamaño $n \times k$ con valores aleatorios entre 0 y 1, de tal forma que cada fila sume 1.

2.5.2. Paso 2

Calcula el valor de los k centroides haciendo uso de la matriz de pertenencias $U(t)$ acorde con la siguiente formula:

$$vk = \frac{\sum_{i=1}^n u_{ki}^m x_i}{\sum_{i=1}^n u_{ki}^m}$$

- v_k : Centroide del cluster k (vector de características promedio ponderado).
- n : Número total de datos o muestras.
- u_{ki} : Grado de pertenencia del dato i al cluster k .
- m : Parámetro de difusidad o fuzzificación ($m > 1$).
- x_i : Vector de características del dato i .
- $\sum_{i=1}^n u_{ki}^m x_i$: Suma ponderada de los datos x_i por el grado de pertenencia elevado a m .
- $\sum_{i=1}^n u_{ki}^m$: Suma de los grados de pertenencia elevados a m (factor de normalización).

2.5.3. Paso 3

Actualizar la matriz $U(t)$, con la matriz $U(t+1)$ reemplazando los elementos de la matriz con los siguientes:

$$u_{ki} = \frac{1}{\sum_{j=1}^c \frac{\|x_i - v_k\|^{2/(m-1)}}{\|x_i - v_j\|}}$$

- u_{ki} : Grado de pertenencia del dato i al cluster k .
- c : Número total de clusters.
- j : Índice que recorre todos los clusters de 1 a c .
- x_i : Vector de características del dato i .
- v_k : Centroide del cluster k .
- v_j : Centroide del cluster j .
- $\|x_i - v_k\|$: Distancia entre el dato i y el centroide k .
- $\|x_i - v_j\|$: Distancia entre el dato i y el centroide j .
- m : Parámetro de difusidad o fuzzificación ($m > 1$).
- $\sum_{j=1}^c \left(\frac{\|x_i - v_k\|}{\|x_i - v_j\|} \right)^{2/(m-1)}$: Suma que normaliza las pertenencias relativas del punto x_i respecto a todos los centroides.

2.5.4. Paso 4

Verificar que $\|U(t) - U(t+1)\| < \epsilon$, si la diferencia de la anterior con la nueva matriz de pertenencias es menor al límite de tolerancia permitido, detiene el proceso, si no, vuelve al paso 2 y sigue iterando hasta alcanzar un máximo de iteraciones (fijado a 100 para efectos prácticos).

3. Algoritmo de K-medias

De acuerdo con lo que explica Bishop (2006), el K-medias puede entenderse como un algoritmo de aprendizaje no supervisado el cual tiene como propósito generar variables latentes categóricas o etiquetar al conjunto de datos dada su similitud en K posibles grupos formados alrededor de K posibles centroides.

Para poder inicializar el algoritmo, es necesario proporcionarle justamente un valor de K para formar esa posible cantidad de grupos con los datos de los que se dispone pero esto representa un problema en si mismo dado que no hay una manera de saber cuantos grupos están presentes en los datos. Para poder abordar esto existen varios enfoques, como por ejemplo, probar y ver con cuantos K se logra minimizar la función de costo o la distancia entre grupos, probar a ver cual da un mayor valor de AIC, BIC o la mayor verosimilitud posible, entre otros más criterios que se han propuesto a lo largo del tiempo. Para el enfoque de este trabajo se decidió quedarse con con el criterio de la distancia entre grupos para determinar el numero de K óptimo de grupos.

Ahora sí, partiendo de un conjunto de datos con N filas y P variables continuas se desea etiquetar a cada fila dado un valor K propuesto, para ello se siguen los siguientes pasos del algoritmo:

- **Paso 1:** Se toman K sujetos aleatorios del conjunto de datos que van a ser los centroides μ_k iniciales.
- **Paso 2:** Se define una medida de distancia que es la norma al cuadrado de la diferencia entre el punto **n-ésimo** y el centroide **k-ésimo** como sigue:

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} ||x_n - \mu_k||^2$$

donde r_{nk} es una variable para identificar la asignación del punto **n-ésimo** al grupo **k-ésimo** si la distancia a dicho grupo es la mínima con respecto a todos los demás posibles grupos.

- **Paso 3:** Con los datos etiquetados en el paso 2 se toman para actualizar el valor de los centroides del siguiente modo:

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum r_{nk}}$$

- **Paso 4:** Se verifica la convergencia con la medida J , si $J^{(i)} = J^{(i-1)}$ entonces se detiene el proceso, si no es el caso, se repiten los pasos 2 y 3 hasta lograr la convergencia.

De esta sencilla forma, se puede aplicar el método de las k – *medias* para la segmentación de datos.

4. Algoritmo de Componentes Principales PCA.

Para poder mostrar lo que esta ocurriendo en espacios multivariantes de más de 3 dimensiones, es necesario recurrir a tecnicas de reducción de dimensionalidad, por lo que en este sentido, para poder representar el comportamiento de los grupos se recurre al uso de PCA.

PCA es una técnica multivariada de aprendizaje no supervisado que permite reducir la dimensionalidad de datos cuantitativos continuos al proyectar la información en un nuevo espacio de variables incorrelacionadas sin perder apenas información en la transformación como se explica en el libro de Johnson and Wichern (2007).

Para ejecutar la técnica, se sugiere primero obtener el vector de medias de la matriz y restarlo a la misma para centrar los datos y garantizar que el subespacio vectorial que va a encontrar la técnica contiene al vector nulo $\vec{0}$ como se muestra a continuación:

$$\begin{aligned}X &= \{x_1, x_2, \dots, x_n\} \\ \bar{X} &= \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\} \\ X_c &= X - \bar{X}\end{aligned}$$

Donde:

- X es la matriz de datos de tamaño $N \times P$.
- \bar{X} Es el vector de medias de tamaño $1 \times P$.
- X_c Es la matriz de datos centrados en la media.

Una vez que se tiene la matriz centrada, se procede con el calculo de su matriz de varianzas y covarianzas como sigue:

$$S = \frac{1}{n-1}(X - \bar{X})'(X - \bar{X})$$

Donde:

- X es la matriz de datos de tamaño $N \times P$.
- \bar{X} Es el vector de medias de tamaño $1 \times P$.
- N Es la cantidad de observaciones en la matriz X .

Una vez que se tiene lista la matriz de covarianzas, es necesario descomponerla en sus valores y vectores propios (eigen valores y eigen vectores) para encontrar el nuevo espacio donde proyectar los datos originales. Para esto se plantea la siguiente ecuación característica sobre la matriz de varianzas y covarianzas:

$$Sv = \lambda v$$

Donde:

- S es la matriz de varianzas y covarianzas de los datos de tamaño $P \times P$.
- λ son los valores propios asociados a la matriz.
- v son los vectores propios asociados a la matriz .

Haciendo un pequeño arreglo, se reescribe la ecuación y queda tal que así:

$$\begin{aligned} Sv - \lambda v &= 0 \\ (S - \lambda I_P)v &= 0 \end{aligned}$$

Posterior a ello, se procede con la resolución del sistema de ecuaciones para hallar los valores propios (λ) y los vectores propios v_i donde cabe recalcar que se pone la restricción de que las componentes al cuadrado del i -ésimo vector propio deben sumar 1.

El resultado esperado son P valores propios que indican cuanta varianza acumula cada una de las componentes en el nuevo sistema de coordenadas donde se van a proyectar los datos originales y P vectores propios que serán los ejes sobre los cuales se proyecten los datos y construyan las nuevas componentes donde los datos proyectados tendrán la particularidad de ser incorrelacionados.

Para hacer la proyección de los datos centrados en el nuevo sistema, basta con aplicar la siguiente operación matricial con los vectores propios encontrados:

$$Z = X_c v$$

Donde:

- Z Son los componentes creados a partir de los datos centrados y los vectores propios.
- X_c Es la matriz de datos centrados de tamaño $N \times P$.

- v es la matriz de vectores propios de tamaño $P \times P$.

Finalmente, se calcula la variabilidad explicada por cada componente mediante los valores propios como sigue:

$$VE = 100 * \frac{\lambda_i}{\sum_{i=1}^P \lambda_i}$$

Donde:

- VE Es la variabilidad explicada por el i -ésimo valor propio.
- λ_i Es el i -ésimo valor propio.

Cabe mencionar que los valores propios conservan la siguiente propiedad:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_P$$

El primer valor propio es mayor o igual al segundo valor propio, el segundo es mayor o igual al tercer y así sucesivamente. El primer valor propio asociado a la primer componente siempre sera el que mayor variabilidad explique (asumiendo que están ordenados de mayor a menor, esto ya es tema de notación y en que orden se realicen las operaciones de matrices).

De esta manera, podemos proyectar en apenas 2 o 3 nuevas componentes la información de tamaño P dimensional que ademas, tiene una alta varianza explicada.

5. Metodología

Para el presente reporte, se propone un marco donde se fija el parámetro del número de grupos k en 3, para el caso del algoritmo fuzzy se usa un parametro de difusividad $M = 1,9$. Para k-medias y para Fuzzy se van a comparar haciendo uso de la distancia Euclidiana y la distancia de Manhattan con un tope de 100 iteraciones y una tolerancia para la convergencia $\epsilon = 0,001$.

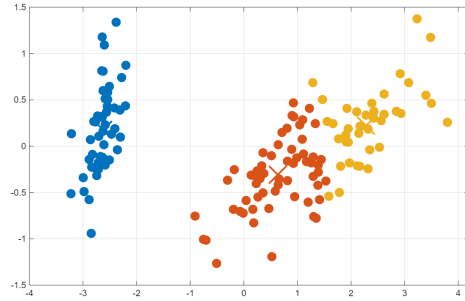
Una vez obtenidas las etiquetas de los clusters para ambos algoritmos, se aplicara PCA para reducir dimensionalidad y observar su comportamiento en un nuevo espacio de 2 dimensiones.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.6	3.4	1.4	0.3
5.0	3.4	1.5	0.2
4.4	2.9	1.4	0.2
4.9	3.1	1.5	0.1

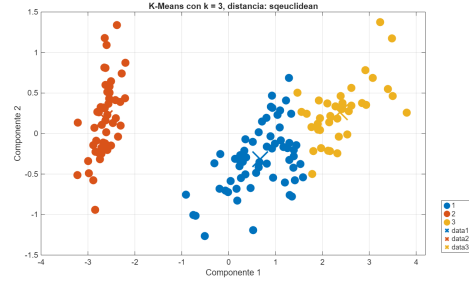
Cuadro 1: Matriz de variables de Iris.

6. Resultados

6.1. Resultados para la distancia Euclidiana



(a) Fuzzy C-means con distancia Euclidiana



(b) K-means con distancia Euclidiana

Figura 1: Comparación entre algoritmos con distancia Euclidiana

$$\begin{bmatrix} 5,0045 & 3,4155 & 1,4813 & 0,2533 \\ 5,8874 & 2,7578 & 4,3639 & 1,3990 \\ 6,7848 & 3,0544 & 5,6578 & 2,0553 \end{bmatrix}$$

(a) Vectores de medias - Fuzzy C-means

$$\begin{bmatrix} 5,9016 & 2,7484 & 4,3935 & 1,4339 \\ 5,0060 & 3,4280 & 1,4620 & 0,2460 \\ 6,8500 & 3,0737 & 5,7421 & 2,0711 \end{bmatrix}$$

(b) Vectores de medias - K-means

Figura 2: Comparación de los vectores de medias obtenidos por dos algoritmos

El algoritmo convergió en 18 iteraciones y a continuación se presentan las primeras 30 observaciones de la matriz de pertenencias del Fuzzy C-means.

0,9984	0,0011	0,0005
0,9848	0,0108	0,0044
0,9877	0,0087	0,0037
0,9788	0,0150	0,0062
0,9972	0,0020	0,0009
0,9538	0,0325	0,0137
0,9874	0,0088	0,0037
0,9998	0,0001	0,0001
0,9501	0,0352	0,0147
0,9896	0,0074	0,0030
0,9795	0,0144	0,0062
0,9957	0,0031	0,0013
0,9811	0,0133	0,0055
0,9443	0,0385	0,0172
0,9166	0,0564	0,0270
0,8732	0,0856	0,0412
0,9637	0,0250	0,0113
0,9984	0,0011	0,0005
0,9286	0,0502	0,0212
0,9874	0,0088	0,0038
0,9797	0,0145	0,0057
0,9911	0,0062	0,0026
0,9726	0,0187	0,0087
0,9873	0,0092	0,0035
0,9784	0,0157	0,0060
0,9832	0,0121	0,0047
0,9973	0,0019	0,0008
0,9965	0,0025	0,0010
0,9967	0,0023	0,0010
0,9874	0,0090	0,0036

6.2. Resultados para la distancia de Mahattan

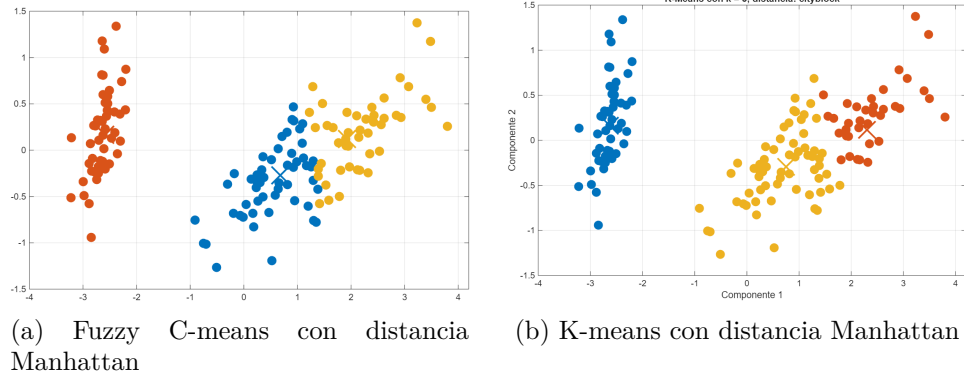


Figura 3: Comparación entre algoritmos con distancia Manhattan

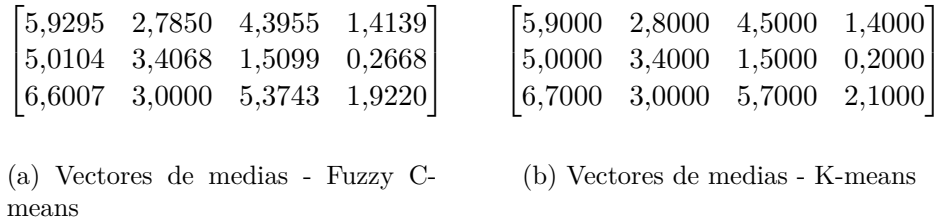


Figura 4: Comparación de vectores de medias ajustados con la referencia

El algoritmo convergió en 18 iteraciones y a continuación se presentan las primeras 30 observaciones de la matriz de pertenencias del Fuzzy C-means.

0,0019	0,9983	0,0010
0,0701	0,9980	0,0445
0,0159	0,9922	0,0083
0,0873	0,9967	0,0455
0,0051	0,9984	0,0035
0,0710	0,9810	0,0398
0,0264	0,9951	0,0119
0,0007	0,9999	0,0003
0,1409	0,9889	0,0687
0,0597	0,9989	0,0371
0,1168	0,9977	0,0551
0,0058	0,9978	0,0025
0,0567	0,9958	0,0322
0,0655	0,9658	0,0358
0,1977	0,9782	0,0998
0,2998	0,9759	0,1916
0,0609	0,9849	0,0351
0,0018	0,9985	0,0010
0,2190	0,9865	0,0931
0,0641	0,9985	0,0584
0,0393	0,9930	0,0141
0,0332	0,9985	0,0239
0,0232	0,9733	0,0128
0,0103	0,9923	0,0049
0,0155	0,9819	0,0070
0,1168	0,9989	0,0850
0,0023	0,9988	0,0011
0,0158	0,9994	0,0067
0,0053	0,9980	0,0022
0,0239	0,9956	0,0121
0,0349	0,9970	0,0195

7. Conclusiones

Tras la realización de la aplicación, se pudo observar que entre el algoritmo Fuzzy y el K-means clásico, existen ligeras diferencias al momento de clasificar a un par de sujetos que se encuentran muy en la frontera de pertenecer a un grupo o a otro. Para este caso en particular, la distancia utilizada no tuvo un efecto tan relevante.

8. Referencias

Referencias

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York.
- Edla, D. R., Lone, T., Tapas, N., and Kuppili, V. (2020). Analysis of high dimensional brain data using prototype based fuzzy clustering. *Clinical Epidemiology and Global Health*, 8(4):1110–1118.
- Johnson, R. A. and Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*. Pearson, Upper Saddle River, NJ, 6th edition.
- Nascimento, S., Mirkin, B., and Moura-Pires, F. (2000). A fuzzy clustering model of data and fuzzy c-means. In *Ninth IEEE International Conference on Fuzzy Systems. FUZZ- IEEE 2000 (Cat. No.00CH37063)*, volume 1, pages 302–307, San Antonio, TX, USA. IEEE.
- Wang, F. and Sun, J. (2015). Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, 29(2):534–564.

9. Anexos

Este reporte se envía con los códigos anexos que corresponden a:

1. El código en MATLAB del fuzzy cmeans y kmeans con las pruebas