Instituto Politécnico Nacional
Escuela Superior de Física y Matemáticas

Departamento de Matemáticas

Reconocimiento de Patrones Enfoque Conceptual

Tesis que para obtener el
Título de
Ingenieríero Matemático
Presenta:

Jorge Daniel González Arostico[1]

Director de Tesis: Dr. César Alberto Escobar Gracia

[1]Esta tesis se realizó con el apoyo parcial de CONACyT y COFFA bajo el proyecto 20082589

Reconocimientos

Agradezco a los Profesores: Adrián Alcántar Torres, Luis Alfonso Gódinez Contreras, Miguel Abel León Henández y Julio César Salas Torres, por haber revisado mi tesis y por sus valiosas sugerencias.

También agradezco a mis amigos: Axinia Figueroa, Cesar Escobar, Christian Curiel, Edgar Vargas, Luis Barbosa, Luis Quintos y Rogelio Ruiz a todos ellos por estár siempre conmigo y formar parte importante de mí vida.

Algoritmos de clasificación Conceptual y Semántico

Diciembre de 2008
Ciudad de México

Jorge Daniel González Arostico

Dedicatoria

Esta tesis se la dedico a:

A mi padre que ha sido el maestro de las cosas importantes en la vida, quien demuestra de forma extraña su creencia en una familia unida, y en todo tiempo ha estado atento de mis pasos cuidando de mí, y más que nada por brindarme su sabia experiencia. Él enseño en lo importante que es adquirir conocimiento y estar preparado para enfrentar la vida.

A mi hermano que siempre creyó en mí y nunca duda ni un segundo en que alcance mis metas, quien siempre me protege me da su compañía y me muestra más allá de lo que imagino, es quien siempre me ayuda a recobrar el sentido de las cosas y me coloca de nuevo en mi centro. Con su existencia me siento contento y sobre todo me siento completo. A él le debo mis logros y la visión de mis metas.

Finalmente a la más grande persona que ha estado conmigo en todo momento y ha tenido paciencia para mostrarme el camino que debe seguir una buena persona. Su fortaleza, optimismo y su cariño que otorga en cada acción para demostrar su infinito amor. Ella ha derramado un mar de sudor por el solo hecho de verme seguir adelante. Gracias a esa persona creo en la vida, en el amor incondicional y sobre todo me ayudo a creer en mí. Gracias madre, gracias por todo no voy a defraudarte.

A todos ellos gracias, mi pequeña familia, que me dieron la oportunidad de llegar hasta aquí y completar mis estudios, sé que fue un trabajo difícil y es por eso que les estaré eternamente agradecido.

Jorge Daniel González Arostico
Diciembre del 2008
Ciudad de México

Contenido

Capítulo 1

Introducción

1.1 El reconocimiento de patrones

El reconocer patrones es una actividad intrínseca del ser humano. Esta actividad la realizamos utilizando nuestros sentidos, a cada instante percibimos sonidos, imágenes, calor, etc. de esta manera recibimos la información de nuestro entorno. En psicología uno de los principales problemas consiste en comprender los procesos biológicos y mentales que transforman los estímulos externos en experiencias perceptivas con significado. Los procesos que nos permiten reconocer patrones son todavía un misterio en muchos de sus aspectos. Sin embargo, se asume frecuentemente un esquema de funcionamiento: Antes del reconocimiento, nuestros órganos sensitivos deben percibir primero un patrón. Después, para representar una experiencia perceptiva con significado, un patrón de la misma clase debe haber sido percibido previamente. Finalmente, se debe recordar la pasada percepción del patrón, y se debe establecer de alguna forma una correspondencia o equivalencia entre la percepción pasada y la presente.

El reconocimiento de patrones como ciencia es uno de los temas principales de la inteligencia artificial, su objetivo es hacer que los sistemas inteligentes interpreten la información que recibe de su entorno tal y como la hacen los seres humanos.

Con la llegada de las computadoras digitales hace treinta aos, un nuevo horizonte lleno de posibilidades se abrió ante nuestros ojos. Estas máquinas equipadas con dispositivos sensoriales (sensores y transductores) son capaces de observar el mundo real. Además de almacenar toda esta información y

recuperarla luego, también podían establecer relaciones entre observaciones pasadas y presentes. Todos los elementos de un sistema de reconocimiento de patrones estuvieron disponibles, y la idea de disear o programar máquinas que reconociesen patrones pronto se hizo uno de los proyectos más ambiciosos y fascinante.

Dentro del Reconocimiento de patrones existen cuatro problemas que son abordados de manera separada:

- **Selección de variables:** Consiste en seleccionar cuál es el tipo de características o rasgos más adecuados para describir los objetos. Se deben localizar los rasgos que inciden en el problema de manera determinante.

- **Clasificación sin aprendizaje:** También conocida como clasificación no supervisada, en éstos problemas no existe ninguna clasificación previa de objetos y en algunas ocasiones ni siquiera se han definido las clases.

- **Clasificación con aprendizaje parcial:** También conocida como de parcialmente supervisada, en éstos problemas existe una muestra de objetos sólo en algunas de las clases definidas

- **Clasificación con aprendizaje:** También es conocida como clasificación supervisada, en este tipo de problemas ya se encuentran definidas las clases, y éstas cuentan con algunos objetos previamente clasificados.

1.2 El problema de la CSA

El tarea fundamental de la CSA (clasificación sin aprendiaje) consiste en hallar la estructura interna de un conjunto de descripciones de objetos en el espacio de representación. Esta estructura interna, obviamente depende en primera instancia, de la selección del propio espacio de representación y de la forma en que los objetos se comparen, es decir del concepto de similaridad que se utilice y de la forma en que éste se emplee. Dicha estructuración se realiza sobre alguna de estas maneras:

1. Restringida: El número de agrupaciones en la que se extrucurarán los objetos está previamante definido.

2. Libre: Se desconoce en cuántas agrupaciones se estructurará el conjunto de objetos una vez definidos el espacio de representación y los conceptos de similaridad y la forma de usarlos. Es decir la estructura solo dependerá exclusivamente de las características del conjunto los objetos.

En cualquiera de los dos casos, un problema de clasificación sin aprendizaje consiste en hallar un procedimiento por el cual se pueda conocer la estructura interna del conjunto de descripciones de objetos dado. Para encontrar esa estructura existen tres formas generales de hacerlo, a saber.

1. El paradigma del conjunto cociente.
2. El paradigma del solapamiento.
3. El paradigma difuso.

1.3 Paradigmas de la CSA.

El *paradigma del conjunto cociente*, consisten en la formación de una partición del conjunto de objetos dado, bajo el supuesto que los mismos serán conjuntos en el sentido clásico de la Teoría de conjuntos. En otras palabras, de lo que se trata es de hallar el conjunto cociente del dado en el espacio de representación en cuestión. Esto supone que los agrupamientos serán ajenos. Aquí las propiedades que caracterizan a un agrupamiento dado contradicen las propiedades que caracterizan a cualquier otro de los restantes agrupamientos obtenidos.

El *paradigma del solapamiento* permite que las agrupaciones tengan elementos comunes, es decir, se trata de hallar un cubrimiento del conjunto de descripciones de objetos dado por subconjuntos (también en el sentido clásico) no necesariamente ajenos. Las propiedades que caracterizan a un agrupamiento dado pudieran ser satisfechas por otro de los agrupamientos restantes.

El *paradigma difuso*, sin embargo, parte de una suposición conceptual diferente: de los objetos no podemos afirmar categóricamente que pertenecen o no a un conjunto dado, sólo podemos hablar de grados de pertenencia (como es característico de la Teoría de los Subconjuntos Difusos, creada por Lotfi A. Zadeh en 1965). Así, el problema de la clasificación sin aprendizaje bajo esta suposiciones consiste en hallar una partición o bien un cubrimiento difuso del

conjunto de objetos dado. Es claro que en estos casos las propiedades que caracterizan a los conjuntos, subconjuntos difusos del universo en cuestión, son satisfechas en cierto grado por todos los objetos del universo de estudio.

En todos estos paradigmas hay factores comunes. Uno de ellos esencial para la solución del problema de la clasificacón sin aprendizaje, es la selección del *criterio de agrupamiento*.

La selección de un criterio de agrupamiento (se definirá mas adelante con precisión) puede realizarce de maneras diferentes.

Se llega al criterio de agrupamiento mediante la modelación matemática del problema y por la misma vía se llega al paradigma de realización de la estructuración del conjunto de objetos. Suponemos el paradigma, es decir, lo imponemos, y condicionamos el criterio de agrupamiento de modo tal que resulte una estructura acorde con el paradigma seleccionado.

El estudio de todos estos paradigmas se puede hacer bajo dos ópticas, que aunque muy relacionadas poseen diferencias, en apariencia sutiles, considerada de suma importancia para los análisis posteriores. Nos referimos a lo que pudiéramos denominar una óptica *clasificatoria* y una óptica *conjuntal*.

En la *Enfoque clasificatorio* se tiene un universo de objetos y se necesita agruparlos de tal modo que los objetos del mismo agrupamiento se parezcan mas entre sí que con objetos de otros agrupamientos.

En la *Enfoque conceptual* se tiene un universo de objetos y se necesita agruparlos de tal modo que los objetos que estén en el mismo agrupamiento cumplan (en cierto grado) la propiedad que caracteriza al agrupamiento (como conjunto en su determinación intencional).

El objetivo fundamental del problema de clasificación sin aprendizaje es el de conocer la estructura interna de una población de objetos dada. Esa población pudiera ser una clase de objetos en un problema de clasificación con aprendizaje o con aprendizaje parcial.

El interés en lograr esa estructura puede ser porque se desea posteriormente clasificar nuevos objetos ya que la "población" a la que se está haciendo referencia no es todo el universo de objetos de un problema en cuestión. Por ejemplo puede ser el interés, la necesidad de analizar los datos con fines de depuración, de seleccionar la mejor representación de la clase en cuestión.

A todos estos intereses asiste un factor común: las causa intrínsecas de la agrupación responden a la similitud en el enfoque clasificatorio, al cumplimiento de una propiedad en el enfoque conceptual.

1.4 Objetivo de la tesis

La presente tesis tiene como objetivo presentar el Reconocimiento de patrones bajo el enfoque conceptual en donde se abordará el problema de la clasificación sin aprendizaje. Para esto veremos 2 tipos de agrupamientos el Conjuntivo y Semántico estos nos ayudará a estructurar el conjunto de objetos de forma restringida y obedecerán el paradigma del conjunto cociente.

Así la estructura temática del texto será el siguiente. En el capítulo 2 explica el enfoque conceptual y en él desarrollaremos la teoría necesaria, así como los métodos conocidos para realizar una clasificación desde el punto de vista conceptual conjuntivo.

En el capítulo 3 desarrollaremos un método para clasificar de manera semántica, es decir relacionar términos semánticamente parecidos en la búsqueda de información sobre algún tema en especifico. En esta parte se dará al final el código fuente en C++ que implementa el métod presentado.

Capítulo 2

Enfoque Conceptual

En muchas ciencias aplicadas a menudo existe un problema para revelar la estructura de una determinada colección de objetos (situaciones, mediciones, observaciones, etc.) Un problema específico de este tipo es el de determinar una jerarquía de subcategorías significativas en la colección. Este problema se ha estudiado intensamente en el área de análisis de agrupamiento. Los métodos que existen formulan subcategorías (grupos) y únicamente se basan en la similitud de parejas (o proximidad) de los objetos, y pasan por alto la cuestión del "sentido" en los grupos obtenidos y no proporcionan alguna descripción de estos. Este capítulo presenta métodos que construyen jerarquías de subcategorías, de modo que cada subcategoría tiene una descripción general apropiada, que es una declaración que involucra atributos de los objetos y tiene una simple interpretación conceptual. Los atributos pueden ser variables de valor nominal o numéricas.

Los métodos descritos en éste capítulo han demostrado que para algunos problema bastante simples, los métodos tradicionales no son capaces de obtener una estructuración de objetos, que sea la más "natural" para las personas, los métodos presentados aquí son capaces de producir una solución de ese tipo. El conocimiento sobre la estructura de los objetos ayuda, por ejemplo, a reducir el espacio de búsqueda en la solución de problemas, dividiendo la adquisición de los conocimientos en subcasos, o en la organización de grandes bases de datos y resumiendo su contenido. Se cree que el problema de la estructuración inteligente de datos por computadora se convertirá en una de las tareas importantes para la investigación de IA (Inteligencia Artificial).

Una forma simple para estructurar datos es la agrupación, el cual es un proceso para determinar una jerarquía de subcategorías dentro de una colección

de objetos. Los métodos tradicionales de agrupación, basan la forman de subcategorías en el "grado de similitud" entre objetos: las subcategorías son colecciones de objetos dentro de un grupo cuya similitud es alta. Existen dos proceso para determinar la jerarquía de subcategorías ya sea con "bottom-up" o un "top-down".

Los métodos bottom-up (llamados "jerárquicos" en la literatura del grupo de análisis) recursivamente fusión única de objetos o colecciones de objetos, que termina con el conjunto original de objetos en la parte superior de la jerarquía (dendograma). Los métodos "top-down" ("no-jerárquicos") dividen recursivamente las colecciones de objetos en subgrupos, que termina solo cuando se asignan los objetos a las jerarquías.

Los "Top-down" son en su mayoría los métodos utilizados en la taxonomía numérica. Depende del grado de similitud para los objetos a ser agrupados, las diferentes versiones de la técnica se obtienen, como relación única, completa, o promedio de relación [14].

Los métodos "top-down" generalmente operan mediante una serie de perturbaciones, mientras realiza una búsqueda de agrupaciones que presenten dispersión mínima de objetos. Algunos métodos, por ejemplo, el ISODATA, utiliza mucha más heurística que ayudan a seleccionar el número óptimo de las agrupaciones. Los procesos aliados de "top-down" para agrupar características en lugar de objetos involucra técnicas de análisis y escalas multidimensionales. Muchos de los métodos de agrupación son sensibles a variables irrelevantes en los datos. El análisis y las escalas multidimensional son usados para seleccionar las variables más relevantes antes de proceder a agrupar los objetos. Estos métodos, sin embargo, están diseñados principalmente para las variables numéricas. Todas las técnicas tradicionales tienen una gran desventaja, porque su única base para la formación de agrupaciones es el grado de similitud (entre los objetos o grupos de objetos), las agrupaciones resultantes no necesariamente tienen alguna interpretación conceptual. El problema del "sentido" en los grupos obtenidos el investigador no lo tiene, esto representa una desventaja importante debido a que no sólo quiere agrupaciones, sino también una explicación de ellos en términos humanos.

En este capítulo describe un método para la determinar una estructura jerárquica de una determinada colección de objetos, la cual en cada nodo representa una descripción generalizada de subcategoría de los objetos. Las descripciones son conceptos conjuntivos que involucran atributos de objetos y tienen una interpretación muy simple. Los presentes métodos son un ejemplo de lo que llamamos en general, un "agrupamiento conceptual".

La etiqueta "agrupamiento conceptual" puede aplicase a cualquier método que determine una estructura en colección de objetos, donde los nodos representan "conceptos" que caracterizan a las subcategorías correspondientes, y los vínculos representan la relación entre los conceptos (por el término "concepto" nos referimos a una descripción, que implica propiedades de objetos y relaciones entre ellos). Los conceptos son descripciones de subcategorías conjuntivas, y los enlaces de interconexión en los niveles de la jerarquía representan el "siguiente nivel de generalidad", relación entre las descripciones (es decir, la descripción anterior es una generalización de las descripciones de todos los sucesores).

2.0.1 Medida de Similitud vs Coherencia Conceptual

Las técnicas tradicionales de análisis agrupacional claramente son no-conceptuales porque no tratar de descubrir el significado de los grupos u organizar los objetos en subcategorías con una breve interpretación conceptual. Como hemos mencionado anteriormente, este comportamiento se atribuye al uso de distancias estándar o medida de similitud como la única base para la agrupación. Con el fin de ser capaces de hacer un "agrupamiento conceptual", se tiene que saber más que el grado de similitud entre dos objetos o grupos de objetos. En concreto, el concepto de similitud se debe sustituir por un concepto más general de "coherencia conceptual".

La similitud entre dos objetos de la población a ser agrupados es caracterizada, en los métodos convencionales de análisis de datos, por el valor de la función de similitud aplicada en las descripciones de objetos (eventos). Las descripciones son normalmente vectores, cuyas componentes representan un valor cualitativo o cuantitativo de las variables utilizadas para describir a los objetos. En muchas ocasiones la distancia recíproca se utiliza como función de similitud. La medida de distancia para tales fines no satisface a todos los principios de la función de la distancia (en concreto, la desigualdad del triángulo). Un estudio de distintas distancia y la medida de similitud son mencionadas por Diday & Simon [4] y Anderberg [3]. Como ya se dijo, la medida convencional de similitud son "contexto libres", es decir, la similitud entre dos eventos A y B es una función sólo de estos 2:

$$Similitud(A, B) = f(A, B) \tag{2.1}$$

Recientemente, algunos autores [5] han introducido un "contexto sensible" en las medidas de similitud, es decir:

$$Similitud(A, B) = f(A, B, E) \tag{2.2}$$

En donde la similitud entre A y B no sólo depende de estos, sino también de otros puntos contexto. Ambos enfoques sólo se basan en el conocimiento de datos individuales. Por lo tanto esos métodos son incapaces de capturar la "propiedad gesticulada" de los objetos, es decir, una propiedad que característiza a determinadas configuraciones de eventos que se consideran en su conjunto, pero no cuando se lo considera como eventos independientes. Con el fin de detectar esas propiedades, el sistema debe conocer no sólo los eventos, sino también determinados "conceptos". Para ilustrar este punto, vamos a considerar un problema de agrupación de eventos

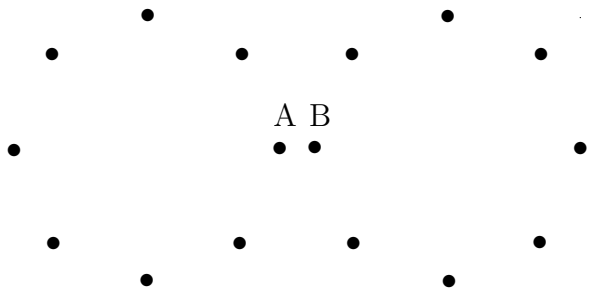


Fig. 1 Ilustración de la agrupación conceptual

Una persona teniendo en cuenta el problema en la fig. 1 por lo general, lo describe como "dos círculos". De este modo, los punto A y B, aunque se está muy cerca entre sí, se colocan en grupos separados. En este caso, la solución implica partir a los eventos en grupos pero no sobre la base de distancia entre ellos, sino sobre el "concepto de pertenencia". Esto significa que los eventos se colocan en el mismo grupo si estos representan el mismo concepto. En nuestro ejemplo, los conceptos son los círculos.

Esta idea es la base conceptual de la agrupación. Desde el punto de vista de

la agrupación conceptual la "similitud" entre dos puntos A y B, que llamaremos "coherencia conceptual", y es una función no sólo de estos puntos sino también sobre el contexto de puntos en E y el conjunto de conceptos C, que están disponibles para describir A y B entre sí:

$$Similitud(A, B) = f(A, B, C, E) \tag{2.3}$$

Para ilustrar una medida de "coherencia conceptual", vamos a suponer que C es el conjunto de conceptos de figuras geométricas, como círculos, rectángulos, triángulos, etc. Una medida de coherencia conceptual puede definirse, por ejemplo, como

$$S(A, B, E, C) = \max_i \frac{\#\mathrm{e}(i) - 1}{\mathrm{area}(i)} \tag{2.4}$$

En donde

i índices figuras geométricas que se especifican en C y abarcan a los puntos A y B

e (i) es el número total de eventos en E cubiertos por la figura i

área (i) es el área de la figura i

(la constante "-1" en el numerador asegura que la "coherencia conceptual" reduce a una medida de similitud convencional, es decir, una reciprocidad de la distancia, cuando no son puntos contextuales en E se tienen en cuenta y C es el espesor de línea en los eventos.)

Esta medida se menciona únicamente para ilustrar la diferencia entre la tradicional similitud y la coherencia conceptual. Este no emplea el método aplicado en el agrupamiento conceptual descrito en este capítulo. La idea conceptual de la agrupación se ha introducido por Michalski [1] y ha evolucionado a partir de trabajos anteriores de él por colaboradores en la generación del cubrimiento uniclass (es decir, la descripciones disjuntas de una clase de objetos especificados sólo ejemplos positivos de la clase). Un programa realizado por computadora sobre diversos resultados experimentales en la determinación de cubrimiento uniclass son descritos por Stepp [12].

2.1 Agrupamiento conceptual conjuntivo

Está técnica se basa en una serie de trabajos realizandos fundamentalmente por Michalski a partir de 1980. La motivación esencial era lograr que el agrupamiento obtenido brindara al especialista del área específica una información acerca del significado conjuntual del agrupamiento. Se supone que los objetos pueden ser descritos en términos de variables cuantitativas y cualitativas. Se afirma que las relaciones entre los valores de las variables y los objetos pueden ser descritas mediante una lógica bivalente.

En este capítulo resumiremos brevemente los conceptos formales de las técnicas de inferencia inductiva aplicadas y de las definiciones del lenguaje descriptivo utilizado que es VL_1 (Sistema Lógico de Variables Valuadas).

Sean $x_1, ..., x_n$ variables discretas que han sido seleccionadas para describir objetos de una población a ser agrupada. Para cada variable se define un conjunto de valores o dominio, éste contiene todos los posibles valores que esta variable puede tomar para cualquier objeto de la población. Supongamos que los conjuntos de valores para son finitos y definidos positivos, {0, 1, 2, ...}, por lo tanto pueden ser representados como

$$D_i = \{0, 1, ..., d_{i-1}\}, \qquad i = 1, 2, ..., n. \tag{2.5}$$

En general, los conjuntos de valores pueden ser diferentes, no sólo con respecto a su extensión, sino también con respecto a la estructura relativa de sus elementos.

Definición 2.1.1 *Un* **evento** *se define como cualquier sucesión de valores de variables $x_1, x_2, ..., x_n$ de la siguiente forma:*

$$e = (r_1, r_2, ..., r_n) \tag{2.6}$$

En donde $r_i \in D_i, i = 1, \ldots, n$

Definición 2.1.2 *El conjunto de todos los posibles eventos,* **E**, *se llama* **espacio de eventos:**

$$\mathbf{E}(d_1, d_2, \ldots, d_n) = \{e_i\}_{i=1}^{d} \tag{2.7}$$

En donde $d = d_1 \cdot d_2 \cdot \cdots \cdot d_n$, es el tamaño del espacio

Ejemplo.- Para describir las definiciones anteriores, tomemos x_1, x_2, x_3, x_4 como variables que escriben a los eventos e_i;

Variable	*Dominio*
x_1	$D_1 = \{0, 1, 2\}$
x_2	$D_2 = \{0, 1\}$
x_3	$D_3 = \{0, 1, 2, 3\}$
x_4	$D_4 = \{0, 1, 2\}$

Puesto que se tiene $|D_1| = d_1 = 3, |D_2| = d_2 = 2, |D_3| = d_3 = 4$, $|D_4| = d_4 = 3$, el tamaño del nuestro universo consta de 72 eventos, se representa como $\mathbf{E}(3, 2, 4, 3)$, en donde el orden de los indices se corresponde con:

	x_1	x_2	x_3	x_4
e_1	(0	0	0	0)
e_2	(0	0	0	1)
e_3	(0	0	1	0)
$\vdots$		$\vdots$		
e_{70}	(2	1	3	0)
e_{71}	(2	1	3	1)
e_{72}	(2	1	3	2)

es decir son enumerados con el orden léxicografico usual.

Definición 2.1.3 *La* **distancia sintática**, $\delta(e_1, e_2)$, *entre los eventos* e_1 *y* e_2 *de* $\mathbf{E}$, *es definida como el número de variables que toman diferentes valores en* e_1 *y* e_2.

Ejemplo.- Tomando dos eventos cualquiera en el espacio anterior, por ejemplo:

	x_1	x_2	x_3	x_4
e_4	(0	0	1	0)
e_{64}	(2	1	1	0)

note que x_1 y x_2 son las únicas variables que toman diferentes valores en los eventos, por lo que su distancia sintática entre el par de eventos resulta ser $\delta(e_4, e_{64}) = 2$.

Definición 2.1.4 *Para el evento* $e = (x_1, x_2, \ldots, x_n)$ *del espacio* $\mathbf{E}$ *se le puede asignar un* **valor único** $\gamma(e)$ *de acuerdo a la formula:*

$$\gamma(e) = 1 + x_n + \sum_{i=n-1}^{1} x_i \prod_{k=n}^{i+1} d_k \tag{2.8}$$

En donde x_n *es el valor de la componente* n *en el evento,* d_k *es el número de valores que la variable* x_k *toma y 1 se ulilizó para que los eventos sea numerados apartir de 1.*

Ejemplo.- En el espacio $\mathbf{E}(5, 6, 4, 3)$ un valor único de la función $\gamma(e)$ para el evento $e = (3, 4, 1, 2)$ es:

$$\gamma(e) = 1 + 2 + 1 \cdot 3 + 4 \cdot 4 \cdot 3 + 3 \cdot 6 \cdot 4 \cdot 3 = 270$$

El número $\gamma(e)$ lo utilizaremos como subíndice del evento, es decir, nos referiremos al evento como e_{270} y en ocasiones también se describirá al evento junto con sus componentes para ser más explícitos. Es fácil ver que a partir del valor único $\gamma(e)$ junto con los valores $d_1, d_2, \ldots, d_n$, se puede determinar al evento correspondiente $e = (x_1, x_2, \ldots, x_n)$. En primer lugar dividiendo a $\gamma(e) - 1$ por d_n, el residuo es el valor de la variable x_n. El resultado de la división anterior se dividirá por d_{n-1} y su residuo es el valor de la variable x_{n-1}, y así sucesivamente, hasta encontrar el valor de x_l.

Por otro lado es importante conocer la forma en como están distribuidos los eventos del espacio, lo cual no ayudará en posteriores cálculos. Un ejemplo de esta representacón del espacio $\mathbf{E}(4,4,4)$, en una forma Euclidiana discreta para los eventos $e_5, e_{10}, e_{18}, e_{24}, e_{28}, e_{35}, e_{38}, e_{43}, e_{54}$ y e_{57}, se muestra en la figura 2.

Esta forma de representar a los eventos (o puntos) ha sido utilizada en diferentes áreas de la ciencia y es en la que más se han apoyado. Sin embargo para nosotros resulta difícil de visualizar la distribución de los eventos cuando el número de variable es mayor a 3, por lo tanto no es práctico ésta

representación. Lo que nos lleva a buscar una forma mucho más sencilla de describir la distribución de los eventos.

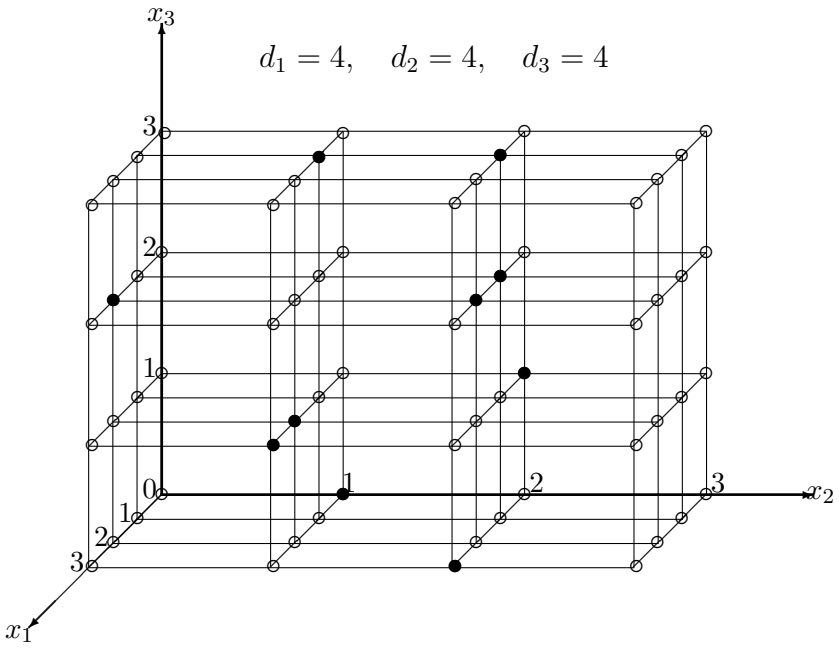


Fig. 2 Representación Euclidiana discreta del espacio **E**(4, 4, 4)

Esta forma alterna la que llamaremos representación plana ó **Gráfica plana**, será construida a partir de los siguientes pasos. En primer lugar se determina v el cual es un número máximo que satisface la relación

$$d_1 \cdot d_2 \ldots, d_v \leq d_{v+1} \cdot d_{v+2} \ldots d_n \tag{2.9}$$

El esquema visual resulta más satisfactorio si los productos en ambos lados de 2.9 son aproximadamente iguales. (Para esto puede ser útil permutaciones de los d_i's). Se traza un rectángulo arbitrario dividiéndolo en $d_1 \cdot d_2 \ldots, d_v$ filas y $d_{v+1} \cdot d_{v+2} \ldots d_n$ columnas de acuerdo a las siguientes reglas:

i. En el primer paso, se divide al rectángulo con líneas horizontales d_1 filas y se le asignan los valores $0, 1, \ldots, d_1 - 1$ (valores de la variable x_1), ordenándolos de abajo hacia arriba. En el paso i, $1 < i \leq v$, se divide

cada fila generado en el paso $i-1$ d_i filas, y se asignan los mismos valores $0, 1, \ldots, d_i$, de abajo hacia arriba ($d_i = d_i - 1$).

ii. En los pasos $v+1, v+2, \ldots, n$ se hace de la misma forma que en el caso anterior, dividiendo al rectángulo con líneas verticales como columnas.

iii. Las líneas generadas en el paso, $i = 1, 2, \ldots, n$ son llamados ejes de x_i. Variando el grosor de los ejes en las diferentes variables: las líneas más delgadas puede ser ejes de x_v y x_n, al lado más delgado de los ejes de x_{v-1}) y x_{n-1}, y asísucesivamente hasta agotar todos los ejes. (por lo tanto, si n es aún y $v = n/2$, la más gruesa será la x_1 ejes y x_{v+1}).

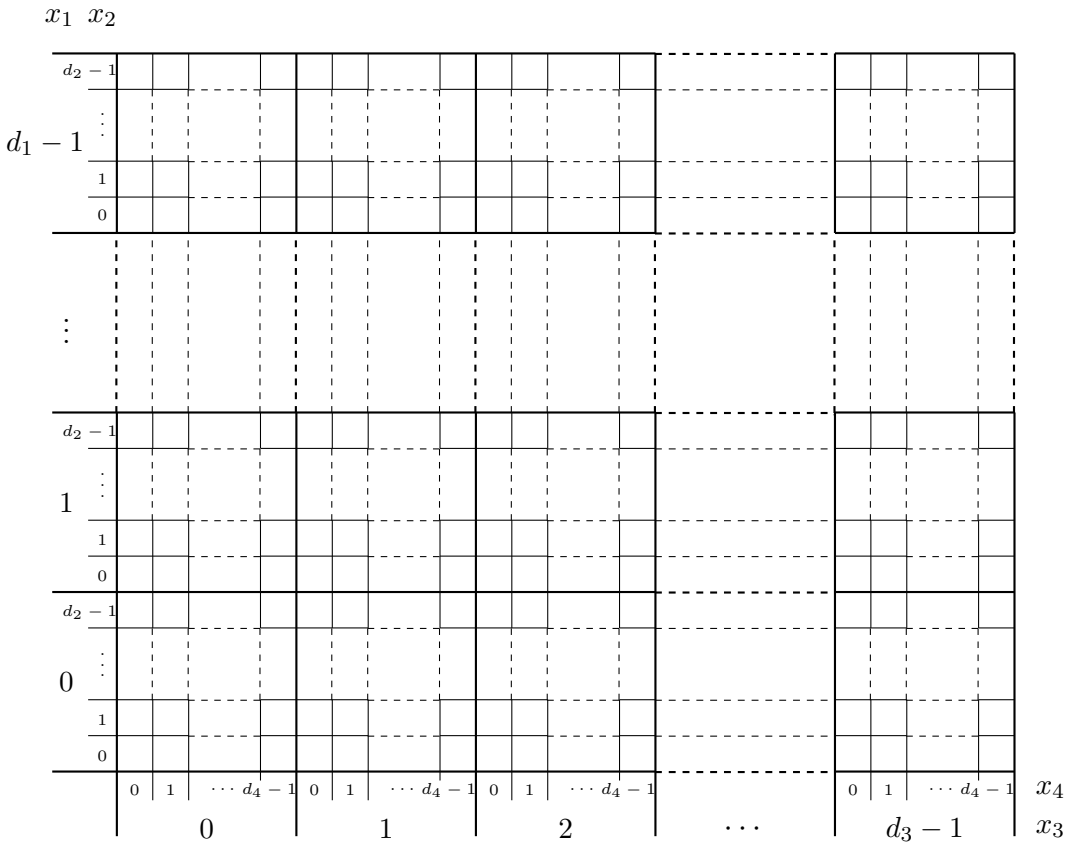


Fig. 3 Diagrama de $\mathbf{E}(d_1, d_2, d_3, d_4)$

En la Figura 3 se presenta una forma general del diagrama para $n = 4$. Un único vector $(x_1, x_2, \ldots, x_n)$ corresponde a cada fila y un único vector $(x_{v+1}, x_{v+2}, \ldots, x_n)$ corresponde a cada columna del diagrama. La intersección de cualquier fila con todas las columnas se llama celda. Cada celda representa un elemento del espacio de eventos **E** determinada correspondiente a la fila y la columna, respectivamente, cuya intersección produce una celda. El diagrama compuesto por $d = d_1 \cdot d_2 \cdots d_n$ celdas (corresponde al número de eventos en **E**.)

Definición 2.1.5 *Una expresión de relación.*

$$[x_i \# R_i] \tag{2.10}$$

En donde R_i, tiene como dominio uno o más elementos del dominio de x_i, y # se establece para cualesquiera de los operadores relacionales o símbolos $(\neq, =, \leq, <, \geq, >)$, y es llamado un VL_1 selector ó un **selector**.

El selector $[x_i = R_i]$ ($[x_i \neq R_i]$) se interpreta como que el valor de $x_i \in \{R_i\}$ (que el valor de $x_i \notin \{R_i\}$). En caso de que la variables sean lineales, el operador "=" puede ser reemplazado por los operadores relacionales $\geq, >, <, \leq$.

Estos son algunos de los ejemplos de selectores:

[altura = alto]
[longitud $\geq$ 2]
[color = azul, rojo], (El color es azul ó rojo)
[tamaño $\neq$ mediano] (El tamaño no es mediano)
[peso = 2..5] (El peso esta entre 2 y 5)

Definición 2.1.6 *Un producto lógico de selectores se llama un* **complejo lógico** *(l-complejo):*

$$\bigcap_{i \in I} [x_i \# R_i] \tag{2.11}$$

En donde $I \subseteq 1, 2, \ldots, n$ un subconjunto de indices y $R_i \subseteq D_i$ es subconjunto de valores del dominio.

Se dice que un evento e satisface a un *l-complejo*, si los valores de las variables en e satisfacen a todos los selectores en el *l-complejo*. Por ejemplo, el evento $e = (2, 7, 0, 1, 5, 4, 6)$ satisface al *l*-complejo $[x_i = 2, 3]$ $[x_3 \leq 3]$ $[x_5 = 3..8]$(una concatenación de selectores implica una conjunción).

Un *l*-complejo puede ser visto como una representación exacta de los eventos a los cuales satisface, por ejemplo, el *l*-complejo mencionado arriba es la representación simbólica de todos los eventos para los cuales x_1 es 2 ó 3, x_3 es menor o igual a 3, y x_5 se encuentra entre 3 y 8.

Cualquier conjunto de eventos para los cuales existen en *l*-complejo que es satisfecho por esos eventos y sólo por ellos, se llama un *s*-complejo. En adelante si α es un *s*-complejo, entonces por $\hat{\alpha}$ denotaremos al *l*-complejo respectivo.

Sea E un conjunto de eventos en $\mathbf{E}$, que son descripciones de objetos a ser agrupados. Los eventos en E son llamados eventos observados, y los eventos en $\mathbf{E} \backslash E$ son objetos no descritos llamados eventos no observados. Sea α un *complejo* que cubre algunos eventos observados y no observados.

Definición 2.1.7 *El número de eventos no observados en α se llama* **dispersión** *y se denota por $s(\alpha)$.*

El número de eventos observados (descripciones de objetos) en α se denota por $p\,(\alpha)$. El número total de eventos en α es entonces la suma de eventos observados y no observados (que se encuentran el $\mathbf{E} \backslash E$), $t\,(\alpha) = p\,(\alpha) + s\,(\alpha)$.

Si *s-complejo* α es presentado como *l-complejo* $\hat{\alpha} = \bigcap_{i \in I} [x_i \# R_i]$, entonces el número $t(\alpha)$ puede ser calculado como:

$$t(\alpha) = \prod_{i \in I} c(R_i) \cdot \prod_{i \in I} d_i \tag{2.12}$$

En donde $I \subseteq \{1, 2, \ldots, n\}$ es el conjunto de indices que de las variables en el complejo α, $c(R_i)$ es la cardinalidad de R_i y d_i es la cardinalidad de los valores que toma de la variable de x_i.

Los complejos pueden ser vistos como $\hat{\alpha}$ que es, como se mencionó anteriormente, una concatención de selectores o solamente como α que es el conjunto de eventos a los que cubre. La dispersión en estos casos resulta la misma (la única diferencia radica en calcularla dependiendo de como es representado el complejo).

Ejemplo.- Si para el espacio de eventos:

$$\begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \end{array} \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \end{array}$$

Sea $E = \{e_2, e_4, e_6, e_8\}$ el conjunto de eventos observados en el espacio, $\mathbf{E}\backslash E = \{e_1, e_3, e_5, e_7\}$ el conjunto de eventos no observados, $\alpha = \{e_1, e_2, , e_5, e_6\}$ el s-complejo, por lo que $p(\alpha) = 2$ (el números de eventos observados) y $s(\alpha) = 2$ (el número de objetos no observados), y $t(\alpha) = 2 + 2 = 4$ (el número total de eventos observados por alfa).

Por otro lado si para $\hat{\alpha} = [x_2 = 0][x_3 = 0, 1]$ el l-complejo, entonces para calcular $t(\alpha)$ tenemos:

$$\begin{array}{lcl} I = \{2, 3\} & & \\ R_2 = \{0\} & \Rightarrow & c(R_2) = 1 \\ R_3 = \{0, 1\} & \Rightarrow & c(R_3) = 2 \\ D_1 = \{0, 1, 2\} & \Rightarrow & d_1 = 3 \end{array}$$

$$t(\alpha) = 1 \cdot 2 \cdot 3 = 6$$

Nota: Esta forma de calcular $t(\alpha)$ se utiliza cuando el complejo esta representado como $\hat{\alpha}$.

Definición 2.1.8 *El* **grado de generalidad** $g(\alpha)$ *de un complejo* α*, se define como:*

$$g(\alpha) = \log \frac{t(\alpha)}{p(\alpha)} = \log \left(1 + \frac{s(\alpha)}{p(\alpha)}\right) \tag{2.13}$$

El *l-complejo* $\hat{\alpha}$ puede ser visto como una descripción generalizada de las descripciones de los eventos en α. La dispersión, como se definió arriba, se puede utilizar como una medida de grado para la cual, la descripción $\hat{\alpha}$ generaliza sobre la descripción de los objetos. Si la dispersión es cero, entonces, la descripción solamente cubre los objetos observados (cero generalización). Cuando la dispersión para un *complejo* dado aumenta, también lo hace el grado para el cual la descripción $\hat{\alpha}$ generaliza sobre las descripciones.

Sea L un conjunto de complejos (o eventos), y R_i el conjunto de todos los valores distintos que tiene la variable x_i.

Definición 2.1.9 *La operación que transforma a L en el complejo* $\bigcap_{i=1}^{n}[x_i = R_i]$ *se llama unión referenciada ó* **refunión**. *El complejo resultante es llamado el complejo de cubrimiento minimal ó* **mc-complejo** *de L y es denotado como RU(L)(refunción).*

Si $R_i = D_i$ para algún i, entonces el selector correspondiente es eliminado del complejo. La refunión es entonces la función que transforma un conjunto de complejos, en un complejo de cubrimiento minimal.

Ejemplo.- Supongamos que x_1 y x_2 tomar los valores $\{0, 1, 2\}$ y x_3 los valores $\{0, 1\}$, es decir, el espacio de eventos $\mathbf{E}(3, 3, 2)$ y consta de 18 elementos. Consideremos a L el conjunto de complejos compuestos como:

$$\hat{\alpha_1} = [x_1 = 0, 2][x_3 = 1]$$
$$\hat{\alpha_2} = [x_1 = 1, 2][x_3 = 1]$$

entonces los valores que toma cada una de las variables en los complejos

$$\begin{array}{lll} R_1 = \{0, 1, 2\} & \text{para} & x_1 \\ R_2 = \{1\} & \text{para} & x_2 \\ R_3 = \{1\} & \text{para} & x_3 \end{array}$$

por lo tanto

$$RU(L) = [x_2 = 1][x_3 = 1]$$

dado que $R_1 = D_1 = \{0, 1, 2\}$ para la variable x_1, por lo que fue eliminado el selector correspondiente. Con la ayuda de las gráficas planas se observa en cada unas de ellas los eventos que son cubiertos por los complejos.

La figura 4 nos muestra con la ayuda de una gráfica plana los 6 eventos que $\hat{\alpha_1}$ cubre.

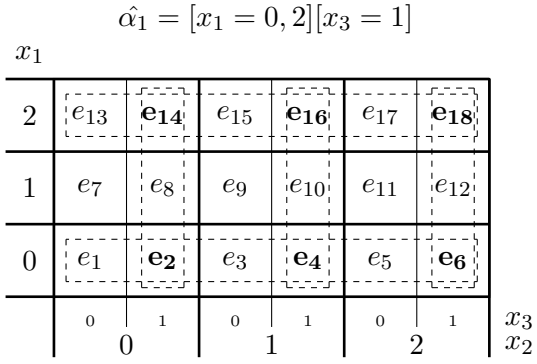


Fig. 4 Ilustración de cubrimiento de $\hat{\alpha_1}$

La figura 5 nos muestra los eventos de $\hat{\alpha_2}$ cubre. Aunque en estos complejos la dispesión es 0. En otros casos las gráficas ayudaría a visualizar facílmente los eventos en dispersión.

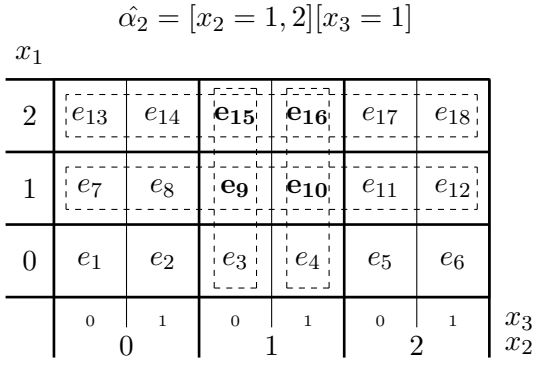


Fig. 5 Ilustración del cubrimiento de $\hat{\alpha_2}$

La refunión hecha apartir de los complejos anteriores, vemos que sólo cubre a tres eventos, aunque es poco lo que cubre eso nos asegura que existe una dispersión minima en el complejo resultante de la refunión.

$$RU(L) = [x_2 = 1][x_3 = 1]$$

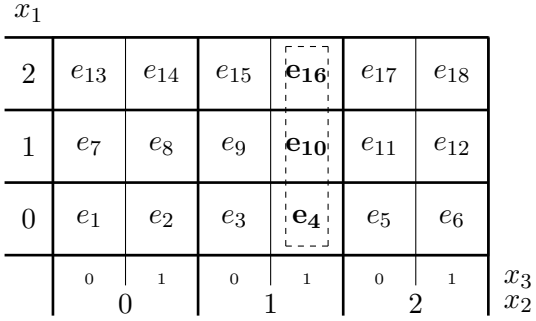


Fig. 6 Ilustración de la Refunión

Teorema 2.1.10 *El mc-complejo de un conjunto de eventos es el complejo de dispersión minimal de entre todos los complejos que cubren al conjunto de eventos.*

Demostración Sea α el $mc - complejo$ de un conjunto de eventos E:

$$\alpha = RU(E) = \bigcap_{i=1}^{n} [x_i = R_i] \tag{2.14}$$

En donde $R_i \subseteq D_i$. Supongase que $\gamma = \bigcap_{i=1}^{n} [x_i = R_i]$ es un complejo que cubre a E y que tiene pequeñas dispersiones en α. Si esto es verdad, entonces existen P_i, tales que $P_i \subset R_i$. Pero R_i, según la definición anterior, contiene a todos lo valores que puede tomar x_i, en los eventos del conjunto E. Por lo tanto, si $P_i \subset R_i$, entonces el complejo α probablemente no cubrirá a todos lo eventos de E, lo que es una contradicción. □

Sea E un conjunto de eventos que cubre el complejo α.

Definición 2.1.11 *El conjunto E es llamado el* **núcleo** *de α, y el complejo $\alpha^* = RU(E)$ es llamado el α corte.*

Del teorema anterior tenemos $\alpha^* \subseteq \alpha$.

Teorema 2.1.12 *Si E_1 y E_2 son dos conjuntos de eventos disjuntos, entonces.*

$$s(RU(E_1)) + s(RU(E_2)) \leq s(RU(E_1 \cup E_2)) \tag{2.15}$$

Demostración De acuerdo con el teorema 1, $RU(E_1)$ y $RU(E_2)$ tienen un pequeño conjunto de posibles dispersiones entre todos los complejos que cobren a E_1 y a E_2 respectivamente. Siendo que E_1 y E_2 son disjuntos, entonces el teorema se cumple. □

Teorema 2.1.13 *Sea α_1 y α_2 dos complejos intersectados, cuya unión de cubrimientos en un conjunto de eventos E. Decimos que $E_1(E_2)$ denota el conjunto de eventos en $\alpha_1(\alpha_2)$ que son cubiertos por solo un complejo (el núcleo relativo del complejo). Si α_1' y α_2' son dos complejos cualesquiera que cubren el mismo conjunto de eventos E. Y si $RU(E_1)$ y $RU(E_2)$ son complejos disjuntos, entonces:*

$$s(RU(E_1)) + s(RU(E_2)) \leq s(\alpha_1') + s(\alpha_2') \tag{2.16}$$

Demostración El teorema es una consecuencia inmediata del teorema anterior y la premisa de que α_1' y α_1' son complejos disjuntos. □

Ejemplo.- Considere el espacio de eventos **E**(3,3,3). Enumerados con el orden léxicografico. La siguiente gráfica plana nos proporciona la descripción de los eventos en el espacio mencionado.

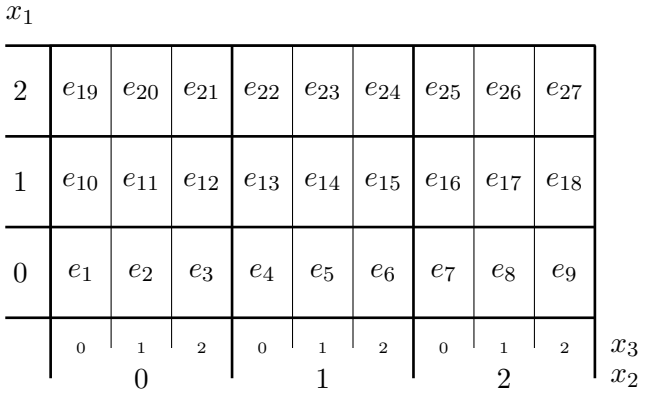


Fig. 4 Espacio de eventos **E**(3, 3, 3)

Sea $\alpha_1 = \{e_1, e_2, e_3, e_{10}, e_{11}, e_{13}, e_{14}, e_{15}\}$ y $\alpha_2 = \{e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{27}\}$, cuyos núcleo relativo para cada uno de ellos son $E_1 = \{e_1, e_2, e_3, e_{10}, e_{11}\}$ y

$E_2 = \{e_{16}, e_{17}, e_{18}, e_{27}\}$ respectivamente. Para $\alpha'_1 = [x_1 = 0, 1][x_2 = 0, 1]$ y $\alpha'_2 = [x_1 = 2][x_2 = 2]$, donde son disjuntos y cubren a todo el conjunto E. Como $RU(E_1) = [x_1 = 1, 2][x_2 = 0]$ y $RU(E_2) = [x_1 = 1, 2][x_2 = 2]$, entonces

$$s(RU(E_1)) + s(RU(E_2)) = 1 + 2 \leq 3 + 2 = s(\alpha'_1) + s(\alpha'_2)$$

como puede verse la refunión de los núcleos nos proporciona menos dispersión que los complejos α'_1 y α'_2.

Definición 2.1.14 *Una* **estrella** $G(e|F)$ *de* e *respecto al conjunto de eventos* F *es el conjunto de todos los complejos maximales respecto a la inclusión que cubre el evento* e *y no cubre evento alguno en* F.

Un complejo α es máximal respecto a la inclusión con relación a la propiedad P, si no existe un complejo α^* con la propiedad P, tal que $\alpha \subset \alpha^*$.

Definición 2.1.15 *Sea E_1 y E_2 dos conjuntos de eventos disjuntos $E_1 \bigcap E_2 = \emptyset$. Un* **cubrimiento** $COV(E_1|E_2)$ *de E_1 con respecto a E_2, es cualquier conjunto de complejos, $\{\alpha_j\}_{j\in J}$, tal que para cada evento $e \in E_1$, existe un complejo $\alpha_j, j \in J$, que lo cubre, y ninguno de los complejos α_j cubre objeto alguno de E_2, es decir,*

$$E_1 \subseteq \bigcup_{j\in J} \alpha_j \subseteq \mathbf{E} \backslash E_2 \tag{2.17}$$

Un cubrimiento en el cual todos los complejos son conjuntos de pares disjuntos se llaman **cubrimiento disjunto**. Si el conjunto E_2 es vació, entonces, un cubrimiento $COV(E_1|E_2) = COV(E_1|\emptyset)$ se denota simplemente como $COV(E_1)$. Un cubrimiento disjunto $COV(E)$ que se integra de k complejos se llama k-partición de E.

Ejemplo.- Considere el espacio de eventos en el ejemplo 2.0.1, entonces el número de complejos que describen al espacio, de acuerdo con el dominio y el número de variables, es de 343. Sea

$$E_1 = \{e_1, e_2, e_3, e_4, e_5, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}\}$$
$$E_2 = \{e_{15}, e_{16}, e_{17}, e_{18}, e_{24}, e_{25}, e_{26}, e_{27}\}$$

ambos tienen 60 complejos que describen a los eventos en ellos, es decir, $\{E_1 \cup E_2\} \subseteq \{\alpha_i\}_{i=1}^{120} \subset \mathbf{E}$. Sea e_4 un evento cuyo complejo que lo describe y cubre es $\alpha_1 = [x_1 = 1][x_2 = 1][x_3 = 0, 1]$, éste no cubre a ninguno evento en E_2 ya que no existe evento satisfaga al complejo en este conjunto. Más aún si $\alpha_2 = [x_2 = 0]$ cubre algunos eventos en E_1, no existe evento alguno en E_2 que cumpla con el selector.

Definición 2.1.16 *La* **dispersión** *(el grado de generalidad) de un cubrimiento es definida como la suma de dispersiones (el grado de generalidad) de los complejos en el cubrimiento.*

2.1.1 Complejo para la representación de agrupamiento

Primero describiremos las siguientes propiedades:

Teorema 2.1.17 *Para cualquier espacio de eventos* $\mathbf{E}$ *y un entero* $k \leq d_1, d_2, \ldots, d_n$ *(donde* d_i *es la cardinalidad del conjunto de valores de la variable* x_i*), existen k particiones disjuntas a pares de complejos* $\alpha_1, \alpha_2, \ldots, \alpha_k$ *que cubren todo el espacio* $\mathbf{E}$*, es decir:*

$$\bigcup_{i=1}^{k} \alpha_i = \mathbf{E} \tag{2.18}$$

Demostración El teorema equivale a decir que cualquier espacio de eventos puede ser particionado en un número arbitrario de complejos (pero, no más grande que la cardinalidad de $\mathbf{E}$). Para verlo, tomaremos cualquier subconjunto de variables tal que el producto aritmético correspondiente de las d_i's sea mayor o igual a k:

$$k \leq k' = \prod_{i \in I} d_i \tag{2.19}$$

Sea $R_i, j = 1, 2, \ldots,$ todas las posibles sucesiones de valores de la variable $x_i, i \in I$. Construyendo el complejo:

$$\alpha_j = \bigcap_{i \in I} [x_i = r_{ij}] \tag{2.20}$$

En donde $r_{ij}, i \in I, j = 1, 2, \ldots$ denota un valor de la variable x_i en las sucesión. Obviamente, los complejos α_i son disjuntos a pares y cubre a todo el espacio **E**. Si $k' > k$, entonces los $k' - k$ complejos se juntan con los restantes complejos, de acuerdo con la siguiente fórmula:

$$\beta[x_i = a] \cup \beta[x_i = b] \equiv \beta[x_i = a, b] \tag{2.21}$$

En donde β denota una conjunción de selectores involucrando las demás variables x_i. Esto siempre es posible, porque para cualquier $x_i, i \in I$, existen d_i complejos α_j que difieren solo en el valor de x_i. □

Desde el punto de vista de agrupamiento, una pregunta muy interesante es que si para cualquier conjunto de eventos E en el espacio **E**, existe siempre un número arbitrario $k \leq c(E)$ de particiones disjuntas a pares de complejos, tal que estos complejos no llenen a todo el espacio **E**, y que esta partición de E en k subconjunto sean no vacíos. Una posible respuesta implica que para cualquier conjunto de eventos puede ser partido en un número de subconjuntos supuestos con anterioridad.

Teorema 2.1.18 (El principio de suficiencia). *Para un espacio de eventos* **E**, *y cualquier conjunto de eventos* $E = \{e_1, e_2, \ldots, e_k\}$, $E \subseteq$ **E**, *existe al menos un conjunto* k *de complejos disjuntos a pares,* $\alpha_1, \alpha_2, \ldots, \alpha_k$, *tal que cada complejo contiene un evento del conjunto* E

$$e_j \in \alpha_j \qquad j = 1, , 2, \ldots, k \tag{2.22}$$

Y la unión de los complejos llena al espacio **E**:

$$\bigcup_{j=1}^{k} \alpha_j = \mathbf{E} \tag{2.23}$$

Demostración La idea básica de la desmostración es mostrar que para cualquier $E = \{e_1, e_2, \ldots, e_k\}$. $E \subseteq$ **E**, esto siempre es posible si construimos un árbol, donde en cada nodo es asignada la variable $x_i, i \in 1, 2, \ldots, n$, y en cada rama del nodo x_i se asignan los elementos de D_i (el conjunto de valores de x_i), y los niveles representan al complejo α_j, tal que cada complejo cubre sólo al evento e_j, y la unión de los complejos cubre al espacio **E**.

Supóngase que $e_j = (x_{1j}, x_{2j}, \ldots, x_{nj}), j = 1, 2, \ldots, k$, donde $x_{ij} \in D_i$. Tome cualquier variable x_p, que tome valores diferentes en eventos de E.

Supóngase que estos valores son $a_1, a_2, \ldots, a_z$. Parta al conjunto de valores, D_p de x_p, en subconjuntos $\{a_1\}, \{a_2\}, \ldots, \{a_{z-1}\}, \mathbf{A_z}$, donde $a_z \in \mathbf{A_z}$ y $\mathbf{A}_z$ es el conjunto $D_p \backslash \{a_1, a_2, a_3, \ldots, a_{z-1}\}$. Es obvio que los complejos $[x_p = a_1], [x_p = a_2], \ldots, [x_p = \mathbf{A_z}]$, parten a ambos conjuntos de eventos, E y al espacio de eventos $\mathbf{E}$, en z subconjuntos no vacíos. Supongamos que estos complejos particionan a E en $E_{a_1}, E_{a_2}, \ldots, E_{\mathbf{A}_z}$ y a $\mathbf{E}$ en $\mathbf{E}_{a_1}, \mathbf{E}_{a_1}, \ldots, \mathbf{E}_{\mathbf{A}_z}$, donde $E_{a_i} \subseteq \mathbf{E_{a_i}}$.

La variable x_p se asigna a la raíz del árbol. En las ramas que viene de la raíz se asignan los valores $a_1, a_2, \ldots, A_z$. En cada nivel del árbol le corresponden los complejos $[x_p = a_1], [x_p = a_2], \ldots, [x_p = \mathbf{A_z}]$, que cubren al conjunto de eventos $E_{a_1}, E_{a_2}, \ldots, E_{\mathbf{A}_z}$, respectivamente.

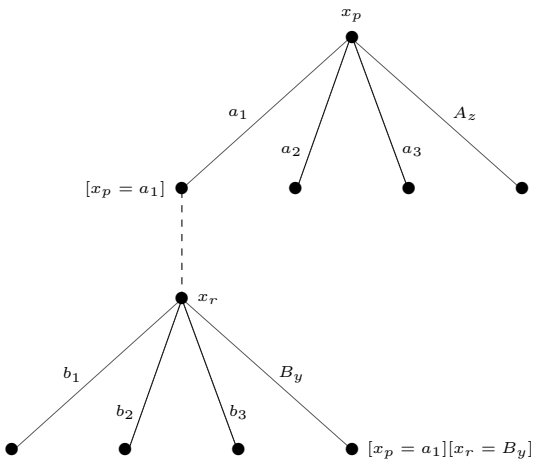


Fig. 8 Una ilustración de la desmostración.

Para cada conjunto de eventos que tenga más de un elemento en el anterior paso se repetirá el proceso con la siguiente modificación. Supongase que E_{a_1} tiene más de un elemento, entonces x_r toma valores $b_1, b_2, \ldots, \mathbf{B_y}$ para los eventos en E_{a_1}. Asigna x_r, a la raíz del nuevo árbol, y colocamos en el árbol la correspondiente hoja de E_{a_1} (i.e., a la hoja hecha por $[x_p = a_1]$ en la figura.), asignando a las ramas de los valores de la raíz $b_1, b_2, \ldots, \mathbf{B}_y$, donde $\mathbf{B}_y = D_i \backslash \{b_1, b_2, \ldots, b_{y-1}\}$. Es obvio que los complejos

$$[x_p = a_1][x_r = b_1], \quad [x_p = a_1][x_r = b_2], \quad \ldots, [x_p = a_1][x_r = B_y]$$

particionan a ambos conjuntos E_{a_1} y a $\sum_{a_1}$ en subconjuntos disjuntos.

Este proceso continua hasta que los niveles obtenidos del árbol correspondan a los complejos y cada uno cubre a solo uno evento de E. Ya que en cada paso de éste proceso se parte simultámente a los eventos en E y en $\mathbf{E}$. Así, estos complejos constituyen el conjunto $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$. □

El teorema anterior dice que el espacio de todos los complejos es suficiente para formar un espacio representado con agrupamientos, es decir que para cualquier conjunto de eventos puede ser agrupado en un número arbitrario de complejos.

Ejemplo.- Para ilustrar el teorema 2.1.18, el conjunto

$$E = \{e_3, e_4, e_7, e_{15}, e_{16}, e_{21}\}$$

	x_1	x_2	x_3
e_3	(0	0	2)
e_4	(0	1	0)
e_7	(0	2	0)
e_{15}	(1	1	2)
e_{16}	(1	2	0)
e_{21}	(2	2	2)

el objetivo del teorema es particionar a E en complejos tales que cada uno cubran a un evento en E, al hacer esto al mismo tiempo el espacio será particionado en estos complejos. Siguiendo el teorema fijamos la variable x_2 como la raíz de muestro árbol la cual toma distintos valores de los eventos en E. Esta variable nos particiona al conjunto inicial en $E_{a_1} = \{e_3, e_{21}\}$, $E_{a_2} = \{e_4, e_{15}\}$ y $E_{a_3} = \{e_7, e_{16}\}$, dado que los subconjuntos obtenidos tiene más de un evento, fijaremos a x_1 como siguiente raíz de nuestros nuevos árboles, ya que es la otra variable que más valores distintos. Los subconjuntos que genera esta variable son $E_{b_1} = \{e_3\}$, $E_{b_2} = \emptyset$, $E_{b_3} = \{e_{21}\}$, $E_{b_4} = \{e_4\}$, $E_{b_5} = \{e_{15}\}$, $E_{b_6} = \emptyset$, $E_{b_7} = \{e_7\}$, $E_{b_8} = \{e_{17}\}$ y $E_{b_9} = \emptyset$. Aunque algunos de los subconjuntos son vacíos los complejos de éste también parcionan al espacio de eventos.

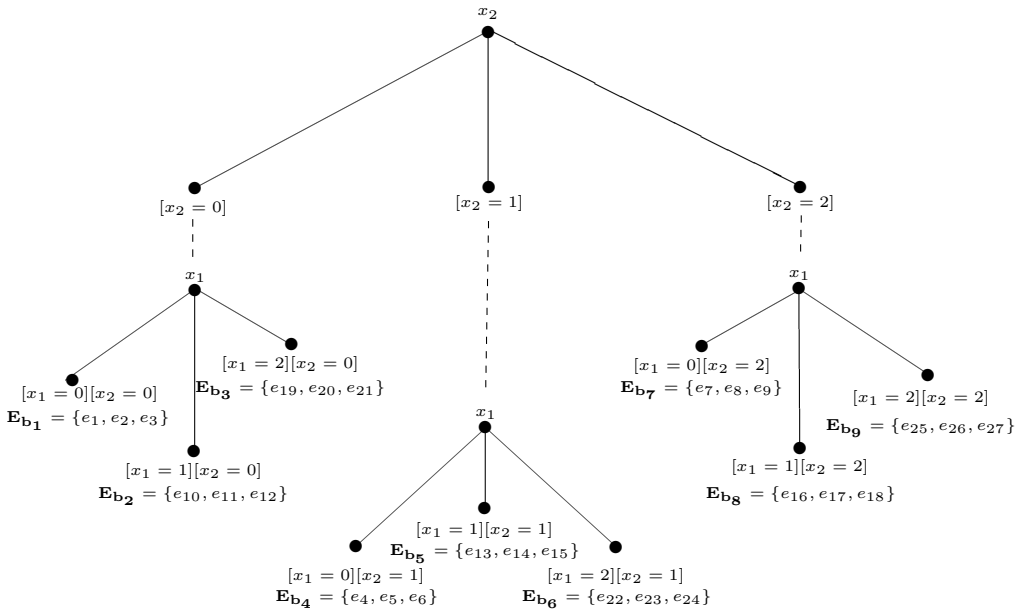


Fig. 9 Estructuración de los 'arboles y partición del espacio **E**.

vemos claramente que $\bigcup\{E_{a_1}\}_{i=1}^{3} \subset \bigcup\{\mathbf{E_{b_1}}\}_{i=1}^{9} = \mathbf{E}$. La forma de los complejo depende de la variable que utilicemos en primera instancia para formar nuestro árbol, por eso siempre es recomendable que esta variable sea la que tome más valores diferentes en E, así los complejos serán fáciles de calcular y sobre todo con la menor cantidad de selectores posible.

La demostración anterior dice que para cualquier conjunto de eventos hay muchas k-particiones. Por lo tanto, surge una pregunta, como seleccionaremos el cubrimiento más atractivo. Para responder estas preguntas , necesitaremos un criterio de calidad para un cubrimiento.

2.1.2 Criterio para evaluar la calidad del agrupamiento.

Sea E el conjunto de eventos, y $COV(E)$ un cubrimiento disjunto de E. Entonces un cubrimiento implica una partición de E en agrupamientos, cada agrupamiento es el conjunto de eventos que contiene un complejo. Las disper-

siones (o grado de generalidad) de un cubrimiento se usan para definir, en una partición, un criterio de calidad. Sin embargo, si E es particionado en eventos individuales, obviamente, la dispersión es cero. Consecuentemente, este criterio puede ser usado solo si el número de agrupamientos es tomado con anterioridad, es decir, para un problema límitado de agrupamientos. En este caso el problema es encontrar un cubrimiento disjunto de E con k complejos, cuyas dispersiones (o grado de generalidad) es mínima. En el caso del problema de un agrupamiento libre (es decir, donde el número de agrupamientos no es tomado con anterioridad), un criterio de calidad particional involucra, además de la dispersión (grado de generalidad), alguna función de costo que depende del número de agrupamientos, por ejemplo, una medida de complejidad de un cubrimiento. En esta sección nos concentraremos solo en el problema de agrupamientos con restricciones. Sin embargo esto no es una limitación seria, pues en la práctica las soluciones interesantes para problemas de agrupamientos no producen sino unos cuantos subconjuntos (pues cuando el número de agrupamientos es grande es preferible organizarlos). Consecuentemente, para obtener una solución general, una condición del algoritmo de agrupamiento es repetirlo varias veces para diferentes k, y la mejor partición obtenida es seleccionada como la solución general.

La dispersión (o grado de generalidad) puede ser insuficiente como criterio para seleccionar un cubrimiento. Podemos buscar un cubrimiento que muestre otras propiedades aparte de la dispersión mínima. Con el fin de utilizar varios criterios para la selección de un cubrimiento al mismo tiempo, adoptaremos el funcional de costo léxicografico.

Una función de evaluación léxicografica LEF (*lexicographic Evaluation Funcional*), se define como una partición de dos listas:

$$A = \langle a - lista, \tau - lista \rangle$$

En donde $a - lista = (a_1, a_2, \ldots, a_i)$ es una lista de atributos usados para evaluar un cubrimiento; y $\tau - lista = (\tau_1, \tau_2, \ldots, \tau_i)$ es una lista de "tolerancias" asignados a los atributos a_i respectivamente, $0 \leq \tau_i \leq 1$.

Sea $V_j, j = 1, 2, \ldots$ todas los posibles cubrimientos disjuntos de el conjunto de E. Si V denota uno de los cubrimientos, y $a_i(V_j)$ denota el valor del atributo a_i del cubrimiento V_j. Se dice que el cubrimiento V es óptimo(mínimal) bajo la función A si para todo j

$$A(V) \overset{\tau}{<} A(V_j) \tag{2.24}$$

En donde

$$\begin{array}{c} A(V) = (a_1(V), a_2(V), \ldots, a_l(V)) \\ A(V_j) = (a_1(V_j), a_2(V_j), \ldots, a_l(V_j)), \qquad j = 1, 2, \ldots \end{array}$$

y $\overset{\tau}{<}$ es la relación , llamada *orden léxicografico de tolerancias*, de modo que:

$$a_1(V_i) - a_1(V) > \Upsilon_1$$

$$\begin{array}{lll} o & |a_1(V_i) - a_1(V)| \leq \Upsilon_1 \quad y & a_2(V_i) - a_2(V) > \Upsilon_2 \\ o \ldots & & \\ \vdots & & \\ o \ldots & y & a_l(V_i) - a_l(V) \geq 0 \end{array} \tag{2.25}$$

En donde

$$\begin{array}{l} \Upsilon_i = \tau_i \cdot (a_{i\,\max} - a_{i\,\min}), \qquad i = 1, 2, \ldots, l-1 \\ a_{i\,\max} = \max_j \{a_i(V_j)\} \\ a_{i\,\max} = \min_j \{a_i(V_j)\} \end{array}$$

Note que si $\tau = (0, 0, \ldots, 0)$ dado que $\overset{\tau}{<}$ denota el orden del léxicografico en el sentido usual. En este caso, A se define sólo por $A = \langle a - lista \rangle$.

Para dar una función A, seleccionamos un conjunto de atributos y los ordenamos en la $a - lista$, y los conjuntos de valores de tolerancias en la $\tau - lista$.

La relación $\overset{\tau}{<}$ particiona a todos los cubrimientos en clases equivalentes y en ordenes de clases lineales, en la primera clase contiene uno ó más cubrimientos óptimos, y en la siguiente clase contiene consecutivamente el cubrimiento menos óptimo.

A continuación se presentan algunos criterios que pueden utilizarse para construir una $a - lista$.

- **Dispersión** (o grado de generalidad g) de un cubrimiento. Minimizaremos las dispersiones y formaremos complejos que "encajen" lo más cerca posible al grupo de eventos. Este criterio es parecido al criterio de minimizar las distancias en las convencionales distacias internas en los algoritmos para agrupamiento basado en la distancia.

- **Interserción**, definida como el promedio del grado de interserción (GI) entre dos complejos cualesquiera. El GI entre dos complejos es el número total de selectores que sobran en ambos complejos después de eliminar de cada partición los complejos disjuntos (selectores cuya referencia conjuntiva no se intersectan). Por ejemplo, el grado de interserción entre el complejo

$$[x_2 = 2, 3][x_4 = 3, 5, 7][x_5 = 2 \cdot \cdot 5]$$

y el complejo

$$[x_1 = 3][x_2 = 1][x_4 = 5 \cdot \cdot 12][x_5 = 1]$$

es 3.

La introducción del GI como un criterio de agrupación proviene de la observación de la gente ya que tienden al preferir particiones de objetos en agrupamientos que no sólo defieren de una, sino de varias características. Este criterio es parecido al criterio de maximización de distancias de agrupación.

- **Desequilibrio**, es definido como

$$\frac{1}{k}\sum_{i=1}^{k}\left|\frac{1}{k}\cdot c(E) - c(E \cap \alpha_i)\right| \tag{2.26}$$

En donde $c(E)$ es el tamaño del conjunto de eventos, y $c(E \cap \alpha_i)$ es el número de eventos cubiertos por el complejo α_i (la cardinalidad del núcleo de α_i). El *desequilibrio* mide la variabilidad del tamaño de los agrupamientos.

- **Dimensionaliad** Se define como el número total de las variables diferentes que involucra el cubrimiento de los complejos. La *dimensionalidad* nos dice cuántas variables se utilizan para describir las agrupaciones, y por lo tanto, cuántas variables tienen que ser medidas para clasificar los objetos en estas agrupaciones.

2.2 Procedimiento STAR y NID.

Antes de describir un algoritmo del agrupamiento conceptual, describiremos dos pequeños procedimientos usados en el algoritmo: STAR y NID. El procedimiento STAR genera las estrellas de un conjunto de eventos respecto a otro conjunto de eventos, y el procedimiento NID transforma un cubrimiento no disjunto, cuando sea posible, en un cubrimiento disjunto con el mismo número de complejos.

2.2.1 Proceso STAR

Sea e_0 un evento y α un complejo. La operación $e_0 \vdash \alpha$ (leída como: e_0 extendido por α), se define como:

$$e_0 \vdash \alpha = \begin{cases} \alpha, & \text{si} \quad e_0 \in \alpha \\ \emptyset, & \text{en otro caso} \end{cases} \tag{2.27}$$

Sea $e_1 = (r_1, r_2, \ldots, r_n)$ y $e_1 \neq e_0$. La operación $e_0 \dashv e_1$ (leída como: e_0 extendido por e_1) es definido como:

$$e_0 \dashv e_1 = \bigcap_{i \in I} (e_0 \vdash [x_i \neq r_i]) \tag{2.28}$$

Denotemos por $G^u(e|E)$ a la unión de los complejos de la estrella $G(e|E)$. Se puede probar que

$$G^u(e|E) = \bigcap_{e_j \in E} (e \dashv e_j) \tag{2.29}$$

Para obtener la estrella $G(e|E)$ de $G^u(e|E)$, el lado derecho de la igualdad, debe convertirse a la unión de complejos maximales (bajo la inclusión). La unión se obtiene al realizar la multiplicación teórica de un conjunto mediante las leyes de absorción.

Esta parte explica una forma más practica para construir las estrellas, que posteriormente serán utilizadas para realizar la mejor búsqueda para determinar la partición k óptima o sub-óptima de un conjunto de eventos. Supongamos en primer lugar, que $F = \{e_1\}$, con $e_1 \neq e$. Para generar la estrella $G(e|\, e_1)$

se determinan todas las variables de e que toman valores diferentes a las de e_1. Supongamos, sin perder generalidad, que todas las variables son nominales y que estas son $x_1, x_2, \ldots, x_k$, y el evento $e_1 = (r_1, r_2, \ldots, r_k, \ldots r_n)$. Bajo estas condiciones los complejos de generalidad máxima que forman la estrella $G(e|e_1)$ son $[x_i \neq r_i], i = 1, 2, \ldots, k$ $([x_i = D_i \setminus r_i])$, ya que estos son complejos más grandes que cubren a e y no cubren e_1. El número de complejos en una estrella $G(e|F)$, cuando F es un evento único, es de un máximo de n (el número de variables), y al menos 1, ya que $e_1 \neq e$.

Supongamos ahora que $F = (e_1, e_2 \ldots, e_s)$. La estrella completa $G(e|F)$ se obtiene mediante la construcción de las estrellas $G(e|\, e_i)$, $i = 1, 2, \ldots s$, posteriormente se usan las leyes de absorción de para eliminar la redundancia.

2.2.2 Proceso NID

Se usa para transformar a un cubrimiento no disjunto en un cubrimiento disjunto. Si $\{\alpha_1, \alpha_2, \ldots, \alpha_l\}$ es un conjunto de complejos no necesariamente disjuntos, y cubren a un conjunto de eventos F. Los pasos para el procesos son los siguientes:

Paso 1.- Si $c(\alpha_i), i = 1, 2, \ldots, l$, denota la cardinalidad de α, (el número total de eventos cubiertos). Determinamos la suma de cardinalidades (aritmética) como:

$$sc = \sum_{i=1}^{l} c(\alpha_i) \tag{2.30}$$

y la cardinalidad de la suma (teórica) de los complejos:

$$cs = c\left(\bigcup_{i=1}^{l} \alpha_i\right) \tag{2.31}$$

Paso 2.- Si sc = cs entonces nos detenemos, L ya es un cubrimiento disjunto.

Paso 3.- Para $i = 1, 2, \ldots, l$, determinamos el núcleo relativo, $NUCLEO_i$, del complejo α_i, es decir, el conjunto que contiene eventos cubiertos por el complejo α_i, y solo por este complejo. Si RESIDUO denota el conjunto de eventos restantes, es decir:$RESIDUO = F \setminus \bigcup_{i=1}^{l} NUCLEO_i$

Paso 4.- Para cada $NUCLEO_i$, determinamos los $mc - complejos$

$$\alpha_i^0 = RU(NUCLEO_i), \qquad i = 1, 2, \ldots, l$$

Paso 5.- Si cualesquiera complejos α_i^0, son intersectan, entonces STOP nos detenemos. El cubrimiento disjunto no puede obtenerse. (Este es una consecuencia directa del teorema del 2.1.10).

Paso 6.- Seleccionamos un eventos del RESIDUO que llamaremos e, y eliminaremos del RESIDUO.

Paso 7.- Para cada pareja $(e, \alpha_i^0), i = 1, 2, \ldots, l$, determinaremos el complejo:

$$\alpha_i^1 = RU(\{e\} \cup \alpha_i^0)$$

Paso 8.- Eliminamos las α_i^1 que se intersecta con cualquier otra $\alpha_j^0, j \neq i$. Si todas las α_i^1 son eliminadas entonces STOP nos detenemos: un cubrimiento disjunto no puede ser obtenido.

Paso 9.- Seleccionamos el mejor complejo, $Mejor - \alpha$, entre las restantes α_i^1, de acuerdo con el LEF:

$$\langle(\Delta spars, -res, -\Delta sel)(\tau_1, \tau_2, \tau_3)\rangle$$

En donde

$\Delta spars$ = La diferencia entre las dispersiones de α_i^1 y α_i^0.
res = El número de eventos en el RESIDUO cubiertos por α_i^1
Δsel = La diferencia entre el número de selectores en α_i^0 y α_i^1
τ_1, τ_2, τ_3 = Las tolerancias del conjunto 0 por defecto(default).

El signo "-" en res y en Δsel indica que el algoritmo maximizará este criterio (minimizando en valor negativo).

Paso 10.- Suponiendo que $Mejor - \alpha$ fue creado juntando e con α_b^0. Tomando ahora a α_b^0 como el $Mejor - \alpha$.

Paso 11.- Si el $RESIDUO = \emptyset$, entonces END finalizamos, en caso contrario iremos al Paso 6.

Con frecuencia ocurre que la cobertura con un mínimo de dispersión total no resulta disjunta. El procedimiento llamado NID trata de transformar un cubrimiento no disjunto en uno disjunto haciendo un pequeño ajuste en cada uno de los agrupamientos. Este procedimiento NID no siempre es éxitoso, sin embargo cuando no se puede crear un cubrimiento disjunto de todos los eventos en F, se crea un cubrimiento disjunto que pueda cubrir el mayor número de eventos en F como sea posible. Los eventos que no están en condiciones de cubrir son reportados como "eventos excepcionales".

El resultado del procedimiento es encontrar un cubrimiento disjunto ó $\{\alpha_1^0, \alpha_2^0, \ldots, \alpha_l^0\}$ del conjunto F, una indicación de que no puede obtenerse un cubrimiento disjunto de los cubrimientos iniciales $\{\alpha_1, \alpha_2, \ldots, \alpha_l\}$.

Ejemplo.- Para ilustrar el proceso NID usaremos un espacio de eventos $\mathbf{E}(3,3,3,3)$. Si el conjunto de eventos F mostrados en la figura 10, tiene a los siguientes complejos no disjuntos que lo cubren, para lo cual aplicaremos NID.

$$\alpha_1 = [x_1 = 1][x_3 = 1] \qquad \alpha_4 = [x_1 = 1][x_2 = 1]$$
$$\alpha_2 = [x_3 = 2][x_4 = 1] \qquad \alpha_5 = [x_1 = 0][x_3 = 2][x_4 = 0, 1]$$
$$\alpha_3 = [x_3 = 1][x_4 = 0, 2]$$

$x_1\,x_2$										
	2				e_{76}					
2	1						e_{69}		e_{71}	
	0				e_{58}		e_{60}		e_{62}	
	2					e_{50}	e_{51}		e_{53}	
1	1	e_{37}		e_{39}	e_{40}		e_{42}	e_{43}	e_{44}	e_{45}
	0				e_{31}	e_{32}	e_{33}		e_{35}	
	2						e_{24}	e_{25}	e_{26}	
0	1				e_{13}			e_{16}		
	0				e_4		e_6		e_8	
		0	1	2	0	1	2	0	1	2
		0			1			2		

x_4 / x_3

Fig. 10 Distribución del conjunto F para el espacio $\mathbf{E}(3, 3, 3, 3)$.

Paso 1.- La suma de las cardinalidades de los complejos es:

$$sc = \sum_{i=1}^{5} c(\alpha_i) = 7 + 7 + 13 + 7 + 3 = 37$$

y

$$cs = c\left(\bigcup_{i=1}^{5} \alpha_i\right) = 28$$

como $sc \neq cs$ ir al paso 3.

Paso 3.- Se determinan los núcleos relativos de cada complejo para obtener el RESIDUO

$$\begin{array}{ccc} Complejo & & NUCLEO \\ \alpha_1 & \Longrightarrow & \{e_{50}, e_{32}\} \\ \alpha_2 & \Longrightarrow & \{e_8, e_{26}, e_{35}, e_{53}, e_{62}, e_{71}\} \\ \alpha_3 & \Longrightarrow & \{e_4, e_6, e_{13}, e_{24}, e_{58}, e_{60}, e_{69}, e_{76}\} \\ \alpha_4 & \Longrightarrow & \{e_{37}, e_{39}, e_{43}, e_{45}\} \\ \alpha_5 & \Longrightarrow & \{e_{16}, e_{25}\} \end{array}$$

$$RESIDUO = F \backslash \bigcup_{i=1}^{5} NUCLEO_i = \{e_{31}, e_{33}, e_{40}, e_{42}, e_{44}, e_{51}\}$$

Paso 4.- Para cada $NUCLEO_i$ calculamos $mc - complejo$

$$\begin{array}{l} \alpha_1^0 = RU(NUCLEO_1) = [x_1 = 1][x_2 = 0, 2] \\ \alpha_2^0 = RU(NUCLEO_2) = [x_3 = 2][x_4 = 1] \\ \alpha_3^0 = RU(NUCLEO_3) = [x_1 = 0, 2][x_3 = 1][x_4 = 0, 2] \\ \alpha_4^0 = RU(NUCLEO_4) = [x_1 = 1][x_2 = 1][x_3 = 0, 2][x_4 = 0, 2] \\ \alpha_5^0 = RU(NUCLEO_5) = [x_1 = 0][x_2 = 1][x_3 = 2][x_4 = 0] \end{array}$$

Paso 5.- Es obvio que los complejos no se intersectan, hemos obtenido un cubrimiento disjunto, iremos al Paso 6 y 7 donde uniremos los eventos del RESIDUO a los complejos obtenidos de tal forma que deben seguir siendo disjuntos.

Paso 6.- Seleccionamos un evento del RESIDUO, tomaremos e_{31} que llamaremos sólo e.

Paso 7.- Calcularemos para todos los complejos ya obtenidos la refunión del evento seleccionado en el paso 6.

$$\alpha_1^1 = RU(\alpha_1^0 \cup e) = [x_1 = 1][x_2 = 0, 2][x_3 = 1][x_4 = 0, 1]$$
$$\alpha_2^1 = RU(\alpha_2^0 \cup e) = [x_3 = 1, 2][x_4 = 0, 1]$$
$$\alpha_3^1 = RU(\alpha_3^0 \cup e) = [x_3 = 1][x_4 = 0, 2]$$
$$\alpha_4^1 = RU(\alpha_4^0 \cup e) = [x_1 = 1][x_2 = 0, 1][x_4 = 0, 2]$$
$$\alpha_5^1 = RU(\alpha_5^0 \cup e) = [x_1 = 0, 1][x_3 = 1, 2][x_4 = 0]$$

Paso 8.- Las α_i^1 resultantes en la eliminación de la interseción con α_i^1 son: α_1^1 y α_3^1.

Paso 9.- Aplicaremos el criterio de calidad LEF a los complejos, el mejor complejos de estos es α_3^1 que nos reduce al máximo el RESIDUO, tiene como dispersiones 5 y diferencia de selectores 1.

Paso 10.- Nuestro $Mejor - \alpha$ ha sido obtenido.

Paso 11.- El $RESIDUO = \emptyset$, hemos finalizado el proceso.

Sabemos el proceso NID no siempre funciona pero en algunos casos resulta muy eficiente aplicarlo por que nos proporciona mejores complejos disjuntos y es por eso que NID es utilizado por los métodos PAF, P, PS y P descritos posteriormente.

2.2.3 Método PAF e Implementación

La porción interna del algoritmo llamado PAF [1], que Michalski plantea en su artí*"Knowledge acquisition through conceptual clustering"*, introduce una técnica de agrupaciones conceptual como un conjunto limitado. Dado un conjunto de eventos, E, a ser agrupados y un número entero, k, PAF aparte al conjunto E en k grupos, cada uno de los cuales tiene una descripción en forma de complejos. Las particiones obtenidas son óptimas o sub-óptimas con respecto a una medida de la calidad para la agrupación.

La estructura general del PAF se basa en la agrupación dinámica, método desarrollado por Diday y sus colaboradores. En la base de las notaciones de la dinámica de agrupaciones, método que consiste de dos funciones:

[1] Polish-American-french

g) La **función de representación**, dado k-particiones de E, produce una serie de k representantes que describen mejor a los grupos.

f) La **función de asignación**, dado un conjunto k de representantes de grupo, que produce una k-partición en agrupamiento que esta compuesta de esos objetos. La que mejor ajusta a los representantes elegidos.

El método trabaja iterativamente, empezando por elegir de manera aleatoria un conjunto inicial k de representantes para el agrupamiento. Una iteración consiste en la aplicación de la función f para las representaciones dadas, seguido de la función g para obtener la partición. Cada iteración término con un nuevo conjunto de representaciones. El proceso sigue hasta que el criterio de calidad de agrupación deja de mejorar las agrupaciones.

PAF tiene un paso análogo a la función g, y forman un conjunto de objetos para la produción de los mejores representantes (en forma de complejos). El proceso consiste en seleccionar un evento representante (*semillas*) de cada grupo para generar las descripciones generalizadas de eventos y son calculados el procedimiento STAR y NID. La función f de la agrupación dinámica está representada en PAF por un procedimiento que determina el conjunto de los eventos cubiertos.

El algoritmo PAF muestra la forma en como opera sobre 10 eventos que se muestran en la siguiente tabla, con un número de agrupaciones k=2 con un mínimo de dispersion total. Estos eventos son descrito por cuatro valores en las variables x_1, x_2, x_3 y x_4, cuyo dominio, $D_i = (0, 1, 2), i = 1, \ldots, 4$. Las variables x_1 y x_2 son lineales, y x_3 y x_4 son nominales.

Eventos	x_1	x_2	x_3	x_4
e_1	0	0	0	1
e_2	0	1	0	0
e_3	0	2	1	2
e_4	1	0	0	2
e_5	1	2	1	1
e_6	2	0	1	0
e_7	2	1	0	1
e_8	2	1	1	2
e_9	2	2	0	0
e_{10}	2	2	2	2

Usado una representación plana para gráficar el conjunto de los 10 eventos mostrados en la tabla.

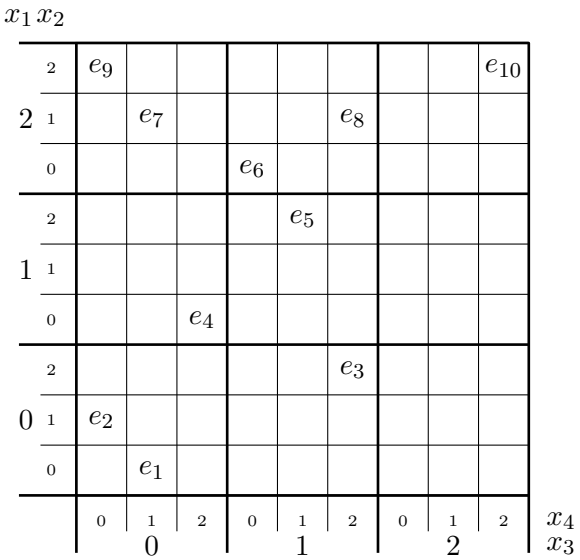


Fig. 11 Distribución de los eventos en el espacio.

Aplicando el diagrama de flujo de PAF mostrado en la figura 4.

ITERACIÓN 1:

Paso 1.- E_0 es un subconjunto compuesto de k eventos (*semillas*) de **E**. La semillas pueden ser seleccionadas de forma arbitraria, o pueden ser elegidos como los eventos que son sintácticamente más distantes de él con respecto a los demás. El último caso la convergencia será más rápida. Por ejemplo, sea $E_0 = (e_1, e_2)$.

Paso 2.- Se calculan las estrellas, $G_j(e_j|E_0 \backslash e_j)$, donde e_j es elemento de E_0, apartir el procedimiento STAR descrito anteriormente:

$G_1(e_1|e_2) = \{[x_2 = 0][x_3 = 0, 1], [x_4 = 1, 2]\}$
$G_2(e_2|e_1) = \{[x_2 = 1, 2], [x_4 = 0, 2]\}$

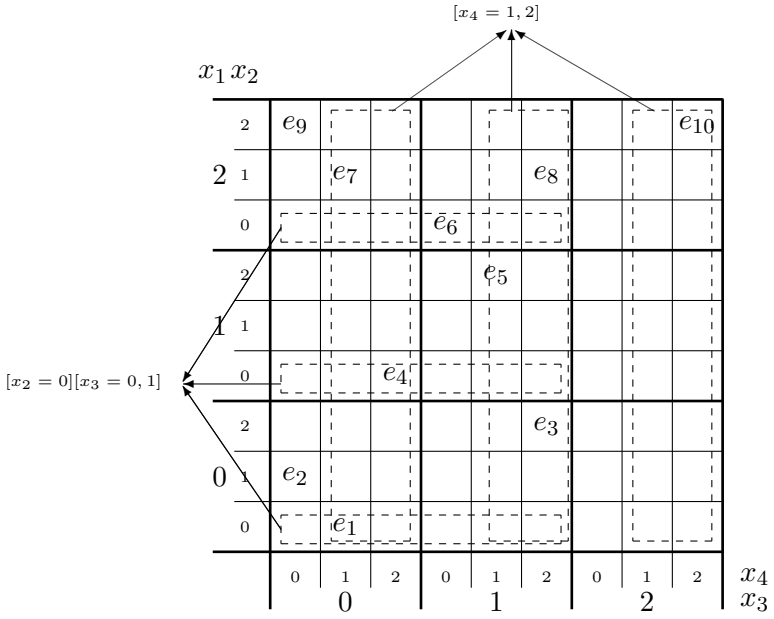


Fig. 13 Ilustración de los complejos en la estralle $G(e_1|e_2)$.

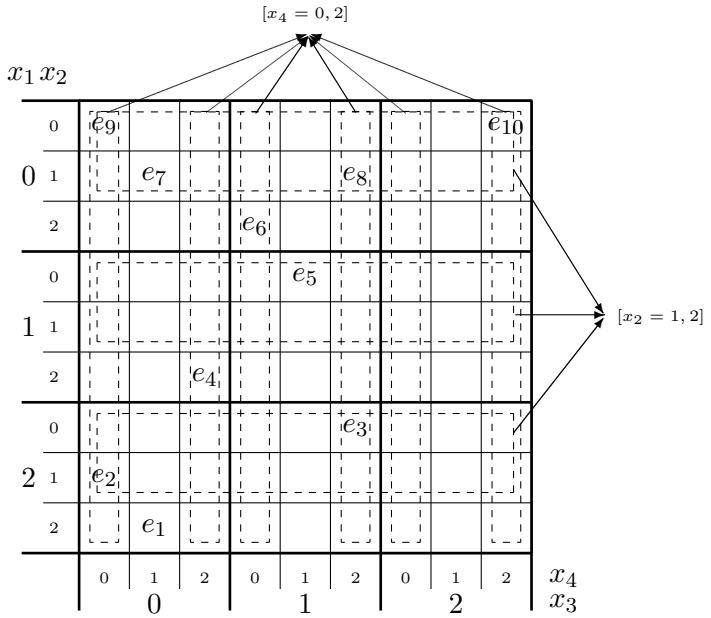


Fig. 14 Ilustración de los complejos en la estralle $G(e_2|e_1)$.

Paso 3.- De cada estrella se selecciona un complejo de tal manera que el conjunto de complejos sea:

a) Cubrimiento disjunto en E, y

b) Cubrimiento óptimo o sub-óptimo de acuerdo al criterio de calidad (LEF) seleccionado (dispersión, intersección, desequilibrio y dimensionalidad). La calidad en este caso es minimizar la dispersión total. Existen cuatro combinaciones de estos complejos:

			Dispersiones
(a)	complejo 1:	$[x_2 = 0][x_3 = 0, 1]$	15
	complejo 2:	$[x_2 = 1, 2]$	47
			62
(b)	complejo 1:	$[x_4 = 1, 2]$	
	complejo 2:	$[x_2 = 1, 2]$	
(c)	complejo 1:	$[x_2 = 0][x_3 = 0, 1]$	
	complejo 2:	$[x_4 = 0, 2]$	
(d)	complejo 1:	$[x_4 = 1, 2]$	
	complejo 2:	$[x_4 = 0, 2]$	

Los complejos (b), (c) y (d), no son disjuntos. Combinación (a) es complejo disjuntos, y tienen mínima dispersión total.

Paso 4.- El criterio de terminación del algoritmo se aplica hasta que se obtenga una cubierta. Este criterio tiene como parámetros (b, p), donde b (*base*) es el número de iteraciones el algoritmo que se realizan, y p (*sonda*) es el número de iteraciones adicionales realizadas más allá por b, en cada iteración se produce una cubierta mejor. En nuestro ejemplo, $b = 2$ y $p = 1$.

Paso 5.- Si la iteración es impar, entonces las nuevas semillas son eventos centrales cubiertos por los mismo complejos (de acuerdo con la distancia sintáctica). Si la iteración es par, entonces las nuevas semillas son eventos alejados al máximo de los centros (de acuerdo con el principio de la adversidad [2]).

[2]Este principio establece que si realmente los eventos más destacados pertenecen a la agrupación determinada, cuando éste actúa como representante de esta, el "*encaje*" entre

En nuestro ejemplo los eventos sintácticamente centrales para los complejos $[x_2 = 1, 2]$ y $[x_2 = 0, 1]$ que cubre a los eventos $(e_2, e_3, e_5, e_7, e_8, e_9, e_{10})$ y a los eventos (e_1, e_4, e_6) respectivamente, son aquellos en los cuales la suma de las distancias sintácticas entre ellos y los demás eventos, del mismo grupo, sea la mínima. En la siguiente tabla se muestra la suma de estas distancias.

Complejo $[x_2 = 1, 2]$		Complejo $[x_2 = 0][x_3 = 0, 1]$	
Eventos	Distancias	Eventos	Distancias
e_2	18	e_1	5
e_3	16	e_4	**5**
e_5	18	e_6	6
e_7	16		
e_8	**15**		
e_9	15		
e_{10}	16		

El nuevo conjunto de semillas es $E_0 = \{e_4, e_8\}$.

Iteración 2:

Paso 2.- La construción de las estrellas $G_1(e_4|e_8)$ y $G_2(e_8|e_4)$ generan los siguientes complejos:

$$G_1(e_4|e_8) = \{[x_2 = 0][x_3 = 0, 1], [x_1 = 0, 1][x_3 = 0, 1], [x_3 = 0, 2]\}$$
$$G_2(e_8|e_4) = \{[x_1 = 2], [x_2 = 1, 2], [x_3 = 1, 2]\}$$

Paso 3.- La combinación de los complejos que son cubrimientos disjuntos son las siguientes:

éste y otros eventos del mismo grupo deberá ser aun mejor que el "*encaje*" entre los eventos de cualquier otro grupo.

			Dispersiones
(a)	complejo 1:	$[x_1 = 2]$	22
	complejo 2:	$[x_1 = 0, 1][x_3 = 0, 1]$	31
			53
(a)	complejo 1:	$[x_2 = 1, 2]$	47
	complejo 2:	$[x_2 = 0][x_3 = 0, 1]$	15
			62

Seleccionamos (a) con un mínimo de distancia total.

Paso 4.- Siendo que es la segunda iteración (b = 2), es la última de las iteraciones base.

Paso 5.- De acuerdo con el paso 3 los complejos obtenidos son: $[x_1 = 2]$ que cubre a los eventos $(e_6, e_7, e_8, e_9, e_{10})$ y el complejo $[x_1 = 0, 1][x_3 = 0, 1]$ que cubre a $(e_1, e_2, e_3, e_4, e_5)$. Dado que esta es una iteración par, las nuevas semillas serán aquellas cuya suma de las distancias sintácticas respecto a otros eventos del mismo grupo es la máxima. Los son eventos $E_0 = (e_2, e_{10})$.

Iteración 3:

Paso 2.- Las estrellas $G_1(e_2|e_{10})$ y $G_2(e_{10}|e_2)$ generan los complejos:
$G(e_2|e_{10}) = \{[x_1 = 0, 1][x_3 = 0, 1], [x_2 = 0, 1][x_3 = 0, 1], [x_3 = 0, 1][x_4 = 0, 1]\}$
$G(e_{10}|e_2) = \{[x_1 = 1, 2], [x_2 = 2], [x_4 = 1, 2]\}$

Paso 3.- La única combinación de complejos con cubrimientos disjunto es la siguiente:

			Dispersiones
(a)	complejo 1:	$[x_2 = 2]$	23
	complejo 2:	$[x_2 = 0, 1][x_3 = 0, 1]$	30
			53

Paso 4.- Esta es la primera iteración p (y también es la última). Si la calidad de agrupación en estaiteración es mejor que la mejor agrupación anterior, se programaran otras p iteraciones, de lo contrario el algoritmo se detiene después

de satisfacer las p iteraciones. Si la dispersión total mínina en la iteración 3, es decir la dispersión de 53 eventos, no es mejor que la anterior dispersión total mínina, que también es 53. Siendo en el ejemplo $p = 1$, el criterio de terminación se cumple en este paso. Por lo que existen dos alternativas de agrupación, cada una con un total dispersiones de 53 eventos:

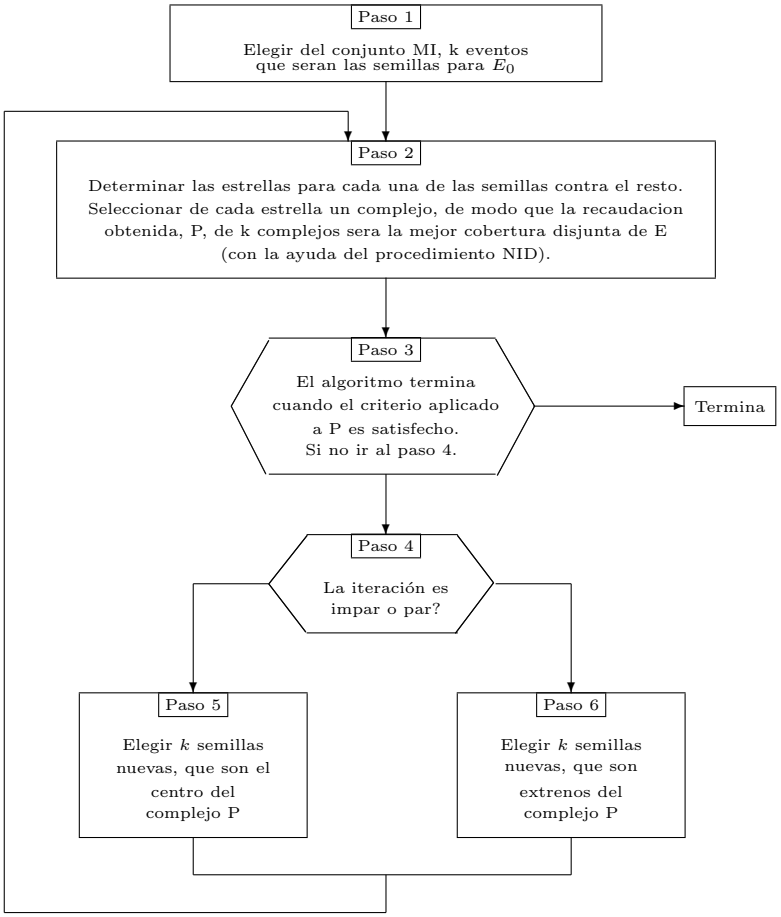


Fig. 15 Diagrama del Procedimiento PAF.

Alternativa 1:	Alternativa 2:
$[x_1 = 2]$	$[x_2 = 2]$
$[x_1 = 0, 1][x_3 = 0, 1]$	$[x_2 = 0, 1][x_3 = 0, 1]$

Con este método se obtiene una estructuración del universo dado en k agrupamientos, de modo tal que las variables que más inciden en la formación de los mismos desempeñarán el papel determinante en los selectores de los complejos. Además, cada agrupamiento viene carácterizado por un conjunto de propiedades relativas a los valores de las variables que describen a los objetos por lo que además de las medidas de similaridad utilizadas se pueden dar **conceptos** que describan de algún modo los agrupamientos obtenidos.

2.3 Método P, PS y S

La cota superior del tamaño de una estrella es n^m, donde m es el número de eventos en F. Normalmente en las leyes de absorción se elimina muchos complejos redundantes, pero el tamaño de una estrella aún puede ser inmanejable. Por lo tanto, una estrella delimitada es la que ha utilizado un parámetro especial, MAXSTAR, que es una cota superior del número de los complejos que ésta pueda contener. La razón es que el tamaño de las estrellas puede ser muy grande cuando el número de variables n es grande. Cada vez que una estrella es superior a esta cifra, los complejos son ordenados de forma ascendente de acuerdo a la dispersión (para cualquier criterio supuesto) y sólo el primero de los complejos se mantiene. La posición de un complejo, en la secuencia ordenada, es el rango del complejo.

En cada iteración del algoritmo PAF, se producen k estrellas, para cada semilla en contra con las restantes $k-1$ semillas. De cada estrella se selecciona un complejo de manera que el conjunto resultante consistirá de k complejos disjuntos (una k-partición), y es óptimo de acuerdo con el criterio asumido. Si se utilizaron estrellas sin acotar, cada una de ellas podría estar compuesta de hasta $n^{(k-1)}$ complejos y, por tanto, hasta $(n^{(k-1)})^k$ conjuntos de complejos por lo que tendría que ser objeto de inspección con el fin de determinar la k-partición óptima.

Se supone que todos los complejos en la estrella son reducidos (es decir, la operación refunión se aplica al núcleo de cada complejo y el $mc - complejo$ resultante se utiliza para sustituir al complejo original en la estrella, véase Definición 2.1.11). Para simplificar la descripción del método vamos a suponer que el criterio para óptimizar los agrupamientos, es reducir las dispersiones al mínimo cubrimiento disjunto representan una partición. Los métodos pueden

ser extendidos a un múlticriterio usando el criterio LEF (que implementa un orden lineal entre las clases de equivalencia de conjuntos de complejos). En el múlticriterio, las dispersiones deben utilizarse como criterio principal con el fin de conservar las propiedades de los métodos descritos.

La mejor estrategia para búsqueda se describe a continuación en los métodos P (paralelo), PS (Paralelo-Secuencial) y S (Secuencial). En estos se realiza una búsqueda eficiente del mejor conjunto de complejos el cual se puede ver como un camino óptimo a través de la búsqueda del árbol cuyos nodos son los complejos. Donde el *i-ésimo* nivel del árbol corresponde a la *i-ésima* estrella. La altura del árbol es de k, y el largo del camino k corresponde a una especial k-partición. El rango de la trayectoria (*pathrank*) de un camino es la suma del rangos de los complejos a lo largo del camino. Las secuencias de complejos (caminos) son considerados en orden ascendente para que el rango de las trayectoria forzen a los complejos con un mínimo de dispersión sea considerados en primer lugar.

2.3.1 Método P

El método P se aplica para valores relativamente pequeños de MAXSTAR y k. Esto es útil cuando se hace un procedimiento de datos en paralelo. Supongamos que cada estrella $G_i = G(e_i|(E_0 \backslash \{e_i\}))$ es formada por el conjunto $\{\alpha_0^i, \alpha_1^i, \ldots, \alpha_k^i\}$. Supongamos que los complejos $\alpha_j^i, j = 0, 1, \ldots, g_i$ están ordenados de forma ascendente de acuerdo con sus dispersiones. La posición de un complejo en la estrella ordenada (indicado por un subíndice, que empieza de 0) se llama el *rango* del complejo (por ejemplo, el complejo α_2^1 tiene rango 2). Para cada α_j^i en la estrella $G_i, i = 1, 2, \ldots, k$, se puede generar las sucesiones:

$$\begin{aligned} P_0 &= (\alpha_0^1, \alpha_0^2, \ldots, \alpha_0^k) \\ P_1 &= (\alpha_0^1, \alpha_0^2, \ldots, \alpha_0^{k-1}, \alpha_1^k) \\ &\vdots \\ P_{gk} &= (\alpha_0^1, \alpha_0^2, \ldots, \alpha_{gk}^k) \\ &\vdots \\ P_\Gamma &= (\alpha_{g1}^1, \alpha_{g2}^2, \ldots, \alpha_{gk}^k) \end{aligned}$$

En donde $\Gamma = (g_1 + 1)(g_2 + 1) \cdots (g_k + 1)$

La suma de rangos de los complejos en cualquiera de estas sucesiones es llamado el rango de la trayectoria (*rankpath*). Se supone que las sucesión $P_j, j = 1, 2, \ldots, \Gamma$ son ordenados en orden ascendente de acuerdo a los rangos de trayectoria (*rankpath*). Por lo tanto, P_0 tiene rango de trayectoria (*rankpath*) igual a 0 (porque todos los complejos en P_0 tienen rango 0); $P_2, P_3, \ldots, P_{k+1}$ tienen rango de trayectoria (*rankpath*) igual a 1, y P_Γ tiene rango de trayectoria $g_1 + g_2 + \cdots + g_k$. El orden de las sucesiones con el mismo rango de trayectoria (*rankpath*) es irrelevante.

Teniendo en cuenta las sucesiones P_j en orden ascendente de acuerdo a los rangos de trayectoria, las operaciones siguientes se realizan para cada sucesión:

Paso 1.- Se prueba si P_j es un cubrimiento de E. Esto se hace de forma consecutiva eliminando de E eventos cubiertos para cada complejo en P_j. Si al fínal E llega a ser el conjunto vacío, P_j es un cubrimiento. Si P_i no es un cubrimiento, ya no se tomará en cuenta en futuras consideraciones.

Paso 2.- Se prueba si P_j es un cubrimiento disjunto. Si es así, se calculan sus dispersiones. Si no es así, se calcula una cota inferior (C.I.) para las dispersiones de un cubrimiento disjunto posible (sin llegar a determinar el cubrimiento disjunto). El C.I. se calcula para determinar el núcleo relativo de cada complejo (es decir, los eventos que son cubiertos sólo por el complejo dado y por algún otro complejo) y entonces se calculan las dispersiones del mc-complejo del núcleo. El C.I. es la suma de las dispersiones obtenidas (este cálculo se basa en el teorema 2.1.13). (el fin de utilizar el C.I. es de evitar, siempre que sea posible, el método computacionalmente costoso NID).

Paso 3.- Si el cálculo de las dispersiones (C.I.) no es un nuevo mínimo (es decir, no es menor que el dispersiones del mejor cubrimiento obtenido hasta el momento), entonces el cubrimiento no se toma en cuenta en futuras consideraciones. Ahora, si se trata de un cubrimiento disjunto, se mantiene como el mejor cubrimiento, y si se trata de un cubrimiento no disjunto, será transformado mediante el procedimiento NID, si es posible, a un cubrimiento disjunto (teniendo en cuenta que algunas operaciones del procedimiento NID ya se han hecho en el paso 2). Si las dispersiones del cubrimiento disjunto obtenido todavía representa un nuevo mínimo, el cubrimiento se mantiene como la mejor hasta el momento. Si las dispersiones no es un nuevo mínimo, o el procedimiento NID falló al tratar de encontrar un cubrimiento disjunto, el cubrimiento no se toma en cuenta en futuras consideraciones.

El cubrimiento disjunto obtenido al final del proceso de búsqueda a través de sucesiones P_j es el resultado de nuestro método. Se trata de un cubrimiento con dispersiones mínimas que pueden unirse a los complejos de las estrellas dadas. La existencia de al menos un cubrimiento disjunto está garantizada por el principio de suficiencia. Una de las ventajas del orden de sucesiones de P_j es que el cubrimiento más prometedor es el que se acerca al comienzo de la lista. Por lo tanto, si el número de sucesiones es muy grande, la búsqueda puede parar antes de llegar a al final, con un bajo riesgo de perder la solución óptima.

2.3.2 Método PS

En el método P, todas las sucesiones P_j se generan primero, y luego se hace una búsqueda lineal ordenada con el fin de determinar el mejor cubrimiento. En este método, la búsqueda del mejor cubrimiento se realiza durante el proceso de generación de sucesiones, utilizando la "primer mejor" estrategia de búsqueda (ver Winston [16]). En concreto, la búsqueda se basa en el algoritmo A^* (véase Nilsson [11]). En cada paso un complejo, se añade al cubrimiento parcial (una sucesión parcial después de la aplicación de NID), que lleva al cubrimiento óptimo más prometedor(de acuerdo a una función de evaluación). Este proceso evita las pruebas (generalmente muchas) de las sucesiones de P_j, para lo cuales es posible predecir que no van a producir un cubrimiento óptimo. El método PS es especialmente aplicable cuando las estrellas G_i son grandes.

En la Figura 16 se ilustra el proceso de los búsqueda. Las ramas de árboles en el nivel i representa a los complejos en la estrella G_i. La trayectoria de la raíz a un nodo en el nivel i representa parcialmente un cubrimiento disjunto con i complejos. Cuando $i = k$, la trayectoria representa un cubrimiento completo disjunto (correspondientes a algunos sucesiones P_j para el que NID fue aplicado).

Paso 1.- Se genera la sucesión $P_0 = (\alpha_1^0, \alpha_2^0, \ldots, \alpha_k^0)$ (es la sucesión de los complejos con dispersiones más pequeñas). Se determina el núcleo relativo de cada complejo y luego es construido el mc-complejo para cada núcleo. Si $s_1, s_2, \ldots, s_k$ denota la dispersiones de los mc-complejos obtenidos. Con base en el teorema 2.1.13, la suma $s_1 + s_2 + \cdots + s_k$ indica una cota inferior de las

dispersiones del mejor cubrimiento disjunto que puede ser construido a partir de los complejos en las estrellas dadas.

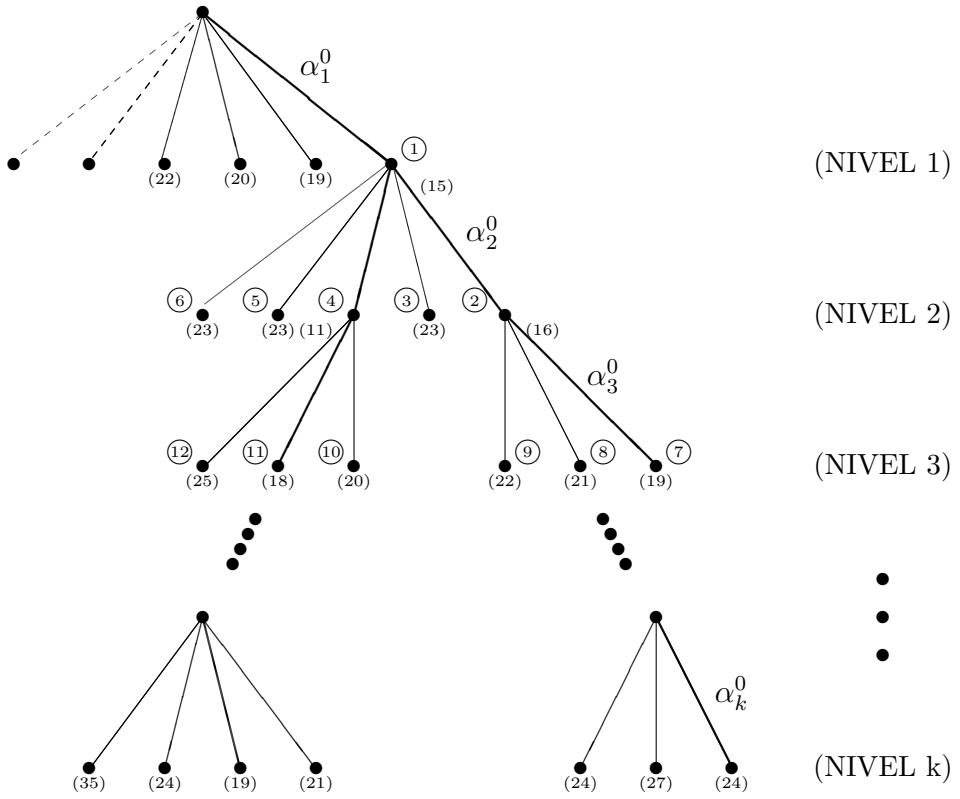


Fig. 16 Ilustración del árbol de busqueda en el proceso PS

Paso 2.- El nodo (1) (fig. 16) se amplía, es decir, α_1^0 es emparejado con cada complejo en G_2, el procedimiento NID se aplica a cada pareja, y las dispersiones se calculan para obtener el par disjunto. Si el procedimiento NID falla, se abandona la trayectoria. El par obtenido es un cubrimiento parcial con $i = 2$ complejos. Los nodos correspondientes para generar cubrimiento parcial (que incluye el resto de los complejos en G_i) son asignados a un valor de la función de evaluación:

$$f = h + g$$

En donde h son la dispersión del cubrimiento parcial disjunto, y g es la suma $s_{i+1} + s_{i+2} + \cdots + s_k$, donde i es el número de complejos en el cubrimiento parcial, {g representa un C.I. son las dispersiones del resto de los complejos a ser determinados, es decir, los complejos que son necesarios para completar la construcción del cubrimiento}.

De acuerdo a la mejor primer estrategia, el nodo debe ampliarse en cada paso y es el primero con el que esta asociado con el menor valor de la función de evaluación. Se ha demostrado que está estrategia produce un cubrimiento óptimo (18). El orden de los nodos expandidos en el árbol de la figura 15 se muestra el número de los círculos. Los valores de la función de evaluación asociados con cada nodo se indica entre paréntesis.

2.3.3 Método S

Este método es parecido al método PS, con la excepción de que las estrellas no son generadas al inicio. Cuando expandamos un nodo en el árbol de búsqueda, en lugar de tomar los complejos de una estrellas determinada, cada estrella se va generando al mismo tiempo. Esto requiere una repetición múltiple del proceso de generación de la estrella, se puede ir guardando en la memoria para almacenar todas las estrellas (que puede ser de grandes).

2.4 Ejemplos

Veremos algunos resultados del método PAF aplicado a 10 eventos mostrados en la figura 17. Los objeto estan descrito por cuatro variables que son: tamaño, número, forma y orientación de eslabón.

x_1 :	Tamaño de eslabón	x_3	Forma de eslabón
	0 - Grande		0 - Circular
	1 - Mediano		1 - Cuadrado
	2 - Chico		2 - Triángulo
x_2 :	Número de eslabón	x_4	Alimeación de eslabón
	0 - 2 Eslabones		0 - Vertical
	1 - 3 Eslabones		1 - Diagonal
	2 - 4 Eslabones		2 - Horizontal

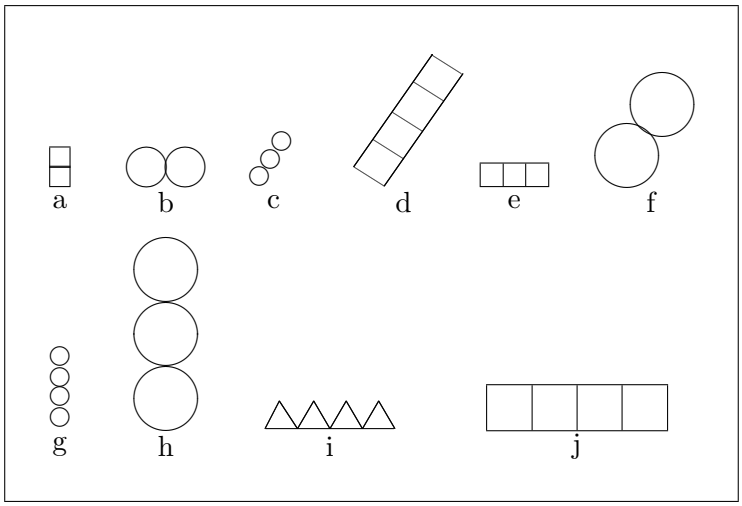


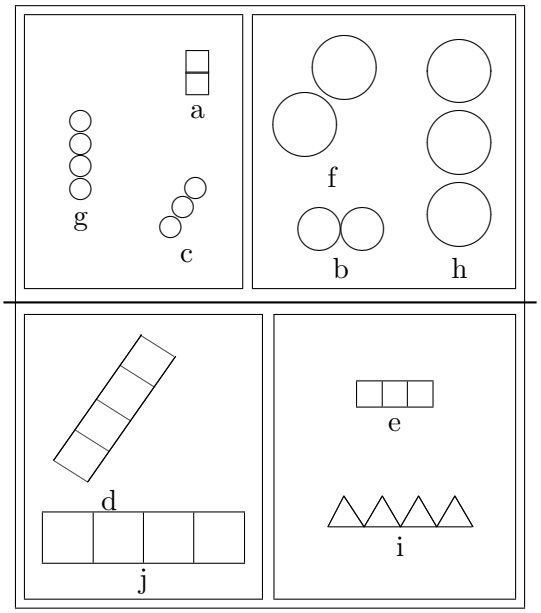
Fig. 17 Conjunto E de eventos.

Los valores de los 10 eventos se muestran en la siguiente tabla:

Símbolo de Cadena	Tamaño de Eslabón	Número de Eslabón	Forma de Eslabón	Orientación de Eslabón
a	chico	2	cuadrada	vertical
b	mediano	2	circular	horizontal
c	chico	3	circular	diagonal
d	mediano	4	cuadrada	diagonal
e	chico	3	cuadrada	horizontal
f	grande	2	circular	diagonal
g	chico	4	circular	vertical
h	grande	3	circular	vertical
i	chico	4	triangular	horizontal
j	grande	4	cuadrada	horizontal

Compararemos los resultados obtenidos del método PAF con otros hechos por un programa llamado NUMTAX que utiliza técnicas tradicionales de agrupación las cuales son descritas en el algoritmo de Sokal & Sneath [18]. Teniendo

un conjunto de eventos, el programa taxonumérico, NUMTAX, organiza los eventos en conjuntos los cuales refleja una distancia numérica entre subcategorias de eventos, donde el nivel más alto viene siendo la unión de todos los eventos. El dendrograma [3] generado por NUMTAX usa una medida de distancia Euclidiana como medida de similitud. Existen 18 diferentes medidas de distancias que fueron probadas en el experimento usando varias combinaciones en transformación de datos. El dendrograma se construye de forma ascendente que parte a los niveles para producir agrupamientos. Los eventos mostrados en la figura 18 son el resultado de dos mejores sub-árboles del dendrograma. Los tres agrupamientos mostrados en la figura 19 fueron obtenidos de manera similar. Las dos ilustraciones son generados a partir de muchos dendrogramas y ninguno de ellos forman agrupamientos que tengan carácteristicas simples.



1. [Forma = circular] u [Orientación = Vertical]

2. [Forma ≠ circular][Orientación ≠ vertical]

Fig. 18 Agrupamiento obtenido a partir del derograma de NUMTAX (k=2)

[3]dendrograma (del griego Dendron "árbol", "Gramma" dibujo ") es un diagrama de árbol utilizado para ilustrar la disposición de las agrupaciones producidas por un algoritmo de agrupamiento.

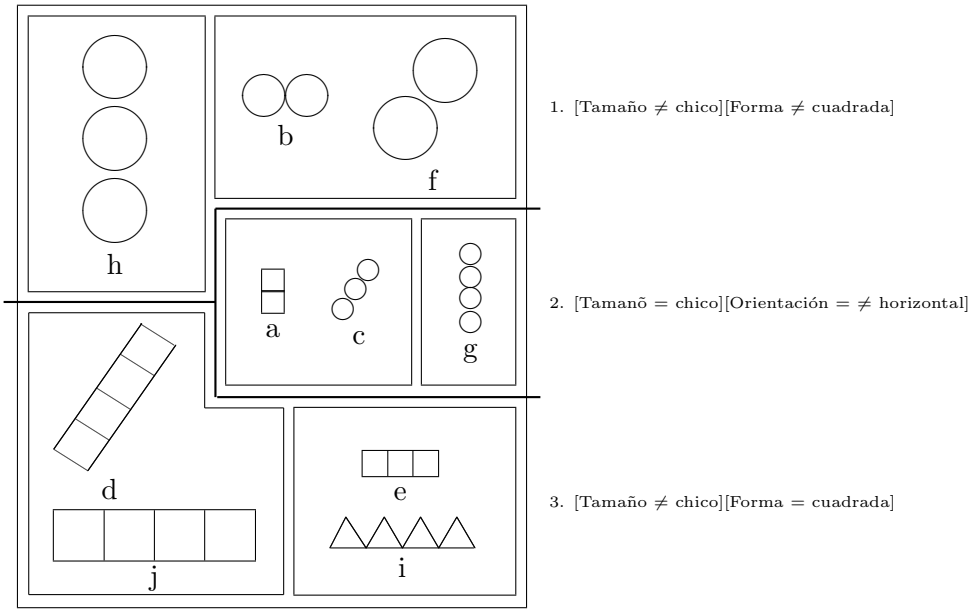


Fig. 19 Agrupamiento obtenido a partir del dendrograma de NUMTAX (k=3).

Ahora veremos los resultados proporcionados por el método PAF utilizando dos diferentes criterios de calidad para la obtención del agrupamiento óptimo:

(a) Dispersiones mínimas, Número de selectores mínimos.
(b) Grado de disjunción mínima, Dispersiones mínimas.

Los resultados de la figura 20 y 21 utilizaron el criterio de calidad (a). Las figuras 22 y 23 muestran los resultados obtenidos del criterio (b). Vemos que las carácteristicas de los agrupamientos son mucho más simples y muestran un sentido en la agrupación que resulta más lógica, muestran que los agrupamientos generados por NUMTAX tiene una forma no muy natural. Cabe mencionar que viendo del punto de vista humano el problema de agrupación de estos 10 objetos es algo muy sencillo, sin embargo las técnicas que existen hasta ahorita no se han aproximado una lógica humana. Se observa en las siguientes figuras resultados que han sido con los que han congeniado con el planteamiento de muchas personas, pero finalmente todo depende para cual sea el objetivo para el que se utilice la organización de datos.

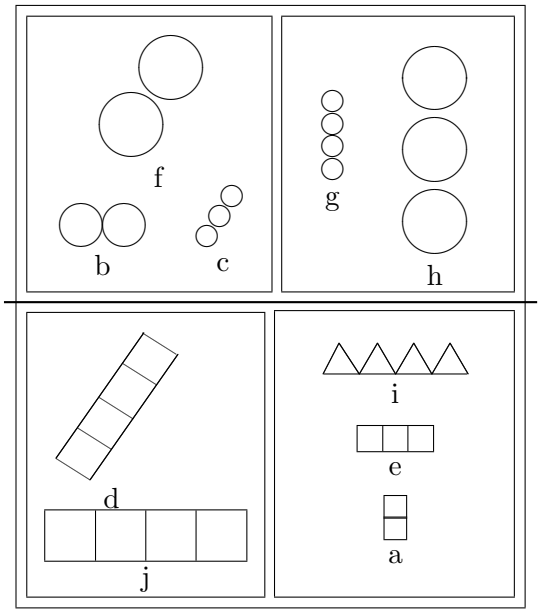


Fig. 20 Agrupamiento obtenido por el método PAF para k=2

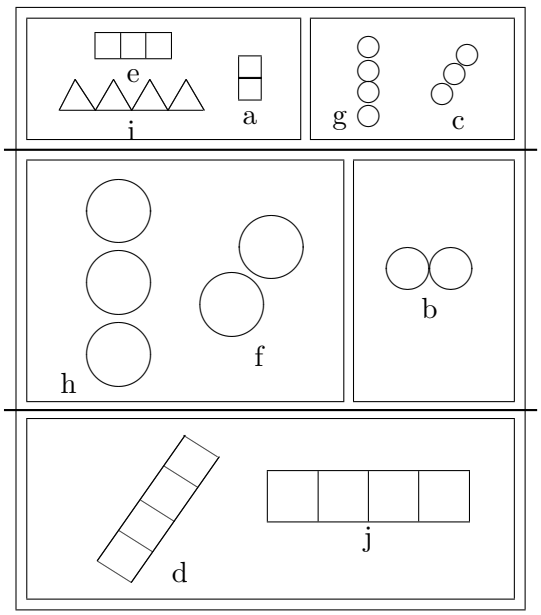


Fig. 21 Agrupamiento generado por PAF para k=3.

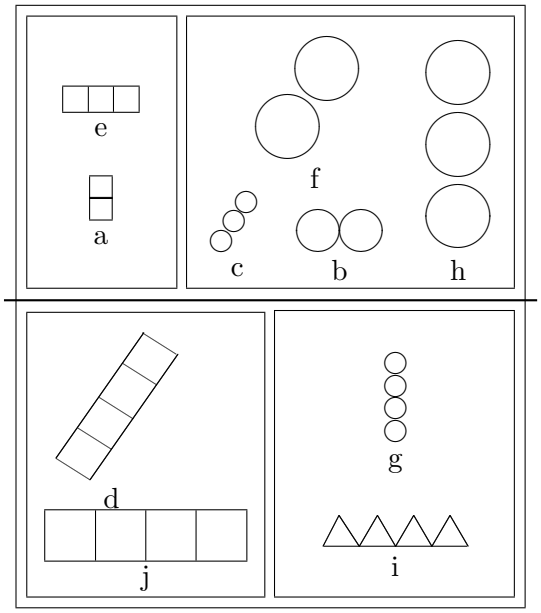


Fig. 22 Otra solución generada por PAF para k=3.

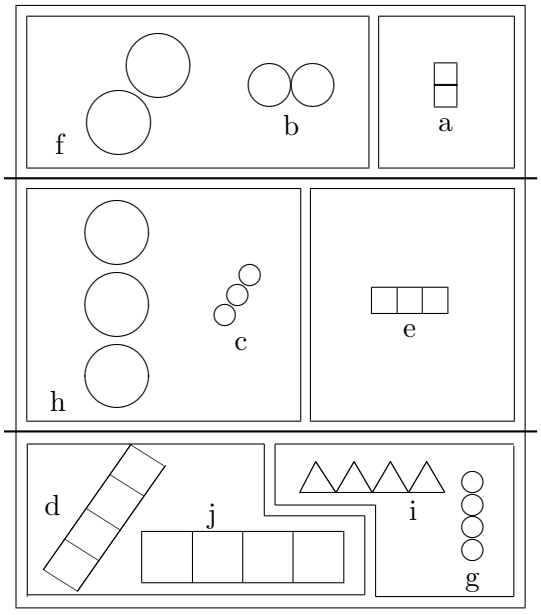


Fig. 23 Otra solución generada por PAF k=3.

Un experimento con personas sujetas a resolver este problema indica que simpatizaron con los agrupamiento obtenidos por PAF. Estas personas organizaron los objetos utilizando las propiedades más notorias en ellos. Las soluciones más usuales fueron:

$$[Forma = circular] \quad \text{vs} \quad [Forma \neq circular]$$
$$\text{y}$$
$$[Numero = 2] \ \text{vs} \quad [Numero = 3] \ \text{vs} \quad [Numero = 4]$$

Sin embargo las agrupaciones producidas por NUMTAX parecieron poco lógicas y más complejas que las descripciones de personas las cuales consideraron más natural y mucho más semejantes a los grupos producidos por PAF.

Los métodos presentados aquí determinan subcategorias de una coleción de objetos que puede ser aplicado a problemas del "mundo real" para dar un enfoque diferente a las tradicionales técnicas de agrupación mediante complejos que los describen a los objetos agrupados, sin basarse en medidas de distancias.

Capítulo 3

Agrupamiento Semántico

Haremos una revisión del modelo de agrupamiento semántico propuesto por C.C. Gotlieb y S. Kurmar en 1968. Este modelo fué realizado como una alternativa para resolver el problema de la recuperación de información.

Uno de los principales problemas en los sistemas de información es Cómo extraer la información pertinente que un usuario necesita?, este problema se agudiza cuando el usuario no está familiarizado con la colección de documentos en los que buscará y por lo tanto no puede asegurar que ha recopilado o recobrado los documentos más importantes de su interés. Un ejemplo de este problema es la búsqueda de una temática en una biblioteca. Es evidente que la persona no está familiarizada con todos los documentos existentes (libros, revistas, etc.) ni sabe cuáles de esos documentos son los que tratan con mayor amplitud la temática que a él le interesa.

El problema de la recuperación de información en documentos puede verse en cuatro etapas, la primera es la selección de los documentos, la segunda es la selección de los términos índice, éstos tiene la función de representar la temática de un documento y además dar la referencia al documento en que se encuentra, la tercera etapa es el agrupamiento de términos semánticamente parecidos, esta etapa también es conocida como formación del espacio de conceptos, la última etapa es la recuperación de los documentos para mostrarlos al usuario.

Como puede verse existen varios problemas que deben resolverse. Uno de los problemas presentes en las etapas antes mencionadas es de formar el espacios de conceptos, y sobre el mismo centraremos nuestra atención.

3.1 Estructura de Términos índice

Empezaremos dando algunas definiciones que nos servirán para entender el modelo.

Definición 3.1.1 *Los identificadores asignados al contenido de los documentos o textos guardados son conocidos como* **términos índice**, *palabras clave o descriptores.*

Cada término índice sirve para describir el contenido del texto. Así un número grande de términos índice puede ser asignados a cada documento o texto particular.

Definición 3.1.2 *Un sistema estructurado de índices consiste de un vocabulario de términos índice,* Υ*, y un mapeo de asociación semántica,* Γ*, dado la relación en la cual un término índice aparece con respecto a otros términos del sistema.*

Consideramos dos tipos de asociación semántica entre términos índice:

(a) **Inclusión** semántica, Γ_n
(b) **Parentesco** semántico, Γ_r

$y \in \Gamma_n\, x$ significa que y está incluido en x; $y \in \Gamma_r\, x$ significa que y esta emparentado con x; se define $\Gamma = \Gamma_n \bigcup \Gamma_r, y \in \Gamma\, x$ significa que y está incluido o emparentado con x, además $\Gamma^{-1}\, x = \{y | x \in \Gamma\, y\}$, es decir, todos los términos que emparentan o incluyen a x.

3.2 Medida de Asociación entre Términos índice

Para poder cuantificar el parecido semántico o la medida de asociación semántica entre dos términos índice, es necesario introducir la siguiente definición.

Definición 3.2.1 *Sea* $\phi(x) = \{x\} \bigcup \Gamma x \bigcup \Gamma^{-1} x$ *como el conjunto de todas las relaciones semántica inmediata de un término* x*.*

Este conjunto contiene además del término x, todos los términos que están contenidos o relacionados con x y aquellos términos que contienen o relacionan a x. Por estas características se le llama a $\phi(x)$ el alcance semántico de x.

La fórmula utilizada para definir la media de asociación semántica, entre un par de términos es:

$$\alpha(x, y) = \frac{| \phi(x) \cap \phi(y) |}{| \phi(x) \cup \phi(y) |} \tag{3.1}$$

End donde $|\phi|$ = número de elementos en el conjunto ϕ. Podemos observar que cuando el alcance semántico tanto de x como de y tengan muchos términos comunes, la medida de asociación $\alpha(x, y)$ es más grande; y toma valor 1 cuando ambos alcances semánticos coinciden. De manera inversa cuando los alcances semánticos de x y de y tiene pocos términos en común la medida de asociación $\alpha(x, y)$ se hace pequeña y cuando no tienen un sólo término en común la medida de asociacíon es 0.

A partir de la medida $\alpha(x, y)$ se define una pseudodistancia (no cumple la desigualdad triangular) entre dos términos x y y como $\delta(x, y) = 1 - \alpha(x, y)$, es decir la pseudodistancia entre dos términos, $\delta(x, y)$, es grande cuando x y y tienen una asociación muy pequeña y la pseudodistancia $\delta(x, y)$ es pequeña cuando x y y tienen una asociación muy grande (por comodidad en lo sucesivo usaremos la palabra distancia en lugar de seudodistancia). La función de distancia satisface las siguientes condiciones.

$$\begin{array}{lll} (i) & \delta(x, y) = 0 & \textit{Reflexiva} \\ (ii) & \delta(x, y) = \delta(y, x) & \textit{Simétrica} \end{array} \tag{3.2}$$

Note que $\delta(x, y) = 0 \not\Rightarrow x = y$; también $\delta(x, y)$ y $\alpha(x, y)$ toman valores entre 0.0 y 1.0.

Entonces, dado un vocabulario de términos índice, Υ, podemos tener una "Matriz de distancia" entre términos, los elementos de la cual son números reales en el intervalo [0, 1]. Si se reemplazan los elementos diferentes de cero

con unos, obtenemos la matriz de adyacencia de un grafo. Este grafo tiene una arista entre cualquier par de términos que posean una medida de asociación no nula. Muchos pares de términos tienen una medida asociación muy demasiado pequeña, lo que tendrá relacionado muchas aristas.

Definición 3.2.2 *Sea $\sigma(\tau)x$ para $x \in \Upsilon$ el conjunto de todos los términos cuya distancia a x es menor o igual que un umbral τ seleccionado en el rango [0, 1].*

Definición 3.2.3 *El grafo $(\Upsilon, \sigma(\tau))$ o $(\Upsilon, M(\tau))$ se llama τ- grafo de términos índice y su número de aristas es una función del umbral τ seleccionado.*

El τ-grafo tiene una arista entre cualquier par de términos con distancia menor o igual que τ.

3.3 Método de Agrupación

En este modelo dos métodos de agrupamiento de la teoría de grafos fueron analizados:

a) Componentes conexas de τ-grafo.
b) Subgrafos completos maximales de un τ-grafo.

3.3.1 Conceptos por componentes conexas

Para el primer método se tiene que las distancias entre diferentes componentes conexas de un τ-grafo constituyen una partición de los vértices del τ-grafo. Además si tenemos un τ-grafo $(V, M(\tau_1))$ con umbral τ_1, entonces decrementando el valor del umbral a un valor τ_2, obtenemos el grafo $(V, M(\tau_2))$, donde $M(\tau_2) \subset M(\tau_1)$. Si $(V, M(\tau_1))$ tiene $C_1, C_2, ..., C_{n1}$ componentes conexas del τ-grafo, entonces el grafo $(V, M(\tau_2))$ tendrá las componentes conexas $P_1, P_2, ..., P_{n_2}$ y se cumple que cada componente P_j está contenida en alguna componente C_1.

Cada componente conexa de un τ-grafo puede ser vista como $C^1 = \{x|y \in C^1, \alpha(x,y) \geq \tau\}$ un concepto tendrá una asociación mayor o igual que el umbral τ con al menos otro término índice del concepto y tendrá una asociación menor que τ con cualquier término que no esté en el concepto, aunque esto último pudiera cumplirlo también con términos índice del propio concepto.

La principal deficiencia de esta definición es el concepto resultante, algunos pares de términos no están fuertemente relacionados entre sí. En el caso extremo donde una larga cadena de términos se podría encontrarse en un solo concepto, la relación no es transitiva, y a menudo sucede que un par de términos distantes a lo largo de la cadena tendrá muy poca asociación semántica .

3.3.2 Conceptos por subgrafos completos maximales

El método de componentes conexas no es considerado en el modelo de Gotlieb y Kumar, precisamente por el hecho de que en un concepto (componente conexa) pueden existir términos que no estén relacionados.

Por esta razón es que el método de subgrafos completos maximales fue seleccionado para realizar el agrupamiento, en este caso cada término en el concepto (subgrafo completo maximal) tiene una asociación mayor o igual que τ con todos los términos del concepto, y una asociación menor con cualquier término que no esté en el concepto $C^2 = \{x|y \in C^2 \Rightarrow \alpha(x,y) \geq \tau\}$. Además los conceptos obtenidos no son particiones del τ-grafo; a los conceptos obtenidos por este método se les da el nombre de **conceptos claros**.

3.4 Unión semántica de conceptos

Al formar el espacio de conceptos lo que se pretende es que los conceptos sean tales que los términos pertenecientes a un concepto estén muy relacionados entre sí, es decir, que tengan gran parecido semántico. Por otra parte los conceptos del espacio deben estar débilmente conectados entre ellos, esto es, deben ser tan disjuntos como sea posible, lo cual significaría que no se confunden. Entonces si se comparan dos conceptos y éstos tienen muchos términos comúnes podríamos afirmar que son muy parecidos o que están muy cercanos

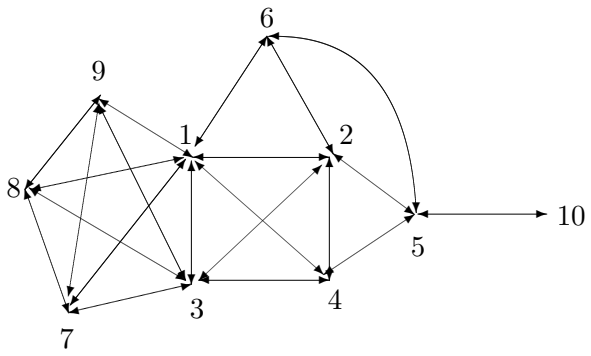
y de manera inversa, si como resultado de la comparación se tienen que ambos tienen pocos términos comunes o no tienen términos comunes, esto significaría que los conceptos son poco parecidos o que son conceptos diferentes. Para este modelo se definen una distancia (pseudodistancia) entre conceptos de la siguiente manera:

$$\delta(C_i, C_j) = 1 - \frac{|\, C_i \cap C_j|}{|\, C_i \cup C_j|} \tag{3.3}$$

De igual forma que en la distancia entre términos, la distancia entre conceptos no es necesario calcular ya que el cociente refleja el parecido entre conceptos que es lo que realmente interesa. Ahora se pueden comparar los conceptos formados antes y definir un nuevo τ-grafo el cual tiene a los conceptos como sus vértices, para los cuales dos vértices están conectados con una arista, si y sólo si la distancia entre los correspondientes conceptos es menor que una distancia umbral mínima δ_0. Los conceptos que pertenezcan a un subgrafo completo maximal del nuevo τ-grafo pueden ser mezclados para formar un nuevo concepto llamado **concepto difuso**, dado así un nuevo espacio de conceptos.

3.4.1 Técnica de Diccionario Fijo

En este modelo se presentan dos maneras de formar los conceptos: La primera técnica es denominada por Gotlieb y Kumar **mezcla de agrupamientos en un diccionario fijo**. Para ilustrar el procedimiento consideremos el grafo que se muestra en la siguiente figura.



Grafo

Sean $t_1, t_2, ..., t_{10}$ términos índice dados, τ y δ como antes. Consideremos dada la siguiente relación entre los términos:

Paso 1.- Especificar δ_0.

δ_0 es la distancia que se permitirá entre conceptos, para el ejemplo se tomará $\delta_0 = 0.50$.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Matriz de adyacencia.

Paso 2 .- Encontrar los subgrafos completos maximales. Los conceptos correspondientes a los subgrafos completos maximales son:

$$\begin{array}{ll} C_1 = \{t_1, t_2, t_3, t; 4\} & C_4 = \{t_2, t_5 t_6\} \\ C_2 = \{t_2, t_3, t_4, t_5\} & C_5 = \{t_1, t_3, t_7, t_8, t_9\} \\ C_3 = \{t_1, t_2, t_6\} & C_6 = \{t_5, t_{10}\} \end{array}$$

Paso 3.- Calcular la matriz de distancias entre conceptos.

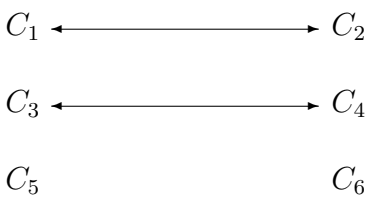
$$\begin{pmatrix} 0 & 0.4 & 0,6 & 0.833 & 0.714 & 1 \\ 0.4 & 0 & 0.833 & 0.6 & 0.875 & 0.8 \\ 0.6 & 0.833 & 0 & 0.5 & 0.875 & 1 \\ 0.833 & 0.6 & 0.5 & 0 & 1 & 0.75 \\ 0.714 & 0.857 & 0.857 & 1 & 0 & 1 \\ 1 & 0.8 & 1 & 0.75 & 1 & 0 \end{pmatrix}$$

Paso 4.- Preguntar si todas las distancias son mayores que δ_0.

En caso de ser afirmativo, se finaliza. De no cumplirse esto se va al siguiente paso.

Paso 5.- Definir la matriz de adyacencia entre conceptos. La matriz de adyacencia con $\delta_0 = 0.50$ (un valor arbitrario entre [0, 1]) es:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



Grafo.

Paso 6.- Encontrar los subgrafos completos maximales correspondientes al grafo representado en la matriz de adyacencia del paso 5. Los conceptos difusos, corresponden a los subgrafos completos maximales de este grafo, serán:

$$R_1 = C_1 \cup C_2 \qquad R_2 = C_3 \cup C_4$$
$$R_3 = C_5 \qquad R_4 = C_6$$

Paso 7.- Combinar los conceptos involucrados en los subgrafos completos maximales encontrados e ir al paso 3.

$$R_1 = C_1 \cup C_2 = \{t_1, t_2, t_3, t_4, t_5\}$$
$$R_2 = C_3 \cup C_4 = \{t_1, t_2, t_5, t_6\}$$
$$R_3 = C_5 = \{t_1, t_3, t_7, t_8, t_9\}$$
$$R_4 = C_6 = \{t_5, t_1 0\}$$

Paso 3.- La matriz de distancias para los nuevos conceptos es:

$$\begin{pmatrix} 0 & 0.5 & 0.75 & 0.833 \\ 0.5 & 0 & 0.875 & 0.8 \\ 0.75 & 0.875 & 0 & 1 \\ 0.833 & 0.8 & 1 & 0 \end{pmatrix}$$

Paso 4.- Como las distancias no son todas mayores que δ_0 entonces seguimos al paso 5.

Paso 5.- La matriz de adyacencia es:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Grafo.

Paso 6 y 7.- Continuando, encontramos los subgrafos completos maximales, entonces los nuevos conceptos difusos serán:

$$\varphi_1 = R_1 \cup R_2 = C_1 \cup C_2 \cup C_3 \cup C_4 = \{t_1, t_2, t_3, t_4, t_5, t_6\}$$
$$\varphi_2 = R_3 = C_5 = \{t_1, t_3, t_7, t_8, t_9\}$$
$$\varphi_3 = R_4 = C_6 = \{t_5, t_1 0\}$$

Paso 3.- En la nueva matriz, todas las distancias son más grandes que $\delta_0 = 0.5$ y entonces $\varphi_1, \varphi_2, \varphi_3$ son los conceptos difusos finales.

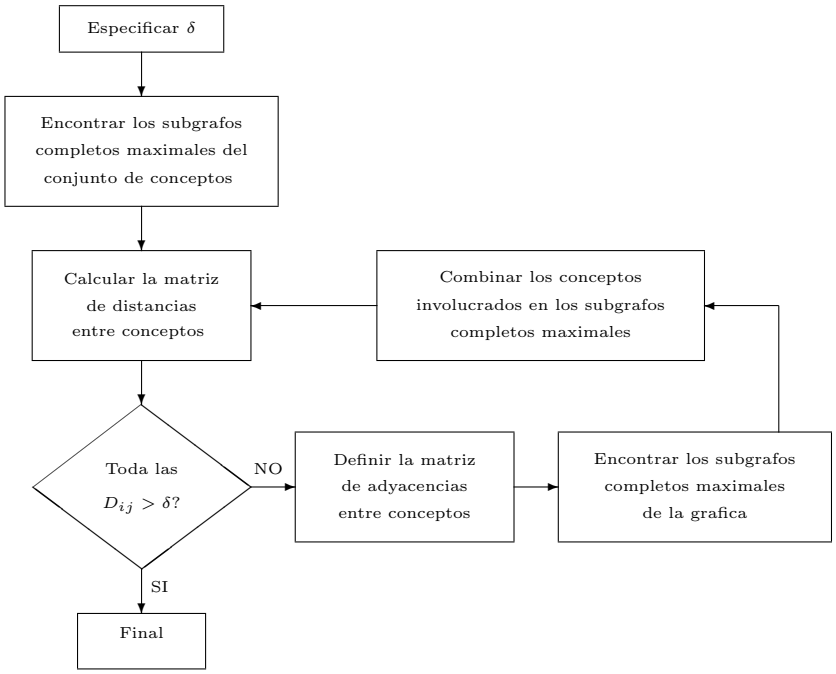


Fig. 24 Diagrama del agrupamiento con tecnica de diccionario fijo.

En el ejemplo anterior se ilustra cómo seis conceptos claros han sido sistemáticamente mezclados para dar tres conceptos difusos, usando $\delta_0 = 0.5$

En la figura 17 se muestran los pasos en la construcción de los conceptos con diccionario fijo.

3.4.2 Técnica de Diccionario Dinámico

La segunda técnica llamada **mezcla de agrupamientos en un diccionario dinámico**, permite a los agrupamientos crecer gradualmente durante los varios estados de la mezcla y los agrupamientos pueden ser generados localmente alrededor de un término requerido por el usuario. Para esta técnica la distancia umbral δ_0 se selecciona como la distancia más pequeña no nula de un

agrupamiento conteniendo el término requerido a cualquier otro agrupamiento.

Para ilustrar el procedimiento usaremos el mismo grafo del ejemplo anterior.

Paso 1.- Especificar δ_0 y el término requerido t.

Sea el término requerido t_4. El parámetro γ (criterio de parada) para el ejemplo lo dejaremos libre.

Paso 2.- Encontrar los subgrafos completos maximales son:

$$\begin{array}{ll} C_1 = \{t_1, t_2, t_3, t_4\} & C_4 = \{t_2, t_5, t_6\} \\ C_2 = \{t_2, t_3, t_4, t_5\} & C_5 = \{t_1, t_3, t_7, t_8, t_9\} \\ C_3 = \{t_1, t_2, t_6\} & C_6 = \{t_5, t_{10}\} \end{array}$$

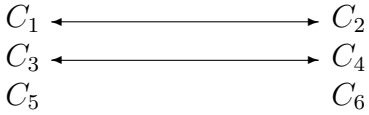
Paso 3.- Calcular las matrices de distancias entre conceptos. La matriz de distancias queda igual que para el ejemplo anterior.

Paso 4.- Poner δ_0 igual a la distancia más pequeña de un agrupamiento conteniendo el término requerido a cualquier otro agrupamiento.

Para este caso $\delta_0 = \delta(C_1, C_2) = 0.4$

Paso 5.- Definir la matriz de adyacencia entre conceptos. La matriz de adyacencia entre conceptos será:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



Grafo.

Paso 6.- Encontrar sólo aquellos subgrafos completos maximales, donde al menos uno de los nodos corresponde a un agrupamiento conteniendo el término requerido y mezclar los agrupamientos.

Definiremos los nuevos agrupamientos como aquellos subgrafos completos maximales del grafo donde al menos uno de los vértices (agrupamientos o conceptos) contiene al término requerido t_4; los otros agrupamientos se mantienen sin cambio. En el ejemplo tenemos:

$$
\begin{aligned}
D_1 &= C_1 \cup C_2 = \{t_1, t_2, t_3, t_4, t_5\} \\
D_2 &= C_3 = \{t_1, t_2, t_6\} \\
D_3 &= C_4 = \{t_2, t_5, t_6\} \\
D_4 &= C_5 = \{t_1, t_3, t_7, t_8, t_9\} \\
D_5 &= C_6 = \{t_5, t_{10}\}
\end{aligned}
$$

Paso 7.- Preguntar por la condición de terminación. Si se cumple terminamos, en caso contrario ir al paso 3.

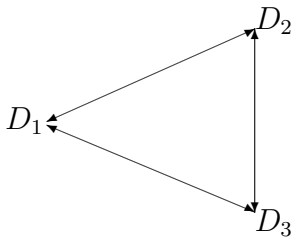
Paso 3.- La matriz de distancias es

$$
\begin{pmatrix}
0 & 0.67 & 0.67 & 0.75 & 0.833 \\
0.67 & 0 & 0.5 & 0.875 & 1 \\
0.67 & 0.5 & 0 & 1 & 0.75 \\
0.75 & 0.857 & 1 & 0 & 1 \\
0.833 & 1 & 0.75 & 1 & 0
\end{pmatrix}
$$

Ahora podemos seleccionar una nueva distancia umbral δ_0, por lo que tenemos $\delta_0 = \delta(D_1, D_2) = \delta(D_1, D_3) = 0.67$

Paso 5.- Tendremos la siguiente matriz de adyacencia

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



D_4 D_5

Grafo.

Paso 6.- Los nuevos agrupamientos son:

$$\varepsilon_1 = D_1 \cup D_2 \cup D_3 = \{t_1, t_2, t_3, t_4, t_5, t_6\}$$
$$\varepsilon_2 = D_4 = \{t_1, t_3, t_7, t_8, t_9\}$$
$$\varepsilon_3 = D_5 = \{t_5, t_{10}\}$$

Paso 7.- Si se cumple terminamos, en caso contrario ir al paso 3.

Paso 3.- La matriz de distancias es

$$\begin{pmatrix} 0 & 0.778 & 0.857 \\ 0.778 & 0 & 1 \\ 0.857 & 1 & 0 \end{pmatrix}$$

En el próximo paso δ_0 será igual a $\delta(\varepsilon_1, \varepsilon_2) = 0.778$, dando

$$F_1 = \varepsilon_1 \cup \varepsilon_2 = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9\}$$
$$F_2 = \varepsilon_3 = \{t_5, t_1 0\}$$

y finalmente δ_0 es seleccionado igual a $\delta(F_1, F_2) = 0.9$ y tenemos un único agrupamiento, $G = F_1 \cup F_2 = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$

Puede notarse que en el proceso de mezcla de agrupamientos δ_0 es el seleccionado automáticamente por el algoritmo, en un procedimiento de agrupamientos como éste, se necesita algún criterio de terminación.

Para un término requerido x, C_x^i son los agrupamientos a los que el término x pertenece en el i-ésimo nivel de agrupamiento, entonces $|\cup_i C_x^i|$ denota el número total de términos que han sido accesados en el i-ésimo nivel de agrupamiento.

Cuando los agrupamientos son conceptos claros $|\cup_i C_x^i| = |\sigma(\tau)x|$.El cociente $|\sigma(\tau)|/|\cup_i C_x^i|$ es utilizado como criterio de terminación. Este cociente da el porcentaje de términos vinculados con x en el agrupamiento i-ésimo.

Como puede apreciarse el problema de la formación del espacio de conceptos dentro de la recuperación de información es un problema de estructuración de universos o de clasificación sin aprendizaje, los objetos en este caso son los términos índice, por lo que se estructura el vocabulario V. La función de semejanza es la medida de asociación semántica x, a partir de ésta se constituye una matriz de distancia que como se mencionó anteriormente no es necesaria ya que x refleja el parecido semántico entre los términos y puede trabajarse directamente sobre $\|\alpha_{ij}\|_{|V|\times|V|}$ la matriz de medida de asociación, donde V es la cardinalidad del conjunto V. Aunque se analizan dos métodos de agrupamiento se trabaja solamente con el de subgrafos completos maximales y se rechaza el de componentes conexas porque en los agrupamientos que éste forma puede existir términos que no sean parecidos semánticamente, sin embargo, al mezclar agrupamientos se incluyen elementos que de igual forma no están relacionados.

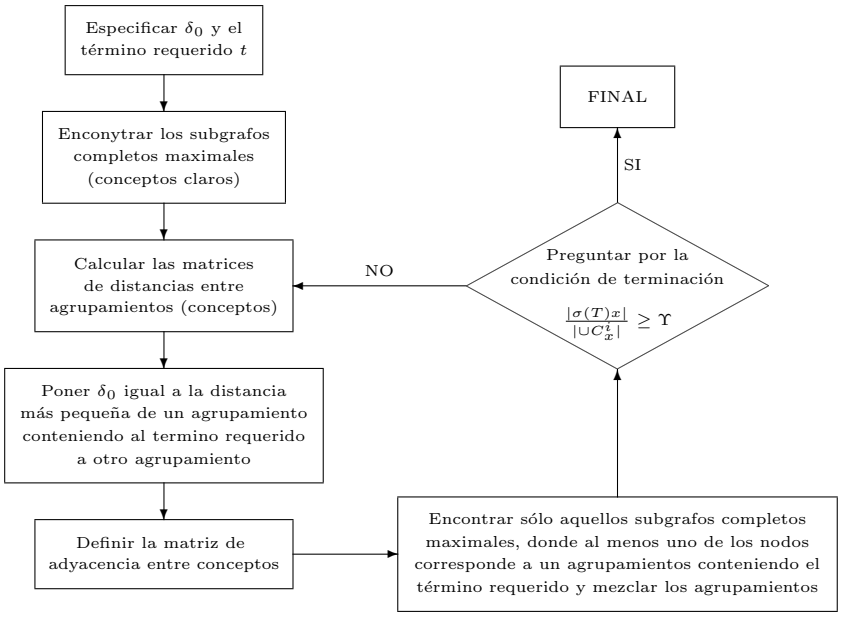


Fig. 25 Diagrama de agrupamiento con tecnica de diccionario dinamico.

Respecto a esto podemos decir que tanto las componentes conexas como los subgrafos completos maximales proporcionan información que es importante en cuanto al parecido entre los términos.

Limitaciones del modelo.

a) Se requiere saber cómo están relacionados los objetos de MI, según las relaciones de inclusión y parentesco (esto no siempre es posible).

b) El modelo utiliza como función de semejanza únicamente a la pseudodistancia $\delta(x, y) = 1 - \delta(x, y)$, que es a su vez calculada a partir de las relaciones entre los objetos.

c) Se utiliza un solo criterio agrupacional, el de subgrafos completos maximales.

d) El criterio agrupacional de subgrafos completos maximales produce en general un cubrimiento y no una partición del conjunto de objetos MI a estructurar.

e) El número de conceptos (agrupamientos) generados usando el criterio de subgrafos completos maximales puede ser una cantidad muy grande , llegando incluso a ser mucho mayor que la cantidad de objetos.

f) El modelo dice encontrar conceptos (agrupamientos) difusos que realmente no lo son ya que no proporcionan el grado de pertenencia de los objetos a los diferentes conceptos.

g) En los problemas prácticos en que se cuenta con un conjunto de objetos descritos por rasgos, que es muy común en las ciencias denominadas como poco formalizadas (Geología, Medicina, Sociología, etc.), el modelo no puede ser aplicado.

Debido a las limitaciones antes mencionadas se tiene que el universo de problemas a los que puede aplicarse el modelo es bastante reducido. Por este motivo, a continuación se plantea la generalización de este modelo de manera que el universo de problemas para los cuales pueda aplicarse sea mayor.

Generalización del modelo.

Del modelo de C. C. Gotlieb y S. Kumar mostrado en la sección anterior puede verse que, como primer paso para su aplicación se necesita de un proceso que, a partir de la muestra inicial MI de objetos, llegue a una matriz de semejanza, que representa precisamente la semejanza entre todos los objetos de MI y como segundo paso, sobre esta matriz se aplica el criterio agrupacional para estructurar la muestra, esto gráficamente puede verse en la siguiente figura.

Como se mencionó, el modelo anterior tiene varias limitaciónes; a continuación se incorporarán algunos conceptos que permiten generalizarlo y así poder aplicarlo a una gama más amplia de problemas.

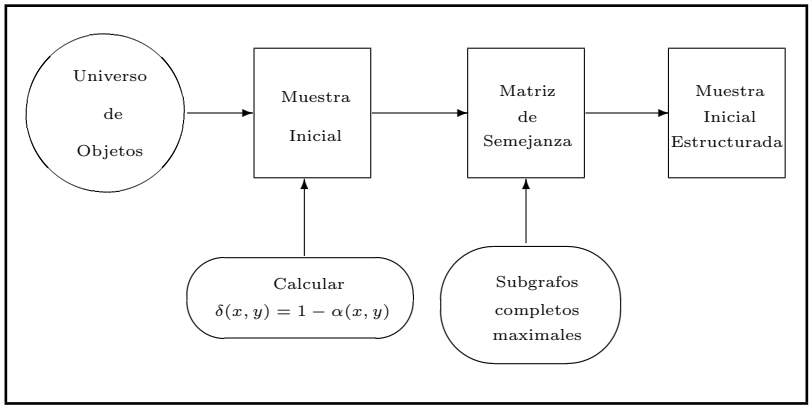


Fig. 26 Modelo de C.C. Gotlieb y S. Kumar

La primera limitación que se mencionó, es que se necesita saber cómo están vinculados según las relaciones de inclusión y parentesco, los objetos de la muestra inicial **MI**. Esto restringe el área de aplicabilidad del modelo ya que existen una cantidad grande de problemas que no se ajustan a estas condiciones, por ejemplo, en las ciencias denominadas poco formalizadas en donde se cuentan con un conjunto de objetos **MI** descritos por un conjunto de rasgos, el problema a solucionar es el mismo que el que pretende resolver el modelo de C. C. Gotlieb y S. Kumar; es decir, se desea estructurar a **MI** y de esta manera conocer relaciones y propiedades entre los objetos que a simple vista no pueden verse, sin embargo, este tipo de problemas no pueden ser resueltos por este modelo. En el enfoque lógico combinatorio del Reconocimiento de Patrones se ha trabajado en la problemática de definir funciones de semejanza en donde las descripciones de los objetos no están exclusivamente en términos cuantitativos, sino que se permiten que sean de cualquier tipo e incluso se da la posibilidad de que exista ausencia de información. Cabe hacer la aclaración que la definición de las funciones de semejanza son producto de un proceso de modelación matemática del problema en particular en que se esté trabajando y no sólo un ejercicio teórico. Puede entonces hacerse uso de las herramientas desarrolladas en este enfoque y de esta manera potenciar al modelo de C. C. Gotlieb y S. Kumar para poderlo aplicar a un número mayor de problemas.

También hay situaciones en las que uno o varios especialistas son los que

determinan en forma cuantitativa la medida de semejanza entre todos los objetos de **MI**, (esto es conocido como similaridades subjetivas) sin tener en cuenta relaciones de parentesco o inclusión. Como puede notarse hay varias maneras de tratar la semejanzas entre objetos y en general cada problema práctico determinado como debe medirse la semejanza de manera conveniente. Dar flexibilidad para la definición de la función de semejanza de acuerdo al problema que se está tratando en la práctica permite potenciar el modelo de Golieb y Kumar y hacerlo más general.

Otra limítate del modelo anterior es que utiliza un sólo criterio agrupacional. Dentro de la clasificación sin aprendizaje en el enfoque lógico-combinatorio de Reconocimiento de Patrones, existen criterios agrupacionales que de igual forma que el criterio de subgrafos completos maximales, están definidos a partir del cumplimiento de ciertas propiedades de las semejanzas entre los objetos; el uso de estos criterios potencia también al modelo de Gotlieb y Kumar, pero algo más importante que el uso de los criterios, es el uso del concepto de criterio agrupacional, ya que el mismo da la posibilidad de definir nuevos criterios de agrupamiento, según se requiera en la practica; es decir, las propiedades de semejanza entre los objetos se define de tal forma que resulten significativas para el problema en cuestión. Además, con la inclusión de otros criterios agrupacionales, las estructuras obtenidas pueden conformar una partición de **MI** y no sólo un cubrimiento.

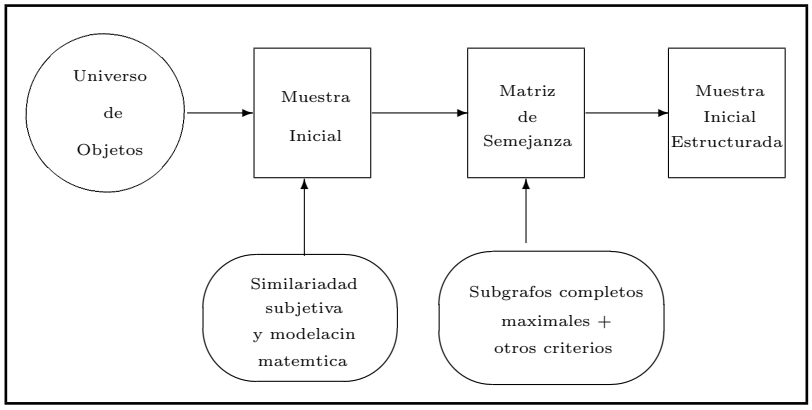


Fig. 27 Modelo Generalizado de C.C. Gotlieb y S. Kumar

Como puede apreciarse la incorporación de los conceptos de la teoría del enfoque lógico-combinatorio del Reconocimiento de Patrones, tanto para definir funciones de semejanza como criterios agrupacionales permite generalizar el modelo de C. C. Gotlieb y S. Kumar, y de esta forma puede ser aplicado a una cantidad mayor de problemas.

3.5 Programa

Este código fuente se realizó sobre la plataforma de programación C++. Este programa consta de 8 funciones de las cuales dará una explicación de cada una de ellas. El método empleado en este programa fue el Agrupamiento Semántico con diccionario fijo.

Las primeras líneas del programa sólo son librerías que nos permiten utilizar herramientas de impresión en pantalla, matemáticas, entradas y salidas de datos etc.

```
// Agrupamiento Conjuntivo con tcnica de diccionario fijo
#include <iostream>
#include <math.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
```

La función Impresión tiene como parámetros entrantes una matriz y un entero que es el tamaño de esta matriz. Como su nombre lo dice su función es el de imprimir una matriz con entradas 0's y 1's, es decir, una matriz de adyacencia.

```
/*****************FUNCIN DE IMPRESIN.******************/
void Imprime(int *A, int n)
{
   int i,j;

   cout<<"\n\n\tLa matriz de Adyacencia es:\n\n";
```

```
    for (i=0; i<n; i++)
    {
        if(A[i*n]>=0)
        {
            cout<<"\t\t\t";
            for(j=0; j<n; j++)
            {
                if(A[i*n+j]>=0)
                cout <<A[i*n+j] <<" ";
            }
            cout<< "\n";
        }
    }
    cout<<"\n\n";
}
```

Esta parte es la de mayor importancia porque es la que dice cuáles son los términos que tienen relación entre sí y forman los conceptos claros (agrupaciones). Los parámetros entrantes es la matriz inicial $*A$ de adyacencia, el parámetro $*C$ es una matriz que guarda los complejos obtenidos en ésta función y n sigue siendo el tamaño de la matriz, finalmente regresa un valor l que es el número de complejos. La forma en como son obtenidos los complejos es de acuerdo a una comparación de términos relacionados. Una matriz auxiliar es la que nos facilita los cálculos convirtiendo los términos índice en números para que la comparación y eleminación de sea mucho más rápida.

```
/***********FUNCIN DE RELACIN DE TRMINOS.************/
int Relaciona(int *A, int *C, int n)
{
    int i,j,k,l;
    float *aux;
    double cn[2*n];

    aux = new float [(2*n)*n];

    for(i=0;i<n;i++)
```

```
{
    for(j=0;j<n;j++)
    {
        aux[i*n+j]=aux[n*n+(i*n)+j]=A[i*n+j];
    }
    for(j=0;j<n;j++)
    {
         if(aux[i*n+j]==1)
         {
             for(k=0;k<n;k++)
             {
                  if(A[j*n+k]==0)
                  {
                       aux[i*n+k]=0;
                  }
             }
         }
         if(aux[n*n+(i*n)+(n-j)]==1)
         {
             for(k=0;k<n;k++)
             {
                  if(A[(n-j)*n+k]==0)
                  {
                       aux[n*n+(i*n)+k]=0;
                  }
             }
         }
    }
}
for(i=0;i<2*n;i++)//convertiremos la matriz auxiliar
a numeros sin importar si este es igual a los trminos
{
    cn[i]=0;
    l=0;
    for(j=0;j<n;j++)
    {
        if(aux[i*n+j]>0)
        {
```

```
                cn[i]=cn[i]+ (j+1)*pow(10,l);
                l++;
            }
        }
    }
    for(i=0;i<2*n;i++)
    {
        for(j=0;j<n;j++)
        {
            C[i*n+j]=0;
        }
    }
    l=0;//elimina trminos iguales e introduce elementos
    a la matriz de complejos
    for(i=0;i<2*n;i++)
    {
        if(cn[i]!=0)
        {
            k=0;
            for(j=0;j<n;j++)
            {
                if(aux[i*n+j]>0)
                {
                    C[l*n+k]=j+1;
                    k++;
                }
            }
            l++;
            for(j=i+1;j<2*n;j++)
            {
                 if(cn[i]==cn[j])
                 cn[j]=0;
            }
        }
    }
    return(l);
}
```

Los parámetros entrantes de la función son $*C$ la matriz de complejos, NoC el número de complejos, n el tamaño de la matriz y *opcion* que es la forma en como la matriz de complejos será mostrada en pantalla.

```
/***********FUNCIN DE IMPRESIN DE COMPLEJOS***********/
void Imprimecomplejos(int *C, int NoC, int n, int opcion)
{
    int i,j;

    if(opcion==1)
    {
        cout<<"\n\n";
        for (i=0; i<NoC; i++)
        {
            cout<<"\t\tC" <<i+1 <<"={";
            for(j=0; j<n; j++)
            {
                if(C[i*n+j]!=0)
                cout<<"t" <<C[i*n+j] <<",";
            }
            cout<< "\b}\n";
        }
        cout<<"\n\n";
    }
    if(opcion==2)
    {
        cout<<"\n\n";
        for (i=0; i<NoC; i++)
        {
            cout<<"\t\tR" <<i+1 <<"={";
            for(j=0; j<n; j++)
            {
                if(C[i*n+j]!=0)
                cout<<"C" <<C[i*n+j] <<",";
            }
            cout<< "\b}\n";
        }
```

```
        cout<<"\n\n";
    }
}
```

Las distancias entre conceptos es otro de los pasos importantes del algoritmo ya que une conceptos claros que tiene menos distancia entre si, siempre y cuando cumpla con delta (en rango permitido entre conceptos). Esta función tiene como parámetros entrantes a C la matriz de conceptos, D la matriz que guardará las distancias, NoC el número de conceptos y n el tamaño de la matriz. La Función "Distancias", en primera instancias cuenta en número de términos que contiene el concepto para después comparar términos entre conceptos. La distancia es obtenida de acuerdo con la formula 3.3.

```
/*****************FUNCIN DE DISTANCIAS******************/
void Distancias(int *C, float *D, int NoC, int n)
{
    int a,b,i,j;
    float iguales;
    float No[NoC];

    for(i=0;i<NoC;i++)
    {
        No[i]=0;
        if(C[i*n]>0)
        {
            for(j=0;j<n;j++)
            {
                if(C[i*n+j]>0)
                {
                   No[i]++;
                }
            }
        }
    }
    for(i=0;i<NoC;i++)
    {
```

```
        for(a=0;a<NoC;a++)
        {
           iguales=0;
           for(j=0;j<n;j++)
           {
               if(C[i*n+j]>0)
               {
                    for(b=0;b<n;b++)
                    {
                        if(C[i*n+j]==C[a*n+b])
                        {
                            iguales++;
                            break;
                        }
                    }
               }
           }
           D[i*NoC+a]=1-(iguales/(No[i]+No[a]-iguales));
           if((No[i]==0) || (No[a]==0))
           {  D[i*NoC+a]=-1;  }
        }
    }
}
```

Como su nombre lo dice la función no tiene mayor importancia que la de imprimir en pantalla las distancias entre conceptos. Tiene por parámetros entrantes D matriz de distancias, NoC el número de conceptos obtenidos.

```
/*****************FUNCIN DE IMPRESIN*****************/
void Imprimedistancias(float *D, int NoC)
{
    int i,j;

    cout<<"\n\n";
    for (i=0;i<NoC;i++)
    {
        cout<<"\t";
```

```
        for(j=0;j<NoC;j++)
        {
            printf("%.3f ", D[i*NoC+j]);
        }
        cout<<"\n";
    }
    cout<<"\n\n";
}
```

Una vez que las distancias son calculadas la función "Adyacencia" nos permite calcular una matriz que dice cuales conceptos son los que serán unidos para dar paso a nuevos que tengan una relación entre si. El nuevo parámetro AD es la matriz que guardará con 1's lo conceptos mayores a *delta*.

```
/***************FUNCIN DE ADYACENCIA******************/
void Adyacencia(float *D, int *AD, int NoC, float delta)
{
    int i,j;

    for(i=0;i<NoC;i++)
    {
        for(j=0;j<NoC;j++)
        {
            if((D[i*NoC+j]>=0) && (D[i*NoC+j]<=delta))
            {  AD[i*NoC+j]=1;  }
            if(D[i*NoC+j]>delta)
            {  AD[i*NoC+j]=0;  }
            if(D[i*NoC+j]==-1)
            {  AD[i*NoC+j]=-1;  }
        }
    }
}
```

"Damenumero" es la función que reconoce las distancias menores a la delta dada, este calculo es la condición de terminación para el programa. Una vez que la distancias son mayores a delta todos los términos índices ya tiene un

concepto al que pertenecen. La función "Damenumero" compara las distancias y regresa un valor que dice si el programa puede continuar o no.

```
/*****FUNCIN IDENTIFICA DISTANCIAS MENORES A DELTA******/
float Damenumero(float *D, int NoC, float delta)
{
    int i,j,valor=1;

    for(i=0;i<NoC;i++)
    {
        for(j=i+1;j<NoC;j++)
        {
            if(D[i*NoC+j]<=delta)
            {
                valor=0;
                break;
            }
        }
    }
    return(valor);
}
```

Esta parte es la que reestructura la matriz inicial de términos índice para dar paso a una nueva iteración del algoritmo. La reestruccturación básicamente es relacionar términos, con ayuda del algoritmo, que anteriormente no tenían relación.

```
/*****************FUNCIN DE ADYACENCIA*****************/
int *Uneconceptos(int *A,int n,int *SC,int NoSC,int *C,int NoC)
{
    int a,t,i,j,k;

    int *Nuevo;
    Nuevo = new int[NoSC*n];

    for(a=0;a<NoSC;a++)
```

```
    {
        for(k=0;k<n;k++)
        {
            Nuevo[a*n+k]=0;
        }
    }
    for(a=0;a<NoSC;a++)
    {
        k=0;
        for(t=1;t<=n;t++)
        {
            for(i=0;i<NoC;i++)
            {
                if(SC[a*NoC+i]>0)
                {
                    for(j=0;j<n;j++)
                    {
                        if(C[ (SC[a*NoC+i] -1)*n +j ]==t)
                        {
                            Nuevo[a*n+k]=t;
                            k++;
                            i=NoC;
                            break;
                        }
                    }
                }
            }
        }
    }
    return(Nuevo);
}
```

Finalmente llegamos al programa principal el cual emplea las funciones anteriores de acuerdo con los pasos en el algoritmo. En primera instancia pregunta el número de términos que serán agrupados, posteriormente se debe introducir *delta* el grado de pertenencia que deseamos obtener en los conceptos.

```
/***************EMPIEZA EL PROGRAMA PRINCIPAL************/
```

```
main(void)
{
    int i,j,k,n,NoC,NoSC;
    float delta,Distante;

    cout<<"\nEste programa utiliza el algoritmo de Agrupamiento
    Conceptual Conjuntivo, para  la elaboracin de agrupamientos
    de objetos. Usando la tcnica de Diccionario Fijo;

    cout<<"\nIntroduce el numero de Trminos ndice: ";
    cin>>n;
    cout<<"\nIntroduce delta: ";
    cin>>delta;

    int *A, *C, *AD, *SC;
    A = new int[n*n];
    C = new int[(2*n)*n];
    AD = new int[(2*n)*(2*n)];
    SC = new int[(2*n)*n];
    float *D;
    D = new float[(2*n)*(2*n)];

    for (i=0; i<n; i++)//Inicia una matriz aleatoria con
    el numero de trminos introducidos
    {
        for(j=0; j<n; j++)
        {
            A[i*n+j]=A[j*n+i]=rand()%2;
            if(i==j)
            A[i*n+j]=A[j*n+i]=1;
        }
    }
    Imprime(A,n);
    NoC=Relaciona(A,C,n);
    do
     {
         cout<<"\tLos Conceptos Claros correspondientes
         son los siguientes:";
```

```
        Imprimecomplejos(C,NoC,n,1);
        Distancias(C,D,NoC,n);
        cout<<"\tLa matriz de distacias entre conceptos es:";
        Imprimedistancias(D,NoC);
        cout<<"\tcon delta=" <<delta;
        Adyacencia(D,AD,NoC,delta);
        Imprime(AD,NoC);

        Distante=Damenumero(D,NoC,delta);

        if(Distante<=delta)
        {   NoSC=Relaciona(AD,SC,NoC);
            cout<<"\tLos Conceptos Claros relacionados
            son:";
            Imprimecomplejos(SC,NoSC,NoC,2);
            C=Uneconceptos(A,n, SC,NoSC, C,NoC);
            NoC=NoSC;
        }
        else
        {   cout<<"\tTodas las distancias ya son mayores
            que " <<delta <<", por lo que no existen mas
            relacin de trminos";
            cout<<", y quedan como Conceptos Claros los
            siguientes:";
            Imprimecomplejos(C,NoC,n,1);
        }
      }
      while(Distante<=delta);

system("PAUSE");
}
```

La simulación se hace en base a una matriz obtenida de modo aleatorio ya que por el momento este programa no contiene alguna base de datos ya que el usuario será el que oriente el programa a sus necesidades.

Capítulo 4

Conclusiones

En este trabajo se presentó el Reconocimiento de patrones en base al enfoque conceptual, el cual se vieron 2 tipos de agrupamiento, Conjuntivo y Semántico. Estos tipos de agrupamiento, siguiendo el problema de la clasificación sin aprendizaje, sirven para revelar la estructura interna de una colección de objetos mediante su jerarquización en subcategorías (agrupaciones). Cada agrupación obtenida tiene una apropiada descripción general y fueron formadas de acuerdo en el paradigma del conjunto cociente que supone que los agrupamientos serán ajenos entre sí.

El agrupamiento conceptual conjuntivo nos brinda 4 métodos (PAF, P, PS y S) para obtener una agrupación con sentido lógico, es decir, cada una tiene asociado un concepto. Estas descripciones están basadas en las técnicas de inferencia inductiva aplicada y en el lenguaje descriptivo V L1 (Sistema Lógico de Variables Valuadas), introducidas por Michalski.

El primer método (PAF) busca los mejores representantes (eventos) de cada agrupación, la limitante del método es que puede o no darnos una agrupación optima ya que está restringida a número de iteraciones que le método realizará.

Los otros 3 métodos (P, PS y P) hacen una búsqueda ramificada sobre las descripciones de objetos para obtener el agrupamiento óptimo, para ello trabaja con una combinación de complejos ordenados de acuerdo con al orden de sus dispersiones esto en ocasiones representa una desventaja cuando el número de complejos es demasiado grande lo cual ocasiona un gran costo computacionalmente.

El agrupamiento Semántico es utilizado para hacer una recolección de información sobre algún tema específico y utiliza los siguientes métodos: Diccionario Fijo y Dinámico. Donde el primero agrupa términos índices de acuerdo a

una distancia fija entre cada término. El segundo trabaja de manera análoga, la diferencia entre ellos radica en que éste método cambia el umbral de tolerancia entre conceptos en cada iteración. Y toma la distancia mínima entre términos de cada iteración del algoritmo. Además utilizan grafos y matrices de adyacencia para la visualización de términos indices relacionados entre si.

El reconocimiento de patrones en si es un tema muy extenso ya que en cada uno de los problemas encierra una gama de enfoques que con el tiempo van mejorando y aumentando su aplicación en diferentes areas. Es por eso que se convirtió en uno de los proyectos más ambiciosos y fascinantes de las últimas décadas.

Bibliografía

[1] Ryszard S. Michalski and Robert Stepp, *Revealing Conceptual Structure in Data by Inductive Inference*. Capitulo del libro, Machine Intelligence 10, D. Michie, J. E. Hayes and Y-H Pao, Editores. John Wiley & Sons Publishing, New York, 1982.

[2] Ryszard S. Michalski *Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and an Algorithm for Partitioning Data Into Conjuntive Concepts*, (Special issue on knowledge acquisition and induction), Policy Abalysis and Information Systems. No.3, 219-244.

[3] Anderberg, M. R., (1973). *Cluster Anlysis for Applications*. New York and London: Academic Press.

[4] Diday, E., & Simon, J. C., (1976). Clustering analysis, *Communication and Cbernetics 10*, (ed. Fu, K. S.). Berlin, Heidelberg, New York: Springer Verlag.

[5] Gowda, K. Chidananda, & Krishna, G., (1978). Disaggregative clustering using the concept of nearest neighbourhood, *IEE Trans. On Systems, Man and Cybernetics, SMC-8, No. 12* 888-894.

[6] Hanani, U., *Multicriteria dynamic clustering*. Reports of IRIA, France (1979).

[7] Ryszard S. Michalski, *Variable-value logic:System VL_1, Procceding of the 1974 International Symposium on Multiple-Valued Logic,* West Virginia University, Morgantown, West Virginia (May 29-31, 1974).

[8] Ryszard S. Michalski, *Synthesis of optimal and quasi-optimal variable-valued logic formulas Proccedings of the 1975 International Symposium on Multiple-Valued Logic,* Bloomington, Indiana (May 13-16, 1975).

[9] Ryszard S. Michalski, *Studies in Inductive Inference and Plausible Reasoning* a research proposal to NSF, (November 1978).

[10] Ryszard S. Michalski, *A Planar Geometrical Model for Representing Multidimensional Discrete Spaces and Multiple-Valued Logic Function* Report No. 897, Department of Computer Science, University of Illinois, Urbana, Illinois (1978).

[11] T. Nilsson, *Principles of Artificial Intelligence* (Toiga Publishing Company, 1980).

[12] R. Stepp, *Learning Without Negative Examples via Variable-Valued Logic Characterizations: The Uniclass Inductive Program AQ7UN1,* Report 982, Department of Computer Science, University of Illinois, Urbana, Illinois (July 1979).

[13] R. Stepp, *A Description and User's Guide for CLUSTER/PAF-A Program for Conjuntive Conceptual Clustering,* to appear as Report of the Department of Computer Science, University of Illinois, Urbana, Illions (1980).

[14] S. Watanabe, *Pattern Recognition as an Inductive Process, in Methodologies of Pattern Recognition* Watanabe, Ed. (Academic Press, New York, 1968).

[15] S. Watanabe, *Knowing and Guessing: A Quantitative Study of Inference and Information* (Wiley, New York, 1969).

[16] P. H. Winston, *Artificial Intelligence* (Addison-Wesley, Reading, Mass., 1977).

[17] Ryszard S. Michalski and J. B. Larson, *Selection of Most Representative Training Examples and Incremental Generation of VL_1 Hypotheses: The Underlying Methodogy and the Decription of Programs ESEL and AQ11*, Report No. 867, Department of Computer Science, University of Illinois, Urbana, Illinois (1978).

[18] Sokal, R. R., & Sneath, P. H. *Principles of Numerical Taxonomy* San Francisco: Freeman.

[19] C. C. Gotlieb and Kumar *Semantic Clustering of Index Terms* Jornual of the Association for Computing Machinery, Vol. 15, No. 4, October 1968, pp. 493-513.

[20] V. E. Giuliano and P. E. Jones, Linear Associative Information Retrieval. In howerton, P. W., and Weeks, D. C. (Eds.), *Vistas in Information Handling, Vol. 1,* Spartan Books, Washington, D. C., 1963, Ch. 2, pp. 30-46.

[21] G. Salton. Associative document retrieval techniques using bibiliographic information. J. ACM 10, 4 (Oct. 1963), 440-457.

[22] M. E. Stevens (Ed.). Proc. Symposium in Statistical Association Methods for Mechanized Documentation. US Government Printing Office, NBS Misc. Pub. 269, Dec. 1965.

[23] H. E. Stiles. The association factor in information retrieval. J. ACM 8, 2 (April 1961), 271-279.

[24] K. Sp¿arck-Jones. Experiments in semantic classification. *Mech. Transl. Comput. Linquist. 8* (June and Oct. 1965), 92-112.

[25] P. A. W. Lewis, P. B. Baxendale and J. L. Bennet. Statistical Discrimination of the Synonymy/Antonymy Relationship Between Words. J. ACM 14, 1 (Jan. 1967), 20-44.

[26] Library of Congress. *Subject Headings* (6th ed.). Washington, D. C., 1957.

[27] Engineers Joint Coucil. *Thesaurus of Engineering Terms* (1st ed.). New York, May 1964.

[28] US Deparment of Health, Eduction and Welfare. *MEDALRS Medical Subject Headings* (3rd ed.). Washington, D. C., Jan. 1964.

[29] C. Berge, *The Theory of Graphs and Its Aplication.* Wiley, New York, 1962.