Algoritmo de agrupamiento basado en patrones utilizando árboles de decisión no supervisados

Andres Eduardo Gutierrez Rodríguez, Milton García Borroto,
José Francisco Martínez Trinidad

Reporte Técnico No. CCC-12-002
21 de noviembre de 2012

Luis Enrique Erro 1
Sta. Ma. Tonantzintla,
72840, Puebla, México.

Algoritmo de agrupamiento basado en patrones utilizando árboles de decisión no supervisados

Andres Eduardo Gutierrez Rodríguez, Milton García Borroto, José Francisco Martínez Trinidad

Coordinación de Ciencias Computacionales,
Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique Erro 1, Sta. Ma. Tonantzintla,
72840, Puebla, México
E-mail: andres@ccc.inaoep.mx; {ariel,fmartine}@inaoep.mx

Resumen. En la clasificación no supervisada muchas veces se desea explicar los resultados obtenidos, más allá de solo enumerar los objetos que pertenecen a cada agrupamiento. Los algoritmos de agrupamiento basado en patrones permiten explicar los resultados a partir de propiedades (patrones) que cumplen los objetos de cada grupo. Una estrategia comúnmente utilizada consiste en encontrar primero los patrones y después utilizarlos para agrupar los objetos, sin embargo, los algoritmos que siguen esta estrategia presentan algunas limitaciones, como por ejemplo que utilizan un proceso de discretización para poder trabajar con datos numéricos; además, inicialmente se obtienen todos los patrones frecuentes y después se aplica algún proceso de filtrado. Posteriormente, en la fase de agrupamiento usando estos patrones se utilizan estrategias muy costosas computacionalmente para construir los agrupamientos. En esta propuesta de investigación doctoral se plantea desarrollar un algoritmo para la extracción de un subconjunto reducido de patrones adecuados para agrupar, sin discretizar previamente los datos numéricos. Basado en esos patrones se pretende desarrollar un algoritmo de agrupamiento eficiente y eficaz que sea capaz de trabajar con datos mezclados e incompletos. Como resultado preliminar, se desarrolló un nuevo algoritmo de extracción de patrones a partir de un bosque de árboles de decisión no supervisados. Adicionalmente, se definió una medida para evaluar la calidad de los grupos de patrones obtenidos. Las comparaciones experimentales muestran el buen desempeño del algoritmo propuesto al agrupar con los nuevos patrones extraídos, comparado contra la alternativa convencional de calcular todos los patrones frecuentes.

Palabras clave. Clasificación no supervisada, extracción de patrones, agrupamiento basado en patrones.

1 Introducción

La clasificación no supervisada [39] es un área dentro del reconocimiento de patrones [3] que se ha aplicado en múltiples disciplinas como aprendizaje automático [2], procesamiento de textos [4], procesamiento digital de imágenes [5], entre otras. Los algoritmos de clasificación no supervisada también se conocen como algoritmos de agrupamiento, durante la investigación doctoral emplearemos este segundo término.

El proceso de agrupamiento consiste en, dado un conjunto de objetos sin etiquetar, formar grupos de objetos siguiendo un determinado criterio [22]. En la literatura se han definido diferentes criterios de agrupamiento:

- para Everitt en 1974 [34] los grupos deben tener una densidad de puntos relativamente alta,
- para Jain y Dubes en 1988 [35] el análisis de grupos consiste en encontrar la estructura intrínseca en los datos ya sea como grupos de individuos o como jerarquía de grupos,
- Pal y Bezdek en 1995 [36] afirman que en un conjunto de objetos se pueden identificar un número determinado de subgrupos naturales,
- en 2001, Halkidi et al. [29] expresaron que los objetos de un grupo se parecen más entre sí que a objetos de otros grupos,
- Bezdek y Hathaway en 2003 [37] establecen que el proceso de agrupamiento consiste en particionar un conjunto de objetos en grupos de objetos similares.

No obstante la diversidad de criterios de agrupamiento, el más utilizado es el basado en el parecido entre los objetos, de manera que los objetos de un grupo se parezcan más entre sí que a objetos de otros grupos [23]. Por lo general, el parecido entre los objetos se determina a partir de una función de comparación.

Debido a la diversidad de los criterios de agrupamiento seguidos por diferentes autores, validar los resultados obtenidos es un proceso subjetivo, que depende del criterio establecido y de la manera en que se mide el cumplimiento de ese criterio. Además, en áreas de aplicación como agricultura [8], bioinformática [9] y procesamiento de textos [10] es necesario explicar el resultado del agrupamiento, más allá de solo evaluar el parecido entre los objetos según la función de comparación. En estos casos los enfoques tradicionales de agrupamiento no brindan una solución adecuada.

Un enfoque diferente para el agrupamiento se basa en describir los agrupamientos a partir de conceptos o patrones que relacionan a los objetos [6, 7]. Este enfoque es conocido como agrupamiento basado en patrones. En la literatura encontramos tres estrategias fundamentales para este enfoque, las que presentan las siguientes limitaciones:

- Construir los patrones a la vez que se agrupan los objetos [2, 11]
    - Los algoritmos son costosos computacionalmente y no obtienen buenos resultados.
- Agrupar primero los objetos y luego encontrar patrones [21, 26]
    - Debido a que los objetos se agrupan usando una función de comparación, puede haber incompatibilidad con los patrones que se extraen en un siguiente paso, los cuales pueden caracterizar a objetos de grupos diferentes, aun cuando éstos no sean

parecidos de acuerdo a la función de comparación que se usó para formar el agrupamiento.

- Extraer todos los patrones frecuentes y a partir de ellos agrupar los objetos [7, 10, 13, 14, 16]
    - Esto hace que el proceso de agrupamiento sea ineficiente y que se tomen en cuenta patrones que pueden no ser adecuados para agrupar.

Adicionalmente, una característica importante en la mayoría de los algoritmos pertenecientes a las tres estrategias antes descritas es que no permiten trabajar con datos mezclados.

1.1 Problema a resolver

La presente investigación doctoral se enfoca en la tercera estrategia, con la cual se han reportado buenos resultados [7, 10, 13, 14, 16], en comparación con las otras dos estrategias. Sin embargo, esta tercera estrategia tiene como una de sus limitaciones importantes la cantidad de patrones extraídos. Es por eso que el problema que se plantea resolver es utilizar patrones para agrupar objetos, extrayendo solo un subconjunto reducido de patrones que sean adecuados para agrupar, en problemas con datos mezclados e incompletos; sin realizar discretización previa de los datos numéricos, la cual puede producir pérdida de información. Asimismo, es importante desarrollar algoritmos de agrupamiento eficientes a partir de los patrones encontrados, ya que los algoritmos actuales son de un alto costo computacional.

2 Conceptos básicos

2.1 Representación de objetos

Sea $DS = \{O_1,...,O_n\}$ un conjunto de objetos. Cada objeto O_i es descrito por un conjunto de atributos $X = \{x_1,...,x_m\}$. Cada atributo x_j toma valores en un conjunto admisible de valores V_j, $x_j(O_i) = v_{jk} \in V_j, j = 1,...,m, i = 1,...,n$, siendo $x_j(O_i)$ el valor del atributo x_j en el objeto O_i. Los atributos pueden ser de diferentes tipos dependiendo de la naturaleza del conjunto V_j, $j = 1,...,m$. Los tipos de valores pueden ser categóricos, numéricos e incluso ausencia de información; es por eso que consideramos al conjunto de objetos *DS* como mezclado e incompleto.

2.2 Comparación entre objetos

Una función de comparación de objetos se define como $\Gamma : X^2 \to S$, siendo S un conjunto completamente ordenado. Una función de comparación de objetos se dice que es de similaridad si al comparar un objeto con sigo mismo devuelve el máximo valor dentro de S. Se dice que es de disimilaridad en caso contrario.

2.3 Patrones

Un patrón P es una expresión que describe o caracteriza a un subconjunto de objetos. Un patrón está compuesto por una conjunción de propiedades $p = (x_j \ op \ v_{jk})$, donde op es un operador relacional; por simplicidad consideramos solo $\leq$, >, =. Por ejemplo, una expresión que caracterice a un conjunto de jóvenes talentos de baloncesto puede ser [($Edad \leq 20$) $\wedge$ ($Estatura > 190$)].

Decimos que un objeto es "caracterizado" por un patrón si el objeto cumple todas las propiedades del patrón; en este caso se dice que el patrón "cubre" al objeto. El "soporte" de un patrón es la fracción de objetos que son caracterizados por él. Un patrón es frecuente si su "soporte" es mayor que un umbral dado.

2.4 Algoritmos de agrupamiento

El principal objetivo de los algoritmos de agrupamiento es, dado una colección de objetos, formar k grupos de objetos $\{G_1,..G_k\}$ siguiendo un determinado criterio [22]. Uno de los algoritmos de agrupamiento (no basado en patrones) más populares es el *K-Means* [29].

Por otra parte, los algoritmos de agrupamiento basado en patrones generan un conjunto de patrones P_k para cada grupo G_k, de manera que cada conjunto de patrones describe cada grupo. Por ejemplo, en la Tabla 1 se muestran los datos de 10 jugadores de baloncesto; un buen agrupamiento es separar los jugadores en jóvenes con talento y sin talento.

Tabla 1. Jugadores de baloncesto (IMC = Índice de masa corporal).

Jugador	Edad	Estatura	IMC	Habilidad	Puntería
1	20	193	22.01	8	8
2	17	181	22.59	10	9
3	23	185	24.25	4	8
4	18	199	22.98	8	10
5	20	194	23.65	7	9
6	23	190	23.55	5	6
7	17	197	23.19	9	7
8	19	188	24.33	7	5
9	22	186	26.59	6	4
10	25	196	24.99	4	6

Los patrones que describen a esos grupos se muestran a continuación.
Patrones de jugadores jóvenes con talento:

- [(*Habilidad* > 6) $\wedge$ (*Puntería* > 6)]. Cubre 5 jugadores: 1, 2, 4, 5 y 7.
- [(*Edad* $\leq$ 20) $\wedge$ (*IMC* $\leq$ 23.65)]. Cubre 5 jugadores: 1, 2, 4, 5 y 7.
- [(*Edad* $\leq$ 20) $\wedge$ (*Estatura* > 190)]. Cubre 4 jugadores: 1, 4, 5 y 7.

Patrones de jugadores sin talento:

- [(*Edad* > 20)]. Cubre 4 jugadores: 3, 6, 9 y 10.
- [(*IMC* > 23.65)]. Cubre 4 jugadores: 3, 8, 9 y 10.

2.5 Validación de los resultados de los algoritmos de agrupamiento

Similar a lo que ocurre con los criterios de agrupamiento, en la literatura se han reportado múltiples criterios para el proceso de validación de los resultados de los algoritmos de agrupamiento, estos criterios evalúan la calidad de los agrupamientos. Para Pal y Bezdek en 1995 [36], la validación de los agrupamientos depende de qué significa un "buen agrupamiento", que se relaciona con el criterio usado para agrupar. Una manera de medir la calidad de los agrupamientos es a través de las medidas o índices de validación. Los índices de validación de agrupamientos se pueden dividir en dos grandes grupos: medidas internas y medidas externas [29]. Las medidas internas evalúan propiedades del agrupamiento como "compacidad" y "separación" de los objetos en los

agrupamientos construidos, por lo general basándose en la similaridad entre los objetos. Las medidas externas evalúan qué tanto se acerca el resultado del agrupamiento a un resultado "ideal" previamente determinado (por lo general las clases predefinidas de una Base de datos).

Los resultados preliminares de esta propuesta de investigación doctoral se evalúan utilizando una de las medidas externas más populares [38], *Rand Statistic* [30], definida como:

$$R = (SS + SD) / M \quad (1)$$

donde M es la cantidad de pares de objetos posibles en los datos, SS la cantidad de pares de objetos que pertenecen a la misma "clase" y quedaron agrupados en el mismo grupo, y SD la cantidad de pares de objetos que no pertenecen a la misma "clase" y que no quedaron agrupados en el mismo grupo. El objetivo es evaluar qué tanto se acercan los agrupamientos a las clases ya definidas en las bases de datos. Los valores de la medida están acotados en el intervalo [0,1]; los valores cercanos a 1 indican que el agrupamiento es correcto.

2.6 Inducción de árboles de decisión no supervisados

Un árbol es un grafo dirigido sin ciclos, donde cada nodo tiene un nodo padre, excepto un nodo especial llamado raíz que no tiene padre. Por lo general, los árboles se utilizan para representar jerarquías.

Los árboles de decisión son clasificadores supervisados ampliamente utilizados [31]. A partir de un conjunto de objetos previamente clasificados, mediante un proceso de inducción se construye un árbol de decisión buscando que los nodos hijos sean lo más “puros” posible, para disminuir el error de clasificación. Para clasificar un objeto nuevo, se recorre el árbol según las propiedades que cumple el objeto, hasta llegar a un nodo hoja (nodo sin hijos) donde se le asigna la clase (clase mayoritaria en ese nodo).

En el caso de los árboles de decisión no supervisados [20] no se tienen en cuenta las clases, porque los objetos no están etiquetados. Así, cada nodo padre es dividido según algún criterio, que puede ser incluso la evaluación de un índice de validación de agrupamientos. El objetivo es que en el árbol quede representado de manera correcta el criterio de agrupamiento. En los resultados preliminares se presenta un algoritmo de agrupamiento basado en árboles de decisión no supervisados [20].

3 Trabajo Relacionado

En esta sección se muestra un análisis crítico de los trabajos relacionados. La Figura 1 muestra una taxonomía de los algoritmos propuestos en la literatura para agrupamiento basado en patrones. Esta taxonomía se divide en tres ramas cada una representando la estrategia que se sigue para la construcción de los agrupamientos, a saber: a) Construir los patrones a la vez que se agrupan los objetos, b) Agrupar primero los objetos y luego encontrar patrones y c) Extraer todos los patrones frecuentes y a partir de ellos agrupar los objetos. Nuestra investigación tiene como objetivo mejorar los métodos y algoritmos de la tercera estrategia, porque con ella se han obtenido buenos resultados [7] en comparación con las otras dos estrategias. Por esta razón es que a continuación se describen los algoritmos de agrupamiento basado en patrones más relevantes que siguen la tercera estrategia.

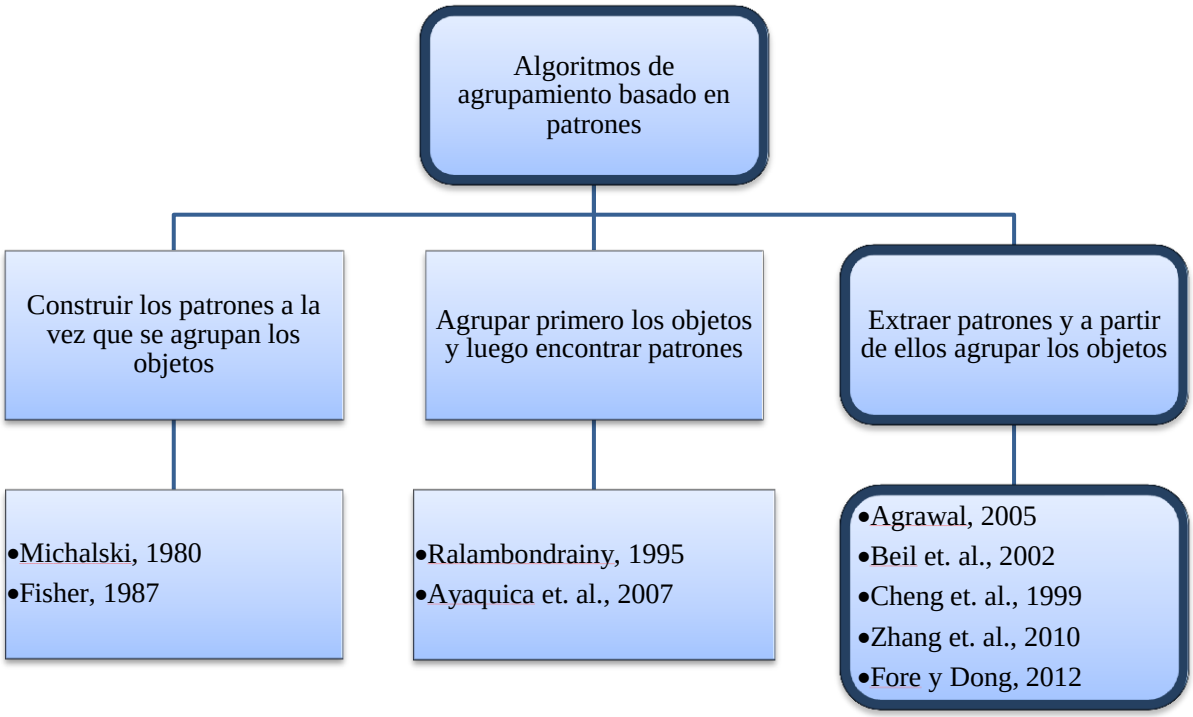

Figura 1. Taxonomía de los algoritmos de agrupamiento basado en patrones.

Los algoritmos de la tercera estrategia se basan en extraer patrones frecuentes en datos categóricos (objetos descritos por atributos no numéricos) para luego agrupar los objetos basándose en los patrones extraídos; es por eso que si existen atributos numéricos se aplica algún proceso de discretización previa.

Agrawal et al. [13] propusieron un algoritmo de agrupamiento (CLIQUE) a partir de los patrones extraídos con *Apriori*, para grandes bases de datos. Los patrones frecuentes extraídos con Apriori determinan los posibles grupos iniciales posteriormente el algoritmo CLIQUE integra métodos basados en densidad y en grafos para construir los agrupamientos.

Cheng et al. [14] proponen ENCLUS, un algoritmo desarrollado a partir de CLIQUE, pero basado en la entropía y que puede agrupar datos numéricos. Sin embargo, los datos numéricos son discretizados apriori, para formar grupos de atributos que determinarán los grupos de objetos. A partir de los grupos formados se crean reglas de asociación.

En la literatura se han propuesto varios algoritmos para agrupar textos basados en secuencias (patrones) frecuentes. Beil et al. [10] propusieron el algoritmo FTC que extrae secuencias frecuentes con *Apriori* y luego agrupa las secuencias con menos solapamiento y que cubran a la Base de datos de textos.

Zhang et al. [16] representan y determinan las similaridades entre los documentos a partir de las co-ocurrencias de secuencias frecuentes. Basados en las secuencias frecuentes, el algoritmo MC agrupa los documentos en una primera etapa. En una segunda etapa, las secuencias con mayor frecuencia en un grupo de documentos son seleccionadas como representante o tópico del grupo.

3.1 Algoritmo CPC

Fore y Dong [7, 25] proponen el algoritmo CPC que agrupa patrones frecuentes diversos por cada grupo, maximizando un criterio de calidad. CPC se utilizó para contrastar los resultados preliminares presentados en esta propuesta, porque ha reportado los mejores resultados. Uno de los parámetros de este algoritmo es el conjunto de patrones previamente extraídos; en [7] proponen extraer patrones con el algoritmo *FP-Growth* [15]. Los objetos no se agrupan hasta la fase final, inicialmente se obtiene un agrupamiento de los patrones extraídos y posteriormente los objetos se incluyen en el grupo donde existen más patrones que los cubra. Una desventaja de este algoritmo es que solamente se puede aplicar en datos categóricos, de modo que los atributos numéricos tienen que ser previamente discretizados.

Un criterio de calidad utilizado para determinar qué par de patrones deben estar en el mismo grupo es *MPQ* [25]. Este criterio se define de la siguiente manera, dados dos patrones *P1* y *P2* que comparten pocos objetos, *MPQ(P1,P2)* asigna una calidad alta si hay un número relativamente alto de patrones (mutuos) que comparten objetos con *P1* y *P2*. Dado un número de grupos *K*, el algoritmo encuentra *K* patrones semilla para inicializar los grupos; estos patrones semilla tienen valores bajos de calidad entre ellos según *MPQ*. Seguidamente se van adicionando a cada grupo los patrones que tienen valores altos de MPQ con los patrones del grupo. El resto de los patrones se adicionan a los grupos teniendo en cuenta los objetos en común.

Una vez que se agrupan los patrones se procede a agrupar a los objetos. Los objetos son asignados a los grupos según los pesos de cada patrón que caracteriza al objeto. Estos pesos favorecen a los patrones con muchas propiedades y con un elevado número de objetos en común en su grupo en relación con otros grupos. De esta manera, cada grupo emite un *SCORE* para cada objeto sumando los pesos de sus patrones. Los objetos finalmente se asignan al grupo de mayor *SCORE* [7].

Un algoritmo general para agrupar con CPC aparece en Algoritmo 1.

Algoritmo 1: Seudocódigo del algoritmo CPC de Fore y Dong en 2010 [7, 25]

```
Entrada: K – Número de grupos,
                P – Conjunto de patrones,
                O – Conjunto de objetos
Salida: GO – Grupos de objetos,
            GP – Grupos de patrones

// Encontrar los patrones semillas. Cada semilla es un conjunto de patrones
// {P1,…,Pk}
Semillas_Iniciales ← Generar_ Semillas_Aleatorias(P, media_soporte)
Semilla ← Semillas_Iniciales.Mejor_Semilla(); // Semilla con los patrones que
// minimizan la medida MPQ(Pi, Pj)
repetir
    desde (i=1 hasta K) hacer
        patron ← Min(MaxMPQ(P, Semilla[j]));  // Encontrar en P un patrón que
// minimize el máximo MPQ con los patrones en Semilla, para j distinto de i y
// j = 1,…,K
        si (MaxMPQ(patron, Semilla[j]) < MaxMPQ(Semilla[i], Semilla[j])) entonces
            Reemplazar(Semilla[i], patron);  // Mejorar la semilla.
        fin
    fin
hasta que No_se_puedan_hacer_mas_reemplazos

// Agrupar los patrones
desde (i=1 hasta K) hacer GP[i] ← Semilla[i]; fin  // Inicializar los grupos de
// patrones con la semilla
```

```
mientras (Hayan_patrones_sin_usar) hacer
    mejor_patron ← ArgmaxMPQp(P, GP[i]);  // i = 1,…,K
    mejor_grupo ← ArgmaxMPQc(mejor_patron, GP);  // i = 1,…,K
    si (MPQ(mejor_patron, mejor_grupo) > 0) entonces
            mejor_grupo.Adicionar(mejor_patron);  // Cada patrón se adiciona en
// el mejor grupo según MPQ
    fin sino break;
fin
Completar_grupos_de_patrones(P, GP);  // Adicionar los patrones que quedaron

// Agrupar los objetos
porcada (objeto en O) hacer
    indice_grupo ← Pos_MaxSCORE(objeto, GP);  // Índice del grupo de mayor SCORE
// para el objeto
    GO[indice_grupo].Adicionar(objeto);  // Cada objeto se adiciona al grupo de
// mayor SCORE
Fin

return GO, GP
```

En la Tabla 2 se muestra un resumen del análisis de los algoritmos del estado del arte, indicando sus características principales. También se incluye en la comparativa las características del algoritmo que se propone desarrollar en esta investigación.

Tabla 2. Comparativa entre los trabajos previos

Algoritmos	Extracción de patrones	Calcula todos los patrones	Filtrado de patrones	Discretización previa
CLIQUE	*Apriori*	Sí	No	Sí
ENCLUS	*Apriori* (modificado)	Sí	No	Sí
FTC	*Apriori*	Sí	No	Sí
MC	*FP-Growt* y BIDE	Sí	Maximales	Sí
CPC	*FP-Growth*	Sí	Minimales (por Clase de Equivalencia)	Sí
Propuesta	Árboles de Decisión no Supervisados	No	Adecuados para agrupar	No

Los trabajos antes descritos tienen dos características principales, una es que es necesario hacer una discretización previa para datos numéricos y la otra es que calculan todos los patrones. Discretizar previamente los datos numéricos es una limitante porque puede generar pérdida de información. Calcular todos los patrones tiene la desventaja de producir información redundante que afecta la eficacia del agrupamiento; además, el costo computacional de calcular todos los patrones es muy alto y se vuelve inaplicable para bases de datos grandes. Es por eso que el método que se propone desarrollar en esta investigación doctoral no discretiza previamente los datos numéricos y calcula solo un subconjunto de buenos patrones.

3.2 Motivación

Como se puede apreciar, se han desarrollado varios trabajos para el agrupamiento basado en patrones, con el propósito de poder interpretar los resultados obtenidos. Sin embargo, los algoritmos existentes presentan dos limitaciones importantes:

- Para datos numéricos es necesario realizar una discretización previa, lo cual genera pérdida de información.

- Los algoritmos tradicionales usados para extraer patrones frecuentes extraen todos los patrones, aun cuando muchos de ellos brinden información redundante al momento de agrupar. Extraer todos los patrones es costoso, lo cual lo hace inaplicable a bases de datos grandes.

El proceso de discretización previa de los datos no solo genera pérdida de información, si no que al discretizar los atributos individualmente se deteriora la calidad de los patrones, ya que los patrones son combinaciones de atributos y valores que deben aparecer simultáneamente. Además, discretizar usualmente define fronteras entre los valores, por lo que es posible que objetos muy parecidos, pero con valores en lados distintos de las fronteras, no sean caracterizados por los mismos patrones [33].

Es por eso que en el marco de esta investigación doctoral consideramos importante desarrollar un algoritmo de agrupamiento basado en patrones que disminuya estas limitaciones. Para ello, es necesario que el algoritmo sea capaz de agrupar datos numéricos sin requerir un proceso previo de discretización para no transformar los datos. Los patrones extraídos deben cumplir un determinado criterio que indique cuáles patrones son útiles para agrupar, evitando que se extraigan todos. Al no extraer todos los patrones, el algoritmo propuesto obtendrá un conjunto reducido de patrones, logrando así mayor eficiencia y mejorando la eficacia, ya que solo se tendrán en cuenta patrones adecuados para agrupar. Para ello nos basaremos en los buenos resultados obtenidos en clasificación supervisada utilizando árboles de decisión para extraer patrones [33], pero este tipo de árboles se adaptarán al problema no supervisado.

4 Propuesta

4.1 Preguntas de investigación

¿Cómo extraer un subconjunto reducido de patrones frecuentes, en bases de datos numéricas o mezcladas e incompletas sin hacer transformaciones en los datos, que sean buenos para agrupar?

¿Cómo construir eficientemente agrupamientos basados en patrones utilizando un subconjunto de los patrones frecuentes?

¿Cómo decidir si un patrón es mejor que otro para describir un agrupamiento?

4.2 Objetivo general

Desarrollar un algoritmo de agrupamiento basado en patrones, extrayendo solo un subconjunto reducido de patrones adecuados para agrupar, que permita trabajar con datos mezclados e incompletos sin transformar los datos y que sea más eficiente y eficaz que los algoritmos existentes.

4.3 Objetivos particulares

- Desarrollar un algoritmo para extraer subconjuntos de patrones frecuentes sin transformar los datos en problemas de agrupamiento.

- Desarrollar un algoritmo de filtrado de los patrones extraídos para seleccionar solo los adecuados para agrupar.

- Diseñar un algoritmo eficiente y eficaz para construir agrupamientos a partir de los patrones extraídos con los algoritmos anteriores.

4.4 Contribuciones

Las principales contribuciones esperadas al término de esta investigación doctoral son las siguientes:

- Un algoritmo de extracción de patrones para datos mezclados sin discretizar previamente los datos numéricos, basado en un bosque de árboles de decisión no supervisados.
- Un algoritmo de filtrado de patrones para seleccionar solo los patrones frecuentes adecuados para agrupar.
- Un algoritmo de agrupamiento a partir de los patrones extraídos, para formar grupos de objetos y patrones de manera más eficiente y con mayor calidad que los algoritmos del estado del arte.

4.5 Metodología

1. Desarrollar un algoritmo de extracción de subconjuntos de patrones frecuentes sin transformar los datos:

a) Desarrollar un nuevo algoritmo de extracción de patrones para datos mezclados sin transformar los datos, a partir de árboles de decisión no supervisados.

a.1) Estudio crítico de los algoritmos de construcción de árboles de decisión no supervisados [17-20].

a.2) Diseñar un algoritmo de construcción de árboles de decisión no supervisados para datos numéricos, sin discretizar. Inicialmente se construirán árboles binarios basados en C4.5 [31], posteriormente se analizarán otros tipos de árboles.

a.3) Analizar como seleccionar la propiedad más adecuada para decidir las divisiones. La función de evaluación de divisiones candidatas representa lo que se entiende por "buen agrupamiento". Se analizarán diferentes criterios de agrupamiento.

a.4) Agregar atributos categóricos al algoritmo de construcción de árboles de decisión no supervisados. Analizar cómo se pueden medir y comparar las divisiones candidatas con tipos de datos diferentes.

a.5) Estudio crítico de las condiciones de paradas o algoritmos de poda existentes.

a.6) Proponer una condición de parada o algoritmo de poda que tenga en cuenta el soporte mínimo deseado al extraer los patrones. Evaluar la conveniencia de restringir la cantidad de niveles que tendrán los árboles.

b) Proponer un algoritmo de construcción de bosque de árboles de decisión no supervisados, para garantizar diversidad en los patrones extraídos.

b.1) Analizar si se extraen patrones adecuados escogiendo divisiones que no sean las mejores.

c) Proponer un algoritmo para extraer los patrones presentes en el bosque de árboles de decisión no supervisados y simplificar las propiedades redundantes.

c.1) Evaluar la manera de incluir peso a los patrones en dependencia de los nodos que éstos representen en el árbol.

2. Desarrollar un algoritmo de filtrado de los patrones extraídos para seleccionar solo los adecuados para agrupar.

d) Estudio crítico de los métodos de filtrado de patrones frecuentes.

e) Proponer un algoritmo de filtrado de patrones que garantice obtener buenos resultados.

e.1) Explorar estrategias como filtrar patrones por clases de equivalencia.

e.2) Explorar estrategias basadas en pesos asignados a los patrones.

3. Diseñar un algoritmo eficiente y eficaz para construir agrupamientos a partir de los patrones extraídos con los algoritmos anteriores:

f) Estudio crítico de los algoritmos de agrupamiento basado en patrones [7, 10, 13, 14, 16], Se partirá con algoritmo CPC [7] que ha reportado los mejores resultados.

g) Proponer un algoritmo de agrupamiento basado en patrones sin las limitaciones de CPC. Los agrupamientos resultantes serán una partición del conjunto de objetos. Al inicio se tendrán en cuenta solo particiones duras, luego se analizarán otras particiones como las difusas.

h) Evaluar la calidad e interpretabilidad de los resultados obtenidos. Las experimentaciones se realizarán utilizando bases de datos del Repositorio UCI [32], que es muy utilizado en la literatura. Se analizará la viabilidad de utilizar otras bases de datos de naturaleza diferente.

h.1) Estudio crítico de las medidas de calidad de los agrupamientos [29, 40]. Analizar los criterios internos y externos de validación. Proponer una medida o seleccionar una existente.

h.2) Estudio crítico de las medidas de calidad de los patrones agrupados. Proponer una medida o seleccionar una existente.

i) Comparación experimental del algoritmo propuesto para agrupamiento basado en patrones con otros algoritmos del estado del arte, en las bases de datos seleccionadas. Analizar los resultados según las medidas de calidad utilizadas.

4.6 Cronograma

Tareas	Semestres						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1. Análisis de la literatura							
2. Algoritmo de extracción de patrones a partir de árboles de decisión no supervisados							
3. Agregar atributos categóricos al algoritmo de construcción de árboles de decisión no supervisados							
4. Algoritmo de construcción de bosque de árboles de decisión no supervisados							
5. Extraer los patrones presentes en el bosque de árboles de decisión no supervisados							
6. Algoritmo de filtrado de patrones para seleccionar solo los adecuados para agrupar							
7. Estudio crítico de los algoritmos de agrupamiento de patrones							
8. Algoritmo de agrupamiento basado en los patrones extraídos							
9. Evaluar la calidad e interpretabilidad de los resultados obtenidos							
10. Comparación experimental							
11. Escritura de artículos							
12. Redacción de la propuesta							
13. Redacción del documento de tesis							
14. Entrega del documento de tesis a los asesores							
15. Entrega del documento de tesis al comité y Defensa de tesis							

5 Resultados preliminares

Como resultado preliminar se propone un nuevo algoritmo de extracción de patrones para datos numéricos, el cual permite extraer un subconjunto reducido de patrones frecuentes. Para agrupar usando los patrones extraídos se utiliza el algoritmo CPC [7]. La comparación experimental muestra que, utilizando el algoritmo de extracción de patrones propuesto, el algoritmo CPC obtiene mejores resultados que extrayendo todos los patrones, que es la estrategia propuesta por los autores de CPC.

El algoritmo de extracción de patrones está basado en árboles de decisión no supervisados [20], ya que con estos árboles podemos expresar propiedades sin discretizar previamente los valores numéricos. Además, los árboles de decisión no supervisados son rápidos de construir y no extraen todas las propiedades posibles, sino solo las que cumplan un criterio determinado. En el algoritmo propuesto como resultado preliminar los árboles que se construyen son binarios.

Para extraer suficientes patrones se requieren muchos árboles diferentes, por lo cual se propuso un algoritmo de inducción de árboles de decisión que permita crear árboles diversos al ser ejecutado varias veces. De cada árbol se extraen los patrones recorriendo los caminos desde la raíz a cualquier nodo del árbol. El resultado final es la unión de todos los patrones, después de simplificarlos y eliminar duplicados.

El algoritmo de inducción de cada árbol de decisión está basado en el algoritmo C4.5 [31], pero con tres diferencias fundamentales. La primera es que no utiliza la información de las clases en ningún momento, ya que en el proceso de agrupamiento los objetos no están etiquetados; la segunda diferencia está en la manera en que se eligen las "divisiones candidatas"; y la tercera en el "criterio de terminación".

El proceso de construcción del árbol es de arriba hacia abajo (*top-down*) y se propone utilizar como parámetro de entrada el min_*soporte* permitido (en porciento), usado para garantizar que los patrones que se extraigan sean frecuentes respecto al umbral. La construcción del árbol se inicia con el nodo raíz N que contiene todos los objetos de la Base de datos. En la primera iteración se dividen los datos en dos subconjuntos, creándose dos nodos hijos. Los nuevos hijos creados son divididos recursivamente con el mismo algoritmo hasta que se cumpla el criterio de terminación, finalizando así el proceso.

Para crear los nodos hijos es necesario evaluar las posibles divisiones candidatas de los objetos del nodo padre para determinar en cuáles subconjuntos se dividirá. El objetivo es que los nodos hijos representen un "buen agrupamiento" del nodo padre. El criterio de "buen agrupamiento" seguido en esta investigación doctoral es que los objetos de un grupo deben parecerse más entre sí que a objetos de otros grupos. Basado en ese criterio la mejor división candidata de un atributo es la que tiene mayor separación entre los valores de los nodos hijos y, a su vez, poca separación en los valores de dichos nodos. Para eso se realiza el siguiente procedimiento:

- Por cada atributo numérico x_j
    1. Se ordenan ascendentemente los valores v_{ji} diferentes y se almacenan en una lista O_j.
    2. Se crea una nueva lista D_j con las diferencias entre cada valor consecutivo de O_j, de manera que cada valor $d_{ji} \in D_j$ se define como $d_{ji} = v_{ji+1} - v_{ji}$.
    3. Se determina la posición k del $\max\{d_{ji}\}$ y se construye la propiedad $p = (x_j \leq v_{jk})$.

4. A partir de la propiedad p se construyen dos nuevos nodos A y B, de manera que en A estarán los objetos de N que cumplen la propiedad p y en B los que no la cumplen.
5. A esta división candidata se le asigna un valor de calidad e_j calculado como:

$$e_j = \frac{\max\{d_{ji}\} - \min\{\overline{d_{ji}^A}, \overline{d_{ji}^B}\}}{\max\{v_{ji}\} - \min\{v_{ji}\}} \tag{2}$$

donde $\overline{d_{ji}^A}$ es el promedio de las diferencias de los valores del atributo x_j en el nuevo nodo A.

- El mayor valor $e_j, j = 1,...,m$ determina qué atributo j y qué división candidata particionará al nodo N.
- Este proceso se repite recursivamente para cada nuevo nodo mientras no se cumpla la condición de terminación, la cual explicaremos más adelante.

Un punto importante en el proceso de construcción del árbol es cómo evaluar una división candidata. En la sección 2.4 se menciona que hay dos propiedades deseables para que un agrupamiento sea considerado bueno, estas propiedades son la "compacidad" y la "separación"; precisamente basado en esas propiedades se definió la medida de calidad de una división candidata. La medida de calidad propuesta e_j no solo toma en cuenta la mayor separación entre los valores de un atributo, a esa mayor separación se le resta el menor promedio de las diferencias de ese atributo en los dos nuevos nodos creados. Decidimos restar por el menor promedio para garantizar al menos un nodo de buena calidad, así el otro nodo se irá particionando en mejores nodos a medida que avance el proceso. Finalmente, se divide por la mayor diferencia en el nodo padre, con el propósito de normalizar el resultado.

La condición de terminación que proponemos es la siguiente:

- Al construir los nuevos nodos A y B, si $(|A| < \min\_soporte) \vee (|B| < \min\_soporte)$, se desecha la división candidata y se exploran los siguientes mayores valores d_{ji}. Si ninguna división candidata cumple la condición, el nodo ya no será dividido.

La Figura 2 muestra los objetos de una Base de datos sintética de 150 objetos. Visualmente se puede apreciar que los objetos están distribuidos en dos grupos. Nos apoyaremos en esta Base de datos como ejemplo para explicar el algoritmo de extracción de patrones.

Supongamos que $\min\_soporte = 0.15$, es decir, solo se extraerán patrones que "caractericen" a más del 15% del total de la Base de datos, esto es 23 objetos. La división inicial se realiza por el atributo *X*, ya que la propiedad $(X \leq 56)$ es la que obtiene el valor de e_j más alto, como indica la línea vertical en la Figura 2.

El proceso continúa mientras no se cumpla la condición de terminación, la línea horizontal en la Figura 2 indica la segunda división $(Y \leq 25)$.

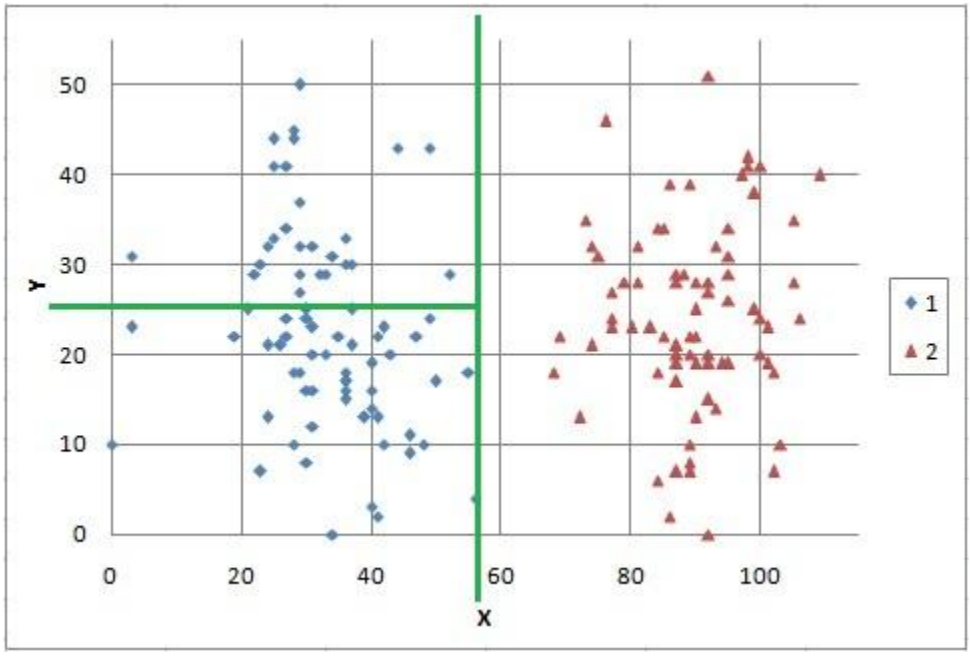

Figura 2. Base de datos sintética "Simple". Las dos líneas (horizontal y vertical) señalan las dos primeras divisiones en el primer árbol de decisión no supervisado construido.

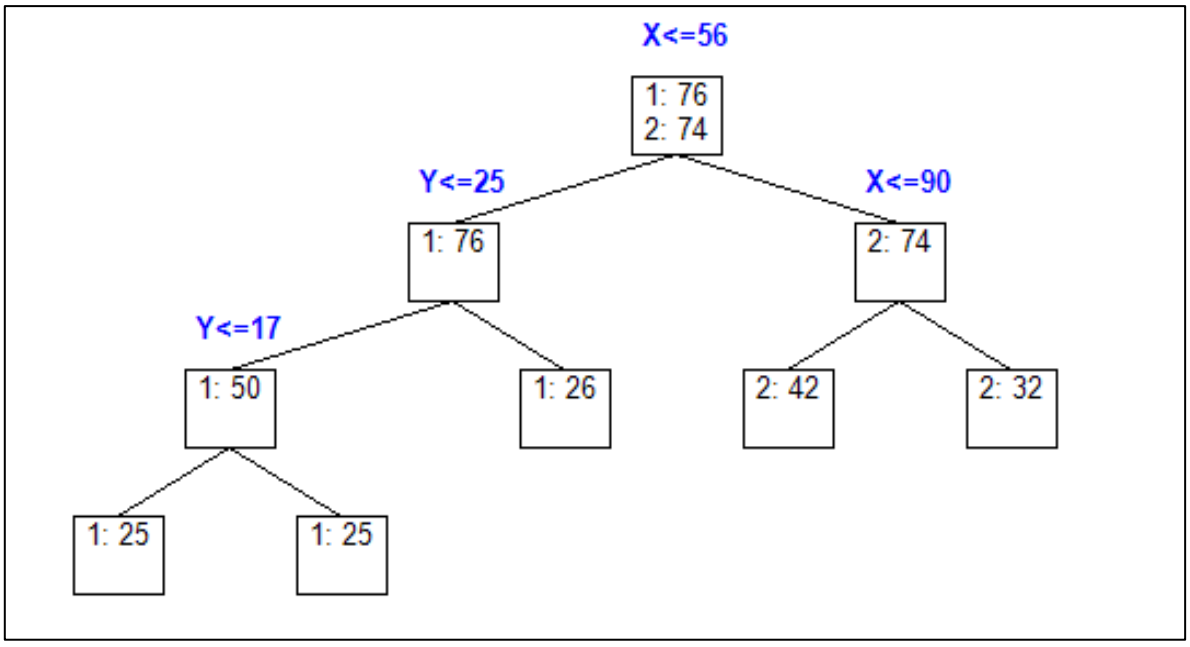

Figura 3. Árbol de decisión no supervisado construido a partir de la base de datos "Simple", con soporte 0.15.

En la Figura 3 se muestra el resultado de construir el árbol completo. Dentro de los rectángulos, que representan a los nodos, se indica cuántos objetos de cada tipo hay en el nodo, por ejemplo en el

nodo raíz hay 76 objetos del tipo 1 y 74 del tipo 2; esta información no se utiliza en la construcción del árbol, solo la ponemos para mostrar cómo queda la distribución de los objetos en el árbol.

Con el objetivo de obtener más patrones, se construye un bosque con árboles diferentes, los cuales permiten obtener patrones diversos. En 2010 García et al. [33] propusieron una estrategia de construcción de un bosque de árboles de decisión para extraer patrones para clasificar. Como resultado preliminar, basándose en esa estrategia, el algoritmo propuesto construye el bosque de árboles de decisión no supervisados. Para ello se selecciona no siempre la mejor división candidata, sino que se incluye la segunda y tercera mejor división para los tres primeros niveles. De esta forma, en total se construyen $3^3 = 27$ árboles diferentes. El algoritmo general para construir un árbol aparece en Algoritmo 2. Los 27 árboles se construyen siguiendo este algoritmo, pero con la estrategia de construcción de un bosque anteriormente descrita.

Algoritmo 2: Seudocódigo del algoritmo recursivo *Construir_Arbol*

```
Entrada: Nodo – Raíz del árbol con todos los objetos en el primer paso
Salida: arbol – Apuntador a la raíz del árbol creado

porcada (atributo_Xj en Nodo.Objetos) hacer
    atributo_Xj.Ordenar;
    dif ← Mayor_Diferencia(atributo_Xj);  // Mayor diferencia válida entre los
// valores consecutivos del atributo_Xj. (Para que la diferencia sea válida debe
// dividir los objetos en dos conjuntos con al menos el mínimo soporte)
    si (No_hay_diferencia_valida) entonces  // Se evalúa el criterio de parada
        retornar Nodo;
    fin
    propiedad ← "atributo_Xj <= dif";  // Construir una propiedad a partir de la
// mayor diferencia
    Division.Crear(propiedad, T);  // Crear una división candidata según la
// propiedad
    Divisiones.Adicionar(Division);
fin

division ← Divisiones.Mejor_Division();  // División que maximiza el criterio de
// calidad
Nodo.NI ← Crear_Nodo_Izquierdo(division, T);  // Crear el nodo "hijo" izquierdo
// usando la mejor división
Nodo.ND ← Crear_Nodo_Derecho(division, T);  // Crear el nodo "hijo" derecho
// usando la mejor división

Construir_Arbol(NI);  // Construir el subárbol izquierdo recursivamente
Construir_Arbol(ND);  // Construir el subárbol derecho recursivamente

return Nodo;
```

Una vez construido el bosque de árboles de decisión no supervisados, se extraen todos los caminos de la raíz a un nodo en cada árbol y esos caminos constituyen los patrones, una vez que se simplifiquen y se eliminen repetidos. En la Figura 4, las flechas muestran los caminos que se toman en cuenta desde la raíz a un nodo en el árbol de la Figura 3. Note que cada flecha constituye un camino y por lo tanto genera un patrón.

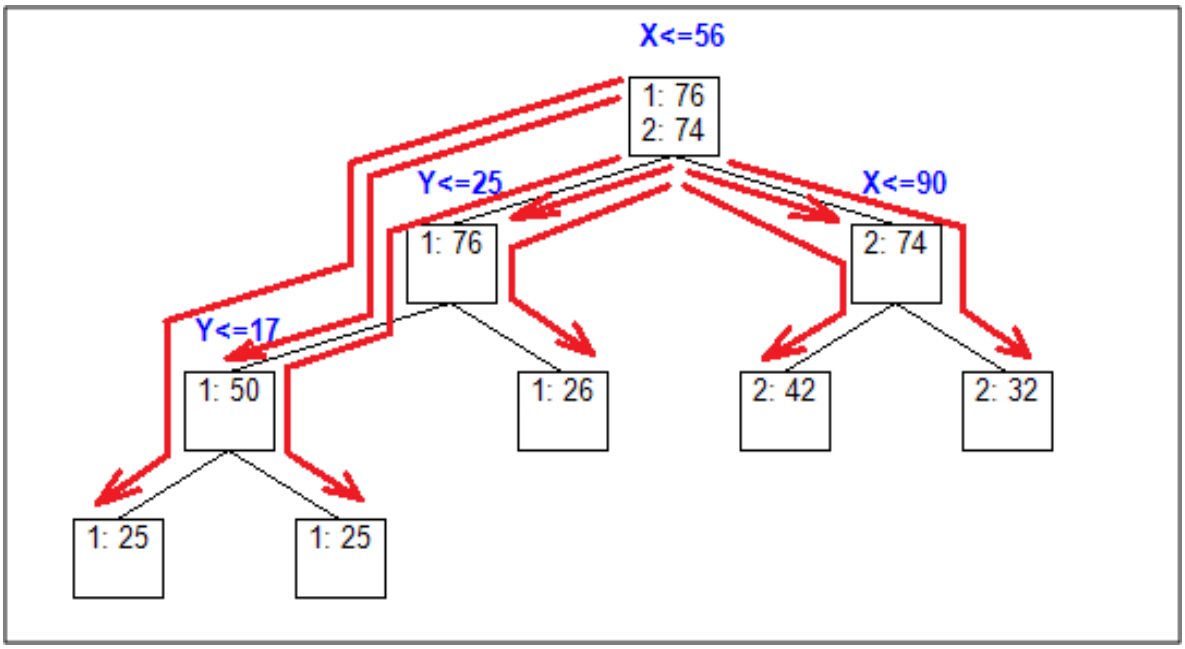

Figura 4. Las flechas muestran los caminos que se toman en cuenta desde la raíz a un nodo para extraer los patrones.

Dentro de un patrón, dos propiedades *A* y *B* del mismo atributo son redundantes si todos los objetos del universo que cumplen *A* también cumplen *B*, pero no viceversa. Si hay propiedades redundantes dentro de un patrón, se elimina la propiedad más general. Por ejemplo, uno de los patrones extraídos del árbol de la Figura 3 es $[(X \leq 56) \wedge (Y \leq 25) \wedge (Y \leq 17)]$; como todos los objetos que cumplen $(Y \leq 17)$ también cumplen $(Y \leq 25)$, y no a la inversa, entonces la propiedad más general es $(Y \leq 25)$ y se elimina del patrón. El patrón resultante es $[(X \leq 56) \wedge (Y \leq 17)]$, que cubre los mismos objetos.

La lista de patrones extraídos de este árbol, después de simplificarlos y eliminar los repetidos, es la siguiente:

- $[(X \leq 56)]$
- $[(X \leq 56) \wedge (Y \leq 25)]$
- $[(X \leq 56) \wedge (Y \leq 17)]$
- $[(X \leq 56) \wedge (Y > 17) \wedge (Y \leq 25)]$
- $[(X \leq 56) \wedge (Y > 25)]$
- $[(X > 56)]$
- $[(X > 56) \wedge (X \leq 90)]$
- $[(X > 90)]$.

Como se aprecia, el algoritmo propuesto extrae patrones para datos numéricos sin tener que discretizar estos datos apriori. Además, no explora todo el espacio de búsqueda, ya que solo construye patrones a partir de divisiones que produzcan buenos árboles y diversos. El proceso de construcción del árbol incluye una medida para evaluar las divisiones candidatas, la cual está enfocada en que los patrones construidos a partir de las divisiones sean buenos para agrupar.

Otro aporte de los resultados preliminares es una nueva medida para evaluar la calidad de los patrones utilizados para describir los agrupamientos. Para eso proponemos evaluar la importancia de cada patrón P_{ij} en el grupo G_i con el que está asociado. La importancia de un patrón en un grupo se mide como la razón entre el soporte del patrón en el grupo $Sg(P_{ij})$ y el soporte total del patrón $S(P_{ij})$. La calidad de cada grupo de patrones se mide como el promedio de la importancia de sus patrones. La calidad de todo el agrupamiento se determina como el promedio de la calidad de cada grupo. Así, la medida de calidad propuesta se define como:

$$Calidad\_Patrones = \frac{\sum_{i=1}^{k}\sum_{j=1}^{l}\frac{Sg(P_{ij})}{S(P_{ij})}}{k \cdot l} \quad (3)$$

donde $k = |G_i|, l = |P_i|$, siendo P_i el conjunto de patrones asociados al grupo G_i. Los valores de la medida están acotados en el intervalo [0,1]; los valores cercanos a 1 indican que el agrupamiento de los patrones es correcto.

5.1 Comparación experimental

El objetivo principal de la comparación experimental es mostrar que los patrones que se obtienen permiten obtener mejores agrupamientos. Una vez que se cuenta con la lista de los patrones extraídos, el proceso que sigue es el de agrupar los objetos basándose en esos patrones. Para ello utilizamos el algoritmo CPC [7], que construye grupos de patrones contrastantes, maximizando un criterio de calidad. Después, basado en esos grupos de patrones, se crean los grupos de objetos.

Para mostrar la mejora en eficacia del algoritmo propuesto para obtener patrones, comparamos los resultados de agrupar utilizando el algoritmo CPC con los nuevos patrones y con los patrones extraídos según la estrategia de CPC. Las comparaciones se realizan en 10 bases de datos, 9 del Repositorio UCI [32] y una sintética.

Todos los valores de atributos en las bases de datos son numéricos, por lo que hay que discretizarlos para extraer patrones con la estrategia utilizada por CPC. El método de discretización utilizado es el método de discretización no supervisado que aparece en Weka 3.6.1 [40]. Con los valores por defecto, este método crea 10 intervalos del mismo tamaño (no con la misma cantidad de objetos) para cada atributo.

En la Tabla 3 se muestra un resumen de las características de cada base de datos, incluyendo el umbral de soporte que se utilizó para extraer los patrones. Ese valor de soporte depende de cada base de datos y se determinó experimentalmente, como se sugiere en [7].

En la Tabla 4 se muestran los valores de calidad de los agrupamientos obtenidos usando el algoritmo K-Means como un valor de referencia. Sin embargo, el objetivo es comparar los resultados del algoritmo propuesto y CPC.

La eficacia de los agrupamientos se midió con el índice externo de validación Rand Statistic [30], ya que Brun et al. en 2007 [38] concluyen que las medidas externas son las adecuadas cuando se conoce información sobre la clase de los objetos. En ese trabajo, Rand Statistic fue la medida externa que mejores resultados obtuvo al predecir el error de agrupamiento.

Tabla 3. Descripción de las bases de datos.

Bases de datos	Objetos	Atributos	Clases	Soporte
Cleveland	303	13	5	0.15
Cloud	108	4	4	0.1
Ecoli	336	7	8	0.05
Glass	214	9	6	0.03
Iris	150	4	3	0.15
Simple	150	2	2	0.15
Vertebral2C	310	6	2	0.15
Vertebral3C	310	6	3	0.15
Vowel	990	10	11	0.1
Wine	178	13	3	0.15

Por otra parte, la calidad de los grupos de patrones se evaluó con la medida propuesta. En negritas se señalan los mejores resultados entre CPC y el algoritmo propuesto, respecto a la calidad de los agrupamientos y respecto a la medida de calidad de los grupos de patrones.

Tabla 4. Evaluación de los resultados de los agrupamientos.

Bases de datos	Rand Statistic			Calidad de los patrones	
	K-Means	CPC	Propuesta	CPC	Propuesta
Cleveland	0.6057	0.6111	**0.6207**	0.3365	**0.4668**
Cloud	0.6225	0.5457	**0.5685**	**0.6215**	0.4342
Ecoli	0.806	0.6941	**0.7970**	0.3559	**0.4204**
Glass	0.6733	**0.6129**	0.5751	0.3885	**0.5388**
Iris	0.8797	0.6938	**0.9412**	0.7242	**0.8957**
Simple	1	0.5547	**1**	0.9166	**0.9892**
Vertebral2C	0.557	0.5548	**0.5921**	0.762	**0.7834**
Vertebral3C	0.6751	0.6399	**0.6752**	**0.5983**	0.2910
Vowel	0.8653	**0.8262**	0.7380	**0.3309**	0.2098
Wine	0.7187	0.6301	**0.6689**	0.5291	**0.6312**
Total	-	2	**8**	3	**7**

* En negritas se señalan los mejores resultados entre el algoritmo propuesto y el algoritmo CPC. El algoritmo K-Means no participa en la comparación.

Como se aprecia en la Tabla 4, el método propuesto es más eficaz en 8 de las 10 bases de datos utilizadas. En cuanto a la calidad de los patrones el método propuesto es mejor en 7 de las 10 bases de datos. Estos resultados muestran que el método propuesto extrae mejores patrones para agrupar, sin discretizar los datos numéricos, lo cual permite obtener agrupamientos de mayor calidad.

6 Conclusiones

Los algoritmos de agrupamiento basado en patrones han reportado buenos resultados y son atractivos porque permiten explicar los agrupamientos obtenidos. Esto se debe a que cada grupo de objetos queda asociado con un grupo de patrones que describe las propiedades que cumplen los objetos del grupo.

Los algoritmos para extraer patrones usualmente calculan todos los patrones frecuentes y después se realiza un costoso proceso de filtrado. Además, la mayoría de los algoritmos para buscar patrones solo están definidos para datos categóricos, por lo que hay que discretizar previamente los datos numéricos.

En la presente investigación doctoral se propone desarrollar un algoritmo de extracción de un subconjunto reducido de patrones adecuados para agrupar, que sea capaz de trabajar con datos mezclados e incompletos, sin discretizar previamente los datos numéricos. Además, basado en los patrones extraídos, se pretende desarrollar un algoritmo de agrupamiento eficiente y eficaz.

Como resultados preliminares de la investigación se desarrolló un nuevo algoritmo para extraer subconjuntos reducidos de patrones sin hacer un proceso previo de discretización de los datos numéricos, a partir de un bosque de árboles de decisión no supervisados. Además, se propone una medida para evaluar la calidad de los grupos de patrones obtenidos. Los experimentos mostraron que los patones obtenidos con el algoritmo propuesto permiten obtener mejores agrupamientos que los que se pueden obtener calculando todos los patrones.

Finalmente, basados en los resultados preliminares, concluimos que los objetivos planteados en la presente investigación doctoral se pueden alcanzar, siguiendo la metodología planteada, en el tiempo previsto.

Referencias

1. Murtagh F. A survey of recent hierarchical clustering algorithms. In *The Computer Journal*, 1983.
2. Fisher D.H. Knowledge acquisition via incremental conceptual clustering. In *Machine Learning*, 1987.
3. K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
4. Iwayama, M. and Tokunaga, T. Cluster-based text categorization: A comparison of category search strategies. In: *Proc. 18th ACM Internat. Conf. on Research and Development in Information Retrieval*, pp. 273–281, 1995.
5. Shi, Jianbo, Malik and Jitendra. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Machine Intell*, 22, 888–905, 2000.
6. R. S. Michalski and R. E. Stepp. Learning from observation: conceptual clustering. In *Machine Learning: An Artificial Intelligence Approach*, pages 331–363, 1983.
7. Fore N. and Dong G. *A Contrast Pattern Based Clustering Algorithm for Categorical Data*. Thesis for Master of Science, Wright State University, 2010.
8. Michalski R.S., et. al. Natural Induction and Conceptual Clustering: A Review of Applications. *Reports of Machine Learning and Inference Laboratory*. George Mason University, 2006.
9. O. M. Harari, I. Jorge, S. Zwir. *Predicting prokaryotic and eukaryotic gene networks by fusing domain knowledge with conceptual clustering algorithms.* Tesis Univ. Granada. Departamento de Ciencias de la Computación e Inteligencia Artificial, 2009.
10. Beil F., Ester M. and Xiu X.. Frequent Term-Based Text Clustering. *SIGKDD 02* Edmonton, Alberta, Canada, 2002.

11. Michalski R.S. Knowledge acquisition through conceptual clustering: a theoretical framework and an algorithm for partitioning data into conjunctive concepts. *Policy Analysis and Information Systems*, v. 4, n. 3, pp. 219-244, 1980.
12. Agrawal R., Imielinski T. and Swami. Mining association rules between sets of items in large databases. In: *Proceedings of the 1993ACM-SIGMOD international conference on management of data (SIGMOD'93)*, Washington, DC, pp 207–216, 1993.
13. Agrawal R, Gehrke J, Gunopulos D and Raghavan P. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11, 5–33, 2005.
14. Cheng C.H., Fu A.W. and Zhang Y. Entropy-based subspace clustering for mining numerical data. In: *Proceeding of the 1999 KDD international conference on knowledge discovery and data mining (KDD'99)*, San Diego, CA, pp 84–93, 1999.
15. Han J., Pei J., Yin Y, and Mao R. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. In *Data Mining and Knowledge Discovery*, 2003.
16. Zhang W., Yoshida T., Tang X. and Wanga Q. Text clustering using frequent itemsets. *Knowledge-Based Systems*, 23: 379–388, 2010.
17. Held M. and Buhmann J.M. Unsupervised On-Line Learning of Decision Trees for Hierarchical Data Analysis. *Proc. Advances of the Neural Information Processing Systems* (NIPS), 1997.
18. Bellot P. and El-Bèze M.. Clustering by means of Unsupervised Decision Trees or Hierarchical and K-means-like Algorithm. *RIAO'2000 Conference Proceedings* – Collège de France, Paris, France, April 12-14, vol. I, pp. 344-363, 2000.
19. Chien et. al. Compact Decision Trees with Cluster Validity for Speech Recognition. http://dx.doi.org/10.1109/ICASSP.2002.1005879. IEEE, 2002.
20. Basak J. and Krishnapuram R. Interpretable Hierarchical Clustering by Constructing an Unsupervised Decision Tree. *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, January 2005.
21. H. Ralambondrainy. A conceptual version of the K-means algorithm. *Pattern Recognition Letters* 16, 1147-1157, 1995.
22. Duda R.O., Hart P.E. and Stork D.E. *Pattern Classification*. John Wiley & Sons Ltd, ISBN: 0471056693, Ch. 10, pp. 517-600, 2001.
23. Webb AR. *Statistical Pattern Recognition*. John Wiley & Sons Ltd, ISBN: 0470845139, Ch. 10, pp. 361-407, 2002.
24. Han J., Cheng H., Xin D. and Yan X. Frequent pattern mining current status and future directions. *Data Min Knowl Disc*, 15: 55–86, 2007.
25. Fore N. and Dong G. Discovering Dynamic Logical Blog Communities Based on Their Distinct Interest Profiles. In: *SOTICS 2011: The First International Conference on Social Eco-Informatic*, ISBN: 9781612081632, IARIA, 2011.
26. Ayaquica I.O., Martínez J.F. and Carrasco J.A. Restricted Conceptual Clustering Algorithms based on Seeds. *Computación y Sistemas*, Vol. 11 No. 2, pp 174-187, 2007.
27. Wang J. and Han J. BIDE: efficient mining of frequent closed sequences. In: *Proceedings of the 20th International Conference on Data Engineering (ICDE'04)*, pp. 79–90, 2004.
28. MacQueen J.B. Some Methods for classification and Analysis of Multivariate Observations". 1. In: *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, pp. 281–297, 1967.
29. Halkidi M., Batistakis Y. and Vazirgiannis M. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, Volume 17 Issue 2-3, December 2001.
30. Morey L. and Agresti A. The Measurement of Classification Agreement: An Adjustment to the Rand Statistic for Chance Agreement. *Educational and Psychological Measurement SPRING 1984*, vol. 44 no. 1 pp. 33-37, 1984.
31. Quinlan JR. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc. 1993.

32. Frank A. and Asuncion A. *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. University of California, School of Information and Computer Science, 2010.

33. García-Borroto M, Martínez-Trinidad JF and Carrasco-Ochoa JA. A New Emerging Pattern Mining Algorithm and Its Application in Supervised Classification. *Lecture Notes in Computer Science*, Volume 6118/2010, 150-157, 2010.

34. Everitt B.S. *Cluster Analysis*. John Wiley & Sons, Inc., New York, 1974.

35. Jain A. and Dubes R. *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice Hall, 1988.

36. Pal N.R. and Bezdek J.C. On Cluster Validity for the Fuzzy c-Means Model. *IEEE Transactions on Fuzzy Systems*, Vol. 3, No. 3, 1995.

37. Bezdek J.C. and Hathaway R.J. Visual cluster validity for prototype generator clustering models. *Pattern Recognition Letters* 24, 1563–1569, 2003.

38. Brun M., Sima, C., Hua J., Lowey J., Carroll B., Suh E. and Dougherty E.R. Model-based evaluation of clustering validation measures. *Pattern Recognition* 40, 807 – 824, 2007.

39. Jain A.K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters* 31 651–666, 2010.

40. Frank E., Hall M.A., Holmes G., Kirkby R., Pfahringer B. and Witten I.H. Weka: A machine learning workbench for data mining. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, pages 1305–1314. Springer, Berlin, 2005.