Reconocimiento de Patrones Auto-Supervisado. Método de las Clases Cerradas

Dr. Argelio Victor de la Cruz Rivera[1], M. en C. María de los Ángeles Alonso Lavernia[1]

[1] Centro de Investigación en Tecnologías de Información y Sistemas
Universidad Autónoma del Estado de Hidalgo (CITIS-UAEH)
argelioc@uaeh.reduaeh.mx, marial@uaeh.reduaeh.mx

Resumen

Se presenta un nuevo algoritmo para resolver problemas de Reconocimiento de Patrones No Supervisado, que puede ser considerado como un método híbrido, por una parte, dada su capacidad para trabajar como método de partición o jerárquico y por otra, debido a su estrategia de funcionamiento que utiliza ideas provenientes de los métodos supervisados.

Se introduce el concepto de Clase Cerrada, y un algoritmo para el cálculo de éstas en una matriz de datos cualquiera, el cual se utiliza como base para el desarrollo de cuatro variantes de clasificación.

Entre las ideas que se utilizan, se encuentra realizar un Reconocimiento No Supervisado sobre los datos, utilizando como información de aprendizaje la propia pertenencia a las clases de cada individuo en el momento que se encuentre el proceso, de aquí que en cierto sentido la clasificación sea también auto-supervisada.

Los resultados pueden ser una partición de los datos en una cantidad de grupos predefinido o la representación mediante un Dendrograma, a través del cual se puede extraer una clasificación para los datos procesados.

1. Introducción

Los métodos de Reconocimiento de Patrones pueden ser divididos en dos grupos fundamentales: Reconocimiento Supervisado y Reconocimiento No Supervisado [9], [12], [13], aunque algunos autores también incluyen métodos de selección de variables y variantes de los anteriores [5], [12], [14], [15].

En general, se parte de suponer que se tiene una matriz de datos M, formada por "n" filas y "m" columnas, donde las filas representan los individuos observados y las columnas los parámetros o variables que han sido medidos a esos individuos, es decir:

$$M = \begin{pmatrix} X_{11} & \dots & X_{1m} \\ & \vdots & \\ X_{n1} & \dots & X_{nm} \end{pmatrix}$$

donde X_{ij} representa el valor del individuo i para la variable j ($i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$)

Los métodos de Reconocimiento No Supervisados tratan de resolver el problema de encontrar una clasificación adecuada de los individuos en "g" clases o grupos, de forma tal que la semejanza entre los integrantes de una clase sea puesta de manifiesto, al igual que las diferencias entre individuos de clases diferentes.

Por otra parte, los métodos de Reconocimiento Supervisados, parten de suponer que los individuos de la matriz M, ya se encuentran clasificados en "g" clases (por cualquier

vía) y el problema a resolver consiste en: clasificar otro conjunto de individuos, caracterizados por las mismas "m" variables, cuya pertenencia a las clases de M es desconocida. Es común en este caso denominar como Información de Aprendizaje a la matriz M.

En el presente trabajo, se introduce un tipo de método con características especiales, puesto que la tarea que desempeña está dirigida a resolver problemas de Reconocimiento No Supervisado, pero la forma en que se ejecuta la clasificación se asemeja a la de un método Supervisado [3], [4] y específicamente a la conocida regla de los k vecinos más cercanos [2].

La idea que se desarrolla consiste en considerar que la clasificación se ejecuta por pasos, de forma tal, que en cada momento los individuos ya clasificados se utilizan como información de aprendizaje para clasificar los restantes.

En el método propuesto se parte de suponer que al inicio del proceso de clasificación, cada individuo constituye una clase independiente y que al estilo de los métodos jerárquicos aglomerativos [1], [6], [7], [11], se ejecuten fusiones en pasos sucesivos según determinados criterios, hasta que la cantidad de clases se haga igual a un número "NC", que constituye un parámetro de entrada a los algoritmos.

Dada esta organización del proceso, es posible la representación de los pasos de clasificación a través de un Dendrograma, lo cual permitirá representar gráficamente configuración de los datos y determinar la cantidad de grupos si es desconocida.

Por otra parte, el método también hereda algunas características de los algoritmos partición [1], [7], [10], puesto que el resultado que se brinda es precisamente una partición de los individuos en una cantidad fija de clases.

La idea general del procedimiento se resume en los siguientes hechos:

1) Para cada grupo existente en el estado "e" de la clasificación (el concepto de estado se define formalmente en el epígrafe 3) se deben obtener los "KNN" individuos más semejantes entre los externos a la clase como tal. El número" KNN" es un dato externo que recibe el algoritmo.
2) Este proceso se ejecuta para cada integrante de la clase de forma individual y toman los "KNN" individuos de máxima semejanza, entre todos los obtenidos forma particular, como los elementos asociados al grupo completo.
3) La clasificación o proceso de fusión, se ejecutará supervisado por aquellas clases las que pertenezcan los individuos más semejantes.
4) La influencia en la clasificación de los "KNN" individuos, estará condicionada el orden que ocupen respecto a los valores de semejanzas, es decir, el individuo más cercano será el de mayor influencia y así sucesivamente en orden decreciente.

Partiendo de éstas ideas básicas, se definen un conjunto de conceptos y propiedades que permiten construir el método propuesto.

2. Vector de Votación

El Cálculo de la influencia mencionada en el cuarto punto se ejecuta correspondencia con un vector de votación que se define a continuación.

Utilizaremos las siguientes notaciones:

- "KNN" denotará el número de individuos más semejantes que serán considerados entre lo n individuos que forman la matriz de datos ($1 \leq KNN \leq n$).

- "KNNx" representará al conjunto formado por los KNN individuos más semejantes al individuo "x" ordenados de mayor a menor parecido, de acuerdo con alguna medida de la semejanza.
- "KNNx(j)" representará al individuo j del conjunto KNNx.
- "a" denota la cantidad de clases a las que pertenecen los individuos del conjunto KNNx. Se observa que $1 \leq a \leq KNN$ dado que más de un individuo puede pertenecer a la misma clase.

Definición 1.- Se llama vector de votación $V(x) = (V_1, V_2, \ldots, V_a)$ respecto a un individuo "x", en relación con las clases $C_1, C_2, \ldots, C_a$, a las que pertenecen los KNN individuos más semejantes a "x", a aquel cuyas componentes V_i se obtienen como sigue:

$$V_i = \sum_{j=1}^{KNN} CTR_{KNNx(j)}(C_i)$$

donde $CTR_{KNNx(j)}(C_i)$ denota la contribución del individuo j respecto a la clase C_i, con

$$CTR_{KNNx(j)}(C_i) = \begin{cases} 0 & \text{si } KNNx(j) \notin C_i \\ \dfrac{KNN-(j-1)}{KNN} & \text{si } KNNx(j) \in C_i \end{cases}$$

Como se puede observar aquellas clases que no estén representadas entre los KNN individuos más semejantes no tendrán votación. Por otra parte, se observa que al encontrarse el conjunto KNNx ordenado de acuerdo a la semejanza, el primer individuo será el de mayor contribución con votación 1 exactamente, mientras que los demás tendrán menor votación, hasta llegar al último cuya votación será 1/KNN.

Se observa también, que en la definición del vector de votación no se toma en cuenta semejanza real existente entre los individuos. Esto puede provocar que un grupo de individuos poco parecido aporte más, lo cual puede provocar resultados no deseados, como se observa en el siguiente ejemplo.

Ejemplo 1.- Si consideramos que KNN=4 y que el primer individuo del conjunto KNNx pertenece a la clase 1 y los restantes a la clase 2, el vector de votación es (1 , 1.5) lo cual inclina la clasificación hacia la clase 2. Sin embargo, pueden ocurrir dos casos como los siguientes:

Parecidos entre el individuo x y los 4 más semejantes

a) 0.9 , 0.88 , 0.87 , 0.86
b) 0.9 , 0.3 , 0.28 , 0.25

Se puede observar que la diferencia en parecido entre el primer individuo y los restantes, para el caso "a", no es sustancial, mientras que en el segundo caso si lo es, y sin embargo, el resultado es el mismo de acuerdo al vector de votación.

Tomando en cuenta esta situación se introduce una variante en la que se utiliza la medida de la semejanza para ponderar el cálculo del vector de votación como sigue:

Componentes del vector de votación ponderado $V_i = \sum_{j=1}^{KNN} CTR_{KNNx(j)}(C_i) * S(KNN_x(j), x)$

donde $S(KNN_x(j), x) \in [0,1]$ representa la semejanza entre el individuo x y el individuo KNNx(j) dentro del conjunto de los más parecidos a "x".

Lo anterior permite tomar en cuenta, tanto la posición entre los más parecidos, como la semejanza entre ellos.

Ejemplo 2.- Si tomamos como referencia los mismos datos del ejemplo 1 y obtenemos los vectores de votación ponderados se obtiene lo siguiente:

a) $V_1=1(0.9)=0.9$, $V_2=3/4(0.88) + 2/4(0.87) + 1/4(0.86)) = 1.31$, luego el vector de votación ponderado es $V(x) = (0.9 , 1.31)$

b) $V_1=1(0.9)=0.9$, $V_2=3/4(0.3) + 2/4(0.28) + 1/4(0.25)) = 0.4275$, luego el vector de votación ponderado es $V(x) = (0.9 , 0.4275)$

Se observa que en el primer caso la clase 2 sigue manteniendo el máximo votación, mientras que en el segundo caso la situación cambia, debido a los bajos valores de semejanzas con relación al primer individuo.

Se puede considerar que cada componente V_i del vector de votación representa medida de la cercanía del individuo "x" a la clase C_i y este hecho puede ser usado como un criterio para ejecutar la fusión de clases semejantes.

Lo anterior significa, que utilizando el vector V(x) se puede decidir cuando individuo "x" será reclasificado como integrante de aquella clase para la cual se obtuvo el máximo de votación, pero que a su vez el resto de los integrantes de la clase, donde estaba clasificado "x", también se verán afectados y por consiguiente reclasificados de la misma forma.

Tal acción provoca una disminución en el número de clases y la desaparición correspondiente al individuo en análisis.

Bajo determinadas condiciones puede ocurrir que algunas componentes del Vector de Votación sean iguales y por tanto se hace necesario establecer criterios solucionar este conflicto.

2.1. Criterios de utilización para el Vector de Votación

Existen dos posibilidades a la hora de utilizar el vector de votación en la fusión clases:

Criterio 1.- Máximo a la Izquierda. Siempre busca el máximo lo más a la izquierda posible del vector, es decir, si existen dos valores iguales se toma la clase del primer valor.

Criterio 2.- Máximo a la Derecha. Contrariamente al anterior, se toma el valor alejado dentro del vector.

En el primer criterio se hace énfasis hacia la importancia creciente de los primeros individuos, mientras que en el segundo, se prioriza la cantidad de individuos semejantes que aportan contribución sobre cada clase. Estos son criterios que introducir el usuario como datos.

Ejemplo 3.- Si KNN = 3, las contribuciones potenciales de cada uno de los individuos (sin considerar la semejanza) son las siguientes: 1, 2/3, 1/3, es decir el individuo semejante aporta 1 a la votación, el siguiente en importancia 2/3 y el último 1/3. Si el primer individuo pertenece a la clase C_1 y los otros dos a la clase C_2, el de votación quedaría como sigue: $V(x) = (1, 1)$, es decir existe la misma contribución hacia ambas clases, sin embargo, bajo el criterio Máximo a la Izquierda, se tomaría como clase más cercana C_1, con lo cual el primer individuo tiene prioridad, mientras que en el segundo caso, se tomaría C_2 lo cual expresa que al encontrarse

individuos, entre los KNN pertenecientes a una misma clase, esto se tendrá en mayor consideración.

3. Clases Cerradas

El método que se propone plantea la utilización del vector de votación para obtener clases con determinadas características de estabilidad. Comenzaremos analizando como se ejecuta el proceso.

Definición 2.- Se llama Iteración al proceso de análisis de todos los individuos de la matriz de datos, ordenados según los valores del individuo de mayor semejanza, para su posible reclasificación según el concepto de votación.

Respecto a esta definición se necesita mencionar dos aspectos importantes:

1. El análisis de los individuos es ordenado por las semejanzas al primer integrante de los KNN correspondiente, ya que se van tomando para el análisis los individuos cuyos valores de semejanza al resto sean los mayores.
2. En una misma iteración un individuo pudiera ser relocalizado más de una vez, debido a la característica hereditaria que le brinda un individuo a los integrantes de su grupo.
3. El proceso de relocalización de individuos implica fusiones entre las clases.

Definición 3.- Un individuo i se llama Estable en la iteración "t" si no es relocalizado (cambiado de clase) en t.

Definición 4.- Un individuo i se llama Absolutamente Estable si luego de alcanzar la estabilidad en la iteración t, éste no tiene posibilidades de ser relocalizado en ninguna iteración posterior.

El siguiente hecho es una consecuencia directa de las definiciones anteriores.

Proposición 1.- Si un individuo i que pertenece a la clase C_L es Absolutamente Estable, entonces todos los integrantes de C_L también lo son. A C_L se le denomina Clase Absolutamente Estable.

Demostración: Utilizaremos el método de reducción al absurdo, para lo cual supondremos que no se cumple la afirmación planteada para tratar de obtener una contradicción. Supongamos entonces, que a pesar de existir un individuo i absolutamente estable en la clase C_L ($i \in C_L$), existe otro individuo $j \in C_L$, que no lo es. Si esto ocurre, entonces el individuo "j" puede ser reclasificado por no ser absolutamente estable (definición 4) y por tanto, esto significa que al ser reclasificado en una iteración t, también el resto de los integrantes de su clase sería reclasificados, incluyendo el "i" que de hecho no sería entonces absolutamente estable. De lo anterior se obtiene una contradicción con la suposición inicial, quedando demostrada así la afirmación planteada.

De lo anterior se deduce que la propiedad de estabilidad absoluta para un individuo, se puede obtener directamente de la estabilidad del resto, es decir, analizando si algún individuo de su clase puede ser reclasificado o no.

Por otro lado la propiedad de estabilidad absoluta puede ser enfocada según la siguiente propiedad:

Proposición 2.- Una clase C_L es Absolutamente Estable sí y sólo si, el valor máximo de cada vector de votación asociado a sus miembros, se corresponde con individuos de la propia clase C_L.

La demostración de esta proposición se obtiene de la aplicación de las definiciones y 4 respectivamente.

Definición 5.- Sea "C_L" una Clase Absolutamente Estable de M. Si para todo individuo $I_j \in M$ no incluido en C_L ($I_j \notin C_L$) se cumple que el valor máximo del vector votación correspondiente V(Ij) no coincide con C_L, se dice que ésta forma una Clase Cerrada de Primer Orden o Pura.

De la definición se observa, que la condición de Clase Cerrada implica, que ninguno de sus miembros tiene posibilidades de ser reclasificado en iteraciones posteriores del estado actual y que a su vez, ningún individuo externo a la clase puede ser incluido ésta. Por tanto, una vez alcanzada esta propiedad una clase permanecerá invariante iteraciones posteriores.

En la siguiente proposición se analiza la cantidad de iteraciones necesarias para obtener una configuración de Clases Cerradas.

Proposición 3.- El número de iteraciones máximo necesario para obtener todas Clases Cerradas para KNN $\in \{1,2\}$ es exactamente 1.

Demostración: Analicemos que ocurre si KNN=1. Para demostar que en una iteración se obtienen todas las Clases Cerradas, es suficiente demostrar, que basta con una iteración para que todos lo individuos adquieran la propiedad de estabilidad absoluta, puesto que al cumplirse esto ningún individuo puede ser reclasificado y por tanto, todas las clases son cerradas. Ahora bien, como el proceso de clasificación se ejecuta relocalizando a cada individuo en la clase donde se encuentre el más semejante a si se ejecuta una segunda iteración no habrá posibilidades de cambios para individuo alguno, pues ya todos se encuentran en la misma clase de su individuo más semejante y por tanto todos son absolutamente estables. Para KNN=2 no hay diferencias, pues la primera componente del vector, siempre será mayor que segunda, lo cual implica que esta última no afecta el resultado con relación primero.

Analicemos qué ocurre cuando se obtienen todas las Clases Cerradas de Primer Orden para una matriz de datos M. Para ello introduciremos el concepto de Estado de Clasificación.

Definición 6.- Se llama Estado de la Clasificación de Primer Orden, a una configuración tal que, todas las clases existentes son cerradas de primer orden.

No obstante, necesitamos analizar qué ocurre cuando se obtiene un estado de primer orden y el número de clases obtenidas 'c' es mayor que el número de clases buscada (NC), lo cual significa que se ha obtenido un estado parcial de la clasificación.

El cambio de un estado a otro se ejecuta de la siguiente forma:

1.- Se obtienen los KNN individuos más semejantes a cada individuo de la matriz datos, considerando solamente aquellos que no pertenezcan a su misma clase.

2.- Para cada clase *l* se consideran las KNN individuos obtenidos para sus integrantes. Si N_l denota la cantidad de elementos de esta clase, entonces se obtiene subconjunto de individuos que oscila entre KNN y KNN*N_l.

3.- Se extraen los KNN de mayor semejanza entre los elementos de este conjunto obtenido y estos serán considerados como los asociados a cada uno de integrantes de la clase.

Bajo esta nueva situación se puede proceder como antes para obtener nuevas Clases Cerradas formadas por fusión de las anteriores. Esta idea nos lleva a la siguiente definición.

Definición 7.- Se llama Clase Cerrada de Orden "e" (e>1), a aquella obtenida producto de la fusión de 2 o más Clases Cerradas de orden e-1 y que cumple con la definición 5. El orden "e" representa el estado actual de la clasificación.

Tomando como base los conceptos explicados se puede implementar un algoritmo de clasificación con varias variantes.

4. Algoritmo de clasificación utilizando el concepto de Clase Cerrada

Utilizando los conceptos estudiados, se puede construir un algoritmo de clasificación con diferentes variantes y formas de presentar los resultados.

Variante 1.- Encontrar las clases cerradas de primer orden.

Variante 2.- Encontrar una partición de los individuos en Nc clases.

Variante 3.- Obtener el último nivel de clases cerradas posible donde el número de clases sea diferente de 1.

Variante 4.- Obtener un dendrograma del proceso de fusión de los individuos y clases, partiendo de n clases y terminado el proceso en una.

De lo anterior se puede observar, que el resultado que se busca no tiene que ser, necesariamente, una configuración de clases cerradas, aunque en todos los casos se utilizará este concepto para obtener los resultados.

Por otro lado, serán necesarios varios parámetros como entrada.

1. Valor de KNN.
2. Criterio para el vector de votación si KNN > 2.
3. Función de semejanza a utilizar.
4. Tipo de resultado buscado (Variante 1, 2, 3 o 4).

Los pasos generales del algoritmo se pueden resumir en los siguientes pasos:

Paso 0. Considerar que cada uno de los "n" individuos a clasificar constituye un grupo individual, es decir, al inicio existirán n clases.

Paso 1. Calcular la matriz de semejanza entre los "n" individuos. Esto puede ser ejecutado opcionalmente.

Paso 2. Obtener para cada individuo "i" de la matriz de datos, los KNN más semejantes (usando la matriz de semejanza o calculándola en este momento), tomando en cuenta que no pertenezcan a su misma clase.

Paso 3. Obtener para cada clase *l* los KNN individuos más semejantes entre los obtenidos para cada uno de sus miembros y considerar que a todos los integrantes de forma individual le corresponden estos, en lugar de los obtenidos en el paso 2.

Paso 4. Obtener el vector de votación según el criterio definido y a partir de este, la clase más parecida a cada individuo.

Paso 5. En correspondencia con el tipo de resultado buscado, ejecutar lo siguiente:

Variante 1.- Obtener las clases cerradas e ir al paso 7.

Variante 2.- Obtener las clases cerradas para el estado actual "e", tomando en cuenta que el proceso se ejecute hasta el final, sólo si el número de clases actual "Na" es mayor que el número de clases buscado "Nc"; puesto que en caso contrario se

salta al paso 7. Esto significa que nunca se obtendrá un número de clases menor que el buscado. Si el proceso finaliza ir al paso 6.

Variante 3.- Guardar el estado actual de la clasificación "e". Obtener un nuevo estado e+1 obteniendo las clases cerradas. Si el número de clases es mayor que 1 ir paso 2. Si el número de clases es uno considerar como estado actual el "e" e ir al paso 7.

Variante 4.- Obtener las clases cerradas para el estado actual "e". Si el número clases es uno ir al paso 7, mientras que en caso contrario ir al paso 2.

Paso 6. Si el número de clases cerradas en el estado actual es igual a Nc entonces obtuvo el estado final concluyendo la clasificación, mientras que en caso contrario, se regresa el Paso 2. Este paso solo se ejecuta para la variante 2.

Paso 7. Para las variantes 1, 2 y 3 se presenta como resultado el estado actual de clasificación. Para la variante 4 se brinda como resultado el Dendrograma resultante del proceso ejecutado.

5. Un Ejemplo

Para mostrar el funcionamiento del algoritmo vamos a considerar una matriz de datos hipotética muy sencilla, la cual está formada por 10 individuos caracterizados por variables numéricas. En la figura 1 se presentan los datos y su representación espacial, mientras que en la tabla 1 se muestra la correspondiente matriz de Distancias Euclidianas, como medida de la semejanza.

	1	2
1	4	2
2	9	6
3	7	7
4	6	6
5	1	1
6	7	4
7	4	1
8	5	1
9	2	2
10	3.5	1.5

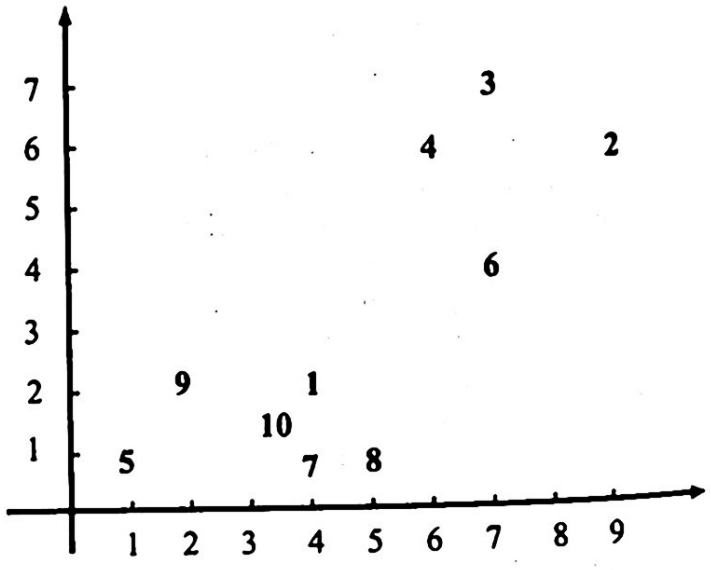

Figura 1. Matriz de Datos y su representación gráfica.

La tarea planteada consiste en clasificar los individuos en 2 grupos utilizando variante 2 del algoritmo presentado. Esto significa que el objetivo no es encontrar clases cerradas exactamente. Por simplicidad utilizaremos como parámetro KNN = así no será necesario utilizar el vector de votación, además de necesitarse una iteración para obtener cada estado de la clasificación (ver proposición 3).

Para ejecutar el algoritmo nos apoyaremos en dos tablas, en las que se resumen distintos estados e iteraciones producto de la aplicación del método.

En la tabla 2 se presenta para cada individuo, el más semejante en cada estado de la clasificación y en la tabla 3, la clase en que se encuentra clasificado en cada estado e iteración específica.

	1	2	3	4	5	6	7	8	9	10
1	0									
2	6.40	0								
3	5.83	2.24	0							
4	4.47	3	1.41	0						
5	3.16	9.43	8.49	7.07	0					
6	3.61	2.83	3	2.24	6.71	0				
7	1	7.07	6.71	5.39	3	4.24	0			
8	1.41	6.40	6.32	5.1	4	3.61	1	0		
9	2	8.06	7.07	5.66	1.41	5.39	2.24	3.16	0	
10	0.71	7.11	6.52	5.15	2.55	4.3	0.71	1.58	1.58	0

Tabla 1. Matriz Triangular Inferior de Distancias Euclidianas.

	Estado	Individuos 1	2	3	4	5	6	7	8	9	10
Individuos más Semejantes	1	10	3	4	3	9	4	10	7	5	1
Distancia		0.71	2.24	1.41	1.41	1.41	2.24	0.71	1	1.41	0.71
Individuos más Semejantes	2	9	1	1	1	10	1	9	9	10	9
Distancia		2	6.4	5.84	4.47	2.55	3.61	2.24	3.16	1.58	1.58

Tabla 2. Individuos más semejantes en cada estado de la clasificación.

	Estado	Iteración	Individuos 1	2	3	4	5	6	7	8	9	10
Clases	1	0	1	2	3	4	5	6	7	8	9	10
		1	10	4	4	4	9	4	10	10	9	10
Clases	2	1	10	4	4	4	10	4	10	10	10	10

Tabla 3. Clase a la que pertenece cada individuo después de terminada la iteración.

Como se puede observar en la Tabla 3, para el estado inicial (iteración 0) se considera que cada individuo constituye una clase independiente.

A continuación se comienza el proceso de reclasificación analizando para cada individuo a qué clase pertenece el más semejante a él.

El proceso iterativo se organiza utilizando los parecidos obtenidos entre los individuos, por tanto el orden de análisis para estos, en la primera iteración es: 1, 7, 10, 8, 3, 4, 5, 9, 2, 6.

Lo anterior significa que el orden en que se coloquen los individuos a clasificar no influye en los resultados, dado que el proceso iterativo es ordenado por las propias

semejanzas entre los individuos. Esto garantiza que las fusiones que se ejecuten siempre sean entre aquellos individuos o clases de mayor parecido.

Lo anterior se observa claramente en las clases que prevalecen después de la primera iteración (tabla 3, fila de la iteración 1).

Por ejemplo se observa (tabla 2, estado 1) que el individuo más semejante a 1 10, pero como este último pertenece a la clase 10, entonces se reclasifica el individuo pasando a formar parte de esa misma clase, seguidamente se tiene que el individuo semejante al 7 es a su vez el 10, luego también queda incluido en esta clase.

El proceso continúa en el orden señalado hasta completar una iteración. El resultado hasta aquí, son tres clases cerradas, la primera formada por los individuos 2, 3, 4 y segunda por 5 y 9 y la tercera formada por 1, 7, 8 y 10 (tabla 3, estado 1, iteración 1).

Como el objetivo planteado es obtener una partición en dos clases, se necesita obtener otro estado de la clasificación.

En la tabla 2, filas del estado 2, se observa el resultado de obtener los individuos semejantes para cada uno, tomando en cuenta que no pertenezcan a su clase.

Se puede apreciar que el proceso iterativo comienza en el individuo 9, el cual, con el resto de los integrantes de su grupo (en este caso 5) se unen al grupo 10. Como e este paso se obtienen dos clases el proceso se detiene obteniéndose como resultado:

Grupo 1: {1, 5, 7, 8, 9, 10}

Grupo 2: {2, 3, 4, 6}

6. Construcción del Dendrograma

Como se había mencionado, otra característica interesante del método propuesto posibilidad de representar el resultado mediante un Dendrograma, lo cual es muy como método exploratorio de los datos en caso de ser completamente desconocido estructura de clases [9].

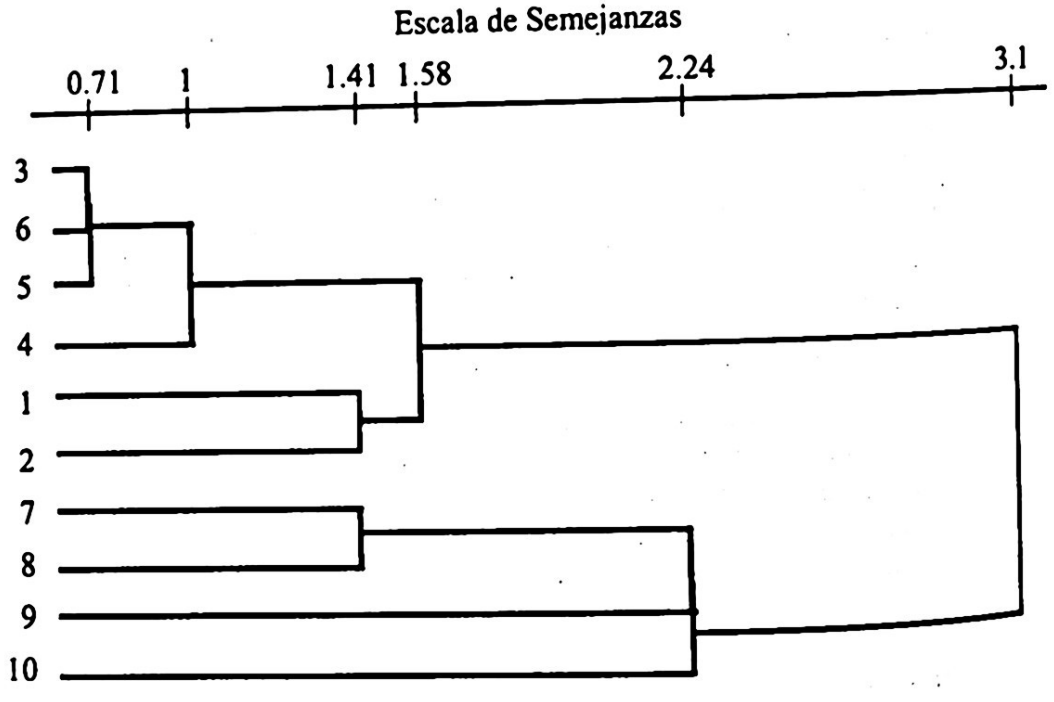

Figura 2. Dendrograma del proceso de agrupación.

En la figura 2 se muestra el Dendrograma resultante del proceso de clasificación efectuado en el conjunto de datos del ejemplo anterior.

A la izquierda se ubican los individuos y en la parte superior la escala que mide los valores de semejanza de los pasos de fusión ejecutados en el proceso de clasificación.

Tomando como punto de partida las tablas 2 y 3, se puede seguir el proceso de construcción del Dendrograma como se muestra en la tabla 4.

Pasos de Fusión	Individuos agrupados	Semejanza
1	3 con 6	0.71
2	(3, 6) con 5	0.71
3	(3, 6, 5) con 4	1
4	1 con 2	1.41
5	7 con 8	1.41
6	(7, 8) con 9	2.24
7	(7, 8, 9) con 10	2.24
8	(3, 6, 5, 4) con (1, 2)	1.58
9	(3, 6, 5, 4, 1, 2) con (7, 8, 9, 10)	3.1

Tabla 4. Proceso de construcción del Dendrograma.

8. Conclusiones

Como se ha podido apreciar, el método propuesto representa una herramienta para abordar el problema de reconocimiento no supervisado que utiliza algunas ideas híbridas provenientes de diversos tipos de métodos.

Así podemos reconocer la estrategia utilizada por los métodos jerárquicos de considerar a cada individuo como grupo independiente, para posteriormente ejecutar sucesivas fusiones, mientras que el resultado constituye una partición de los individuos en una cantidad de clases que preferentemente debe ser conocida.

Por último, la idea de utilizar los mismos individuos de la matriz para supervisar la clasificación y específicamente los KNN más semejantes, recuerda la forma en que trabajan los métodos de reconocimiento supervisado.

El uso de la matriz de semejanza como punto de partida presenta la dificultad propia del crecimiento cuadrático que ésta experimenta con el aumento de la cantidad de individuos a clasificar, pero presenta la ventaja de contar con toda la información relativa a todas las semejanzas dos a dos entre los individuos, disminuyendo sensiblemente la cantidad de cálculos, no obstante, un algoritmo utilizando directamente la Matriz Observacional se encuentra también implementado, pero ello no varia la teoría presentada.

Es de destacar que estamos en presencia de un método con el que se puedan abordar problemas donde las variables pueden ser tanto numéricas como no numéricas, permitiendo incluso la existencia de información incompleta [4], [8].

Es importante, mencionar que este método permite obtener una historia basada en el concepto de clases cerradas, en la cual se muestra, utilizando el Dendrograma, la clasificación de los individuos de un extremo a otro, es decir, partiendo de considerar

cada individuo como un grupo independiente, hasta considerarlos todos como miembros de uno solo.

Finalmente, es de mencionar que se han realizado algunos estudios comparativos de este método con algoritmos jerárquicos y de partición clásicos, utilizando datos simulados y casos reales conocidos, pero dada la extensión del trabajo estos resultados no se presentan.

Es de destacar, que en todos los casos se observó un mejor comportamiento utilizando el concepto de clases cerradas que se propone.

Referencias

[1] Anderberg, M. R.: "Cluster Analysis for applications", Academic Press, New York (1973).

[2] Cover, T.M., Hart, P.E.: "Nearest Neighbor Pattern classification". IEEE trans. Information Theory, IT-13 Pp. 21-27 (1967)

[3] De la Cruz, A. V.: "Un sistema de algoritmos para reconocimiento de patrones supervisado con posibilidades de auto-aprendizaje. Ejemplos de aplicación". Jornada Científica del Instituto de Geofísica y Astronomía A.C.C. 18-20 Octubre 1984

[4] De la Cruz, A. V.: "SRPS: Un sistema para reconocimiento de patrones supervisado". Informática'90, Reporte de participación. Pp. 227-240 (1990)

[5] De la Cruz, A. V.: "Reconocimiento Semi-Supervisado. Método de las Clases Cerradas.". International Conference, Science and Technology for Development, CIMAF'95. La Habana, Cuba, 23-27 Enero (1995)

[6] Dhillon, I.S., Modha, D.S.: "A Data-Clustering Algorithm on Distributed Memory Multiprocessors". Large-Scale Parallel Data Mining. Lecture Notes in Artificial Intelligence 1759. Subseries of Lecture Notes in Computer Science. Edited by J.G. Carbonell and J. Siekmann. Pp. 245-260 (2000)

[7] Everitt, Brian: "Cluster Analysis", Social Science Research Council (1974)

[8] Gower, J.C.: "A general coefficient of similarity and some of its properties" biometrics Vol.27, No.4, Pp. 857-874 (1971)

[9] Han, J., Kamber, M.: "Data Mining Concepts and Techniques". Morgan Kaufmann Publishers. 550 Pp. (2000)

[10] Hartigan, J. A.: "Clustering Algorithms". Wiley (1975)

[11] Johnson, E.L., Kargupta, H.: "Collective, Hierarchical Clustering from Distributed, Heterogeneous Data". Large-Scale Parallel Data Mining. Lecture Notes Artificial Intelligence 1759. Subseries of Lecture Notes in Computer Science. Edited by J.G. Carbonell and J. Siekmann. Pp. 221-244 (2000)

[12] Kanal, L.N., Dattatteya, G.R.: "Pattern Recognition". Encyclopedia of Artificial Intelligence. Vol. 2. Second Edition. John Wiley & Sons. Pp. 1116-1129 (1992)

[13] Ripley, B. D.: "Pattern Recognition and Neural Networks": Cambridge UNIVERSITY PRESS. 403 Pp. (1996)

[14] Ruiz, J., Guzmán, A., Martínez, J. F.: "Enfoque lógico combinatorio reconocimiento de patrones". Instituto Politécnico Nacional, México (1999)

[15] Schalkoff, R.: "Pattern Recognition Statistical, Structural and Neural Approaches" John Wiley and Sons, Inc. 364 Pp. (1992)